



HAL
open science

Étude d'endommagement sous choc en dynamique moléculaire par développement d'un algorithme d'analyse in-situ massivement parallèle

Killian Babilotte

► **To cite this version:**

Killian Babilotte. Étude d'endommagement sous choc en dynamique moléculaire par développement d'un algorithme d'analyse in-situ massivement parallèle. Science des matériaux [cond-mat.mtrl-sci]. Université Paris-Saclay, 2024. Français. NNT : 2024UPASP085 . tel-04761505

HAL Id: tel-04761505

<https://theses.hal.science/tel-04761505v1>

Submitted on 31 Oct 2024

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.

Étude d'endommagement sous choc en
dynamique moléculaire par
développement d'un algorithme
d'analyse in-situ massivement parallèle
Damage study under shock with molecular dynamics by
developing a massively parallel in-situ analysis algorithm

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°564 : Physique en Île-de-France (PIF)
Spécialité de doctorat : Physique
Graduate School : Physique, Référent : Faculté des sciences d'Orsay.

Thèse préparée dans l'unité de recherche **LMCE (Université Paris-Saclay, CEA)**, sous
la direction de **Laurent VIDEAU**, Directeur de recherche, la co-direction de
Jean-François MOLINARI, Professeur, et le co-encadrement de **Laurent SOULARD**,
Ingénieur-Chercheur, et **Alizée DUBOIS**, Ingénieure-chercheuse.

Thèse soutenue à Paris-Saclay, le 26 septembre 2024, par

Killian BABILOTTE

Composition du jury

Membres du jury avec voix délibérative

Raymond NAMYST Professeur des universités, Université de Bordeaux	Président
Thibault DE RESSEGUIER Directeur de recherche CNRS, Institut P'	Rapporteur & Examineur
Lionel LACASSAGNE Professeur des universités, Sorbonne Université	Rapporteur & Examineur
Cindy ROUNTREE Directrice de recherche, CEA Saclay	Examinatrice
Paul FOSSATI Ingénieur-Chercheur, CEA Saclay	Examineur
Laurent COLOMBET Directeur de recherche, LIHPC - CEA, DAM, DIF	Examineur

Titre : Étude d'endommagement sous choc en dynamique moléculaire par développement d'un algorithme d'analyse in-situ massivement parallèle

Mots clés : Dynamique moléculaire, HPC, Étiquetage en composantes connexes (ECC), Simulation, Endommagement dynamique

Résumé : Dans le contexte de l'étude en condition extrême de la matière condensée, de nombreuses questions restent encore ouvertes pour comprendre certains phénomènes, notamment à cause de la difficulté de leur étude expérimentale. Selon les phénomènes étudiés, l'insuffisance de données expérimentales peut limiter la modélisation des phénomènes, ou, lorsque les modèles existent, elle peut empêcher leur calibration ou d'évaluer leur pertinence. La dynamique moléculaire (DM) est une technique de choix pour étudier ces phénomènes car elle permet de simuler la réponse d'un système uniquement à partir de la description des interactions atomiques, pour lesquelles des modèles bien éprouvés existent souvent. Cependant, une des contreparties de cette technique est qu'elle nécessite une grande quantité d'atomes pour atteindre une échelle suffisamment représentative des phénomènes étudiés et pouvoir ainsi les observer en s'affranchissant au mieux des effets de tailles. Dans la dernière décennie, de nombreux efforts ont été fait pour adapter les codes de DM aux dernières architectures de supercalculateurs et permettent maintenant de simuler des systèmes à plusieurs milliards d'atomes. Ces simulations de DM sont devenues de véritables expériences numériques et génèrent de gigantesques quantités de données à traiter. Désormais, le temps de post-traitement et d'analyse de ces données peut devenir prépondérant comparativement à celui de la simulation et devient un enjeu essentiel.

Dans cette thèse, nous avons développé un algorithme d'analyse de simulation de DM capable d'analyser ces simulations de plusieurs milliards d'atomes avec un coût de calcul de l'ordre du pourcent du temps total de simulation. L'algorithme détecte et caractérise des zones d'intérêts dans la simulation à partir de critères définis par l'utilisateur, le rendant ainsi très versatile. Nous avons fondé cet algorithme sur les techniques d'étiquetage en composantes connexes parallélisées en mémoire partagée que nous avons étendu à un parallélisme hybride en mémoire partagée et distribuée. Cette extension à la parallélisation en mémoire distribuée permet d'une part de traiter la quantité de données générée par les simulations, et nous a permis d'autre part d'incorporer notre analyse dans le code de DM du CEA exaStamp pour un traitement in-situ des simulations. Ce traitement in-situ de la simulation permet de diminuer la quantité de données à stocker sur disque et d'augmenter la fréquence d'analyse sur une simulation comparé à ce qui pourrait être fait en post-traitement. Une caractérisation des erreurs associées aux mesures réalisées par notre analyse a été effectuée, le rendant directement exploitable par des physiciens. Nous l'avons en particulier appliqué à divers cas physiques : d'éjection de matière type "micro-jetting" et "splashing" où des détections d'agrégats sont nécessaires, ainsi que des simulations d'endommagement sous choc (écaillage) impliquant la détection de vides.

Title : Damage study under shock with molecular dynamics by developing a massively parallel in-situ analysis algorithm

Keywords : Molecular dynamics, HPC, Connected components labeling (CCL), Simulation, Dynamic damage

Abstract : In the perspective of the study of condensed matter under extreme conditions, many questions remain open in order to understand some phenomena, due to the difficulty of studying them experimentally. Depending on the phenomena studied, the lack of experimental data may prevent the modeling of these phenomena, or, when models do exist, it may prevent their calibration or settling which one to use. Molecular dynamics (MD) is the technique of choice for studying these phenomena, as it enables the response of a system to be simulated solely from the description of atomic interactions, for which well-tested models often exist. However, one of the disadvantages of this technique is that it requires a large number of atoms to reach a scale that is sufficiently representative of the phenomena being studied, in order to be able to observe them and overcome size effects as far as possible. Over the past decade, numerous efforts have been made to adapt DM codes to the latest supercomputer architectures, enabling us to simulate systems with several billion atoms. These MD simulations have become true numerical experiments, generating huge quantities of data to be processed. Nowadays, the time required for post-processing and analysis of this data can become more important than for simulation, and is becoming an essential issue.

In this thesis, we have developed an analysis algorithm for MD simulation capable of processing multi-billion-atom workloads with a computational cost of the order of one percent of the total simulation time. The algorithm detects and characterizes areas of interest in the simulation based on user-defined criteria, making it highly versatile. We have based this algorithm on connected components labeling techniques, parallelized in shared-memory, which we have extended to hybrid shared and distributed memory parallelism. This extension to distributed-memory settings not only enables us to handle the quantity of data generated by the simulations, but has also enabled us to incorporate our analysis into the CEA MD code *exaStamp* for in-situ processing of the simulations. This in-situ processing of the simulation reduces the amount of data to be stored on disk and increases the frequency of analysis on a simulation compared to what could be done in post-processing. A characterization of the errors associated with the measurements made by our analysis has been carried out, making it directly exploitable by physicists. In particular, we have applied it to a number of physical cases : micro-jetting and splashing where aggregate detection is required, as well as shock damage simulations (spalling) involving void detection.

Remerciements

Avant de remercier toutes les personnes ayant activement contribué à cette aventure qu'est la thèse, je voudrais remercier le Pr. Kamel Boukheddaden qui m'a donné l'envie de me lancer dans cette aventure en premier lieu.

Je remercie mes encadrants qui m'ont accompagné tout au long de cette thèse : Laurent qui m'a fait confiance depuis mon stage de master, qui m'a proposé ce sujet, qui m'a fait bénéficier de sa grande expérience sur la physique des chocs et du CEA DAM, en plus d'être un bon médiateur. Je te souhaite une bonne retraite. Alizée dont j'ai été le premier doctorant, qui m'a apporté sa fougue et son temps. C'est aussi grâce à toi que j'ai eu la chance d'avoir Jean-François comme co-directeur de thèse et de passer quelques semaines à Lausanne. Je remercie mes directeurs : Jean-François pour son enthousiasme et ses encouragements qui m'ont beaucoup motivé, les pistes qu'il m'a donné, les très bonnes semaines que j'ai passé dans son laboratoire et avec son équipe (merci aussi pour nos parties d'échecs mémorables). Laurent de m'avoir fait confiance durant toute la thèse, et de m'avoir offert l'opportunité de chercher et d'encadrer un stagiaire en fin de master HPC durant six mois, Nicolas que je remercie pour son travail de stage avec moi. Une expérience qui aura été très formatrice pour moi. Merci une fois de plus à vous quatre de m'avoir guidé durant ces trois ans et pour vos relectures de ce manuscrit.

Une autre expérience qui aura été très formatrice durant cette thèse et qui n'est pas reflétée dans ce manuscrit, a été d'enseigner en TD les bases de l'architecture d'un ordinateur à des étudiants en deuxième année de licence. Pour cela, je remercie Soraya en charge de ce cours, ainsi que Sébastien et souhaite le meilleur à Gabriel pour ses dernières années de thèse.

Je remercie messieurs les rapporteurs Thibault de Resseguier, et Lionel Lacassagne pour leur relecture, ainsi que pour leurs commentaires et remarques très constructives qui m'ont permis d'améliorer ce manuscrit. Je remercie aussi le président du jury Raymond Namyst qui m'a aussi suivi durant ces trois ans comme membre de mon comité de suivi. Et je remercie les examinateurs Cindy Rountree, Paul Fossati et Laurent Colombet d'avoir accepté d'être membre du jury et pour leurs questions pertinentes lors de la soutenance.

Je remercie aussi les membres du laboratoire dans lequel j'ai évolué durant ces trois ans. Sandra dont personne ne peut se passer au laboratoire et qui nous aide toujours avec gentillesse. Nicolas mon chef d'équipe qui m'a toujours ouvert sa porte si besoin. Paul notamment pour son aide morale sur exaStamp, et son expertise de la mécanique en MD. Thierry que je remercie lui aussi pour son travail sur exaStamp. Jean-Philippe pour l'école d'été à Marseille. Ainsi que tous les autres membres du laboratoire avec qui j'ai pu échanger (souvent autour d'un café) : Claire, Lucas, Frédéric, Bruno, Fabien, Paul, Gregory, Bernard, Agnès, Jean Bernard, Marc, Muriel, Théo et tous les autres.

Je remercie aussi les doctorants et post doctorants du laboratoire : Olivier, Clémentine, Luc, Clément, Yoana. Je remercie en particulier Corentin avec qui j'ai partagé l'encadrement

d'Alizée et surtout pour notre séjour américain. Je vous souhaite à tous de réussir et surtout bon courage aux doctorants pour la fin de thèse.

Je remercie aussi les anciens membres du laboratoire que j'ai rencontrés au CEA mais que je côtoie surtout en dehors autour d'un verre : Robinson, Éléonore, Caroline, et Nacim.

Bien évidemment, je n'oublie pas l'équipage mangeur de raviolis Pauline et Eliott qui ont réalisé cette aventure en même temps que moi, sur lesquels j'ai toujours pu compter pour maintenir le cap (particulièrement lors des derniers jours de rédaction de ce manuscrit) et qui pourront toujours compter sur moi. Notre équipage n'aurait pas été complet sans notre Capitaine, toujours là pour avoir une petite attention, qui nous a supporté lors de nos soirées de plaintes respectives et que je remercie aussi.

Je remercie mes amis de master Alan, Marius, Damien et Guillaume avec qui j'ai découvert le HPC et avec qui j'ai pu traverser la pandémie de Covid.

Je remercie aussi mes amis de toujours Baptiste, Benoit et Henri. Je partage tellement de choses avec vous depuis tellement longtemps, plus de la moitié de ma vie pour certains, et depuis mes débuts à la fac pour d'autres, qu'il est impossible pour moi de lister ici tout ce pourquoi je vous remercie ! Mes débuts à la fac ont aussi impliqué la rencontre de ma binôme de TP favorite, Valentine merci à toi aussi ! Mary je ne t'oublie pas et je te remercie notamment de me prêter encore Henri de temps en temps depuis votre rencontre sur les bancs de l'amphi jusqu'à bientôt votre futur mariage.

Je remercie aussi toute ma famille qui m'a toujours soutenu et bien évidemment en particulier mes parents !

J'ai gardé la meilleure pour la fin, Chloé, cela fait maintenant plus de sept ans que nous grandissons ensemble. Sans toi je n'aurais jamais autant accompli de choses. Ces trois dernières années n'auront pas été faciles et tu m'as supporté tout du long, en particulier lors de ces derniers mois avec la rédaction du manuscrit, où je sais que je n'ai pas été facile. Merci pour tout ce que tu fais. Merci d'être aussi géniale. Merci d'apporter autant de joie à ma vie. Je t'aime !

Je dédie cette thèse à mon chat, Bisous.

Table des matières

1	Introduction générale	7
2	Dynamique Moléculaire pour études sous chocs	18
2.1	Calcul des forces d'interaction	19
2.2	Schéma numérique - Intégration Verlet vitesse	21
2.3	Algorithmes de résolution	23
2.3.1	Liste de Verlet	23
2.3.2	Cellules liées	24
2.4	Code de DM : exaStamp	26
2.4.1	Décomposition de domaine	26
2.4.2	Plateforme de développement : exaNBody	28
2.5	Implémentation d'un potentiel de Tantale	30
2.5.1	Embedded Atom Model (EAM)	31
2.5.2	Fonctions du potentiel de Ravelo et al. [54]	32
2.5.3	Reproduction du tenseur des contraintes	35
2.6	Implémentation d'un potentiel modèle	37
2.6.1	Fonction de paire et paramètres du modèle	37
2.6.2	Reproduction du comportement fragile et ductile	38
2.6.3	Essai du potentiel en traction sur plaque trouée	40
2.7	Système d'adimensionnement	41
3	Développement d'un algorithme de détection et caractérisation automatique de zones d'intérêts de simulation de dynamique moléculaire	45
3.1	Choix de l'algorithme	46
3.2	Projection sur grille régulière	48
3.3	Prototype d'algorithme d'ECC 3D en mémoire distribuée et partagée	54
3.4	Optimisation de notre prototype	66
3.5	Comparaison des performances des deux algorithmes	68
3.6	Conclusion et perspectives	73
4	Étude paramétrique du comportement de notre algorithme	75
4.1	Étude de convergence de la solution pour une sphère parfaite	75
4.2	Amélioration de l'algorithme	78
4.3	Évaluation de l'erreur algorithmique	81
4.3.1	Volume de saturation	83
4.3.2	Loi d'incertitude de notre algorithme	83
4.3.3	Résultats pour la sphère parfaite	85
4.4	Convergence de la solution sur un cas réel	86

5	Applications physiques	91
5.1	Micro-jetting	91
5.2	Splashing	93
5.2.1	Description du phénomène	93
5.2.2	Caractérisation statistique des propriétés des agrégats	94
5.3	Endommagement	96
5.4	Analyse macroscopique	100
5.5	Analyse paramétrique tantale	103
5.5.1	Comparaison entre les mesures macroscopique par vitesse de surface libre et les calculs dans le plan d'écaillage	103
5.5.2	Distributions des pores sur une simulation de 8,3 milliards d'atomes .	107
5.6	Comparaison Tantale et potentiel modèle	110
5.6.1	Observation phénoménologique	110
5.6.2	Analyse des quantités macroscopiques adimensionnées	110
5.7	Perspectives d'application : détection de surfaces	114
6	Conclusion générale	118
	Bibliographie	121
A	On-the-fly clustering for exascale molecular dynamics simulations	130
B	Micro-jetting : Areal density calculation from a triangular groove	156

Chapitre 1

Introduction générale

L'étude du comportement des matériaux en conditions extrêmes, notamment sous des sollicitations ultra dynamiques de type onde de choc, est cruciale pour le développement de nombreux domaines d'activités humaines. Par exemple, dans le secteur de l'énergie, en vue du développement de la fusion nucléaire, de grandes installations expérimentales sont nécessaires, parmi lesquelles on peut citer les installations à confinement inertiel, comme le Laser Mégajoule (LMJ), très grand instrument du CEA DAM, ou le National Ignition Facility (NIF), équivalent américain du LMJ. La compréhension et la maîtrise des phénomènes liés à la physique des chocs sont essentielles à la réalisation des expériences sur ces instruments et à l'exploitation de leurs résultats. Ces phénomènes peuvent notamment générer des débris qui peuvent être néfastes aux installations. Prévenir l'apparition des débris et limiter leur impact sont devenus des enjeux majeurs pour la continuité de l'exploitation des installations. Dans le domaine de l'aérospatial, un des problèmes majeurs à la mise en orbite de satellites ou à l'envoi de véhicules spatiaux est la quantité de débris spatiaux en orbite autour de la terre. Ces débris se déplacent à des vitesses de plusieurs kilomètres par seconde, ainsi, même les plus petits représentent un danger pour les véhicules spatiaux. En cas d'impact, si les satellites ne sont pas suffisamment protégés, ils peuvent être entièrement ou partiellement détruits, et surtout générer de nouveaux débris qui eux même pourront détruire d'autres satellites et ainsi de suite. Cette réaction en chaîne porte le nom de syndrome de Kessler, un ingénieur de la NASA qui a été un des premiers à travailler sur la problématique des déchets spatiaux et qui a évoqué en 1978 ce scénario où la quantité de déchets dépasserait un seuil critique qui rendrait impossible l'utilisation de satellites et l'exploration spatiale pour plusieurs générations [1, 2]. Il est donc important de comprendre les mécanismes d'endommagement et d'éjection de matière qui font suite à la propagation de chocs pour améliorer la protection des objets spatiaux.

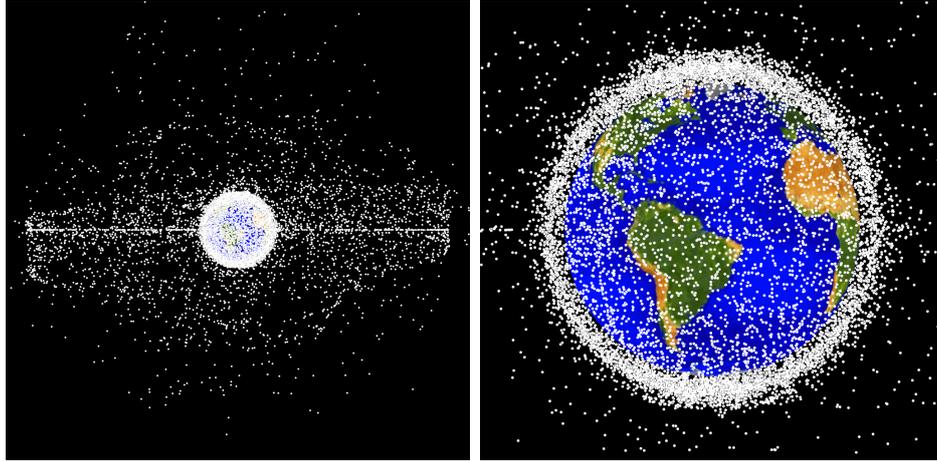


FIGURE 1.1 – En 2019, les débris représentent 95% des objets en orbite autour de la terre, dont 32,7% sont d’origine délibérée. Les objets représentés ici sont ceux d’une taille supérieure à 10 cm, avec à gauche une représentation à environ 35 785 km d’altitude et à droite une altitude de 2 000 km. Crédit : NASA ODPO

Physique des chocs

Une onde de choc est définie par la propagation supersonique d’une discontinuité de pression dans un milieu, et se caractérise par son amplitude et son temps de maintien. Afin d’étudier expérimentalement les phénomènes physiques liés à la propagation d’un choc, trois méthodes principales sont utilisées pour les générer. Elles se différencient par l’amplitude des chocs qu’elles produisent et le taux de déformation qu’elles induisent dans les échantillons. Ces trois méthodes sont :

- La détonation d’explosif, qui génère des chocs non soutenus pouvant avoir des amplitudes d’une centaine de GPa et qui induit des taux de déformation de l’ordre de $10^4 s^{-1}$. Un choc non soutenu étant un choc dont l’amplitude décroît au cours du temps, contrairement au choc soutenu dont l’amplitude reste constante. Cette technique est bien souvent destructive pour l’échantillon.
- L’impact par projectile, qui peut générer des chocs soutenus et non soutenus avec des amplitudes de l’ordre de la dizaine de GPa mais qui peut induire des taux de déformation allant de $10^4 s^{-1}$ à $10^6 s^{-1}$, la durée de sollicitation étant modulée par une variation de l’épaisseur du projectile ou de la cible.
- Le tir laser, qui génère des chocs soutenus ou non. Il est réalisé en focalisant un faisceau laser sur une cible qui monte très vite à haute température et ablate une fine couche de matière qui se transforme en plasma. Ce plasma s’expand très rapidement et génère ainsi par réaction un choc dans la cible. Cette cible peut, soit être directement l’échantillon, soit être une cible intermédiaire qui vient ensuite impacter l’échantillon. Cette deuxième méthode permet de limiter la montée en température de l’échantillon par le laser. Les chocs générés par impulsion laser permettent d’atteindre des niveaux de pression allant de la dizaine de GPa à plusieurs TPa en fonction de l’installation, les plus hauts niveaux de pression sont atteints avec les plus grands instruments tels que le LMJ ou le NIF. Les taux de déformations associés à cette méthode sont aussi d’ordres très élevés allant de $10^7 s^{-1}$ à $10^9 s^{-1}$, fonction de la durée d’impulsion du laser, de l’ordre de la nanoseconde ou de la femtoseconde.

La propagation d'onde de choc peut conduire à une multitude de phénomènes physiques : des phénomènes d'éjection de matière comme le *micro-jetting* ou le *splashing*, ou à des phénomènes internes au matériau qui subit le choc comme un changement de microstructure, un changement de phase ou de l'endommagement. Cette liste est non exhaustive mais présente ceux qui nous intéresseront durant cette thèse.

- Le *micro-jetting*, se produit lorsqu'une onde de choc se réfléchit contre une surface libre qui présente au moins un défaut de planéité. La réflexion d'un choc sur la surface libre conduit à une mise en vitesse très rapide de cette dernière et lorsque celle-ci présente un défaut, l'interaction entre le choc et le défaut engendre une concentration de contrainte à son niveau, ce qui peut produire un jet de matière qui se fragmente ensuite en micro débris. Ces débris sont minuscules mais très rapides et peuvent causer, dans le cas d'expériences, d'importants dégâts sur les équipements présents au sein de l'installation. Cet effet est similaire à l'effet Munroe utilisé pour les charges creuses, mais dans ce cas, il est non désiré car les défauts de surface sont souvent dû aux procédés de fabrication (polissage, usinage, ...). Un schéma du phénomène est présenté dans la figure 1.2.
- Le *splashing*, se produit lors de la collision d'une sphère avec une surface, provoquant l'éjection d'une couronne de matière aux points de contact entre la sphère et la surface. Une image d'un impact hypervélocé réalisé par l'agence spatiale européenne (ESA) pour étudier ce qu'il peut se passer dans l'espace est présentée dans la figure 1.3. Une sphère d'aluminium de 1,2 cm a impacté une plaque de 18 cm d'aluminium à environ 6,8 km/s et a causé un choc d'amplitude supérieur à 365 GPa. L'échantillon a été coupé en deux post-mortem et une sphère similaire à celle ayant impacté la plaque a été placée dans le cratère comme référence. La partie supérieure de l'échantillon illustre le phénomène de *splashing*. L'étude du *splashing* présente aussi un intérêt pour la caractérisation expérimentale du *micro-jetting*. Une des techniques expérimentales utilisée pour caractériser un microjet de matière consiste à placer une cible mince en face du microjet et à mesurer la mise en vitesse de cette cible suite à la collision avec les micro débris du jet. Ce dispositif est une fenêtre d'Asay [3]. Ainsi, comprendre précisément l'interaction entre une sphère et une plaque lors de leur collision permettrait également d'améliorer les diagnostics expérimentaux.
- L'écaillage est un phénomène d'endommagement qui se produit lorsque deux ondes de détentes se croisent et mettent suffisamment en tension le matériau pour provoquer la nucléation de vides qui peuvent ensuite croître et coalescer. Ces événements peuvent même conduire à une rupture du matériau et l'éjection d'une "écaille" ainsi formée. Une des situations pouvant conduire à de l'écaillage est schématisée dans la figure 1.2. Une autre illustration d'écaillage est fournie dans la figure 1.3, dans la partie inférieure de l'échantillon. Cette écaille de matière a été causée par l'impact de la sphère précédemment décrit. L'écaille ne s'est pas détachée de l'échantillon dans cette configuration. Comme pour le *micro-jetting*, ces débris peuvent causer des dégâts qui doivent être maîtrisés lors d'une expérience sur des grandes installations (LMJ, NIF, ...). Ce dernier phénomène sera l'objet d'étude principal de la thèse.

La caractérisation in-vivo de ces phénomènes est primordiale pour proposer et confronter à l'expérience des modèles dynamiques afin de prédire le comportement de systèmes plus complexes par la simulation. Cependant de nombreuses difficultés se posent à l'étude expérimentale in-vivo de ces phénomènes. Tous se produisent à des échelles de temps très courtes, ce qui limite le nombre de mesures possibles durant l'expérience. Pour les expérimentations laser, les échelles de tailles sont aussi très petites de l'ordre du μm ce qui complique les observations

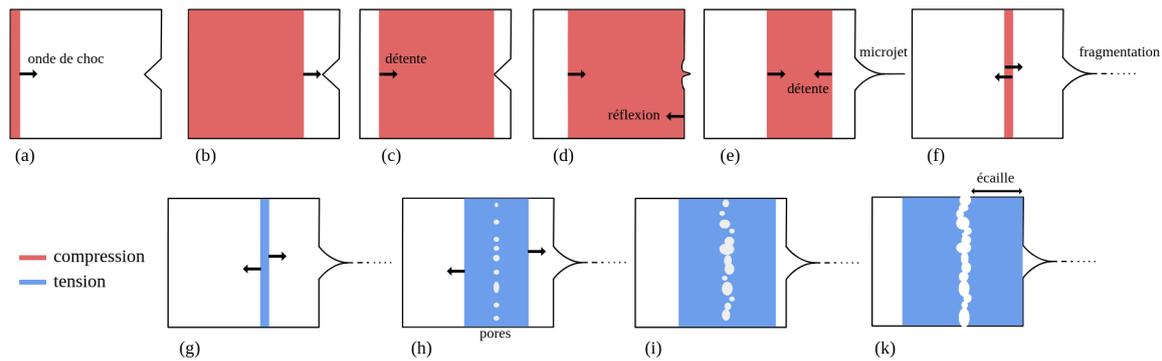


FIGURE 1.2 – Schéma de la propagation d’un choc conduisant à la formation d’un microjet de par la présence d’un défaut triangulaire, et à la formation d’une écaille de par la mise en tension : (a) génération d’une onde de choc, (b) propagation du choc, (c) apparition d’une détente derrière le choc et rencontre du choc avec un défaut de surface, (d) réflexion du choc contre une surface libre et début de formation d’un microjet, (e) propagation de la détente suite à la réflexion de l’onde de choc, (f) fragmentation du microjet, (g) croisement des ondes de détentes qui induit une mise en tension dans l’échantillon, (h) nucléation de pores, (i) croissance et coalescence des pores, et (k) rupture suite à la coalescence et création d’une écaille.

et limite aussi les mesures. De plus, les matériaux qui nous intéressent sont souvent des métaux, matériaux opaques, qui dans le cas de l’endommagement, rendent l’observation des pores encore plus compliquée et nécessite l’utilisation de rayons X. Actuellement, une des seules mesures in-vivo facilement mise en oeuvre pour étudier l’endommagement de métaux est la mesure de la vitesse de la face arrière de l’échantillon à partir de laquelle on déduit les conditions de chargement en posant certaines hypothèses. Les échantillons peuvent être étudiés post-mortem si la récupération est possible soit par tomographie X, soit par microscopie électronique à balayage. À notre connaissance, seulement deux expériences ont réussi à obtenir des mesures in-situ de la fraction volumique de vide [4], et des images des pores au cours du temps [5]. Ces expériences sont très difficiles à réaliser et nécessitent l’utilisation de très grands instruments, elles ont respectivement été réalisées sur le plus grand accélérateur linéaire du monde (3,2 km) du SLAC National Accelerator Laboratory, et sur un des trois plus grands synchrotrons du monde (Advanced Photon Source de l’Argonne National Laboratory). Pour répondre au besoin et au manque d’information in-situ sur ces phénomènes, la dynamique moléculaire classique (DM) s’est alors imposée comme un moyen complémentaire aux expériences, si bien que 50% des publications liées à l’écailage utilisent la DM (Fig. 1.4).

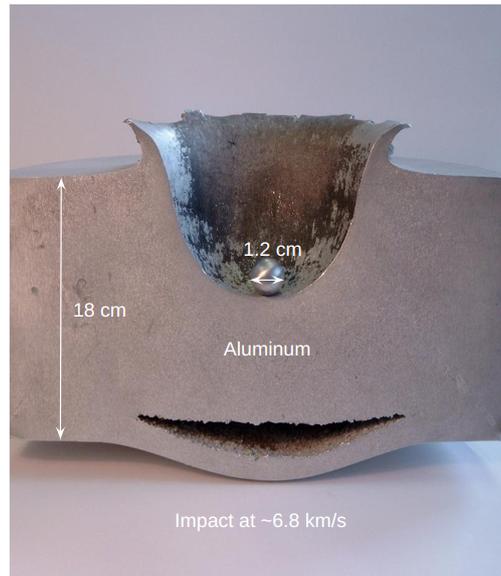


FIGURE 1.3 – Image d'un impact hypervélocité réalisé par l'ESA que nous avons modifié pour faire apparaître les caractéristiques de l'essai. La coupe a été réalisée après l'essai et une sphère d'aluminium similaire à celle ayant impacté la plaque a été placée pour servir de référence à la photo. Crédit : ©ESA

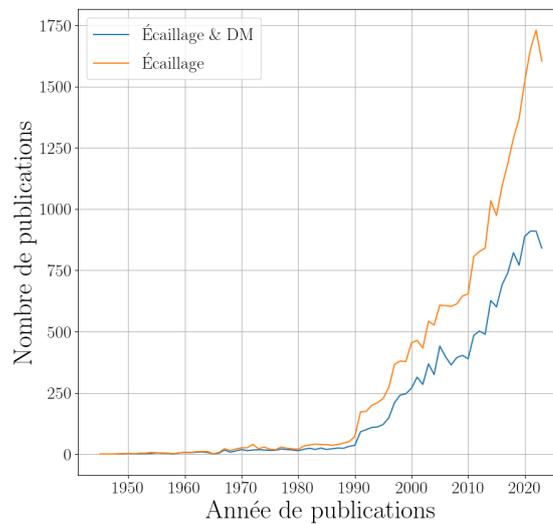


FIGURE 1.4 – Nombre de publications par an ayant pour objet l'écaillage (orange), dont celles utilisant la DM (bleu). Base de données *Web of Science* de l'ISI - *Institute for Scientific Information*

Dynamique moléculaire classique

La DM est une technique de simulation qui modélise la matière à l'échelle de l'atome en simulant leur mouvement uniquement à partir de leurs interactions mutuelles et de l'équation de la dynamique classique. En plus d'avoir une résolution spatiale à l'échelle de l'atome, l'avantage principal de DM est qu'elle peut restituer la physique de phénomènes sans avoir besoin de modèle spécifiques à ceux-ci. Cet avantage l'a rendue populaire dans de très nombreux domaines que nous ne pouvons pas tous citer : en chimie pour la découverte de nouvelle molécule ou pour comprendre l'interaction entre différentes molécules, en biologie pour étudier le repliement de protéines ou les interactions qu'elles peuvent avoir, en physique de la matière condensée pour comprendre la structure de la matière ou comme dans notre cas son comportement sous certaines sollicitations. Cela repose sur une bonne description des interactions entre atomes qui reste encore le point bloquant.

Un autre facteur ayant favorisé le développement de la DM a été le développement des supercalculateurs. En effet, les simulations de DM nécessitent de grandes ressources de calculs, le coût de calcul dépendant du nombre d'atomes simulés et du type de potentiel utilisé. La figure 1.5 présente le nombre de publications par an (de 1946 à 2023) ayant pour objet la DM, le calcul haute performance (HPC), et les publications communes au deux domaines (DM & HPC). Ces données ont été récupérées de la base de données *Web of Science* de l'*ISI - Institute for Scientific Information*. On peut constater un rebond dans le nombre de publications de DM entre 1990 et 1991 lors de l'apparition de publications communes avec le HPC. De plus, ce nombre de publications communes croît plus rapidement que le nombre de publications de DM, et représente maintenant 8,3% des publications de DM. Un autre rebond important à noter, est celui des publications communes de DM et HPC durant la deuxième moitié des années 2000 lors de l'arrivée des premiers processeurs multicoeurs.

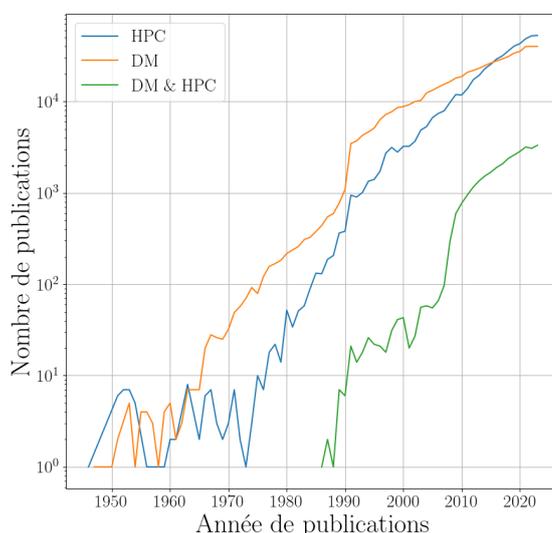


FIGURE 1.5 – Nombre de publications par an ayant pour objet la DM (orange), le HPC (bleu), et celles communes au deux (vert). Axe des ordonnées logarithmique. Base de données *Web of Science* de l'*ISI - Institute for Scientific Information*

Cette augmentation de la puissance de calcul au fil des ans permet de simuler des systèmes de plus en plus grands avec pour objectif de se rapprocher au maximum de l'échelle de la mécanique des milieux continus (supérieure au μm^3). Ces simulations nécessitent plusieurs dizaines de milliards d'atomes. Cependant, cette augmentation du nombre d'atomes entraîne aussi une augmentation de la quantité de données à traiter. Le record actuel du nombre d'atomes simulés est de 20 billions¹ [6] et date de 2018, pourtant à notre connaissance, la plus grosse simulation réalisée en écaillage est de 268 millions d'atomes [7], soit un écart de 5 ordres de grandeur entre les deux. Plusieurs facteurs expliquent cet écart. Premièrement la forme mathématique de l'interaction utilisée dans le cas du record est la plus simple possible et la moins coûteuse (un potentiel Lennard-Jones). Dans un cas d'écaillage de métaux, il n'est pas possible d'utiliser ce type d'interaction pour décrire correctement le système. Le potentiel d'interaction utilisé pour la plus grosse simulation d'écaillage (un potentiel EAM) fait aussi partie des potentiels les moins coûteux mais est déjà dix fois plus coûteux que celui utilisé pour établir le record de simulation. Un autre facteur important est que, pour une étude de physique, avec les mêmes ressources de calculs, on peut préférer faire 10 simulations 10 fois plus petites ou 100 simulations 100 fois plus petites pour faire varier certains paramètres et étudier leur influence. Mais ces facteurs combinés, n'expliquent que deux à trois ordres de grandeurs de différences. Les derniers facteurs d'écarts, s'expliquent certainement par la quantité de données générées par ces simulations. En effet, d'après les auteurs des simulations de plasticité à 268 millions d'atomes, chaque simulation réalisée génère environ 5 exaoctets de données de trajectoire atomique par jour de calcul, une quantité comparable aux capacités de stockage estimées de Google dans le monde² (à l'époque). Les auteurs n'ont bien évidemment pas pu stocker une telle quantité de données. Seul un certain nombre de pas de temps de simulation ont été analysés avec un programme extérieur (OVITO [8]) au code de simulation (LAMMPS [9]). La simulation record à 20 billions d'atomes est restée uniquement en mémoire vive et aucune réelle analyse n'a été réalisée. Son but était uniquement de servir de démonstrateur de performances du code. Stocker uniquement les positions d'un seul pas de temps d'une telle simulation, représenterait déjà 480 To de données. Une sauvegarde conjointe des positions, des vitesses et d'un identifiant des atomes représenterait 1,12 Po de données brutes. Le traitement in-situ s'impose afin d'analyser ces simulations. Si les codes de DM ont suffisamment évolué pour réaliser de telles simulations, les codes d'analyse eux n'ont pas suivi cette optimisation. Le plus utilisé pour les simulations de DM en physique de la matière condensée est OVITO. Il est doté de nombreuses analyses mais présente les défauts de ne pas être parallélisé en mémoire distribuée et de ne pouvoir être utilisé qu'en post-traitement. Le fait de ne pas être parallélisé en mémoire distribuée le limite à ne pouvoir traiter que des systèmes qui tiennent dans la mémoire vive d'un seul noeud de calculs, souvent de l'ordre de 256 Go, et jusqu'à quelques téraoctets dans des cas exceptionnels de noeuds dédiés. Cela limite donc ses capacités de traitement à des systèmes de l'ordre du millions, à quelques centaines de millions d'atomes au maximum selon les analyses utilisées. Par ailleurs, sa nature de code de post-traitement, oblige à écrire sur disque les données de simulations que l'on veut analyser, ce qui prend du temps, des ressources, et qui limite la fréquence d'analyse. L'écart entre les performances des codes d'analyses et des codes de DM apparaît donc comme un important facteur limitant à l'exploitation de simulation de DM grande échelle pour la physique. Le développement de codes d'analyses in-situ, aussi bien

1. billion : $10^{12} \Leftrightarrow$ trillion sur l'échelle anglo saxonne.

2. "a simulation of the kind reported here generates about 5 exabytes (1 exabyte = 10^{18} bytes) of atomic trajectory data in just one day on the Sequoia supercomputer, an amount comparable to the estimated world wide storage capacity of Google."

adapté que les codes de DM, aux dernières architectures de supercalculateur devient essentiel pour répondre au besoin de réaliser des simulations DM grande échelle, s'apparentant à de véritables expériences numériques et générant de gigantesques quantités de données.

Calcul haute performance

Le HPC est une discipline de l'informatique qui vise à obtenir le maximum de performance des ordinateurs et en particulier des supercalculateurs, essentiellement pour le calcul scientifique. Pour tirer un maximum de performance des supercalculateurs, le HPC se focalise autant sur le point de vue matériel de l'ordinateur (la science de l'architecturer physiquement) que sur le point de vue logiciel (la science d'adapter les programmes à l'ordinateur). Au début de l'informatique, les ordinateurs étaient uniquement dédiés au calcul scientifique, toute la science de l'informatique pouvait donc déjà s'apparenter à du HPC mais on distingue réellement le début du HPC à partir des années 1950-1960 avec les premiers supercalculateurs. Depuis leur apparition, la puissance des supercalculateurs n'a cessé de croître de façon exponentielle. Leur architecture s'adaptant pour dépasser les palliers de performances régulièrement atteints. Ces évolutions d'architecture ont été très rapides avec notamment :

- Des ordinateurs superscalaires, c'est à dire capable de réaliser plusieurs opérations en 1 cycle d'horloge.
- Des ordinateurs multi-coeurs, c'est à dire ayant plusieurs coeurs de calcul partageant une mémoire centrale (dont les coeurs pouvaient être superscalaires).
- Des ordinateurs vectoriels, c'est à dire que leurs instructions sont capables d'agir sur plusieurs espaces mémoire à la fois.
- Des ordinateurs distribués ou multiprocesseurs, c'est à dire, qu'ils sont connectés par réseau et possèdent chacun leur mémoire centrale.

Différentes combinaisons de ces architectures ont dominé le marché selon les périodes avec néanmoins une généralisation des supercalculateurs multiprocesseurs à partir de la fin des années 1980. En 1993, le projet du top500 a été créé, il consiste à classer et répertorier deux fois par an les 500 supercalculateurs les plus puissants du monde avec le benchmark linpack. Ce benchmark consiste à effectuer un maximum d'opérations d'algèbre linéaire et à mesurer le nombre d'opérations flottantes par secondes (FLOPS ou Flop/s) exécutées par le calculateur. Ce projet permet ainsi de voir quelles architectures dominent ainsi que l'évolution au global des supercalculateurs dans le monde. La figure 1.6 est extraite du site du top500, elle présente en jaune les performances en fonction du temps du calculateur classé premier, en bleu les performance du 500e calculateur et en vert la puissance cumulée des 500 calculateurs dans le classement. De l'année 1993, début du top500, à 2007, les processeurs monocoeurs superscalaires dominaient le classement mais ont très vite été remplacés par des processeurs multicoeurs superscalaires jusqu'au début des années 2010 où sont apparus les processeurs multicoeurs superscalaires avec jeux d'instructions vectoriels et où le nombre de machines dotées d'accélérateurs matériels types GPU n'a cessé d'augmenter jusqu'à nos jours. Cette architecture pourrait bientôt devenir majoritaire avec actuellement 194 systèmes dont 9 dans le top 10 (Juin 2024).

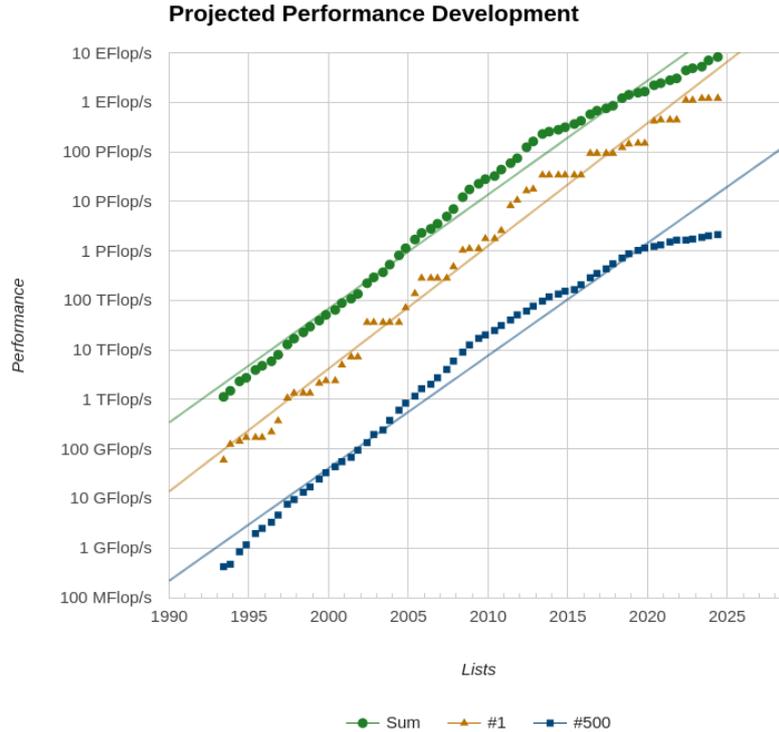


FIGURE 1.6 – Figure du top500 qui présente les performances en Flop/s par an du supercalculateur le plus puissant (orange), du 500e calculateur le plus puissant (bleu) et de la puissance cumulé des 500 calculateurs les plus puissant du monde (vert).

Cette augmentation croissante et très rapide de la complexité des architectures nécessite beaucoup d’effort d’un point de vue logiciel pour maintenir les codes à niveau. Il faut d’une part tirer un maximum de performances des machines mais aussi d’autre part maintenir une pérennité des codes. Il est arrivé que certaines architectures fassent figure d’étoile filante en disparaissant très rapidement après avoir nécessité un portage des codes. C’est dans l’optique de répondre à ces deux besoins de performances et de pérennité qu’a été développé au CEA le code de DM *exaStamp* à partir de 2012 [10] au moment de cette transition dans les architectures de supercalculateur. Simultanément, le développement de bibliothèques de portabilité de performances telles que Kokkos [11] et leur intégration dans les codes de DM (ex LAMMPS) ont eu lieu.

Dans cette recherche constante de la performance en HPC et en DM, on peut noter le projet du groupe D. E. Shaw Research, qui en plus de développer son propre code de DM, développe entièrement ses propres supercalculateurs spécifiquement pour ses calculs de DM en biochimie à partir de FPGA³. La dernière génération de leur calculateur (*Anton 3*) a ainsi le record du temps de simulation de DM possible en 1 jour de calcul, 100 μs avec 1 millions d’atomes, et des performances générales de deux ordres de grandeur plus élevés que n’importe quel autre supercalculateur sur leur type de simulation [12].

3. field-programmable gate array

Objectifs et organisation du manuscrit

Le principal objectif de cette thèse sera de développer un code d'analyse in-situ de DM adapté à des simulations d'au moins plusieurs milliards d'atomes, dans le but d'étudier des phénomènes ultra dynamiques liés à la propagation de chocs dans les métaux. Ce code d'analyse devra s'intégrer dans le code de DM HPC du CEA : exaStamp, avec pour condition de s'exécuter en un temps raisonnable comparé à la DM. Les principales capacités d'analyses qu'il devra fournir seront d'identifier et de caractériser pour les simulations d'endommagements les pores qui se forment, et pour les simulations d'éjections les agrégats de matières générés. Le deuxième objectif principal sera d'appliquer l'analyse développée à des simulations de cas typiquement étudiés au sein du laboratoire pour démontrer ses capacités et ouvrir de nouvelles perspectives d'études aux physiciens.

En dehors de ce premier chapitre d'introduction, ce manuscrit s'organise en 5 chapitres :

- Le chapitre 2 présente d'abord les aspects essentiels au fonctionnement de la DM, les familles de potentiels d'interactions, le schéma numérique utilisé et les principaux algorithmes de résolution utilisés en DM. Puis, il présente le code de DM que nous utilisons exaStamp et ses spécificités. Ce chapitre se finit sur une partie de mes contributions :
 - L'implémentation de deux potentiels, réalisée pour les simulations d'endommagement : un potentiel de Tantale et un potentiel modèle.
 - Une proposition de système d'adimensionnement des résultats pour une comparaison de matériaux cristallins n'ayant pas le même formalisme de potentiel en DM.
- Le chapitre 3 présente le choix de l'algorithme d'analyse que j'ai effectué pour accomplir les objectifs : l'étiquetage par composante connexe (ECC). Puis il présente la première version de l'algorithme que j'ai développé, parallélisé en mémoire partagée et en mémoire distribuée, suivi de sa deuxième version, et de la comparaison de leurs performances avec la DM.
- Le chapitre 4 présente l'étude paramétrique du comportement de l'algorithme développé au chapitre 3, afin d'estimer l'incertitude de mesure associée. Cette étude nous permet notamment de proposer une amélioration de l'algorithme d'analyse.
- Le chapitre 5 présente les simulations sur lesquelles l'algorithme a été testé : les simulations d'éjections de matière (splashing, et micro-jetting), puis les simulations d'écaillage réalisées avec les potentiels implémentés au chapitre 2.
- Le chapitre 6 présente notre conclusion générale ainsi que les perspectives ouvertes par cette thèse.

Chapitre 2

Dynamique Moléculaire pour études sous chocs

La dynamique moléculaire est une technique de modélisation de la matière qui consiste à simuler le mouvement des atomes au cours du temps à partir du principe fondamental de la dynamique (2ème loi de Newton) $\sum \mathbf{F} = m\mathbf{a}$. À partir de la position des atomes, la somme des forces \mathbf{F} s'exerçant sur ces derniers est calculée, en connaissant leur masse m , on peut déduire leur accélération \mathbf{a} qu'on intègre pour obtenir leur vitesse, qui permet de déterminer leur position au pas de temps suivant. L'intérêt de la dynamique moléculaire est de pouvoir étudier le comportement de la matière uniquement à partir du calcul des forces interatomiques. Dans un cas comme le nôtre cela présente l'énorme avantage de pouvoir étudier le phénomène d'écaillage sans avoir besoin d'un modèle d'endommagement. En mettant le système dans les conditions de chargement recherchées, le phénomène sera naturellement restitué. Le problème principal de la dynamique moléculaire est alors de pouvoir calculer les forces interatomiques correctement. Selon les principes utilisés pour le calcul des forces, on distingue la dynamique moléculaire quantique ou *ab initio* (AIMD), de la dynamique moléculaire classique (DM).

L'AIMD repose sur la résolution des équations de la physique quantique pour calculer les forces, une des techniques de résolution les plus communes est la DFT¹ qui repose sur les travaux de Hohenberg et Kohn qui stipulent que l'on peut décrire un système de façon exact entièrement à partir de sa densité électronique [13]. Le problème devient alors d'arriver à calculer cette densité électronique, le plus souvent à partir de l'équation de Kohn-Sham [14], une reformulation de l'équation Schrödinger. Pour résoudre cette équation, différentes approximations sont faites selon le système étudié et permettent d'étudier des systèmes de quelque atomes à quelques milliers d'atomes en fonction du coût de calcul des différentes approximations et, du nombre d'électrons à prendre en compte.

Pour pouvoir simuler des systèmes de plus grande taille et donc étudier des phénomènes se produisant à plus grande échelle, il faut diminuer le coût de calcul des forces de plusieurs ordres de grandeur. Situation possible en s'affranchissant du calcul des effets quantiques et donc en passant à des calculs de DM. Cette approximation est majeure. Cependant, elle se justifie par plusieurs points. Premièrement, dans la plupart des cas, le calcul des forces interatomiques repose tout de même sur une base de calculs *ab initio* sur lesquels les potentiels d'interactions sont ajustés. Ce qui réduit l'approximation faite en s'affranchissant des calculs quantiques parce que l'on considère qu'ils sont pris en compte dans l'ajustement du potentiel. Deuxièmement, dans un cadre comme le nôtre, où nous voulons étudier le comportement de

1. Density Functional Theory

la matière suite à la propagation d'un choc, sa réflexion et l'endommagement qui peut en découler, le résultat final sera probablement plus sensible à la taille du système qu'au choix du potentiel interatomique. Troisièmement, le coût de calcul d'une AIMD est prohibitif pour une simulation d'écaillage, plusieurs milliers d'années seraient nécessaires à sa réalisation en utilisant les calculateurs les plus puissants. Nous n'avons donc pas d'autre choix que d'utiliser la DM classique.

Ce chapitre présentera d'abord plus en détails la DM avec les différentes catégories de potentiels (Sec. 2.1), le schéma numérique utilisé pour résoudre l'équation du mouvement (Sec. 2.2), et les algorithmes numériques utilisés pour appliquer le schéma (Sec 2.3). Puis nous présenterons le code de calcul exaStamp (Sec. 2.4), avant de décrire les potentiels des matériaux que nous avons implémentés et décidé d'étudier durant cette thèse : le Tantale (Sec 2.5) et un matériau modèle (Sec. 2.6). Nous finirons par présenter le système d'adimensionnement que nous avons utilisé pour comparer les résultats entre les différents potentiels (Sec. 2.7).

2.1 Calcul des forces d'interaction

Comme nous l'avons dit précédemment, le calcul des forces en DM est crucial car c'est sur lui que va reposer toute la simulation et sa physique. Le calcul des forces d'interactions se fait par la dérivation des énergies potentielles d'interactions, souvent abrégées potentiels ou appelées champs de forces selon les communautés. On distingue trois grandes familles de potentiels en DM :

- Les potentiels empiriques ou semi-empiriques
Historiquement, ces potentiels sont les premiers à avoir été développés. Ils sont décrits par des fonctions de formes arbitraires fondées sur des considérations physiques. Les paramètres de ces fonctions sont ajustés pour reproduire des propriétés du matériau étudié. Ces propriétés peuvent être diverses comme le paramètre de maille, les constantes élastiques, ou les énergies de défauts d'empilements. Ces propriétés peuvent soit avoir été mesurées expérimentalement, soit avoir été calculées en *ab initio*. Reproduire parfaitement toutes les propriétés d'un matériau reste néanmoins impossible et implique souvent de devoir faire des compromis en fonction de ce que l'on cherche à étudier. On peut tout de même noter que ces potentiels sont assez robustes pour des études qualitatives, en extrapolation de leur domaine d'ajustement.
- Les potentiels numériques
Cette classe de potentiel est la dernière à être apparue et désigne les potentiels qui utilisent le *machine learning*. Ces potentiels ne reposent pas sur un formalisme physique. Ils prennent en entrée ce que l'on appelle des "descripteurs" de l'environnement atomique à partir desquels ils prédisent les forces interatomiques. Ces potentiels sont entraînés sur des bases de données *ab initio* et essaient de reproduire les forces issues des configurations de la base de données. Ces potentiels peuvent souvent être plus précis que les potentiels empiriques mais sont souvent catastrophiques en extrapolation hors de leur domaine d'entraînement.
- Les potentiels modèles
Ces potentiels peuvent être rattachés aux potentiels semi-empiriques car ils reposent sur les mêmes formalismes. Leur différence vient du fait que leurs paramètres ne sont pas ajustés pour reproduire un matériau en particulier mais sont utilisés pour des études purement phénoménologiques.

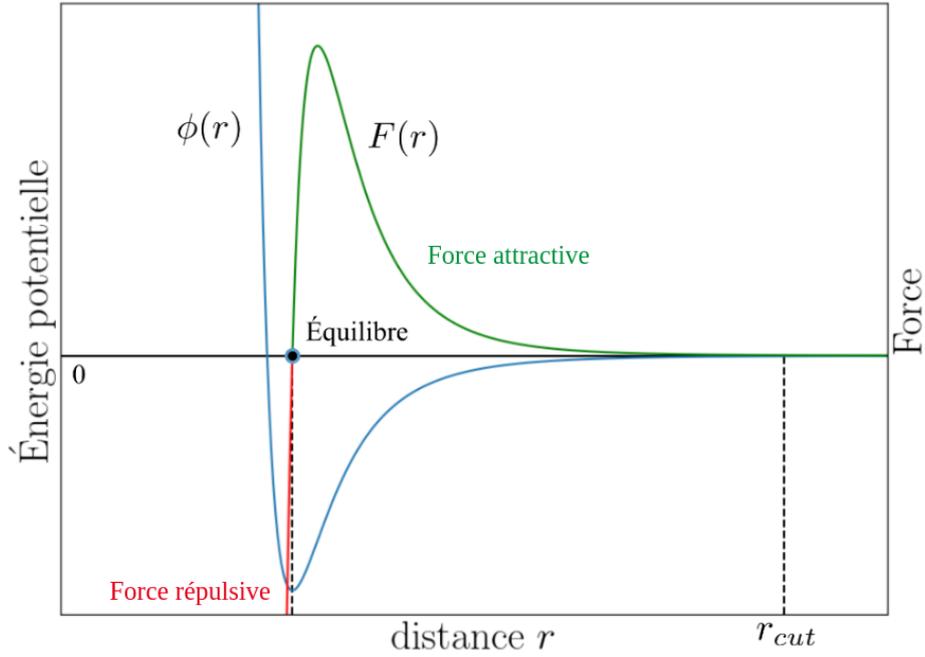


FIGURE 2.1 – Exemple de potentiel de paires et de la force qui en découle.

Nous n'avons pas évoqué le coût de calcul de ces différentes familles car il est trop dépendant du formalisme sous-jacent employé. Parmi les nombreux formalismes qui existent nous pouvons distinguer les potentiels de paires qui comme leur nom l'indique dépendent uniquement d'interaction de paires. L'énergie d'un atome s'exprime alors uniquement comme la somme des énergies de paires. Les potentiels de paires sont les potentiels les plus simples et les moins coûteux, ils ont été les premiers à avoir été développés et sont souvent la fondation de potentiels plus complexes. Le potentiel de pair le plus emblématique est certainement le Lennard-Jones [15, 16] qui décrit l'interaction entre deux atomes de type gaz rare mais qui est aussi utilisé en chimie comme bloc de base pour construire des interactions moléculaires. La figure 2.1 présente la forme de ce potentiel d'interaction $\phi_{ij}(r)$ avec r la distance séparant les deux atomes i et j , et la force $F_{ij}(r)$ qui en découle.

L'autre grande classe de formalisme que l'on peut distinguer sont les potentiels à plusieurs corps ou à N-corps, qui en plus des interactions de pair, vont prendre en compte des effets liés à l'interaction d'au moins trois corps entre eux. Ces effets peuvent être pris en compte de bien des façons pour pouvoir toutes les présenter, nous choisissons d'en présenter deux particulièrement bien adaptés pour les métaux. Le formalisme EAM² [17] est un des premiers à être apparu et prend en compte en plus des interactions de paires un terme d'*embedding* qui modélise la densité électronique du système, elle-même calculée à partir de la somme de fonction dépendant uniquement de la distance interatomique r . Nous implémenterons un potentiel qui utilise ce formalisme dans la section 2.5 et reviendrons plus en détails sur celui-ci à ce moment. L'autre formalisme que nous choisissons d'introduire est le formalisme MEAM³ [18, 19] qui est fondé sur le formalisme EAM mais qui introduit en plus des contributions angulaires aux interactions de paires et ajoute un terme d'écrantage à ces mêmes interactions. Utiliser des termes angulaires est une technique commune des potentiels à N-corps et introduit

2. Embedded Atom Method
3. Modified Embedded Atom Model

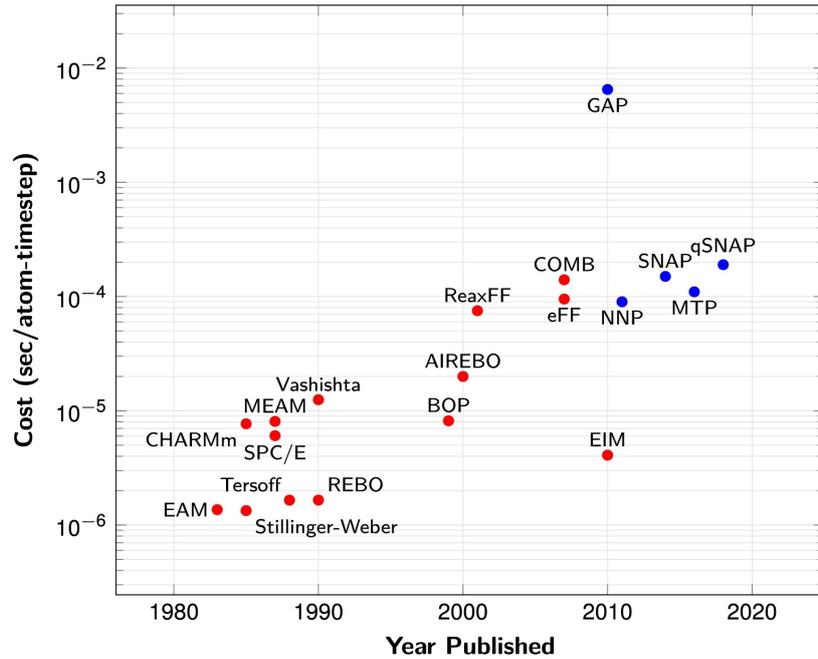


FIGURE 2.2 – Figure extraite de [9], qui exprime le coût de calcul de différents formalismes de potentiels dans LAMMPS en fonction de l’année de la publication du formalisme. En rouge sont représentés des potentiels à N-corps semi-empiriques et en bleu sont représentés des potentiels numériques.

plus de physique aux potentiels mais le calcul de ce type de termes est assez coûteux, la figure 2.2, tirée de [9], présente en rouge le coût de calcul dans LAMMPS⁴ de différents potentiels à N-corps en fonction de leur année de publication. On peut remarquer que les potentiels EAM sont parmi les potentiels les moins chers en particulier comparés aux potentiels MEAM qui sont d’un ordre de grandeur plus cher. Les potentiels en bleu sont des potentiels numériques, leur coût est deux ordres de grandeur plus élevés que les EAM.

Les simulations d’endommagement étant souvent dans des domaines d’extrapolations des potentiels et plutôt limitées par la taille des systèmes étudiés, dans cette thèse, nous nous concentrerons uniquement sur des potentiels semi-empiriques ou modèles, les potentiels numériques étant pour l’instant inadaptés à ce type de simulation. La forme exacte des potentiels que nous utiliserons et leurs propriétés seront présentés dans les sections 2.5 et 2.6. La section suivante introduit le schéma numérique utilisé en DM pour résoudre l’équation du mouvement des atomes à partir du calcul des forces.

2.2 Schéma numérique - Intégration Verlet vitesse

Comme évoqué précédemment, l’équation résolue en DM au cours du temps est celle du principe fondamental de la dynamique à partir du calcul des forces atomiques qui dérivent d’énergies potentielles d’interactions.

$$\mathbf{F}(\mathbf{x}, t) = m\mathbf{a}(t) \quad (2.1)$$

4. Large-scale Atomic/Molecular Massively Parallel Simulator : code DM open source qui fait référence dans la communauté.

à partir de conditions initiales \mathbf{x}_0 et \mathbf{v}_0 , les positions et vitesses initiales des atomes, on peut calculer les positions et les vitesses à un instant suivant $t + \Delta t$, avec Δt le pas de temps, en utilisant le schéma d'Euler :

$$\begin{cases} \mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \mathbf{v}(t)\Delta t + \mathcal{O}(\Delta t^2) \\ \mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \mathbf{a}(t)\Delta t + \mathcal{O}(\Delta t^2) \end{cases} \quad (2.2)$$

Ces développements de Taylor d'ordre 2 vont cependant avoir tendance à dériver numériquement quand le nombre de pas de temps devient grand et donc ne conservent plus l'énergie du système. On peut cependant développer la série de Taylor des positions à l'ordre 4 et obtenir les équations :

$$\begin{cases} \mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \mathbf{v}(t)\Delta t + \frac{\mathbf{a}(t)\Delta t^2}{2} + \frac{\mathbf{j}(t)\Delta t^3}{6} + \mathcal{O}(\Delta t^4) \\ \mathbf{x}(t - \Delta t) = \mathbf{x}(t) - \mathbf{v}(t)\Delta t + \frac{\mathbf{a}(t)\Delta t^2}{2} - \frac{\mathbf{j}(t)\Delta t^3}{6} + \mathcal{O}(\Delta t^4) \end{cases} \quad (2.3)$$

avec \mathbf{j} l'à-coup, dérivée de l'accélération par rapport au temps. En sommant ces deux équations, on obtient ainsi :

$$\mathbf{x}(t + \Delta t) = 2\mathbf{x}(t) - \mathbf{x}(t - \Delta t) + \mathbf{a}(t)\Delta t^2 + \mathcal{O}(\Delta t^4) \quad (2.4)$$

C'est le schéma de Störmer-Verlet [20-22], qui a l'avantage d'être stable numériquement et de conserver l'énergie pour un petit pas de temps. Il faut cependant stocker en mémoire les positions de deux pas de temps et il ne calcule pas explicitement les vitesses des atomes. Or la vitesse est utile pour calculer des grandeurs physiques comme l'énergie cinétique, la température ou la pression du système. Il faut alors calculer la vitesse explicitement avec l'équation :

$$\mathbf{v}(t) = \frac{\mathbf{x}(t + \Delta t) - \mathbf{x}(t - \Delta t)}{2\Delta t} + \mathcal{O}(\Delta t^2) \quad (2.5)$$

La vitesse se retrouve alors avec un pas de temps de décalage par rapport à la position. Pour éviter ce problème et stocker un vecteur de moins en mémoire lorsque l'on veut le calcul des positions, alors on peut utiliser le schéma dit de "Verlet vitesse" [23] qui calcule explicitement la vitesse et la position à chaque incrément de temps Δt avec les équations :

$$\begin{cases} \mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \mathbf{v}(t)\Delta t + \frac{\mathbf{a}(t)\Delta t^2}{2} + \mathcal{O}(\Delta t^3) \\ \mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \frac{[\mathbf{a}(t + \Delta t) + \mathbf{a}(t)]\Delta t}{2} + \mathcal{O}(\Delta t^3) \end{cases} \quad (2.6)$$

Et en passant par le calcul intermédiaire d'un demi pas de temps $\Delta t/2$ pour les vitesses on peut réécrire le système d'équations suivant :

$$\mathbf{v}(t + \Delta t/2) = \mathbf{v}(t) + \frac{\mathbf{a}(t)\Delta t}{2} \quad (2.7)$$

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \mathbf{v}(t + \Delta t/2)\Delta t \quad (2.8)$$

$$\mathbf{a}(t + \Delta t) = \frac{\mathbf{F}(\mathbf{x}(t + \Delta t))}{m} \quad (2.9)$$

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t + \Delta t/2) + \frac{\mathbf{a}(t + \Delta t)\Delta t}{2} \quad (2.10)$$

qui se résout en mettant à jour l'accélération entre les équations (2.8) et (2.10) par le calcul des forces \mathbf{F} qui ne dépendent que de la position atomique. L'avantage de ce schéma est de

ne stocker que deux vecteurs pour avoir accès à la position et la vitesse des atomes à chaque pas de temps en plus de fournir une vitesse synchronisée avec la position, tout en conservant l'énergie si le pas de temps est suffisamment petit. C'est ce schéma que nous utilisons dans notre code de DM et le reste de cette thèse. La section suivante présente les algorithmes utilisés pour mettre en œuvre efficacement ce schéma.

2.3 Algorithmes de résolution

Intéressons nous maintenant aux méthodes permettant d'implémenter le schéma de "Verlet vitesse" présenté dans la section précédente (Sec. 2.2). La majorité du temps de calcul du schéma de "Verlet vitesse", et de façon générale, tous les schémas d'intégration en DM, est concentrée dans le calcul des forces d'interactions car si ce calcul est résolu de façon naïve, chaque atome doit calculer au moins une fois la valeur de son interaction avec tous les autres quand un terme de paires est impliqué. Si on considère un nombre d'atomes N alors cela nous donne une complexité algorithmique minimale de $\mathcal{O}(N^2)$. La première optimisation à avoir été introduite repose sur un constat physique simple. La majorité des termes d'interactions de paires sont des termes d'interactions courte portée, c'est-à-dire qu'à mesure que la distance interatomique r croît, la valeur de l'énergie potentielle tend vers zéro. Si l'énergie décroît alors au moins en r^3 , on peut introduire une distance à partir de laquelle on peut négliger le terme d'interaction. Cette distance est communément appelé rayon de coupure et se note r_{cut} . Informatiquement, en testant si la distance séparant deux atomes est supérieure au rayon de coupure, alors on peut s'économiser le calcul du terme d'interaction. Cependant, la complexité algorithmique de ce calcul reste en $\mathcal{O}(N^2)$ car l'on doit tout de même calculer la distance de chaque atome avec tous les autres. À partir de l'introduction de ce rayon de coupure, deux optimisations majeures ont par la suite été développées. L'une pour réduire la fréquence à laquelle la distance interatomique est calculée et l'autre pour diminuer la complexité algorithmique de ce calcul : la liste de Verlet et la méthode des cellules liées.

2.3.1 Liste de Verlet

La méthode des listes de Verlet a été introduite par Loup Verlet dans le même article qui présente le schéma numérique de Störmer-Verlet [20]. Elle consiste à réduire la fréquence à laquelle on calcule les distances interatomiques r en utilisant des listes de voisins. Le principe, illustré par la figure 2.3, est d'ajouter au rayon de coupure r_{cut} un incrément Δr pour obtenir un rayon de Verlet, noté r_{verlet} pour créer une liste de voisins associés à chaque atome, qui contient tous les atomes inclus dans ce rayon. Ces listes de voisins sont aussi nommées listes de Verlet. Cette méthode assure que, tant qu'aucun atome du système ne s'est déplacé de plus de Δr , entre un instant t_1 et t_2 , alors la liste des atomes interagissant avec un atome donné, est contenue dans sa liste de Verlet construite à l'instant t_1 et donc ne nécessite pas de recalculer les distances entre tous les atomes mais seulement avec ceux contenus dans sa liste de Verlet. Si à l'instant t_2 , un atome s'est déplacé de plus de Δr par rapport à l'instant t_1 où les listes ont été construites, alors on calcule les distances entre tous les atomes pour résoudre le calcul des forces et l'on en profite pour reconstruire les listes de Verlet, avec t_2 qui devient alors le nouvel instant t_1 . La complexité du calcul des forces d'interactions devient donc linéaire ($\mathcal{O}(N)$) pour les instants où la liste de voisin n'est pas reconstruite car la taille des listes de Verlet est bien inférieure au nombre d'atomes total du système. Mais pour les instants où les listes nécessitent d'être reconstruites, la complexité de leur construction reste

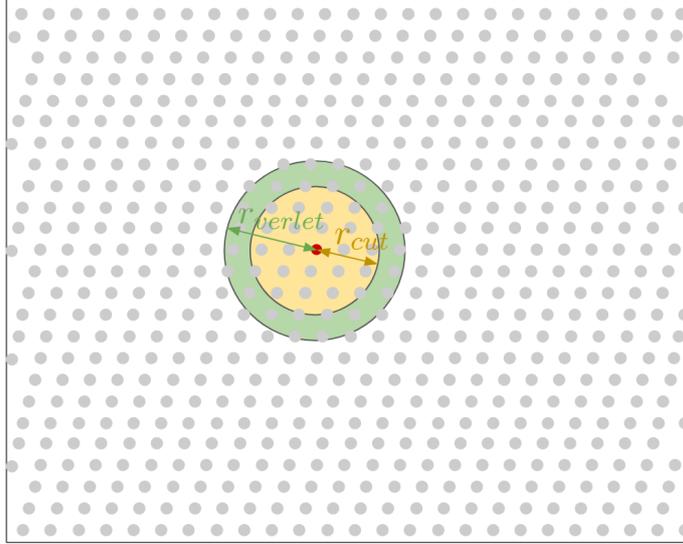


FIGURE 2.3 – Illustration du rayon de Verlet qui sert à déterminer la liste de voisins d'un atome.

en $\mathcal{O}(N^2)$, la complexité globale de l'algorithme reste donc quadratique du nombre d'atomes. La prochaine optimisation présentera un moyen de diminuer définitivement la complexité à une complexité linéaire (Subsec. 2.3.2).

On peut noter que dans cette version, cette technique nécessite de sauvegarder la position de chaque atome à l'instant où les listes sont construites (t_1) pour pouvoir calculer son déplacement à un instant t_2 . Cela peut être gourmand en mémoire dans certains cas. Pour contourner ce problème, une variation de cette méthode est de calculer à chaque pas de temps le déplacement maximal d'un atome et d'incrémenter un compteur C de ce déplacement. Il suffit alors de comparer la valeur de ce compteur à Δr , s'il est strictement inférieur alors les listes n'ont pas besoin d'être reconstruites, sinon on réinitialise le compteur à zéro et l'on reconstruit les listes. Ainsi, les listes sont rafraîchies plus souvent mais l'empreinte mémoire est diminuée (1 valeur vs N valeurs).

En pratique lorsque les atomes oscillent autour de leur position d'équilibre, les listes de Verlet ne seront pas fréquemment rafraîchies, voire pas du tout. Alors que si le système est hautement dynamique, comme dans notre cas d'étude, les listes de Verlet risquent d'être souvent mises à jour, voire à chaque pas de temps de la simulation. Il faut donc augmenter le rayon de Verlet pour diminuer la fréquence de rafraîchissement. Cependant, augmenter ce rayon implique une augmentation de la taille des listes, et donc l'empreinte mémoire de la méthode, mais aussi du temps de calcul des interactions car chaque atome doit boucler sur des listes plus longues pour trouver les atomes avec lesquels il interagit. Il faut donc trouver un compromis empirique pour la valeur du rayon de Verlet qui minimise le temps de calcul pour chaque cas et chaque potentiel.

2.3.2 Cellules liées

La méthode des cellules liées [24] a deux intérêts principaux. Le premier est de diminuer la complexité du calcul des forces d'interactions de $\mathcal{O}(N^2)$ à $\mathcal{O}(N)$. Le second est de permettre un maillage de l'espace de simulation qui facilite le parallélisme. Cette méthode peut s'utiliser

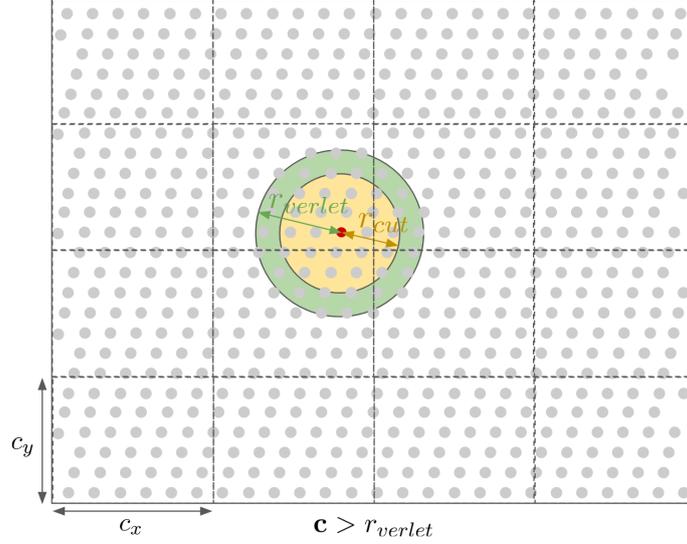


FIGURE 2.4 – Illustration des cellules liées en plus du rayon de Verlet, pour diminuer la complexité algorithmique de la DM à $\mathcal{O}(N)$, N étant le nombre d’atomes.

en conjonction de la technique des listes de Verlet et consiste à découper le domaine de la simulation en cellules, elle est illustrée dans la figure 2.4. Dans notre cas, les cellules seront des pavés droits de volume égaux et de dimension \mathbf{c} tel que :

$$\mathbf{c} = \begin{pmatrix} c_x \\ c_y \\ c_z \end{pmatrix} = \mathbf{x}_{form} \cdot cell_{size} \quad (2.11)$$

avec $\mathbf{x}_{form} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$, un vecteur contenant les coefficients multiplicatifs à appliquer au scalaire $cell_{size}$ dans chaque direction de l’espace pour avoir des conditions limites périodiques. S’il n’y a pas de conditions limites périodiques à appliquer, alors le système peut être maillé avec des cubes d’arêtes de dimension $cell_{size}$. Dans notre cas, le vecteur \mathbf{x}_{form} sera automatiquement calculé par le code à partir de la dimension du système étudié et la valeur de $cell_{size}$ choisie par l’utilisateur. Le pré-requis à la méthode des cellules liées est que chaque composante de \mathbf{c} doit-être supérieure à r_{verlet} dans le cas où elle est couplée à la technique des listes de Verlet et supérieur au r_{cut} dans le cas où elle est utilisée seule. En remplissant ce pré-requis, on peut stocker chaque atome dans la cellule à laquelle il appartient sous forme de liste et ainsi lorsqu’un atome doit construire sa liste de voisins dans le cas du couplage avec la technique des listes de Verlet ou simplement calculer les forces qui s’exercent sur lui quand la méthode est utilisée seule, l’atome n’a qu’à itérer sur les atomes de sa cellule et de ses voisins directes soit 27 cellules au total en 3D. Si l’on considère qu’elles contiennent n atomes au total et que le système en contient N alors la complexité dans les deux cas est de $\mathcal{O}(n \cdot N)$. La dimension de ce système de 27 cellules étant de $3\mathbf{c}$ et étant petite devant la dimension totale du système, alors $n \ll N$ et donc la complexité totale devient $\mathcal{O}(N)$.

Dans notre cas, la méthode des cellules liées sera toujours couplée à la technique des listes de Verlet, nous adopterons par la suite du manuscrit le terme “cellules de Verlet” pour désigner les cellules obtenues par méthode des cellules liées en conjonction de la technique des listes de Verlet car leur dimension est plus ou moins fixée par la taille du rayon de Verlet

et nous permettra de les différencier des autres cellules que nous utiliserons dans le cadre de cette thèse.

Nous venons de présenter le premier intérêt à l'utilisation de la méthode des cellules liées pour les simulations de DM ayant des potentiels d'interactions courte portée. Nous présenterons le second intérêt, la facilitation du parallélisme, dans la section suivante car cette facilitation peut être traitée très différemment selon les codes de DM, nous en profiterons donc pour présenter le code de calcul que nous avons utilisé et auquel nous avons contribué durant cette thèse : `exaStamp`.

2.4 Code de DM : `exaStamp`

Dans cette section, nous présenterons le code de DM que nous avons utilisé, auquel nous avons contribué, et ses spécificités. Ce code se nomme `exaStamp`. Il est développé par le CEA DAM pour répondre à ses besoins et est né de la réécriture du code `STAMP`⁵ pour s'adapter aux nouvelles architectures de calculateurs, en particulier ceux de classe "exaflopique". Il est développé en C++¹⁷ et dès sa conception a intégré un parallélisme sur trois niveaux : le multi-processing via MPI⁶, le multi-threading via OMP⁷, et la vectorisation via l'utilisation d'instructions SIMD⁸ [10, 25]. Désormais, il intègre un niveau supplémentaire de parallélisme pour une grande partie de ses fonctionnalités : l'offloading sur GPU⁹ via CUDA¹⁰ et en partie via HIP¹¹ [26]. L'une des particularités d'`exaStamp` est son équilibrage de charge [27, 28] qui a été développé pour répondre au besoin du CEA DAM de réaliser des simulations hautement hétérogènes comme le micro-jetting [29, 30], le splashing [31-33] ou les simulations d'écaillages que nous voulons réaliser durant cette thèse. Nous présenterons cet équilibrage de charge dans la prochaine sous section (Subsec. 2.4.1). L'autre particularité majeure d'`exaStamp` est qu'il est conçu de façon modulaire afin de faciliter la programmation, la maintenabilité, et la modification du code pour s'adapter aussi bien aux changements d'architectures ou pour répondre à de nouveaux besoins. Cette modularité l'a conduit à évoluer en un nouveau projet `exaNBody` [26] qui a extrait toutes les fonctionnalités sur lesquelles des simulations à N corps peuvent se fonder. Pour le moment, trois codes reposent sur celui-ci : `exaStamp` pour les simulation de DM, `exaDEM` pour les simulations de *Discrete Element Method*, et `exaSPH` pour les simulations de *Smoothed-Particle Hydrodynamics*. À ce jour, parmi ces codes seuls `exaNBody` et `exaDEM` sont disponibles sur GitHub en open source [34, 35] mais `exaStamp` devrait le devenir sous peu. Cette section se conclura par la sous section qui présente ce projet (Subsec. 2.4.2).

2.4.1 Décomposition de domaine

Les simulations pour lesquelles `exaStamp` a été conçu sont des simulations hautement hétérogènes, par hétérogène, nous entendons que sur le domaine total de la simulation, il y a de grandes disparités aussi bien en termes de densités d'atomes, que de températures ou de

5. Simulateur Temporel Atomistique et Moléculaire Parallèle

6. Message Passing Interface

7. OpenMP (Open Multi-Processing)

8. Single Instruction Multiple Data

9. Graphical Processing Unit

10. Compute Unified Device Architecture

11. Heterogeneous-Compute Interface for Portability

pressions. Une illustration de ce type de simulation est présentée dans la figure 2.5. Cette haute hétérogénéité pose de grand problème d'équilibrage de la charge de calcul lorsque exécutée en parallèle. Si l'exécution était purement séquentielle, la charge de calcul serait concentrée par définition sur un seul processeur et donc ne poserait pas de questions sur la façon de répartir la charge de travail. Lorsqu'on exécute un code en parallèle, l'on cherche à répartir la charge de calcul entre les différentes entités qui vont accomplir la tâche pour que ce calcul soit efficace. On comprend aisément qu'en ayant ne serait-ce que deux entités pour exécuter une tâche, si l'on donne 90% de la tâche à accomplir à une entité et 10% à l'autre, cela ne sera pas aussi efficace que de donner équitablement 50% à l'une et à l'autre.

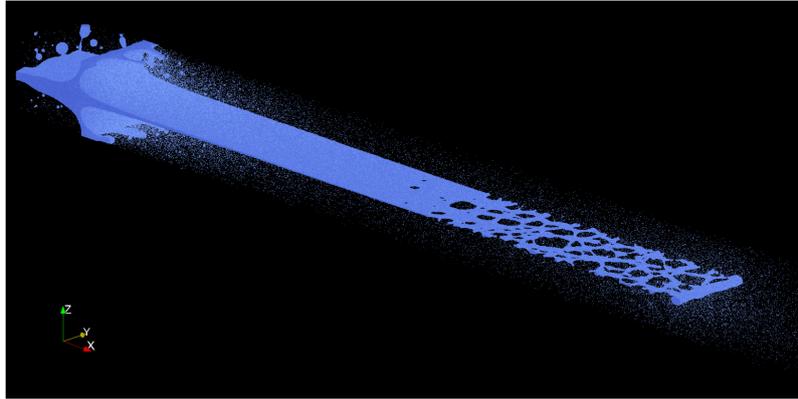


FIGURE 2.5 – Figure issue de la thèse de Raphaël Prat [28]. Illustration de simulation hautement hétérogène : micro-jet d'étain après une nanoseconde d'un système de 200 millions d'atomes.

Pour répondre à ce problème d'équilibrage de charge en simulation de DM hétérogènes, Raphaël Prat a consacré sa thèse [28] à l'implémentation et l'étude de différentes méthodes d'équilibrage de charge dans *exaStamp*. Il a notamment interfacé *exaStamp* avec la librairie de partitionnement *Zoltan* [36] développée aux *Sandia National Laboratories (USA)*. Deux méthodes d'équilibrage de charge ont été retenues et sont utilisées conjointement, une étant utilisée au niveau des processus MPI et l'autre est utilisée au sein des processus MPI pour équilibrer la charge de calcul entre les threads. La première méthode, celle utilisée pour partitionner les processus MPI, décompose le domaine de simulation en plusieurs sous domaines et chaque sous domaine est confié à un unique processus. La brique de base pour le découpage en sous domaine est la cellule de Verlet et le découpage se fait en utilisant la méthode RCB¹² [37]. Cette méthode minimise une fonction de coût définie à partir du nombre de cellules et du nombre d'atomes contenu dans un sous domaine en coupant l'espace de façon récursive jusqu'à obtenir un nombre de sous domaines égal aux nombres de processus demandés. Les sous domaines générés sont tous des pavés réguliers de tailles variables (mais toujours des multiples des cellules de Verlet). Les simulations étant hautement hétérogènes, l'équilibrage de charge entre les différents sous domaines MPI n'est pas suffisant pour équilibrer le calcul. De grandes disparités de densité peuvent persister en interne. C'est pour cela qu'intervient la deuxième méthode d'équilibrage de charge. Cette dernière utilise l'AMR¹³ pour subdiviser les cellules de Verlet au besoin et les répartir entre les différents threads contenus dans un processus MPI pour équilibrer la charge de calcul au sein du

12. Recursive Coordinate Bisection

13. Adaptive Mesh Refinement

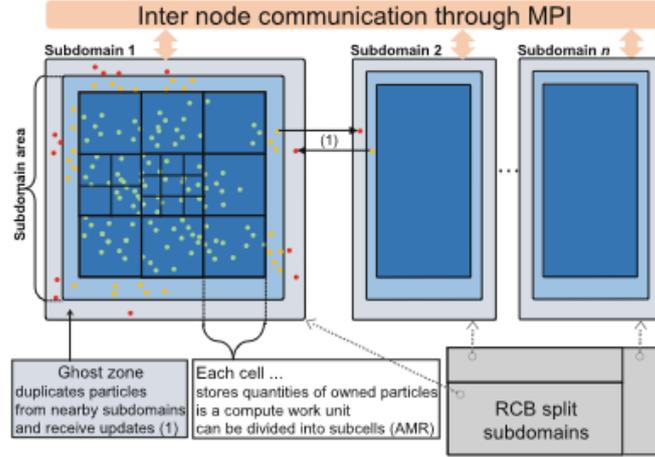


FIGURE 2.6 – Figure issue de l’article [26]. Vue d’ensemble de la décomposition de domaines inter-processus et de l’équilibrage de charge intra-processus du code exaStamp et plus généralement de exaNBody.

sous-domaine. La structure utilisée pour l’AMR est une structure en octree. La figure 2.6 récapitule et illustre le couplage de ces deux méthodes telles qu’elles sont implémentées dans exaStamp. Sur les mêmes simulations hétérogènes, exaStamp démontrait de meilleures performances que LAMMPS qui n’utilise à notre connaissance que du partitionnement par RCB.

2.4.2 Plateforme de développement : exaNBody

Comme évoqué en introduction de cette section, un des points forts d’exaStamp est sa modularité qui a conduit au projet exaNBody [34] en tant que plateforme de développement et support de deux autres applications : exaDEM [35] et exaSPH. Ce projet a récemment fait l’objet d’une publication [26] à laquelle j’ai contribué. Ce papier présente notamment la pile logicielle qui compose le projet et est représenté dans la figure 2.7. Cette pile se compose de notre librairie Onika qui sert de support d’exécution et encapsule les appels aux bibliothèques bas niveaux (CUDA, HIP, ...) en fonction de la plateforme cible, ensuite vient exaNBody le framework qui encapsule les composants génériques pour l’exécution des applications parallèles (le schéma numérique, les I/O, l’équilibrage de charge, ...), puis viennent les applications qui spécialisent les composants ou en fournissent de nouveaux dédiés à cette application (potentiels interatomiques en DM, loi de Hooke en DEM, ...).

Le développement d’exaStamp puis d’exaNBody sous cette forme modulaire de composants a été motivé par le besoin d’avoir un logiciel durable (sustainable software), c’est-à-dire conserver un logiciel performant tout en maintenant le code suffisamment simple pour intégrer de nouvelles évolutions. Dans cette idée, chaque composant est une brique logicielle dont la structure de donnée est “inconsciente” (Oblivious data structure), c’est-à-dire, qu’elle n’a pas d’a priori sur la façon dont sont traitées les données à l’intérieur de cette brique. La seule chose dont sont au courant les structures “inconscientes” sont les données qu’elles prennent en entrée et celles qu’elles renvoient en sortie. Cela permet de mettre en place un flot de données entre les composants qui est déterminé au runtime sous la forme d’un flux de tâches

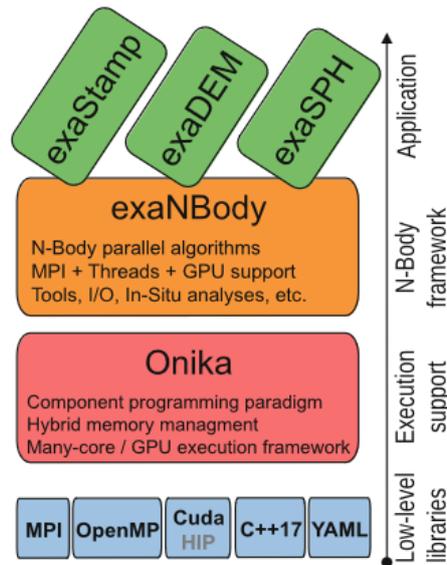


FIGURE 2.7 – Figure issue de l’article [26]. Illustration de la pile logicielle de exaNBody et ses applications.

séquentielles (STF¹⁴) [38], qui sont en fait parallèles sans le savoir (“inconscience”). D’un point de vue implémentation, chacun de ces composants est une classe qui hérite de la classe `OperatorNode` et déclare ses entrées, sorties explicitement avec l’API `ADD_SLOT` fournie avec la classe mère. Chacun de ses composants sont compilés indépendamment et sont assemblés à l’exécution par le programme principal à partir d’un fichier d’input au format YAML qui décrit cet assemblage. Le résumé illustratif de tout ce processus est présenté dans la figure 2.8.

14. Sequential Task Flow

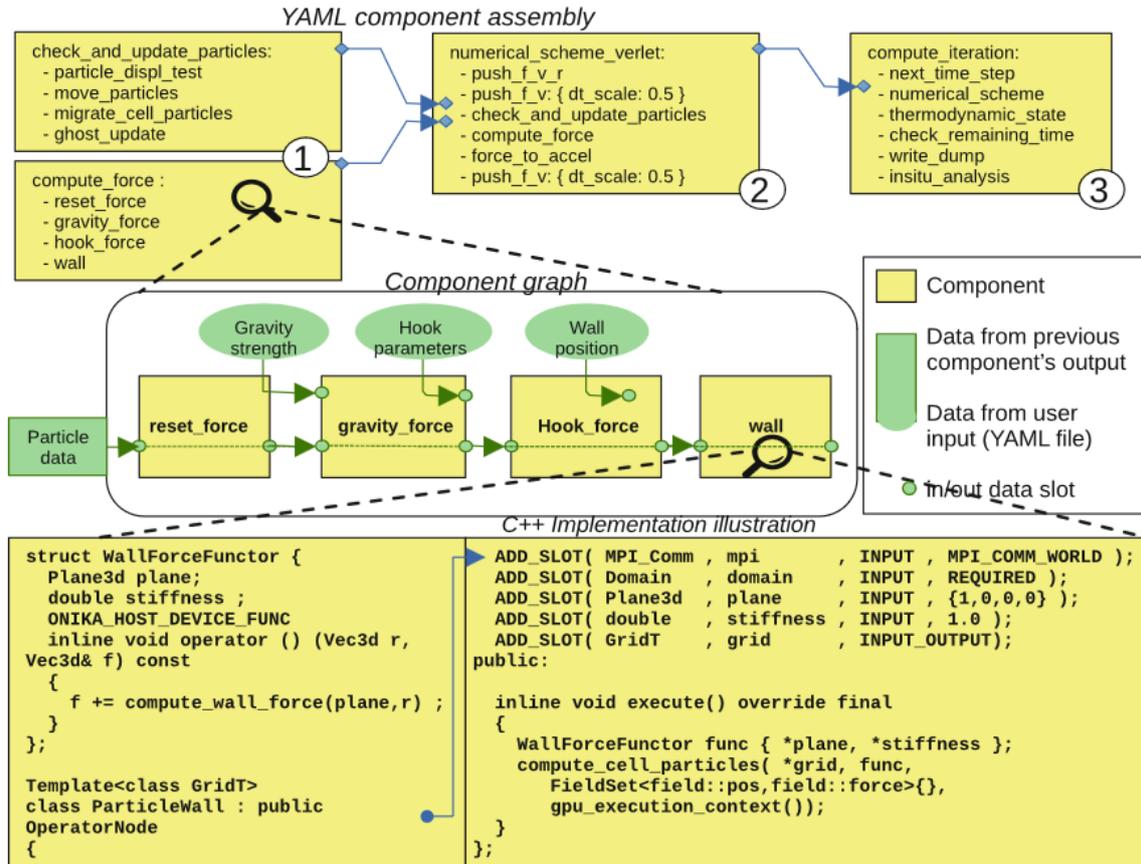


FIGURE 2.8 – Figure issue de l’article [26]. Exemple d’assemblage de composants d’`exaNBody` en utilisant la syntaxe `YAML`. 1) Assemblage de composant développé en `C++` sous la forme d’un flux de tâches séquentielles, formant un lot de composants. 2) et 3) illustre l’agrégation de lots de composants sur 2 niveaux jusqu’à créer l’ensemble de la simulation.

Conclusion intermédiaire

Nous avons présenté les grandes lignes de la dynamique moléculaire telle que nous l’utilisons : les différentes familles de potentiels, le schéma numérique, les algorithmes de résolutions et le code que nous utilisons pour effectuer nos simulations. Nous allons maintenant présenter les systèmes physiques que nous allons simuler :

- du Tantale, un matériau étalon pour les études de physique des chocs
- des matériaux modèles dont on peut contrôler la fragilité et la ductilité par leur potentiel et qui n’ont encore jamais été utilisés pour étudier l’endommagement sous choc

2.5 Implémentation d’un potentiel de Tantale

Nous avons décidé d’étudier le comportement du Tantale sous choc et son endommagement en DM pour plusieurs raisons. Une des premières raisons est que le Tantale est un matériau stable sur une grande plage thermodynamique, c’est-à-dire, qu’il reste dans la

même phase cristalline, en l’occurrence cubique centré (BCC¹⁵), dans une grande gamme de température et de pression, notamment le long de son Hugoniot (Chap 1.). La physique de la dynamique de l’endommagement n’étant pas encore totalement comprise, ajouter à ce stade de l’étude un phénomène de changement de phase risquerait de trop compliquer l’interprétation des résultats. Dans un futur plus ou moins proche, des études de ce type devront tout de même être menées, étant donné que la plupart des éléments subissent un ou des changements de phases sous choc. Une autre des raisons de porter notre choix sur le Tantale est que son endommagement sous choc a été et est toujours grandement étudié, aussi bien expérimentalement [39-47], qu’en DM [48-53], en partie à cause de la première raison que nous avons évoquée. Nous pourrions donc fournir avec notre étude des résultats comparables à l’état de l’art dans le domaine. Finalement, la dernière raison qui nous a poussé à choisir ce matériau, est la disponibilité dans la littérature d’un potentiel de Tantale de type EAM spécialement ajusté pour les études de chocs, sur des données DFT et des données expérimentales [54]. Le type de potentiel EAM étant relativement peu cher en temps de calcul, il permet donc de faire des simulations de “grande” taille, le facteur le plus limitant en termes d’approximation dans notre cas. Nous avons donc choisi d’implémenter ce potentiel dans exaStamp pour réaliser la partie de notre étude sur un matériau réel. La première sous section de cette section présente le formalisme EAM général, la seconde présente son formalisme dans le cadre de ce potentiel et la divergence que nous avons observée entre une des fonctions décrite dans le papier de référence et les données fournies, puis la dernière sous section présente la reproduction des mesures de tenseur de contraintes que nous avons réalisées pour valider notre implémentation.

Toutes les études de DM précédemment citées [48-53] utilisent ce potentiel mais sont pourtant restées limitées en taille malgré la puissance de calcul disponible de nos jours. Nous verrons dans le chapitre suivant (Chap. 3) le verrou que nous avons identifié et comment nous proposons de le lever afin d’améliorer ces études.

2.5.1 Embedded Atom Model (EAM)

Le modèle de l’atome immergé, ou Embedded Atom Model, est un modèle d’interaction potentielle, développé par Daw et Baskes [17], qui considère comme pour les potentiels de paires que l’énergie d’un atome E_i s’exprime sous la forme d’une somme de termes de paires ϕ qui dépendent uniquement de la distance interatomique r_{ij} mais ajoute un terme F qui représente l’influence qu’aurait un nuage d’électron uniforme de densité $\bar{\rho}_i$ sur l’atome i s’il était immergé dedans. Le modèle exprime ainsi :

$$E_i = F(\bar{\rho}_i) + \sum_{i \neq j} \phi(r_{ij}) . \quad (2.12)$$

Pour calculer la densité électronique $\bar{\rho}_i$, ce modèle considère qu’elle s’exprime comme la somme de fonctions de paire w qui dépendent aussi uniquement de la distance interatomique r_{ij} de sorte que :

$$\bar{\rho}_i = \sum_{i \neq j} w(r_{ij}) . \quad (2.13)$$

Ce modèle dépend donc uniquement de trois fonctions $\phi(r_{ij})$, $F(\bar{\rho}_i)$ et $w(r_{ij})$ dont les formes dépendent du potentiel considéré. La sous section suivante présente les formes utilisées par Ravelo et al. [54] pour leur potentiel de Tantale et deux jeux de paramètres qu’ils ont obtenus en les ajustant sur une base de données DFT et des points expérimentaux.

15. Body Centered Crystal

2.5.2 Fonctions du potentiel de Ravelo et al. [54]

Pour décrire la fonction $F(\bar{\rho}_i)$ Ravelo et al. [54] ont choisi d'utiliser une fonction de Rydberg [55] telle que formalisée par Rose et al. [56] pour l'ajuster sur des courbes froides calculées en DFT (ici sur celle du Tantale). Ainsi :

$$F(\bar{\rho}_i) = E_{EOS}(a^*) - \frac{1}{2} \sum_{i \neq j} \phi(r_{ij}) \quad (2.14)$$

avec :

$$E_{EOS}(a^*) = -E_c(1 + a^* + f_3 a^{*3} + f_4 a^{*4})e^{-a^*}, \quad (2.15)$$

$a^* = \alpha(a/a_0 - 1)$ un paramètre adimensionné proportionnel à la déformation linéaire et $\alpha = \sqrt{(9B_0V_0)/E_c}$ une mesure adimensionnée de la force de répulsion anharmonique. Les paramètres V_0 le volume d'équilibre, a_0 le paramètre de maille, l'énergie de cohésion E_c et le module d'élasticité isostatique B_0 sont des constantes obtenues à température et pression nulle. Pour le terme d'interaction de paire ϕ , ils ont choisi d'utiliser une fonction de Rydberg modifiée pour calculer la partie répulsive du potentiel et d'utiliser une fonction polynomiale de courte portée pour la partie attractive telle que :

$$\phi(r) = \begin{cases} U_0(1 + r^* + \beta_3 r^{*3} + \beta_4 r^{*4})e^{-r^*} & (0 \leq r \leq r_s), \\ U_0(r_c - r)^s \sum_{i=1}^4 a_i (r_c - r)^{(i-1)} & (r_s \leq r \leq r_c), \\ 0 & (r > r_c) \end{cases} \quad (2.16)$$

avec $r^* = \alpha_p(r/r_1 - 1)$ la distance adimensionnée de façon analogue à a^* , et où les paramètres $r_1, r_s, s, \alpha_p, U_0, \beta_3$ et β_4 sont des paramètres d'ajustements et les coefficients a_i sont calculés pour que la fonction de Rydberg, la fonction polynomiale et leurs dérivées de première, seconde et troisième ordre soit égales entre elle en $r = r_s$. Pour la fonction de poids w ils ont choisi de reprendre la forme arbitraire utilisée dans [57] qui ressemble qualitativement à une gaussienne et qui s'annule au delà de r_c telle que :

$$w(r) = \begin{cases} \rho_0 \left[\frac{r_c^p - r^p}{r_c^p - r_0^p} \right]^q & (0 \leq r \leq r_c), \\ 0 & (r > r_c) \end{cases} \quad (2.17)$$

où r_0 est la distance au plus proche voisin à température et pression nulle, ρ_0 est une constante de normalisation de l'équation (2.13) et les paramètres p, q et r_c sont des paramètres d'ajustements.

Ils fournissent deux jeux de paramètres d'ajustements récapitulés dans la table suivante :

Parameter	Ta1	Ta2
E_c (eV)	8.1	8.1
α	4.934	4.950
a_0 (Å)	3.304	3.304
f_3	0.0	-0.035744
f_4	0.0	-0.020879
U_0 (eV)	0.80	1.1094
r_1 (Å)	2.860	2.7826
α_p	4.70	4.6463
β_3	0.0	0.160
β_4	0.0	0.060
r_s (Å)	2.86	2.8683
s	7.50	8.00
r_c (Å)	5.30	5.5819
a_1 (Å ^{-s})	-0.11745	-0.039742
a_2 (Å ^{-(s+1)})	0.12309	0.038568
a_3 (Å ^{-(s+2)})	-0.043904	-0.012714
a_4 (Å ^{-(s+3)})	0.0053178	0.0014196
p	4.0	5.9913
q	4.0	8.0
ρ_0	0.077723	0.074870

TABLE 2.1 – Table issue de l’article [54], qui récapitule l’ensemble des paramètres pour les fonctions de Ta1 et Ta2.

Le premier jeu correspondant au potentiel Ta1 a été obtenu en fixant les termes d’ordre 3 et 4 des fonctions de Rydberg à 0, à la différence du jeu du potentiel Ta2.

On peut noter que la fonction $F(\bar{\rho}_i)$ telle que définie par Ravelo et al., n’a pas une forme qui dépend explicitement de $\bar{\rho}_i$, mais de a^* et ne peut pas être résolue directement lors d’une simulation par la formule (2.14). Cette fonction doit être tabulée en amont sur un cristal parfait (à $T = 0$ K), dont on fait varier le volume en incrémentant le paramètre de maille a et pour lesquels on calcule d’une part le terme $F(\bar{\rho}_i)$ avec les équations (2.14) et (2.15) et d’autre part le terme $\bar{\rho}_i$ avec les équations (2.13) et (2.17) qu’on tabule ensemble pour associer aux différentes valeurs de $\bar{\rho}_i$ la valeur de $F(\bar{\rho}_i)$. Cette procédure n’est pas décrite dans leur papier et a posé quelques soucis lors de l’implémentation de ce potentiel dans exaStamp. LAMMPS, le code majoritairement utilisé dans la communauté pour ce type de simulation implémente les potentiels EAM de façon entièrement tabulé pour les 3 fonctions $\phi(r_{ij})$, $F(\bar{\rho}_i)$ et $w(r_{ij})$ et les tabulations de la majorité des potentiels interatomiques de ce type utilisés par LAMMPS se trouvent dans la base de données de potentiels interatomiques du NIST [58]. ExaStamp ne tabule pas toute les fonctions EAM et le parti pris des utilisateurs est plutôt de favoriser l’utilisation de fonctions analytiques car la tabulation peut dans certains cas entraîner une accumulation d’erreurs à grand temps de simulation et faire légèrement dériver l’énergie du système. Dans le cas de ce potentiel, nous avons donc seulement implémenté la fonction $F(\bar{\rho}_i)$ de façon tabulée et les 2 autres fonctions ($\phi(r_{ij})$ et $w(r_{ij})$) de façon analytique. La figure 2.9 tirée du papier de référence présente la forme des fonctions $\phi(r_{ij})$ et $F(\bar{\rho}_i)$.

Pour vérifier notre implémentation, nous avons comparé ces fonctions avec celles que nous avons implémentées (Fig. 2.10). Seul Ta1 est représenté mais Ta2 a aussi été vérifié, les fonctions en jaune sont les tabulations fournies par les auteurs pour LAMMPS et les fonctions

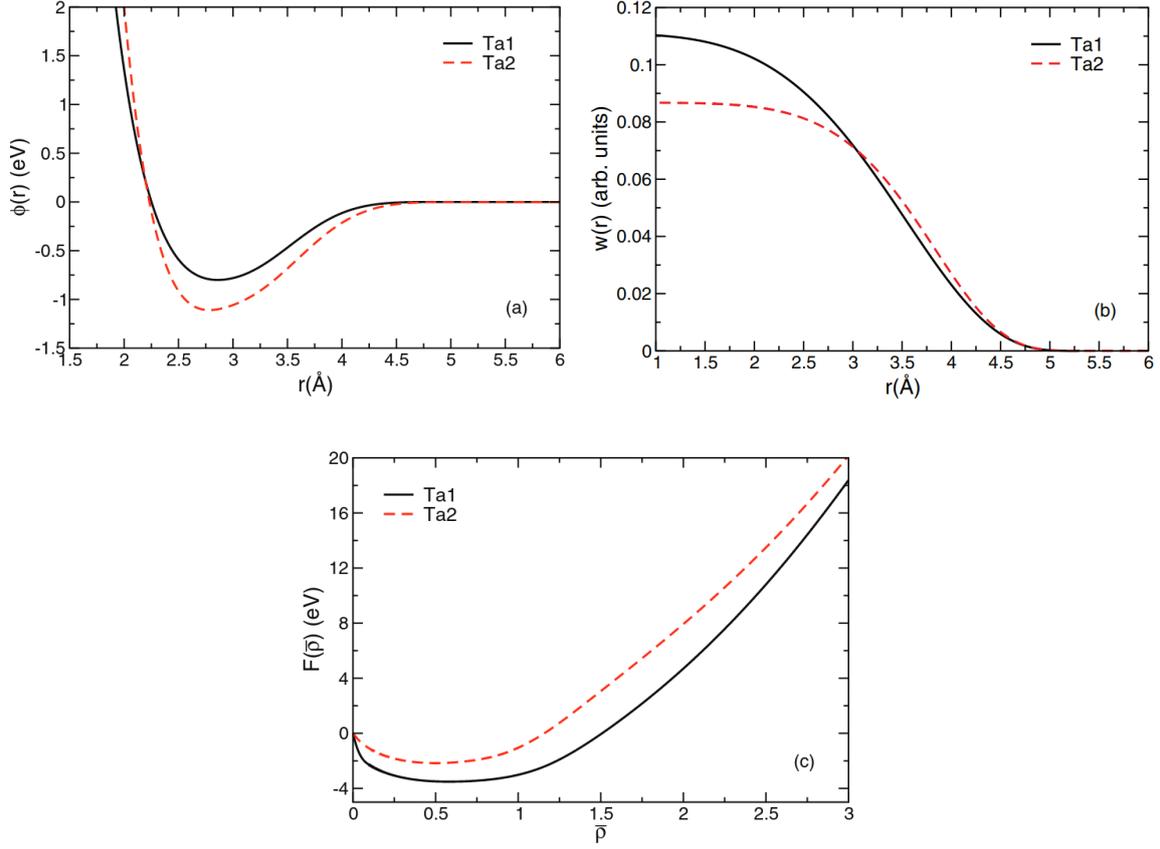


FIGURE 2.9 – Figures extraites de l’article [54]. Fonctions des modèles EAM Ta1 et Ta2 : (a) $\phi(r)$ (Eq. 2.16), (b) $w(r)$ (Eq. 2.17), (c) $F(\bar{\rho})$ (Eq. 2.14).

en pointillé rouge sont nos fonctions. Le terme de paire $\phi(r_{ij})$ est parfaitement reproduit par notre implémentation, ainsi que la fonction $w(r_{ij})$. Par contre, nous avons observé une différence notable pour la fonction d’“embedding” $F(\bar{\rho}_i)$. Cette différence est localisée à faible valeur de $\bar{\rho}_i$ entre 0 et 0.08, le reste de la fonction coïncide parfaitement avec la tabulation des auteurs. Nous avons rigoureusement respecté la description de la fonction réalisée dans le papier. Nous expliquons cette différence par le fait que la fonction dans sa forme stricte tend asymptotiquement vers 0 quand la densité tend vers 0, la force s’exerçant sur un atome étant en partie issue de la dérivée de cette fonction, dans les cas où la “densité” $\bar{\rho}_i$ autour d’un atome deviendrait nulle, le calcul des forces s’exerçant sur lui donnerait une force infinie. Les auteurs ont donc du régulariser la forme de la fonction pour éviter ce problème. Cependant ils ont omis la description de cette procédure de régularisation dans leur article.

Pour notre implémentation dans exaStamp nous avons gardé la tabulation des auteurs pour la fonction $F(\bar{\rho}_i)$, d’une part pour éviter le problème de force infinie, et d’autre part car toutes les études réalisées avec LAMMPS utilisent cette fonction. Utiliser une autre fonction reviendrait à simuler un autre matériau et nous ne pourrions pas avoir des résultats comparables à la littérature.

Afin de définitivement valider notre implémentation de ce potentiel, nous comparons dans la sous section suivante des propriétés physiques des matériaux Ta1 et Ta2, en l’occurrence les valeurs du tenseur des contraintes en fonction de la déformation uniaxiale des matériaux

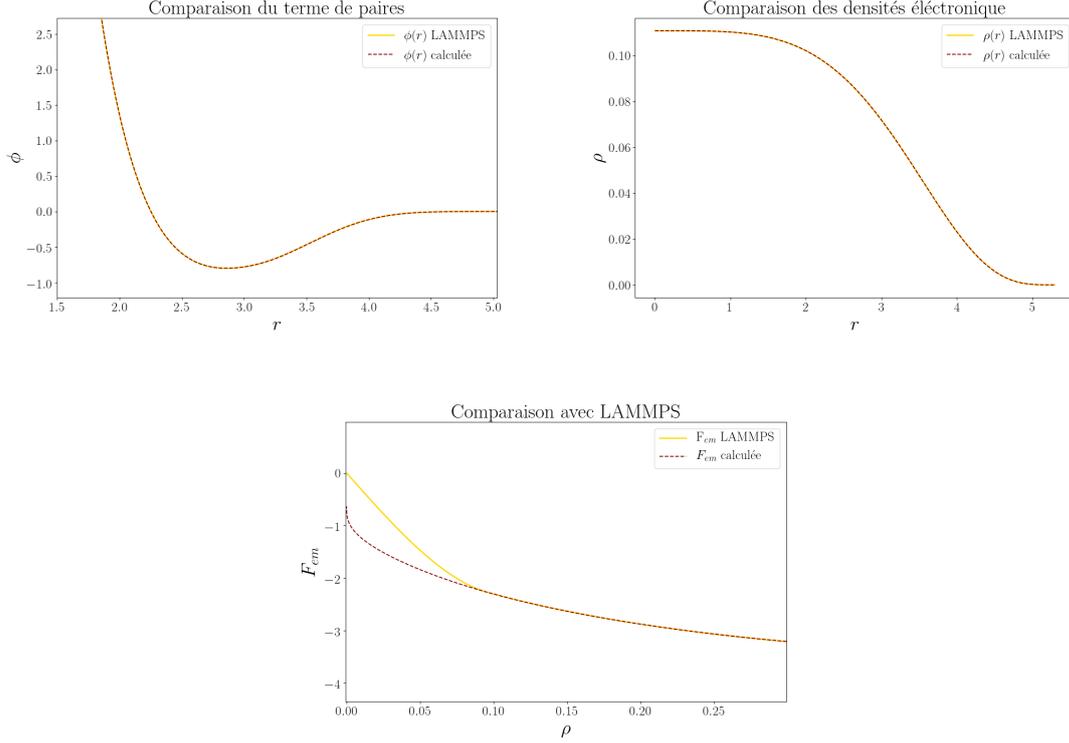


FIGURE 2.10 – Comparaison des tabulations des fonctions $\phi(r)$, $F(\bar{\rho})$ et $w(r)$ de Ravelo et al. disponibles sur le site du NIST [58] pour Ta1 et de notre implémentation de ces fonctions.

en compression dans différentes orientations cristallines.

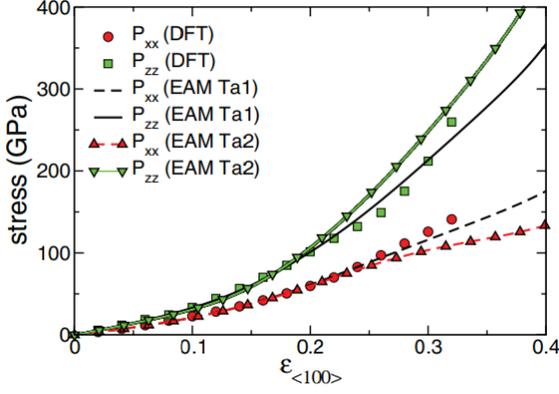
2.5.3 Reproduction du tenseur des contraintes

Une des grandeurs physiques utilisée par Ravelo et al. pour ajuster leurs potentiels est la valeur du tenseur des contraintes en fonction de la déformation uniaxiale en compression des cristaux dans trois orientations cristallines : $\langle 100 \rangle$, $\langle 110 \rangle$ et $\langle 111 \rangle$. Le tenseur est calculé par l'intermédiaire du théorème du viriel et est exprimé dans la convention positive :

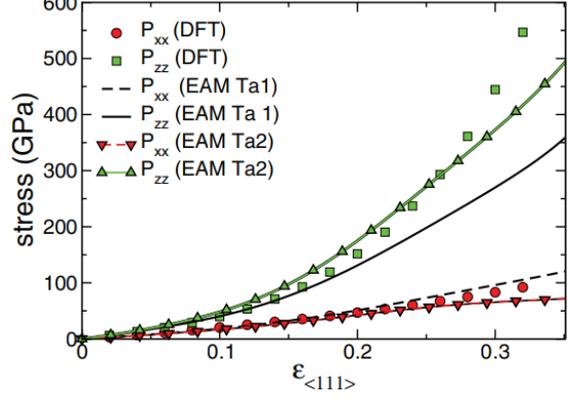
$$\bar{\sigma} = \begin{pmatrix} P_{xx} & P_{xy} & P_{xz} \\ P_{yx} & P_{yy} & P_{yz} \\ P_{zx} & P_{zy} & P_{zz} \end{pmatrix}. \text{ La figure 2.11 présente dans la partie supérieure les courbes de}$$

Ravelo et al. pour les orientations $\langle 100 \rangle$, et $\langle 111 \rangle$. La partie inférieure présente nos courbes pour ces mêmes orientations. Figure 2.11c, nous avons reporté les points DFT de Ravelo et al. et adopté leur légende. Figure 2.11d, nous avons adopté une légende différente et tracé en plus la composante P_{yy} qui est confondue avec P_{xx} comme attendu pour un cristal BCC déformé uniaxialement selon $\langle 111 \rangle$.

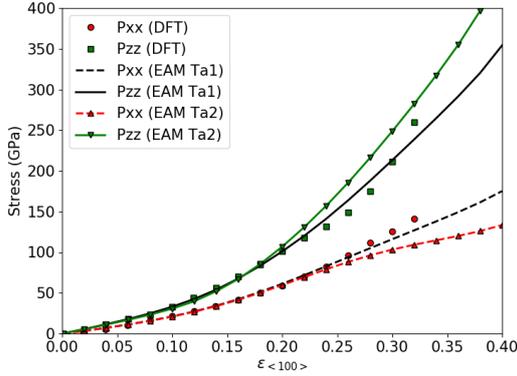
Dans notre cas, pour appliquer la déformation à notre système, nous avons généré un cristal parfait orienté selon $\langle 100 \rangle$ dans notre repère de simulation puis nous avons appliqué à chaque composante du vecteur \mathbf{x}_{form} (Eq. 2.11) les coefficients de déformation nécessaire pour obtenir la déformation voulue dans chaque direction. Cette méthode simple est rendue possible par exaStamp, car les positions des atomes sont stockées en coordonnées réduites



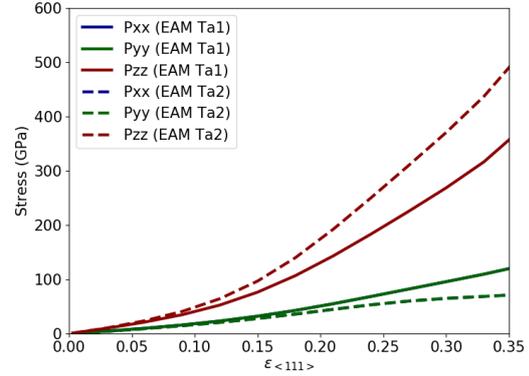
(a) Figure extraite de [54]



(b) Figure extraite de [54]



(c) Notre reproduction de (a)



(d) Notre reproduction de (b)

FIGURE 2.11 – Comparaison des valeurs de tenseur des contraintes de Ravelo et al. et celles obtenues avec notre implémentation. Les figures supérieures (a et b) sont extraites de l'article [54], et les figures inférieures (c et d) sont issues de notre reproduction de leurs simulations avec notre implémentation. Nous avons reporté sur (c) les points DFT de (a) et utilisé la même légende.

par rapport au vecteur \mathbf{x}_{form} . Par exemple, LAMMPS ne peut appliquer des déformations que selon la direction z de son repère, l'utilisateur doit donc générer lui-même au préalable les configurations atomiques orientées dans la direction z .

Nous reproduisons parfaitement les valeurs de tenseur des contraintes des potentiels de Ravelo et al., nous validons donc notre implémentation de ces potentiels. Dans la suite de cette thèse, nous choisissons uniquement d'utiliser Ta1 car les valeurs de tenseurs de contraintes sont plus proches des valeurs DFT, sa phase BBC est bien plus stable en pression que Ta2 comparé à la phase HCP¹⁶ que les potentiels EAM qui ont tendance à favoriser en compression [54], dans le cas du Tantale cette phase n'est pas stable, et dans leur Erratum [59], Ravelo et al. ont seulement fourni des valeurs de a_i assez précises pour garantir la continuité de classe C^3 uniquement pour Ta1, celles de l'article ne l'étant pas initialement.

16. Hexagonal Close Packed

2.6 Implémentation d'un potentiel modèle

Nous décrivons maintenant l'autre potentiel que nous avons décidé d'utiliser durant cette thèse. Ce potentiel est un modèle dont les paramètres sont ajustés non pas pour reproduire un matériau particulier comme le potentiel que nous avons présenté précédemment, mais pour réaliser des études purement phénoménologiques. Ce potentiel modèle permet de jouer sur la fragilité et la ductilité du matériau simulé uniquement à partir d'un paramètre. Il a été proposé par Rajan et al. [60] et a principalement été utilisé pour étudier la physique de l'usure par abrasion en tribologie [61-64]. En utilisant ce potentiel, Aghababaei et al. [65] ont notamment mis en évidence l'existence d'une taille caractéristique de contact en dessous de laquelle deux aspérités frottant l'une contre l'autre, auront un comportement purement ductile avec une disparition progressive des aspérités, et au-dessus de laquelle un comportement fragile est observé avec la création de fractures et une croissance des débris générés. Ce potentiel est très simple car il ne contient qu'un terme de paire mais à tout de même une physique très intéressante. Il n'a encore jamais été utilisé pour des études d'endommagement sous choc, durant cette thèse, l'objectif avec ce potentiel sera de tester dans un premier temps s'il est possible de l'utiliser pour ce type de simulation très dynamique, dans un second temps d'étudier l'influence de la fragilité et la ductilité du matériau sur l'endommagement, et éventuellement retrouver une distance caractéristique de transition fragile ductile. Nous essayerons de remplir ces objectifs dans le chapitre (Chap. 5), dans cette section nous présentons dans une première sous section le modèle et ses paramètres, puis la deuxième sous section présentera une reproduction de simulations de la littérature que nous avons réalisée pour valider notre implémentation, et finalement la dernière sous section présente un premier essai du potentiel en traction sur une plaque trouée comme preuve de concept de son applicabilité à des simulations d'écaillage.

2.6.1 Fonction de paire et paramètres du modèle

Ce potentiel modèle est un potentiel de Morse [66] modifié avec une fonction polynomiale pour introduire plus ou moins de fragilité ou de ductilité dans la simulation en contrôlant la partie attractive du potentiel à partir d'un unique paramètre : le rayon de coupure. Usuellement, le rayon de coupure n'est pas un paramètre variable une fois que le potentiel a été ajusté car modifier le rayon de coupure implique un changement du matériau simulé mais dans notre cas nous cherchons à avoir différents matériaux de ductilité différentes. Pour le formalisme du potentiel, nous utilisons celui de [61] pour leur simulation 3D et il s'exprime tel que :

$$\frac{\phi(r)}{\epsilon_0} = \begin{cases} (1 - e^{-\alpha(r-r_0)})^2 - 1 & (0 < r \leq 1.05r_0), \\ c_1 \frac{r^3}{6} + c_2 \frac{r^2}{2} + c_3 r + c_4 & (1.05r_0 \leq r \leq r_{cut}), \\ 0 & (r_{cut} \leq r) \end{cases} \quad (2.18)$$

avec r_0 la distance au fond du puit de potentiel, ϵ_0 la profondeur du puit, r_{cut} le rayon de coupure, α le paramètre qui contrôle la largeur du puit et les coefficients c_n sont calculés pour que le terme de Morse, le terme polynomial et leurs dérivées soient continus en $1.05r_0$ et à r_{cut} . Le facteur 1.05 permet de conserver le même fond de puits jusqu'à 5% de déformation et donc de conserver les mêmes propriétés élastiques pour les différents matériaux, tout en permettant de modifier la queue du potentiel et donc leurs propriétés plastiques. Plus le rayon de coupure sera grand, plus le matériau sera ductile, et plus le rayon de coupure sera

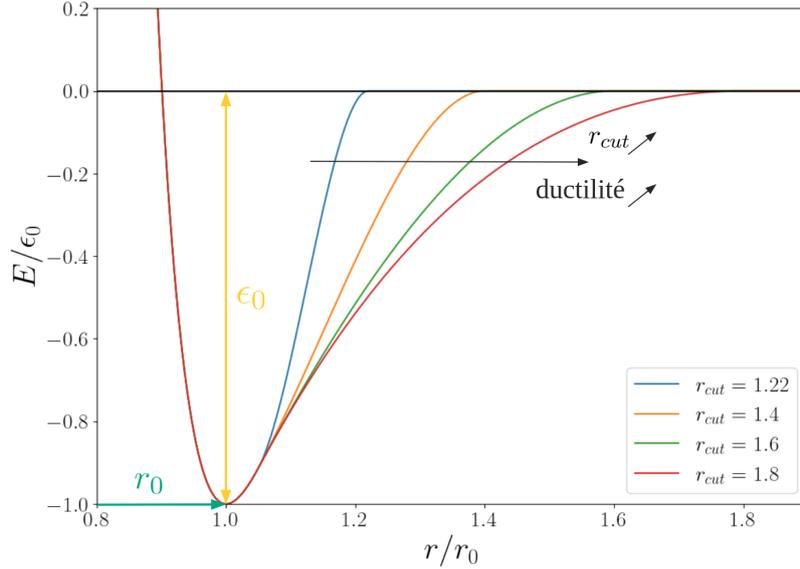


FIGURE 2.12 – Fonctions de paire du potentiel modèle (Eq. 2.18) pour différentes valeurs de r_{cut} et $\alpha = 7$.

petit, plus le matériau sera fragile. La figure 2.12 illustre le terme de paire pour plusieurs valeurs de rayon de coupure.

La phase cristalline stable de ce potentiel est cubique faces centrées (FCC¹⁷), on peut noter que r_0 est la distance entre les atomes diagonaux d’une face. Le système est entièrement adimensionné par rapport aux paramètres r_0 , ϵ_0 et la masse d’un atome m_0 , néanmoins pour lancer une simulation avec exaStamp, il faut donner une valeur et une unité à ces paramètres. Nous avons fixé les valeurs de ces paramètres à 1 et laissé les unités par défaut d’exaStamp pour ces grandeurs.

2.6.2 Reproduction du comportement fragile et ductile

Pour valider notre implémentation, nous avons décidé de reproduire des simulations de collision d’aspérités sphériques sur des surfaces en glissement. Nous faisons ce choix car, contrairement au potentiel précédent, il ne cherche pas à reproduire les propriétés d’un matériau particulier. Les données à disposition dans la littérature décrivant ses propriétés physiques sont donc assez limitées. Par exemple, les valeurs de tenseur des contraintes en déformation sont absentes. Cette fois nous voulons donc reproduire la phénoménologie liée à ce potentiel telle qu’elle est étudiée dans la communauté et non pas ses propriétés intrinsèques comme dans le cas du Tantale.

Nous présentons ici seulement une simulation présentant un comportement fragile par l’apparition de fracture car nous verrons que celle-ci illustre aussi le comportement ductile du potentiel. Mais nous avons aussi réalisé d’autres simulations présentant un comportement purement ductile tel qu’attendu en diminuant la taille des aspérités [65]. Pour cette simulation, nous avons utilisé un rayon de coupure égal à $1.22r_0$ avec $\alpha = 7$. Le système est composé d’aspérités de $40r_0$ de rayon sur des plaques de surface $200 \times 200r_0^2$ et d’épaisseur $75r_0$ séparée de $50r_0$. Le système est en condition limite périodique dans les directions x et y et

17. Faces Centered Cubic

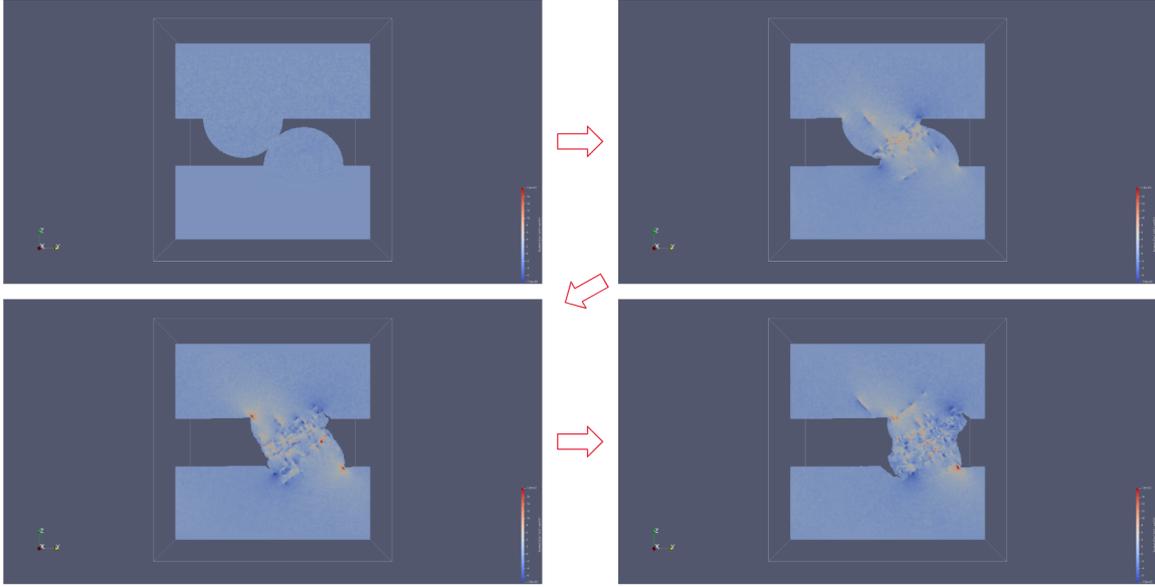


FIGURE 2.13 – Simulation de tribologie modélisant la collision de deux aspérités sphériques avec notre potentiel modèle. Quatre coupes 2D dans un plan normal au mouvement des plaques et centré dans le domaine de simulation à des instants différents. Le sens chronologique est indiqué par les flèches rouge. Le champ représenté en couleur est la trace du tenseur des contraintes.

laissé libre selon z la direction de normale aux plaques. La plaque inférieure est fixe et la plaque supérieure se déplace à une vitesse de $0.05r_0/t_0$ avec un pas de temps de calcul fixé à $0.005t_0$. Pour imposer ces conditions, une couche de 5 mailles cristallines ($5\sqrt{2}r_0$) est figée dans la plaque inférieure et une couche de même épaisseur voit sa vitesse selon y imposée à $0.05r_0/t_0$.

Nous présentons dans la figure 2.13, quatre coupes 2D selon x de la simulation issues d’instants différents : le contact entre les aspérités, l’émission de défauts plastiques suite à la collision des aspérités, l’apparition de fissures aux points de contacts des aspérités avec leur plaque mis en cisaillement par le mouvement des plaques l’une par rapport à l’autre, et la propagation des fissures. Le champ représenté en couleur est la trace du tenseur de contraintes, et permet de voir : les défauts plastiques générés par la collision des aspérités (notamment des dislocations mises en évidence par des dipôles de contraintes), et les zones sollicitées en traction et en compression. Les aspérités sont de l’ordre de grandeur de la distance caractéristique délimitant le comportement ductile du comportement fragile tel que mis en évidence par [65] et nous observons bien que sous cette distance les sollicitations mécaniques imposées au matériau génèrent des défauts plastiques représentatifs d’un comportement ductile et qu’au dessus de cette distance le matériau se fissure pour accommoder les contraintes auxquelles il est soumis, phénomène typique d’un comportement fragile. Avec le même potentiel, mais des aspérités de plus petite taille, les aspérités coulissent seulement l’une sur l’autre et génèrent uniquement des défauts plastiques, les fissures ne sont jamais observées.

Nous reproduisons correctement les simulations de [61], et en particulier le comportement fragile et ductile du potentiel modèle donc nous validons notre implémentation de ce potentiel dans exaStamp. Avant de l’utiliser directement dans des simulations d’écaillage relativement coûteuses en calcul, nous voulons vérifier son applicabilité avec des simulations moins coûteuses

comme preuve de concept d'obtention d'un comportement fragile pour notre cas. C'est l'objet de la sous section suivante (Subsec. 2.6.3).

2.6.3 Essai du potentiel en traction sur plaque trouée

Pour vérifier dans un premier temps l'applicabilité de ce potentiel modèle à des cas d'écaillages, nous voulons vérifier sur des cas plus simples, moins coûteux en calcul, et auxquels nous pouvons nous rapporter, si le potentiel fait preuve d'un comportement fragile. Pour cela, nous avons choisi de soumettre des plaques préalablement trouées à une traction uniaxiale. Une plaque nécessite moins de matière à simuler, et en rapportant les pores d'une simulation d'écaillage aux trous déjà présents, nous évitons la simulation de la propagation d'un choc, sa réflexion et la nucléation des pores, d'où le moindre coût de calcul. Nous avons testé plusieurs configurations de trous, d'abord avec un seul trou au centre de la plaque dont nous avons fait varier le diamètre, puis avec deux trous dans deux configurations, une dans laquelle les trous sont à la verticale l'un de l'autre, et une configuration dans laquelle ils sont à la diagonale l'un de l'autre. Dans ces configurations à deux trous nous avons fait varier le diamètre des trous mais aussi la distance les séparant. Pour chaque configuration, nous avons imposé la traction uniaxiale en imposant une vitesse constante de valeur absolue $0.05r_0/t_0$ et de sens opposé aux bords gauche et droit de la plaque.

La figure 2.14 présente une simulation d'une configuration à deux trous en diagonal l'un par rapport à l'autre, avec des trous d'un diamètre de $20r_0$ et séparés d'une distance de $2r_0$. La traction se fait selon l'axe des x par les bords gauche et droit de la plaque. Quatre instants différents sont représentés et l'ordre chronologique est donné par les flèches rouge. Le champ représenté est la trace du tenseur des contraintes avec en bleu les zones en tension et en rouge les zones en compression. Le premier instant présente la configuration lorsque l'on vient de la soumettre en traction. Le deuxième instant présente les contraintes exercées sur la plaque avant rupture, on peut observer le phénomène de striction sur les bords supérieur et inférieur de la plaque, ainsi que sur le ligament de matière présent entre les trous. Le troisième instant présente la rupture du ligament séparant les trous, l'initiation d'une fissure partant du trou supérieur, et la concentration de contraintes de traction sur le trou inférieur en quatre points. L'instant final présente la rupture complète de la plaque. La fissure supérieure initiée à l'instant précédent a bifurqué en deux fissures quasiment dès son initiation alors que la concentration de contrainte de traction sur le trou inférieur en 4 points a résulté en une seule fissure qui a bifurqué plus tard que la fissure supérieure, et dont une des bifurcations à elle même bifurqué avant rupture de la plaque. On peut aussi constater la présence d'un dipôle de contrainte à la droite de la fissure inférieure, signe de la présence d'une dislocation, certainement émise par la propagation de la fissure. Nous avons positionné les trous suffisamment proche pour observer une rupture du ligament qui s'apparente plus à une rupture ductile en striction, ce qui nous semble être le cas ici, lorsque la distance entre les trous était plus grande, la rupture s'apparentait plus à une rupture fragile avec l'émission de fissures depuis chaque trou de l'un vers l'autre.

Nous arrivons à obtenir avec ces simulations tests à la fois un comportement ductile des matériaux en DM et en particulier un comportement fragile qui habituellement s'observerait à des échelles de simulation bien plus grandes que celles atteignables en DM en simulant de vrais matériaux comme le Tantale. Nous validons donc que ces simulations sont des preuves de concepts de l'applicabilité de ce potentiel à des simulations d'écaillages. De plus, la reproduction du phénomène de ramification des bifurcations de fissure est intéressant pour l'étude de propagation de fissure et montre que l'utilisation de ce potentiel dans ce domaine

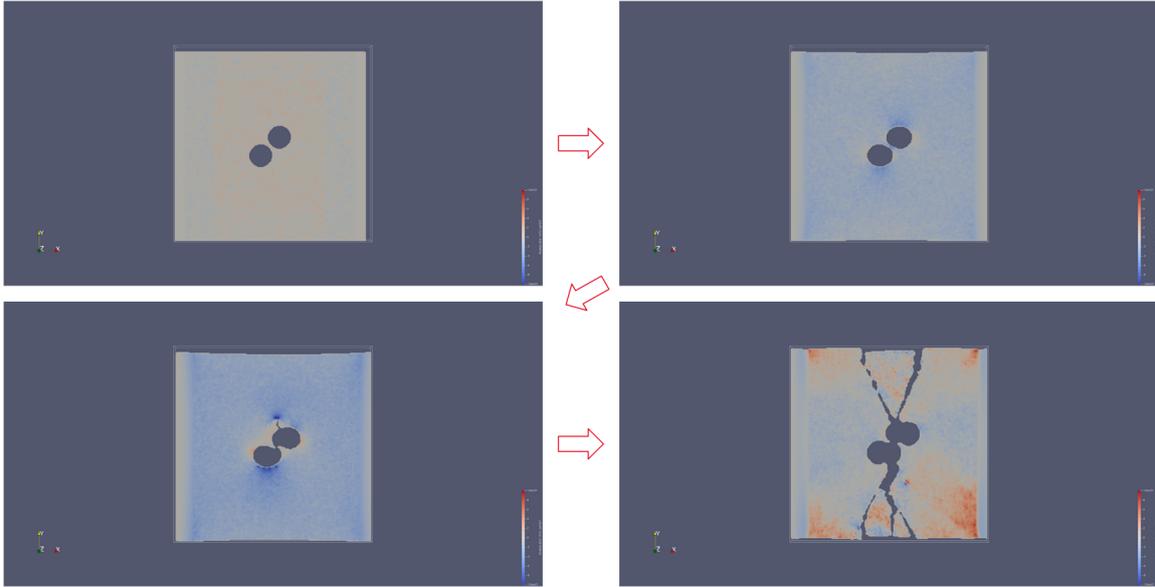
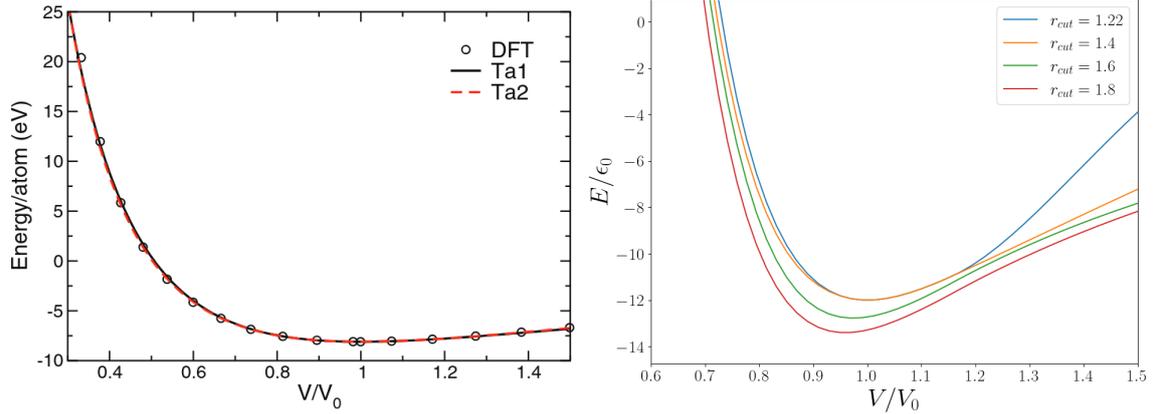


FIGURE 2.14 – Simulation de traction sur une plaque trouée. Quatre instants différents sont représentés, le sens chronologique est indiqué par les flèches rouge. Le champ représenté en couleur est la trace du tenseur des contraintes.

est prometteuse mais cela ne fait pas partie du cadre de cette thèse.

2.7 Système d'adimensionnement

Pour conclure cette partie sur les potentiels que nous avons implémentés pour nos études d'écaillage, nous allons présenter le système d'adimensionnement que nous proposons pour pouvoir comparer les simulations réalisées avec le Tantale et celles réalisées avec le potentiel modèle. Avoir un système d'adimensionnement est nécessaire à la comparaison de ces simulations car les simulations sur le Tantale sont dimensionnées par rapport à la "réalité" alors que les simulations du potentiel modèle sont déjà adimensionnées par rapport aux grandeurs ϵ_0 (énergie de fond de puit du terme d'interaction de paire du potentiel), r_0 (la distance au plus proche voisin et d'équilibre du terme d'interaction de paire du potentiel) et m_0 (la masse d'un atome). Une solution serait d'adimensionner les simulations de Tantale comme les simulations du matériau modèle, en choisissant le fond de puit du terme d'interaction de paire du potentiel comme grandeur d'adimensionnement de l'énergie (U_0 dans la table des paramètres, Tab. 2.1) et la distance r_s à ce fond de puit comme grandeur d'adimensionnement de longueur. Cependant, comme nous l'avons vu dans la section 2.5, dans le cas du potentiel EAM, l'énergie d'un atome ne dépend pas que de la somme d'un terme d'interaction de pair mais aussi d'une fonction d'embedding non négligeable dans le calcul de l'énergie d'un atome. Dans le cas du potentiel modèle, ϵ_0 a été choisi car il était représentatif de l'énergie du système à l'équilibre, et ce n'est pas le cas pour U_0 dans le cas du Tantale. De même pour la r_0 dans le cas du modèle et de r_s dans le cas du Tantale. Nous devons donc trouver des grandeurs d'énergie et de longueur représentatives de ces systèmes à l'équilibre.



(a) Figure issue de [54]. Équation d'état du Tantale à 0K. (b) Équations d'état à 0K du potentiel modèle en fonction de différentes valeurs de r_{cut} .

FIGURE 2.15 – Équations d'états à 0K des deux formalismes de potentiel que nous avons implémenté.

Le point commun de nos systèmes est qu'ils sont tous deux des cristaux, et qui plus est des cristaux cubiques donc nous pouvons déjà avoir une longueur caractéristique de ces systèmes à l'équilibre : le paramètre de maille. Il correspond à la longueur d'une arête de la maille cristalline. Dans le cas du Tantale, ce paramètre noté a_0 dans la table 2.1, vaut 3.304\AA . Par la suite nous le noterons a_{Ta} pour réserver la notation a_0 au potentiel modèle. En effet, dans le cas du potentiel modèle, en gardant r_0 comme la distance au fond du puit, alors le paramètre du cristal vaut $\sqrt{2}r_0$ et nous le noterons a_0 par la suite. Dans le même esprit, pour déterminer une énergie de référence pour l'adimensionnement, nous pouvons calculer l'énergie de chaque cristal à l'équilibre. C'est-à-dire qu'en prenant un cristal parfait pour chacun de nos systèmes et en faisant varier le volume de la maille cristalline, on peut calculer l'énergie associée à chaque maille cristalline. L'énergie minimum est celle au volume d'équilibre (équivalent au paramètre de maille d'équilibre). Cependant, nos deux cristaux n'ont pas le même nombre d'atomes par maille cristalline conventionnelle, il y en a deux dans le cas du Tantale (cristal BCC), et quatre dans le cas du modèle (cristal FCC), nous normalisons donc ces énergies de cristaux au repos par le nombre d'atomes qu'ils contiennent. Cette courbe d'énergie par atome en fonction du volume de la maille cristalline parfaite est appelée équation d'état à 0K ou courbe froide. C'est notamment une des données qui sert à l'ajustement de potentiel, comme celui de Ravelo et al. [54]. Nous présentons la courbe froide du Tantale dans la figure 2.15a. On peut remarquer que l'énergie d'équilibre par atome est le même pour les potentiels Ta1 et Ta2 : $E_c = -8.1\text{eV}$, alors que si nous avons appliqué l'adimensionnement uniquement par le terme de paire, nous aurions eu une valeur différente pour Ta1 et Ta2. Nous noterons l'énergie d'adimensionnement du Tantale E_{Ta} par la suite.

Dans le cas du potentiel modèle, nous devons calculer cette équation d'état à 0K pour chaque valeur de rayon de coupure. Nous reportons quatre de ces courbes pour les valeurs de rayon de coupure [$1,22 r_0$; $1,4 r_0$; $1,6 r_0$; $1,8 r_0$] dans la figure 2.15b. On peut remarquer que le niveau d'énergie et le volume d'équilibre sont les mêmes pour les rayons de coupure $1,22 r_0$ et $1,4 r_0$ à la différence des deux autres. Cela s'explique par le fait que ces rayons de coupure sont inférieurs à $\sqrt{2}r_0$ donc seuls les premiers voisins ont une influence sur le niveau d'énergie à l'équilibre, et chacun de ces premiers voisins est à exactement r_0 de l'atome considéré avec une contribution de $-\epsilon_0$ à l'énergie de l'atome. Le niveau d'énergie d'équilibre E_0 est alors de $-12\epsilon_0$ (un atome de cristal FCC ayant 12 premiers voisins). Nous utiliserons donc

seulement ces deux rayons de coupure dans la suite de thèse pour nos simulations. Pour différencier ces potentiels plus facilement, nous désignerons le potentiel avec une valeur de $r_{cut} = 1,22r_0$ comme le potentiel fragile et celui avec une valeur de $r_{cut} = 1,4r_0$ comme le potentiel ductile. Malgré le fait que ces deux potentiels présentent des comportements fragiles et ductiles, ils le sont plus ou moins l'un par rapport à l'autre.

Pour finir d'adimensionner nos systèmes, il nous reste à définir une masse. La masse d'un atome restant représentative dans nos deux cas, nous gardons cette grandeur pour l'adimensionnement. Ainsi nous pouvons récapituler nos grandeurs d'adimensionnement dans le tableau suivant :

Grandeur	Tantale		Modèle	
longueur	a_{Ta}	3.304 Å	a_0	$\sqrt{2}r_0$
masse	m_{Ta}	180.948 Da	m_0	1
énergie	E_{Ta}	8.1 Ev	E_0	12 ϵ_0
vitesse	$v_{Ta} = \sqrt{\frac{E_{Ta}}{m_{Ta}}}$	2078.2 m/s	$v_0 = \sqrt{\frac{E_0}{m_0}}$	3.464
temps	$t_{Ta} = \frac{a_{Ta}}{v_{Ta}}$	0.1589 ps	$t_0 = \frac{a_0}{v_0}$	0.408

TABLE 2.2 – Table récapitulative des valeurs que nous utiliserons pour adimensionner nos simulations dans le but de les comparer.

Conclusion et perspectives

Dans ce chapitre, nous avons présenté dans un premier temps, le principe de la dynamique moléculaire, les grandes familles de potentiels classiques, le schéma numérique que nous utilisons pour résoudre les équations du mouvement des atomes dans le temps, les principales optimisations algorithmiques utilisées dans le cas de potentiel courte portée, le code que nous utilisons : exaStamp et ses particularités. Nous avons ensuite présenté dans un second temps, les deux potentiels que nous avons implémentés dans exaStamp pour réaliser nos simulations d'écaillage, les raisons qui nous ont poussé à utiliser ces potentiels, et la validation de nos implémentations. Durant l'implémentation du premier potentiel, nous avons observé une divergence entre une des fonctions décrites dans le papier de référence de ce potentiel et la tabulation de cette même fonction fournie par les auteurs de ce papier. L'implémentation du second potentiel nous a conduit, après avoir reproduit des résultats de la littérature, à réaliser une preuve de concept de son applicabilité à des simulations d'écaillage car il n'a encore jamais été utilisé à ces fins. Pour conclure ce chapitre, nous avons présenté le système d'adimensionnement que nous avons établi pour pouvoir comparer les simulations que nous allons réaliser avec ces potentiels.

Les perspectives ouvertes par ce chapitre dans le cadre de cette thèse sont de pouvoir simuler en DM la propagation d'un choc non soutenu, sa réflexion en une seconde onde de détente, la mise en tension par le croisement de ces ondes, et l'endommagement qui peut en découler afin d'étudier le phénomène d'écaillage sur deux matériaux : le Tantale et un modèle dont on peut contrôler la fragilité/ductilité. Ces simulations seront l'objet du chapitre 5. Nous avons évoqué durant la présentation de notre choix du Tantale que les simulations d'écaillage en DM sont jusqu'à présent limitées en taille de simulation, alors qu'a priori la puissance de calcul est de nos jours suffisante pour augmenter la taille de ces simulations d'au moins un ordre de grandeur, et que la taille de simulation est un facteur limitant aux études

d'écaillage en DM. Nous consacrerons donc le chapitre suivant (Chap. 3) à la résolution de ce problème. Enfin, les perspectives plus générales ouvertes par ce chapitre est le possible intérêt d'utiliser le potentiel modèle que nous avons implémenté pour des études phénoménologiques de propagation de fissure, notre preuve de concept ayant aussi démontré que ce potentiel peut faire bifurquer une fissure en de multiple ramifications à petite échelle et donc à moindre coût.

Chapitre 3

Développement d'un algorithme de détection et caractérisation automatique de zones d'intérêts de simulation de dynamique moléculaire

Pour pouvoir étudier l'endommagement en DM de façon plus détaillée que les études déjà publiées [48-53], nous voulons détecter et caractériser les pores au cours du temps sur des simulations de très grandes tailles. Par caractériser, nous entendons déterminer le volume, la position et des facteurs de forme des pores. De précédentes études de ce type ont déjà été réalisées [41, 43, 67-69], cependant ces études ont plusieurs facteurs limitants : la non disponibilité des codes utilisés, une incertitude sur les résultats de mesure de l'algorithme non-quantifiée, une étude non systématique des simulations d'endommagement certainement dû à un coût de calcul prohibitif, et le plus limitant, leurs tailles. La taille de ces simulations est d'une part petite comparée aux capacités des codes de calcul de DM actuels, quelques dizaines de millions d'atomes contre un record à 20 billions¹ d'atomes en 2018 [6] et le premier billion d'atomes atteint dès 2008 [70](REF/FIG). Et d'autre part, ces tailles de simulations sont très éloignées des échelles expérimentales, tout en ne contenant pas suffisamment de statistiques sur les pores pour discriminer différentes distributions de taille. À notre connaissance, l'absence d'étude d'endommagement en DM à plusieurs milliards d'atomes est due à l'absence d'outils d'analyses capables de les traiter. Donc, si nous voulons améliorer les précédentes études et nous rapprocher des échelles expérimentales, nous devons augmenter les tailles des simulations étudiées en fournissant un outil capable de les traiter, avec pour objectif d'atteindre des simulations micrométriques.

Le Tantale étant un cristal cubique centré (2 atomes par maille) et ayant un paramètre de maille de 3,304 angstrom à 300 K et 10^5 Pa, il faut 5,5 milliard d'atomes pour simuler une plaque d'un micron par un micron sur cent nanomètres et 55 milliards d'atomes pour simuler un cube d'un micromètre cube. Le code de DM ExaStamp est capable de simuler cet ordre de grandeur d'atomes avec le potentiel EAM que nous avons implémenté précédemment (Chap. 2). Cependant, ces calculs sont relativement coûteux. Une simulation de 8,3 milliard d'atomes ($1\mu\text{m} \times 1\mu\text{m} \times 150\text{nm}$) impactés à 1500 m/s utilise 12,7 millions d'heures de calcul avec 256 noeuds du calculateur CEA-HF [71, 72] (soit 32 768 coeurs pendant 15,2 jours).

1. billion : 10^{12} \Leftrightarrow trillion sur l'échelle anglo saxonne.

Néanmoins, nous ne pouvons pas raisonnablement stocker ne serait-ce qu'un centième de la simulation, celle-ci représentant 65000 pas de temps et chaque pas de temps pesant 360 GB (compressé avec zlib, niveau 6, 18% de gain) cela représenterait déjà 234 TB de données. Le code d'analyse devra donc s'exécuter *in-situ*, c'est-à-dire, en même temps que notre simulation pour éviter de stocker sur disque l'ensemble des données de la simulation et écrire seulement celles d'intérêt. Le code devra donc aussi être capable de traiter cette quantité massive de données, ce qui implique d'être parallélisé en mémoire distribuée pour pouvoir stocker cette quantité de données en mémoire et la traiter en un temps raisonnable. Ces contraintes de développement étant fortes et le problème d'identification de pores pouvant s'apparenter à un problème d'identification de zones d'intérêts (caractérisées ici par l'absence de matière), nous nous ajoutons la contrainte d'avoir un algorithme généralisable à d'autres type de simulations pouvant elles aussi se généraliser à de l'identification de zones d'intérêts comme les simulations d'éjections de matière (*splashing, micro-jetting, ...*) où l'on cherche à détecter et caractériser des agrégats de matières. Cette contrainte supplémentaire nous permettra de rentabiliser les forts coûts de développement de l'algorithme.

L'algorithme devra donc détecter et caractériser des zones dites d'intérêts à partir de critères physiques définis par l'utilisateur, dans des simulations de DM de plusieurs milliard d'atomes, en parallèle pour pouvoir traiter la quantité de données et de façon *in-situ* pour pouvoir analyser plus d'1% de nos simulations.

3.1 Choix de l'algorithme

Informatiquement pour pouvoir calculer les propriétés d'un objet, il faut pouvoir l'identifier de façon unique c'est-à-dire lui attribuer un identifiant, une étiquette ou encore une "couleur" unique selon la terminologie utilisée dans les différents domaines. Pour définir un objet ayant de multiples composants, une façon de procéder est d'attacher à ses composants une étiquette unique qui identifie l'objet. De nombreux problèmes fondamentaux consistent ainsi à identifier à partir d'une liste d'objets simples des groupes d'objets qui définissent de nouveaux objets plus complexes. Ces classes de problèmes se rattachent aux problèmes de "clustering" dans le domaine du "data-mining", et se rattachent aussi aux problèmes d'étiquetage de graphes. L'algorithme de clustering le plus connu et le plus utilisé est certainement DBSCAN [73, 74]. Il permet de regrouper en cluster des ensembles de points en k-dimensions, sans a priori sur le nombre de clusters, et à partir de 2 paramètres : Eps , la distance entourant un point, et $MinPts$, le nombre de points minimum dans les environs d'un point pour le considérer comme un point central. La complexité moyenne de cet algorithme est de $\Theta(n \log(n))$ et de $\mathcal{O}(n^2)$ dans le pire des cas. DBSCAN est très utile pour traiter les données non structurées que sont les nuages de points et a été amélioré de nombreuses façons pour diminuer son temps d'exécution. Une des techniques les plus communes est d'utiliser une forme de donnée structurée [75] supplémentaire telle qu'une grille de cellule. STING [76] est une des premières illustrations de l'optimisation par utilisation d'une grille de K cellules qui maillent l'espace des points avec $K \ll N$, N étant le nombre de points à clusteriser. STING diminue ainsi la complexité algorithmique à $\mathcal{O}(K)$ mais au détriment de la précision : "[Les régions renvoyées par STING sont une approximation du résultat de DBSCAN.]²", les paramètres de l'algorithme ne dépendent plus que d'un paramètre de densité c définie tel que $c = \frac{MinPts+1}{Eps^2 \cdot \pi}$ et d'un paramètre de granularité pour la grille tel qu'"[à mesure que la granularité tend vers zéro, les régions renvoyées par STING approchent le résultat de

2. The regions returned by STING are an approximation of the result by DBSCAN.

DBSCAN]³”. Utiliser une structure maillant l’espace présente un avantage supplémentaire : une parallélisation efficace de l’algorithme par découpage spatial [77].

Pour ainsi diminuer la complexité algorithmique de DBSCAN en exploitant la structuration des données, ces algorithmes utilisent en fait une autre grande famille d’algorithmes de clustering : les algorithmes d’étiquetage par composantes connexes (ECC)⁴. Ces algorithmes utilisent l’algorithmique de graphe pour établir des clusters de nœuds en déterminant à partir de leurs connectivités une unique racine par cluster et créent ainsi une forêt enracinée, chaque arbre de la forêt étant ainsi un cluster dont la racine est l’étiquette finale. En assimilant un maillage structuré à un graphe où chaque cellule du maillage est un nœud du graphe et en postulant la connectivité (le voisinage de la cellule), on peut ainsi appliquer et spécialiser les algorithmes d’ECC. Pour le cas d’image 2D cette spécialisation a été poussée dès les premiers algorithmes. Cette spécialisation peut-être définie par au moins trois critères que sont : la forme du voisinage considéré, le nombre de passes effectuées, et la façon dont sont gérées les équivalences.

Les algorithmes pionniers d’ECC remontent aux travaux fondateurs de Rosenfeld & Pfaltz en 1966 [78], puis à Shapiro & Haralick en 1981 [79], Lumia et al. en 1983 [80], et finalement Ronse & Devijver en 1984 [81] avec en parallèle de cela Hoshen & Kopelman qui ont dès 1976 appliqué à la physique des algorithmes similaires pour détecter les zones de percolation dans une simulation [82]. Hoshen & Kopelman ne se réfèrent pas directement aux algorithmes d’ECC mais utilisent un algorithme de type *union-find* avec une *connectivité-4* et une seule passe. Ici les termes *union-find* [83], *connectivité-4*, et *une passe* classifient le type d’algorithme d’ECC utilisé. *Union-find* se réfère à la technique utilisée pour étiqueter les pixels de l’image, et nous expliciterons cette technique plus en détails dans la section 3.3. La connectivité définit le voisinage direct de chacun des nœuds du maillage, ici avec une grille régulière 2D une *connectivité-4* signifie que les pixels voisins d’un pixel p sont les pixels qui se situent directement au-dessus et en-dessous de lui, ainsi qu’à sa gauche et sa droite. L’algorithme d’Hoshen-Kopelman n’itère qu’une seule fois sur chaque pixel indépendamment de son contenu, on dit alors qu’il n’effectue qu’une seule passe, c’est un algorithme direct. Il existe deux autres catégories d’ECC définies par leur interaction avec les cellules du maillage : les algorithmes itératifs ou multi-passes qui effectuent plusieurs itérations sur chaque cellule du maillage considéré jusqu’à convergence du résultat, et les algorithmes de type contour tracing qui suivent les cellules en bordure de composante. La plupart des algorithmes d’ECC sont des algorithmes directs (avec un nombre fini de passe) car il garantissent une complexité en $\mathcal{O}(K)$ contrairement aux algorithmes itératifs qui dans le pire des cas auront une complexité en $\mathcal{O}(K^2)$ et aux algorithmes de contour tracing qui ont une complexité minimale de $\mathcal{O}(K)$ mais qui présentent le désavantage de ne pas nécessairement tenir compte des possibles trous au sein d’une composante connexe. L’avantage supplémentaire des algorithmes d’ECC comparé aux algorithmes de type DBSCAN est qu’ils ne sont pas nécessairement fondés sur la densité, c’est-à-dire qu’ils peuvent agréger des régions de l’espace sur des critères supplémentaires et offrent ainsi plus de versatilité.

Le seul pré-requis aux algorithmes d’ECC lorsqu’ils sont appliqués à un maillage est de pouvoir distinguer les mailles à connecter des autres, soit en ayant un maillage qui les distingue déjà, soit en ayant un critère qui permet de les différencier. Les mailles à connecter sont dites de “premier plan” et les autres mailles sont dites de “fond”. Les algorithmes d’ECC étant plus efficaces que les algorithmes de type DBSCAN de par leur complexité moindre, tout

3. As the granularity approaches zero, the regions returned by STING approach the result of DBSCAN.

4. Connected Component Labeling (CCL)

en étant plus versatile, nous choisissons d'utiliser ce type d'algorithme. Néanmoins, ce choix pose deux problèmes. Premièrement, nous devons au préalable générer une grille régulière représentative de notre simulation pour pouvoir appliquer l'ECC, nos données de DM étant un nuage de points. La section suivante (Sec. 3.2) sera donc consacrée à la génération de cette grille régulière. Deuxièmement, pour répondre à nos besoins, notre algorithme d'ECC doit être parallélisé en mémoire distribuée, or nous verrons dans la section 3.3 que pour répondre à ce besoin nous avons dû développer un prototype d'algorithme d'ECC 3D en mémoire distribuée et partagée.

3.2 Projection sur grille régulière

La nécessité d'obtenir un maillage de l'espace représentatif d'une simulation de DM n'est pas nouvelle, et est d'abord apparue pour calculer les grandeurs hydrodynamiques [84] puis pour tester la validité de la description à l'échelle continue des ondes de choc [85]. Nous avons repris le formalisme de Hardy de projection sur grille régulière, et nous allons présenter les principales caractéristiques de ce formalisme. Hardy a introduit la notion de fonction de localisation qui détermine la contribution aux propriétés macroscopiques d'une particule dans l'espace. Cette fonction de localisation $\Delta(\mathbf{r}_i - \mathbf{x})$ est définie telle que devant être strictement positive, maximum en $\mathbf{x} = \mathbf{r}_i$, devenant nulle lorsque $|\mathbf{r}_i - \mathbf{x}| > l/2$, et normalisée tel que :

$$\iiint \Delta(\mathbf{r}_i - \mathbf{x}) d^3x = 1 \quad (3.1)$$

avec \mathbf{r}_i la position de la particule considérée et l la distance de projection. À partir de cette fonction, Hardy définit une "densité" de masse locale $\rho(\mathbf{x})$ et une "densité" de quantité de mouvement locale $\mathbf{p}(\mathbf{x})$ tel que :

$$\rho(\mathbf{x}) = \sum_i m_i \Delta(\mathbf{r}_i - \mathbf{x}) \quad , \quad (3.2)$$

et

$$\mathbf{p}(\mathbf{x}) = \sum_i (m_i \mathbf{v}_i) \Delta(\mathbf{r}_i - \mathbf{x}) \quad , \quad (3.3)$$

m_i étant la masse de la particule i et \mathbf{v}_i sa vitesse avec lesquelles il définit la vitesse locale du système $\mathbf{V}(\mathbf{x})$ comme :

$$\mathbf{V}(\mathbf{x}) = \frac{\mathbf{p}(\mathbf{x})}{\rho(\mathbf{x})} \quad , \quad (3.4)$$

puis avec le théorème d'équipartition de l'énergie il estime la température locale du système $T(\mathbf{x})$ à partir de l'énergie cinétique exprimé dans le référentiel du centre de masse local [86] :

$$T(\mathbf{x}) = \frac{2}{d_l k_b} \frac{\sum_i (\frac{1}{2} m_i |\mathbf{v}_i|^2) \Delta(\mathbf{r}_i - \mathbf{x}) - \frac{1}{2} \rho(\mathbf{x}) |\mathbf{V}(\mathbf{x})|^2}{\sum_i \Delta(\mathbf{r}_i - \mathbf{x})} \quad , \quad (3.5)$$

avec d_l le nombre de degrés de liberté par atome et k_b la constante de Boltzmann.

Ces expressions sont toutes indépendantes du potentiel utilisé mais il introduit ensuite la "densité" d'énergie $E(\mathbf{x})$ comme :

$$E(\mathbf{x}) = \sum_i \left[\frac{1}{2} m_i |\mathbf{v}_i|^2 + \frac{1}{2} \sum_j \phi_{ij} \right] \Delta(\mathbf{r}_i - \mathbf{x}) \quad , \quad (3.6)$$

en supposant que le potentiel d'interaction ϕ_{ij} est un potentiel de paires, dépendant uniquement de la distance \mathbf{r} les séparant. C'est à partir de cette expression qu'il définit les autres grandeurs intensives comme la pression locale du système $P(\mathbf{x})$ ou le flux de chaleur $Q(\mathbf{x})$, qui dépendent donc cette fois du potentiel et avec une forte contrainte sur le type de ce potentiel. Il a par la suite étendu en 2016 [87] ces expressions aux potentiels à plusieurs corps sous la contrainte que les forces qui en dépendent respectent la troisième loi de Newton, condition automatiquement remplie lorsqu'ils dépendent uniquement de la distance inter-particules \mathbf{r}_{ij} (comme les potentiels EAM). À partir de ces expressions, on peut calculer les propriétés locales dans un maillage en introduisant la notation suivante :

$$\omega_{ik} = \iiint_{V_k} \Delta(\mathbf{r}_i - \mathbf{x}) dx^3 , \quad (3.7)$$

ω_{ik} la contribution de la particule i à la cellule k dont la normalisation (Eq. 3.1) garantie que :

$$\sum_{k=1}^{n_{\text{voxel}}} \omega_{ik} = 1 . \quad (3.8)$$

On propose de généraliser le formalisme de Hardy à n'importe quelle propriétés extensives f_i , et intensives g_i définies pour chaque particule i , indépendamment de la forme du potentiel, en utilisant les 2 formules suivantes :

$$\tilde{f}_k = \sum_{i=1}^{N_{\text{atoms}}} f_i \omega_{ik} , \quad (3.9)$$

$$\tilde{g}_k = \frac{\sum_{i=1}^{N_{\text{atoms}}} g_i \omega_{ik}}{\sum_i \omega_{ik}} . \quad (3.10)$$

On peut ainsi réécrire les équations (3.2), (3.3), (3.4), (3.5) telles que :

$$\tilde{m}_k = \sum_{i=1}^{N_{\text{atoms}}} m_i \omega_{ik} \quad (3.11)$$

$$\tilde{\mathbf{p}}_k = \sum_{i=1}^{N_{\text{atoms}}} m_i \mathbf{v}_i \omega_{ik} \quad (3.12)$$

$$\tilde{\mathbf{v}}_k = \frac{\tilde{\mathbf{p}}_k}{\tilde{m}_k} = \frac{\sum_{i=1}^{N_{\text{atoms}}} m_i \mathbf{v}_i \omega_{ik}}{\sum_{i=1}^{N_{\text{atoms}}} m_i \omega_{ik}} \quad (3.13)$$

$$\Leftrightarrow \tilde{\mathbf{v}}_k = \frac{\sum_{i=1}^{N_{\text{atoms}}} \mathbf{v}_i \omega_{ik}}{\sum_i \omega_{ik}} , \text{ si les particules ont la même masse}$$

et

$$\tilde{T}_k = \frac{1}{d_l k_b} \frac{\sum_i^{N_{\text{atoms}}} m_i |\mathbf{v}_i|^2 \omega_{ik} - \tilde{m}_k |\tilde{\mathbf{v}}_k|^2}{\sum_i \omega_{ik}} \quad (3.14)$$

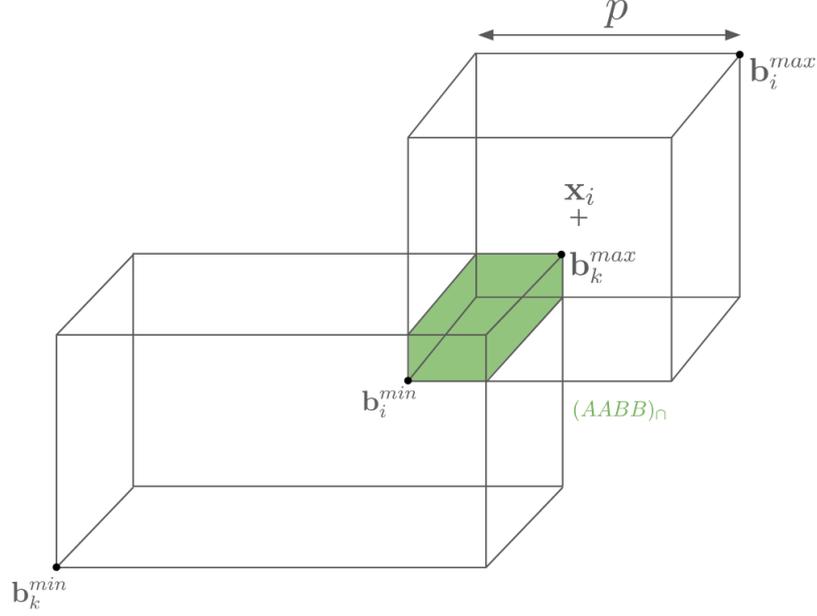


FIGURE 3.1 – Exemple d'intersection d'*Axis Align Bounding Box* (AABB), une étant centrée sur un atome de position \mathbf{x}_i et l'autre représentant une cellule de la grille régulière.

et définir la pression locale indépendamment de la forme du potentiel utilisé par le théorème du viriel, où on remarque une contribution potentielle et une contribution cinétique :

$$\tilde{P}_k = \frac{1}{3} \left[\sum_i^{N_{\text{atoms}}} Tr \left(\sum_{j \neq i} \mathbf{F}_i \cdot \mathbf{r}_{ij}^T \right) \omega_{ik} + \sum_i^{N_{\text{atoms}}} m_i |\mathbf{v}_i|^2 \omega_{ik} - \tilde{m}_k |\tilde{\mathbf{v}}_k|^2 \right] \frac{1}{V_k} \quad (3.15)$$

Le Tantale et notre matériau modèle (Chap. 2) étant tous les deux des cristaux cubiques (cubique centré et cubique face centré), nous avons décidé d'utiliser comme fonction de localisation une fonction porte Π de largeur p et de fixer p à la valeur du paramètre de maille du cristal. Le calcul de la contribution ω_{ij} s'écrit alors :

$$\omega_{ik} = \iiint_{V_k} \Pi \left(\frac{\mathbf{x} - \mathbf{x}_i}{p} \right) dx^3 \quad (3.16)$$

L'utilisation d'autres fonctions de localisation de forme sphérique ou gaussienne est envisageable. Cependant le calcul d'intersection avec de telles fonctions est non trivial [88]. N'ayant pas de solution analytique [89], le calcul de l'intégrale peut nécessiter une phase de renormalisation numérique supplémentaire. L'avantage d'utiliser cette fonction avec un maillage 3D de pavé régulier est de pouvoir calculer cette intégrale à un très faible coût en utilisant des *Axis Align Bounding Box* (AABB).

On définit l'AABB du pavé régulier k à partir de 2 points \mathbf{b}_k^{min} et \mathbf{b}_k^{max} qui le délimitent (Fig. 3.1) et l'AABB de l'atome i projeté par les points $\mathbf{b}_i^{min} = \mathbf{x}_i - \frac{p}{2}$ et $\mathbf{b}_i^{max} = \mathbf{x}_i + \frac{p}{2}$. L'intersection de ces 2 AABB est alors :

$$(AABB)_\cap = (AABB)_k \cap (AABB)_i = \left\{ \begin{array}{l} \mathbf{b}_\cap^{min} = \max(\mathbf{b}_k^{min}, \mathbf{b}_i^{min}) \\ \mathbf{b}_\cap^{max} = \min(\mathbf{b}_k^{max}, \mathbf{b}_i^{max}) \end{array} \right\}$$

de masse, on réalise 1 multiplication supplémentaire (Eq. 3.11) par intersection. Pour calculer la vitesse, on doit calculer la quantité de mouvement : 3 multiplications de plus avec la contribution de masse par intersection, qui sera ensuite divisée par la masse des cellules (Eq. 3.13). Pour calculer la température, on ajoute 3 multiplications, 2 additions, 1 racine carrée, et 1 multiplication pour calculer le terme $|\mathbf{v}_i|^2$, puis 1 multiplication avec la contribution de masse par intersection pour obtenir le terme cinétique, auquel on retire le terme cinétique du centre de masse, calculé par cellule avec 3 multiplications, 2 additions, 1 racine carrée, 1 multiplication, et la soustraction au terme précédent. Et le calcul de la température se finit : par 1 division de la somme des contributions et 1 multiplication par le terme $1/(3k_b)$ (Eq. 3.14).

Au total pour projeter masse, vitesse et température dans la grille, nous aurons donc :

- $nb_{atoms} \times 6 + nb_{atoms} \times N \times (3 + 3 + 9 + 1 + 1 + 3 + 1 + 2) + nb_{cells} \times 1 = 6 nb_{atoms} + 23 nb_{atoms} \times N + nb_{cells}$ additions
- $nb_{atoms} \times N \times (3 + 1 + 1 + 3 + 3 + 1 + 1) + nb_{cells} \times 1 = 13 nb_{atoms} \times N + nb_{cells}$ multiplications
- $nb_{cells} \times (1 + 1) = 2 nb_{cells}$ divisions
- $nb_{atoms} \times N + nb_{cells}$ racines carrées
- $9 nb_{atoms} \times N$ comparaisons

Si on considère un système de 100 M d'atomes dans 50 M de voxels, avec une taille de projection p de l'ordre de la taille des dimensions de nos cellules, donc avec $N = 27$: on effectue 62,75 milliards additions, 35,15 milliards multiplications, 100 millions divisions, 2,75 milliards racines carrées, et 24,3 milliards comparaisons.

Une fois la projection réalisée, pour visualiser les résultats, en adoptant le format de fichier VTK [90] utilisé par Paraview [91, 92] chaque MPI peut écrire un fichier XML `.vti` (le format pour les images) représentant son domaine et ces fichiers sont groupés par un fichier `.pvti` ouvrable par Paraview. La figure 3.2 a été réalisée par ce moyen et représente le champ de densité dans une coupe 2D d'une plaque de Tantale en train d'écailler. Les zones bleues de cette image ($\rho = 0$) sont les pores que nous souhaitons détecter et caractériser avec l'algorithme d'ECC. On peut aussi observer un résultat intéressant qui est qu'avec cette résolution de grille plus fine que la maille du cristal, les plans atomiques sont visibles tout en gardant une continuité de la matière grâce à la fonction de projection.

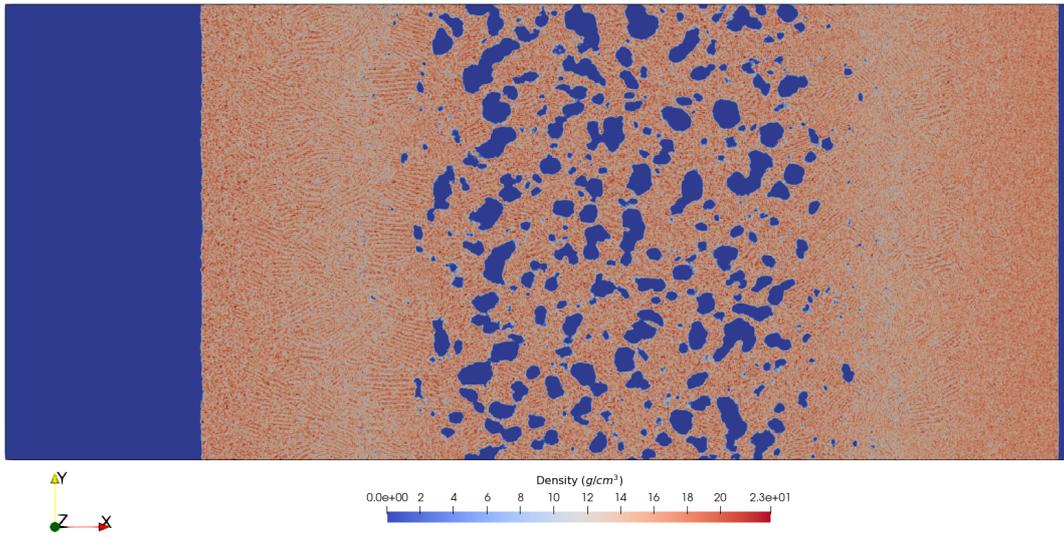


FIGURE 3.2 – Coupe 2D d’une simulation d’écaillage de Tantale projetée sur grille régulière. Le champ de densité est représenté.

3.3 Prototype d’algorithme d’ECC 3D en mémoire distribuée et partagée

Maintenant que nous avons choisi la famille d’algorithmes à utiliser pour résoudre notre problème, les algorithmes d’ECC, et que nous remplissons les critères pour pouvoir les utiliser, en nous étant doté d’une grille régulière 3D représentative de notre simulation, nous devons choisir en particulier l’algorithme à utiliser.

He et al. ont publié en 2017 [93] un état de l’art des algorithmes d’ECC. Leur état de l’art se concentre uniquement sur les algorithmes dédiés aux images avec une *connexité-8* en 2D (la plus commune et la plus complexe). Ils justifient que les algorithmes à deux passes sont les plus efficaces et présentent les différentes optimisations séquentielles développées dans ce cadre car ces algorithmes sont la base aussi bien des algorithmes pour les images 3D et nD, et des algorithmes parallèles. On distingue deux principaux types d’optimisations, les optimisations fondées sur le scan du voisinage d’un pixel ou d’un groupe de pixel et les optimisations sur la stratégie de gestion des étiquettes. Nous ne rentrerons pas dans le détail de chaque optimisation car les optimisations sur le scan du voisinage sont toutes fondées sur la façon dont est parcourue l’image et nous verrons plus tard dans cette section que notre image n’est pas parcourue dans ce sens. Nous présenterons seulement une seule méthode de gestion d’étiquette, celle-ci étant la seule option raisonnable pour les algorithmes parallèles d’après les conclusions de Cabaret et al. [94].

La conclusion de la review de He et al. est qu’il faut implémenter les optimisations des images 2D aux images 3D/nD, et qu’il faut encore lever de nombreux verrous pour les algorithmes parallèles, notamment pour les opérations de résolution des équivalences qui sont séquentielles, et le traitement des frontières. La plupart des algorithmes d’ECC parallèle ont commencé à être publiés à cette période-ci avec notamment : le papier [94] sur lequel nous fondons notre choix de stratégie pour la gestion des équivalences, et l’apparition du projet YACCLAB [95, 96] qui implémente et benchmark les différents algorithmes d’ECC publiés dans la littérature. Ce projet est disponible sur Github [97] sous forme de bibliothèque mais nous n’avons pu utiliser aucun de ces algorithmes pour plusieurs raisons.

Premièrement, les versions 3D des algorithmes ne sont pas toujours disponibles ou n’étaient pas publiées au moment de notre implémentation [98]. Deuxièmement, toutes les versions parallèles de ces algorithmes sont des versions parallélisées en mémoire partagée, or nous devons aussi et surtout avoir un algorithme parallélisé en mémoire distribuée. On peut expliquer l’absence d’algorithme parallèle en mémoire distribuée par le fait que les machines ont depuis longtemps largement assez de mémoire pour stocker une image même volumineuse. Le besoin de résoudre le problème d’ECC pour des images en mémoire distribuée étant absent, les algorithmes n’ont pas été développés. Les algorithmes récents d’ECC parallélisé en mémoire distribuée ne sont pas destinés à faire du clustering d’images [99, 100] mais directement du clustering de graphe (type réseau de communication, ou réseau social) ces algorithmes sont donc d’une tout autre catégorie en étant bien souvent multipasses donc avec une complexité $\mathcal{O}(n^2)$. Troisièmement, pour implémenter l’algorithme dans exaStamp nous devons utiliser la structure de donnée `grid_cell_values` qui contient la grille régulière de projection. Cette structure de donnée était déjà présente dans le code pour une autre application (calculs de TTM⁶ qui couplent la DM avec une deuxième température, celle des électrons, stockée dans une grille régulière). Or cette structure de données ne stocke pas ses voxels dans le sens direct de parcours spatial (*raster scan*). Ses voxels sont eux aussi des sous

6. Two Temperature Model

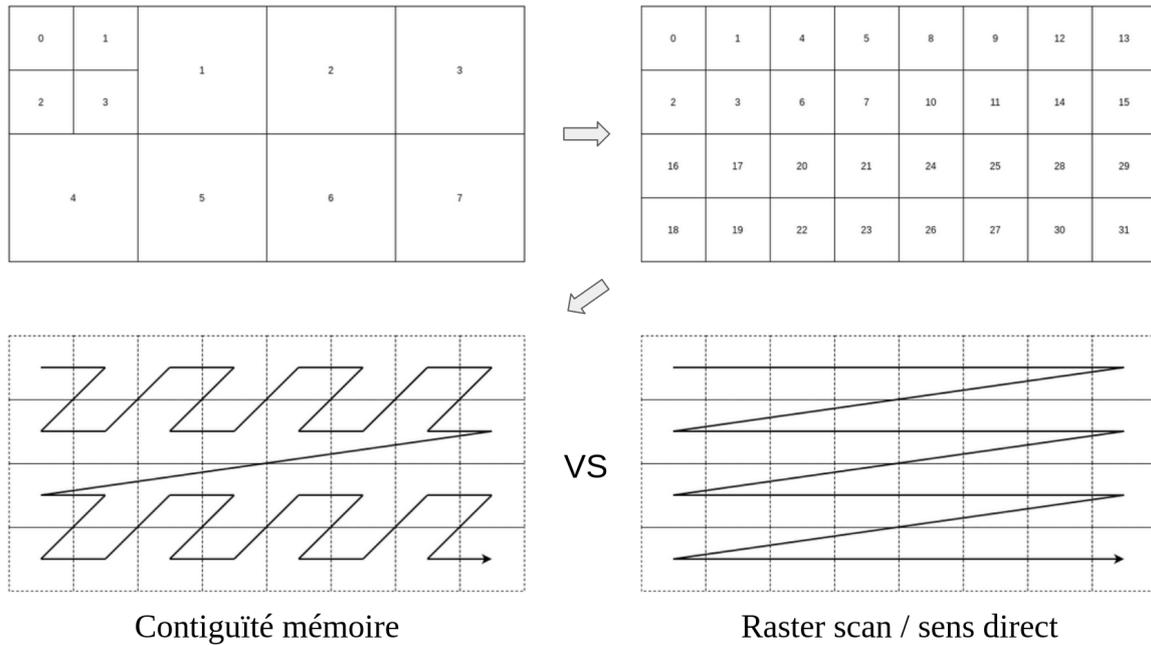


FIGURE 3.3 – Illustration 2D d'un découpage des cellules de Verlet en 2 selon chaque direction de l'espace et le parcours mémoire contigu engendré, comparé à un parcours en raster scan.

cellules des cellules de Verlet et sont stockés en *raster scan* par cellule, cellules qui sont elles même stockées en sens direct. On obtient ainsi le parcours mémoire représenté dans la figure 3.3.

Ce parcours mémoire ne permet pas d'appliquer les optimisations de scan de voisin en ECC. Il est fait pour simplifier les algorithmes de communications de cellules fantômes et d'équilibrage de charge qui, lorsqu'ils ont besoin d'envoyer une cellule de données, n'ont besoin que du pointeur mémoire vers le début de la cellule et de sa quantité de données $nb_{sub} \times nb_{champs}$, les champs de données étant eux aussi stockés par cellules (Fig. 3.4). Cette structure de donnée s'apparente à une structure de type *AoS*⁷. Dans notre cas, la structure est entièrement composée de champs de type `double` et le tableau est contigu en mémoire en étant alloué sous forme de *memory pool*.

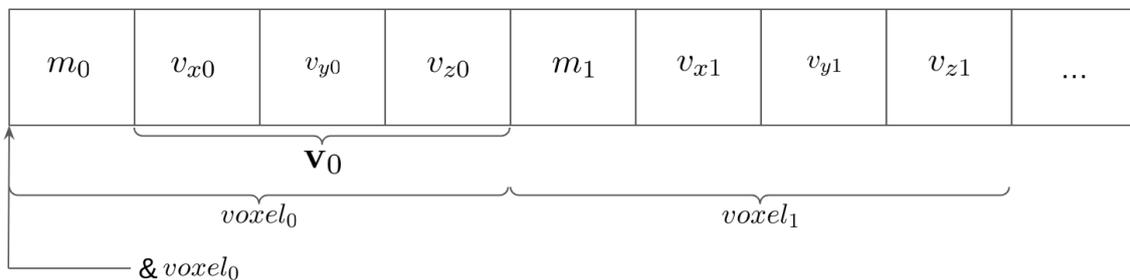


FIGURE 3.4 – Exemple d'organisation mémoire de la structure `grid_cell_values` n'ayant pour champs que la masse et la vitesse des voxels. Le `&` symbolise l'adresse mémoire du tableau.

7. *Array of Structures*

Nous avons donc implémenté notre propre algorithme d'ECC 3D parallèle en mémoire partagée et distribuée en nous appuyant sur les conclusions des travaux précédents lorsqu'elles étaient applicables à notre cas. Notre algorithme est un algorithme à 2 passes avec une *connexité-26* (l'extension de la *connexité-8* en 2D) et une gestion des équivalences avec une stratégie *union-find*. Nous parcourons notre image dans le sens de la contiguïté mémoire, ce parcours n'étant pas en *raster scan*, on ne peut pas avoir d'a priori sur les voisins ayant déjà été labélisés, donc nous ne réalisons aucune optimisation sur le masque de scan, chaque pixel scanne l'ensemble de son voisinage (les 26 voisins directs). Adopter le sens de parcours *raster scan* classique malgré notre structure mémoire serait possible mais engendrerait un surcoût algorithmique important pour calculer l'adresse mémoire des voisins dans le masque de scan, celle-ci dépendant de la position au sein d'une cellule, du nombre de subdivisions considérées et de la position de la cellule dans l'espace. De plus les algorithmes d'ECC étant surtout *memory bound*⁸ et en particulier les algorithmes scannant par voxel [94], nous avons estimé que le gain engendré par une diminution du nombre d'accès mémoire serait insuffisant pour compenser l'augmentation du temps d'accès à la mémoire et du coût de calcul supplémentaire.

La stratégie *union-find* est une stratégie qui consiste à utiliser une table d'équivalence avec une relation d'ordre partielle pour déterminer si des étiquettes sont équivalentes ou non et tire son nom des deux principales fonctions interagissant avec cette table : **union** et **find**. La table d'équivalence T est une table pour laquelle chaque entrée e a une sortie et dont chaque sortie est nécessairement une entrée de la table. Cette table est un graphe orienté. Chaque entrée est un nœud de ce graphe et la sortie associée est le nœud vers lequel il pointe. Un nœud pointe toujours vers un nœud ayant un identifiant plus faible que le sien ou vers soi-même ($T[e] \leq e$). Lorsqu'un nœud pointe vers lui-même on dit alors que c'est une racine. L'ensemble des nœuds pointant directement ou indirectement vers une même racine forment un arbre et l'ensemble des arbres de la table forme ainsi une forêt enracinée. La fonction **find** est une fonction qui prend en entrée une étiquette et renvoie la racine associée à cette étiquette (Alg. 1). La fonction **union** est une fonction qui prend en entrée deux étiquettes de la table d'équivalence et qui va unir les deux arbres associés à ces étiquettes en faisant pointer la racine de l'une vers l'autre (Alg. 2).

Algorithm 1 Algorithme *find*

Require: e une étiquette, T une table d'équivalence.

```

1: procedure FIND( $e, T$ )
2:    $r_e \leftarrow T[e]$ 
3:   while  $r_e \neq T[r_e]$  do
4:      $r_e \leftarrow T[r_e]$ 
5:   end while
6:   return  $r_e$  la racine de l'étiquette  $e$  dans la table  $T$ .
7: end procedure

```

La première passe de l'étiquetage par composante connexe consiste à itérer sur chaque voxel du domaine, attribuer des étiquettes temporaires aux voxels dits de premier plan (les voxels à connecter) et établir la table d'équivalence qui lie les étiquettes temporaires

8. *memory bound* : qualificatif pour les codes/algorithmes dont la vitesse d'exécution est limitée par la vitesse à laquelle la machine accède à la mémoire, par opposition aux codes *compute bound*, qui sont principalement limités par la vitesse à laquelle la machine exécute un calcul

Algorithm 2 Algorithme *union*

Require: e_1 , et e_2 deux étiquettes à unir, T une table d'équivalence.

```
1: procédure UNION( $e_1, e_2, T$ )
2:    $r_1 \leftarrow$  FIND( $e_1, T$ )
3:    $r_2 \leftarrow$  FIND( $e_2, T$ )
4:   if  $r_1 < r_2$  then
5:      $T[r_2] = r_1$ 
6:   else
7:      $T[r_1] = r_2$ 
8:   end if
9: end procédure
```

par une relation d'ordre partiel vers leur étiquette définitive. Dans notre cas, les données de notre grille régulière sont des champs réels et un filtre de données doit être appliqué pour sélectionner les voxels de premier plan. Nous appliquons ce filtre à la volée durant la première passe sous forme d'une condition C à remplir pour être un voxel de premier plan et éviter de faire une passe supplémentaire qui applique seulement ce filtre. La condition C est implémentée sous forme d'une lambda fonction qui prend en entrée l'identifiant d'une cellule, capture les pointeurs vers les champs dont elle a besoin et renvoie vrai seulement si la cellule est considérée de premier plan. Ce choix a été fait pour laisser la flexibilité à l'utilisateur de pouvoir choisir parmi les différents filtres celui qu'il veut appliquer ou implémenter facilement un nouveau filtre adapté à ses besoins.

Dans le cadre de cette thèse, les principales conditions que nous utiliserons pour déterminer si un voxel est de premier plan seront des conditions de densité. Nous considérons qu'une cellule est vide si sa densité $\rho_k < \rho_{seuil}$, cependant, il n'est pas nécessaire de calculer la densité de la cellule pour vérifier ce critère. Le volume des voxels étant constant on peut définir un critère de masse seuil tel que $m_{seuil} = \rho_{seuil}V_k$ et simplement comparer la masse d'un voxel à ce nouveau critère pour économiser K divisions (une division par voxel). On peut aussi définir le critère inverse à partir de la négation de cette condition tel qu'un voxel de matière est un voxel dont $m_k > m_{seuil}$. Pour détecter la surface de la matière ou d'un pore on peut aussi définir le critère tel que $(m_k > m_{seuil}) \wedge \exists(m_{voisin \in k} < m_{seuil})$.

La première passe de notre étiquetage par composante connexe itère sur chaque voxel du domaine et teste si elle remplit la condition définie au préalable pour qu'il soit considéré comme un voxel de premier plan. Si ce voxel est de premier plan, alors on scanne son voisinage pour savoir s'il contient des voxels de premier plan. S'il n'y a pas de voxel de premier plan, alors on introduit une nouvelle étiquette dans la table d'équivalence qui pointe vers elle-même. S'il n'y a qu'un voxel de premier plan alors on affecte à notre voxel l'étiquette racine de notre voisin. S'il y a plus d'un voxel voisin de premier plan alors on affecte à notre voxel l'étiquette racine la plus petite de notre voisinage et l'on unit les racines voisines (Alg. 3).

On peut définir aussi une autre fonction optionnelle de fermeture transitive, ici appelée **reduce**, qui fait pointer l'ensemble des nœuds de la table d'équivalence vers leur racine (Alg. 4). Elle permet de réduire le chemin que doit parcourir une étiquette dans le graphe pour trouver sa racine en l'y reliant directement. Algorithmiquement cette fonction n'est pas nécessaire pour réaliser l'étiquetage par composante connexe mais est utile pour réduire le temps d'exécution de la deuxième passe d'étiquetage qui attribue les étiquettes définitives aux voxels de l'image (Alg. 5). En effet, en ayant recours à la fonction **reduce** avant la deuxième

Algorithm 3 Algorithme 1^{ere} passe d'ECC

Require: G une grille régulière, C une condition qui retourne si un pixel est de premier plan et T une table d'équivalence. On note i l'indice d'un voxel.

```
1: procedure 1ERE_PASSE( $G, C, T$ )
2:   for all  $v_i \in G$  do
3:     if  $C(v_i) = True$  then
4:        $r_{tmp} \leftarrow nb_{voxels} + 2$  (valeur strictement supérieur à la plus grande
        étiquette possible.)
5:       for all  $v_n \in voisinage_i$  do
6:         if  $(v_n \neq 0) \wedge (FIND(v_n, T) < r_{tmp})$  then
7:            $r_{tmp} \leftarrow FIND(v_n, T)$ 
8:         end if
9:       end for
10:      if  $r_{tmp} = nb_{voxels} + 2$  then
11:         $v_i \leftarrow i + 1$ 
12:         $T[i + 1] \leftarrow i + 1$ 
13:      else
14:         $v_i \leftarrow r_{tmp}$ 
15:        for all  $(v_n \neq 0) \in voisinage_i$  do
16:           $UNION(r_{tmp}, v_n, T)$ 
17:        end for
18:      end if
19:    end if
20:  end for
21: end procedure
```

passee, chaque cellule peut simplement remplacer son étiquette temporaire par l'étiquette associée à celle-ci dans la table plutôt que d'appeler systématiquement la fonction **find**.

Algorithm 4 Algorithme de fermeture transitive par la fonction *reduce*

Require: T une table d'équivalence.

```

1:
2: procedure REDUCE( $T$ )
3:   for all  $e \in T$  do
4:      $T[e] \leftarrow \text{FIND}(e, T)$ 
5:   end for
6: end procedure

```

Algorithm 5 Algorithme 2^{eme} passe d'ECC

Require: G une grille régulière étiquetée par la première passe, et T la table d'équivalence associée.

```

1: procedure 2EME_PASSE( $G, T$ )
2:   REDUCE( $T$ )
3:   for all  $v_i \in G$  do
4:      $v_i \leftarrow T[v_i]$ 
5:   end for
6: end procedure

```

Pour rendre notre algorithme parallèle en mémoire partagée, chaque thread au sein d'un processus MPI effectue dans un premier temps la première passe d'ECC sur son bloc de travail et possède sa propre table d'équivalence puis exécute la fonction **reduce**. Les blocs de travail de voxel sont calculés à l'aide d'un algorithme de partitionnement spatial RCB (Recursive Coordinate Bisection) [37]. Ensuite un thread est chargé de connecter tous les blocs en itérant sur les voxels de 3 des 6 faces de chaque bloc avec comme voisinage les 9 voxels du bloc mitoyen et effectue un **union** dans chacune des tables d'équivalences des blocs lorsqu'il rencontre deux voxels de premier plan à connecter (Fig. 3.5a). Pour connecter les différents domaines MPI, un thread dans chaque MPI itère sur les voxels des 6 faces du domaine et lorsque un voxel est de premier plan, il liste ses voisins situés dans la couche de voxels fantômes qui sont aussi de premier plan et écrit les triptyques de données constitués de l'identifiant du voxel, l'identifiant du voxel voisin et l'étiquette racine associée au voxel (Alg. 6) et (Fig. 3.6a-3.6b). Il faut cependant durcir la condition sur l'épaisseur de la couche de cellules fantômes pour garantir d'identifier correctement les voxels de premier plan dans la couche fantôme : $e_{gl} > \frac{p}{2} + s$ (Eq. 3.18).

Algorithm 6 Algorithme de création des messages envoyés au MPI principal

Require: G une grille régulière étiquetée par la première passe, C la condition qui retourne si un pixel est de premier plan, T la table d'équivalence associée à G , et M la table contenant le message à envoyer. On désigne l'ensemble des voxels à la frontière de G par Frt_G et on note les voxels fantômes g .

```
1: procedure CREATION_MSG( $G, C, T$ )
2:   for all  $v_i \in Frt_G$  do
3:     if  $v_i \in T$  then
4:       for all  $g_i \in voisinage_i$  do
5:         if  $C(g_i) = True$  then
6:            $M(v_i, g_i) \leftarrow T(v_i)$ 
7:         end if
8:       end for
9:     end if
10:  end for
11: end procedure
```

Algorithm 7 Algorithme de traitement des messages reçu par le MPI principal

Require: M l'ensemble des messages contenant les triptyques de données envoyés par les MPI secondaires, J la table de jointure, et H la table de hachage qui sert à stocker les messages en attente de connexion. On note un triptyque trp et son contenu (v, g, e) .

```
1: procedure TRAITEMENT_MSG( $M, J, H$ )
2:   for all  $trp_i \in M$  do
3:     if  $e_i \in J$  then
4:        $e_i \leftarrow \text{FIND}(e_i, J)$ 
5:     else
6:        $J(e_i) \leftarrow e_i$ 
7:     end if
8:     if  $g_i \in H$  then
9:        $e_g \leftarrow H(g_i) \text{ UNION}(e_i, e_g, J)$ 
10:       $H(g_i), H(v_i) \leftarrow \text{FIND}(e_g, J)$ 
11:    else
12:       $H(v_i) \leftarrow e_i$ 
13:    end if
14:  end for
15:  REDUCE( $J$ )
16: end procedure
```

Ensuite chaque MPI envoie un premier message à un MPI maître contenant le nombre de triptyques qu'il va envoyer dans le message suivant. Le MPI maître peut alors allouer l'espace nécessaire pour recevoir l'ensemble des triptyques. Une fois les triptyques reçus, il dépile les messages en itérant sur les triptyques et utilise deux tables de hachage⁹ pour effectuer la

9. Structure de données qui stocke des valeurs en les associant à des clés uniques et qui permet en

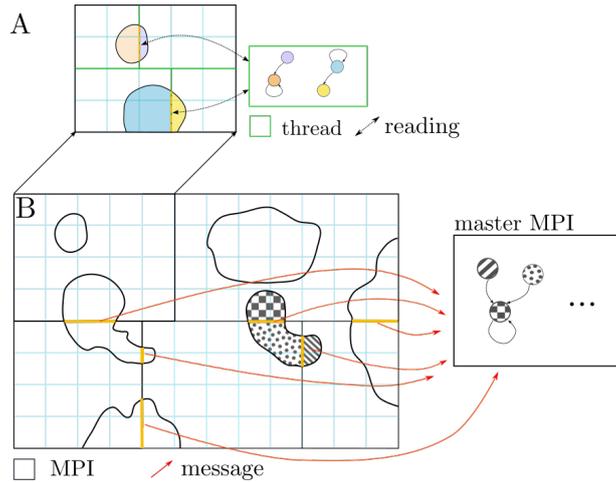


FIGURE 3.5 – Illustration 2D de la connexion entre les différentes régions parallèles : **A** entre les blocs de travail des threads, **B** entre les différents domaines MPI.

connexion par composante connexe (Alg. 7) et (Fig. 3.6c). Pour nos tables de hachage, nous utiliserons systématiquement des `unordered_map`, le conteneur natif du C++ implémentant les tables de hachage, et nous utiliserons leur méthode `reserve` avec le nombre maximum d'éléments qu'elles pourront recevoir pour éviter un changement de fonction de hachage et la reconstruction des tables durant leur utilisation.

Une des tables stocke les identifiants de voxel comme clés et leur associe une étiquette comme valeur. La deuxième table est une table d'équivalence qui sert à résoudre la connexion entre composantes connexes, nous l'appellerons table de jointure pour la différencier des tables d'équivalence des threads. Quand il traite un triptyque, le thread recherche d'abord dans la table de jointure si l'étiquette du triptyque est dans la table. Si cette étiquette est dans la table de jointure, alors il récupère sa racine avec un `find`. Si cette étiquette n'est pas dans la table de jointure, il l'insère en tant que racine. Ensuite il regarde dans la table de hachage si elle contient l'identifiant du voxel fantôme du triptyque. Si la table de hachage ne contient pas cet identifiant alors il insère l'identifiant du voxel non fantôme et sa racine dans la table de hachage et passe au triptyque suivant. Si la table de hachage contient l'identifiant du voxel fantôme alors il récupère l'étiquette racine associée et effectue une `union` entre la racine du fantôme et celle du voxel non fantôme dans la table de jointure. Une fois l'`union` effectuée, l'étiquette racine associée aux identifiants des 2 voxels dans la table de hachage sont aussi mis à jour avec la nouvelle racine. Une fois tous les messages traités, le thread exécute la fonction `reduce` sur la table de jointure. Il écrit enfin le message destiné à chaque MPI à partir du message initial qu'il a reçu. Le message sera de la taille du nombre de triptyques et contiendra seulement la racine associée à l'étiquette initialement présente. Les messages sont renvoyés aux MPI secondaires. Chacun de leur thread lit alors le message, insère dans sa table d'équivalence les nouvelles racines lorsque nécessaire (Fig. 3.6d). Ils n'ont alors plus qu'à de nouveau exécuter la fonction `reduce` sur leur table d'équivalence et exécuter la deuxième passe d'ECC sur leurs cellules. Nous résumons notre algorithme dans la figure 3.7. Les performances de cet algorithme seront discutées dans la section 3.5.

moyenne d'accéder, insérer, et supprimer un élément, en un temps constant ($\Theta(1)$). Dans le pire des cas, la complexité de ces opérations passe à $\mathcal{O}(n)$.

MPI n	...		3	3	3			18	18		e_{3D}	
ghost layer	...	0	1	1	0	1	1	0	0	1	1	C_{gid}
ghost layer	...	0	0	1	1	1	0	0	1	1	0	C_{gid}
MPI m	...		2	2	10	10				43	43	e_{3D}

message of MPI n	35	24	3	36	24	3	36	25	3	38	25	3
	38	26	3	39	26	3	42	29	18	42	30	18
	43	30	18	...	gid	vid	e_{3D}					
message of MPI m	24	35	2	24	36	2	25	36	2	25	38	10
	26	38	10	26	39	10	29	42	43	30	42	43
	30	43	43	...	gid	vid	e_{3D}					

message to MPI n and m	2	2	2	2
	2	2	18	18
	18	...		

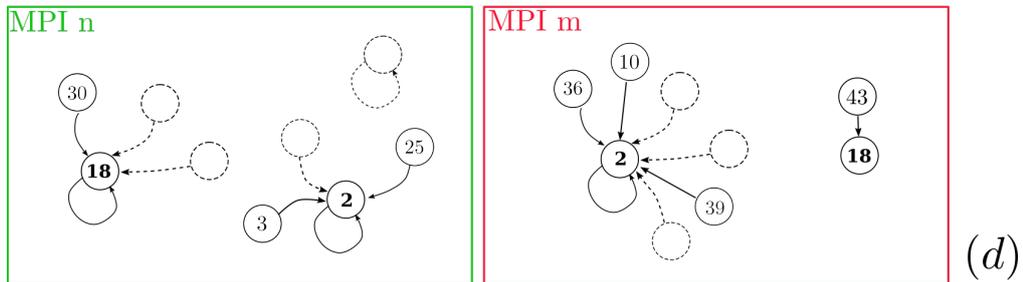


FIGURE 3.6 – (a) Exemple d’étiquetage 3D à la frontière d’un MPI n et m ayant une frontière commune après la première passe. (b) Messages associés à l’exemple (a), réalisés avec l’algorithme 6 et envoyés au MPI principal par les MPI n et m . (c) Message envoyé par le MPI principal en réponse après qu’il ait appliqué l’algorithme 7. (d) Chaque thread de chaque MPI insère dans sa table d’équivalence les racines définitives de ses étiquettes à partir du message qu’il a reçu en (c).

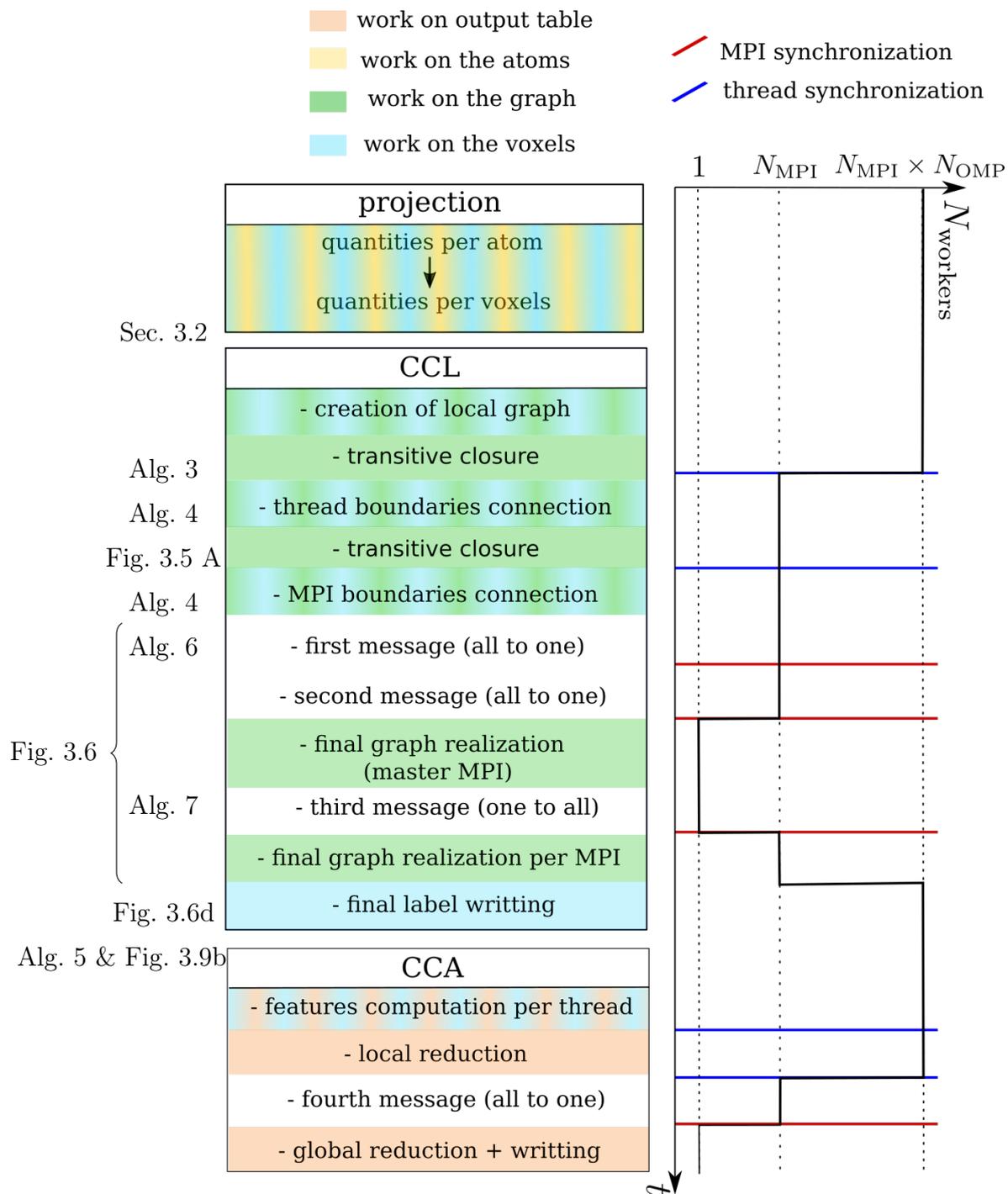


FIGURE 3.7 – Schéma représentant l’algorithme de la projection à l’analyse par composantes connexes. Les couleurs représentent les structures de données avec lesquelles l’algorithme interagit. Les synchronisations induites par l’algorithme sont représentées par des lignes horizontales de couleurs différentes seront leur nature. Un diagramme de Gantt idéalisé (un équilibrage de charge parfait entre les travailleurs est considéré) est associé à la droite du diagramme de l’algorithme.

Si nous reprenons maintenant le cas de la figure 3.2, nous pouvons appliquer notre algorithme avec une condition de densité $\rho = 0$ pour sélectionner les voxels à connecter. Nous obtenons ainsi la figure 3.8 entièrement étiquetée.

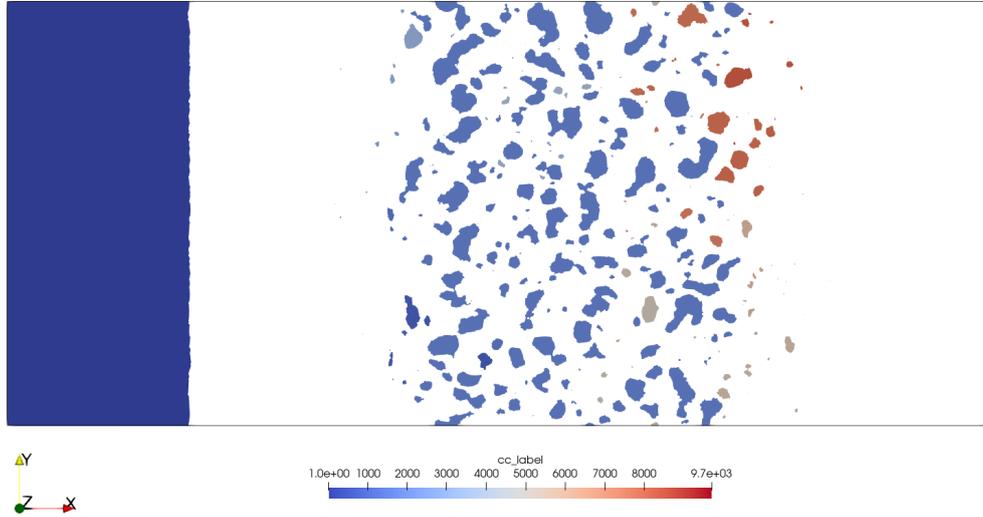
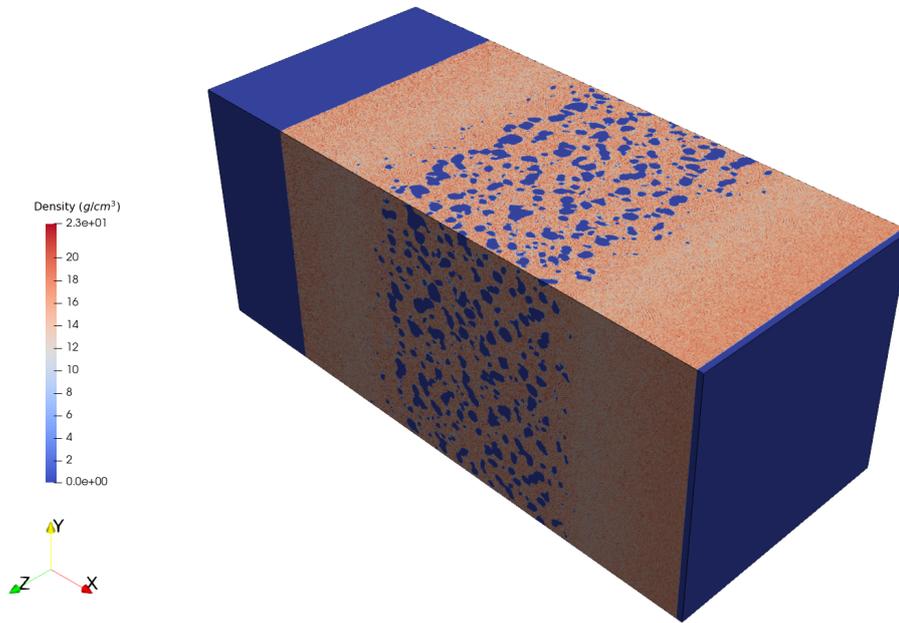
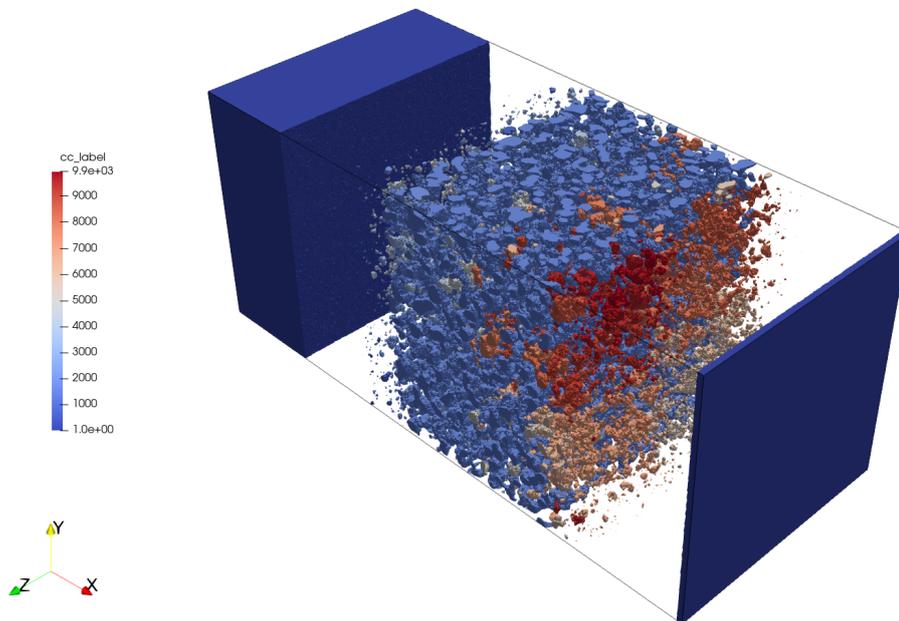


FIGURE 3.8 – Coupe 2D de la simulation d'écaillage de Tantale projetée sur grille régulière illustrée en figure 3.2 et étiquetée par notre algorithme.

Cette vue 2D de la simulation peut donner une impression d'un mauvais étiquetage, plusieurs clusters de voxels ayant la même couleur mais n'étant pas connectés dans cette vue. Une vue 3D comparant le champs de densité et l'étiquetage de cette simulation est présentée en figure 3.9 et illustre la connexion 3D de ces clusters due au réseau de pores en percolation. Les grands volumes de vides de par et d'autres de la simulation selon l'axe des x ne sont bien évidemment pas des pores. Le volume le plus à gauche a été introduit par la compression du système avec un piston pour produire le choc responsable de l'écaillage après sa réflexion en surface libre (i.e. l'autre volume de vide à droite de la simulation). Ces volumes sont détectés et étiquetés par l'algorithme et doivent être retirés en post-traitement des données.



(a) Vue 3D du champs de densité de la simulation en (Fig. 3.2)



(b) Vue 3D de l'étiquetage réalisé avec la condition $\rho = 0$

FIGURE 3.9 – Simulation d'écaillage de Tantale impliquant 83,3 millions d'atomes et 595,3 millions de voxels.

3.4 Optimisation de notre prototype

Maintenant que nous avons un prototype fonctionnel d’algorithme d’ECC 3D parallélisé en mémoire distribuée et partagée, nous voulons l’optimiser suffisamment dans cette section pour qu’il soit utilisable en production sur des très gros systèmes et que son coût reste négligeable comparativement à celui de la DM. Cette version optimisée de l’algorithme fait l’objet d’une publication (annexe à ce chapitre). La section suivante présentera ses performances comparativement à la première version de l’algorithme et à celle de la DM.

Le point le plus limitant de notre prototype pour accomplir notre but de traiter des très gros systèmes est son traitement un peu rustre des frontières MPI pour lier les différents domaines. En effet, si on augmente la taille de notre système d’un certain facteur et que l’on augmente de ce même facteur le nombre de MPI pour le traiter, le nombre de frontières MPI va se multiplier et la quantité de données à traiter par le MPI principal peut exploser si le système devient trop grand. Cette explosion de données vient du fait que pour chaque paire de voxel et voxel fantôme replissant la condition pour être de premier plan génère un triptyque de données à envoyer au MPI principal. Nous proposons donc d’optimiser l’algorithme en diminuant la quantité de données envoyée pour connecter les frontières MPI.

On peut constater une redondance d’informations dans les messages que nous envoyons pour traiter une connexion entre deux composantes connexes (Fig. 3.6b) aux frontières des domaines MPI. Il est donc possible de diminuer la taille des messages à traiter par le MPI principal en faisant disparaître cette redondance. Pour cela, les processus MPI doivent choisir quelle données envoyer par pair de composantes connexes au MPI principal et ce choix doit concorder avec le choix de leur MPI voisin pour que le MPI principal soit en mesure d’établir une connexion. Pour que ce choix concorde, une solution serait de faire communiquer entre eux les MPI voisins pour se mettre en accord. Cependant, ces communications supplémentaires peuvent être coûteuses et sont évitables. Ce choix des données à envoyer peut concorder sans communications si chaque MPI peut appliquer indépendamment une même procédure que son voisin et qui garantie la concordance des messages.

Nous avons ainsi optimiser notre algorithme en le faisant évoluer de deux façons. La première, présentée dans le prochain paragraphe, est le développement d’une procédure qui garantie d’une part la concordance des messages aux frontières pour effectuer les connexions entre composantes connexes et d’autre part qui garantie que ces messages sont de taille minimum. La deuxième, présentée ici, a été de s’affranchir de la structure de données `grid_cell_values`, et d’utiliser notre propre structure `image_3D` (Fig. 3.10) linéarisée dans l’espace pour permettre : d’effectuer un parcours en *raster scan* contigu en mémoire, d’appliquer les optimisations usuelles de scan de voisins, et d’indicer bien plus facilement nos voxels dans le code en n’ayant plus à prendre en compte le découpage en sous cellules. De plus, nous avons profité de ce changement de structure mémoire pour réorganiser les champs de données les uns à la suite des autres dans une *memory pool* de la même façon que les structures de données VTK, ce qui facilite leur utilisation pour la visualisation (Sec. 3.2). Aussi, nos champs sont tous de type `double`, la contiguïté mémoire de la structure est donc assurée. C’est désormais une structure de type *SoA*¹⁰.

La procédure que nous avons développé pour garantir la concordance des messages aux frontières et que les messages sont de taille minimum, consiste à effectuer une phase supplémentaire d’ECC 2D sur les couches de voxels fantômes et non fantômes aux frontières pour établir un triptyque unique de données par paire d’étiquette 2D fantôme et non fantôme

10. Structure of Arrays

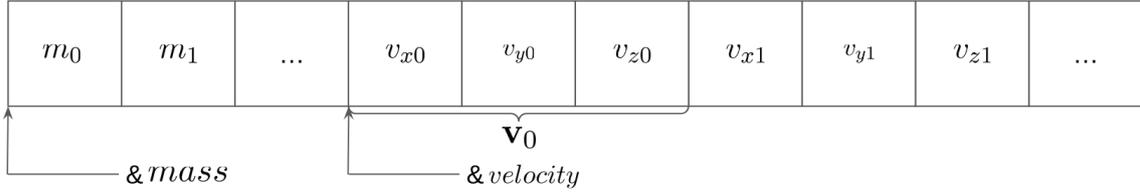


FIGURE 3.10 – Exemple d’organisation mémoire de la structure `image_3D` n’ayant pour champs que la masse et la vitesse des voxels. Les `&` symbolisent l’adresse mémoire des tableaux.

connecté en 3D. Les étiquettes 2D sont déterminées à partir des positions absolues des cellules dans l’espace de simulation, et l’ECC 2D doit être réalisé pour chaque surface voisine du domaine MPI qui le réalise pour garantir que le message correspondra à celui envoyé par son voisin sans qu’ils aient eu à échanger entre eux. Nous parlons de surface voisine pour les différencier des frontières du domaine MPI. Les frontières du domaine sont les 6 faces qui délimitent le domaine MPI local alors que les surfaces voisines sont les parties de ses frontières en contact entre deux domaines MPI. Plusieurs surfaces voisines peuvent être présentes sur une frontière à cause de la décomposition de domaine par le partitionnement RCB de la DM. L’ECC 2D est effectué avec les algorithmes 3 et 5 spécialisés pour la 2D (connexité-8, masque de Rosenfeld avec arbre de décision [93]).

Une fois l’ECC 2D effectuée, une passe supplémentaire sur les frontières MPI est effectuée et lie les étiquettes 2D fantômes et non fantômes en les stockant dans une table de hachage ayant pour clé la paire d’étiquette 2D et pour valeur l’étiquette 3D associée à l’étiquette 2D non-fantôme. Les données de cette table constituent ensuite les nouveaux triptyques de données envoyés au MPI principal par les MPI secondaires. Le processus MPI principal peut alors appliquer l’algorithme 7 comme précédemment mais il aura beaucoup moins de données à traiter. Dans cette nouvelle version, les identifiants de cellule v_{id} et g_{id} dans les messages sont remplacés par les étiquettes 2D e_{2D}^v pour la partie non-fantôme et e_{2D}^g pour la partie fantôme. La figure 3.11 reprend l’exemple de la figure 3.6 mais applique cette fois la procédure que nous venons de décrire. On peut déjà constater sur cet exemple simple que la taille des messages est drastiquement réduite. Ce changement et notre autre modification de la structure mémoire n’induisent pas de changement sur le résumé de l’algorithme (Fig. 3.7).

Les performances de cet algorithme sont discutées en détails dans l’article en annexe mais sont abordées dans la section suivante. Cette section confronte les performances des deux versions de l’algorithme pour vérifier que ces changements sont judicieux et optimisent effectivement les performances de notre prototype.

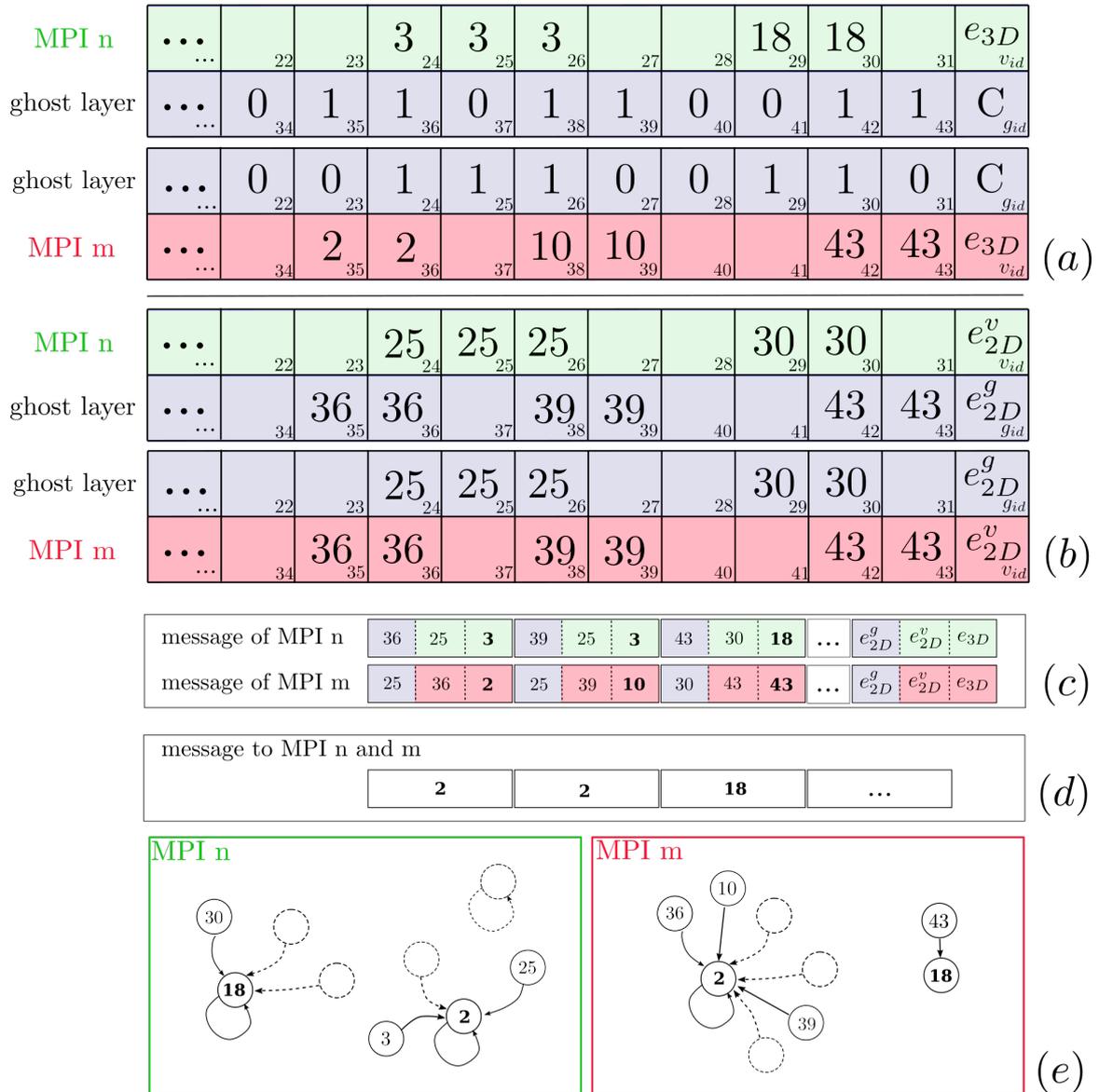


FIGURE 3.11 – (a) Exemple d’étiquetage 3D à la surface d’un MPI n et m ayant une surface commune après la première passe. (b) Étiquetage 2D associés à l’exemple (a). (c), réalisés avec l’algorithme 6 et envoyés au MPI principal par les MPI n et m . (d) Message envoyé par le MPI principal en réponse après qu’il ait appliqué l’algorithme 7. (e) Chaque thread de chaque MPI insère dans sa table d’équivalence les racines définitives de ses étiquettes à partir du message qu’il a reçu en (c).

3.5 Comparaison des performances des deux algorithmes

Pour vérifier que les modifications que nous avons apportées à l’algorithme sont judicieuses, nous comparons les performances des deux versions de l’algorithme sur les mêmes cas que ceux de l’article en annexe. Dans les figures suivantes, “original analysis” désigne la première version

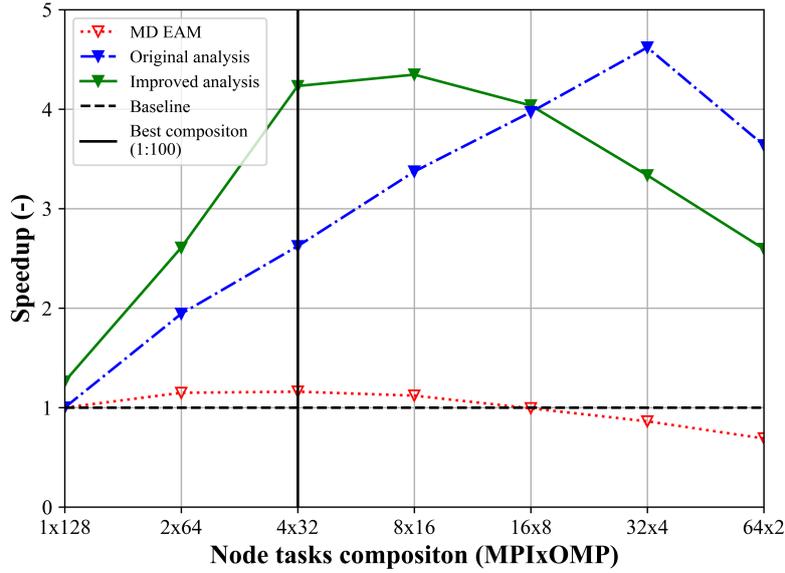


FIGURE 3.12 – L’accélération des différents algorithmes en fonction de la répartition en tâches MPI et OMP au sein d’un noeud.

de l’algorithme (Sec. 3.3) et “improved analysis” désigne la deuxième version de l’algorithme (Sec. 3.4). Nous commençons par étudier l’influence de la répartition des threads OMP et des processus MPI au sein d’un nœud du calculateur CEA-HF sur le temps d’exécution de l’algorithme (Fig. 3.12). Le cas de test ici est une simulation d’écaillage de Tantale qui implique 81,9 millions d’atomes et est projeté sur 58,9 millions de voxels. Le potentiel utilisé est celui présenté pour le Tantale dans le chapitre 2. Nous utilisons 4 noeuds bi-socket de processeurs AMD milan EPYCC 7763 64-Core. La mesure s’effectue sur 300 itérations de DM et 30 itérations d’analyses.

L’accélération relative des différents algorithmes est tracée en fonction de la répartition en tâches MPI et OMP. La répartition 1 processus MPI et 128 threads OMP sert de référence à la mesure d’accélération des algorithmes et le point de mesure de la première version de l’algorithme d’analyse sert aussi de référence pour la deuxième version. Pour tous les algorithmes, répartir les threads en plus de processus accélère leur exécution jusqu’à un certain point à partir duquel ils commencent à décélérer. Augmenter le nombre de MPI tout en gardant constant le nombre total de tâches parallèles a pour conséquence d’augmenter le nombre de communications et donc de ralentir l’exécution des algorithmes, mais nos noeuds de calculs étant des noeuds NUMA¹¹ cela a aussi pour conséquence d’augmenter la localité des données en mémoire pour les threads OMP et donc d’accélérer l’accès aux données et par extension accélérer l’exécution des algorithmes. C’est la compétition de ces deux mécanismes qui va fixer si l’algorithme accélère ou décélère au global. L’effet NUMA étant d’autant plus important entre les niveaux les plus hauts dans la hiérarchie d’accès mémoire (Annexe CITE), l’accélération due à la localité des données est plus importante sur les premières transitions de niveaux. Le CPU AMD EPYC 7763 a quatre niveaux de hiérarchie NUMA (64 → 32 → 16 → 8), le nœud de calcul ajoute un niveau supplémentaire en ayant 2 sockets (128 → 64). La DM étant un algorithme majoritairement *compute bound*,

11. NUMA : Non Uniform Memory Access

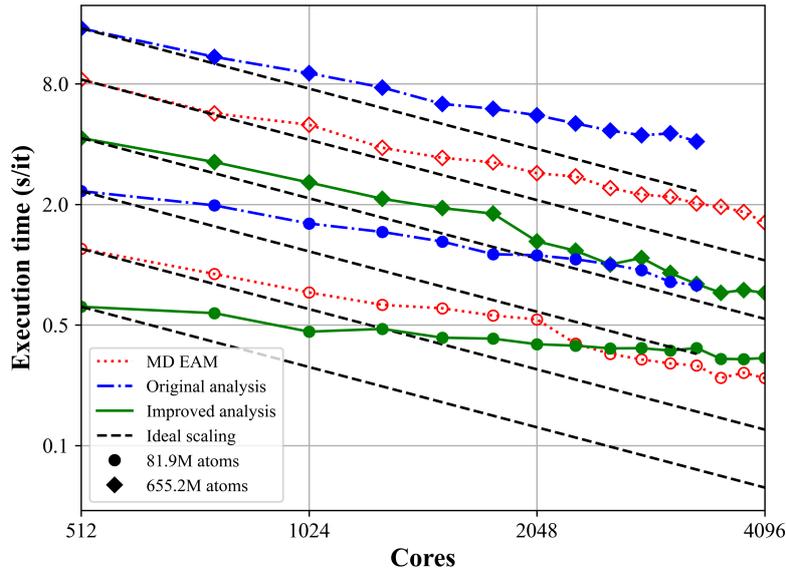
elle ne profite que peu des améliorations d'accès mémoire et seule une localité sur les deux premiers niveaux de hiérarchie NUMA bénéficie à son calcul avant d'être contrebalancé par les communications. Par contre, l'algorithme d'analyse est majoritairement *memory bound* dans ses deux versions, il profite donc grandement des améliorations d'accès mémoire. La première version de l'algorithme arrive à en profiter jusqu'au dernier niveau de hiérarchie (grappe de 8 cœurs) alors que la deuxième version de l'algorithme en profite jusqu'à l'avant dernier niveau de hiérarchie (groupe de 16 cœurs). On observe aussi que la première version de l'algorithme continue d'avoir une accélération au-delà du dernier niveau de hiérarchie NUMA (32 MPI avec 4 threads OMP par processus), donc cela est dû à un phénomène supplémentaire.

En effet, dans notre algorithme, augmenter le nombre de MPI augmente le nombre de surfaces et la surface totale à traiter pour connecter les régions MPI mais fournit aussi plus de travailleurs pour effectuer cette tâche. Cela concentre aussi d'autant plus de données à traiter sur le MPI maître. Dans la première version de l'algorithme, le travail de connexion aux frontières des domaines est simple car chaque voxel de premier plan est envoyé au maître mais cela génère une grande quantité de données à traiter, alors que dans la deuxième version de l'algorithme, le travail de connexion aux frontières est plus complexe avec la réalisation d'ECC 2D mais diminue grandement la quantité de données à traiter par le MPI maître. Il y a donc un équilibre de charge de travail différent dans les deux versions. La concurrence de ces trois phénomènes : localité mémoire, quantité de communications, et charge algorithmique, a un optimum de performance différent pour chaque algorithme. Cependant, pour choisir la répartition MPI/OMP par noeud nous devons choisir celle qui minimise l'ensemble du temps de calcul DM+analyse, et dans le cas où l'analyse n'est appelé que tous les 100 pas de temps de DM, la meilleur répartition est de 4 MPI par noeud et 32 threads par MPI. Avec cette répartition, la nouvelle version de l'algorithme est bien plus performante que l'ancienne et sera discutée avec la figure 3.13.

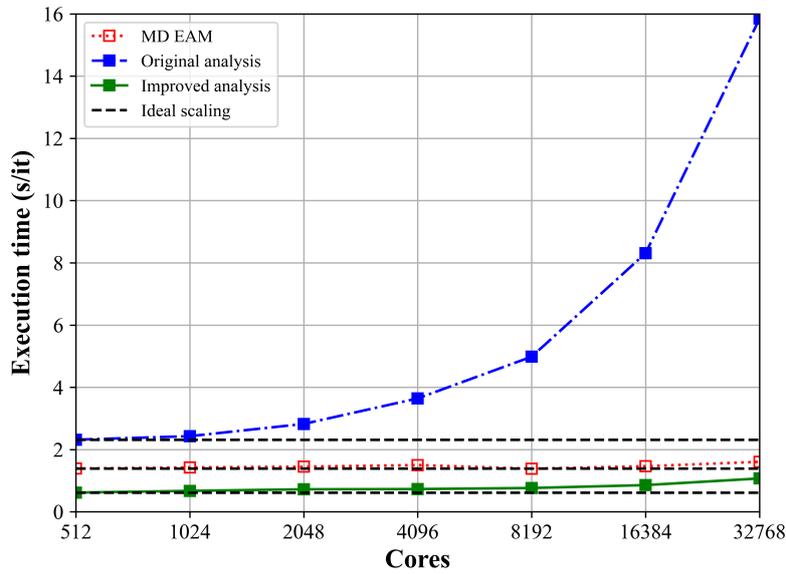
Comparons maintenant les performances des deux versions de l'algorithme sur nos résultats de scalabilité forte et de scalabilité faible. La sous figure 3.13a présente les performances de scalabilité forte sur deux charges de travail différentes, la première de 81,9 millions d'atomes est la même que celle de la figure 3.12, et la deuxième est une charge 8 fois plus grande de 655,2 millions d'atomes et 471,6 millions de voxels. La sous figure 3.13b présente les performances de scalabilité faible, la charge de travail initiale est celle de 81,9 millions d'atomes avec 58,9 millions de voxels et celle finale est de 5,2 milliards d'atomes avec 3,7 milliards de voxels. La comparaison du coût d'une itération d'analyse par rapport à une itération de DM est résumé dans la figure 3.14 pour l'ensemble de ces cas. Dans cette figure récapitulative, le temps d'exécution d'une analyse est divisée par le temps d'exécution d'une itération de DM dans ce cas de configuration. Ainsi, si un point à une valeur inférieure à 1, alors il est plus performant que la DM et s'il est supérieur à 1 alors il performe moins bien que la DM. De plus, la tendance de chaque courbe indique le comportement du scaling de l'analyse comparé à celui de la DM. Si une courbe est croissante, alors l'analyse scale moins bien que la DM. Si la courbe est constante, alors l'analyse et la DM scale de la même façon. Et si la courbe est décroissante, alors l'analyse scale mieux que la DM.

Les deux résultats principaux que l'on peut tirer de cette comparaison est que : la nouvelle version de l'algorithme est en moyenne entre 3 et 4 fois plus rapide que la première (Fig. 3.14), et elle scale bien mieux, en particulier pour la scalabilité faible qui a une performance 20 fois meilleure que précédemment pour le point de mesure à 32 768 coeurs et 5,2 milliard d'atomes (Fig. 3.13b).

Le gain de performance brut s'explique principalement par le changement de structure



(a) Scalabilité forte. Axe log-log.



(b) Scalabilité faible. Axe des abscisses logarithmique.

FIGURE 3.13 – Comparaison des performances des 2 versions de l'algorithme.

mémoire utilisée. La nouvelle structure mémoire s'apparente à une structure type SoA¹² et favorise le mécanisme de chargement mémoire par cache lines, la vectorisation par le compilateur, et le prefetching [101], comparé à l'ancienne structure qui s'apparente à une structure type AoS¹³. En mémoire, les données sont stockées dans ce que l'on appelle des pages mémoires et ces pages sont composées de cache lines, la plus petite quantité de mémoire auquel le CPU peut accéder. Lorsque le processeur veut effectuer une opération sur une

12. SoA : Struct of Arrays

13. AoS : Array of Structs

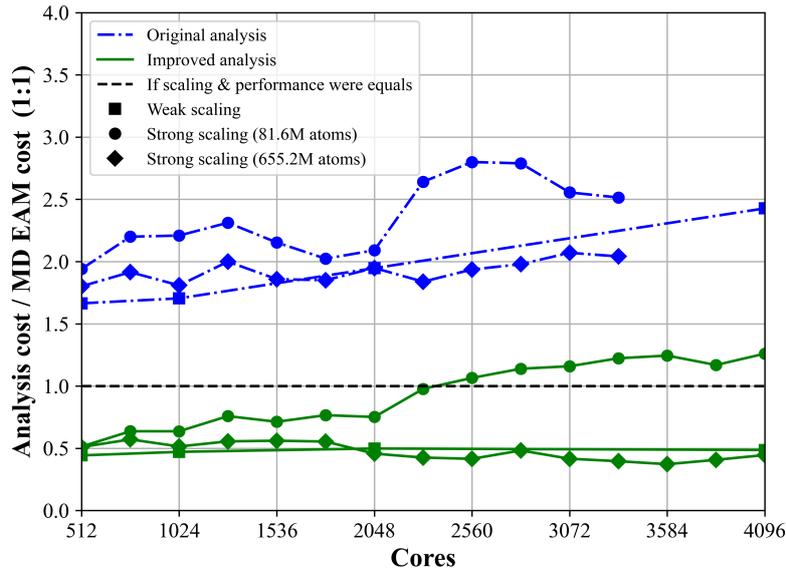


FIGURE 3.14 – Coût d’une itération d’analyse en fonction du coût d’une itération de DM

donnée en mémoire, par exemple une variable de type `double` (8 octets), il charge dans ses caches toute la cache line contenant cette variable (Fig. 3.15). La taille de ces cache lines varie d’un processeur à l’autre, dans notre cas, les cache lines font 64 octets [102]. Le processeur est ainsi capable de charger 8 `double` d’un coup et d’accéder à ces 8 valeurs. Si ces 8 valeurs sont utiles au processeur, on économise des chargements supplémentaires mais si seulement 1 des 8 est utile alors les autres 7 valeurs vont occuper de la place dans les caches pour rien et forcer plus souvent le processeur à effectuer des chargements. De plus le processeur est aussi doté de registres et d’ALU¹⁴ SIMD¹⁵ de 256 bits (32 octets), c’est à dire qu’il est capable d’effectuer des opérations arithmétiques et logiques sur des “vecteurs” de 4 `double` contigus en mémoire pour le prix d’une opération scalaire. Tous ces mécanismes favorisent non seulement l’ECC mais aussi la projection des atomes sur la grille régulière qui prend plus de la moitié du temps de calcul de l’analyse. Ce mécanisme favorise principalement la projection des atomes sur grille régulière qui prend plus de la moitié du temps de calcul de l’analyse, et possiblement l’ECC. L’ECC est surtout favorisé par le chargement des cache lines qui contiennent le champs des étiquettes de façon contiguë et doit donc diminuer la quantité de *cache miss*. L’autre gain apporté par l’utilisation de cette structure aux performances brutes de l’algorithme, est de pouvoir parcourir l’image en raster scan. Ceci nous a permis d’appliquer l’optimisation simple qui consiste à scanner seulement la moitié des voisins d’un voxel lors de la première passe d’ECC car ils sont les seuls à avoir potentiellement déjà été étiquetés. Une itération de l’algorithme coûte maintenant moins que le coût d’une itération de la DM avec un potentiel peu coûteux comme le EAM.

Le gain majeur en scalabilité faible s’explique par la modification du traitement de la connexion au voisinage des domaines MPI. La réduction de la quantité de données envoyée au MPI maître est d’au moins un ordre de grandeur et évite de créer un goulot d’étranglement séquentiel. En particulier lorsqu’une gigantesque composante connexe unique recouvre sur une frontière, la première version de l’algorithme envoie tous les voxels de la frontière au

14. ALU : Arithmetic and Logic Unit

15. SIMD : Single Instruction Multiple Data

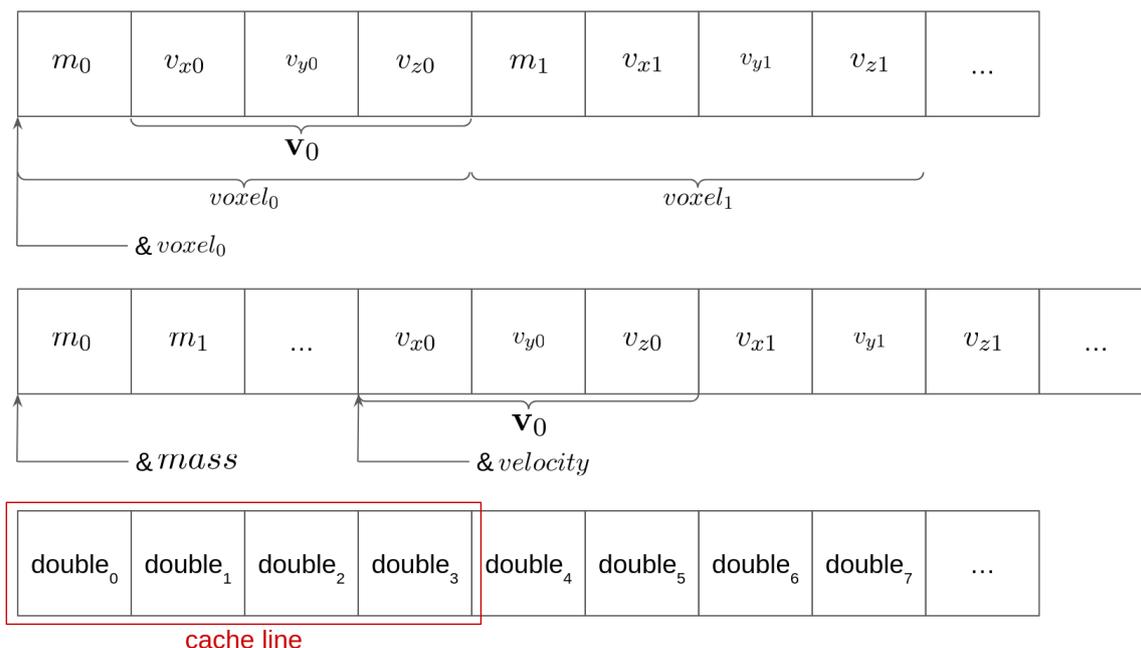


FIGURE 3.15 – Comparaison de l’organisation mémoire de la structure `grid_cell_values` et `image_3D` n’ayant pour champs que la masse et la vitesse des voxels, suivie d’une illustration du chargement d’une cache line (ici de 32 octets mais celles des processeurs Zen3 sont de 64 octets [102]).

maître alors que la nouvelle version de l’algorithme n’envoie que deux triptyques de données (un de par et d’autre de la frontière) pour la traiter.

3.6 Conclusion et perspectives

En conclusion, dans ce chapitre nous avons développé un algorithme de détection et de caractérisation de zones dites d’intérêts pour des simulations de DM de plusieurs milliards d’atomes. Dans le cadre de nos simulations d’endommagement, ces zones d’intérêts sont les pores qui nucléent, croissent et éventuellement qui coalescent suite au croisement d’ondes de détente. Notre algorithme est fondé sur un algorithme d’ECC que nous avons parallélisé en mémoire partagée et distribuée. Nous avons présenté une première version de notre algorithme que nous avons ensuite amélioré dans une deuxième version en réarrangeant la structure mémoire associée à l’algorithme et en modifiant la façon dont les frontières MPI sont traitées. Le coût d’une itération de notre algorithme représente en général la moitié du coût d’une itération de DM avec un potentiel EAM, un des potentiels les moins coûteux en DM. De plus, l’algorithme n’a besoin d’être exécuté à une fréquence d’au moins 1 ordre de grandeur plus faible que la DM pour les phénomènes ultra rapides et de plus de 2 ordre de grandeur dans la plupart des cas, ramenant ainsi le surcoût d’analyse à 5% au maximum et 0,5% en moyenne. Le coût d’application de l’algorithme est donc conscrit et nous avons rempli nos objectifs de développement. Cependant, nous n’avons pas caractérisé l’incertitude de mesure associée à l’algorithme en fonction de ses paramètres de projection et de résolution de grille. Cette étape est cruciale avant de pouvoir l’appliquer à un ensemble de simulations et fournir des résultats physiques fiables et exploitables. Cette étude fera donc l’objet du

chapitre suivant.

Nos développements conduisent aussi à de nouvelles perspectives. D'une part, d'un point de vue fonctionnel, étendre notre ECC 3D à un ECC 4D, avec le temps comme 4e dimension permettrait une détection automatique des événements de nucléation ou de disparition, et de coalescence ou de fragmentation, en plus de permettre un suivi temporel des zones d'intérêts. Et d'autre part, d'un point de vue des performances, la restructuration mémoire introduite par la deuxième version de l'algorithme pourrait permettre l'utilisation de bibliothèque d'ECC pour les calculs intra domaine MPI et potentiellement améliorer les performances de l'algorithme. Cependant nous avons vu que la majorité du temps de calcul de l'analyse est dédiée à la projection des atomes sur la grille régulière. L'utilisation de GPU pour cette partie calculatoire de l'analyse serait certainement une amélioration plus pertinente dans un premier temps. Une autre option d'utilisation des GPU, tout aussi pertinente, serait de dédier les GPU uniquement aux calculs de DM, laisser l'analyse sur CPU et de recouvrir les temps de calculs étant donné que l'analyse à une fréquence d'appel plus faible que la DM et ainsi rendre l'analyse virtuellement gratuite en temps de calcul.

Chapitre 4

Étude paramétrique du comportement de notre algorithme

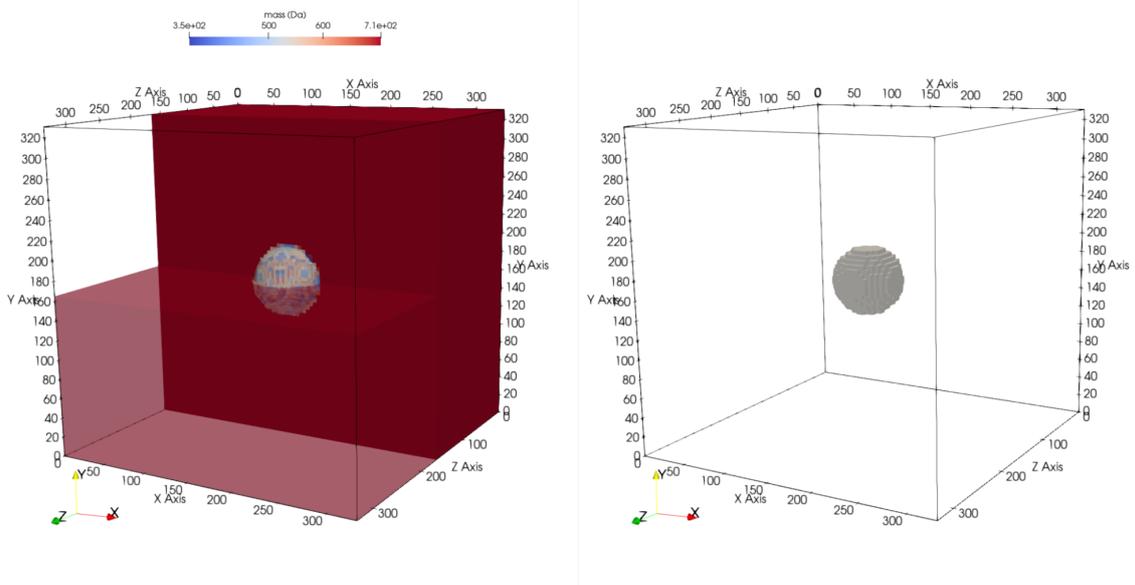
Dans le chapitre précédent (Chap. 3), nous avons présenté l'algorithme que nous avons développé pour pouvoir analyser des simulations de DM grande taille impliquant plusieurs milliards d'atomes. Nous avons caractérisé ses performances en temps par rapport à la DM à paramètres de grille constant en fonction de la répartition du parallélisme et de la charge de calcul. Avant de pouvoir appliquer notre algorithme à des études physiques, nous devons quantifier l'influence de la finesse de grille (son seul paramètre en dehors des paramètres de projection fixés par la physique) sur les résultats qu'il produit afin de choisir adéquatement la valeur de ce paramètre et connaître l'incertitude de mesure associée. L'objectif de ce chapitre sera donc de déterminer, dans un premier temps, l'influence de la finesse de grille sur les résultats pour le scénario simple d'une sphère incluse dans un réseau cristallin. Le but est de vérifier que l'algorithme converge vers la solution théorique lorsque l'on raffine la grille, et de caractériser l'erreur commise par l'algorithme. À partir de ces simulations, nous pourrions déterminer l'incertitude de mesure et ainsi étudier dans un second temps, l'influence de la finesse de grille sur un cas réel d'application.

4.1 Étude de convergence de la solution pour une sphère parfaite

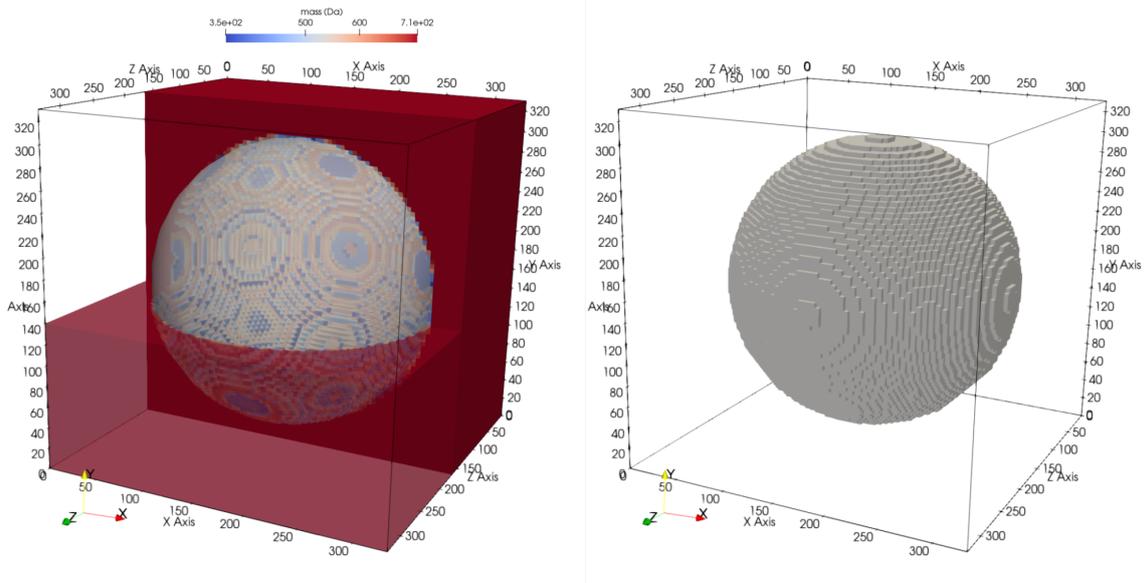
Le cas de référence que nous avons choisi pour étudier l'influence de la finesse de grille sur les résultats est celui d'une sphère parfaite. Ce cas est un scénario simple à mettre en place, et représentatif de nombreux cas physiques d'intérêt. Par exemple, dans le cas d'éjection de matière liquide ("*splashing*", "*micro-jetting*"), la tension de surface aura tendance à rendre les agrégats de matières sphériques, et dans le cas de l'écaillage de matériaux ductiles, on s'attend à ce que les pores soient sphériques avant leur coalescence (postulat de nombreux modèles d'endommagement macroscopique).

Nous avons généré des échantillons cubiques de Tantale sans défauts à 0 K de 33 nm d'arête (100 mailles atomiques), en conditions limites périodiques, dans lesquels nous avons retiré les atomes contenus dans des sphères de différents volumes (une sphère par échantillon, centrée dans l'échantillon). Nous avons ensuite appliqué notre algorithme (Chap. 3) avec un filtre de densité nulle ($\rho_s < 10^{-3}\rho_{Ta}$, avec ρ_{Ta} la densité du Tantale à 300K et ρ_s la densité seuil) pour sélectionner les cellules vides de l'échantillon, les clusteriser, et calculer le

volume obtenu. Nous avons fait varier le rayon des sphères de 4 angstroms à 160 angstroms et nous avons échantillonné linéairement 1000 points. Nous présentons deux de ces échantillons projetés sur grille régulière dans la figure 4.1, dans la partie haute, le rayon de la sphère fait 34,73 angstroms (volume $1,78 \times 10^{-7} \mu m^3$), et dans la partie basse, le rayon de la sphère fait 135 angstroms (volume $1,03 \times 10^{-5} \mu m^3$). La partie gauche de la figure est obtenue avec 2 coupes, une première définie par un plan normal à l'axe z positionné au milieu de l'échantillon qui met en transparence les cellules entre l'observateur et le plan, puis une deuxième coupe cette fois selon l'axe y, orthogonal à la première coupe et qui cache les cellules en transparence au dessus de ce plan. La sphère qui a servi à supprimer les atomes dans la simulation est ajoutée en transparence dans une couleur neutre. Le champs de couleur correspond à la masse des cellules. Ce champ est uniforme dans l'échantillon à l'exception des cellules en bordure du vide. La partie droite de la figure présente les cellules vides détectées par l'algorithme (les autres cellules ayant été cachées).



(a) Échantillon avec une sphère de rayon à 34,73 angstroms (volume $1,78 \times 10^{-7} \mu\text{m}^3$)



(b) Échantillon avec une sphère de rayon à 135 angstroms (volume $1,03 \times 10^{-5} \mu\text{m}^3$)

FIGURE 4.1 – Deux de nos échantillons, avec à gauche, une vue de la sphère que nous avons retiré dans l'échantillon, et à droite, la sphère détecté par notre algorithme.

Pour chacun de ces 1000 volumes générés, nous avons mesuré avec 12 résolutions de grilles différentes le volume de vide détecté par l'algorithme. Dans la figure 4.2, les mesures de 5 de ces volumes de sphère sont reportées en bleu, ainsi que les mesures correspondantes aux cas de sphères de la figure 4.1. La plus faible résolution de grille ($1,13 \times 10^8$ voxels/ μm^3) correspond à une grille régulière dont les voxels font la taille des cellules de Verlet. La plus haute résolution utilisée dans ces mesures ($1,96 \times 10^{11}$ voxels/ μm^3) correspond à une grille régulière dont les cellules de Verlet ont été subdivisées en 12 selon chaque direction de

l'espace. La résolution de grille utilisée pour la figure 4.1 correspond à une subdivision des cellules de Verlet en 5 selon chaque direction de l'espace ($1,41 \times 10^{10}$ voxels/ μm^3). La valeur de volume théorique utilisée dans la figure 4.2 correspond au volume de la sphère. Ce volume est reporté dans la figure 4.2 en pointillé noir pour chacun des 5 cas et nous y associons les mesures avec un gradient de couleur bleu. On retrouve bien le comportement attendu de notre analyse : elle converge vers la solution à mesure que la grille est raffinée. De plus, plus le volume de la sphère est grand, plus le volume calculé par l'analyse est proche du volume théorique pour une résolution de grille fixée. Cependant, on peut constater que pour des petits volumes de sphères, l'écart entre la solution théorique et notre calcul est assez conséquent. Pour réduire cet écart, nous proposons dans la section suivante une amélioration de l'algorithme.

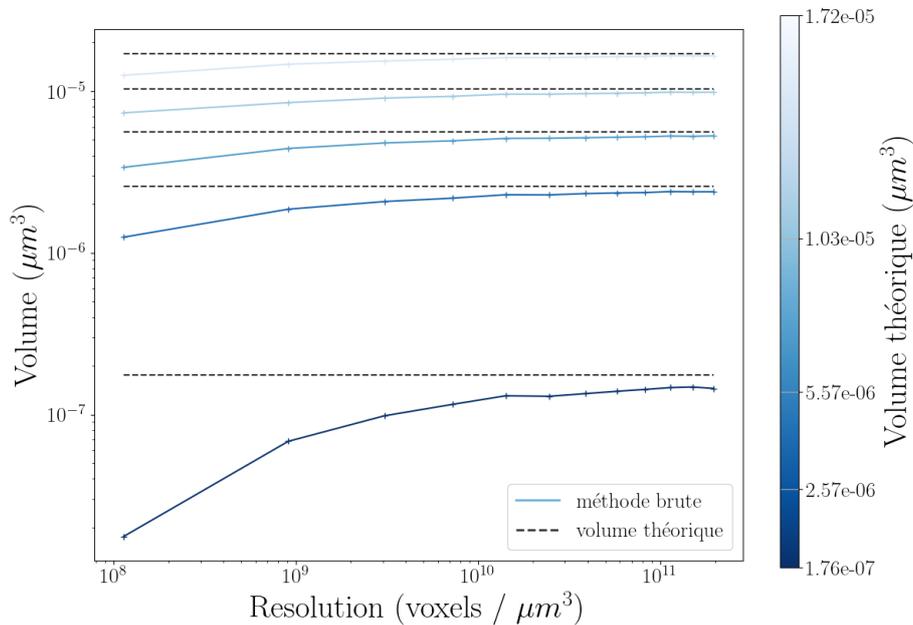


FIGURE 4.2 – Cinq séries de mesure de volume à différentes résolution de grille comparé aux volumes théoriques associés. Axes logarithmiques.

4.2 Amélioration de l'algorithme

Le problème principal de devoir décrire une sphère avec des cubes est qu'il faut une infinité de cube infiniment petit pour pouvoir la paver. Informatiquement, nous ne pouvons avoir qu'un nombre fini de cube pour résoudre ce problème (limitation en mémoire). Augmenter la subdivision de nos cellules augmente leur nombre de façon cubique, limitant ainsi rapidement la résolution que nous pouvons atteindre en pratique. L'autre problème engendré par cette description est illustré dans la figure 4.3, pour une résolution de grille donnée, toutes les sphères dont le volume hachuré en noir est compris entre celui des sphères en rouge, produiront le même volume en vert.

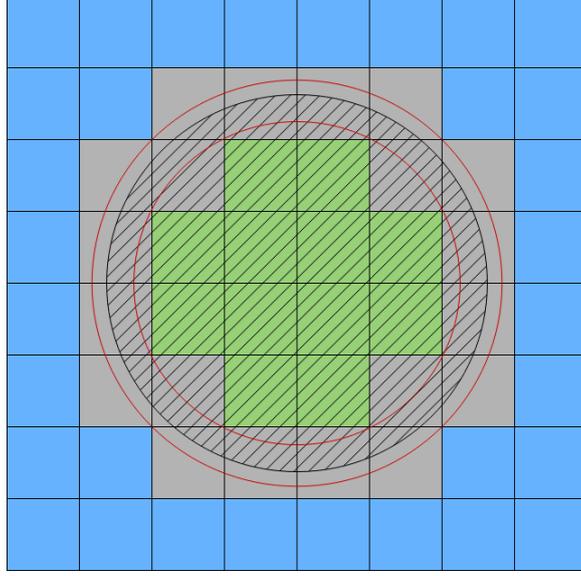


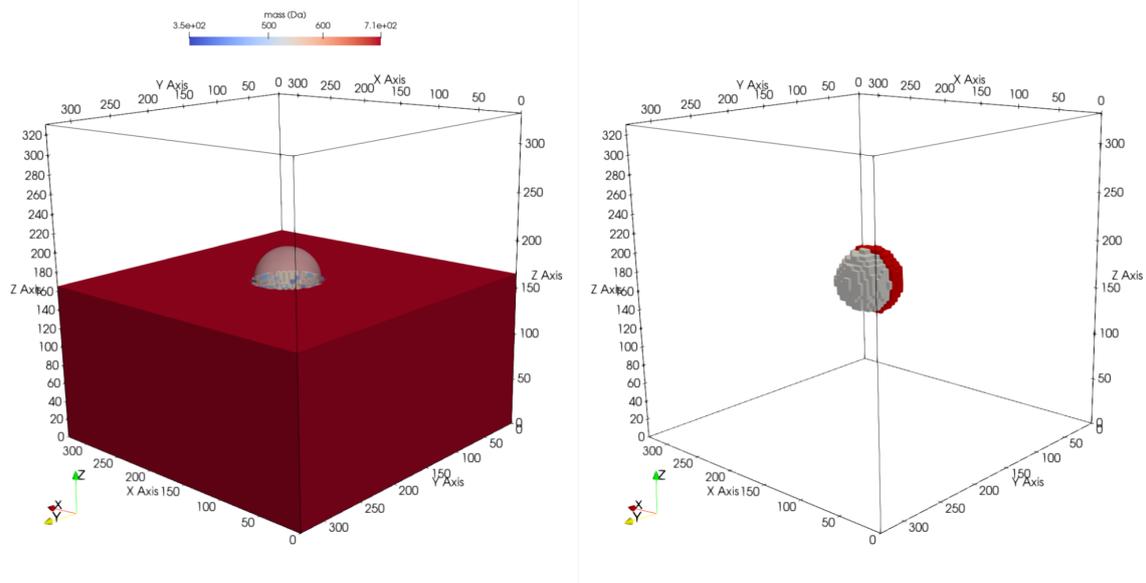
FIGURE 4.3 – Illustration d’un échantillon de matière contenant un pore, avec en bleu les cellules pleines, en gris les cellules contenant du vide et de la matière, et en vert les cellules entièrement vides. Avec la méthode brute, l’ensemble des sphères noires comprises dans les limites rouges auront le même volume vert.

Pour pallier en partie ces problèmes, nous proposons d’utiliser des considérations physiques pour prendre en compte les cellules partiellement vides (les cellules grises de la figure 4.3) et d’ajouter au volume du pore uniquement la fraction de vide dans ces cellules. La fraction de vide d’une cellule est déterminée à partir de l’écart entre la densité de la cellule ρ et une densité ρ_0 fixé par l’utilisateur (Eq. 4.1). Ce choix peut être discuté et doit être motivé par la physique étudiée, dans notre cas, nous choisissons $\rho_0 = \rho_{Ta}$ avec ρ_{Ta} la densité du Tantale solide à 300K.

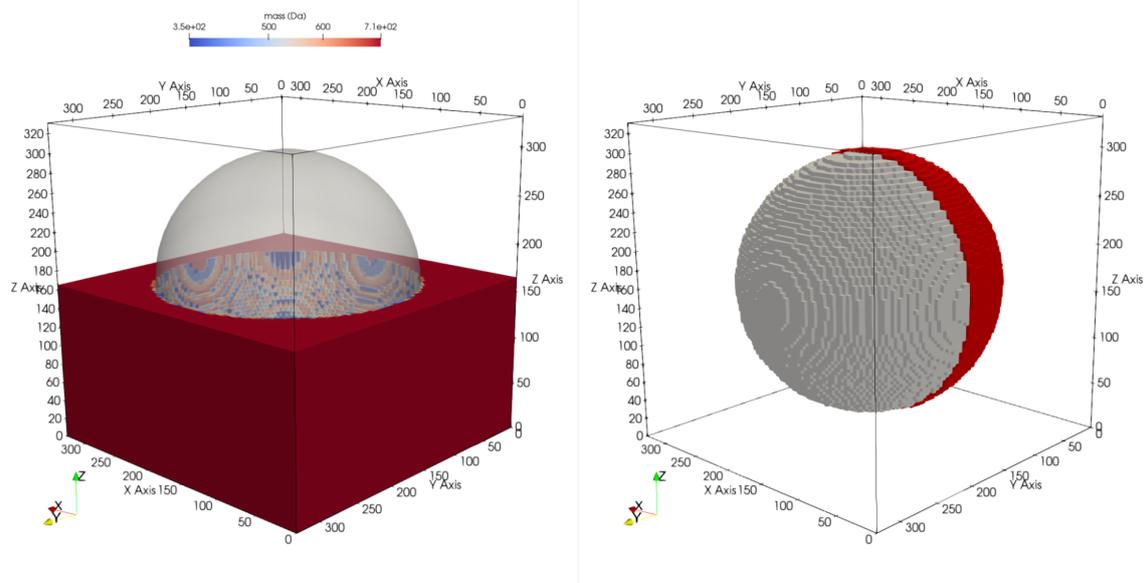
$$V_{voxel}^{corr} = s^3 - \frac{\rho}{\rho_0} s^3, \quad s \text{ l’arête d’un voxel} \quad (4.1)$$

$$\Leftrightarrow V_{voxel}^{corr} = s^3 - \frac{m}{\rho_0}, \quad \text{avec } m \text{ la masse de la cellule.} \quad (4.2)$$

Cette technique implique aussi de devoir relever le seuil de sélection des cellules ρ_s , pour pouvoir inclure les voxels partiellement vides. Cela pose un choix supplémentaire, lui aussi discutable et qui doit être motivé par la physique. Dans cette configuration, nous choisissons $\rho_s = 0,8\rho_{Ta}$ pour garder une marge de sécurité et ne pas sélectionner trop de cellules. Dans la figure 4.4, nous présentons les mêmes échantillons que dans la figure 4.1 avec dans la partie gauche une coupe de l’échantillon selon l’axe z qui cache les cellules au-dessus du plan, et représente la sphère que nous avons utilisée pour supprimer des atomes en transparence. La partie droite présente cette fois les sphères de cellules sélectionnées par chaque version de l’algorithme. Les cellules de couleur neutre sont celles entièrement vides (Sec. 4.1) et les cellules rouges sont celles avec la sélection à 80% de densité. La moitié des cellules rouges sont cachées pour permettre de voir la sphère de cellules vides. La “peau” de cellules supplémentaires due à la nouvelle sélection est ainsi mieux mise en évidence.



(a) Échantillon avec une sphère de rayon à 34,73 angstroms (volume $1,78 \times 10^{-7} \mu\text{m}^3$)



(b) Échantillon avec une sphère de rayon à 135 angstroms (volume $1,03 \times 10^{-5} \mu\text{m}^3$)

FIGURE 4.4 – Échantillons de la figure 4.1, avec à gauche une vue de la sphère retirée. À droite une comparaison des sphères sélectionnées selon la méthode utilisée, avec en rouge les cellules de la méthode avec correction, dont la moitié ont été cachées pour voir la sphère de cellule grises sélectionnées par la méthode brute.

La réalisation des mêmes mesures que précédemment avec cette nouvelle méthode que nous appellerons “méthode corrigée” ou “méthode avec correction”, permet de comparer ses résultats avec ceux de la méthode initiale, que nous appellerons “méthode brute”, qui sélectionne uniquement les voxels vides et somme leurs volumes sans correction. La figure 4.5 reprend la figure 4.2 et présente les deux méthodes avec en rouge la méthode corrigée et

en bleu la méthode brute. On peut observer que la méthode corrigée est bien meilleure que la méthode brute, converge très vite vers la solution théorique, et produit de bons résultats même pour une faible résolution de grille grâce à la physique qu'elle prend en compte. De plus, le surcoût de calcul de cette nouvelle méthode est négligeable.

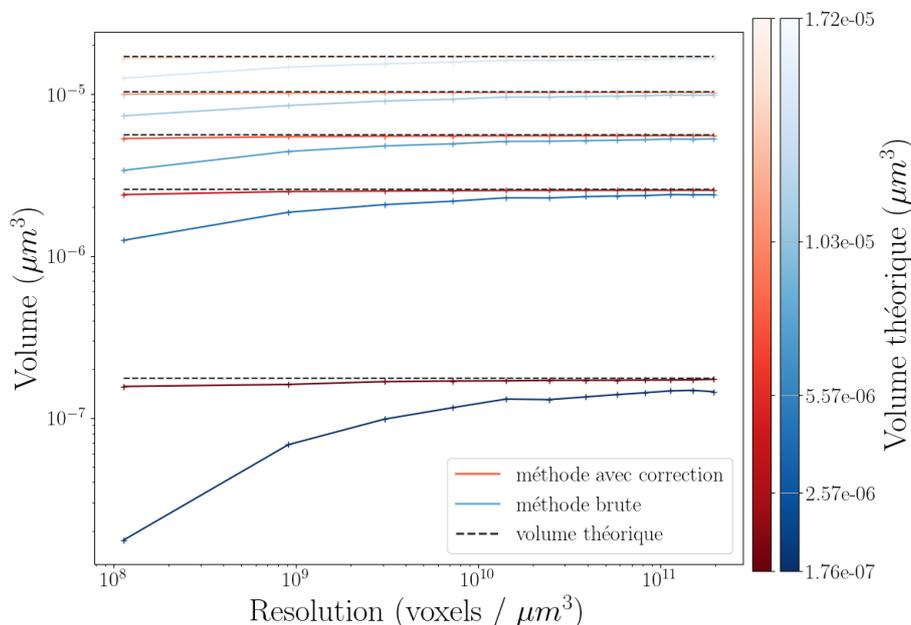


FIGURE 4.5 – Comparaison des cinq séries de mesure de volume à différentes résolutions de grille de la figure 4.2 avec la méthode avec correction. Axes logarithmiques.

Nous avons validé la pertinence de notre méthode de correction, et la convergence de l'analyse avec les deux méthodes. Maintenant, nous allons étudier plus en détail l'erreur commise par l'analyse avec les deux méthodes.

4.3 Évaluation de l'erreur algorithmique

En reprenant l'ensemble de nos mesures précédentes, nous pouvons calculer l'erreur relative commise par l'analyse σ_V (Eq. 4.3) pour chacune des deux méthodes, et tracer cette erreur relative en fonction du volume calculé par l'analyse V_c avec chaque finesse de grille. La figure 4.6 présente à gauche (4.6a) en bleu les résultats de la méthode brute et à droite (4.6b) en vert, les résultats de la méthode avec correction. Nous avons divisé les valeurs de volume calculées par le volume d'une maille cristalline (a^3) pour adimensionner les résultats, et pour la même raison, nous avons exprimé la finesse de la grille par la valeur de s/a avec s la longueur de l'arête de nos voxels et a le paramètre de maille. Les deux sous figures sont tracées avec des axes logarithmiques en abscisse et en ordonnée.

$$\sigma_V = \frac{|V_{th} - V_c|}{V_{th}} \quad (4.3)$$

De l'observation de ces courbes, nous avons estimé que malgré les nombreuses oscillations (principalement dû à l'effet décrit avec la figure 4.3) de nos courbes, à grand volume, leur

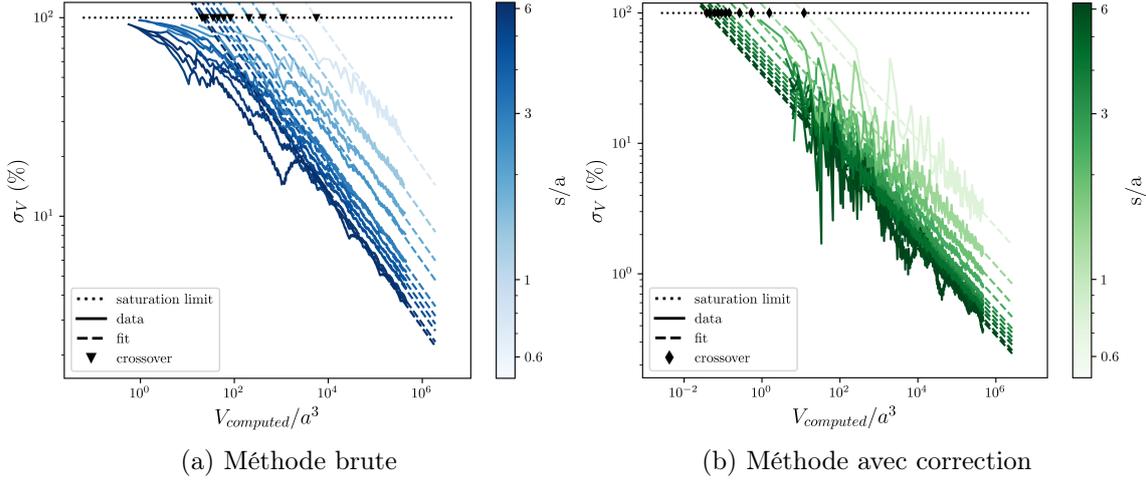


FIGURE 4.6 – Erreur relative en fonction du volume calculé adimensionné par le volume d’une maille cristalline et en fonction de la résolution de la grille exprimé en fonction de s/a , avec s l’arête d’un voxel et a le paramètre de maille. Ajustement paramétrique réalisé avec le système (Eq. 4.5 - 4.6)

comportement est quasi-linéaire dans un diagramme log-log. Nous avons alors posé que pour chaque courbe, à grand volume, on peut écrire :

$$\log(\sigma_V) = p \log(V_c/a^3) + q \quad (4.4)$$

Nous pourrions ensuite directement faire une régression linéaire sur chaque courbe pour obtenir les paramètres p et q de chacune d’elles, avec p la pente et q l’ordonnée à l’origine. Néanmoins, nous aurions obtenu une valeur de p différente pour chaque courbe. Afin d’extraire une loi commune à ces observations, nous avons contraint notre problème pour imposer une pente p commune à chaque courbe, en adoptant la formulation matricielle suivante :

$$A x = b \quad (4.5)$$

avec :

$$A = \begin{pmatrix} \log(V_{c_0}^0/a^3) & 1 & 0 & 0 & \dots & 0 & 0 \\ \log(V_{c_1}^0/a^3) & 1 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \log(V_{c_n}^0/a^3) & 1 & 0 & 0 & \dots & 0 & 0 \\ \log(V_{c_0}^1/a^3) & 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \log(V_{c_n}^1/a^3) & 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \log(V_{c_0}^m/a^3) & 0 & 0 & 0 & \dots & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \log(V_{c_n}^m/a^3) & 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix}, \quad x = \begin{pmatrix} p \\ q_0 \\ q_1 \\ \vdots \\ q_m \end{pmatrix}, \quad b = \begin{pmatrix} \log(\sigma_{V_0}^0) \\ \log(\sigma_{V_1}^0) \\ \vdots \\ \log(\sigma_{V_n}^0) \\ \log(\sigma_{V_0}^1) \\ \vdots \\ \log(\sigma_{V_n}^1) \\ \vdots \\ \log(\sigma_{V_0}^m) \\ \vdots \\ \log(\sigma_{V_n}^m) \end{pmatrix} \quad (4.6)$$

On la résout avec la méthode des moindres carrés, où l’on minimise le résidu $r = \frac{1}{2} \|Ax - b\|^2$. Nous avons ainsi obtenu une pente $p \approx -\frac{1}{3}$. Cette valeur n’a pas de signification particulière

mais a peut être une justification géométrique. À partir de l'équation (4.4), on a :

$$e^{\log(\sigma_V)} = e^{-\frac{1}{3} \log(V_c/a^3)+q} \quad (4.7)$$

$$\Leftrightarrow \sigma_V = (V_c/a^3)^{-1/3} \cdot e^q \quad (4.8)$$

$$\Leftrightarrow \sigma_V = \frac{a}{V_c^{1/3}} \cdot e^q \quad (4.9)$$

avec e^q qui dépend de s/a et de la méthode employée. Pour chaque méthode, on peut déterminer le point d'intersection entre les droites formées par l'équation (4.9) et la constante représentant 100% d'erreur pour déterminer le volume à partir duquel notre analyse sature pour chaque résolution de grille. Ces points d'intersection sont représentés par des triangles dans la figure 4.6a et des losanges dans la figure 4.6b.

4.3.1 Volume de saturation

On reporte ces points pour les deux méthodes dans la figure 4.7 en fonction de la valeur de s/a . Les triangles bleus correspondent à la méthode brute et les losanges verts correspondent à la méthode avec correction. Les deux axes de cette figure sont logarithmiques. On observe une nouvelle fois un comportement linéaire pour les deux méthodes et dont les pentes sont similaires. On pose donc :

$$\log(V_{sat}/a^3) = g \log(s/a) + h \quad (4.10)$$

La même méthode que précédemment contraint notre problème pour avoir une pente commune g aux deux courbes. On obtient ainsi $g \approx 9/4$ donc :

$$V_{sat}/a^3 = \left(\frac{s}{a}\right)^{9/4} \cdot e^h \quad (4.11)$$

avec $e_{corr}^h = 0,126$ pour la méthode avec correction et $e_{brute}^h = 75,54$. Comme précédemment, la valeur $g \approx 9/4$ n'a pas de signification particulière mais a peut être une justification géométrique.

Comme dans la section précédente (Sec. 4.2), on constate que la méthode corrigée est bien plus efficace que la méthode brute.

4.3.2 Loi d'incertitude de notre algorithme

Maintenant, en utilisant l'équation du volume de saturation, on peut normaliser l'abscisse de nos points de la figure 4.6 par V_{sat}/a^3 et ainsi obtenir la figure 4.8. En bleu, nous avons les courbes de la méthode brute et en vert celles de la méthode corrigée.

Ainsi, on obtient la superposition de toutes nos courbes indépendamment de la méthode et de la résolution de grille. L'équation maîtresse de ces courbes est alors :

$$\sigma_V = \left(\frac{V_c/a^3}{V_{sat}/a^3}\right)^{-1/3} \cdot e^k \quad (4.12)$$

$$\Leftrightarrow \sigma_V = \left(\frac{V_c}{V_{sat}}\right)^{-1/3} \cdot e^k \quad (4.13)$$

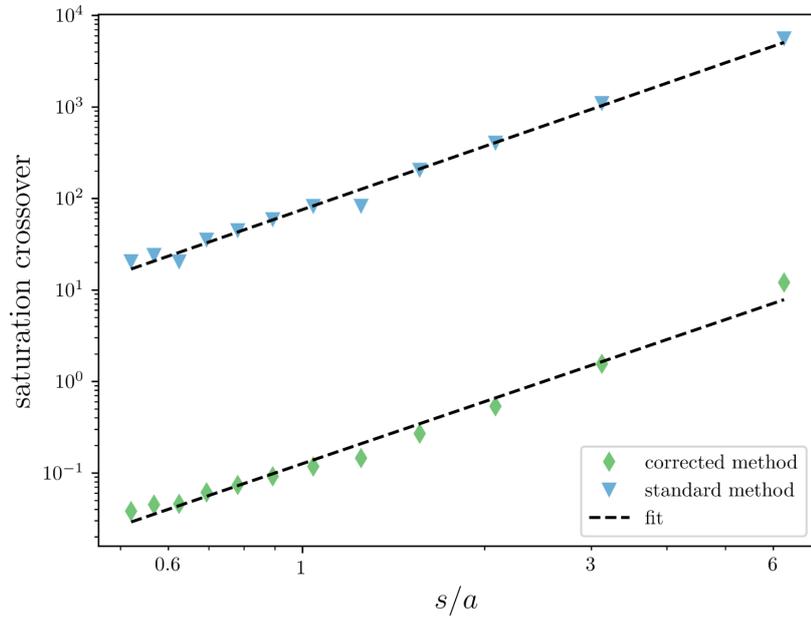


FIGURE 4.7 – V_{sat}/a^3 en fonction de s/a pour les deux méthodes. (Eq. 4.11)

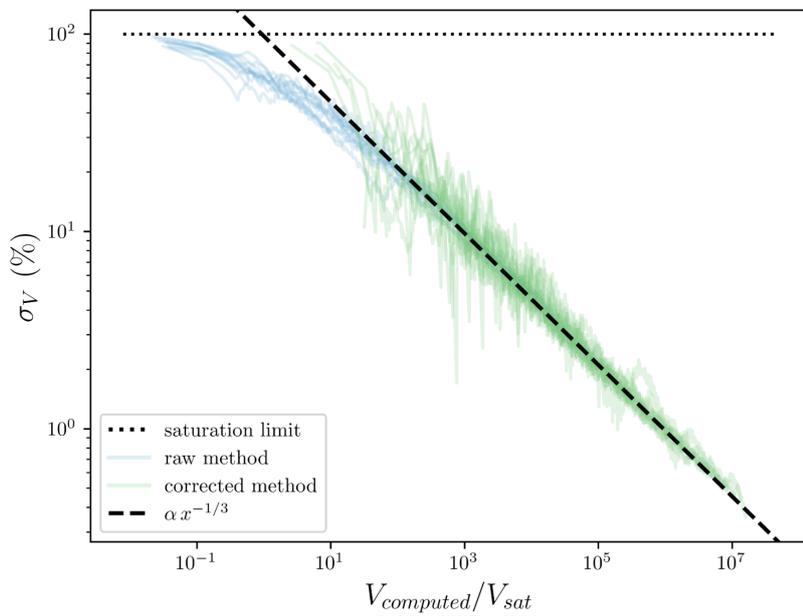


FIGURE 4.8 – Erreur relative (Fig. 4.6) avec l'abscisse normalisée par le volume de saturation V_{sat} . (Eq. 4.13)

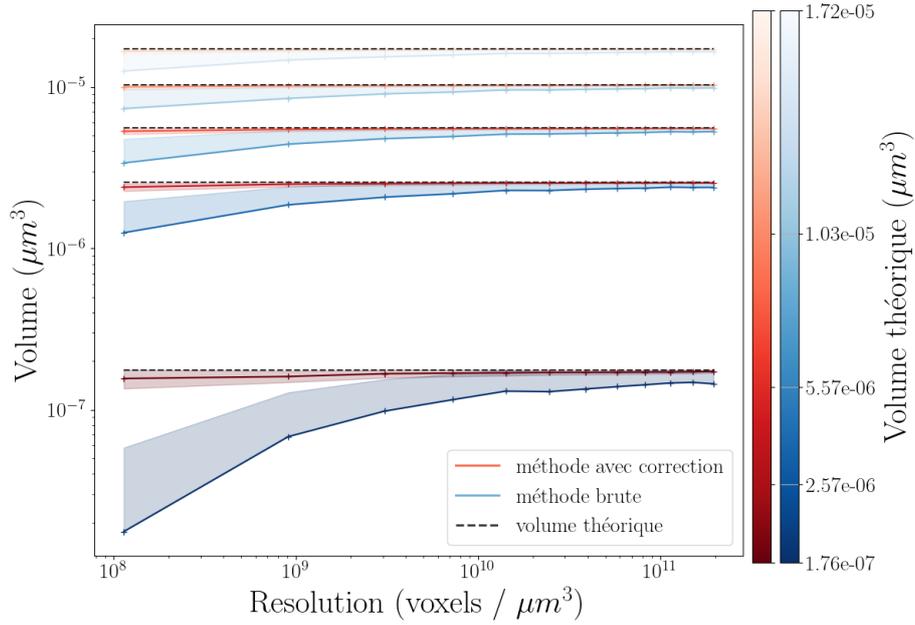


FIGURE 4.9 – Comparaison des cinq séries de mesure de volume à différentes résolution de grille de la figure 4.5 avec notre estimation de l’erreur (Eq. 4.16).

avec k l’ordonnée à l’origine de la droite d’équation $\log(\sigma_v) = \frac{-1}{3} \log\left(\frac{V_c/a^3}{V_{sat}/a^3}\right) + k$. En injectant l’équation (4.11) on a :

$$\Leftrightarrow \sigma_V = \left(\frac{V_c}{(s/a)^{9/4} \cdot a^3 \cdot e^h}\right)^{-1/3} \cdot e^k \quad (4.14)$$

$$\Leftrightarrow \sigma_V = \frac{a}{V_c^{1/3}} \left(\frac{s}{a}\right)^{3/4} \cdot e^{h/3+k} \quad (4.15)$$

$$\Leftrightarrow \sigma_V = \alpha \frac{a}{V_c^{1/3}} \left(\frac{s}{a}\right)^{3/4} \quad (4.16)$$

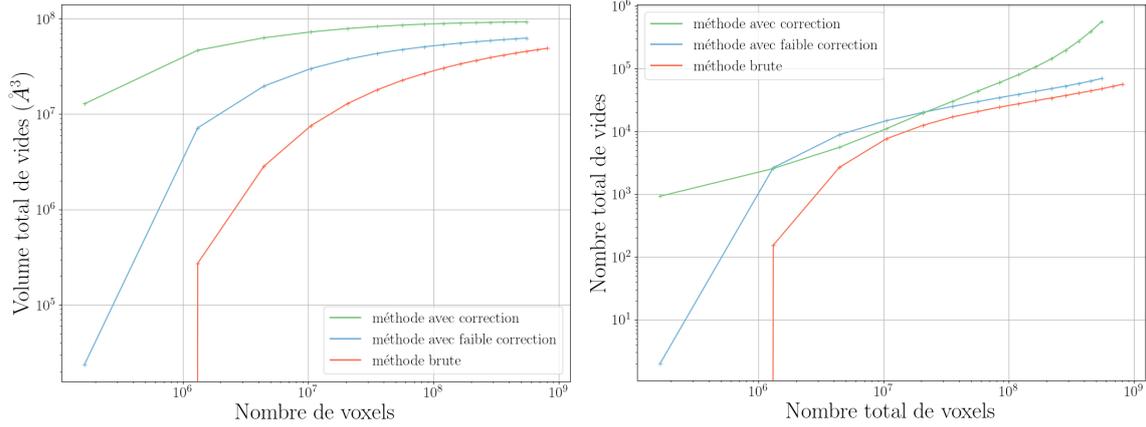
avec $\alpha = e^{h/3+k}$ qui ne dépend que de la méthode, tel que $\alpha_{corr} = 0,53$, et $\alpha_{brute} = 4,45$. Aussi, par identification (Eq. 4.9) : $e^q = \alpha \left(\frac{s}{a}\right)^{3/4}$.

Ainsi, nous avons une estimation de l’incertitude de mesure associée à nos deux méthodes, qui ne dépend que de paramètres connus : le volume calculé, la résolution de la grille et le paramètre de maille du cristal.

4.3.3 Résultats pour la sphère parfaite

Nous pouvons alors utiliser cette estimation (Eq. 4.16) pour encadrer nos précédents résultats de convergence (Fig. 4.5) dans le cas d’une sphère parfaite. Nous obtenons ainsi la figure 4.9.

Nous pouvons noter que dans le cas de la méthode brute, notre erreur de mesure ne peut que sous estimer le volume mesuré car par construction elle ne sélectionne que des voxels entièrement vides et ne peut donc pas ajouter de vide en trop. Dans le cas de la méthode



(a) Volume total de vides en fonction du nombre total de voxels (b) Nombre total de vides en fonction du nombre total de voxels

FIGURE 4.10 – Résultats de convergence sur le cas d’écaillage étudié dans le chapitre 2 (échantillon de 81,9 millions d’atomes initialement $100 \text{ nm} \times 100 \text{ nm} \times 150 \text{ nm}$, impacté à 1500 m/s). Méthode avec correction : $\rho_s = 0,5\rho_{Ta}$, Méthode avec faible correction : $\rho_s = 0,1\rho_{Ta}$, avec ρ_s la densité seuil de sélection.

corrigée, si les paramètres de densité seuil ρ_s et de correction ρ_0 ne sont pas bien adaptés, il est possible de calculer trop de vide par rapport à la réalité de la simulation, même si ce n’est pas le cas ici. C’est pourquoi dans la figure, nous encadrons les résultats de la méthode brute avec une incertitude uniquement par le haut alors que nous donnons une incertitude par le haut et le bas dans le cas de la méthode corrigée avec l’équation 4.16. Nous pouvons constater que dans le cas de la méthode corrigée, le volume théorique est systématiquement inclus dans notre barre d’erreur. Dans le cas de la méthode brute, à faible résolution de grille et si le volume à mesurer est trop petit, notre incertitude est sous-estimée et n’inclut pas le volume théorique. Nous préconisons donc d’utiliser plutôt la méthode corrigée lorsque c’est possible et d’utiliser une résolution de grille élevée pour la méthode brute de sorte à ce que les résultats soient suffisamment convergés.

4.4 Convergence de la solution sur un cas réel

Étudions maintenant la convergence de notre algorithme sur un cas réel. Nous reprenons notre cas d’écaillage de Tantale qui a servi de base à nos mesures de performances dans le chapitre 3 (échantillon de 81,9 millions d’atomes initialement $100 \text{ nm} \times 100 \text{ nm} \times 150 \text{ nm}$, impacté à 1500 m/s). Le pas de temps étudié est celui ayant servi de point de départ aux mesures de performances. Pour la méthode corrigée, on fixe $\rho_0 = \rho_{Ta}$. Pour le choix du seuil de sélection des voxels ρ_s , nous prenons deux valeurs différentes, $\rho_s = 0,5\rho_{Ta}$ et $\rho_s = 0,1\rho_{Ta}$ pour étudier l’influence de ce choix sur les résultats. Dans cette section, nous désignerons respectivement ces deux choix comme : la méthode avec correction, et la méthode avec faible correction.

La figure 4.10a présente ainsi le volume total de vides détectés par chaque méthode en fonction du nombre de voxels utilisés pour mailler le système (équivalent à un tracé en fonction de la résolution de la grille). Les axes de la figure sont logarithmiques. On peut constater que la méthode avec correction a convergé vers une valeur de 10^8 \AA^3 alors que les

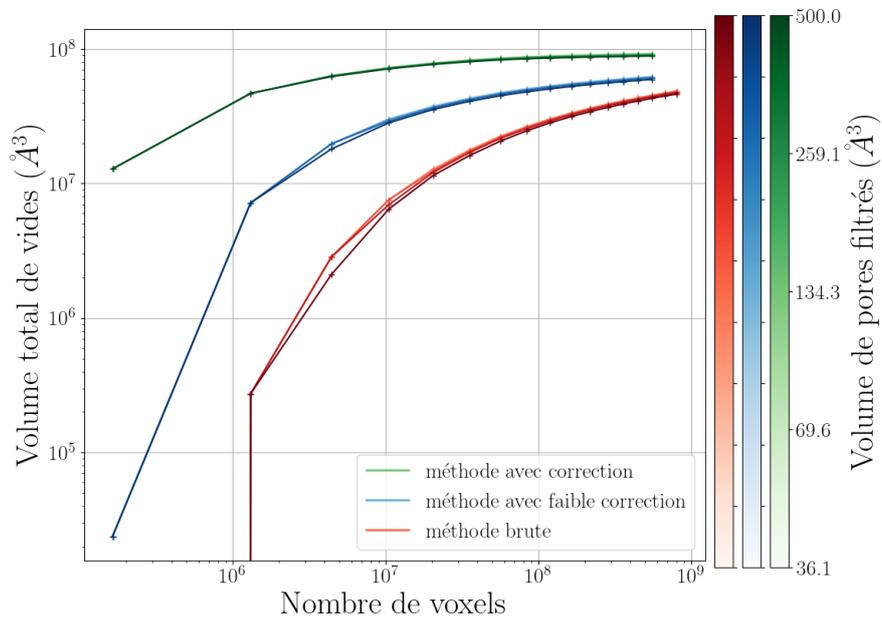
méthodes brute et avec faible correction n'ont pas encore convergé. Au vu des résultats sur la sphère parfaite (Sub-sec. 4.3.3), cette observation entre la méthode corrigée et brute était attendue. Le résultat de la méthode avec faible correction n'est pas surprenant car elle est un cas intermédiaire entre la méthode brute et la méthode avec correction. Nous pourrions donc valider la convergence de notre algorithme sur un cas réel et maintenir nos préconisations sur la méthode à utiliser. Cependant, dans un cas réel, nous n'avons pas qu'un seul pore mais une multitude. Nous devons donc aussi vérifier la convergence du nombre de pores détectés par l'algorithme en fonction de la finesse de grille et de la méthode utilisée. La figure 4.10b présente cette étude de convergence. Sur cette mesure, la convergence du nombre de pores n'est pas mise en évidence et le nombre de pores détectés par la méthode avec correction tend même à diverger. Nous avons donc a priori un problème. Mais comme nous l'avons vu précédemment, les mesures de tout petits pores ne sont pas très fiables. Nous choisissons donc de retirer de notre traitement les pores d'un volume inférieur à un seuil et de regarder l'influence de seuil sur nos résultats. Nous faisons varier ce seuil, du volume d'une maille cristalline ($36,1 \text{ \AA}^3$) à $5 \times 10^2 \text{ \AA}^3$ avec une suite géométrique de 5 points.

La figure 4.11 reprend ainsi les résultats de la figure 4.10 mais avec comme gradient de couleur le volume des pores filtrés. On constate que ce filtre n'a pas d'influence sur les courbes de volumes total mais corrige le problème du nombre de pores dès le premier niveau de filtre. Cela s'explique d'un point de vue physique par le fait que les volumes de vides inférieurs au volume d'une maille atomique peuvent difficilement être considérés comme étant des pores, et s'apparenteraient plutôt à des lacunes atomiques, ou des fluctuations locales de densité dues, par exemple, à la vibration des atomes. Par ailleurs, la précision actuelle de l'algorithme rend difficile un commentaire précis des petits volumes. Un suivi temporel des pores avec de l'ECC 4D pourrait lever les inconnus à ce propos et raffiner l'analyse. Supprimer des traitements les pores trop petits ($\sim 10^2 \text{ \AA}^3$) semble suffisamment pertinent et efficace pour fournir une bonne analyse de la simulation et en particulier celles de grande tailles qui sont notre objectif principal.

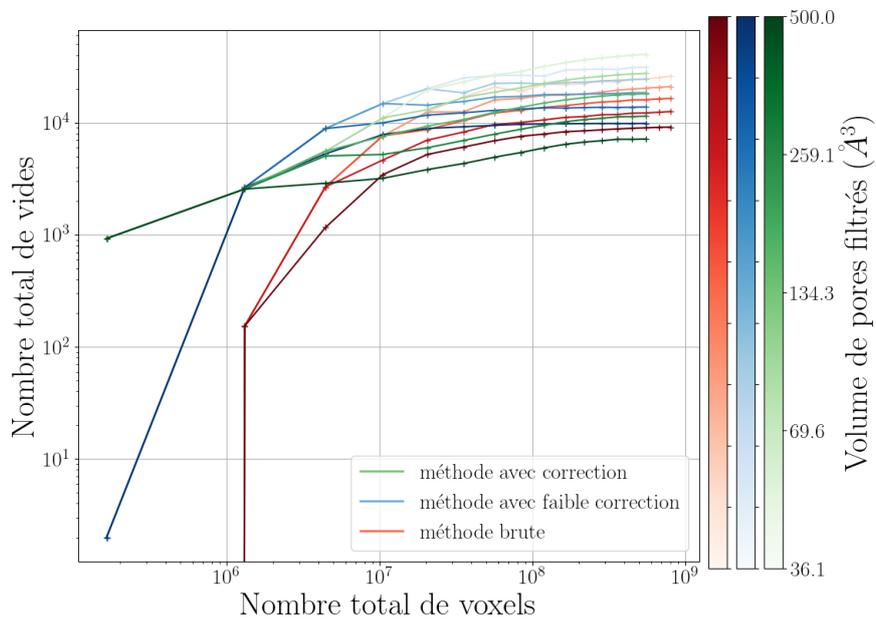
Une autre observation intéressante peut être faite sur la figure 4.11b, à petite résolution de grille, la méthode avec faible correction peut détecter plus de pores que la méthode avec correction. Cela peut sembler en contradiction avec ce que nous avons décrit dans le précédent paragraphe mais s'explique par un autre effet liée à la méthode. La méthode avec correction sélectionne plus de cellules que celle avec faible correction. Il peut arriver qu'elle sélectionne suffisamment de cellules supplémentaires entre deux pores pour les faire coalescer et ainsi diminuer le nombre de pores total. Cet effet est plus probable à faible/moyenne résolution de grille.

Finalement, nous obtenons bien une convergence de nos résultats sur un cas d'application réel, aussi bien en terme de volume, qu'en terme de nombre de pores. De plus, ayant estimé l'incertitude de mesure de volume liée à l'algorithme et ses différentes méthodes dans la section 4.3, nous pouvons l'appliquer aux données de la figure 4.12.

Nous pouvons constater que les incertitudes de mesures à faible résolution de grille et pour la méthode avec correction ne se recouvrent pas. Nous expliquons ceci par le fait que nous avons déterminé notre loi d'incertitude sur une sphère parfaite. Or, les plus gros pores de la simulation ne sont plus vraiment sphériques comme le montre la figure 4.13. Et les géométries de ces pores, pour un même volume donné, n'ont pas la même surface. Ils en ont plus qu'une sphère. Or, on sait que l'algorithme commet des erreurs sur les voxels en surface des pores. Il faudrait donc étudier d'autres formes géométriques qu'une sphère pour quantifier notre erreur et améliorer notre description de l'incertitude.



(a) Volume total de vides en fonction du nombre total de voxels



(b) Nombre total de vides en fonction du nombre total de voxels

FIGURE 4.11 – Étude de convergence de la figure 4.10 en filtrant les pores d'un volume inférieur à un seuil. Les nuances de couleurs indiquent la valeur de ce seuil.

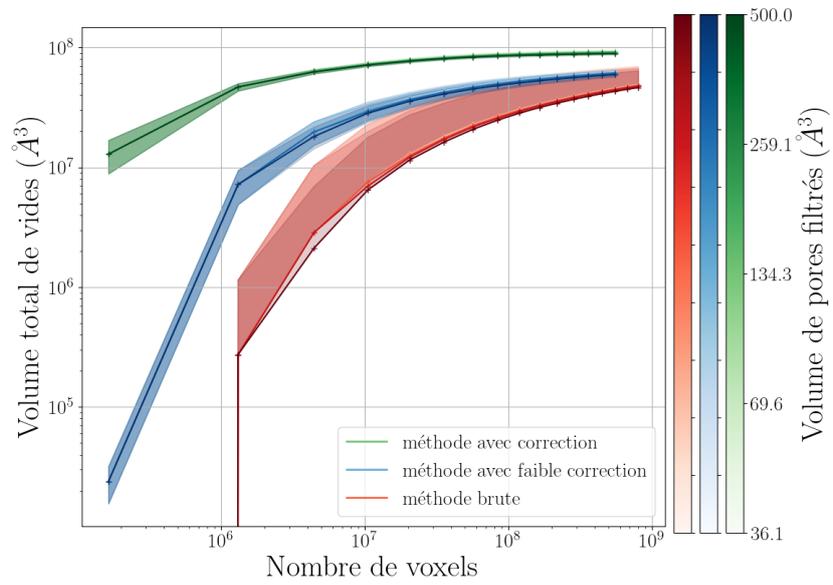


FIGURE 4.12 – Cas de la figure 4.11a avec une incertitude déterminée par l'équation (4.16).

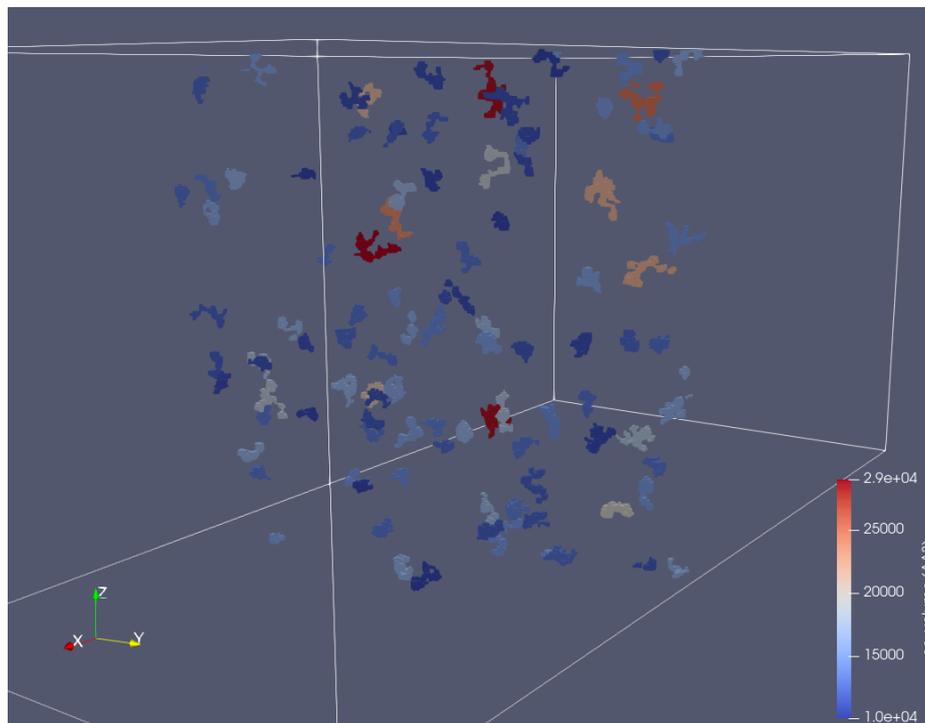


FIGURE 4.13 – Pores détectés par notre algorithme ayant un volume supérieur à 10^4Å^3 dans le cas étudié des figures 4.10 à 4.12.

Conclusion et perspectives

En conclusion, dans ce chapitre, en faisant varier la résolution de notre grille régulière, nous avons démontré la convergence de notre algorithme sur un cas d'école et un cas d'application réel. Mais cette convergence étant trop lente, elle nous a amené à améliorer les résultats de notre analyse en proposant une méthode de correction fondée sur des considérations physiques. Nous avons ensuite mis en place une méthodologie pour déterminer l'erreur de mesure liée à notre algorithme. Puis nous avons utilisé cette méthodologie sur le cas d'une sphère parfaite. Nous nous sommes ensuite servis de la loi d'incertitude que nous avons établie sur le cas d'application réel.

Cela nous a permis de mettre en évidence plusieurs points. Premièrement, l'utilisation de la méthode avec correction est fortement recommandée lorsque c'est possible. Deuxièmement, plus le seuil de sélection des cellules est haut et la résolution de grille élevée, plus l'algorithme a tendance à inclure des faux positifs d'un ou deux voxels. Ces faux positifs n'influencent pas les résultats de volume total mais ils faussent les résultats sur le nombre de pores détectés. Il est alors crucial de les exclure des résultats, en retirant des traitements les vides trop petits pour être qualifiés de pores, et possiblement dû aux fortes vibrations atomiques liées à la température élevée du système. Troisièmement, l'incertitude de mesure que nous avons établie dans le cas d'une sphère parfaite sous estime certainement la véritable incertitude de mesure dans un cas d'application réel. Il faudrait donc étudier d'autres formes géométriques pour améliorer notre description de l'erreur.

Chapitre 5

Applications physiques

L'application de notre algorithme d'analyse in-situ de simulation de DM capable de traiter des simulations impliquant plusieurs milliards d'atomes est présentée dans ce chapitre. L'objectif sera d'une part de démontrer les capacités de l'algorithme et d'autre part de fournir de nouvelles informations pour l'étude des différents phénomènes que nous allons traiter. Ces phénomènes sont ceux que nous avons introduits dans le chapitre 1 et que nous avons déjà pu brièvement rencontrer pour le cas de l'écaillage (Chap. 3 et 4). La première partie de ce chapitre est consacrée aux phénomènes d'éjection de matière et la seconde partie à l'écaillage. Dans la partie dédiée aux phénomènes d'éjection de matière, nous introduisons l'article réalisé par les physiciens du laboratoire sur le *micro-jetting*, puis nous présentons une étude que nous avons réalisée sur des simulations de *splashing*. La partie dédiée à l'endommagement s'organise en 5 sections. La première présente deux façons de générer de l'écaillage en DM. La deuxième présente les méthodes que nous utilisons pour obtenir les quantités macroscopiques qui décrivent la simulation et qui seront étudiées par la suite. La troisième présente l'étude que nous avons réalisée sur le tantale. La quatrième compare les résultats obtenus pour le tantale avec ceux que nous avons obtenus pour les deux versions (fragile et ductile) du potentiel modèle implémenté au chapitre 2. La dernière section présente une nouvelle perspective d'application de notre algorithme.

Éjection de matière

5.1 Micro-jetting

Comme introduit dans le chapitre 1, le *micro-jetting* est un phénomène d'éjection de matière qui peut se produire lorsqu'une onde de choc se réfléchit sur une surface libre qui présente au moins un défaut de planéité. La réflexion du choc sur la surface libre conduit à une mise en vitesse très rapide de cette dernière, et engendre une concentration de contrainte au niveau du défaut, ce qui peut produire un jet de matière qui se fragmente ensuite en micro débris. Ces débris sont minuscules mais très rapides et peuvent causer, dans le cas d'expériences, d'importants dégâts sur les équipements présents au sein de l'installation. Il est donc important de comprendre ce mécanisme pour prévenir son apparition ou prendre des mesures pour limiter son action.

Grâce à l'algorithme que nous avons développé et implémenté (Chap. 3), les physiciens du laboratoire ont pu analyser une simulation de *micro-jetting* d'un échantillon d'étain de

$2 \mu\text{m}$ de long et de section $0,162 \text{ nm} \times 0,162 \text{ nm}$ (environ 2 milliards d'atomes) contenant un défaut triangulaire. Ils ont notamment pu extraire la masse et la vitesse des agrégats de matières au cours du temps et comparer ces résultats à ceux fournis par deux modèles développés au sein du laboratoire : BMPT-2 [30] et FPZ [103]. Ils ont aussi pu extraire le volume de chacun des agrégats et calculer la distribution de ces volumes. Ceci a mis en évidence au moins deux processus de fragmentation différents. Ces résultats sont présentés dans la publication [29] fournie en annexe B. La figure 5.1 est extraite de l'article et présente deux images de la simulation étudiée.

Dans cette publication, les auteurs ont aussi comparé leurs modèles à des expériences. Pour calculer la quantité de masse éjectée durant ces expériences, le dispositif expérimental utilisé est une fenêtre d'Asay [3]. Ce dispositif consiste à placer une fine couche de métal en face du jet de matière et suffisamment loin pour considérer que l'ensemble des processus de fragmentation ait eu lieu. Lorsque les particules entrent en contact avec le film de métal, ce dernier est mis en mouvement et sa vitesse est calculée à l'aide d'un VISAR¹ ou d'un PDV² (effet Doppler). Grâce à cette vitesse et en utilisant certaines hypothèses, on peut calculer la masse éjectée par conservation de l'énergie. Une de ces hypothèses est notamment qu'il n'y pas de *back-splashing*, c'est-à-dire, que la collision entre un agrégat et le film ne génère pas d'éjection de matière et donc que l'ensemble de l'énergie cinétique de la particule est transmise au film. Pour étudier cette hypothèse on peut alors réaliser des simulations de *splashing* en DM, cela fera l'objet de la section suivante.

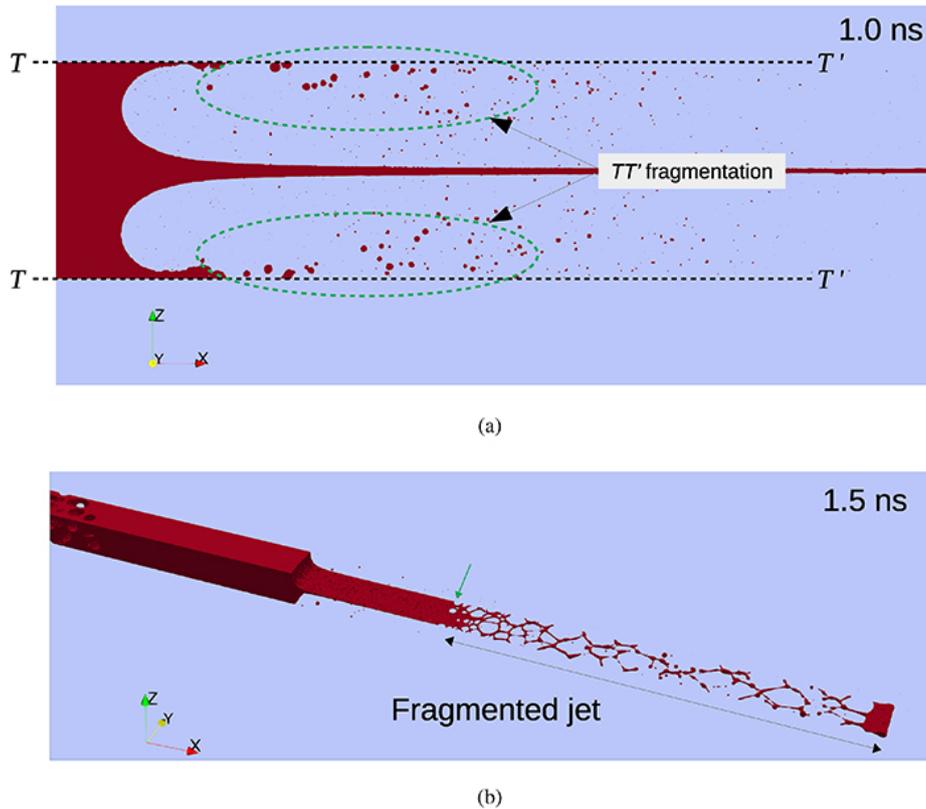


FIGURE 5.1 – Figure extraite de [29], microjet d'étain liquide à deux instants de la simulation

1. Velocity Interferometer System for Any Reflector
2. Photonic Doppler Velocimetry

5.2 Splashing

5.2.1 Description du phénomène

Le *splashing* est un phénomène d'éjection de matière qui peut se produire lorsqu'un agrégat de matière vient impacter une surface. Si cet impact est suffisamment intense alors une couronne de matière peut être éjectée dans le sens opposé à celui de l'impact initial. De nombreux phénomènes physiques à différentes échelles de tailles peuvent s'apparenter à du splashing, de l'impact de météorite, aux impacts de gouttes micrométriques à haute vitesse. La compréhension du phénomène pour des impacts d'objets de l'ordre du centimètre répond principalement des enjeux de sécurité, que ce soit dans le domaine de la défense pour la conception de blindage ou dans le domaine de l'aérospatial pour la protection des satellites (Chap. 1). Pour des impacts de particules micrométriques, la compréhension de ce phénomène est essentielle pour le procédé de fabrication additive qu'est le *cold spray*. Ce procédé consiste à projeter à l'aide d'un gaz des microparticules principalement métalliques à haute vitesse sur une surface. Une des autres applications d'intérêt pour l'étude de ce phénomène à l'échelle micrométrique est de savoir si l'hypothèse suivante est valide : l'ensemble de l'énergie cinétique de la particule incidente est transmise à la cible, utilisée pour calculer la masse des particules éjectées par micro-jetting dans un dispositif de type fenêtre d'Asay.

Pour étudier cette hypothèse, nous avons appliqué notre algorithme en post-traitement à des simulations précédemment réalisées au laboratoire principalement pour étudier la formation du jet de matière éjectée sous forme de couronne. Les simulations que nous avons traitées sont issues de l'article [32] et consistent en l'étude systématique de l'impact d'une bille de 50 nm de diamètre d'étain sur une plaque de section 600 nm × 600 nm et d'une épaisseur de 100 nm (environ 1,2 milliards d'atomes) à différentes vitesses. Le potentiel utilisé pour ces simulations est un potentiel MEAM³ [104], ce type de potentiel est environ dix fois plus coûteux en temps de calcul que ceux que nous avons implémentés dans le chapitre 2 mais est nécessaire pour décrire l'étain convenablement. Ces simulations ayant nécessité plusieurs dizaines de millions d'heures de calculs au total, nous avons uniquement post-traité les quelques centaines de pas de temps qui ont été sauvegardés. Ce post-traitement a été réalisé sur 128 noeuds du calculateur CEA-HF [71] et a nécessité environ une dizaine d'heures de calculs "utilisateur" soit environ deux cents milliers d'heures de calculs CPU, dont à peu près 70% du temps a consisté à charger les fichiers en entrée.

Nous présentons ici uniquement la simulation ayant généré le plus d'éjection de matière. La vitesse d'impact de la sphère est de 2000 m/s, la vitesse la plus élevée de cette étude. Nous présentons une visualisation du dernier pas de temps de cette simulation dans la figure 5.2. La couleur des agrégats représente leur volume. L'amas principal formé par la cible et la couronne non fragmentée est exclu du traitement. Nous discutons de quelques propriétés de ces agrégats dans la sous section suivante.

3. MEAM : Modified Embedded Atom Method

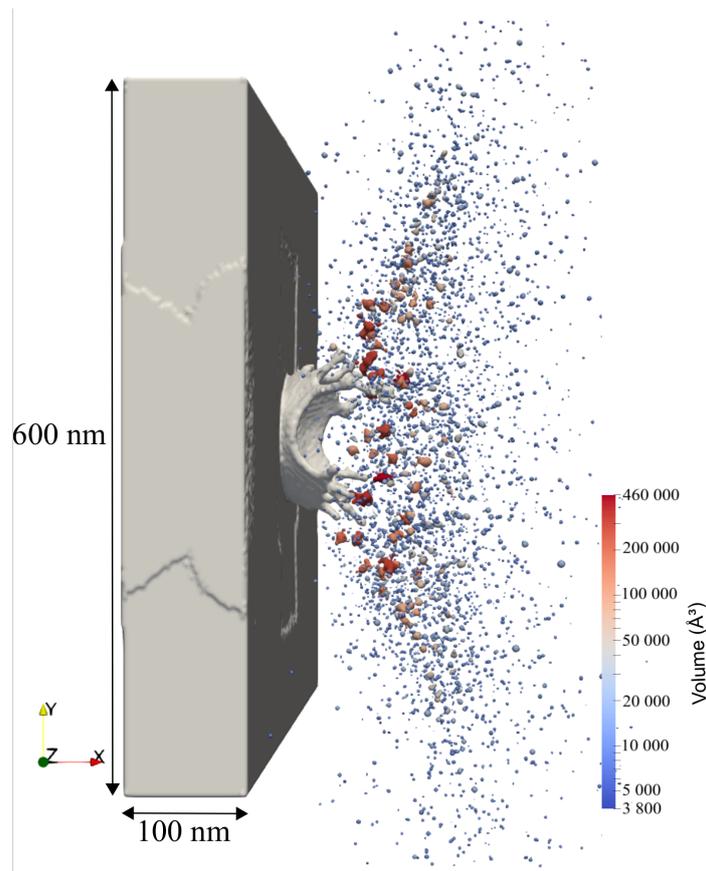


FIGURE 5.2 – Simulation de *splashing*, les agrégats de matière sont colorés en fonction de leur volume (cible exclue) avec une échelle logarithmique.

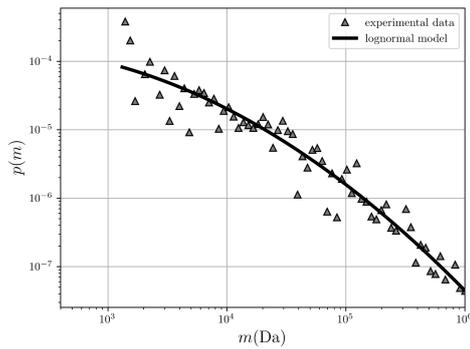
5.2.2 Caractérisation statistique des propriétés des agrégats

Pour l'analyse, la projection des atomes a été réalisée sur une grille régulière contenant 29,41 milliards de cellules cubique d'arête égale à 4 Å avec un seuil de sélection des cellules égal à 70% de la densité de l'étain à condition ambiante.

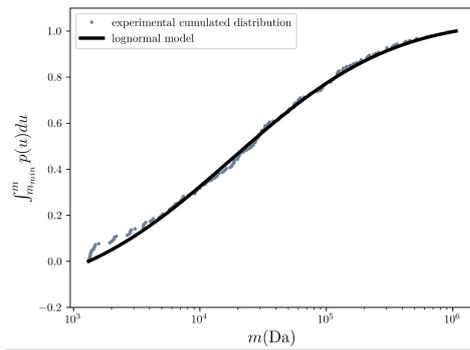
La figure 5.3 présente en (a) la distribution de masse des agrégats sur un diagramme log-log et couvre 3 ordres de grandeurs. En (b) la cumulée de cette distribution de masse est représentée, sur laquelle nous avons ajusté la cumulée d'une loi lognormale. Cette loi n'est pas rejetée par nos données et est souvent associée à des phénomènes statistiques multiplicatifs [105-107].

La sous figure (c) présente la distribution de vitesse des particules, dont la cumulée est présentée en (d) et sur laquelle nous avons ajusté la cumulée d'une loi normale. Cette loi est en parfait accord avec nos données. On peut constater que la moyenne de cette distribution est centrée sur une valeur inférieure à la vitesse d'impact initiale ($15 \text{ \AA/ps} \Leftrightarrow 1500 \text{ m/s}$) mais que certaines particules ont une vitesse supérieure à celle ci.

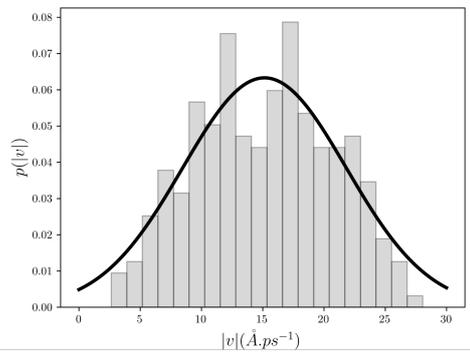
Finalement, nous présentons dans la figure 5.4 la répartition de l'énergie cinétique des agrégats de matière en fonction de leur masse et du carré de leur vitesse. L'énergie cinétique est exprimée en fraction de l'énergie cinétique incidente. On peut constater que l'énergie cinétique des agrégats est majoritairement gouvernée par leur masse plutôt que par leur vitesse. Les agrégats les plus rapides ont une énergie cinétique d'un ordre de grandeur



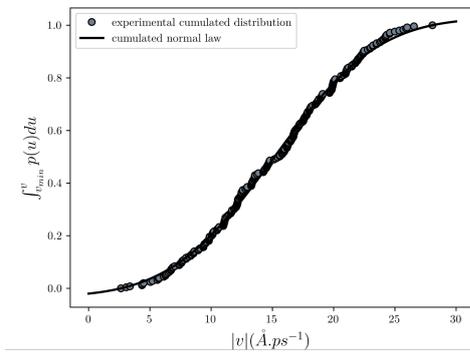
(a) Distribution de masse



(b) Cumulé de la distribution de masse



(c) Distribution de vitesse



(d) Cumulé de la distribution de vitesse

FIGURE 5.3 – Distributions des agrégats de la simulation de la figure 5.2

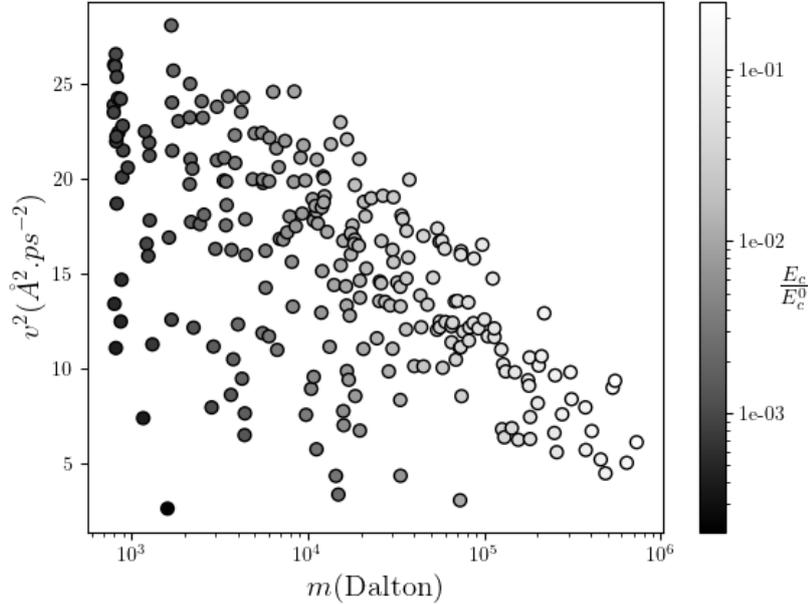


FIGURE 5.4 – Répartition de l'énergie cinétique des agrégats en fonction de leur masse et de leur vitesse. L'énergie cinétique est exprimée en fonction de l'énergie cinétique du projectile incident.

inférieur aux plus massifs. De plus l'ensemble de l'énergie cinétique de tous ces agrégats représente 3% de l'énergie cinétique de la bille avant impact. On peut donc conclure que dans ce cas, l'hypothèse posée pour les fenêtres d'Asay (toute l'énergie cinétique de la particule est transmise à la fenêtre) est une approximation raisonnable du comportement de la matière, l'erreur commise étant de l'ordre de 3%.

5.3 Endommagement

Description du phénomène

Le phénomène d'endommagement qui va nous intéresser dans cette partie est l'écaillage. Ce phénomène bien qu'étant macroscopique (Fig. 5.5, bas de l'image) a une origine microscopique comme l'illustre la figure 5.6. Cette origine est la nucléation de micro pores suite à une mise en tension suffisamment intense de la matière, qui peuvent ensuite croître et coalescer. Cette coalescence pouvant aller jusqu'à la fracturation complète du matériau. La figure 5.6 présente deux images réalisées au laboratoire avec un microscope électronique à balayage (MEB) pour observer une surface de tantale écaillée par tir laser. L'image de gauche a un champ de 2,92 mm et l'image de droite est la même surface avec un champ de l'ordre de 50 μm . On peut clairement y observer les cupules formées par la coalescence des pores.

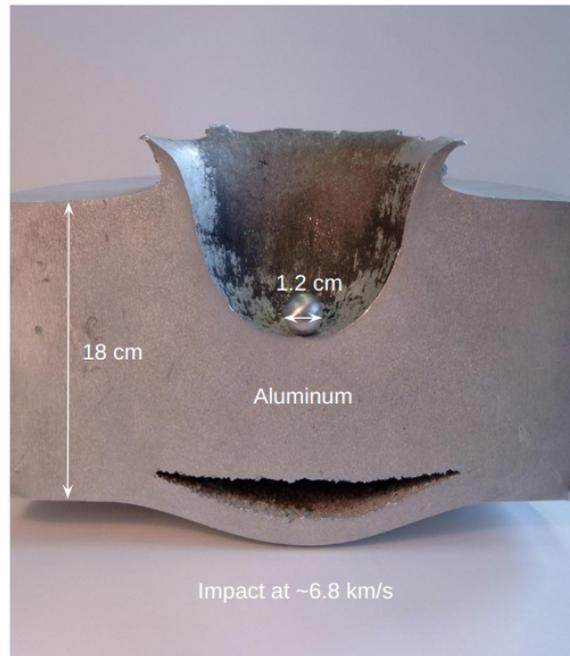


FIGURE 5.5 – Image d'un impact hypervélocité réalisé par l'ESA que nous avons modifié pour faire apparaître les caractéristiques de l'essai. La coupe a été réalisée après l'essai et une sphère d'aluminium similaire à celle ayant impacté la plaque a été placée pour servir de référence à la photo. Crédit : ©ESA

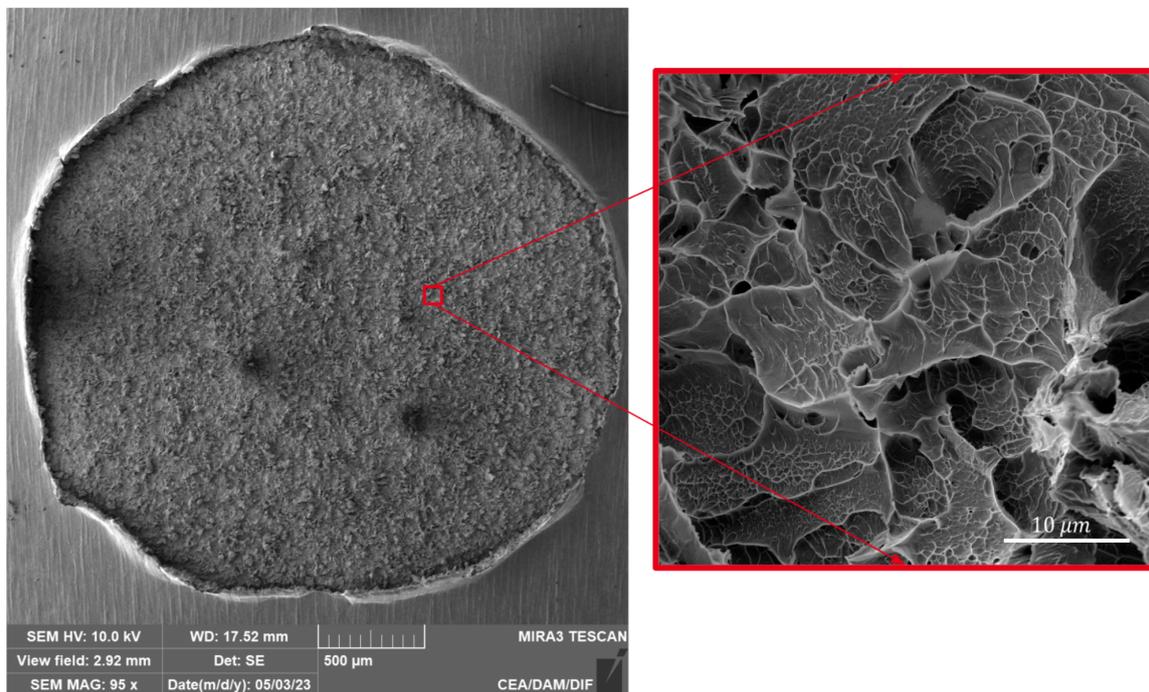
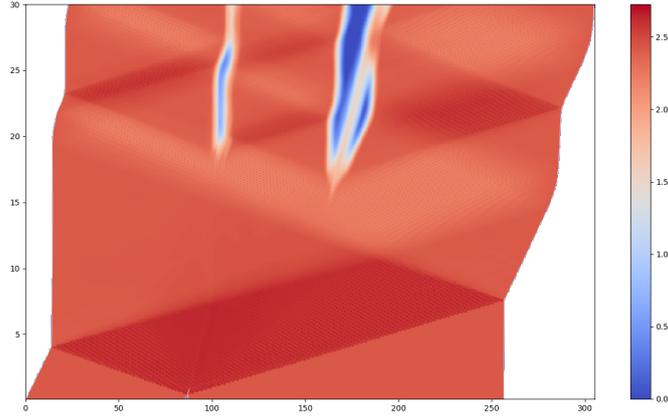
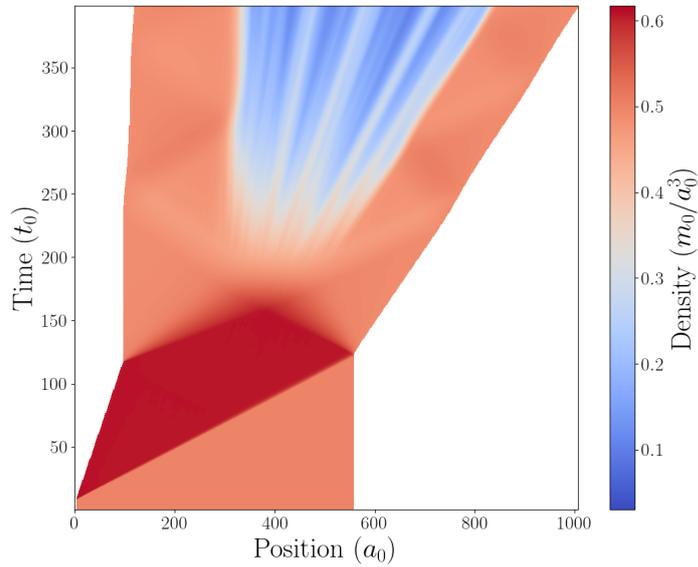


FIGURE 5.6 – Images MEB d'une surface de tantale écaillée par tir laser. Expérience réalisée au laboratoire, et image réalisée par N. Bruzy.

Pour générer ce phénomène en DM, deux techniques principales peuvent être utilisées. La première consiste à envoyer un projectile sur une cible à la vitesse désirée. Ainsi, à l'impact, une onde de choc se propage dans la cible et dans le projectile jusqu'à atteindre les deux surfaces libres sur lesquelles elles se réfléchissent en onde de détente. En prenant une cible deux fois plus épaisse que le projectile, on s'assure que les ondes de détente se croisent approximativement au milieu de la cible (cela suppose que les ondes de détente se propagent à la même vitesse que l'onde de choc). La figure 5.7a présente un diagramme de marche ou diagramme x-t d'une simulation de ce type que nous avons réalisée. Ce type de diagramme illustre en abscisse la position d'une tranche de matière dans la simulation dans la direction de propagation du choc, et en ordonnée le temps. Le champ de couleur peut ensuite représenter différentes quantités physiques associée aux tranches de matière. Dans le cas présent le champ de densité est représenté.



(a) Impact par projectile



(b) Impact par piston

FIGURE 5.7 – Diagrammes x-t de deux simulations réalisées avec le potentiel fragile. Le champ de densité est représenté.

L'autre technique principale utilisée pour simuler l'écaillage consiste à impacter la matière avec un piston que l'on déplace à une vitesse souhaitée. Le piston est modélisé par un potentiel purement répulsif. Tant que le piston est présent dans la simulation, le choc est soutenu et lorsqu'on le retire, une onde de détente se propage depuis la surface qu'il laisse libre. Si on retire le piston au moment où le choc qu'il a initié se reflète sur la surface libre, alors les ondes de détente se croiseront au milieu de l'échantillon. L'avantage d'utiliser cette technique est de diminuer le coût de calcul par rapport à la technique précédente car on économise les atomes utilisés dans le projectile. Cependant, cela suppose de connaître le moment où le choc se réfléchit sur la surface libre en face arrière. Pour cela, on peut soit le détecter en sondant périodiquement la matière, soit en utilisant la relation d'Hugoniot qui lie la vitesse de propagation du choc v_s à la vitesse de déplacement des particules v_p (équivalent à la

vitesse du piston) pour la matière considérée. Souvent cette relation est de la forme :

$$v_s = c_0 + b v_p \quad (5.1)$$

avec c_0 et b des coefficients ajustés empiriquement selon le matériau. L'ouvrage [108] répertorie les données acquises par le LASL pour plusieurs centaines de matériaux grâce à plus de 5000 expériences, pour le tantale, $c_0 = 3,43 \text{ km/s}$ et $b = 1,19$. Pour le potentiel modèle que nous allons utiliser, si l'on souhaite utiliser cette technique, il faudra déterminer les valeurs de ces coefficients pour sa version ductile et fragile. Pour nos études paramétriques des sections suivantes, nous utiliserons l'impact par piston.

À partir des informations contenues dans ces diagrammes nous allons pouvoir calculer les variables macroscopiques qui caractérisent notre chargement et l'endommagement résultant. La section suivante présente comment nous obtenons ces grandeurs.

5.4 Analyse macroscopique

Les deux paramètres caractéristiques de l'écaillage essentiellement étudiés dans la littérature sont la tension d'écaillage σ_{spall} et le taux de déformation $\dot{\epsilon}$. Expérimentalement, ces paramètres sont calculés à partir de la vitesse de la surface libre en face arrière obtenue avec un vélocimètre (VISAR, PDV) et en posant certaines hypothèses. Les hypothèses principalement utilisées sont celles de l'approximation acoustique (CITE), à partir laquelle on peut calculer la tension d'écaillage et le taux de déformation que nous noterons respectivement σ_{fsv} et $\dot{\epsilon}_{fsv}$. Cette méthode fournit les expressions suivantes :

$$\sigma_{fsv} = \frac{1}{2} \rho_b c_b \Delta v_{fs} \quad (5.2)$$

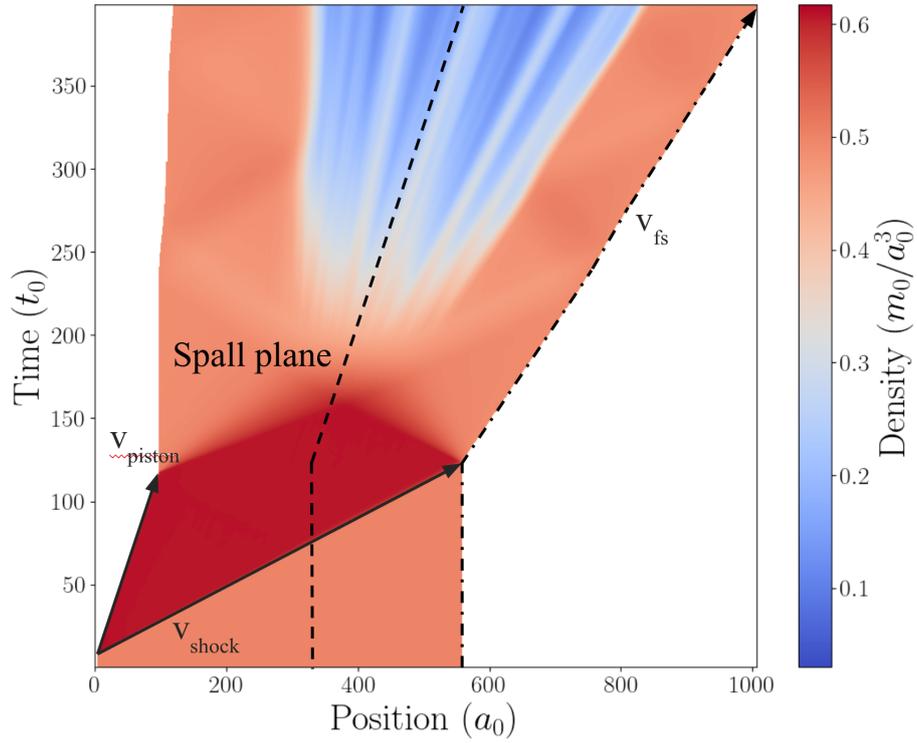
$$\dot{\epsilon}_{fsv} = \frac{1}{2} \frac{\Delta v_{fs}}{\Delta t_{fs}} \frac{1}{c_b} \quad (5.3)$$

avec ρ_b la densité du matériau *bulk*, c_b la vitesse du son dans le matériau *bulk*, Δv_{fs} la différence de vitesse entre la vitesse maximale atteinte en face arrière v_{max} et la vitesse minimale v_{min} atteinte avant le prochain rebond de vitesse, et Δt_{fs} le temps écoulé entre le point de vitesse minimum et maximum. Dans nos simulations, le signal de vitesse de la surface libre est obtenu à partir de la vitesse des particules de la dernière tranche de matière. Elle est représentée sur le diagramme x-t de la figure 5.8a. Le signal correspondant est tracé dans la figure 5.8b et il est annoté avec v_{max} , v_{min} , t_{max} , t_{min} . L'approximation acoustique suppose la contribution non linéaire de la compressibilité de la matière négligeable et l'existence d'une seule vitesse du son dans le matériau. Cependant, quand la valeur de la tension d'écaillage est grande, la contribution non linéaire de la compressibilité de la matière est à prendre en compte. Dans ce but, Kanel et al. [109] propose de prendre en compte cet effet avec l'hugoniot (Eq. 5.1) extrapolé en tension. Cela fournit une nouvelle expression de la tension d'écaillage noté σ_{kanel} par la suite :

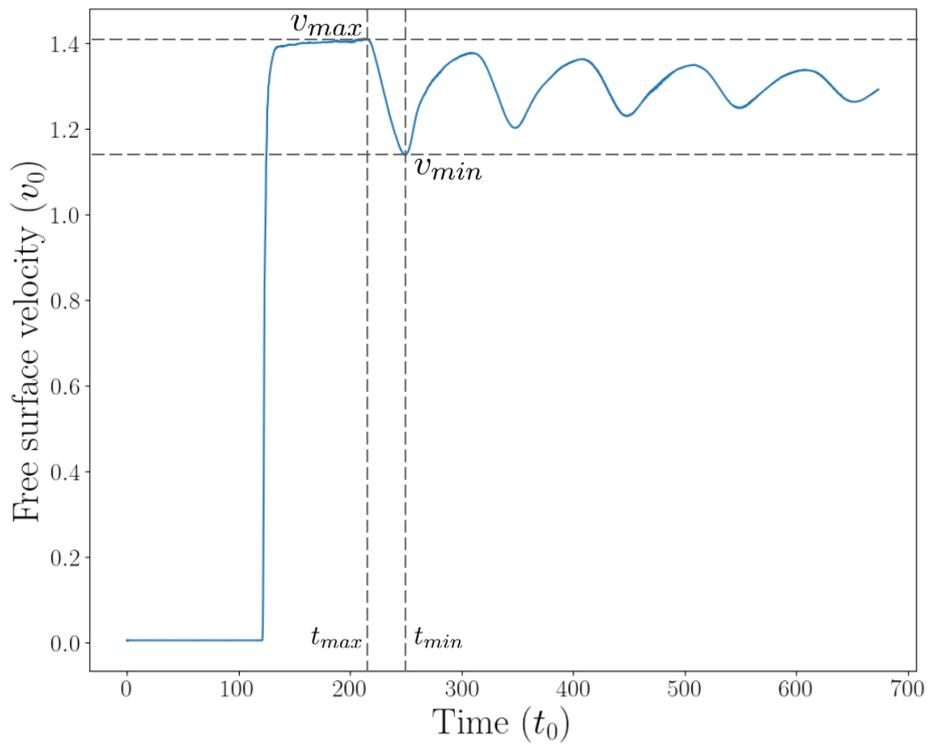
$$\sigma_{kanel} = \frac{1}{2} \rho_b (c_0 - b \Delta v_{fs} / 2) \Delta v_{fs} \quad (5.4)$$

en suivant la même approche pour le taux de déformation on peut alors l'écrire comme :

$$\dot{\epsilon}_{kanel} = \frac{1}{2} \frac{\Delta v_{fs}}{\Delta t_{fs}} \frac{1}{(c_0 - b \Delta v_{fs} / 2)} \quad (5.5)$$



(a) Diagramme x-t de la figure 5.7b annoté



(b) Signal de vitesse en surface libre.

D'autres modèles existent pour estimer la valeur de la tension d'écaillage et du taux de déformation à partir de la vitesse de surface libre mais la plus utilisée reste l'approximation acoustique (Eq. 5.2-5.3). Nous nous limiterons donc à l'utilisation de ces deux formules pour calculer ces paramètres à partir du signal de vitesse de la surface libre.

Cependant en DM, il est possible de calculer ces paramètres en tout point de l'espace de simulation, et notamment dans le plan d'écaillage. Ayant fait en sorte de relâcher le piston au moment où le choc se réfléchit sur la surface libre, les ondes de détente se croisent au milieu de l'échantillon, nous définissons donc le plan d'écaillage comme étant le plan à mi-distance des frontières de l'échantillon. On peut ainsi suivre l'évolution de la pression et du taux de déformation dans ce plan au cours du temps. La pression est calculée dans une tranche avec l'équation 3.15 du chapitre 3 et le taux de déformation est calculé comme étant :

$$\dot{\epsilon} \approx \frac{\delta v_x}{\delta x} \quad (5.6)$$

avec x la position dans la direction de choc et v_x la vitesse dans cette direction. Le plan suivi au cours du temps est représenté en pointillé dans le diagramme x - t de la figure 5.8a. La pression est tracée en bleu dans la figure 5.9a, tout comme le taux de déformation dans la figure 5.9b. Dans ces deux figures, nous ajoutons les courbes correspondant au nombre de pores détectés par notre algorithme dans la simulation (en violet) et la porosité calculée dans la tranche (en rouge). La porosité étant définie comme la fraction de cellules vides dans la tranche sur le nombre de cellules qu'elle contient. Les courbes sont annotées des mesures de la tension d'écaillage σ_{spall} au moment où les premiers pores apparaissent, de la tension maximum subie par la tranche σ_{max} , du taux de déformation maximum subit en traction par la tranche $\dot{\epsilon}_{xx}$ et du taux de déformation au moment où la tension est maximale $\dot{\epsilon}_{\sigma_{max}}$.

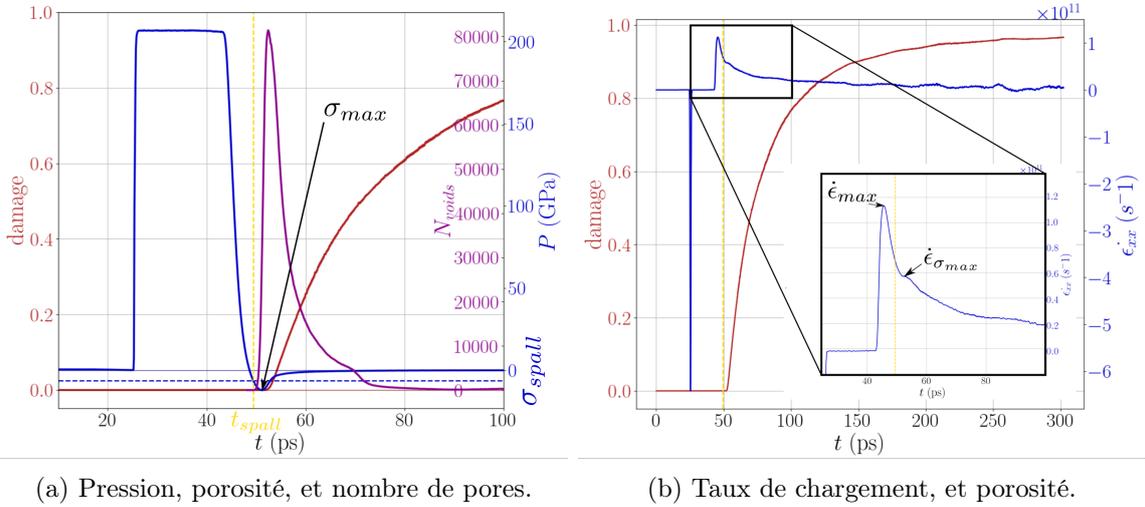


FIGURE 5.9 – Données extraites du plan d'écaillage définie comme dans la figure 5.7b. Simulation de tantale impacté à v_0 . Les figures sont annotées des grandeurs qui en sont extraites.

Le cas présenté est une simulation d'un échantillon de tantale de section $100 \text{ nm} \times 100 \text{ nm}$ et de longueur 150 nm (83 millions d'atomes) impacté à la vitesse v_0 ($2078,2 \text{ m/s}$). Nous pouvons observer plusieurs choses. Premièrement, les premiers pores apparaissent dans la simulation avant que la tension d'écaillage maximum σ_{max} soit atteinte, contrairement aux

hypothèses de l'approximation acoustique qui supposent qu'ils apparaissent à cet instant. Deuxièmement, le moment où σ_{max} est atteint et où la pression commence à retourner à zéro, correspond au moment où les pores coalescent suffisamment pour contrebalancer la nucléation et donc faire diminuer le nombre de pores. Troisièmement, on peut constater que la porosité commence à croître de façon notable après que la coalescence commence à dominer et que la pression soit retournée à zéro. Seul un modèle d'endommagement inertiel comme [110] pourrait reproduire un tel comportement.

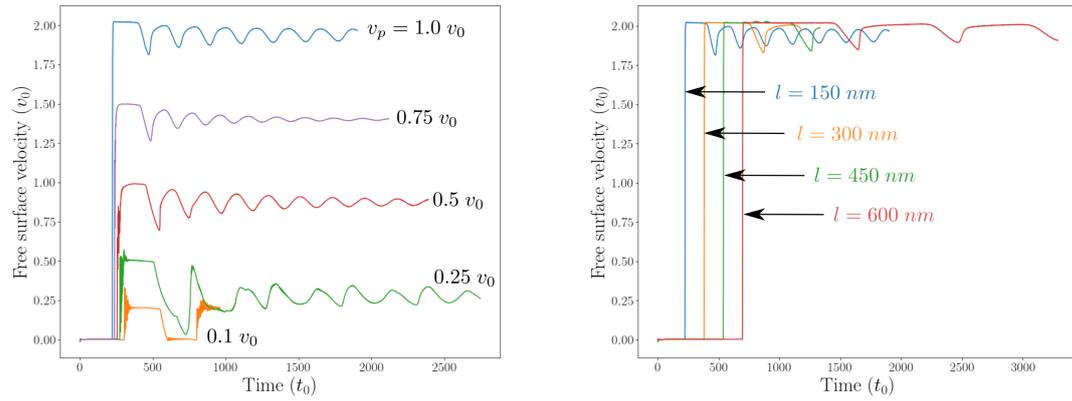
Dans les sections suivantes, nous allons exploiter les mesures que nous venons de décrire pour caractériser les simulations de tantale (Sec. 5.5) et les comparer aux simulations utilisant le potentiel modèle (Sec. 5.6).

5.5 Analyse paramétrique tantale

Cette section se décompose en deux sous-sections. La première sous-section présentera une étude dans laquelle, nous allons d'une part faire varier la vitesse d'impact sur un échantillon de dimension initiale constante pour majoritairement faire varier la tension d'écaillage, et d'autre part, faire varier la longueur initiale de l'échantillon à vitesse d'impact constante pour modifier le taux de déformation. L'objectif sera de vérifier si les valeurs de ces paramètres mesurés à partir de la vitesse en surface libre sont comparables à celles calculées dans le plan d'écaillage. La deuxième sous section présentera une analyse de la distribution en volume des pores et leur distribution spatiale pour une simulation de 8,3 milliards d'atomes.

5.5.1 Comparaison entre les mesures macroscopique par vitesse de surface libre et les calculs dans le plan d'écaillage

Les dimensions initiales de l'échantillon que nous avons utilisé pour plusieurs vitesses d'impact est de section $100 \text{ nm} \times 100 \text{ nm}$ et de longueur 150 nm (83 millions d'atomes). Nous avons impacté notre échantillon à six vitesses différentes : $2 v_0$, $1 v_0$, $0,75 v_0$, $0,5 v_0$, $0,25 v_0$, et $0,1 v_0$, avec v_0 la vitesse d'adimensionnement définie au chapitre 2, pour le tantale, $v_0 = 2078,2 \text{ m/s}$. Les signaux de vitesse en surface libre pour ces expériences sont présentés dans la figure 5.10a sauf pour l'impact à $2 v_0$ qui a fait fondre l'échantillon (le signal de vitesse restant constant une fois la surface mise en vitesse). Nous excluons donc cette simulation de notre étude. Nous pouvons aussi constater que la simulation impactée à $0,1 v_0$ ne génère pas d'écaillage car la vitesse de la surface libre retourne entièrement à zéro après la mise en vitesse. Cette simulation est donc aussi exclue de notre étude. Pour les expériences où nous avons fait varier la longueur initiale de l'échantillon, la vitesse d'impact est constante, égale à v_0 , la section reste inchangée, et les longueurs testées sont : 150 nm , 300 nm , 450 nm , et 600 nm . Les signaux de vitesse de la surface libre de ces simulations sont présentés dans la figure 5.10b.



(a) Variation de la vitesse d'impact

(b) Variation de la longueur de l'échantillon

FIGURE 5.10 – Signaux de vitesse de surface libre des simulations de tantale

À partir de ces mesures, nous avons calculé les σ_{fsv} et $\dot{\epsilon}_{fsv}$ de chaque expérience comme présenté dans la section précédente (Sec. 5.4). Nous comparons ces résultats obtenus avec ceux de la littérature pour le tantale dans la figure 5.11. Cette figure présente la tension d'écaillage en fonction du taux de déformation avec des axes logarithmiques. Cette figure est extraite du *supplementary materials* de l'article [48] sur laquelle nous avons superposé nos points. Cette figure inclut à la fois des résultats expérimentaux, et de simulations de DM aussi bien pour des échantillons monocristallins, que des échantillons polycristallins ou à microstructure inconnue. Nous pouvons constater que nos mesures sont cohérentes avec celles de la littérature.

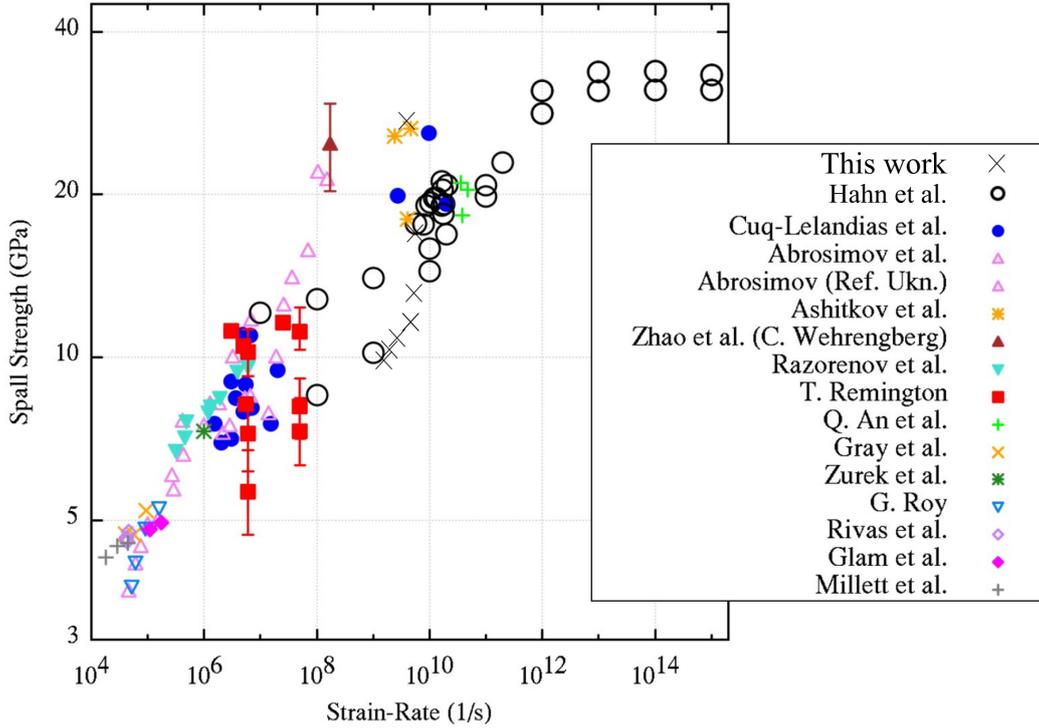


FIGURE 5.11 – Figure extraire du *supplementary materials* de l'article [48] qui présente la tension d'écaillage en fonction du taux de déformation du tantale de la littérature, et sur laquelle nous avons placé nos points (croix noire).

Nous voulons maintenant vérifier que ces mesures de tension d'écaillage et de taux de déformation sont aussi cohérentes avec les calculs réalisés dans le plan d'écaillage. Ceci validerait l'utilisation de l'approche acoustique pour mesurer ces grandeurs. Nous avons donc tracé dans la figure 5.12 nos points de σ_{max} , σ_{spall} , et σ_{kanel} en fonction du σ_{fsv} . La courbe en pointillée noire correspond à ce que l'on obtiendrait si ces tensions étaient parfaitement équivalentes.

Nous pouvons constater que la tension maximale σ_{max} est cohérente avec σ_{fsv} à faibles valeurs de tension mais s'éloigne pour de plus grandes valeurs de tension. Ce comportement était attendu pour l'approximation acoustique et l'utilisation de l'hugoniot introduite par Kanel et al. [109] vient en effet corriger cet écart. Cependant, si les calculs de σ_{kanel} et σ_{fsv} sont globalement cohérents avec le σ_{max} car ils supposent que l'écaillage a lieu à cette valeur de tension, on peut constater qu'ils ne sont pas en accord avec la valeur réelle de tension au moment de l'écaillage σ_{spall} . Cet écart reste néanmoins contenu à un facteur inférieur à 2, en particulier si on compare les valeurs de σ_{kanel} avec celles de σ_{spall} .

Regardons maintenant ce qu'il en est pour le taux de déformation. La figure 5.13 présente les données que nous avons obtenues pour les trois mesures $\dot{\epsilon}_{max}$, $\dot{\epsilon}_{\sigma_{max}}$, et $\dot{\epsilon}_{kanel}$ en comparaison à la mesure $\dot{\epsilon}_{fsv}$. Dans cette figure, les taux de déformations sont exprimés en ratio du taux de déformation $\dot{\epsilon}_{fsv}$. Nous pouvons déjà constater que le $\dot{\epsilon}_{kanel}$ est quasiment strictement équivalent au $\dot{\epsilon}_{fsv}$. Ensuite, un autre constat que nous pouvons faire est que le $\dot{\epsilon}_{max}$ est quasiment toujours le double du $\dot{\epsilon}_{\sigma_{max}}$. Mais nous pouvons surtout observer qu'il y a un ordre de grandeur d'écart entre le taux de déformation vu par la matière lorsque la tension est maximale et le taux de déformation calculé avec la vitesse de la surface libre,

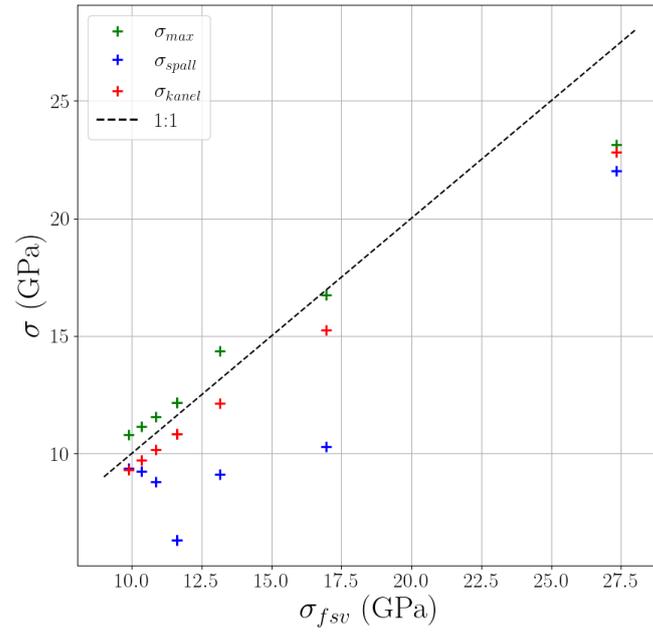


FIGURE 5.12 – Comparaison des valeurs de tension d'écaillage calculées selon différentes hypothèses

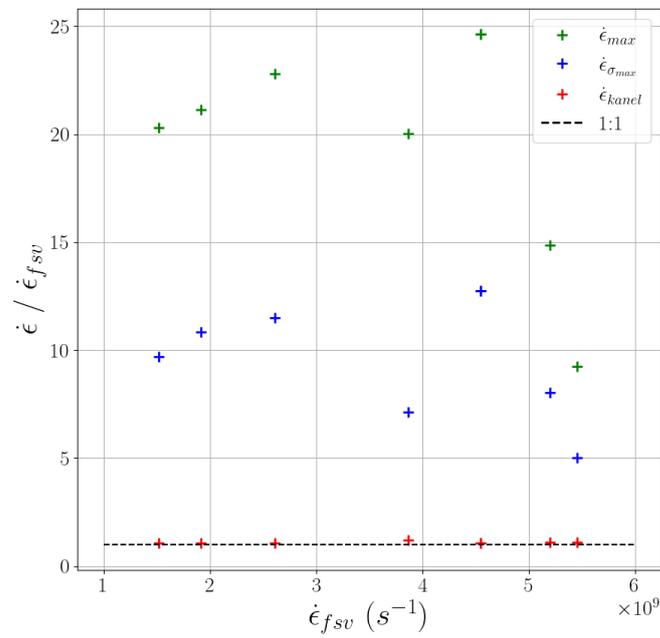


FIGURE 5.13 – Comparaison des valeurs de taux de chargement calculées selon différentes hypothèses

résultat déjà observé dans la littérature [48]. La mesure du taux de déformation se basant sur les vitesses de surfaces libres semble ainsi moins fiable que celle mesurée localement dans le plan d'écaillage. Cependant, il faut tout de même nuancer ces résultats car la gamme de $\dot{\epsilon}$ explorée s'étale sur un peu moins d'une décade et le calcul du taux de déformation avec l'équation 5.6 est une approximation de ce dernier. Pour statuer définitivement sur cette observation, il faudrait explorer une plus grande gamme de $\dot{\epsilon}$ sur au moins deux ou trois décades.

En conclusion de cette sous section, nous pouvons dire que lorsqu'il est seulement possible de calculer la tension d'écaillage et le taux de déformation à partir de la vitesse de la surface libre, alors il est plus judicieux d'utiliser l'expression de Kanel et al. 5.4 plutôt que l'approximation acoustique seule car elle améliore les résultats en ayant une expression qui reste simple et qui ne nécessite pas plus de calcul.

5.5.2 Distributions des pores sur une simulation de 8,3 milliards d'atomes

De nombreuses études dans la littérature explore l'influence de la distribution en volume des pores sur les modèles de croissance dynamique des matériaux ductiles [111-114]. La conclusion commune de ces travaux est que sous certaines hypothèses sur la distribution de la pression de cavitation, plus le taux de chargement est élevé, plus la distribution de taille des vides est étroite. Parallèlement, des expériences utilisant des installations à rayon X ont caractérisé ces distributions [115] et ont souvent montré qu'elles étaient à queue épaisse, en les ajustant typiquement par plusieurs lois de puissance. Ces données ont même été comparées avec succès à des simulations de DM [116, 117]. Plus récemment, d'autres mesures expérimentales à rayons X ont fourni des microstructures initiales de vides pour des simulations par éléments finis [118, 119]. Dans ces travaux, la distribution des vides a été ajustée avec une distribution log-normale. Son implémentation dans des simulations par éléments finis démontre un impact clair de la distribution de taille des vides sur la réponse macroscopique.

Nous voulons donc étudier une simulation de DM de grande taille avec notre nouvel outil d'analyse pour voir si nous pouvons apporter de nouveaux éléments de réponse. Nous avons choisi de nous concentrer sur l'étude d'un seul cas mais en maximisant sa taille. Nous regarderons en particulier la distribution en volume des pores et une répartition spatiale. Ce cas utilise lui aussi un échantillon ayant une longueur initiale de 150 nm et est impacté à v_0 mais cette fois il possède une section de $1\mu m \times 1\mu m$. Cette simulation implique 8,3 milliards d'atomes. Nous avons choisi de maximiser la section pour augmenter la quantité de pores générée et donc améliorer la qualité de nos statistiques en plus de démontrer l'applicabilité de notre algorithme à des simulations de cette taille (un des objectifs majeurs de cette thèse). La figure 5.14 présente une image de cette simulation.

La simulation a nécessité environ 10 millions d'heures de calculs et a été exécutée sur 256 nœuds (32 768 cœurs) du calculateur CEA-HF. L'analyse in-situ de la simulation a utilisé 0,2 % du temps de la simulation. Nous avons utilisé la méthode avec correction développée dans le chapitre 4 avec un seuil de sélection ρ_s égal à 50% de la densité du tantale *bulk* ρ_{Ta} . Les pores avec un volume inférieur au volume de 3 mailles cristallines ont été retirés du traitement.

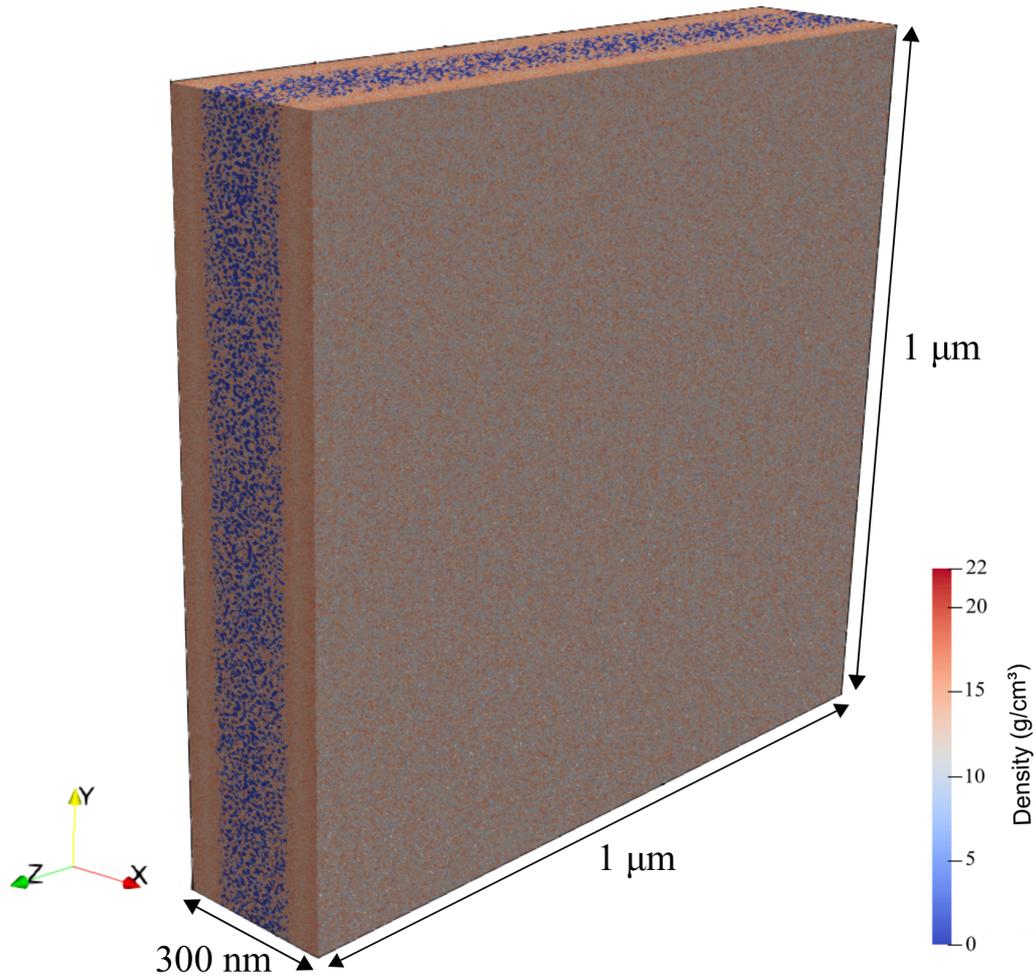


FIGURE 5.14 – Image de l'échantillon de tantale de dimension initiale $1\mu\text{m} \times 1\mu\text{m} \times 150\text{nm}$ (8,3 milliards d'atomes) impacté à v_0

La figure 5.15 présente la cumulée de la distribution du volume des pores au cours du temps avec en encart la distribution au cours du temps. Nous avons ajusté une loi log-normale qui semble en très bon accord avec nos données. Le même type de loi que celle utilisée dans [118, 119] et est souvent associée à des phénomènes statistiques multiplicatifs [105-107].

Regardons maintenant des informations sur l'arrangement spatial des pores en calculant la distance avec leur premier voisin et leur fonction de corrélation de paire. La distribution des distances au premier voisin est tracée dans la figure 5.16a et est comparée à une distribution poissonnienne qui aurait pour paramètre la densité moyenne expérimentale. La cumulée de ces distributions est tracée en encart de la figure. On peut clairement constater que la distribution poissonnienne est incompatible avec nos données et donc que les pores ne sont pas répartis spatialement de façon aléatoire, ce qui est confirmé par la fonction de corrélation de paire tracée dans la figure 5.16b, réalisée pour nos données expérimentales et pour un cas poissonnien généré pour avoir la même densité de pores. Ce résultat n'est pas étonnant car il a déjà été montré de nombreuses fois dans la littérature que les sites privilégiés de nucléation de pores sont les défauts contenus dans le matériau (joints de grains, macles, ...). Si notre matériau était exempt de défaut avant la propagation du choc, celle-ci a généré de nombreux défauts dans l'échantillon. Ces défauts sont probablement majoritairement des macles. En

perspective, une étude à différents taux de chargement et sur d'autres microstructures initiales serait intéressante pour voir l'évolution des distributions que nous avons évoquées.

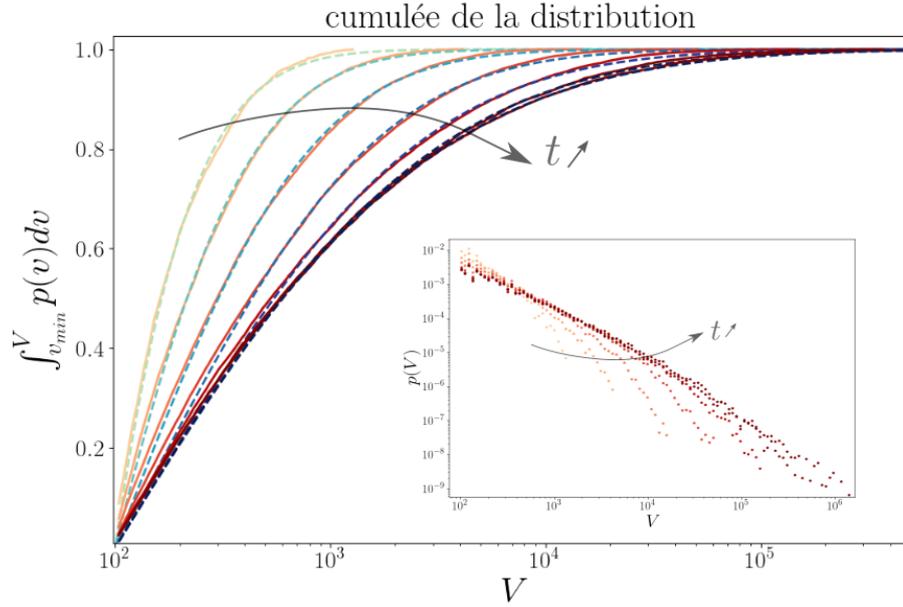
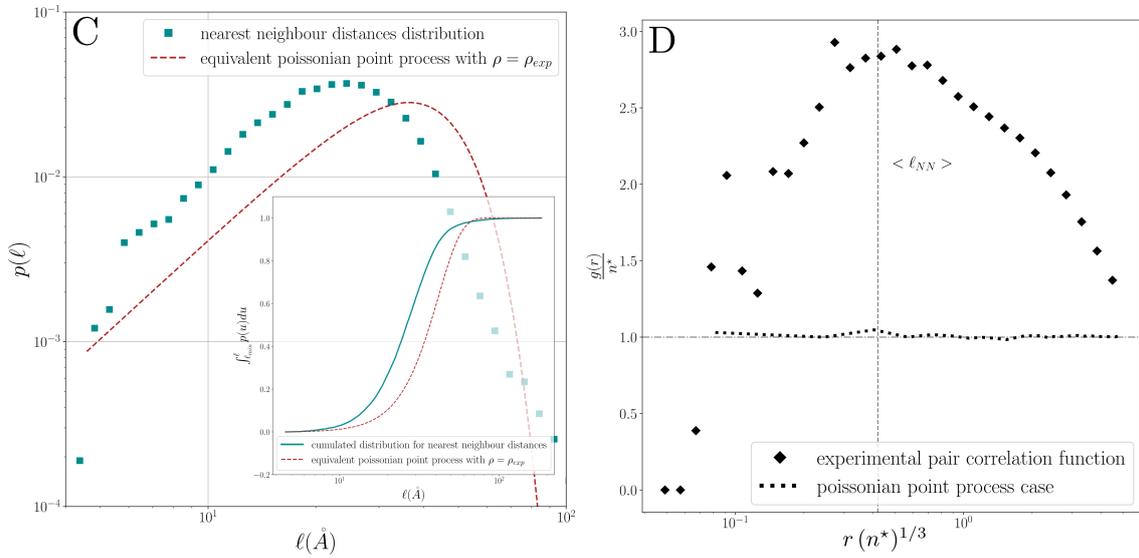


FIGURE 5.15 – Cumulé de la distribution en volume des pores à différents instants et la distribution de volume des pores en encart.



(a) Distribution de la distance au premier voisin et sa cumulé en encart. (b) Fonction de distribution radiale $g(r)$

FIGURE 5.16 – Comparaison de nos données à un cas poissonien qui aurait la même densité de pores.

En conclusion de cette sous section, nous avons démontré que notre analyse développée aux chapitres 3 et 4 était d'une part applicable efficacement sur des simulations à plusieurs

milliards d'atomes, et d'autre part qu'elle peut servir à amener des éléments de réponse sur la distribution de volume des pores pour les modèles d'endommagement dynamique.

5.6 Comparaison Tantale et potentiel modèle

Dans cette section, nous allons comparer les résultats de notre étude de tantale de la sous section 5.5.1 avec ceux que nous avons obtenus pour la version fragile et ductile du potentiel modèle présenté dans le chapitre 2, en les adimensionnant avec la méthode aussi présentée dans le chapitre 2. La première sous section propose une observation phénoménologique de la différence entre la version fragile du potentiel modèle et du tantale. La seconde sous section présente les différentes valeurs de tension d'écaillage en fonction du taux de déformation que nous avons obtenues pour les deux versions potentiel modèle et les compare aux résultats du tantale. L'objectif est de déterminer s'il peut-être intéressant d'utiliser ce potentiel modèle pour des études d'écaillage et si notre système d'adimensionnement permet effectivement de comparer les résultats entre ces deux matériaux très différents.

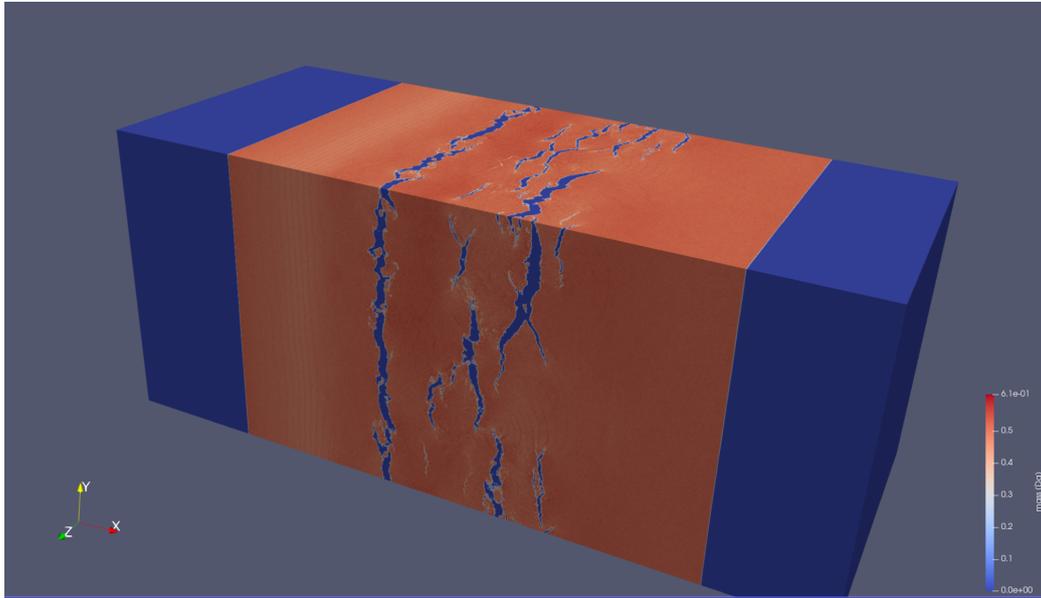
5.6.1 Observation phénoménologique

Nous proposons d'observer dans cette sous section deux images issues des simulations que nous avons réalisées pour distinguer, dans notre cas, les notions 'fragile' et "ductile". La première image est issue d'une simulation de la version fragile du potentiel modèle par impact de projectile (Fig. 5.17a). Le diagramme de marche de cette simulation est celui présenté dans la figure 5.7a. Le champ de densité est représenté dans l'image, les zones bleues étant des zones vides (densité nulle) et les nuances de rouge indiquent les zones de forte densité. La deuxième image (Fig. 5.17b) est issue d'une simulation de tantale impacté par piston. Nous pouvons clairement observer la différence de nature entre les vides nucléés dans les deux simulations. Dans la simulation "fragile", les vides s'apparentent à des micro-fissures alors que le tantale présente un comportement purement ductile. Ce comportement fragile du potentiel modèle s'explique par son rayon de coupure court qui réduit la portée de la partie attractive de son terme d'interaction, ainsi les atomes de ce matériau peuvent rapidement perdre leur voisin en tension et ont une facilité accrue pour le clivage.

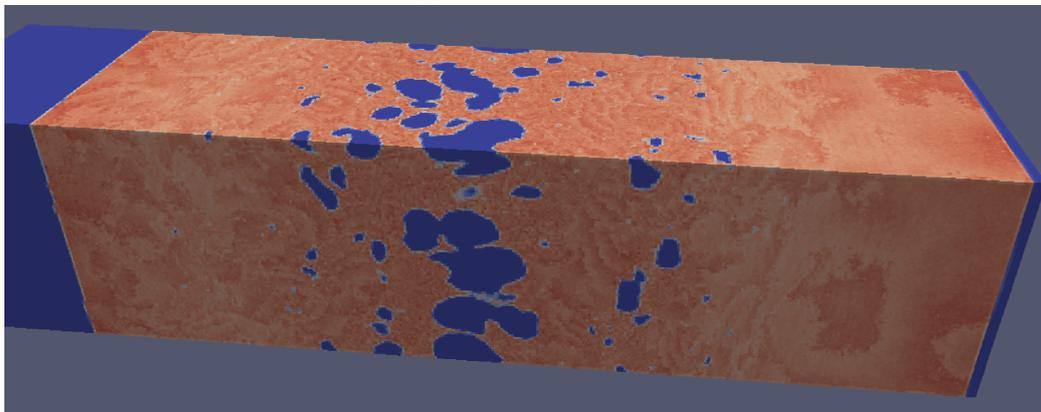
5.6.2 Analyse des quantités macroscopiques adimensionnées

Les expériences que nous avons réalisées sur les deux matériaux modèles pour étudier les valeurs de tension d'écaillage et de taux de déformation ont utilisé des échantillons de taille initiale fixe impactés à différentes vitesses. La taille des échantillons est fixée en nombre de mailles cristallines dans chaque direction de l'espace et correspond au nombre de mailles cristallines utilisées dans le cas du tantale : 303 mailles par 303 mailles de section sur 454 mailles de longueur. On peut noter que ces mailles cristallines comprennent deux fois plus d'atomes que celles du tantale et portent donc le nombre d'atomes simulés à 166 millions.

Pour être le plus proche possible de nos conditions de simulation du tantale, nous avons aussi réalisé ces simulations par impact piston. Cependant cela a nécessité au préalable de calculer les coefficients de la relation d'hugoniot (Eq. 5.1) de ces matériaux pour déterminer à quel moment relâcher le piston pour que les ondes de détente se croisent approximativement au centre de l'échantillon. Pour calculer ces coefficients, nous avons réalisé des chocs pistons sur de longs échantillons à différentes vitesses d'impact puis suivi la vitesse de propagation

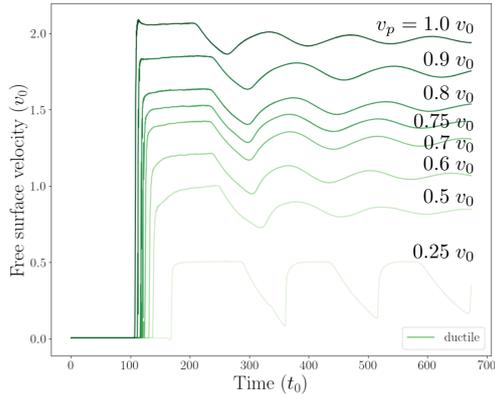


(a) Potentiel fragile

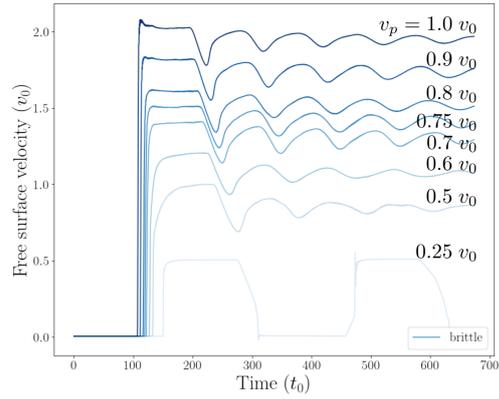


(b) Tantale

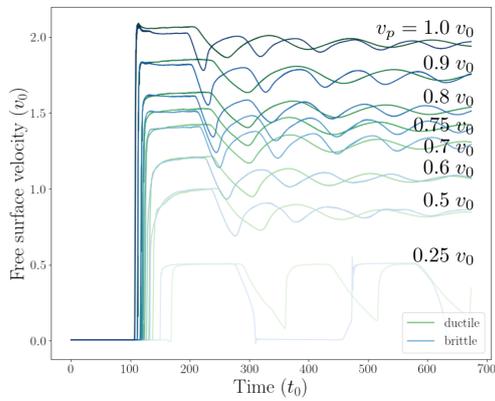
FIGURE 5.17 – Comparaison visuelle du comportement du potentiel fragile et du comportement du tantale. Le champ de densité représenté avec en bleu les zones vides de la simulation.



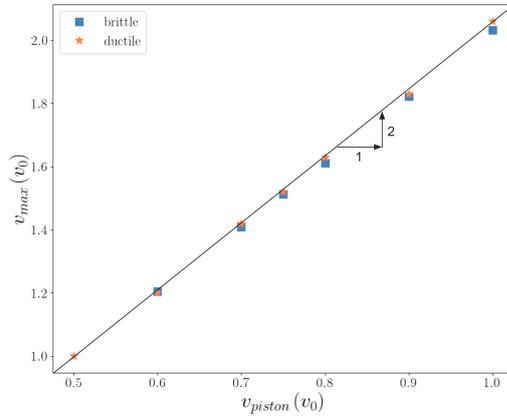
(a) Potentiel ductile



(b) Potentiel fragile



(c) Superposition de (a) et (b)



(d)

FIGURE 5.18 – (a-c) Signaux de vitesse de surface libre des matériaux modèles. (d) Vitesse maximum de la vitesse de surface libre en fonction de la vitesse d'impact.

du choc. Ainsi pour chaque vitesse d'impact piston testée, nous avons associé une vitesse de propagation de choc puis nous avons ajusté la relation (Eq. 5.1) sur ces données. Nous avons ainsi obtenu ces deux relations (exprimée en fonction de v_0) :

$$v_s^{fragile} = 2,878 + 1,552v_p \quad (5.7)$$

$$v_s^{ductile} = 2,656 + 1,178v_p \quad (5.8)$$

La figure 5.18a présente les différents signaux de vitesse de surface libre obtenus pour le matériau ductile aux différentes vitesses d'impacts utilisées, la figure 5.18b présente ceux pour le matériau fragile et la figure 5.18c superpose ces signaux. Ce potentiel n'ayant jamais été utilisé au préalable pour des simulations d'endommagement, nous avons vérifié que dans l'ensemble des cas, la valeur maximum de vitesse en face arrière était bien égale au double de la vitesse d'impact (Fig. 5.18d). C'est effectivement le cas pour les deux matériaux.

Nous pouvons constater avec les signaux de vitesse en face arrière que l'impact à $0,25 v_0$ a provoqué l'écaillage de l'échantillon ductile mais pas l'échantillon fragile. Ce résultat peut

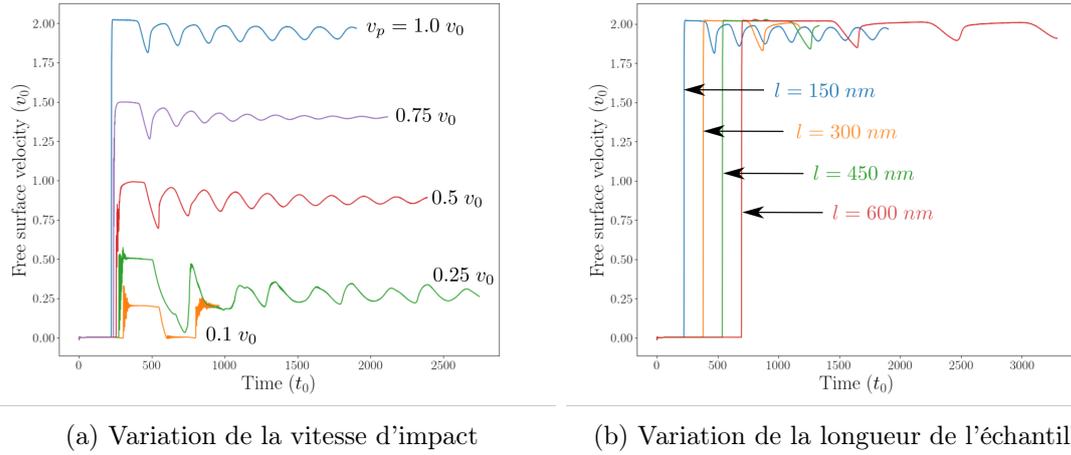


FIGURE 5.19 – Signaux de vitesse de surface libre des simulations de tantale de la figure 5.10 adimensionnés

être surprenant, nous l'interprétons comme étant du à la différence de raideur du potentiel. Le potentiel fragile ayant un rayon de coupure plus court que le potentiel ductile, la partie "attractive" du potentiel est plus raide et donc les forces d'attractions entre atomes sont plus importantes dans la version fragile du potentiel que la ductile. Cette hypothèse sera confirmée plus loin par l'étude de σ_{kanel} . La tension nécessaire pour écailler le matériau fragile est systématiquement plus importante que pour la version ductile.

Ne connaissant pas la vitesse du son dans ces matériaux et ayant déjà calculé leur relation d'hugoniot (Eq. 5.7-5.8), nous utiliserons les équations de Kanel pour calculer la tension d'écaillage et le taux de déformation.

La figure 5.19 présente de nouveau les signaux de vitesse de surface libre du tantale mais cette fois adimensionnés avec le système présenté dans le chapitre 2 pour comparer ses valeurs de σ_{kanel} et $\dot{\epsilon}_{kanel}$ à celle du potentiel modèle. La figure 5.20 présente l'ensemble de ces résultats avec en noir les résultats pour le tantale, en bleu ceux du matériau modèle fragile, et en vert ceux matériau modèle ductile. Nous pouvons constater que les résultats des matériaux modèles sont assez localisés dans leur gamme de tension d'écaillage et de taux de déformation. Nous pouvons aussi observer que dans le cas du matériau modèle ductile, la vitesse d'impact a surtout fait varier la tension d'écaillage et très peu le taux de déformation. Le matériau fragile subit un taux de déformation systématiquement plus élevé. $\dot{\epsilon}_{kanel}$ étant surtout corrélé au temps typique que le matériau met à casser, ce qui est cohérent avec le fait que le matériau fragile "clive" brutalement.

Les résultats du matériau modèle ductile sont plus proches de ceux du tantale. Finalement, nous pouvons conclure que le choix du système d'adimensionnement pour comparer des résultats d'écaillage en DM de cristaux n'ayant pas le même formalisme de potentiel semble judicieux car d'une part les vitesses d'impact générant de l'endommagement sont comprises aux alentours de v_0 et d'autre part les résultats de tension d'écaillage et de taux de déformation dans des gammes comparables. Il est intéressant de noter que malgré leur différence de cristallographie et donc des propriétés plastiques, le tantale et le matériau ductile sont comparables en terme de propriété d'écaillage.

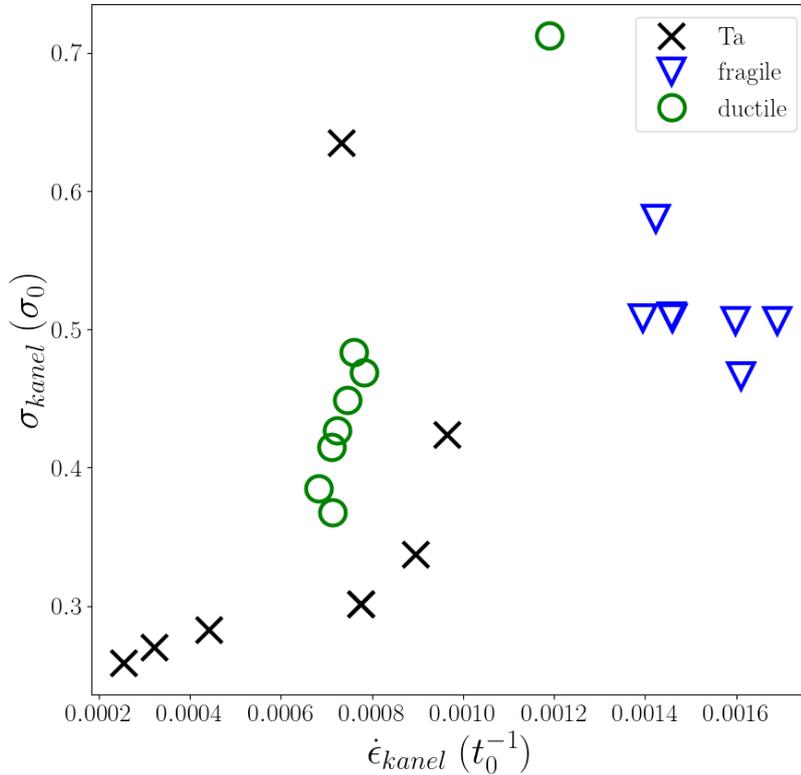


FIGURE 5.20 – Comparaison des valeurs adimensionnés de tension d'écaillage et de taux de déformation des matériaux modèles et du tantale

En conclusion, dans cette section, nous avons observé que le potentiel modèle que nous avons implémenté dans le chapitre 2 est utilisable pour des simulations d'écaillage et qu'il présente bien un comportement fragile ou ductile selon la version utilisée. Nous avons ensuite calculé la relation d'hugoniot qui lie la vitesse des particules à celle du choc pour ces deux matériaux, et grâce au système d'adimensionnement défini dans le chapitre 2, nous avons pu faire une première comparaison entre les résultats que nous avons obtenus pour le tantale et pour les matériaux modèles. La section suivante présente une nouvelle perspective d'application de notre algorithme et du potentiel modèle.

5.7 Perspectives d'application : détection de surfaces

Lorsque nous avons introduit le phénomène d'écaillage dans ce chapitre (Sec. 5.3), nous avons présenté une image MEB d'une surface de tantale écaillée. Une thèse expérimentale (Corentin Thouenon) a débuté au laboratoire pour caractériser la rugosité de ces surfaces et remonter aux distributions de pores qui les ont engendrés. Avec les nouvelles techniques qui seront développées, il serait intéressant de regarder si les surfaces écaillées obtenues en DM sont comparables à celles expérimentales.

Dans cette optique, nous avons implémenté dans notre analyse un filtre de détection de surface qui permet de connecter par ECC l'ensemble des cellules appartenant à une surface.

Ce filtre se fonde sur la densité de matière ; si une cellule a une densité supérieure à un certain seuil et qu'au moins une de ses cellules voisine a une densité inférieure à ce seuil, alors cette cellule est considérée comme une cellule de surface. La position de chacune des cellules d'une surface peut ensuite être sauvegardée pour générer une image. Nous avons utilisé cette technique pour générer la figure 5.21 de gauche. C'est une surface de tantale impactée à $0,75 v_0$, la surface fait 200 nm par 200 nm de section. Le rendu visuel 3D a été réalisé avec ParaView et un filtre d'isosurface. Pour comparaison, l'image expérimentale précédente est placée à sa droite. Nous pouvons déjà remarquer que la surface obtenue en DM à fondu sous l'effet du choc et ne sera probablement pas comparable à la surface expérimentale. Outre la différence d'échelle de taille, les taux de déformation obtenus en DM sont d'un à deux ordres de grandeur supérieure aux mesures expérimentales.

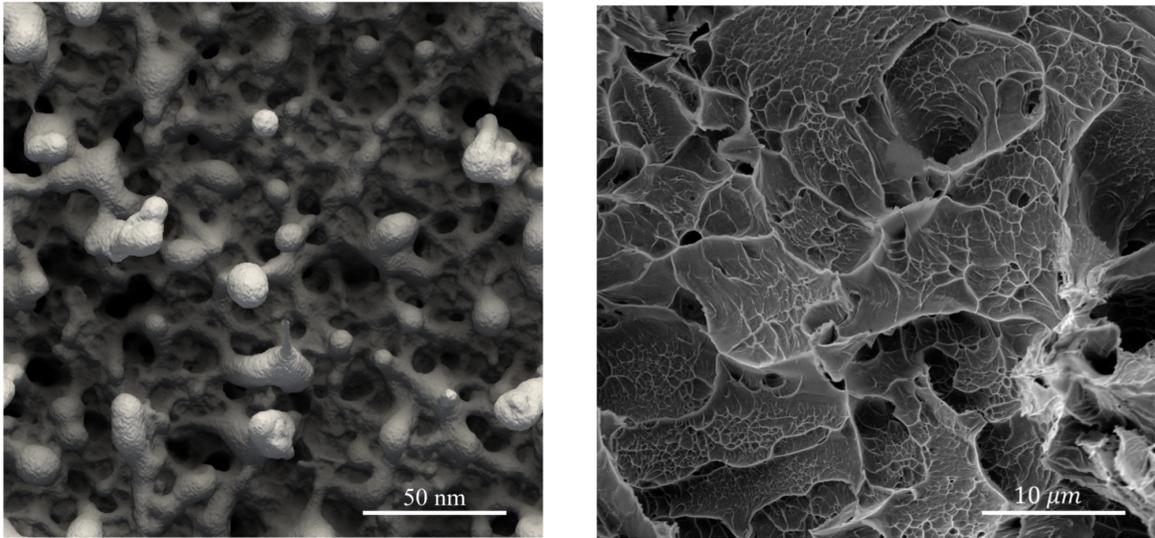


FIGURE 5.21 – Gauche, une surface de tantale écaillée en DM après un impact de $0,75 v_0$, extraite avec notre algorithme et rendu en 3D avec ParaView. Droite, surface de tantale écaillé expérimentalement par tir laser présenté dans la figure 5.6

Néanmoins, si nous regardons les surfaces obtenues dans le cas du potentiel modèle par la même technique (Fig. 5.22), les surfaces n'ont pas fondu et se rapprochent visuellement de la surface expérimentale. Ainsi, l'étude de ces surfaces générées en DM avec les nouveaux outils qui sont développés dans le cadre de la thèse expérimentale est une perspective intéressante pour de nouvelles comparaisons expériences-simulations.

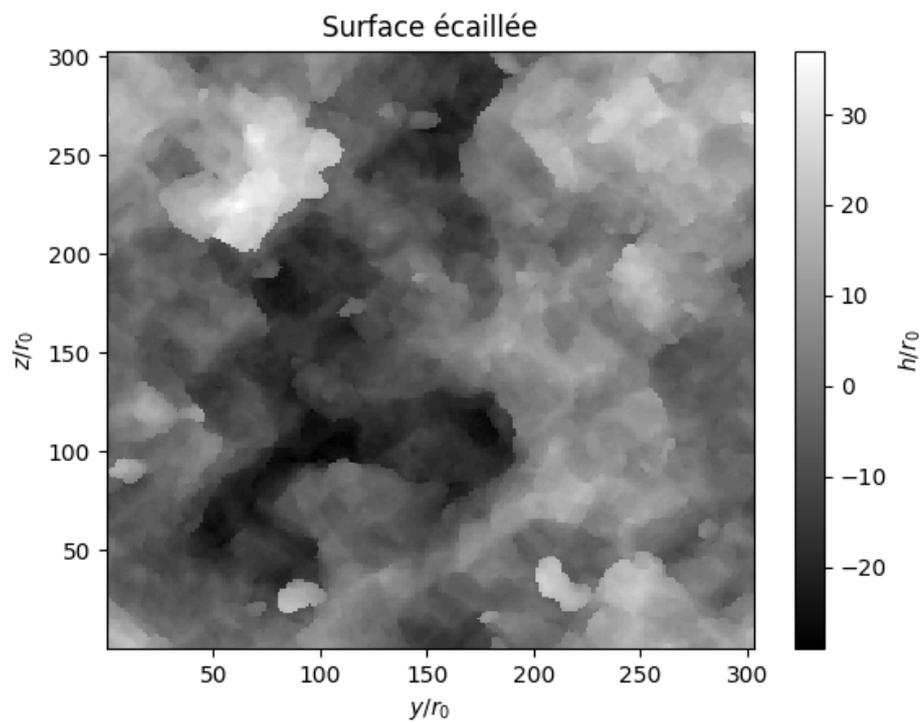
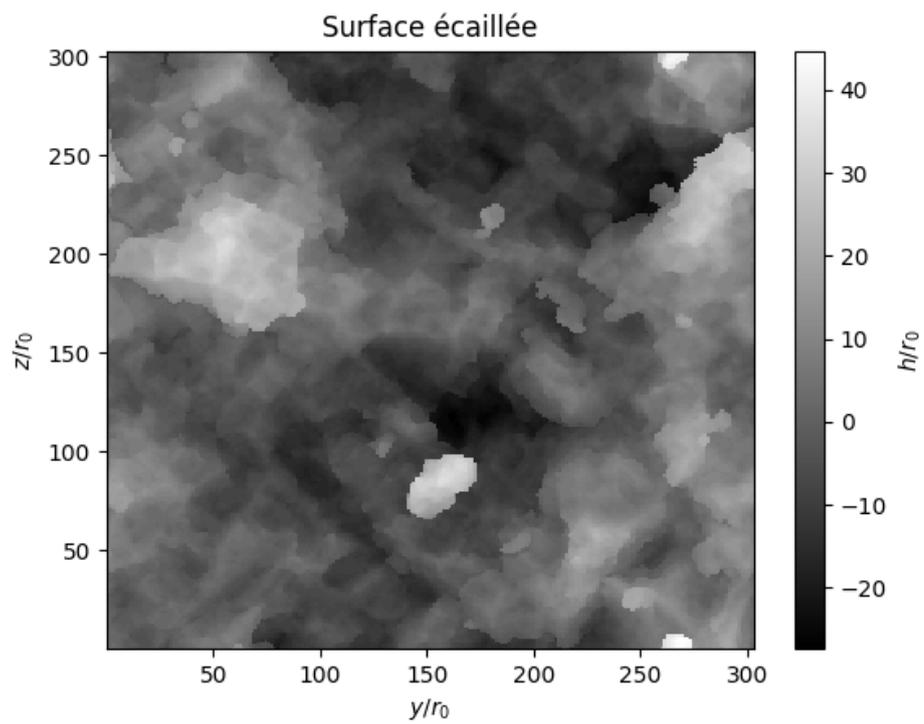


FIGURE 5.22 – Surfaces écaillées de part et d'autre de l'écaille pour la simulation du potentiel fragile impacté à $0.7v_0$

Conclusion

Les objectifs majeurs de ce chapitre étaient d'une part de démontrer les capacités de notre algorithme et d'autre part de fournir de nouveaux éléments pour l'étude des phénomènes d'éjections de matières et d'écaillage. Pour démontrer ses capacités, nous avons vu dans la section 5.2 que notre algorithme était capable de post-traiter des simulations de plus d'un milliard d'atomes afin d'analyser sous un nouveau jour d'anciennes simulations coûteuses en temps de calcul, mais aussi qu'il était capable de traiter de façon in-situ de nouvelles simulations à plus de 8,3 milliard d'atomes. Nous avons donc démontré qu'il peut soutenir la charge de calcul qui nous avait été fixée. Ensuite, nous avons vu qu'il pouvait s'appliquer à diverses situations physiques, le micro-jetting (Sec. 5.1), le splashing (Sec. 5.2), et l'écaillage (Sec. 5.3). Ce qui démontre, d'une part, qu'il est versatile, et d'autre part, que grâce à son filtre de sélection programmable il est très modulable pour de futures applications ou besoins spécifiques (Sec. 5.7).

Ces capacités de traitements nous ont permis d'obtenir diverses nouvelles informations sur les phénomènes simulés. Dans le cas du splashing (Sec. 5.2), nous avons notamment pu observer que l'énergie cinétique de l'ensemble des agrégats de matière éjectées par l'impact représentait 3% de l'énergie cinétique du projectile incident. Observation qui conforte l'hypothèse utilisée expérimentalement pour les diagnostic expérimentaux type fenêtre d'Asay qui postule que l'énergie dissipée par l'éjection d'agrégats est négligeable. Dans le cas de l'écaillage, nous avons d'abord pu observer que, dans nos cas, l'apparition des pores avait lieu avant d'atteindre la tension d'écaillage maximum, et que la croissance des vides était dominée par des effets d'inertie (Sec. 5.4). Nous avons ensuite pu observer qu'une loi log-normale pouvait s'ajuster aux distributions de volume de pores (Sec. 5.5.2). Enfin, nous avons pu comparer les valeurs de tension d'écaillage et de taux de déformation calculées grâce au signal de vitesse de surface libre et celles calculées dans le plan d'écaillage (Sec. 5.5.1).

Ce chapitre nous a aussi permis de vérifier que le potentiel modèle que nous avons implémenté au chapitre 2 pouvait être utilisé dans des simulations d'écaillage et que le système d'adimensionnement présenté dans le chapitre 2 permet effectivement de comparer en DM les résultats de matériaux ayant des formalismes de potentiel différents.

Chapitre 6

Conclusion générale

Le contexte de cette thèse porte sur l'étude de certains phénomènes liés à la propagation de chocs qui soulèvent encore des questions sur les mécanismes qui dictent leur physique, principalement à cause de la difficulté de leur étude expérimentale. Pour contourner ces problèmes, la dynamique moléculaire s'est imposée dans la littérature comme une technique de choix pour la simulation et l'étude de ces phénomènes, car elle présente l'avantage de ne pas avoir besoin d'autres modèles que celui du potentiel d'interaction entre atomes pour les reproduire, si le système est soumis aux bonnes conditions et que le potentiel d'interaction est suffisamment bien décrit. Cependant, l'inconvénient de cette technique de simulation est que bien souvent, l'échelle de taille atteignable avec ces simulations est très éloignée de celle désirée. Mais, avec le développement continu du calcul haute performance, les limites de taille de simulation atteignables en DM sont sans cesse repoussées. Avant le début de cette thèse, les codes de DM les plus performants étaient déjà capables de simuler plusieurs milliards d'atomes. Pourtant, très peu de simulations à cette échelle étaient réalisées à cause de la complexité de traitement des grandes quantités de données qu'elles génèrent et l'absence d'outils d'analyse adaptés à cette échelle. La présentation de ce contexte et des enjeux associés a été l'objet du premier chapitre.

La problématique de cette thèse était donc de développer un code d'analyse in-situ de DM adapté à des simulations d'au moins plusieurs milliards d'atomes, dans le but d'étudier des phénomènes ultra dynamiques liés à la propagation de chocs dans les métaux, avec pour condition de s'exécuter en un temps raisonnable comparé à la DM, et de s'adapter facilement à une diversité d'usage.

Pour répondre à cette problématique, nous avons organisé cette thèse en 4 chapitres principaux (Chap. 2 à 5). Un premier chapitre (Chap. 2) dans lequel nous avons présenté les méthodes et les outils utilisés en DM. Un deuxième chapitre (Chap. 3) dans lequel nous avons développé l'algorithme d'analyse, dont nous avons ensuite caractérisé la précision dans le chapitre suivant (Chap. 4). Et enfin nous avons appliqué notre algorithme à des simulations de plusieurs milliards d'atomes dans le dernier chapitre (Chap. 5).

Le chapitre 2 a présenté dans un premier temps le principe de la dynamique moléculaire, avec les grandes familles de potentiels classiques. Nous avons également décrit le schéma numérique que nous utilisons pour résoudre les équations du mouvement des atomes dans le temps, ainsi que les principales optimisations algorithmiques utilisées dans le cas de potentiel courte portée. Enfin nous avons présenté le code que nous avons utilisé : exaStamp. Nous avons ensuite présenté dans un second temps, les deux potentiels que nous avons implémentés dans exaStamp pour réaliser les simulations d'écaillage du chapitre 5, les raisons qui nous

ont poussé à utiliser ces potentiels, et la validation de nos implémentations. Nous avons ensuite conclu ce chapitre en présentant le système d'adimensionnement que nous avons établi pour pouvoir comparer les simulations que nous avons réalisées avec ces potentiels.

Dans le chapitre 3, nous avons développé l'algorithme d'analyse adapté à des simulations de DM de plusieurs milliards d'atomes qui détecte et caractérise les zones dites d'intérêt de la simulation. Ces zones d'intérêts sont définies par l'intermédiaire d'un filtre de sélection programmable, ce qui permet à l'algorithme de s'adapter à une grande diversité d'usage. L'algorithme est fondé sur la projection des atomes sur une grille régulière et sur un algorithme d'étiquetage en composantes connexes (ECC) 3D que nous avons parallélisé en mémoire partagée et distribuée. Une première version a été développée et a ensuite été améliorée dans une deuxième version en réarrangeant la structure mémoire associée utilisée et en modifiant la façon dont les frontières MPI sont traitées. Les mesures de performances de l'algorithme ont montré que le coût d'une itération de notre algorithme représente en général la moitié du coût d'une itération de DM avec un potentiel EAM, un des potentiels les moins coûteux en DM. De plus, l'algorithme n'ayant besoin d'être exécuté qu'à une fréquence d'au moins 1 ordre de grandeur plus faible que la DM pour les phénomènes ultra rapides et de plus de 2 ordres de grandeur dans la plupart des cas, le surcoût d'analyse est ramené au maximum à 5% du coût de la DM et seulement à 0,5% en moyenne. La deuxième version de l'algorithme fait l'objet d'une publication soumise dans *Computer Physics Communications* et est actuellement en cours de *review*. La version soumise est fournie en annexe A.

Le chapitre 4 nous a servi à calculer l'incertitude de mesure associée à l'algorithme d'analyse en fonction de la résolution de la grille régulière utilisée. La convergence des résultats de notre algorithme a été démontrée sur un cas d'école et un cas d'application réel. Cependant, nous avons observé que la convergence était trop lente, ce qui nous a amené à améliorer les résultats de notre analyse en proposant une méthode de correction fondée sur des considérations physiques. Un article présentant ces résultats et une comparaison à un outil de la littérature (Ovito) pour des simulations de petites tailles est en cours de rédaction en vue d'une soumission dans *Journal of Computational Physics*.

Dans le chapitre 5, nous avons démontré les capacités de traitement de l'algorithme aussi bien en termes de charge de calcul (jusqu'à 8,3 milliards d'atomes), qu'au niveau de la diversité d'analyse qu'il peut fournir. Il a été exploité dans une publication sur le phénomène de *micro-jetting* que nous avons introduite et fournie en annexe B. Nous avons ensuite appliqué l'algorithme à des simulations de *splashing* et montré que seulement 3% de l'énergie cinétique du projectile incident était transmise aux particules éjectées suite au choc. Cette information conforte l'hypothèse utilisée dans les dispositifs expérimentaux de type fenêtre d'Asay pour calculer la masse des particules incidentes. Cette hypothèse étant que l'ensemble de l'énergie cinétique des particules incidentes était transmise à la fenêtre. Nous avons ensuite conclu le chapitre sur une étude d'écaillage. Cette étude nous a permis de vérifier que le potentiel modèle implémenté au chapitre 2 qui n'avait encore jamais été utilisé dans ce type de simulation pouvait l'être. Grâce au système d'adimensionnement présenté au chapitre 2, nous avons pu comparer les résultats de taux de chargement et de tension d'écaillage obtenus avec les signaux de vitesse de surface libre pour les matériaux modèle et le tantale. De plus, dans le cas du tantale, nous avons analysé au cours du temps la distribution de pores d'une simulation d'un échantillon de taille $1 \mu\text{m} \times 1 \mu\text{m} \times 150 \text{nm}$ et observé qu'elle pouvait être ajustée avec une loi log-normale.

Les perspectives ouvertes par cette thèse sont nombreuses. D'un point de vue purement algorithmique, l'ECC peut encore être amélioré en particulier dans son traitement des frontières entre threads et entre domaines MPI qui sont devenus les points limitants de cette partie de l'algorithme. D'un point de vue implémentation, porter la projection sur grille régulière sur GPU pourrait certainement diminuer son temps d'exécution. Cependant, la perspective qui nous semble la plus pertinente pour l'utilisation de machines équipées de GPU, est de réserver ceux-ci aux calculs de DM bien plus coûteux en ressources que l'analyse, et de réaliser une exécution concurrente de l'analyse sur CPU pour recouvrir les deux temps de calcul et rendre l'analyse virtuellement gratuite.

D'un point de vue fonctionnalité, étendre l'ECC 3D à de l'ECC 4D, avec pour 4ème dimension le temps, serait très intéressant pour automatiquement détecter les événements de nucléations, coalescences, fragmentation et disparition. Ceci permettrait aussi de suivre en particulier l'ensemble de l'historique d'un élément traqué. Ces informations permettraient d'avoir des analyses encore plus fines pour enrichir à terme de nouveaux modèles.

D'un point de vue application, nous avons vu en fin du chapitre 5 qu'une perspective de nouvelle application pour l'algorithme est la détection et l'extraction de surfaces d'écaillage obtenues en DM pour d'éventuelles comparaisons à l'expérience. Mais les autres possibilités sont nombreuses. Toute situation physique, nécessitant la détection et la caractérisation de zones particulières, peut bénéficier de l'analyse. Par exemple pour le suivi de l'évolution d'un élément dans un autre, que ce soit des inclusions de gaz ou de liquide dans une matrice métallique comme dans le cas des inclusions de bulle d'hélium dans du tungstène qui est rencontré dans les installations de type tokamak.

De plus, l'ensemble de notre méthode n'étant pas uniquement propre à la DM mais aux simulations particulières en général, elle pourrait être étendue à d'autres type de simulation comme la *Discrete Element Method* (DEM) ou la *Smoothed-Particle Hydrodynamics* (SPH). Dans le cadre du projet exaNBody qui englobe les codes exaDEM, exaSPH et exaStamp, ce portage devrait être disponible dans un futur proche.

Bibliographie

1. KESSLER, D. J. & COUR-PALAIS, B. G. Collision frequency of artificial satellites : The creation of a debris belt. *Journal of Geophysical Research : Space Physics* **83**, 2637-2646 (1978) (cf. p. 7).
2. KESSLER, D. J. Collisional cascading : The limits of population growth in low earth orbit. *Advances in Space research* **11**, 63-66 (1991) (cf. p. 7).
3. ASAY, J. Thick-plate technique for measuring ejecta from shocked surfaces. *Journal of Applied Physics* **49**, 6173-6175 (1978) (cf. p. 9, 92).
4. COAKLEY, J., HIGGINBOTHAM, A., MCGONEGLE, D. *et al.* Femtosecond quantification of void evolution during rapid material failure. *Science advances* **6**, eabb4434 (2020) (cf. p. 10).
5. MACNIDER, B., JONES, D., CALLANAN, J. *et al.* In situ measurement of damage evolution in shocked magnesium as a function of microstructure. *Science Advances* **9**, eadi2606 (2023) (cf. p. 10).
6. TCHIPEV, N., SECKLER, S., HEINEN, M. *et al.* TweTriS : Twenty trillion-atom simulation. *The International Journal of High Performance Computing Applications* **33**, 838-854 (2019) (cf. p. 13, 45).
7. ZEPEDA-RUIZ, L. A., STUKOWSKI, A., OPPELSTRUP, T. *et al.* Probing the limits of metal plasticity with molecular dynamics simulations. *Nature* **550**, 492-495 (2017) (cf. p. 13).
8. STUKOWSKI, A. Visualization and analysis of atomistic simulation data with OVITO—the Open Visualization Tool. *Modelling and simulation in materials science and engineering* **18**, 015012 (2009) (cf. p. 13).
9. THOMPSON, A. P., AKTULGA, H. M., BERGER, R. *et al.* LAMMPS—a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Computer Physics Communications* **271**, 108171 (2022) (cf. p. 13, 21).
10. CIEREN, E. *Molecular Dynamics for Exascale Supercomputers* Theses (Université de Bordeaux, oct. 2015) (cf. p. 16, 26).
11. EDWARDS, H. C., TROTT, C. R. & SUNDERLAND, D. Kokkos : Enabling manycore performance portability through polymorphic memory access patterns. *Journal of Parallel and Distributed Computing* **74**. Domain-Specific Languages and High-Level Frameworks for High-Performance Computing, 3202-3216 (2014) (cf. p. 16).

12. SHAW, D. E., ADAMS, P. J., AZARIA, A. *et al.* *Anton 3 : twenty microseconds of molecular dynamics simulation before lunch* in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (2021), 1-11 (cf. p. 16).
13. HOHENBERG, P. & KOHN, W. Inhomogeneous electron gas. *Physical review* **136**, B864 (1964) (cf. p. 18).
14. KOHN, W. & SHAM, L. J. Self-consistent equations including exchange and correlation effects. *Physical review* **140**, A1133 (1965) (cf. p. 18).
15. JONES, J. E. On the determination of molecular fields.—II. From the equation of state of a gas. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* **106**, 463-477 (1924) (cf. p. 20).
16. FISCHER, J. & WENDLAND, M. On the history of key empirical intermolecular potentials. *Fluid Phase Equilibria* **573**, 113876 (2023) (cf. p. 20).
17. DAW, M. S. & BASKES, M. I. Embedded-atom method : Derivation and application to impurities, surfaces, and other defects in metals. *Physical Review B* **29**, 6443 (1984) (cf. p. 20, 31).
18. BASKES, M. Application of the embedded-atom method to covalent materials : a semiempirical potential for silicon. *Physical review letters* **59**, 2666 (1987) (cf. p. 20).
19. BASKES, M. I. Modified embedded-atom potentials for cubic materials and impurities. *Physical review B* **46**, 2727 (1992) (cf. p. 20).
20. VERLET, L. Computer" experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. *Physical review* **159**, 98 (1967) (cf. p. 22, 23).
21. STÖRMER, C. Sur les trajectoires des corpuscules électrisés dans l'espace. Applications à l'aurore boréale et aux perturbations magnétiques. *Radium (Paris)* **4**, 2-5 (1907) (cf. p. 22).
22. in. *Solving Ordinary Differential Equations I : Nonstiff Problems* 355-474 (Springer Berlin Heidelberg, Berlin, Heidelberg, 1993) (cf. p. 22).
23. SWOPE, W. C., ANDERSEN, H. C., BERENS, P. H. *et al.* A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules : Application to small water clusters. *The Journal of chemical physics* **76**, 637-649 (1982) (cf. p. 22).
24. PLIMPTON, S. Fast Parallel Algorithms for Short-Range Molecular Dynamics. *Journal of Computational Physics* **117**, 1-19 (1995) (cf. p. 24).
25. CIEREN, E., COLOMBET, L., PITOISET, S. *et al.* *ExaStamp : a parallel framework for molecular dynamics on heterogeneous clusters* in *Euro-Par 2014 : Parallel Processing Workshops : Euro-Par 2014 International Workshops, Porto, Portugal, August 25-26, 2014, Revised Selected Papers, Part II 20* (2014), 121-132 (cf. p. 26).

26. CARRARD, T., PRAT, R., LATU, G. *et al.* *ExaNBody : A HPC Framework for N-Body Applications* in *Euro-Par 2023 : Parallel Processing Workshops* (éd. ZEINALIPOUR, D., BLANCO HERAS, D., PALLIS, G. *et al.*) (Springer Nature Switzerland, Cham, 2024), 342-354 (cf. p. 26, 28-30).
27. PRAT, R., CARRARD, T., SOULARD, L. *et al.* AMR-based molecular dynamics for non-uniform, highly dynamic particle simulations. *Computer Physics Communications* **253**, 107177 (2020) (cf. p. 26).
28. PRAT, R. *Équilibrage dynamique de charge sur supercalculateur exaflopique appliqué à la dynamique moléculaire* Theses (Université de Bordeaux, oct. 2019) (cf. p. 26, 27).
29. SOULARD, L., DURAND, O., BURIE, J.-R. *et al.* Micro-jetting : Areal density calculation from a triangular groove. *Journal of Applied Physics* **135**, 205902. eprint : https://pubs.aip.org/aip/jap/article-pdf/doi/10.1063/5.0209692/19969354/205902_1_5.0209692.pdf (mai 2024) (cf. p. 26, 92).
30. SOULARD, L. Micro-jetting : A semi-analytical model to calculate the velocity and density of the jet from a triangular groove. *Journal of Applied Physics* **133**, 085901 (2023) (cf. p. 26, 92).
31. DURAND, O., SOULARD, L., BABILOTTE, K. *et al.* Ejecta size distributions from large-scale molecular dynamics simulations of the high velocity impact of droplets of liquid metal on rigid and liquid surfaces. *Bulletin of the American Physical Society* **67** (2022) (cf. p. 26).
32. SOULARD, L., CARRARD, T. & DURAND, O. Molecular dynamics study of the impact of a solid drop on a solid target. *Journal of Applied Physics* **131**, 135901 (2022) (cf. p. 26, 93).
33. SOULARD, L., DURAND, O., PRAT, R. *et al.* High velocity impact of a spherical particle on a surface : Theory and simulation of the jet formation. *Journal of Applied Physics* **129**, 205104 (2021) (cf. p. 26).
34. *exaNBody github* <https://github.com/Collab4exaNBody/exaNBody> (cf. p. 26, 28).
35. *exaDEM github* <https://github.com/Collab4exaNBody/exaDEM> (cf. p. 26, 28).
36. BOMAN, E. G., CATALYUREK, U. V., CHEVALIER, C. *et al.* The Zoltan and Isorropia Parallel Toolkits for Combinatorial Scientific Computing : Partitioning, Ordering, and Coloring. *Scientific Programming* **20**, 129-150 (2012) (cf. p. 27).
37. BERGER & BOKHARI. A Partitioning Strategy for Nonuniform Problems on Multiprocessors. *IEEE Transactions on Computers* **C-36**, 570-580 (1987) (cf. p. 27, 59).
38. AGULLO, E., AUMAGE, O., BRAMAS, B. *et al.* Bridging the gap between OpenMP and task-based runtime systems for the fast multipole method. *IEEE Transactions on Parallel and Distributed Systems* **28**, 2794-2807 (2017) (cf. p. 29).

39. ROY, G. *Vers une modélisation approfondie de l'endommagement ductile dynamique : investigation expérimentale d'une nuance de tantale et développements théoriques*. thèse de doct. (Université de Poitiers, 2003) (cf. p. 31).
40. MITCHELL, A. & NELLIS, W. Shock compression of aluminum, copper, and tantalum. *Journal of Applied Physics* **52**, 3363-3374 (1981) (cf. p. 31).
41. SOULARD, L., BONTAZ-CARION, J. & CUQ-LELANDAIS, J. Experimental and numerical study of the tantalum single crystal spallation. *The European Physical Journal B* **85**, 1-15 (2012) (cf. p. 31, 45).
42. VIDEAU, L., COMBIS, P., LAFFITE, S. *et al.* Laser-driven spall experiments in ductile materials in order to characterize Johnson fracture model constants in *AIP Conference Proceedings* **1426** (2012), 1011-1014 (cf. p. 31).
43. CUQ-LELANDAIS, J.-P. *Etude du comportement dynamique de matériaux sous choc laser subpicoseconde* Theses (ISAE-ENSMA Ecole Nationale Supérieure de Mécanique et d'Aérotechnique - Poitiers, déc. 2010) (cf. p. 31, 45).
44. REMINGTON, T., HAHN, E., ZHAO, S. *et al.* Spall strength dependence on grain size and strain rate in tantalum. *Acta Materialia* **158**, 313-329 (2018) (cf. p. 31).
45. MARTIN, C. T., SOLIERI, N., FORNASIERE, E. *et al.* First observation of spalling in tantalum at high temperatures induced by high energy proton beam impacts. *European Journal of Mechanics-A/Solids* **85**, 104149 (2021) (cf. p. 31).
46. PERREAULT, C., HUSTON, L. Q., BURRAGE, K. *et al.* Strength of tantalum to 276 GPa determined by two x-ray diffraction techniques using diamond anvil cells. *Journal of Applied Physics* **131** (2022) (cf. p. 31).
47. KATAGIRI, K., OZAKI, N., OHMURA, S. *et al.* Liquid structure of tantalum under internal negative pressure. *Physical Review Letters* **126**, 175503 (2021) (cf. p. 31).
48. HAHN, E. N., GERMANN, T. C., RAVELO, R. *et al.* On the ultimate tensile strength of tantalum. *Acta Materialia* **126**, 313-328 (2017) (cf. p. 31, 45, 104, 105, 107).
49. HAHN, E. N., FENSIN, S. J., GERMANN, T. C. *et al.* Orientation dependent spall strength of tantalum single crystals. *Acta Materialia* **159**, 241-248 (2018) (cf. p. 31, 45).
50. CHEN, J., HAHN, E., DONGARE, A. *et al.* Understanding and predicting damage and failure at grain boundaries in BCC Ta. *Journal of Applied Physics* **126** (2019) (cf. p. 31, 45).
51. WU, D., CHEN, K., ZHU, Y. *et al.* Unveiling grain size effect on shock-induced plasticity and its underlying mechanisms in nano-polycrystalline Ta. *Mechanics of Materials* **160**, 103952 (2021) (cf. p. 31, 45).
52. WANG, J., WANG, F., ZENG, X. *et al.* Unraveling the plasticity performance and melting in single crystal tantalum damaged by shock compression. *Engineering Fracture Mechanics* **276**, 108921 (2022) (cf. p. 31, 45).

53. CHEN, P., WANG, X., WANG, P. *et al.* Shock response of pre-existing spall damage in copper. *Journal of Applied Physics* **131**, 015903 (2022) (cf. p. 31, 45).
54. RAVELO, R., GERMANN, T., GUERRERO, O. *et al.* Shock-induced plasticity in tantalum single crystals : Interatomic potentials and large-scale molecular-dynamics simulations. *Physical Review B* **88**, 134101 (2013) (cf. p. 31-34, 36, 42).
55. RYDBERG, R. Graphische darstellung einiger bandenspektroskopischer ergebnisse. *Zeitschrift für Physik* **73**, 376-385 (1932) (cf. p. 32).
56. ROSE, J. H., SMITH, J. R., GUINEA, F. *et al.* Universal features of the equation of state of metals. *Physical Review B* **29**, 2963 (1984) (cf. p. 32).
57. HOLIAN, B., VOTER, A., WAGNER, N. *et al.* Effects of pairwise versus many-body forces on high-stress plastic deformation. *Physical Review A* **43**, 2655 (1991) (cf. p. 32).
58. *NIST Interatomic Potentials Repository* URL : <https://www.ctcms.nist.gov/potentials> Website DOI : 10.18434/m37 (cf. p. 33, 35).
59. RAVELO, R., GERMANN, T. C., GUERRERO, O. *et al.* Erratum : Shock-induced plasticity in tantalum single crystals : Interatomic potentials and large-scale molecular-dynamics simulations [Phys. Rev. B 88, 134101 (2013)] (2014) (cf. p. 36).
60. RAJAN, V., WARNER, D. & CURTIN, W. An interatomic pair potential with tunable intrinsic ductility. *Modelling and Simulation in Materials Science and Engineering* **24**, 025005 (2016) (cf. p. 37).
61. AGHABABAEI, R., WARNER, D. H. & MOLINARI, J.-F. On the debris-level origins of adhesive wear. *Proceedings of the National Academy of Sciences* **114**, 7935-7940 (2017) (cf. p. 37, 39).
62. MOLINARI, J.-F., AGHABABAEI, R., BRINK, T. *et al.* Adhesive wear mechanisms uncovered by atomistic simulations. *Friction* **6**, 245-259 (2018) (cf. p. 37).
63. WATTEL, S., GARCIA-SUAREZ, J. & MOLINARI, J.-F. Understanding the mechanisms of adhesive wear for heterogeneous materials through atomistic simulations. *Extreme Mechanics Letters* **57**, 101913 (2022) (cf. p. 37).
64. GARCIA-SUAREZ, J., BRINK, T. & MOLINARI, J.-F. Breakdown of Reye's theory in nanoscale wear. *Journal of the Mechanics and Physics of Solids* **173**, 105236 (2023) (cf. p. 37).
65. AGHABABAEI, R., WARNER, D. H. & MOLINARI, J.-F. Critical length scale controls adhesive wear mechanisms. *Nature communications* **7**, 11816 (2016) (cf. p. 37-39).
66. MORSE, P. M. Diatomic molecules according to the wave mechanics. II. Vibrational levels. *Physical review* **34**, 57 (1929) (cf. p. 37).

67. HEMANI, H., WARRIER, M., SAKTHIVEL, N. *et al.* Voxel based parallel post processor for void nucleation and growth analysis of atomistic simulations of material fracture. *Journal of Molecular Graphics and Modelling* **50**, 134-141 (2014) (cf. p. 45).
68. YANG, X., ZENG, X., WANG, F. *et al.* Spallation fracture dependence on shock intensity and loading duration in single-crystal aluminum. *Computational Materials Science* **210**, 111060 (2022) (cf. p. 45).
69. WANG, Y., ZENG, X., YANG, X. *et al.* Shock-induced spallation in single-crystalline tantalum at elevated temperatures through molecular dynamics modeling. *Computational Materials Science* **201**, 110870 (2022) (cf. p. 45).
70. GERMANN, T. C. & KADAU, K. Trillion-atom molecular dynamics becomes a reality. *International Journal of Modern Physics C* **19**, 1315-1319 (2008) (cf. p. 45).
71. *CEA-HF website* <https://www-hpc.cea.fr/fr/EXA.html> (cf. p. 45, 93).
72. *TOP500 website* <https://www.top500.org/lists/top500/list/2023/11/>. Nov. 2023 (cf. p. 45).
73. ESTER, M., KRIEGEL, H.-P., SANDER, J. *et al.* A density-based algorithm for discovering clusters in large spatial databases with noise. in. **96** (1996), 226-231 (cf. p. 46).
74. SCHUBERT, E., SANDER, J., ESTER, M. *et al.* DBSCAN revisited, revisited : why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)* **42**, 1-21 (2017) (cf. p. 46).
75. HE, Y., TAN, H., LUO, W. *et al.* Mr-dbscan : an efficient parallel density-based clustering algorithm using mapreduce in 2011 IEEE 17th international conference on parallel and distributed systems (2011), 473-480 (cf. p. 46).
76. WANG, W., YANG, J., MUNTZ, R. *et al.* STING : A statistical information grid approach to spatial data mining in *Vldb* **97** (1997), 186-195 (cf. p. 46).
77. WANG, Y., GU, Y. & SHUN, J. Theoretically-efficient and practical parallel DBSCAN in *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (2020), 2555-2571 (cf. p. 47).
78. ROSENFELD, A. & PFALTZ, J. L. Sequential operations in digital picture processing. *Journal of the ACM (JACM)* **13**, 471-494 (1966) (cf. p. 47).
79. SHAPIRO, L. G. & HARALICK, R. M. Structural descriptions and inexact matching. *IEEE Transactions on pattern analysis and machine intelligence*, 504-519 (1981) (cf. p. 47).
80. LUMIA, R., SHAPIRO, L. & ZUNIGA, O. A new connected components algorithm for virtual memory computers. *Computer Vision, Graphics, and Image Processing* **22**, 287-300 (1983) (cf. p. 47).
81. RONSE, C. & DEVIJVER, P. A. *Connected components in binary images : the detection problem* (John Wiley & Sons, Inc., 1984) (cf. p. 47).

82. HOSHEN, J. & KOPELMAN, R. Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm. *Phys. Rev. B* **14**, 3438-3445 (8 oct. 1976) (cf. p. 47).
83. GALLER, B. A. & FISHER, M. J. An improved equivalence algorithm. *Communications of the ACM* **7**, 301-303 (1964) (cf. p. 47).
84. KIRKWOOD, J. & IRVING, J. The statistical mechanical theory of transport processes. IV. The equations of hydrodynamics. *J. Chem. Phys* **18**, 817-829 (1950) (cf. p. 48).
85. HARDY, R. J. Formulas for determining local properties in molecular-dynamics simulations : Shock waves. *The Journal of Chemical Physics* **76**, 622-628 (1982) (cf. p. 48).
86. ROOT, S., HARDY, R. J. & SWANSON, D. R. Continuum predictions from molecular dynamics simulations : Shock waves. *The Journal of chemical physics* **118**, 3161-3165 (2003) (cf. p. 48).
87. HARDY, R. J. Atomistic formulas for local properties in systems with many-body interactions. *The Journal of Chemical Physics* **145**, 204103 (2016) (cf. p. 49).
88. KHAWAJA, H. A., SCOTT, S. A., VIRK, M. S. *et al.* Quantitative analysis of accuracy of voidage computations in CFD-DEM simulations. *The Journal of Computational Multiphase Flows* **4**, 183-192 (2012) (cf. p. 50).
89. STROBL, S., FORMELLA, A. & PÖSCHEL, T. Exact calculation of the overlap volume of spheres and mesh elements. *Journal of Computational Physics* **311**, 158-172 (2016) (cf. p. 50).
90. SCHROEDER, W., MARTIN, K. & LORENSEN, B. *The Visualization Toolkit (4th ed.)* (Kitware, 2006) (cf. p. 52).
91. AHRENS, J., GEVECI, B. & LAW, C. in (éd. HANSEN, C. D. & JOHNSON, C. R.) 717-731 (Elsevier Inc., Burlington, MA, USA, 2005) (cf. p. 52).
92. *ParaView website* <https://www.paraview.org/> (cf. p. 52).
93. HE, L., REN, X., GAO, Q. *et al.* The connected-component labeling problem : A review of state-of-the-art algorithms. *Pattern Recognition* **70**, 25-43 (2017) (cf. p. 54, 67).
94. CABARET, L., LACASSAGNE, L. & ETIEMBLE, D. Parallel light speed labeling : an efficient connected component algorithm for labeling and analysis on multi-core processors. *Journal of Real-Time Image Processing* **15**, 173-196 (2018) (cf. p. 54, 56).
95. GRANA, C., BOLELLI, F., BARALDI, L. *et al.* YACCLAB-yet another connected components labeling benchmark in 2016 23rd International Conference on Pattern Recognition (ICPR) (2016), 3109-3114 (cf. p. 54).
96. BOLELLI, F., CANCELLA, M., BARALDI, L. *et al.* Toward reliable experiments on the performance of connected components labeling algorithms. *Journal of Real-Time Image Processing* **17**, 229-244 (2020) (cf. p. 54).

97. YACCLAB *github* <https://github.com/prittt/YACCLAB> (cf. p. 54).
98. MAURICE, N., LEMAITRE, F., SOPENA, J. *et al.* *LSL3D : a run-based Connected Component Labeling algorithm for 3D volumes in International Conference on Image Analysis and Processing* (2022), 132-142 (cf. p. 54).
99. LAMM, S. & SANDERS, P. *Communication-efficient Massively Distributed Connected Components in 2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS)* (2022), 302-312 (cf. p. 54).
100. XU, J., GUO, H., SHEN, H.-W. *et al.* Asynchronous and Load-Balanced Union-Find for Distributed and Parallel Scientific Data Visualization and Analysis. *IEEE Transactions on Visualization and Computer Graphics* **27** (2021) (cf. p. 54).
101. SMITH, A. J. Cache memories. *ACM Computing Surveys (CSUR)* **14**, 473-530 (1982) (cf. p. 71).
102. *Zen 3 Memory Hierarchy* https://en.wikichip.org/wiki/amd/microarchitectures/zen_3#Memory_Hierarchy (2024) (cf. p. 72, 73).
103. DURAND, O. & SOULARD, L. Mass-velocity and size-velocity distributions of ejecta cloud from shock-loaded tin surface using atomistic simulations. *Journal of Applied Physics* **117** (2015) (cf. p. 92).
104. VELLA, J. R., CHEN, M., STILLINGER, F. H. *et al.* Structural and dynamic properties of liquid tin from a new modified embedded-atom method force field. *Physical Review B* **95**, 064202 (2017) (cf. p. 93).
105. SODERLUND, J., KISS, L. B., NIKLASSON, G. A. *et al.* Lognormal Size Distributions in Particle Growth Processes without coagulation. *Physical Review Letters* **80** (1998) (cf. p. 94, 108).
106. LEE, K. W., LEE, Y. J. & HAN, D. S. The lognormal size distribution theory for brownian coagulation in the low knudsen number regime. *Journal of colloid and interface science* **188**, 486-492 (1997) (cf. p. 94, 108).
107. POLIZZI, S., PEREZ, F.-J., ARNEODO, A. *et al.* Power-law and log-normal avalanche size statistics in random growth processes. *Physical Review E* **104** (2021) (cf. p. 94, 108).
108. MARSH, S. P. *LASL shock Hugoniot data* (Univ of California Press, 1980) (cf. p. 100).
109. KANEL, G. I. Spall fracture : methodological aspects, mechanisms and governing factors. *International Journal of Fracture* **163**, 173-191 (2010) (cf. p. 100, 105).
110. MOLINARI, A. & MERCIER, S. Micromechanical modelling of porous materials under dynamic loading. *Journal of the Mechanics and Physics of Solids* **49**, 1497-1516 (2001) (cf. p. 103).
111. CZARNOTA, C., MERCIER, S. & MOLINARI, A. Modelling of nucleation and void growth in dynamic pressure loading, application to spall test on tantalum. *International Journal of Fracture*, 177-194 (2006) (cf. p. 107).

112. CZARNOTA, C., JACQUES, N., MERCIER, S. *et al.* Modelling of dynamic ductile fracture and application to the simulation of plate impact tests on tantalum. *Journal of Mechanics and Physics of Solids*, 16-24-1650 (2008) (cf. p. 107).
113. WRIGHT, T. W. & RAMESH, K. T. Statistically informed dynamics of void growth in rate dependent materials. *International Journal of Impact Engineering* **36**, 1242-1249 (2009) (cf. p. 107).
114. WILKERSON, J. W. On the micromechanics of void dynamics at extreme rates. *International Journal of Plasticity* **95**, 21-42 (2017) (cf. p. 107).
115. BONTAZ-CARION, J. & PELLEGRINI, Y.-P. X-Ray microtomography analysis of dynamic damage in Tantalum. *Advances engineering materials* **8**, 480-486 (2006) (cf. p. 107).
116. SOULARD, L., BONTAZ-CARION, J. & CUQ-LELANDAIS, J.-P. Experimental and numerical study of the tantalum single crystal spallation. *The European Physical Journal B* **85** (2012) (cf. p. 107).
117. BONTAZ-CARION, J., SOULARD, L., LESCOUTE, E. *et al.* The X-Ray microtomography backed by molecular dynamics simulations in the analysis of shock-induced damage in ductile materials. *Material Science Forum* **905**, 182-189 (2006) (cf. p. 107).
118. MARVI-MASHHADI, M., VAZ-ROMERO, A., SKET, F. *et al.* Finite element analysis to determine the role of porosity in dynamic localization and fragmentation : Application to porous microstructures obtained from additively manufactured materials. *International Journal of PLasticity* **143**, 102999 (2021) (cf. p. 107, 108).
119. VISHNU, A., VADILLO, G. & RODRIGUEZ-MARTINEZ, J. Void growth in ductile materials with realistic porous microstructures. *International Journal of Plasticity* **167** (2023) (cf. p. 107, 108).

On-the-fly clustering for exascale molecular dynamics simulations

Killian Babilotte^{a,b}, Alizée Dubois^{a,b}, Thierry Carrard^{a,b}, Paul Lafourcade^{a,b}, Laurent Videau^{a,b},
Jean-François Molinari^c, Laurent Souldard^{a,b}

^aCEA, DAM, DIF, F-91297 Arpajon, France

^bUniversité Paris-Saclay, CEA, Laboratoire Matière en Conditions Extrêmes, F-91680 Bruyères-le-Châtel, France

^cInstitute of Civil Engineering, Institute of Materials, École Polytechnique Fédérale de Lausanne (EPFL), CH 1015
Lausanne, Switzerland

Abstract

Computational resources have experienced exponential growth in the last decades enabling the simulation of complex physical problems at the cost of a massive increase in data storage. This is especially true for N-body simulations now reaching billions or trillions particles in certain cases. To overcome the drawbacks of data storage on disk for post-processing purposes, *on-the-fly* analysis has gained momentum but still represents a challenge in both its implementation and efficiency without impacting the simulation engine performances. This work provides a new *in-situ* procedure for features detection in massive N-body simulations, leveraging state-of-the-art techniques from various fields. Based on a *discrete-to-continuum* paradigm shift, particles and their respective physical quantities are projected onto a 3D regular grid before applying image analysis algorithms to group voxels based on specific user-defined criteria. A significant extension to the hybrid parallelism of connected component analysis within the image processing community is also introduced in the present study. Traditionally operating in shared memory parallelism, this extension now incorporates both distributed and shared memory approaches. The implementation is carried out within the exaStamp classical Molecular Dynamics code, a variant of the open-source exaNBody platform (see section 6). This adaptation allows for the *on-the-fly* analysis of multi-billion atoms samples with at most a 1.3% overhead. In addition, the entire framework is benchmarked up to 32768 cores. The applicability of the present approach is demonstrated on the case of a spall fracture in a tantalum sample as well as high velocity impact of a tin droplets on a rigid surface.

Keywords: In-Situ Analysis, Large Scale Molecular Dynamics, Connected Components Labeling, Visualization, Splashing, Spall Fracture.

Email address: alizee.dubois@cea.fr (Alizée Dubois)

Preprint submitted to Computer Physics Communications

August 23, 2024

1. Introduction

Given the recent exponential increase in computational power, simulation and understanding of complex problems have been unlocked at the cost of a substantial increase in data generation for post-analysis purposes. Atomistic simulations for instance are increasingly performed with samples' size previously non reachable, due to the emergence of supercomputers and systematic efficient parallelism in classical molecular dynamics (MD) engines. While performing large-scale atomistic simulations became systematic, their post-processing has become a significant challenge [1]. Data storage being at the heart of the problem, post-analysis of cutting-edge simulations have now to be set aside in favor of *on-the-fly* analysis [2, 3].

MD simulations are well-suited for modeling ultra-fast phenomena over sub-microsecond timescales, such as shock wave propagation or hypervelocity impacts [4, 5, 6] at the microscopic scale, leading to complex and nonlinear materials behavior such as ejecta or void nucleation. The core of such simulations resides in the force field used, a function that models interatomic interactions and contains the underlying physics of the simulated specimen. Such force field is typically calibrated using state-of-the-art *ab-initio* calculations and experimental results and many forms exist, going from the simple Lennard-Jones pair potential to highly complex Machine Learning potentials (MLIP).

The development of high-efficiency codes and the utilization of hardware acceleration techniques have made large-scale MD simulations feasible [7]. LAMMPS [8, 9] has emerged as the standard code in the field of condensed matter physics, renowned for its high performance, extensive capabilities, and open-source accessibility. Other MD codes, such as SPaSM [10], have also achieved significant milestones. For instance, SPaSM conducted the first trillion-atom simulation in 2008 [11] and, more recently, the ls1mardyn code accomplished a twenty-trillion-atom case in 2019 [12].

Despite these advancements, the memory footprint generated by such large-scale simulations, coupled with the absence of *in-situ* techniques, has hindered their analysis and usage in producing meaningful physical results. Currently, post-processing such simulations is dominated by OVITO [13], which has been augmented with a wide range of powerful features in the last decade. However, its external use and its partial parallelism make it challenging to employ on simulations with more than a few tens of million atoms.

In the present work, we present an efficient *on-the-fly* analysis tool capable of handling substantial amounts of data without imposing a significant overhead on the MD simulation. The developed *in-situ* algorithm projects physical quantities associated with discrete positions onto a 3D regular grid, already used by the HPC engine for parallelism purposes. The algorithm allows the application of various filters to delineate connected areas based on specific voxel properties. To achieve this, state-of-the-art 2D image connected component labeling (CCL) algorithms were extended to 3D and parallelized using both distributed and shared memory paradigms with MPI and OpenMP. Benchmarks were conducted on multi-billion atom workloads using up to 256 nodes of bi-socket AMD EPYC 7763 64-core CPUs, totaling 32,768 cores.

The considered physical applications involve pore nucleation, spall fracture, and material ejection—examples that require a high input/output flow, large-scale systems, and tracking numerous quantities during the simulation. The versatility of the developed algorithm is demonstrated in de-

tecting both pores and aggregates. While currently embedded in the MD code, the algorithm has the potential to become an autonomous routine callable by any simulations working on a regular grid. Additionally, the algorithm showcases its capability in identifying the volume and position of distinct zones based on binarized quantities, as well as in projecting physical fields onto these objects. In the case of several billion voxels, the on-the-fly application of the algorithm exhibits an overhead cost of less than 1.3%.

2. Related work

The analysis presented in this article provides significant opportunities for novel statistical analysis in simulations approaching 'mesoscale' sizes. In this paragraph, we will justify our method choices among various options for achieving the overall goal of feature detection through clustering. However, to our knowledge, neither LAMMPS nor OVITO currently offers a large-scale clustering feature; in particular, OVITO's capabilities are limited to a few tens of millions of atoms. Consequently, a comparison at an equivalent scale with another algorithm from the literature is currently impossible.

2.1. DBSCAN vs CCL comparison

For a given quantity defined at the continuum level-scale, various methods can be employed to extract physical insights through clustering. In this work, we leverage the connected components labeling (CCL) method. The first presentation of a similar method was by Hoshen and Kopelman [14], while alternative clustering techniques like DBSCAN [15, 16] could have been considered. However, DBSCAN, by its algorithmic structure, demonstrates a best-case complexity of $\mathcal{O}(n_p \log n_p)$ and at worst, a $\mathcal{O}(n_p^2)$ complexity, where n_p represents the number of data points in the dataset.

In contrast, the CCL method, tailored for image processing (employing two-pass algorithms), showcases a linear $\mathcal{O}(n_{vox})$ complexity (n_{vox} being the number of voxels). This renders it a computationally efficient method, especially well-suited for quantities easily binarized. Moreover, unlike DBSCAN, which primarily focuses on aggregating points, CCL offers the capability to identify both voids and aggregates, essential features for specific applications in materials science. It is for these reasons that we have chosen the CCL method as the most suitable approach for our work.

Through the grid feature option recently implemented in LAMMPS [17], OVITO, using a projection based on the *gaussian density method* [18], offers a toolbox for volumetric region identification. However, the analysis is currently limited to a few million atoms [19]. This emphasizes the need for scalable solutions in the field.

2.2. Structured data vs unstructured graph

Connected Components Labeling (CCL) is a fundamental technique in graph theory and comprises two main algorithm families: one dedicated to general unstructured graphs and the other applied to structured data such as images.

When it comes to image processing, CCL is employed for labeling connected pixels in a binarized image. This crucial step is part of the connected component analysis (CCA) chain, where features are subsequently extracted from labeled areas. The most efficient CCL algorithms are

two-pass algorithms with a complexity, as mentioned previously, of $O(n_{vox})$, where n_{vox} is the number of voxels. Labels are temporarily assigned to pixels in the first pass, and a label equivalence graph is constructed. In the second pass, the equivalence graph is used to assign a single label to each connected region. Various CCL algorithms, using different masks, equivalence graph construction methods, and browsing approaches (pixel-based [20], run-length [21]), exist. CCA feature extraction could be performed during the first pass, eliminating the need for a second pass [22]. Consequently, the terms CCL and CCA are sometimes used interchangeably in the literature. While several efficient parallel algorithms for image analysis have emerged in shared memory [23], as noted in a recent review [24], there is a highlighted lack of efficient hybrid parallelism implementations in 3D. We explain this gap in literature by the fact that nowadays standard images easily fits in computers memory.

But in the context of general graphs, the graphs sizes could easily oversize the amount of memory available in one computer. Those graphs could also already been distributed by nature in many cases like networks. For those reasons, the parallel algorithms that have been developed to label those graphs have been developed in distributed memory. Those algorithms are multi-pass, with the number of passes capped at n , the number of vertices. In worst-case scenarios, this results in a complexity of $O(n^2)$. Examples of such algorithms' implementations can be found in recent literature papers [25, 26, 27]. To reduce the message sizes, these algorithms perform some "compression" steps where each MPI worker pre-solves its local graphs. This approach was inspired by shared memory algorithms [28]. However, within our specific context of 3D image treatment, we operate on image structures that enable the use of more efficient and standard shared memory algorithms, which we have further adapted for distributed memory parallelization. A recent paper [29] attempted a similar approach on 3D images but combined a structured data approach on each MPI with an unstructured approach on the graph for the global MPI communication. In our case, we consistently leverage the spatial structuring of our data throughout the entire process.

3. Numerical methods

3.1. Molecular Dynamics

Classical molecular dynamics (MD) is a special case of N-Body simulations where interatomic forces derive from a potential incorporating physical features of the simulated material, which are subsequently used to update both positions and velocities using, e.g. the Verlet integration scheme [30]. In the present work, we used the ExaStamp MD code [31], now based on the exaNBody platform (See section 6) [32]. Even if the problem involves N self-interacting atoms, MD considers, in many cases, a cutoff radius r_{cut} beyond which atomic interactions are neglected. This feature allows for cell-based spatial division of the domain through several MPI processes (Figure 1) with a $O(N_{atom})$ computational complexity [33]. Various spatial decomposition methods are available in ExaStamp thanks to its coupling with the Zoltan library. However, ExaStamp benefits from its own schemes for very heterogeneous systems [34], such as microjetting [35] and splashing [36] processes. Additionally, our algorithm will plug itself in during the MD computation; if a worker has just finished computing forces, it can jump to projection.

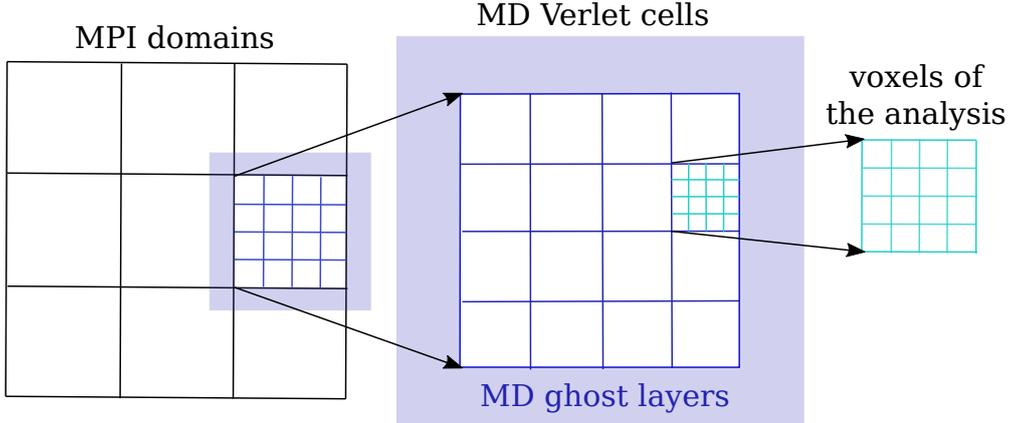


Figure 1: The initial step involves partitioning the physical sample into multiple MPI domains. The Molecular Dynamics (MD) calculation is based on a Verlet cell decomposition, utilizing the extinction of the potential beyond a specified cutoff radius. A regular grid with a greater subdivision is over-imposed on this primary partitioning, the relevant physical quantities are projected on this refined grid.

3.2. Body projection on regular grid

Transposing discrete information from atomistic simulations to the continuum systematically involves a projection step onto a regular grid through a localization function that ensures the bijection between both formalisms. To do so, an additional regular grid is superimposed on the Verlet cell-based spatial decomposition of MD (see Figure 1). The user can subdivide it in each spatial direction, creating “voxels” with a final size denoted as s . It’s worth noting that, similar to the original grid of Verlet cells, this secondary grid (light blue in Figure 1) has the same ghost layer as the original MD but projected onto the voxel grid. It will be utilized for CCL.

The coupling of discrete and continuum information, for which Hardy was a pioneer, has found extensive applications [37, 38, 39]. This approach is still employed in LAMMPS, as seen in models like the two-temperature model (TTM) and heat transfer models [40]. Furthermore, Hardy’s description was utilized for the study of MD simulations through 2D visualization techniques [41, 42]. Our work is based on Hardy’s technique [37], which involves mapping a particle i with position r_i onto a grid using a non-negative localization function denoted $\Delta(r_i - x)$, which peaks at $x = r_i$ and vanishes for $|r_i - x| > l/2$, with l being the spatial binning resolution. To preserve the projected physical quantity, such as atomic mass [37, 39], the following normalization is used:

$$\iiint \Delta(r_i - x) d^3x = 1, \quad (1)$$

and the contribution of an atom i to the voxel j written ω_{ij} reads:

$$\omega_{ij} = \iiint_{V_j} \Delta(r_i - x) dx^3 \quad (2)$$

with V_j the volume of voxel j . Relation (1) then ensures that:

$$\sum_{j=1}^{n_{\text{voxel}}} \omega_{ij} = 1, \quad \text{for } i = 1, \dots, n_{\text{atoms}} \quad (3)$$

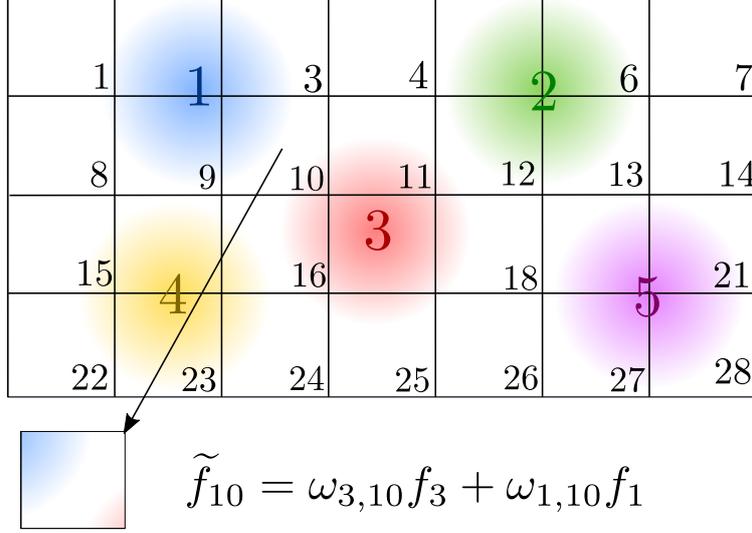


Figure 2: Each cell of the regular voxel grid has a unique *voxel_{id}*. The atoms are projected onto each voxel using a certain weight ω_{ij} , where i represents the atoms and j represents the voxel (see Eq.2 for the exact definition). The physical quantity \tilde{f}_j is then defined as an average of the atomic values f_i (Eq.4).

This formalism allows any extensive quantity f attached to atoms to be regularized and expressed at the j^{th} voxel scale (Figure 2):

$$\tilde{f}_j = \sum_{i=1}^{N_{\text{atoms}}} \omega_{ij} f_i \quad (4)$$

with f_i representing the value of the attached property to the i^{th} atom.

For intensive properties such as stress, Hardy provides specific calculations for potential interactions that are radially symmetric. We choose to use a generic formulation that, given an intensive property g_i defined for an atom i , smoothes the projected value \tilde{g}_j in the voxel j as:

$$\tilde{g}_j = \frac{\sum_{i=1}^{N_{\text{atoms}}} \omega_{ij} g_i}{\sum_i \omega_{ij}} \quad (5)$$

The localization function is chosen by the user. In this paper, a box function was used, denoted as Π in the following, with an edge length p . For most of the analysis, $p = a$ was chosen, where a is a distance typical of the studied system such as the lattice crystal parameter. The advantage of using such a localization function is that the intersection between a voxel and the atom cube can be computed very efficiently using the Axis-Aligned Bounding Box (AABB) technique. This allows the analysis to be performed in-situ without adding too much overhead to the simulation.

For a given atom i , we denote ω_{ij} as the amount of intersection between the atom cube and the j^{th} voxel, given by (6), equivalent to (2).

$$\omega_{ij} = \iiint_{V_{voxel}^j} \Pi\left(\frac{x - x_i}{p}\right) dx^3 \quad (6)$$

The estimation of ω_{ij} can play a critical role in modifying the smoothing introduced by the method.

From a parallel implementation standpoint, it is crucial to note that no communications are involved in the projection step (see Figure 3). The ghost layers are already filled with the appropriate particles to compute the MD Verlet scheme in the next iteration, and we only need to project them onto the grid. Each MPI process computes its local grid using the available threads. We loop over all our atoms simply with a `pragma omp for` clause. By dividing the atom’s position by s , we determine the local index of its voxel within the MPI reference frame. The value of s/a is then used to determine the number of neighboring voxels for computing the intersection in each spatial direction. To prevent race conditions, we employ a `pragma omp atomic update` clause during the grid value update.

3.3. Connected Components Labeling

To perform CCL in a distributed and shared memory context, where the computational cost increases linearly with the number of voxels, an even spatial decomposition of voxels is the best policy to distribute the workload among parallel workers. However, the distributed memory spatial decomposition is determined by the MD workload, as it is the most computationally intensive part. Refer to section 3.1 for more details on this decomposition. An unavoidable imbalance could appear among our MPI process workload due to this policy. On the shared memory side, we are free to apply the policy of our choice for spatial decomposition. In this case, we opted for Recursive Coordinate Bisection (RCB) to achieve a more even load balancing across the threads. Once the threads have obtained their working block of voxels through the RCB calculation, each of them initiates a first pass on their work block by applying Algorithm 1. The algorithm assigns a temporary label to each cell that meets a filter condition defined by the user and constructs its local equivalence table (Figure 3 for the creation of the local graph and Figure 4 for the label assignment). The $voxel_{id}$ is an absolute label computable by all MPI processes associated with each voxel. It will be used later to connect the boundaries. The `unordered_map` C++ object (native hash table) was chosen as the structure for the equivalence table due to its $\Theta(1)$ complexity for looking up, inserting, or deleting elements. To ensure that the `unordered_map` does not re-hash during computation and has enough capacity to hold all the temporary labels assigned to the work block, we use the `reserve` method with a parameter of $n/8$ (worst-case scenario theoretical number of labels), where n is the number of voxels in the work block.

The equivalence table is implemented as a directed graph, where the keys of the `unordered_map` represent graph vertices, and the associated values represent the vertices they are connected to via directed edges. A vertex can only point to another vertex with a lower id or to itself. A vertex pointing to itself is a root of the graph, i.e., the local connected component label. This guarantees that when a vertex is found in the graph, following the edges will always lead to its root. Moreover, if a vertex is not found in the table, it means that it needs to be inserted into the table as a root pointing to itself. When a thread has finished its first pass (cf Figure 4), it performs what is called a “union” phase in CCL literature on its equivalence table. The path compression step of the

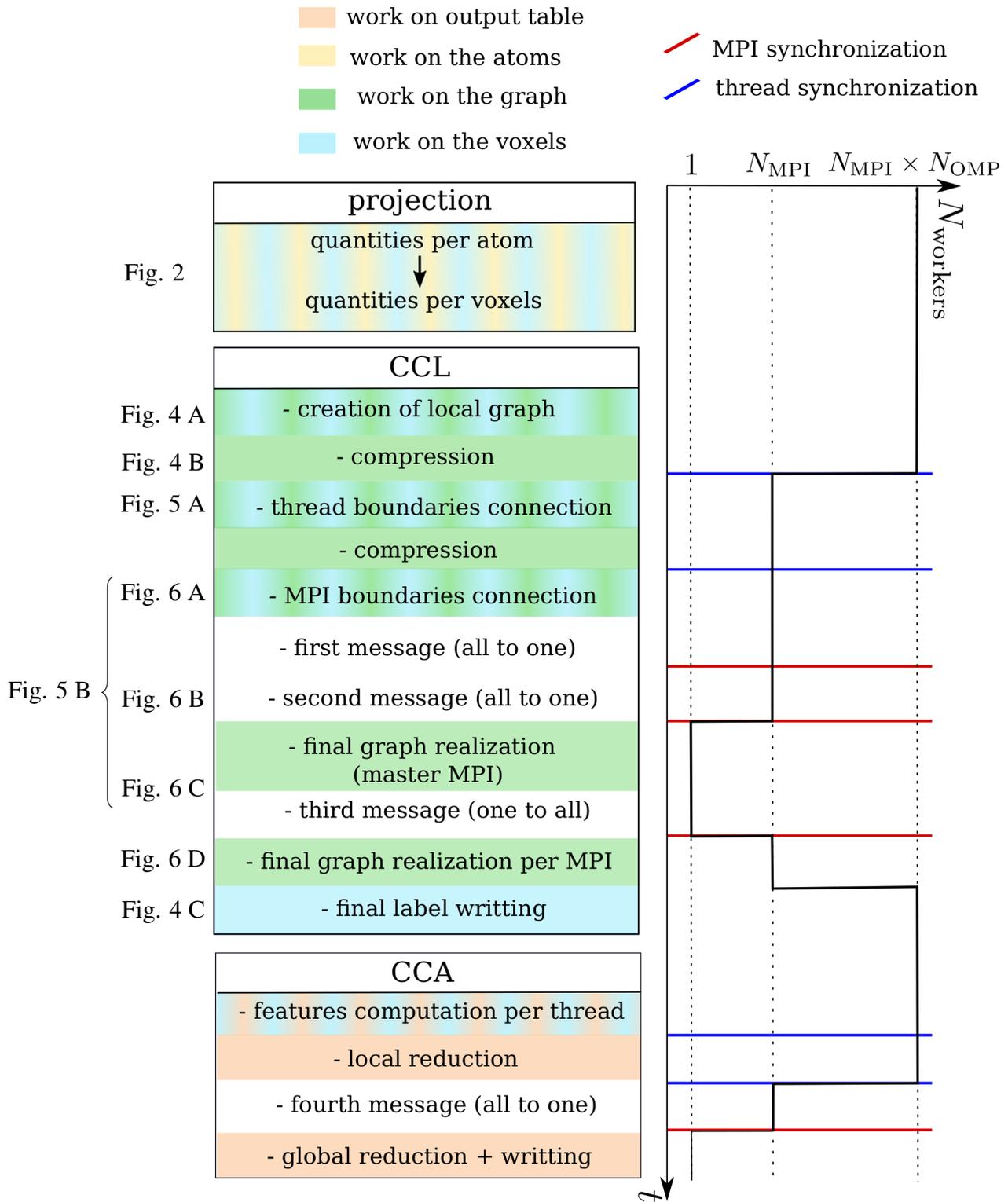


Figure 3: The schematic representation of the algorithm spans from the projection on the regular grid to CCL and CCA. The color represents the type of data format managed by the workers (atoms, voxels, graph vertices, table). Various synchronizations are highlighted by horizontal lines. An associated ideal Gantt chart of the workers is represented on the right side of the diagram, assuming perfect load balancing between workers.

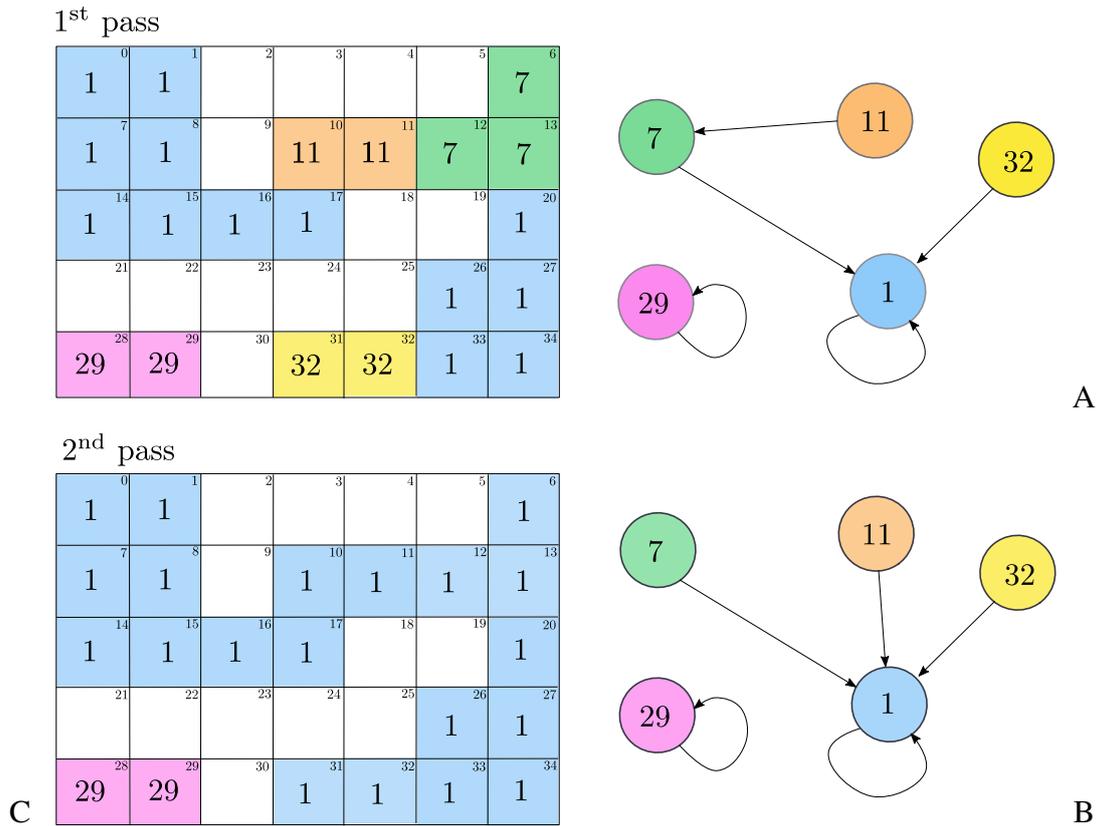


Figure 4: **A** - Connection phase. Each voxel has a unique $voxel_{id}$ computable by every thread of each MPI through the absolute position of the voxel. In Algorithm 1, the first pass of the CCL linearly examines the voxels of its domain. It assigns the label $voxel_{id} + 1$ to a voxel selected by the filter if it is isolated (e.g., voxel 6 in the illustration). If it is spatially in contact with an already labeled voxel, it takes on the label of the connected voxel (e.g., voxel 1 in the illustration takes on the 1th label). The graph is constructed accordingly. **B** - Compression phase. The graph is reduced to point directly towards the roots (the smallest label of the connected components). **C** - Second pass. Final assignment of root label to each voxel.

Algorithm 1 CCL first pass called by each OpenMP thread

```
for each voxel in work block do
  if (filter condition is true) then
     $label_{tmp} \leftarrow nb_{voxel}^{tot} + 2$ 
    for each voxel in scan mask do
      if ( $voxel\_root$  exists) & ( $voxel\_root < label_{tmp}$ ) then
         $label_{tmp} \leftarrow voxel\_root$ 
      end if
    end for
    if ( $label_{tmp} = nb_{voxel}^{tot} + 2$ ) then
       $label_{tmp} \leftarrow voxel_{id} + 1$ 
      Insert  $label_{tmp}$  as root in the equivalence table
    else
      for each voxel in scan mask do
        if ( $voxel\_root$  exists) then
           $voxel\_root \leftarrow label_{tmp}$ 
        end if
      end for
    end if
  end if
end for
```

union-find algorithm involves reducing the length of branches in the graph by making each vertex point directly to its root (Figure 4). The Union-Find strategy was chosen rather than the Suzuki equivalence management [43], based on the conclusion of [23], stating that it is the only viable strategy for parallel algorithms.

After all threads have completed their “union” step, a designated thread takes on the responsibility of connecting the individual equivalence tables (one for each thread) into a single global equivalence table per MPI (Figure 3 and 5 for thread boundary connection). To achieve this, the thread iterates over each voxel on the three outermost boundaries of the six faces of each work block. Subsequently, it scans the neighboring voxel on the opposite side of the boundary. If both the current voxel and its neighbor have labels, the label with the lower index becomes the root of the other and is then inserted into the respective local and global equivalence tables. Once each work block has been joined, every thread performs once more an “union” phase due to possible element insertions in their respective equivalence tables (2nd compression step in Figure 3). With the local regions of each MPI process now temporarily labeled and connected, the processes are ready to initiate communication for establishing the final graph of connected components and assigning definitive labels to the voxels (refer to section 3.4).

3.4. MPI communications and boundaries connections

CCL necessitates the construction of a global graph, a problem known to require global synchronization when solved in parallel [44, 45]. This implies a negative impact on weak scaling

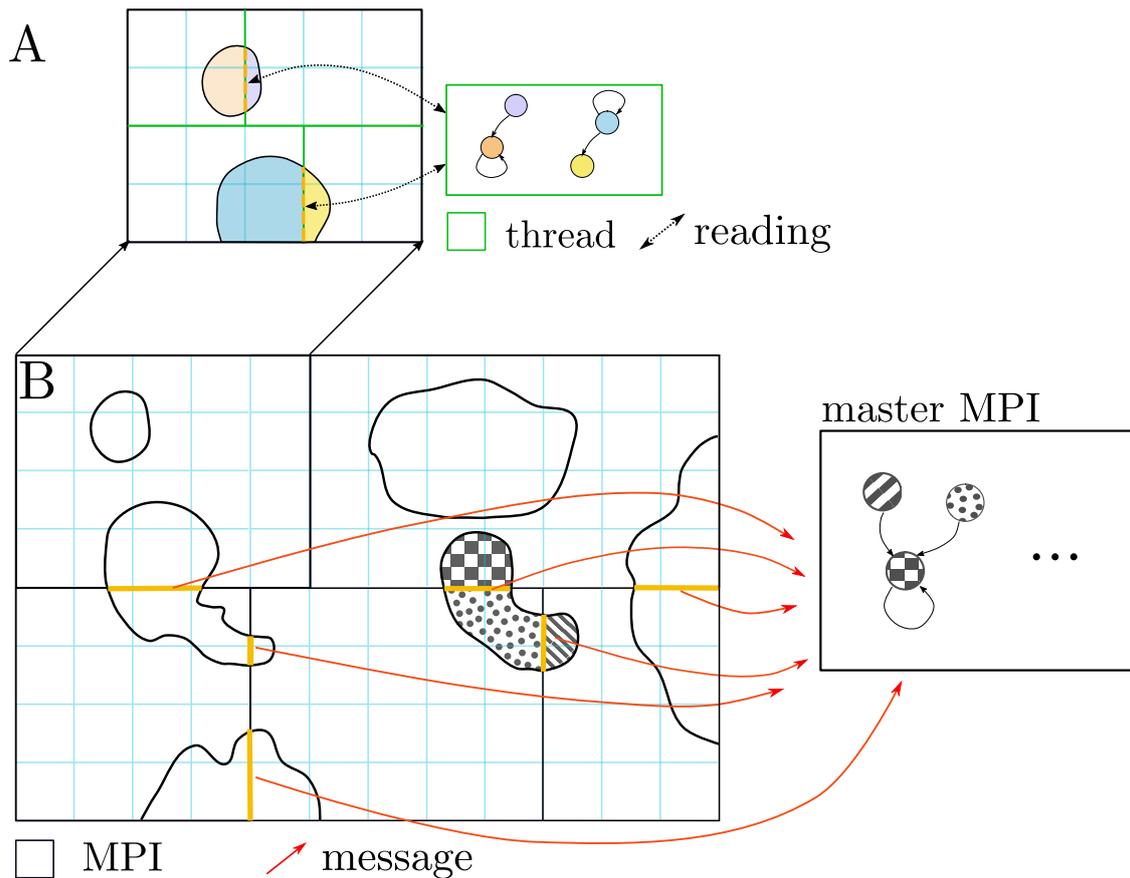


Figure 5: A schematic illustration of the 2D boundary connections when an aggregate is spread across different threads or MPIs. **A** - Local graphs are connected at the thread level and, after a compression step, produce a "global" graph per MPI. **B** - At the MPI level, the boundary connection requires a two-step method. The first step is represented on the left, each MPI sends a message to the master MPI containing an equivalence table, which can be summarized schematically with graphs, containing the boundary connection information. This information is subsequently used in a second step, here on the right, so that the master MPI connect those graphs and send back to each MPI the roots corresponding to each of their aggregates (Figure 6).

performance, especially in distributed memory settings where synchronization relies on communications. To mitigate this impact, it is crucial to minimize the size of communications. As mentioned earlier (see 2.2), solving a generic unstructured graph through CCL involves multi-pass algorithms, resulting in multiple rounds of communications in distributed memory settings. However, in our case, the structured nature of our 3D image data allows us to leverage spatial proximity information among voxels to connect MPI boundaries. This optimization enables us to reduce communications to a single round of $3 N_{\text{MPI}}$ messages (see Figure 3). Our scheme is straightforward, involving two all-to-one communications, followed by a final one-to-all communication step. The first all-to-one communication determines the size of the second all-to-one reduction, as well as the size of the last one-to-all communication. Saturation of the MPI performing the reduction is avoided thanks to our limited number of MPI processes (ranging from 1 to 4 processes per node on a bi-socket AMD Milan) and the “compression” step performed by the threads, which significantly reduces the information to be processed. The second all-to-one message contains an equivalence table resulting from a “2D” CCL map created by each MPI at its boundaries, connecting its surface’s voxels to those of the ghost layer. This 2D CCL follows the algorithm 1 and relies on the existence of an absolute $voxel_{id}$ for each voxel. The map establishes an equivalence between the absolute $voxel_{id}$ of the ghost layer, the ones of the MPI surface voxels, and their corresponding roots (Figure 6, message of MPI n and m). Here, MPI n and m shared a boundary. Each message points to a distinct root specific to its originating MPI (Bold text on the diagram).

The master MPI performing the graph reduction establishes the connection between 3D labels using the matching 2D labels and solves the resulting graph by updating it with each insertion. Subsequently, this MPI propagates the final label roots back to the other MPIs through the last one-to-all communication (Figure 6 one-to-all final message). Once this step is completed, threads in each MPI process perform a second CCL pass on their graph, replacing their temporary labels with the final ones. This ensures that each process is aware of the final graph. The renumbering of labels helps limit the number of labels used at each time step, which will be beneficial for future extensions of the 4D CCL (time tracking of the aggregates). Additionally, it simplifies the communication steps for the subsequent feature extraction stage. It is important to note that this *all-to-one* communication involves only minimal data, specifically containing information about aggregates located at the 2D boundaries of MPI processes. This surface-level information is significantly smaller compared to the 3D connection graph within each MPI area. Our approach is designed to maintain a constant communication cost for operations, ensuring constant algorithmic complexity involving a single phase of collective communication which concentrates final labels on one MPI process.

3.5. Connected Component Analysis

After assigning labels to the connected components, each MPI process identifies the number of connected components within its subdomain. With this information, each thread allocates memory buffers for the various physical properties that the user intends to compute. The length of each buffer is equal to the number of connected components in the MPI subdomain. Subsequently, each thread traverses its working block and computes various physical properties (such as volume, center of mass, velocity, sphericity, etc.) of its local connected components using equations (4) and

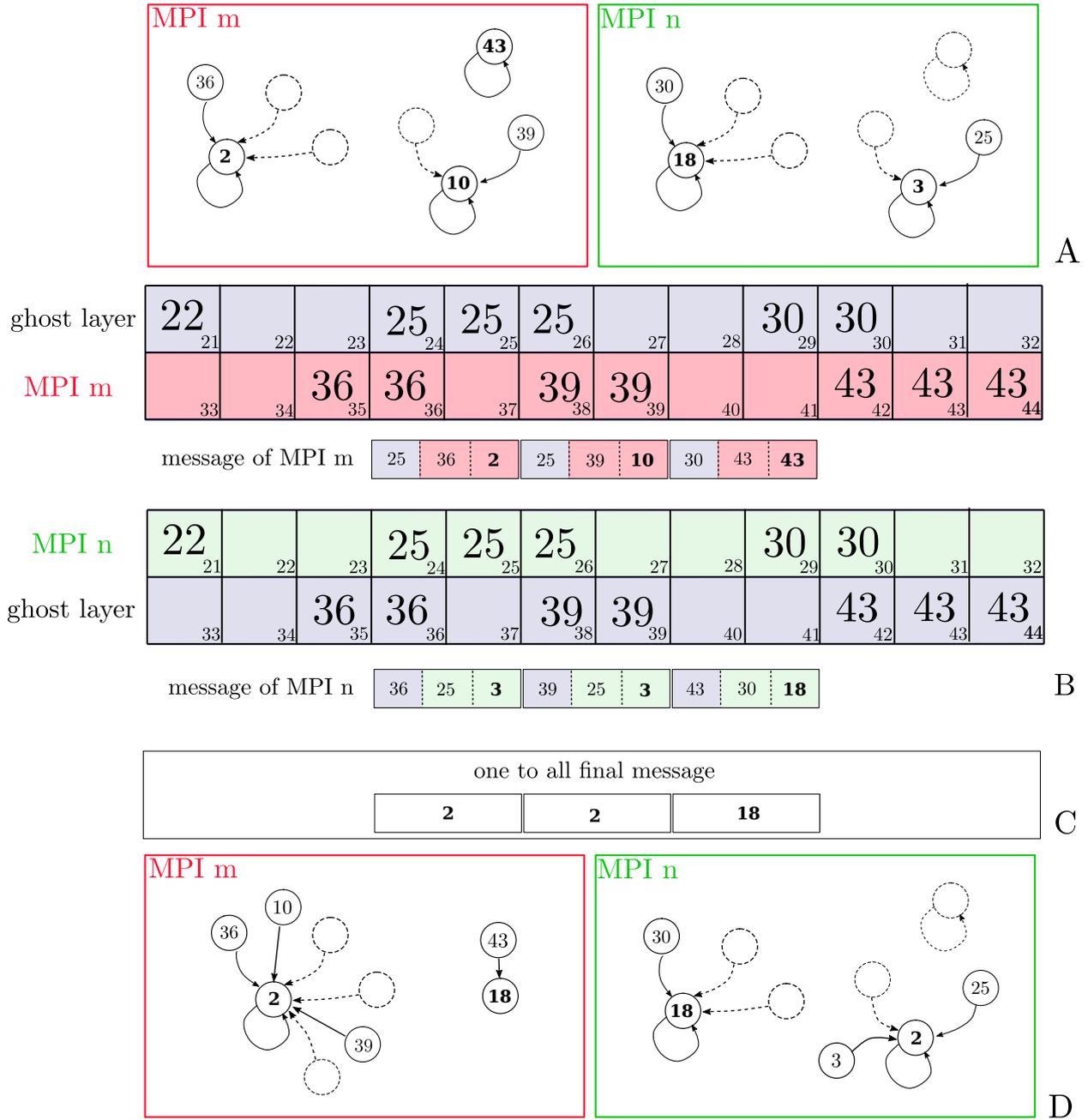


Figure 6: **A** - Elements of the local graph of two given MPIs, m and n **B** - Local equivalence table created by each MPI at its boundary. Each MPI has access to its layer of surface voxels as well as its ghost layer. Using the global $voxel_{id}$, a new connection is established by associating the label of each voxel, if connected, with the local root of the MPI. This list (the message from MPI m) is sent to the master MPI (see Figure 3). **C** - The master MPI performs a reduction and sends back a one-to-all message containing the new roots associated with each of these "surface" aggregates in the global frame of reference. **D** - This process enables each MPI to realize its final graph.

(5). Here, voxel j represents the j^{th} connected component, and atoms i represent the i^{th} voxel of the connected components. Following this computation, a reduction is performed on the threads' buffers, resulting in a single buffer sent to the primary MPI. The primary MPI then conducts a reduction to analyze the entire simulation and generates a comma-separated values (CSV) file containing all the information. Optionally, if the user chooses, the primary MPI can propagate its analysis back to all the other secondary MPIs, which can write this information into the regular grid for visualization purposes. The code uses the VTK library format for generating files in `.pvti` and `.vti` formats. Figure 9 was produced with this feature and visualized using ParaView [46].

4. Performance evaluation

4.1. Experimental setup

The supercomputer CEA-HF, currently ranked 30nd on the TOP500 list [47], was utilized for our benchmarks. It comprises 6330 bi-socket nodes hosting AMD EPYC 7763 64C 2.45GHz CPUs and interconnects with the Atos BXI V2 network. A total of 256 nodes were used for our benchmarks, accounting for 32768 cores. The code was compiled with `icpc 20.0.4`, optimization flag `-O3 -march=core-avx2`, and linked to the MKL library. Although we are using an AMD platform, compiling our code with Intel's `icc` resulted in a 40% efficiency increase compared to `g++ 11.2.0` with optimization flag `-O3 -march=native`. The impact of this divergence on performance was investigated using `maqao` [48], and it was found that the mathematics library used by `gcc` performs less than the Intel MKL one. Our results are then presented using an `icc`-compiled binary, and future work will be dedicated to exploring other compilers such as `aocc` and `clang++`. Our analysis of the spalling case was performed on a 300 iterations MD simulation with a 1 fs timestep, as shown in Figure 9. An EAM potential [49] was used for all benchmarks, as it is recognized to be one of the most computationally efficient [8], while accurately reproducing the relevant physical properties of metals. The selection of the EAM potential could accentuate the limitations of our algorithm owing to its extremely low computational cost. However, a minimal additional overhead has been measured, which would become truly negligible when employing any other more resource-intensive potential.

4.2. MPI vs OpenMP comparison

We conducted an initial benchmark to explore the impact of different MPI process and OpenMP thread distributions on our hybrid code, which utilizes both distributed and shared memory. An initial workload of 81.9 million atoms was distributed across 4 nodes, approximately 20 million atoms per node, for this benchmark, involving 58.9 million voxels. It is noteworthy that this initial workload is already 4 and 80 times larger than similar spallation studies on Tantalum [50, 51]. As illustrated in Figure 7, each combination of MPI \times OMP workers per node is presented on the x-axis, while the left-side y-axis is dedicated to the speed-up or slow-down relative to the first decomposition of (1 MPI \times 128 OMP). The higher the point, the greater the speed-up. If the code were agnostic to the MPI \times OMP decomposition, horizontal lines would have been obtained. In the current scenario, performance strongly correlates with MPI \times OMP across the nodes. The optimal performance for MD simulations and analysis was achieved with different MPI and thread configurations, showcasing improved performance with higher numbers of MPI. MD, being more

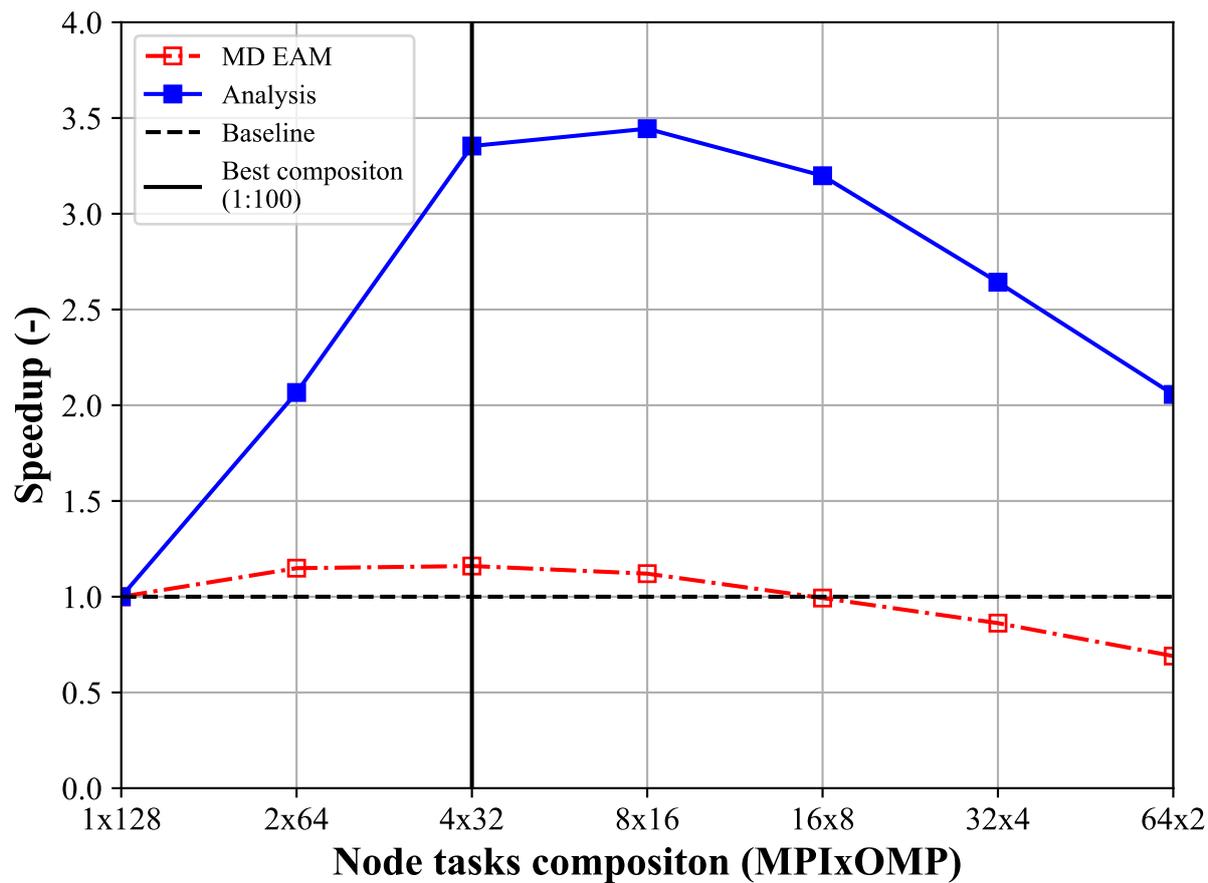


Figure 7: Benchmark of a 81.9 M atoms spallation case with an EAM potential and 58.9M voxels for the analysis. The benchmark is run on 4 nodes of bi-socket AMD milan EPYC 7763 64-Core. 300 iterations are timed for the MD (red) and 30 iterations are timed for the analysis (blue). The total time spent for various MPI \times OMP decompositions on a node is divided by the time taken for the decomposition (1 MPI \times 128 OMP). Higher is better.

compute-bound with threads computing on their local data, is less susceptible to memory-bound limitations. Conversely, communications become a bottleneck for the simulation, and the code benefits most from threads that are local to sockets. Ultimately, the best results were obtained with 4 MPI workers and 32 OMP threads each. In the analysis, a plateau is observed between configurations of 4×32 and 8×16 (8 MPI processes per node and 16 threads per MPI process). This underscores the importance of considering the specific workload and algorithmic requirements when determining the optimal parallelization scheme. Our work utilized Non-Uniform Memory Access (NUMA) nodes with 5 different levels of memory access. While our thread programming model is unaware of this, we observed that the analysis greatly benefited from memory placement across 3 of those levels, confirming that voxel-based CCL algorithms are memory-bound [23]. This also illustrates that our collective communication phase is not a performance bottleneck limiting scalability. Increasing the number of MPIs with a fixed problem size increases communications and data transmission due to an increase in 2D surfaces. In parallel, reducing the number of OpenMP threads decreases the workforce managing communications. Despite these adjustments, our performance improved from 1×128 and 4×32 (Figure 7), highlighting the relevance of our communication strategy.

To prioritize MD simulations, we have decided to use the MPIxOMP distribution that best supports them ($4 \text{ MPI processes} \times 32 \text{ threads}$) for our upcoming strong and weak scaling analyses, as well as future benchmarks. This decision is based on the crucial role of MD in our simulations.

4.3. Strong and Weak Scaling

Two constant workloads were employed for the strong scaling benchmark: the previous one from section 4.2 with 81.9 million atoms and 58.9 million voxels, and a 8 times larger one with 655.2 million atoms and 471.6 million voxels. We initially investigated the speedup achieved through an increase in the MPI worker number. The results are illustrated in Figure 8-A, where the y-axes represent the execution time in seconds per iteration, and the x-axis indicates the number of cores (with a $4 \text{ MPI} \times 32 \text{ threads}$ ratio per nodes for each simulation), both axis are on a logarithmic scale. In the ideal scenario of strong scaling, the expected outcome would be a linearly decreasing line (dotted lines on Figure 8-A). Deviations from perfect scalability are observed, and the scalability is better with a larger workload (655.2M atoms) for both the MD codes and the analysis. The analysis performed slightly better than the MD code for the bigger workload in terms of scaling and execution time. Whereas the analysis has an initial execution time lower than the MD for the original workload but scale much less than the MD in this case and end up with a slightly higher execution time than the MD. A careful study of the impact of communications on the global connection cost will be presented in Section 5.2.

For the investigation of weak scaling, workloads need to be adjusted according to the number of workers. The workload per processor is kept constant, ideally resulting in constant computation time even as more processors are added, indicating good scalability (dotted line Figure 8-B). An initial workload of 81.9 million atoms/58.9 million voxels across 4 nodes was considered, successively doubling up to a workload of 5.2 billion atoms/3.7 billion voxels on 256 nodes, i.e., 32768 cores. Performances are presented in Figure 8-B with the same axes as for the strong scaling. Near-ideal behavior is achieved by MD simulations in weak scaling, a well-known result attributed to the short-range potential used, making MD-specific computations local. This good results is also

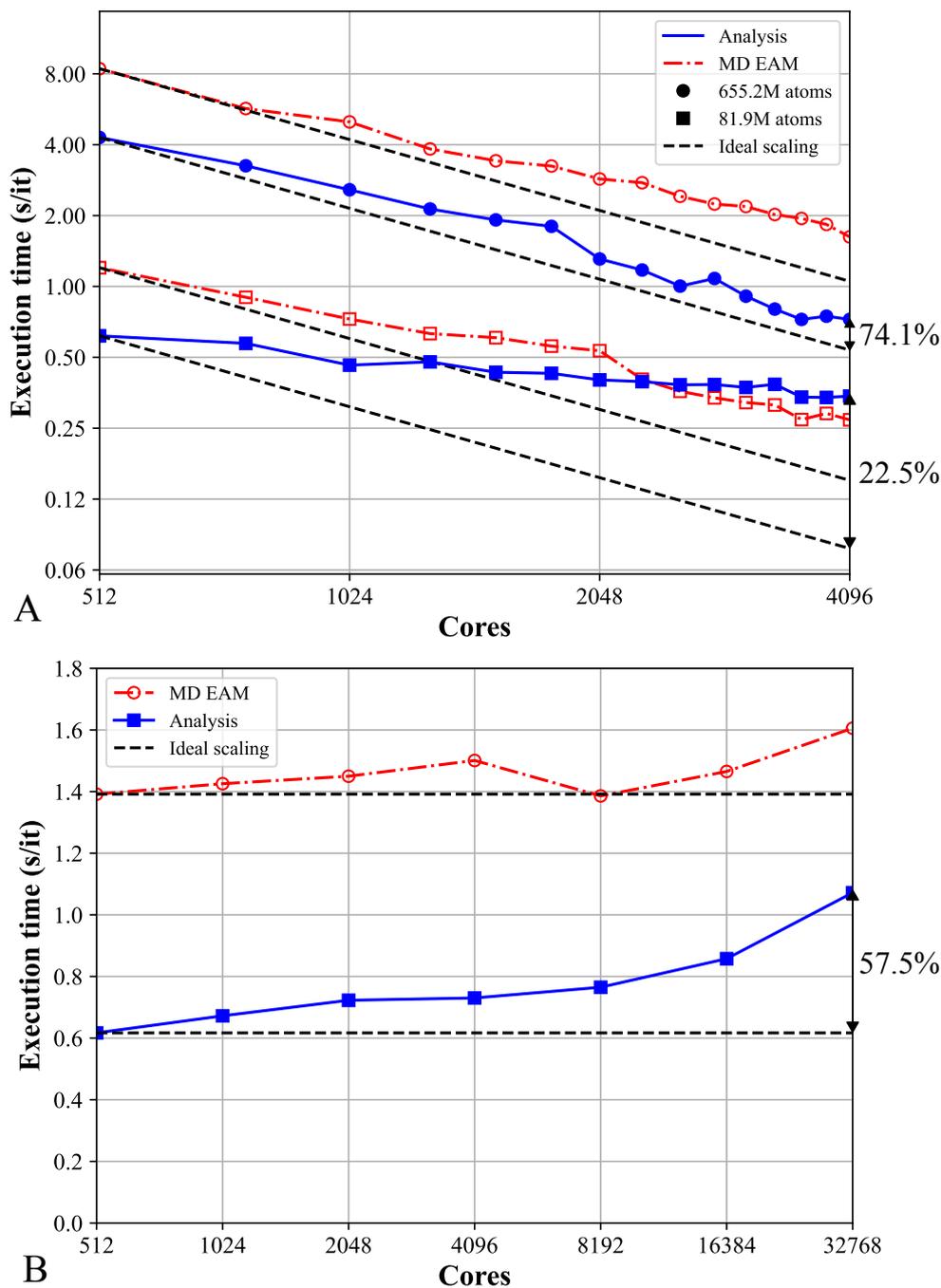


Figure 8: **A-B** - Benchmarks run with a $4 \text{ MPI} \times 32 \text{ OMP}$ repartition per node on bi-socket AMD milan EPYC 7763 64-Core nodes. Values are averaged over 300 steps for MD (red) and 30 steps for analysis (blue). The dotted lines represent the ideal behavior. The final efficiencies achieved by the analysis are denoted in black. **A** - Strong scaling tests were conducted for two different workloads: 81.9 million atoms (squares), the same as in Figure 7, and 655.2 million atoms (circles), that is 8 times bigger. Log scale on both axis. **B** - Weak scaling test results from the 81.9 million atoms workload of Figure 7 on 512 cores, up to a 5.2 billion atoms workload on the 32 768 cores. Log scale on x-axis.

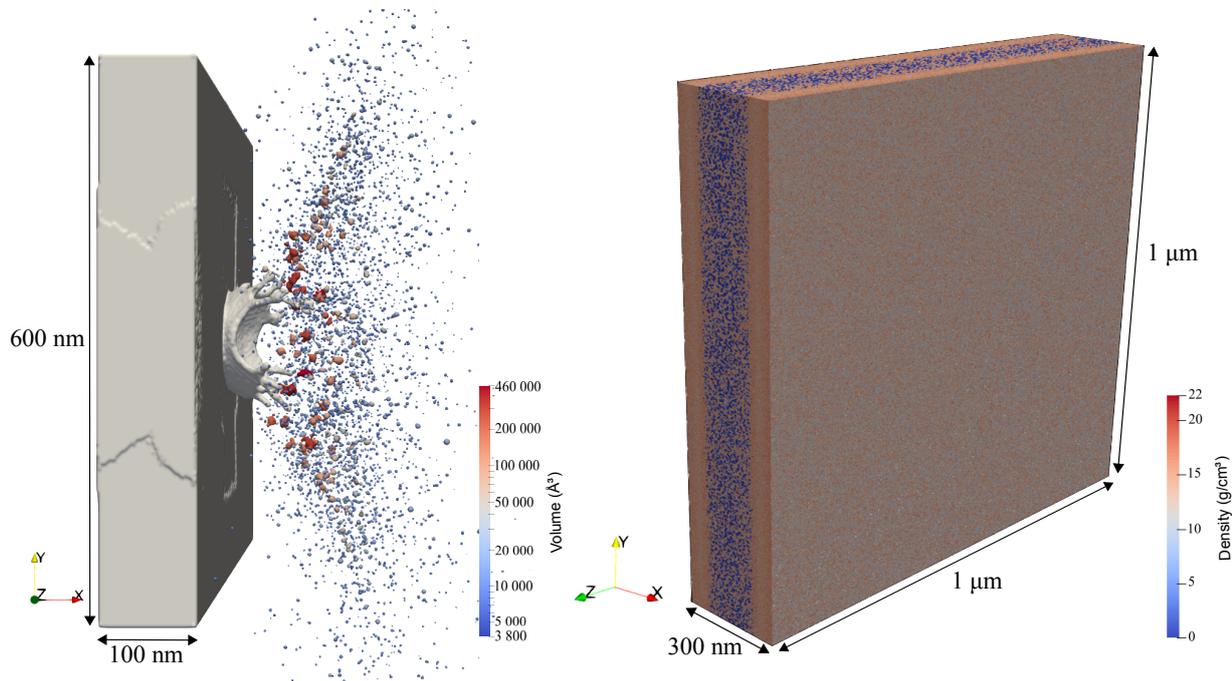


Figure 9: Splashing experiment of 1.2B atoms projected on 29.4 B voxels. Spall Fracture experiment of 8.3B atoms projected on 5.2B voxels.

imputable to the quality of the communication scheme used to exchange ghost layers. While the analysis did not exhibit ideal weak scaling, reaching an endpoint of 57.5% efficiency with 32 768 cores represents a significant accomplishment. The difficulty in scaling correctly in weak scaling is inherent to the CCL algorithm, which, being a global problem, inevitably experiences a divergence in communication cost with the increasing number of connected components to treat. Our results, despite executing across a much larger number of cores (2^{15} vs 2^9), demonstrate equivalent efficiency in strong scalability (74.1% vs 79.13%) and a much better weak scalability (57.5% vs 8.13%) compared to a recent equivalent approach in the literature using an asynchronous version of the 3D algorithm [29]. The relative analysis cost, in comparison to that of the MD simulation, will be discussed in Section 5.3.

5. Results and discussions

5.1. Two Physical Application Examples

Two numerical experiments, each exploring the algorithm’s capabilities in different scenarios, void or aggregates detection, were conducted.

The first experiment simulated a splashing scenario on the CEA-HF supercomputer using 256 nodes and an MEAM potential [52]. In this simulation, 1.2 billion tin atoms were modeled. A drop with a diameter of 50 nm impacted a solid surface measuring $600 \times 600 \times 100$ nm at a

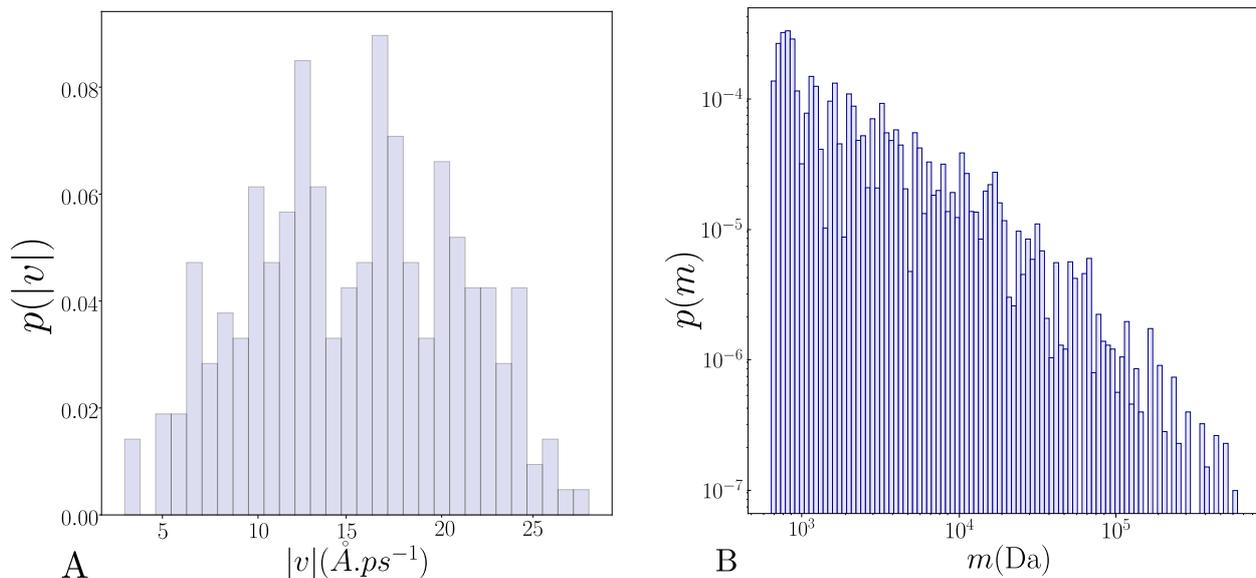


Figure 10: **A** - Probability distribution of the velocity norm for the aggregates detected during the splashing experiment. **B** - Probability distribution of the mass for the aggregates detected during the splashing experiment.

velocity of 2000 m.s^{-1} , resulting in the generation of droplet jets (Figure 9 A). The projection was performed on 29.41 billion voxels with a spacing of 4 \AA and a density threshold (ρ_{th}) of $0.68 \rho_{Sn}$. The algorithm provided the position, velocity, masses, and volumes of all generated droplets at any time during the simulation. The distribution of velocity and masses of the aggregates can be visualized (Figure 10 A-B).

The second experiment involved spall fracture, conducted on a Tantalum sample with dimensions of $(1 \times 1 \times 0.2) \mu\text{m}^3$, containing 8.3 billion atoms. The simulation utilized an EAM potential [49]. In this setup, a piston impacted the sample at a velocity of 2078 m/s. The convergence of the two release waves, initiated by the removal of the piston and the reflection of the shock on the opposite face, occurs towards the center of the sample, leading to negative pressure. This phenomenon induces the formation of pores within the medium (nucleation), which subsequently grow and coalesce until, in some instances, the sample fractures (see Figure 9 B). The projection was performed on 5.2 billion voxels with a voxel size of $s = 3.3 \text{ \AA}$ and a density threshold of $\rho_{\text{threshold}} = 10^{-4} \rho_{\text{Tantalum}}$. The pore volume distribution at a given time follows a fat-tailed distribution (Figure 11 B). The total number of detected pores over time (Figure 11-A) follows an initial phase of exponential growth dominated by pore nucleation, and then decreases as coalescence becomes the dominant process. A comprehensive parametric physical study is planned for the near future.

5.2. Workers' Gantt chart

To justify our communication strategy for the final global connection compared to the asynchronous communication choice that could have been made, we extract the workers' Gantt chart (Figure 12 - (upper)) from the experimental point depicted as a vertical line in Figure 7 and the starting point of both strong (81.9M atoms workload) and weak scaling benchmark (Figure 8).

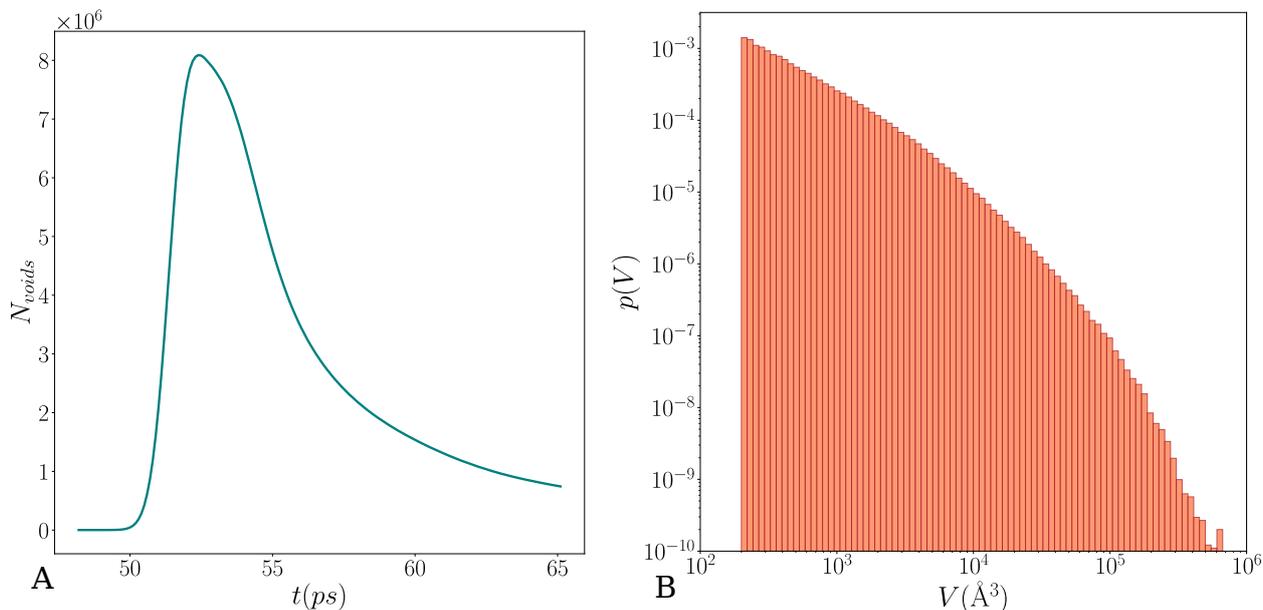


Figure 11: **A** - Number of pores detected for each time step of the spall fracture simulation. **B** - Probability distribution function of pore volumes at $t = 61.5 ps$

The x-axis represents the time of computation, while the y-axis has one line per worker. Colored phases represent work phases, while white ones represent stall phases (wait or communication) for the workers. The time spent by the workers in other MD routines is shown in grey. An integrated version is presented below (Figure 12 - (lower)), where the actual number of workers not stalled is plotted against time. It represents the sum of active workers across the time of the upper figure. A comparison with the theoretical chart (Figure 3) highlights several things.

First, we observe a significant load imbalance in the CCL part due to domain decomposition that favors MD performances based on the atoms' number and density in each MPI domain. With this policy, four of our MPI domains have to handle the massive void (18.8% of the simulation space) introduced by the piston that generated the shock in the simulation (see Sec. 5.1). This puts a lot of stress on the boundary connections handled by one thread per MPI and increases the CCL time proportion to 33.4% of the total analysis time; a perfectly load-balanced case could reduce this proportion to 5.5%. Nevertheless, this domain decomposition policy has the advantage of greatly load balancing the projection part of the analysis, which represents the major part of the analysis (58%).

Secondly, we observe that the final graph realization on one thread takes only 1.6% of the total analysis time. We can conclude that the scalability of our CCL algorithm is intrinsically limited by the connection at workers' boundaries, the MD load balancing, and Figure 12 confirms that our choice of communication strategy is efficient. An improvement for CCL could be to parallelize the local boundaries connections with every thread and join them with a pyramidal merge [23]. To improve the overall cost of analysis, a focus should be made on the projection part that is heavily computational and could benefit from GPU porting.

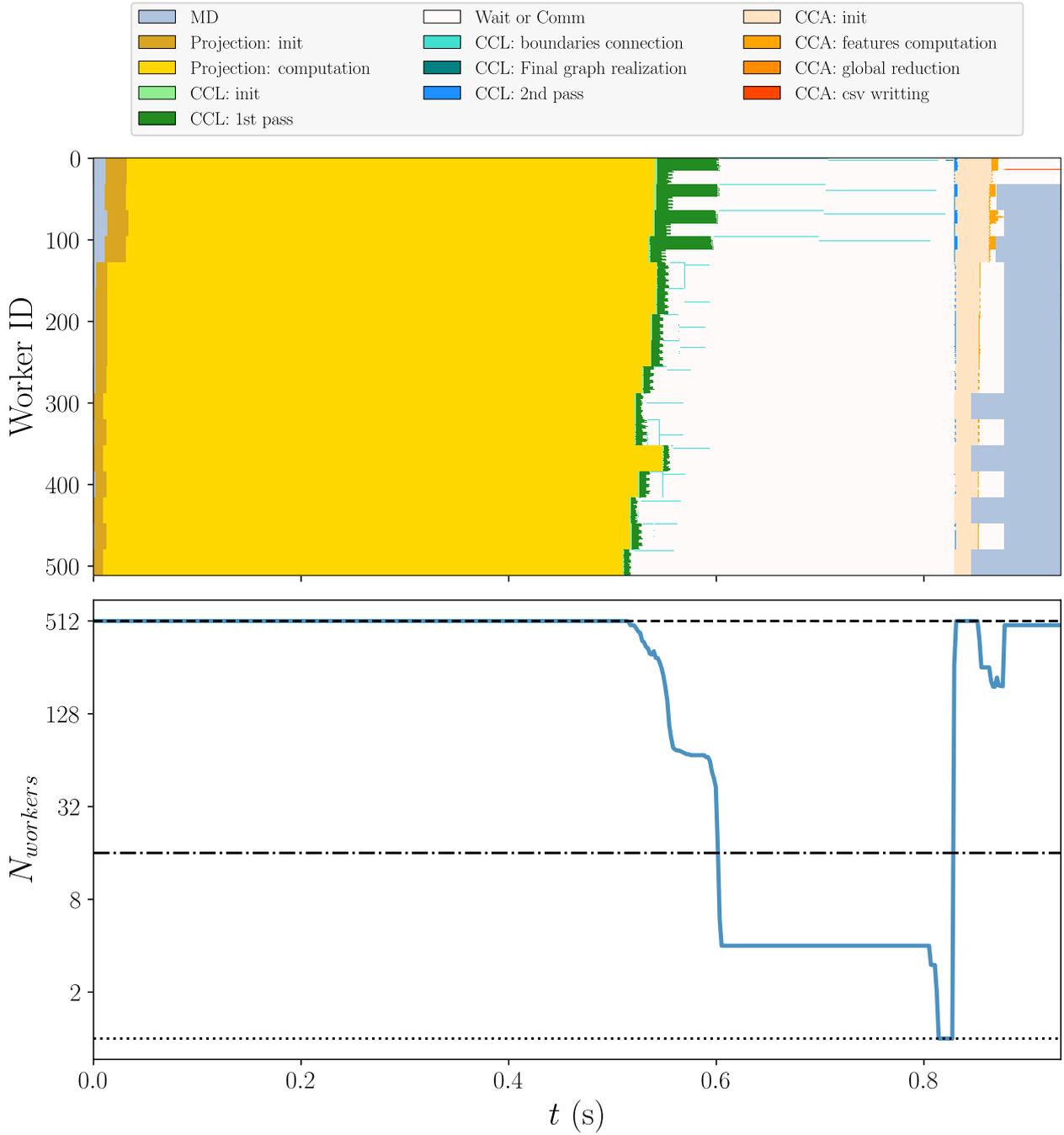


Figure 12: **Upper** - Gantt chart depicting the schedule of workers throughout the analysis, with color coding representing their respective activities. Benchmark on the 81.9M atoms workload run with 4 nodes and a $4 \text{ MPI} \times 32 \text{ OMP}$ repartition per node. **Lower** - The total number of workers not in a stall operation (Wait or Comm) over time is depicted. The lower dotted line corresponds to $N = 1$ worker. The intermediate line corresponds to $N = N_{MPI}$ workers, and the upper line corresponds to $N = N_{MPI} \times N_{OMP}$ workers. It corresponds to the integration over the workers not in a stall operation of the upper graph.

5.3. Analysis Cost

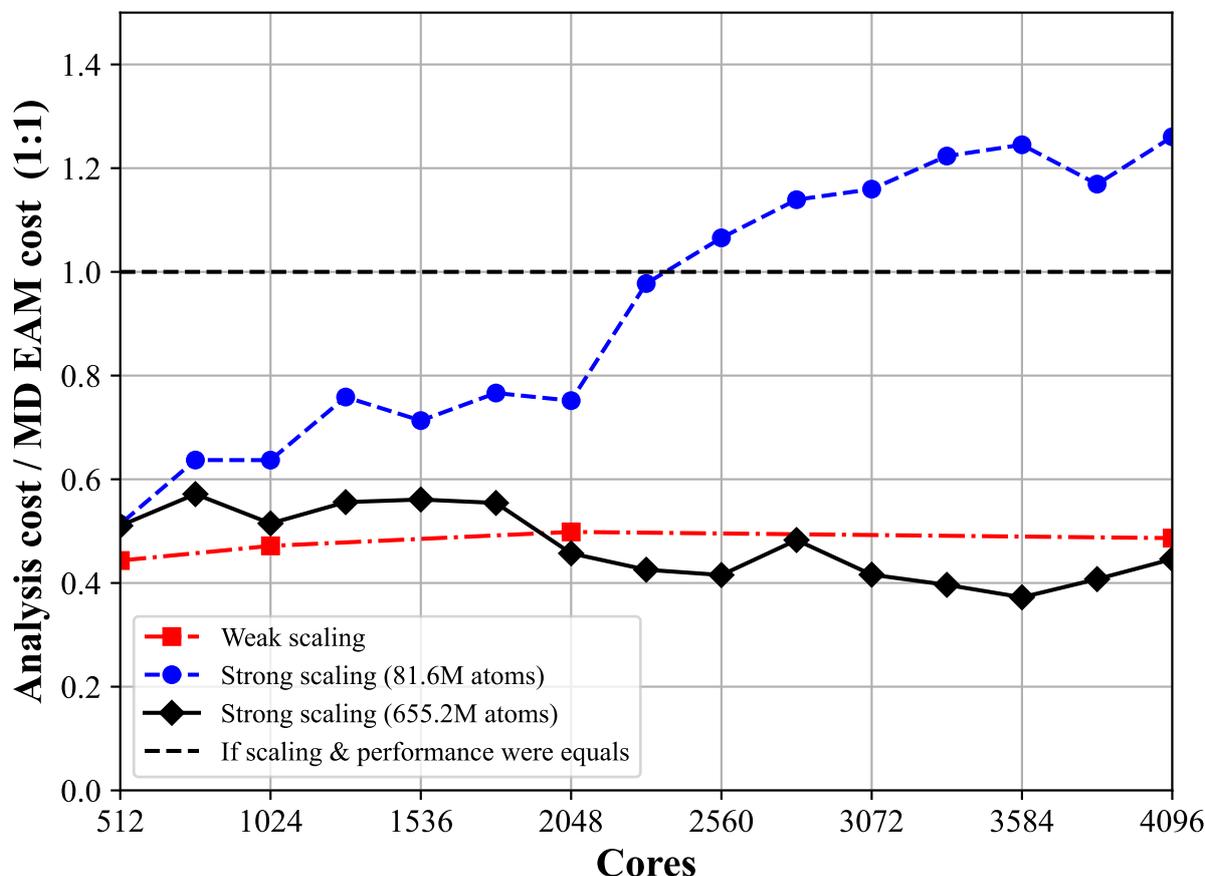


Figure 13: The ratio between the cost of one iteration of the analysis and one iteration of Molecular Dynamics is plotted. Data from both weak and strong scaling studies are presented. If both codes, MD and analysis, were scaling and performing exactly the same, the data would lie on the horizontal dotted line. Below the dotted line, the analysis is better performing and above the MD is.

We examine the comparative costs of our analysis and the MD simulation, as illustrated in Figure 13. The analysis cost, quantified as the ratio between the time required for one analysis iteration and that of a single MD iteration, is depicted on the y-axis, while the number of cores is plotted on the x-axis. Data from both strong and weak scalings are presented. Ideally, if both analysis and MD had similar scaling behavior, the ratio would remain constant. However, a slight increase is observed for the smaller workload in strong scaling, indicating lower scaling on the analysis side. In average our analysis scale as the MD and an iteration cost the half of a MD iteration with an EAM potential. Nevertheless, the analysis only incurs 1.3 times the time of MD in the worst-case scenario, which is acceptable given the cost-effective nature of the EAM potential used in the present work. In comparison, Machine Learning Interatomic Potentials (MLIP) employed in MD simulations are approximately 100 times more expensive than EAM potentials. Consequently, the relative cost of our analysis would decrease from 1.3 to 0.013, rendering it negligible,

especially if the analysis is run at a lower frequency, typically one analysis every 100 steps of MD. Performing our analysis at each MD iteration is physically unnecessary. While the MD timestep is fixed to ensure energy conservation, the analysis frequency has to align with the characteristic time of the physics being studied. Very fast processes will be about 100 MD timesteps, reducing the analysis relative cost. Moreover, such phenomena may only occur at specific points during the simulation. For example, spall fracture involving pore nucleation and growth typically requires a few hundred thousand timesteps for the shockwave to propagate, reflect on free surfaces, and initiate pore formation, which is the only part where an analysis is needed.

6. Conclusion

We introduce a high-performance in-situ analysis toolchain for MD, demonstrating exceptional efficiency with a mere 1.3 percent overhead in the worst-case scenario. Successfully applied to real-world simulations with billions of atoms, our toolchain facilitates future novel physical studies. We enhance our capabilities by introducing a hybrid parallelization approach of Connected Component Labelling (CCL) algorithm in 3D, enabling the efficient processing of large-scale datasets. Future plans involve extending our algorithm to 4D, introducing time as the fourth dimension for automatic detection of nucleation, coalescence, fragmentation, and dissipation events. This extension will enable tracking connected components and their properties over time, offering valuable insights into dynamic processes. With its interdisciplinary nature and relevance across multiple fields, our toolchain holds the potential to benefit a broad audience. Easily adaptable to various N-body simulations, ranging from classical Molecular Dynamics to Smoothed Particle Hydrodynamics and the Discrete Element Method, it can seamlessly transform into a module capable of executing 3D CCL on different data types, such as X-ray tomography or outputs of hydrocodes. This adaptability relies on their alignment with a regular 3D grid and the provision of physical insights through binary analysis.

CRedit authorship contribution statement

Killian Babilotte: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – original draft & editing, Visualization. **Alizée Dubois:** Conceptualization, Methodology, Formal analysis, Writing – original draft & editing, Visualization, Supervision. **Thierry Carrard:** Conceptualization, Methodology, Software, Validation, Resources. **Paul Lafourcade:** Software, Validation, Writing - Review & Editing. **Laurent Videau:** Supervision. **Jean-François Molinari:** Writing - Review & Editing, Supervision. **Laurent Soulard:** Conceptualization, Methodology, Validation, Resources, Writing - Review & Editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

Software implementation and data availability

Data will be made available upon request. To date, the ExaNBody HPC platform is already available under an APACHE 2 license at <https://github.com/Collab4exaNBody/exaNBody.git>. The exaStamp code will be made available at the end of 2024 under an APACHE 2 license at <https://github.com/Collab4exaNBody/exaStamp.git>, with the corresponding documentation available at https://github.com/Collab4exaNBody/doc_exaStamp.git. These two GitHub repositories are currently private but can be made available upon request. Additional applications are currently in the process of being released.

Acknowledgements

This work was supported by CEA/DAM. We thank François Bottin and Remi Lemaire for careful proofreading and insightful advices.

References

- [1] L. A. Zepeda-Ruiz, A. Stukowski, T. Oettel, V. V. Bulatov, Probing the limits of metal plasticity with molecular dynamics simulations, *Nature* 550 (2017) 492–495.
- [2] D. F. Puleri, S. Roychowdhury, P. Balogh, J. Gounley, E. W. Draeger, J. Ames, A. Adebisi, S. Chidyagwai, B. Hernández, S. Lee, et al., High performance adaptive physics refinement to enable large-scale tracking of cancer cell trajectory, in: *2022 IEEE International Conference on Cluster Computing (CLUSTER)*, IEEE, 2022, pp. 230–242.
- [3] E. Suchyta, J. Y. Choi, S.-H. Ku, D. Pugmire, A. Gainaru, K. Huck, R. Kube, A. Scheinberg, F. Suter, C. Chang, et al., Hybrid analysis of fusion data for online understanding of complex science on extreme scale computers, in: *2022 IEEE International Conference on Cluster Computing (CLUSTER)*, IEEE, 2022, pp. 218–229.
- [4] M. Wicklein, S. Ryan, D. White, R. Clegg, Hypervelocity impact on cfrp: Testing, material modelling, and numerical simulation, *International Journal of Impact Engineering* 35 (2008) 1861–1869.
- [5] S.-y. Ren, Z.-z. Gong, Q. Wu, G.-m. Song, Q.-m. Zhang, P.-l. Zhang, C. Chen, Y. Cao, Satellite breakup behaviors and model under the hypervelocity impact and explosion: A review, *Defence Technology* (2022).
- [6] X. Yang, X. Zeng, F. Wang, J. Ding, H. Zhao, B. Xue, Spallation fracture dependence on shock intensity and loading duration in single-crystal aluminum, *Computational Materials Science* 210 (2022) 111060.
- [7] S. Karamati, C. Hughes, K. S. Hemmert, R. E. Grant, W. W. Schonbein, S. Levy, T. M. Conte, J. Young, R. W. Vuduc, “smarter” nics for faster molecular dynamics: a case study, in: *2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, IEEE, 2022, pp. 583–594.
- [8] A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in’t Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, et al., LAMMPS—a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales, *Computer Physics Communications* 271 (2022) 108171.
- [9] S. Plimpton, Fast parallel algorithms for short-range molecular dynamics, *Journal of Computational Physics* 117 (1995) 1–19.
- [10] K. Kadau, T. C. Germann, P. S. Lomdahl, Molecular dynamics comes of age: 320 billion atom simulation on bluegene/l, *International Journal of Modern Physics C* 17 (2006) 1755–1761.
- [11] T. C. Germann, K. Kadau, Trillion-atom molecular dynamics becomes a reality, *International Journal of Modern Physics C* 19 (2008) 1315–1319.
- [12] N. Tchipev, S. Seckler, M. Heinen, J. Vrabc, F. Gratl, M. Horsch, M. Bernreuther, C. W. Glass, C. Niethammer, N. Hammer, et al., Twetris: Twenty trillion-atom simulation, *The International Journal of High Performance Computing Applications* 33 (2019) 838–854.
- [13] A. Stukowski, Visualization and analysis of atomistic simulation data with ovito—the open visualization tool, *Modelling and simulation in materials science and engineering* 18 (2009) 015012.

- [14] J. Hoshen, R. Kopelman, Percolation and cluster distribution. i. cluster multiple labeling technique and critical concentration algorithm, *Phys. Rev. B* 14 (1976) 3438–3445.
- [15] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise., volume 96, 1996, pp. 226–231.
- [16] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, X. Xu, DbSCAN revisited, revisited: why and how you should (still) use dbSCAN, *ACM Transactions on Database Systems (TODS)* 42 (2017) 1–21.
- [17] S. Plimpton, LAMMPS pre-release 22 december 2022, <https://github.com/lammps/lammps/pull/3405>, 2022.
- [18] M. Krone, J. Stone, T. Ertl, K. Schulten, Fast Visualization of Gaussian Density Surfaces for Molecular Dynamics and Particle System Trajectories, in: M. Meyer, T. Weinkauff (Eds.), *EuroVis - Short Papers*, 2012.
- [19] A. Stukowski, Ovito release 3.8.1, identification of volumetric regions using the gaussian density method, 2023.
- [20] L. He, Y. Chao, K. Suzuki, K. Wu, Fast connected-component labeling, *Pattern Recognition* 42 (2009) 1977–1987.
- [21] L. He, Y. Chao, K. Suzuki, A run-based one-and-a-half-scan connected-component labeling algorithm, *International Journal of Pattern Recognition and Artificial Intelligence* 24 (2010) 557–579.
- [22] N. Ma, D. G. Bailey, C. T. Johnston, Optimised single pass connected components analysis, in: 2008 International Conference on Field-Programmable Technology, IEEE, 2008, pp. 185–192.
- [23] L. Cabaret, L. Lacassagne, D. Etienne, Parallel light speed labeling: an efficient connected component algorithm for labeling and analysis on multi-core processors, *Journal of Real-Time Image Processing* 15 (2018) 173–196.
- [24] L. He, X. Ren, Q. Gao, X. Zhao, B. Yao, Y. Chao, The connected-component labeling problem: A review of state-of-the-art algorithms, *Pattern Recognition* 70 (2017) 25–43.
- [25] S. Lamm, P. Sanders, Communication-efficient massively distributed connected components, in: 2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS), IEEE, 2022, pp. 302–312.
- [26] R. Abdolazimi, M. Heidari, A. Esmaeilzadeh, H. Naderi, Mapreduce preprocess of big graphs for rapid connected components detection, in: 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), IEEE, 2022, pp. 0112–0118.
- [27] S. Lamm, P. Sanders, Communication-efficient massively distributed connected components, *IEEE IPDPS* 17 (2022).
- [28] J. Shun, L. Dhulipala, G. Blelloch, A simple and practical linear-work parallel algorithm for connectivity, in: *Proceedings of the 26th ACM symposium on Parallelism in algorithms and architectures*, 2014, pp. 143–153.
- [29] J. Xu, H. Guo, H.-W. Shen, M. Raj, X. Wang, X. Xu, Z. Wang, T. Peterka, Asynchronous and load-balanced union-find for distributed and parallel scientific data visualization and analysis, *IEEE Transactions on Visualization and Computer Graphics* 27 (2021).
- [30] L. Verlet, Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules, *Phys. Rev.* 159 (1967) 98–103.
- [31] E. Cieren, L. Colombet, S. Pitoiset, R. Namyst, Exastamp: a parallel framework for molecular dynamics on heterogeneous clusters, in: *Euro-Par 2014: Parallel Processing Workshops: Euro-Par 2014 International Workshops*, Porto, Portugal, August 25-26, 2014, Revised Selected Papers, Part II 20, Springer, 2014, pp. 121–132.
- [32] T. Carrard, R. Prat, G. Latu, K. Babilotte, P. Lafourcade, L. Amarsid, L. Soulard, Exanbody: A hpc framework for n-body applications, in: D. Zeinalipour, D. Blanco Heras, G. Pallis, H. Herodotou, D. Trihinas, D. Balouek, P. Diehl, T. Cojean, K. Furlinger, M. H. Kirkeby, M. Nardelli, P. Di Sanzo (Eds.), *Euro-Par 2023: Parallel Processing Workshops*, Springer Nature Switzerland, Cham, 2024, pp. 342–354.
- [33] M. P. Allen, D. J. Tildesley, *Computer Simulation of Liquids*, Oxford University Press, 1987.
- [34] R. Prat, T. Carrard, L. Soulard, O. Durand, R. Namyst, L. Colombet, Amr-based molecular dynamics for non-uniform, highly dynamic particle simulations, *Computer Physics Communications* 253 (2020) 107177.
- [35] L. Soulard, Micro-jetting: A semi-analytical model to calculate the velocity and density of the jet from a triangular groove, *Journal of Applied Physics* 133 (2023) 085901.
- [36] L. Soulard, T. Carrard, O. Durand, Molecular dynamics study of the impact of a solid drop on a solid target, *Journal of Applied Physics* 131 (2022) 135901.

- [37] R. J. Hardy, Formulas for determining local properties in molecular-dynamics simulations: Shock waves, *The Journal of Chemical Physics* 76 (1982) 622–628.
- [38] S. Root, R. J. Hardy, D. R. Swanson, Continuum predictions from molecular dynamics simulations: Shock waves, *The Journal of chemical physics* 118 (2003) 3161–3165.
- [39] R. J. Hardy, Atomistic formulas for local properties in systems with many-body interactions, *The Journal of Chemical Physics* 145 (2016) 204103.
- [40] J. A. Zimmerman, E. B. WebbIII, J. Hoyt, R. E. Jones, P. Klein, D. J. Bammann, Calculation of stress in atomistic simulation, *Modelling and simulation in materials science and engineering* 12 (2004) S319.
- [41] Y. Sumith, S. C. Maroo, A direct two-dimensional pressure formulation in molecular dynamics, *Journal of Molecular Graphics and Modelling* 79 (2018) 230–234.
- [42] E. L. Granados-Bazán, S. E. Quiñones-Cisneros, U. K. Deiters, Structure and contact angle in sessile droplets of binary mixtures of lennard-jones chains: A molecular dynamics study, *Langmuir* 37 (2021) 10945–10957.
- [43] K. Suzuki, I. Horiba, N. Sugie, Linear-time connected-component labeling based on sequential local operations, *Computer Vision and Image Understanding* 89 (2003) 1–23.
- [44] A. Azad, A. Buluç, Lacc: A linear-algebraic algorithm for finding connected components in distributed memory, in: 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS), IEEE, 2019, pp. 2–12.
- [45] J. C. Bennett, H. Abbasi, P.-T. Bremer, R. Grout, A. Gyulassy, T. Jin, S. Klasky, H. Kolla, M. Parashar, V. Pascucci, et al., Combining in-situ and in-transit processing to enable extreme-scale scientific analysis, in: SC’12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, IEEE, 2012, pp. 1–9.
- [46] Paraview website, 2023. <https://www.paraview.org/>.
- [47] Top500 website, november 2023. <https://www.top500.org/lists/top500/list/2023/11/>.
- [48] C. Valensi, W. Jalby, M. Tribalat, E. Oseret, S. Ibnamar, K. Camus, Using maqao to analyse and optimise an application, in: 2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), IEEE Computer Society, 2019, pp. 423–424.
- [49] R. Ravelo, T. Germann, O. Guerrero, Q. An, B. Holian, Shock-induced plasticity in tantalum single crystals: Interatomic potentials and large-scale molecular-dynamics simulations, *Physical Review B* 88 (2013) 134101.
- [50] E. N. Hahn, T. C. Germann, R. Ravelo, J. E. Hammerberg, M. A. Meyers, On the ultimate tensile strength of tantalum, *Acta Materialia* 126 (2017) 313–328.
- [51] Y. Wang, X. Zeng, X. Yang, T. Xu, Shock-induced spallation in single-crystalline tantalum at elevated temperatures through molecular dynamics modeling, *Computational Materials Science* 201 (2022) 110870.
- [52] J. R. Vella, M. Chen, F. H. Stillinger, E. A. Carter, P. G. Debenedetti, A. Z. Panagiotopoulos, Structural and dynamic properties of liquid tin from a new modified embedded-atom method force field, *Physical Review B* 95 (2017) 064202.

Micro-jetting: Areal density calculation from a triangular groove

Cite as: J. Appl. Phys. **135**, 205902 (2024); doi: [10.1063/5.0209692](https://doi.org/10.1063/5.0209692)

Submitted: 22 March 2024 · Accepted: 9 May 2024 ·

Published Online: 29 May 2024



View Online



Export Citation



CrossMark

Laurent Soulard,^{1,2,a)}  Olivier Durand,^{1,2}  Jean-René Burie,¹ and Killian Babilotte¹ 

AFFILIATIONS

¹CEA, DAM, DIF, F-91297 Arpajon, France

²Université Paris Saclay, CEA, Laboratoire Matière en Conditions Extrêmes, Bruyères-le-Châtel 91680, France

^{a)}Author to whom correspondence should be addressed: laurent.soulard@cea.fr

ABSTRACT

We present a method for calculating the mass ejected during the reflection of a shock wave on a triangular groove. This calculation is based on the combination of two models taken from the literature, BMPT-2, on the one hand, for the calculation of the velocity and density of the jet, and fragmentation zone propagation (FZP) on the other hand, for the calculation of the ejected mass, certain parameters required for FZP being determined by BMPT-2. Compared with previous work, FZP has been extended to deal with the various stages of jet formation. The approach was first evaluated on tin using a large-scale molecular dynamics simulation. This first step validated the overall phenomenology and the associated theoretical tools, and enabled us to propose a procedure for adjusting FZP. Next, we used the BMPT-2/FZP combination to analyze ejected mass measurement experiments using Asay foil. The areal mass curves are well reproduced with few parameters, showing that there is no inconsistency between BMPT-2/FZP and the experiments. Finally, a more detailed analysis of the results obtained enables us to set the limits of the jet thickness at the moment of rupture, and to propose a simple analytical form of its profile compatible with the model used.

© 2024 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/5.0209692>

I. INTRODUCTION

When a shock wave emerges on a surface (known as a “free surface”) in contact with vacuum or air, the projection of very fast particles is a constant problem in many applications, whether to avoid them, to protect against them (to protect a measuring device, for example), or to use them (cold-spray). This phenomenon has therefore been the subject of a great deal of work in the literature, whether through experiment, theory, or numerical simulation. The two main causes of material ejection under shock are micro-spallation and micro-jetting. Micro-spallation is the spallation process that follows the reflection on a free surface of an unsustained shock wave of sufficient intensity to melt the material, either directly by the shock or during the ensuing release wave. Although it can be coupled to the micro-jetting, this phenomenon will not be studied in this work. The second cause of material ejection is the micro-jetting and is due to the presence of non-planar areas on the free surface, such as grooves caused by the passage of a tool. The reflection of the impact on these grooves can, under certain very

specific conditions,¹⁻³ causes the formation of a (micro-)jet which will grow and then break up into particles which can be very fast.

It is often important to know the characteristics of these particles, in particular, their mass and velocity, and the relationship between these two quantities. In this article, we postulate that these quantities are closely linked to the jet rupture mechanism. The aim of this work is therefore to present a set of theoretical tools to better understand this mechanism. To do this, we will consider the particular case of a triangular groove, although the generalization to other groove shapes is possible. The approach we are proposing is based on two models that we have already published elsewhere. The first is the BMPT-2 model³ (modified model of Birkhoff, Duncan, Macdougall, Pugh, and Taylor), which estimates jet velocity and density. It is based on a detailed description of the interaction of the shock with the groove, enabling us to identify the different phases in the development and rupture of the jet. The second model is the fragmentation zone propagation (FZP) model, which was published around ten years ago.^{4,5} It proposes a formalism for calculating the ejected mass and makes it possible to link

23 August 2024 09:13:19

the velocity and size of particles, and is based on a break-up mechanism observed in classical molecular dynamics simulations. The originality of the present work lies in combining these two models, with BMPT-2 supplying a number of parameters to FZP. The approach will first be developed using a classical molecular dynamics simulation and then applied to experimental results.

Micro-jetting has already been extensively studied in the literature. The simulation of micro-jetting at experimental scales is the domain of macroscopic methods solving the equations of continuum mechanics, and various works have implemented these methods.^{6–15} The main difficulties, apart from the specifically numerical problems, are (i) taking account of surface tension and viscosity, both of which play a potentially important role in the case of micro-jetting, and (ii) the lack of knowledge of the jet fragmentation mechanisms. In general, the properties of the jet are correctly calculated in relation to the experiment (in particular its characteristic velocities). However, predicting the mass of material ejected requires, in principle, a specific model of jet fragmentation, otherwise the rupture can be considered as a simple numerical artefact whose result depends on the calculation conditions (the mesh, in particular).^{8,9,16} It is, therefore, difficult to predict from these simulations quantities such as the size and shape of the ejecta, as well as their velocity. Classical molecular dynamics (MD), widely used to describe the micro-jetting, is an alternative to continuum mechanics. The advantage of molecular dynamics is that it does not require models other than the potential function describing interatomic interactions. There are, however, a number of difficulties associated with (i) the potential function, which is not necessarily valid for the whole of the vast thermodynamic domain covered by micro-jetting (shock, release, melting...), and (ii) the computational capabilities, which drastically limit the size of the samples that can be simulated (about three orders of magnitude separate computation and experiment). As a result, significant scaling effects are to be expected compared with experiments, particularly if surface tension and viscosity play an important role. Furthermore, the small size of the samples means that the micro-structure of the experimental samples cannot really be taken into account, which can have a significant effect on the final result. However, visualization of the jet obtained by ultrafast optical shadowgraphy^{17–23} or X shadowgraphy,^{24–30} shows that the jet adopts a filamentary structure, which seems to depend relatively little on the shape of the groove and the nature of the material. This result is important because it validates, at least qualitatively, the relevance of classical molecular dynamics (MD) for the study of micro-jetting, since very similar jet structures have been observed in this type of simulation.^{4–6,16,31–44}

From a theoretical point of view, the properties of the jet are calculated either by treating this phenomenon as a limiting case of Richtmyer–Meshkov instabilities (RMI)^{45–49} or as a special case of Birkhoff *et al.* theory (BMPT) of shaped charges.^{50,51} Each of these approaches has its advantages and disadvantages, and its own field of application: rather sinusoidal grooves for RMI models,^{52–59} rather triangular grooves for theories derived from BMPT.^{3,17} From the prediction of the evolution of the shape of the jet, most models for calculating the areal mass deduce the latter either from the missing mass,⁶⁰ or from characteristic velocities and the areal mass of the jet,^{56–58,61} but without specifically taking the rupture into

account. In other words, it is assumed that the entire jet fragments, which is not so clear from the MD simulations.³⁹ In addition, these models do not allow the relationship between ejecta velocity and mass to be calculated. The originality of the approach we propose in this article is that the calculation of the ejected mass is based on an explicit description of the fragmentation of the jet. We will see that the fragmentation mechanism chosen, based on experimental observations and the analysis of numerical simulations, is strongly linked to the shape of the jet, and, therefore, to the mass of potentially ejectable materials.

This article is organized as follows. The first part (Sec. II) briefly recalls the theoretical aspects of jet formation (Sec. II A) and the calculation of areal density σ , i.e., the FZP model (Sec. II B), which are based on a break-up mechanism already studied in previous papers. The combination of models BMPT-2 and FZP is evaluated in Sec. III, first on the basis of a molecular dynamics simulation (Sec. III A) and then on the basis of experimental results (Sec. III B). The consequences of this model are discussed in Sec. IV.

II. THEORY

Throughout this article, velocities will be defined in two coordinate systems. The laboratory coordinate system, in which velocities will be denoted u , and a moving jet coordinate system, in which velocities will be denoted v . Positions will always be defined in the moving coordinate system. The moving coordinate system will be defined as appropriate in each case.

A. The steps in the formation of a jet

The sample is a plate of infinite dimension in the y direction. A network of isosceles triangular grooves with axes of symmetry SS' perpendicular to the yz plane were made on one of the two surfaces of the plate [Fig. 1(a)]. Let us consider a particular groove. We define two other axes TT' parallel to SS' passing through the two tops of the groove. A sustained incident shock S_i with a velocity u_s parallel to SS' propagates through the sample. The formation of a jet resulting from the reflection of S_i on the surface of the groove can be seen as a two-step process. The first step is the passage of the shock wave from the bottom to the top of the groove [Fig. 1(b)]. To a first approximation, this step is independent of the environment of the groove, in particular, the presence of other grooves. The second step corresponds to all the processes that take place after the first step. It is highly dependent on the groove environment.

In a first step, the free surface of the groove is deflected from its initial orientation and set in motion at velocity u_B toward the axis of symmetry SS' . At the same time, two oblique release waves \mathcal{D}_1 and \mathcal{D}'_1 propagate through the sample [Fig. 1(b)]. When the two opposing surfaces meet, the result is either a reflected shock in the material or a jet in front of the free surface, depending on the intensity of the shock and the shape of the groove.^{1–3} In this document, we only consider the jet case, but it is important to bear in mind that the other solution is sometimes possible. Step 1 is complete when the shock reaches the top of the groove. The duration of step 1 (t_{s1}) is simply the transit time of the shock from the bottom of the groove to its top. Of course, this time depends on the

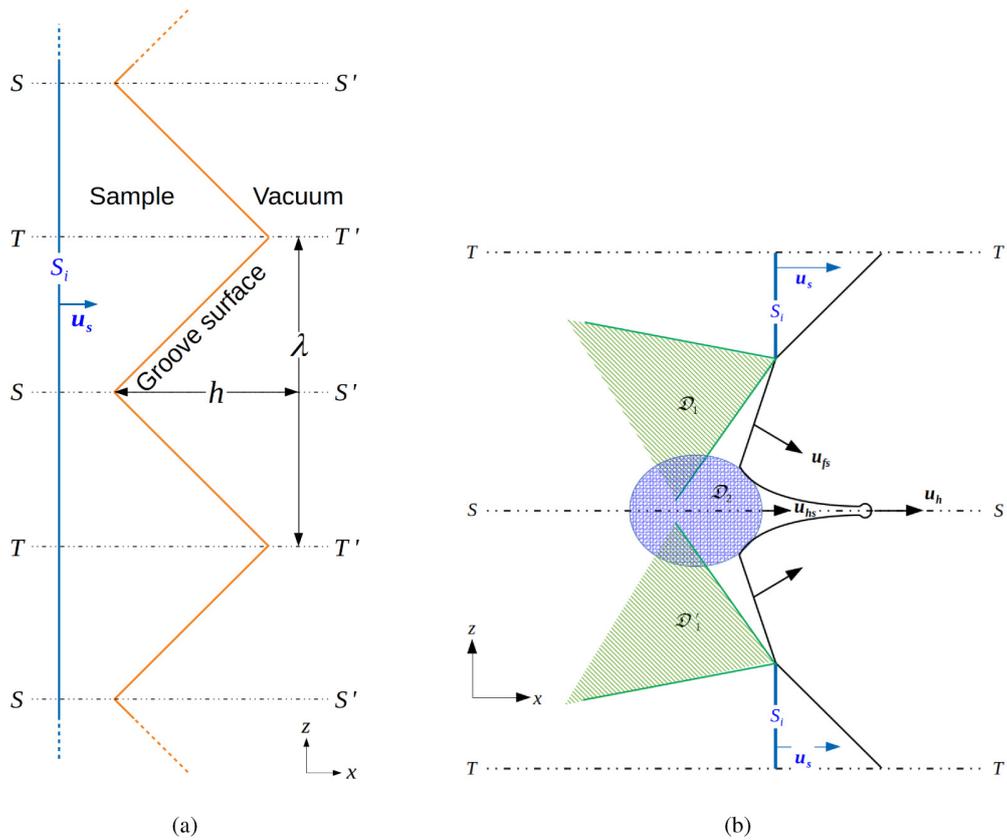


FIG. 1. Formation and development of a jet in the configuration studied in this article. (a) A shock S_i with a velocity u_s (solid blue line) propagates along the the axis of symmetry SS' of the groove. The two axes TT' are parallel to SS' and pass through the two tops of the groove. (b) The reflection of S_i on the surface of the groove sets it in motion at velocity u_{fs} and causes two release waves (\mathcal{D}_1 and \mathcal{D}'_1) to propagate in the sample (green hatched zones). The formation of the jet, as proposed in the BMPT-2 model,³ causes a third release wave \mathcal{D}_2 (blue hatched zone) to propagate and deform the jet.

intensity of the impact and the depth of the groove, but for the usual micro-jetting experiments, carried out using a laser shock or a gun (such as those presented in Sec. III B), the grooves are generally less than a millimeter deep, which corresponds to transit times of less than a microsecond for shocks traveling at a few km/s. This duration is small compared with the overall duration of an experiment.

The flows of step 1 were analyzed in detail in a previous article³ and only a brief reminder is proposed in this paragraph. Following the usual formalism of shaped charge theory,^{50,51} we assume that the flows are stationary in the coordinate system moving with the collapse point F of the groove. In this coordinate system, the flow is parallel to the free surface before F and splits into two parts parallel to SS' after F , due to the conservation of momentum: a first part toward the front (the jet) and a second part toward the rear (the “slug,” following the shaped charge terminology). The essential difference with a shaped charge is that the slug interacts with a dense material (the shocked material) and not with the gas resulting from the detonation of the explosive. This

interaction stops the propagation of the slug, pushes it forward, and modifies the velocity of the jet as proposed by the BMPT-2 model.³ BMPT-2 defines two characteristic velocities. The first is the velocity of the jet u_h , and the second (u_{hs}) corresponds to the intersection of the Hugoniot of the slug and the isentrope passing through the shocked state (hs point). This point defines the foot of the jet during step 1. Since $u_h > u_{hs}$, the velocity decreases between the head and foot of the jet. In addition, hs is the point of origin of a rarefaction wave \mathcal{D}_2 due to the release of the shock material into the jet.

The second step is more complex to model than the first because it depends on the groove environment: path groove, isolated groove, etc. Let us consider the case of a network of grooves. The fans of the release waves resulting from the reflection of the shock on the free surface of each groove begin to intersect when the shock reaches the top of the grooves (zone 1, Fig. 2). This intersection of the release waves has two consequences. First, the release waves are transformed into compression waves propagating backwards after reflection from the surface of a neighboring groove,

23 August 2024 09:13:19

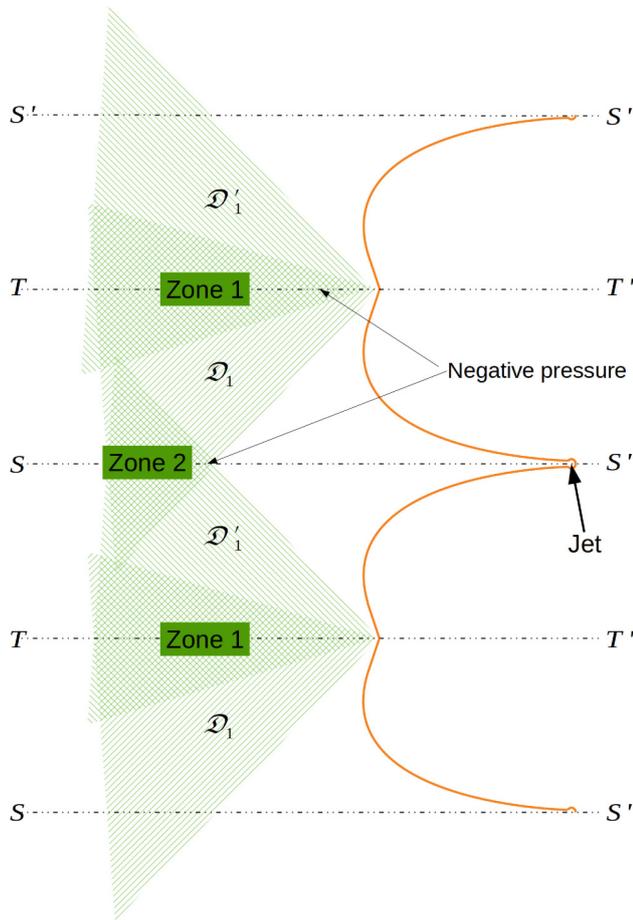


FIG. 2. Flow when the shock reaches the top of the grooves in the case of a periodic network of identical grooves. Zone 1 is a region of negative pressure, located in the immediate vicinity of the surface, and resulting from the intersection of the release waves D_1 and D_1' from two adjacent grooves. Zone 2 is a second region of negative pressure, located far from the surface, and resulting from the crossing of the release waves D_1 and D_1' from the same groove.

causing the surface to slow down. This slowdown propagates along the surface of the groove and reaches the jet, causing it to change shape. Second, as the pressure along the last wave of the release fan is zero, the pressure immediately becomes negative in the crossing zone. Depending on the intensity of the shock, the minimum pressure in the crossing zone may be lower than the spall pressure, causing damage to the material in the region around the TT' axes. As the two release waves intersect very close to their origin, the material deformation rate is very high in the area of the intersection, causing possible disintegration of the material in the immediate vicinity of the surface and therefore the ejection of small particles in this area (TT' fragmentation). In the case of an isolated groove, this process does not occur. It should be noted that a second zone of negative pressure can occur around the SS' axes,

but located quite far from the ejection zone and disrupting it little or not at all the first moments of this process. However, for longer periods of time, this zone of negative pressure can also cause damage to the material, or even breakage, thus interrupting the supply of material to the jet. As the shock is sustained, the velocity of the surface located between the jets tends toward an average velocity u_{pfs} close to the velocity of the flat free surface. This means that the velocity profile changes gradually: the head of the jet maintains the same u_h velocity, but the velocity of its foot (u_f) gradually changes from u_{hs} to $\approx u_{pfs}$.

B. Calculation of the areal density

We are going to assume that the fragmentation of the jet is not instantaneous, but that it propagates from the head of the jet toward its foot, as suggested in Refs. 4 and 5. In these articles, we observed, from MD simulations, that jet fragmentation occurs in several stages. First, holes appear in the jet. They are randomly distributed along the y direction but are created at approximately the same x position. As the jet stretches, these holes grow and lead to the formation of a network of filaments of materials, mainly along the x axis. These filaments then break away from the bulk material. The smallest filaments quickly take on a spherical shape under the effect of surface tension. The larger filaments, subjected to strong deformation along x , break up into a number of fragments whose shape rapidly tends toward the sphere. We are now going to greatly simplify this process by assuming that the break-up of the jet occurs at time t at point x . In other words, we neglect the details of the break-up process.

We now place ourselves in a moving coordinate system linked to the foot of the jet, whose position we assume to be known. Let $x_{fr}(t)$ be the position of the fragmentation front at time t and $v_{fr}(t)$ its velocity. At time t , the breakup occurs at point $x_{fr}(t)$ and a particle p moves away from the jet. Its velocity is constant because we assume that the particle p is in ballistic flight (interactions between ejected particles are neglected, as the particle's internal vibrations). We assume that the velocity v of this particle p is the velocity $v(x = x_{fr}(t))$ of the corresponding material element of the jet just before fragmentation. v_{fr} can be positive or negative, provided that $v_{fr} < v$ (this approximation has already been proposed by Harrison⁶²). If we represent the fragmentation front as a plane perpendicular to x , the quantity of material m_p passing through this plane between times t and $t + dt$ is

$$m_p(t) = \rho_j(x, t)e_p(x, t)y_j(v - v_{fr})dt, \quad (1)$$

where $\rho_j(x, t)$ is the local jet density at time t , $e_p(x, t)$ is the local thickness of the jet at time t and y_j is the depth of the jet along the y axis. We assume that (i) the thickness of the jet at the breaking point is always equal to a critical thickness e_{cr} and (ii) the jet density is constant. To determine some properties of v_{fr} , it will be necessary to return to the mechanisms of jet formation. Since e_{cr} is constant, the position of the critical thickness depends only on the evolution of the jet shape over time. As the thickness of the jet increases continuously between the head and the foot of the jet, the fragmentation must necessarily initiate toward the head of the jet, and then propagate toward its foot. Several steps can be defined, as

23 August 2024 09:13:19

described in Sec. II A. The FZP model assumes that each step leads to a particular evolution of the jet shape, corresponding to specific zones Z_n , the Z_0 zone being the region closest to the head of the jet. Note that the zones may be identified by discontinuities in the slope of the profile describing the jet surface in the plane (x, z) . The second assumption of FZP is that v_{fr} is specific and constant for each zone Z_n . This assumption leads of course to a strong constraint on the shape of the jet. We will come back to this later. Let $v_{fr}^{(n)}$ be the velocity v_{fr} of the Z_n zone in which the fragmentation occurs at time t . We assume a relationship such as $v(x, t) = x/t$ (this hypothesis was also adopted by Harrison⁶²). This choice is the result of previous observations based on MD simulations⁴ and will be confirmed in Sec. III A. Taking these assumptions into account, Eq. (1) becomes

$$m_p(t) = \rho_j e_{cr} \gamma_j \left(x_{fr}(t^{(n)}) - v_{fr}^{(n)} t^{(n)} \right) \frac{dt}{t}, \quad (2)$$

with $t^{(n)}$ being the time at which the fragmentation front enters the zone Z_n , and we assume that $t < t^{(n+1)}$. The contribution $m_c^{(n)}$ of the zone Z_n to the cumulative mass $m_c(t)$ is obtained by integrating Eq. (2) between $t^{(n)}$ and t ,

$$m_c^{(n)}(t) = \rho_j e_{cr} \gamma_j \left(x_{fr}(t^{(n)}) - v_{fr}^{(n)} t^{(n)} \right) \log \frac{t}{t^{(n)}}. \quad (3)$$

To obtain an expression as a function of ejecta velocity, we use the relation

$$t = \frac{x_{fr}(t^{(n)}) - v_{fr}^{(n)} t^{(n)}}{v - v_{fr}^{(n)}}, \quad (4)$$

and inject this result into Eq. (3),

$$m_c^{(n)}(v) = \rho_j e_{cr} \gamma_j t^{(n)} \left(v^{(n)} - v_{fr}^{(n)} \right) \log \frac{v^{(n)} - v_{fr}^{(n)}}{v - v_{fr}^{(n)}}, \quad (5)$$

with $v^{(n)}$ the velocity of matter at the boundary between zones n and $n - 1$. Note that $\{v^{(n)}\}$ and $\{v_{fr}^{(n)}\}$ are not independent and are linked by the relationship

$$v_{fr}^{(n)} = \frac{v^{(n)} \dot{\epsilon}(t^{(n+1)}) - v^{(n+1)} \dot{\epsilon}(t^{(n)})}{\dot{\epsilon}(t^{(n+1)}) - \dot{\epsilon}(t^{(n)})}, \quad (6)$$

with $\dot{\epsilon}(t^{(n)}) = 1/t^{(n)}$. We will see later that in some cases, $|v_{fr}^{(n)}| \rightarrow \infty$. Equation (5) can then be written as

$$m_c^{(n)}(v) \approx \rho_j e_{cr} \gamma_j t^{(n)} (v^{(n)} - v). \quad (7)$$

Since $|v_{fr}^{(n)}|$ is infinite, $t^{(n)} = t^{(n+1)}$ and $m_c^{(n)}$ is simply the mass included in the parallelepiped $e_{cr} \gamma_j (x_{fr}^{(n)} - x)$. This situation corresponds to a zone in which the thickness of the jet is the same everywhere. The fragmentation therefore occurs

simultaneously at every point in the zone, hence the infinite value of $v_{fr}^{(n)}$.

To fit Eq. (5) [or Eq. (3)] to an experimental areal density (which is of course proportional to the cumulative mass), we need to choose the number of zones N_z (which will be numbered from 0 to $N_z - 1$). The point $u^{(N_z)}$ is the velocity at the end of the last zone, and the point $v^{(0)}$ is chosen as the head of the jet. Under these conditions, Eq. (5) depends on the density of the jet and its velocity, since $v^{(0)} = v_h$. These two parameters can be calculated using the BMPT-2 model we recently proposed,³ assuming that the equation of state (EOS) of the material is known. The other parameters are $\{v^{(n)}\}$ and $\{t^{(n)}\}$, allowing the calculation of $\{v_{fr}^{(n)}\}$ [Eq. (6)]. Examples of theoretical $m_c(v)$ profiles of are shown in Fig. 3.

The assumption of a constant velocity of the fracture front per zone severely limits the possible shape of the jet, as we will see. We saw that $v(x, t) = x/t$. We assume that the jet is not compressible and we neglect the shear in the jet. Consider a velocity v . So at every point on the jet with coordinate x ,

$$v = \frac{x(v, t)}{t} = \frac{x(v, t_{ref})}{t_{ref}}, \quad (8)$$

where $t_{ref} > 0$ is any arbitrary time. Let $\tau(t) = \frac{t}{t_{ref}}$. Therefore,

$$x(v, t) = \tau(t) x(v, t_{ref}), \quad z(v, t) = \frac{1}{\tau(t)} z(v, t_{ref}). \quad (9)$$

Let x_Σ and z_Σ be the coordinates of the outer surface of the jet in a Z_n zone at a given time t_{ref} . In general, x_Σ and z_Σ can be related by the relation

$$x_\Sigma(t_{ref}) = \mathcal{F}_n(z_\Sigma(t_{ref})), \quad (10)$$

with \mathcal{F}_n being a function to be determined. Since Eq. (9) are valid whatever the value of z corresponding to a given x ,

$$x_\Sigma(t) = \tau(t) \mathcal{F}_n(\tau(t) z_\Sigma(t)). \quad (11)$$

Let t_{fr} be the time when a jet element with velocity v reaches the critical thickness e_{cr} . At this moment, we have $z_\Sigma(t_{fr}) = \frac{1}{2} e_{cr}$, the jet being centered on $z = 0$, and $x_\Sigma(t_{fr}) = x_{fr}(t_{fr})$,

$$x_{fr}(t_{fr}) = \tau(t_{fr}) \mathcal{F}_n\left(\frac{1}{2} e_{cr} \tau(t_{fr})\right). \quad (12)$$

Let us assume that the jet element with the coordinate $x_{fr}(t_{fr})$ is located in the Z_n zone. Since $\frac{dx_{fr}(t_{fr})}{dt_{fr}} = v_{fr}^{(n)}$, the function \mathcal{F}_n is a solution of the differential equation

$$v_{fr}^{(n)} t_{ref} = \mathcal{F}_n(A) + A \frac{d\mathcal{F}_n(A)}{dA}, \quad (13)$$

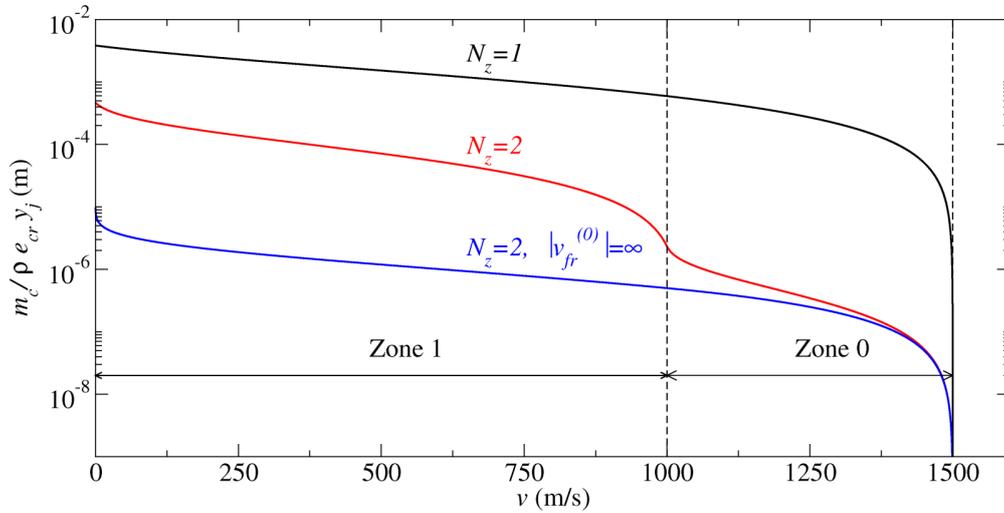


FIG. 3. Examples of cumulative mass calculated with FZP. The black curve corresponds to the case where there is only one zone ($v_{fr}^{(0)} = -166$ m/s), the red curve to the case of two zones with finite fragmentation front velocities ($v_{fr}^{(0)} = 994$ m/s, $t^{(1)} = 100$ ns, $v_{fr}^{(1)} = 1000$ m/s and $v_{fr}^{(1)} = -10$ m/s), and the blue curve to the case of two zones with an infinite fragmentation front velocity for the first zone ($t^{(0)} = t^{(1)}$ and so $|v_{fr}^{(0)}| = \infty$, $v_{fr}^{(1)} = 1000$ m/s).

with $A(t_{fr}) = \frac{1}{2} e_{cr} \tau(t_{fr})$. Since $v_{fr}^{(n)}$ is a constant, the solution of Eq. (13) is

$$\mathcal{F}_n \left(\frac{1}{2} e_{cr} \tau(t_{fr}) \right) = v_{fr}^{(n)} t_{ref} + \frac{K_n}{\frac{1}{2} e_{cr} \tau(t_{fr})}, \quad (14)$$

and so

$$x_{fr}(t_{fr}) = v_{fr}^{(n)} t_{fr} + \frac{K_n}{\frac{1}{2} e_{cr}} \quad (15)$$

with

$$K_n = \frac{1}{2} e_{cr} \left(x_{fr}(t^{(n)}) - v_{fr}^{(n)} t^{(n)} \right). \quad (16)$$

The relationship (15) is true at the breaking point but it will be assumed that it can be generalized to the entire jet. To do this, we replace $x_{fr}(t_{fr})$ by $x_{\Sigma}(t)$ and $\frac{1}{2} e_{cr}$ by $z_{\Sigma}(t)$ in Eq. (15). This rather strong approximation will be checked in Sec. III A. With this assumption, Eq. (15) can be generalized as follows:

$$z_{\Sigma}(t) = \frac{1}{2} e_{cr} \frac{x_{fr}(t^{(n)}) - v_{fr}^{(n)} t^{(n)}}{x_{\Sigma}(t) - v_{fr}^{(n)} t}. \quad (17)$$

At a fixed instant, z_{Σ} is therefore a simple homographic function of x_{Σ} . FZP, which is based on two assumptions regarding the rupture of the jet (existence of a critical rupture thickness and propagation of the rupture at a locally constant velocity) imposes the shape of the jet, and therefore the mass of material that can

potentially be ejected. We will now evaluate these two hypotheses using a molecular dynamics simulation and by experiments dedicated to this study.

III. APPLICATIONS

A. MD simulation

The aim of the MD simulation presented in this section is to verify the various hypotheses formulated in Sec. II and to propose a procedure for fitting the cumulative mass (or areal density) to an experimental curve (in this section, from the simulation). MD is an interesting tool because it allows both hydrodynamic and fragmentation aspects to be taken into account without any model other than the potential energy function. While many studies have used MD simulation to describe and understand micro-jetting processes,^{4,5,31–34,39,43,44,55,63–66} only a few have been used to test theoretical models. Concerning FZP, a preliminary version has already been tested in the past^{4,16} with a very simplified model (Embedded-Atom Method) for tin,⁶⁷ describing rather poorly the properties of the real material. The BMPT-2 model was also evaluated using simulations on simplified tin, as well as experimental data.³ A preliminary study using a more physical potential (the Modified Embedded-Atom Method) (MEAM) based on the work of Ravelo and Baskes⁶⁸ enabled us to analyze the role of phase transitions in tin.³⁹ Here, we propose a new analysis of jet fragmentation using the same MEAM potential. Even though it is imperfect, MEAM is a great improvement on EAM and, although it is a little more costly from a computational point of view, it still allows simulations to be carried out with enough atoms to avoid scale effects as far as possible. As this potential reproduces quite correctly the properties of tin under shock,⁶⁹ as well as quantities such

as viscosity and surface tension,⁷⁰ we will attempt a qualitative comparison with some experimental data, but it should be emphasized that neither this potential, nor the simulated dimensions of the sample, nor the chosen microstructure, allow quantitative information to be extracted from the simulation for a precise comparison with experiment. In short, we will limit ourselves to verifying the relevance of our hypotheses, with a model leading to physics that is sufficiently reasonable to be representative of reality. We will concentrate on a single, large-scale simulation (for the MD) in order to minimize artifacts due to the size of the sample and statistical uncertainties on thermodynamic quantities or on the measurement of quantities such as the velocity of the fragmentation front.

1. Simulation set-up

The calculation conditions (groove geometry and intensity of the shock) were chosen so as to be in the liquid state after the passage of the shock, to get as close as possible to the theoretical hypotheses in the previous section. We have chosen a fairly simple configuration: a triangular groove with a half-angle $\beta = 45$ degrees. It should be noted that experimental^{17,30} and smoothed particle hydrodynamics (SPH) simulations⁷¹ results corresponding to this geometric configuration (or very similar configurations) are available in the literature. The sample is a parallelepipedic single crystal in β phase, whose length L_x is parallel to the crystallographic axis (100), the depth L_y is parallel to (010) and the width L_z is parallel to (001). We have chosen $L_x = 2 \mu\text{m}$ and $L_y = L_z = 0.162 \mu\text{m}$ (i.e., $\approx 2 \times 10^9$ atoms). The boundary conditions are free along x , and periodic along y and z . The surface at the left end ($x = 0$) is in contact with a piston modeled by a purely repulsive potential. We therefore simulate a network of grooves similar to that shown Fig. 1(a). All the grooves being exactly the same, this configuration is rather artificial and could lead to artifacts, but we'll admit that, if they exist, they do not alter our results.

The system is first thermalized at 300 K for 100 ps using a Langevin thermostat. It is then left to rest for a further 100 ps in the microcanonical ensemble so as to "forget" any artefacts induced by the thermostat. The piston is then set in motion instantaneously at a velocity $u_\pi = 2000$ m/s. The simulation was done using the ExaStamp code developed at CEA.⁷²⁻⁷⁴

2. General results

A sustained shock then develops and propagates toward the groove. The characteristics of this shock have already been studied:⁶⁹ velocity $u_s = 5881$ m/s, pressure jump $p_{S_1} = 84.4$ GPa, density behind the shock $\rho_{S_1} = 10912$ kg/m³.

This shock is strong enough to melt the tin directly behind the shock front (the melting pressure under shock is around 50 GPa). The corresponding experimental values are $u_s = 5661$ m/s, $p_{S_1} = 82.6$ GPa et $\rho_{S_1} = 11277$ kg/m³⁷⁵ and therefore relatively close to the values given by the MEAM potential. This discrepancy of 3.9% on the shock velocity is relatively small and reflects the limits of the potential, the use of a perfect single crystal rather than polycrystalline samples in the experiments, and of course experimental uncertainties. We will therefore use the experimental relationship $u_s = 2590 + 1.49u_p$ m/s⁷⁵ to calculate the expected quantities with BMPT-2. The shock reaches the bottom of the groove at

$t = t_{bg} = 0.526$ ns, and the top of the defect at $t = 0.540$ ns (and, therefore, $t_{s_1} = 14$ ps). A jet is formed which will grow and then break.

To characterize the jet, we plotted the profiles of various quantities (density, velocity, etc.) along the x axis over a domain of $\pm 4 \text{ \AA}$ in z , so as not to include in the calculation any empty spaces that would distort the quantities calculated. Quantities are evaluated in slabs of 3 \AA length along x , and depth of L_y along y . The position of the jet head is defined as that of the first slice from the right whose density is greater than 1000 kg/m^3 . The result shows that it stabilizes very quickly around an average velocity of 6800 m/s (Fig. 4). There is a slight slowdown at the head of the jet, which can be attributed to viscosity, surface forces or, more generally, to the general evolution of hydrodynamics during the development of the jet. A comparison of the results obtained from the MD with the predictions of the BMPT-2 (Table I) shows this model only partially reproduces the numerical results. The jet velocity is fairly well calculated (6637 m/s), i.e., an underestimate of the velocity measured in MD of only -2.4% . The value obtained with the BMPT model⁵⁰ (5858 m/s) underestimates the jet velocity (-13%), while the model proposed by Karkhanis *et al.*¹¹ gives a value higher than that of the MD (7746 m/s, $+13\%$).

The material velocity and density profiles at short times [Fig. 5(a)] clearly show the two steps described in Sec. II A. Initially, the velocity profiles have a common value at $x = 1.99 \mu\text{m}$ (≈ 3565 m/s) which we identify as the velocity u_{hs} . The corresponding density is $\rho_{hs} = 7727$ kg/m³. Starting from the right, we first observe a quasi-linear decrease in velocity between the velocity of the head of the jet and a velocity of about ≈ 4200 m/s. We identify this velocity as the velocity of the foot of the jet ($u_f = 4200$ m/s). u_f is close to the velocity that a flat surface perpendicular to the direction of propagation of the shock would have after the shock has been reflected (it should be slightly higher than $2u_\pi$). The velocity is then stable and forms a plateau whose length increases with time. Finally, the velocity decreases again, passing through the point u_{hs} for a more or less constant value of $x \approx 1.99 \mu\text{m}$, before finally reaching the velocity u_π of the piston (2000 m/s). The velocity profiles between u_π and u_{hs} are characteristic of a release wave (they are self-similar). They correspond to the propagation of the release in the shocked material from point hs . The velocity profiles then evolve toward a stable structure to long times (on the time scale of the simulation) in three parts [Fig. 5(b)]. The point hs disappears because the first stage described in Sec. II A has ended.

We define the ejected mass as the total mass of material detached from the bulk material. To calculate the ejected mass, we used the Connected Components Analysis in a three-dimensional MPI/OpenMP version. The method has been implemented in the ExaStamp code and automatically detects voids or clusters and calculates their dynamic, structural, and thermodynamic properties. The duration of an analysis is of the order of the duration of an MD iteration and therefore does not disrupt the simulation if the frequency of analyses is not too high.

As described in Sec. II A, there are two sources of ejection which are, chronologically, first the crossing of the release waves \mathfrak{D}_1 and \mathfrak{D}'_1 at the top of the grooves (Fig. 2), then the fragmentation of the jet. These two processes are clearly visible in Figs. 6(a) and 6(b).

23 August 2024, 09:13:19

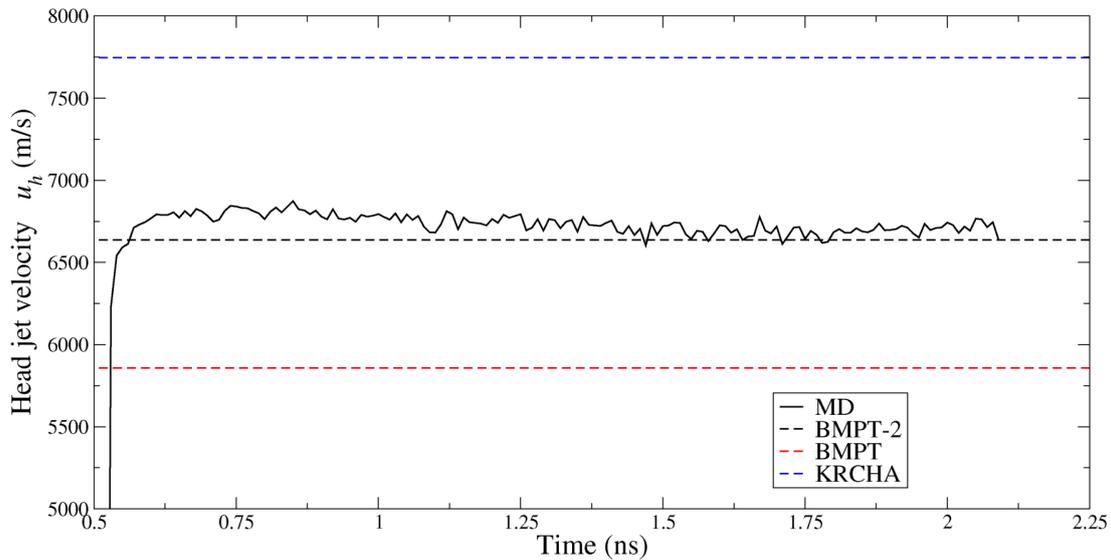


FIG. 4. MD simulation: time evolution of jet velocity. Dashed horizontal lines correspond to the BMPT-2³ model (black line), the BMPT⁵⁰ model (red line), and the Kharkanis¹¹ model (blue line).

These two processes are well separated in time and are easy to distinguish. In Fig. 7(a), we have plotted the accumulated ejected mass m_c calculated along the x axis from the head of the jet. The two processes described above are clearly apparent. For times between 0.5 and 1.5 ns, the mass remains low at around 10^{-20} kg and located close to the initial position of the groove surface. It then increases sharply from 1.5 ns to reach 2.8×10^{-18} kg at the end of the simulation. The location of the fragmentation is then close to the head of the jet and corresponds to the start of the jet break, as shown in the Fig. 6(b). The cumulative mass plotted against the ratio v/u_h [Fig. 7(b)] shows that, to a very good approximation, all the curves overlap, indicating the ballistic nature of the ejecta's trajectory. Note that this result is consistent with the experimental observations of Hammerberg *et al.*⁷⁶

Figure 7(c) shows the temporal evolution of the final cumulative mass [i.e., the maximum values of the curves in Figs. 7(a) or 7(b)]. The curve stabilizes at around 2 ns, followed by a jump corresponding to the appearance of large ejecta due to the jet's great thickness and the relatively low strain rate at this point.

Figure 8 shows the distribution of ejecta volume V using a binning procedure to the power of 2. The volume corresponding to

the head bulge has been eliminated. For volumes greater than 10^4 \AA^3 , the curves of volume density $N(V)$ (i.e., the number of ejecta by unit volume) stabilize and clearly comprise two parts: for $10^4 \leq V \leq 10^6 \text{ \AA}^3$, the distribution can be approximated by a power law $N(V) \propto V^{-1.37}$, and by an exponential law $N(V) \propto \exp(-\frac{V}{8.7 \times 10^6})$ (V in \AA^3) for $10^6 \leq V \leq 5 \times 10^7 \text{ \AA}^3$. We find a result already obtained with other potentials and other materials, which tends to show the fairly general nature of this type of distribution, which could correspond to two modes of jet fragmentation, as described in detail in Ref. 32. The power law relates to ejecta resulting directly from random fragmentation of the jet, whereas the exponential law relates to the secondary fragmentation of large ejecta, more or less cylindrical in shape, according to a Poissonian process. The largest volumes ($> 5 \times 10^7 \text{ \AA}^3$) again seem to be close to a power law. In fact, these ejecta are formed toward the foot of the jet, a region in which the rate of deformation, and therefore the energy available for fragmentation, is too low to initiate secondary fragmentation. However, this trend is only indicative, as the number of particles is small and it is possible that the size of the simulated sample influences the result.

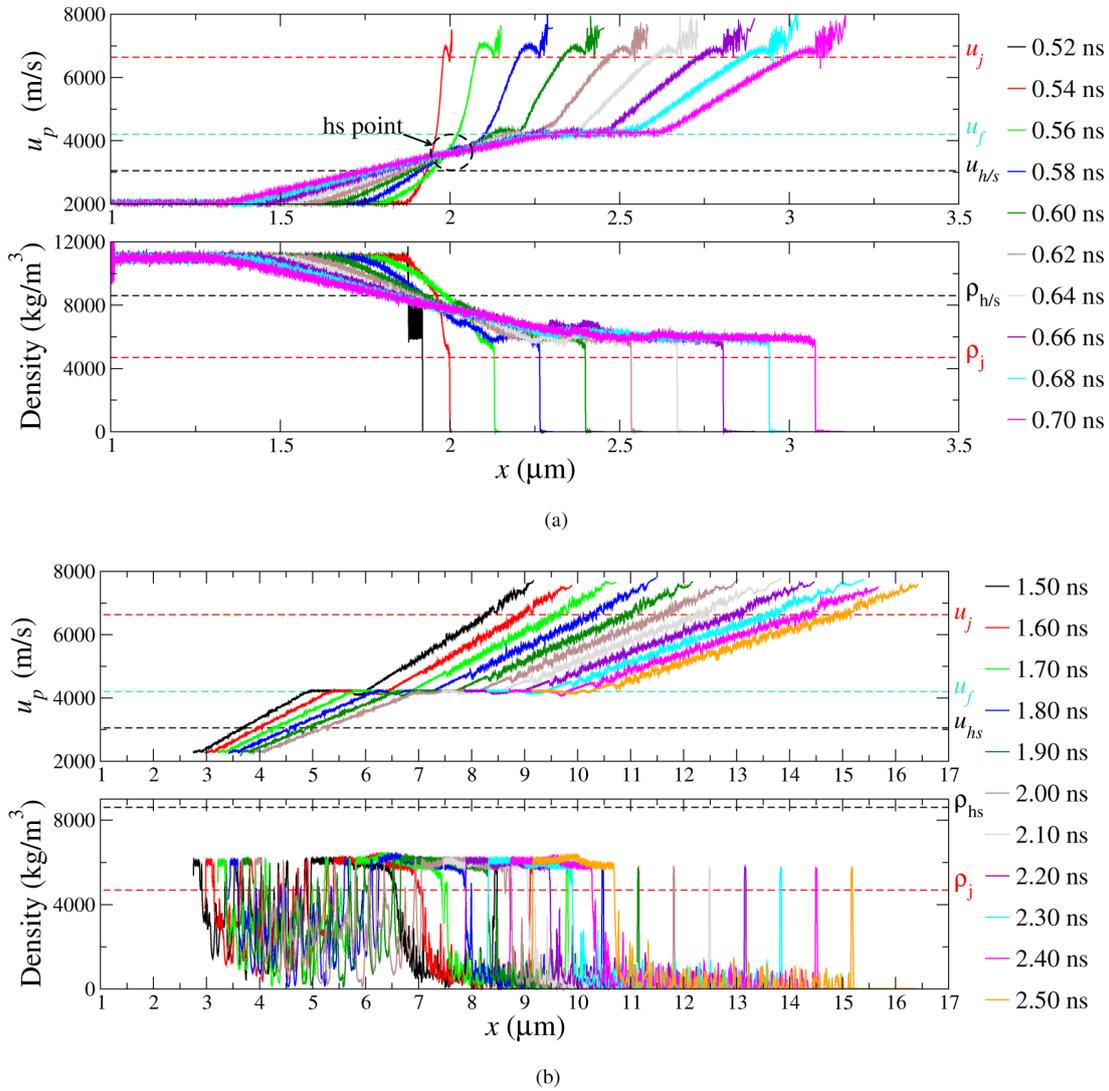
3. Comparison with FZP and BMPT-2.

We now want to compare the FZP predictions with the MD results, and to do this we will take a close look at the structure of the jet.

The head of the jet is a bulge resulting from surface tension. Obviously, FZP cannot be used to describe its break-up. However, it corresponds to a quantity of ejected mass $m_{c,b}$ that will have to be taken into account later. In the area adjacent to the head of the jet, holes appear from $t_{jh} \approx 1.13$ ns. The position of the holes does

TABLE I. Comparison of the velocity u_h and density ρ_j of the jet, and the velocity u_{hs} and density ρ_{hs} at the hs point, extracted from the molecular dynamics (MD) simulation and predicted with the BMPT-2 model.

u_h (m/s)		ρ_j (kg/m ³)		u_{hs} (m/s)		ρ_{hs} (kg/m ³)	
MD	BMPT-2	MD	BMPT-2	MD	BMPT-2	MD	BMPT-2
6800	6637	≈6000	4694	3565	3051	7727	8607



23 August 2024 09:13:19

FIG. 5. MD simulation: material velocity and density profiles. u_{hs} and ρ_{hs} are the theoretical velocity and density at the hs point, u_h and ρ_j are the theoretical velocity and density of the head of the jet, u_f the velocity of foot of the jet. (a) Short times. (b) Long times.

not correspond to a unique value of the x coordinate. They appear randomly in an area between $x \approx 5.4 \mu\text{m}$ and the head of the jet, corresponding to a velocity range of $5940 < v < u_h$ m/s. This region defines the Z_0 zone and so $v^{(0)} = u_h - u_f$ and $t^{(0)} = t_{fh} - t_{bg}$. In Z_0 , everything happens as if $|v_{fr}^{(m)}|$ adopts a very large value. The cumulative mass $m_{c,0}$ of this zone can therefore be calculated using Eq. (7),

$$m_{c,0} \approx \rho_j e_{cr} y_j t^{(0)} (v^{(0)} - v). \quad (18)$$

The next step is to measure the velocity of the fragmentation front in the rest of the jet (i.e., zone Z_1). To do this, we define a

first fragmentation front position as the point where the cumulative mass reaches its maximum value $m_{c,max}$. Figure 9 shows that the corresponding points (black circles) are clearly aligned and lead to a velocity $u_{m_{c,max}} = 4199$ m/s. We note that $u_{m_{c,max}}$ is equal (taking into account the uncertainty of the measurements) to the velocity of foot of the jet u_f (4200 m/s). However, the fragmentation front position thus defined includes TT' fragmentation. To eliminate it, we will assume that this process has been completed and contributes a constant value $m_{c,TT'}$ to the final value of the cumulative mass. The position of the real fragmentation front is then the point on the profile corresponding to $m_{max,c}^* = m_{c,max} - m_{c,TT'}$, and Fig. 7(a) shows that $m_{c,TT'} \approx 0.15 \times 10^{-18}$ kg. The corresponding

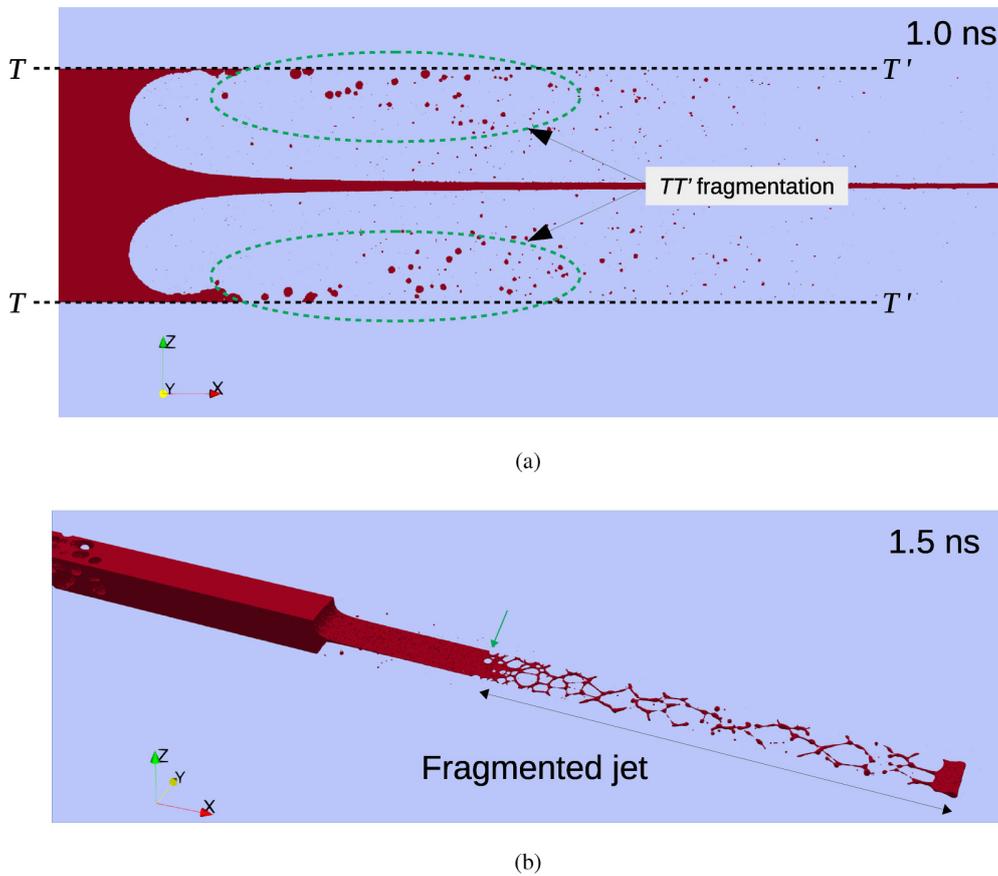


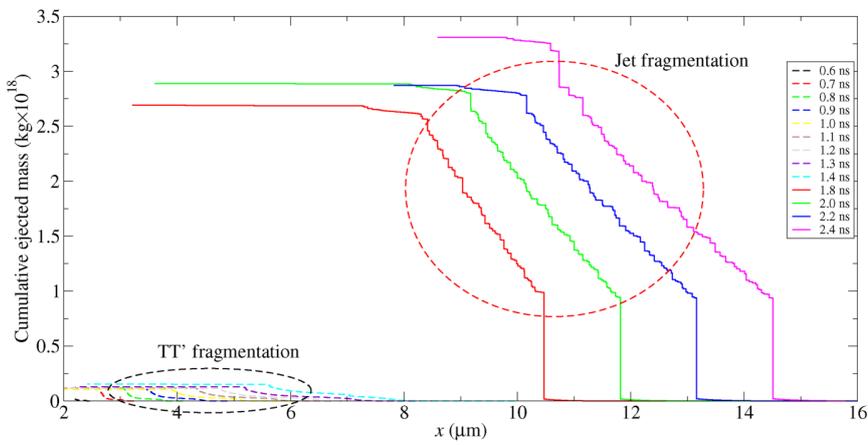
FIG. 6. (a) TT' fragmentation at $t = 1$ ns. (b) View of the jet at the very beginning of the rupture at $t = 1.5$ ns.

points lead to an average velocity $u_{m_{max,c}}^* = 3733$ m/s (red diamonds of Fig. 9). We now define a third front defined as the left-hand limit of the jet zone in which holes appear [this limit is indicated by a green arrow in Fig. 6(b)] (the *holes front*). The successive positions of this front are extracted from a visual analysis of a set of images such as Fig. 6(b) and correspond to the blue stars in Fig. 9. These dots line up and lead to a propagation velocity $u_{holes} = 3850$ m/s. As $u_{m_{max,c}}^* \approx u_{holes}$, the two definitions of fragmentation front position are in practice equivalent from the point of view of measuring the velocity of propagation of the fragmentation front, so depending on the information available, either of these definitions may be used. In this work, we chose the velocity u_{holes} , because the position of the holes in the jet is easily detectable, and the appearance of the holes leads to a strong perturbation of the density profiles, which is also easy to measure.

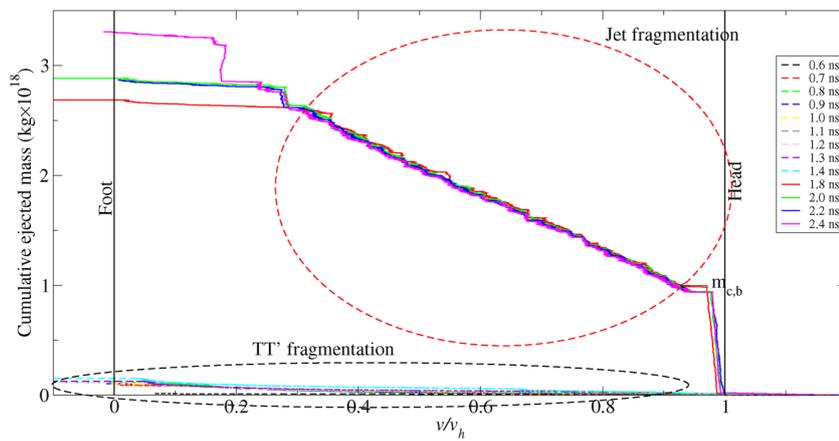
The final step is to assess the thickness of the jet. Thickness (Fig. 10) is measured by detecting the jet surface using a density criterion, as we did to measure jet velocity. For a given x coordinate, we calculate the density profile along the z axis, defined as the

mean value with respect to y . For each profile, two points are identified, corresponding to the two sides of the jet. The thickness is the absolute value of the difference between these two coordinates. The thickness is an average value along y , and significant variations around this value are possible. At time $t = 1.1$ ns (i.e., just before the holes appear in the zone Z_0), the thickness of the jet is constant in the Z_0 zone and is approximately 1.2 nm. On the other hand, the thickness of the jet increases progressively in zone Z_1 . At time $t = 1.5$ ns, corresponding to the start of zone Z_1 fragmentation, the thickness of the jet is no longer significant in zone Z_0 because of its fragmentation, hence the very chaotic curve observed in this zone.

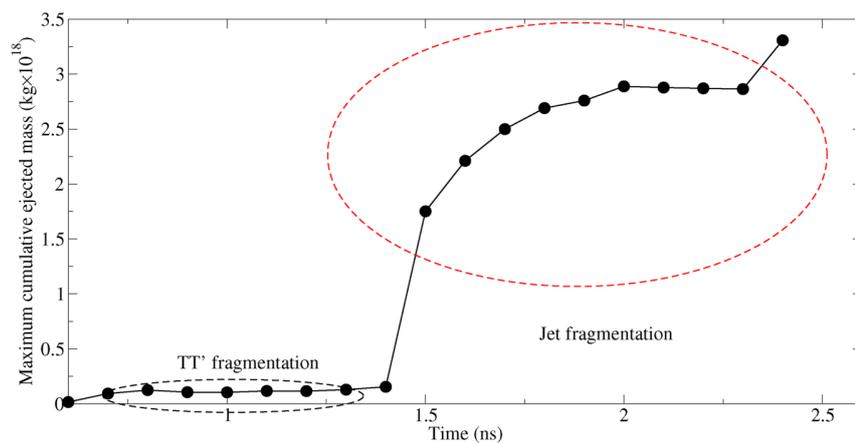
The constant thickness of the zone is the result of two opposing effects (Fig. 11). The first is the formation of the bulge at the head of the jet, which progressively sucks in the surrounding material and causes a local thickening of the jet. The second is the establishment of a material velocity gradient in the jet due to the difference between the velocities at the head and foot of the jet. Since the matter is assumed to be incompressible, this velocity gradient leads to a thinning of the



(a)



(b)



(c)

FIG. 7. MD simulation: cumulative ejected mass as a function of (a) the position and (b) the velocity of the ejecta. (c) Final cumulative ejected mass.

23 August 2024 09:13:19

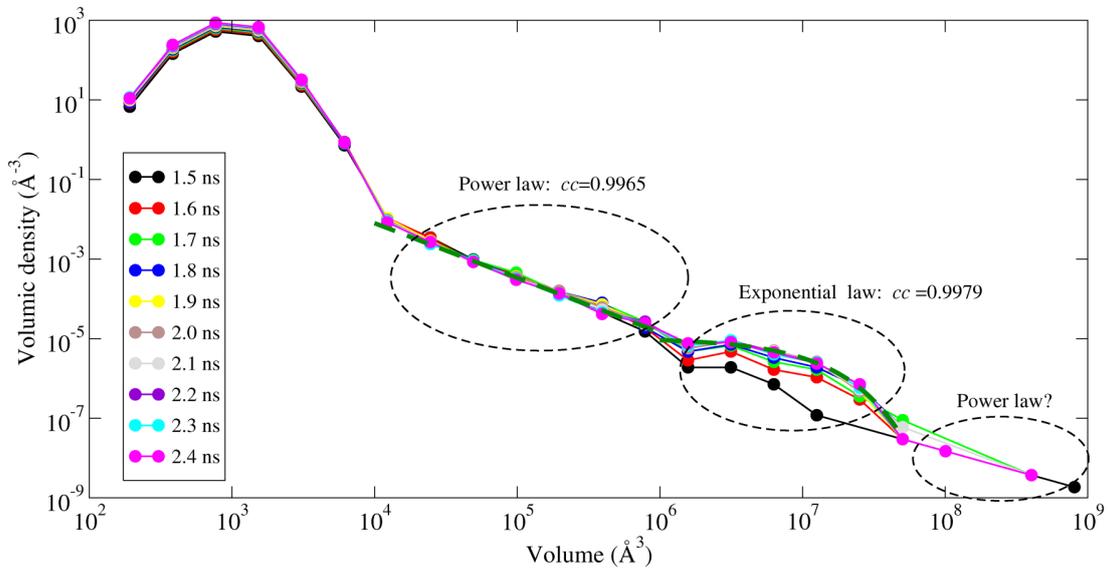


FIG. 8. Ejecta volume distribution. The dark green lines correspond to a power law or an exponential law. cc are de correlation coefficients between the numerical points and the proposed analytical forms.

jet. The combination of these two effects results in a zone of approximately constant thickness.

The final parameters we used are listed in Table II (all the parameters are expressed in coordinates system moving with the foot of the jet). From Fig. 7, we have $m_{c,b} = 10^{-18}$ kg. At the end of

the Z_0 zone [i.e., $v = v^{(1)}$ in Eq. (18)], the total cumulative mass is $m_{c,b} + m_{c,0} = 1.62 \times 10^{-18}$ kg.

Agreement with the MD is good for zone Z_0 and fairly good for zone Z_1 (Fig. 12). Adjusting $v_{fr}^{(1)}$ on the basis of the MD gives a velocity of -1463 m/s in the coordinate system

23 August 2024 09:13:19

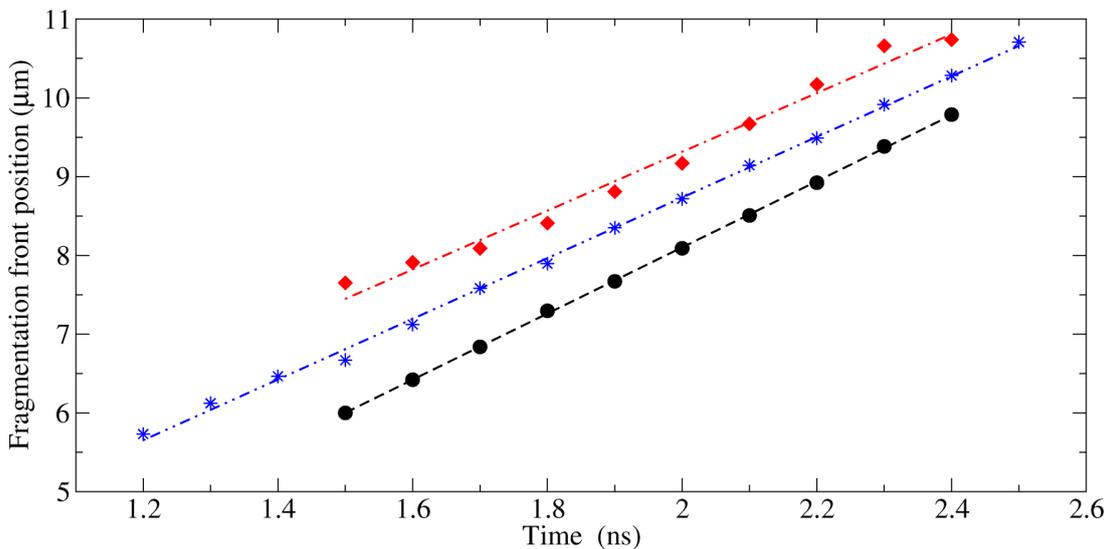


FIG. 9. Front trajectory. The black circles correspond to the $m_{c,max}$ front, the red diamonds to the $m_{max,c}^*$ front, and the blue stars to the holes front.

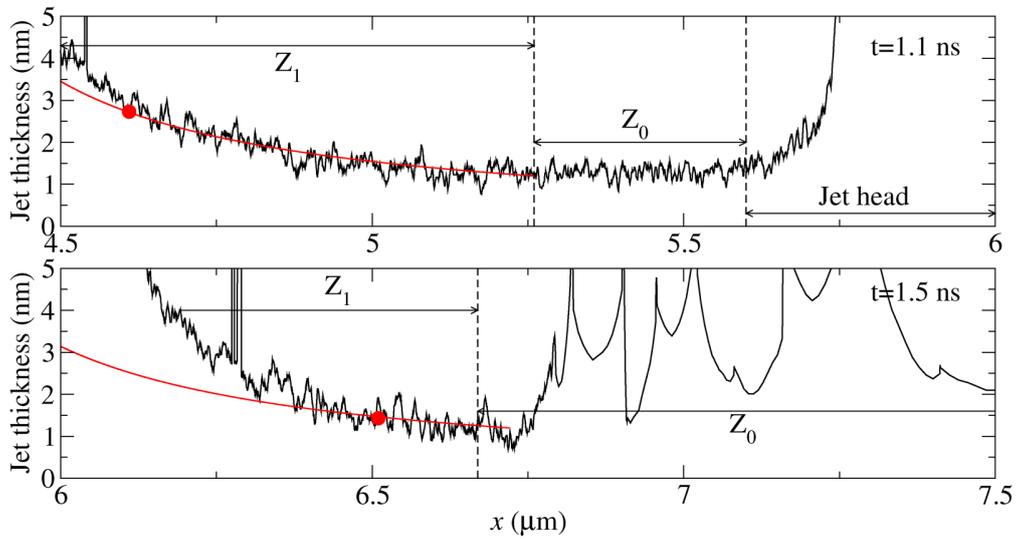


FIG. 10. Thickness of the jet just before the holes appear in the zone Z_0 (top), and at the start of the fragmentation of the zone Z_1 (bottom). The red curves correspond to the theoretical thickness calculated using Eq. (17).

moving with the foot of the jet, which is significantly different from the measured velocity (-350 m/s). This improves the agreement with MD only at the end of the zone, which could indicate the possible existence of a third zone. However, we will not pursue this analysis further.

The theoretical jet shape corresponding to the MD simulation was calculated using the parameters in Table II for the two instants of the Fig. 10. For $t = 1.1$ ns, we observe good agreement between the calculated profile (in red) and the profile obtained from the simulation (in black) up to a limit fixed approximately by the red

23 August 2024 09:13:19

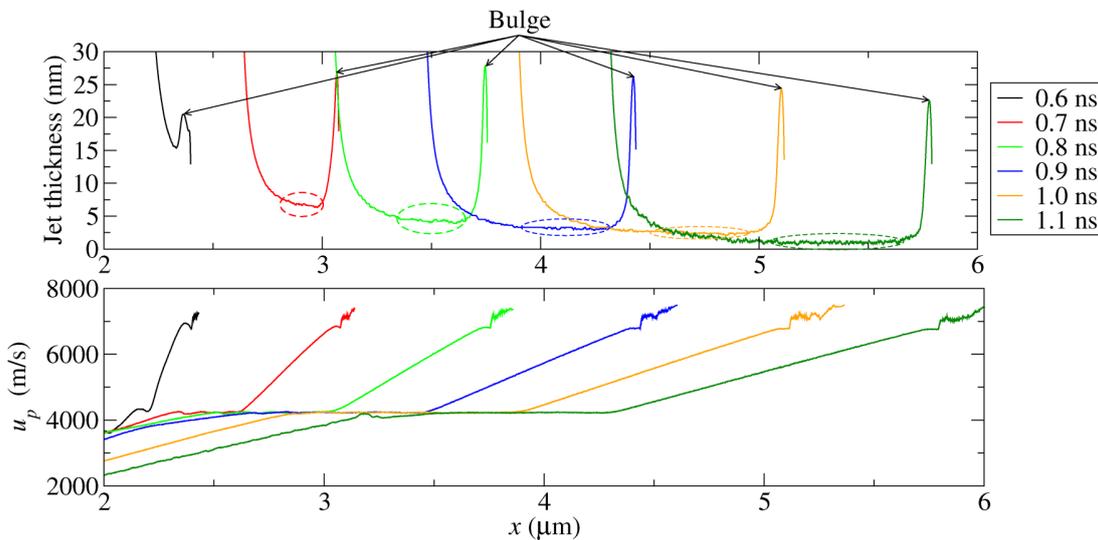


FIG. 11. Top: changes in the thickness of zone Z_0 (marked by ellipses) and the bulge before the start of the jet break. Bottom, corresponding material velocity profiles, illustrating the gradient to which zone Z_0 is subjected.

TABLE II. FZP parameters corresponding to the MD simulation.

Z_n	ρ_j (kg/m ³)	e_{cr} (nm)	$t^{(n)}$ (s)	$v_{fr}^{(n)}$ (m/s)	$v^{(n)}$ (m/s)	$u_{fr}^{(n)}$ (m/s)	$u^{(n)}$ (m/s)
Z_0	4694	1.2	...	$\pm\infty$	2437	$\pm\infty$	
Z_1	4694	1.2	0.974×10^{-9}	-350	1740	3850	5940

dot. At a later time ($t = 1.5$ ns), the agreement is still correct in the area bounded by the red dot, corresponding to the same element of the jet. The relationship (17) therefore provides an acceptable description of the jet shape within the validity limit of the FZP model.

The difference observed between the FZP/BMPT-2 combination simply marks the limit of these models. In the end, however, the differences remain fairly small, and the phenomenology described by FZP is well observed in the simulation. This shows that there is no incompatibility between the simulation and the proposed modeling. We now turn to a real case, using experimental data.

B. Asay foil experiments

To measure the ejected mass, a very common experimental device is the Asay foil:⁷⁷⁻⁷⁹ a thin metal foil is placed in front of the free surface and is thus impacted by the particles (Fig. 13).

Using a velocimetry device [VISAR (Velocity Interferometer System for Any Reflector), PDV (Photonic Doppler Velocimetry)],

the velocity of the foil $\mathbf{u}_s(t)$ and its evolution over time can be used to determine the mass of material impacting the target at a given moment. To achieve this, it is necessary to make a number of assumptions and several theoretical treatments have been proposed in the literature.^{77,80-84} In this work, we will assume that the relationship between $\mathbf{u}_s(t)$ and the mass having impacted the target is known and that it is based on the simplest theory.⁷⁷ The choice of this model is purely illustrative and does not change the proposed approach. We assume (i) that the jet appears at time $t = 0$ and (ii) that all particles are created at this time. In other words, we neglect for the moment the duration of all the processes taking place between the jet's appearance and the start of its rupture. This approximation is justified if the Asay is placed far enough away from the sample so that the characteristic duration of the different stages of jet breakup is very small compared with the time of flight of the particles. Under these conditions, all particles impacting the foil at time t_e move at the same constant velocity $\mathbf{u}_e(t_e)$. Assuming that the collision between a particle and the foil is totally inelastic (no back-splashing), the conservation of momentum leads to⁷⁷

$$m_c(t) = M_s(0)\mathcal{H}(t), \quad \mathcal{H}(t) = \int_{t_0}^t \frac{t_e du_{s,x}(t_e)}{x_s(0) + \int_{t_0}^{t_e} u_{s,x}(t') dt'} dt_e, \quad (19)$$

where $M_s(0)$ is the initial mass of the foil, t_0 the time corresponding to the first impact on the foil, x_s the position of the impact foil surface. To eliminate the dependence on S_m , the areal density $\sigma = m_c/S_m$ is generally used,

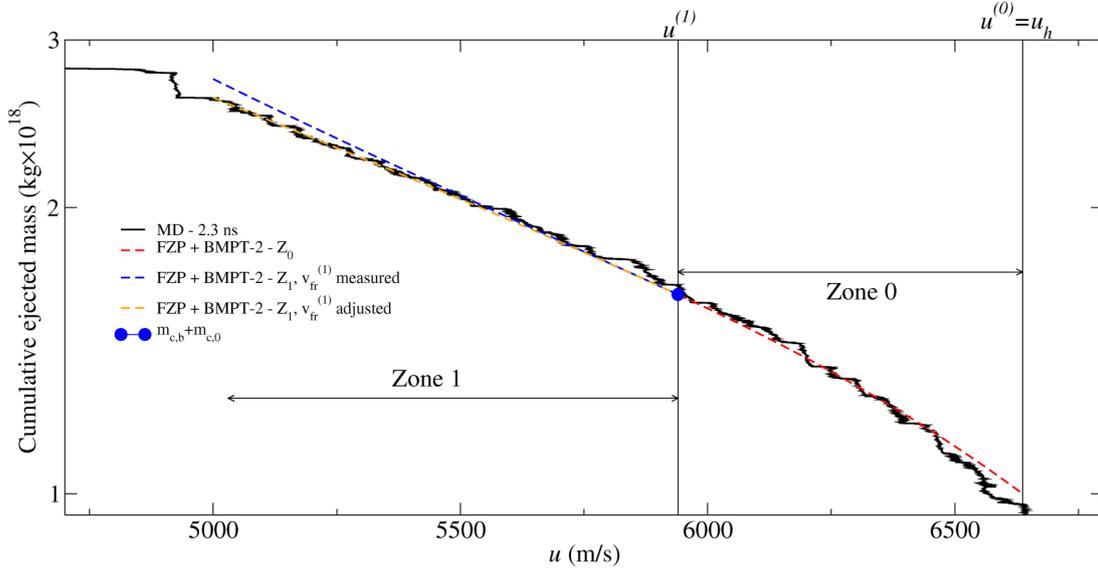


FIG. 12. Comparison of the cumulative ejected mass of the MD model (black curve) and FZP (semi-logarithmic representation). The red curve corresponds to zone Z_0 and is calculated using Eq. (18). The blue curve corresponds to zone Z_1 using the velocity $v_{fr}^{(1)}$ from the simulation (Table II), and the orange curve by adjusting $v_{fr}^{(1)}$ on the cumulative ejected mass obtained from simulation.

23 August 2024 09:13:19

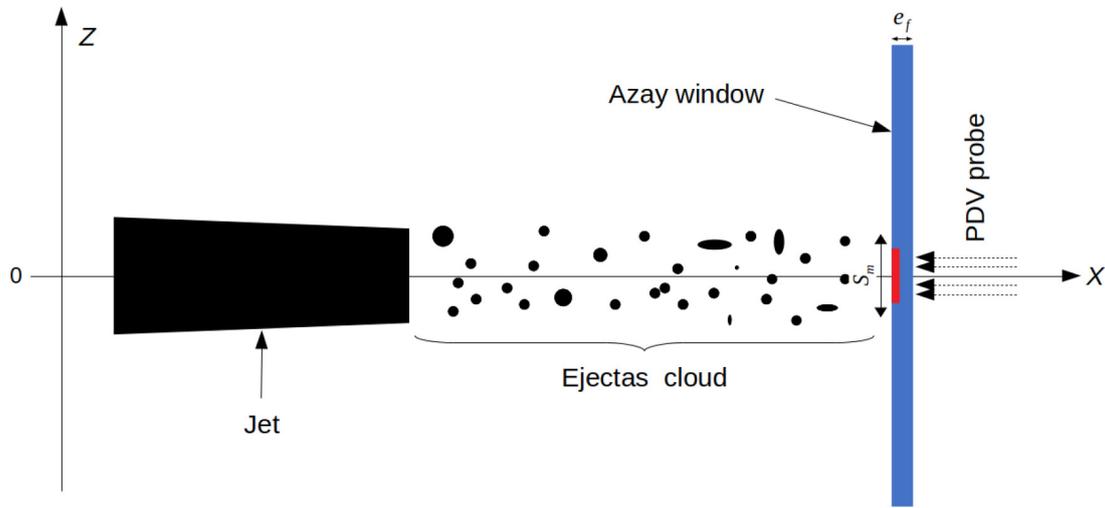


FIG. 13. Asay foil experiment. A jet partially breaks up to form a cloud of ejecta. This cloud impacts a metal foil whose velocity is measured by a PDV probe. S_m is the measurement surface.

$$\sigma(t) = \int_{t_0}^t \frac{m_e(t_e)}{S_m} dt_e = \rho_f e_f \mathcal{H}(t), \quad (20)$$

with ρ_f being the density of the foil at ambient conditions and e_f the thickness of the foil (we neglect the variation in the density of the sheet, i.e., the variation in its thickness, so that ρ_f is constant).

The link between $\sigma(t)$ [or $m_e(t)$] and $\mathcal{H}(t)$ is very interesting because its measurement, which is now well mastered, could provide thermodynamic (density, temperature) and structural (liquid, solid) information on the properties of the material in the jet but also in the bulk material (the problem is quite similar to the study of spalling, in which the velocity of the free surface bears witness, albeit very indirectly, to the damage mechanisms within the material).

We performed ejection experiments using a powder gun facility. A copper projectile was propelled at a velocity of 1550 m/s, against a 60 mm diameter tin plate uniformly grooved on its free

surface with a network of triangular grooves [wavelength λ and depth h , Fig. 1(a)]. The shock generated a pressure of 24 GPa in the sample, so that tin could melt. The tin sample was in contact with vacuum (pressure of 10–2 mbar). We performed four different experiments, with different grooves, in order to generate different mass distributions of ejecta. For each experiment, the mass distribution was inferred from the velocity measurement of an Asay foil, made of steel, using a photonic Doppler velocimetry (PDV) probe.

The characteristics of the four experiments analyzed in this study are presented in Table III. The distance d between the grooved surface and the foil is larger than 10 cm so that the angle $\tan^{-1}(\lambda/2d)$ remains small enough to ensure that the speeds on the foil are uniform. The shock pressure is calculated from the experimental velocity of the free surface and the relationship (u_s, u_p) given in Sec. III A 2. Jet velocity and density are calculated using the model BMPT-2. We note that the groove shapes of the S_2 , S_3 and S_4 experiments are almost identical (same β angle). Figure 14 shows the evolution of experimental areal densities.

TABLE III. Characteristics of the four experiments (shots S_1, \dots, S_4). The velocity of the planar free surface (u_{pfs}) is measured directly during the experiment. The pressure at the emergence of the shock on the free surface (p_{S_i}) is deduced from u_{pfs} and the $u_s = 2590 + 1.49u_p \text{ m/s}^{75}$ relation. The velocity v_j and density ρ_j of the jet are deduced from the BMPT-2 model.

Shot	$\lambda \times h$ ($\mu\text{m} \times \mu\text{m}$)	Distance target/probe (cm)	Foil thickness (μm)	β (degrees)	u_{pfs} (m/s) (Exp.)	p_{S_i} (GPa) (Exp.)	v_j (m/s) (BMPT-2)	ρ_j (kg/m^3) (BMPT-2)
S_1	25×8	10	100	57.38	1854	26.35	1304	5942
S_2	60×8	10	100	75.06	1855	26.35	935	6578
S_3	200×25	30	100	75.96	1850	26.25	907	6613
S_4	400×50	50	400	75.96	1780	24.98	897	6646

23 August 2024 09:13:19

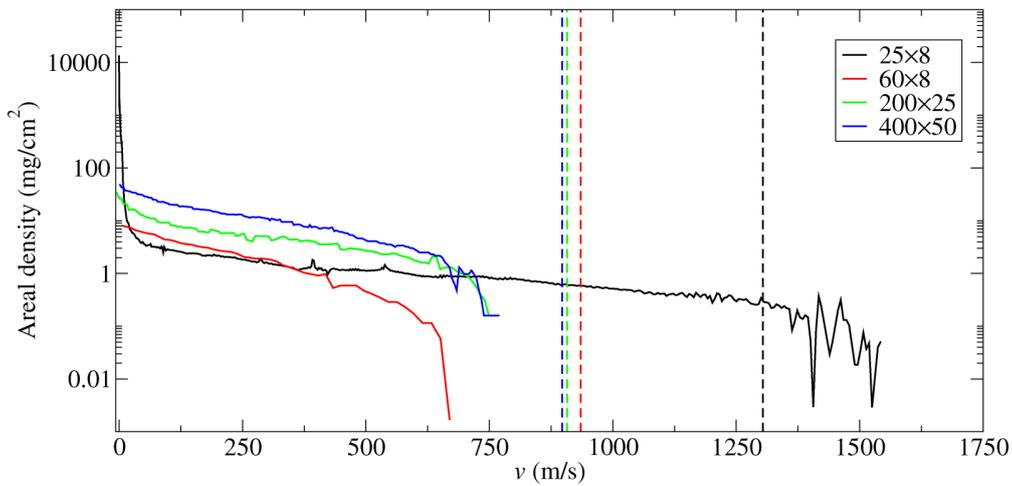


FIG. 14. Experimental areal density. Velocities are measured in the reference frame of the experimental free surface velocity u_{pfs} (Table III). The vertical lines show the jet velocities v_j calculated with the BMPT-2.

The use of FZP for Asay foil requires a number of precautions. The measurement surface S_m is a circle of radius r_m centered at $y = 0$ and $z = 0$. Let us consider the case where a single jet density of the foil at ambient areal mass $\sigma = m_c/S_m$ is independent of the choice of S_m as long as $r_m \leq e_p$. The situation is different in the case of the usual experimental configurations involving networks of grooves assumed to be identical. For the condition of homogeneity of the velocity over the entire measurement zone to be met, the distribution of the z components of the ejecta velocities must be homogeneous and of sufficient maximum amplitude to spray the target identically at all points. In experiments, the distance between the grooved surface and the probe is generally large (a few cm) compared with the wavelength of the groove network (less than a mm), so that this condition is verified even for relatively low v velocities. Finally, it must be assumed that the jets, once formed, do not interact with each other. The cumulative mass measured is then simply the sum of the contributions of each of the jets, which are all identical to a first approximation since it is assumed that the grooves themselves are all identical. We will therefore assume that the areal mass corresponding to a Z_n zone is

$$\sigma_n = \frac{N_j m_c^{(n)}}{S_m}, \quad (21)$$

where N_j is the number of jets contributing to the measured ejected mass. As we are working per unit area, the shape of the foil plays no role and we can assume, for simplicity, a square foil with side $2r_m$. In addition, $y_j = 2r_m$, $N_j = 2r_m/\lambda$ and $S_m = 4r_m^2$, so

$$\sigma_n = \frac{\rho_j}{\lambda} \mu_n \left(v^{(n)} - v_{fr}^{(n)} \right) \log \frac{v^{(n)} - v_{fr}^{(n)}}{v - v_{fr}^{(n)}}, \quad (22)$$

with $\mu_n = e_{cr} t^{(n)}$. Equation (6) shows that $v_{fr}^{(n)}(t^{(n)}) = v_{fr}^{(n)}(\mu_n)$. Fixing $\{v^{(n)}\}$, we see that only the products $e_{cr} t^{(n)}$ can be calculated from the σ curve.

In all cases, ρ_j is calculated using BMPT-2. Observation of the experimental curves shows that the zone of high velocity (i.e., the zone Z_0) is linear, which corresponds to the hypothesis of an infinite (or very high) v_{fr} velocity [Eq. (7)]. The slope sets the coefficient $\mu_0 = e_{cr} t^{(0)} = e_{cr} t^{(1)}$, and the minimum velocity of the zone fixes $v^{(1)}$. For the moment, let us set the number of zones at 2. The parameters to be determined are μ_1 , μ_2 , $v^{(0)}$ and $v^{(2)}$, the latter parameter setting the limit of validity of FZP.

- $v^{(0)}$. There are two possible choices. Either $v^{(0)} = v_h$ calculated with BMPT-2 but the maximum velocity measured (v_{max}) must be greater than v_h , or $v^{(0)} = v_{max}$.
- $v^{(2)}$. With the exception of shot S_1 , we can assume $v^{(2)} = v_{min}$ (the minimum measured velocity). We will come back to the case of shot S_1 later.
- μ_1, μ_2 . We can either choose the μ_1 coefficient deduced from the Z_0 zone, and adjust the μ_2 coefficient to the Z_1 zone, or adjust the coefficients μ_1 and μ_2 on the Z_1 zone, and calculate the whole profile with these coefficients.

The four possible options are summarized in Table IV and the corresponding parameters in Table V.

TABLE IV. The four possible choices for adjusting experimental surface masses, and their applicability to the experiments analyzed.

Option	$v^{(0)}$	μ_1	Applicable to shot
1.0	v_{max}	From Z_0	all
1.1	v_{max}	From Z_1	all
2.0	v_h	From Z_0	S_1
2.1	v_h	From Z_1	S_1

23 August 2024 09:13:19

TABLE V. FZP parameters from the experiments, depending on the choices made for the fitting procedure (see Table IV).

Shot	$v^{(1)}$ (m/s)	Option	$\mu_0 = \mu_1$ (m s)	$v_{fr}^{(0)}$ (m/s)	μ_2 (m s)	$v_{fr}^{(1)}$ (m/s)
S ₁	880	1.1	3.98×10^{-14}	$\pm\infty$	1.23×10^{-10}	20
		1.0	3.77×10^{-14}	$\pm\infty$	1.23×10^{-10}	20
		2.1	3.98×10^{-14}	$\pm\infty$	1.23×10^{-10}	20
		2.0	3.68×10^{-14}	$\pm\infty$	1.23×10^{-10}	20
S ₂	464	2.1	5.41×10^{-13}	$\pm\infty$	7.94×10^{-12}	-28
		2.0	2.76×10^{-13}	$\pm\infty$	1.24×10^{-10}	39
S ₃	331	2.1	3.78×10^{-12}	$\pm\infty$	2.82×10^{-10}	0.5
		2.0	3.61×10^{-12}	$\pm\infty$	2.97×10^{-10}	0.9
S ₄	357	2.1	1.89×10^{-11}	$\pm\infty$	3.69×10^{-10}	-14
		2.0	1.36×10^{-12}	$\pm\infty$	1.41×10^{-9}	15

For S₁, $v_{max} > v_h$ and all the options are applicable. The adjustment procedure shows that $v^{(2)}$ must be at least equal to 20 m/s. It is therefore possible that a third zone exists below this velocity, but the small size of this zone and the experimental resolution do not allow a sufficiently discriminating adjustment to be made. We note that v_h marks the boundary between a relatively noiseless region ($v < v_h$) and a very noisy region ($v > v_h$), which could indicate that this region corresponds to particles ejected in front of the jet head, resulting, for example, from the interaction of the shock with surface defects in the groove itself. The four options give comparable agreement with the experiment, with the four mass per unit area curves being virtually indistinguishable (Fig. 15). Finally, the velocity $v_{fr}^{(1)}$ of the rupture front in the Z₁ zone is around 20 m/s, whatever option is chosen.

For S₂, S₃ and S₄, $v_{max} < v_h$ and only options 1.0 and 1.1 are therefore applicable. For S₂ (Fig. 16), none of the options is really satisfactory. On the other hand, for S₃ (Fig. 17) and S₄ (Fig. 18), the situation is fairly similar to that for S₁, since both options reproduce the Z₁ zone very well, and the Z₀ zone fairly well, except for large values of v . Note that these regions correspond to the smallest areal masses, and are therefore the most sensitive to experimental uncertainty. In these three cases, $v_{fr}^{(1)}$ is close to that of the S₁, and is around a few tens of m/s. Finally, the third zone envisaged in case S₁ was not observed, since the choice $v^{(2)} = v_{min}$ for the three cases leads to good agreement with the experiments.

The experimental results are therefore fairly faithfully reproduced by FZP. The equality $\mu_1 = \mu_0$ is verified to a very good approximation for the S₁, S₃ and S₄ experiments, which, in the context of FZP, indicates that the critical thickness is constant throughout the jet.

IV. DISCUSSION

The experimental curves can be used to determine only the parameters μ_n and cannot therefore give any information on the value of the critical thickness in the absence of information on the times $t^{(n)}$. However, it is possible to extract a few orders of magnitude from the temporal characteristics of the experiment. The $t^{(0)} = \frac{\mu_0}{e_{cr}}$ curves are plotted in Fig. 19 for values of e_{cr} between 1 nm and 20 μ m. The critical thickness derived from the MD is clearly incompatible with the experimental μ_0 , since it leads to $t^{(0)}$ values between $\approx 40 \mu$ s and ≈ 15 ms. Experimental measurements are made over maximum durations of a few tens of μ s, giving an upper limit at $t^{(0)}$, and Fig. 19 shows that the critical thickness must be a few μ m. However, in both the simulation and the experiment, FZP

23 August 2024 09:13:19

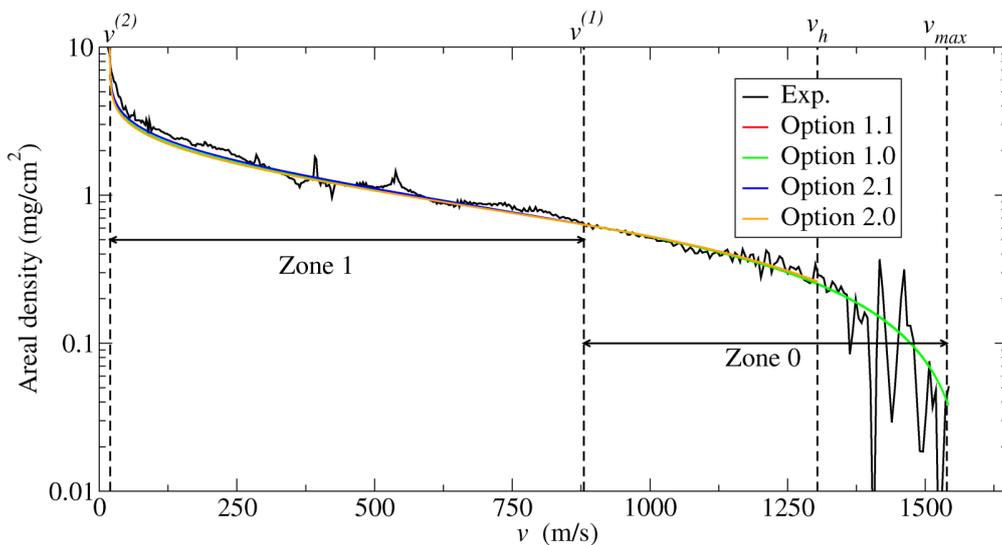


FIG. 15. Shot S₁. Comparison of the experimental mass per unit area (black curve) and its fit with FZP and the choice of fitting procedure.

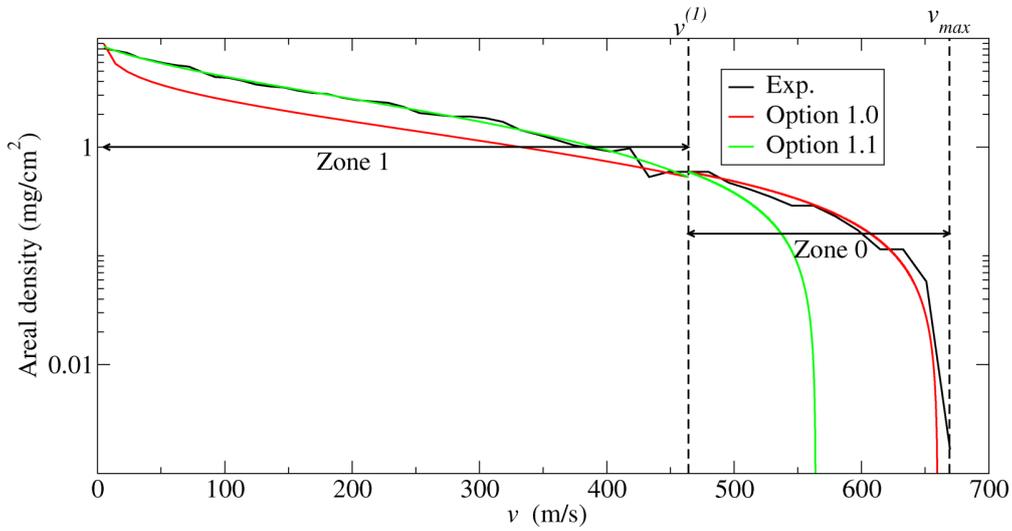


FIG. 16. Shot S_2 . Comparison of the experimental mass per unit area (black curve) and its fit with FZP and the choice of fitting procedure.

reproduces the observed data fairly well. We can therefore assume that his hypotheses are valid in both cases, and in particular the existence of a critical fracture thickness. However, the origin of e_{cr} is probably different in the experiment and the simulation. In the simulations, in the absence of initial structural defects, and in particular grains, the thickness results from variations in the thickness of the jet due to thermal fluctuations, as has been shown.⁶⁵ The presence of grains, and therefore grain boundaries, adds a further point of brittleness. Indeed, even in the case of melting, the jet

must retain, at least in part, the initial microstructure, as the characteristic times of the microjet are very short and melting may not be complete. The critical thickness would then be linked to the size of the grains, i.e., a few μm .

It is not possible to calculate the shape of the jet from the experimental data, as these only give indications of the velocities of the fronts, not their positions, and can therefore only be used to determine the product $\mu_n = e_{cr} t^{(n)}$. Nevertheless, we can expect an analytical form described by Eq. (17).

23 August 2024 09:13:19

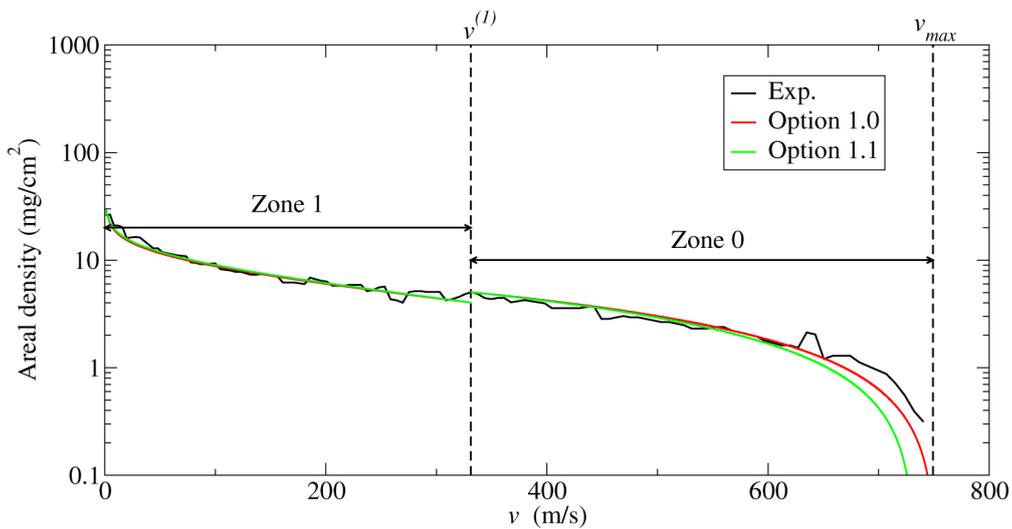


FIG. 17. Shot S_3 . Comparison of the experimental mass per unit area (black curve) and its fit with FZP and the choice of fitting procedure.

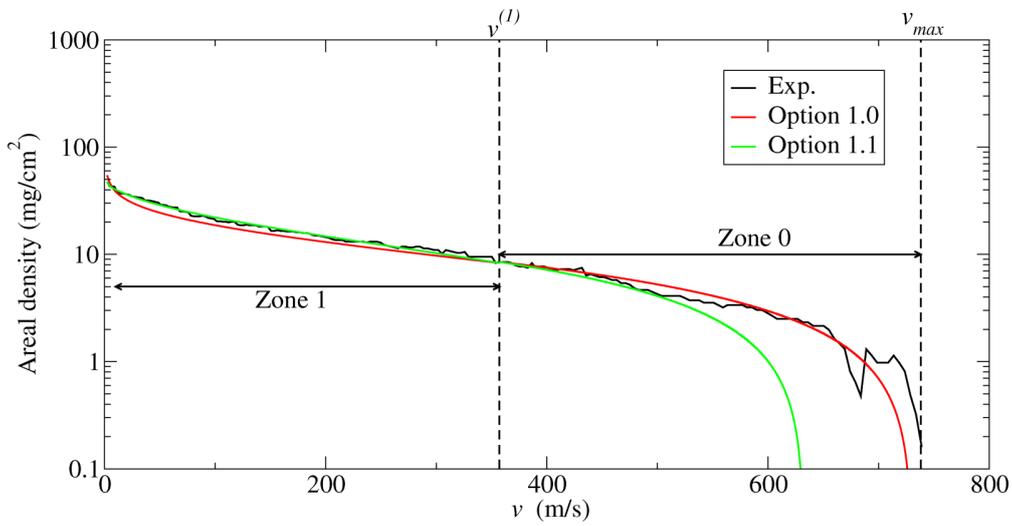


FIG. 18. Shot S₄. Comparison of the experimental mass per unit area (black curve) and its fit with FZP and the choice of fitting procedure.

In the literature, the mass per unit area is often adjusted using an empirical exponential law [work by Monfared *et al.*⁸⁵ and Self-Similar Velocity Distribution (SSVD) model,^{62,76} for example]. It should be noted that this type of formula does not result from a hypothesis about a jet fragmentation mechanism, but from the observation of experimental curves of mass per unit area. The results we obtained in this work do not correspond very well to an exponential distribution, as shown in the Fig. 20. It is, of course,

always possible to improve the fit by adjusting by zone, as we did with FZP. However, the FZP+BMPT-2 combination reduces the number of adjustable parameters to 3 for two adjustment zones, i.e., the same number of parameters as would result from the use of two exponential laws (the parameters are linked to ensure the continuity of the curve). The advantage of the model we are proposing is that it gives a precise physical meaning to each of the parameters, and therefore potentially allows them to be measured directly by an

23 August 2024 09:13:19

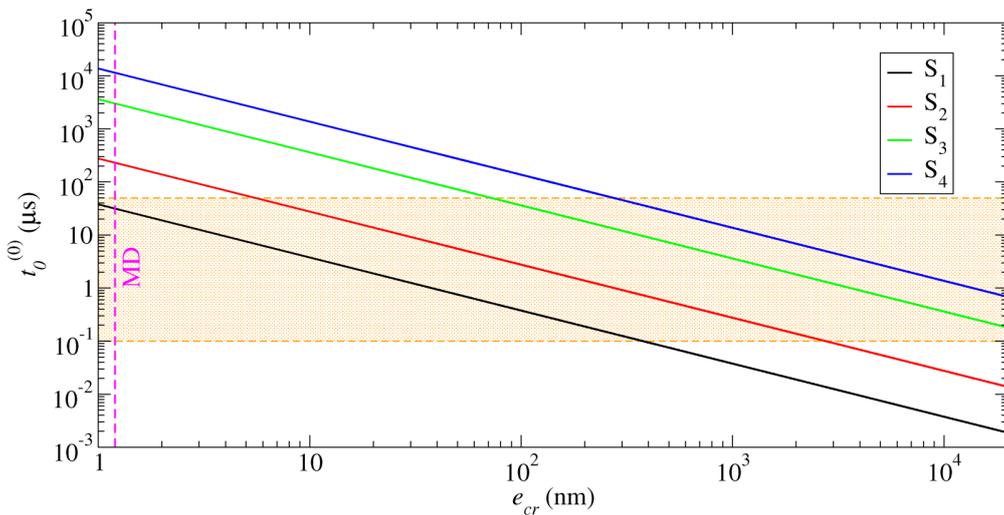
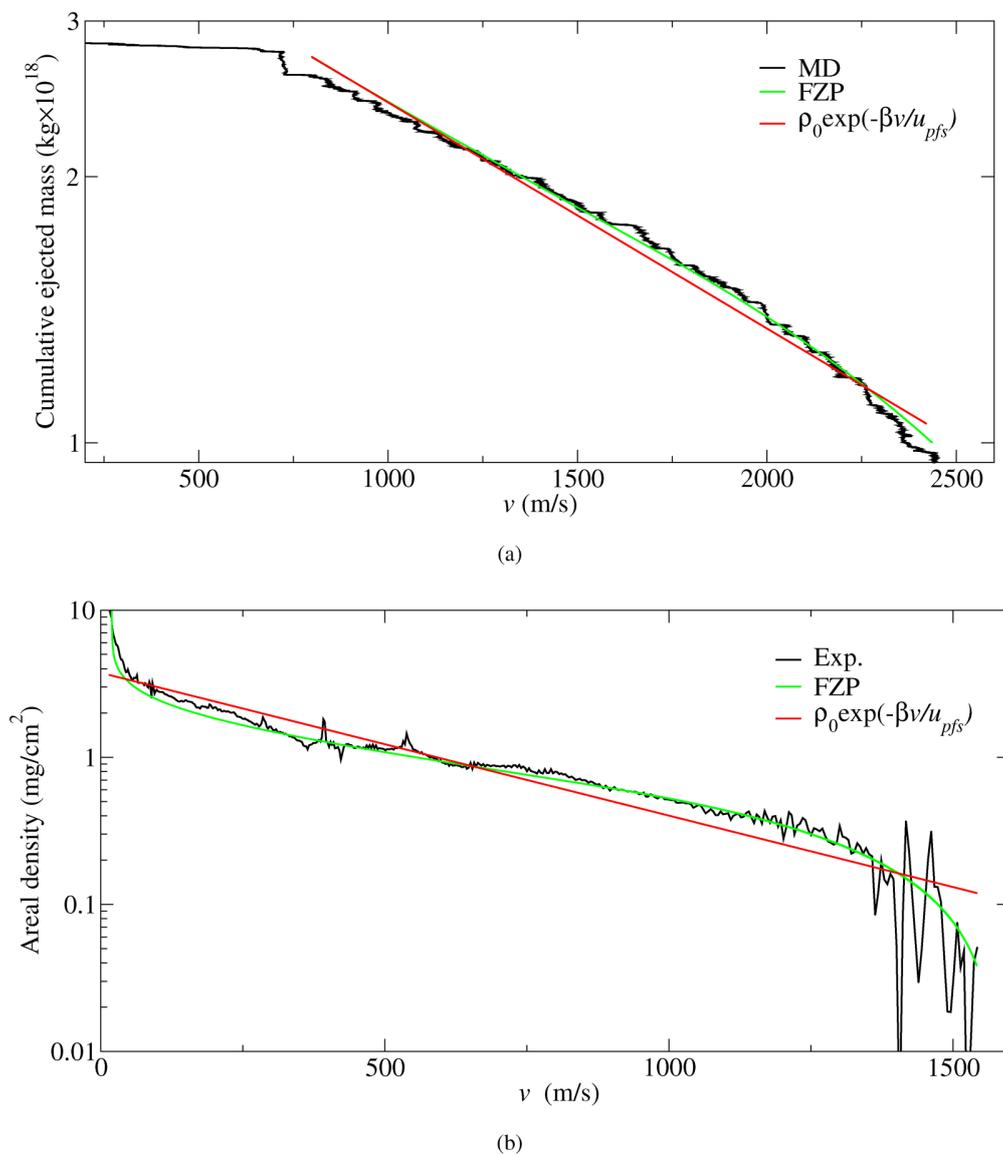


FIG. 19. Curves $t_0^{(0)} = \frac{\mu_0}{\rho_0 v}$ for the four experiments, using the values in Table V. The vertical magenta line corresponds to the critical thickness of the MD simulation. The orange zone corresponds to the duration of the observations.



23 August 2024 09:13:19

FIG. 20. Comparison of the FZP model (green curves) and an exponential law (red curves). The region of adjustment is that of the FZP validity domain. (a) MD results. (b) Experimental results (S_1 experiment).

appropriate experiment. This is not necessarily the case with an empirical law, such as the exponential law.

V. CONCLUSION

In this paper, we propose a method combining two models, FZP and BMPT-2, to calculate the ejected areal mass when a sustained shock reflects off a triangular groove. Based on both experimental observations and MD analyses, FZP is built around two

main hypotheses. The first hypothesis suggests that the fragmentation of the jet begins near its head and then propagates toward its foot. The entire jet is divided into a number of zones (one to three), each characterized by a constant propagation velocity. The second hypothesis of FZP is that jet rupture only occurs below a critical thickness, which may be specific to each zone. The analytical form resulting from these assumptions makes it possible to relate the cumulative areal mass to the velocity and requires a certain number of parameters, including the density and velocity of

the jet. These two quantities can be calculated using the BMPT-2 model, presented in detail in the previous article.³ The relevance of the FZP + BMPT-2 combination was first tested on a MD simulation and then on experiments. Our analysis is based on a single simulation because, since computing time is necessarily limited, we prefer a simulation that limits size effects as much as possible to multiple simulations of smaller dimensions, which are therefore likely to be much less accurate. This choice is based on our previous work in which we observed that BMPT-2 remains valid for a varied collection of opening angles and groove depths³ and that the FZP hypotheses are verified for grooves of different shapes, depths, and wavelengths from the one used in this work, and do not seem to depend on the choice of interatomic potential selected (EAM, MEAM).^{4,5} In the case of MD, it is possible to obtain a very accurate estimate of the cumulative areal mass based on measurements of the critical thickness and propagation velocities of the fracture front. It is also possible to deduce the theoretical shape of the jet in accordance with the FZP hypotheses. The very simple analytical expression obtained leads to good agreement with the jet profile obtained directly from the simulation. The experimental results obtained with the Asay foil also agree well with the FZP formalism. Analysis of the results suggests, however, that the origin of the fracture is different: thermal fluctuations in the case of MD, remanence of the initial microstructure in the case of the experiment. The result is very different thicknesses of rupture: of the order of nm in the case of MD, and a few μm in the case of experiments. Nevertheless, the general phenomenology is the same and always can be described by FZP. If FZP is not specific to a particular groove shape, BMPT-2 as such is only applicable to the case of a triangular groove. For grooves of other shapes, the solution is either to modify BMPT-2 accordingly, or to use other models such as those derived from RMI theory, and associate them to FZP.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Laurent Soulard: Conceptualization (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Supervision (equal); Writing – original draft (equal); Writing – review & editing (equal). **Olivier Durand:** Conceptualization (equal); Formal analysis (equal); Investigation (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Jean-René Burie:** Conceptualization (equal); Investigation (equal). **Killian Babilotte:** Software (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

¹J. Thouvenin, *Détonique* (Eyrolles, 1997).

²G. Ben-Dor, *Shock Wave Reflection Phenomena* (Springer, 2007).

³L. Soulard, “Micro-jetting: A semi-analytical model to calculate the velocity and density of the jet from a triangular groove,” *J. Appl. Phys.* **133**(8), 085901 (2023).

⁴O. Durand and L. Soulard, “Mass-velocity and size-velocity distributions of ejecta cloud from shock-loaded tin surface using atomistic simulations,” *J. Appl. Phys.* **117**(16), 165903 (2015).

⁵O. Durand and L. Soulard, “Modeling from molecular dynamics simulations of ejecta production induced by shock-loaded metallic surfaces,” *J. Dyn. Behav. Mater.* **3**, 280–290 (2017).

⁶S. A. Dyachkov, A. N. Parshikov, and V. V. Zhakhovsky, “Shock-produced ejecta from tin: Comparative study by molecular dynamic and smoothed particle hydrodynamics methods,” *J. Phys.: Conf. Ser.* **653**, 012043 (2015).

⁷Y. N. Shi, Q. Zhao, P. Wang, Q. J. Feng, and C. S. Qin, “Study on oblique shock-induced micro-jet ejection from a grooved metal surface,” in *Advances in Fluid Mechanics XI, WIT Transactions on Engineering Sciences*, edited by C. A. Brebbia (WIT Press, 2016), Vol. 105, pp. 253–262.

⁸C. Roland, T. de Resseguier, A. Sollier, E. Lescoute, L. Soulard, and D. Loison, “Hydrodynamic simulations of microjetting from shock-loaded grooves,” *AIP Conf. Proc.* **1793**, 100027 (2016).

⁹C. Roland, T. de Resseguier, A. Sollier, E. Lescoute, D. Tandiand, M. Toulminet, and L. Soulard, “Ballistic properties of ejecta from a laser shock-loaded groove: SPH versus experiments,” *AIP Conf. Proc.* **1979**, 080012 (2018).

¹⁰V. Karkhanis, P. Ramaprabhu, W. T. Buttler, J. E. Hammerberg, F. J. Cherne, and M. J. Andrews, “Ejecta production from second shock: Numerical simulations and experiments,” *J. Dyn. Behav. Mater.* **2**, 265–279 (2017).

¹¹V. Karkhanis, P. Ramaprabhu, F. J. Cherne, J. E. Hammerberg, and M. J. Andrews, “A numerical study of bubble and spike velocities in shock-driven liquid metals,” *J. Appl. Phys.* **123**(2), 025902 (2018).

¹²V. Karkhanis and P. Ramaprabhu, “Ejecta velocities in twice-shocked liquid metals under extreme conditions: A hydrodynamic approach,” *Matter Radiat. Extrem.* **4**(4), 044402 (2019).

¹³F. Wang, J. Liu, A.-M. He, P. Wang, and J.-G. Wang, “Ejecta production from metal Sn into inert gases,” *J. Appl. Phys.* **127**(19), 195901 (2020).

¹⁴K. K. Mackay, F. M. Najjar, S. J. Ali, J. H. Eggert, T. Haxhimali, B. E. Morgan, H. S. Park, Y. Ping, H. G. Rinderknecht, C. V. Stan, and A. M. Saunders, “Hydrodynamic computations of high-power laser drives generating metal ejecta jets from surface grooves,” *J. Appl. Phys.* **128**(21), 215904 (2020).

¹⁵M. P. Hennessey, H. K. Springer, and J. L. Belof, “Modification of Richtmyer–Meshkov instabilities via layered explosive charge design,” *J. Appl. Phys.* **134**(24), 245901 (2023).

¹⁶O. Durand, S. Jaouen, L. Soulard, O. Heuze, and L. Colombet, “Comparative simulations of microjetting using atomistic and continuous approaches in the presence of viscosity and surface tension,” *J. Appl. Phys.* **122**, 135107 (2017).

¹⁷T. de Resseguier, E. Lescoute, A. Sollier, G. Prudhomme, and P. Mercier, “Microjetting from grooved surfaces in metallic samples subjected to laser driven shocks,” *J. Appl. Phys.* **115**(4), 044402 (2014).

¹⁸D. S. Sorenson, P. Pazuchanics, R. Johnson, R. M. Malone, M. I. Kaufman, A. Tibbitts, T. Tunnell, D. Marks, G. A. Capelle, M. Grover, B. Marshall, G. D. Stevens, W. D. Turley, and B. LaLone, Ejecta particle-size measurements in vacuum and helium gas using ultraviolet in-line Fraunhofer holography, Report No. LA-UR-15-25993 (Los Alamos National Laboratory, 2015).

¹⁹T. de Resseguier, C. Roland, E. Lescoute, A. Sollier, D. Loison, L. Berthe, G. Prudhomme, and P. Mercier, “Experimental study of microjetting from triangular grooves in laser shock-loaded samples,” *AIP Conf. Proc.* **1793**, 100025 (2017).

²⁰A. Sollier and E. Lescoute, “Characterization of the ballistic properties of ejecta from laser shock-loaded samples using high resolution picosecond laser imaging,” *Int. J. Impact Eng.* **136**, 103429 (2020).

²¹G. Prudhomme, T. de Resseguier, C. Roland, A. Sollier, E. Lescoute, D. Loison, and E. Brambrink, “Velocity and mass density of the ejecta produced from sinusoidal grooves in laser shock-loaded tin,” *J. Appl. Phys.* **128**(15), 155903 (2020).

²²M. Xu, W. Song, C. Wang, P. Wang, J. Shao, and E. Tang, “Theoretical and experimental study on the hypervelocity impact induced microjet from the grooved metal surface,” *Int. J. Impact Eng.* **156**, 103944 (2021).

- ²³D. R. Guildenbecher, A. McMaster, A. Corredor, B. Malone, J. Mance, E. Rudziensky, D. Sorenson, J. Danielson, and D. L. Duke, "Ultraviolet digital holographic microscopy (DHM) of micron-scale particles from shocked Sn ejecta," *Opt. Express* **31**(9), 14911–14936 (2023).
- ²⁴G. Chu, T. Xi, M. Yu, W. Fan, Y. Zhao, M. Shui, W. He, T. Zhang, B. Zhang, Y. Wu, W. Zhou, L. Cao, J. Xin, and Y. Gu, "High-energy x-ray radiography of laser shock loaded metal dynamic fragmentation using high-intensity short-pulse laser," *Rev. Sci. Instrum.* **89**(11), 115106 (2018).
- ²⁵G. Prudhomme, J.-E. Franzkowiak, T. de Resseguier, E. Brambrink, C. Roland, D. Loison, E. Lescoute, and A. Sollier, "Ejecta from periodical grooves in tin foils under laser-driven shock loading," *AIP Conf. Proc.* **1979**, 080010 (2018).
- ²⁶T. de Resseguier, G. Prudhomme, C. Roland, E. Brambrink, D. Loison, B. Jodar, E. Lescoute, and A. Sollier, "Picosecond x-ray radiography of microjets expanding from laser shock-loaded grooves," *J. Appl. Phys.* **124**(6), 065106 (2018).
- ²⁷T. de Resseguier, C. Roland, G. Prudhomme, E. Brambrink, J. E. Franzkowiak, D. Loison, E. Lescoute, A. Sollier, and L. Berthe, "Picosecond radiography combined with other techniques to investigate microjetting from laser shock-loaded grooves," *AIP Conf. Proc.* **1979**, 080011 (2018).
- ²⁸J. Xin, A. He, W. Liu, G. Chu, M. Yu, W. Fan, Y. Wu, T. Xi, M. Shui, Y. Zhao, P. Wang, Y. Gu, and W. He, "X-ray radiography of microjetting from grooved surfaces in tin sample subjected to laser driven shock," *J. Micromech. Microeng.* **29**(9), 095011 (2019).
- ²⁹A. M. Saunders, S. J. Ali, H.-S. Park, J. Eggert, F. Najjar, C. Huntington, T. Haxhimali, B. Morgan, Y. Ping, and H. G. Rinderknecht, "Development of high power laser platforms to study metal ejecta interactions," *AIP Conf. Proc.* **2272**, 120025 (2020).
- ³⁰W. Liu, T. Xi, A. He, T. Zhou, J. Xin, N. Liu, and P. Wang, "Numerical and experimental study of the second ejection from a grooved tin surface under laser-driven shock loading," *Int. J. Impact Eng.* **161**, 104135 (2022).
- ³¹O. Durand and L. Soulard, "Large-scale molecular dynamics study of jet breakup and ejecta production from shock-loaded copper with a hybrid method," *J. Appl. Phys.* **111**, 044901 (2012).
- ³²O. Durand and L. Soulard, "Power law and exponential ejecta size distributions from the dynamic fragmentation of shock-loaded Cu and Sn metals under melt conditions," *J. Appl. Phys.* **114**, 194902 (2013).
- ³³B. Li, F. P. Zhao, H. A. Wu, and S. N. Luo, "Microstructure effects on shock-induced surface jetting," *J. Appl. Phys.* **115**(7), 073504 (2014).
- ³⁴H. An-Min, W. Pei, S. Jian-Li, and D. Su-Qing, "Molecular dynamics simulations of jet breakup and ejecta production from a grooved Cu surface under shock loading," *Chin. Phys. B* **23**(4), 047102 (2014).
- ³⁵J.-L. Shao, P. Wang, and A.-M. He, "Microjetting from a grooved Al surface under supported and unsupported shocks," *J. Appl. Phys.* **116**(7), 073501 (2014).
- ³⁶J.-L. Shao, P. Wang, and A.-M. He, "Influence of shock pressure and profile on the microjetting from a grooved Pb surface," *Model. Simul. Mater. Sci. Eng.* **25**(1), 015011 (2017).
- ³⁷F. Wu, Y. Zhu, X. Li, P. Wang, Q. Wu, and H. Wu, "Peculiarities in breakup and transport process of shock-induced ejecta with surrounding gas," *J. Appl. Phys.* **125**(18), 185901 (2019).
- ³⁸G. Tang, B. Li, X. Deng, L. Wang, W. Zhu, W. Hu, and F. Gao, "Comparative investigation of microjetting from tin surface subjected to laser and plane impact loadings via molecular dynamics simulations," *Mech. Mater.* **148**, 103479 (2020).
- ³⁹O. Durand, L. Soulard, L. Colombet, and R. Prat, "Influence of the phase transitions of shock-loaded tin on microjetting and ejecta production using molecular dynamics simulations," *J. Appl. Phys.* **127**, 175901 (2020).
- ⁴⁰B. Wu, F. Wu, P. Wang, A. He, and H. Wu, "Shock-induced ejecta transport and breakup in reactive gas," *Phys. Chem. Chem. Phys.* **22**(26), 14857–14867 (2020).
- ⁴¹S.-W. Sun, G.-Q. Tang, Y.-F. Huang, L.-Z. Cao, and X.-P. Ouyang, "Comparative investigation of microjetting generated from monocrystalline tin surface and polycrystalline tin surface under plane impact loading," *Chin. Phys. B* **30**(10), 104701 (2021).
- ⁴²D. Jin, X. Deng, and W. Liu, "Eulerian peridynamic modeling of microjetting from a grooved aluminum sample under shock loading," *J. Appl. Phys.* **131**(10), 105108 (2022).
- ⁴³X. Yang, H. Zhao, X. Gao, G. Lei, and Z. Chen, "Molecular dynamics study on micro jet in single crystal aluminum," *Acta Mech. Sin.* **39**(3), 122232 (2023).
- ⁴⁴B. Wu, A. He, X. Wang, H. Sun, and P. Wang, "Numerical investigation of ejecta mass of twice-shocked liquid Sn," *J. App. Phys.* **133**(16), 165903 (2023).
- ⁴⁵R. Richtmyer, "Taylor instability in shock acceleration of compressible fluids," *Commun. Pure Appl. Math.* **13**, 297–319 (1960).
- ⁴⁶E. E. Meshkov, "Instability of the interface of two gases accelerated by a shock wave," *Izv. AN SSSR. Mekhanika Zhidkosti i Gaza* **4**(5), 151–157 (1969).
- ⁴⁷K. O. Mikaelian, "Effect of viscosity on Rayleigh–Taylor and Richtmyer–Meshkov instabilities," *Phys. Rev. E* **47**(1), 375–383 (1993).
- ⁴⁸K. O. Mikaelian, "Freeze-out and the effect of compressibility in the Richtmyer–Meshkov instability," *Phys. Fluids* **6**(1), 356–368 (1994).
- ⁴⁹K. O. Mikaelian, "Analytic approach to nonlinear hydrodynamic instabilities driven by time-dependent accelerations," *Phys. Rev. E* **81**(1,2), 016325 (2010).
- ⁵⁰G. Birkhoff, D. P. MacDougall, E. M. Pugh, and S. Geoffrey Taylor, "Explosives with lined cavities," *J. Appl. Phys.* **19**(6), 563–582 (1948).
- ⁵¹W. P. Walters and J. A. Zukas, *Fundamentals of Shaped Charges* (John Wiley & Sons, 1989).
- ⁵²K. O. Mikaelian, "Analytic approach to nonlinear Rayleigh–Taylor and Richtmyer–Meshkov instabilities," *Phys. Rev. Lett.* **80**(3), 508–511 (1998).
- ⁵³G. Dimonte, C. E. Frerking, M. Schneider, and B. Remington, "Richtmyer–Meshkov instability with strong radiatively driven shocks," *Phys. Plasmas* **3**(2), 614–630 (1996).
- ⁵⁴G. Dimonte and P. Ramaprabhu, "Simulations and model of the nonlinear Richtmyer–Meshkov instability," *Phys. Fluids* **22**(1), 014104 (2010).
- ⁵⁵G. Dimonte, G. Terrones, F. J. Cherne, T. C. Germann, V. Dupont, K. Kadau, W. T. Buttler, D. M. Oro, C. Morris, and D. L. Preston, "Use of the Richtmyer–Meshkov instability to infer yield stress at high-energy densities," *Phys. Rev. Lett.* **107**(26), 264502 (2011).
- ⁵⁶G. Dimonte, G. Terrones, F. J. Cherne, and P. Ramaprabhu, "Ejecta source model based on the nonlinear Richtmyer–Meshkov instability," *J. Appl. Phys.* **113**(2), 024905 (2013).
- ⁵⁷F. J. Cherne, J. E. Hammerberg, M. J. Andrews, V. Karkhanis, and P. Ramaprabhu, "On shock driven jetting of liquid from non-sinusoidal surfaces into a vacuum," *J. Appl. Phys.* **118**(18), 185901 (2015).
- ⁵⁸R. J. R. Williams, "The late time structure of high density contrast, single mode Richtmyer–Meshkov flow," *Phys. Fluids* **28**(7), 074108 (2016).
- ⁵⁹J. E. Hammerberg, W. T. Buttler, F. J. Cherne, M. J. Andrews, V. Karkhanis, P. Ramaprabhu, G. D. Stevens, and W. D. Turley, *J. Dyn. Behav. Mater.* **3**, 316–320 (2017).
- ⁶⁰J. R. Asay and L. D. Bertholf, "A model for estimating the effects of surface roughness on mass ejection from shocked materials," Report No. SAND-78-1256 (Sandia Laboratories, 1978).
- ⁶¹R. J. R. Williams, "Ejecta sources and scalings," *AIP Conf. Proc.* **1979**, 080015 (2018).
- ⁶²A. K. Harrison, "The spike dynamics source model for ejecta in the flag code," in *Proceedings of the ASME/JSME/KSME Joint Fluids Engineering Conference, 2019, ASME-JSME-KSME Joint Fluids Engineering Conference (AJK-FED 2019)*, San Francisco, CA, July 28–August 1, 2019 (Amer. Soc. Mechanical Engineers, 2019), Vol. 5, p. V005T05A058.
- ⁶³K. Kadau, T. C. Germann, N. G. Hadjiconstantinou, P. S. Lomdahl, G. Dimonte, B. L. Holian, and B. J. Alder, "Nanohydrodynamics simulations: An atomistic view of the Rayleigh–Taylor instability," *Proc. Natl. Acad. Sci. U.S.A.* **101**(16), 5851–5855 (2004).
- ⁶⁴J.-L. Shao, P. Wang, A.-M. He, S.-Q. Duan, and C.-S. Qin, "Atomistic simulations of shock-induced microjet from a grooved aluminium surface," *J. Appl. Phys.* **113**(15), 153501 (2013).
- ⁶⁵O. Durand, L. Soulard, E. Bourasseau, and G. Filippini, "Investigation of the static and dynamic fragmentation of metallic liquid sheets induced by random surface fluctuations," *J. Appl. Phys.* **120**, 045306 (2016).

- ⁶⁶B. Wu, F. Wu, Y. Zhu, P. Wang, A. He, and H. Wu, "Molecular dynamics simulations of ejecta production from sinusoidal tin surfaces under supported and unsupported shocks," *AIP Adv.* **8**(4), 045002 (2018).
- ⁶⁷F. A. Sapozhnikov, G. V. Ionov, V. V. Dremov, L. Soulard, and O. Durand, "The embedded atom model and large-scale MD simulation of tin under shock loading," *J. Phys.: Conf. Ser.* **500**, 032017 (2014).
- ⁶⁸R. Ravelo and M. Baskes, "Equilibrium and thermodynamic properties of grey, white, and liquid tin," *Phys. Rev. Lett.* **79**(13), 2482 (1997).
- ⁶⁹L. Soulard and O. Durand, "Observation of phase transitions in shocked tin by molecular dynamics," *J. Appl. Phys.* **127**, 165901 (2020).
- ⁷⁰E. Bourasseau, G. Filippini, A. Ghoufi, and P. Malfreyt, "Calculation of the surface tension of pure tin from atomistic simulations of liquid-vapour systems," *Mol. Phys.* **112**(80), 2654–2657 (2014).
- ⁷¹W.-B. Liu, D.-J. Ma, A.-M. He, and P. Wang, "Ejecta from periodic grooved Sn surface under unsupported shocks," *Chin. Phys. B* **27**(1), 016202 (2018).
- ⁷²E. Cieren, L. Colombet, S. Pitoiset, and R. Namyst, "ExaStamp: A parallel framework for molecular dynamics on heterogeneous clusters," in *Euro-Par 2014: Parallel Processing Workshops, Lecture Notes in Computer Science* (Springer International Publishing, 2014), Vol. 8806, p. 121.
- ⁷³E. Dirand, L. Colombet, and B. Raffin, "Tins: A task-based dynamic helper core strategy for *in situ* analytics," in *Supercomputing Frontiers. SCFA 2018. Lecture Notes in Computer Science*, edited by R. Yokota, W. Wu (Springer, Cham, 2018), Vol. 10776, pp. 159–178.
- ⁷⁴R. Prat, L. Colombet, and R. Namyst, "Combining task-based parallelism and adaptive mesh refinement techniques in molecular dynamics simulations," *ICPP18, International Conference on Parallel Processing, August 2018, Eugene* (ICPP, 2018).
- ⁷⁵S. P. Marsh, *LASL Shock Hugoniot Data* (University of California Press, 1980).
- ⁷⁶J. E. Hammerberg, W. T. Buttler, A. Llobet, C. Morris, J. Goett, R. Manzanares, A. Saunders, D. Schmidt, A. Tainter, W. Vogan-McNeil, and C. Wilde, "Proton radiography measurements and models of ejecta structure in shocked Sn," *AIP Conf. Proc.* **1979**, 080006 (2018).
- ⁷⁷J. R. Asay, L. P. Mix, and F. C. Perry, "Ejection of material from shocked surfaces," *Appl. Phys. Lett.* **29**(5), 284–287 (1976).
- ⁷⁸J. R. Asay, "Thick-plate technique for measuring ejecta from shocked surfaces," *J. Appl. Phys.* **49**(12), 6173–6175 (1978).
- ⁷⁹C. W. McCluskey, M. D. Wilke, W. W. Anderson, M. E. Byers, D. B. Holtkamp, P. A. Rigg, M. D. Furnish, and V. T. Romero, "Asay window: A new spall diagnostic," *Rev. Sci. Instrum.* **77**(11), 113902 (2006).
- ⁸⁰Y. Chen, H. Hu, T. Tang, G. Ren, Q. Li, R. Wang, and W. T. Buttler, "Experimental study of ejecta from shock melted lead," *J. Appl. Phys.* **111**(5), 053509 (2012).
- ⁸¹Y. Chen, G. Ren, T. Tang, Q. Li, and H. Hu, "Experimental study of micro-spalling fragmentation from melted lead," *Shock Waves* **26**(2), 221–225 (2016).
- ⁸²Y. Chen, R. Hong, H. Chen, T. Tang, and G. Ren, "An improved Asay window technique for investigating the micro-spall of an explosively-driven tin," *Rev. Sci. Instrum.* **88**(1), 013904 (2017).
- ⁸³I. L. Tregillis, A. Koskelo, and A. K. Harrison, "Analytic solutions for Asay foil trajectories with implications for ejecta source models and mass measurements," *J. Appl. Phys.* **130**(12), 124504 (2021).
- ⁸⁴I. L. Tregillis and A. Koskelo, "A mathematical framework for ejecta cloud dynamics with application to source models and piezoelectric mass measurements," *J. Appl. Phys.* **130**(14), 144501 (2021).
- ⁸⁵S. K. Monfared, D. M. Oro, M. Grover, J. E. Hammerberg, B. M. LaLone, C. L. Pack, M. M. Schauer, G. D. Stevens, J. B. Stone, W. D. Turley, and W. T. Buttler, "Experimental observations on the links between surface perturbation parameters and shock-induced mass ejection," *J. Appl. Phys.* **116**(6), 063504 (2014).

