CSMA 2024
16ème Colloque National en Calcul des Structures
13-17 Mai 2024, Presqu'île de Giens (Var)

HPCrack : Un code massivement parallèle pour les simulations numériques des problèmes de rupture à partir d'images

Xiaodong Liu[1], Julien Réthoré[1]

[1] *Nantes Université, Ecole Centrale Nantes, CNRS, GeM, UMR 6183, F-44000, Nantes, France, {xiaodong.liu,julien.rethore}@ec-nantes.fr*

Résumé — HPCrack se démarque dans la simulation réaliste de la rupture de matériaux, opérant directement à l'échelle de voxel à partir d'images de grande envergure, sans idéalisation géométrique. Développé en C/C++, ce code polyvalent offre une flexibilité avec des configurations de parallélisme indépendantes, adaptées à diverses plates-formes. Cette approche en éliminant les idéalisations géométriques, permet une représentation précise des microstructures complexes tout en garantissant une efficacité computationnelle optimale.
Mots clés — Massivement parallèle, Simulation de rupture, Modélisation microstructurale, Efficacité computationnelle

1 Introduction générale

1.1 Contexte de développement

La croissance exponentielle de la taille des images captées par les capteurs contemporains pose des défis considérables, en particulier dans le domaine de l'imagerie tomographique. Les détails microstructuraux présents dans ces images massives deviennent difficilement accessibles en raison des limitations inhérentes aux méthodes traditionnelles d'analyse. Face à ce défi, notre projet se concentre sur une approche innovante : effectuer la simulation directement à l'échelle micro pour tirer pleinement parti des informations contenues dans ces images.

D'autre part, la modélisation microstructurale représente un défi complexe, accentué par la structure intrinsèquement complexe des matériaux et leur hétérogénéité. La procédure de maillage à l'échelle microstructurelle devient particulièrement ardue, entraînant des processus lourds et imposant des compromis dans l'idéalisation géométrique. Dans ce contexte, notre code HPCrack se positionne comme une réponse novatrice, proposant une approche basée sur la génération automatique de maillage à l'échelle des voxels. Cette approche élimine l'intervention humaine dans la phase de génération de maillage, assurant ainsi une modélisation plus précise et détaillée des propriétés microstructurales des matériaux.

Il est intéressant de noter que la genèse du code HPCrack remonte à la thèse du Xiaodong Liu [1], où une première version en C a été développée pour traiter des problèmes de conduction thermique et élastique à partir d'images à l'échelle de voxel. La version actuelle en C++ avec des solveurs plus stable et plus efficace a été conçue et perfectionnée au cours de son postdoc, reflétant ainsi une évolution significative du code pour répondre à des exigences plus avancées et diversifiées dans le domaine de la simulation numérique, notamment dans le domaine complexe de la propagation de rupture [2].

1.2 Introduction du HPCrack

Le code HPCrack a été spécialement développé pour la simulation de la propagation de fissures à partir d'images, en utilisant la méthode du champ de phase. Pour garantir une efficacité optimale, nous avons innové en proposant l'utilisation de la méthode des éléments finis sans matrices. Cette approche novatrice ne se contente pas seulement d'optimiser l'efficacité du parallélisme, mais elle offre également l'avantage significatif de réduire considérablement les besoins en mémoire, étant donné l'absence de la matrice de rigidité.

Le solveur employé dans HPCrack est le solveur itératif du gradient conjugué préconditionné. Afin d'accélérer la convergence de ce solveur itératif, nous avons introduit une couche de méthodes de multi-

grilles. Cette amélioration stratégique vise à optimiser la résolution du système linéaire associé à chaque itération, renforçant ainsi l'efficacité globale du processus de simulation.

En ce qui concerne la partie parallélisme, nous avons mis en place un système hybride MPI/OpenMP afin de garantir une adaptabilité optimale sur des machines dotées d'architectures variées. Cette approche hybride tire parti du parallélisme de tâches offert par OpenMP, idéal pour les architectures de mémoire partagée, et du parallélisme de données fourni par MPI, adapté aux environnements de calcul distribué. La flexibilité de HPCrack se manifeste pleinement dans sa capacité à configurer indépendamment toutes les options de parallélisme. Cela signifie que le code peut être exécuté de manière exclusive avec OpenMP, MPI ou une combinaison des deux, offrant ainsi une adaptabilité totale aux préférences et aux ressources disponibles. Cette caractéristique permet au code de s'ajuster facilement à une gamme étendue de configurations, depuis des ordinateurs portables jusqu'aux infrastructures régionales et nationales. Cette flexibilité architecturale renforce la portabilité du code HPCrack, lui permettant de s'adapter efficacement à une diversité de configurations matérielles.

Métadonnées du code HPCrack sont présentées dans le tableau 1. Comme indiqué dans le tableau, HPCrack est développé en langage C/C++. En effet, grâce à sa conception en C/C++, à l'absence de dépendances spécifiques, et à l'utilisation d'un Makefile dédié, HPCrack est hautement portable sur différentes machines. Cette portabilité renforce la flexibilité du code, permettant aux utilisateurs de déployer et d'exécuter efficacement HPCrack sur une variété de plates-formes sans rencontrer des problèmes liés à des configurations spécifiques.

Description des métadonnées du code	
Version actuelle du code	v 1.0
Lien permanent vers le code/dépôt utilisé pour cette version du code	`https://src.koda.cnrs.fr/xiaodong/hpcrack`
Licence juridique du code	CeCILL-B
Système de gestion de versions utilisé	git
Langages, outils et services utilisés pour le code	C++, MPI, OpenMP
Exigences de compilation, environnements d'exploitation & dépendances	Bibliothèque MPI, bibliothèque C++

TABLE 1 – Métadonnées du code HPCrack

En ce qui concerne les opérations d'entrée/sortie (I/O) de HPCrack, le code est configuré pour traiter des données d'entrée au format RAW, généralement des images. Pour la sortie, les résultats sont sauvegardés au format VTK (Visualization Toolkit), offrant ainsi une représentation visuelle et une interprétation efficace des données simulées.

Cette configuration simplifie l'intégration du code avec différentes sources de données d'entrée au format RAW, tandis que la sortie au format VTK assure une compatibilité avec des outils de visualisation largement utilisés dans le domaine de la modélisation et de la simulation.

2 Quelques examples de calculs

2.1 L'amorçage et la propagation des fissures dans la fonte

La Figure 1 expose les niveaux de gris d'une image de la fonte obtenue par [3] à l'échelle microscopique. Comme illustré, les nodules de graphite présentent des dimensions très réduites par rapport à l'échelle de l'échantillon, et leur géométrie est arbitraire. Cette complexité géométrique rend la génération de maillage extrêmement difficile. En optant pour une approche à l'échelle de voxel, ce problème est contourné.

La Figure 2 met en évidence l'amorçage de fissures dans la fonte. Contrairement à la propagation des fissures dans les matériaux homogènes, la trajectoire des fissures est fortement influencée par la présence de nodules de graphite. L'impact significatif de ces nodules sur l'amorçage et la propagation des fissures est clairement démontré. En conséquence, la surface de la fissure ne suit plus une géométrie plane, comme représenté dans la Figure 3. De manière concomitante, l'observation de fissures multiples

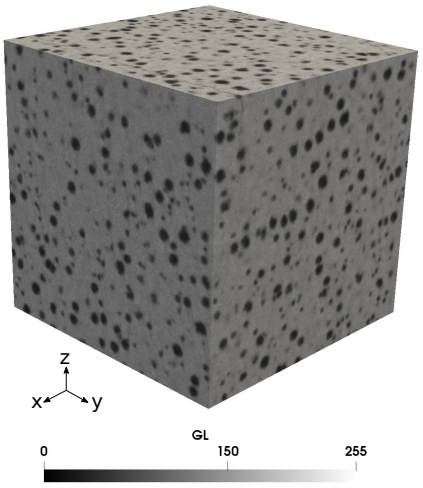

FIGURE 1 – L'image de la fonte

confirme l'effet complexe induit par la microstructure particulière de la fonte sur les phénomènes de fissuration.

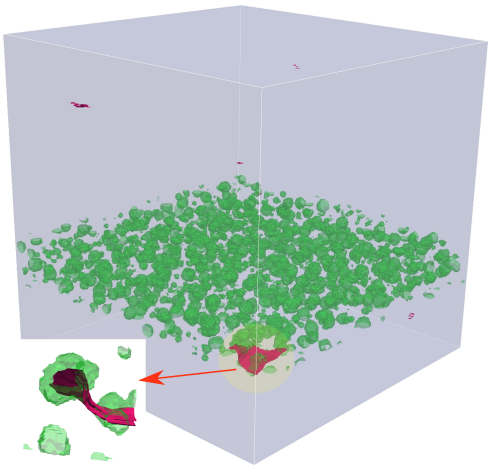

FIGURE 2 – L'amorçage de fissures

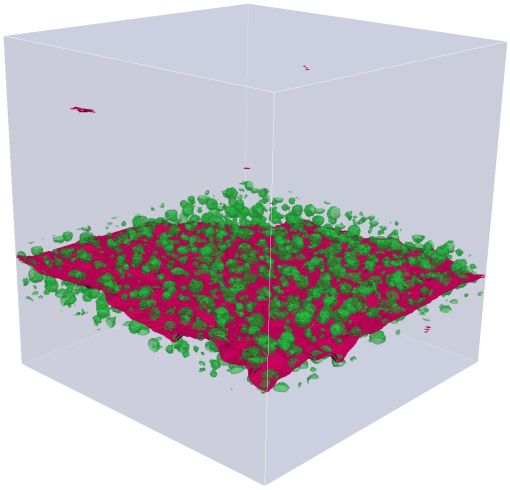

FIGURE 3 – Surface de fissurations

2.2 La propagation de fissures dans un matériau triphasé

Dans cette partie, nous explorons attentivement l'impact des interfaces dans la fonte, en la considérant comme un matériau triphasé complexe. Pour accéder à ces interfaces invisibles caractérisées par une épaisseur extrêmement fine, nous avons entrepris le calcul du gradient des niveaux de gris. Cette approche nous permet de dériver les variations subtiles de la transition entre les phases, contribuant ainsi à la délimitation précise de ces interfaces microstructurales.

Comme illustré dans la Figure 4, la trajectoire de la fissure a été perturbée de manière significative par la présence des nodules de graphite et des interfaces dans la fonte. On observe que la fissure progresse en suivant les contours des interfaces entre les nodules et la matrice métallique. Cette tendance s'explique par la recherche active de la fissure pour exploiter les variations de propriétés mécaniques et géométriques aux interfaces, induisant ainsi une trajectoire complexe et non linéaire.

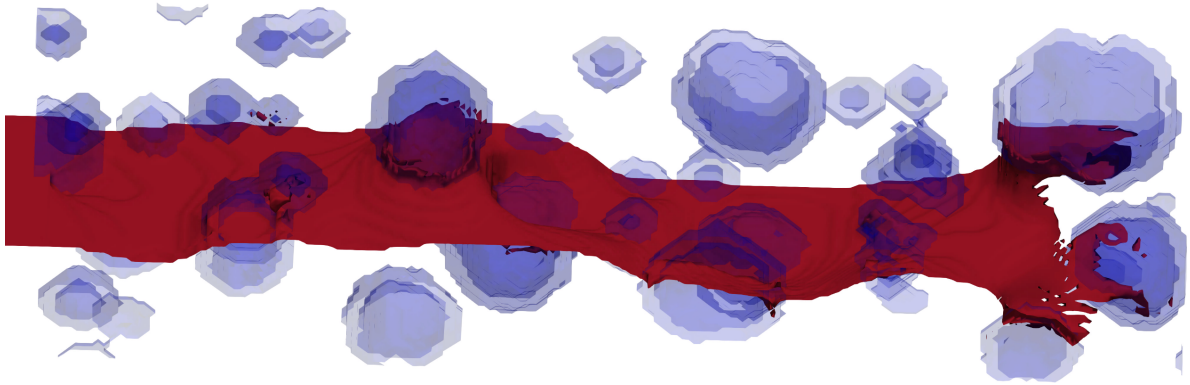

FIGURE 4 – Surface de fissurations

3 Conclusions et Perspectives

En conclusion, le code HPCrack représente une avancée significative dans le domaine de la simulation numérique des problèmes de rupture à partir d'images. Son approche massivement parallèle permet de traiter efficacement des images de grande taille, offrant ainsi la possibilité d'explorer les détails microstructuraux avec une précision inégalée. Les motivations sous-jacentes au développement de HPCrack, liées à la complexité croissante des images tomographiques et des modèles microstructuraux, ont été abordées de manière novatrice. La génération automatique de maillage à l'échelle des voxels constitue une contribution majeure, éliminant les contraintes liées à l'idéalisation géométrique dans la modélisation microstructurale.

Les perspectives pour le code HPCrack sont extrêmement prometteuses et étendent son domaine d'application bien au-delà des simulations de rupture. Le code présente un potentiel considérable dans des domaines variés tels que la conduction thermique, la corrélation d'images, et d'autres phénomènes physiques complexes. La flexibilité intrinsèque du code permet son adaptation à une diversité de problématiques scientifiques et technologiques.

L'extension du code HPCrack vers une version GPU représente une perspective extrêmement intéressante et prometteuse. Cette initiative ouvre la voie à des améliorations significatives en termes de performances de calcul parallèle, capitalisant sur la puissance des unités de traitement graphique. Les GPU, réputés pour leur capacité à traiter efficacement des tâches massivement parallèles, peuvent considérablement accélérer les simulations numériques, offrant ainsi des résultats plus rapides et plus précis.

Références

[1] Xiaodong Liu. *A massively parallel matrix free finite element based multigrid method for simulations of heterogeneous materials using tomographic images*. PhD thesis, École centrale de Nantes, 2019.

[2] Xiaodong Liu, Julien Réthoré, and Antonius Adrianus Lubrecht. An efficient matrix-free preconditioned conjugate gradient based multigrid method for phase field modeling of fracture in heterogeneous materials from 3d images. *Computer Methods in Applied Mechanics and Engineering*, 388 :114266, 2022.

[3] Johann Rannou, Nathalie Limodin, Julien Réthoré, Anthony Gravouil, Wolfgang Ludwig, Marie-Christine Baïetto-Dubourg, Jean-Yves Buffiere, Alain Combescure, François Hild, and Stéphane Roux. Three dimensional experimental and numerical multiscale analysis of a fatigue crack. *Computer methods in applied mechanics and engineering*, 199(21-22) :1307–1325, 2010.