

Visualisation des Maillages AMR *Tree-Based*

Visualization of Tree-Based AMR Meshes

Guéno   HAREL, Jacques-Bernard LEKIEN, and Philippe P. P  BA  

R  sum  —We present the first systematic treatment of the problems posed by the visualization and analysis of large-scale, parallel adaptive mesh refinement (AMR) simulations on an Eulerian grid. When compared to those obtained by constructing an intermediate unstructured mesh with fully described connectivity, our primary results indicate a gain of at least 80% in terms of memory footprint with a better rendering, while retaining similar execution speed. In this article, we describe the key concepts together with the methodology that facilitates the design, implementation, and optimization of algorithms operating directly on fine meshes. This native support for AMR meshes has been contributed to the open source Visualization Toolkit (VTK). This work pertains to a broader long-term vision, with the dual goal to both improve interactivity when exploring large AMR data sets in 2 and 3 dimensions, and optimize resource utilization.

1 INTRODUCTION

Nous pr  sentons ici une solution pour r  soudre les probl  mes li  s    la visualisation interactive et    l’analyse de maillages    raffinement adaptatif (AMR) sur grille rectilign  aire. La prise en compte *native* de cette repr  sentation a fait l’objet d’une contribution dans la biblioth  que *open source* VTK [4] sous la d  nomination `vtkHyperTreeGrid` (HTG), accompagn  e d’articles explicatifs [8], [9]. En se comparant    une approche n  cessitant la construction d’un maillage non structur   (NS), nos premiers r  sultats montrent une consommation m  moire r  duite d’un ordre de grandeur, un meilleur rendu visuel, ainsi qu’un temps de restitution inf  rieur. Ces r  sultats ont   t   permis par la mise en place de concepts innovants, facilitant l’  criture d’algorithmes optimis  s. Ce travail s’inscrit dans une vision globale d’am  lioration de l’interactivit   dans l’exploration de maillages AMR tout en optimisant l’utilisation des ressources informatiques.

2 CONTEXTE

Les simulations num  riques s’ex  cutant sur les super calculateurs HPC du CEA [2] g  n  rent des r  sultats de calcul de plus en plus volumineux. Parmi les codes de simulation, ceux qui utilisent l’AMR sont particuli  rement efficaces pour le suivi de d  tails fins au sein de tr  s grands domaines de calcul. L’AMR offre un compromis entre pr  cision num  rique, empreinte m  moire et co  t de calcul    travers une adaptation locale du maillage. Cette approche est limit  e    des probl  mes physiques o   les ph  nom  nes significatifs sont localis  s spatialement comme en astrophysique,

cf. RAMSES [12] ou dans les calculs de propagation d’onde [11]. Depuis la premi  re description AMR [5], plusieurs impl  mentations ont   t   propos  es et d  velopp  es. Tout particuli  rement, la repr  sentation du maillage sous la forme d’un arbre de d  composition (AMR *tree-based*), illustr   en Figure 1    gauche, est beaucoup moins gourmande en m  moire que son   quivalent par blocs (AMR *patch-based*). De nombreux codes AMR sont concern  s par cette approche en commen  ant par une racine initiale (*octree*) comme dans RAMSES, ou en utilisant une grille rectilign  aire de cellules racines comme dans RAGE/SAGE [7] ou HERA [10], un code de simulation d’hydrodynamique d  velopp   au CEA.

3 APPROCHE

Jusqu’   maintenant, le stockage, l’analyse et la visualisation de maillage n  cessitaient leur conversion totale ou partielle en maillages structur  s (S) ou non structur  s (NS), perdant tout l’int  r  t du choix de l’AMR fait par le code de simulation. Notre approche a   t   avant tout dict  e par la volont   de conserver cette repr  sentation compacte. VTK offre diff  rentes structures de donn  es accompagn  es d’une tr  s large gamme de traitements (*filters* dans la nomenclature VTK). Nous avons donc d  velopp   une nouvelle structure de donn  es `vtkHyperTreeGrid` dans VTK. Celle-ci r  pond nativement aux services de la structure NS, permettant d’utiliser l’ensemble des filtres pr  sents dans VTK. Par ailleurs, nous avons cr  e un ensemble de filtres sp  cialis  s, illustr  s en dimension 3 par la Figure 2. Afin de faciliter l’  criture de tels filtres ainsi que d’optimiser leur ex  cution, nous proposons l’emploi de *curseurs* et *super curseurs* pour le parcours des arbres AMR, organis  s en une hi  rarchie de complexit   variable afin de r  pondre aux besoins des diff  rents filtres, cf. Figure 1    droite. Parmi ces traitements, le filtre de rendu est particuli  rement

- G. H. : CEA, DAM, DIF, F-91297 Arpajon, France
E-mail: guenole.harel@cea.fr
- J.-B. L.: CEA, DAM, DIF, F-91297 Arpajon, France
E-mail: jacques-bernard.lekien@cea.fr
- P. P. P.: NexGen Analytics, Sheridan, WY 82801, U.S.A.
E-mail: philippe.pebay@ng-analytics.com

intéressant en dimension 2. Il consiste à produire un maillage NS minimal tenant compte du point de vue (les mailles AMR non visibles à l'écran ne sont pas produites, cf. Figure 3) et de la résolution de l'écran (le parcours de l'arbre s'arrête avant les mailles sub-pixel, cf. Figure 4). La valeur de la maille affichée peut être celle du code ou une valeur calculée par rapport aux mailles plus fines. Les performances d'affichage de ce filtre, en temps de restitution comme en mémoire, sont bornées par le nombre de pixels à l'écran et non par la taille du maillage AMR. Le temps de parcours de l'arbre reste négligeable devant le temps de rendu.

4 RÉSULTATS

Afin d'exploiter les données produites par nos différents codes de simulation numérique, le CEA a développé un outil métier basé sur VTK et ParaView dédié à la visualisation parallèle, Love [3]. Les résultats suivants ont été obtenus en utilisant cet outil. La simulation 3D d'astrophysique Aquarius [1] arrangée sur une grille mère uniforme de 32 mailles par direction spatiale comporte plus de 4 millions de mailles non raffinées réparties sur 14 niveaux.

TABLE 1

Comparaison de la consommation mémoire entre les différentes structures pour un maillage avec une grandeur.

	Maillage	Grandeur
AMR stockage Hercule	< 1 Mo	36.6 Mo
NS VTK Love	386 Mo	32.1 Mo
AMR VTK Love	15 Mo	36.6 Mo
S VTK Love	~ 0 Mo	10 ¹¹ Mo

TABLE 2

Comparaison du temps de restitution incluant le rendu entre les différentes structures pour un maillage avec une grandeur.

	Maillage	Grandeur	FPS	Iso-surface
NS VTK Love	33.9 s	20.4 s	2	14.3 s
AMR VTK Love	1.2 s	0.6 s	44	13.8 s

Les Tables 1 et 2 comparent, respectivement, les coûts de stockage avec la bibliothèque d'entrées/sorties du CEA, Hercule [6], ou en mémoire entre différentes structures VTK, ainsi que les temps de restitution pour une exécution séquentielle de Love.

Grâce à cette nouvelle structure, nous avons pu dépouiller un maillage de 300 millions de mailles sur une station de travail comportant 16 Go de mémoire, dépouillement impossible à faire avec les autres structures VTK. Le chargement se fait en moins d'une minute en séquentiel, avec un calcul d'iso-surface, illustré Figure 5, réalisé sur la même machine.

5 CONCLUSION

Cette nouvelle structure et ses filtres ont été reversés dans VTK. De nouveaux filtres de rendu optimisés

sont en cours d'implémentation afin de supprimer l'étape de production d'un maillage NS intermédiaire conformément à la vision que nous avons sur ce sujet.

Les apports de l'utilisation de cette nouvelle structure VTK sont multiples. Dans le cas du rendu 2D optimisé, l'exploitation d'un gros maillage est facilitée par l'exploration interactive, possible sur de simples stations de travail, ainsi que par le contrôle de l'information affectée au niveau des mailles affichées. En diminuant les ressources nécessaires en dimension 2 ou 3, on dispose d'une plus grande amplitude dans l'utilisation de celles-ci. Ce dernier point est particulièrement important dans le cadre du chainage d'un code de simulation et de traitements graphiques délégués à Love. Les traitements graphiques peuvent dorénavant s'effectuer pendant la phase de calcul d'un nouveau temps de simulation en optimisant les ressources de chacun de ces acteurs.

RÉFÉRENCES

- [1] Aquarius. <http://irfu.cea.fr/Projets/PYMSES/downloads.html>.
- [2] CEA's Tera supercomputer. <http://www-hpc.cea.fr/en/complexe/tera.htm>.
- [3] D. Aguilera, T. Carrard, G. C. de Verdière, J.-P. Nominé, and V. Tabourin. Parallel software and hardware for capability visualization of HPC results. *Numerical Modeling of Space Plasma Flows : Astronom*, 2007.
- [4] L. Avila, U. Ayachit, S. Barré, J. Baumes, F. Bertel, R. Blue, D. Cole, D. DeMarle, B. Geveci, W. Hoffman, B. King, K. Krishnan, C. Law, K. Martin, W. McLendon, P. Pébay, N. Russell, W. Schroeder, T. Shead, J. Shepherd, A. Wilson, and B. Wylie. *The VTK User's Guide*. Kitware, Inc., eleventh edition, 2010.
- [5] M. J. Berger and J. Olinger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, 53(3) :484–512, 1984.
- [6] O. Bressand, L. Colombet, A. Fontaine, G. Harel, and J.-B. Lekien. Hercule : A library of scientific data management for numerical simulation. *CHOCS*, 41 :29–37, 2012.
- [7] M. Gittings, R. Weaver, M. Clover, T. Betlach, N. Byrne, R. Coker, E. Dendy, R. Hueckstaedt, K. New, W. R. Oakes, D. Ranta, and R. Stefan. The RAGE radiation-hydrodynamic code. *Computational Science & Discovery*, 1(1), 2008.
- [8] G. Harel, J.-B. Lekien, and P. P. Pébay. Two new contributions to the visualization of AMR grids. In *eprint arXiv* ; Jan. 2017.
- [9] G. Harel, J.-B. Lekien, and P. P. Pébay. Visualization and analysis of large-scale, tree-based, adaptive mesh refinement simulations with arbitrary rectilinear geometry. In *eprint arXiv* ; Jan. 2017.
- [10] H. Jourden. HERA : A hydrodynamic AMR platform for multi-physics simulations. In *Adaptive Mesh Refinement Theory and Application*, volume 41 of LNCSE, pages 283–294. Springer, 2005.
- [11] M. Nguyen-Dinh, N. Lardjane, C. Duchenne, and O. Gainville. Direct simulations of outdoor blast wave propagation from source to receiver. *Shock Waves*, 2017. doi :10.1007/s00193-017-0711-2.
- [12] R. Teyssier. Cosmological hydrodynamics with adaptive mesh refinement. a new high resolution code called RAMSES. *Astronomy and Astrophysics*, 385 :337–364, 2002.

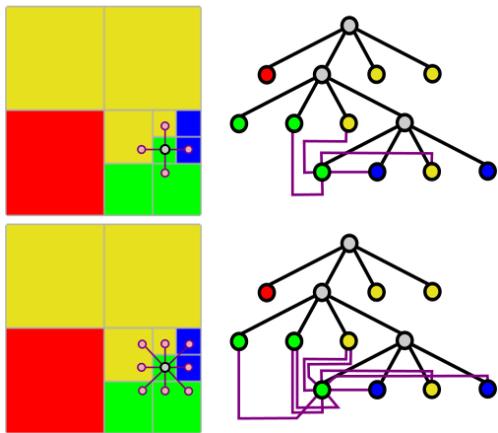


FIGURE 1. Description des supercursers de Von Neumann (haut) et de Moore (bas) d'une maille dans une grille AMR binaire de dimension 2 et de profondeur 3 (gauche); le même voisinage considéré du point de vue de la représentation en arbre (droite).

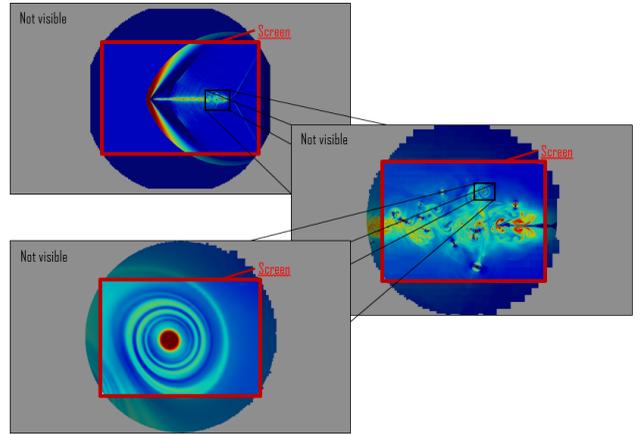


FIGURE 3. Rendu d'agrandissements successifs avec prise en compte du point de vue délimité par les mailles contenues dans un disque encadrant l'écran.

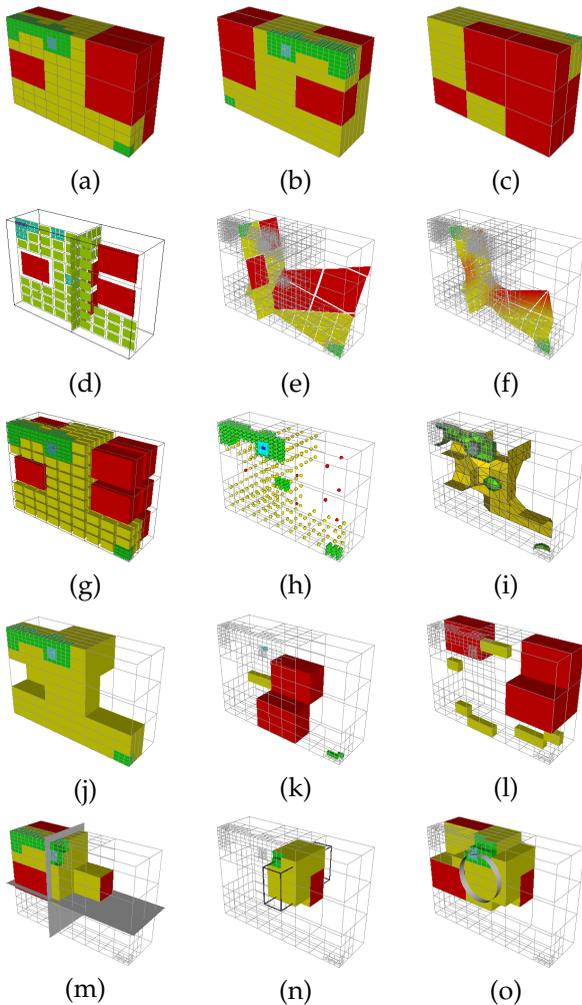


FIGURE 2. Visualisations obtenues en appliquant les filtres natifs *Geometry* (a), *AxisReflection* (b&c), respectivement par rapport à un ou deux plans parallèles aux axes, *AxisCut* (d), *PlaneCutter* (e&f), respectivement en mode primal ou dual, *ToUnstructured* (g), *CellCenters* (h), *Contour* (i), *Threshold* (j), *ExtractSelected{Ids,Locations}* (k&l), et *AxisClip* (m-o) à une grille *hypertree* de dimension 3.

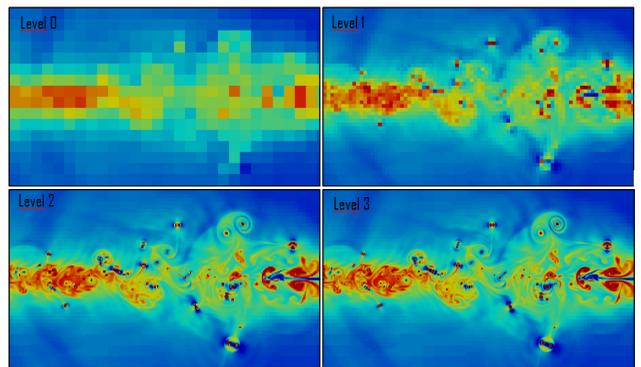


FIGURE 4. Différents rendus d'un même maillage AMR de dimension 2, avec prise en compte de mailles de niveaux de plus en plus élevés, illustrant l'effet de la décroissance du seuil de pixelisation.

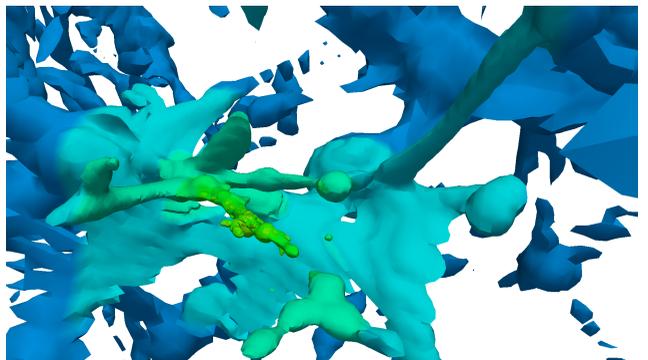


FIGURE 5. Vue rapprochée d'une partie d'une iso-surface générée par le filtre natif *Contour* appliqué à un maillage AMR 3D de grande taille.