Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 00 (2016) 000–000

Procedia Engineering

www.elsevier.com/locate/procedia

25th International Meshing Roundtable

Spatially accelerated shape embedding in multimaterial simulations

Kenneth Weiss[a,*], George Zagaris[a], Robert Rieben[a], Andrew Cook[a]

[a]*Lawrence Livermore National Laboratory, Livermore, CA, USA*

Abstract

Multimaterial numerical simulations typically embed the shape of their materials into the computational mesh by determining the *volume fractions* of each material within the mesh elements, which requires the means to numerically describe the material boundaries. In this paper, we present a mesh-agnostic technique for generating and querying an implicit function defining the *containment* field of a surface with complex geometric boundaries. Specifically, given a closed, oriented surface representing a material boundary, we construct an *In/Out octree* to accelerate *point containment* queries which can efficiently determine whether an arbitrary point in space is enclosed by the surface. We apply this technique to initialize material volume fractions in the elements of a multimaterial high-order finite element Arbitrary Lagrangian-Eulerian (ALE) hydrodynamics code.

Peer-review under responsibility of the organizing committee of IMR 25.

Keywords: spatial acceleration; octree; point containment; multimaterial; volume fractions; high order; finite elements; discontinuous Galerkin;

1. Introduction

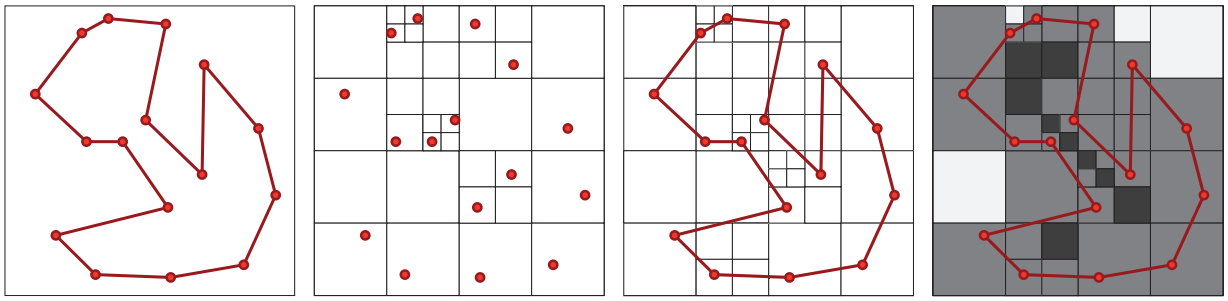
Advanced numerical methods modeling the interaction and/or mixing between different materials typically rely on numerical representations of embedded materials in the computational mesh. Examples include: the Volume of Fluids (VOF) method [1], the Immersed or Embedded Boundary technique [2,3], the Cartesian Cut Cell method [4] and the Interface-Enriched Generalized Finite Element method (IGFEM) [5]. The numerical representation describes the location and shape of the material boundaries, which are often not aligned with the elements of the computational mesh. These are typically encoded using per-material *volume fractions* describing the composition of each element.

While simple shapes, e.g., circles and spheres, can be described analytically using an implicit equation, constructing a numerical representation for arbitrary geometries with complex shapes, such as those designed in CAD software, can be challenging. Common approaches employ an auxiliary representation of the material boundary (i.e. the surface mesh) as an implicit function [6,7], which can be queried on demand.

The primary focus of this work is the efficient construction and querying of implicit volumetric representations for arbitrary and complex geometries. Specifically, given a material, Ω_i, defined by its material interface boundaries, $\partial\Omega_i$

* Corresponding author.
E-mail address: kweiss@llnl.gov

Peer-review under responsibility of the organizing committee of IMR 25.



(a) Input surface mesh $\partial\Omega_i$ (b) After inserting vertices (c) After inserting cells (d) After coloring leaves

Fig. 1. Generating an In/Out octree (d) over an input surface mesh (a) (shown in 2D).

and orientation according to outward-facing unit normals, we define an implicit *In/Out* function

$$\phi_i(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in \Omega_i \text{ (inside) or } \mathbf{x} \in \partial\Omega_i \text{ (on boundary)} \\ 0, & \text{if } \mathbf{x} \notin \Omega_i \text{ (outside)} \end{cases} \tag{1}$$

and utilize an octree as a spatial acceleration structure to efficiently represent ϕ_i. The In/Out octree supports point containment queries to determine whether a given point $\mathbf{x}$ is inside or outside $\partial\Omega_i$.

In the following section, we define the In/Out octree, describe how to generate and query the structure and provide some performance results. We then describe its application to initializing the material volume fractions of curvilinear elements in a multimaterial high order finite element Arbitrary Lagrangian-Eulerian (ALE) hydrodynamics code [8].

2. In/Out containment octree

We utilize a sparse octree data structure which recursively refines the embedding space of the surface $\partial\Omega_i$ into blocks that are entirely inside the surface, blocks that are entirely outside the surface and blocks that intersect the surface (see Figure 1). The latter index the cells of the mesh that they intersect and facilitate exact queries near the surface in a resolution-independent manner.

The octree's refinement is driven by the PM_2 criterion [9]. Namely, each leaf block of the octree can index at most one vertex of $\partial\Omega_i$, and all elements indexed by a block $\mathbf{b}$ must be incident in a common vertex $\mathbf{v}$, regardless of whether $\mathbf{v}$ is indexed by $\mathbf{b}$. Although this tends to generate deeper trees, it reduces the complexity of the geometry indexed within leaf blocks, yielding faster queries.

The In/Out octree is generated in several stages. We first compute the bounding box of the mesh to initialize the tree's spatial extents (Figure 1a). We then insert the vertices and cells of $\partial\Omega_i$ into the tree, refining as necessary to satisfy the PM_2 refinement criterion (Figures 1b and 1c). Finally, we 'color' the leaf blocks of the tree using a bottom-up hierarchical flood fill algorithm according to whether they intersect the surface (gray), or are entirely inside (black) or outside (white) $\partial\Omega_i$ (Figure 1d).

The structure of the tree admits a straightforward querying algorithm. Given a query point $\mathbf{x}$, we find the leaf block of the tree that contains $\mathbf{x}$, and use its 'color' to determine point containment. By convention, points outside the bounding box of the tree are considered outside $\partial\Omega_i$. Within gray blocks, we find the closest point on $\partial\Omega_i$ to $\mathbf{x}$. Since $\partial\Omega_i$ has outward facing normals, the point $\mathbf{x}$ is inside the surface when the dot product of its (pseudo)-normal [10] with the vector from the surface to the query point is negative.

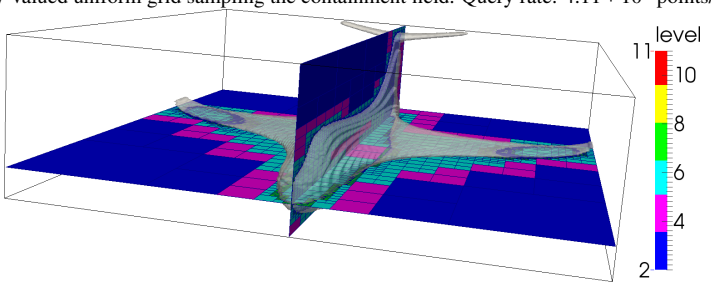
Due to our refinement criterion, gray blocks typically index only a few cells from the mesh, and queries can be performed very quickly. Table 1 lists query times and rates on a JET surface mesh at two resolutions: an (original) *high resolution* mesh composed of 373K triangular faces, and a simplified *low resolution* model with 30K faces (decimated via quadric edge collapse in Meshlab [11]). After generating the In/Out octree, we queried the containment field at the vertices of several uniform grids at increasing resolutions (from 17^3 to 257^3 samples).

Despite the order of magnitude difference in the sizes of the surfaces, we were able to query both surfaces at a sustained rate of several million queries per second (see columns 3 and 5 of Table 1). The timing difference was

Table 1. Query performance of the In/Out octree on the JET model at two resolutions. The containment field was sampled on uniform grids of increasing resolution from 17^3 to 257^3. Experiments were performed on a single core of a 2.8 GHz Intel Xeon processor with 64 GB RAM.

	JET (low resolution)		JET (high resolution)	
Sampling grid	Time (s)	Rate	Time (s)	Rate
17^3	$1.35 \cdot 10^{-3}$	$3.65 \cdot 10^6$	$3.47 \cdot 10^{-3}$	$1.41 \cdot 10^6$
33^3	$9.12 \cdot 10^{-3}$	$3.94 \cdot 10^6$	$1.65 \cdot 10^{-2}$	$2.18 \cdot 10^6$
65^3	$6.69 \cdot 10^{-2}$	$4.11 \cdot 10^6$	$9.54 \cdot 10^{-2}$	$2.88 \cdot 10^6$
129^3	0.49	$4.4 \cdot 10^6$	0.58	$3.69 \cdot 10^6$
257^3	3.32	$5.11 \cdot 10^6$	3.71	$4.58 \cdot 10^6$

Fig. 2. In/Out octree on low resolution JET model. Colors indicate leaf block levels along planar slices. Shown with contour (transparent) reconstructed from a 65^3 binary-valued uniform grid sampling the containment field. Query rate: $4.11 * 10^6$ points/sec.



mostly due to the increased octree depth for the high resolution model. Figure 2 illustrates the decomposition of an In/Out octree generated on the low resolution JET model.

Although the results in Table 1 sample the containment field along a regular grid, the real power of this representation is that the field can be sampled at arbitrary points in space, as we demonstrate in the following section.

3. Application to material shaping

In this section, we describe how the In/Out octree can be employed to initialize per-material volume fractions in the elements of an unstructured mesh within a multimaterial high order ALE hydrodynamics code.

We integrated the point containment query into LLNL's `Blast` hydrocode [12] which sets up the per-material volume fractions of its high order elements through a `volume_fraction` grid function in the discontinuous Galerkin (DG) basis [13]. The material volume fractions can be interpolated at any point in the computational mesh by means of DG basis expansion, where the overall volume fraction expansion coefficients of each element can be obtained by integrating over the element quadrature points and applying L2 projection [14].

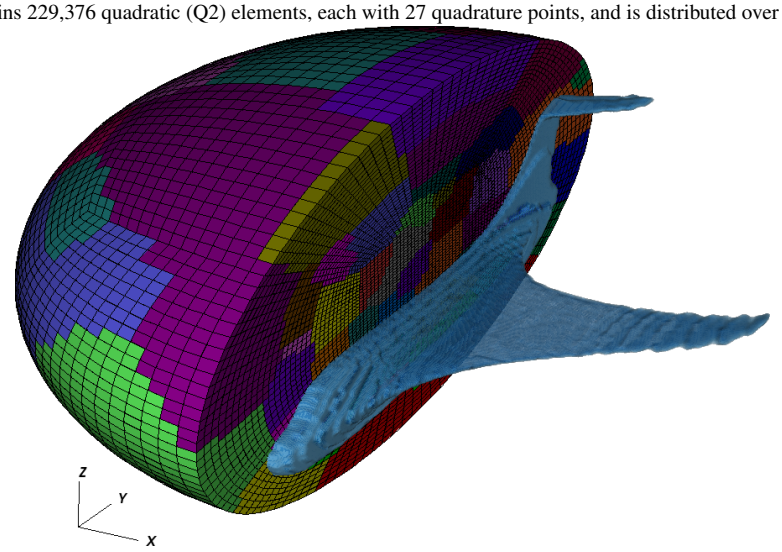
The code provides hooks for registering a callback function that returns 1 for points within the material and 0 otherwise. This setup has been effectively employed on combinations of simple analytic primitives (e.g. spheres, planes, cubes, ...), but it has been difficult to extend this setup to more complicated shapes. We utilized this callback mechanism to register an In/Out octree's point containment query for each desired material surface.

In a distributed computing context, we replicate the octree across each rank in the simulation and locally query the containment field at sample points associated with the degrees of freedom of each element in the subdomain. Figure 3 illustrates the `volume_fraction` grid function of the high resolution JET model embedded within a 128-rank curvilinear computational mesh with quadratic (Q2) elements.

4. Concluding remarks

We have utilized an In/Out octree as an implicit representation for the containment field of a surface and have applied this to initializing material volume fractions within a high order ALE hydrocode. In future work, we intend to

Fig. 3. Volume rendering of `volume_fraction` grid function for JET model embedded in an unstructured ALE hydrodynamics simulation. The computational mesh contains 229,376 quadratic (Q2) elements, each with 27 quadrature points, and is distributed over 128 subdomains.



extend the surface representation to more general element types, such as bilinear quads and NURBS patches. We are also exploring porting the generation and querying kernels to emerging HPC architectures, e.g., GPUs and Xeon Phi.

Acknowledgements

The authors would like to thank Rich Hornung, Brian Pudliner and Rob Neely for their keen interest and support in this work and Britton Olson for providing the JET model. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344.

References

[1] C. Hirt, B. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, Journal of Computational Physics 39 (1981) 201 – 225.
[2] C. Peskin, The immersed boundary method, Acta Numerica 11 (2002) 479–517.
[3] P. Colella, D. Graves, T. Ligocki, D. Trebotich, B. V. Straalen, Embedded boundary algorithms and software for partial differential equations, Journal of Physics: Conference Series 125 (2008) 012084.
[4] D. Ingram, D. Causon, C. Mingham, Developments in Cartesian cut cell methods, Mathematics and Computers in Simulation 61 (2003) 561 – 572.
[5] S. Soghrati, P. Geubelle, A 3D interface-enriched generalized finite element method for weakly discontinuous problems with complex internal geometries, Computer Methods in Applied Mechanics and Engineering 217220 (2012) 46 – 57.
[6] V. Mello, L. Velho, G. Taubin, Estimating the in/out function of a surface represented by points, in: Symposium on Solid Modeling and Applications, 2003, pp. 108–114.
[7] H.-J. Kim, T. Tautges, EBMesh: An embedded boundary meshing tool, in: Proceedings of the 19th International Meshing Roundtable, Springer, 2010, pp. 227–242.
[8] V. A. Dobrev, T. V. Kolev, R. N. Rieben, High-order curvilinear finite element methods for Lagrangian hydrodynamics, SIAM Journal on Scientific Computing 34 (2012) B606–B641.
[9] H. Samet, Foundations of Multidimensional and Metric Data Structures, The Morgan Kaufmann series in computer graphics and geometric modeling, Morgan Kaufmann, 2006.
[10] J. Baerentzen, H. Aanaes, Signed distance computation using the angle weighted pseudonormal, IEEE Transactions on Visualization and Computer Graphics 11 (2005) 243–253.
[11] Meshlab, Accessed August 2016. URL: `http://meshlab.sourceforge.net`.
[12] Blast: High-order finite element Lagrangian hydrocode, Accessed August 2016. URL: `http://www.llnl.gov/CASC/blast`.
[13] B. Cockburn, G. Karniadakis, C.-W. Shu, The development of discontinuous Galerkin methods, in: Discontinuous Galerkin Methods, Springer, 2000, pp. 3–50.
[14] MFEM: Modular finite element methods, Accessed August 2016. URL: `mfem.org`.