Journal of Computational Physics 241 (2013) 76–94

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Wall distance search algorithm using voxelized marching spheres

Beatrice Roget *, Jayanarayanan Sitaraman

University of Wyoming, Department of Mechanical Engineering, Laramie, WY 82071, United States

ARTICLE INFO

Article history:
Received 3 August 2012
Received in revised form 30 December 2012
Accepted 27 January 2013
Available online 9 February 2013

Keywords:
Numerical algorithms
Computational fluid dynamics
Turbulence modeling
Computer graphics
Minimum wall distance
Voxel-based methods

ABSTRACT

Minimum distance to a solid wall is a commonly used parameter in turbulence closure formulations associated with the Reynolds Averaged form of the Navier Stokes Equations (RANS). This paper presents a new approach to efficiently compute the minimum distance between a set of points and a surface. The method is based on sphere voxelization, and uses fast integer arithmetic algorithms from the field of computer graphics. Using a simple test case where the number of points (N_p) and surface elements (N_b) can be independently specified, the present method is empirically estimated to be $O(N_p^{0.8} N_b^{0.5})$. An unstructured grid around an aircraft configuration (DLR-F6) is chosen as the test case for demonstration and validation. Multi-processor computations (up to 256 processors) are conducted to study efficiency and scalability. Encouraging results are obtained, with the sphere voxelization algorithm demonstrated to be more efficient than all of the alternate methods for computing minimum distances. However, a load imbalance does exist, which negatively impacts the scalability for large number of cores. A simple method for load re-balancing is formulated and tested, which results in significant improvements in both efficiency and scalability.

1. Introduction

Minimum distance to a solid wall is a primary parameter that is utilized in several turbulence models. For example, in the commonly used Spalart–Allmaras turbulence model, the turbulence destruction source term is inversely proportional to the square of the wall distance. For a grid with N_p field points and N_b boundary faces, the cost of direct exhaustive computation of wall distances is of $O(N_p \times N_b)$, which is quite expensive for grids utilized in the state-of-the-art RANS based CFD calculations (where N_b is of order hundred thousand and N_p is of order several million). While the applicability of wall-distance based damping functions for geometrically and topologically complex problems is a topic of active debate in the turbulence modelling community, wall distance still remains a widely used parameter.

The primary motivation for this work stems from the need for efficiently re-computing minimum distance functions for unsteady simulations involving moving/deforming bodies. This is especially relevant in the case of massively parallel simulations involving overlapping grids. For example, the multi-mesh/multi-solver approach pioneered by the U.S Army HELIOS [1] framework uses a grid system that is composed of multiple body-conforming meshes (near-body meshes) that extend a short distance from the solid wall, and are embedded in a nested Cartesian mesh capable of adaptive mesh refinement (AMR). Clearly, the minimum distance function has to be evaluated accurately for all of the overlapping mesh systems to

* Corresponding author. Tel.: +1 3077662973.
E-mail addresses: broget@uwyo.edu (B. Roget), jsitaram@uwyo.edu (J. Sitaraman).

0021-9991/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2013.01.035

achieve consistent turbulence closure. Finally, in addition to scalability, efficiency and accuracy, software modularity is central for successful implementation and demonstration in such multi-solver frameworks.

The present work explores a new method of computing the minimum distance function efficiently and accurately, while maintaining software modularity. Several algorithms have been devised by various researchers in the past to compute the minimum wall distance. They can, in general, be classified into three groups:

1. k—d tree based search approaches [2–4]
2. Differential equation based approaches [5,6]
3. Advancing surface front methods [7]

The following section describes each method and highlights their respective limitations in a multi-grid, multi-solver, parallel environment.

2. Background

The k—d tree approach (octree, ADT, etc. are subsets of this group), which is an adaptation of the classic nearest-neighbor search, constructs a digital tree of the surface elements and follows a divide-and-conquer algorithm by eliminating regions that do not intersect with the sphere corresponding to the "current best" minimum distance. This approach is quite efficient for points that lie close to the boundary surface. However, as points are further away from the surface, lesser number of regions (and hence tree branches) can be eliminated from comprehensive checking, leading to poor scalability–because grid partitions lying further from the surface will incur much more computations compared to those that lie closer to the surface.

Differential equation based (Poisson, Eikonal) approaches are derived by posing the minimum distance search as a wave propagation problem, i.e. minimum distance is computed as the time taken by a wave front of unit speed emanating from the surface tesselation to reach a given query point. This is a purely hyperbolic problem that is observed to exhibit stability problems near sharp corners. Most often, dissipative terms are added to stabilize and smoothen the behavior of these equations. Once formulated, these equations can be discretized and solved using any of the standard approaches such as finite-difference, finite-volume or finite-element methods. Owing to discrete formulation, the differential equation based approaches are quite scalable and much easier to implement as parallel algorithms. However, the accuracy of the wall distances is driven by the order of accuracy of the discrete approximation (of the governing PDE) as well as the level of convergence obtained. True minimum distance can be achieved only in the limit of zero grid spacing and machine-zero convergence.

Advancing front methods utilize a painting technique using the nodal connectivity of cells. Beginning at the cells with wall boundaries, cell vertices are incrementally tagged by using surface element indices stored at nodal neighbors as candidates for evaluating minimum distance. This approach can be perceived as a topological equivalent of the wave propagation approach. Therefore, in the presence of non-uniform grids and highly varying cell sizes, the computed minimum distances could accumulate errors. Furthermore, advancing front methods have inherent defects in scalability as processes controlling partitions further away have to wait idle until the front approaches them after passing through closer partitions.

In a multi-grid, multi-solver context, differential equation based minimum distance approaches can be said to lack software modularity, since they require corresponding implementations in all participating solvers and grid assembly tools. Further, convergence of the distance function is not guaranteed in an overlapping mesh system, because of its strong non-linear dependence with complex geometry. Therefore, search-based methods that can exactly determine the distance function become attractive owing to their software modularity. Furthermore, advancing front methods and tree based methods to some extent rely on cell-to-node connectivity maps to accelerate their search process. This requirement adversely affects computing minimum distances in topologically disjoint overlapping mesh systems.

In the present study, we explore a new method in an attempt to obtain all of the qualities desirable in a multi-solver framework, namely scalability, accuracy, efficiency and modularity. In particular, to maintain modularity, the problem is formulated as requiring the minimum possible input information–only the surface tesselation (surface nodes and their connectivity) and a point cloud (points to which minimum distance need to be estimated) are necessary.

Similar to the tree approach, the proposed algorithm is based on searching rather than differential equations. The method attempts to eliminate as many surface elements as possible in order to compute exact distances for the smallest subset possible. It does so by considering discretized spheres on a structured auxiliary grid. These spheres are centered on each query point and their radius is increased until a subset of candidate faces is identified for exact minimum distance evaluation. This method does not however eliminate the load imbalance of the search process. For points that are a large distance away from the surface, the sphere radius will become large and hence the number of surface elements to consider will be larger than for points closer to the surface. However, the present study shows that the imbalance is not as large as for tree-based methods – since even at the limit of infinite distance only a subset of the faces needs to be checked– and hence can be mitigated by rebalancing the load in anticipation of this problem. Furthermore, similar to the tree method, the present method has guaranteed accuracy since it is based on exact computation of minimum distances (albeit for a small subset of surface elements).

It is worth noting that a naive implementation of the proposed method of voxelized marching spheres may lead to inadequate improvements in accuracy, efficiency, and scalability. In the present work, several optimizations are performed such