Journal of Computational Physics 322 (2016) 345–364

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Parallel level-set methods on adaptive tree-based grids

Mohammad Mirzadeh [a,*], Arthur Guittet [a], Carsten Burstedde [c], Frederic Gibou [a,b]

[a] Department of Mechanical Engineering, University of California, Santa Barbara, CA 93106-5070, United States
[b] Department of Computer Science, University of California, Santa Barbara, CA 93106-5110, United States
[c] Institute for Numerical Simulation and Hausdorff Center for Mathematics, University of Bonn, 53115 Bonn, Germany

ARTICLE INFO

Article history:
Received 28 June 2015
Received in revised form 18 February 2016
Accepted 10 June 2016
Available online 16 June 2016

Keywords:
Quadtree/Octree grids
Parallel computing
Semi-Lagrangian method
Level-set method
Adaptive mesh refinement
Space filling curves
Stefan problem

ABSTRACT

We present scalable algorithms for the level-set method on dynamic, adaptive Quadtree and Octree Cartesian grids. The algorithms are fully parallelized and implemented using the `MPI` standard and the open-source `p4est` library. We solve the level set equation with a semi-Lagrangian method which, similar to its serial implementation, is free of any time-step restrictions. This is achieved by introducing a scalable global interpolation scheme on adaptive tree-based grids. Moreover, we present a simple parallel reinitialization scheme using the pseudo-time transient formulation. Both parallel algorithms scale on the Stampede supercomputer, where we are currently using up to 4096 CPU cores, the limit of our current account. Finally, a relevant application of the algorithms is presented in modeling a crystallization phenomenon by solving a Stefan problem, illustrating a level of detail that would be impossible to achieve without a parallel adaptive strategy. We believe that the algorithms presented in this article will be of interest and useful to researchers working with the level-set framework and modeling multi-scale physics in general.

1. Introduction

The level-set method, originally proposed by Sethian and Osher [37], is a popular and powerful framework for tracking arbitrary interfaces that undergo complicated topological changes. As a result, the level-set method has been used to a wide range of applications such as multiphase flows, image segmentation, and computer graphics [35,46]. An important feature of this method is that the location of the interface is defined implicitly on an underlying grid. This convenience, however, comes at a price. First, compared to an explicit method, e.g. front tracking [28,54], the level-set method is typically less accurate and mass conservation could be a problem, although progress has been made in resolving this issue [19]. Second, the level-set function has to be defined in a one dimension higher space than that of the interface. If only the location of the interface is needed, the added dimension greatly increases the overall computational cost for uniform grids. One way to avoid this problem is by computing the level-set only close to the interface, e.g. as in the narrow-band level-set method [1] or, more recently, by using a hash table to restrict both computation and storage requirements [10].

Another approach that can address both problems is the use of local grid refinement. In [48] the idea of using tree-based grids for level-set calculations was first introduced and later extended in [38,31] for fluid simulations. More recently, authors in [33] proposed second-order accurate level-set methods on Quadtree (two spatial dimensions) and Octree (three spatial

* Corresponding author.
E-mail address: m.mirzadeh@engineering.ucsb.edu (M. Mirzadeh).

http://dx.doi.org/10.1016/j.jcp.2016.06.017
0021-9991/

dimensions) grids. The use of adaptive tree-base grids in the context of the level-set method is quite advantageous because (i) it gives fine-grain control over errors, which typically occur close to the interface and (ii) it can effectively reduce the dimensionality of the problem by focusing most of the grid cells close to the interface. Fortunately, constructing the tree is quite simple in the presence of an interface that naturally defines an ideal metric for refinement. However, even though the use of adaptive grids can dramatically reduce the computational cost, performing high-resolution three dimensional calculations of complex interfacial problems, e.g. crystal growth in binary alloys [51], could be prohibitively expensive or even impossible on a serial machine. In this paper we extend the level-set technology on Quad-/Octrees by proposing parallel algorithms for distributed memory machines using a domain decomposition technique.

One of the main challenges in parallelizing algorithms on adaptive grids is handling the grid itself. One option is to replicate the entire grid on each process and to employ serial ordering techniques, as implemented in an earlier version of the `deal.II` library [6], or to use serial graph partitioners such as METIS [29]. This approach, however, is only scalable to a few hundred processes at best and is limited by the size of the grid itself that can fit in memory. Even though parallel general-purpose partitioners have since been popularized [30,8] and the scalability of partitioning algorithms for unstructured grids has been improved (see e.g. [41]), their use adds extra overhead that can limit the overall scalability. Refining a grid consisting of multiple trees using recursive coordinate bisection has been implemented in the SIERRA framework [47]. Interestingly, tree-based grids have a nice spatial ordering that naturally leads to the concept of space-filling curves (SFCs) and can be efficiently exploited for parallel load balancing [4,21,14].

The idea of using SFCs for parallel partitioning of Quad-/Octrees is not new in itself and has been used by many researchers. For instance, `Octor` [55] uses a Morton curve (also known as Z-curve) for traversing the leaves of an Octree for indexing and load balancing and has been scaled up to 62 000 CPU cores [12]. `Dendro` [43] is an example of a so-called linear Octree code in which new algorithms are introduced for parallel partitioning and the development of a parallel geometric multigrid that has been scaled up to about 32 000 cores [44]. More recently, authors in [13] extended these ideas by optionally allowing for a collection, or a "forest", of connected Octrees, which is partitioned in parallel using a global Morton curve. The `p4est` library [11] provides a publicly available implementation of these algorithms that is equally efficient for a single tree as well as multiple trees and has been shown to scale to more than 458 k CPU cores [26]. Applications built with `p4est` have scaled to 1.57 M cores [40] and 3.14 M cores [34]. In fact, the algorithms presented in this paper are implemented on top of the `p4est` API. Due to the need for multiple adaptation and partitioning operations in each time step, the semi-Lagrangian method we describe below presents a stringent test of the algorithms and implementation both in terms of scalability and absolute run time.

Parallel level-set algorithms can be categorized into two groups: parallel advection algorithms and parallel reinitialization algorithms. Eulerian advection schemes can easily be parallelized but unfortunately are limited by the CFL condition, which could be very restrictive for adaptive grids. Semi-Lagrangian methods combine the unconditional stability of Lagrangian methods and the ease of use of Eulerian grids and have been successfully used for advecting the level-set function on tree-based grids [31–33]. However, parallelizing the semi-Lagrangian algorithm in a domain decomposition context is not an easy task. The reason for this is twofold. First, depending on the CFL number, the departure points may end up outside the ghost region and in remote processes that are potentially far away. This requires a very dynamic and nonuniform communication pattern that is challenging to implement. For an adaptive grid, the situation is even more complex due to the asymmetric nature of the communication pattern (cf. section 3). Second, load balancing could be an issue for large CFL numbers and nonuniform velocity fields, due to clustering of departure points, which can thus considerably restrict the scalability of the algorithm. Both of these problems, of course, could be avoided by choosing $\text{CFL} \leq 1$ but that would defeat the purpose of using the semi-Lagrangian algorithm in the first place.

Nonetheless, several parallel semi-Lagrangian algorithms have been proposed. A simple domain decomposition technique was used in [52] where the width of the ghost layer is fixed based on the maximum CFL number to ensure that all departure points are covered by the ghost layer. Good scalings were reported for small CFL numbers ($\text{CFL} \leq 2$). However, for large CFL numbers, this leads to a large volume of communication that can limit the scalability. In [18] the authors propose a more sophisticated domain decomposition approach which uses a "dynamic ghost layer". Here the width of the ghost layer is dynamically determined at runtime based on information from previous time steps. Unfortunately, this approach also suffers from excessive communication overhead for large numbers of processes. More recently, the authors in [58] used a domain decomposition strategy on a cubed sphere but with a single layer of ghost nodes. Interpolation on remote processes is then handled by sending query points to the corresponding process and asking for the interpolated result. This approach seems to provide good scalability for transporting a single tracer up to about 1000 cores for $\text{CFL} \sim 10$. At higher CFL numbers, the method begins to loose scalability due to an increase in communication volume. Finally, note that although we are mainly interested in parallel semi-Lagrangian methods, one could resort to finite difference or finite element discretization methods if small CFL numbers are acceptable. Indeed several algorithms of this type have been proposed with applications to modeling dendritic crystal growth [57], multiphase flows [50,20,39], and atomization process [23].

In many applications, it is desirable that the level-set function has the signed-distance property, i.e., $|\nabla\phi| = 1$. Generally, there are two approaches to enforce this property, either by solving the pseudo-time transient reinitialization equation [49, 36]

$$\phi_\tau + S(\phi_0)\,(|\nabla\phi| - 1) = 0,$$