Journal of Computational Physics 341 (2017) 447–473

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A novel partitioning method for block-structured adaptive meshes

Lin Fu, Sergej Litvinov [1], Xiangyu Y. Hu [*], Nikolaus A. Adams

Institute of Aerodynamics and Fluid Mechanics, Technische Universität München, 85748 Garching, Germany

ARTICLE INFO

Article history:
Received 15 December 2015
Received in revised form 26 September 2016
Accepted 3 November 2016
Available online 18 November 2016

Keywords:
Lagrangian particle method
Smoothed-particle hydrodynamics
Multi-resolution cell-linked list
Dynamic ghost particle method
Adaptive mesh refinement
Grid partitioning

ABSTRACT

We propose a novel partitioning method for block-structured adaptive meshes utilizing the meshless Lagrangian particle concept. With the observation that an optimum partitioning has high analogy to the relaxation of a multi-phase fluid to steady state, physically motivated model equations are developed to characterize the background mesh topology and are solved by multi-phase smoothed-particle hydrodynamics. In contrast to well established partitioning approaches, all optimization objectives are implicitly incorporated and achieved during the particle relaxation to stationary state. Distinct partitioning sub-domains are represented by colored particles and separated by a sharp interface with a surface tension model. In order to obtain the particle relaxation, special viscous and skin friction models, coupled with a tailored time integration algorithm are proposed. Numerical experiments show that the present method has several important properties: generation of approximately equal-sized partitions without dependence on the mesh-element type, optimized interface communication between distinct partitioning sub-domains, continuous domain decomposition which is physically localized and implicitly incremental. Therefore it is particularly suitable for load-balancing of high-performance CFD simulations.

1. Introduction

Massively parallel computing is essential for Computational Fluid Dynamics (CFD) to cope with flow simulations involving complex geometries or a wide spectrum of length scales [1]. To reduce computational costs, unstructured or adaptive structured mesh topologies are frequently employed as state-of-the-art mesh discretization methods [2]. Refined mesh elements are distributed at flow regions of interest or regions containing discontinuities. Such nonuniform mesh topologies are the consequence of the compromise between computational accuracy and efficiency [3]. However, partitioning problems arise with the utilization of such locally refined mesh topologies when the number of processor cores increases in massively parallel computing environments. The optimization strategy for load-balancing and inter-processor communication becomes the critical bottleneck and limits the computational performance. A partitioning method should accomplish approximately equal-sized domain decomposition with minimum neighbor communication patterns as it reduces processor operation idle time and inter-processor communication time. Moreover, each individual sub-domain is expected to be physically localized and continuous since it facilitates data management and reduces communications [4]. Partitioning operations for parallel computing can be categorized into static partitioning and dynamic partitioning. For the latter, additional requirements are

* Corresponding author.
E-mail addresses: lin.fu@tum.de (L. Fu), sergej.litvinov@aer.mw.tum.de (S. Litvinov), xiangyu.hu@tum.de (X.Y. Hu), nikolaus.adams@tum.de (N.A. Adams).
[1] Current address: Computational Science and Engineering Laboratory, ETH Zürich, Clausiusstrasse 33, CH-8092, Switzerland.

http://dx.doi.org/10.1016/j.jcp.2016.11.016
0021-9991/

that small topology changes should only result in slight modification of the partitioning (implicitly incremental) and the repartitioning time cost is minimized [5].

Classical partitioning methods can be roughly classified as geometry-based and graph-based approaches [4,6]. Several open-source codes, such as Metis [7], have been developed to meet these principles. The former is based on the physical coordinates of the original background mesh and generally leads to very fast algorithms. The Recursive Coordinate Bisection (RCB) method recursively divides the computational domain into equal-sized sub-domains by cutting planes that are orthogonal to the coordinate axes and has been proposed by Berger and Bokhari [8] to partition meshes generated by Adaptive Mesh Refinement [9]. RCB is an efficient method to provide a basic partitioning for structured or unstructured meshes when the connectivity information between mesh elements is not at hand. However, the partitioning quality suffers from possible discontinuous sub-domains and inefficient communication requirements [4]. The key idea of Space-Filling Curve (SFC) [10,11] partitioning is to construct a linear sequence of mesh elements through mapping the higher-dimensional mesh onto one dimension. Partitioning is obtained by cutting the one-dimensional sequence into desired pieces. The advantages of this approach are that it is fast, robust and geometrically localized. It also provides an efficient data structure as all elements can be assigned a globally unique index, enabling direct access. Furthermore, physical adjacency more or less is preserved. However, similarly to RCB, communication needs cannot be controlled explicitly during the partitioning, and disconnected sub-domains may exist [4]. Nivarti et al. [12] develop an inexpensive algorithm to improve the spatial locality of SFC partitioning by sacrificing a certain degree of load balance. Despite the moderate partitioning quality, geometry-based partitioning approaches play important roles in applications without inherent connectivity information, such as for particle based simulations.

In terms of graph-based approaches, a graph, where vertices represent the mesh elements to be partitioned and edges stand for communications between elements, is generated to characterize the background mesh. Load-balancing optimization and edge-cut minimization are achieved via Recursive Spectral Bisection (RSB) [13] or a multilevel graph partitioning strategy [14,15] Thus, the detailed connectivity information of the partitioning mesh must be available. RSB generates subgraphs by exploiting the eigenvectors of the Laplacian matrix associated with the target graph. It is quite expensive due to the eigenvector calculation. The multilevel partitioning paradigm is much more efficient and mainly consists of three substages: (i) graph coarsening which generates a sequence of coarser graphs preserving the essential properties of the original graph; (ii) partitioning of the representative coarsest graph; and (iii) graph refinement (Kernighan–Lin [16] heuristics and its variant Fiduccia–Mattheyses algorithm [17] are widely employed) which improves the partition by interpolating the coarser partitioned graph back to the original one in combination with a fast local search algorithm [18]. These approaches have significant advantages over geometry-based approaches by generating high-quality domain decomposition with controllable interface communication, although they are more time consuming. However, classical graph-based approaches may suffer from the problem that the optimization objective function is not always adequate. The partition objective of minimizing the total communication message size is not directly equivalent to minimizing edge-cuts because edge-cuts are in general not proportional to the communication volume [19]. For many applications, the number of graph vertices located at the boundaries of partitioning sub-domains (boundary vertices) indeed characterizes the real communication cost much more accurately than the number of edge-cuts [20]. Boundary properties of partitioning sub-domain, e.g. aspect ratio [21], connectedness and smoothness [22], are also crucial in communication reduction [23]. This problem can be improved largely by introducing the Hypergraph partitioning concept, which encodes the communication volume better, with higher running time [24]. With respect to dynamic load-balancing which introduces additional optimization objectives, the problem arises in parallelizing the widely-used Kernighan–Lin [16] method on distributed-memory architecture and designing the multiple optimization metrics [25]. In such a scenario, graph-based diffusive partitioners [26,20] are generally applied as they produce partitions, that have more implicit incrementality and higher-quality than those from classical multilevel partitioning strategies. A drawback of these graph-based partitioners is that strictly-connected partitions cannot be guaranteed in principle [20]. Nevertheless, graph-based partitioning methods are widely accepted in well-established software [7].

We will explain below, that the partitioning result has high similarity to a stationary multi-phase fluid. Based on this observation, we propose a new partitioning method based on multi-phase particle simulation. Using the topology information from a background mesh, a set of physically motivated model equations is formulated and solved by a meshless multiphase Smoothed Particle Hydrodynamics method [27]. While most graph-based partitioning approaches explicitly optimize the communication by minimizing the edge-cuts, we seek to minimize the number of boundary vertices and optimize the partitioning-sub-domain shapes. All optimization objectives are implicitly incorporated and optimized during the particle evolution process. Our proposed partitioning method guarantees connected partitioning, optimized interface communication, good load-balancing and possesses the implicitly incremental property. Moreover, it does not depend on the specific mesh element type and can be straightforwardly extended to unstructured mesh partitioning.

2. Physically motivated models

2.1. Mathematical concepts

For parallel simulations, the load-balancing requirement can be formulated as a problem in graph theory. Considering an undirected graph $G = (V, E)$, where V denotes a set of vertices corresponding to the computation units and E denotes a