Advances in Engineering Software 121 (2018) 131–146

Contents lists available at ScienceDirect

Advances in Engineering Software

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

Scalable generation of large-scale unstructured meshes by a novel domain decomposition approach

Jianjun Chen[a,b], Zhoufang Xiao[c,*], Yao Zheng[a,b], Jianfeng Zou[a,b,*], Dawei Zhao[a,b], Yufeng Yao[d]

[a] Center for Engineering and Scientific Computation, Zhejiang University, Hangzhou 310027, China
[b] School of Aeronautics and Astronautics, Zhejiang University, Hangzhou 310027, China
[c] School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China
[d] Faculty of Environment and Technology, University of the West of England, Bristol BS16 1QY, United Kingdom

ARTICLE INFO

Keywords:
Mesh generation
Domain decomposition
Parallel algorithms
Dual graph
Large-scale meshes

ABSTRACT

A parallel algorithm is proposed for scalable generation of large-scale tetrahedral meshes. The key innovation is the use of a mesh-simplification based domain decomposition approach. This approach works on a background mesh with both its surface and its interior elements much larger than the final elements desired, and decomposes the domain into subdomains containing no undesirable geometric features in the inter-domain interfaces. In this way, the most time-consuming part of domain decomposition can be efficiently parallelized, and other sequential parts consume reasonably limited computing time since they treat a very coarse background mesh. Meanwhile, the subsequent parallel procedures of mesh generation and improvement are most efficient because they can treat individual subdomains without compromising element quality. Compared with published state-of-the-art parallel algorithms, the developed parallel algorithm can reduce the clock time required by the creation of one billion elements on 512 computer cores from roughly half an hour to less than 4 minutes.

1. Introduction

Various simulation codes have been parallelized to exploit their potentials of running efficiently on increasingly powerful parallel computing machines. However, the clock time to finish a complex simulation is much longer than that expected by the end-user. A significant percentage of this time is consumed by the mesh generation step. For instance, it takes weeks or more to prepare a block-structured mesh for an external flow simulation over a complete aircraft model, even by an engineer with expertise in applying state-of-the-art meshing tools [1]. Unstructured mesh generation does not require a painful block-decomposition process; hence, it is more straightforward and automatic. Nevertheless, the process of generating an unstructured mesh in excess of 10^9 elements still consumes about 1.5 days if being executed sequentially [2]. In practice, when the input geometry is becoming very complicated, or the resultant mesh quality is set at a high standard, mesh generation is often becoming a trial-and-error process. Its execution possibly needs to be repeated several times before an ideal mesh is obtained. Therefore, the clock time for preparing a large-scale unstructured mesh is usually comparable with or more than the time consumed for conducting parallel simulations. To fully exploit the computing power of the present and future emerging parallel computers, an environment where both the simulation and its mesh generation codes are parallelized is now highly demanded [2-7] by research developers as well as industry end-users.

Various parallel mesh generation approaches have been developed since 1990s. In general, these approaches could be classified into the algorithm-parallel ones and the problem-parallel ones [8]. Presently, the problem-parallel approaches are preferred in many studies because of their better capability of reusing state-of-the-art sequential mesh generation codes. Domain decomposition is an essential step of the problem-parallel approaches. It subdivides a problem domain into many subdomains such that the following meshing procedure could be conducted on these subdomains in parallel. The performance of a parallel mesh generation algorithm is thus highly dependent on its domain decomposition approach.

Among various performance indices, the scalability of a parallel mesher should be highlighted nowadays since the computers used in present simulations may be configured with hundreds of thousands of computer cores. It is not uncommon that the required computational mesh by such simulations may contain billions of elements or more. To reduce the meshing time into the order of minutes, the parallel mesher must demonstrate its scalability on several hundred or more computer cores. However, in most existing parallel mesh generation studies, the

* Corresponding authors.
E-mail addresses: xiaozf@hdu.edu.cn (Z. Xiao), zoujianfeng@zju.edu.cn (J. Zou).

https://doi.org/10.1016/j.advengsoft.2018.04.005
Received 24 November 2016; Received in revised form 6 April 2018; Accepted 9 April 2018

principal motivation was to overcome the memory bottleneck of large-scale mesh generation, and the scalability is not a major concern. Consequently, the achievable scalability of existing parallel meshers is not always desirable. A typical example is the parallel mesher we developed a few years ago [9]. Its domain decomposition approach inputs a surface mesh of the problem domain, and decomposes this mesh by inserting interface meshes in the interior of the domain. After resolving the intersections in subdomain boundaries properly, the reliability of this approach has been well demonstrated for complicated configurations. Nevertheless, the efficient parallelization of this domain decomposition approach remains a challenging issue. No matter how many additional computer cores are invested, the time cost of the domain decomposition step scales up in accordance with the increase of the required computational mesh size. As a result, a poor scalability is observed for this parallel mesher when hundreds of computers cores are invested to create large-scale meshes. It must be emphasized that this poor scalability is not only observed for the early parallel mesher developed by us [9]. For one reason or another, the scalability issue is not well resolved in many other existing parallel meshers. For instance, in one test of the parallel advancing front technique (AFT) proposed in [2], 121 million elements could be created on 8 computer cores at a relative speed (els/sec/core) of 9,423. However, it was reported that this speed value was reduced by more than one order (being 788) when 512 computer cores are used to create one billion elements.

In this study, a novel domain decomposition approach is proposed to achieve the scalable generation of large-scale tetrahedral meshes. It inputs a problem domain depicted by a CAD model, and then decomposes the domain into the required number of subdomains depicted by the same fine surface triangulation as the final mesh desired. To be clear, Fig. 1 presents an example illustrating the main steps involved in the proposed approach.

(1) With the input sizing function scaled up, a coarse surface background mesh (SBM) is created by meshing the CAD surfaces. After that, a volume background mesh (VBM) is created with the SBM and the scaled-up sizing function as inputs (see Fig. 1a).
(2) Both the SBM and the VBM are simplified and their simplifications are decomposed into *subsurfaces* and *subdomains*, respectively. Here, a subsurface refers to the set of surface faces bounding a subdomain. By adopting the mesh simplification approach developed in [10], not only badly shaped faces and small dihedral angles are prevented from appearing on inter-domain interfaces, but also those small angles are prevented from appearing on inter-surface interfaces (see Figs. 1b and 1c).
(3) Subdomain boundaries are remeshed according to the input sizing function. A 2D Delaunay mesher is used to triangulate the inter-domain faces individually, while an advancing front surface mesher is reused to mesh subsurfaces to ensure the generation of surface mesh fitting into the original CAD model.

This novel domain decomposition approach has enabled us to build up an efficient and scalable parallel pre-processing pipeline. This pipeline inputs a CAD model and then executes the subsequent steps of surface meshing, volume meshing and mesh improvement in a fully parallel and automatic manner. Numerical experiments will be presented to demonstrate that the developed approach is robust and applicable to geometry models of a complicated level experienced in industry. Further scalable parallel performance will be revealed by detailed comparisons with other state-of-the-arts approaches reported in the public domain.

The remaining sections are organized as follows. In Section 2, the typical problem-parallel mesh generation approaches are briefly reviewed and compared, followed by a summary of our contributions. Sections 3 and 4 present the techniques adopted in the domain decomposition step and the parallel procedures of mesh generation and quality improvement, respectively. Section 5 provides various numerical examples demonstrating the effectiveness and the efficiency of proposed approach. Finally, Section 6 concludes the study.

2. Related work

2.1. Literature review

Depending on how the inter-domain interfaces are treated in parallel mesh generation, the problem-parallel mesh generation approaches are thus summarized [11] as: (1) those meshing interfaces as they mesh subdomains [12], (2) those postmeshing the interfaces, and (3) those premeshing the interfaces.

Chrisochoides and Nave proposed a typical approach that meshes interfaces as it meshes subdomains [12]. Delaunay refinement is executed on each submesh and a remote data gathering is required to enforce the mesh conformity if the formed cavity during the insertion of a new point crosses inter-domain interfaces. Here, the cavity refers to all elements whose circumcircles include the new point. Fig. 2a illustrates a cavity in relation with the insertion of a new point P belonging to the submesh M_0, in which the cavity triangles ABC and BDC belong to M_0, but the cavity triangles DEC and EFC belong to the submeshes distributed on different processes (M_1 and M_2, respectively). Thus, the time-consuming inter-process gathering operations must be executed to insert P, which could degrade the scalability of the parallel point insertion algorithm substantially.

The second approach, which meshes interfaces after it meshes subdomains, follows totally different workflow with the first approach. Fig. 2b shows the interface after meshing a simple rectangular domain by using 4 processes. Note that buffer zones are set up between adjacent subdomains, in this case, around 4 interface lines and 1 interface point. Evidently, it remains a challenging task to robustly and efficiently treat these buffer zones.

Cougny and Shephard [13] adopted the second approach to parallelize their octree-based tetrahedral mesher. It meshes most interior octants in parallel and defines a cavity zone on each process by combing unmeshed octants. Consequently, different buffer zones are set up between adjacent cavities, around inter-domain faces, lines and points, respectively. The cavity zones are filled up to an order of meshing non-buffer zones at first, and then meshing face-related, line-related and point-related buffer zones successively. Desirable scalability performance was reported when 32 processes were executed [13]; however, it is not clear how this algorithm would perform when more computing resources are invested.

Löhner [2,14] presented two parallel advancing front techniques (AFT) in which the meshing stage of subdomain interiors takes the precedence to that for interfaces. The first technique [14] distributes the workloads by using an octree background mesh. Many buffer zones are formed near inter-domain interfaces by prohibiting new elements starting from active fronts crossing over interfaces, and these zones will be meshed after shifting the octants to cover them. The second technique [2] distributes workloads by using a domain defining grid (DDG), which has the same fine surface triangulation as the final mesh desired, but a much coarser interior mesh. The scalability of the early algorithm is limited since its applied parallelism is at a local level of the active front. By comparison, the scalability of the new algorithm is much higher by applying the volume-level parallelism, although further improvement is possible. As reported in [2], the relative speed value of the new parallel AFT at a case of creating one billion elements on 512 computer cores is smaller by more than one order than its counterpart at a case of creating 121 million elements on 8 computer cores.

With respect to the third parallel mesh generation approach, the meshing procedure of each subdomain is completely decoupled; thus, the subdomain mesh generation step could achieve very high parallel efficiencies. However, the efficient parallelization of domain decomposition step may become very difficult. For instance, the algorithms proposed in [9,15-16] adopts a recursive bi-division scheme to