Computers and Structures 257 (2021) 106649

Contents lists available at ScienceDirect

Computers and Structures

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

A load balancing algorithm for the parallel automated multilevel substructuring method

Cheolgyu Hyun, Phill-Seung Lee *

Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea

ARTICLE INFO

Article history:
Received 4 December 2020
Accepted 28 July 2021
Available online 8 September 2021

Keywords:
Structural dynamics
Eigenvalue problem
Finite element method
Component mode synthesis
AMLS method
Parallel algorithm

ABSTRACT

The objective of this paper is to present a load balancing algorithm for the parallel automated multilevel substructuring (PAMLS) method. In the PAMLS method, load balancing is highly dependent on the computation time for the transformation and back transformation procedures corresponding to substructures. To balance the workload among threads, the proposed algorithm consists of two types of granularity: coarse-grained and fine-grained parallel algorithms. According to the level of substructures, the coarse-grained parallel algorithm splits both the transformation and back transformation procedures and assigns them to threads. Through fine-grained parallelism, more threads are exploited for the transformation of each substructure compared to threads used in the original PAMLS method. Without repartitioning, the proposed algorithm significantly improves the efficiency of the PAMLS method.

1. Introduction

Component mode synthesis (CMS) methods [1–6] have been widely used to solve frequencies and mode shapes of large and complex structures [7–16]. In conventional CMS methods, a structural finite element (FE) model is divided into small substructures with an interface among them, and each substructure is individually reduced via truncation of its eigensolutions. Since this approach is suitable for parallel implementation that carries out many calculations simultaneously, there have been various studies to parallelize CMS methods [17–20].

Among CMS methods, the automated multilevel substructuring (AMLS) [21–24] is a successful method to reduce the large interface DOFs efficiently and robustly. The AMLS method partitions an FE model into many levels of substructures by using the nested dissection algorithm [25]. Then the FE model is transformed by synthesizing substructure eigensolutions and constraint modes. The approximate solutions are computed through the back transformation of solutions obtained by the reduced model. The AMLS method can serve as an alternative to the Lanczos method [26,27] when computing a large number of eigensolutions [28,29] and has been applied to various engineering problems [30–32].

After Kaplan [22] proposed the directions in parallelism for the AMLS method, Elssel and Voss [33] developed the parallel AMLS (PAMLS) method. Each thread first performs the AMLS transformation of a subtree consisting of substructures. The rest of the substructures (i.e. partitioned interfaces) are then transformed in parallel. Yang et al. [34] applied a multilevel approach to nonlinear implicit dynamics, where the maximum allowed imbalance among substructures is set by a static load balancer [35,36]. Both methods [33,34] successfully parallelize the computation associated with substructures, including interfaces.

The objective of this paper is to present a novel load balancing algorithm that improves the parallel efficiency of the PAMLS method. Since the transformation and back transformation procedures spend most of the computation time in the AMLS method, we focus on improving the parallel efficiency of those procedures. In general, a load imbalance among the transformations of substructures would occur even if the number of substructures assigned to each thread is the same and the number of their degrees of freedom (DOFs) is well balanced. As a result, there is an idle time at the synchronization point before the transformation of a parent substructure. To reduce this idle time, the proposed algorithm consists of two types of granularity: coarse-grained parallelism and fine-grained parallelism.

In coarse-grained parallelism, the transformation and back transformation procedures are split into tasks corresponding to the predefined subtrees and the rest of the substructures. To balance the computational load in each task, we employ the approach of Escaig et al., in which the number of assigned tasks (i.e. substructures) is set to be larger than the number of threads [37]. We determine the number of subtrees by using a given cutoff level instead of the number of threads. Moreover, for the back

* Corresponding author.
E-mail address: phillseung@kaist.edu (P.-S. Lee).

https://doi.org/10.1016/j.compstruc.2021.106649
0045-7949/