Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 196 (2007) 2084–2096
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

An iterative parallel workload balancing framework for direct condensation of substructures

Ozgur Kurc, Kenneth Will *

Computer-Aided Structural Engineering Center, School of Civil and Environmental Engineering, Georgia Institute of Technology, Atlanta, GA, USA

Received 14 September 2005; accepted 6 July 2006

Abstract

This study presents a workload balancing framework which diffuses the condensation time imbalances of substructures in a homogeneous computing environment. The structure is initially partitioned in such a way that the number of substructures is equal to the number of processors. Then, the estimated condensation time imbalance of the initial substructures is adjusted by iteratively transferring nodes from the substructures with slower estimated condensation times to the substructures with faster estimated condensation times. Examples which illustrate the applicability and efficiency of this framework are presented. In these examples, the effect of utilizing different repartitioning and equation numbering algorithms are investigated.

Keywords: Substructuring; Parallel solution; Direct condensation; Workload balancing; Partitioning; Repartitioning

1. Introduction

Substructure based linear static analysis solution approaches have again become a popular solution method as parallel computation is being implemented in finite element codes [1,9]. In the substructure based solution approach, the structure is first divided into substructures. Then, the substructure level stiffness matrices and force vectors are assembled and the internal equations are transferred to the substructure interfaces using condensation algorithms. As a result, the solution is converted into the solution of dense interface equations. After having computed the interface displacements, each substructure's internal displacements are computed. In the parallel implementation of this approach, the substructures are distributed to processors and the condensations and the solution of interface equations are performed in parallel. The main advantage of this approach is that the preparation of the stiffness matrix and force vectors, substructures' condensation, and element result computations can be performed in parallel without any communication between the processors.

Direct solvers are often used to condense the substructures' stiffness matrices in substructure based solution approaches [3,18] since they are robust and very suitable for problems having multiple loading conditions. On the other hand, any imbalance in the condensation times of the substructures reduces the efficiency of the parallel solution since the interface solution cannot begin until all processors finish the condensation. Hence, the partitioning of the structure into substructures is crucial for such methods.

Various partitioning approaches exist in the literature [8]. The basic goal of many of the partitioning approaches is to minimize the communication between processors while maintaining balanced computational loads in each partition. The computational loads of processors can be balanced if the computational cost can be represented by a single weight value assigned to a node or an element. However, when a direct condensation method is used, such weight definitions are insufficient to provide a balanced distribution of the computational load [7]. Other variables

* Corresponding author.
E-mail address: kenneth.will@ce.gatech.edu (K. Will).

0045-7825/$ - see front matter
doi:10.1016/j.cma.2006.07.015

affect the condensation time such as the equation numbering, the profile of the stiffness matrix, and the number of the internal and interface equations. Moreover, the variables which affect the condensation time depend on the way in which the structure is partitioned. In other words, the computational load of each substructure can only be estimated after partitioning. Secondly, the profile of the stiffness matrix depends on how the equations are numbered. If an active column storage scheme is used, the equation numbering is generally performed by using a profile minimization algorithm. Such algorithms are based on heuristic approaches. Therefore, it is very difficult to predict the effect of any partition changes on the equation numbering and hence on the condensation time. As a result, it is also difficult to partition a structure while balancing the workload for direct condensation.

In order to overcome such difficulties, Fulton and Su [4] developed a dynamic processor assignment strategy. Once the structure was partitioned, they estimated the computational loads for condensing each substructure's stiffness matrix and assigned more processors to the substructures which were estimated to require more computation. Escaig et al. [2] partitioned the structure in such a way that the number of substructures was larger than the number of processors. During the parallel solution, the substructures were condensed by the first available processor. This way, the processors stayed less idle during condensations but the size of the interface problem increased as the number of substructures increased. Both Fulton and Su's [4] and Escaig et al.'s [2] approaches were designed for shared memory computers.

For distributed memory computers, Pinar and Hendrickson [15] proposed a general partitioning framework for various applications where existing graph partitioning techniques are unsuitable. The method was based on first performing an initial partitioning and then exchanging vertices among neighbor partitions until the imbalance among the partitions, which is computed by a cost function, was minimized. The framework was then tested on two applications – partitioning so that the overlapped subdomains were balanced and partitioning to minimize the sum of the computation and communication costs. For the test problems solved, not only the imbalance among partitions decreased but also the solution performance was improved. A similar procedure was utilized by Yang and Hsieh [20] where they attempted to iteratively balance the condensation times of substructures. Each substructure's workload was estimated by computing the operation count for sparse matrix condensation at each iteration. Each element within a given substructure was assigned a weight factor based on the substructure's workload. Then, either the structure was repartitioned or the substructures were modified by element migration. The iterations continued until a desired balance was obtained. For the test problems considered by Yang and Hsieh [20], this approach provided more balanced substructures and decreased the condensation time. On the other hand, the time consumed during the iterations was so high that the method was suitable only for nonlinear dynamic analysis and not for a linear static analysis.

This study focuses on developing a fast and efficient workload balancing framework for direct condensation that can be used prior to a parallel substructure based solution algorithm [13] which is designed for solving linear systems having multiple loading conditions on commonly available PC clusters. As a result, the parallel solution algorithm utilizes direct solvers where the condensations and interface equation solutions are performed by an active column and a parallel variable band solver, respectively. For large size problems that exceed the in-core memory capacities of the computers, the out-of-core version of the active column solver [19] is utilized. Although the presented method attempts to balance the condensation times of active column matrices, it can also be utilized for other solution approaches such as the local solution in the FETI method [3] or sparse matrix condensations [20].

The presented workload balancing framework iteratively searches for more balanced substructures by modifying them according to their estimated condensation time ratios. Nodal graphs are utilized in order to work with models composed of both frame and shell elements. Moreover, the framework has a parallel structure, in other words, all the computations during the iterations are performed in parallel. Thus, the time consumed during the workload balancing step is decreased and the proposed method becomes more suitable for linear problems.

2. Method

The workload balancing framework is designed as an independent program that prepares input data for a parallel substructure based solver. The target parallel environment is PC cluster systems connected using either a network hub or switch so that many structural design offices can utilize their existing hardware to perform parallel computation. Currently, only homogeneous PC clusters, a cluster composed of identical PC's, was considered. The MPICH [14] message passing library was used for parallelism. The program was developed with C++ and FORTRAN programming languages and implemented under the Windows operating system.

2.1. Workload balancing algorithm

The flow chart of the algorithm is presented in Fig. 1. The solution begins by converting the structural information into the nodal graph representation. The nodal graph is then partitioned into 'n' parts by using METIS [12] where 'n' is equal to the number of available processors. After assigning a single substructure to each processor, the nodal graph and the initial partitioning information are distributed to every processor. The processors first extract their assigned substructure's subgraphs from the nodal graph using the partitioning information. A subgraph is actually the nodal graph of a substructure with