 ICCES, vol.13, no.3, pp.55-56

A Framework for Parallel Adaptive FEM Computations with Dynamic Load Balancing

Z. Bittnar[1], B. Patzák[1], D. Rypl[1]

This paper deals with the design of framework for adaptive FEM analysis with dynamic load balancing in nondedicated parallel cluster computing environments. It describes in detail the structure and design of individual components of the framework.

The application of adaptivity paradigm to engineering problems results in computationally very demanding analysis in terms of both computational time and computer resources (memory, disk space, etc.). These demands can be alleviated by performing the analysis in a parallel computing environment. Typical parallel application decreases the demands on memory and other resources by spreading the task over several mutually interconnected computers and speeds up the response of the application by distributing the computation to individual processors.

The adopted parallelization approach is based on domain decomposition, which requires a partitioning of the problem into a set of subdomains, the number of which is equal to or greater than the number of available processors. The partitioning of the problem can be fixed (static load balancing) or can change during solution (dynamic load balancing). The latter option is often necessary in order to achieve good load balancing between processors resulting in optimal scalability.

The paper describes in detail structure and design of parallel adaptive load balanced algorithm. The typical solution step sequence consists of several stages. In the first stage, once the solution is obtained, the error of the achieved solution is evaluated. When its limit is reached, the adaptive remeshing is performed in parallel. Then, the solution state has to be transferred from the old to the new discretization. This includes solution vectors as well as internal history variables at individual integration points. Moreover, in nonlinear problems, consistency recovery is performed by restarting equilibrium iterations. The second stage is responsible for the load balance recovery, which is achieved by repartitioning the problem domain and by transferring the work (represented by finite elements) from one subdomain to another. The application has to continuously monitor the solution process and detect work imbalance. When imbalance is detected, the decision has to be made whether to recover load balance or continue with existing work distribution, depending on the magnitude of load imbalance and the cost of load recovery. Work transfer requires serialization of problem data into a byte stream that is sent over the network and unpacked, followed by topology update to reflect new partitioning.

[1]Department of Mechanics, Faculty of Civil Engineering, Czech Technical University in Prague,Czech Republic

The capabilities and performance of developed framework are demonstrated on a 2D nonlinear fracture analysis of brazilian splitting test. The results show improvements compared to parallel adaptive simulations without load balancing. Future work will focus on application to large-scale 3D adaptive simulations on inhomogeneous nondedicated computer clusters.