GPU-based Parallel implementation of Explicit Meshfree methods

Yong Cai, Xiangyang Cui, Guangyao Li*, Wenyang Liu

State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body,
Hunan University, Changsha, 410082, China
*Corresponding author: gyli@hnu.edu.cn

Abstract:

In the field of numerical analysis, meshfree methods are those that do on requrie conneciton between nodes of the simulaiton domain, but are rather based on interaction of each node with all its neighbours. Meshfree methods enable the simulation of some otherwise difficult types of problems, at the cost of extra computing time and programmig effort. However, because of the limitation of time step due to the conditional stability, when the meshfree method is used for explicit solution, the analyses of large-scale problem always require a long computing time. Graphics processing unit (GPU) is a parallel device with single instruction, multiple data classification. GPU offers high computation power and increased memory bandwidth at a relatively low cost, and it is well suited for problems that can be expressed as data-parallel computations with high arithmetic intensity. Nowadays, it has gained more and more attention as a kind of general parallel processor, followed by various general purpose GPU computing technologies represented by NVIDIA CUDA. In this paper, meshfree parallel implementations using GPU are discussed. Particulary, parallelization of EST and a novel triangular element without cube degree of freedom are shown. The appropriate strategies to correctly parallelize the mesh free methods are discussed. Finally, GPU is used for whole explicit iterative process by mapping one element or one node to one CUDA thread, and parallel solved in any order. The numerical examples indicate that this method can greatly improve the computational efficiency with the same computing precision on the ordinary commercial GPU when compared to the sequential one.

Key words: Smooth finite element; GPU computing; Parallel computing; Nonlinear analysis; Contact-impact simulation