21st International Conference **ENGINEERING MECHANICS 2015**
Svratka, Czech Republic, May 11 –14, 2015
Paper **#055, pp. 254–255**

Parallel Computing Procedure for Dynamic Relaxation Method on GPU Using NVIDIA's CUDA

Václav Rek[a], Ivan Němec[b]

Institute of Structural Mechanics, Brno University of Technology, Veveří 95; 602 00, Brno, CZ

[a]rekv@seznam.cz, [b]nemec@fem.cz

Keywords: finite element method, dynamic relaxation, finite difference method, central difference method, parallel computing, GPU, GPGU, NVIDIA CUDA

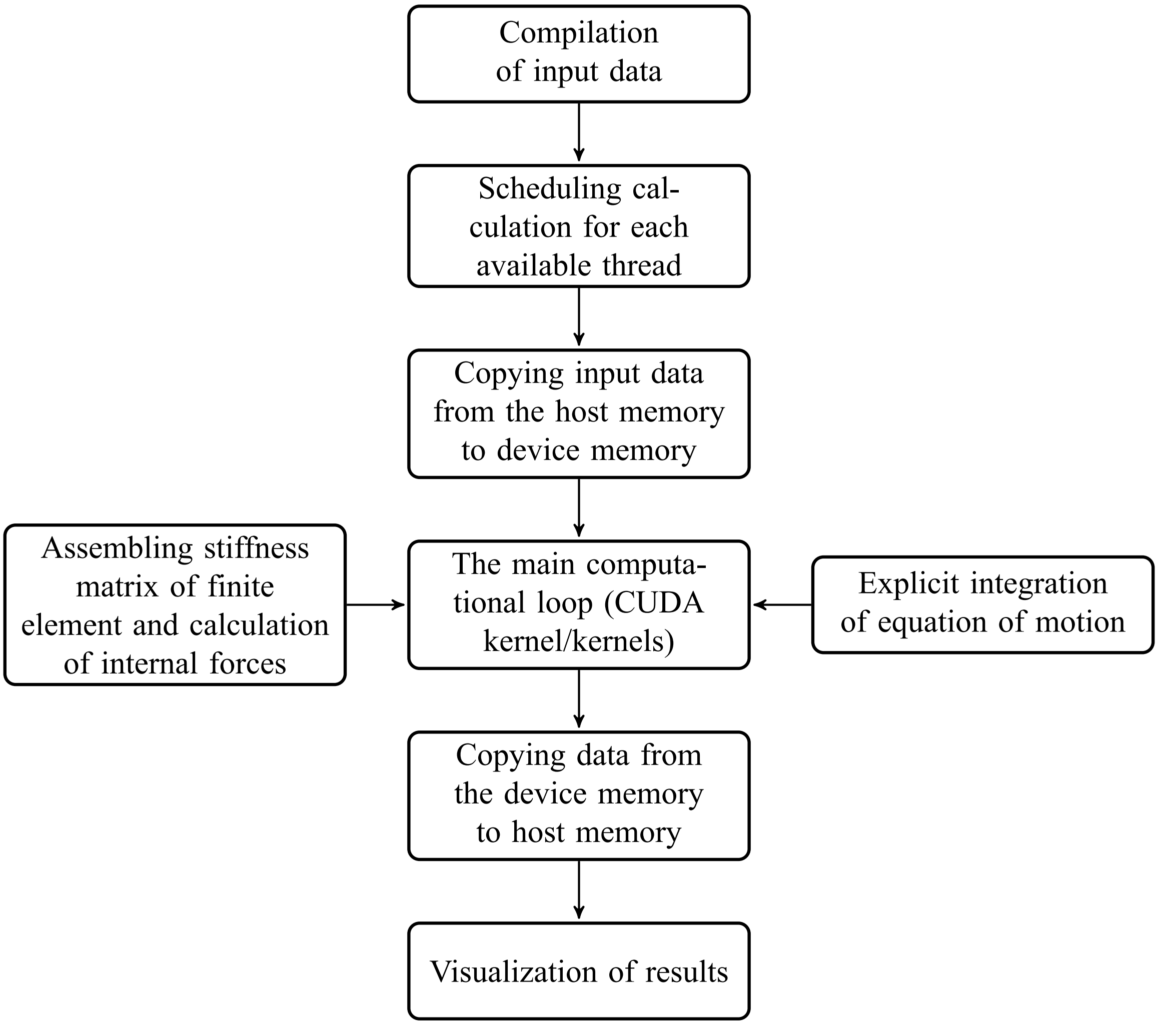
Abstract: This paper introduces a procedure for parallel computing with the Dynamic Relaxation method (DR) on a Graphic Processing Unit (GPU). This method facilitates the consideration of a variety of nonlinearities in an easy and explicit manner. Because of the presence of inertial forces, a static problem leads to a transient dynamic problem where the Central Difference Method is used as a method for direct integration of equations of motion which arise from the Finite Element model. The natural characteristic of this explicit method is that the scheme can be easily parallelized. The assembly of a global stiffness matrix is not required. Due to slow convergence of this method, the high performance which GPUs provide is strongly suitable for this kind of computation. NVIDIA's CUDA is used for general-purpose computing on graphics processing units (GPGPU) for NVIDIA's GPUs with CUDA capability.

Introduction

Dynamic Relaxation (DR) method is basicaly an explicit integration scheme for solution of equations of motion by Central Difference Method, which is commonly used in Dynamic Analysis. DR was originaly introduced by Day [1] for the finite difference analysis of concrete pressure vessels and in the next years was investigated mainly by P. Underwood [2]. The conditional stability character of explicit methods leads in some cases to necessity to use exceptionaly small integration step. As a consequence, the method was time consuming and remained at the mere periphery of scientific interest. Since year 2006, when CUDA (Compute Unified Device Architecture) was introduced by NVIDIA company, explicit methods have become more attractive. Availability of a huge number of parallel threadsdirectly implemented in hardware enabled their effective usage.

In the past, it was common to use CPU (Central Processing Unit) for parallel run of computional time-consuming tasks. Parallel programming was especially supported by software libraries OpenMP (Open Multi-Processing), MPI (Message Passing Interface) etc. These libraries for parallel run of variety of computations in computational mechanics are very well known to interested researchers especially for domain decomposition method (see [3]). CUDA and related technologies like OpenCL (Open Computing Language) or Microsoft Direct-Compute are currently investigated by many researchers. Their applications occurs in research interested mainly in information technologies, in algorithms for artificial inteligence (diagnosis and synthesis of voice, image recognition, etc.), for programming of shaders for production of special effects in computer visualization or in computational fluid dynamics (see [6]).

Paradigm of usage of CUDA technology in computer science is also available for computional structural mechanics and dynamics, namely for explicit methods.



References

[1] A.S. Day, An Introduction to Dynamic Relaxation, The Engineer 219 (1965) 218-221.

[2] P. Underwood, Dynamic Relaxation, Computational Methods for Transient Analysis (1983) 245-265.

[3] J. Kruis, Domain Decomposition Methods for Distributed Computing, Saxe-Coburg Publications, 2006.

[4] E. Oñate, Structural Analysis with the Finite Element Method. Linear Statics: Volume 1: Basis and Solids, Springer, 2009.

[5] E. Oñate: Structural Analysis with the Finite Element Method. Linear Statics: Volume 2: Beams, Plates and Shells, Springer, 2013.

[6] R. Couturier, Designing Scientific Applications on GPUs, CRC Press, 2013.

[7] S. Cook, CUDA Programming: A Developer's Guide to Parallel Computing with GPUs, Morgan Kaufmann, 2012.

[8] S. Prata, C Primer Plus, fifth ed., Sams Publishing, 2004.