Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 305 (2016) 827–848
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Finite element numerical integration for first order approximations on multi- and many-core architectures

Krzysztof Banaś[a,*], Filip Krużel[b], Jan Bielański[a]

[a] Department of Applied Computer Science and Modelling, AGH University of Science and Technology, Mickiewicza 30, 30-059 Kraków, Poland
[b] Institute of Computer Science, Cracow University of Technology, Warszawska 24, 31-155 Kraków, Poland

Received 25 November 2015; received in revised form 29 February 2016; accepted 23 March 2016
Available online 4 April 2016

Abstract

The paper presents investigations on the performance of the finite element numerical integration algorithm for first order approximations and three processor architectures, popular in scientific computing, classical x86_64 CPU, Intel Xeon Phi and NVIDIA Kepler GPU. We base the discussion on theoretical performance models and our own implementations for which we perform a range of computational experiments. For the latter, we consider a unifying programming model and portable OpenCL implementation for all architectures. Variations of the algorithm due to different problems solved and different element types are investigated and several optimizations aimed at proper optimization and mapping of the algorithm to computer architectures are demonstrated. The experimental results show the varying levels of performance for different architectures, but indicate that the algorithm can be effectively ported to all of them. The conclusions indicate the factors that limit the performance for different problems and types of approximation and the performance ranges that can be expected for FEM numerical integration on different processor architectures.

Keywords: Finite element method; First order approximation; Numerical integration; Multi-threaded programming; Multi-core processors; Graphics processors

1. Introduction

Finite element numerical integration forms an indispensable part of practically all complex finite element codes. Apart from special formulations for specific problems (such as e.g. applications in structural mechanics for beams, plates, etc.) where it can be avoided using some analytically obtained formulas, it is universally used for creating entries to finite element stiffness matrices and load vectors (matrices and right hand side vectors for systems of linear equations).

* Corresponding author.
E-mail address: pobanas@cyf-kr.edu.pl (K. Banaś).

http://dx.doi.org/10.1016/j.cma.2016.03.038

The recent progress in hardware for numerical computations caused the widespread use of multi-core and massively multi-core (often termed many-core) processors with vector processing capabilities. These created the necessity to reconsider the implementations of scientific codes, taking into account the possible options for multi-threading and the changing execution environment characteristics.

The subject of numerical integration on modern multi-core processors, especially graphics processors, has been investigated in several contexts. The first works concerned GPU implementations for specific types of problems, such as higher order FEM approximations in earthquake modeling and wave propagation problems [1], GPU implementations of some variants of discontinuous Galerkin approximation [2] or higher order approximations for electromagnetics problems [3].

The second important context for which finite element numerical integration was considered is the creation of domain specific languages and compilers. Interesting works for this approach include [4–7].

Finally, numerical integration, with more or less details, has been discussed in the context of the whole simulation procedure by the finite element codes. Some works in this area include [8–14]. Usually more attention to numerical integration has been given in articles that consider the process of creation of systems of linear equations for finite element approximations, with the important examples such as [15–18].

1.1. The place of the current research

The present paper forms a continuation of our earlier works, devoted solely to the subject of finite element numerical integration. The first papers considered higher order approximations, starting with first theoretical and experimental investigations in [19,20] and culminating in larger articles devoted to the implementation and performance of numerical integration kernels for multi-core processors with vector capabilities [21] (especially IBM Power XCell processor) and GPUs [22].

The investigations in the current paper on one hand touch the narrower subject of only first order approximations, but on the other hand concern two important types of finite elements (geometrically linear and non-linear) and are performed, in a unifying way, for three different processor architectures used in scientific computing. This scope, combined with the depth of investigations concerning the performance, differentiate the article from the other on this subject.

The aim of the research presented in the current paper is to thoroughly analyze the process of finite element numerical integration for first order approximations, in a possibly broad context, and to provide indications of the expected performance and execution time when finite element calculations are performed on contemporary processors. We base our implementations and perform numerical experiments for OpenCL as a unifying platform for different processor architectures, but include also considerations for other programming models, based on literature on the subject and theoretical performance models. Hence the conclusions of the paper can be twofold. On one hand, more specific comments can be given on the performance of OpenCL implementations for different processor architectures. On the other hand, more general statements concerning the execution time for numerical integration independent of the programming model can be formulated, that may be useful not only for creation of finite element programs but also for using them.

Hence, the intended audience includes not only people designing and developing finite element codes, but also users of finite element programs and developers of algorithms (such as e.g. linear algebra libraries) related to finite element computations.

We intentionally leave the subject of finite element assembly (or in general, the use of computed entries to local element stiffness matrices in further steps of solution procedures) out of the scope of the paper (apart from some general comments) since we believe it is a broad subject, that should be considered separately, what we plan to do in forthcoming papers. We believe that the subject of numerical integration is important on its own and worth of independent studies.

The paper is organized as follows. First, finite element numerical integration is described in the form investigated in our work. Then, in Section 3, its implementation, in the context of finite element simulations and for different types of problems, elements and computing platforms is analyzed. Several variants of the procedure are developed, and, in the next section, their optimization and mapping to different processor architectures using OpenCL is discussed. Section 5 presents the results of computational experiments that test the performance of procedures with short discussion of results. The last section presents the final conclusions of the paper.