Available online at www.sciencedirect.com
ScienceDirect
Mathematics and Computers in Simulation xxx (xxxx) xxx
MATHEMATICS AND COMPUTERS IN SIMULATION
www.elsevier.com/locate/matcom

Original articles

Higher order finite elements in space and time for anisotropic simulations with variational integrators. Application of an efficient GPU implementation

M. Bartelt*, O. Klöckner, J. Dietzsch, M. Groß

Technische Universität Chemnitz, Professorship of Applied Mechanics and Dynamics, Reichenhainer Straße 70, D-09126 Chemnitz, Germany

Received 27 July 2018; received in revised form 26 June 2019; accepted 20 September 2019
Available online xxxx

Abstract

The efficient use of resources is one of the most important things in the current development. In the field of lightweight constructions, rubber and reinforced polymer composite materials become more and more important. This constantly increases the demands on the simulation software. The development of robust and efficient algorithms for these simulations is the aim of this paper.

The application of higher order finite elements in space has a good influence on solution quality in space, but causes a higher computational cost compared to linear elements. We propose higher order finite elements in space and time in combination with a reinforced viscoelastic material formulation in a variational framework. Therefore, a higher order approximation in space and time is applied. The application of variational based time integrators guarantees the preservation of the total balance of linear momentum and the total balance of angular momentum. In order to fulfill the total balance of energy, an extension with discrete gradients is developed for the variational framework. The achieved time stepping scheme represents a very robust and consistent algorithm for the application of transient finite element simulations with reinforced viscoelastic materials and boundary conditions.

In an implementation, however, the higher order approximation in space and time combined with the viscoelastic material suffers from a high computational effort. Hence, an efficient implementation is required in order to reduce the computational time to a minimum. In our approach, we face this problem by using a GPU and the programming architecture Cuda from NVIDIA, which allows a massive parallelization of time-consuming parts of the simulation. We introduce a pipeline design for the GPU implementation, which provides multiple advantages. This design allows a simple porting of an already existing implementation by means of self-managing pipeline-stages. However, a significant speedup is still achieved due to further optimizations which exploit the architecture of GPUs. In addition, when combining both hardware resources GPU and CPU the computational time can be reduced significantly once more. Therefore, our GPU implementation easily allows a distribution of computational effort between both GPU and CPU. Finally, we show in numerical examples the reached speedup of this approach, and the impact of combining the GPU and the CPU is studied in detail.

Keywords: Energy conservation; Variational integrator; Higher order accuracy; Fiber reinforced viscoelasticity; CUDA; GPU

* Corresponding author.
E-mail addresses: matthias.bartelt.scientific@freenet.de (M. Bartelt), oliver.kloeckner@s2015.tu-chemnitz.de (O. Klöckner), julian.dietzsch@mb.tu-chemnitz.de (J. Dietzsch), michael.gross@mb.tu-chemnitz.de (M. Groß).

https://doi.org/10.1016/j.matcom.2019.09.018

1. Introduction

The increasing development of lightweight construction in the automotive and aircraft fields caused the application of more complex materials. Therefore, the use of viscoelastic materials is essential, because effects like relaxation and dumping cannot be reproduced by hyperelastic materials. Furthermore, different materials classes are combined to new materials that depend on several material properties like reinforced viscoelastic materials. To represent the physical properties of the real material in simulations the requirements increase continuously relating to the robustness and efficiency. Therefore, we present an application of different material formulations in one example to cover a large field of real materials. This includes the application of several formulations based on a NEO-HOOKIAN and SAINT VENANT-KIRCHHOFF model for the elastic and viscoelastic parts, further information are listed in Reference [11,12,21]. The extension to a reinforced viscoelastic material formulation is based on the work of [1].

Compared to linear elements in space, higher order elements in space require more computational effort, but in cases of very accurate solutions they can produce faster results, see References [3,15,18]. Motivated by this approach, we compare the use of different approximation orders in space to extend the work of [2].

Beside the approximation in space and the material formulation the time stepping scheme have to be kept in mind. The stability and the robustness strictly depend on the chosen approximation and the order of approximation in time. A very robust class of time stepping schemes are variational integrators, see [13]. A mechanical system can be formulated by a LAGRANGE function. Compared to [4], the approximation in time does not start with the equations of motion but with the approximation in the LAGRANGE function. The advantages of this formulation are the preservation of the total balance of linear momentum and the total balance of angular momentum. The linear approximation of the LAGRANGE function can be extended to a higher order approximation and can also be used to formulate constraint systems, see [16,20]. The disadvantage of this concept is that the error in the total balance of energy is only bounded but not exactly fulfilled. Therefore, we adapt the concept of discrete gradients from [9] in the same way like [12]. This leads to a time stepping scheme of higher order accuracy and preservation of the total balance of energy. In order to achieve an efficient implementation in addition to the physical accuracy, we show an opportunity for an implementation and parallelization on a GPU.

So-called general-purpose graphics processing units “GPGPU” are nowadays commonly used for accelerating various applications, e.g. high performance finite elements as in [8], where especially linear algebraic operations are accelerated. Compared to a CPU, a GPU is specialized in performing the same instruction on multiple data sets in parallel which leads to a high level of computational parallelism. In fact, linear algebraic operations like matrix–vector operations and matrix–matrix operations benefit from this feature, and with CUBLAS, see [5], a library already exists to accelerate such operations. However, even though those libraries are specialized on BLAS operations, they suffer from performance loss when e.g. a great number of matrix–vector operations with few entries is required. Furthermore, in cases where self-written code provides a significant speedup in comparison to an equivalent BLAS function, e.g. due to adaptions to the problem, it should be possible to port this code on the GPU. Therefore, we want to introduce an implementation in C and C++ for the GPU, which not only focuses on these BLAS operations. Our approach follows a pipeline design and uses the programming architecture Cuda in order to exploit the maximum possible performance of NVIDIA GPUs like the TESLA K20C. The pipeline consists of multiple stages, which either execute self-written and ported code or functionalities from libraries, and are optimized for the GPU based on [6,7,17,19]. These stages are connected with their predecessors and successor and the pipeline manages them and guarantees the right of execution and synchronization with the CPU. Additionally, the pipeline design allows further improvements, e.g. pipeline splitting in order to overcome the limited amount of memory space on the GPU and the parallel execution of stages based on the functionality of streams on Cuda. Another crucial advantage is the ability of distributing computational effort easily between the GPU and CPU to increase the speedup significantly. However, the impacts of the distribution on the performance have to be studied as well to assure the maximum utilization on both hardware resources, the GPU and CPU.

In this work Section 1 contains a brief summary of the developed time stepping schemes with a higher order approximation in time. The GALERKIN variational integrators have the advantages that, first, they are very robust and, second, preserve the balance of total linear momentum and the balance of total angular momentum. However, they cannot preserve the balance of total energy in the order of the NEWTON tolerance.

In Section 2 we introduce a discrete gradient that bounds the balance of the total energy in the order of the Newton tolerance. This section also includes the theoretical conservation of the balance of total linear momentum,

Please cite this article as: M. Bartelt, O. Klöckner, J. Dietzsch et al., Higher order finite elements in space and time for anisotropic simulations with variational integrators. Application of an efficient GPU implementation, Mathematics and Computers in Simulation (2019), https://doi.org/10.1016/j.matcom.2019.09.018.