Discontinuous Galerkin method applied to elastodynamics and its implementations for GPU computing

Pierre-Yves Guerder

January 13th, 2011

Keywords: Discontinuous Galerkin, GPU, finite elements method, spectral elements method, elastodynamics
Description: GPU implementation of the discontinuous Galerkin method in elastodynamics
Last modification: March 20, 2013

Introduction

The context is about finding an approximate numerical solution to the elastic wave equation, including for non-linear problems and for materials with a complex shape. The number of available methods to solve this problem is relatively small, which makes the Discontinuous Galerkin (DG) method particularly interesting. We will describe first the principle and improvements of the DG method, then its alternatives and finally its GPU implementations.

Galerkin method

Principle

In 2-D, the space is divided into a mesh of contiguous triangular or quadrilateral elements; in 3-D, those are tetrahedral volumes. The Finite Elements Method (FEM) approximates the unknown solution of the differential equation with polynomials on each element of the grid but does not allow a local statement on each element. [1]

In the Finite Volume Method (FVM), the equation is solved by calculating the local average on each of those volumes. It is an iterative process and the value at t_n only depends on the value at t_{n-1} and on the fluxes that enter and leave the volume. This method is rather simple to implement but does not allow high-order accuracy. [1]

The DG method combines the advantages of those two methods: using polynomials of arbitrary order of the FEM and using numerical fluxes at the interfaces of the FVM. The main difference between DG and the FEM is that polynomials must be continuous at the interface of the elements in the FEM whereas the solution can be discontinuous across the element interfaces in DG, which allows the incorporation the well-established ideas of numerical flux functions from the FVM. So, by choosing carefully the values of those fluxes, it is possible to minimize the error introduced by the approximation. [1]

The DG method was first introduced by Reed and Hill in 1973, to solve hyperbolic partial differential equations in the context of the neutron transport equation [2]. It is also applied to elliptic problems, but those don't appear in elastodynamics (only in elastostatics where there is no time dependency).

Solving the equation with DG can be done in two ways: modal (decomposition of the local solution in a linear combination of several resonating modes) or nodal (resolution at chosen points before interpolating by decomposing the local solution in a linear combination of orthogonal polynomials). Moreover, the method can be p-adaptive (the order of the polynomials can very from one element to another), h-adaptive (the element size can vary locally) or hp-adaptive (both can vary) [3].

Application of DG-FEM to elastodynamics

The elastic wave equation can be written as a conservation equation which links the time derivative of a state vector Q to the space derivatives of the flux vectors F, G et H and to the source vector S. This form makes the DG resolution easier, especially for the nodal method described by J. S. Hesthaven [1] and also facilitates the inclusion of non-linear effects.

In computational seismology, M. Käser [4] proposed to use the DG-FEM combined with the ADER (Arbitrary high-order DERivatives) method, in order to solve the equation with high orders of accuracy (practically up to order 10 for polynomials), both in time and in space. Time accuracy is crucial when a wave is propagating over several wavelengths. Then, J. de la Puente [5] extended this method to 3-D anisotropic materials and coupled it with viscoelastic effects.

Recently, C. E. Castro [6] improved the ADER-DG method by adding space-variable coefficients that allow to deal with heterogeneous materials, including inside the same cell of the mesh. However, this improvement only concerns 2-D

for now and does not deal well with discontinuities inside a cell. A recurrent issue is about simulating the absorbing (where the wave is not reflected) and free surface boundaries. This can be done by imposing the numerical fluxes [4].

Alternatives

Using the Spectral Elements Method (SEM) combines the advantages of the spectral methods (accuracy) with those (flexibility) of the FEM. However, this method uses high orders (typically 4 to 10) [7] for space integration but not for time integration (order 2; order 4 at most), which is problematic when the wave has several wavelengths [8]. In the SEM, the mass matrix is global. Mass-lumping consists in approximating the mass matrix to make it diagonal, so that its inversion becomes trivial [3].

The SEM is computationally more efficient than ADER-DG for the same result: 4 min. compared to 200 min. for ADER-DG for the same case. This is because DG requires more elements in the mesh and more calculations per element [5]. However, the SEM must use a conforming hexahedral mesh (the six sides of each hexahedral element must match up exactly with the sides of neighboring elements). Moreover, the mesh should honor the first- and second-order discontinuities in the model and the size of the elements should reflect the distribution of wave speeds. So, the design of a mesh is problem-dependent and more complicated than in DG [7]. To sum up, the SEM is faster but DG is more flexible.

GPU implementations

The FEM and the Finite Difference Method (FDM) require a lot of calculations to provide a solution. However, since the problem is divided into several elements, most of the calculations can be done at the same time on a massively parallel architecture. Graphical processors (GPUs) are massively parallel compared to CPUs. D. Michéa [9] implemented a FDM using the CUDA programming language on an NVIDIA graphic card and decreased the calculation time by a factor between 20 to 60 compared to CPUs. They noticed that the disadvantage of GPUs compared to CPUs is the limited amount of shared memory and number of registers. That's why the main difficulty in GPU programming is to keep the number of accesses to the global memory as low as possible. In order to do so, the threads groups have to be well distributed so that they can share as much data as possible.

A. Klöckner [10] has implemented a DG method in a software called *Hedge*. This enables nodal DG to run on GPUs. The authors applied their algorithm to the resolution of Maxwell's equation with polynomial orders from 1 to 9. The speedup between CPUs and GPUs varies from 14 to 65, the maximum being reached for the order 4.

References

[1] J. S. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods*. Springer, 2007. 1

[2] W. H. Reed and T. R. Hill, "Triangular mesh methods for the neutron transport equation," *Tech. Rep. LA-UR-73-479*, 1973. 1

[3] V. Etienne, E. Chaljub, J. Virieux, and N. Glinsky, "An hp-adaptive discontinuous galerkin finite-element method for 3-d elastic wave modelling," *Geophys. J. Int.*, pp. 941–962, 2010. 1, 2

[4] M. Käser and M. Dumbser, "An arbitrary high-order discontinuous galerkin method for elastic waves on unstructured meshes – i. the two-dimensional isotropic case with external source terms," *Geophys. J. Int.*, pp. 855–877, 2006. 1, 2

[5] J. de la Puente, M. Käser, M. Dumbser, and H. Igel, "An arbitrary high-order discontinuous galerkin method for elastic waves on unstructured meshes – iv. anisotropy," *Geophys. J. Int.*, pp. 1210–1228, 2007. 1, 2

[6] C. E. Castro, M. Käser, and G. B. Brietzke, "Seismic waves in heterogeneous material: subcell resolution of the discontinuous galerkin method," *Geophys. J. Int.*, pp. 250–264, 2010. 1

[7] D. Komatitsch and J. Tromp, "Spectral-element simulations of global seismic wave propagation—i. validation," *Geophys. J. Int.*, pp. 390–412, 2002. 2

[8] M. Dumbser and M. Käser, "An arbitrary high-order discontinuous galerkin method for elastic waves on unstructured meshes – ii. the three-dimensional isotropic case," *Geophys. J. Int.*, pp. 319–336, 2006. 2

[9] D. Michéa and D. Komatitsch, "Accelerating a three-dimensional finite-difference wave propagation code using gpu graphics cards," *Geophys. J. Int.*, pp. 389–402, 2010. 2

[10] A. Klöckner, T. Warburton, J. Bridge, and J. Hesthaven, "Nodal discontinuous galerkin methods on graphics processors," *Journal of Computational Physics*, pp. 7863–7882, 2009. 2