



High-Order Time Discretization Of The Wave Equation By Nabla-P Scheme

Hélène Barucq, Henri Calandra, Julien Diaz, Florent Ventimiglia

► To cite this version:

Hélène Barucq, Henri Calandra, Julien Diaz, Florent Ventimiglia. High-Order Time Discretization Of The Wave Equation By Nabla-P Scheme. ESAIM: Proceedings, 2014, 45, pp.67 - 74. 10.1051/proc/201445007 . hal-01111071

HAL Id: hal-01111071

<https://inria.hal.science/hal-01111071>

Submitted on 29 Jan 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HIGH-ORDER TIME DISCRETIZATION OF THE WAVE EQUATION BY NABLA-P SCHEME.*

HÉLÈNE BARUCQ¹, HENRI CALANDRA², JULIEN DIAZ³ AND FLORENT VENTIMIGLIA⁴

Abstract. High-order Discontinuous Galerkin Methods (DGM) are now routinely used for simulation of wave propagation, especially for geophysical applications. However, to fully take full advantage of the high-order space discretization, it is relevant to use a high-order time discretization. Hence, DGM are currently coupled with ADER schemes, which leads to high-order explicit time schemes, but requires the introduction of auxiliary unknowns. The memory can thus be considerably cluttered up. In this work, we propose a new high order time scheme, the so-called Nabla-p scheme. This scheme does not increase the storage costs since it is a single step method which does not require introducing auxiliary unknowns. Numerical results show that for a given accuracy, the new scheme requires less computational burden regarding both the memory and the computational costs than the DG-ADER scheme.

Résumé. Les méthodes de Galerkin Discontinues (DGM) d'ordre élevé sont maintenant couramment utilisées pour la simulation de la propagation des ondes, en particulier pour les applications géophysiques. Cependant, pour profiter pleinement d'une discrétisation en espace d'ordre élevé, il est pertinent d'utiliser une discrétisation en temps d'ordre élevé. C'est pourquoi les DGM sont souvent associées à des schémas ADER, ce qui conduit à des formulations en temps explicites et d'ordre élevé. Toutefois, les méthodes DG-ADER nécessitent d'introduire des inconnues auxiliaires, ce qui peut encombrer considérablement la mémoire. Dans ce travail, nous proposons un nouveau schéma temporel d'ordre élevé, le schéma Nabla-p. Ce schéma n'augmente pas les coûts de stockage car il s'agit d'une méthode à un pas, sans utilisation d'inconnues auxiliaires. Les résultats numériques montrent qu'il nécessite moins d'espace mémoire et qu'il coûte moins cher que la méthode DG-ADER pour une précision donnée.

INTRODUCTION

Numerical simulation of wave propagation in realistic domains requires the development of advanced numerical methods which are capable of reproducing the effects of heterogeneities on waves. The solution of wave problems is involved in seismic imaging processes such as the Reverse Time Migration (RTM) technique. In that context, the accuracy of the numerical solution is not the only concern because the image of the subsurface is produced by solving a collection of wave equations. That means that, given a level of accuracy that ensures the quality of the image, the use of the memory and the computational time must also be considered to determine

* Thanks to DIP strategic action support. [<http://dip.inria.fr/>]

¹ EPI Magique 3D, INRIA Bordeaux Sud-Ouest, IPRA-LMA.

² TOTAL Exploration/Production.

³ EPI Magique 3D, INRIA Bordeaux Sud-Ouest, IPRA-LMA.

⁴ EPI Magique 3D, INRIA Bordeaux Sud-Ouest, IPRA-LMA and TOTAL Exploration/Production.

if the numerical method is efficient for industrial purposes. RTM is well-known to be computationally intensive and besides the huge progresses of scientific computing, high-order numerical methods provide efficient alternative to decrease the computational costs. In case of heterogeneous media including topography effects, it is relevant to handle high-order Discontinuous Galerkin Methods (DGM) which have demonstrated a high level of flexibility by allowing the use of different orders of approximation to solve a problem. Moreover, these methods are multi-thread oriented, which is mandatory to benefit from the last computing architectures. Now DGMs only address the issue of discretizing the equations in space. The most widely time discretization method for wave problems involves the Leap-Frog (LF) scheme. The Leap-Frog scheme belongs to the family of explicit time schemes, which provide a direct representation of the wave fields as compared to implicit formulations. But the stability is ensured under the so-called Courant Friedrichs Lewy (CFL) condition. The LF scheme is of order two only, and it does allow to take full advantage of the high order space discretization. It is thus interesting to combine DGMs with high-order time schemes. This has been achieved with DG-ADER methods [3], which are an extension of the Modified Equation Technique [4,5]. Regarding the memory use, DG-ADER schemes are relevant because they are single step time integration procedures. Thus, they only require to store the solution at the previous time step. Nevertheless, memory limits can be reached when solving 3D wave equations, in particular because their implementation is based upon auxiliary unknowns. That is why we propose a new time scheme which requires less memory than DG-ADER methods for a given level of accuracy. The construction of the new scheme is based on the fact that DGMs are well-suited for the approximation of high-order space operators. By exploiting this property, it is relevant to construct high-order time schemes which involve high-order space operators. This can be done by following the same approach than for the Modified Equation technique [4,5] but by working with the continuous problem directly. By this way, we address the time discretization directly while the classical used technique consists in applying the space discretization first and next the time integration scheme. In this work, we show that it is possible to construct a single-step time scheme of arbitrary order which demonstrates a high level of accuracy while requiring acceptable computational costs. For a given accuracy, we show that the new scheme allows for using much coarser meshes than with DG-ADER methods. The storage and the computational times are thus considerably reduced.

1. DISCRETIZATION OF THE WAVE EQUATION

1.1. DG space discretization

To simplify the presentation, we focus on the acoustic wave equation but the method can be applied to the elastodynamic wave equation too. We consider the following system in a bounded domain $\Omega \subset \mathbf{R}^n$, with boundary $\partial\Omega$, $n=1,2,3$:

$$\begin{cases} \rho(\mathbf{x}) \frac{\partial \mathbf{v}(\mathbf{x}, t)}{\partial t} + \nabla p(\mathbf{x}, t) = 0 & \text{in } \Omega \times [0, T] \\ \frac{1}{\mu(\mathbf{x})} \frac{\partial p(\mathbf{x}, t)}{\partial t} + \nabla \cdot (\mathbf{v}(\mathbf{x}, t)) = 0 & \text{in } \Omega \times [0, T] \end{cases} \quad (1)$$

where ρ and μ are respectively the density and the compressibility modulus of Ω , p is the scalar pressure and \mathbf{v} the velocity vector. We do not discuss here the regularity properties of ρ and μ and we simply assume that these two functions are positive, bounded and piecewise continuous. For the sake of simplicity, we omit the source function. The initial condition is taken such that $p(\mathbf{x}, t) = \sin(2\pi(\mathbf{x} - t))$ and we use periodic boundary conditions. Now we consider a triangulation \mathcal{T}_h of Ω composed of triangles or tetrahedra K with edges Γ and we use a Discontinuous Galerkin Method similar to the one proposed in [2]. Let V_h be the space defined by :

$$V_h = \{v \in L^2(\Omega), v|_K \in \mathcal{P}_p(K), \forall K \in \mathcal{T}_h\}$$

where \mathcal{P}_p is the space of polynomials of degree $\leq p$. The DG formulation of (1) reads : Find $(p_h(t), \mathbf{v}_h(t)) \in V_h \times V_h^d$, such that, $\forall (q, \mathbf{w}) \in V_h \times V_h^d$

$$\left\{ \begin{array}{l} \sum_{K \in \mathcal{T}_h} \int_K \rho \frac{\partial \mathbf{v}_h}{\partial t} \cdot \mathbf{w} - \sum_{K \in \mathcal{T}_h} \int_K \nabla p_h \cdot \mathbf{w} + \sum_{\Gamma \in \Gamma_{\text{int}}} \int_{\Gamma} [[\mathbf{w}]] \{ \{ p_h \} \} \cdot \mathbf{n} = 0, \\ \sum_{K \in \mathcal{T}_h} \int_K \frac{1}{\mu} \frac{\partial p_h}{\partial t} q - \sum_{K \in \mathcal{T}_h} \int_K \mathbf{v}_h \cdot \nabla q + \sum_{\Gamma \in \Gamma_{\text{int}}} \int_{\Gamma} [[q]] \{ \{ \mathbf{v}_h \} \} \cdot \mathbf{n} = 0, \end{array} \right. \quad (2)$$

In (2) we use classical DG notations. Let Γ_{int} be the set of interior edges. The terms $[[\bullet]]$, $\{ \{ \bullet \} \}$ stand for the jump and the mean of a function over an interior edge Γ . To define these quantities, we denote arbitrarily by K^+ and K^- the two elements sharing Γ . \mathbf{n} is the normal to Γ pointing from K^+ to K^- . We then have

$$[[q]] = q^+ - q^- \text{ and } \{ \{ \mathbf{w} \} \} = \frac{1}{2} (w^+ + w^-).$$

To obtain 2, we have used the fact that the solution to the continuous problem $v(\mathbf{x}, t)$ and $p(\mathbf{x}, t)$ satisfy $[v \cdot \mathbf{n}] = 0$ and $[p] = 0$ on each edge. Using for instance the classical Lagrange basis functions, we obtain the semi-discretized scheme :

$$\left\{ \begin{array}{l} \mathcal{M}_v \frac{d \mathbf{V}}{dt} + \mathcal{K}_p \mathbf{P} = 0 \\ \mathcal{M}_p \frac{d \mathbf{P}}{dt} + \mathcal{K}_v \mathbf{V} = 0 \end{array} \right. \quad (3)$$

where the mass matrices \mathcal{M}_v , \mathcal{M}_p are easily invertible since they are block diagonal and the stiffness matrices \mathcal{K}_p , \mathcal{K}_v are sparse. By multiplying by \mathcal{M}_v^{-1} and \mathcal{M}_p^{-1} , we get a simpler formulation :

$$\left\{ \begin{array}{l} \frac{d \mathbf{V}}{dt} - \mathcal{A}_p \mathbf{P} = 0 \\ \frac{d \mathbf{P}}{dt} - \mathcal{A}_v \mathbf{V} = 0 \end{array} \right. \quad (4)$$

which is more convenient to exhibit the time scheme integration process.

1.2. DG-ADER Time discretization

A classical way to compute the time derivative of first-order system consists in using the classical second-order Leap-Frog scheme which leads to the following system:

$$\left\{ \begin{array}{l} \frac{\mathbf{V}^{n+1} - \mathbf{V}^n}{\Delta t} - \mathcal{A}_p \mathbf{P}^{n+\frac{1}{2}} = 0 \\ \frac{\mathbf{P}^{n+\frac{3}{2}} - \mathbf{P}^{n+\frac{1}{2}}}{\Delta t} - \mathcal{A}_v \mathbf{V}^n = 0 \end{array} \right. \quad (5)$$

This scheme is second-order accurate only, which limits the level of convergence of the numerical method. That is why DG-ADER [3] schemes have been developed. DG-ADER method is equivalent to the Modified Equation Technique (MET) [4, 5], when using the same time step and the same order for the time discretization in the whole domain. The principle of the method is based on the use of a Taylor expansion in which the time derivatives are replaced thanks to (4). Using a 4th-order Taylor expansion of the pressure and the velocity, we

obtain:

$$\begin{aligned} \frac{\mathbf{V}(t^n + \Delta t) - \mathbf{V}^n(t^n)}{\Delta t} &= \partial_t \mathbf{V}(t^n + \frac{\Delta t}{2}) + \frac{\Delta t^2}{24} \partial_t^3 \mathbf{V}(t^n + \frac{\Delta t}{2}) + O(\Delta t^4) \\ \frac{\mathbf{P}(t^n + \frac{3\Delta t}{2}) - \mathbf{P}^n(t^n + \frac{\Delta t}{2})}{\Delta t} &= \partial_t \mathbf{P}(t^n + \Delta t) + \frac{\Delta t^2}{24} \partial_t^3 \mathbf{P}(t^n + \Delta t) + O(\Delta t^4) \end{aligned} \quad (6)$$

Using (4), we then get the 4th-order ADER scheme:

$$\begin{cases} \frac{\mathbf{V}^{n+1} - \mathbf{V}^n}{\Delta t} = \mathcal{A}_p \mathbf{P}^{n+\frac{1}{2}} + \frac{\Delta t^2}{24} \mathcal{A}_p \mathcal{A}_v \mathcal{A}_p \mathbf{P}^{n+\frac{1}{2}} \\ \frac{\mathbf{P}^{n+\frac{3}{2}} - \mathbf{P}^{n+\frac{1}{2}}}{\Delta t} = \mathcal{A}_v \mathbf{V}^n + \frac{\Delta t^2}{24} \mathcal{A}_v \mathcal{A}_p \mathcal{A}_v \mathbf{V}^n \end{cases} \quad (7)$$

In practice, this scheme is solved by applying the following algorithm :

- Computation of \mathbf{P}
 - $Q = \mathcal{A}_p \mathbf{P}^{n+\frac{1}{2}}$
 - $W = \mathcal{A}_v Q$
 - $\mathbf{V}^{n+1} = \mathbf{V}^n + \Delta t \left(Q + \frac{\Delta t^2}{24} \mathcal{A}_p W \right)$
- Computation of \mathbf{V}
 - $W = \mathcal{A}_v \mathbf{V}^{n+1}$
 - $Q = \mathcal{A}_p W$
 - $\mathbf{P}^{n+\frac{3}{2}} = \mathbf{P}^{n+\frac{1}{2}} + \Delta t \left(W + \frac{\Delta t^2}{24} \mathcal{A}_v Q \right)$

Note that we two auxiliary variables Q and W are introduced. This scheme requires also three times more multiplications by the stiffness matrices than the LF scheme, but the stability condition is multiplied by 2.8 [6, 7]. There is thus a offsetting effect. However, if we use higher-order time schemes, the improved stability condition does not counterbalance rising multiplications.

1.3. Nabla-p Time discretization

We propose here an alternative to DG-ADER scheme by applying the MET to the **continuous** wave equation (1). For a 4th-order scheme, we apply (6) on the continuous variables and we replace the derivative terms ∂_t and ∂_t^3 thanks to the continuous equation (2). We then obtain the semi-discretized time scheme:

$$\begin{cases} \frac{\mathbf{v}^{n+1} - \mathbf{v}^n}{\Delta t} = -\nabla p^{n+\frac{1}{2}} - \frac{\Delta t^2}{24} \nabla \nabla \cdot \nabla p^{n+\frac{1}{2}} \\ \frac{p^{n+\frac{3}{2}} - p^{n+\frac{1}{2}}}{\Delta t} = -\nabla \cdot \mathbf{v}^{n+1} - \frac{\Delta t^2}{24} \nabla \cdot \nabla \nabla \cdot \mathbf{v}^{n+1} \end{cases} \quad (8)$$

This method has already been applied to the second-order formulation of the acoustic wave equation [1] and it performs very well. Using a Green formula three times, with test functions q and \mathbf{w} regular enough, we then

obtain the following system :

$$\left\{ \begin{aligned} \sum_K \int_K \rho(\mathbf{x}) \frac{\mathbf{v}_h^{n+1} - \mathbf{v}_h^n}{\Delta t} \cdot \mathbf{w} &= \sum_K \int_K p_h^{n+\frac{1}{2}} \nabla \cdot \mathbf{w} + \frac{\mu(\mathbf{x})}{\rho(\mathbf{x})} \frac{\Delta t^2}{24} \sum_K \int_K p_h^{n+1/2} \nabla \cdot \nabla \nabla \cdot \mathbf{w} \\ &- \sum_{\Gamma_{int}} \int_{\Gamma} \left\{ \left\{ p_h^{n+1/2} \right\} \right\} [[\mathbf{w}]] \cdot \mathbf{n} - \frac{\Delta t^2}{24} \sum_{\Gamma_{int}} \int_{\Gamma} \left\{ \left\{ \frac{\mu(\mathbf{x})}{\rho(\mathbf{x})} \nabla \cdot \nabla p_h^{n+1/2} \right\} \right\} [[\mathbf{w}]] \cdot \mathbf{n} \\ &+ \frac{\Delta t^2}{24} \sum_{\Gamma_{int}} \int_{\Gamma} \left\{ \left\{ \frac{1}{\rho(\mathbf{x})} \nabla p_h^{n+1/2} \right\} \right\} \cdot \mathbf{n} [[\mu(\mathbf{x}) \nabla \cdot \mathbf{w}]] \\ &- \frac{\Delta t^2}{24} \sum_{\Gamma_{int}} \int_{\Gamma} \left\{ \left\{ p_h^{n+1/2} \right\} \right\} \left[\left[\frac{\mu(\mathbf{x})}{\rho(\mathbf{x})} \nabla \nabla \cdot \mathbf{w} \right] \right] \cdot \mathbf{n} \\ \sum_K \int_K \frac{1}{\mu(\mathbf{x})} \frac{p_h^{n+3/2} - p_h^{n+1/2}}{\Delta t} \cdot q &= \sum_K \int_K \mathbf{v}_h^{n+1} \nabla q + \frac{\mu(\mathbf{x})}{\rho(\mathbf{x})} \frac{\Delta t^2}{24} \sum_K \int_K \mathbf{v}_h^{n+1} \nabla \nabla \cdot \nabla q \\ &- \sum_{\Gamma_{int}} \int_{\Gamma} \left\{ \left\{ \mathbf{v}_h^{n+1} \right\} \right\} \cdot \mathbf{n} [[q]] - \frac{\Delta t^2}{24} \sum_{\Gamma_{int}} \int_{\Gamma} \left\{ \left\{ \frac{\mu(\mathbf{x})}{\rho(\mathbf{x})} \nabla \nabla \cdot \mathbf{v}_h^{n+1} \right\} \right\} \cdot \mathbf{n} [[q]] \\ &+ \frac{\Delta t^2}{24} \sum_{\Gamma_{int}} \int_{\Gamma} \left\{ \left\{ \mu(\mathbf{x}) \nabla \cdot \mathbf{v}_h^{n+1/2} \right\} \right\} \left[\left[\frac{1}{\rho(\mathbf{x})} \nabla q \right] \right] \cdot \mathbf{n} \\ &- \frac{\Delta t^2}{24} \sum_{\Gamma_{int}} \int_{\Gamma} \left\{ \left\{ \mathbf{v}_h^{n+1/2} \right\} \right\} \cdot \mathbf{n} \left[\left[\frac{\mu(\mathbf{x})}{\rho(\mathbf{x})} \nabla \cdot \nabla q \right] \right] \end{aligned} \right. \quad (9)$$

System (9) admits then the overall representation:

$$\left\{ \begin{aligned} \frac{\mathbf{V}^{n+1} - \mathbf{V}^n}{\Delta t} &= \left(\mathcal{A}_p + \frac{\Delta t^2}{24} \mathcal{B}_p \right) \mathbf{P}^{n+1/2} = \mathcal{A}_p^* \mathbf{P}^{n+1/2} \\ \frac{\mathbf{P}^{n+\frac{3}{2}} - \mathbf{P}^{n+\frac{1}{2}}}{\Delta t} &= \left(\mathcal{A}_v + \frac{\Delta t^2}{24} \mathcal{B}_v \right) \mathbf{V}^{n+1} = \mathcal{A}_v^* \mathbf{V}^{n+1}, \end{aligned} \right. \quad (10)$$

where \mathcal{A}_p^* and \mathcal{A}_v^* are the matrices containing original DG stiffness matrices \mathcal{A}_p and \mathcal{A}_v plus additional 4th-orders terms \mathcal{B}_p and \mathcal{B}_v . By construction, the number of non-zero elements of \mathcal{B}_p and \mathcal{B}_v is the same than for \mathcal{A}_p and \mathcal{A}_v . Hence, the computational cost of one iteration of the Nabla scheme is the same than for one iteration of the LF. Moreover, concerning the CFL, that we are not able to define theoretically, it is in practice more or less equal to 95% of the LF CFL. Then compared to DG-ADER, the Nabla-p CFL will not allows us to counterbalance any overcost induced by the method, as we will see in the next Section.

It should be noted that ADER scheme can also be rewritten as in (10), with $\mathcal{A}_p^* = \mathcal{A}_p + \frac{\Delta t^2}{24} \mathcal{A}_p \mathcal{A}_v \mathcal{A}_p$ and $\mathcal{A}_v^* = \mathcal{A}_v + \frac{\Delta t^2}{24} \mathcal{A}_v \mathcal{A}_p \mathcal{A}_v$. However, the use of auxiliary unknowns should be preferred, because the stencils of the matrices $\mathcal{A}_p \mathcal{A}_v \mathcal{A}_p$ and $\mathcal{A}_v \mathcal{A}_p \mathcal{A}_v$ are much larger than the stencils of the matrices \mathcal{A}_p and \mathcal{A}_v . To be more specific, after a half an iteration of the Leap-Frog and the Nabla scheme, the values of the pressure (resp. the velocity) on one element are modified by the values of the velocity (resp. the pressure) on the element and on its neighbor; after half an iteration of the ADER scheme, the values of the pressure (resp. the velocity) are also modified by the the values of the of velocity (resp. the pressure) on the second-level neighbors (the neighbors of the neighbors) and of the third-level neighbors. Hence the number of non-zero elements of the global matrix can be multiplied by a factor 5 in 2D and 10 in 3D compared with the original matrix, which increases dramatically the storage cost. The cost of three multiplications by the original matrix is not larger than the cost of one multiplication by the global matrix. Moreover, in a parallel framework, where the information should be communicated from one processor to another, we ADER schemes requires to communicate to one processor not only the values of its neighbors, but also the values of its second- and third- level neighbors.

2. 2D NUMERICAL RESULTS

We have performed a 2D comparison between the LF scheme, the 4th order DG-ADER scheme and Nabla scheme using \mathcal{P}_3 , \mathcal{P}_4 and \mathcal{P}_5 -elements in space. The domain is a 1000m by 1000m homogeneous square, the simulation duration is 60 seconds. We consider a regular mesh composed of isosceles right-angled triangles. The size of the legs of the triangles is denoted by h and is initially equal to 100m. We then refine the mesh by a factor δ varying from 1 to 16 (6.25m being the smallest space step). We consider periodic boundary conditions and the initial data are such that:

$$p(\mathbf{x}, t) = \sin(2\pi(\mathbf{x} - t)).$$

In Fig. 1, (resp. Fig. 2 and Fig. 3), we represent the relative \mathcal{L}^2 -error as a function of the space step h for \mathcal{P}_3 (resp \mathcal{P}_4 and \mathcal{P}_5) polynomials. Table 1 (resp 2 and 3) provides the relative \mathcal{L}^2 -error and the time step for a given refinement factor δ as well as the convergence order (CV) between two refinements. Note that, for \mathcal{P}_3 polynomials, we do not obtain the expected fourth order convergence. This is due to the fact that the original DG method does not converge with the optimal order [2, 6, 7]. To obtain the fourth order convergence, we need to use at least \mathcal{P}_4 polynomials. Table 2 and Table 3, indicate even a higher convergence rate for coarser meshes. For a given space step, Nabla and ADER clearly outperform LF, whatever the polynomial degree. For \mathcal{P}_3 elements, DG-ADER and Nabla exhibit the same level of performance, but Nabla provides better results than DG-ADER when the polynomial degree is increased. Indeed for \mathcal{P}_5 polynomials the error delivered by Nabla is ten times smaller than the one delivered by DG-ADER. However, comparing the error for a given mesh is not sufficient to claim Nabla scheme performs better. We also need to take into account the computational cost.

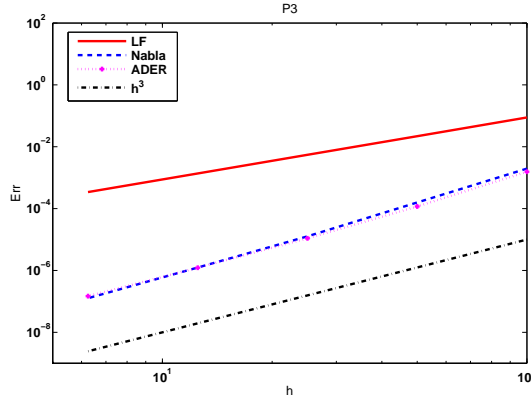
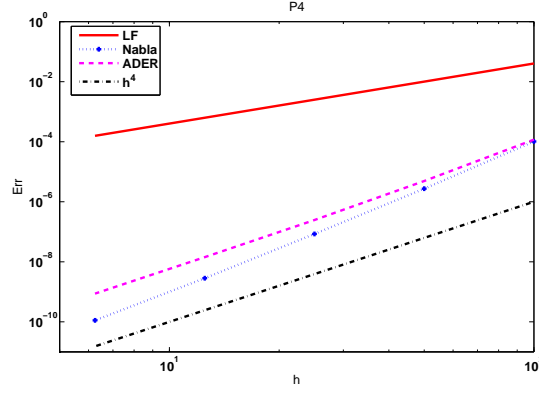


FIGURE 1. \mathcal{L}^2 -error with \mathcal{P}_3 -elements

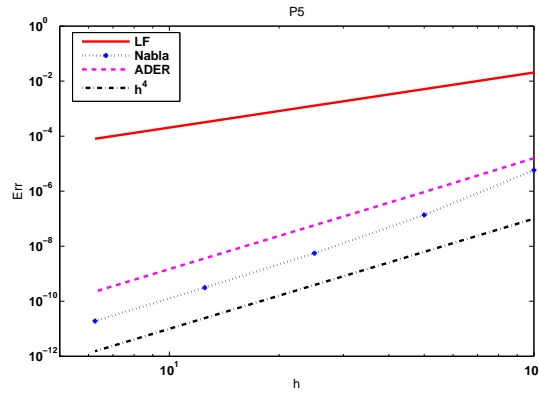
Leap-Frog				Nabla-3			DG-ADER		
δ	Error	Time Step	CV	Error	Time Step	CV	Error	Time Step	CV
1	$8.8 \cdot 10^{-2}$	$6.2 \cdot 10^{-2}$		$1.9 \cdot 10^{-3}$	$5.8 \cdot 10^{-2}$		$1.5 \cdot 10^{-3}$	$1.0 \cdot 10^{-1}$	1
2	$2.1 \cdot 10^{-2}$	$3.1 \cdot 10^{-2}$	2.0	$1.5 \cdot 10^{-4}$	$2.9 \cdot 10^{-2}$	3.6	$1.1 \cdot 10^{-4}$	$5.3 \cdot 10^{-2}$	3.8
4	$5.4 \cdot 10^{-3}$	$1.5 \cdot 10^{-2}$	2.0	$1.2 \cdot 10^{-5}$	$1.4 \cdot 10^{-2}$	3.6	$1.0 \cdot 10^{-5}$	$2.6 \cdot 10^{-2}$	3.4
8	$1.3 \cdot 10^{-3}$	$7.7 \cdot 10^{-3}$	2.0	$1.2 \cdot 10^{-6}$	$7.2 \cdot 10^{-3}$	3.3	$1.2 \cdot 10^{-6}$	$1.3 \cdot 10^{-2}$	3.2
16	$3.4 \cdot 10^{-4}$	$3.9 \cdot 10^{-3}$	2.0	$1.2 \cdot 10^{-7}$	$3.6 \cdot 10^{-3}$	3.2	$1.4 \cdot 10^{-7}$	$6.7 \cdot 10^{-3}$	3.0

TABLE 1. \mathcal{P}_3 results

We focus here on the complexity of the algorithm, i.e the number of operations during the simulations, rather than on the CPU time of the numerical code. We believe that the first criterion is more objective than

FIGURE 2. \mathcal{L}^2 -error with \mathcal{P}_4 -elements

Leap-Frog				Nabla-3			DG-ADER		
δ	Error	Time Step	CV	Error	Time Step	CV	Error	Time Step	CV
1	$4.0 \cdot 10^{-2}$	$4.2 \cdot 10^{-2}$		$1.0 \cdot 10^{-4}$	$4.0 \cdot 10^{-2}$		$1.1 \cdot 10^{-4}$	$7.2 \cdot 10^{-2}$	
2	$1.0 \cdot 10^{-2}$	$2.1 \cdot 10^{-2}$	2.0	$2.7 \cdot 10^{-6}$	$2.0 \cdot 10^{-2}$	5.2	$4.9 \cdot 10^{-6}$	$3.6 \cdot 10^{-2}$	4.6
4	$2.5 \cdot 10^{-3}$	$1.0 \cdot 10^{-2}$	2.0	$8.4 \cdot 10^{-8}$	$1.0 \cdot 10^{-2}$	5.0	$2.4 \cdot 10^{-7}$	$1.8 \cdot 10^{-2}$	4.3
8	$6.2 \cdot 10^{-4}$	$5.3 \cdot 10^{-3}$	2.0	$2.8 \cdot 10^{-9}$	$5.0 \cdot 10^{-3}$	4.9	$1.4 \cdot 10^{-8}$	$9.1 \cdot 10^{-3}$	4.1
16	$1.5 \cdot 10^{-4}$	$2.6 \cdot 10^{-3}$	2.0	$1.1 \cdot 10^{-10}$	$2.5 \cdot 10^{-3}$	4.7	$8.7 \cdot 10^{-10}$	$4.5 \cdot 10^{-3}$	4.0

TABLE 2. \mathcal{P}_4 resultsFIGURE 3. \mathcal{L}^2 -error with \mathcal{P}_5 -elements

the second one, which is sensitive to many parameters, such as the architecture of the platform, the type of parallelism (MPI, OpenMP,...) ... The complexity depends on the number of non zero elements of the stiffness matrices and on the number of time steps. Let us denote by N the number of triangles of the mesh and by n the number of degrees of freedom of one element. We have $N = 2 \left(\frac{1000}{h} \right)^2$ and $n = \frac{(p+1)(p+2)}{2}$. The size of the stiffness matrices is $n^2 N \times n^2 N$ for \mathcal{A}_p and \mathcal{B}_p and $n^2 N \times 2n^2 N$ for \mathcal{A}_v and \mathcal{B}_v . The number of non zero

Leap-Frog				Nabla-3			DG-ADER		
δ	Error	Time Step	CV	Error	Time Step	CV	Error	Time Step	CV
1	$2.0 \cdot 10^{-2}$	$3.0 \cdot 10^{-2}$	1	$5.7 \cdot 10^{-6}$	$2.8 \cdot 10^{-2}$	1	$1.5 \cdot 10^{-5}$	$5.2 \cdot 10^{-2}$	1
2	$5.1 \cdot 10^{-3}$	$1.5 \cdot 10^{-2}$	4.0	$1.3 \cdot 10^{-7}$	$1.4 \cdot 10^{-2}$	5.4	$9.2 \cdot 10^{-7}$	$2.6 \cdot 10^{-2}$	4.1
4	$1.2 \cdot 10^{-3}$	$7.5 \cdot 10^{-3}$	4.0	$5.5 \cdot 10^{-9}$	$7.0 \cdot 10^{-3}$	4.6	$5.7 \cdot 10^{-8}$	$1.3 \cdot 10^{-3}$	4.0
8	$3.2 \cdot 10^{-4}$	$3.7 \cdot 10^{-3}$	4.0	$3.1 \cdot 10^{-10}$	$3.5 \cdot 10^{-3}$	4.1	$3.6 \cdot 10^{-9}$	$6.5 \cdot 10^{-3}$	4.0
16	$8.0 \cdot 10^{-5}$	$1.9 \cdot 10^{-3}$	4.0	$1.9 \cdot 10^{-11}$	$1.7 \cdot 10^{-3}$	4.0	$2.2 \cdot 10^{-10}$	$3.2 \cdot 10^{-3}$	4.0

TABLE 3. \mathcal{P}_5 results

elements is $4N \times 2n^2$. The factor 4 comes from the fact that one element communicates with itself and its three neighbors, and the factor 2 corresponds to the two components of the velocity.

Hence the cost of one iteration of the LF and the Nabla-p scheme is approximately $2 \times 8Nn^2$ (two multiplications by the stiffness matrices at each iteration). While the cost of one iteration of DG-ADER scheme is $6 \times 8Nn^2$ (six multiplications by the stiffness matrices at each iteration). The global cost of the computation is obtained by multiplying the previous results by the number of time steps which is given by $60/\Delta t$.

For instance, the numerical costs for reaching an accuracy of 10^{-3} (0.1%) are $1.2 \cdot 10^{11}$ for LF- \mathcal{P}_3 , $2.5 \cdot 10^8$ for Nabla-3- \mathcal{P}_3 and 4.10^8 for DG-ADER- \mathcal{P}_3 . To reach an accuracy of 10^{-4} (0.01%) the numerical costs are $3.2 \cdot 10^{11}$ for LF- \mathcal{P}_4 , 2.10^9 for Nabla-3- \mathcal{P}_3 , $8.1 \cdot 10^8$ for Nabla-3- \mathcal{P}_4 and $3.2 \cdot 10^9$ for DG-ADER- \mathcal{P}_3 and $1.3 \cdot 10^9$ for DG-ADER- \mathcal{P}_4 . These results indicate that the Nabla-3 scheme generates less computational costs than LF (100 times smaller) and DG-ADER (2 times smaller). This is more significant for higher accuracy levels. The Nabla reaches an accuracy of 10^{-10} for a cost of $3.3 \cdot 10^{12}$ (\mathcal{P}_4) or $1.2 \cdot 10^{12}$ (\mathcal{P}_5) while DG-ADER requires $1.5 \cdot 10^{13}$ (\mathcal{P}_5).

CONCLUSION

We have proposed a new high-order time scheme, the Nabla-p scheme, which involves high-order space operators. These operators are easily discretized using DG methods and the cost of one time iteration is the same as the LF scheme. Numerical results show that, for a given accuracy, Nabla-p scheme generates less computational burden than LF or DG-ADER schemes, especially when using high-order polynomials for the space discretization.

REFERENCES

- [1] C. Agut, J. Diaz and A. Ezziani (2012) *High-Order Schemes Combining the modified Equation Approach and Discontinuous Galerkin Approximations for the Wave Equation*, Commun. Comput. Phys., 11 (2012), pp. 691-708.
- [2] S. Delcourte, L. Fezoui and N. Glinsky-Olivier (2009) *A high-order discontinuous Galerkin method for the seismic wave propagation*, ESAIM.
- [3] M. Dumbser and M. Käser (2000) *An arbitrary High-Order discontinuous Galerkin method for elastic waves on unstructured meshes. The two-dimensional isotropic case with external source terms*, Geophysical J. Int., Vol. 142, pp. 000-000.
- [4] M. A. Dablain (1986) *The application of high order differencing for the scalar wave equation*, Geophysics, Vol. 1, pp. 51:54-56.
- [5] G. R. Shubin and J. B. Bell (1987) *A modified equation approach to construct a fourth-orders methods for acoustic wave propagation*, SIAM J. Sci. Statist. Comput. 8:135-151.
- [6] H. Fahs and S. Lanteri (2010) *A high-order non-conforming discontinuous Galerkin method for time-domain electromagnetics* J. Comput. Appl. Math., Vol. 234, pp. 1088-1096
- [7] H. Fahs (2009) *High-order Leap-Frog based biscontinuous Galerkin bethod for the time-domain Maxwell equations on non-conforming simplicial meshes* Numer. Math. Theor. Meth. Appl., Vol. 2, No. 3, pp. 275-300