Adaptive time discontinuous Galerkin method for numerical modelling of wave propagation in shell and 3D structures

Bing Tie – Denis Aubry

Laboratory MSSMat (CNRS UMR 8579)
Ecole Centrale Paris
Grande Voie des Vignes
F-92295 Châtenay-Malabry cedex
{bing.tie, denis.aubry}@ecp.fr

ABSTRACT. *This paper presents an adaptive time discontinuous Galerkin method tailored to the numerical modelling of the wave propagation phenomena through shell and 3D structures. To achieve a reliable and efficient numerical implementation, several important computational issues concerning adaptive computation are discussed, namely the variable transfer between unmatched adaptively refined finite element meshes and the improvement of the convergence of the implicit dynamic solver by using a frequency dependent relaxation coefficient. Numerical examples of large-sized engineering structures are given to illustrate the interest and efficiency of the presented method.*

RÉSUMÉ. *Cet article présente une méthode adaptative de Galerkin discontinue en temps, destinée à la modélisation numérique des phénomènes de propagation d'ondes dans des structures coques ou tridimensionnelles. Dans l'objectif d'une implémentation numérique fiable et efficace de la méthode, plusieurs aspects importants sont discutés, notamment le transfert des variables entre des maillages adaptatifs non compatibles et l'amélioration de la convergence du solver dynamic implicite par l'utilisation d'un coefficient de relaxation dépendant de la fréquence. Des exemples numériques de structures industrielles de grande taille sont présentés afin d'illustrer l'intérêt et l'efficacité de la méthode.*

KEYWORDS: *space-time formulation, time discontinuous Galerkin (DG) method, elastic wave propagation, adaptive computing, implicit solver.*

MOTS-CLÉS : *formulation espace-temps, méthode de Galerkin discontinue en temps, propagation des ondes élastiques, calcul adaptatif, solver implicite.*

REMN – 15/2006. Space or/and time adaptive strategies, pages 729 to 757

1. Introduction

The numerical modelling of wave propagation phenomena through engineering structures still remains a challenging issue, especially when medium or high frequency ranges are studied. In such case, very small finite elements and time discretization steps, compared to the dimensions of engineering structures and the time interval of analysis, are required to capture wave fronts moving throughout the whole structure. Therefore, the use of an uniformly refined mesh generally results in huge CPU and memory consuming computations or even prohibitive problem sizes. To obtain reliable modelling of wave phenomena with an optimal size of the numerical model, the adaptive computing, which has been successfully applied to static linear or non linear problems, is nowadays more and more considered as a promising way to handle this challenge. An efficient adaptive method should be able to automatically concentrate numerical efforts by mesh refining around the wave fronts and coarsening away from the wave fronts, which are generally unknown beforehand.

Adaptive methods introduce a particular feature that is not standard for the structural elastodynamics computing, for which the classical approach consists in doing the time discretization by finite difference schemes and the space discretization by a fixed spatial finite element mesh. With adaptive methods, the discretisation of the studied structure changes in the time to follow the wave propagation. As a consequence, the mechanical fields should be transferred between adaptively evolving meshes while inappropriate transfers can lead to unstable computing. For this reason, there has been growing interests in the space-time Discontinuous Galerkin (DG) finite element method during the last few years, as far as it provides a suitable frameworks for implementing adaptive dynamic solvers by correctly managing the mesh evolution in time to ensure the stability in a natural way. Besides, the space-time DG method is also proved to be more accurate than the conventional Newmark family time-stepping schemes and its numerical built-in damping allows filtering high frequency numerical noises and giving proper representation of wave fronts.

The purpose of this paper is to present a two-field adaptive time DG method, which is based on the two-fields space-time DG formulations ((Hulbert *et al.*, 1988), (Li *et al.*, 1998), (Tie *et al.*, 2003)). By the choice of adopted space-time elements, the method can be finally written as a time-stepping scheme and only spatial FE meshes are adaptively changed during the dynamic computing ((Wiberg *et al.*, 1999), (Leclère, 2001), (Tie *et al.*, 2003)). The adaptive remeshing procedure is driven by *a posteriori* error estimates that evaluate local unbalanced dynamic forces coming from the FE solutions. Thus it belongs to the class of residual methods well known for the *a posteriori* error estimates ((Babuska *et al.*, 1978), (Oden *et al.*, 1989), (Bank *et al.*, 1992), (Johnson *et al.*, 1992), (Aubry *et al.*, 1999), (Tie *et al.*, 2003)). Data coming from theoretical analysis of elastic wave propagation are also used to define remeshing size maps. While this method has already been successfully applied to

analyse wave propagation phenomena (Tie *et al.*, 2003), in this paper, we consider more particularly its application to large-sized engineering shell and 3D structures. To achieve a reliable and efficient numerical implementation, two important computational issues are considered: the variable transfer between unmatched adaptively refined finite element meshes and the improvement of the convergence of the implicit dynamic solver by using a frequency dependent relaxation coefficient. Numerical examples of large-sized engineering structures are given to illustrate the interest and efficiency of the presented method.

The remainder of the paper is organised in the following way: at first a brief introduction of the time discontinuous Galerkin method is given in Section 2. Section 3 presents the adaptive remeshing strategy adopted in our work. Section 4 discusses the important issue of the variables transfer within the framework of the adaptive computing. An implicit bloc Gauss-Seidel DG solver with improved convergence rate is proposed in Section 5. Finally, numerical examples showing the wave propagation through large-sized engineering structures are presented in Section 6, followed by concluding remarks.

2. Two fields time discontinuous Galerkin method

We consider an elastic structure Ω that is submitted to external dynamic body and boundary forces $\boldsymbol{f}$ and $\boldsymbol{g}$. During the time interval of analysis $[0, T]$, the dynamic equilibrium of Ω is governed by the following partial differential equation:

$$\mathbf{Div}\boldsymbol{\sigma}(\mathbf{u}) + \boldsymbol{f} = \rho\dot{\boldsymbol{v}} \quad \text{in} \quad \Omega\times]0,T[\qquad [1]$$

and the following boundary and initial conditions:

$$\begin{array}{rcll} \mathbf{u} & = & \mathbf{0} & \text{in} \quad \Gamma_u\times]0,T[\\ \boldsymbol{\sigma}(\mathbf{u}).\boldsymbol{n} & = & \boldsymbol{g} & \text{in} \quad \Gamma_\sigma\times]0,T[\\ \mathbf{u}(\boldsymbol{x},0) & = & \mathbf{u}_I & \text{in} \quad \Omega \\ \boldsymbol{v}(\boldsymbol{x},0) & = & \boldsymbol{v}_I & \text{in} \quad \Omega \end{array} \qquad [2]$$

Here, $\mathbf{u}(\boldsymbol{x},t)$ and $\boldsymbol{v}(\boldsymbol{x},t)$ are respectively the displacement and velocity fields defined in $\Omega\times]0,T[$, a superposed dot indicates partial differentiation with respect to time t, $\mathbf{u}_I$ and $\boldsymbol{v}_I$ are respectively the initial displacement and velocity, $\boldsymbol{\sigma}$ is the Cauchy stress tensor, ρ is the mass density, $\Gamma = \Gamma_\sigma \cup \Gamma_u$ with $\Gamma_\sigma \cap \Gamma_u = \emptyset$ is the boundary of Ω and $\boldsymbol{n}$ is the unit outward normal to Γ. In this paper, homogeneous Dirichlet boundary conditions are assumed without lose of generality of our purpose. Under the hypothesis of small deformations, the Cauchy stress tensor $\boldsymbol{\sigma}$ is given from the infinitesimal strain tensor $\boldsymbol{\epsilon}$ by the generalized Hooke's law:

$$\boldsymbol{\sigma}(\mathbf{u}) = \boldsymbol{C} : \boldsymbol{\epsilon}(\mathbf{u}) \qquad [3]$$

where $\boldsymbol{C}$ is the fourth order elasticity tensor. Several formulations exist for the space-time DG method (Hulbert *et al.*, 1988). Herein, a two-fields formulation is considered

((Johnson *et al.*, 1992), (Li *et al.*, 1998), (Tie *et al.*, 2003)). Both the displacement $\mathbf{u}$ and the velocity $\boldsymbol{v}$ fields are considered as primary unknowns and the following compatibility equation between $\mathbf{u}$ and $\boldsymbol{v}$ is added to complete the two-fields system:

$$\mathbf{Div}(\boldsymbol{\sigma}(\dot{\mathbf{u}} - \boldsymbol{v})) = \mathbf{0} \qquad [4]$$

We need also to write appropriate boundary and initial conditions for the field $\boldsymbol{v}$:

$$\begin{array}{rcl} \dot{\mathbf{u}} - \boldsymbol{v} & = & \mathbf{0} \quad \text{on} \quad \Gamma_u \times]0,T[\\ \boldsymbol{\sigma}(\dot{\mathbf{u}} - \boldsymbol{v}).\boldsymbol{n} & = & \mathbf{0} \quad \text{on} \quad \Gamma_\sigma \times]0,T[\end{array} \qquad [5]$$

Now to establish the associated weak formulation, the integration of virtual works is made in the whole space-time domain $S = \Omega \times]0,T[$, which is subdivided into N space-time slabs: $S_n = \Omega \times]t_n, t_{n+1}[$. Between two successive space-time slabs, the primary unknowns $\mathbf{u}$ and $\boldsymbol{v}$ can be discontinuous. Then, the weak formulation in each space-time slab S_n, expressing the dynamic equilibrium and the displacement-velocity compatibility reads as:

$$\begin{array}{l} (\rho\dot{\boldsymbol{v}}, \boldsymbol{w}_v)_{S_n} + (\boldsymbol{\sigma}(\mathbf{u}), \boldsymbol{\epsilon}(\boldsymbol{w}_v))_{S_n} + (\rho[\boldsymbol{v}(t_n)], \boldsymbol{w}_v(t_n^+))_\Omega \\ = (\boldsymbol{f}, \boldsymbol{w}_v)_{S_n} + (\boldsymbol{g}, \boldsymbol{w}_v)_{\Gamma_\sigma \times]t_n, t_{n+1}[} \\ \\ (\boldsymbol{\sigma}(\dot{\mathbf{u}} - \boldsymbol{v}), \boldsymbol{\epsilon}(\boldsymbol{w}_u))_{S_n} + (\boldsymbol{\sigma}([\mathbf{u}(t_n)]), \boldsymbol{\epsilon}(\boldsymbol{w}_u(t_n^+)))_\Omega = 0 \end{array} \qquad [6]$$

where $(\boldsymbol{w}_u, \boldsymbol{w}_v)$ denote the virtual space-time test functions, $[\bullet(t_n)] = \bullet(t_n^+) - \bullet(t_n^-)$ the jump quantities in time at t_n, $(\bullet, \bullet)_D$ the integration over the space-time or space domain D.

The consistency of the two-fields time-discontinuous Galerkin formulation [6] is straightforward. Furthermore, its unconditional stability feature can be easily proven by choosing $\boldsymbol{w}_v = \boldsymbol{v}$ and $\boldsymbol{w}_u = \mathbf{u}$. We remark that to guarantee such a stability feature, it is essential to choose an appropriate operator to impose the compatibility condition [4] and to add terms dealing with the time discontinuities $[\boldsymbol{v}(t_n)]$ and $[\mathbf{u}(t_n)]$.

As far as the numerical discretization of the weak formulation [6] is concerned, each space-time slab S_n can be discretized using a completely free space-time FE mesh. However, the choice adopted herein is to use a structured mesh that is the combination of one linear or quadratic finite element in time and a free FE mesh in space. As the use of a quadratic element in time has not been shown to be especially advantageous with respect to the use of a linear one, all computations presented in this paper are made with a linear element in time:

$$\begin{array}{rcl} \mathbf{u}_{\Delta t_n}(x,t) & = & \dfrac{t_{n+1} - t}{\Delta t_n}\mathbf{u}_n(x) + \dfrac{t - t_n}{\Delta t_n}\mathbf{u}_{n+1}(x) \\ \boldsymbol{v}_{\Delta t_n}(x,t) & = & \dfrac{t_{n+1} - t}{\Delta t_n}\boldsymbol{v}_n(x) + \dfrac{t - t_n}{\Delta t_n}\boldsymbol{v}_{n+1}(x) \end{array} \qquad [7]$$

where $\Delta t_n = t_{n+1} - t_n$ is the nth time step, $(\mathbf{u}_n, \mathbf{u}_{n+1}, \boldsymbol{v}_n, \boldsymbol{v}_{n+1})$ are respectively the approximated solutions of $(\mathbf{u}(t_n^+), \mathbf{u}(t_{n+1}^-), \boldsymbol{v}(t_n^+), \boldsymbol{v}(t_{n+1}^-))$, which are continuous in space. Then after time integration, we get the following system to solve for velocities $(\boldsymbol{v}_n, \boldsymbol{v}_{n+1})$.

$$\begin{aligned}
&(\rho \boldsymbol{v}_n, \boldsymbol{w})_\Omega + \frac{\Delta t_n^2}{6}(\boldsymbol{\sigma}(\boldsymbol{v}_n), \boldsymbol{\epsilon}(\boldsymbol{w}))_\Omega + \frac{2}{3}(\rho \boldsymbol{v}_{n+1}, \boldsymbol{w})_\Omega \\
&= \boldsymbol{f}_1^{ext}(\boldsymbol{w}) + \frac{5}{3}(\rho \boldsymbol{v}_n^-, \boldsymbol{w})_\Omega - \frac{2\Delta t_n}{3}(\boldsymbol{\sigma}(\mathbf{u}_n^-), \boldsymbol{\epsilon}(\boldsymbol{w}))_\Omega \\
&(\rho \boldsymbol{v}_{n+1}, \boldsymbol{w})_\Omega + \frac{\Delta t_n^2}{6}(\boldsymbol{\sigma}(\boldsymbol{v}_{n+1}), \boldsymbol{\epsilon}(\boldsymbol{w}))_\Omega + \frac{\Delta t_n^2}{3}(\boldsymbol{\sigma}(\boldsymbol{v}_n), \boldsymbol{\epsilon}(\boldsymbol{w}))_\Omega \\
&= \boldsymbol{f}_2^{ext}(\boldsymbol{w}) + (\rho \boldsymbol{v}_n^-, \boldsymbol{w})_\Omega - \Delta t_n(\boldsymbol{\sigma}(\mathbf{u}_n^-), \boldsymbol{\epsilon}(\boldsymbol{w}))_\Omega
\end{aligned} \quad [8]$$

where $\boldsymbol{w}$ denotes the virtual spatial test functions, $(\mathbf{u}_n^-, \boldsymbol{v}_n^-)$ the approximated solutions of $(\mathbf{u}(t_n^-), \boldsymbol{v}(t_n^-))$ at the end of previous space-time slab S_{n-1} and $(\boldsymbol{f}_1^{ext}, \boldsymbol{f}_2^{ext})$ the external loading terms after the time integration. Once the velocity unknowns $(\boldsymbol{v}_n, \boldsymbol{v}_{n+1})$ are calculated, the displacement unknowns $(\mathbf{u}_n, \mathbf{u}_{n+1})$ are updated:

$$\begin{aligned}
(\boldsymbol{\sigma}(\mathbf{u}_n), \boldsymbol{\epsilon}(\boldsymbol{w}))_\Omega &= (\boldsymbol{\sigma}(\mathbf{u}_n^-), \boldsymbol{\epsilon}(\boldsymbol{w}))_\Omega - \frac{\Delta t_n}{6}(\boldsymbol{\sigma}(\boldsymbol{v}_{n+1} - \boldsymbol{v}_n), \boldsymbol{\epsilon}(\boldsymbol{w}))_\Omega \\
(\boldsymbol{\sigma}(\mathbf{u}_{n+1}), \boldsymbol{\epsilon}(\boldsymbol{w}))_\Omega &= (\boldsymbol{\sigma}(\mathbf{u}_n^-), \boldsymbol{\epsilon}(\boldsymbol{w}))_\Omega + \frac{\Delta t_n}{2}(\boldsymbol{\sigma}(\boldsymbol{v}_{n+1} - \boldsymbol{v}_n), \boldsymbol{\epsilon}(\boldsymbol{w}))_\Omega
\end{aligned} \quad [9]$$

Let $\mathcal{M}_n$ denote the spatial mesh for the space-time slab S_n and $(U_n, U_{n+1}, V_n, V_{n+1})$ the vectors of nodal values of the FE solutions $(\mathbf{u}_{h,n}, \mathbf{u}_{h,n+1}, \boldsymbol{v}_{h,n}, \boldsymbol{v}_{h,n+1})$ on $\mathcal{M}_n$ of $(\mathbf{u}_n, \mathbf{u}_{n+1}, \boldsymbol{v}_n, \boldsymbol{v}_{n+1})$, then the weak form [6] is finally recast into the following matrix form:

$$\begin{bmatrix} M_{nn}^* & \frac{2}{3}M_{nn} \\ \frac{\Delta t_n^2}{3}K_{nn} & M_{nn}^* \end{bmatrix} \begin{Bmatrix} V_n \\ V_{n+1} \end{Bmatrix} = \begin{Bmatrix} \bar{F}_n(\boldsymbol{f}, \boldsymbol{g}, U_n^-, V_n^-) \\ \bar{F}_{n+1}(\boldsymbol{f}, \boldsymbol{g}, U_n^-, V_n^-) \end{Bmatrix} \quad [10]$$

with:

$$\begin{aligned}
M_{nn}^* &= M_{nn} + \tfrac{\Delta t_n^2}{6}K_{nn} \\
\bar{F}_n(\boldsymbol{f}, \boldsymbol{g}, U_n^-, V_n^-) &= F_d(\boldsymbol{f}, \boldsymbol{g}) + \frac{5}{3}M_{n,n-1}V_n^- - \frac{2\Delta t_n}{3}K_{n,n-1}U_n^- \\
\bar{F}_{n+1}(\boldsymbol{f}, \boldsymbol{g}, U_n^-, V_n^-) &= F_s(\boldsymbol{f}, \boldsymbol{g}) + M_{n,n-1}V_n^- - \Delta t_n K_{n,n-1}U_n^-
\end{aligned} \quad [11]$$

where K_{nn} and M_{nn} are respectively the usual stiffness and mass matrices defined on the spatial FE mesh $\mathcal{M}_n$, $K_{n,n-1}$ and $M_{n,n-1}$ are respectively the stiffness and

mass matrices coupling the two spatial FE meshes $\mathcal{M}_{n-1}$ and $\mathcal{M}_n$ of two successive space-time slabs S_n and S_{n-1}. For example, the inner product term $K_{n,n-1}U_n^-$ projects the displacement $\mathbf{u}_n^-$ from $\mathcal{M}_{n-1}$ onto $\mathcal{M}_n$. For simplicity concern, the detailed formulations of the loading vectors $(F_d(\boldsymbol{f},\boldsymbol{g}), F_s(\boldsymbol{f},\boldsymbol{g}))$ depending upon the external loads $(\boldsymbol{f},\boldsymbol{g})$ are not given here.

Once the velocity unknowns (V_n, V_{n+1}) are solved, the displacement unknowns (U_n, U_{n+1}) are updated using [9], which reads in following matrix form:

$$\begin{aligned} U_n &= K_{nn}^{-1}K_{n,n-1}U_n^- - \frac{\Delta t_n}{6}(V_{n+1}-V_n) \\ U_{n+1} &= K_{nn}^{-1}K_{n,n-1}U_n^- + \frac{\Delta t_n}{2}(V_{n+1}+V_n) \end{aligned} \qquad [12]$$

As the matrix of the linear system of Equations [10] is full and not symmetric, it is preferable to solve it using iterative algorithms by which only the matrix M_{nn}^* is computed and factorized while the extra-diagonal matrices are put into the right-hand side. Obviously, as any other implicit solver, the two-fields time-discontinous Galerkin solver is memory and computing time consuming. Furthermore in the case of adaptive computing, the update of the displacement fields (U_n, U_{n+1}) using [12] requires also the factorisation of the stiffness matrix K_{nn} and so doubles the CPU and memory needs. Therefore, to improve the performance of the adaptive implicit DG solver, two important issues are considered in this paper: improvement of the convergence rate of the implicit iterative DG solver; interpolation operator between unmatched adaptive spatial meshes.

3. Adaptive remeshing stategy

The adaptive strategy is defined in the following way: at each time step, the spatial FE mesh used at the previous time step is taken as the initial mesh on which the first FE solutions are computed. Then a size map is defined with respect to these FE solutions. According to the size mesh, a new spatial mesh is built by a whole adaptive remeshing of the studied structure if it is necessary. At most one mesh adaption is performed on each time step and this is sufficient because small time step is used to avoid large discretisation errors in time.

To define the size map for the mesh adaption, local error indicators are calculated as the main purpose of the adaptive computing is to achieve a required accuracy in FE solutions with an optimally sized FE mesh. Several classes of methods for the *a posteriori* error estimates have been developed in the literature for static problems, namely the so-called Z2-method proposed by Zienkiewicz *et al.* (1992) based on the post stress smoothing processes, the method proposed by Ladevèze (1983) that explicitly builds static admissible stress fields and the residual forces method, which investigates the unbalanced residuals using enriched finite elements bases.

In this work, two method of computing local error indicators are considered. The first one is the residual forces method for which the dynamic unbalanced residuals are evaluated by solving local residual problems defined on local patches of hierarchical refined finite elements (see (Tie *et al.*, 2003) for more details of its numerical implementation). The other less expansive one simply calculates the time discontinuities of displacement in elastic energy norm and the time discontinuities of velocity in kinetic energy norm at the beginning of each space-time slab t_n. Then the local error indicators are used in the following way: the comparison of the sum of local error indicators with respect to a user prescribed threshold first tells whether the mesh adaption should be performed or not. In the former case, the error map is used to define a size map of a new mesh. The size map is defined in such a way that a nearly homogeneous error distribution is achieved over all the elements on the new mesh. According to the principle of the equi-distribution of error, the so-built new mesh is nearly optimal.

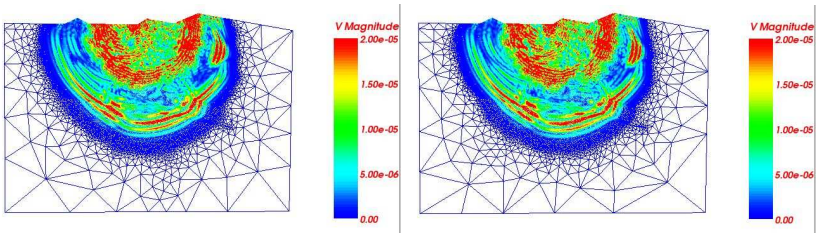


Figure 1. *2D adaptive computing of the propagation of P-waves and S-waves in foothills areas (see Section 6.1 for the definition of the foothills areas). Are presented for a time step: left, adaptive mesh built using error indicators calculated by the residual method; right, adaptive mesh built using error indicators calculated by evaluating the time jumps of* $(\mathbf{u}, \boldsymbol{v})$ *at the beginning of this time step*

These two methods generally give quantitatively different error indicators. Nevertheless the size maps defined by the two methods result in nearly identical adaptive meshes (Figures 1 and 2, see also Section 6.1 for the definition of the foothills example), because of the use of the following parameter together with the local error indicators to define the size maps. This parameter is the shortest wavelength $\lambda_{min}(f_{max})$ related to the highest frequency under study f_{max}. In the case of 2D, plate and 3D structures, $\lambda_{min}(f_{max})$ can be calculated by the theoretical analysis of the elastic wave propagation. Using $\lambda_{min}(f_{max})$, the adaptive strategy adopted in our work is defined in such a way that the adaptive remeshing systematically places N_{elt} elements within the shortest wavelength of interest around the wave fronts. Generally, we sug-

gest to choose $N_{elt} \in [4, 10]$ in numerical analyses. So the minimum size of space finite elements h_{min} around the wave fronts is calculated as follows:

$$h_{min} = \frac{\lambda_{min}}{N_{elt}} \qquad [13]$$

In the case of a heterogeneous structure composed of substructures having different mechanic properties, as in the case of multilayered foothills areas (Section 6.1), each substructure has its own minimum size of finite elements h_{min}.

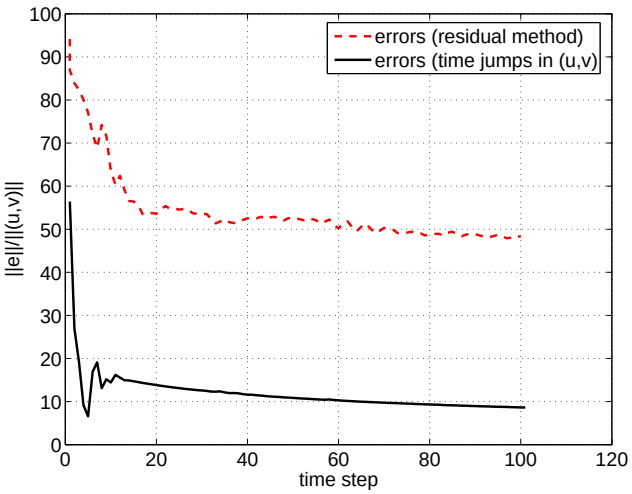


Figure 2. *2D adaptive computing of the propagation of P-waves and S-waves in foothills areas. Comparison of the global level of error indicators calculated using the two different methods: the residual method and the method evaluating the time jumps of* $(\mathbf{u}, \boldsymbol{v})$ *at the beginning of each time step*

In 2D or 3D structures, the slowest direct waves are the shear waves: $c_{min} = c_s$ and so the shortest wavelength is determined by the shear waves: $\lambda_{min} = \lambda_s = \frac{c_s}{f_{max}}$. In plate structures, the bending waves are dispersive: its velocity depend on the frequency, and the shortest wavelength is determined the bending waves: $\lambda_{min} = \lambda_f(f_{max})$. We remark that the bending wave velocity c_f depends not only upon the frequency f but also upon the shell thickness. For the same highest frequency under study f_{max}, the thinner is the plate, the smaller should be the finite elements. Theoretical wave analysis is necessary to find the dispersion equation, which can be explicitly written in the case of the Mindlin plate modeling (Leclère, 2001). In the case of curved shells, no analytical explicit dispersion equation is available and we have simply applied the discretization parameters $(\Delta t, h_{min})$ defined from the Mindlin plates to the shell modeling for the numerical results presented herein (Boullard, 2004).

Finally the time step Δt is chosen according to the propagation velocity of the fastest wave c_{max} (in general, it is determined by the pressure wave, $c_{max} = c_p$) and by the highest frequency of interest involved in the external loads, denoted by f_{load}, that should be taken into account:

$$\Delta t = min(\frac{1}{N f_{load}}, \frac{h_{min}}{c_{max}}) \quad [14]$$

In our numerical analyses, we usually take $N \in [10, 25]$ in order to ensure an accurate modelling of the external loads.

4. Interpolation issues of the adaptive time DG solvers

During the adaptive transient computing, unmatched FE meshes are successively generated, so mechanical fields, namely the displacement, the velocity and the stress fields, should be transferred from one mesh to the other. This transfer should be done using appropriate operators to avoid introduction of numerical errors. The choice of interpolation operators of our adaptive time DG solver is presented and discussed in this section.

We consider the space-time slab $S_n = \Omega \times]t_n, t_{n+1}[$ on which a new adaptive mesh $\mathcal{M}_n$ is built, while the mesh adaptively built for the previous space-time slab S_{n-1} is $\mathcal{M}_{n-1}$. Then the adaptive time DG solver has to deal with two kinds of variables transfer.

The first concerns the calculation of the velocity fields in the space-time slab S_n, more precisely the calculation of the terms $(\boldsymbol{\sigma}(\mathbf{u}_n^-), \boldsymbol{\epsilon}(\boldsymbol{w}))_\Omega$ and $(\rho \boldsymbol{v}_n^-, \boldsymbol{w})_\Omega$ of the right-hand side of the Equation [8]. The integration of these terms is computed at the Gauss quadrature points of the mesh $\mathcal{M}_n$. To compute the velocity term $(\rho \boldsymbol{v}_n^-, \boldsymbol{w})_\Omega$, the transfer is straightforward as $\boldsymbol{v}_n^-$ is known everywhere in the structure due to the FE interpolation operator $\mathcal{P}_{\mathcal{M}_{n-1}}$ defined on the mesh $\mathcal{M}_{n-1}$. To compute the displacement term $(\boldsymbol{\sigma}(\mathbf{u}_n^-), \boldsymbol{\epsilon}(\boldsymbol{w}))_\Omega$, the transfer is more tactful because inappropriate transfer can result in the diffusion of the wave fronts that are actually localised in space. For example it is well-known that the global smoothing operator using a least-squares method of the discontinuous FE stress field $\boldsymbol{\sigma}(\mathbf{u}_n^-)$ generally results in the diffusion of localised zones of large stresses (Hinton *et al.*, 1974). Hence our choice consists in using a local element-wise smoothing operator to have a stress field defined everywhere in each element of the mesh $\mathcal{M}_{n-1}$ (Aubry *et al.*, 2003). In fact in each element a polynomial stress field is locally built using a least squares method and according to the stress values obtained at Gauss points of the mesh $\mathcal{M}_{n-1}$. In this way, the localised character of the wave fronts is conserved. When linear finite elements are used, the stress field is constant in each element and no smoothing process is necessary.

The second kind of variables transfer concerns the update of the displacement after the calculation of the velocity using [9] or [12]. The calculation of the term $(\boldsymbol{\sigma}(\mathbf{u}_n^-), \boldsymbol{\epsilon}(\boldsymbol{w}))_\Omega$ being already presented, the update of the displacement requires therefore the calculation, the storage and factorisation of the stiffness matrix K_{nn} according to [12], which is memory and CPU consuming. A solution much less expensive, which furthermore works well, consists in updating the displacement in the following way:

$$\begin{aligned} U_n &= \mathcal{P}_{\mathcal{M}_{n-1}}(U_n^-) - \frac{\Delta t_n}{6}(V_{n+1} - V_n) \\ U_{n+1} &= \mathcal{P}_{\mathcal{M}_{n-1}}(U_n^-) + \frac{\Delta t_n}{2}(V_{n+1} + V_n) \end{aligned} \qquad [15]$$

where $\mathcal{P}_{\mathcal{M}_{n-1}}$ is simply the FE interpolation operator defined by the mesh $\mathcal{M}_{n-1}$.

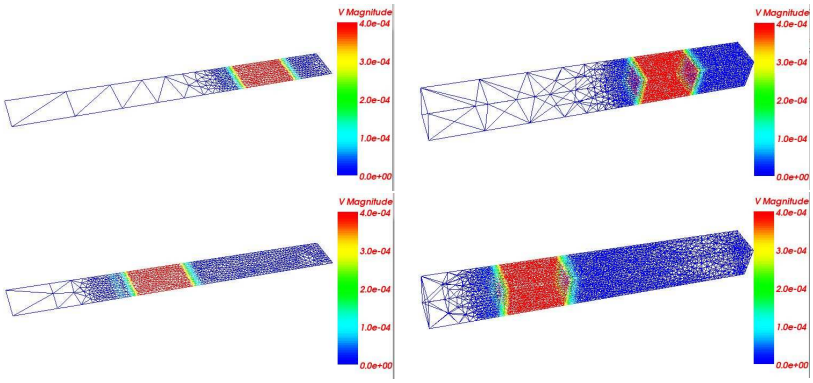


Figure 3. *Wave propagation in an elastic beam. Left, 2D adaptive computing; Right, 3D adaptive computing*

To illustrate the quality of the interpolation operators adopted in our adaptive time DG solver, we present here the 2D and 3D numerical modelling of the wave propagation in an elastic beam (Figure 3). Figures 4 and 5 present the comparison of accelerations obtained at a point by different modellings: the modelling with a uniformly refined mesh considered as a reference solution, the adaptive modelling for which the displacement update is done by solving the system [12] and the adaptive modelling for which the displacement update is done by using [15]. Good agreement is observed between the different modellings, which is generally observed by all our numerical computations. For the 2D analysis, two reference solutions are obtained one with a Q4 (four-nodes square element) mesh and the other with a T3 (three-nodes triangular element) mesh. It can be remarked that the observed small oscillations around the reference Q4 solution is rather caused by the use of T3 element than by

the interpolation process. For the 3D modelling, only tetra4 (four-nodes tetrahedral) elements are used.

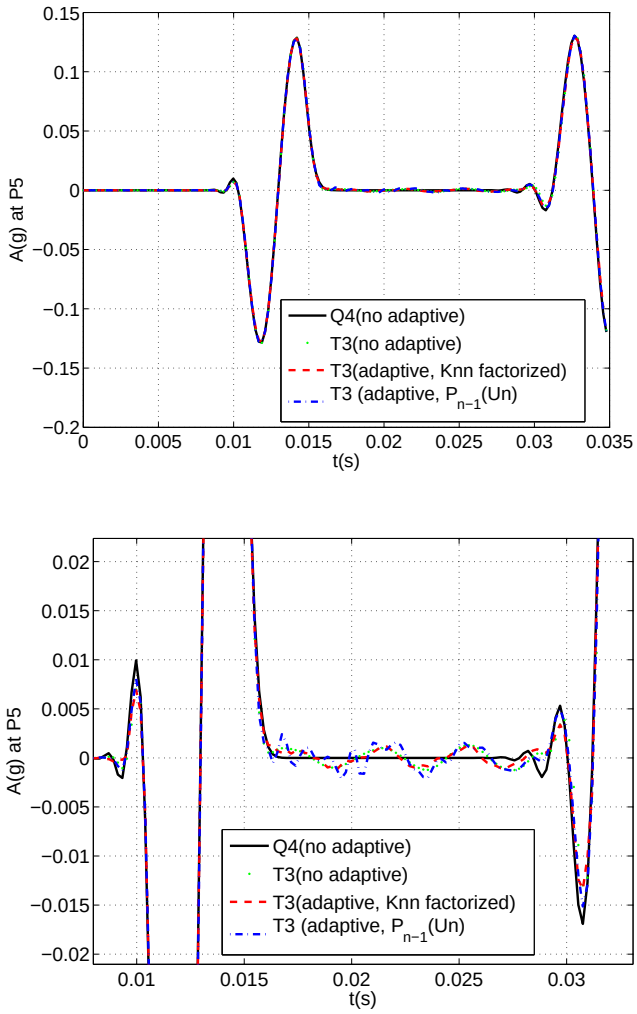


Figure 4. *Accelerations obtained at a point in the 2D beam by different modellings. Q4 (no adaptive): uniformly refined mesh with Q4 element; T3 (no adaptive): uniformly refined mesh with T3 elements; T3 (adaptive, K_{nn} factorized): Adaptive computing using T3 elements and the displacement update is done by solving the system [12]; T3 (adaptive, $P_{n-1}(U_n)$): Adaptive computing using T3 elements and the displacement update is done by using [15]*

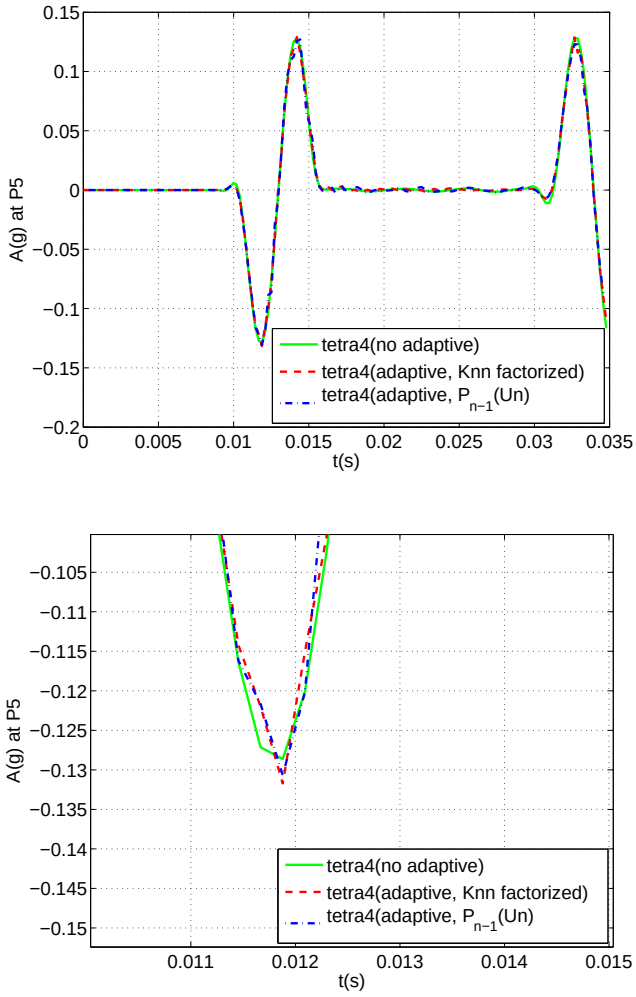


Figure 5. *Accelerations obtained at a point in the 3D beam by different modelling. tetra4 (no adaptive): uniformly refined mesh with tetra4 elements; tetra4 (adaptive, K_{nn} factorized): Adaptive computing using tetra4 elements and the displacement update is done by solving the system [12]; tetra4 (adaptive, $P_{n-1}(U_n)$): Adaptive computing using tetra4 elements and the displacement update is done by using [15]*

As far as the mesh adaption is concerned, we can remark that the FE mesh is adaptively refined around the propagating wave fronts but only slightly coarsened far away behind the wave fronts. Same observation is obtained by the numerical examples

presented in Section 6 and indicates that the mesh coarsening can be more delicate to manage than the mesh refining. In fact, the mesh coarsening should be done only if waves of the frequency range of interest have moved away, so that the accuracy of numerical modelling is maintained.

5. Implicit DG solver of Gauss-Seidel type with frequency-dependent relaxation

As M^* the matrix of the linear system of Equations [10] is full and not symmetric, it is preferable to solve it using iterative algorithms. The implicit DG solver proposed by Li *et al.* (1998) is based on such an iterative algorithm, which is actually a bloc Jacobi iterative algorithm without relaxation. We propose here a bloc Gauss-Seidel iterative algorithm with furthermore a frequency-dependent relaxation. We will prove that the frequency-dependent relaxation improves convergence rate of the iterative solver by uniformly reducing errors in the whole frequency domain of interest.

Hereafter two iterative algorithms to solve the system [10] are considered: the bloc Jacobi one et the bloc Gauss-Seidel one. Given an initial predictor of $V^0 = (V_n^0, V_{n+1}^0)$, the bloc Jacobi or Gauss-Seidel iterative algorithms reads as:

Jacobi algorithm, for $i \geq 0$:

$$\begin{aligned} V_{n+1}^{i+1} &= V_{n+1}^i + \alpha M_{nn}^{*-1}(\bar{F}_{n+1} - \frac{\Delta t^2}{3} K_{nn} V_n^i - M_{nn}^* V_{n+1}^i) \\ V_n^{i+1} &= V_n^i + \alpha M_{nn}^{*-1}(\bar{F}_n - \frac{2}{3} M_{nn} V_{n+1}^i - M_{nn}^* V_n^i) \end{aligned} \quad [16]$$

Gauss-Seidel algorithm, for $i \geq 0$:

$$\begin{aligned} V_{n+1}^{i+1} &= V_{n+1}^i + \alpha M_{nn}^{*-1}(\bar{F}_{n+1} - \frac{\Delta t^2}{3} K_{nn} V_n^i - M_{nn}^* V_{n+1}^i) \\ V_n^{i+1} &= V_n^i + \alpha M_{nn}^{*-1}(\bar{F}_n - \frac{2}{3} M_{nn} V_{n+1}^{i+1} - M_{nn}^* V_n^i) \end{aligned} \quad [17]$$

where α is the relaxation coefficient.

5.1. *Analysis of convergence*

The stability and convergence conditions of the time discontinuous DG solvers will be written in term of β, the eigenvalues of the matrix B^* defined as:

$$B^* = M_{nn}^{*-1} M_{nn} = I - \frac{\Delta t^2}{6} M_{nn}^{*-1} K_{nn} \quad [18]$$

It can be shown that $0 < \beta < 1$, as:

$$\beta = \frac{1}{\frac{(\Delta t \omega)^2}{6} + 1} \quad [19]$$

where ω^2 are the eigenvalues of the following eigensystem:

$$K_{nn}\Psi = \omega^2 M_{nn}\Psi \qquad [20]$$

where Ψ is the eigenvector corresponding to the eigenvalue ω^2. We recall that $\omega = 2\pi f$, f being the frequency.

It can be proved that both proposed solvers unconditionally converge and that, as can be expected, the bloc Gauss-Seidel DG solver converges more rapidly than the bloc Jacobi one. Furthermore, for the bloc Gauss-Seidel solver, an optimal relaxation coefficient is found, which is frequency-dependent and improves the convergence rate of the solver. These results are summarised in the following propositions, whose proofs are given in Appendix 9.1).

Proposition 5.1 *The implicit bloc Jacobi DG solver [16] unconditionally converges for* $\forall\alpha \in]0, 1.268[$. *The relaxation does not improve its convergence rate.*

Proposition 5.2 *The implicit bloc Gauss-Seidel DG solver [17] unconditionally converges for* $\forall\alpha \in]0, 2[$. *Without the relaxation (*$\alpha = 1$*), the implicit DG solver of Gauss-Seidel type converges more rapidly then the one of Jacobi type.*

Proposition 5.3 *There exists an optimal relaxation coefficient* $\alpha_{opt}(f)$ *that improves the convergence of the implicit bloc Gauss-Seidel DG solver in the whole frequency domain. It is frequency-dependent:* $\alpha_{opt}(f) \in [1, \alpha_{max}]$ *is the real root of the following quartic equation:*

$$\lambda_{A_{GS}}\alpha^4 - 4\alpha + 4 = 0 \qquad [21]$$

where $\lambda_{A_{GS}}$ *is the convergence rate, called also the spectral radius of the iterative matrix* A_{GS}*, of the implicit bloc Gauss-Seidel DG solver without relaxation.* α_{max} *is the optimal relaxation coefficient corresponding to the frequency value* $f = \frac{\sqrt{6}}{2\pi\Delta t_n}$*, for which* $\lambda_{A_{GS}}$ *reaches its maximal value (see [38]). The numerical value of* α_{max} *is nearly* 1.1415.

In Figure 6, we present the frequency dependent relaxation coefficient $\alpha_{opt}(f)$ and also the convergence rates of three different bloc Gauss-Seidel DG solvers: the one without relaxation ($\alpha = 1$), the one with the constant relaxation coefficient α_{max} and the one with the frequency dependent optimal relaxation coefficient $\alpha_{opt}(f)$. Hence, with $\alpha_{opt}(f)$, the convergence rate is improved in the whole frequency domain. With α_{max}, the convergence rate is improved for the frequencies around $f = \frac{\sqrt{6}}{2\pi\Delta t_n}$, determined by the time step Δt_n, and is deteriorated otherwise.

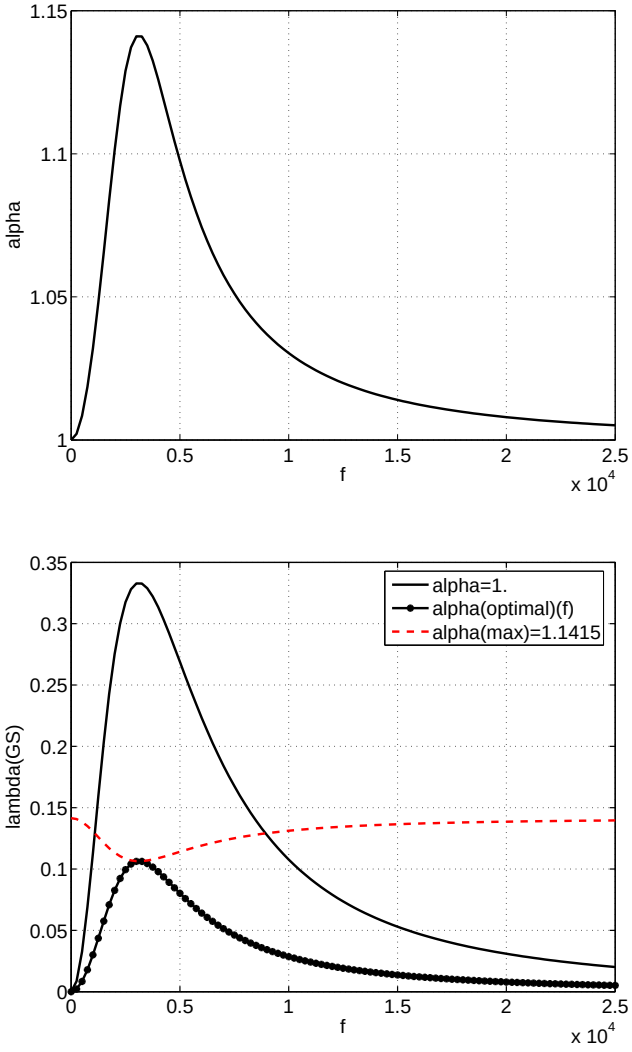


Figure 6. *Above, optimal relaxation coefficient $\alpha_{opt}(f)$ of the implicit bloc Gauss-Seidel DG solver that depends on the frequency f; Underneath, convergence ratio λ_{GS} of the implicit bloc Gauss-Seidel DG solver depending upon de choice of the relaxation coefficient α*

5.2. *Implicit bloc Gauss-Seidel DG solver with frequency-dependent relaxation for the adaptive computing*

During the adaptive computing, from one space-time slab to the other, the spatial FE mesh is adapted to follow the wave propagation. The time continuity of the primary unknowns $(\mathbf{u}, \boldsymbol{v})$ is weekly imposed between two unmatched FE meshes as it is allowed by the weak formulation of the time DG method. Therefore, all convergence analyses previously presented remain valid in the case of the adaptive computing.

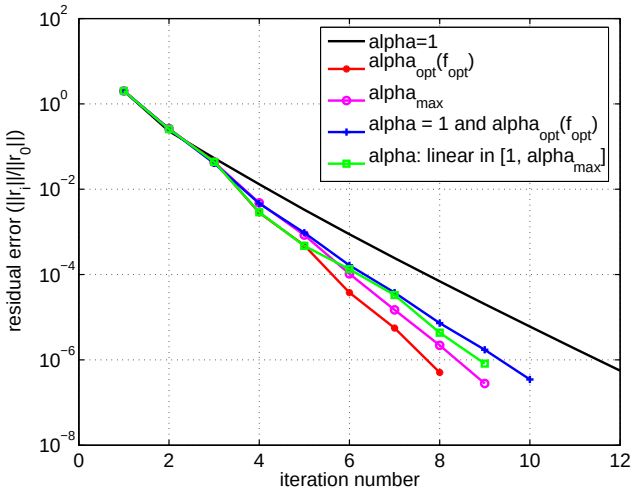
For the numerical computing, it is impossible to take a continuously frequency-dependent relaxation coefficient. The strategy that we propose here is to compute a relaxation coefficient, which is constant but depends on h_{min} the smallest element size of spatial FE mesh in the following way:

$$\alpha = \alpha_{opt}(f_{opt}) \text{ with: } f_{opt} = \frac{c_{max}}{N_{elt} h_{min}} \qquad [22]$$

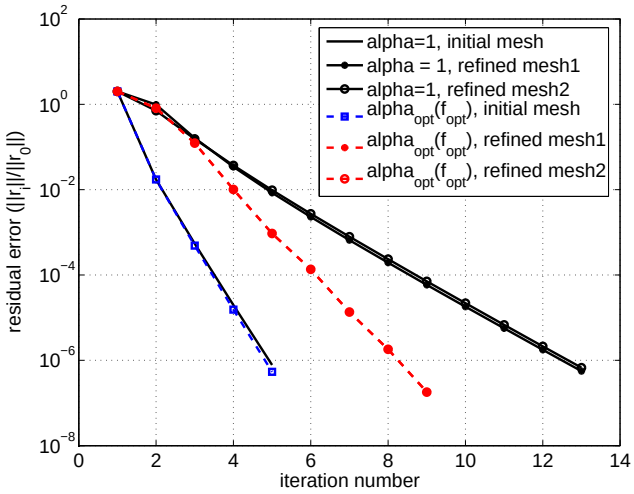
where c_{max} is the largest velocity of wave propagation under study. According to the strategy for choosing the space-time discretization parameters (see the formula [13]), f_{opt} is larger than the highest frequency of interest f_{max} that is linked to c_{min} the smallest velocity of waves. Therefore, the relaxation constant $\alpha_{opt}(f_{opt})$ is still frequency dependent, as it actually depends on the frequency range of interest of the studied problem.

Figure 7(a) compares, in the case of a 2D structure with a uniform FE mesh with about 100 thousands dofs (degrees of freedom), the numerical convergence of the Gauss-Seidel type DG solver with different choices of the relaxation coefficient α: (i) without relaxation $\alpha = 1$; (ii) with $\alpha = \alpha_{max}$; (iii) with $\alpha = \alpha_{opt}(f_{opt})$; (iv) with α alternately equals to 1 and $\alpha_{opt}(f_{opt})$; (v) with α takes linearly distributed values belonging to $[1, \alpha_{opt}(f_{opt})]$. All our numerical tests have shown that the previously proposed strategy [22] is the best one, 33 $\%$ reduction of total iteration number is systematically obtained.

In the case of the adaptive computing, the calculation generally starts on a coarse mesh where the implicit DG solver converges rapidly. But the convergence rate is deteriorated on adaptively refined meshes if no relaxation is made, as it is shown by Figure 7(b). Indeed, the refined meshes captured higher frequencies for which the convergence rate is worse, then if the relaxation coefficient is adaptively taken as function of the smallest element size of adaptively refined FE meshes, the convergence rate is improved for the higher frequency errors decrease more rapidly. Furthermore, such an optimal convergence rate is maintained for large sized systems. Figure 8 shows the convergence feature of the solver on a 3D refined mesh with more than 2 millions dofs and improvement of convergence obtained by the use of the relaxation on a shell mesh with more than 80 thousand dofs.



(a)



(b)

Figure 7. *(a) Numerical convergence of the implicit bloc Gauss-Seidel DG solver with different choices of the relaxation coefficients α; (b) Improvement of convergence of the implicit bloc Gauss-Seidel DG solver on adaptively refined meshes*

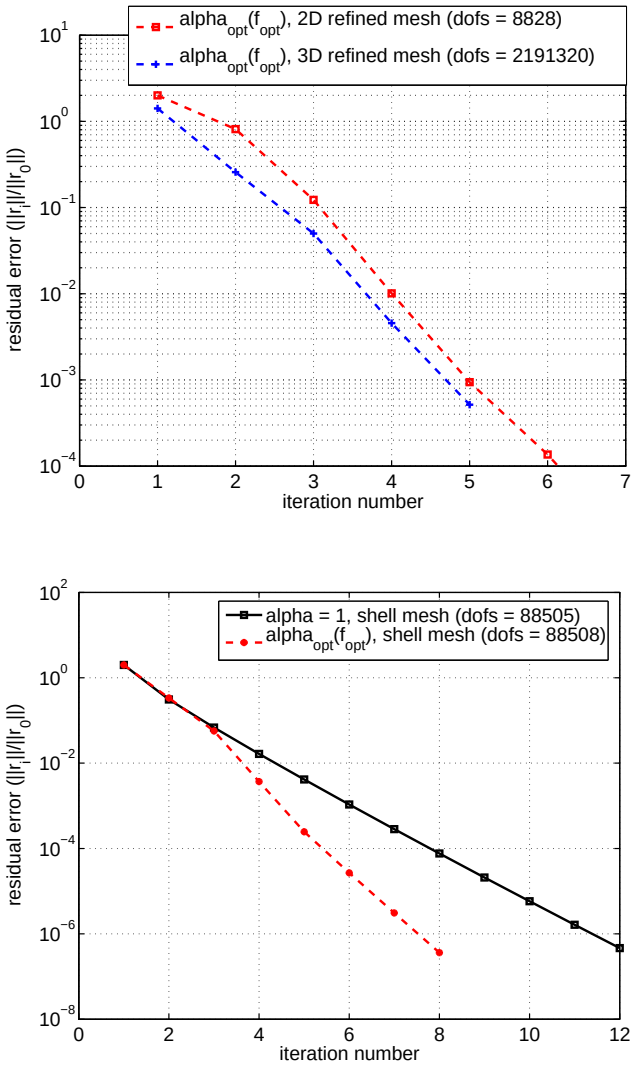


Figure 8. *Numerical convergence of the implicit bloc Gauss-Seidel DG solver with optimal relaxation coefficients for large sized systems. Above, 3D modelling; Underneath, shell modelling*

6. Numerical examples

Now the presented adaptive solver is applied to industrial problems and several numerical examples are presented to illustrate its efficiency.

6.1. *Elastic wave propagation in foothills areas*

For the oil industry, foothills areas that are nowadays poorly explored appear very attractive. However, the exploration in foothills areas remains challenging and among the difficulties to handle, the modelling of the wave propagation is a difficult problem to address due to the complexity of geologic framework of the foothills areas, namely the velocity contrast between the layers resulting in complex wave propagation patterns. Furthermore, the dimension of the studied domain is very large compared to the implied wavelengths. In our case, the size of the domain to be studied should be at least 100 times the smallest wavelength. Therefore a uniformly refined mesh that precisely describes the multi-layered system and finely captures wave propagation can be very expensive or even prohibitive.

Figure 9. *Foothills areas. Left, 2D uniformly refined mesh showing the velocity contrast between the layers, the velocity in near surface is lower than in the substratum; Right, 2D initial coarse mesh for the adaptive computing*

6.1.1. *2D modeling*

We present at first a 2D adaptive modelling of propagating pressure (P-) and shear (S-) waves in a foothills area. Its somewhat simplified geologic structure and the coarse initial mesh used to start the adaptive computing are shown in Figure 9. The external load, a Ricker-type signal whose frequency range is centred around $25Hz$, is vertically applied on a surface point. Figure 10 shows that the wave propagation patterns of P- and S-waves are captured by the adaptively refined meshes. The P-waves fronts are perturbed by the multi-layered system, and the S-waves are continuously triggered at the interfaces by the P-waves propagating more rapidly. By the automatic mesh adaption, larger finite elements are used in deeper layers because waves propagate more rapidly there. However, there is no mesh coarsening observed for this example because the whole domain is finally filled with wave fronts.

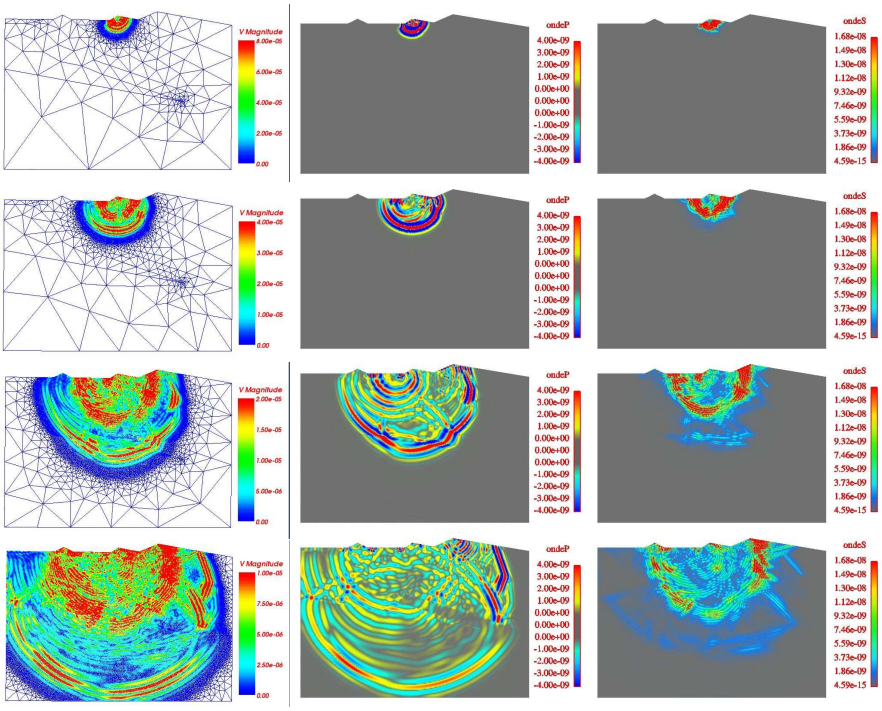


Figure 10. *2D adaptive computing of the propagation of P-waves and S-waves in foothills areas. Are presented: Left, adaptive meshes with the isovalues of* $||\boldsymbol{v}||$*; Center, P-waves; Right, S-waves*

6.1.2. *3D modeling*

With the same number of dofs, it is well known that the 3D FE computation is generally much more CPU and time consuming than the 2D computation because of the higher element connectivity in a 3D mesh. Hence the 3D modelling of the wave propagation in the foothills areas remains still a challenging case if the area to be studied is large. The 3D computation presented here considers an area whose dimension is only 60 times the smallest wavelength. Its geologic structure is shown in Figure 11(a). A coarse initial mesh is used to start the adaptive computing (Figure 11(b)). In Figures 12 and 13, adaptive meshes and the propagation of P- and S waves are presented. The beginning of the wave propagation is finely captured. But such a calculation remains still very CPU and time consuming, the last 3D adaptive mesh shown here has more than 2 millions dofs.

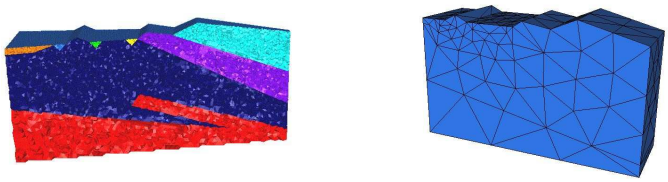


Figure 11. *Foothills areas. Left, 3D mesh showing the velocity contrast between the layers, the velocity in near surface is lower than in the substratum; Right, 3D initial coarse mesh for the adaptive computing*

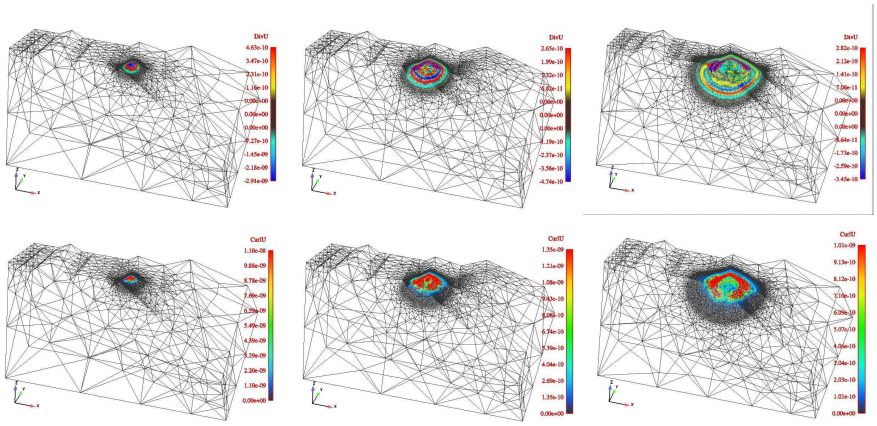


Figure 12. *3D adaptive computing of waves propagation in foothills areas. Above, adaptive meshes and P-waves; Underneath, adaptive meshes and S-waves*

6.2. *Shock wave propagations during space launcher stage separation*

Shock waves accompanying very fast moving loads are well known phenomena: when the velocity of the moving load is faster than the wave velocity of the material, high deformation gradients are produced and localised in the structure. The pyrotechnic cut used for the stage separation of a space launcher can be modelled as a moving load and can generates high level shock waves, which are dangerous for embarked electronic equipment and payload (satellites, etc.). To try to control the dynamic environment of the payload, a good understanding of wave propagations across structures of the launcher is required.

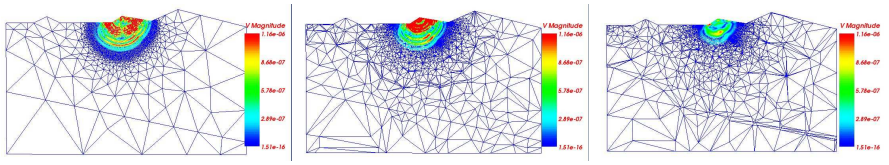


Figure 13. *Planar cuts in the depth of foothills areas at a time step. From left to right, the planar cut moves away from the loading point (Remark: the mesh of a planar cut can be distorted, which is actually a graphic effect.)*

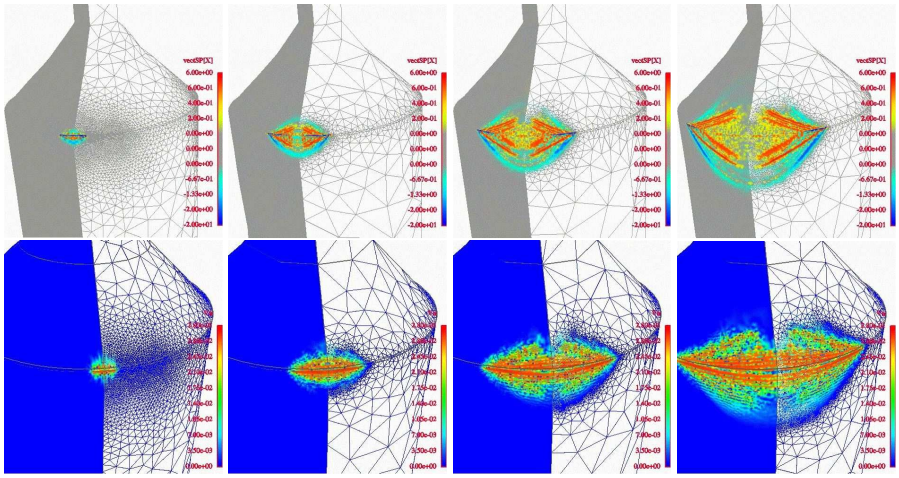


Figure 14. *Adaptive computing of the propagation of shock waves in space launcher structures submitted to pyrotechnic cut loads. Are presented: Above, adaptive meshes, pressure (blue colors) and shear (red and yellow colors) shock waves; Underneath, adaptive meshes and bending shock waves*

Here the adaptive modelling of shock wave propagation in two connected shell substructures of a space launcher is presented. The pyrotechnic cut is modelled as a moving load along the interface between the conical and the cylindrical substructures. The speed of the moving load (more than 7000 m/s) is higher than the propagation speed of P-wave in the structures (about 5000 m/s). Figure 14 shows simultaneously the propagation of membrane P- and S-waves and of the bending waves in structures of a space launcher. The efficiency of the adaptive DG solver to get precise capture of shock waves is illustrated. The FE mesh is coherently adapted showing consider-

able refinement around the propagating wave fronts but also mesh coarsening behind them. Due to the supersonic moving loads, well-known Mach cones of pressure, shear and bending shock waves are consequently generated. Otherwise, due to the curvature of the structures, the conversion of waves is generalised: several wave fronts are observed related to either pressure or shear and, in the cylindrical upper part and the conical lower part, wave propagation phenomena are significantly different. Finally we remark that, as the bending waves are dispersive - i.e. their velocity is frequency dependant - a series of shock wave fronts is observed.

7. Conclusions

In this paper, an adaptive two-fields time DG method is presented and its reliable and efficient numerical implementation is discussed. More particularly, the interpolation of mechanical variable between unmatched adaptive meshes is considered and the adopted interpolation operators are shown to be appropriate and efficient by the presented numerical examples. Otherwise, an implicit bloc Gauss-Seidel DG solver with frequency dependent relaxation is proposed. Its convergence is improved especially when the FE mesh is adaptively refined and starts capturing high frequencies components. The method has been successfully applied to large-sized engineering shell and 3D structures. The advantage of automatic adaptive remeshing is illustrated by these applications.

Nevertheless, the 3D adaptive computing of the wave propagation still remains a challenging issue and the sole introduction of adaptivity is not sufficient. We believe that a coupled strategy of the mesh and the parallel computing is a promising way to take up the challenge. Otherwise, our experiences have shown that the mesh coarsening is more delicate to handle than the mesh refining. But the mesh coarsening is very important to optimise the problem size. So, more precise errors indicators or more adaptive astute strategies are expected in order to highly optimise adaptive FE meshes.

Acknowledgements

The supports of the CNES (Centre National d'Etudes Spatiales, French National Space Study Center) are gratefully acknowledged.

8. References

Aubry D., Jay G., Tie B., Muzzolini R., " A combined mesh and model adaptive strategy for the scaling issues in the numerical modelling of the ductile damage in thin panels", *Computer Methods in Applied Mechanics and Engineering*, vol. 192, n° 28-30, p. 3285 - 3330, 2003.

Aubry D., Tie B., Lucas D., " Adaptive strategy for transient/coupled problems. Applications to thermoelasticity and elastodynamics", *Comput. Methods Appl. Mech. Engrg.*, vol. 176, p. 41-50, 1999.

Babuska I., Rheinboldt W. C., " Error estimates for adaptive finite element computations", *Journal of Numerical Analysis*, vol. 15, n° 4, p. 736-754, 1978.

Bank R., Smith R., " A posteriori error estimates based on hierarchical bases", *Industrial and Applied Mathematics*, vol. 30, p. 921-935, 1992.

Boullard A., Propagation des ondes dans les coques simples ou en nid d'abeille soumises des charges mobiles : Application au lanceur Ariane 5 sous chocs pyrotechniques, Thèse, Ecole Centrale de Paris, Laboratoire Mécanique des Sols, Structures et Matériaux, 2004.

Hinton E., Campbell J., " Local and global smoothing of discontinuous finite element functions using a least squares method", *International Journal for Numerical Methods in Engineering*, vol. 8, p. 461-480, 1974.

Hulbert M., Hughes T., " Space-time finite element methods for elastodynamics: formulations and error estimates", *Comput. Methods Appl. Mech. Engrg*, vol. 66, p. 339-363, 1988.

Johnson C., Hansbo P., " Adaptive finite elements methods in computational mechanics", *Computer Methods in Applied Mechanics and Engeneering*, vol. 101, p. 143-181, 1992.

Ladevèze P., " Error estimate procedure in the finite element method and application", *J. Num. Anal.*, vol. 3, n° 20, p. 485-509, 1983.

Leclère J. M., Modélisation parallèle de la propagation d'ondes dans les structures par éléments finis adaptatifs, Thèse, Ecole Centrale de Paris, Laboratoire Mécanique des Sols, Structures et Matériaux, 2001.

Li M., Wiberg N., " Implementation and adaptativity of a space-time finite element method for structural dynamics", *Computer Methods in Applied Mechanics and Engeneering*, vol. 156, p. 211-229, 1998.

Oden J., Demkowicz L., Rachkowick W., Westermann T., " Toward a universal $h - p$ adaptive finite element strategy: A posteriori error estimates based on hierarchical bases", *Computer Methods in Applied Mechanics and Engineering*, vol. 77, p. 113-180, 1989.

Tie B., Aubry D., Boullard A., " Adaptive computation for elastic wave propagation in plate/shell structures under moving loads", *Revue européenne des Éléments finis*, vol. 12, n° 6, p. 717 - 736, 2003.

Wiberg N., Li M., " Adaptive finite element procedures for linear and non-linear dynamics", *Num. Meth. Eng.*, vol. 46, p. 1781-1802, 1999.

Zienkiewicz O., Zhu J., " The superconvergent patch recovery and a priori error estimates", *International Journal for Numerical Methods in Engineering*, vol. 33, p. 1331-1382, 1992.

9. Appendix

Here, we give the proof of the unconditional convergence of the implicit DG solvers of Jacobi and Gauss-Seidel type.

9.1. *Unconditional convergence of the implcit bloc Jacobi DG solver*

Proposition 9.1 *The implicit bloc Jacobi DG solver [16] unconditionally converges for* $\forall \alpha \in]0, 1.268[$.

PROOF. The iterative algorithm [16] can be put into the following matrix form:

$$V^{i+1} = \alpha \begin{bmatrix} M_{nn}^{*-1} & 0 \\ 0 & M_{nn}^{*-1} \end{bmatrix} \begin{Bmatrix} \bar{F}_n \\ \bar{F}_{n+1} \end{Bmatrix} + ((1-\alpha)I + \alpha A_J)V^i \qquad [23]$$

with:

$$V^{i+1} = \begin{Bmatrix} V_n^{i+1} \\ V_{n+1}^{i+1} \end{Bmatrix} \qquad [24]$$

so, the recurrence relation of the iterative corrections is:

$$V^{i+1} - V^i = R_J(V^i - V^{i-1}) \qquad [25]$$

with $R_J = ((1-\alpha)I + \alpha A_J)$ and the matrix A_J is defined is:

$$A_J = \begin{bmatrix} 0 & -\frac{2}{3}B^* \\ -2(I - B^*) & 0 \end{bmatrix} \qquad [26]$$

with:

$$B^* = M_{nn}^{*-1} M_{nn} = I - \frac{\Delta t^2}{6} M_{nn}^{*-1} K_{nn} \qquad [27]$$

The necessary and sufficiency condition of convergence is that the spectral radius of the iterative matrix R_J is strictly inferior to 1. We note that $R_J = A_J$ when $\alpha = 1$. As the matrices B^* and $(I - B^*)$ have the same eigenvectors Φ, we conclude that the eigenvectors of the matrix A_J can be written under the following form:

$$\Phi_J = \begin{Bmatrix} r\Phi \\ \Phi \end{Bmatrix} \qquad [28]$$

We get therefore the eigenvalues of the matrix A_J depending upon the eigenvalues of the matrix B^*:

$$\lambda_{A_J}^2 = -\frac{4}{3}(\beta - \frac{1}{2})^2 + \frac{1}{3} \qquad [29]$$

It is straightforward that $|\lambda_{A_J}| \leq \frac{\sqrt{3}}{3}$ if $0 \leq \beta \leq 1$.
Now, we calculate the eigenvalues of the matrix B^*. It is easy to show that:

$$B^*\Phi = \frac{1}{\frac{(\Delta t\omega)^2}{6}+1}\Phi \qquad [30]$$

where Ψ is the eigenvector of the eigensystem [20]. So, we have:

$$\beta = \frac{1}{\frac{(\Delta t\omega)^2}{6}+1} \leq 1 \qquad [31]$$

Hence, the bloc Jacobi DG solver without relaxation (*i.e.* $\alpha = 1$) unconditionally converges.
Finally we get the eigenvalues of the iterative matrix R_J:

$$\lambda_{J\pm} = 1 - \alpha \pm \alpha|\lambda_{A_J}| \qquad [32]$$

It is straightforward that $|\lambda_J| < 1$ if $\alpha \in]0, \frac{2\sqrt{3}}{1+\sqrt{3}}[$.

Corollaire 9.2 *The relaxation does not improve the convergence of the implicit bloc Jacobi DG solver.*

PROOF. It can be shown that: if $\alpha > 1$, $\lambda_{J-} < -|\lambda_{A_J}|$; if $\alpha < 1$, $\lambda_{J+} > |\lambda_{A_J}|$.

9.2. *Unconditionnal convergence of the implcit bloc Gauss-Seidel DG solver*

Proposition 9.3 *The implicit bloc Gauss-Seidel DG solver [17] unconditionally converges for* $\forall\alpha \in]0, 2[$.

PROOF. The iterative algorithm [17] can be put into the following matrix form:

$$V^{i+1} = \alpha \begin{bmatrix} M_{nn}^{*-1} & -\frac{2}{3}M_{nn}^{*-1}M_{nn}M_{nn}^{*-1} \\ 0 & M_{nn}^{*-1} \end{bmatrix} \begin{Bmatrix} \bar{F}_n \\ \bar{F}_{n+1} \end{Bmatrix} - R_{GS}V^i \qquad [33]$$

with:

$$R_{GS} = (1-\alpha)I + \begin{bmatrix} -\frac{4}{3}\alpha^2 B^*(I-B^*) & \frac{2}{3}\alpha(\alpha-1)B^*(I-B^*) \\ 2\alpha(I-B^*) & 0 \end{bmatrix} \qquad [34]$$

so, the recurrence relation of the iterative corrections is:

$$V^{i+1} - V^i = R_{GS}(V^i - V^{i-1}) \qquad [35]$$

When there is no relaxation, *i.e.* $\alpha = 1$, we obtain the iterative matrix A_{GS} that reads as:

$$A_{GS} = \begin{bmatrix} -\frac{4}{3}B^*(I - B^*) & 0 \\ 2(I - B^*) & 0 \end{bmatrix} \quad [36]$$

B^* is defined by Equation [27]. As in the preceding proof, the necessary and sufficiency condition of convergence is that the spectral radius of the matrix R_{GS} is strictly inferior to 1. Let us consider at first the case without relaxation and write the eigenvectors of the matrix A_{GS} under the following form:

$$\Phi_{GS} = \begin{Bmatrix} r\Phi \\ \Phi \end{Bmatrix} \quad [37]$$

We recall that Φ denotes the eigenvectors of B^*. If N is the dimension of the matrix B^*, then it can be shown that the matrix A_{GS} has N eigenvalues equal to zero and N eigenvalues depending upon the eigenvalues β of the matrix B^*:

$$\lambda_{A_{GS}} = -\frac{4}{3}(\beta - \frac{1}{2})^2 + \frac{1}{3} \quad [38]$$

It is obvious that $|\lambda_{A_{GS}}| \leq \frac{1}{3}$ as $0 \leq \beta \leq 1$.
When the relaxation is added, the eigenvalues λ_{GS} of the iterative matrix R_{GS} are:

$$\lambda_{GS} = 1 - \alpha + \alpha\frac{\alpha^2\lambda_{A_{GS}} \pm \sqrt{(\alpha^2\lambda_{A_{GS}})^2 - 4(\alpha - 1)\lambda_{A_{GS}}}}{2} \quad [39]$$

It can be proved that $|\lambda_{GS}| < 1$ for $\alpha \in]0, 2[$.

Corollaire 9.4 *Without the relaxation ($\alpha = 1$), the implicit bloc Gauss-Seidel DG solver converges more rapidly then the implicit bloc Jacobi DG solver.*

PROOF. It is obvious if we compare the spectral radius of the iterative matrices A_J and A_{GS} (see Equations [29] and [38]).

Proposition 9.5 *There exists an optimal relaxation coefficient $\alpha_{opt}(f)$ that improves the convergence of the implicit bloc Gauss-Seidel DG solver in the whole frequency domain. It is frequency-dependent: $\alpha_{opt}(f) \in [1, \alpha_{max}]$ is the real root of the following quartic equation:*

$$\lambda_{A_{GS}}\alpha^4 - 4\alpha + 4 = 0 \quad [40]$$

where α_{max} is the optimal relaxation coefficient corresponding to the frequency value $f = \frac{\sqrt{6}}{2\pi\Delta t_n}$, for which the spectral radius $\lambda_{A_{GS}}$ of the matrix A_{GS} reaches its maximal value (see Equation [38]). The numerical value of α_{max} is nearly 1.1415.

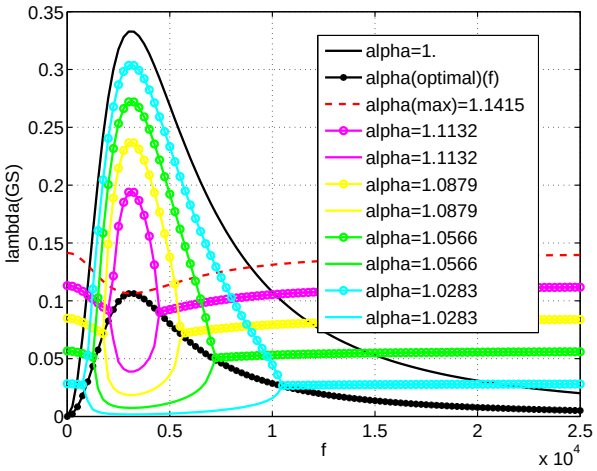


Figure 15. *Convergence ratio of the implicit bloc Gauss-Seidel DG solver* λ_{GS} *depending upon de choice of the relaxation coefficient* $\alpha \in [1, \alpha_{max}]$

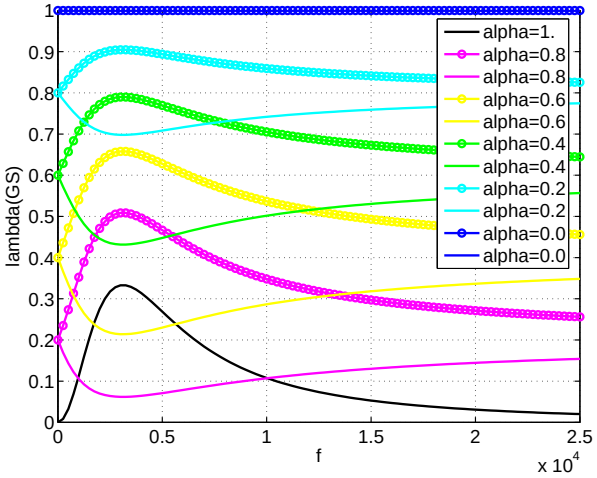


Figure 16. *Convergence ratio of the implicit bloc Gauss-Seidel DG solver* λ_{GS} *depending upon de choice of the relaxation coefficient* $\alpha \in [0, 1]$

PROOF. A detailed analysis of the spectral radius λ_{GS} of the iterative matrix R_{GS} (see Equation [39]) can show that, for a given frequency f, the optimal relaxation coefficient $\alpha_{opt} \in [1, \alpha_{max}]$ and is the one which corresponds to the bifurcation point of two situations: $|\lambda_{GS}|$ has two identical values or two different values (see Figures 15, 16 and 17). Therefore, α_{opt} should vanish the discriminant of [39], resulting in Equation [40].

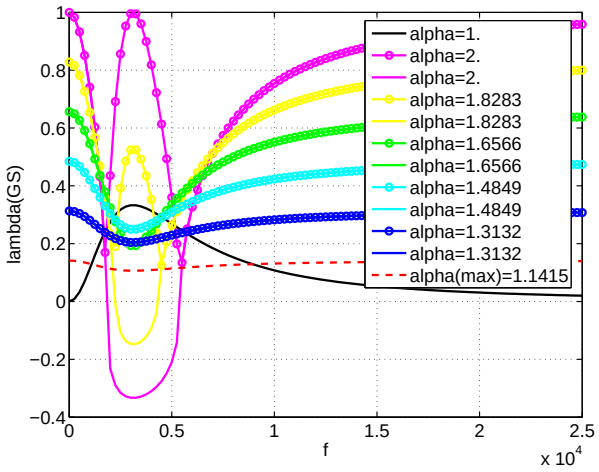


Figure 17. *Convergence ratio of the implicit bloc Gauss-Seidel DG solver λ_{GS} depending upon de choice of the relaxation coefficient $\alpha \in [\alpha_{max}, 2]$*