A Parallel Spectral Element Method for Applications to Computational Mechanics

Fabio Maggio[1], Luca Massidda[1], Javier Sabadell[2], and Giuliana Siddi[1]

[1] Center for Advanced Studies, Research and Development in Sardinia (CRS4), Cagliari, Italy
[2] WesternGeco, Al-Khobar, Saudi Arabia

Abstract. We present a fully unstructured, parallel spectral element method based on domain decomposition. The implementation aims at improving accuracy and performace of a large number of applications in computational mechanics. We accomplish this target by merging the high accuracy of spectral methods with the flexibility of finite elements in treating problems with complex geometry. Parallelism is coarse grain and exploited at the algebraic level: the global linear system is distributed among processors, and then solved by iterative techniques. For evolutive problems it is possible to adopt explicit time-marching schemes, thus resulting in very competitive performances. We provide a couple of numerical experiments, based on the Helmholtz equation and the acoustic problem, and give a measure of accuracy and efficiency, illustrating the method in both steady and evolutive problems.

1 Introduction

In recent years, the spectral element method has been applied to the numerical treatment of applications in various fields of computational mechanics. It enables significant reductions of computational resources respect to more traditional techniques like finite elements or finite differences (see for instance [14], [4], [5], and [10]). The main advantage of spectral elements is their high order approach, which equals the accuracy of low order methods using reduced number of grid points. This is an appealing feature in real case problems, often unaffordable because of the exceedingly large amount of computational effort required. Although spectral elements are not so flexible as finite elements in efficient mesh generation, current research is still under development for the complete integration of mesh generation and fast solvers within the bounds of spectral elements. This work is oriented toward the above target. It illustrates an efficient parallel solver, based on spectral elements and domain-decomposition, for the solution of the Helmholtz equation and the scalar acoustic problem. The method enjoyes both the flexibility of finite element solvers, producing a framework capable of dealing with CAD-oriented geometries, and interaction with the best-known engineering tools. On the other hand, we resort to coarse parallelism adopting a fully unstructured multiprocessor paradigm, based on the balanced decomposition of the algebraic problem among subdomains.

This is the natural development, and the extension to elliptic problems, of previous work aimed at the theoretical analysis of the spectral approach for wave propagation problems (see [4], [5]).

2 Problem formulation

Whe consider the scalar reference problem

$$\begin{aligned}
\alpha\ddot{u} + \beta\dot{u} + \gamma u - \nabla\cdot(\kappa\nabla u) &= f && \text{in } \Omega\times(0,T)\\
u &= \phi && \text{on } \Gamma_D\times(0,T)\\
\frac{\partial u}{\partial n} &= \psi && \text{on } \Gamma_N\times(0,T)\\
\sqrt{\frac{\alpha}{\kappa}}\,\dot{u} + \frac{\partial u}{\partial n} &= 0 && \text{on } \Gamma_{NR}\times(0,T)\\
u|_{t=0} &= u_0 && \text{in } \Omega\\
\dot{u}|_{t=0} &= u_1 && \text{in } \Omega
\end{aligned} \tag{1}$$

defined over a bounded domain $\Omega \subset \mathbb{R}^2$, subject to an external force distribution f and such that $\partial\Omega = \Gamma_D \cup \Gamma_N \cup \Gamma_{NR}$. Equation (1) is a general formulation for the linear acoustic problem, including non-reflecting boundary conditions on Γ_{NR} to simulate behaviour of of the solution u in unbounded domains. When $\alpha = \beta = 0$, $\Gamma_{NR} = \emptyset$, and initial conditions are dropped out, problem (1) reduces to the Helmholtz equation defined in Ω, with $\partial\Omega = \Gamma_D \cup \Gamma_N$.
From now on we refer, without loss of generality (see, e.g., [15], chap. 6) to the case with homogeneous boundary conditions (i.e. $\phi = \psi = 0$). Thus, the variational formulation of problem (1) reads:
find $u : (0,T) \to H_0^1(\Omega)$ satisfying the initial conditions and such that

$$\int_\Omega \alpha\,\ddot{u}\,v\,d\Omega + \int_\Omega \beta\,\dot{u}\,v\,d\Omega + \int_\Omega \gamma\,u\,v\,d\Omega + \mathcal{A}(u,v) =$$

$$-\int_{\Gamma_{NR}} \sqrt{\frac{\alpha^3}{\kappa}}\,\dot{u}\,v\,d\Gamma + \int_\Omega f\,v\,d\Omega \qquad \forall v \in H_0^1(\Omega) \tag{2}$$

where $H_0^1(\Omega) := \{w \in H^1(\Omega) : w|_{\Gamma_D} = 0\}$ and $\mathcal{A}(u,v) := \int_\Omega \kappa\,\nabla u\cdot\nabla v\,d\Omega$.
The formulation (2) easily addresses the Helmholtz case, for which $u, v \in H_0^1(\Omega)$, and terms involving time-derivatives vanish.

3 The spectral element discretization

Spectral elements are usually regarded as a generalization of finite elements based on the use of high order piecewise polynomial functions. The great asset of the

method is the capability to provide an arbitrary increase of accuracy simply enhancing the algebraic degree of these functione (the *spectral* degree).
On the practical ground this operation is completely transparent for the users, who limit themselves to choose the spectral degree at run-time, devolving on the computational code the task to build up suitable quadrature points and new degrees of freedom. Obviously, increasing the spectral degree has also the effect to raise the computational complexity of the problem.
On the other hand, one can also play on the grid refinement to improve the accuracy of the numerical solution, thus following the standard finite element approach (spectral elements are a so-called *h-p* method). When the "exact" solution u is regular, the numerical solution $u_{H,n}$ provided by the proposed method is expected to converge to u as

$$||u - u_{H,n}||_1 \leq C\, H^n\, \exp(-n) \tag{3}$$

where $||\cdot||_1$ is the H^1 norm and the constant C is independent from H and n. Therefore, the set up of numerical analyses should take into account the balance between benefits provided by accuracy (obtained by a suitable compromise between spectral degree and mesh refinement), and the computational cost.
Our spectral element method stems from the decomposition of the computational domain Ω into a family $\mathcal{S}_H$ of non-overlapping quadrilaterals Ω_k (hexahedra in 3D), with typical linear size H and such that the intersection between two elements is either empty, or a vertex, or an edge, and $\overline{\Omega} = \bigcup_k \overline{\Omega_k}$.
We fix the spectral degree $n \geq 1$, and build up a suitable approximation of $H_0^1(\Omega)$ by introducing the space

$$Q_n(\Omega_k) := \left\{ v = \hat{v} \circ F_k^{-1},\ \hat{v} \in \mathbf{Q}_n(\widehat{\Omega}) \right\} \tag{4}$$

where $\widehat{\Omega} := [-1,1]^2$, F_k is a suitable mapping between $\widehat{\Omega}$ and Ω_k, and $\mathbf{Q}_n(\widehat{\Omega})$ is the space of polynomials in the variables (x_1, x_2) defined in $\widehat{\Omega}$ with degree less than or equal to n, with respect to each variable. Due to lack of space we do not provide further details: interested readers can refer to [2]. We are now in the position to introduce the finite-dimensional subspace

$$X_{H,n}(\Omega) := \left\{ v_{H,n} \in \mathcal{C}^0(\overline{\Omega})\ :\ v_{H,n}|_{\Omega_k} \in Q_n(\Omega_k),\ \forall\ \Omega_k \in \mathcal{S}_H \right\}, \tag{5}$$

along with its subspace

$$V_{H,n}(\Omega) := \left\{ v_{H,n} \in X_{H,n}(\Omega)\ :\ v_{H,n} = 0 \text{ on } \Gamma_D \right\}. \tag{6}$$

The finite dimensional space $V_{H,n}(\Omega)$ is the spectral approximation of $H_0^1(\Omega)$; hence our spectral element method is completely defined once we provide a method to evaluate integrals and derivatives appearing in (2). This is done by means of the Gauss-Lobatto points and the collocation derivative matrix (refer to [2] and [1]).
Following the approach described above, one obtains a set of second order ordinary differential equations:

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{D}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} = \mathbf{F}\ , \tag{7}$$

where $\mathbf{M}$, $\mathbf{D}$ and $\mathbf{K}$ are the mass, damping and stiffness matrix, respectively, $\mathbf{U}$ is the vector of unknown solutions, and $\mathbf{F}$ is the vector of external loads.

4 Domain decomposition and parallel solution

The first step to handle problem (2) by our method, is to create a geometry-model and discretize it by means of a grid of quadrilaterals. While the construction of pure quadrileral meshes is easily performed by the most common grid generators, the 3D case is more delicate. Nevertheless, nowadays several state-of-the-art preprocessing tools are available, able to generate pure hexahedra meshes even involving 3D domains with complex shapes. Grids of hexahedra can be generated as multiblock mapped meshes, or via decomposition of tetrahedra or direct methods (see [12] for an interesting review on the argument).
Once the physical domain has been discretized by the grid, and the polynomial degree has been chosen, the spectral grid-points (the Gauss-Lobatto nodes) are built; this process is carried on element-by-element (see [2] for details).
The computation of spectral nodes is done by the solver rather than by the mesh generator; thus, any quadrilateral or hexahedra grid in finite element format can be used as an input, and the dimensions of the files containing the mesh do not exceed reasonable limits.
When dealing with the Helmholtz equation, (7) is already in the form of an algebraic problem

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{8}$$

Concerning the acoustic problem, the discretization of first and second order time-derivatives in (7) by means of backward finite-difference schemes (see [11]), still gives rise to the algebraic problem (8), where $\mathbf{A}$ is a combination of the mass, damping, and stiffness matrices, $\mathbf{x}$ is the unknown solution at a given instant, and the right hand side $\mathbf{b}$ can be evaluated using solutions at previous time-steps and prescribed external loads.
Nevertheless, it is possible to make use of explicit time-marching schemes, thus obtaining a significant save of computational resources for a large class of applications in computational mechanics. This is the choice we adopt through this paper. In this case, matrix $\mathbf{A}$ in (8) is diagonal, and the bottleneck is the evaluation of $\mathbf{b}$, resulting in a matrix-vector product per each time-step. Therefore, for both steady and evolutive problems, the problem is to perform parallel matrix-vector products in optimal way (in fact, this is also the base of preconditioned iterative solvers).
The starting point is the decomposition of the grid in a number of subdomains: this task is not trivial since the produced partition should be well balanced both in terms of number of degrees of freedom, and in terms of communication to be performed among subdomains during the parallel solution of (8). There are few public domain tools developed by groups devoted to this research subject: among them we chose Metis ([9]), developed at the University of Minnesota.
Once the set of quadrilaterals $\mathcal{S}_H$ has been split, subdomains are associated with

processors following a 1-to-1 distribution. Performing the parallel solution of the problem at hand is then a three-steps process:

1. each processor assembles that portion of the spectral matrix **A** corresponding to degrees of freedom set on those nodes assigned to the processor itself. In order to do that, information is exchanged among processors via MPI.
2. The solution of the distributed algebraic problem is devolved on a parallel iterative solver, which takes care of the communication needed for performing matrix-vector products, and provide the local solution.
3. The latter are gathered by a master process, which provide the global solution of the problem, still via MPI.

For the second step we adopt Aztec ([8]), by Sandia National Laboratories. There are a numbers of options whose proper choice can greatly speed up the parallel iterative solution of (8). In particular, it turns out that preconditioners, rather than solvers, can affect significantly the final performance (see [13]). This is particularly true for methods based on Schwarz additive technique. This class of preconditioner, coupled with algebraic agglomeration technique, exhibis a better behaviour when the number of subdomains increase beyond the number of available processors. To this aim, a modified version of Aztec has been developed ([6] and [7]).

5 Numerical examples

In this section we provide a couple of experiments aimed to show accuracy and parallel performance of the method. Finally we include an example of geophysical application with complex geometry.
Numerical tests have been performed on two different architectures: a cluster of high performance PC's equipped with Linux and the Myrinet packet-communication, and an IBM SP, a massively parallel machine with shared memory. Features of the two computers are resumed in Table 1.

Machine	IBM-SP	PC cluster
Number of nodes/processors	1(high)/16	4/8
Processors	POWER3-II	Pentium III
CPU Clock rate (MHz)	375	800
RAM (Gbytes)	16 (shared)	4 (distributed)
First level cache (Kbytes)	64 per proc.	256 per proc.
Communication	MPI	MPI-Myrinet

Table 1. Characteristics of IBM SP and cluster of PC's used for numerical experiments

5.1 The acoustic problem

We consider problem (1) with $\alpha = 1/(\rho\, c^2)$, $\beta = \gamma = 0$, and $\kappa = (1/\rho)$, where ρ and c are the density and the propagation velocity of the medium were acoustic waves travel. In order to check (3), we perform several analyses of a simple reference problem defined over $\Omega := [0,1]^2$, with analytical solution

$$u(x,y) = \sin(4\pi x)\; \cos(4\pi y)\; cos(2\pi t) \tag{9}$$

We fix $\rho = c = 1$, $T = 10^{-3}$ s and $\Delta t = 10^{-5}$ s, corresponding to 100 time-steps. We choose a so small Δt to keep away the effect of time-discretization, not included in (3). In all tests we use a regular grid made by squares with horizontal and vertical edges.
First we fix the number of elements (64 elements, corresponding to $H = 0.125$), and measure the error

$$\epsilon = \frac{||u - u_{H,n}||_0}{||u||_0} \tag{10}$$

for different values of n: this is reported in figure 1, left. Since $||u||_0$ is constant and the plot has a logarithmic scale for ϵ, the error trend is expected to approximate a straight line.
In a second batch of experiments we fix the spectral degree ($n = 1$), let H change and measure ϵ. The error behaviour is displayed in figure 1, right. Both scales are logarithmic, and still the expected behaviour is a straight line. Since ϵ is evaluated in L^2 norm, the slope of the error curve is expected to be $n + 1$.

As a reference test for measuring parallel performance, we use a literature case concerning wave propagation in single flat layer ([4]). The computational domain is still a square with edges 2500 m long, but the medium is no longer homogeneous, consisting of a horizontal uniform layer overlying a homogeneous half-space: propagation velocities are $c_1 = 2000$ m/s and $c_2 = 4000$ m/s, respectively. Density is homogeneous ($\rho = 1$ g$\cdot$cm^{-3}).
The set up of the experiment is as follows. We use regular grid as in the previous case, with fixed spectral degree $n = 9$. Then we perform several runs using a variable number of processors and letting the number of elements change in such a way that processors have to manage the same number of degrees of freedom in different analyses. Therefore, since the computational effort per processor in constant, we expect that CPU-time does not vary for the different tests. We measure the parallel efficiency as

$$\eta(p) := \frac{[CPU - time]_1}{[CPU - time]_p} \tag{11}$$

where $[CPU - time]_p$ is the CPU-time for a run with p processors. Results are reported in figure 2, left. It can be appreciated that the method enjoys efficiency always beyond 80% when implemented on the IBM SP. This is a good result for an algorithm based on a fully unstructured approach. Even if the performance for the cluster of PC's is not so good, it should be considered that the cost of the latter is by far lower than the cost of a massively parallel

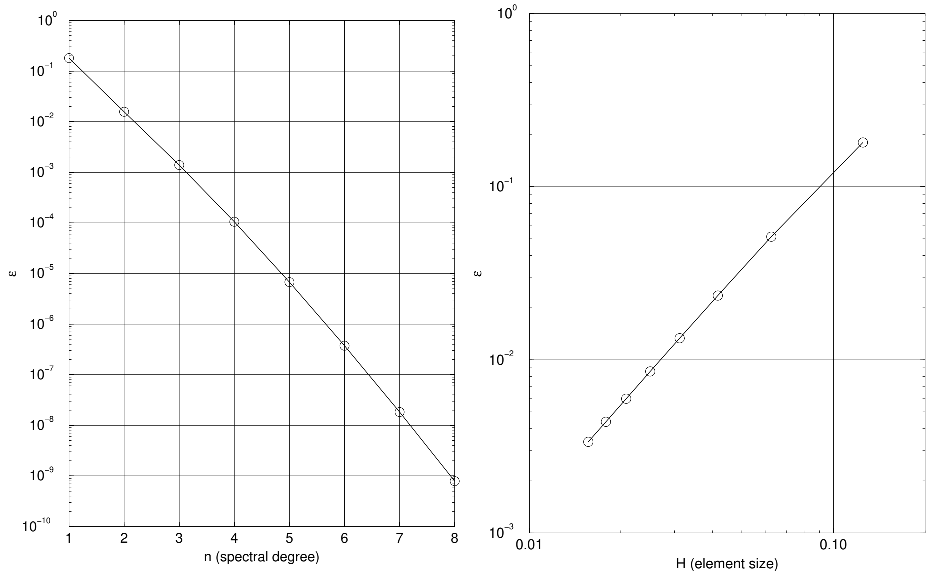

Fig. 1. Accuracy of the numerical solution with respect to spectral degree n *(left)* and element size H *(right)*

machines. Furthermore, PC technology is improving very rapidly these days, and probably the gap between the CPU performance is becoming less significant in the very near future. The last test for acoustics is devoted to an application with complex shape: a geophysical model of a salt-dome due to Dablain ([3]), with homogeneous density ($\rho = 1\ \mathrm{g{\cdot}cm^{-3}}$) and propagation velocity as in figure 3. The model dimensions are 10 000 m × 3 000 m. We use a fully unstructured mesh with approximately 39 000 quadrilaterals and spectral degree $n = 4$, corresponding to 618 000 grid-points. A point source is set at a depth of 100 m, overlying the dome. The signal has a frequency content up to 50 Hz. Figure 4 displays a zoom of the computational grid in the region of the dome. In figure 5 we report few snapshots of the solution for different times.

5.2 The Helmholtz equation

We consider equation (1) in the particular case $\alpha = \beta = \gamma = 0$, and $\Gamma_{NR} = \emptyset$, that is the Poisson equation. At the numerical level this is a critical case of the Helmholtz equation since matrix $\mathbf{A}$ in (8) has a worse condition number than the case with $\gamma \neq 0$. We consider a reference test with $\Omega := [0, 1000]^2$, and analytical solution given by the fundamental solution of the Laplace equation in 2D

$$u(x, y) = c\ \ln[(x - x_0)^2 + (y - y_0)^2] \tag{12}$$

where $(x_0, y_0) := (-100, -100)$. We choose $\kappa = 1$, and still use a regular grid made by squares with horizontal and vertical edges.

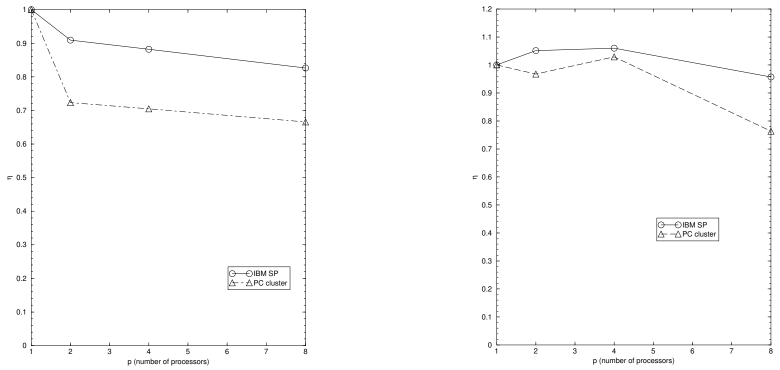

Fig. 2. Parallel efficiency as a function of number of processors. Acoustic problem *(left)* and Poisson equation *(right)*

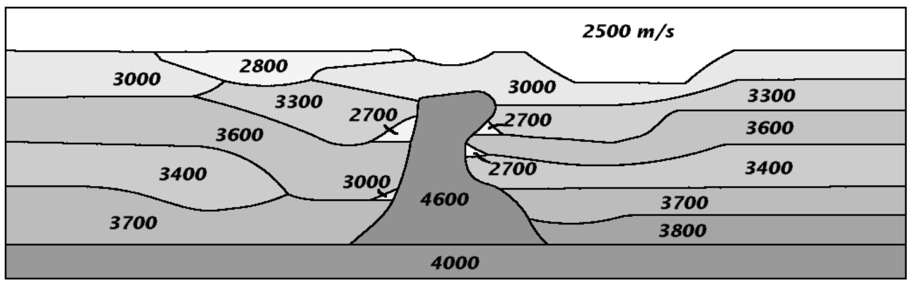

Fig. 3. The salt-dome model. Numbers refer to wave velocity (m/s).

First we fix the number of elements (64 elements, corresponding to $H = 125$), and measure the error ϵ defined in the previous section. The behaviour of ϵ as a function of n and H is displayed in figure 6.

Then we use the same experiment for measuring parallel performance. We set spectral degree $n = 7$ and run different analyses keeping constant the number of deegres of freedom per processor. The trend of η is reported in figure 2, right. We remark that η has a less regular behaviour than in the acoustic case: as a matter of fact, increasing the size of the overall matrix has the effect to raise the condition number of $\mathbf{A}$, and the Schwarz preconditioner we use has different performances when different number of subdomains are used.

6 Conclusions

We presented a parallel method for the solution of steady and evolutive problems, based on spectral elements and domain-decomposition. A major asset of the method is the flexibility in addressing both applications with complex geometries and large scale problems, thanks to the capability of playing indifferently on the

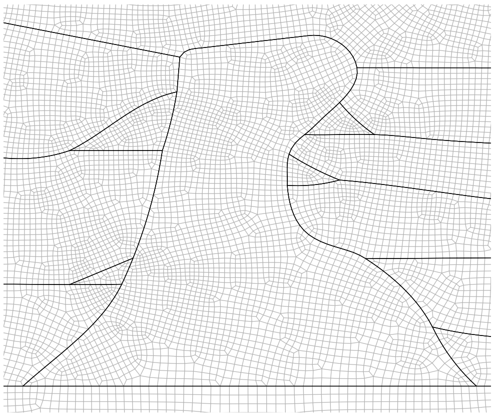

Fig. 4. A zoom of the grid in the dome region

mesh refinement or on the polynomial order to increase the accuracy of the numerical solution. Numerical experiments performed on two basic problems show that the method is well suited both for steady and evolutive problems. In the first case it can benefit from the coupling with state-of-the-art algebraic solvers and preconditioners. In both situations the proposed method exhibits good parallel efficiency, even with implemented on a cluster of PC's, which can be regarded as a good compromise between hardware cost and performace.

Acknowledgments
The work at CRS4 has been partly supported by Sardinia Region Authorities and MURST (Ministero per l'Universitá e la Ricerca Scientifica e Tecnologica).

References

1. Canuto, C., M.Y. Hussaini, A. Quarteroni, and T.A. Zang, 1988, *Spectral methods in fluid dynamics*, Springer Verlag, New York
2. Casadei, F., E. Gabellini, G. Fotia, F. Maggio, and A. Quarteroni, *A hybrid spectral-element/finite-element method for complex wave propagation applications*, CRS4-TECH-REP-00/21, submitted to Computer methods in applied mechanics and engineering
3. Dablain, M.A., *The application of high-order differencing to the scalar wave equation*, Geophysics, 51, pp. 54-66, 1986
4. Faccioli, E., F. Maggio, A. Quarteroni, and A. Tagliani, *Spectral-domain decomposition methods for the solution of acoustic and elastic wave equations*, Geophysics, 61 (4), pp. 1160-1174, 1996
5. Faccioli, E., F. Maggio, R. Paolucci, and A. Quarteroni, *2D and elastic wave propagation by a pseudo-spectral domain decomposition method*, Journal of Seismology, 1, 237–251, 1997
6. Formaggia, L., and M. Sala, *Technical Report 4.4*, Project Idemas (Brite-Euram/Ofes, Project N° BE97-4162, Contract N° BRPR-CT97-0591), 2001

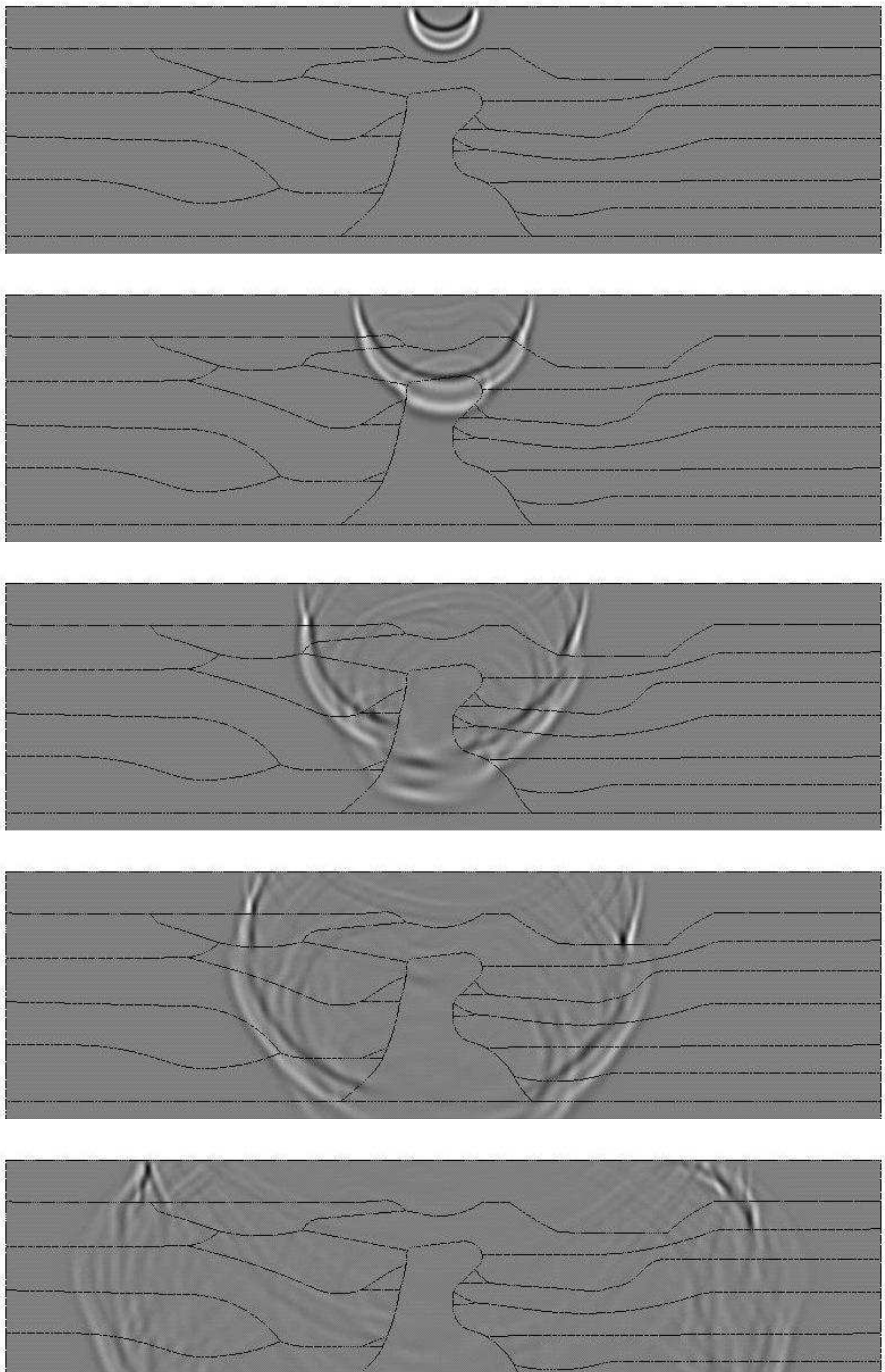

Fig. 5. Solutions of the salt-dome simulation. Snapshots are taken at T=0.25 s, T=0.50 s, T=0.75 s, T=1.00 s, T=1.50 s (from top to bottom). Pressure values are normalized to the range [-1,+1] (from black to white).

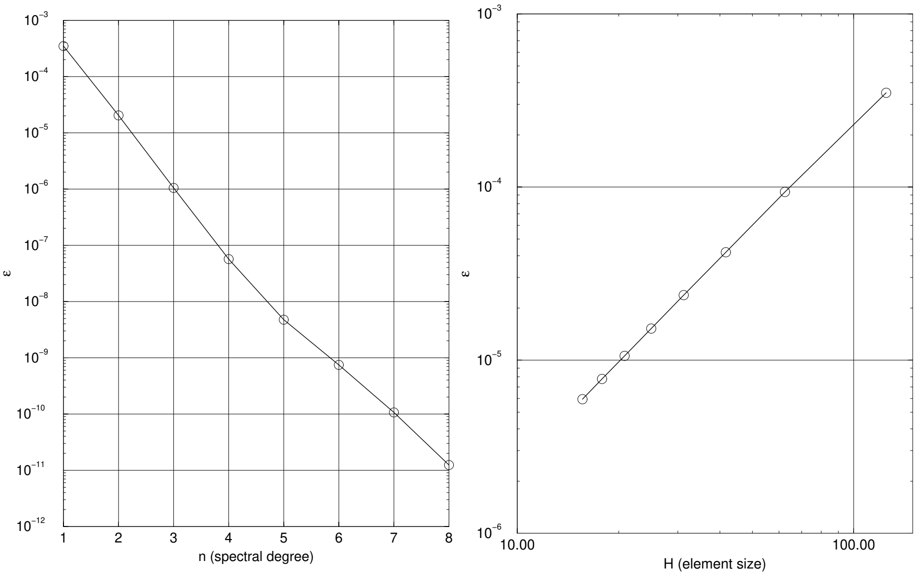

Fig. 6. Accuracy of the numerical solution with respect to spectral degree n *(left)* and element size H *(right)*

7. Formaggia, L., F. Maggio, M. Sala and V. Selmin, *Complex industrial test cases - Validation Report 7.3*, Project Idemas (Brite-Euram/Ofes, Project N° BE97-4162, Contract N° BRPR-CT97-0591), 2001
8. Hutchinson, S.A., L.V. Prevost, J,N. Shadid, C. Tong, and R.S. Tuminaro, *Aztec User's Guide - Version 2.0* (http://www.cs.sandia.gov/CRF/aztec1.html), 1998
9. Karypis, G., and V. Kumar, *METIS - A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices*, (http://www-users.cs.umn.edu/ karypis/metis/), 1998
10. Komatitsch, D., and J.P. Vilotte, *The spectral element method: an efficient tool to simulate the seismic response of 2D and 3D geological structures*, Bulletin of the Seismological Society of America, 88, pp. 368-392, 1998
11. Maggio, F., and A. Quarteroni, *Acoustic wave simulation by spectral methods*, East-West Journal of Numerical Mathematics, 2 (2), pp. 129-150, 1994
12. Owen, S.J., *A Survey on Unstructured Mesh Generation Technology*, Proceedings 7*th* International Meshing Roundtable, Dearborn, MI
13. Paglieri, L., C. Vittoli, E. van der Weide, and H. Deconink, *Modules for parallel Krilov solvers and preconditioners - Technical Report 4.1*, Project Idemas (Brite-Euram/Ofes, Project N° BE97-4162, Contract N° BRPR-CT97-0591), 1999
14. Priolo, E., and G. Seriani, *A numerical investigation of Chebychev spectral element method for acoustic waves propagation*, in Vichnevetsky, R., ed. Proc. 13th IMACS World Congress on Computational and Applied Mathematics, Criterion Press, pp. 551-556, 1991
15. Quarteroni, A., and A. Valli, *Numerical approximation of partial differential equations*, Springer Verlag, Berlin, 1994