



High-performance finite-element simulations of seismic wave propagation in three-dimensional nonlinear inelastic geological media

Fabrice Dupros^{a,*}, Florent De Martin^a, Evelyne Foerster^a, Dimitri Komatitsch^{b,c}, Jean Roman^d

^a BRGM, BP 6009, 45060 Orléans Cedex 2, France

^b University of Pau, CNRS UMR 5212 MIGP, INRIA Bordeaux Sud-Ouest Magique-3D Project, 64013 Pau, France

^c Institut Universitaire de France, 103 Boulevard Saint-Michel, 75005 Paris, France

^d INRIA Bordeaux Sud-Ouest HiePACS Project, PRES de Bordeaux, CNRS UMR 5800 LaBRI, 33405 Talence Cedex, France

ARTICLE INFO

Article history:

Received 2 December 2008

Received in revised form 21 September 2009

Accepted 28 December 2009

Available online 4 January 2010

Keywords:

Seismic numerical simulation

Finite-element method

Parallel sparse direct solver

Nonlinear soil behaviour

ABSTRACT

We present finite-element numerical simulations of seismic wave propagation in non linear inelastic geological media. We demonstrate the feasibility of large-scale modeling based on an implicit numerical scheme and a nonlinear constitutive model. We illustrate our methodology with an application to regional scale modeling in the French Riviera, which is prone to earthquakes. The PaStiX direct solver is used to handle large matrix numerical factorizations based on hybrid parallelism to reduce memory overhead. A specific methodology is introduced for the parallel assembly in the context of soil nonlinearity. We analyse the scaling of the parallel algorithms on large-scale configurations and we discuss the physical results.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Numerical simulations of seismic wave propagation are an important tool for risk mitigation and assessment of damage in future hypothetical earthquake scenarios. The literature about earthquake modeling and three-dimensional site amplification based on an elastic behavior of the soil is abundant (e.g. [1,2]). These simulations take into account the effect of topography, but an important issue that is not addressed in these articles is the use of a nonlinear constitutive law to describe the inelastic behavior of the soil. Using such a law leads to several difficult problems from a numerical point of view and this problem is not often addressed in the literature, in particular in the case of very large-scale problems on thousands of processors. Another important aspect is to use a fully 3D basin model and not a very simplified model consisting of flat layers for instance. Starting from a classical and robust numerical approach called the initial stress method [3,4], we build a robust parallel methodology to tackle this problem. We overcome two classical limitations for large-scale modeling: first of all we consider the memory overhead coming from the sparse direct solver generally used for this class of problems. By using an MPI-thread implementation of the PaStiX¹ linear solver [5], we get a maximum gain of a factor of 6. The second bottleneck is the load-balancing owing to the fact that the nonlinearity is not evenly distributed in space nor in time. We use a suitable two-level algorithm mixing a graph-coloring algorithm for the upper nonlinear layer and a classical mesh partitioning approach for the rest of the domain we obtain a speedup of 3.6× in terms of elapsed time. We analyse the scaling of our algorithms on up to 1024 processors and we apply them to a model of the French Riviera in France [6].

* Corresponding author. Tel.: +33 2 38 64 46 76; fax: +33 2 38 64 39 70.

E-mail addresses: f.dupros@brgm.fr (F. Dupros), f.demartin@brgm.fr (F. De Martin), e.foerster@brgm.fr (E. Foerster), dimitri.komatitsch@univ-pau.fr (D. Komatitsch), Jean.Roman@inria.fr (J. Roman).

¹ <http://www.gforge.inria.fr/projects/pastix/>.

2. Related work

For three-dimensional geological structures, several numerical methods can be used to solve the seismic wave equation. Finite-difference methods (FDM) [7], finite-element methods (FEM) [8], spectral and pseudo-spectral methods [9], and spectral-element methods (SEM) [10] have been used in the last decades. From a numerical point of view, finite-differences approach are limited by the number of points required to accurately sample the wavelength, which leads to a very large mesh and large memory consumption in the case of realistic 3D problems. Pseudo-spectral approaches with Chebychev or Legendre polynomials partially overcome these drawbacks, but it is uneasy to take a complex geometry into account because of the need to map the domain of interest smoothly to a rectangular reference grid. Another possibility is to consider the variational form of the equation. Finite-element methods and spectral-element methods are based on this approach with the use of high-order polynomials for the approximation in the case of SEM. In the case of complex geometry, one of the main advantages is that the free surface condition is naturally taken into account. Moreover, the spectral-element method provides a significant improvement in terms of accuracy and computational efficiency. The frequency domain approach is also used in geophysics, especially for inverse problems in the acoustic case (e.g. [11,12]), but far more often the problem is solved in time (for most direct problems and also for inverse problems in the more complex elastic, viscoelastic or poroelastic cases (see e.g. [13,14])). For very large-scale problems most of the published literature in the last decades is in time, not in frequency. Some software packages designed for earthquake engineering and local site amplification analysis are based on a frequency domain approach but generally under the strong assumption of linearity of the material behavior; this is for instance the case of MISS3D.² In our case, the time-domain is more suitable than the frequency-domain approach because we consider a very large-scale nonlinear problem to estimate local site amplification coming from the nonlinear properties of the soil as well as topography.

Regarding the seismic wave equation, applications to large-scale modeling with more than a thousand processors have been reported based either on FDM, FEM or SEM. For instance in [15] the authors present recent and past computation of strong ground motion in the Tokyo basin in Japan. The computations are carried out on the Japanese Earth Simulator supercomputer and a high-order finite-difference method is used to solve the equation. Simulations of an earthquake from the year 1855 with four billions unknowns on 1024 processors are analyzed. In [16,17], simulations of seismic wave propagation are reported at the scale of the full Earth. The SEM is implemented on the Earth Simulator supercomputer or on the Marenostrum supercomputer in Spain on respectively 1944 processors and 2166 processor cores. The authors pay particular attention to data locality and mesh partitioning to enhance parallel performance. The feasibility of large-scale explicit FEM simulations in seismology has been demonstrated for instance in [18]. For the Los Angeles basin (USA), the authors describe computations with 3000 processors with good results in terms of scaling. Mesh adaptivity (AMR – Adaptive Mesh Refinement) has also been considered for large-scale simulations [19]. This approach is rarely used to study the propagation of seismic waves in geological media because in a geological structure each pressure (P) or shear (S) body wave that propagates across a discontinuity of the model (for instance the interface between two geological layers, or a fault) can generate up to four waves depending on its incidence angle of the material properties of the layers, a transmitted P wave, a reflected P wave, a transmitted S wave and a reflected S wave. These four waves will in turn quickly generate 16 waves when they reach another discontinuity of the model; and so on. Since geological media are full of interfaces and faults, the model is therefore quickly full of waves propagating everywhere and it is not efficient to try to track the wavefronts and apply AMR to reduce the numerical cost in regions that would not contain any wave.

References describing nonlinear simulations with complex constitutive models in three-dimensional geological structures are not very common. Since the early 1970s, efforts have been mainly devoted to one-dimensional computations, with the development of several computer programs (e.g. SHAKE [20], CyberQuake [21] among others). Most of these programs use the equivalent linear approach [22], which deals with a viscoelastic multi-layered soil model. However, observations from many recent strong motion events have demonstrated that nonlinear soil behavior strongly affects the seismic motion of near-surface deposits, resulting in shear wave velocity reduction, irreversible settlements, and in some cases pore-pressure build-up leading to liquefaction. The equivalent linear approach cannot be considered as a constitutive model able to represent the nonlinear soil behavior and several drawbacks of this approach have already been listed in literature. On the contrary, using an appropriate nonlinear (e.g. elastoplastic) constitutive model for soil deposits may permit to reproduce the complex soil behavior under seismic loading and is preferred in recent analyses. To our knowledge, Xu et al. [23] is the only reference with a study of local site amplification at relatively large-scale in 3D. The simulations described are based on a simplified basin model and the problem size is rather modest. Moreover the explicit numerical method used in that article is difficult to extend to model strong nonlinearities because selecting the right time step to ensure numerical stability is uneasy [24].

In this article we introduce a parallel methodology for large-scale nonlinear modeling of seismic wave propagation. The article is organized as follows: Section 2 discusses the numerical problem under study; and in Section 3 we consider the parallel algorithms for the finite-element solver and the PaStiX sparse direct solver. Section 4 describes our model of the French Riviera. Sections 5 and 6, respectively present an analysis of the parallel performance of our algorithms and of the physical results.

² <http://www.mssmat.ecp.fr/structures/missuk.html>.

3. Numerical modeling of seismic wave propagation in inelastic media using a finite-element method

3.1. Governing equations

In a three-dimensional medium the equation of motion can be written in terms of the components of the symmetric Cartesian stress tensor as

$$\rho \ddot{u}_i = f_i + \sigma_{ij,j} \quad (1)$$

where ρ is the density of the medium, \ddot{u}_i and f_i , the i th component of the particle acceleration $\ddot{\mathbf{u}}$ and body force, respectively, σ_{ij} , the ij th element of the stress tensor $\boldsymbol{\sigma}$. A dot over a symbol indicates a time derivative and a comma between subscripts denotes a spatial derivative (e.g. $u_{ij} = \partial u_i / \partial x_j$). A summation convention over repeated subscripts is assumed. The weak form of Eq. (1) is obtained using the virtual work principle over domain V with boundary Γ (e.g. [3]) as:

$$\int_V \delta \boldsymbol{\epsilon}^T : \boldsymbol{\sigma} dV - \int_V \delta \mathbf{u}^T \mathbf{f} dV - \int_\Gamma \delta \mathbf{u}^T \mathbf{t} d\Gamma = 0 \quad (2)$$

where V and Γ are the volume and the surface area of the domain under study, respectively. $\delta \boldsymbol{\epsilon}$ is the virtual strain tensor related to the virtual displacement vector $\delta \mathbf{u}$, \mathbf{f} is the body force vector and \mathbf{t} is the traction vector acting on Γ . T denotes the transposed symbol. Eq. (2) is valid for linear as well as nonlinear stress–strain relationships. Within a Galerkin formulation, we consider that the basis functions used to express test function $\delta \mathbf{u}$ is the same as that used to express the unknowns displacement field.

3.2. Absorbing boundary conditions – Paraxial elements

In order to avoid wave reflections at the boundaries of the domain, paraxial elements are used on these boundaries. We use the simple approximation introduced by Engquist and Majda [25] such that the stress applied by a wave impinging on a boundary (generally noted \mathbf{t} in Eq. (2)) is approximated by

$$\mathbf{t}(x_1, x_2, x_3, t) = \begin{pmatrix} -\rho\beta\partial_t u_1 \\ -\rho\beta\partial_t u_2 \\ -\rho\alpha\partial_t u_3 \end{pmatrix}. \quad (3)$$

where (x_1, x_2, x_3) is the local coordinate of a paraxial element and α and β are the P and S-wave velocity, respectively. For this approximation, the elastodynamics dispersion relation is well approximated only when the direction of propagation of the wave is close to the normal of the edge of a paraxial element, or at high frequency [26]. Let us mention that better absorbing boundary conditions have been developed such as CPML [27] but are still not included in our the code.

3.3. Nonlinear constitutive model

Two broad classes of soil models are used in the literature:

- equivalent linear models, based on a viscoelastic constitutive model.
- cyclic nonlinear models, based mainly on an elastoplastic constitutive model.

The equivalent linear approach is most commonly used in practice. It assumes that a multi-layered soil subjected to a symmetric cyclic shear loading exhibits a hysteresis loop (see Fig. 1, e.g. [28]), which relates the shear stress τ to the cyclic distortion γ . This hysteresis loop is first characterized by the secant shear modulus G_{sec} , which represents the loop inclination:

$$G_{sec} = \frac{\tau_c}{\gamma_c} \quad (4)$$

where τ_c and γ_c are the shear stress and shear strain amplitude, respectively. The loop area A_{loop} represents the energy dissipation and is conveniently described by the damping ratio ξ , given by:

$$\xi = \frac{1}{2\pi} \frac{A_{loop}}{G_{sec}\gamma_c^2} \quad (5)$$

Parameters G_{sec} and ξ are often referred to as *equivalent linear* material parameters. The equivalent linear procedure then consists in providing $G - \gamma$ and $\xi - \gamma$ curves, expressing the evolution of both parameters with respect to the cyclic distortion. Such a linear procedure is hence not capable of predicting permanent strains or failure for high seismic distortion levels. Nevertheless, the same assumption allows a very efficient class of computational models to be used for earthquake engineering.

On the contrary, cyclic nonlinear models, which mainly considered an elastoplastic constitutive behavior for soil deposits, are able to reproduce the intrinsic complex features of soil behavior under seismic loading in a wide range of shear strains, namely from 10^{-6} to 10^{-2} , such as stiffness degradation, irrecoverable displacement, volumetric strain generation, etc. The use of models based on the elastoplasticity theory is more suitable than an equivalent-linear approach as they represent a rational mechanical process. A variety of nonlinear models have been developed for this purpose, which are characterized by

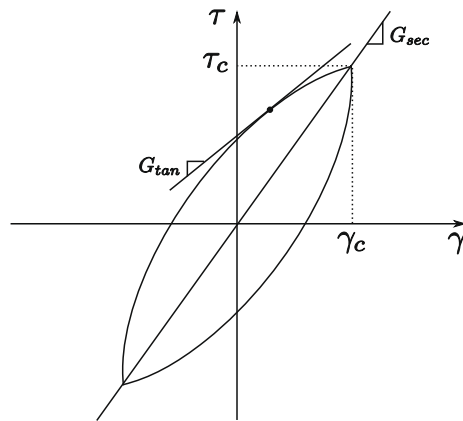


Fig. 1. Definition of parameters of an equivalent linear model.

a backbone curve and a series of rules that govern unloading-reloading behavior. More details on such models can be found for instance in [28]. Some other advanced constitutive models use the critical state concept, with one or more yield stress conditions depending on the loading type (monotonous or cyclic), represented as yield surfaces in the stress space, in order to describe the limit between linear elastic and inelastic domain behavior. Some models also propose progressive mobilization of plasticity through strain hardening mechanisms and specific flow rules that relate the plastic volumetric and shear strain rates to the stress state through plastic multipliers see (e.g. [29,30] among others).

In such advanced models, parameters should be chosen such that they are closely related to the rheology that describes the material properties at various strain levels. In some cases these rheological models do not necessarily have physical parameters. Sometimes there are indirect parameters that cannot be measured in the laboratory. Thus, one of the obstacles in using such models is the difficulty in identifying their parameters. In addition, the lack of knowledge of soil properties is common in seismic studies and a complete geotechnical description of a site is very rare. In this study, we use a rigid-perfectly plastic model (the so-called Mohr–Coulomb model), in order to circumvent these difficulties. Fig. 2 (left) shows the yield surface in the Mohr plane, characterized by the cohesion c and the internal friction angle φ . Fig. 2 (right) describes the shape of a hysteresis loop in the plane $\gamma - \tau$. Thus, by using this model, only two new parameters (i.e. c and φ) are needed in addition to the elastic parameters.

4. The GEFDYN parallel finite-element software package

4.1. Parallel implementation strategy

4.1.1. Outline of the computational procedure

In the case of a nonlinear simulation, an explicit numerical scheme could be considered but with the difficulty of selecting the right time step and controlling the error. Automatic time-stepping selection strategies are described for instance in

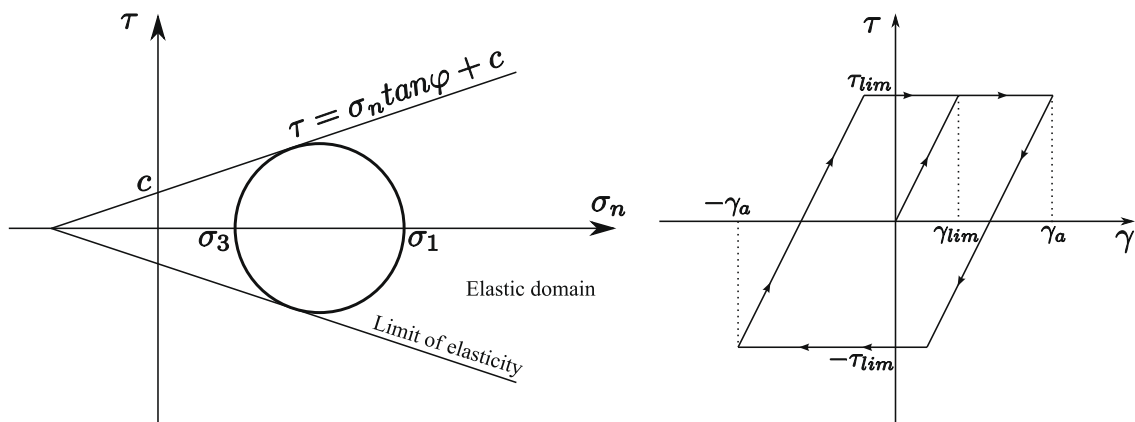


Fig. 2. Left: Schematic representation of Mohr–Coulomb criteria in the plane $\sigma_n - \tau$ (i.e. normal stress – shear stress). The cohesion is defined by the letter c and the limit of elasticity is defined by the straight line $\tau = \sigma_n \tan \varphi + c$. σ_1 and σ_3 are major and minor principal stress defining the Mohr circle. Right: Schematic representation of the nonlinear stress–strain constitutive law. Arrows indicate the stress–strain path of a full hysteresis under sinusoidal cyclic loading. Plasticity arises when the stress state reaches the limit of elasticity of Mohr–Coulomb criteria.

[31,24]; and are based mainly on a trial and error procedure with several global evaluations of the displacement vector based on different numerical schemes (in order to make sure that different time stepping procedures give a very similar result, thus making sure that they are all stable and accurate). The very significant additional cost of computing the displacement vector several times based on different time schemes would severely lower the expected speedup in our application. Moreover, explicit methods could also drift from equilibrium and become unstable, and a similar technique would then be needed as a correction.

Our numerical method associated with the parallel resolution of Eq. (1) is described in Fig. 3. The dynamic nonlinear problem is discretized based on an implicit numerical scheme and the Newmark constant average acceleration method. Based on the initial stress method [4], the nonlinear stress–strain relationship is solved with an incremental formulation. A modified Newton–Raphson loop is introduced with successive evaluations of nonlinear vector force for the equilibrium iterations. In the context of large-scale computations, one of the main advantages of our approach is to compute the tangential stiffness matrix only once, at the beginning of the computation.

Re-assembly of the stiffness matrix could accelerate convergence because it corresponds to the full Newton–Raphson method with quadratic convergence. For instance, with non-associative elastoplasticity, the stiffness matrix (though initially symmetric) can become non-symmetric because the flow rule is not associative with the yield function and consequently, non-symmetric solvers are required [3]. This kind of problem is intrinsic to the use of non-associative elastoplasticity and exists in other domains such as mechanical engineering [32]. In our case, we choose a dilatancy angle equal to the internal friction angle so that the law is associated (a dilatancy angle different from the internal friction angle would lead to a non-associative law) and the numerical operator leads to symmetric positive definite matrices. However, even if an associative law is used in our simulations to avoid non-symmetric systems, the condition number of the matrix at the elemental level (defined as the quotient between the largest and smallest eigenvalues of a matrix) increases owing to the fact that the bedrock is assumed to behave linearly whereas the soft sediments behave nonlinearly. As a result, when the soft sediments enter a nonlinear regime, their stiffness decreases locally in the physical domain; which results in locally large condition numbers.

4.1.2. Sparse linear solver

As we compute the tangential stiffness matrix at the beginning of the computation, direct linear solvers are then the best option because of their ability to compute the factorized matrix once and for all and store the coefficients to later perform forward and backward substitutions. Iterative solvers have demonstrated their efficiency in several domains, but finding the suitable preconditioner could require extensive experiments and possibly fail in some cases [33,34]. Emerging hybrid solvers (e.g. [35,36]) seem to be a promising alternative. Their numerical efficiency for our class of problem needs to be investigated. Moreover, their scalability on large-scale applications has not been extensively studied yet. For instance, one important issue

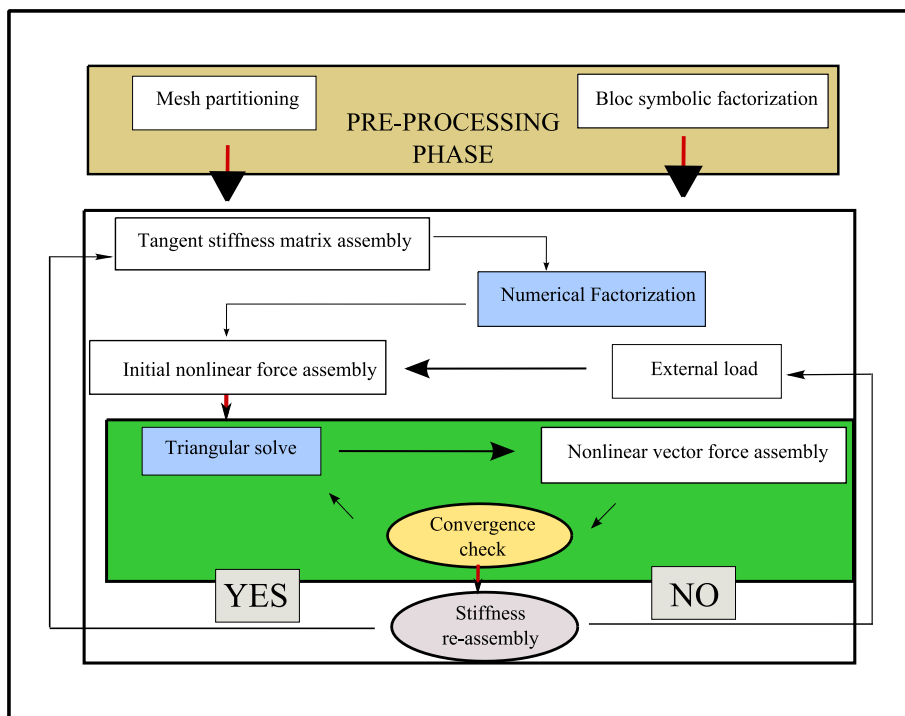


Fig. 3. Computational diagram of the GEFDYN parallel software package. In green we show the iterative Newton–Raphson phase and in blue the computational phase related to sparse matrix computations.

of a hybrid solver is scalability because there is a balance between the improvement coming from the increase of the number of processor cores (mainly for the part of the problem solved by direct solvers) and the worsening of the preconditioning phase because of the increase of the number of subdomains (the interface problem being solved by an iterative approach).

Owing to their robustness, direct resolutions are often used in industrial codes despite their memory consumption and they are very efficient for problems for which many or multiple right-hand-side solutions are required. In addition, the factorizations used nowadays in direct solvers can take advantage of the superscalar capabilities of modern processors by using blockwise algorithms and BLAS 3 primitives (Basic Linear Algebra Subprograms). The efficiency of sparse direct solvers has been recently demonstrated on thousands of cores [37]. In the case of the PaStiX solver, problem sizes of 83 millions of unknowns have been successfully handled. In our case the limited knowledge that the geological community has of the geological structure of the French Riviera prevents us from refining our model beyond a certain limit and we therefore do not expect to go beyond a problem with 83 millions of unknowns. This remark is true for the vast majority of three-dimensional sedimentary basin studies focusing on inelastic site effects because the shape and the structure of the sedimentary basin is often poorly known from a geological point of view.

The performance of the global simulation is driven by:

- The ability to get good performance during the parallel assembly phase. It corresponds to the evaluation of the nonlinear stress–strain relationship.
- The parallel performance of the sparse matrix computations. This includes factorization as well as forward and backward solve.

The methodology introduced in our article is pseudo-explicit in terms of parallel performance because we perform only a few numerical factorizations (thanks to the modified Newton–Raphson algorithm) and we mainly perform a parallel vector assembly. The main additional phase is the forward/backward solve that exhibits rather good scalability.

4.2. Parallel assembly

4.2.1. Mesh-partitioning approach

One widely-used strategy to implement parallel finite-element computations is to split the global mesh into several chunks distributed over the different computing units. The tradeoff is therefore to balance the number of elements assigned to each participating processor and to minimize the connectivity between the different subdomains, corresponding to explicit communications during the assembly phase. To perform that, METIS³ [38] or related software packages are generally used to provide a balanced decomposition. Two different strategies are generally used: node cut or element cut [39]. In the first one, the domain is divided based on a cut through the nodes. This leads to uniqueness of the assignment of elements. On the other hand, the element-cut algorithm is based on duplication of the elements of the mesh and unique attribution of the nodes to the different subdomains. The total number of elements computed by the processors is therefore greater than the initial size of the problem.

References [40,39], that discuss the performance of these approaches underline the impact of the size of the domain compared with the number of processors used. For instance, the surface to volume ratio is not favorable to the element-cut approach as the size of the buffer-zone (halo) required to ensure independent computations per subdomain will significantly grow with the number of processors. In the context of the element-cut methodology, the cost of the exchange of contributions could however easily be overcome based on overlapping of communication by computations. We report results on the parallel behavior of both algorithms for homogeneous and heterogeneous problems.

4.2.2. Graph-coloring based methodology

The previously described strategies are based on the assumption of equivalent cost for all the elements. The situation is rather different in our modeling because the geological layers have different constitutive laws associated with a different CPU cost.

A classical mesh partitioning approach leads to limited performance for this class of problems because the near-surface sedimentary layer is modeled with more complexity in terms of mechanical behavior (nonlinear) and the deep layers with less complexity (linear). Even if we assume an equal repartition of the load in the sedimentary layer, a weighted graph cannot be used because the nonlinear effects are not homogeneous in time or in space. The large number of references including dynamical load balancing in various areas of computational physics shows the popularity and the efficiency of such techniques [41]. However we can try to take advantage of the nature and the geometry of the problem to design another partitioning solution. The key pieces of information needed are the following:

- The deep geological layers exhibit uniform behavior in terms of CPU cost per element.
- Nonlinearity will arise in the sediments layer only, which means that these layers will be responsible for the load imbalance in the simulation.

³ <http://www.glaros.dtc.umn.edu/gkhome/views/metis/>.

- Based on physical and geometrical considerations, at the same time step two neighboring points will have a higher probability to have the same mechanical behavior than two distant points, at the same time step.

The idea is to consider a two-level approach for the decomposition of our problem. For the deepest layer of our geological model, which represents the main part of the computational domain and which we handle based on a mesh partitioning strategy, we use classical nonblocking communications fully overlapped by computing the outer elements (i.e. the edge elements) first, then starting the non blocking MPI calls, and finally computing the inner elements while the communications are traveling across the network.

Algorithm 1. Mesh coloring algorithm.

```

for  $i = 1$  to total candidate elements
  for  $i = 1$  to total colors
    compute  $\sum distance^2$  for elements marked with current color
    compute balancing in case of assignment to current colors
  endfor
  Restriction to the top quarter of the best color in terms of distance.
  Pick best color in terms of balancing satisfying previous criteria.
endfor

```

The sedimentary layers located near the surface are decomposed based upon graph-coloring of Algorithm 1, whose goal is to scatter neighboring elements but also to balance the number of elements between processors. One could also use more general graph-coloring theory to tackle this problem but the simple algorithm we suggest provides satisfactory results. The main difficulty is to express a tradeoff between balancing the number of elements and honoring the scattering criterion. Each candidate element for a color is supposed to maximize the distance with all the other elements marked with this color, the distance being expressed between barycenters. We limit the list of candidate colors based on a distance criteria to the best top quarter and we pick the most efficient color candidate in terms of load balancing. This colored partitioning induces a non-overlapping communication scheme because no interior or frontier elements can be defined.

The *MPI_Allgather* call is used in order to exchange contributions between the different processors (i.e. for the near-surface geological layer, which has a small size compared to the main layer described above). The main reason for using it comes from the algorithm used, which minimizes the connection between the elements on the same processor core (as opposed to graph-partitioning libraries such as METIS or SCOTCH⁴ that are widely used in the literature to distribute a finite-elements mesh). This means that we maximize the number of communications required to build the global vector with summation of the relevant contributions. This methodology is satisfactory because we perform computations on a relatively modest vector size (the memory cost is of the order of tens of megabytes). In such a case, sending all the contributions to the target processor using point-to-point communications would be far more costly and with no opportunity for overlap.

The increase of the number of processors involved in the collective communications is balanced by the decrease of the size of the data to be exchanged. This leads to a nearly constant cost, which is very small compared to the cost of the computations required to handle the nonlinear constitutive law computations. Fig. 4 gives an overview of the sedimentary layer partitioning with mesh-based method or graph-coloring algorithm, the layers located in the linear bedrock are decomposed based on the mesh decomposition approach in both cases. This is illustrated on the right panel figure with a cut showing the linear subdomains. This methodology is efficient because our global algorithm is an explicit-like code with several evaluations of vector contributions and matrix triangular solves. In case of re-assembly of the stiffness matrix one needs to move elements between subdomains, which induces an additional cost.

4.3. Main characteristics of the PaStiX parallel sparse direct solver

Solving large sparse symmetric positive definite systems $Ax = b$ of linear equations is a crucial and time-consuming step that arises in many scientific and engineering applications. As developed in Section 4.1.2, direct solver is a good choice to handle our numerical problem. Past ix solver do not consider the factorization with dynamic pivoting but only consider matrices with a symmetric sparse pattern. In this context, the block structure of the factors and the numerical operations are known in advance and consequently allow the use of static (i.e. before the effective numerical factorization) regulation algorithms for data distribution and computational task scheduling. We can therefore develop some efficient algorithms for the direct factorization that exploit very tightly the specificities of the targeted architecture and we can reach high efficiency in terms of both runtime and memory consumption. Our solver is based on a supernodal method with a right-looking formulation which, having computed the factorization of a column-block corresponding to a node of the block elimination tree, then immediately sends the data to update the column-blocks corresponding to ancestors in the tree. In a parallel context, we can locally aggregate contributions to the same block before sending the contributions. This can significantly reduce the

⁴ <http://www.gforge.inria.fr/projects/scotch/>.

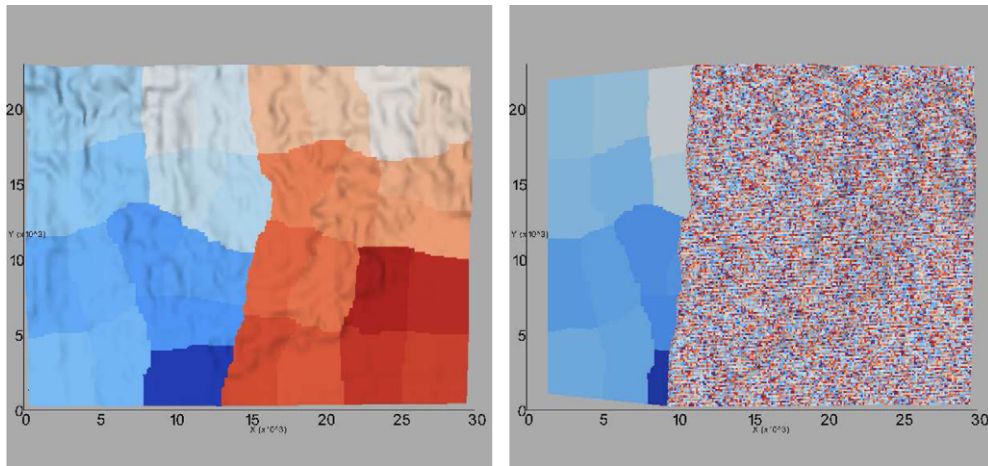


Fig. 4. Top view of the three-dimensional domain of computation – each color represents a processor: classical partitioning (left) – coloring algorithm with a cut to show the subdomains located below the sediments (right).

number of messages for exchanges between processors in a full-MPI context but leads to a prohibitive memory bottleneck caused by the local aggregation of contributions, in particular for large 3D problems [5].

On SMP-node architectures, in order to fully take advantage of shared memory, an efficient approach is to use a hybrid implementation. The rationale that motivates this hybrid implementation is that the communications within a SMP node can be advantageously replaced with direct accesses to shared memory between processors in the SMP node using threads. We can thus avoid the local aggregation overhead when a column-block modifies another one that is associated with a processor belonging to the same SMP node; in this case, the update is performed directly in shared memory without any communication [42]. This implementation efficiently reduces memory consumption because the memory bottleneck caused by the local aggregation in the full MPI implementation is reduced in a ratio proportional to the number of threads used in each SMP node. We also observe an improvement of the run-time performance and of the global scalability for the factorization. However, this hybrid implementation requires an MPI implementation that is really “thread-safe” as all threads may compute, receive and send messages via MPI for communications involving threads belonging to different SMP nodes. All these techniques are integrated in a static mapping and scheduling algorithm based on a combination of 1D and 2D block distributions.

We use MPI for inter-node communication and multithreading for intra-node communication based on explicit thread programming (for instance POSIX thread). The irregularity and the complexity of the problem prevent us from using OpenMP directives.

5. French Riviera model

We simulate seismic wave propagation from a possible seismic source to a local site located within the French Riviera, and the related ground motion response in the urban area of Nice. Left Fig. 5 shows a map view of the study area in which the main city of Nice is surrounded by a rectangle bounding box. A 3-D view of the study area is shown in the right Fig. 5, in which topography has been kept but geology has been simplified. The origin O of coordinates (x, y, z) is located at latitude/longitude $43.66^\circ/7.11^\circ$. The positive x axis goes from West to East. The size of the domain is $30 \text{ km} \times 23 \text{ km} \times 10 \text{ km}$. From bottom to top, there are four layers representing the seismological bedrock (two layers), the engineering bedrock (one layer) and the sediments (one layer). The lowermost seismological bedrock is supposed to be infinite and the thickness of the other layers is approximately: 900 m, 400 m and 100 m.

Bedrocks are assumed to have a linear elastic behavior whereas sediments are modeled using the Mohr–Coulomb criterion. Parameters that describe the elastic behaviors is presented in Table 1.

The nonlinear cohesion parameter is set to 100 kPa. In order to avoid numerical instabilities within the sediments during seismic simulations, the internal friction angle (φ) is set to zero so that the critical state line shown in Fig. 2 is parallel to the normal stress axis. This choice also allows us to skip the static stress initialization phase generally required before performing dynamic nonlinear analyses. Moreover, since the appearance of soil nonlinearities may generate high frequencies as reported by [43], two meshes of the domain have been used to accurately model maximum frequencies of 0.5 Hz and 0.6 Hz. They respectively contain 2,470,593 and 5,632,011 degrees of freedom. The earthquake source is modeled by a double-couple moment tensor (e.g. [44]). Several active faults exist around the French Riviera, we choose the Blausasc fault whose parameters are: strike = 204° , dip = 77° and rake = 15° . The shear-wave radiation pattern of this source mechanism (i.e. Fig. 6) targets directly the city of Nice, which is located along the strike of the fault. The source coordinates are $x = 18.128 \text{ km}$, $y = 10.139 \text{ km}$ and $z = -5 \text{ km}$. Its time variation is a hyperbolic tangent function whose rise time is three

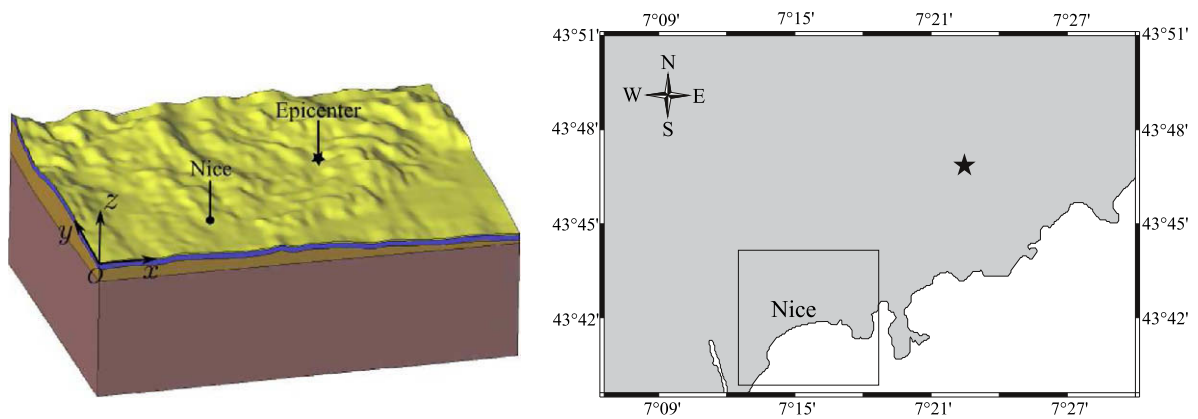


Fig. 5. 3-D view of the study area with the different geological layers (left). Map view of the study area. The rectangle shows the approximate size of the city of Nice, France. The epicenter is shown by the star (right).

Table 1
Elastic parameters of the model.

| Material | Density (kg/m ³) | Vp (m/s) | Vs (m/s) |
|-------------------------|------------------------------|----------|----------|
| Seismological bedrock 1 | 2200 | 4330 | 2500 |
| Seismological bedrock 2 | 2100 | 2598 | 1500 |
| Engineering bedrock | 2000 | 1385 | 800 |
| Sediments | 1800 | 595 | 300 |

seconds such that the maximum energy of the source spectrum is accurately modeled by the meshes of the domain, as shown in Fig. 7. In order to clearly observe the appearance of nonlinearity, we locate some receivers (also called seismic recording stations) around the maximum intensity of the southwestern lobe of the shear wave radiation pattern (i.e. close to the city of Nice).

6. Parallel performance

6.1. Description of the parallel systems

Three different computing systems are used as experimental platforms. Their main characteristics are described in Tables 2 and 3. We believe that these architectures are representative of current parallel systems mainly composed of clusters of SMP nodes. Decryphon contains IBM P575 nodes. JADE is the largest system we have used, it is an SGI (ALTIX ICE) system located at the French National Computing Center, CINES/GENCI. Borderline is a linux system with IBM x3755 nodes. We use Decryphon, Borderline and JADE platforms for scalability results, and JADE supercomputer for the memory benchmark, the parallel assembly results and the physical simulations.

6.2. Parallel assembly phase

6.2.1. Node cut and element cut approaches

In this part we present results for the assembly phase, considering classical mesh-partitioning techniques. We first consider an homogeneous problem; all the geological layers defined have the same mechanical properties and therefore we evaluate only the performance of the partitioning without any load-balancing consideration. Fig. 8 summarizes results obtained on JADE up to 1024 processors. The scalability and the imbalance are measured relatively to the performance with 32 processors.

We observe a rather good scalability for both approaches with a small advantage for the node cut partitioning. This is related to remarks in [39] on the dual nature of these methodologies and comparable parallel performance in terms of iso-efficiency. The halo required in the case of node cut introduces an imbalance that increases with the number of processors. The connectivity of the mesh and geometrical considerations explain this behavior with an average of 53% of imbalance on 1024 processors for the 0.6 Hz mesh. In Table 4, we show the ratio of extra computations required for the node cut approach. The price to pay for avoiding communications during assembly increases with the number of processors. An average of 18% is measured for the smaller case on 256 processors and 25% for the larger case on 1024 processors. These results underline the severe limitations of node cut partitioning in terms of load-balancing or of extra computations required.

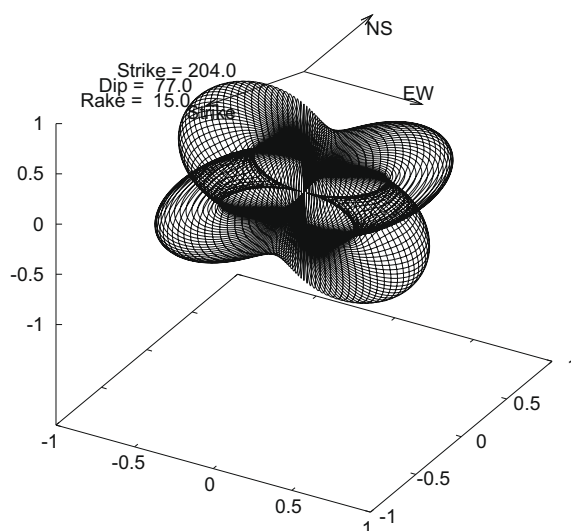


Fig. 6. Shear-wave radiation pattern in an infinite homogeneous media of the double-couple point source model used in this study.

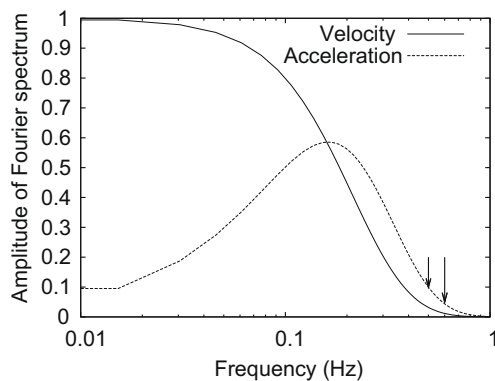


Fig. 7. Velocity and acceleration Fourier spectra of the source signal. The two arrows indicate the maximum frequency of that can be accurately computed by the two meshes: 0.5 Hz and 0.6 Hz, respectively.

Table 2

Characteristics of the parallel systems that we have used.

| Name | Number of SMP nodes | Number of processors | Interconnect |
|------------|---------------------|----------------------|----------------|
| Borderline | 10 | 80 | Infiniband |
| Decrypton | 16 | 256 | IBM-Federation |
| JADE | 1536 | 12288 | Infiniband |

Table 3

Characteristics of the SMP nodes.

| Name | Processor family | Number of processors | Cache memory (Mb) | Total memory (GB) |
|------------|------------------|----------------------|-------------------|-------------------|
| Borderline | Opteron-2218 | 8 | 2 | 30 |
| Decrypton | Power5 | 16 | 36 | 28 |
| JADE | Xeon-E5472 | 8 | 12 | 30 |

In Fig. 9, we consider a more realistic situation in which the domain is composed of heterogeneous geological layers. The results of the right table represent the percentage of load imbalance between subdomains. The heterogeneous case is based on the complete modeling of the French Riviera region but with physical parameters that prevent strong nonlinearity from appearing, contrary to the nonlinear example.

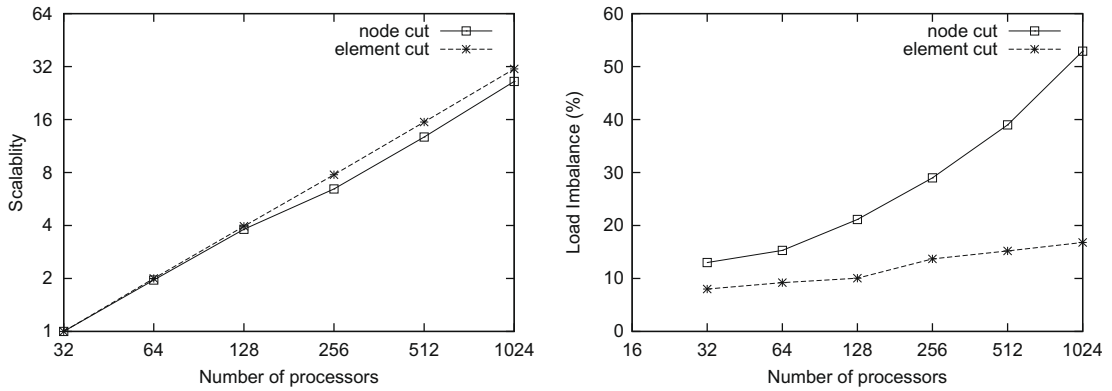
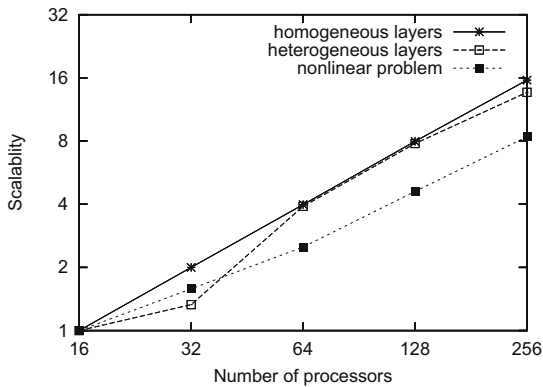


Fig. 8. Relative scalability and load imbalance for node or element cuts in the case of the 0.6 Hz mesh.

Table 4

Extra computations for the node cut approach for the 0.5 Hz mesh (left) and the 0.6 Hz mesh (right).

| CPU | Extra computations (%) | CPU | Extra computations (%) |
|-----|------------------------|------|------------------------|
| 16 | 3.51 | 64 | 6.37 |
| 32 | 5.22 | 128 | 9.22 |
| 64 | 7.87 | 256 | 13.49 |
| 128 | 11.48 | 512 | 18.87 |
| 256 | 17.42 | 1024 | 24.98 |



| CPU | Homogeneous (%) | Heterogeneous (%) | Non linear (%) |
|-----|-----------------|-------------------|----------------|
| 16 | 3.3 | 13.1 | 294 |
| 32 | 3.4 | 18.6 | 412 |
| 64 | 4.8 | 22.4 | 545 |
| 128 | 5.4 | 25.2 | 600 |
| 256 | 7.3 | 63.41 | 849 |

Fig. 9. Impact of the numerical behavior of our algorithm on load balancing in the case of the 0.5 Hz mesh.

This allows us to evaluate the impact of uniformly-distributed static overhead coming from the sedimentary layer and dynamic imbalance coming from the nonlinear mechanical property of this layer. For the heterogeneous case, we measure an imbalance of 68% on 256 processors or cores. With physical nonlinearity, the situation is worse because of the spatial distribution of the zones in which plasticity occurs in the sediment layer. The static element cut partitioning used here is not efficient for this dynamical imbalance. A ratio of more than 8 is observed between the faster and the slower subdomain computations.

6.2.2. Graph-coloring method results

In this section we provide analysis on the behavior of our coloring algorithm. All the comparisons are made with the element cut mesh partitioning method. The Figs. 10 and 11 present results on 64 processors, based on simulations of 10 s of duration corresponding to the beginning of the nonlinear phase. In Fig. 10 we measure the imbalance, at each time step, between the distributed subdomains of the sedimentary layer. We consider the mesh-partitioning algorithm in this case. This illustrates the dynamical aspect of the CPU load. We see an increase of the values close to the peak observed in the seismograms, it corresponds to the beginning of the nonlinear phase.

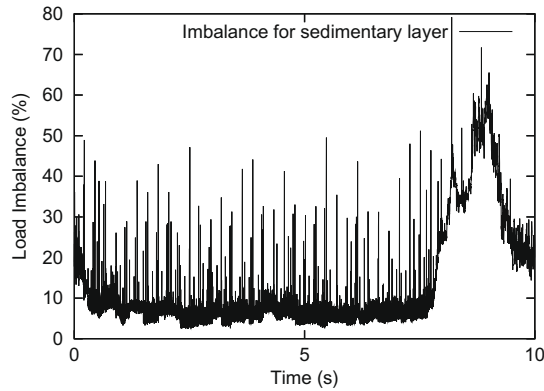


Fig. 10. Variation with time of the imbalance induced by the variation of the numerical cost in the sedimentary layer with a non linear rheology.

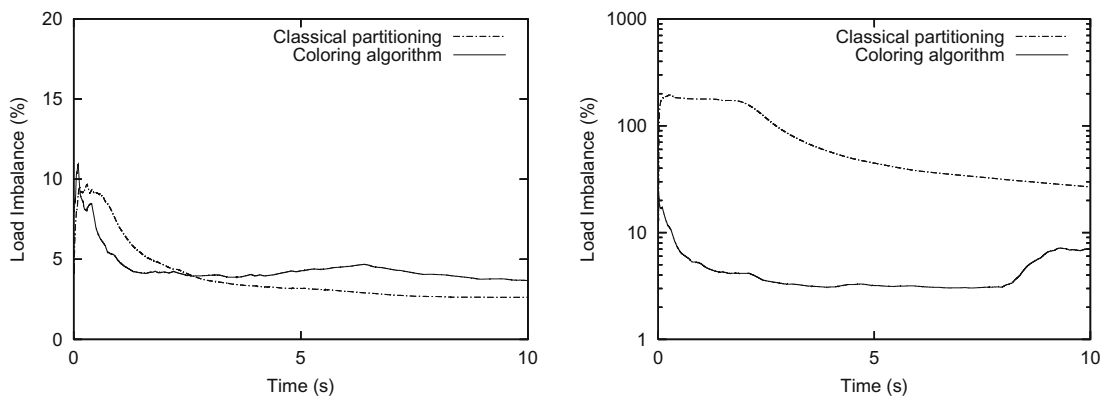


Fig. 11. Comparison of cumulative load imbalance in the non linear sedimentary layer and in the other linear geological layers.

A complementary analysis is represented in Fig. 11 with a comparison of the cumulative load imbalance during the simulation for graph-coloring or mesh-partitioning methodologies. The first plot is for the elements located in the linear bedrock, we can observe a small imbalance for both approaches, it is the expected behavior as this part of geometry is decompose in the same way. The right panel presents the imbalance for the sedimentary layer, the vertical axis is logarithmic. We observe the benefits of our coloring algorithm as the imbalance is significantly reduced. In case of coloring, the cumulative imbalance is under 25% whereas the same computation with mesh partitioning algorithm for the top layer exhibits an average of 200% of imbalance.

Table 5 illustrates the scalability of our coloring algorithm on up to 256 processors for the Nice05Hz test case. The right panel is the timing results of the gathering phase required after the RHS assembly for the sedimentary layer. We represent the relative speedup using the timing on 32 processors as a reference. The *MPI_Allgather* subroutine is called first, this operation is followed by a local summation of the relevant contributions for each subdomain. The global cost is nearly constant when we double the number of processors involved in the collective communication we also divide by nearly a factor two the size of the messages collected on each subdomain. This is coming from the distribution of the RHS vector on the different processors. Typically this phase represents a small fraction of the total assembly cost; for instance on 256 processors it represents an average of 3%. As we increase the number of processors, network contention or latency overhead could appear an further investigation required. The left panel is the comparative speedup between the elements-cut and the coloring approaches. The timing result on 32 processors for the elements-cut method is used as a reference to compute the relative speedups. A very slow decrease of the elapsed time can be noticed for the classical algorithm. This comes from the strong

Table 5
Comparative speedup of the elements-cut and coloring algorithms for the Nice05Hz test case.

| CPU | 32 | 64 | 128 | 256 |
|----------------|-----|-----|------|------|
| Elements-cut | 1 | 1.5 | 2.9 | 5.3 |
| Graph-coloring | 2.7 | 5.1 | 10.2 | 18.8 |

| CPU | 16 | 32 | 64 | 128 | 256 | 512 |
|-----------------|----|------|------|------|------|------|
| Global exchange | 1 | 1.22 | 1.12 | 0.87 | 0.90 | 0.93 |

imbalance of the CPU-cost between subdomains in this case, we have observed a maximum of 840% on 256 processors. For the coloring method, the imbalance is much more reduced and the speedup is $3.6\times$ compared to elements-cut partitioning on the same number of processors. The overall scalability of the method is rather good with a speedup of $6.9\times$ from 32 to 256 processors.

6.3. Sparse parallel direct solver

In this section we analyse the performance of the PaStiX linear solver [5] considering our French Riviera model. The characteristics of the matrices are described in Table 6, both of them are in double precision, symmetric and positive-definite. We give the dimension of the matrix (columns), the number of off-diagonal terms in the triangular part of each matrix (NNZ_A) and the number off-diagonal entries in the factorized matrix L (NNZ_L). The number of flops for the LDL^T factorization is also reported.

6.3.1. Scalability results

The first concern is the scalability of the factorization and also of the solve phase. Tables 7 and 8 show the results obtained on Borderline and Decryphon platforms and underline the SMP effects of our algorithm.

In most of the cases with a fixed number of processors, the best configuration corresponds to the use of the maximum number of threads. For example, with 32 processors on Decryphon (2 MPI processes and 16 threads for each MPI process), we can factorize our matrix in 268 s whereas the time measured for the full MPI version (with one thread per MPI process) is 283 s. We recall that the full MPI version requires much more communications for the aggregation phase. The rather small difference (5%) on Decryphon between these two approaches comes from the efficiency of the network for inter-node and intra-node communications. At the shared memory level, the optimized MPI implementation delivers good performance in comparison with multithreading for PaStiX algorithm. The differences between the SMP and MPI versions of our algorithm are far more important on the Borderline platform. For example, we measure in Table 8 a factorization time of 215 s on 64 processors with the SMP version and 775 s with the full MPI version. We can also notice a gain of 37% for the hybrid solve phase considering the same configuration. On this platform we do not benefit from optimized intra-node communications based on shared memory and therefore the multithreaded approach is far more efficient inside a given node. The impact of communications can also be analyzed for the SMP version: if we use 16 processors, the difference between using 2 or 8 threads inside the node is close to 30% on Borderline whereas the difference is only 5% on Decryphon platform.

Table 9 summarizes the timing results on JADE platform on up to 1024 processors. We consider the hybrid implementation with 8 threads on each node for both examples. For the Nice05Hz test case, we observe a speedup until 256 processors. Between 256 and 1024 processors, performance appears begin declining. The increase of the number of processors leads to a fine grain parallelism and we do not fully exploit the processors involved in the computation. The situation is better for the Nice06Hz test case because the matrix size is larger. In both cases the solve phase time decreases on up to 1024 processors.

6.3.2. Memory consumption

An important bottleneck for large-scale inelastic simulations is the memory consumption of sparse direct solvers. Another advantage of the hybrid approach implemented in the PaStiX software package is its ability to reduce memory overhead; as local data structures are allocated at the MPI process level a very large amount of memory is saved.

Table 6

Matrix characteristics from our French Riviera model.

| Name | Columns | NNZ_A | NNZ_L | Flops |
|----------|---------|-------------|------------|----------|
| Nice05Hz | 2470593 | 96.789E+06 | 3.925E+09 | 2.56E+13 |
| Nice06Hz | 5632011 | 224.147E+06 | 13.777E+09 | 1.73E+14 |

Table 7

CPU time results on decryphon platform: hybrid (left) and MPI (right) approaches for the Nice05Hz mesh.

| CPU | MPI | Threads | Factorization (s) | Solve (s) |
|-----|-----|---------|-------------------|-----------|
| 8 | 4 | 2 | 820 | 5.77 |
| 8 | 2 | 4 | 810 | 5.96 |
| 16 | 8 | 2 | 464 | 3.4 |
| 16 | 4 | 4 | 450 | 3.64 |
| 16 | 2 | 8 | 436 | 3.64 |
| 32 | 16 | 2 | 242 | 2.06 |
| 32 | 8 | 4 | 236 | 1.99 |
| 32 | 4 | 8 | 257 | 2.31 |
| 32 | 2 | 16 | 268 | 2.33 |

| CPU | Factorization (s) | Solve (s) |
|-----|-------------------|-----------|
| 8 | 877 | 5.64 |
| 16 | 454 | 3.46 |
| 32 | 283 | 2.0 |

Table 8

CPU time on the borderline: hybrid (left) and MPI (right) approaches for the Nice05Hz mesh.

| CPU | MPI | Threads | Factorization (s) | Solve (s) |
|-----|-----|---------|-------------------|-----------|
| 8 | 4 | 2 | 1240 | 3.98 |
| 8 | 2 | 4 | 1140 | 4.2 |
| 16 | 8 | 2 | 944 | 2.45 |
| 16 | 4 | 4 | 719 | 2.54 |
| 16 | 2 | 8 | 679 | 3.19 |
| 32 | 8 | 4 | 452 | 1.37 |
| 32 | 4 | 8 | 342 | 1.58 |
| 64 | 8 | 8 | 215 | 0.8 |

| CPU | Factorization (s) | Solve (s) |
|-----|-------------------|-----------|
| 8 | 2110 | 4.57 |
| 16 | 1340 | 2.45 |
| 32 | 830 | 1.57 |
| 64 | 775 | 1.28 |

Table 9

CPU time on JADE platform: hybrid approach – Nice05Hz mesh (left) and Nice06Hz mesh (right).

| CPU | Factorization (s) | Solve (s) |
|------|-------------------|-----------|
| 16 | 359 | 4.99 |
| 32 | 175 | 2.53 |
| 64 | 108 | 1.44 |
| 128 | 64.9 | 0.76 |
| 256 | 46.6 | 0.46 |
| 512 | 39.6 | 0.30 |
| 1024 | 44.6 | 0.26 |

| CPU | Factorization (s) | Solve (s) |
|------|-------------------|-----------|
| 64 | 673 | 4.62 |
| 128 | 369 | 2.57 |
| 256 | 237 | 1.44 |
| 512 | 196 | 0.91 |
| 1024 | 153 | 0.86 |

Considering a fixed number of processors, the best results are always obtained by maximizing the number of threads, it is illustrated in Table 10 with results for both test cases on 128 processors using different combination for the number of threads and the number of MPI processes. Using the best configuration, the memory is therefore divided by a factor 3.5 for the smaller matrix and by a factor 2.4 for the larger example. Fig. 12 depicts the memory consumption on up to 1024 processors. For the Nice06Hz mesh (right panel), using 256 processors on the JADE platform requires the cumulative memory available on 64 computing nodes for the MPI version (1.6 TB) whereas the hybrid implementation (278 GB) only requires the cumulative memory of 16 nodes. The storage of the matrix factors represents a decreasing part of the total memory used when we increase the number of MPI processes. It represents 112 GB for the Nice06Hz mesh and 32 GB for the Nice05Hz mesh. We can evaluate the contribution of aggregation buffers to global memory consumption: for the Nice05Hz mesh on 256 processors, the part of these buffers is 66% for the best hybrid version and 95% for the full MPI version. For the larger case, the ratio is 59% for hybrid PaStiX and 93% for the MPI implementation on 256 processors. Using multithreading inside the node restrain the increase of the memory consumption due to communication buffers. The full MPI version prevent from using all the processors on each SMP node (2 or 4 computing processors on each 8-processors node) because of the memory overhead. We have successfully performed calculations on up to 1024 processors based on the hybrid implementation.

In terms of scalability or memory efficiency the advantage of the hybrid algorithm is clear compared with the classical MPI approach both in terms of reduction of memory consumption and CPU time.

7. Physical results and analysis

7.1. Nonlinear behaviour at a specific receiver

We have performed three simulations with a different magnitude for the hypothetical earthquake. Fig. 13 shows displacement, velocity and acceleration time histories for magnitude $M_w = 5.7$, $M_w = 6.0$ and $M_w = 6.2$, respectively, for the mesh accurate up to 0.5 Hz.

For the $M_w = 5.7$ event, we clearly see that the soft sedimentary layer has the same behavior in both the linear and the nonlinear simulations (the permanent displacement is due to the near-field and/or the intermediate field terms generated by

Table 10

Detailed memory consumption on 128 processors for the hybrid approach – Nice05Hz mesh (left) and Nice06Hz mesh (right).

| CPU | MPI | threads | memory(GB) |
|-----|-----|---------|------------|
| 128 | 64 | 2 | 186 |
| 128 | 32 | 4 | 92.9 |
| 128 | 16 | 8 | 53.2 |

| CPU | MPI | threads | memory(GB) |
|-----|-----|---------|------------|
| 128 | 64 | 2 | 430 |
| 128 | 32 | 4 | 230 |
| 128 | 16 | 8 | 180 |

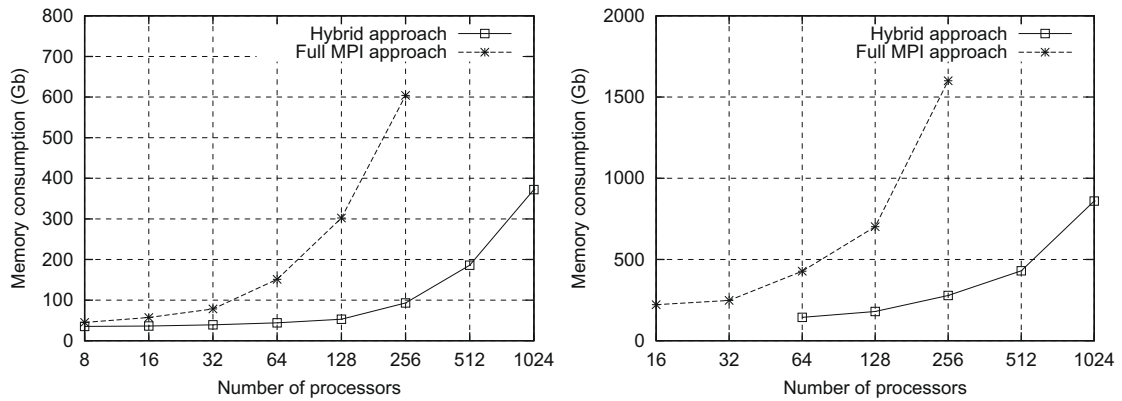


Fig. 12. Memory consumption for the hybrid and full MPI approaches – Nice05Hz mesh (left) and Nice06Hz mesh (right).

the shear dislocation). When increasing the magnitude to 6.0 and 6.2, nonlinear effects appear and increase the permanent displacement owing to the fact that the material enters the plasticity domain (i.e. γ_{lim} in Fig. 2). For the velocity, we can see for the magnitude 6.2 event that the dissipation of energy due to the start of the hysteresis decreases peak ground velocity on the shear wave portion (i.e. between 8 s and 12 s). The effect of the nonlinear constitutive law on acceleration is much more difficult to interpret; the global trend is to add an important high frequency content (in agreement with [43]) and to increase peak ground acceleration after the shear wave portion (i.e. from 12 s to the end). Since a high frequency content is added by the nonlinear behaviour of the soil, numerical dispersion may take place and the mesh accurate up to 0.5 Hz may not be able to correctly take this phenomenon into account.

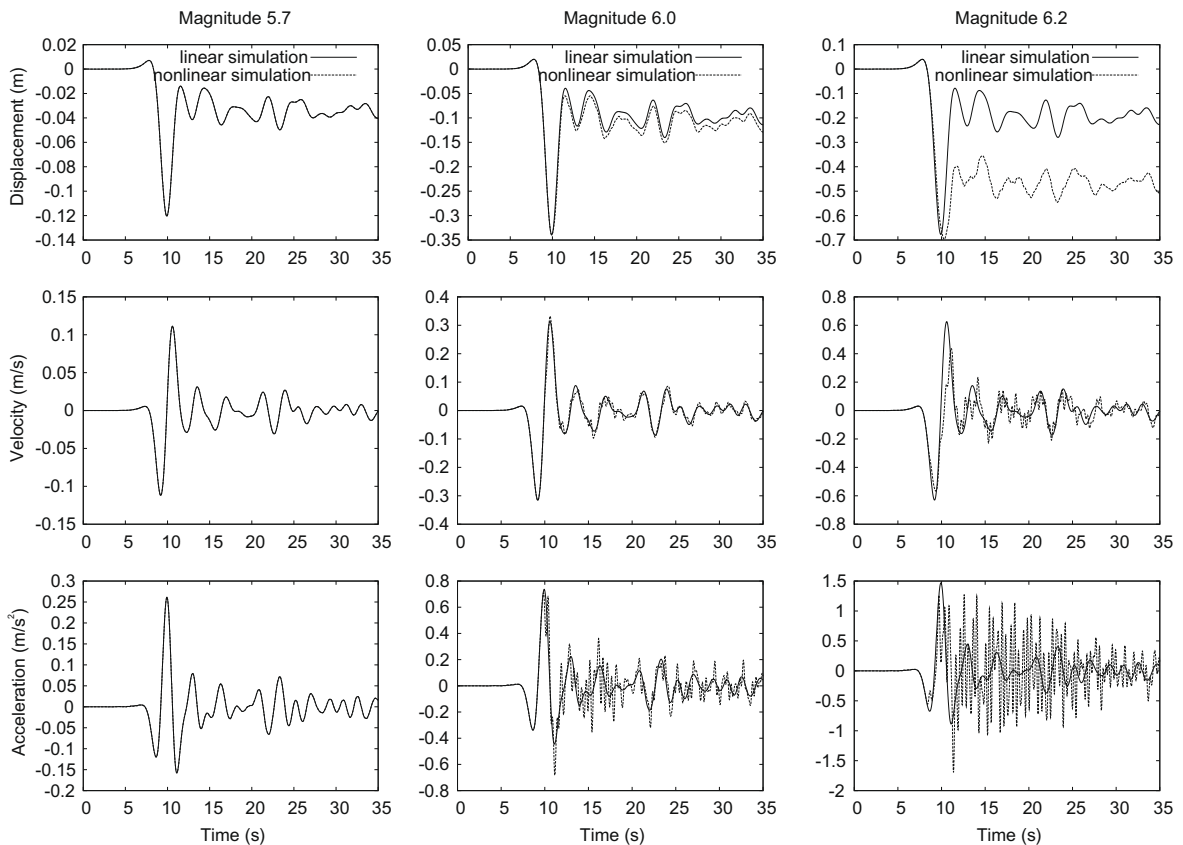


Fig. 13. East-west component of displacement (top), velocity and acceleration (bottom) at a receiver for earthquakes of magnitude 5.7 (left), 6.0 (middle) and 6.2 (right).

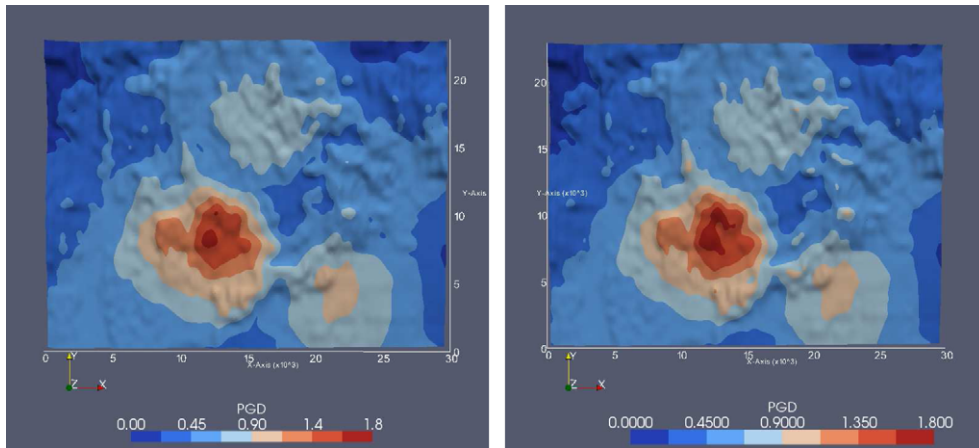


Fig. 14. Peak ground displacements during the linear (left) and nonlinear (right) simulations for the 0.6 Hz mesh.

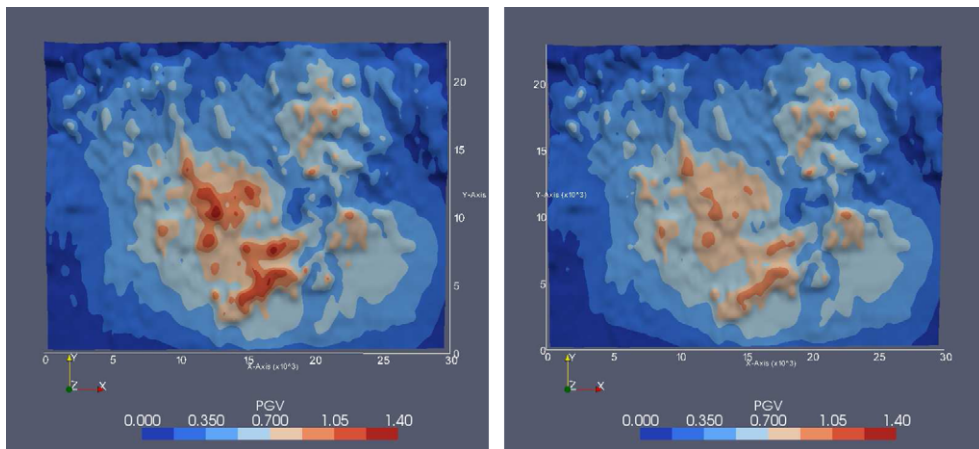


Fig. 15. Peak ground velocities during the linear (left) and the nonlinear (right) simulations for the 0.6 Hz mesh.

7.2. Global effect of nonlinearity on peak ground displacements and peak ground velocities

Figs. 14 and 15 (in the Appendix), show the Peak Ground Displacement (PGD) and the Peak Ground Velocity (PGV), respectively, for the linear and nonlinear simulations for the mesh accurate up to 0.6 Hz.

We first notice that the radiation pattern of the PGD is consistent with the theoretical radiation pattern shown in Fig. 6. We also observe that the nonlinear behaviour of the soft sediments increases the PDG, in particular in areas where the PDG of the linear simulation exhibits high values. This observation is consistent with the general behaviour we expect from nonlinear constitutive law such as the elastic perfectly plastic one used in this study (e.g. [28]).

The influence of the nonlinear behaviour on PGV is even clearer. Despite the fact that topography is present and increases PGV at the top of the hills for the linear simulation (e.g. [45]), the global effect of the constitutive law (topography being present or not) is to decrease PGV of 28% at some locations, owing to the fact that the nonlinear behaviour of the soil dissipates energy (we mention that since the value of the cohesion of the Mohr–Coulomb criterion as well as the shear wave velocity of the sediment layer are taken arbitrarily, an accurate quantitative assessment of the reduction of PGVs during a real earthquake is still difficult to model).

8. Conclusions and future work

We have presented numerical finite-element simulations of seismic wave propagation in geological media with a non linear rheology. PaStiX direct solver is used to avoid the classical convergence problems of iterative linear methods which would arise here because our media have nonlinearities, and to benefit from computations for multiple right-hand sides but having the same structure. We have underlined the impact of the hybrid programming algorithm used to overcome

memory overhead. We have also analyzed the parallel assembly, which is the other time consuming part, for different algorithms and have demonstrated the limitations of classical mesh partitioning. In order to reduce the imbalance, we have then introduced a coloring-like approach that takes advantage of the knowledge of the problem. An analysis of the behavior of our algorithms is provided on up to 1024 processors. The geology and geometry of the geological model of the French Riviera that we have used is relatively simple. In future work we could consider a more detailed model that could be used to evaluate more quantitatively strong seismic ground motion in and around the city of Nice. Another direction is to consider hybrid methods introduced for instance in [35,36] for the sparse linear solver and evaluate the potential benefits in terms of scalability or memory consumption.

Acknowledgments

This work was supported in part by the French ANR under grants NUMASIS ANR-05-CIGC and QSHA ANR-05-CTT. We acknowledge CINES/GENCI, France for providing access and support to their computing platform JADE. We also acknowledge the M3PEC computing center, France for access to the IBM Decryphon architecture. We acknowledge the Computing Center of Region Centre (CCSC) for providing access to the Phoebus computing system. We thank Xavier Lacoste and Mathieu Favergé from INRIA Bordeaux Sud-Ouest, Bacchus project for the support on the solver usage and Faiza Boulahya and Luc Frauciel from BRGM, France for discussions on graph-coloring techniques. We also thank two anonymous reviewers and the associate editor Costas Bekas for comments that helped to improve the manuscript.

References

- [1] M. Bouchon, C.A. Schultz, M.N. Tököz, Effect of three-dimensional topography on seismic motion, *J. Geophys. Res.* 101 (1996) 5835–5846.
- [2] S.J. Lee, D. Komatitsch, B.S. Huang, J. Tromp, Effects of topography on seismic wave propagation: an example from northern Taiwan, *Bull. Seismol. Soc. Am.* 99 (1) (2009) 314–325.
- [3] O.C. Zienkiewicz, R.L. Taylor, *The Finite Element Method. Basic Formulation and Linear Problems*, vol. 1, McGraw-Hill, 1989.
- [4] O.C. Zienkiewicz, S. Valliappan, I.P. King, Elasto-plastic solutions of engineering problems initial stress, finite element approach, *Int. J. Numer. Methods Eng.* 1 (1969) 75–100.
- [5] P. Hénon, P. Ramet, J. Roman, PaStiX: a high-performance parallel direct solver for sparse symmetric definite systems, *Parallel Comput.* 28 (2) (2002) 301–321.
- [6] E. Foerster, G. Courrioux, H. Aochi, F. De Martin, S. Bernardie, Seismic hazard assessment through numerical simulation at different scales : application to Nice city (French Riviera), in: *Proceedings of the 14th World Conference on Earthquake Engineering*, October 12–17, Beijing, China, 2008, p. 4.
- [7] R. Madariaga, Dynamics of an expanding circular fault, *Bull. Seismol. Soc. Am.* 66 (3) (1976) 639–666.
- [8] J. Lysmer, L.A. Drake, A finite element method for seismology, in: B. Alder, S. Fernbach, B.A. Bolt (Eds.), *Methods in Computational Physics*, vol. 11, Academic Press, New York, USA, 1972, pp. 181–216 (Chap. 6).
- [9] E. Tessmer, D. Kosloff, 3-D elastic modeling with surface topography by a Chebyshev spectral method, *Geophysics* 59 (3) (1994) 464–473.
- [10] D. Komatitsch, J.P. Vilotte, The spectral-element method: an efficient tool to simulate the seismic response of 2D and 3D geological structures, *Bull. Seismol. Soc. Am.* 88 (2) (1998) 368–392.
- [11] C. Riyanti, Y. Erlangga, R.-E. Plessix, W. Mulder, C. Vuik, C. Oosterlee, A new iterative solver for the time-harmonic wave equation, *Geophysics* 71 (2006) E57–E63.
- [12] F. Sourbier, S. Operto, J. Virieux, P.R. Amestoy, J.-Y. L'Excellent, FWT2D: a massively parallel program for frequency-domain full-waveform tomography of wide-aperture seismic data – part 1: Algorithm, *Comput. Geosci.* 35 (3) (2009) 487–495.
- [13] J. Tromp, D. Komatitsch, Q. Liu, Spectral-element and adjoint methods in seismology, *Commun. Comput. Phys.* 3 (2008) 1–32.
- [14] C. Tape, Q. Liu, A. Maggi, J. Tromp, Adjoint tomography of the Southern California crust, *Science* 325 (5943) (2009) 988–992.
- [15] T. Furumura, L. Chen, Parallel simulation of strong ground motions during recent and historical damaging earthquakes in Tokyo, Japan, *Parallel Comput.* 31 (2) (2005) 149–165.
- [16] D. Komatitsch, S. Tsuboi, C. Ji, J. Tromp, A 14.6 billion degrees of freedom, 5 teraflops, 2.5 terabyte earthquake simulation on the Earth Simulator, in: *Proceedings of the 2003 ACM/IEEE Conference on Supercomputing*, November 15–21, Phoenix, Arizona, USA, p. 4.
- [17] D. Komatitsch, J. Labarta, D. Michéa, A simulation of seismic wave propagation at high resolution in the inner core of the Earth on 2166 processors of MareNostrum, *Lecture Notes in Comput. Sci.* 5336 (2008) 364–377.
- [18] V. Akcelik, J. Bielak, G. Biros, I. Epanomeritakis, A. Fernandez, O. Ghattas, E.J. Kim, J. Lopez, D. O'Hallaron, T. Tu, J. Urbanic, High resolution forward and inverse earthquake modeling on terascale computers, in: *Proceedings of the 2003 ACM/IEEE Conference on Supercomputing*, November 15–21, Phoenix, Arizona, USA, 2003, p. 52.
- [19] C. Burstedde, M. Burtsher, O. Ghattas, G. Stadler, T. Tu, L.C. Wilcox, ALPS: a framework for parallel adaptive PDE solution, *J. Phys.: Conf. Ser.*, vol. 180, 2009.
- [20] P.B. Schnabel, J. Lysmer, H.B. Seed, SHAKE: a computer program for earthquake response analysis of horizontally-layered sites, University of California, Berkeley, USA, Tech. Rep., 1972, Report No. UCB/EERC-72/12.
- [21] E. Foerster, H. Modaressi, Nonlinear numerical method for earthquake site response analysis II- case studies 5 (3) (2007) 325–345.
- [22] I.M. Idriss, H.B. Seed, Seismic response of horizontal soil layers, *J. Soil Mech. Found. Division, ASCE* 94 (1968) 1003–1031.
- [23] J. Xu, J. Bielak, O. Ghattas, J. Wanga, Three-dimensional nonlinear seismic ground motion modeling in inelastic basins, *Phys. Earth Planetary Interiors* 137 (2003) 81–95.
- [24] A.J. Abbo, Finite element algorithms for elastoplasticity and consolidation, Ph.D. Dissertation, University of Newcastle, United Kingdom, 1997.
- [25] B. Engquist, A. Majda, Absorbing boundary conditions for numerical simulation of waves, *Math. Comput.* 31 (139) (1977) 629–651.
- [26] H. Modaressi, Modélisation numérique de la propagation des ondes dans les milieux poreux anélastiques, Ph.D. Dissertation, École Centrale de Paris, France, 1987.
- [27] D. Komatitsch, R. Martin, An unsplit convolutional perfectly matched layer improved at grazing incidence for the seismic wave equation, *Geophysics* 72 (5) (2007) SM155–SM167.
- [28] S. Kramer, *Geotechnical Earthquake Engineering*, Prentice-Hall, 1996.
- [29] Z. Mroz, On the description of anisotropic work hardening, *J. Mech. Phys. Solids* 15 (1967) 163–175.
- [30] D. Aubry, J.C. Hujeux, F. Lassoudire, Y. Meimon, A double memory model with multiple mechanisms for cyclic soil behaviour, in: *Proceedings of the International Symposium Num. Mod. Geomech (NUMOD)*, Balkema, 1982, pp. 3–13.
- [31] L.X. Luccioni, J.M. Pestana, R.L. Taylor, Finite element implementation of non-linear elastoplastic constitutive laws using local and global explicit algorithms with automatic error control, *Int. J. Numer. Methods Eng.* 50 (5) (2001) 1191–1212.
- [32] X. Lei, C.J. Lissenden, Pressure sensitive nonassociative plasticity model for DRA composites, *J. Eng. Mater. Technol.* 129 (2) (2007) 255–264.

- [33] J. Lu, J. Peng, A. Elgamal, Z. Yang, K.H. Law, Parallel finite element modeling of earthquake ground response and liquefaction, *Earthquake Eng. Eng. Vib.* 3 (2004) 23–37.
- [34] T. George, A. Gupta, V. Sarin, An empirical analysis of iterative solver performance for spd systems, IBM T.J. Watson Research Center, Tech. Rep. RC24737, 2009.
- [35] J. Gaidamour, P. Hénon, A parallel direct/iterative solver based on a Schur complement approach, in: *Proceedings of the IEEE International Conference on Computational Science and Engineering (CSE'08)*, July 16–18, São Paulo, Brazil, pp. 98–105.
- [36] A. Haidar, On the parallel scalability of hybrid linear solvers for large 3D problems, Ph.D. Dissertation, INPT Toulouse, France, 2008.
- [37] A. Gupta, S. Koric, T. George, Sparse matrix factorization on massively parallel computers, IBM T.J. Watson Research Center, Tech. Rep. RC24809, 2009.
- [38] G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, *SIAM J. Sci. Comput.* 20 (1) (1998) 359–392.
- [39] B. Patzák, D. Rypl, Z. Bittnar, Parallel explicit finite element dynamics with nonlocal constitutive models, *Comput. Struct.* 79 (2001) 2287–2297.
- [40] J. Lu, Parallel finite element modeling of earthquake site response and liquefaction, Ph.D. Dissertation, University of California, San Diego, USA, 2006.
- [41] M.A. Bhandarkar, L.V. Kalé, E. de Sturler, J. Hoeflinger, Adaptive Load Balancing for MPI Programs, in: *Computational Science – ICCS 2001*, International Conference, San Francisco, CA, USA, May 28–30, 2001, pp. 108–117.
- [42] P. Hénon, P. Ramet, J. Roman, On using an hybrid MPI-Thread programming for the implementation of a parallel sparse direct solver on a network of SMP nodes, in: *Proceedings of the Sixth International Conference on Parallel Processing and Applied Mathematics*, Workshop HPC Linear Algebra, Poznan, Poland, *Lectures Notes in Computer Science* 3911, 2005, pp. 1050–1057.
- [43] I. Beresnev, K. Wen, Nonlinear soil response – a reality?, *Bull Seism. Soc. Am.* 86 (6) (1996) 1964–1978.
- [44] K. Aki, P.G. Richards, *Quantitative Seismology*, second ed., University Science Books, 2002.
- [45] M. Bouchon, Effect of topography on surface motion, *Bull. Seism. Soc. Am.* 63 (3) (1973) 615–632.