3D Parallel Elastodynamic Modeling of Large Subduction Earthquakes

Eduardo Cabrera[1], Mario Chavez[2,3], Raúl Madariaga[3], Narciso Perea[2],
and Marco Frisenda[3]

[1] Supercomputing Dept., DGSCA, UNAM, C.U., 04510, Mexico DF, Mexico
[2] Institute of Engineering, UNAM, C.U., 04510, Mexico DF, Mexico
[3] Laboratoire de Géologie CNRS-ENS, 24 Rue Lhomond, Paris, France
eccf@super.unam.mx, marioch48@hotmail.com,
raul.madariaga@ens.fr, narpere@hotmail.com, mfrisenda@yahoo.it

Abstract. The 3D finite difference modeling of the wave propagation of M>8 earthquakes in subduction zones in a realistic-size earth is very computationally intensive task. We use a parallel finite difference code that uses second order operators in time and fourth order differences in space on a staggered grid. We develop an efficient parallel program using message passing interface (MPI) and a kinematic earthquake rupture process. We achieve an efficiency of 94% with 128 (and 85% extrapolating to 1,024) processors on a dual core platform. Satisfactory results for a large subduction earthquake that occurred in Mexico in 1985 are given.

Keywords: Elastodynamic, modeling, earthquakes, parallel computing.

1 Introduction

The 19/09/1985 a large Ms 8.1 subduction earthquake occurred on the Mexican Pacific coast with an epicenter at about 340 km from Mexico City is shown in Fig. 1A. The rupture area of this event of about 180 x 100 km is also shown in this figure. In Fig. 1B, a profile from the Mexican coast and beyond Mexico City shows the tectonic plates involved in the generation of this type of earthquakes in Mexico. Finally, the kinematic representation of the average slip associated to the mentioned earthquake is presented in Fig. 1C. As the recurrence time estimated for this highly destructive type of events in Mexico is of only a few decades, there is a seismological and engineering interest in modeling them [1].

Herewith, we developed an efficient parallel program using message passing interface (MPI) with a kinematic specification of the rupture process in the fault. In Ch. 2 we synthesize the elastodynamics of the problem; the data parallelism approach decomposition proposed and the MPI implementation are presented in Ch. 3. The study of the efficiency of the proposed parallel program is discussed in Ch. 4 and in Ch. 5 results obtained for the modeling of the seismic wave propagation of the 19/09/1985 Ms 8.1 subduction earthquake are given.

F. Cappello et al. (Eds.): EuroPVM/MPI 2007, LNCS 4757, pp. 373–380, 2007.

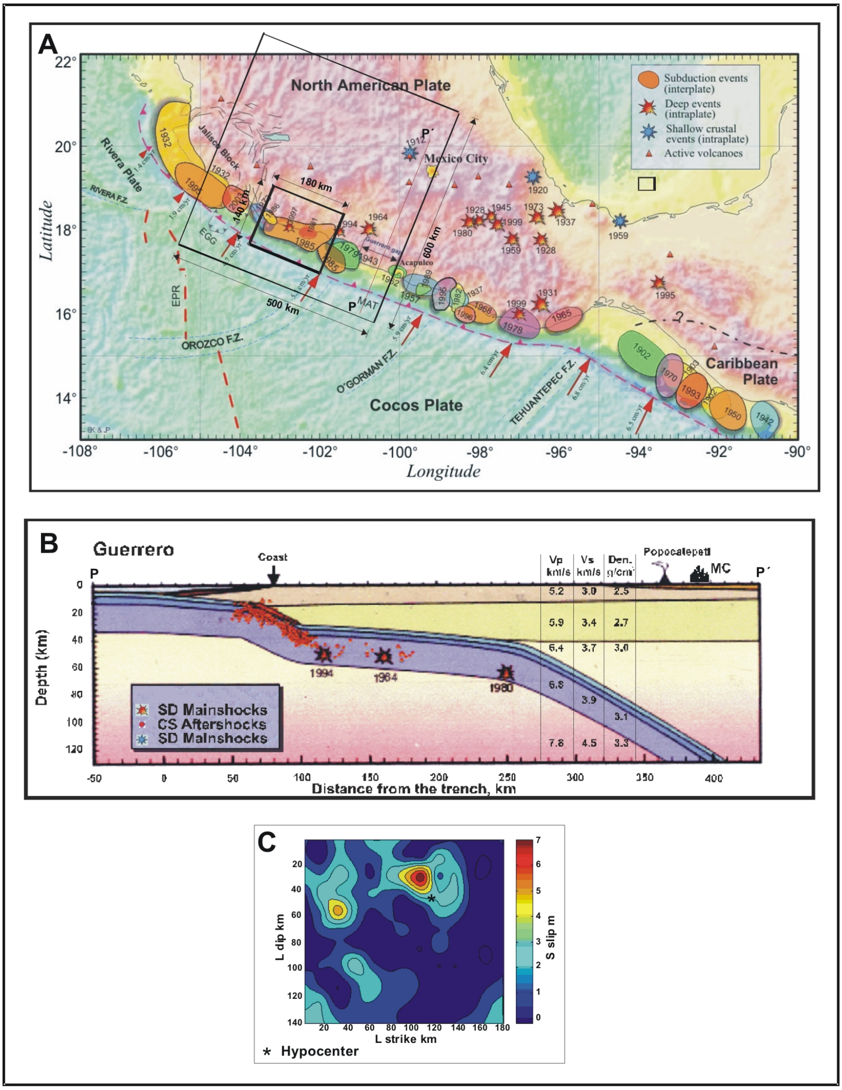

Fig. 1. A) Inner rectangle is the rupture area of the 19/09/1985 Ms 8.1 earthquake on surface projection of the 500x600x124 km earth crust volume 3DFD discretization; B) profile P-P´; C) Kinematic slip distribution of the rupture of the 1985 earthquake [4]

2 Elastodynamics and the 3DFD Algorithm

A synthesis of the elastodynamic formulation and its algorithm description of the elastic wave propagation problem are presented by following [2]. The elastic wave equation in a 3D medium occupying a volume V and boundary S, the medium may be described using Lamé parameters $\lambda(\overline{x})$ and $\mu(\overline{x})$ and mass density $\rho(\overline{x})$, where

$(\overline{x}) \in \Re^3$. The velocity-stress form of the elastic wave equation consists of nine coupled, first order partial differential equations for the three particle velocity vector components $v_{ij}(\overline{x},t)$ and the six independent stress tensor components $\sigma_{ij}(\overline{x},t)$, where i,j = 1,2,3 and assuming that $\sigma_{ij}(\overline{x},t) = \sigma_{ji}(\overline{x},t)$:

$$\frac{\partial v_i(\overline{x},t)}{\partial t} - b(\overline{x})\frac{\partial \sigma_{ij}(\overline{x},t)}{\partial x_j} = b(\overline{x})\left[f_i(\overline{x},t) + \frac{\partial m^a_{ij}(\overline{x},t)}{\partial x_j} \right]. \tag{1}$$

$$\frac{\partial \sigma_{ij}(\overline{x},t)}{\partial t} - \lambda(\overline{x})\frac{\partial v_k(\overline{x},t)}{\partial x_k}\delta_{ij} - \mu(\overline{x})\left[\frac{\partial v_i(\overline{x},t)}{\partial x_j} + \frac{\partial v_j(\overline{x},t)}{\partial x_i} \right] = \frac{\partial m^s_{ij}(\overline{x},t)}{\partial t} \tag{2}$$

where $b=1/\rho$, and f_i is the force source tensor and $m^a_{ij}(\overline{x},t) = 1/2\left[m_{ij}(\overline{x},t) - m_{ji}(\overline{x},t)\right]$, $m^s_{ij}(\overline{x},t) = 1/2\left[m_{ij}(\overline{x},t) + m_{ji}(\overline{x},t)\right]$ are the moment antisymmetric and symmetric source tensors and δ_{ij} is Dirac´s δ. The traction boundary condition (normal component of stress) must satisfy

$$\sigma_{ij}(\overline{x},t)n_j(\overline{x}) = t_i(\overline{x},t)\ . \tag{3}$$

for $\overline{x}$ on S, where $t_i(\overline{x},t)$ are the components of the time-varying surface traction vector and $n_i(\overline{x})$ are the components of the outward unit normal to S. The initial conditions on the dependent variables are specified at V and on S at time t = t_0 by

$$v_i(\overline{x},t) = v^0_i(\overline{x}),\ \sigma_{ij}(\overline{x},t) = \sigma^0_{ij}(\overline{x}). \tag{4}$$

On output, the code produces both seismograms and 2D plane slices. If the orientation of interest is on a particular axis defined by the dimensionless unit vector $\overline{b}$, then the particle velocity seismogram is:

$$v_b(\overline{x}_r,t) = b_k v_k(\overline{x}_r,t) = b_1 v_1(\overline{x}_r,t) + b_2 v_2(\overline{x}_r,t) + b_3 v_3(\overline{x}_r,t). \tag{5}$$

Details about the staggered finite difference scheme on which the algorithm used is based can be found in [3].

3 Parallel Implementation

We use data parallelism for efficiency. The best parallel programs are those where each processor gets almost the same amount of work while trying to minimize communications. Using this kind of partition, the domain is decomposed into small pieces (subdomains) and distributed among all processors; therefore, each processor solves its own subdomain problems. 3D domain decomposition is shown in Fig. 2.

For the process discussed in this paper, 1D, 2D, and 3D decomposition are possible; however, we encourage the 3D one because it is well-balanced, extremely efficient and the more appropriate for the elastic wave propagation code as large problems –too big to fit on a single processor.

We use message passing interface (MPI) to parallelize 3DFD. The fourth order spatial finite difference scheme requires two additional planes of memory on every face of the subdomain to compute properly the finite difference solution independently from the other processes; therefore, we allocate padded subdomains of memory for every face of the subdomain cube (shown in the bottom of Fig. 2) to assure the precise functioning of the staggered finite difference scheme used.

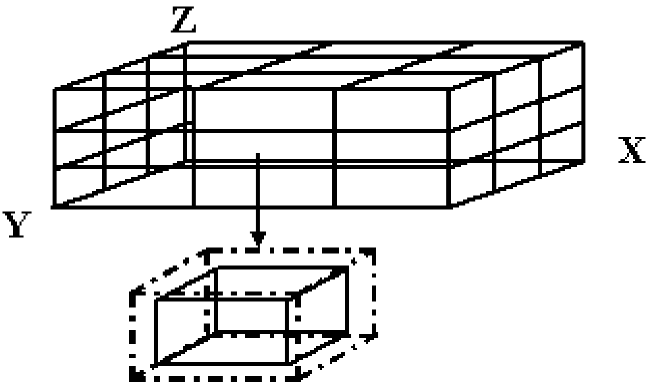

Fig. 2. 3D decomposition using data parallelism and an independent subdomain with ghost cells as dashed lines

Parallel I/O is used in the program that allows us to model a large realistic-size model. The input basic run parameters and geometry data which are scalars are read and broadcast by processor 0. The earth model data is read by all processors using collective I/O. The output part of the program uses, as well, collective I/O to write plane slices and seismograms. We do not measure the time spent in such phases because it is in the time-step loop where the majority of the time is spent. We use MPI *shift* commands to communicate neighboring's edges.

4 Efficiency

Speedup, Sp, and efficiency, E, among others, are the most important metrics to characterize the performance of parallel programs. Theoretically, speedup is limited by Amdahl's law [5]. For a scaled-size problem, one must estimate the running time on a single processor [2]. Sp and E are defined as

$$Sp \equiv \frac{mT_1(n/m)}{T_m(n)}, \qquad E \equiv \frac{T_1(n/p)}{mT_m(n)}. \tag{6}$$

where T_1 is the serial time execution and T_m is the parallel time execution on *m* processors for a size problem *n*.

We can estimate the cost for this parallel algorithm straightforward without I/O timings because the largeness of the work occurs in the elastic wave propagation. Therefore, we must estimate computation and communication terms

$$T(n,m)=\tau_{comp}(n,m)+\tau_{comm}(n,m) \quad . \tag{7}$$

where τ_{comp} is the computation cost and τ_{comm} represent the communication cost on *m* processors for a size problem *n*.

There are two main machine constants which most impact the speed of message communication that are *bandwidth,* β*,* -message dependant- which is (the reciprocal of) transmission time/byte, and latency, ι, represents the startup cost of sending an message -independent of message size. Therefore, the cost to send a single message with χ length of data is $\iota+\chi\beta$.

We use 128 processors of UNAM HP Cluster Platform 4000, which has Opteron dual core processors (1,368 cores) of 2.6GHz with Infiniband interconnection (known in short as KanBalam [6]). KanBalam has the following time constants: β= 1×10^{-9}, ι=13×10^{-6} , and the computation time per flop, Γ= 1.9×10^{-13}, all of them are in seconds. The size of each subdomain is $Nx\times Ny\times Nz$, that we call *R* for simplicity, where *Nx*=500, 1000, 2000, 4000; *Ny*=600, 1200, 2400, 4800 and *Nz*=124, 248, 496, 992 are model size per direction; therefore, the cost of performing a finite difference calculation on $npx\times npy\times npz$ *,m,* processors is $\mathrm{A}\Gamma R^3/m$, where A is the number of floating operations in the finite difference scheme (velocity-stress consists of nine coupled variables). As we stated above, this scheme requires us to communicate two neighboring's planes in the 3D decomposition plus four extra planes for cubic extrapolation –necessary if the user specifies a receiver or slice plane not on a grid node; therefore, communication costs for a 1D decomposition are –at most- $8(\iota+4\beta R^2)$, where the factor 4 is the size in bytes of memory of each data grid. $16(\iota+4\beta R^2/\sqrt{m})$ is the cost for a 2D decomposition, and for a 3D decomposition we have $24(\iota+4\beta R^2/m^{2/3})$. In short, for a 3D decomposition we have the following

$$T(n,m)=\mathrm{A}\Gamma R^3/m+24\left(\iota+4\beta R^2/m^{2/3}\right). \tag{8}$$

and

$$Sp\equiv\frac{\mathrm{A}\Gamma R^3}{\mathrm{A}\Gamma R^3/m+24\left(\iota+4\beta R^2/m^{2/3}\right)} \; . \tag{9}$$

The communication cost depends on both the order the finite difference scheme and the type of processor decomposition.

Results for different size models and number of processors (P) from 1-1,024 are shown in Table 1 and Fig. 3. I/O timings are not reported.

Table 1. Scaled-sized model: processors used in each axis, timings, speedup, efficiency and memory per subdomain (mps) obtained (The 1,024 processors results are based on (8) and (9))

Size model and spatial step (dh, km)	P	Px	Py	Pz	Total run time (s)	Speedup (Sp)	Efficiency (E)	Mps (GB)
500x600x124 (1)	1	1	1	1	34187.7	1	1	2.08
1000x1200x248 (0.5)	16	1	4	4	33201.5	16.47	1.03	1.042
2000x2400x496 (0.25)	128	4	8	4	36230.3	120.8	0.94	1.042
4000x4800x992 (0.125)	*1024*	*16*	*16*	*4*	*39986.3*	*876*	*0.85*	*1.042*

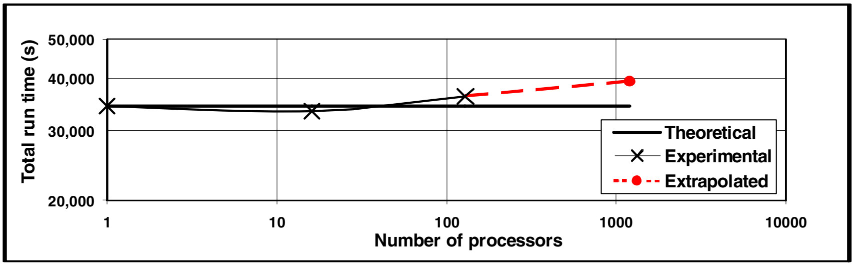

Fig. 3. Running time for the models run on KanBalam. (The 1,024 processors results are based on (8) and (9)).

5 Results for the 19/09/1985 Mexico's Ms 8.1 Subduction Earthquake

Herewith we present two examples of the type of results that were obtained with the 3D parallel MPI code implemented: the low frequency velocity field patterns in the X direction, Fig. 2 and the seismograms obtained at observational points in the so-called near and far fields of the wave propagation pattern.

Three spatial discretizations of the earth crust volume were used: dh = 1, 0.5, and 0.25 km. In Fig. 4A, we present two snapshots of the wave propagation patterns in the X direction obtained 48 and 120s after the initiation of the kinematic rupture of the seismic source, They correspond to the dh = 0.5 km discretization, notice in Fig. 4A that at 48s the main seismic effects are occurring in the near field, i.e. on top of the seismic source, while the opposite is observed at 120s, when the seismic waves are fully developed in the far field, where Mexico City is located with respect to the source.

In Fig. 4B the synthetic seismograms obtained for dh = 1, 0.5 and 0.25 km, at an observation site practically on top of the largest “subevent” of the 1985 Mexico earthquake (Fig. 1C) are presented, Notice in Fig. 4B, that, the maximum amplitude of the seismograms are very similar; however, the effect of the “numerical noise” of the seismogram associated to the coarser discretization of 1 km of the seismic source is drastically reduced for the corresponding to the 0.25 km one. This effect is clearly shown in the Fourier Amplitude spectra of the seismograms, which are shown on the

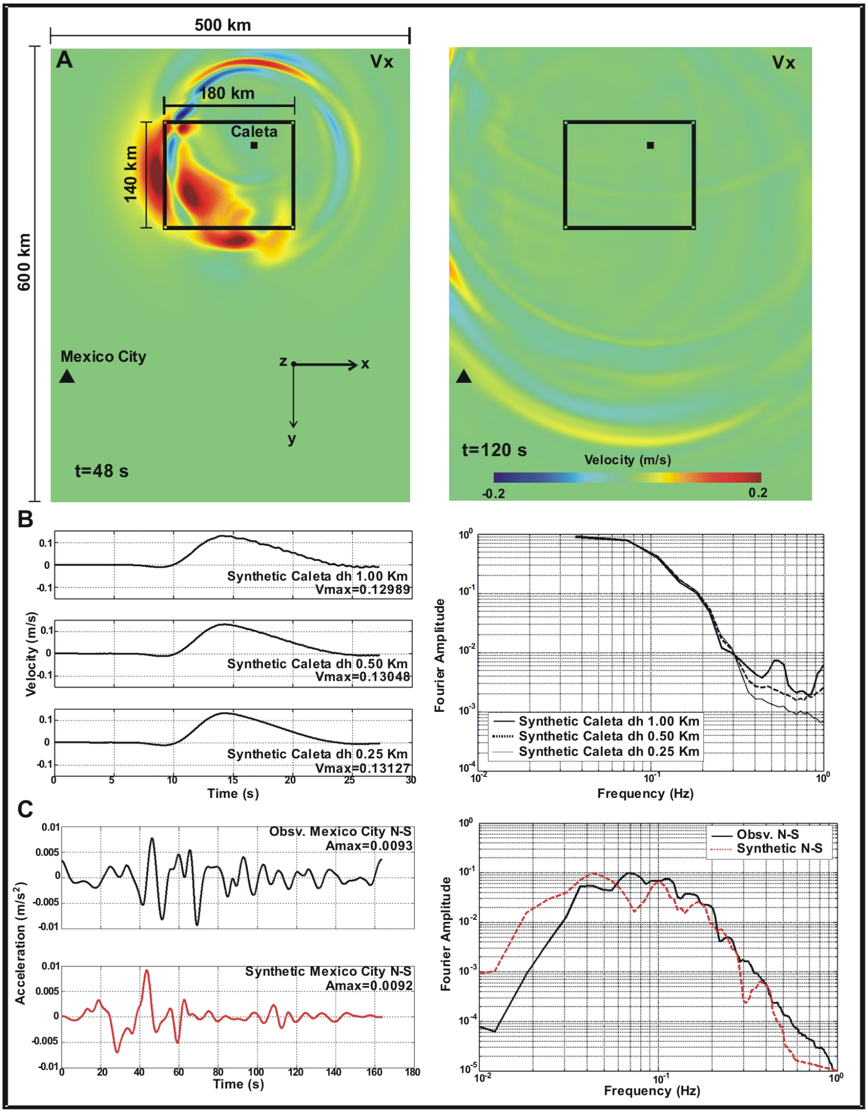

Fig. 4. A) Snapshot of the velocity wavefield in the X direction of propagation for f<= 0.2 Hz in the surface of the domain of interest; B) Left side seismogram, right side Fourier amplitude spectra obtained for the dh=1, 0.5 and 0.25 km discretizations; C) Left side observed and synthetic accelerograms north-south direction, right side Fourier amplitude spectra for the Mexico's Ms 8.1 earthquake

right side of Fig. 4B. In the latter, the mentioned "numerical noise" of the dh = 1 km discretization is associated to the bump between 0.5 and 0.6 Hertz of its Fourier amplitude spectra, versus the "no bump" at the same frequencies of the dh = 0.25 km discretization.

Finally, in Fig 4C we show the observed and synthetic (for a spatial discretization dh = 0.5km) low frequency, North-south accelerograms of the 19/09/1985 Ms 8.1 Mexico earthquake, and their corresponding Fourier Amplitude spectra for the firm soil Tacubaya site in Mexico City, i.e. at a far field observational site. Notice in these figures that the agreement between the observed and the synthetic accelerograms is reasonable both in the time and in the frequency domain.

6 Conclusions

We decomposed a realistic-size domain in 1D, 2D and 3D using data parallelism. Each processor allocates memory for its own subdomain and two plane faces of padding for each face in order to compute independently the finite difference calculation. We improve I/O using collective communications, but they are no reported in getting the performance of the implementation. The efficiency achieved is of 94% for 128 and of 85% extrapolating to 1,024 processors of the HP Cluster Platform 4000 Opteron dual core supercomputer of UNAM. The low frequency synthetic seismograms obtained with the parallel code implemented, particularly the ones for a spatial discretization of 0.5 and 0.25km show a good fit, both in the time and in the frequency domain with the observations of the Mexico's 19/09/1985 Ms 8.1 subduction earthquake.

Acknowledgments

We acknowledge DGSCA, UNAM for the support we received to use the HP Cluster Platform 4000 Opteron dual core supercomputer (KanBalam). We would also like to thank the staff of the Supercomputing Department at DGSCA, UNAM, particularly to José Luis Gordillo, Eduardo Murrieta and Adrián Durán. Thanks to Martha Mora for her grammatical suggestions.

References

[1] Chavez, M., Olsen, K., Cabrera, E.: Broadband Modeling of Strong Ground Motions for Prediction Purposes for Subduction Earthquakes Occurring in the Colima-Jalisco Region of Mexico. 13WCEE, Vancouver, B.C., Canada, August 1-6, 2004, Paper No 1653 (2004)

[2] Minkoff, S.E.: Spatial Parallelism of a 3D Finite Difference Velocity-Stress Elastic Wave Propagation code. SIAM J. Sci. Comput. 24(1), 1–19 (2002)

[3] Madariaga, R.: Dynamics of an Expanding Circular Fault. Bull. Seismol. Soc. Amer. 66, 639–666 (1976)

[4] Mendoza, C., Hartzell, S.: Slip Distribution of the 19 September 1985 Michoacan, Mexico, Earthquake: near Source and Telesismic constrains. Bull. Seismol. Soc. Amer. 79, 655–699 (1989)

[5] Amdahl, G.: Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities. In: Conference Proceedings, AFIPS, pp. 483-485 (1967)

[6] Supercomputing Department, DGSCA, UNAM, Supercomputer KanBalam http://www.super.unam.mx/index.php?op=eqhw