Frequency-domain acoustic wave modeling using a hybrid direct-iterative solver based on a domain decomposition method: a tool for 3D full-waveform inversion?

F. Sourbier, A. Haidar, L. Giraud, S. Operto and J. Virieux

Gene around the world at CERFACS. February 29, 2008.

Introduction

Schur complement method

Complexity

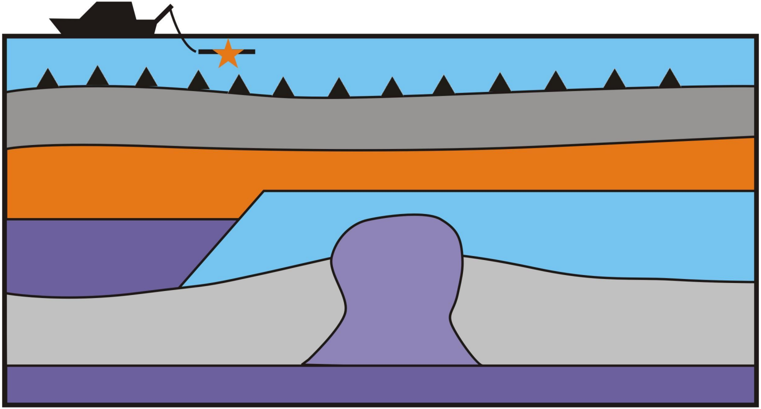
3D case

Accuracy of the solution

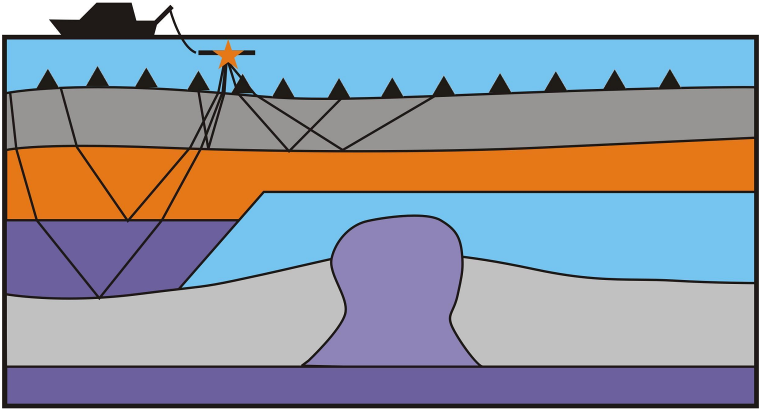
Conclusions

- Consortium Seiscope 2006-2008 : Global offset seismic imaging, http ://seiscope.unice.fr/
- Permanent researchers (S. Operto, A. Ribodetti, J. Virieux), Phd students (H. Ben Hadj Ali, R. Brossier, V. Etienne), Research engineer (F. Sourbier)
- **Sponsors** : BP, CGG-VERITAS, EXXON-MOBIL, SHELL, TOTAL
- **Collaborators** : CERFACS (A. Haidar), ENSHEEIT-IRIT (P. Amestoy, L. Giraud), ENS LYON (E. Agullo, J.Y. L'Excellent), INRIA-CAIMAN (S. Lantéri, N. Glinsky, M. Ben Jemaa)

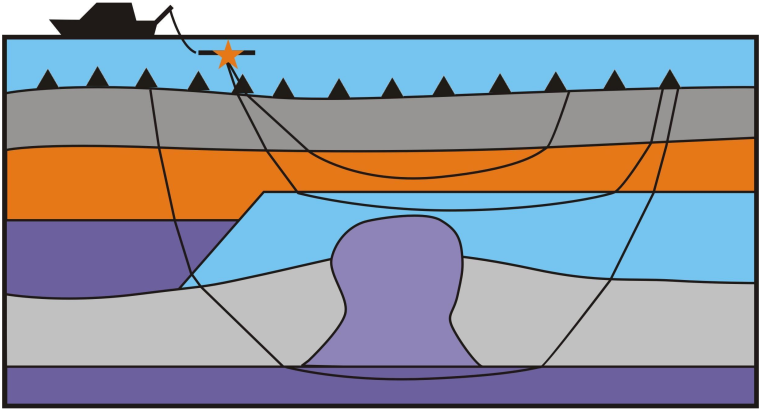
Global offset seismic acquisition



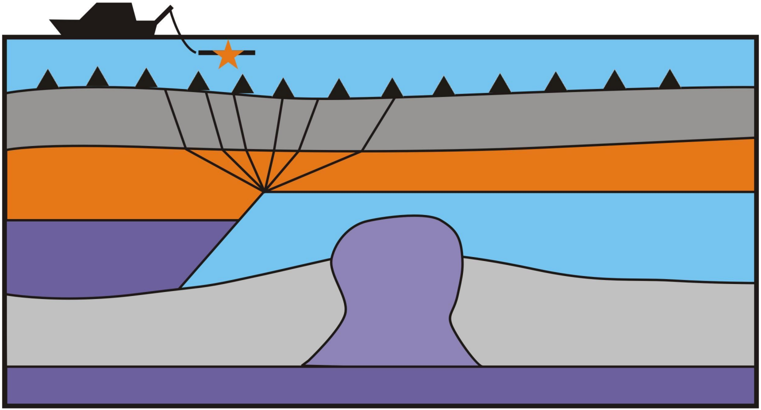
Reflections



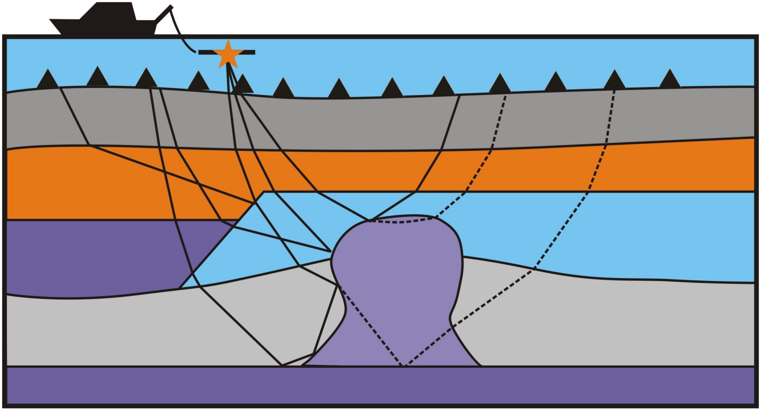
Refraction



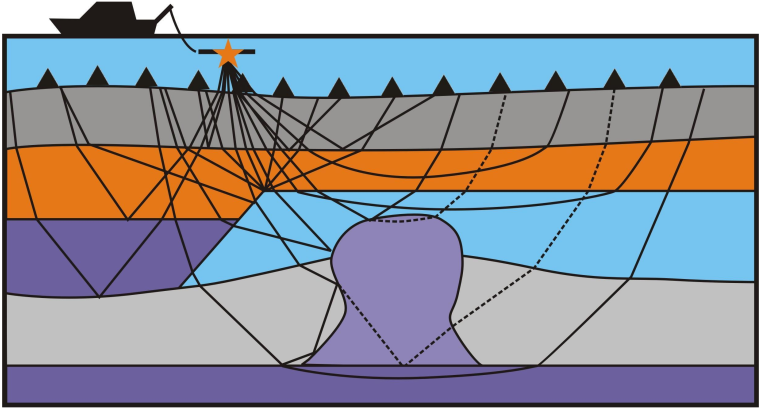
Diffraction



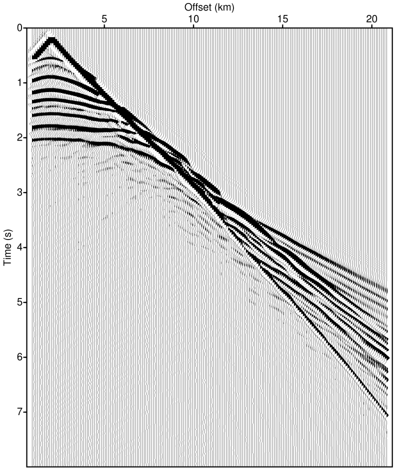
Reflection from steep dips, multiple reflections and transmission accoss strong velocity contrats

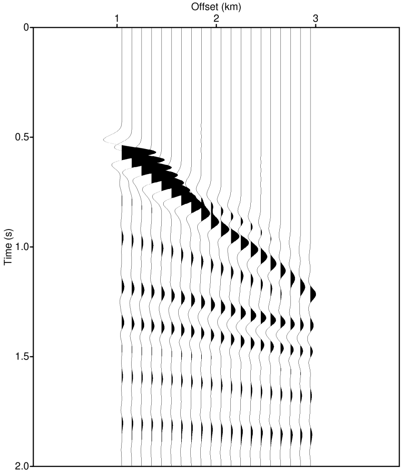


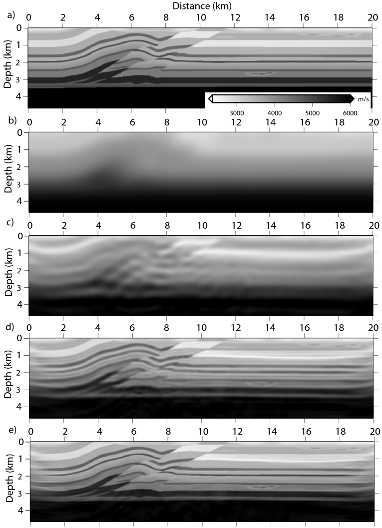
Need the complete solution of the wave equation



Results : time seismograms




Offset (km)
1
2
3
Time (s)
0
0.5
1.0
1.5
2.0


Distance (km)
a)
b)
c)
d)
e)
Depth (km)
m/s
3000
4000
5000
6000

Frequency-domain wave propagation modeling, ie. resolution of a large and sparse system of linear equations.

Goal

- Solve very large sparse linear system with MRHS and several millions of unknows (3D case) with an efficient method in terms of CPU times and memory.
- Design an efficient tool for large 3D frequency-domain full-waveform inversion problems.

Frequency-domain wave propagation modeling, ie. resolution of a large and sparse system of linear equations.

Goal

- Solve very large sparse linear system with MRHS and several millions of unknows (3D case) with an efficient method in terms of CPU times and memory.
- Design an efficient tool for large 3D frequency-domain full-waveform inversion problems.

Problem

- Overcome the huge memory requirement in direct solver.
- Reduce the iteration count in iterative solver.

Frequency-domain wave propagation modeling, ie. resolution of a large and sparse system of linear equations.

Goal

- Solve very large sparse linear system with MRHS and several millions of unknows (3D case) with an efficient method in terms of CPU times and memory.
- Design an efficient tool for large 3D frequency-domain full-waveform inversion problems.

Problem

- Overcome the huge memory requirement in direct solver.
- Reduce the iteration count in iterative solver.

Method

- Mixing direct solver (LU factorization) and iterative solver (GMRES) by a Schur complement approach.

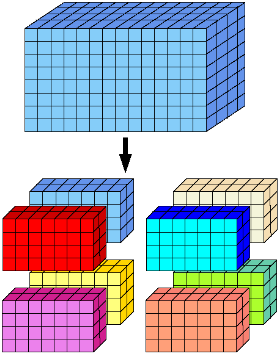
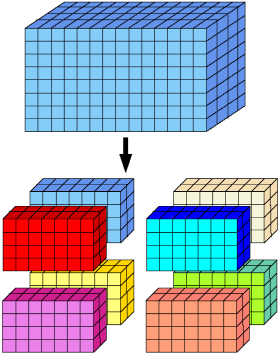


Fig.: Splitting of the computational domain into n sub-domains.



- Direct solver is applied on each sub-domain (interior nodes).

Fig.: Splitting of the computational domain into n sub-domains.

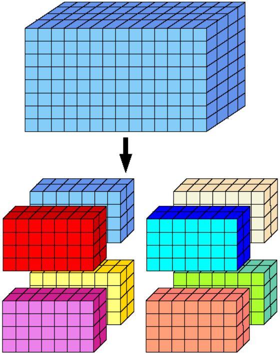


Fig.: Splitting of the computational domain into n sub-domains.

- Direct solver is applied on each sub-domain (interior nodes).
- Iterative solver is used to solve the interface nodes between adjacent domains.

The visco-acoustic wave equation (**Helmholtz equation**) is written in the frequency domain as :

$$\frac{\omega^2}{\kappa(\mathbf{x})} p(\mathbf{x},\omega) + \nabla \left(\frac{1}{\rho(\mathbf{x})} \nabla p(\mathbf{x},\omega) \right) = -s(\mathbf{x},\omega) \tag{1}$$

where $\rho(\mathbf{x})$ is density, $\kappa(\mathbf{x})$ is the bulk modulus, ω is angular frequency ; $p(\mathbf{x},\omega)$ and $s(\mathbf{x},\omega)$ denote the pressure and source respectively.
Equation 1 can be recast in matrix form as

$$\mathbf{AP} = \mathbf{s} \tag{2}$$

After reordering the interior nodes by sub-domain and labeling the interface nodes last, the system $\mathbf{A}\mathbf{p} = \mathbf{s}$ becomes

$$\begin{bmatrix} A_{ii}^1 & & & & A_{ib}^1 \\ & A_{ii}^2 & & & A_{ib}^2 \\ & & \ddots & & \vdots \\ & & & A_{ii}^n & A_{ib}^n \\ A_{bi}^1 & A_{bi}^2 & \cdots & A_{bi}^n & \bar{A}_{bb} \end{bmatrix} \begin{bmatrix} p_i^1 \\ p_i^2 \\ \vdots \\ p_i^n \\ \bar{p}_b \end{bmatrix} = \begin{bmatrix} s_i^1 \\ s_i^2 \\ \vdots \\ s_i^n \\ \bar{s}_b \end{bmatrix} \tag{3}$$

where p_i^j denote unknowns located at interior nodes of sub-domain j and $\bar{p}_b$ denote unknowns located at all interface nodes. Note that indices b and i label interface and interior nodes respectively while the exponent labels sub-domains. The number of sub-domains is denoted n.

The system (3) can be written in compact form as

$$\begin{bmatrix} A_{ii} & A_{ib} \\ A_{bi} & \bar{A}_{bb} \end{bmatrix} \begin{bmatrix} p_i \\ \bar{p}_b \end{bmatrix} = \begin{bmatrix} s_i \\ s_b \end{bmatrix} \tag{4}$$

Eliminating p_i from the second block row of eq. (4) leads to the following reduced system for $\bar{p}_b$

$$(\bar{A}_{bb} - A_{bi}A_{ii}^{-1}A_{ib})\bar{p}_b = s_b - A_{bi}A_{ii}^{-1}s_i \tag{5}$$

The matrix $S = \bar{A}_{bb} - A_{bi}A_{ii}^{-1}A_{ib}$ is the **dense Schur complement matrix**. A Krylov method is used to solve this sytem of linear equation.

- Preconditioning ie. transforming the original linear system into one which is likely to be easier to solve with an iterative method.
- $S\bar{p}_b = f \Rightarrow SM\bar{p}_b = fM$
- M_{as} : classical additive Schwarz preconditioner (exactly the inverse of S for 2 sub-domains).

Advantages

- Local problem (sub-domain) assigning to one group of processors.
- Naturally parallel.
- S is computed at an affordable memory and computational cost with MUMPS.
- Same formulation in 2D/3D, acoustic, elastic, FD, FV ...

Drawbacks

- S is a **dense** matrix of order the number of interface grid point.
- Need an efficient **preconditioner** to accelerate the convergence of the Krylov method.

- Direct solver : Memory for factorization
- Hybrid solver : Memory for **local** factorization + **dense local** Schur matrix + preconditioner

D	Direct Solver	Hybrid Solver
2	$ov15N^2Log_2N$	$15N^2Log_2N/k + 2(2N/k)^2[4 + (k-2)(4k+1)]$
3	$ov35N^4$	$107N^4/k$

Tab.: Memory complexity of the direct and hybrid solvers. N denotes the dimension size of either a 2D N^2 and 3D N^3 grid. k is the number of sub-domain along each direction ($n = k^2$ and $n = k^3$ in 2D and 3D respectively). ov is the memory overhead coefficient (~ 2).

- Direct solver : Cpu time for factorization and resolution phases
- Hybrid solver : Cpu time for local factorization & Schur computed + setup preconditioner + iterative solver + local resolution phases

- Cubic homogeneous model (0.5 to 8 $\times 10^6$ dof)
- Section of the Overthrust model (Channel) (1.6 $\times 10^6$ dof)
- Overthrust model at frequency 7 Hz (5.6 $\times 10^6$ dof)
- 75 % of the Overthrust model at frequency 10 Hz (9.9 $\times 10^6$ dof)

dof : degrees of freedom ie. number of unknows.

- Solved problem size vary from 0.5 to 8 millions of unknowns. $N = 80, 100, 160, 200$.
- Frequency is 15 Hz, mesh spacing is 50 m and velocity equals to 3000 m/s. This lead to 4 grid point per P-wavelength.

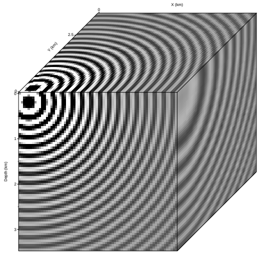


Fig.: 15-Hz monochromatic wavefield computed with the hybrid approach.

N^3	80^3	100^3	160^3	200^3
k^3	2^3	4^3	5^3	5^3
$(N/k)^3$	40^3	25^3	33^3	41^3
schur (Mb)	181-194	27-112	71-302	$\sim$ 800
facto (Mb)	1214	188	496	1214
Memproc (Mb)	1576-1602	242-412	638-1100	$\sim$ 2814
Mem (Gb)	12-13	15-26	80-137	$\sim$ 352
MemTheo (Gb)	17	21	131	$\sim$ 273
cpu_facto (s)	152	12	43	153
cpu_pcond (s)	75	21	100	-
cpu_rhs (s)	31	120	553	-
cpu_total (s)	249	163	759	-

Tab.: Cubic homogeneous model. *Memproc* : average space per working process ; *Mem* : total memory actually used. *MemTheo* : total theoritical memory complexity ($107N^4/k$).

- Small target of the model ($2.3 \times 8.8 \times 7.0\ km^3$).
- Solved problem size : 1.6 millions of unknowns.
- Grid $56 \times 186 \times 151$ with PML layer.
- Frequency is 5 Hz, mesh spacing is 50 m, 14 grid point per P-wavelength.

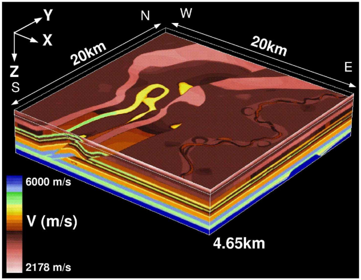


Fig.: 3D SEG/EAGE Overthrust model.

process	20	32	40	60	168
facto (Mb)	1141	767	592	359	84
schur (Mb)	339	224	250	111	36
Memproc (Mb)	1818	1215	1092	582	155
Mem (Gb)	36	39	44	35	26
iter Gmres	23	36	38	102	151
cpu_facto (s)	142	77	53	26	4
cpu_pcond (s)	-	134	84	46	7
cpu_rhs (s)	-	205	118	110	38
cpu_tot (s)	-	416	255	182	49

Tab.: *Memproc* : average space per working process ; *Mem* : total memory actually used.

process	20	32	40	60
facto (Mb)	1141	767	592	359
schur (Mb)	339	224	250	111
Memproc (Mb)	1818	1215	1092 *(1410)*	582 *(972)*
Mem (Gb)	36	39	44 *(56)*	35 *(58)*
iter Gmres	23	36	38	102
cpu_facto (s)	142	77	53 *(814)*	26 *(569)*
cpu_pcond (s)	-	134	84	46
cpu_rhs (s)	-	205	118 *(100)*	110 *(2)*
cpu_tot (s)	-	416	255 *(914)*	182 *(571)*

Tab.: *Memproc* : average space per working process ; *Mem* : total memory actually used. In parentheses are results for direct solver.

- Solved problem size : 5.6 millions of unknowns
- Frequency is 7 Hz, mesh spacing is 75 m and minimun velocity equals to 2179 m/s. This lead to 4 grid point per P-wavelength.
- Grid $73 \times 277 \times 277$ with PML layer.
- *Mas* failed, *Mascs* ok

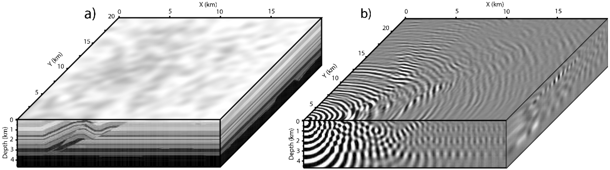


Fig.: a) 3D SEG/EAGE overthrust model. b) 7-Hz monochromatic wavefield computed with the hybrid approach.

solver	DDM	DDM	FWM (75%)
process	128	192	192
Memproc (Mb)	1194	716	1603
Mem (Gb)	153	137	308
iter Gmres	134	235	-
cpu_facto (s)	69	28	6425
cpu_pcond (s)	125	69	-
cpu_rhs (s)	410	391	9
cpu_total (s)	651	525	6434

Tab.: *Memproc* : average space per working process ; *Mem* : total memory actually used.

Theoritical total memory for the direct solver FWM3D : 860 Gb $(2 \times 35 \times 73 \times 277^3 \times 8)$

- Solved problem size : 9.9 millions of unknowns
- Only 75% of the model.
- Frequency is 10 Hz, mesh spacing is 50 m. This lead to 4 grid point per P-wavelength.
- Grid 104 $\times$ 310 $\times$ 310 with PML layer.
- *Mas* failed, *Mascs* ok

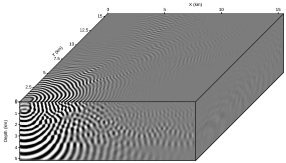


Fig.: 10-Hz monochromatic wavefield computed with the hybrid approach in 75% of the 3D SEG/EAGE overthrust model.

process	192
splitting	$3 \times 8 \times 8$
sub-domain	$35 \times 39 \times 39$
sizeIntrf_tot	629882
sizeIntrf	4139-8058
schur (Mb)	137-519
facto (Mb)	834
iter Gmres	282
cpu_facto (s)	89
cpu_pcond (s)	444
cpu_rhs (s)	1686 (swap ?)
cpu_total (s)	2956

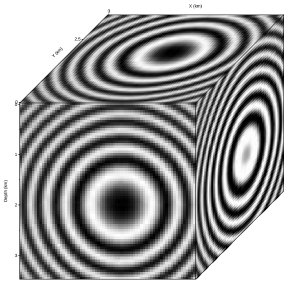
Theoritical total memory for the direct solver FWM3D : 1.7 Tb
$(2 \times 35 \times 104 \times 310^3 \times 8)$

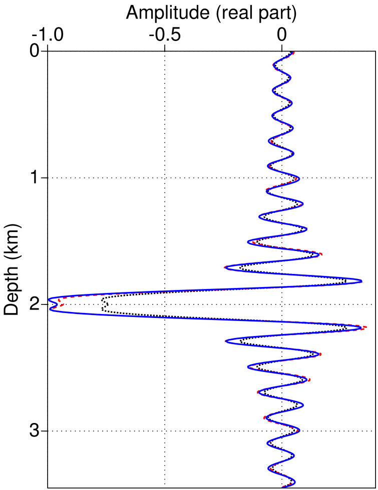
Which GMRES's convergence tolerance for a 'good accuracy' ?

- Estimated the L^2 norm of residuals errors ie. the distance between the frequency wavefields computed with the direct and hybrid solver. Set 10^{-3} seems to be a good compromise between accuracy and iteration count (cf. Technical Report N°006).
- Comparison with analytical solution (3D).
- Compute and compare numerical seismograms (2D).

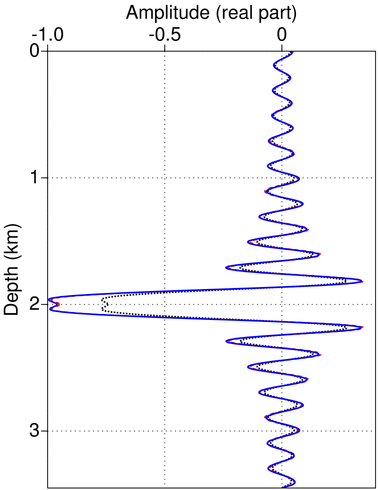
3D analytical Green function in a homogeneous media :

$$GF(\mathbf{r}) = \frac{1}{4\pi}\frac{\rho}{\mathbf{r}}e^{\frac{i\omega\mathbf{r}}{c}}$$

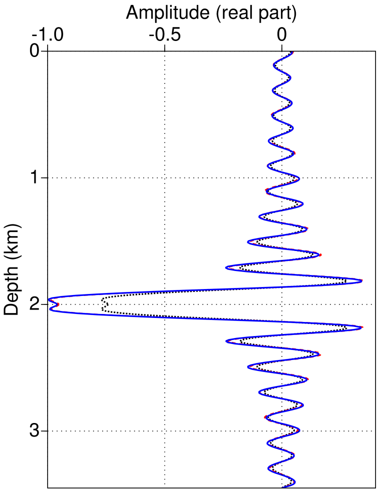




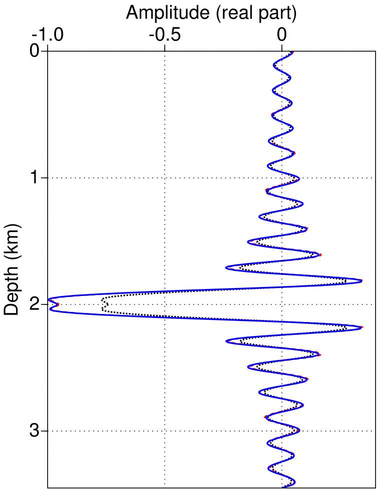
- Analytical : black line.
- FWM : blue line.
- DDM : red line
- $\epsilon = 10^{-1}$



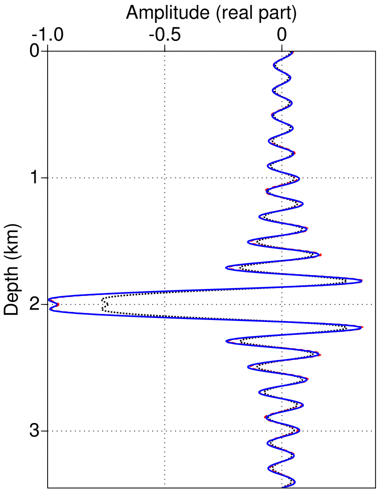
- Analytical : black line.
- FWM : blue line.
- DDM : red line
- $\epsilon = 10^{-2}$



- Analytical : black line.
- FWM : blue line.
- DDM : red line
- $\epsilon = 10^{-3}$

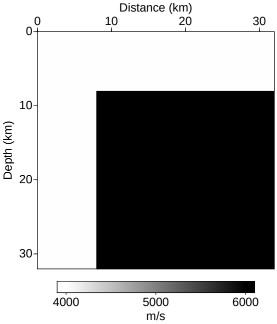


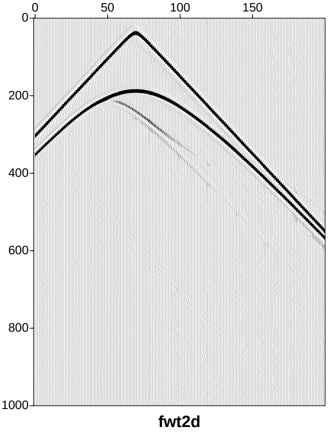
- Analytical : black line.
- FWM : blue line.
- DDM : red line
- $\epsilon = 10^{-4}$

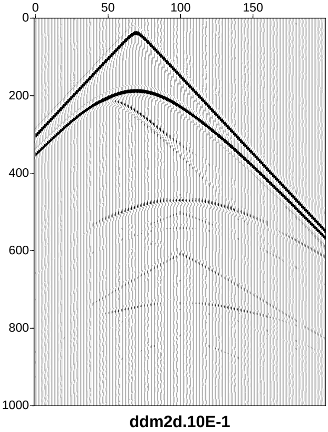


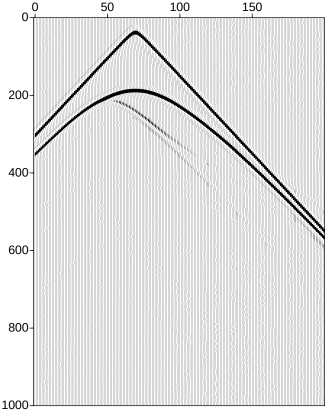
- Analytical : black line.
- FWM : blue line.
- DDM : red line
- $\epsilon = 10^{-5}$

- 2D corner-edge model (cf. Documentation of FWT2D).
- Velocity grid of 841 $\times$ 841 including PML layers with a 40 m grid interval.
- One source located at 11 km and 5 km depth and 200 receivers equally-spaced (160 m) at 4.5 km depth.

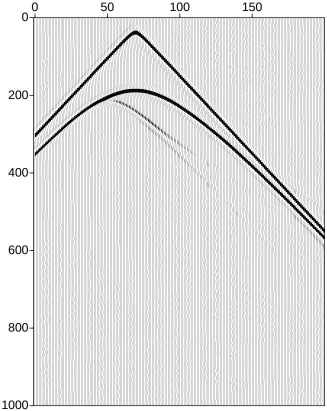



0
50
100
150
0
200
400
600
800
1000
fwt2d

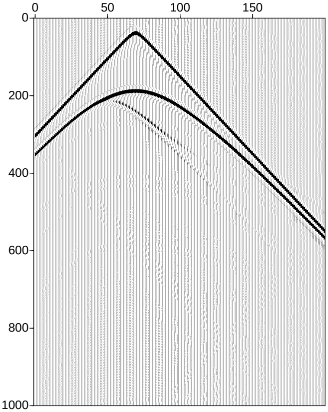

0
50
100
150
0
200
400
600
800
1000
ddm2d.10E-1



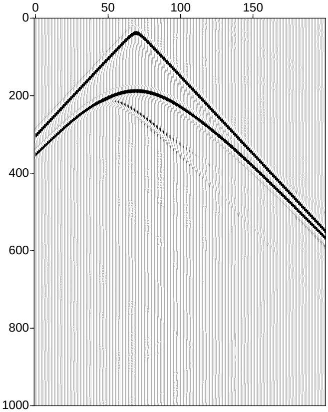
fwt2d



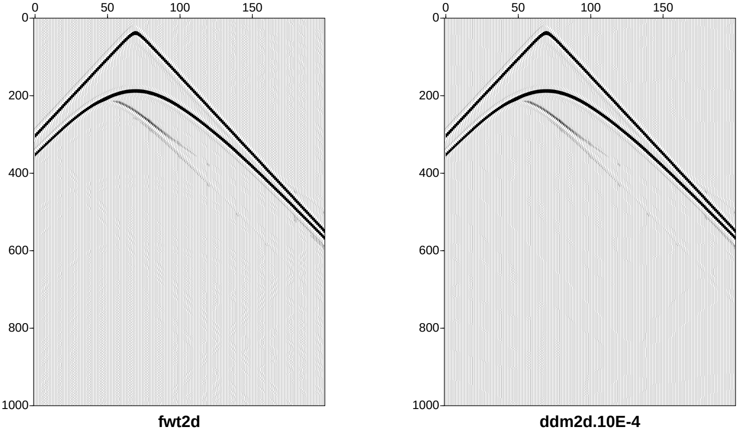
ddm2d.10E-2

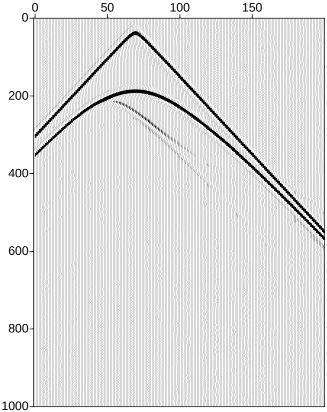


fwt2d

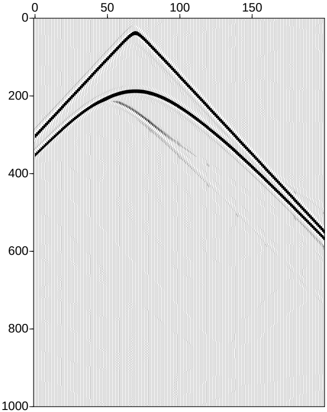


ddm2d.10E-3


0
50
100
150
0
200
400
600
800
1000
fwt2d
0
50
100
150
0
200
400
600
800
1000
ddm2d.10E-4



fwt2d



ddm2d.10E-5

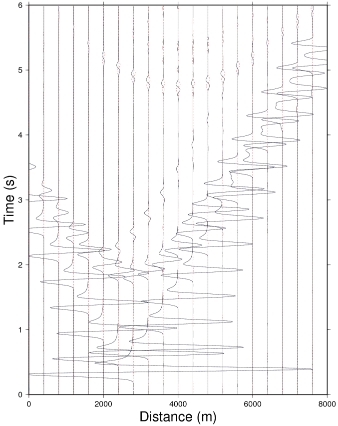


Fig.: FWM : blue line. DDM : black line. Residuals : red line. $\epsilon = 10^{-1}$.

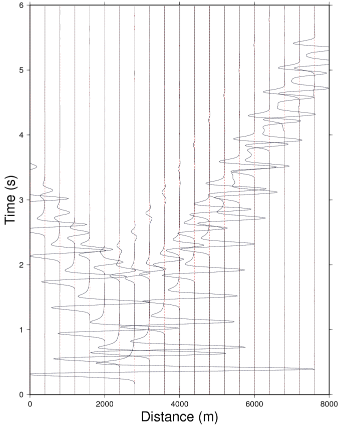


Fig.: FWM : blue line. DDM : black line. Residuals : red line. $\epsilon = 10^{-2}$.

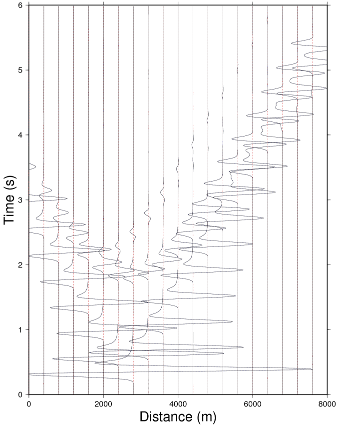


Fig.: FWM : blue line. DDM : black line. Residuals : red line. $\epsilon = 10^{-3}$.

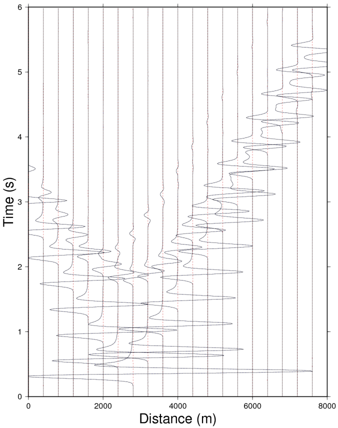


Fig.: FWM : blue line. DDM : black line. Residuals : red line. $\epsilon = 10^{-4}$.

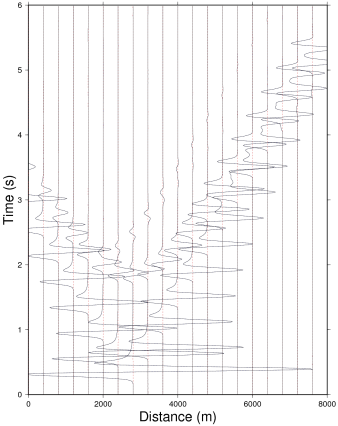
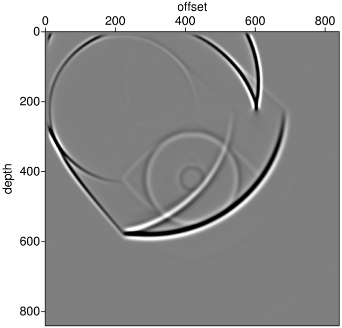


Fig.: FWM : blue line. DDM : black line. Residuals : red line. $\epsilon = 10^{-5}$.



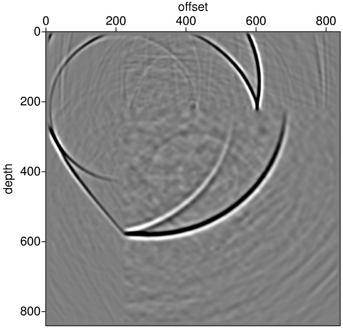


Fig.: Snapshot computed in the corner edge model with the DDM2D code with 2x2 and 4x4 domains. $\epsilon = 10^{-1}$

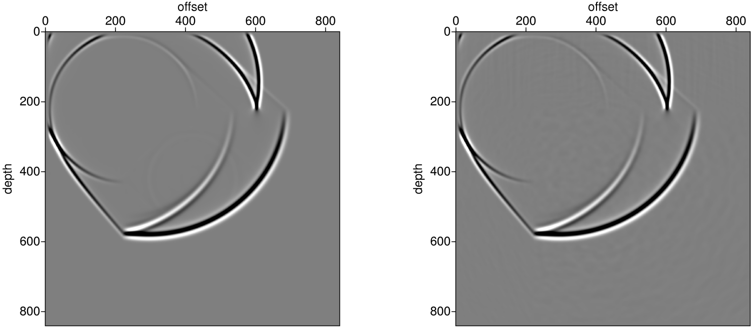


Fig.: Snapshot computed in the corner edge model with the DDM2D code with 2x2 and 4x4 domains. $\epsilon = 10^{-2}$

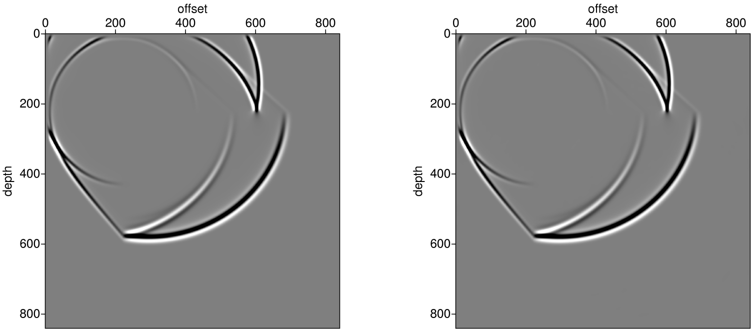


Fig.: Snapshot computed in the corner edge model with the DDM2D code with 2x2 and 4x4 domains. $\epsilon = 10^{-3}$

3D Overthrust model at 7 Hz with 128 process.

ϵ	10^{-1}	10^{-2}	10^{-3}
iter	17	72	134
cpu_rhs (s)	52	220	410

Tab.: Extrapolation of the elapsed time for one source for the hybrid solver as a function of the Gmres's tolerance.

3D Overthrust model at 7 Hz with 128 process.

ϵ	10^{-1}	10^{-2}	10^{-3}
iter	17	72	134
cpu_rhs (s)	52	220	410

Tab.: Extrapolation of the elapsed time for one source for the hybrid solver as a function of the Gmres's tolerance.

- The cpu time for one source is strongly controlled by the Gmres's tolerance, ϵ.

- The memory cost of the hybrid solver is dramatically smaller that of the direct solver especially for 3D problems.
- The time requirement of the hybrid solver is also significantly smaller that of the direct solver for **single-source problem.**

- Optimization of the iterative solver in the hybrid approach for muti-source problems will be the key point to design an efficient modeling tool for large 3D frequency-domain full-waveform inversion problems.
- Implementation of 2 levels of parallelism in the hybrid approach (each sub-domain is processed on few processors rather than on one) would contribute to still improve the time performance of the hybrid solver.

Section of the Overthrust model (Channel)

-	1-level	2-levels	direct
process	60	60	60
sub-dom	60	15	-
facto (Mb)	359	659	-
schur (Mb)	111	425	-
Memproc (Mb)	582	1508	972
Mem (Gb)	35	23	58
iter Gmres	102	19	-
cpu_facto (s)	26	97	-
cpu_pcond (s)	46	97	-
cpu_Gmres (s)	61	45	-
cpu_rhs (s)	110	58	2
cpu_tot (s)	182	253	571

This study was funded by the SEISCOPE ANR-05-NT05-2_42427 and the consortium SEISCOPE (http ://seiscope.unice.fr) sponsored by BP, CGG-VERITAS, EXXON-MOBIL, SHELL and TOTAL. The simualtions presented were computed on the Pc Cluster of Mesocentre SIGAMM. We thanks P. Amestoy (ENSHEEIT-IRIT) and J.-Y. L'Excellent (INRIA-ENS LYON) for providing us the MUMPS solver. We thank the CERFACS for providing us the GMRES solver.