Anisotropic elastic waves or how to model the subsurface more realistically

GdT INRIA Magique3D, 24 November 2014

Depth Imaging Partnership

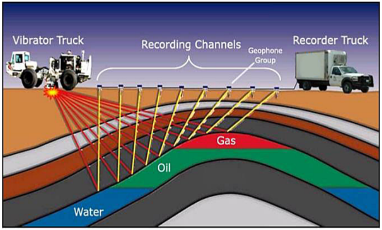
- Hélène BARUCQ, Julien DIAZ – INRIA Magique3D, Pau
- Henri CALANDRA – TOTAL E&P, Houston
- Emmanuel AGULLO – INRIA Hiepacs, Bordeaux
- George BOSILCA – ICL, University of Tennessee

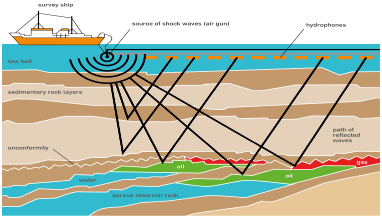
Outline

Geophysics context

Upstream:

- Searching for underground oil and gas (Seismic Imaging)
- Drilling and operating the wells (Reservoir simulation)





- Midstream: transportation, storage and wholesale marketing
- Downstream: refining and purifying, marketing and distribution

Main SI techniques

Kirchhoff / Gaussian Beam

- old techniques (pre-study, approximated)
- depth migration algorithms
- simplified: vertical propagation

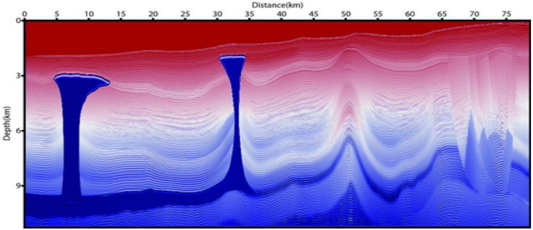
Reverse Time Migration (RTM)

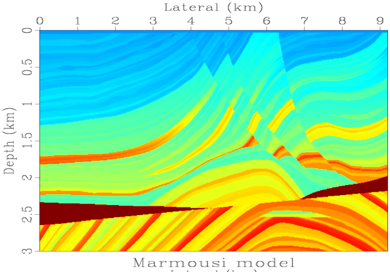
- current widely used technique
- depth migration algorithm
- accurate: full wave equation

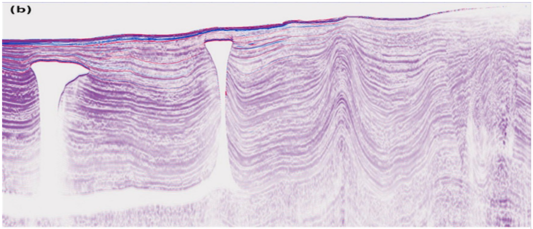
Full Waveform Inversion (FWI)

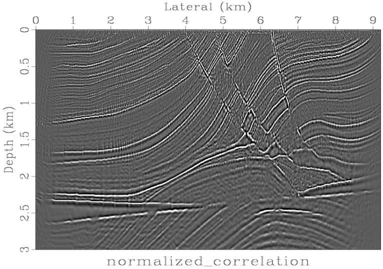
- upcoming technique
- Inverse problem -based
- complete: nature of the layers

RTM result

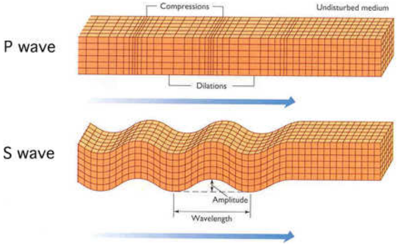

Distance(km)
0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75
Depth(km)
0 3 6 9

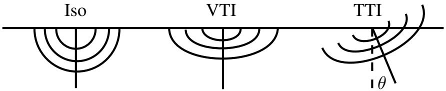

Lateral (km)
0 1 2 3 4 5 6 7 8 9
Depth (km)
0 0.5 1 1.5 2 2.5 3
Marmousi model


(b)


Lateral (km)
0 1 2 3 4 5 6 7 8 9
Depth (km)
0 0.5 1 1.5 2 2.5 3
normalized_correlation

From the simplest to the most realistic





type	(pseudo-)acoustic	elastic	poor-elastic
waves	P	$P, S_1, S_2(3D)$	...
isotropy	ρ, c	ρ, V_p, V_s	$\rho, V_p, V_s, \kappa, \mu$
VTI	ε	$\varepsilon, \delta, \gamma(3D)$	...
TTI	$\theta, \phi(3D)$	...	...

Equations and implementations

Space discretization

- Finite Diff., Discontinuous Galerkin $\Rightarrow$ (multi-, block-)diagonal matrix
- Finite Element, Finite Volume $\Rightarrow$ random sparse matrix

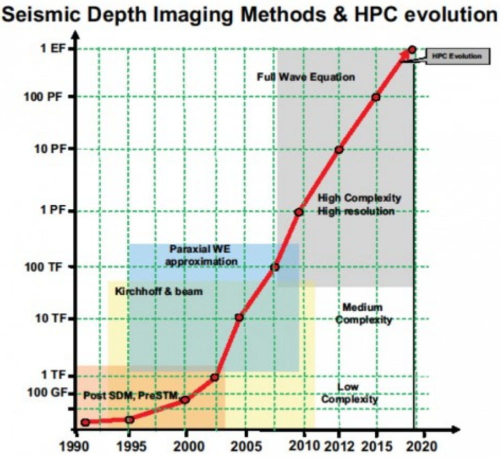
Time-dependent or harmonic

- Time schemes: Leap-Frog, Runge-Kutta, ...
- Freq. solvers: MUMPS, multigrid...

- 1st or 2nd order (auxiliary variable)
- source or initial condition (e.g. Ricker)

- Hou10ni: DG, time & freq., Gar6more: analytic (Julien)
- Montjoie: FEM, time & freq. (Marc)
- DIVA-DG (TOTAL)
  - TMBM (Simon, Lionel): DG, time
  - THBM (Marie, Théo., Flo.): DG, freq.

HPC context



From one-way to full-wave

- more computations
- more storage

From isotropy to TTI

- more computations
- more storage

Miscellaneous

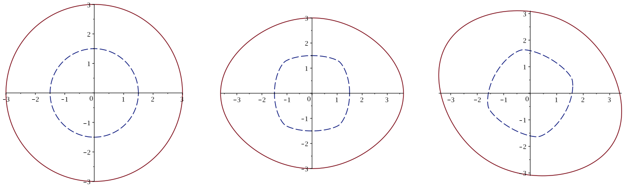
- big data issues (imaging)
- uncertainty quantification

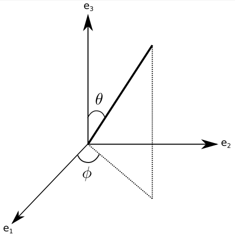
Outline

What TTI is?

Geophysics anisotropy

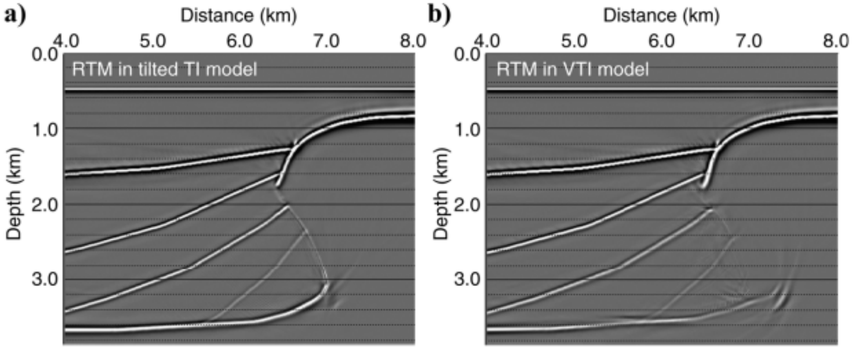
Earth's crust (geological layers of rocks) is assumed to be locally polar anisotropic, also called transversely isotropic (TI).





type	isotropy	Vertical TI	Tilted TI
tensor	sparse ref.	sparse modif.	full dense
wavefronts	circles	non-circles	non-circles
3D S waves	coupled	decoupled	decoupled

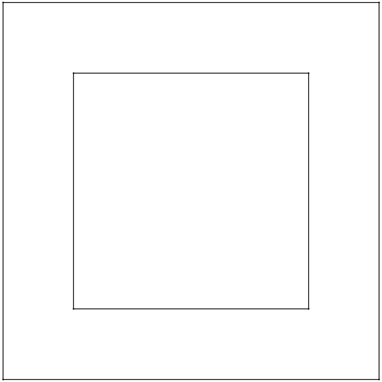
TTI example


a)
Distance (km)
4.0
5.0
6.0
7.0
8.0
0.0
1.0
2.0
3.0
Depth (km)
RTM in tilted TI model
b)
Distance (km)
4.0
5.0
6.0
7.0
8.0
0.0
1.0
2.0
3.0
Depth (km)
RTM in VTI model

The problem: PML

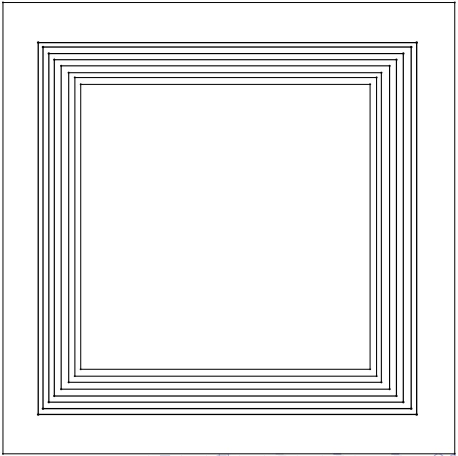
Perfectly Matched Layers

Add a surrounding zone



PML are instable in TI

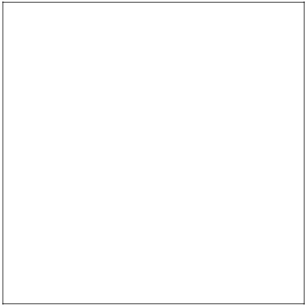
Add a taper zone from TI to iso.



Idea: ABC

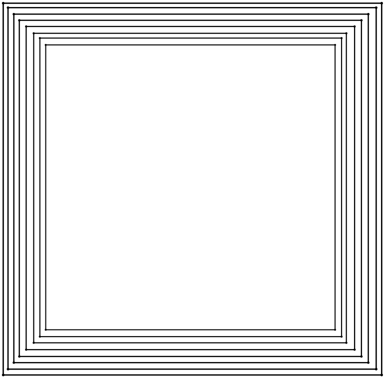
Absorbing Boundary Condition

No additional zone!



ABC induces spurious waves

Add a taper zone (only one)



Enquist-Majda methodology

Transparent Boundary Condition (TBC)

- Fourier transform in the directions parallel to the boundary
- ODE: $\partial_x \widehat{W} + M\widehat{W} = 0$, M invertible
- (Λ_i, P_i) eigenvalues/eigenvectors of M
- $\widehat{W} = \widehat{W}_0 e^{Mx} \Rightarrow$ TBC

From TBC to ABC: approximation of a pseudo-differential operator

2D isotropic eigenvalues

$$\begin{cases} \lambda_p & = & \sqrt{k^2 - w^2/V_p^2} & \approx & \frac{iw}{V_p} \\ \lambda_s & = & \sqrt{k^2 - w^2/V_s^2} & \approx & \frac{iw}{V_s} \end{cases}$$

Enquist-Majda methodology

Transparent Boundary Condition (TBC)

- Fourier transform in the directions parallel to the boundary
- ODE: $\partial_x \widehat{W} + M\widehat{W} = 0$, M invertible
- (Λ_i, P_i) eigenvalues/eigenvectors of M
- $\widehat{W} = \widehat{W_0} e^{Mx} \Rightarrow$ TBC

From TBC to ABC: approximation of a pseudo-differential operator

2D VTI eigenvalues

$$\lambda_{P/S} = \sqrt{\alpha k^2 - \beta \rho w^2 \pm \sqrt{\gamma k^4 - \eta \rho k^2 w^2 + \xi \rho^2 w^4}}$$

and TTI is worse... $\Rightarrow$ this methodology seems not usable!

P-waves and S-waves splitting

Elastic sub-systems

VTI P-waves:

$$\begin{cases} \rho\partial_t v_x &= \partial_x \sigma_{xx} \\ \rho\partial_t v_z &= \partial_z \sigma_{zz} \\ \partial_t \sigma_{xx} &= C_{11}\partial_x v_x + C_{13}\partial_z v_z \\ \partial_t \sigma_{zz} &= C_{13}\partial_x v_x + C_{33}\partial_z v_z \end{cases} \tag{1}$$

VTI S-waves:

$$\begin{cases} \rho\partial_t v_x &= \partial_x \sigma_{xx} + \partial_z \sigma_{xz} \\ \rho\partial_t v_z &= \partial_x \sigma_{xz} + \partial_z \sigma_{zz} \\ \partial_t \sigma_{xx} &= -2C_{66}\partial_z v_z \\ \partial_t \sigma_{zz} &= -2C_{66}\partial_x v_x \\ \partial_t \sigma_{xz} &= C_{66}(\partial_x v_z + \partial_z v_x) \end{cases} \tag{2}$$

P-waves and S-waves splitting

Elastic sub-systems ABC

VTI P-waves:

$$\begin{cases} \sigma_{xx} \approx -\rho V_p \kappa v_x \\ \sigma_{xz} \approx 0 \end{cases} \tag{1}$$

VTI S-waves:

$$\begin{cases} \sigma_{xx} \approx 0 \\ \sigma_{xz} \approx -\rho V_s v_z \end{cases} \tag{2}$$

Elastic P-S system ABC

$$\begin{cases} \sigma_{xx} \approx -\rho V_p \kappa v_x \\ \sigma_{xz} \approx -\rho V_s v_z \end{cases} \tag{3}$$

Geometrical point of view

Hypothesis: elliptic anisotropy $\boxed{\delta = \varepsilon}$

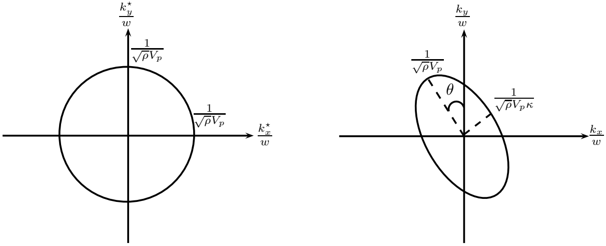


Figure: P-waves **slowness curves** of isotropic (left) and TTI (right) cases

Circle equation:

$$\rho V_p^2 \kappa^2 k_x^{\star 2} + \rho V_p^2 k_z^{\star 2} = 1 \tag{4}$$

Rotated ellipsoid equation:

$$\rho V_p^2 \kappa^2 (k_x \cos\theta + k_z \sin\theta)^2 + \rho V_p^2 (-k_x \sin\theta + k_z \cos\theta)^2 = 1 \tag{5}$$

Change of coordinates

Propagation modes k from the geometry:

$$k_x = \mu_1 k_x^\star + \mu_2 k_z^\star \quad \text{and} \quad k_z = \mu_3 k_x^\star + \mu_4 k_z^\star \tag{6}$$

Velocity v from the second-order equation:

$$v_x = \alpha_1 v_x^\star + \alpha_2 v_z^\star \quad \text{and} \quad v_z = \alpha_3 v_x^\star + \alpha_4 v_z^\star \tag{7}$$

Stress tensor σ from the first-order equation:

$$\sigma_{xx} = \widetilde{\beta}_1 p^\star \quad \text{and} \quad \sigma_{zz} = \widetilde{\beta}_2 p^\star \quad \text{and} \quad \sigma_{xz} = \widetilde{\beta}_3 p^\star \tag{8}$$

$\rightarrow$ apply the opposite change of coordinate on the isotropic ABCs

Change of coordinates 1/3

Propagation modes

$$k_x = \mu_1 k_x^\star + \mu_2 k_z^\star \quad \text{and} \quad k_z = \mu_3 k_x^\star + \mu_4 k_z^\star \tag{9}$$

For a vertical boundary: $\mu_3 = 0$ (otherwise $\mu_2 = 0$)

Circle equation:

$$\rho V_p^2 \kappa^2 k_x^{\star 2} + \rho V_p^2 k_z^{\star 2} = 1 \tag{10}$$

Rotated ellipsoid equation:

$$\rho V_p^2 \kappa^2 (k_x \cos\theta + k_z \sin\theta)^2 + \rho V_p^2 (-k_x \sin\theta + k_z \cos\theta)^2 = 1 \tag{11}$$

$$\begin{cases} \mu_1 &= \sqrt{1/(\kappa^2 \cos^2\theta + \sin^2\theta)} \\ \mu_2 &= -(\kappa^2 - 1)\cos\theta \sin\theta / \kappa \sqrt{\kappa^2 \cos^2\theta + \sin^2\theta} \\ \mu_4 &= \frac{1}{\kappa}\sqrt{\kappa^2 \cos^2\theta + \sin^2\theta} \end{cases} \tag{12}$$

Change of coordinates 2/3

Velocity v

$$v_x = \alpha_1 v_x^\star + \alpha_2 v_z^\star \quad \text{and} \quad v_z = \alpha_3 v_x^\star + \alpha_4 v_z^\star \tag{13}$$

Propagation modes change of coordinates:

$$k_x = \mu_1 k_x^\star + \mu_2 k_z^\star \quad \text{and} \quad k_z = \mu_3 k_x^\star + \mu_4 k_z^\star \tag{14}$$

Second-order TTI elastic system, with $V = (v_x, v_z)$:

$$\rho \partial_t^2 V = [C_{xx} k_x^2 + C_{xz} k_x k_z + C_{zz} k_z^2] V \tag{15}$$

Second-order isotropic elastic system:

$$\rho \partial_t^2 V^\star = \rho V_p^2 [I_{xx} k_x^{2^\star} + I_{xz} k_x^\star k_z^\star + I_{zz} k_z^{2^\star}] V^\star \tag{16}$$

Change of coordinates 2/3

Velocity v

$$v_x = \alpha_1 v_x^\star + \alpha_2 v_z^\star \quad \text{and} \quad v_z = \alpha_3 v_x^\star + \alpha_4 v_z^\star \tag{13}$$

Propagation modes change of coordinates
Second-order TTI elastic system
Second-order isotropic elastic system

$$\left\{ \begin{array}{lcl} v_x & = & \dfrac{\sqrt{\rho} V_p}{\sqrt{\kappa^2 \cos^2\theta + \sin^2\theta}} [-(\kappa \cos^2\theta + \sin^2\theta) v_x^\star - (\kappa - 1)\cos\theta \sin\theta v_z^\star] \\ v_z & = & \dfrac{\sqrt{\rho} V_p}{\sqrt{\kappa^2 \cos^2\theta + \sin^2\theta}} [(\kappa - 1)\cos\theta \sin\theta v_x^\star - (\kappa \cos^2\theta + \sin^2\theta) v_z^\star] \end{array} \right.$$

Change of coordinates 3/3

Stress tensor σ

$$\sigma_{xx} = \widetilde{\beta}_1 p^\star \quad \text{and} \quad \sigma_{zz} = \widetilde{\beta}_2 p^\star \quad \text{and} \quad \sigma_{xz} = \widetilde{\beta}_3 p^\star \tag{14}$$

Propagation modes change of coordinates:

$$k_x = \mu_1 k_x^\star + \mu_2 k_z^\star \quad \text{and} \quad k_z = \mu_3 k_x^\star + \mu_4 k_z^\star \tag{15}$$

First-order TTI elastic system:

$$\begin{cases} \rho \partial_t v_x &= k_x \sigma_{xx} + k_z \sigma_{xz} \\ \rho \partial_t v_z &= k_x \sigma_{xz} + k_z \sigma_{zz} \\ \partial_t \sigma_{xx} &= \widetilde{C_{11}} k_x v_x + \widetilde{C_{13}} k_z v_z + \widetilde{C_{16}} (k_x v_z + k_z v_x) \\ \partial_t \sigma_{zz} &= \widetilde{C_{13}} k_x v_x + \widetilde{C_{33}} k_z v_z + \widetilde{C_{26}} (k_x v_z + k_z v_x) \\ \partial_t \sigma_{xz} &= \widetilde{C_{16}} k_x v_x + \widetilde{C_{26}} k_z v_z + \widetilde{C_{66}} (k_x v_z + k_z v_x) \end{cases} \tag{16}$$

Change of coordinates 3/3

Stress tensor σ

$$\sigma_{xx} = \widetilde{\beta}_1 p^\star \quad \text{and} \quad \sigma_{zz} = \widetilde{\beta}_2 p^\star \quad \text{and} \quad \sigma_{xz} = \widetilde{\beta}_3 p^\star \tag{14}$$

First-order isotropic elastic system:

$$\begin{cases} \rho \partial_t {v_x}^\star & = & k_x^\star p^\star \\ \rho \partial_t {v_z}^\star & = & k_z^\star p^\star \\ \partial_t p^\star & = & \rho V_p^2 k_x^\star {v_x}^\star + \rho V_p^2 k_z^\star {v_z}^\star \end{cases} \tag{15}$$

Velocity v change of coordinates:

$$v_x = \alpha_1 v_x^\star + \alpha_2 v_z^\star \quad \text{and} \quad v_z = \alpha_3 v_x^\star + \alpha_4 v_z^\star \tag{16}$$

Change of coordinates 3/3

Stress tensor σ

$$\sigma_{xx} = \widetilde{\beta}_1 p^\star \quad \text{and} \quad \sigma_{zz} = \widetilde{\beta}_2 p^\star \quad \text{and} \quad \sigma_{xz} = \widetilde{\beta}_3 p^\star \tag{14}$$

Propagation modes change of coordinates
First-order TTI elastic system
First-order isotropic elastic system
Velocity v change of coordinates

$$\begin{cases} \sigma_{xx} &= -\sqrt{\rho} V_p (\kappa \cos^2\theta + \sin^2\theta) p^\star \\ \sigma_{zz} &= -\sqrt{\rho} V_p (\kappa \sin^2\theta + \cos^2\theta) p^\star \\ \sigma_{xz} &= \sqrt{\rho} V_p (\kappa - 1) \cos\theta \sin\theta p^\star \end{cases} \tag{15}$$

TTI ABC formulation

P-wave ABC

Isotropic:

$$\begin{cases} \sigma_{xx}^* & = & -\rho V_p v_x^* \\ \sigma_{xz}^* & = & 0 \end{cases}$$

TTI (change of coordinates):

$$\begin{cases} \sigma_{xx} & = & -\rho V_p \frac{\kappa \cos^2\theta + \sin^2\theta}{\sqrt{\kappa^2\cos^2\theta + \sin^2\theta}} \left[(\kappa \cos^2\theta + \sin^2\theta) v_x + (\kappa - 1)\cos\theta \sin\theta v_z \right] \\ \sigma_{xz} & = & -\rho V_p \frac{(\kappa - 1)\cos\theta \sin\theta}{\sqrt{\kappa^2\cos^2\theta + \sin^2\theta}} \left[(\kappa \cos^2\theta + \sin^2\theta) v_x + (\kappa - 1)\cos\theta \sin\theta v_z \right] \end{cases}$$

S-wave ABC

$$\begin{cases} \sigma_{xx} & = & 0 \\ \sigma_{xz} & = & -\rho V_s v_z \end{cases}$$

Same methodology in 3D

$$
\begin{cases}
\sigma_{xx} = -\sqrt{\rho} V_p & \dfrac{\kappa \sin^2\phi + \cos^2\phi(\kappa\cos^2\theta + \sin^2\theta)}{\sqrt{\kappa^2\cos^2\theta\cos^2\phi + \kappa^2\sin^2\phi + \sin^2\theta\cos^2\phi}} [\\
& +(\kappa\sin^2\phi + \cos^2\phi(\kappa\cos^2\theta + \sin^2\theta))v_x \\
& -(\kappa - 1)\cos\phi\sin\phi\sin^2\theta v_y - (\kappa - 1)\cos\theta\sin\theta\cos\phi v_z] \\
\sigma_{xz} = -\sqrt{\rho} V_p & \dfrac{(\kappa - 1)\cos\theta\sin\theta\cos\phi}{\sqrt{\kappa^2\cos^2\theta\cos^2\phi + \kappa^2\sin^2\phi + \sin^2\theta\cos^2\phi}} [\\
& -(\kappa\sin^2\phi + \cos^2\phi(\kappa\cos^2\theta + \sin^2\theta))v_x \\
& +(\kappa - 1)\cos\phi\sin\phi\sin^2\theta v_y + (\kappa - 1)\cos\theta\sin\theta\cos\phi v_z] \\
\sigma_{xy} = -\sqrt{\rho} V_p & \dfrac{(\kappa - 1)\cos\phi\sin\phi\sin^2\theta}{\sqrt{\kappa^2\cos^2\theta\cos^2\phi + \kappa^2\sin^2\phi + \sin^2\theta\cos^2\phi}} [\\
& -(\kappa\sin^2\phi + \cos^2\phi(\kappa\cos^2\theta + \sin^2\theta))v_x \\
& +(\kappa - 1)\cos\phi\sin\phi\sin^2\theta v_y + (\kappa - 1)\cos\theta\sin\theta\cos\phi v_z]
\end{cases}
\tag{16}
$$

TTI results

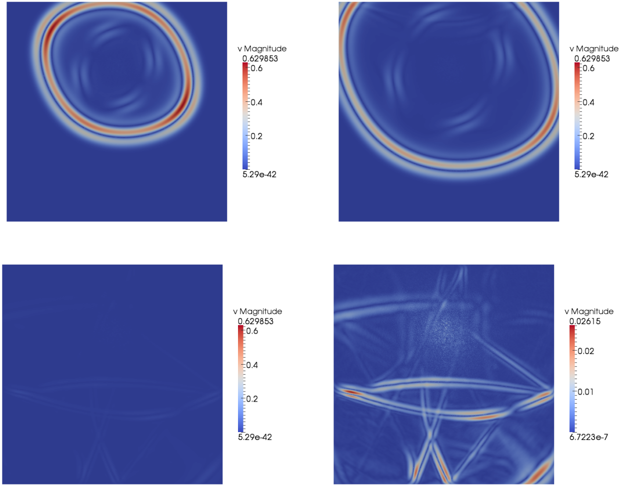


Figure: Velocity magnitude at different time steps of the simulation

TTI ABC versus simple ABCs

TTI medium, with different kind of ABCs:

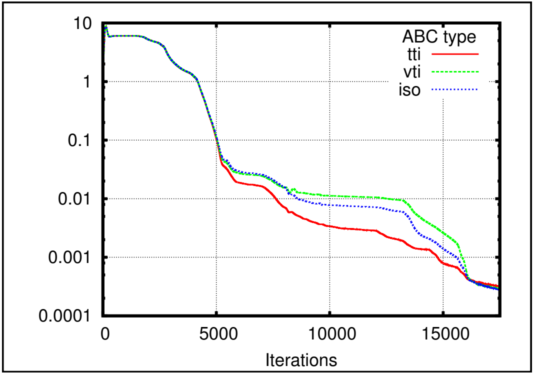
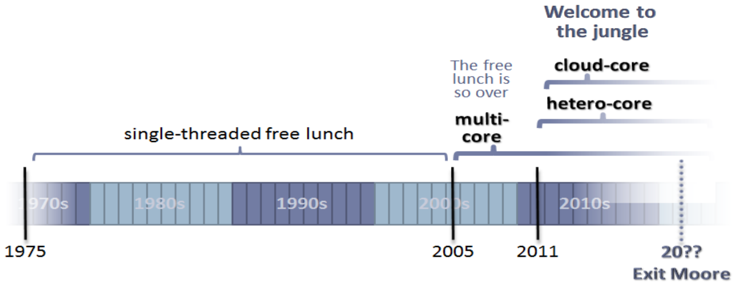


Figure: TTI ABC versus simple ABCs

$\Rightarrow$ TTI ABC is better than isotropic or VTI ABCs!

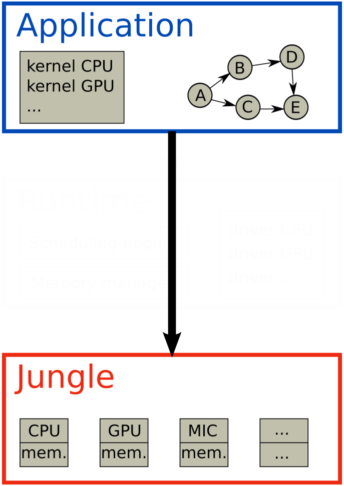
Outline

HPC evolution



- Monoprocessors: easy to program, automatic updates
- Multicores: start thinking parallel, local and remote memories
- Accelerators: peak vs real performances, special language
- Manycores: at least two levels of parallelism

The problem: heterogeneity



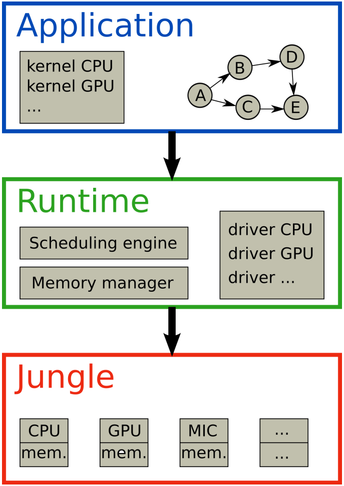
Classical approach:

- MPI over CPUs
- CUDA over GPUs

implies:

- big programming effort
- difficult to maintain
- hardware-dependent

Idea: use a runtime and task-based programming



Runtime system:

- abstraction layer
- hiding heterogeneity

Scheduler decides:

- where to execute
- when to execute

Memory manager:

- does the transfer
- guarantees consistency

Discretization

Elastic wave equation (first order)

$$\begin{cases} \rho(\mathbf{x})\partial_t v(\mathbf{x},t) & = & \nabla.\underline{\underline{\sigma}}(\mathbf{x},t) \\ \partial_t \underline{\underline{\sigma}}(\mathbf{x},t) & = & \underline{\underline{C}}(\mathbf{x}) : \underline{\underline{\epsilon}}(v(\mathbf{x},t)) \end{cases} \quad (17)$$

Discontinuous Galerkin with Leap-frog scheme

Iteration on n:

$$\begin{cases} M_v \dfrac{v_h^{n+1} - v_h^n}{\Delta t} + R_{\underline{\underline{\sigma}}} \underline{\underline{\sigma}}_h^{n+1/2} & = & 0 \\ M_{\underline{\underline{\sigma}}} \dfrac{\underline{\underline{\sigma}}_h^{n+3/2} - \underline{\underline{\sigma}}_h^{n+1/2}}{\Delta t} + R_v v_h^{n+1} & = & 0 \end{cases} \quad (18)$$

M_v and $M_{\underline{\underline{\sigma}}}$ block-diagonal matrices $\Rightarrow$ easily invertible!

Sequential algorithm

Quasi-explicit reformulation

$$\begin{cases} v_h^{n+1} & = \quad v_h^n + M_v^{-1}[\Delta t R_{\underline{\underline{\sigma}}} \underline{\underline{\sigma}}_h^{n+1/2}] \\ \underline{\underline{\sigma}}_h^{n+3/2} & = \quad \underline{\underline{\sigma}}_h^{n+1/2} + M_{\underline{\underline{\sigma}}}^{-1}[\Delta t R_v v_h^{n+1}] \end{cases} \tag{19}$$

Algorithme 1 : DIP sequential

Data : N_h, Δ_t, N_t

Result : v_h, $\underline{\underline{\sigma}}_h$

$[v_h^1, \underline{\underline{\sigma}}_h^{3/2}] \leftarrow$Initialization$(N_h, \Delta_t)$;

for $n = 1..N_t$ **do**

  for $K = 1..N_h$ **do**

    $v_{h_K}^{n+1} \quad \leftarrow$ UpdateVelocity$(v_{h_K}^n, \underline{\underline{\sigma}}_{h_{\overline{K}}}^{n+1/2}, \Delta_t)$;

  end

  for $K = 1..N_h$ **do**

    $\underline{\underline{\sigma}}_{h_K}^{n+3/2} \leftarrow$ UpdateStress$(\underline{\underline{\sigma}}_{h_K}^{n+1/2}, v_{h_{\overline{K}}}^{n+1}, \Delta_t)$;

  end

end

Parallel algorithm

Algorithme 2 : DIP parallel

Data : N_p, N_h, Δ_t, N_t

Result : v_h, $\underline{\underline{\sigma}}_h$

$N_{h_{loc}} \leftarrow$ DomainDecomposition(N_p);

$[v_h^1, \underline{\underline{\sigma}}_h^{3/2}] \leftarrow$Initialization($N_{h_{loc}}, \Delta_t$);

for $n = 1..N_t$ **do**

  $\underline{\underline{\sigma}}_{h_{\overline{K}}}^{n+1/2} \leftarrow$ Communication($\underline{\underline{\sigma}}_{h_K}^{n+1/2}$);

  for $K = 1..N_{h_{loc}}$ **do**

    $v_{h_K}^{n+1} \leftarrow$ UpdateVelocity($v_{h_K}^n, \underline{\underline{\sigma}}_{h_{\overline{K}}}^{n+1/2}, \Delta_t$);

  end

  $v_{h_{\overline{K}}}^{n+1} \leftarrow$ Communication($v_{h_K}^{n+1}$);

  for $K = 1..N_{h_{loc}}$ **do**

    $\underline{\underline{\sigma}}_{h_K}^{n+3/2} \leftarrow$ UpdateStress($\underline{\underline{\sigma}}_{h_K}^{n+1/2}, v_{h_{\overline{K}}}^{n+1}, \Delta_t$);

  end

end

MPI version = one domain per process

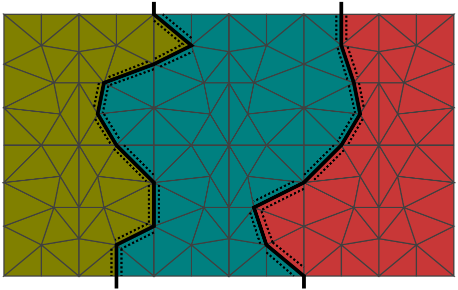


Figure: Domain decomposition example

Load balancing limitations:

- order of (discretization of) each cell and direct neighbors
- architecture specifications (e.g. SIMD, cache size, ...)

$\Rightarrow$ not straightforward to estimate the computation time for a domain!

Load imbalance example

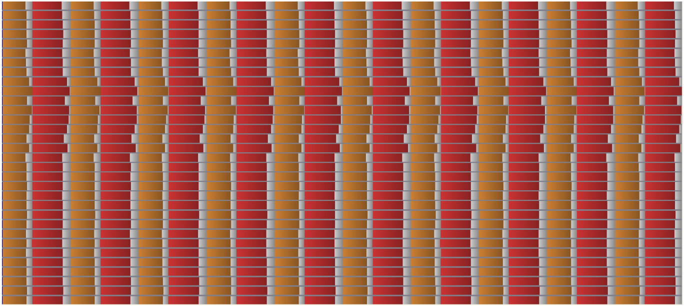


Figure: Trace analysis of 10 timesteps on 32 MPI processes

Each line represents a core activity during the execution time:

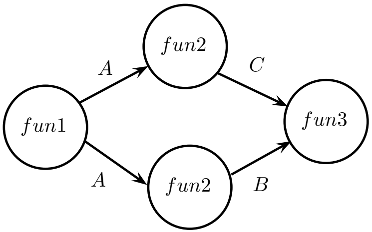
- ORANGE – computation of the velocity field
- RED – computation of the stress tensor
- GREY – waiting time (synchronization)

First step: the DAG (Direct Acyclic Graph)

Sequential code

$A \leftarrow fun1();$
$B \leftarrow fun2(A);$
$C \leftarrow fun2(A);$
$C \leftarrow fun3(B, C);$

- Describe the functions as tasks
- Form the dataflow: specify data dependencies



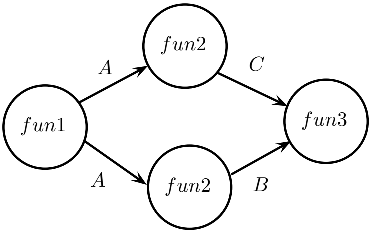
Second step: task description

Parametrized model

Task_fun1()
 WRITE A $\rightarrow$ A Task_fun2(1)
 $\rightarrow$ A Task_fun2(2)

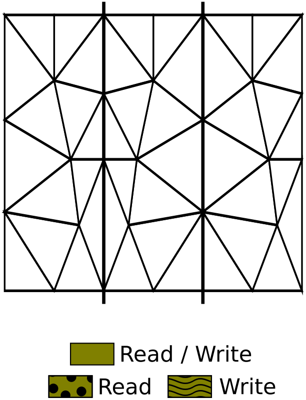
Task_fun2(n)
n=1..2
 READ A $\leftarrow$ A Task_fun1()
 WRITE X $\rightarrow$ (n==1) ? B Task_fun3()
 $\rightarrow$ (n==2) ? C Task_fun3()

Task_fun3()
 READ B $\leftarrow$ B Task_fun2(1)
 RW C $\leftarrow$ C Task_fun2(2)



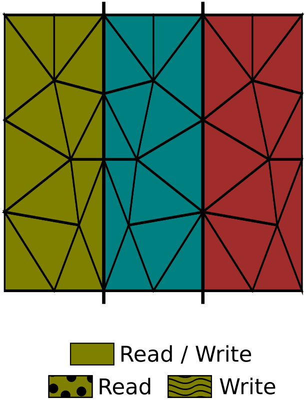
Multidomain DAG

Basic COMPUTE and EXCHANGE model:



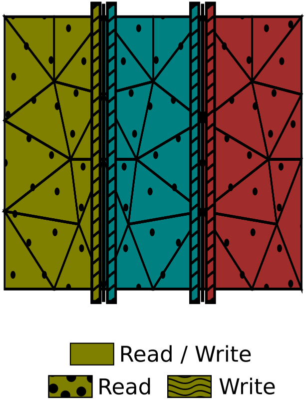
Multidomain DAG

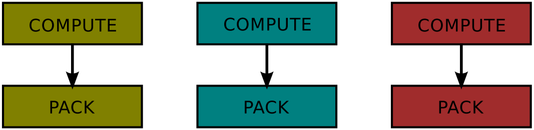
Basic COMPUTE and EXCHANGE model:



Multidomain DAG

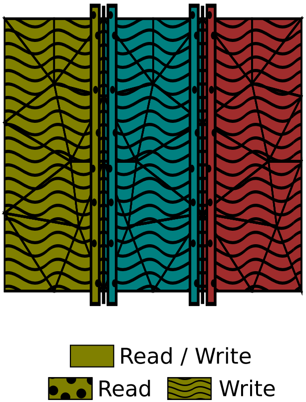
Basic COMPUTE and EXCHANGE model:

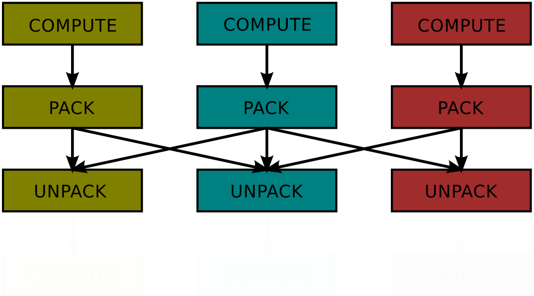




Multidomain DAG

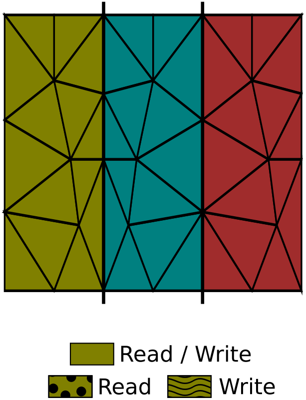
Basic COMPUTE and EXCHANGE model:

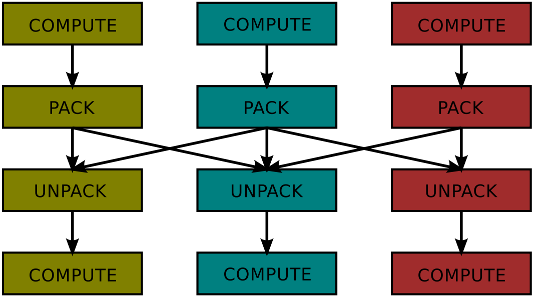




Multidomain DAG

Basic COMPUTE and EXCHANGE model:





Fine granularity

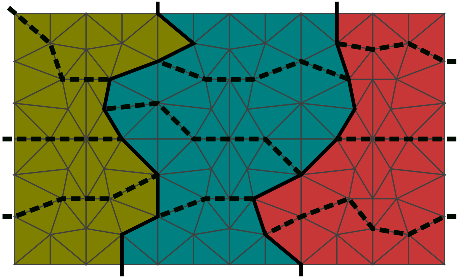


Figure: Subdivision example

More than one domain per CPU

- exhibit deeper parallelism
- allow dynamic flexibility
- reduce the boundary size

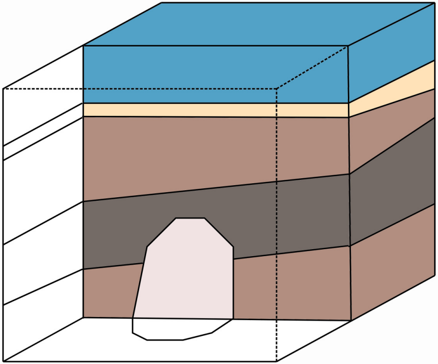
Geophysics test case

Realistic test case:

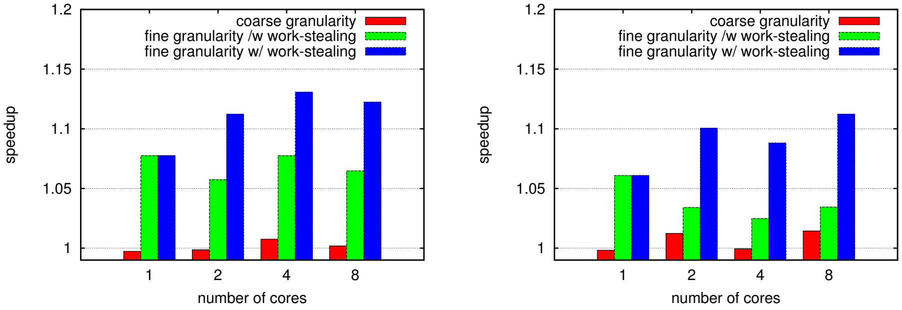
- 3D elastic
- TTI (anisotropy)
- multi-layers

Hybrid discretization:

- unstructured tetrahedra
- P1-P2-P3 orders
- boundary conditions



One processor behavior



⇒ both fine granularity and work-stealing are essential!

ccNUMA machine

- 8 processors (previous Intel Xeon E7-8837)
- Total of 64 CPU cores in ccNUMA architecture

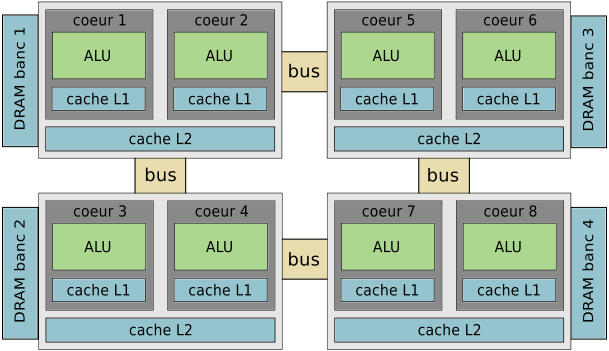
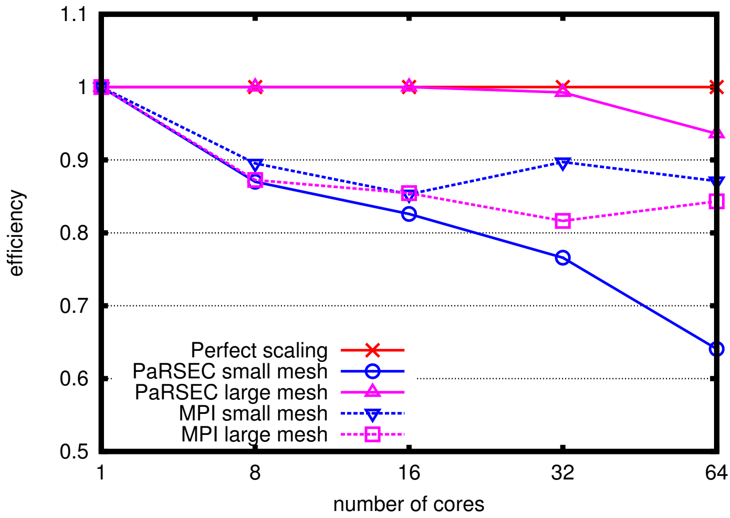


Figure: cache-coherent Non-Uniform Memory Access (ccNUMA) scheme

ccNUMA results - efficiency



Trace comparison

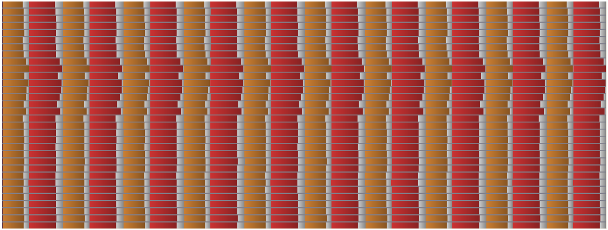


Figure: MPI-based $t = 2.517s$

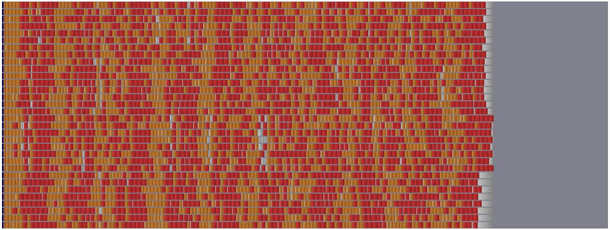
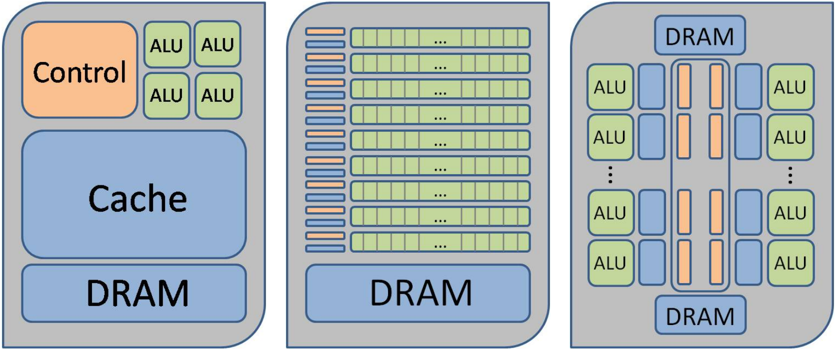
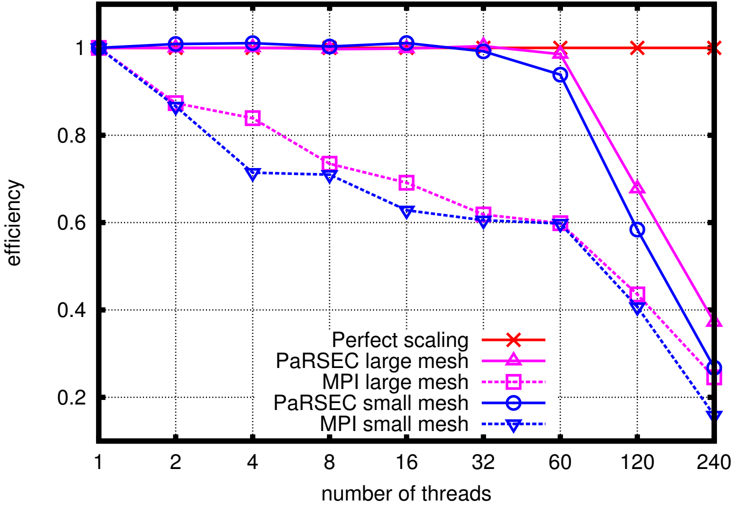


Figure: PaRSEC version (NUMA-aware, granularity x6) $t = 2.060s$

Architectures



Intel Xeon Phi results - efficiency



Conclusion & perspectives

Context

- Geophysics: Seismic Imaging (waves)
- TOTAL collaboration: DIP (TMBM)

Results

- TI without additional cost: ABC (with taper) vs PML
- (portable) perfect efficiency: task-based vs MPI parallelism

Next

- many updates in TMBM
- GPU option with OpenACC