

**Modélisations par réseaux d'automates cellulaires et  
simulations parallèles du phénomène de subduction-érosion  
en tectonique des plaques**

**Thomas LEDUC**

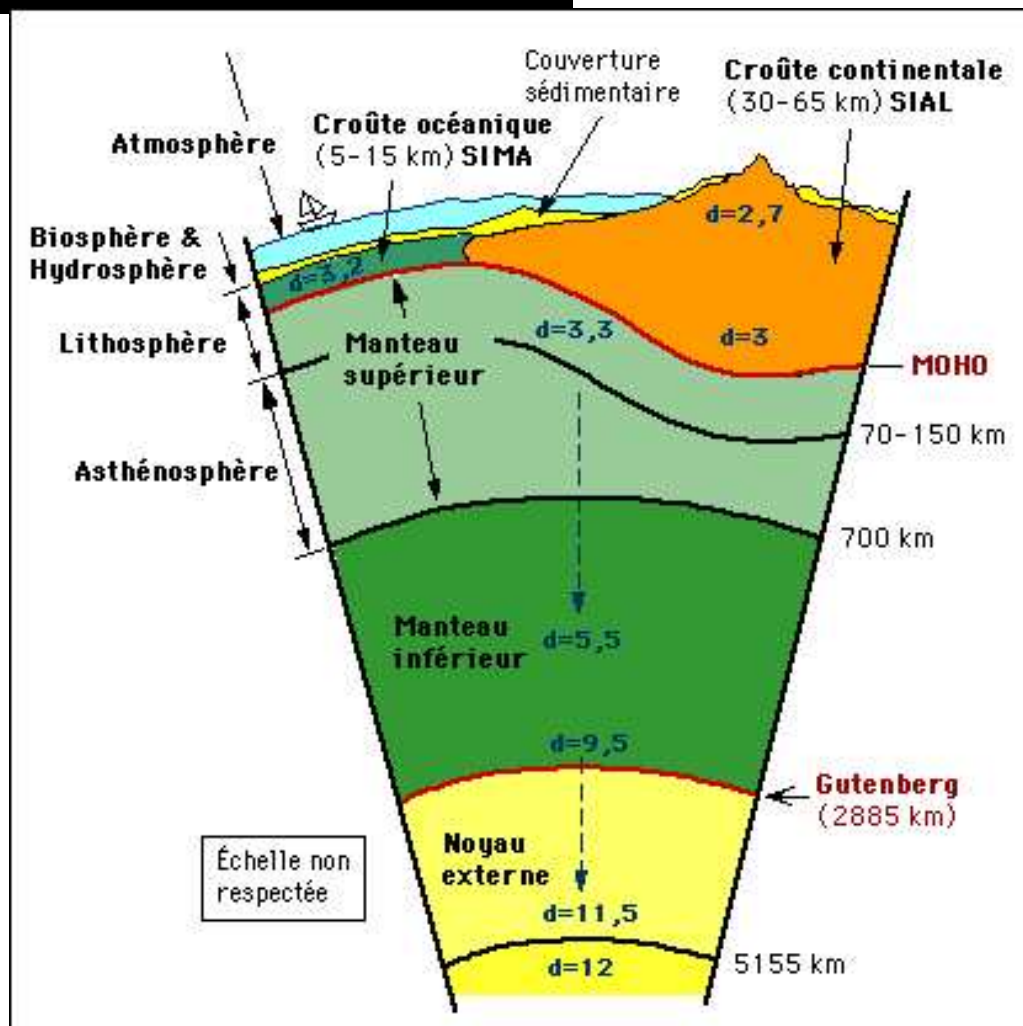
**Juillet 1999**

## Contenu de l'exposé

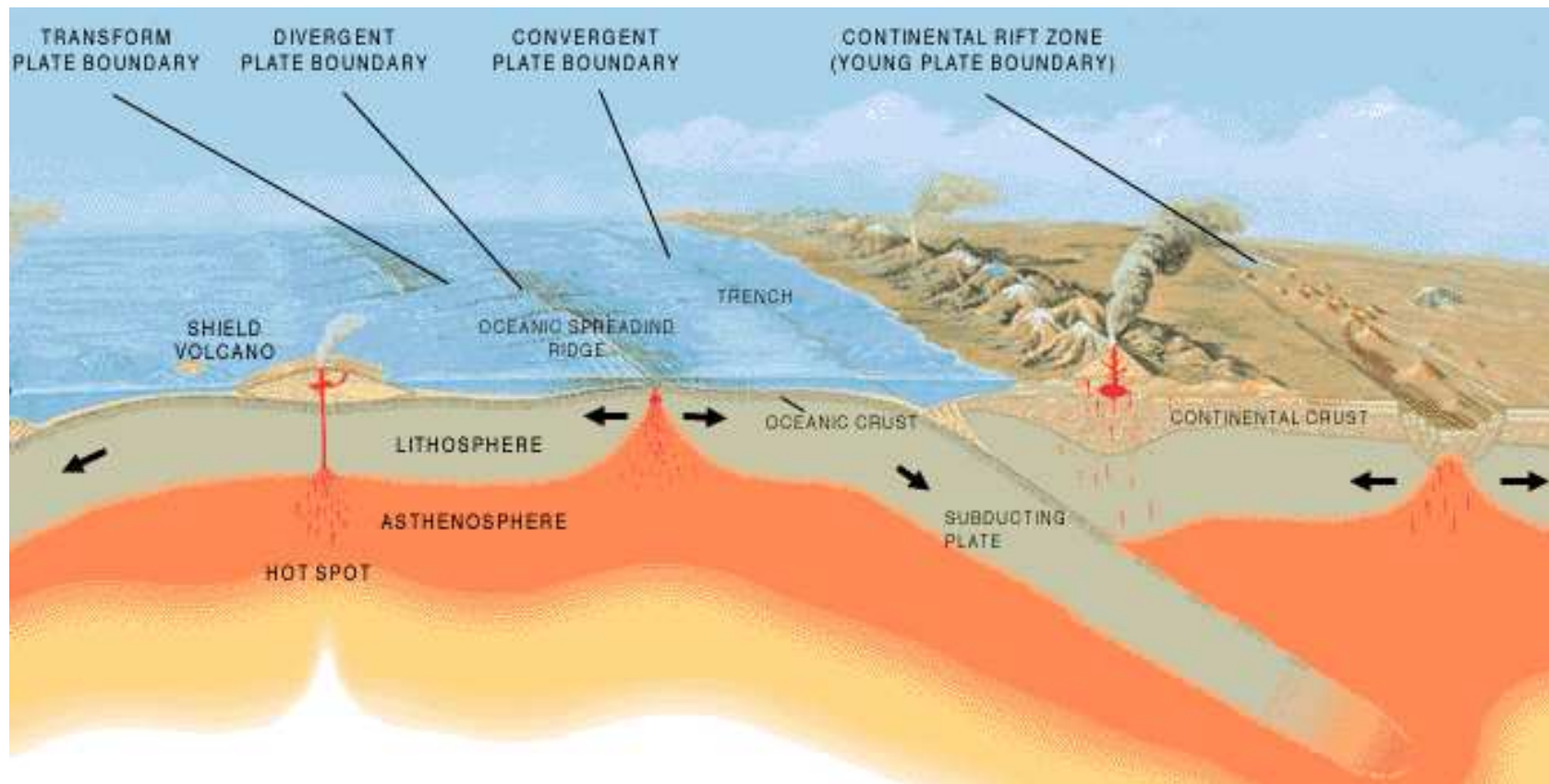
- ➡ rappels sur la tectonique des plaques,
- ➡ choix d'une modélisation discrète et SPM,
- ➡ présentation des modèles,
- ➡ présentation des simulations et de leurs optimisations parallèles,
- ➡ résultats et performances,
- ➡ conclusion et évolutions.

# Le contexte géotectonique

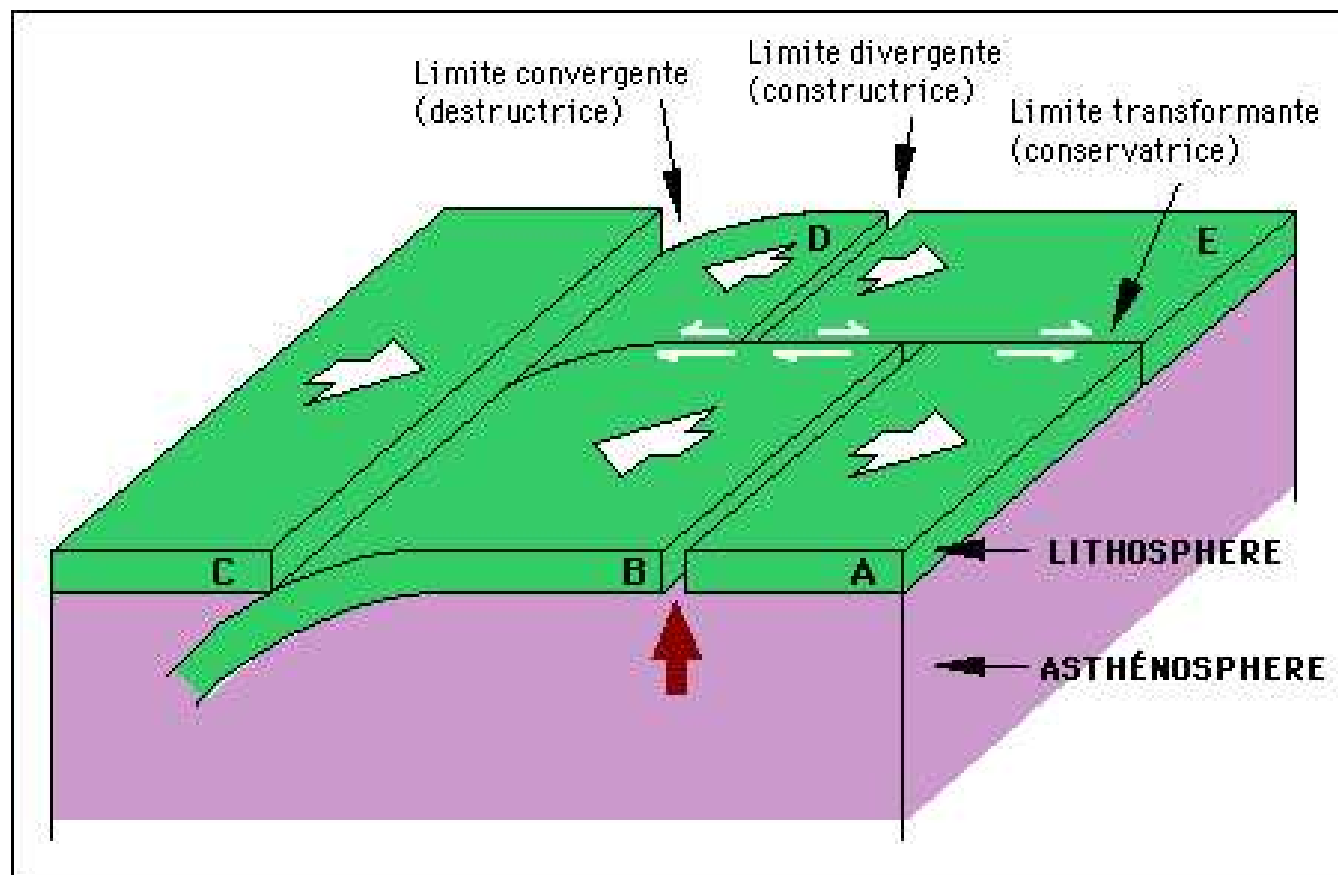
↳ le globe terrestre est une structure stratifiée...



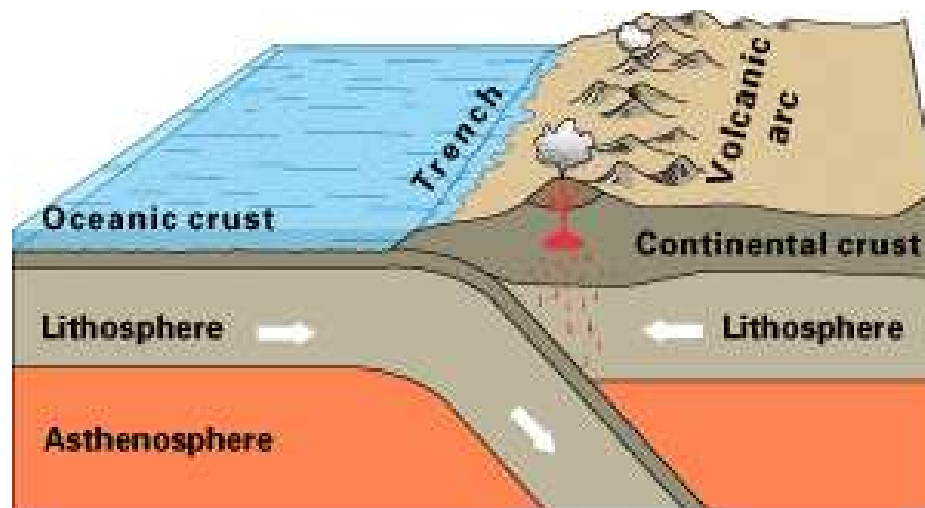
↳ composée d'une multitude de calottes sphériques : les plaques...



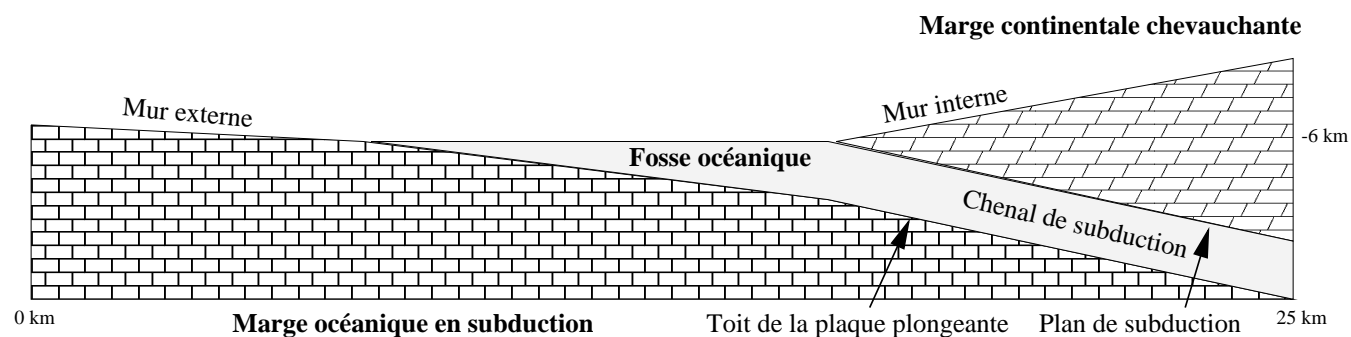
↳ en mouvement permanent...



↳ cas particulier d'une subduction de marge océanique sous une marge continentale,



Oceanic-continental convergence



## L'existant vs. notre stratégie de modélisation

↳ l'existant en matière de modélisation : simulations analogiques expérimentales et simulations numériques avec résolution d'EDP par éléments finis. But : satisfaire des *critères de similarité physique*.

Problèmes :

▶ transferts d'échelles et aspects thermiques,  
▶ indications à petites échelles sur l'épaississement ou l'amincissement.  
Comportement de *milieu*, pas de prise en compte des *individualités*.

↳ modélisation par systèmes dynamiques discrets. La conjonction de règles élémentaires locales engendre des comportements d'ensemble complexes :

▶ modélisation de la convection par une méthode LGA,  
▶ modélisations en sismologie, en érosion de terrain. . .

↳ réseau d'automates cellulaires : juxtaposition d'une même règle de transition, selon un mode opératoire synchrone, sur des cellules placées aux sommets d'un graphe simple orienté, connexe et d-régulier,

localité  
uniformité  
synchronisation  
parallélisme



## Analogies avec le SPM

↳ le *Sand Pile Model* 1D :

▶ physique de l'état granulaire (Bak, Tang et Wiesenfeld - 1980),

▶ réseau d'automates cellulaires en ligne,

▶ un automate correspond à une hauteur de pile de sable,

▶ loi de transition :

$$\text{avec : } \mathbb{I}(n) = \begin{cases} 0 & \text{si } n < 2 \\ 1 & \text{sinon} \end{cases}$$

$$C_j^{t+1} = C_j^t - \mathbb{I}(C_j^t - C_{j-1}^t) - \mathbb{I}(C_j^t - C_{j+1}^t) + \mathbb{I}(C_{j-1}^t - C_j^t) + \mathbb{I}(C_{j+1}^t - C_j^t)$$

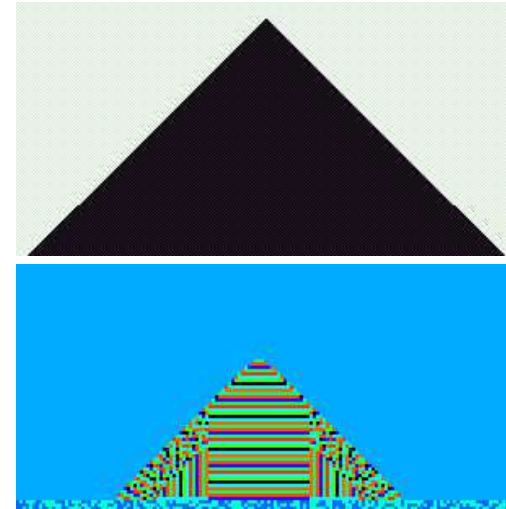
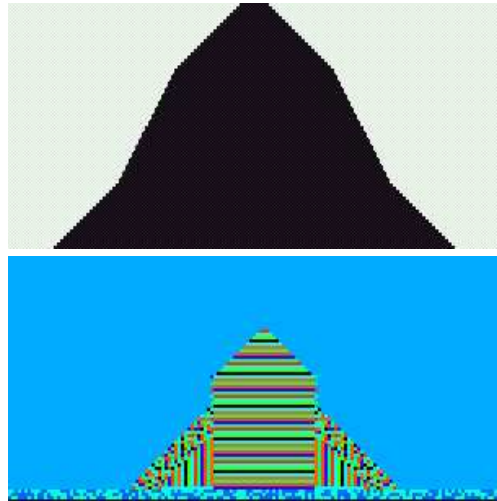
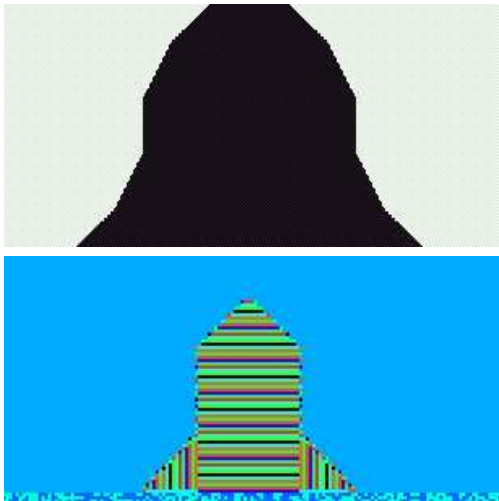
↳ étendu au cas bi-dimensionnel :

▶ grille régulière en 2 dimensions d'automates cellulaires,

▶ individualisation de chaque "grain" (avec attributs de nature et de couleur pour chaque automate),

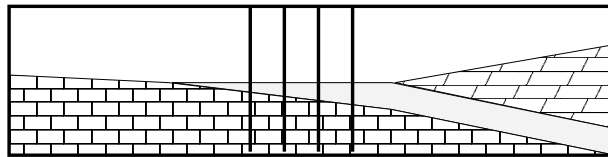
▶ loi de transition plus complexe à écrire puisque fonction d'un voisinage plus grand (voisinage de Moore étendu),

↳ rendus visuels comparés :

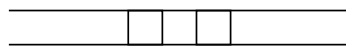


# Les modèles

Automate cellulaire 1D



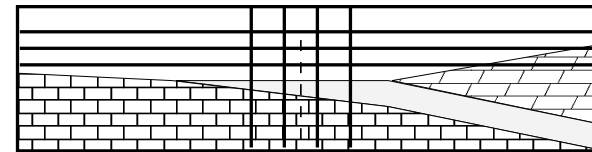
Vecteur de cellules :  $C^t$



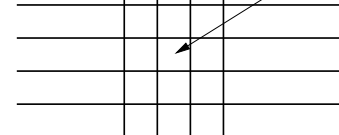
$C_j^t$   
(7 altitudes par cellule)

découpage en colonnes de la coupe  
étudiée

Automate cellulaire 2D



Matrice de cellules :  $C^t$



découpage en blocs de la coupe  
étudiée

démarche ascendante à base de règles d'interactions locales mutuelles.

### ↳ modélisation 1D :

▶ ligne finie de 1000 automates cellulaires,

▶ état d'une cellule : 7 hauteurs de couches, un coefficient de "marche" et un coefficient de "vieillesse"  
 ⇒ nombre fini d'états,

▶ 2 voisins,

▶ imbrication de trois "moteurs" correspondant à trois échelles de temps,

▶ mécanisme additionnel pour la transmission d'une donnée globale ( $Coeff_{Translation}$ ),

### ↳ modélisation 2D :

▶ grille régulière de 200000 automates cellulaires,

▶ état d'une cellule : 6 entiers bornés correspondant à la couleur, la nature, le vieillissement, la "courbure", la "mémoire" et l'altitude de la cellule courante,

▶ 24 voisins (voisinage de Moore étendu),

▶ mécanisme de réplication des cellules des bords internes singularisées par un état "frontière" invariant,

Représentation de  $(C^t)_{t \in \mathbb{N}}$  par un "jeu de bascule" entre deux tableaux C et CC.

↳ **algorithme simplifié 1D**

répéter

$CC \leftarrow C$

$C \leftarrow F_{lente}(CC, Coeff_{Tr})$

répéter

$CC \leftarrow C$

$C \leftarrow F_{moyenne}(CC, Coeff_{Tr})$

répéter

$CC \leftarrow C$

$C \leftarrow F_{rapide}(CC)$

jusqu'à condition d'arrêt 1

jusqu'à condition d'arrêt 2

saisie conditionnelle d'image

jusqu'à condition d'arrêt 3

↳ **algorithme simplifié 2D**

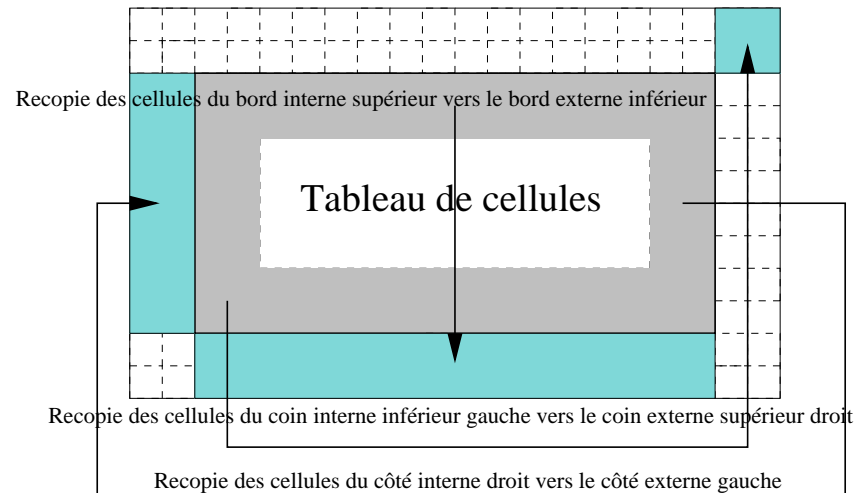
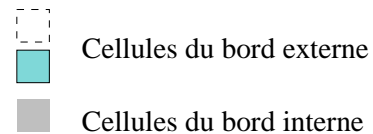
répéter

$CC \leftarrow C$

réplication des bords de CC

$C \leftarrow F(CC)$

jusqu'à condition d'arrêt



↳ 3 "moteurs" imbriqués du modèle 1D :

(on les retrouve en 2D à l'exception de la translation du front de la marge océanique)

- échelle de temps lente :  
subduction et génération des "brisures de pente",  
dégradation et érosion de la base de la plaque continentale,  
émission ou non du signal  $Coeff_{translation}$ ,
- échelle de temps intermédiaire :  
translation éventuelle du front de la marge océanique,  
nivellement de la fosse océanique,  
affaissements de terrain au sein de la plaque continentale,
- échelle de temps rapide :  
avalanches superficielles sur la plaque continentale.

## Les simulations

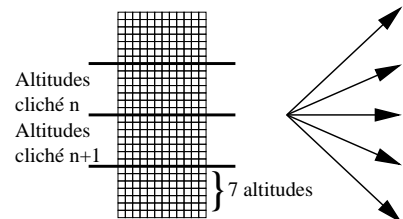
↳ de la nécessité de développer nos propres outils...

la plate-forme Cellular (= langage Cellang, compilateur cellc, machine virtuelle avcam et viewer cellview) ne correspond pas à nos besoins.

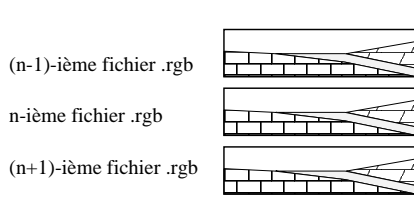
↳ et de bien distinguer les traitements des post-traitements graphiques.

▶ en 1D

Unique fichier de résultats  
(ensemble des altitudes stockées)



Ensemble de fichiers au format RGB  
(issus d'un premier post-traitement)



▶ en 2D

conversion par convert de fichiers SunRaster ou Raw au format GIF, puis assemblage par dmconvert des images GIF en une animation MPEG-1 vidéo

Raw → GIF → MPV

↳ **stratégie (commune) d'optimisation parallèle :**

▶ architectures MIMD (asynchrones, distribuées avec bus ou réseau de communication) - modèle par échange de messages

▶ méthode de décomposition de domaine

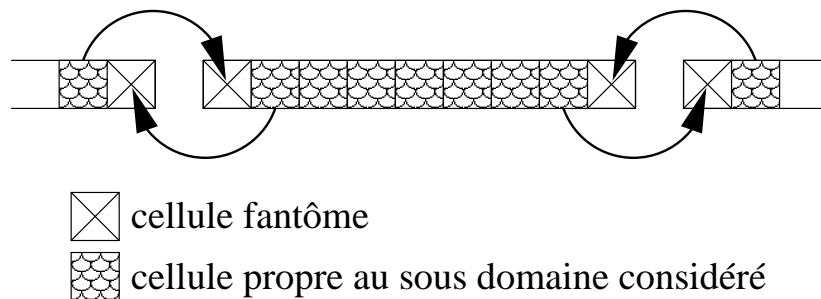
⇒ exploite la localité des données,

⇒ permet une résolution concurrente des sous-problèmes

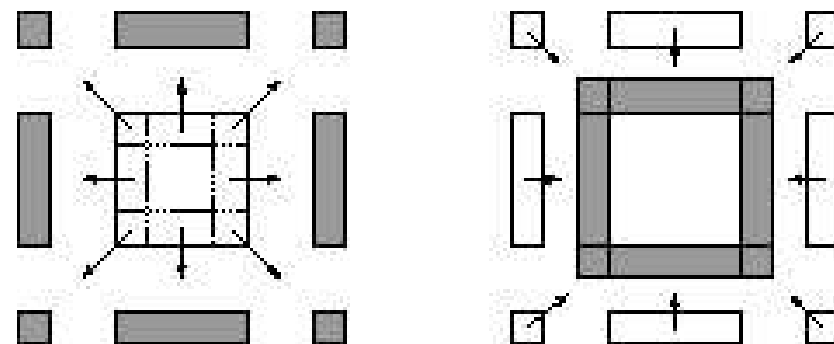
▶ méthode de décomposition de domaine avec recouvrement

⇒ résoudre le problème du stockage en local d'informations non locales...

en 1D



en 2D





↳ **algorithme simplifié :**

▶ recouvrir les temps de communication par des temps de calcul,

▶ principes communs aux deux fonctions de transition globales :

1. Envoi non bloquant des bords internes,
2. Mise à jour du sous-domaine (bords internes et externes exclus),
3. Réception bloquante des bords externes,
4. Réévaluation des états des bords internes.

avec quelques sauvegardes périodiques de la configuration du réseau d'automates cellulaires...

↳ gestion des entrées-sorties

désynchronisation d'un processus "superviseur" du reste des "itérateurs".

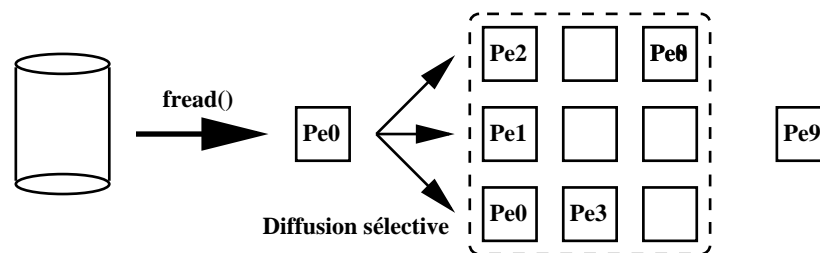
▶ Rôle du "superviseur" en 1D :

(c'est l'"itérateur" de gid nul)

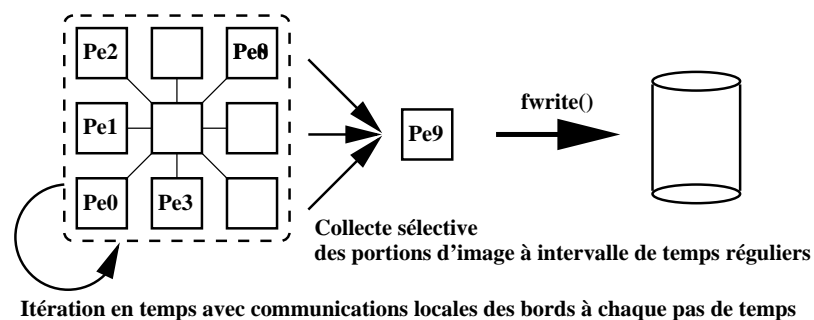
1. il diffuse les paramètres d'initialisation,
2. il collecte, somme et rediffuse les valeurs de  $Coef f_{Translation}$  (pvm\_reduce, pvm\_bcast),
3. il collecte périodiquement les attributs d'altitudes sur les sous-domaines (pvm\_gather) et les stocke sur disque.

▶ Rôle du "superviseur" en 2D :

Initialisation - modèle à 10 Pe



Itérations (date t) - modèle à 10 Pe

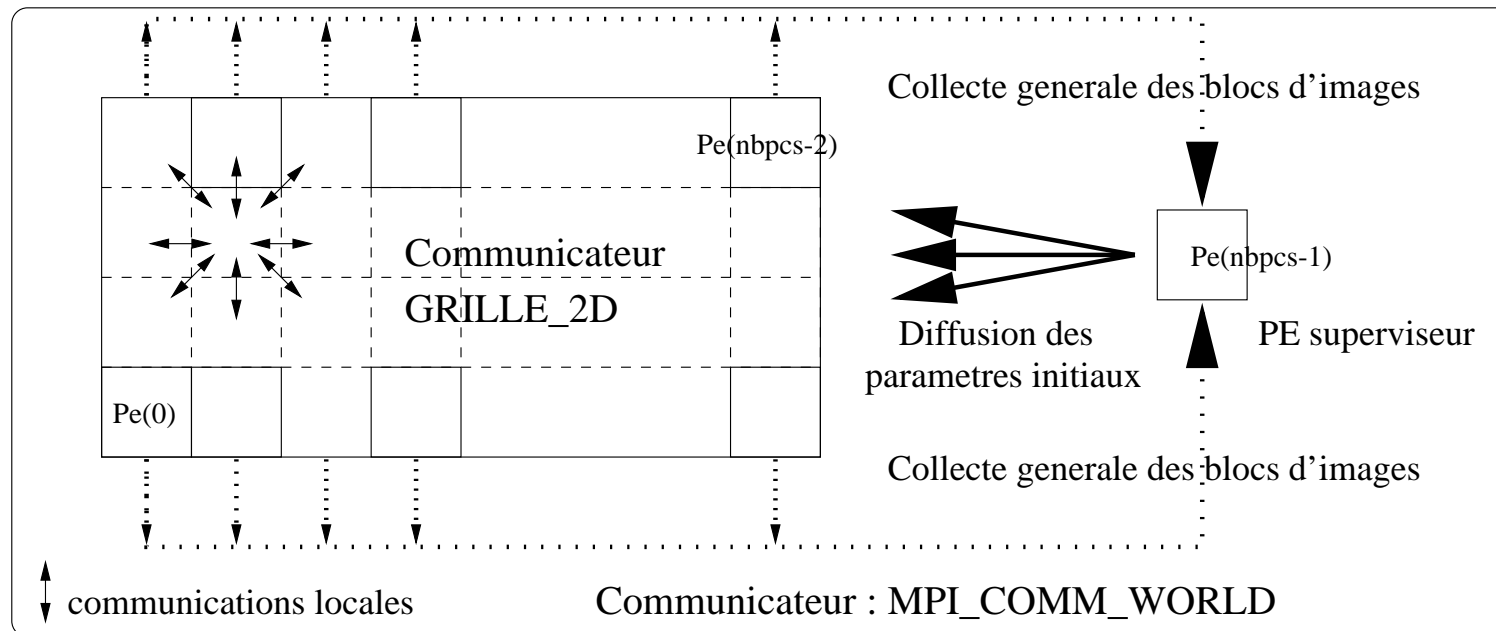


↳ **spécificités de la parallélisation 2D :**

de l'intérêt d'utiliser MPI et non plus PVM.

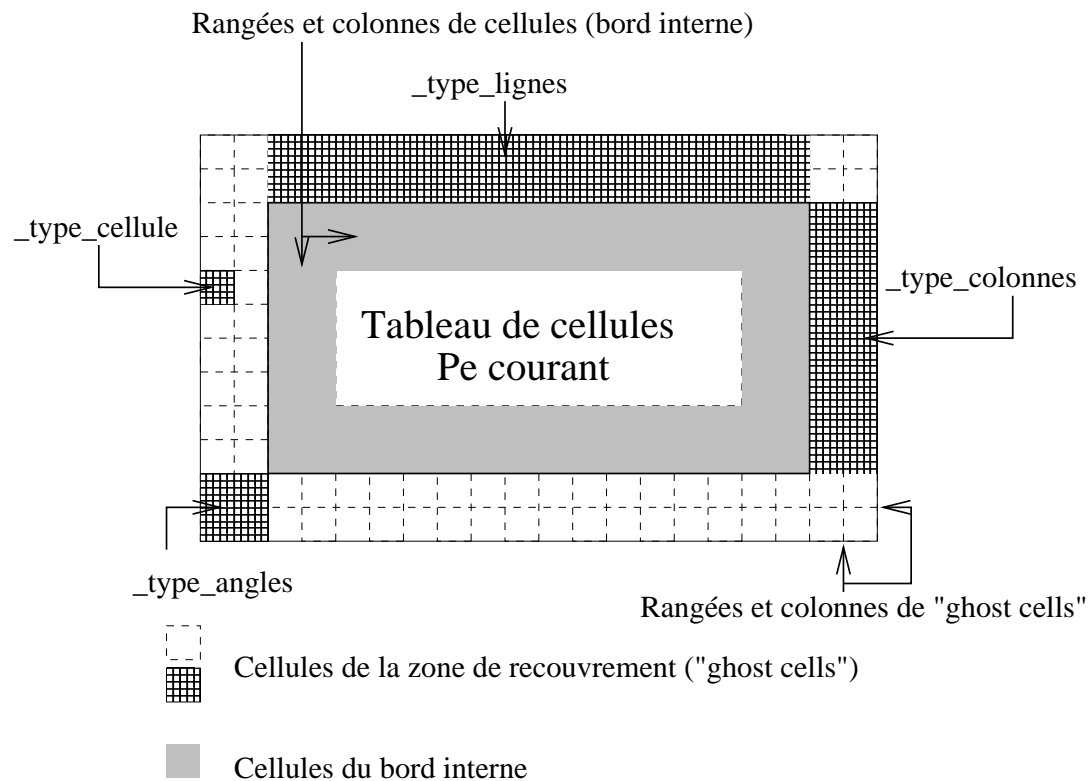
▶ topologie cartésienne de processus :

- partitionnement de MPI\_COMM\_WORLD en deux sous-communicateurs,
- les "itérateurs" sont disposés sur une topologie cartésienne virtuelle régulière à 2D (MPI\_Cart\_create())



► types de données dérivés :

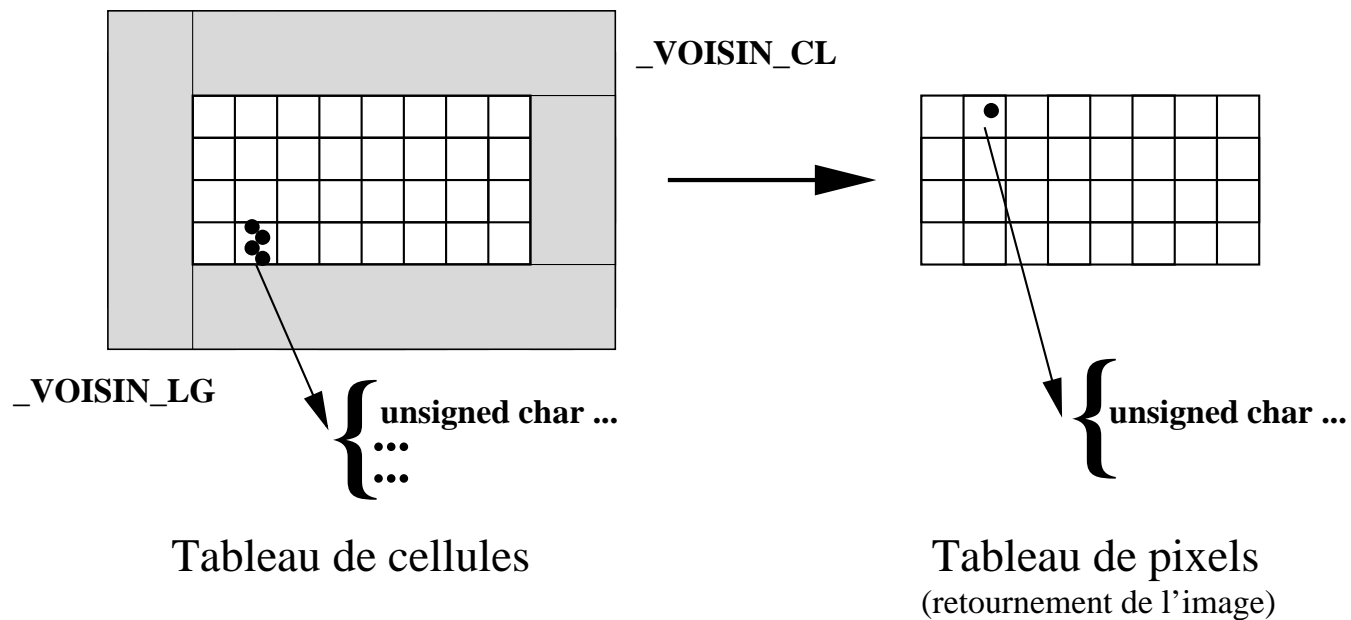
- le type `_type_cellule` (type hétérogène, `MPI_Type_struct()`),
- les types `_type_lignes`, `_type_angles`, `_type_colonnes` (types de données homogènes vectoriels à pas constant, `MPI_Type_vector()`),



émission des résultats sur les "itérateurs" : il suffit de récupérer la valeur du champ couleur de la configuration courante

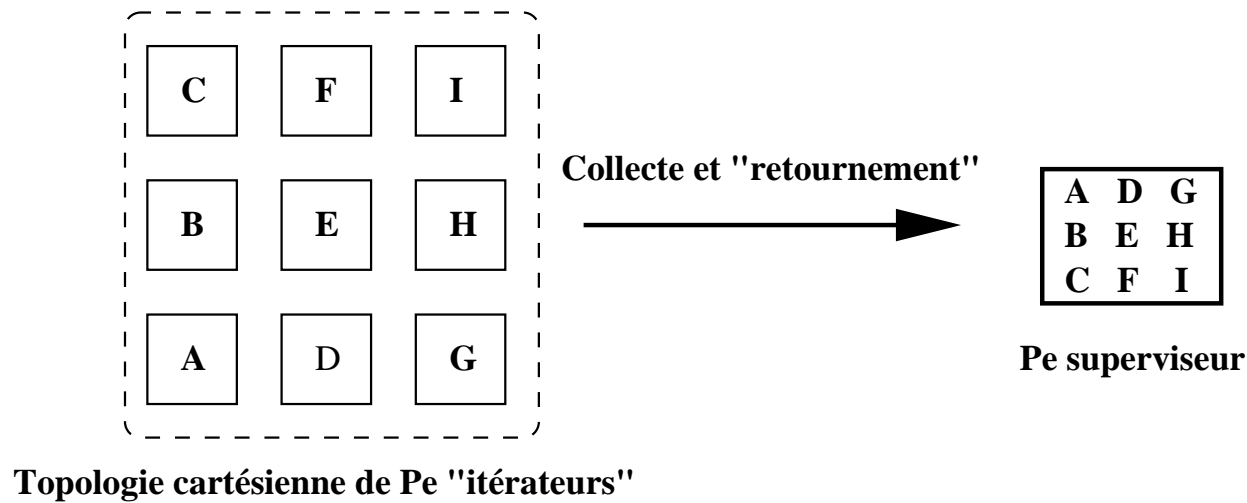
⇒ type `_type_bloc_image` (type homogène à pas variable où le tableau des cellules est considéré comme un tableau d'octets, `MPI_Type_indexed()` - tailles des champs alignés en mémoire obtenues par `MPI_Type_extent()`),

### Extraction et retournement des pixels d'un Pe quelconque



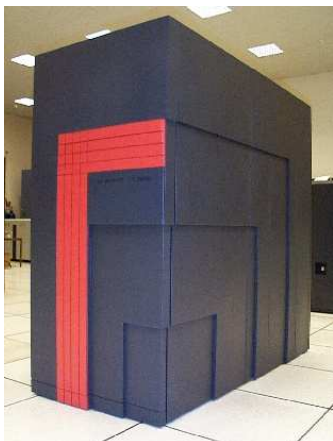
collecte des résultats sur le "superviseur" : `_type_image_globale` (type homogène à pas constant, primitive `MPI_Type_vector()`),

**Collecte et retournement des blocs d'image sur le superviseur**



## Résultats et performances

► le CRAY T3E :



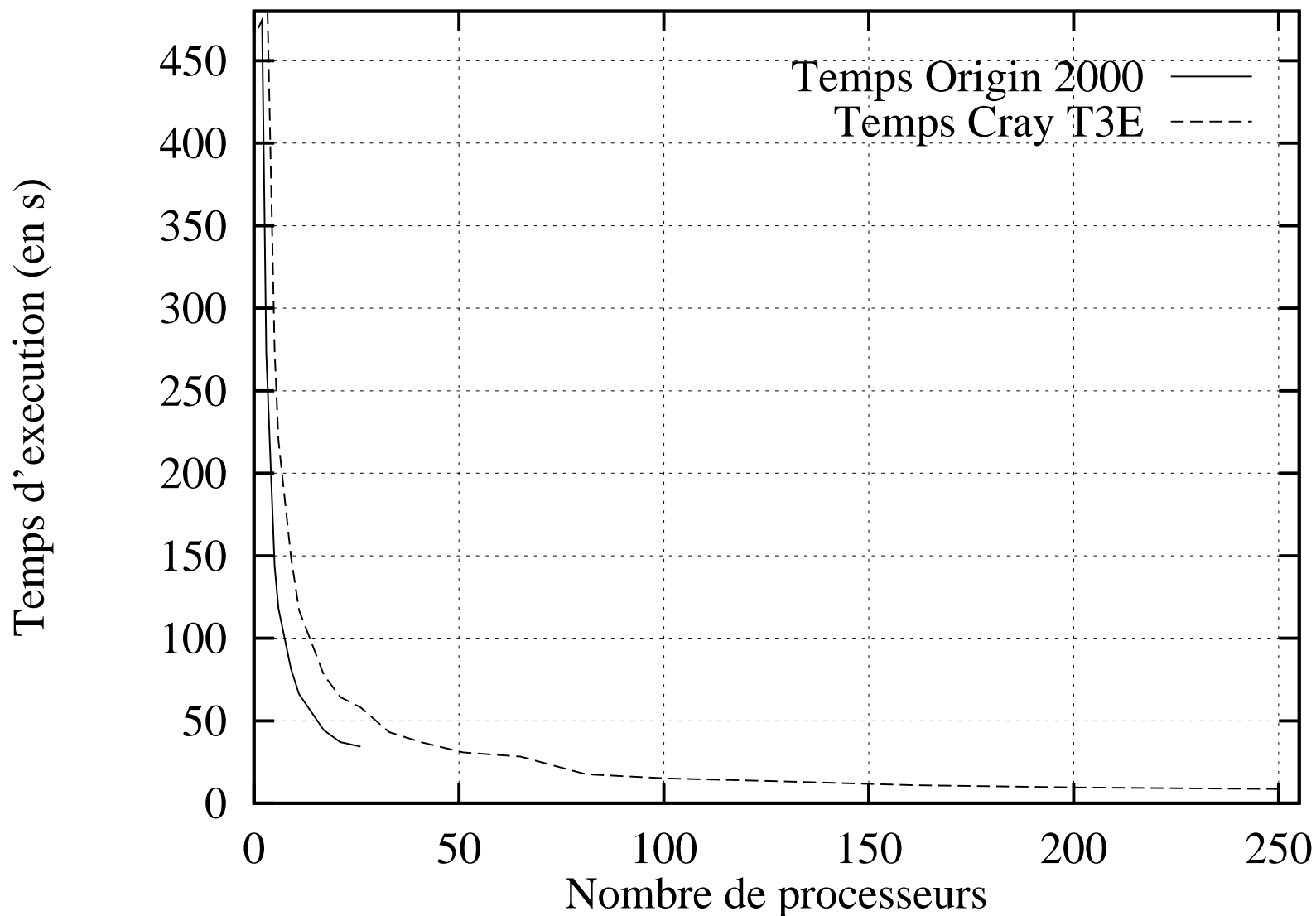
256 Dec Alpha 21164 - 600 Mflop/s,  
 caches de données de 8 ko et 96 ko,  
 latence : 1  $\mu$ s,  
 bande passante : 480 Mo/s,

► l'ORIGIN 2000 :



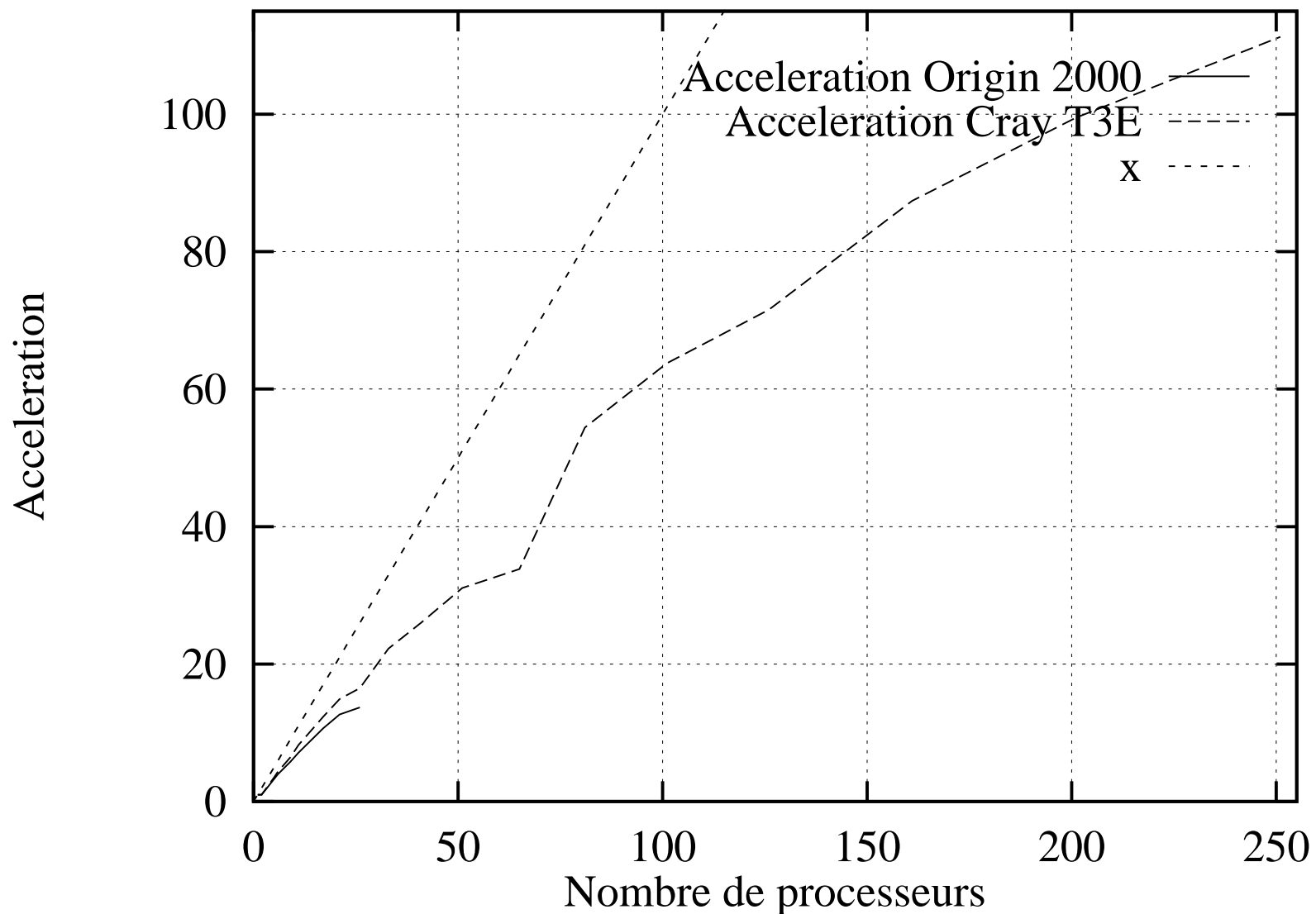
32 R10000 - 390 Mflop/s,  
 caches de données de 32 ko et 4 Mo,  
 latence : 773 ns,  
 bande passante : 780 Mo/s,

Temps d'execution du modele bi-dimensionnel

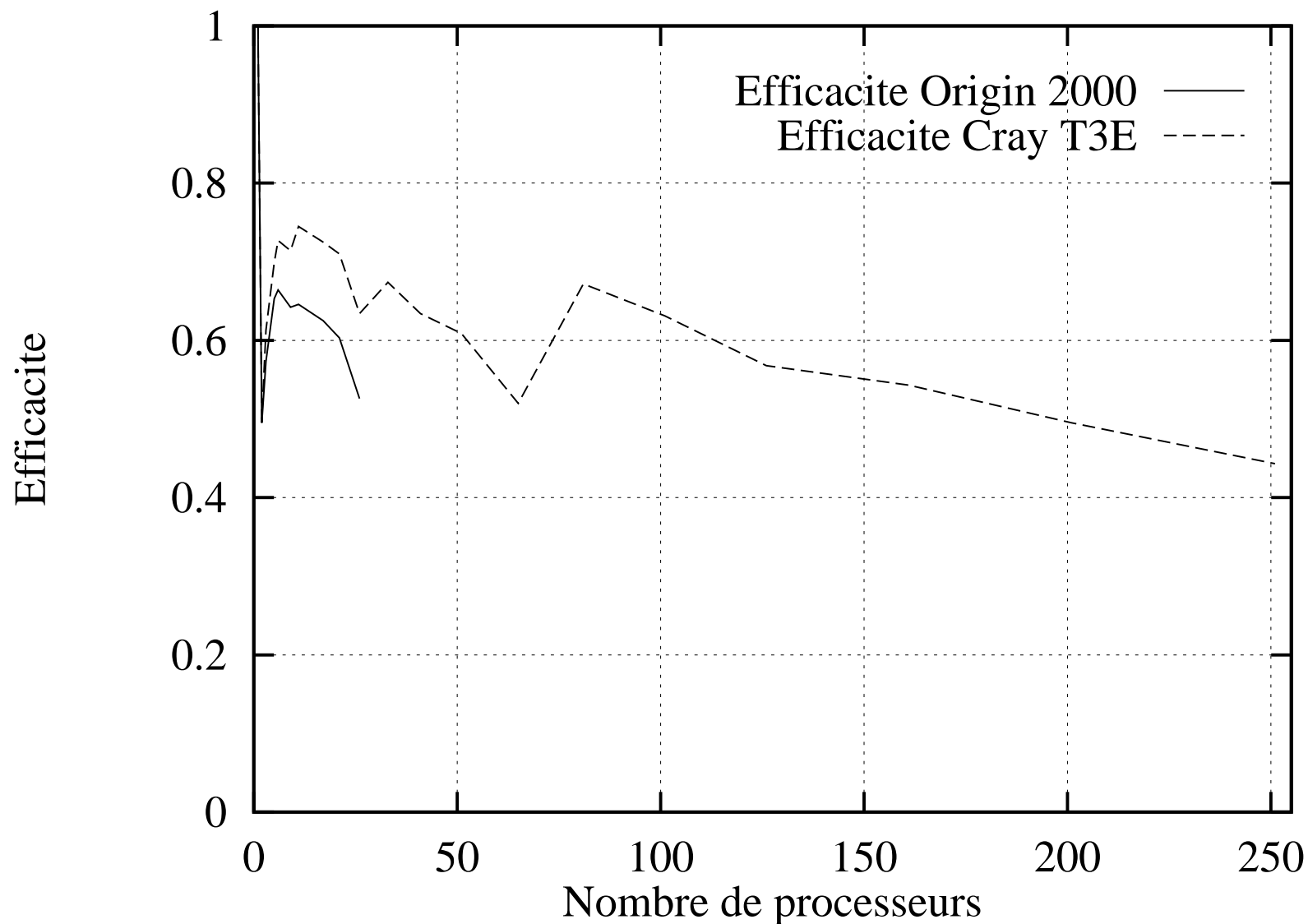




### Acceleration du modele bi-dimensionnel



Efficacite du modele bi-dimensionnel



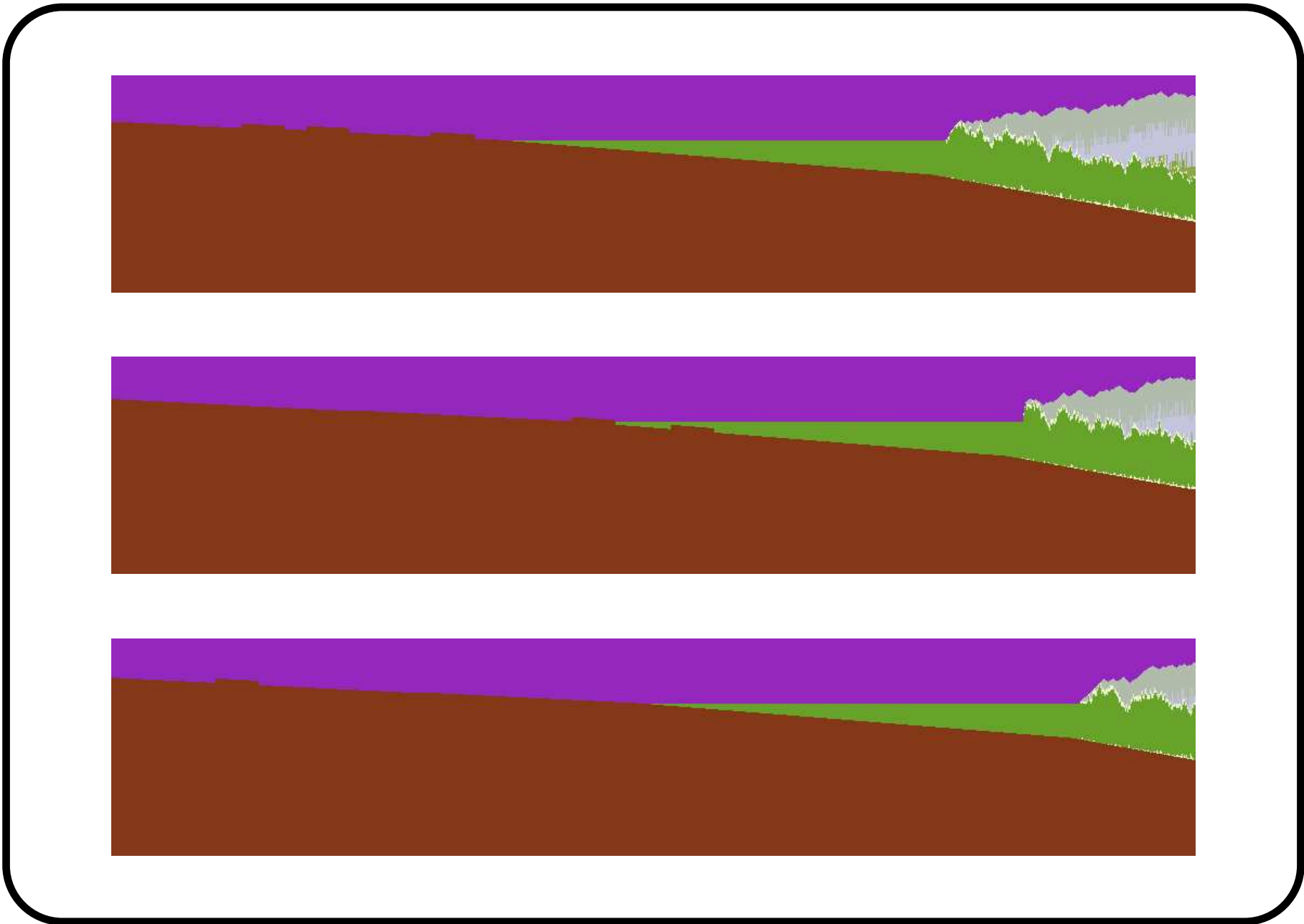
↳ **minimiser le ratio taille des interfaces / taille des sous-domaines :**

nb proc	Topologies associées (nb de processeurs en ligne * nb de processeurs en colonnes)											
	Ratios associés (rapport du nb de cellules aux interfaces sur le nb total de cellules d'un sous-domaine)											
1	1 * 1 <b>2,35</b>											
2	1 * 2 4,23	2 * 1 <b>2,74</b>										
4	1 * 4 7,78	2 * 2 4,61	4 * 1 <b>3,50</b>									
5	1 * 5 9,45	5 * 1 <b>3,88</b>										
8	1 * 8 14,14	2 * 4 8,14	4 * 2 5,36	8 * 1 <b>5,00</b>								
10	1*10 17,00	2 * 5 9,81	5 * 2 <b>5,73</b>	10*1 <b>5,73</b>								
16	2 * 8 14,48	4 * 4 8,87	8 * 2 <b>6,83</b>									
20	1*20 28,86	2*10 17,33	4 * 5 10,52	5 * 4 9,22	10*2 <b>7,54</b>	20*1 9,22						
25	1*25 33,60	5 * 5 <b>10,87</b>	25*1 <b>10,87</b>									
32	4 * 8 15,15	8 * 4 <b>10,28</b>										

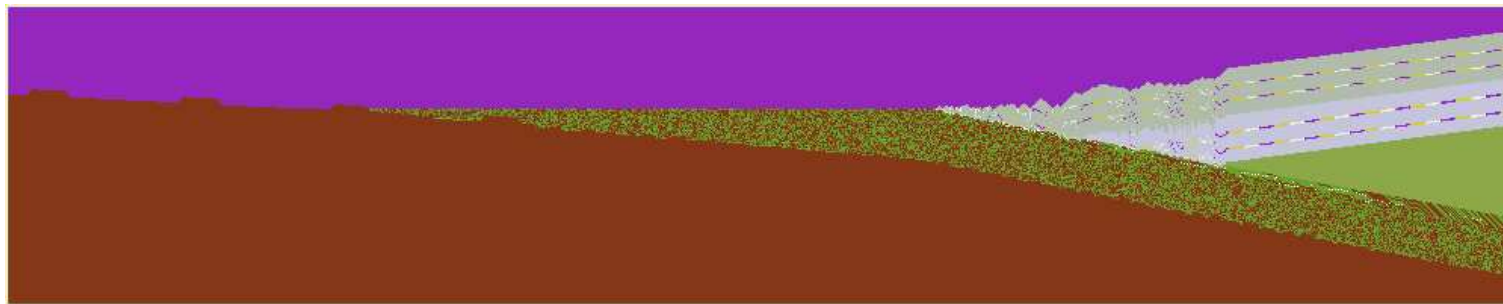
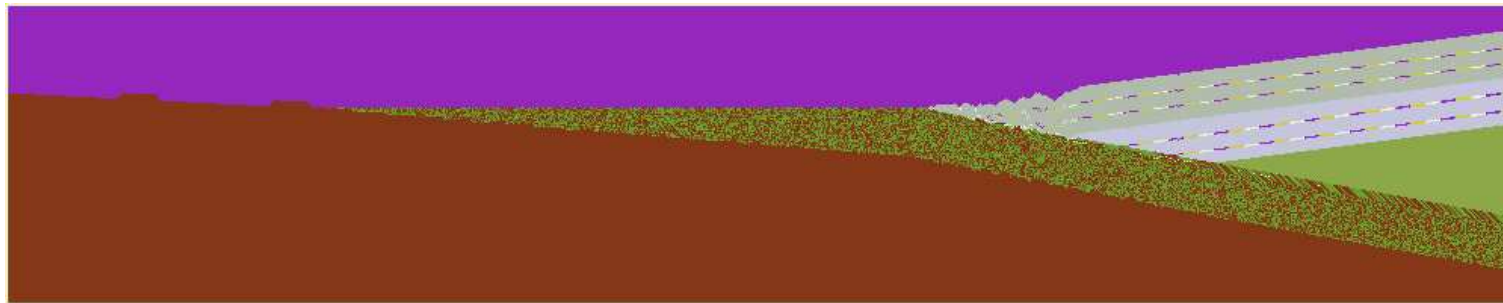
nb proc	Topologies associées (nb de processeurs en ligne * nb de processeurs en colonnes)												
	Ratios associés (rapport du nb de cellules aux interfaces sur le nb total de cellules d'un sous-domaine)												
40	1*40 44,67	2*20 29,14	4*10 17,98	5 * 8 15,48	8 * 5 11,91	10*4 <b>10,97</b>	20*2 <b>10,97</b>	40*1 15,48					
50	1*50 50,20	2*25 33,86	5*10 18,30	10*5 <b>12,59</b>	25*2 <b>12,59</b>	50*1 18,30							
64	8 * 8 <b>16,47</b>												
80	2*40 44,89	4*20 29,70	8*10 19,25	10*8 17,11	20*4 <b>14,27</b>	40*2 17,11							
100	1 * 100 66,80	2*50 50,40	4*25 34,38	5*20 29,97	10 * 10 19,87	20*5 <b>15,82</b>	25*4 <b>15,82</b>	50*2 19,87	100* 1 29,97				
125	5*25 34,64	25*5 <b>17,36</b>	125* 1 34,64										
160	4*40 45,32	8*20 30,79	20*8 <b>20,18</b>	40*4 <b>20,18</b>									
200	1 * 200 80,08	2 * 100 66,93	4*50 50,79	5*40 45,53	8*25 35,40	10 * 20 31,32	20 * 10 22,84	25*8 <b>21,63</b>	40*5 <b>21,63</b>	50*4 22,84	100* 2 31,32	200* 1 45,53	
250	5*50 50,98	10 * 25 35,90	25 * 10 <b>24,24</b>	50*5 <b>24,24</b>	125* 2 35,90	250* 1 50,98							

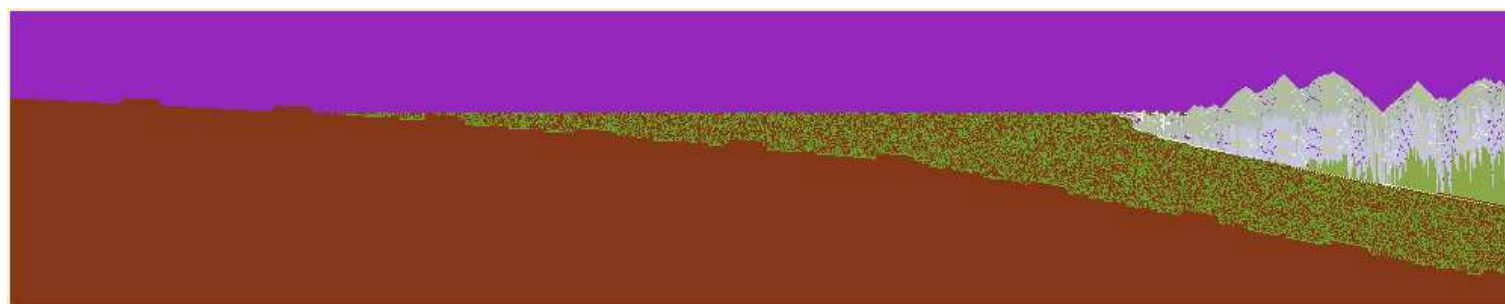
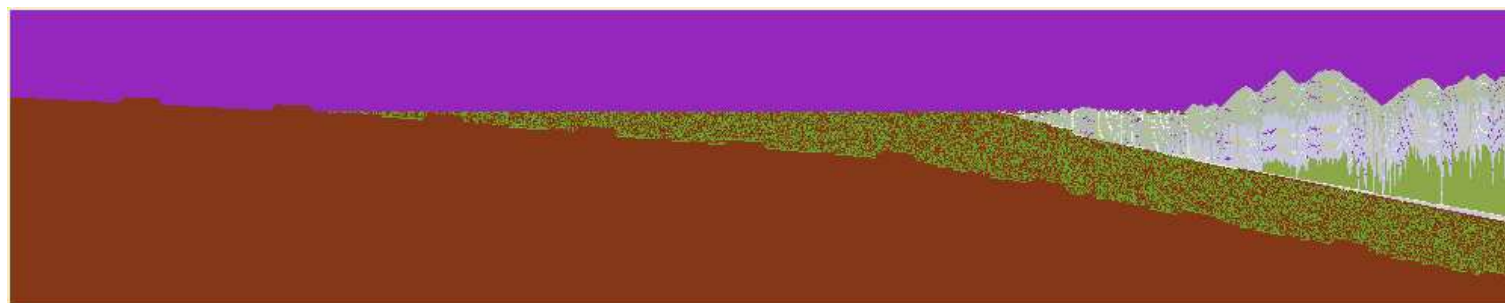
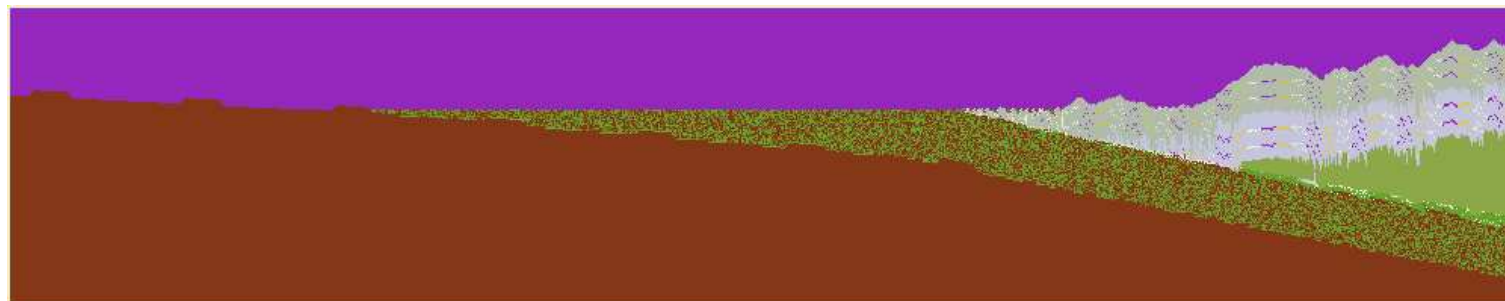
↳ Copies d'écran de la simulation 1D :





↳ Copies d'écran de la simulation 2D :







## Conclusions

- ↳ les vertus de l'approche discrète,
- ↳ la 1D intermédiaire facilitant le passage à la 2D,
- ↳ du bon "rendement" du modèle 1D en terme de rendu-visuel et de la simulation 2D en terme d'optimisation parallèle,
- ↳ justification de l'intérêt des nombreuses fonctionnalités de MPI,
- ↳ des simulations à valeur didactique pour l'illustration d'un phénomène géotectonique.

## Evolutions

- ↳ analyser le comportement des simulations,
- ↳ plate-forme logicielle spécifique  $\mathcal{LAC}$ , envisager la 3D...