

Introduction

Passage de la notion de modèle à la notion de système de modélisation i.e. différents composants en interaction.

Composant :

domaine physique (géographique)
une physique (équations) + méthodes numériques
un maillage

Quelques cas particuliers :

Faire fonctionner un seul composant (modèle homogène)

Domaines \neq
 $\psi_s - \nu =$
maillage $=$ \rightarrow Déc. de domaine
"classique"

Raffinement de maillage

Domaines \neq ou emboîtés \rightarrow raffinement
 $\psi_s - \nu =$ \rightarrow Déc. domaine
maillage à \neq résolutions
• sans recouvrement
• avec recouvrement total

Raccord de modèles existants

Domaines \neq \rightarrow Déc. de domaine
 $\psi_s - \nu =$
maillages \neq "Mortar"

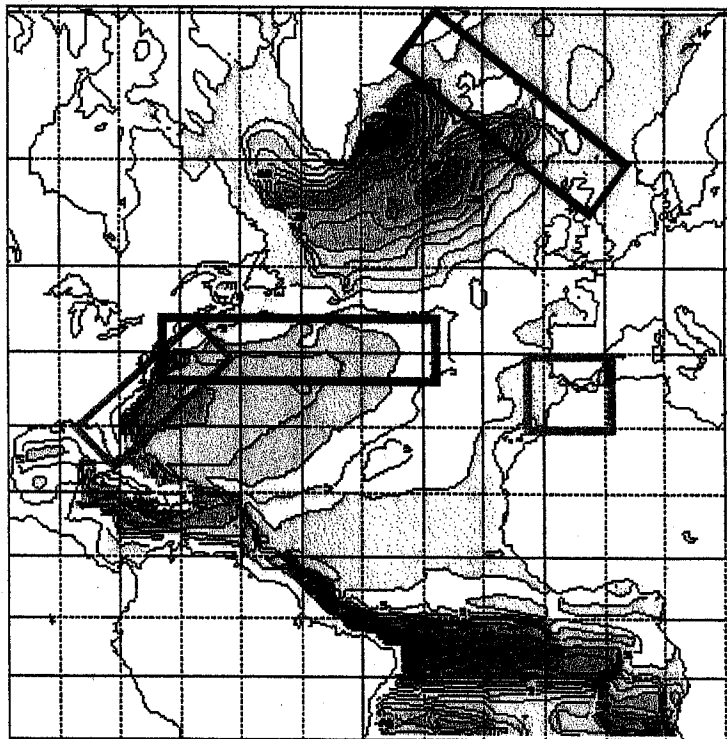
Faire fonctionner ensemble plusieurs composants

Domaines \neq \rightarrow Couplage de
 $\psi_s - \nu \neq$ modèles
maillages \neq

Ocean model = a modelling system, i.e. different components in interaction

component :

**geographical domain
physics
numerics
resolution**



Problem : make these components work together

Physical aspects *it must make sense (cf oceanographers)*

Mathematical aspects *consistency between the solutions of the different components*

Numerical aspects *efficiency*

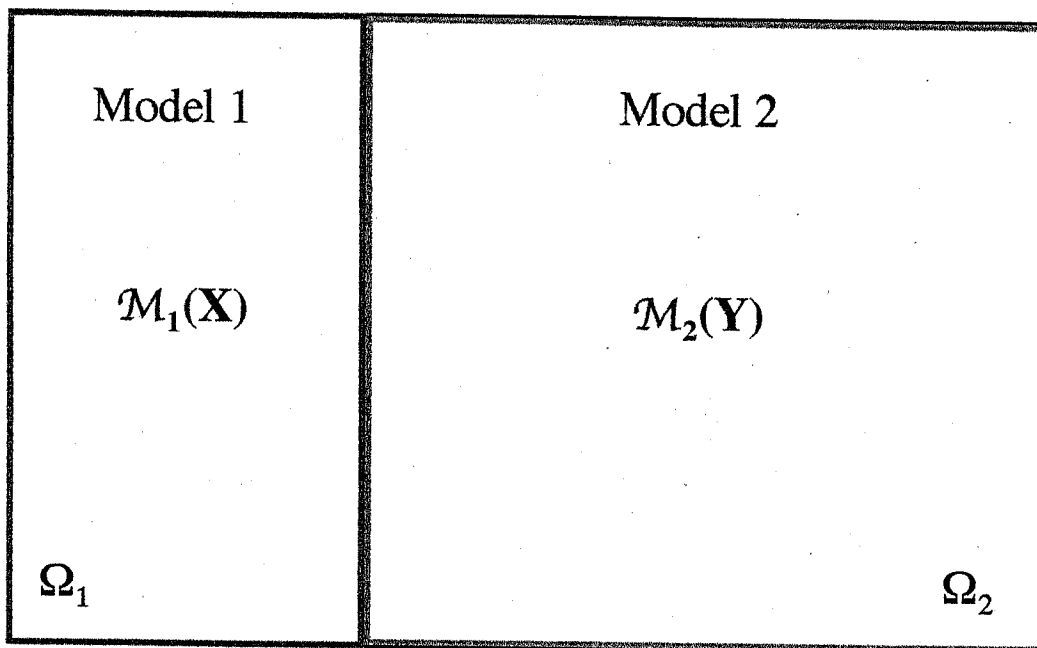
Programming aspects *« black box » approach
parallel implementation*

**Step 1 : mesh refinement (same physics and numerics –
L. Debreu, PhD thesis)**

Step 2 : model nesting and coupling (starting project)

Step 2 : model nesting and coupling

Key point : consistency between model solutions



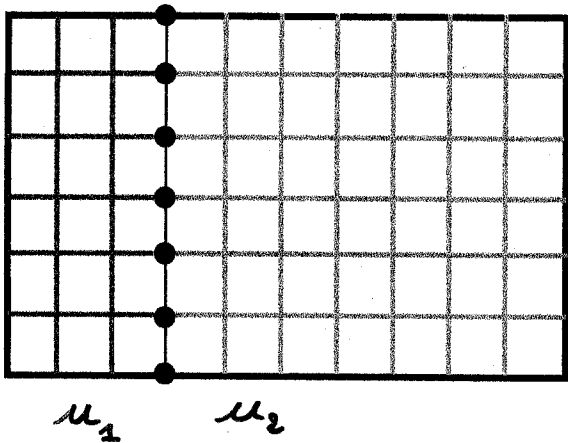
Defining equivalent variables

Defining consistency criteria

Making the solutions verify these criteria

Model Coupling – 1. Defining equivalent variables

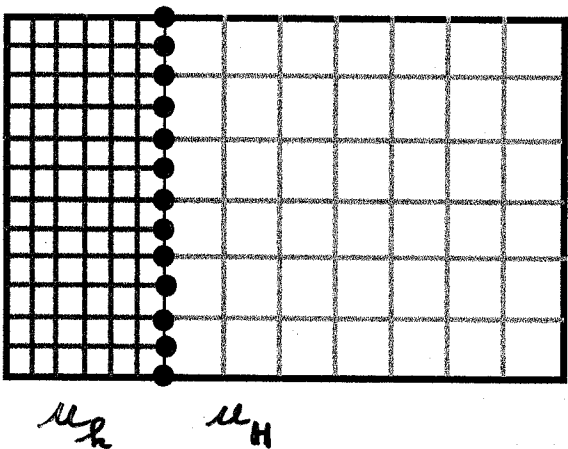
Pb : find mappings T_1 and T_2 : $u_1 = T_1(X)$ $u_2 = T_2(Y)$
 such that u_1 and u_2 are consistent



Same physics, same resolutions

→ classical DD problem

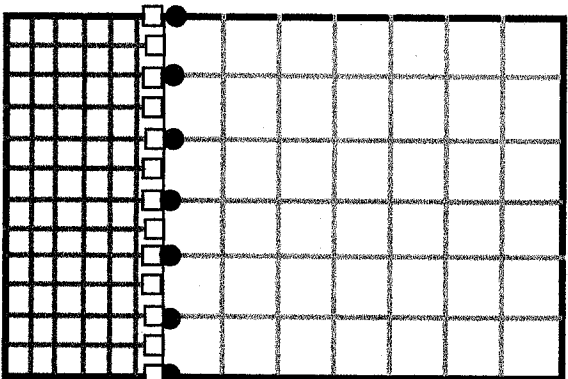
$$T_1 = T_2 = \text{Id}$$



Same physics, different resolutions

→ mesh refinement problem

T_1, T_2 : interpolation operators



Different physics, different resolutions

→ coupling problem

T_1, T_2 : depending on the physics + interpolation operators

$$X = \begin{pmatrix} u \\ v \\ h \end{pmatrix} \quad Y = (\psi) \quad \rightarrow \quad \begin{aligned} u &= -\frac{\partial \psi}{\partial y} \\ v &= \frac{\partial \psi}{\partial x} \\ h &= \int_0^y \psi \end{aligned}$$

Model Coupling – 2. Defining consistency criteria

Pb : In which sense must u_1 and u_2 be consistent ?

Mesh refinement – Berger & Oliger algorithm	u_H is updated using u_h (copy, averaging,...) + flux conservation through interfaces <i>at the end of each coarse timestep</i>
« classical » ocean- atmosphere coupling	$u_1(t_n)$ is used as a BC to compute $u_2(t_{n+1})$ $u_2(t_n)$ is used as a BC to compute $u_1(t_{n+1})$ <i>every few ocean timesteps</i>

—————→ **« Algorithmic » consistency**

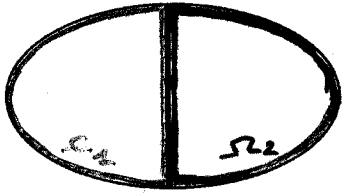
DD methods	$u_1 = u_2$ in a weak sense (mortar method) $u_1 = u_2$ and $\partial u_1 / \partial n = \partial u_2 / \partial n$ in a strong sense (e.g. Schwarz methods) <i>at each timestep</i>
------------	---

—————→ **mathematical consistency**

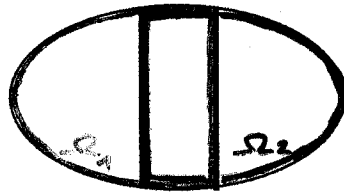
→ address such criteria in the context of model coupling.

Model Coupling – 3. Make the solutions verify the criteria

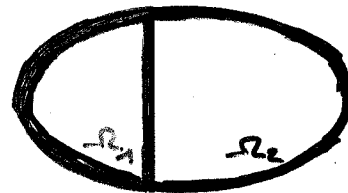
non-overlapping



partially overlapping



totally overlapping



Iterative methods :

$$\begin{cases} L_1 u_1^{n+1} = f_1 & \text{in } \Omega_1 \\ B_1(u_1^{n+1}) = B_1(u_2^n) & \text{on } \Gamma_{12} \end{cases} \quad \begin{cases} L_2 u_2^{n+1} = f_2 & \text{in } \Omega_2 \\ B_2(u_2^{n+1}) = B_2(u_1^n) & \text{on } \Gamma_{21} \end{cases}$$

Overlapping Schwarz : $B_1 = B_2 = \text{Id}$

Non-overlapping Schwarz : $B_i = c_i + \partial/\partial n_i$

Mortar elements : B_i depends on the iterative solver

Pb : cost of each iteration !!

Issue : Find efficient BCs, that ensure a rapid convergence

NUMERICAL POINT OF VIEW :

- better conditionning
- allows local mesh refinement
- decompose a domain of complex shape into several simple sub-domains

PROGRAMMING POINT OF VIEW :

- data are naturally distributed on the processors
- most of the work can be done independently on the different processors
- efficient sequential algorithms may be used on each subdomain

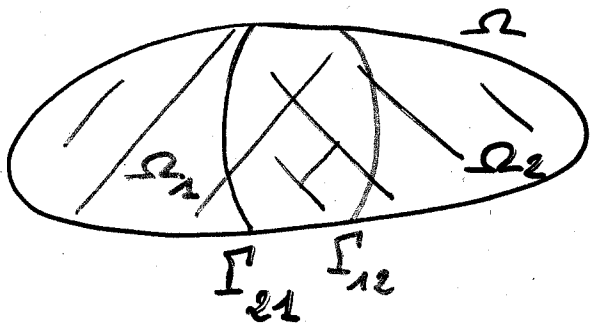


⇒ Managing the connections between neighbours and eventually the overlapping areas



⇒ Which partitionning ? Which domain decomposition method ?

Les méthodes de Schwarz avec recouvrement



$$\begin{cases} Lu = f & \text{dans } \Omega \\ u = g & \text{sur } \partial\Omega \end{cases}$$

Schwarz additif

* Initialisation: u_1^0 donné sur Ω_1 , u_2^0 donné sur Ω_2

* Itération:

Tant que $\|u_1^m - u_2^m\|_{\Gamma_{12} \cup \Gamma_{21}} > \epsilon$

résoudre:

$$\begin{cases} Lu_1^m = f & \text{dans } \Omega_1 \\ u_1^m = g & \text{sur } \partial\Omega_1 \cap \partial\Omega \\ u_1^m = u_2^{m-1} & \text{sur } \Gamma_{12} \end{cases} \quad \text{et} \quad \begin{cases} Lu_2^m = f & \text{dans } \Omega_2 \\ u_2^m = g & \text{sur } \partial\Omega_2 \cap \partial\Omega \\ u_2^m = u_1^{m-1} & \text{sur } \Gamma_{21} \end{cases}$$

- Le recouvrement est essentiel
- Parallélisation évidente (comm: u_j^{n-1} sur Γ_{ij} , \Leftrightarrow de charge)
- Extension à N sous-domaines immédiate
- "additif": On note: T_i opérateur de projection sur Ω_i
 L_i restriction de L à Ω_i
alors $u^{n+1} - u = \left(\text{Id} - \sum_{i=1}^2 T_i \circ L_i^{-1} \circ T_i \circ L \right) (u^n - u)$
- convergence
- régularité de la solution, du fait du recouvrement
- surcoût de calcul dû au recouvrement.

Schwarz multiplicatif

* Initialisation: u_2^0 donné sur Ω_2

* Itération:

Tant que $\|u_1^n - u_2^n\|_{\Gamma_{12} \cup \Gamma_{21}} > \varepsilon$

résoudre:

$$\begin{cases} L u_1^m = f & \text{dans } \Omega_1 \\ u_1^m = g & \text{sur } \partial\Omega_1 \cap \partial\Omega \\ u_1^m = u_2^{n-1} & \text{sur } \Gamma_{12} \end{cases}$$

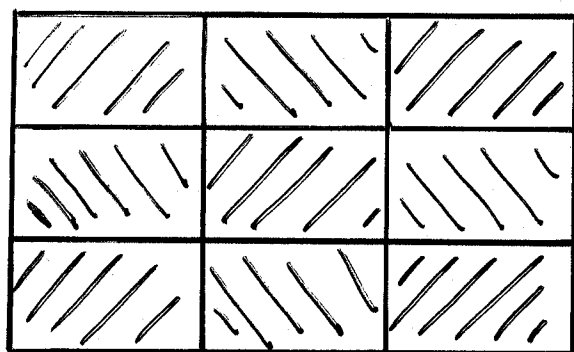
Puis

$$\begin{cases} L u_2^m = f & \text{dans } \Omega_2 \\ u_2^m = g & \text{sur } \partial\Omega_2 \cap \partial\Omega \\ u_2^m = u_1^m & \text{sur } \Gamma_{21} \end{cases}$$

→ Convergence + rapide (analogie Jacobi / Gauss-Seidel)

→ Parallélisation: moins naturelle

numérotation rouge-noir (P pros, 2P ss-domaines)



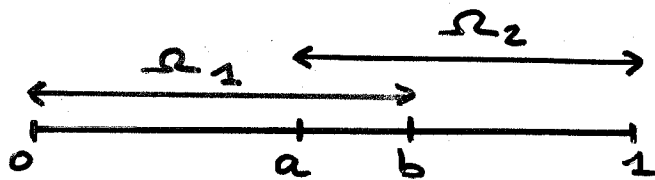
→ "Multiplicatif":

$$u^{n+1} - u = \prod_{i=1}^2 (\text{Id} - T_i^t L_i^{-1} T_i L) (u^n - u)$$

Exemple 1-D

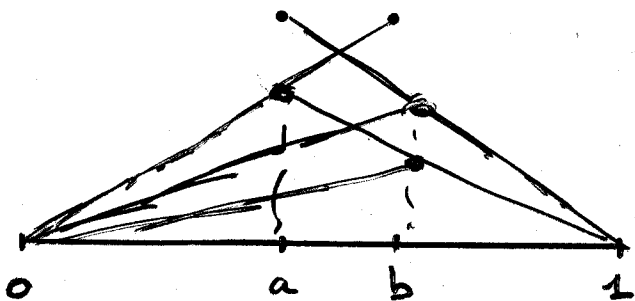
$$\begin{cases} u''(x) = 0 & \text{sur }]0, 1[\\ u(0) = u(1) = 0 \end{cases} \rightarrow \text{sol. exacte : } u(x) = 0$$

$$[0, 1] = [0, b] \cup [a, 1], \quad a < b$$

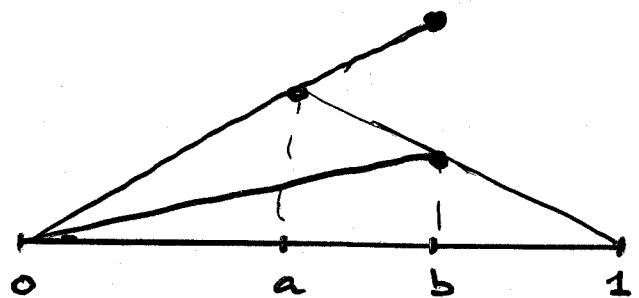


$$u^0(x) = 1$$

Schwarz additif



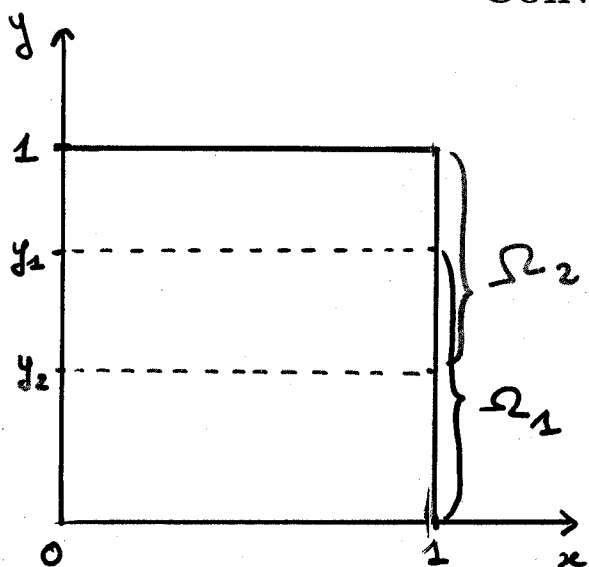
Schwarz multiplicatif



→ Plus le recouvrement est grand, plus la convergence est rapide (en nb d'itérations), mais plus le coût de chaque itération est important.

ici : convergence en $\left(\frac{a}{b} \frac{1-b}{1-a}\right)^m$

Convergence properties



u^* solution of the initial problem

$e_1^n = u_1^n - u^*$ error on Ω_1 at iteration n

$e_2^n = u_2^n - u^*$ error on Ω_2 at iteration n

$$\left\{ \begin{array}{l} -\Delta e_1^n = 0 \quad \text{in } \overset{\circ}{\Omega}_1 \\ e_1^n = 0 \quad \text{on } \partial\Omega_1 \cap \partial\Omega \\ e_1^n = e_2^{n-1} \quad \text{on } \gamma_{12} \end{array} \right. \quad \text{and} \quad \left\{ \begin{array}{l} -\Delta e_2^n = 0 \quad \text{in } \overset{\circ}{\Omega}_2 \\ e_2^n = 0 \quad \text{on } \partial\Omega_2 \cap \partial\Omega \\ e_2^n = e_1^{n-1} \quad \text{on } \gamma_{21} \end{array} \right.$$

$$e_1^n(x, y) = \sum_{k=1}^{\infty} a_k^n \sin k\pi x \frac{\sinh k\pi y}{\sinh k\pi y_1}$$

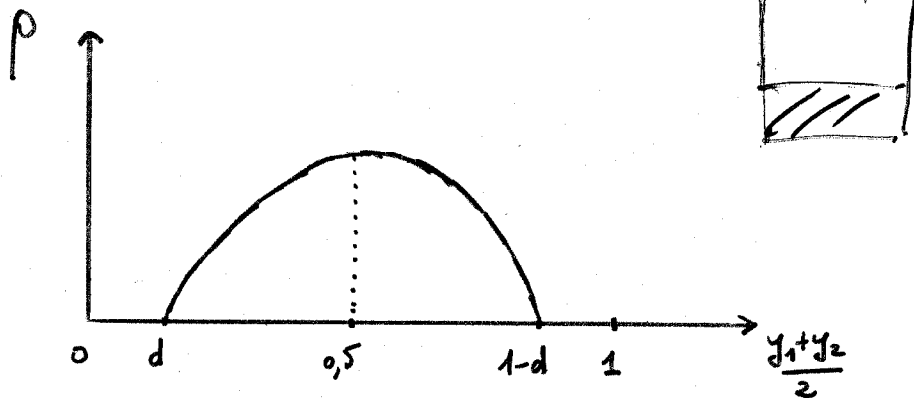
$$\text{and} \quad e_2^n(x, y) = \sum_{k=1}^{\infty} b_k^n \sin k\pi x \frac{\sinh k\pi(1-y)}{\sinh k\pi(1-y_2)}$$

Convergence proof :

$$|e_1^n(x, y)| \leq \rho^{n/2} \sum_{k=1}^{\infty} |a_k^0|$$

$$\text{with } \rho = \frac{\sinh \pi y_2 \sinh \pi(1-y_1)}{\sinh \pi y_1 \sinh \pi(1-y_2)} \quad \parallel \quad \underline{(0 < \rho < 1)}$$

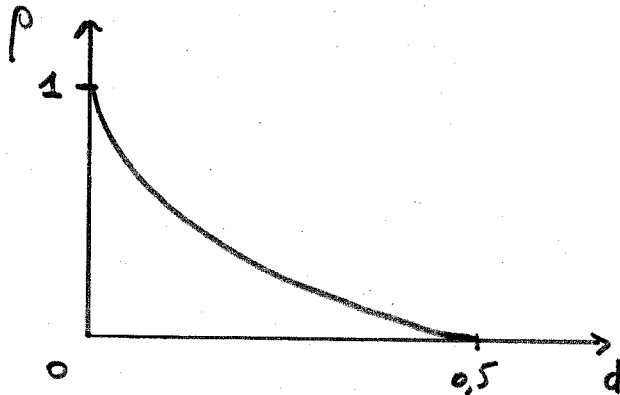
PARTITIONNING : the overlapping length $d = (y_1 - y_2)/2$ is supposed constant



\Rightarrow The convergence rate is minimum for domains of equal sizes !!
But load-balancing considerations impose this choice.

OVERLAPPING LENGTH : $y_1 = 0.5 + d$ and $y_2 = 0.5 - d$ yield

$$\rho(d) = \left(\frac{\sinh \pi(0.5 - d)}{\sinh \pi(0.5 + d)} \right)^2$$



- The greater the overlapping length is, the faster convergence is achieved. BUT the amount of computations at each iteration increases. \rightarrow compromise...
- The concavity of the curve is important.

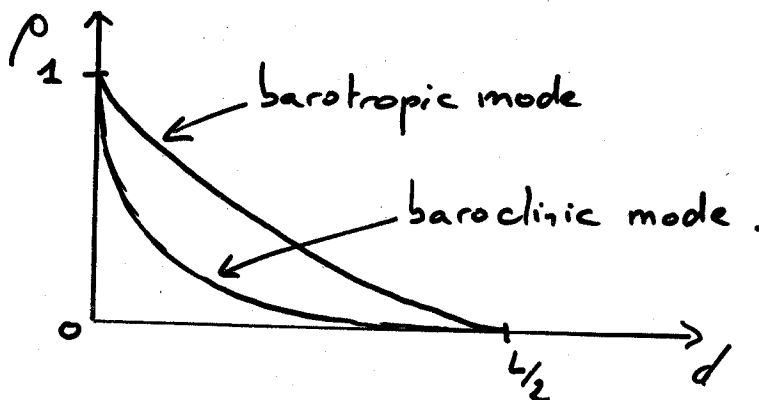
Application to QG equations

AT EACH TIME STEP:

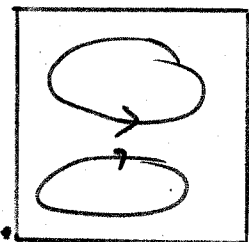
$$\begin{cases} -\Delta\Phi = f & \text{for barotropic mode} \\ -\Delta\Phi + \frac{1}{R_d^2}\Phi = f & \text{for baroclinic modes} \end{cases} \quad \text{on } [0, L] \times [0, L]$$

CONVERGENCE RATE:

$$\rho = \left(\frac{\sinh \beta(L - d/2)}{\sinh(\beta L + d/2)} \right)^2 \quad \text{with } \beta = \sqrt{\frac{1}{R_d^2} + \frac{\pi^2}{L^2}}$$



Example:



$$L \times L = 4000 \text{ km} \times 4000 \text{ km}.$$

Barotropic mode: $d_{opt} = 450 \text{ km} \rightarrow \rho = 0.4$

Baroclinic modes: $R_d = 10 - 50 \text{ km}$

$d_{opt} = 50 - 150 \text{ km} \rightarrow \rho = 10^{-12} \text{ to } 10^{-15}$

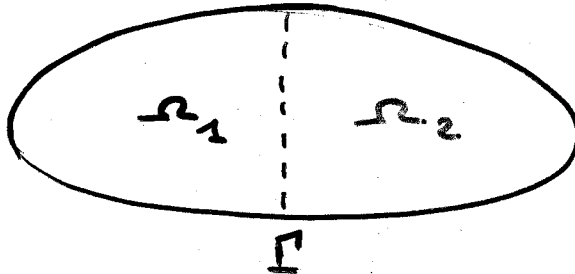
Méthodes de Schwarz sans recouvrement

→ Éviter le surcoût dû au recouvrement

→ Difficulté de recouvrement pour des géométries complexes



Idée : permettre aux cond. aux limites d'évoluer sur Γ , bien qu'il n'y ait pas de recouvrement.



$$\text{On veut : } \begin{cases} Lu_1 = f \text{ dans } \Omega_1 \\ u_1 = g \text{ sur } \partial\Omega_1 \cap \partial\Omega \end{cases} \quad \begin{cases} Lu_2 = f \text{ dans } \Omega_2 \\ u_2 = g \text{ sur } \partial\Omega_2 \cap \partial\Omega \end{cases}$$

$$\text{et } \begin{cases} u_1 = u_2 \text{ sur } \Gamma \\ \frac{\partial u_1}{\partial n} = \frac{\partial u_2}{\partial n} \text{ sur } \Gamma \end{cases}$$

Méthode Dirichlet-Neumann (Björstad-Widlund 86; Marini-Quarteroni 89)

$$\begin{cases} L u_1^{m+1} = f \text{ dans } \Omega_1 \\ u_1^{m+1} = g \text{ sur } \partial\Omega_1 \cap \partial\Omega \\ u_1^{m+1} = \theta u_2^m + (1-\theta) u_1^m \parallel \text{sur } \Gamma \\ \theta \in]0, 1[\end{cases} \quad \begin{cases} L u_2^{n+1} = f \text{ dans } \Omega_2 \\ u_2^{n+1} = g \text{ sur } \partial\Omega_2 \cap \partial\Omega \\ \frac{\partial u_2^{n+1}}{\partial n} = \frac{\partial u_1^m}{\partial n} \parallel \text{sur } \Gamma \end{cases}$$

Méthode mixte (Lions, 89)

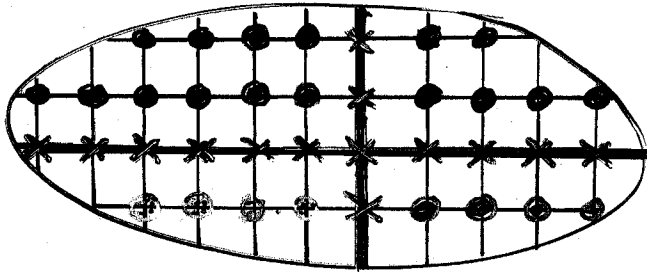
$$\begin{cases} L u_1^{m+1} = f \text{ dans } \Omega_1 \\ u_1^{m+1} = g \text{ sur } \partial\Omega_1 \cap \partial\Omega \\ \frac{\partial u_1^{m+1}}{\partial n_1} + \lambda u_1^{m+1} = \frac{\partial u_2^m}{\partial n_1} + \lambda u_2^m \parallel \Gamma \end{cases} \quad \begin{cases} L u_2^{n+1} = f \text{ dans } \Omega_2 \\ u_2^{n+1} = g \text{ sur } \partial\Omega_2 \cap \partial\Omega \\ \frac{\partial u_2^{n+1}}{\partial n_2} + \lambda u_2^{n+1} = \frac{\partial u_1^m}{\partial n_2} + \lambda u_1^m \parallel \Gamma \end{cases}$$

→ Même volume de comm. qu'avec recouvrement, moins de calculs par itération

→ Choix de θ, λ ??

Méthode de Schur

$$\begin{cases} Lu = f & \text{dans } \Omega \\ u = g & \text{sur } \partial\Omega \end{cases} \rightarrow Ax = b$$



n ss-domaines

Inconnues $x^{(i)}$ sur le ss-domaine Ω_i

$$x^{(i)} = \begin{pmatrix} x_{\text{I}}^{(i)} \\ x_{\text{F}}^{(i)} \end{pmatrix} \begin{array}{l} \leftarrow \text{les pts intérieurs à } \Omega_i \\ \leftarrow \text{les pts d'interface de } \Omega_i \end{array}$$

Le système $Ax = b$ peut donc être ré-écrit :

$$\begin{bmatrix} A_{\text{II}}^{(1)} & & & & A_{\text{IF}}^{(1)} \\ & \circ & & & \vdots \\ & & & & A_{\text{IF}}^{(m)} \\ & & & & \vdots \\ A_{\text{FI}}^{(1)} & \dots & A_{\text{FI}}^{(m)} & & A_{\text{FF}} \end{bmatrix} \begin{bmatrix} x_{\text{I}}^{(1)} \\ \vdots \\ x_{\text{I}}^{(m)} \\ x_{\text{F}} \end{bmatrix} = \begin{bmatrix} b_{\text{I}}^{(1)} \\ \vdots \\ b_{\text{I}}^{(m)} \\ b_{\text{F}} \end{bmatrix} \quad \text{Structure "flèche"}$$

Le système est donc :

$$\begin{cases} A_{\text{II}}^{(i)} x_{\text{I}}^{(i)} + A_{\text{IF}}^{(i)} x_{\text{F}} = b_{\text{I}}^{(i)} & i=1 \dots m \\ \sum_{i=1}^m A_{\text{FI}}^{(i)} x_{\text{I}}^{(i)} + A_{\text{FF}} x_{\text{F}} = b_{\text{F}} \end{cases}$$

En éliminant les $x_I^{(i)}$, $i=1..n$, la dernière équation

devient:

$$\underbrace{\left(A_{FF} - \sum_{i=1}^m A_{FI}^{(i)} A_{II}^{(i)-1} A_{IF}^{(i)} \right)}_S x_F = \underbrace{b_F - \sum_{i=1}^m A_{FI}^{(i)} A_{II}^{(i)-1} b_I^{(i)}}_{b'_F}$$

$S = \text{Complément de Schur}$

1. On résout $S x_F = b'_F$

2. On résout $A_{II}^{(i)} x_I^{(i)} = b_I^{(i)} - A_{IF}^{(i)} x_F$ $i=1..n$ en //

Pb: résolution de $S x_F = b'_F$

→ Ne pas calculer S explicitement, mais utiliser seulement le produit $S \times$ vecteur. (gradient conjugué, GPCRES, ...)

$$S v = A_{FF} v - \sum_{i=1}^m A_{FI}^{(i)} A_{II}^{(i)-1} A_{IF}^{(i)} v$$

$A_{IF}^{(i)} v$: produit matrice vecteur local en //

$A_{II}^{(i)-1} A_{IF}^{(i)} v$: résolution locale de $A_{II}^{(i)} y = A_{IF}^{(i)} v$ en //

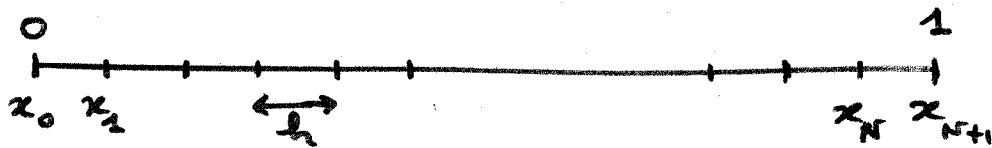
$A_{FI}^{(i)} A_{II}^{(i)-1} A_{IF}^{(i)} v$: produit matrice vecteur local en //

$A_{FF} v = \sum_{i=1}^m A_{FF}^{(i)} v$: produits locaux en // puis assemblage

→ Préconditionnement.

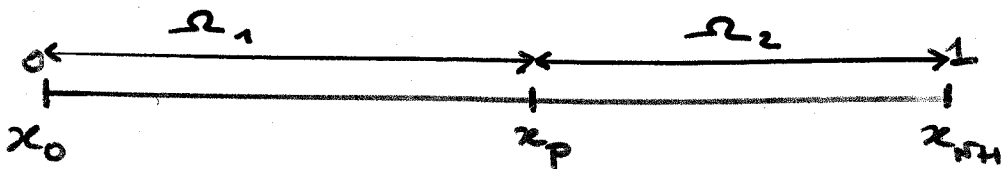
Exemple 1-D

$$\begin{cases} -u''(x) = f & \text{sur }]0, 1[\\ u(0) = u(1) = 0 \end{cases}$$



$$u''(x_i) \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}$$

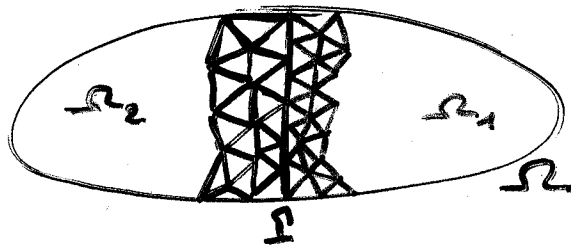
$$\frac{1}{h^2} \begin{bmatrix} 2 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & -1 & 2 & -1 \\ 0 & & & -1 & 2 \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_r \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_r \end{bmatrix}$$



$$\begin{bmatrix} 2 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & -1 & 2 & -1 \\ 0 & & & -1 & 2 \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_{p-1} \\ u_{p+1} \\ \vdots \\ u_r \\ u_p \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_{p-1} \\ f_{p+1} \\ \vdots \\ f_r \\ f_p \end{bmatrix}$$

Mortar elements - Eléments-joints
 (méthode de Schur duale) Bernardi-Maday-Patera
 1989

Idée: raccord "faible" dans le cas de maillages non-coïncidents



$$\begin{cases} -\Delta u = f & \text{dans } \Omega \\ u = 0 & \text{sur } \partial\Omega \end{cases}$$

$$\Leftrightarrow \text{Trouver } u \in H_0^1(\Omega) \text{ tq } J(u) = \min_{v \in H_0^1(\Omega)} J(v)$$

avec $J(v) = \frac{1}{2} \int_{\Omega} |\nabla v|^2 - \int_{\Omega} f v$

Si on décompose en plusieurs sous-domaines :

$$\begin{cases} -\Delta u_i = f_i & \text{dans } \Omega_i \\ u_i = 0 & \text{sur } \partial\Omega_i \cap \partial\Omega \\ + \text{continuité aux interfaces} \end{cases}$$

On pose $J_i(v_i) = \frac{1}{2} \int_{\Omega_i} |\nabla v_i|^2 - \int_{\Omega_i} f_i v_i$

$$\text{Trouver } (u_1, u_2) \in H \text{ tels que } J_1(u_1) + J_2(u_2) = \min_{(v_1, v_2)} J_1(v_1) + J_2(v_2)$$

$$\text{avec } H = \left\{ (v_1, v_2) \in H^1(\Omega_1) \times H^1(\Omega_2) \mid \begin{array}{l} v_i = 0 \text{ sur } \partial\Omega_i \cap \partial\Omega, \\ v_1 = v_2 \text{ sur } \Gamma \end{array} \right\}$$

On dualise la contrainte : $\int_{\Gamma} \rho (u_1 - u_2) = 0, \forall \rho$

$$\mathcal{L}(v_1, v_2; \rho) = J_1(v_1) + J_2(v_2) + \int_{\Gamma} \rho (u_1 - v_2)$$

On cherche le point-selle de $\mathcal{L}(u_1, u_2; \lambda)$ vérifiant :

$$\mathcal{L}(u_1, u_2; \rho) \leq \mathcal{L}(u_1, u_2; \lambda) \leq \mathcal{L}(v_1, v_2; \lambda), \forall v_1, v_2, \rho$$

On dérive par rapport à v_1, v_2, ρ :

$$\begin{cases} \int_{\Omega_1} \nabla u_1 \nabla v_1 + \int_{\Gamma} v_2 \lambda = \int_{\Omega_1} f v_1 & \forall v_1 \in H^1(\Omega_1), v_1 = 0 \text{ sur } \partial\Omega_1 \cap \partial\Omega \\ \int_{\Omega_2} \nabla u_2 \nabla v_2 + \int_{\Gamma} v_2 \lambda = \int_{\Omega_2} f v_2 & \forall v_2 \dots \\ \int_{\Gamma} \rho (u_1 - u_2) = 0 \end{cases}$$

On discrétise ces équations :

- maillages (indépendants) de Ω_1, Ω_2 et Γ
- bases de fonctions sur ces maillages

$$\begin{cases} A_1 u_1 + B_1^t \lambda = f_1 \\ A_2 u_2 - B_2^t \lambda = f_2 \\ B_1 u_1 - B_2 u_2 = 0 \end{cases}$$

que l'on peut résoudre par méthode de Schur.

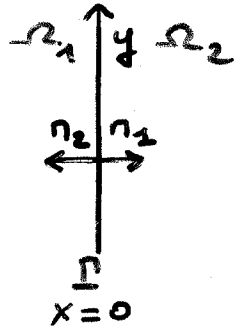
$$\begin{bmatrix} A_1 & 0 & B_1^t \\ 0 & A_2 & -B_2^t \\ B_1 & -B_2 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \lambda \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ 0 \end{bmatrix}$$

Méthodes "G.L.A"

Motivation: rapidité de convergence

* Théorie des conditions aux limites artificielles exactes

(Engqvist - Najda, 77; Halpern, 86; Halpern - Schatzmann, 89)



A résoudre :

$$Lu = f \text{ dans } \Omega = \Omega_1 \cup \Omega_2$$

$$\text{Soit } \Lambda_1 : u_0 \rightarrow \frac{\partial w}{\partial n_1}, \text{ où } w \text{ est solution de } \begin{cases} Lw = 0 & \Omega_2 \\ w = u_0 & \Gamma \end{cases}$$

$$\text{Soit } B_1 = \frac{\partial}{\partial n_1} - \Lambda_1$$

$$\text{Alors } v_1, \text{ solution de } \begin{cases} Lv_1 = f & \text{dans } \Omega_1 \\ B_1(v_1) = 0 & \text{sur } \Gamma \end{cases}$$

est l'exacte restriction de u sur Ω_1 si $\text{Supp}(f) \subset \Omega_2$

* On définit de même B_2 .

Avec ces opérateurs, la méthode de Schwarz

$$\begin{cases} Lu_1^{n+1} = f & \text{dans } \Omega_1 \\ B_1(u_1^{n+1}) = B_1(u_2^n) & \text{sur } \Gamma \end{cases} \quad \begin{cases} Lu_2^{n+1} = f & \text{dans } \Omega_2 \\ B_2(u_2^{n+1}) = B_2(u_1^n) & \text{sur } \Gamma \end{cases}$$

converge en exactement 2 itérations.

Pb: Λ_1, Λ_2 ne sont pas des opérateurs aux dérivées partielles (non-locaux, mise en oeuvre coûteuse et difficile)

Idée: construire des op. aux dérivées partielles qui approximent Λ_1, Λ_2

Exemple: $L u = c u + a \frac{\partial u}{\partial x} + b \frac{\partial u}{\partial y} - \nu \Delta u = f$

* On résout $\begin{cases} L w = 0 & \text{sur } \Omega_2 \\ w = u_0 & \text{sur } \Gamma \end{cases}$

par transformation de Fourier en y :

$$c \hat{w} + a \frac{\partial \hat{w}}{\partial x} + i b \xi \hat{w} + \nu \xi^2 \hat{w} - \nu \frac{\partial^2 \hat{w}}{\partial x^2} = 0$$

$$\rightarrow \hat{w}(x, \xi) = \hat{u}_0(\xi) e^{\lambda(\xi)x}$$

$$\text{avec } \lambda(\xi) = \frac{a - \sqrt{a^2 + 4c\nu + 4ib\nu\xi + 4\xi^2\nu^2}}{2\nu}$$

* D'où $\frac{\partial \hat{w}}{\partial x}(x=0, \xi) = \lambda(\xi) \hat{u}_0(\xi)$

D'où $\Lambda_1(u_0) = \text{T.F.}^{-1}(\lambda(\xi) \hat{u}_0(\xi))$

$$\Lambda_1(u_0)(y) = \int_{\mathbb{R}} \text{TF}^{-1}(\lambda)(y) u_0(1-y) ds$$

Complicé !!

* Développement limité de $\lambda(\xi)$:

$$\lambda(\xi) = \alpha_0 + \alpha_1 i \xi + \alpha_2 \xi^2 + \dots$$

d'où $\Lambda_1 = \alpha_0 \text{Id} + \alpha_1 \frac{\partial}{\partial y} - \alpha_2 \frac{\partial^2}{\partial y^2} + \dots$

d'où des approximations de $B_1 = \frac{\partial}{\partial x} - \Lambda_1$

→ Si on tronque à l'ordre 0: $B_1 = \frac{\partial}{\partial x} - \alpha_0 \text{Id}$ méthode "mixte"

→ On tronque en général à l'ordre l (e.g. Nataf & Rojter, 95)
→ gain important en efficacité.

→ Conditions optimisées pour écoulements tangents à Γ
(Taphet & Halpern, 98)