

Chapitre 9

Découpage

9.1 Présentation du problème

L'approche modulaire orientée objet est très intéressante pour l'utilisateur. Elle décrit le système à simuler en termes techniques et physiques. Les réflexions de la phase technique permettent de définir clairement des entités à la fois sur la hiérarchie et le découpage des modules. Ils sont habituellement des césures claires entre les composants, où des découpages paraissent naturels et faciles à faire (comme par exemple entre pompe, tuyau, capteur). L'impact du choix du découpage sur l'efficacité de la simulation n'est pas *a priori* connu. Le découpage qui est déduit du monde technique et physique, est-il numériquement efficace? Cette question est à la fois difficile et importante. On trouve une première discussion à ce sujet dans le cadre de l'environnement *Motor-2* dans [61]; nous la poursuivons ici.

La modularité de la description d'un système aboutit à un partitionnement du système global en sous-modules. Le chapitre précédant en donne quelques exemples. Posons le problème : quelle est la meilleure façon de découper un système pour que, non seulement la représentation reste la plus proche possible du système physique tel que nous le prenons, et pour que l'on atteigne la meilleure efficacité? Bien entendu, l'efficacité ne dépend pas seulement du découpage, mais en première ligne des algorithmes de résolution employés. L'aspect « résolution » n'est pas traité ici ; nous nous penchons sur l'influence du découpage sur l'efficacité des calculs. Pour mieux comprendre les conséquences du découpage, trois types d'investigations ont été faites :

- 1° Quelques résultats peuvent être obtenus par des considérations théoriques dans le cas simple d'une série de résistances (mur multicouche 1D en régime permanent).
- 2° Pour le cas d'une plaque 2D, nous avons effectué quelques expérimentations avec *Motor-2*. Les bords de la plaque sont soumis à des températures différentes et nous avons mesuré le temps nécessaire dans le cas de différents découpages pour déterminer la distribution des températures sur la plaque.

3° Dans un troisième test, nous avons mesuré l'influence des profondeurs de hiérarchisation sur le temps de calcul. L'exemple est une simulation dynamique d'un local sollicité par un rayonnement solaire.

Nous considérons la simulation la plus efficace, celle qui est la plus rapide et qui consomme le moins de ressources possible. Dans les réflexions théoriques, nous voulons minimiser le nombre d'itérations ou accélérer la vitesse de convergence. Pour les expérimentations numériques, nous avons mesuré le temps de calcul nécessaire pour effectuer la simulation.

9.2 Considérations théoriques

9.2.1 Problème

Considérons une série de n résistances thermiques de valeur $1/h_i = r_i$. Entre les résistances se trouvent les températures T_i avec T_0 et T_n aux bords. Une résistance r_i est placée entre les températures T_i et T_{i+1} (voir la figure 9.1).

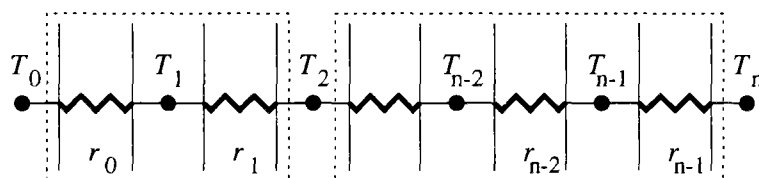


FIG. 9.1 - Une série de résistances thermiques avec des températures imposées T_0 et T_n . Les cases pointillées indiquent un partitionnement binaire possible.

Le flux de chaleur qui traverse un nœud s'écrit :

$$\phi_i = \frac{1}{r_{i-1}}(T_{i-1} - T_i) + \frac{1}{r_i}(T_i - T_{i+1}) = 0$$

Cette expression correspond à un problème de conduction thermique à travers une paroi multicouche avec des températures imposées sur les deux côtés (condition aux limites de type DIRICHLET). Le problème s'écrit sous la forme matricielle

$$\begin{bmatrix} -\frac{1}{r_0} - \frac{1}{r_1} & \frac{1}{r_1} & 0 & 0 & 0 \\ \frac{1}{r_1} & -\frac{1}{r_1} - \frac{1}{r_2} & \ddots & 0 & 0 \\ 0 & \frac{1}{r_2} & \ddots & \frac{1}{r_{n-3}} & 0 \\ 0 & 0 & \ddots & -\frac{1}{r_{n-3}} - \frac{1}{r_{n-2}} & \frac{1}{r_{n-2}} \\ 0 & 0 & 0 & \frac{1}{r_{n-2}} & -\frac{1}{r_{n-2}} - \frac{1}{r_{n-1}} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_{n-2} \\ T_{n-1} \end{bmatrix} = \begin{bmatrix} -\frac{1}{r_0} T_0 \\ 0 \\ \vdots \\ 0 \\ -\frac{1}{r_{n-1}} T_n \end{bmatrix}$$

La solution classique pour ce système d'équations est pour une température T_i au milieu, $1 \leq i \leq n-1$

$$T_i = \frac{\frac{T_0}{\sum_{p=0}^{i-1} r_p} + \frac{T_n}{\sum_{p=i}^{n-1} r_p}}{\frac{1}{\sum_{p=0}^{i-1} r_p} + \frac{1}{\sum_{p=i}^{n-1} r_p}} \quad (9.1)$$

Le problème que l'on se pose maintenant est comment trouver un partitionnement *optimal* pour que la convergence d'un algorithme de type NEWTON-RAPHSON se fasse le plus rapidement possible. Au niveau physique, on peut interpréter le partitionnement comme la division de la paroi en plusieurs ensembles de couches. Ces ensembles sont des *boîtes noires* avec leurs propres températures et flux thermiques et ses algorithmes internes pour les déterminer (cf. chap. 6). Un partitionnement est considéré comme *optimal* quand la convergence est la plus rapide de celles que l'on obtient avec tous les partitionnements possibles au niveau supérieur. Ce n'est pas le coût global en calculs, car nous ne faisons aucune hypothèse sur les algorithmes internes des sous-blocs. Le coût en calcul d'un module composé est la somme du coût correspondant aux sous-modules plus le coût de résolution des couplages.

9.2.2 Partitionnement binaire

Nous ne considérons que les cas où les résistances ou les blocs de résistances sont groupés par paires. C'est ce que nous appelons un partitionnement binaire. Par conséquent, la vitesse de convergence au niveau supérieur (entre les deux blocs) est simple à déterminer. Si le partitionnement est fait à la température T_i entre la résistance r_i et r_{i+1} , nous obtenons la valeur de T_i , s'il y a itération entre deux blocs, par la condition de flux (expression de l'interface)

$$T_i = \frac{1/r_{i-1}T_{i-1} + 1/r_i T_{i+1}}{1/r_{i-1} + 1/r_i} \quad (9.2)$$

Maintenant, T_{i-1} et T_{i+1} peuvent être obtenus par l'équation 9.1 (l'expression des sous-modules). Elles sont donc calculées par leurs blocs (avec une méthode numérique quelconque) et avec les températures aux limites T_0 et T_i pour T_{i-1} , T_i et T_n pour T_{i+1} . Ces températures internes dans les blocs se calculent donc comme

$$T_{i-1} = \frac{\frac{T_0}{\sum_{p=0}^{i-2} r_p} + \frac{T_i}{\sum_{p=i-1}^{i-1} r_p}}{\frac{1}{\sum_{p=0}^{i-2} r_p} + \frac{1}{\sum_{p=i-1}^{i-1} r_p}}$$

et

$$T_{i+1} = \frac{\frac{T_i}{\sum_{p=i}^i r_p} + \frac{T_n}{\sum_{p=i+1}^{n-1} r_p}}{\frac{1}{\sum_{p=i}^i r_p} + \frac{1}{\sum_{p=i+1}^{n-1} r_p}}$$

En combinant la condition de flux (éq. 9.2) et les deux solutions des blocs pour les températures T_{i-1} et T_{i+1} autour du point de partitionnement, nous obtenons une équation qui exprime la température à la césure T_i en fonction d'elle-même $T_i = f(T_i, T_0, T_n)$. Ceci est la base de la méthode itérative pour calculer T_i . On peut ensuite exprimer le rapport de convergence comme

$$\frac{1}{1/r_{i-1} + 1/r_i} \left[\frac{1/r_{i-1}}{1 + \frac{r_{i-1}}{\sum_{j=0}^{i-2} r_j}} + \frac{1/r_i}{1 + \frac{r_i}{\sum_{j=i+1}^{n-1} r_j}} \right] \quad (9.3)$$

Cette expression 9.3 est positive et inférieure à 1, si tous les coefficients de transfert h sont positifs. Pour accélérer la convergence, il faut minimiser cette expression en choisissant un endroit de découpage i . Minimiser l'expression 9.3 revient à maximiser l'expression suivante

$$\frac{\frac{1}{\sum_{j=0}^{i-1} r_j} + \frac{1}{\sum_{j=i}^{n-1} r_j}}{\frac{1}{r_{i-1}} + \frac{1}{r_i}} \quad (9.4)$$

Pour un ensemble de résistances données r_j , l'expression 9.4 doit être maximale. Pour mieux comprendre cette expression, regardons les deux exemples d'une isolation entre deux blocs conductifs et d'un élément conducteur entre deux blocs de résistances.

Si toutes les résistances r_j sont égales et ont la même valeur r sauf une r_k dont la valeur R est beaucoup plus grande que r , alors le découpage doit être fait au nœud T_{k+1} si r_k se trouve dans la deuxième moitié ($k > n - 1 - k$) et au nœud T_k pour r_k dans la première moitié ($k < n - 1 - k$). Ceci veut dire que la meilleure césure doit être faite au ras de l'isolant du côté la plus proche du bord (voir fig. 9.2). Ce résultat soutient donc l'avis qu'une isolation sépare deux blocs et un découpage à cet endroit paraît naturel.

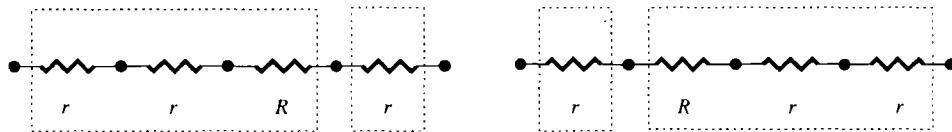


FIG. 9.2 - Le découpage binaire se fait à la résistance isolante du côté la plus proche du bord.

Si toutes les résistances r_j sont égales et ont la même valeur r sauf une r_k dont la valeur est beaucoup plus petite que r , alors le découpage optimal se trouve aux nœuds T_1 ou T_{n-1} , c'est à dire aux bords de la chaîne de résistances. Les deux découpages (T_1 ou T_{n-1}) donnent la même vitesse de convergence. Si $k = 0$ ou $k = n - 1$, le découpage doit être au bout opposé de la petite résistance.

Nous pouvons conclure de ces deux exemples que le partitionnement binaire doit se trouver proche des grandes résistances et loin des petites résistances. Une règle simple résultante peut être le raisonnement suivant : une grande résistance entraîne un petit flux, nous avons donc un couplage faible et nous pouvons couper ici. Bien entendu, ces résultats ne sont valables aussi directement que pour des systèmes non-capacitifs.

9.3 Expérimentation : le cas 2D

Le cas 2D ne se laisse pas analyser aussi facilement que le cas monodimensionnel. Nous l'avons expérimenté numériquement sur quelques cas. Le problème étudié est une plaque composée de 8×8 éléments comme dans la figure 9.3.

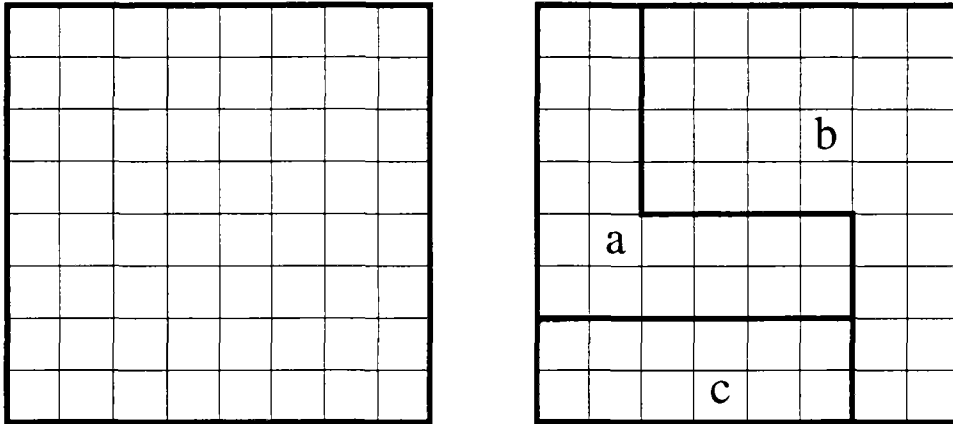


FIG. 9.3 - Une surface carrée avec un maillage de 8×8 éléments. À droite se trouve un exemple d'un partitionnement en trois sous-modules.

L'équation interne des éléments est du type différences finies sans capacité interne dans les nœuds (plaque non-capacitive).

Nous générons des partitionnements de plaque par un logiciel écrit en Prolog qui est dérivé des travaux de ALLAZ [2]. Ce programme prend comme paramètre le nombre de sous-modules souhaités et génère l'ensemble de tous les partitionnements possibles. Par quelques étapes intermédiaires, nous créons la description en format SYMBOL pour Motor-2. Ceci comprend les fichiers nécessaires des cellules élémentaires, les modules composés avec les interfaces internes pour le raccordement entre les mailles et le module composé principal qui raccorde les partitions.

Ensuite la simulation par Motor-2 est lancée. Nous déterminons le coût des calculs comme étant le temps nécessaire pour que Motor-2 calcule l'état stationnaire. Comme il n'y a pas de capacité dans les éléments, cela veut dire que l'on calcule un pas de temps. Pour chaque configuration, nous stockons plusieurs informations comme le temps de simulation, la longueur des frontières internes, le nombre de partitions, le nombre d'éléments par partitions, etc. Suite aux variations inévitables de temps mis pour des configurations identiques à cause des charges différentes des ordinateurs, les mêmes cas sont simulés plusieurs fois afin d'obtenir des meilleures estimations sur le coût de calcul.

Une première étude qualitative se fait en regardant simplement le fichier créé. Des résultats plus quantitatifs peuvent être obtenus par un post-processeur qui crée une table des temps nécessaires et qui calcule la multi-régression linéaire de toutes ces données. Ce post-processeur est écrit en Maple et utilise ses fonctionnalités de calcul formel. Finalement on obtient une corrélation entre le temps de calcul et des puissances de la surface des partitions et de la longueur interne des frontières jusqu'à l'ordre 3. Nous mesurons également les différences entre les valeurs initiales et les solutions pour toutes les cellules.

Plusieurs configurations ont été testées. La plaque a été coupée en deux, trois ou quatre partitions. Plusieurs centaines de formes différentes pour chaque nombre de partition ont été étudiées. Différentes températures aux bords mon-

trent une influence des conditions aux limites sur les coûts de calcul. Mais on constate aussi que la distribution des conductivités sur la plaque et leurs changements par rapport aux partitions influencent le temps nécessaire.

L'interprétation des résultats que nous avons obtenus n'est pas évidente. Nous avons trouvé quelques configurations dont les résultats semblent à première vue contradictoires ou tout au moins difficiles à expliquer.

9.3.1 Cas homogène

Les tests suivants ont été exécutés sur une plaque homogène de conductivité $0.001W/m^2K$. Les températures de bord étaient fixées par une condition de DIRICHLET à $5^{\circ}C$ en bas, $10^{\circ}C$ en haut, $15^{\circ}C$ à gauche, et $20^{\circ}C$ à droite. L'état initial est de $19^{\circ}C$ partout. La figure 9.4 montre la solution, c'est à dire le régime permanent, du problème. On voit bien les températures fixées aux bords. Le champ de températures à l'intérieur de la plaque crée une forme de selle. Le « haut » de plaque se trouve derrière, le « bas » au premier plan de la figure.

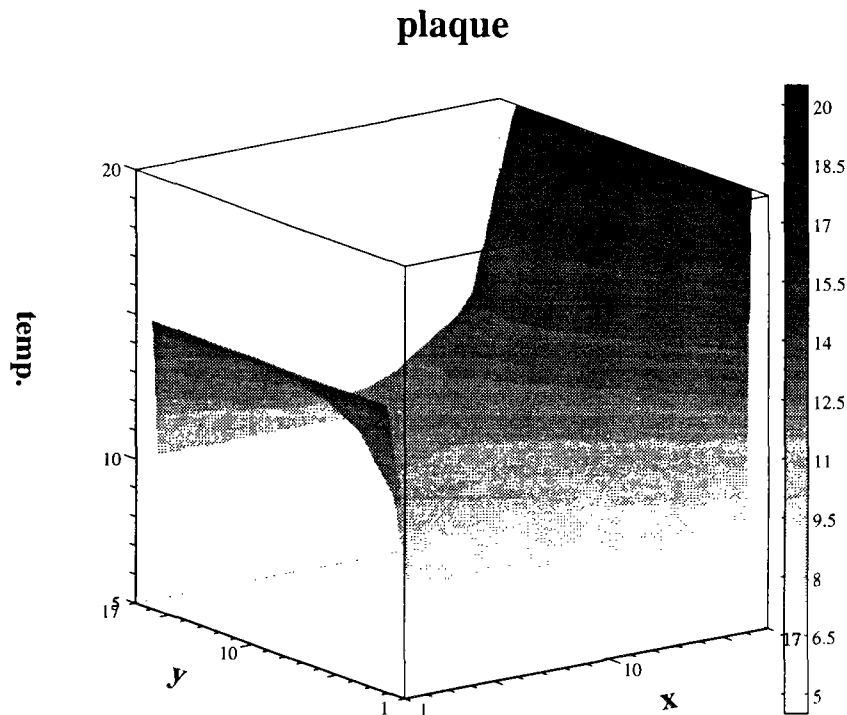


FIG. 9.4 - Le champ de températures sur la plaque avec des températures imposées aux bords de $5^{\circ}C$ en bas, $10^{\circ}C$ en haut, $15^{\circ}C$ à gauche, et $20^{\circ}C$ à droite.

Qualitativement on peut remarquer que le meilleur découpage est celui où les partitions sont de tailles comparables. L'introduction d'une très petite partition dans une configuration aboutit à une simulation plus longue. Ce résultat paraît

naturel, car une nouvelle petite partition ne change que très peu les partitions existantes, mais elle accroît la longueur des frontières et donc la taille du système d'équations qui est à résoudre (inversion de la *Jacobienne*).

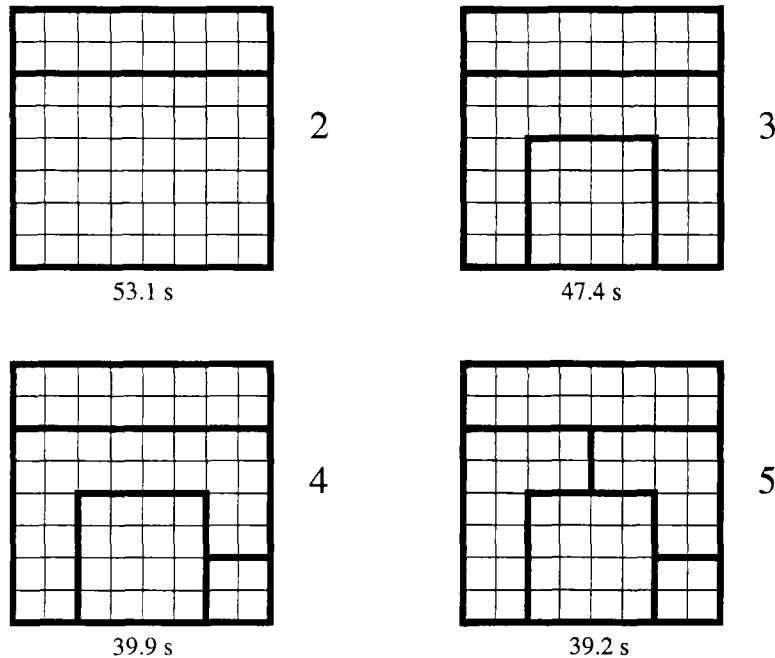


FIG. 9.5 - Exemple des partitionnements où l'ajout d'une partition supplémentaire accélère les calculs.

Par ailleurs, si un partitionnement existe déjà, par exemple imposé par la description physique, il peut être parfois favorable d'introduire des partitions supplémentaires. En général, si on a un nombre faible de partitions, on accélère la simulation en ajoutant d'autres partitions (d'une taille comparable). La figure 9.5 montre des temps de calcul décroissant pour un découpage en deux, trois, quatre et cinq partitions. Il faut noter ici, qu'il n'est pas évident de trouver un tel exemple. On n'est pas vraiment certain de diminuer à coup sûr le temps de calcul en ajoutant une partition.

Cette conclusion qualitative est confirmée par des corrélations. Elles sont calculées par le post-processeur sur un ensemble de partitionnements de la plaque. Regardons premièrement les corrélations qui sont indépendantes de la différence entre solution et valeur initiale. Nous désignons par C_i les coûts de calcul (le temps nécessaire) pour un partitionnement donné; C_2 désigne donc les coûts de calcul pour un partitionnement binaire. Les σ_i sont les sommes normées des i^{es} puissances des surfaces des sous-modules, et l est la longueur normée des frontières internes. Les normes sont établies par rapport à la surface globale (64) et par rapport à la longueur interne globale ($2 \times 7 \times 8 = 112$). La corrélation entre la formule construite et les durées des simulations est de 0.79 pour le partitionnement binaire, 0.69 pour le partitionnement tertiaire, et 0.51

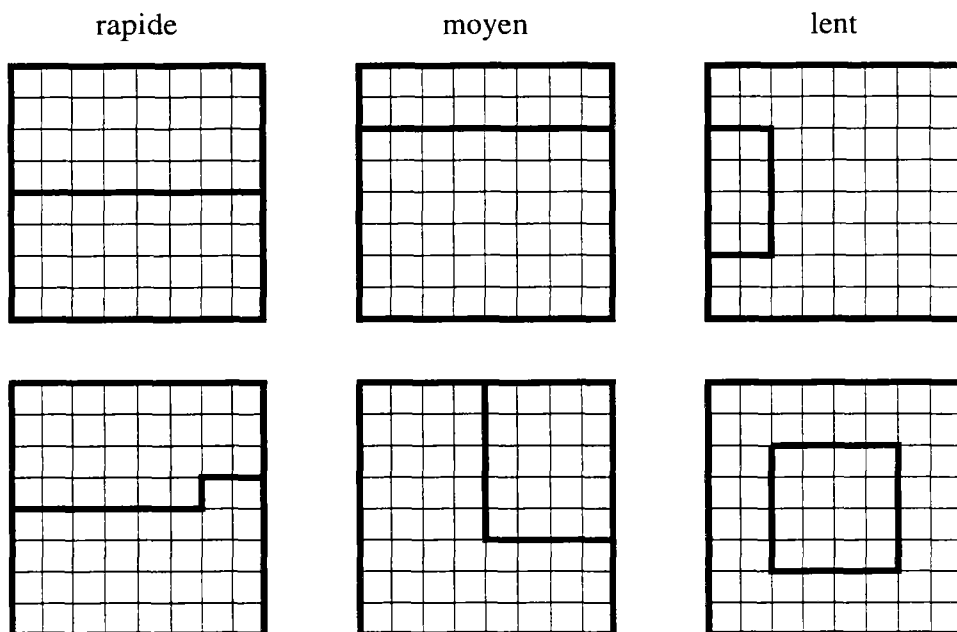


FIG. 9.6 - Exemple des partitionnements en deux avec un temps de calcul croissant.

pour le partitionnement quatuoraire.

2 partitions corrélation = 0.79 (cf. fig. 9.6).

$$C_2 = 219.5 - 1093.3\sigma_1 + 2300.82\sigma_2 - 1095.9\sigma_3 + 0.4l - 98.3l^2 - 34.1l^3$$

3 partitions corrélation = 0.69 (cf. fig. 9.7).

$$C_3 = 215.3 - 130.1\sigma_1 - 1775.3\sigma_2 + 4163.8\sigma_3 + 109.6l - 33.1l^2 - 6.4l^3$$

4 partitions corrélation = 0.51 (cf. fig. 9.8).

$$C_4 = 268.3 + 123.4\sigma_1 - 562.4\sigma_2 + 819.3\sigma_3 + 145.8l - 222.4l^2 + 57.7l^3$$

Les résultats et les corrélations dépendent aussi des conditions aux limites. C'est surtout le cas, si le partitionnement suit des isothermes de la solution. Ici, le même partitionnement peut donner des résultats très différents, si les conditions aux limites ne correspondent pas aux frontières des partitions. Cela peut

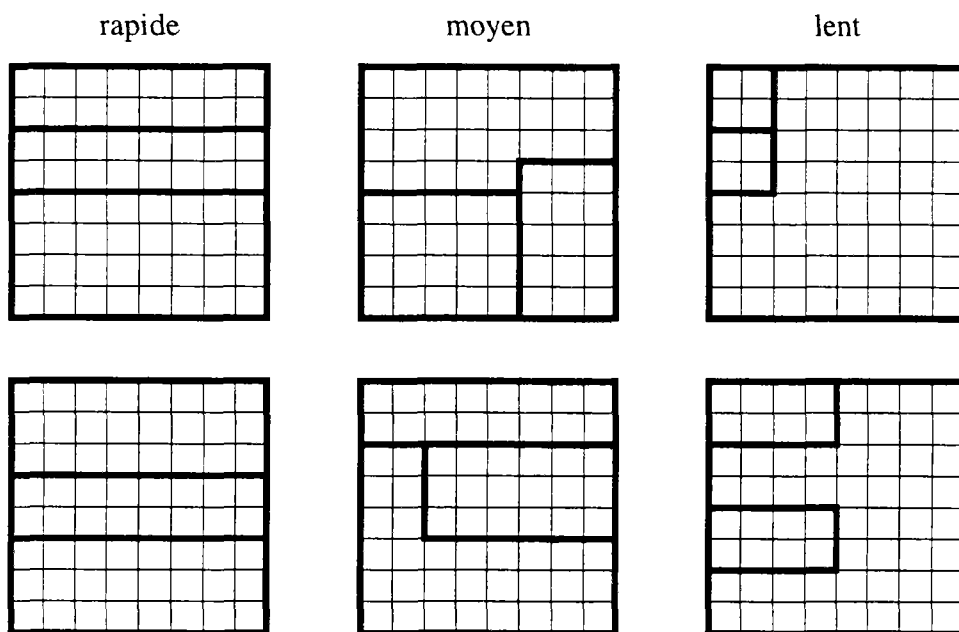


FIG. 9.7 - Exemple des partitionnements en trois avec un temps de calcul croissant.

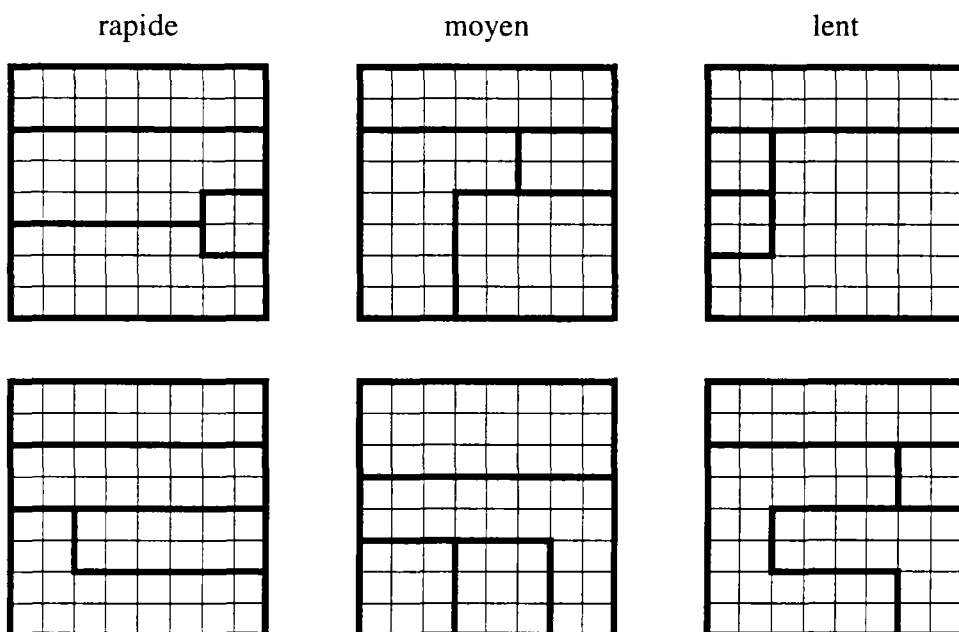


FIG. 9.8 - Exemple des partitionnements en quatre avec un temps de calcul croissant.

s'expliquer par le fait que la valeur initiale est plus éloignée de la solution, et il faut donc plus de temps pour converger. En introduisant cette différence dans les corrélations, nous avons défini un paramètre supplémentaire ξ qui contient la somme des différences entre les valeurs initiales et les solutions le long des frontières.

$$\xi_i = \int_{\text{frontières}} (T_{\text{init}}(s) - T_{\text{sol}}(s))^2 ds$$

où T_{init} est la valeur initiale, T_{sol} la valeur de la solution, s suit les frontières internes, et i un entier (1, 2, 3). Les corrélations obtenues sur les résultats ne changent que peu. Elles valent 0.86, 0.71, 0.52 pour les formules suivantes.

2 partitions corrélation = 0.86.

$$C = 408.1 - 806.9\sigma_1 + 1593.9\sigma_2 - 756.6\sigma_3 + 1262.8\xi_1 + 3306.5\xi_2 + 2966.7\xi_3 - 78.1l + 108.7l^2 - 24.4l^3$$

3 partitions corrélation = 0.71.

$$C = -213.3 + 49.8\sigma_1 - 2448.6\sigma_2 + 4856.0\sigma_3 - 3680.1\xi_1 - 14984.3\xi_2 - 16699.0\xi_3 + 495.8l - 256.4l^2 + 35.1l^3$$

4 partitions corrélation = 0.52.

$$C = 207.6 + 116.1\sigma_1 - 523.3\sigma_2 + 785.5\sigma_3 - 1802.9\xi_1 - 6840.4\xi_2 - 7271.4\xi_3 + 24.6l - 143.3l^2 + 42.1l^3$$

On voit que les coefficients des paramètres ξ_i sont assez importants, mais les corrélations ne s'améliorent que très peu. Généralement on peut dire que les corrélations ne nous donnent pas vraiment des bons indices sur la forme des découpages. Aucun des chiffres n'est donc tellement particulier que l'on puisse le prendre comme critère.

9.3.2 Cas hétérogène

Les exemples ci-dessus utilisent une conductivité uniforme. Quelques tests ont été également faits pour une plaque avec une conductivité inhomogène. Comme premier cas, nous avons pris un $K = 0.001W/m^2K$ dans la moitié gauche et un $K = 1W/m^2K$ dans la moitié droite. Pour un partitionnement binaire (deux sous-domaines), on peut remarquer que le temps de calcul nécessaire augmente pour toutes les configurations de 2 à 10 % par rapport au cas homogène. Néanmoins, il y a une exception remarquable. Si la frontière entre les deux partitions suit exactement la discontinuité au milieu de la plaque, la simulation s'accélère. Ce résultat est une confirmation des résultats théoriques obtenus pour le cas monodimensionnel. Couper un système à l'endroit où se trouve une césure physique rend la simulation plus efficace.

Ce dernier résultat pouvait être confirmé par d'autres tests. Si l'on introduit des discontinuités qui suivent des partitions existantes, le temps de calcul diminue généralement entre 3 et 12 %. L'accélération est plus petite, si le partitionnement est régulier avec des grandes zones. Elle est plus grande pour un partitionnement irrégulier et des petits sous-domaines.

9.4 Tests sur une simulation dynamique

Finalement, nous avons testé l'influence du découpage sur la vitesse de la simulation à l'aide d'un exemple dynamique. Nous considérons une pièce simple carrée comme un problème bidimensionnel (fig. 9.9). Elle est composée de trois murs, quatre coins et le quatrième mur est remplacé par une fenêtre par laquelle entre le rayonnement solaire. L'environnement de la pièce se trouve à température constante, 19°C , partout.

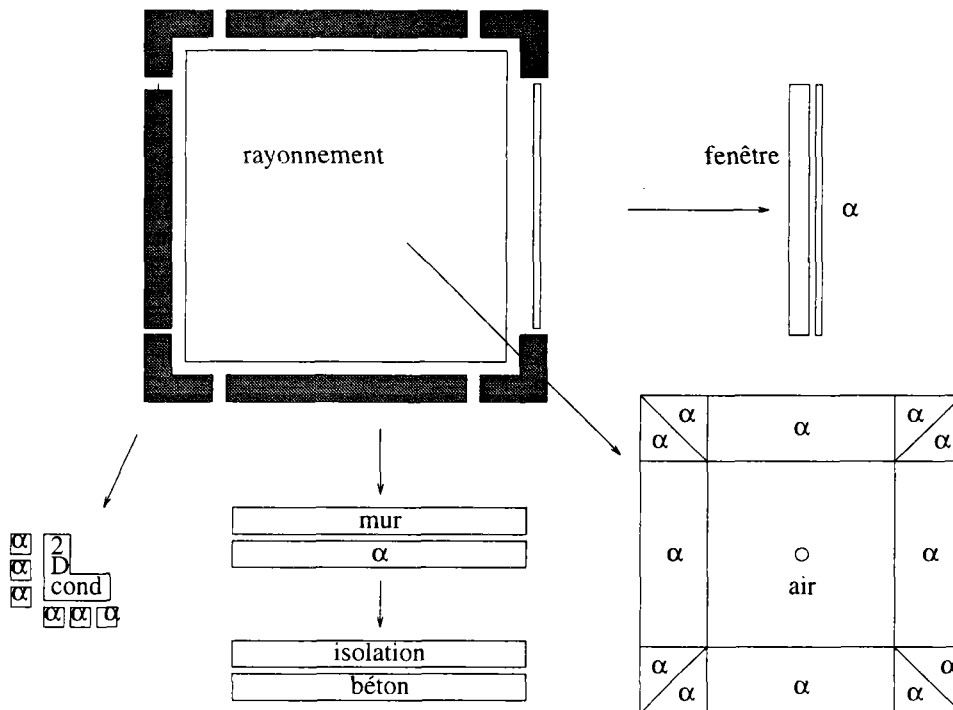


FIG. 9.9 - La pièce et un découpage possible.

Au niveau physique, nous ne supposons une conduction bidimensionnelle que dans les coins ; les murs sont suffisamment bien décrits par une conduction monodimensionnelle. La température interne est considérée comme homogène, un seul nœud capacitif le représente. Des coefficients d'échange de chaleur décrivent le flux entre les surfaces des murs et l'air intérieur. À l'exception du rayonnement solaire, le rayonnement entre les surfaces est infrarouge, et la fenêtre se comporte donc pour ce phénomène comme un mur. Le flux solaire est absorbé en partie par la fenêtre, mais la plupart est reçu par la surface à l'opposé de la fenêtre (mur nord). Le rayonnement solaire absolu varie en fonction

des données météorologiques.

Les niveaux mathématiques et algorithmiques ne sont pas détaillés pour les modules élémentaires de cet exemple. Ici notre préoccupation principale est l'influence du découpage sur le temps de calcul nécessaire. Néanmoins, on peut remarquer un point important de la modularité de Motor-2, car les modèles utilisés ont des origines très différentes. Nous utilisons les modèles déjà utilisés de coefficient d'échange et de conduction monodimensionnelle, mais aussi un modèle modal [52] pour la conduction bidimensionnelle dans les coins, et un modèle de rayonnement [17].

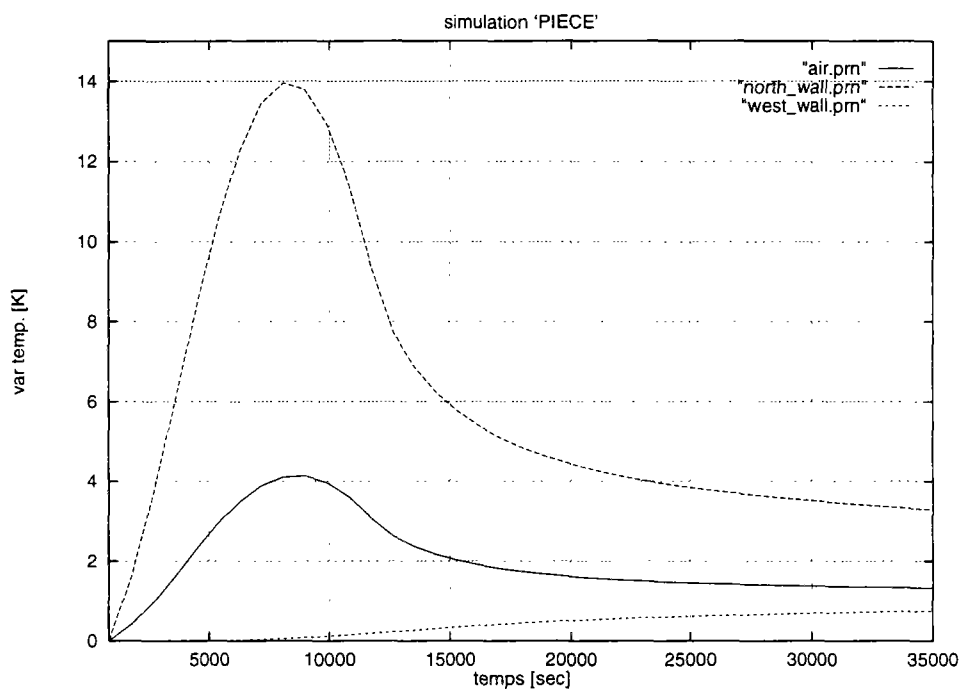


FIG. 9.10 - Résultats de quelques données de la simulation.

Nous avons mesuré le temps nécessaire pour une simulation de 46.800 secondes avec un pas de temps de 900 secondes. Le flux absorbé dans la fenêtre et sur la face nord du local varie en sinusoïde. C'est le moteur de la dynamique de notre système. Dans la figure 9.10 les évolutions de quelques variables sont tracées. On remarque bien le chauffage rapide de l'air intérieur qui n'a qu'une très petite capacité. La ligne continue montre l'évolution de la température de la surface nord. Ici est absorbé le flux solaire qui chauffe l'air et en moindre degré le béton du mur. À l'intérieur du mur ouest entre la couche de béton et celle d'isolant, l'augmentation de la température ne commence de se faire sentir qu'après 10.000 secondes (la ligne pointillée).

La figure 9.11 montre deux découpages différents. L'arbre PROF à gauche est volontairement ramifié profondément. Nous avons des modules terminaux jusqu'au niveau trois au-dessous du module principal. À l'opposé de ce découpage se trouve la configuration PLAT à droite, où tous les modules terminaux sont directement couplés par le module principal. Nous avons donc une structure plate

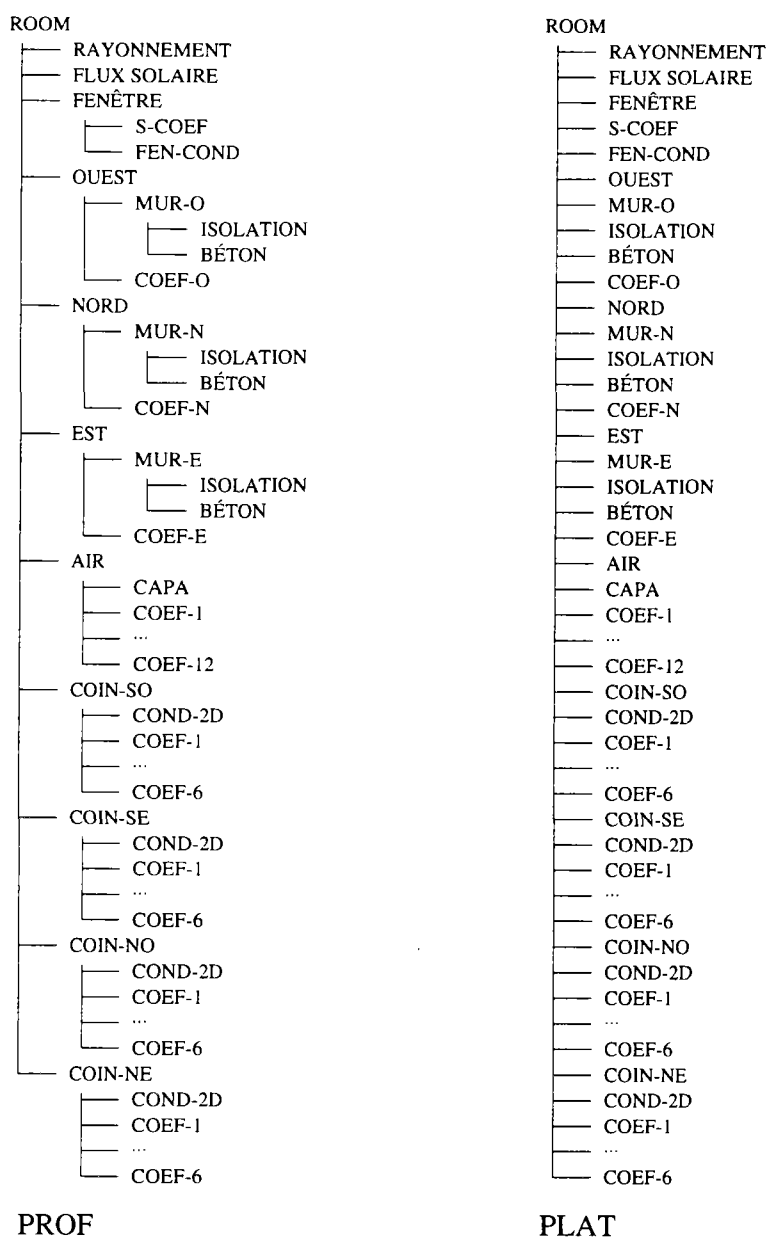


FIG. 9.11 - Différents découpages extrêmes de la pièce.

comme on la trouve généralement dans les environnements non-hiérarchisés.

Trois autres configurations ont été testés également. Leur découpage est montré dans la figure 9.12. Le degré de ramification des trois configurations se trouve entre les deux extrêmes précédents. Dans la configuration AIR-FEN-PLAT à gauche, nous avons gardé la structure hiérarchique. Seuls les sous-modules de la fenêtre et de l'air interne ont été ramenés au plus haut niveau. La configuration COIN-PROF au milieu est comparable à la structure plate, mais les sous-modules des coins ont été laissés hiérarchisés. Dans la dernière configuration COIN-MUR-PROF à droite, les sous-modules du mur nord ont été résolus et mis au plus haut niveau, pour le reste, c'est la même configuration que

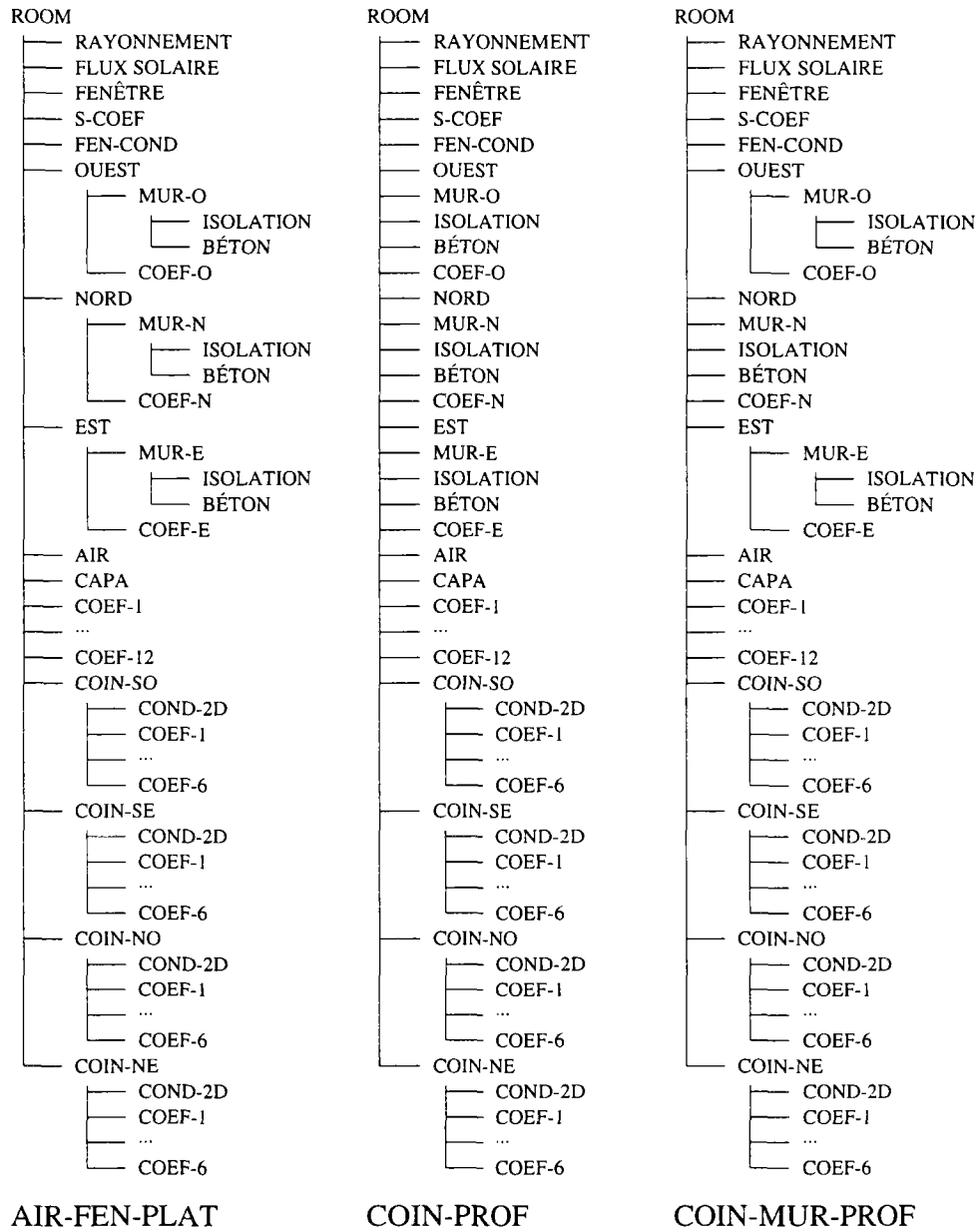


FIG. 9.12 - Découpages intermédiaires de la pièce.

AIR-FEN-PLAT.

	PLAT	PROF	AIR-FEN-PLAT	COIN-PROF	COIN-MUR-PROF
temps [sec]	198,4	166,1	152,1	163,9	153,6
gain [%]		16,1	23,3	17,4	22,6

FIG. 9.13 - Les temps utilisateur des simulations pour les différentes configurations.

Pour toutes les configurations, nous avons appliqué la méthode GLOBALE pour la résolution des modules composés. Les temps d'exécution nécessaires

pour chacune des configurations sont listés dans le tableau 9.13. On remarque tout de suite que la configuration où tous les modules sont mis à plat, est la plus lente. C'est certainement dû à l'évaluation numérique d'une grande matrice *Jacobienne*, bien que la plupart des interfaces ne varient pas. La configuration PROF avec beaucoup de ramifications est déjà 16 % plus rapide. Mais toutes les configurations intermédiaires sont encore plus rapides. Le gain de vitesse peut atteindre 23,3 % pour la configuration AIR-FEN-PLAT. À peu près le même résultat est obtenu avec la configuration COIN-MUR-PROF. Dans les deux cas, les sous-modules des coins et les murs des cotés n'ont pas été mis à plat. Les valeurs numériques ne changent que très peu dans ces sous-modules. La matrice globale de dépendances n'est donc pas beaucoup perturbée par l'omission de ces modules.

Pour bien accélérer un tel système, il faut savoir *a priori* où se passe les calculs et quels sont les endroits qui restent plus ou moins stables. Si l'on arrive à bien séparer ces régions, on peut espérer des gains de vitesses considérables.

9.5 Conclusion

Les expérimentations effectuées sur la simple plaque aussi bien que sur l'exemple du local montrent parfois une bonne correspondance entre un découpage guidé par des considérations techniques et un découpage en vue d'une efficacité numérique. Néanmoins, quelques configurations singulières parmi les découpages de la plaque nécessitent des coûts importants sans que nous sachions d'où vient ce temps de calculs excessif. Les corrélations entre les vrais coûts et les fonctions construites ne sont pas suffisamment grandes pour conclure avec certitude un schéma de découpage efficace.

Il faut noter que d'autres approches modulaires de la simulation ne s'appuient pas du tout sur des considérations techniques pour la mise en œuvre de la modularité. L'environnement *Spark* contient son propre algorithme interne pour la réduction des équation. Ceci peut aussi être vu comme comme une sorte de « découpage ».

Sous le nom de « décomposition de domaine » (*domain decomposition*) s'est établie récemment une brache de mathématiques et génie d'ingénieur qui étudie une classe de méthodes numérique pour obtenir des solutions par combinaison des solutions des sous-domaines. On s'y interroge également sur les sous-structurations adéquates pour des grands systèmes d'équations différentielles partielles. Ces équations ont souvent la même structure qu'elles proviennent d'une modélisation par différences ou par éléments finis. La combinaison des solutions des composants d'une physique différente comme c'est le cas dans *Motor-2* n'est pas traité.

Le problème du découpage est d'un intérêt évident, notamment dans la conception des programmes liés à une architecture d'ordinateurs à haute performance.

