CSMA 2019
14ème Colloque National en Calcul des Structures
13-17 Mai 2019, Presqu'île de Giens (Var)

Méthodes de remaillage parallèle pour le calcul intensif de structures industrielles

C. Bovet[1], O. Ciobanu[2], A. Parret-Fréaud[2,⋆]

[1] *ONERA – The French Aerospace Lab, F-92322 Châtillon, France, christophe.bovet@onera.fr*
[2] *Safran Tech, Modelling & Simulation, Rue des Jeunes Bois, Châteaufort, 78114 Magny-Les-Hameaux, France*
{oana-alexandra.ciobanu, augustin.parret-freaud}@safrangroup.com

Résumé — Ce travail aborde la problématique de l'adaptation de maillage parallèle au cours de calculs de structures par élément finis résolu par des méthodes de décomposition de domaine sans recouvrement. L'objectif est de disposer d'algorithmes permettant d'orchestrer efficacement des phases de remaillage séquentiel appliquées à des sous-domaines distincts dans un contexte distribué, en restant autant que possible agnostique aux outils et librairies sous-jacents. Deux algorithmes de ce type sont présentés et leurs performances illustrées sur la base d'un cas métier à géométrie simplifiée.
Mots clés — remaillage parallèle, décomposition de domaine.

1 Introduction

L'accroissement continu des ressources de calcul ouvre de nouvelle perspectives dans le domaine de la conception en rendant possible le recours accru aux simulations numériques sur des modèles haute fidélité au cours des étapes des processus de conception. Dans ce contexte, disposer de solveurs adaptés aux architectures de calcul massivement parallèles actuelles est un prérequis incontournable. Cependant, il est également indispensable, à terme, de pouvoir accéder à des solutions dont le niveau de précision puisse respecter une exigence initialement fixée, ou bien resulte d'un optimum par rapport à un budget de calcul donné. Pour se faire, il est nécessaire de disposer de méthodes efficaces d'adaptation de maillage sur la base d'estimateurs d'erreur et de métriques *ad hoc*.

L'adaptation de maillage dans un cadre HPC en contexte industriel soulève plusieurs difficultés. D'une part, les outils de remaillage séquentiels affichent souvent une efficacité limitée dès lors que les maillages en jeu sont volumineux (plusieurs dizaines de millions de nœuds) et les géométries complexes, entrainant des goulets d'étranglement lors de calculs adaptatifs, voire l'impossibilité de générer le nouveau maillage cible. D'autre part, la réalisation de calculs HPC haute fidélité sur des maillages fins passe nécessairement par le recours à des techniques de parallélisme à mémoire distribuée où la donnée est répartie sur différents noeuds de calcul. Il est donc crucial de disposer de méthodes de remaillage à même de gérer le caractère distribué des maillages d'entrée et de paralléliser au mieux les opérations d'adaptation. Outre la réduction des données échangées entre processus, l'une des principales difficultés liées aux approches de remaillage parallèle réside dans le traitement de l'interface entre les sous-domaines afin de garantir la conformité des maillages générés.

Si les techniques de remaillage parallèle restent, à notre connaissance, très peu utilisées en calcul de structures, la littérature est riche de plusieurs approches développées durant les deux dernières décennies (voir, par exemple, [1] pour une revue de certaines méthodes). De même que pour les solveurs, deux approches coexistent : parallélisation des opérateurs de raffinement (sous-division, insertion/suppression des nœuds, basculement d'arêtes, etc.) ou orchestration itérative de phases séquentielles de maillage. Dans cette seconde catégorie, en fonction de l'ordre d'application des étapes de maillage, on identifie les algorithmes suivants :

- maillage des interfaces des sous-structures puis distribution et maillage parallèle des sous-domaines [2] ;
- processus itératif de maillage parallèle des sous-domaines avec interface figée [3], où la même étape est répétée jusqu'à traitement des interfaces volumiques de tous les sous-domaines ;
- processus itératif de remaillage parallèle des sous-domaines à interface figée suivi d'une nouvelle

sous-structuration afin de remailler complètement les interfaces [4] ;
— algorithme en 5 étapes [5, 6] (développé dans la librairie PaMPA et compatible avec les (re)mailleurs séquentiels MMG3D, TetGen et Gmsh) : sous-découpage initial (1) et identification des zones à remailler (2), extraction (3) et redistribution aux processus disponibles des zones pour remaillage (4), enfin réintégration par fusion des nœuds aux interfaces figées (5), éventuellement suivi d'un repartitionnement dynamique afin de rééquilibrer les charges entre processus ;
— raffinement de maillage parallèle de sous-domaines à interface figée à l'aide de motifs, avec traitement de l'interface *a priori* [7] ou *a posteriori* [8, 9] ;
— remaillage à l'aide de cellules fantômes [10] : décomposition en sous-domaines avec recouvrement d'une cellule puis remaillages parallèle des sous-domaines avec interfaces figées, la position finale des interfaces étant la moyenne des nouvelles positions des interfaces partagées.

Les approches développées dans cet article se situent dans le cadre d'application des méthodes de décomposition de domaine sans recouvrement. Lors de précédents travaux [11], la méthode FETI (Finite Element Tearing and Interconnecting) a été équipée d'un algorithme de gradient conjugué multi-préconditionné adaptatif, donnant naissance à une nouvelle méthode de décomposition de domaine nommée *Adaptive MultiPreconditionned FETI* (AMPFETI). Cette méthode s'est révélée suffisamment robuste pour pemettre son utilisation en vue de la résolution de cas métiers industriels en régime non linéaire sur plusieurs centaines de coeurs de calcul. On cherche à présent à se doter d'approches de remaillage parallèle tirant avantageusement partie des spécificités du cadre des décomposition de domaine sans recouvrement. Pour se faire, notre choix s'est porté sur le développement d'approches de type orchestration itératives de phases séquentielles de (re)maillage, qui ont pour avantages de permettre la réutilisation des outils de maillage séquentiels existants. Par ailleurs, dans l'optique de réaliser efficacement des étapes de remaillage en cours de calcul réalisés par décomposition de domaine sans pénaliser les performances globales, le choix a été fait de reposer sur une unique décomposition tant pour l'étape de remaillage parallèle que pour le solveur. Enfin, on cherche également à préserver la géométrie des interfaces entre sous-domaines de sorte d'éviter les échanges de données entre processus lors des phases de transferts de champs et de permettre le recyclage des espaces de Krylov du solveur itératif entre les différentes phases d'adaptation du maillage.

Les deux parties suivantes présenteront successivement deux approches de remaillage parallèle développée au cours de ce travail, ainsi qu'une étude de performance sur un cas simplifé. L'ensemble des développements logiciels a été réalisé dans la plateforme éléments finis Z-set [12]. La bibliothèque de remaillage MMG3D [13] est utilisée comme outil de remaillage séquentiel de manière non intrusive, afin notamment de permettre l'usage d'autres outils du même type sous réserve de leur interfaçage avec le code Z-set.

2 Approche non intrusive de remaillage parallèle de type maître-esclave

En préalable, on rappelle que les deux algorithmes présentés par la suite s'appliquent sur un maillage initialement découpé en sous-domaines non-recouvrants et distribués sur plusieurs processus distincts. Par ailleurs, le choix de ne reposer que sur la bibliothèque MMG3D comme outil de remaillage séquentiel n'est pas anodin et impose de disposer d'un maillage volumique 3D en entrée de chaque étape de remaillage, quelle que soit la zone (sous-domaine ou interface) considérée. Pour ce faire, les algorithmes proposés reposent sur des opérations booléennes (union et découpage) pour lesquelles le choix a été de limiter au maximum les calculs géométriques au profit d'approches par connectivité reposant uniquement sur les identifiants (numéros) globaux des nœuds. Par exemple, l'opération d'union booléenne de différents maillages se résume à fusionner les nœuds qui possèdent le même identifiant global. À condition de disposer d'une numérotation globale consistante qui garantit l'absence de nœuds doublons non détectés ou faussement détectés, cette stratégie est très robuste et rapide car aucun calcul géométrique et donc aucune tolérance n'est nécessaire. Cepenant, garantir cette numérotation consistante dans un contexte distribué est une tâche plus ou moins complexe suivant les approches retenues. Enfin, les échanges de portions de maillage entre processus s'effectuent, le cas échéant, à travers le protocole de communication MPI.

2.1 Détail de l'algorithme

Le principe de l'algorithme de type maître-esclave est détaillé dans l'algorithme 1.

Algorithm 1: Remaillage parallèle maître-esclave

Data: Décomposition initiale $(\Omega_i)_{1\leq i\leq N}$ avec connectivités nodales
Input: Carte de taille de maille, profondeur de peaux
Output: Nouveaux maillages adaptés préservant la géométrie initiale des domaines

1 **foreach** *domaine* **do**
2 Extraction des maillages de peau via des parcours de graphe;
3 Envoi des peaux au domaine maître par communication MPI ;
4 Le process maître reçoit les peaux ;
5 Le process maître unie les peaux via un assemblage booléen ;
6 Le process maître remaille la peau unifiée en figeant la topologie de l'interface décomposition de domaine et en figeant complètement l'interface entre la peau et les domaines esclaves ;
7 Le process maître décompose la peau unifiée remaillée ;
8 Le process maître envoie aux domaines esclaves les parties remaillées ;
9 **foreach** *domaine* **do**
10 Recevoir la peau remaillée en provenance du domaine maître par communication MPI ;
11 Remailler son intérieur avec les interfaces décomposition de domaines figées

L'avantage principal de cet algorithme est sa simplicité : d'une part la consistance de la numérotation globale des nœuds est directement assurée par le process maître, et d'autre part le parallélisme employé est facile à visualiser et ne se complexifie pas lorsque le nombre de domaines augmente.

L'extensibilité de cette approche semble par contre assez limitée. En effet, le coût de la phase de remaillage de la peau unifiée par le processus maître croit avec le nombre de domaines : c'est un goulet d'étranglement évident. Cette approche paraît réservée à un nombre faible de domaines ou à des problèmes où la taille de l'interface reste négligeable vis-à-vis de la taille des sous domaines.

2.2 Application sur cas académique

On considère un cas académique représentatif constitué par une aube de turbine multiperforée du type de celles équipant les compresseurs haute pression des turbomoteurs d'aéronefs, dont la géométrie d'entrée est présentée en figure 1 et pour lequel on génère un maillage initial comportant 2 millions de noeuds.

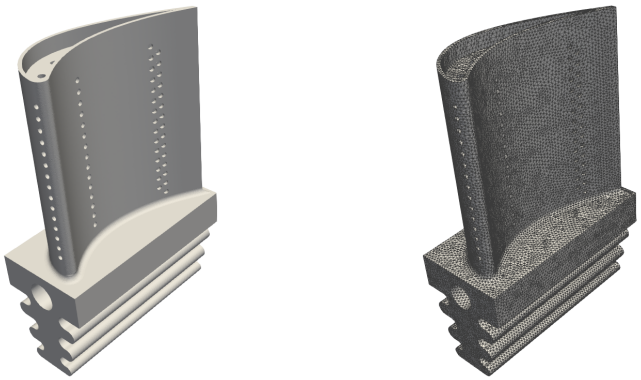

FIGURE 1 – Aube de turbine multiperforée : géométrie (gauche) et exemple maillage d'entrée à (100 000 nœuds, droite).

Afin d'évaluer les performances de la méthode, on applique successivement l'algorithme maître-esclave sur des configurations croissantes de sous-domaines et donc de processus (extensibilité forte), la configuration à 1 sous-domaine correspondant à un appel séquentiel de MMG3D. Dans tous les cas

considérés, le maillage initiale reste identique tout comme la consigne de remaillage appliquée, qui permet d'obtenir un maillage d'environ 9 millons de nœuds. L'étude a été réalisées sur un calculateur constitué de noeuds de type Intel Haswell à 24 cœurs et 128 Go de RAM reliés par un réseau de type Infiniband Mellanox, en affectant un coeur à chaque processus de remaillage.

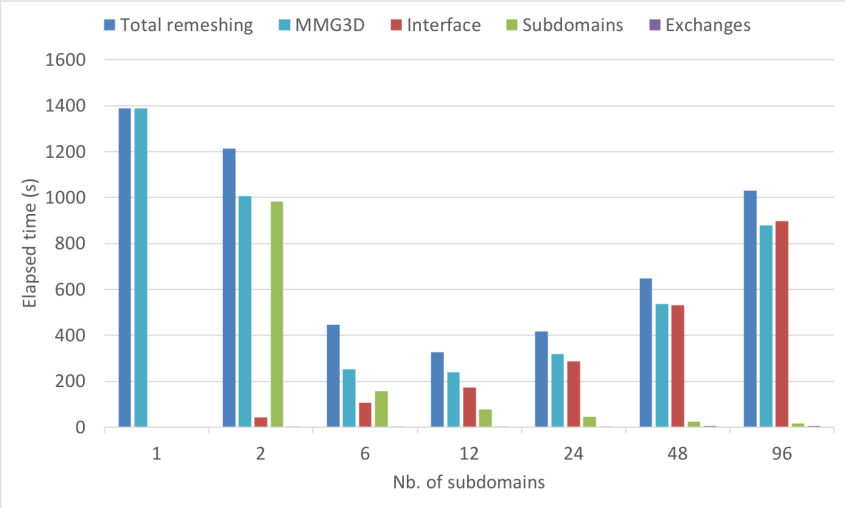

FIGURE 2 – Détails des temps apparents des différentes phases de l'algorithme 1 en fonction de la décomposition initiale : temps cumulé total (*Total Remeshing*), temps cumulé dans les appels à MMG3D (*MMG3D*), temps de remaillage de la zone d'interface volumique par le processus maître (*Interface*), temps de remaillage des sous-domaines en parallèle (*Subdomains*), temps d'échanges des zones d'interface avec le processus maître (*Exchanges*).

Le temps apparent global de l'algorithme est tracé sur la figure 2 pour les différentes configurations de sous-domaines. Sur le problème considéré, on observe que les configurations à 12 et à 24 processeurs sont optimales, le temps de calcul diminuant de 23 min en séquentiel à 6 min pour la configuration à 12 sous-domaines. On remarque également que l'exécution parallèle reste plus rapide que l'exécution séquentielle pour toutes les configurations testées, mais que le temps de calcul augmente au delà de 24 sous-domaines. En effet, en étudiant la répartition du temps apparent des différentes étapes de l'algorithme on observe que le coût de remaillage de la zone d'interface sur le premier processus (étape séquentielle, barre « Interface ») augmente considérablement par rapport au coût de remaillage de chaque sous-domaine (étape parallèle, barre « Subdomains »). Cela est du au fait que la taille de la zone devient trop importante par rapport à la taille de chaque sous-domaine, ainsi que par l'ensemble des contraintes de figeage appliquées en bord de zone pour permettre l'assemblage avec les sous-domaines après remaillage.

De manière générale, on note que le temps total de remaillage est dominé par le temps cumulé passé dans MMG3D (remaillage de l'interface et des sous-domaines, barre « MMG3D ») et que le coût des échanges MPI (barre « Exchanges ») reste négligeable.

3 Approche non intrusive de remaillage parallèle par coloriage

3.1 Détail de l'algorithme

Cet algorithme repose sur le coloriage du graphe de connectivité des sous-domaines en un ensemble minimal de couleurs de telle sorte que deux domaines partageant une interface commune se voient attribuer des couleurs différentes. Par la suite, l'algorithme itère successivement sur l'ensemble des couleurs. A chaque itération, les interfaces des sous-domaines correspondants à la couleur dite *active* qui n'ont pas été remaillées lors d'une précédente itération sont remaillés simultanément et imposées aux sous-domaines voisins de couleurs non parcourues. Une fois l'ensemble des interfaces remaillées, les sous-domaines sont remaillés indépendemment à leur tour en figeant les interfaces. Le principe de l'algorithme est détaillé dans l'algorithme 2.

L'avantage principal de cet algorithme est son extensibilité. En effet, le coût de la phase de remaillage est réparti sur l'ensemble des domaines. Il dépend de deux facteurs : la taille des domaines et le nombre de couleurs issu du coloriage du graphe de connectivité des domaines. Le nombre de couleurs croit faiblement avec le nombre de domaines et semble stagner en pratique. De plus, les dernières couleurs sont souvent inactives. À l'inverse, garantir la consistance de la numérotation globale des nœuds à chaque étape est plus complexe car plusieurs domaines remaillent en même temps. Également le parallélisme employé est plus ardu à visualiser et se complexifie lorsque le nombre de domaines augmente. Par exemple, un esclave peut toucher plusieurs fois la couleur active. Le choix des profondeurs de peaux internes et externes est empirique, il doit aboutir à un bon compromis entre la taille des maillages à remailler et le nombre d'étape d'optimisations qu'effectue le remailleur pour vérifier les contraintes géométriques imposées.

3.2 Application sur cas académique

Afin d'évaluer comparativement l'apport de ce second algorithme sur les performances globales de l'étape de remaillage, on réalise à nouveau une étude d'extensibilité avec la même configuration que pour l'algorithme maître-esclave.

La figure 2 illustre le temps de remaillage séquentiel et la répartition des différentes étapes pour un nombre de sous-domaines allant de 2 à 24. L'évolution du temps de remaillage apparent total est en

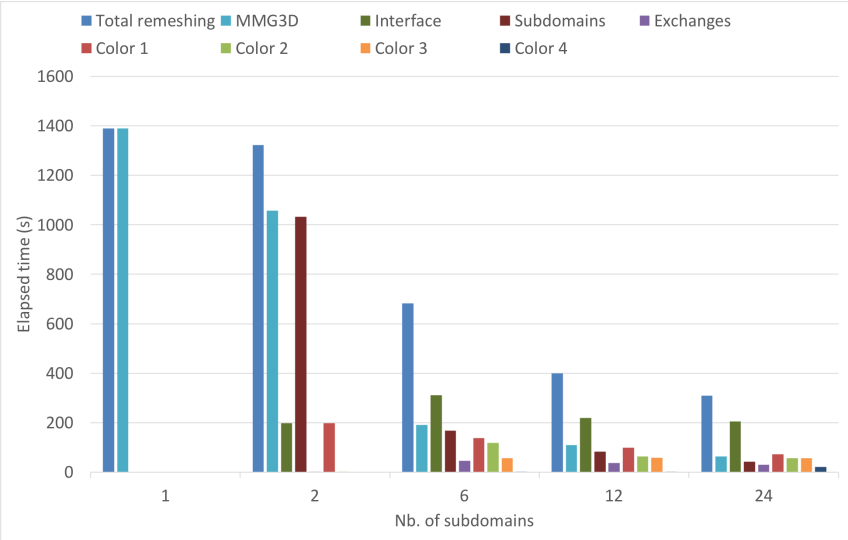

FIGURE 3 – Détails des temps apparents des différentes phases de l'algorithme 2 en fonction de la décomposition initiale : temps cumulé total (*Total Remeshing*), temps cumulé dans les appels à MMG3D (*MMG3D*), temps cumulé de remaillage des zones d'interface volumique en parallèle par couleurs (*Interface*), temps de remaillage des sous-domaines en parallèle (*Subdomains*), temps d'échanges des zones d'interface entre processus (*Exchanges*), temps associé au remaillage des différentes couleurs (*Color #i*).

constante décroissance avec l'augmentation du nombre de processus. En outre, le temps cumulé de remaillage des zones d'interface est stable avec l'augmentation du nombre de sous-domaines tandis que le remaillage cumulé des sous-domaine décroit de manière similaire à ce qui a été observé sur l'algorithme de type maître-esclave. On remarque que le temps associé aux remaillages des différentes couleurs est également stable bien que légèrement supérieur pour la première couleur, ce qui peut s'expliquer par un plus grand nombre d'interfaces à remailler. Enfin, les temps d'échanges sont très faibles quoique plus importants que pour le précédent algorithme, ce qui peut s'expliquer par des temps d'attente du fait de la synchronisation des différents processus à chaque étape impliquant des remaillage de zones d'interface.

Les performances préliminaires observées sont particulièrement prometteuses dans l'optique de disposer d'un algorithme extensible. Cependant, l'étude nécessite à présent d'être élargie à un plus grand nombre de processus. De même, l'influence de l'épaisseur (nombre d'éléments) des zones volumiques d'interfaces mérite d'être étudiée dans l'optique d'améliorer le temps de retour en jouant sur le compromis entre les tailles de maillage en jeu et la complexité du processus de remaillage résultant. En effet, il est attendu que cette complexité augmente avec la diminution de l'épaisseur de la zone car les

contraintes de figeage peuvent alors entraîner un nombre significatif d'itérations d'optimisation pour améliorer la qualité des éléments.

4 Conclusion

Cet article a présenté deux algorithmes à même d'effectuer une phase d'adaptation de maillage à partir d'un maillage initialement découpé en sous-domaines, en conservant la topologie des interfaces entre sous-domaines. Afin d'en évaluer les performances, une étude d'extensibilité forte a été menée sur la base d'une structure industrielle simplifiée de type aube multiperforé. Cette étude montre que l'algorithme de type maître-esclave, relativement simple à mettre en œuvre, semble intéressant losque appliqué à un faible nombre de sous-domaines mais présente des limites évidentes avec l'augmentation de ceux-ci. Le second algorithme de remaillage par couleur, plus complexe, semble cependant présenter une meilleure extensibilité avec l'augmentation du nombre de sous-domaines et ses performances seront étudiées sur des configurations à plus grand nombre de sous-domaines. Les derniers résultats seront montrés lors de la conférence.

Références

[1] Nikos Chrisochoides. Parallel mesh generation. In *Numerical solution of partial differential equations on parallel computers*, pages 237–264. Springer, 2006.

[2] Philippe P. Pébay, Michael B. Stephenson, Leslie A. Fortier, Steven J. Owen, and Darryl J. Melander. pcamal : An embarrassingly parallel hexahedral mesh generator*. In Michael L. Brewer and David Marcum, editors, *Proceedings of the 16th International Meshing Roundtable*, pages 269–284, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[3] Adrien Loseille, Victorien Menier, and Frédéric Alauzet. Parallel generation of large-size adapted meshes. *Procedia Engineering*, 124(57 – 69), 2015.

[4] T. Coupez, H. Digonnet, and R. Ducloux. Parallel meshing and remeshing. *Applied Mathematical Modelling*, 25(2) :153 – 175, 2000. Dynamic load balancing of mesh-based applications on parallel.

[5] Cédric Lachat. *Design and validation of distributed-memory, parallel remeshing algorithms based on asequential remesher*. Theses, Université Nice Sophia Antipolis, December 2013.

[6] Arnaud Bardoux. Remaillage parallèle pour le calcul haute performance. Master's thesis, Université de strasbourg, August 2016.

[7] H L. De Cougny and Mark Shephard. Parallel refinement and coarsening of tetrahedral meshes. *International Journal for Numerical Methods in Engineering - INT J NUMER METHOD ENG*, 46 :1101–1125, 11 1999.

[8] Steven J. Owen, Ryan M. Shih, and Corey D. Ernst. A template-based approach for parallel hexahedral two-refinement. *Computer-Aided Design*, 85 :34 – 52, 2017. 24th International Meshing Roundtable Special Issue : Advances in Mesh Generation.

[9] Steven J. Owen, Judith A. Brown, Corey D. Ernst, Hojun Lim, and Kevin N. Long. Hexahedral mesh generation for computational materials modeling. *Procedia Engineering*, 203 :167 – 179, 2017. 26th International Meshing Roundtable, IMR26, 18-21 September 2017, Barcelona, Spain.

[10] Sandeep Menon, Kyle G. Mooney, K.G. Stapf, and David P. Schmidt. Parallel adaptive simplical re-meshing for deforming domain cfd computations. *Journal of Computational Physics*, 298 :62 – 78, 2015.

[11] Christophe Bovet, Augustin Parret-Fréaud, Nicole Spillane, and Pierre Gosselet. Adaptive multipreconditioned FETI : Scalability results and robustness assessment. *Computers & Structures*, 193 :1–20, December 2017.

[12] Transvalor S.A. *Z-set 8.7 user manual*, 2018.

[13] Charles Dapogny, Cécile Dobrzynski, and Pascal Frey. Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems. Technical report, March 2013.

Algorithm 2: Remaillage parallèle par coloriage

Data: Décomposition initiale $(\Omega_i)_{1\leq i\leq N}$ avec les connectivités nodales (2-à-2).
Input: Carte de taille de maille, profondeur de peau interne et profondeur de peau externe
Output: Nouveaux maillages remaillés préservant la géométrie initiale des domaines

begin Initialisation en parallèle
    Création du graphe de connectivité des domaines ;
    Coloriage de ce graphe via un algorithme glouton de telle sorte qu'aucun domaine n'ait la même couleur que ses voisins;
    Tri des couleurs en fonction du nombre de domaines qui les peuplent ;
foreach *couleur* **do**
    Passage de *couleur* au statut actif ;
    foreach *domaine* Ω **do**
        Calcul du statut maître, esclave, inactif.
            — Ω est maître s'il est de la couleur active et si l'un de ses voisins à une couleur non traitée
            — Ω est esclave si sa couleur est non traitée et si l'un de ses voisins est de la couleur active
            — Ω est inactif s'il n'est ni maître ni esclave
    forall *domaines esclaves* **do**
        Extraire les peaux en contact avec un domaine voisin actif;
        Communiquer ces peaux aux voisins actifs ;
    forall *domaines maîtres* **do**
        Extraire les peaux internes en contact avec un domaine voisin esclave;
        Recevoir les peaux des voisins esclaves et faire l'assemblage booléen;
        Remailler la peau unifiée en figeant :
            — la topologie de l'interface décomposition de domaine
            — complètement l'interface entre la peau et les domaines esclaves
            — complètement l'interface entre la peau et le domaine actif
        Décomposer la peau remaillée;
        Faire l'assemblage entre la peau interne et l'intérieur du domaine actif;
        Renvoyer les peaux remaillées aux domaines voisins esclaves ;
    forall *domaines esclaves* **do**
        Recevoir les peaux remaillées par les voisins actifs ;
        Faire l'assemblage booléen;
forall *domaines* Ω **do**
    Remailler son intérieur avec les peaux figées