Maillage de connexion via une transformation en ondelettes et interpolation locale par des RBF

Mesh connection with Wavelet transform and RBF local interpolation.

PHAN Anh-Cang

Laboratoire des Sciences de l’Information et des Systèmes (LSIS, UMR CNRS 7296)
Ecole doctorale mathématiques et informatique de Marseille (ED 184)
Campus de Luminy, case postale 925, 13288 Marseille cedex 9, France.
Anh-cang.PHAN@univ-amu.fr

RÉSUMÉ. Nous présentons une méthode de connexion de maillages entre deux domaines maillés de différents niveaux de résolution. Le maillage de liaison est créé à l'aide d'une interpolation avec des fonctions de base radiale locale combinée à une transformation en ondelettes B-spline. Cela garantit que la continuité entre les deux zones est préservée et que le maillage de connexion est modifié progressivement pour tenir compte des résolutions différentes entre les deux zones. Cette méthode pourrait être étendue à des applications liées au remplissage de trous, à la couture de maillages de subdivision et à la composition d'objets 3D.

ABSTRACT. We introduce a connection method between two mesh areas at different resolutions. The connecting mesh is based on a local interpolation with radial basis functions and a Lifted B-spline wavelet transform. This ensures that the continuity between these mesh areas is preserved and the connecting mesh is modified gradually in resolution between coarse and fine areas. This method could be expanded to applications related to filling holes, pasting subdivision meshes and composition of 3D objects.

MOTS-CLÉS : transformée en ondelettes; maillage triangulaire; maillage de connexion; ondelettes B-splines.

KEYWORDS: Wavelet transform; triangular meshes; mesh connection; B-spline wavelets.
1. Introduction

Complex shapes are generated by a set of assembled patches or separate mesh areas which could be at different resolution levels. Their surfaces could appear cracks, gaps or holes. It leads to a problem about missing surfaces in Computer Aided Design (CAD) applications such as unwanted object shapes, cracks along the boundaries of the patches, or even incorrect connections between meshes. To overcome these problems, cracks must be removed. Our work aims at constructing a high quality connecting mesh between two selected mesh areas of a model using the Radial Basis Function (RBF) local interpolation and the wavelet transform so that we can preserve the continuity between these selected mesh areas to produce a smooth surface.

2. Related work

In many cases, subdivision of the whole input mesh is not necessary, only some areas that need to be subdivided to make them smoother. This is important to reduce the unnecessary subdivision process, and save the refinement time, the storage space. Some research (Noor Asma Husain et al., 2011), (Pakdel et al., 2004), (Pakdel et al., 2005), (Husain et al., 2010) related to the incremental subdivision method with Butterfly, Loop and Catmull-Clark schemes. The main work of these methods is to generate a smooth surface by refining only some selected areas of a mesh with the same a subdivision scheme, and then removing cracks with simple triangulation. However, this simple triangulation has some undesired side-effects. It changes the connectivity, valence of odd vertices, and the surrounding areas. This not only alters the limit subdivision surface, but also reduces its smoothness. Moreover, it produces high valence vertices which lead to long faces. They create ripple effects on the subdivision surface. In addition, there have been also research works relevant to connecting and pasting meshes in (Zhang et al., 2010), (Fu et al., 2004), (Barequet et al., 1995).

These methods consist in connecting the meshes of a surface at the same resolution level which adopt various criteria to compute the planar shape from a 3D surface patch by minimizing their differences. The approaches are computationally expensive and memory consuming. Therefore, we proposed a new mesh connection method without needing to handle cracks, modify original boundaries of mesh areas of a model, and subdivide the closest faces around the original boundaries.

3. Background

3.1. Wavelet multiresolution representation of curves and surfaces

Wavelet has been applied successfully to a wide variety of applications in modeling environment (Mallat, 1998), (Olsen et al., 2008), (Bertram, 2002). Wavelet tool supports the multiresolution representations of curves and surfaces (Lounsbery et al., 1997), (Eck et al., 1995); curve smoothing at different resolution levels (Olsen et al., 2008), (Guskov et al., 1999); overall form edition of a curve while preserving its details (Suciati et al., 2009); and curve approximation (Stollnitz et al., 1996), (Khodakovsky et al., 2000). Recently, multiresolution (MR) settings based on wavelets have
been proposed for many curve and surface subdivision types: B-splines (Bertram et al., 2004), Doo subdivision (Samavati et al., 2002), and Loop (Bertram, 2004). The wavelet transform allows a decomposition of curves and surfaces at different resolutions while maintaining geometric details. Wavelet analysis provides a set of tools to represent functions hierarchically (Stollnitz et al., 1995). The coarse scaling function represents coarse curves or surfaces, encodes an approximation of the function. The wavelet function represents the difference between coarse and fine curves or surfaces, and encodes the missing details. These tools can efficiently facilitate geometric modeling operations.

The combination of B-splines and wavelets leads to the idea of B-spline wavelets (Bertram et al., 2004). B-spline wavelets form a hierarchical basis for the space of B-spline curves and surfaces in which every object has a unique representation. Taking advantage of the lifting scheme, Lifted B-spline wavelets (Sweldens et al., 1996) are a fast computational tool for multiresolution analysis with a computational complexity linear in the number of control points for a given B-spline curve. They allow multiresolution analysis of B-spline curves, representing a curve at multiple resolution levels, editing curves, etc. The Lifted B-spline wavelet transform includes two phases: the forward and the backward B-spline wavelet transforms. From a fine curve at the decomposition level \( j+1 \), \( C^{j+1} \), the forward B-spline wavelet transform decomposes \( C^{j+1} \) into a sequence of coarser approximations of the curve, \( C^k \) (\( 0 \leq k \leq j \)), and detail (error) vectors. The detail vectors are a set of wavelet coefficients containing the geometric differences with respect to the finer levels. The backward B-spline wavelet transform can be used to reconstruct fine resolution curves from a coarse curve and detail vectors. Given a curve at the decomposition level \( j \), \( C^j \), the backward B-spline wavelet transform synthesis \( C^j \) and the detail vectors into the finer curves, \( C^k \) (\( k \geq j + 1 \)).

In our approach, we apply the Lifted B-spline wavelet transform for multiresolution analysis of discrete boundary curves. We can expand to apply the other wavelets for the multiresolution representations of the discrete boundary curves.

### 3.2. Radial Basis Function (RBF) local interpolation

To extrapolate local frames (tangents, curvatures) between two meshes we also need a local interpolation method. We choose Radial Basis Functions in order to operate on continuous spaces. Practical solutions on large point sets involve the local interpolation methods to the surface reconstruction such as RBF local interpolations (Casciola et al., 2005) (Branch et al., 2006), fast RBF methods (Carr et al., 2001) (Turk et al., 2002), and plane fitting methods (Hoppe et al., 1992). These methods consider subsets of nearest neighboring points at a time to compute the local interpolation functions on a surface as shown in Fig. 1.

The basic idea of the RBF local interpolation is to find an implicit function \( s(x) \) by using a set of control points corresponding to a set of the signed-
distance function values. Given a set of $N$ distinct control points (or centers) $X = \{x_k = (x_k, y_k, z_k)\}_{k=1}^N \subset \mathbb{R}^3$ corresponding to a set of distance function values $f = \{f_k\}_{k=1}^N \subset \mathbb{R}$. For each $x_k \in X, k = 1, \ldots, N$, we determine a subset of nearest neighboring points $X_k = \{x_k\} \cup \{x_j \in X; x_j \in \text{Neighbors}(x_k)\}$ as shown in Fig. 1, where $\text{Neighbors}(x_k)$ is the nearest neighboring points of $x_k$. Let $I_k$ be the set of indexes of $X_k$ and $N_k = |X_k|$ be the number of the nearest neighbors of $x_k$.

We want to approximate the signed-distance function $f(x)$ by an interpolation $s(x)$. An RBF local interpolation function $s : \mathbb{R}^3 \rightarrow \mathbb{R}$ on $X_k$ is determined by:

$$s_k(x) = \sum_{j \in I_k} \lambda_j \phi(||x - x_j||) \quad [1]$$

It requires satisfying the local interpolation constraints on $X_k$.

$$s_k(x_i) = f_i = \sum_{j \in I_k} \lambda_j \phi(||x_i - x_j||), i \in I_k \quad [2]$$

where $\phi(||x - x_j||)$ are the radial basis functions (RBFs); the points $x_j$ are referred to as the control points of the RBF and are also the nearest neighboring points of $x_k$; $\lambda_j$ are the weights of the RBFs; $||x||$ is the Euclidean norm. The basis function is normally chosen from the families of the spline functions of smoothing, such as the biharmonic $\phi(r) = r$ ($r = ||x - x_j||$), the triharmonic $\phi(r) = r^3$, the Gaussian $\phi(r) = e^{-(\frac{r}{h})^2}$, and so on. In our approach, we choose the Gaussian basis function and $h$ to be the average distance from $x_k$ to the control points $x_j$. Combining eq. 1 and eq. 2 leads to:
$$\begin{bmatrix} \phi_{1,1} & \cdots & \phi_{1,N_k} \\ \vdots & \ddots & \vdots \\ \phi_{N_k,1} & \cdots & \phi_{N_k,N_k} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_{N_k} \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_{N_k} \end{bmatrix}$$

[3]

Equation 3 may be re-written in the matrix form:

$$\Phi_{X_k} \Lambda_{X_k} = F_{X_k}$$

[4]

where $\phi_{i,j} = \phi(||x_i - x_j||)$, $\Phi_{X_k} = (\phi_{i,j})$ with $i, j \in I_k$, $\Lambda_{X_k} = (\lambda_1, \lambda_2, ..., \lambda_{N_k})^T$, $F_{X_k} = (f_1, ..., f_{N_k})^T$. After solving the linear system of eq. 4 to compute the unknown weights $\Lambda$ of the basis functions, a data set is simply reconstructed by computing the local interpolation function values $s_k(x)$ at $x \in X_k$ using eq. 1.

4. Mesh Connection method overview

4.1. Basic concepts and topology representation

Let $M_1^l$ and $M_2^k$ be two meshes and $p_i$, $q_i$ their vertices. An edge connecting $p_i$ to $q_i$ is denoted $e_i$ or $p_i q_i$. An edge is usually shared by two faces. If it is shared by only one, it corresponds to a boundary edge and its end vertices are called boundary vertices. We need to construct a connecting mesh CM between two meshes $M_1^l$ and $M_2^k$ so that it can preserve the continuity between them as illustrated in Fig. 2.

Figure 2. Basic concepts and topology representation related to the algorithm.

Let us introduce the notations used in the following:

- $s$ : the number of intermediate discrete curves (also called the number of new created boundary curves) of CM created between $M_1^l$ and $M_2^k$ (see Fig. 4). It is a user defined threshold and controls the resolution of CM.
- $j$ : the order number of the decomposition step to create intermediate discrete curves, also called decomposition level. Since two boundary curves between $M_1^l$ and $M_2^k$ will be created in each decomposition level $j$, $1 \leq j \leq \frac{s}{2}$. 
– \( C_1^j \) and \( C_2^j \) : the two boundary curves of \( \text{CM} \) at level \( j \) which approximate the two original boundary curves \( C_1^0 \) and \( C_2^0 \) of meshes \( M_1^j \) and \( M_2^j \).

– \( N(C_1^j) \) : the number of vertices of boundary curve \( C_1^j \) at level \( j \). It corresponds to the density of vertices of the boundary curve \( C_1^j \).

– \( p_i^j \) : the vertices \( i \) on boundary curves \( C_1^j \) and \( C_2^j \). \( (p_i^0 = p_i \) and \( q_i^0 = q_i) \)

– \( L_i^j \) : the list of the boundary vertex pairs \( (p_i^{j-1}, q_k^{j-1}) \). That means, the boundary vertices \( p_i^{j-1} \in C_1^{j-1} \) are paired with the boundary vertices \( q_k^{j-1} \in C_2^{j-1} \).

– \( L_q^j \) : the list of the boundary vertex pairs \( (q_k^{j-1}, p_i^{j-1}) \). That means, the boundary vertices \( q_k^{j-1} \in C_2^{j-1} \) are paired with the boundary vertices \( p_i^{j-1} \in C_1^{j-1} \).

### 4.2. Algorithm overview

The idea of our algorithm is to create the new boundary curves \( C_1^j \) and \( C_2^j \) between \( M_1^j \) and \( M_2^j \) based on the previously created boundary curves \( C_1^{j-1} \) and \( C_2^{j-1} \) using the Lifted B-spline wavelet transform and the RBF local interpolation. Then, we connect each new boundary curve \( C_1^j \) to \( C_1^{j-1} \), and \( C_2^j \) to \( C_2^{j-1} \). \( C_1^j \) is created in a direction from \( C_1^{j-1} \) to \( C_2^{j-1} \) and conversely for \( C_2^j \). The algorithm (Fig. 3) consists of the following main steps detailed in the next section:

– **Step 1.** Boundary detection : read the input geometry model of two subdivided meshes \( M_1^j \) and \( M_2^j \) at different resolution levels. Detect and mark boundary vertices of two boundaries \( C_1^0 \) and \( C_2^0 \) in \( M_1^j \) and \( M_2^j \) respectively.

– **Step 2.** Boundary vertex pairs and boundary curve creation : pair the boundary vertices of two boundary curves \( C_1^{j-1} \), \( C_2^{j-1} \) based on the distance between them. If the distance between them is very narrow, we go to Step 3 to connect the boundary curve pair \( (C_1^{j-1}, C_2^{j-1}) \). In contrast, we perform a boundary curve creation to create new boundary curves \( C_1^j \), \( C_2^j \). It first produces vertices of two new boundary curves from the paired boundary vertices by the linear interpolation method and then project them onto the surface CM using the RBF local interpolation method. It finally refines or coarsens these new boundary curves applying the wavelet transforms and vertex insertion and deletion operations.

– **Step 3.** Boundary curve connection : perform a boundary triangulation for each boundary curve pair \( (C_1^{j-1}, C_1^j) \) and \( (C_2^{j-1}, C_2^j) \).

– **Step 4.** Repeat steps 2 through 3 until two mesh areas \( M_1^j \) and \( M_2^j \) has been connected or patched by all newly created triangles.

### 5. Boundary curve creation and connection

The idea is to create two new boundary curves \( C_1^j \) and \( C_2^j \) from the paired vertices in each level \( j \). Paired vertices are obtained by shortest distances between vertices of each boundary. That is, new boundary vertices \( p_i^j \in C_1^j \) and \( q_k^j \in C_2^j \) are created by boundary vertex pairs \( (p_i^{j-1}, q_k^{j-1}) \in L_1^j \) and \( (q_k^{j-1}, p_i^{j-1}) \in L_2^j \) respectively. A new
boundary curve $C_{1}^{j}$ are created in a direction from $C_{1}^{j-1}$ to $C_{2}^{j-1}$ and a new boundary curve $C_{2}^{j}$ are created in a direction from $C_{2}^{j-1}$ to $C_{1}^{j-1}$ as shown in Fig. 4.

We assume $N(C_{0}^{0}) \leq N(C_{2}^{0})$ and let the density of vertices of the two boundary curves $C_{1}^{j}$ and $C_{2}^{j}$ be two functions $N(C_{1}^{j})$ and $N(C_{2}^{j})$ defined by :

\[ N(C_{1}^{j}) = N(C_{0}^{0}) + \frac{j}{s+1} [N(C_{2}^{0}) - N(C_{1}^{0})] \]
\[ N(C_{2}^{j}) = N(C_{2}^{0}) - \frac{j}{s+1} [N(C_{2}^{0}) - N(C_{1}^{0})] \]  \[5\]

The boundary curve creation is computed in three phases as follows :

1) Phase 1 : Create vertices of two new boundary curves with the linear interpolation.
- Create vertices of the discrete boundary curve $C^j_1$ in a direction from $C^j_{1-1}$ to $C^j_{2-1}$ (see Fig. 4): for each boundary vertex pair $(p^{j-1}_i, q^{j-1}_k) \in L^j_1$, we apply the linear interpolation equation eq. 6 to create new boundary vertices $p^j_i \in C^j_1$.

$$p^j_i = p^{j-1}_i + \frac{j}{s+1}(q^{j-1}_k - p^{j-1}_i)$$

[6]

Where $i$ is the index of boundary vertices of $C^j_1$, $1 \leq i \leq N(C^j_{1-1})$ and $k$ the index of boundary vertices of $C^j_{2-1}$, $1 \leq k \leq N(C^j_{2-1})$.

- In the same way, we create the new boundary vertices $q^j_k \in C^j_2$ by eq. 7.

$$q^j_k = q^{j-1}_k + \frac{j}{s+1}(p^{j-1}_i - q^{j-1}_k)$$

[7]

Where $k$ is the index of a boundary vertex on $C^j_2$, $1 \leq k \leq N(C^j_{2-1})$ and $i$ the index of the boundary vertex on $C^j_{1-1}$, $1 \leq i \leq N(C^j_{1-1})$.

2) Phase 2: Project created boundary vertices onto the surface CM using the RBF local interpolation. The goal is to improve the result surface CM after applying phase 1. Since new boundary vertices $p^j_i$ are created by the linear interpolation in phase 1, they can lie on a flat surface $H$ producing a flat surface CM as shown in Fig. 5a).

![Figure 5](image1.png)

**Figure 5.** The connecting mesh CM is created with and without using the RBF local interpolation.

New created boundary vertices $p^j_i$ and $q^j_k$ of curves $C^j_1$ and $C^j_2$ do not lie on the surface CM as wanted because we use linear interpolation in phase 1 without considering informations of surface curvature (see Fig. 5b and Fig. 6). As a result, the produced connecting mesh will not ensure the continuity between two meshes. In order to preserve the continuity between two mesh areas at the joined part, and avoid creating boundary vertices not on CM, we apply a method of the implicit surface reconstruction with RBF local interpolation. That is, we first apply phase 1, and then project the created boundary vertices onto surface CM along surface normals using RBF local interpolation to fit RBFs to surface CM as shown in Fig. 6.

![Figure 6](image2.png)

**Figure 6.** Projection the vertices of two new boundary curves onto the surface CM.
local interpolation function values $s_k$ for these vertices (using eq. 1). Then, we project them onto surface CM with the projection distances $s_k$ along normals of vertices $q^{i-1}_k$ and update the vertices of boundary curve $C^j_2$ as their projected vertices. Similarly, we also apply the same for the created vertices $p^{1}_i \in C^j_1$.

In general, it is not necessary to use such a large number of off-surface vertices. Theoretically, a single off-surface vertex might be sufficient for the surface reconstruction. Therefore, we propose to introduce a set of local control vertices $Q_k$ corresponding to each vertex $q^i_k$ for the local reconstruction of the RBF interpolation function as follows:

- $Q_k = Q_1 \cup Q_2$.
- $Q_1 = \{q^{i-1}_k\} \cup \{q_1 \in \text{Neighbors}(q^{i-1}_k)\}$. $Q_1$ is the local neighbors of $q^{i-1}_k$ referred to as on-surface vertices.
- $Q_2 = \{q_2 = q_1 + d \cdot n(q_1); q_1 \in Q_1\}$. $Q_2$ is referred to as off-surface vertices.

A set of local control vertices $Q_k$ corresponding to a set of distance function values:

$f(q) = 0$ for $q \in Q_1$ and $f(q) = d$ for $q \in Q_2$.

3) Phase 3: Refine or coarsen these new boundary curves with the wavelet transforms. Since the number of vertices of $C^j_1$ and $C^j_2$ is now $N(C^{j-1}_1)$ and $N(C^{j-1}_2)$ respectively, we need to increase and reduce the resolutions of $C^j_1$ and $C^j_2$ must be $N(C^j_1)$ and $N(C^j_2)$. We apply the Lifted B-spline wavelet transform for the multiresolution representations of the boundary curve $C^j_1$ and $C^j_2$ to refine the boundary curve $C^j_1$, coarsen the boundary curve $C^j_2$. Then, we perform the vertex insertion or deletion operations to control the density of vertices of $C^j_1$ and $C^j_2$. The result is to obtain two boundary curves $C^j_1$ with the density of vertices $N(C^j_1)$ and $C^j_2$ with the density of vertices $N(C^j_2)$. This ensures that the connecting mesh CM is changed gradually in resolution from one area of the surface to another area.

After creating two boundary curves $C^j_1$ and $C^j_2$, we connect each new boundary curve to each previously created boundary curve: $C^{j-1}_1$ to $C^j_1$ and $C^{j-1}_2$ to $C^j_2$ based on the method of stitching the matching borders proposed by (Barequet et al., 1995). The basic idea is the implementation of the boundary triangulation based on the distance between boundary vertices. We consider the distance between three adjacent vertices of two boundaries before connecting them together to create a triangular face (see Fig. 5). This process terminates when we reach the last vertices of both boundaries.

Figure 7. Figure shows boundary curve connection.
6. Results

Our algorithm has been implemented on Matlab. All experimental results in this paper were obtained on a PC 2.27GHz CPU Core i5 with 3GB Ram. We have applied our algorithm to various types of 3D objects. From two original coarse mesh areas $M_0^1$ and $M_0^2$ of a model, we apply Loop scheme at level 2 for $M_0^1$ and Butterfly scheme at level 1 for $M_0^2$ to obtain two subdivided mesh areas $M_2^1$ and $M_2^2$. We then apply our CM2D-RBFW algorithm for connecting them. Results are shown as in Fig. 8 and Fig. 9 which illustrate the change of different values of $s$ and $d$. They show images of resulting meshes (the top row), surfaces (the middle row), and the Gaussian curvatures (the bottom row).

The number of intermediate discrete curves $s$ and the distance function value $d$ are parameters defined by user. In our approach, the initial constraint for $d$ in $[−\frac{t}{2}, \frac{t}{2}]$, where $t$ is chosen to be the average distance between $p_i \in C_1^0$, $q_k \in C_2^0$ and their local neighbors. That is, to create off-surface vertices, on-surface vertices are projected at most distance $\frac{t}{2}$. This assumption is naturally satisfied in almost all practical applications (Alexa et al., 2003). This figures show the reconstruction of the Cylinder and the Tiger with and without validation of values $d$. The resulting surfaces are deformed when taking invalid projection distance values $d = −0.009; 0.005; 0.001$ for the Cylinder and $d = −0.002; 0.005$ for the Tiger. The surface of the Cylinder and the Tiger is visually smooth when choosing a valid projection distance value $d = 0.0001$ and $d = 0.0002$ respectively.

The experimental results show that our approach satisfies the constraint of the continuity between mesh areas and produces high quality and smooth connecting meshes.

7. Conclusion

We introduced a new incremental simple and efficient method which generates a high quality connecting mesh and a smooth surface. The surface is changed gradually in resolution from one area of the surface to another area. The algorithm additionally doesn’t subdivide the closest faces around selection areas of a mesh model. The results of our method show high quality connecting meshes and smooth surfaces. In addition, smooth surfaces generated by our method have proper connectivity and geometry. Our method could be extended to applications related to filling holes, pasting subdivision meshes and incremental subdivisions. We are now working on a scheme to subdivide the connecting mesh when the two meshes are subdivided again so that the connection remains valid.

8. Bibliographie


Figure 8. The connecting meshes are produced for two mesh areas of the Cylinder model with $s = 4$ and $d = -0.009; -0.002; 0.0001; 0.005; 0.01$.

Figure 9. The connecting meshes are produced for two mesh areas of the Tiger model with $s = 3$ and $d = -0.002; 0.0002; 0.005$


ANNEXE POUR LE SERVICE FABRICATION
A FOURNIR PAR LES AUTEURS AVEC UN EXEMPLAIRE PAPIER
DE LEUR ARTICLE ET LE COPYRIGHT SIGNE PAR COURRIER
LE FICHIER PDF CORRESPONDANT SERA ENVOYE PAR E-MAIL

1. ARTICLE POUR LA REVUE :

2. AUTEURS :
   PHAN Anh-Cang

3. TITRE DE L’ARTICLE :
   Maillage de connexion via une transformation en ondelettes et interpolation locale par des RBF

4. TITRE ABRÉGÉ POUR LE HAUT DE PAGE MOINS DE 40 SIGNES :
   Maillage de connexions

5. DATE DE CETTE VERSION :
   19 mai 2012

6. COORDONNÉES DES AUTEURS :
   – adresse postale :
     Laboratoire des Sciences de l’Information et des Systèmes (LSIS, UMR CNRS 7296)
     Ecole doctorale mathématiques et informatique de Marseille (ED 184)
     Campus de Luminy, case postale 925, 13288 Marseille cedex 9, France.
     Anh-cang.PHAN@univ-amu.fr
   – téléphone : +33 605589077
   – e-mail : Anh-cang.PHAN@univ-amu.fr

7. LOGICIEL UTILISÉ POUR LA PRÉPARATION DE CET ARTICLE :
   LATEX, avec le fichier de style article-hermes.cls, version 1.2 du 03/03/2005.

8. FORMULAIRE DE COPYRIGHT :
   Retourner le formulaire de copyright signé par les auteurs, téléchargé sur :
   http://www.revuesonline.com