

# PJE09: Dépliage de structures gonflables

**ZAMBAITI Mario Sergio**

Encadré par: E. Monteiro



# SOMMAIRE

1. Sujet du PJE09
2. Etat de l'art
3. Démarche de dépliage
4. Analyse de la démarche
5. Analyse des résultats
6. Conclusions
7. Questions

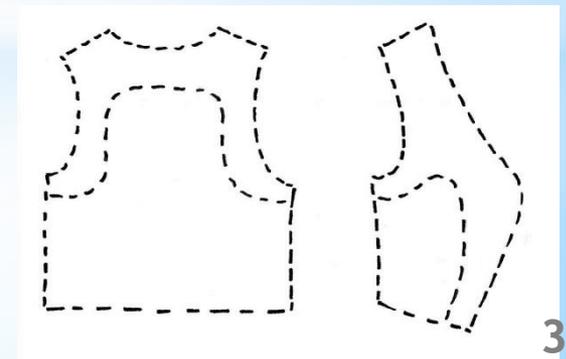
# Sujet du PJE09

Dépliage de structures gonflables:

- \* Montgolfières, Vêtements, ballon aérostatiques, jeux gonflables pour enfants.



- \* Avoir les formes des laizes en 2D pour reconstruire une structure 3D après assemblage et gonflage.



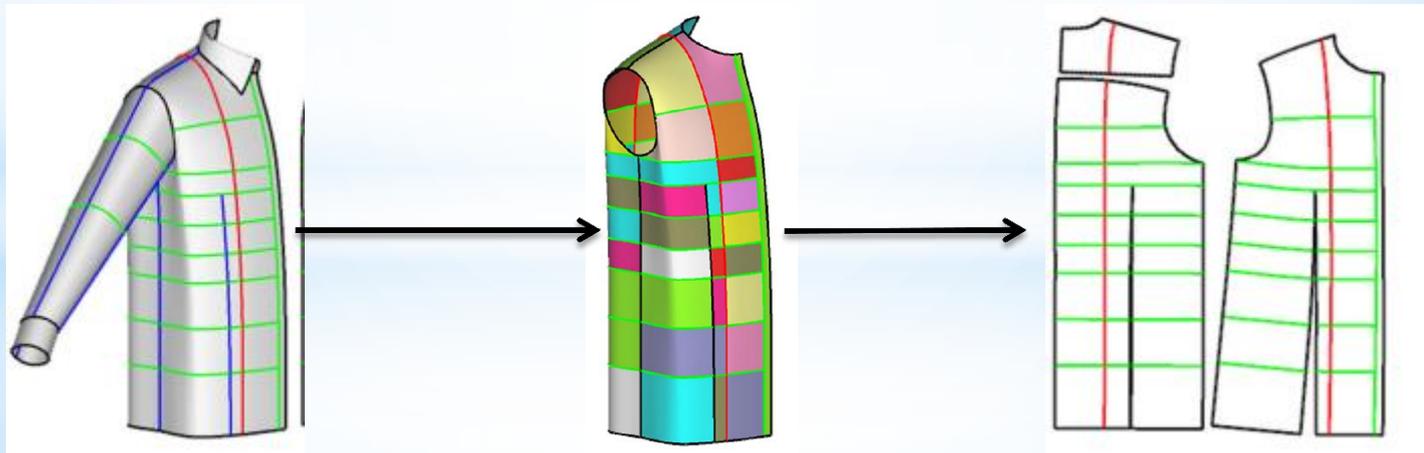
# Etat de l'art

WireWarping: A fast surface flattening approach with length-preserved feature curves[1]

- \* Conservation des longueurs de la structure 3D
- \* Le concepteur choisi les lignes à conserver:

- \* Lignes de contour
- \* Lignes de coupe
- \* Lignes de forme

→ Décomposition des laizes en patches



[1]:WireWarping: A fast surface flattening approach with length-preserved feature curves Charlie C.L. Wang, Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong. 28 April 2007

# Etat de l'art

\* Division des contours des patches en nœuds  $q_j^i$

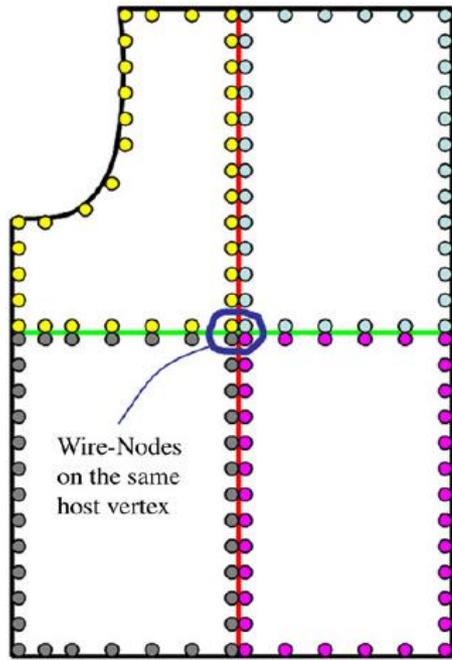
\* Repère des vertex  $v(q_j^i)$

\* Calcul des données 3D entre noeuds

\* Longueurs 3D :  $l_i$   $\longrightarrow$  conservées

$$l_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2}$$

\* Angles 3D :  $\alpha_i$   $\longrightarrow$  angles 2D :  $\theta_i$



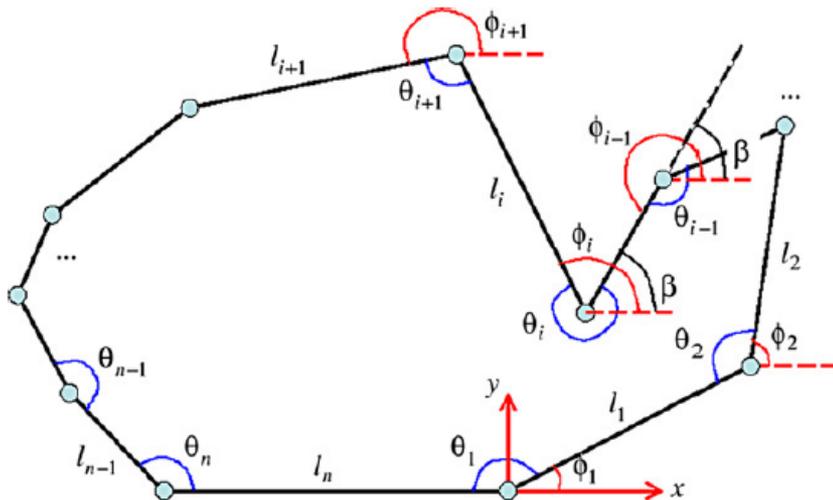
$$\alpha_i = \cos^{-1} \frac{((x_{i+1} - x_i), (y_{i+1} - y_i), (z_{i+1} - z_i)) \times ((x_i - x_{i-1}), (y_i - y_{i-1}), (z_i - z_{i-1}))}{l_i + l_{i-1}}$$

# Etat de l'art

\* Le dépliage = optimisation des angles avec  $l_i = \text{constants}$

$$\min \sum_{i=1}^n 1/2(\theta_i - \alpha_i)^2$$

$$n\pi - \sum_{i=1}^n \theta_i \equiv 2\pi \quad \sum_{i=1}^n l_i \cos \phi_i \equiv 0 \quad \sum_{i=1}^n l_i \sin \phi_i \equiv 0$$

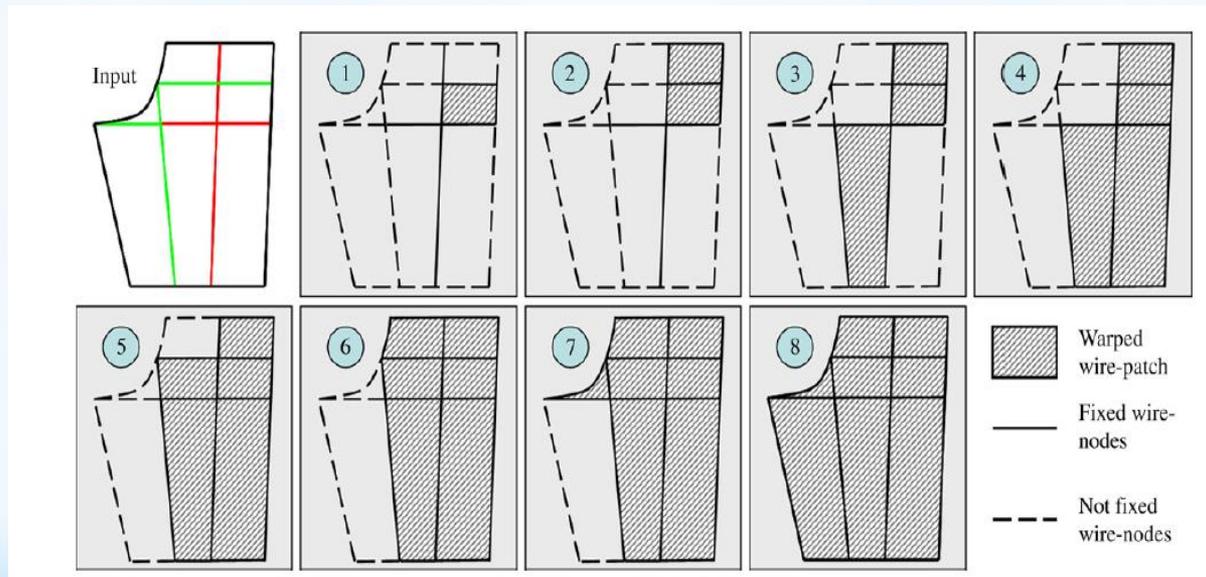


Avec  $\phi_i = \pi - \theta_i + \phi_{i-1}$

$$(x_i, y_i) = (l_i \cos \phi_i, l_i \sin \phi_i) + (x_{i-1}, y_{i-1})$$

# Etat de l'art

- \* Le dépliage progressif  $\longrightarrow$  Boucle d'optimisations plus petites
- \* évidence de forme  $\rho(P_i) = n'/n$



- \* Le dépliage global  $\longrightarrow$  Optimisation globale de tous les angles

$$\sum_{q_{kev}} \theta_k \equiv 2\pi$$

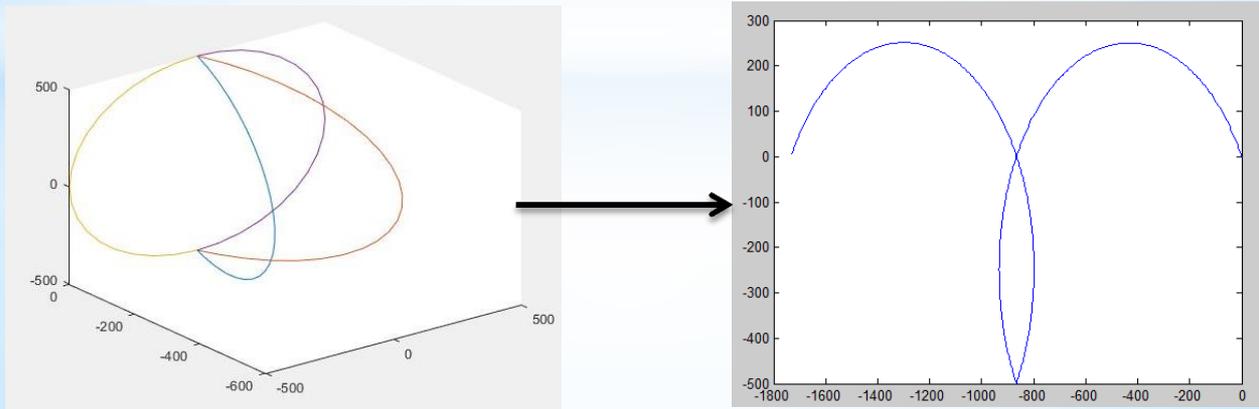
# Etat de l'art

Erreur de dépliage: variation moyenne entre les angles

$$E_{\text{ang}} = \frac{1}{N(\Omega_a)} \sum_{a \in \Omega_a} \frac{|\alpha_a - \theta_a|}{\alpha_a}$$

Evaluation de la méthode WireWarping:

- \* Il ne propose pas une démarche pour fixer les contours des laizes ni les courbes de forme
- \* La conservation de l'angle et longueur 3D est impossible dans la plus part des cas



# Démarche de dépliage

On part d'une structure 3D pour obtenir la forme des laizes à découper pour construire la structure 3D initiale avec son assemblage.

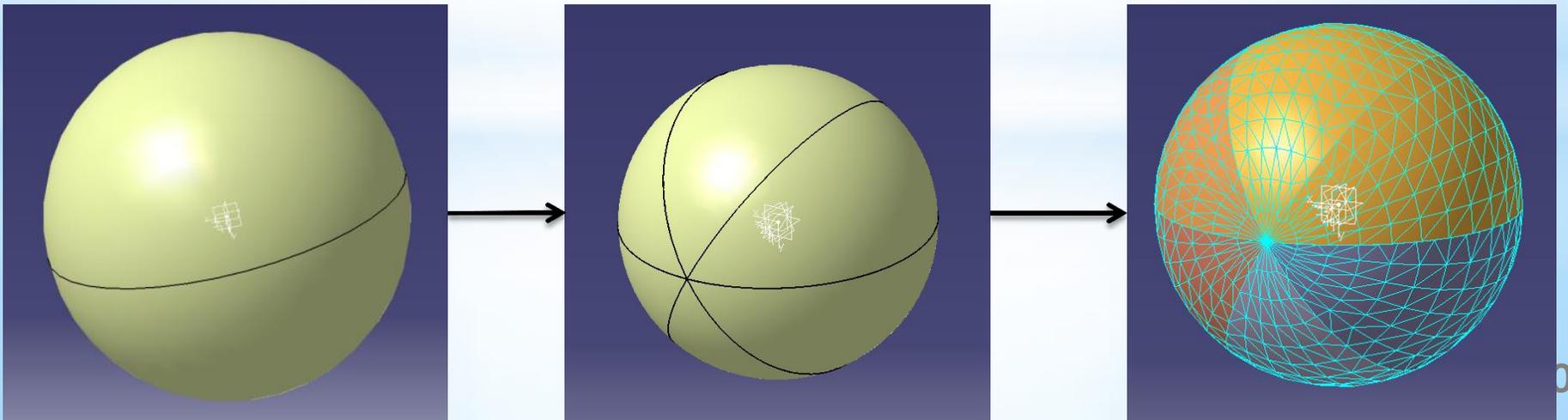
1. Division de la structure 3D, maillage et export.
2. Importation des laizes.
3. Calcul des nœuds et courbes.
4. Calcul données 3D nécessaires pour le dépliage.
5. Dépliage (deux méthodes).
6. Représentation des laizes dépliées.

\* Nous allons tester la démarche avec un sixième de sphère.

# Démarche de dépliage

Division de la structure 3D, maillage et export.

- \* Sous CATIAv5
- \* On découpe les laizes et patches dans le module « Generativeshape design » dans « FORME ».
- \* Ensuite on réalise le maillage ou facettisation de chacun des patches réalisés, pour cela j'utilise le module « STL Prototypage rapide » qui se trouve dans « USINAGE ».
- \* Export en format « .stl »



# Démarche de dépliage

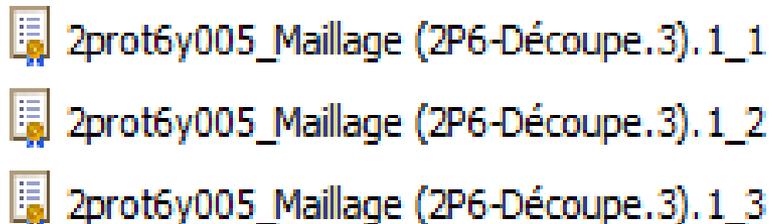
## Importation des laizes

Pour l'importation des fichiers .stl:

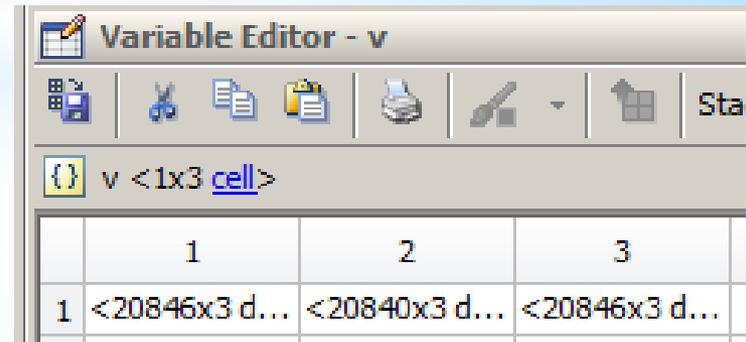
- \* `[v,f,n]=stlread(filename)`
- \* `[v*,f*]= patchslim(v,f)`

Pour l'importation de tous les patches d'une laize (boucle):

- \* `[v,f,n]=importstl(filename)` où filename fini par 1\_1
- \* Toutes les variables sous forme de vecteur de matrices.
- \* Données du Patch i dans variable{i}



2prot6y005\_Maillage (2P6-Découpe.3).1\_1  
2prot6y005\_Maillage (2P6-Découpe.3).1\_2  
2prot6y005\_Maillage (2P6-Découpe.3).1\_3

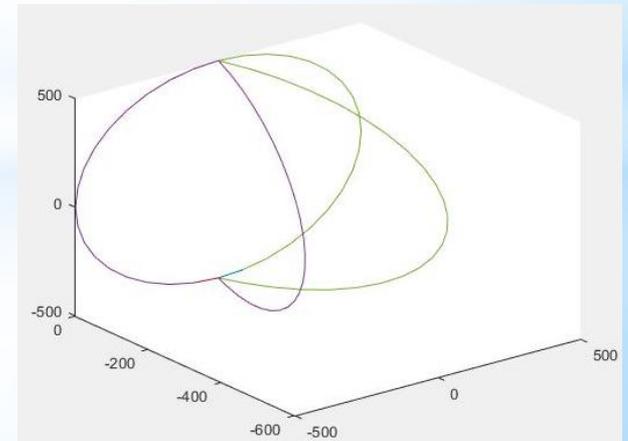
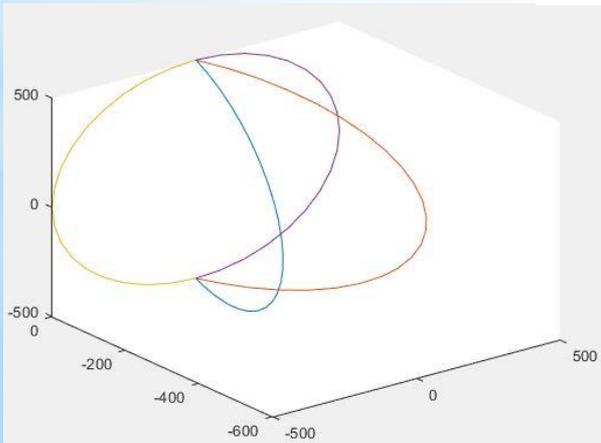


# Démarche de dépliage

Calcul des nœuds et courbes:

- \* Identifier les nœuds des courbes, « intersect ».
- \* Placer les nœuds dans l'ordre (« sort »+ distance 3d.m).
- \* Formation des courbes (par coïncidence de nœuds).
- \* Création de la matrice vertex.

```
d12=distance3d(I12,i123);  
[~,is12]=sort(d12);  
c12=I12(is12,:);
```



# Démarche de dépliage

Calcul données 3D nécessaires pour le dépliage:

\*  $l = \text{longueur3d}(\text{courbe})$

$$l_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2}$$

\*  $\alpha = \text{angle3d}(\text{courbe}, l)$

$$\alpha_i = \cos^{-1} \frac{((x_{i+1} - x_i), (y_{i+1} - y_i), (z_{i+1} - z_i)) \times ((x_i - x_{i-1}), (y_i - y_{i-1}), (z_i - z_{i-1}))}{l_i + l_{i-1}}$$

# Démarche de dépliage

Dépliage (deux méthodes) « fmincon() » dans la fonction [θ]=optimisationdetous(α,l,A,B)

\* [coût]=fonctioncout.m  $\longrightarrow$   $\min \sum_{i=1}^n 1/2(\theta_i - \alpha_i)^2$

\* [0]=nonlinconst(θ,l)  $\longrightarrow$   $\sum_{i=1}^n l_i \cos \phi_i \equiv 0$  et  $\sum_{i=1}^n l_i \sin \phi_i \equiv 0$

\* Contraintes linéaires [A][θ] = [B]  $\longrightarrow$   $n\pi - \sum_{i=1}^n \theta_i \equiv 2\pi$

\* Pour notre sixième de sphère avec un seul patch la démarche est indépendante de la méthode

# Démarche de dépliage

Dépliage (deux méthodes), selon la méthode il faut rajouter une contrainte linéaire différente  $[A^*][B^*]$ .

\* Dépliage globale:

\* Fonction de contour fermé pour chaque Patch  $\rightarrow n\pi - \sum_{i=1}^n \theta_i \equiv 2\pi$

\* Fonction des vertex  $\rightarrow \sum_{q_{k\epsilon v}} \theta_k \equiv 2\pi$

\* Dépliage progressif:

\* Fonction calcul évidence de forme  $\rightarrow \rho(P_i) = n'/n$

\* Fonction de compléter les vertex  $\rightarrow \theta_i = 2\pi - \sum_{q_{k\epsilon v}} \theta_k$

\* Fonction d'angles déjà calculés (contrainte  $\theta_i=B$ )

# Démarche de dépliage

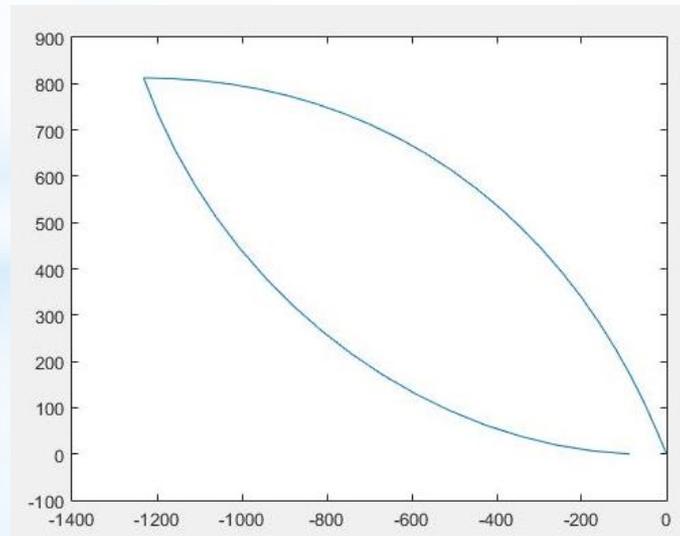
Représentation des laizes dépliées:

- \* Calcul de l'erreur de dépliage

$E = \text{erreur}(\theta, \alpha)$

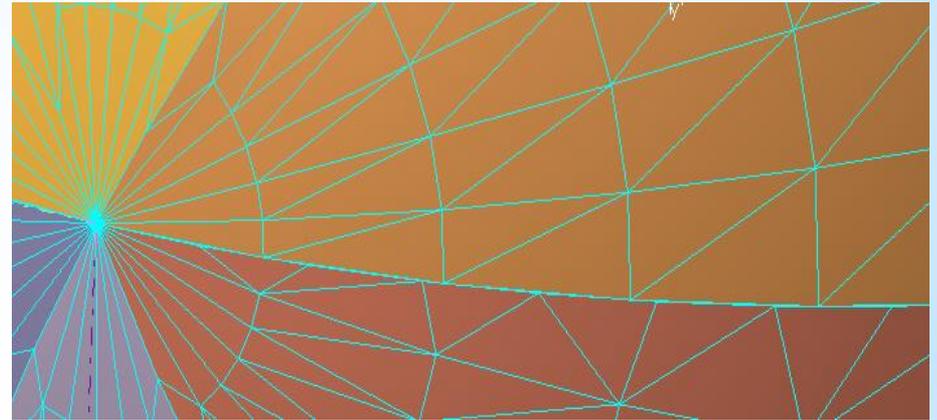
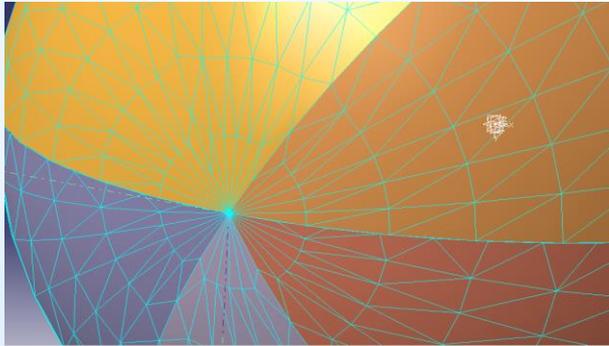
$$E_{\text{ang}} = \frac{1}{N(\Omega_a)} \sum_{\alpha \in \Omega_a} \frac{|\alpha_a - \theta_a|}{\alpha_a}$$

- \* `[xy]=plot2d(theta,l)` nous retourne les coordonnées des nœuds en 2D
- \* Fonction qui assemble les patches de chaque laize



# Analyse de la démarche

- \* L'outil de division en patchs et maillage utilisé en Catia n'est pas l'idéal : erreur de correspondance des nœuds avec forme très simple.

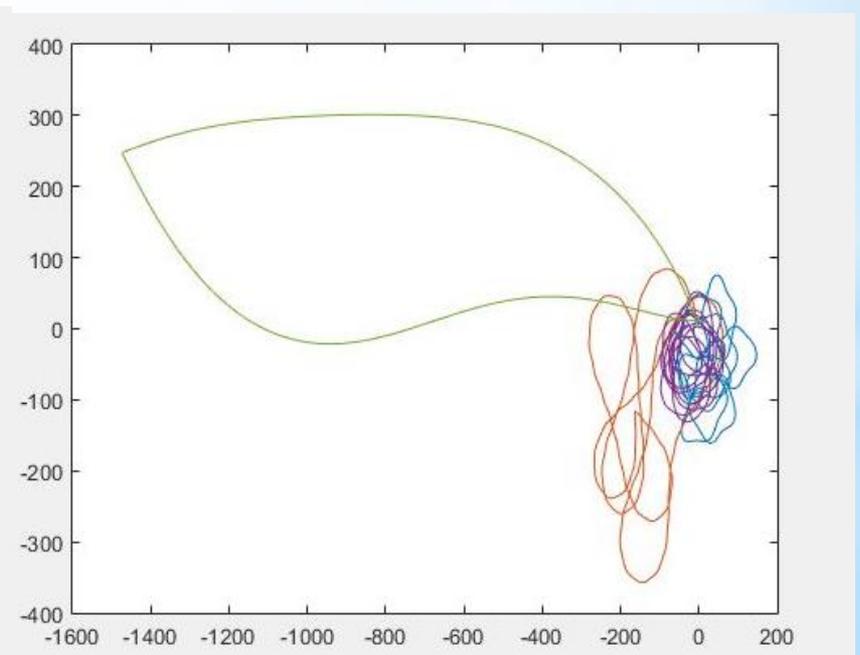
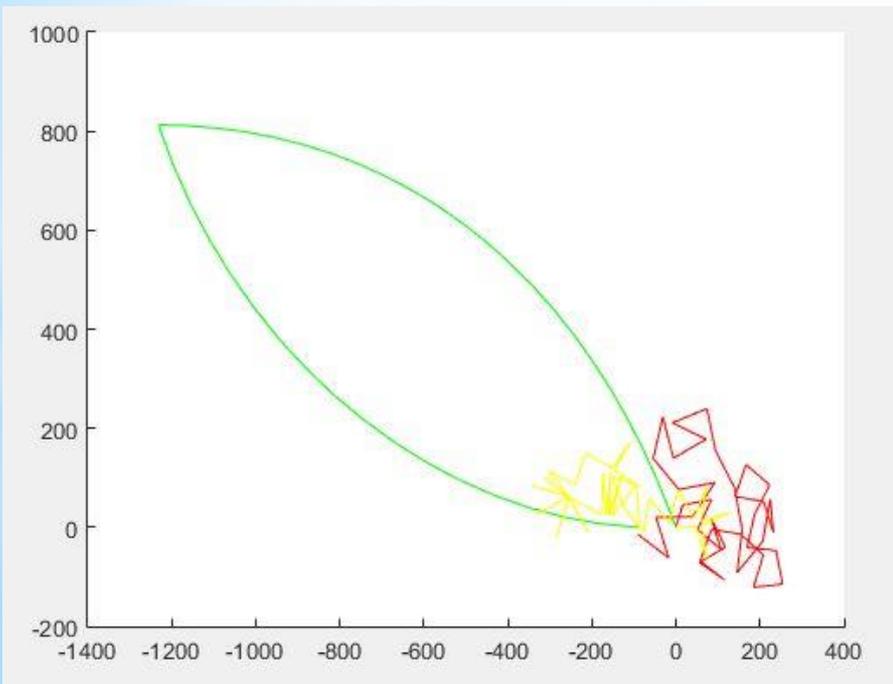


- \* Le dépliage commence à l'étape du calcul de données 3D, les 3 étapes précédentes pourrai être fait avec un bon export de données du logiciel CAO.
- \* Le temps de calcul de cette démarche est plus court de l'attendu, en plus l'optimisation converge assez vite.

# Résultats de la démarche

Influence des conditions initiales

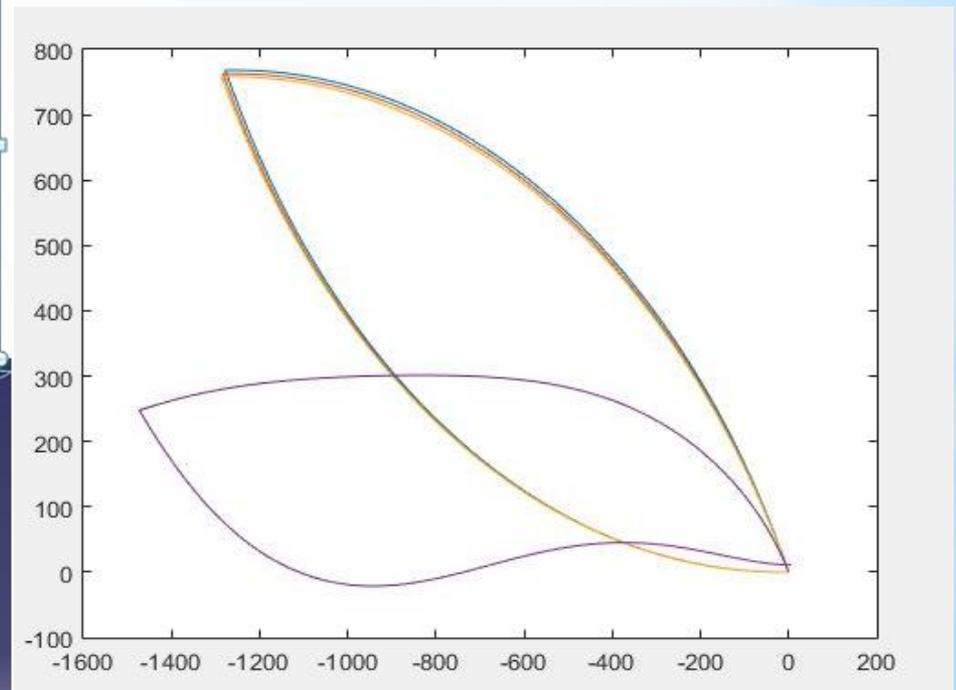
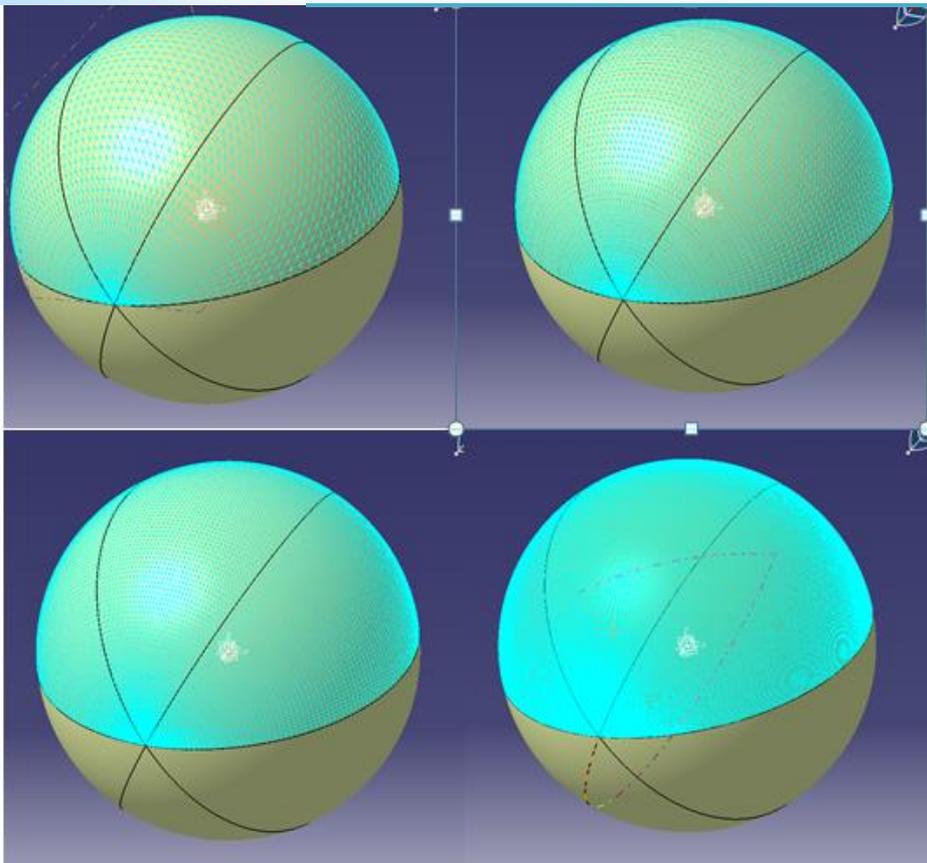
Points initiaux	$\alpha$	$\alpha/1.2$	$\alpha/1.4$	$\alpha/2$	0
Erreur de dépliage	0.012	0.657	0.657	--	--



# Résultats de la démarche

## Influence de la taille du maillage

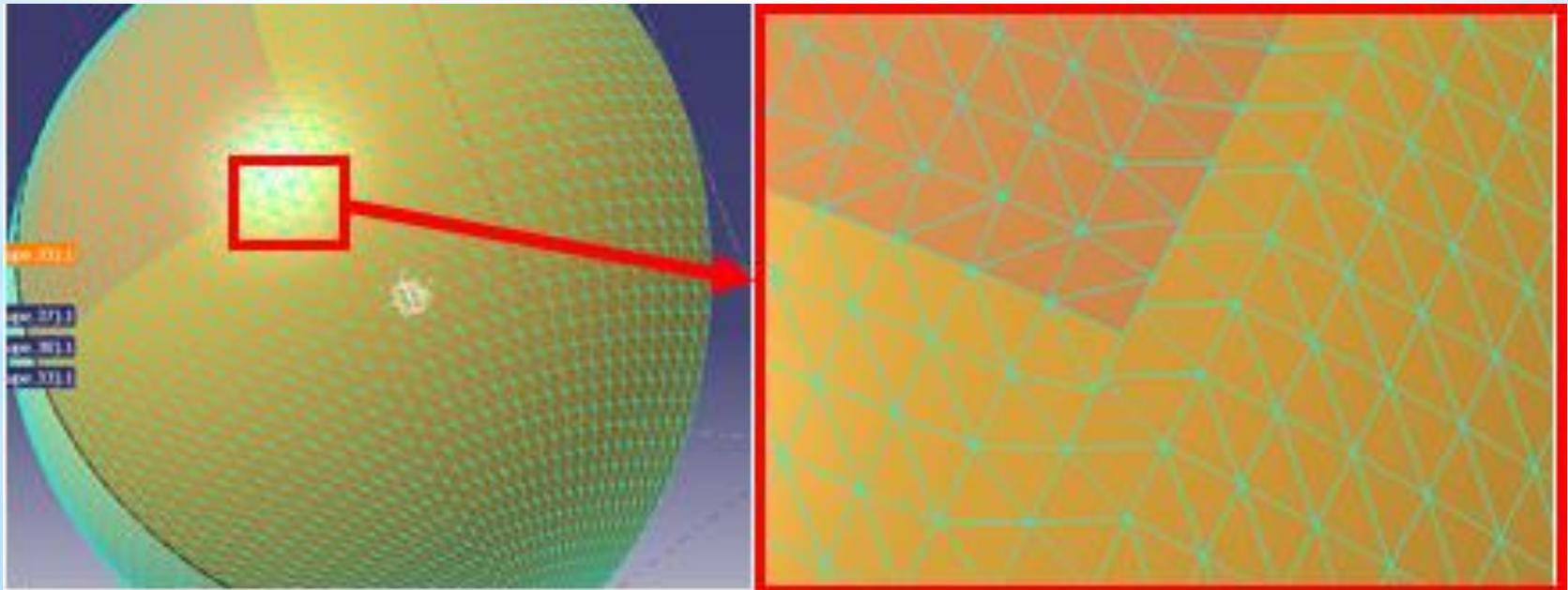
Taille de la flèche du maillage	0.2	0.1	0.05	0.01
Temps de l'optimisation (s)	0.63	0.97	1.62	11.02
Erreur de dépliage	0.012	0.0085	0.0059	0.0027



# Résultats de la démarche

Laize divisée en 2 patchs: « [directeurglobalw.m](http://directeurglobalw.m) »

- \* Script sur mesure pour tester l'amélioration du dépliage avec 2 patchs
- \* Erreur de maillage: non correspondance des nœuds, vertex, intersection courbes, etc...

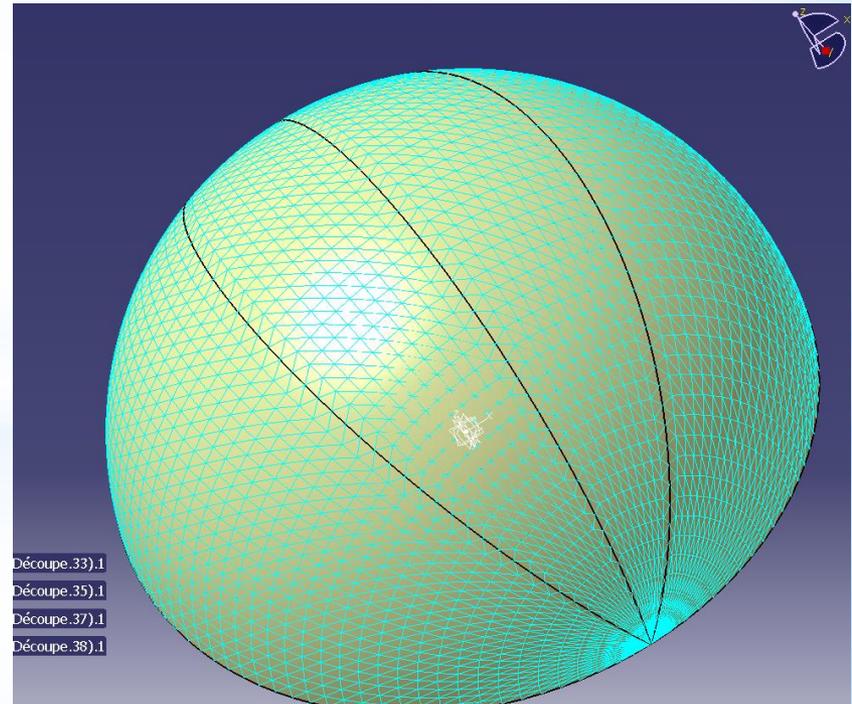


# Résultats de la démarche

Laize divisée en 2 patchs: « [directeurglobalw.m](http://directeurglobalw.m) »

\* J'ai essayé de modifier le maillage sans résultats.

\* Trouvé cette autre division, mais le script était trop spécifique et plus de temps.



# Conclusion

- \* Démarche : structure 3D → laizes à découper
- \* L'analyse des résultats nous fait penser que la méthode est bonne et surtout simple à programmer et bas temps de calcul.
- \* Difficulté de généraliser la méthode, surtout la partie de calcul des nœuds, courbes et vertex, sans augmenter énormément le temps de calcul et mémoire.
- \* Perspectives:
  - \* Tester le scrip `depliageglobalw.m` avec un bon maillage et voir l'influence de la divisions en plusieurs patches.
  - \* Considérer l'utilisation d'un module plus approprié pour le maillage et la discrétisation et exportation des courbes.
  - \* Tester l'assemblage des laizes avec une reproduction à l'échelle. Et développer une démarche plus généraliste

# BIBLIOGRAPHIE

- \* [1]: WireWarping: A fast surface flattening approach with length-preserved feature curves Charlie C.L. Wang, Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong. 28 April 2007
- \* [2]: Computing length-preserved free boundary for quasi-developable mesh segmentation. Wang CCL. IEEE Transactions on Visualization and Computer Graphics 2008;14(1):25-36.
- \* [3]: <http://www.mathworks.com/matlabcentral/fileexchange>
- \* [4]: [fr.mathworks.com/help/matlab/functionlist.html](http://fr.mathworks.com/help/matlab/functionlist.html)

# QUESTIONS

