

Python Scripts for Automatic Mesh Generation Using BlockMesh

SuperMarine Mixer Geometry Builder

Gabriel St-Pierre-Lemieux

Université de Sherbrooke

April 2018

Au sujet de la présentation

Le Plan de la présentation

- 1 Pourquoi avoir fait le script SuperMarine?
- 2 Comment marche SuperMarine dans OpenFOAM?
- 3 La conception du script. Cela inclut:
 - 1 Une description de l'outil BlockMesh.
 - 2 L'algorithme derrière le script, sans le code.

Trouvez SuperMarine sur GitHub

[Cliquez sur l'Octocat!]



<https://github.com/Spationaute/SuperMarine>

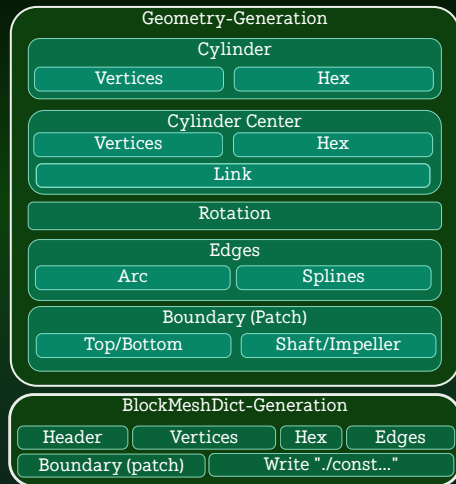
L'idée en général

Pourquoi SuperMarine?

- Produit un maillage structuré composé d'hexaèdres
- Utilise un outil déjà existant dans OpenFOAM, BlockMesh
- Utilise un script Python permettant l'exploitation des outils d'OpenFOAM au maximum

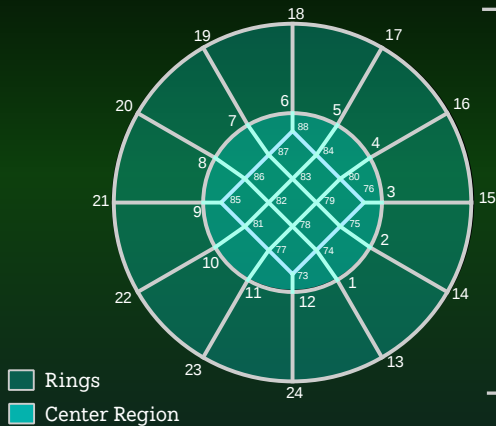
Le concept

- Créer un cylindre avec un nombre arbitraire de séparations
- Couper la forme de l'hélice dans les séparations
- Faire tourner les différents niveaux pour courber les pales



L'idée en général

Axial Cut



Side View

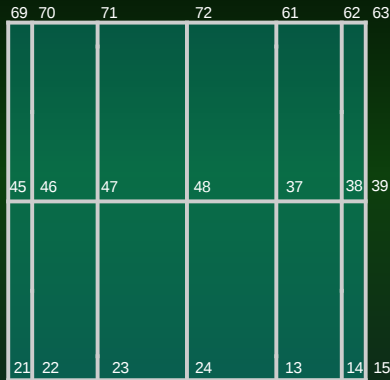


Figure: SuperMarine Schema

À propos du script

Le Bon

- SuperMarine est un script Python utilisant une bibliothèque répandue: Numpy
 - Aucune bibliothèque exotique
 - Code simple, facile à intégrer
- Le script génère une cuve de type "mixer"
- Le maillage est facile à prédire
 - Il est facile de calculer le nombre de cellules lors d'un changement dans la finesse du maillage
 - Le maillage est hexaédral et simple à contrôler

Le Mauvais

- Le code peut être encore amélioré
- Une bonne perception spatiale est nécessaire
- Peut manquer de précision pour certains cas industriels

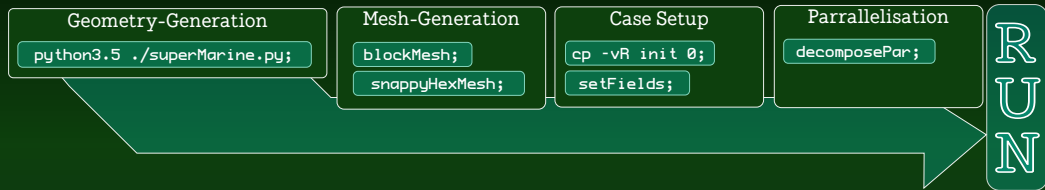


(Dessin tronqué de DenisM79 sur DeviantArt)

À propos du script

Variable	Effet
DIVI RQUAD	Nombre de divisions, angulaires, radiale et axiale par bloc Le rayon pour chaque QUADrants L'ensemble doit contenir au moins deux éléments, Le premier rayon correspond au cercle interne qui sera divisé par un carré.
NSEC	Le nombre de SECTeurs à utiliser pour construire le cercle. Doit être un multiple de 4.
HLAY	Hauteur (axiale) de chaque niveau du cylindre. L'ensemble doit contenir au moins un élément.
SHAFT	Niveaux où le centre est retiré pour mettre un arbre (Shaft). L'ensemble doit contenir le même nombre d'éléments que HLAY.
IMPELLERCUT	Quels blocs seront retirés pour créer la forme de l'hélice. Le paramètre doit être constitué d'une matrice booléenne $NSEC \times NCAR \times HLAY$. Si VRAI, le bloc correspondant est retiré.
SQRRATIO	Le RATIO du rayon que va occuper le carré dans le cercle central. Doit être entre 0 et 1.

Préparation d'un cas type



SuperMarine est un script Python qui utilise Numpy pour créer un dictionnaire BlockMesh. Le script est donc utilisé pour créer la géométrie, d'une manière automatique, avant l'étape du maillage.

Exemple: Hélice Marine

Configuration:

```
01  DIVI = [10, 10, 20]
02  RQUAD = [ 0.05, 0.1, 0.350, 0.615]
03  NSEC = 36
04  HLAY = [ 0.25, 0.60, 0.80, 1.05, 2.75]
05  SHAFT = [ 0, 1, 1, 1, 1]
06  LVLROT = [ 0, 0, 90, 90, 90, 90]
07  IMPELLERCUT =
      np.zeros((NSEC, len(RCAD), len(HLAY)))
08  IMPELLERCUT[:, : (NSEC//3), [0,1], 2] = 1
09  IMPELLERCUT[0:NSEC, 0, 2] = 1
10  SQRRATIO = 0.62112
```

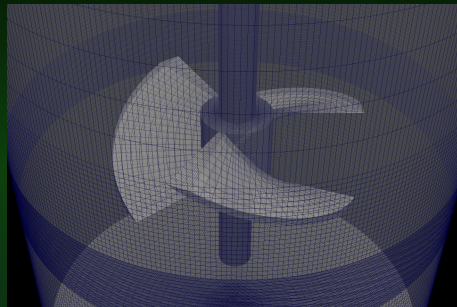


Figure: Maillage d'une hélice marine

Exemple: Marine Impeller

Configuration:

```
01  DIVI = [5, 5, 10]
02  RQUAD = [ 0.05, 0.1, 0.350, 0.615]
03  NSEC = 72
04  HLAY = [ 0.25, 0.60, 0.80, 1.05, 2.75]
05  SHAFT = [ 0, 1, 1, 1, 1]
06  LVLROT = [ 0, 0, 70, 70, 70, 70]
07  IMPELLERCUT =
      np.zeros((NSEC, len(RCAD), len(HLAY)))
08  IMPELLERCUT[:, : (NSEC//8), [0, 1], 2] = 1
09  IMPELLERCUT[0:NSEC, 0, 2] = 1
10  SQRRATIO = 0.62112
```

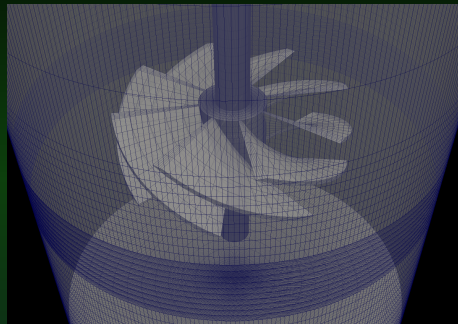


Figure: Maillage d'une hélice marine avec 8 pales

Writing a BlockMesh Dictionary

Anatomie d'un dictionnaire BlockMesh

- 1 **En-tête** : Présente les informations sur le fichier, incluant le numéro de version, le format et la version cible.
- 2 **Vertices** : Contient les points utilisés pour créer la géométrie.
- 3 **Blocks** : Contient des ensembles de points définissant les blocs.
- 4 **Edges** : Contient les informations sur les arcs/splines ou d'autres cas spéciaux.
- 5 **Boundary** : Contient des ensembles de points définissant chaque face d'un objet.

Écrire un dictionnaire BlockMesh: Vertices

BlockMeshDict

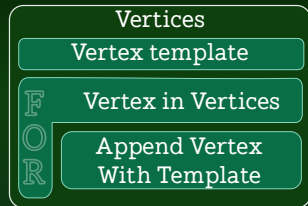
```
01 convertToMeter = 1;
02 vertices
03 (
04     (0.0500000000 0.0000000000 0.0000000000)
05     (0.0433012702 -0.0250000000 0.0000000000)
06     (0.0250000000 -0.0433012702 0.0000000000)
07     ...
08 );
```

superMarine.py

```
01 vtemp="({:20.10f} {:20.10f} {:20.10f})"
02 BlockMesh+=["", "vertices", "("]
03 for v in vertex:
04     BlockMesh.append(vtemp.format(*v))
05 BlockMesh+=[");"]
```

Écrire BlockMeshDict

[Lines 352-443]



Chaque section subséquente est construite de la même façon.

Dessiner le cylindre

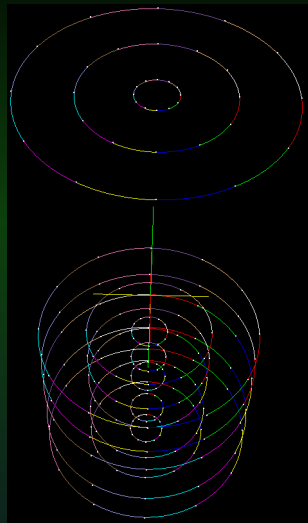
Définition des points du cylindre [Lines 92-109]

- 1 Les points sont créés en appliquant une matrice de rotation sur un vecteur unitaire.

$$\text{rotz}(A, \theta) = A \cdot \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

[Def. at lines 58-62]

- 2 Les points sont multipliés avec chacun des rayons définis par l'utilisateur. Puis, le produit est ajouté à l'ensemble.
- 3 Cet ensemble est utilisé pour définir chaque nouveau niveau.

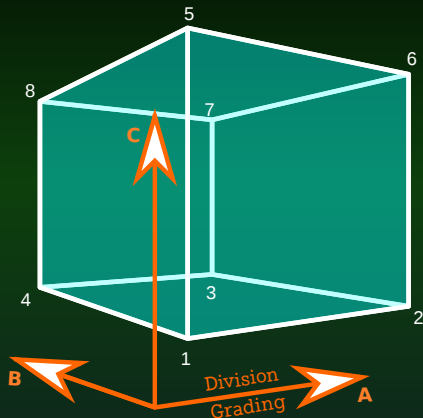


Écrire un dictionnaire BlockMesh: Hex

Chaque coin de l'hexaèdre doit être défini dans un ordre précis. En utilisant la règle de la main droite, le dessous doit-être défini en premier. Puis, dans le même ordre, le dessus doit être défini.

BlockMeshDict

```
01 blocks
02 (
03   hex ( 0 1 13 12 48 49 61 60 )
         ( 10 10 20 ) simpleGrading ( 1 1 1 )
04   hex ( 1 2 14 13 49 50 62 61 )
         ( 10 10 20 ) simpleGrading ( 1 1 1 )
05   hex ( 2 3 15 14 50 51 63 62 )
         ( 10 10 20 ) simpleGrading ( 1 1 1 )
06   ...
07 );
```



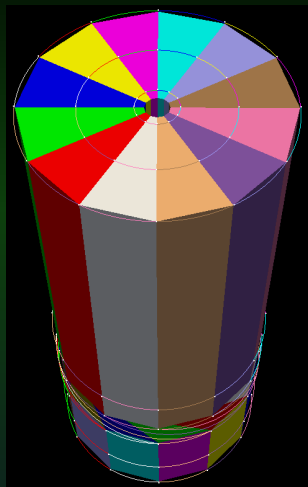
Construire les blocs

Définition des hexaèdres du cylindre [Lines 114-134]

- 1 Les hexaèdres sont définis en utilisant un gabarit contenant l'ordre des points à sélectionner.

[0, 1, 1+NSEC, NSEC, NSEC*nCad, 1+NSEC*nCad,
1+NSEC+nCad*NSEC, NSEC+nCad*NSEC]

- 2 Le gabarit est appliqué pour chaque élément du cylindre, à l'exception des blocs que l'utilisateur a définis comme hélice.
- 3 Une exception doit être faite pour le dernier bloc du quadrant, puisqu'il contient les derniers points et les premiers points du cercle.



Remplir les trous

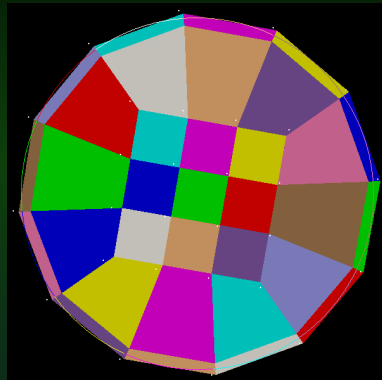
Définition des points centraux [Lines 137-151]

- 1 Pour chaque niveau:
 - 1 Quatre points équidistants sont sélectionnés sur le cercle central.
 - 2 Entre ces points, une interpolation est faite pour générer une grille.
$$ratio * (p_0 + X * \frac{p_1 - p_0}{NSEC // 4}) \quad \forall X \in [0..NSEC // 4]$$

Définition des hexaèdres centraux

[Lines 159-190]

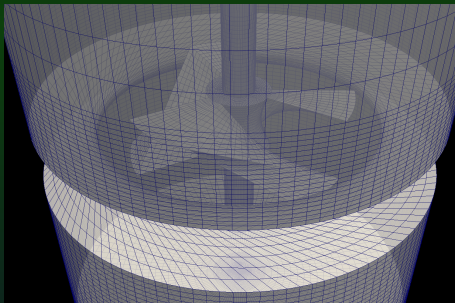
- 1 Pour chaque niveau, l'hexaèdre est créé en utilisant un gabarit sur la grille.
- 2 Pour chaque niveau, un autre gabarit est utilisé pour rejoindre le carré central avec le cercle central.



Expliquez ces splines

Pourquoi ajouter des arcs et des splines?

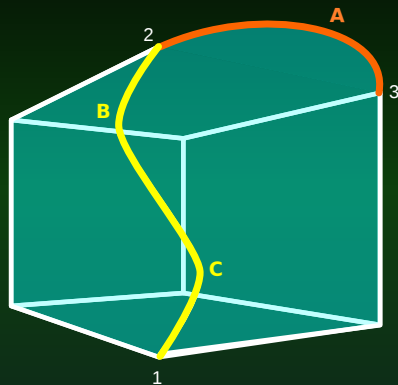
- 1 Quand des niveaux du cylindre sont tournés, l'utilisation de lignes droites résulterait en une géométrie tordue dans laquelle le niveau tourné formerait une selle.
- 2 Les lignes qui joignent les deux niveaux doivent être corrigées pour suivre la forme du cylindre.



Écrire un dictionnaire BlockMesh: Edges

BlockMeshDict

```
01 edges
02 (
03   arc 271 272
04     ( 0.24748737341529 0.24748737341529176 2.75 )
05   arc 272 273
06     ( 0.338074039201173 0.09058666578588254 2.75 )
07   arc 273 274
08     ( 0.33807403920117 -0.09058666578588195 2.75 )
09   spline 108 156 (
10     ( 0.0996917334 -0.0078459096 0.6100000000 )
11     ( 0.0987688341 -0.0156434465 0.6200000000 )
12     ...
13     ( 0.0156434465 -0.0987688341 0.7800000000 )
14     ( 0.0078459096 -0.0996917334 0.7900000000 )
15   )
16   ...
17 );
```



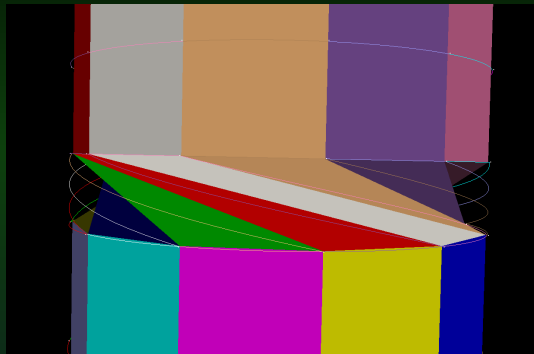
Suivre les courbes

Définition des arcs [Lines 202-217]

- 1 Un arc est créé pour chaque paire de points horizontaux composant le cylindre.
- 2 Il faut ajouter une exception pour les derniers points. Ils doivent joindre les premiers et non les points suivants.

Définition des Splines [Lines 222-240]

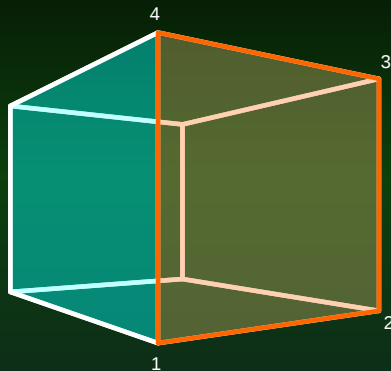
- 1 Une spline est générée si une paire de points formant une ligne verticale ne sont pas alignés axialement.
- 2 La spline est produite à l'aide d'une interpolation entre les deux points, pour la position axiale, et les coordonnées du cercle pour les autres axes.



Écrire un dictionnaire BlockMesh: Les Patches

BlockMeshDict

```
01 boundary
02 (
03 walls
04 {
05     type wall;
06     faces
07     (
08         (36 37 85 84)
09         (37 38 86 85)
10         (38 39 87 86)
11         ...
12     );
13 }
14 ...;
15 );
```



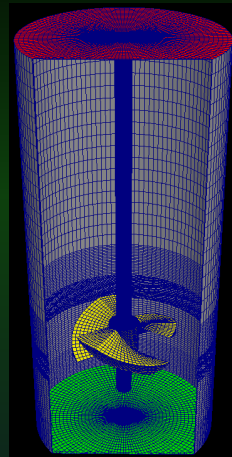
Joindre les Patches

Définition des patches [Lines 244-348]

- 1 Un ensemble de gabarits est appliqué sur le secteur limitrophe pour créer le patch du mur (wall).
- 2 Pour l'hélice (impeller) et l'arbre (shaft), le script filtre chaque cellule et applique le gabarit correspondant.
- 3 La même chose est faite pour le bas (bottom) et le haut (top) du cylindre.

Note: Joindre deux faces

Il est possible de joindre deux faces qui ne partagent pas les mêmes coins. Mais cela est compliqué, car il faut que les maillages produits correspondent.



Conclusion

- 1 Python permet la création d'outils puissants pour créer des dictionnaires BlockMesh
- 2 SuperMarine donne un exemple en permettant la production de cuves cylindriques de type "mixer".
- 3 Le code est indépendant, libre, et peut-être utilisé avec la suite d'OpenFOAM.

Brise mousse en forme de cône

- 1 Génération de plusieurs formes de brise-mousse coniques.
- 2 Utilisé avec une zone de raffinement.

(Not Yet Published)

Cellules électrolytiques

- 1 Le script est utilisé pour créer une cellule électrolytique.

Askari E., Proulx P., Passalacqua A., *Modeling of Bubbly Flow Using CFD-PBM Solver in OpenFOAM: Study of Local Population Balance Models and Extended Quadrature Method of Moments Applications*, Chemengineering, MDPI, 8 Jan 2018.