

Visualisation scientifique avec ParaView

3^e partie

Pier-Luc St-Onge
et Alex Razoumov



**Digital Research
Alliance** of Canada



Diapositives, données et codes : <https://folio.vastcloud.org/ecolehivernale>

- le lien "Fichier ZIP ..." permet de télécharger `paraview.zip` (~30 Mo)
- une fois le contenu extrait, on obtient `codes/`, `data/` et `slides2.pdf` (en anglais)

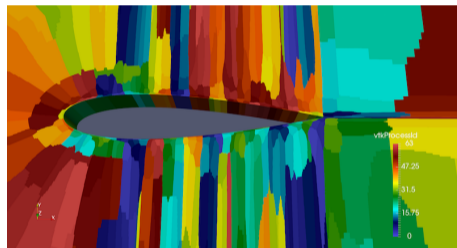


Installez ParaView 6.0.x sur votre ordinateur à partir de <http://www.paraview.org/download>

Plan de l'atelier

● Animation

- trois approches
- animation de la caméra
- exercice récapitulatif avec un script et une animation
- une seule ligne du temps avec plusieurs propriétés



ANIMATION DANS PARAVIEW

Méthodes d'animation (1/2)

1. Animation par images clés : animez n'importe quelle propriété de n'importe quel objet du pipeline
 - permet de créer facilement des animations originales, mais aux possibilités limitées
 - dans *View* → *Time Manager*
 - ★ configurez *Number of frames* = 90
 - ★ sélectionnez un objet et une propriété dans les menus déroulants
 - ★ cliquez sur “+” pour créer une nouvelle piste
 - ★ au besoin, double-cliquez sur la piste pour l'éditer
 - ★ appuyez sur “▶” pour démarrer l'animation
 - on peut y combiner plusieurs pistes / chronologies
 - exemples : avec le fichier `data/sineEnvelope.nc`
 - 1.1 animation d'une tranche (*Slice*)
 - 1.2 animation d'une surface de contour

Méthodes d'animation (2/2)

2. Utilisez la capacité de ParaView à reconnaître une séquence de fichiers similaires

- animation temporelle uniquement, mais très pratique
- essayez de charger la séquence `data/2d*.vtk` et de l'animer
 - ★ visualisez une image (représentation *Slice* ou *Surface*)
 - ★ appuyez sur "▶"

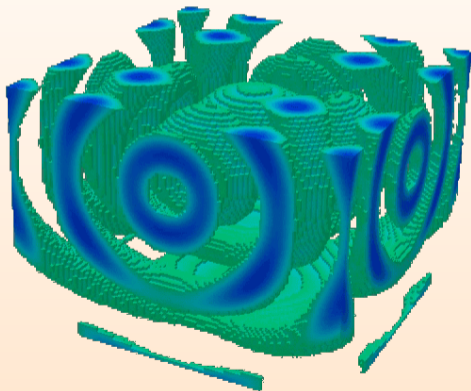
3. Scriptez votre animation en Python

- courbe d'apprentissage plutôt abrupte, mais méthode très puissante ; permet de faire tout ce que vous pouvez faire dans l'interface graphique
- scénario d'utilisation typique : générer une image par fichier d'entrée
- un exercice plus simple sans fichier d'entrée : voir la diapositive suivante

Exercice : animation de la croissance d'une fonction (1/2)

➡ À partir du fichier `data/sineEnvelope.nc` (grille 100^3), reproduisez la vidéo <https://youtu.be/Oc3DAJvArIU>

1. Échelle de couleurs
`erdc_rainbow_bright` avec
inversion de la fonction de transfert
2. Filtre *Threshold* avec
Lower Threshold = 1.2
3. Filtre *Clip* pour reproduire l'image ci-contre
 - origine $O = (49.5, 70, 49.5)$
 - normale $N = (0, 1, 0)$
4. (Suite à la page suivante...)




Exercice : animation de la croissance d'une fonction (2/2)

Deux solutions possibles :

1. Dans le *Time Manager*, animez `Clip1` en faisant varier la composante Y de l'origine du plan de coupe (soit `Clip Type - Origin (1)`) de 0 à 99
 - ➡ pour un résultat optimal, enregistrez l'animation sous forme de séquence de fichiers PNG : `File` → `Save Animation`, nom `frame` (cliquez `OK`) et suffixe `%04d`
 2. Via un script Python, il faut :
 - utiliser `Tools` → `Start/Stop Trace` pour enregistrer la configuration initiale du `Clip`
 - sauvegarder le **script Python** correspondant
 - éditer **une partie du code** sous forme de boucle de sorte à faire varier la position de l'origine du plan de coupe et à sauvegarder un fichier PNG à chaque itération
 - exécuter le script dans ParaView pour produire les 100 images
- ➡ Dans les deux cas, fusionnez les PNG à l'aide d'un outil tiers, par exemple :
- ```
ffmpeg -r 30 -i frame%04d.png -c:v libx264 -pix_fmt yuv420p \
-vf "scale=trunc(iw/2)*2:trunc(ih/2)*2" movie.mp4
```

# Animation de la caméra dans l'interface graphique

Bonne ressource d'introduction : <https://docs.paraview.org/en/latest/UsersGuide/animation.html>

1. Créez une visualisation statique quelconque et ajustez la vue à votre convenance
2. Cliquez sur  ou *Adjust Camera* (icône située à gauche, en haut de la fenêtre de rendu)
  - assurez-vous que les coordonnées de *Center of Rotation* et de *Focal Point* sont égales, car c'est ce qui permet de centrer la visualisation lors des rotations ; notez ces coordonnées pour la suite
3. Dans le *Time Manager*, enlevez les pistes précédentes et essayez ce qui suit :

## 3a. Orbite circulaire

- ajustez le nombre d'images à 120
- choisir *Camera* et *Follow Path*, et cliquez sur "+"
- double-cliquez sur la nouvelle piste
- sélectionnez la 1<sup>re</sup> image clé, cliquez sur *Create Orbit*, ajustez le *Center* au besoin et cliquez *OK*
- double-cliquez sur la cellule *Path ...*, ajustez le *Camera Focus* au besoin et cliquez *OK*
- cliquez sur *OK* pour fermer l'éditeur d'images clé de la caméra

## 3b. Chemin personnalisé

- choisir *Camera* et *Follow Path*, et cliquez sur "+"
- double-cliquez sur la nouvelle piste
- double-cliquez sur la cellule *Path ...*
- sélectionnez *Camera Position*
  - une boucle avec des sphères apparaissent
  - déplacez quelques sphères dans l'espace
- on peut aussi changer le point focal (*Camera Focus*) et la direction du haut de la caméra (*Up Direction*)
- cliquez sur *OK* dans la fenêtre *Animation Keyframes*

4. Cliquez sur "▶"

# Exercice : animez la caméra en direction du point focal

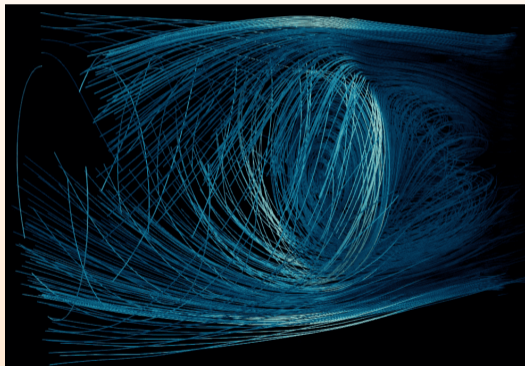
Utilisez l'animation par image clé pour faire voler la caméra vers le point focal jusqu'à  $2/3$  de la distance initiale

- Dans le *Time Manager*, enlevez les pistes précédentes, ajoutez une nouvelle piste pour la Camera selon `Interpolate cameras` et double-cliquez sur celle-ci
- À l'image clé au temps  $t = 0$  :
  - notez la position actuelle de la caméra  $\mathbf{p}_{\text{initial}}$
  - notez la position actuelle du point focal  $\mathbf{p}_{\text{focal}}$
- À l'image clé au temps  $t = 1$  :
  - Choix 1) calculez la nouvelle position de la caméra selon :
$$\frac{\mathbf{p}_{\text{initial}}}{3} + \frac{2\mathbf{p}_{\text{focal}}}{3}$$
  - Choix 2) faites un zoom approximatif avec la souris ; ensuite, dans la fenêtre d'édition de la position, cliquez sur *Use Current*
  - Vérifiez si le point focal a changé ou non ; corrigez au besoin

# Animation d'un écoulement stationnaire (1/4)

## Lignes de courant à travers une tranche

- Ce jeu de données ne comporte pas de dimension temporelle, or nous souhaitons quand même produire une animation avec des lignes de courant
- <https://youtu.be/67qTsw8-kKk> ou [hidden/radialSlice.mp4](#) sur l'ordinateur de l'instructeur
- <https://youtu.be/2uSRrZ6oW0o> ou [hidden/xySlice.mp4](#) sur l'ordinateur de l'instructeur
- À la page suivante, on a une piste possible, mais quelles autres idées créatives pouvons-nous explorer ?



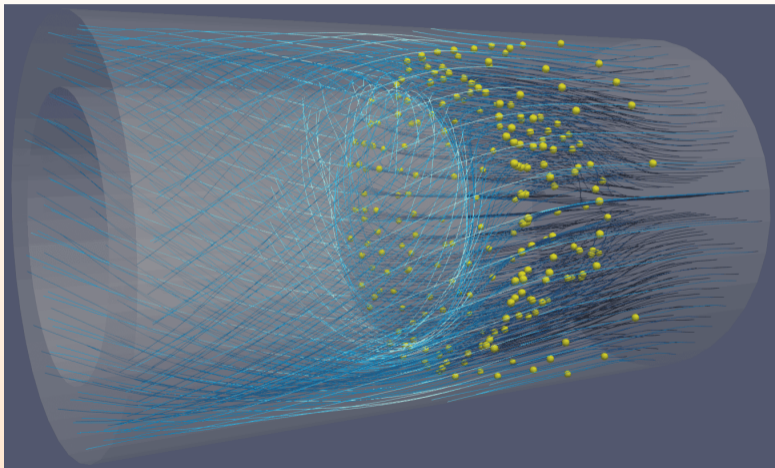
# Animation d'un écoulement stationnaire (2/4)

## Lignes de courant à travers une tranche

1. Chargez la vitesse  $V$  du fichier `disk_out_ref.ex2` ; coloriez selon  $V$
2. Obtenez une tranche (filtre *Slice*) au centre du volume le long de l'axe  $Z$ 
  - *Origin*  $O = (0, 0, 0)$
  - *Normal*  $N = (1, 0, 0)$
3. Cherchez et ajoutez un filtre *Stream Tracer With Custom Source* avec
  - *Input* = `disk_out_ref.ex2`
  - *Seed Source* = `Slice1`
4. Ajoutez ensuite un filtre *Tube* avec *Radius* = `0.015`
5. Dans le *Time Manager*, avec le `Slice1`, animez *Slice Type* - *Origin* (0) de `-1` à `1` (au lieu de l'intervalle complet `[-5.75, 5.75]`) avec 100 images
6. Utilisez un arrière-plan noir et coloriez la vitesse  $V$  selon l'échelle `blue2cyan`
7. Décochez *Show Plane* de la tranche `Slice1`
8. Sauvegardez l'animations sous forme de fichiers PNG (pour encoder à 10 ips)

# Animation d'un écoulement stationnaire (3/4)

Animer des glyphs naviguant le long des lignes de courant



<https://youtu.be/cTBXeybfFyk> ou [hidden/timeContours.mp4](#) en démonstration

# Animation d'un écoulement stationnaire (4/4)

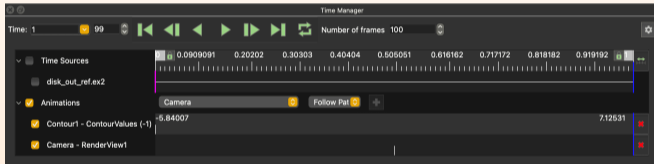
Animer des glyphes naviguant le long des lignes de courant

1. Dans le *Time Manager*, supprimez la piste de l'exemple précédent
2. Ajustez le *Slice1* de sorte à avoir :
  - *Origin*  $O = (0, 0, 8)$
  - *Normal*  $N = (0, 0, -1)$
3. À partir du *StreamTracer...*, ajoutez un filtre *Contour*
  - sélectionnez *Contour By = IntegrationTime*
  - configurez la valeur de contour à  $0.1$ , soit un petit pas de temps après *Slice1*
4. Ajoutez un filtre *Glyph* à la sortie du filtre *Contour* et configurez des sphères avec *Scale Array = No scale array* et un *Scale Factor = 0.3* ; coloriez en jaune
5. Dans le *Time Manager*, animez le *Contour1* via *Isosurfaces* de  $-0.9$  à  $1.5$
6. La vidéo publiée a été produite avec 2000 images à 60 ips
  - cette haute résolution temporelle était propice à une vidéo finale
  - pour déboguer l'animation, **100 images** sont amplement suffisantes

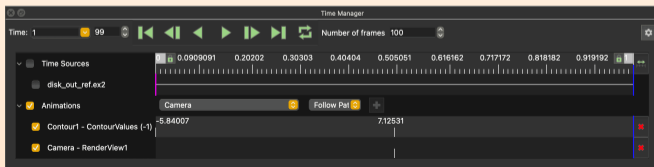
# Exercice : plusieurs pistes dans une seule animation

À partir de l'animation de l'isosurface par *IntegrationTime* de  $t = 0$  à  $t = 1$  :

1. Amenez le curseur à  $t = 0$  et ajustez la vue
2. Ajoutez une seconde piste pour animer la caméra selon *Follow Path* de  $t = 0.5$  à  $t = 1$ , soit pendant que la première piste continue de jouer sa seconde moitié



3. Maintenant, faites en sorte de compléter l'animation de l'isosurface avant la rotation



# Combiner plusieurs pistes dans une même animation

- En principe, vous pouvez ajouter à l'animation autant de pistes que vous le souhaitez, chacune avec son propre intervalle de temps et ses propres variables
- Voici un exemple de soumission à la compétition *Visualize This 2017* de WestGrid par Nadya Moisseeva (UBC)

Mode: Sequence Time: 46.3951935914553 Start Time: 0 End Time: 50 No. Frames: 750

|                                     | Time                        | 0.000e+0           | 5.000e+0 | 1.000e+1 | 1.500e+1           | 2.000e+1           | 2.500e+1         | 3.000e+1           | 3.500e+1         | 4.000e+1         | 4.500e+1 | 5.000e+1 |  |
|-------------------------------------|-----------------------------|--------------------|----------|----------|--------------------|--------------------|------------------|--------------------|------------------|------------------|----------|----------|--|
| <input checked="" type="checkbox"/> | TimeKeeper1 - Time          |                    |          |          |                    |                    |                  |                    |                  |                  |          |          |  |
| <input checked="" type="checkbox"/> | Contour7 - Isosurfaces (-1) |                    |          |          |                    | -0.500920190903914 |                  |                    |                  | 5.10504690092901 |          |          |  |
| <input checked="" type="checkbox"/> | Contour2 - Isosurfaces (-1) | -0.500780935631845 |          |          |                    |                    | 4.88465395492028 |                    |                  |                  |          |          |  |
| <input checked="" type="checkbox"/> | Contour4 - Isosurfaces (-1) |                    |          |          | -0.500780935631845 |                    |                  |                    | 4.88465395492028 |                  |          |          |  |
| <input checked="" type="checkbox"/> | Contour3 - Isosurfaces (-1) |                    |          |          | -0.500780935631845 |                    |                  |                    | 4.88465395492028 |                  |          |          |  |
| <input checked="" type="checkbox"/> | Contour6 - Isosurfaces (-1) |                    |          |          |                    | -0.500920190903914 |                  |                    |                  | 5.10504690092901 |          |          |  |
| <input checked="" type="checkbox"/> | Contour8 - Isosurfaces (-1) |                    |          |          |                    | -0.50092           |                  |                    |                  | 5.10505          |          |          |  |
| <input checked="" type="checkbox"/> | Contour5 - Isosurfaces (-1) |                    |          |          | -0.500780935631845 |                    |                  |                    | 4.88465395492028 |                  |          |          |  |
| <input checked="" type="checkbox"/> | Camera                      |                    |          |          |                    |                    |                  |                    |                  |                  |          |          |  |
| <input checked="" type="checkbox"/> | Contour9 - Isosurfaces (-1) |                    |          |          |                    |                    |                  | -0.500920190903914 |                  | 5.10504690092901 |          |          |  |
| <input checked="" type="checkbox"/> | WindTracer - Opacity        | 0                  | 0.2      | 0.2      | 0                  |                    |                  |                    |                  |                  |          |          |  |
| <input checked="" type="checkbox"/> | Contour1 - Isosurfaces (-1) | -0.500780935631845 |          |          |                    |                    | 4.88465395492028 |                    |                  |                  |          |          |  |

Contour1 Isosurfaces

<https://youtu.be/eT8tj8BoYfg> ou [hidden/complexAnimation.mp4](https://folio.vastcloud.org/ecolehivernale/hidden/complexAnimation.mp4) sur l'ordinateur de l'instructeur