Visualisation scientifique de configurations de transport de sédiment avec OpenFoam, Paraview et Blender

Antoine MATHIEU – Cyrille BONAMY

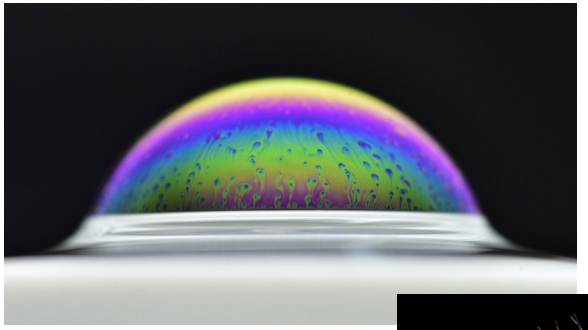
Directeur de thèse : Julien CHAUCHAT

11/2020

La visualisation scientifique

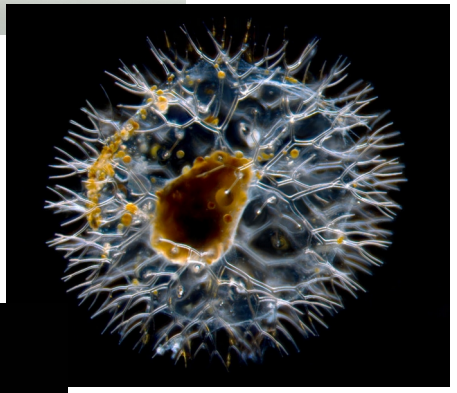
Quel est l'intérêt de la visualisation scientifique ?

- **Mettre en valeur** la recherche avec un aspect esthétique



M. Pasquet, Laboratoire de physique des solides (CNRS / Université Paris-Saclay), S. Guichard

- **Attirer l'attention** de l'auditoire lors d'une présentation



T. Biard, LOG (CNRS / Université de Lille / Université Littoral Côte d'Opale)

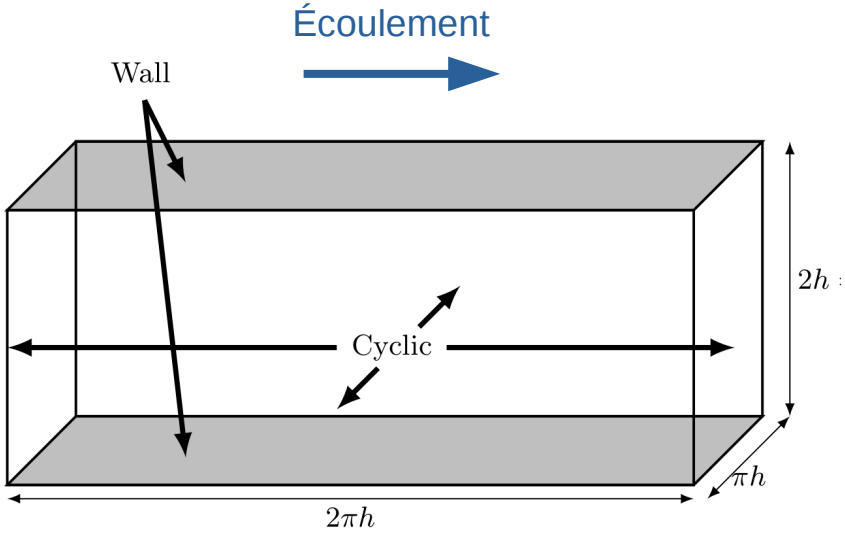
- Permettre de mieux **communiquer** sur ses travaux de recherche

A. Mathieu, LEGI (CNRS / Université Grenobles-Alpes)

Contexte

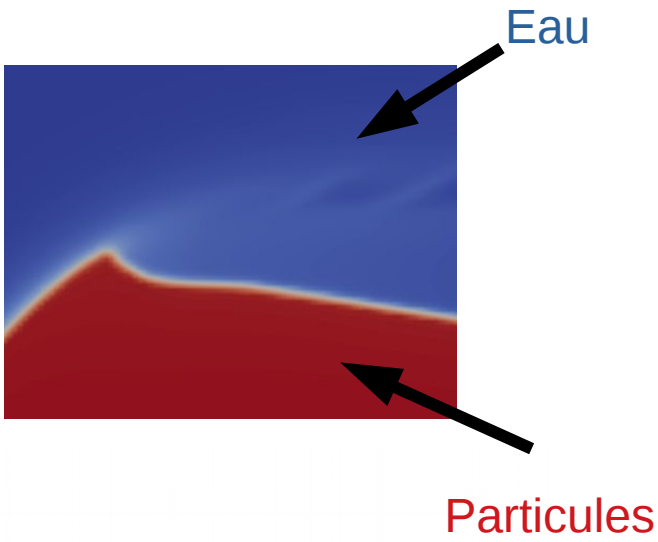
Simulations diphasiques à turbulence résolue de transport de particules solides dans un écoulement unidirectionnel d'eau:

- Mettre en évidence les interactions entre particules et la turbulence du fluide
- Mieux comprendre les mécanismes de mise en suspension des particules
- Proposer de meilleures paramétrisations pour les modèles à turbulence moyennée
- Prédire l'évolution du littoral à grande échelle



Modèle diphasique Eulérien-Eulérien

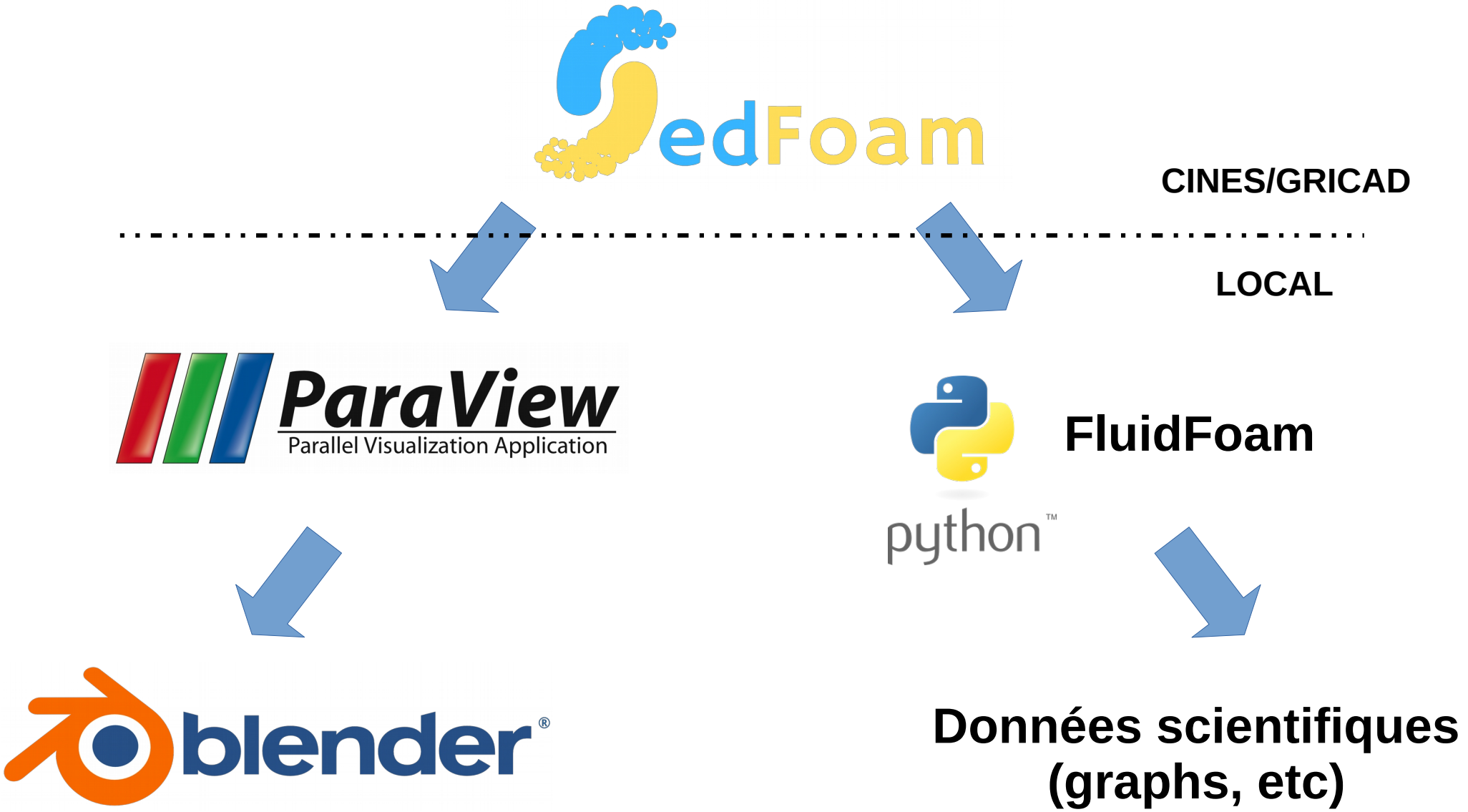
- Les particules et l'eau sont vus comme deux milieux continus



- Des équations couplées de conservation de la masse et de quantité de mouvement sont résolues pour prédire l'écoulement

- Le solveur (sedFoamLES) est implémenté dans la boite à outil de CFD opensource OpenFoam

Workflow


edFoam
CINES/GRICAD
LOCAL
ParaView
Parallel Visualization Application
FluidFoam
python
blender
Données scientifiques
(graphs, etc)

Workflow

Fichiers d'entrée
(~ 100 Ko)

Champs 3D
(~ 400 Mo)

ParaView
Parallel Visualization Application

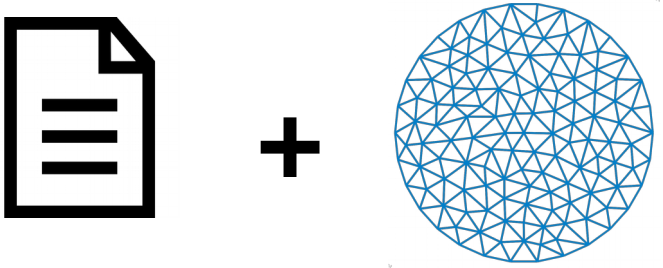
Surfaces exportées
(~ 100 Mo)

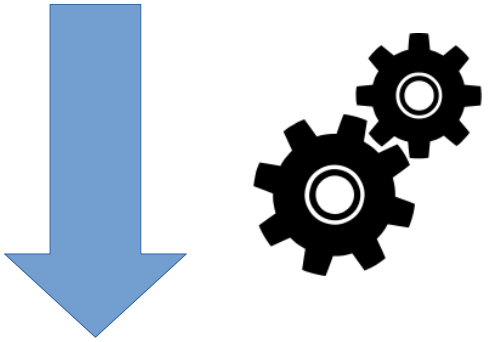
blender

Image .png
(~ 10 Mo)

Fichiers d'entrée

Maillage





Fichiers de sortie
(champs 3D)

Workflow

Fichiers d'entrée (~ 100 Ko)

↓

edFoam

Champs 3D (~ 400 Mo)

↓

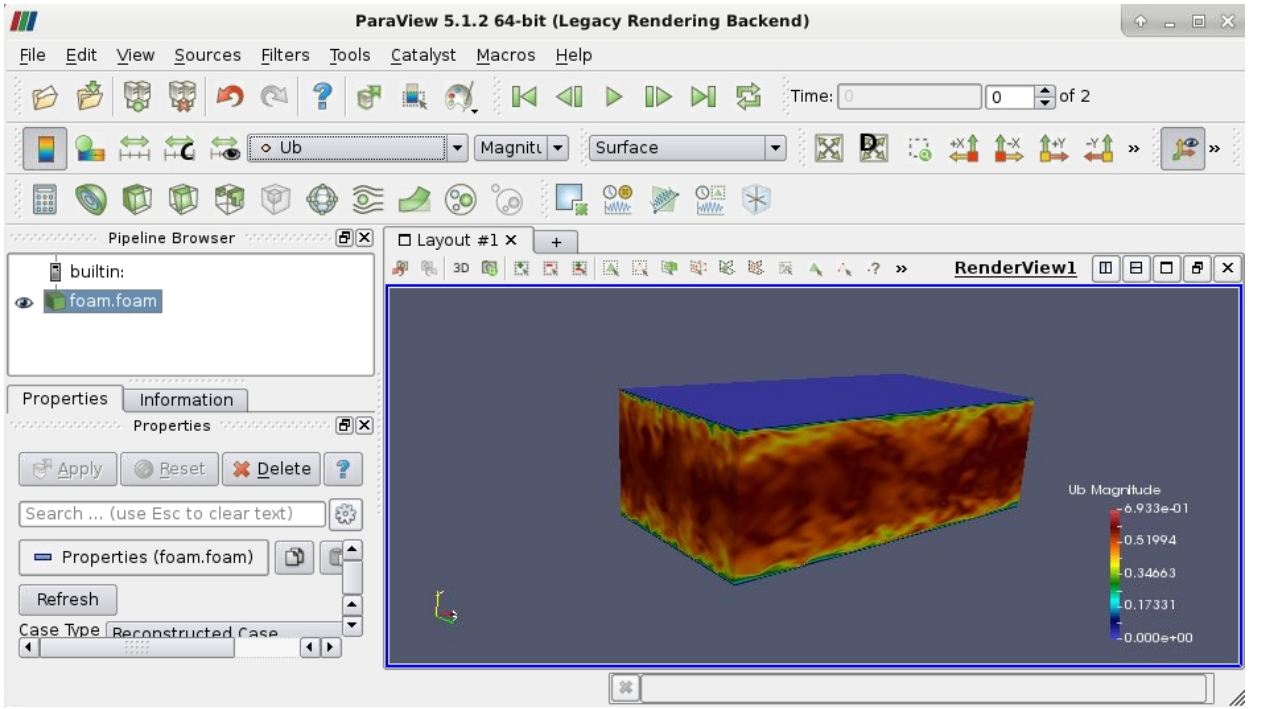
ParaView
Parallel Visualization Application

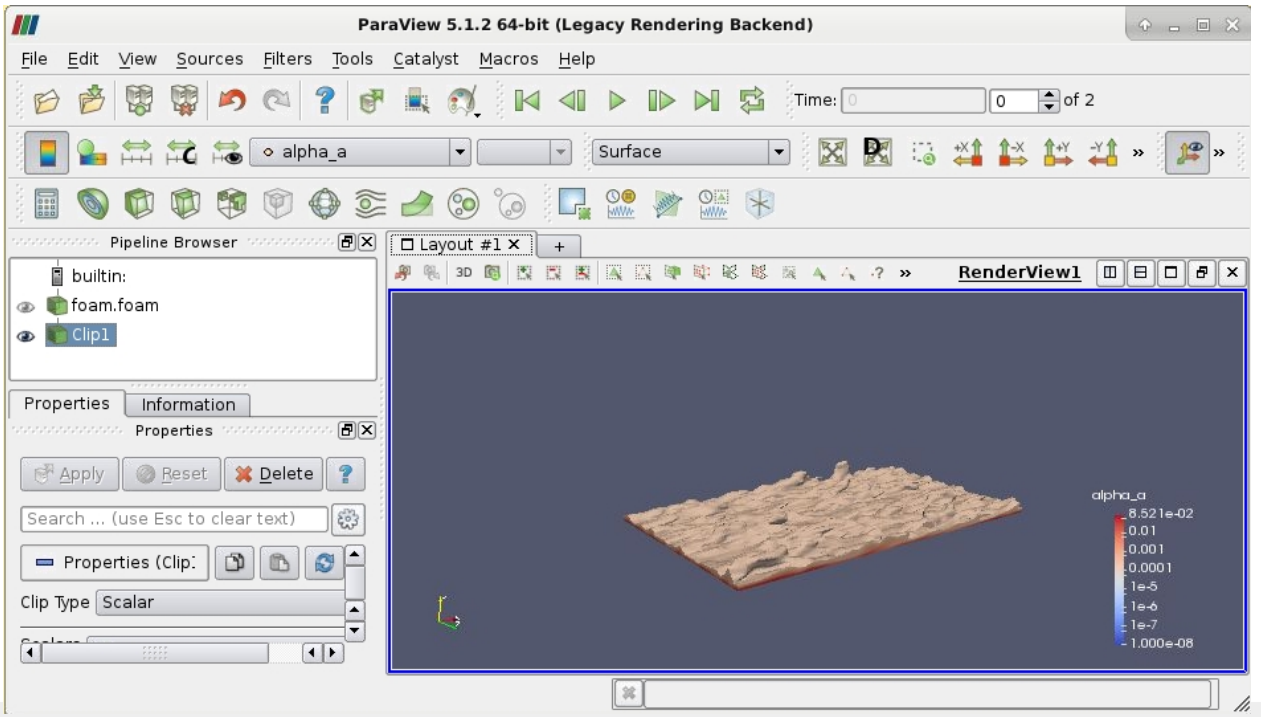
Surfaces exportées (~ 100 Mo)

↓

blender®

Image .png (~ 10 Mo)





Workflow

Fichiers d'entrée (~ 100 Ko)

↓

OpenFOAM

Champs 3D (~ 400 Mo)

↓

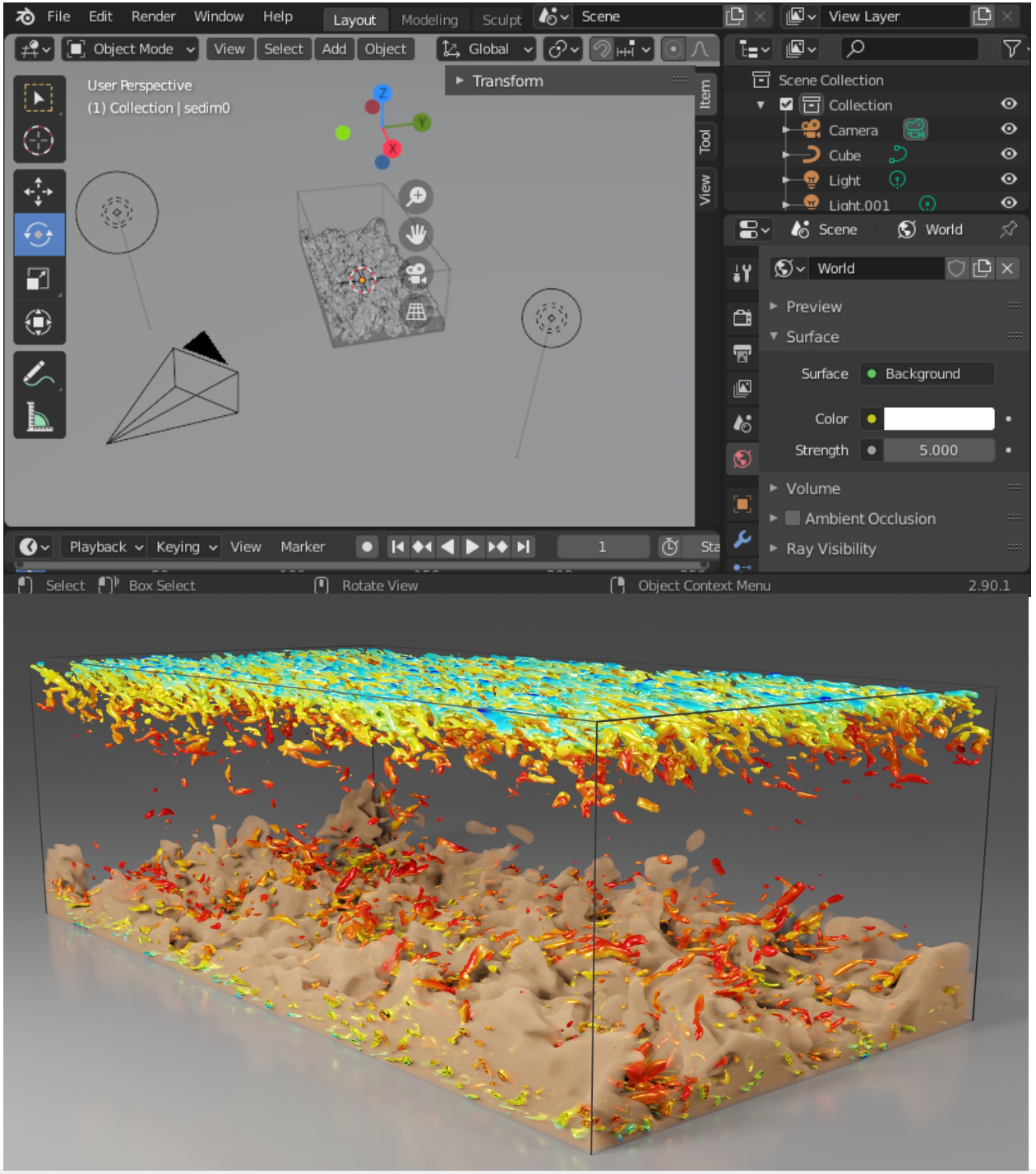
ParaView
Parallel Visualization Application

Surfaces exportées (~ 100 Mo)

↓

blender

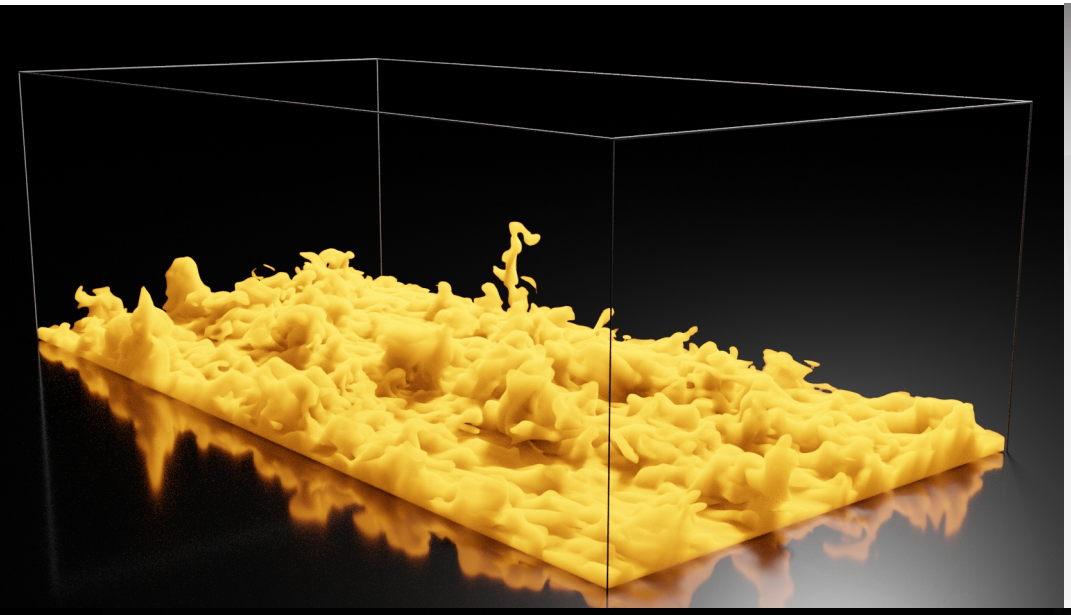
Image .png (~ 10 Mo)

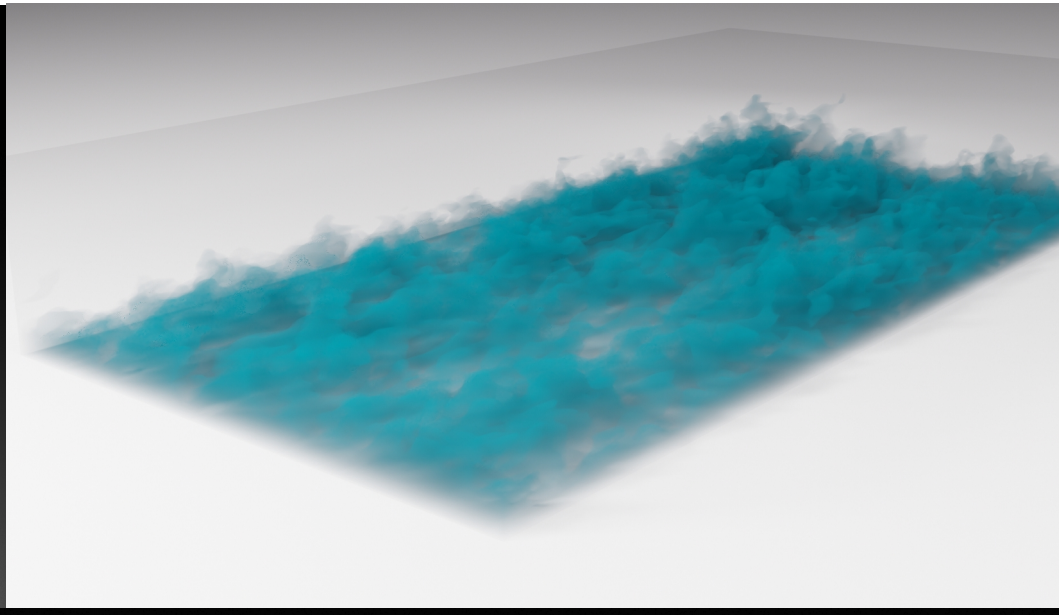


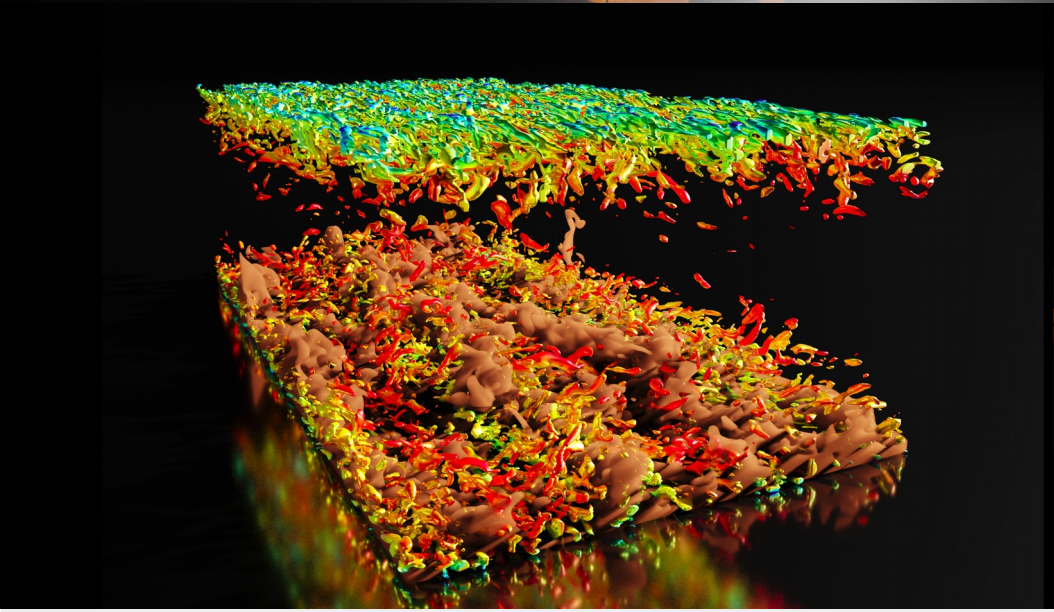
Logiciel libre de modélisation, animation et rendu 3D

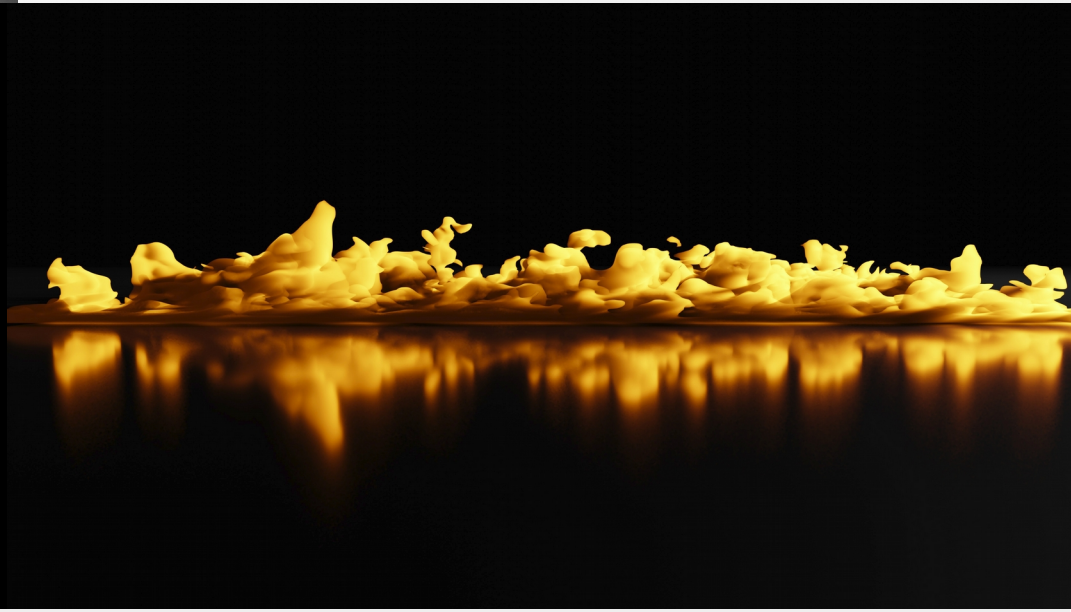
- Sculpture 3D, texturage, éclairage, etc
- Moteur de rendu photoréaliste de type **ray tracing**
- Communauté d'utilisateur très active
- Scriptable avec **python**

Exemples

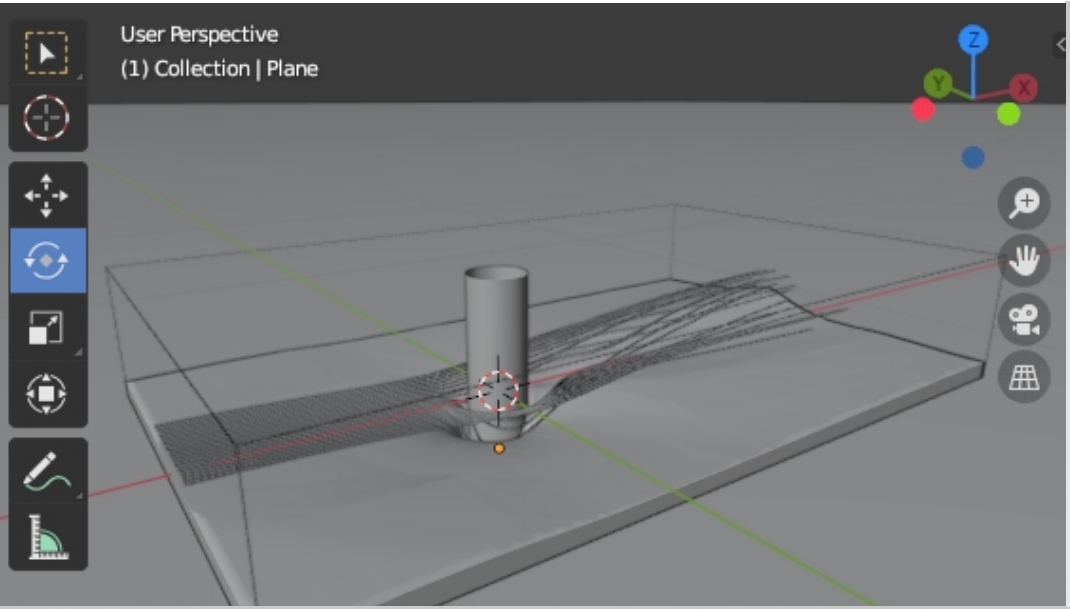


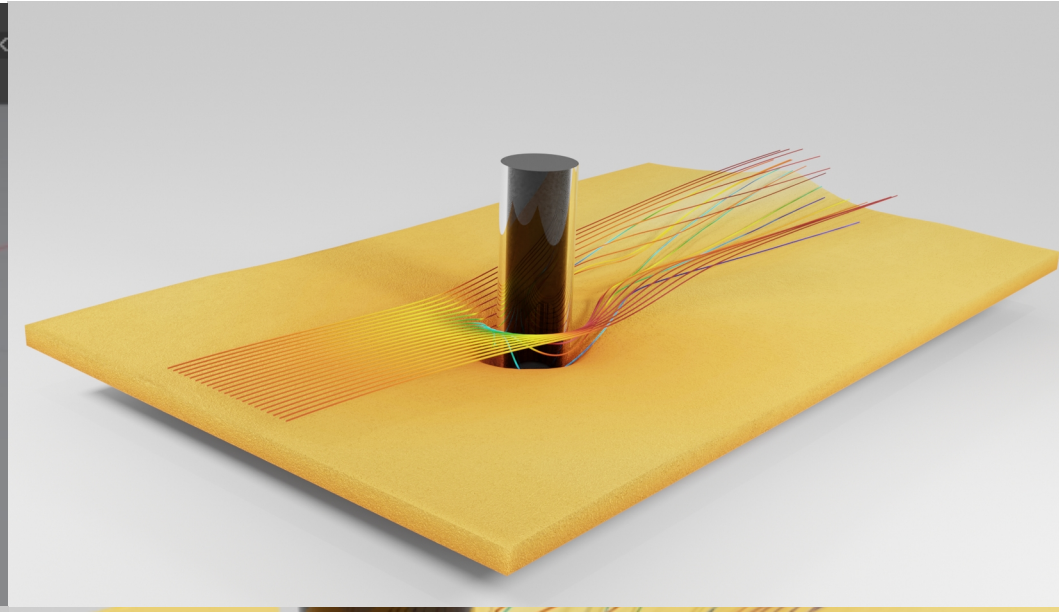


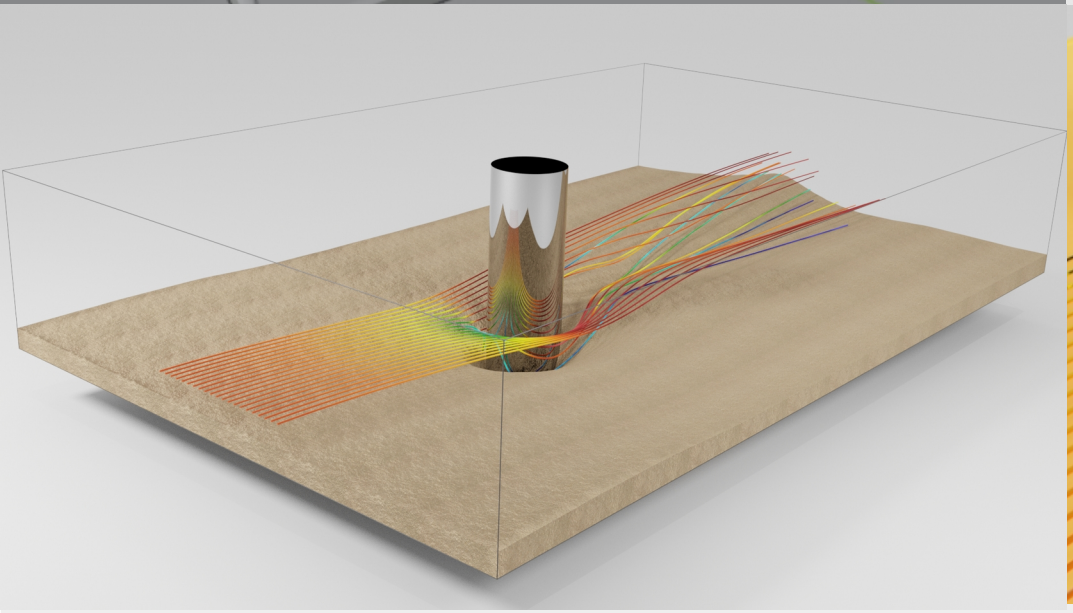


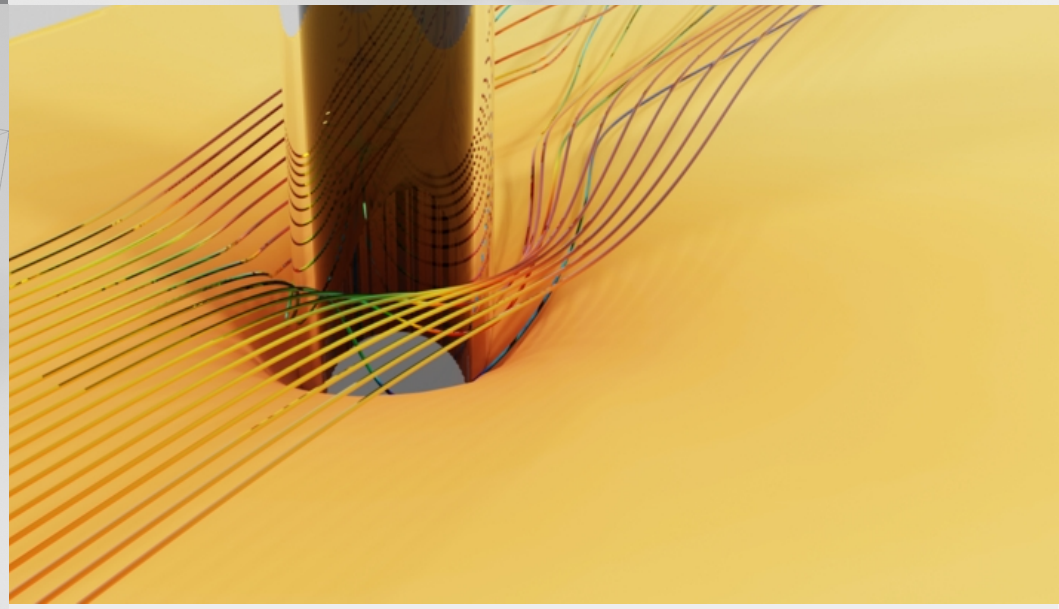


Autre exemple: erosion autour d'un cylindre





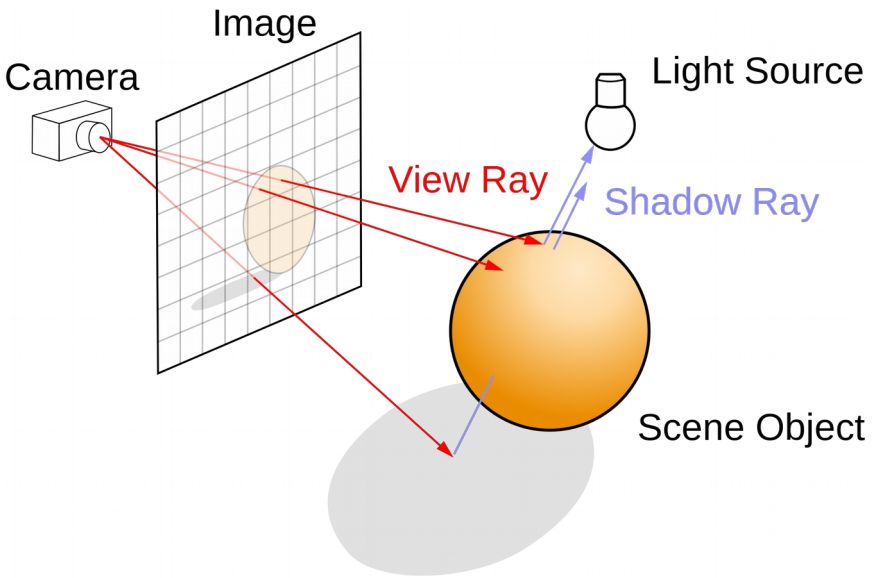




Ray tracing

Principe de fonctionnement

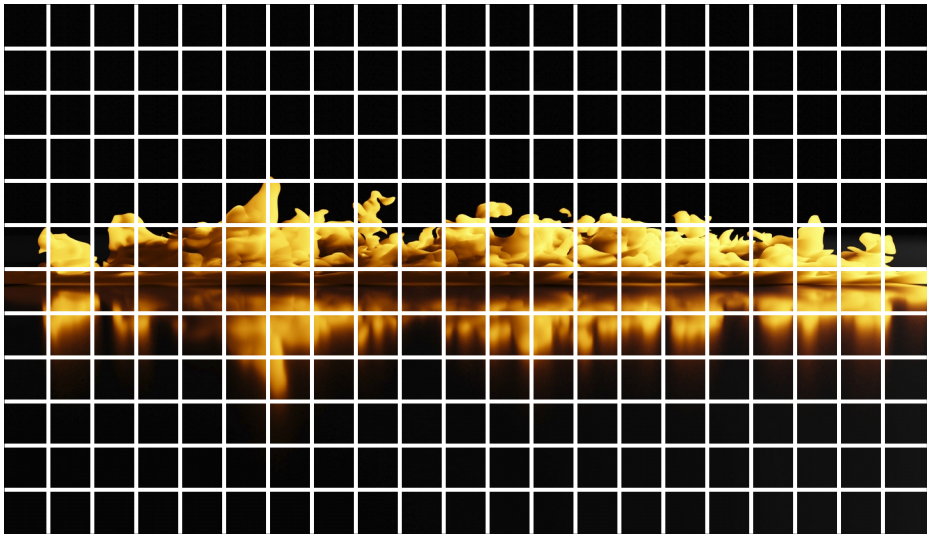
- Un rayon pour chaque pixel est lancé depuis la caméra
- Le premier point d'impact du rayon sur un objet définit l'objet concerné par le pixel correspondant
- Des rayons sont ensuite lancés depuis le point d'impact en direction de chaque source de lumière pour déterminer sa luminosité
- Cette luminosité, combinée avec les propriétés de la surface de l'objet déterminent la couleur finale du pixel



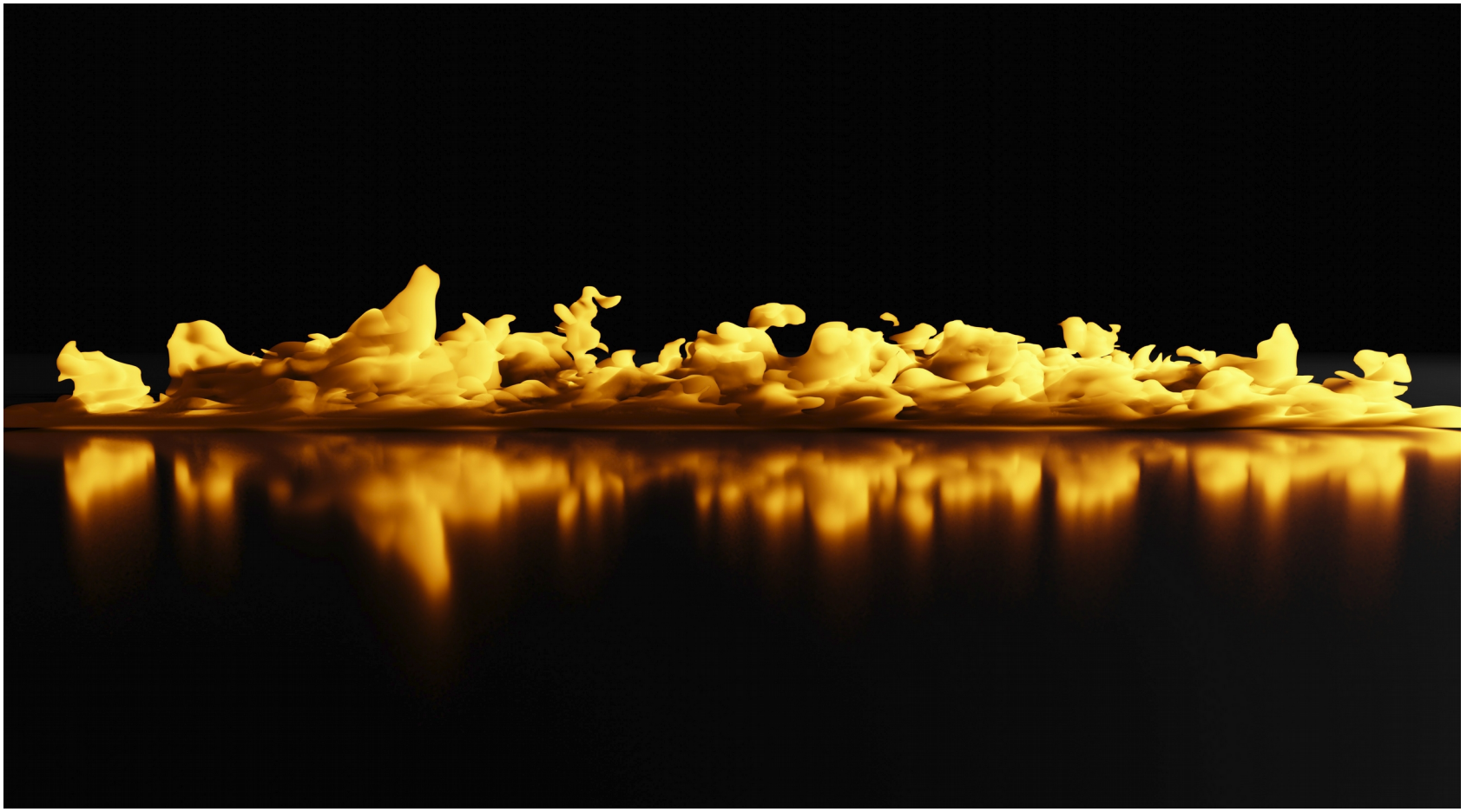
Ray tracing

Principe de fonctionnement

- L’image finale est découpée en “tuiles”
- Un CPU/GPU = 1 tuile à la fois
- Plusieurs itérations pour le rendu d’une tuile (plusieurs rayons par pixels)

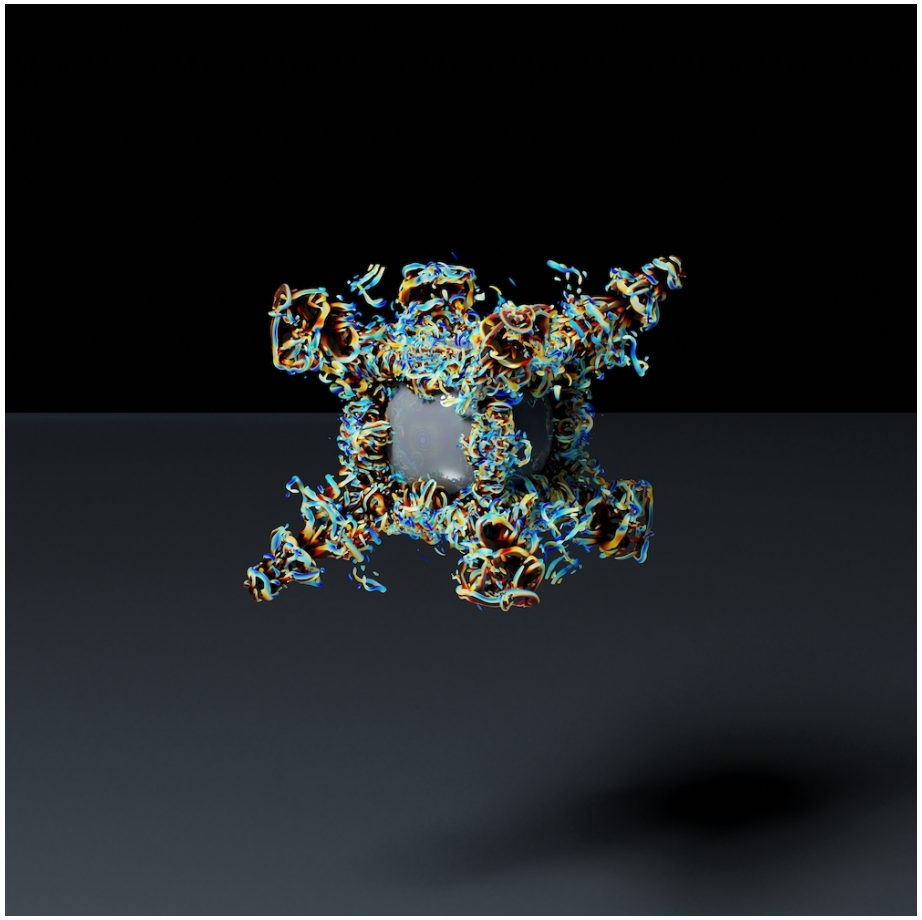


Temps de rendu (6000x4000px)



CPU (4 coeurs)	CPU (32 coeurs)	GPU (2 K80)
1h30	11min40	9min30

Temps de rendu (4000x4000px)



CPU (32 coeurs)	GPU (1 NV100)
10min	4min45

Conclusion

- La visualisation scientifique permet d'améliorer la communication et valoriser la recherche
- Blender, logiciel libre permet de fournir des visualisations 3D photoréalistes de simulations
- Le traitement des données se fait sur paraview avant d'appliquer des textures, éclairage, etc.
- Important speed-up en utilisant des GPU