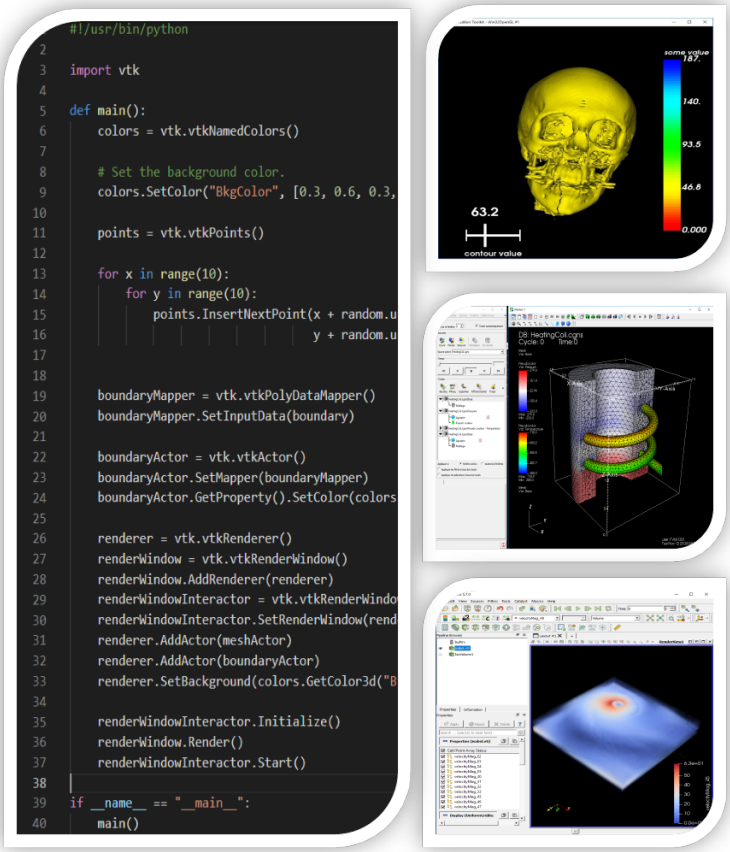
Calcul intensif et Sciences des données

Visualisation et Approches in-situ

EI9IS322



TP VTK miniapp en C++

Fabien VIVODTZEV
Benjamin FOVET

But de ce TP : intégrer VTK dans une « mini-app » permettant la lecture/écriture de maillages, leur transformation (concaténation, translation), l'évaluation de leur qualité et leur visualisation.

Mots clés : Spécifications, modélisation, développement, tests

1. Installation des outils nécessaires à la miniapp VTK en C++

Avant de commencer, assurez-vous d'avoir :

- Git
- CMake
- VTK 8.2.0
- Un compilateur C++ (g++)
- Google Test (optionnel) pour lancer les tests unitaires

2. Préparation de la structure de la miniapp

- Récupérez le code à l'adresse suivante : https://github.com/bfovet/minimesh :

```
$ git clone https://github.com/bfovet/minimesh
```

- Compilez le code :

```
$ cd minimesh
$ mkdir build && cd build
$ cmake .. -DCMAKE_CXX_STANDARD=11
$ make
```

3. Travail attendu et notation

La note de ce TP prendra en compte les éléments suivants :

- L'historique de développement (commits git) devra être compréhensible : concentrez-vous sur une fonctionnalité à la fois, committez régulièrement des modifications cohérentes,
- Le code doit respecter les standards C++ (C++11 au minimum) et doit évidemment compiler sans erreur, et si possible sans avertissements,
- Faites attention à la gestion de la mémoire et utilisez au maximum les pointeurs intelligents,
- Implémentez les spécifications telles que décrites ci-après,
- Utilisez les classes VTK appropriées,
- Privilégiez l'utilisation de classes, le C++ est un langage objet

4. Spécifications de la miniapp à développer

La miniapp devra

- Lire et écrire des maillages non structurés (vtkUnstructuredGrid)
  - Import au format .vtu (XML) et .vtk (Legacy)
  - Export au format .vtu (XML) et autres laissés au choix
- Concaténer des maillages (N fichiers en entrée vers 1 fichier en sortie) avec option de fusion des nœuds doubles (nœuds ayant les mêmes coordonnées)
- Translater des maillages (1 fichier en entrée vers 1 fichier en sortie), étant donné trois coordonnées spatiales
- Calculer la qualité d'un maillage (forme et taille des cellules) et afficher le champ scalaire associé au maillage
- Laisser l'utilisateur interagir et visualiser un maillage avec affichage des arêtes et champs scalaires

Données d'entrée

Deux fichiers d'entrées au format TOML sont fournis dans le répertoire inputs : `merge.toml` et `translate.toml`. La classe OptionParser permet de les lire.

Commandes

Pour avoir l'aide des commandes, utilisez l'option ***help*** :

```
$ ./miniapp –help
```

La commande ***view*** devra permettre de visualiser et interagir avec un maillage en prenant en argument un fichier .vtu :

```
$ ./miniapp --view mesh.vtu
```

La commande ***transform*** devra prendre en argument un fichier TOML (`merge.toml` ou `translate.toml`) et lancer l'opération souhaitée :

```
$ ./miniapp --transform merge.toml
```