

Manipulation de champs avec SALOME

Anthony GEAY (CEA/DEN), Guillaume Boulant (EDF/R&D)

Journée des Utilisateurs de SALOME
(Mardi 15 novembre 2011)



cea



La manipulation de champs

Une définition

- « *Accéder aux valeurs des champs de manière sélective (composante, temps, zone géométrique) pour l'utilisation dans des opérations mathématiques (essentiellement)* »
- Champ : grandeur physique dont la valeur dépend de l'espace et du temps = $\mathbf{F}(\mathbf{r},t)$

Quelques situations d'usage

- **Pré-traitement** : création d'un champ sur une zone géométrique pour modéliser un chargement ou des conditions aux limites
- **Inter-traitement** : projection de champs entre maillages par interpolation aux interfaces
- **Post-traitement** : calcul des grandeurs d'intérêt à partir des données brutes
- **Recalage** : changement d'échelle, d'unité, rotation, translation
- **Analyse de la qualité** : extraction de mailles et de champs en fonction d'un critère
- **Visualisation** : sous-échantillonnage spatial et/ou temporel

Dans le cadre technique de SALOME

Un modèle de donnée

- Modèle MED (Modèle d'Échange de Donnée) qui décrit les maillages et les champs
- Connue au travers de son implémentation de persistance med «fichier»

Une bibliothèque logicielle (C++, python)

- Aperçu de l'interface de programmation (API) de MED «mémoire»
- Architecture des composants (MEDCoupling, MEDLoader, REMAPPER)

Une interface graphique (module SALOME)

- Pour la mise en œuvre des cas d'usage principaux

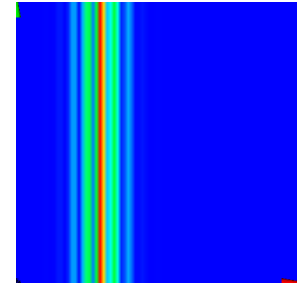
Présentation d'aujourd'hui

Une bibliothèque logicielle (C++, python)

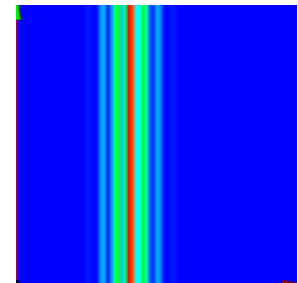
Un premier exemple d'utilisation de l'API

- Addition de champs définis sur un même maillage

```
1 from MEDLoader import MEDLoader, ON_NODES
2
3 medfilename = "timeseries.med" # med source filename
4 meshname = "Grid_80x80"       # name of the support mesh
5 dimrestriction = 0            # 0=no restriction
6 fieldname = "Pulse"           # name of the field series
7
8 # Load the support mesh
9 mesh = MEDLoader.ReadUMeshFromFile(medfilename, meshname, dimrestriction)
10
11 # Load the field at timestamps 3
12 iteration, order = (3, -1)    # timestamps to consider
13 p3=MEDLoader.ReadField(ON_NODES, medfilename, meshname, dimrestriction,
14                         fieldname, iteration, order)
15 p3.setMesh(mesh)
16
17 # Load the field at timestamps 4
18 iteration, order = (4, -1)    # timestamps to consider
19 p4=MEDLoader.ReadField(ON_NODES, medfilename, meshname, dimrestriction,
20                         fieldname, iteration, order)
21 p4.setMesh(mesh)
22
23 result = p3+p4
24 result.setName("p3+p4")
25
26 outfilename = "addition.med"
27 MEDLoader.WriteField(outfilename, result, True)
```

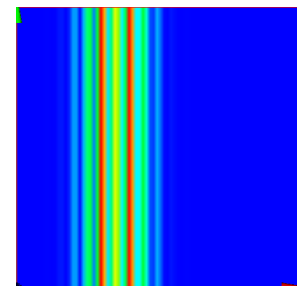


+



—

=



Autres applications de cet exemple

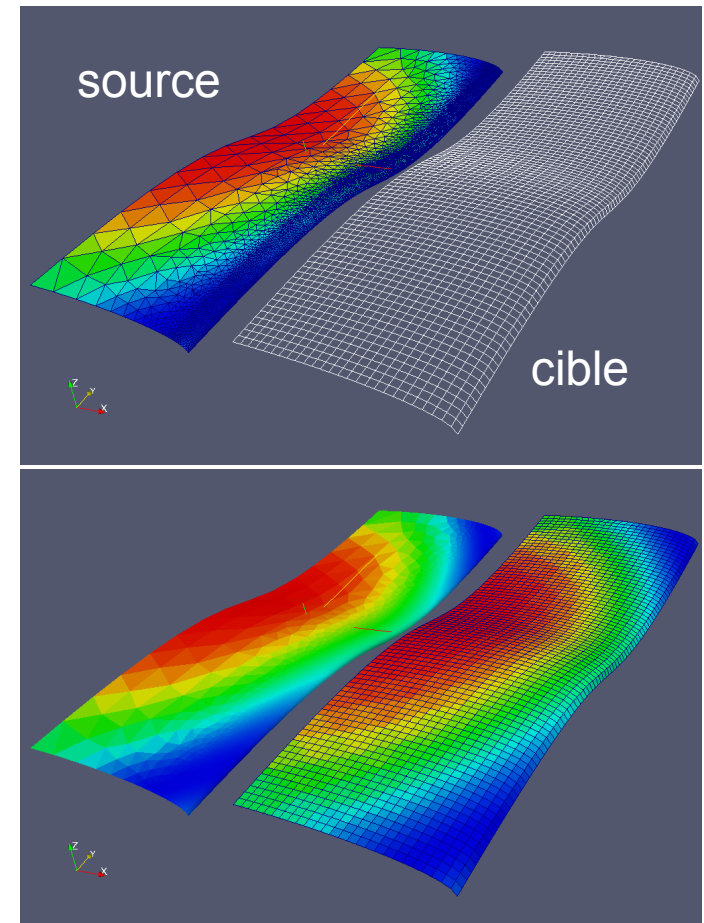
- Changement d'échelle, toute combinaison linéaire de champs

Une bibliothèque logicielle (C++, python)

Exemple 2 : Projection d'un champ entre deux maillages

- **Entrée** : champ sur un maillage source + maillage cible
- **Sortie** : le champ est créé sur le maillage cible par interpolation P0P0 (entre cellules)

```
1 from MEDLoader import *
2 from MEDCouplingRemapper import *
3
4 # Read the source mesh and create a field on it
5 msource = MEDLoader.ReadUMeshFromFile("meshsource.med", "meshsource", 0)
6
7 equation = "319.*cos(((x)*(x)*3+(y-0.52)*(y-0.52)+(z-0.1)*(z-0.1))*7)"
8 fsource=msource.fillFromAnalytic(ON_CELLS,1,equation)
9 fsource.setName("Temperature")
10 fsource.setNature(ConservativeVolumic)
11
12 # Read the target mesh
13 mtarget = MEDLoader.ReadUMeshFromFile("meshtarget.med", "meshtarget", 0)
14
15 # Remapper of type P0P0 (interpolation from cells to cells)
16 remap = MEDCouplingRemapper()
17 remap.prepare(msource,mtarget,"P0P0")
18
19 defaultValue = 1e100
20 ftarget = remap.transferField(fsource,defaultValue)
21 ftarget.setName("Temperature")
22
23 outfilehame = "createsource_fielddtarget.med"
24 MEDLoader.WriteField(outfilehame,ftarget,True)
```



Types d'interpolation disponibles :

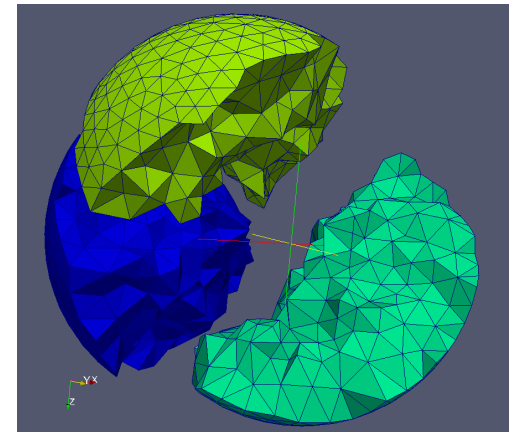
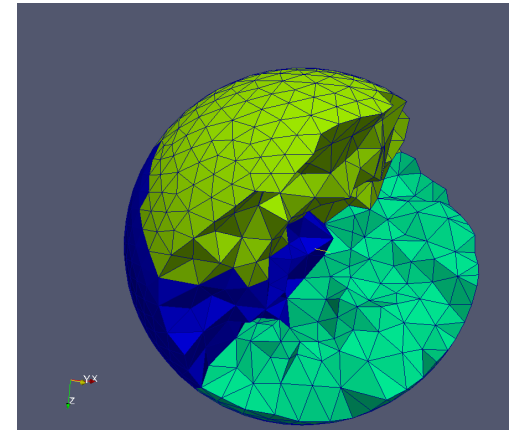
- P0P0, P1P0, P0P1, P1P1
- 1D, 2Dcurve, 2D, 3Dsurf, 3D
- Pour P0P0 : + (2Dcurve,2D) et (3Dsurf, 3D)

Une bibliothèque logicielle (C++, python)

Exemple 3 : fusion de champs définis sur des domaines distincts

- **Entrée** : 3 domaines sur chacun desquels est défini un champ
- **Sortie** : 1 domaine unique avec un champ unique

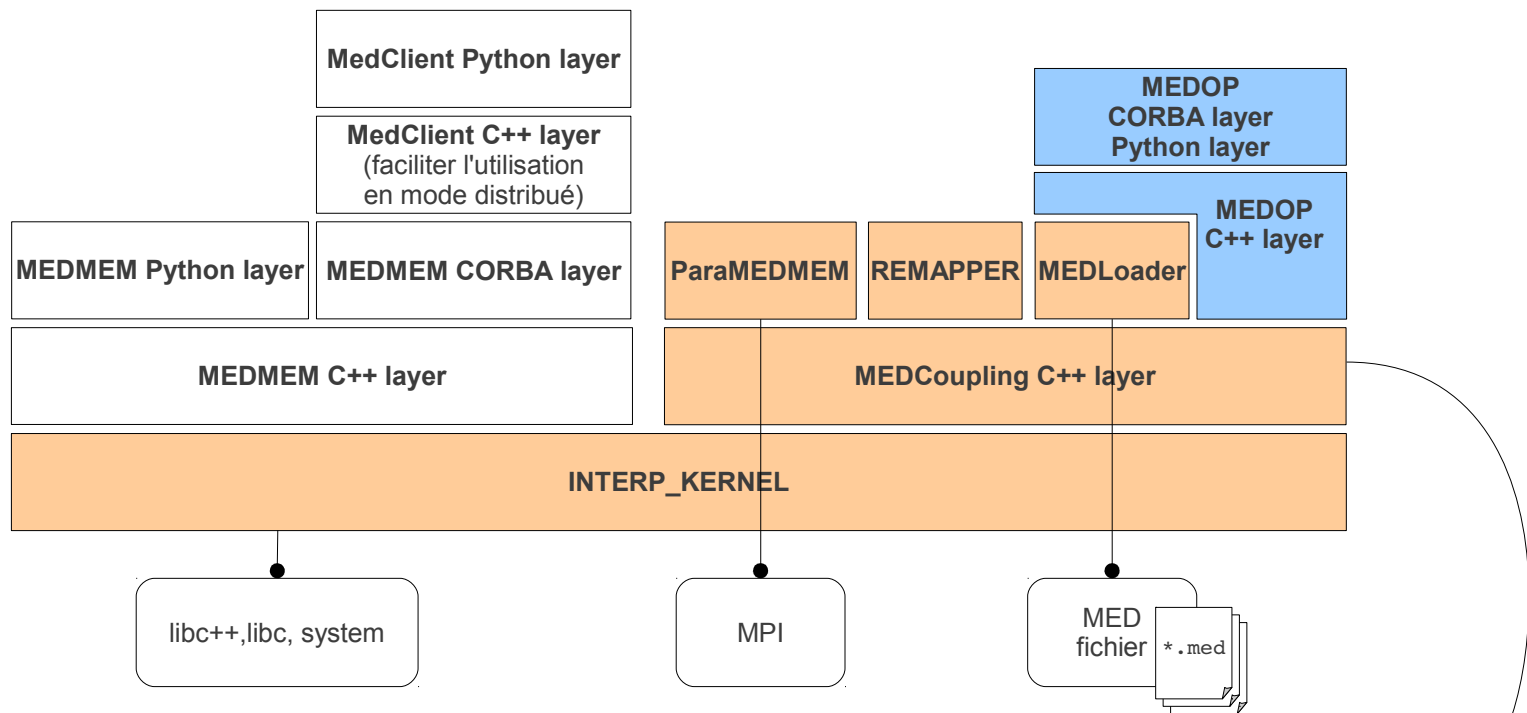
```
1 from MEDLoader import *
2
3 it,dt = (-1,-1)
4 mesh = MEDLoader.ReadUMeshFromFile("field1.med","DomainMesh_1",0)
5 f1 = MEDLoader.ReadField(ON_CELLS,"field1.med","DomainMesh_1",0,"field1",it,dt)
6 f1.setMesh(mesh)
7
8 mesh = MEDLoader.ReadUMeshFromFile("field2.med","DomainMesh_2",0)
9 f2 = MEDLoader.ReadField(ON_CELLS,"field2.med","DomainMesh_2",0,"field2",it,dt)
10 f2.setMesh(mesh)
11
12 mesh = MEDLoader.ReadUMeshFromFile("field3.med","DomainMesh_3",0)
13 f3 = MEDLoader.ReadField(ON_CELLS,"field3.med","DomainMesh_3",0,"field3",it,dt)
14 f3.setMesh(mesh)
15
16 fmerge=MEDCouplingFieldDouble.MergeFields([f1,f2,f3])
17 fmerge.setName("merge")
18
19 MEDLoader.WriteField("merge.med",fmerge,True)
```



Une bibliothèque logicielle (C++, python)

Architecture des composants

- **MEDCoupling** : structures de données et interfaces d'échange (CORBA)
- **MEDLoader** : services de lecture et écriture aux formats MED fichier et SAUV (Castem)
- **REMAPPER** : outils d'interpolation



- Orienté couplage et interpolation, compacité des données, compatibilité avec la visualisation (VTK)
- Aucune dépendance logicielle, sinon les standards libc, libc++ et system

- 300 tests C++, 210 tests python, 30 tests CORBA (valgrind 0, no warnings)

Une interface graphique (module SALOME)

Statut actuel

- En cours de développement, prévu à l'intégration dans SALOME 7 (fin 2012)
- Un prototype mis au point sur la base des 8 cas d'usage ci-dessous

Les cas d'utilisation:

- Exemple 1: Explorer des sources de données
- Exemple 2: Rassembler des champs issus de différentes sources
- Exemple 3: Appliquer une opération mathématique sur des champs
- Exemple 4: Comparer des champs issues de différentes sources
- Exemple 5: Créer un champ sur un domaine spatial
- Exemple 7: Créer un champ à partir d'une image to[mp]ographique
- Exemple 8: Continuer l'analyse dans PARAVIS

Une interface graphique (module SALOME)

Exemple 1: Explorer des sources de données

The screenshot displays the SALOME 6.3.1 interface. The main window shows a 3D visualization of a pulse field on a grid. A context menu is open over the 'it=3' item in the 'Arbre d'étude' tree, with 'Visualize' selected. The 'Console Python' window shows the execution of Python commands to inspect the data source.

Arbre d'étude (Study Tree):

- XMED
 - timeseries.med
 - Grid_80x80
 - Pulse (ON_NODES)
 - it=0
 - it=1
 - it=2
 - it=3**
 - it=4
 - it=5
 - it=6
 - it=7
 - it=8
 - it=9
- Post-Pro

Console Python:

```
Python 2.6.6 (r266:84292, Apr 26 2011, 20:36:01)
[GCC 4.4.5] on linux2
type help to get general information on environment
>>> import xmed
>>> xmed.setConsoleGlobals(globals())
>>> from xmed import load, get, put, dup, ls, la, save, view, doc, wipe
>>> view(get(3))
field name (id)          = Pulse (3)
mesh name (id)          = Grid_80x80 (0)
discretization          = ON_NODES
(iter, order)           = (3,-1)
data source              = file:///home/gboullant/development/projets/salome/MEDOP/XMED/xmed/resources/datafiles/timeseries.med
>>>
```

Une interface graphique (module SALOME)

Exemple 2: Rassembler des champs issus de différentes sources

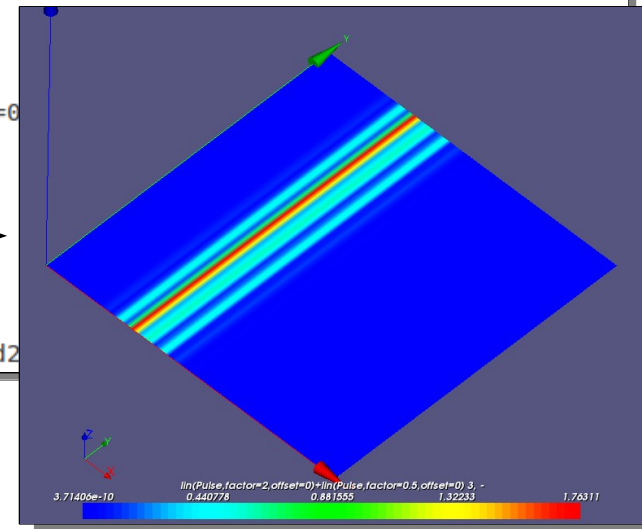
The image illustrates the process of saving workspace data in SALOME. The main window shows a 3D visualization of a pulse field on a mesh. A context menu is open over the 'medop_view' object, with 'Use in workspace' selected. A dialog box 'Sélection d'un alias pour le champ' is shown, allowing the user to define an alias 't2' for the selected field. A 'Save workspace data' dialog is also open, showing the file 'output.med' being saved in the directory '/home/gboulant'. The console window shows the execution of commands to save the workspace data.

```
>>> p3
field name (id) = Pulse (3)
mesh name (id) = Grid_80x80
discretization = ON_NODES
(iter, order) = (3, 1)
data source = file:///home/gboulant/development/projets/salome/MEDOP/XMED/xmed/resources/datafiles/smallmesh_varfield.med
>>> t2=xmed.field.proxy.newFieldPro
field name (id) = testfield2
mesh name (id) = My2DMesh
discretization = ON_NODES
(iter, order) = (1, 1)
data source = file:///home/gboulant/development/projets/salome/MEDOP/XMED/xmed/resources/datafiles/smallmesh_varfield.med
>>> save('/home/gboulant/output.med')
>>>
```

Une interface graphique (module SALOME)

Exemple 3: Appliquer une opération mathématique sur des champs

```
>>> p3|
field name (id) = Pulse (3)
mesh name (id) = Grid_80x80 (0)
discretization = ON_NODES
(iter, order) = (3,-1)
data source = file:///home/gboulant/development/projets/salome/MEDOP/XMED/xmed/resources/datafiles/timeseries.med
>>>
>>> r=p3+10 # les valeurs de r sont celles du champ p3 augmentées d'un offset de 10
>>> r2=10*p3 # changement d'échelle d'un facteur 10
>>> r3=2*p3+p4/2 # combinaison linéaire des champs p3 et p4
>>> r4=pow(p3,5) # les valeurs de r4 sont celles de p3 élevées à la puissance 5
>>>
>>> ls
r2 (id=13, name=lin(Pulse,factor=10,offset=0))
r4 (id=17, name=fct(Pulse,"abs(u)^5,"-1"))
p3 (id=3, name=Pulse)
p4 (id=4, name=Pulse)
r3 (id=16, name=lin(Pulse,factor=2,offset=0)+lin(Pulse,factor=0.5,offset=0))
t2 (id=11, name=testfield2)
r (id=18, name=lin(Pulse,factor=1,offset=10))
>>>
>>> view(r3)
>>>
>>> p3+t2
INF: Addition of Pulse and testfield2
ERR: ERROR: Mesh ids are different for the field operands Pulse and testfield2
```



Une interface graphique (module SALOME)

Exemple 4: Comparer des champs issues de différentes sources

The screenshot shows the SALOME 6.3.1 interface with the following components:

- Arbre d'étude (Study Tree):** Shows a hierarchy of objects: XMED, parametric_01.med (mesh1, field1), parametric_02.med (mesh2, field2), and Post-Pro. A context menu is open over 'field1', with 'Change underlying mesh' selected.
- Scène VTK:** Three visualization windows showing a 3D field on a mesh. A dialog box 'Remplacement du maillage support' is open, prompting for a mesh selection. The selected mesh is 'mesh1'.
- Workspace:** A table showing the state of objects in the workspace.
- Console Python:** A Python script showing the execution of commands to create a duplicate of field2 on mesh1.

Workspace Table:

Name	Value
mesh1	No value
field1 (ON_NODES)	No value
it = 3	No value
dup(field2) (ON_NODES)	No value
it = 3	No value

Console Python:

```
>>> ls
a3      (id=3, name=field1)
b3      (id=10, name=dup(field2))

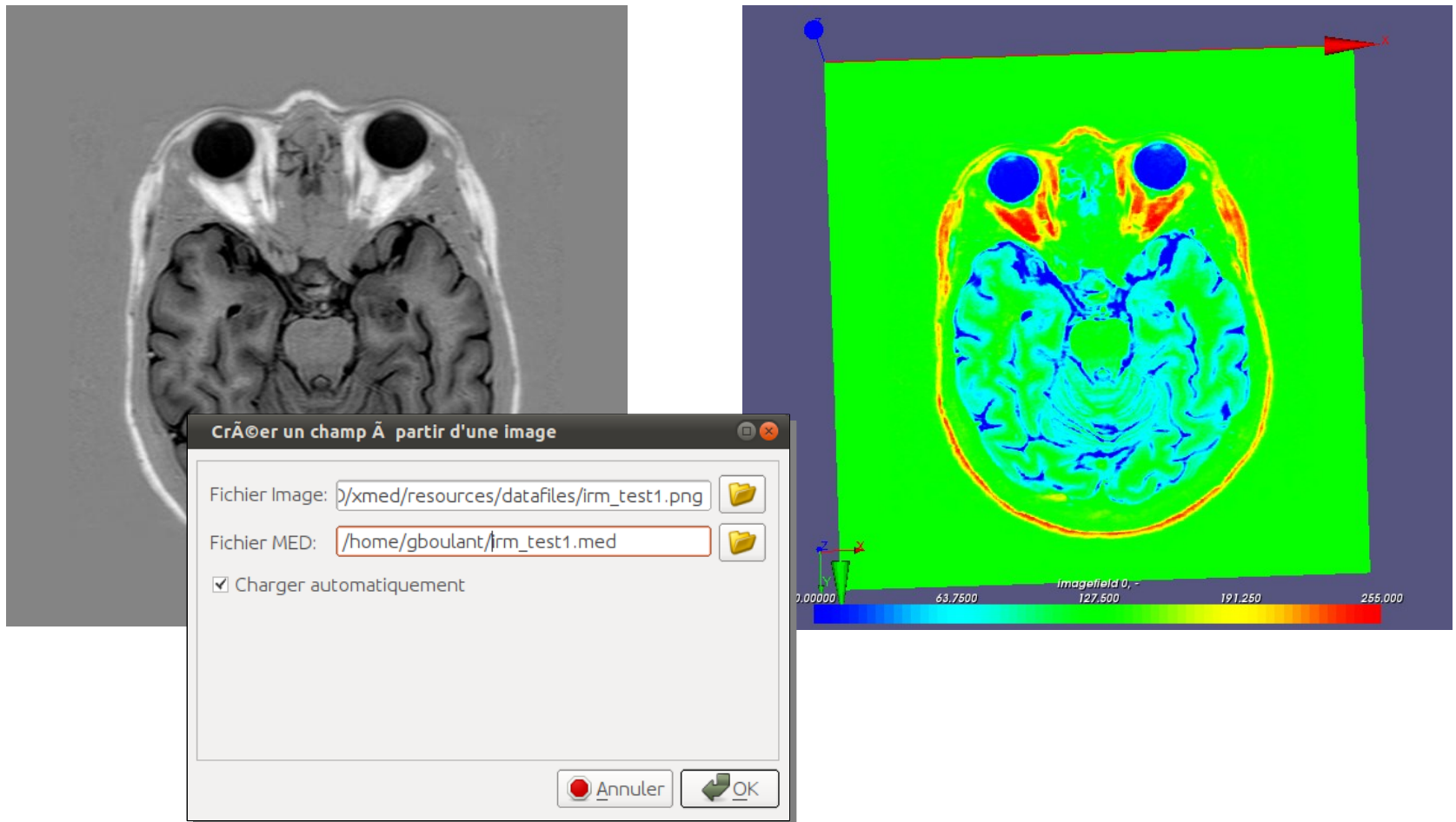
>>> d=a3-b3
INF: Substraction of field1 by dup(field2)
field name (id)      = field1-dup(field2) (11)
mesh name (id)       = mesh1 (0)
discretization       = ON_NODES
(iter, order)        = (3,-1)
data source           = mem://field1-dup(field2)
>>> view(d)
>>>
```

Field Visualization: A 3D plot of a field on a mesh. The color scale ranges from 0.00235534 (blue) to 0.148146 (red). The field is labeled 'field1-dup(field2) 3, -'.

Une interface graphique (module SALOME)

Exemple 7: Créer un champ à partir d'une image to[pm]ographique

- Image to[pm]ographique = champ scalaire (niveaux de gris) sur grille cartésienne



Conclusions et perspectives

Conclusions

- Une bibliothèque logicielle (C++,python) utilisable ...
 - ... en « scripting » pour les cas d'usage listés en début de présentation
 - ... en dépendance logicielle d'un code de calcul (MEDCoupling vient sans pré-requis)
 - Déjà disponible en version industrielle dans SALOME 6
- Une interface graphique (module SALOME)
 - Combinaison GUI – TUI inspirée des logiciels comme Octave ou Matlab
 - Assister au maximum la sélection des données à manipuler (GUI) pour offrir une syntaxe de commande (TUI) la plus proche de l'écriture formelle

Perspectives

- Distribution « packagée » dans SALOME 7 (début 2013) sous forme du module MED :
 - Bibliothèque MEDMEM à base de MEDCoupling (MEDMEM historique disparaît)
 - Interface Graphique version industrialisée
- En prévision pour l'interface graphique
 - Développement des moyens de sélection des données (groupe de mailles, composantes, sous-échantillonnage temporel)
 - Extension de la liste des fonctions mathématique disponibles dans le langage de commande TUI
 - Intégration graphique des outils pour la projection de champs

Fin de la présentation

« Regarde comme les étoiles brillent.
Pourtant certaines sont mortes il y a longtemps déjà.
Mais elles éclairent encore le ciel, chaque nuit. »

Conte pour enfants