

Orientations et Premiers Développements pour la Plate-forme de Données BIGMECA

A. Marano

COPIL des Travaux de la Chaire BIGMECA

2 juin 2020



- Acquisition de données massives et multimodales

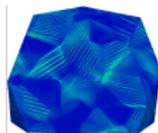
EBSD



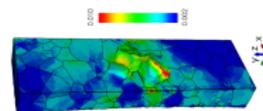
in situ DCT automatisée



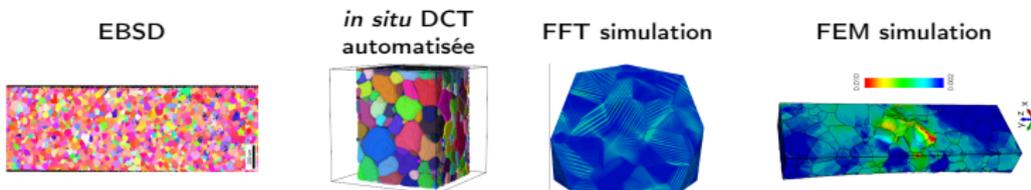
FFT simulation



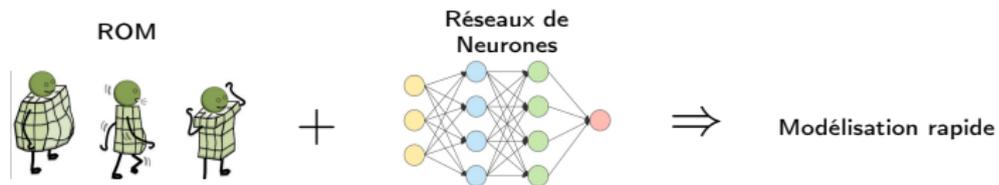
FEM simulation



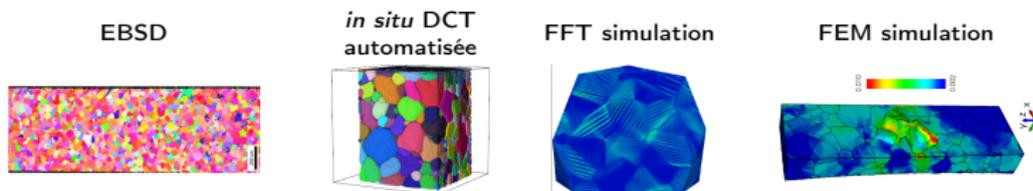
- Acquisition de données massives et multimodales



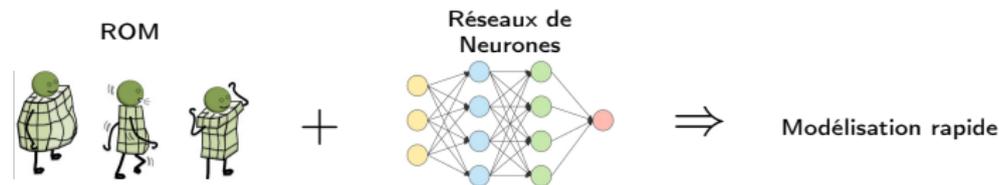
- Avancées en modélisation



- Acquisition de données massives et multimodales



- Avancées en modélisation



- Partage et valorisation des données matériaux
 - OpenData et Transparence des processus
 - Bases de données accessibles pour la communauté sc.

ACCUMULATION

Experiments

Numerical
simulation

Numerical
microstructure
generation

Au cœur des travaux : le(s) flux de données

ACCUMULATION

Experiments

Numerical
simulation

Numerical
microstructure
generation

MULTIMODAL DATASET

Au cœur des travaux : le(s) flux de données

ACCUMULATION

Experiments

Numerical
simulation

Numerical
microstructure
generation

MULTIMODAL DATASET

PROCESSING

Statistical
analysis

Image
processing

ML
pre-processing

Data
visualization

Au cœur des travaux : le(s) flux de données

ACCUMULATION

Experiments

Numerical
simulation

Numerical
microstructure
generation

MULTIMODAL DATASET

PROCESSING

Statistical
analysis

Image
processing

ML
pre-processing

Data
visualization

M. LEARNING

ROM

Neural
networks

Metamodels

Au cœur des travaux : le(s) flux de données

ACCUMULATION

Experiments

Numerical simulation

Numerical microstructure generation

MULTIMODAL DATASET

PROCESSING

Statistical analysis

Image processing

ML pre-processing

Data visualization

M. LEARNING

ROM

Neural networks

Metamodels

MODELING

Statistically representative models

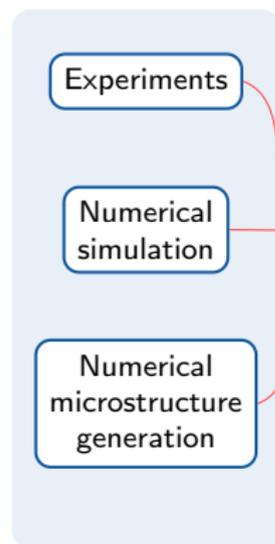
Fast mechanical predictions

Uncertainty quantification

Inverse Problems

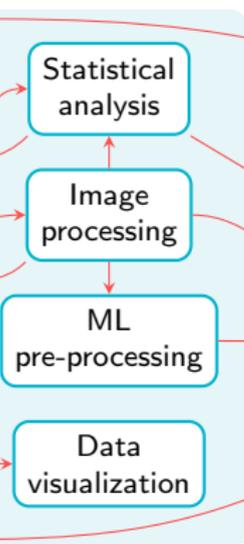
Au cœur des travaux : le(s) flux de données

ACCUMULATION

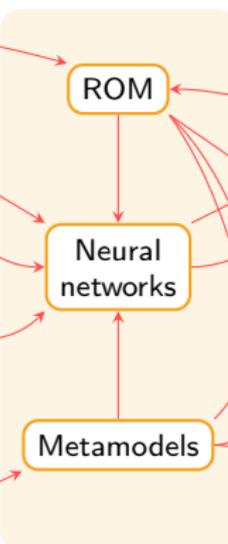


MULTIMODAL DATASET

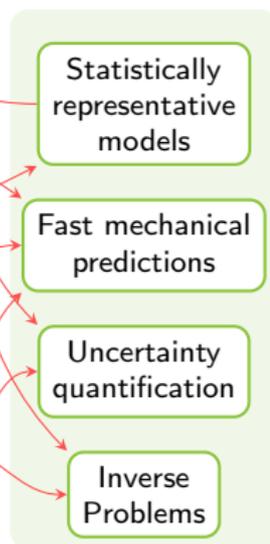
PROCESSING



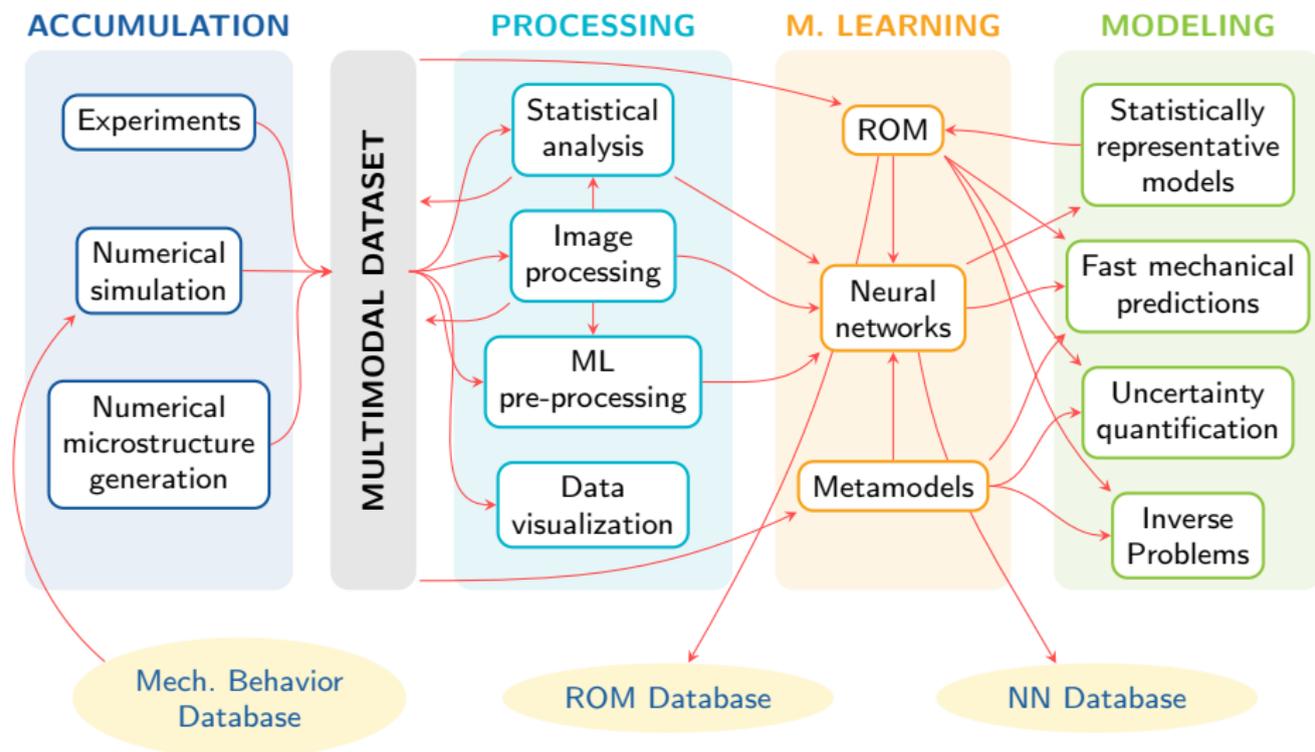
M. LEARNING



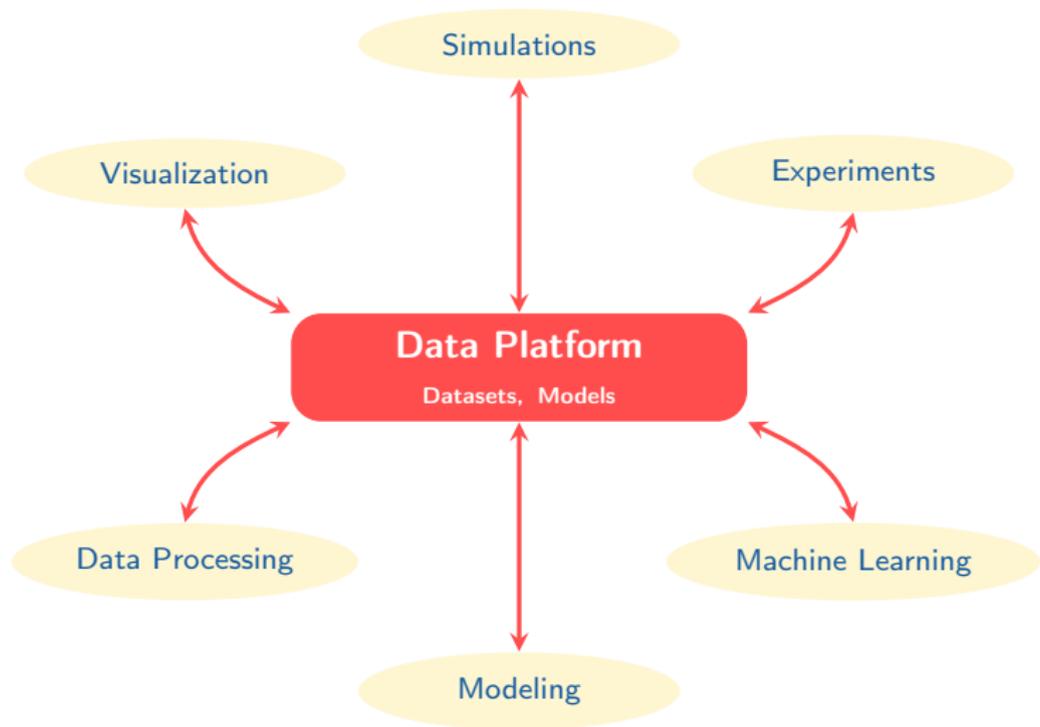
MODELING



Au cœur des travaux : le(s) flux de données



Solution : Plate-forme de données uniformisée



- 1 Traitement et gestion de données massives et multimodales
- 2 Revue Bibliographique
- 3 Orientations proposées et Démonstrateurs
- 4 Conclusion et Perspectives

- 1 **Traitement et gestion de données massives et multimodales**
 - 1.1 Problèmes à résoudre
 - 1.2 Plateforme de données : Analyse fonctionnelle
- 2 Revue Bibliographique
- 3 Orientations proposées et Démonstrateurs
- 4 Conclusion et Perspectives

- ① Traitement et gestion de données massives et multimodales
 - 1.1 Problèmes à résoudre
 - 1.2 Plateforme de données : Analyse fonctionnelle

Multimodalité : Nature

- Expérimentale :
EBSD, DCT, XCT, MEB
- Simulation : FFT, FEM
- Génération numérique de
microstructure
- Données CAO

Multimodalité : Nature

- Expérimentale : EBSD, DCT, XCT, MEB
- Simulation : FFT, FEM
- Génération numérique de microstructure
- Données CAO

Multimodalité : Forme

- Organisation : Images, Maillages...
- Système de coordonnées (repère, échelle)
- Dimension (2D, 3D, 4D...)
- Encodages, Format de Fichiers

Multimodalité : Nature

- Expérimentale : EBSD, DCT, XCT, MEB
- Simulation : FFT, FEM
- Génération numérique de microstructure
- Données CAO

Multimodalité : Forme

- Organisation : Images, Maillages...
- Système de coordonnées (repère, échelle)
- Dimension (2D, 3D, 4D...)
- Encodages, Format de Fichiers

Taille des données

- 1 Scan DCT $\sim 100Go$
- Simulation FFT/FEM $\sim 5Go$
- Dizaines d'échantillons \rightarrow Dizaines de To de données à gérer

- Comparaison visuelle de données issues de modalités différentes :
 - Recalage des repères et des échelles
 - Format et outil de visualisation unifié

- Comparaison visuelle de données issues de modalités différentes :
 - Recalage des repères et des échelles
 - Format et outil de visualisation unifié
- Compatibilité avec des outils usuels de visualisation / simulation au Cdm
 - Paraview (*.vtk*)
 - Zmaster / Zset (*.geof*)
 - Matlab (*.mat*)
 - Amitex_ FFTP (*.vtk*)

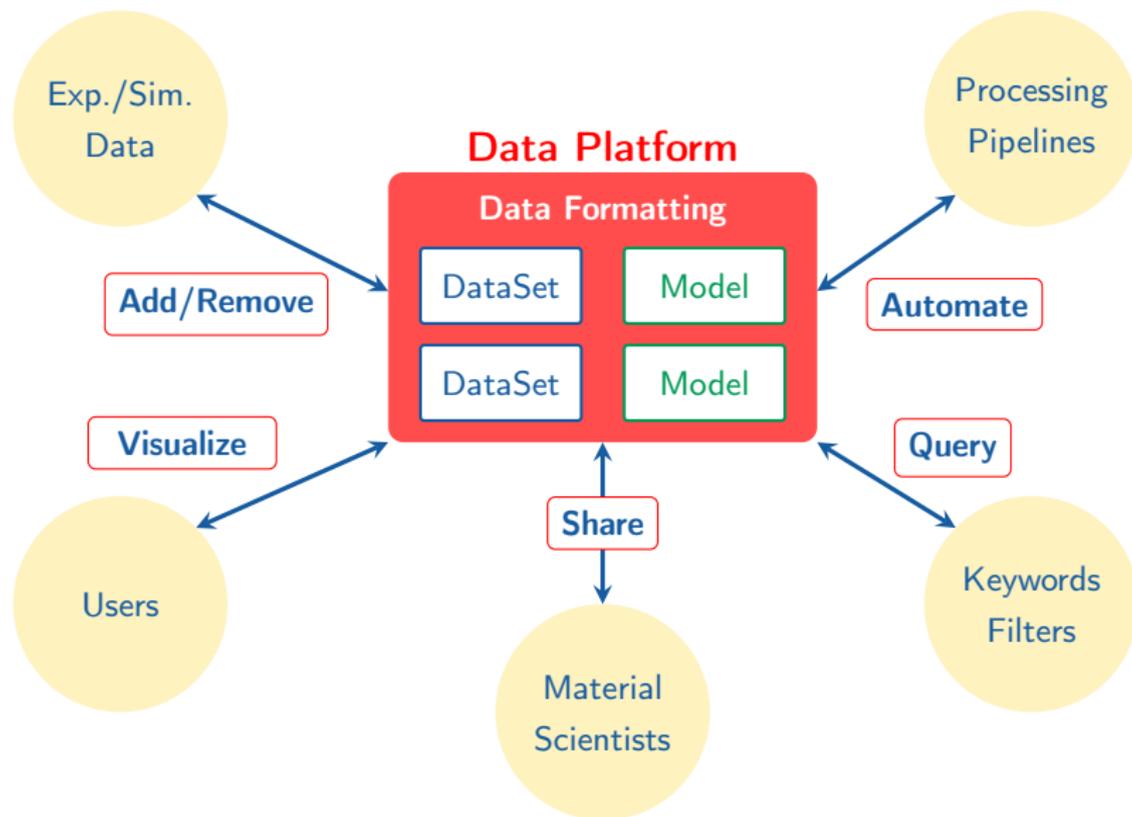
- Comparaison visuelle de données issues de modalités différentes :
 - Recalage des repères et des échelles
 - Format et outil de visualisation unifié
- Compatibilité avec des outils usuels de visualisation / simulation au Cdm
 - Paraview (*.vtk*)
 - Zmaster / Zset (*.geof*)
 - Matlab (*.mat*)
 - Amitex_ FFTP (*.vtk*)
- Manipulation et stockage efficace
 - Compression
 - Accès rapide et sélectif

- Open Science
 - Accès libre aux données pour la communauté scientifique
 - Métadonnées relatives aux traitements de données effectués
 - {Données + Métadonnées} → Reproductibilité, Transparence
 - Format d'échange standard, "universel"

- Open Science
 - Accès libre aux données pour la communauté scientifique
 - Métadonnées relatives aux traitements de données effectués
 - {Données + Métadonnées} → Reproductibilité, Transparence
 - Format d'échange standard, "universel"

- Accès distant aux données
 - Plateforme WEB
 - Serveur de stockage

- ① Traitement et gestion de données massives et multimodales
 - 1.1 Problèmes à résoudre
 - 1.2 Plateforme de données : Analyse fonctionnelle



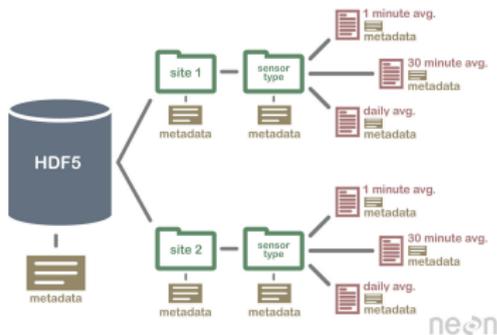
- 1 Traitement et gestion de données massives et multimodales
- 2 **Revue Bibliographique**
 - 2.1 Format HDF5
 - 2.2 Initiatives similaires : Tour d'horizon
- 3 Orientations proposées et Démonstrateurs
- 4 Conclusion et Perspectives

2 Revue Bibliographique

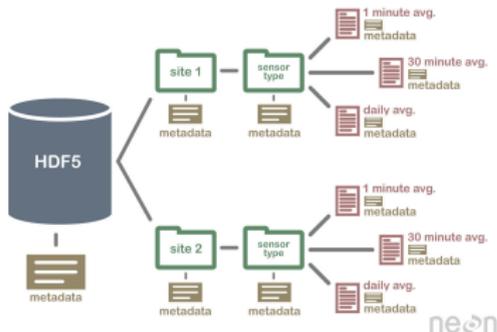
2.1 Format HDF5

2.2 Initiatives similaires : Tour d'horizon

- Format devenu standard dans la communauté sc.
 - Extensible : pas de limite de taille, ni de nombre de contenus
 - Autodescriptif : Métadonnées pour chaque élément
 - Hiérarchique : groupes, datasets et liens entre eux (arbre)



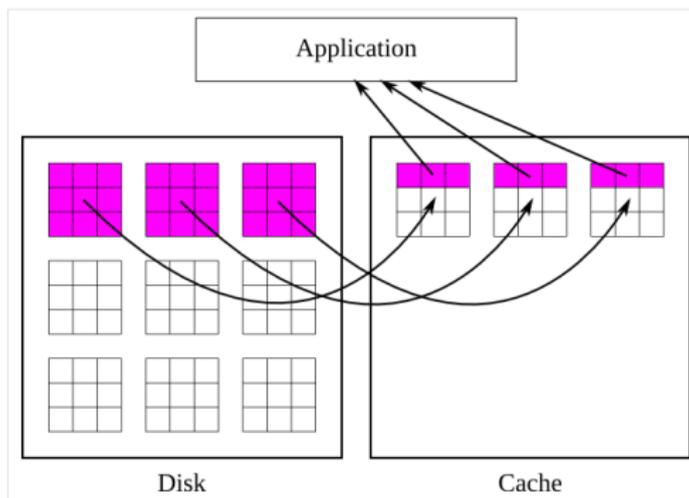
- Format devenu standard dans la communauté sc.
 - Extensible : pas de limite de taille, ni de nombre de contenus
 - Autodescriptif : Métadonnées pour chaque élément
 - Hiérarchique : groupes, datasets et liens entre eux (arbre)



- Nombreuses API et format associés existants
 - h5py, PyTables
 - Nexus, EOS, DREAM3D
 - Format XDMF (HDF5 + XML) → Visualisable avec Paraview

- Librairie HDF5
 - Stockage par morceaux (chunk storage)
 - Algorithmes de compression efficaces
 - I/O efficace, parallélisable
 - "Lazy structure" en mémoire

- Librairie HDF5
 - Stockage par morceaux (chunk storage)
 - Algorithmes de compression efficaces
 - I/O efficace, parallélisable
 - "Lazy structure" en mémoire



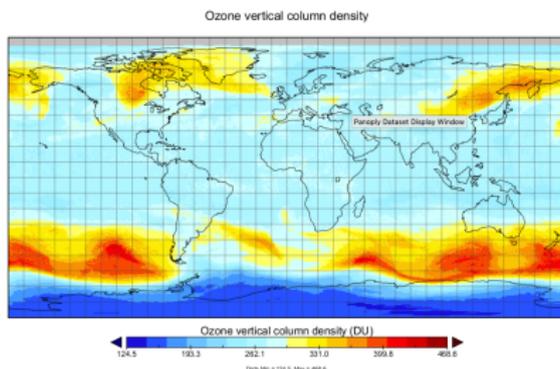
2 Revue Bibliographique

2.1 Format HDF5

2.2 Initiatives similaires : Tour d'horizon

Earth Observing System (EOS)

- Données satellitaires, géosciences/climatologie
- Système de fichiers standardisés, format **hdf5**
- Format de grille adapté à la surface de la Terre
- Données et outils hébergées sur des serveurs de la NASA
- Partage des données à distance via le protocole OPeNDaP (similaire HTTP)



Material Genome Initiative (USA)

- Objectif : établir des protocoles d'échange et réutilisation des données en mécanique des matériaux
- Exemple : Base ab initio *The Material Project*

Material Genome Initiative (USA)

- Objectif : établir des protocoles d'échange et réutilisation des données en mécanique des matériaux
- Exemple : Base ab initio *The Material Project*

AFLOW

- Base données DFT
- Input sim.° standard
- Automatisation sim.°
- RESTful API (~ WEB)

Material Genome Initiative (USA)

- Objectif : établir des protocoles d'échange et réutilisation des données en mécanique des matériaux
- Exemple : Base ab initio *The Material Project*

AFLOW

- Base données DFT
- Input sim.° standard
- Automatisation sim.°
- RESTful API (~ WEB)

NIST

- Dépôt de données mat. libre
- Pas de format imposé
- métadonnées auto-descr.
- RESTful API (~ WEB)

- Format données d'imagerie neutrons, rayons- X, muons
- Une mesure + Métadonnées essai = 1 fichier **hdf5**

- Format données d'imagerie neutrons, rayons- X, muons
- Une mesure + Métadonnées essai = 1 fichier **hdf5**
- **Profil de données** spécifié → fichier description XML

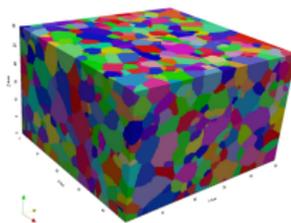
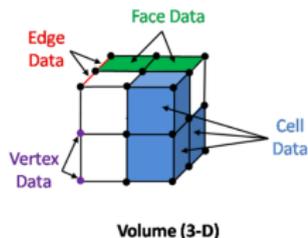
- Format données d'imagerie neutrons, rayons- X, muons
- Une mesure + Métadonnées essai = 1 fichier **hdf5**
- **Profil de données** spécifié → fichier description XML

```
1 <?xml version="1.0" ?>
2 <definition
3   xmlns="http://definition.nexusformat.org/nxd1/3.1"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5   xsi:schemaLocation="http://definition.nexusformat.org/nxd1/3.1 ../nxd1.xsd"
6   category="base"
7   name="verysimple"
8   version="1.0"
9   type="group" extends="NXobject">
10
11   <doc>
12     A very simple NeXus NXDL file
13   </doc>
14   <group type="NXentry">
15     <group type="NXdata">
16       <field name="counts" type="NX_INT" units="NX_UNITLESS">
17         <doc>counts recorded by detector</doc>
18       </field>
19       <field name="two_theta" type="NX_FLOAT" units="NX_ANGLE">
20         <doc>rotation angle of detector arm</doc>
21       </field>
22     </group>
23   </group>
24 </definition>
```

- Représentation digitale de microstructure polycristallines 3D
- 1 microstructure = 1 fichier **hdf5/xdmf**
 - Données stockées sur grilles, organisation matricielle
 - Hiérarchique : elements, features, ensembles

- Représentation digitale de microstructure polycristallines 3D
- 1 microstructure = 1 fichier **hdf5/xdmf**
 - Données stockées sur grilles, organisation matricielle
 - Hiérarchique : elements, features, ensembles
- Traitement de données : Pipeline
 - Filtres interopérables, API, ouverts à modules ext.
 - Reconstruction microscopie (3-beam EBSD 3D)
Caractérisation statistique, Maillage
 - Métadonnées traitement automatiquement incluses

- Représentation digitale de microstructure polycristallines 3D
- 1 microstructure = 1 fichier **hdf5/xdmf**
 - Données stockées sur grilles, organisation matricielle
 - Hiérarchique : elements, features, ensembles
- Traitement de données : Pipeline
 - Filtres interopérables, API, ouverts à modules ext.
 - Reconstruction microscopie (3-beam EBSD 3D)
Caractérisation statistique, Maillage
 - Métadonnées traitement automatiquement incluses



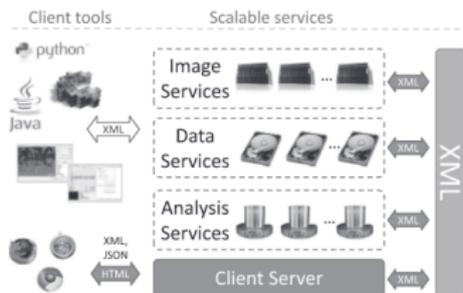
Grid representing a data matrix for the microstructure. The columns are labeled 1, 2, 3, 4, 5, 6, 7, 8, 9,, N-1, N. The rows are labeled with various properties:

	1	2	3	4	5	6	7	8	9	N-1	N
Size												
Shape												
# of Neighbors												
Avg. Orientation												
Surface Area												
Phase												
Centroid												
Avg. Neighbor Distance												

- [1] Read HSEBSD File
- [2] Threshold Objects
- [3] Convert Orientation Representation
- [4] Align Sections (Misorientation)
- [5] Neighbor Orientation Comparison (Bad Data)
- [6] Neighbor Orientation Correlation
- [7] Segment Features (Misorientation)
- [8] Find Feature Sizes
- [9] Find Feature Centroids
- [10] Find Feature Shapes
- [11] Minimum Size
- [12] Find Feature Neighbors
- [13] Minimum Number of Neighbors
- [14] Fill Bad Data
- [15] Generate IPF Colors
- [16] Find Feature Average Orientations
- [17] Create Element Array from Feature Array
- [18] Generate IPF Colors
- [19] Find Feature Reference Misorientations
- [20] Create Element Array from Feature Array
- [21] Extract Component as Attribute Array
- [22] Write DREAM.3D Data File

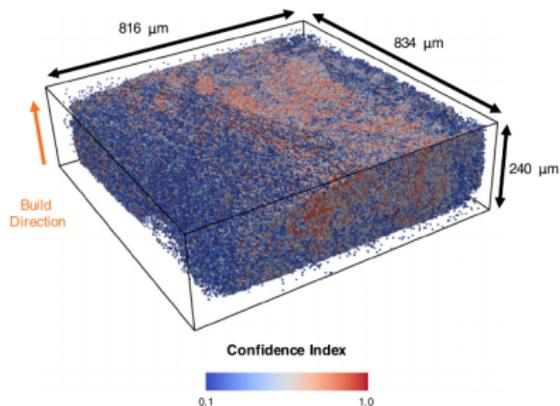
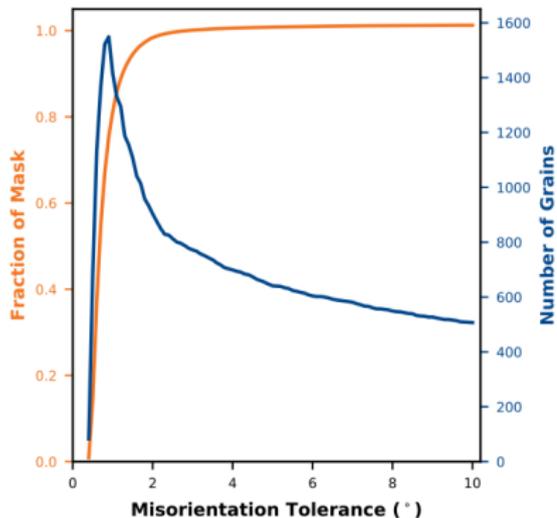
- Plateforme analyse et la gestion des Bio-images
- Annotation et recherche puissantes (tags, Name/Value pairs, metadonnées)

- Plateforme analyse et la gestion des Bio-images
- Annotation et recherche puissantes (tags, Name/Value pairs, métadonnées)
- Implémentation Web :
 - Fonctions = serveurs (requêtes HTTP / XML)
 - Possibilité d'intégrer modules ext. (plug-ins)
 - Scaling : ajout de serveurs supplémentaires



- Plug-in *DREAM3D* dans *BISQUE* (2018)
- Outils de Visualisation *HDF5* & images 3D
- Enregistrement systématique des métadonnées de traitement/provenance
- Pipelines *DREAM3D* parallélisées et paramétrés en cloud HPC
→ études paramétriques automatisées depuis navigateur web

Optimisation des paramètres de reconstruction de la microstructure d'un volume EBSD 3D (tolérance sur la désorientation)



- 1 Traitement et gestion de données massives et multimodales
- 2 Revue Bibliographique
- 3 Orientations proposées et Démonstrateurs**
 - 3.1 Solutions techniques proposées
 - 3.2 Profil de données
 - 3.3 Démonstrateurs
- 4 Conclusion et Perspectives

- 3 Orientations proposées et Démonstrateurs
 - 3.1 Solutions techniques proposées
 - 3.2 Profil de données
 - 3.3 Démonstrateurs

- Code source Python
 - Répandu, nombreux packages, utilisé au CdM
 - Compatible Pymicro
 - packages Machine Learning

- Code source Python
 - Répandu, nombreux packages, utilisé au CdM
 - Compatible Pymicro
 - packages Machine Learning
- Système de fichier : **HDF5**

- Code source Python
 - Répandu, nombreux packages, utilisé au Cdm
 - Compatible Pymicro
 - packages Machine Learning
- Système de fichier : **HDF5**
- API librairie HDF5 : **PyTables**
 - API Python/HDF5 haut niveau
 - Recherche/Indexation avancées
 - Compression des données facilitée
 - *ViTables* : visualisation arborescence HDF5
 - en supplément, *h5py* : package classique

- Extension **XDMF** systématique
 - Fichier XML associé au HDF5
 - Arborescence XML \Leftrightarrow Hiérarchie HDF5
 - Précision des types, dimension, adresse et nature des données dans le HDF5

- Extension **XDMF** systématique
 - Fichier XML associé au HDF5
 - Arborescence XML \Leftrightarrow Hiérarchie HDF5
 - Précision des types, dimension, adresse et nature des données dans le HDF5

- Visualisation
 - Paraview : XDMF/H5 \longrightarrow visualisation directe

- Extension **XDMF** systématique
 - Fichier XML associé au HDF5
 - Arborescence XML \Leftrightarrow Hiérarchie HDF5
 - Précision des types, dimension, adresse et nature des données dans le HDF5
- Visualisation
 - Paraview : XDMF/H5 \rightarrow visualisation directe
- Stockage des champs
 - Image 3D : trivial
 - Maillage : Valeurs nodales associée à un maillage stocké.
 - Simple à formater en XDMF \rightarrow Paraview
 - Simple à manipuler (projections de champs)

- Compression de données
 - Effectuée à l'aide de PyTables
 - Algorithme Blosc (performances I/O accrues)
 - Blosc semble rendre les données illisibles par Paraview
 - Etude performances I/O à mener pour choix optimal

- Compression de données
 - Effectuée à l'aide de PyTables
 - Algorithme Blosc (performances I/O accrues)
 - Blosc semble rendre les données illisibles par Paraview
 - Etude performances I/O à mener pour choix optimal
- Interactions avec outils/fichiers ext.
 - Module de code I/O *.geof*
 - package *vtk* Python
 - Fichiers *.mat* = HDF5 (*h5py*)
 - Intégration avec Pymicro : *.geof*, *.vtk*, *Zset*
 - AMITEX_ FFTP (FFT) : *vtk*. Evolution du code vers HDF5/XDMF possible

- 3 Orientations proposées et Démonstrateurs
 - 3.1 Solutions techniques proposées
 - 3.2 Profil de données
 - 3.3 Démonstrateurs

Definition

Spécifie la manière dont sont représentées les données (hiérarchie, types, contenu) représentant les objets à manipuler/étudier.

Definition

Spécifie la manière dont sont représentées les données (hiérarchie, types, contenu) représentant les objets à manipuler/étudier.

- Objets d'étude BIGMECA :
Lien Microstructures ↔ Comportement Mécanique

Definition

Spécifie la manière dont sont représentées les données (hiérarchie, types, contenu) représentant les objets à manipuler/étudier.

- Objets d'étude BIGMECA :
Lien Microstructures ↔ Comportement Mécanique
- Objectif : Convergence des données expérimentales et simulées

Definition

Spécifie la manière dont sont représentées les données (hiérarchie, types, contenu) représentant les objets à manipuler/étudier.

- Objets d'étude BIGMECA :
Lien Microstructures ↔ Comportement Mécanique
- Objectif : Convergence des données expérimentales et simulées
- Différents niveaux de cohérence physique des données :
 - ① Microstructure
 - ② État mécanique (référence, déformé(s))
 - ③ Dimension (mesure macro, champ, statistiques...)
 - ④ Type (mesures, simulations)

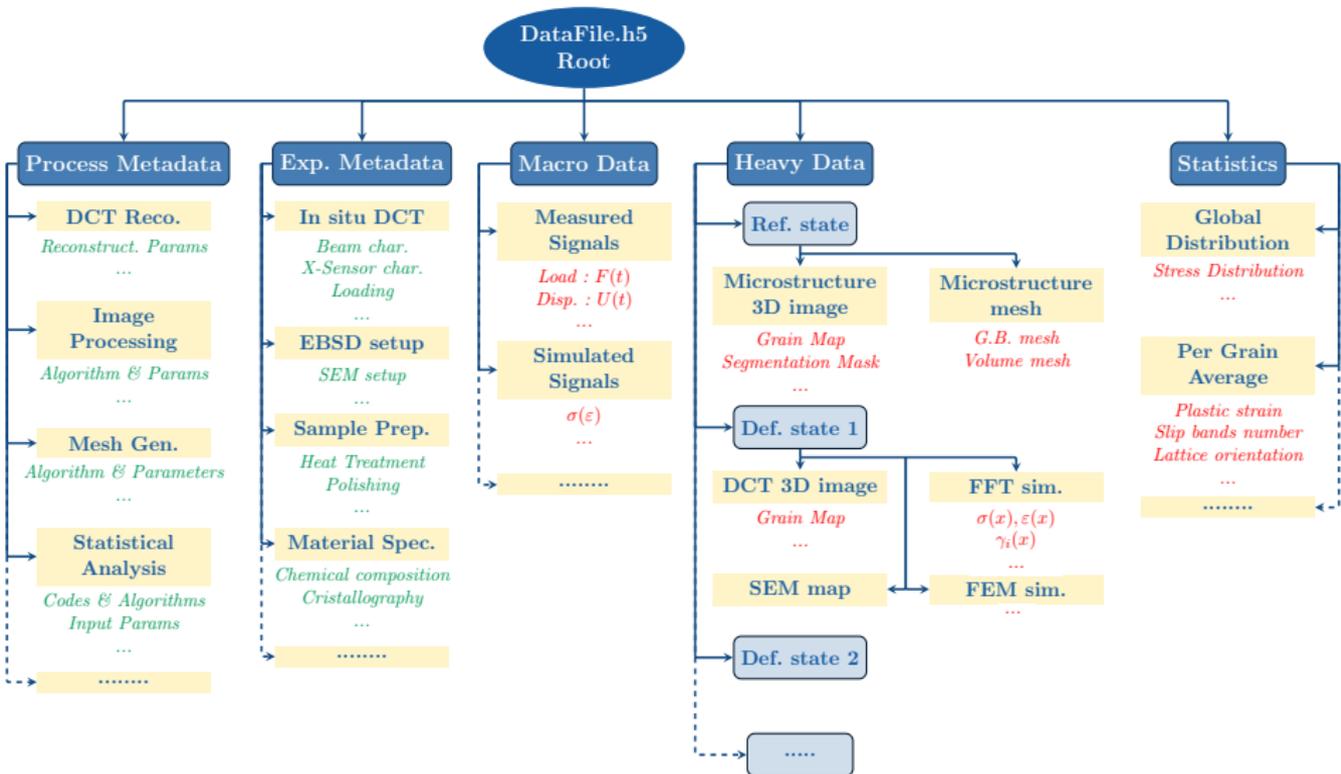
- **1 fichier = 1 microstructure/ 1 échantillon**
 - Cohérent avec la pratique (tomographie)
 - Cohérent avec initiatives similaires (DREAM3D, NEXUS)

- **1 fichier = 1 microstructure/ 1 échantillon**
 - Cohérent avec la pratique (tomographie)
 - Cohérent avec initiatives similaires (DREAM3D, NEXUS)
- Convergence des supports de données 3D, autant que possible
 - Origine et échelle commune des repères
 - Stockage sur des grilles communes
 - Stockage Multi-résolution
 - **Techniques de transfert de champs nécessaires**

- **1 fichier = 1 microstructure/ 1 échantillon**
 - Cohérent avec la pratique (tomographie)
 - Cohérent avec initiatives similaires (DREAM3D, NEXUS)
- Convergence des supports de données 3D, autant que possible
 - Origine et échelle commune des repères
 - Stockage sur des grilles communes
 - Stockage Multi-résolution
 - **Techniques de transfert de champs nécessaires**
- Conservation des données brutes intactes
 - Fichier séparé

Modèle de données

Exemple



- Enjeu majeur :
Développement d'un standard de description de la mécanique des microstructures pour la plasticité cristalline

- **Enjeu majeur :**
Développement d'un standard de description de la mécanique des microstructures pour la plasticité cristalline
- Construction progressive
 - Au fur et à mesure de la complexification de nos travaux
 - Échanges avec la communauté scientifique

- **Enjeu majeur :**
Développement d'un standard de description de la mécanique des microstructures pour la plasticité cristalline
- Construction progressive
 - Au fur et à mesure de la complexification de nos travaux
 - Échanges avec la communauté scientifique
- Bases de données de modèles
 - Keras (Machine Learning) : permet l'exportation HDF5
 - ROM : Bases réduites peuvent être stockées comme des champs classiques, selon le même format

- 3 Orientations proposées et Démonstrateurs**
 - 3.1 Solutions techniques proposées
 - 3.2 Profil de données
 - 3.3 Démonstrateurs

Structure de donnée mise en place

Implémentation

- Classe Python "SampleData" ↔ données d'un échantillon
 - Fichier .h5 et .xdmf associé
 - Implémente le profil de données
- Compatible avec
 - .h5 / .mat / .vtk / .geof
 - Maillages & Images 3D

Structure de donnée mise en place

Implémentation

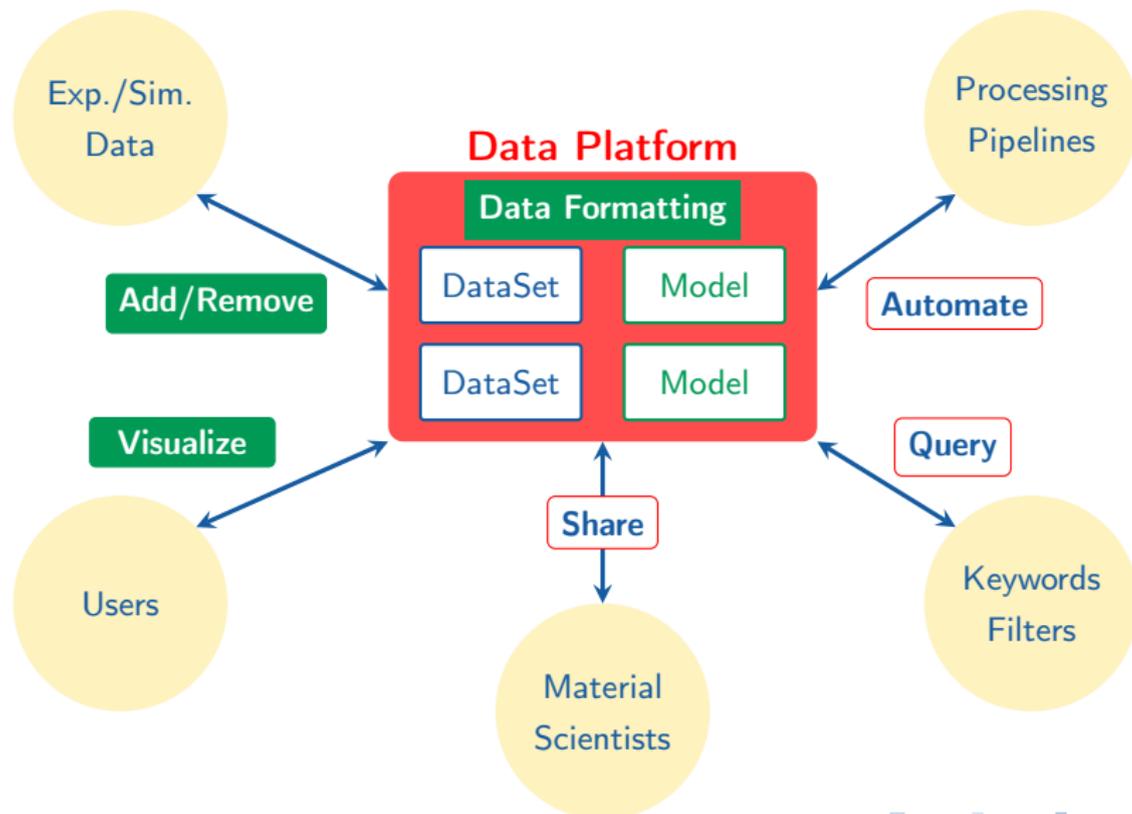
- Classe Python "SampleData" ↔ données d'un échantillon
 - Fichier .h5 et .xdmf associé
 - Implémente le profil de données
- Compatible avec
 - .h5 / .mat / .vtk / .geof
 - Maillages & Images 3D
- **Addition / Soustraction / Modification de données**

- Classe Python "SampleData" ↔ données d'un échantillon
 - Fichier .h5 et .xmf associé
 - Implémente le profil de données
- Compatible avec
 - .h5 / .mat / .vtk / .geof
 - Maillages & Images 3D
- **Addition / Soustraction / Modification de données**
- **Formatage automatique des données**
 - Construction du fichier .h5 + compression
 - Construction fichier XDMF synchronisée

- Classe Python "SampleData" ↔ données d'un échantillon
 - Fichier .h5 et .xdmf associé
 - Implémente le profil de données
- Compatible avec
 - .h5 / .mat / .vtk / .geof
 - Maillages & Images 3D
- **Addition / Soustraction / Modification de données**
- **Formatage automatique des données**
 - Construction du fichier .h5 + compression
 - Construction fichier XDMF synchronisée
- **Visualisation :**
 - Fichier .h5 et .xdmf → Paraview
 - Basé sur *PyTables* → ViTables (arbre hdf5)

Structure de donnée mise en place

Implémentation



Exemple 1 : échantillon polycristallin

Présentation des données

- Petit volume extrait d'un scan DCT d'un alliage ALi

Exemple 1 : échantillon polycristallin

Présentation des données

- Petit volume extrait d'un scan DCT d'un alliage ALi
- Données brutes
 - *Grain map* (image 3D) issue de la reconstruction
- Données ajoutées
 - Maillage 2D / 3D – *Franck NGuyen* (Matlab)
 - Simulation FFT, élasticité anisotrope, texture isotrope, AMITEX_FFTP

Exemple 1 : échantillon polycristallin

Présentation des données

- Petit volume extrait d'un scan DCT d'un alliage ALi
- Données brutes
 - *Grain map* (image 3D) issue de la reconstruction
- Données ajoutées
 - Maillage 2D / 3D – *Franck NGuyen* (Matlab)
 - Simulation FFT, élasticité anisotrope, texture isotrope, AMITEX_FFTP
- Construction du fichier de données multimodales
 - Recalage des maillages/images (origine/échelle commune)
 - Création des fichiers .h5 / .xdmf synchronisés
 - Stockage sur la même image 3D des champs simulés, et de la *Grain map*

Exemple 1 : échantillon polycristallin

Code de construction

Création de la structure de données

```
# Create Sample Data instance to store current sample multimodal data
File Name = 'Big_Meca_demo'
Sample Name = 'Big_Meca_demo'
complib = 'zlib'
bitshuffle = False
Sample = samples.SampleData(filename= File Name,
                             sample_name= Sample Name,
                             complib=complib,
                             bitshuffle=bitshuffle)
```

Ajout de la *Grain Map*

```
# Add Grain map 3D image from .mat file
File = 'Mesh_test_input1_trans.mat'
Name = 'Grain map'
Father_node = '/'
Get_Data = ['/im3D']
Description = ''' Reconstructed grain map from DCT imaging'''
Sample.add_image_from_file(imagefile= File,
                            image_name= Name,
                            location='/',
                            Mat_fields = Get_Data)
```

Ajout Maillage surfacique

```
# Add surface mesh from .mat file
Name = 'Surface_mesh'
Description = ''' Grain Boundary Mesh '''
Variable_mat_Nodes = '/FV_surf/vertices'
Variable_mat_Elements = '/FV_surf/faces'
Type_Element = ['Triangle',1]
Sample.add_mesh_from_file(meshfile= File,
                          meshname= Name,
                          location='/',
                          description= Description,
                          Mat_nodes = Variable_mat_Nodes,
                          Mat_elements = Variable_mat_Elements,
                          Mat_element_type = Type_Element,
                          Mat_transpose = True)
```

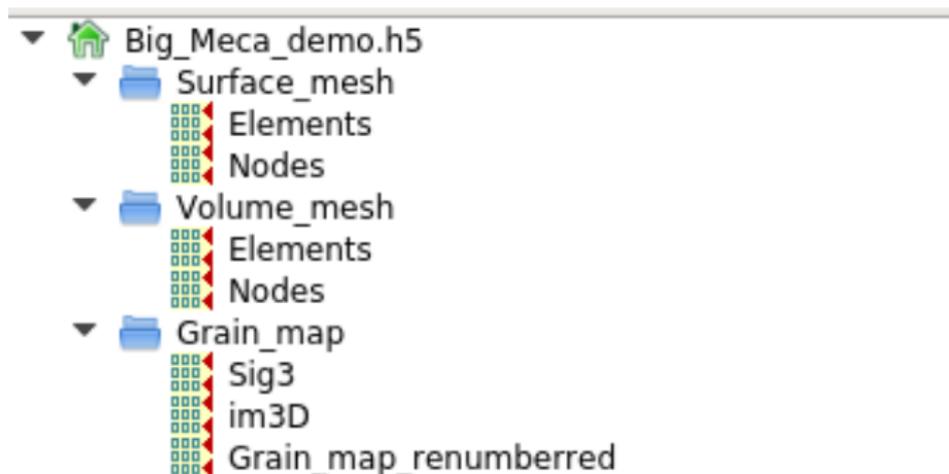
Renumérotation *Grain Map*

```
# Grain map renumbered construction and addition to SampleData
where = '/' + Name + Get_Data[0]
Image = Sample.h5_dataset.get_node(where)
Grains_list = np.unique(Image[:,:,:])
Image_renumbered = np.zeros(Image.shape)

# renumber grains
for k in np.arange(len(Grains_list)):
    Image_renumbered[np.where(Image[:,:,:] == Grains_list[k])] = np.int64(k)
# add renumbered grain map as supplementary field to the microstructure 3D image
Sample.add_field_to_image(image_location='/',
                          image_name='Grain_map',
                          field=Image_renumbered,
                          fieldname='Grain_map_renumbered')
```

Exemple 1 : échantillon polycristallin

Fichier HDF5 (ViTables)



Exemple 1 : échantillon polycristallin

Fichier XDMF

```
<?xml version='1.0' ?>
<!DOCTYPE Xdmf SYSTEM "Xdmf.dtd"[]>
<Xdmf xmlns:xi="http://www.w3.org/2003/XInclude" Version="2.2">
  <Domain>
    <Grid Name="Grain_map" GridType="Uniform">
      <Topology TopologyType="3DCoRectMesh" Dimensions="168 171 189"/>
      <Geometry Type="ORIGIN DXDYDZ">
        <DataItem Format="XML" Dimensions="3">0. 0. 0.</DataItem>
        <DataItem Format="XML" Dimensions="3">1. 1. 1.</DataItem>
      </Geometry>
      <Attribute Name="im3D" AttributeType="Matrix" Center="Cell">
        <DataItem Format="HDF" Dimensions="167 170 188" NumberType="Int" Precision="16">Big_Meca_demo.h5:/Grain_map/im3D</DataItem>
      </Attribute>
      <Attribute Name="Grain_map_renumbered" AttributeType="Matrix" Center="Cell">
        <DataItem Format="HDF" Dimensions="167 170 188" NumberType="Float" Precision="64">Big_Meca_demo.h5:/Grain_map/Grain_map_renumbered</DataItem>
      </Attribute>
      <Attribute Name="Sig3" AttributeType="Matrix" Center="Cell">
        <DataItem Format="HDF" Dimensions="167 170 188" NumberType="Float" Precision="64">Big_Meca_demo.h5:/Grain_map/Sig3</DataItem>
      </Attribute>
    </Grid>
    <Grid Name="Surface_mesh" GridType="Uniform">
      <Topology TopologyType="Triangle" NumberOfElements="903068">
        <DataItem Format="HDF" Dimensions="903068 3" NumberType="Int" Precision="64">Big_Meca_demo.h5:/Surface_mesh/Elements</DataItem>
      </Topology>
      <Geometry Type="XYZ">
        <DataItem Format="HDF" Dimensions="1764657 3" NumberType="Float" Precision="64">Big_Meca_demo.h5:/Surface_mesh/Nodes</DataItem>
      </Geometry>
    </Grid>
    <Grid Name="Volume_mesh" GridType="Uniform">
      <Topology TopologyType="Tetrahedron" NumberOfElements="829294">
        <DataItem Format="HDF" Dimensions="829294 4" NumberType="Int" Precision="64">Big_Meca_demo.h5:/Volume_mesh/Elements</DataItem>
      </Topology>
      <Geometry Type="XYZ">
        <DataItem Format="HDF" Dimensions="145669 3" NumberType="Float" Precision="64">Big_Meca_demo.h5:/Volume_mesh/Nodes</DataItem>
      </Geometry>
    </Grid>
  </Domain>
</Xdmf>
```

Exemple 1 : échantillon polycristallin

Fichier HDF5 (ViTables)

Maillage Volumique **Groupe**

General System attributes **User Attributes**

Database

Name: Volume_mesh
Path: /Volume_mesh
Type: group

Group

Number of children: 2

Child name	Type
Elements	leaf
Nodes	leaf

Maillage Volumique **Attributs**

General System attributes **User Attributes**

User attributes: 6

Name	Value	Datatype
Field_dim	{}	python
element_topology	['Tetrahedron', 1]	python
mesh_description		string
xdmf_geometry_path	Domain/Grid[3]Geometry	string
xdmf_path	Domain/Grid[3]	string
xdmf_topology_path	Domain/Grid[3]Topology	string

Champ Contraintes **Dataset**

General System attributes **User Attributes**

Database

Name: Sig3
Path: /Grain_mapSig3
Type: carray

Dataspace

Dimensions: 3
Shape: (167, 170, 188)
Data Type: float32
Compression: zlib

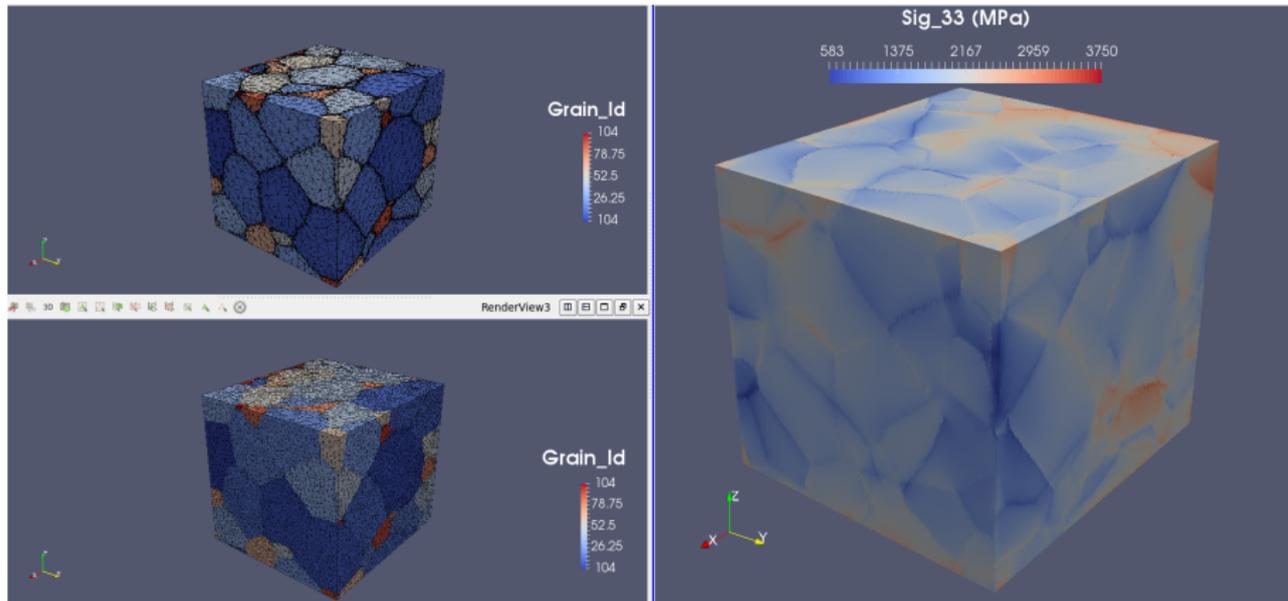
Champ Contraintes **Attributs**

User attributes: 1

Name	Value	Datatype
xdmf_path	Domain/Grid[Attribute3]	string

Exemple 1 : échantillon polycristallin

Visualisation



Exemple 2 : éprouvette creuse

Présentation des données

- Éprouvette creuses présentant un défaut géométrique de fabrication (*thèse A. Aublet*).

Exemple 2 : éprouvette creuse

Présentation des données

- Éprouvette creuses présentant un défaut géométrique de fabrication (*thèse A. Aublet*).
- Données
 - Segmentation : image binaire issue de la reconstruction XCT
 - Maillages surfaciques et volumiques de la segmentation
 - Maillage issu de la CAO de la pièce idéale (sans défauts)
 - Champ scalaire quelconque défini sur le maillage

Exemple 2 : éprouvette creuse

Présentation des données

- Éprouvette creuses présentant un défaut géométrique de fabrication (*thèse A. Aublet*).
- Données
 - Segmentation : image binaire issue de la reconstruction XCT
 - Maillages surfaciques et volumiques de la segmentation
 - Maillage issu de la CAO de la pièce idéale (sans défauts)
 - Champ scalaire quelconque défini sur le maillage
- Enjeu : Permettre la visualisation comparée simplement

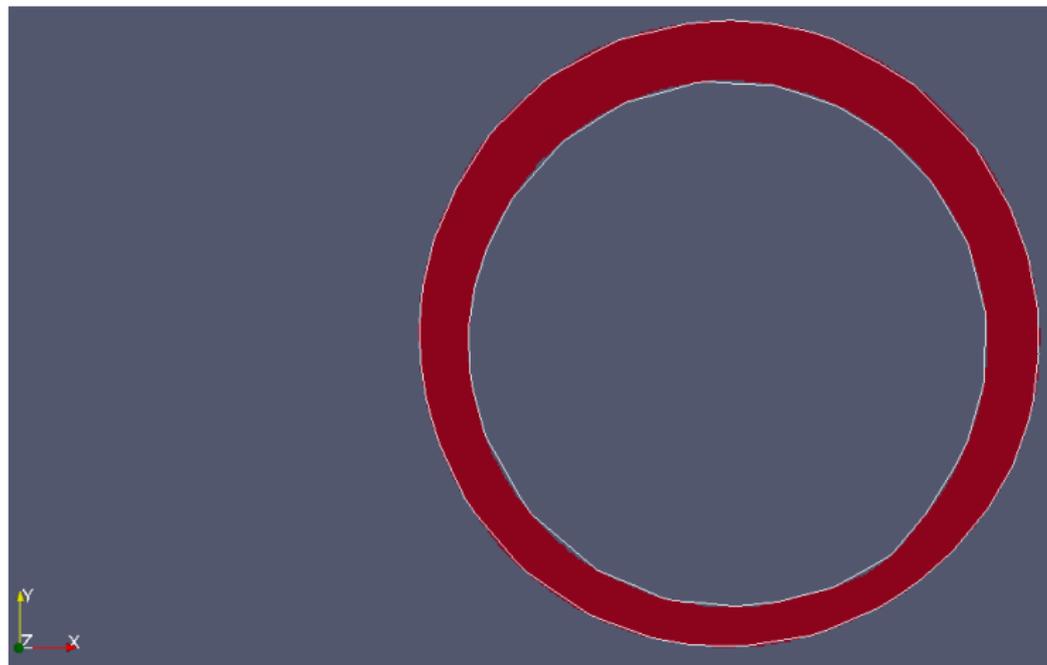
Exemple 2 : éprouvette creuse

Coupe longitudinale, maillage volumique



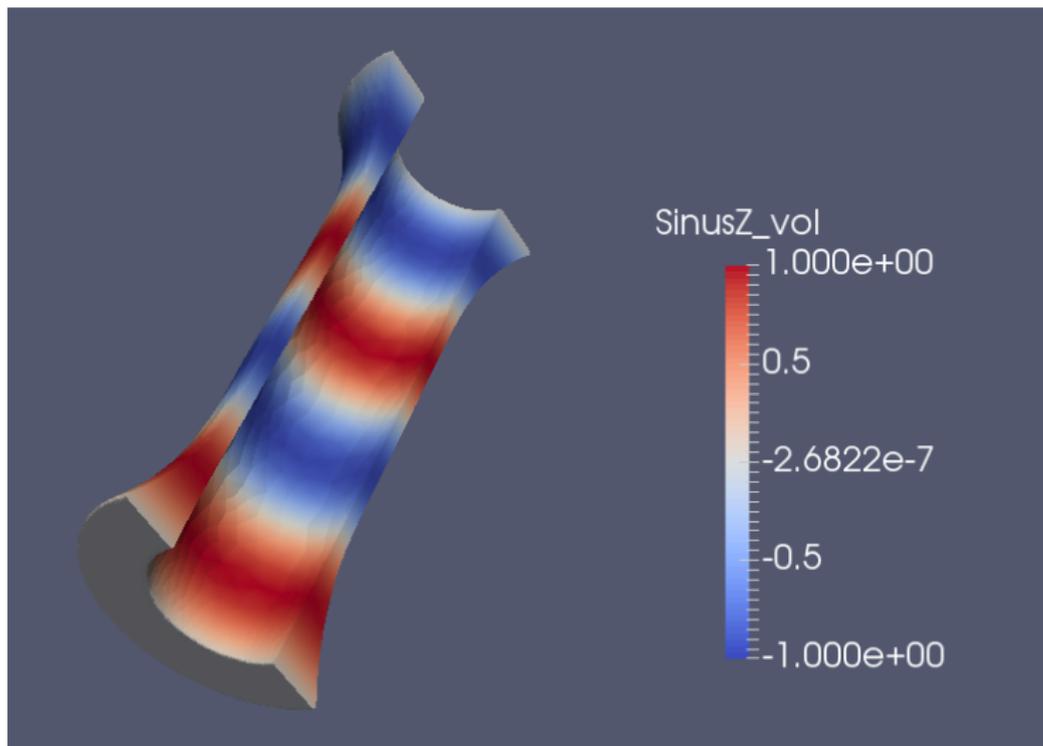
Exemple 2 : éprouvette creuse

Coupe transversale, maillage surfacique



Exemple 2 : éprouvette creuse

Champ scalaire



Exemple 2 : éprouvette creuse

Comparaison de maillages

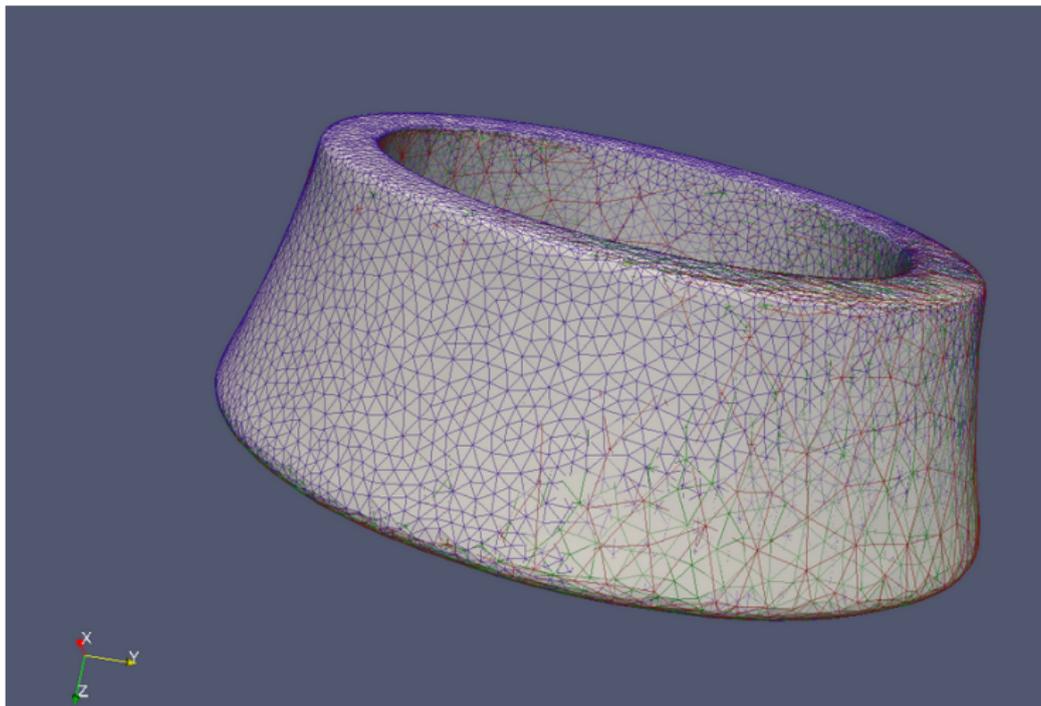


Table of Contents

- 1 Traitement et gestion de données massives et multimodales
- 2 Revue Bibliographique
- 3 Orientations proposées et Démonstrateurs
- 4 Conclusion et Perspectives

Conception de la plate-forme de données

- Analyse des différents aspects du problème
- Revue biblio : pistes identifiées pour chacun de ces aspects
- Rapport technique détaillé

Conception de la plate-forme de données

- Analyse des différents aspects du problème
- Revue biblio : pistes identifiées pour chacun de ces aspects
- Rapport technique détaillé

Solutions techniques proposées

- Fonctions d'ajout/suppression, formattage, et visualisation des données
- Implémentation en Python simple à utiliser et validée
- Apte à gérer tous les types de données prévues dans BIGMECA
- Modèles de données : organisation générale proposée, à préciser.

- Package Python (H. Proudhon) : gestion essais DCT

- Package Python (H. Proudhon) : gestion essais DCT

Intérêts techniques

- Convergence modèle de données DCT Pymicro avec SampleData (extension multimodale)
- Forward simulation DCT avec champs simulés
- Packages I/O *vtk*, *.geof*, *Zset*, du package de cristallographie de Pymicro

- Package Python (H. Proudhon) : gestion essais DCT

Intérêts techniques

- Convergence modèle de données DCT Pymicro avec SampleData (extension multimodale)
- Forward simulation DCT avec champs simulés
- Packages I/O *vtk*, *.geof*, *Zset*, du package de cristallographie de Pymicro

Intérêts pratiques

- Code distribué sur PyPI / GitHub
- Documentation

Objectifs

- Quantifier effet de l'incertitude sur les positions des joints de grains lors de la reconstruction DCT
- Développer une chaine de traitement automatisée s'appuyant sur la plate-forme de données

Objectifs

- Quantifier effet de l'incertitude sur les positions des joints de grains lors de la reconstruction DCT
- Développer une chaîne de traitement automatisée s'appuyant sur la plate-forme de données

Chaîne de traitement

- Paramétrage de la position des joints de grains : modes propres
- Entraînement \rightarrow ROM
- Modèle réduit + Zoom structural + Estimation incertitude paramètre de position du joint de grain (DCT)
 \Rightarrow Quantification de l'incertitude sur la réponse mécanique

- Automatisation des simulations FFT avec AMITEX_FFTP
- Test de performances compression sur un Dataset lourd (ex : Moyenne par grain d'un champ depuis dataset compressé)
- Intégration dans BISQUE ? Plate-forme WEB ?
- Serveur de stockage ?