Plate-forme de Données BIGMECA

A. Marano

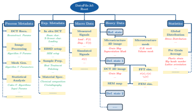
COPIL – Chaire BIGMECA

13 janvier 2021

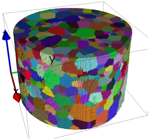
Reminder – Last COPIL

Report

Data Format



Integration with Pymicro

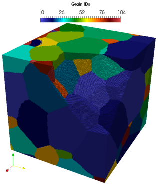


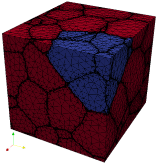
SampleData Class

Generic & Simple API

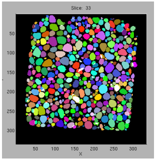
Meshes / Images / Arrays

Metadata – Data Compression





DCT microstructural uncertainty propagation



Contents

Integration with Pymicro Package

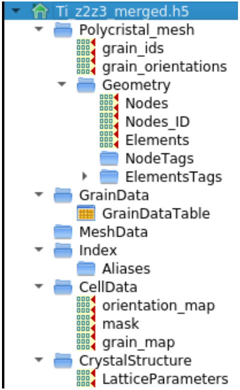
- **Package integration :**
  - integrate with X-Ray/FEM tools and file formats
  - Pymicro Microstructure Class inherited from SampleData
  - GIT repository
  - Updated documentation (not yet online)
  - Non regression tests updated
- **Code improvements :**
  - Compression fine tuning of datasets
  - Generic Data Model mechanism $\longrightarrow$ subclasses for specific applications
  - Access to data with : Path, Node name, Index name, Aliases
  - Scalar, vector and tensor fields defined on 2D/3D Images/Meshes
  - Introduction of structured tables arrays (multifields arrays)

Integration with Pymicro Package

```python
def minimal_data_model(self):
    """Data model for a polycrystalline microstructure.

    Specify the minimal contents of the hdf5 (Group names, paths and group
    types) in the form of a dictionary {content: location}. This extends
    `~pymicro.core.SampleData.minimal_data_model` method.

    :return: a tuple containing the two dictionnaries.
    """
    minimal_content_index_dic = {'Image_data': '/CellData',
                                 'grain_map': '/CellData/grain_map',
                                 'mask': '/CellData/mask',
                                 'Mesh_data': '/MeshData',
                                 'Grain_data': '/GrainData',
                                 'GrainDataTable': ('/GrainData/'
                                                    'GrainDataTable'),
                                 'Crystal_data': '/CrystalStructure',
                                 'lattice_params': ('/CrystalStructure'
                                                    '/LatticeParameters'), }
    minimal_content_type_dic = {'Image_data': '3DImage',
                                'grain_map': 'Array',
                                'mask': 'Array',
                                'Mesh_data': 'Mesh',
                                'Grain_data': 'Group',
                                'GrainDataTable': GrainData,
                                'Crystal_data': 'Group',
                                'lattice_params': 'Array', }
    return minimal_content_index_dic, minimal_content_type_dic
```



Integration with Pymicro Package

```python
class GrainData(tables.IsDescription):
    """
      Description class specifying structured storage of grain data in
      Microstructure Class, in HDF5 node /GrainData/GrainDataTable
    """
    # grain identity number
    idnumber = tables.Int32Col()  # Signed 64-bit integer
    # grain volume
    volume = tables.Float32Col()  # float
    # grain center of mass coordinates
    center = tables.Float32Col(shape=(3,))  # float  (double-precision)
    # Rodriguez vector defining grain orientation
    orientation = tables.Float32Col(shape=(3,))  # float  (double-precision)
    # Grain Bounding box
    bounding_box = tables.Int32Col(shape=(3, 2))  # Signed 64-bit integer
```

GrainDataTable

	bounding_box	center	idnumber	orientation	volume
0	[[72,74], [10,18], [3, 9]]	[0.24174044,...	12	[0.9882962 ,-0.09180787, 0.14053537]	39.
1	[[0, 1], [71,72],...	[0.0104857 ,0.24655765,...	28	[0.06005969, 0.9892864 ...	1.
2	[[41,71], [10,41],...	[0.15370159,...	38	[0.8752973 ,-0.42830908...	10272.
3	[[0,24], [37,69],...	[0.0228855 ,0.15506616,...	40	[0.49868646,...	4705.
4	[[0,11], [33,56],...	[-0.01661 , 0.12986557, ...	53	[-0.02484809, 0.8941689 , ...	1891.
5	[[68,74], [49,72],...	[0.22497821,...	61	[0.76558834, 0.6130021 ...	1093.

Integration with BasicTools

- **Goals :**
  - Integrate tool for mesh data I/O and generic mesh objects
  - Cooperation/Compatibility with Safran

- **BT Integration :**
  - BT Mesh classes $\longrightarrow$ SampleData Mesh Group
  - SampleData Mesh Group $\longrightarrow$ BT Mesh classes
  - Nodes, Elements, Sets, XDMF format for nodal fields/element fields

- **Problems to solve :**
  - Convention to store IP fields
  - Visualization of IP fields ? Create Paraview plug-in (analogous to BT plug-in) ?

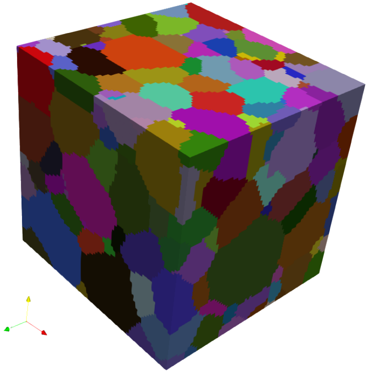
- **Alternatives :**
  - Integrate with *meshio* open source package

Use case : DCT-X experiment sample digital twin

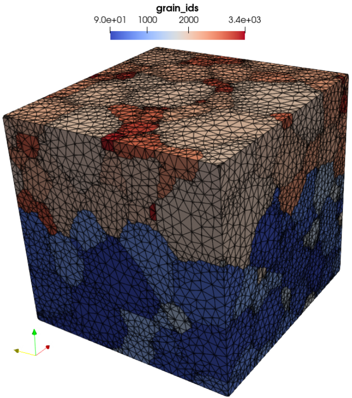
- PSICHE line from SOLEIL synchrotron facility, *Ti* grade 2 sample, undeformed state.
- Subvolume extracted from 2 merged scan reconstructions.
- Data platform used to construct a dataset gathering :
  - Raw data of reconstructed grain map
  - Dilated and cleaned grain map (morphology operators)
  - Grains mesh for FEM computations
  - Stress and Strain fields from 3 FFT simulations with 3 different boundary conditions (anisotropic elasticity).
  - Grains geometric and physical data in a structured table.

Digital twin reconstructed from DCT experiment

Grain map

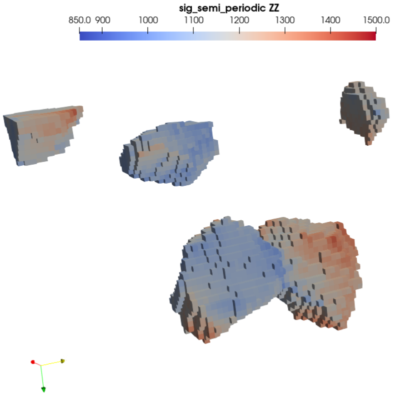


Grains mesh (Grain Id field)

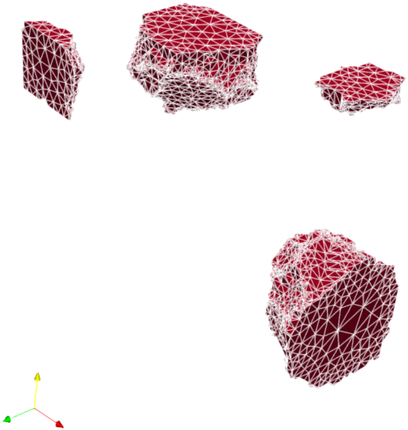


Grains visualization

Using data set content and Paraview Threeshold filter, visualization of fields on particular grains is straghtforward (+ clips, slices...).



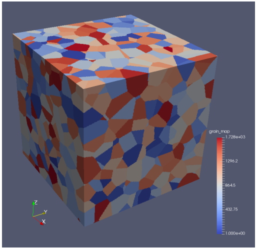
axial stress field (FFT simulation)

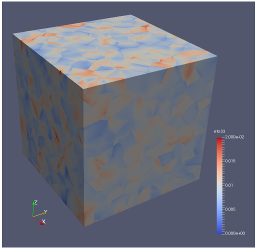


Mesh

Conclusion – Perspectives

- **Summary :**
  - Complete Integration with Pymicro
  - Functional integration with BasicTools
  - Operational and efficient V1 Data Platform
- **First half of 2021 $\longrightarrow$ Support for Numerical processing chains :**
  - **H. Proudhon** : anisotropic modeling of polycrystals
    Neper $\longrightarrow$ Amitex_fftp
  - **J. Bertoldo** : Tomographic images segmentations with CNN
  - **H. Launay** : ROM Database for welding defects via Machine Learning

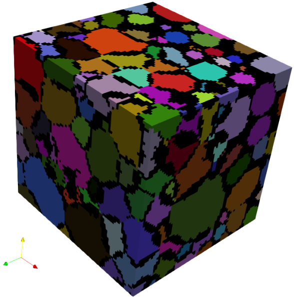




Contents

Problem Presentation

DCT reconstruction output



Problem

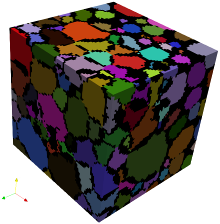
- Exact position of GB ?
- Standard : dilate grains and use output as digital twin
- Simulation results uncertainty associated to standard method ?

Challenges

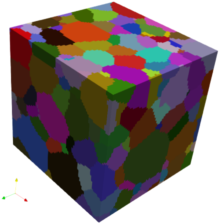
- Very large dimension
- How to chose simulations to perform ?
- How to propagate uncertainty ?

Data

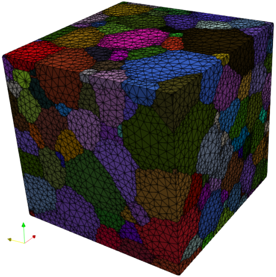
DCT reconstruction output



Grain Map after dilatation



Polycrystal mesh



Method

- **General expectations :**
  - Low sensitivity at the meso/macro scale
  - High potential sensitivity at micro/local scale (localization, stress concentration, damage initiation...)

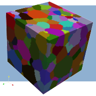
- **Reduce problem dimension :**
  - Construct a parametric representation of grain boundary position of reasonable dimension

- **Start simple :**
  - Anisotropic elasticity
  - Avoid modeling of random microstructure and Monte-Carlo simulations
  - Estimate sensitivity by simulating limit cases (to be defined) that remain in the uncertainty zone

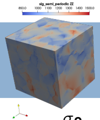
Numerical Chain Overview

Digital Twin 0

AMITEX



FFT simulation

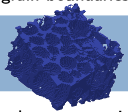


σ_0

"Dilated" grain map
Standard digital twin
Full Microstructure Simulation

GB parametric model

"Thick grain boundaries" mesh



Eigen modes computation $\{\varphi_i\}$

Reduced base of
Plausible microstructures

Data stored in SampleData file between each step

Goal : fully automated process

Sensitivity quantification

Generate N structural zooms

BC : σ_0 (Neumann) or u_0 (Dirichlet)

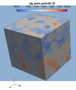
Control indicator $E = \sigma^j_{Neu} - \sigma^j_{Dir}$

Local deviation : $\max\left(\sigma^j - \sigma_0\right)$

Macro deviation : $||\sigma^j - \sigma_0||_{gr\,/\,VER}$

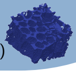
Local submodels selection

Deform mesh 0 with $u_i = \pm\alpha_i\varphi_i$



Field projection

$W_i^* = \mathcal{W}^*(\sigma_0, u_i)$



Look for N max of $\mathbf{W} = [W_i^*]$
$\rightarrow$ coordinates of N interest
zones and GB deformations

Required Technical developments

- **Automatic meshing of multi-phase image :**
  - Matlab script relying on F. Nguyen tools (currently deployed)
  - To be automated through a Pymicro's *Microstructure* class method
- **Field transfer :**
  - Handling/storage of IP fields
  - Enable and automate field transfer between FFT simulation results and Zset
- **Processing Chain automation :**
  - Automate I/O interface between SampleData and simulation softwares (Zset, Amitex_FFTP)
  - Define storage datamodel

Perspectives

- **Enrich geometric modeling :**
  - Random model of plausible microstructures
  - Eigen modes amplitudes $\longrightarrow$ random variables
  - Monte-Carlo simulation
- **Reduced order modeling and machine learning :**
  - Hyper-reduction of the parametric model simulation
  - Clustering of the parametric space with associated hyper-reduced models
  - Accelerate Monte-Carlo simulations
- **Physics :**
  - Crystal plasticity (localization, crystal rotation)
  - Damage
- **Computational ressources :**
  - 500.000 h obtained on the Jean Zay (GENCI/ IDRIS) Cluster
  - We trust that more can be obtained after first results