FFT-BASED SOLVERS TO EVALUATE STRESS DISTRIBUTIONS IN RPV STEELS

L.Gélébart
CEA Paris-Saclay DEN/DMN/SRMA

5th September 12h20-13h

This project received funding under the Euratom research and training programme 2014-2018 under grant agreement N° 661913

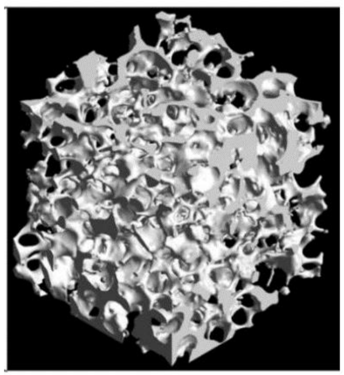
Outline

- GENERAL Context
  - FFT based solvers for heterogeneous materials
  - The AMITEX_FFTP code (specificities and use)
- SOTERIA Context
  - Stresses at Grain Boundaries
  - Application to RPV steels

General context

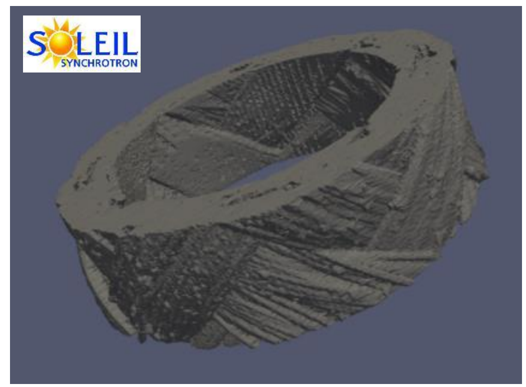
Heterogeneous materials

- Porous ceramics



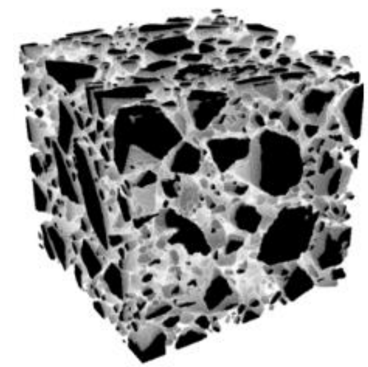
Ackermann &al. Materials 2014

- SiC/SiC composite tube



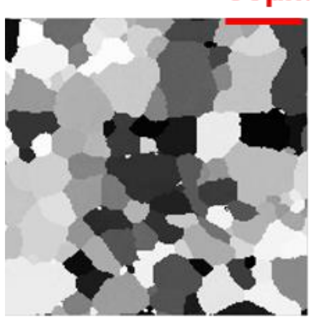
from CHEN Y. Thesis, CEA, ENPC, 2017

- Concrete

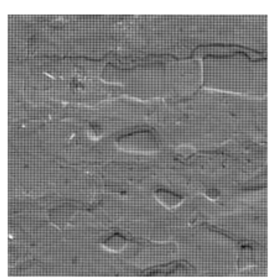


from F. Bernachy, CEA, 2017

- Polycrystals => SOTERIA application!

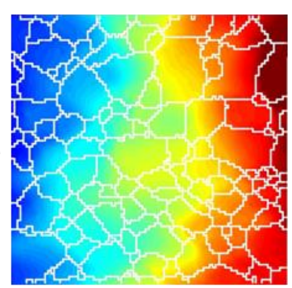


M. Dexet thesis, CEA, LMS-X, 2006



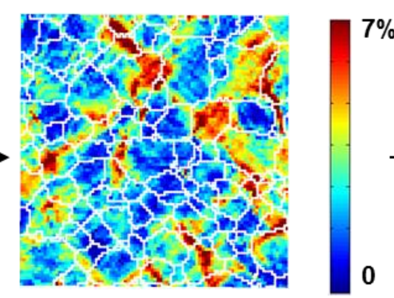
Microgrid

→



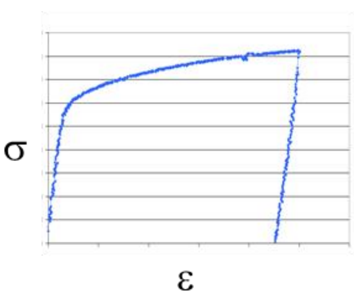
Experimental displacement field

→



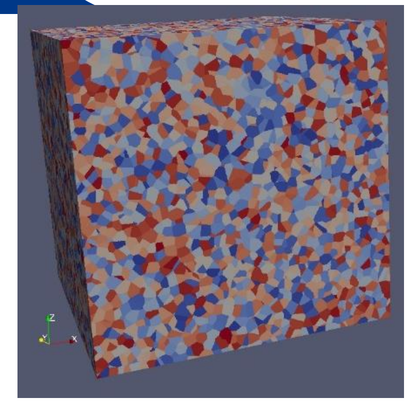
Experimental strain **field**

+ σ



Macroscopic response

FFT-based solvers for heterogeneous materials



Simulation of heterogeneous materials

- A **« Representative » Volume Element (RVE)**
- A **Constitutive behavior law** for each phase
- An **« Average loading »** : uniaxial stress (tensile test) for example
- A type of **Boundary Conditions**: **Periodic BC** is a good choice

Natural trends

- **Increase the spatial resolution** to obtain a better description of local fields
- **Increase the size of the RVEs** to obtain representative results
- **« Physically based »** constitutive behaviors are more and more complex

Standard FEM solvers

- **Numerical limits (memory size & computation time)**

« FFT-based » solvers

- **No tedious meshing procedure (input=digital image)**
- **Much more efficient than standard FEM solvers**
- **Easy to implement**
- **Well-suited for Parallelism => PUSH BACK THE LIMITS!**

FFT-based solvers for heterogeneous materials

FIX-POINT algorithm (Moulinec-Suquet 1994)

Problem to solve

$$\sigma(x) = c(x) : \varepsilon(x)$$
$$div(\sigma(x)) = 0$$
$$< \varepsilon(u(x)) >= E$$
$$+ \text{periodicity} + \text{compatibility}$$

$$\sigma(x) = c_0 : \varepsilon(x) + \underbrace{(c(x) - c_0) : \varepsilon(x))}_{\tau(x)}$$

Rewriting of the problem

$$\sigma(x) = c_0 : \varepsilon(x) + \tau(x)$$
$$\tau(x) = (c(x) - c_0) : \varepsilon(x)$$
$$div(\sigma(x)) = 0$$
$$< \varepsilon(u(x)) >= E$$
$$+ \text{periodicity} + \text{compatibility}$$

Auxiliary problem

$$\sigma(x) = c_0 : \varepsilon(x) + \tau(x)$$
$$div(\sigma(x)) = 0$$
$$< \varepsilon(u(x)) >= E$$
$$+ \text{periodicity} + \text{compatiblity}$$

Solution for the auxiliary problem

$$\varepsilon(x) = -\Gamma_0 * \tau(x) + E$$

Applying the Green operator
Simple in Fourier space (FFT)
Mura 1997

Moulinec-Suquet, 1994

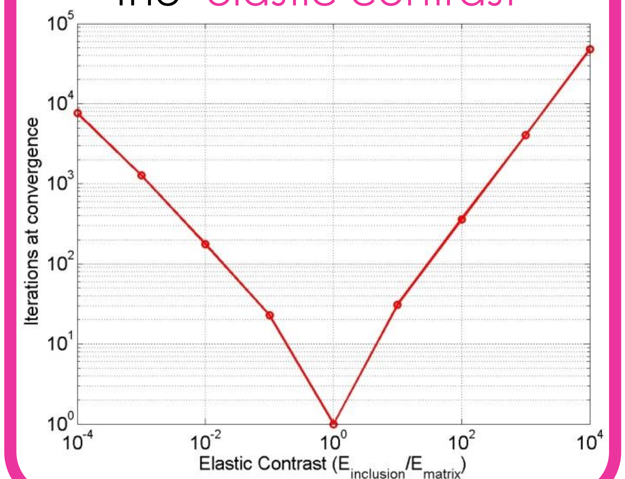
$$\tau(x) = (c(x) - c_0) : \varepsilon(x)$$

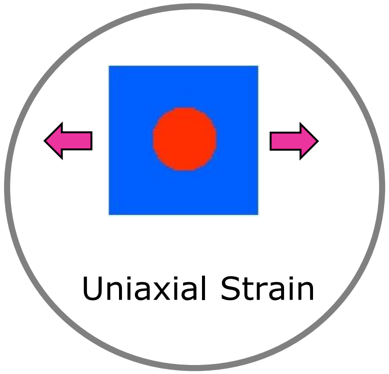
FFT-based solvers for heterogeneous materials

- Drawbacks of the method as proposed in 1994 by Moulinec & Suquet

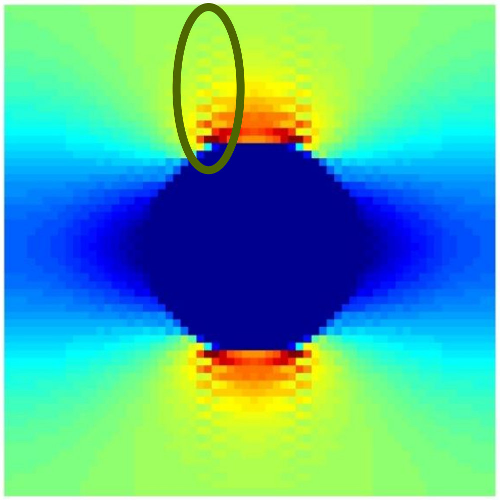
Sensitivity to

the elastic contrast



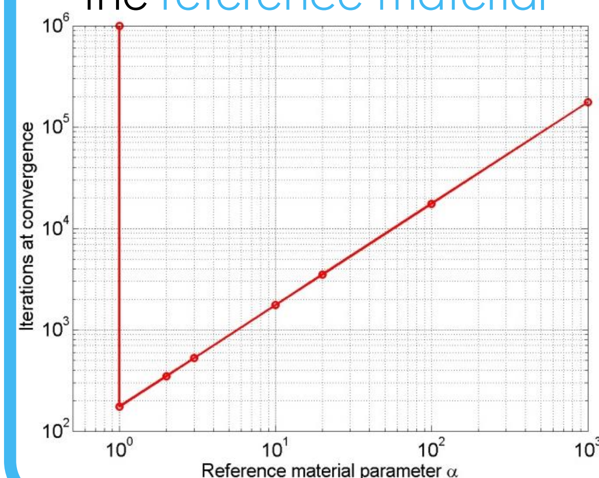


Spurious Oscillations



Sensitivity to

the reference material



FFT-based solvers for heterogeneous materials

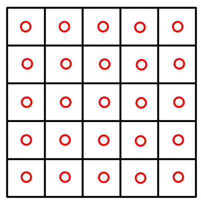
- A lot of work and improvements since 1994!

  - **Modified Discrete Green Operator** *(Willot 2015, Schneider 2016…)*
    - sensitivity to the elastic contrast
    - spurious oscillations
  - **Algorithms** *(Zeman 2010, Brisard 2010, Gélébart 2013…)*
    - sensitivity to the reference material
  - **Composite voxels** *(Kabel2015, Gélébart 2015…)*
    - spurious oscillations
      (very efficient for thin interphases)

FFT-based solvers for heterogeneous materials

❑ Discrete Green operators

- **CLASSICAL DGO : truncated Continuous Green Operator** (Moulinec-Suquet 1994)



$$\boldsymbol{div}(\sigma) = \mathbf{0}$$
$$\varepsilon = (grad(\boldsymbol{u}))^{sym}$$
$$\sigma = c_0 : \varepsilon + \tau$$

⇨

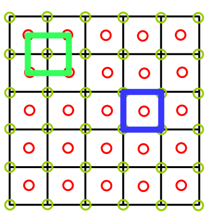
$$i\hat{\sigma}.\boldsymbol{k_a} = \mathbf{0}$$
$$\hat{\varepsilon} = i\widehat{\boldsymbol{u}^*} \otimes^{sym} \boldsymbol{k_a}$$
$$\hat{\sigma} = c_0 : \hat{\varepsilon} + \hat{\tau}$$

⇨

$$\hat{\varepsilon} = -\widehat{\Gamma}_0 : \hat{\tau}$$

u, ε, σ at voxels centers

- **MODIFIED DGO : DISCRETE DIFFERENTIAL OPERATORS = contour integrals**



$\boldsymbol{u}$ at voxels corners ⟹ $\varepsilon = (grad(\boldsymbol{u}))^{sym} \cong \frac{1}{v} \int_{\partial v} \boldsymbol{u} \otimes^{sym} \boldsymbol{n}\, ds$ at voxels centers

σ at voxels centers ⟹ $\boldsymbol{div}(\sigma) \cong \frac{1}{v} \int_{\partial v} \sigma . \boldsymbol{n}\, ds$ at voxels corners

… ⇨

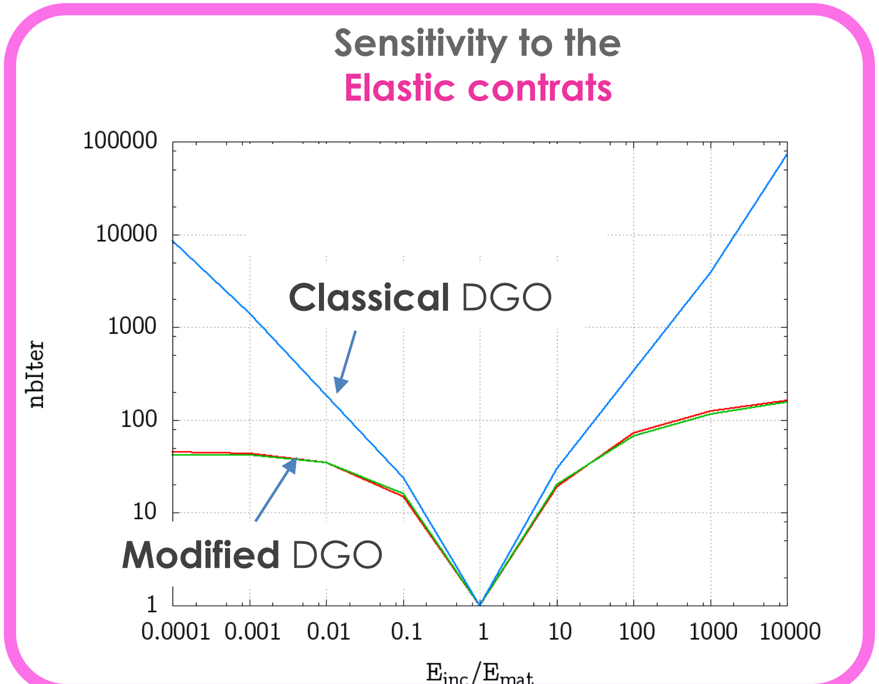
$$i\hat{\sigma}.\widetilde{\boldsymbol{k_a}} = \mathbf{0}$$
$$\hat{\varepsilon} = i\widehat{\boldsymbol{u}^*} \otimes^{sym} \widetilde{\boldsymbol{k_a}}$$
$$\hat{\sigma} = c_0 : \hat{\varepsilon} + \hat{\tau}$$

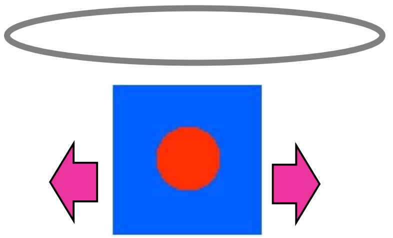
⇨

$$\hat{\varepsilon} = -\widehat{\widetilde{\Gamma}}_0 : \hat{\tau}$$

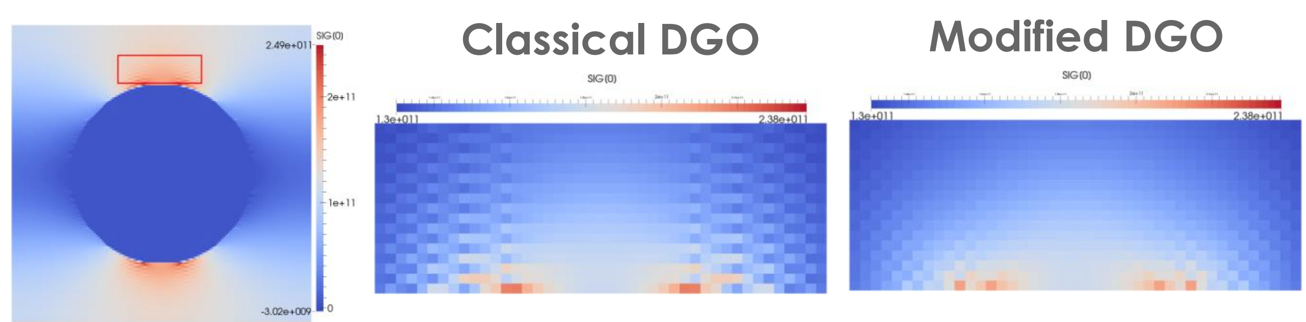
FFT-based solvers for heterogeneous materials

- Modified Discrete Green Operator



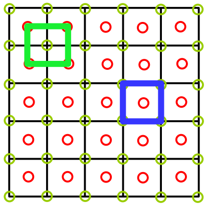


Uniaxial Strain



FFT-based solvers for heterogeneous materials

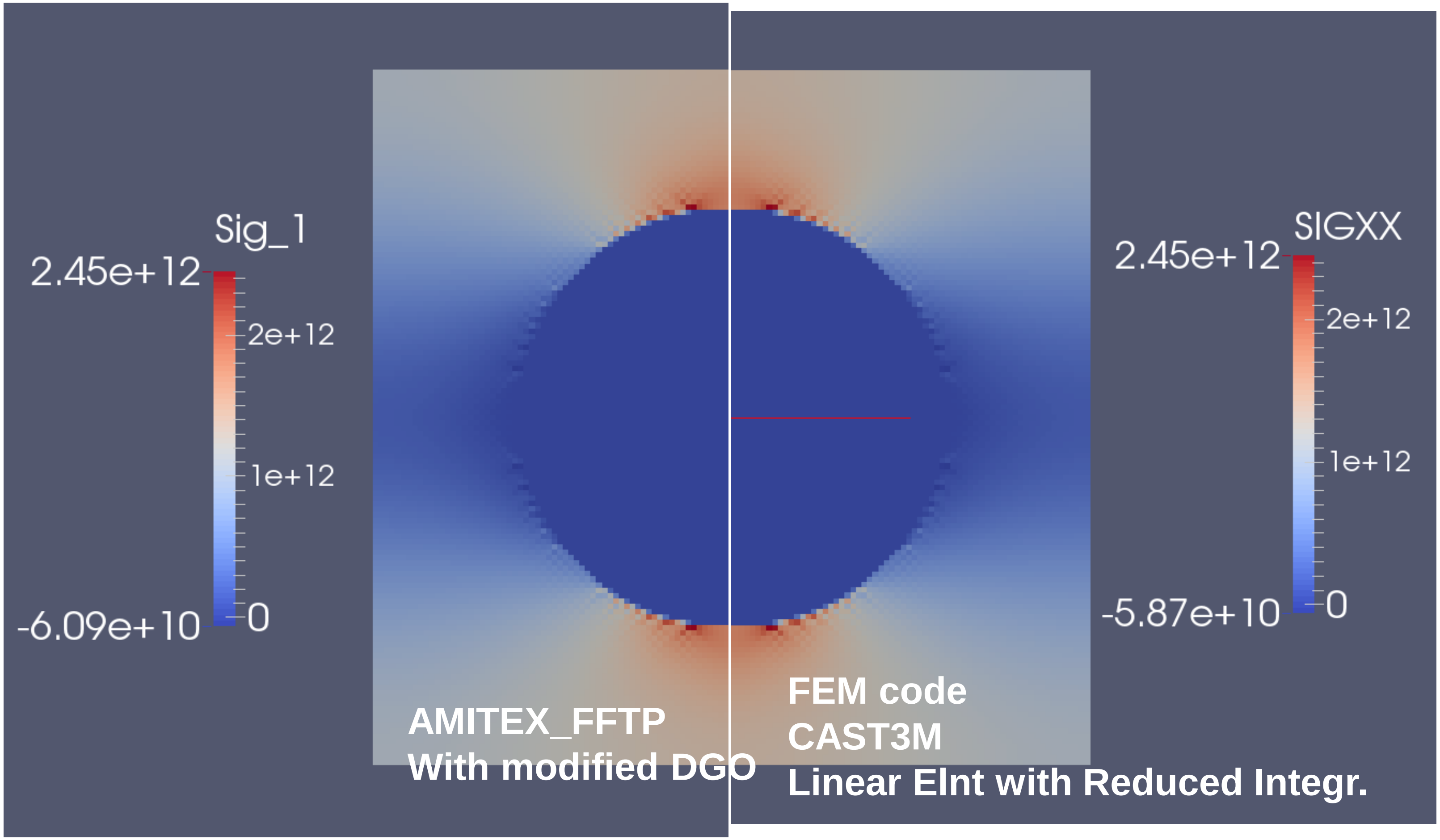
- Modified Discrete Green Operator



↔ **Linear FINITE ELEMENTS with Reduced integration** **(Schneider &al., IJNME 2016)**

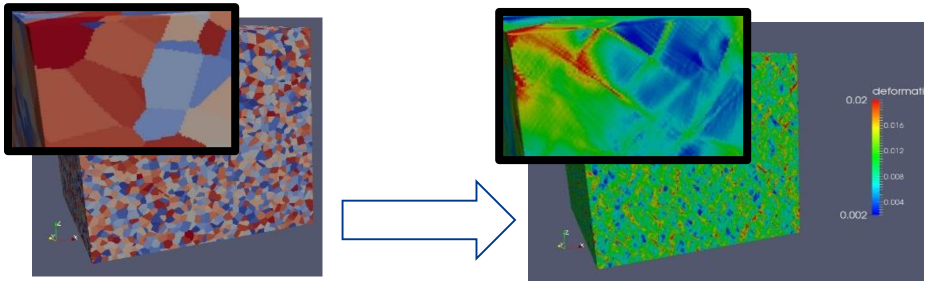
FFT-based solvers for heterogeneous materials

- Modified Discrete Green Operator



➡ AMITEX_FFTP : a **FE method** with an **FFT-based solver**

The AMITEX_FFTP code



AMITEX

http://www.maisondelasimulation.fr/projects/amitex/html/overview.html
old version on the website, to be refreshed
contact L. Gelebart for a recent version

❑ Highly parallel implementation (MPI)

- **Models**
  - Mechanics : **SMALL STRAINS** and **FINITE STRAINS**
  - Diffusion
- **Algorithm**
  - Fix Point + Convergence acceleration
- **Behavior**
  - User defined : **umat** compatibility => **coupling with mfront!**
  - « Composite » voxels
- **Various loading types**

The AMITEX_FFTP code

SOTERIA

- Highly parallel implementation (distributed memory with MPI)

$$\varepsilon^0(x) = E$$

$$\tau^k(x) = \sigma(\varepsilon^k(x)) - c_0 : \varepsilon^k(x)$$

$$\tau^k(x) \rightarrow \hat{\tau}^k(\xi)$$

$$\hat{\varepsilon}^{k+1}(\xi) = -\hat{\Gamma}_0(\xi) : \hat{\tau}^k(\xi) \qquad \hat{\varepsilon}^{k+1}(0) = E$$

$$\hat{\varepsilon}^{k+1}(\xi) \rightarrow \varepsilon^{k+1}(x)$$

Distributed memory // implementation **(MPI)**

- Behavior : « local » in real space
- Green Operator : « local » in Fourier space
- FFT & iFFT : « non-local » (needs data transfer)

The AMITEX_FFTP code

- Highly parallel implementation (MPI)

  - Decomposition

    - ✓ Decomposition 1D (slabs)

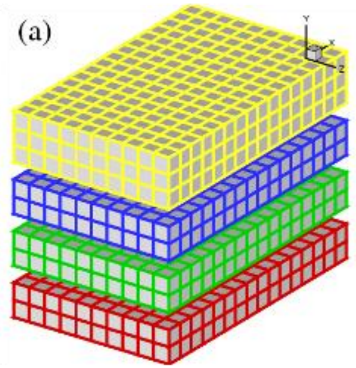


Image **N³**
N processes max
-> Not so much for HPC!

    - ✓ Decomposition 2D (pencils)

http://www.2decomp.org/

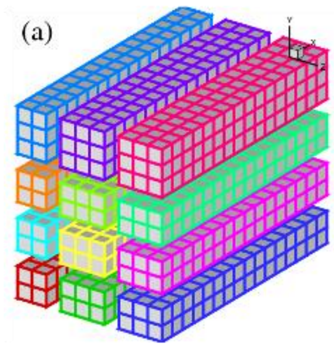


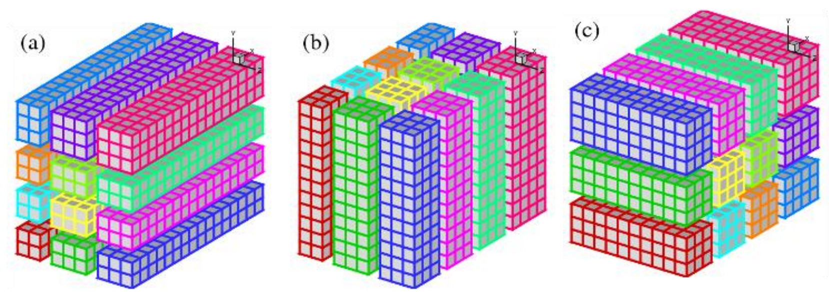
Image **N³**
N² processes max

  - 3D-FFT = succession of 1D-FFT

Requires the transposition of data

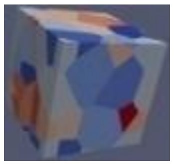
- Communications (MPI_ALLTOALL)!
  - 2decomp library

http://www.2decomp.org/



The AMITEX_FFTP code

- Highly parallel implementation (MPI)



- *Polycrystal (voronoï),* ***dislocation-based Crystal Plasticity (49 var.int.)****, Small Strains*
- *Cluster poincare (Maison de la Simulation)* 16 cores (2x8) / node sandy bridge E5-2670

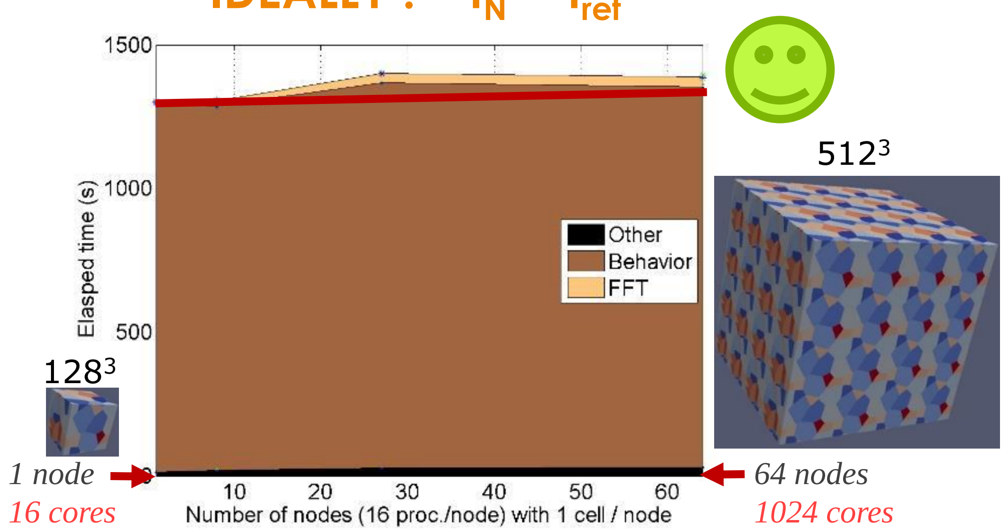
Weak scalability

Number of nodes = **N**, Problem size = **NxK0**

Elapsed time on 1 node : t_{ref}

Elapsed time on N nodes : t_N

IDEALLY : $t_N = t_{ref}$



The AMITEX_FFTP code

mpirun amitex_fftp –nm **mate.vtk** -nz **zone.vtk** -m **Material.xml** -c **Loading.xml** -a **Algorithm.xml** -s result

To run amitex in // — Geometry — The output

❑ Input geometry : 3D images (vtk format) : **mate.vtk** and **zone.vtk**

The microstructure consists of :

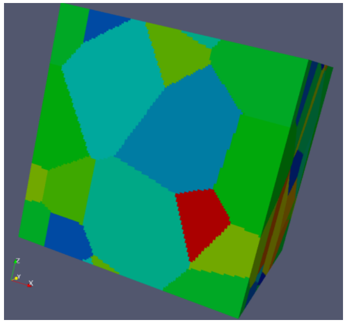
- one or different **materials** (one material = one constitutive law)

  mate.vtk : 3D image defining the ID of the materials
- each material can be devided in different **zones** (where the coefficients are constants)

  zone.vtk : 3D image defining the ID of the zones

<u>In our case (a polycrystal) :</u>

- only ONE material : "-nm **mate.vtk**" can be omitted (the ID is 1 everywhere)
- the grain definition is given by the 3D image **zone.vtk**



❑ The default output :

- unit cell and "per material" average (and std dev.) of stresses and strains at each computation time

The AMITEX_FFTP code

- Algorithm.xml

Algorithm = **Fix-point** (Basic_Scheme)
+ **Convergence acceleration**
+ default criterion : 10^{-4}

Use of the **modified Discrete Green Operator** *(Filter Type="Default")*
+ **Finite Strains** framework

```xml
<?xml version ="1.0" encoding="UTF-8"?>
<Algorithm_Parameters>

  <Algorithm Type="Basic_Scheme">
    <Convergence_Acceleration Value="true"/>
    <Convergence_Criterion Value="Default"/>
  </Algorithm>

  <Mechanics>
    <Filter Type="Default"/>
    <Small_Perturbations Value="false"/>
  </Mechanics>

</Algorithm_Parameters>
```

The AMITEX_FFTP code

❑ Loading.xml

Additionnal outputs :
Stress and strain fields
« Per zone » average and std dev.

Computation time discretization

Increments for fields outputs (vtk format)
Number of « per zone » output (equally spaced in time)

Evolution the average applied tensor :
9 components
if Driving = Stress : Piola-Kirchoff stress
if Driving = Strain : Displacement gradient

Remark : 6 components in small strain framework
if Driving = Stress : Cauchy stress
if Driving = Strain : Linearized Strain

```
<?xml version="1.0" encoding="UTF-8"?>
<Loading_Output>

<!-- OUPUT QUANTITIES -->
<Output>
   <vtk_StressStrain Strain = "1" Stress = "1"/>
   <Zone numM="1"> </Zone>
</Output>

<!-- SUCCESSIVE LOADINGS (only one here)-->
<Loading Tag="1">
   <Time_Discretization Discretization="Linear" Nincr="1000" Tfinal="333.33"/>

   <Output_vtkList>18 40 80 160 320 640 1000</Output_vtkList>
   <Output_zone Number="50"/>

  <!-- tensile test in the z direction -->
   <xx Driving="Stress" Evolution="Constant"  />
   <yy Driving="Stress" Evolution="Constant"  />
   <zz Driving="Strain" Evolution="Linear" Value="0.1" />
   <xy Driving="Stress" Evolution="Constant"  />
   <xz Driving="Stress" Evolution="Constant"  />
   <yz Driving="Stress" Evolution="Constant"  />
   <yx Driving="Strain" Evolution="Constant"  />
   <zx Driving="Strain" Evolution="Constant"  />
   <zy Driving="Strain" Evolution="Constant"  />
</Loading>

</Loading_Output>
```

The AMITEX_FFTP code

❑ Material.xml

Lamé coefficient of the reference material

Name of the dynamic library + Name of the behavior law

Constant coefficients

Constant per zone coefficients : => Euler angles of each grain

Initial internal variables

```xml
<?xml version="1.0" encoding="UTF-8"?>

<Materials>
  <!-- REFERENCE MATERIAL -->
  <Reference_Material Lambda0=" 2.0431e+5" Mu0="0.8756e+5"/>

  <!-- MATERIAL 1 -->
  <Material numM="1" Lib="/home/gelebart/SIMULATIONS/libUmatAmitex.so" Law="umatBCCSOTERIA" >

      <Coeff Index="1" Type="Constant" Value="236.412E3"/>
       .
       .
      <Coeff Index="27" Type="Constant" Value="200"/>
      <Coeff Index="28" Type="Constant_Zone" File="/home/gelebart/SIMULATIONS/PHI1.or" />
      <Coeff Index="29" Type="Constant_Zone" File="/home/gelebart/SIMULATIONS/PHI.or" />
      <Coeff Index="30" Type="Constant_Zone" File="/home/gelebart/SIMULATIONS/PHI2.or" />

      <IntVar Index="1" Type="Constant" Value="0."/>
      <IntVar Index="2" Type="Constant" Value="0."/>
       .
       .
      <IntVar Index="107" Type="Constant" Value="0."/>
  </Material>

</Materials>
```

The AMITEX_FFTP code

❑ The SOTERIA umat BCC implementation :

- A set of **Fortran files** (L. Vincent)

```
$ ls *.F

det.F
inv.F
jaconr86.F
ugd_algebre.F
ugd_cine.F
ugd_rk21.F
ugd_rk43.F
ugd_rkmod.F
ugd_umat.F
umat.F
```

- Compatible with the ***umat*** format

```
$ more umat.F

 SUBROUTINE UMATBCCSOTERIA ( STRESS, STATEV, ddsdde, sse, spd, scd,
   &            rpl, dasdat, drplde, drpldt,
   &            STRAN, DSTRAN, TIME, DTIME,
   &            TEMP, DTEMP, PREDEF, DPRED,
   &            CMNAME, NDI, NSHR, NTENS, NSTATV,
   &            PROPS, NPROPS, COORDS,
   &            drot, pnewdt, celent, DFGRD0, DFGRD1,
   &            NOEL, NPT, layer, kspt, KSTEP, KINC )
.
.
.
```

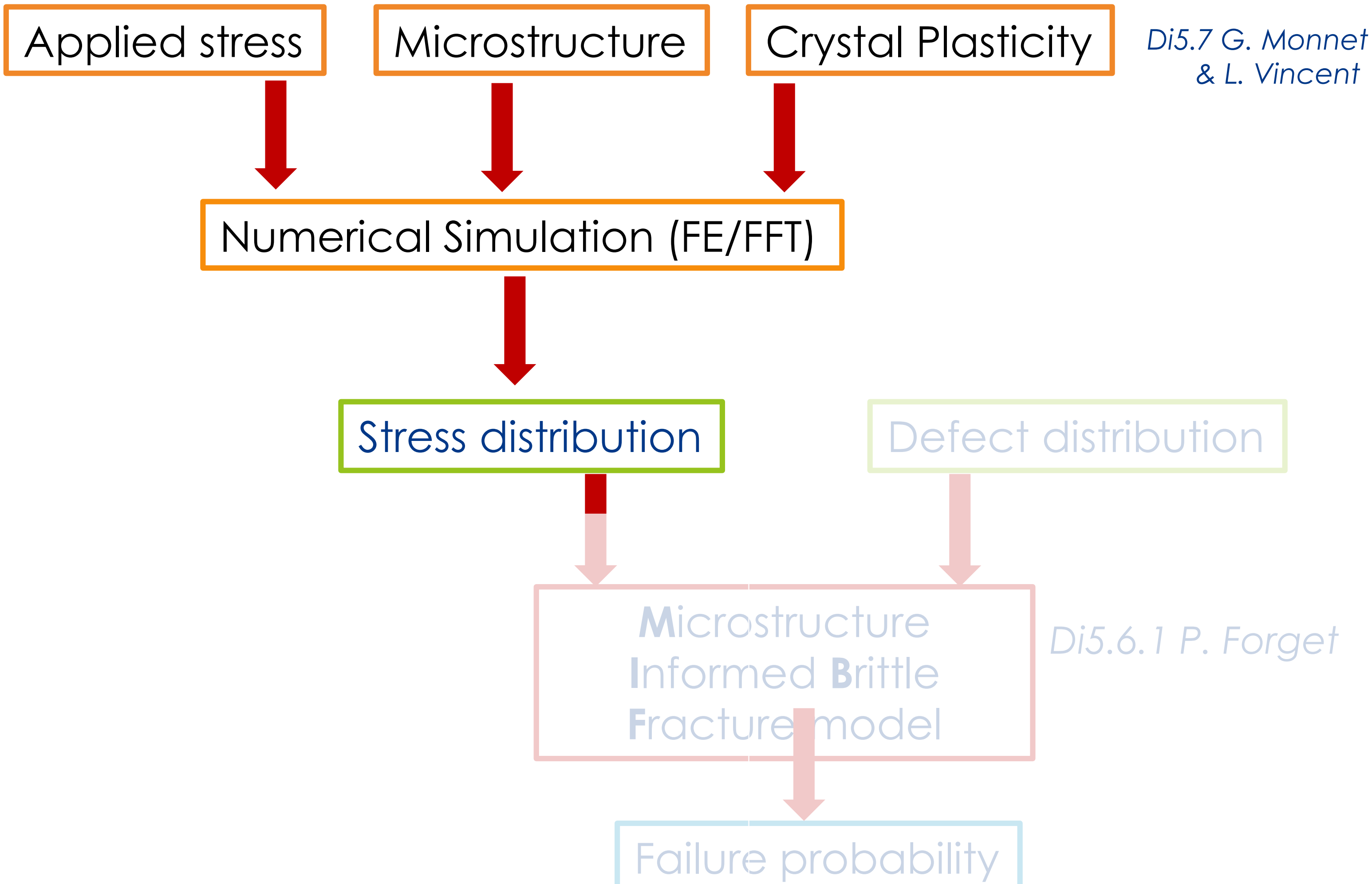
- a ***Makefile*** to automatically generate the dynamic library *libUmatAmitex.so* from the Fortran files

- The **same Fortan implementation** of the behavior law can be used in :
  - AMITEX_FFTP
  - CAST3M *(the CEA FEM code)*

- No need to modify the AMITEX source code to introduce a new behavior law !

SOTERIA context

- Reliability of the Reactor Pressure Vessels



SOTERIA context

- Reliability of the Reactor Pressure Vessels

PERFOM60 (2007-2011)

FE

- Per grain average stress

SOTERIA (2014-2018)

FFT-based simulation

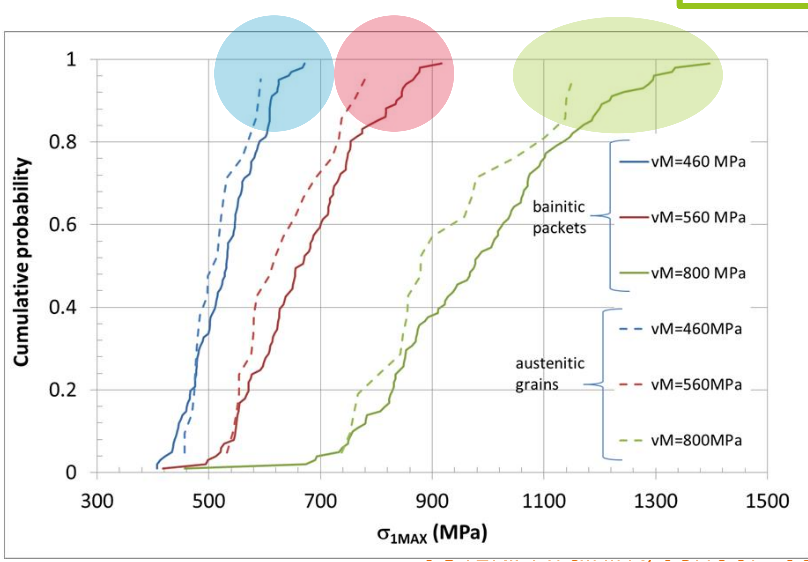
- Per grain average stress
- Large number of grains (>1000?)
- Average stress at Grain Boundaries
- Improved Crystal Plasticity law

INTRA-granular brittle fracture

Stress distribution

INTER-granular Brittle fracture

Microstructure **I**nformed **B**rittle **F**racture model



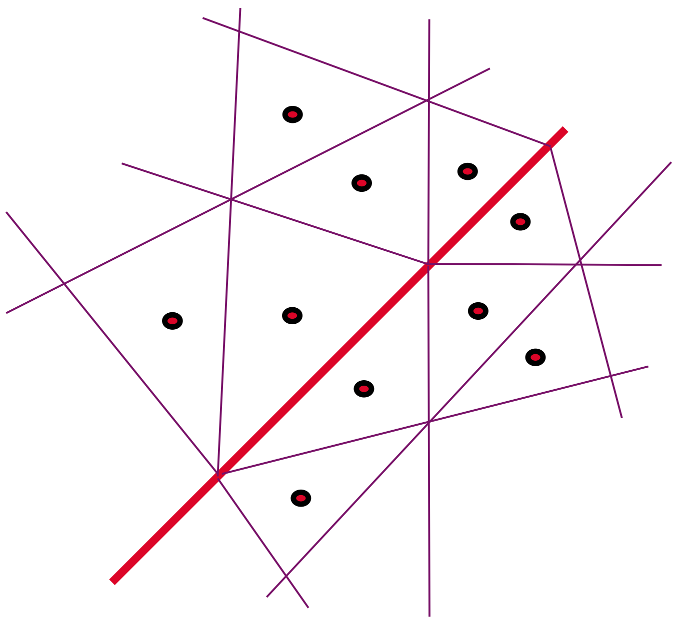
Stresses at grain boundaries

- Methodologies to evaluate stresses at grain boundaries from FFT simulations
- Validation with FEM (and a conforming mesh)
- Extension to Finite Strains
- Application to RPV steels

Stresses at grain boundaries

- Remark for FEM (conforming mesh)

Stresses are defined at Gauss Point (not on the grain boundary)



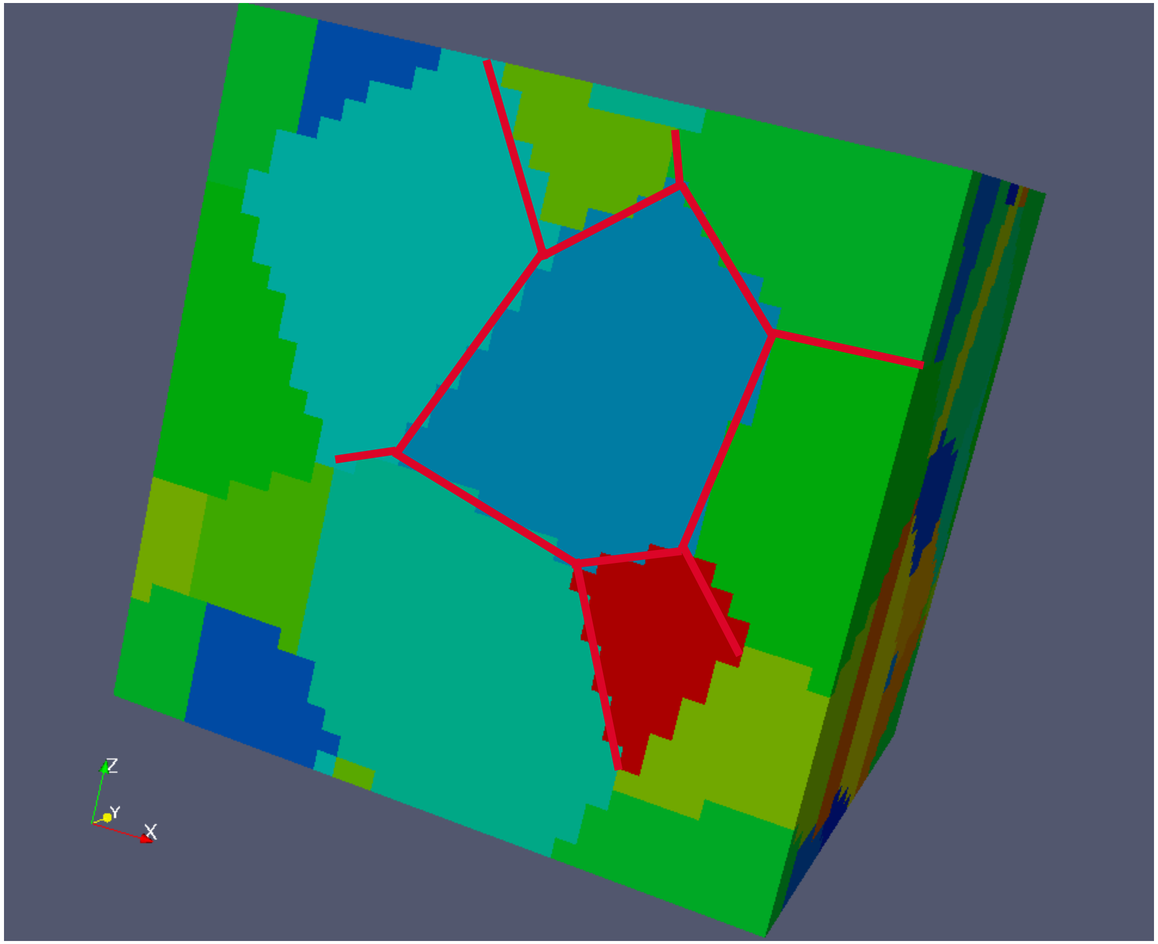
Grain Boundary (2D)

Even with a conforming mesh, the evaluation of stresses at grain boundaries **is not straightforward!**

Stresses at grain boundaries

- Methodology for FFT (regular grid)

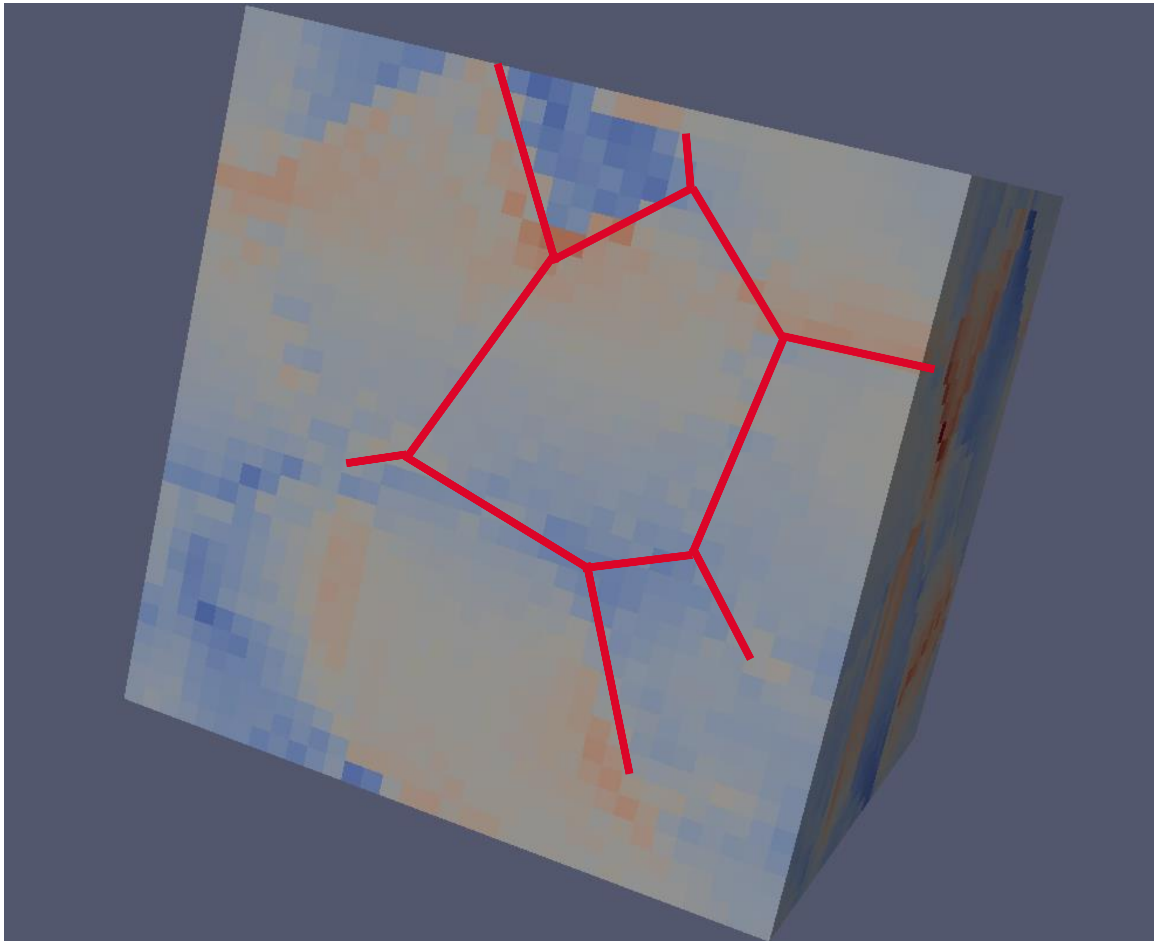
The « mesh » does not coincide with grain boundaries



Stresses at grain boundaries

- Methodology for FFT (regular grid)

Stresses are defined on voxels



<u>PROPOSITION 1</u>

Post-treatment:
projection of the stress field on grain boundaries

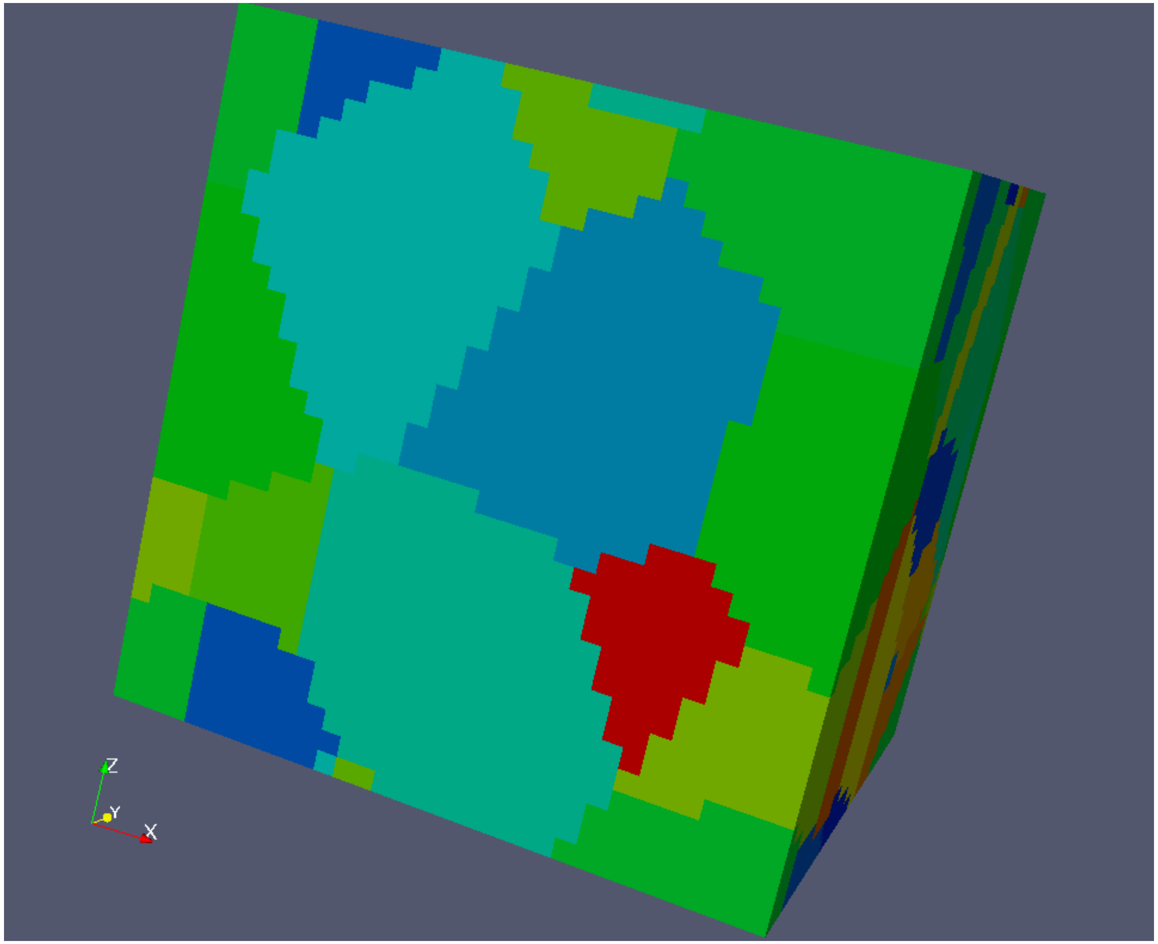
Stresses at grain boundaries

- Methodology for FFT (regular grid)

PROPOSITION 2

« Composite » voxels to account for grain boundaries in FFT simulations

+ PROPOSITION 1 (post treatment)



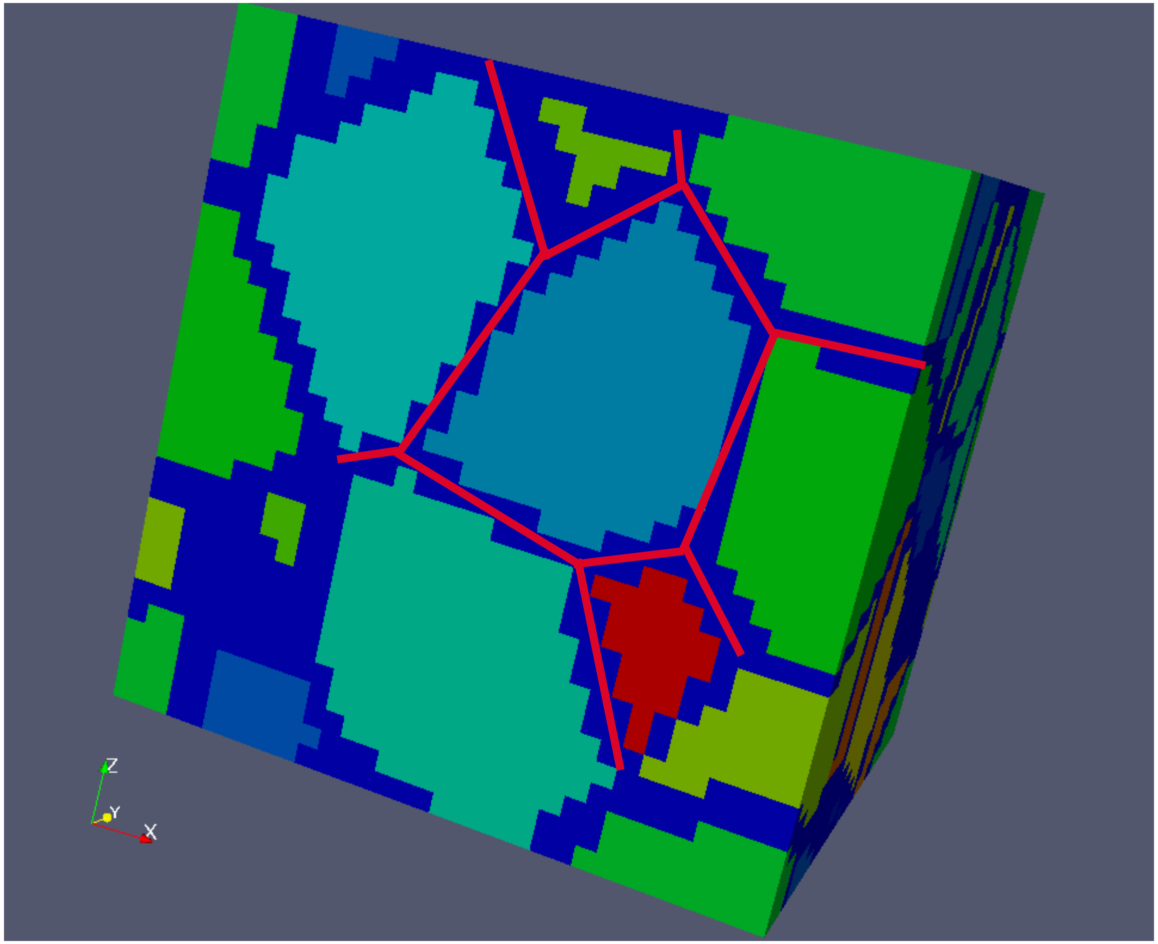
Stresses at grain boundaries

- Methodology for FFT (regular grid)

<u>PROPOSITION 2</u>

« Composite » voxels to account for grain boundaries in FFT simulations

+ PROPOSITION 1 (post treatment)



Stresses at grain boundaries

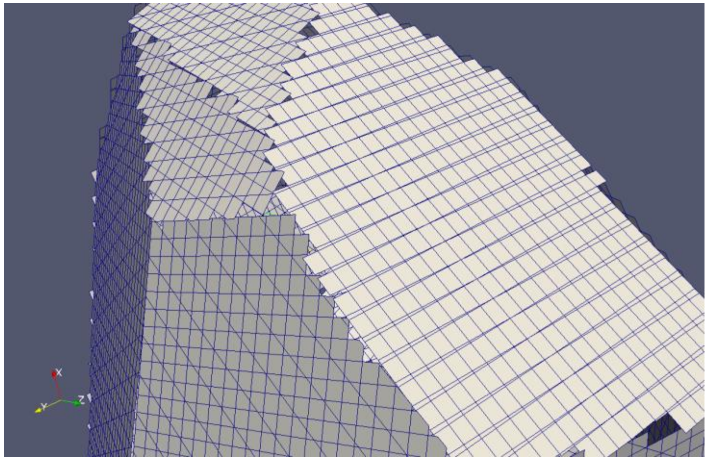
❑ Methodology for FFT (regular grid)

DEVELOPMENT OF SPECIFIC PRE AND POST TREATMENTS

PRE-TREATMENT

✓ **Grain Boundary decomposition**

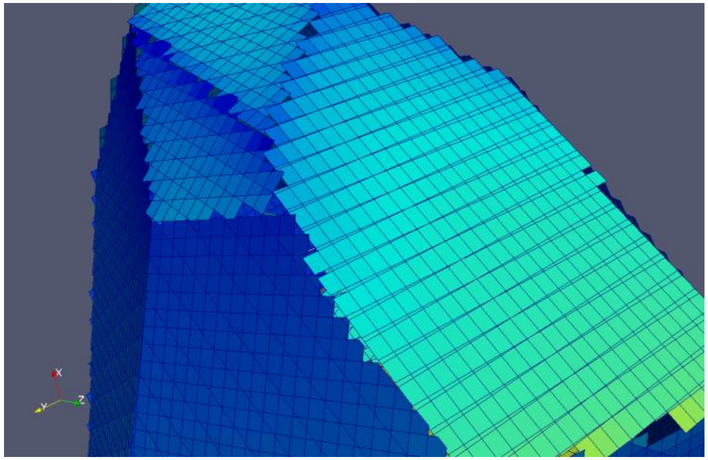
- Each **GB** is devided into « **facets** »
- One « facet » = **intersection between a GB and voxel**
  - ➢ polygon with 3, 4, 5 or 6 corners
  - ➢ *To be improved for triple lines*
- **One « composite » voxel :**
  - ➢ **Facet area S_i**
  - ➢ **Volume Fractions**
  - ➢ **Normal vector n_i**



POST

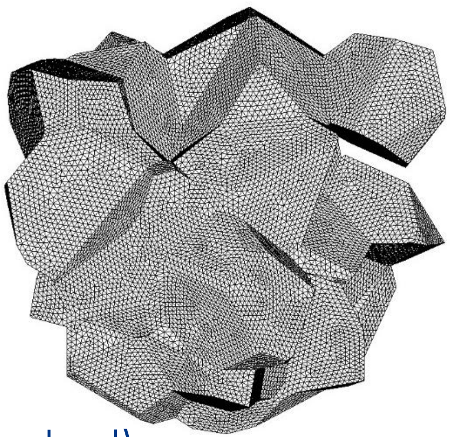
✓ **Evaluating average normal stress at Grain Boundaries**

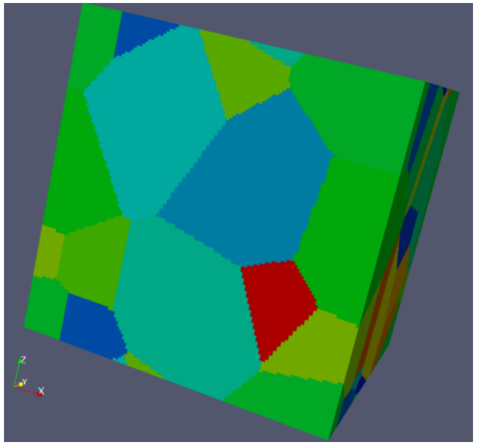
$$t = \frac{\sum n_i.(\sigma_i.n_i)S_i}{\sum S_i}$$



Stresses at grain boundaries

- Validation with FEM (CAST3M)





- A 27grains periodic Voronoi aggregate

- A simple Crystal Plasticity law

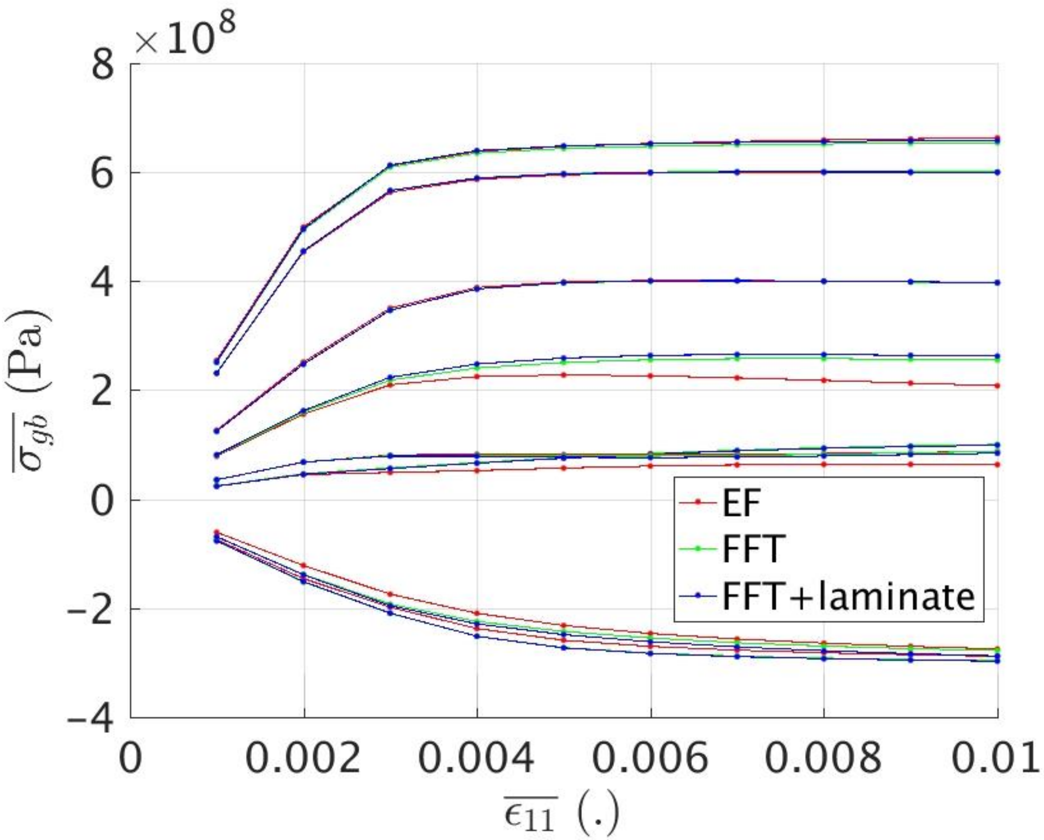
  - ➡ Anisotropic elasticity (austenitic steel)
  - ➡ **12** slip systems (FCC)
  - ➡ Norton law (τ_0=200MPa, **n=10**, $\dot{\gamma}_0=10^{-4}s^{-1}$)

- Loading : uniaxial tensile test (1%, $10^{-4}s^{-1}$)

Stresses at grain boundaries

- Result :« Per Grain Boundary » average normal stresses (167 Grain Bound.)

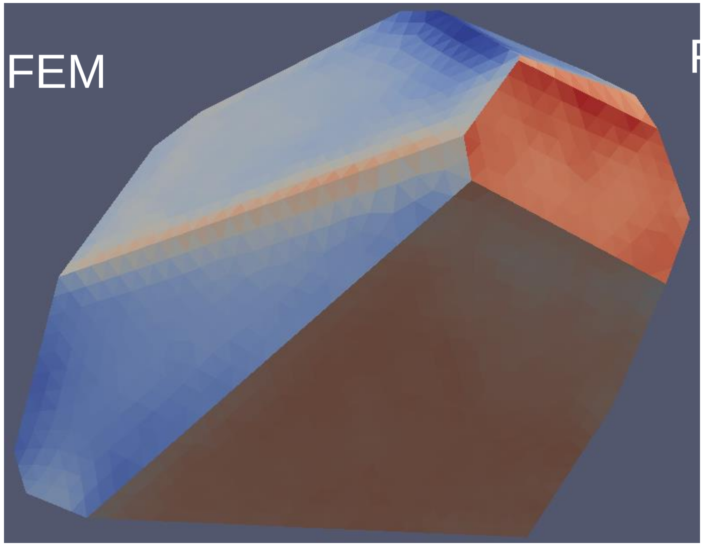
8 Grain Boundaries
4 with the best FE/FFT agreements
4 with the worst agreements

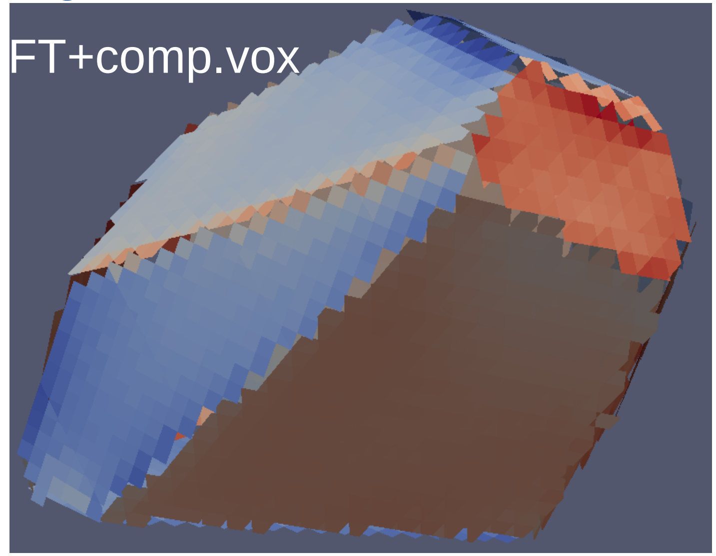


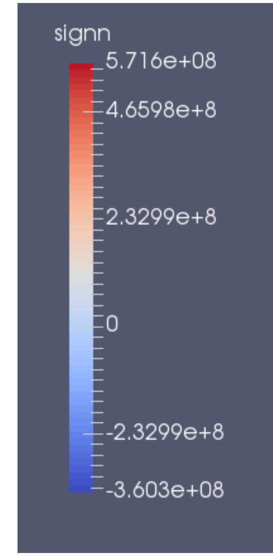
- ✓ Good agreement FFT/FEM
- ✓ FEM probably not fully converged
- ✓ No significant effect of composite voxels (FFT+laminate)

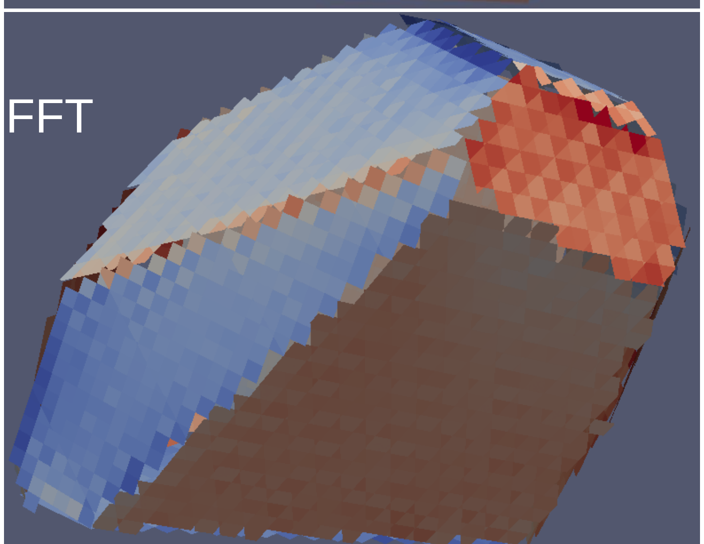
Stresses at grain boundaries

- Result : normal stress field at grain boundaries









Composite voxels « smooth »
spurious oscillations

Stresses at grain boundaries

SOTERIA

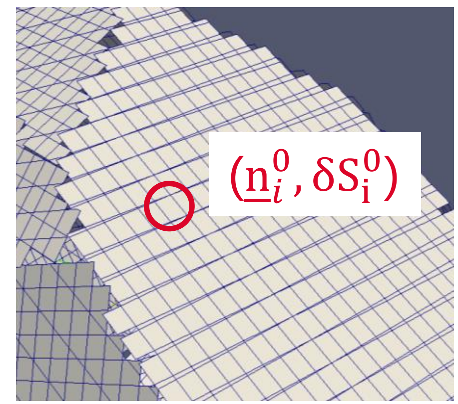
- Extension to Finite Strains

The grain boundary (initially plane) : rotates and deforms !

✓ **FEM** : evaluation of the normal stress on the deformed mesh of the GB

✓ **FFT** : the GB is not meshed explicitely… but

- The intersection of GB with the grid is known in the reference configuration $(\underline{n}_i^0, \delta S_i^0)$



- Transport equation of an infinitesimal surface vector:

$$\underline{n} dS = \det(\mathbf{F})\mathbf{F}^{-T} . \underline{n}^0 dS^0$$

$$\underline{n}_i \delta S_i = \det(\mathbf{F}_i)\, \mathbf{F}_i^{-T} . \underline{n}_i^0 \delta S_i^0 \qquad \delta S_i = \det(\mathbf{F}_i) \left\| \mathbf{F}_i^{-T} . \underline{n}_i^0 \right\| \delta S_i^0 \qquad \underline{n}_i = \frac{\mathbf{F}_i^{-T} . \underline{n}_i^0}{\left\| \mathbf{F}_i^{-T} . \underline{n}_i^0 \right\|}$$

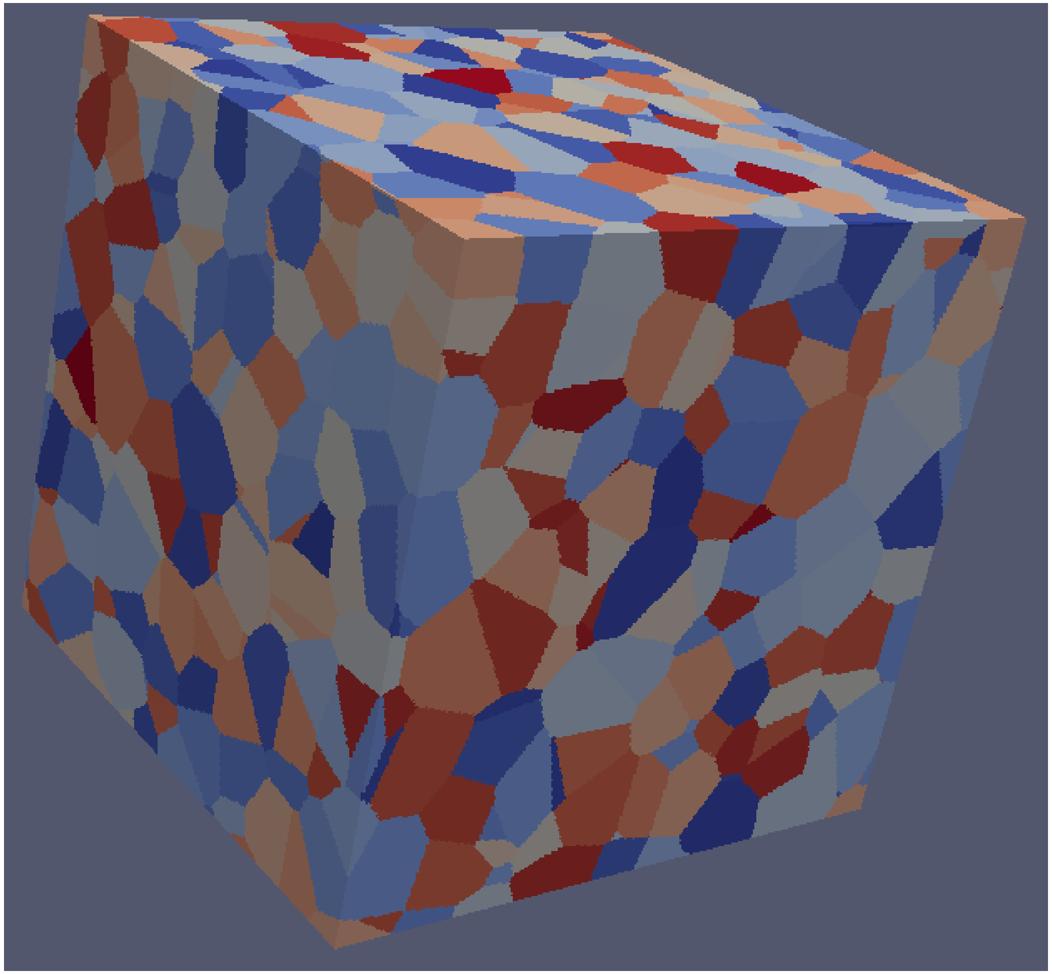
- Surface average of the normal stress in the deformed configuration

$$\overline{\sigma_{gb}} = \frac{\int_S \underline{n} . \boldsymbol{\sigma} . \underline{n} dS}{S} \cong \frac{\sum \underline{n}_i . \boldsymbol{\sigma}_i . \underline{n}_i \delta S_i}{\sum \delta S_i}$$

Application to RPV

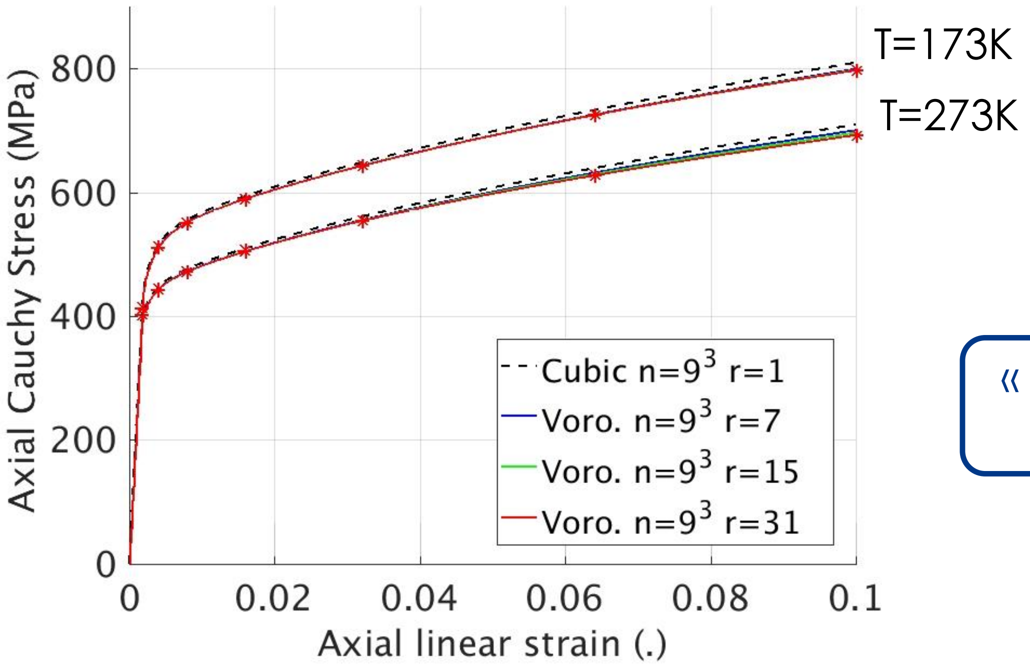
- SOTERIA Crystal Plasticity law for BCC (Finite Strains) (G.Monnet EDF & L. Vincent CEA)

  - Microstructure
    - Voronoï **729** grains
    - Resolution r = 7, 15, 31
  - Temperatures : 173K, 273K
  - Tensile test : **10%**, 3.10^{-4} s^{-1}



Application to RPV

❑ Macroscopic Behavior



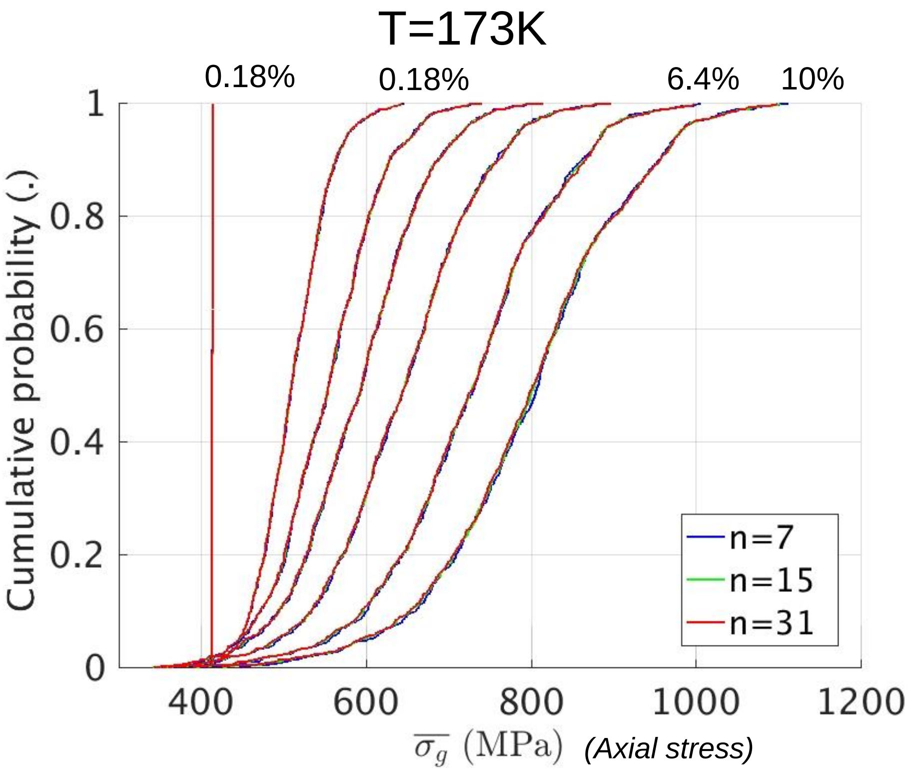
« cubic » simplified microstructure
OK for macroscopic behavior

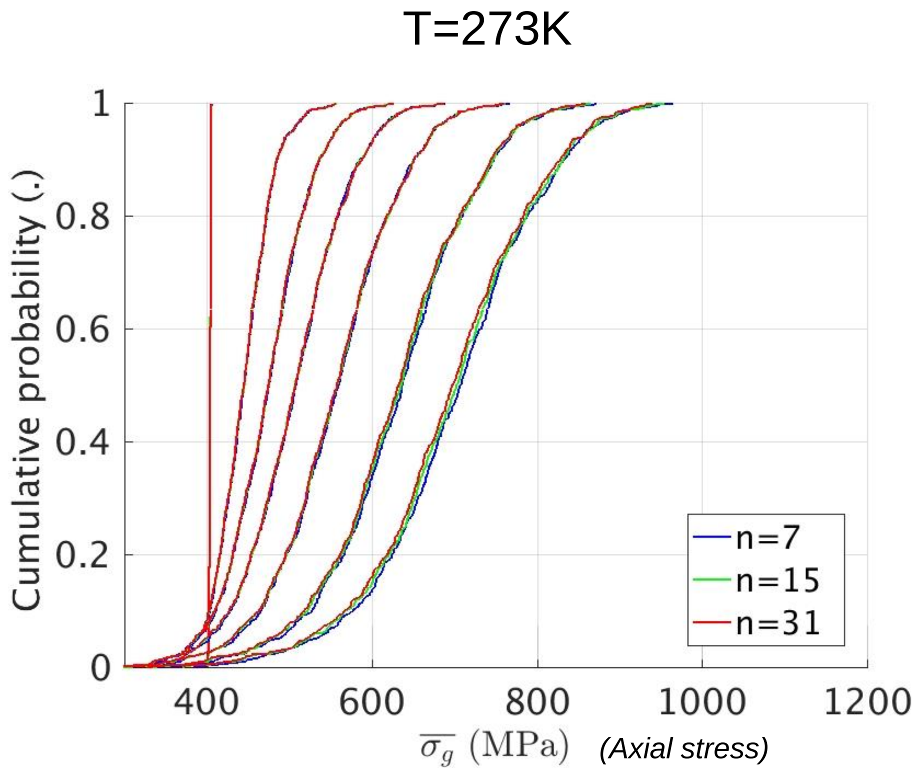
729 grains

Cubic	r=1 :	~ 5min	/1 node (12 procs)
Voronoï	r=7 :	~30 min	/10 nodes (280procs)
	r=15:	~3h	/10 nodes (280procs)
	r=31:	~10h	/30 nodes (840procs)

Application to RPV

- **Per Grain** average stress distribution (729 grains)

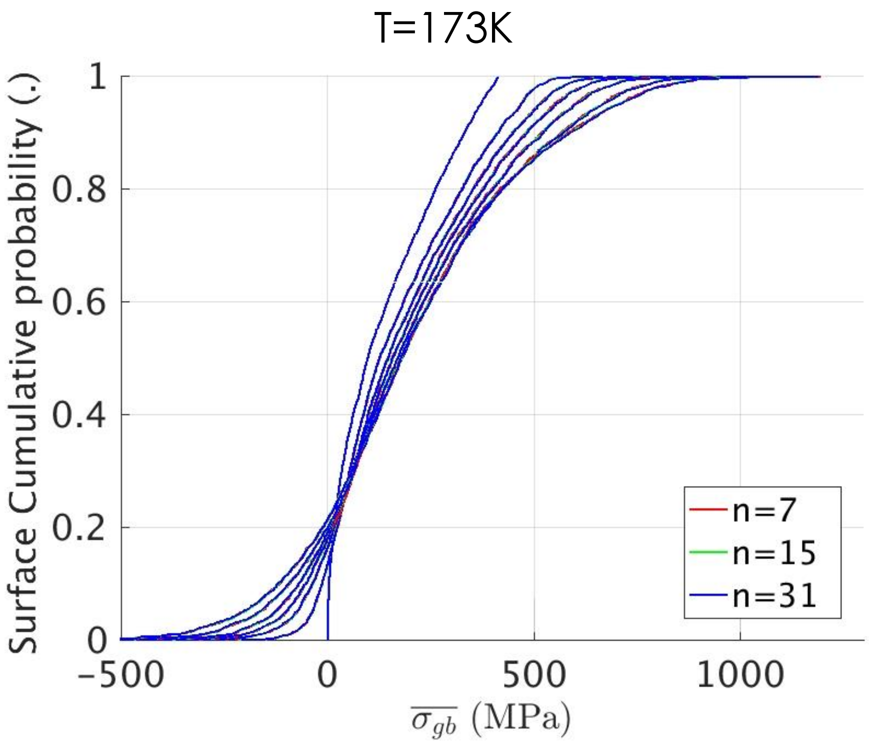


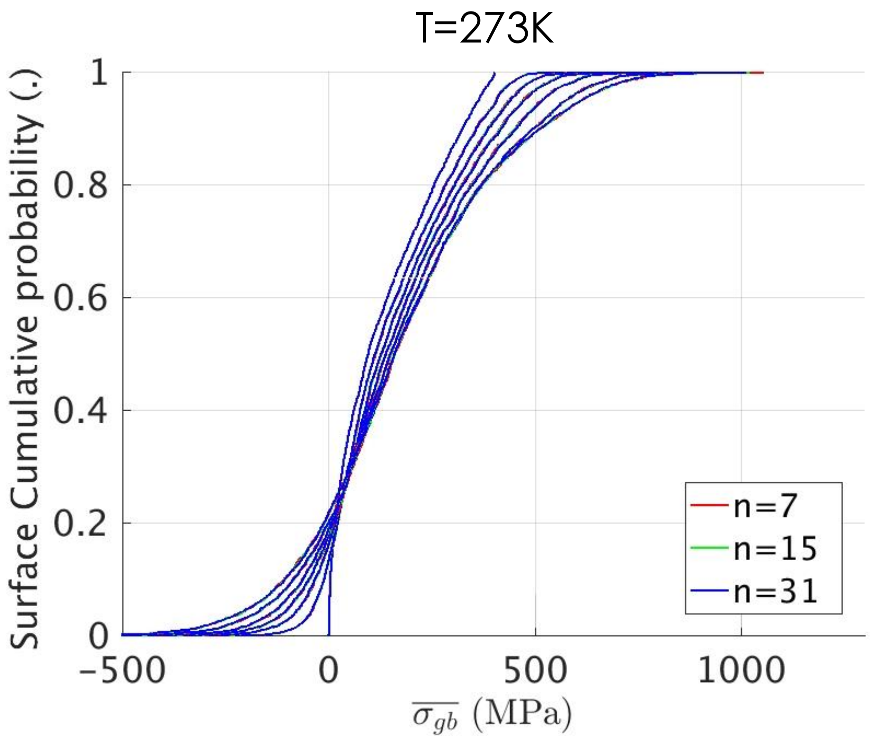


Input for MIBF for **INTRA-**granular fracture
Resolution 7 is enough !

Application to RPV

- **Per Grain Boundary** average stress distribution (4445 GB) 99% of the total area

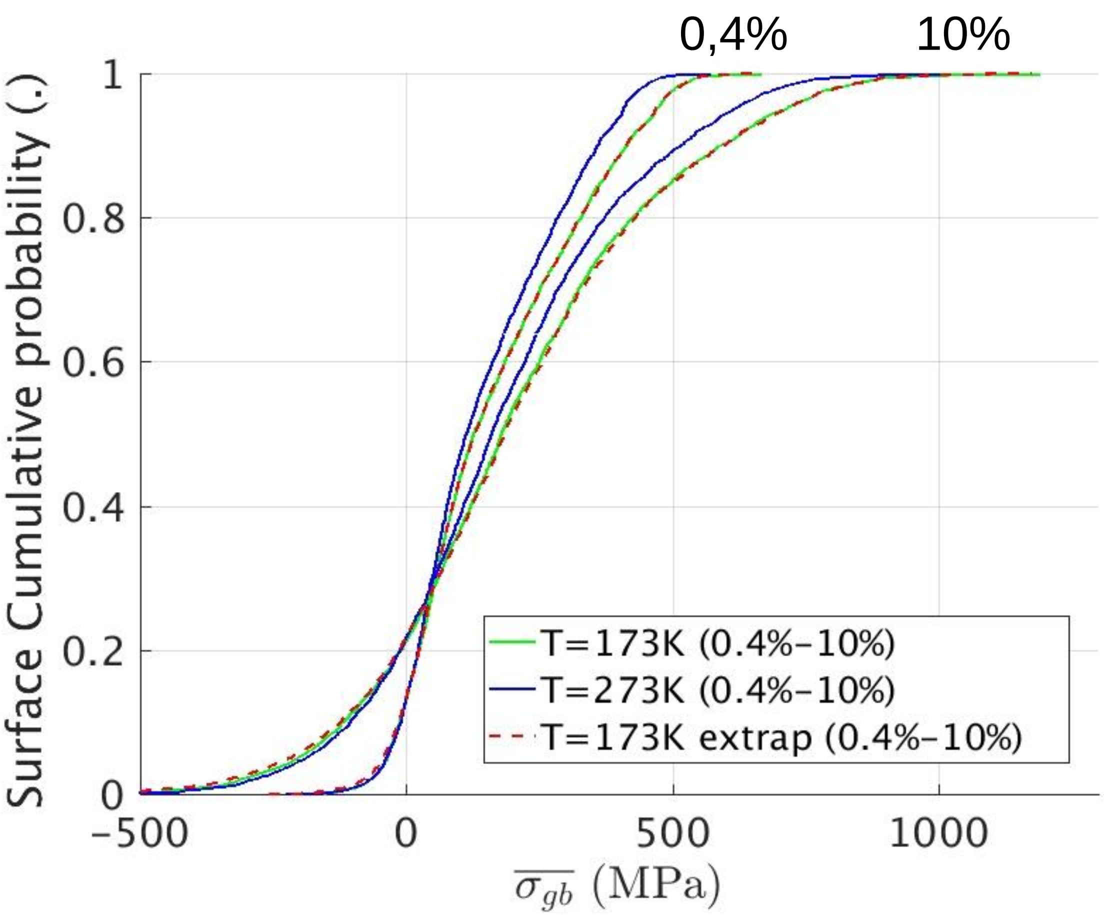




Input for MIBF : **INTER-**granular fracture
Resolution 7 is enough !

Application to RPV

- **Per Grain Boundary** average stress distribution (4445 GB) 99% of the total area



Temperature extrapolation ?

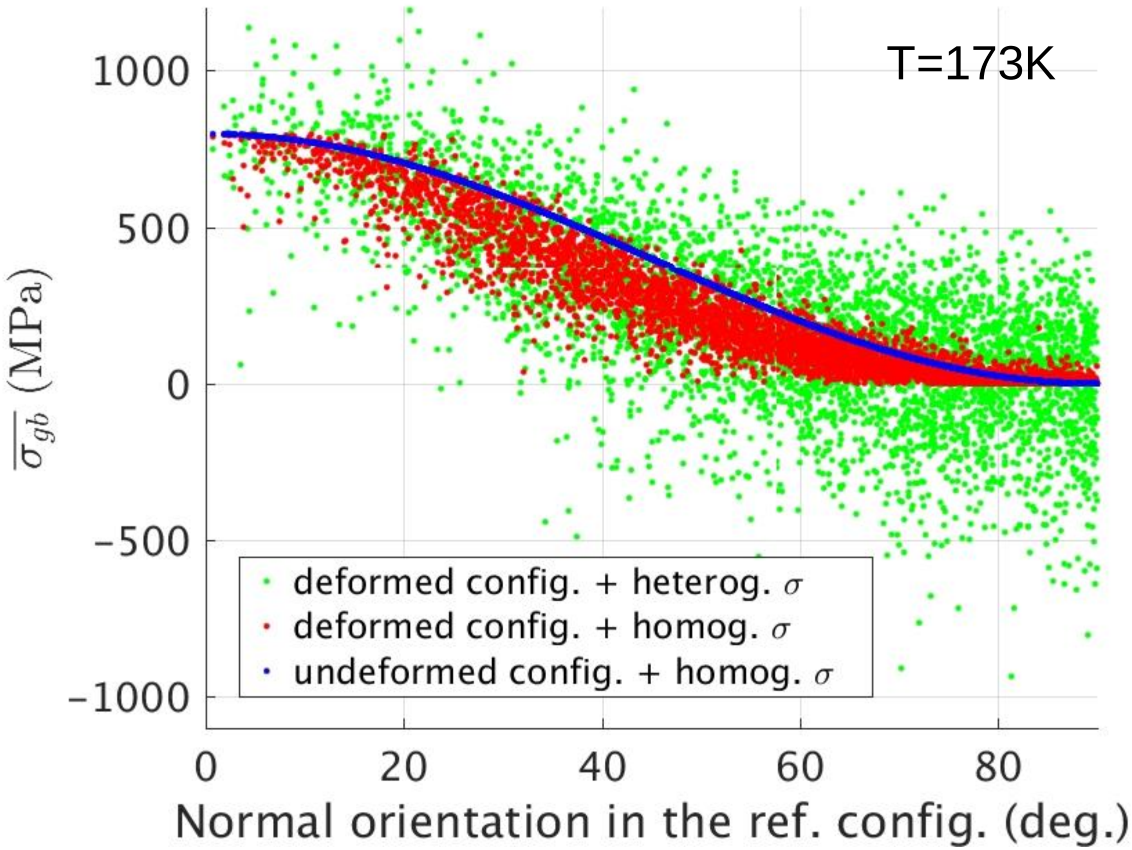
$$F_{173K}\left(\overline{\sigma_{gb}}\right) \cong F_{273K}\left(\overline{\sigma_{gb}} \times \alpha\right)$$

with $\alpha = \dfrac{\overline{\sigma}_{173K}}{\overline{\sigma}_{273K}}$

Application to RPV

- **Per Grain Boundary** average stress distribution (4445 GB) 99% of the total area

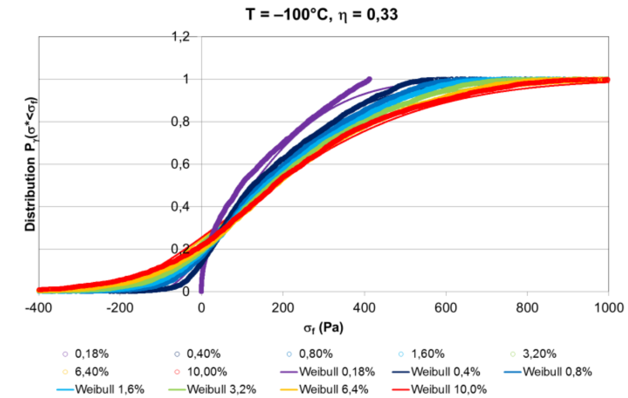
  Effect of the GB rotation (and deformation)?

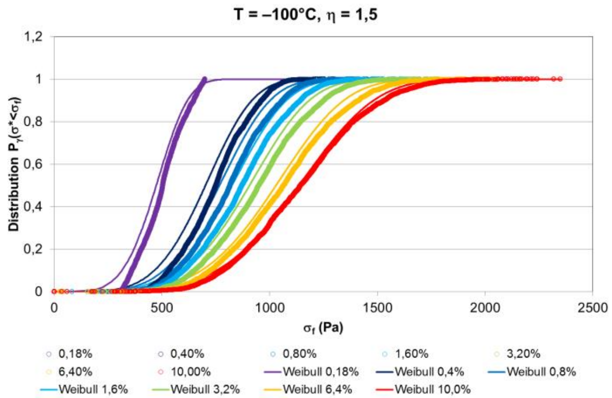


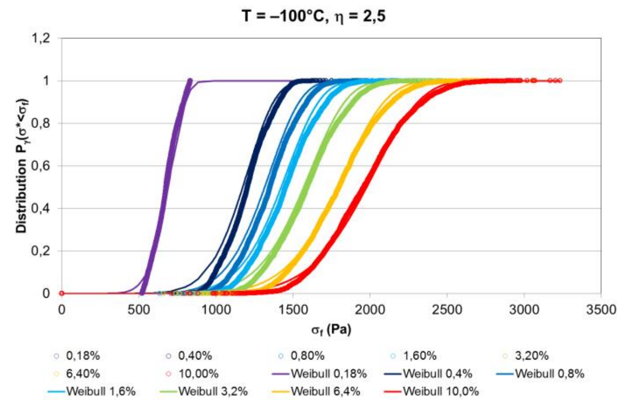
The rotation (and deformation) of the GBs is significant (red points)

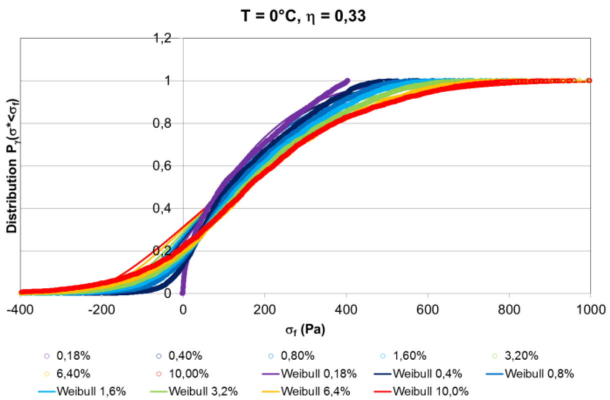
Application to RPV

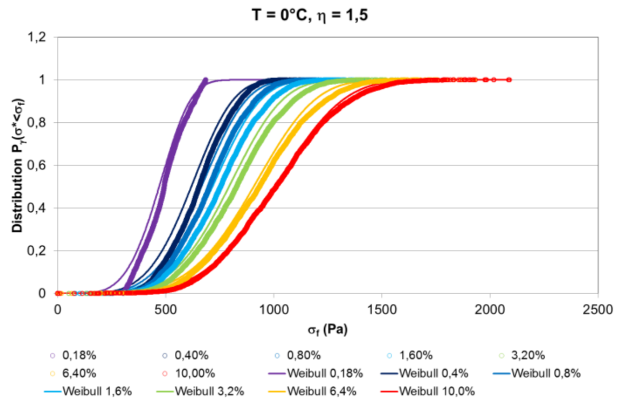
❑ **USE for MIBF : a model fitted on the distribution of stresses at GB** (P. Forget Di5.6.1)

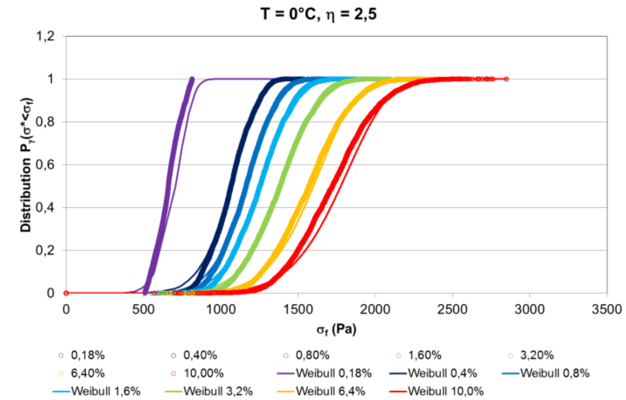












Conclusions

❑ GENERAL context

- FFT-based solvers :

  a very powerfull technique for the simulation of heterogeneous materials

- AMITEX_FFTP :

  efficient parallel code,

  quite general (lots of possible applications),

  quite simple to use

❑ SOTERIA context

- Evaluation of stresses at GB : OK with FFT

- Application to RPV :