



Micromechanical Analysis of Stress-Strain Inhomogeneities with Fourier transforms (MASSIF)

Anthony Rollett, Ricardo Lebensohn¹, Sukbin Lee⁵,
Seth Wilson², Benjamin Anglin³, Sean Donegan⁴, Reeju
Pokharel¹, Evan Lieberman¹ and Tugce Ozturk

Materials Science & Engineering, Carnegie Mellon Univ.

¹Materials Science & Technology, Los Alamos Natl. Lab.

²Ames Lab., ³Bettis Lab., ⁴Bluequartz, ⁵UNIST, Korea

*Discussions with P. Suquet, H. Moulinec, F. Franchetti, S. Berbenni;
support from HPCMO/PETTT, NDSEG, BES, NSF, and CMSN acknowledged*



Carnegie Mellon

CARNEGIE MELLON UNIVERSITY

DEPARTMENT OF
MATERIALS SCIENCE AND ENGINEERING

DREAM.3D: FFT output

The screenshot displays the DREAM.3D software interface for configuring a pipeline. The main window is titled "Untitled Pipeline - DREAM3D".

Filter Library: Shows a tree view with categories: Filter Library, Core, Generic, IO, Output, and Input. Under IO, there are filters: Read DREAM.3D Data File, Read Dx File (Feature...), and Import Orientation F...

Filter List: Contains a search bar and a list of filters including: Abaqus Hexahedron Writer, Add Bad Data, Add Orientation Noise, Align Sections (Feature Centroid), Align Sections (Feature), Align Sections (List), Align Sections (Misorientation), Align Sections (Mutual Information), and Calculate Frequency Histogram. The filter count is 160.

Pipeline Steps: The pipeline consists of two steps: [1] Read DREAM.3D Data File and [2] Write Los Alamos FFT File. The second step is highlighted with a blue border.

Write Los Alamos FFT File Configuration:

- Parameters:** Output File is set to `/Users/rollett/code/3d_FIB/Small_IN100_Output/IN`.
- Required Objects:**
 - Cell Data:** Feature Ids, Phases, and Euler Angles are all set to `VolumeDataContainer` and `CellData`.

Bottom Panels:

- Bookmarks:** A table with columns Name and Path.
- Pipeline Issues:** A table with columns Filter, Description, and Code.
- Prebuilt Pipelines:** Shows a tree view under Workshop > Reconstruction, listing steps (01) through (09) such as SmallIN100 Import, Initial Visualization, Threshold, Alignment, Cleanup, Segmentation, Merge Twins, and Minimum Size.

Status Bar: Displays "Pipeline Complete".

Outline: Status

- Mathematical basis for MASSIF *aka* the “FFT method”, which discretizes a microstructure on a regular grid in order to use Fast Fourier Transforms to solve the (partial) differential equations for stress equilibrium.
- Wide range of applications of a spectral method to calculating microstructure-property relationships in polycrystals using Green’s functions and Fast Fourier transforms (FFTs).
- The method was originated by Moulinec & Suquet (for elastic loading) with a focus on composite materials (in turn based on work by Kröner); it was further developed by Lebensohn (for viscoplastic deformation) with a focus on polycrystalline materials (metals, ceramics, ice).
- Rollett’s group adopted the FFTW package and made it MPI parallel (within FFTW). An alternative parallel FFT scheme (3d instead of slab) is also available, based on work by Yang Wang (PSC).
- Recent developments include thermoelastic, elasto-viscoplastic and dual grid (e.g. void growth); Roters and Eisenlohr have incorporated the FFT as a solver inside FE (<http://damask.mpie.de>).

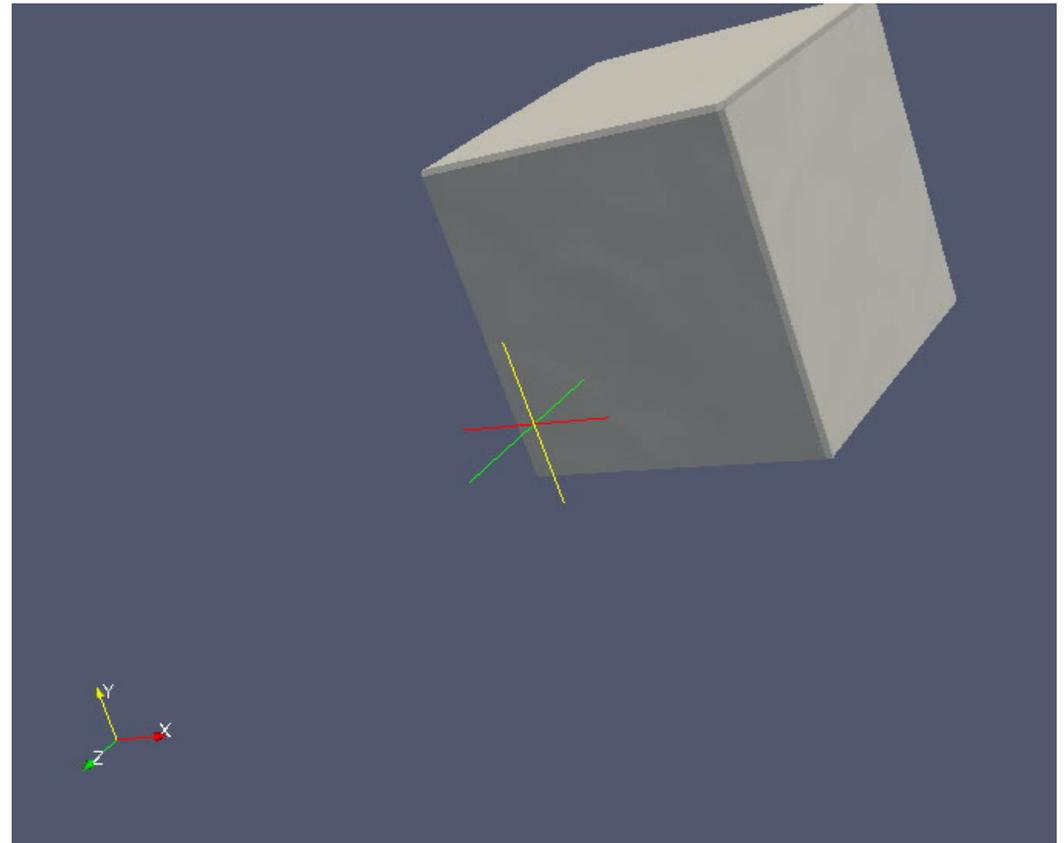
5 **Status, Big Data, RVE, Validation**

- Availability of the MASSIF (FFT) code(s): on request to lebenso@lanl.gov, rollett@cmu.edu.
- Scalability:
 - The computational part of the runtime scales as well as the FFT i.e. nearly linear in the number of gridpoints.
 - (Thermo-)Elastic calculations are fastest.
 - Viscoplastic calculations require more time because of the need to solve a 5x5 non-linear equation to get slip rates (each gridpoint, each step, each iteration).
 - Elasto-viscoplastic calculations require yet more time because of the need for small strain increments (at least through yield).
- Potential connection to big data: although not yet exercised, every reason to expect to be able to iterate back and forth with Dream3D to design microstructures and test their micromechanical response. Depending on domain size, step count etc., rapid accumulation of data.
- Domain size: appears to be small (as a number of grains) for elastic, much larger for viscoplastic.
- Validation against experimental data: for both FFT and FE with crystal plasticity, this is not complete.

6

Topics, Examples

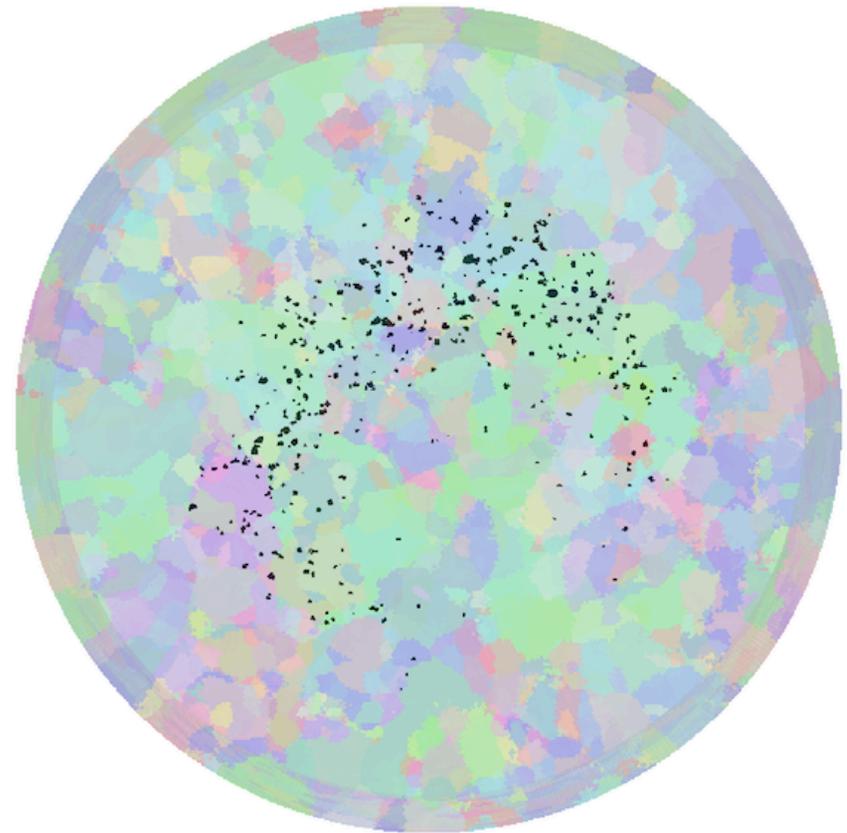
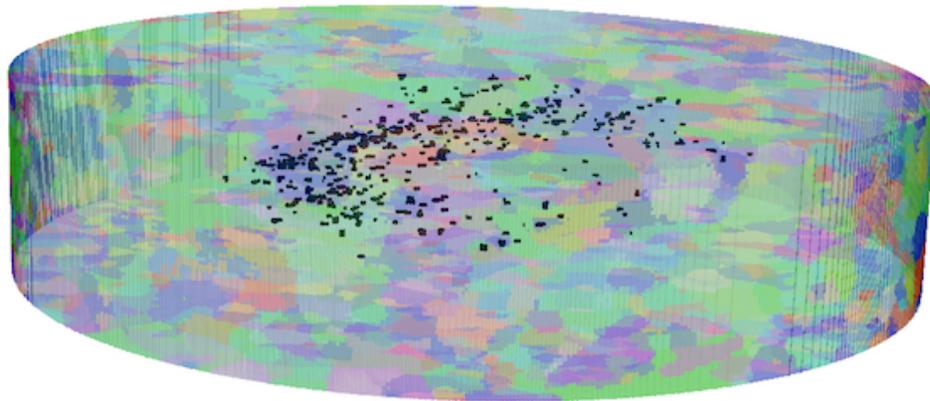
- Misorientation development in polycrystalline copper: 2D (from EBSD) and 3D (from HEDM) comparisons.
 - Elastic response of low volume fraction foams.
 - Stress hot spots during (visco-)plastic deformation in relation to microstructure.
 - Analysis (elastic) of twinning in tensile deformation of Zr polycrystal.
 - Dependence of strain (rate) distribution in a metal-metal composite with hard particles in a soft matrix, motivated by studies of W-Ni-Fe.
-
- Use of the thermoelastic (eigenstrain) method for computing stress fields between dislocations
 - Analysis (thermoelastic) of stress concentration in thermal barrier coatings and the role of interface roughness
 - Analysis of plastic deformation in ice
 - Analysis (thermoelastic) of driving forces for whisker growth from thin films.
 - Fatigue crack initiation – comparison with SEM, EBSD, HEDM data.
 - Comparisons with crystal plasticity finite element calculations.
 - Analysis (elasto-viscoplastic) of a shock experiment leading to incipient spalling in copper.



7

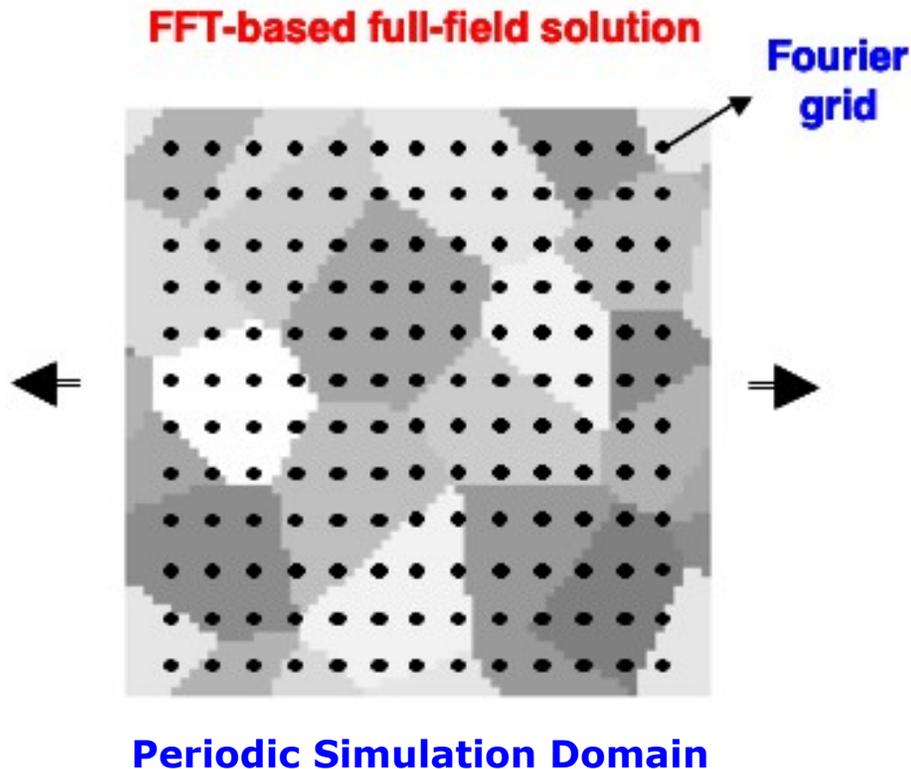
Voids from Post-Shock Image

Image of the surface of a polycrystalline Cu sample subjected to a mild shock; resulting voids superimposed on image. Elastoviscoplastic FFT being used to obtain micromechanical fields and learn about void nucleation.



Evan Lieberman, David Menasche, Bob Suter,
Ricardo Lebensohn, Curt Bronkhorst, Ed
Kober, ADR

8 Microstructure-Property Simulation with FFT



Faster than FEM for large problems (order $N \log[N]$)

Requires periodic boundary conditions

Proposed by Moulinec & Suquet for linear (1994) and non-linear composites (1998)

Extended by Lebensohn for viscoplasticity for polycrystals (2001), on a suggestion by Canova about using FFTs

Solving Stress Equilibrium

$\varepsilon(X_f)$ and $\sigma(X_f) \rightarrow$ Elasticity

$\dot{\varepsilon}(X_f)$ and $\sigma(X_f) \rightarrow$ Viscoplasticity

Moulinec & Suquet, *Comput. Methods Appl. Mech. Engrg.* **157** 69-94 (1998).

Michel, Moulinec & Suquet *CMES-Comput. Mod. Eng. Sci.* **1** 79-88 (2000).

Lebensohn, *Acta Mater.* **49** 2723-2737 (2001); *Acta Mater.* **56** 3914-3926 (2008).

Rollett, *et al.*, *MSMSE*, **18** 074005 (2010); Anglin *et al.*, *Comp. Matls. Sci.* **87** 209 (2014).

Lebensohn *et al.*, *Intl. J. Plasticity*, **32-33**, 59 (2012).

Advantages & Disadvantages of the FFT Method

- Caveat: intended for materials problems, not for solving problems with load-bearing structures. FE widely understood whereas FFT approach ~unknown.
- Advantage: no need to make a mesh. 3D meshes, especially conforming to microstructure notoriously time consuming and difficult to make a mesh that is free of element quality problems. Nevertheless, commercial solutions exist, e.g., Simpleware (also Jessica Zhang, MechE/CMU).
- Advantage: direct instantiation with 3D images from serial sectioning, 3D x-ray microscopy, or other sources.
- Drawbacks: periodic structure required in at least one direction out of three; With buffer zones, however, many materials testing situations can be modeled.
- Advantage: model microstructures can be easily generated, enabling *microstructural design* to be investigated.
- Elastic and viscoplastic versions of the model have been devised to date: an elasto-plastic model has been developed and published. A thermo-elastic version has been published. A dual grid version exists (Fourier & material grids).
- Comparisons to Finite Element method calculations show good agreement.
- Advantage: Time required for equivalent calculation is much less for the FFT method, thanks to the $N\log(N)$ scaling. E.g., for 10 millions degrees of freedom, viscoplastic, vpFFT requires of order $1/10^{\text{th}}$ time as FE (with crystal plasticity).

10 Finite Element vs. FFT (verification)

Fcc
Rolling to 40%

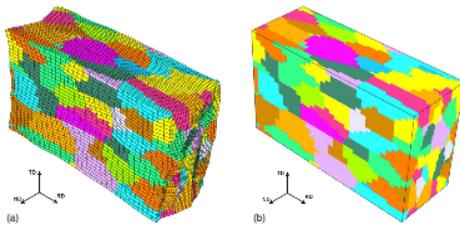
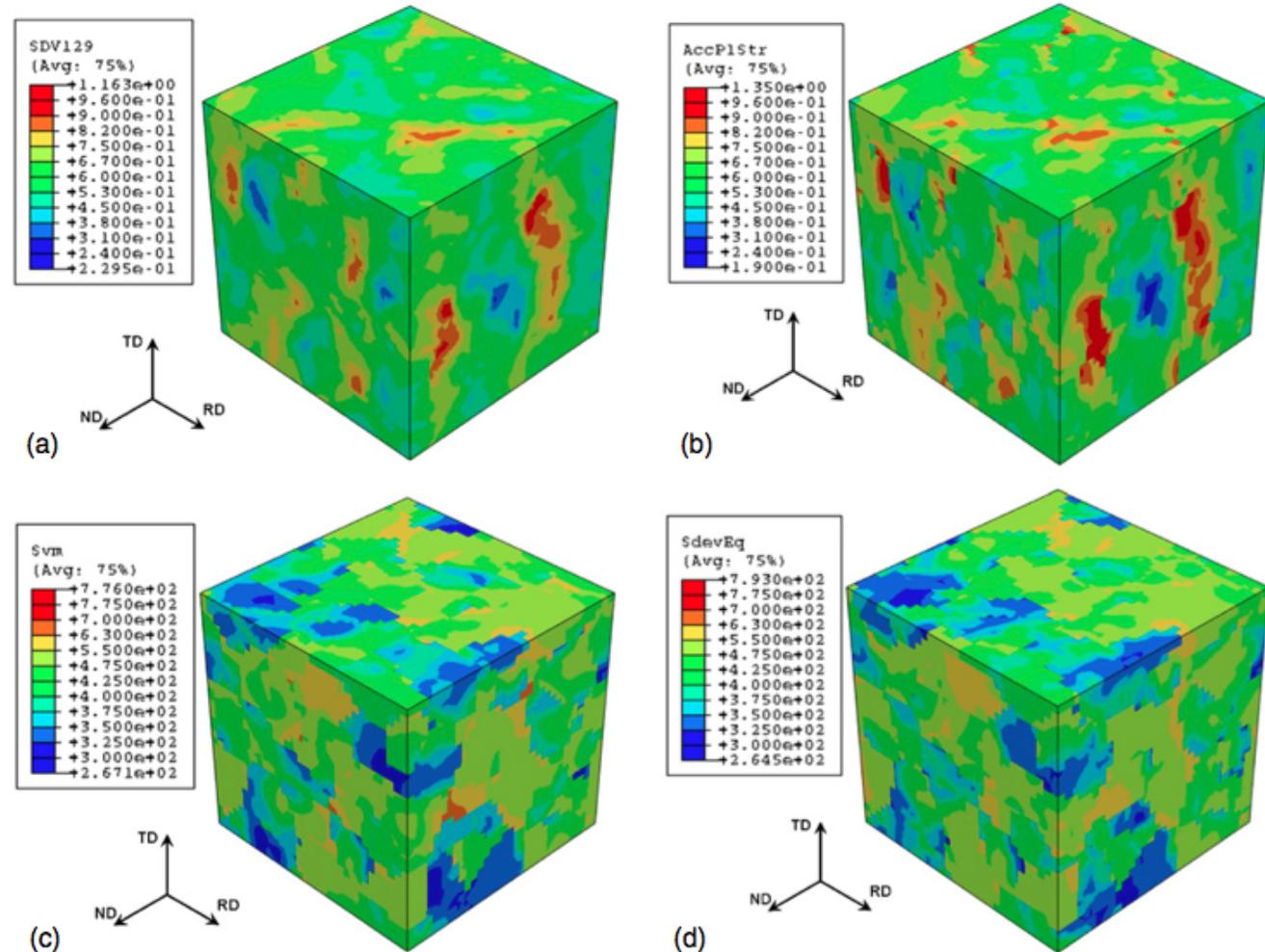


Figure 3. Deformed configuration of the RVE from rolling simulations after 40% thickness reduction; (a) FE and (b) FFT.



See also:
Eisenlohr *et al.*, IJP 46 37-53 (2013)

Figure 4. Results from rolling simulations; top row—distribution of accumulated plastic strain, bottom row—stress distribution; left—FE, right—FFT. Note that the scales used for comparison are the same; only the maximum and minimum values differ.

Inputs

- Any code that computes micro-mechanical fields needs to know:
 - what type of calculation (elastic, thermoelastic, viscoplastic ...)
 - boundary conditions (type of strain to be imposed, magnitude ...)
 - materials properties (elastic moduli, slip systems, twinning ...)
- The (FFT) image-based method, not surprisingly, needs an image of the material (as opposed to a mesh). Think of sampling the material on a regular grid (uniform point spacing). Each gridpoint can be a different material but, in practice, we are interested in gradients across bulk features (grains, particles, lamellae ...); therefore apply the rule of thumb that 10 points across a feature of interest.
- For the FFT codes, there are 3 input files:
 - i) the image (generally 1 line per gridpoint with orientation + grain ID + phase ID);
 - ii) the constitutive properties (e.g. *fcc.sx*);
 - iii) the control file (e.g. *options.in*, *fft.in*).

Outputs

- Any code that computes micro-mechanical fields outputs many different files, generally one (set of) file(s) per field, as well as average quantities over the simulation domain. Thus the FFT codes output the following (non exhaustive!):
 - stress+strain history
 - error/convergence history
 - orientation field (*tex* files); generally include von Mises σ & ε .
 - stress tensor field (*sfield* files)
 - strain tensor field (*dfield* files)
 - stress gradient, scalar
 - orientation gradient, scalar, kernel average misorientation
 - packaged sets of fields as *VTI* or *VTK* files for immediate viewing in *Paraview*

Post Simulation Analysis; 3D Viewing

- Many researchers write their own post-simulation analysis tools.
- *FFT2dx* is one such program that converts certain output files to Paraview inputs (below) and computes derivative fields (e.g. orientation gradients, principal stresses from eigenanalysis).
- There are many software for viewing objects, fields etc. in 3D
- Paraview is an open source package that is supported by the DOE Nuclear Defense Laboratories; it has proven to be sufficient for most of our needs.
- There is substantial scope for improving and streamlining the output. For example, output to a single HDF file with a helper file for Paraview.
- Avizo and Fiji are useful for analyzing Computed Tomography data, for example.

vpFFT/ pvpFFT/ teFFT/ pteFFT code and input

This guide steps through the files required in order to run the parallel version of Ricardo Lebensohn's code. Please note that you should not give the code to anyone else without first consulting with both myself and Ricardo Lebensohn since it is still a "research code" that is under development. Ricardo is willing to collaborate under any reasonable request but this code is not yet ready to be distributed in manner, for example, in which VPSC is distributed. No formal licensing statement is included at present although the code is registered at Los Alamos and has a release number.

0. The Fourier grid is a regular grid of points. The data files are always written with x varying fastest and z slowest. As mentioned above, the coordinates of the points are implicit. The index for each point is given as 3 integers (see above) so, in principle, the points could be listed in any order. A few of the subroutines check this but not all. In order to compute the real space coordinates, you *must* take account of the grid shape as given by the deformation gradient at the top of the input/output files. You will have to be particularly careful if you examine simulation results with shear deformations. Eventually we will implement a dual grid method in which the second grid tracks variable strain/displacement at each material point (and material properties are interpolated back to the Fourier grid, which must remain regular).

vpFFT input: 2

1. You need a copy of the code. The best way to get it is to get access to our repository at CMU and use “git” to clone a copy. Otherwise, ask me (Rollett) for a tarball. It comes with a Makefile, which has various possible compilation targets built into it. We have run it on a variety of Macs, linux systems and large supercomputers (notably hawk, blacklight and hopper).
2. You need an image of the microstructure that you want to use as the starting point or instantiation of the simulation. The dimensions of the grid (see the next section) must be a power of two* in all three directions in order to use discrete Fourier transforms in a direct fashion. The image file (ascii text) has one line per grid point with NO header line. The Euler angles (Bunge convention) are given, in degrees, in columns 1-3. The coordinates of each point (integers, not real space) are specified in columns 4 through 6. Column 7 is the grain number and column 8 is the phase number. There is no limit on the number of grains. The current limit on the number of phases is 3, but you could increase that by changing the size of certain arrays inside the code (check with me if uncertain).

*** The parallel version uses FFTW (as opposed to Numerical Recipes) and this package permits arbitrary dimensions.**

vpFFT input: 3

3. You will find a control file called options.in as part of the tarball.
[Some explanations provided for the entries in options.in]

vpFFT: options.in 1

```
# options.in

# parsing conventions:
# --[flag] [options]
# flag order does not matter
# conventional whitespace does not matter
# comments can begin with either # or !

# note : command line arguments supercede these settings

--numphases 2
--phase 0 Zrpl.sx Zrel.sx
--phase 1 dummy.sx dummy1.sx
#--phase 0 thermelast-ADR2.dat

# uncomment to enable
!--multiple_file_input image
--verbose
!--binary
!--time_stamp
!--no_directory_creation
--output_prefix Zr_100step-

# npts1 npts2 npts3 number of Fourier points (must be npts1*npts2*npts3)
#--dimensions 128 128 128 2097152
#--dimensions 64 64 64 262144
#--dimensions 32 32 16 16384
--dimensions 512 512 64 16777216
```

vpFFT: options.in 2

```

# Y1 Y2 Y3 (one per line)
--direct_space_base
1. 0. 0.
0. 1. 0.
0. 0. 4.
--microstructure_file ZrFFT_512_512_64.txt
#--microstructure_file micro32b.in
#--microstructure_file TXFFT.txt
#--microstructure_file sphtx.tex
#--microstructure_file out-64x64x64.tex
--rve_dimensions 1. 1. 4.
# iudot: flag for vel.grad. (1:unknown-1:known) DO NOT CHANGE
--iudot_boundary_condition
    0      1      1
    1      0      1
    1      1      1
# vel.grad
--udot_boundary_condition
    -0.5    0.    0.
    0.    -0.5    0.
    0.     0.    1.0
# DO NOT CHANGE
--cauchy_flag
    1      0      0
           1      0
           0
--cauchy_stress
    0.     0.     0.
           0.     0.
           0.

```

```

--eqincr 0.0001
# ictrl (1-6: strain comp, 0: VM eq, -1: tdot)
#--ictrl 0
--ictrl -1
#--thermctrl 3
#--deltat 1000

# INFORMATION ABOUT RUN CONDITIONS
--nsteps 100
--ithermo 0
--error 0.000000001
--itmax 150
--irecover 0 # read grain states from STRESS.IN
(1) or not (0)?
--isave 0 # write grain states in STRESS.OUT
(1) or not (0)?
--iupdate 1 # update tex & RVE dim (1) or not
(0)?
--iuphard 1
!--voronoi # turns voronoi initialization on/
off
!--voronoi_gridsize 64
!--voronoi_numgrains 25
!--voronoi_nonperiodic # default :
periodic
!--force_nonrandom_voronoi 55645 # each positive
number corresponds to
# a
deterministic voronoi structure

```

vpFFT: options.in 3

```
!--voronoi_cushion 32          # parallel: will search for seeds at most
                              #   this far from every local domain boundary
                              #   not set: use default
!--random_texture -1          # initialize texture (random mdf,odf)
                              #   each positive number corresponds to a
                              #   deterministic grain id - orientation mapping
                              #   negative: orientations chosen at random
!--list_texture texture.wts # initialize texture using grain id - orientation
                              #   mapping specified in this file (.wts format)
!--angles_in_degrees          # specify list_texture input units

# OUTPUT FLAGS
# Not output if commented.
# Negative : will output only after the last timestep.
--write_fields_files_every    -1 #..timesteps
--write_strs_strn_curve_every  1 #..timesteps
--write_statistics_file_every -1 #..timesteps
--write_tex_file_every        10 #..timesteps
--write_stress_strain_vti_every -1 #..timesteps
--write_stress_deriv_magn_vti_every 10 #..timesteps
```



These lines control which files are output, and the frequency with which they are written; a “-1” means only at the last step.

Constitutive Properties: elastic

In almost all circumstances, you should specify the full anisotropic stiffness modulus, as in “C”.

0

13929.	7082.	5765.	0.000	0.000	0.000	Ice (MPa)
7082.	13929.	5765.	0.000	0.000	0.000	
5765.	5765.	15010.	0.000	0.000	0.000	
0.000	0.000	0.000	3014.	0.000	0.000	
0.000	0.000	0.000	0.000	3014.	0.000	
0.000	0.000	0.000	0.000	0.000	3423.5	

1

9000. 0.33

ISO

YOUNG(MPa),NU (V+R/2)

Constitutive Properties: plastic

SLIP SYSTEMS FOR Ti CRYSTAL

HEXAGONAL icryst

1. 1. 1.58734 90. 90. 120. cdim(i),cang(i)
 6 nmodesx (total # of modes listed in the file)
 3 nmodes (# of modes to be used in the calculation)
 1 2 4 mode(i) active modes = prismatic, basal, pyramidal<c+a>

PRISMATIC <a>

1 3 10 1.0 0.0 0 modex,nsmx,nrsx,gamd0x,twshx,isectwx
 30.0 30.0 10.0 80.0 10.0 tau0xf,tau0xb,tau1x,thet0,thet1
 1.0 1.0 1.0 1.0 1.0 1.0 1.0 hselfx, hlatex(1,im),im=1,nmodes
 1 0 -1 0 -1 2 -1 0
 0 -1 1 0 2 -1 -1 0
 -1 1 0 0 -1 -1 2 0

BASAL <a>

2 3 10 1.0 0.0 0 modex,nsmx,nrsx,gamd0x,twshx,isectwx
 50.0 50.0 10.0 80.0 10.0 tau0xf,tau0xb,tau1x,thet0,thet1
 1.0 1.0 1.0 1.0 1.0 1.0 1.0 hselfx, hlatex(1,im),im=1,nmodes
 0 0 0 1 2 -1 -1 0
 0 0 0 1 -1 2 -1 0
 0 0 0 1 -1 -1 2 0

PYRAMIDAL <a>

3 6 10 1.0 0.0 0 modex,nsmx,nrsx,gamd0x,twshx,isectwx
 50.0 50.0 10.0 80.0 10.0 tau0xf,tau0xb,tau1x,thet0,thet1
 1.0 1.0 1.0 1.0 1.0 1.0 1.0 hselfx, hlatex(1,im),im=1,nmodes
 1 0 -1 1 -1 2 -1 0
 0 -1 1 1 2 -1 -1 0
 -1 1 0 1 -1 -1 2 0
 -1 0 1 1 -1 2 -1 0
 0 1 -1 1 2 -1 -1 0
 1 -1 0 1 1 1 -2 0

PYRAMIDAL <c+a>

4 12 10 1.0 0.0 0 modex,nsmx,nrsx,gamd0x,twshx,isectwx
 50.0 50.0 10.0 80.0 10.0 tau0xf,tau0xb,tau1x,thet0,thet1
 1.0 1.0 1.0 1.0 1.0 1.0 1.0 hselfx, hlatex(1,im),im=1,nmodes
 1 0 -1 1 -1 -1 2 3
 1 0 -1 1 -2 1 1 3
 0 -1 1 1 1 1 -2 3
 0 -1 1 1 -1 2 -1 3
 -1 1 0 1 2 -1 -1 3
 -1 1 0 1 1 -2 1 3
 -1 0 1 1 2 -1 -1 3
 -1 0 1 1 1 1 -2 3
 0 1 -1 1 -1 -1 2 3
 0 1 -1 1 1 -2 1 3
 1 -1 0 1 -2 1 1 3
 1 -1 0 1 -1 2 -1 3

TENSILE TWIN {10-12}

5 6 10 1.0 0.0 1 modex,nsmx,nrsx,gamd0x,twshx,isectwx/ twin
 shear = 0.167
 45.0 45.0 30.0 800.0 130.0 tau0xf,tau0xb,tau1x,thet0,thet1
 1.0 1.0 1.0 1.0 1.0 1.0 1.0 hselfx, hlatex(1,im),im=1,nmodes
 1 0 -1 2 -1 0 1 1
 0 1 -1 2 0 -1 1 1
 -1 1 0 2 1 -1 0 1
 -1 0 1 2 1 0 -1 1
 0 -1 1 2 0 1 -1 1
 1 -1 0 2 -1 1 0 1

COMPRESSIVE TWIN {11-22}

6 6 10 1.0 0.0 1 modex,nsmx,nrsx,gamd0x,twshx,isectwx/ twin
 shear = 0.225
 45.0 45.0 30.0 800.0 130.0 tau0xf,tau0xb,tau1x,thet0,thet1
 1.0 1.0 1.0 1.0 1.0 1.0 1.0 hselfx, hlatex(1,im),im=1,nmodes
 2 -1 -1 2 2 -1 -1 -3
 1 1 -2 2 1 1 -2 -3
 -1 2 -1 2 -1 2 -1 -3
 -2 1 1 2 -2 1 1 -3
 -1 -1 2 2 -1 -1 2 -3
 1 -2 1 2 1 -2 1 -3

Output of vpFFT

The following is a list of the output files from the viscoplastic FFT code and some idea of what each one contains.

1. At each step, an image of the grid is output, e.g. [name-tex-step#-processor#.txt](#). The “name” prefix indicates the name for the simulation that you assigned in [options.in](#). The next 4 digits are the increment or step number. The suffix “_out” is to indicate an output file. The content of the file is exactly like the input, except that there are four header lines to make it compatible with popLA texture software (and Carlos Tomé’s analysis programs). The header lines are like this:

```
Y1=    0.418    0.000    0.000    0.017
Y2=    0.000    1.000    0.000
Y3=    0.000    0.000    2.358
B      262144
```

The first 3 lines are the deformation gradient (grid shape), with the strain increment as the 4th number on line 1. By “deformation gradient” is meant the matrix “F” as used in solid mechanics that is equal to dX/dx (new coordinates versus original coordinates). In more prosaic terms, each term on the leading diagonal gives the stretch values. The 4th line has “B” for Bunge Euler angles, followed by the point count. The following lines are exactly as in the input file, except that the von Mises equivalent stress and strain are inserted as columns 5 and 6. Column 4 is redundant (inherited from older versions). This output file can be processed with [FFT2dx](#), which is a C program that performs several types of analysis on the output of the FFT code set. This latter program requires a separate guide (!) to understand, e.g. distance map calculations. The 1st 3 columns in the [tex](#) files contain the Euler angles for each point (as updated by the simulation) and the format is such that the file is equivalent to a “.WTS” format file that can be digested by popLA-related software, e.g. my [wts2pop.f](#) program. The latter program converts a list of orientations to pole figures, orientation distributions etc.

Output of vpFFT: 2

2. At the end of the simulation, there are files called *name-stress-step#-processor#.vti*, which contains an image of the von Mises equivalent stress in the simulation domain, and *name-strain_rate-step#-processor#.vti*, which has the strain rate. Optionally also *name-stress_deriv_magn-step#-processor#.vti*. These are collected together as sets (one folder per step) with a *pvti* file in the main directory; you can click on the *pvti* file to directly view the image in Paraview without any further work.

One problem is that if you started with a cube, for example, and simulated a tensile test, the resulting grid should be stretched to correspond to the tensile deformation. This is done by copying the (leading diagonal of the) deformation gradient information into header of the VTK file. Unfortunately, Paraview does not pay attention to the shape of the domain in the *pvti+vti* file set as of late 2014.

3. At the end of the simulation, there are files called *name-dfield-step#-proc#.output* and *name-sfield-step#-proc#.output*, which contain the full tensor strain-rate and stress fields, respectively. The “proc#” refers to parallel computations where each processor outputs its own section of the image (slab decomposition). These are just multi-column data (text) files and you cannot view them with Paraview directly. These can be converted to vtk files by using *cnvrt_field.f90*. In particular, if you have a series of dfield files (currently only from the serial *fft4* version), this program will compute the accumulated strain as *dfield-accum-strain.vtk*.

Output of vpFFT: 3

4. Also at the end, there are these files:

[conv_out.txt](#)

Just a list of the steps.

[err_out.txt](#)

This contains the error history, step by step.

[str_str_out.txt](#)

This contains the stress-strain history and you can use it to plot a stress-strain curve, e.g. to check the correctness of the hardening law.

5. In recent versions of the parallel version of the viscoplastic code, there are output files for the magnitude of the derivative of the stress such as [Stress-Deriv-Magn-0000000000.vtk](#), with numbers corresponding to the number of processors used. In more recent versions (as noted previously) there may be an additional number that specifies the strain step. You can assemble a complete image of the Stress derivative magnitude by executing the following commands. The initial “cp” command just makes a copy of the CPU 0 part of the file (which contains the header lines) and the following “cat” commands append each CPU’s portion of the image.

```
cp Stress-Deriv-Magn-0000000000.vtk Stress-Deriv-Magn.vtk
cat Stress-Deriv-Magn-0000000001.vtk >> Stress-Deriv-Magn.vtk
cat Stress-Deriv-Magn-0000000002.vtk >> Stress-Deriv-Magn.vtk
cat Stress-Deriv-Magn-0000000003.vtk >> Stress-Deriv-Magn.vtk
etc. etc. (for however many CPUs you ran on).
```

6. In some simulations, we use buffer layers because the specimen or test being simulated should not be treated as periodic because there are free surfaces. The simplest way to do this is to use multiple phases, where one assigns one phase (usually the highest number) to be the buffer layer (which has zero strength or stiffness). For example, for the bicrystal tension test simulation, phase 1 is the actual material and phase 2 is the buffer layer. When one examines the stress or strain-rate images in Paraview, it is usually a good idea to threshold by the phase number and retain only phase 1 (in this example).

$$(1) \quad \boldsymbol{\varepsilon}(\mathbf{x}) = \mathbf{C}^{-1}(\mathbf{x}) : \boldsymbol{\sigma}(\mathbf{x}) + \boldsymbol{\varepsilon}^*(\mathbf{x})$$

stiffness tensor of homogeneous solid

$$(2) \quad \boldsymbol{\sigma}(\mathbf{x}) = \boldsymbol{\sigma}(\mathbf{x}) + \mathbf{C}^0 : \boldsymbol{\varepsilon}(\mathbf{x}) - \mathbf{C}^0 : \boldsymbol{\varepsilon}(\mathbf{x})$$

$$\boldsymbol{\sigma}(\mathbf{x}) = \mathbf{C}^0 : \boldsymbol{\varepsilon}(\mathbf{x}) + (\boldsymbol{\sigma}(\mathbf{x}) - \mathbf{C}^0 : \boldsymbol{\varepsilon}(\mathbf{x}))$$

$$\boldsymbol{\sigma}(\mathbf{x}) = \mathbf{C}^0 : \boldsymbol{\varepsilon}(\mathbf{x}) + \boldsymbol{\tau}(\mathbf{x})$$

perturbation in stress field, associated with the heterogeneity in the elastic properties

$$(3) \quad \sigma_{ij,j} = 0$$

$$\mathbf{C}_{ijkl}^0 \mathbf{u}_{k,lj}(\mathbf{x}) + \tau_{ij,j}(\mathbf{x}) = 0$$

periodic boundary conditions in RVE

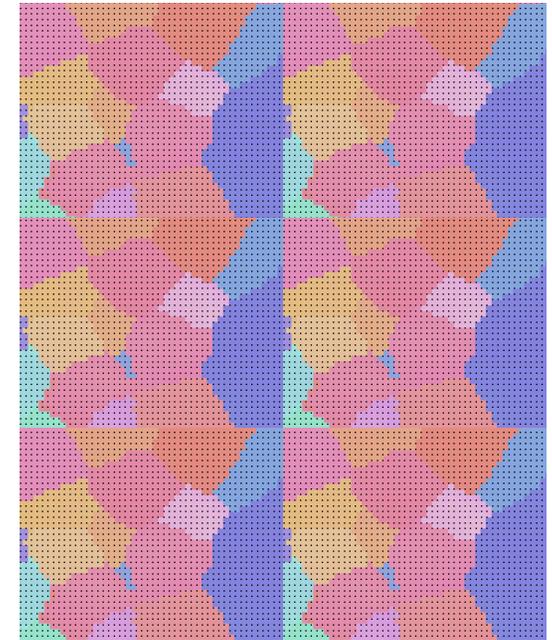
$$(4) \quad \mathbf{C}_{ijkl}^0 \mathbf{G}_{km,lj}(\mathbf{x} - \mathbf{x}') + \delta_{im} \delta(\mathbf{x} - \mathbf{x}') = 0$$

$$(5) \quad \tilde{\varepsilon}_{ij}(\mathbf{x}) = \text{sym} \left(\int_{\mathbb{R}^3} \mathbf{G}_{ik,jl}(\mathbf{x} - \mathbf{x}') \tau_{kl}(\mathbf{x}') d\mathbf{x}' \right) \Rightarrow \tilde{\varepsilon}_{ij} = \Gamma_{ijkl}^0 * \tau_{kl}$$

$$\Rightarrow \text{fft}(\varepsilon_{ij} = \Gamma_{ijkl}^0 * \tau_{kl}) \Rightarrow \hat{\varepsilon}_{ij} = \hat{\Gamma}_{ijkl}^0 : \hat{\tau}_{kl}$$

Notation

Strain: $\boldsymbol{\varepsilon}$
 Stress: $\boldsymbol{\sigma}$
 Stiffness: \mathbf{C}
 Perturbation Stress: $\boldsymbol{\tau}$
 Displacement: \mathbf{u}
 Green's function: \mathbf{G}
 Xformed Green's: Γ

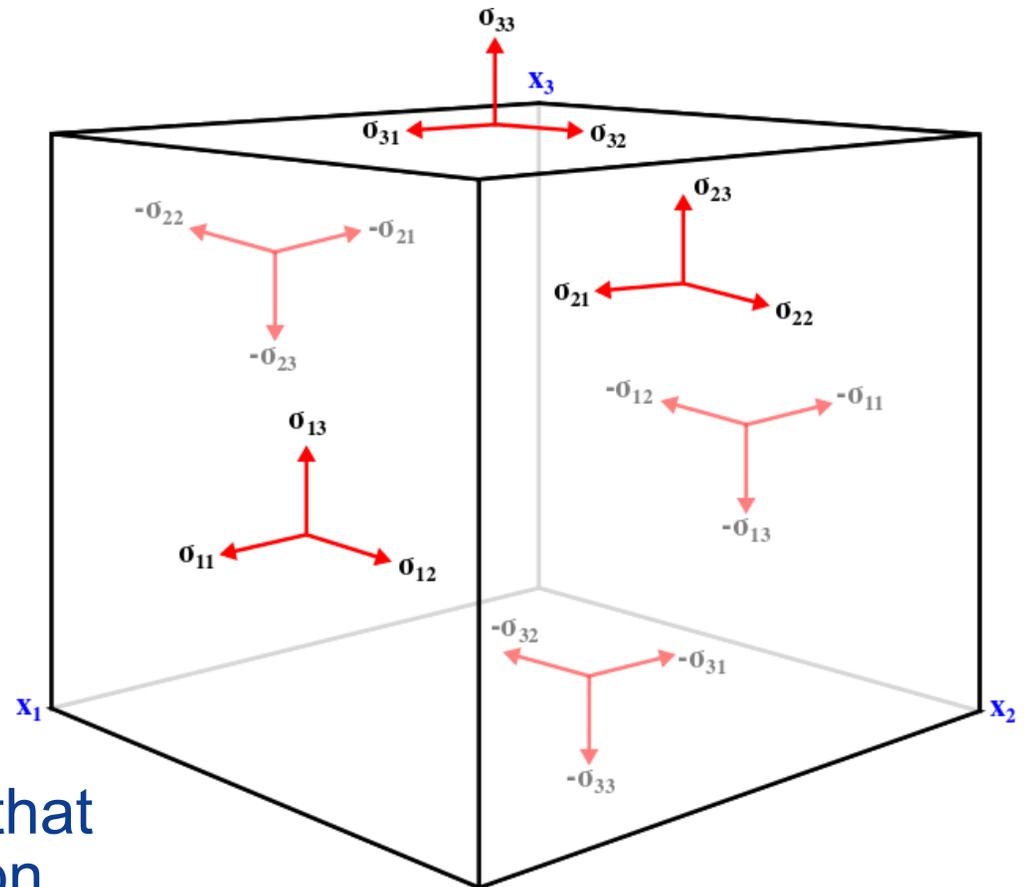


Γ obtained by solving Eq (4) in Fourier space, using Mura's approach

Stress Equilibrium

$$\sigma_{ij,j} = 0$$

$$\begin{cases} \frac{\delta\sigma_{xx}}{\delta x} + \frac{\delta\sigma_{xy}}{\delta y} + \frac{\delta\sigma_{xz}}{\delta z} = 0 \\ \frac{\delta\sigma_{yx}}{\delta x} + \frac{\delta\sigma_{yy}}{\delta y} + \frac{\delta\sigma_{yz}}{\delta z} = 0 \\ \frac{\delta\sigma_{zx}}{\delta x} + \frac{\delta\sigma_{zy}}{\delta y} + \frac{\delta\sigma_{zz}}{\delta z} = 0 \end{cases}$$



Stress equilibrium ensures that there is not net body force on the element.

Compatibility

For infinitesimal strains, compatibility is satisfied if the following equation holds:

$$\epsilon_{ij} = \frac{1}{2} (u_{i,j} + u_{j,i})$$

[largely from [en.wikipedia.org/wiki/Compatibility_\(mechanics\)](https://en.wikipedia.org/wiki/Compatibility_(mechanics))] Compatibility ensures that a unique strain (tensor) field is obtainable from a continuous, single-valued, displacement field. Conceptually, if a continuous body is thought to be divided into infinitesimal volumes, compatibility describes the necessary conditions under which the body deforms without developing gaps or overlaps between said volumes. In the context of infinitesimal strain theory, these conditions are equivalent to stating that the displacements in a body can be obtained by integrating the strains.

Thermoelastic FFT

Modified Hooke's Law (linear elasticity) with incorporated eigenstrains, ϵ^* , in reference to a homogeneous medium, where τ is a perturbation field associated with the heterogeneity in the elastic properties:

$$\sigma_{ij}(x) = C_{ijkl}^o : (\epsilon_{kl}(x) - \epsilon_{kl}^*(x)) + \tau_{ij}(x)$$

Application of stress equilibrium:

$$C_{ijkl}^o u_{k,lj}(x) + \tau_{ij,j}(x) = 0$$

Thermoelastic FFT

Application of Green's function, using Mura's approach:

$$C_{ijkl}^o G_{km,lj}(x - x') + \delta_{im} \delta(x - x') = 0$$

Application of Fourier Transform:

$$C_{ijkl}^o \xi_l \xi_j \hat{G}_{km} = \delta_{im}$$

Periodic Green's function in frequency space:

$$\hat{\Gamma}_{ijkl}^o = -(\xi_p \xi_q C_{ipkq}^o)^{-1} \xi_j \xi_l$$

Thermoelastic FFT

Application of periodic Green's function to perturbation in stress field:

$$\tilde{u}_k = \int_V G_{ki}(x - x') \tau_{ij,j}(x') dx'$$

Application of compatibility:

$$\epsilon_{ij}(x) = E_{ij} + \text{sym} \left(\int_V G_{ik,jl}(x - x') \tau_{kl}(x') dx' \right)$$

Augmented Lagrangian

Response equation at each iteration:

$$\frac{\delta w}{\delta e}(x, e^i) + C^o : e^i(x) = C^o : \epsilon^i(x) + \lambda^{i-1}(x)$$

Stress field at each iteration:

$$\lambda^i(x) = \lambda^{i-1}(x) + C^o : (\underbrace{\epsilon^i(x)}_{\text{Current strain}} - \underbrace{e^i(x)}_{\text{Trial strain}})$$

λ and e are auxiliary fields in the augmented Lagrangian method

Initializations for teFFT

$$\begin{aligned} E^0 &= \langle \epsilon^*(x) \rangle + (C^0)^{-1} : \Sigma \\ \lambda^0(x) &= C^0 : (E^0 - \epsilon^*(x)) \\ e^0(x) &= E^0 \end{aligned}$$

Note: it is straightforward to develop a field of eigenstrains to represent (displacive) phase transformation, twinning, dislocations etc.

teFFT Iteration

$$\tau^i(x) = \lambda^{i-1}(x) - C^o : e^{i-1}(x) + C(x) : \epsilon^*(x) \quad \mathbf{1.}$$

$$\hat{\tau}^i(\xi) = \text{fft}(\tau^i(x)) \quad \mathbf{2.}$$

$$\epsilon^i(x) = E^{i-1} + \text{sym} \left(\text{fft}^{-1} \left(\hat{\Gamma}^o : \hat{\tau}^i(\xi) \right) \right) \quad \mathbf{3.}$$

$$\sigma^i(x) + C^o : (C^{-1}(x) : \sigma^i(x) + \epsilon^*(x)) = \lambda^{i-1}(x) + C^o : \epsilon^i(x) \quad \mathbf{4.}$$

$$\sigma^i(x) = (I + C^o : C^{-1}(x)) [\lambda^{i-1}(x) + C^o : (\epsilon^i(x) - \epsilon^*(x))] \quad \mathbf{4.}$$

$$e^i(x) = C^{-1}(x) \sigma^i(x) + \epsilon^*(x) \quad \mathbf{5.}$$

$$\lambda^i(x) = \lambda^{i-1}(x) + C^o : (\epsilon^i(x) - e^i(x)) \quad \mathbf{6.}$$

$$E^i = \langle \epsilon^i(x) \rangle + C^{o^{-1}} : (\Sigma - \langle \sigma^i(x) \rangle) \quad \mathbf{7.}$$

FFT Errors, convergence

Stress field errors:

$$\text{err}[\lambda^i(x)] = \frac{\langle |||C^o : (\epsilon^i(x) - e^i(x))||| \rangle}{||\langle \sigma^i(x) \rangle||}$$

Strain field errors:

$$\text{err}[e^i(x)] = \frac{\langle |||\epsilon^i(x) - e^i(x)||| \rangle}{E}$$

Examples given later; in general, iteration is terminated after the change in the fields becomes small enough, or the error magnitude has dropped to small enough values.

S.P. Donegan
PhD thesis
(2013)

Derivation of the teFFT Method

The following derivation is adapted from Mura [85]. Mura's derivation solves for the exact stress and strain fields. The periodic Green's function in such a formulation contains a singularity at $\mathbf{x} = \mathbf{x}'$, and therefore no solution. To obtain a solution at this point, the approach of defining a homogeneous reference medium is utilized [49]. This derivation will be presented for the case of purely elastic strains. The derivation for the inclusion of eigenstrains is mathematically identical [85]. Begin with the constitutive equation relating stress and strain, which is Hooke's law:

$$\sigma_{ij} = C_{ijkl}\epsilon_{kl} \quad (\text{A.1})$$

where σ_{ij} is the stress tensor, ϵ_{kl} is the strain tensor, and C_{ijkl} is the elastic stiffness tensor. All tensor relations in equation A.1 can be defined as functions of position, \mathbf{x} . Incorporating the stress equilibrium condition from equation 2.13 leads to the following set of differential equations:

$$\begin{cases} \sigma(\mathbf{x}) = C(\mathbf{x}) : \epsilon(\mathbf{x}) \\ \sigma_{ij,j} = 0 \end{cases} \quad (\text{A.2})$$

where ‘:’ represents a double tensor contraction. The homogeneous reference medium can be defined as having prescribed stiffness coefficients C° , which are the point average of $C(\mathbf{x})$ over the entire domain:

$$C^\circ(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N C(\mathbf{x}) \quad (\text{A.3})$$

where N is the total number of points in the domain. Note that although C° is defined with respect to position in equation A.3, it is the same for all \mathbf{x} . Equations A.2 can now be rewritten with respect to the homogeneous reference medium:

$$\begin{cases} \sigma(\mathbf{x}) = C^\circ : \epsilon(\mathbf{x}) + \tau(\mathbf{x}) \\ \sigma_{ij,j} = 0 \end{cases} \quad (\text{A.4})$$

where $\tau(\mathbf{x})$ is the perturbation in the stress field:

$$\tau(\mathbf{x}) = (C(\mathbf{x}) - C^\circ) : \epsilon(\mathbf{x}) \quad (\text{A.5})$$

By considering the compatibility relationship in equation 2.14, the equations A.4 can be rewritten with respect to the displacement field:

$$C_{ijkl}^\circ u_{k,lj}(\mathbf{x}) + \tau_{ij,j}(\mathbf{x}) = 0 \quad (\text{A.6})$$

where u_k is the displacement vector along the x_k -direction. Substituting for $\tau(\mathbf{x})$ based on equation A.5 yields:

$$C_{ijkl}^\circ u_{k,lj}(\mathbf{x}) = (C_{ijkl}^\circ - C_{ijkl}(\mathbf{x})) \epsilon_{kl,j}(\mathbf{x}) \quad (\text{A.7})$$

Donegan, PhD thesis

Suppose $\epsilon_{ij}(\mathbf{x})$ is of the form of a wave of single amplitude $\bar{\epsilon}_{ij}(\boldsymbol{\xi})$:

$$\epsilon_{ij}(\mathbf{x}) = \bar{\epsilon}_{ij}(\boldsymbol{\xi}) \exp(i\boldsymbol{\xi} \cdot \mathbf{x}) \quad (\text{A.8})$$

where i is $\sqrt{-1}$ and $\boldsymbol{\xi}$ is a wave vector for a given period. The displacement can also be represented as a single wave of the same amplitude:

$$u_i(\mathbf{x}) = \bar{u}_i(\boldsymbol{\xi}) \exp(i\boldsymbol{\xi} \cdot \mathbf{x}) \quad (\text{A.9})$$

Equations A.8 and A.9 can be substituted into equation A.7:

$$C_{ijkl}^\circ \bar{u}_k \xi_l \xi_j = -i(C_{ijkl}^\circ - C_{ijkl}(\mathbf{x})) \bar{\epsilon}_{kl} \xi_j \quad (\text{A.10})$$

Equation A.10 arises from the fact that $(i\boldsymbol{\xi} \cdot \mathbf{x})_{,l} = i\xi_l$. For any given $\bar{\epsilon}_{ij}$ there are three unknown \bar{u}_i . To solve this system, the following simplifications are defined:

$$\begin{aligned} K_{ik}(\boldsymbol{\xi}) &= C_{ijkl}^\circ \xi_l \xi_j \\ X_i &= -i(C_{ijkl}^\circ - C_{ijkl}(\mathbf{x})) \bar{\epsilon}_{kl} \xi_j \end{aligned} \quad (\text{A.11})$$

Equations A.11 allow equation A.10 to be written as a system of equations:

$$\begin{aligned} K_{11}\bar{u}_1 + K_{12}\bar{u}_2 + K_{13}\bar{u}_3 &= X_1 \\ K_{21}\bar{u}_1 + K_{22}\bar{u}_2 + K_{23}\bar{u}_3 &= X_2 \\ K_{31}\bar{u}_1 + K_{32}\bar{u}_2 + K_{33}\bar{u}_3 &= X_3 \end{aligned} \quad (\text{A.12})$$

The amplitude of the displacement is now:

$$\bar{u}_i(\boldsymbol{\xi}) = \frac{X_j N_{ij}(\boldsymbol{\xi})}{D(\boldsymbol{\xi})} \quad (\text{A.13})$$

Donegan, PhD thesis

Noting that $\mathbf{K}(\boldsymbol{\xi})$ is the following matrix:

$$\mathbf{K}(\boldsymbol{\xi}) = \begin{pmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{pmatrix} \quad (\text{A.14})$$

then $\mathbf{N}(\boldsymbol{\xi})$ is the cofactor matrix of $\mathbf{K}(\boldsymbol{\xi})$:

$$\mathbf{N}(\boldsymbol{\xi}) = \begin{pmatrix} -K_{23}K_{32} + K_{22}K_{33} & K_{23}K_{31} - K_{21}K_{33} & -K_{22}K_{31} + K_{21}K_{32} \\ K_{13}K_{32} - K_{12}K_{33} & -K_{13}K_{31} + K_{11}K_{33} & K_{12}K_{31} - K_{11}K_{32} \\ -K_{13}K_{22} + K_{12}K_{23} & K_{13}K_{21} - K_{11}K_{23} & -K_{12}K_{21} + K_{11}K_{22} \end{pmatrix} \quad (\text{A.15})$$

and $D(\boldsymbol{\xi})$ is the determinant of $\mathbf{K}(\boldsymbol{\xi})$:

$$D(\boldsymbol{\xi}) = -K_{13}K_{22}K_{31} + K_{12}K_{23}K_{31} + K_{13}K_{21}K_{32} - K_{11}K_{23}K_{32} - K_{12}K_{21}K_{33} + K_{11}K_{22}K_{33} \quad (\text{A.16})$$

Substituting equation A.13 into equation A.9 yields the following:

$$u_i(\mathbf{x}) = \frac{X_j N_{ij}(\boldsymbol{\xi})}{D(\boldsymbol{\xi})} \exp(i\boldsymbol{\xi} \cdot \mathbf{x}) \quad (\text{A.17})$$

By the symmetry of C_{ijkl} , the following is true:

$$K_{ki} = C_{kji} \xi_j \xi_l = C_{klij} \xi_l \xi_j = C_{ijkl} \xi_l \xi_j = K_{ik} \quad (\text{A.18})$$

Expanding X_j in equation A.17 yields:

$$u_i(\mathbf{x}) = -i(C_{ijkl}^o - C_{ijkl}(\mathbf{x})) \bar{\epsilon}_{kl} \xi_j N_{ij}(\boldsymbol{\xi}) D^{-1}(\boldsymbol{\xi}) \exp(i\boldsymbol{\xi} \cdot \mathbf{x}) \quad (\text{A.19})$$

Donegan, PhD thesis

Linear elasticity allows for the superposition of solutions. Equation A.19 can be written as a summation over all frequencies:

$$u_i(\mathbf{x}) = -i \sum (C_{ijkl}^o - C_{ijkl}(\mathbf{x})) \bar{\epsilon}_{kl} \xi_j N_{ij}(\boldsymbol{\xi}) D^{-1}(\boldsymbol{\xi}) \exp(i\boldsymbol{\xi} \cdot \mathbf{x}) \quad (\text{A.20})$$

Substituting equation A.20 into the compatibility equations yields:

$$\begin{aligned} \epsilon_{ij}(\mathbf{x}) = & \frac{1}{2} \sum (C_{ijkl}^o - C_{ijkl}(\mathbf{x})) \bar{\epsilon}_{kl} \xi_j (\xi_l N_{ik}(\boldsymbol{\xi}) + \xi_j N_{jk}(\boldsymbol{\xi})) D^{-1}(\boldsymbol{\xi}) \\ & \times \exp(i\boldsymbol{\xi} \cdot \mathbf{x}) \end{aligned} \quad (\text{A.21})$$

$\epsilon_{ij}(\mathbf{x})$ can now be written as a Fourier integral:

$$\epsilon_{ij}(\mathbf{x}) = \int_{-\infty}^{\infty} \bar{\epsilon}_{ij}(\boldsymbol{\xi}) \exp(i\boldsymbol{\xi} \cdot \mathbf{x}) d\boldsymbol{\xi} \quad (\text{A.22})$$

where $\bar{\epsilon}_{ij}(\boldsymbol{\xi}) = \frac{1}{2\pi^3} \int_{-\infty}^{\infty} \epsilon_{ij}(\mathbf{x}) \exp(-i\boldsymbol{\xi} \cdot \mathbf{x}) d\mathbf{x}$. Substituting this representation of strain in frequency space into equation A.20 yields:

$$\begin{aligned} u_i(\mathbf{x}) = & -i \frac{1}{2\pi^3} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (C_{ijkl}^o - C_{ijkl}(\mathbf{x})) \epsilon_{kl}(\mathbf{x}') \xi_j N_{ij}(\boldsymbol{\xi}) D^{-1}(\boldsymbol{\xi}) \\ & \times \exp(i\boldsymbol{\xi} \cdot (\mathbf{x} - \mathbf{x}')) d\boldsymbol{\xi} d\mathbf{x}' \end{aligned} \quad (\text{A.23})$$

Now consider a Green's function of the following form:

$$G_{ij}(\mathbf{x} - \mathbf{x}') = \frac{1}{2\pi^3} \int_{-\infty}^{\infty} N_{ij}(\boldsymbol{\xi}) D^{-1}(\boldsymbol{\xi}) \exp(i\boldsymbol{\xi} \cdot (\mathbf{x} - \mathbf{x}')) d\boldsymbol{\xi} \quad (\text{A.24})$$

Substituting equation A.24 into equation A.23 yields:

$$u_i(\mathbf{x}) = - \int_{-\infty}^{\infty} (C_{ijkl}^o - C_{ijkl}(\mathbf{x})) \epsilon_{kl}(\mathbf{x}') G_{km,j}(\mathbf{x} - \mathbf{x}') d\mathbf{x}' \quad (\text{A.25})$$

Equation 2.17 can now be verified. First multiply the equation A.24 by the homogeneous stiffness tensor:

$$\begin{aligned} C_{ijkl}^{\circ} G_{km,lj}(\mathbf{x} - \mathbf{x}') &= \frac{1}{2\pi^3} \int_{-\infty}^{\infty} C_{ijkl}^{\circ} N_{km}(\boldsymbol{\xi}) D^{-1}(\boldsymbol{\xi}) \xi_i \xi_j \exp(i\boldsymbol{\xi} \cdot (\mathbf{x} - \mathbf{x}')) d\boldsymbol{\xi} \\ &= \frac{1}{2\pi^3} \int_{-\infty}^{\infty} K_{ik}(\boldsymbol{\xi}) N_{km}(\boldsymbol{\xi}) D^{-1}(\boldsymbol{\xi}) \exp(i\boldsymbol{\xi} \cdot (\mathbf{x} - \mathbf{x}')) d\boldsymbol{\xi} \end{aligned} \quad (\text{A.26})$$

Since $\mathbf{N}(\boldsymbol{\xi})$ is the cofactor matrix of $\mathbf{K}(\boldsymbol{\xi})$, the following is true:

$$K_{ik}(\boldsymbol{\xi}) N_{km}(\boldsymbol{\xi}) D^{-1}(\boldsymbol{\xi}) = \delta_{im} \quad (\text{A.27})$$

where δ_{im} is the Kronecker delta. The Dirac delta function, $\delta(\mathbf{x} - \mathbf{x}')$, can also be redefined:

$$\begin{aligned} \delta(\mathbf{x} - \mathbf{x}') &= \delta(x_1 - x'_1) \delta(x_2 - x'_2) \delta(x_3 - x'_3) \\ &= -\frac{1}{2\pi^3} \int_{-\infty}^{\infty} \exp(i\boldsymbol{\xi} \cdot (\mathbf{x} - \mathbf{x}')) d\boldsymbol{\xi} \end{aligned} \quad (\text{A.28})$$

Substituting equation A.28 and equation A.27 into equation A.26 yields equation 2.17. The solution to equation 2.17 is the convolution of the periodic Green's function with the stress perturbation fields, which yields equation 2.18. Use of compatibility yields equation 2.19. Setting $\Gamma_{ijkl} = \text{sym}(G_{ik,jl})$ simplifies equation 2.19:

$$\tilde{\epsilon}_{ij} = \Gamma_{ijkl} * \tau_{kl} \quad (\text{A.29})$$

In frequency space, equation A.29 appears as follows:

$$\hat{\tilde{\epsilon}}_{ij} = \hat{\Gamma}_{ijkl}^{\circ} : \hat{\tau}_{kl} \quad (\text{A.30})$$

Applying the Fourier transform to equation 2.17 yields equation 2.20. From the above definitions of $\mathbf{N}(\boldsymbol{\xi})$ and $\mathbf{D}(\boldsymbol{\xi})$, it follows that:

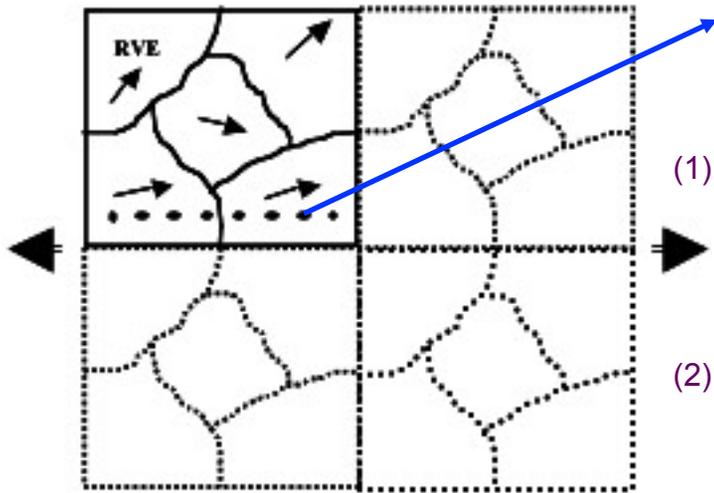
$$\hat{G}_{km} = N_{km} D^{-1} = A_{km}^{-1} \quad (\text{A.31})$$

The differentiated Green's function, $\hat{\Gamma}_{ijkl}^{\circ}$, can now be written in terms of $\mathbf{A}(\boldsymbol{\xi})$:

$$\hat{\Gamma}_{ijkl}^{\circ} = -\frac{1}{2}(\xi_l \xi_j A_{ik}^{-1} + \xi_l \xi_i A_{jk}^{-1}) \quad (\text{A.32})$$

Equation A.32 allows for a solution for the periodic Green's function in frequency space at all $\boldsymbol{\xi}$.

FFT Approach for Viscoplastic Polycrystals



Rate-sensitive approach ($n =$ Viscoplastic exponent)

$$(1) \quad \dot{\epsilon}(x) = \dot{\gamma}_o \sum_s m^s(x) \left(\frac{m^s(x) : \sigma'(x)}{\tau_o^s(x)} \right)^n$$

Schmid Tensor
Threshold Stress (Hardening of deformed system)

$$(2) \quad \begin{aligned} \sigma'(x) &= \sigma'(x) + L^o : \dot{\epsilon}(x) - L^o : \dot{\epsilon}(x) \\ &= L^o : \dot{\epsilon}(x) + \left(\sigma'(x) - L^o : \dot{\epsilon}(x) \right) \\ &= L^o : \dot{\epsilon}(x) + \tau(x) \end{aligned}$$

Stiffness of a Linear Reference Medium
Fluctuation (Heterogeneity Field) Function of Solution
→ Requires Iterative Procedure

$$(3) \quad \begin{cases} L_{ijkl}^o v_{k,lj}(x) + \tau_{ij,j}(x) - p_{,i}(x) = 0 & \text{in RVE} \\ v_{k,k}(x) = 0 & \text{in RVE} \\ \text{periodic boundary conditions across RVE} \end{cases}$$

Equilibrium + Incompressibility

$$\text{Slip Geometry: } m_{ij}^{(s)} = b_i^{(s)} n_j^{(s)} = \mathbf{b}^{(s)} \otimes \mathbf{n}^{(s)}$$

FFT code parallelized with FFTW

$$(4) \quad \begin{cases} L_{ijkl}^o G_{km,lj}(x-x') - H_{m,i}(x-x') + \delta_{im} \delta(x-x') = 0 \\ G_{km,k}(x-x') = 0 \end{cases}$$

Green's Function Method

Upon Convergence:

Velocity, Strain-Rate and Rotation-Rate Fields are obtained

→ **Morphology, Hardening and Texture Evolution**

$$(5) \quad \tilde{v}_{i,j}(x) = \text{sym} \left(\int_{R^3} G_{ik,jl}(x-x') \tau_{kl}(x') dx' \right)$$

$$\Rightarrow v = \Gamma^o * \tau \Rightarrow \hat{v} = \hat{\Gamma}^o : \hat{\tau} \Rightarrow \text{FFT}$$

R.A. Lebensohn, *Acta Materialia*, **49**, 2723-2737 (2001)

FFT solution - basics

- We use standard definitions of various quantities such as Cauchy stress (σ), deviatoric stress (σ'), velocity gradient, v , symmetric and skew-symmetric parts of strain rate, symmetric and skew-symmetric Schmid tensors.
- We distinguish between local values (lower case) and RVE average values (upper case).

$N_1 \times N_2 \times N_3$ Fourier grid points

$$\dot{E}_{ij} = D_{ij} = \frac{1}{2} (V_{i,j} + V_{j,i})$$

$$\dot{\Omega}_{ij} = \frac{1}{2} (V_{i,j} - V_{j,i})$$

$$\tilde{\epsilon}_{ij}(x) = (\tilde{u}_{i,j}(x) + \tilde{u}_{j,i}(x))/2$$

$$v_i(\mathbf{x}) = \dot{E}_{ij} x_j + \tilde{v}_i(\mathbf{x}) = D_{ij} x_j + \tilde{v}_i(\mathbf{x})$$

$$\dot{\epsilon}_{ij}(v_k(\mathbf{x})) = \dot{E}_{ij} + \tilde{\epsilon}_{ij}(\tilde{v}_k(\mathbf{x})) = D_{ij} + \tilde{\epsilon}_{ij}(\tilde{v}_k(\mathbf{x}))$$

$$m^s(\mathbf{x}) = \frac{1}{2} (n^s(\mathbf{x}) \otimes b^s(\mathbf{x}) + b^s(\mathbf{x}) \otimes n^s(\mathbf{x}))$$

$$\alpha^s(\mathbf{x}) = \frac{1}{2} (n^s(\mathbf{x}) \otimes b^s(\mathbf{x}) - b^s(\mathbf{x}) \otimes n^s(\mathbf{x}))$$

$$\dot{\gamma} = \dot{\gamma}_o ([m^s : \sigma'] / \tau_o^s)^n$$

FFT method for viscoplasticity

- Given a heterogeneous RVE with periodic boundary conditions, the local solutions for stress and strain-rate for each Fourier point can be obtained from the requirement for stress equilibrium (left) and incompressibility (right) as follows:

$$L_{ijkl}^o \dot{u}_{k,lj}(x) + \tau_{ij,j}(x) - p_{,i}(x) = 0 \quad \text{and} \quad \dot{u}_{k,k}(x) = 0$$

- Here $u\text{-dot}_k$ is the vector velocity field (varies with position x); differentiating this w.r.t. the l^{th} -direction, $u\text{-dot}_{k,l}$ is the gradient in the velocity field in that direction, $p(x)$ is the hydrostatic pressure field (+ve = compression) and $\tau_{ij}(x)$ is a perturbation field. The key point of the approach is to introduce a homogeneous reference medium of stiffness, L_{ijkl}^o , such that $\tau_{ij}(x)$ is derived from:

$$\begin{aligned} \sigma'(x) &= \sigma'(x) + L^o : \dot{\epsilon}(x) - L^o : \dot{\epsilon}(x) \\ &= L^o : \dot{\epsilon}(x) + \left(\sigma'(x) - L^o : \dot{\epsilon}(x) \right) \\ &= L^o : \dot{\epsilon}(x) + \tau(x) \end{aligned}$$

FFT method – Green's function

- The Green's function method can be used to solve the system of differential equations for stress equilibrium. The local fluctuations in velocity, velocity-gradient and strain-rate fields can, accordingly, be expressed as convolutions in the real space (note the different order of differentiation on the two Green's functions, G):

$$\tilde{u}_k(x) = \int_{R^3} G_{ki,j}(x-x') \tau_{ij}(x') dx' \quad \text{and} \quad \tilde{u}_{i,j}(x) = \int_{R^3} G_{ik,jl}(x-x') \tau_{kl}(x') dx'$$

- Since a convolution integral in real space can be expressed as a product in the Fourier space, we have this tensorial product:

$$\hat{\hat{\epsilon}}_{ij}(\xi) = \hat{\Gamma}_{ijkl}(\xi) \hat{\tau}_{kl}(\xi)$$

- where " $\hat{\hat{\epsilon}}$ " denotes the Fourier transform of the corresponding tensors.

$$C_{ijkl}^o \xi_l \xi_j \hat{G}_{km} = \delta_{im} \quad \Gamma_{ijkl} = \text{sym}(G_{ik,jl})$$

- Note that the displacements are also useful when one needs to update a material grid, for example.

FFT method - iterative solution

- If $\tau_{ij}(x)$ is known, $\tau\text{-hat}_{ij}(\xi)$ can readily be obtained by means of the FFT algorithm and the local fluctuation in the strain-rate field can be calculated from the Equations already given.
- However, the perturbation field can only be determined if the local strain-rate in Eq. (3) is known. This requires the implementation of an iterative method, explained elsewhere, with an initially specified strain-rate field, such that the deviatoric stress at each point can be obtained by solving:

$$\dot{\epsilon}(x) = \dot{\gamma}_o \sum_s m^s(x) ([m^s(x) : \sigma'(x)] / \tau_o^s(x))^n \quad (1)$$

- Here m^s is the Schmid tensor for the s^{th} slip system s , τ_o^s is the corresponding critical resolved stress, n is the rate-sensitivity exponent (typically between 10 and 60) and $\dot{\gamma}_o$ is a normalization factor (reference shear rate). For each iteration, the (implicit) equation above provides a new value of the deviatoric stress, σ' , which is then inserted into the previous equation to obtain a new value of strain rate.

Iteration: Augmented Lagrangians

- Iteration steps involve ...

- Multiply Γ by transformed σ to get d in Fourier space

$$\hat{d}_{ij}^{i+1}(\xi^{\mathbf{d}}) = -\hat{\Gamma}_{ijkl}^{sym}(\xi^{\mathbf{d}}) \hat{\varphi}_{kl}^i(\xi^{\mathbf{d}}),$$

$$\nabla_{\xi^{\mathbf{d}}} \neq 0; \quad \text{and} \quad \hat{d}_{ij}^{i+1}(\mathbf{0}) = 0$$

- The strain rate in real space is just the inverse FFT of the Fourier space d .

$$\tilde{d}_{ij}^{i+1}(\mathbf{x}^{\mathbf{d}}) = \text{fft}^{-1} \left\{ \hat{d}_{ij}^{i+1}(\xi^{\mathbf{d}}) \right\}$$

- The strain rate at each point is, however, also solved via the rate-sensitive relationship

$$\begin{aligned} \sigma'^{i+1}(\mathbf{x}^{\mathbf{d}}) + L^o : \dot{\gamma}_o \sum_{s=1}^{N_s} m^s(\mathbf{x}^{\mathbf{d}}) \left(\frac{m^s(\mathbf{x}^{\mathbf{d}}) : \sigma'^{i+1}(\mathbf{x}^{\mathbf{d}})}{\tau^s(\mathbf{x}^{\mathbf{d}})} \right)^n \\ = \lambda^i(\mathbf{x}^{\mathbf{d}}) + L^o : (\dot{E} + \tilde{d}^{i+1}(\mathbf{x}^{\mathbf{d}})) \end{aligned}$$

- The last step in an iteration is to get the estimate of the next set of Lagrange multipliers.

$$\lambda^{i+1}(\mathbf{x}^{\mathbf{d}}) = \lambda^i(\mathbf{x}^{\mathbf{d}}) + L^o : (\tilde{\varepsilon}^{i+1}(\mathbf{x}^{\mathbf{d}}) - \tilde{d}^{i+1}(\mathbf{x}^{\mathbf{d}}))$$

Michel, J. C., H. Moulinec *et al.* (2000). "A computational method based on augmented Lagrangians and fast Fourier Transforms for composites with high contrast." *CMES - Computer Modeling in Engineering & Sciences* 1(2): 79-88

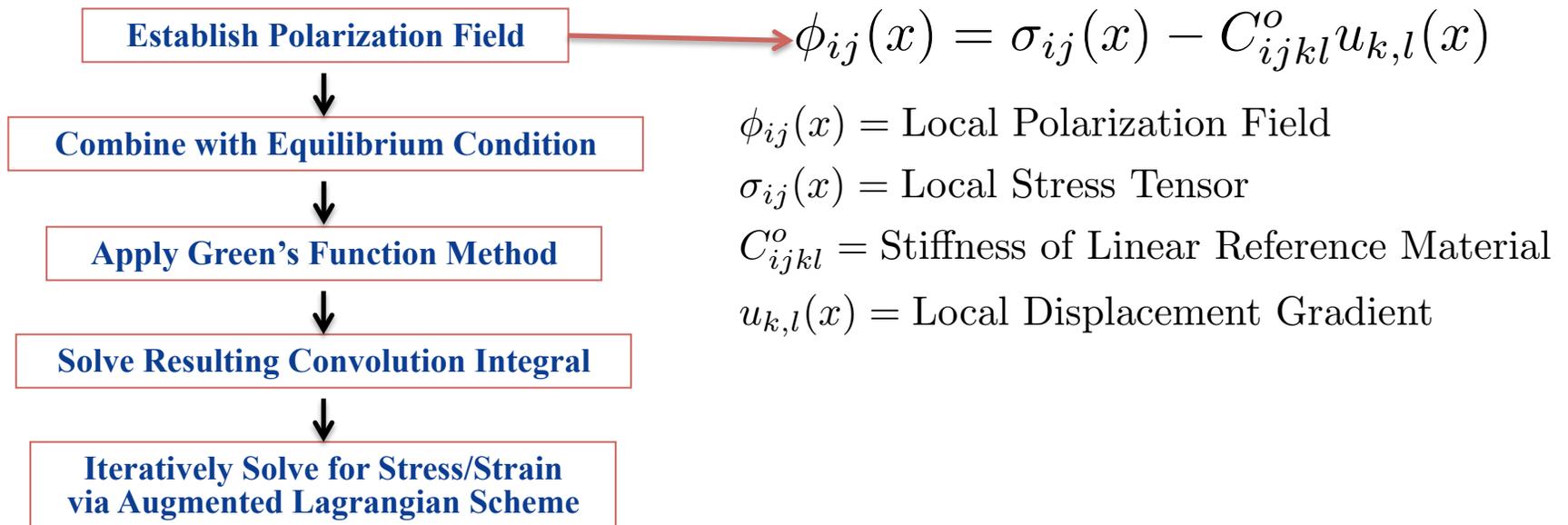
Elasto-Viscoplastic FFT Model

Constitutive Equations

$$\boldsymbol{\epsilon}^{total}(x) = \boldsymbol{\epsilon}^e(x) + \boldsymbol{\epsilon}^p(x) = \mathbf{C}^{-1}(x) : \boldsymbol{\sigma}(x) + \boldsymbol{\epsilon}^{p,t}(x) + \dot{\boldsymbol{\epsilon}}^p(x, \boldsymbol{\sigma}) \Delta t$$

$$\dot{\boldsymbol{\epsilon}}^p(x) = \dot{\gamma}_0 \sum_{s=1}^{N_s} \mathbf{m}^s(x) \left(\frac{|\mathbf{m}^s(x) : \boldsymbol{\sigma}'(x)|}{\tau^s(x)} \right)^n \times \text{sgn}(\mathbf{m}^s(x) : \boldsymbol{\sigma}'(x))$$

\mathbf{m}^s = Schmid Tensor $\dot{\gamma}_0$ = Reference Shear Rate $\boldsymbol{\sigma}'$ = Deviatoric Stress τ^s = CRSS \mathbf{C} = Stiffness Tensor



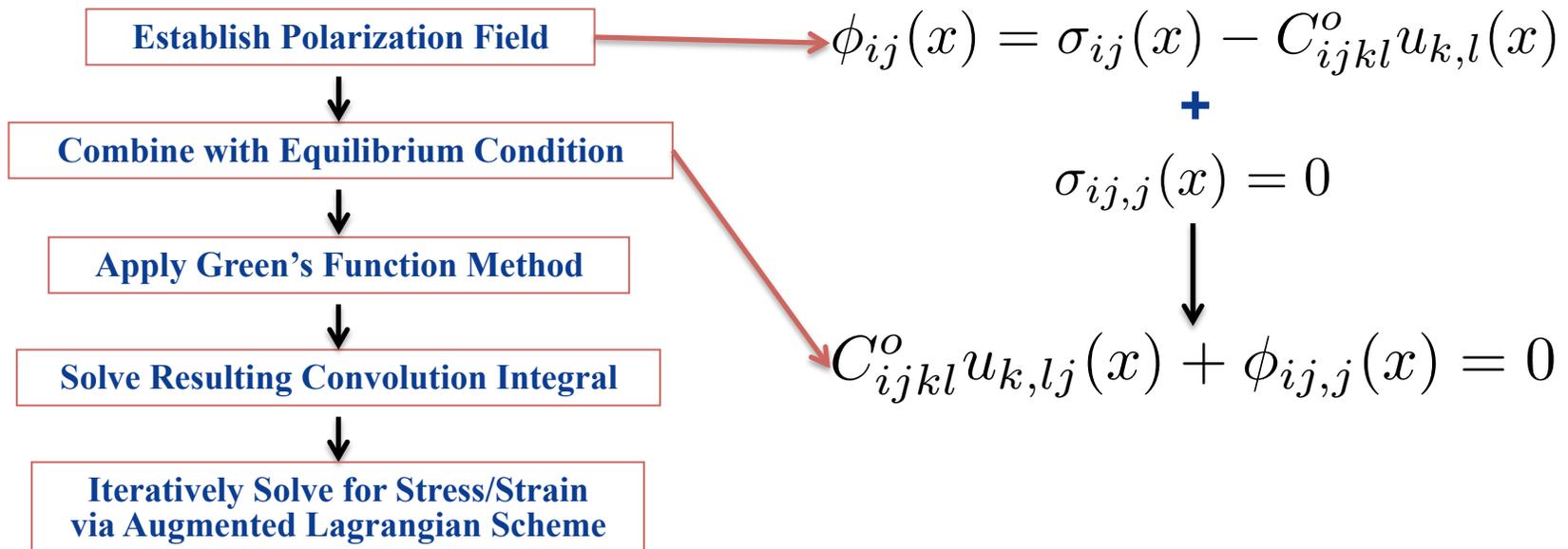
Elasto-Viscoplastic FFT Model

Constitutive Equations

$$\boldsymbol{\epsilon}^{total}(x) = \boldsymbol{\epsilon}^e(x) + \boldsymbol{\epsilon}^p(x) = \mathbf{C}^{-1}(x) : \boldsymbol{\sigma}(x) + \boldsymbol{\epsilon}^{p,t}(x) + \dot{\boldsymbol{\epsilon}}^p(x, \boldsymbol{\sigma}) \Delta t$$

$$\dot{\boldsymbol{\epsilon}}^p(x) = \dot{\gamma}_0 \sum_{s=1}^{N_s} \mathbf{m}^s(x) \left(\frac{|\mathbf{m}^s(x) : \boldsymbol{\sigma}'(x)|}{\tau^s(x)} \right)^n \times \text{sgn}(\mathbf{m}^s(x) : \boldsymbol{\sigma}'(x))$$

\mathbf{m}^s = Schmid Tensor $\dot{\gamma}_0$ = Reference Shear Rate $\boldsymbol{\sigma}'$ = Deviatoric Stress τ^s = CRSS \mathbf{C} = Stiffness Tensor



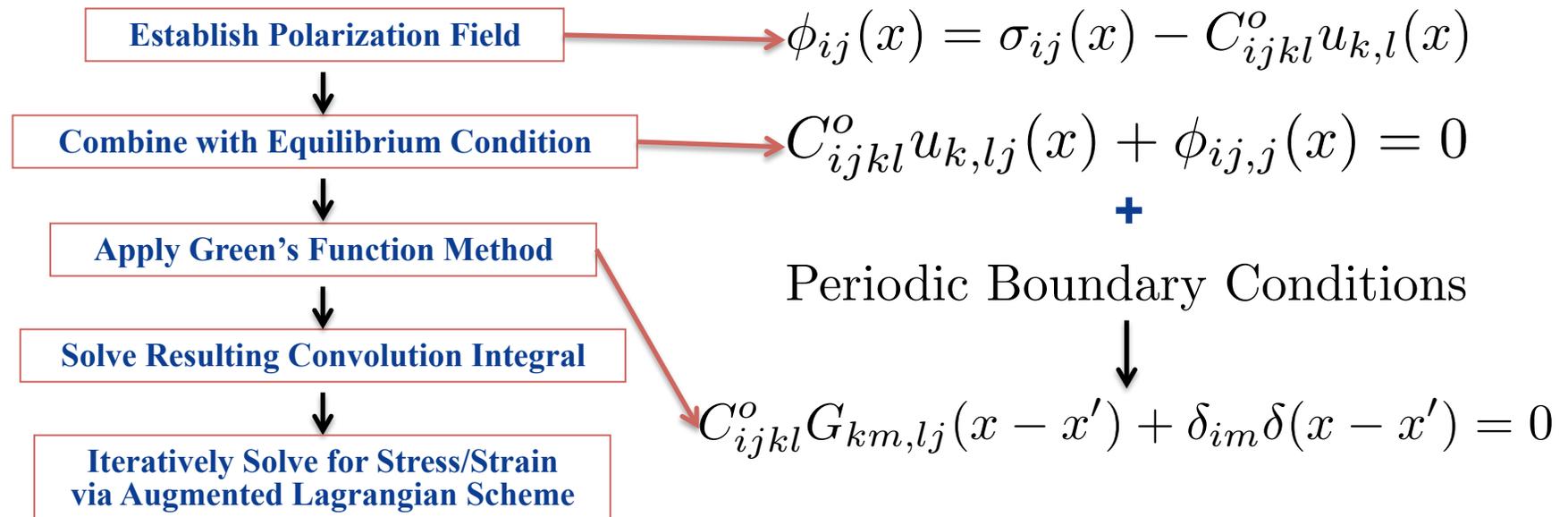
Elasto-Viscoplastic FFT Model

Constitutive Equations

$$\boldsymbol{\epsilon}^{total}(x) = \boldsymbol{\epsilon}^e(x) + \boldsymbol{\epsilon}^p(x) = \mathbf{C}^{-1}(x) : \boldsymbol{\sigma}(x) + \boldsymbol{\epsilon}^{p,t}(x) + \dot{\boldsymbol{\epsilon}}^p(x, \boldsymbol{\sigma}) \Delta t$$

$$\dot{\boldsymbol{\epsilon}}^p(x) = \dot{\gamma}_0 \sum_{s=1}^{N_s} \mathbf{m}^s(x) \left(\frac{|\mathbf{m}^s(x) : \boldsymbol{\sigma}'(x)|}{\tau^s(x)} \right)^n \times \text{sgn}(\mathbf{m}^s(x) : \boldsymbol{\sigma}'(x))$$

\mathbf{m}^s = Schmid Tensor $\dot{\gamma}_0$ = Reference Shear Rate $\boldsymbol{\sigma}'$ = Deviatoric Stress τ^s = CRSS \mathbf{C} = Stiffness Tensor



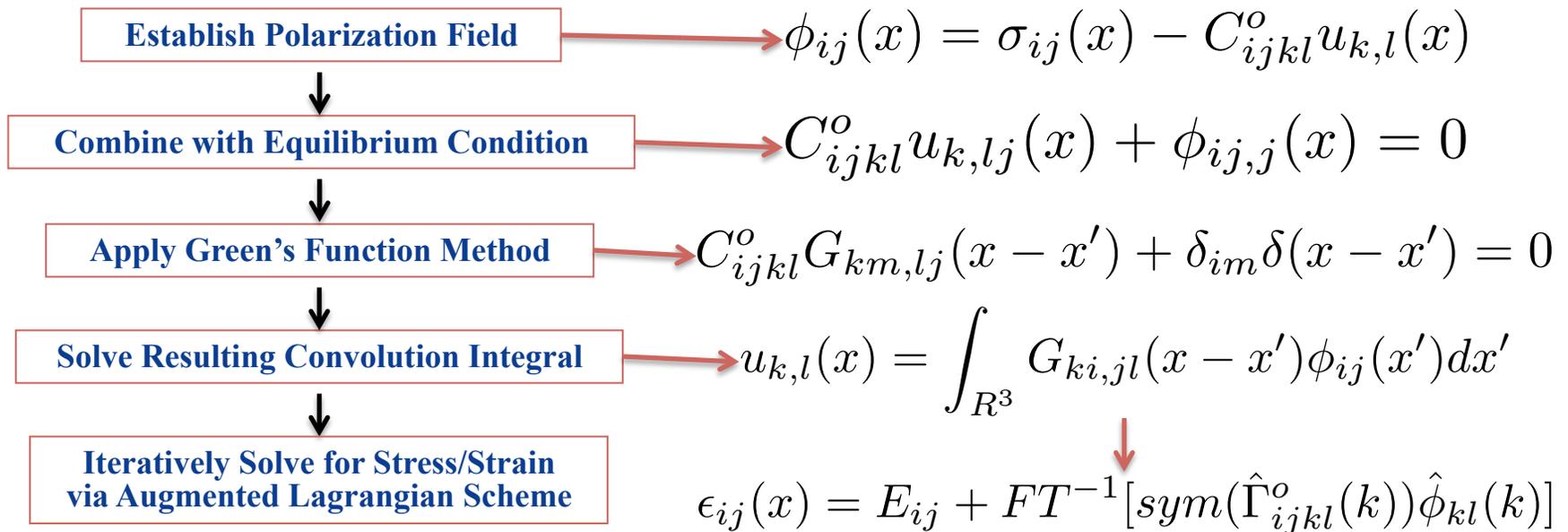
Elasto-Viscoplastic FFT Model

Constitutive Equations

$$\boldsymbol{\epsilon}^{total}(x) = \boldsymbol{\epsilon}^e(x) + \boldsymbol{\epsilon}^p(x) = \mathbf{C}^{-1}(x) : \boldsymbol{\sigma}(x) + \boldsymbol{\epsilon}^{p,t}(x) + \dot{\boldsymbol{\epsilon}}^p(x, \boldsymbol{\sigma}) \Delta t$$

$$\dot{\boldsymbol{\epsilon}}^p(x) = \dot{\gamma}_0 \sum_{s=1}^{N_s} \mathbf{m}^s(x) \left(\frac{|\mathbf{m}^s(x) : \boldsymbol{\sigma}'(x)|}{\tau^s(x)} \right)^n \times \text{sgn}(\mathbf{m}^s(x) : \boldsymbol{\sigma}'(x))$$

\mathbf{m}^s = Schmid Tensor $\dot{\gamma}_0$ = Reference Shear Rate $\boldsymbol{\sigma}'$ = Deviatoric Stress τ^s = CRSS \mathbf{C} = Stiffness Tensor



Examples of Applications

- Application examples of the FFT-based method are intended to show the reader how to extract micromechanical fields from microstructures.

Example: Dislocation Stress Fields

Why use FFT?

Current Dislocation Dynamics codes compute stress fields based on formulae from Hirth & Lothe. Anisotropic elasticity is considered to be expensive.

Expected Result?

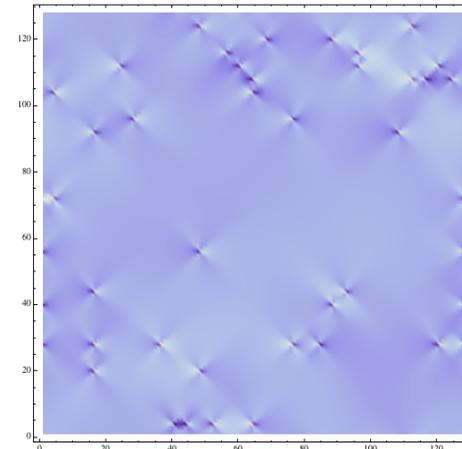
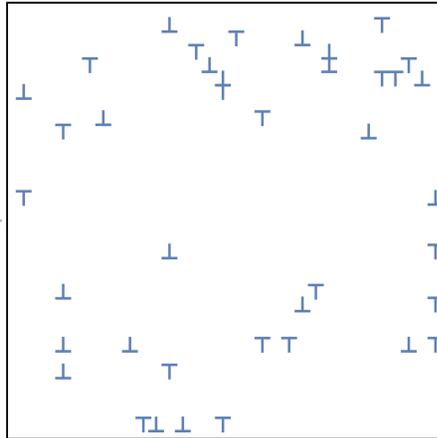
The expected result is that using the FFT method will return accurate stress fields and that calculations with anisotropic elasticity will be no more expensive than for isotropic.

How was it done?

In 2D, dislocations are modeled as units of eigenstrain at gridpoints. The computation is thermoelastic.

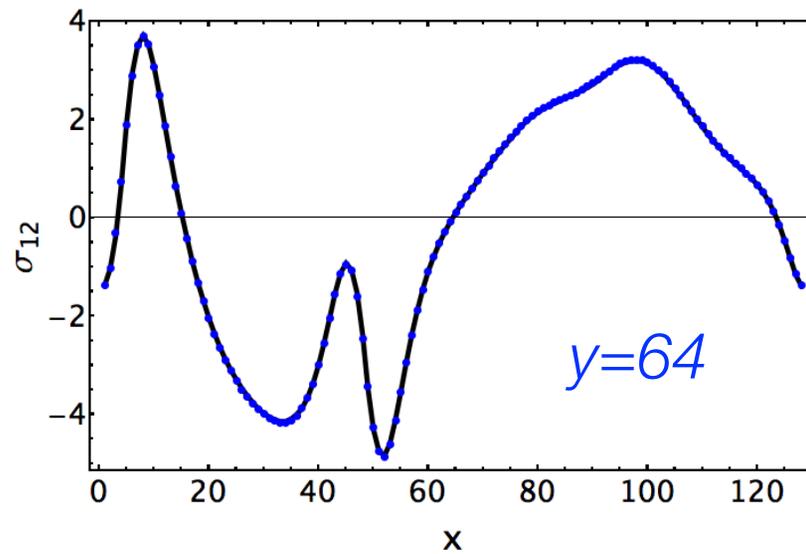
PhD by John Chapman advised by R.A. LeSar & A.D. Rollett; NSF support

Dislocation Stress Fields

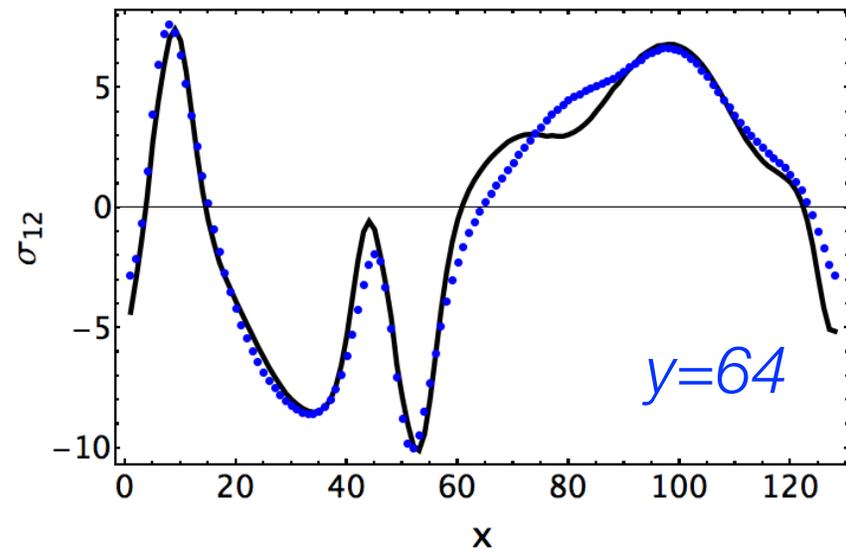
 β_{21}
 σ_{21}


medium gray, 0; white, 1;
dark gray, -1; black, -2

isotropic



anisotropic



Example: Fatigue Cracks

Why use FFT?

We observe cracks on the surface of a specimen. The micromechanical fields allow us to derive the *resolved shear stresses* on specific slip systems.

Expected Result?

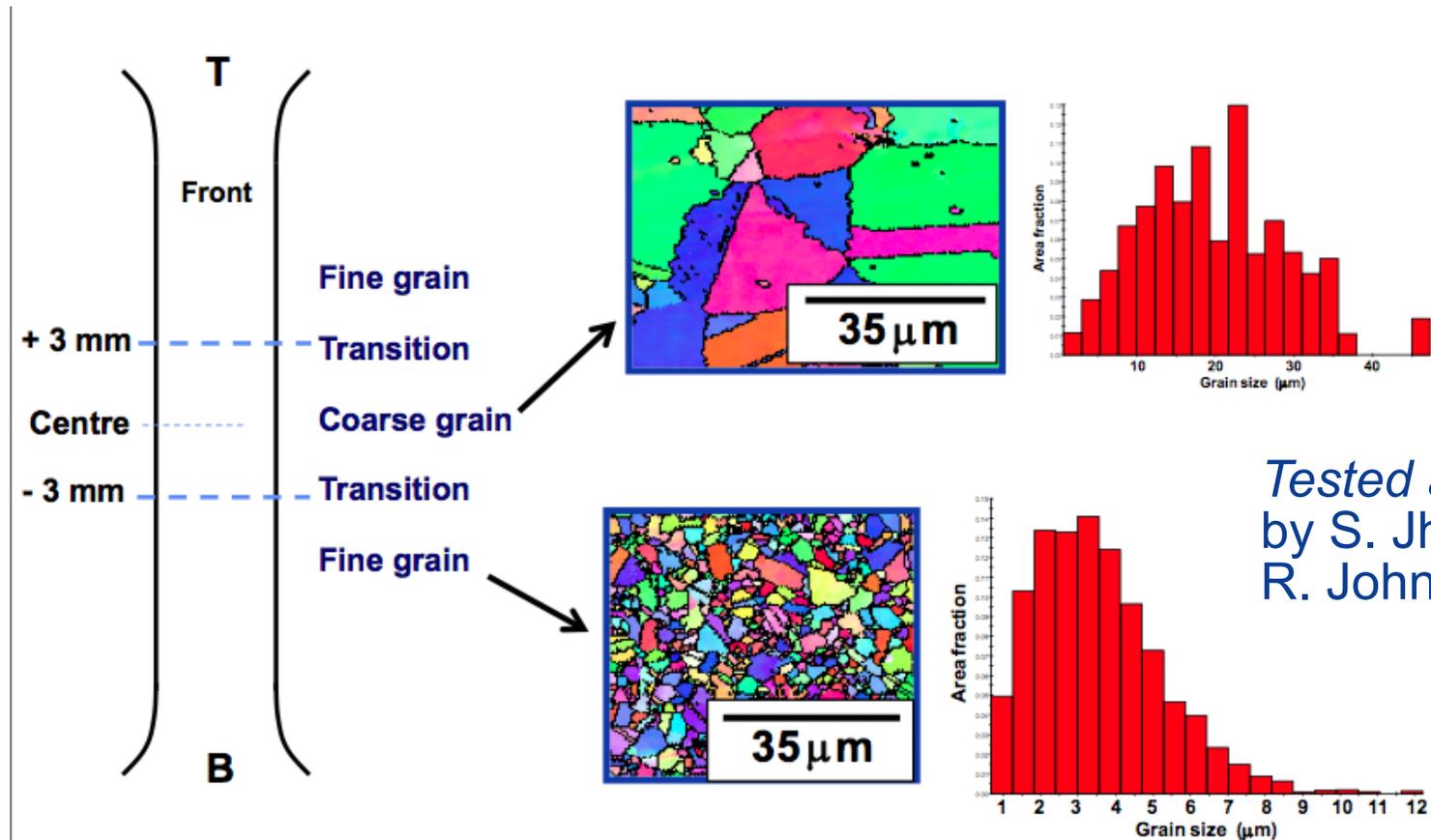
The expected result is that the grains with cracks would show high resolved shear stresses as a result of the elastic anisotropy of the material.

How was it done?

It is simple to re-format an EBSD map; mainly one has to add a grain_ID and a phase_ID.

PhD by Clay Stein; AFOSR support

LSHR: Graded Microstructure

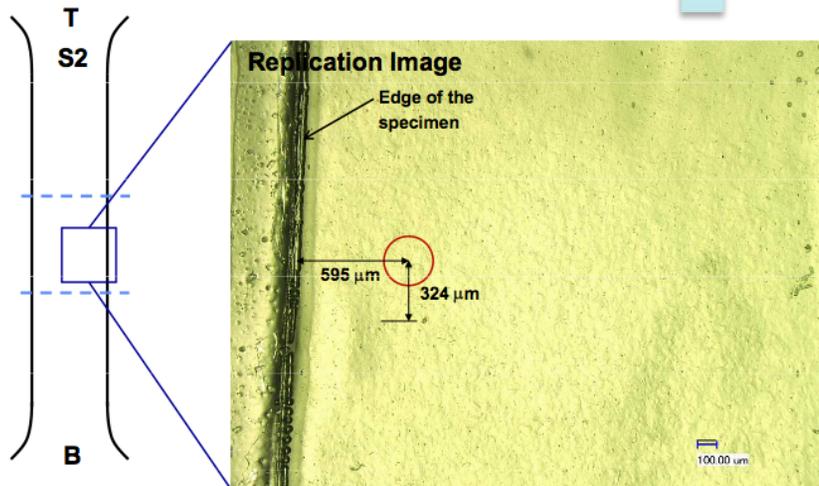
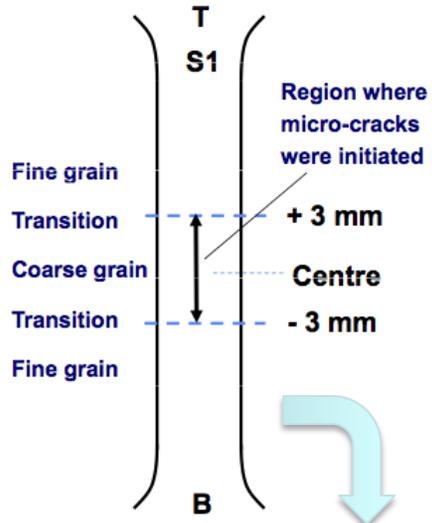


Specimen = 11-488
 Stress max. = 1050 MPa
 Stress ratio, $R = 0.05$
 Frequency, $\nu = 10$ Hz

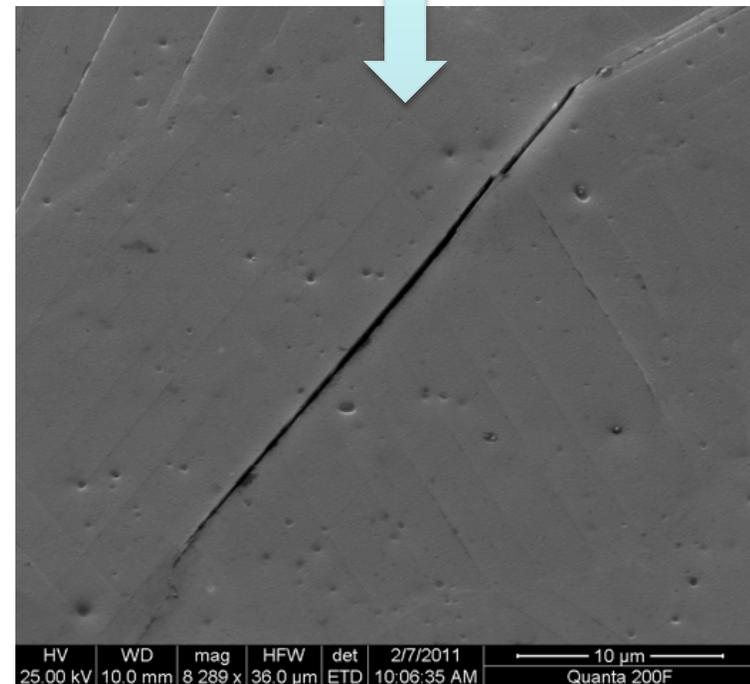
Loading = Axial; Sinusoidal
 T = Room temperature
 Specimen surface = Electropolished
 Small-crack monitoring method = Replication

Nominal: wt.%: 3.5Al, 0.03 B, 0.03C, 20.7Co, 12.5Cr, 2.7Mo, 1.5Nb, 1.6Ta, 3.5Ti, 4.3W, 0.05Zr, bal Ni.

LSHR: Microcracks

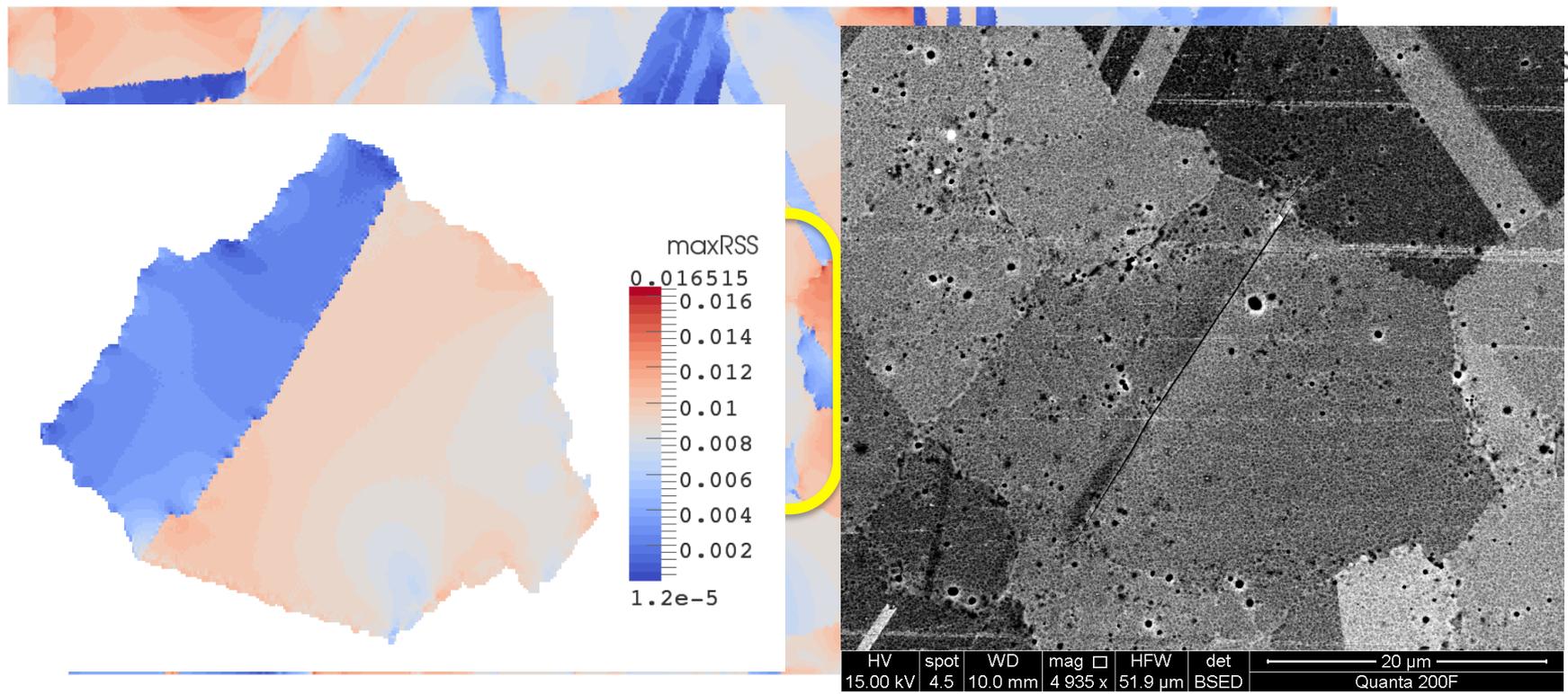


Grain size is of order 20 μm . In a volume 6 x 3 x 1 mm, there are of order 2 million grains. Approximately 10 (possible) microcracks observed. Ratio ≈ 1 in 10^6



Example of micro-crack and RSS

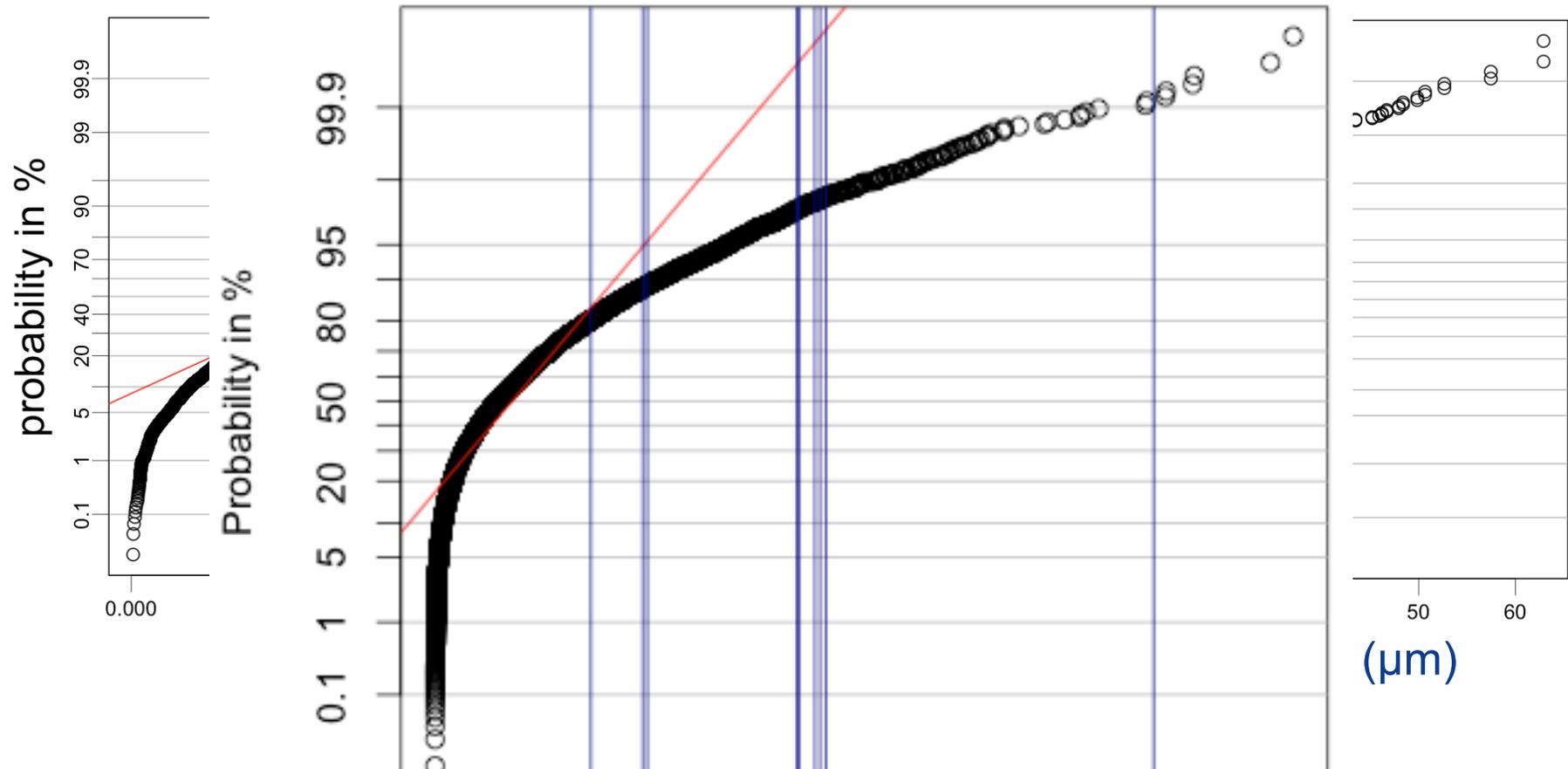
Full field stress resolved onto available slip systems at each point (2D calculation, based on EBSD map). Note alternation in stress across twin boundaries, as noted by Neumann *et al.* Maximum resolved shear stress measured. Similar to Schmid factor but accounts for anisotropy, neighbor interactions etc.



Fatigue crack initiation, slip localization and twin boundaries in a nickel-based superalloy:

COSSMS (2014) C. Stein, S.-B. Lee, A.D. Rollett

Probability plots made with R – www.r-project.org



Conclusions: from examination of several micro-cracks, computing the *resolved shear stress* (using elastic FFT) on coherent-twin-parallel slip systems does *not* account for the crack initiation. Boundary length appears to play a significant role, which is reasonable in light of the importance of ALA grains. All this obtained from surfaces, i.e. 2D data. Distributions are *not* log-normal.

Example: Thermal Barrier Coatings

Why use FFT?

Thermal barrier coatings are known to fail through cracking. The micromechanical fields allow us to analyze for the potential causes.

Expected Result?

The expected result is that the anisotropy in thermal expansion in the bond coat lead to high resolved shear stresses as a result of the elastic anisotropy of the material. In addition, it was anticipated that microstructure could affect the result.

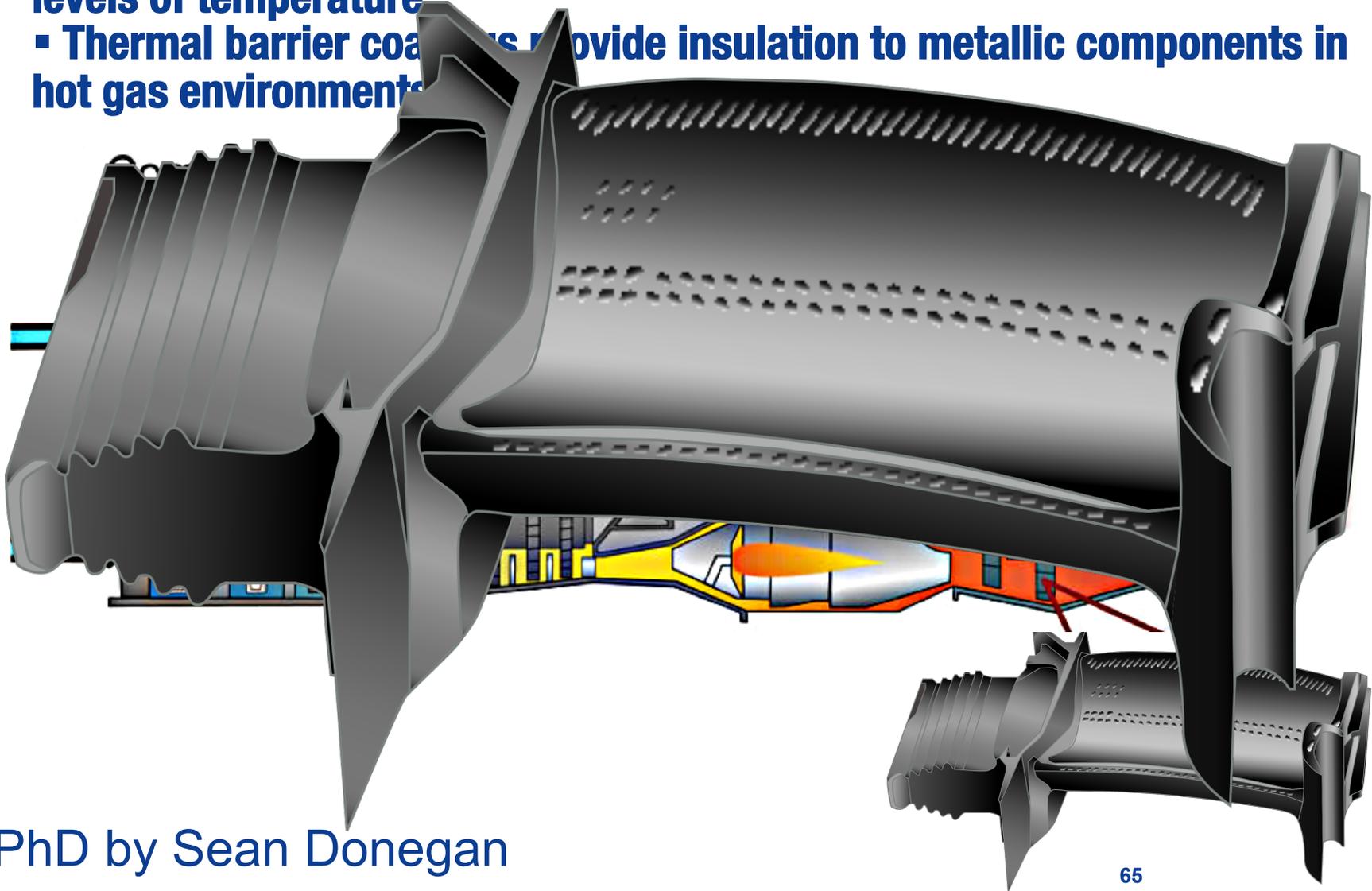
How was it done?

No measured microstructures were available. Synthetic 3D microstructures (315^3) were generated to represent typical TBC +bond coat+substrate microstructures.

PhD by Sean Donegan; DOE/NETL support

Thermal Barrier Coatings: Introduction

- Metallic components in gas-turbine engines are exposed to extreme levels of temperature
- Thermal barrier coatings provide insulation to metallic components in hot gas environments



PhD by Sean Donegan

Synthetic Structure Creation

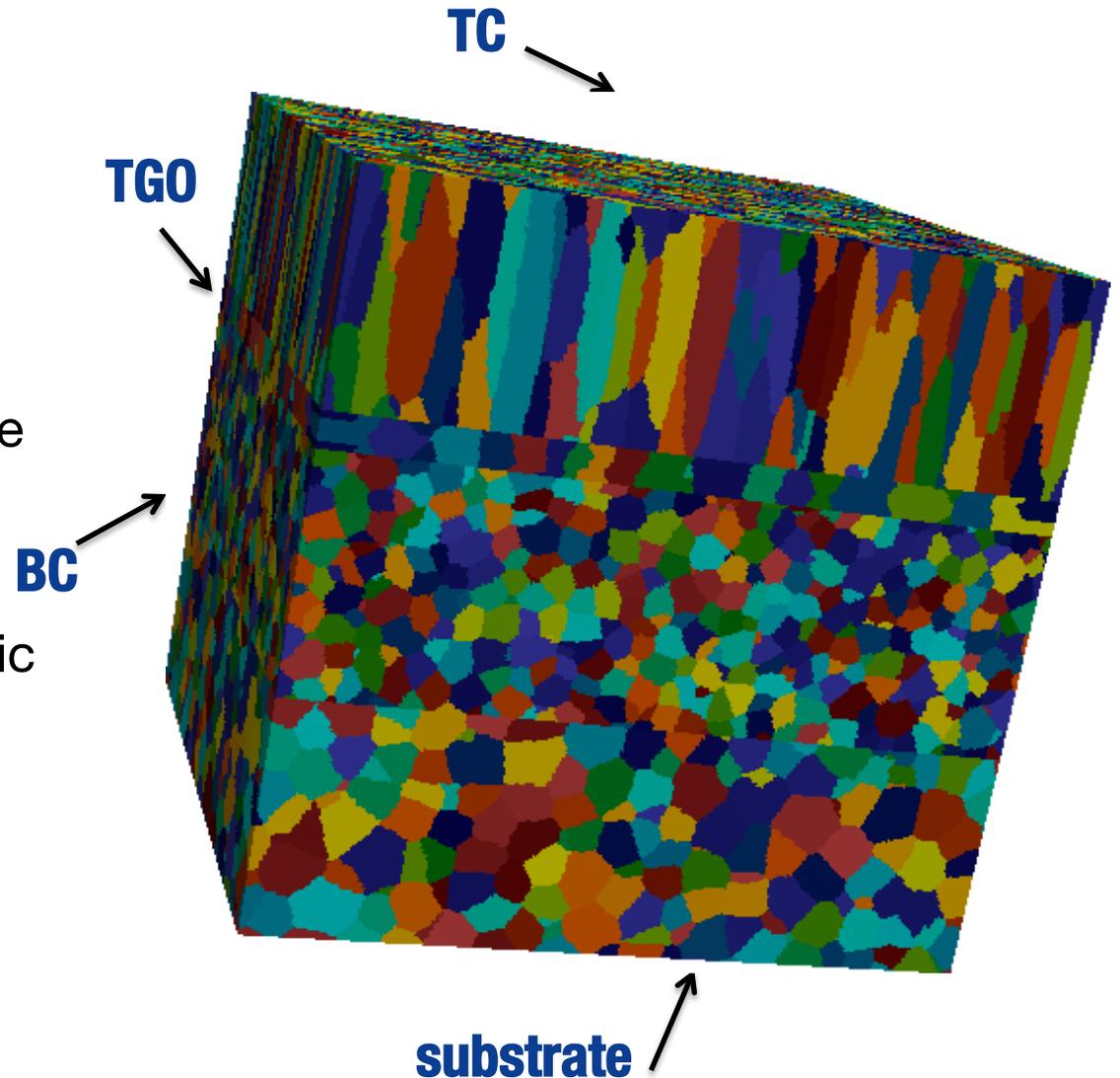
Microstructure, especially at the interfaces between TBC layers, plays a crucial role in failure. To better appreciate the role of microstructure, *DREAM3D* is used to generate test microstructures.

DREAM3D is a tool used to generate and analyze synthetic material microstructures. DREAM3D can create a 3D microstructure from a set of statistical data.

BC = Bond Coat

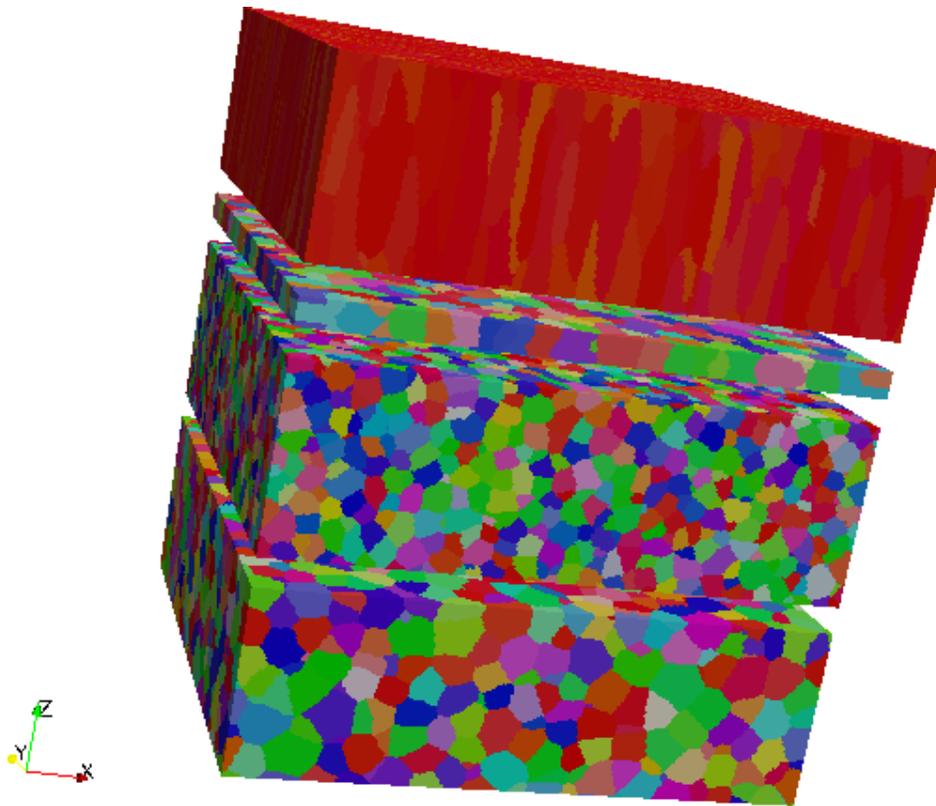
TGO = thermally grown oxide

TC = Top Coat (yttria)

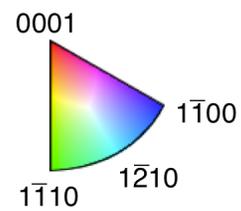
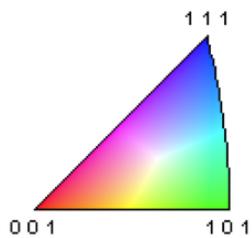
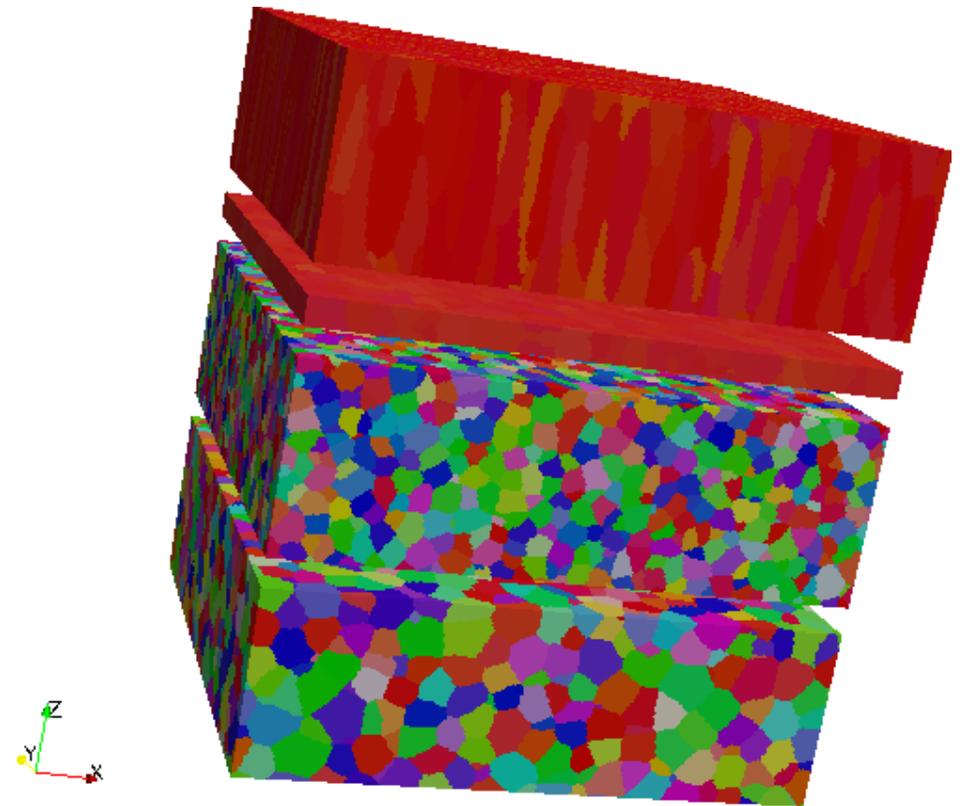


Texture Control

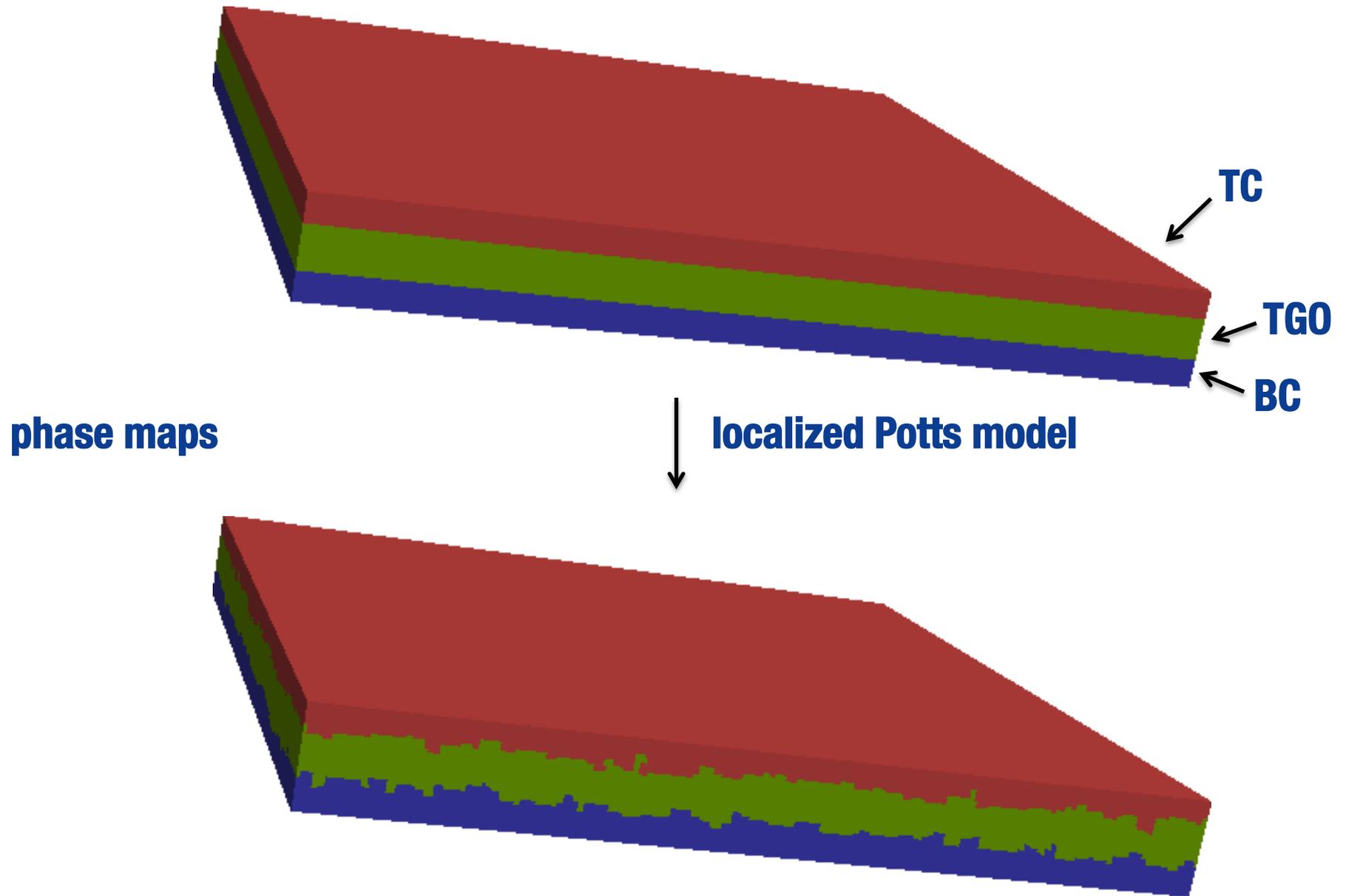
TGO: no texture



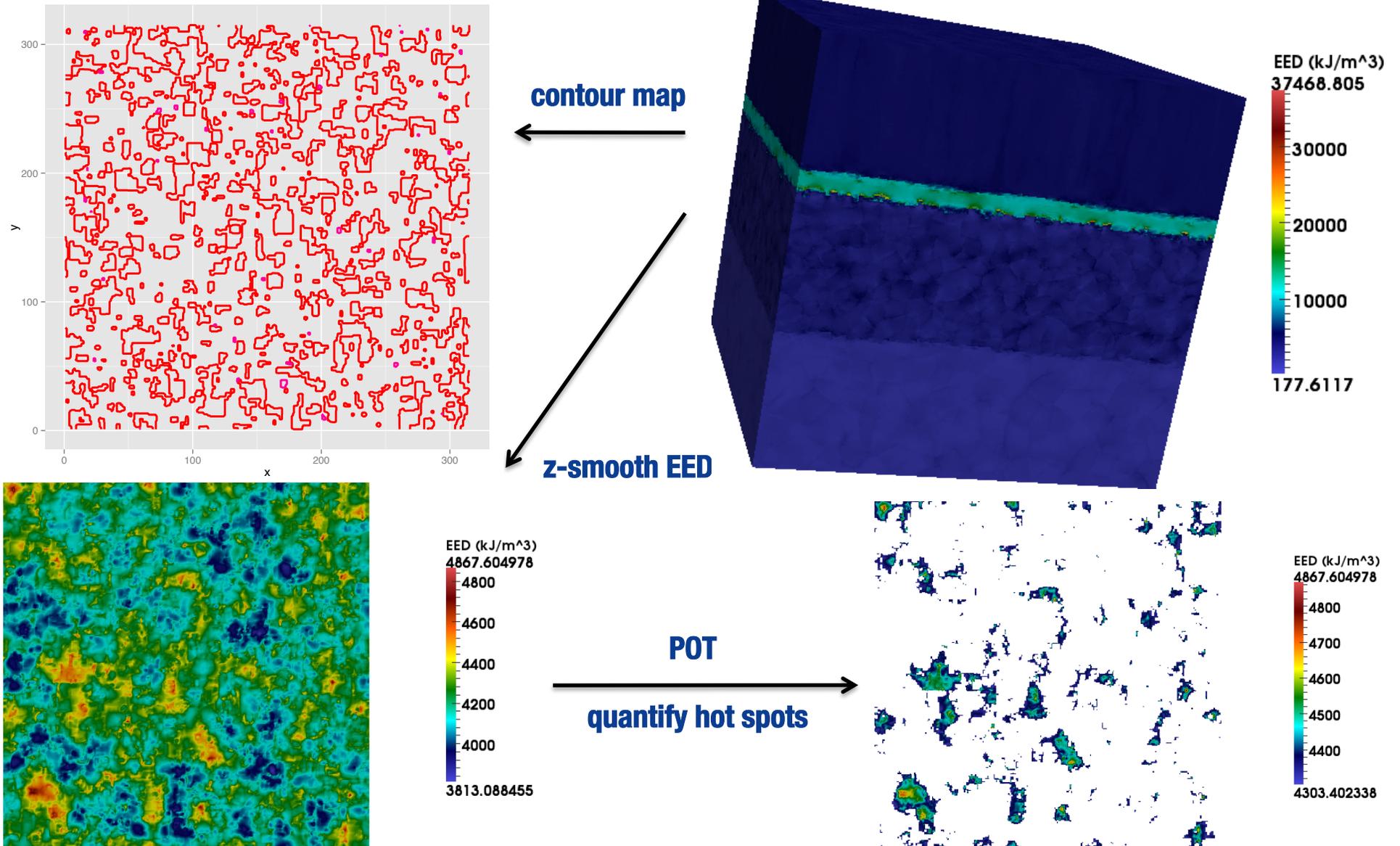
TGO: texture



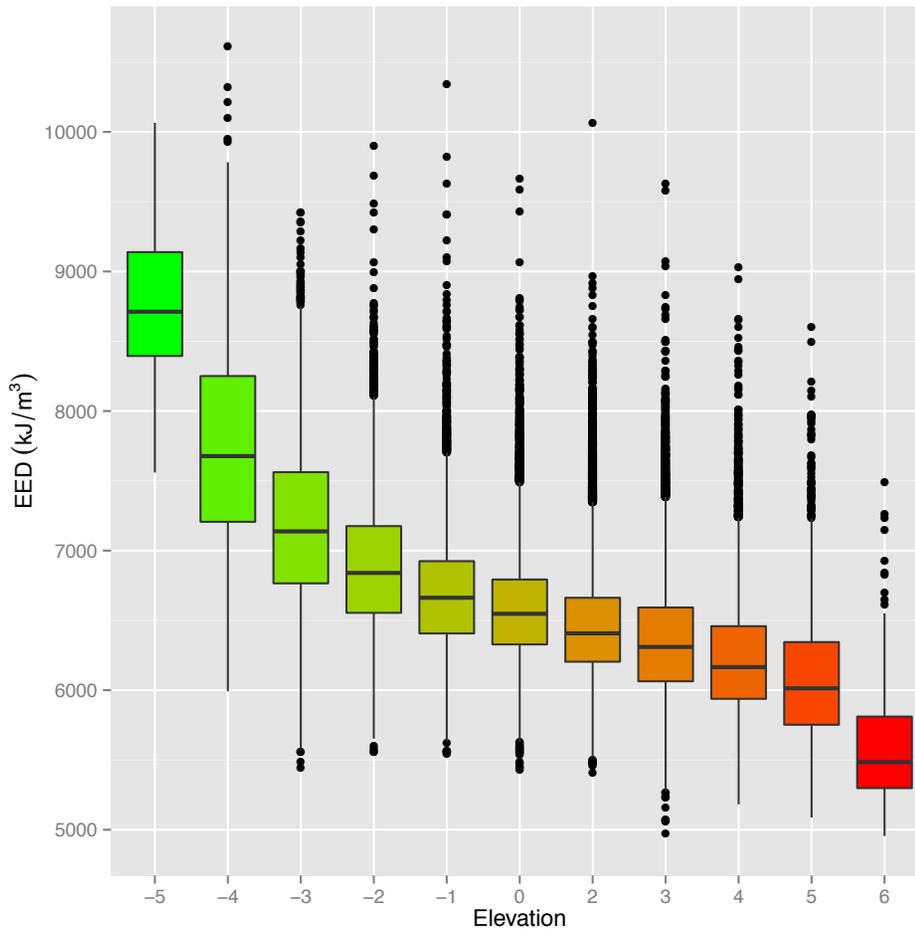
Potts Model Interface Relaxation



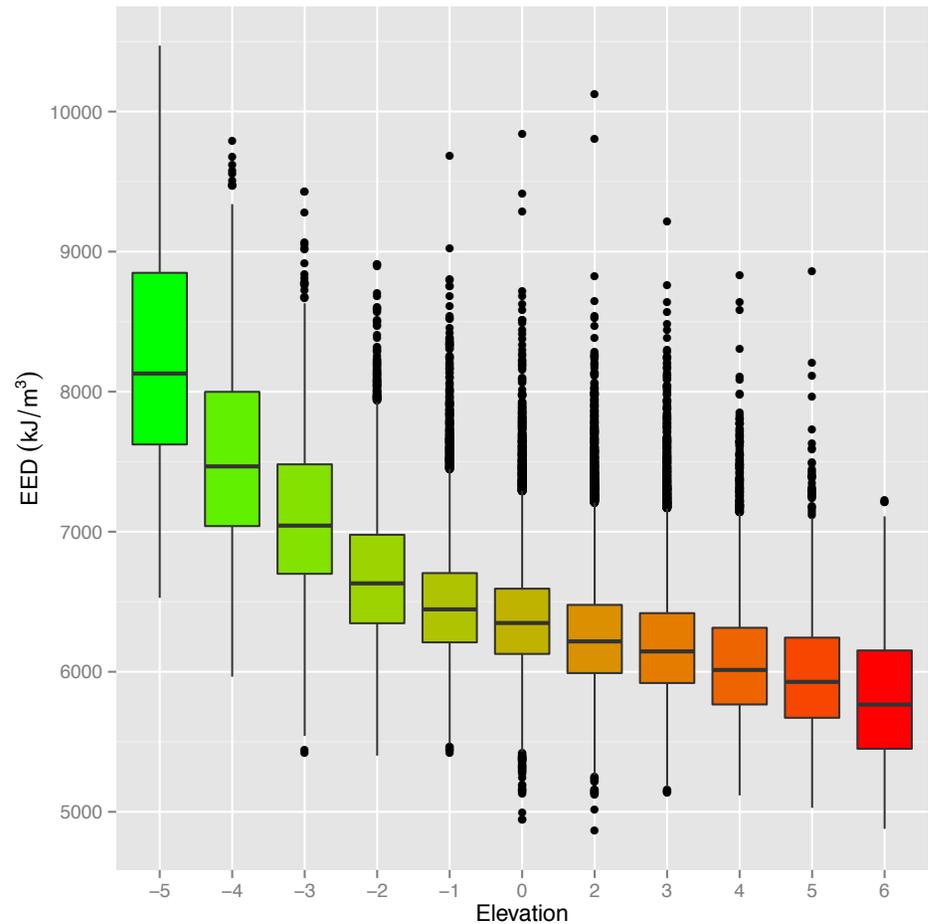
Correlate Hot Spots to Microstructure



Boxplots: BC/TGO (Lower) Interface

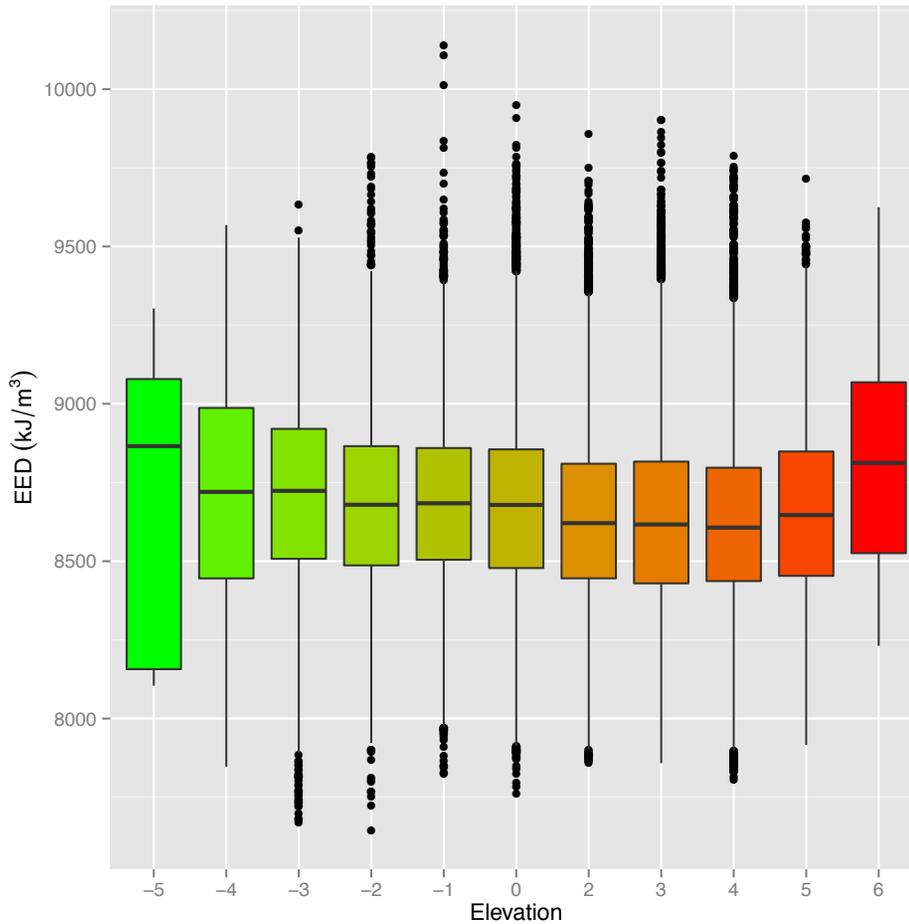


**columnar top coat,
textured TGO, (Ni,Pt)Al
bond coat**

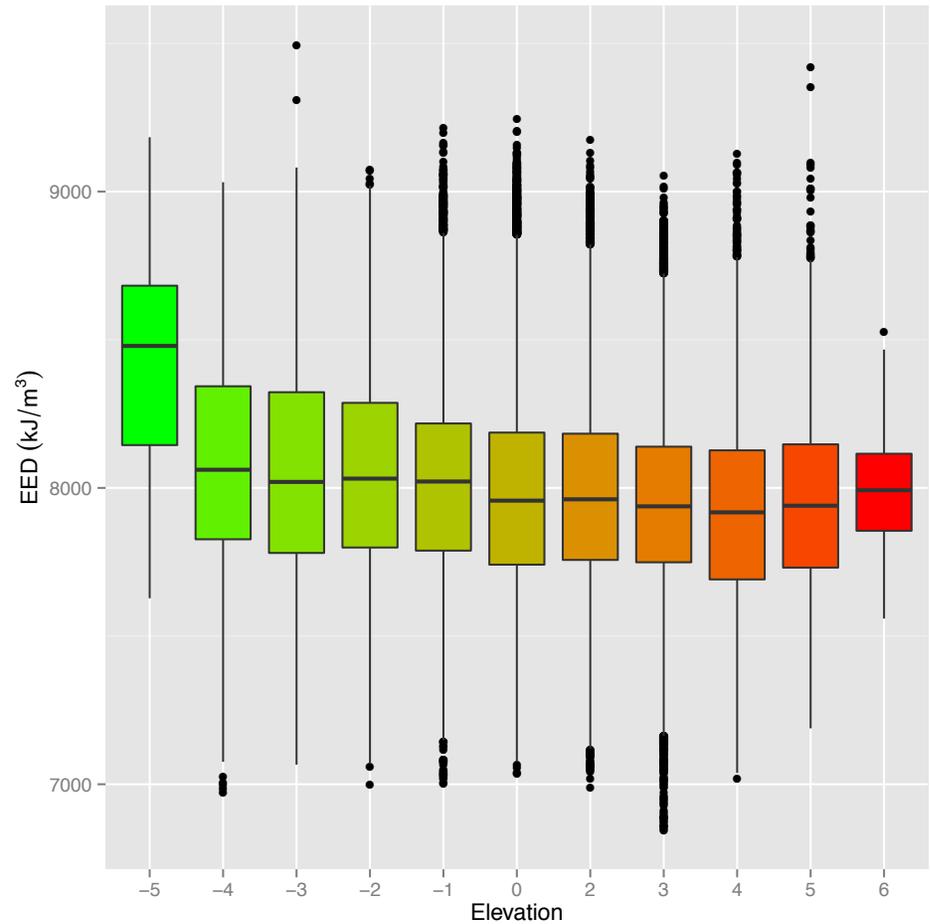


**splat top coat, textured
TGO, (Ni,Pt)Al bond coat**

Boxplots: TGO/TC (Upper) Interface



**columnar top coat,
textured TGO, (Ni,Pt)Al
bond coat**



**splat top coat, textured
TGO, (Ni,Pt)Al bond coat**

Example: Copper, tension

Why use FFT?

The objective is to validate the crystal plasticity model embedded in vpFFT and evpFFT in their various forms (e.g. dual grids, e.g. orientation gradient hardening).

Expected Result?

The desired result is that the FFT calculations will succeed in reproducing the experimental results, with appropriate choice of model and parameters. In fact, the variations in the computed results are small compared to difference with experiment.

How was it done?

High Energy Diffraction Microscopy (HEDM) was performed for multiple strain steps (snapshots) in tension tests on copper and zirconium.

PhDs by Reemu Pokharel and Jonathan Lind; DOE/BES support

Comparing Simulation with the 3D Copper Experiment

- Choose a central layer in the original undeformed volume. Locate the nearest equivalent layer in a deformed volume. Compute, pointwise, the change in orientation (resulting from the plastic strain). Separate that change (misorientation) into a magnitude (rotation angle) and an axis of rotation.
- Instantiate a simulation with the initial, undeformed volume. Simulate uniaxial tension with different models. Compare the same chosen layer against its initial state, point by point.

