SCORE Documentation

Release 1.0

Markus Kuehbach

August 28, 2016

SCORE is an MPI-parallelized 3D cellular automaton for the simulation of microstructure evolution during primary recrystallization in polycrystals.

The source code is developed by Markus Kühbach at the Institute of Physical Metallurgy and Metal Physics with additional contributions from Luis Antonio Barrales-Mora.

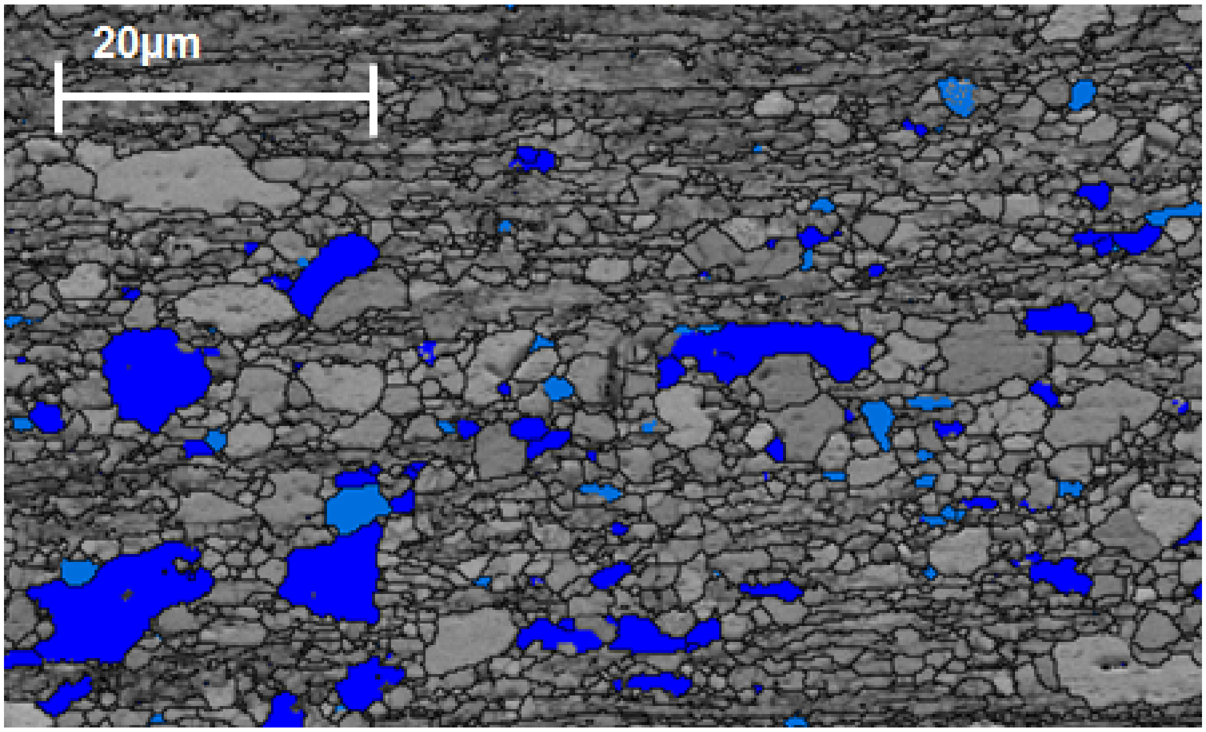
The compilation of SCORE utilizes standard Linux tools.

CHAPTER 1

How to get? How to compile?

1. Check your Linux installation for a working installation of **cmake** and **make**
2. Pull the project from the git repository https://github.com/mkuehbach/SCORE
3. Check your compiler. Both the Intel compiler (v14.0 tested) and the GNU compiler (v4.8.3 tested) can be utilized.
4. Make sure there is a folder with a **src**, a **build** folder, and the CMakeLists.txt file.
5. Make sure that the source code files are in **src** while a **SCORE.*.uds** parameter file is in **build**.
6. Open a console and type *cd build*
7. Only once when setting up a new computer *cmake -DCMAKE_BUILD_TYPE=Release ..* (up to O2 optimization tested)
8. Compile the program with *make*
9. Find happily the binary in the **build** folder

Learn about the physical background to the model



Band contrast SEM/EBSD image of an electropolished AA8079 sample after 70% cold-rolling and isothermal annealing at 300 deg C for 6 minutes. .. The rolling direction is parallel to the scale bar. Blue denotes grains in cube orientation, high-angle grain boundaries drawn in black.

Recrystallization and recovery term a variety of deformation microstructure restoration processes during which the stored elastic energy in the deformed microstructure drives the elimination of excess dislocations and the migration of grain boundaries. Specifically the *static recrystallization* refers to a form of recrystallization which occurs after the deformation was applied. For microstructure evolution during the annealing of cold-rolled sheet material this static recrystallization is of significant technical interest. The physical mechanisms and the strength of the driving forces distinguishes it from static recovery and other forms of grain growth.

Scope of the model:

The SCORE model is designed for the simulation of the growth of new nuclei under the effect of anisotropic grain boundary mobility and heterogeneous deformation microstructures, i.e. for simulating *static recrystallization*. Furthermore, SCORE encompasses approximate models for the nucleation stage. As with every model the nature of the processes is depicted in simplified manner by making the following assumptions:

- A microstructure volume comprises cubic voxel as the smallest discretization unit.
- The deformation microstructure can be idealized in contiguous regions with homogeneous properties (orientation, dislocation density).
- Therein nuclei appear which are strain- and excess dislocation free.
- Grain boundary migration follows classical rate theory according to Turnbull's model.
- Grain boundary migration is simplified as a flat boundary sweeping the deformation structure.
- The intrinsic grain boundary mobility is only dependent on the crystallographic disorientation of the adjoining grains.
- The deformation microstructure remains static in terms of orientation and boundaries content.
- However the (statistically stored) dislocation density is allowed to evolve. This is referred to as recovery modeling, for which approximate physical models exist.
- Recrystallizing nuclei do not rotate or change their orientation during growth.
- A potential growth inhibitating effect of second-phase particles is modeled in accordance with the Zener-Smith theory.

Relevance of the model:

These conditions apply well to many cases of cold-rolled aluminium, nickel, copper alloys and steels, but require that the driving force stemming for dislocations is at least an order of magnitude larger than that of curvature. Under these conditions the SCORE model can predict the course of microstructure transformation in particular kinetics, indicate potential changes in the macrotexture, resolv spatially the impingement of individual nuclei and hence allows to extract very accurately the grain size distribution. The benefit of the model compared to similar approaches is that it enables the simultaneous initialization and execution of multiple computational domains over which the results can be averaged or analyzed in all details collectively. This renders the simulations statistical significant and sufficiently discretized to minimize bias and to quantify uncertainty that persist to unknown extend in other recrystallization models.

Further reading:

Cotterill, P., Mould, P. R.
Recrystallization and Grain Growth in Metals
Surrey University Press, London, 1976

Humphreys, F. J., Hatherly, M.
Recrystallization and Related Annealing Phenomena
Pergamon Press, 2003
ISBN: 978-0-08-044164-1

Gottstein, G.
Physical Foundations of Materials Science
Springer, Berlin, 2010
http://dx.doi.org/doi:10.1007/978-3-662-09291-0

Gottstein, G., Shvindlerman, L. S.
Grain Boundary Migration in Metals: Thermodynamics, Kinetics, Applications

CRC Press, Boca Raton, 2010
ISBN 9781420054354

Hallberg, H.
Approaches to Modeling of Recrystallization
Metals, 2011, 1, 16-48
http://dx.doi.org/doi:10.3390/met1010016

CHAPTER 3

2. Parameterization

Parameterizing and setting up advanced recrystallization models can appear cumbersome at first glance owing to the variety of settings and input parameter. In order to resolve many of the questions, please find in the following a detailed description of all input parameter to the SCORE model. The structure of the **.uds** file identifies where data should be read as strings (**s**), integers (**i**) or double precision floating point numbers (**f**). In general, the parameter values are categorized loosely into these groups.

In any case each parameter key and value needs enclosure in two double quotes,i.e. "Key" and "Value". In *.uds* files

3.1 Basic parameterization

|| ParameterList || *(ss) || Parameter, Value

CAEnsembleSize

How many automaton domains should be devised. By default the domains are of equal size and become listed into an automaton pool from which the processes pick and execute sequentially the domains.

The random number generator assures the domains to become differently initialized. This is because each process has a different seed.

XMAX

Determines a global recrystallized volume fraction at which the simulation is stopped, the results postprocessed and the program exits.

TIMEMAX

Serves to stop the simulation as does XMAX control but when a simulated realtime (not program execution time!), is reached.

Anyhow, the simulation does not simulate growth farther in time than the maximum annealing time provided.

NMAX

Controls the stopping of the simulation after a certain number of integration steps.

CellSize

Reads in as micrometer, defines the edge length, and thus the discretization of the cubic voxel.

It should be scaled such that even the smallest grain of interest is represented by a sphere equivalent radius of at least 10 cells.

Cells smaller than 100 nanometer certainly are no longer in accordance with a mesoscopic model at the micron-scale.

3DCAEdgeLengthInCellsRD

3DCAEdgeLengthInCellsND

3DCAEdgeLengthInCellsTD

Defines the voxelization of each automaton domain, while RD is parallel to x, ND parallel to y, and TD to z, currently limited to 1600 as uint32 is utilized to store the unique assignment of cells and grains

MeanDefGrainSizeinRD

MeanDefGrainSizeinND

MeanDefGrainSizeinTD

Reads in micron, only of significance when **DefStructureSynthesis** is “1”

DefStructureSynthesis

Determines which deformation microstructure synthesis model is desired. “1” renders a grid of equally-sized cuboid grains should be constructed, “2” constructs discrete Poisson-Voronoi.

In order to avoid bias, the tessellation is sampled at random from a much larger continuous realization of a Poisson point process.

MeanDefSizePoissonDiameter

Reads in micron, only of significance when **DefStructureSynthesis** is “2”

NucleationDensityLocalCSR

Assuming each domain maps a random point process, how many nuclei would on average be observed in the domain.

CSRNucleation

Allows the placement of bulk nuclei inside the domain when set to either “1” or “2”.

“1” enforces always the same number of nuclei in each automaton which are placed in space according to random 3D coordinates.

“2” enforces that the domain itself is perceived as framing a much larger Poisson point process from which all points that fall in the domain are taken and mapped on the voxel grid.

Mind that “1” and “2” sample slightly different, though uncorrelated point processes. For domains of **EdgeLengthInCells** sizes larger 1000 spatial correlations of the PRNG become an issue. If in doubt test and utilize the MersenneTwister.

GBNucleation

Allows the placement of nuclei at boundaries among deformed grains when set to either “1” or “2”.

“1” enforces nuclei at all boundaries regardless the disorientation of the two adjoining grains.

“2” enforces nucleation at discrete grain boundaries only to occurr at high-angle boundaries, assuming a default threshold disorientation of 15degree

GBNucDensity2Number

GBNucRhoDiff2Density

Currently heuristic grain boundary nucleation rate modeling parameter.

GBNucMaximumScatter

Controls how strongly are grain boundary nuclei at most deviating from the matrix orientation.

IncubationTimeModel

“1” assumes nucleation to occur and grow at t=0seconds, i.e. instantaneously/site-saturated.

“2” assumes nuclei to start growing after a delay, the probability distribution of the delay times is a Rayleigh distribution.

IncubationTimeScatter Scales in seconds the only parameter of the Rayleigh distribution according to which the nuclei become placed.

Mobility model

Determines as to how the grain boundary mobility is parameterized from a disorientation among two grains.

"1" follows the approach taken by Sebald and Gottstein to leave boundaries separating less than 15deg as practical immobile,

all others of equal mobility and only to provide the close to 40deg about 111 disorientations in misorientation space an higher value ("GS")

"2" adopts an approach going back to Humphreys and Huang that low-angle grain boundaries are practical immobile only up to a disorientation of several degrees after which a smooth sigmoidal shape is assumed.

At the moment in this model 2, close to 40°<111> boundaries are not specifically more mobile than other high-angle grain boundaries.

3.2 Output flags

RenderingOfMicrostructure

"0" no microstructure snapshot is outputted, further speeds up the simulation because minimizing file interaction

"2" 2D zsections of the RDND plane are outputted taken at TD = 0.5, reliable option when the structure evolution is to be judged qualitatively, as one would section in SEM/EBSD

"3" 3D binary files are written with MPI I/O that store the voxelized microstructure. **Mind color model and hint on how to import into visualization environment.**

RenderingColorModel

Defines the coloring model of the rendering as this allows directly the RGB triplets to be read into visualization software

"1" disjoint grains with unsigned int IDs

"2" IPFZ coloring with aka inverse polefigure coloring.

Please mind that the details (white point location, RGB gradients) differ from vendor to vendor.

For details consider the code and mind the possibility of stretching the RGB gradient asymmetrically over the standard triangle.

RenderingBoundaries

Allows to identify all voxel located at a boundary among deformed grains to be plot when set to "1".

As with generating 3D raw data, these options should be omitted whenever possible to reduce file interaction, MPI I/O is utilized.

OutputLogBoundaries

If set to "1" a file detailing all connected voxel the form a discrete grain boundary patch are outputted with properties of the adjointing grains.

OutputSingleGrainStats

When set to "1" renders for each automaton a detailed history of how all grains have evolved to study impingement topologies and growth concurrencies.

OutputOfHPF

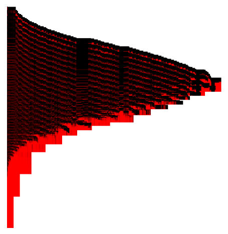
Calculate host grain distance function as detailed in the model description that can be found in the reference section.

OutputRXFrontStats

Developers option: when set to "1" all CA! output detailed loginformation on the status of their interface cell list.

This can help to find minimal sized container lengths in order to optimize further program performance and to reduce the dynamic memory footprint.

> *The basic idea of the automaton is that it accumulates a linear container of interface cells and runs two managing list that store positions*
>
> *in this container that can be written to safely or which have recrystallized strongly enough to infect new cells. These lists are maintained throughout each simulation step.*
>
> *The purpose of this list management is to minimize re-allocations and to maximize spatial and temporal locality in the cache lines.*
>
> *Most microstructure paths S_V(X) tend to have a two-fold characteristic in the way that at the beginning of the transformation when*
>
> *growth dominates more cells infect neighbors than already recrystallized cells become decommissioned for recycling.*
>
> *At some point however the area of the RX front that is this umimpinged to grow reduces progressively while at the same time more and more cells become decommissioned/switched off.*
>
> *Working on a linear array to which always fragments are appended this renders a progressively stronger and fragmented list containing gaps the activity check of which creates unnecessary overhead.*
>
> **Mind the defragmentation option** to reduce these gaps and as well the developer functionality **FRAGMENTATION RADAR** which when defined active in the code #define FRAGMENTATION_RADAR
>
> *generates a 2D graph that elucidates the occupancy of the cell array suggesting room for further improvement.*



A fragmentation radar is read like a book from the upper left corner to the lower right.

Each pixel in each horizontal line represents a the average number of active cells in a block of cells **BLOCKLENGTH**.

Red denotes low activity (fragmented) while black is optimal.

Each vertikal lines denotes one simulation step.

3.3 Integrator and performance

|| IntegratorAccuracy || *(ss) || Parameter, Value

MAXFILLPERSTEP

> Controls the volume increment that the fastest migrating boundary protrudes into a cell. Reducing this value from the default "0.1" renders simulations slower,
>
> as the number of integration steps scales approximately linear. When increased up to its maximum 0.5 cuts integration accuracy.

RediscretizationSteps

> Controls into how many equi-sized time steps the interval [0,tmax] becomes rediscretized.

InitialCacheSizeRXCells

How many of all cells in the network should be preallocated to become at some point active in the front.

For finely discretized automata with even the smallest grains 5-10 cells in spherical equivalent radius values between 0.1-0.2 are sensible defaults.

The **OutputRXFrontStats** can provide information on how to reduce further this number, in any case however, the automaton reallocates memory cells, if permitted by the operating system, but at the cost of execution speed.

ReCacheSizeRXCells

Controls with how many elements the cell container is extended when more cells become active then expected. In such case, the smaller **ReCacheSiteRXCells** the more potential small reallocations.

3.4 Material properties

|| MaterialProperties || *(ss) || Parameter, Value

ZeroKelvinShearModulusG0

Reads in Pascal, mind at zero Kelvin!

FirstOrderdG0dT

Reads in Pascal, linear shear modulus temperature coefficient

ZeroCelsiusBurgersVector

Reads in meter, mind at zero degree Celsius!

AlloyConstantThermalExpCoeff

FirstOrderThermalExpCoeff

SecondOrderThermalExpCoeff

Thermal lattice expansion coefficients appearing in Willey's model, C constants see Hatch JE, Aluminum, see also Touloukian.

MeltingTemperature

Reads in degrees Celsius!

3.5 Grain boundary migration

LAGBm0

Reads in m^4/Js, Preexponential factor intrinsic mobility of low-angle boundaries, default truncation at 15degrees disorientation among adjoining grains

LAGBHact

Reads in eV, activation enthalpy low-angle grain boundaries

HAGBm0

HAGBHact

GSm0

GSHact

Corresponding values for general high-angle grain boundaries, and those special **GS** boundaries close to 40deg111 in misorientation space

RHModelHAGBm0

Reads in m^4/Js, preexponential factor intrinsic mobility of high-angle boundaries

RHModelHAGBHact

Reads in eV, activation enthalpy high-angle grain boundaries

RHModelLAGBHAGBCut

RHModelLAGBHAGBTrans

RHModelLAGBHAGBExponent

parameterizes sigmoidal model "2" as **m0** * exp(-**Hact**/kT)* (1.0 - **Cut** exp(- **Trans** (DisoriAngle/Trans)^ **Exponent**)

3.6 Recovery modelling

|| RecoveryParameter || *(ss) || Parameter, Value

Mind that the following properties have not yet been fully validated!

RecoveryConsider

allow for a gradual reduction of the dislocation density to mimic a reduction of the stored energy, choose the rate determin step

"1" Erik Nes's lateral jog drift controlled thermally activated glide aka vacancy diffusion controlled

"2" Erik Nes's solute diffusion aka drag controlled network growth approach

All other numbers, no recovery

RecoveryVacancyDiffGeometry

Vacancy core diffusion model parameter

RecoveryVacancyDiffPreexp

Vacancy core diffusion preexponential

RecoveryVacancyDiffHact

eV, Vacancy core diffusion activation enthalpy

RecoverySoluteDiffPreexp

Solute drag diffusion model preexponential

RecoverySoluteDiffHact

eV, Solute drag diffusion activation enthalpy

RecoverySoluteLsPropFactor

Solute drag solute pinning point distance proportionality factor

RecoverySoluteConcentration

Solute drag solute concentration

RecoveryParameterAlpha3

Nes model alpha3 constant

RecoveryParameterKappa2

Nes model jog separation kappa2 constant

RecoveryParameterC3

Nes model C3 parameter

RecoveryParameterC4

Nes model C4 parameter

RecoveryParameterC5

Nes model C5 parameter

following:

Nes, E.
Recovery Revisited
Acta Metallurgica et Materialia, 1995, 43, 2189-2207
http://dx.doi.org/doi:10.1016/0956-7151(94)00409-9

3.7 Particle drag modelling

|| ZenerDragParameter || *(ss) || Parameter, Value

ZenerConsider
Should the classical Zener-Smith drag relation be utilized to simulate slowed grain boundary migration
"0" no
"1" drag is constant for all boundaries in the whole domain, then the dispersion is taken from the first line in **EvoDraggingParticles**
"2" drag is time-dependent in accord with the particle dispersion evolution detailed in **EvoDraggingParticles**

ZenerAlpha
Prefactor appearing in the various geometric limiting cases, classically 3/2

ZenerGamma
Grain boundary energy 0.324 J/m^2 from Meyers Murr Interfacial

3.8 Texture components

|| IdealComponents || *(sffff) || Name, IdealcomponentBunge1, IdealcomponentBunge2, Idealcomponent-Bunge3, IntegrationRange

Here typical modal orientations that are often identified with Euler space symmetry positions or orientations which show strong maxima in experimental ODFs aka Standardlagen are listed. Each orientation in the simulation is categorized to the closest of all these idealcomponents. Is an orientation not included in the integration range specified of any component, it is considered random/phon.

3.9 Microstructure snapshots

|| RenderMicrostructure || *(sf) || Local recrystallized fraction, local value

Add one line for each automaton-local recrystallized fraction at which microstructure snapshots (either 2D or 3D) should be taken. The format is "X" 0.1, i.e. to render at ten percent recrystallized.

|| RenderOnlyTheseRegions || *(i) || CAID indices start from 0 to n minus one

List all CA IDs [0, CAEnsembleSize-1] that should be included in the plotting. **This option allows to reduce drastical the amount of snapshots taken and helps reducing file system congestion.**

3.10 Annealing profile T(t)

|| ProcessingSchedule || *(ff) || time, temperature linearized heating profiles

For numerical performance it is admissible to coarsen the linearization of measured heating profiles rather than to import thousands of points.

3.11 Deformed grains pool

|| DeformedGrainsPool || *(ffffff) || Bunge e1, Bunge e2, Bunge e3, rho0, dsubav0, disoriav0

By default the deformed grain pool provides a list of grains all of which can but not necessarily are required of all properties different. Random sampling from this list and placing on sequentially on a 3D grid generates spatially uncorrelated MODF. *Mind that a departure from the well-known MacKenzie misorientation distribution is per se not a sign of spatial correlation in the case of a non-random ODF, as the MacKenzie distribution is inferred from a random texture!*

for the interested reader:

Mason, J. K., Schuh, C. A.
The generalized Mackenzie distribution: Disorientation angle distributions for arbitrary textures
Acta Materialia, 57, 2007, 4186-4197
http://dx.doi.org/doi:10.1016/j.actamat.2009.05.016

3.12 Recrystallized grains pool

|| RXGrainsPool || *(ffff) || Bunge e1, Bunge e2, Bunge e3, tincub

By default this is the current way of devising in a flexible manner many different recrystallization nuclei.

3.13 Further settings

|| HeuristicRXFrontListDefragmentation || *(sf) || Recrystallized fraction, local value

This option allows to defragment the cell container by copying explicitly pieces of information from cells far apart from the start of the container to the gap formed by cells which have become decommissioned, i.e. *INACTIVE*. This allows in the subsequent time steps to reduce the total number of cells that have to be checked, and hence increases performance and cache-locality. Experience has provided sensitive defaults.

|| EvoDraggingParticles || *(ff) || time, fr

Can be utilized to prescribe the Zener drag by a priori Zener drag simulations. *Please mind that strictly speaking when Zener drag affects significantly microstructure transformation it occurs often concurrently to RX which is when such concept* **cannot** *be applied.*

|| SingleGrainVolOverTime || *(sf) || Recrystallized fraction, local value

Controls at which automaton-local recrystallized fraction the size of the grains should be measured and stored in main memory such that it can later serve to interpolate the grain size during postprocessing.

3. Simulations

A simulation is issued by the following command line call:

```
mpiexec -n <proc> <scorempi> <Input.uds> <SimulationID>
```

<*scorempi*> denotes the name of the binary file. In its current version two input arguments are necessary:

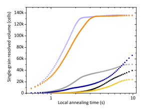
<Input.uds>
: this is the settings file of *uds* extension, relative path references are utilized by the program

<SimulationID>
: this is a positive integer with which all simulation result can be uniquely labeled

Mind that executing multiple simulations with the same ID in the same directory overwrites unpredictively previously obtained results without prompting!

CHAPTER 5

4. Model output



Kinetics, macrotexture and grain size distribution are the key state parameters of interest.

It is practical to distinguish the output in two categories.
The first considers the simulated state variable values and a depiction of the microstructure.
The second considers information of the model performance.

All output from a simulation is written in the output folder using relative addressing. The *example/jmak* folder provides a simple recrystallization case study of 1000 nuclei growing into a deformed monocrystal. The output contains several files, all of which have a prefix **SCORE.<SimulationID>** to make distinguishing results simple. Most output is plain ASCII formatted in such a way that it can be directly imported in post-processing, and spreadsheet tools.

5.1 Recrystallization kinetics

The **Rediscretized.Kinetics.csv** file contains a table with the typical time, recrystallized fraction as well as the Avrami plot logarithmic linearization. The kinetics of each automaton can be obtained by switching on the **OutputRXFrontStats** with the first column giving the individual consumption of each domain and the further columns provide details about the microstructural path an the memory utilization of the interface cell lists.

5.2 Macrotexture evolution

The **Rediscretized.Macrotexture.csv** file contains a table with the evolution of the macrotexture components as they were defined as **IdealOrientations**. Counts are listed in cells summarized and rediscretized over all domains with the total amount of cells in the ensemble to the right.

5.3 Grain size distribution

The **Rediscretized.FinalGrainSize.csv** file contains a table with the final size of all grains aggregated over all domains. The header line provides basic descriptive statistics of the whole ensemble. Beneath, a table contains a sorted list of all grains with their size, the domain they grew, in their orientation class as an index of the n-th **IdealOrientation**, the normalized cumulative grain size distribution to inspect tail departures, and the incubation time after which they were instantiated in the simulation. Noteworthy, this file contains only nuclei that participated in the transformation.

5.4 2D sections

A section of the microstructure at various recrystallized volume fractions can be printed. For this logpoints have to be defined in the input file block named **RenderMicrostructure**. Exemplarily, if one desires to output the microstructure at 25% and 76.8% recrystallized volume fraction, two lines should be placed in this block.

“X” 0.25
“X” 0.768

Lode Vandevenne’s open source lodePNG (https://github.com/lvandeve/lodepng/) tool is utilized to render indexed RGB **png** files that are stored in the output directory. The coordinate origin is the lower left corner. The coordinate system identifies x as the first coordinate, y as the second and z as the third. This is to of relevance for all implicit addressing in the source code. By default the whole domain is sectioned at the z coordinate equal to 0.5.

5.5 3D data

It is also possible to render three-dimensionally the structure. **Mind, though, that this equals a file size of 4 byte per cell, i.e. a simulation with only 1000 x 1000 x 1000 already requires 4 GB of disk space.** However, it is possible to restrict which simulation domains are rendered by indenting explicitly in the input file the respective automaton IDs under **RenderMicrostructure**.

These files are stored as binary **raw** files in the working directory. At the moment there is no header associated with the file so remembering their size is important.

For both 2D and 3D data two coloring modes are currently implemented. The first is coloring according to the grain ID. The second is to colorize according to inverse polefigure orientation mapping in the stereographic standard triangle of under fcc crystal symmetry and triclinic sample symmetry.

5.6 Single grain growth charts

The **SingleGrainData** sheet compiles the detailed evolution, nucleation sites, orientation, state of the deformed grains, automaton-resolved to allow insights into the evolution of particular grains, their incubation time, and much more. Furthermore, but currently commented out is an option to export these data by plain MPI I/O for post-processing purposes.

5.7 Performance

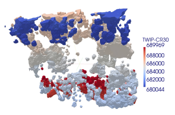
First the **ProfilingLog** csv file present a summary of all automata that were executed. Along with MPI_Wtime-ings in front and past barriers, in conjunction with a detailed information about how many nuclei in which orientation into which deformation structure, this file is the first address for a statistical interpretation of the data.

In particular for developers the **RXFrontsStats** csv file provides detailed information about the course of microstructure evolution as well as microstructural path parameter, discrete interfacial area the status of the recrystallization front memory management containers, etc.

The developmental hybrid (MPI/OpenMP) parallelized extension of the model adds a further output file **OMPThread-Profiling** which details how much time the OpenMP threads spent in each section of the code and synchronization.

CHAPTER 6

5. Visualize



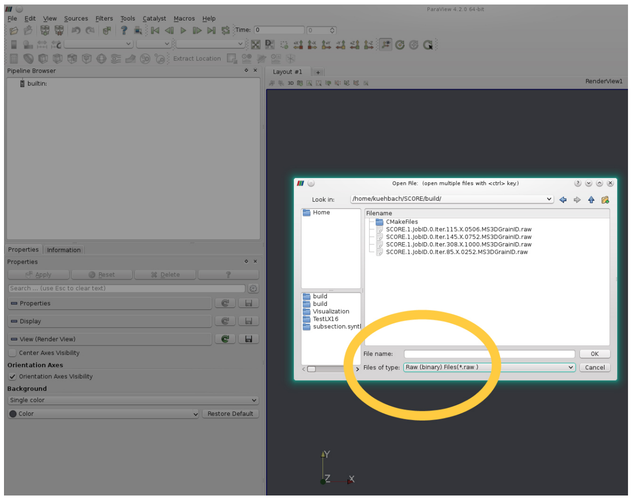
The 3D voxelized microstructure snapshots that become generated read as plain implicit 3D arrays.
As such, they can readily be imported into visualization software, the details how to do so are described below:

6.1 How to import SCORE .bin files into Paraview?

By default, little endian is assumed. Then **.raw** files generated by SCORE can be imported readily into Paraview as shown in the following screenshots:

0. Paraview is an open source visualization and analysis software (http://www.paraview.org/).
1. Load the corresponding raw-file with the Paraview binary RAW importer
2. Interpret as little endian, unsigned int, extend 1, a cubic domain of size N x N x N cells is imported by [[0,N-1], [0,N-1], [0,N-1]
3. Data origin is lower front left.
4. When the structure is rendered as **GrainID** a simple thresholding of grains of possible.
5. This applies as well for importing grain boundaries data.
6. Inverse polefigure coloring encodes each cell as a combination of three 1B unsigned char (R,G,B).

Click to enlarge on the images! The example refers to a 200 x 200 x 200 structure.

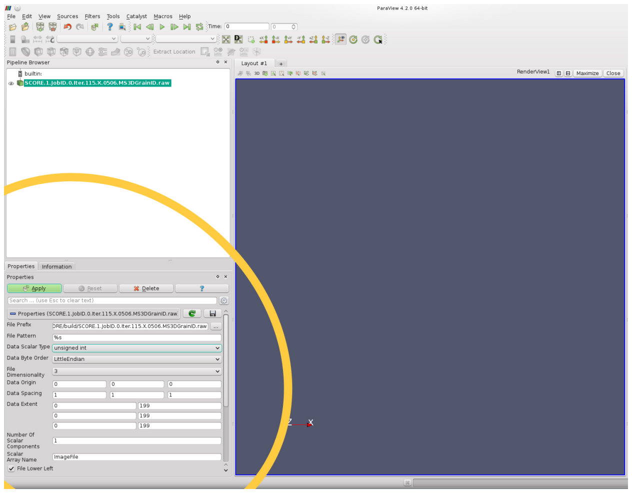


Open File: (open multiple files with <ctrl> key.)

Look in: /home/kuehbach/SCORE/build/

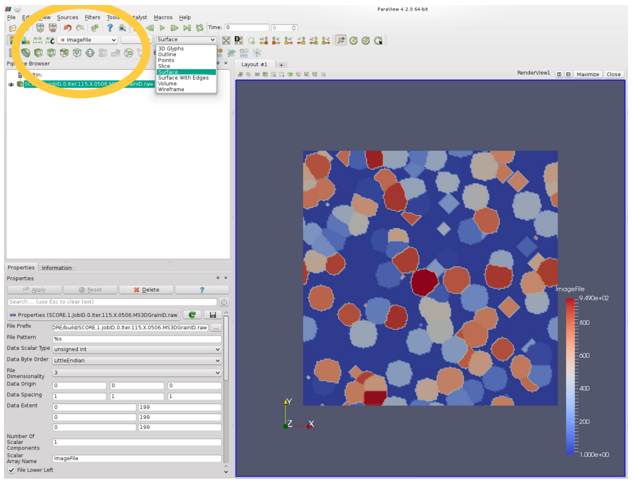
Filename
CMakeFiles
SCORE.1.JobID.0.Iter.115.X.0506.MS3DGrainID.raw
SCORE.1.JobID.0.Iter.145.X.0752.MS3DGrainID.raw
SCORE.1.JobID.0.Iter.308.X.1000.MS3DGrainID.raw
SCORE.1.JobID.0.Iter.85.X.0252.MS3DGrainID.raw

File name:
Files of type: Raw (binary) Files(*.raw)



Properties (SCORE.1.JobID.0.Iter.115.X.0506.MS3DGrainID.raw)

File Prefix ORE/build/SCORE.1.JobID.0.Iter.115.X.0506.MS3DGrainID.raw
File Pattern %s
Data Scalar Type unsigned int
Data Byte Order LittleEndian
File Dimensionality 3
Data Origin 0 0 0
Data Spacing 1 1 1
Data Extent 0 199
0 199
0 199
Number Of Scalar Components 1
Scalar Array Name ImageFile
✓ File Lower Left



Properties | Information

Properties

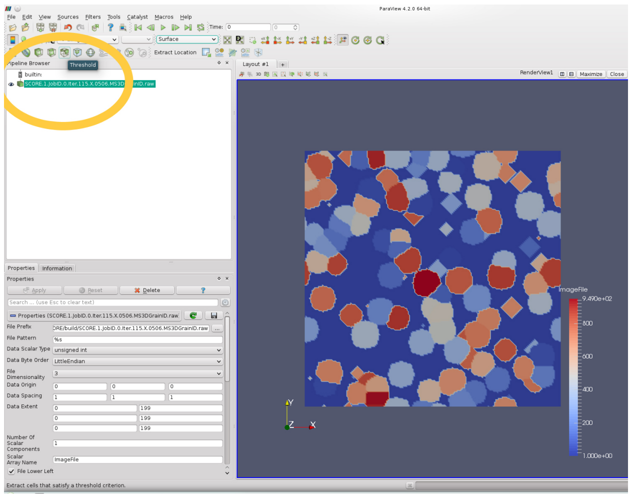
Apply | Reset | Delete | ?

Search ... (use Esc to clear text)

Properties (SCORE.1.JobID.0.Iter.115.X.0506.MS3DGrainID.raw)

File Prefix	ORE/build/SCORE.1.JobID.0.Iter.115.X.0506.MS3DGrainID.raw	
File Pattern	%s	
Data Scalar Type	unsigned int	
Data Byte Order	LittleEndian	
File Dimensionality	3	
Data Origin	0	0	0
Data Spacing	1	1	1
Data Extent	0	199	
	0	199	
	0	199	
Number Of Scalar Components	1	
Scalar Array Name	ImageFile	

✓ File Lower Left



Properties | Information

Properties

Apply | Reset | Delete | ?

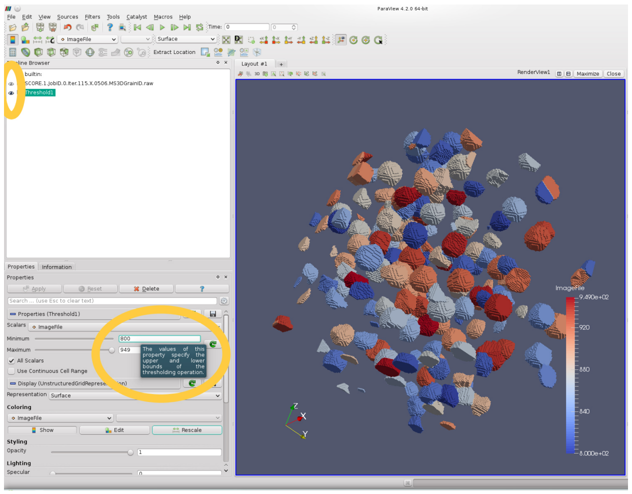
Search ... (use Esc to clear text)

Properties (SCORE.1.JobID.0.Iter.115.X.0506.MS3DGrainID.raw)

File Prefix	ORE/build/SCORE.1.JobID.0.Iter.115.X.0506.MS3DGrainID.raw	
File Pattern	%s	
Data Scalar Type	unsigned int	
Data Byte Order	LittleEndian	
File Dimensionality	3	
Data Origin	0	0	0
Data Spacing	1	1	1
Data Extent	0	199	
	0	199	
	0	199	
Number Of Scalar Components	1	
Scalar Array Name	ImageFile	

✓ File Lower Left

Extract cells that satisfy a threshold criterion.



Properties (Threshold1)
Scalars: ImageFile
Minimum: 800
Maximum: 949
The values of this property specify the upper and lower bounds of the thresholding operation.
✓ All Scalars
Use Continuous Cell Range
Display (UnstructuredGridRepresentation)
Representation: Surface
Coloring: ImageFile
Styling
Opacity: 1
Lighting
Specular: 0

CHAPTER 7

References

This model:

Kühbach, M., Gottstein, G., Barrales-Mora, L.A.
A Statistical Ensemble Cellular Automaton Microstructure Model for Primary Recrystallization
submitted to “Acta Materialia” on Dec, 27, 2015

Funding:

The authors gratefully acknowledge the support from the DFG in the frame of the Reinhart Koselleck project (GO 335/44-1).
Furthermore, we acknowledge the support from the FZJuelich and RWTH Aachen University within the project JARAHPC projects.

CHAPTER 8

Version history

v1.0 first version,

with deformed grain boundary detection and nucleation model, Poisson-Voronoi structure, Poissonian nucleation

site-saturated and time-dependent nucleation, MPI-parallelized, summary statistics and basic voxel structures

MPI/OpenMP extension

In addition the code has been modified for thread-parallelism with OpenMP, and the source code is made available in the **SCORE_MPIOMP_20151224.zip** file that resides in the **hybrid** folder. The user is motivated to extract the archive in a separate folder and to build the program as it was described already for the MPI-only version.

This hybrid SCORE model works as the MPI-parallelized version, and enabled a reduction of the execution time by a factor of 9, at most, as it was benchmarked on a Bullx X5675 Xeon two socket cluster blade system with exclusive usage of the 12 physical cores (OMP_NUM_THREADS=12) and explicitly bound the threads to the cores (KMP_AFFINITY). It is possible already with this model to handle Poissonian nucleation in cuboidal deformed grain structures.

However, **not all internal functions have already been thread-parallelized, such as the grain boundary identification and grain boundary nucleation, and hence these functionalities are claimed as disfunctional at the moment**.

9.1 Further details for developers.

In contrast to the pure MPI version in which each automaton domain expands completely in one implicit array (the **mycellgrid**), the hybrid version subdivides statically this grid in the y and the z direction. Each such automaton region is now handled by the **caregionMemHdl** class which allocates storage and bookkeeps two interface lists. According to this list, cells are either categorized as located completely in the simulation domain or close to its boundary, which is a one cell thick layer.

The cells that are close to this boundary require synchronization with other regions to avoid data races and to maintain a consistent view of the infection states. For this **halo** regions are defined, into which the threads first infect in parallel, then synchronize **syncHalo** and again read out in parallel the changes that occurred in the halo regions and require synchronization in the corresponding memory regions. This minimizes thread synchronization.

At the moment the cell list defragmentation functionality is disabled.

CHAPTER 10

Licence

The project is licenced under the GNU v2.0

CHAPTER 11

Questions, contributions

Just let me know or contact *markus.kuehbach@rwth-aachen.de*