Constructing 3D grain shapes from a given 3DXRD experimental data

By Hamidreza Abdolvand

Department of materials, university of Oxford, Parks Road, Oxford, OX1 3PH, UK

Email: hamid.abdolvand@gmail.com

hamidreza.abdolvand@materials.ox.ac.uk

A brief report on constructing 3D grain shapes based on a given 3DXRD experimental data is given here. The developed code is a Fortran file that can be run as a standalone code or can be linked to MATLAB.

1. Mathematics:

A code is developed to implement a weighted Voronoi tessellation (also known as Laguerre tessellation) technique [1, 2] to generate 3D grain geometries based on the center-of-mass position and volume of the grains measured in a 3DXRD experiment. The implementation of this code for modeling deformation in Hexagonal Close Packed polycrystals is explained in references [3-5]. In general, for a given set of n sites $p_i = (s_i, w_i), i = 1{:}n$ where $s_i \in R^d$ is a seed point and $w_i \in R$ is a weight, a weighted Voronoi tessellation is a convex subdivision of d-dimensional space into cells C_i fulfilling [1]

$$C_i = \left\{X \in R^d \middle| \|X - s_i\|^2 - w_i < \|X - s_j\|^2 - w_j, i \neq j\right\} \quad (1)$$

where w_i can be squared-radius of each grain which is measured in the 3DXRD experiment. If the center-of-mass $\mu_a$of a grain, say G_a, is known, it is possible to estimate the seed point (s_a) for the grain G_a using Eq. 2,

$$s_a = 2\mu_a - \mu_b \quad (2)$$

where μ_b is the center-of-mass position of the grain $G_b$that is constructed by setting its seed point position $s_b$equal to μ_a. In order to show how good grain shapes and neighbors can be captured using this method, the developed code is used for simulating grain shapes that were measured using electron backscatter diffraction technique (EBSD) and results are shown in Fig. 1.

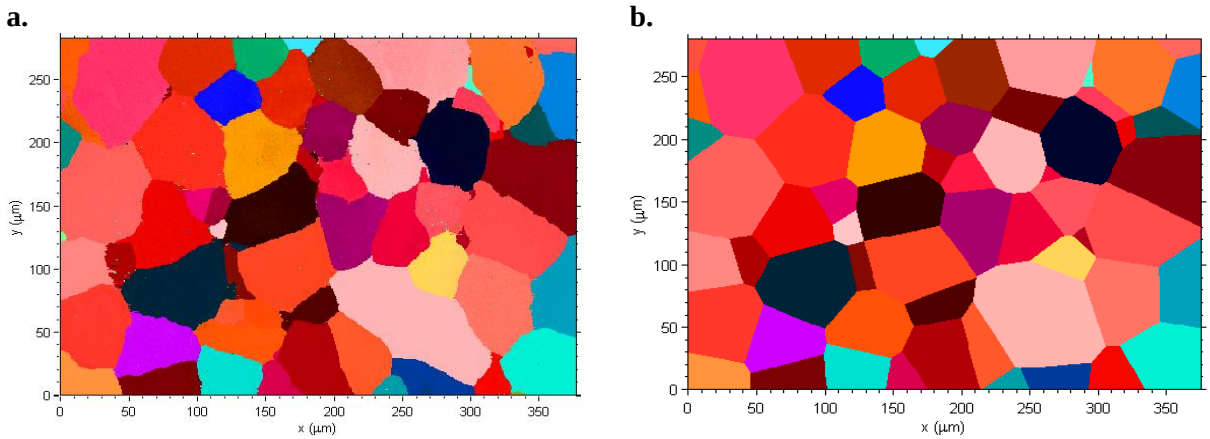

Fig. 1. (a) an EBSD map of the material used for a 3DXRD experiment at the ESRF (MA-1213); (b) Weighted Voronoi estimation of the EBSD map show in (a); same color coding is used [4].

2. The code

Two codes are provided: a Matlab code (grain_generator.m) and a Fortran code (RevisedVoronoi). Generally the user do not need to modify the Fortran code; running grain_generator produces the files required for constructing 3D microstructure.

In order to run the code, user need to have a file containing following information (with the indicated order):

1: grainno; 2: mean_IA; 3: grainvolume; 4: x-position (in mm); 5: y-position; 6: z-position; 7: rodx- (in rodrigues); 8: rody; 9:rodz; 10: U11; 11:U12; 12: U13;13:U21; 14: U22; 15: U23; 16: U31; 17: U32; 18: U33; 19: eps11;20: eps22; 21: eps33; 22: ;eps23; 23: eps13; 24: eps12; 25: eps11_s;26: eps22_s 27:eps33_s; 28: eps23_s ; 29: eps13_s; 30: eps12_s; 31: sig11; 32: sig22; 33: sig33; 34: sig23; 35: sig13; 36: sig12; 37:sig11_s; 38: sig22_s; 39: sig33_s; 40: sig23_s; 41:sig13_s; 42: sig12_s;

In order to construct a 3D volume, first the dimension of the volume to be simulated must be given. Let's say a volume of -410<x<410, -400<y<400, -520<z<520 is probed in a 3DXRD experiment (dimensions are in microns). Normally it is required to cut the grains at the edges as the provided code might not be able to assign a proper shape to those grains. Since probed volume will be discretized in the code, it is required to define how fine or coarse the volume needs to be discretized. With fine discretization, a better approximation of grain boundary will be achieved in expense of longer simulation time.
The parameters that are required to be defined in "grain_generator.m" for simulation are as follows (see fig. 2):

nstpsz:	a parameter that defines how coarse or fine the selected volume must be discretized. Smaller number results in longer simulation but better approximation of grain boundaries. Note: it is necessary to have "nstpsz" a natural multiplication of the size of the selected volume, i.e. mod((xfin-x0), nstpsz) must be zero (also for y and z directions)
x0	the starting point in the x-direction for simulating the 3D micro-structure.
xfin	the final point in the x-direction for simulating the 3D micro-structure.
y0	the starting point in the y-direction.
yfin	the final point in the y-direction.
z0	the starting point in the z-direction.
zfin	the final point in the z-direction.
ngus	an estimation of the number of grains that fall into the selected volume. If it is less than 10000, you do not need to change this value; however for higher number of grains, it is necessary to change this in the source FORTRAN file.
WVORN*	a parameter that defines if Voronoi tessellation (0) should be used for simulating microstructure or the weighted Voronoi tessellation (1).
NameOfFile	This is the name of the file containing 3DXRD information; the data in the file should follow the same configuration as given above. Format of the file should be (format (42(f16.5)))

* Note: in order to construct the weighted Voronoi tessellation, the radius of each grain is required (Eq. 1). In the program, radius of each grain is calculated using the given volume from 3DXRD and the actual physical volume calculated from the center of the mass of grains- assuming that grains are space filling and there is no missing grain or porosity in the volume. This assumption can be easily changed in the Fortran code if it is required (search for "normalize the measured volume" in the Fortran file).

```
%% Program parameters
clear;
nstpsz=2; % mesh size resolution: This identifies the step size used to mesh the volume. Higher value results in better resolution but sl
% note that the difference between two upper baound and lower bound should
% be a natural multiple of nstpsz, i.e.:: mod (xfin-x0,nstpsz)=0 same for y
% and z
%%%% this is the size of the volume that you want to study.
x0=-200;   % smallest value in the x-direction (it can be used to cut grains at the edges)
y0=-200;   % smallest value in the y-direction
z0=-200;   % smallest value in the z-direction
xfin=200;  % biggest value in the x-direction
yfin=200;  % biggest value in the y-direction
zfin=200;  % biggest value in the z-direction
ngus=10000; %estimated number of grains that fall into this volume--- (use an upper bound number)
WVORN=0;    % a parameter for using voronoi or weighted voronoi: 0 for Voronoi; 1 for weighted Voronoi.
NameOfFile='esrf5n.dat';  % this is the name of the file that has COM, Volume, Rodrigues, etc data.
                          % The order of the data here should be the same
                          % as the one given above.
```

Fig. 2. Required parameters for constructing 3D micro structure.

3. Outputs

3.1 GRAI001.dat

This is the main file that contains the position of each voxel in 3D space, the grain that each voxel belongs to, and the Euler angles associated with each voxel.

Remark: Euler angles are given following Bunge notation for transforming crystal coordinate to the sample (lab) coordinate. This is the transpose of those given in Rodrigues space.

An example of the headers of the GRAI001 file:

Grain ID (consistent with 3DXRD input file)	**X position of the voxel**	**Y position of the voxel**	**Z position of the voxel**	φ_1	φ	φ_2
713	**-100**	**26**	**200**	**126.052**	**328.93**	**251.95**

Euler angles are in degrees and positions are in microns.

3.2 grain_euler.dat

This file contains the ID of each grain and the corresponding Euler angles (following the Bunge notation as explained above)

3.3 grain_euler2.dat

This file contains the grain ID, Euler angles of each grain, grain measured volume (from 3DXRD), and the estimated grain diameter (to check if grain diameters are calculated properly)

An example of the headers of the grain_euler2.dat file:

Grain ID	**Grain ID (check point for program)**	φ_1	φ	φ_2	**Grain measured volume**	**Grain estimated diameter**
713	**713**	**126.052**	**328.93**	**521.95**	**52.69**	**0 (if weighted Voronoi is off) Or a value of it is on**

3.4 neighbours.dat

This file contains information about which grain is neighbor of which. It can be used for statistical analysis of local neighborhood or grain-grain interaction (see for instance [4, 5]). The order of data in this file is as follows: grainID, number of neighbors that the current grain has, ID of the neighboring grain.

Example-1:

713 3 4208 828 970

It means grain number 713 has 3 neighbors in the simulated volume which are 4208, 828, and 970. Grain IDs are consistent with the 3DXRD input file.

Example-2:

815 5 1058 4117 810 4080 1059

It means grain number 815 has 5 neighbors which are grains 1058, 4117, 810, 4080, and 1059.

Remark: please be cautious in using grains that are located at the edge of the volume as the number of estimated neighbors of those grains might not be right (the searching subroutine looks for neighboring grains that fall into the simulated volume not the measured volume.)

4. Visualization

For visualization the Matlab code uses MTEX software (version 3.5) for constructing 3D microstructure based on data given in GRAI001.dat. This subroiutine is originally developed for analyzing 3DEBSD data. If you don't have MTEX, you will get error in MATLAB. You can download MTEX, version 3.5 from the following lik:

http://mtex-toolbox.github.io/download.html

Pole figure is another output of the grain_generator. Please note that if you want to have a sensible plole figure, it is recommended to use more grain; hence, for quick polefigure check, you can somulate the whole probed volume using bigger step size (nstpsz).

Example-1:

In this example, results from experiment MA-1213 performed at ESRF [4] is used for further analysis; in this experiment a volume covering -369<x<414, -406<y<407, -478<z<470 μm was probed. In this example we will construct the grains that fall into the volume made by a cube -200<x,y,z<200 μm. The used step size is 2 micron which is quite small and is computationally expensive, but it gives a better estimation of grain boundaries.

Please note that the 3DXRD data must be in the same directory as where grain_generator.m is (so as the RevisedVoronoi.exe) file.

Parameters used for simulation:

nstpsz=2;

x0=-200;

y0=-200;

z0=-200;
xfin=200;
yfin=200;

zfin=200;
ngus=10000;
WVORN=0;
NameOfFile='esrf5n.dat';

Results:

a. b.

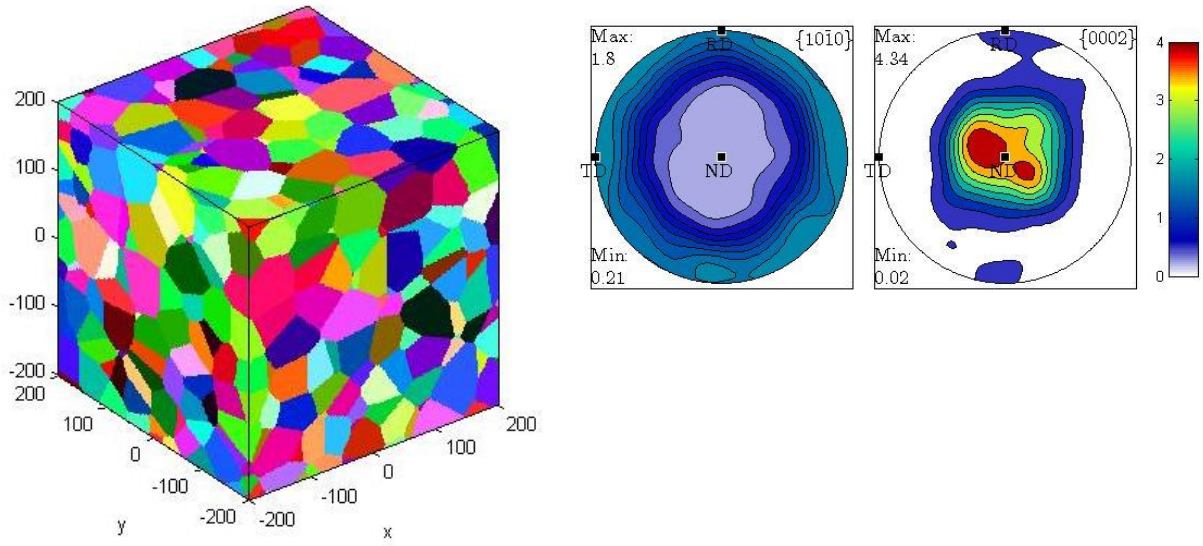

Fig. 3. (a) A simulated micro-structure of MA-1213 experiment and (b) the corresponding pole figure

Example-2

In this example, results from experiment MA-1213 performed at ESRF is used for further analysis; we will construct the grains that fall into the volume made by a cube -400<x,y,z<400 μm. The used step size is 10 micron which is quite big, but it gives a good quick statistical analysis of 3DXRD data.

Parameters used for simulation:

nstpsz=10;

x0=-400;

y0=-400;

z0=-400;
xfin=400;
yfin=400;
zfin=400;
ngus=10000;
WVORN=0;
NameOfFile='esrf5n.dat';

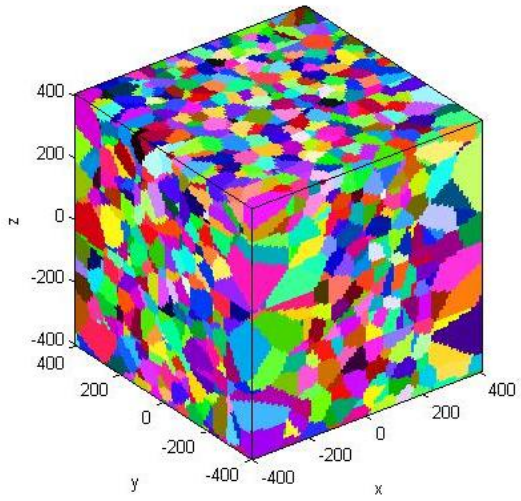

Fig. 4. A simulated micro-structure of MA-1213 experiment for a volume that covers (-400<x,y,z<400)

Please note that the front grains are not modeled properly as in MA-1213 experiment not all of the sample immersed in beam (beam size was smaller than the sample width); that is why Voronoi simulation for grains close to surface is not really reliable. However, one can avoid having this artifact by using a smaller volume (not covering y>350).

This kind of simulation (big nstpsz on the whole probed volume) will give a better understanding of the sample average texture.

Example-3

In this example, results from experiment MA-1213 performed at ESRF is used for further analysis; we will construct the grains that fall into the volume made by a cube -100<x,y,z<100 μm. The used step size is 2 micron and we try to check the effect of using grain volume on the simulated micro structure.

Parameters used for simulation:

nstpsz=2;

x0=-200;

y0=-200;

z0=-200;

xfin=200;
yfin=200;
zfin=200;
ngus=10000;
WVORN=0; for fig. 5a and WVORN =1 for fig 5b.
NameOfFile='esrf5n.dat';

a. b.

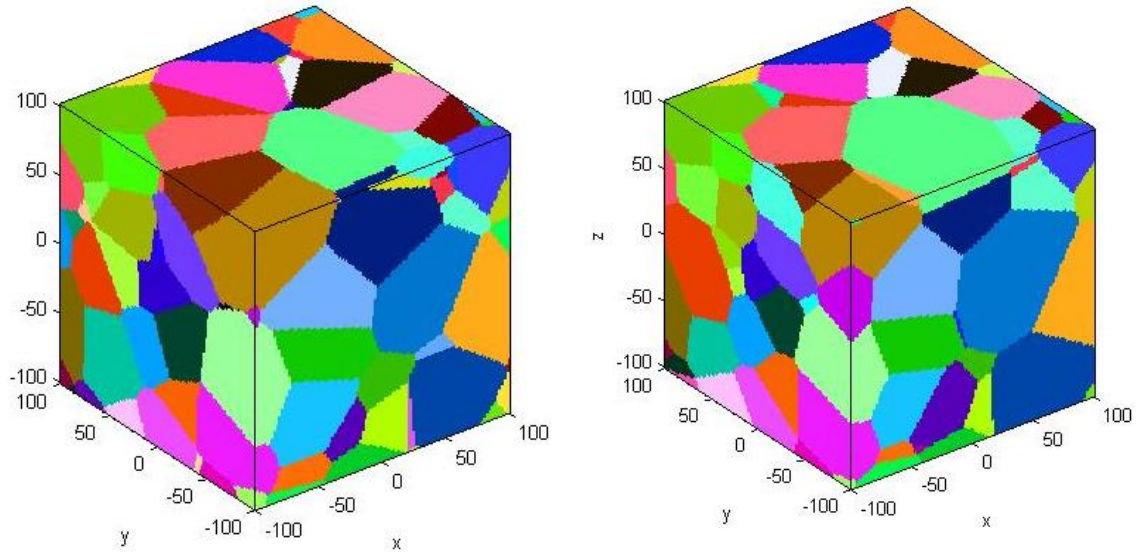

Fig. 5. (a) A simulated micro-structure of MA-1213 experiment with Voronoi tessellation and (b) weighted Voronoi tessellation.

5. Further analysis

File GRAI001.dat contains the position of each voxel, grain ID of each voxel, and the associated Euler angles of each voxel. This data can be used for constructing 3D input file for finite element packages such as Abaqus [6] to model stress distribution within each grain. Author has developed such files and if the reader need to have access to them, please do not hesitate to contact Hamidreza Abdolvand.

6. How to run the Files

Copy and extract all of the files into one unique folder. Please make sure that following files are in the same folder:

grain_generator.m; RevisedVoronoi.exe; and the input file from 3DXRD experiment.

Please make sure that the input file from 3DXRD experiment has the same format as it is used in esrf5n.dat (format (42(f16.5))).

Open "grain_generator.m" file and set the input parameters (see Fig.2) and simply run the matlab file.

For installing MTEX:

Go to following webpage:

http://mtex-toolbox.github.io/download.html

download and extract MTEX V3.5 into toolbox directory of MATLAB. Change the current directory in matlab into MTEX directory and run "startup_mtext.m" file. This will install MTEX on your machine but not permanently. If you want to install MTEX permanently you can run "install_mtex.m" file. Once you have MTEX running in matlab, all of the command in "grain_generator" must work without any problems.

7. In case you have problem

Please don’t hesistate to contact Hamidreza Abdolvand.

8. Refrences

[1] Lyckegaard A, Lauridsen EM, Ludwig W, Fonda RW, Poulsen HF. Advanced Engineering Materials 2011;13:165.
[2] Okabe A, Boots B, Sugihara K. Spatial tessellations: concepts and applications of Voronoi diagrams. New York: John Wiley & Sons, Inc., 1992.
[3] Abdolvand H, Majkut M, Oddershede J, Wright JP, Daymond MR. Acta Materialia 2015;93:235.
[4] Abdolvand H, Majkut M, Oddershede J, Wright JP, Daymond MR. Acta Materialia 2015;93:246.
[5] Abdolvand H, Majkut M, Oddershede J, Schmidt S, Lienert U, Diak BJ, Withers PJ, Daymond MR. International Journal of Plasticity 2015;70:77.
[6] Abaqus. Standars user's manual version 6.5: Habitt. Karrlsson & Sorensen, Inc., 2005.