TECHNISCHE UNIVERSITÄT
BERGAKADEMIE FREIBERG

Die Ressourcenuniversität. Seit 1765.

PERSONAL PROGRAMMING PROJECT

Big Data Visualization of Large and Ultra-Large Scale Atomistic Simulations (with>10Mio atoms)

Padmanabha Pavan Chandra Vundurthy
Computational Materials Science
Matriculation Nr.**62750**

Under the Supervision of:
Dr.-Ing. Arun Prakash
Micromechanical Materials Modelling

ACKNOWLEDGEMENTS

I am grateful to Dr.-Ing. Arun Prakash, for his wonderful supervision throughout the project. His constant motivation and careful guidance play a crucial role in completion of this project. He always encouraged me for an "out of the box" thinking, which helped me a lot in approaching things differently.

ABSTRACT

Because of the anisotropic behaviour of the polycrystalline and the existence of grain to grain interactions, the stress calculated by Hooke's Law would not be too reliable using the lattice strains.The stress/strain states of the material are obtained by averaging over all stresses in the constituent grains with different orientations.With knowledge of the average stress state tensor in each grain (which can be significantly different from the global stress state tensor, the resolved shear stress (RSS) on the corresponding activated slip system can be directly calculated and directly identified as the CRSS value. For these reasons, an analysis for properties of the grains in a polycrystal is proposed. This includes an analysis of over 20 million atoms for the average properties like average stress tensor, orientation and other required quantities.

In this Personal Programming Project, the big data analysis of large and ultra large scale atomistic simulations which analyses different property parameters and assembles the average properties of a grain in a polycrystal for different structure types using FORTRAN programming language. The project also extends over to Visualization of the analysis using PYTHON programming language. The visualizations depict the variations in average volume of grains in a conglomerate(nearest neighbour grains) and average stresses with respect to average volume and average stress in the polycrystal with the help of polar plots.

MOTIVATION

Big Data Analysis is the future of Materials Science. It changed the way researchers conduct their work, analyse properties and trends in their data, and even discover new materials.The ease of generating atomic structure by performing atomistic simulations and using it again and again for different analysis is the key advantage of Data Science in Material science.MGI and DSI have revolutionised the method of research in Materials Science.Big-data analytics tools plays a crucial role in searching for meaningful patterns in the large amounts of data. Big data unifies the three paradigms of theory, experiment, and computation/simulation.In comparison to data encountered in other fields, Materials science data tend to be particularly heterogeneous in terms of their type and source. So, processing the data is relatively complex than generating it.By using existing dataset of large and ultra large atomistic simulations , extracting and processing the data and then extending it for identifying the micromechanical properties like CRSS, Local Stress/Strain states in a grain in a Polycrystalline Material helps in improving material modelling techniques for further research.

Contents

1 INTRODUCTION

1.1 BIG DATA:WHY, WHAT AND HOW?

In the current scenario of rapid advancement in digital technologies, data is generated from various sources and the fast transition has led to the growth of Big Data. Data science aims at researching big data and knowledge extraction from data. Big Data Analytics has an evolutionary breakthrough in many fields like banking, agriculture, chemistry and marketing with the collection of large datasets. Digitalization is the key concept and foundation for the enhancement of Big Data which is beyond the traditional processing database management techniques. The three Vs (Volume, Variety and Velocity) are three defining properties of Big Data referring to the amount of data, number of types of data and speed of the data respectively. Two more Vs have emerged over the past few decades, namely, Value and Veracity referring to the quality of the data and its reliability.
According to Oracle, benefits of Big Data and Data Analytics are :

- Big Data makes it possible for gaining complete answers with the availability of quality information.
- More complete answers reflect more confidence in the data and provides a reliable approach to tackling problems.

The crucial challenge for big data analysis techniques is its scalability and security.The major challenge in this case is to pay more attention for designing storage sytems and to elevate efficient data analysis tool that provide guarantees on the output when the data comes from different sources. Furthermore, design of machine learning algorithms to analyze data is essential for improving efficiency and scalability.Big data analytics and data science are becoming the research focal point in industries and academia. [1–4]. Although the initiation of the concept of Big Data ("information explosion") and its research started a few decades ago, its applications and benefits have just begun.

1.2 DATA SCIENCE IN MATERIALS SCIENCE

Materials Science is the field which depends on experimental observations and the simulation results to understand the behaviour of different models. Big Data being generated by experiments and simulations of Materials Science in its approach to understand different characteristics, has lead to many opportunities in the field of Data driven techniques supporting Materials Genome Initiative(MGI). As mentioned in [5, 6], the Materials Genome Initiative(MGI) has emphasized the need for data informatics for further upcoming research in the emerging fields of materials informatics, identifying the human ability to collect data "Big Data". The budding of materials informatics has lead to Data-Driven techniques which play a crucial role in estimating the processing-structure-property-performance relationships in materials. Its forward models(property prediction) and inverse models(materials discovery) utilize the current areas of regression techniques and methods of Machine Learning creating a major research and development area.
According to [6], the standardization of data formats and interfaces, integrated workflow tools, training on best practices for software development can lead to major digitalization in materials science which enables a researcher to focus more on analysing the data and study the behaviour of models rather than spending a lot of time in generating the data.

1.3 DRIVING CONCEPT IN A NUTSHELL

In a single crystal, the physical and mechanical properties often differ with orientation. It can be observed from looking at a model of crystalline structure, that atoms should be able to slip over one another or distort in relation to one another easier in some directions than others. Slip generally

occurs in closed packed planes, those containing greatest number of atoms per length. A slip plane and slip direction constitute a slip system. Critical resolved shear stress (CRSS) is the component of shear stress, resolved in the direction of slip, necessary to initiate slip in a grain. Resolved shear stress (RSS) is the shear component of an applied tensile or compressive stress resolved along a slip plane that is other than perpendicular or parallel to the stress axis. The RSS is related to the applied stress by a geometrical factor,m,typically the Schmid factor. The *Taylor* factor has been shown to be more accurate for polycrystal metals.

A polycrystalline Representative Volume Element (RVE) is a material piece which must include a sufficient number of grains, in terms of crystallographic orientation, shape and size. The unique stress and strain tensor would represent the material. The local analysis of the stress and strain field is the only one which preserves both the local behaviour and the local load balance, and which can give a realistic evaluation of the heterogeneity inside the material. As the stress state in each grain is not necessarily equal to the macroscopic applied stress, this introduces variability into the conditions in which slip occurs. With the knowledge of the average stress tensor in each grain (which can be significantly different from the global stress tensor), RSS can be directly calculated and identified as CRSS value.

Hence, the analysis of average properties of grains based on their structure type enables the user to estimate the stress concentration which will activate dislocation motion in the available glide planes. These dislocations are geometrically necessary to ensure that the strain in each grain is equivalent at the grain boundary, so that the compatibility criteria are satisfied and also to estimate onset of plasticity in polycrystals.

2 GROUND WORK

2.1 COMPARISON OF SPEED

In order to estimate the performance efficiency of programming languages in reading Big Data files, a priliminary comparison study is done between FORTRAN, PYTHON(Pandas using python engine) and Cython for a task. The task includes:

- opening Big Data Files of various sizes.
- reading and storing the data.
- appending the time elapsed for opening and reading each file
- a comparison plot is drawn between Number of Lines and Time elapsed using each language.

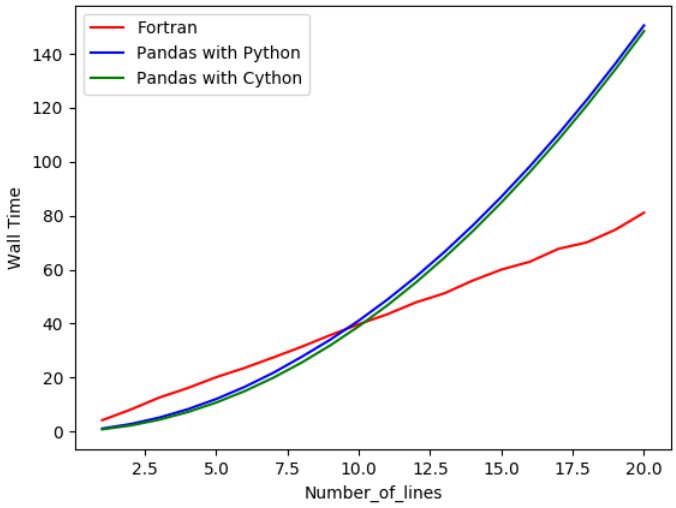


Figure 1: Comparison of Speed

Input Files: *python_pandas_func.py, cython_pandas_func.pyx, read_array2d.f90*
Output File: cython_pandas_timeline.dat,python_pandas_timelines.dat,timelines.dat,plot_py_cy_fo.py
Command:$ gfortran -o <read_array2d.f90>
Inference:The Comparison of speeds can be observed in the fig.1 for Number of lines in a file vs Wall Time. With expertise, a User can alter the speeds of a program which can help reduce the computational cost of the program. Parallel Computing can be another alternative to increase the processing speed of any program.

2.1.1 DESIGNING THE PARAMETER FILE

The most important feature here is the *parameter file* which is designed in a such a way that it provides a wide variety of options to the User. The user, instead of being controlled by the options provided by the programme, has complete authority over the programme via the *parameter file*. In general, Big Data programmes being heavily time consuming and costly, cannot be repeatedly analysed whenever data is required.It is highly effective if the User extracts all the information needed from a single analysis of the Big Data file. The Paramater File fulfils this purpose.

```
1   # Number of Files to be Read
2   2
3   # File Name  ||  Number of rows || Number of columns    || Number of header lines
4   Sorted_GrNum_00200.txt  18632909    11  8
5   Sorted_Stress_00200.txt 18632910    13  9
6   # Data Info: Number of Columns  ||  File Number ||  Column Number in the File
7   # User:Please specify the Number of Parameters required from the files:
8   6
9   atomID  1  1  1
10  Coord 3  1  3
11  struct_type  1  1  10
12  GrNum  1  1  11
13  Stress  6  2  6
14  Volume  1  2  13
15  # User: Please specify the Output filename into which the analysis is to be written:
16  Out_Average_Properties.txt
17
18
```

Figure 2: Parameter File

File Name: *<Parameter_File_BigData.txt>*

Inference:All the required properties of a User can be analysed just by altering the data of Parameter File without accessing the program(User Friendly). In the fig.2, the following properties can be observed:

- **Number of Files to be Read**:In line 2, the user can specify N number of Data files(No Limit) from which the programme should read certain parameters.
- **File Name**: From line 4 to line (4+N), the User needs to mention the names of the data files.
- **File atrributes**:Along with the file names, the attributes of the files like Number of rows, Number of columns, Number of header lines are specified; each separated by a 'tab'.
- **Number of parameters**:In line 8, the User can specify the number of required/interested parameters(unlimited).
- **Parameters**: From line 9 and below the user can specify the required parameters with appropriate Tags which can be used later for 'tag check' while running the programming.The Parameter Tag should NOT be changed.It means 'atomID' should remain same and any other name like 'ATOMID' cannot be used.
- **Parameter Attributes**: Alongside the Parameter names, their attributes like Number of columns(it occupies in the file), File number(w.r.t the sequence of filenames mentioned) and Column number(it's start column in the file) should be specied; each separated by a 'tab'.
- **Output File name**: In line 16, the name of the output file into which the Average Properties of the Polycrystal after analysis should be written is to be specified.

2.2 COMMON NEIGHBOR ANALYSIS

While observing the Grains and their properties in a polycrystalline material via OVITO using the supplied Big Data file, the common neighbour analysis and color coding is performed and the Neighbor Grains are noted down manually in fig.3. The grains and their corresponding neighbours are stored in a text file with grain number being the starting number and separated by a 'tab' in between each other.

```
24    24  19  23  66  68  16  25  42  8   15
25    25  30  75  94  19  24  42  26  4
26    26  4   25  42  38  39
27    27  36  52  102 73
28    28  43  91  70  52  68
29    29  51  6   18  83  112 106 94
30    30  60  75  25  4   96
```

Figure 3: Nearest Neighbor Grains

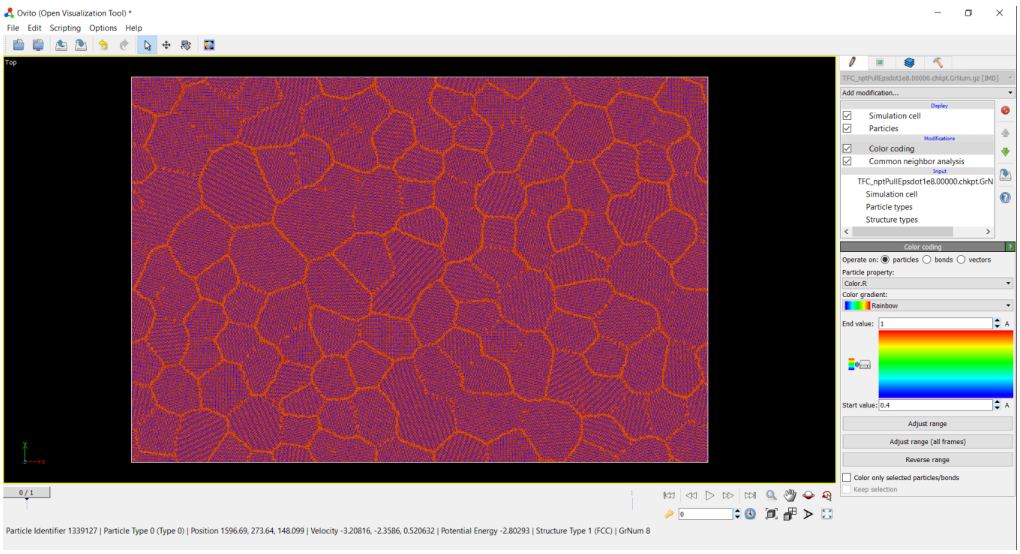


Figure 4: Common Neighbor Analysis

File Name: *<Neighbours.txt>*

According to CNA [7, 8], which pairs of particles are considered nearest neighbors (bonded) is determined by the cutoff radius parameter. Particles closer to each other than the cutoff radius are considered bonded. For face-centered cubic (FCC) and hexagonal close-packed (HCP) structures the cutoff radius must lie midway between the first and the second neighbor shell of neighbors. For body-centered cubic (BCC) materials the cutoff radius should lie between the second and the third neighbor shell. OVITO provides a list of optimal cutoff distances for fcc and bcc crystal structures formed by common pure elements. The coordination structure types are encoded as integer values:

- 0 = Other, unknown coordination structure
- 1 = FCC, face-centered cubic
- 2 = HCP, hexagonal close-packed
- 3 = BCC, body-centered cubic
- 4 = ICO, icosahedral coordination
- 5 = DIA, cubic diamond structure

3 IMPLEMENTATION OF ALGORITHM

3.1 PART 1: BIG DATA ANALYSIS USING FORTRAN

3.1.1 SORTING THE DATA FILE

The primieval task before working with large data files is to make the files ready for analysis.The data files need to be sorted with respect to their atomic_id number. Two important prerequisites for the programme to work perfectly fine is that:

- Each data file should contain 'Atom_ID' number in their first column irrespective of any data in the other available columns
- Each data file needs to be sorted based on their Atom_Id (first)column.

Command:$:*sort -n <Input File> <Output File>*
Input File$:<TFC_nptPullEpsdot1e8.00000.chkpt.GrNum.txt>
Output File$:<Sorted_TFC_nptPullEpsdot1e8.00000.chkpt.GrNum.txt>

3.1.2 READING THE PARAMETER FILE

The first step in analysing a Big Data file is to know what are the requirements/interests of the User. The parameter file, as mentioned in fig.2, is read and the source information of the files(Number of Files and their attributes), Parameters(Number of parameters and their attributes) are stored.

```
!***OPEN AND READ THE PARAMETER FILE FOR THE USER INPUT***!
open (unit=201, file='Parameter_File_BigData.txt', iostat = error)
if (error .ne. 0) then
        write(*,*) "file cannot be opened"
end if
read(201, *)
read(201,*)num_files
print*,"Number of Files==",num_files
read(201, *)
allocate(filename_arr(num_files))
allocate(file_attributes_arr(3,num_files))
!READ FOR STORING THE FILE NAMES AND FILE ATTRIBUTES
a = 0
do while(a<num_files)
    read(201,*)filename_arr(a+1), file_attributes_arr(:,a+1)
    a = a+1
end do
a = 1
do a = 1,num_files
    print*,"filename::==",filename_arr(a)
    print*, "file attributes::==",file_attributes_arr(:,a)
end do
read(201,*)
read(201,*)
read(201,*)num_props
!!ALLOCATE THE ALLOCATABLE ARRAY DIMENSIONS FROM THE INPUT READ FROM PARAMETER FILE
allocate(prop_name_arr(num_props))
allocate(prop_attributes_arr(3,num_props))
b = 0
do while(b<num_props)
    read(201,*)prop_name_arr(b+1), prop_attributes_arr(:,b+1)
    b = b+1
end do
read(201,*)
read(201,*) Output_filename     !!Specify the Output File Name
close(201)
!CLOSE THE PARAMETER FILE
```

Figure 5: Code for Reading Parameter File

FORTRAN File:$<Big_Data_Avg.f90>Ṗarameter_File_BigData.

Inference:In line 31, Parameter file is opened and the User data is stored in variables. From the fig.2 and fig.5:

- *num_files* stores the value of number of files the User wants to read.
- *filename_arr* is a 1D array which stores the names of the files inputted by User.
- *file_attributes_arr* is a 2D array which stores the attributes of each file in corresponding order of rows as file numbers.
- *prop_name_arr* is a 1D array which stores the Parameter properties in the same sequence.
- *prop_attributes_arr* is a 2D array which stores the attributes of parameter properties in corresponding order of rows as file numbers.
- *Output_filename* is a 'character' variable which reads the name of the Output file into which the Average Properties after analysis are to be written.

3.1.3 EXTRACTION OF DATA

- **Assigning Atom_ID array**:Once the attributes of the parameter properties from data files are known from the fig.5, immediate step is to extract the necessary data quantities into a location which can be used repeatedly for analysis. For extracting the data, one needs have a reference point in each file which maintains the data accuracy or its quality and extracts necessary data quantitites w.r.t the reference point. The reference point here is the '*Atom_ID Number*'. This number is the first column of each file which is used for extracting data from these files and saving into an array. The User mentions the reference 'atomID' column number from a certain file. This is obtained from the 'Parameter tag check' for '*atomID*'. Inorder to extract the '*atomID*' column the data from the file, the data file needs to be read into a 2D array '*temp_arr*'. This array is an *allocatable array* which can be reused agin later.
- **Parameter Tag Check**: Since a Big Data File is already read into an array,for the benefit of Computation Cost, a check should be condicted for the presence of any other Parameters mentioned by the User in the same file. This check is conducted by '*Parameter Tag check*' via the help of *prop_name_arr* and *prop_attribute_arr* obtained in fig.5. Firstly, the property name is checked (like*Coord*) and then their file number is verified like at line 114 and 116 in fig.6.

```
109             !EXTRACTING THE ATOM ID ARRAY FROM TEMPORARY ARRAY
110             atomid_arr = temp_arr(1,1:file_attributes_arr(1,atomid_filenum)-file_attributes_arr(3,atomid_filenum))
111             iprop_arr(1,:)=atomid_arr       !ASSIGNING ATOM_ID ARRAY TO THE FIRST COLUMN OF IPROP ARRAY
112             !!!***START CHECKING FOR PARAMETER TAGS***!!!
113             do while (d<=num_props)
114                 if (prop_attributes_arr(2,d)==atomid_filenum)then
115                 !**-----------------------------------------**!
116                     if (trim(prop_name_arr(d))=='Coord') then
117                         iprop_arr(2:4,:)=temp_arr(prop_attributes_arr(3,d):prop_attributes_arr(3,d)+2,:)
118                         !print*,"=========", iprop_arr(1:4,:)
119                     else if (trim(prop_name_arr(d))=='struct_type') then
120                         iprop_arr(5,:)=temp_arr(prop_attributes_arr(3,d),:)
121                         !print*,"=========", iprop_arr(1:5,:)
122                     else if (trim(prop_name_arr(d))=='GrNum') then
123                         iprop_arr(6,:)=temp_arr(prop_attributes_arr(3,d),:)
124                         !print*,"=========", iprop_arr(1:6,:)
125                     else if (trim(prop_name_arr(d))=='Stress') then
126                         iprop_arr(7:12,:)=temp_arr(prop_attributes_arr(3,d):prop_attributes_arr(3,d)+5,:)
127                         !print*,"=========", iprop_arr(1:12,:)
128                     else if (trim(prop_name_arr(d))=='Volume') then
129                         iprop_arr(13,:)=temp_arr(prop_attributes_arr(3,d),:)
130                         !print*,"=========", iprop_arr(1:13,:)
131                     end if
132                 !**-----------------------------------------**!
133                 end if
134                 d = d+1
135             end do
```

Figure 6: Parameter Tag Check

- **Assembling an array of desired quantities**: The extracted *atomid_array* and other quantities, if any, in the same file are assembled in the *iprop_arr*. It is a 2D array which assembles all the necessary quantities in different columns. AT line 126, in fig.6 it can be observed that *Stress* is extracted from a file and is assembled in it from 7 to 12 columns.

- **Properties in other Files**:At line 18, in fig.6, looping over files other than the one containing *atomid_arr* for properties and extracting them into *iprop_arr*.At line 244, in 7The speciality of the program is its feature called *counter*, which speeds up the iteration by terminating the program after extracting the quantities w.r.t *atomID*, instead of looping over the entire file even after extracting the data.

```
235                 !*CHECK FOR STRESS*!
236                 else if (trim(prop_name_arr(f))=='Stress') then
237                     g = 1
238                     counter = 1
239                     do while(g<=size(atomid_arr))
240                         h = counter
241                         do while(h<=span)
242                             if (atomid_arr(g)==temp_arr(1,h))then
243                                 iprop_arr(7:12,g)=temp_arr(prop_attributes_arr(3,f):prop_attributes_arr(3,f)+5,h)
244                                 counter = h !COUNTS THE NO. OF LINES READ AND STARTS FROM THIS LINE IN THE NEXT ITERATION, SAVING TIME
245                                 goto 65
246                             end if
247                             h = h+1
248                         end do
249 65 continue
250                         g = g+1
251                     end do
```

Figure 7: Assembling iprop array

- **THREE DIMENSIONAL AVERAGE PROPERTIES ARRAY**: Looping over the size of the *atomID array*, level1 is considered as the structure type, Columns as Average Properties like *Coordinates,Stresses,Grain Volume, Number of atoms, Von Mises Stress* and Rows as Grain Numbers.The structure type ranges from 0,1,2,..Maximum structure type in the file and additionally 3 more variants like *fcc+hcp, fcc+hcp+other defects, All structure types.* Therefore, the size of array *avg_props* is *(Maximum Structure type in file+4,12, Number of Grains in the crystal)*. It is +4 instead of 3 because Structure Type start from 0, which is other defects, and FORTRAN indexing starts from 1.In fig.8, the highly efficient 3D array with intricate operations can be observed. At line 315, the Summation of volume over a grain number can be observed and the values are saved in the 10 column of the *avg_prop_arr*.

```
309     !!**LEVEL 1 = STRUCTURE TYPE; COLUMNS = AVERAGE PROPERTIES, NUMBER OF ATOMS, GRAIN VOLUME, VON MISES STRESS; ROWS = GRAIN NUMBER**!!
310     q = 1
311     n = 1
312     do q = 1,size(atomid_arr)
313         avg_props(int(iprop_arr(5,q))+1,4:9,int(iprop_arr(6,q)))= &
314         & avg_props(int(iprop_arr(5,q))+1,4:9,int(iprop_arr(6,q))) + iprop_arr(7:12,q)*iprop_arr(13,q)  !STRESS * VOLUME
315         avg_props(int(iprop_arr(5,q))+1,10,int(iprop_arr(6,q))) = &
316         & avg_props(int(iprop_arr(5,q))+1,10,int(iprop_arr(6,q))) + iprop_arr(13,q)     !SUMMATION OF VOLUME FOR A GRAIN
317         avg_props(int(iprop_arr(5,q))+1,1:3,int(iprop_arr(6,q))) = &
318         & avg_props(int(iprop_arr(5,q))+1,1:3,int(iprop_arr(6,q))) + iprop_arr(2:4,q)   !SUMMATION OF COORDINATES FOR A GRAIN
319         avg_props(int(iprop_arr(5,q))+1,11,int(iprop_arr(6,q))) = &
320         & avg_props(int(iprop_arr(5,q))+1,11,int(iprop_arr(6,q))) + 1   !NUMBER OF ATOMS IN A PARTICULAR GRAIN FOR A STRUCTURE TYPE
321     end do
322     !!**ADDING THREE ELEMENTS IN LEVEL-1: 4->FCC+HCP, 5->FCC+HCP+OTHER DEFECTS, 6->ALL STRUCTURE TYPES**!!
323     p = 1
324     r = 1
325     do p = 1,grnmax
326         avg_props(structmax+2,:,p)=avg_props(2,:,p)+avg_props(3,:,p)
327         avg_props(structmax+3,:,p)=avg_props(1,:,p)+avg_props(2,:,p)+avg_props(3,:,p)
328         do r=1,structmax+1
329             avg_props(structmax+4,:,p)=avg_props(structmax+4,:,p)+avg_props(r,:,p)
330         end do
331     end do
```

Figure 8: 3D Average Property Array

- **Writing Output File**: Once the analysis is complete, the Output is written into the Output file with filename mentioned by the user in the Parameter File. The file automatically is stored in the same folder as program. For the benefit of user, some Terminal printing facility is available which informs the user when the analysis is finished and the program is writing the output to a file and its name.

```
367     !!WRITING THE DATA TO A TEXT FILE!!
368     !For other defect structure types!
369     print*,"DEAR USER, THE AVERAGE PROPERTIES ARE CALCULATED AND BEING WRITTEN INTO A FILE"
370     !!!***OPEN THE FILE INTO WHICH THE RESULT IS TO BE STORED***!!!
371     open(210, file = Output_filename, status='new')  !!!***THE FILE NAME INTO WHICH THE DATA IS TO BE WRITTEN IS SPECIFIED***!!!
372     write(210,*)"#********************PERSONAL PROGRAMMING PROJECT*************************"
373     write(210,*)"#AUTHOR: PADMANABHA PAVAN CHANDRA VUNDURTHY(62750) (MSc.Computational Materials Science) &
374     & under the supervision of DR. ARUN PRAKASH"
375     write(210,*)"#TITLE: Big Data Visualization of Large and Ultra Large Scale&
376     & Atomistic Simulation"
377     write(210,*)"#Analysis of Average Properties of the Grain separated &
378     & in blocks of structure types: 0=Other Unknown Defects,1=FCC,2=HCP,3=BCC,...&
379     & FCC+HCP, FCC+HCP+0, All structure types."
380     do t = 1,num_files
381         write(210,*)"#file name->",t,":=",trim(filename_arr(t))
382         write(210,*)"#file attribites in sequence::==", file_attributes_arr(:,t)
383     end do
384     write(210,*)"#Macroscopic Von Mises Stress:->",Macro_VM
385     u = 1
386     o = 1
387     do u = 1,structmax+4
388         do o = 1,grnmax
389             if (o==1)then
390                 if(u==structmax+2)then
391                     write(210,*)"#======================STRUCTURE TYPES:FCC+HCP======================"
392                 else if(u==structmax+3)then
393                     write(210,*)"#======================STRUCTURE TYPES:FCC+HCP+other defects======================"
394                 else if (u==structmax+4)then
395                     write(210,*)"#======================ALL STRUCTURE TYPES======================"
396                 else
397                     write(210,*)"#======================STRUCTURE TYPE->",u-1,"======================"
398                 end if
399                 write(210,*)"#GRAIN NUMBER| COORDINATES: X Y Z | STRESS XX YY ZZ ZY XZ XY | GRAIN VOLUME | NUMBER OF ATOMS | VON-MISES STRESS |"
400                 write(210,*)"#AVERAGE VON MISES STRESS FOR THIS STRUCTURE TYPE IS",vm_struct_arr(u)
401             end if
402             write(210,*)o,avg_props(u,:,o)
403         end do
404     end do
405     close(210)
```

Figure 9: Writing the Output File

At line 371, in fig.9, the output file name read from the 'parameter file' is provided for the program to write the Output.

3.2 PART2: VISUALIZATION USING PYTHON

3.2.1 ARGPARSE MODULE

- The **argparse module** makes it easy to write user-friendly command-line interfaces. The program defines what arguments it requires, and *argparse* will figure out how to parse those out of *sys.argv*. The argparse module also automatically generates help and usage messages and issues errors when users give the program invalid arguments. Filling an ArgumentParser with information about program arguments is done by making calls to the *add_argument()* method. Generally, these calls tell the ArgumentParser how to take the strings on the command line and turn them into objects. This information is stored and used when *parse_args()* is called.

```
Implementation of Argparse Module for taking User Input from Command Line Interface
'''
parser = argparse.ArgumentParser(description='Big Data Visulization of Large and Ultra Large Scale Atomistic Simulations')
#SET: ranges from 1 to 4 types of visualization Plots
parser.add_argument('set',type=int,help='Select the set of visualization:Set1:Average Volume Polar Plots;Set2:Average Stress Polar Plots;Set3:Distribution Plots;Set4:All the plots')
parser.add_argument('Max_GrainNumber', type=int, help='Specify the Maximum grains in the polycrystal')
parser.add_argument('Max_Struct_Typ', type=int, help='Specify the Maximum Structure type in the polycrystal')
parser.add_argument('start_grnnum', type=int, help='Specify the start of grain number the user is interested in..')
parser.add_argument('iteration', type=int, help='an integer specifying the iteration of grain numbers at interest')
parser.add_argument('end_grnnum', type=int, help='Specify the end of grain number the user is interested in..')
parser.add_argument('structure_type', help='Mention the interested structure type/types to be considered for visualization\nA:Other defects\nB:Fcc\nC:Hcp\nD:Fcc+hcp\nE:fcc+hcp+other de
parser.add_argument('Avg_prop_file_name', default = [], nargs='?', type=argparse.FileType('r'), help=\
    'File containing the Average properties of grains analysed from the Big Data file')
parser.add_argument('Neighbour_Grains',default=[],nargs='?',type=argparse.FileType('r'),help='File containing the Nearest Neighbouring Grains')
#The User input are read into the 'sys.argv' list in their original format
args = parser.parse_args()
for l in range(1,7):
    sys.argv[l]=int(sys.argv[l])
```

Figure 10: Code for Parsing

In fig.10, the following arguments were passed:

- **set:**Select the set of visualization;Set1:Average Volume Polar Plots;Set2:Average Stress Polar Plots;Set3:Distribution Plots;Set4:All the Plots
- **Max_GrainNumber:**Specify the Maximum grains in the polycrystal.
- **Max_Struct_Typ:**Specify the Maximum Structure Type in the polycrystal
- **start_grnnum:**Specify the start of grain number the user is interested in..
- **iteration:**an integer specifying the iteration of grain numbers at interest
- **end_grnnum:**Specify the end of grain number the user is interested in
- **structure_type:**Mentions the interested structure type/types to be considered for visualization;A:Other defects;B:Fcc;C:Hcp;D:Fcc+hcp;E:fcc+hcp+other defects;F:all structure types
- **Avg_prop_filename:**File containing the Average properties of grains analysed from the Big Data file
- **Neighbour_Grains:**File containing the Nearest Neighbouring Grains

With the help of *sys.argv*, all the User information can be handled as in line 204.

3.2.2 PLOTTING FUNCTIONS

- *Volume_Plots:*This function is useful for plotting Polar plots depicting the Average Volumes in a conglomerate w.r.t the Average Volume in a polycrystal.The complete circle depicts the 'Average Fraction of grain in the polycrystal'; the sector angle is the Volume Fraction w.r.t neighbours in the conglomerate;the radius of the sector is the Volume fraction of grain in polycrystal; color scheme of each sector is based on the 'divergence color mapping' based on the polarized Von Mises stress values from 'blue' to 'white' to 'red'.
  Input Command:$ Volume_Plots(f,gr_vol_frac,vol_list,tot_cong_stress,nng,vol_mean)
- *Stress_Plots:*This function is useful for plotting Polar plots depicting the Average Stresses in a conglomerate. The two complete circles depict the Average stress in a Polycrystal and Average stress in the conglomerate. The angle of sector is the Volume fraction w.r.t neighbors(in the current conglomerate); Radii if the sectors is the Stress in Grains(calculated Von Mises);color scheme of each sector is based on the 'divergence color mapping' based on the polarized Von Mises stress values transition from 'blue' to 'white' to 'red' or in between them.
  Input Command$: Stress_Plots(q,gr_vol_frac,vol_list,tot_cong_stress,nng,Macro_VM, stress_in_cong,hydstress_cong)

– *Distribution_Plots:* This function plots 2 dimensional graphs for 3cases. The Grain size is calculated from its volume using equation for Volume of a sphere. Its dimensions are as specified in the supplied data file.
  * Grain size vs Polarized hydrostatic_stress, hydrostatic_stress, triaxiality.
  * Grain size vs Polarized Von Mises,Von Mises stress
  * Grain size vs Hydrostatic stress,triaxiality

Von Mises Stress in a Grain:

$$\sigma_v = \sqrt{\frac{1}{2}\left[(\sigma_{11}-\sigma_{22})^2 + (\sigma_{22}-\sigma_{33})^2 + (\sigma_{33}-\sigma_{11})^2\right] + 3\left(\sigma_{12}^2+\sigma_{23}^2+\sigma_{31}^2\right)} \tag{1}$$

Polarized Von Mises Stress in a Grain:

$$\sigma_{polarised\_VM} = \sigma_{MacroVonMisesStress} - \sigma_v \tag{2}$$

Hydrostatic Stress in a Grain

$$\sigma_m = \frac{1}{3}I_1 = \frac{1}{3}\left(\sigma_1+\sigma_2+\sigma_3\right) \tag{3}$$

Polarized Hydrostatic Stress in a Grain

$$\sigma_{polarised\_Hyd} = \sigma_{MacroHydrostaticStress} - \sigma_m \tag{4}$$

Triaxiality:The Ratio of Hydrostatic Stresses and Von Mises Stresses. The stress triaxiality is is an important parameter in explaining the geometry dependence of J-R curves.By comparing the stress triaxiality across the ligament for the specimen and the cracked component it is possible to assess whether the cracked component exhibits the similar fracture behaviour as the specimen.

$$\text{Triaxiality}, \eta = \frac{\sigma_m}{\sigma_{VM}} \tag{5}$$

$$\eta = \frac{\left(\sigma_1+\sigma_2+\sigma_3\right)/3}{\sqrt{\frac{1}{2}\left[(\sigma_1-\sigma_2)^2 + (\sigma_2-\sigma_3)^2 + (\sigma_3-\sigma_1)^2\right]}} \tag{6}$$

*Input Command:$*Distribution_Plots(r,gr_size,polarized_stress_hyd,hyd_stress,triaxiality,polarized_stress_vm,vm_s
*Final Input Command$:*visualize_data.py 1 122 2 8 0 0 D test2.txt neighbors.txt

The user input is function_1; grain number 8; structure type Fcc+hcp which can be read from the command line

4 RUN! & OUTPUT

4.1 OUTPUT OF ANALYSIS: FORTRAN

Input Filename:*Big_Data_Avg.f90*
Output Filename:*Out_Average_Properties_000.txt*
Command Used: *gfortran -o BigData Big_Data_Avg.f90*

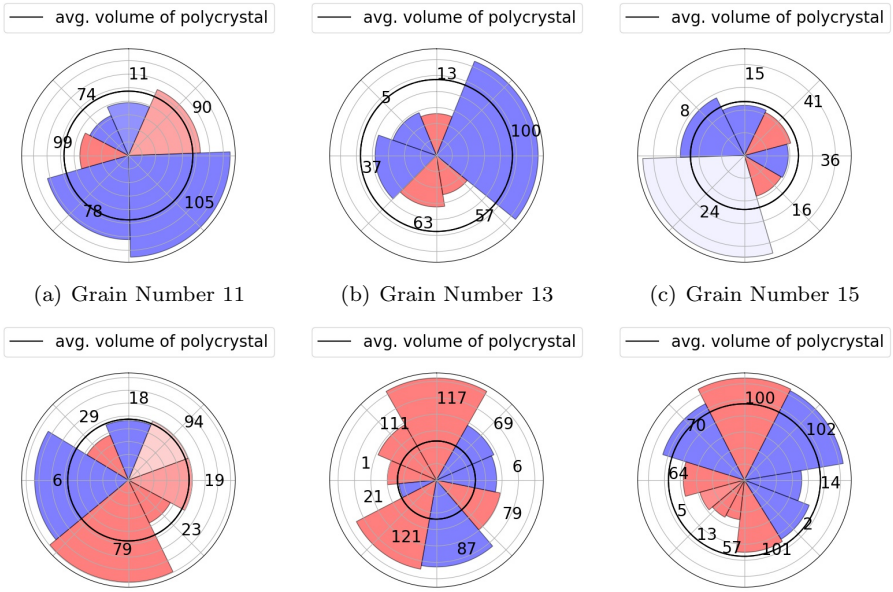
```
#**********PERSONAL PROGRAMMING PROJECT**************************
#AUTHOR: PADMANABHA PAVAN CHANDRA VUNDURTHY(62750) (MSc.Computational Materials Science)  under the supervision of DR. ARUN PRAKASH
#TITLE: Big Data Visualization of Large and Ultra Large Scale Atomistic Simulation
#Analysis of Average Properties of the Grain separated  in blocks of structure types: 0=Other Unknown Defects,1=FCC,2=HCP,3=BCC,... FCC+HCP, FCC+HCP+0, All structure types.
#file name->           1 :=Sorted_00000_GrNUM.txt
#file attribites in sequence::==    18632909          11           8
#file name->           2 :=Sorted_Stress_00001.txt
#file attribites in sequence::==    18632910          13           9
#Macroscopic Von Mises Stress:->   36.3856125
#======================STRUCTURE TYPE->        0 ======================
#GRAIN NUMBER| COORDINATES: X Y Z | STRESS XX YY ZZ ZY XZ XY | GRAIN VOLUME | NUMBER OF ATOMS | VON-MISES STRESS |
#AVERAGE VON MISES STRESS FOR THIS STRUCTURE TYPE IS   326.650848
          1   432.741547      1034.44836      77.3495483     -811.026428     -716.803345     -610.946533     -67.5365067     -79.6209946     0.659035504      421322.719
          2   406.115051      799.608459      75.9181824     -788.860413     -770.771240     -828.792480     -9.57763577     -51.3218269     -16.3815689      306090.750
          3   888.489624      655.835632      75.8183670     -991.877747     -1033.17593     -637.436768     -8.16437721     -11.8227434     -10.1940355      357057.094
          4   1384.20996      608.589966      74.7481461     -805.394714     -679.517151     -609.736267      39.5209618     -7.58723354     -53.8090324      350403.250
          5   497.694275      259.767578      75.0820389     -669.968750     -781.577637     -383.774933     -30.9644871      17.8341599      81.4959717      287461.625
          6   130.113358      840.379150      76.5878983     -1086.48950     -1087.50867     -653.679321     -28.2191124     -73.8073578     -100.556442      424181.031
          7   769.827271      155.372955      75.8221893     -808.344543     -846.308472     -412.757996      21.1091900     -73.2035675      6.44057941      336378.031
          8   1624.18860      313.299194      76.5558319     -679.990845     -593.929016     -372.642731      5.53026962      15.0568819      68.6287231      501773.250
          9   743.418335      303.223938      77.0325699     -773.728394     -693.703735     -789.049011      60.4842644      41.1435814     -8.10856247      368593.094
         10   1151.30005      723.310974      76.9029770     -501.962891     -298.072021     -412.825043      74.8614120      73.2609558     -49.4360123      272336.688
         11   1126.58228      257.324890      77.2834473     -602.725464     -707.894653     -817.555054      25.7849636     -17.5692596     -37.6587715      336139.812
         12   543.836975      351.313690      77.5519562     -813.284973     -812.297546     -469.028229      23.2262840      15.3496704     -20.2469254      265500.156
         13   529.162170      187.050354      76.2216034     -710.003174     -670.335205     -298.208069     -62.6268845      43.2074242     -30.2604313      292890.344
         14   307.664246      508.551666      76.3852615     -720.233215     -564.227783     -528.043396     -15.7117195      23.6106396      82.0408173      303005.188
         15   1733.97314      270.553009      76.8233414     -754.375183     -930.757996     -390.836334     -14.8211527     -16.9994793     -78.0294495      337119.219
         16   708.928894      315.666779      76.9113159     -608.619019     -786.770508     -317.247192      18.4688454     -40.9863129      51.3136711      324527.344
         17   1294.36084      948.118286      76.9160538     -571.489929     -832.604309     -292.712952     -9.76564789     -49.3924370      14.6376820      427822.719
         18   593.924805      721.379211      75.6586533     -1086.09094     -866.227051     -667.543701     -57.3408775     -15.5207167     -124.159706      269466.875
         19   1287.80591      601.410278      76.8024521     -733.278625     -836.402893     -399.716766     -11.7273788      33.3333969      8.09501076      402635.969
         20   1131.69824      1125.99902      76.3123703     -875.842590     -982.650757     -494.604736      51.8273582      23.7355022      13.0347471      215199.844
         21   453.341003      909.125244      76.4666443     -759.030151     -843.186523     -447.797699     -6.44706249     -39.2174606      16.1598949      393557.219
         22   1556.86877      1080.55469      75.4864807     -916.226624     -1030.62085     -325.841064      1.76506376     -20.1628628      106.220314      333511.469
         23   64.0333176      595.339172      75.9362106     -936.080688     -978.175903     -422.508972     -63.8120270     -49.2292557      107.777634      263335.812
         24   992.215576      459.316132      75.0406265     -1132.43066     -1158.69617     -551.078857      7.45950937     -13.0121326     -34.5318069      466899.688
         25   1586.65747      600.721985      76.9979019     -1113.02942     -998.060669     -589.133179     -24.3848858     -22.2088833      50.7378731      477434.031
         26   1419.48242      517.992737      76.2396622     -699.999207     -878.550781     -686.997192      49.8789139     -39.2118454     -135.210388      395318.031
         27   84.1235657      92.9419479      76.3966064     -858.131104     -707.361633     -802.770874     -60.9660339      38.4694939     -7.58636236      309649.625
         28   222.524658      320.382294      76.8099289     -880.661926     -784.881775     -571.594421      16.4166069      12.2184534     -89.6977539      284957.250
         29   1617.87366      832.300293      75.2613373     -1082.21082     -1127.95947     -398.712555     -9.29895210     -24.2657738      82.5025330      207283.750
```

Figure 11: Average Properties Output File

Inference: Fig.11 is a snapshot of the bigger output file, attached in the appendix, which contains all the analysed average properties in an efficient format along with the mention of the Input File names, File attributes, Macroscopic Von Mises Stress, Von Mises Stress of each structure type. The headers and comments make the output file easier for the User to understand and self explanatory. Each block of structure type is separated by headers mentioning the structure type, sequence of quantities, Von Mises stress in that particular structure type.

4.2 OUTPUT OF VISUALIZATION:PYTHON

4.2.1 VOLUME PLOTS



(a) Grain Number 11 (b) Grain Number 13 (c) Grain Number 15

(d) Grain Number 18 (e) Grain Number 117 (f) Grain Number 100

Figure 12: Visualization of Average Volume of Grains in a conglomerate w.r.t Average Volume of Polycrystal

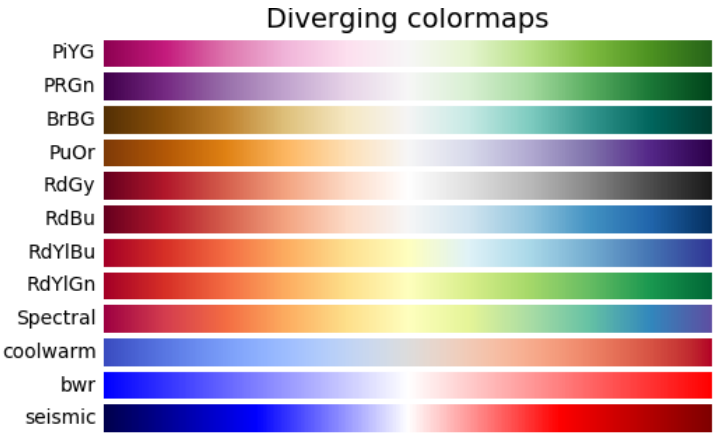
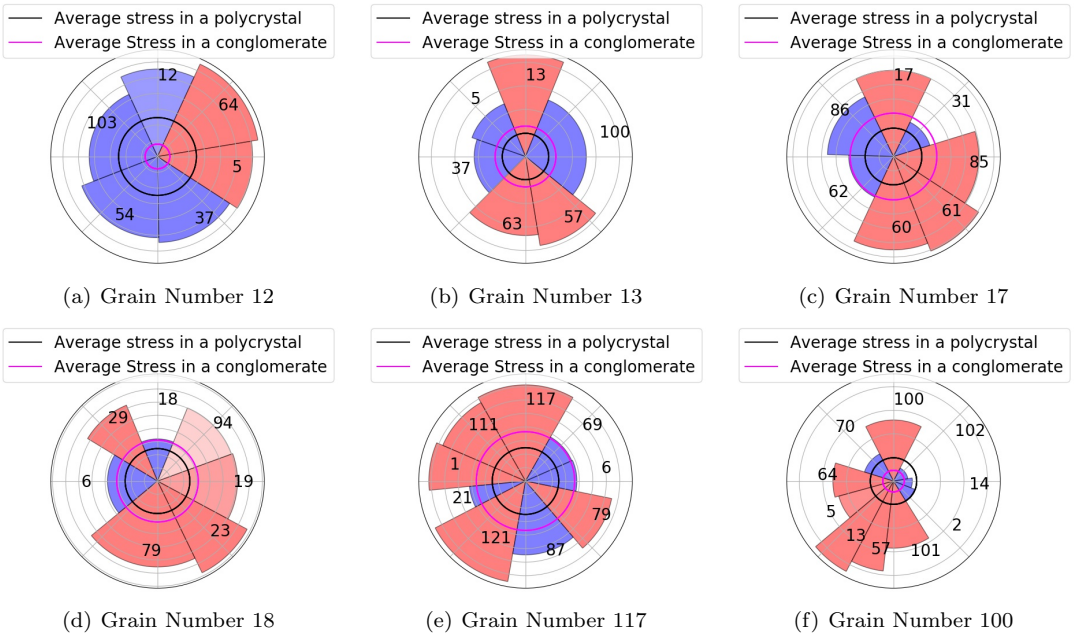


Figure 13: Diverging Color Maps

Inference: It can be observed from the above plots that the circle in 'black' represents the average volume of polycrystal. The angle between the sectors is the volume fraction with respect to neighbors(in terms of current conglomerate). The radius of the sector is the average volume of the grain in the polycrystal. Color of each sector is based on 'Divergence color maps' known as blue-white-red transition(bwr) which is based on of the polarization stress tensor. This particular plots are for Fcc+Hcp structure type by selecting 'D' in the parsing options of the file. In the fig.13, the diverging color map scheme 'bwr is used for coloring the sectors based on the polarized stress tensors.'The structure type considered is fcc+hcp.

4.2.2 STRESS PLOTS



(a) Grain Number 12 (b) Grain Number 13 (c) Grain Number 17

(d) Grain Number 18 (e) Grain Number 117 (f) Grain Number 100

Figure 14: Visualization of average stress in grains in a conglomerate with respect to average stress of polycrystal and average stress of the conglomerate

Inference: It can be observed from the above plots that the circle in 'black' represents the average stress in a polycrystal and a circle in 'magenta' represents average stress in a conglomerate. The angle between the sectors is the volume fraction with respect to neighbors(in terms of current conglomerate). The radius of the sector is the average stress of the grain in the polycrystal. Color of each sector is based on 'Divergence color maps' known as blue-white-red transition(bwr) which is based on the polarization stress tensor. This particular plots are for Fcc+Hcp structure type by selecting 'D' in the parsing options of the file. Similarly, like in fig.13the diverging color map scheme 'bwr' is used.The structure type considered here is fcc+hcp

4.2.3 DISTRIBUTION PLOTS

- **Function:Distribution_Plots** : CASE 1

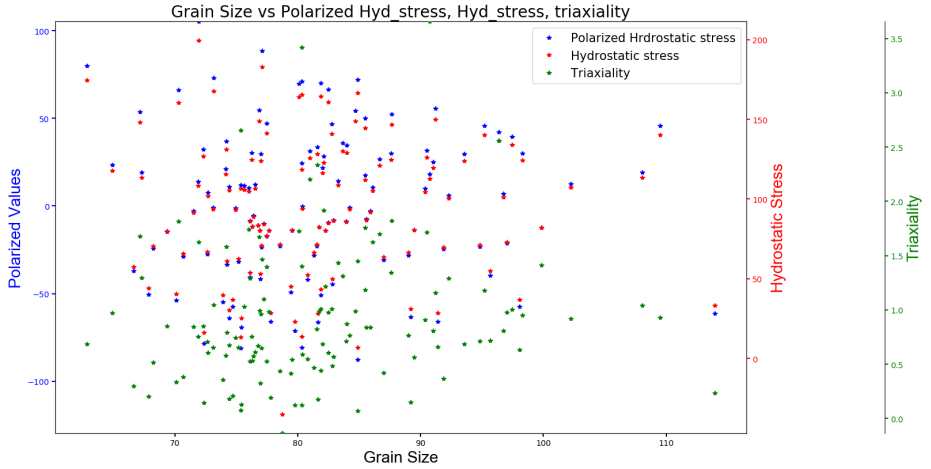


Figure 15: Visualization of Grain size vs polarized hydrostatic stress, hydrostatic stress, triaxiality

- **Function:Distribution_Plots** : CASE 2

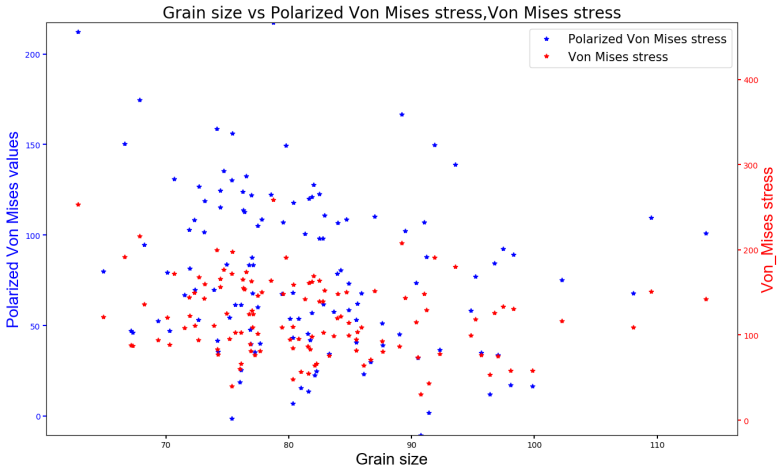


Figure 16: Visualization of Grain size vs polarized Von Mises stress, Von Mises stress

- **Function:Distribution_Plots** : CASE 3

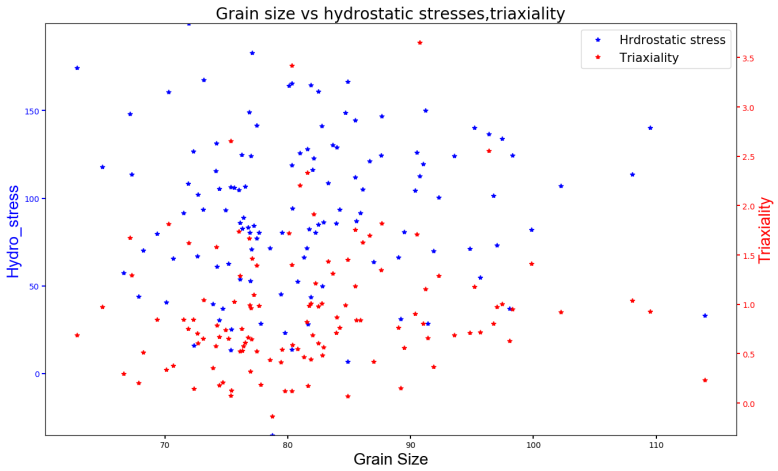


Figure 17: Visualization of Grain size vs hydrostatic stress,triaxiality

5 CODE TESTING

All the test cases are run via the same Fortran file. The only change is the name of the input file via parameter file.

5.1 TEST CASE 1

Aim: To verify the analysis of Stress_*xx* by replacing the Stress_*xx* value in all atoms w.r.t their Grain Number and considering all other stresses as '0' and constant volume(=0.5).

File Used: Generate Data file using *testcase_sig_xx_1.f90* and replace it in the place of stress file name in parameter file.

Expected Result: Since all other stresses are '0'. The Stress_*xx* and Von Mises Stress should be same as generated for each Grain for all Structure Types.

Obtained Result:

10	==========STRUCTURE TYPE-> 0 ==========											
11	IN NUMBER| COORDINATES: X Y Z | STRESS XX YY ZZ ZY XZ XY | GRAIN VOLUME | NUMBER OF ATOMS | VON-MISES STRESS |											
12	AGE VON MISES STRESS FOR THIS STRUCTURE TYPE IS 62.9401207											
13	1	492.741547	1034.44896	77.3495483	1.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	9894.00000	19788.0000	1.00000000
14	2	406.115051	799.609465	75.9181824	2.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	7152.50000	14305.0000	2.00000000
15	3	888.489624	655.835632	75.8183670	3.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	8134.50000	16269.0000	3.00000000
16	4	1304.20996	606.585966	74.7401461	4.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	8263.50000	16527.0000	4.00000000
17	5	497.694275	289.767578	76.0820389	5.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	7015.50000	14031.0000	5.00000000
18	6	130.113358	840.379150	76.5878983	6.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	9696.50000	19193.0000	6.00000000
19	7	769.827271	155.372955	75.8221893	7.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	7815.00000	15630.0000	7.00000000
20	8	1624.10860	313.295197	76.5858319	8.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	12227.0000	24454.0000	8.00000000
21	9	743.418335	303.223938	77.0325699	9.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	8975.50000	17951.0000	9.00000000
22	10	1151.30005	723.310974	76.9029770	10.0000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	6725.00000	13450.0000	10.0000000

Figure 18: Test Case 1

Inference:!!Verified!! **Output File**: Out_Average_Properties_testcase1.txt

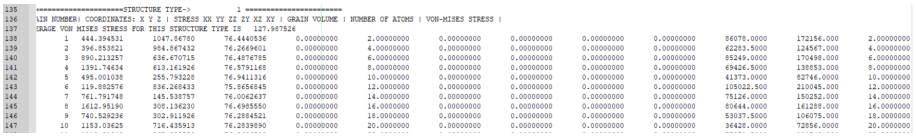
5.2 TEST CASE 2

Aim: To verify the analysis of Stress_*yy* by replacing the Stress_*yy* value in all atoms w.r.t their Grain Number*2 and considering all other stresses as '0' and constant volume(=0.5).

File Used: Generate Data file using *testcase_sig_yy_2.f90* and replace it in the place of stress file name in parameter file.

Expected Result: Since all other stresses are '0'. The Stress_*yy* and Von Mises Stress should be same as generated for each Grain for all Structure Types.

Obtained Result:



==========STRUCTURE TYPE->				1 ==========								
AIN NUMBER| COORDINATES: X Y Z | STRESS XX YY ZZ ZY XZ XY | GRAIN VOLUME | NUMBER OF ATOMS | VON-MISES STRESS |												
GRADE VON MISES STRESS FOR THIS STRUCTURE TYPE IS 127.987526												
	1	444.394531	1047.86780	76.4440536	0.00000000	2.00000000	0.00000000	0.00000000	0.00000000	0.00000000	86078.0000	172156.000	2.00000000
	2	396.853821	954.867432	76.2669401	0.00000000	4.00000000	0.00000000	0.00000000	0.00000000	0.00000000	62283.5000	124567.000	4.00000000
	3	890.213257	636.670715	76.4674785	0.00000000	6.00000000	0.00000000	0.00000000	0.00000000	0.00000000	85249.0000	170498.000	6.00000000
	4	1391.74634	613.161926	76.5791168	0.00000000	8.00000000	0.00000000	0.00000000	0.00000000	0.00000000	69426.5000	138853.000	8.00000000
	5	495.001038	255.793228	76.9411316	0.00000000	10.0000000	0.00000000	0.00000000	0.00000000	0.00000000	41373.0000	82746.0000	10.0000000
	6	119.882576	836.268433	75.5656845	0.00000000	12.0000000	0.00000000	0.00000000	0.00000000	0.00000000	105022.500	210045.000	12.0000000
	7	761.791748	145.538757	76.0062637	0.00000000	14.0000000	0.00000000	0.00000000	0.00000000	0.00000000	75126.0000	150252.000	14.0000000
	8	1612.95190	308.136230	76.6985550	0.00000000	16.0000000	0.00000000	0.00000000	0.00000000	0.00000000	80644.0000	161288.000	16.0000000
	9	740.529236	302.911926	76.2884521	0.00000000	18.0000000	0.00000000	0.00000000	0.00000000	0.00000000	53037.5000	106075.000	18.0000000
	10	1153.03625	716.435913	76.2539890	0.00000000	20.0000000	0.00000000	0.00000000	0.00000000	0.00000000	36425.0000	72856.0000	20.0000000

Figure 19: Test Case 2

Inference: !!Verified!! **Output File**: Out_Average_Properties_testcase2.txt

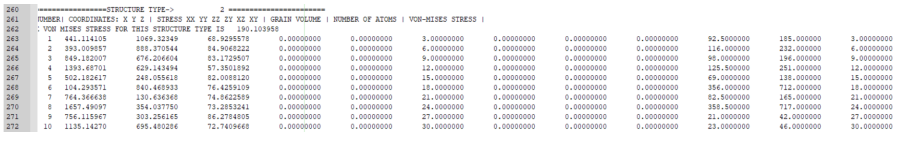
5.3 TEST CASE 3

Aim: To verify the analysis of Stress_*zz* by replacing the Stress_*zz* value in all atoms w.r.t their Grain Number*3 and considering all other stresses as '0'.

File Used: Generate Data file using *testcase_sig_zz_1.f90* and replace it in the place of stress file name in parameter file.

Expected Result: Since all other stresses are '0'. The Stress_*zz* and Von Mises Stress should be same for all Grains and for all Structure Types.

Obtained Result:



==========STRUCTURE TYPE->				2 ==========								
UMBER| COORDINATES: X Y Z | STRESS XX YY ZZ ZY XZ XY | GRAIN VOLUME | NUMBER OF ATOMS | VON-MISES STRESS |												
: VON MISES STRESS FOR THIS STRUCTURE TYPE IS 190.103958												
	1	441.114105	1069.32349	68.9295578	0.00000000	0.00000000	3.00000000	0.00000000	0.00000000	0.00000000	92.5000000	185.000000	3.00000000
	2	393.009857	888.370544	84.9068222	0.00000000	0.00000000	6.00000000	0.00000000	0.00000000	0.00000000	116.000000	232.000000	6.00000000
	3	849.182007	676.206604	83.1729507	0.00000000	0.00000000	9.00000000	0.00000000	0.00000000	0.00000000	98.0000000	196.000000	9.00000000
	4	1393.68701	629.143494	57.3501992	0.00000000	0.00000000	12.0000000	0.00000000	0.00000000	0.00000000	125.500000	251.000000	12.0000000
	5	502.182617	248.055618	82.0088120	0.00000000	0.00000000	15.0000000	0.00000000	0.00000000	0.00000000	69.0000000	138.000000	15.0000000
	6	104.293571	840.468933	76.4259109	0.00000000	0.00000000	18.0000000	0.00000000	0.00000000	0.00000000	356.000000	712.000000	18.0000000
	7	764.366638	130.636365	74.0622559	0.00000000	0.00000000	21.0000000	0.00000000	0.00000000	0.00000000	82.5000000	165.000000	21.0000000
	8	1657.49097	354.037750	73.2853241	0.00000000	0.00000000	24.0000000	0.00000000	0.00000000	0.00000000	358.500000	717.000000	24.0000000
	9	756.115967	303.256165	86.2784805	0.00000000	0.00000000	27.0000000	0.00000000	0.00000000	0.00000000	21.0000000	42.0000000	27.0000000
	10	1135.14270	695.480286	72.7409668	0.00000000	0.00000000	30.0000000	0.00000000	0.00000000	0.00000000	23.0000000	46.0000000	30.0000000

Figure 20: Test Case 3

Inference: !!Verified!! **Output File**: Out_Average_Properties_testcase3.txt

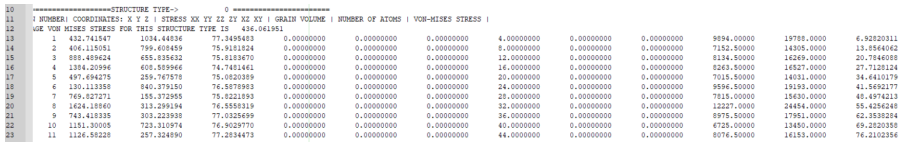
5.4 TEST CASE 4

Aim: To verify the analysis of Stress_*yz* by replacing the Stress_*yz* value in all atoms w.r.t their Grain Number*4 and considering all other stresses as '0' and constant volume(=0.5).

File Used: Generate Data file using *testcase_sig_yz_4.f90* and replace it in the place of stress file name in parameter file.

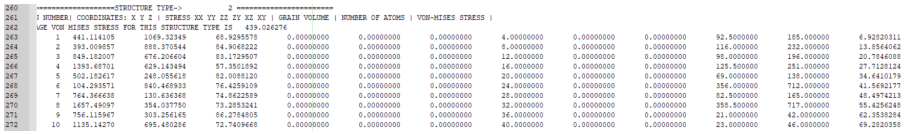
Expected Result: Since all other stresses are '0'. The Stress_*yz* should be same as generated for each Grain in all structure type and Von Mises Stress should be constant throughout the structure type.

Obtained Result:



==========STRUCTURE TYPE->				0 ==========								
J NUMBER| COORDINATES: X Y Z | STRESS XX YY ZZ ZY XZ XY | GRAIN VOLUME | NUMBER OF ATOMS | VON-MISES STRESS |												
AGE VON MISES STRESS FOR THIS STRUCTURE TYPE IS 436.061951												
	1	432.741547	1034.44836	77.3495483	0.00000000	0.00000000	0.00000000	4.00000000	0.00000000	0.00000000	9894.00000	19788.0000	6.92820311
	2	406.115051	799.602459	75.9181824	0.00000000	0.00000000	0.00000000	8.00000000	0.00000000	0.00000000	7152.50000	14305.0000	13.8564062
	3	888.489624	655.835632	75.8183670	0.00000000	0.00000000	0.00000000	12.0000000	0.00000000	0.00000000	8134.50000	16269.0000	20.7846088
	4	1384.20996	608.589966	74.7481461	0.00000000	0.00000000	0.00000000	16.0000000	0.00000000	0.00000000	8263.50000	16527.0000	27.7128124
	5	497.494275	259.767578	75.0820389	0.00000000	0.00000000	0.00000000	20.0000000	0.00000000	0.00000000	7015.50000	14031.0000	34.6410179
	6	130.113358	840.379150	76.5878983	0.00000000	0.00000000	0.00000000	24.0000000	0.00000000	0.00000000	9596.50000	19193.0000	41.5692177
	7	769.827271	155.372955	75.8221893	0.00000000	0.00000000	0.00000000	28.0000000	0.00000000	0.00000000	7815.00000	15630.0000	48.4974213
	8	1624.18560	313.299194	76.5558319	0.00000000	0.00000000	0.00000000	32.0000000	0.00000000	0.00000000	12227.0000	24454.0000	55.4256248
	9	743.418335	303.223938	77.0325699	0.00000000	0.00000000	0.00000000	36.0000000	0.00000000	0.00000000	8975.50000	17951.0000	62.3538284
	10	1151.30005	723.310974	76.9029770	0.00000000	0.00000000	0.00000000	40.0000000	0.00000000	0.00000000	6725.00000	13450.0000	69.2820358
	11	1126.58228	257.324890	77.2834473	0.00000000	0.00000000	0.00000000	44.0000000	0.00000000	0.00000000	8076.50000	16153.0000	76.2102356

Figure 21: Test Case 4_1



```
==================STRUCTURE TYPE-> 2 ====================
 NUMBER| COORDINATES: X Y Z | STRESS XX YY ZZ ZY XZ XY | GRAIN VOLUME | NUMBER OF ATOMS | VON-MISES STRESS |
GE VON MISES STRESS FOR THIS STRUCTURE TYPE IS  439.026276
    1  441.114105  1069.32349  68.9295578  0.00000000  0.00000000  0.00000000  4.00000000  0.00000000  0.00000000  92.5000000  185.000000  6.92820311
    2  393.009857  888.370544  84.9065222  0.00000000  0.00000000  0.00000000  8.00000000  0.00000000  0.00000000  116.000000  232.000000  13.8564062
    3  849.182007  676.206604  83.1729507  0.00000000  0.00000000  0.00000000  12.0000000  0.00000000  0.00000000  98.0000000  196.000000  20.7846088
    4  1393.66701  629.143494  57.3501892  0.00000000  0.00000000  0.00000000  16.0000000  0.00000000  0.00000000  125.500000  251.000000  27.7128124
    5  502.182617  248.055618  82.0088120  0.00000000  0.00000000  0.00000000  20.0000000  0.00000000  0.00000000  69.0000000  138.000000  34.6410179
    6  104.293571  840.468933  76.4259109  0.00000000  0.00000000  0.00000000  24.0000000  0.00000000  0.00000000  356.000000  712.000000  41.5692177
    7  764.366638  130.636368  74.8622589  0.00000000  0.00000000  0.00000000  28.0000000  0.00000000  0.00000000  82.5000000  165.000000  48.4974213
    8  1657.49097  354.037750  73.2853241  0.00000000  0.00000000  0.00000000  32.0000000  0.00000000  0.00000000  355.500000  717.000000  55.4256248
    9  756.115967  303.256165  86.2784805  0.00000000  0.00000000  0.00000000  36.0000000  0.00000000  0.00000000  21.0000000  42.0000000  62.3538284
   10  1135.14270  695.480286  72.7409668  0.00000000  0.00000000  0.00000000  40.0000000  0.00000000  0.00000000  23.0000000  46.0000000  69.2820358
```

Figure 22: Test Case 4_2

Inference: !!Verified!! **Output File**: Out_Average_Properties_testcase4.txt

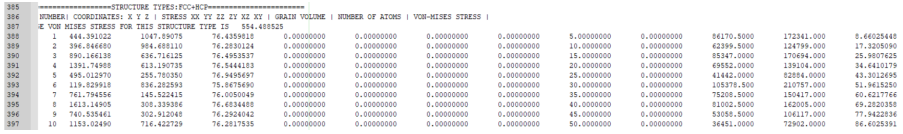
5.5 TEST CASE 5

Aim: To verify the analysis of Stress_*xz* by replacing the Stress_*xz* value in all atoms w.r.t their Grain Number*5 and considering all other stresses as '0' and constant volume(=0.5).

File Used: Generate Data file using *testcase_sig_xz_5.f90* and replace it in the place of stress file name in parameter file.

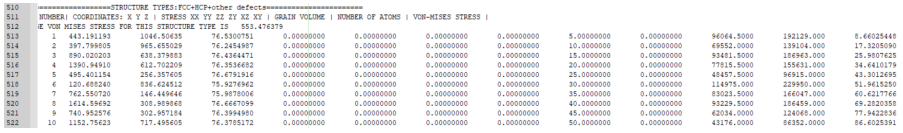
Expected Result: Since all other stresses are '0'. The Stress_*xz* should be same as generated for each Grain in all structure type and Von Mises Stress should be constant throughout the structure type.

Obtained Result:



```
==================STRUCTURE TYPES:FCC+HCP====================
 NUMBER| COORDINATES: X Y Z | STRESS XX YY ZZ ZY XZ XY | GRAIN VOLUME | NUMBER OF ATOMS | VON-MISES STRESS |
GE VON MISES STRESS FOR THIS STRUCTURE TYPE IS  584.488525
    1  444.391022  1047.89075  76.4359818  0.00000000  0.00000000  0.00000000  0.00000000  5.00000000  0.00000000  86170.5000  172341.000  8.66025448
    2  396.846680  984.685110  76.2830124  0.00000000  0.00000000  0.00000000  0.00000000  10.0000000  0.00000000  62399.5000  124799.000  17.3205090
    3  890.166138  636.716125  76.4953537  0.00000000  0.00000000  0.00000000  0.00000000  15.0000000  0.00000000  85347.0000  170694.000  25.9807625
    4  1391.74988  613.190735  76.5444183  0.00000000  0.00000000  0.00000000  0.00000000  20.0000000  0.00000000  69552.0000  139104.000  34.6410179
    5  495.012970  255.780350  76.9495697  0.00000000  0.00000000  0.00000000  0.00000000  25.0000000  0.00000000  41442.0000  82884.0000  43.3012695
    6  119.829918  836.282593  75.8675690  0.00000000  0.00000000  0.00000000  0.00000000  30.0000000  0.00000000  105375.500  210757.000  51.9615250
    7  761.794556  145.522415  76.0050049  0.00000000  0.00000000  0.00000000  0.00000000  35.0000000  0.00000000  75208.5000  150417.000  60.6217766
    8  1613.14905  308.339386  76.6534485  0.00000000  0.00000000  0.00000000  0.00000000  40.0000000  0.00000000  81002.5000  162005.000  69.2820358
    9  740.535461  302.912048  76.2924042  0.00000000  0.00000000  0.00000000  0.00000000  45.0000000  0.00000000  53058.5000  106117.000  77.9422836
   10  1153.02490  716.422729  76.2817535  0.00000000  0.00000000  0.00000000  0.00000000  50.0000000  0.00000000  36451.0000  72902.0000  86.6025391
```

Figure 23: Test Case 5_1



```
==================STRUCTURE TYPES:FCC+HCP+other defects====================
 NUMBER| COORDINATES: X Y Z | STRESS XX YY ZZ ZY XZ XY | GRAIN VOLUME | NUMBER OF ATOMS | VON-MISES STRESS |
GE VON MISES STRESS FOR THIS STRUCTURE TYPE IS  553.476379
    1  443.191193  1046.50635  76.5300751  0.00000000  0.00000000  0.00000000  0.00000000  5.00000000  0.00000000  96064.5000  192129.000  8.66025448
    2  397.799805  965.655029  76.2454987  0.00000000  0.00000000  0.00000000  0.00000000  10.0000000  0.00000000  69552.0000  139104.000  17.3205090
    3  890.020203  638.379883  76.4364471  0.00000000  0.00000000  0.00000000  0.00000000  15.0000000  0.00000000  93481.5000  186963.000  25.9807625
    4  1390.94910  612.702209  76.3536682  0.00000000  0.00000000  0.00000000  0.00000000  20.0000000  0.00000000  77815.5000  155631.000  34.6410179
    5  495.401154  256.357605  76.6791916  0.00000000  0.00000000  0.00000000  0.00000000  25.0000000  0.00000000  48457.5000  96915.0000  43.3012695
    6  120.658240  836.624512  75.9276962  0.00000000  0.00000000  0.00000000  0.00000000  30.0000000  0.00000000  114975.000  229950.000  51.9615250
    7  762.550720  146.449646  75.9875006  0.00000000  0.00000000  0.00000000  0.00000000  35.0000000  0.00000000  83023.5000  166047.000  60.6217766
    8  1614.59692  308.969848  76.6667099  0.00000000  0.00000000  0.00000000  0.00000000  40.0000000  0.00000000  93229.5000  186459.000  69.2820358
    9  740.952576  302.957184  76.3994980  0.00000000  0.00000000  0.00000000  0.00000000  45.0000000  0.00000000  62034.0000  124065.000  77.9422836
   10  1152.75623  717.495605  76.3755172  0.00000000  0.00000000  0.00000000  0.00000000  50.0000000  0.00000000  43176.0000  86352.0000  86.6025391
```

Figure 24: Test Case 5_2

Inference: !!Verified!! **Output File**: Out_Average_Properties_testcase5.txt

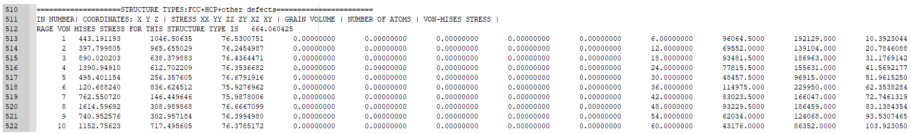
5.6 TEST CASE 6

Aim: To verify the analysis of Stress_*xy* by replacing the Stress_*xy* value in all atoms w.r.t their Grain Number*5 and considering all other stresses as '0' and constant volume(=0.5).

File Used: Generate Data file using *testcase_sig_xy_6.f90* and replace it in the place of stress file name in parameter file.

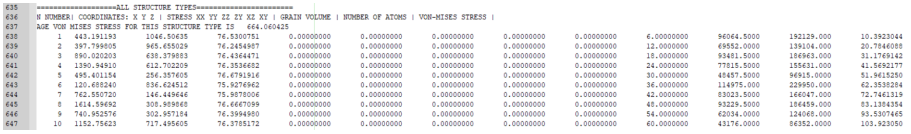
Expected Result: Since all other stresses are '0'. The Stress_*xy* should be same as generated for each Grain in all structure type and Von Mises Stress should be constant throughout the structure type.

Obtained Result:



```
==========STRUCTURE TYPES:FCC+HCP+other defects==========
IN NUMBER| COORDINATES: X Y Z | STRESS XX YY ZZ ZY XZ XY | GRAIN VOLUME | NUMBER OF ATOMS | VON-MISES STRESS |
RAGE VON MISES STRESS FOR THIS STRUCTURE TYPE IS   664.060425
     1   443.191193   1046.50635   76.5300751   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   6.00000000   96064.5000   192129.000   10.3923044
     2   397.799805   965.655029   76.2454987   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   12.0000000   69552.0000   139104.000   20.7846085
     3   890.020203   638.379883   76.4364471   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   18.0000000   93481.5000   186963.000   31.1769142
     4   1390.94910   612.702209   76.3536682   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   24.0000000   77815.5000   155631.000   41.5692177
     5   495.401154   256.357605   76.6791916   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   30.0000000   48457.5000   96915.0000   51.9615250
     6   120.688240   836.624512   75.9276962   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   36.0000000   114975.000   229950.000   62.3538284
     7   762.550720   146.449646   75.9878006   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   42.0000000   83023.5000   166047.000   72.7461319
     8   1614.59692   308.969968   76.6667099   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   48.0000000   93229.5000   186459.000   83.1384354
     9   740.952576   302.957184   76.3994980   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   54.0000000   62034.0000   124068.000   93.5307465
    10   1152.75623   717.495605   76.3785172   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   60.0000000   43176.0000   86352.0000   103.923050
```

Figure 25: Test Case 6_1



```
==========ALL STRUCTURE TYPES==========
N NUMBER| COORDINATES: X Y Z | STRESS XX YY ZZ ZY XZ XY | GRAIN VOLUME | NUMBER OF ATOMS | VON-MISES STRESS |
AGE VON MISES STRESS FOR THIS STRUCTURE TYPE IS   664.060425
     1   443.191193   1046.50635   76.5300751   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   6.00000000   96064.5000   192129.000   10.3923044
     2   397.799805   965.655029   76.2454987   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   12.0000000   69552.0000   139104.000   20.7846085
     3   890.020203   638.379883   76.4364471   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   18.0000000   93481.5000   186963.000   31.1769142
     4   1390.94910   612.702209   76.3536682   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   24.0000000   77815.5000   155631.000   41.5692177
     5   495.401154   256.357605   76.6791916   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   30.0000000   48457.5000   96915.0000   51.9615250
     6   120.688240   836.624512   75.9276962   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   36.0000000   114975.000   229950.000   62.3538284
     7   762.550720   146.449646   75.9878006   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   42.0000000   83023.5000   166047.000   72.7461319
     8   1614.59692   308.969968   76.6667099   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   48.0000000   93229.5000   186459.000   83.1384354
     9   740.952576   302.957184   76.3994980   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   54.0000000   62034.0000   124068.000   93.5307465
    10   1152.75623   717.495605   76.3785172   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   60.0000000   43176.0000   86352.0000   103.923050
```

Figure 26: Test Case 6_2

Inference: !!Verified!! **Output File**: Out_Average_Properties_testcase6.txt

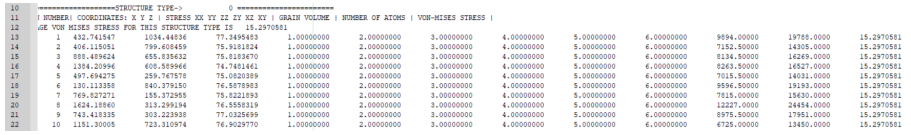
5.7 TEST CASE 7

Aim: To verify the analysis of Volume by replacing the Volume for half of the grains with 0.5 and other half grains with 2.0 and the stress values constant as 1.0 2.0 3.0 4.0 5.0 6.0 respectively for all grains.

File Used: Generate Data file using *testcase_sig_volchange_7.f90* and replace it in the place of stress file name in parameter file.

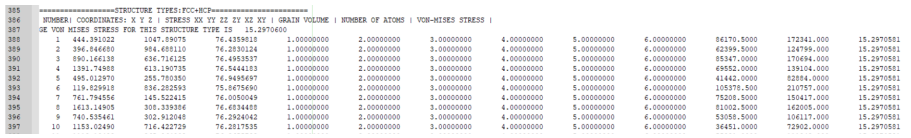
Expected Result: Since all other stresses are constant,the Von Mises Stress should be constant throughout all grains throughout all the structure types.

Obtained Result:



```
==========STRUCTURE TYPE->          0 ==========
| NUMBER| COORDINATES: X Y Z | STRESS XX YY ZZ ZY XZ XY | GRAIN VOLUME | NUMBER OF ATOMS | VON-MISES STRESS |
GE VON MISES STRESS FOR THIS STRUCTURE TYPE IS   15.2970581
     1   432.741547   1034.44836   77.3495483   1.00000000   2.00000000   3.00000000   4.00000000   5.00000000   6.00000000   9894.00000   19788.0000   15.2970581
     2   406.115051   799.608459   75.9181824   1.00000000   2.00000000   3.00000000   4.00000000   5.00000000   6.00000000   7152.50000   14305.0000   15.2970581
     3   888.489624   655.835632   75.8183670   1.00000000   2.00000000   3.00000000   4.00000000   5.00000000   6.00000000   8134.50000   16269.0000   15.2970581
     4   1394.20996   608.559966   74.7481461   1.00000000   2.00000000   3.00000000   4.00000000   5.00000000   6.00000000   8263.50000   16527.0000   15.2970581
     5   497.694275   259.767578   75.0820389   1.00000000   2.00000000   3.00000000   4.00000000   5.00000000   6.00000000   7015.50000   14031.0000   15.2970581
     6   130.113358   840.379150   76.5878983   1.00000000   2.00000000   3.00000000   4.00000000   5.00000000   6.00000000   9596.50000   19193.0000   15.2970581
     7   769.627271   155.372955   75.5221893   1.00000000   2.00000000   3.00000000   4.00000000   5.00000000   6.00000000   7815.00000   15630.0000   15.2970581
     8   1624.18860   313.299194   76.5558319   1.00000000   2.00000000   3.00000000   4.00000000   5.00000000   6.00000000   12227.0000   24454.0000   15.2970581
     9   743.418335   303.223938   77.0325699   1.00000000   2.00000000   3.00000000   4.00000000   5.00000000   6.00000000   8975.50000   17951.0000   15.2970581
    10   1151.30005   723.310974   76.9029770   1.00000000   2.00000000   3.00000000   4.00000000   5.00000000   6.00000000   6725.00000   13450.0000   15.2970581
```

Figure 27: Test Case 7_1



```
==========STRUCTURE TYPES:FCC+HCP==========
 NUMBER| COORDINATES: X Y Z | STRESS XX YY ZZ ZY XZ XY | GRAIN VOLUME | NUMBER OF ATOMS | VON-MISES STRESS |
GE VON MISES STRESS FOR THIS STRUCTURE TYPE IS   15.2970600
     1   444.391022   1047.89075   76.4359818   1.00000000   2.00000000   3.00000000   4.00000000   5.00000000   6.00000000   86170.5000   172341.000   15.2970581
     2   396.646600   984.655110   76.2330124   1.00000000   2.00000000   3.00000000   4.00000000   5.00000000   6.00000000   62399.5000   124799.000   15.2970581
     3   890.166138   636.716125   76.4953537   1.00000000   2.00000000   3.00000000   4.00000000   5.00000000   6.00000000   85347.0000   170694.000   15.2970581
     4   1391.74988   613.190735   76.5444183   1.00000000   2.00000000   3.00000000   4.00000000   5.00000000   6.00000000   69552.0000   139104.000   15.2970581
     5   495.012970   255.780350   76.9495697   1.00000000   2.00000000   3.00000000   4.00000000   5.00000000   6.00000000   41442.0000   82884.0000   15.2970581
     6   119.829918   836.282593   75.8675690   1.00000000   2.00000000   3.00000000   4.00000000   5.00000000   6.00000000   105378.500   210757.000   15.2970581
     7   761.794556   145.522415   76.0050049   1.00000000   2.00000000   3.00000000   4.00000000   5.00000000   6.00000000   75208.5000   150417.000   15.2970581
     8   1613.14905   308.339356   76.6334458   1.00000000   2.00000000   3.00000000   4.00000000   5.00000000   6.00000000   81002.5000   162005.000   15.2970581
     9   740.535461   302.912048   76.2924042   1.00000000   2.00000000   3.00000000   4.00000000   5.00000000   6.00000000   53058.5000   106117.000   15.2970581
    10   1153.02490   716.422729   76.2817535   1.00000000   2.00000000   3.00000000   4.00000000   5.00000000   6.00000000   36451.0000   72902.0000   15.2970581
```

Figure 28: Test Case 7_2

Inference: !!Verified!! **Output File**: Out_Average_Properties_testcase7.txt

5.8 TEST CASE 8

Aim: To verify the analysis of Average Von Mises Stress for all grains in each Structure Type.

File Used: Generate a Data file from *testcase_VM.f90* and replace it in the place of stress file name in parameter file.

Expected Result: Since all other stresses are '0' except sigma_xx which is equal to '1' in all cases, the Von Mises Stress should be constant(=1) throughout all grains throughout all the structure types. The Von Mises stress written in begining of each structure type should be same.

Obtained Result:

```
260  ================STRUCTURE TYPE->          2 ====================
261  N NUMBER| COORDINATES: X Y Z | STRESS XX YY ZZ IY XZ XY | GRAIN VOLUME | NUMBER OF ATOMS | VON-MISES STRESS |
262  AGE VON MISES STRESS FOR THIS STRUCTURE TYPE IS   1.00000000
263        1   441.114105      1069.32349      68.9295578      1.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      185.000000      185.000000      1.00000000
264        2   393.009857      885.370544      54.9068222      1.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      232.000000      232.000000      1.00000000
265        3   849.182007      676.206404      83.1729507      1.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      196.000000      196.000000      1.00000000
266        4   1393.68701      629.143494      57.3501892      1.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      251.000000      251.000000      1.00000000
267        5   502.182617      248.055618      82.0088120      1.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      138.000000      138.000000      1.00000000
268        6   104.293571      840.445933      76.4259109      1.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      712.000000      712.000000      1.00000000
269        7   764.366638      130.636368      74.8622589      1.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      165.000000      165.000000      1.00000000
270        8   1657.49097      354.037750      73.2853241      1.00000000      0.00000000      0.00000000      0.00000000      0.00000000      0.00000000      717.000000      717.000000      1.00000000
```

Figure 29: Test Case 8_1

Inference: !!Verified!! **Output File**: Out_Average_Properties_test8.txt

5.9 TEST CASE 9

Aim: To test the Volume Plots, if the Neighbors are accurate and the plotted Volumes are true.

File Used: Neighbors.txt and fig12(e)

Expected Result: Neighbours are exactly beside each other and their Volume positions match comparitively.

Obtained Result: They are perfectly beside each other according to fig:30 and their volumes comparison reflect their true values

```
115 115 79  87  92  43  66
116 116 51  69  73  55  73  32  83
117 117 111 1   21  121 87  79  6   69
118 118 35  104 95  58  53  47
```

Figure 30: Test Case 9_1

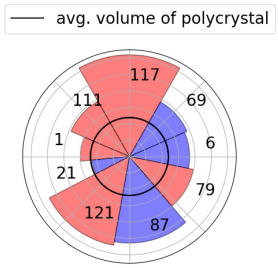


Figure 31: Test Case 9_2

Inference:!!Verified!!

5.10 TEST CASE 10

Aim: To test the Stress Plots, if the stress is same in all the grains.

File Used: Neighbors.txt, Input file from testcase 7 and fig12(e)

Expected Result: Since the stress tensor in fig.27 in all the grains is same and the corresponding Von Mised stress is also same, the plot shold have equivalent stresses, that means, radius of all the sectors is same.

Obtained Result: As in below figure:

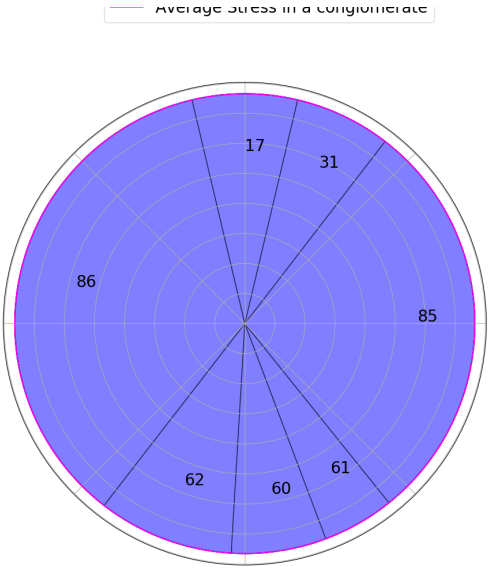


Figure 32: test case 10_1

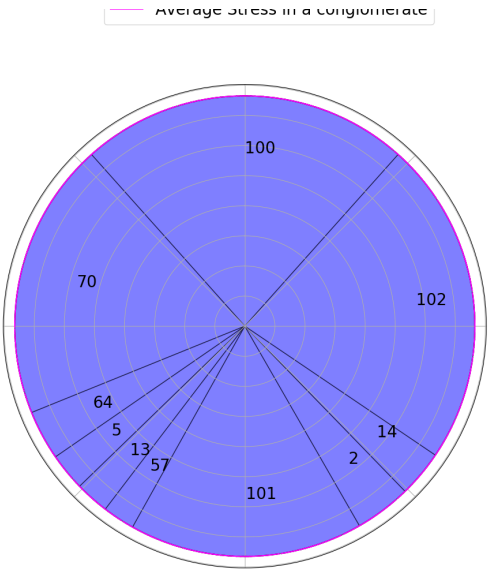


Figure 33: test case 10_2

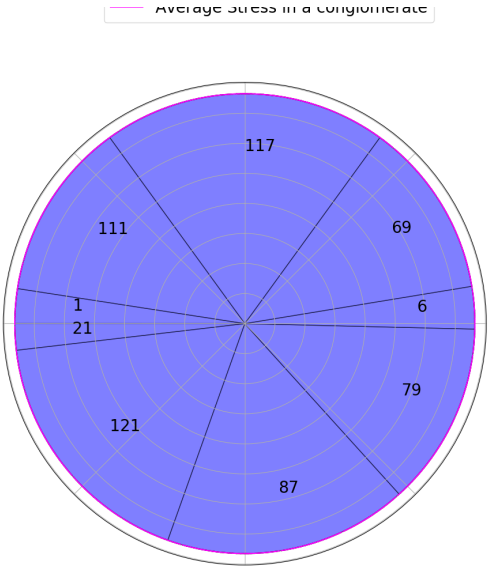


Figure 34: test case 10_3

Inference:!!Verified!! As observed in above figures, in grain 17,100,117, the sectors are having same radius.

5.11 TEST CASE 11

Aim: To check for the accuracy of the Distribution Plots.

File Used: Input file from testcase 7 and Input file from testcase 8

Expected Result: The plots should be consistant, as the stress values are same for all the grains in both the files.

Obtained Result: As shown below:

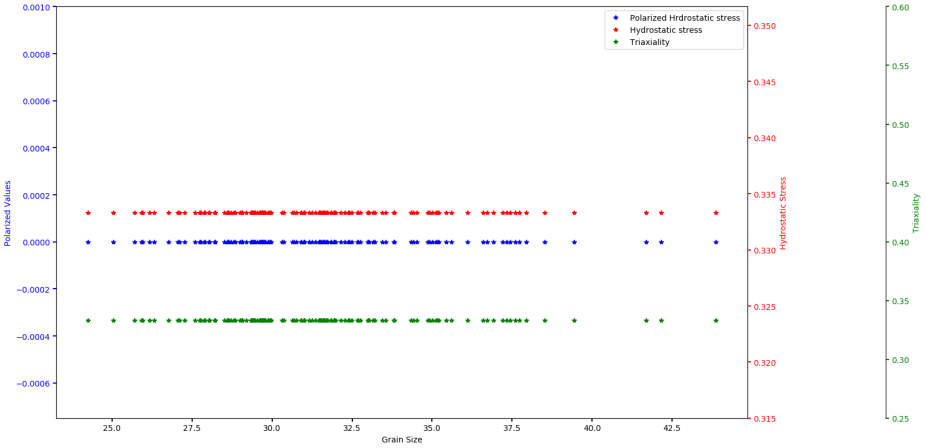


Figure 35: Test Case 11_1

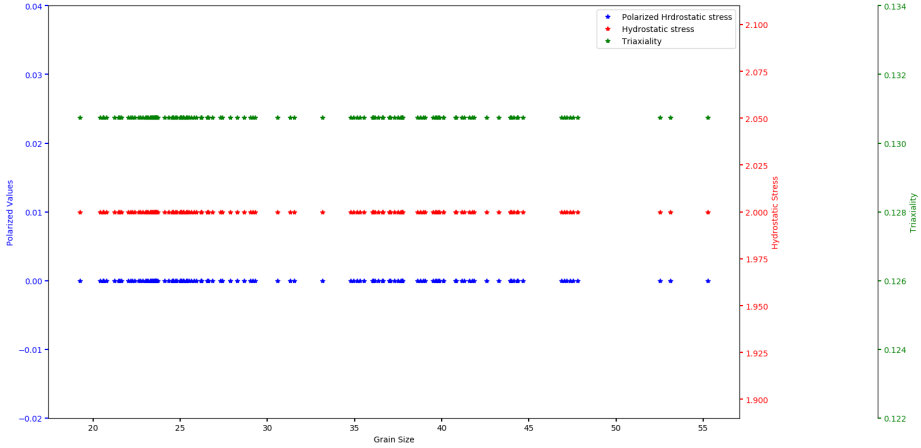


Figure 36: Test Case 11_1

Inference:!!Verified!!

6 CONCLUSION

In the current programming project, the following tasks have been accomplished.

- A parameter file has been designed which can be used for storing the User interests and can be used as a medium to extract and analyse desired parameter properties like stress, volume, coordinates.
- The Big data files are handled as per the requirement of the user based on his interested quantities, carefully avoiding repeated handling of the file and reducing computational cost.

- With the help of a 3Dimensional array in FORTRAN, all the necessary quantities are assembled after analysis based on their structure type and grain number.
- The 3D array is later written into a file specified by the user in the 'Parameter file', separating properties related to structure types as blocks of data.
- This Output file is then used for the Visualization via Python Programming language. The data from this file is read using numpy into an array and calculations are done for obtaining quantities which are required for visualization,like, Von Mises Stresses for a conglomerate, Grain volume fraction for a conglomerate, hydrostatic stresses, Grain size and other quantities.
- Using 'Argparse module' in Python, a command line interface options are designed for the User to obtain various plots, namely
  - Visualization of Average Volume of grains in a Conglomerate with respect to average volume in the polycrystal and colored based on divergence color maps(bwr) on polar plots.
  - Visualization of Average Stress in grains in a Conglomerate with respect to average stress in the polycrystal and average stress of the conglomerate and colored based on divergence color maps(bwr) on polar plots.
  - Distribution plots, namely:
    - Grain size vs polarized hydrostatic stresses, hydrostatic stresses, triaxiality.
    - Grain size vs polarized von mises stresses, von mises stresses.
    - Grain size vs hydrostatic stresses, triaxiality.

With the help of these analysed and visualized data, a further research or analysis can be initiated to study the behaviour of polycrystals, grain boundaries and dislocations. This helps the user to access the properties of the polycrystal and extend to any other properties with an advantage of time.

References

[1] M. Xiaofeng and C. Xiang, "Big data management: concepts, techniques and challenges [j]," *Journal of computer research and development*, vol. 1, no. 98, pp. 146–169, 2013.

[2] P. Zikopoulos, C. Eaton *et al.*, *Understanding big data: Analytics for enterprise class hadoop and streaming data*. McGraw-Hill Osborne Media, 2011.

[3] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 1, pp. 97–107, 2014.

[4] "A survey on big data analytics: Challenges, open research issues and tools," *International Journal of Advanced Computer Science and Applications*.

[5] A. Agrawal and A. Choudhary, "Perspective: Materials informatics and big data: Realization of the "fourth paradigm" of science in materials science," *Apl Materials*, vol. 4, no. 5, p. 053208, 2016.

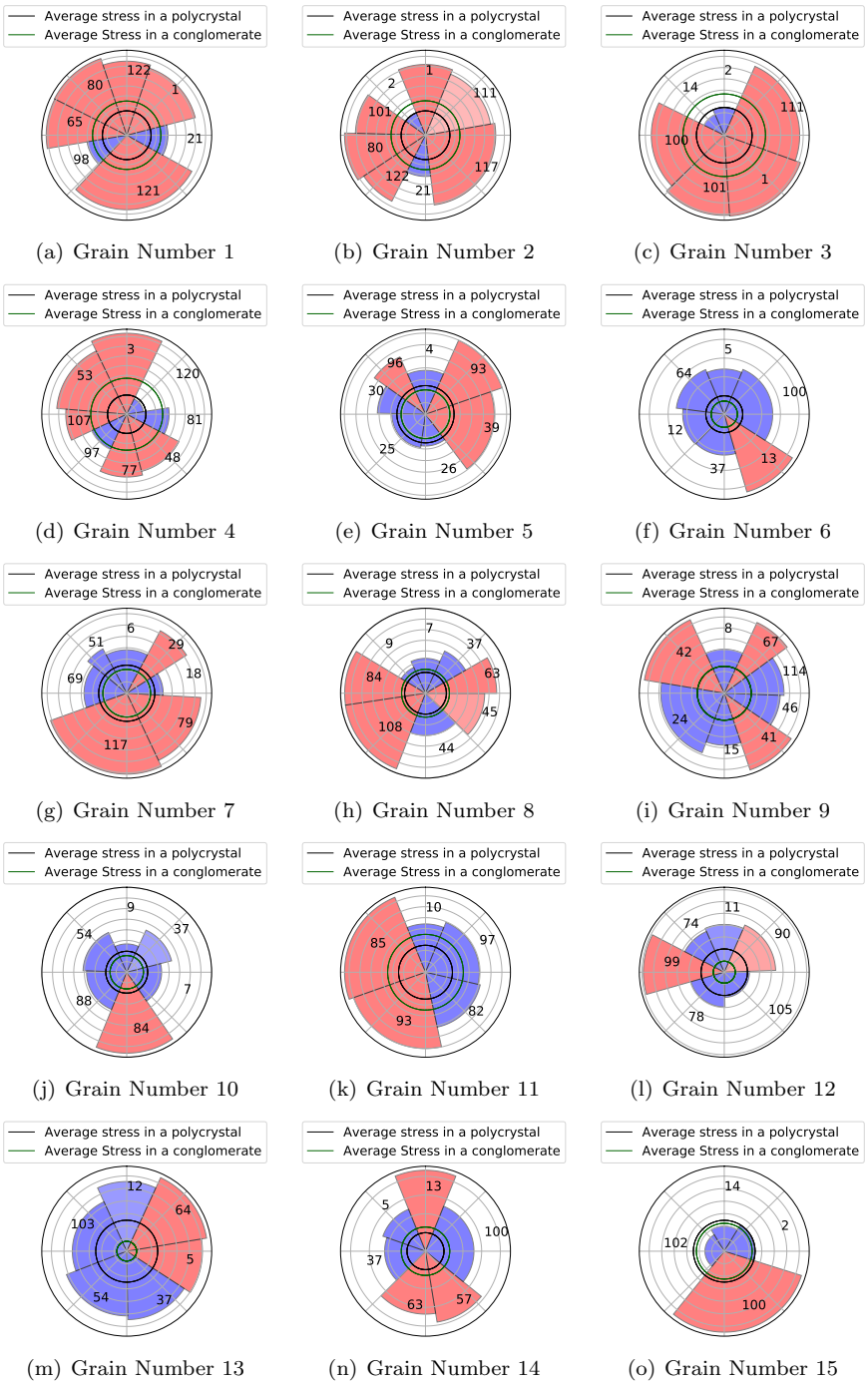
[6] A. Prakash and S. Sandfeld, "Chances and challenges in fusing data science with materials science: The working group "3d data science" is headed by prof. dr. stefan sandfeld." *Practical Metallography*, vol. 55, no. 8, pp. 493–514, 2018.

[7] J. D. Honeycutt and H. C. Andersen, "Molecular dynamics study of melting and freezing of small lennard-jones clusters," *The Journal of Physical Chemistry*, vol. 91, no. 19, pp. 4950–4963, 1987.

[8] A. Stukowski, "Structure identification methods for atomistic simulations of crystalline materials," *Modelling and Simulation in Materials Science and Engineering*, vol. 20, no. 4, p. 045021, 2012.

APPENDIX

Visualization of Grain Stresses w.r.t its neighbours and the Average Stress in the polycrystal and its conglomerate.



(a) Grain Number 1 (b) Grain Number 2 (c) Grain Number 3

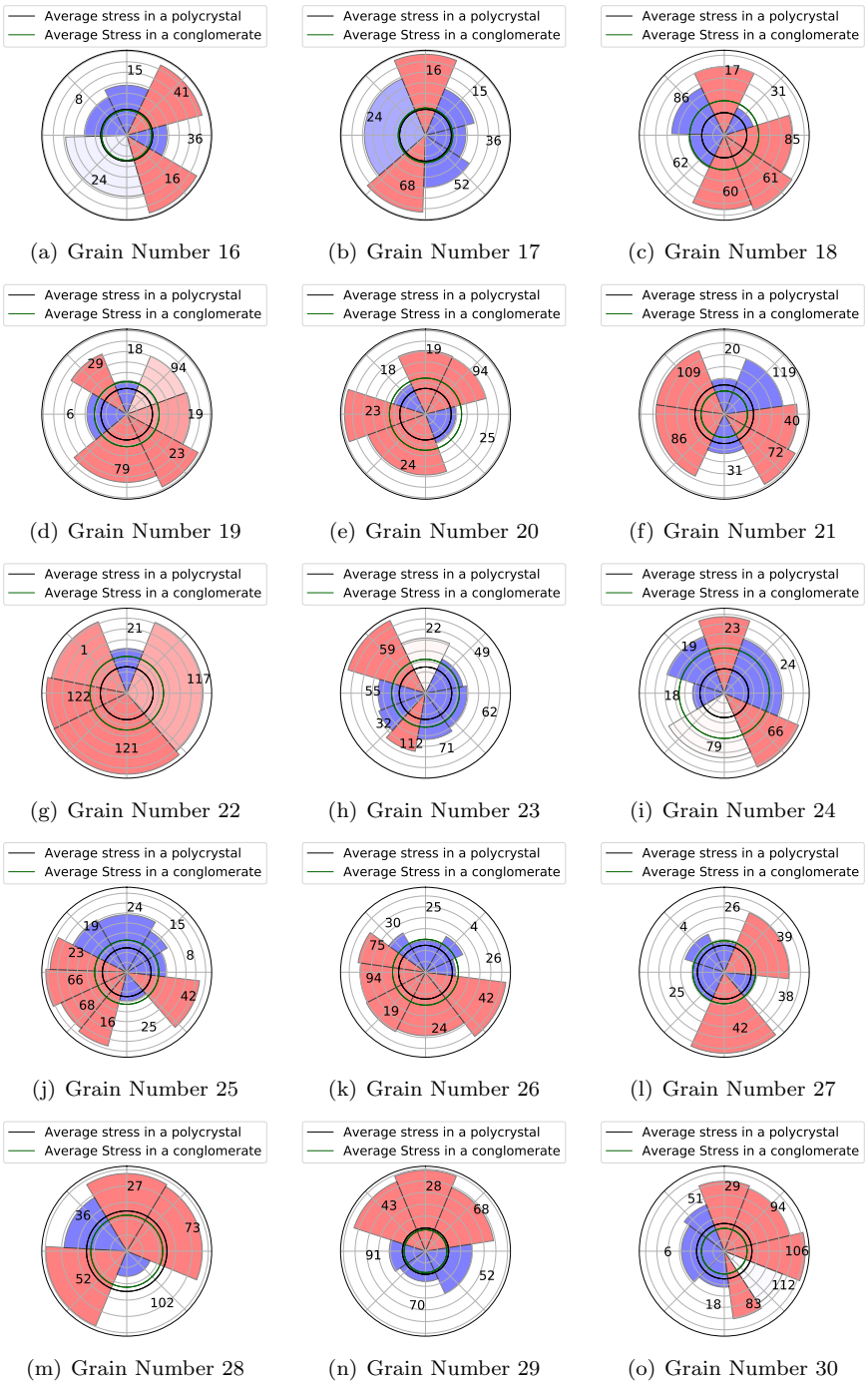
(d) Grain Number 4 (e) Grain Number 5 (f) Grain Number 6

(g) Grain Number 7 (h) Grain Number 8 (i) Grain Number 9

(j) Grain Number 10 (k) Grain Number 11 (l) Grain Number 12

(m) Grain Number 13 (n) Grain Number 14 (o) Grain Number 15

Figure 37: Visualization of Stress in Grains in a conglomerate for grain numbers 1-15



(a) Grain Number 16 (b) Grain Number 17 (c) Grain Number 18

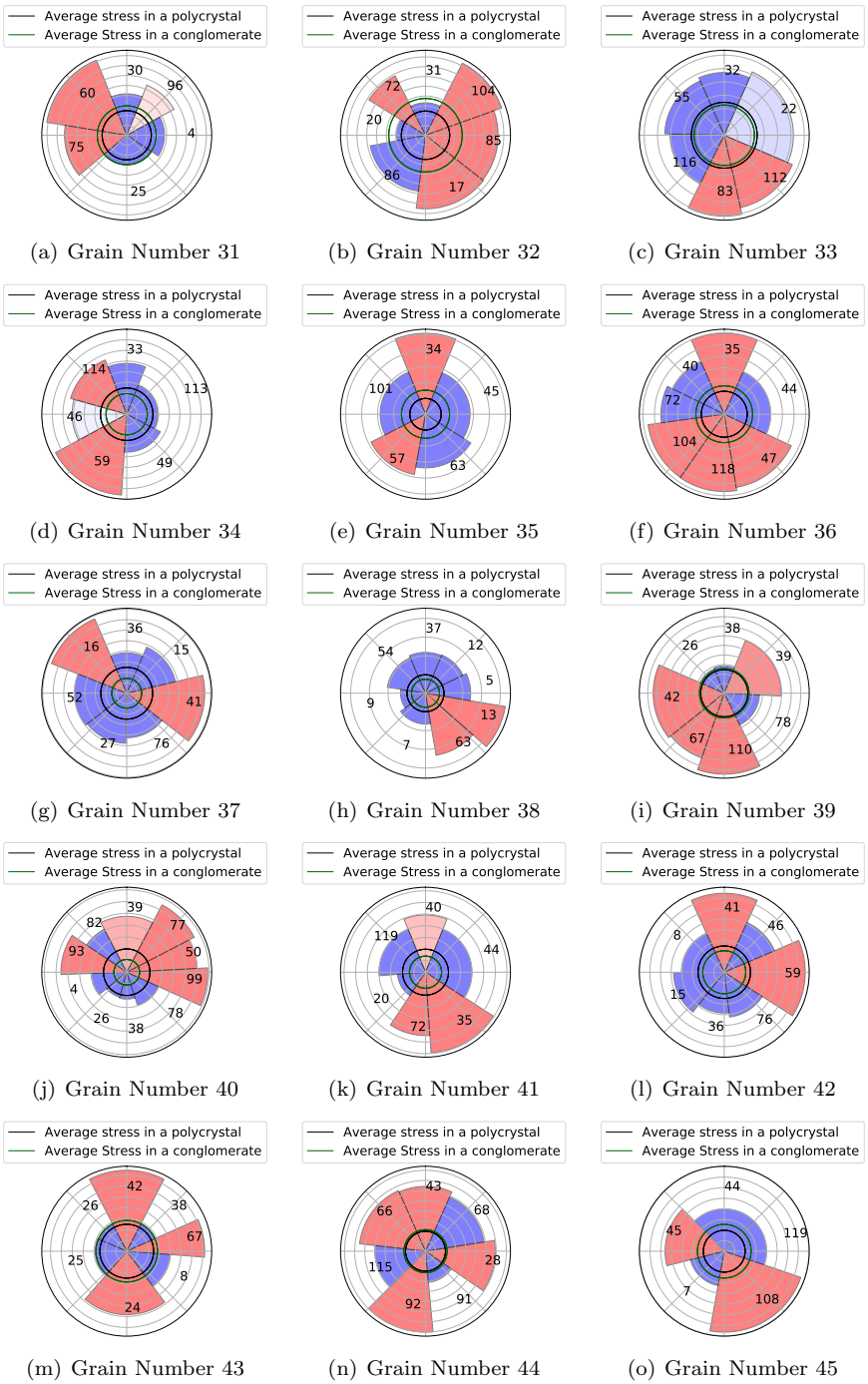
(d) Grain Number 19 (e) Grain Number 20 (f) Grain Number 21

(g) Grain Number 22 (h) Grain Number 23 (i) Grain Number 24

(j) Grain Number 25 (k) Grain Number 26 (l) Grain Number 27

(m) Grain Number 28 (n) Grain Number 29 (o) Grain Number 30

Figure 38: Visualization of Stress in Grains in a conglomerate for grain numbers 16-30



(a) Grain Number 31 (b) Grain Number 32 (c) Grain Number 33

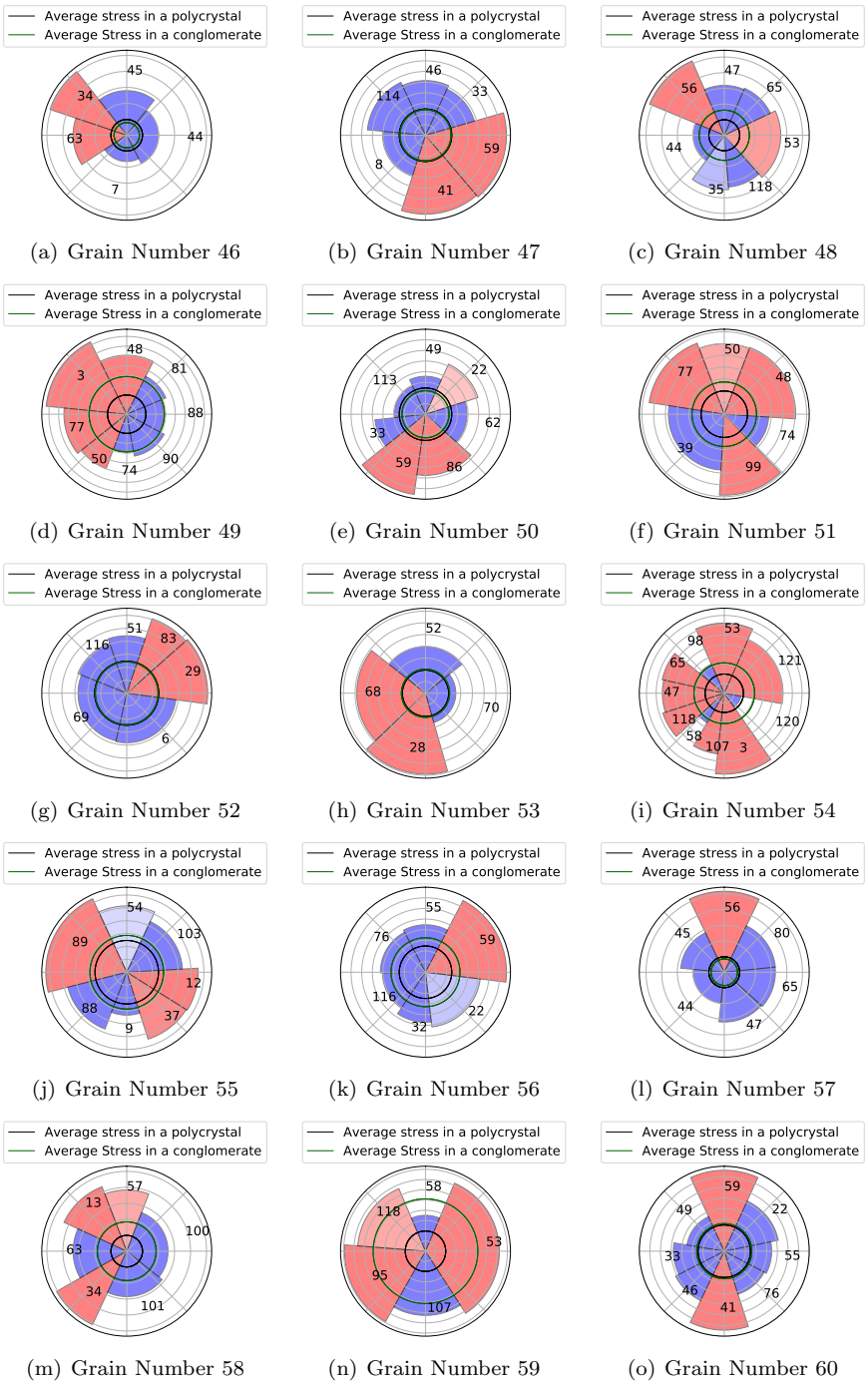
(d) Grain Number 34 (e) Grain Number 35 (f) Grain Number 36

(g) Grain Number 37 (h) Grain Number 38 (i) Grain Number 39

(j) Grain Number 40 (k) Grain Number 41 (l) Grain Number 42

(m) Grain Number 43 (n) Grain Number 44 (o) Grain Number 45

Figure 39: Visualization of Stress in Grains in a conglomerate for grain numbers 31-45



(a) Grain Number 46 (b) Grain Number 47 (c) Grain Number 48

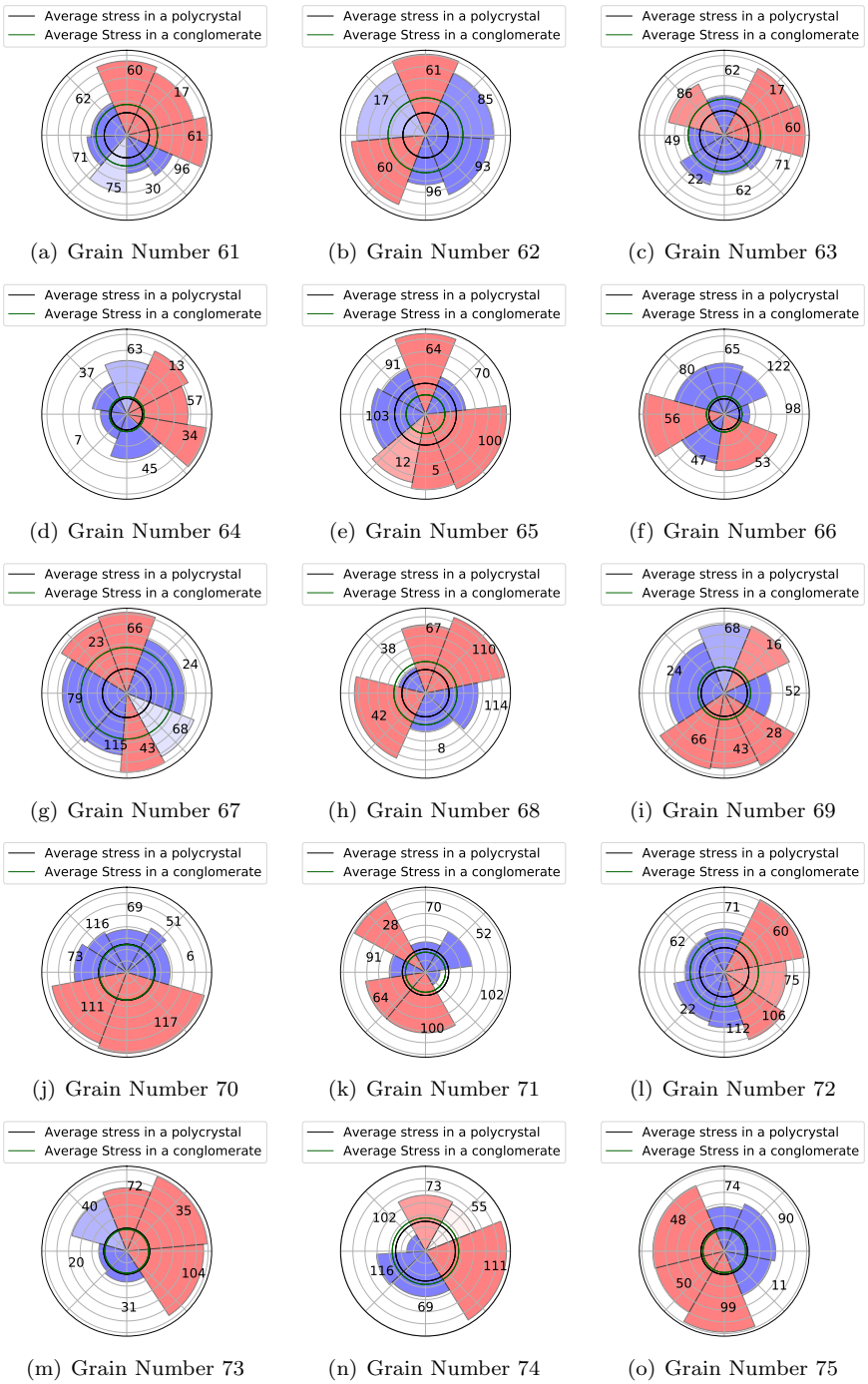
(d) Grain Number 49 (e) Grain Number 50 (f) Grain Number 51

(g) Grain Number 52 (h) Grain Number 53 (i) Grain Number 54

(j) Grain Number 55 (k) Grain Number 56 (l) Grain Number 57

(m) Grain Number 58 (n) Grain Number 59 (o) Grain Number 60

Figure 40: Visualization of Stress in Grains in a conglomerate for grain numbers 46-60



(a) Grain Number 61 (b) Grain Number 62 (c) Grain Number 63

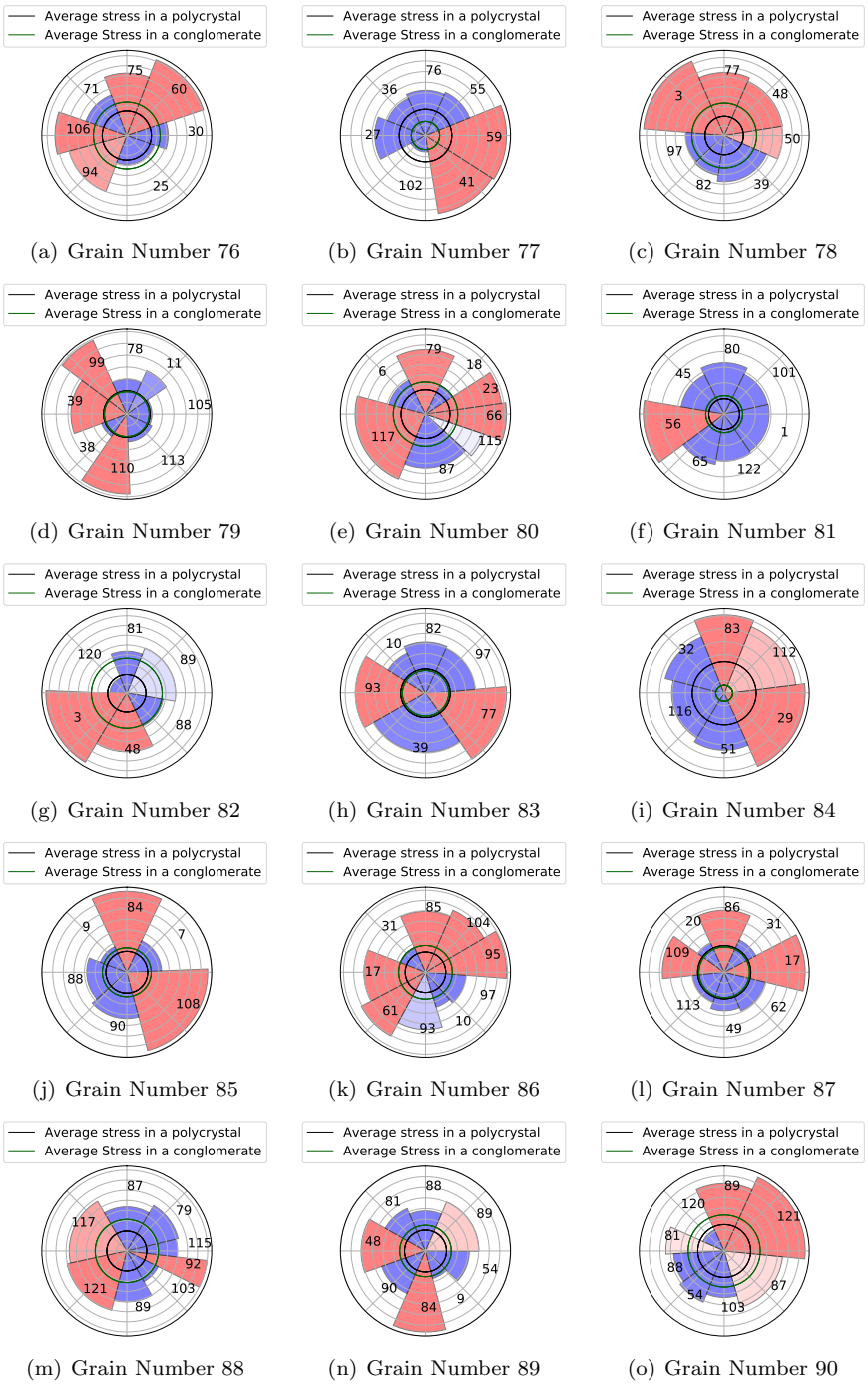
(d) Grain Number 64 (e) Grain Number 65 (f) Grain Number 66

(g) Grain Number 67 (h) Grain Number 68 (i) Grain Number 69

(j) Grain Number 70 (k) Grain Number 71 (l) Grain Number 72

(m) Grain Number 73 (n) Grain Number 74 (o) Grain Number 75

Figure 41: Visualization of Stress in Grains in a conglomerate for grain numbers 61-75



(a) Grain Number 76 (b) Grain Number 77 (c) Grain Number 78

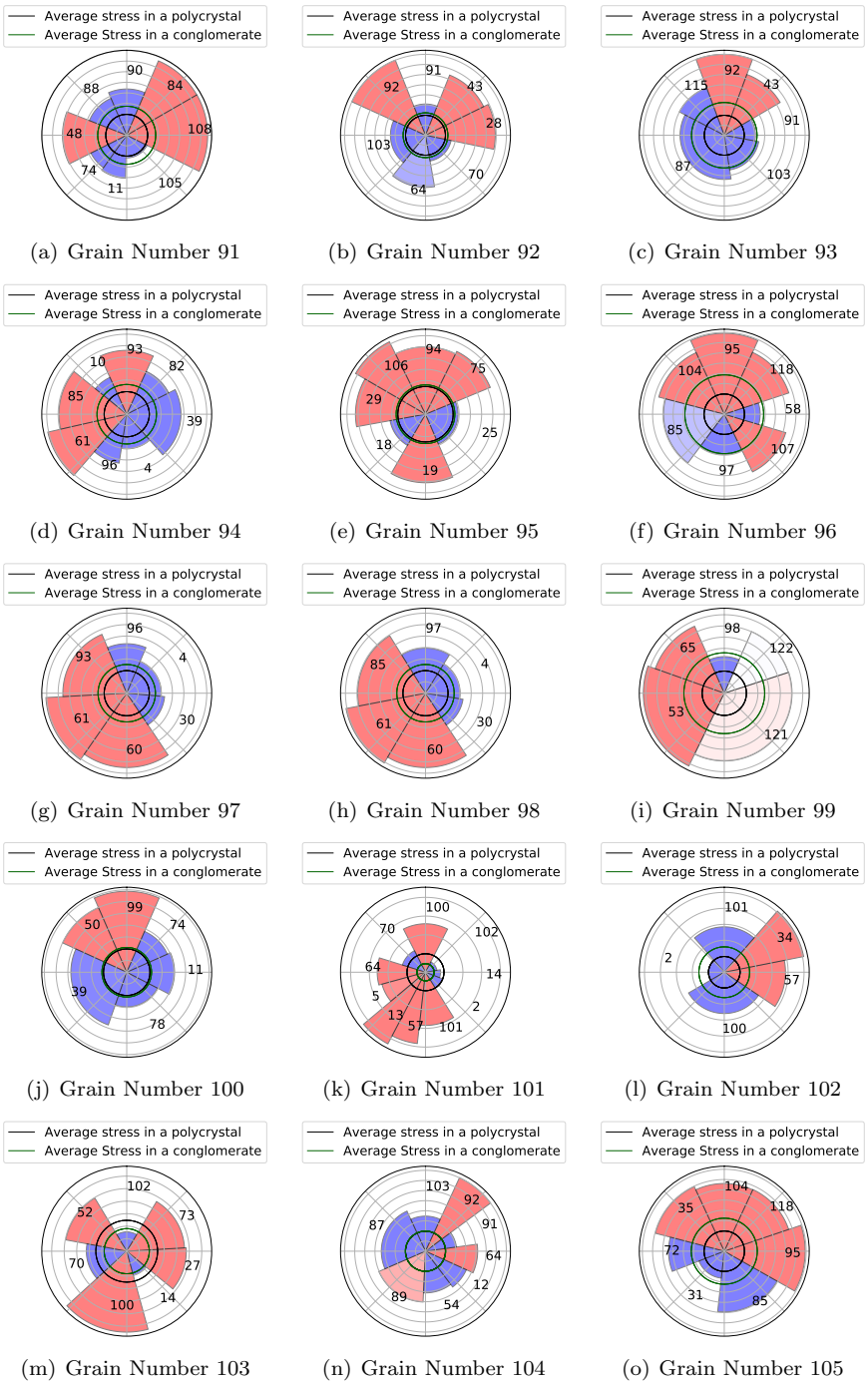
(d) Grain Number 79 (e) Grain Number 80 (f) Grain Number 81

(g) Grain Number 82 (h) Grain Number 83 (i) Grain Number 84

(j) Grain Number 85 (k) Grain Number 86 (l) Grain Number 87

(m) Grain Number 88 (n) Grain Number 89 (o) Grain Number 90

Figure 42: Visualization of Stress in Grains in a conglomerate for grain numbers 76-90



(a) Grain Number 91 (b) Grain Number 92 (c) Grain Number 93

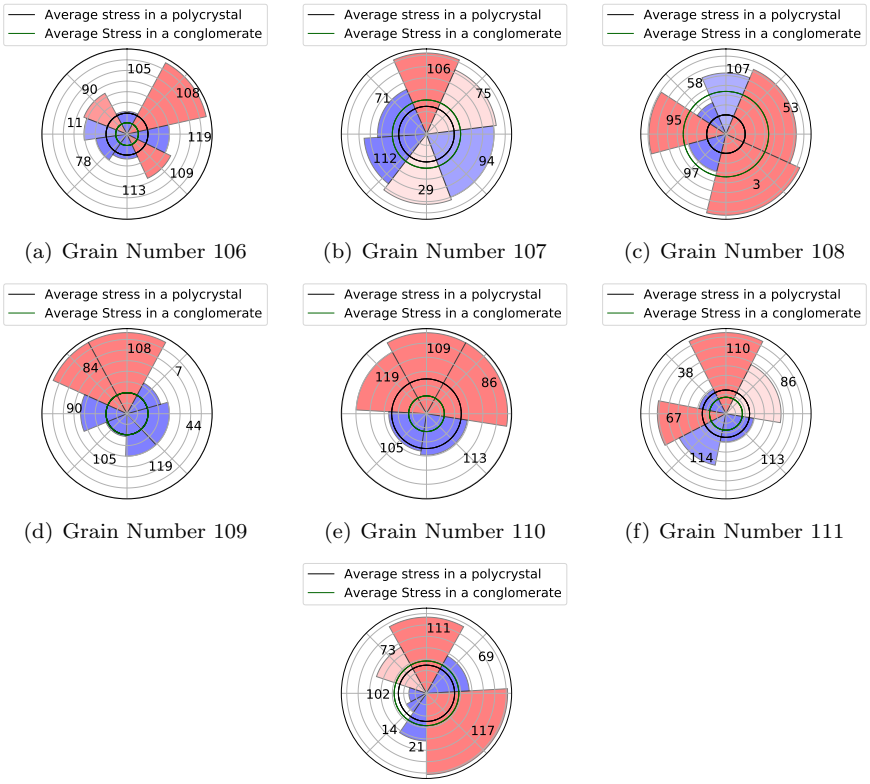
(d) Grain Number 94 (e) Grain Number 95 (f) Grain Number 96

(g) Grain Number 97 (h) Grain Number 98 (i) Grain Number 99

(j) Grain Number 100 (k) Grain Number 101 (l) Grain Number 102

(m) Grain Number 103 (n) Grain Number 104 (o) Grain Number 105

Figure 43: Visualization of Stress in Grains in a conglomerate for grain numbers 91-105



(a) Grain Number 106

(b) Grain Number 107

(c) Grain Number 108

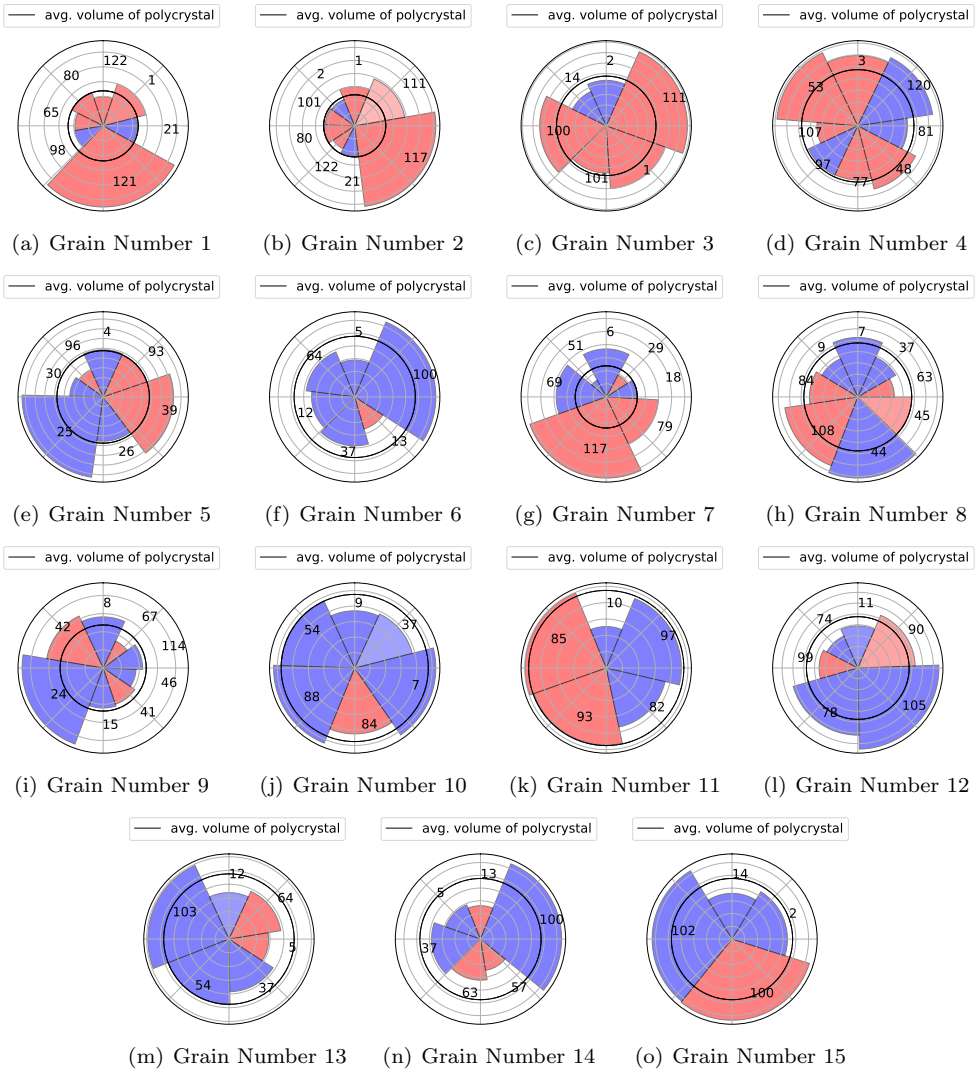
(d) Grain Number 109

(e) Grain Number 110

(f) Grain Number 111

(g) Grain Number 112

Figure 44: Visualization of Stress in Grains in a conglomerate for grain numbers 106-112



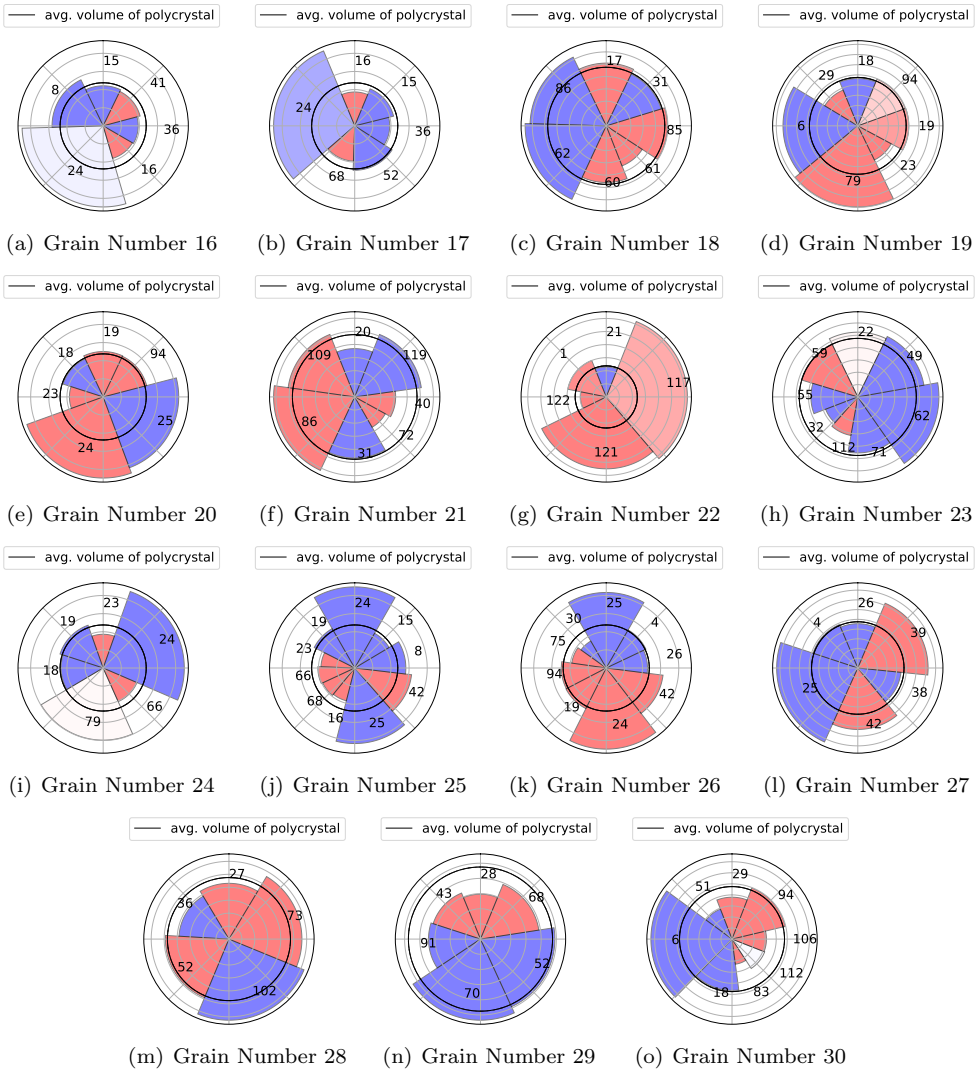
(a) Grain Number 1 (b) Grain Number 2 (c) Grain Number 3 (d) Grain Number 4

(e) Grain Number 5 (f) Grain Number 6 (g) Grain Number 7 (h) Grain Number 8

(i) Grain Number 9 (j) Grain Number 10 (k) Grain Number 11 (l) Grain Number 12

(m) Grain Number 13 (n) Grain Number 14 (o) Grain Number 15

Figure 45: Visualization of Stress in Grains in a conglomerate for grain numbers 1-15



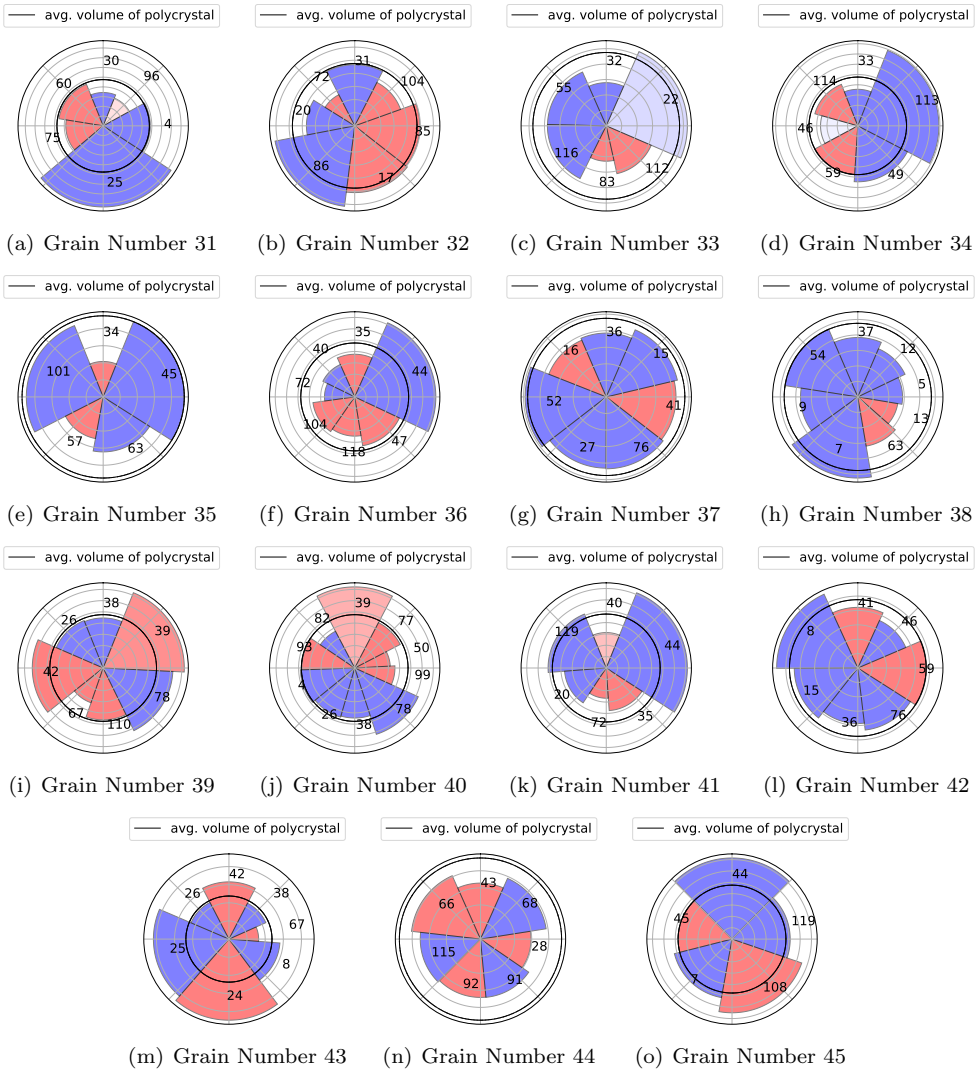
(a) Grain Number 16 (b) Grain Number 17 (c) Grain Number 18 (d) Grain Number 19

(e) Grain Number 20 (f) Grain Number 21 (g) Grain Number 22 (h) Grain Number 23

(i) Grain Number 24 (j) Grain Number 25 (k) Grain Number 26 (l) Grain Number 27

(m) Grain Number 28 (n) Grain Number 29 (o) Grain Number 30

Figure 46: Visualization of Volume of Grains in a conglomerate for grain numbers 16-30



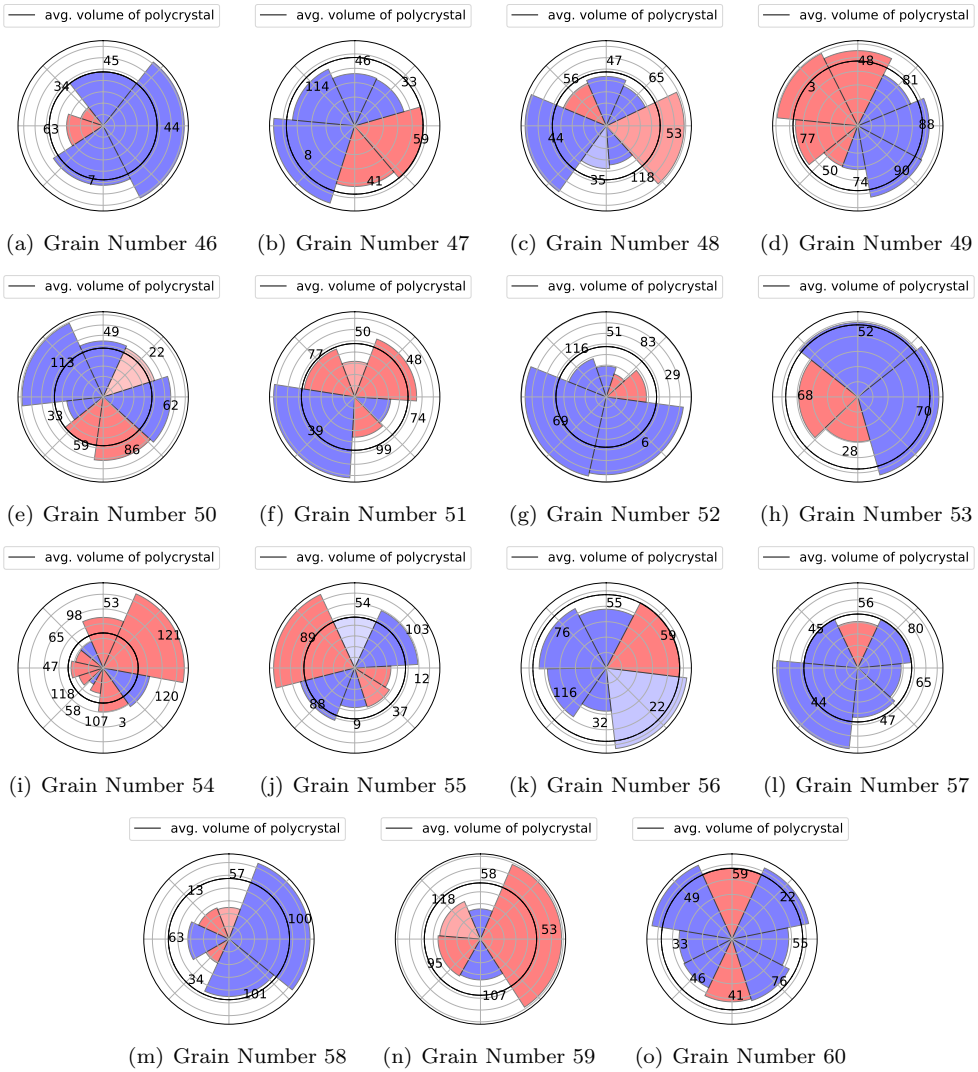
(a) Grain Number 31 (b) Grain Number 32 (c) Grain Number 33 (d) Grain Number 34

(e) Grain Number 35 (f) Grain Number 36 (g) Grain Number 37 (h) Grain Number 38

(i) Grain Number 39 (j) Grain Number 40 (k) Grain Number 41 (l) Grain Number 42

(m) Grain Number 43 (n) Grain Number 44 (o) Grain Number 45

Figure 47: Visualization of Volume of Grains in a conglomerate for grain numbers 31-45



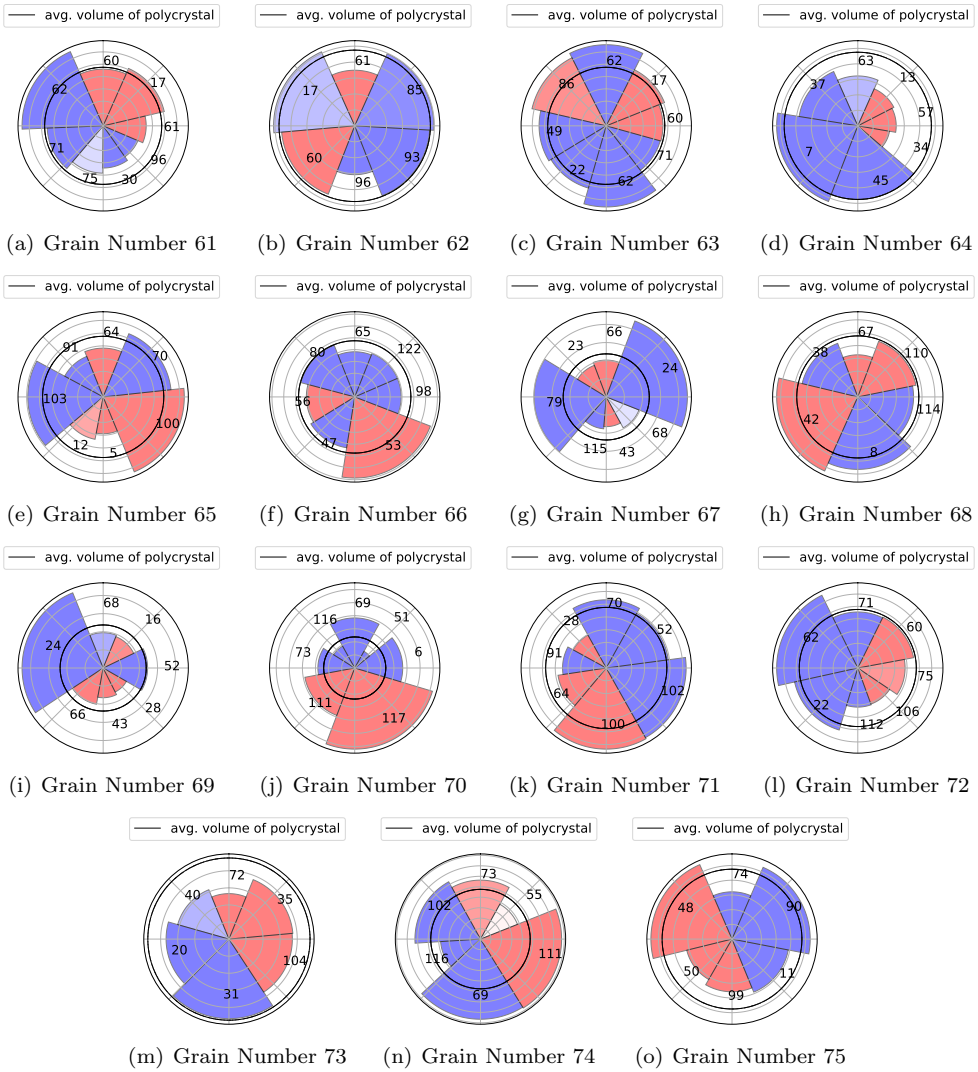
(a) Grain Number 46 (b) Grain Number 47 (c) Grain Number 48 (d) Grain Number 49

(e) Grain Number 50 (f) Grain Number 51 (g) Grain Number 52 (h) Grain Number 53

(i) Grain Number 54 (j) Grain Number 55 (k) Grain Number 56 (l) Grain Number 57

(m) Grain Number 58 (n) Grain Number 59 (o) Grain Number 60

Figure 48: Visualization of Volume of Grains in a conglomerate for grain numbers 46-60



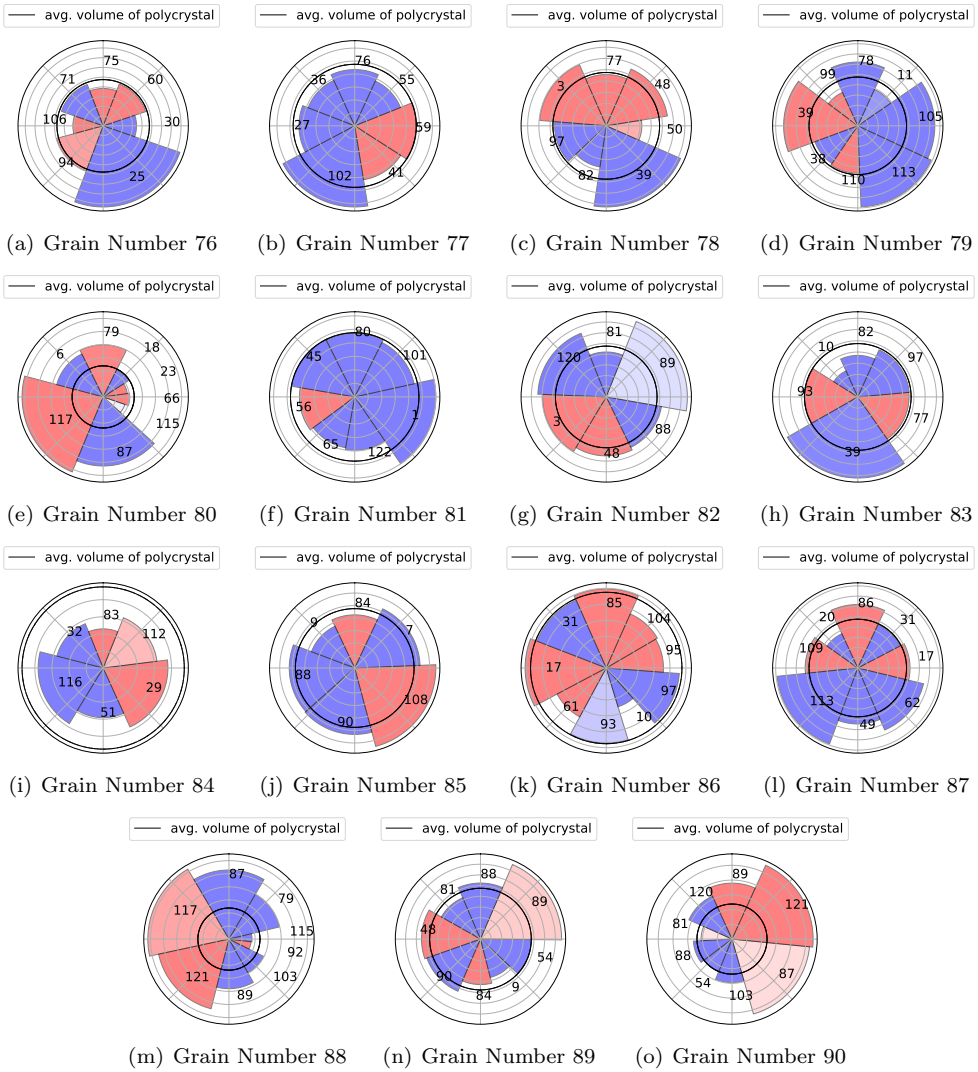
(a) Grain Number 61 (b) Grain Number 62 (c) Grain Number 63 (d) Grain Number 64

(e) Grain Number 65 (f) Grain Number 66 (g) Grain Number 67 (h) Grain Number 68

(i) Grain Number 69 (j) Grain Number 70 (k) Grain Number 71 (l) Grain Number 72

(m) Grain Number 73 (n) Grain Number 74 (o) Grain Number 75

Figure 49: Visualization of Volume of Grains in a conglomerate for grain numbers 61-75



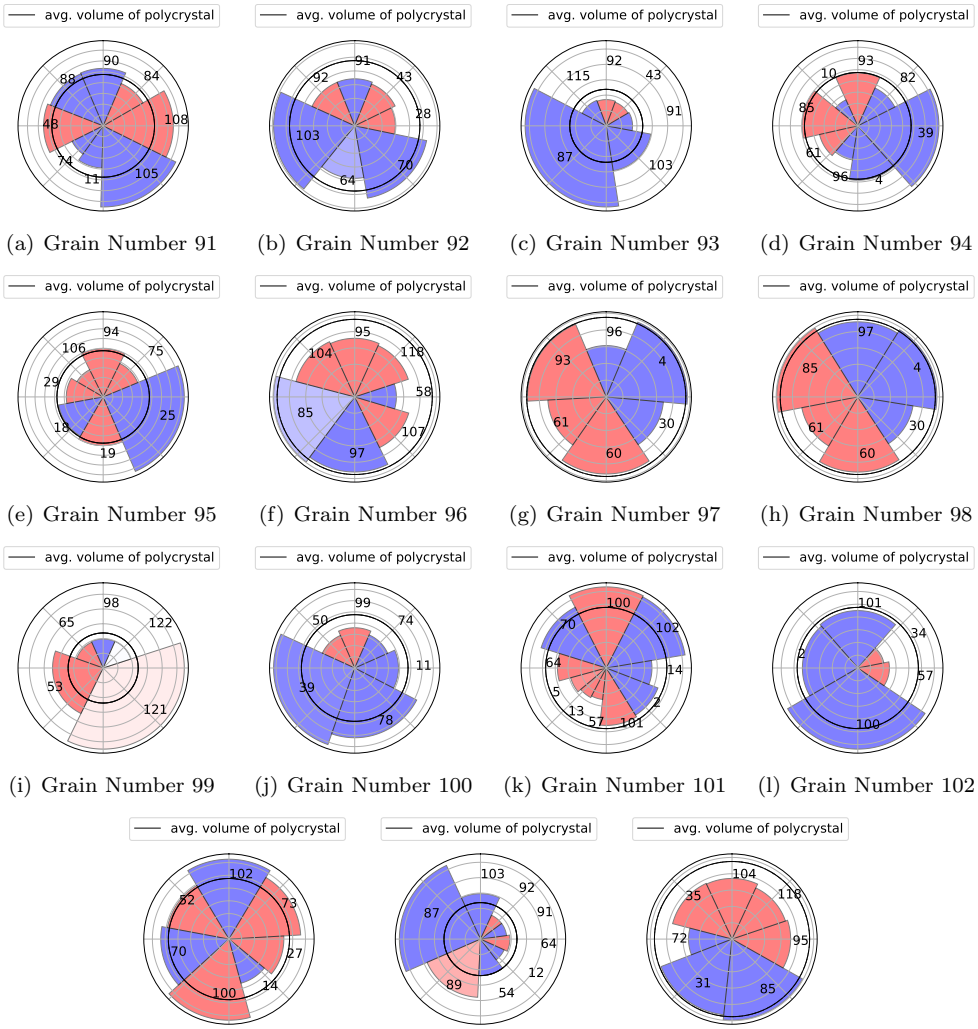
(a) Grain Number 76 (b) Grain Number 77 (c) Grain Number 78 (d) Grain Number 79

(e) Grain Number 80 (f) Grain Number 81 (g) Grain Number 82 (h) Grain Number 83

(i) Grain Number 84 (j) Grain Number 85 (k) Grain Number 86 (l) Grain Number 87

(m) Grain Number 88 (n) Grain Number 89 (o) Grain Number 90

Figure 50: Visualization of Volume of Grains in a conglomerate for grain numbers 76-90



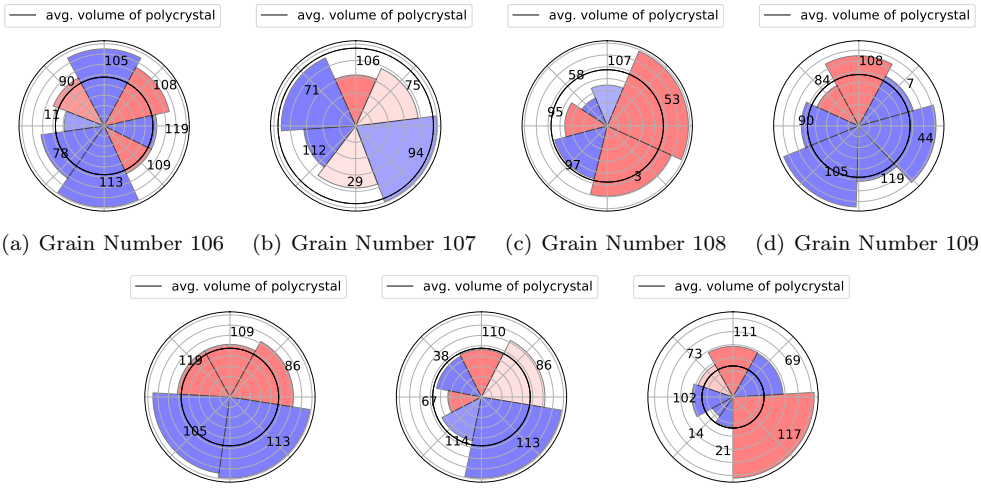
(a) Grain Number 91 (b) Grain Number 92 (c) Grain Number 93 (d) Grain Number 94

(e) Grain Number 95 (f) Grain Number 96 (g) Grain Number 97 (h) Grain Number 98

(i) Grain Number 99 (j) Grain Number 100 (k) Grain Number 101 (l) Grain Number 102

(m) Grain Number 103 (n) Grain Number 104 (o) Grain Number 105

Figure 51: Visualization of Volume of Grains in a conglomerate for grain numbers 91-105



(a) Grain Number 106 (b) Grain Number 107 (c) Grain Number 108 (d) Grain Number 109

(e) Grain Number 110 (f) Grain Number 111 (g) Grain Number 112

Figure 52: Visualization of Volume of Grains in a conglomerate for grain numbers 106-112