

# A Flexible and Efficient Output File Format for Grain-Scale Multiphysics Simulations

Martin Diehl<sup>1</sup> · Philip Eisenlohr<sup>2</sup> · Chen Zhang<sup>2</sup> · Jennifer Nastola<sup>1</sup> · Pratheek Shanthraj<sup>1</sup> · Franz Roters<sup>1</sup>

Received: 8 September 2016 / Accepted: 22 November 2016 / Published online: 15 March 2017  
© The Author(s) 2017. This article is an open access publication

**Abstract** Modern high-performing structural materials gain their excellent properties from the complex interactions of various constituent phases, grains, and subgrain structures that are present in their microstructure. To further understand and improve their properties, simulations need to take into account multiple aspects in addition to the composite nature. Crystal plasticity simulations incorporating additional physical effects such as heat generation and distribution, damage evolution, phase transformation, or changes in chemical composition enable the compilation of comprehensive structure–property relationships of such advanced materials under combined thermo-chemo-mechanical loading conditions.

**Electronic supplementary material** The online version of this article (doi:10.1007/s40192-017-0084-5) contains supplementary material, which is available to authorized users.

✉ Martin Diehl  
m.diehl@mpie.de

Philip Eisenlohr  
eisenlohr@egr.msu.edu

Chen Zhang  
zhanc43@egr.msu.edu

Jennifer Nastola  
j.nastola@mpie.de

Pratheek Shanthraj  
p.shanthraj@mpie.de

Franz Roters  
f.roters@mpie.de

<sup>1</sup> Microstructure Physics and Alloy Design, Max-Planck-Institut für Eisenforschung GmbH, Max-Planck-Straße 1, 40237 Düsseldorf, Germany

<sup>2</sup> Chemical Engineering and Materials Science, Michigan State University, East Lansing, MI 48824, USA

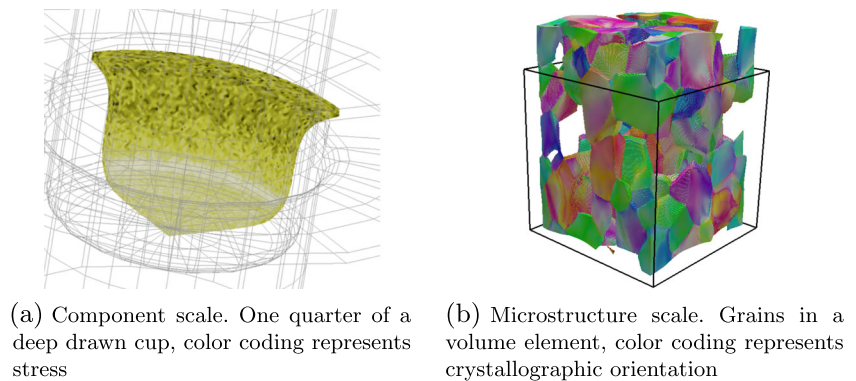
Capturing the corresponding thermo-chemo-mechanical response at the microstructure scale usually demands specifically adopted constitutive descriptions per phase. Furthermore, to bridge from the essential microstructure scale to the component scale, which is often of ultimate interest, a sophisticated (computational) homogenization scheme needs to be employed. A modular simulation toolbox that allows the problem-dependent use of various constitutive models and/or homogenization schemes in one concurrent simulation requires a flexible and adjustable file format to store the resulting heterogeneous data. Besides dealing with heterogeneous data, a file format suited for microstructure simulations needs to be able to deal with large (and growing) amounts of data as (i) the spatial resolution of routine simulations is ever increasing and (ii) more and more quantities are taken into account to characterize a material. To cope with such demands, a flexible and adjustable data layout based on HDF5 is proposed. The key feature of this data structure is the decoupling of spatial position and data, such that spatially variable information can be efficiently accommodated. For position-dependent operations, e.g., spatially resolved visualization, the spatial link is restored through explicit mappings between simulation results and their spatial position.

**Keywords** Crystal plasticity · Heterogeneous data · Microstructures · HDF5 · XDMF · DAMASK

## Introduction

Storage of results obtained by multiphysics simulations at the grain scale require a data format that can handle rather complex data structures in a problem-dependent way. Here, we present a flexible and expandable file structure based on

**Fig. 1** Simulation results at the two length scales considered within the proposed data format (Color figure online)



HDF5 [1, 2] that enables the storage of heterogeneous data at two (or more) scales.

Heterogeneity might be a direct consequence of the properties of the investigated microstructure, e.g. different slip and twin systems due to different lattice structures in a multiphase material. Additionally, when using different constitutive models with their specific output for certain phases or regions of the microstructure, the results will also differ between the selected models (see [3]).

For multiscale simulations, a homogenization scheme typically bridges between two (or even more) scales. It is therefore required to store aggregated/homogenized results as well as the underlying, lower-scale quantities. A direct consequence of such a multiscale approach is the existence of multiple data resulting in the lower scale being present “in parallel” at the same upper-scale point.

The requirements for a data structure that can store such heterogeneous results and the proposed solution will be discussed for the following two examples:

- Microstructure-scale simulation of a Ti-6Al-4V alloy
- Component-scale deep drawing simulation of a dual-phase (DP) steel sheet

We first introduce the general structure of the proposed file format and a possible implementation, discuss the details on the above mentioned examples, and finish with concluding remarks and an outlook on further usage of the proposed data format.

## A Flexible and Adjustable Data Structure

### General Concept

Consider an arbitrarily shaped body that is discretised into  $N_c$  (e.g. hexahedral) cells (equivalent to “material points”) with an associated set of data. Since the amount and type of data might be rather different among the different cells, a rigid file format that has the same fixed data

layout at every point as found, for instance, in [4, 5], is problematic. Instead, for a flexible, memory-efficient, and self-explanatory data structure, the simulation results are stored in HDF5 file format according to their intrinsic structure plus an additional mapping connecting such data to their spatial location as detailed in depth later. This has the additional advantage that each sub-model (such as a constitutive law for plasticity) can store its output results independently of that of others. For position-dependent operations such as spatially resolved visualization, an explicit mapping between simulation results and their spatial position is introduced. Writing an XML file following the XDMF<sup>1</sup> standard based on the mapping from results to spatial positions, the VTK library [6] can be used to visualize the results directly from the stored data using, for instance, Paraview<sup>2</sup> or VisIt<sup>3</sup>. An implementation of the above concept that is suitable for the Düsseldorf Advanced Material Simulation Kit (DAMASK) will be outlined in the following.

### Structure of DAMASK Results

DAMASK [3] is a multiphysics simulation toolbox developed at the Max-Planck-Institut für Eisenforschung. It enables the flexible combination of homogenization schemes at the component scale (1, Fig. 1a) and constitutive descriptions at the microstructure scale (2, Fig. 1b) for every material point. At both scales (1 and 2), universal (a) as well as model-specific (b) results exist and might be requested as output following the naming scheme shown in Table 1.

To uniquely identify every instance of each output type that is requested in a simulation, instances are named by prepending its description with an incrementing counter. Taking plasticity constitutive laws as an exemplary case, two instances would result from either different deformation

<sup>1</sup>[www.xdmf.org](http://www.xdmf.org)

<sup>2</sup>[www.paraview.org](http://www.paraview.org)

<sup>3</sup>[visit.llnl.gov](http://visit.llnl.gov)

**Table 1** Four types of outputs considered in the data structure

Component		Microstructure	
Universal	Model-specific	Universal	Model-specific
materialpoint	homogenization	constituent	constitutive

Component scale and microstructure scale both feature universal output (such as deformation gradients or temperature) as well as model-specific output (such as dislocation density or rate of hydrate formation)

physics (e.g., dislocation density-based versus phenomenological power law) or *same* deformation physics but different adjustable parameters (e.g., phenomenological power law for Cu and Nb in a composite microstructure).

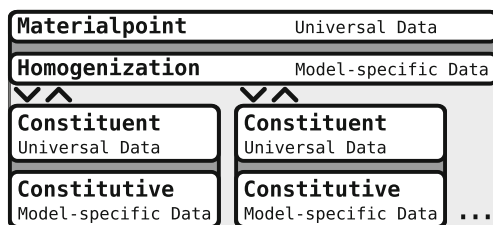
DAMASK requires the storage of data resulting from a material point model for a series of time steps. At each time increment, every DAMASK material point can have output at the component scale of (1a) universal material point quantities, such as average deformation, stress, or temperature, as well as (1b) specific homogenization scheme quantities. Similarly, results at the microstructure scale can be (2a) universal constituent quantities, such as deformation, stress, heat capacity, or temperature, and (2b) specific constitutive quantities, such as dislocation density, slip resistance, heat rate, or damage state. The structure of the output data at one material point is visualized in Fig. 2.

## Geometry

The cell geometry used in the simulation is stored in terms of initial nodal coordinates (`nodes`), cell connectivity (`connectivity`), and cell types (`cellType`) within a group—the HDF5 equivalent of a folder—called `geometry` as illustrated in Fig. 3.

## Mapping Between Data and Spatial Positions

Two mappings, from data to cells as well as from cells to data, allow for position-dependent manipulations of the data



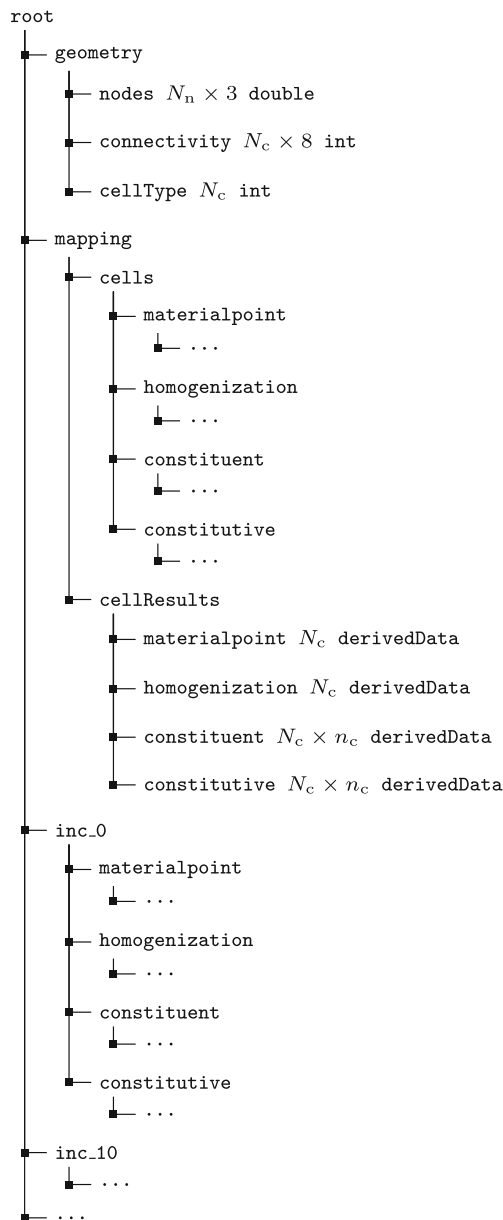
**Fig. 2** Output data on a materialpoint. A constitutive model calculates universal data with the help of model-specific data (state variables) and parses it to the constituent. A homogenization scheme calculates homogenized universal data from the universal data of individual constituents based on internal model-specific data (state variables)

sets and are maintained under the top-level group mapping (see Fig. 3).

The first mapping, called `cells`, is directly stored under the name of the respective instance. This one-dimensional integer array indexes every cell that is included in the respective instance of `materialpoint`, `homogenization`, `constituent`, or `constitutive`. Therefore, any spatial resolution at the microstructure scale *within* a single cell is not maintained, i.e., the same cell might be indexed by multiple constituent or constitutive results—even from the same instance.

As there might be multiple results available at any cell location, the second mapping from spatial positions to data, called `cellResults`, is more elaborate. First, results need to be distinguished among `materialpoint`, `homogenization`, `constituent`, and `constitutive`. This is accomplished by separate mappings for each of these types as individual arrays that span the overall geometry (see Fig. 3). Each of these arrays consists of HDF5 “derived datatypes”. For the cases of `materialpoint` and `homogenization` results, the derived datatype consists of a string “Name” to hold the name of the instance and one integer “Position” to store the index into the respective result array. This is shown schematically in Fig. 4 for two `materialpoint` outputs named `1_mp1` and `2_matpoint2` of length  $N_1$  and  $N_2$ , respectively. Here, “Name” in `mapping/cellResults` could point to “1\_mp1” or “2\_matpoint2” and “Position” must be a number  $1 \dots N_1$  or  $1 \dots N_2$ . More specifically, if the `cellResults` entry at position 4 is [Name:“1\_mp1”, Position:20], cell No. 4 holds data from the array position 20 of the data (termed `data1` and `data2`) in group `1_mp1`. For constituent and constitutive results, in contrast, multiple instances can be present at a materialpoint (i.e., a cell). Therefore, multiple of above pairs (instance name and result index) are stored per cell. This allows to express that, at a given cell, data from more than one group (or different results from the same group when the string is the same) might exist. A screenshot of the `HDFviewer`<sup>4</sup> is given in Fig. 5 to show how such a resulting two-dimensional array of derived datatypes is visualized.

<sup>4</sup>[www.hdfgroup.org/products/java/hdfview](http://www.hdfgroup.org/products/java/hdfview)



**Fig. 3** Geometry, data mappings, and time increments as the top-level hierarchy. Global geometric information are nodal coordinates (nodes), cell connectivity (connectivity), and cell types (cellType). The cell-to-results mapping is kept in cellResults and results-to-cell mapping is stored in cells and grouped by type. Each time increment inc.i splits results by type into the groups materialpoint, homogenization, constituent, and constitutive for every respective instance

The connections between cellResults and cells are exemplarily shown for materialpoint and constitutive results in Fig. 6: A one-to-one relation exists for the materialpoint results in each of the six cells, but, since four of these cells (1, 3, 4, and 5) contain multiple constitutive data, such a mapping does not exist for constitutive results.

In summary, these mappings allow to answer the two essential questions encountered in data visualization and analysis:

- cells: At which spatial position is the given data located?
- cellResults: Which data exist at a given spatial position?

For visualizing data, the cells mapping allows to establish an integer array in order to make use of the *coordinate* feature specified in the XDMF standard. Using this array in an XML file following the XDMF standard (“light data”) together with the HDF5 file containing the actual output data (“heavy data”) allows the use of the VTK library for visualizing the data spatially resolved (Fig. 1b).

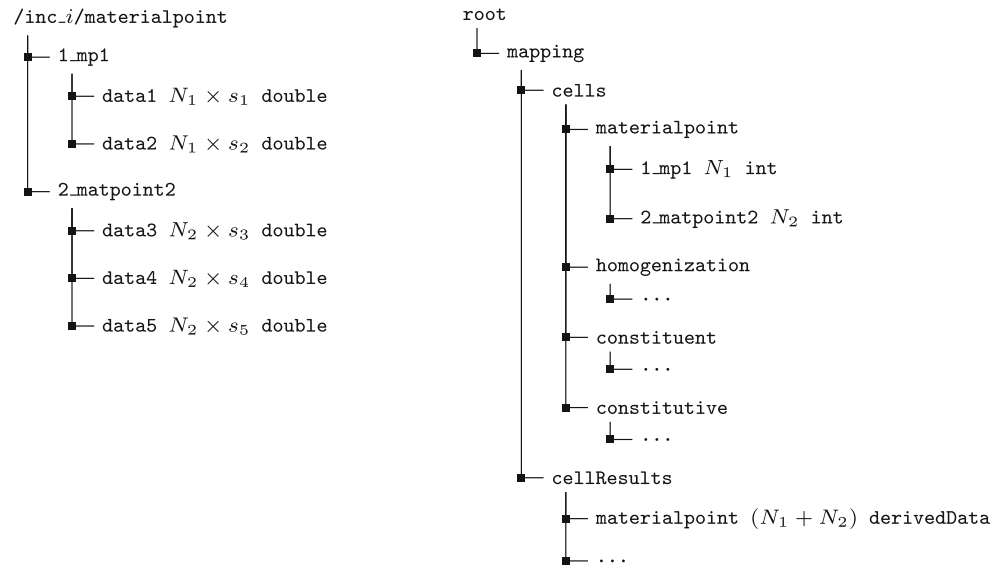
### Time Sequencing

As time provides a natural sequence of results, it serves as the primary categorization. All data of each time increment  $i$  (at time  $t_i$ ) is stored in a separate group called inc.i. This group itself contains four subgroups that hold output at the component scale (materialpoint and homogenization) and at the microstructure scale (constituent and constitutive). Figure 3 summarizes this top-level hierarchy.

**Materialpoint** The data per materialpoint instance is stored in a subgroup of the materialpoint group with the description of the instance prepended by a unique numerical identifier  $k$ . The first dimension of each data array equals the number  $N_k$  of points requested by the respective materialpoint instance  $k$ , while the subsequent dimensions depend on the data type (scalar, vector, tensor). Figure 4 shows an exemplary tree structure where two output sets called “mp1” and “matpoint2” with two and three vector outputs, respectively, of size  $s_{j=1,\dots,5}$  have been requested by the user. Possible output data could be, e.g., stress, strain, temperature, or concentration of a solute species.

**Homogenization** The homogenization results per instance are stored in a similar way as for the materialpoint results, yet a further level of subgroups is introduced to differentiate among the various categories of homogenization schemes. Consider, as an example, a homogenization scheme called “thermomechHomog” being the 2<sup>nd</sup> active one. Any mechanical output of that homogenization scheme would be stored in inc.i/homogenization/2\_thermomechHomog/mechanical, while thermal output would live in inc.i/homogenization/2\_thermomechHomog/thermal.

**Fig. 4** Materialpoint results of increment  $i$ . Materialpoint instance “mp1” (called 1\_mp1) contains two vectorial results (data1, data2) that apply in cells indexed by /mapping/cells/materialpoint/1\_mp1. The map cellResults/materialpoint links cells to positions in the result arrays of both materialpoint instances and contains  $N_1$  result indices for 1\_mp1 and  $N_2$  result indices for 2\_matpoint2



**Constituent** The storage of the constituent results, i.e., the universal results at the microstructure scale, follows the structure of the materialpoint results. Exemplary output data could be stress or strain (before homogenization, i.e., of each constituent at the microstructure scale) as well as plastic velocity gradient or heat rate.

**Constitutive** The storage of the constitutive results is similar to the structure of the homogenization results. Subgroups are introduced per constitutive type, for instance plasticity, damage, or thermal, that hold the specific outputs.

## Application Examples

Two common—but complex—simulation set ups are presented in the following to show how the presented data structure might be used. These examples are based on existing simulation studies and are selected in order to address a variety of different requirements.

### Micromechanics of a Ti-6Al-4V Alloy

Consider modeling the deformation of a two- or three-dimensional microstructure obtained, for example via EBSD as shown in [7]. By mirroring the measured microstructure in all spatial directions, a periodically repeated volume element is obtained on which periodic boundary conditions can be applied. For such a volume element,<sup>5</sup> a spectral method

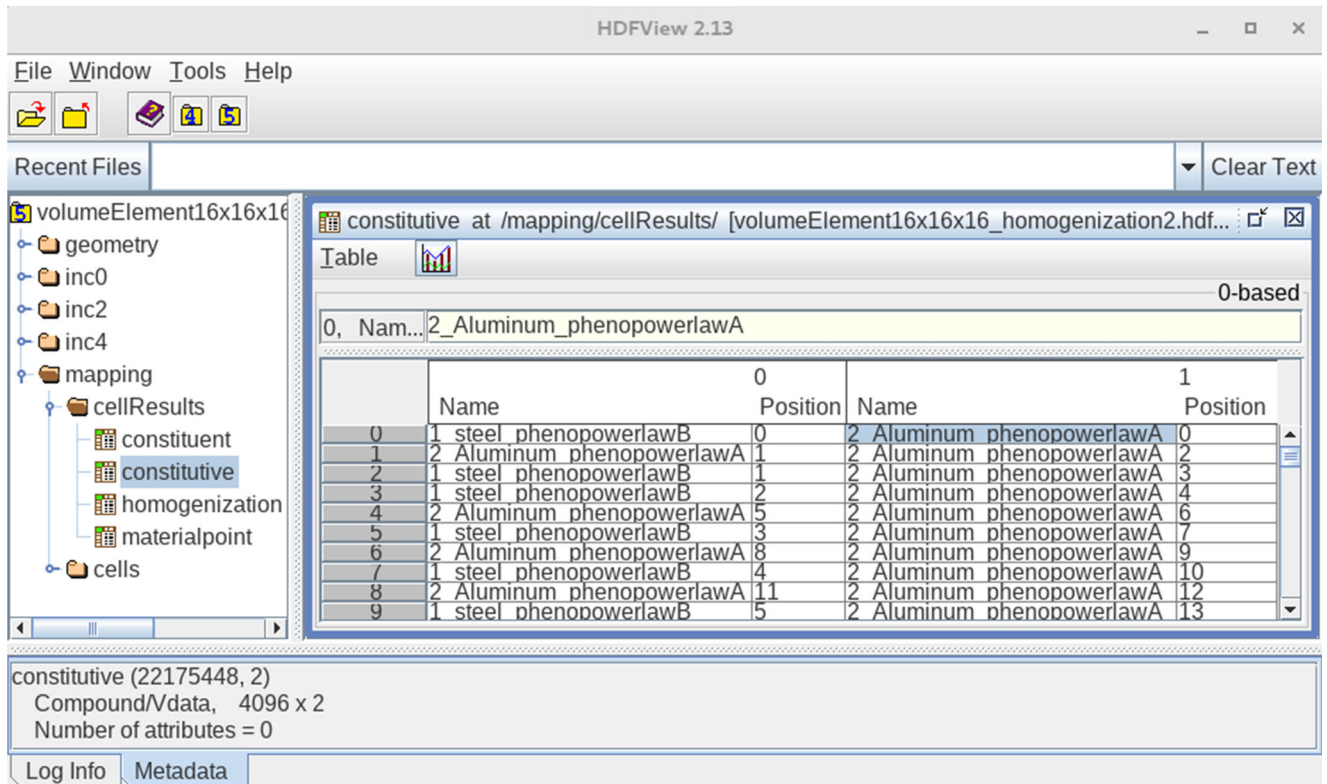
<sup>5</sup>The volume element might be “representative” if it contains all important microstructural features and is of sufficient size.

[8, 9] can be used as a boundary value solver. For such a spatially resolved microstructure simulation, no homogenization scheme is used<sup>6</sup>; hence, universal outputs at the component scale would be equivalent to the corresponding outputs at the microstructure scale. Universal outputs of interest at the microstructure scale are tensors of stress (“P”) and deformation gradient (“F”) as well as crystallographic orientation (in quaternion notation, denoted as “Q”). Since the stress and deformation gradient are equivalent at the constituent and the materialpoint level, writing them out at both levels would simply double the data. While this is generally left to the user as an option, here the data is stored only in constituent together with the orientation vector—which cannot be meaningfully defined in a homogenization context—leaving materialpoint empty.

The microstructure of the considered Ti-6Al-4V alloy contains volume fractions of 0.8 and 0.2 for the  $\beta$  (body-centered cubic) and  $\alpha$  (hexagonal) phase, respectively. If the structure is discretized into  $N_c = 512 \times 512$  cells,  $N_1 = 209715$  cells contain  $\beta$ -phase and  $N_2 = 52429$  contain  $\alpha$ -phase. The selected constitutive law for plasticity in the  $\alpha$ -phase is especially designed for modeling of hexagonal titanium following [10] for dislocation glide and [11] for mechanical twinning. Plastic deformation occurs due to slip on the  $\langle 11.0 \rangle \{00.1\}$ ,  $\langle 11.0 \rangle \{10.0\}$ ,  $\langle 11.0 \rangle \{\bar{1}1.1\}$ , and  $\langle 11.3 \rangle \{\bar{1}0.1\}$  systems and by twinning on the  $\langle \bar{1}0.1 \rangle \{10.2\}$ ,  $\langle 11.6 \rangle \{\bar{1}\bar{1}.1\}$ ,  $\langle 10.\bar{2} \rangle \{10.1\}$ , and  $\langle 11.\bar{3} \rangle \{11.2\}$  systems. The combined edge and screw dislocation density per slip system and the twin volume fraction

<sup>6</sup>In DAMASK, formally a Taylor homogenization scheme with one constituent is used that effectively just passes quantities from the constituent to the materialpoint level.



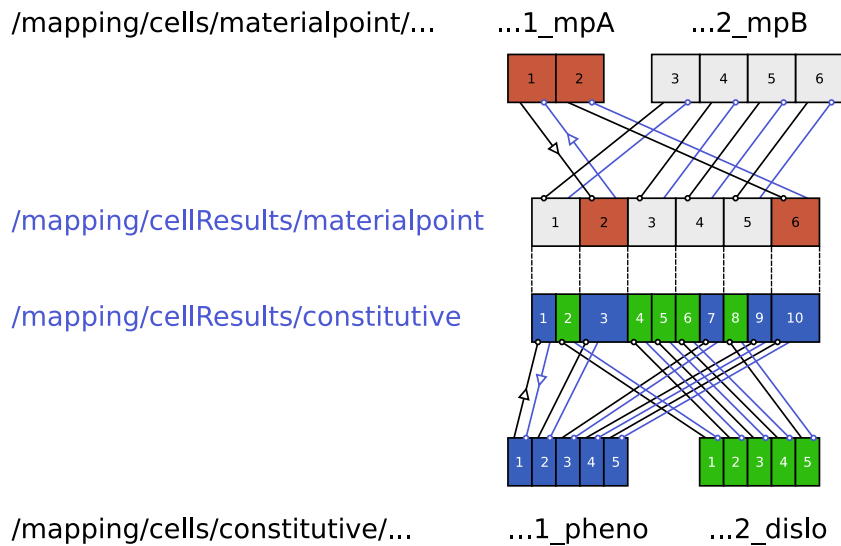


**Fig. 5** Screenshot of the *HDFviewer* visualizing a two-dimensional array mapping/*cellResults/constitutive* consisting of a derived datatype. In the example, every material point is made

up of two constituents with names “1\_steel\_phenopowerlawB” and “2\_Aluminum\_phenopowerlawA” (Color figure online)

per twin system will be reported by the constitutive model. In the absence of a more sophisticated model for plasticity in the  $\beta$ -phase, a phenomenological description [12] is

applied. The  $\langle 111 \rangle \{110\}$  and  $\langle 111 \rangle \{112\}$  slip systems exclusively carry plastic deformation in the body-centered cubic phase. The resistance to slip will be used as an output to



**Fig. 6** Relationship between */mapping/cellResults* and */mapping/cells* for materialpoint and constitutive results. For the six cells, a one-to-one mapping to the materialpoint output data of instances 1\_mpA and 2\_mpB exists. Such a mapping does not exist for the constitutive results

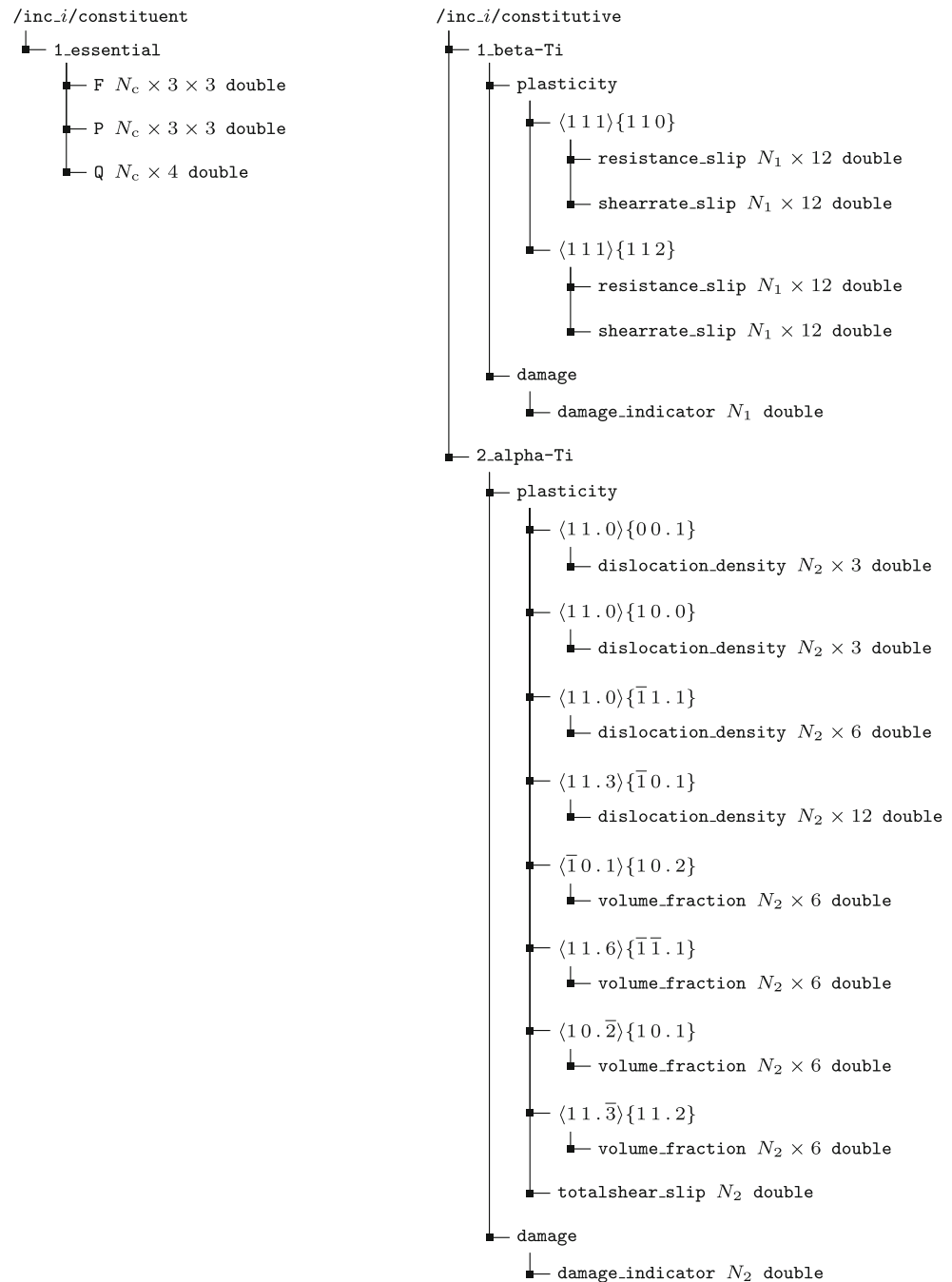
since cells might aggregate data. Here, cells 1, 3, 4, and 5 hold two constitutive outputs, where in cells 1, 4, and 5, this data comes from different instances, and in cell 3, it comes from two different positions within the same instance (2\_dislo) (Color figure online)

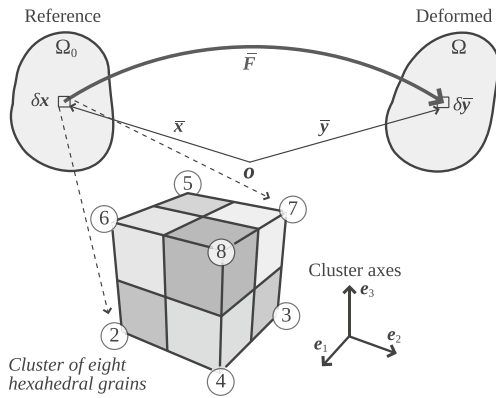
inspect the state of the material. As a further contribution to inelastic deformation, isotropic damage is considered in both phases following [13]. The variable  $\varphi$  indicating the material state of damage is of interest for post processing, especially for the visualization of crack surfaces. Figure 7 illustrates the data structure resulting from above setup of the spatially resolved microstructure simulation, where the plasticity laws internally subgroup outputs per deformation system family.

## Deep Drawing of a Dualphase Steel Sheet

Crystal plasticity simulations of engineering components on the meter scale require a homogenization scheme since the millions of individual grains contained in a component cannot be spatially resolved. Therefore, each integration point in a finite element method (FEM) simulation statistically represents the underlying microstructure by homogenizing between a few to several thousand grains. As a typical

**Fig. 7** Data structure of the micromechanical simulation example. See text for details





**Fig. 8** RGC grain cluster used to homogenize eight grain orientations at every materialpoint [14]

example for component-scale simulations, deep drawing of a sheet in combination with the relaxed grain cluster (RGC) homogenization scheme [14] as shown in [15] is used. An example of a cluster used for homogenization that contains  $n_c = 8$  constituents of two separate phases is shown in

Fig. 8. Each individual cluster contains too few constituents (each representing a single crystallographic orientation) to capture the macroscopic crystallographic orientation distribution (texture) that usually results in substantial anisotropy. Hence, the macro texture—measured via X-ray diffraction or through electron backscatter diffraction (EBSD)—is first discretized [16] into as many individual orientations as there are materialpoints times the number of orientations per cluster, and then these are distributed among the cluster constituents across all simulation cells.

Both phases (martensite and ferrite) are considered to have a body-centered cubic crystal structure with  $\langle 111 \rangle \{110\}$  and  $\langle 111 \rangle \{112\}$  slip systems. Crystal plasticity of these phases is described with the dislocation density-based and temperature-dependent constitutive law outlined in [17]. As heat generation due to plastic work dissipation is expected to have an influence on the plastic flow behavior for the fast deformation of high-strength steels, a temperature field is explicitly accounted for. Figure 9 illustrates the data structure containing stress “P”, deformation gradient “F”, and temperature “T” as materialpoint results,

**Fig. 9** Data structure of the deep drawing simulation example (note that  $N_1 + N_2 = n_c N_c$ )





stress “P” and deformation gradient “F” of each cluster constituent, and a number of constitutive results that might have been requested. Note that every materialpoint hosts eight clustered constituents; therefore, the number of overall constituent (as well as constitutive) results is eight times larger than the number of cells, reflected by `cellResults` holding eight “derivedData” entries per cell.

## Conclusion and Outlook

The presented data structure allows storage of not only data sets that have spatially invariant structure, but it can capture spatially varying data as obtained, for instance, from multiphysics micromechanical simulations. Therefore, this output capability has been implemented in the Düsseldorf Advanced Material Simulation Kit. The use of HDF5 as the underlying file format combines the advantages of plain text files, i.e., human readability, with those of (compressed) binary files, i.e., memory-efficient storage and fast access times. In addition to what has been outlined here, HDF5 features attributes that can be employed to hold supporting information of individual data (such as physical units) or of the simulation as a whole (such as configuration options used). Moreover, the structured data layout simplifies the development of numerous post processing tools that either generate and add derived data (e.g., strain measures from the deformation gradient) or statistically analyze the data. As a lightweight method to visualize the simulation results based on the existing HDF5 file, we envision the use of an adjustable XDMF wrapper.

**Acknowledgments** This research was carried out in the project *TCMPrecipSteel* of the SPP 1713 *Strong coupling of thermo-chemical and thermo-mechanical states in applied materials* of the Deutsche Forschungsgemeinschaft (DFG). The support of *G. Heber* from the *HDF Group* in designing the data structure is gratefully acknowledged.

**Funding Information** Open Access Funding provided by Max Planck Society.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

- Folk M, Heber G, Kozioł Q, Pourmal E, Robinson D (2011) An overview of the HDF5 technology suite and its applications, pp 36–47. <https://doi.org/10.1145/1966895.1966900>
- Schmitz GJ (2016) Microstructure modeling in integrated computational materials engineering (ICME) settings: can HDF5 provide the basis for an emerging standard for describing microstructures? *JOM* 68(1):77–83. <https://doi.org/10.1007/s11837-015-1748-2>
- Roters F, Eisenlohr P, Kords C, Tjahjanto DD, Diehl M, Raabe D (2012) DAMASK: the Düsseldorf Advanced Material Simulation Kit for studying crystal plasticity using an FE based or a spectral numerical solver. In: Cazacu O (ed) *Procedia IUTAM: IUTAM Symposium on Linking Scales in Computation: From Microstructure to Macroscale Properties*, vol 3. Elsevier, Amsterdam, pp 3–10. <https://doi.org/10.1016/j.piutam.2012.03.001>
- Jackson MA, Groeber MA, Uchic MD, Rowenhorst DJ, De Graef M (2014) h5ebds: an archival data format for electron backscatter diffraction data sets. *Integr Mater Manuf Innov* 3(1):4. <https://doi.org/10.1186/2193-9772-3-4>
- Groeber MA, Jackson MA (2014) DREAM.3D: A digital representation environment for the analysis of microstructure in 3D. *Integr Mater Manuf Innov* 3(1):5. <https://doi.org/10.1186/2193-9772-3-5>
- Schroeder W, Martin K, Lorensen B (2006) *The visualization toolkit*, 4th edn, Kitware
- Tasan CC, Hoefnagels JPM, Diehl M, Yan D, Roters F, Raabe D (2014) Strain localization and damage in dual phase steels investigated by coupled in-situ deformation experiments-crystal plasticity simulations. *Int J Plast* 63:198–210. <https://doi.org/10.1016/j.ijplas.2014.06.004>
- Moulinec H, Suquet P (1994) A fast numerical method for computing the linear and nonlinear properties of composites. *Comptes rendus de l'Académie des sciences. Série II, Mécanique, physique, chimie, astronomie* 318:1417–1423
- Eisenlohr P, Diehl M, Lebensohn RA, Roters F (2013) A spectral method solution to crystal elasto-viscoplasticity at finite strains. *Int J Plast* 46:37–53. <https://doi.org/10.1016/j.ijplas.2012.09.012>
- Alankar A, Eisenlohr P, Raabe D (2011) A dislocation density-based crystal plasticity constitutive model for prismatic slip in  $\alpha$ -titanium. *Acta Mater* 59(18):7003–7009. <https://doi.org/10.1016/j.actamat.2011.07.053>
- Wang H, Wu PD, Wang J, Tomé C. N. (2013) A crystal plasticity model for hexagonal close packed (HCP) crystals including twinning and de-twinning mechanisms. *Int J of Plast* 49:36–52. <https://doi.org/10.1016/j.ijplas.2013.02.016>
- Peirce D, Asaro RJ, Needleman A (1983) Material rate dependence and localized deformation in crystalline solids. *Acta Metall* 31(12):1951–1976. [https://doi.org/10.1016/0001-6160\(83\)90014-7](https://doi.org/10.1016/0001-6160(83)90014-7)
- Shanthraj P, Sharma L, Svendsen B, Roters F, Raabe D (2016) A phase field model for damage in elasto-viscoplastic materials. *Computer Methods in Applied Mechanics and Engineering* 312:167–185. <https://doi.org/10.1016/j.cma.2016.05.006>
- Tjahjanto DD, Eisenlohr P, Roters F (2010) A novel grain cluster-based homogenization scheme. *Model Simul Mater Sci Eng* 18:015006. <https://doi.org/10.1088/0965-0393/18/1/015006>
- Tjahjanto DD, Eisenlohr P, Roters F (2015) Multiscale deep drawing analysis of dual-phase steels using grain cluster-based RGC scheme. *Model Simul Mater Sci Eng* 23:045005. <https://doi.org/10.1088/0965-0393/23/4/045005>
- Eisenlohr P, Roters F (2008) Selecting sets of discrete orientations for accurate texture reconstruction. *Comput Mater Sci* 42(4):670–678. <https://doi.org/10.1016/j.commatsci.2007.09.015>
- Ma A, Roters F, Raabe D (2007) A dislocation density based constitutive law for BCC materials in crystal plasticity FEM. *Comput Mater Sci* 39:91–95. <https://doi.org/10.1016/j.commatsci.2006.04.014>