

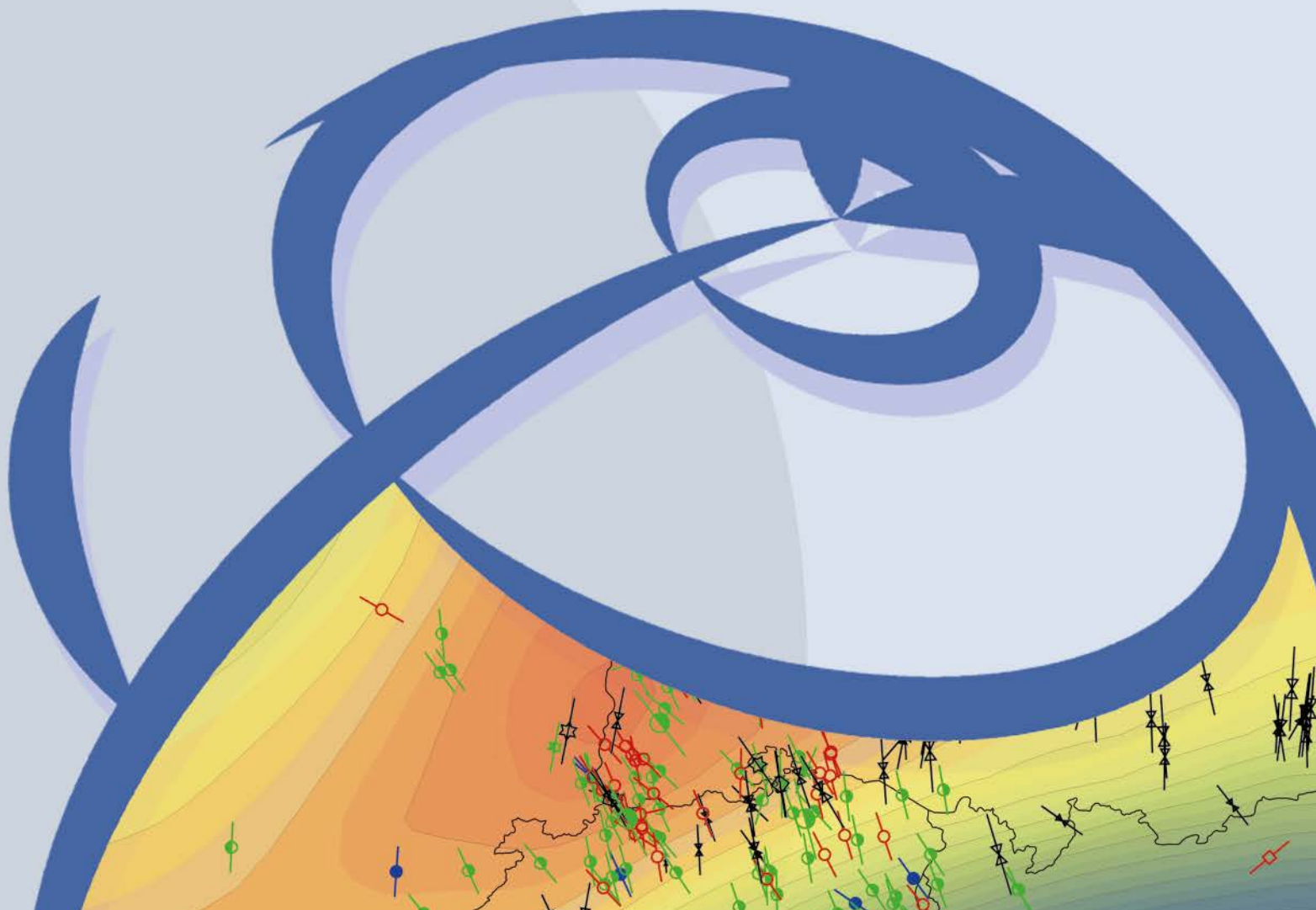


WSM WORLD STRESS MAP

WSM *Technical Report 21-01*

Manual of the Python Script HIPSTER v1.3

Moritz Ziegler



Recommended citation of the report:

Ziegler, M. O. (2021). Manual of the Python Script HIPSTER v1.3. World Stress Map Technical Report 21-01, GFZ German Research Centre for Geosciences.
DOI: <http://doi.org/10.48440/wsm.2021.001>

The software described in this report, including data examples, is published as:

Ziegler, M. O. (2021). Python Script HIPSTER v1.3. GFZ Data Services.
DOI: <http://doi.org/10.5880/wsm.2021.001>

The software described in this report can be downloaded from:

<http://github.com/MorZieg/hipster>

Previous version of the report:

Ziegler, M. O. (2019). Manual of the Python Script HIPSTER v1.0. World Stress Map Technical Report 19-03, GFZ German Research Centre for Geosciences.
DOI: <http://doi.org/10.2312/wsm.2019.003>

Previous versions of the software are published as:

Ziegler, M. O. (2019). Python Script HIPSTER v1.01. GFZ Data Services.
DOI: <http://doi.org/10.5880/wsm.2019.003>

Imprint

World Stress Map Project
GFZ German Research Centre for Geosciences

Telegrafenberg
D-14473 Potsdam
Published in Potsdam, Germany
January 2021
<http://doi.org/10.48440/wsm.2021.001>



WSM Technical Report 21-01

Manual of the Python Script HIPSTER v1.3

Moritz Ziegler

GFZ German Research Centre for Geosciences, Potsdam, Germany

Table of Contents

List of Tables.....	II
List of Figures.....	II
Abstract	3
Changelog v1.3	4
1 Introduction.....	5
2 Description of HIPSTER.....	7
3 Application.....	9
3.1 Settings	9
3.2 Input file	10
3.2.1 Abaqus input file.....	10
3.2.2 Apple PY input file	11
3.2.3 Abaqus element and material definition	12
3.3 Assign material properties to elements	13
3.4 HIPSTER Output file	13
3.4.1 Abaqus	13
3.4.2 MOOSE.....	15
4 Examples.....	17
4.1 Fully-functional example	17
4.2 Minimum working examples	18
5 Acknowledgement.....	19
References	19

List of Tables

Table 0-1 Structure of the GitHub repository3

Table 3-1 Description of variables in HIPSTER9

Table 4-1 Material properties in the example17

Table 4-2 Description of minimum working examples18

List of Figures

Figure 1-1 Homogeneous vs. inhomogeneous material properties5

Figure 2-1 Assignment of material properties 7

Figure 2-2 Distribution types.....8

Abstract

In geosciences 3D geomechanical-numerical models are used to estimate the in-situ stress state. In such a model each geological unit is populated with the rock properties Young's module, Poisson ratio, and density. Usually, each unit is assigned a single set of homogeneous properties. However, variable rock properties are observed and expected within the same geological unit. Even within small volumes large variabilities may occur.

The Python script HIPSTER (Homogeneous to Inhomogeneous rock Properties for Stress Tensor Research) provides an algorithm to include inhomogeneities in geomechanical-numerical models that use the solver Abaqus® or the MOOSE Framework. The user specifies the mean values for the rock properties Young's module, Poisson ratio and density, and their variability for each geological unit. The variability of the material properties is individually defined for each of the three rock properties in each geological layer. For each unit or unit subset HIPSTER generates a normal or uniform distribution for each rock property. From these distributions for each single element or subset of elements HIPSTER draws individual rock properties and writes them to a separate material file. This file defines different material properties for each element. The file is included in the geomechanical-numerical analysis solver deck and the numerical model is solved as usual.

The HIPSTER script files and example files are provided for download at <http://github.com/MorZieg/hipster>. Table 0-1 gives an overview of the repository and files including a short explanation.

Table 0-1 Structure of the GitHub repository

The folders and files in the GitHub repository <http://github.com/MorZieg/hipster>.

File Name	Explanation
hipster.py	Main script with auxiliary functions. See Page 9.
CITATION.bib	The recommended citation for the software.
LICENSE	The full GPL v3.0 license text.
README.md	Readme file that contains relevant information on the usage of the software.
examples/	Folder that contains example files.
/fully-functional	Abaqus® input file and Apple Py geometry with a model that contains three geological units and can be run in a solver. See page 17.
/mwe	Minimum working example that allows to test the different options of HIPSTER. See page 18.

Changelog v1.3

The following changes have been implemented compared to Version 1.01:

- Due to the end of the support for Python 2.x HIPSTER has been updated to run with Python 3.x.
- In addition to the assignment of material properties to individual elements, HIPSTER is now able to assign material properties to element sets as well.
- The range of output files has been extended to include the syntax of the MOOSE Framework.
- Several minor improvements in functionality and readability have been included:
 - The main function has been cleaned and code was outsourced.
 - A sanity check for the provided input is included.
 - Several functions were cleaned and made more comprehensive.
 - Additional comments have been included.

1 Introduction

Geomechanical-numerical models estimate the continuous in-situ stress state in a model volume. The approach is based on reference stress magnitude data records along with the geometry of the mechanically relevant geological units and their material properties. These properties are the Young's module, Poisson ratio, and density. In models, it is usually assumed that these material properties are homogeneous within the individual geological units. That means that all element that belong to a unit are assigned identical material properties. In reality, this is predominately not the case. On the contrary, most geological units are known to be inhomogeneous in their lithology and consequently in their material properties. Large scaled and observed trends or features delineated by material properties can be included in the model geometry. Still, apparently arbitrary changes in the lithology within a unit are rarely considered within a geomechanical-numerical model, although they may have a significant impact on the results.

It is usually not possible or feasible to observe the detailed 3D distribution of material properties within a geological unit. However, information on the range of material properties are often available. In order to address the statistical influence of inhomogeneities in the material properties in a geomechanical model, each element needs to be assigned individual material properties (Figure 1-1). The range of possible values for each property and each unit depends on the observed/measured values. If laboratory measurements indicate a very homogeneous unit within the model area, variations are small. On the other hand, if large inhomogeneities are observed, large variations from one element to the next are expected.

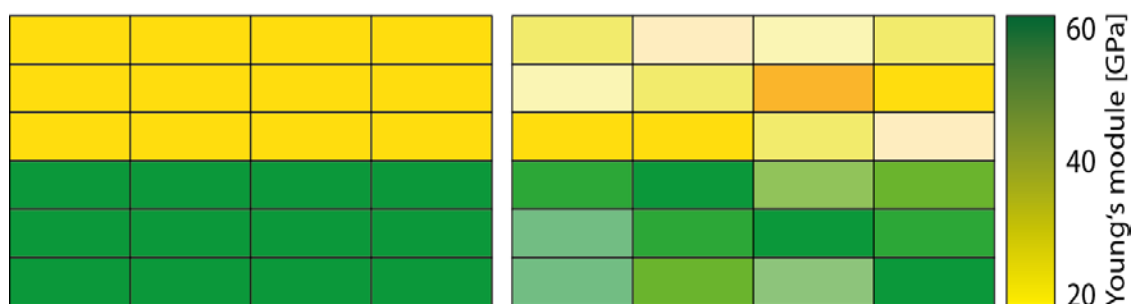


Figure 1-1 Homogeneous vs. inhomogeneous material properties

Colour-coded Young's module assigned to elements in two different geological units. Left: The two geological units have different but homogeneous material properties (yellow and green). Right: The two geological units have different and inhomogeneous material properties (different shades of yellow and green). The mean values of the material properties in each unit remain comparable to those in the left image.

The Python script HIPSTER (Homogeneous to Inhomogeneous rock Properties for Stress Tensor Research) allows to “inhomogenize” material properties in a geomechanical-numerical model for the solver Abaqus® and the Moose Framework. For each geological unit, the material parameters Young's module, Poisson ratio, and density, and their according variability

are provided by the user. HIPSTER then randomly selects values from the specified probability density function for each parameter and each element. Eventually, the material properties are written to a file that allows the direct use in an Abaqus® solver deck or can be included in an input file in MOOSE syntax.

2 Description of HIPSTER

HIPSTER (Homogeneous to Inhomogeneous rock Properties for Stress TENSOR Research) is a Python 3 script which provides a fully automated assignment of individual material properties to all elements or a selection of elements of a finite element mesh. Each element or element subset of a geological unit is assigned individual material properties that are retrieved from a user-defined distribution for the corresponding unit. Thereby, a representation in the model of inhomogeneous materials within geological units is achieved (Figure 2-1).

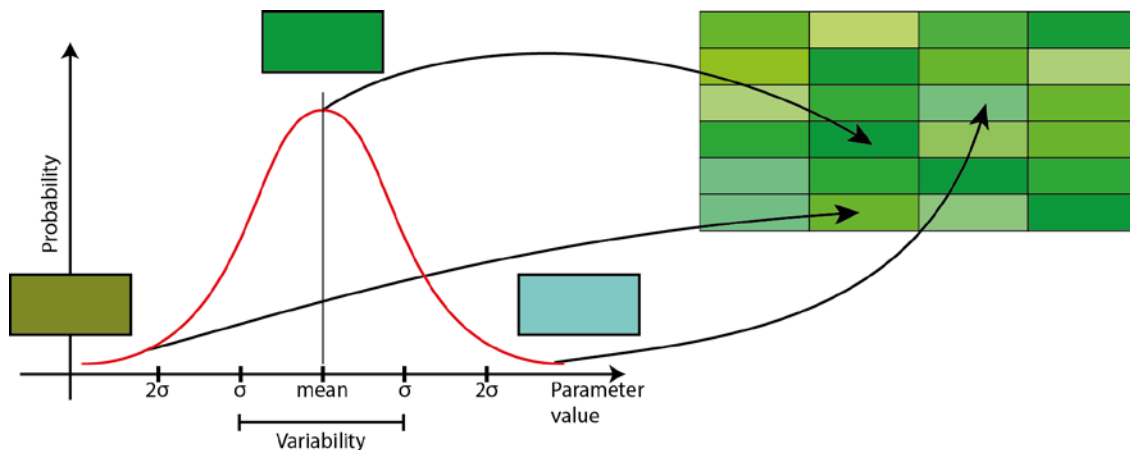


Figure 2-1 Assignment of material properties

For each individual element the value of a material property is drawn and assigned from a normal distribution defined for the corresponding lithological unit.

The user defines individual material properties (Young's module, Poisson ratio, and density) for each lithological unit present in the model. Each property is defined by its mean value and its variability. Thus, for each geological unit 6 values are defined. The mean and variability is completely independent for each material property and each geological unit. In order to disregard inhomogeneities for a material property the variability is set to 0. Thus, the influence of inhomogeneities can be evaluated, for example, (1) only for an individual geological unit, (2) an individual material property throughout all geological units, (3) an individual material property in an individual geological unit, or (4) all material properties in all units.

The initial idea of HIPSTER is to assign individual material properties to each single element. In certain scenarios it is not feasible or appropriate to do so. For example, very large models (>100,000 elements) may contain too many elements to assign material properties to individual elements in a Monte Carlo analysis that estimates the variability in the stress state due to material properties inhomogeneity. Furthermore, continuity is expected in the ansatz-functions between neighbouring elements. This averaging of the stress state as a result of the material properties smooths possible spikes in the stress state and reduces the variability as a result of the inhomogeneities in the material properties. To mitigate this, a lithological

unit can be subdivided into subsets of elements. Then each subset – instead of each element – is assigned individual material properties.

For each element or element subset, HIPSTER randomly draws values from the input data distribution. It supports two types of distribution from which the material parameters are randomly drawn: a normal distribution and a uniform distribution (Figure 2-2).

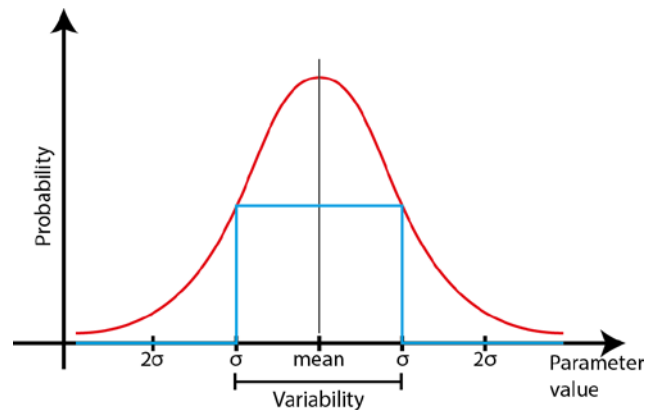


Figure 2-2 Distribution types

The probability for material property values is defined by the type of distribution. A normal distribution (red) has the potential for extreme outliers while a uniform distribution (blue) constrains the material properties to a defined range. Both types of distribution are defined by a mean value and a variability.

If the user chooses a normal distribution (red line in Figure 2-2), the variability is assumed to be the standard deviation σ . In the resulting distribution of material properties throughout the model, the probability to assign material properties similar to the mean value is high. In addition, extreme outliers of the mean value \pm several standard deviations may be present.

If the user chooses a uniform distribution (blue line in Figure 2-2), the variability is assumed to define the maximum range of possible values (mean value \pm variability). A uniform distribution contains only the material property values from the defined range and does not include any possible outliers. Furthermore, the material properties are equally distributed throughout the specified range.

Eventually, HIPSTER checks the reasonability of the resulting materials of each element (e.g. no negative values) and then writes the generated inhomogeneous material properties to a file that is in Abaqus® or MOOSE syntax. The file is included in the solver deck instead of the homogeneous material definition.

3 Application

HIPSTER is provided for free download from the GitHub repository <http://github.com/MorZieg/hipster>. As a Python 3 script several functions required for script execution are provided in one file, which can be copied and configured for each project. The additional Python package *numpy* is required. After the preparations are concluded, the script is run by the command *python hipster.py*.

3.1 Settings

HIPSTER requires user input, which is explained in the following. The user needs to set at least six variables that control the input and execution of the script. The variables are inserted into the input section of the Python script and are designed as shown in Table 3-1.

Table 3-1 Description of variables in HIPSTER

The variables are defined in the user input section of *hipster.py*.

Variable	Description
input_geometry	The name (and path) to a file that contains the input geometry in Abaqus® input format (*.inp or *.geom) or element set definitions (e.g. Apple PY output file *.set). It is mandatory that the file extensions are either *.inp/*.geom or *.set. If the assignment of material properties should be element-set-wise instead of element-wise append “elset” before the file extension. Example: <code>input_geometry = 'solver_deck_elset.inp'</code>
distrib	Specify the type of distribution for the material properties. Example: <code>distrib = 'normal' or 'uniform'</code>
strata	The names of all geological units (i.e. Abaqus® element sets) that are represented in the 3D geological model. The unit names must be identical to the element set names in Abaqus®. When using element-set-wise assignment make sure that the element sets that belong to the same unit have an identical unit name followed e.g. by numbers differentiating the element sets. Example: <code>strata = ['Unit_1', 'Unit_2', 'Unit_3']</code>
output_name	The filename for the output material definition file. If a file with the same name already exists in the directory in which HIPSTER is executed, it will be overwritten without any prior notification. For Abaqus® the file will be a *.mat file, for Moose a *.txt file. Example: <code>output_name = 'materials'</code>
solver	Specify the output syntax by the name of the solver Abaqus® or MOOSE. Example: <code>solver = 'abaqus'</code>
Unit_Name	For each unit represented in the model, the material property range is defined in an individual variable of the <i>rock</i> class. The variable name is required to be identical to the name in the variable <i>strata</i> . The variable contains: name of the unit, Young's module, Young's modules variability, Poisson ratio, Poisson ratios variability, density, density's variability. Example: <code>Top = rock('Top', 16E9, 2.5E9, 0.29, 0.015, 2200, 100)</code>

3.2 Input file

The input file for the mesh is required to be in Abaqus® file format. HIPSTER processes either Abaqus® input files (*.inp or *.geom) or element set files (*.set) provided by Apple PY (Ziegler et al., 2020), which also follow Abaqus® syntax. The type of file extension controls the further execution of the script and hence is crucial. So are the last few letters in the file name. If they are “*elset.*”, HIPSTER assigns the material properties element-set-wise instead of element-wise. In the following the syntax is explained.

3.2.1 Abaqus input file

If the file is exported from a pre-processor in Abaqus® syntax (e.g. from HyperMesh), HIPSTER usually works out of the box. This is also the case if the solver deck has already been run in Abaqus® or Moose. In the Abaqus® input file, four types of lines relevant for HIPSTER can be distinguished. They are read by the function `elementread`:

1. Comment lines are marked by `**` and are ignored by HIPSTER.
2. Keyword lines define nodes, elements, sets, etc. and begin with `*` and a following keyword. Only the keyword lines for element (`*ELEMENT`) definitions are relevant for HIPSTER. `*ELEMENT` initiates the definition of elements in sets. They occur more or less frequently, depending on the number of element sets and element types present in the model. HIPSTER requires at least one element set but supports more than one. In the keyword line the name of the element set and the type of elements defined (brick elements, tetrahedrons, wedges) is described. The name of the element set is required to be identical to the name defined in the HIPSTER variable `strata`.
3. Data lines follow keywords. They contain the information that is extracted by the function `elementread`. An element is defined by comma-separated numbers: The element ID is followed by a number of node IDs according to the number of nodes required for the corresponding element type. HIPSTER only reads the element ID. For HIPSTER to work as designed, each element definition needs to end without a comma. If a comma is used, the software assumes that the definition of the element continues in the next line.
4. The file end is marked by `*****`. If the end of the file is not marked by this, HIPSTER will end up in an infinite loop.

An example for an input file follows. Three dots indicate additional data lines left out here in order to keep the example short.

```
**      Template:  ABAQUS/STANDARD 3D
**
*NODE
```

```

1,    0.0,  1.0,  0.0
2,    1.0,  1.0,  0.0
...
*ELEMENT,TYPE=C3D4,ELSET=Quaternary
1,    41,   42,   43,   44
2,    45,   46,   47,   48
...
*ELEMENT,TYPE=C3D8R,ELSET=Quaternary
100,  1,    2,    3,    4,    5,    6,    7,
8
101,  5,    6,    7,    8,   11,   12,   13,
14
...
*ELEMENT,TYPE=C3D4,ELSET=Cretaceous
200,  51,   52,   53,   54
201,  55,   56,   57,   58
*****

```

3.2.2 Apple PY input file

If an Apple PY (Ziegler et al., 2020) file is chosen as input file, HIPSTER only reads Abaqus® element set information from this file. For the Python function `elsetread` the following lines are relevant in the input file:

1. Comment lines are marked by `**` and are ignored by HIPSTER.
2. Keyword lines define nodes, elements, sets, etc. and begin with `*` and a following keyword. Only the keyword lines for element sets (`*ELSET`) definitions are regarded here. HIPSTER requires at least one element set but supports more than one. In the keyword line, the name of the element set is stated. The name has to be identical with the name defined in the variable *strata*.
3. Data lines follow keywords. They contain the information that is extracted by this script. An element set contains comma-separated element IDs. For HIPSTER to work as designed, each element set needs to end without a comma. If a comma is used, the software assumes that the definition of the element set continues in the next line.
4. The file end is marked by `*****`. If the end of the file is not marked by this, HIPSTER will end up in an infinite loop.

An example for an Apple Py element set file follows. Three dots indicate additional data lines left out here in order to keep the example short.

```

**   Element set file created by Apple Py v1.01
**
*ELSET, ELSET=Quaternary
1,    2,    3,    4,    5,    6,    7,    8,
5,    6,    7,    8,   11,   12,   13
*ELSET, ELSET=Basalt
51,   52,   53,   54,   55,   56
*****

```

3.2.3 Abaqus element and material definition

Abaqus® requires three pieces of information in order to assign material properties to an element. An element is defined and at the same time assigned to an element set (here “Top”) by the following Abaqus command.

```
*ELEMENT, TYPE=C3D8R, ELSET=Top
    1, 2228, 2064, 264, 2229, 2225, 2063, 263,
    2226
    2, 2231, 2065, 265, 2232, 2228, 2064, 264,
    2229
```

Initially, the elements are grouped and assigned to sets corresponding to their configuration (here C3D8R) and lithological unit (here Top). The first column contains the element ID followed by the number of nodes that constitute the element. Each of such an element set usually contains a large number of elements but could also contain only as few as one element.

Alternatively, an element set definition can contain only element IDs of elements whose constituting nodes are defined elsewhere.

```
*ELSET, ELSET=Top
    1, 2, 3, 4, 6, 8,
    22, 34, 32
```

If an element-set-wise assignment of material properties is desired, all element sets belonging to a unit have to contain the unit name in their element set name. For example, element sets belonging to unit *Top* can be named *Top_1*, *Top_2*, etc.

For each individual combination of material properties, an individual material instance with a unique name is created. Each material instance contains values of the elastic properties Young’s module, Poisson ratio, and density.

```
*MATERIAL, NAME=Mat-Sediments
*ELASTIC, TYPE = ISOTROPIC
35.0E+09 , 0.25 , 0.0
*DENSITY
2600
```

Eventually, element sets are assigned a material via solid sections. This combines the name of the element set with the name of the material.

```
*SOLIDSECTION, ELSET=Top, MATERIAL=Mat-Sediments
```


3.3 Assign material properties to elements

HIPSTER reads the element IDs of the elements of all units and the element set names. If an element-wise assignment of material properties is desired, the individual material properties for each element are drawn from the user defined distribution of material properties of the accordingly named unit. Furthermore, individual element sets containing each one element are created and the material properties are assigned.

If an element-set-wise assignment of material properties is desired, each element set which contains the name of a unit is assigned individual material properties according to the units specifications. No new element sets need to be generated.

The material properties are drawn from the specified range using either the Python numpy function *random.normal* or for a uniform distribution the Python numpy function *random.uniform*. Before the final assignment, the material properties are checked for reasonability. Negative values are not allowed, as well as values larger than 0.49 for the Poisson ratio. If such values occur, new material properties are drawn from the distribution for the affected elements. The assigned materials are directly written to the output files.

If one material property of a unit has a variability of 0, the mean value of this property is assigned to all elements of that unit. If all three material properties have a user-defined variability of 0, HIPSTER automatically assigns the user-defined mean value for all three material properties to all elements of that unit. In this case, HIPSTER does not create individual material definitions and element sets for each element of the corresponding unit in order to speed up the runtime and reduce file size.

3.4 HIPSTER Output file

HIPSTER produces two output files that are added to the Abaqus® or MOOSE input files. They are used instead of a regular material property definition. The two output files are distinguished by the Poisson ratio. In the material file for modelling a lithostatic stress state, the Poisson ratio is set to 0.49 in every element in order to provide large horizontal stresses (*_lithostatic.*). This is useful for some modelling approaches. The second output file contains reasonable Poisson ratios as defined by the user.

3.4.1 Abaqus

The files HIPSTER produces for usage in Abaqus®, contain three separate sections that are exemplified in the following. In the first section, the Abaqus® material properties are defined for each single element of each geological unit. The material definitions are grouped according to the units. The Young's module precision is 1000 Pa, the Poisson ratio is rounded to the 3rd digit, and the density is rounded to 1 kg/m³. The material names are composed of the

name of the geological unit and the element ID together with the prefix “rock”. Three dots indicate further data lines left out here in order to keep the example short.

```
** Inhomogeneous material properties
** Created by HIPSTER v1.3 - http://github.com/MorZieg/hipster
**
** Material definitions
**
** Bottom
**
*MATERIAL, NAME=rock_Bottom_100
*ELASTIC, TYPE = ISOTROPIC
6.1193923e+10, 0.255, 0.0
*DENSITY
2891
**
*MATERIAL, NAME=rock_Bottom_101
*ELASTIC, TYPE = ISOTROPIC
6.2438611e+10, 0.253, 0.0
*DENSITY
2832
...
**
** Top
**
*MATERIAL, NAME=rock_Top_1
*ELASTIC, TYPE = ISOTROPIC
1.1647865e+10, 0.294, 0.0
*DENSITY
2102
**
*MATERIAL, NAME=rock_Top_2
*ELASTIC, TYPE = ISOTROPIC
8.025761e+09, 0.278, 0.0
*DENSITY
2249
...
```

The second section contains element set definitions. Since in Abaqus® it is not possible to assign materials directly to an element, the detour via element sets has to be made. For each individual element, an element set is created that contains only this single element. The name of the element set is composed of the attributed geological unit and the element number. No element sets are generated if an element-set-wise assignment is chosen.

```
** Element sets
*ELSET, ELSET=Bottom_100
100
*ELSET, ELSET=Bottom_101
101
```

```

...
*ELSET, ELSET=Top_1
1
*ELSET, ELSET=Top_2
2
...

```

Eventually, the material properties are assigned to the element sets using the Abaqus® command `*SOLIDSECTION`. Now each element has its individual material properties assigned.

```

** Solidsections
**
** Top
*SOLIDSECTION,ELSET=Top_100, MATERIAL=rock_Top_100
*SOLIDSECTION,ELSET=Top_101, MATERIAL=rock_Top_101
...
**
** Bottom
*SOLIDSECTION,ELSET=Bottom_1, MATERIAL=rock_Bottom_1
*SOLIDSECTION,ELSET=Bottom_2, MATERIAL=rock_Bottom_2
...

```

It becomes evident that for models with many elements the material definition files may have a large physical size. Accordingly, the runtime of HIPSTER increases with model size.

The output files are integrated into the solver deck file by the include command. Instead of the material definitions (`*MATERIAL`) and the assignment of materials to element sets (`*SOLIDSECTION`), the HIPSTER output file is included with a single line `*INCLUDE, INPUT=material.mat`. Therefore, the material file has to be located in the same folder as the solver deck/input file. Then the solver deck is run as usual.

3.4.2 MOOSE

HIPSTER generates two `*.txt` files whose content is included in the MOOSE input file. They contain the material properties for a lithostatic stress state and for the final stress state. The material definitions are grouped according to the units. The Young's module precision is 1000 Pa, the Poisson ratio is rounded to the 3rd digit, and the density is rounded to 1 kg/m³. In the following a brief example is presented. Three dots indicate further data lines that are left out in order to keep the example short.

```

# Inhomogeneous material properties
# Created by HIPSTER v1.3 - http://github.com/MorZieg/hipster
# 2020-09-30T09:40:39
#
[Material]
  [Top_1_Young_Poisson]

```

```

        type = ComputeIsotropicElasticityTensor
        block = 'Top_1_HEX8'
        youngs_modulus = 2.49e+10
        poissons_ratio = 0.252
    []
    [Top_1_dens]
        type = GenericConstantMaterial
        block = 'Top_1_HEX8'
        prop_names = density
        prop_values = 2643
    []
    [Top_2_Young_Poisson]
        type = ComputeIsotropicElasticityTensor
        block = 'Top_2_HEX8'
        youngs_modulus = 2.76e+10
        poissons_ratio = 0.242
    []
    [Top_2_dens]
        type = GenericConstantMaterial
        block = 'Top_2_HEX8'
        prop_names = density
        prop_values = 2658
    []
...
    [stress]
        type = ComputeLinearElasticStress
    []
[]

```

If an element-wise assignment of material properties is chosen, an additional file is produced that contains element set definitions. For each individual element, an element set is created that contains only this single element. The name of the element set is composed of the attributed geological unit and the element number. No element sets are generated if an element-set-wise assignment is chosen. This file needs to be included in the geometry file before the solver deck is run as usual.

4 Examples

4.1 Fully-functional example

In order to learn the functionality of HIPSTER, a working input file is provided that can be used in a solver. The exemplified model contains three geological units: Top, Middle, and Bottom (Table 4-1). HIPSTER is used to assign inhomogeneities of all three material properties to the Middle unit, which is a key area for exploration. In the Top unit, only inhomogeneities in the density are of interest. The Bottom unit is assumed to be homogeneous.

Table 4-1 Material properties in the example

The material properties mean values and variabilities in the example.

Unit	Young's module [GPa]		Poisson ratio		Density [kg/m ³]	
	mean	variability	mean	variability	mean	variability
Top	25	0	0.25	0.00	2500	200
Middle	35	2.5	0.21	0.02	2625	250
Bottom	45	0	0.24	0.00	2850	0

A fully functional Abaqus® solver deck is provided in the file `solver_deck.inp`. It contains node definitions and three element sets that constitute three geological units. The file `elements.set` is a reduced (but otherwise original) Apple PY output file with three geological units. The according HIPSTER code is provided in the following.

```
input_geometry = 'solver_deck.inp'
distrib = 'uniform'
strata = ['Top', 'Middle', 'Bottom']
output_name = 'test'
solver = 'abaqus'

Top = rock('Top', 25E9, 0, 0.25, 0.0, 2500, 200)
Middle = rock('Middle', 35E9, 2.5E9, 0.21, 0.02, 2625, 250)
Bottom = rock('Bottom', 45E9, 0, 0.24, 0, 2850, 0)
```

4.2 Minimum working examples

In order to be able to test the various input and output options, four minimum working examples are provided. They are not intended to run in a solver but still feature a realistic geometry code so that HIPSTER can be tested and the different outputs visualised. The four example files represent the four different cases of input situations and are described in detail in Table 4-2. The following HIPSTER variables can be used and only need adaptation of the `input_geometry` variable.

```
input_geometry = 'mwe_model_elset.set'  
distrib = 'uniform'  
strata = ['Top', 'Bottom']  
output_name = 'mwe'  
solver = 'moose'
```

```
Top = rock('Top', 25E9, 0, 0.25, 0, 2500, 0)  
Bottom = rock('Bottom', 25E9, 2.5E9, 0.25, 0.1, 2500, 500)
```

Table 4-2 Description of minimum working examples

The variables are defined in the user input section of `hipster.py`.

Name	Description
<code>mwe_model.geom</code>	A simple model that contains two units (Top, Bottom) which should be assigned individual material properties for each element.
<code>mwe_model_elset.geom</code>	A simple model that contains two units (Top, Bottom). Each unit is subdivided into four element subsets. Each subset should be assigned individual material properties based on a distribution defined for the according lithological unit.
<code>mwe_elements.set</code>	Element sets generated by APPLE PY for a simple model that contains two units (Top, Bottom) which should be assigned individual material properties for each element.
<code>mwe_elements_elset.set</code>	Element sets generated by APPLE PY for a simple model that contains two units (Top, Bottom). Each unit is subdivided into four element subsets. Each subset should be assigned individual material properties based on a distribution defined for the according lithological unit.

5 Acknowledgement

The author would like to thank Sophia Morawietz and Oliver Heidbach for discussion and comments. Furthermore, the author would like thank Solveig Strutzke for typesetting and Kirsten Elger for final checks and supporting the publication.

The work leading to these results has received funding from Germany's Federal Ministry for Education and Research under the FONA/GEO:N programme under grant agreement No. 03G0869A (Project SUBI). The authors acknowledge the financial support for the iCross project by the Federal Ministry of Education and Research (project number 02NUK053D), the Helmholtz Association and the Helmholtz Centre Potsdam, GFZ German Research Centre for Geosciences.

References

Ziegler, Moritz, Ziebarth, Malte, & Reiter, Karsten. 2020. Manual of the Python Script Apple PY v1.3. *WSM Technical Report; 20-02*, doi: 10.2312/wsm.2020.002.

