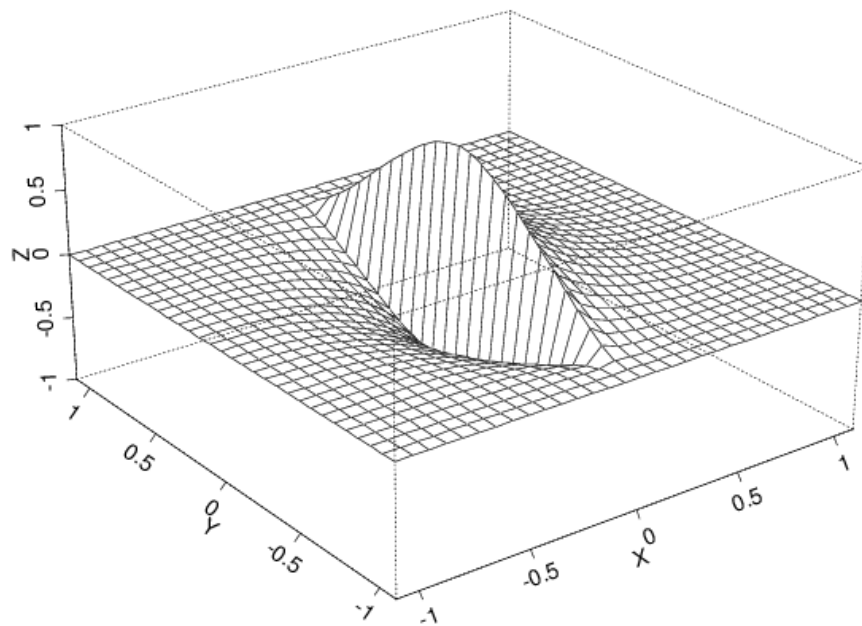


HAVANA user manual

Version 6.0



Note no
Authors

Date

SAND/03/10
Per Røe
Frode Georgsen
Anne Randi Syversveen
April 7, 2010

The authors

Several people are, or have been, involved in the development of Havana at NR, including Kristin L. Munthe, Petter Mostad, Geir Aamodt, Jon Gjerde, Bjørn Fredrik Nielsen, Oddvar Lia, Knut Utne Hollund, Ariel Vazquez Almendral, Christian Skaug, Harald H. Soleng, Per Røe, Frode Georgsen, Bjørn Fjellvoll and Anne Randi Syversveen.

Norwegian Computing Center

Norsk Regnesentral (Norwegian Computing Center, NR) is a private, independent, non-profit foundation established in 1952. NR carries out contract research and development projects in the areas of information and communication technology and applied statistical modeling. The clients are a broad range of industrial, commercial and public service organizations in the national as well as the international market. Our scientific and technical capabilities are further developed in co-operation with The Research Council of Norway and key customers. The results of our projects may take the form of reports, software, prototypes, and short courses. A proof of the confidence and appreciation our clients have for us is given by the fact that most of our new contracts are signed with previous customers.

Title	HAVANA user manual
Authors	Per Røe , Frode Georgsen , Anne Randi Syversveen
Date	April 7, 2010
Publication number	SAND/03/10

Abstract

HAVANA is a program for simulating subseismic faults in petroleum reservoirs, and for integrating the effects of these faults into the reservoir description. The HAVANA project has a long history, the original sponsors being Statoil, BP, and Norsk Hydro. Other sponsors include Conoco Norge AS, Saga Petroleum AS and Centre for Integrated Petroleum Research at University of Bergen.

Keywords	fault, simulation, sealing, stochastic, software
Target group	HAVANA users
Availability	Open
Project	
Project number	
Research field	Fault modelling
Number of pages	44
Copyright © 2010	Norwegian Computing Center

Contents

1	User Reference	7
1.1	Model file syntax	7
1.2	Commands common to the whole program	7
1.2.1	Command: INPUT_AND_OUTPUT_DIRECTORY	7
1.2.2	Command: INPUT_DIRECTORY	8
1.2.3	Command: OUTPUT_DIRECTORY	8
1.2.4	Command: READ	8
1.2.5	Command: SEED	8
1.2.6	Command: INPUT_SEED_FILE	9
1.2.7	Command: LEVEL_OF_INFORMATION	9
1.2.8	Command: OUTPUT_LOG	9
1.3	Input of fault model from RMS	9
1.3.1	Command: INPUT_FAULTS	9
1.3.2	Command: FAULT_LINE_POLYGON	10
1.3.3	Command: SFM_PARAMETERS	10
1.3.4	Command: VARIO_TYPE	10
1.3.5	Command: FAULT_DISPLACEMENT_LENGTH	11
1.3.6	Command: FAULT_LENGTH_HEIGHT	11
1.4	Simulate: Simulate sub-seismic faults	11
1.4.1	Command: INPUT_FAULTS	11
1.4.2	Command: FAULT_LINE_POLYGON	12
1.4.3	Command: SFM_PARAMETERS	12
1.4.4	Command: VARIO_TYPE	12
1.4.5	Command: SFM_FAULT_DISPLACEMENT_LENGTH	12
1.4.6	Command: SFM_FAULT_LENGTH_HEIGHT	12
1.4.7	Command: OUTPUT_HAVANA_FAULTS	12
1.4.8	Command: FAULTS_STATISTICS	12
1.4.9	Command: SIMULATION_VOLUME	13
1.4.10	Command: VERTICAL_BUFFER_SIZE	13
1.4.11	Command: INPUT_FAULT_CENTER_LINES	13
1.4.12	Command: FAULT_CENTER_LINES_OPTION	14
1.4.13	Command: INPUT_WELL_PATHS	14
1.4.14	Command: INPUT_WELLOBS_OF_FAULTS	15
1.4.15	Command: INPUT_WELLOBS_OF_NOFAULTS	15
1.4.16	Command: NUMBER_OF_FAULTS	15
1.4.17	Command: RELATIVE_INTENSITY	15
1.4.18	Command: RELATIVE_INTENSITY_GRID	17
1.4.19	Command: DISPLACEMENT_INTENSITY	17
1.4.20	Command: DISPLACEMENT_INTENSITY_GRID	17
1.4.21	Command: OUTPUT_DISPLACEMENT_INTENSITY	17
1.4.22	Command: DISPLACEMENT_INTENSITY_PARAMETERS	18
1.4.23	Command: DISPLACEMENT	18

1.4.24	Command: FAULT_DISPLACEMENT_LENGTH	20
1.4.25	Command: FAULT_LENGTH_HEIGHT	20
1.4.26	Command: FAULT_AVERAGE_REVERSED DRAG	21
1.4.27	Command: ORIENTATION_GROUPS	21
1.4.28	Command: STRIKE	21
1.4.29	Command: DIP	22
1.4.30	Command: FAULT_TRUNCATION	23
1.4.31	Command: REPULSION	23
1.4.32	Command: DISPLACEMENT_WEIGHT	24
1.4.33	Command: NUMBER_OF_FAMILIES	24
1.4.34	Command: CHILDREN_PARAMETERS	25
1.4.35	Command: CHILDREN_STRIKE	25
1.4.36	Command: CHILDREN_DIP	25
1.4.37	Command: CHILDREN_PARAMETERS_GENERAL	25
1.4.38	Command: CHILDREN_PARAMETERS_MOTHER_TYPE	27
1.4.39	Command: RELAY_RAMPS	27
1.4.40	Command: NUMBER_OF_ITERATIONS	28
1.4.41	Command: FAULTNAME_PREFIX	28
1.5	ModifySurfaceFault: Change SFM fault properties	29
1.5.1	Command: INPUT_FAULTS	29
1.5.2	Command: FAULT_LINE_POLYGON	29
1.5.3	Command: SFM_PARAMETERS	29
1.5.4	Command: VARIO_TYPE	29
1.5.5	Command: FAULT_DISPLACEMENT_LENGTH	29
1.5.6	Command: FAULT_LENGTH_HEIGHT	29
1.5.7	Command: OUTPUT_FAULTS	29
1.5.8	Command: HORIZON	29
1.5.9	Command: TRANSFORM	29
1.5.10	Command: SCALE_DISPLACEMENT	30
1.5.11	Command: LOCAL_ROTATION	30
1.6	SimulateSurfaceFault: Simulate SFM faults	31
1.6.1	Command: INPUT_FAULTS	31
1.6.2	Command: FAULT_LINE_POLYGON	31
1.6.3	Command: SFM_PARAMETERS	31
1.6.4	Command: VARIO_TYPE	31
1.6.5	Command: FAULT_DISPLACEMENT_LENGTH	31
1.6.6	Command: FAULT_LENGTH_HEIGHT	31
1.6.7	Command: OUTPUT_FAULTS	31
1.6.8	Command: FAULT_VOLUME_WIDTH	31
1.6.9	Command: VARIOGRAM_GEOMETRY	31
1.6.10	Command: DISTRIBUTION	32
1.6.11	Command: FAULT_PICKS_FILE	32
1.6.12	Command: FAULT_PICK_UNCERTAINTY	32
1.7	FaultTip: Fault displacement field modeling	34
1.7.1	Command: INPUT_FAULTS	34
1.7.2	Command: FAULT_LINE_POLYGON	34
1.7.3	Command: SFM_PARAMETERS	34
1.7.4	Command: VARIO_TYPE	34
1.7.5	Command: FAULT_DISPLACEMENT_LENGTH	34

1.7.6	Command: FAULT_LENGTH_HEIGHT	34
1.7.7	Command: OUTPUT_FAULTS	34
1.7.8	Command: SEISMIC_RESOLUTION	34
1.7.9	Command: SIMULATION	34
1.7.10	Command: HORIZON	34
A	File formats	36
A.1	Input of Elliptic faults	36
A.2	Output of fault statistics	37
A.3	Fault center lines file.	37
A.4	Well data files	37
A.4.1	Input of well paths	37
A.4.2	Input of well observations of faults	38
A.4.3	Input of well observations of faults with depth uncertainty	38
A.4.4	Input of well intersection thresholds	39
A.5	Havana-specific file type: Havana faults	39
A.5.1	Format for the "EllFaults" file	39
A.6	Horizons	40
B	License manager	41
C	Installation script and start-up script	42
	Index	43

1 User Reference

HAVANA is run from the UNIX command line by writing “havana” directly followed by the name of the model file. If no model file name is given, the program will look for a file named “havana.model” in the current directory and use this file, if possible.

The program is fully controlled by commands and arguments/parameters in one or more model files.

1.1 Model file syntax

Comments can be inserted into model files using ‘!’ (for single line comments) or pairs of ‘<’ and ‘>’ (for multi-line comments). All lines containing a ‘!’ are regarded as blank from that point on. Everything starting with a ‘<’ and ending with a ‘>’ is also disregarded.

All non-blank entries must be commands: They must start with a command word and end with a ‘\’ character. Between the command word and the backslash is the argument list, which in general may contain any number of arguments.

The command ACTION divides the model file into sections. Before the first ACTION command comes a section with some commands pertaining to the whole program, see Section 1.2. The allowed commands in each following section depends on the specific ACTION command heading it.

Within each section, the commands may appear in any order, but any command (except READ commands) may appear at most once. READ commands make it possible to read commands from other files, so that one may split the model file into several smaller files (see Section 1.2.4).

The different sections correspond to different modules within HAVANA. Each module is independent of the others in that it reads all of its input from files, and writes all results to files before the next module is started. This is slightly inefficient, but has the advantage that HAVANA may be restarted at any module that causes it to terminate, instead of at the very beginning.

The whole model file is checked before any part of the actual program is run. If any errors are found, they are listed, and the program is terminated. Note that there may still be errors in the input files read by the program: It is not checked that for example ECLIPSE input files or RMS input files are correct before the program execution reaches the place where they are read in.

1.2 Commands common to the whole program

Many HAVANA commands may only appear in specific sections, i.e., below specific ACTION commands. However, some commands may appear before any action commands, or in any section, or indeed anywhere. These are listed below.

1.2.1 Command: INPUT_AND_OUTPUT_DIRECTORY

Optional: May only appear before any of the ACTION commands.

Description: All files read and written by the program will be assumed located relative to this directory.

Arguments: One. The name of a directory, either relative to the one where the program is run, or absolute.

Examples: INPUT_AND_OUTPUT_DIRECTORY ../mydir \

 INPUT_AND_OUTPUT_DIRECTORY /user/geir/havana/example/ \

1.2.2 Command: INPUT_DIRECTORY

Optional: May only appear in the ACTION sections.

Description: This command overrides any INPUT_AND_OUTPUT_COMMAND with relation to INPUT files, and redirects the search for input files in the current ACTION section.

Arguments: One. The name of a directory, either relative to the one where the program is run, or absolute.

Example: INPUT_DIRECTORY newfield/data \

1.2.3 Command: OUTPUT_DIRECTORY

Optional: May only appear in the ACTION sections.

Description: This command overrides any INPUT_AND_OUTPUT_COMMAND with relation to OUTPUT files, and redirects the writing of output files in the current ACTION section.

Arguments: One. The name of a directory, either relative to the one where the program is run, or absolute.

Example: OUTPUT_DIRECTORY results \

1.2.4 Command: READ

Optional: May appear anywhere, any number of times.

Description: This command includes any set of valid HAVANA commands written in a separate file. The effect is as if the commands had appeared directly in the main model file. Any commands may appear in such files (including new READ commands) *except* ACTION commands. This guarantees that one will always know what modules are run just by reading the top model file.

Arguments: The name of one or more files with HAVANA commands.

Examples: READ simulate.model \

 READ sim1.model sim2.model \

1.2.5 Command: SEED

Optional: If neither a seed number or a seed file are available, a random seed number will be used. If both are available, the seed number given in the seed file is used.

Description: Sets the seed for the random number generator.

Arguments: One. An integer between 0 and 4294967295.

Example: SEED 74839254 \

1.2.6 Command: INPUT_SEED_FILE

Optional: If neither a seed number or a seed file are available, a random seed number will be used. If both are available, the seed number given in the seed file is used.

Description: Reads and sets the seed for the random number generator from a file. If the file does not exist, it will be created. The value of the seed after the simulation is written out to the file.

Arguments: One. The name of an ASCII file containing one single integer between 0 and 4294967295.

Example: INPUT_SEED_FILE seed.dat \

1.2.7 Command: LEVEL_OF_INFORMATION

Optional: May appear anywhere. If it appears before any ACTION command, it affects the whole program, while if it appears below an ACTION command, it affects only that section.

Description: The command regulates the amount of information output to screen and to the log file. There are three levels of information: 0, 1, and 2. When LEVEL_OF_INFORMATION does not appear, information level 0 is used. Note that even if the information level is zero, information at level 1 will be output to the log file.

Arguments: One. An integer 0, 1 or 2. If LEVEL_OF_INFORMATION appears without any argument, it corresponds to information level 1.

Example: LEVEL_OF_INFORMATION 2 \

1.2.8 Command: OUTPUT_LOG

Optional: May only appear before any ACTION commands.

Description: Causes information from the program to be saved in a log file. The information is the same as that appearing on the screen, but if the information level is zero, information at level 1 will still be output to the log file.

Arguments: One. The name of the log-file.

Example: OUTPUT_LOG logfile.dat \

1.3 Input of fault model from RMS.

Several commands are needed for input of a fault model from RMS. These commands can only appear below the ACTION keyword, but they are common for all actions.

The directory containing the fault model file, and the fault surfaces is given by the INPUT_FAULTS command, the fault displacement is deduced from the fault lines given by the FAULT_LINE_POLYGON command, additional parameters needed to generate the displacement field are given by the SFM_PARAMETERS, VARIO_TYPE, FAULT_DISPLACEMENT_LENGTH and FAULT_LENGTH_HEIGHT keywords. The last two commands are renamed to SFM_FAULT_DISPLACEMENT_LENGTH and SFM_FAULT_LENGTH_HEIGHT for the SIMULATE action due to the need for similar parameters in the model used for simulating faults.

1.3.1 Command: INPUT_FAULTS

Necessary

Description: Specifies the name of the directory containing the fault files. The directory must contain a faultmodel.txt file together with the fault surfaces as point sets.

To obtain the faultmodel.txt file the **RMS_FAULT_MODEL_FILENAME** environment variable should be set to the desired location of this file. The file is then generated whenever a **Fault modelling** job is run within RMS.

The fault surfaces as point sets are generated by extracting the fault surfaces from the structural model within RMS and export the resulting point sets using the **Roxar text** format.

Arguments: One. The directory.

Example:

```
INPUT_FAULTS faults/sfm_faults \
```

1.3.2 Command: **FAULT_LINE_POLYGON**

Necessary

Description: List of files with fault line polygons exported from RMS. One file corresponds to one horizon, and contains fault lines for more than one fault.

The fault lines are generated in RMS by extracting fault lines from the horizon model within the structural model. The resulting fault lines are placed in the Horizons list and exported using the **Roxar text** format.

Arguments: One or more. Each argument specifies a file containing a set of fault line polygons.

Example:

```
FAULT_LINE_POLYGON
TopC_faultlines.points ;
BaseA_faultlines.points \
```

1.3.3 Command: **SFM_PARAMETERS**

Necessary

Description: Parameters describing the fault and corresponding influence area.

Arguments A list of faults with corresponding parameters. For each fault the fault name, fault throw distribution (=1.0 all throw is distributed downwards, =0.0 all throw is distributed upwards), reverse drag distance (measured laterally) and whether the fault is normal (=1) or reverse (= -1) is given.

If the name 'default' is given the following parameters applies to all faults with no explicitly given parameters.

Example:

```
SFM_PARAMETERS
F3 1.0 500 1 ;
F1 0.7 2000 -1 ;
default 0.5 1000 1 \
```

1.3.4 Command: **VARIO_TYPE**

Necessary

Description: Specifies a variogram model used when modeling the displacement field on the fault surface.

Arguments: Four or five:

1. Variogram type. Possible variogram types are:
 - GAUSSIAN - Gaussian variogram.
 - SPHERICAL - Spherical variogram.
 - EXPONENTIAL - Exponential variogram.
 - GENERAL_EXPONENTIAL - General exponential variogram. An additional parameter giving the power must be given.
2. Range in strike direction. Higher range gives smoother fault surfaces.
3. Range in dip direction. Higher range gives smoother fault surfaces.
4. Anisotropy angle in degrees. Use this if you want to rotate the direction of the ranges.
5. The power. Only for general exponential variograms.

Examples:

```
VARIO_TYPE SPHERICAL 1000 500 0.0 \
```

1.3.5 Command: FAULT_DISPLACEMENT_LENGTH

Optional

Description: The relationship between maximum displacement and length of fault.

Arguments: Two constants, a and b, where $length = (maximumDisplacement/b)^{1/a}$

Default values, if command is not defined, are 1.0 and 0.05.

Examples: `FAULT_DISPLACEMENT_LENGTH 1.0 0.05 \`

1.3.6 Command: FAULT_LENGTH_HEIGHT

Optional

Description: The relationship between length and height of fault.

Arguments: One constant, c, where $height = (length/c)$ Default value, if command is not defined, is 2.0.

Examples: `FAULT_LENGTH_HEIGHT 2.0 \`

1.4 Simulate: Simulate sub-seismic faults

ACTION Simulate \

This module is used when parametric faults are simulated, possibly conditioned on the presence of known faults and well observations of the geological layers. The simulation is done according to the specified intensity maps and distributions for the fault properties.

1.4.1 Command: INPUT_FAULTS

See [1.3.1](#).

1.4.2 Command: FAULT_LINE_POLYGON

See 1.3.2.

1.4.3 Command: SFM_PARAMETERS

See 1.3.3.

1.4.4 Command: VARIO_TYPE

See 1.3.4.

1.4.5 Command: SFM_FAULT_DISPLACEMENT_LENGTH

See 1.3.5.

1.4.6 Command: SFM_FAULT_LENGTH_HEIGHT

See 1.3.6.

1.4.7 Command: OUTPUT_HAVANA_FAULTS

Necessary

Description: Specifies a directory for output of the deterministic and simulated faults. The faults are all written on the HAVANA format used by HAVANA 5. The output directory will only contain the simulated elliptic faults.

Arguments: One. The name of the directory where the faults are to be written.

Example: OUTPUT_HAVANA_FAULTS outhfdir \

1.4.8 Command: FAULTS_STATISTICS

Optional

Description: The most important data for the simulated faults are output to a file on an easy-to-read format.

Arguments: One, two or three. First, the name of the output file for the statistics. The format for this file is presented in Section A.2. The length of the fault is not the major diagonal of the elliptic plane, but rather the part of this diagonal that is not truncated away by other faults. Similarly for the height values.

To obtain better compatibility with the format for inputting Elliptic faults, one may add the word "NoTruncations" as the second argument of this command. Then, the untruncated lengths and heights will be output. The option "TruncInfo" will output the number of faults which truncates the given one, and their fault names.

Fault statistics file written with the "NoTruncations" and "TruncInfo" may be imported in RMS.

Examples:

```
FAULTS_STATISTICS statistics.dat \
```

```
FAULTS_STATISTICS statistics.dat NoTruncations \
```

```
FAULTS_STATISTICS statistics.dat TruncInfo \
```

```
FAULTS_STATISTICS statistics.dat NoTruncations TruncInfo \
```

1.4.9 Command: SIMULATION_VOLUME

Necessary

Description: Defines the boundary of the volume where the faults are simulated, represented by their centerpoints.

Arguments: One of two forms:

- The arguments are the names of the top and bottom horizons of the reservoir for which faults are to be simulated. Laterally, the centerpoints of faults will then only be placed where both horizons exist and have non-missing values. Vertically, centerpoints of faults may be placed between the horizons, but also in a buffer below and above the horizons. The size of this buffer is such that all faults will normally intersect the reservoir. However, the size of the buffer may also be specified by the user, using the keyword VERTICAL_BUFFER_SIZE, see the next section.
- The command has six numbers as arguments. These numbers specify the range of the centerpoints of generated faults in the following manner: minimum and maximum for the *x*-coordinate, and then for the *y*- and *z*-coordinates.

Examples:

```
SIMULATION_VOLUME
  ../irapsurfaces/munin_top.igri
  ../irapsurfaces/munin_bot.igri \
```

```
SIMULATION_VOLUME
  457000 466000 6574000 6587000 2100 3500 \
```

1.4.10 Command: VERTICAL_BUFFER_SIZE

Optional

Description: Defines the size of the vertical buffer above and below the reservoir used when simulating faults; see the SIMULATION_VOLUME command. If the VERTICAL_BUFFER_SIZE command is not used, the program computes a suitable buffer size.

Arguments: One. The size of the buffer.

Examples:

```
VERTICAL_BUFFER_SIZE 0 \
```

```
VERTICAL_BUFFER_SIZE 50 \
```

1.4.11 Command: INPUT_FAULT_CENTER_LINES

Optional

Description: Fault center lines are used to simulate (elliptic) new faults. The fault center lines are read from a ASCII file.

File format for the ASCII fault center lines file:

```
n                ! Number of faults to be simulated from fault center lines
missing value    ! The definition of the missing value
Faultname_1      ! name 1 of n
```

```

normal_1      ! Normal fault 1, reverse fault 0
n_1           ! Number of fault center lines points for this fault
x_11 y_11 z_11 ! Point 1 of n_1 points
dipD_11      ! Dip direction, 1 for east, 0 for west
dipA_11      ! Dip angle, 0 to 90 degrees
throw_11     ! Local throw
x_12 y_12 z_12 ! Point 2 of n_1 points
dipD_12
dipA_12
throw_12
:
Faultname_2  ! name 2 of n
normal_2
n_2
x_21 y_21 z_21 ! Point 1 of n_2 points
dipD_21
dipA_21
throw_21
:

```

Note: The values for normal/reverse fault, dip direction, dip angle and local throw can be given as missing values if they are not known. The values will then be drawn from the stochastic model. The choice of missing value is defined in the fault center lines file.

Arguments: The name of a file containing the fault center lines.

Example:

```

INPUT_FAULT_CENTER_LINES
    faultcenterlines.dat \

```

1.4.12 Command: FAULT_CENTER_LINES_OPTION

Optional

Description: Specifies the method used in INPUT_FAULT_CENTER_LINES for generating faults from fault center lines. The only legal values are 0 and 1. The algorithm used for option 1 is using the two endpoints of the fault center line to define the length, strike and location of the centre. The displacement is found as the maximum observed displacement. The dip is estimated from the observed dips. The algorithm used for option 0 is using all the points for estimating the length, strike and location of the centre. The default is to use algorithm 0.

Example: FAULT_CENTER_LINES_OPTION 1 \

1.4.13 Command: INPUT_WELL_PATHS

Optional:

Description: This command is used to specify the wells. For each well, the name of the well must be given, together with the name of a file specifying the well path. The format for this file is given in Section [A.4.1](#).

Arguments: A list of triples of arguments. Each triple consists of the name of the well, the name of the file containing the well path specification, and the height of the kelly bushing. With three elements per well, the depth measurements in the z-coordinate then refer to depth

below the kelly bushing. If only two arguments are found, the depth measurements are assumed to be below mean sea level (MSL) and not below the kelly bushing (KB).

Examples:

```
INPUT_WELL_PATHS
  A1 wellpathA1.dat 100
  A2 wellpathA2.dat 100 \
```

```
INPUT_WELL_PATHS
  A4 wellpathA1.dat
  A5 wellpathA2.dat \
```

1.4.14 Command: INPUT_WELLOBS_OF_FAULTS

Optional

Description: Specifies a collection of points in the reservoir where a fault has been observed.

The points are read from an ASCII file. For each point, one may also specify an interval for the throw, strike and dip of the fault at the point, and one may specify whether it is normal, or whether it is dipping eastwards. Missing information may be replaced with a '?' in the file. See Section [A.4.2](#) for a precise specification of the format.

Arguments: One. The name of a file containing the well fault observations.

Example: INPUT_WELLOBS_OF_FAULTS wellobs.dat \

1.4.15 Command: INPUT_WELLOBS_OF_NOFAULTS

Optional

Description: Used to put restrictions on faults intersecting the well paths. For given intervals along the well paths, one may specify that no fault intersecting the well in this interval has displacement (at the intersection point) above a certain threshold. The intervals and thresholds are read from a file; see Section [A.4.4](#) for the format of this file.

Arguments: One. The name of a file containing the intervals and thresholds.

Example: INPUT_WELLOBS_OF_NOFAULTS welldata.dat \

1.4.16 Command: NUMBER_OF_FAULTS

Necessary

Description: Determines the number of new faults to be simulated. Note that if deterministic faults, these are additional.

Arguments: One. A positive integer.

Example: NUMBER_OF_FAULTS 100 \

1.4.17 Command: RELATIVE_INTENSITY

Optional

Description: Specifies a trend function for the relative intensity of faults. The term intensity is defined as the expected number of events (i.e. fault center points) per unit area. HAVANA normalizes the values in the intensity field, so that multiplying all the values in the trend maps with a fixed constant C will not change the result.

This command is an alternative to command `DISPLACEMENT_INTENSITY` in Section 1.4.19.

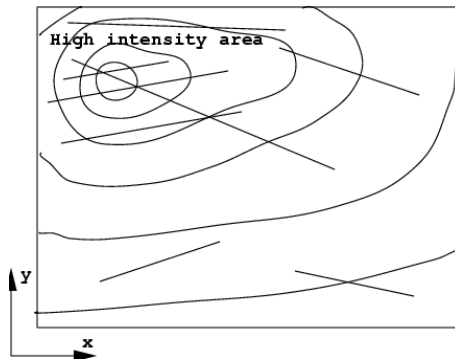


Figure 1.1. Relative intensity and simulated faults.

There are different possible ways of specifying spatially varying trend functions:

1. Using the **Constant** keyword:
The argument is one real number.
This trend function is a constant position independent value.
2. Using the **MultiSurface** keyword:
The arguments are $2N$ file names. The first N files are 2D maps containing a depth surface (TVD). The last N files are 2D maps containing a value of the variable this trend function represents. All files represent grids which must cover exactly the same area and have the same grid resolution. The depth surfaces must be specified in sorted order with the most shallow surface first and the deepest one as the last one. The surfaces should not intersect each other to ensure the same order in all points (x, y) .

The procedure for defining the value of the trend function at position (x, y, z) is as follows:

- Find the grid cell index corresponding to the position (x, y) .
- If the z coordinate is between two depth surfaces, the value will be the linear interpolation of the values of the two grids, interpolating along the vertical line through the point.
- If the z coordinate is above the top or below the bottom depth surfaces, the value in the first or last value file in position (x, y) is assigned.

As one can see from this procedure, a 3D trend is defined from values of the trend function located at N different surfaces in space.

Arguments: One. A trend function. Note that the relative intensity must be non-negative, with some positive values.

Examples:

```
RELATIVE_INTENSITY
  Constant 1.0 \
```

```
RELATIVE_INTENSITY
  MultiSurface
  munin_top.s
```



```

munin_bot.s
munin_top_intensity.s
munin_bot_intensity.s \

```

1.4.18 Command: RELATIVE_INTENSITY_GRID

Optional Only used if command RELATIVE_INTENSITY is specified.

Description: Specifies number of simulation box gridcells nx , ny , nz in x -, y - and z - direction respectively for the relative intensity grid. The default numbers are: $nx = 50$, $ny = 50$ and $nz = 10$.

Arguments: Tree. Integers.

Example: RELATIVE_INTENSITY_GRID 100 100 1 \

1.4.19 Command: DISPLACEMENT_INTENSITY

Optional

Description: Specifies a trend function for the displacement intensity of simulated mother faults. This command is an alternative to the commands RELATIVE_INTENSITY (Section 1.4.17) and REPULSION (Section 1.4.31).

Arguments: One. A **MultiSurface** trend function. For a description of its format, see Section 1.4.17.

Example:

```

DISPLACEMENT_INTENSITY
MultiSurface
munin_top.s
munin_bot.s
munin_displ_intensity.s
munin_displ_intensity.s \

```

1.4.20 Command: DISPLACEMENT_INTENSITY_GRID

Optional Only used if command DISPLACEMENT_INTENSITY is specified.

Description: Specifies number of simulation box gridcells nx , ny , nz in x -, y - and z - direction respectively for the displacement intensity grid. The default numbers are: $nx = 50$, $ny = 50$ and $nz = 1$. Note that the number of gridcells will influence the execution time of the Metropolis algorithm heavily.

Arguments: Tree. Integers.

Example: DISPLACEMENT_INTENSITY_GRID 100 100 1 \

1.4.21 Command: OUTPUT_DISPLACEMENT_INTENSITY

Optional

Description: When the DISPLACEMENT_INTENSITY command is used, one may use the OUTPUT_DISPLACEMENT_INTENSITY command to output the displacement intensity of the realization produced by the simulation. One may also output the target displacement intensity, for comparison. This target intensity is computed by the program from the input in the DISPLACEMENT_INTENSITY command. Both intensities are output on a simple grid format:

```

nx ny nz
for (i=0; i< nx*ny*nz; i++)
    grid[i]

```

This is the grid used internally in the program when it is trying to match the target intensity.

Arguments: One or two. The first argument is the name of the file where the result intensity will be written out. If there is a second argument, it should also be a file name, and the target intensity will be written out there.

Examples:

```

OUTPUT_DISPLACEMENT_INTENSITY simDisplIntensity.dat \
OUTPUT_DISPLACEMENT_INTENSITY simDisplIntensity.dat
                                targetIntensity.dat \

```

1.4.22 Command: DISPLACEMENT_INTENSITY_PARAMETERS

Optional

Description: One may use this command to change from their default settings some of the parameters used in the displacement intensity simulation. Specifically, the first argument is the number of blocks (in the displacement intensity grid) used when smoothing the displacement intensity before matching it with the target density. The default value is 0. The second (and optional) argument is the constant used in the error estimation of the simulation. The default value is 0.00000001. A larger value will give realizations which match the target density less well, but the convergence of the iteration will be faster. A smaller (but positive) argument will make the program try harder to match the exact target density, but the convergence will be slower.

Arguments: One or two. The first is the number of grid cells used in smoothing, while the second is the constant used in the error computations when matching a simulated displacement density with the target density.

Examples:

```

DISPLACEMENT_INTENSITY_PARAMETERS 4 \
DISPLACEMENT_INTENSITY_PARAMETERS 0 10 \

```

1.4.23 Command: DISPLACEMENT

Necessary

Description: Specifies parameters for the distribution of the maximal fault displacements. The displacement of a fault is illustrated in Figure 1.2, and the distribution of these follow the a truncated probability distribution like the one in Figure 1.3.

Arguments: There are two required and one optional sub commands within this command: **Range** and **FractalDimension** (required), and **Asymmetry** (optional).

The arguments following **Range** are two decimal numbers. The first one is minimum displacement and the last one is maximum displacement.

The argument following **FractalDimension** is a decimal number determining the fractal dimension of the distribution.

The arguments following **Asymmetry** are two decimal numbers. The first number specifies how much of the displacement takes place on the footwall side, and how much on the

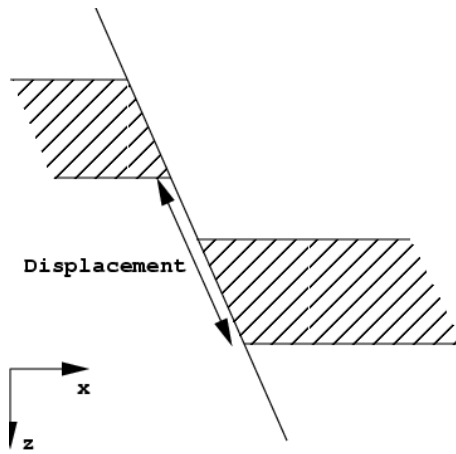


Figure 1.2. Illustration of measuring the displacement of a fault.

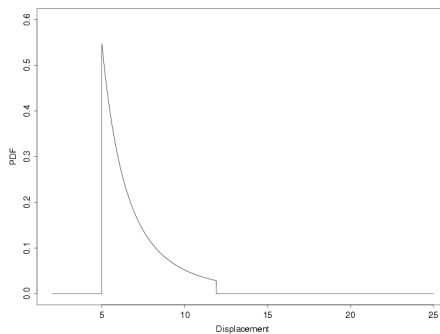


Figure 1.3. A truncated fractal probability density function, with fractal dimension $d = 2.4$ and range from 5 to 12 meters.

hangingwall side. If it is 1, all displacement takes place on the hangingwall side; if it is zero (the default), there is equally much displacement on either side, and if it is -1, all displacement takes place on the footwall side. The second number specifies the uncertainty around the first number. If it is greater than zero, the asymmetry number will be drawn for each fault, from a normal distribution with expectation and standard deviation given by the two numbers. Note that if the **Asymmetry** sub-command does not appear, displacement will always be equally divided between the footwall and hangingwall sides.

Examples:

```
DISPLACEMENT
  Range 7.5 30
  FractalDimension 2.0 \
```

This example generates slumps:

```
DISPLACEMENT
  Range 10.0 30
  FractalDimension 2.0
  Asymmetry 1.0 \
```

1.4.24 Command: FAULT_DISPLACEMENT_LENGTH

Necessary

Description: Specifies parameters for the relationship between (maximum) fault displacement and (maximum) fault length. The fault length l is assumed to approximately be a function of the displacement d . The relationship is as follows: $l \approx (d/c_1)^{1/p}$. The uncertainty in this relationship is modeled by multiplying the right hand side in the equation above by a stochastic variable with lognormal distribution. The fault length is then $l = (d/c_1)^{1/p}V_1$ where V_1 has a lognormal distribution, so that $\log_e(V_1)$ has a normal distribution with expectation zero and standard deviation σ_1 .

Arguments: Three. The exponent p , the constant c_1 and the standard deviation σ_1 , all real numbers.

Values of σ_1 close to 0 (0.1 - 0.2) indicate small uncertainty while larger values (0.5 - 1) indicate more uncertainty in the relationship.

Example: `FAULT_DISPLACEMENT_LENGTH 1.10 0.01 0.05 \`

1.4.25 Command: FAULT_LENGTH_HEIGHT

Necessary

Description: Specifies parameters for the relationship between (maximum) fault length and (maximum) fault height. The fault height h is illustrated in Figure 1.4, and is approximately following the relationship $h \approx l/c_2$ as a "function" of the fault length l . The uncertainty in this relationship is modeled by multiplying the right-hand side in the equation above by a stochastic variable with lognormal distribution. The fault height is then $h = (l/c_2)V_2$ where V_2 has a lognormal distribution, so that $\log_e(V_2)$ has a normal distribution with expectation zero and standard deviation σ_2 .

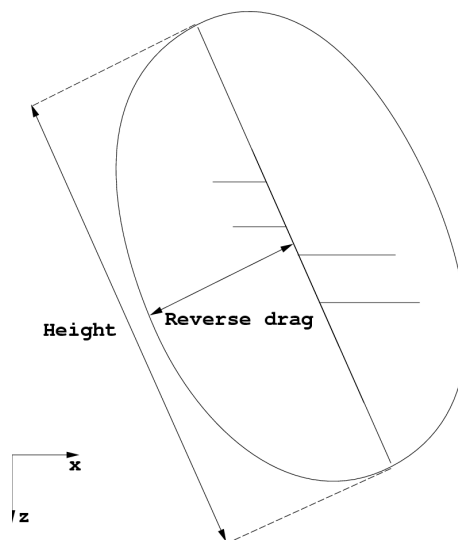


Figure 1.4. Height and reverse drag of a fault.

Arguments: Two. The parameter c_2 and the standard deviation σ_2 .

Values of σ_2 close to 0 (0.1 - 0.2) indicate small uncertainty while larger values (0.5 - 1) indicate more uncertainty in the relationship.

Example: `FAULT_LENGTH_HEIGHT 2.0 0.1 \`

1.4.26 Command: FAULT_AVERAGE_REVERSEDTRAG

Necessary

Description: Specifies parameters for the relationship between the average size of the fault plane and the (maximum) reverse drag of the fault. The reverse drag r (see Figure 1.4) is assumed to approximately follow the relationship $r \approx c_3 \sqrt{lh}$ as a function of the fault height h and the fault length l . The uncertainty in this relationship is modeled by multiplying the right hand side in the equation above by a stochastic variable with lognormal distribution. The reverse drag is then $r = c_3 \sqrt{lh} V_3$ where V_3 has a lognormal distribution, so that $\log_e(V_3)$ has a normal distribution with expectation zero and standard deviation σ_3 .

Arguments: Two. The parameter c_3 and the standard deviation σ_3 .

Values of σ_3 close to 0 (0.1–0.2) indicate small uncertainty while larger values (0.5–1) indicate more uncertainty in the relationship.

Example: `FAULT_AVERAGE_REVERSEDTRAG 0.40 0.1 \`

1.4.27 Command: ORIENTATION_GROUPS

Optional

Description: The strike, dip, and “dip down east” parameters of a fault are collectively described as the “orientation” of the fault in this manual. One may specify several distinct groups of faults and then control the orientation of the faults in each group separately. In each group, the orientation may in fact vary across the reservoir.

To use more than one group of faults in this sense, one must use the command ORIENTATION_GROUPS. It specifies the probability for *mother faults* to belong to the different groups. Orientation parameters for each of the groups must be specified in the STRIKE and DIP commands.

Arguments: Positive decimal numbers specifying the probability of each of the orientation groups. HAVANA normalizes the specified values to probabilities. The number of values will give the number of orientation groups.

Examples: `ORIENTATION_GROUPS 0.3 0.7 \`

1.4.28 Command: STRIKE

Necessary

Description: Specifies the probability distribution with related parameters for the strike. The strike of a fault is the angle between its intersection line with a horizontal plane and north (i.e., the y -coordinate direction), see Figure 1.5. The angle is measured in degrees, between 0 and 180, clockwise.

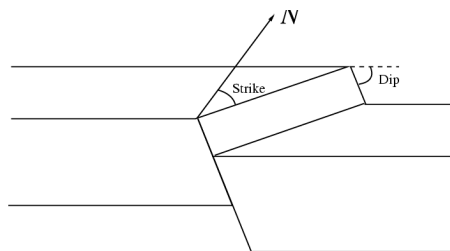


Figure 1.5. Measurement of strike and dip.

This command accepts one or more parameter sets. If the command `ORIENTATION_GROUPS` is used, one set for each orientation group should be specified, otherwise only one set. If `ORIENTATION_GROUPS` is specified and only one parameters set is specified, the same parameters will be used for all groups. The sets of numbers for each group must be separated with a `' ; '` character (Note the blank space before and after). The probability distributions are specified by the keyword **Gaussian**. Additional truncation limits can be specified by the subcommand **Limits**.

Arguments: Each parameter set consists of one or two sub commands as described below.

First the sub command **Limits** with two parameters. The first one is the minimum strike value and the second is the maximum strike value. This sub command is optional.

Then follows the sub command **Gaussian**. A Gaussian distribution with in general spatially varying trend functions giving the expectation and standard deviation. The expectation is specified by sub command **Expectation** followed by a trend function. The standard deviation is specified by the sub command **Stdev** followed by a trend function. See Section 1.4.17 for the available trend functions and their format. The trend functions may be used to get different strike situations in different parts of the reservoir.

Example:

```

STRIKE
  Limits 25 35
  Gaussian
    Expectation
      Constant 30
    Stdev
      Constant 3 ;

  Gaussian
    Expectation
      Constant 150
    Stdev
      Constant 3 ;
\

```

1.4.29 Command: DIP

Necessary

Description: Specifies the distribution of the dip angle for the faults. The dip of a fault is the inclination angle between the fault plane and the horizontal, see Figure 1.5. The dip is given in degrees, between 0 and 90.

This command accepts one or more parameter sets. If command `ORIENTATION_GROUPS` is specified one set for each orientation group can be specified, otherwise only one set. If `ORIENTATION_GROUPS` is specified and only one parameters set is specified, the same parameters will be used for all groups. The sets of numbers for each group must be separated with a `' ; '` character (Note the blank space before and after).

Arguments: Each parameter set consists of four sub commands **ProbDownEast**, **ProbNormal**, **Expectation** and **Stdev**.

The sub command **ProbDownEast** is followed by a decimal number which is the probability

for the fault plane dipping down towards the east. The sub command **ProbNormal** is followed by the probability of having a normal fault (contrary to a reverse fault).

Then, the expectation and standard deviation of the Gaussian distribution of the fault dip angle is specified by the sub commands **Expectation** and **Stdev**, each followed by a trend function, see section 1.4.17. These trend functions may be used to let the dip vary across the reservoir.

Example:

```
DIP
  ProbDownEast 0.7
  ProbNormal 1.0
  Expectation
    Constant 60.0
  Stdev
    Constant 4.0 ;

  ProbDownEast 0.5
  ProbNormal 1.0
  Expectation
    Constant 40.0
  Stdev
    Constant 4.0 ;
\
```

1.4.30 Command: FAULT_TRUNCATION

Necessary

Description: Specifies a parameter controlling when one fault should truncate another. When two fault planes intersect, it is always the fault appearing first in the ordered list of faults that may or may not truncate the other one; the second fault can never truncate the first.

If the given parameter is greater than or equal to zero, and less than or equal to one, then truncation will be decided according to the following rule: The length of the line of intersection between the fault planes is compared with the length of the extension of this line in the latter fault of the list. If the ratio between these lengths is above the given parameter, then truncation occurs; otherwise, it does not.

If the given parameter is less than zero, then truncation is decided stochastically: The relative intersection fraction is computed as above, and is then used as the probability for intersection.

Arguments: One. A decimal parameter, the truncation limit. Values close to 1 indicate very little truncation, while values closer to 0 indicate truncation of all faults intersecting. Negative numbers indicate stochastic truncation.

Example: FAULT_TRUNCATION 0.01 \

1.4.31 Command: REPULSION

Optional

Description: Specifies parameters regarding the spatial interaction (repulsion) between parent faults. This command is an alternative to command DISPLACEMENT_INTENSITY in

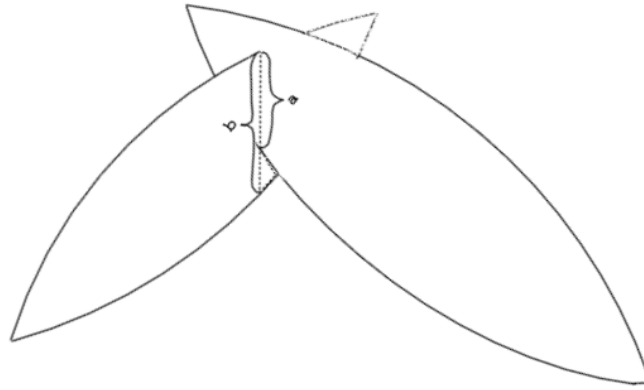


Figure 1.6. Relative intersection determining the truncation of faults.

Section [1.4.19](#).

Arguments: Two. The first argument is the interaction range, a decimal number. If two parent faults are further apart than the interaction range, they will not repel each other. The second argument is the maximum negative interaction potential, indicating the strength of the interaction. If the absolute value of this parameter is large, the repulsion is strong, if it is close to zero, the repulsion is weak.

Example: `REPULSION 2000.00 -1.00 \`

1.4.32 Command: `DISPLACEMENT_WEIGHT`

Optional

Description: The parameter specified here is an exponent for the fault displacements when these are used to determine family members for the mother faults. The probability for the i th mother with displacement d_i to be selected as the mother for a new fault is originally $d_i / \sum d_i$. This implies that very large faults easily will become mothers for too many children faults. To decrease the impact of the displacement size a weight is introduced. The weighting function is d_i^p , where d_i is the displacement and p is the exponent parameter given here. The new probabilities are $d_i^p / \sum d_i^p$. Giving values larger than 1 increase the significance of the displacement in this relationship, while values smaller than 1 gives more

Arguments: One. A decimal number for the exponent p . Default is $p = 1$.

1.4.33 Command: `NUMBER_OF_FAMILIES`

Necessary

Description: Specifies the expected number of families to be used. This number should be the desired expected sum of the seismic faults, the mother faults in earlier simulated fault sets and the new simulated mother faults.

The actual number of families is drawn from a binominal distribution with the given number as the expectation value.

Setting the expected number of families to 1 will usually prevent simulation of new mother faults, given that the number of input mother faults are sufficiently large.

Arguments: One. An integer.

Example: NUMBER_OF_FAMILIES 10 \

1.4.34 Command: CHILDREN_PARAMETERS

Necessary: if children faults are to be simulated and neither CHILDREN_PARAMETERS_GENERAL nor RELAY_RAMPS is given.

Description: Specifies how the center points of the children faults are distributed, relatively to the mother faults. These points are placed using a multinormal distribution around the center point of the mother fault.

Arguments: Three. The standard deviations along the length of the fault, along the height of the fault, and normal to the fault plane, respectively. All these numbers are relative to the dimensions of the current mother fault. For example, using the parameters 0.5 0.5 0.5 means that almost all children faults will be placed inside the mother fault ellipsoid. Increasing the first number means that children may appear further away along the length of the mother fault. Increasing the last number means that children may appear further away from the mother in the direction normal to the mother fault plane.

Example: CHILDREN_PARAMETERS 0.9 0.3 0.7 \

1.4.35 Command: CHILDREN_STRIKE

Necessary: if children faults are to be simulated and neither CHILDREN_PARAMETERS_GENERAL nor RELAY_RAMPS is given.

Description: The fault strike for the children faults is assumed to follow a normal distribution where the strike of the mother fault is the expected value. This command specifies the standard deviation in this distribution, in degrees.

Arguments: One. The standard deviation of the child fault strike, a decimal number.

Example: CHILDREN_STRIKE 10.0 \

1.4.36 Command: CHILDREN_DIP

Necessary: if children faults are to be simulated and neither CHILDREN_PARAMETERS_GENERAL nor RELAY_RAMPS is given.

Description: The fault dip for the child faults is assumed to follow a normal distribution, where the dip of the mother fault is the expected value. This command specifies the standard deviation in the distribution.

Arguments: One. The standard deviation of the child fault dip, a decimal number.

Example: CHILDREN_DIP 2.5 \

1.4.37 Command: CHILDREN_PARAMETERS_GENERAL

Optional

Description: Specifies how children faults are simulated, relatively to the mother faults. This command is an alternative to the CHILDREN_PARAMETERS, CHILDREN_DIP and CHILDREN_STRIKE commands.

Arguments: The command has 14 or more arguments specifying fault names and the 14 parameters controlling the simulation of children faults.

The children faults are distributed around a single point. The first three arguments specify the position of this point relative to the centerpoint of the mother fault. The arguments are

the shift in the strike, dip and reverse drag direction. The values should be relative to of the size of the mother fault (length, height and reverse drag). For example, if 1 0 0 is used, the centerpoints of the children faults will be distributed around one of the fault tips.

The distribution for the children fault centerpoints is a combination of two multinormal distributions, one in each direction along the length of the mother faults. Argument four and five specifies the standard deviations along the length of the mother fault in both directions, the next two arguments are the standard deviation along the height, and normal to the fault plane, respectively. Again, all these numbers are relative to the size of the current mother fault.

For example, using the parameters 0.5 0.5 0.5 0.5 means that almost all children faults will be placed inside an ellipsoid equal in size to the mother fault ellipsoid. Increasing the first number means that children may appear further away along the length of the mother fault. Increasing the last number means that children may appear further away from the mother in the direction normal to the mother fault plane.

The intensity of children fault centerpoints are expressed by $\exp(-x^a)$, where x is the distance from the centerpoint scaled by the standard deviations. The a can be set using argument 8. For a multinormal distribution a value of 2.0 must be used.

Argument 9 controls the children fault intensity on the hangwall v.s footwall side of the mother fault. The values accepted are from -1 to 1. A value of 1 will cause zero intensity at the footwall side and double intensity at the hangwall side. A value of 0.5 will cause at 50% increased/decreased intensity at the footwall/hangwall sides. A value of 0 will cause no shift, i.e., same intensity on both sides.

The fault strike and dip for the children faults is assumed to follow normal distributions. Argument 10 is the expected dip direction, relative to the dip direction of the mother fault. Argument 11 is the dip standard deviation. The values should be given degrees.

Argument 12 and 13 is similar to 10 and 11, but for strike.

It is possible to specify several parameter sets separated by ;. Argument 14 is a number specifying the fraction of children to be simulated using the specified set of parameters. The "numbers" will be scaled if their sum is different from one.

Parameter sets may be given for specific mother faults by entering a list of fault names starting from argument 15. Faults specified will only use parameter sets where it is specified.

Note that at least one parameter set must be *without* faultnames to account for unspecified mother faults. Due to the stochastic nature of the Simulate action there is always a possibility of a new "unknown" mother fault.

Example:

```
CHILDREN_PARAMETERS_GENERAL
  1.0 0.0 0.0      ! center for intensity strike/dip/normal dir.
  0.1 1.0 0.1 0.05 ! length, length2, height, width
  2.0              ! A in exp(1/A)
  0.0              ! HangwallFrac
  0.0 0.5         ! Dip: exp. value rel. to mother and st.dev.
-30.0 1.0        ! Strike: exp. value rel. to mother and st.dev.
  0.6             ! fraction of children
;
```

```

! A parameter set using a compact input style:
! Co  Cd  Cn  L1  L2  H   W   A   HW  Ed  Sd  Es  Ss  F
  1.0  0.0  0.0  0.1  1.0  0.1  0.05 2.0  0.0  0.0  0.5  90.0 1.0  0.4
;
  1.0 0.0 0.0      ! center offset strike/dip/normal
  0.1 1.0 0.1 0.05 ! length, length2, height, width
  2.0              ! childrenA
  0.5              ! childrenHangwallFrac
  0.0 0.5          ! Dip: exp. value rel. to mother and st.dev.
90.0 1.0          ! Strike: exp. value rel. to mother and st.dev.
  1.0              ! fraction of children
HF1 HF2 HF3      ! list of fault names for this param. set
/

```

1.4.38 Command: CHILDREN_PARAMETERS_MOTHER_TYPE

Optional

Description: Specifies how the center points of the children faults are distributed, relatively to the mother faults, if the mother fault is of type RMS. There are two different possibilities. The RMS fault is approximated by an elliptical fault. This method is very fast, but not very accurate. The other possibility is to use the triangle structure of the RMS. This method is accurate, but very slow, depending on the density of the triangularization and the size of the intensity field given in RELATIVE_INTENSITY_GRID or DISPLACEMENT_INTENSITY_GRID.

Arguments: One. The two options are elliptic or rms. The default value is elliptic.

Example:

```

CHILDREN_PARAMETERS_MOTHER_TYPE rms \
CHILDREN_PARAMETERS_MOTHER_TYPE elliptic \

```

1.4.39 Command: RELAY_RAMPS

Optional

Description: Specifies the relay ramp intensity between two mother faults. The intensity field is given as an area between parts of the two mother faults, with a planar top and bottom boundary. The children faults are distributed according to the given parameters.

Arguments: The first two parameters specifies the names of the interacting mother faults. The next two parameters specify the intensity field between two mother faults as a function of the fractional length of each mother fault, that is, the part of the faults to be included in the relay ramp field. Four parameters describe the intensity field between the interacting mothers. The two first numbers represent the size of the intensity along the strike of the two mothers. The intensity field varies according to a linear function between the mother faults, and a breakline, dividing the intensity field in two different parts. The third number describes the relative distance from the first mother fault to the breakline. A number 0.5 means that the breakline divides the intensity field between the two mothers in half. The fourth number the size of the intensity at the break line. Please note that the intensity sizes are relative, that is, the intensity field given by 1.0 1.0 0.5 0.5 and the field 2.0 2.0 0.5 1.0 are both equal. The next number is the likelihood of having a connecting fault between the two mother faults. A number 1.0 means that there will be a connecting fault, and a number 0.0

means that there will be no connecting fault. The two parameters that follow the likelihood of a connecting fault, specifies the minimum and maximum displacement of the connecting fault. If the maximum value of the displacement is set too low, it may be impossible to draw a connecting fault. The next two numbers gives the standard deviation of strike and dip for the drawn children. Note that the expected strike is parallel to the axis of the relay ramp as defined between the two mother faults. The last parameter is an number specifying the fraction of children, for the two mother faults, to be used for the relay ramp simulation. For example, if the relay ramp structure has a fraction of 0.3 and the children parameter general structure has a fraction of 0.7, this means that 70% of the simulated children, belonging to the two mother faults, will be simulated from the children parameter general family of children faults. The remaining 30% is simulated from the relay ramp family of children faults. It is possible to specify several relay ramps, separated with ;.

Example:

```
RELAY_RAMPS
HF1 HF2          ! mother faults
0.5 0.5          ! relative length of intensity field along mother faults
1.0 1.0 0.5 0.5 ! intensity field between mothers
0.5 3 5          ! connecting fault between mothers
10 10           ! strike and dip standard deviations
0.3 ;           ! fraction of children
HF3 HF4          ! mother faults
0.3 0.2         ! relative length of intensity field along mother faults
1.0 0.5 0.25 0.1 ! intensity field between mothers
0.8 2 4         ! connecting fault between mothers
15 10          ! strike and dip standard deviations
0.5 \
```

1.4.40 Command: NUMBER_OF_ITERATIONS

Optional

Description: Specifies the number of iterations to be used in the simulation procedure for the mother faults.

Arguments: One. An integer.

Example: NUMBER_OF_ITERATIONS 50000 \

1.4.41 Command: FAULTNAME_PREFIX

Optional

Description: Specifies a prefix for the fault names for the generated faults. Defaults to 'HF'.

Arguments: One. A string.

Example: FAULTNAME_PREFIX MyFaults \

1.5 ModifySurfaceFault: Change SFM fault properties

ACTION ModifySurfaceFault \

The necessary and optional keywords for the ModifySurfaceFault action are described here.

1.5.1 Command: INPUT_FAULTS

See 1.3.1.

1.5.2 Command: FAULT_LINE_POLYGON

See 1.3.2.

1.5.3 Command: SFM_PARAMETERS

See 1.3.3.

1.5.4 Command: VARIO_TYPE

See 1.3.4.

1.5.5 Command: FAULT_DISPLACEMENT_LENGTH

See 1.3.5.

1.5.6 Command: FAULT_LENGTH_HEIGHT

See 1.3.6.

1.5.7 Command: OUTPUT_FAULTS

Necessary

Description: Specifies the name of the directory where faults are written to.

Arguments: One. The directory.

Example: OUTPUT_FAULTS transfsurfaceFaults \

1.5.8 Command: HORIZON

Optional

Description: Horizons that should be modified according to the changes in fault displacement.

Arguments: At least one. Name of files with horizons. Either xyz format or Storm format.

Example:

```
HORIZON horizons/topC.xyz ;
        horizons/baseA.xyz \
```

1.5.9 Command: TRANSFORM

Optional

Description: Specifies transformations performed on the faults.

Arguments: At least three. Name of subcommand, name of fault, distance for translation.

Possible subcommands are TRANSLATE_X (translation parallel to the global x-axis), TRANSLATE_Y (translation parallel to the global y-axis), TRANSLATE_NORM (translation parallel to the faults normal vector projected to the global xy-plane).

Note: The fault lines and horizons (if specified) on output are not changed according to the

transformations.

Example:

```
TRANSFORM
TRANSLATE_NORM F1 1000 ;
TRANSLATE_X F2 300 \
```

1.5.10 Command: SCALE_DISPLACEMENT

Optional

Description: Specifies the factor for the change of displacement .

Arguments: At least two. Name of fault, multiplier for displacement change. The displacement at every point on the fault surface is multiplied by this factor.

Result: The fault line polygons are moved according to the change in displacement. If the HORIZON command is given together with this command, the points on the horizon are moved according to the changed displacement.

Example:

```
SCALE_DISPLACEMENT
F3 1.2 ;
F1 0.7 \
```

1.5.11 Command: LOCAL_ROTATION

Optional

Description: Specifies rotations performed on the faults by changing azimuth and/or dip angle.

Arguments: At least three. Name of subcommand, name of fault, change of angle given in degrees. Possible subcommands are CHANGE_AZIMUTH and CHANGE_DIP

Note: The fault lines and horizons (if specified) on output are not changed according to the rotations.

Example:

```
LOCAL_ROTATION
CHANGE_AZIMUTH F2 45 ;
CHANGE_DIP F1 90 \
```

1.6 SimulateSurfaceFault: Simulate SFM faults

ACTION SimulateSurfaceFault \

This action is used to generate stochastic realisations of a set of faults on the SFM format.

Here are the available commands:

1.6.1 Command: INPUT_FAULTS

See 1.3.1.

1.6.2 Command: FAULT_LINE_POLYGON

See 1.3.2.

1.6.3 Command: SFM_PARAMETERS

See 1.3.3.

1.6.4 Command: VARIO_TYPE

See 1.3.4.

1.6.5 Command: FAULT_DISPLACEMENT_LENGTH

See 1.3.5.

1.6.6 Command: FAULT_LENGTH_HEIGHT

See 1.3.6.

1.6.7 Command: OUTPUT_FAULTS

Necessary

Description: Directory used for output of simulated faults.

Arguments: One. Output fault directory.

Example: OUTPUT_FAULTS faults/output \

1.6.8 Command: FAULT_VOLUME_WIDTH

Necessary

Description: Distance from input fault surface to the edge of the fault volume. The total width of the fault volume is twice the input value.

Arguments: One. The width

Example: FAULT_VOLUME_WIDTH 100 \

1.6.9 Command: VARIOGRAM_GEOMETRY

Optional

Description: Variogram used when simulating the fault surface.

Default is a gaussian variogram without any anisotropy, with range 1000 in strike direction and range 2000 in dip direction.

Arguments: Four or five.

1. Variogram type. Possible variogram types are:

- GAUSSIAN - Gaussian variogram.
 - SPHERICAL - Spherical variogram.
 - EXPONENTIAL - Exponential variogram.
 - GENERAL_EXPONENTIAL - General exponential variogram. An additional parameter giving the power must be given.
2. Range in strike direction. Higher range gives smoother fault surfaces.
 3. Range in dip direction. Higher range gives smoother fault surfaces.
 4. Anisotropy angle in degrees. Use this if you want to rotate the direction of the ranges.
 5. The power. Only for general exponential variograms.

Example: VARIOGRAM_GEOMETRY GAUSSIAN 1000 2000 0.0 \

1.6.10 Command: DISTRIBUTION

Optional

Description: Distribution for points on fault surface within fault volume. Uniform and triangular distributions ensure that the simulated fault surface is within the given fault volume.

Uniform distribution is currently used as default.

Arguments: : One. Name of distribution. There are three possible distributions:

- UNIFORM - Uniform distribution within given fault volume.
- TRIANGULAR - Triangular distribution within given fault volume with mode equal to base case.
- NORMAL - Normal distributed with mean equal to base case, and standard deviation equal to half the distance to the border of the fault volume.

Example: DISTRIBUTION UNIFORM \

1.6.11 Command: FAULT_PICKS_FILE

Optional

Description: File with well observations of faults. If no file is given, no fault observations are used during simulation.

Arguments: One. Name of file containing fault picks on RMS format.

Example: FAULT_PICK_FILE wellPicks.txt \

Example of fault picks file:

F1	Well-9	169800.000	570800.000	1727.5908
F10	W8	169047.814	570891.481	1789.8117
F9	W1	169958.321	569800.000	1832.5233

1.6.12 Command: FAULT_PICK_UNCERTAINTY

Optional

Description: Common uncertainty for all given fault picks. We only support uncertainty normally to the reference plane for the fault.

The uncertainty of the fault picks is modelled with a normal distribution with standard deviation equal to half the input uncertainty. The uncertainty is however restricted by fault volume boundaries.

Arguments: One. Uncertainty in fault picks.

Example: FAULT_PICK_UNCERTAINTY 20 \

1.7 FaultTip: Fault displacement field modeling

ACTION FaultTip \

The necessary and optional keywords for the FaultTip action are described here.

1.7.1 Command: INPUT_FAULTS

See 1.3.1.

1.7.2 Command: FAULT_LINE_POLYGON

See 1.3.2.

1.7.3 Command: SFM_PARAMETERS

See 1.3.3.

1.7.4 Command: VARIO_TYPE

See 1.3.4.

1.7.5 Command: FAULT_DISPLACEMENT_LENGTH

See 1.3.5.

1.7.6 Command: FAULT_LENGTH_HEIGHT

See 1.3.6.

1.7.7 Command: OUTPUT_FAULTS

Necessary

Description: Specifies the name of the directory where faults are written to.

Arguments: One. The directory.

Example: OUTPUT_FAULTS transfsurfaceFaults \

1.7.8 Command: SEISMIC_RESOLUTION

Necessary

Description: The seismic resolution, which defines standard deviation of uncertainty in observations.

Arguments: One. One constant value.

Example: SEISMIC_RESOLUTION 10.0 \

1.7.9 Command: SIMULATION

Optional

Description: Indicator telling whether we should simulate or predict.

Arguments: One. 1 for simulation, 0 for prediction. Default value is 1.

Example: SIMULATE 0 \

1.7.10 Command: HORIZON

Optional

Description: Horizons that should be modified according to the changes in fault displacement.

Arguments: At least one. Name of files with horizons. Either xyz format or Storm format.

Example:

```
HORIZON horizons/topC.xyz ;  
        horizons/baseA.xyz \
```

A File formats

A.1 Input of Elliptic faults

The format is as follows: The program will discard the first lines, as long as they do not start with a number. Then, it will interpret each remaining line of the file as data for one fault. The numbers read are interpreted as:

- x-coordinate of the center point of the fault.
- y-coordinate of the center point of the fault.
- z-coordinate of the center point of the fault.
- The total maximal displacement.
- The asymmetry of the displacement (A number between -1 and 1: -1 means that all displacement happens on the foot-wall side; 1 means that all displacement happens on the hanging-wall side, and 0 means a symmetric fault).
- The strike, measured clockwise from the north, in degrees.
- The dip, in degrees, such that vertical faults have dip 90.
- The total (untruncated) length of the fault.
- The total (untruncated) height of the fault plane.
- The reverse drag of the fault (distance from center point to where the fault operator dies out, measured in the direction normal to the fault plane).
- Whether the fault dips down on the east side: The input must be either 1 (if the fault dips down on the east side), or 0.
- Whether the fault is normal: The input must be either 1 (if the fault is normal), or 0.

This list of numbers may be followed by a list of integers. These integers will be interpreted as a list of faults truncating this one: The integers refer to the indices of the other faults in the list. As an example, the following will be a legal input file. It contains two faults truncating each other, and a third, untruncated fault:

This is my own list of faults.
Only two faults here, both truncating each other:

x	y	z	Displacement	Asym	Strike	Dip	Length	Height	Rev.drag	D.East	Norm	Trunc
473248.40	6250139.54	9635.31	4.58	0.0	28.97	83.56	264.91	131.95	33.28	1	1	2
470231.52	6253973.31	10550.87	13.30	0.0	139.94	87.84	683.88	339.61	18.69	1	1	1
475477.07	6250081.58	10034.45	1.43	0.0	19.99	85.73	90.53	46.14	30.40	1	1	

Note also that if the goal was to “link” the two first faults above together, so that they represent one curved fault, the truncation above does not give the desired result. The fault planes are linked as expected, but the two fault operators are both truncated away “outside” the truncation line, while they are added together “inside” the truncation line.

If one wants to link several faults together to represent one curved fault, one should use negative line numbers for truncation. This will give the effect that the fault will be truncated by the plane

going through the intersection line between the two faults, and dividing the angle between the two fault planes in two.

In the example below, the three faults are linked together to represent one curved fault:

The three faults are linked together to represent one curved fault.

x	y	z	Displacement	Asym	Strike	Dip	Length	Height	Rev.drag	D.East	Norm	Trunc
120.00	120.00	2.00	30.00	0.0	65.00	78.99	200.0	50.00	50.00	1	1	-2
100.00	100.00	2.00	30.00	0.0	5.00	80.99	200.0	50.00	50.00	1	1	-1 -3
80.00	50.00	2.00	30.00	0.0	87.00	90.00	200.0	50.00	50.00	1	1	-2

A.2 Output of fault statistics

The format for the output of fault statistics using the keyword `FAULTS_STATISTICS` (see Section 1.4.8) is very similar to the format for inputting Elliptic faults (see Section A.1). The only differences are that in the faults statistics file, the *truncated* lengths and heights of the faults will normally be output, and that the indices indicating fault truncations that may appear at the end of each line in the format for inputting Elliptic faults do not appear in the statistics format.

A.3 Fault center lines file

File format for the ASCII fault center lines file:

```
n                ! Number of faults to be simulated from fault center lines
missing          ! The definition of the missing value
Fault name 1 of n
n_1              ! Number of fault center lines for this fault
x_11 y_11 z_11  ! Point 1 of n_1 points
dipD_11         ! Dip direction, 1 for east, 0 for west
dipA_11         ! Dip angle, 0 to 90 degrees
throw_11        ! Local throw
x_12 y_12 z_12  ! Point 2 of n_1 points
dipD_12
dipA_12
throw_12
:
Fault name 2 of n
n_2
x_21 y_21 z_21  ! Point 1 of n_2 points
dipD_21
dipA_21
throw_21
:
```

A.4 Well data files

We have collected below the different file formats currently used in HAVANA for input and output of well data.

A.4.1 Input of well paths

This file describes a single well path. It contains with N points specified with their (x, y, z) coordinates. The header consists of the integer number N . This number is ignored by Havana. Then follows N lines, each specifying the (x,y,z) coordinates of a point on the well path. The z coordinate is positive, indicating depth below the reference height. By default the reference height is the sea level, but if a non-zero height is given for the height of the kelly bushing, then this height is used. The well path starts directly above the first given point (at the reference height), ends at the last given point, and is linear between any pair of consecutive points.

Example:

```
2
460000 6580000 3000
461000 6581000 3200
```

See Section 1.4.13 for usage of such files.

A.4.2 Input of well observations of faults

This is an ASCII file that contains well observations of faults. Each line in the file represents one fault observation. The location of the observation may be given in two ways: Either, one may give the well name and the distance from Kelly Bushing, (i.e., the measured depth), or one may give the x, y, and z coordinates of the location. The program determines which option is used by determining whether the first item on the line is a text string or a number.

Following the specification of the location, there may be any number of items; as many as seven are read by the program. Each of these seven items must be either a number, or the character '?', which, of course, signifies missing data. If there are less than seven items on the line, the effect is the same as if the missing items had been '?'.

The seven items have the following meaning:

- Minimum fault throw at the observation point.
- Maximum fault throw at the observation point.
- Minimum dip azimuth of the fault (in degrees).
- Maximum dip azimuth of the fault (in degrees).
- Minimum dip of the fault (in degrees).
- Maximum dip of the fault (in degrees).
- Whether the fault is normal (signified by 1) or reverse (signified by 0).

To define the dip azimuth of a fault, take that normal vector to the fault plane that points upwards and project it to the horizontal plane. Then measure its angle (in degrees) with the vector pointing north (in the y coordinate direction), measuring the angle clockwise from north. This produces an angle between 0 and 360 degrees.

Examples:

```
460000 6580000 3000      8      8.2    130    132
461000 6581000 3200      ?      ?      124    125    88     ?     1
```

or, using measured depth to specify position:

```
A1      3945              8      8.2    130    132
A1      4164              ?      ?      124    125    88     ?     1
```

Note how the program reads the data line by line. Thus when there is no information for the last items on a line, it is not necessary to fill out the end of the line with question marks. See Section 1.4.14 for usage of these files.

A.4.3 Input of well observations of faults with depth uncertainty

This is an ASCII file that contains well observations of faults, just as in A.4.2 but with uncertainty in depth. The location of the observation must be given as the well name and the minimum and maximum distance from Kelly Bushing, (i.e., the measured depth).

Example:

A1	3945	3957	8	8.2	130	132				
A1	4164	4187	?	?	124	125	88	?	1	

A.4.4 Input of well intersection thresholds

This file is used to specify intervals along well paths where there are no faults, or at least no faults with displacement above a certain threshold. Each line in the file corresponds to one such interval. The interval is specified by writing first the well name, and then the distances from Kelly Bushing (i.e. the measured depths) of the start and the end of the interval. The line is ended with a single number: The maximal displacement any fault intersecting the line can have (at the intersection point).

Example:

A1	0	3500	5
A2	0	2000	10
A2	2000	3500	5

A.5 Havana-specific file type: Havana faults

The HAVANA faults directory is used for generating input to HAVANA 5.

HAVANA faults are stored in a directory containing the four files "version", "Condition", "EllFaults", and "EllFaultGhosts", and the four directories "IRAPfaults", "IRAPghostFaults", "ParametricFaults", and "ParametricGhostFaults". Elliptic, HAVANA-generated faults are stored in the "EllFaults" file,

The truncation rules are given in the file .truncTable. First is the number of faults to be truncated. Then, on each line, is the name of the fault to be truncated, followed by the number of faults and the fault names.

The other directories and files are always empty when generated with HAVANA 6.

Note that the names of the files and directories cannot be changed. When reading faults from a directory in the Havana faults format, the program will look for files and directories with the names described above, and ignore all other files and directories.

A.5.1 Format for the "EllFaults" file

The first number in the file is the number of faults the file contains. Then follows for each fault:

- The name of the fault.
- The position of the fault, in UTM/TVD coordinates.
- The length of the length, width and reverse drag axes of the ellipsoide. (Note that these numbers are half of the corresponding diameters of the ellipsoid; thus the length of the fault is twice the given number).
- The total maximal displacement of the fault.
- The asymmetry number (between -1 and 1) indicating how much of the displacement takes place on the hanging-wall side of the fault.
- A unit vector normal to the fault plane.
- A unit vector along the length of the fault. this vector will always have a zero z -component.
- A unit vector along the height of the fault. The displacement takes place along this vector.

- The family number of the fault.
- The number of planes truncating the fault. Then, for each such truncating plane, the indices of the faults truncating it.

A.6 Horizons

Horizons may be read and written on several different formats:

- STORM STORMGRID_BINARY format.
- RMS "CLASSIC" ASCII format.

The program will automatically recognize these formats.

B License manager

Starting with version 5.1 Havana has a new license manager controlling the permitted users, the expiration date, and the available modules. The license manager is part of the havana program itself and hence it does not need any daemon running in the background. All you need is a license file.

Hence to run HAVANA you need

1. A license file, obtainable from `support-havana@nr.no`.
2. A model file
3. Either give the command

```
unix> havana -l /full_path/license.file model.file
```

or if the Havana installation script in Appendix C has been used to create a start-up script, or if the environment variable `HAVANA_LICENSE_FILE` is set to the full path of the license file, just type `unix> havana model.file`

where `unix>` is the unix shell prompt. If required, contact the local system manager to get instructions for setting the environment variable.

C Installation script and start-up script

Starting with version 5.2 Havana is delivered with a perl installation script called `install_havana`. In order to run this script, make sure that your perl installation is v5.8.0 or newer.

The installation script places the havana binaries and the license file in directories chosen by the user, creates a start-up script, and places a soft link in, e.g., `\usr\bin\havana`. The start-up script automatically keeps track of the license file location and selects the correct binary for the platform used.

With the installation script correctly set up users do not need to set the variable `HAVANA_LICENSE_FILE` any more.

Index

- ACTION, 7
- Action
 - FaultTip, 34
 - ModifySurfaceFault, 29
 - Simulate, 11
 - SimulateSurfaceFault, 31
- CHILDREN_DIP, 25
- CHILDREN_PARAMETERS, 25
- CHILDREN_PARAMETERS_GENERAL, 25
- CHILDREN_PARAMETERS_MOTHER_TYPE, 27
- CHILDREN_STRIKE, 25
- comment, 7
- DIP, 22
- DISPLACEMENT, 18
- DISPLACEMENT_INTENSITY, 17
- DISPLACEMENT_WEIGHT, 24
- DISPLACEMENT_INTENSITY_PARAMETERS, 18
- DISPLACEMENT_INTENSITY_GRID, 17
- DISTRIBUTION, 32
- environment variable, 41, 42
- FAULT_AVERAGE_REVERSED DRAG, 21
- FAULT_CENTER_LINES_OPTION, 14
- FAULT_DISPLACEMENT_LENGTH, 11, 20
- FAULT_LENGTH_HEIGHT, 11, 20
- FAULT_LINE_POLYGON, 10
- FAULT_PICK_UNCERTAINTY, 32
- FAULT_PICKS_FILE, 32
- FAULT_TRUNCATION, 23
- FAULT_VOLUME_WIDTH, 31
- FAULTNAME_PREFIX, 28
- FAULTS_STATISTICS, 12
- HORIZON, 29, 34
- INPUT_AND_OUTPUT_DIRECTORY, 7
- INPUT_DIRECTORY, 8
- INPUT_FAULT_CENTER_LINES, 13
- INPUT_FAULTS, 9
- INPUT_SEED_FILE, 9
- INPUT_WELL_PATHS, 14
- INPUT_WELLOBS_OF_FAULTS, 15
- INPUT_WELLOBS_OF_NOFAULTS, 15
- installation, 42
- KB, 15
- kelly bushing, 14, 37
- LEVEL_OF_INFORMATION, 9
- license file, 42
- license manager, 41
- LOCAL_ROTATION, 30
- mean sea level, 15
- MSL, 15
- multi-line comments, 7
- NUMBER_OF_FAMILIES, 24
- NUMBER_OF_FAULTS, 15
- NUMBER_OF_ITERATIONS, 28
- ORIENTATION_GROUPS, 21
- OUTPUT_DIRECTORY, 8
- OUTPUT_FAULTS, 29, 31, 34
- OUTPUT_HAVANA_FAULTS, 12
- OUTPUT_LOG, 9
- OUTPUT_DISPLACEMENT_INTENSITY, 17
- READ, 8
- RELATIVE_INTENSITY, 15
- RELATIVE_INTENSITY_GRID, 17
- RELAY_RAMPS, 27
- REPULSION, 23
- reverse drag, 20
- SCALE_DISPLACEMENT, 30
- SEED, 8
- SEISMIC_RESOLUTION, 34
- SFM_PARAMETERS, 10
- SIMULATION, 34
- SIMULATION_VOLUME, 13
- start-up script, 42
- STRIKE, 21

syntax, 7

TRANSFORM, 29

VARIO_TYPE, 10

VARIOGRAM_GEOMETRY, 31

VERTICAL_BUFFER_SIZE, 13