HAVANA user manual

Version 7.0.0

Norsk Regnesentral
NORWEGIAN COMPUTING CENTER

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, Anne Randi Syversveen, Maria Vigsnes, Håvard Goodwin, Vera Louise Hauge, Eyvind Aker, Vegard Berg Kvernelv and Hanna Zdanowicz.

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 information and communication technology and applied statistical-mathematical modelling. The clients include a broad range of industrial, commercial and public service organisations 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 in us is given by the fact that most of our new contracts are signed with previous customers.

Title	HAVANA user manual
Authors	**Frode Georgsen , Per Røe , Anne Randi Syversveen , Maria Vigsnes , Håvard Goodwin , Eyvind Aker , Hanna Zdanowicz , Vegard Berg Kvernelv**
Date	February 14, 2018
Publication number	

Abstract

HAVANA is a program for simulating 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.

Contents

1   Introduction

1.1 Background

Below are some of the major milestones in the history of HAVANA. The years are listed when development of new functionality was started, however the development spanned in most cases several years.

1991 NR Project initiated to model sub-seismic scale faults. Sponsored by BP, Hydro and Statoil.

1993 First HAVANA version delivered.

1993-1995 Improved fault modelling. Sponsored by Hydro and Statoil.

1996-1998 HAVANA version 2 and 3. Sponsored by Conoco, Hydro, Saga and Statoil.

1998 Parametric (pillar-based) fault model for simulation of seismic scale faults.

1999 Calculation of fault seal properties.

2000 Havana version 4.0.

2000 Structural uncertainty modelling.

2002 Havana version 5.0.

2005-2007 Fault Facies. Modelling of fault zone facies. In cooperation with Centre for Integrated Petroleum Research at the University of Bergen. Sponsored by the The Research Council of Norway, Statoil and ConocoPhillips.

2008 New surface-based fault model inspired by new structural model in RMS 9.0. Sponsored by Statoil.

2010 HAVANA version 6.0.

2011 HAVANA version 6.1, 6.2 and 6.3.

2011 Started integration of HAVANA in RMS.

2012 HAVANA version 6.4 with a new implementation of simulation of sub-seismic faults.

1.2 Basic ideas

HAVANA is one program that consists of several separate actions. These actions are run as separate modules, but some actions depend on other actions have been run before either to ensure that the necessary input is generated or to give a meaningful result.

2 Getting started

HAVANA uses a text based model file in XML format as input to define all operations to be done. In this model file all input, output and model parameters are specified. First of all the model file must have the correct structure, see 2.1. HAVANA has a close link to RMS version 2012 or newer. An initial structural framework must be created in RMS and exported to HAVANA format using a job in RMS. HAVANA imports that structural framework as the most important input for further modelling with HAVANA. A suitable directory structure for organizing all HAVANA-input and -output must be defined.

When a modelfile is available, HAVANA is started from the command line with the HAVANA model file as an argument:

```
> havana.exe havana.xml
```

If no modelfile name is given, the program will look for a file named "havana.xml" in the current directory and use this file, if possible.

HAVANA can also be started with the options to create an example modelfile, that can be edited and modified to the actual case. This is described in section 2.4.

2.1 XML-modelfile

The HAVANA modelfile is based on the XML (eXtended Markup Language) format. This gives a hierarchical structure with the lowest level is a single element (keyword) of the form

```
<name>
 argument
</name>
```

A higher level keyword can consist of one or several single elements, or one or several keywords on a lower level like

```
<command>
   <name> argument </name>
</command>
```

The highest level is the keyword `<havana>` which must start the model file and the linked end tag `</havana>` which ends the model file. An example of a simple model file is:

```
<havana>
  <project-settings>
    <log-settings>
      <screen-level> 5 </screen-level>
      <log-file>
        <file> logfile.dat </file>
        <level>    3 </level>
      </log-file>
    </log-settings>
    <io-settings>
      <input-directory> ./Input </input-directory>
```

```
        <output-directory> ./Output </output-directory>
     </io-settings>
  </project-settings>


  <import-rms-fault-data>
     <input-structural-model-directory> StructuralModel </input-structural-model-directory>
     <output-havana-faults-directory> OrigFaultSet </output-havana-faults-directory>
     <debug-output-directory> Debug </debug-output-directory>
     <default-displacement-settings>
        <asymmetry>  0.5     </asymmetry>
        <range>        1000    </range>
     </default-displacement-settings>
     <displacement-variogram>
        <type>   Spherical </type>
        <strike-range> 2000 </strike-range>
        <dip-range>     1000 </dip-range>
     </displacement-variogram>
  </import-rms-fault-data>
</havana>
```

The keyword `<project-settings>` defines general settings that apply to all parts of HAVANA
including all the possible actions. `<import-rms-fault-data>` is an example of one of several ac-
tions that are the main parts of the program. See Chapter 4 for an overview of all actions. A model
file can contain one action, as in the case above or several actions. In case of the latter, the actions
are run one at the time according to the following order regardless of the order they are specified
in the model file:

1. ImportRMSFaultData

2. FaultUncertaintyModel

3. ModifyFaultSurface

4. SimulateFaultSurface

5. ModifyDisplacement

6. SimulateDisplacement

7. RemoveFaults

8. AddFaults

9. SetDisplacement

10. SimulateSecondaryFaults

11. SimulateFaults

12. GroupFaults

13. UpdatePoints

14. UpdateGrid

15. UpdatePermeability

16. CreateFaultGrid

17. MergeFaultGrid

18. CalculateFaultSeal

The keywords on a lower level than the actions, are called commands, like the keyword `input-structural-model-directory` above. See Chapter 6 for a detailed detailed description of all actions and commands.

Comments can be inserted into model files using pairs of '$<!--$' and '$-->$' for single and multi line comments.

2.2 RMS structural model

The basic input for HAVANA is a structural model exported from RMS. The description of this model is given in several files collected in a directory. The files define the volume of interest (bounding box), the name and description of the faults and the horizons, the truncation rules between the faults, the intersection lines between horizons and faults (fault lines) and the definition of the fault blocks. In the model file example above, the action `<import-rms-fault-data>` contains a command `<input-structural-model-directory>` which states the name of the directory defining the structural model that HAVANA will use. A comprehensive description of the file formats in the directory with the structural model is given in appendix A.1.1.

2.3 Directory structure

The start-up directory for HAVANA is the directory where the program is run from. All references to directories in the model file should be relative to this directory. In `<project-settings>` the keyword `<io-settings>` defines an input- and output-directory that applies to all actions. In later actions the specification of input- and output-files or -directories are relative to these settings. Sometimes the files that are output from one action is input to an other action. In case of that, the directory specified in the latter action must be relative to the input-directory specified in `<project-settings>`. The following example illustrates this where a new action `<modify-fault-surface>` builds on the action `<import-rms-fault-data>` in the example above:

```
<modify-fault-surface>
  <input-havana-faults-directory> ../Output/OrigFaultSet </input-havana-faults-directory>
  <output-havana-faults-directory> ModifiedFaultSet </output-havana-faults-directory>
  <output-rms-faults-points-directory> FaultSurface </output-rms-faults-points-directory>

  <modification>
    <fault-name> F6 </fault-name>
    <operation> TRANSLATE_X </operation>
    <value>      100 </value>
  </modification>

</modify-fault-surface>
```

2.4 Modelfile examples

For each of the actions in HAVANA an example modelfile template can be generated.

First an enviroment variable `HAVANA_EXAMPLE_FILES` must be set to the example-files directory.

To generate an example file for an action HAVANA must be executed with the action name as argument. A second argument (optional) specifies the location and name of the output file. If the second argument is omitted, the file is written as the action name in the current folder. It is not possible to overwrite existing files. If the file specified in the second argument exists, the program will stop with a message that the file already exists.

All actions must have defined project settings before it can be used in HAVANA, see 2.1. To generate an example of this, run HAVANA with "project-settings" as an argument. This file shows where havana actions must be inserted.

It is also possible to use "havana-manual" as argument. In that case, the user manual is written to the chosen location.

Example:

```
> havana.exe import-rms-fault-data my-examples/importrmsfaultdata.xml
```

Valid arguments are:

project-settings
import-rms-fault-data
fault-uncertainty-model
modify-fault-surface
simulate-fault-surface
modify-displacement
simulate-displacement
remove-faults
add-faults
set-displacement
simulate-secondary-faults
group-faults
update-points
update-grid
update-permeability
create-fault-grid
merge-fault-grid
calculate-fault-seal
havana-manual

3 Fault model

3.1 Introduction

The main fault model in HAVANA is the Surface Fault Model. This fault model is very flexible and well suited for modelling seismic scale faults. The faults are represented along the same lines as in Hoffman and Neave [6]. When simulating faults below seismic resolution (sub-seismic faults) an elliptic fault model is used, but within the more general framework of the Surface fault Model. The model described in this section is more thoroughly presented in Georgsen et al. [3].

3.1.1 Fault terminology

This section defines some basic fault terminology, that explains how geological concepts are interpreted in the development of HAVANA.
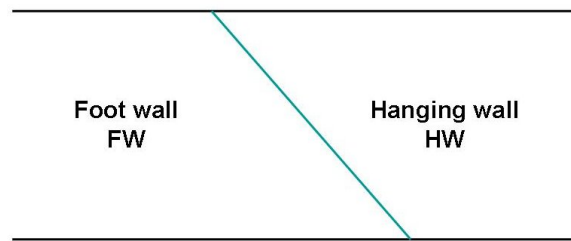
Figure 3.1. The hanging wall and footwall sides of a fault.
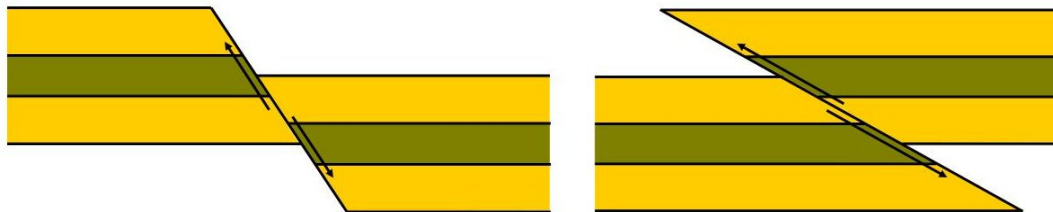
Figure 3.2. A normal fault to the left and a reverse fault to the right.

The hanging wall (HW) occurs above the fault while the footwall (FW) occurs below the fault, see Figure 3.1. A normal fault is formed when the earth crust is extended and when the hanging wall moves down relative to the footwall. Normal faults are usually steep, with a typical dip of 60°. A reverse (thrust) fault is formed during shortening of the crust as the hanging wall is pushed up relative to the footwall. Reverse faults usually have low dips, usually around 30°. See Figure 3.2 for examples of normal and reverse faults.

3.1.2 Strike and dip

For all faults, regardless of fault model used, we have defined a reference point (X_0, Y_0, Z_0), given in UTM-coordinates, and strike and dip angles. The strike angle θ is the clockwise rotation angle of strike line from north and is defined for values in $[0, 2\pi]$. The dip angle ϕ from the horizontal plane is defined in $[0, \frac{\pi}{2}]$. Note that strike angle in HAVANA is defined so that the hanging wall is

on the right hand side when looking in strike direction, and hence the dip azimuth is $strike + \frac{\pi}{2}$. This definition may not always correspond to strike in other settings (e.g. the old versions of HAVANA), where strike is defined in $[0, \pi]$ and additional information of dip direction is needed.

The dip and strike vectors define a local coordinate system. In this local coordinate system, the x-axis follows along the reference plane with positive direction such that the hanging wall side of the fault is to the right, the local y-axis goes down along the fault in dip direction. The local z-axis is defined as the upward-pointing normal to the plane, pointing into the hanging wall. This local coordinate system is illustrated in Figure 3.3.
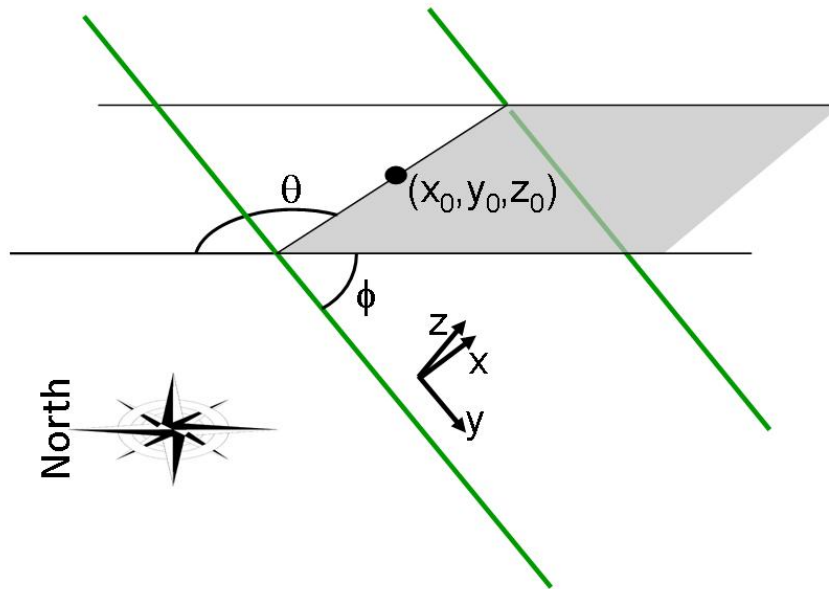
Figure 3.3. Local coordinate system (x, y, z) defined by reference point (X_0, Y_0, Z_0), fault strike angle θ and dip angle ϕ. The z-axis points in hanging wall direction. The edges of the reference plane is drawn by green lines.

The description above gives the following transformation from UTM-coordinates (X, Y, Z) to the local coordinate system

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \sin\theta & \cos\theta & 0 \\ \cos\theta\cos\phi & -\sin\theta\cos\phi & \sin\phi \\ \cos\theta\sin\phi & \sin\theta\sin\phi & -\cos\phi \end{bmatrix} \begin{bmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{bmatrix}. \tag{3.1}$$

3.1.3 3D displacement field

A key feature of the fault model in HAVANA is that a 3D displacement field is assigned to all faults. This 3D displacement field is an approximation of the displacement of the volume surrounding the fault, and can be used both to fault or restore point sets or horizons.

The fault incidence is a complicated event carried out through several steps where every mathematical formulation is a simplification. The displacement of rock caused by the faulting can be looked upon as a continuous process in the sense that two neighbouring points before the fault incidence are also neighbours afterwards due to the nature of the hard material involved. This suggests that the faulting operation can be described by a vector field on the whole volume affected by the fault. Every single point in the volume is moved along a 3D vector from its original position to the faulted position, and the whole vector field can be constructed by the assumption of continuity. In the development of Havana so far, only dip-slip faults are considered. This means

that the displacement on the fault surface can be modelled by one 2D displacement field and the displacement off the fault surface as a function of this. Section 3.2.2 below describes how the 2D displacement field can be constructed from intersection points between horizons and faults and user-defined parameters, so-called fault lines.

3.2 Surface Fault Model

The surface-based fault model is characterised by the ability to model almost any surface. The only limitation is that there exists a reference plane that the fault surface can be mapped onto.

3.2.1 Surface fault parameterisation

In the Surface Fault Model format a reference plane is defined by the given strike and dip and reference point, see Section 3.1.2. The fault surface is defined on this reference plane and all fault operations are performed in the corresponding local coordinate system.

The fault surface is defined as a function $z = f(x, y)$ for all (x, y) on the fault. The value of f is the deviation from the reference plane along the local z-axis (i.e. normal to the reference plane). Figure 3.4 shows a vertical cross section at $x = x'$ where the fault surface is represented by $f(x, y)$ as the deviation from the reference plane. Internally $f(x, y)$ is represented as a set of z-values corresponding to the nodes of a 2D grid on the reference plane.
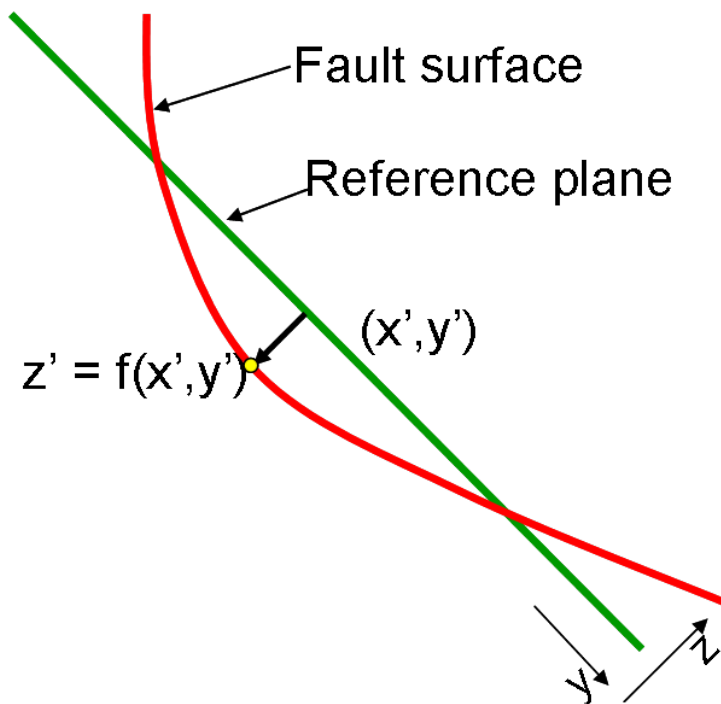
Figure 3.4. Vertical cross section through the fault at $x = x'$. The fault surface is represented by the function $z = f(x, y)$ as the normal deviation from the reference plane.

3.2.2 Constructing 2D displacement fields from fault lines

By identifying fault lines, defined as intersection lines between horizons and the fault surface, information about fault displacement along the fault surface can be found. A description on how a set $\mathbf{d} = \{d(x_1, y_1), d(x_2, y_2), \ldots, d(x_n, y_n)\}$ of displacement values based on fault line information from RMS is given in Havana Technical Documentation [8]. Figure 3.5 illustrates two displacement values d_1 and d_2 associated with two horizons H_1 and H_2 intersecting the fault.
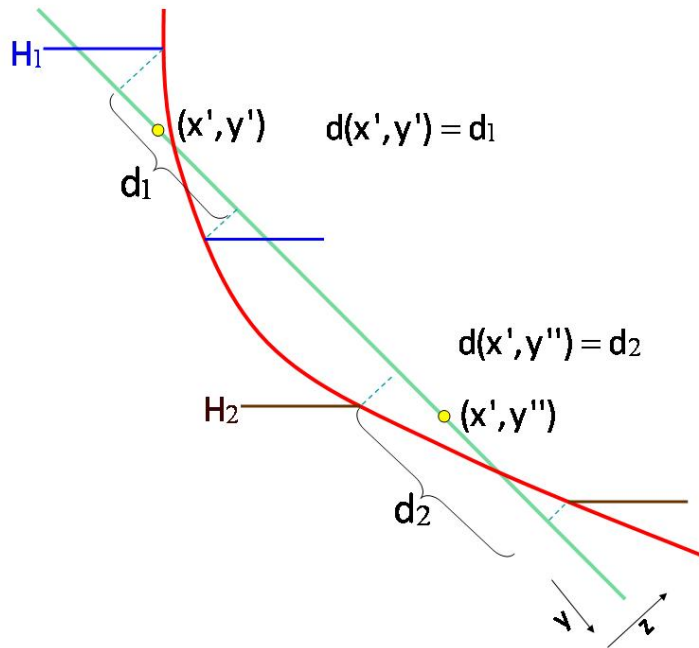
Figure 3.5. Vertical cross section showing how the 2D displacement field $d(x, y)$ defined on the fault surface is calculated from fault line observations.

The 2D displacement field on the fault reference plane is denoted $d(x, y)$ and a regular 2D-grid is defined in the xy-plane. For each fault a hanging wall(HW) / foot wall(FW) displacement ratio can be found, either based on theoretical considerations or estimated from data. According to Gibson et al. [4], this ratio can be predicted from the fault dip. This displacement ratio, given as a user-specified parameter $\gamma \in [0, 1]$ is used to populate the 2D displacement grid. $|\gamma| = 1$ and $\gamma = 0$ means that all displacement is on the hanging wall side and foot wall side respectively. The displacement observed from fault lines and the value for the HW/FW displacement ratio gives an estimate for the position where the original (unfaulted) horizon intersects the fault surface and the associated displacement value.

The HW/FW displacement ratio for the fault in Figure 3.5 is 0.7, meaning that most of the displacement is on the hanging wall side. This defines the positions $f(x', y')$ and $f(x', y'')$ where the unfaulted version of horizons H_1 and H_2 cross the fault surface. The values for the displacement function d are identified as $d(x', y') = d_1$ and $d(x', y'') = d_2$.

The displacement values $\mathbf{d}$ calculated from fault lines can be used in simple kriging (see Goovaerts [5]) to populate the whole 2D field giving a displacement value for all grid locations on the fault surface. When this simple kriging is performed, the fault tip imported from the RMS structural model is also used for conditioning since the displacement is known to be zero at the fault tip.

Figure 3.6 shows how HAVANA modifies the generated displacement field when the throw is changed. The change is highest close to the fault surface and decreases away from the fault along the influence range.

3.2.3 Trend estimation

To perform the interpolation by simple kriging mentioned above, a trend for the displacement field must be estimated. The trend function is given by an ellipse with centre point (x_0, y_0), length (l_x) in strike direction and height (l_y) in dip direction. Argumentation for the elliptic shape of the fault displacement is found in Barnett et al. [1] and Walsh and Watterson [11]. The ideal,
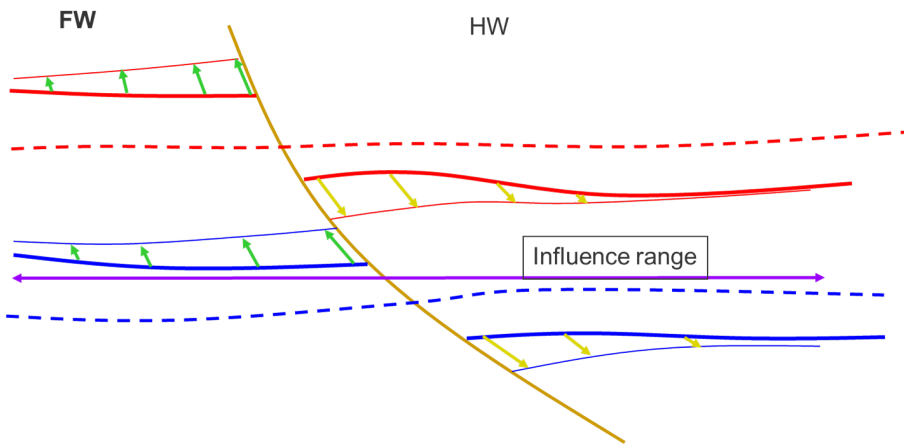
Figure 3.6. Vertical cross section showing how the 2D displacement field $d(x, y)$ is modified when the throw of the fault is changed.

theoretical case has maximum displacement at the centre point and zero displacement reached at the elliptic tip-line. The normalized displacement profile used in the present model is found in Walsh and Watterson [10] and is given by

$$\mu_0(x, y) = 2 \cdot (1 - r(x, y)) \cdot \sqrt{\frac{(1 + r(x, y))^2}{4} - r(x, y)^2}, \tag{3.2}$$

where $r(x, y) = \sqrt{\left(\frac{x - x_0}{l_x}\right)^2 + \left(\frac{y - y_0}{l_y}\right)^2}$ is the normalized radial distance from the centre. The tip-line is the elliptic line defined by $r(x, y) = 1$. Outside this line the trend equals zero. A thorough mathematical description of the estimation method is given in the Havana Technical Documentation ([8]).

3.2.4 Example of displacement field estimation

For illustration, the Emerald field reservoir model (RMS tutorial example) is used. Figure 3.7 shows the structural model for the Emerald field indicating one large truncating fault (F1), and one smaller fault (F2) which is truncated by F1. There are four horizons in the model, and the fault lines for these horizons are drawn on the fault surfaces. The upper four lines represent the intersection between the horizons and the fault on the footwall side and the lower four lines represent the intersection between the horizons and the fault on the hanging wall side. The differences between lines of identical colours represent the displacements and the large displacement on F1 is clearly visible.

Figure 3.8 shows the F1 fault with displacement values found from fault lines as described above with HW/FW displacement ratio 0.7. The dots represents grid nodes in the local 2D grid, with coloured nodes representing known displacement values and unknown values at noncoloured nodes. The maximum displacement value, located in the pink area, is 590 meters. A trend field for the displacement is estimated using the algorithm described above. The trend is shown in Figure 3.9. The elliptical shape of the trend is clearly visible. It is also obvious that the estimated length and height of this ellipse by far extend the actually modelled fault. This reflects the large displacement in the limited area where there are actually identified fault lines. A priori the length of the ellipse is assumed to be 20 times the maximum displacement, and the height is assumed to be half the length. The centre of the trend ellipse is estimated to be further north-east than the location of the maximum value found from faultlines. The maximum displacement in the trend ellipse is estimated to 545 meters.

The result of the simple kriging with the elliptical trend and the known displacement values is
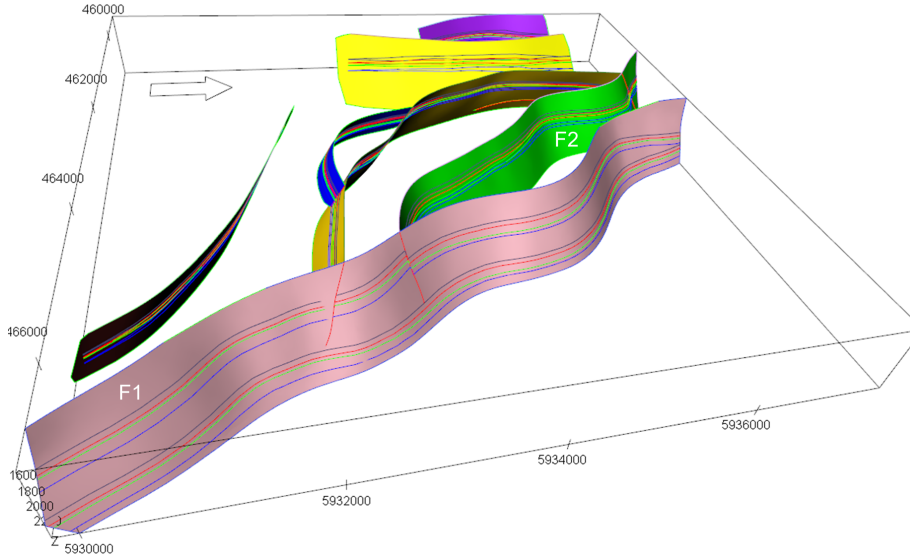
Figure 3.7. The Emerald field structural model with fault lines for four horizons drawn on each fault surface.

shown in Figure 3.10. The correlation structure follows a spherical variogram with a range of 2000 meter in strike direction (x) and 1000 meter in dip direction (y). This displacement field corresponds exactly to the known values at the location where these are found. Since the ellipse centre for the trend does not coincide with the maximum known value, the high displacement values are shifted to south-west along the fault.

3.2.5 3D displacement vector field

When a 2D displacement field on the fault surface is constructed as described above, a full 3D displacement field can be constructed in the volume around the fault surface. The displacement decreases away from the fault surface until it dies out at a distance R from the fault surface along the z-axis (Barnett et al. [1]). The distance is referred to as the reverse drag radius and defines the fault tip line where the displacement reaches zero. This decrease is non-linear, and along the z-direction the profile follows the curve illustrated in 3.11 (Cardozo et al. [2]).

The displacement field within the fault tip line is defined as a vector field in 3D, with a vector

$$\mathbf{D}(x,y,z) = \{D_x(x,y,z)\vec{\mathbf{u}}_1, D_y(x,y,z)\vec{\mathbf{u}}_2, D_z(x,y,z)\vec{\mathbf{u}}_3\} \tag{3.3}$$

for each spatial location. Here $\vec{\mathbf{u}}_1, \vec{\mathbf{u}}_2, \vec{\mathbf{u}}_3$ are the unit vectors along x-,y- and z-axes respectively. The displacement vector field is given by

$$D_x(x,y,z) = 0 \tag{3.4}$$

$$D_y(x,y,z) = \begin{cases} d(x,y) \cdot \gamma \cdot \left[1 - \frac{|z-f(x,y)|}{R}\right]^2 & \text{for } f(x,y) < z < R, \\ d(x,y) \cdot (\gamma-1) \cdot \left[1 - \frac{|z-f(x,y)|}{R}\right]^2 & \text{for } -R < z < f(x,y) \\ 0 & \text{for } z < -R \text{ or } z > R. \end{cases} \tag{3.5}$$

$$D_z(x,y,z) = f(x, y + D_y(x,y,z)) - f(x,y). \tag{3.6}$$

Note that this definition implies no strike-slip displacement ($D_x = 0$). The expression for the z-component given by 3.6 implies that the distance from the point to the fault surface (along the normal) remains constant through the displacement operation.

Figure 3.12 gives a sketch of the 3D displacement function in a cross-section for $x = x'$. The decreasing profile of $D_y(x,y,z)$ away from the fault surface is shown in Figure 3.11.
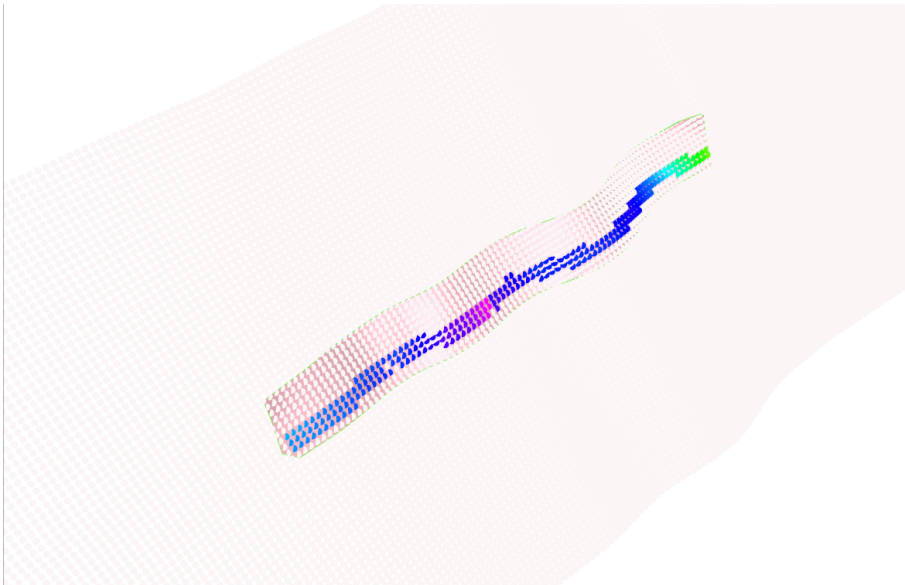
Figure 3.8. The fault F1 from the Emerald field structural model with known displacement values found from fault lines in a local regular grid. Non-coloured nodes mean no known value.

The effect of the 3D displacement operator is shown in Figures 3.13 and 3.14 for a linear and a listric fault respectively and with different parameter values. Figure 3.13 shows an example with six horizons and a planar fault where the displacement is equally distributed on hanging wall and footwall sides ($\gamma = 0.5$) and where the reverse drag radius is 1000 m in one case and 3000 m in the other.

In Figure 3.14 six horizons are subject to a listric fault with all displacement to the hanging wall side and with different reverse drag radius.

3.2.6 Fault surface uncertainty model

The fault surface uncertainty model is defined from a prior best guess surface and hanging wall and footwall uncertainty envelope surfaces. Section 4.2 describes how the envelope is generated in HAVANA. These surfaces define the prior uncertainty model for the fault surface together with a user defined variogram. The position of the fault surface for a location (x, y) in the local coordinate system is given by a user specified probability distribution P. The options for this distribution are uniform, triangular and normal. For the two first, the envelopes defines exact minimum and maximum limits, while for the latter it defines two standard deviations, meaning that the fault surface is inside with a probability of approximately 95%.

In the presence of well picks with associated observation uncertainty and well path points that do not hit the fault a conditioned uncertainty envelope is calculated so that the fault surface fulfill the conditioning and simualtion and prediction can be performed inside the updated uncertainty envelopes. Details on how the a posterior uncertainty model is calculated is found in Havana Technical Documentation [8].

3.3 Truncations

Truncations in HAVANA are specified in pairs of truncating and truncated faults giving a set of truncation rules. For each pair an indicator specifies if the fault is truncated on the hanging wall or footwall side of the truncating fault. Two faults intersecting each other do not necessarily define a truncation, so the model has flexibility to handle both λ-, y- and x-faults. This means that the truncation rules are applied to join fault blocks together in a specific fault network representation.
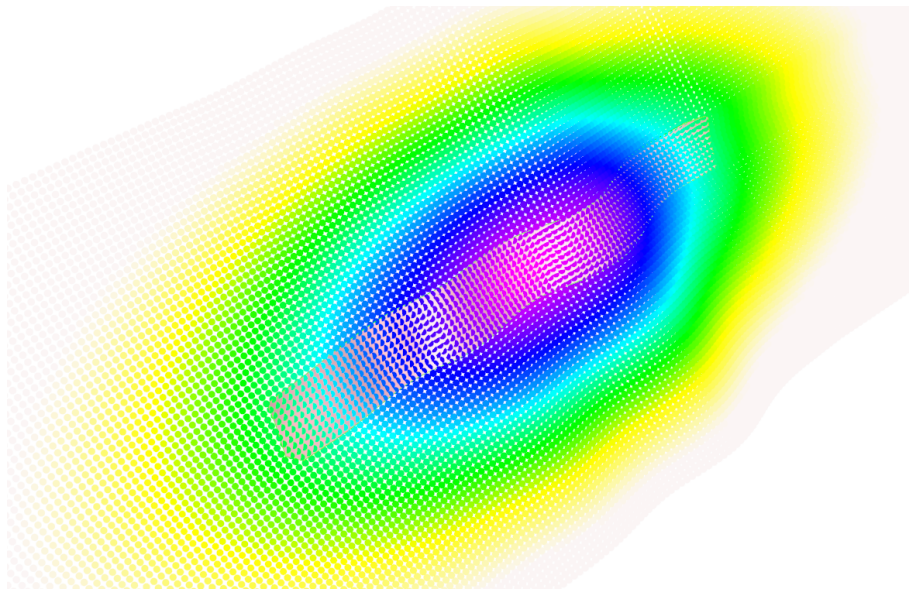
Figure 3.9. The fault F1 from the Emerald field structural model with the elliptical trend estimated from the data and input parameters.

This is partly what is described as the fused fault block approach in Hoffman and Neave [6]. The 3D displacement field for the truncated fault is also truncated. This is illustrated in Figure 3.15 showing two faults where the smaller is truncated by the larger. All displacement is on the hanging wall side. The volume affected by the displacement operator of the larger fault is shown in green in the left picture, while the volume affected by the displacement operator of the smaller truncated fault is shown in red to the right.

Figure 3.16 shows the effect of truncation on the estimated displacement field for fault F2 in the Emerald field model. This fault is truncated by F1. The left figure shows the displacement field for F2 without taking the truncation into account, while the right figure shows the displacement field when accounting for the truncation by F1.

The truncation rules can be applied to define an ordering of faults. This is based on the principle that a truncating fault is older than the faults it truncates. The fault lines from the structural model represents the net effect of the displacement from several faults. To find the fault lines representing the actual displacement from the fault when the fault incident occured, a sequential restoration of the faults is needed. The fault lines from the youngest fault is in this sense correct and the 3D displacement field for this fault can be generated. If the fault lines from the second youngest fault are influence by the younger, these must be restored back to its original position by applying a reverse fault operator for the younger fault. These restored fault lines are then used for generating the displacement field for this second fault. This procedure is applied to all faults in the correct order.

Since every point in the reservoir volume might be affected by the displacement from several faults, a net displacement vector for the point should account for the joint effect from the displacement of all faults that influence the point. The net displacement operator is found by restoring the point to its non-faulted position for every fault, and doing this sequentially from the youngest to the oldest fault.

In the restoration, or de-faulting operation, a reverse fault operator is needed. To find an explicit expression for the inverse of the 3D displacement vector D defined by expressions (3.4) - (3.6) is not possible since there is no closed expression giving $d(x, y)$ as a function of y. Thereby giving
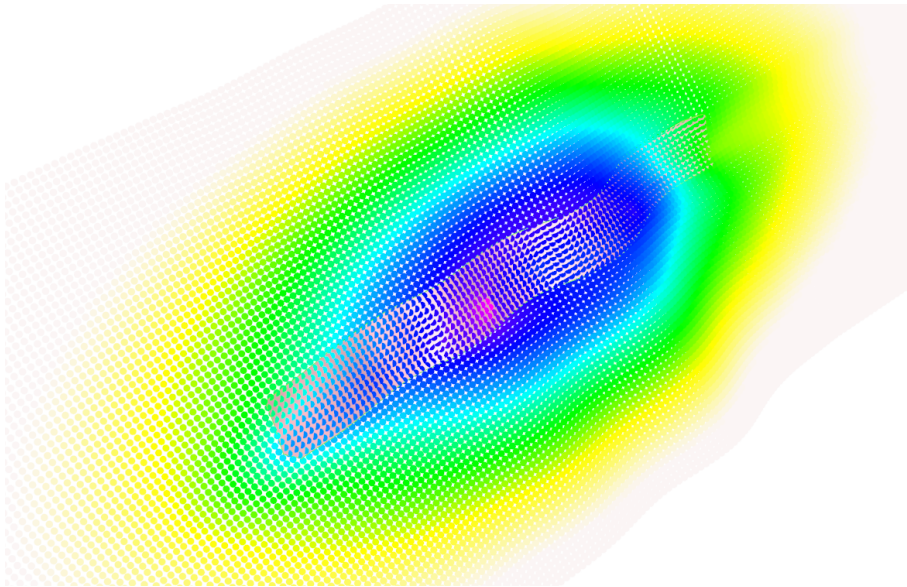
Figure 3.10. The fault F1 from the Emerald field structural model with the displacement field kriged with values found from fault lines in a local regular grid. White dot means no known value.

an undefined equation set. One possible solution is to create a 3D grid for the volume influenced by the fault and calculate the faulted position for each node in the grid. The inverse operator could then by defined by mapping back from the cube of faulted positions using interpolation. With many faults, however, this is both memory and CPU demanding. The alternative approach used, is to calculate the inverse of the 2D displacement function $d(x, y)$ as one grid for hanging wall side and one grid for footwall side. The faulted point is projected onto the fault surface and the inverse 2D function is used by taking into account the side of the fault and the distance to the fault surface. Since this will only give correct answer for constant displacement, an iterative approach using binary search is used until the discrepancy is within some predefined tolerance level. Since the displacement function can be assumed to be quite smooth, this gives a precise and fast approximation.

In some fault sets there are cyclic truncations. This may happen if several faults are active at the same time, and is also sometimes done in RMS to model a single fault in multiple parts. HAVANA does not currently handle this properly, however HAVANA tries to detect this and give a warning.

3.4 Attribute Groups

HAVANA supports grouping of faults by attributes which can be convenient for filtering. Attribute groups can be created in various ways. The group-faults action can assign values according to the length of the fault and simulate-secondary-faults can created an attribute group for the simulated faults specifically. Not all faults will necessarily have a value assigned within an attribute group, in which case they are considered to be undefined within that attribute group.

Attribute groups can be used by a subset of the Havana actions to serve as a filtering mechanism that depends on the action being executed. When the name of an attribute group is provided along with a set of value names (group names), only the faults that have the corresponding values will be considered in the execution of the action.

An example of an attribute group can be an attribute group based on length. The name of the attribute group might be "Size" and the possible values "Short" and "Long" (and unassigned). Each of the faults are assigned one of the two values or left unassigned. When generating fault
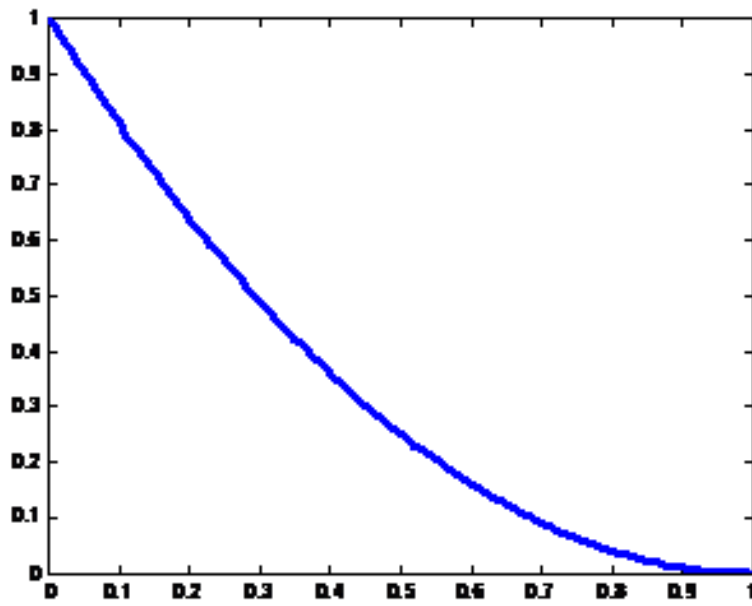
Figure 3.11. Profile showing how the displacement along the local y-direction decreases with distance from the fault surface.

statistics, it is then possible to generate statistics only for "Long" faults.
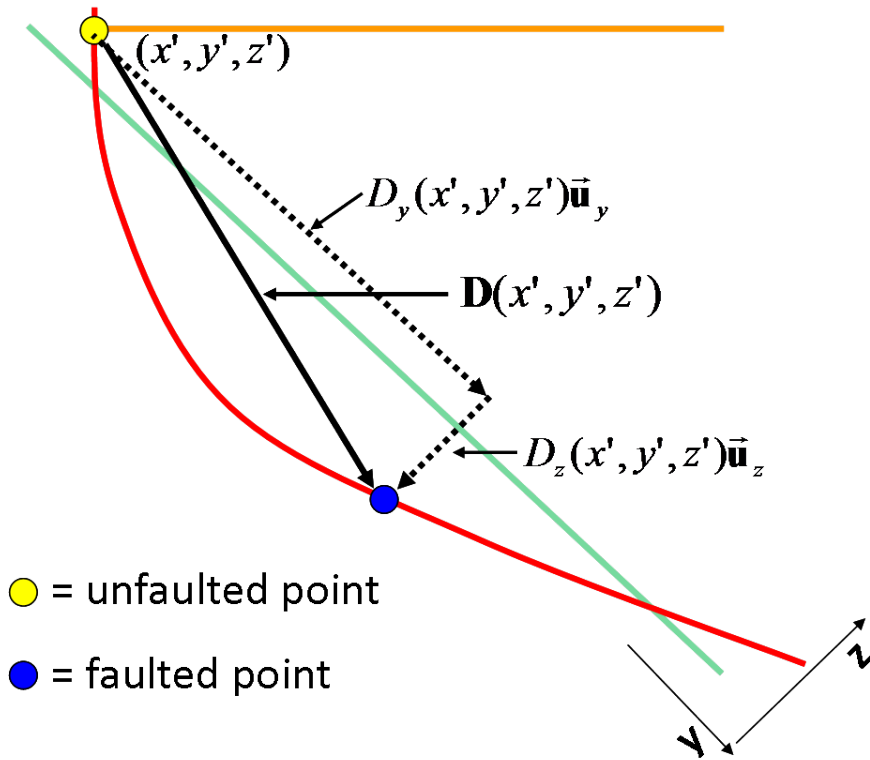
Figure 3.12. Vertical cross section showing the 3D displacement vector $\mathbf{D}$ decomposed into D_y and D_z.
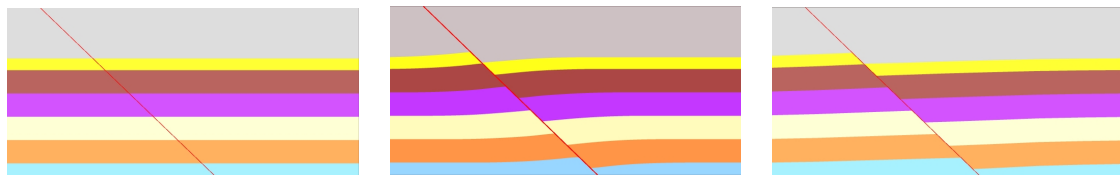
Figure 3.13. Left: Unfaulted. Middle: Fault with displacement equally distributed to hanging wall and footwall sides and reverse drag radius 1000m. Right: Fault with displacement equally distributed to hanging wall and footwall side and reverse drag radius 3000m.
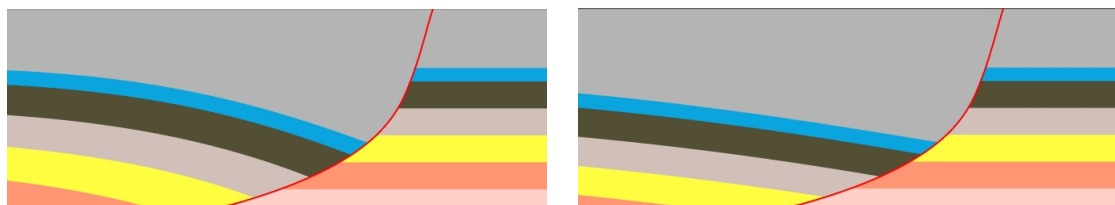
Figure 3.14. Listric fault with all displacement on hanging wall side Left: Reverse drag radius 2500 m. Right: Reverse drag radius 5000m.
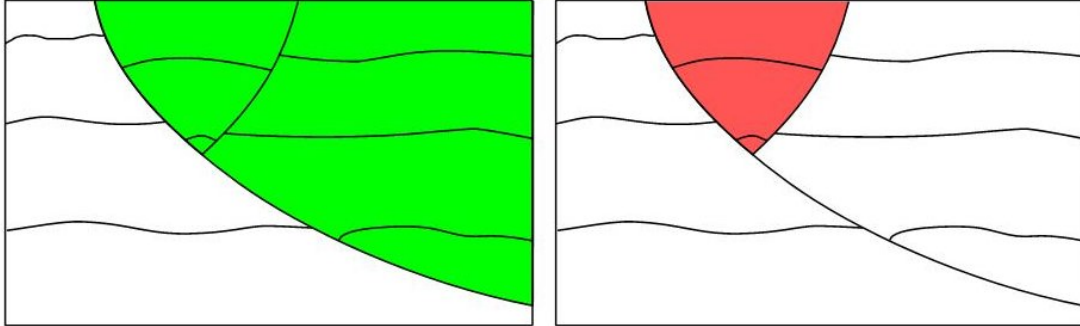
Figure 3.15. Volume affected by large fault in green and volume affected by smaller fault in red.
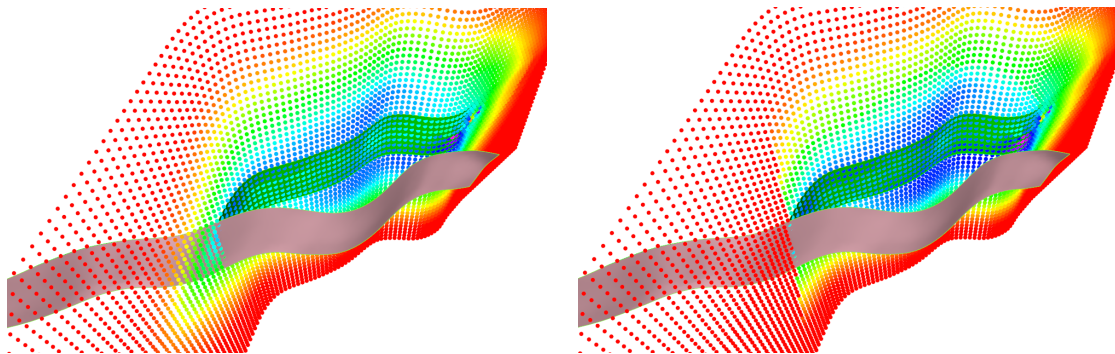
Figure 3.16. Emerald field model showing displacement field for fault 2. To the left the truncation by fault F1 is not accounted for in the displacement field. To the right the displacement field is truncated by F1.

4 HAVANA actions

The HAVANA program is run by performing different actions with commands and user specified parameters. This chapter gives a description of the purpose and result of all actions and how they depend on other actions, the most important input and output and general advice for using the action. A full detailed description of all keywords that are available for the actions is given in the user reference manual (see Section 6).

The following actions are available in the current version of HAVANA:

ImportRMSFaultData - reads input data generated from RMS structural model

FaultUncertaintyModel - sets up an envelope or interval around the faults that define the possible volume for the fault's location

ModifyFaultSurface - changes position or dip / strike of the fault surface

SimulateFaultSurface - simulates the position of faults according to a specified uncertainty model

ModifyDisplacement - changes the displacement field for faults by scaling or adding displacement

SimulateDisplacement - simulates the displacement field and fault tip line according to given uncertainty structure

RemoveFaults - removes faults from an existing set of faults

AddFaults - adds new faults to an existing set of faults

SetDisplacement - gives a new or existing fault a displacement field

SimulateSecondaryFaults - generates new sub-seismic faults around larger, seismic-sized primary faults

SimulateFaults - generates new sub-seismic faults according to global trends

GroupFaults - groups the faults into two groups based on size

UpdatePoints - updates point sets (horizons and fault lines) after modifications done by other HAVANA actions

UpdateGrid - updates eclipse grid after modifications done by other HAVANA actions

UpdatePermeability - updates the permeability values in an existing grid according to new small faults

CreateFaultGrid - create local grid refinement for 3D modelling of the fault zone

MergeFaultGrid - merge fine-scale fault zone grid with coarse scale host grid

CalculateFaultSeal - calculate fault sealing parameters and export fault transmissibility multipliers

FaultStatistics - calculate statistics of a fault set

The actions are listed in an order such that in the cases where there is dependency between two actions, the former in the list must be run before the other in the same workflow.

4.1 ImportRMSFaultData

This action should always be run first when a new structural model is going to be used by HA-VANA. The purpose of the action is to read all the data from the structural model directory, to construct the set of surface faults in this model including truncations and estimate the displacement fields for the faults.

4.1.1 Displacement fields

The estimation of the displacement fields is based on input fault lines from the structural model and prior parameters given by the user through the commands <displacement-variogram>, <displacement-settings> and <default-displacement-settings>. The trend for the 2D displacement field defined on the fault surface is an ellipse. The user can specify prior values for the relationship between the length and the height of the ellipse and the relationship between the maximum displacement at the ellipse center and the length. The input fault lines are used as data points since they give information about the displacement at the intersections between the horizons and the fault. The trend ellipse is estimated by an optimization algorithm where an objective function is minimized. This objective function is constructed such that it measures the squared discrepancy between the prior ellipse and the observed displacement in the position of the data points and in addition penalizes discrepancies from the user-specified parameters. When the trend has been estimated, the 2D displacement field is constructed by the use of simple kriging on the data values and with the specified variogram. The variogram defines the correlation between displacement values at different locations and has impact on the smoothness of the field.

The HAVANA default values is that the length is twice the height and that the length is 100 times the maximum displacement. The default variogram for the 2D displacement field is a spherical variogram with a range of 1000 meter in strike direction and 2000 meter in dip direction.

After the action is run, the HAVANA logfile gives detailed information of the estimated displacement trend in the following manner:

```
F5
  Inital elliptic fault tip model:
    Fault_length_height parameter:        2.0
    Max displacement length ratio:        0.01
    Max displacement length power:        1.0
    length = 100.00 * max_displacement^1.00
    height = 0.50 * length
    Range:  1000.0

  Estimated trend:
    Trend center : (       717.12, -73.82, 0.00)
    Trend max    :         59.69
    Trend length :       5309.51
    Trend height :       2654.76
    In pos.      : (       717.12, -73.82, 0.00)
```

The 3D displacement field is constructed from the 2D-field by two other parameters, namely the hanging wall/footwall asymmetry (default 0.7) and the influence range of the displacement field in the direction of the horizontal projection of the fault's normal vector (default 1000 meter).

The parameters for the displacement fields can be specified explicitly for each fault, this is done by using <displacement-settings>. The command <default-displacement-settings> will be applied to all faults that are not specified separately. This command must be used if other values

than the HAVANA default values are desired.

In some cases, the fault lines in the structural model does not give relevant information about the displacement of the fault. Then the user can specify that a displacement field should not be estimated by this action and the estimation is skipped. In this case the user must use the action `<set-displacement>` (see Section 6.10) to give the fault a displacement field.

4.1.2 Output

The most important output from the action is a directory with all files that define the fault set on the internal HAVANA format (see Appendix A.10). In addition, if the user specifies a debug output directory, this action will give valuable information for analyzing the structural model and the estimated displacement field and to perform general quality control. If debug output is chosen, a separate directory with this information is written, including files on RMS internal points format of the generated displacement fields. These files can be imported to RMS for visualization. The user reference gives a specification of the debug output (6.2.8).

4.2 FaultUncertaintyModel

4.2.1 Background and purpose

When a fault model is being built, the faults' locations are uncertain due to several sources. The most important are:

- Interpretation uncertainty.

- Depth migration uncertainty.

- Seismic resolution.

The interpretation uncertainty comes from the fact that around faults the quality of the seismic data usually is very poor, together with the fact that faults are modelled as two dimensional planes even though they actually are three dimensional structures. The interpretation uncertainty encompasses both the existence of faults and fault patterns, and the location and local shape of the faults. The depth migration uncertainty is due to uncertainty in velocity model, and uncertainty in the actual seismic signal path within the reservoir due to non-horizontal velocity contrasts. In the present version of HAVANA only one measurement for uncertainty is used, encompassing both the migration uncertainty and the interpretation uncertainty. Correlation in the uncertainty due to the continuity of depth migration errors is currently ignored. From the seismic interpretation we attain a base case for the fault geometry. The uncertainty is modelled as an envelope around this base case where the fault is allowed to reside. The action `<fault-uncertainty-model>` is used to construct two surfaces (as point sets), one on each side of the fault, that defines the uncertainty envelope.

4.2.2 Input and method

The input for construction of the envelope is either a set of points on each side of the fault or a constant value giving the envelope's distance from the fault on each side. In either case a boundary surface on both the footwall(FW) and the hanging wall(HW) side is constructed and will be the output from the action. In the case that point sets are input, these points are treated as conditioning points in simple kriging interpolation to populate a 2D grid on each side of the fault. Every input point is classified as either HW-point or FW-point depending on which side of the fault surface it is located. In the kriging a constant trend is assumed for each side. These trends are taken as the average distance from the fault surface to the input point along the faults' normal vector. The default variogram being used is a spherical with range 1000 meter in strike direction and 2000 meter in dip direction. Figure 4.1 shows a fault and the uncertainty envelope defined

by boundary surfaces in constant distance 500 meters from the base case location for the fault surface.
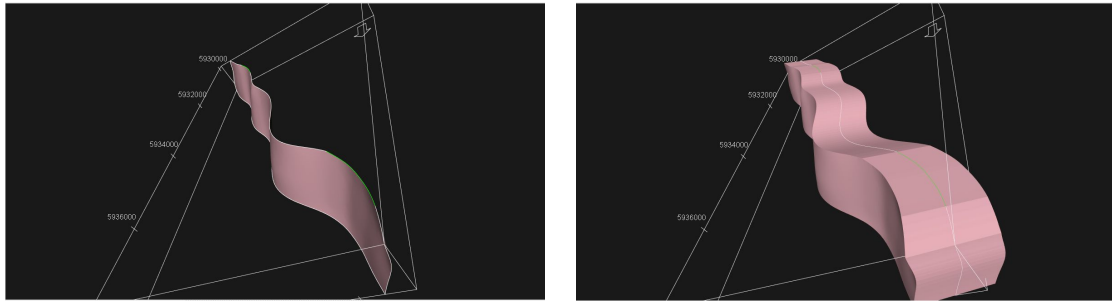
Figure 4.1. Emerald field model showing fault F1 in base case location (left) and with uncertainty volume at a constant distance of 500 meters on each side (right).

4.2.2.1 Use in HAVANA workflows

This action must be run before the action `<simulate-fault-surface>`, since the latter depends on an uncertainty model being defined. The same is the case when using the action `<modify-fault-surface>` with translation relative to an uncertainty envelope.

4.3 ModifyFaultSurface

The purpose of this action is to change the position and/or orientation of the fault surface in a deterministic way. This includes translation in the $xy-$plane and rotation by changing the strike and the dip. Translation can be done parallel to the x- or y-axis or along the projection of the faults' normal vector in the $xy-$plane. When moving the fault along the normal-projection, this can be done by an absolute distance or relatively within a fault uncertainty envelope. In the latter case the value given must be between -1 and 1 with negative value meaning moving in footwall direction and positive in hanging wall direction. In case this option is used, the action `<fault-uncertainty-model>` described in Section 4.2 must be run prior to `<modify-fault-surface>` so that the uncertainty envelopes for the relevant faults are generated. On the other hand, if an uncertainty envelope is defined for a fault and moving an absolute distance in any direction is chosen, the envelope for this fault is ignored. Note also that the uncertainty envelope with constant value is defined in the faults' normal direction, so if the fault has a dip, the effective translation in the $xy-$plane will be smaller than the absolute distance given for the envelope.

The input and output in this action are the original fault set on internal HAVANA format and the modified fault set on the same format. In addition the modified faults are written as RMS point sets.

4.3.1 Change dip or strike

It is possible to change dip or strike of a fault around a specified global point. This implies a rotation around a point. If no point is specified, the faults' reference point is used. If the rotation point does not coincide with the fault surface, it is projected onto the fault surface along the fault's normal vector.

The rotation is done in three steps. First the reference plane is translated to the new point, this is done without moving the fault surface. The reference plane is then rotated and the fault surface follows this rotation. The last step is to translate the reference plane an equal distance back with respect to the rotation. Figure 4.2 shows a fault rotated 30 degrees both around the reference point, and around another point.
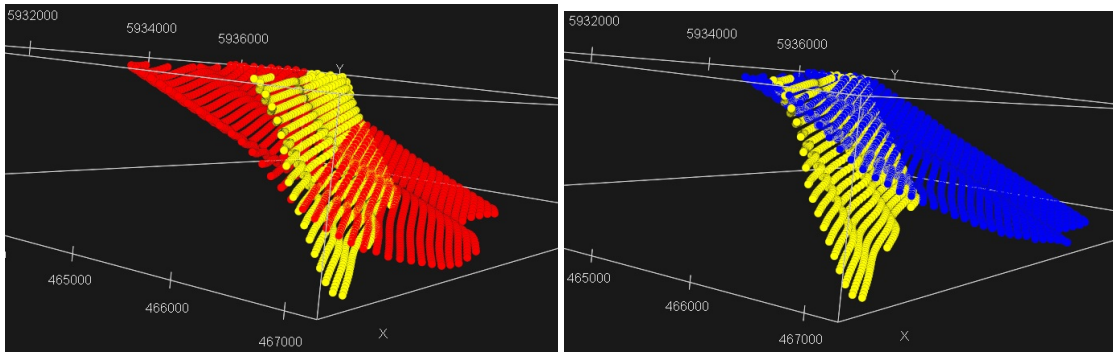
Figure 4.2. The left figure shows the original fault (yellow) and the fault with a dip-rotation of 30 degrees around the reference point (red). In the right figure the same fault has a dip rotation of 30 degrees (blue), but the rotation is done around a point near the top of the fault surface.

4.3.2 Well conditioning

In this action it is possible to condition the fault geometry to well picks. This is done by choosing well-conditioning and giving an input file with well picks specified by the format in Appendix A.4.1 together with a variogram to specify the correlation along the fault. The default variogram is spherical with range 1000 meter in strike direction and 2000 meter in dip direction. With this option, the well observations are conditioned by interpolating the fault surface using simple kriging interpolation. The well conditioning can be used without any modifications just to condition the initial fault surface to observations or it can be used to condition a modification, for instance a translation, to the well picks. Figure 4.3 illustrates the effect of moving a fault unconditioned or conditioned to well picks.
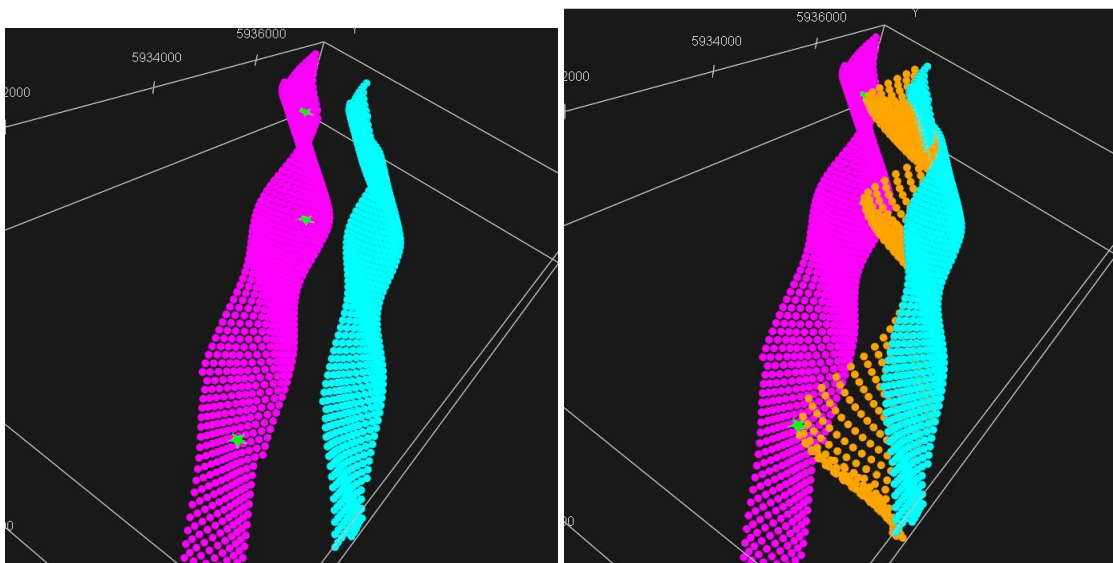
Figure 4.3. To the left is the original fault (purple) together with the same fault moved 1000 m. along the faults normal direction (cyan). Three well pick points are shown in green. In the figure to the right, the fault is moved with the same distance, but conditioned to the well picks (yellow). fault to the right.

4.4 SimulateFaultSurface

The purpose of this action is to generate one or more simulated realizations of the fault surface geometry within an uncertainty envelope set up by the action FaultUncertaintyModel described in Section 4.2 and by using the fault surface uncertainty model described in Section 3.2.6.

4.4.1 Method

The position of the simulated fault surfaces around the base case can follow either uniform, normal or triangular probability distribution. In the case of uniform distribution the simulated fault surfaces are evenly distributed within the envelope, and in the case of triangular distribution all fault surfaces are located inside the uncertainty envelope with highest probability at the base case and lower probability towards the edges of the uncertainty envelope. In the case of a normal distribution approximately 95% of the simulated fault surfaces will be inside the uncertainty envelope. All calculations are done within a Gaussian (normal-distributed) domain. This is achieved by transforming the input data (fault surface and well observations) using quantile mapping, while the resulting simulated surface is transformed back to the real domain using the same technique. Each realization of a fault surface is made by first simulating a 2D Gaussian field with the specified variogram and zero mean. This gives a residual surface between the base case specified in the input and the simulated surface. Then at each position in the 2D field, the limits of the uncertainty interval is found from the envelopes, and for the drawn residual value at this point a number between 0 and 1 is found using the normal distribution quantile. The actual simulated residual is then found by using the same quantile in the given distribution (uniform, normal or triangular). This residual is added to the fault surface ensuring that the simulated surface has the right spatial distribution and resides inside the uncertainty envelope. See Havana Technical Documentation [8] for more details.

In addition to simulation, prediction can be chosen. This gives the mean surface in the fault uncertainty model together with one surface on each side representing the p'th and (100-p)'th percentile of the fault uncertainty model.

Figure 4.4 shows the predicted mean surface for Fault F1 in the Emerald case conditioned to three well picks, and 10 simulated, conditioned realizations from a uniform distribution inside the uncertainty envelope with width 1000 meters.
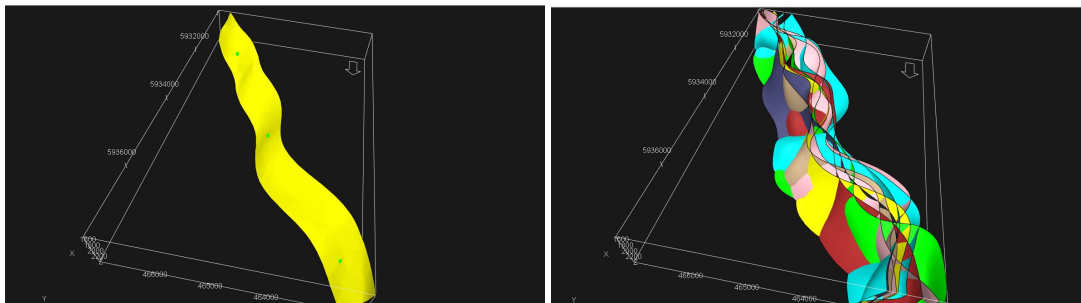
Figure 4.4. To the left is mean fault generated by using the predict command, conditioned to three well picks (in green). In the figure to the right, 10 simulated realizations with uniform distribution inside a constant envelope of 500 meters to each side and conditioned to the three well picks. The variogram being used is a Gaussian with range 2000 meters. fault to the right.

4.4.2 Input parameters

The parameters for the simulation process are a list of which faults that are being simulated, the type of distribution (default is uniform), number of realizations (default 1), variogram for the fault geometry (default is spherical with range 1000 meter in strike direction and 2000 meter in dip direction) and random seed (default is a seed taken from the local computers current time). A random seed is necessary on input and the random seed generator inside HAVANA ensures that the realizations are independent samples from the fault uncertainty model. Simulation can be chosen for either one, some or all faults and all the realizations of the simulated faults are written as RMS point sets to files with unique names.

The conditioning to well picks is done using simple kriging with an uncertainty on the observations. This uncertainty is interpreted as two times the standard deviation in a normal distribution for the fault pick location. It is also possible to specify well path points for wells that the fault surface should avoid. In this case the uncertainty envelopes are modified such that the well path points to avoid are outside the envelopes.

4.5 ModifyDisplacement

4.5.1 Purpose and effect

This action can be used to change the displacement field for one or several faults. There are two options, either a scaling of the existing displacement field by a constant factor, or adding a throw to the displacement everywhere. In the first case the size of the fault remains the same as the fault tip line is unchanged, but in the other case a new fault tip line will be estimated. Normal faults have positive displacement. This means that adding a positive value for the throw increases the size of the fault while adding a negative value decreases it. For reverse faults, the displacement is negative, so to increase the size of the fault, a negative value has to be added to the throw.

4.5.2 Method

In case of scaling the 2D displacement function $d(x, y)$ defined in 3.2.5 is modified to be $d^*(x, y) = Cd(x, y)$ where C is the scaling factor. This scaling influences also the displacement away from the fault by the same scaling factor. Adding a constant D means that we get $d^*(x, y) = d(x, y) + D$. This initiates a process for estimating a new displacement field and new fault tip line. The user has the option to specify new parameters for the displacement field by a new variogram and the commands `<displacement-settings>` (for named faults) or `<default-displacement-settings>` (for other modified faults). If these commands are not given, the parameter values used in the estimation of the original displacement fields are used. Note that new parameter values are only relevant when adding throw.

After performing the modifications the modified faults (with displacement fields) are saved to files on the internal HAVANA format to be used in other actions. Point sets for the updated fault surfaces can also be exported to RMS point set files to be used to build updated fault surfaces in RMS.

Figure 4.5 shows the displacement field for the F1 fault in the Emerald case before and after adding a throw of 50 meters.
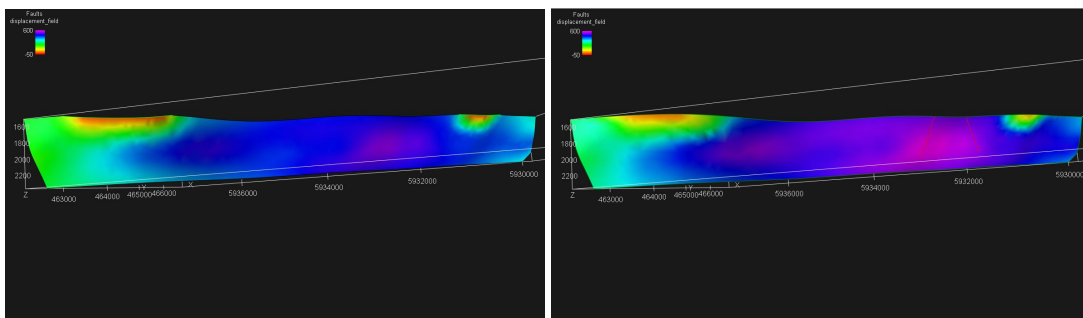
Figure 4.5. To the left is the original displacement field estimated after import from RMS. In the figure to the right a throw of 50 meters has been added.

4.6 SimulateDisplacement

Note that this Action is not fully tested. The present version is to be regarded as an experimental version mainly for research and development purposes. In this action, the fault displacement fields and the corresponding fault tip lines can be predicted or simulated from a set of input displacement observations. In both the case of prediction and simulation, the original displacement field, either found as described in 3.2.2 or given through the action `<set-displacement>`, is sampled to give a set of values that are used to estimate the trend and as conditioning values for the simple kriging. The values are sampled by every fifth grid node in local x- and y-direction inside the fault tip line, giving that every 25th value is used. This procedure means that the fault lines are not used in this action. The trend is estimated based on these sampled values as described in 3.2.3. There is a significant uncertainty in the displacement value, and this uncertainty is related to the seismic resolution which in the case of faults represents the smallest observable fault displacement. An effect of this is that the fault tip of the estimated trend can be different from the fault tip found when importing the fault by the action `<import-rms-fault-data>`.

4.7 RemoveFaults

This action removes one or several faults from the fault set. The names of the faults that are removed are specified in the modelfile. The effect is that a reduced fault set is generated and written to file. This fault set can be used in other actions to study the effect of the removal. The truncations related to the removed faults are also removed, so a new set of updated truncation rules is made. The typical follow-up of this action is to update the grid (by the action `<update-grid>` (see 6.15) or to update point sets (like horizons) by the action `<update-points>` (see 6.14).

4.8 AddFaults

In the AddFaults action new faults are added to the original fault set. New faults can either be included as elliptic faults, or as fault surfaces extracted from RMS without displacement field. The displacement field for these faults must be set in the action `<set-displacement>`, see 4.9. In each case new truncation rules related to the new faults are read from file and added to the fault set. The action `<add-faults>` returns a modified fault set with old and new faults.

4.8.1 Elliptic faults

The new elliptic faults are read from file, where each new fault is given by one line. See A.2 for the file format. The parameters defining an elliptic fault are $(x, y, z)-$ coordinates for the centre point of the ellipse, the maximum displacement, the asymmetry of the displacement, the strike and dip angle, the length and height of the ellipse, the range and the slip type (NORMAL or REVERSE). For the elliptic faults a grid resolution for representing the fault as a surface on a 2D grid can be specified. The default grid resolution is 10 meter. All the faults that are added as elliptic faults must be specified in the same input file.

4.8.2 Surface faults

General fault surfaces extracted from RMS can be read from file as point sets. These files are found in the export directory for the structural model that comes from RMS. There are one file per fault in this directory. These faults have no displacement field. To generate a displacement field, the action `<set-displacement>` must be run on the modified fault set after the new fault is added.

4.8.3 Place in workflows

The typical follow-up of this action is to set the displacement by the action `<set-displacement>` if necessary and then update the grid (by the action `<update-grid>` (see 6.15) or to update point

sets (like horizons) by the action `<update-points>`(see 6.14)

Figure 4.6 shows the Emerald fault set with the original faults and with three added faults. Truncation rules are applied to the new faults
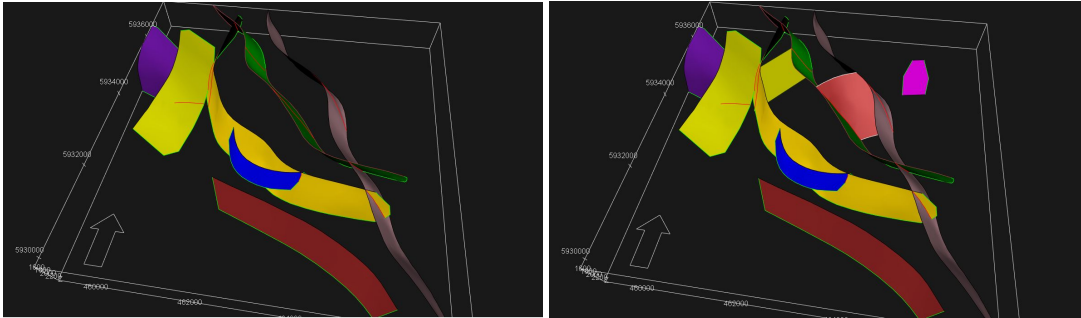
Figure 4.6. To the left is the original Emerald fault set. To the right three new faults have been added.

4.9 SetDisplacement

In this action elliptic displacement fields are added to existing faults or fault surfaces. The action is relevant when new fault surfaces without displacement field have been added to the fault set in the command `<add-faults>`, see 4.8, however can also replace existing displacement field in any of the faults given in the fault set.

4.9.1 Setting displacement from maximum displacement

The displacement field can be specified by giving the maximum displacement (using the command `<set-max-displacement>`), which requires that fault displacement parameters are given or default values being used. In case they should be user-given, this is done by the command `<displacement-settings>` in the same manner as for the action `<import-rms-fault-data>` (see Section 4.1) The default values for the displacement settings are the same as for `<import-rms-fault-data>`. The position where the maximum displacement occurs is either specified through user input or set automatically to the center point of the fault surface. The user can specify the name of the horizon where the maximum displacement should be located. This requires that the fault lines are given as input on file. This file can be found as *fault_lines.rms* in the folder with the exported structural model from RMS specified by the command `<input-structural-model-directory>` under the action `<import-rms-fault-data>`. Then the maximum displacement is put at the center of the fault lines for this specific horizon. If fault lines are given, but no horizon specified, the maximum displacement is located at the centre point of all fault line points for the given fault. Otherwise, the maximum displacement is set at the centre point of the fault surface.

4.9.2 Setting displacement from elliptic fault

The displacement field can be set by specifying the elliptic fault parameters in an input file of the same format that was used when adding new elliptic faults in the action `<add-faults>` (see A.2 for the fileformat). This requires that the centre point of the ellipse is located at or close to the fault surface of the corresponding fault.

4.10 SimulateSecondaryFaults

The purpose of this action is to simulate smaller (sub-seismic) secondary faults clustered around larger (seismic sized) faults. The simulated faults are elliptic faults. The input to the action is an original fault set on internal HAVANA format consisting of the primary faults. The modelfile specifies several aspects of the simulation algorithm.

4.10.1 Simulation volume

The primary faults must be simulated inside a well-defined volume. The default is the bounding box given by the initial structural models horizon modelling box. Sometimes using the default modelling box is inappropriate in particular in the vertical direction since it may be too large. In this case it is useful to specify the top and base surfaces of the modelling volume explicitly using the command to specify simulation volume directly. The simulation volume in the $xy-$plane is defined by $x-$ and $y-$ reference coordinates and length in each direction, while the top and bottom of the volume can either be given by maps (2D surfaces) or a constant $z-$ value. The recognized file types for the maps in the present version of HAVANA are Storm Binary, IRAP Classic Ascii and Roxar Text. To avoid any edge effects on the top and bottom of the volume, a vertical buffer that is added to the top and the bottom of the volume is used. If this is not specified by the user, it is calculated by HAVANA based on the potential size of the displacements of the secondary faults.

4.10.2 Displacements and distribution of secondary faults

The total number of simulated faults is deterministic and specified by the user. The number of simulated faults that is assigned to each primary fault depends on the displacement of the primary fault. The larger displacement, the higher number of secondary faults attached. For each secondary fault a maximum displacement value is drawn from a truncated fractal distribution (see user reference, Section 6.11.14 for more details). Only primary faults with a maximum displacement larger than the drawn maximum displacement are considered to have this simulated fault associated and the probability for that is proportional to the primary fault's maximum displacement. This procedure easily implies that very large faults get nearly all secondary faults associated with them. This effect can be reduced (or increased) by introducing a displacement weight. To give smaller primary faults higher probability of having secondary faults, the weight should be set to a number smaller than 1.0. The default value is 1.0. For example, if there are two primary faults with displacement 100 meters and 10 meters respectively, approximately 91% of the secondary faults will be assigned to the larger fault with default displacement weight. If this displacement weight is set to 0.7 the fraction is reduced to 83 %.

The other parameters for defining the displacement field for the simulated faults are HW/FW asymmetry, relationship between maximum displacement and length, relationship between length and height and relationship between influence range away from the fault and length and height. All these are subject to probability distributions with user-specified parameters. Details are in the user reference, Section 6.11.14.

4.10.3 Simulation of position and orientation of secondary faults

The simulated secondary faults can belong to different groups where the faults from different groups have different probability distributions for position and orientation relative to the primary fault. This is regulated by sub-commands under the command `<simulation-settings>`. Each simulation setting can apply to one or several primary faults, and the same primary fault can belong to several simulation settings. When the latter is the case, a fraction is specified for each simulation setting telling the relative fraction of the secondary faults belonging to the primary fault that follows this specific simulation setting. If for instance there are two simulation settings for primary fault F1, one with fraction specified to 1.0 and the other with fraction 2.0. Then if 30 secondary faults are associated to F1, 10 of them will get their parameters from the first simulation settings, and the 20 others from the second one. If not all primary faults are covered by some specific simulation setting, at least one simulation setting must be specified with no primary fault name. This simulation setting will then apply to all primary faults that are not covered by any of the other simulation settings. If no simulation setting without a primary fault name is given,

it means that the primary faults that are not associated with any simulation setting will have no secondary faults.

The secondary faults are drawn independently of each other, meaning that there is no interaction between them. The position of the center point is decided by three stochastic variables for position in the strike-, dip- and normal-direction of the primary fault. Along the strike- and dip-direction the position is Gaussian distributed with mean and standard deviation given by the user. The parameters are relative to the primary fault, with zero being at the center of the primary fault and -1 and 1 being at the fault tip position. The default mean value is 0.0, but through the possibility of choosing a non-zero mean value, the secondary faults can be asymmetrically distributed along the fault in either of the directions. Choosing standard deviation equal to 0.5, which is the default value, means that approximately 95% of the simulated faults will have center point within the fault tip. In the normal direction the same principle applies, but here it is possible to choose absolute distance instead of relative distance. Relative distance means relative to the reverse drag radius (influence range) (see Section 3.2.5) for the primary fault, but this can be substituted by a mean and standard deviation measured in absolute distance from the primary fault. In the normal direction positive distance means on the hanging wall side, while negative is on the footwall side. An input parameter gives the relative intensity for which side of the primary fault the simulated faults will be located. The default value for this parameter is 0.5 meaning that there is equal intensity on hanging wall and footwall side. This means that the distribution for the position in the normal direction will follow the Gaussian distribution given through the input mean and standard deviation. If a value other than 0.5 is chosen, the distribution will no longer be a Gaussian, since the drawn values from the underlying Gaussian distribution will be adjusted by the asymmetry intensity value. The effect of this parameter must be found by trial and error. It should be noted that distances away from the primary fault is measured in the faults' local coordinate system, so when the dip of the fault is not vertical, this distance is not measured horizontally. This influences the intensity distribution and might give unwanted effects. The more the dip deviates from the vertical plane, the more the possibility to visualize the intensity grid around the fault should be applied, as a tool to choose the right parameter settings. This is shown in Figure 4.7.
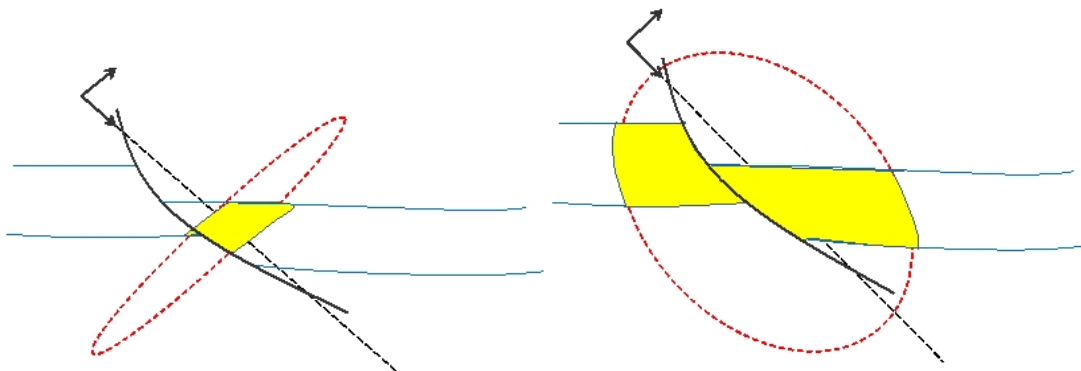
Figure 4.7. Two the left is shown in yellow the intensity (inside 1 standard deviation) for position of secondary faults in the dip and normal direction from the primary fault. In this case the distribution is narrow in the dip direction. To the right is shown the intensity (inside 1 standard deviation) for position of secondary faults in the dip and normal direction from the primary fault in the case with a wider distribution in dip direction. The blue horizons are for instance the top and base of the model box.

The orientation of the secondary faults is set by drawing strike and dip values from truncated Gaussian distribution. One strike- and one dip-value is drawn for each secondary fault and this value is relative to the local strike and dip of the primary fault.

Figure 4.8 shows 250 secondary faults simulated around all six primary faults in the fault set

with distribution according to a default displacement weight of 1.0. In the figure to the left all secondary faults belong to the same group of simulation settings. The mean value for the strike angle is 0.0 and the standard deviation is small so all secondary faults have nearly the same relative strike angle. In the figure to the right the faults are simulated in two equally large groups of simulation settings where the only difference is that the mean strike value is 30.0 degrees.
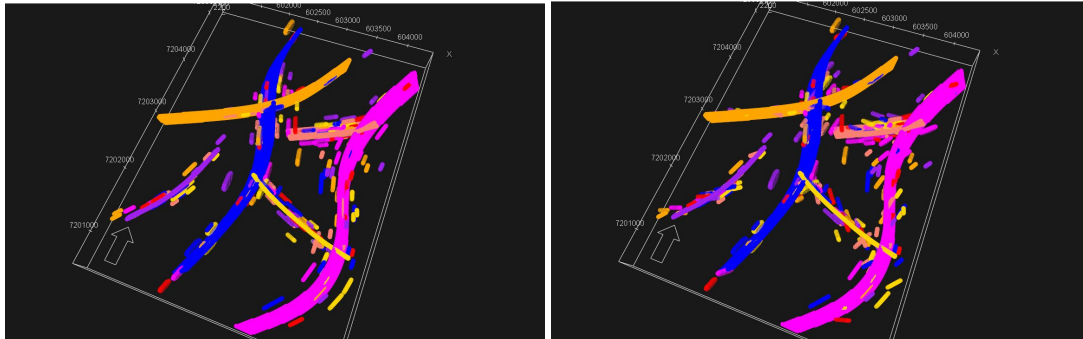
Figure 4.8. To the left 250 secondary faults distributed around six primary faults. The strike angle relative to the strike angle of the primary faults is close to zero. To the right the primary faults are simulated from two equally large groups with relative strike angle close to zero in one group and close to 30 degrees in the other group.

Figure 4.9 shows to right the intensity grid for the positioning of the primary faults from the simulations described above in one specific layer together with some of the primary faults. To the right is the standard deviation for the positioning in the normal direction of the primary fault increased by a factor of five, so the faults are spread further away from the primary faults. This is clearly visible in the intensity.
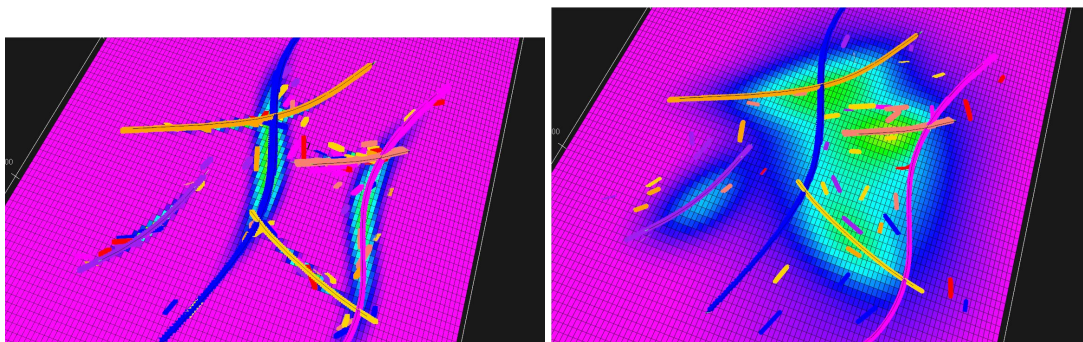
Figure 4.9. Intensity grids for two simulations where the only difference is the spread of the secondary faults position in the normal direction of the primary fault. To the left is the standard deviation 0.1 and to the right is it 0.5.

Figure 4.10 shows a simulation with 250 secondary faults all clustered around one primary fault. The position of the primary faults follows the default distribution, that is centered around the center of the primary fault. The right figure shows the intensity grid for this simulation

Figure 4.11 shows a simulation with 250 secondary faults all clustered around one primary fault. The position of the primary faults in the strike direction now has a mean of 0.5 meaning half way from the center of the primary fault to the fault tip and a standard deviation of 0.3 The right figure shows the intensity grid for this simulation
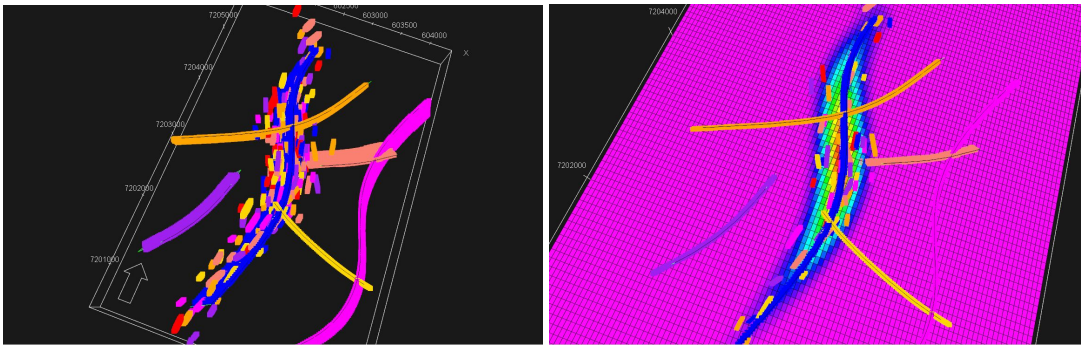
Figure 4.10. Secondary faults simulated around one primary fault. The Gaussian distribution for the position has mean zero and standard deviation 0.5 in strike and dip direction and 0.1 in the normal direction. The faults are shown to the left and together with the intensity grid to the right.
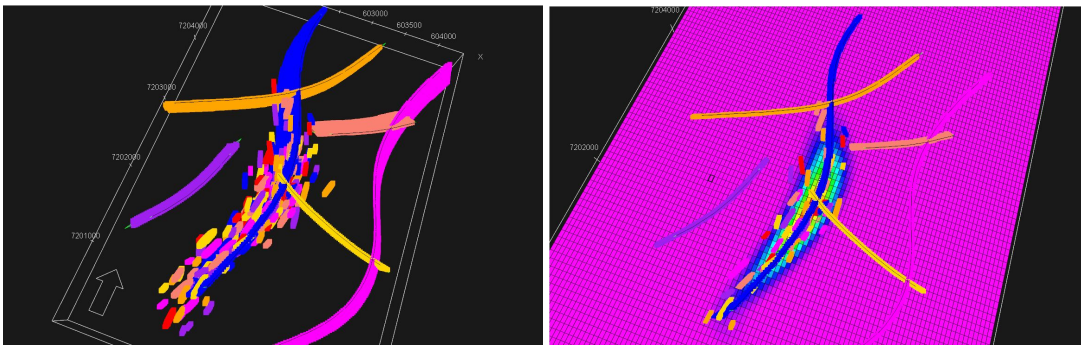
Figure 4.11. Secondary faults simulated around one primary fault. The Gaussian distribution for the position has mean 0.5 and standard deviation 0.3 in strike direction and mean zero in dip and normal direction. The standard deviation is 0.5 in dip direction and 0.1 in the normal direction. The faults are shown to the left and together with the intensity grid to the right.

4.10.4 Truncations

Adding new simulated faults to the fault set means that new truncations are introduced. The basic rule is that a secondary fault inherits all truncations that its primary fault has. So if F2 is truncated by F1, then the secondary fault F2S1 associated to F2 will also be truncated by F1. Another rule is that if two faults are on different sides of a third fault that truncates both, no truncation occurs between these two faults. In addition truncations between simulated secondary faults must be decided. For simplification, it is always the fault appearing first in the list (order of simulation) that may or may not truncate a fault later in the list, so F2S1 might truncate F2S2, but the opposite is not possible. The truncation can be decided either by a deterministic or a stochastic procedure. The deterministic rule, which is the default case, for deciding if F2S1 truncates F2S2 is as follows: The length of the intersection line between F2S1 and F2S2 is compared with the length of the extension of this line through F2S2. If the ratio between these lengths is above a user-defined limit, F2S1 is truncating F2S2, otherwise there is no truncation. The limit should be a number between 0.0 and 1.0 with values close to zero means much truncation and values close to 1 makes it difficult to truncate. This is illustrated in Figure 4.12. If the truncation is determined stochastically, the ratio computed above acts as the probability for truncation and is used in a random drawing.

4.10.5 Output

The modified fault set with all the simulated faults are written to a directory on internal HAVANA format, in addition the fault surfaces can be written as RMS point sets. To better organize the output a fault name prefix for the simulated faults can be specified.
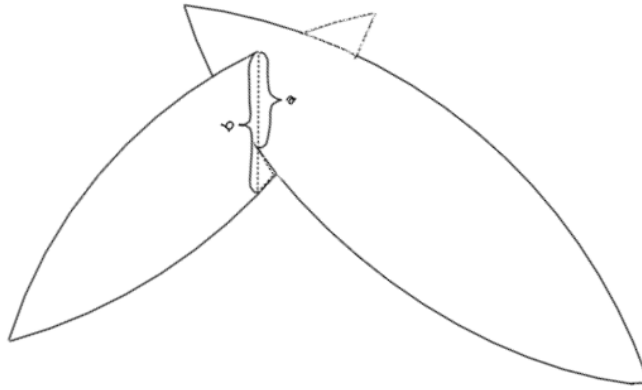
Figure 4.12. Relative intersection determining the truncation of faults.

For QC purposes intensity grids from the simulation can be written. These are 3D grids giving the relative intensity for the position of the center points of the simulated elliptic faults. The user can specify that relative intensity grids are written to file for individual primary faults. It is also possible to write a relative intensity grid which takes into account all the primary faults at once. The size of the output grids (all having the same size) must be specified. The output grid is bounded by the simulation volume definition and the output is in storm binary 3D grid format. By importing the relative intensity grid into RMS and resample it into the 3D modelling grids, it is easy to do a quality check on the spatial distribution for secondary faults. The relative intensity grid produced when accounting for all primary faults can alternatively be used as an input intensity trend in the action `<simulate-faults>` described in Section 4.11.

A summary of the simulated faults can be produced to give a snapshot of the most important data defining the faults. In this list the length of the faults can either be the length of the whole ellipse including the parts that are truncated away by other faults, or the length of the untruncated part of the fault. The latter option is the default. This list might also contain the number and names of the faults that truncate this one. Default is that this option is turned of.

4.11 SimulateFaults

This action is also used to simulate and add new faults to the fault set, but these faults are distributed in the simulation volume according to a global intensity trend. This action can be combined with SimulateSecondaryFaults to add different types of simulated faults. The description of how to define the simulation volume and the output is very similar to the one in Section 4.10 above. The parameters used to draw the displacement and thereby the length of the simulated faults are the same as for simulation of secondary faults and truncations between simulated faults and between simulated and the original faults are decided in the same manner as between the simulated secondary faults described above.

4.11.1 Position intensity

The position of the center point for the simulated faults follows a global relative trend. This can either by a constant value, meaning that the faults are uniformly distributed around the simulation volume or a 2D trend map. This trend defines a probability distribution for the drawing of the xy-position for new faults. In this way certain areas can be defined to have higher concentration of faults than others. In the present version of HAVANA, there is no intensity in depth, so all z-values are equally likely. The input trend is sampled into a 2D-grid that is used in the drawing of the position based on the relative intensity. The resolution of this grid can be specified by the

user. To add more details into the trend, a finer grid is needed. Default grid resolution is 50×50, but it is recommended to set a resolution at least as fine as the resolution of the input 2D map to capture the input trend details.

Figure 4.13 shows 250 faults simulated according to a uniform trend for position and a global intensity trend without any repulsion.
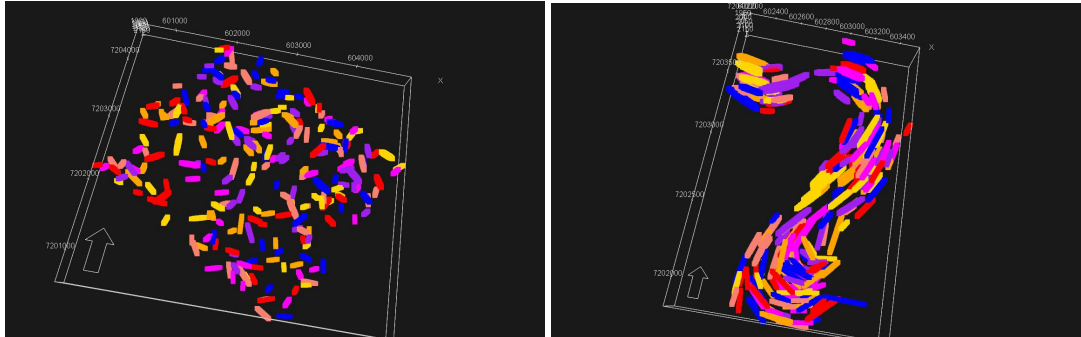
Figure 4.13. To the left, 250 simulated faults without any trend, to the right, the same number of faults following an intensity trend. Both cases have the same parameters and no repulsion.

4.11.2 Orientation

The orientation of the faults is decided by the strike and the dip that are drawn from user-specified truncated Gaussian distribution. The mean for both strike and dip can be a constant value or a 2D trend. To ensure that the orientation remains within certain limits both minimum and maximum for strike and dip can be given. In the case where 2D trend maps are used for strike and dip, be sure that minimum and maximum either is not used or is consistent with the input trend. The default is a constant mean of zero, standard deviation of 10 degrees and no truncation limits. The user can specify a probability for the drawn fault to be normal. The default is that all faults are normal. This means that to get any reverse faults a value less than 1.0 must be given for this probability. Several groups of orientation can be specified together with a fraction of faults using this specific group of orientation parameters. An example of the use of two different groups is for instance the case where we want to turn the strike 180 degrees so that the dip is the other way.

4.11.3 Interaction between faults

The faults may interact with each other meaning that they are not drawn as independent samples. The interaction is defined is through a repulsion function consisting of two parameters; a range (R) and a potential (γ). Two faults do repulse each other when the distance between their center points is shorter than the range. The potential states the strength of the repulsion. To obtain the desired interaction pattern, the user-specified number of simulated faults N are drawn independently of each other. Then a MCMc (Monte Carlo Markov chain) simulation is performed. This is an iterative algorithm that eventually converges to the desired joint distribution for the set of all faults. The number of iterations can be specified by the user, and has a default value of 1000.

The distance function d_{ij} between the two faults f_i and f_j is defined as

$$d_{ij} = \frac{\|P_i - P_j\|}{1 + \sin(\theta_{ij})} \tag{4.1}$$

where $\|P_i - P_j\|$ is the euclidean distance between the center points for faults f_i and f_j and θ_{ij} is the angle between the two faults' normal vectors. This means that for parallel faults, $d_i j$ is equal to the distance between the center points, and it decreases to half the distance as the faults become more perpendicular to each other. A repulsion function β_{ij} between the two faults with $d_{ij} < R$

is then defined as

$$\beta_{ij} = \gamma(\frac{R - d_{ij}}{R})^2.$$

(4.2)

If $d_{ij} \geq R$ the repulsion is 0. The MCMc-algorithm is based on drawing an index k uniformly between 1 and N for each iteration. Then a new fault f_k^* is drawn as a candidate to replace the present fault f_k^0. The probability for replacing f_k^0 by f_k^* is

$$\alpha_k = \max(1, \exp\{\sum_{j \neq k}(\beta_{jk}^* - \beta_{jk}^0)\})$$

(4.3)

where β_{jk}^* is the repulsion function between faults f_k^* and f_j and β_{jk}^0 is the repulsion function between faults f_k^0 and f_j.

An effect of this algorithm is that faults that are parallel repulse each other less than if there is a higher angle between them, so faults simulated with repulsion is more likely to be parallel than independently simulated faults.

4.12 GroupFaults

This action takes the set of all faults and splits it into two groups based on certain criteria. The main purpose for doing this is to define which faults that are small and should be applied for updating the permeability in an existing grid (by the action `<update-permeability>`, see Section 4.15)and which faults that should be used to update the whole grid (by the action `<update-grid>`, see Section 4.14).

4.12.1 Grouping criteria

The way the action works is that a tag is added to the fault identificator in the Havana internal format files, so the group identity is recognized by later HAVANA actions. The groups are identified by their name, one name for large faults and one name for small faults. The grouping can be specified to be by fault set or by size or both. Let the original imported faults from RMS be the "original fault set" and the fault set after simulating sub-seismic faults are added be the "modified fault set". In this action we can tag faults in the "original fault set" to be large faults by using the command `<group-by-faultset>`. If we do not use `<group-by-faultset>`, the original faults will be treated in the same way as all new simulated faults in the "modified fault set".

There are several different size criteria for grouping faults into the two groups, the group for the large faults and the group for the small faults. The criteria are either fault length, fault area or maximum displacement. When applying one of these criteria it might be that the number of faults in the group of large faults becomes too large to be practical for grid updating. Therefore there exists a third criteria. The user can specify that only a maximum number of faults are allowed to be in the group of large faults. The rest will be grouped into the group of small faults.

To summarize: First (if specified) define the original imported faults to be in the group of large faults, then select all the simulated faults that satisfy the criteria to be in the group of large faults, but if this number of faults exceed a user-defined maximum value N, only the N largest faults are included in the group of large faults.

4.13 UpdatePoints

In all the previously described actions such as `<modify-fault-surface>`, `<modify-displacement>`, `<simulate-fault-surface>`, `<simulate-displacement>`, `<set-displacement>`, `<add-faults>` and `<remove-faults>` the faults are modified. Actually the faults are regarded as 3D operators that move points in space and the mentioned actions modify the fault operators, but does not change throw, add or remove throw for any horizons or 3D grids. The action `<update-points>` actually

apply the changes of the faults to point sets for horizons or fault lines or any point set within the modelling volume. The `<update-points>` action takes as input two fault sets, one representing the original faults and one representing the modified faults. The modifications defined by the differences between these two fault sets represent the changes to be applied to the input horizon point sets (or any point set) and fault lines.

If fault block id's are given in the input point set, the updated point set will be tagged with smart-fault-tags. A smart-fault-tags is the fault block definition of the relevant block, i.e. for each relevant fault, the name of the fault and which side of this fault the fault block is located.

4.13.1 Moving horizon points

A horizon is given by a set of points. To move the horizon means moving each of the points sequentially. The points are moved by first reversing the fault operations by moving the point back to original (pre-faulted) position for all faults in order from the youngest to the oldest.

The reverse fault operation is simplified to only finding the inverse displacement function on the fault surface, defined such that it gives the original y-position for the faulted point $(x, y, F(x, y))$ and taken into account on which side of the fault the point is. This is in contrast to defining a full inverse 3D-operator.

Then the point is moved to its faulted position by applying the displacement function defined by Formulas 3.4 - 3.6 for all faults on the point in order from the oldest to the youngest. The effect of updating horizon points after modifying the displacement is shown in Figure 4.14.
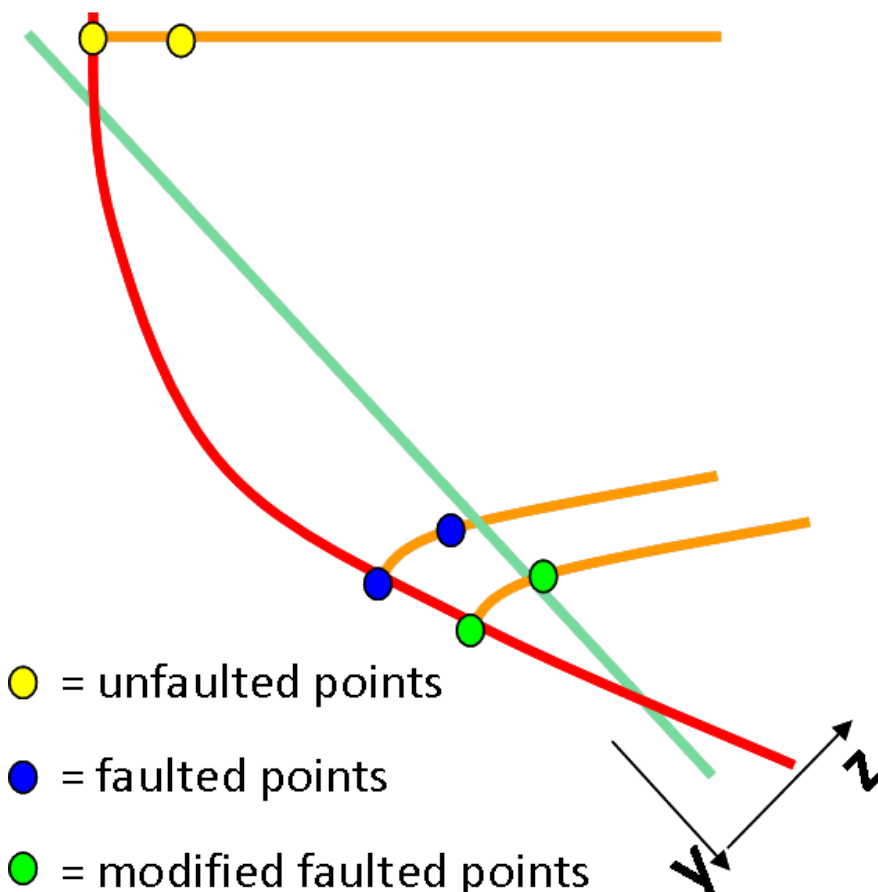
Figure 4.14. Illustrating how two points on a horizon are updated by first unfaulting the faulted point by a backward operator and then faulting it again with the modified forward operator.

4.13.2 Calculating new fault lines

Fault lines are also sets of points, and these sets can be moved along the description above. However, fault lines are defined to be located on the fault surface, and a straightforward moving of the point does not ensure that it stays on the surface for secondary faults that lie inside the volume affected by the fault displacement operator of the changed fault. Therefore a more complex algorithm is needed. A point on such a fault line is first moved as above. The moved point is then projected down on the fault surface to the point normally on the reference plane, but with the same z level as the moved point.

4.13.3 Filtering of points

The `<update-points>` action can optionally filter both horizon points and fault line points that either swithces fault block, or are sufficiently near a fault so that the interpretation of fault side in Havana might conflict with the interpretation of fault side in RMS.

For all fault line points and horizon points the side of the relevant fault can be found before and after modification, and the filtering is done by comparing these sides and removing those points that have changed side by the modification.

The RMS export of the structural model will contain information about whether horizon points are active or inactive. Inactive means that they are prolonged over the fault, but not affected by the fault operator. This means that they overlap the active points that have been faulted. The same export also contains information about fault line points that are above an erosion surface, and consequently have been air-interpreted. These points will not be used in the calculation of the displacement fields. It is possible for the user to choose if inactive horizon points and eroded fault line points are filtered out or not in the updating.

4.13.4 Updating fault block ids

When the input point set is tagged with fault block id's, these will be updated according to the changes between the original and modified fault set. If new faults are added the relevant fault blocks will be divided, and if faults are removed, relevant fault blocks will be merged. The points in the point set will be given updated fault block ids.

In some cases, points located close to a fault may get a wrong fault block after the fault have been given an add throw. This is relevant in the extension of the fault. If this is identified, the block id of these points will be updated.

The output point set will be tagged with smart-fault-tags according to updated fault block ids.

4.14 UpdateGrid

This action reads the original fault set created by the import action `<import-rms-fault-data>` and a modified fault set created by one of the actions `<add-faults>`, `<modify-displacement>` and `<remove-faults>`. The `<update-grid>` action also reads an input eclipse grid. This grid must be built from the same structural model that is read by the `<import-rms-fault-data>` action and must contain fault block information. The consistency between the structural model and the input grid is required to be able to handle fault block information correctly and to be able to modify the displacement of existing faults in the grid. Note however that only faults defined with displacement along grid pillar lines (coord lines) can be updated, removed or inserted. Stair case faults cannot be modified, removed or inserted by this version of HAVANA.

If new faults are added in the modified fault set by the `<add-faults>` action, these faults are identified and the eclipse grid is updated according to these. The fault trace is found, having fault surfaces defined by grid pillar lines, the relevant fault blocks are divided and the fault displace-

ment is applied to the grid.

If the modified fault set contains new or modified displacement for existing faults, the `<update-grid>` action will modify the faults displacement in the grid.

If the modified fault set has fewer faults than the original fault set because the action `<remove-faults>` is used, the faults will also be removed from the grid. Note however that only faults that are not truncating or are truncated by other faults can be removed completely from the grid. All other faults that are removed will still contain a minor displacement and will still be present in the FAULTS keyword in the Eclipse files.

The action `<update-grid>` returns a modified eclipse grid. A group name can be given if the action `<group-faults>` has been run. Then only the faults matching the group name are used to update the grid.

4.14.1 Find fault trace

The new faults are identified by comparing the original and the modified fault set.

The new faults could potentially be small, hence in order to ensure that the fault trace reaches across the fault blocks to be divided by the fault, the fault surface is extended to cross the entire grid. The fault trace is identified on a pillar level; hence the pillars located closest to the fault surface at the fault surface reference z coordinate, defines the fault trace along the extended fault surface. The truncation rules from the modified fault set is applied to the fault trace.

The segments along the fault surface where the fault displacement field is active is added to the eclipse grid.

4.14.2 Divide blocks

The blocks to be divided by the new faults are identified according to the fault displacement and the truncation rules. New block id/name is given to the part of the block located on the hanging wall (HW) side of a new fault. For each of the columns in the blocks to be divided, the respective fault side is identified; hence the fault surface of the new faults will be on a pillar level.

4.14.3 Add displacement

Each corner point of the cells within the blocks to be divided is moved according to the fault displacement and the respective column's fault side. The corner points are moved in the z direction only. The neighbour inactive cells are moved according to the fault displacement in order to avoid unwanted boundary effects.

4.15 UpdatePermeability

To account for larger faults that are added to the fault set in an Eclipse grid, the action `<update-grid>` (see 4.14) should be run. However, if faults are small, typically smaller than the Eclipse grid blocks, they may not be visible in the updated grid. Then to account for their impact, the 3D permeability parameter for the existing grid should be updated. This action does exactly that. The action `<group-faults>` (see 4.12) should be run prior to this action to identify which faults that are active when the permeability is updated. The input to this action is a fault set tagged with group names and an Eclipse grid with permeability values that is going to be updated. Each cell of the input grid can be subdivided into a user-defined refinement where the permeability impact of the fault is calculated. The permeability on the fault surfaces is either determined by a deterministic value or by a probability distribution. The user can specify if the permeability value for the fault surface is absolute or relative to the rock volume surrounding the fault. By default the permeability is updated by the fault displacement and the fault surface effect. Any one of

these can be omitted by the choice of the user, but not both. The output grid for the calculated permeability values is by default of the same dimension as the input grid, but might be different.

4.15.1 Updating algorithm

When the permeability is updated due to the impact of new faults, the following algorithm is used:

1. For each grid cell in the input grid there are data values for permeability in the x-, y- and z-direction. These values are stored at the center of the grid-cell, or if a subdivision is chosen, at the cell center of the sub-grid with the values taken from the main grid.

2. The fault surface effect is calculated if desired by modifying all sub-cells crossed by a fault with the drawn fault permeability. Here the fault thickness plays a role. This can be drawn from a user-specified distribution and adjusted by the maximum displacement (not default).

3. The displacement effect of the faults is included if desired, by moving the data points according to the displacement operator.

4. The output grid is read if this is not the same as the input grid. Each cell is subdivided, if desired. Note that for best results, these sub-cells should be of approximately the same size as the sub-blocks in the input grid.

5. The data points are sampled into the sub-blocks (or full blocks) of the output grid and the upscaling of the three permeability values is performed by averaging first within the sub-cells and then over all sub-cells in the grid cell. Two averaging methods are available: Harmonic-arithmetic (which is the default) meaning that the average in the sub-cell is harmonic and then an arithmetic average is taken over the calculated values for each sub-cell, and arithmetic-harmonic which does the opposite.

4.16 CreateFaultGrid

The fault zone grid is created based on the Eclipse fault trace. In addition the grid cells have to have a FaultBlock attribute saying which block the different cells belong to.

The fault zone grid is then created according to the following algorithm:

1. The top and bottom surfaces for each block is obtained from the grid.

2. All the top and bottom surfaces for the respective grid blocks are extrapolated according to the refinement distance given.

3. A local grid is generated for each fault segment in the grid. The top and bottom of the local grid is obtained by taking the most extreme value from the extrapolated grid top and bottom surfaces for each grid pillar.

4. The rest of the layers are generated by splitting the pillars evenly. The number of grid layers in the local refinement depends on the total number of layers in the grid.

We make sure that each column in the grid only is refined once. The algorithm also ensures that the grid layering is continous within the local grid refinement for a single fault.

The local grid is then made by making a grid only containing the local grid refinements along the fault traces.

4.17 MergeFaultGrid

The merging of the local grid and global grid is pretty straight forward. The different parts of the local grid are added to the coarse grid as local grid refinements. The only issue is to map the local

grid cells to the corresponding grid cells in the global grid. This is done by comparing the location of the pillars in the local grid and the global grid.

4.18 CalculateFaultSeal

This action is used to model the fault sealing effect due to clay smear along the fault surface. The sealing effect is represented as transmissibility multipliers, given as input to fluid simulators (e.g. Eclipse). The action is calculating the transmissibility multipliers of the faults based on estimated fault displacement, shale gauge ratio (SGR), fault permeability and fault thickness (later refereed to as fault seal parameters).

Required input data is an RMS structural model of the faults and a corresponding Eclipse grid that is basis for the fault grid. The grid must contain the fault trace data (Eclipse FAULTS keyword), fault block identifiers given by a grid parameter named FAULTBLOCK, absolute permeabilities (e.g. PERMX, PERMY, and PERMZ) and volume of clay/shale (e.g. VShale) to estimate the amount of clay smear along the fault.

The calculations are performed on the fault surface that is described in a coordinate system defined by the 2D reference plane of the fault. The grid cell faces of the fault trace is mapped onto the fault surface and also described in the same 2D reference plane. Figure 4.15 illustrates how a cell face in the Eclipse grid is mapped to the fault surface and described by the 2D reference plane.
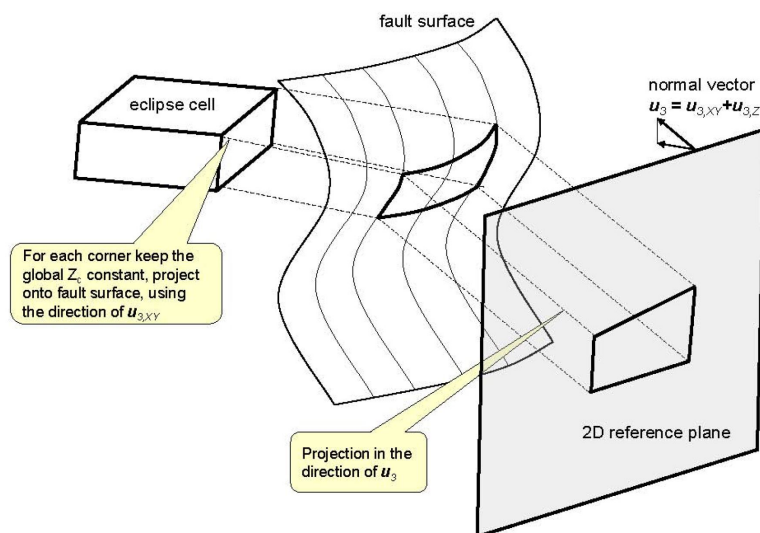
Figure 4.15. Mapping of Eclipse cell face onto the fault surface that is described in the coordinate system defined by the 2D reference plane.

The mutual interface area between pair of mapped grid cell faces across the fault surface is identified (i.e. between grid cells on the hanging wall and foot wall side that has overlapping cell faces). The fault seal parameters are then calculated at the center point of each mutual interface. Properties of the grid cells like e.g. permeability and shaliness that are required when computing SGR and transmissibility, are obtained from the pair of cells contributing to the interface area. The transmissibility multipliers defined as the ratio between transmissibility with and without fault, are computed between the same pair of cells using the estimated mutual interface area, fault thickness and fault permeability. The transmissibilitiy multipliers are finally exported to an Eclipse grid using the MULTIPLY and EDITNNC keywords depending on whether the pair of grid cells are logically nearest neighbours (MULTIPLY) or not (EDITNNC). The Eclipse grid is written to file in .grdecl format.

The permeability along the fault surface is modelled according to Manzocchi or Sperrevik depending on the XML command `<fault-permeability-method>`. The Manzocchi permeability relation [13] is given by

$$\log k_f = -4\text{SGR} - \frac{1}{4}\log(D)(1-\text{SGR})^5 \tag{4.4}$$

where k_f denotes the fault permeability, D is the fault displacement and SGR is the shale gouge ratio.

The Sperrevik permeability relation [14] is given by

$$k_f = a_1 \exp\{-[a_2\text{SGR} + a_3 Z_\text{max} + (a_4 Z_f - a_5)(1-\text{SGR})^7]\} \tag{4.5}$$

where Z_max and Z_f is the maximum burial depth of the faulted sediments and the burial depth at the time of deformation, respectively. The parameters a_1 to a_5 are empirical constants.

The transmissibility T_i of cell i is defined according to Ponting [12] where

$$T_i = K_i \frac{|\mathbf{A} \cdot \mathbf{d}_i|}{||\mathbf{d}_i||^2} \tag{4.6}$$

Here is K_i the permeability of cell i, $\mathbf{A} = [A_x, A_y, A_z]$ is the area of the mutual interface decomposed into its directional components and $\mathbf{d}_i = [d_{i,x}, d_{i,y}, d_{i,z}]$ is the vector (i.e. distance) from the cell center to the center point of the cell face which contributes to $\mathbf{A}$. Note that this cell face center is different from the center point of the mutual interface. See Figure 4.16 for details. In the Eclipse grid the permeability of the cells is directional, causing K_i in Eq. 4.6 to be replaced by $K_{i,x}$, $K_{i,y}$ or $K_{i,z}$ depending on the direction of the cell face. The direction is defined as the normal to the cell face surface. The transmissibility between neighboring cells i and j is give by

$$T_{ij} = \frac{1}{T_i^{-1} + T_j^{-1}}. \tag{4.7}$$

According to Manzocchi et al. [13] the properties of the fault zone can be considered explicitly using the fault thickness $\mathbf{t}_f$ and the fault permeability k_f. Thus, the transmissibility of the faulted zone on the hanging wall (HW) or foot wall (FW) side can be expressed as

$$T_f = k_f \frac{|\mathbf{A} \cdot \frac{1}{2}\mathbf{t}_f|}{||\frac{1}{2}\mathbf{t}_f||^2}. \tag{4.8}$$

The fault thickness $\mathbf{t}_f$ is distributed equally on the HW and FW side, therefore $\mathbf{t}_f$ is multiplied with $\frac{1}{2}$ in the above formula.

The fault thickness reduces the effective distance from the cell center to the cell face, giving a modified transmissiblity of cell i

$$T_i' = K_i \frac{|\mathbf{A} \cdot (\mathbf{d}_i - \frac{1}{2}\mathbf{t}_f|)}{||\mathbf{d}_i - \frac{1}{2}\mathbf{t}_f||^2}. \tag{4.9}$$

From Eqs. 4.8 and 4.9 we finally obtain the transmissibility taking into account the fault zone between cells i and j:

$$T_{ij}' = \frac{1}{T_i'^{-1} + T_j'^{-1} + 2T_f^{-1}} \tag{4.10}$$

The transmissibility multipliers are then given by the ratio T_{ij}'/T_{ij}.
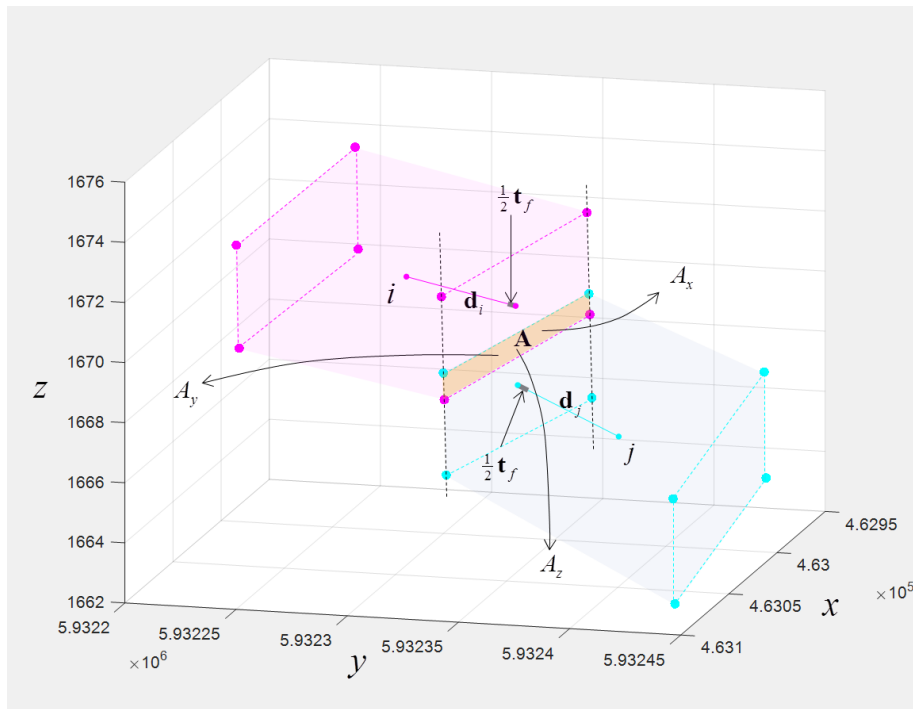
Figure 4.16. Schematic illustration of two cell faces and their mutual interface area $\mathbf{A}$ decomposed into A_x in the yz-plane, A_y in the xz-plane and A_z in the xy-plane. Also shown are the directional vectors $\mathbf{d}$ and fault thickness vectors $\mathbf{t}_f$ of cell i and j on the HW and FW side of the fault, respectively.

4.19 FaultStatistics

This action is used for calculating statistics of a fault set. The fault set can be given as input either from a Havana internal format (written with `<output-rms-faults-points-directory>` from a previous action), or from a structural model (created with the `<import-rms-fault-data>` action). There are two types of output: general statistics of the fault set and displacement profiles.

4.19.1 General statistics

The general statistics can be calculated and written to a file by providing the `<output-statistics-file>` keyword.

The statistics calculated for each fault in a fault set are:

- **Length**
- **Untruncated Length**
- **Height**
- **Max displacement**
- **Strike direction** Measured from N-S, clockwise.
- **Dip direction**
- **Fault Type**

Fault Type is an indicator of whether the fault is a normal or reverse fault. It can also be undefined, for instance in the case of a scissor fault. If a structural model is provided instead of a fault set, then statistics for each horizon are calculated as well. The exception for horizons is height, which will have the value -1.

An example of the fault statistics file is shown in A.6.

4.19.2 Displacement profiles

There are two types of displacement profiles created in Havana. The first is displacement profiles along each fault. These profiles are calculated along a line on the fault surface. The line is chosen as a horizontal line from the reference plane such that one of the points corresponds with the maximum displacement of the fault. If the structural model is given as input the displacement profiles will also be calculated along each horizon.

The second profile is to calculate the cumulative displacement along an arbitrary given line. Here a line (start point and end point) will be given as input and total displacement of the fault set will be calculated along this line. The displacement is calculated by stepping through the input line. For each step an infinite line normal to the input line is created and the displacement for all faults it intersects is summed up. The resulting displacement profile will be written to file.

See A.7 for examples of these output files.

4.19.3 Intensity maps

The fault statistics action produces intensity maps of an input fault set. There are two types of maps, fault intensity and displacement intensity. For both maps a surface is created, and for each point on the surface the number of faults is counted or displacement is added up based within an input range. These maps are created based on the horizons in input fault set. The output maps can also be smoothed with a Gaussian smoother. See figure 4.17 for an example.
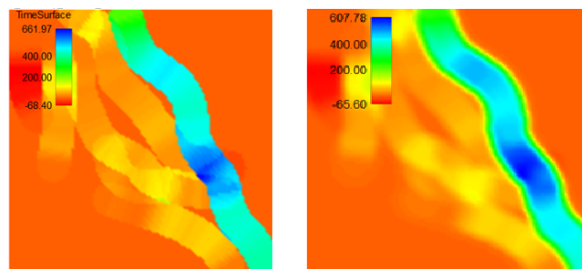
Figure 4.17. Intensity maps with displacement for Emerald. Left is without smoothing and right is with smoothing.

5 Main workflows

This chapter contains some of the main workflows that will typically be run by HAVANA. A lot of other workflows are available, and the user reference in Chapter 6 will give all the detailed description that is needed to set up alternative workflows. Appendix B.1 shows an example on how a workflow in RMS can be set up based on different HAVANA workflows.

5.1 Modify fault surfaces and update the structural model in RMS

There are two possible ways to perform this workflow. Alternative 1 consists of updating the fault surfaces by HAVANA and let RMS update the structural model based on extracted point sets marked with side of fault. Alternative 2 lets HAVANA update the point sets by the action UpdatePoints.

Alternative 1

1. Export the structural model from RMS by the export to HAVANA option in the horizon model.

2. Run HAVANA action ImportRmsFaultData and produce an original fault set on internal HAVANA format.

3. Run HAVANA action ModifyFaultSurface with the original fault set as input and produce a modified fault set.

4. Import the updated fault surfaces to RMS.

5. Update the RMS structural model with extracted point sets marked with side of fault.

Alternative 2

1. Export the structural model from RMS by the export to HAVANA option in the horizon model.

2. Run HAVANA action ImportRmsFaultData and produce an original fault set on internal HAVANA format.

3. Run HAVANA action ModifyFaultSurface with the original fault set as input and produce a modified fault set.

4. Run HAVANA action UpdatePoints with original and modified fault sets as input and produce updated horizons and faultlines.

5. Import the updated fault surfaces, horizons and fault lines to RMS.

6. Update the RMS structural model.

5.2 Simulate fault surfaces within uncertainty envelopes and update structural model in RMS

As in Section 5.1 above there are two possible workflows here, with or without the update of the point sets performed by HAVANA.

Alternative 1

1. Export the structural model from RMS by the export to HAVANA option in the horizon model.

2. Run HAVANA action ImportRmsFaultData and produce an original fault set on internal HAVANA format.

3. Run HAVANA action FaultUncertaintyModel to set up uncertainty envelopes around faults.

4. Run HAVANA action SimulateFaultSurface with the original fault set and the uncertainty envelopes as input to produce realizations from the uncertainty model for the faults, possibly conditioned to well picks and well path points.

5. Import the updated fault surfaces to RMS.

6. Update the RMS structural model with extracted point sets marked with side of fault.

Alternative 2

1. Export the structural model from RMS by the export to HAVANA option in the horizon model.

2. Run HAVANA action ImportRmsFaultData and produce an original fault set on internal HAVANA format.

3. Run HAVANA action FaultUncertaintyModel to set up uncertainty envelopes around faults.

4. Run HAVANA action SimulateFaultSurface with the original fault set and the uncertainty envelopes as input to produce realizations from the uncertainty model for the faults, possibly conditioned to well picks and well path points.

5. Run HAVANA action UpdatePoints with original and modified fault sets as input and produce updated horizons and faultlines.

6. Import the updated fault surfaces, horizons and fault lines to RMS.

7. Update the RMS structural model.

5.3 Modify throw and update structural model in RMS

1. Export the structural model from RMS by the export to HAVANA option in the horizon model.

2. Run HAVANA action ImportRmsFaultData and produce an original fault set on internal HAVANA format.

3. Run HAVANA action ModifyDisplacement with the original fault set as input to produce a modified fault set.

4. Run HAVANA action UpdatePoints with original and modified fault sets as input and produce updated horizons and fault lines.

5. Import the updated horizons and fault lines to RMS.

6. Update the RMS structural model.

5.4 Modify throw and update 3D simulation grid

1. Export the structural model from RMS by the export to HAVANA option in the horizon model.

2. Export a grid on Eclipse format from RMS with fault block indices.

3. Run HAVANA action ImportRmsFaultData and produce an original fault set on internal HAVANA format.

4. Run HAVANA action ModifyDisplacement with the original fault set as input to produce a modified fault set.

5. Run HAVANA action UpdateGrid with original and modified fault set as input and produce a grid updated by the displacement modifications.

6. Import the updated grid to the RMS grid model.

5.5 Add new elliptic faults and update structural model in RMS

1. Export the structural model from RMS by the export to HAVANA option in the horizon model.

2. Run HAVANA action ImportRmsFaultData and produce an original fault set on internal HAVANA format.

3. Run HAVANA action AddFaults to add elliptic faults with user-defined displacement and produce a modified fault set.

4. Run HAVANA action UpdatePoints with original and modified fault sets as input and produce updated horizons and fault lines.

5. Import the the new fault surfaces and updated horizons and fault lines to RMS.

6. Update the RMS structural model.

5.6 Add new interpreted faults and update structural model in RMS

1. Make two initial structural models A and B in RMS. These are identical, except that new faults are added to model B.

2. Export the two initial structural models A and B from RMS by the export to HAVANA option in the horizon model.

3. Run HAVANA action ImportRmsFaultData on both models and produce two original fault sets A and B on internal HAVANA format.

4. Run HAVANA action SetDisplacement on original fault set B to set correct displacement fields to the new faults. This fault set is modified fault set B on output.

5. Run HAVANA action UpdatePoints with original fault set A and modified fault set B as input and produce updated horizons and fault lines.

6. Import the updated horizons and fault lines to RMS.

7. Update the RMS structural model.

5.7 Add new elliptic faults and update 3D simulation grid

1. Export the structural model from RMS by the export to HAVANA option in the horizon model.

2. Export a grid on Eclipse format from RMS with fault block indices.

3. Run HAVANA action ImportRmsFaultData and produce an original fault set on internal HAVANA format.

4. Run HAVANA action AddFaults to add elliptic faults with user-defined displacement and produce a modified fault set.

5. Run HAVANA action UpdateGrid with original and modified fault set as input and produce a grid updated with the new faults.

6. Import the updated grid to the RMS grid model.

5.8 Add new interpreted faults and update 3D simulation grid

1. Make two initial structural models A and B in RMS. These are identical, except that new faults are added to model B.

2. Export the two initial models A and B to grids on Eclipse format from RMS.

3. Run HAVANA action ImportRmsFaultData on both models and produce two original fault sets A and B on internal HAVANA format.

4. Run HAVANA action SetDisplacement on original fault set B to set correct displacement fields to the new faults. This fault set is modified fault set B on output.

5. Run HAVANA action UpdateGrid with original fault set A and modified fault set B as input and produce a grid updated with the new faults.

6. Import the updated grid to the RMS grid model.

5.9 Remove faults and update structural model in RMS

1. Export the structural model from RMS by the export to HAVANA option in the horizon model.

2. Export a grid on Eclipse format from RMS.

3. Run HAVANA action ImportRmsFaultData and produce an original fault set on internal HAVANA format.

4. Run HAVANA action RemoveFaults to remove faults and produce a modified fault set.

5. Run HAVANA action UpdatePoints with original and modified fault set as input and produce updated horizons and fault lines.

6. Import the updated horizons and fault lines to RMS.

7. Update the RMS structural model.

5.10 Remove fault and update 3D simulation grid

1. Export the structural model from RMS by the export to HAVANA option in the horizon model.

2. Export a grid on Eclipse format from RMS.

3. Run HAVANA action ImportRmsFaultData and produce an original fault set on internal HAVANA format.

4. Run HAVANA action RemoveFaults to remove faults and produce a modified fault set.

5. Run HAVANA action UpdateGrid with original and modified fault set as input and produce a grid updated without the removed faults.

6. Import the updated grid to the RMS grid model.

5.11 Simulate new sub-seismic faults following global intensity trend and update permeabilities in existing grid and update grid geometry

1. Export the structural model from RMS by the export to HAVANA option in the horizon model.

2. Export a grid on Eclipse format from RMS. This grid contains fault block indices and permeability values.

3. Run HAVANA action ImportRmsFaultData and produce an original fault set on internal HAVANA format.

4. Run HAVANA action SimulateFaults to produce a modified fault set including simulated faults.

5. Run HAVANA action GroupFaults to split the modified fault set in two groups of large and small faults and give group names to the modified fault set.

6. Run HAVANA action UpdatePermeabilities on the modified faults belonging to the group of small faults to update the permeability values in the Eclipse grid.

7. Run HAVANA action UpdateGrid on the modified faults belonging to the group of large faults to update the Eclipse grid.

8. Import the updated grid to the RMS grid model.

5.12 Simulate new sub-seismic faults within damage zone around existing faults and update permeabilities in existing grid and update grid geometry

1. Export the structural model from RMS by the export to HAVANA option in the horizon model.

2. Export a grid on Eclipse format from RMS. This grid contains fault block indices and permeability values.

3. Run HAVANA action ImportRmsFaultData and produce an original fault set on internal HAVANA format.

4. Run HAVANA action SimulateSecondaryFaults to produce a modified fault set including simulated secondary faults around the primary faults in the original fault set.

5. Run HAVANA action GroupFaults to split the modified fault set in two groups of large and small faults and give group names to the modified fault set.

6. Run HAVANA action UpdatePermeabilities on the modified faults belonging to the group of small faults to update the permeability values in the Eclipse grid.

7. Run HAVANA action UpdateGrid on the modified faults belonging to the group of large faults to update the Eclipse grid.

8. Import the updated grid to the RMS grid model.

5.13 Fault zone grids

Support for fault zone grids was originally implemented as part of the FaultFacies project in cooperation with CIPR at the University of Bergen.

The idea is to better capture the fluid flow effect of the fault by modeling the faults as 3D volumes, instead of just as 2D surfaces populated with transmissibility multipliers. To be able to do this a grid representation of the fault zone must be generated. This is done by the **CreateFaultGrid**, action, while the **MergeFaultGrid** action is used to merge the fault zone grid into the general background grid after the modelling of facies and petrophysical properties within the faultzone is finished.

A simplified workflow is:

1. The location of the fault zone must be found or given.

2. The fault zone area is split into a separate grid, where the grid resolution is increased, and with continous top and bottom surfaces across the faults.

3. Fault facies are modeled inside the local grid.
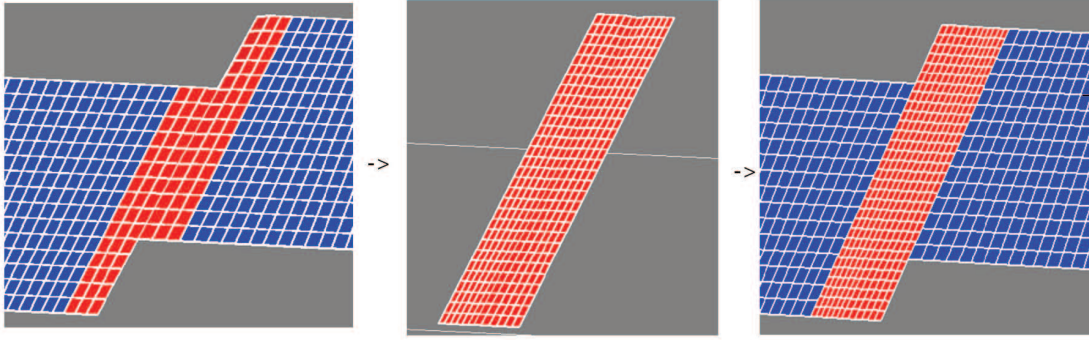
Figure 5.1. Local grid overview. The first figure shows the original coarse grid with the desired fault zone in red. The second figure show the refined local grid with continous top and bottom surfaces. The third figure shows the local grid merged into the coarse grid.

4. The local grid is merged together with the coarse grid.

6 User Reference

This chapter contains a list of all possible commands that can be given in a HAVANA modelfile.

Before the first action comes a section with some necessary commands pertaining to the whole program, namely `<project-settings>`. The allowable commands in each of the following sections depend on the specific action command.

Each action is independent of the others in that it reads all of its input from files, and writes all results to files before the next action is started. This is slightly inefficient, but has the advantage that HAVANA may be restarted at any action 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. The input values are checked whether they are legal, and the input files are checked to be existing and possible to open. If 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 the input files are correct before the program execution reaches the place where they are read.

All workflows start with the action `<import-rms-fault-data>`. All communication between the actions goes through the internal Havana format, described in Appendix A.10. The communication is handled with the commands `<input-havana-faults-directory>` and `<output-havana-faults-directory>`.

6.0.1 XML syntax

The numbering in the following sections shows the command grouping. A command with no sub-numbering expects a value to be given, otherwise, it is only a grouping of other commands. Within each grouping, the commands may appear in any order.

All commands are optional, unless otherwise stated. A necessary command under an optional is only necessary if the optional is given.

6.1 `<project-settings>` **(necessary)**

The `<project-settings>` command is necessary and should be specified prior to any HAVANA action. `<project-settings>` should only be specified one time in each model file, even if the model file consist of several actions.

Example:
```
<project-settings>
    <log-settings>
        <screen-level> 5 </screen-level>
        <log-file>
            <file> logfile.dat </file>
            <level>    1 </level>
        </log-file>
    </log-settings>
    <io-settings>
        <input-directory> . </input-directory>
        <output-directory> . </output-directory>
    </io-settings>
</project-settings>
```

6.1.1 `<log-settings>` **(necessary)**
Description: Controls the logging during the run.
Argument: Commands specifying the logging to screen and file.
Repeatable: No

6.1.1.1 `<screen-level>`
Description: The command regulates the amount of information output to screen. There are six levels of information: 0, 1, 2, 3, 4 and 5. If level 0 is used, only warnings and errors are written. Level 4 and 5 give debug information.
Argument: One. An integer 0, 1, 2, 3, 4 or 5.
Default: 1
Repeatable: No

6.1.1.2 `<log-file>` **(necessary)**
Description: Controls the logging to file during the run.
Argument: Commands specifying the logging to file.
Repeatable: No

6.1.1.2.1 `<file>` **(necessary)**
Description: Specifies the file for logging of information from the run.
Argument: One. The name of the log-file.
Default:
Repeatable: No

6.1.1.2.2 `<level>`
Description: The command regulates the amount of information output to the log file. There are six levels of information: 0, 1, 2, 3, 4 and 5. If level 0 is used, only warnings and errors are

written. Level 4 and 5 give debug information.

Argument: One. An integer 0, 1, 2, 3, 4 or 5.

Default: 1

Repeatable: No

6.1.2 `<io-settings>` **(necessary)**

Description: Controls the main input and output directories of the run.

Argument: Commands specifying the input and output directories.

Repeatable: No

6.1.2.1 `<input-directory>` **(necessary)**

Description: Main input directory of the run. This path will be relative to the working directory.

Argument: One. The directory.

Default:

Repeatable: No

6.1.2.2 `<output-directory>` **(necessary)**

Description: Main output directory of the run. This path will be relative to the working directory.

Argument: One. The directory.

Default:

Repeatable: No

6.2 `<import-rms-fault-data>`

This action is used to import fault data from RMS.

The structural model can be exported from RMS. For details about this export format, see A.1. The directory containing the structural model is given by the `<input-structural-model-directory>` command. In this action the displacement fields are generated from fault lines given in the structural model.

It is optional to specify displacement settings and displacement variogram. If this is not specified, default values are used. Displacement setting parameters can be specified under command `<default-displacement-settings>` as new default values for all faults, or individually for each fault under command `<displacement-settings>`. Variogram is specified under `<displacement-variogram>` command.

If the fault data should be used in succeeding actions, the command `<output-havana-faults-directory>` is necessary. For debugging of the structural model data, the command `<debug-output-fault-info-directory>` could be specified.

Example:

```
<import-rms-fault-data>
    <input-structural-model-directory> StructuralModel </input-structural-model-directory>
    <output-havana-faults-directory> origFaultSet </output-havana-faults-directory>
    <debug-output-fault-info-directory> Directory </debug-output-fault-info-directory>
    <displacement-variogram>
        <type> Spherical </type>
        <strike-range> 1500 </strike-range>
        <dip-range> 1500 </dip-range>
        <anisotropy-angle> 0 </anisotropy-angle>
    </displacement-variogram>
    <default-displacement-settings>
        <asymmetry> 0.7 </asymmetry>
        <range> 2000 </range>
        <max-displacement-length-relation>
            <ratio> 0.01 </ratio>
            <power> 1 </power>
        </max-displacement-length-relation>
        <length-height-ratio> 2.0 </length-height-ratio>
    </default-displacement-settings>
    <displacement-settings>
        <fault-name> F1 </fault-name>
        <asymmetry> 0.7 </asymmetry>
        <range> 2000 </range>
        <max-displacement-length-relation>
            <ratio> 0.01 </ratio>
            <power> 1 </power>
        </max-displacement-length-relation>
        <length-height-ratio> 2.0 </length-height-ratio>
        <estimate-displacement> no </estimate-displacement>
    </displacement-settings>
    <displacement-settings>
```

```
            <fault-name> F2 </fault-name>
            <range> 5000 </range>
        </displacement-settings>
    </import-rms-fault-data>
```

6.2.1 `<input-structural-model-directory>` (necessary)

Description: Specifies the name of the directory containing the fault files from the RMS export.
The directory is exported from RMS2012 or newer by using the job Extract Fault Data (see
Figure 6.1). This job is enabled by the environment variable
RMS_ENABLE_HAVANA_EXPORT which must be set e.g. like this using Linux C-shell:
setenv RMS_ENABLE_HAVANA_EXPORT 1

Argument: One. The directory.

Default:

Repeatable: No
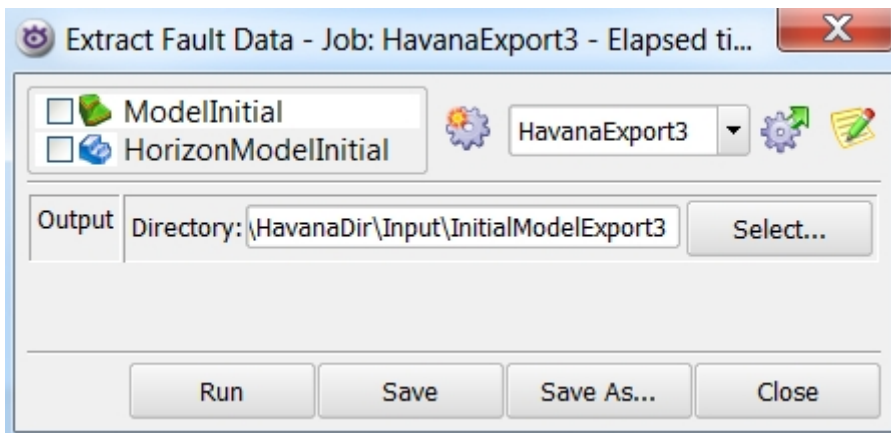
Figure 6.1. panel showing how to export the RMS structural model to files for RMS2012 and newer versions.

6.2.2 `<output-havana-faults-directory>` (necessary)

Description: Specifies the name of the directory where the original faults are written to. These
faults are written on the internal Havana format (see Appendix A.10).

Argument: One. The directory.

Default:

Repeatable: No

6.2.3 `<output-havana-structural-model-directory>`

Description: Specifies where to output the structural model with the imported RMS faults. This
structural model can then be used as input to other Havana actions.

Argument: One. The directory.

Default:

Repeatable: No

6.2.4 `<debug-output-fault-info-directory>`

Description: This command is useful in QC of the structural model and debugging of Havana. A
lot of potentially useful information is written to the given directory. See description of the
output in 6.2.8.

Argument: One. The directory.

Default:

Repeatable: No

6.2.5 `<displacement-variogram>`

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

Argument: Commands specifying the variogram.

Default:

Repeatable: No

6.2.5.1 `<type>` **(necessary)**

Description: Specifies the variogram type. Possible types are GAUSSIAN, SPHERICAL, EXPONENTIAL or GENERAL_EXPONENTIAL. For the GENERAL_EXPONENTIAL type an additional parameter `<power>` must be given.

Argument: One. The variogram type.

Default: SPHERICAL

Repeatable: No

6.2.5.2 `<strike-range>` **(necessary)**

Description: Specifies the range in strike direction. Higher range gives smoother fault surfaces.

Argument: One. A positive number.

Default: 1000

Repeatable: No

6.2.5.3 `<dip-range>` **(necessary)**

Description: Specifies the range in dip direction. Higher range gives smoother fault surface.

Argument: One. A positive number.

Default: 2000

Repeatable: No

6.2.5.4 `<anisotropy-angle>` **(necessary)**

Description: Specifies the anisotropy angle in degrees. Use this for rotating the direction of the ranges.

Argument: One. Degree in the interval 0 to 360.

Default: 0

Repeatable: No

6.2.5.5 `<power>`

Description: Specifies the power. Only relevant for general the variogram type GENERAL_EXPONENTIAL.

Argument: One. A number.

Default:

Repeatable: No

6.2.6 `<default-displacement-settings>`

Description: Specifies default values for displacement settings.

Argument: Commands specifying default values.

Repeatable: No

6.2.6.1 `<asymmetry>`

Description: Parameter giving the throw distribution, i.e. asymmetry. If asymmetry = 1.0 all throw is distributed on hanging wall side, if asymmetry = 0.0 all throw is distributed on foot wall.

Argument: One. A number in the interval 0.0 to 1.0.

Default: 0.7

Repeatable: No

6.2.6.2 `<range>`

Description: Parameter giving the range of the displacement field.

Argument: One. A positive number.

Default: 1000

Repeatable: No

6.2.6.3 `<max-displacement-length-relation>`

Description: Specifies the relation between maximum displacement and length of fault. The relation is given as $length = (maximumDisplacement/b)^{(1/a)}$, where a is the power and b is the ratio.

Argument: Commands specifying the relationship.

Repeatable: No

6.2.6.3.1 `<ratio>` **(necessary)**

Description: The ratio parameter b in the relation between maximum displacement and length of fault.

Argument: One. A positive number.

Default: 0.01

Repeatable: No

6.2.6.3.2 `<power>` **(necessary)**

Description: The power parameter a in the relation between maximum displacement and length of fault.

Argument: One. A positive number.

Default: 1.0

Repeatable: No

6.2.6.4 `<length-height-ratio>`

Description: The relation between length and height of fault, given by a constant c. The relation is given as $height = length/c$.

Argument: One. A positive number.

Default: 2.0

Repeatable: No

6.2.6.5 `<estimate-displacement>`

Description: Specifies whether the displacement will be estimated for the the fault.

Argument: One. Yes or no.

Default: Yes

Repeatable: No

6.2.7 `<displacement-settings>`
Description: Specifies displacement settings parameters for a given fault.
Argument: Commands specifying displacement settings.
Repeatable: Yes

6.2.7.1 `<fault-name>`
Description: Name of fault that the displacement settings are specified for.
Argument: One. Name of fault.
Default:
Repeatable: No

6.2.7.2 `<asymmetry>`
See 6.2.6.1.

6.2.7.3 `<range>`
See 6.2.6.2.

6.2.7.4 `<max-displacement-length-relation>`
See 6.2.6.3.

6.2.7.4.1 `<ratio>` **(necessary)**
See 6.2.6.3.1.

6.2.7.4.2 `<power>` **(necessary)**
See 6.2.6.3.2.

6.2.7.5 `<length-height-ratio>`
See 6.2.6.4.

6.2.7.6 `<estimate-displacement>`
See 6.2.6.5.

6.2.8 Specification of debug output

The following information is given in the following directories:

fault_info contains a point set file for each fault on RMS internal points format (see A.10). The point set have the following attributes:

Dip Local dip in degrees.

Strike Local strike in degrees.

Displacement Local displacement attributed to the current fault.

TotalDisplacement Local total displacement, both from this fault and all truncating faults. Corresponds to the observed displacement.

FwMoved How much a point on the foot wall side of the fault has been moved. Used for inverse movement.

HwMoved How much a point on the hanging wall side of the fault has been moved. Used for inverse movement.

displacement_field_generation contains a file for each input fault on RMS internal points format (A.10) containing information used in the generation of the displacement field. This information includes:

DataPoints Displacement data obtained from fault lines. A value of 0 indicates that there are no displacement data for the given point. The data points are estimated based on the restored fault lines.

Trend The elliptic trend estimated from the data points.

Displacement The values in the final displacement field for the fault.

RelDisplacement Relative displacement, displacement scaled to a value between 0 and 10.

side_of_fault contains a grid for each fault on STORM binary format with codes telling which side of the fault the grid cells are. The values are

0 if the cell is outside the volume affected by the fault.

1 if the cell is on the hanging wall side of the fault.

2 if the cell is on the foot wall side of the fault.

distance_to_fault contains a grid file for each input fault. The grid is populated with distances to the fault surface for the given fault. The distance is positive on the hanging wall side of the fault, and negative on the foot wall side. For points outside the fault surface, the distance to the reference plane is given.

fault_points_sided contains the fault line points sorted per horizon, fault and side of fault. Contains two subdirectories, one with the fault lines on Roxar text format (see A.8), and one with the fault lines on RMS internal points format (A.10), tagged with a segment number visualizing how HAVANA splits the set of fault line points according to truncations.

restored_fault_lines contains the restored fault line points in the same formats as described for fault_points_sided. In addition the fault lines for each horizon are given in Roxar text format (A.8) and internal points format (A.10).

fault_surface_extrapolation contains a set of surface files for each fault visualizing the fault surface on various steps in the fault surface extrapolation.

fault_surfaces contains a file for each fault surface on Roxar text format (A.8) with the surface fault points.

6.3 `<fault-uncertainty-model>`

This action is used to establish an uncertainty envelope around the fault consisting of one surface on the hanging wall side and one surface on the footwall side of the fault surface. There are two ways of specifying input to create the uncertainty envelope for a fault. The first is to specify a width directly, the second is to specify point sets representing interpretations of the envelope surfaces at the hanging wall and foot wall side of the fault. This can be done by reading a point set from file and interpolate these points to create the uncertainty envelope.

For a fault uncertainty envelope to be generated, either command `<input-pointset>` or `<constant-envelope-width>` or both must be given. However, for a specific fault only one of these commands are used (constant envelope width is ignored if both are specified for a specific fault).

Example:

```
<fault-uncertainty-model>
    <input-havana-faults-directory> origFaultSet </input-havana-faults-directory>
    <output-havana-faults-directory> modifiedFaultSet </output-havana-faults-directory>
    <output-fault-envelope-directory> faultSurface </output-fault-envelope-directory>
        <geometry-variogram>
            <type> Spherical </type>
            <strike-range> 1500 </strike-range>
            <dip-range> 1500 </dip-range>
            <anisotropy-angle> 0 </anisotropy-angle>
        </geometry-variogram>
    <fault-uncertainty-envelope>
    <input-pointset>
        <fault-name> F1 </fault-name>
        <file>test </file>
    </input-pointset>
    <input-pointset>
        <fault-name> F2 </fault-name>
        <file> test2 </file>
    </input-pointset>
    <constant-envelope-width>
        <fault-name> F1 </fault-name>
        <width> 250 </width>
    </constant-envelope-width>
    <constant-envelope-width>
        <fault-name> F2 </fault-name>
        <width> 45 </width>
    </constant-envelope-width>
    </fault-uncertainty-envelope>
</fault-uncertainty-model>
```

6.3.1 `<input-havana-faults-directory>` **(necessary)**

Description: Specifies the name of the directory where the faults on the Havana internal format are read from (see Appendix A.10).

Argument: One. The directory.

Default:

Repeatable: No

6.3.2 `<output-havana-faults-directory>` **(necessary)**
See 6.2.2.

6.3.3 `<output-fault-envelope-directory>`
Description: Specifies the name of the directory where the faults with their uncertainty envelopes
   are written to. Only faults that have an envelope on at least one side are written. The fault
   surface is also written. The surfaces are written on Roxar Text format (see Appendix A.8).
Argument: One. The directory.
Default:
Repeatable: No

6.3.4 `<geometry-variogram>`
Description: Specifies a variogram model used when simulating or predicting the fault surface.
Argument: Commands specifying the variogram.
Default:
Repeatable: No

6.3.4.1 `<type>` **(necessary)**
Description: Specifies the variogram type. Possible types are GAUSSIAN, SPHERICAL,
   EXPONENTIAL or GENERAL_EXPONENTIAL. For the GENERAL_EXPONENTIAL type
   an additional parameter `<power>` must be given.
Argument: One. The variogram type.
Default: SPHERICAL
Repeatable: No

6.3.4.2 `<strike-range>` **(necessary)**
Description: Specifies the range in strike direction. Higher range gives smoother fault surfaces.
Argument: One. A positive number.
Default: 1000
Repeatable: No

6.3.4.3 `<dip-range>` **(necessary)**
Description: Specifies the range in dip direction. Higher range gives smoother fault surface.
Argument: One. A positive number.
Default: 2000
Repeatable: No

6.3.4.4 `<anisotropy-angle>` **(necessary)**
Description: Specifies the anisotropy angle in degrees. Use this for rotating the direction of the
   ranges.
Argument: One. Degree in the interval 0 to 360.
Default: 0
Repeatable: No

6.3.4.5 `<power>`
Description: Specifies the power. Only relevant for general the variogram type
   GENERAL_EXPONENTIAL.

Argument: One. A number.
Default:
Repeatable: No

6.3.5 `<fault-uncertainty-envelope>` **(necessary)**

Description: Specifies the envelope input.
Argument: Commands specifying the envelope input.
Repeatable: No

6.3.5.1 `<input-pointset>`

Description: Specifies input point set used to define uncertainty envelope for a given fault. The command must be repeated for pointset to be specified, and can be repeated for the same fault (meaning that point sets from several files for the same fault are pooled into one big set).
Argument: Commands specifying name of fault and input file.
Repeatable: Yes

6.3.5.1.1 `<fault-name>` **(necessary)**

Description: Name of fault that the point set file is specified for.
Argument: One. Name of fault.
Default:
Repeatable: No

6.3.5.1.2 `<file>` **(necessary)**

Description: Name of file with point set used to define the uncertainty envelope for the given fault.
Argument: One. The filename.
Default:
Repeatable: No

6.3.5.2 `<constant-envelope-width>`

Description: Specifies a constant envelope distance for a given fault. This is the distance between the envelope surfaces. Hence a distance of 100 meters means that the width from fault surface to each of the envelope surfaces is 50 meters. The command must be repeated for each fault that should be given a constant envelope width. If command `<input-pointset>` is given for the same fault, this command is ignored.
Argument: Commands specifying name of fault and constant envelope width.
Repeatable: Yes

6.3.5.2.1 `<fault-name>` **(necessary)**

Description: Name of fault that the constant envelope width is specified for. If 'default' is given as argument the value applies to all faults with no explicitly given value.
Argument: One. Name of fault.
Default:
Repeatable: No

6.3.5.2.2 `<width>` **(necessary)**

Description: Specifies a constant distance away from the fault along the normal vector that
defines the uncertainty envelope on each side of the fault surface. The horizontal distance is
therefore larger than the width due to the dip of the fault surface reference plane (see Figure
6.2).

Argument: One. A positive number.

Default:

Repeatable: No
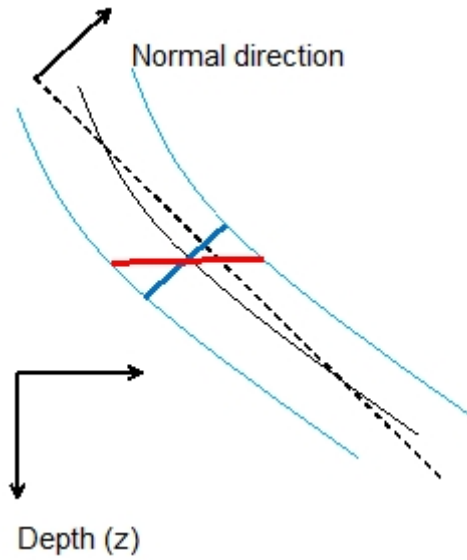
Figure 6.2. Width of uncertainty envelope is specified by user to be equal to the thick blue line. Due to the
dip of the fault surface reference plane, the horizontal width is equal to the red line.

6.4 `<modify-fault-surface>`

This action is used to modify the fault surface of a set of faults. The `<modification>` command can be given independently of each other for various (or the same) faults. The modifications are performed in order listed under command `<operation>`.

Example:

```
<modify-fault-surface>
    <input-havana-faults-directory> origFaultSet </input-havana-faults-directory>
    <output-havana-faults-directory> modifiedFaultSet </output-havana-faults-directory>
    <output-rms-faults-points-directory> surfaceFaults </output-rms-faults-points-directory>
    <modification>
        <fault-name> F1 </fault-name>
        <operation> TRANSLATE_NORM </operation>
        <value> 0.8 </value>
    </modification>
    <modification>
        <fault-name> F2 </fault-name>
        <operation> CHANGE_DIP </operation>
        <value> 0.8 </value>
        <point>
            <x> 602000 </x>
            <y> 7201000 </y>
            <z> 2100 </z>
        </point>
    </modification>
    <well-conditioning>
        <well-picks-file> well-file </well-picks-file>
        <geometry-variogram>
            <type> Spherical </type>
            <strike-range> 1500 </strike-range>
            <dip-range> 1500 </dip-range>
            <anisotropy-angle> 0 </anisotropy-angle>
        </geometry-variogram>
    </well-conditioning>
</modify-fault-surface>
```

6.4.1 `<input-havana-faults-directory>` **(necessary)**
See 6.3.1.

6.4.2 `<output-havana-faults-directory>`
See 6.2.2.

6.4.3 `<output-rms-faults-points-directory>` **(necessary)**
Description: Specifies the name of the directory where the faults are written to. The faults are written on Roxar text format (see Appendix A.8). In addition a file named Truncations.txt containing all the truncation rules for the fault set is written to the mother directory.
Argument: One. The directory.
Default:

Repeatable: No

6.4.4 `<modification>` **(necessary)**

Description: Specifies a modification of fault surface for a given fault. The command must be
repeated for each fault that should be modified.

Argument: Commands specifying name of fault and modification.

Repeatable: Yes

6.4.4.1 `<fault-name>` **(necessary)**

Description: Name of fault that the modification is specified for.

Argument: One. Name of fault.

Default:

Repeatable: No

6.4.4.2 `<operation>` **(necessary)**

Description: Specification of the modification for the given fault. Possible commands are:

CHANGE_STRIKE and CHANGE_DIP: Specifies rotations on the faults by changing
azimuth and/or dip angle.

TRANSLATE_X, TRANSLATE_Y and TRANSLATE_NORM: Specifies transformations
performed on the fault; i.e. parallel to the x or y-axis or translation parallel to the faults
normal vector projected to the global xy-plane. A positive value means moving in hanging
wall direction and a negative value in foot wall direction.

TRANSLATE_NORM_RELATIVE: Translates the fault parallel to the normal vector
projected to the global xy-plane with a factor relative to the distance to the fault uncertainty
envelope on the relevant side of the fault. Note that fault uncertainty envelope must be
defined by the action `<fault-uncertainty-envelope>`. In case these envelopes are not
properly defined, the translation is ignored.

Argument: One. The name of the command.

Default:

Repeatable: No

6.4.4.3 `<value>` **(necessary)**

Description: Specifies the value of the modification operation.

Argument:

CHANGE_STRIKE and CHANGE_DIP: One. Change of angle given i degrees.

TRANSLATE_X, TRANSLATE_Y and TRANSLATE_NORM: One. Distance for translation.
For TRANSLATE_NORM a positive value means moving in hanging wall direction and a
negative value in foot wall direction.

TRANSLATE_NORM_RELATIVE: One. Translation factor. Must be between -1 and 1,
positive number meaning moving in hanging wall direction and negative in foot wall
direction.

Default:

Repeatable: No

6.4.4.4 `<point>`

Description: Specifies the point where the rotation is made if CHANGE_STRIKE or CHANGE_DIP is used. All three coordinates are not required, if one of x, y or z is missing the respective reference coordinate of the fault is used. If the rotation point does not coincide with the fault surface, it is projected onto the fault surface along the fault's normal vector.

Argument: Point in global coordinates.

Default: Reference point.

Repeatable: No

6.4.4.4.1 `<x>`

Description: Global x-coordinate.

Default: x-coordinate of reference point.

6.4.4.4.2 `<y>`

Description: Global y-coordinate

Default: y-coordinate of reference point.

6.4.4.4.3 `<z>`

Description: Global z-coordinate

Default: z-coordinate of reference point.

6.4.5 `<well-conditioning>`

Description: Specifies the input required for well conditioning during the modifications.

Argument: Commands specifying the input for well conditioning.

Repeatable: No

6.4.5.1 `<well-picks-file>` **(necessary)**

Description: File with well observations of faults. Used for conditioning on fault surface and uncertainty envelope. The well picks file has RMS format as described in Appendix A.4.1.

Argument: One. The filename.

Default:

Repeatable: No

Example:

```
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
```

6.4.5.2 `<geometry-variogram>`

See 6.3.4.

6.4.5.2.1 `<type>` **(necessary)**

See 6.3.4.1.

6.4.5.2.2 `<strike-range>` **(necessary)**

See 6.3.4.2.

6.4.5.2.3 `<dip-range>` **(necessary)**
See 6.3.4.3.

6.4.5.2.4 `<anisotropy-angle>` **(necessary)**
See 6.3.4.4.

6.4.5.2.5 `<power>`
See 6.3.4.5.

6.5 `<simulate-fault-surface>`

This action is used to generate stochastic realisations of a set of faults. Note that the action `<fault-uncertainty-envelope>` must be performed prior to this action in order to generate uncertainty volumes around the faults that are going to be simulated.

Example:

```
<simulate-fault-surface>
    <input-havana-faults-directory> origFaultSet </input-havana-faults-directory>
    <output-havana-faults-directory> modifiedFaultSet </output-havana-faults-directory>
    <output-rms-faults-points-directory> surfaceFaults </output-rms-faults-points-directory>
    <simulation-settings>
        <distribution> NORMAL </distribution>
        <number-of-realizations> 2 </number-of-realizations>
        <geometry-variogram>
            <type> General_exponential </type>
            <strike-range> 1500 </strike-range>
            <dip-range> 1500 </dip-range>
            <anisotropy-angle> 0 </anisotropy-angle>
            <power> 1 </power>
        </geometry-variogram>
        <seed> 123123123 </seed>
    </simulation-settings>
    <well-picks>
        <file> well_picks </file>
        <uncertainty> 10 </uncertainty>
    </well-picks>
    <well-picks-negative>
        <file> well_picks_negative </file>
    </well-picks-negative>
    <simulate-fault> F1 </simulate-fault>
    <simulate-fault> F3 </simulate-fault>
    <prediction-envelope-percentile> 5 </prediction-envelope-percentile>
    <predict-fault> F1 </predict-fault>
</simulate-fault-surface>
```

6.5.1 `<input-havana-faults-directory>` **(necessary)**
See 6.3.1.

6.5.2 `<output-havana-faults-directory>`
See 6.2.2. The specified directory will contain the predicted fault set. If one realization is simulated, a directory with suffix `_Simulated` is generated with the simulated fault set. If more than one realization is simulated the simulated fault set is not written on internal Havana format.

6.5.3 `<output-rms-faults-points-directory>` **(necessary)**
See 6.4.3. To this directory the predicted faults are written with prefix `predicted_` and the uncertainty envelopes for the simulated and/or predicted faults has prefix `fw_uncertainty_` and `hw_uncertainty_` respectively. If only one realization is simulated, the simulated faults are given prexix `simulated`, but if more than one realization is simulated, the simulated faults are

written to a separate sub-directory with name `Simulated_n` where `n` is the realization number.

6.5.4 `<simulation-settings>`
Description: Specifies the settings for the simulations.
Argument: Commands specifying the simulation settings.
Repeatable: No

6.5.4.1 `<distribution>`
Description: Distribution for points on fault surfaces within fault uncertainty envelope specified under the ACTION FaultUncertaintyModel. Uniform and triangular distributions ensure that the simulated fault surface is within the given fault uncertainty envelope. Possible commands are:

UNIFORM - Uniform distribution within the given fault uncertainty envelope.

TRIANGULAR - Triangular distribution within given fault uncertainty envelope 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.

PIECEWISE_UNIFORM - Piecewise uniform distribution within the given fault uncertainty envelope, where either side of the base case is uniformly distributed, and the base case equals the mean.

Argument: One. Name of distribution.
Default: UNIFORM
Repeatable: No

6.5.4.2 `<number-of-realizations>`
Description: Number of realizations of the simulation.
Argument: One. The number of realizations
Default: 1
Repeatable: No

6.5.4.3 `<geometry-variogram>`
See 6.3.4.

6.5.4.3.1 `<type>` **(necessary)**
See 6.3.4.1.

6.5.4.3.2 `<strike-range>` **(necessary)**
See 6.3.4.2.

6.5.4.3.3 `<dip-range>` **(necessary)**
See 6.3.4.3.

6.5.4.3.4 `<anisotropy-angle>` **(necessary)**
See 6.3.4.4.

6.5.4.3.5 `<power>`

See 6.3.4.5.

6.5.4.4 `<seed>`

Description: Controls the initial seed of the random generator.

Argument: Either a name of a file with seed or an integer number between 0 and 4294967295.

Default: The random generator is initialized at random by the computers current time.

Repeatable: No

6.5.5 `<well-picks>`

Description: Well observations of faults used for conditioning on fault surface and uncertainty envelope.

Argument: Commands specifying the input required for the conditioning.

Repeatable: No

6.5.5.1 `<file>` **(necessary)**

Description: Specifies the input file with well observations of faults on RMS format as described in Appendix A.4.1.

Argument: One. The filename.

Default:

Repeatable: No

Example:

```
        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
```

6.5.5.2 `<uncertainty>`

Description: Common uncertainty for all given fault picks. The value is interpreted as 2 times the standard deviation in a normal distribution for the observation uncertainty of the fault pick along the faults' normal vector that the fault is allowed to vary. The uncertainty envelopes are conditioned to the point that is located this distance away from the fault pick on each side of the fault. If the given uncertainty envelope is closer to the fault pick than this distance the closest point is used in the conditioning.

Argument: One. A non-negative number.

Default: 0

Repeatable: No

6.5.6 `<well-picks-negative>`

Description: Wells with no observed faults used for conditioning on fault surface and uncertainty envelopes. These well observations are used to limit the extent of the fault uncertainty volume, ensuring that the modified fault surface does not cross the well path. It is recommended to use a resolution typical for well trajectories (e.g. 15-30 m) instead of a finer resolution in order to reduce computation time when conditioning to these well picks with no observed faults.

Argument: Commands specifying the input required for the conditioning.

Repeatable: No

6.5.6.1 `<file>` **(necessary)**

Description: Specifies the input well files with no observations. Points inside the fault uncertainty envelopes for each fault is used. For each well-file all the points are set to either HW or FW side, depending on which side of the fault most of the points occur. File format described in Appendix A.4.2.

Argument: One. The filename.

Default:

Repeatable: Yes

Example:

```
169800.000    570800.000    1727.5908
169047.814    570891.481    1789.8117
169958.321    569800.000    1832.5233
```

6.5.7 `<simulate-fault>` **(necessary)**

Description: Specifies the name of a fault that should be simulated. The command must be repeated for each fault that should be simulated. If 'default' is given as argument the value applies to all faults with no explicitly given command.

Argument: Name of fault.

Default:

Repeatable: Yes

6.5.8 `<prediction-envelope-percentile>`

Description: Specifies the percentile in the distribution for points within the fault uncertainty envelope that is used for output of updated uncertainty envelopes. For value p, the $(100 - p)$ and p percentile surfaces are exported. Only relevant when command `<predict-fault>` is given.

Argument: One. Percentile as positive number. A value above 50 gives no meaning.

Default: 5

Repeatable: No

6.5.9 `<predict-fault>`

Description: Specifies the name of a fault that should be predicted. The command must be repeated for each fault that should be predicted. If 'default' is given as argument the value applies to all faults with no explicitly given command. The trend surface and the uncertainty envelopes defined by the command `<prediction-envelope-percentile>` is calculated and written. In case there are well picks with associated uncertainty defined by `<well-picks>` the predicted surfaces are conditioned on these.

Argument: Name of fault.

Default:

Repeatable: Yes

6.6 `<modify-displacement>`

This action is used to modify the displacement field of a set of faults. The `<modification>` command can be given independently of each other for various (or the same) faults. The modifications are performed in the order listed under command `<operation>`.

Example:
```
<modify-displacement>
    <input-havana-faults-directory> origFaultSet </input-havana-faults-directory>
    <output-havana-faults-directory> modifiedFaultSet </output-havana-faults-directory>
    <output-rms-faults-points-directory> surfaceFaults </output-rms-faults-points-directory>
    <debug-output-fault-info-directory> faultInfo </debug-output-fault-info-directory>
    <displacement-variogram>
        <type> Spherical </type>
        <strike-range> 1500 </strike-range>
        <dip-range> 1500 </dip-range>
        <anisotropy-angle> 0 </anisotropy-angle>
    </displacement-variogram>
    <default-displacement-settings>
        <max-displacement-length-relation>
            <ratio> 0.01 </ratio>
            <power> 1 </power>
        </max-displacement-length-relation>
        <length-height-ratio> 2.0 </length-height-ratio>
    </default-displacement-settings>
    <modification>
        <fault-name> F1 </fault-name>
        <operation> SCALE_DISPLACEMENT </operation>
        <value> 0.8 </value>
    </modification>
    <modification>
        <fault-name> F3 </fault-name>
        <operation> ADD_THROW </operation>
        <value> 10 </value>
        <displacement-settings>
            <max-displacement-length-relation>
                <ratio> 0.01 </ratio>
                <power> 1 </power>
            </max-displacement-length-relation>
            <length-height-ratio> 2.0 </length-height-ratio>
        </displacement-settings>
    </modification>
</modify-displacement>
```

6.6.1 `<input-havana-faults-directory>` **(necessary)**
See 6.3.1.

6.6.2 `<output-havana-faults-directory>`
See 6.2.2.

6.6.3 `<output-rms-faults-points-directory>`

See 6.4.3.

6.6.4 `<debug-output-fault-info-directory>`

Description: Specifies the name of the directory where the debug fault information is written to. The global data and point sets for all faults are written.

Argument: One. The directory.

Default:

Repeatable: No

6.6.5 `<displacement-variogram>`

See 6.2.5.

6.6.5.1 `<type>` **(necessary)**

See 6.2.5.1.

6.6.5.2 `<strike-range>` **(necessary)**

See 6.2.5.2.

6.6.5.3 `<dip-range>` **(necessary)**

See 6.2.5.3.

6.6.5.4 `<anisotropy-angle>` **(necessary)**

See 6.2.5.4.

6.6.5.5 `<power>`

See 6.2.5.5.

6.6.6 `<default-displacement-settings>`

See 6.2.6.

6.6.6.1 `<max-displacement-length-relation>`

See 6.2.6.3.

6.6.6.1.1 `<ratio>` **(necessary)**

See 6.2.6.3.1.

6.6.6.1.2 `<power>` **(necessary)**

See 6.2.6.3.2.

6.6.6.2 `<length-height-ratio>`

See 6.2.6.4.

6.6.7 `<modification>` **(necessary)**

Description: Specify a modification of the displacement field for a given fault. The command must be repeated for each fault that should be modified.

Argument: Commands specifying name of fault and modification.

Repeatable: Yes

6.6.7.1 `<fault-name>` **(necessary)**

Description: Name of fault that the modification is specified for.

Argument: One. Name of fault.

Default:

Repeatable: No

6.6.7.2 `<operation>` **(necessary)**

Description: Specification of the modification for a given fault. A modified displacement grid for each fault is written to a file in the directory specified in the command `<output-havana-faults-directory>`. This modified fault is used in the action `<update-points>` or `<update-grid>`. Possible commands are:

SCALE_DISPLACEMENT: Specifies a factor for change of displacement. The displacement at every point on the fault surface is multiplied with this factor.

ADD_THROW: Specifies a number for the change of throw by adding a constant value. The throw at every point on the fault surface is increased by this value. If the value is positive, the foot wall side is moved up and hanging wall side is moved down. If the value is negative, the foot wall side is moved down and hanging wall side is moved up. Note that the fault tip is changed when adding or subtracting fault throw. Hence it is necessary to save the fault surface point sets and use the new fault surface point sets to update the structural model in RMS.

Argument: One. The name of the command.

Default:

Repeatable: No

6.6.7.3 `<value>` **(necessary)**

Description: Specifies the value of the modification operation.

Argument:

SCALE_DISPLACEMENT: One. A non-negative number.

ADD_THROW: One. A number.

Default:

Repeatable: No

6.6.7.4 `<displacement-settings>`

See 6.2.7. Only relevant when operation ADD_THROW is given under `<operation>`.

6.6.7.4.1 `<max-displacement-length-relation>`

See 6.2.6.3.

6.6.7.4.1.1 `<ratio>` **(necessary)**

See 6.2.6.3.1.

6.6.7.4.1.2 `<power>` **(necessary)**

See 6.2.6.3.2.

6.6.7.4.2 `<length-height-ratio>`

See 6.2.6.4.

6.7 `<simulate-displacement>`

In this action the fault displacement fields and the corresponding fault tip lines can be predicted or simulated from a set of input displacement observations.

Example:

```
<simulate-displacement>
    <input-havana-faults-directory> origFaultSet </input-havana-faults-directory>
    <output-havana-faults-directory> modifiedFaultSet </output-havana-faults-directory>
    <output-rms-faults-points-directory> surfaceFaults </output-rms-faults-points-directory>
    <debug-output-fault-info-directory> faultInfo </debug-output-fault-info-directory>
    <displacement-variogram>
        <type> spherical </type>
        <strike-range> 2000 </strike-range>
        <dip-range> 1000 </dip-range>
        <anisotropy-angle> 90 </anisotropy-angle>
    </displacement-variogram>
    <default-displacement-settings>
        <max-displacement-length-relation>
            <ratio> 0.05 </ratio>
            <power> 1 </power>
        </max-displacement-length-relation>
        <length-height-ratio> 2.0 </length-height-ratio>
    </default-displacement-settings>
    <simulation-settings>
        <seismic-resolution> 10 </seismic-resolution>
        <seed> seed.txt </seed>
    </simulation-settings>
    <simulate>
        <fault-name> default </fault-name>
        <operation> SIMULATE </operation>
        <displacement-settings>
            <max-displacement-length-relation>
                <ratio> 0.05 </ratio>
                <power> 1 </power>
            </max-displacement-length-relation>
            <length-height-ratio> 2.0 </length-height-ratio>
        </displacement-settings>
    </simulate>
</simulate-displacement>
```

6.7.1 `<input-havana-faults-directory>` **(necessary)**
See 6.3.1.

6.7.2 `<output-havana-faults-directory>`
See 6.2.2.

6.7.3 `<output-rms-faults-points-directory>`
See 6.4.3.

6.7.4 `<debug-output-fault-info-directory>`

See 6.6.4.

6.7.5 `<displacement-variogram>`

See 6.2.5.

6.7.5.1 `<type>` **(necessary)**

See 6.2.5.1.

6.7.5.2 `<strike-range>` **(necessary)**

See 6.2.5.2.

6.7.5.3 `<dip-range>` **(necessary)**

See 6.2.5.3.

6.7.5.4 `<anisotropy-angle>` **(necessary)**

See 6.2.5.4.

6.7.5.5 `<power>`

See 6.2.5.5.

6.7.6 `<default-displacement-settings>`

See 6.2.6.

6.7.6.1 `<max-displacement-length-relation>`

See 6.2.6.3.

6.7.6.1.1 `<ratio>` **(necessary)**

See 6.2.6.3.1.

6.7.6.1.2 `<power>` **(necessary)**

See 6.2.6.3.2.

6.7.6.2 `<length-height-ratio>`

See 6.2.6.4.

6.7.7 `<simulation-settings>` **(necessary)**

Description: Specifies the simulation settings.
Argument: Commands specifying the simulation settings.
Repeatable: No

6.7.7.1 `<seismic-resolution>` **(necessary)**

Description: The seismic resolution, which defines standard deviation of the uncertainty in observations.
Argument: One. A constant value.
Default:
Repeatable: No

6.7.7.2 `<seed>`

See 6.5.4.4.

6.7.8 `<simulate>` **(necessary)**

Description: Specifies the input required for simulating or predicting the displacement field for a given fault. The command must be repeated for each fault that should be simulated/predicted. If 'default' is given as argument under `<fault-name>` the commands applies to all faults with no explicitly given commands.

Argument: Commands specifying the simulation input.

Repeatable: Yes

6.7.8.1 `<fault-name>` **(necessary)**

Description: Name of fault to be simulated. If 'default' is given as argument the commands applies to all fault with no explicitly given commands.

Argument: One. The fault name.

Default:

Repeatable: No

6.7.8.2 `<operation>` **(necessary)**

Description: Specification of the operation for a given fault. Possible commands are SIMULATE, PREDICT and UNCHANGED.

Argument: One. The name of the command.

Default:

Repeatable: No

6.7.8.3 `<displacement-settings>`

See 6.2.7.

6.7.8.3.1 `<max-displacement-length-relation>`

See 6.2.6.3.

6.7.8.3.1.1 `<ratio>` **(necessary)**

See 6.2.6.3.1.

6.7.8.3.1.2 `<power>` **(necessary)**

See 6.2.6.3.2.

6.7.8.3.2 `<length-height-ratio>`

See 6.2.6.4.

6.8 `<remove-faults>`

This action removes the faults specified under the command `<remove-fault>` from the fault set.
The new fault set without the removed faults is written to file. The truncation rules are updated
by removing truncation rules that involve any of the removed faults.

Example:

```
<remove-faults>
    <input-havana-faults-directory> origFaultSet </input-havana-faults-directory>
    <output-havana-faults-directory> modifiedFaultSet </output-havana-faults-directory>
    <output-rms-faults-points-directory> surfaceFaults </output-rms-faults-points-directory>
    <remove-fault> F1 </remove-fault>
    <remove-fault> F6 </remove-fault>
</remove-faults>
```

6.8.1 `<input-havana-faults-directory>` **(necessary)**
See 6.3.1.

6.8.2 `<output-havana-faults-directory>` **(necessary)**
See 6.2.2.

6.8.3 `<output-rms-faults-points-directory>`
See 6.4.3.

6.8.4 `<remove-fault>` **(necessary)**
Description: Specifies the name of the fault that should be removed from the fault set. The
command must be repeated for each fault that should be removed.
Argument: One. The fault name
Default:
Repeatable: Yes

6.9 `<add-faults>`

This action is used to include new faults to the fault set. New faults can be included as elliptic faults or as fault surface extracted from RMS without displacement field. Elliptic faults that are added have specified displacement field, but the fault surfaces extracted from RMS have not. The displacement field for these faults can be set by the action `<set-displacement>`. New truncation rules related to the new faults are read from file and added to the fault set.

Example:
```
<add-faults>
    <input-havana-faults-directory> origFaultSet </input-havana-faults-directory>
    <output-havana-faults-directory> modifiedFaultSet </output-havana-faults-directory>
    <output-rms-faults-points-directory> outputFaults </output-rms-faults-points-directory>
    <truncation-rules-file> truncation_rules.txt </truncation-rules-file>
    <add-fault-surface>
        <fault-name> F11 </fault-name>
        <file> new_rms_fault/F11_grid </file>
    </add-fault-surface>
    <add-elliptic-faults>
        <file> displacement_field.ell </file>
        <surface-grid-resolution> 50 </surface-grid-resolution>
    </add-elliptic-faults>
</add-faults>
```

6.9.1 `<input-havana-faults-directory>` **(necessary)**
See 6.3.1.

6.9.2 `<output-havana-faults-directory>` **(necessary)**
See 6.2.2.

6.9.3 `<output-rms-faults-points-directory>`
See 6.4.3.

6.9.4 `<truncation-rules-file>`
Description: Specifies the name of the file with new truncation rules to be added. The file format is plain ascii where each truncation rule is given by one line of three parameters, i.e. the truncated fault, the truncating fault and the side of the truncating fault where the fault is truncated.
Argument: One. The filename.
Default:
Repeatable: No
Example:
```
        F11 F1 hw
        F12 F4 hw
        #F13 F1 fw
```

6.9.5 `<add-fault-surface>`

Description: Specifies the name and the input file of a fault surface extracted from RMS. This fault surface has no displacement field. The command must be repeated for each fault surface that should be included in the fault set.

Argument: Commands specifying name of fault and input file.

Repeatable: Yes

6.9.5.1 `<fault-name>` **(necessary)**

Description: Name of fault which fault surface should be included.

Argument: One. The fault name.

Default:

Repeatable: No

6.9.5.2 `<file>` **(necessary)**

Description: Name of file with fault surface data. The file format is described in Appendix A.1.2. The file is usually read from the RMS export directory for the structural model.

Argument: One. The filename.

Default:

Repeatable: No

6.9.6 `<add-elliptic-faults>`

Description: Specifies the input required for including elliptic faults to the fault set. The faults are imported on simplified elliptic format.

Argument: Command specifying the input file and surface grid resolution.

Repeatable: No

6.9.6.1 `<file>` **(necessary)**

Description: Specifies the input file with the elliptic faults. The format is described in Appendix A.2.

Argument: One. The filename.

Default:

Repeatable: No

6.9.6.2 `<surface-grid-resolution>`

Description: Specifies the resolution of the surface grid in the x and y direction for the elliptic faults added.

Argument: One. A positive number.

Default: 10.0

Repeatable: No

6.10 `<set-displacement>`

This action is used to add elliptic displacement field to existing faults or fault surfaces. The action is relevant when new fault surfaces without displacement field have been added to the fault set in the command `<add-faults>`, however can also replace existing displacement field in any of the faults given in the fault set.

The displacement field can be specified by giving the maximum displacement (command `<set-max-displacement>`), whereto displacement settings can be given, however is optional. If this is not specified, default values are used. Displacement setting parameters can be specified under command `<default-displacement-settings>` as new default values for all faults, or individually for each fault under command `<displacement-settings>`. The centre point of the ellipse is found based on the given horizon (command `<horizon>`) and fault line points (command `<input-fault-lines-file>`) if these commands are given. Otherwise the maximum displacement is set at the centre point of the fault surface.

Alternatively the displacement field can be set by specifying the elliptic fault parameters in an input file by the command `<set-displacement-from-file>`. This requires that the centre point of the ellipse is specified to be located at or close to the fault surface of the corresponding fault.

Example:

```
<set-displacement>
    <input-havana-faults-directory> origFaultSet </input-havana-faults-directory>
    <output-havana-faults-directory> modifiedFaultSet </output-havana-faults-directory>
    <output-rms-faults-points-directory> surfaceFaults </output-rms-faults-points-directory>
    <debug-output-fault-info-directory> faultInfo </debug-output-fault-info-directory>
    <default-displacement-settings>
        <asymmetry> 0.7 </asymmetry>
        <range> 2000 </range>
        <type> NORMAL </type>
        <max-displacement-length-relation>
            <ratio> 0.01 </ratio>
            <power> 1 </power>
        </max-displacement-length-relation>
        <length-height-ratio> 2.0 </length-height-ratio>
    </default-displacement-settings>
    <set-max-displacement>
        <fault-name> F1 </fault-name>
        <max-displacement> 200 </max-displacement>
        <horizon> TopC </horizon>
        <displacement-settings>
            <asymmetry> 0.7 </asymmetry>
            <range> 2000 </range>
            <type> NORMAL </type>
            <max-displacement-length-relation>
                <ratio> 0.01 </ratio>
                <power> 1 </power>
            </max-displacement-length-relation>
            <length-height-ratio> 2.0 </length-height-ratio>
        </displacement-settings>
    </set-max-displacement>
```

```
            <input-fault-lines-file> fault_lines </input-fault-lines-file>
            <set-displacement-from-file> displacement_field.ell </set-displacement-from-file>
        </set-displacement>
```

6.10.1 `<input-havana-faults-directory>` **(necessary)**
See 6.3.1.

6.10.2 `<output-havana-faults-directory>` **(necessary)**
See 6.2.2.

6.10.3 `<output-rms-faults-points-directory>`
See 6.4.3.

6.10.4 `<debug-output-fault-info-directory>`
See 6.6.4.

6.10.5 `<default-displacement-settings>`
See 6.2.6.

6.10.5.1 `<asymmetry>`
See 6.2.6.1.

6.10.5.2 `<range>`
See 6.2.6.2.

6.10.5.3 `<type>`
Description: Parameter giving the fault type. Possible commands are NORMAL and REVERSE.
Argument: One. The fault type.
Default: NORMAL
Repeatable: No

6.10.5.4 `<max-displacement-length-relation>`
See 6.2.6.3.

6.10.5.4.1 `<ratio>` **(necessary)**
See 6.2.6.3.1.

6.10.5.4.2 `<power>` **(necessary)**
See 6.2.6.3.2.

6.10.5.5 `<length-height-ratio>`
See 6.2.6.4.

6.10.6 `<set-max-displacement>`
Description: Specifies the input required for setting the displacement field based on a given
    maximum displacement value. The command must be repeated for each fault that should be
    given a displacement field based on maximum displacement.
Argument: Commands specifying the input.

Repeatable: Yes

6.10.6.1   `<fault-name>` **(necessary)**

Description: Name of fault which should be given a maximum displacement field.
Argument: One. The fault name.
Default:
Repeatable: No

6.10.6.2   `<max-displacement>` **(necessary)**

Description: Maximum displacement of the displacement field to be added to the given fault. The maximum displacement is located in the centre point of the fault surface. If horizon name is given under `<horizon>` and fault lines are given (command `<input-fault-lines-file>`), the maximum displacement is set at the centre point of this horizon. If fault lines are given but no horizon specified, the maximum displacement is located at the centre point of all fault line points for the given fault.
Argument: One. Maximum displacement as a positive number.
Default:
Repeatable: No

6.10.6.3   `<horizon>`

Description: The horizon which the maximum displacement should be located at. This requires that the fault line set is given under command `<input-fault-lines-file>` and that fault line points for this horizon is given in the fault line set.
Argument: One. The name of the horizon.
Default:
Repeatable: No

6.10.6.4   `<displacement-settings>`

See 6.2.7.

6.10.6.4.1   `<asymmetry>`

See 6.2.6.1.

6.10.6.4.2   `<range>`

See 6.2.6.2.

6.10.6.4.3   `<type>`

See 6.10.5.3.

6.10.6.4.4   `<max-displacement-length-relation>`

See 6.2.6.3.

6.10.6.4.4.1   `<ratio>` **(necessary)**

See 6.2.6.3.1.

6.10.6.4.4.2   `<power>` **(necessary)**

See 6.2.6.3.2.

6.10.6.4.5 `<length-height-ratio>`

See 6.2.6.4.

6.10.7 `<input-fault-lines-file>`

Description: Specifies a file with fault line points. The fault lines are used to find the center point of the displacement field ellipse, and is only relevant when command
`<set-max-displacement>` is given. The file format is described in Appendix A.1.3 and usually the input file is the file *fault_-lines.rms* located in the RMS export directory for the structural model.

Argument: One. The filename.

Default:

Repeatable: No

6.10.8 `<set-displacement-from-file>`

Description: Specifies the filename for elliptic displacement. Imports elliptic displacement of simplified elliptic format, and add this to the fault specified for each line. The format is described in Appendix A.2.

Argument: One. The filename.

Default:

Repeatable: No

6.11 `<simulate-secondary-faults>`

This action is used to simulate secondary faults that are clustered around larger primary faults. Typically the primary faults are either faults that are observed from seismics or large sub-seismic faults. The primary faults must be present in the input fault set before this action is called. The secondary faults are simulated according to probability distributions specified through user input.

Example:

```
<simulate-secondary-faults>
    <input-havana-faults-directory> origFaultSet </input-havana-faults-directory>
    <output-havana-faults-directory> modifiedFaultSet </output-havana-faults-directory>
    <output-rms-faults-points-directory> surfaceFaults </output-rms-faults-points-directory>
    <output-intensity-grid>
        <total-intensity> yes </total-intensity>
        <nx> 100 </nx>
        <ny> 100 </ny>
        <nz> 10 </nz>
        <file-name> intensity.storm </file-name>
        <primary-fault> F1 </primary-fault>
        <primary-fault> F6 </primary-fault>
    </output-intensity-grid>
    <seed> -238746 </seed>
    <output-fault-summary>
        <file-name>fault_statistics.dat </file-name>
        <report-truncated-length> yes </report-truncated-length>
        <report-truncating-faults> no </report-truncating-faults>
    </output-fault-summary>
    <output-faultname-prefix> simulated_ </output-faultname-prefix>
    <number-of-simulated-faults> 150 </number-of-simulated-faults>
    <surface-grid-resolution> 10 </surface-grid-resolution>
    <fault-truncation>
        <limit> 0.0 </limit>
        <stochastic> no </stochastic>
        <original-faults-truncate> no </original-faults-truncate>
    </fault-truncation>

    <simulation-volume>
        <x-reference> 0.0 </x-reference>
        <x-length> 1000.0 </x-length>
        <y-reference> 0.0 </y-reference>
        <y-length> 2000.0 </y-length>
        <rotation-angle> 0.0 </rotation-angle>
        <top>
            <surface-file> top.irap </surface-file>
        </top>
        <bottom>
            <surface-file> base.irap </surface-file>
        </bottom>
        <vertical-buffer-size> 30.0 </vertical-buffer-size>
```

```
        </simulation-volume>

<displacement-settings>
    <displacement>
        <minimum> 10.0 </minimum>
        <maximum> 30.0 </maximum>
        <fractal-dimension> 2.0 </fractal-dimension>
    </displacement>
    <asymmetry>
        <ratio> 0.5 </ratio>
        <standard-deviation> 0.1 </standard-deviation>
    </asymmetry>
    <max-displacement-length-relation>
        <power> 1.0 </power>
        <ratio> 0.01 </ratio>
        <standard-deviation> 0.1 </standard-deviation>
    </max-displacement-length-relation>
    <length-height-ratio>
        <ratio> 2.0 </ratio>
        <standard-deviation> 0.1 </standard-deviation>
    </length-height-ratio>
    <range-length-height-ratio>
        <ratio> 0.4 </ratio>
        <standard-deviation> 0.1 </standard-deviation>
    </range-length-height-ratio>
    <displacement-weight> 1.0 </displacement-weight>
</displacement-settings>

<group-name> group1 </group-name>

<simulation-settings>
    <primary-fault>
        <name> F1 </name>
        <number-of-faults> 5 </number-of-faults>
    </primary-fault>
    <primary-fault-group>
        <name> faultgroup1 </name>
    </primary-fault-group>
    <position>
        <strike-direction>
            <mean> 0.0 </mean>
            <standard-deviation> 0.5 </standard-deviation>
        </strike-direction>
        <dip-direction>
            <mean> 0.0 </mean>
            <standard-deviation> 0.5 </standard-deviation>
        </dip-direction>
        <normal-direction>
            <absolute-distance> no </absolute-distance>
```

```
            <mean> 0.0 </mean>
            <standard-deviation> 0.5 </standard-deviation>
        </normal-direction>
        <asymmetry-intensity> 0.5 </asymmetry-intensity>
    </position>
    <dip>
        <mean> 0.0 </mean>
        <standard-deviation> 0.1 </standard-deviation>
        <minimum> -1.0 </minimum>
        <maximum> 1.0 </maximum>
    </dip>
    <strike>
        <mean> 5.0 </mean>
        <standard-deviation> 10.0 </standard-deviation>
        <minimum> -10.0 </minimum>
        <maximum> 20.0 </maximum>
    </strike>
    <fraction-of-simulated-faults> 1.0 </fraction-of-simulated-faults>
    <probability-antithetic-fault> 0.5 </probability-antithetic-fault>
    </simulation-settings>
</simulate-secondary-faults>
```

6.11.1 `<input-havana-faults-directory>`

See 6.3.1.

6.11.2 `<input-havana-structural-model-directory>`

Description: If `<input-havana-faults-directory>` is not provided, this keyword is required.
   Specifies the path to a structural model processed by HAVANA.

Argument: One. The directory.

Default:

Repeatable: No

6.11.3 `<output-havana-faults-directory>`

See 6.2.2. Will have no impact if `<input-havana-structural-model-directory>` is used.

6.11.4 `<output-havana-structural-model-directory>`

Description: If `<output-havana-faults-directory>` is not provided, this keyword is required.
   Specifies where to output the structural model with the simulated secondary faults
   included. Will only have an effect if `<input-havana-structural-model-directory>` is
   used.

Argument: One. The directory.

Default:

Repeatable: No

6.11.5 `<output-rms-faults-points-directory>`

See 6.4.3.

6.11.6 `<output-intensity-grid>`

Description: The position of secondary faults are simulated according to local intensities around the primary faults. This command is used to compute one or several grids containing the intensity around the faults in each grid cell center. The grid is defined on the simulation volume and is written on Storm 3D grid format as a 3D parameter for import into RMS.

Argument: Commands specifying the output grids with intensity fields.

Repeatable: No

6.11.6.1 `<total-intensity>`

Description: Indicates whether a grid with the total accumulated intensities for all primary faults is generated.

Argument: One - 'yes' or 'no'.

Default: 'yes'

Repeatable: No

6.11.6.2 `<nx>` **(necessary)**

Description: Number of grid cells in 'x'-direction

Argument: One - A positive integer.

Repeatable: No

6.11.6.3 `<ny>` **(necessary)**

Description: Number of grid cells in 'y'-direction

Argument: One - A positive integer.

Repeatable: No

6.11.6.4 `<nz>` **(necessary)**

Description: Number of grid cells in 'z'-direction

Argument: One - A positive integer.

Repeatable: No

6.11.6.5 `<file-name>` **(necessary)**

Description: Specifies the name of the output intensity file(s). If total intensity is chosen, this intensity will be written to a file with the name define in this command. For primary faults (see `<primary-fault>`) the string `_fault-name` will be added.

Argument: One. The filename.

Repeatable: No

6.11.6.6 `<primary-fault>`

Description: Specifies the name of the faults that an intensity field will be generated for. In this case other primary faults will not be accounted for.

Argument: One. The name of the fault.

Repeatable: Yes

6.11.7 `<seed>` **(necessary)**

See 6.5.4.4

6.11.8 `<output-fault-summary>`

Description: The most important data for the simulated faults are output to a file on an ease-to-read format, see A.5.

Argument: Commands specifying the format of the fault summary file.
Repeatable: No

6.11.8.1 `<file-name>` **(necessary)**
Description: Specifies the name of the output fault summary file.
Argument: One. The filename.
Repeatable: No

6.11.8.2 `<report-truncated-length>`
Description: If this command has 'yes' as argument, 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 inputing Elliptic faults, one may give 'no' as argument. Then, the untruncated lengths and heights will be output.
Argument: One - 'yes' or 'no'.
Default: 'yes'
Repeatable: No

6.11.8.3 `<report-truncating-faults>`
Description: If 'yes' the number of faults truncating each fault and their names is reported. Otherwise no truncations are reported.
Argument: One - 'yes' or 'no'.
Default: 'no'
Repeatable: No

6.11.9 `<output-faultname-prefix>`
Description: Specifies a prefix for the fault names for the generated faults.
Argument: One, a string.
Default: Empty string.
Repeatable: No

6.11.10 `<number-of-simulated-faults>` **(necessary)**
Description: Total number of secondary faults being simulated around the primary faults.
Argument: One, a non-negative number.
Repeatable: No

6.11.11 `<surface-grid-resolution>`
Description: Specifies the resolution of the surface grid in the x and y direction for the simulated faults added.
Argument: One. A positive number.
Default: 10.0
Repeatable: No

6.11.12 `<fault-truncation>` **(necessary)**
Description: Specifies parameters 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 truncation is not specified to be stochastic, 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 limit, then truncation occurs; otherwise, it does not. If truncation is decided to be stochastic, the relative intersection fraction is computed as above, and is is then used as the probability for intersection.

Argument: Commands specifying when simulated faults that cross each other are truncated.
Repeatable: No

6.11.12.1 `<limit>`

Description: Limit for truncation
Argument: One. A decimal number in between 0 and 1 (both inclusive). Values close to 1 indicate very little truncation, while values close to 0 indicate truncation of all faults intersecting.
Default: 0.0
Repeatable: No

6.11.12.2 `<stochastic>`

Description: Indicator of whether the truncation should be decided by drawing stochastically or not.
Argument: One. 'yes' or 'no'.
Default: 'no'
Repeatable: No

6.11.12.3 `<original-faults-truncate>`

Description: Indicator of whether original faults always should truncate simulated faults regardless of the criterion mentioned above.
Argument: One. 'yes' or 'no'.
Default: 'no'
Repeatable: No

6.11.13 `<simulation-volume>`

Description: Specifies the volume that the fault are simulated within. If this command is not given, the whole bounding box specified in the structural model is used as simulation volume. The top and/or bottom of the volume can be given as maps. In that case the surfaces need to cover the lateral area defined by `<x-reference>`, `<x-length>`, `<y-reference>` and `<y-length>` and the top and bottom surface. Note that the simulation volume might be rotated around the z-axis.
Argument: Commands specifying the simulation volume.
Repeatable: No

6.11.13.1 `<x-reference>`

Description: The global x-coordinate of the south-western corner of the simulation volume
Argument: One.
Default: The x-coordinate of south-western corner of the bounding box.
Repeatable: No

6.11.13.2 `<x-length>`

Description: The length of the (possibly rotated) global edge of the simulation volume.
Argument: One. A positive number.
Default: The length of the edge of the bounding box

Repeatable: No

6.11.13.3 `<y-reference>`

Description: The global y-coordinate of the south-western corner of the simulation volume
Argument: One.
Default: The y-coordinate of south-western corner of the bounding box.
Repeatable: No

6.11.13.4 `<y-length>`

Description: The length of the (possibly rotated) global edge of the simulation volume normal to the edge given by `<x-length>`.
Argument: One. A positive number.
Default: The length of the corresponding edge of the bounding box
Repeatable: No

6.11.13.5 `<rotation-angle>`

Description: The azimuth rotation in degrees. That means the clockwise rotation from north.
Argument: One.
Default: 0.0
Repeatable: No

6.11.13.6 `<top>`

Description: The top surface of the simulation volume. This must be either a surface defined on a file or a constant depth (z) value.
Argument: Commands specifying the volume top.
Default: The top of the bounding box
Repeatable: No

6.11.13.6.1 `<surface-file>`

Description: File specifying the top surface. Can be on either of the allowed fileformats. Either this or `<constant>` must be given.
Argument: One. The filename.
Repeatable: No

6.11.13.6.2 `<constant>`

Description: Constant depth for the top surface. Either this or `<surface-file>` must be given.
Argument: One.
Repeatable: No

6.11.13.7 `<bottom>`

Description: The bottom surface of the simulation volume. This must be either a surface defined on a file or a constant depth (z) value.
Argument: Commands specifying the volume bottom.
Default: The bottom of the bounding box
Repeatable: No

6.11.13.7.1 `<surface-file>`

Description: File specifying the bottom surface. Can be on either of the allowed fileformats. Either this or `<constant>` must be given.

Argument: One. The file name.

Repeatable: No

6.11.13.7.2 `<constant>`

Description: Constant depth for the bottom surface. Either this or `<surface-file2>` must be given.

Argument: One.

Repeatable: No

6.11.13.8 `<vertical-buffer-size>`

Description: Defines the size of the vertical buffer above and below the simulation volume in order to avoid edge effects.

Argument: One. A positive number

Default: Automatically calculated by HAVANA.

Repeatable: No

6.11.14 `<displacement-settings>` **(necessary)**

Description: Parameters describing how the displacement fields for the secondary faults are generated

Argument: Commands specifying the displacment field

Repeatable: No

6.11.14.1 `<displacement>` **(necessary)**

Description: Parameters describing the fractal distribution used to draw the maximum displacement for a fault. The displacement of a fault is illustrated in Figure 6.3, and the distribution of these follow a truncated probability distribution like the one in Figure 6.4.

Argument: Commands specifying the fractal distribution

Repeatable: No

6.11.14.1.1 `<minimum>` **(necessary)**

Description: Minimum displacement

Argument: One. A non-negative number.

Repeatable: No.

6.11.14.1.2 `<maximum>` **(necessary)**

Description: Maximum displacement

Argument: One. A non-negative number larger than the minimum.

Repeatable: No.

6.11.14.1.3 `<fractal-dimension>` **(necessary)**

Description: The dimension of the fractal distribution
Argument: One. A decimal number
Repeatable: No.
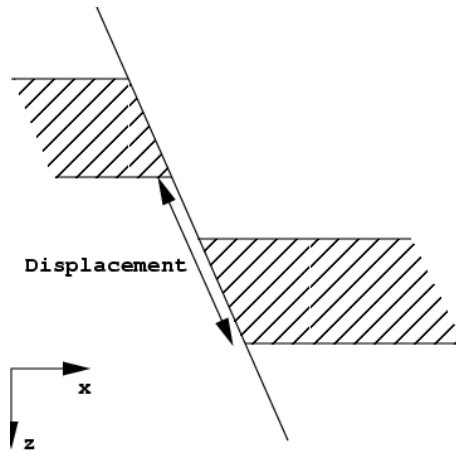
Figure 6.3. Illustration of measuring the displacement of a fault.
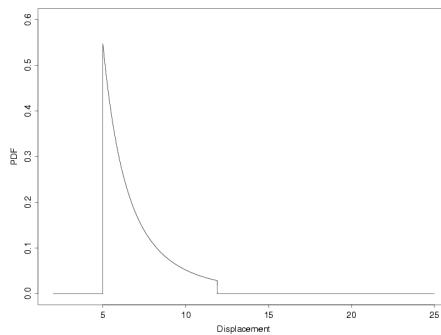
Figure 6.4. A truncated fractal probability density function, with fractal dimension $d = 2.4$ and range from 5 to 12 meters.

6.11.14.2 `<asymmetry>`

Description: Gives the distribution of the asymmetry factor of the displacement, that is how much of the displacement is taking place on the hanging wall side and how much on the foot wall side.
Argument: Commands specifying the probability distribution for the displacement asymmetry.
Repeatable: No

6.11.14.2.1 `<ratio>`

Description: Mean value for the displacement asymmetry in a normal distribution. An asymmetry of 1.0 means that all throw is on hanging wall, while 0.0 means all on foot wall.
Argument: One. A decimal number in interval 0.0 to 1.0.
Default: 0.5

Repeatable: No.

6.11.14.2.2 `<standard-deviation>`

Description: Standard deviation for the displacement asymmetry.
Argument: One. A non-negative decimal number
Default: 0.1
Repeatable: No.

6.11.14.3 `<max-displacement-length-relation>`
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

$$l = (d/c_1)^{1/p} V_1 \tag{6.1}$$

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.
Argument: Commands specifying the relationship between maximum displacement and fault length.
Repeatable: No

6.11.14.3.1 `<power>`

Description: The exponent p in expression 6.1
Argument: One. A non-negative decimal number
Default: 1.0
Repeatable: No.

6.11.14.3.2 `<ratio>`

Description: The constant c_1 in expression 6.1
Argument: One. A non-negative decimal number
Default: 0.01
Repeatable: No.

6.11.14.3.3 `<standard-deviation>`

Description: The standard deviation σ_1 in the normal distribution for $\log_e(V_1)$.
Argument: One. A non-negative decimal number.
Default: 0.1
Repeatable: No.

6.11.14.4 `<length-height-ratio>`
Description: Specifies parameters for the relationship between (maximum) fault length and (maximum) fault height. The fault length h is illustrated in Figure 6.5, 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

$$h = (l/c_2)V_2 \tag{6.2}$$

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.

Argument: Commands specifying the relationship between the fault length and height.
Repeatable: No

6.11.14.4.1 `<ratio>`

Description: The constant c_2 in expression 6.2
Argument: One. A non-negative decimal number
Default: 2.0
Repeatable: No.

6.11.14.4.2 `<standard-deviation>`

Description: The standard deviation σ_2 in the normal distribution for $\log_e(V_2)$.
Argument: One. A non-negative decimal number.
Default: 0.1
Repeatable: No.

6.11.14.5 `<range-length-height-ratio>`

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 6.5) is assumed to approximately follow the relationship $r \approx c_3\sqrt{lh}$ as a function of the fault length l and the fault height h. 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

$$r = (c_3)\sqrt{lh}V_3 \tag{6.3}$$

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.

Argument: Commands specifying the relationship between the fault length and height.
Repeatable: No

6.11.14.5.1 `<ratio>`

Description: The constant c_3 in expression 6.3
Argument: One. A non-negative decimal number
Default: 2.0
Repeatable: No.

6.11.14.5.2 `<standard-deviation>`

Description: The standard deviation σ_3 in the normal distribution for $\log_e(V_3)$.

Argument: One. A non-negative decimal number.

Default: 0.1
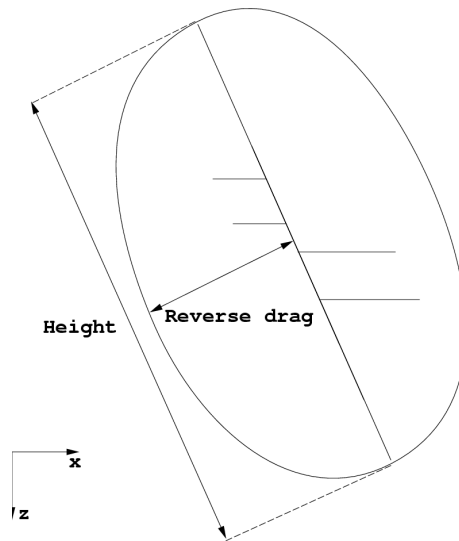
Repeatable: No.

Figure 6.5. Height and reverse drag of a fault.

6.11.15 `<output-attribute-name>`

Description: Specifies the attribute name given to the simulated faults.

Argument: One. An attribute name.

Repeatable: No.

6.11.16 `<output-group-name>`

Description: Specifies the group name given to the simulated faults.

Argument: One. A group name.

Repeatable: No.

6.11.17 `<simulation-settings>`

Description: Parameters specifying the position and orientation of the simulated faults around one, some or all primary faults. A certain fraction of the total simulated primary faults while be simulated according to these settings. An optional number of settings can be specified such that secondary faults around different primary faults follow different settings, or such that there exist different groups of secondary faults around the same primary faults. If no faults are specified under `<primary-fault-name>` this means that either all faults or all the faults that are not specified in another set of simulations settings has these settings.

Argument: Commands specifying the settings for a group of secondary faults

Repeatable: Yes.

6.11.17.1 `<primary-fault>`

Description: Information about the primary fault that the secondary faults with this specific settings are simulated around.

Argument: Commands specifying information about the primary fault.

Repeatable: Yes.

6.11.17.1.1 `<name>` **(necessary)**

Description: Name of the primary fault that the secondary faults with this specific settings are simulated around.

Argument: A fault name of a fault that exists in the fault set defined by `<input-havana-faults-directory>`.

Repeatable: No.

6.11.17.1.2 `<number-of-faults>`

Description: Number of simulated faults that will be assigned to the primary fault.

Argument: An integer number.

Repeatable: No.

6.11.17.2 `<primary-fault-attribute>`

Description: Information about the attribute of primary faults that the secondary faults with this specific settings are simulated around.

Argument: Commands specifying information about the primary fault attribute.

Repeatable: No.

6.11.17.3 `<primary-fault-group>`

Description: Information about the group of primary faults that the secondary faults with this specific settings are simulated around.

Argument: Commands specifying information about the primary fault group.

Repeatable: Yes.

6.11.17.3.1 `<name>` **(necessary)**

Description: Name of the primary fault group that the secondary faults with this specific settings are simulated around.

Argument: A name of group of faults that exist in the fault set defined by `<input-havana-faults-directory>`.

Repeatable: No.

6.11.17.4 `<position>`

Description: Parameters describing how the secondary faults are position relative to the primary fault they belong to. Note that the normal direction is not horizontal direction if the primary fault has s dip different from 90 degrees. Hence the relative intensity distribution does not only depend on the location of the primary fault surface, but also on the dip of it. Use the functionality (by command `<output-intensity-map>`)to output the calculated relative intensity for QC of your specification of the secondary faults' spatial distribution around the primary faults.

Argument: Commands specifying the position.

Repeatable: No.

6.11.17.4.1 `<strike-direction>`

Description: Parameters specifying how the center points of the primary faults is distributed along the strike direction of the primary fault.

Argument: Commands specifying the intensity.

Repeatable: No.

6.11.17.4.1.1 `<mean>`

Description: Mean position of the center point of the secondary fault, relative to the size of the primary fault. This means that a value of 0 gives the mean position at the center of the primary fault, -1 means at the fault tip of the primary fault in negative strike direction and 1 at the fault tip in positive strike direction.
Argument: One. A decimal number
Default: 0.0
Repeatable: No.

6.11.17.4.1.2 `<standard-deviation>`

Description: Standard deviation in the Gaussian intensity for the position of the center point of the secondary fault, relative to the size of the primary fault. A value of 0.5 means that almost all (95%) of the simulated faults will be placed within the length of the primary fault. Combined with a mean of zero it means that almost all secondary faults are place inside the fault tip line in strike direction. Be aware of the effect that small standard deviation in dip direction can create unexpected relative intensity if the primary fault has considerable dip.
Argument: One. A non-negative decimal number.
Default: 0.5
Repeatable: No.

6.11.17.4.2 `<dip-direction>`

Description: Parameters specifying how the center points of the primary faults is distributed along the dip direction of the primary fault.
Argument: Commands specifying the intensity.
Repeatable: No.

6.11.17.4.2.1 `<mean>`

Description: Mean position of the center point of the secondary fault, relative to the size of the primary fault. This means that a value of 0 gives the mean position at the center of the primary fault, -1 means at the fault tip of the primary fault in negative dip direction and 1 at the fault tip in positive dip direction.
Argument: One. A decimal number
Default: 0.0
Repeatable: No.

6.11.17.4.2.2 `<standard deviation>`

Description: Standard deviation in the gaussian intensity for the position of the center point of the secondary fault, relative to the size of the primary fault. The intensity is assumed symmetric around the mean. A value of 0.5 means that almost all (95%) of the simulated

faults will be placed within the height of the primary fault. Combined with a mean of zero it means that almost all secondary faults are place inside the fault tip line in dip direction.

Argument: One. A non-negative decimal number.

Default: 0.5

Repeatable: No.

6.11.17.4.3 `<normal-direction>`

Description: Parameters specifying how the center points of the primary faults is distributed along the direction normal to the fault plane of the primary fault.

Argument: Commands specifying the intensity.

Repeatable: No.

6.11.17.4.3.1 `<absolute-distance>`

Description: Decides if distance given for mean and standard deviation is relative to the primary faults influence range or in absolute distance. If 'no' -1 and 1 is at fault tip position, if 'yes' the values must be given in meters.

Argument: One. 'yes' or 'no

Default: 'no'

Repeatable: No.

6.11.17.4.3.2 `<mean>`

Description: Mean position of the center point of the secondary fault, relative to the size of the primary fault. If relative distance is used, this means that a value of 0 gives the mean position at fault surface of the primary fault, -1 means at the maximum range (reverse drag) of the primary fault in negative normal direction (foot wall) and 1 at the maximum range of the primary fault in positive direction. If absolute distance is used, it must be given in meters where the sign defines the side of the primary fault.

Argument: One. A decimal number

Default: 0.0

Repeatable: No.

6.11.17.4.3.3 `<standard deviation>`

Description: Standard deviation in the Gaussian intensity for the position of the center point of the secondary fault, relative to the size of the primary fault. The intensity is assumed symmetric around the mean. If relative distance is used, a value of 0.5 means that almost all (95%) of the simulated faults will be placed within twice the reverse drag distance of the primary fault. If absolute distance is used the standard deviation is specified in meters. Combined with a mean of zero it means that almost all secondary faults are place inside the range of the primary fault.

Argument: One. A non-negative decimal number.

Default: 0.5

Repeatable: No.

6.11.17.4.4 `<asymmetry-intensity>`

Description: The intensity of positioning the center point of the secondary faults on the hanging wall or foot wall side of the primary fault. A value of 0.5 means equal distribution, while higher value means higher intensity on the hanging wall side and lower value means higher intensity on the foot wall side.

Argument: One. A number within 0 to 1.

Default: 0.5

Repeatable: No.

6.11.17.5 `<dip>`

Description: The fault dip for the secondary fault is assumed to follow a normal distribution. If the drawn dip falls outside the interval $[0, 90]$ the strike is adjusted with $180 \, \text{deg}$.

Argument: Commands specifying the mean and standard deviation in the distribution

Repeatable: No.

6.11.17.5.1 `<mean>`

Description: The mean for the dip of the secondary fault, relative to the dip of the primary fault.

Argument: One. A decimal number.

Default: 0.0

Repeatable: No.

6.11.17.5.2 `<standard-deviation>`

Description: The standard deviation for the dip of the secondary fault, relative to the dip of the primary fault.

Argument: One. A non-negative decimal number.

Default: 1.0

Repeatable: No.

6.11.17.5.3 `<minimum>`

Description: The minimum value for the deviation of the dip of the secondary fault from the dip of the primary fault.

Argument: One. A decimal number.

Default: -1000000.0

Repeatable: No.

6.11.17.5.4 `<maximum>`

Description: The maximum value for the deviation of the dip of the secondary fault from the dip of the primary fault.

Argument: One. A decimal number.

Default: 1000000.0

Repeatable: No.

6.11.17.6 `<strike>`

Description: The fault strike for the secondary fault is assumed to follow a normal distribution.

Argument: Commands specifying the mean and standard deviation in the distribution

Repeatable: No.

6.11.17.6.1 `<mean>`

Description: The mean for the strike of the secondary fault, relative to the dip of the primary fault.
Argument: One. A decimal number.
Default: 0.0
Repeatable: No.

6.11.17.6.2 `<standard-deviation>`

Description: The standard deviation for the strike of the secondary fault, relative to the dip of the primary fault.
Argument: One. A non-negative decimal number.
Default: 10.0
Repeatable: No.

6.11.17.6.3 `<minimum>`

Description: The minimum value for the deviation of the strike of the secondary fault from the strike of the primary fault.
Argument: One. A decimal number.
Default: -1000000.0
Repeatable: No.

6.11.17.6.4 `<maximum>`

Description: The maximum value for the deviation of the strike of the secondary fault from the strike of the primary fault.
Argument: One. A decimal number.
Default: 1000000.0
Repeatable: No.

6.11.17.7 `<fraction-of-simulated-faults>`

Description: The fraction of secondary faults to be simulated using this specific set of simulation settings. The numbers for all sets will be scaled if their sum is different from one.
Argument: One. A positive decimal number
Default: 1.0
Repeatable: No.

6.11.17.8 `<displacement-weight>`

Description: The parameter specified here is an exponent for the fault displacements when these are used to determine which primary fault a secondary fault is belonging to. The probability for the ith primary faultwith displacement d_i to be selected as the primary fault for a new fault is originally $d_i / \sum d_i$. This implies that very large faults easily will become primary faults for too many secondary 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 then 1 increase the significance of the displacement in this relationship, while values smaller than 1 gives more
Argument: One. A decimal number.

Default: 1.0
Repeatable: No.

6.11.17.9 `<probability-antithetic-fault>`

Description: The probability for simulating an antithetic fault. If drawn fault is antithetic, 180 degrees is added to strike value.

Argument: One. Decimal number between 0.0 and 1.0.

Default: 0.0

Repeatable: No.

6.12 `<simulate-faults>`

This action is used to simulate faults located according to a specified intensity.

Example:

```
<simulate-faults>
    <input-havana-faults-directory> origFaultSet </input-havana-faults-directory>
    <output-havana-faults-directory> modifiedFaultSet </output-havana-faults-directory>
    <output-rms-faults-points-directory> surfaceFaults </output-rms-faults-points-directory>
    <seed> seed.txt </seed>
    <output-fault-summary>
        <file-name>fault_statistics </file-name>
        <report-truncated-length> yes </report-truncated-length>
        <report-truncating-faults> no </report-truncating-faults>
    </output-fault-summary>

    <output-faultname-prefix> subseismic </output-faultname-prefix>

    <number-of-simulated-faults> 150 </number-of-simulated-faults>

    <surface-grid-resolution> 10 </surface-grid-resolution>

    <fault-truncation>
        <limit> 0.0 </limit>
        <stochastic> no </stochastic>
        <original-faults-truncate> no </original-faults-truncate>
    </fault-truncation>

    <simulation-volume>
        <x-reference> 0.0 </x-reference>
        <x-length> 1000.0 </x-length>
        <y-reference> 0.0 </y-reference>
        <y-length>1000.0 </y-length>
        <rotation-angle> 0.0 </rotation-angle>
        <top>
            <surface-file> top.irap </surface-file>
        </top>
        <bottom>
            <surface-file> base.irap </surface-file>
        </bottom>
        <vertical-buffer-size> 30.0 </vertical-buffer-size>
    </simulation-volume>

    <displacement-settings>
        <displacement>
            <minimum> 10.0 </minimum>
            <maximum> 30.0 </maximum>
            <fractal-dimension> 2.0 </fractal-dimension>
        </displacement>
        <asymmetry>
```

```
            <ratio> 0.5 </ratio>
            <standard-deviation> 0.1 </standard-deviation>
        </asymmetry>
        <max-displacement-length-relation>
            <power> 1.0 </power>
            <ratio> 0.01 </ratio>
            <standard-deviation> 0.1 </standard-deviation>
        </max-displacement-length-relation>
        <length-height-ratio>
            <ratio> 2.0 </ratio>
            <standard-deviation> 0.1 </standard-deviation>
        </length-height-ratio>
        <range-length-height-ratio>
            <ratio> 0.4 </ratio>
            <standard-deviation> 0.1 </standard-deviation>
        </range-length-height-ratio>
</displacement-settings>


<relative-intensity>
    <type> Trend2D </type>
    <file-name> trend.irap </file-name>
    <nx> 50 </nx>
    <ny> 50 </ny>
</relative-intensity>


<orientation>
    <fraction> 1.0 </fraction>
    <strike>
        <mean>
            <type> Constant </type>
            <value> 10.0 </value>
        </mean>
        <standard-deviation> 0.5 </standard-deviation>
        <minimum> 8.0 </minimum>
        <maximum> 15.0 </maximum>
    </strike>
    <dip>
        <mean>
            <type> Trend2D </type>
            <file-name> dip.irap </file-name>
        </mean>
        <standard-deviation> 2.0 </standard-deviation>
        <minimum> 0.0 </minimum>
        <maximum> 60.0 </maximum>
    </dip>
    <probability-normal-fault> 1.0 </probability-normal-fault>
    <probability-antithetic-fault> 0.5 </probability-antithetic-fault>
</orientation>
```

```
        <repulsion>
            <range> 2000 </range>
            <potential> -1.0 </potential>
        </repulsion>

        <group-name> group1 </group-name>

        <number-of-iterations> 5000 </number-of-iterations>
    </simulate-faults>
```

6.12.1 `<input-havana-faults-directory>`
See 6.3.1. In case this command is not given it means that new faults are simulated without any pre-existing faults.

6.12.2 `<input-havana-structural-model-directory>`
See 6.11.2.

6.12.3 `<output-havana-faults-directory>`
See 6.2.2.

6.12.4 `<output-havana-structural-model-directory>`
See 6.11.4.

6.12.5 `<output-rms-faults-points-directory>`
See 6.4.3.

6.12.6 `<seed>` **(necessary)**
See 6.5.4.4

6.12.7 `<output-fault-summary>`
See 6.11.8.

6.12.8 `<output-faultname-prefix>`
See 6.11.9

6.12.9 `<number-of-simulated-faults>` **(necessary)**
See 6.11.10.

6.12.10 `<surface-grid-resolution>`
See 6.11.11.

6.12.11 `<fault-truncation>` **(necessary)**
See 6.11.12.

6.12.12 `<simulation-volume>`
See 6.11.13. In case the command `<input-havana-faults-directory>` is not given, it means that there is no default bounding box. In that case this command must be given and all sub-commands except `<rotation-angle>` and `<vertical-buffer-size>` must be given.

6.12.13 `<displacement-settings>` **(necessary)**
See 6.11.14

6.12.14 `<relative-intensity>`
Description:
Argument: Commands specifying the relative intensity of faults
Repeatable: No.

6.12.14.1 `<type>`
Description: Specify which type of relative intensity to use. There are two options; constant, and
a trend given as a surface.
Argument: One of the strings "Constant", "Trend2D".
Default: "Constant"
Repeatable: No.

6.12.14.2 `<file-name>`
Description: Name of file containing intensity grid. In this grid has missing values at all four
corner points for a position, the intensity is set to zero here.
Argument: string
Repeatable: No.

6.12.14.3 `<nx>`
Description: Grid resolution in x direction when calculating relative intensity. Should only be
used if type is "Trend2D". The grid resolution is recommended to be of roughly the same
order or finer than the grid resolution of the input trend map to capture he details of the
input trend.
Argument: integer > 0.
Default: 50
Repeatable: No.

6.12.14.4 `<ny>`
Description: Grid resolution in y direction when calculating relative intensity. Should only be
used if type is "Trend2D". The grid resolution is recommended to be of roughly the same
order or finer than the grid resolution of the input trend map to capture he details of the
input trend.
Argument: integer > 0.
Default: 50
Repeatable: No.

6.12.15 `<orientation>` **(necessary)**
Description: Specifies parameters for strike, dip, probability for normal fault and probability for
antithetic fault. The command is repeatable, meaning that we can specify groups of faults
with different orientation. The fraction of each group is given by the command "fraction".
Argument:
Repeatable: Yes.

6.12.15.1 `<fraction>`
Description: Fraction of faults using these orientation parameters.
Argument: Double > 0.0.

Repeatable: No.

6.12.15.2 `<strike>`
Description: The fault strike for the fault is assumed to follow a (truncated) normal distribution.
Argument: Commands specifying the mean, standard deviation, minimum and maximum in the distribution.
Repeatable: No.

6.12.15.2.1 `<mean>`
Description: The mean for the strike of the fault, given as a constant value or as a 2D trend.
Argument:
Default:
Repeatable: No.

6.12.15.2.1.1 `<type>`
Description: Specify which type of mean to use. There are two options; constant and a trend given as a surface.
Argument: One of the strings "Constant" or "Trend2D".
Default: "Constant"
Repeatable: No.

6.12.15.2.1.2 `<file-name>`
Description: Name of file containing intensity grid if type is given as 2DTrend. If the grid has missing values at all four corner points for a position, the trend is set to zero here.
Argument: string
Repeatable: No.

6.12.15.2.1.3 `<value>`
Description: Constant mean value if type is given as "Constant".
Argument: double
Default: 0.0
Repeatable: No.

6.12.15.2.2 `<standard-deviation>`
Description: The standard deviation for the strike of the fault.
Argument: One. A non-negative decimal number.
Default: 10.0
Repeatable: No.

6.12.15.2.3 `<minimum>`
Description: The lower truncation value for the strike of the fault.
Argument: One. A decimal number.
Default: -1000000.0
Repeatable: No.

6.12.15.2.4 `<maximum>`

Description: The upper truncation value for the strike of the fault.
Argument: One. A decimal number.
Default: 1000000.0
Repeatable: No.

6.12.15.3 `<dip>`
Description: The fault dip for the fault is assumed to follow a (truncated) normal distribution.
Argument: Commands specifying the mean, standard deviation, minimum and maximum in the distribution
Repeatable: No

6.12.15.3.1 `<mean>`

Description: The mean for the dip of the fault, given as a constant value or as a 2D trend. Same arguments as for mean strike.
Argument:
Default:
Repeatable: No.

6.12.15.3.2 `<standard-deviation>`

Description: The standard deviation for the dip of the fault.
Argument: One. A non-negative decimal number.
Default: 10.0
Repeatable: No.

6.12.15.3.3 `<minimum>`

Description: The lower truncation value for the dip of the fault.
Argument: One. A decimal number.
Default: -1000000.0
Repeatable: No.

6.12.15.3.4 `<maximum>`

Description: The upper truncation value for the dip of the fault.
Argument: One. A decimal number.
Default: 1000000.0
Repeatable: No.

6.12.15.4 `<probability-normal-fault>`
Description: The probability for simulating a normal fault, and not a reverse.
Argument: One. Decimal number between 0.0 and 1.0.
Default: 1.0
Repeatable: No.

6.12.15.5 `<probability-antithetic-fault>`
See 6.11.17.9.

6.12.16 `<repulsion>`

Description: Parameters describing the pairwise repulsion between faults. This command cannot be used if there are more than one group of orientations since the repulsion feature tries to align simulated faults that are close to each other, and this will be in conflict with a specification that the user want to create multiple groups of faults with different orientations. See Section 4.11.3 for a description on how this repulsion is included in the simulation algorithm.

Argument:

Default: No interaction between faults.

Repeatable: No.

6.12.16.1 `<range>`

Description: Parameter describing the range of the pairwise repulsion between faults.

Argument: Decimal number > 0.0.

Default:

Repeatable: No.

6.12.16.2 `<potential>`

Description: Parameter describing the maximum negative interaction potential, indicating the strength of interaction. Large absolute value means strong repulsion.

Argument: Negative decimal number or zero.

Default: 0.0

Repeatable: No.

6.12.17 `<output-attribute-name>`

See 6.11.15.

6.12.18 `<output-group-name>`

See 6.11.16.

6.12.19 `<number-of-iterations>`

Description: Number of iterations used in the Metropolis-Hastings simulation algorithm. The number of iterations must be larger than the number of faults to be simulated and often many times larger. It is recommended to try alternative numbers of iterations and choose one where the accept rate printed on screen or log files tend to converge to a constant value. An accept rate of 1 indicates that there is no effect of using the repulsion and the repulsion distance and/or the potential should be increased.

Argument: Integer > 0.

Default: 1000

Repeatable: No

6.13 `<group-faults>`

This action is used to group faults into two groups by adding a group tag to existing faults. Faults can be grouped into a deterministic and a simulated group. Grouped by size of either length, area or displacement. It can also be grouped with a combination of deterministic and large simulated into one group, and small simulated into another. The number specifies the maximum number of faults that will be added to the large fault group.

If <group-by-faultset> and <group-by-size> with <number> is used, all the original faults will first be placed in the large group, then the group will be filled up with the largest faults meeting the criteria in <group-by-size> until the total number of faults matches <number>. The specified <number> is then the total number of faults that will be placed in the large group, original and simulated.

Example:

```
<group-faults>
    <input-havana-faults-directory> modifiedFaultSet </input-havana-faults-directory>
    <output-havana-faults-directory> FaultSetGrouped </output-havana-faults-directory>
    <name-large-faults-group> Small </name-large-faults-group>
    <name-small-faults-group> Large </name-small-faults-group>
    <group-by-faultset>
        <original-havana-faults-directory> origFaultSet </original-havana-faults-directory>
    </group-by-faultset>
    <group-by-size>
        <length> 100 </length>
        <!--<area> 200 </area>-->
        <!--<displacement> 30 </displacement>-->
        <number> 0 </number>
    </group-by-size>
</group-faults>
```

6.13.1 `<input-havana-faults-directory>` **(necessary)**
See 6.3.1.

6.13.2 `<output-havana-faults-directory>` **(necessary)**
See 6.2.2.

6.13.3 `<attribute>`
Description: Specifies the output attribute name of the faults grouped in this action.
Argument: One. The attribute name.
Repeatable: No.

6.13.4 `<name-large-faults-group>` **(necessary)**
Description: Specifies the name to tag the large group of faults. Deterministic and large simulated faults can be grouped here.
Argument: One. The name.
Default:
Repeatable: No

6.13.5 `<name-small-faults-group>` **(necessary)**
Description: Specifies the name to tag the small group of faults.

Argument: One. The name.
Default:
Repeatable: No

6.13.6 `<group-by-faultset>`

Description: Groups the faults into a deterministic group and a simulated group, by comparing the input havana faults with a given original havana faults.
Repeatable: No

6.13.6.1 `<original-havana-faults-directory>`

Description: Original fault set. This is the fault set before running simulation of sub-seismic faults, and the faults that will be grouped in the large group of faults.
Argument: One. The directory.
Default:
Repeatable: No

6.13.7 `<group-by-size>`

Description: Group the faults by size. Only one of the three keywords can be given. If `<group-by-faultset>` is given the simulated faults that meets the criteria is grouped in the large faults group with the deterministic faults. If it is not given, all faults that meet the criteria is grouped in the large fault group. If a number is given, the large group faults is restricted to the number of all largest faults that meets the criteria.
Repeatable: No

6.13.7.1 `<length>`

Description: Length of the fault as criteria.
Argument: One. A positive number.
Default:
Repeatable: No

6.13.7.2 `<area>`

Description: Area of the fault as criteria.
Argument: One. A positive number.
Default:
Repeatable: No

6.13.7.3 `<displacement>`

Description: Max displacement of the fault as criteria.
Argument: One. A positive number.
Default:
Repeatable: No

6.13.7.4 `<number>`

Description: The total number of faults to be grouped in the large faults group. The number must be larger than the number of deterministic faults if `<group-by-faultset>` is used.
Argument: One. A positive number.
Default: 0. No restriction.
Repeatable: No

6.14 `<update-points>`

This action is used to update points after modification of the fault set. Horizons and fault lines can be updated by specifying the command `<update-structural-model>`. All horizons and/or fault lines can be updated, or only a selected subset. The output will be given on RMS internal point format (see Appendix A.10) and will be tagged with updated smart-fault-tag.

Selected point set (e.g. horizons and/or fault line point set) can be updated by specifying these under command `<update-pointset>`. However, will not be tagged with smart-fault-tag.

Both `<update-structural-model>` and `<update-pointset>` can be given in same model file.

Example:

```
<update-points>
    <input-original-havana-faults-directory> oFS </input-original-havana-faults-directory>
    <input-modified-havana-faults-directory> mFS </input-modified-havana-faults-directory>
    <update-structural-model>
        <input-structural-model-directory> StructModel </input-structural-model-directory>
        <update-horizons>
            <output-directory> horizons </output-directory>
            <horizon-name> topA </horizon-name>
            <horizon-name> topB </horizon-name>
        </update-horizons>
        <update-faultlines>
            <output-directory> faultLines </output-directory>
            <print-only-faultlines-from-changed-faults> yes
                </print-only-faultlines-from-changed-faults>
            <horizon-name> topA </horizon-name>
            <horizon-name> topB </horizon-name>
        </update-faultlines>
    </update-structural-model>
    <update-pointset>
        <input-file> TopA.rms </input-file>
        <input-file> TopB.rms </input-file>
        <output-directory> pointSet </output-directory>
    </update-pointset>
    <filter-settings>
        <filter-distance> 10 </filter-distance>
        <filter-points-crossing-faults> yes </filter-points-crossing-faults>
        <filter-inactive-horizon-points> yes </filter-inactive-horizon-points>
        <filter-eroded-faultline-points> yes </filter-eroded-faultline-points>
        <filter-from-wells>
            <well-data-file> well </well-data-file>
            <minimum-filtering-distance> 10 </minimum-filtering-distance>
            <filter-points-on-opposite-side-of-well-point> yes
                </filter-points-on-opposite-side-of-well-point>
        </filter-from-wells>
    </filter-settings>
</update-points>
```

6.14.1 `<input-original-havana-faults-directory>`

Description: Specifies the name of the directory where the original faults (unchanged fault set) on the Havana internal format are read from (see Appendix A.10)

Argument: One. The directory.

Default:

Repeatable: No

6.14.2 `<input-havana-structural-model-directory>`

See 6.11.2. Necessary if `<input-original-havana-faults-directory>` is not given.

6.14.3 `<input-modified-havana-faults-directory>` **(necessary)**

Description: Specifies the name of the directory where the modified faults (changed fault set) on the Havana internal format are read from (see Appendix A.10)

Argument: One. The directory.

Default:

Repeatable: No

6.14.4 `<update-structural-model>`

Description: Specifies the input and output directories required for updating the horizons and fault lines in the structural model. Horizons and/or fault lines can be updated. This is decided by specifying the commands (`<update-horizons>` and `<update-faultlines>`) and all or only a subset of these point sets can be chosen to be updated. At least one of the two commands (`<update-horizons>` and `<update-faultlines>`) must be given. If only fault lines from changed faults should be written, the command `<print-only-faultlines-from-changed-faults>` can be given.

Argument: Commands specifying the input and output directories.

Repeatable: No

6.14.4.1 `<input-structural-model-directory>`

See 6.2.1. Necessary when using `<input-havana-structural-model-directory>`.

6.14.5 `<output-havana-structural-model-directory>`

See 6.11.4. No effect if `<input-havana-structural-model-directory>` is not given.

6.14.5.1 `<update-horizons>`

Description: Specifies the input required for updating the horizons of the structural model. The `<output-directory>` command is necessary to update horizons. As default, all horizons are updated. However, by specifying one or more horizon names under `<horizon-name>`, only the subset of specified horizons will be updated.

Argument: Commands specifying output directory and subset of horizons to be updated.

Repeatable: No

6.14.5.1.1 `<output-directory>` **(necessary)**

Description: Directory to write the output horizon files.

Argument: One. Name of directory.

Default:

Repeatable: No

6.14.5.1.2 `<horizon-name>`

Description: Name of horizon to be updated if only a subset of horizons should be updated. If this command is not given, all horizons will be updated.

Argument: One. Name of horizon.

Default:

Repeatable: Yes

6.14.5.2 `<update-faultlines>`

Description: Specifies the input required for updating the fault lines of the structural model. The `<output-directory>` command is necessary to update fault lines. As default, all fault lines are updated. However, by specifying one or more horizon names under `<horizon-name2>`, only the subset of specified fault lines will be updated.

Argument: Commands specifying output directory and subset of horizons to which fault lines to be updated.

Repeatable: No

6.14.5.2.1 `<output-directory>` **(necessary)**

Description: Directory to write the output fault line files.

Argument: One. Name of directory.

Default:

Repeatable: No

6.14.5.2.2 `<horizon-name>`

Description: Fault lines belonging to the specified horizon name is updated. This command is used if the user don't want to update fault lines for all horizons, but only a subset of horizons. If this command is not given, all fault lines will be updated.

Argument: One. Name of horizon.

Default:

Repeatable: Yes

6.14.5.2.3 `<print-only-faultlines-from-changed-faults>`

Description: Specifies whether only fault lines from changed faults should be written to file. If 'yes' only changed faults are written.

Argument: One. Yes or no.

Default: No

Repeatable: No

6.14.6 `<update-pointset>`

Description: Specifies the input files of the point set to be updated and the output directory.

Argument: Commands specifying the input files and output directory.

Repeatable: No

6.14.6.1 `<input-file>` **(necessary)**

Description: File with point set to be updated. The point set can be on Roxar Text format (see Appendix A.8) or RMS internal point format (see Appendix A.10). The command must be repeated for each point set to be updated.

Argument: One. The filename.

Default:
Repeatable: Yes

6.14.6.2   `<output-directory>` **(necessary)**
Description: Directory to write the updated point set.
Argument:
Default:
Repeatable: No

6.14.7 `<update-node-file>`
Description: Specifies the input file for update of file for use with Statoil UpdatePillarsByKriging program.
Argument: Commands specifying the input and output files.
Repeatable: No

6.14.7.1   `<input-file>` **(necessary)**
Description: Specifies the input file with fault pillar nodes that should be updated.
Argument: One. The filename.
Default:
Repeatable: No

6.14.7.2   `<output-file>` **(necessary)**
Description: Name of output file which specifies how much each node is moved. The file format is the same as for the input file.
Argument: One. The filename.
Default:
Repeatable: No

6.14.8 `<filter-settings>`
Description: Specifies the filter settings.
Argument: Commands specifying filter settings.
Repeatable: No

6.14.8.1   `<filter-distance>`
Description: This command gives a custom filtering distance for horizon and fault line points. Sometimes RMS interpret points to be on the different side as they are interpreted in Havana. This filtering removes the points that are so near a fault surface that this can be a problem.
Argument: One. A positive number.
Default: 1.0
Repeatable: No

6.14.8.2   `<filter-points-crossing-faults>`
Description: Specifies whether points crossing faults when moving horizons should be filtered or not. If 'yes' these points are filtered.
Argument: One. Yes or no.
Default: Yes
Repeatable: No

6.14.8.3 `<filter-inactive-horizon-points>`

Description: Specifies whether horizon points that are tagged as inactive in input should be filtered or not. If 'no' these points will be updated and written to file.

Argument: One. Yes or no.

Default: Yes

Repeatable: No

6.14.8.4 `<filter-eroded-faultline-points>`

Description: Specifies whether fault line points that are tagged as eroded in input should be filtered or not. If 'no' these points will updated and written to file.

Argument: One. Yes or no.

Default: Yes

Repeatable: No

6.14.8.5 `<filter-from-wells>`

Description: Fault line points close to well points are filtered if they are on the same side and within a certain distance from a well point. Optionally, also points on opposite side can be filtered.

Argument: Commands specifying the input required for filtering from wells.

Repeatable: No

6.14.8.5.1 `<well-data-file>` **(necessary)**

Description: Input file with well data. The file is a plain ascii text file, which can be exported from RMS. Each line contains first two strings which are not used, and then x, y and z coordinates of a point.

Argument: One. The filename.

Default:

Repeatable: No

6.14.8.5.2 `<minimum-filtering-distance>` **(necessary)**

Description: Specifies the minimum distance for filtering.

Argument: One. A non-negative number.

Default:

Repeatable: No

6.14.8.5.3 `<filter-points-on-opposite-side-of-well-point>`

Description: Specifies whether points on opposite side of the well point should be filtered or not.

Argument: One. Yes or no.

Default: No

Repeatable: No

6.15 `<update-grid>`

This action is used to update the eclipse grid according to the differences between the modified and original fault set. The changes that are supported are new faults that are added, removed faults and faults with changed displacement field. The fault surface is defined by grid pillar lines. The fault blocks affected by the new and removed faults are divided and merged, and the displacement is updated in all corner points according to the displacement field of the new fault set. The truncation rules are taken into account. The input 3D corner point grid must be consistent with the structural model that is represented by the original fault set.

Example:

```
<update-grid>
    <input-original-havana-faults-directory> oFS </input-original-havana-faults-directory>
    <input-modified-havana-faults-directory> mFS </input-modified-havana-faults-directory>
    <input-grid-file> emerald_structmodgrid1.grdecl </input-grid-file>
    <input-grid-file> fault_block_parameter.grdecl </input-grid-file>
    <output-grid-file> output.grdecl </output-grid-file>
    <input-structural-model-directory> StructuralModel </input-structural-model-directory>
    <group-name> Large </group-name>
</update-grid>
```

6.15.1 `<input-original-havana-faults-directory>` **(necessary)**
See 6.14.1.

6.15.2 `<input-modified-havana-faults-directory>` **(necessary)**
See 6.14.3.

6.15.3 `<input-grid-file>` **(necessary)**
Name of the input grid file. The grid parameter 'FaultBlock' is required in this action. This parameter can be included in the same grid file as the grid coordinates, or as a separate grid file. The grid file format is described in Appendix A.3.

Description: Specifies the name of the input eclipse grid.
Argument: One. The file name.
Default:
Repeatable: Yes

6.15.4 `<output-grid-file>` **(necessary)**
Description: Specifies the name of the output eclipse grid which is updated by adding new faults, removing existing faults or changing throw of existing faults.
Argument: One. The file name.
Default:
Repeatable: No

6.15.5 `<input-structural-model-directory>` **(necessary)**
See 6.2.1.

6.15.6 `<attribute>`
Description: Specifies the attribute name of which faults to use in the UpdateGrid action. The attribute name of the faults is set under `<group-faults>` or specified when simulating

faults.

Argument: One. The attribute name.

Repeatable: No.

6.15.7 `<group-name>`

Description: Specifies the group names of which faults to use in the UpdateGrid action. The
group name of the faults is set under `<group-faults>` or specified when simulating faults.

Argument: One. The group name.

Repeatable: yes.

6.16 `<update-permeability>`

There are two ways in HAVANA to study the impact of faults on production and flow properties. The first is to add the faults to the ECLIPSE grid (action `<update-grid>`, see 6.15), the second is to add them directly to some gridded permeability field representing the reservoir.

The choice between the two options may have a great impact on the result. Generally, faults that are larger than the ECLIPSE grid cells should be entered into the ECLIPSE grid. Both their displacement and their fault seal effect are then reasonably well represented. However, if the faults are equal or smaller in size than the ECLIPSE grid blocks, they may very well disappear completely. Such faults should be added to the permeability, using the action `<update-permeability>`. This module works reasonably for such faults, but faults that are greater than the ECLIPSE grid blocks may not be very well represented. Note in particular that if you add a fault to the grid by the action `<update-grid>` and also update permeability with the same fault, there will often be an inconsistency. The action `<update-grid>` have to approximate the fault surface by defining pillar lines to represent the fault surface. In the action `<update-permeability>` this is not any limitation since the effect is sampled into the correct location. Hence there will be a difference between these two representations of the fault. It is recommended to only choose one of the options, either add the fault to the grid or to the permeability, not both.

Example:

```
<update-permeability>
    <input-havana-faults-directory> origFaultSet </input-havana-faults-directory>
    <output-permeability-directory> OuputPerm </output-permeability-directory>
    <seed> 53467 </seed>
    <group-name> Small </group-name>
    <fault-permeability>
        <relative-to-neighborhood> no </relative-to-neighborhood>
        <distribution>
            <type> Constant </type>
            <mu> 0.1 </mu>
        </distribution>
    </fault-permeability>
    <fault-thickness>
        <multiply-max-displacement> yes </multiply-max-displacement>
        <distribution>
            <type> Normal </type>
            <mu> 0.01 </mu>
            <sigma> 0.005 </sigma>
        </distribution>
    </fault-thickness>
    <input-permeability>
        <eclipse-grid> perm_grid.grdecl </eclipse-grid>
        <subdivide>
            <nx> 3 </nx>
            <ny> 3 </ny>
            <nz> 3 </nz>
        </subdivide>
        <keywords>
            <permx> PERMX </permx>
```

```
                <permy> PERMY </permy>
                <permz> PERMZ </permz>
            </keywords>
        </input-permeability>
        <output-permeability>
            <averaging> arit-harm </averaging>
            <displacement> yes </displacement>
        </output-permeability>
    </update-permeability>
```

6.16.1 `<input-havana-faults-directory>` **(necessary)**

See 6.14.1.

6.16.2 `<output-permeability-directory>`

Description: Specifies the directory where PERMX, PERMY and PERMZ are written.

Argument: One. The name.

Repeatable: No.

6.16.3 `<seed>`

See 6.5.4.4.

6.16.4 `<attribute>`

Description: Specifies the attribute name of which faults to use in the update permeability action. The attribute name of the faults is set under `<group-faults>` or specified when simulating faults.

Argument: One. The attribute name.

Repeatable: No.

6.16.5 `<group-name>`

Description: Specifies the group names of which faults to use in the update permeability action. The group name of the faults is set under `<group-faults>` or specified when simulating faults.

Argument: One. The group name.

Repeatable: yes.

6.16.6 `<fault-permeability>` **(necessary)**

Description: Specifies the permeability or distribution of permeabilities for the fault planes.

Argument: Commands specifying permeability distribution.

Default:

Repeatable: No.

6.16.6.1   `<relative-to-neighborhood>`

Description: Specifies whether the permeability for the fault planes is relative to the surrounding rock permeability or absolute.

Argument: Yes/No.

Default: No.

Repeatable: No.

6.16.6.2 `<distribution>` **(necessary)**

Description: Specifies the probability distribution for permeability.
Argument: Commands specifying distribution type.
Default:
Repeatable: No.

6.16.6.2.1 `<type>` **(necessary)**

Description: Distribution type.
Argument: "Constant", "Normal" or "LogNormal". "Constant" means that the permeability is deterministically set to the specified value. The other two alternatives defines a stochastic distribution.
Default:
Repeatable: No.

6.16.6.2.2 `<mu>` **(necessary)**

Description: Parameter in distribution. Constant value for constant distribution, mean value for normal distribution and mean in corresponding normal distribution for log-normal.
Argument: A real number.
Default:
Repeatable: No.

6.16.6.2.3 `<sigma>`

Description: Parameter in distribution. Standard deviation for normal. distribution, standard deviation in corresponding normal distribution for log normal.
Argument: A non-negative real number.
Default:
Repeatable: No.

6.16.7 `<fault-thickness>`

Description: Specifies the fault thickness or distribution of fault thicknesses for the fault surface permeability effect. If this command is omitted, the fault thickness is assumed to be equal to the size of the blocks the fault goes through.
Argument: Commands specifying thickness distribution.
Default:
Repeatable: No.

6.16.7.1 `<multiply-max-displacement>`

Description: Specifies whether the drawn thickness should be multiplied with maximum displacement or not.
Argument: Yes/No.
Default: No.
Repeatable: No.

6.16.7.2 `<distribution>` **(necessary)**

See 6.16.6.2

6.16.8 `<input-permeability>` **(necessary)**

Description: Specifies the input eclipse grid with permeability values and possible refinement of the grid to include fault effects on permeabilities.

Argument: Commands specifying input permeability grid.

Default:

Repeatable: No.

6.16.8.1 `<eclipse-grid>` **(necessary)**

Description: Specifies the file name for the file containing the input ECLIPSE grid. Keywords that must appear in the ECLIPSE files are: SPECGRID, COORD and ZCORN. If no ACTNUM keyword is encountered, all blocks are assumed to be active. Note that the grid must also contain the permeabilities for PERMX, PERMY, PERMZ before the fault effect is modifying this. For more details on input of ECLIPSE data, see Appendix A.3.

Argument: File name.

Default:

Repeatable: No.

6.16.8.2 `<subdivide>`

Description: These are commands used to subdivide each cell of the input permeability grid the x-, y- and z-directions according to the specified refinement. The permeability value, equal to that of the whole cell, is assigned to each sub cell. This is a way to get a more accurate inclusion of faults. A trilinear approximation is used.

Argument: Commands giving the number of sub-intervals for each direction.

Default:

Repeatable: No.

6.16.8.2.1 `<nx>`

Description: Specifies subdivision of eclipse cells in x direction.

Argument: A positive integer value.

Default: 1

Repeatable: No.

6.16.8.2.2 `<ny>`

Description: Specifies subdivision of eclipse cells in y direction.

Argument: A positive integer value.

Default: 1

Repeatable: No.

6.16.8.2.3 `<nz>`

Description: Specifies subdivision of eclipse cells in z direction.

Argument: A positive integer value.

Default: 1

Repeatable: No.

6.16.8.3 `<keywords>`

Description: Specifies name of the permeability keywords in the ECLIPSE grid.

Argument: Commands for permeability keywords in x-, y- and z-direction.

Default:
Repeatable:　No.

6.16.8.3.1　`<permx>`

Description:　Specifies name of permx parameter in ECLIPSE grid.
Argument:　Name.
Default:　PERMX
Repeatable:　No.

6.16.8.3.2　`<permy>`

Description:　Specifies name of permy parameter in ECLIPSE grid.
Argument:　Name.
Default:　PERMY
Repeatable:　No.

6.16.8.3.3　`<permz>`

Description:　Specifies name of permz parameter in ECLIPSE grid.
Argument:　Name.
Default:　PERMZ
Repeatable:　No.

6.16.9 `<output-permeability>`

Description:　Specifies commands relevant for the resulting permeability ECLIPSE grid. If this
command is omitted, the same grid format as for the input grid is used with the same file
name, and all the following commands are given default values.
Argument:　Commands necessary to specify the homogenization method for updating of the
permeability based on faults.
Default:
Repeatable:　No.

6.16.9.1　`<eclipse-grid>`

See 6.16.8.1. If this command is not specified, the input grid is used as format for output, and
only permeability grids are written to files.

6.16.9.2　`<subdivide>`

See 6.16.8.2.

6.16.9.3　`<averaging>`

Description:　Specifies method for averaging. Two possible methods, arithmetic-harmonic or
harmonic-arithmetic.
Argument:　"arit-harm" or "harm-arit".
Default:　"harm-arit"
Repeatable:　No.

6.16.9.4　`<displacement>`

Description:　Specifies if displacement effect should be included. Either this or fault surface effect
or both should be included.

Argument: Yes/No.
Default: Yes.
Repeatable: No.

6.16.9.5 `<fault-surface-effect>`

Description: Specifies if fault surface effect should be included. Either this or displacement effect or both should be included.

Argument: Yes/No.
Default: Yes.
Repeatable: No.

6.17 `<create-fault-grid>`

This action is used to create local grid refinement used for 3D modelling of the fault zone. It is both possible to generate a grid that only consists of the fault zone grid or a grid consisting both of the coarse background grid and the locally refined fault zone grid.

Example:
```
<create-fault-grid>
    <input-grid-file> emerald_structmodgrid1.grdecl </input-grid-file>
    <output-grid>
        <grid-type> LOCAL </grid-type>
        <file-name> grid_LOCAL.grdecl </file-name>
        <keyword> SPECGRID </keyword>
        <keyword> ACTNUM</keyword>
        <keyword> PERMZ</keyword>
    </output-grid>
    <output-grid>
        <grid-type> GLOBAL </grid-type>
        <file-name> grid_GLOBAL.grdecl </file-name>
        <keyword> COORD </keyword>
    </output-grid>
    <refinement-settings>
        <x-factor> 3 </x-factor>
        <y-factor> 3 </y-factor>
        <z-factor> 2 </z-factor>
        <width-in-number-of-grid-cells> 2 </width-in-number-of-grid-cells>
    </refinement-settings>
</create-fault-grid>
```

6.17.1 `<input-grid-file>` **(necessary)**

Description: The ECLIPSE grid that is basis for the fault zone grid. This grid must contain the fault trace data (ECLIPSE `FAULTS` keyword), and fault block identifiers given by a grid parameter named `FaultBlock`. The format for this file is given in Appendix A.3. This parameter can be created in RMS from the 'Grid index parameters...' job under `Parameter utilities` in the context menu for the grid. The command must be repeated for each input grid.

Argument: One. Name of eclipse grid file.

Default:

Repeatable: Yes

6.17.2 `<output-grid>`

Description: Specifies the output ECLIPSE grid either containing the coarse grid given by the `<input-grid-file>` commands and the fault zone grid given as local grid refinement, or only containing the finescaled faultzone grid. The command must be repeated for each output grid.

Argument: Commands specifying the grid details.

Repeatable: Yes

6.17.2.1 `<grid-type>` **(necessary)**

Description: Specifies the output grid type. Available commands are:

GLOBAL: Eclipse file containing the coarse grid given by the `<input-grid-file>` commands and the fault zone grid given as local grid refinement.

LOCAL: Eclipse file only containing the finescaled faultzone grid.

Argument: One. The name of the command.

Default:

Repeatable: No

6.17.2.2 `<file-name>` **(necessary)**

Description: Specifies the name of the output eclipse grid.

Argument: One. The filename.

Default:

Repeatable: No

6.17.2.3 `<keyword>`

Description: Specifies an output keyword. The command must be repeated for each keyword to be included in this output grid. If no keywords are listed for the last file, all remaining keywords are written to this file. The available keywords are: SPECGRID, COORD, ZCORN, ACTNUM, PERMX, PERMY, PERMZ, CARFIN and ACTNUM.

Argument: One. Name of keyword.

Default:

Repeatable: Yes

6.17.3 `<refinement-settings>` **(necessary)**

Description: Specifies the grid refinement settings.

Argument: Commands specifying the grid refinements settings.

Repeatable: No

6.17.3.1 `<x-factor>` **(necessary)**

Description: The refinement factor in x-direction.

Argument: One. A positive integer.

Default:

Repeatable: No

6.17.3.2 `<y-factor>` **(necessary)**

Description: The refinement factor in y-direction.

Argument: One. A positive integer.

Default:

Repeatable: No

6.17.3.3 `<z-factor>` **(necessary)**

Description: The refinement factor in z-direction.

Argument: One. A positive integer.

Default:

Repeatable: No

6.17.3.4 `<width-in-number-of-grid-cells>` **(necessary)**

Description: Specifies how many cells on each side of the fault trace that is refined.

Argument: One. Number of cells.

Default:

Repeatable: No

6.18 `<merge-fault-grid>`

This action is used to merge a fine-scale fault zone grid with a coarse-scale host grid. The fault zone grid must be created using the action `<create-fault-grid>`. The fault zone grid is added to the coarse grid as a set of local grid refinements. The file format for the Eclipse grid files used here is given in Appendix A.3.

Example:

```
<merge-fault-grid>
    <input-global-grid-file> grid_file.grdecl </input-global-grid-file>
    <input-global-grid-file> poro.grdecl </input-global-grid-file>
    <input-local-grid-file> grid_file2.grdecl </input-local-grid-file>
    <input-local-grid-file> perm.grdecl </input-local-grid-file>
    <output-merged-grid>
        <file-name> eclipseout.grdecl </file-name>
        <keyword> PERMY </keyword>
        <keyword> CARFIN </keyword>
    </output-merged-grid>
    <output-merged-grid>
        <file-name> eclipseout2.grdecl </file-name>
            <keyword> SPECGRID </keyword>
            <keyword> COORD </keyword>
            <keyword> PERMX </keyword>
    </output-merged-grid>
    <local-grid-name> fzg </local-grid-name>
</merge-fault-grid>
```

6.18.1 `<input-global-grid-file>` (necessary)

Description: Specifies the original coarse-scale host grid that the fault zone grid shall be merged with. The command must be repeated for each input global grid.

Argument: One. Name of eclipse grid file.

Default:

Repeatable: Yes

6.18.2 `<input-local-grid-file>` (necessary)

Description: Specifies the fine-scaled local fault zone grid generated by the `<create-fault-grid>` ACTION. The command must be repeated for each input local grid.

Argument: One. Name of eclipse grid file.

Default:

Repeatable: Yes

6.18.3 `<output-merged-grid>` (necessary)

Description: Specifies the output ECLIPSE grid file where the merged grid shall be written to. The command must be repeated for each output grid file.

Argument: Commands specifying the output ECLIPSE grid.

Repeatable: No

6.18.3.1 `<file-name>` (necessary)

Description: Specifies the name of the output merged grid.

Argument: One. The file name.
Default:
Repeatable: No

6.18.3.2 `<keyword>` **(necessary)**
See 6.17.2.3.

6.18.4 `<local-grid-name>`
Description: Name used for local grid refinements created from fault zone grid.
Argument: One. The local grid refinement name.
Default: LGR
Repeatable: No

6.19 `<calculate-fault-seal>`

This action is calculating the fault seal parameters along the fault surface. The main output parameters are displacement, shale gouge ratio (SGR), fault permeability, fault thickness, and transmissibility multipliers. Optionally, shear smear factor (SSF), clay smear potensial (CSP) and probabilistic SSF (PSSF), can be computed and used when calculating the transmissibility multipliers (implementation in progress).

SSF is indicating the potential for seal breach while CSP is a measure of the length continuity of the clay smeared along the fault. If the calculated SSF at a point along the fault is less or equal than a critical value (SSF_c) the clay is assumed to seal off the point and a transmissibility multiplier of zero is returned. If the calculated CSP at a point along the fault is greater than a critical value (CSP_c) the length continuity of the clay smeared is assumed to be sufficient to seal off the point and a transmissibility multiplier of zero is returned. For points with $SSF > SSF_c$ or $CSP \leq CSP_c$ the transmissibility multiplier is estimated using the fault zone permeability that is calculated from the the estimated SGR (or ESGR).

Required input data is an RMS structural model of the faults and a corresponding Eclipse grid with the grid parameters permx, permy, permz. Parameters indicating the clay/shale content of the grid cells is also required and optionally a parameter indicating lithology type can be used in SSF and CSP calculations to identify whether a cell is clay or not. The computed transmissibility multipliers of the faults are exported to a new Eclipse grid in .grdecl format.

Example:

```
<calculate-fault-seal>
    <input-havana-faults-directory> origFaultSet </input-havana-faults-directory>
    <input-structural-model-directory> structuralmodel </input-structural-model-directory>
    <input-grid-file> eclipsegrid.grdecl </input-grid-file>
    <grid-parameters>
        <permx> PERMX </permx>
        <permy> PERMY </permy>
        <permz> PERMZ </permz>
    </grid-parameters>
    <output-grid-file> eclipsegrid_out.grdecl </output-grid-file>

    <calculate-on-fault> F1 </calculate-on-fault>
    <calculate-on-fault> F4 </calculate-on-fault>
    <calculate-on-fault> F6 </calculate-on-fault>
    <output-fault-attributes> true </output-fault-attributes>

    <shale-gauge-method> SGR </shale-gauge-method>
    <shale-gauge-parameters>
        <clay-parameter> VShale </clay-parameter>
    </shale-gauge-parameters>

    <clay-smear-method> SSF </clay-smear-method>
    <clay-smear-parameters>
        <smear-limit> 10 </smear-limit>
        <smear-calculation-method> minimum </smear-calculation-method>
        <continuous-clay-parameters>
            <clay-parameter> VShale </clay-parameter>
```

```
        <clay-parameter-cut-off> 0.5 </clay-parameter-cut-off>
      </continuous-clay-parameters>
    </clay-smear-parameters>


    <fault-permeability-method> Manzocchi </fault-permeability-method>
    <fault-thickness-method> ratio </fault-thickness-method>
    <fault-thickness-parameters>
        <ratio> 0.01 </ratio>
    </fault-thickness-parameters>
  </calculate-fault-seal>
```

6.19.1 `<input-havana-faults-directory>` **(necessary)**

Description: Specifies the directory containing the original Havana faults. These faults must be written in the internal Havana format and is generated by the `<import-rms-fault-data>` action.

Argument: The directory

Repeatable: No

6.19.2 `<input-structural-model-directory>` **(necessary)**

Description: Specifies the name of the directory containing the fault files from the RMS export. See `<import-rms-fault-data>` action for further details.

Argument: The directory

Repeatable: No

6.19.3 `<input-grid-file>` **(necessary)**

Description: The ECLIPSE grid that is basis for the fault grid. This grid must contain the fault trace data (ECLIPSE `FAULTS` keyword), fault block identifiers given by a grid parameter named `FaultBlock`, absolute permeability properties (e.g. ECLIPSE keywords `PERMX`, `PERMY`, and `PERMZ`) and volume of clay/shale from petrophysical modelling (e.g. `VShale`). These parameters can be created in RMS. The grid file format is described in Appendix A.3.

Argument: File name

Repeatable: No

6.19.4 `<grid-parameters>` **(necessary)**

Description: List the keywords in the ECLIPSE grid file that is holding the permeability values in each grid cell.

Repeatable: No

6.19.4.1 `<permx>`

Description: Specifies the name of the ECLIPSE grid parameter holding the absolute permeability in X direction.

Argument: The parameter name

Default: PERMX

Repeatable: No

6.19.4.2 `<permy>`

Description: Specifies the name of the ECLIPSE grid parameter holding the absolute permeability in Y direction.

Argument: The parameter name

Default: PERMY
Repeatable: No

6.19.4.3 `<permz>`
Description: Specifies the name of the ECLIPSE grid parameter holding the absolute permeability in Z direction.
Argument: The parameter name
Default: PERMZ
Repeatable: No

6.19.5 `<output-grid-file>` **(necessary)**
Description: Specifies the name of the output ECLIPSE grid which is updated with the computed fault transmissibility multipliers.
Argument: File name
Repeatable: No

6.19.6 `<calculate-on-fault>` **(necessary)**
Description: Specifies the name of fault to run fault seal calculations on. The fault name is according to the original Havana fault names.
Argument: Fault name
Repeatable: Yes

6.19.7 `<output-fault-attributes>`
Description: Specifies if computed fault attributes like SGR, permeability, transmissibility etc. should be saved to files using RMS attributes text format. The data can be visualized in RMS as point set in the 3D viewer. Each fault is associated with one output file with file name <fault name>_ attributes.txt
Argument: Yes/No
Default: No
Repeatable: No

6.19.8 `<shale-gouge-method>` **(necessary)**
Description: Specifies whether shale gouge ratio (SGR) or effective SGR (ESGR) should be used when computing the permeability of the fault zone. Currently, only SGR is implemented.
Argument: SGR or ESGR
Default: SGR
Repeatable: No

6.19.9 `<shale-gouge-parameters>` **(necessary)**
Description: Holds the keywords used to identify the clay content of the grid cells and weight distribution (if ESGR).
Repeatable: No

6.19.9.1 `<clay-parameter>` **(necessary)**
Description: Specifies the name of the ECLIPSE grid parameter holding the clay/shale content of each grid cell.
Argument: The parameter name
Repeatable: No

6.19.9.2 `<esgr-weight-settings>`

Description: Specifies the weight settings if ESGR is selected in `<shale-gouge-method>`.

Repeatable: No

6.19.9.2.1 `<std-dev-factor>`

Description: Specifies the standard deviation σ of the normal distribution used to compute the weights in ESGR. σ is specified relative to the total fault displacement D. I.e. `<std-dev-factor>` specifies the factor $f = \sigma/D$ if ESGR is selected in `<shale-gouge-method>`.

Argument: Value

Default: 1.0

Repeatable: No

6.19.10 `<clay-smear-method>`

Description: Specifies the optional clay smear method to be used when computing transmissibilities. Currently supported methods are shale smear factor (SSF), clay smear potential (CSP) and probabilistic SSF (PSSF). Currently, PSSF is not implemented.

Argument: SSF, CSP or PSSF

Default: None

Repeatable: No

6.19.11 `<clay-smear-parameters>`

Description: Holding necessary key words defining the parameters of the clay smear method. Required when `<clay-smear-method>` is specified.

Repeatable: No

6.19.11.1 `<smear-limit>`

Description: The critical SSF or critical CSP depending on whether (P)SSF or CSP is selected in `<clay-smear-method>`

Argument: Value

Default: 10

Repeatable: No

6.19.11.2 `<smear-calculation-method>`

Description: The method that is used when combining SSF and CSP of individual clay layers along the fault displacement to compute the SSF and CSP of a point on the fault. Accepted values are MINIMUM and SUM, where minimum is selecting the minimium SSF (or maximum CSP) of the individual clay layers found. The sum refers to summing the contributions from the individual layers similarly to the RMS fault seal work flow (to be implemented).

Argument: MINIMUM or SUM

Default: MINIMUM

Repeatable: No

6.19.11.3 `<continuous-clay-parameters>`

Description: Holding necessary key words when a continuous clay parameters is used to filter grid cells into clay or non-clay cells. Either this key word or the key word `<discrete-facies-parameters>` is required when `<clay-smear-parameters>` is specified.

Repeatable: No

6.19.11.3.1 `<clay-parameter>`

Description: Specifies the name of the ECLIPSE grid parameter holding the clay/shale content of each grid cell. Note that this parameter can differ from the `<clay-parameter>` under the key word `<shale-gouge-parameters>`, but is often the same.

Argument: Parameter name

Repeatable: No

6.19.11.3.2 `<clay-parameter-cut-off>`

Description: Specifies the cut-off value used when filter grid cells into clay or non-clay cells. Cells with clay/shale content larger or equal to the provided value is identified as clay cells.

Argument: Value

Default: 0.50

Repeatable: No

6.19.11.4 `<discrete-facies-parameters>`

Description: Holding necessary key words when discrete facies are used to filter grid cells into clay or non-clay cells. Either this key word or the key word `<continuous-clay-parameters>` is required when `<clay-smear-parameters>` is specified.

Repeatable: No

6.19.11.4.1 `<facies-parameter>`

Description: Specifies the name of the ECLIPSE grid parameter holding the facies (or lithology) identification numbers (or similar) of each grid cell.

Argument: Parameter name

Repeatable: No

6.19.11.4.2 `<clay-facies>`

Description: Specifies the facies identification numbers (id) that should be considered as clay and their corresponding weights to be used when computing clay smear.

Repeatable: Yes

6.19.11.4.2.1 `<id>`

Description: Facies (or lithology) id. Grid cells with with given facies id are considered as clay cells.

Argument: Value

Repeatable: No

6.19.11.4.2.2 `<weight>`

Description: Weight of facies (or lithology). Increasing weight means increasing contribution to clay smear.

Argument: Value

Default: 1.0

Repeatable: No

6.19.12 `<fault-permeability-method>` **(necessary)**

Description: Specifies the permeability methods to be used when computing the fault permeability. Currently supported methods are MANZOCCHI, SPERREVIK and SGR_CURVE

Argument: Permeability method
Default: MANZOCCHI
Repeatable: No

6.19.13 `<sperrevik-parameters>`

Description: Specifies the parameters in the Sperrevik permeability relation [14] when the `<fault-permeability-method>` is set to SPERREVIK. The keyword is then necessary.

Repeatable: No

6.19.13.1 `<a1>`

Description: Specifies the parameter a_1 in the Sperrevik permeability relation.

Argument: Value
Default: 80000
Repeatable: No

6.19.13.2 `<a2>`

Description: Specifies the parameter a_2 in the Sperrevik permeability relation.

Argument: Value
Default: 19.4
Repeatable: No

6.19.13.3 `<a3>`

Description: Specifies the parameter a_3 in the Sperrevik permeability relation.

Argument: Value
Default: 0.00403
Repeatable: No

6.19.13.4 `<a4>`

Description: Specifies the parameter a_4 in the Sperrevik permeability relation.

Argument: Value
Default: 0.0055
Repeatable: No

6.19.13.5 `<a5>`

Description: Specifies the parameter a_5 in the Sperrevik permeability relation.

Argument: Value
Default: 12.5
Repeatable: No

6.19.13.6 `<zmax>`

Description: Specifies the maximum burial depth Z_{max} (in meters) of the faulted sediment in the Sperrevik permeability relation.

Argument: Value
Default: 3500
Repeatable: No

6.19.13.7 `<zf>`

Description: Specifies the burial depth Z_f (in meters) at the time of the deformation in the Sperrevik permeability relation.

Argument: Value

Default: 2000

Repeatable: No

6.19.14 `<sgr-curve-parameters>`

Description: Parameters defining the user specified SGR to fault permeability relationship. The keyword is required when `<fault-permeability-method>` is SGR_ CURVE.

Repeatable: No

6.19.14.1 `<kfz-sgr-curve-filename>`

Description: Name of the file containing the tabulated SGR and fault zone permeability (k_{fz}) values. SGR must be in the first column and k_{fz} in the second. Currently, only numbers separated by spaces with an optional text header is supported.

Argument: File name

Repeatable: No

6.19.15 `<clay-smear-method>`

Description: Specifies additional method to asses the flow potential of the fault. E.g. Shale Smear Factor (SSF) and Clay Smear Potensial (CSP). Currently not implemented.

Argument:

Default:

Repeatable: No

6.19.16 `<fault-thickness-method>` **(necessary)**

Description: Specifies the fault thickness method to be used when computing the fault thickness. Currently supported methods are: RATIO, RATIOMAXMIN and CONSTANT. If RATIO is selected the fault thickness is given by the fault displacement multiplied with a factor (i.e. ratio between fault thickness and displacement). If RATIOMAXMIN is selected the fault thickness is additionally truncated by a maximum and minimum limit. If CONSTANT is selected the fault thickness is assigned a constant value.

Argument: One. Thickness method

Default:

Repeatable: No

6.19.17 `<fault-thickness-parameters>` **(necessary)**

Description: Specifies the parameters that are used when computing the fault thickness depending on the method that is selected in `<fault-thickness-method>`.

Argument:

Default:

Repeatable: No

6.19.17.1 `<ratio>`

Description: Specifies the ratio between fault thickness and displacement that is used for computing fault thickness when `<fault-thickness-method>` is set to RATIO or RATIOMAXMIN.

Argument: Value

Default: 0.01
Repeatable: No

6.19.17.2 `<maximum>`

Description: Specifies the maximum limit (in meters) of the computed fault thickness when `<fault-thickness-method>` is set to RATIOMAXMIN.

Argument: Value
Default: 1
Repeatable: No

6.19.17.3 `<minimum>`

Description: Specifies the minimum limit (in meters) of the computed fault thickness when `<fault-thickness-method>` is set to RATIOMAXMIN.

Argument: Value
Default: 0
Repeatable: No

6.19.17.4 `<constant>`

Description: Specifies the fault thickness (in meters) when `<fault-thickness-method>` is set CONSTANT. All faults in the model is assigned the same thickness.

Argument: Value
Default: 10
Repeatable: No

6.20 `<fault-statistics>`

This action is used to create fault statistics and displacement profiles from an existing fault set. Input can be either a fault set on the internal Havana format or a structural model. The fault statistics is calculated if `<output-file>` is given, and if `<output-profiles-directory>` is used then the displacement profiles are calculated. If structural model is given as input (`<input-havana-structural-model-directory>`) the statistics and profiles will also be calculated per horizon.

Example:

```
<fault-statistics>
  <input-havana-structural-model-directory> havanaStructuralModel </input-havana-structural-mo
  <output-statistics-file> fault-statistics.log </output-statistics-file>
  <profile>
    <output-directory> profiles </output-directory>
    <value> displacement </value>
  </profile>
  <group-name> group1 </group-name>
  <accumulated-profile>
    <output-directory> accumulated_profiles </output-directory>
    <value> displacement </value>
    <breadth> 1000.0 </breadth>
    <line>
      <point-start>
        <x> 461110</x>
        <y> 5932880</y>
        <z> 1000 </z>
      </point-start>
      <point-end>
        <x> 464890 </x>
        <y> 5936260</y>
        <z> 1000 </z>
      </point-end>
    </line>
  </accumulated-profile>
  <output-intensity-surface>
    <intensity-type> fault </intensity-type>
    <intensity-type> displacement </intensity-type>
    <output-directory> intensity </output-directory>
    <horizon-name> BaseA </horizon-name>
    <output-file-name> intensity_surface </output-file-name>
    <range> 500 </range>
    <range-smoother> 100 </range-smoother>
    <nx> 100 </nx>
    <ny> 100 </ny>
  </output-intensity-surface>
</fault-statistics>
```

6.20.1 `<input-havana-faults-directory>`
See 6.3.1.

6.20.2 `<input-havana-structural-model-directory>`

Description: Input fault set given as a structural model. Can be used instead of `<input-havana-faults-directory>`. The calculations will be done per horizon if this is used. The structural model is created with the `<import-rms-fault-data>` action.

Argument: One. The directory.

Default:

6.20.3 `<output-statistics-file>`

Description: Specifies the directory name to where the faults statistics file is written. See A.6 for an example.

Argument: One. The name.

Default:

Repeatable: No

6.20.4 `<attribute>`

Description: Specifies the attribute name of which faults to use in the FaultStatistics action. The attribute name of the faults is set under `<group-faults>` or specified when simulating faults.

Argument: One. The attribute name.

Repeatable: No.

6.20.5 `<group-name>`

Description: Specifies the group names of which faults to use in the FaultStatistics action. The group name of the faults is set under `<group-faults>` or specified when simulating faults.

Argument: One. The group name.

Repeatable: yes.

6.20.6 `<ignore-bounding-box>`

Description: Whether to ignore the bounding box of the faults when calculating statistics. If yes, then statistics such as fault length is estimated beyond the bounding box.

Argument: Yes/No

Default: No

Repeatable: No.

6.20.7 `<profile>`

Description: Calculate the displacement/throw profile for all faults.

Repeatable: No.

6.20.7.1 `<output-directory>`

Description: Specifies the directory name to where the faults profile file is written. See A.7 for an example.

Argument: One. Name of the directory.

Default:

Repeatable: No

6.20.7.2 `<value>`

Description: Specifies the value for the accumulated profile.

Argument: Displacement or throw.

Default: Displacement

Repeatable: No

6.20.8 `<accumulated-profile>`

Description: Calculate the accumulated profile of all faults through a given line.
Repeatable: No.

6.20.8.1 `<output-directory>`

Description: Specifies the directory name to where the accumulated displacement profiles are written.
Argument: One. Name of the directory.
Default:
Repeatable: No

6.20.8.2 `<value>`

Description: Specifies the value for the accumulated profile.
Argument: Displacement or throw.
Default: Displacement
Repeatable: No

6.20.8.3 `<width>`

Description: Limits the part of the fault set that is used to calculate displacement/throw to the area `<width>` meters to the left and right of the specified line. All faults outside this area are excluded from the calculation. If this argument is not provided or negative, an infinite line is used.
Argument: One, the width of the area in meters (resulting in an area of total width two times this value).
Repeatable: No

6.20.8.4 `<line>`

Description: Calculate the accumulated displacement of all fault through a given line.
Argument: One. The name.
Repeatable: yes.

6.20.8.4.1 `<point-start>`

Description: Specifies the starting point of the line.
Argument: Point in global coordinates.
Repeatable: No

6.20.8.4.1.1 `<x>`

Description: Global x-coordinate.
Default: x-coordinate of reference point.

6.20.8.4.1.2 `<y>`

Description: Global y-coordinate
Default: y-coordinate of reference point.

6.20.8.4.1.3 `<z>`

Description: Global z-coordinate
Default: z-coordinate of reference point.

6.20.8.4.2 `<point-end>`

Description: Specifies the end point of the line.
Argument: Point in global coordinates.
Repeatable: No

6.20.8.4.2.1 `<x>`

Description: Global x-coordinate.
Default: x-coordinate of reference point.

6.20.8.4.2.2 `<y>`

Description: Global y-coordinate
Default: y-coordinate of reference point.

6.20.8.4.2.3 `<z>`

Description: Global z-coordinate
Default: z-coordinate of reference point.

6.20.9 `<output-intensity-surface>`

Description: Generate intensity surface for given horizons.
Repeatable: No.

6.20.9.1 `<intensity-type>`

Description: Type of the intensity surface. Allowed values are "fault" and "displacement".
Repeatable: Yes.

6.20.9.2 `<output-directory>`

Description: Directory where surface files are stored.
Repeatable: No.

6.20.9.3 `<output-file-name>`

Description: Suffix for the file names.
Repeatable: No.

6.20.9.4 `<horizon-name>`

Description: Name of the horizon for which intensity surfaces are generated. If not specified –
   surfaces are generated for all horizons given in the structural model.
Repeatable: Yes.

6.20.9.5 `<range>`

Description: Specifies how far from the surface point fault observations may lie to be taken into
   account .
Repeatable: No.

6.20.9.6 `<range-smoother>`

Description: Parameter of a gaussian kernel function used for smoothing the surface. The value of the parameter should be non-negative. For value equal to 0 smoothing is not applied.

Repeatable: No.

6.20.9.7 `<nx>`

Description: Resolution of the surface in X-direction

Repeatable: No.

6.20.9.8 `<ny>`

Description: Resolution of the surface in Y-direction

Repeatable: No.

A   File formats

A.1 Input of structural model exported from RMS

The Havana export of the RMS structural model was introduced in RMS 2011. This export is available by choosing the `Extract fault data to files...` job for the desired horizon model.

The whole structural model is exported to a single folder. The definition of the structural model is given in the `fault_model.rms` file. All fault line points are exported in a single file named `fault_lines.rms`, while the fault surfaces for each fault is given in files names `<fault-name>_grid.rms` and the horizons are exported in files with the same name as the horizon.

The RMS 2012 export has the same file formats as the previous RMS 2011 export, but with some additional data added and some data removed. HAVANA is compatible with the 2012 export format and register the version of the export format through the version number given at the beginning of the structural model file. The RMS 2012 export format has version number 3.0.

A.1.1 Structural model definition

The structural model file contains the following sections:

BoundingBox The definition of the used fault model bounding box.

Faults Number of faults, and name of each fault in this structural model. For each fault azimuth, dip and type of fault is given. The dip and azimuth information is also given by the transformation matrix in the fault file, and is hence not used.

Truncations The fault truncations. Note that HAVANA currently only supports simple truncations, i.e. only one truncation rule per line.

Horizons The names of the horizons in the structural model.

FaultBlocks The definition of the fault blocks in the model.

In the RMS 2012 export the format of the faults in the structural model file is "`fault_name fault_type age_group usage`", where the age group and usage are new parameters. The usage is given as either USED, NO_THROW, GHOST or UNUSED. The NO_THROW parameter means that the fault have no displacement. The GHOST parameter means that the fault is only there due to truncations. Further, the input parameters azimuth and dip are removed.

The horizon format is given as "`horizon_name horizon_type age_group`", where the two latter parameters are new. The horizon type gives whether the horizon is a erosion surface, i.e. UNCONFORMITY or an ordinary horizon, i.e. DEPOSITION.

Age is also given for the fault blocks, however in the form of an age interval. The fault block format is as follows "`block_id min_age max_age neighbor_faults`".

Example of a structural model definition file:

```
Version 3.0
#
# syntax:
# Lines starting with '#' are comment lines
```

```
# Data is separated by white space which is either ' ' or a tab
#
# This file is designed to allowed a external program create a fault model
# similar to the one used inside rms.
#
# To read this file look for the keywords 'Version,BoundingBox,Faults,Truncations'
# Each section has a fixed format after that
#
#
#----------------------------------------------------
# Keyword to the section
BoundingBox
#
# The bounding box for the fault model. The bounding box can be
# rotated arround the vertical axis. It is defined by a center
# point, the length of the three sides and anti clockwise rotation
# when viewed from the top
#
#
# Box center east north depth (must used doubles to read this offset)
463087.683838 5933652.235840 1896.718506
# Box size east north depth
7762.404297 7672.138184 866.392700
# Box rotation in degrees (clockwise viewing from the top)
0.000000
BoundingBoxEnd
#
#
#----------------------------------------------------
# Keyword to the section
Faults
#
# First the number of faults
FaultCount 7
#
# Each fault is formed into a single valued surface form a particular
# gridding direction. The dip & dip az (in degrees) is the gridding
# direction used. The format is as follows. fault names should have
# no white space in them. The dip & dipaz define a normal. This normal
# points towards the above side of the fault.
#
# Fault-Type N: normal, R: reverse, U: undefined
#
# Usage:
#     UNUSED      Not used in horizon model
#     GHOST       Inactive fault, but has to be in the model because
#                 it truncates active fault(s)
#     NO_THROW    Active fault with no throw
#     USED        Active fault
```

```
#
# Format: fault_name fault_type age_group usage
#
F1 N 0 USED
F2 N 0 USED
F3 N 0 USED
F4 N 0 USED
F5 N 0 USED
F6 N 0 USED
F7 R 0 USED
FaultsEnd
#
#
#----------------------------------------------------
# Keyword to the section
Truncations
#
# We don't really know how many truncations we might get so
# keep reading them until you find 'TruncationsEnd'. In the
# user interface:
# About = Hanging wall or HW
# Below = Footwall or FW
# keep reading them until you find 'TruncationsEnd'
#
# faultA > faultB     ...faultA is truncated above faultB
# faultA < faultB     ...faultA is truncated below faultB
# faultA < faultB & > faultC
# faultA is truncated where it is below faultB and above faultC
#
# The lists should only contain faults named in the Faults section
#
F2 > F1
F3 > F1
F3 > F2
F4 < F3
#
TruncationsEnd
#
#
#----------------------------------------------------
# Keyword to the section
Horizons
#
# Format: horizon_name horizon_type age_group
#
# Horizon name cannot contain spaces.
# Horizon type is either DEPOSITION or UNCONFORMITY.
# Age group is an integer representing the age of the horizon and is used
# together with the fault age to truncate faults against unconformities.
```

```
#
TopC DEPOSITION 0
TopB DEPOSITION 0
TopA DEPOSITION 0
BaseA DEPOSITION 0
#
HorizonsEnd
#
#
#----------------------------------------------------
# Keyword to the section
FaultBlocks
#
# Format:  block_id min_age max_age neighbor_faults
#
# The fault block id is an integer. We then list which side
# the fault block is of neighbor faults:
#
# Like for truncations > means HW side and < means FW side
#
0 0 0 > F1
1 0 0 < F1 > F2
2 0 0 > F5 > F6
3 0 0 < F1 > F3 > F4 < F5 > F6
4 0 0 < F1 < F2 > F3 < F4 > F6
5 0 0 < F1 < F2 < F3 > F6
6 0 0 < F6 > F7
7 0 0 < F6 < F7
#
FaultBlocksEnd
#
#
```

A.1.2 Fault surface file

The fault file is named `<fault-name>_grid` and has a format similar to the `fault_model` file. The file contains the following sections:

TransformMatrix4x4 A 4x4 transformation matrix defning the transformations between local and global coordinates.

NU and NV number of points in local u and v directions.

Data NU*NV values with the local w-value (distance to reference plane) for each point on the fault surface.

Thickness NU*NV values, currently only used to define the fault tip. Positive thickness values means that the point is within the fault tip, while zero or a negative value means that the point is outside the fault tip line.

Example fault surface file:

```
RMS_fault_grid_version 2.0
#
```

```
# Comment-lines start with #
#
# Definition of local space by 4x4 matrix
# Use the matrix M to transform local grid points to user coordinates
#
# [x, y, z, w] = M * [u, v, data[u][v], 1.0
#
# u,v are indexes into the data/thickness sections.
# Rotation, reference point and grid increments are embedded in the
# matrix.
#
TransformMatrix4x4
-48.6113950022 -47.0015702157 73.6735003873 467405.018948
87.3895432872 -26.1451409223 40.9816952212 5929790.22863
0 -84.30470926 -53.7839752852 2677.77285235
0 0 0 1
EndTransformMatrix4x4
#
# Number of cells in U direction
#
NU 98
#
# Number of cells in V direction
#
NV 99
#
# 2d array of grid values.  NU * NV values.
#
Data
-0.840338
-0.88161
-0.970237
-1.08047
-1.19489
    .
    .
EndData
#
# Thickness attribute.
# 2d array of grid values.  NU * NV values.
# Positive where the fault is active, negative outside the active
# area.
# The 0-contour of this attribute is the tipline of the fault.
#
AttributeThickness
-0.248483
-0.209946
-0.119374
0.0196044
```

```
0.190451
     .
     .
EndAttributeThickness
```

A.1.3 Fault line file

The fault lines for alle the fault surfaces and all the horizons are given in the `fault_lines` file. This file is on RMS internal points format, see section A.9. For each fault line point the following attributes are given:

FaultSide Side of fault that the fault line point belongs to. Either `hw` or `fw`.

Fault Name of fault that the fault line point belongs to.

Horizon Name of horizon that the fault line point belongs to.

FaultBlock ID of fault block where this fault line point is located.

Extrapolation Indicator telling if the fault line point is eroded (0) or not (1)

Example file:

```
String   FaultSide
String   Fault
String   Horizon
Discrete FaultBlock
Discrete Extrapolation
466914.549  5929860.689  2105.067  hw  F1  TopC  1 0
466758.129  5929773.680  1764.236  fw  F1  TopC  9 0
466865.918  5929948.068  2104.987  hw  F1  TopC  1 0
466705.468  5929858.816  1764.563  fw  F1  TopC  9 0
                          .
                          .
```

A.1.4 Horizon file

The horizons are exported to files with the same name as the horizons. They are exported as point sets on the RMS internal points format, see section A.9. Two attributes are given for each point, a `FaultBlock` attribute giving witch fault block the point belongs to, and a `Active`. The points form a rectangle for each fault block. The nodes that contribute to cells touching the fault block have an Active value of 1, and the points completely outside the block has Active value 0.

Example file:

```
Discrete  FaultBlock
Discrete  Active
459206.482  5929816.167  1730.060    13  UNDEF
459406.482  5929816.167  1731.072    13  UNDEF
459606.482  5929816.167  1732.178    13  UNDEF
459806.482  5929816.167  1733.431    13  UNDEF
                       .
                       .
```

A.2 Input of Elliptic faults

Havana can read and write this format. The format is as follows:

The lines starting with a # are comment lines.

Each remaining line of the file contains data for one fault. Number of faults are counted from the list. The numbers read are interpreted as:

- Name of fault.

- 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 0 and 1: 0 means that all displacement happens on the foot-wall side; 1 means that all displacement happens on the hanging-wall side, and 0.5 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 range 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 is normal: The input must be either 1 (if the fault is normal), or 0.

As an example, the following will be a legal input file.

```
#Total number of output Elliptic faults: 14
#FaultName      x          y          z      Displacement Asym  Strike   Dip    Length    Height    Range    Norm

HF1         460120.71  6583039.71  2813.25      25.98   0.00    53.65   67.55  1268.99   760.03   197.72     1
HF2         463713.80  6580121.80  2941.61      23.56   0.00    57.86   84.53  1219.82   667.54   165.42     1
HF3         460057.22  6584665.58  2634.95      21.38   0.00   137.97   79.91  1087.29   671.07   159.27     1
HF4         460115.32  6581230.04  2505.64      22.61   0.00   167.44   67.92  1015.46   438.66   141.55     1
HF5         462476.17  6581641.80  2617.91      25.78   0.00   152.40   75.83  1307.16   662.10   181.02     1
HF6         462160.33  6583824.32  2993.71      26.77   0.00   154.53   85.95  1331.55   668.31   214.93     1
HF7         463932.37  6581564.19  2984.36      22.66   0.00   151.45   76.84  1098.69   610.64   196.89     1
HF8         463211.27  6584549.08  2707.61      24.34   0.00   155.60   88.43  1220.21   642.74   194.54     1
HF9         460863.05  6581489.07  2656.46      19.56   0.00   155.97   75.53   956.30   512.86   126.46     1
HF10        464787.89  6582171.35  2728.76      20.05   0.00   147.89   77.78  1016.95   478.11   176.26     1
HF11        462628.23  6580536.22  2993.08      22.50   0.00    61.82   78.77  1116.67   693.65   163.70     1
HF12        463722.50  6581516.10  2619.67      23.37   0.00   155.87   82.07  1132.20   548.95   160.55     1
HF13        461410.30  6581764.77  2553.99      19.91   0.00   154.21   72.05   969.70   470.18   141.55     1
HF14        464945.67  6581687.74  2639.33      22.69   0.00   162.62   80.09  1054.77   473.31   148.67     1
```

A.3 Eclipse grid file

Eclipse grid files are used to define a grid in the actions 6.15, 6.17 and 6.18. The format of the file is as follows (each list of values after a keyword is ended by a /):

1. Some header lines

2. A line beginning with the keyword SPECGRID to indicate the grid size

3. A line containing four integers and a string where the three first integers are nx, ny and nz of the grid

4. A blank line

5. A line containing the keyword COORD

6. 2*(nx+1)*(ny+1) values giving the coordinates of the nodes on the top and bottom surface of the grid. The nodes generally start in the NW corner and cycle W to E and then N to S

7. A blank line

8. A line containing the keyword ZCORN

9. (nx+1)*(ny+1)*(nz+1) values giving the position of the grid nodes starting on the top in NW, then cycle W to E, then N to S and finally top to bottom

10. There might by several parameters, like ACTNUM, PERMX, PERMY, PERMZ given for the grid nodes, separated by blank lines

Example of an Eclipse grid file:

```
-- =========================================================================
--
-- Exported from StructModGrid1
--
-- Exported by user georgsen from RMS2013 at 2012-03-28 11:34:59
--
-- =========================================================================

SPECGRID
  74  74  76   1  F  /

COORD
    459320.997    5929968.905    1680.605    459321.004    5929968.905    1792.603
    459421.012    5929968.905    1681.504    459420.987    5929968.905    1793.727
    459521.028    5929968.905    1682.359    459520.967    5929968.905    1794.826
    459621.046    5929968.905    1683.165    459620.944    5929968.905    1795.896
.
.
.
    466621.166    5937368.900    2124.704    466620.812    5937368.902    2272.490
    466720.994    5937368.902    2125.454    466720.997    5937368.900    2273.277
  /

ZCORN
    1680.605    1681.504    1681.504    1682.359    1682.359    1683.165
    1683.165    1683.921    1683.921    1684.621    1684.621    1685.274
    1685.274    1685.864    1685.864    1686.382    1686.382    1686.814
      .
      .
      .
    2269.395    2270.160    2270.160    2270.931    2270.931    2271.708
    2271.708    2272.490    2272.490    2273.277    /

FaultBlock
          7         7         7         7         7
          7         7         7         7         3
          3         3         3         3         3
        .
        .
        .
```

```
         1        1        1        1        1
         1        1        1        1
   /
```

A.4 Well data files

A.4.1 Input of fault wellpicks

This file is used for giving points of faults observed in well for conditioning of the fault surface It contains the coordinates of well picks along with fault name and well name. The first column is the fault name. The second column is the well name and is discarded when the file is read.

Example:

```
 F1            Well_A            463174.625            5933349.000            1598.8566
 F2            Well_E            460241.906            5935144.000            1625.7587
```

A.4.2 Input of wells with no observed faults

This file is used for giving well paths where no faults have been observed. These are the global coordinates for each negative observation point. It is not fault specific, each point can affect more than one fault.

Example:

```
   463174.625            5933349.000            1598.8566
   460241.906            5935144.000            1625.7587
```

A.5 Output of fault summary

The format for the output of simulated fault summary using the command
<output-fault-summary> (see Section 6.11.8) is very similar to the format for inputing Elliptic faults (see Section A.2). The only differences are that in the faults summary 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 inputing Elliptic faults do not appear in the statistics format.

A.6 Output of fault statistics

The output of fault statistics calculated from action 4.19.

Example:

```
FaultName Horizon Length UntruncatedLength Height MaxDisplacement StrikeDirection DipDirection Fau
F1 -       19091   9400   1200 568.92    330.889  58.0019  Normal
F2 -       12460.1 7000   1000 82.4527   323.134  59.2168  Normal
F3 -       11164.2 6800   1300 86.0651   144.153  61.5255  Normal
F4 -       3150.13 2200   600  37.3582   119.818  62.8024  Normal
F5 -       4074.12 4200   900  90.4258   115.898  67.2979  Normal
F6 -       3994.26 3700   1100 108.362   2.25989  37.7407  Normal
F7 -       2488.86 2300   1100 76.0598   347.616  45.8082  Reverse
F1 TopC    8864.91 19464.9 -1   560.224  330.889  58.0019  Normal
F1 TopB    8867.76 19467.8 -1   564.925  330.889  58.0019  Normal
F1 TopA    8877.49 19477.5 -1   571.894  330.889  58.0019  Normal
F1 BaseA   8666.87 19466.9 -1   576.15   330.889  58.0019  Normal
```

A.7 Output of fault displacement profiles

The output of fault statistics displacement profile from action 4.19. The first is the local x-coordinate and the second is the displacement.

Example:

```
-4511.55,415.857
-4411.55,415.692
-4311.55,415.923
-4211.55,416.501
-4111.55,415.672
-4011.55,417.969
-3911.55,421.132
-3811.55,424.227
-3711.55,427.323
-3611.55,430.446
-3511.55,433.529
-3411.55,437.13
```

A.8 Roxar Text

The format is plain ascii file with three numbers on each line, and the last line with three times 999.000000.

Example:

```
466490.565018  5930425.855033  1903.495902
466540.312948  5930339.097905  1906.702286
466587.374000  5930250.846000  1909.408000
466633.177000  5930161.894000  1911.640000
466678.775000  5930072.829000  1913.265000
466725.124000  5929984.180000  1914.480000
466773.217439  5929896.503115  1915.457989
999.000000  999.000000  999.000000
```

A.9 RMS Internal Point Format

We use the format from RMS2010.

Example:

```
Discrete   FaultBlock
String     FaultTag
Float      VerticalSep
461674.174      5929715.370      2122.258      1 UNDEF UNDEF
461674.156      5929706.245      2137.951      3 UNDEF UNDEF
461657.138      5929715.049      2107.089      1 UNDEF UNDEF
461670.600      5929715.101      2119.513      1 UNDEF UNDEF
461703.690      5929715.205      2149.951      1 UNDEF UNDEF
461792.751      5929715.363      2232.304      3 UNDEF UNDEF
461851.181      5929748.293      2230.288      1 UNDEF UNDEF
462134.406      5929907.184      2207.154      1 UNDEF UNDEF
```

A.10 Internal Havana Format

This format is used to store a fault set of SFM faults and to be communicated between different modules in HAVANA. A separate directory is used with a number of files. One file is named `FaultSet.txt` and the others are one file for each fault. The format of the file for each fault is the same as used for the export of the structural model from RMS in the RMS2011.

The `FaultSet.txt`-file has the following contents:

- Bounding box: xMin, yMin, zMin, lX, lY, lZ, 0

- Dimensions of truncation matrix

- Truncation matrix

- Number of faults

- For each fault: First line: Indicator saying if the fault has been changed. Second line: Totally 11 items:

  1. The file name of the file containing the data for the fault

  2. The name of the fault

  3. Slip type of fault (normal/reverse)

  4. HW/FW distribution of the displacement

  5. Displacement range

  6. Grid x-min value

  7. Grid y-min value

  8. Grid length in x-direction

  9. Grid length in y-direction

  10. Indicator telling if the fault has displacement data

  11. First parameter for relationship between maximum displacement and length of fault.

  12. Second parameter for relationship between maximum displacement and length of fault.

  13. Parameter for relationship between length and height of fault.

  14. Group name telling if the fault belong to group of large faults or small faults.

```
# Bounding box
459206.482 5929816.17 1463.52214 7762.4043 7672.13818 866.3927 0
# Truncation rules
7 7
0 0 0 0 0 0 0
1 0 0 0 0 0 0
1 1 0 0 0 0 0
0 0 2 0 0 0 0
0 0 0 0 0 0 0
0 0 2 0 0 0 0
0 0 0 0 0 0 0

# Number of faults
7
# FORMAT for each fault:
# Is changed (1 means fault is changed, 0 means unchanged)
```

```
# Filename - Fault name - Slip type - Distribution - Range - Xmin - Ymin - lx - ly - Has displacem
1
F1.grid F1 0 0.7 1686.09419 -4614.79831 -3004.21555 10600.0002 7500.00072 1 1 0.01 2 NA
1
F2.grid F2 0 0.5 1000 -6749.79832 -648.790323 11000 1100 1 1 0.01 2 NA
1
F3.grid F3 0 1 500 -3698.71561 -669.357798 8700.00008 1100 1 1 0.01 2 NA
1
F4.grid F4 0 0.5 1000 -4897.01488 -719.402949 7399.99993 999.99995 1 1 0.01 2 NA
1
F5.grid F5 0 0.5 1000 -2363.69866 -702.054795 5000.00016 1100 1 1 0.01 2 NA
1
F6.grid F6 0 0.5 1000 -2129.09092 -724 4600.00003 1100 1 1 0.01 2 NA
1
F7.grid F7 0 0.5 1000 -1395.13515 -661.081081 2800.00009 1000 1 1 0.01 2 NA
```

The file specifying the data for the fault contains the following: (Lines starting with # are comments)

- Transformation matrix specified by keyword: TransformMatrix4x4. The first line contains four numbers with the x-components of the strike vector, dip vector, normal vector and reference point. The second line contains the y-components, the third line the z-components and the fourth line the numbers 0 0 0 1. This matrix is concluded with the keyword EndTransformMatrix4x4.

- Number of cells for fault surface. Two lines, with the first specified by NU and the number of cells in x-direction, and the second with NV and the number of cells in y-direction.

- Start point for fault surface grid. Two lines, with the first specified by Umin and the start value in local x-direction, and the second with Vmin and the start value in local y-direction.

- Size of fault surface grid. Two lines, with the first specified by LU and the number of grid cells in the local x-direction, and the second with LV and the number of grid cells in the local y-direction.

- Number of cells for displacement grid. Two lines, with the first specified by NU and the number of cells in x-direction, and the second with NV and the number of cells in y-direction.

- Start point for displacement grid. Two lines, with the first specified by U0 and the start value in local x-direction, and the second with V0 and the start value in local y-direction.

- Size of displacement grid. Two lines, with the first specified by U-length and the number of grid cells in the local x-direction, and the second with V-length and the number of grid cells in the local y-direction.

- Center point for elliptic displacement trend. Three lines. The first with x and value, the next with y and value and the third with z and value.

- Size of elliptic trend. Three lines: Values for length, height and max trend.

- Data grid. Starts and ends with keywords Data and EndData. In between, all the grid values for the fault surface, one value on each line.

- Fault thickness represented by a thickness attribute. This attribute also defines the fault tip line. Start and ends with keywords AttributeThickness and EndAttributeThickness. In between, all the grid values for the thickness, one value on each line.

- Displacement field. Starts and ends with keywords AttributeDisplacement and EndAttributeDisplacement. In between, all the grid values for the displacement, one value on each line.

B   Example RMS workflow for running HA-VANA

B.1

Figure B.1 shows an example of a RMS workflow where HAVANA is run with three separate actions that is called from IPL-scripts (Parts 1-3 in the workflow).
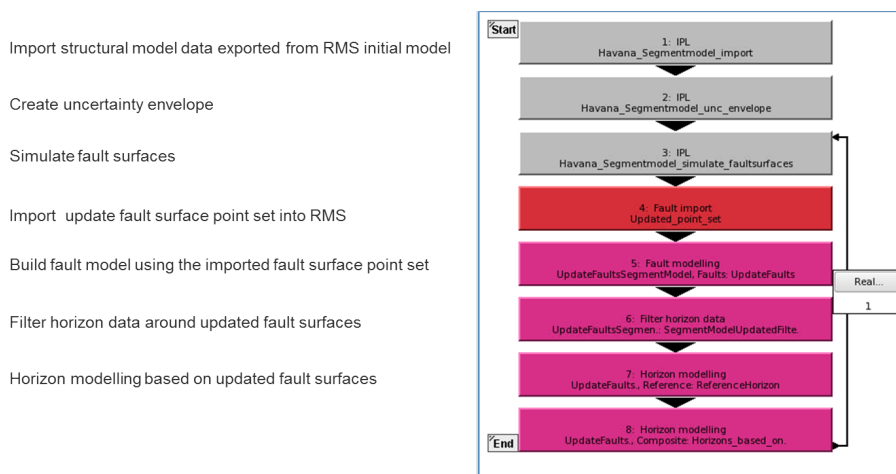

Figure B.1. Example of a workflow set up in RMS to run HAVANA.

This is an example of an IPL-script, namely the first in the workflow (Havana_-Segmentmodel_-import):

```
// IPL: Run Havana version 6
// Import RMS Initial model into Havana

String modelFile
String modelFilePath
String command
String havanaDir
String checkLogFile,logFilePath

// Define functions and common IPL variables
// NOTE: Remember to update the include file "Define_common_variables.ipl" to fit your Havana proj
Include("IPL_common/Define_common_functions.ipl")
Include("IPL_common/Define_common_variables.ipl")

// ------- Assign variables ------------------

// Model file
modelFile  = "ImportRMSFaults_uneroded.xml"
```

```
// File path for directory with model file relative to where Havana is started.
havanaDir  = "HavanaDir/SegmentModel/Input"

// Log file to check whether Havana has completed with no errors
checkLogFile = "ImportRMSFaults_uneroded.log"

// ------  End assignment -------------------

logFilePath = projectPath + "/" + havanaDir + "/" + checkLogFile

IF checkEnabled THEN
  command = removeFile + logFilePath
  ShellCommand(command)
  Print("Remove log file: ",logFilePath)
ENDIF

modelFilePath = projectPath + "/" + havanaDir + "/" + modelFile
command = progHavana6 + " " + modelFilePath
Print("Run command: ",command)
ShellCommand(command)

IF checkEnabled THEN
  // Scan through log file and check if OK

  IF FileExists(logFilePath)= 1 THEN
    IF CheckHavanaLogFileForErrors(logFilePath) = 0 THEN
      Print("Errors occured when running Havana")
HALT
    ENDIF
  ELSE
    Print("Error: Can not find log file: ",logFilePath)
    HALT
  ENDIF
ENDIF

Print("Finished running Havana 6 from IPL job")
Print("  ")
```

C   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 D 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.

D   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.

References

[1] Barnett, J.A.M., Mortimer, J., Rippon, J.H., Walsh, J.J. and Watterson, J.: Displacement Geometry in the Volume Containing a Single Normal Fault. *The American Association of Petroleum Geologists Bulletin*, **71**(8), 925–937 (1987) 26, 28

[2] Cardozo, N., Røe, P., Soleng, H.H., Fredman, N., Tveranger, J. and Schueller, S.: A methodology for efficiently populating faulted corner point grids with strain. *Petroleum Geoscience*, **14**(2), 205–216 (2008) 28

[3] Georgsen, F., Røe, P., Syversveen, A.R. and Lia, O.: Fault displacement modelling using 3D vector fields. *Computational Geosciences*, **16**(2), 247–259 (2012) 23

[4] Gibson, J.R., Walsh, J.J. and Watterson, J.: Modelling of bed contours and cross-sections adjacent to planar normal faults. *Journal of Structural Geology*, **11**(3), 317–328 (1989) 26

[5] Goovaerts, P.: Geostatistics for Natural Resources Evaluation. *Oxford University Press*, 125–184 (1997) 26

[6] Hoffman, K.S. and Neave, J.W.: The fused fault block approach to fault network modelling. In: Jolley, S.J., Barr. D., Walsh, J.J. and Knipe, R.J. (Eds) *Structurally Complex Reservoirs.* Geological Society, London, Special Publication **292**, 75–87 (2007) 23, 30

[7] Røe, P., Abrahamsen, P., Georgsen, F., Syversveen, A.R. and Lia, O: Flexible Simulation of Faults. *SPE Annual Technical Conference and Exhibition, Florence, Italy, 19—22 September*, SPE 134912 (2010)

[8] Georgsen, F., Røe, P., Syversveen, A.R., Vigsnes, M and Olsen, H. G.: Havana technical documentation, *NR-note*, SAND/06/2013 (2013) 25, 27, 29, 40

[9] Røe, P: RMS structural model export format, SAND/22/2010

[10] Walsh, J.J. and Watterson, J.: Distributions of cumulative displacement and seismic slip on a single normal fault surface. *Journal of Structural Geology*, **9**(8), 1039–1046 (1987) 27

[11] Walsh, J.J. and Watterson, J.: Displacement gradients on fault surfaces. *Journal of Structural Geology*, **11**(3), 307–316 (1989) 26

[12] Ponting, D.K.: Corner point geometry in reservoir simulation. *ECMOR I - 1st European Conference on the Mathematics of Oil Recovery* 56

[13] Manzocchi, T., Walsh, J.J., Nell, P., Yielding, G.: Fault transmissibility multipliers for flow simulation models. *Petroleum Geoscience* **5**, 53–63 (1999) 56

[14] Sperrevik, S., Gillespie, P.A., Quentin, J.F., Halvorsen, T., Knipe, R.J.: Empirical estimation of fault rock properties. *Norwegian Petroleum Society Special Publications* **11**, 109–125 (2002) 56, 150