



# Surface based fault format in Havana

Report from preliminary project

**Note no**

**SAND/01/09**

**Authors**

**Per Røe, Frode Georgsen, Petter Abrahamsen and Ariel  
Almendra Vazquez**

**Date**

**January 2009**

## **Norsk Regnesentral**

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

<b>Title</b>	<b>Surface based fault format in Havana</b>
<b>Authors</b>	<b>Per Røe, Frode Georgsen, Petter Abrahamsen and Ariel Almendral Vazquez</b>
Date	January
Year	2009
Publication number	SAND/01/09

### **Abstract**

This document describes the implementation of support for the surface based fault model introduced in RMS 9.0 in Havana. The focus of this document is integration with RMS, representation of the faults with corresponding displacement fields, and integration with existing Havana functionality.

Keywords	Fault modeling, structural modeling, Havana
Target group	StatoilHydro and NR
Availability	Internal
Project number	388011
Research field	Structural modeling
Number of pages	19
© Copyright	Norsk Regnesentral



# Contents

<b>1</b>	<b>Introduction</b> .....	<b>6</b>
<b>2</b>	<b>Fault representation</b> .....	<b>6</b>
2.1	Fault surface.....	7
2.2	Fault operator .....	7
2.3	Truncation rules.....	8
<b>3</b>	<b>IRAP RMS integration</b> .....	<b>10</b>
<b>4</b>	<b>Generation of the fault displacement fields</b> .....	<b>10</b>
4.1	Importing displacement field from RMS .....	11
<b>5</b>	<b>Uncertainty on fault surfaces</b> .....	<b>12</b>
<b>6</b>	<b>Integration with existing Havana functionality</b> .....	<b>14</b>
6.1	Modeling of subseismic faults .....	14
6.2	SUM.....	14
6.3	Fault sealing properties .....	14
6.4	Faulting and restoration of ECLIPSE grids .....	14
6.5	FaultFacies specific functionality.....	14
6.5.1	Local grid refinements along fault surfaces .....	14
6.5.2	Strain calculations.....	14
<b>7</b>	<b>Conclusion</b> .....	<b>15</b>
7.1	Issues with RMS integration.....	15
<b>8</b>	<b>Appendix A – Fault model file format</b> .....	<b>16</b>
<b>9</b>	<b>Appendix B ModifySurfaceFault model file example</b> .....	<b>18</b>
<b>10</b>	<b>Appendix C SurfaceFaultCenterLines model file example</b> .....	<b>19</b>

# 1 Introduction

Until recently pillar-based formats have been the default way to represent and model faults, but recently surface-based formats where the fault surface is represented as a general surface is gaining ground. These formats have several large advantages compared to pillar-based formats, most notably a simpler representation which is easier to work on, and the possibility to model more complex fault structures like for example listric faults.

It is crucial that Havana supports the new fault format. Support for a surface-based fault format was implemented in RMS 9.0, and this format has already become very popular. RMS is the main tool both for generation of input-data to Havana, and visualization and post-processing of the results from Havana.

Havana is based on a three-dimensional representation of the fault, which both encompasses the fault plane and a three-dimensional fault operator representing the deformation of the volume around the fault induced by the fault.

This document focuses on the work needed to support the new fault format in Havana and integrate it with already existing functionality.

## 2 Fault representation

Both the geometry of the fault surface and the faulting effect on the surrounding volume is needed to be able to use the new fault model with existing Havana functionality. The fault geometry representation should be similar to the fault plane representation in RMS 9.0, to make integration between Havana and RMS easier.

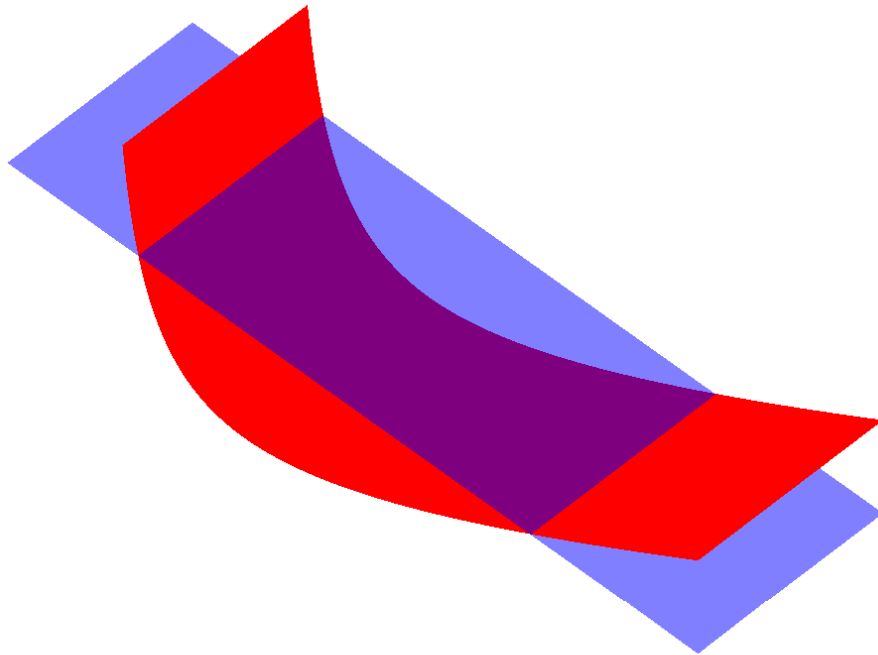


Figure 1 Listric fault with reference plane

## 2.1 Fault surface

The new surface-based fault format is characterized by the ability to model almost any surface. The only limitation is that it must be possible to define a reference plane that the fault surface can be mapped onto. A grid is defined on this reference plane, and for each node in the grid the location of the corresponding point in the fault surface is given by the distance from this point to the reference plane.

Figure 1 shows a listric fault in red, with the corresponding reference plane shown in transparent blue. The reference plane is defined by a normal vector for the plane, and a point that the plane goes through. The direction of the normal vector is used to define the side of fault, denoted as hanging wall (HW) and footwall (FW) sides.

## 2.2 Fault operator

We propose defining the faulting operator as a general 3 dimensional vector field, which is split in two, one field for each side of the fault surface. These vector fields define for any given point how this point was moved by the displacement generating the fault. This vector field should be continuous inside each fault block, and should represent a one to one relationship between the faulted and unfaulted volume, as shown in Figure 2.

The vector field will be stored in several parts. On the fault surface two sets of vectors are defined, one set for each side of the fault. The fault displacement away from the fault surface is represented with two 3D sets of vectors, one for each side, and the displacement for any given point can then be found through interpolation.

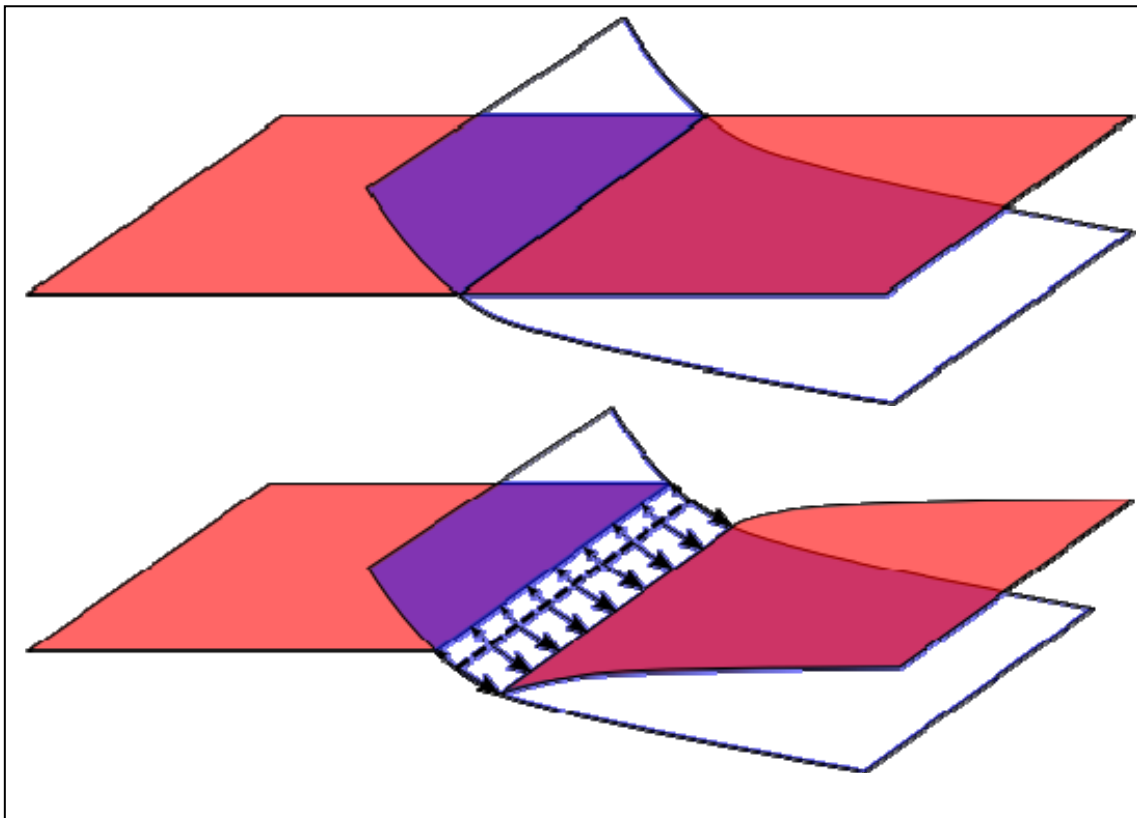


Figure 2: Fault operator displacing a horizon.

## 2.3 Truncation rules

The truncation rules are given as a list of all faults in the project. For each fault a list is given of faults truncating this fault and on which side of the truncating faults the untruncated part of the truncated fault is located. This is similar to how truncations already are defined in Havana, except that currently the active part of the fault is implicitly given by a center pillar for pillar based faults, or by the center point for elliptic faults.

Fault name:	Truncation rule:
F1	
F2	HW F1
F3	HW F1 HW F2
F4	FW F3
F5	
F6	FW F3
F7	

Figure 3 Truncation rules as defined in IRAP RMS

In contrast with RMS where the truncations only affect the fault plane, the truncations in Havana also must truncate the area affected by the fault operator. (This is currently not implemented in Havana, something which often leads to artifacts in restoration, faulting or strain calculations.) This may also be the case, even when the fault planes of the two faults do not intersect. An example of this is a graben as shown in Figure 4, where the smaller fault only affects the area between the two faults.

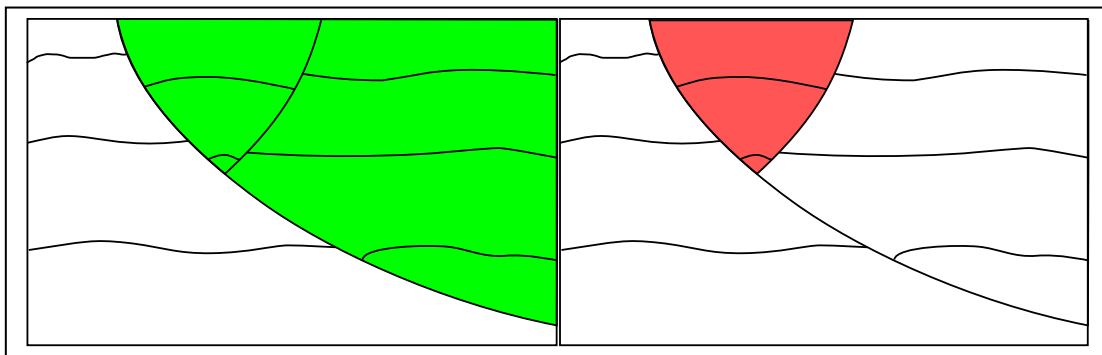


Figure 4: Graben seen from the side, with two faults. The area influenced by the large fault in green, area influenced by the small fault in red.

When truncating the fault planes it is sometimes necessary that the even though a specific part of the fault is truncated, the fault operator associated with this part of the fault should still be active. This is illustrated in Figure 5 and Figure 6. In Figure 5 the fault operator is only given for the untruncated part of the minor faults, resulting in that when the throw of the two minor faults is modified, the area in red is not affected, resulting in discontinuities. This is alleviated in Figure 6 by also taking the area affected by the fault operator associated with the truncated part of the faults.



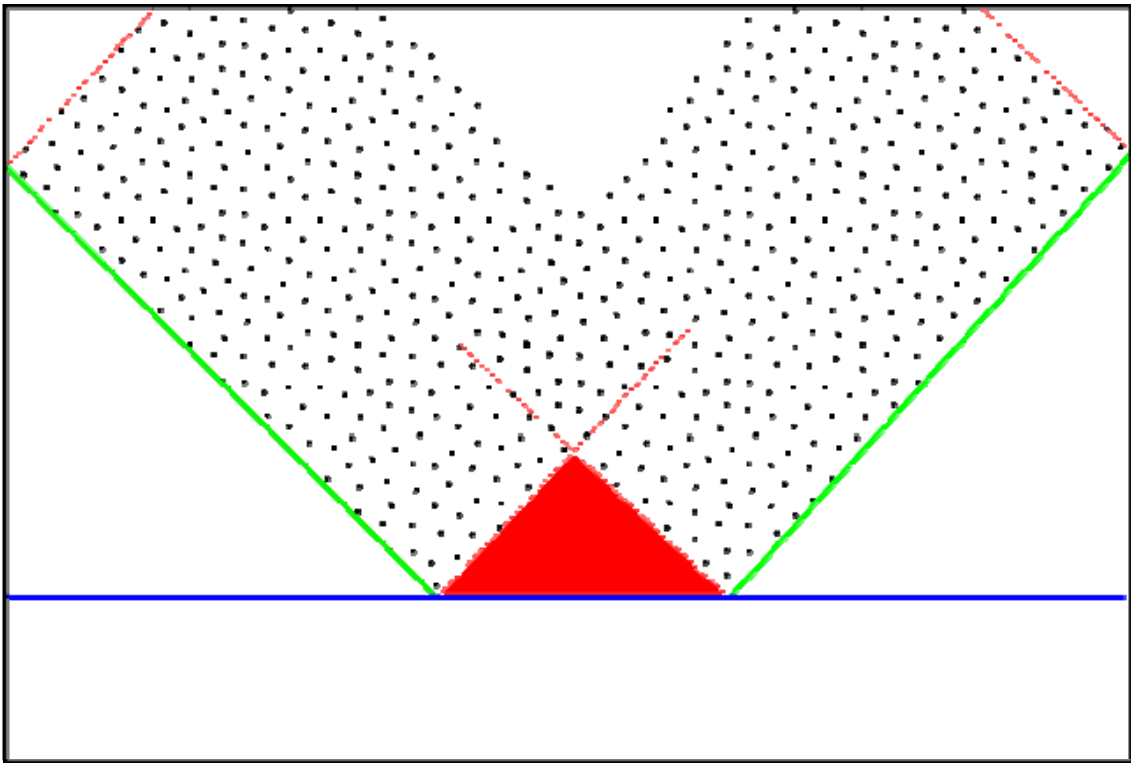


Figure 5: Case with three faults, a truncating fault in blue, and two truncated faults in red with incorrect area of influence.

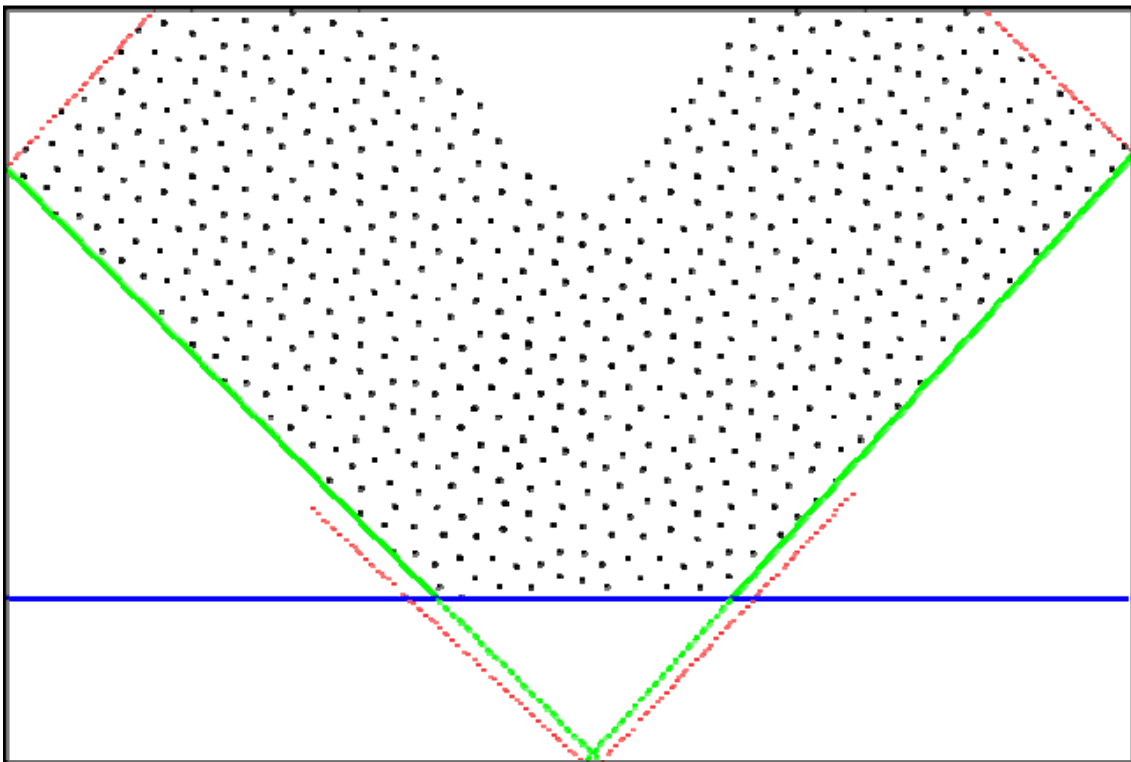


Figure 6: Same faults as in Figure 5, but where the area affected by the truncated part of the truncated faults are still taken into account.

### 3 IRAP RMS integration

Presently IRAP RMS does not have the functionality needed to export the complete fault model. Instead we get the building blocks needed to reassemble the fault model inside Havana. It is possible to export the following information from RMS 2009:

- The fault plane as a point set
- The normal vector defining the reference plane, and the hanging wall and footwall sides of the fault.
- The fault lines (defined by the intersection between the fault planes and the horizons) grouped by horizons.
- The truncation rules.

The export of the truncation rules and the normal vector will only be possible if an environment variable is set. No GUI is provided for this in RMS 2009. An example of an exported file with fault model information is given in Appendix A.

The resulting faults from Havana can be imported into RMS again through a structural modeling job. Havana can generate new fault surfaces and horizons, and these can be imported directly into the original structural model as long as the topology of the model has not changed. The modifications of the model that are allowed are the removal of faults, and moving faults as long as the truncation rules stay the same. However, adding faults and movement of faults which implies new truncation rules is not possible.

### 4 Generation of the fault displacement fields

The proposed format for defining the displacement fields is very general, and can encompass many different models for fault displacement. However, generating these displacement fields can be far from trivial, especially considering the limited amount of data available.

Possible ideas for the generation of the displacement fields include the following:

- Translation from an already existing format that includes a displacement field (PFM or Elliptic) should be straight forward.
- The fault throw can be approximated from the fault lines (intersections between fault surfaces and horizons). The question here is how reliable the data used to generate the fault lines are.
- If we have the displacement field on the fault surfaces it might be possible to generate the 3D fields by extrapolating from the fault surface in the direction of the normal vector associated with the reference plane. One possible model for extrapolation is the quadratic model employed in the PFM model. It's however not clear if this model will work for listric faults.

- If both the faulted and restored horizon sets are given it is possible to generate sets of displacement vectors on these horizons. Together with the shape of the fault surface this can be used to interpolate the displacement field for the whole faulted area. A potential problem is the decoupling of the displacement for the various faults.

#### 4.1 Importing displacement field from RMS

Our first attempt on generating the fault displacement field when importing the faults from RMS will be by using the information about horizon/fault intersections to generate the fault displacement field on the fault surface, and then extrapolate this field into the area around the fault surface.

The intersection between the fault surfaces and the horizons can be exported as fault lines from RMS. These fault lines are exported for each horizon, and grouped by faults. The fault lines for one fault are represented as a closed polygon given by a point set. These files are produced by exporting to the Roxar Ascii format. The fault lines for the Top A horizon in the Emerald 2009 project in RMS are presented in Figure 7.

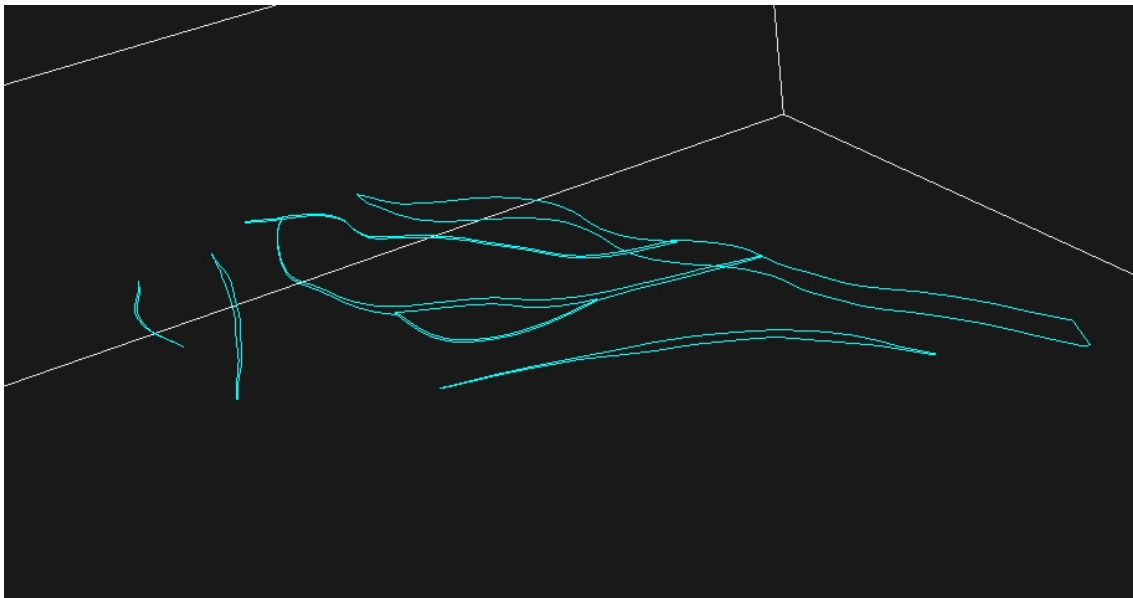


Figure 7: Faultlines from the Top A horizon in Emerald 2009 project represented as closed polygons.

As can be seen from Figure 7, not all polygons fade smoothly off to zero. This is due to truncations. When manipulating the displacement fields we need to find a method that works on the displacement fields as they were when the fault event occurred by going through the faults in chronological order.

At present it is not possible to decide the above and below side of a fault based on the output generated by RMS. This can be overcome by identifying for the relevant faults if they are normal or reverse in the model file.

The displacement can be changed for each fault by a ratio, where a value above 1.0 means increasing the displacement and a value below 1.0 means decreasing it. It is also possible to specify a number between 0 and 1 telling how much of the change that is going up and how much is going down. A value of 0.5 implies that the change is evenly distributed between the

up and down side. On each side of the fault surface the displacement decreases away from the fault plane as:

$$\left(1 - \frac{|x|}{r}\right)^2 \cdot D \quad 0 < |x| < r$$

where  $x$  is the distance from the fault surface measured in the normal direction,  $r$  is the range of the displacement field and  $D$  is the displacement at the fault surface. This is a positive number on the up side of the fault and a negative number on the down side.

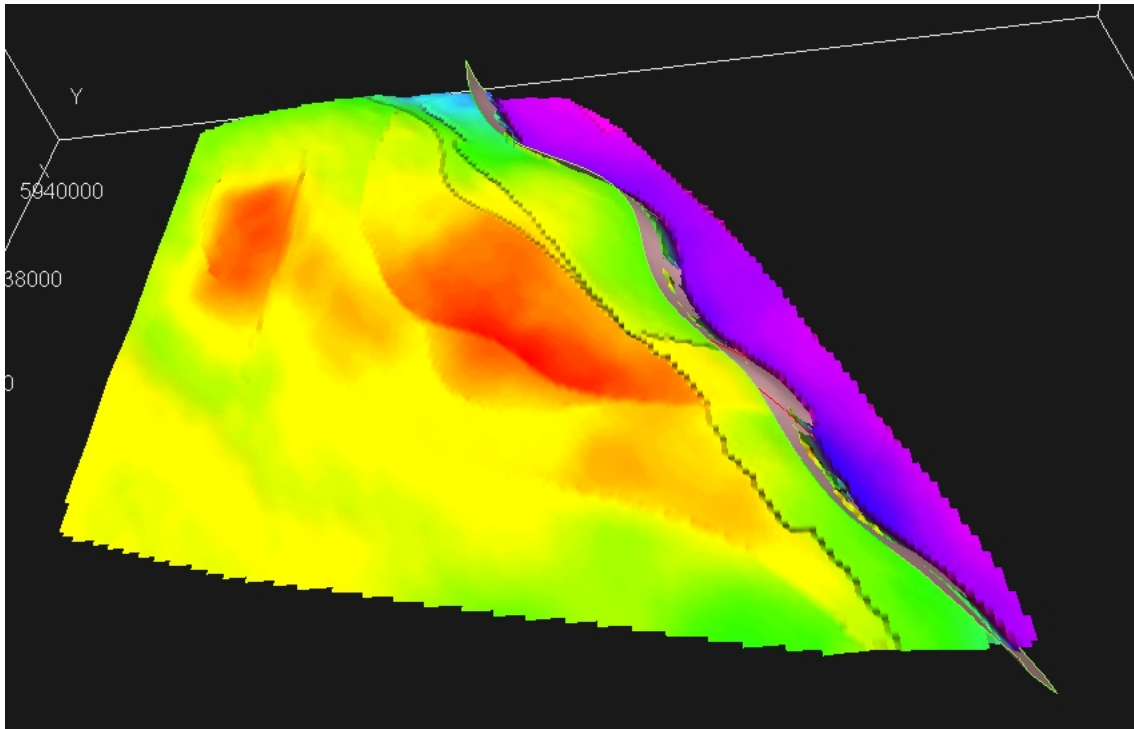


Figure 8: Top C horizon from Emerald 2009 project after the displacement has been changed at the fault F1.

Figure 8 shows an example from the Emerald 2009 project where the displacement of fault F1 has been decreased at the TopC horizon. This figure highlights one of the issues that need to be followed up from this prototype project. A sharp edge is shown on the left (upper) part of the horizon. Ideally the change in displacement should decrease continuously to zero towards the range of the displacement field. However the local grid defined by the fault surface defines a local domain with limited extent due to the size of fault surface. Points to the left of the sharp edge are not changed even if they are inside the range because they fall outside the local domain. The smaller the dip angle the bigger this issue will be, but it can be solved by extending the fault surface up and down. How this extension is to be performed is not obvious, but different approaches should be tested.

## 5 Uncertainty on fault surfaces

The fault can be seen as a mapping defined on the reference plane as shown in Figure 8.

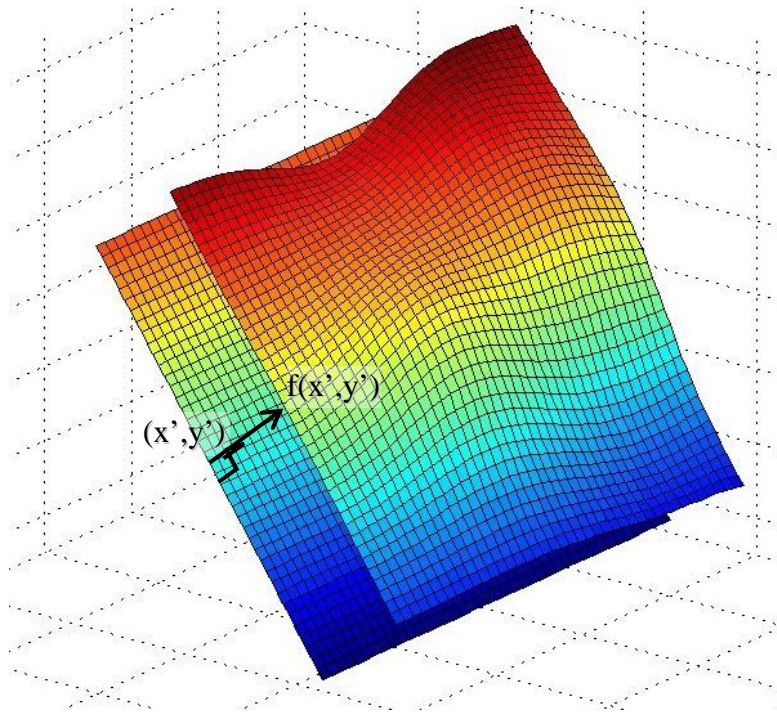


Figure 8 Mapping defining the fault surface.

A possible way to model uncertainty on the position of the fault surface is to consider the deviations as:

$$f(x', y') = \beta_0 + \beta_1 \cdot x' + \beta_2 \cdot y' + \varepsilon(x', y'),$$

where the coefficient  $\beta_0$  determine translation and the coefficients  $\beta_1$ , and  $\beta_2$ , determine the rotation of the reference plane. The residual term,  $\varepsilon(x', y')$ , is an arbitrary deviation from this plane. Uncertainty can be added by treating the coefficients as Gaussian distributed stochastic variables and the residual as a Gaussian random field. Standard techniques for prediction (kriging) and conditional simulation (Monte Carlo) can be used in the  $(x', y')$ -coordinate system.

The best approach is probably to assume a prior Gaussian distribution on the coefficients. Wells crossing the fault (and well paths near the fault) can be used to determine the posterior Gaussian distribution of the coefficients and the shape of the residual.

Note that more complex solutions are also possible:

$$f(x', y') = \beta_0 + \beta_1 \cdot x' + \beta_2 \cdot y' + \beta_3 \cdot x'^2 + \beta_4 \cdot y'^2 + \beta_5 \cdot x' \cdot y' + \beta_6 \cdot g(x', y') + \varepsilon(x', y'),$$

where  $g(x', y')$  is any known function such as a spline surface.

## 6 Integration with existing Havana functionality

The main goal is to make it possible to use the new surface based fault format in the cases where presently the pillar-based PFM format is used in Havana.

### 6.1 Modeling of subseismic faults

The subseismic faults will still be modeled as elliptic faults. However, the location and size of these faults can be conditioned on a set of mother faults. These mother faults are usually of seismic scale, and are presently modeled using the PFM model.

To be able to use faults on the new fault format, the new fault format has to be integrated with the already existing Havana fault model. Information used in the conditioning of subseismic faults is fault location, and fault displacement at the fault surface.

### 6.2 SUM

The structural uncertainty modeling will be done using a different algorithm than the present implementation in Havana, since the present implementation requires restoration of the surfaces using the inverse fault operator. One way of implementing the structural uncertainty modeling is to model the uncertainty on the fault surfaces, and horizon surfaces individually, but in a consistent manner.

### 6.3 Fault sealing properties

Modeling of fault sealing properties requires that the fault also is represented as an ECLIPSE fault. This can both be done by RMS, or internally in Havana. Further information about fault displacement is needed for calculations of clay and gauge smear.

### 6.4 Faulting and restoration of ECLIPSE grids

Faulting and restoration of ECLIPSE grids based on faults in the new format seems to be very complex. Havana needs to handle stair-cased much better than what is done today. Some form of regridding might be necessary to make this work.

### 6.5 FaultFacies specific functionality

As part of the FaultFacies project, some functionality was added to Havana:

#### 6.5.1 Local grid refinements along fault surfaces

One of the main parts of the FaultFacies workflow is the generation of the faultzone grid along the fault surface. Presently only faults that follow the grid pillars are supported. Staircased faults are not supported. This must be alleviated before it is possible to use to generate fault zone grids around faults in the new format.

#### 6.5.2 Strain calculations

Strain is calculated by derivation of the displacement field associated with the fault. So if we are able to generate the displacement field, it should be quite easy to use this field to obtain the strain.

## 7 Conclusion

The goal of the preliminary project was to define the integration with RMS, and experiment with different parameters that can be changed for the surface based faults, especially in the context of history matching.

We were able to import fault information from RMS, and to modify the fault location, and the dip and strike of the fault. The modified fault surfaces can be imported back into RMS.

We also started to experiment with the import of fault displacement information, and to generate a fault displacement operator that can be used to move horizons in accordance with changes in fault throw. We have made a simple model that works on a single fault, but further work is needed to find a better way to calculate the displacement field, and to be able to handle several interacting faults.

### 7.1 Issues with RMS integration

We encountered several issues with the RMS integration that must be handled in cooperation with Roxar:

- The export of the structural model should be a single operation in RMS. Exporting fault surfaces, fault lines, and the fault model file individually leads to a lot of extra work, and results in a lot of information must be given in the Havana model files. The current approach is also error-prone since there are some implicit requirements to the naming and location of the files exported from RMS.
- The ordering of the exported fault lines for a single horizon does not match the ordering of the faults in the fault model. Since there also is no tagging of the fault lines, this makes it difficult for us to match the fault lines with the corresponding faults.
- Information regarding if a fault is normal or reverse, used to determine which side of the fault that has gone up or down is not given in the fault model files, and must presently be given explicitly in the Havana model file.

## 8 Appendix A – Fault model file format

An example of a fault model file is given below. The file is outputted during the fault modeling job in RMS if the RMS\_FAULT\_MODEL\_FILENAME environment variable is set.

```
Version 1.0
#
# 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.
#
# The file is broken into three sections:
#
# The Bounding box of the model
# The faults
# The fault to fault truncations
#
#
# To read this file look for the keywords
'Version,BoundingBox,Faults,Truncations'
# Each section has a fixed format after that
#
# To generate this file you need to set the ennvirionmetal variable
'RMS_FAULT_MODEL_FILENAME'
# to the required file name and run the fault modeling job
#
#
#-----
# 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 1896.718491
# 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.
#
# format: Fault name azimuth dip
F1 60.914519 57.463298
F2 44.124806 90.000000
F3 231.591778 90.000000
F4 213.517344 90.000000
F5 208.746356 90.000000
F6 91.633742 90.000000
F7 79.927519 90.000000
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
F6 < F3
#
TruncationsEnd
#
#

```

## 9 Appendix B ModifySurfaceFault model file example

```
LEVEL_OF_INFORMATION 2 \  
  
ACTION      ModifySurfaceFault \  
  
INPUT_FAULTS surfaceFaults \  
  
HORIZON  
Horizons/TopC.storm faultLines/TopCFaultLines.txt \  
  
SCALE_DISPLACEMENT  
F3 1.2 1.0 500 1 ;  
F1 0.7 0.5 2000 1 \  
  
LOCAL_ROTATION  
CHANGE_AZIMUTH F2 45 ;  
CHANGE_AZIMUTH F1 90 ;  
CHANGE_DIP F3 45 \  
  
TRANSFORM  
TRANSLATE_X F6 1000 ;  
TRANSLATE_NORM F1 2000 ;  
TRANSLATE_NORM F2 2000 ;  
TRANSLATE_Y F3 50 \  
  
OUTPUT_FAULTS transsurfaceFaults \  

```

Example of a model file for the ModifySurfaceFault action is given above. This action can be used to translate the reference plane for the fault surface globally, rotate it locally or change the displacement of the faults.

The INPUT\_FAULTS command gives the folder containing the fault model described in the previous section.

The HORIZON command consists of one line for each horizon included in the fault model. Each line consists of two parameters: The filename for the horizon represented on STORM format and the filename for the fault lines represented on RoxarASCII-format. The moved horizons are written to the output directory with the same file name as the input horizons.

The SCALE\_DISPLACEMENT command gives the information necessary to change the displacement of the faults. Each line represents one fault and has five parameters:

- *Name of the fault*

- *Displacement change factor*. This is a positive number representing the factor that the displacement is multiplied by. > 1.0 means increasing displacement, < 1.0 means decreasing.

- *Vertical distribution of displacement.* This is a number in the interval [0.0,1.0] and represents the fraction of the displacement that goes down. 1.0 means only downward displacement and 0.0 only upward.

- *Range of displacement field.*

- *Fault type.* Indicator where 1 means normal and -1 reverse.

The LOCAL\_ROTATION command specifies changes to the azimuth and dip angles for the fault reference plane. Each line consists of three parameters. The first is either CHANGE\_AZIMUTH or CHANGE\_DIP, the second is the name of the fault and the last is the change of angle given in degrees.

The TRANSFORM command specifies global transformation of the faults. Each line consists of three parameters. Type of action, fault name and distance. The following types of action are possible: TRANSLATE\_X, TRANSLATE\_Y, TRANSLATE\_Z and TRANSLATE\_NORM. The first three means translation along the x-, y- and z-axis respectively, while the last means translation along the xy-projection of the normal vector for the fault.

The OUTPUT\_FAULTS command specifies the directory-name for the output from the program.

## Appendix C SurfaceFaultCenterLines model file example

```
LEVEL_OF_INFORMATION 2 \  
  
ACTION SurfaceFaultCenterLines \  
  
INPUT_FAULTS input_faults \  
  
FAULT_LINE_POLYGON  
  TopB_faultlines \  
  
FAULT_TYPE  
  F1 N ;  
  F2 N ;  
  F3 N ;  
  F4 N ;  
  F5 N ;  
  F6 N ;  
  F7 R \  
  
OUTPUT_FILE centerLines.dat \  

```

Example of a model file for the SurfaceFaultCenterLines action is given above. This action generates a center-line file that can be used as input to the Simulate action through the INPUT\_FAULT\_CENTER\_LINES command.

The INPUT\_FAULTS command gives the folder containing the fault model, which must have the same contents as for the ModifySurfaceFault action. The FAULT\_LINE\_POLYGON command gives the filename for a file with fault line polygons used to find the fault displacement.

The FAULT\_TYPE keyword denotes if the input faults are normal (N), or reverse (R). If nothing is given, the faults are assumed to be normal.

The OUTPUT\_FILE keyword gives the output file, which can be used as input to the Simulate action.