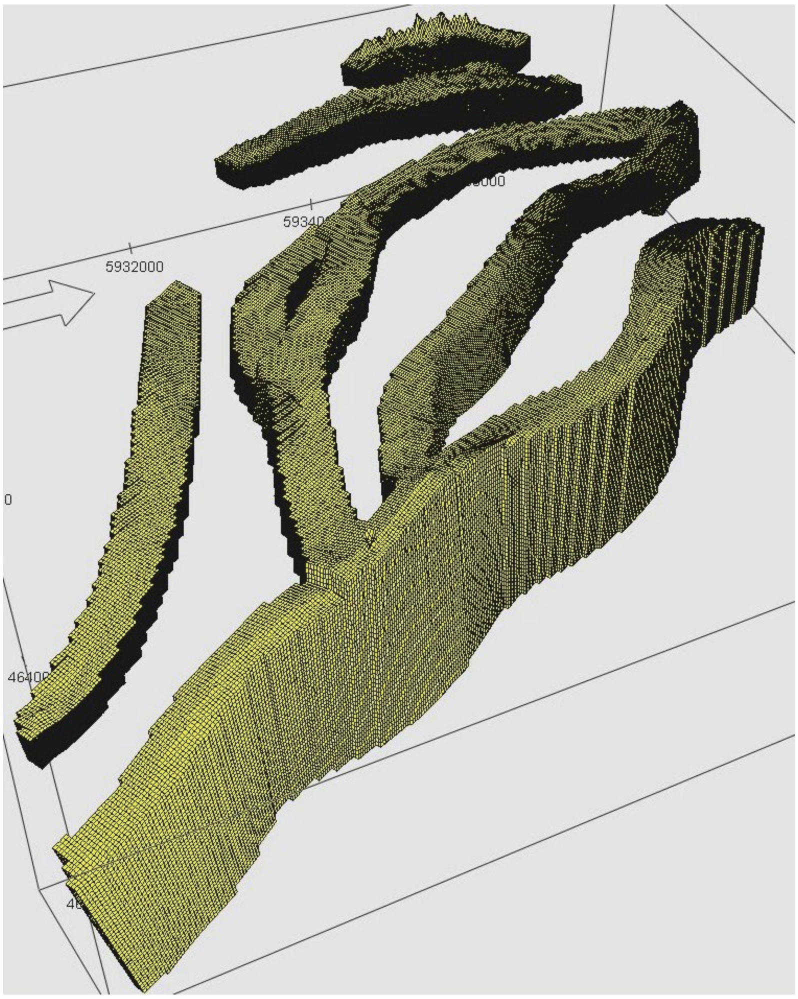
Norsk Regnesentral
NORWEGIAN COMPUTING CENTER

Note

Local grid refinements in Havana

Gridding of fault zones



Note no SAND/15/07

Authors Per Røe

Date 20/12 2007

About the authors

Per Røe is research scientist at the Norwegian Computing Center.

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.

Norsk Regnesentral
Norwegian Computing Center
Postboks 114, Blindern
NO-0314 Oslo, Norway

Besøksadresse
Office address
Gaustadalléen 23
NO-0373 Oslo, Norway

Telefon · telephone
(+47) 22 85 25 00
Telefaks · telefax
(+47) 22 69 76 60

Internett · internet
www.nr.no
E-post · e-mail
nr@nr.no

Title	**Local grid refinements in Havana**
Authors	**Per Røe**
Date	20/12 2007
Year	2007
Publication number	SAND/15/07

Abstract

Local grid refinements along fault traces are used for gridding of fault zones. The idea here is that the faults are not only planes, but instead occupy a volume. The grid inside the fault zone is finer since we want to model a higher degree of heterogeneities inside the fault zone than in the rest of the grid.

The generation of the local grid for the fault zones is implemented in Havana, using the ECLIPSE grid format. The ECLIPSE grid format is a standard grid format for modeling and simulating grids and Havana has extensive support both for this format, and for general fault modeling.

Keywords	Fault zone modeling, local grids
Target group	Project partners, NR, CIPR
Availability	Open
Project number	808005
Research field	Structural modeling
Number of pages	10

Contents

1 Introduction

This document describes the generation of a local grid refinement using Havana. The local grid refinements envelopes the faults, and is used as basis for the modeling of the fault zones.

2 Technical description

The local grid refinement is done in a corner point grid in the ECLIPSE format.

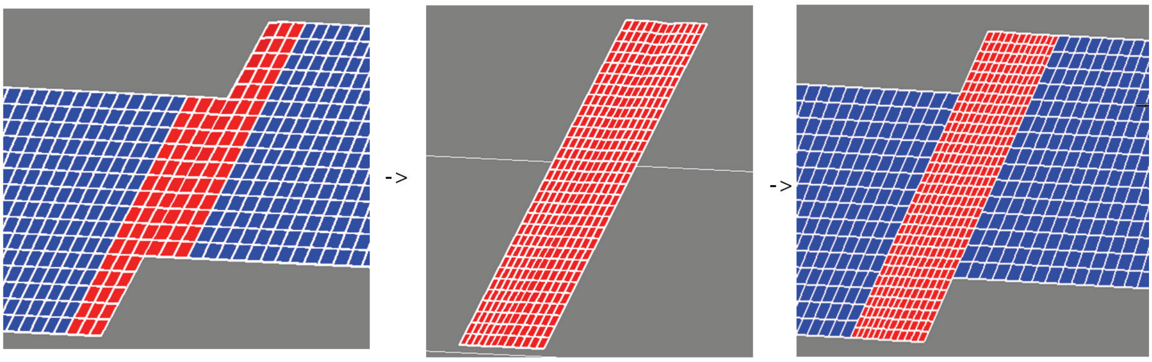


Figure 1. Local grid overview. The first figure shows the original coarse grid, with the desired fault zone in red. The second figure shows the refined local grid with continuous top and bottoms. The third figure shows the local grid merged into the coarse grid.

A simplified workflow is given below:

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 the top and bottom surfaces are smoothened.
3. Fault facies are modeled inside the local grid.
4. The local grid is merged together with the coarse grid.

2.1 Making the fault zone

Support for three different methods for defining the location of the local grid is implemented:

1. Explicit definition of the position of the local grid.
2. Using a FAULTED Boolean parameter.
3. Using the ECLIPSE FAULTS parameter, defining the fault traces.

The first method has the disadvantage that it gives a lot of manual work, making it extremely cumbersome, and virtually impossible to use on all but the simplest cases.

The second method works reasonably well, but since the crossings between faults is not well defined, the interpolation gives upper and lower surfaces that are the same for both faults, giving a too large local grid.

The solution to these problems is to give the fault location explicitly. Since we are working on the Eclipse grid, this is easiest done by using the FAULTS keyword. The algorithm is illustrated in Figure 2-Figure 4.

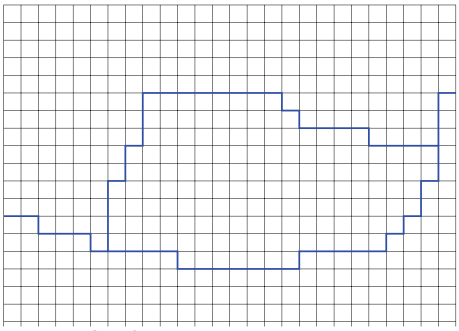


Figure 2. Grid from above, the blue lines represent the fault surfaces.

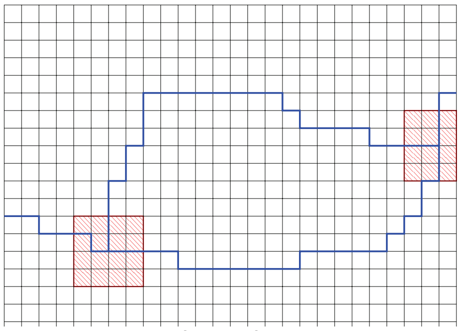


Figure 3. First we refine the fault intersections.

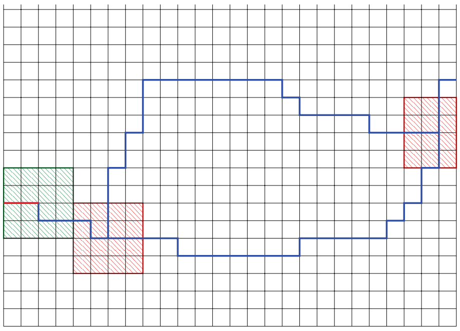


Figure 6. First section of first fault

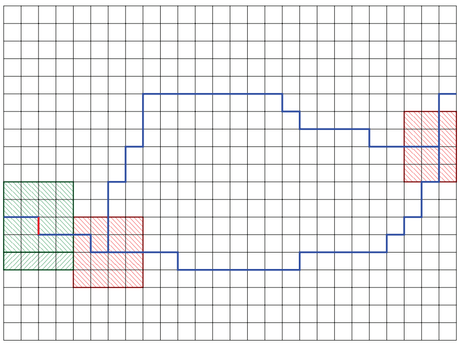


Figure 5. Second section of first fault. We only refine area not covered zone around first section.

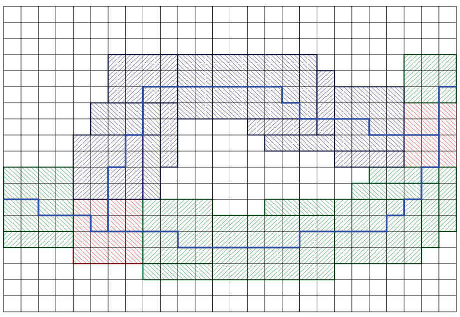


Figure 4. Final result

The fault intersections are taken first to ensure that the area around the fault intersections have smooth top and bottom surfaces. After this we refine the defined area around each segment of the faults, but making sure that each area only is refined once.

The top and bottom surfaces of the grid are found by first finding the top and bottom cell in each area, and then interpolate the surface from this cell. The top and bottom cells are found by checking if there are some gaps between neighbor cells, indicating faults, and when doing the interpolation we check for gaps to make sure that we only interpolate when needed.

The exported local grid consists of a grid made up by all the single local grids that were made above. Since an Eclipse grid must be square, the rest of the grid is set inactive.

2.2 Merging of local grid and coarse grid

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.

2.3 Generation of upper and lower intensities

The LGREclipse action can also generate upper and lower intensities for the local grid. The upper and lower intensities are scaled according to distance to fault surface, position in grid (near bottom or top), and the fault throw.

3 Model file

Below is an example model file using the LGREclipse action to generate a local grid around the fault zone.

```
ACTION          LGREclipse \

! Reading the Eclipse grid and permeability data:
INPUT_ECLIPSE  grids/coarsegrid.grdecl \

! Local grid refinement, x, y and z direction
GRID_REFINEMENT  FAULTS  2 2 2  3 \

GENERATE_UPPER_LOWER \

EXPORT_LOCAL_GRIDS grids/localgrid_nostrain.grdecl \
```

The INPUT_ECLIPSE keyword specifies the input file in ECLIPSE format. This file should contain ECLIPSE fault information using the FAULTS keyword. The GRID_REFINEMENT keyword gives the parameters for the local grid refinement. The first parameter specifies the method for finding the local grid. This should usually be FAULTS. Other options are possible, and discussed in Making the fault zone. The next three numbers are the grid refinement in x, y and z direction, while the last number is the number of cells on each side of the fault surface that shall be included in the local grid.

The GENERATE_UPPER_LOWER keyword specifies that UPPER and LOWER parameters should be calculated, while the EXPORT_LOCAL_GRIDS keyword gives the output ECLIPSE file for the local grid.

The LGREclipse action can also be used for merging a local grid with a coarse grid. An example of a model file using this functionality is given below:

```
ACTION          LGREclipse \

! Reading the Eclipse grid and permeability data:
INPUT_ECLIPSE  grids/coarsegrid.grdecl \

! The output grid:
OUTPUT_ECLIPSE grids/mergedgrid.grdecl  \

IMPORT_LOCAL_GRID LGR1 \
IMPORT_INPUT_ECLIPSE grids/localgridfinal.grdecl \
```

The INPUT_ECLIPSE keyword specifies the original coarse grid, which also must have been used when originally generating the local grid. The OUTPUT_ECLIPSE keyword specifies the output file for the merged grid. IMPORT_LOCAL_GRID and IMPORT_INPUT_ECLIPSE specify the name that will be used for the local grid and the file containing the local grid. The local grid is added to the merged grid as a local grid refinement, identified with the given name.

4 Shortcomings

There are still some areas where the local grid refinement can be improved, including the following:

- The method for interpolating the top and bottom surfaces is still pretty crude, giving irregular surfaces in some cases, especially if the original surface curves a lot.
- The grids inside the fault zone are not always continuous; the corners of neighbor cells do not always correspond. Especially the case with grids that contain a varying number of active grid cells in the different columns, give complicated grids inside the fault zone.
- The calculating of the UPPER and LOWER intensities is very slow, and gives intensities that does not correspond well with reality. In the future the FaultZone action will be used to calculate the UPPER and LOWER intensities instead.