DFN Generator Petrel plugin Installation and Use guide

Michael Welch, Michael Lüthje and Simon Oldfield

07/09/2021

Installation

There are two methods of installing the Petrel DFN Generator plugin:

- If you have the Ocean API installed and wish to make changes to the DFN Generator code, you can install the plug-in through Ocean.
- If you do not have the Ocean API installed or do not wish to make any changes in the code, you can install the plug-in directly using the .pip installation files.

Installation using Ocean

If you have the Ocean API installed and linked to Petrel, you can install the plug-in to Petrel by recompiling the DFNGenerator_Ocean project. It is necessary to shut down Petrel before recompiling the code. The plug-in should then be available next time you restart Petrel.

You may need to change some of the properties and references in the DFNGenerator_Ocean project to point to the correct version of Petrel. The following properties and references are version specific:

DFNGenerator_Ocean project properties:

- Change the code in the Post-build event command line to point to the correct version of Petrel.
- Change the folder names to point to the correct version of Petrel.

DFNGenerator_Ocean project references: Change the following references to point to the correct version of Petrel:

- Slb.Ocean.Basics
- Slb.Ocean.Coordinates
- Slb.Ocean.Core
- Slb.Ocean.Data
- Slb.Ocean.Geometry
- Slb.Ocean.Petrel
- Slb.Ocean.Petrel.Configuration
- Slb.Ocean.Petrel.DomainObject.FrameworkModeling
- Slb.Ocean.Petrel.DomainObjectExtensions
- Slb.Ocean.Petrel.DomainObjectExtensions.Prestack
- Slb.Ocean.Petrel.Modeling
- Slb.Ocean.Petrel.Seismic
- Slb.Ocean.Petrel.UI.Controls
- Slb.Ocean.Units

Installation from pip files

If you do not have the Ocean API installed, the current version of the DFN Generator Petrel plug-in can be installed directly using the .pip installation files, available from the same repository as the source code. You will need to download the correct .pip file for your version of Petrel.

After starting Petrel, open the Ocean Plugin Manager (File – Options – Plugin manager), as shown in Figure 1. Then click on Install plug-in, locate and open the .pip file. It is advisable to copy the .pip file locally onto the C drive before installing, as installing over a network can cause problems. Please make sure to select the .pip file corresponding to the Petrel version being used such as DFN_2017.pip for Petrel 2017.

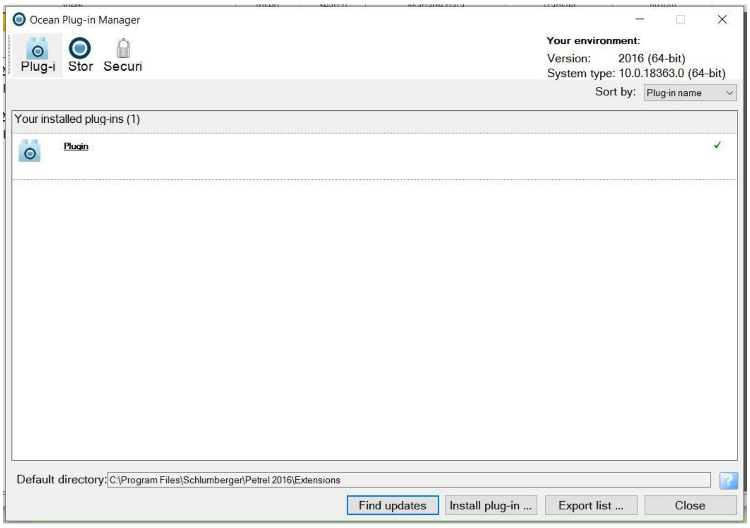

Figure 1: Ocean Plug-in Manager window.

Close the Plug-in Manager and restart Petrel and the plugin will be installed and located in the "Process Pane".

Running Models

In the Petrel processes pane, double-click on DFNGenerator in the Plug-Ins folder (Figure 2) to open the DFN Generator module (Figure 3). This has several tabs of user-adjustable settings; however most of these can be left at their default values. In general you will only need to use the “Main settings” tab to run models.

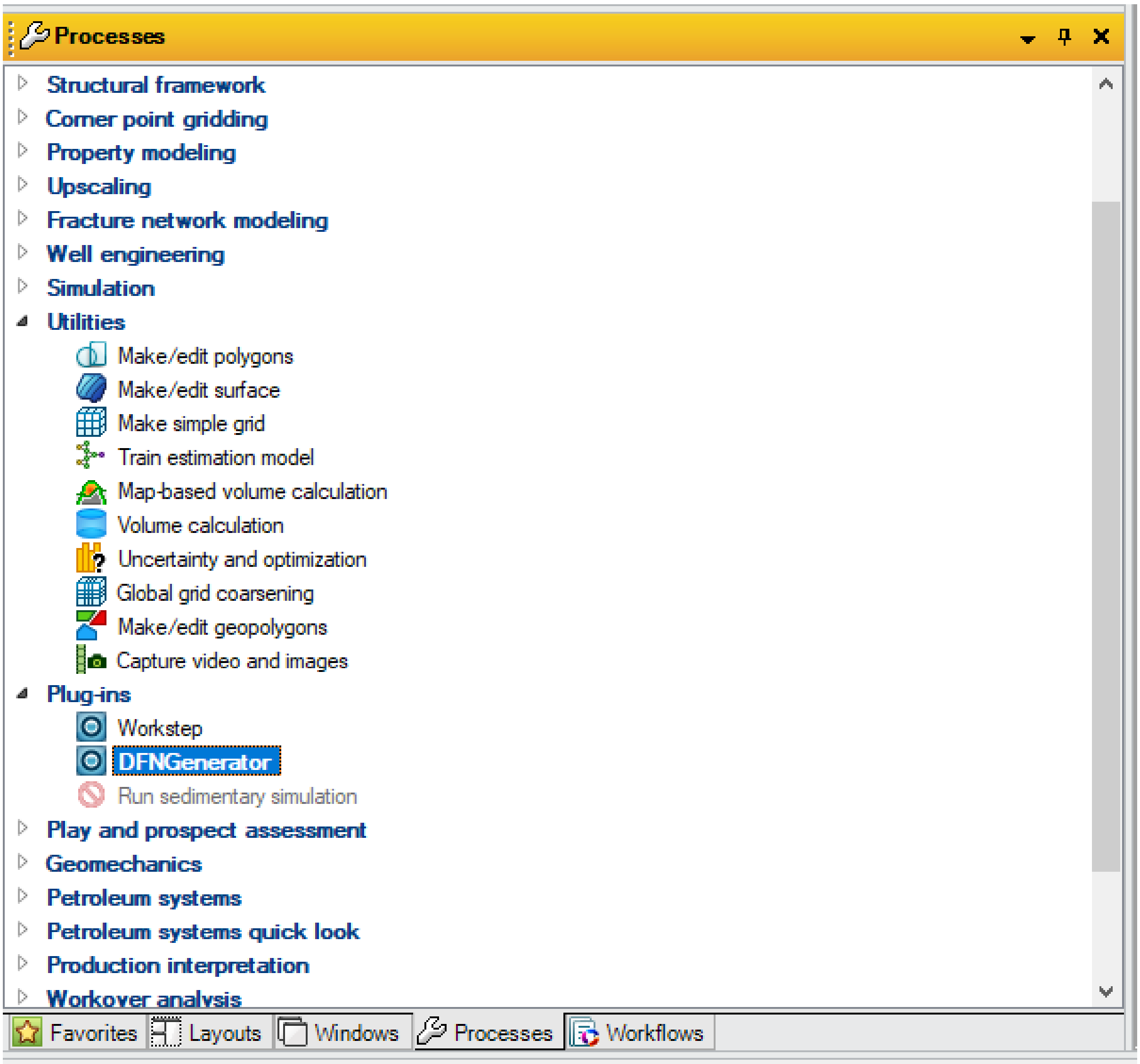

Figure 2: Opening the DFN Generator plug-in.

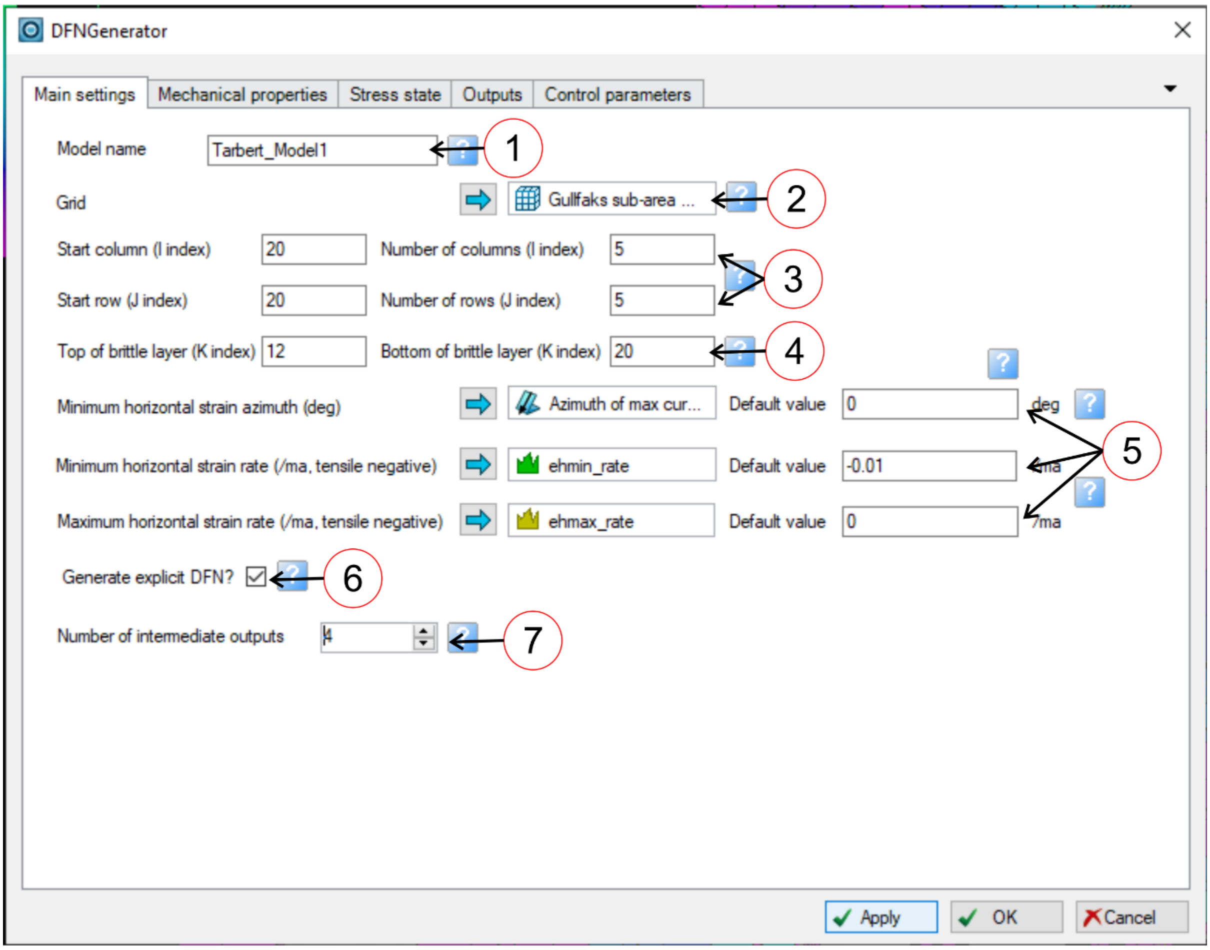

Figure 3: The Main Settings tab of the DFN Generator plug-in, populated with typical properties and values.

Main settings

The only input data that *must* be specified to run the DFN Generator is the Grid object; all other settings can be left at their default values. However you will generally need to enter data for the other settings on the "Main settings" tab before running a model. These settings are shown on Figure 3:

- **Model name (1):** Assigning each fracture model a unique name will help to identify the model output. This name will be applied to the output folders containing the implicit fracture properties as well as the DFN objects. NB The parameters used to generate each fracture model are recorded on the comments tab of the output folders and DFNs.
- **Grid (2):** The Petrel grid object representing the model geometry, and containing the input properties. Select the required grid object in the Models pane and click the blue arrow to drop it in the box (Figure 4).
- **Rows and columns (3):** Use the column and row controls to limit the model to a part of the overall grid. Leave these blank to run the model on the entire grid. NB since the model may take several hours to run on a large grid, it is recommended to run a test on a small part of the grid beforehand.
- **Brittle layer (4):** Use these controls to specify the brittle layer containing the layer-bound fractures, by inputting the K-index of grid layers at the top and bottom. Note that the brittle layer can span multiple grid layers, and need not correspond to stratigraphic layers or zones. If the brittle layer spans multiple grid layers, these will be amalgamated into "cell stacks" for the purpose of calculation. In the output data, the same fracture density values will be applied to every cell in the stack; these represent the fracture densities across the entire brittle layer.

It is only possible to model one brittle layer at a time. However it is of course possible to rerun the model several times to generate multiple DFNs for different brittle layers.

- **Horizontal strain (5):** Fracturing is driven by a biaxial applied horizontal strain. This comprises three components: a minimum and maximum strain rate and an orientation. Each component can be specified either as a grid property or as a default value applicable to all cells. Specifying a strain component as a grid property allows modelling of lateral variations in the strain orientation and rate. To do this, select the required grid property in the Models pane and click the blue arrow to drop it in the appropriate box (Figure 5). If no grid property is specified (i.e. the box is blank), the model will apply the value specified in the default box to all cells in the model; this value will also be applied to any cells where the specified grid property is null or undefined.

  Note that, following geomechanical convention, extensional strain is represented by negative values and compressional strain by positive values. The minimum horizontal strain rate therefore represents the maximum extension rate, and should have predominantly negative values; typical values are in the range -0.001 to -0.1/ma. The maximum horizontal strain rate may be negative, zero or positive. The minimum horizontal strain azimuth represents the direction of maximum extension, and controls the orientation of the fractures in the DFN: the primary fracture set forms perpendicular to this and the secondary fracture set forms parallel to this. If a grid property is specified, the fractures will bend at cell boundaries to reflect changes in the strain orientation. Note that the strain azimuth must be specified in degrees; the DFN Generator does not perform automatic conversion of data specified in radians.

  Horizontal strain data can be derived from any source, and the best way to generate it will vary based on the geological setting and available data. However preliminary horizontal strain data can often be generated quickly and easily from the horizon curvature, allowing rapid generation of first-pass fracture models. The procedure for this is described below.

- **Generate explicit DFN? (6):** By default, the DFN Generator generates both implicit fracture data (e.g. fracture density and porosity values, as grid properties) and explicit DFNs (geometric representations of the fracture network, as Petrel DFN objects). However if the explicit DFN is not required, it is possible to reduce the runtime by generating only the implicit fracture data.
- **Intermediate outputs (7):** One advantage of forward modelling fracture growth is we can easily generate a series of models representing intermediate stages in the evolution of the fracture network. This can be useful for risk analysis, as it provides multiple, geologically realistic fracture models representing low, mid and high case scenarios. Use this control to set the number of intermediate models to be output. A separate and clearly labelled folder containing implicit fracture data, and a clearly labelled Petrel DFN object, will be output for each intermediate stage. If this is set to 0, output will only be generated for the final, fully developed fracture network.

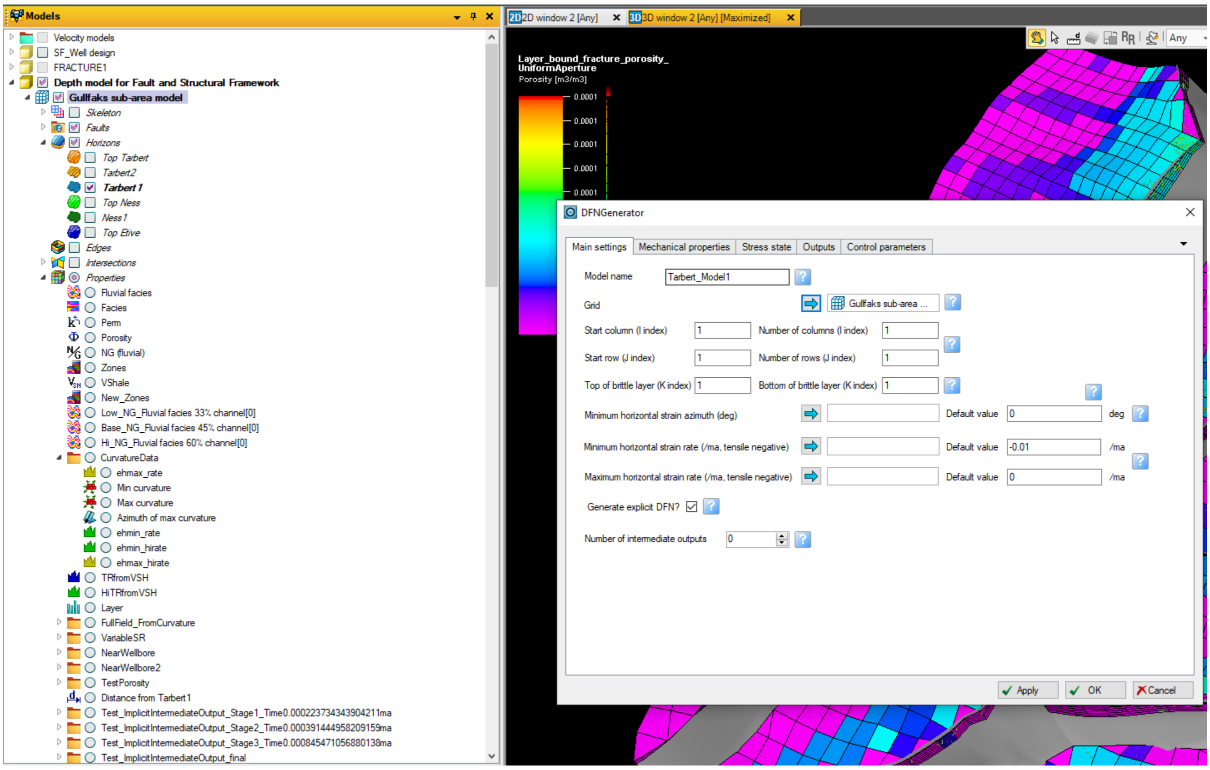

Figure 4: Select a grid object.

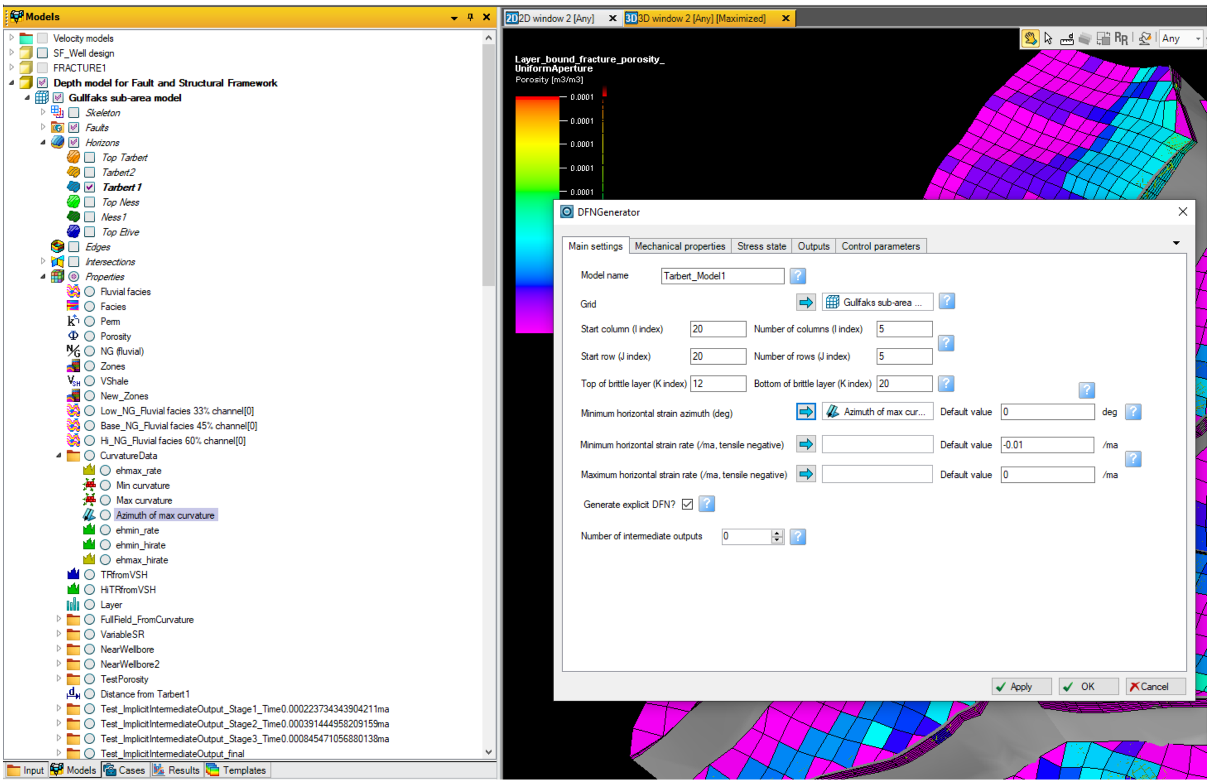

Figure 5: Select a grid property representing the minimum (most tensile) strain orientation. The model has already been set to run on a small 5x5 cell portion of the total grid, with a brittle layer spanning grid layers 12 to 20 inclusive.

Mechanical properties

By default, the mechanical properties are set to typical values for brittle limestone or chalk. However if more detailed or specific mechanical data is available, this can be used instead. To do this, go to the Mechanical Properties tab (Figure 6).

As with the horizontal strain data, mechanical properties can be can be specified either as a grid property or as a default value applicable to all cells. Specifying a mechanical property as a grid property allows modelling of lateral variations in the property, if these are known (e.g. where Young's Modulus data is available from inversion of a seismic cube). To do this, select the required grid property in the Models pane and click the blue arrow to drop it in the appropriate box (Figure 6). If no grid property is specified (i.e. the box is blank), the model will apply the value specified in the default box to all cells in the model; this may be more appropriate is more limited mechanical property data is available (e.g. where mechanical property measurements are available from lab tests on a core sample). The default value will also be applied to any cells where a specified grid property is null or undefined.

Note that mechanical property data must be specified in SI units: especially Pa (*not* GPa) for Young's Modulus and J/m2 for the crack surface energy[1]. The DFN Generator does not perform unit conversions, so the grid properties must be converted to SI units beforehand (use the Property Calculator). It is advisable to use the General template for properties rather than property-specific templates, to avoid erroneous unit conversions.

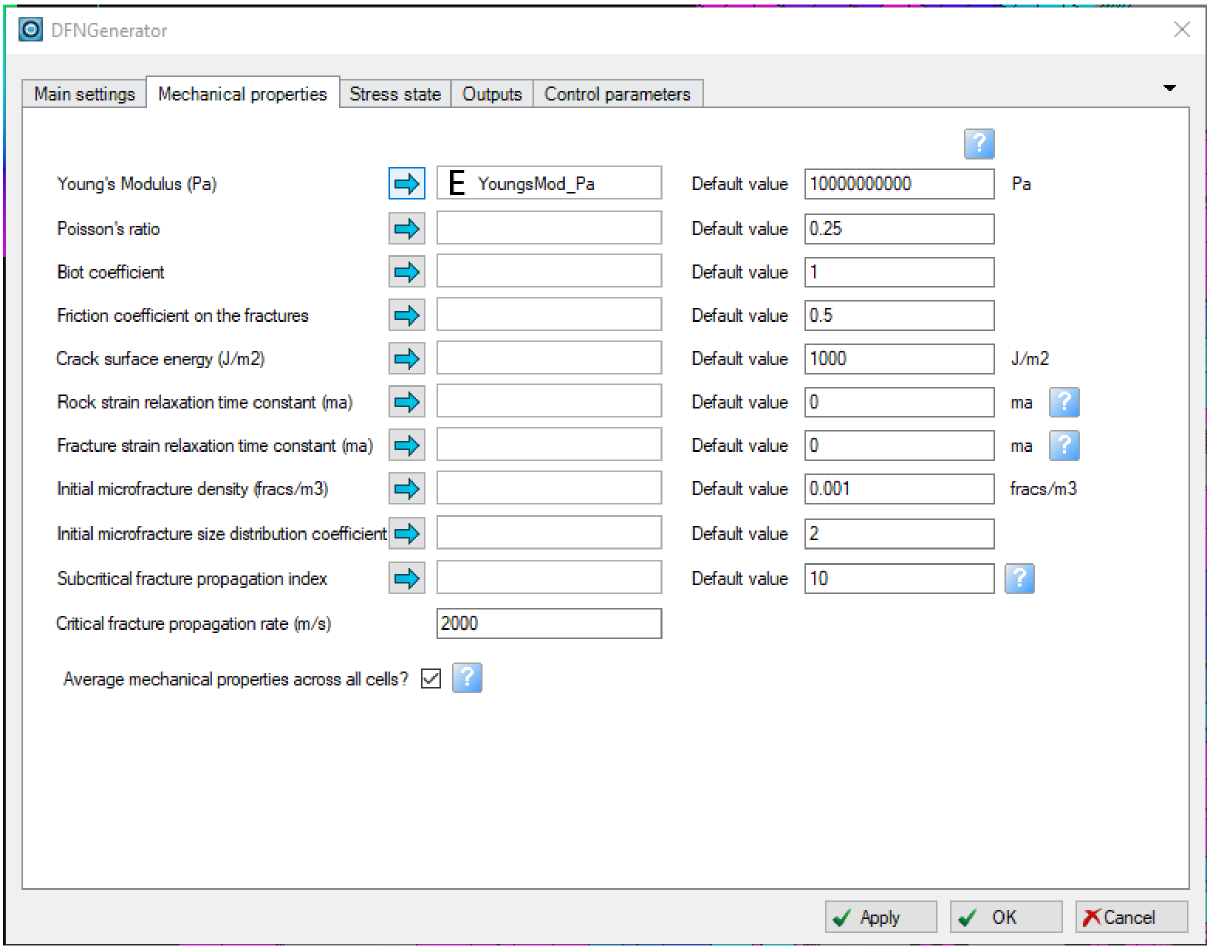

Figure 6: The Mechanical Properties tab.

[1] Except for strain relaxation time which is specified in ma.

Stress state

By default, the in situ stress at the start of the deformation episode is set to the viscoelastic equilibrium stress state for hydrostatic fluid pressure at the current depth of burial of the brittle layer. The vertical effective stress is therefore set to the current lithostatic stress minus the hydrostatic fluid pressure, and the horizontal effective stress is equal to the vertical effective stress. This stress state can be adjusted by changing the following settings on the Stress State tab (Figure 7):

- **Depth at time of deformation (1):** If deformation occurred at a shallower (or deeper) depth than the current depth of burial of the brittle layer, set this depth here (in metres, positive downwards). The initial in situ stress will be adjusted to match the equilibrium stress state at the specified depth of burial.
- **Initial fluid overpressure (2):** If the brittle layer was overpressured at the time of deformation, set the overpressure here. A high fluid overpressure will promote the development of Mode 1 dilatant fractures rather than Mode 2 shear fractures.
- **Initial stress relaxation (3):** This controls the initial horizontal stress, prior to deformation. A value of 1 represents viscoelastic equilibrium, $\sigma_{h0}' = \sigma_v'$; a value of 0 represents elastic equilibrium, where $\sigma_{h0}' = \nu/(1-\nu) * \sigma_v'$. If blank, the initial horizontal stress will be set to the critical stress state (i.e. the Mohr-Coulomb failure stress) for each cell.

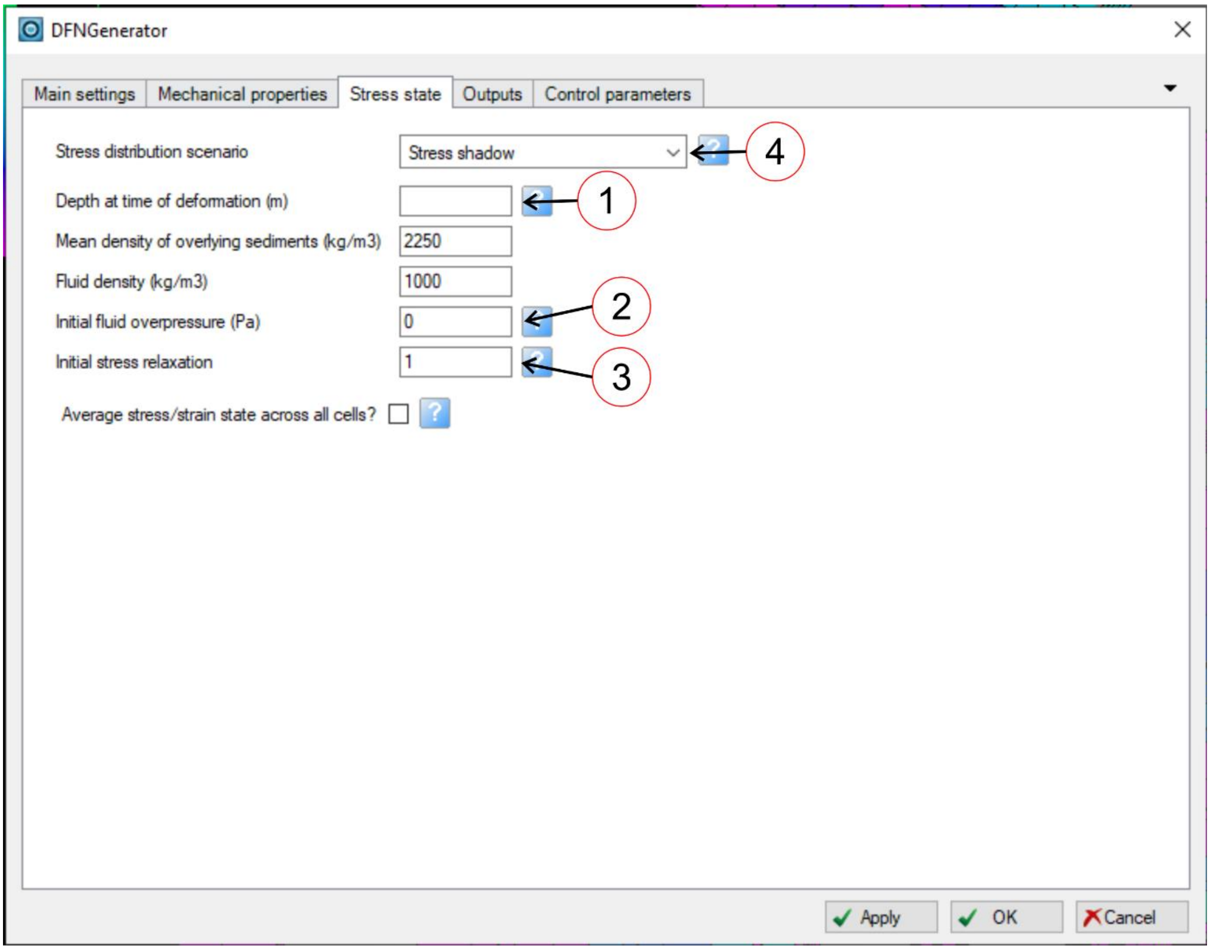

Figure 7: The Stress State tab.

It is also possible to modify or turn off the stress shadow effect by changing the stress distribution scenario (4). This controls the spacing of the fractures. At present two settings are available:

- **Stress shadow:** fractures are surrounded by an elastic stress shadow, in which the applied strain is accommodated by displacement on the fractures rather than elastic strain in the rockmass. Stress shadows of adjacent fractures may not overlap, and propagating fractures will be deactivated if they overlap the stress shadow of another fracture. This sets a minimum spacing for the layer-bound fractures, which is proportional to the layer thickness.
- **Evenly distributed stress:** There are no stress shadows around the fractures and therefore no minimum fracture spacing; instead, displacement on the fractures reduces the elastic strain throughout the rockmass. This setting can lead to a large number of fractures being generated.

Outputs

This tab allows the user to specify additional data to be output for both the implicit fracture models and the explicit DFNs (Figure 8). It is also possible to write output directly to file.

If fracture porosity is output, it is necessary to specify a method for calculating the fracture aperture. This will also determine the aperture property assigned to individual fractures in the explicit DFNs. Four methods are available:

- **Uniform aperture:** All fractures are assigned an arbitrary user-specified aperture. Different apertures can be specified for fractures oriented perpendicular to the minimum (i.e. most tensile) and maximum horizontal stress.
- **Size-dependent aperture:** Fracture aperture is proportional to the minimum fracture dimension (fracture diameter for microfractures, layer thickness for layer-bound fractures). Different scaling factors can be defined for fractures striking perpendicular to the minimum (i.e. most tensile) and maximum horizontal stress.
- **Dynamic aperture:** Calculates the equilibrium elastic aperture for dilatant fractures subject to a tensile normal stress. An arbitrary user-defined multiplier can also be applied. The calculation is based on the in situ stress state at the end of deformation; NB if this is compressive (e.g. for Mode 2 shear fractures), the resulting fracture porosity will be 0.
- **Barton-Bandis aperture:** Calculates the aperture for shear fractures subject to a compressive normal stress using the Barton-Bandis formula (Bandis et al. 1983, Int J Rock Mech, Min Sci & Geomech Abs 20, 249-268). The calculation is based on the in situ stress state at the end of deformation and various parameters related to the fracture morphology; the default values given here are typical for shear fractures in sedimentary rocks.

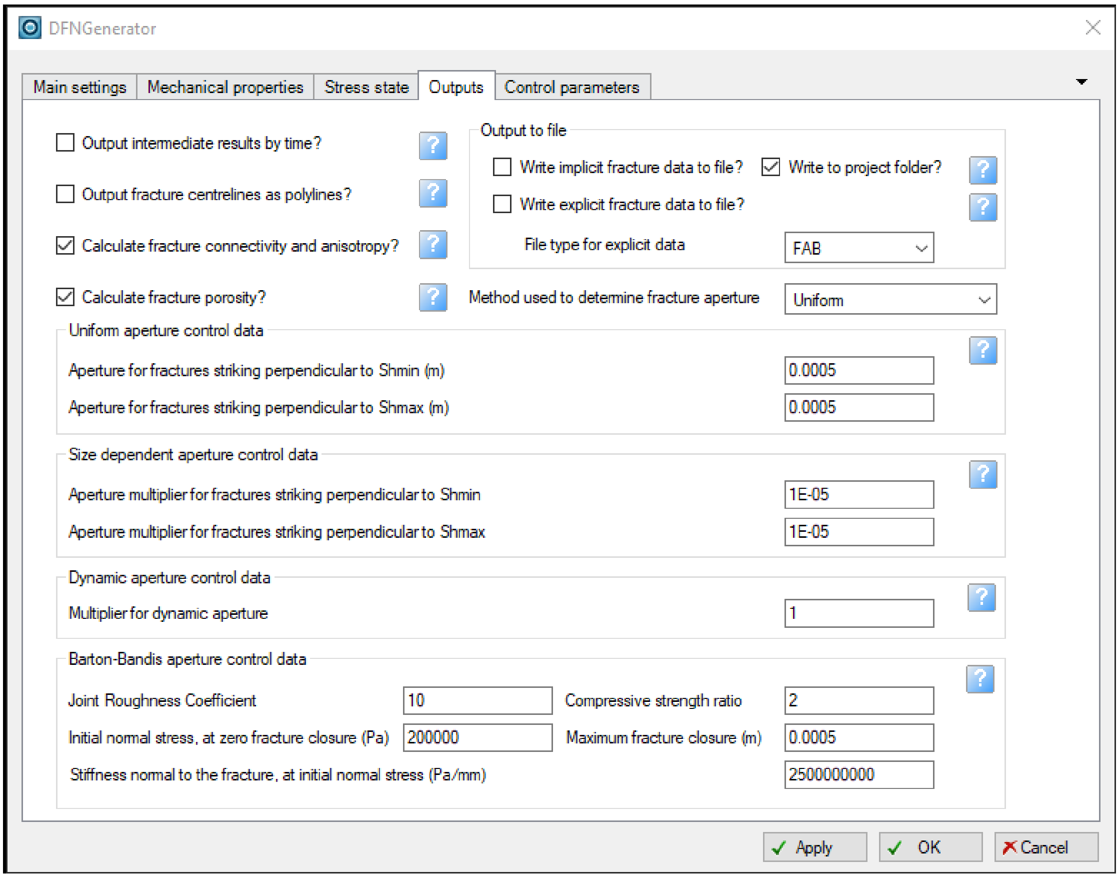

Figure 8: The Outputs tab.

Control parameters

This tab allows the user to adjust the parameters that control the calculation (Figure 9). In most cases these should be left at the default values. However some controls that may be useful include:

- **Number of fracture sets (1):** By default the fracture models include 2 fracture sets, oriented perpendicular to the minimum and maximum horizontal stresses. This is sufficient for modelling a single stage of tectonic deformation in intact rock. However to model polygonal or strike-slip fractures, or if there have been previous deformation episodes (so that there are pre-existing fractures oblique to the principal horizontal stresses), it is necessary to include more fracture sets in the model. Increasing the number of fracture sets will however increase the runtime of the model and the amount of output data. A model with 6 fracture sets (oriented at 30° intervals from the minimum horizontal stress direction) allows accurate representation of polygonal and strike-slip fractures with manageable runtimes. If more than 2 fracture sets are used, it is also recommended to check the microfractures against stress shadows from all sets (2), to allow more realistic placement of seed macrofractures.
- **Fracture mode (3):** By default the model will include both Mode 1 dilatant and Mode 2 shear fractures, although typically one of these modes will predominate, depending on which is energetically optimal. However it is possible to force the model to contain only Mode 1 dilatant or Mode 2 shear fractures.
- **Layer thickness cutoff (4):** Since the fracture spacing is proportional to the thickness of the brittle layer, then in cell stacks where it becomes very thin (e.g. where there is pinch-out), an excessive number of fractures may be generated which will increase model runtime. To avoid this we can specify a minimum layer thickness cutoff; the explicit DFN will not be generated in cell stacks thinner than this cutoff. 1m is a suitable thickness cutoff for full-field models; however it may need to be adjusted for small-scale models.
- **Minimum radius for microfractures to be included in explicit DFN:** By default the explicit DFN will contain only layer-bound macrofractures; circular microfractures are represented only in the implicit fracture model. However it is possible to include the larger microfractures in the DFN as well, by specifying a minimum microfracture radius (which should be between 0 and half the layer thickness). Leave this blank to exclude microfractures from DFN. Note that if microfractures are included the DFN can become very large; it is recommended to do this only in local models (e.g. near wellbore models) comprising a few cells.

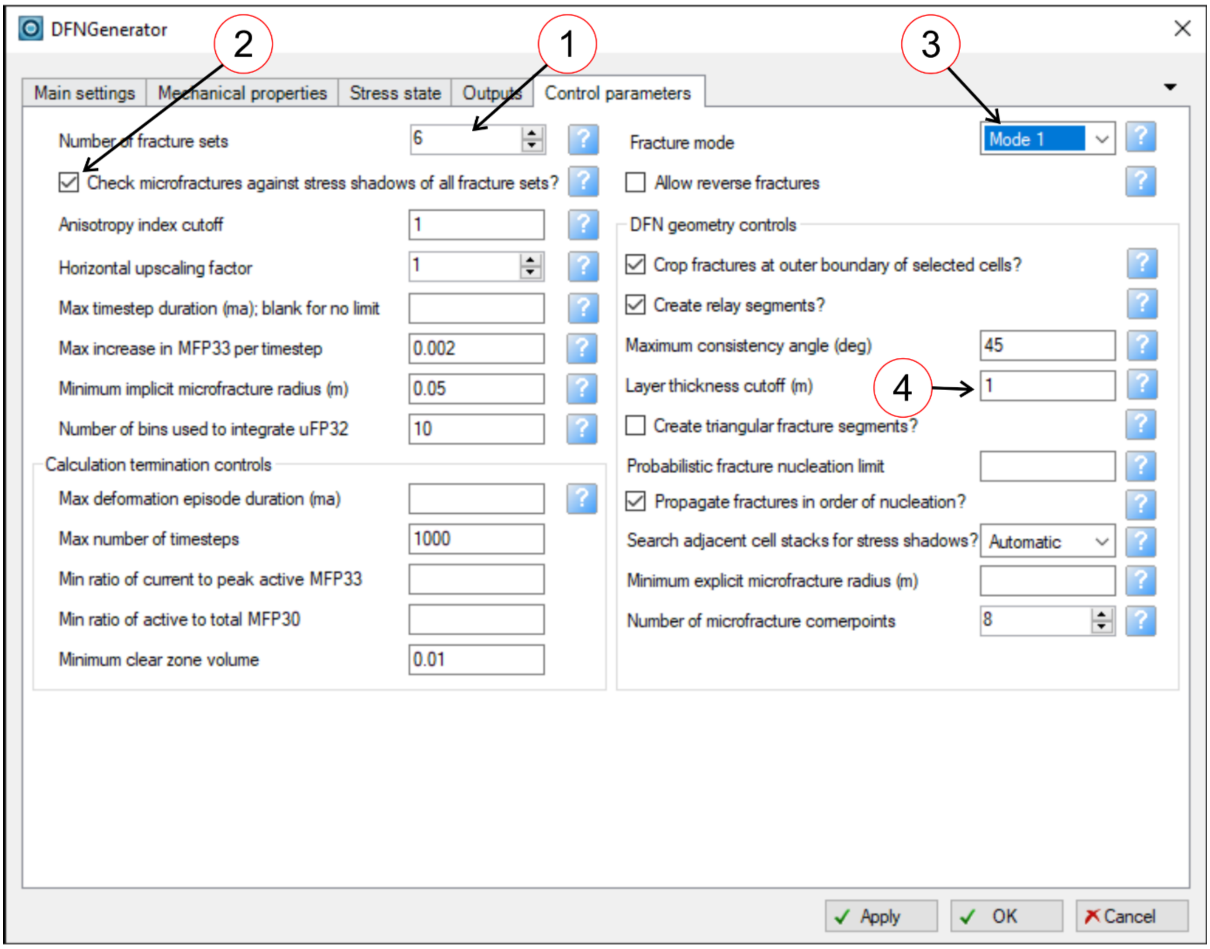

Figure 9: The Control Parameters tab.

Running the model

Once the input data has been entered, click the “Apply” or “OK” buttons to run the model (both buttons have the same effect, except that “OK” closes the module after running). Updates on the different stages of the calculation will be given in the Message Log window and progress will be shown on the Progress Bar. It is possible to abort the calculation by clicking the “Stop” button, although any results will be lost. A full model on a large grid may take several hours to complete, but a model of a small subsection of a grid (e.g. 5x5 cells) should run in a minute or less.

When the model has finished running, the output data should be visible in the Models pane, below the selected Grid object (see below).

Generating strain data from curvature

As noted above, the horizontal strain data can be derived from any source. The most appropriate source will depend on the geological setting and the mechanism thought to be responsible for fracturing: for example a regional tectonic fracture set could be modelled using a uniform extensional strain (specified using default values), fractures caused by folding or diapirism could be modelled using a structural restoration or backstripping algorithm to calculate strain, while fault-related fractures could be modelled by using the Petrel Tectonic Model module to calculate local strain around the main faults.

For a first-pass fracture model, however, estimates of horizontal strain can be generated quickly and easily from the horizon curvature, using existing Petrel functionality (NB this requires a license for the Petrel structural analysis package). This can be used to very rapid results for a preliminary assessment, although it should be noted that curvature gives only an approximation for strain, and one of the methods described above should be used for more detailed studies. The procedure for generating strain data from curvature is as follows:

1. Select a horizon at or near the top of the brittle layer. Convert this to a Regular Surface (double-click on the horizon in the Models pane to open the Settings dialog, go to the Operations tab and click on Make Surface; the surface should then appear at the bottom of the Input pane).
2. It may be necessary to smooth the surface, to remove curvature resulting from noise or very localised structures. Double-click on the surface in the Input pane to open the Settings dialog, go to the Structural Analysis tab and select Surface Smoothing from the Data Clean-up and Processing folder (). Good results can generally be obtained with I and J region half-widths of 2, the Combo Smooth averaging method, Smooth Edges, and 2 passes (Figure 10). Click on Run to carry out the smoothing operation. This will generate a new surface in a folder labelled “Smoothed Surfaces”.
3. Double-click on the (smoothed) surface to open the Settings dialog, go to the Operations tab and open the Curvature Operations folder to find the algorithms for calculating surface curvature. The three required horizontal strain components can be generated using the Min Curvature, Max Curvature and Azimuth of Max Curvature algorithms (select each icon in turn and then click on Run; the outputs should appear as three surface attributes, as shown on Figure 11).
4. The surface attributes can now be converted to grid properties using the Geometric Modelling module on the Structural Modelling tab. Open this module and select the “Constant or surface in segments or zones” method. Generally it is easiest to write the curvature data to all cells in the grid (set the Constant or Surface to Same for all Zones and Same for All Segments), although it is also possible to write the data only to specific stratigraphic zones. Click on the check box in the “All Segments, All Zones” cell of the table to activate the blue arrow button, then select the Min Curvature attribute in the Input pane and click on the blue arrow button to drop this into the table (Figure 12). Set the Property Template to Strain, and then click on “Apply”. This should create a new grid property on the Models pane labelled “Min curvature”. Repeat this procedure to create grid properties for Max curvature and for Azimuth of Max Curvature (use the Dip Azimuth Property Template for the Azimuth of Max Curvature).
5. The maximum curvature is used as a proxy for the maximum extensional strain. It must therefore be inverted, to convert positive curvature values into negative strain values. It is usually also necessary to apply a multiplier to give geologically realistic strain rates; these are typically -0.001 to -0.1/ma, requiring a multiplier of c.10-100. This can be done using the Property Calculator. Right click on the Max Curvature property in the Models pane and select “Calculate”. Create a new property “EhminRate” from the Max Curvature property by typing “EhminRate=-100*” in the Calculator command line, selecting the Max Curvature property in the Models pane, clicking on

the blue arrow button in the calculator to drop it into the command line, selecting Strain from the Property Template drop-down menu, and pressing Return to run the calculation (Figure 13). Repeat the procedure to create a new “EhmaxRate” property from the Min Curvature property. Note that the maximum curvature is used to calculate the minimum horizontal strain rate and vice versa; this is because the minimum horizontal strain represents the most negative (i.e. most extensional) horizontal strain.

6. The EhminRate, EhmaxRate and Azimuth of Max Curvature properties can now be used to populate the Minimum horizontal strain rate, Maximum horizontal strain rate and Minimum horizontal strain azimuth settings on the DFN Generator Main settings tab.

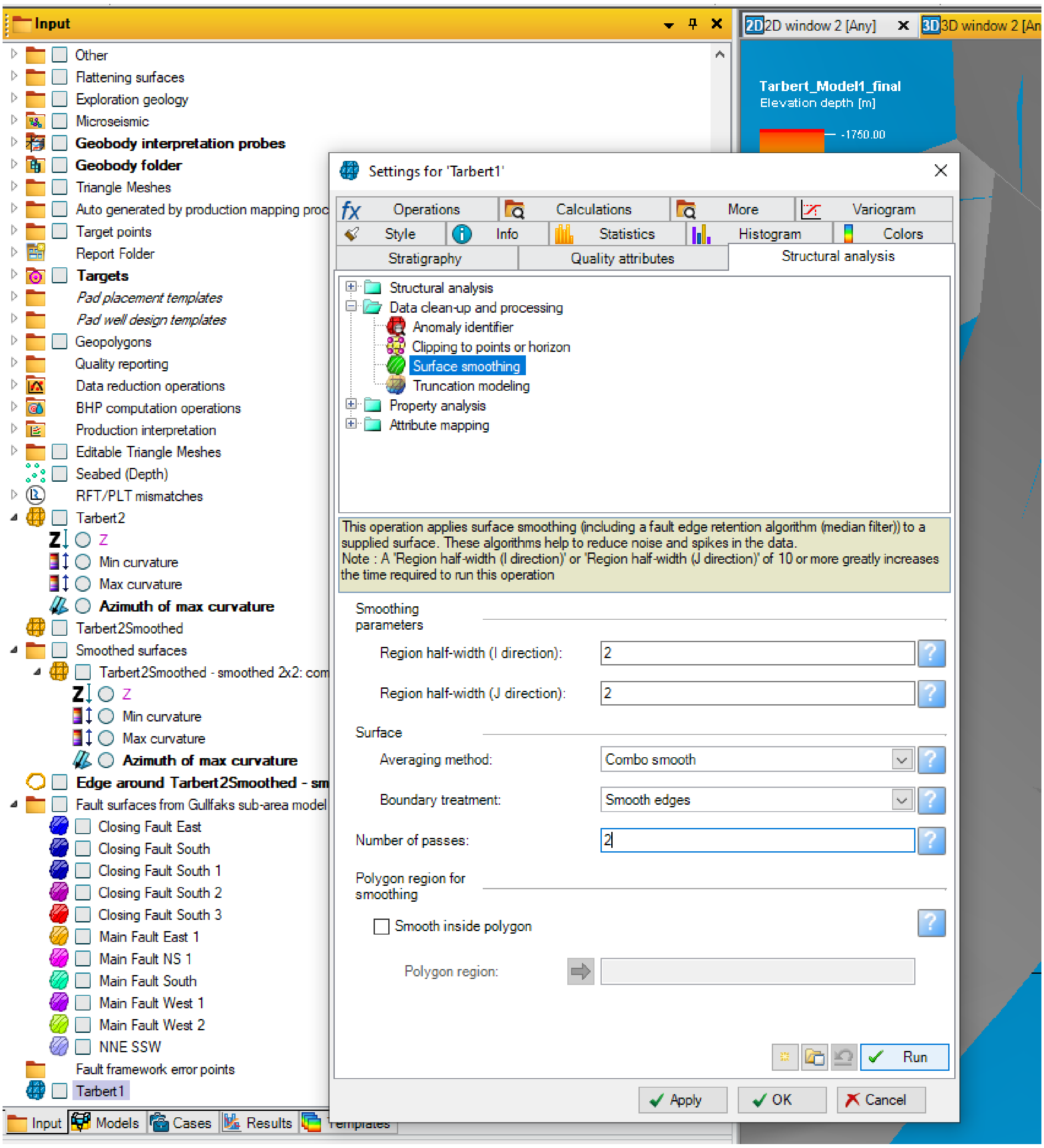

Figure 10: Using the Surface Smoothing algorithm to smooth a surface.

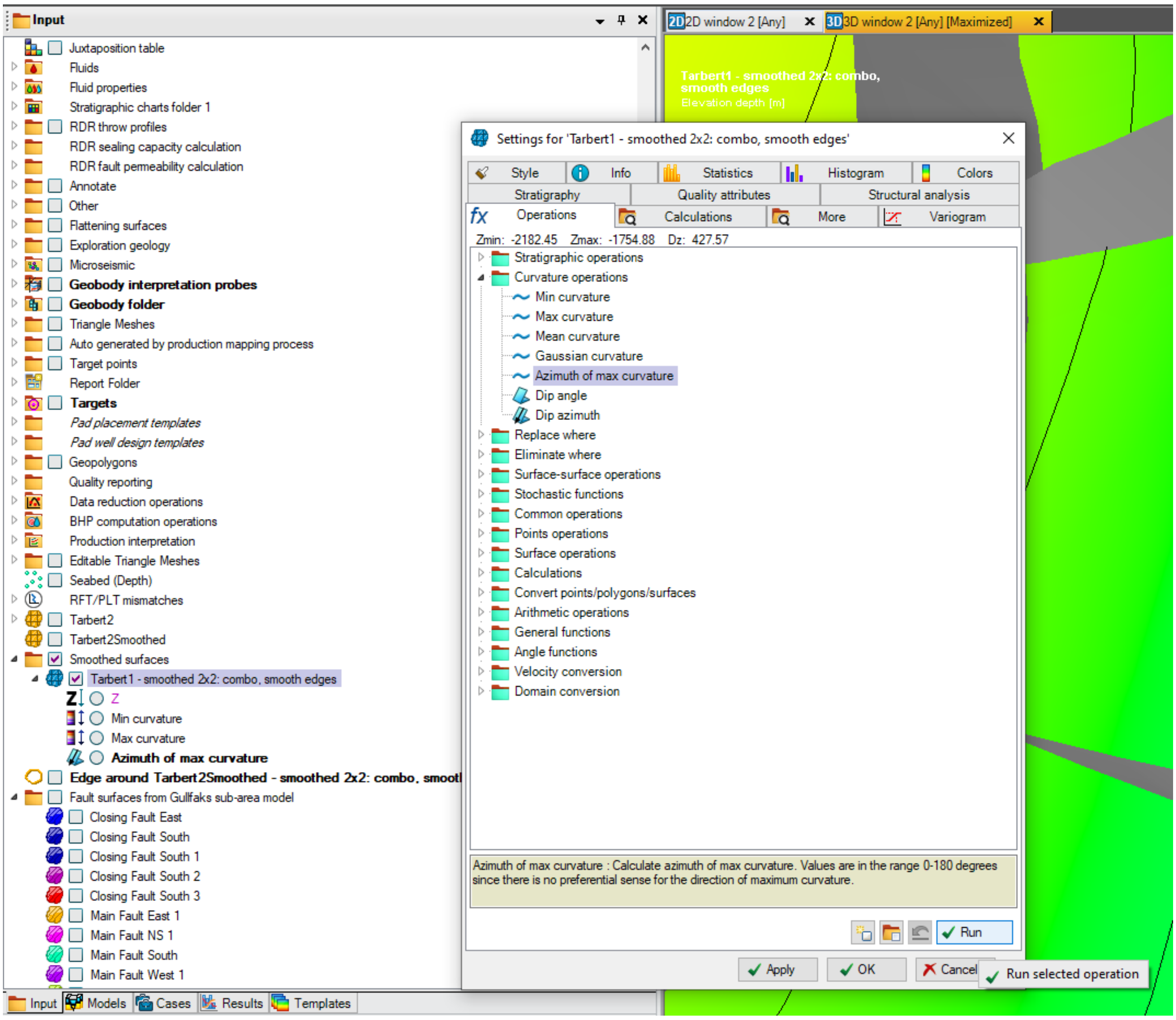

Figure 11: Calculating curvature as a surface attribute using the Curvature Operations algorithms.

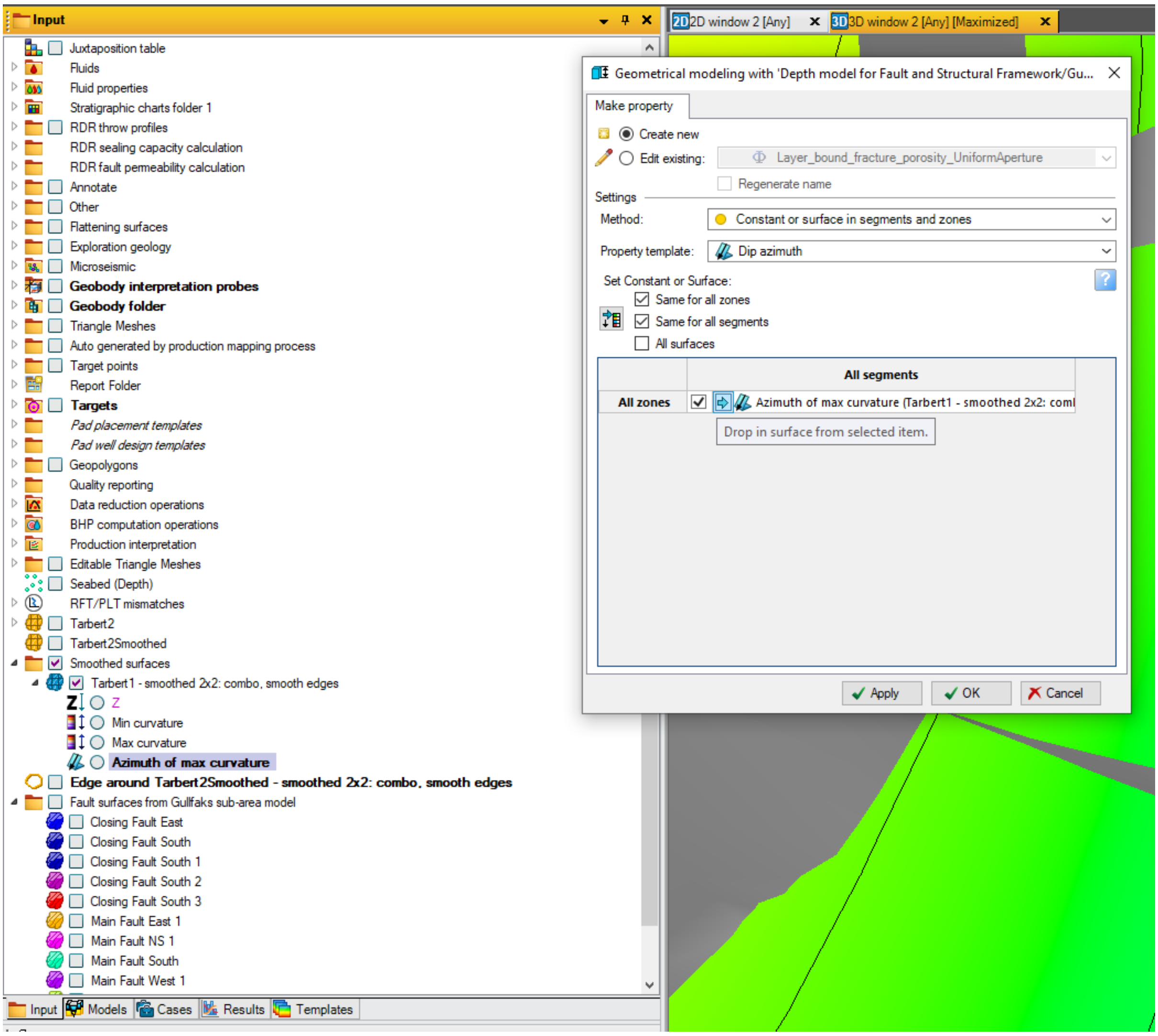

Figure 12: Converting the curvature surface attributes to grid properties using the Geometric Modelling module.

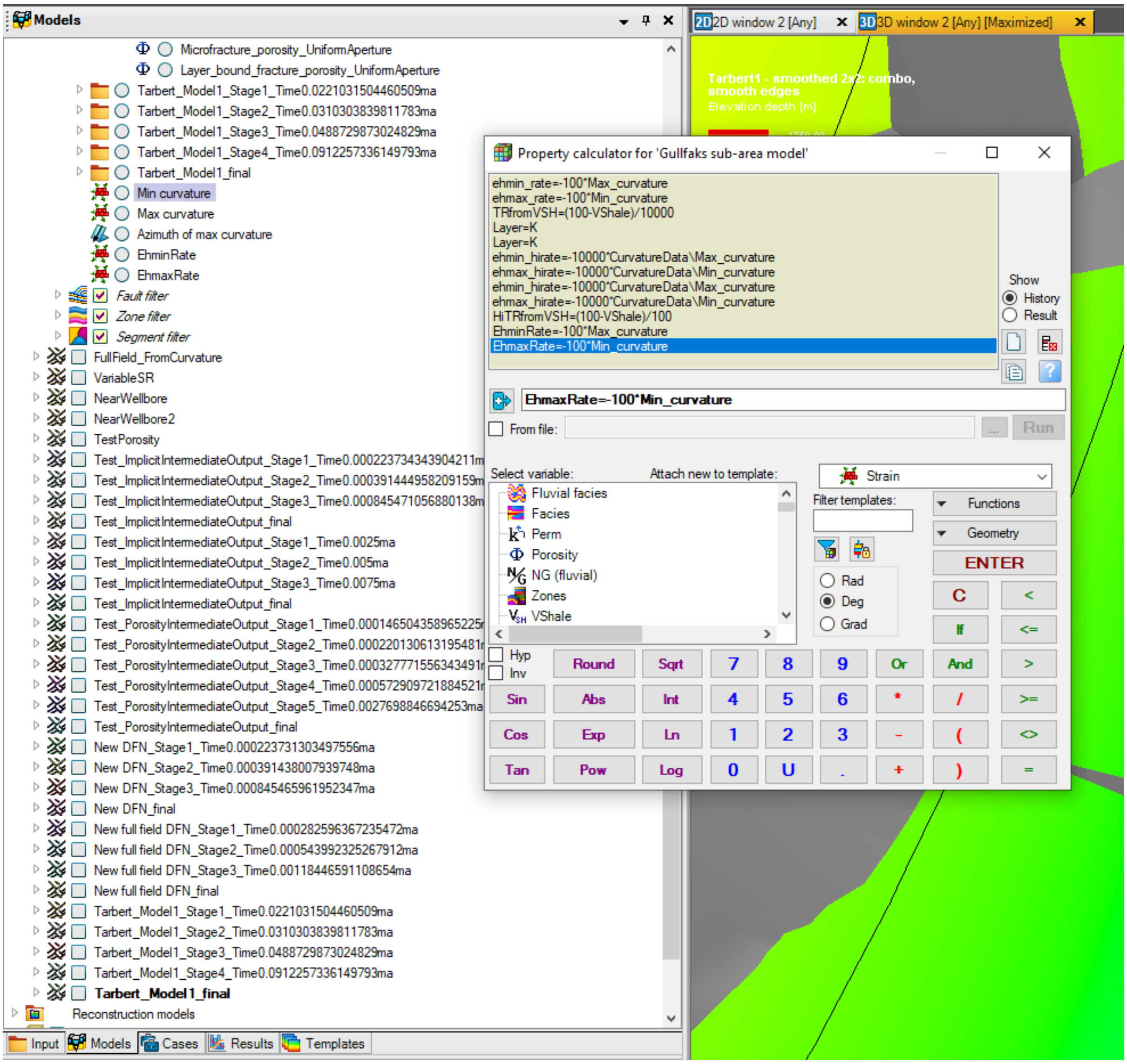

Figure 13: Using the Property Calculator to invert the curvature data to generate strain rate properties.

Output Data

By default, the DFN Generator generates both implicit fracture data (e.g. fracture density and porosity values, as grid properties) and explicit DFNs (geometric representations of the fracture network, as Petrel DFN objects). A model run with default settings will comprise four fracture sets: a vertical Mode 1 set and an inclined Mode 2 set, striking perpendicular to the minimum and maximum horizontal strain directions (although there may be no fractures in some of the sets). Implicit fracture data will be generated for each set, as well as for the combined fracture network.

The implicit property data will be placed in a folder or folders underneath the Grid object in the Models pane. If intermediate stages of the model were output, one folder will be generated for each intermediate stage as well as a folder for the final fracture network. All folders will be clearly labelled with the model name and stage, and the parameters used to generate the model will be listed on the Comments tab of the folder Settings dialog.

Each folder will contain multiple subfolders: one subfolder containing data for each of the fracture sets, and two subfolders containing data representing the entire network, as shown on Figure 14. The data contained in these folders is as follows:

- **In the fracture set folders:**
    - Total density of layer-bound fractures in the set, expressed as the number of fractures per unit volume (P30).
    - Total density of layer-bound fractures in the set, expressed as the total fracture area per unit volume (P32).
    - Total density of microfractures in the set, expressed as the total fracture area per unit volume (P32).
    - Mean length of layer-bound fractures, given by MFP32/(layer thickness * MFP30).
    - Proportion of layer-bound fracture tips from this fracture set that are not connected to other fractures (I-nodes).
    - Proportion of layer-bound fracture tips from this fracture set that are linked to other fractures by relay zones (R-nodes).
    - Proportion of layer-bound fracture tips from this fracture set that terminate against other fractures (Y-nodes).
    - The time taken for the fracture set to reach saturation and stop growing.
- **In the Fracture Anisotropy folder:**
    - P32 anisotropy, defined as (P32_hmin – P32_hmax) / (P32_hmin + P32_hmax), where P32_hmin and P32_hmax are the total area of fractures perpendicular to the minimum and maximum horizontal strain directions respectively. P32 anisotropy is 0 for a fully isotropic fracture set (P32_hmin = P32_hmax) and 1 for a fully anisotropic fracture set (P32_hmax = 0).
    - Porosity anisotropy, defined as (ϕ_hmin – ϕ_hmax) / (ϕ_hmin + ϕ_hmax), where ϕ_hmin and ϕ_hmax are the total porosity of fractures perpendicular to the minimum and maximum horizontal strain directions respectively. Porosity anisotropy is 0 for a fully isotropic fracture set ((ϕ_hmin = ϕ_hmax) and 1 for a fully anisotropic fracture set (ϕ_hmax = 0).
    - Proportion of layer-bound fracture tips from the entire fracture network that are not connected to other fractures (I-nodes).
    - Proportion of layer-bound fracture tips from the entire fracture network that are linked to other fractures by relay zones (R-nodes).

- Proportion of layer-bound fracture tips from the entire fracture network that terminate against other fractures (Y-nodes).
- The time taken for the entire fracture network to reach saturation and stop growing.

- **In the Fracture Porosity folder:**
    - Total density of all layer-bound fractures in the entire fracture network, expressed as the total fracture area per unit volume (P32).
    - Total density of all microfractures in the entire fracture network, expressed as the total fracture area per unit volume (P32).
    - Total porosity of all layer-bound fractures in the entire fracture network; porosity is calculated as defined on the Output tab of the DFN Generator interface.
    - Total porosity of all microfractures in the entire fracture network; porosity is calculated as defined on the Output tab of the DFN Generator interface.

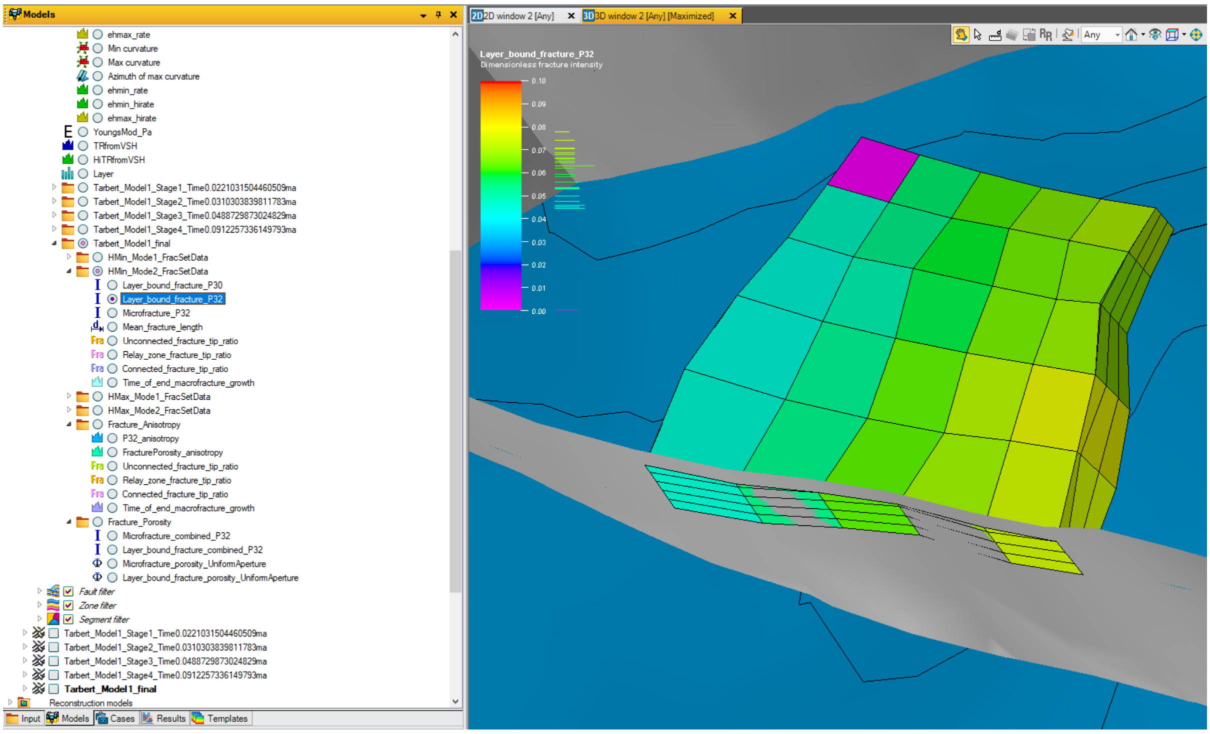

Figure 14: Implicit fracture property output data. In this example, the total area (P32) of Mode 2 layer-bound fractures striking perpendicular to the direction of maximum extension (i.e. HMin) is displayed.

A Petrel DFN object will also be generated for each intermediate stage, and for the final fracture network. These will likewise be clearly labelled with the model name and stage, and the parameters used to generate the model will be listed on the Comments tab of the Settings dialog for the DFN objects. They each comprise a collection of planar geometric objects representing individual fracture segments (Figure 15). Note that long fractures may comprise multiple segments, allowing them to bend between cell stacks, following lateral variations in the strain orientation. Each segment is assigned an aperture, calculated as defined on the Output tab of the DFN Generator interface.

It is possible to follow the growth of individual fractures by comparing the intermediate stage DFNs (Figure 16). This also highlights various events that occur during the growth of the fracture network as a whole, such as linking of initially independent fracture to form relay zones, and the late growth of secondary fractures connecting the long primary fractures.

Microfractures are not represented in the explicit DFN unless specified on the Control Parameters tab of the DFN Generator interface. It is recommended that this is only done for small models (e.g. near-wellbore models), to avoid excessive runtimes.

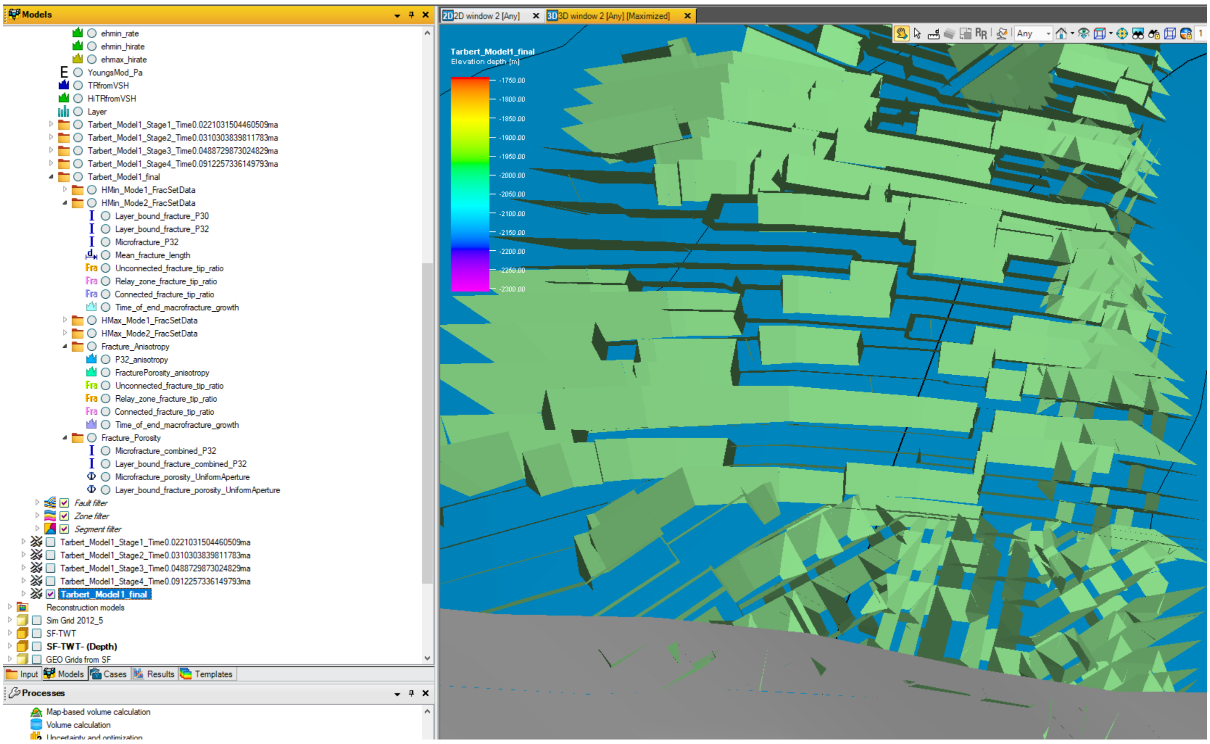

Figure 15: Explicit DFN representing the final fracture network as a collection of planar objects. Individual fractures may comprise multiple planar segments, allowing them to bend between cell stacks.

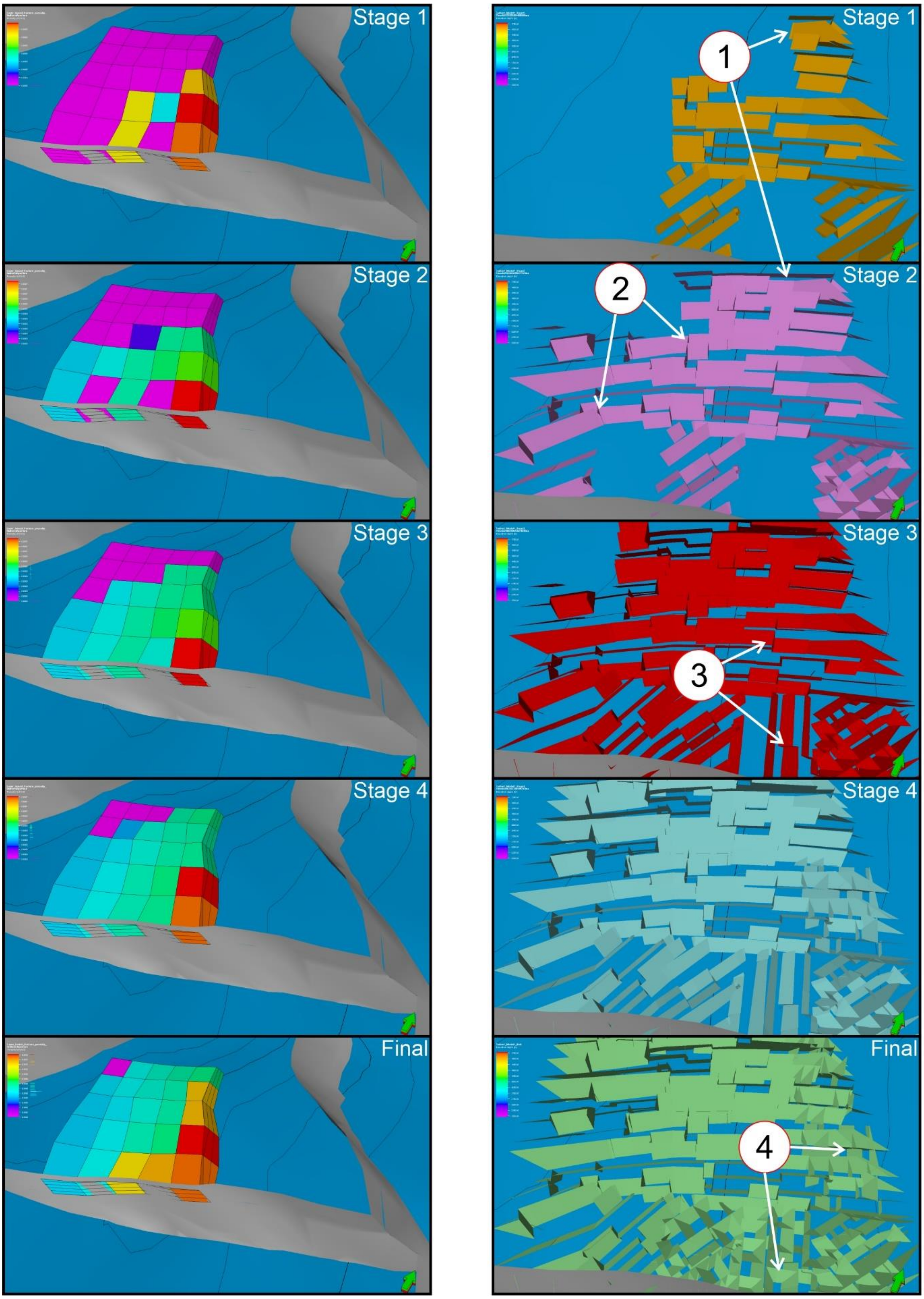

Figure 16: Implicit and explicit output representing the intermediate and final stages of fracture growth. On the explicit DFNs, note: 1) the growth of individual fractures between intermediate stages, 2) the curved profile of the long primary fractures across multiple cell stacks, 3) the relay zones along these primary fractures, where initially independent fractures have linked up due to stress shadow interaction, and 4) the short secondary fractures connecting the primary fractures.

Using the DFN Generator with the Petrel Workflow Editor

The Petrel Workflow Editor provides a means to automate elements of model running. This is particularly useful when testing different parameters for the DFN Generator, as we can use it to automate tests and label outputs appropriately to aid in administering our experiments.

The workflow editor in Petrel is available within the 'Workflows' tab, normally found below the input panel. Right-click in the blank space below the workflow and select New workflow in the context menu (Figure 14).

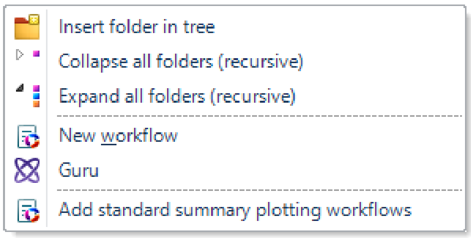

Figure 17: Workflows context menu showing the New workflow command.

Note, that if you right-click on a workflow you can also add new workflow variables, which can be renamed as you choose and subsequently called in the workflow itself. These can be assigned to Petrel objects such as grid properties or numerical values. Since the workflow editor does not allow property input boxes to be blank, we will create a null workflow variable for use where we will not assign a grid property to a parameter (for example if we want to use the default values for mechanical properties). Right-click on your new workflow and select New workflow variable from the context menu (Figure 18). Right-click the new variable that has been generated and name this variable 'Null' (or any other name you would like to use).

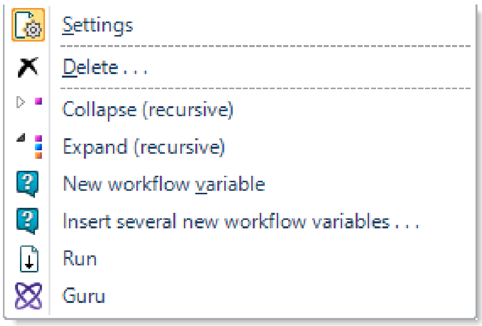

Figure 18: Workflow context menu showing the New workflow variable command.

Then double-click the title of your workflow to open the Workflow Editor window (Figure 19):

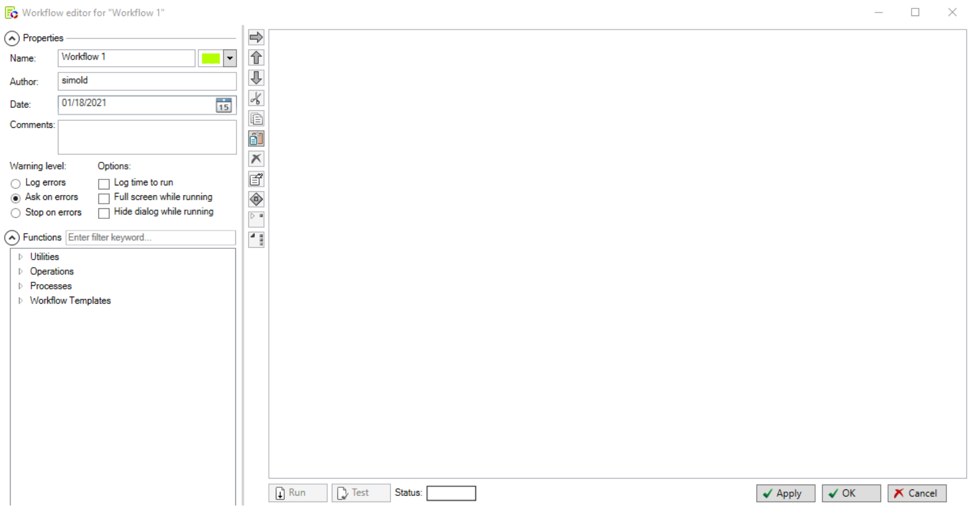

Figure 19: Workflow Editor window.

This window allows you to build the workflow. In the functions pane (shown to the left in Figure 16), you can access most functionality available within Petrel. Under Processes > Plug-ins > DFNGenerator, you add the DFN Generator to your workflow. Once added to the workflow, you can set all the DFN Generator parameters in the window on the right. These can be set to constant values or to variables.

In the example shown in Figure 20, we use the workflow editor to set the value of selected parameters. Using the 'Numerical expression' found under the 'Utilities' part of the workflow editors 'Functions' menu (bottom-left in the screen shot), we define variable names, beginning with dollar-signs, for several different numerical values (lines 1-6). In line 7, we assign a grid property (Azimuth of maximum curvature) to a workflow variable, and in line 8, we create a string expression for the model name, based on the variables we have previously defined. Finally in line 9, we then insert the variable names into the appropriate DFN Generator input parameters (Figure 18).

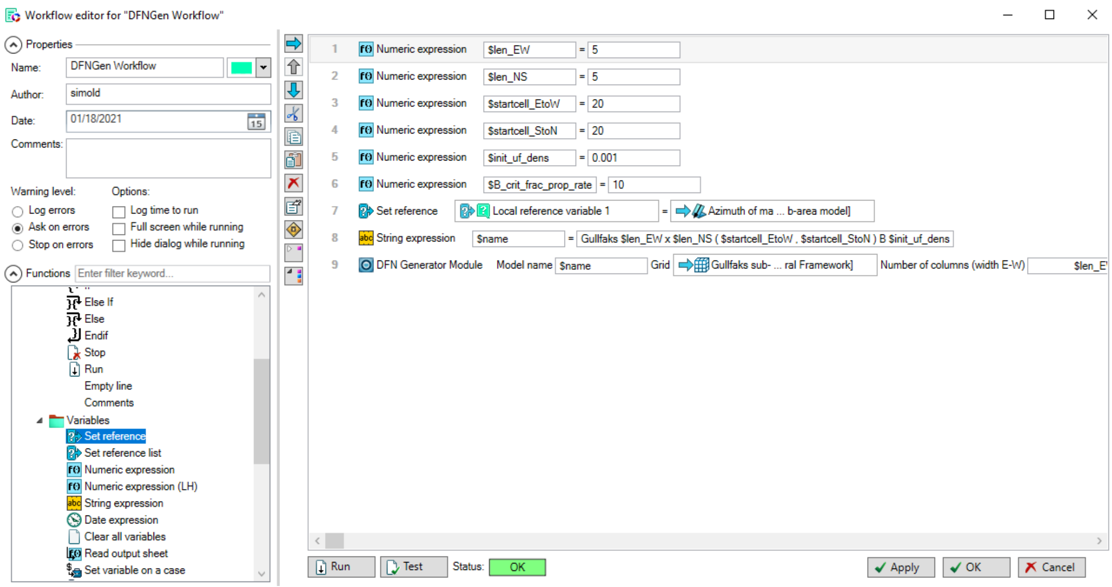

Figure 20: Creating a workflow in the Workflow Editor window.

The use of variables to set the DFN Generator input parameters is shown in more detail in Figure 21. In this example, the input parameters are set to the same values as illustrated previously in this user guide. Note that where numeric expressions have been applied the variable name has been inserted into the appropriate input box of line 9. The ‘Set reference’ declaration has been used to insert a grid property to override the default azimuth, using the workflow variable ‘Local reference variable 1’, defined in the workflow pane in the same way as the ‘Null’ variable defined earlier.

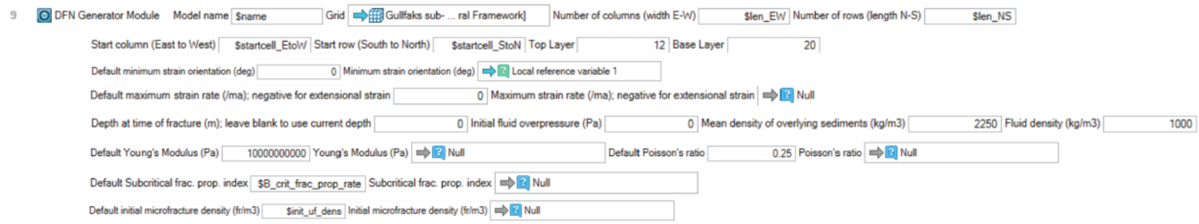

Figure 21: Using variables to set input parameters for DFN Generator. Note that in the Workflow Editor window, this would all be shown on one line and it would be necessary to scroll right to see all the input boxes.

Note that we have not assigned a property to the ‘Null’ workflow variable, leaving its value as null or undefined. We have inserted it into each location that a grid property could have been dropped into the workflow, so that the tool will ignore the undefined value and instead use the default values. This prevents the workflow editor from declaring an error due to missing parameters.

The workflow editor also allows for the use of loops and logic statements. Their use offers a route to automatic automatic generation of multiple models, testing different paremters or scenarios.. Once ran, the output model will be named as defined in line 8. This ensures that each model run is appropriately named and easily identified.

To better understand these elements of the workflow editor, we recommend consulting the Petrel help pages on general usage of the workflow editor.

Further information and contact details

More details of the algorithm used in the DFN Generator module, as well as analysis of the key controls on the development of fracture networks, can be found in

> Welch, M. J., Lüthje, M., & Glad, A. C. 2019. Influence of fracture nucleation and propagation rates on fracture geometry: insights from geomechanical modelling. Petroleum Geoscience, 25(4), 470-489.

and in more detail, in the book

> M. Welch, M. Lüthje and S. Oldfield. Modelling the Evolution of Natural Fracture Networks - Methods for Simulating the Nucleation, Propagation and Interaction of Layer-Bound Fractures. Springer. 2020

The latter also contains examples of the application of this plug-in to outcrop and subsurface examples of fractured horizons. It can be ordered direct from the publisher, at https://www.springer.com/gp/book/9783030524135.

DFN Generator has been developed with funding from the Danish Hydrocarbon Research and Technology Centre (DHRTC) under the Advanced Water Flooding programme. The software is still under development and is not currently available as a commercial release. Please note therefore that DFN Generator comes with no warranty and DHRTC and the authors accept no liability for any consequence arising from its use.

There is also no formal support or service level agreement for the software. However if you encounter any problems, or have any comments or suggestions, please contact Michael Welch (mwelch@dtu.dk) or Mikael Lüthje (mikael@dtu.dk) and we will try to help you. Please also report any bugs that you encounter or requests for functionality enhancements in the same way.