JAEA-Testing
2008-007

JAEA-Testing

GEOMASS System

Takuya OHYAMA and Hiromitsu SAEGUSA

Crystalline Environment Research Group
Geological Isolation Research and Development Directorate

March 2009

Japan Atomic Energy Agency

日本原子力研究開発機構

本レポートは独立行政法人日本原子力研究開発機構が不定期に発行する成果報告書です。
本レポートの入手並びに著作権利用に関するお問い合わせは、下記あてにお問い合わせ下さい。
なお、本レポートの全文は日本原子力研究開発機構ホームページ（http://www.jaea.go.jp）
より発信されています。

独立行政法人日本原子力研究開発機構　研究技術情報部　研究技術情報課
〒319-1195　茨城県那珂郡東海村白方白根 2 番地 4
電話 029-282-6387, Fax 029-282-5920, E-mail:ird-support@jaea.go.jp

This report is issued irregularly by Japan Atomic Energy Agency
Inquiries about availability and/or copyright of this report should be addressed to
Intellectual Resources Section, Intellectual Resources Department,
Japan Atomic Energy Agency
2-4 Shirakata Shirane, Tokai-mura, Naka-gun, Ibaraki-ken 319-1195 Japan
Tel +81-29-282-6387, Fax +81-29-282-5920, E-mail:ird-support@jaea.go.jp

JAEA-Testing 2008-007

GEOMASS System

Takuya OHYAMA and Hiromitsu SAEGUSA

Tono Geoscientific Research Unit
Geological Isolation Research and Development Directorate
Japan Atomic Energy Agency
Akeyo-cho, Mizunami-shi, Gifu-ken

(Received December 24, 2008)

As a part of the research and development regarding characterisation of deep geological environment, the GEOMASS (GEOLOGICAL MODELLING ANALYSIS AND SIMULATION SOFTWARE) system has been developed by the Japan Atomic Energy Agency in order to carry out geological and hydrogeological modelling and groundwater flow simulation and so on. The GEOMASS system integrates a commercial geological interpretation system (EarthVision®), which is used for geological modelling and visualisation, with a proprietary code for groundwater flow (FracAffinity). This integrated system allows users to make rapid improvement of models as data increases. Also, it is possible to perform more realistic groundwater flow simulation due to the capability of modelling the rock mass as a continuum with discrete hydro-structural features in the rock mass.
This paper consists of "Overview of GEOMASS system", "FracAffinity Theoretical Background" and "FracAffinity User Guide" and is edited as a GEOMASS system manual. "Overview of GEOMASS system" describes the outline of this system. "FracAffinity Theoretical Background" describes the information of technical background of FracAffinity software. "FracAffinity User Guide" describes the structure of the FracAffinity input files, the usage of FracAffinity Interface and flow-solver.
Updating of the FracAffinity has been continued as needed and FracAffinity version3.3 is the latest version at present (July 2008).

Keywords: Geological Modelling, Visualisation, Groundwater Flow Simulation, Integrated Simulation System

JAEA-Testing 2008-007

GEOMASSシステム

日本原子力研究開発機構
地層処分研究開発部門　東濃地科学研究ユニット
大山　卓也，三枝　博光

（2008年12月24日受理）

日本原子力研究開発機構（以下、原子力機構）では、複雑な水理地質環境における地下水流動を効率的にモデル化・解析するため，これらを一貫して実施できる GEOMASS (GEOLOGICAL MODELLING ANALYSIS AND SIMULATION SOFTWARE) システムを開発してきた。本システムは、地質構造モデルの構築とモデル化・解析結果の可視化を行う市販ソフトウェアである EarthVision®と、原子力機構が所有権を有する地下水流動解析コードである FracAffinity が統合された環境となっており、情報量の増加などによるモデルの更新に迅速に対応することを可能としている。また、岩盤の連続構造と断層や割れ目などの不連続構造を同時に扱える手法を取り入れることにより、より現実的な地下水流動の表現を可能としている。

本報告書は，GEOMASS システムのマニュアルとして整備したものであり，“Overview of GEOMASS system”、“FracAffinity Theoretical Background”および“FracAffinity User Guide”から構成されている。“Overview of GEOMASS system”では GEOMASS システムの概要、“FracAffinity Theoretical Background”では FracAffinity が有する機能の理論的背景、“FracAffinity User Guide”では FracAffinity の入力ファイルの構成や解析を実行する際のインターフェイスの使用方法などについて述べるものである。

なお、FracAffinity は継続的に更新されており、本報告書では 2008 年 7 月時点での最新版である Version3.3 について述べる。

東濃地科学センター（駐在）：〒502-6132　岐阜県瑞浪市明世町山野内1-64

Contents

目次

Figures

Tables

1. Introduction

1.1 Background

A principal mechanism for the transport of radionuclides from a repository to the surface is the movement of groundwater. Understanding groundwater flow through the sub-surface geological environment is thus a key requirement of site characterisation activities associated with the disposal of radioactive waste. Developing this understanding requires characterisation of the geological structure, and the acquisition and modelling of hydrogeological information. However, these aspects of site characterisation are often treated separately during site investigations, and a need to make site characterisation data more accessible for hydrogeological modelling has long been recognised.

In fractured rocks, the fractures provide the principal pathways for groundwater flow, and modelling is either undertaken using equivalent porous medium or discrete fracture network approaches. However, both of these approaches and the methods used to integrate results from them have well-known shortcomings. First, in a pure porous medium approach there are difficulties in representing the fast flow channels or preferential pathways due to fractures and faults. Second, pure fracture network models fail to account for any flow in permeable parts of the intact rock in which the fractures are located, which means that the simulated flow is critically dependent on the connectivity of the fracture network [1),2),3)]

In order to overcome both the problems of the lack of integration of geological and hydrogeological investigations and the limitations of standard porous medium or fracture network modelling, a new software system, GEOMASS (Geological Modelling, Analysis and Simulation Software), has been developed by the Japan Atomic Energy Agency (JAEA).

GEOMASS has been developed and applied in the Mizunami Underground Research Laboratory (MIU) Project, in which a purpose-built underground research laboratory (URL) is being constructed in the Cretaceous Toki Granite in the Tono area, central Japan. However, GEOMASS provides an approach to integrated geological and hydrogeological modelling and visualisation that is suitable for site investigations in all types of host rock, and for projects unrelated to radioactive waste disposal.

This document provides an overview of the approach and key features of GEOMASS for users and potential users of the system. Detailed information on the operation and theory of the GEOMASS system is explained in the EarthVision User's Guide [4)], the FracAffinity Theoretical Background (Chapter 7) and the FracAffinity User Guide (Chapter 8).

1.2 Document Scope and Structure

As noted above, the purpose of this document is to provide an overview, theoretical background and user guide of the GEOMASS system for users and potential users of the system. GEOMASS incorporates many standard geological and hydrogeological interpretation and modelling techniques, and description of these techniques are outside the scope of this document. Rather, it is the intention to describe the novel approaches that have been implemented within GEOMASS. These include:

- The coupling of geological and hydrogeological modelling and visualisation in a fully integrated fashion, providing a sophisticated platform for investigating the impact of geological structure on groundwater flow.
- The hybrid medium approach to groundwater flow modelling adopted in FracAffinity.
- Routines in FracAffinity for modelling heterogeneity and time dependence (e.g. dynamic shaft excavation modelling).

This manual of the GEOMASS system is presented in nine chapters:

- Chapter 1 provides an introduction to the GEOMASS system.
- Chapter 2 describes how the different elements of GEOMASS are linked together into an integrated system.
- Chapter 3 describes the approach to geological modelling using EarthVision. Note that this chapter is rather brief, since EarthVision uses standard geological interpretation methods, and it is the intention of this chapter to describe the novel aspects of the GEOMASS.
- Chapter 4 describes the approach to hydrogeological modelling using FracAffinity. This chapter describes many of the novel hydrogeological modelling approaches that have been implemented in FracAffinity.
- Chapter 5 outlines the visualisation, using EarthVision, of models and information generated with FracAffinity, including an example from the MIU Project.
- Chapter 6 provides an overview of how GEOMASS is applied.
- Chapter 7 provides the Theoretical background of FracAffinity 3.3.
- Chapter 8 provides the User Guide of FracAffinity 3.3.
- Chapter 9 describes the Summary.

A Glossary is provided, which defines terms as used in the GEOMASS system.

2. The GEOMASS System

The GEOMASS system is an integrated suite of Three-D software applications for geological modelling, hydrogeological modelling and visualisation. EarthVision, a commercial geological modelling system, provides the capability to construct Three-D geological models. Groundwater flow is modelled using the proprietary code FracAffinity, which uses a hybrid medium concept to represent the flow domain and is based on a finite difference method. Visualisation of the output from FracAffinity against the geological models is provided by EarthVision.

2.1 Geological Modelling Using EarthVision

Geological modelling systems are families of software programmes which allow the construction of integrated Three-D models of site investigation data, including sub-surface geology, surface geography and engineering features. These systems concentrate on the development of static models which aim to produce a descriptive representation of the surface and sub-surface, thereby providing an increased understanding of the area of interest. Three-D geological modelling in GEOMASS is undertaken using EarthVision.

Three-D geological models are commonly built by combining a structural framework with information on rock and fluid properties (e.g. porosity, hydraulic conductivity, fluid density or hydrochemistry). The structural framework consists of Two-D surfaces which represent stratigraphic boundaries or discontinuities (e.g. faults, fractures and dykes) (Fig. 2.1). In EarthVision, all surfaces are represented by a regular xy grid in which the elevation z is defined as a variable at each xy point.

Spatial data, such as groundwater chemistry and formation properties can be visualised and interpolated within EarthVision (Fig. 2.2). The system can represent existing boreholes and can be used to plan and predict future drilling and excavation activities.

A strength of the GEOMASS system is the ability to pass the geological framework automatically from EarthVision to FracAffinity. This reduces the time for generating the flow grid to a matter of hours, allowing concentration on the practical investigation of the impact of geological uncertainty on the flow model. The ability to simulate several geological interpretations provides valuable information for decision making in performance assessment and site investigation work. The transfer of information uses an ASCII format so that FracAffinity may be run separately from EarthVision.

This is a blank page.

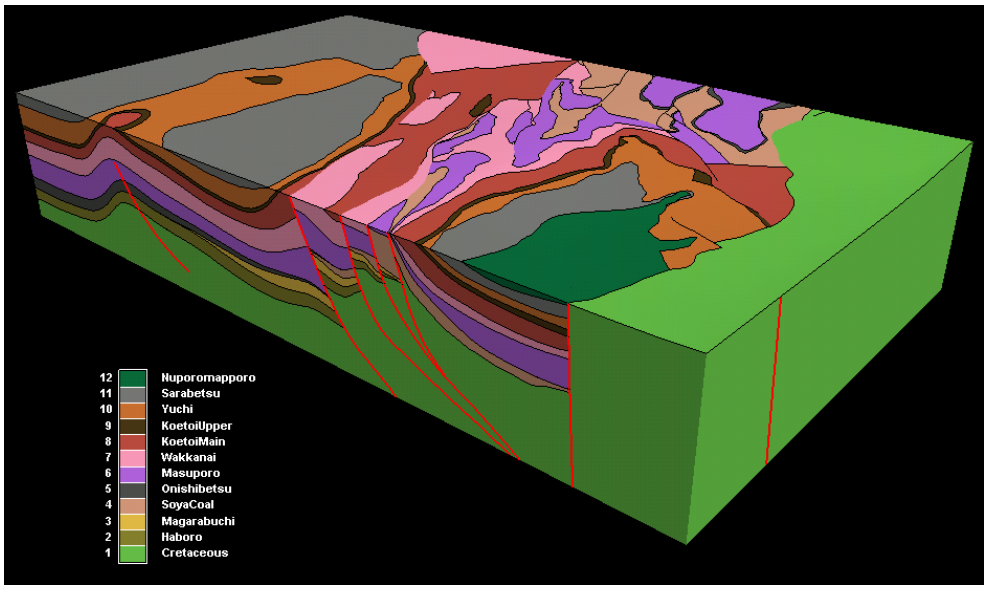


Fig. 2.1 Geological model of the Horonobe area, Hokkaido, northern Japan, illustrating the geological structure of the region.

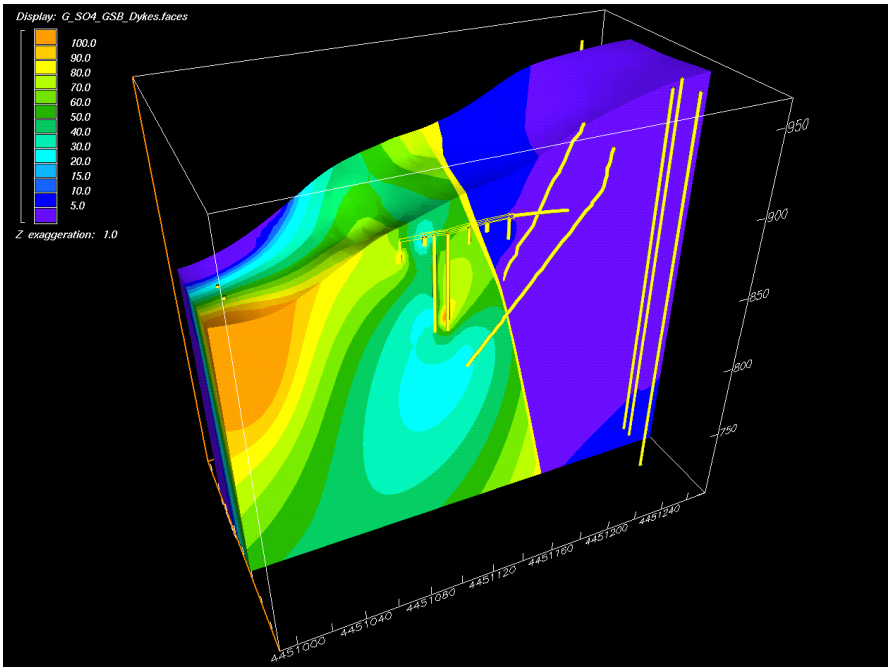


Fig. 2.2 Three-D model of the concentration of sulphate in groundwater at the El Berrocal site in Spain, the location of a uranium vein. The model is based on borehole measurements that have been interpolated in Three-D space with respect to the geological structure. The borehole data have been interpolated separately north of the uranium vein and to the south of it. The Three-D model demonstrates that the highest levels of sulphate occur within the mineralised vein and support a hypothesis that the source of the sulphate in the groundwater at El Berrocal is pyrite, mineralised within the vein [5)].

This is a blank page.

2.2 Groundwater Flow Modelling using FracAffinity

FracAffinity is a Three-D transient flow simulator based on the concept of a hybrid medium in which the rock contains both a permeable matrix and fractures. The hybrid model is designed to overcome the shortcomings of separate porous medium and discrete feature approaches, as outlined in Chapter 1. Groundwater flow is modelled for steady-state and transient cases with either fully saturated or partially saturated (unsaturated) conditions (see Section 4.1 for more details).

The basic idea behind the hybrid approach is to represent a volume of fractured rock as two main components: *discrete features* and *intact rock* (this is shown schematically in Fig. 2.3, and using an example from the Tono area in Fig. 2.4). Discrete features are linear objects such as faults, fractures, unconformities or dykes that introduce linear/surface variations in the properties of the rock. A distinction is made between *deterministic discrete features* (DDFs), whose geometry might be determined accurately by a regional geological investigation (e.g. large-scale faults), and *stochastic discrete features* (SDFs), which are smaller-scale features about which only partial information is available (e.g. fractures or sub-seismic faults). The *intact rock* is the remaining rock which is either completely intact or contains only "micro-fractures".

Properties of the intact rock are most naturally thought of as being associated with a volume (a "cell"). The approach adopted in FracAffinity is to populate the cells in a grid with flow properties (conductivity, porosity and specific storage coefficient) which may be different in different stratigraphic zones. The rock properties may be uniform or spatially variable in different stratigraphic zones and may also be conditioned on borehole data. The development of the hybrid medium is described in more detail in Section 4.2.

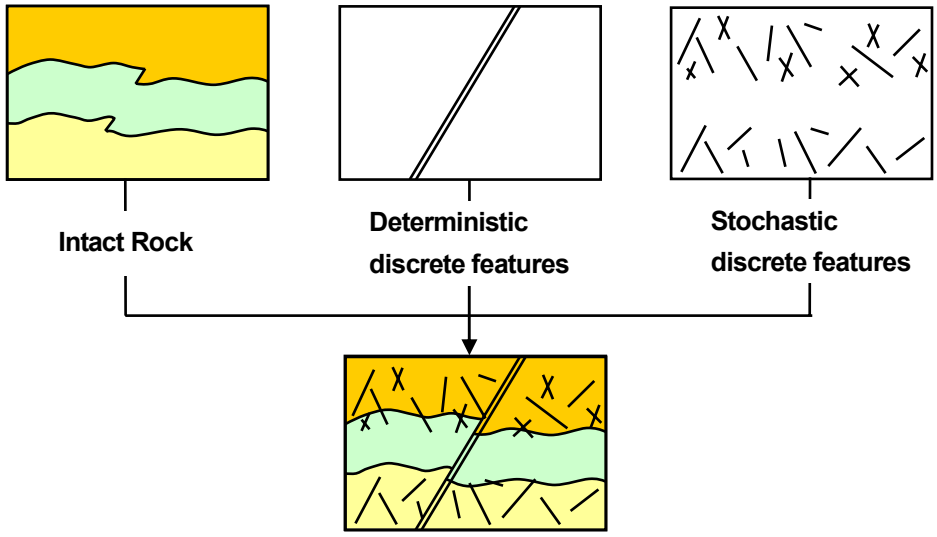


Fig. 2.3 Schematic diagram of the constituent components of the FracAffinity hybrid medium: intact rock, deterministic discrete features and stochastic discrete features.

This is a blank page.

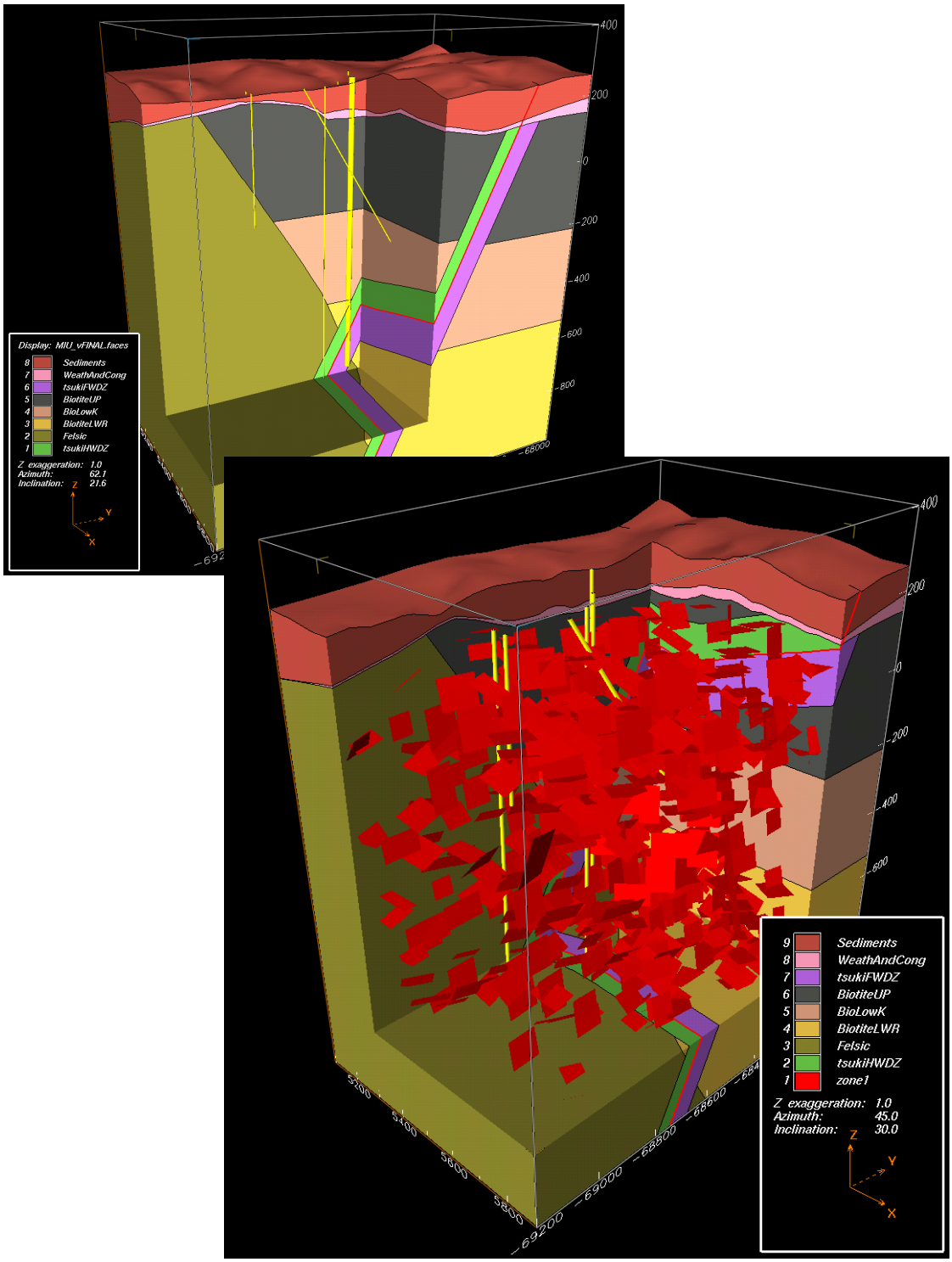


Fig. 2.4 Illustration of a hybrid medium model based on data from the Tono area. The upper figure illustrates the geological structure developed in EarthVision, with the coloured zones representing intact rock and the thin red line representing a DDF (Deterministic Discrete Feature), in this case the Tsukiyoshi Fault. The lower figure shows a set of SDFs (Stochastic Discrete Features) generated in FracAffinity, which represent fractures within the Toki Granite.

This is a blank page.

A particular feature of the hybrid medium is the ability to represent heterogeneity on all scales. At the large scale, heterogeneity is represented by the geological framework provided by EarthVision. This allows conceptual models (e.g. relative importance of fracture and matrix flow) and hydrogeological parameters to be specified on a zone-by-zone basis and complete flexibility in the discretisation of the geological sequence. At the medium scale, hydraulic anisotropy may be represented by a fractal model with anisotropic correlation functions. At the small scale heterogeneity is modelled by consideration of aspects of the geological structure that may cause anisotropy in the hydraulic properties of the rock mass and local conductivity tensors may be applied in terms of anisotropy factors in the three principal directions. In addition, the anisotropy of discrete features may be represented by differences in the in-plane and cross-plane hydraulic conductivities and by modified conductivities along feature intersections. The representation of heterogeneity is described in more detail in Section 4.3.

In addition to representing the structure of the rock mass, FracAffinity includes methods for modelling engineering features, i.e. dynamically modified shafts (referred to as excavations in GEOMASS), static vertical shafts or horizontal galleries, and boreholes. All of these features are represented as cylinders. Techniques have been developed to link these features into the irregular hybrid medium network. These techniques aim to realistically simulate pressure gradients surrounding the engineered feature, particularly in the presence of local desaturation. The methods of incorporating shafts, galleries and boreholes are described in more detail in Section 4.4.

In order to integrate all of these natural and man-made features into a single flow network, specialised routines have been developed in FracAffinity for modifying the basic hybrid medium grid, so that the user can focus grid generation on the elements of interest, e.g. desaturation around a shaft and movement of the water table. Grid refinement algorithms are described in more detail in Section 4.5.

This is a blank page.

2.3 Analysis and Visualisation of Flow Simulation Results

The results from FracAffinity are output to standard text files or to files with EarthVision-compatible formats for rapid visualisation and analysis alongside the Three-D geological interpretation. This provides a platform for verifying data input, analysing and understanding the groundwater flow simulations, and presenting the results to stakeholders. Features which can be visualised in Three-D include:

- Networks of stochastic features.
- Heterogeneous property interpolations.
- The Three-D flow network.
- The Three-D head, pressure and saturation fields, which may be animated through time to understand the spatial and temporal evolution of transient simulations (e.g. Fig. 2.5).
- Pathlines.

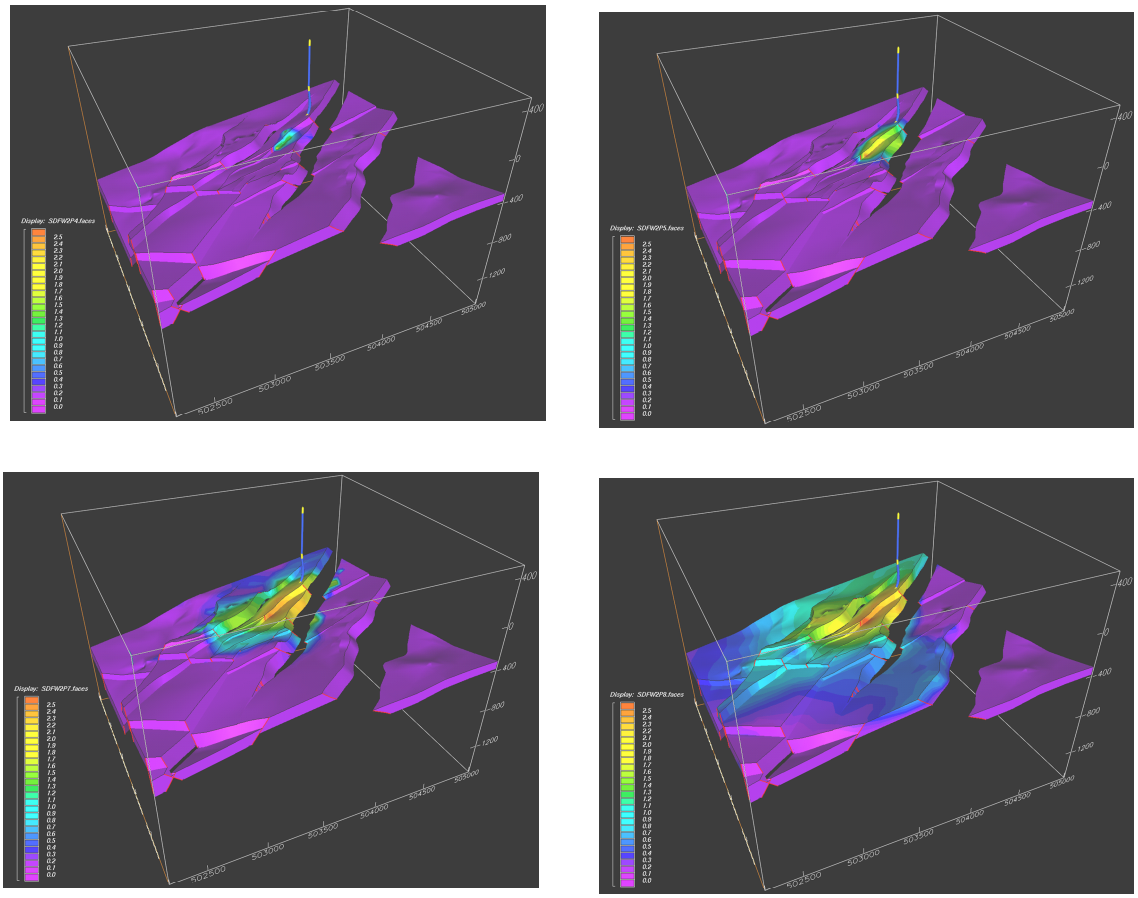


Fig. 2.5 Visualisation of the spatial and temporal evolution (Four-D) of the pressure response to a hypothetical model of a pumping test in a heterogeneous model. Colours indicate the drawdown in head (units are arbitrary). By visualising the results in Four-D both the temporal and spatial responses can be analysed and the region influenced by the pump test may be predicted taking into account the spatial variability of the rock mass in a manner not possible using simple analytical techniques such as application of the radial flow equation.

This is a blank page.

3. Geological Modelling with GEOMASS

3.1 Geological Modelling in Site Characterisation

The GEOMASS concept recognises the development and application of Three-D geological modelling as a central task within the site investigation workflow. Geological modelling offers a wide range of benefits and applications in projects associated with the disposal of radioactive waste. It provides a platform for geologists to optimise their role in disposal projects and can act as a central focus between site characterisation and performance assessment, with many benefits to both. Geological models can be used to monitor the progress made in disposal and R&D projects by recording and archiving each stage in the process. Geological models can be used to make quantitative predictions prior to data acquisition, and may then be used to evaluate the success of those predictions.

Incorporation of Three-D geological modelling into a site investigation should not be undertaken on a task-by-task basis. A central plan should be developed in which the data inputs into the Three-D geological models are identified and related to data acquisition and interpretation methods being applied in the field, and an application strategy identified in which the key applications of the Three-D geological models are listed so that the models can be constructed with the appropriate detail and accuracy, and so that the outputs of the Three-D geological models can be determined accordingly.

The construction of the Three-D geological model is part of an iterative cycle of structural interpretation, which consists of the following steps:

- ***Site characterisation and data acquisition***, including field mapping, geophysical data acquisition and borehole drilling. A clear plan and objectives for the site characterisation need to be established at an early stage, for example ensuring that if high-quality hydrochemical data are required, these data are collected prior to disruption of the geological environment.
- ***Data interpretation***, for example seismic interpretation and interpretation of borehole cores. This work needs to be undertaken with a good understanding of parallel interpretation work and the application of the results. For example, standard rules for identifying stratigraphic boundaries in cores and well logs need to be developed.
- ***Structural interpretation*** is generally undertaken with reference to Two-D and Three-D structural templates which define valid structural styles [6),7)]. Structural templates are developed from examples of fault systems throughout the world and from analogue models, and can be used by the interpreter to correlate data – particularly seismic data and to understand the kinematic evolution of fault systems. Development of structural templates will allow the geological evolution of the site to be interpreted and put into context.
- ***Three-D geological modelling***, by integration of available data. The Three-D geological modelling needs to take account of all data available, to be fully traceable and to be archived in a suitable fashion.
- ***Validation of the Three-D geological model***, using quantitative techniques [8)]. In order to evaluate the validity of Three-D geological models, particularly in faulted sedimentary

terrains, it is important that the Three-D interpretation is verified. This can be achieved using software such as Three-D Move [9].

- ***Evaluation and assessment*** of further data requirements to refine the Three-D geological interpretation. The development and evaluation of Three-D geological models can be used to guide the next cycle of data acquisition and interpretation, by indicating key areas of uncertainty in the interpretation.

3.2 Key EarthVision Concepts for GEOMASS

3.2.1 Geologic Structure Builder and Sequence File

As noted in Section 2.1, geological models are constructed in EarthVision by combining a series of Two-D gridded surfaces, which represent either rock boundaries (referred to as horizons in EarthVision) or faults. The Two-D grids are combined into an integrated Three-D model using an EarthVision program called the Geologic Structure Builder. The Geologic Structure Builder can model complexly layered, faulted and non-faulted geological sequences. Using the Geologic Structure Builder, fault-surface intersections, and the resulting fault blocks are constructed according to a fault hierarchy, specified by the user or automatically generated by the program. Structural horizons are automatically intersected and truncated by a geological sequencing technique based on user-specified depositional, erosional, and unconformable surface relationships. The geological sequence is then partitioned into the fault blocks.

The information on which the geological model is constructed using the Geologic Structure Builder is stored in a text file known as the Sequence File. The Sequence File contains all of the information needed to create a Three-D geological model, including the names of the grids that are used to represent faults and horizons, and the relationships that define how these grids are combined.

In GEOMASS, a specially developed program has been written to export the geological structure constructed in the Geologic Structure Builder to a series of ASCII-formatted files that define the geometry of geological surface in Three-D using a grid. These files are then used to recreate the geological structure as a FracAffinity flow network, based on the information in the Sequence File.

3.2.2 Formula Processor

Another key EarthVision program for modelling using GEOMASS is the Formula Processor, which is a file manipulation program that can perform a wide variety of complex grid and field operations on grids, ASCII files, or constants.

Typically, the Formula Processor is used for calculation of pressure and saturation fields based on nodal heads calculated by FracAffinity. These fields can then be modelled in the Geologic Structure Builder, and analysed as illustrated in Chapter 5.

4. Hydrogeological Modelling with GEOMASS

4.1 Flow Equations and Modelling Approach

In FracAffinity, the domain is discretised as an unstructured network (see Section 4.2). The nodes of the network represent regions of space which have head and saturation values, while the connections between nodes represent flow paths. Groundwater flow is modelled by solving for the head at each node and by balancing the groundwater flows along the connections, with consideration of storativity effects. This network is equivalent to a finite-difference discretisation of the governing equations.

4.1.1 Saturated-Flow

For saturated flow calculations, the discretised equations on a FracAffinity network are written as:

$$\sigma_i \frac{\partial h_i}{\partial t} = \frac{1}{V_i} \sum_{\text{conn i-j}} C_{i,j}(h_j - h_i) + q_i \tag{4.1.1}$$

where the i and j subscripts indicate nodes, and:

h is the head (m),
σ is the specific storage coefficient (m^{-1}),
t is the time (s),
V is the volume (m^3),
C is the conductance (m^2/s), and
q is a source of water, usually zero ($m^3/s/m^3$).

The conductance is a calculated from the hydraulic conductivity and geometric data:

$$C_{i,j} = \frac{K_{i,j} A_{i,j}}{d_{i,j}} \tag{4.1.2}$$

where:

K is the hydraulic conductivity (m/s),
A is the cross sectional area (m^2), and
d is length (m).

4.1.2 Unsaturated Flow

For unsaturated flow, the situation is similar to saturated flow, and the nodal equation, which is based on the mixed form of Richards' equation, is:

$$\phi_i \frac{\partial s_i}{\partial t} + \sigma_i \frac{\partial p_i}{\partial t} = \frac{1}{V_i} \sum_{\text{conn i-j}} C_{i,j}(p_i, p_j)(p_j - p_i + z_j - z_i) + q_i \tag{4.1.3}$$

where:

p is the pressure head (m),
z is the vertical coordinate (m),
s is relative saturation (-),

ϕ is the porosity (-),
σ is the specific storage coefficient (m^{-1}),
t is the time (s),
V is the volume (m^3),
C is the (pressure- or saturation-dependent) conductance (m^2/s), and
q is a source of water, usually zero ($m^3/s/m^3$).

For this equation, there are additional relationships between p and s, that have to be determined to derive the conductance given the nodal pressure heads (actually the dependence is on the saturation). These relationships are commonly referred to as the water retention model, and, in FracAffinity, these are most typically modelled using the van Genuchten relationships [10].

The non-linear relationship between hydraulic conductivity and pressure head is controlled by two functions. The first relates the pressure head (usually referred to as suction for when the pressure head is negative) and the saturation or moisture content. The second relationship is that between the saturation and hydraulic conductivity. FracAffinity provides the ability to use the van Genuchten [10] model for these functions, as well as allowing the user to define their own functions as piece-wise linear continuous curves.

However, the use of the van Genuchten relationship is known to cause difficulties with flow solvers reaching convergence [11),12]. This is due to the highly-non-linear nature of the relationships, as illustrated in Fig. 4.1. Therefore, the non-linearity in the van Genuchten relationship can be reduced by specifying a non-linear fraction that smooths the relationship between hydraulic conductivity and pressure head (Fig. 4.1).

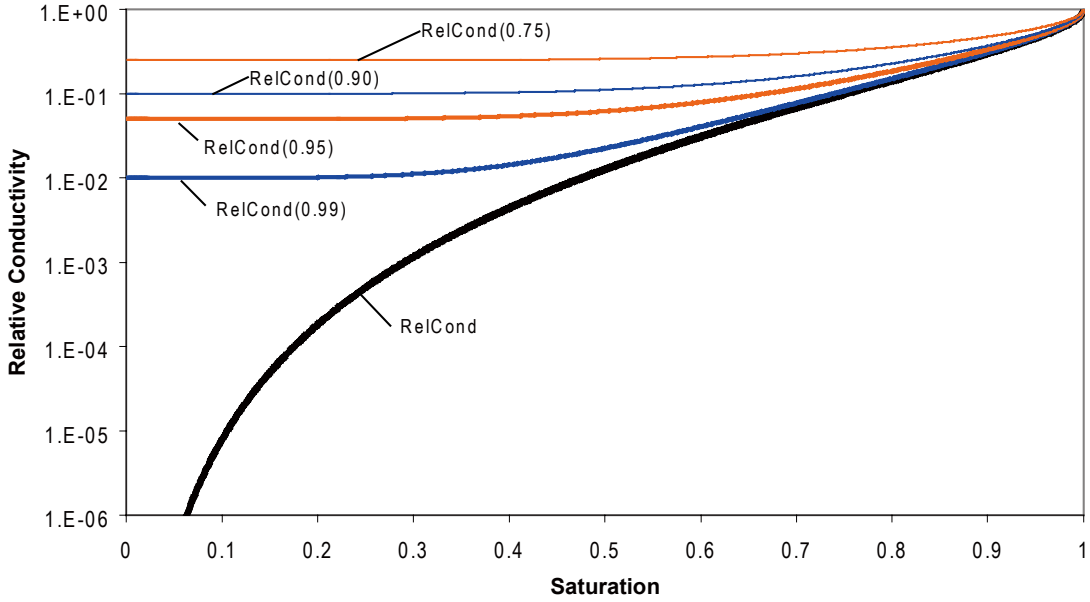


Fig. 4.1 Effect of Non-linear fraction for a Van Genuchten WRM (Water Retention Model). The effect of the non-linear fraction on the relative conductivity for a typical Van Genuchten retention model[10]. The values in the legend indicate the non-linear fraction represented by the associated curve.

4.1.3 Solving the Flow Equations

FracAffinity employs a range of schemes for solving for the flow for the different types of flow problems addressed:

- Steady-state saturated flow: This system is linear and requires the solution of a symmetric matrix equation. This is solved using an iterative approach that employs a preconditioning matrix.
- Steady-state unsaturated flow: This system is non-linear because of the dependence of conductance on saturation, and is solved by an iterative approach utilising two schemes, a Picard scheme based on a linear approximation to the non-linear problem, and a Newton scheme that uses a full Jacobian matrix for the non-linear system, where the Jacobian represents the best linear approximation to a differentiable function near a given point.
- Transient flow: Transient flow is calculated using the DYLAN solver which employs a variable-order predictor-corrector method based on an algorithm published by Byrne and Hindmarsh [13]. At each timestep, DYLAN performs the following calculations:
  - Predict the solution at the end of the step.
  - Correct the prediction, using Newton non-linear iterations.
  - Determine the error, and decide whether to accept the step.
  - Determine the duration of the next timestep (or for a re-taken step if the error was unacceptable).
  - Determine the order of the method for the next timestep.

4.2 Flow Networks (FracAffinity network) and the Hybrid Medium

As noted above, FracAffinity uses a network approach to simulate groundwater flow in the hybrid medium. For the intact rock, a rectilinear grid is generated and the approach is equivalent to finite difference. The generation of the hybrid medium occurs in a stepwise manner with features being added one at a time. First, the intact rock is generated and then each discrete feature, beginning with the DDFs, is added one at a time. As the discrete features are added, the network is modified by re-routing connections via nodes on the feature being introduced. Following the addition of the discrete features, the engineering features are added to the network (engineering features are discussed in Section 4.4).

4.2.1 Flow Network Properties

The network that is used in FracAffinity is made up of legs (also referred to as connections in FracAffinity) and nodes. Each connection connects precisely two nodes. A node can be at the end of many connections (there is no maximum number of connections per node).

FracAffinity stores the physical properties needed to define the system by associating them with either connections or nodes. The assignment of these properties occurs during the grid generation phase. Fig. 4.2 summarises the physical properties, showing those that are associated with nodes and those that are associated with connections. Each connection is divided into two parts, indicated by the thick cross line in the middle of the figure. Splitting the connections in this way enables the correct averaging of conductivities to be made. The break into two parts is directly related to the volume that each node is associated with. If we view each node as being at a cell centre, then the break occurs at the cell boundary (Fig. 4.3 and 4.4).

Note that each part of a connection also has a short section adjacent to a node and a long section. This structure has been established to ensure that fast pathways are not established across low-conductivity features, and that low-conductivity features do not block flow along high-conductivity features.

4.2.2 Generating the Flow Networks

FracAffinity creates separate grids for the intact rock and for each of the discrete features. These are then merged into the final hybrid network. The intact rock grid is initially rectangular in the x-y plane with regions outside the model domain removed. The vertical structure is determined from the top and bottom surfaces and a user-defined refinement surface. A grid of cells with rectangular cross-section but varying vertical extent is created. Any cell that is completely outside the model region is omitted. Then a network is created by placing a node at each cell centre and creating connections between neighbours. Where a cell has no neighbour, a boundary node is added and a connection made to this. Additional nodes are then added and connected to handle any thin geological layers.

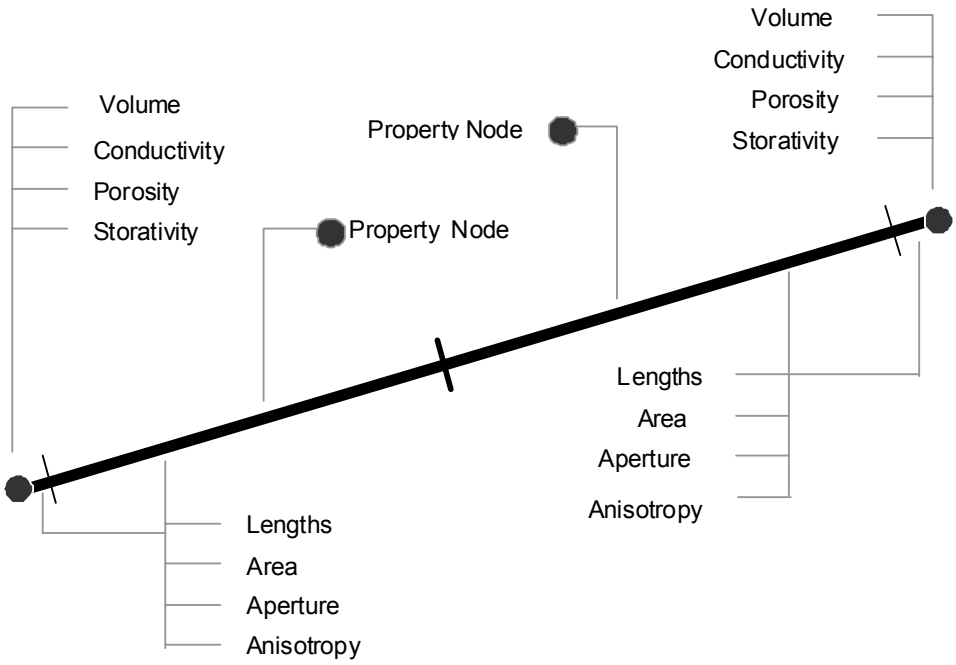


Fig. 4.2 The properties assigned to network nodes and connections in FracAffinity. The data stored for nodes and connections are the same for intact rock and discrete features except for the aperture, which is irrelevant for intact rock connections.

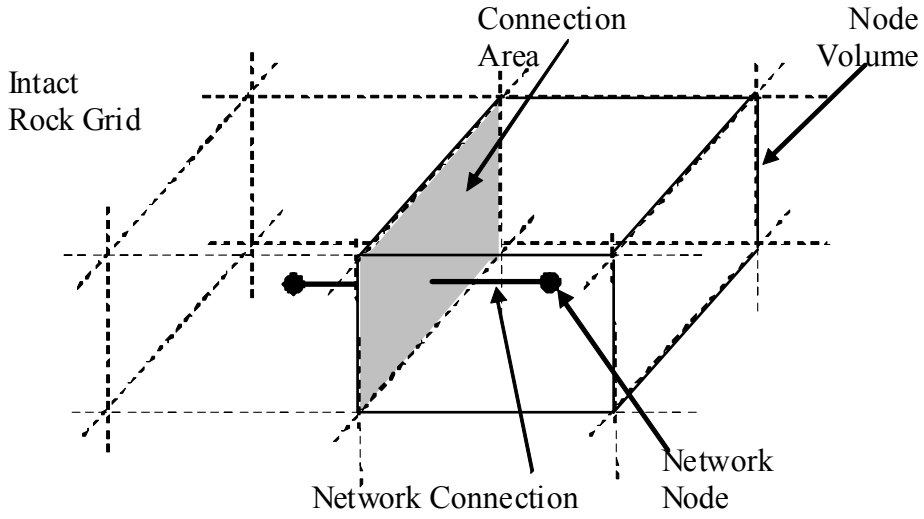


Fig. 4.3 The geometric properties for the intact rock.

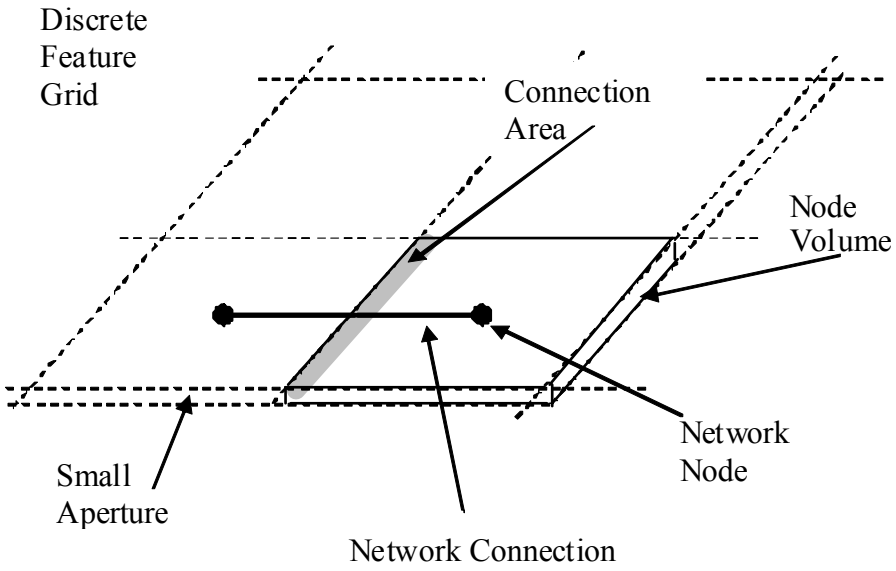


Fig. 4.4 The geometric properties for a discrete feature.

Discrete features are two-dimensional, so the network that is generated for them is also two-dimensional. The network produced for each DDF is the Two-D equivalent of that produced for the intact rock. A rectangular grid of cells is created, with, in cases where a polygon has been defined, cells omitted if any part of the cell lies outside the polygon that defines the limits of the fault. Nodes are placed at cell centres and neighbours are joined by connections. Boundary nodes are created where no neighbouring cell exists.
Each SDF is planar. As the SDFs are generated, those that are completely outside the model region are discarded. Features are cropped by a bounding cuboid, so that networks are not generated for large parts of features that are outside the model region. A network is directly generated. There is no need for an intermediate SDF grid because the properties are all uniform. A two-dimensional coordinate system in the plane of the fracture is used.

4.2.3 Combining the Flow Networks

The final network that is used in FracAffinity for solving the flow problem, the hybrid medium network, is created by merging the networks which form the intact rock, DDFs and SDFs and engineering features. Merging is done one feature at a time. That is, merging starts with the intact rock network and the DDF networks are added one by one. Then the SDF networks are added one by one. There is no creation of a combined network for all DDFs or SDFs.
It is difficult to draw pictures of this in Three-D, but the algorithm is clear from considering a Two-D network to which One-D features are added, as shown in Fig. 4.5. There are three connections in the Two-D network that cross the One-D feature. Each of these is rewired as shown in Fig. 4.6. Notice that none of the nodes are moved. Each of the connections that cross the new feature is split in two and re-routed via the nearest node. The final step in the merging process is to modify the data (and create new data) for the re-routed connections.

An example of a flow network, visualised against the geological model that it represents, is illustrated in Fig. 4.7, and without the geological model in Fig. 4.8.

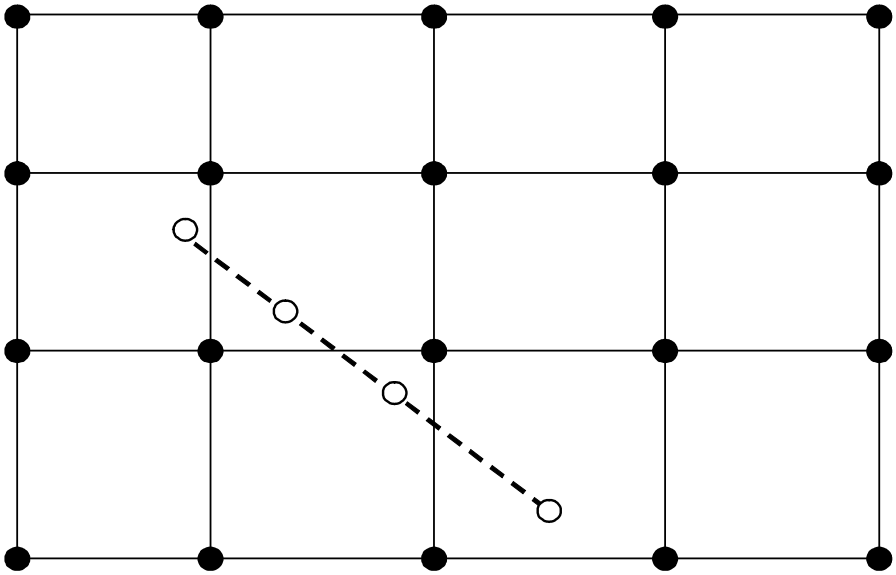


Fig. 4.5 Two-D network with One-D feature to be merged.

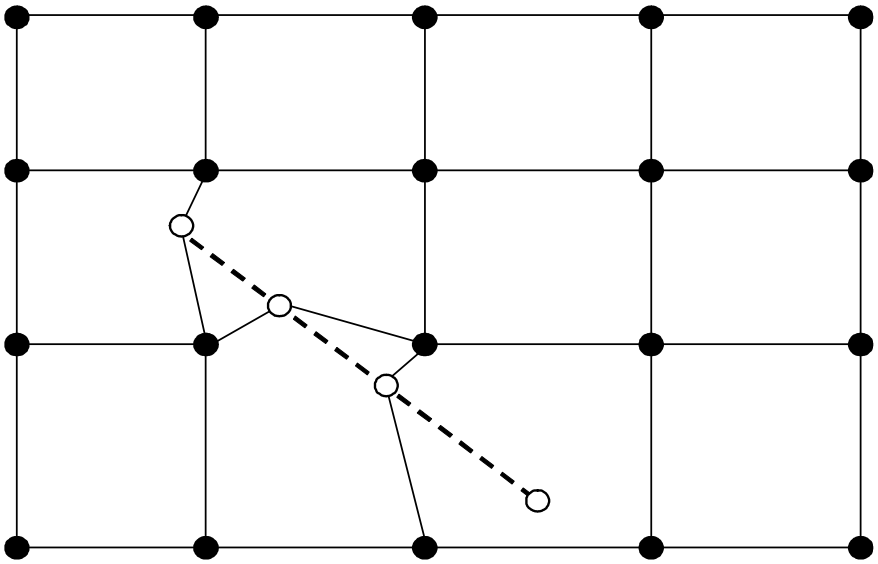


Fig. 4.6 Two-D network after merging the One-D feature.

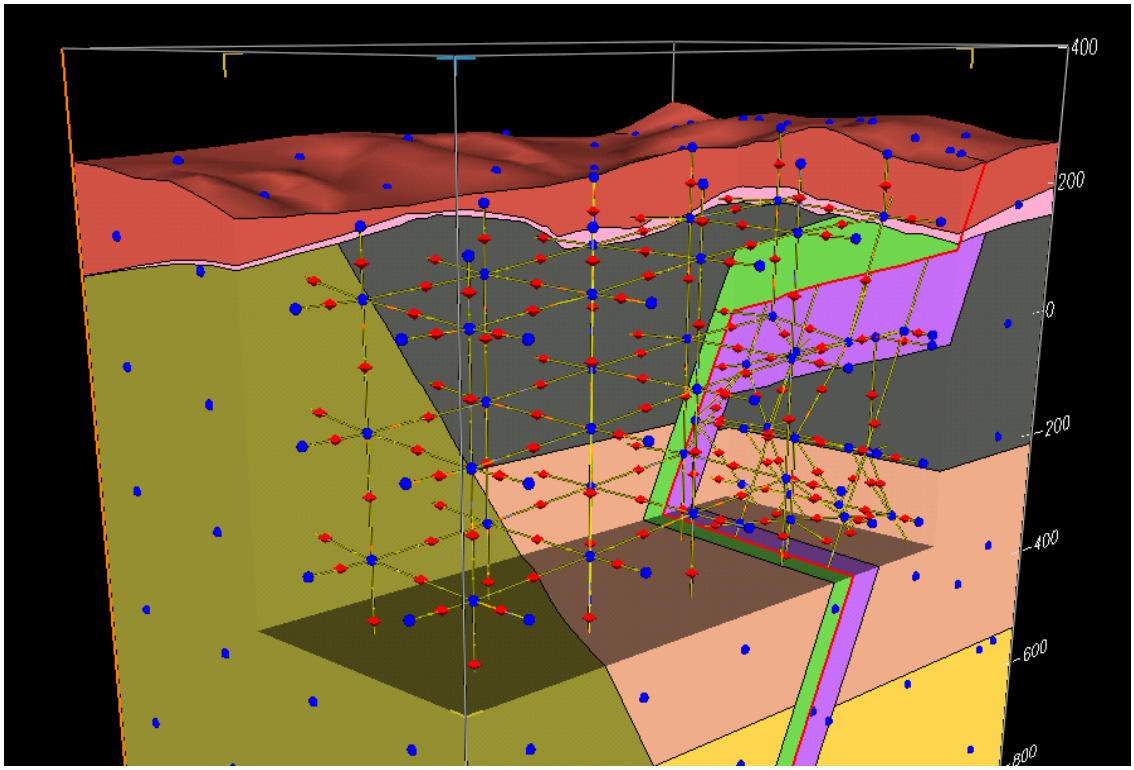


Fig. 4.7 Illustration of a hybrid medium flow network, visualised against the geological model from which it has been derived. In the figure, blue spheres represent nodes and red diamonds mark the centre of connections. Note that in the model, the grid spacing is deliberately coarse to allow for visualisation of nodes and connections. The figure illustrates the regularity of the intact rock grid, and how the hybrid medium flow network becomes more irregular around a DDF (Deterministic Discrete Feature) (marked by the red line through the geological model).

This is a blank page.

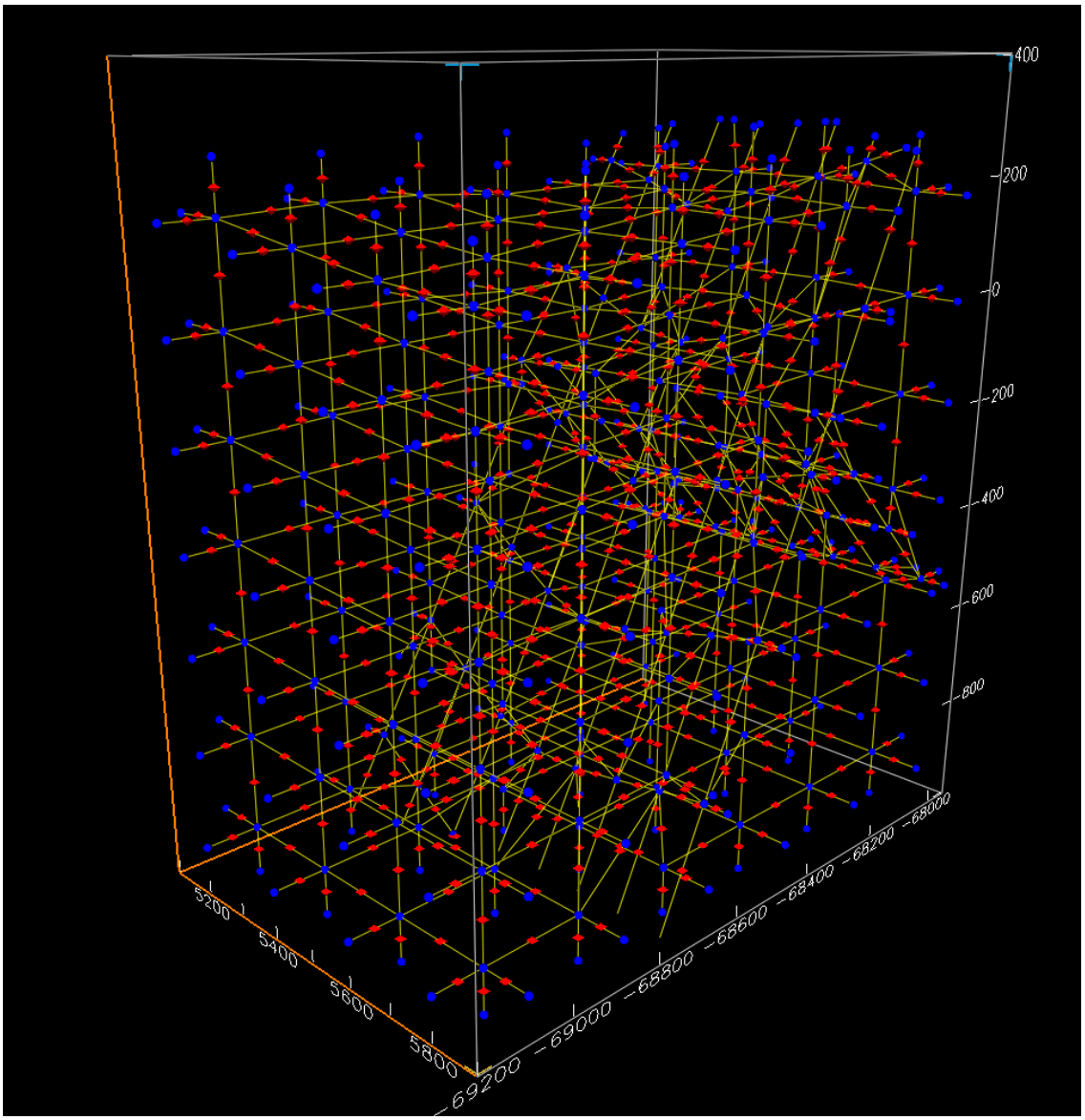


Fig. 4.8 Illustration of a hybrid medium flow network. In the figure, blue spheres represent nodes and red diamonds mark the centre of connections. Note that in the model, the grid spacing is deliberately coarse to allow for visualisation of nodes and connections. The figure illustrates the regularity of the intact rock grid, and how the hybrid medium flow network becomes more irregular around a DDF (Deterministic Discrete Feature) (represented by a dipping plane of nodes).

This is a blank page.

4.3 Heterogeneity at Multiple Scales

A key feature of GEOMASS is the ability to investigate the impact of heterogeneity of a geological system on groundwater flow at all scales. This section describes the key features that are used to model heterogeneity.

4.3.1 Large-scale Geological Structure

The overall approach adopted in GEOMASS, one in which the geological structure is developed in a dedicated geological modelling system and the use of a flow network to solve for flow, allows for detailed geological structures to be considered in the hydrogeological model.

Features that can be included within GEOMASS that are often difficult to represent in hydrogeological models include:

- ***Fault Structures***: In nature, faults have complex geological structures, which impact significantly on fluid flow. Faults are typically segmented, i.e. they are not laterally continuous and include breaks (relay zones) across which groundwater can flow. In addition, at the large scale, faults contain deformation zones that modify the hydraulic properties in the hangingwall and footwall of faults. Furthermore, faults can have complex folds (e.g. ramp-flat structures), and contractional faults cause problems with some modelling systems due to repetition of stratigraphic surfaces in the vertical orientation (multiple elevations are required at the same horizontal location to describe the location of a stratigraphic surface across a contractional fault). All of these situations are readily modelled in GEOMASS, in an intuitive and straightforward fashion.
- ***Unconformities and Facies Changes***: Several hydrogeological modelling codes have difficulty with modelling sedimentary facies or lithological variations within the geological sequence. Often, modelling systems require that a rock layer is modelled across the area of interest, even though in reality it may pinch-out. In GEOMASS, the user has the flexibility to model any geological structure.

Fig. 4.9 illustrates the modelling of some of these features using EarthVision; such structures are passed seamlessly to FracAffinity, using the Sequence File approach discussed in Chapter 3.

Indeed, it is this Sequence File approach that is particularly powerful in ensuring that the impact of geological structure can be properly investigated in GEOMASS. With finite element approaches to hydrogeological modelling, the development of the flow grid can be time consuming and typically constrains the hydrogeological modelling to consideration of a single structural representation. Uncertainty in geological structure then has to be investigated through manipulation of parameter values. In GEOMASS, multiple geological interpretations can be readily tested, with modification of the flow network possible in hours.

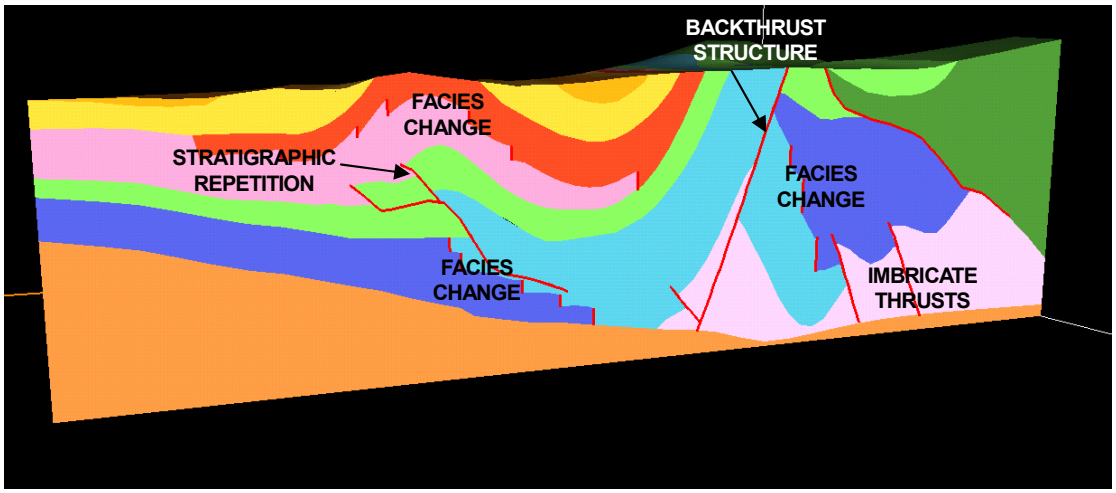


Fig. 4.9 Model of a fold and thrust belt, illustrating some of the complex geological structures that can be modelled within GEOMASS [14].

The flexibility and speed of the geological modelling means that the GEOMASS system is a powerful tool for sensitivity studies, and can be used to ask important "what-if" questions, for example to discover the potential impact of undetected faults, or alternative conceptual models of fault linkage.

The other aspect of the integration of the geological structure with the hydrogeological modelling is the ability to specify hydraulic properties on a zone-by-zone basis. This is typical for most groundwater modelling codes, however, GEOMASS has enhanced capabilities in this area, for example the ability to model small-scale fractures individually for specific fault blocks (Fig. 4.10).

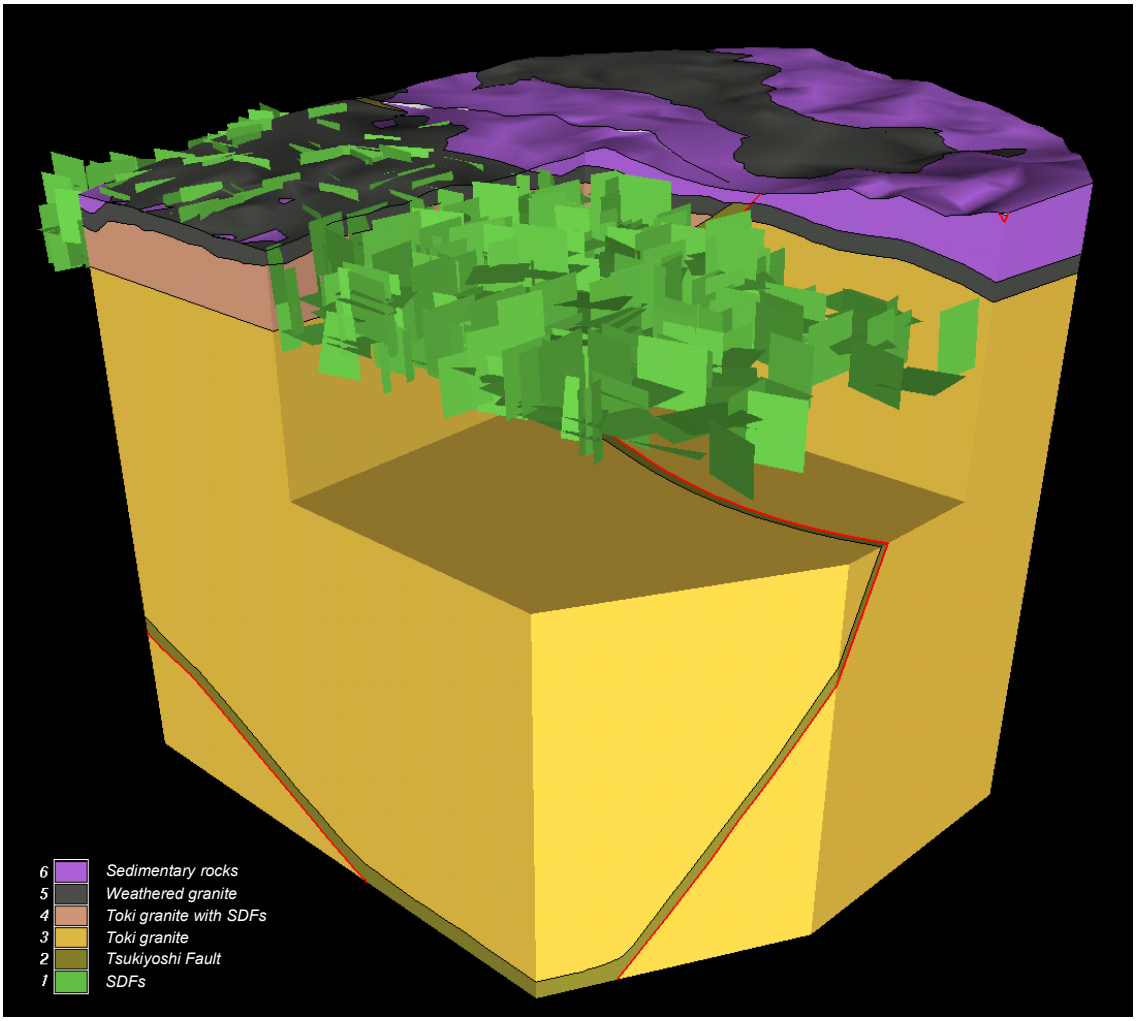


Fig. 4.10 Geological model of the Tono area with representation of water-conducting fracture zones (WCFZs) as SDFs (Stochastic Discrete Features) in the upper part of the Toki Granite in the hangingwall of the Tsukiyoshi Fault. The SDFs were generated in the zone labelled *graniteSDFs*, which has its base at –200 mAOD (meters Above Ordnance Datum). Three sets of WCFZs are represented by two sub-vertical and one sub-horizontal set. 860 SDFs were generated by FracAffinity and are visualised in the geological model.

4.3.2 Heterogeneous Hydraulic Conductivity Distributions

In addition to modelling the impact of the large-scale geological structure on groundwater flow, routines have been introduced in FracAffinity for modelling medium-scale heterogeneity, through flexible specification of hydraulic properties. There are three main ways that heterogeneous property distributions can be introduced into FracAffinity at the medium scale: import of conductivity from an external file, specification of anisotropic conductivity tensors, and generation of a fractal property field.

The fractal property generator used in FracAffinity is a Three-D extension of the Mid-Point Displacement Method, a well-known Two-D method developed by Peitgen and Saupe [15]. The approach uses borehole data to development a variogram, and to successively refine the mesh

This is a blank page.

on which fractal properties are derived from the variogram. The model has been successfully tested on borehole data from the Tono area [16].
In GEOMASS, fractal interpolation of rock mass properties can be undertaken for intact rock and DDFs. The conductivity fields generated for these features are illustrated in Fig. 4.11 and 4.12.

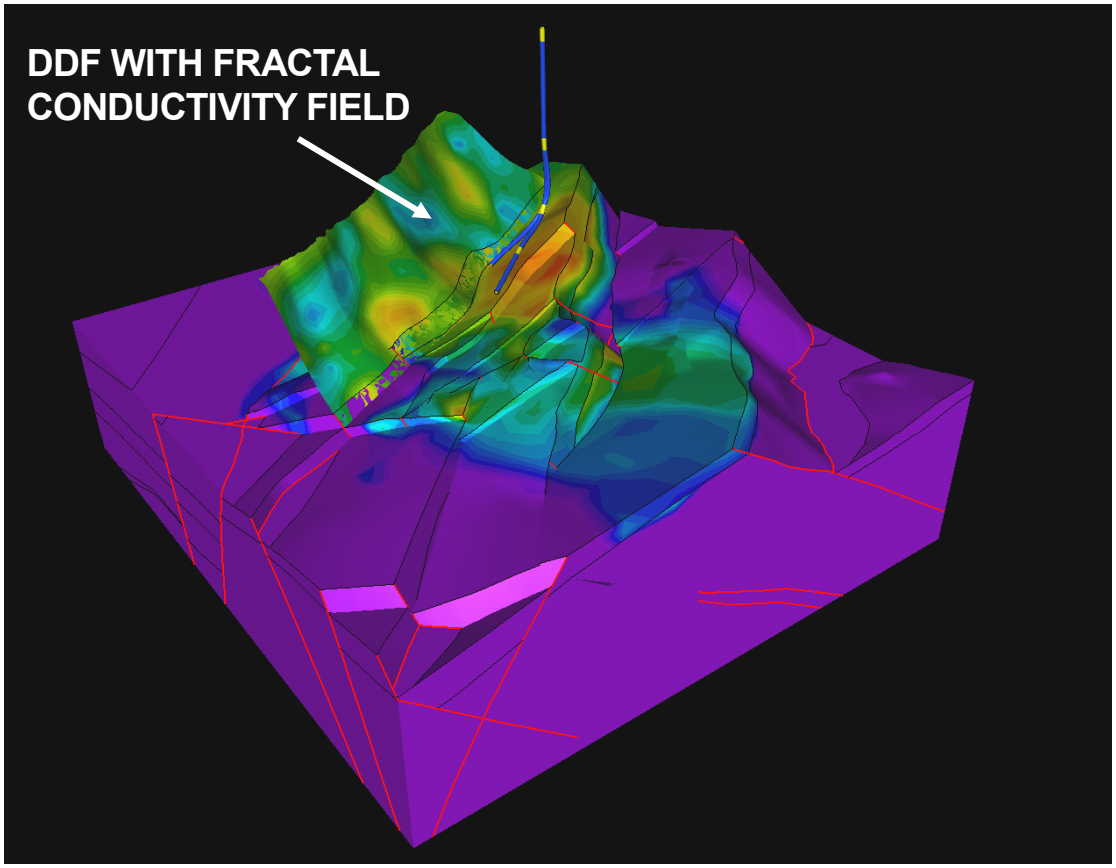


Fig. 4.11 Illustration of the interpolation of a fractal conductivity field on a fault (DDF: Deterministic Discrete Feature) using GEOMASS. The colour coding for the geological model illustrates drawdown associated with pumping from a borehole, indicated by the blue cylinder.

This is a blank page.

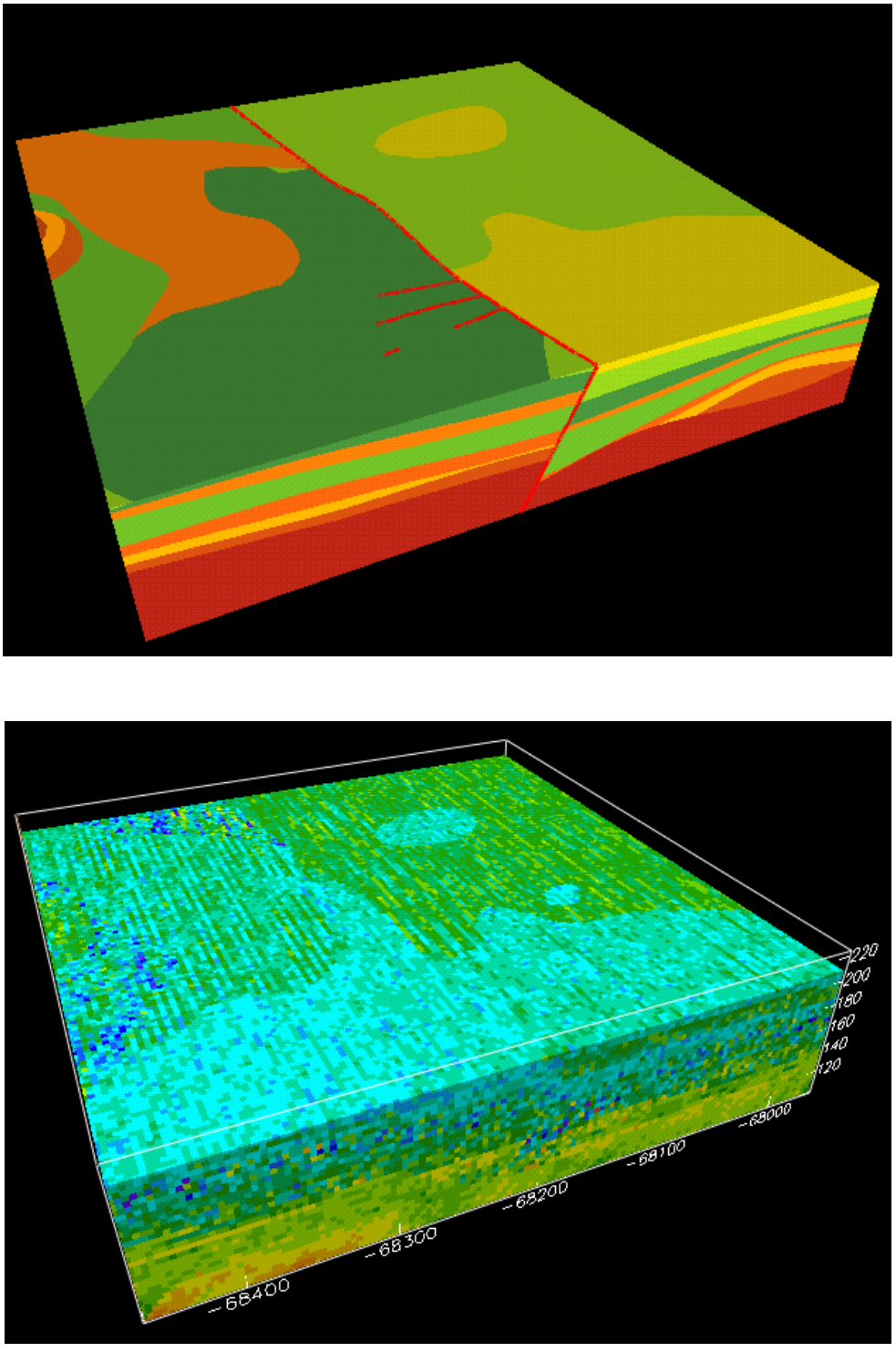


Fig. 4.12 Interpolation of heterogeneous rock mass properties on a zone-by-zone basis using GEOMASS. The upper figure illustrates a Three-D geological model of the sedimentary sequence in the Tono area and the lower figure an interpolation of borehole resistivity (in ohms, resisticity is an indicator of hydraulic conductivity) based on the Three-D geological interpretation using a fractal model. Resistivity values are not shown because the key aspect of the figures is to compare the geological structure pattern with the resisitivity interpretation pattern.

This is a blank page.

4.3.3 Discrete Feature Heterogeneity

As noted in Section 4.3.1, geological structure frequently has an impact on hydraulic conductivity distributions. At the small scale, FracAffinity has several methods for modelling this impact:

- ***Anisotropic conductivity for discrete features:*** In FracAffinity, the user can specify a scaling factor, so that the flow across a discrete feature is scaled in proportion to the flow along the feature. This is useful for situations when faults are composed of a core of low-permeability material (e.g. fault gouge), which restricts flow across the fault, and an outer zone of deformed high-permeability material (e.g. cataclasite), which permits flow in the plane of the fault.
- ***Enhanced conductivity at discrete feature intersections:*** In certain instances, especially in extensional environments, intersections between fractures can be preferential flow paths, and can focus flow. Therefore, FracAffinity has the capability to explicitly model fracture intersections, and for the user to specify an enhanced conductivity value for the intersection. As with all special features in FracAffinity, this capability can be selected and applied flexibly by the user. It requires the modification of the flow network, since, as illustrated in Fig. 4.13, the routine method for generating the hybrid medium flow network in FracAffinity would not explicitly model fracture intersections.

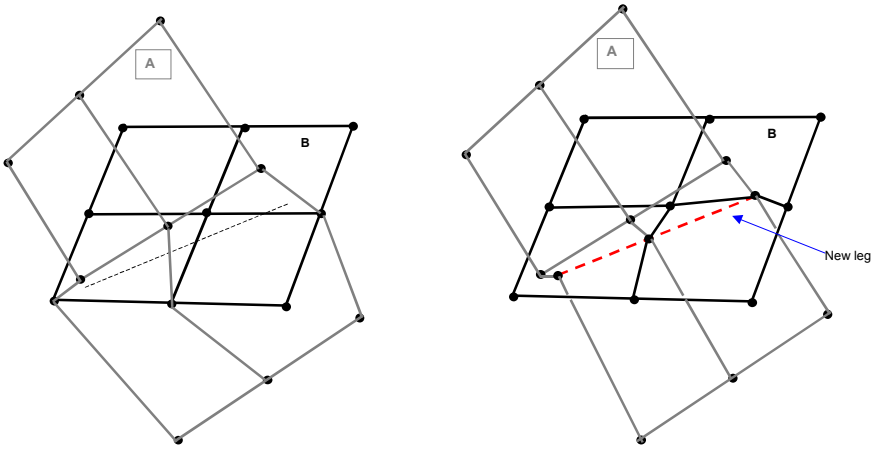


Fig. 4.13 Schematic diagram illustrating the gridding of the intersection of two discrete features in FracAffinity. The left figure illustrates the case without explicit modelling of fracture intersections, and the figure on the right illustrates the case with explicit modelling of these intersections. Fracture A is introduced to the flow network first and is intersected by Fracture B. For the case on the left the true fracture intersection is not modelled by the re-routing of connections associated with Fracture A. For the case on the right, additional nodes are placed in the model to represent the intersection, and connections to these nodes are made from both fractures.

4.4 Excavations of Shafts and Galleries, and Boreholes

FracAffinity has novel routines for the modelling of dynamic excavation of shafts, galleries and access ways (referred to as excavations) and for modelling boreholes. In addition, static shafts and galleries can be modelled, but here we concentrate on excavations and boreholes.

4.4.1 General Excavation Model

Time dependency is a key feature of FracAffinity, and the code includes a general excavation model that allows for the dynamic excavation of an engineered structure (a shaft, gallery or inclined access tunnel), by specifying a piece-wise linear excavation rate that is applied during transient modelling.

The general excavation model works by identifying a series of nodes that become active as the excavation is constructed, and a corresponding set of nodes that become inactive. This is illustrated in Fig. 4.14.

The nodes that are active at any one time are boundary nodes that have suitable boundary conditions applied (typically seepage, atmospheric pressure or specified flux). The lining of the excavation can be simulated by specifying a no-flux boundary over any defined length of the feature, at any time after construction.

The general excavation also allows for modelling the effects of construction by allowing dynamic rescaling of conductivity near the feature. This allows the impact of grouting or the development of an excavation damage zone to be simulated in the model.

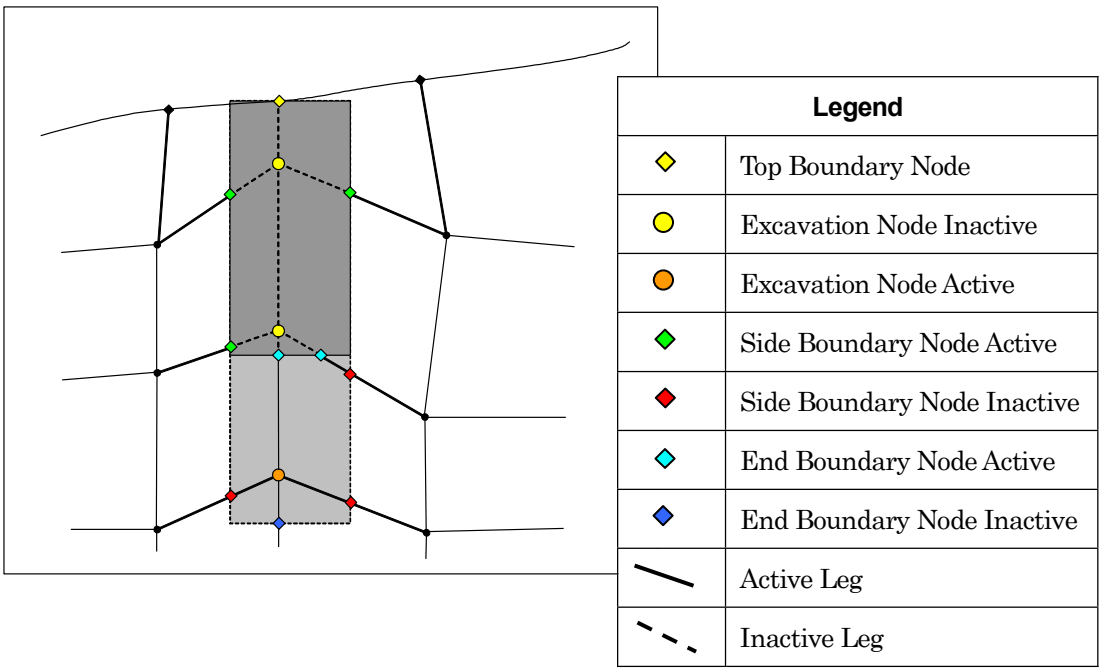


Fig. 4.14 Schematic illustration of excavated and boundary nodes for dynamic shaft excavation. The illustration shows an intermediate stage in shaft excavation, with the excavated shaft shown in dark grey and the shaft not yet excavated in light grey.

4.4.2 Borehole Model

In FracAffinity, pumping tests can be modelled by introducing borehole sections into the flow network. A borehole section is considered to be a hydraulically-isolated section of a borehole, i.e. a section of a borehole isolated by a packer system. No allowance is made for any possible leakage across the packers.

Introducing a borehole into the flow network requires consideration of how the high hydraulic conductivity of the borehole will interact with the moderate hydraulic conductivity of the surrounding rock. In FracAffinity, this is achieved by developing a suitable method for discretising the borehole section, linking the borehole to the flow network and then assigning the hydraulic conductivity. Flow along the borehole assumes Poiseuille flow [17], and is relative to the square of the radius of the borehole.

The borehole section is defined as a series of points and these are linked to the flow network by adding a series of additional connections, as illustrated in Fig. 4.15, and by reassigning the area and volumes for the connections under consideration onto the new connections. However, Fig. 4.15 illustrates a rather idealised situation, and it is more probable that a borehole will lie closer to one grid node than another, as illustrated in Fig. 4.16. Therefore, the properties are assigned in proportion to the square root of its volume and the inverse square of its distance from the borehole.

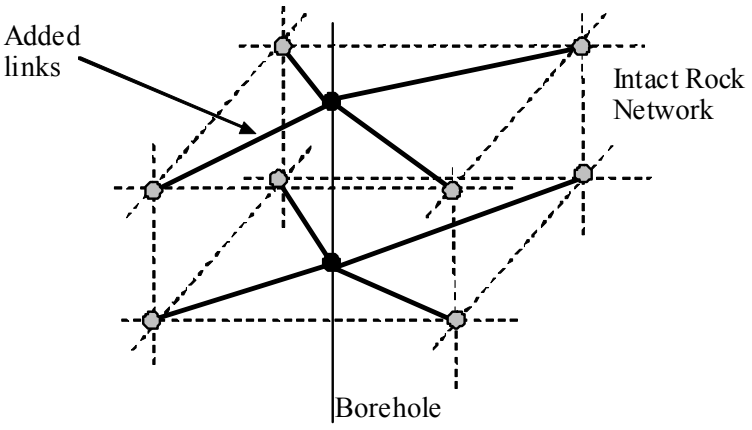


Fig. 4.15 The links for connecting borehole nodes to the intact rock network for a simple situation.

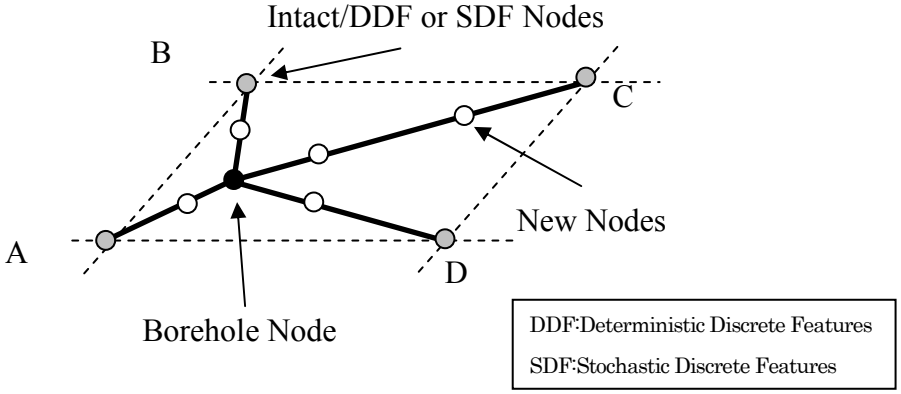


Fig. 4.16 Typical set of links for connecting a borehole to the flow network.

4.5 Grid Refinement

A special attribute of FracAffinity is the ability to model unsaturated flow on regional scales (i.e. km scale). This is a particularly challenging task, since unsaturated flow systems contain steep pressure gradients, and, as described in Section 4.1, the dependence of conductivity on pressure and saturation means that the flow equations are highly non-linear. Dealing with non-linearity in the flow equations has been managed by development of the transient flow solver.

This section deals with the routines that have been introduced in FracAffinity to increase the grid resolution in areas where steep pressure gradients occur. In addition to refining the grid for modelling unsaturated flow, grid refinement algorithms have been introduced to cope with special geological features, such as thin formations.

4.5.1 Local Refinement

For regional-scale modelling, the intact rock is generally set with grid spacing of several tens of metres. However, for detailed modelling of features of interest, for example flow into a shaft, grid spacings of these sizes is inappropriate, because head gradients are large. Grid sizes of a few metres are required when including features such as shafts or a moving water table in regional models, especially for unsaturated flow models.

Therefore, FracAffinity has the capability to introduce cuboidal regions in which the grid can be refined to any user-specified dimensions. These cuboidal regions refine the grid in the region specified and allow nested modelling without the need for transferring boundary conditions from one model to another. Fig. 4.17 illustrates the use of a locally refined cuboid for modelling a shaft and movement of the water table on a regional scale.

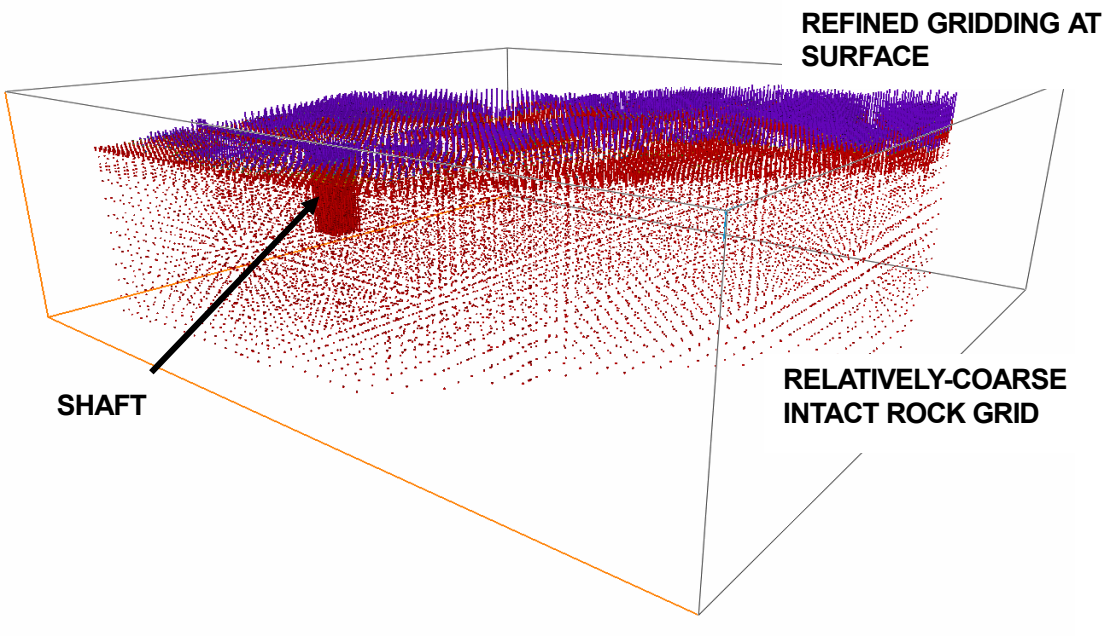


Fig. 4.17 Illustration of refined gridding around the shaft and the ground surface.

4.5.2 Thin Formations

The geological environment of the Tono area, for which GEOMASS has been developed, contains thin, laterally persistent lithological and stratigraphic units. These include the top of the Toki Granite, which has been highly weathered to a depth of several metres to several tens of metres, and the basal unit of the overlying sedimentary succession, which is a high-conductivity conglomerate of a few metres thickness.

These units are highly important to the regional groundwater flow system. However, for regional intact rock grid spacings of a few tens of metres, it is possible that these features are not suitably represented in a simple flow network. An illustration of the potential problems with thin formations is illustrated in Fig. 4.18 and 4.19.

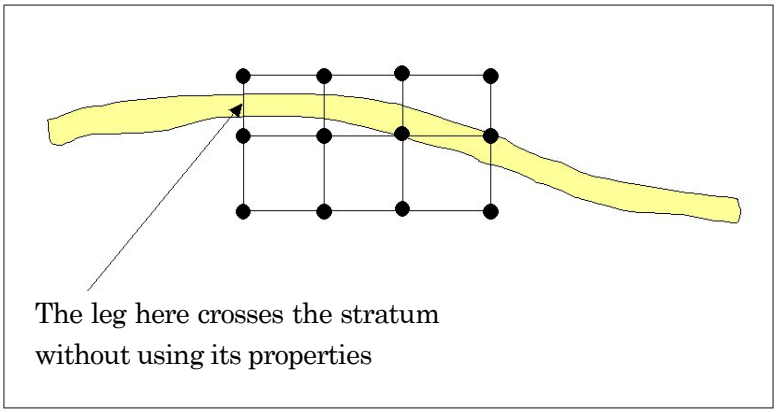


Fig. 4.18 Potential problem with thin formations of low conductivity. In the example shown, the connection on the left would take its properties from the location of the nodes, which are not located in the thin formation.

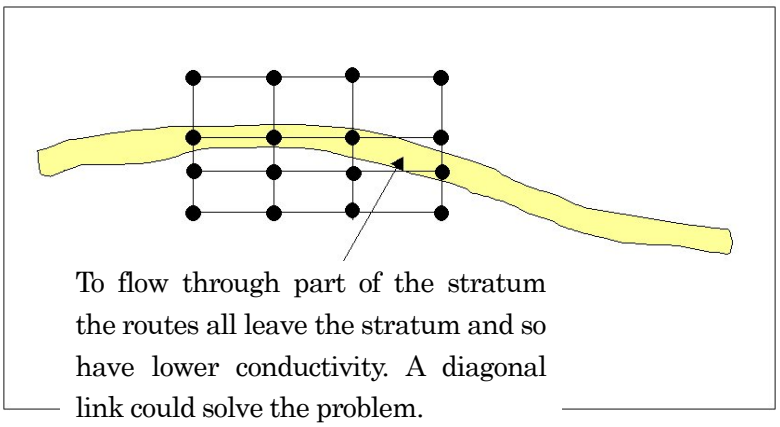


Fig. 4.19 Potential problem with thin formations of high conductivity. In the example shown, flow along the high-conductivity formation would not be modelled because there is no connection along the formation on the right-hand side.

The potential problems that could be introduced by thin formations are avoided in FracAffinity by adding new nodes along legs that cross formations specified as thin formations.

4.5.3 Water Table Refinement

Water tables will follow the shape of the topography, and, therefore, it is not always suitable to use a cuboid to define a region in which the grid should be refined, to model unsaturated groundwater flow on a regional scale. This is particularly true where there is a significant change in elevation from one side of the model to another, e.g. from a recharge to a discharge zone, or where the topography is particularly undulating.

In FracAffinity, the user can refine the flow network by introducing a Two-D grid that represents an approximate height of the water table. Additional layers of nodes are introduced into the model around this grid, thereby increasing the resolution for modelling movements of the water table. This capability is not restricted to modelling of water tables, and can be used to increase the resolution around any sub-horizontal feature, i.e. it is another flexible modelling tool within FracAffinity.

5. Visualisation and Post-Processing with GEOMASS

In addition to providing the geological framework on which the flow network is based, the link between EarthVision and FracAffinity that is provided in GEOMASS allows for the visualisation and analysis of modelling undertaking in FracAffinity using EarthVision. A special feature of this capability is that it allows the hydrogeological aspects of the modelling to be visualised alongside the geological interpretation, which provides a basis for enhanced understanding and analysis of the groundwater flow modelling results.

There are several ways of visualising calculations undertaken in FracAffinity using EarthVision. These focus on the use of the 3D Viewer (see Chapter 3). The interaction can be used to support the following tasks:

- Generation of the flow network.
- Analysis of the results of groundwater flow modelling.

These tasks are described in this section.

5.1 Visualisation of the Flow Network

A key aspect of any modelling project is to ensure that the grid on which the model is to be based is a reasonable representation of the geological interpretation and conceptual groundwater flow model. As illustrated in Fig. 4.18 and 4.19, it is possible that the results from the groundwater flow modelling are affected by unsuitable grids. Therefore, it is recommended that the flow network is visualised prior to simulation of groundwater flow.

The flow network can be visualised in several ways, and many of the methods available in GEOMASS have been illustrated previously in this document. The following key methods are highlighted here:

- Visualisation of nodes and connections. Fig. 4.6 and 4.7 illustrate the visualisation of the flow network alongside the geological model in the EarthVision 3D Viewer. This type of visualisation can be used to check that key flow features, such as low-conductivity faults or high-conductivity layers, have been properly represented in the model. Note that the visualisation of the full grid can be difficult and time consuming. Therefore, it is generally advisable to focus visualisation on key features or regions of the model, rather than visualising the entire grid at once. FracAffinity has the capability to generate the different parts of the hybrid medium independently, and this capability can be useful in checking the flow network.
- Visualisation of node locations. A useful way of checking the specification of refined gridding zones is to visualise just the nodes in the flow network (Fig. 4.17). This can be useful in checking that locally refined regions have been set correctly in the input file, and such checking is good practice during modelling.
- Visualisation of property values. All of the property values that are used in FracAffinity can be visualised and checked using EarthVision. This is particularly useful for verifying that there have been no errors in the input files, and that the correct conductivity, porosity and storage coefficients are being used. An example is provided in Fig. 4.12.
- Visualisation of SDF networks. The SDF networks generated in FracAffinity can be visualised and checked in EarthVision (Fig. 2.4 and 4.10).

5.2 Visualisation of Simulation Results

GEOMASS provides additional capabilities for understanding the results of groundwater flow modelling by allowing visualisation of Three-D head, pressure and saturation fields and by visualisation of pathlines. The Three-D nature of the flow field can be visualised and interrogated by progressively slicing through it parallel to the horizontal or vertical axes. Typically, the head field is exported from FracAffinity to EarthVision, and file manipulation routines (accessed through the EarthVision Formula Processor program as described in Chapter 3) are used to calculate the pressure (by subtracting the elevation from the head) and then the saturation (by gridding the horizon representing zero pressure). In addition, time-lapse Three-D animations of head, pressure and saturation field can be generated, for example to understand the combined spatial and temporal impact of a pump test on groundwater heads (Fig. 2.5).

Fig. 5.1-5.3 illustrate a series of visualisations of groundwater flow modelling results visualised using EarthVision.

Pathlines use the implied flow velocities in the network to follow non-sorbed particles. A steady-state or transient flow field can be used. The pathline plot will show the location of a particle, usually based on a calculation of the local effective flow velocity at a node.

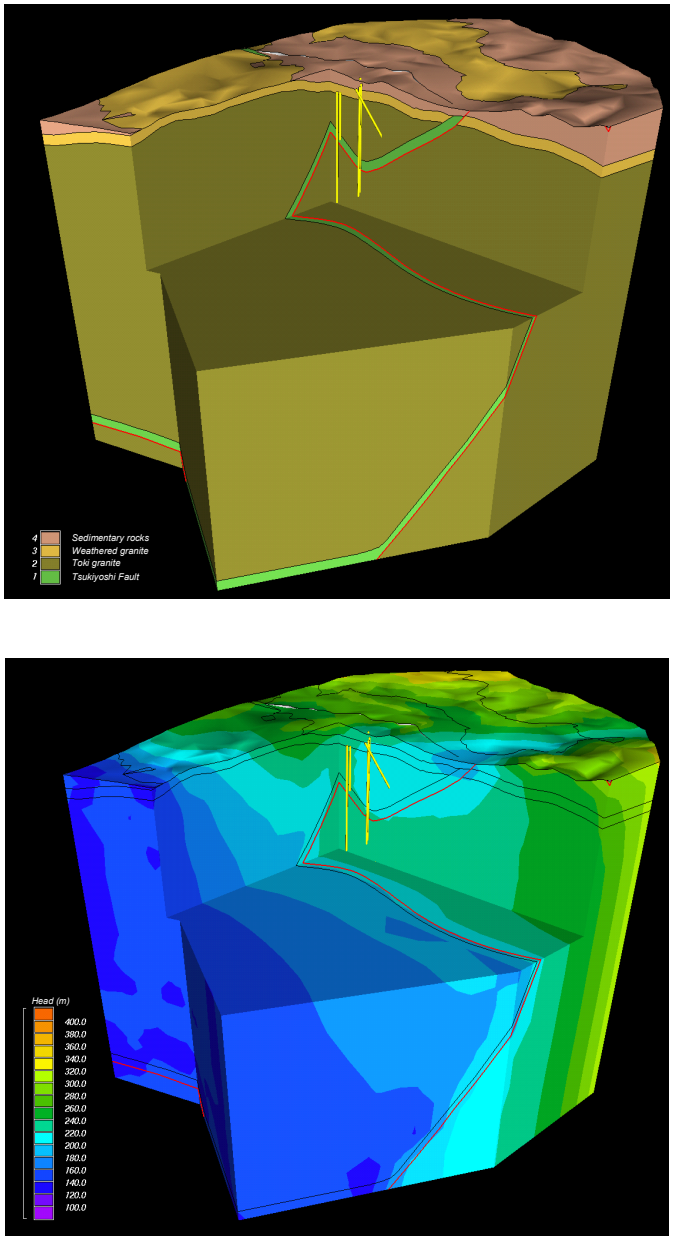


Fig. 5.1 Visualisation of the head field from a groundwater flow model of the Tono area. The figure shows the geological model (top) and the Three-D head field (bottom). The Three-D head field clearly shows the influence of a low-permeability fault (the Tsukiyoshi Fault), which is shown by the red line in both figures, on the head field, with large drops in head from one side of the fault to the other. In the lower figure, the colours show the heads in 40 m head intervals. The colouring in the top figure indicates different rock units, with the bright green zone representing a high-permeability deformation zone in the hangingwall of the Tsukiyoshi Fault.

This is a blank page.

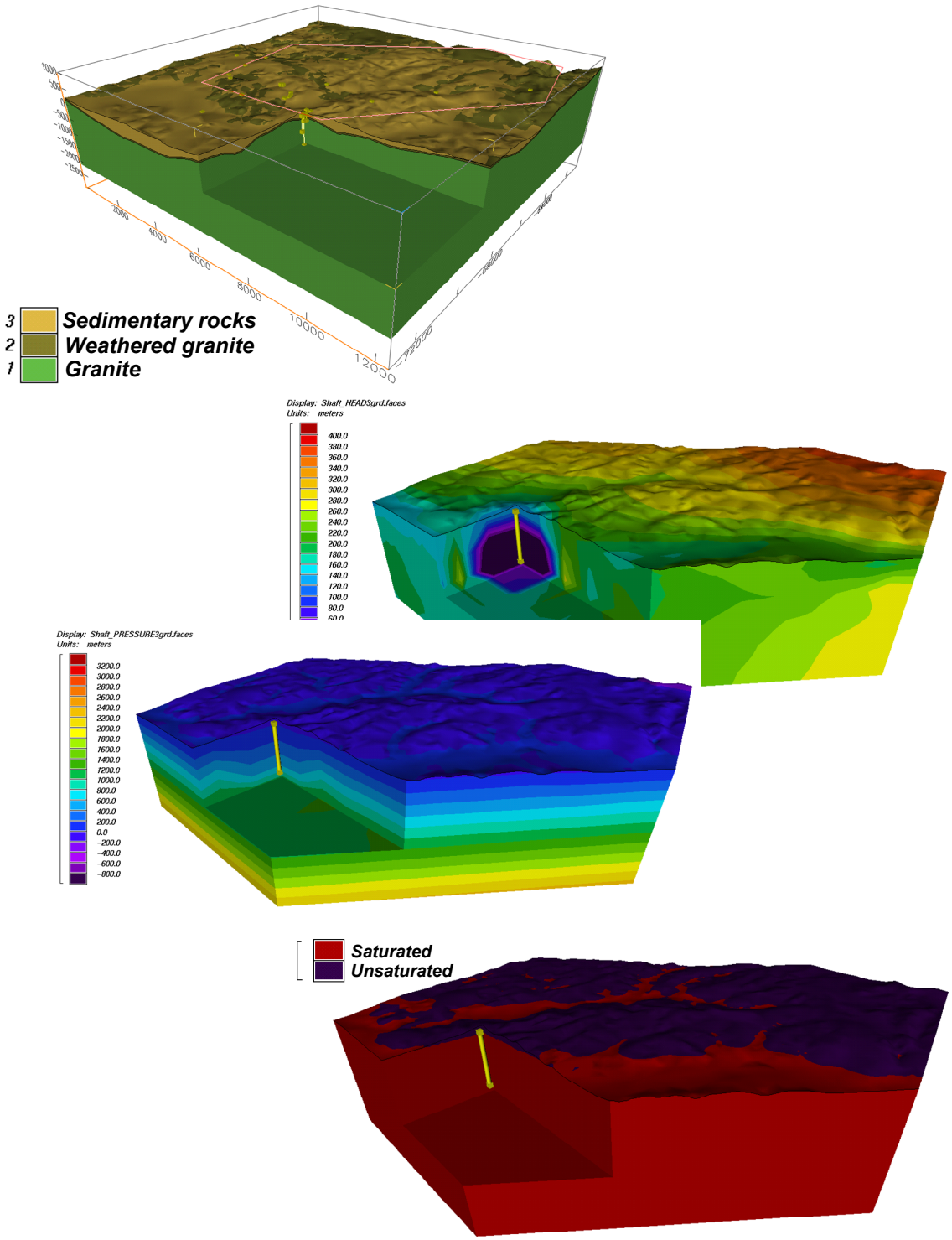


Fig. 5.2 A series of visualisations illustrating the geological structure model (top), Three-D head field (middle right), Three-D pressure field (middle left) and Three-D saturation field (bottom) for the Tono area, for the steady-state flow field calculated at the end of a transient unsaturated flow model with a dynamic shaft indicated by the yellow line. In the saturation figure the red indicates full saturation.

This is a blank page.

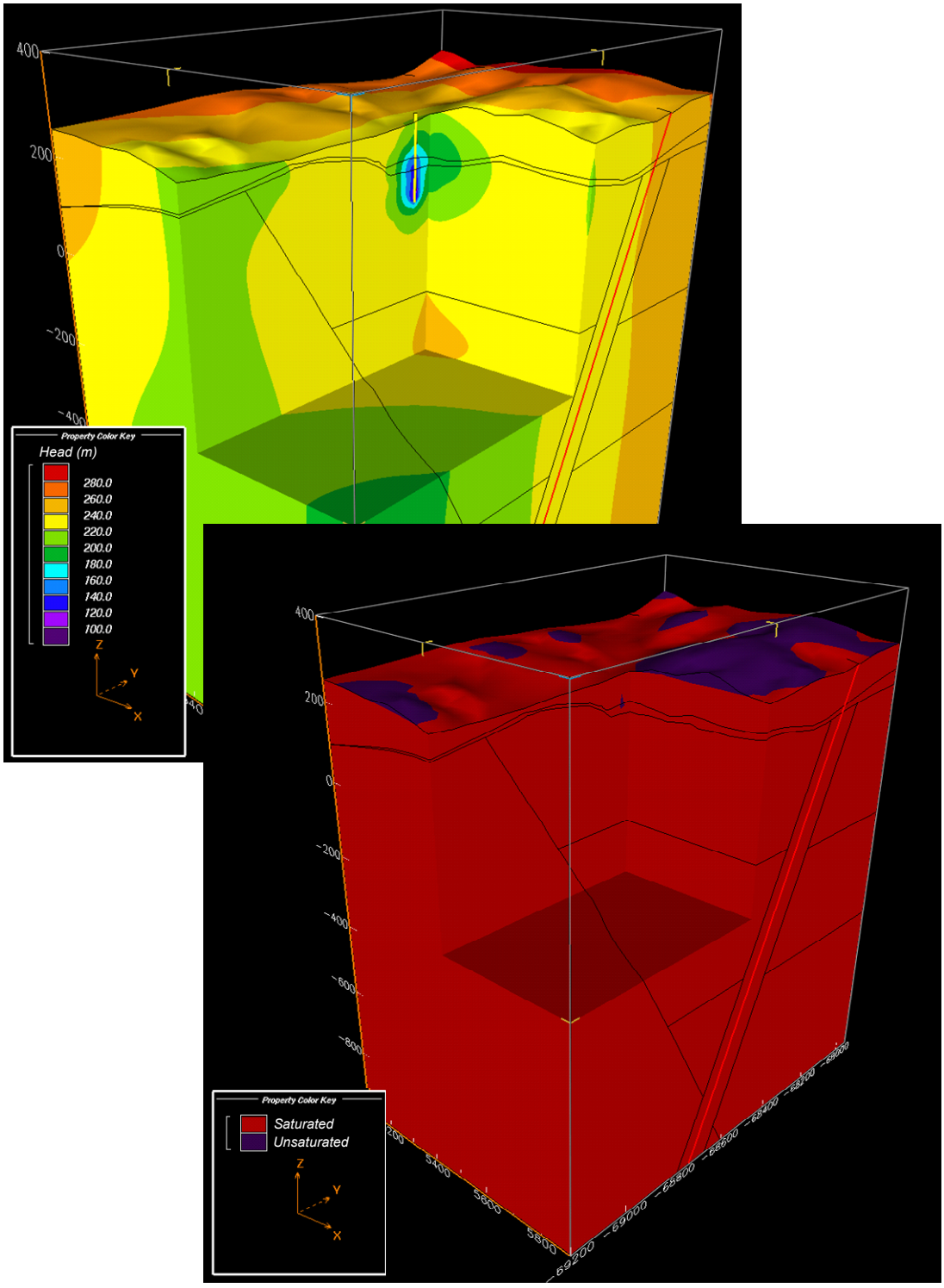


Fig. 5.3 Visualisation of groundwater flow modelling based on the geological structure model illustrated in Fig. 2.4 (upper figure – this model did not include the SDFs). The top figure illustrates the head field in response to excavation of a shaft illustrated by the yellow line. The bottom figures illustrates the saturation represented by pressure; negative pressure (unsaturated) is shown in purple, and positive pressure (saturated) is shown in red.

This is a blank page.

5.3 Example of Sensitivity Calculations at Tono

The site in which the MIU URL is being constructed by JAEA is characterised by several regional lineaments (faults), with uncertain hydrogeological properties. However, these structures are expected to affect significantly the hydrogeological response to shaft excavation, in particular the flux of groundwater into the URL, and the magnitude and distribution of the drawdown around it. Therefore, JAEA has used GEOMASS to undertake sensitivity calculations to investigate the role of regional faults on the head distribution in the MIU Construction Site [18].

Fig. 5.4 illustrates how the visualisation of interpolations of nodal heads against the fault interpretation can help to assist in the understanding and communication of such sensitivity modelling using GEOMASS. The plots shown in Fig. 5.4 have been developed by gridding the nodal heads within the geological model, and show a horizontal slice through the calculated head profile.

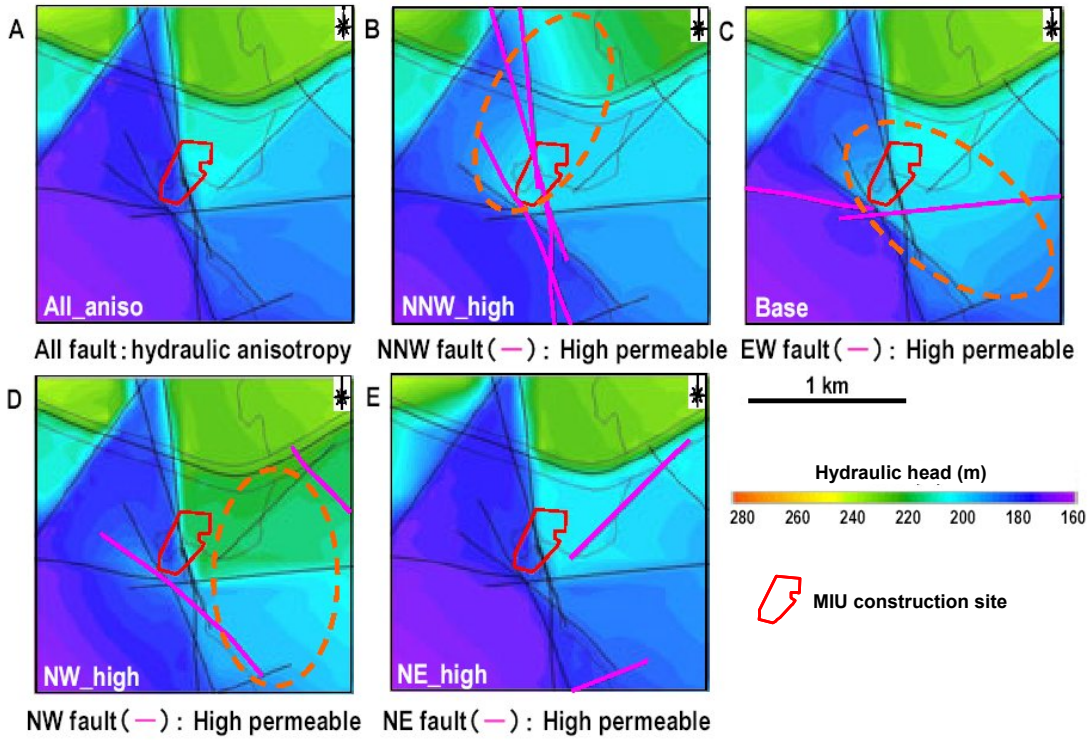


Fig. 5.4 Results of sensitivity studies during Step 2(surveys on the existing boreholes to assess the properties of the Toki granite and the structural discontinuities in Surface-based Investigation Phase) of the site characterisation. The thick blue lines are faults, and the thin blue lines boundaries between lithological units. In A, all faults are modelled with anisotropic transmissivity, which is higher than the intact rock. In B, only NNW-trending faults (shown by pink lines) are modelled with high transmissivity. In C, only EW-trending faults (shown by pink lines) are modelled with high transmissivity. In D, only NW-trending faults (shown by pink lines) are modelled with high transmissivity. In E, only NE-trending faults (shown by pink lines) are modelled with high transmissivity. Areas identified by dashed ellipses are those that JAEA consider to be significant differences between the model runs.

This is a blank page.

6. Application of GEOMASS

6.1 Workflow

The typical workflow used in modelling with GEOMASS is illustrated in Fig. 6.1, including the linking between EarthVision and FracAffinity.

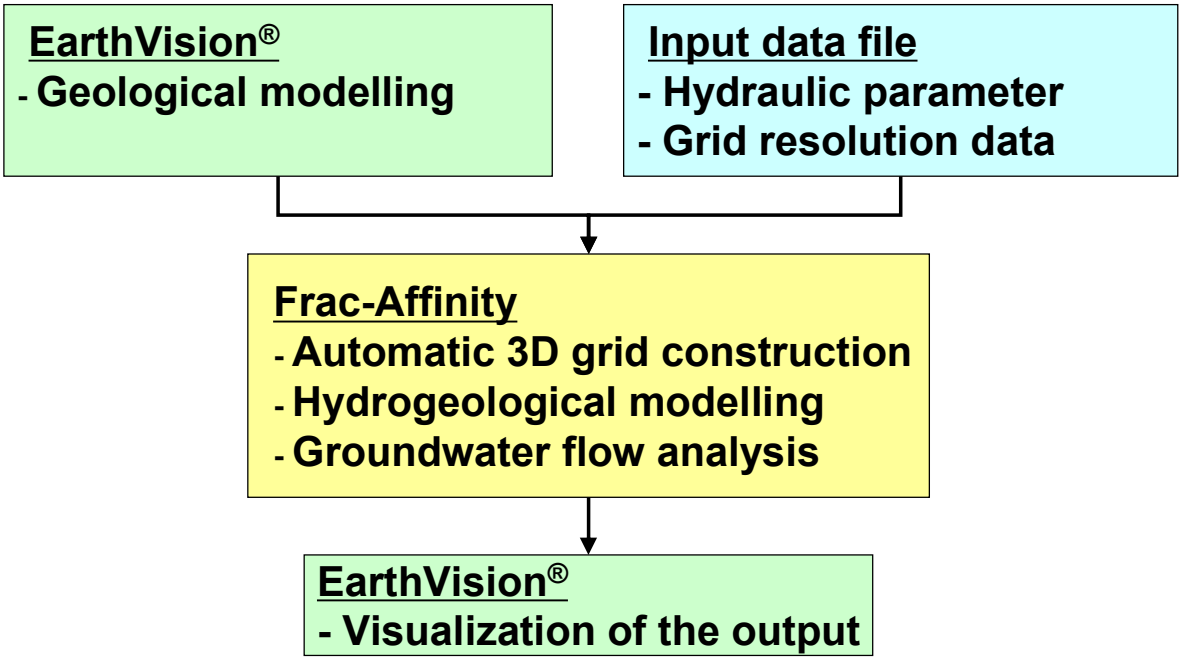


Fig. 6.1 Workflow used in the GEOMASS system.

6.2 EarthVision

EarthVision has a easy-to-use graphical user interface (GUI), illustrated in Fig. 6.2. The interface allows access to a series of tools to be applied in developing a geological model, the most important of which are:

- ***Analytical Tools:*** Data, grids, plots and associated files can be manipulated using the Graphic Editor in conjunction with the Digitizer Editor, which allow interactive object positioning, plot-montage construction, and map-drafting capabilities; seismic data editing and digitising are also available. These tools are accessed via the Edit menu.
- ***Modelling Tools:*** Two-D and Three-D surface grids are developed from surface and property data, and can be integrated into geological models using the Geologic Structure Builder, which is a program for constructing Three-D models of geological sequences, from simple layer-cake stratigraphic successions, to complexly layered, faulted and non-faulted structures based on Two-D and Three-D surfaces. Property data can be introduced to these models, as illustrated in Fig. 2.2.
- ***Visualisation:*** Geological maps can be viewed as cross-sections, surface maps or in Three-D by use of the 3D-Viewer. The 3D-Viewer is used to view both the raw data (prior to any interpolation), Two-D grid surfaces and Three-D grid volumes after calculation, complex geological structures, and finally the full Three-D model, once this file is created.

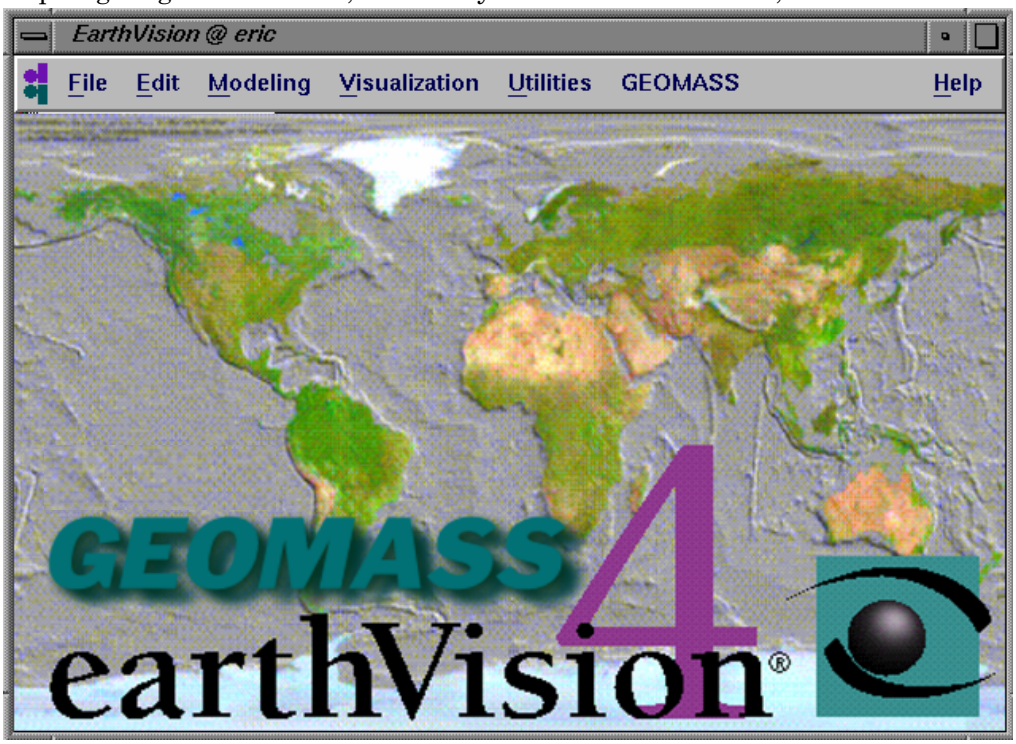


Fig. 6.2 The EarthVision GUI (Graphical User Interface). Note that this version of the interface has been modified to include GEOMASS scripts. The figure shows the GUI for EarthVision Version 4.

6.3 FracAffinity

FracAffinity is accessed via an interface application, the FracAffinity Interface, which allows the user to specify the data necessary to run the FracAffinity flow solver and to translate the output of FracAffinity into a form which can be used by EarthVision. The user interaction with the FracAffinity Interface is via specified data files and command line prompts. There are two main modes of operation: pre-processing to model a geological system in FracAffinity and post-processing FracAffinity output files for visualisation or analysis.

6.3.1 Pre-Processing – FracAffinity Input

The FracAffinity Interface is started for pre-processing from the command line when within the required FracAffinity Interface directory. This directory will usually be the one containing the stratigraphic information (i.e. the Sequence File and related surface files) exported from EarthVision. A typical directory structure for FracAffinity files is shown in Fig. 6.3, although the system is flexible and a user can define their own directory structure.

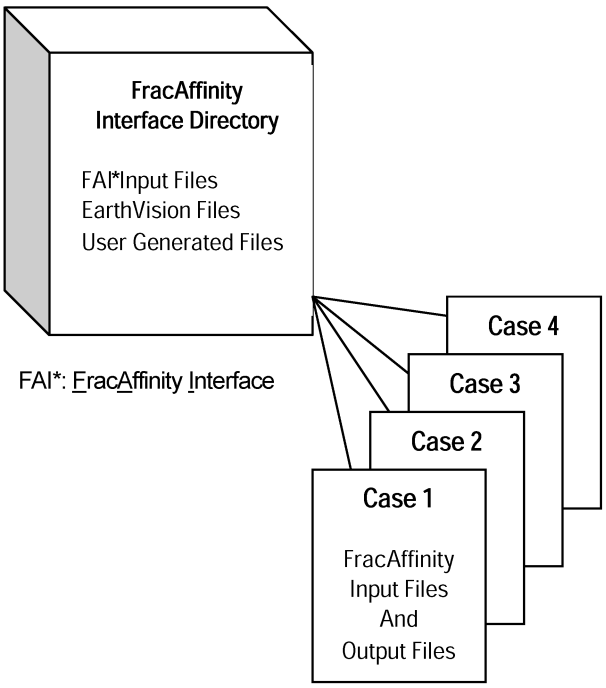


Fig. 6.3 Directory structure for FracAffinity.

The input mode of the FracAffinity Interface is used to define the geological structure and hydrogeological properties, the boundary conditions and the flow solver controls used in the simulation. The information required to model a geological system in GEOMASS is outlined below.

- ***Geological Framework.*** Information on the geological framework of a region containing the FracAffinity model region is specified in the Sequence File, which defines the fault blocks bounded by faults (which may also be DDFs) and the stratigraphic sequences in each fault block. There are two types of information for the fault blocks that form the

model region (see Fig. 6.4). The first type specifies the fault block name, the name of the fault that divides it, the file containing the dividing fault surface, the bounding polygon and the names of the faults block above (hangingwall fault block) and below (footwall fault block) the dividing fault. The second type of information concerns the sequence of stratigraphic zones within each fault block. This includes the names of the stratigraphic zones in the fault block and the names of the files containing the data specifying the upper surfaces of the zones.

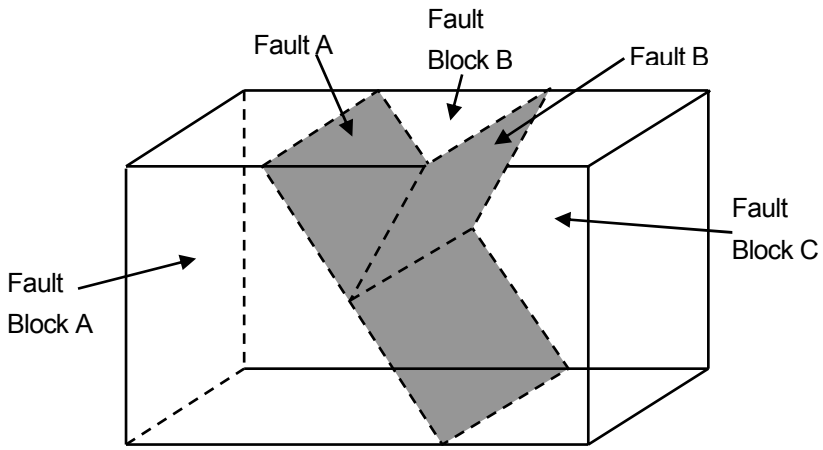


Fig. 6.4 A simple fault block decomposition of a region.

- ***Model Region Information.*** The Model Region defines the external boundaries of the region that is to be modelled by FracAffinity. It is composed of a polygonal boundary defined in the *xy* plane and is bounded above and below (in the *z* direction) by surfaces. Therefore, the sides of the model region are vertical. The model region may also include internal boundaries such as those modelling shafts or galleries, information in boreholes and the excavation timetable for time-dependent shafts. Regions where conductivity is to be rescaled may also be specified.
- ***Shafts, Galleries and Boreholes.*** The user can define an internal model region in the form of a dynamic excavation, horizontal cylindrical gallery or a vertical cylinder, and a borehole. A timetable for the lining and construction of a shaft is entered for dynamic excavations. Multiple excavations are allowed, although they must not overlap or be too close together.

  If boreholes are included, then the input file must contain information about the geometry of the boreholes and gridding information. Boreholes are defined in sections, which are hydraulically separated, either by a packer or through being in different boreholes.
- ***Flow Solver Controls.*** Before the rock properties are input, the interface will request information related to the flow solver. The flow solver provides a "no flow" option in case the user wishes to construct the hybrid medium but not solve for flow, otherwise the user may select steady-state or transient solutions for saturated and unsaturated conditions.

- ***Intact Rock Model Parameters.*** Parameters that may be defined for the intact rock in each of the different stratigraphic zones include rock conductivity, porosity and storage coefficients. In addition, an intact rock water retention model must be selected if an unsaturated calculation is to be performed, which can either be a van Genuchten model (recommended) or a user-specified piece-wise linear model.

  The global grid parameters are specified, including a parameter to limit the global resolution of the grid by preventing large numbers of very small cells from being constructed. It is also possible to refine the flow network around a Two-D grid to allow for better resolution of the region above and below the water table surface when running unsaturated flow models, especially in transient mode. However, the approach is generic and may be used to refine the grid for other reasons.
- ***Discrete Feature Models.*** Input information on DDFs includes geometry, conductivity, porosity, specific storage coefficient, water retention model and grid. The geometric information is specified by reference to surface and polygon files. The conductivity file defines the model parameters that will be used to generate conductivity values in the DDFs and it also defines whether DDFs will be included in the generation of enhanced conductivity intersections. Porosity and specific storage parameters are defined, as well as the water retention model for unsaturated calculations. Finally, the grid parameters for the DDF gridder are input.

  To generate an SDF rock model, information on the geometry and physical properties (conductivity, porosity and specific storage coefficient) is required. The geometric information is specified by formation name and the model or distribution to be used. The scalar conductivity gives a conductivity at each node and, by default, is isotropic. However, anisotropic conductivity can be defined by specifying different values for the in-plane and normal directions. As for the DDF rock model, a water retention model is required for unsaturated calculations and the grid parameters must be specified.
- ***Boundary Conditions.*** Boundary conditions at specified faces and on engineered features are input in the boundary conditions file. There are two principle options for the type of boundary condition on external boundaries, namely head or flux conditions. For unsaturated cases, suction, recharge and seepage conditions can also be used. Point conditions, such as sources or sinks within the model region, can also be defined.
- ***Flow Solver Output.*** The parameters defined in the flow solver output definition file are the locations at which outputs, typically head, should be calculated. The times at which outputs should be calculated are defined in the flow solver control. The main output types that can be specified in FracAffinity are monitor points, monitor boreholes, flux planes and pathlines. There can be two additional types that can be specified: monitor lines and monitor planes, which are intended as quick ways of generating large numbers of monitor points in a line or a plane.

Once the geological model has been specified, the interface will automatically start FracAffinity and exit, leaving FracAffinity running in the background. The progress of the FracAffinity run can be assessed by looking at output that FracAffinity writes to the terminal and to the FracAffinity log file (see below).

6.3.2 Post-Processing – FracAffinity Output

The main output files created in FracAffinity are:

- ***Log File.*** The information in this file includes diagnostic information such as the date and time at which the calculation started and finished, number of nodes in the problem, the number of SDFs generated and water balance reports. A water balance report is written for each monitor output time. The total water in the system is reported, using the nodal volumes, porosities and saturation values. This is followed by a breakdown of water flows (volumes per unit time). This is presented by "boundary", which includes the internal and external physical boundaries, but also includes storativity and resaturation effects.
- ***General Report Files.*** In addition to the initial head conditions data file, a general output file is produced that gives both data and results for all nodes and legs. The file has three blocks of information: one for nodes, one for boundary nodes and one for legs.
- ***SDF Summary File.*** This output file summarises SDF position, angle, length, width, aperture, conductivity, porosity and storage coefficient values.
- ***Top Surface Outflow File.*** The top surface outflow file lists every boundary point on the top surface and gives the location, flow and flux (flow/area) at that point.
- ***Shaft Outflow File.*** This file lists every active boundary point on the shaft surface and gives the location, flow and flux (flow/area) at that point.
- ***Pathline Files.*** The output files produced in pathline calculations provide detailed information about the path, including time, coordinates, distance, volumetric flow, Darcy velocity, conductivity and porosity.

6.3.3 Post-Processing for EarthVision

The purpose of using the FracAffinity Interface in post-processing mode is to enable the user to select FracAffinity output files either for direct analysis or for visualisation in EarthVision. Illustrations of the post-processing of FracAffinity hydrogeological models are provided in Chapters 4 and 5; this section provides an overview of the output files.

The FracAffinity Interface is started in post-processing mode from the command line. The user selects the form of FracAffinity output they wish to analyse from the following:

- ***Rock Properties Output.*** The data available for visualisation may include intact rock, DDFs, SDFs or the full hybrid medium at a range of output times. The user selects the time at which the conductivity, porosity and specific storage coefficient will be written to a named EarthVision file for visualisation.
- ***Head and Velocity Output.*** The FracAffinity Interface will read the appropriate head, saturation or velocity information (at the output time nearest to the user's selection) from the relevant FracAffinity output file and then write a file with the specified filename in a format suitable for visualisation in EarthVision.
- ***Monitor Points.*** The interface will provide a list of monitor points for which data are available. The FracAffinity Interface will read the monitor output file and write the selected monitor point information to a file in a form suitable for viewing in EarthVision.
- ***Monitor Lines or Borehole Sections.*** The interface will provide a list of monitor points for which data are available. The FracAffinity Interface will read the monitor output file and

write the information for each of the monitor points which comprise the selected monitor line to a file in a form suitable for viewing in EarthVision.

- ***Monitor Planes.*** The interface will provide a list of monitor planes with their parameters for which data is available. The FracAffinity Interface will read the monitor output file and write the information for each of the monitor points which comprise the selected monitor plane to a file in a form suitable for viewing in EarthVision.
- ***Flux Planes.*** The interface will provide a list of flux planes for which data are available. The FracAffinity Interface will read the appropriate FracAffinity flux plane output file and write the information for each of the selected flux planes in a form suitable for viewing in EarthVision.
- ***Pathlines.*** The interface will provide a list of starting points of pathlines for which the data are available. The FracAffinity Interface will read the selected pathline output file and write the information for each of the pathlines in a form suitable for viewing in EarthVision.
- ***Network.*** The data available for visualisation may include intact rock, DDFs, SDFs or the full hybrid medium. If there are excavations in the model then FracAffinity will prompt for the time at which the network is to be produced. The network may be visualised in EarthVision following post-processing with a Perl script.
- ***SDFs.*** If this option is selected, FracAffinity will prompt for a filename and write the coordinates of the four corners of each SDF to an ASCII file. The SDF data may be visualised in EarthVision following post-processing using two awk scripts.

7. Theoretical Background

7.1 Chapter description

This chapter describes the technical details of FracAffinity Version 3.3. It is a companion chapter to the FracAffinity User Guide (Chapter 8) which provides information on the file formats and use of FracAffinity Version 3.3. FracAffinity enables the user to simulate three-dimensional flow in a heterogeneous, fractured rock. The main feature in FracAffinity is that it adopts a "hybrid medium" approach to the representation of fractured rock, which includes both permeable, intact rock and fractures. This approach has been successfully applied in a number of site investigations [19]. This is different from the more traditional porous medium or fracture network approaches which include only one component of a fractured rock.

The hybrid model is designed to overcome the well-known shortcomings of the porous medium and fracture network approaches. Firstly, the difficulties, for a pure porous medium approach, of representing the fast flow channels or preferential pathways due to fractures and faults. Secondly, the failure of a pure fracture network model to account for any flow in permeable parts of the intact rock in which the fractures are located, which means that the simulated flow and thus, the predictive capabilities, are critically dependent on the connectivity of the fracture network [1),2),3)].

The basic idea behind the hybrid medium approach is to represent a volume of rock as two main components: "discrete features" and "intact rock" (this is shown schematically in Fig. 7.1.1). Discrete features are objects such as faults or fractures that introduce linear/surface variations in the properties of the rock. A distinction is made between "deterministic discrete features" (DDFs), which are relatively large scale features whose geometry might be determined accurately by a regional geological investigation, and "stochastic discrete features" (SDFs) which are smaller scale features about which only partial information is available (e.g. from scarce outcrops, core logs or shaft and tunnel walls). The intact rock is the remaining rock which is either completely intact or contains only "micro-fractures".

An additional feature of the hybrid medium approach is that it incorporates spatial variability in the properties of the intact rock and discrete features, both as a result of stratigraphy and more local heterogeneity. The stratigraphy is accounted for by using the output from the three-dimensional geological modelling package EarthVision [4)] as the framework in which rock properties are distributed. The local heterogeneity is simulated using geostatistical and fractal methods to generate interpolations of borehole data and the larger scale heterogeneity by specifying the anisotropic scaling of the hydraulic conductivity.

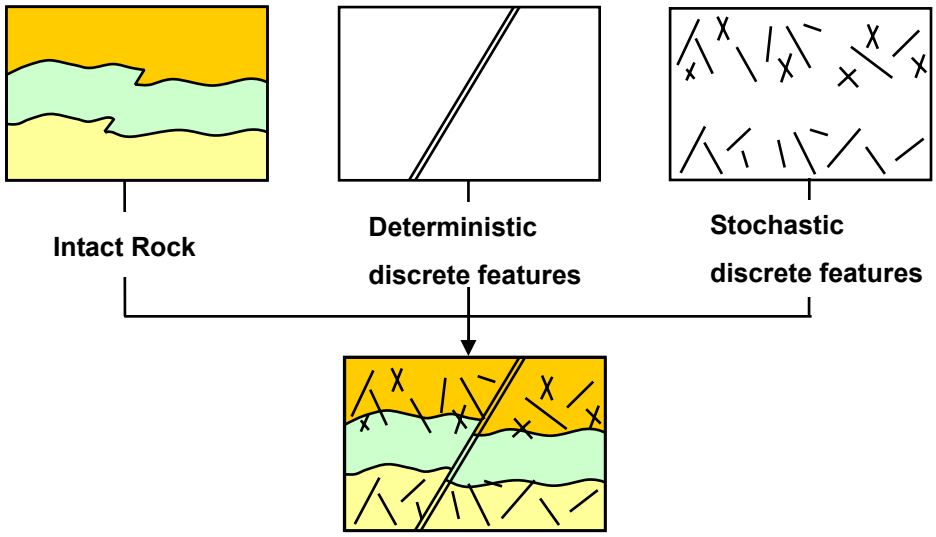


Fig. 7.1.1 Schematic of the constituent components of a hybrid medium: intact rock; deterministic discrete features; and stochastic discrete features.

Properties of the intact rock are most naturally thought of as being associated with a volume (a "cell"). The approach therefore adopted in FracAffinity is to populate the cells in a grid with flow properties (hydraulic conductivity, porosity and specific storage coefficient) which may be different in different stratigraphic zones. The rock properties may be uniform or spatially variable in different stratigraphic zones and may also be conditioned on borehole data.

Properties of the discrete features are more naturally associated with channels inside the features [20]. The approach adopted in FracAffinity is therefore to generate a network of flow channels. In the large-scale deterministic features this is undertaken indirectly, by first gridding the feature's surfaces, and then identifying the equivalent network. In the stochastic features, the flow pathways network is identified directly. Initially, the intact rock, DDFs and SDFs are handled separately. They must however be "merged" to form a hybrid medium. The approach adopted in FracAffinity is to convert the intact rock grid to an equivalent flow network, so that the representations of the three components of the hybrid medium are compatible. The three networks can then be joined, although particular care must be taken in handling the connectivity within and between the three networks, since the connectivity of pathways plays a key role in flow.

In addition to representing the structure of the rock mass, FracAffinity includes models for dynamically modified shafts, static vertical shafts or horizontal galleries and boreholes. All of these features are represented as cylinders. Techniques have been developed to link these features into the irregular model represented by the irregular hybrid medium network. These techniques aim to realistically simulate pressure gradients surrounding the engineered feature, particularly in the presence of local desaturation.

The final step in FracAffinity is to use the network representation of the hybrid medium as input to a finite-difference flow solver. FracAffinity solves for the head at each node by balancing the water flows along the connections with the storativity effects.
In this document, we refer to the head, h, and the pressure head, p, both being measured in metres of water. The pressure head is the pressure, P in Pascals, above atmospheric measured in metres of water. The head is this plus the z-coordinate and is sometimes called the static or piezometric head. So,

$$p = \frac{P}{\rho g}, \quad h = p + z \quad . \tag{7.1.1}$$

For a further discussion of these concepts, the reader is referred to a standard text book, such as de Marsily [17].
For a steady-state saturated calculation, this is simply a linear algebra problem - we use a Symmetric Preconditioned Conjugate Gradient Scheme. For transient calculations, we use a variable order backward difference scheme (essentially a Gear's method). For the unsaturated cases the same basic approaches are used, although the matrices are no longer symmetric which makes them harder to handle. The FracAffinity solver is capable of dealing with both fully saturated and partially saturated systems.
In Section 7.2 of this chapter we provide detailed information on the structure of the hybrid medium and the construction of the FracAffinity network which is used by the flow solver. Sections 7.3-7.5 describe in more detail the process of constructing intact rock, deterministic feature and stochastic feature rock models. Section 7.6 shows the approach that is used to model a shaft excavation by dynamically updating the FracAffinity network during the flow calculation and Section 7.7 describes the borehole model available in FracAffinity. In Sections 7.8 and 7.9 the flow calculations and solution method are discussed for both saturated and unsaturated systems.

7.2 The Hybrid Medium

This section describes the concepts behind the hybrid medium approach. We start by describing the properties of the networks: how are they constructed and linked among different parts of the domain, and finish showing how they are combined to produce a final domain.

7.2.1 Overview of the Hybrid Medium

The algorithms used for gridding components of FracAffinity (Intact Rock, DDFs and SDFs) and combining these grids into a network include the following steps:

- Assigning network properties.
- Generating a network for Intact Rock.
- Generating networks for Discrete Features.
- Combining the networks.
- Attaching (associating) boundary conditions.
- Eliminating disconnected features.

FracAffinity uses a network approach to simulate groundwater flow in the hybrid medium. For the intact rock a rectilinear grid is generated and the approach is essentially equivalent to finite difference. The generation of the hybrid medium occurs in a stepwise manner with features being added one at a time. Firstly, the intact rock is generated and then each discrete feature, beginning with the DDFs, is added one at a time. As the discrete features are added the network is modified by re-routing connections via nodes on the feature being introduced. Following the addition of the discrete features the engineering features are added to the network. However, prior to the construction of the hybrid medium some pre-processing is undertaken, for example to calculate the intersections between discrete features (if requested by the user) and to determine the gridding of the engineered features.

The general scheme for generating the hybrid medium, outlined above, is described in more detail in this section, whilst the precise algorithms for generating the intact rock, DDFs, SDFs, shaft models and boreholes are described in the subsequent sections.

7.2.2 Flow Network Properties

The network that is used in FracAffinity is made up of legs (referred to as connections internally in FracAffinity) and nodes. There are some special “boundary nodes”, but these do not concern us yet. Each connection connects precisely two nodes. A node can be at the end of many connections (there is no maximum number of connections per node). Note that it is possible for there to be two connections between the same pair of nodes. This causes no difficulties for the solver, and it is far more simple to leave both in than to try to combine them.

FracAffinity stores the physical properties needed to define the system by associating them with either connections or nodes. The assignment of these properties occurs during the grid generation phase and is discussed in the relevant section below. Here, the properties themselves are listed and explained.

Fig. 7.2.1 summarises the properties, showing those that are associated with nodes, and those associated with connections. Each connection (including connections to the boundary nodes) is divided into two parts, indicated by the thick cross line in Fig. 7.2.1.

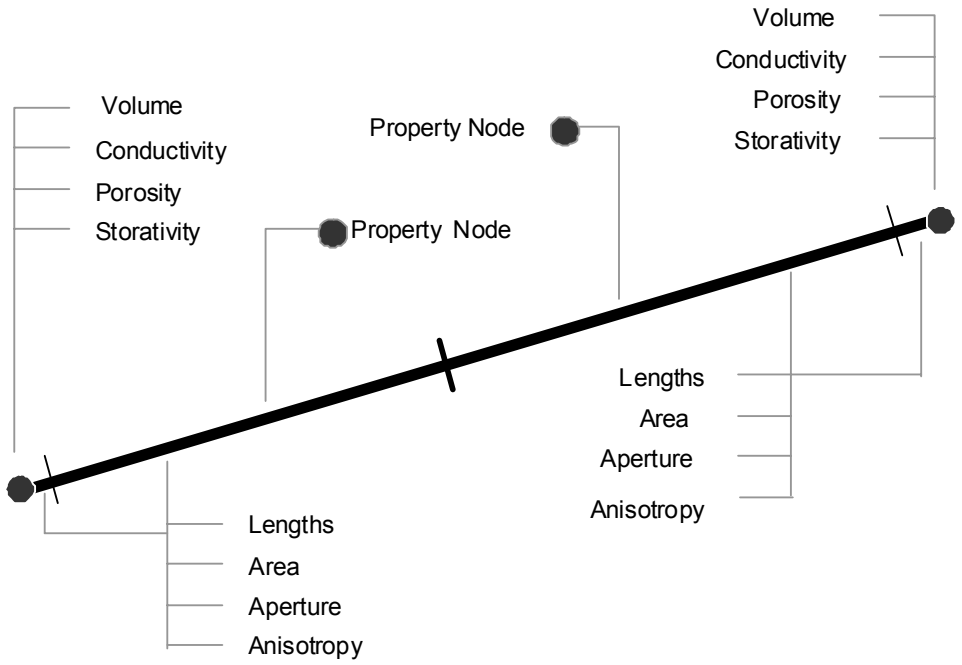


Fig. 7.2.1 The properties assigned to network nodes and connections in FracAffinity.

These are referred to as the forward and backward connections. Splitting the connections in this way enables the correct averaging of conductivities to be made. Further reasons for breaking connections down will become apparent when we discuss the way networks are merged as discrete features are added. The break into the two parts is directly related to the volume that each node is associated with. If we view each node as being at a cell centre, then the break occurs at the cell boundary.

The data stored for the nodes and connections is the same for Intact Rock and Discrete Features except for the aperture, which is irrelevant for Intact Rock connections.

Each half-connection is potentially further divided into two – a section near to the end node (the near-section) and a section not containing the end node (the far-section). This split is crucial in maintaining the correct properties for intact rock and discrete features (Fig. 7.2.1), as we shall see later. When the networks are first created (as a purely intact rock or for a single discrete feature) the split is not used (the near-section is given zero length).

Properties on connections and nodes can be split into two types: geometric and physical. Most of the geometric properties are stored with the connections and the physical properties are stored with the nodes. The exception to this is the volume, which is a nodal property.

Fig. 7.2.2 shows the geometric data for the intact rock and a discrete feature.

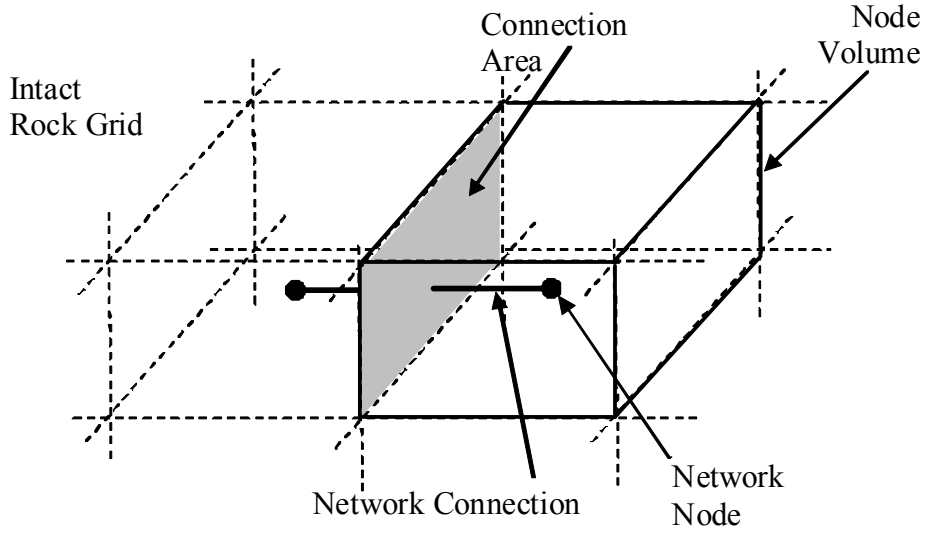


Fig. 7.2.2 The geometric properties for the Intact Rock.

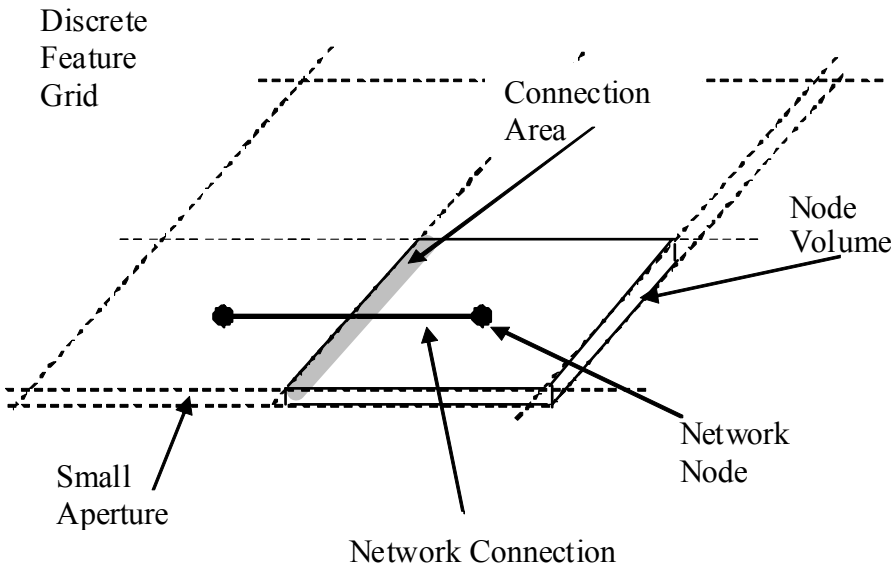


Fig. 7.2.3 The geometric properties for a Discrete Feature.

We start with the connection data. Each section (near and far) has a length. For the far-section, this is the geometric distance to the middle of the connection. The length of the near-section is always small – either zero or half of a discrete feature aperture (if the connection is from a discrete feature node), when a connection is created the near-section length is zero.

The near- and far-section have the same area (and aperture for discrete features). The area is the cross-sectional area for the connection (area of the face between cells) and is set by the geometry of the grid. The area will be that of the cell that contains the node in the direction orthogonal to the connection. For discrete features, the area will be the aperture of the feature multiplied by the width of the connection.

The aperture is only used for discrete features – it is just the aperture of the feature. It is only used for output calculations (pathlines and particle tracks). The final item of "data" on a half-connection is a property node. This is a link to say which nodal property should be used for the physical properties of the far-section of the connection. Initially, this will always be the node at the end of the half-connection, but this can change as networks are merged – see the later discussion.

Each part of each leg stores the anisotropy factor that applies for the relevant direction. For discrete features, these will be the in-plane values. For intact rock, they will be the x-, y- or z-anisotropy. After an intersection with a discrete feature, the near-section anisotropy will have the normal-direction anisotropy of the discrete feature.

The data associated with each node is set when the node is created and is not changed as networks are merged. The volume is the cell volume of the grid used to create the network. For a discrete feature, the volume is the aperture times the area on the grid cell. Note that the volume of intact rock nodes may include some volume that is actually a discrete feature. The assumption that discrete features have small aperture means that this double counting is not important. For boreholes, the intact rock volume is reassigned when new nodes are added – see Section 7.8.

The hydraulic conductivity, porosity and storativity are all self-explanatory. These record the local value, and so will vary from node to node where there is heterogeneity.

Flow calculations are made based on a mass balance approach. Ultimately, they rely on Darcy's law. In flow calculations, the important property for each connection is its overall conductance. Conductance [m^2/s] multiplied by head difference [m] gives flow [m^3/s]. Conductance is basically Area × Hydraulic Conductivity/Length. Note that the conductance is purely a derived quantity. Although it has the same units as transmissivity, the latter arises by integrating the hydraulic conductivity across a length (typically aquifer thickness or length of a borehole section). For the multi-part connections we have to combine the individual hydraulic properties. The resistance (inverse conductance) of a multi-part connection is the sum of the individual resistances, and hence the overall conductance, C, can be calculated from

$$\frac{1}{C} = \sum_i \frac{l_i}{A_i K_i}, \tag{7.2.1}$$

where l, A and K are the length, area and hydraulic conductivity of the separate parts of the connection (four in all). Notice that setting the length of any part to zero does not invalidate this formula (even though that part has infinite conductance).

Another use for the connection properties is to calculate the travel-time. The travel time down a connection is basically given by Length/Velocity, which is equal to Length×Area×Porosity/Flow. The flow is constant along the connection, and so the travel time is given by

$$\tau = \frac{\sum_i l_i A_i \theta_i}{Q}, \tag{7.2.2}$$

where l, A and θ are the length, area and porosity of the separate parts of the connection and Q is the flow rate along the leg.

In the two previous formulae, the nodal properties used (porosity and hydraulic conductivity) in each part of the connection are those of the end node (for the near-sections) and property node (for the far-sections).

7.2.3 Generating the Flow Networks

FracAffinity creates separate grids for the intact rock and for each of the discrete features. These are then merged into the final hybrid network. In this section the gridding of the individual components is explained briefly, with more details provided in the following sections.

(1) Intact Rock

The intact rock grid is initially rectangular in the x-y plane with regions outside the model domain removed. The vertical structure is determined from the top and bottom surfaces and a user-defined refinement surface. This is described in more detail in Section 7.3.

A grid of cells with rectangular cross-section but varying vertical extent is created. Any cell that is completely outside the model region is omitted. Then a network is created by placing a node at each cell centre and creating connections between neighbours. Where a cell has no neighbour, a boundary node is added and a connection made to this.

If local refinement is used then the network has more than one node in a cell.

Additional nodes are then added and connected to handle any thin geological layers. This is described in more detail in Section 7.3.

At this stage, the outer shape of the model region is only crudely followed by the network. There may be nodes outside the model region and hence there will be full connections that cross the boundary as well as the connections to boundary nodes that are expected to cross the boundary. The final tidying up of the network, including placing the boundary nodes precisely on the boundary, is the final stage of the process once individual networks have been merged. This is discussed later together with the application of boundary conditions and removal of disconnected parts of the system. (Sections 7.2.5 & 7.2.6).

(2) Discrete Features

Discrete features are two-dimensional, so the network that is generated for them is also two-dimensional. For DDFs, the scheme uses an intermediate grid, in a similar fashion to the intact rock. Although DDFs may not necessarily be planar, they are defined in terms of an x-y grid, so this causes no special difficulties.

For SDFs, the network is created directly (there is no intermediate grid of cells created).

Deterministic Discrete Features

The network produced for each DDF is the two-dimensional equivalent of that produced for the intact rock. The DDFs may be limited within the hybrid medium by use of a 2D polygon as described in Section 7.4. A rectangular grid of cells is created, with, in cases where a polygon has been defined, cells omitted if any part of the cell lies outside the polygon which defines the limits of the fault. Nodes are placed at cell centres and neighbours are joined by connections. Boundary nodes are created where no neighbouring cell exists.

Where a DDF intersects the centre-line of a shaft or gallery and none of the DDF nodes are within the feature, the nearest node on the DDF is moved onto the centre-line. This ensures the boundary conditions for the shafts and galleries are applied to DDFs. In the same way as the intact rock, the final network does not perfectly match the model region boundaries, and this is also sorted out later (Section 7.2.5).

Stochastic Discrete Features

Each SDF is planar. As the SDFs are generated, those that are completely outside the model region are discarded. Features are "cropped" (that is reduced in size) by a bounding cuboid, so that networks are not generated for large parts of features that are outside the model region. The feature for which a network is generated is therefore a polygon (it will be a rectangle if no cropping was necessary). A network is directly generated on this polygon. For SDFs there is no intermediate grid of cells because the properties are all uniform. The length of the connections in the SDF network are controlled by the GRID_SIZE parameter. A two-dimensional coordinate system in the plane of the fracture is used.

Fig. 7.2.4 shows an example of the network that is produced for a single SDF. Note that lines of nodes are introduced along the edges of the feature. This is in contrast to the DDF network which has boundary nodes on the edge. The different approach is because DDFs are expected to be large feature whereas SDFs are expected to be small. Connections are omitted if both nodes are outside the feature. Nodes are omitted if there are no connections to them.

Where an SDF intersects the centre-line of a shaft or gallery, the nearest node on the SDF is moved onto the centre-line. This ensures the boundary conditions for the shafts and galleries are applied to SDFs.

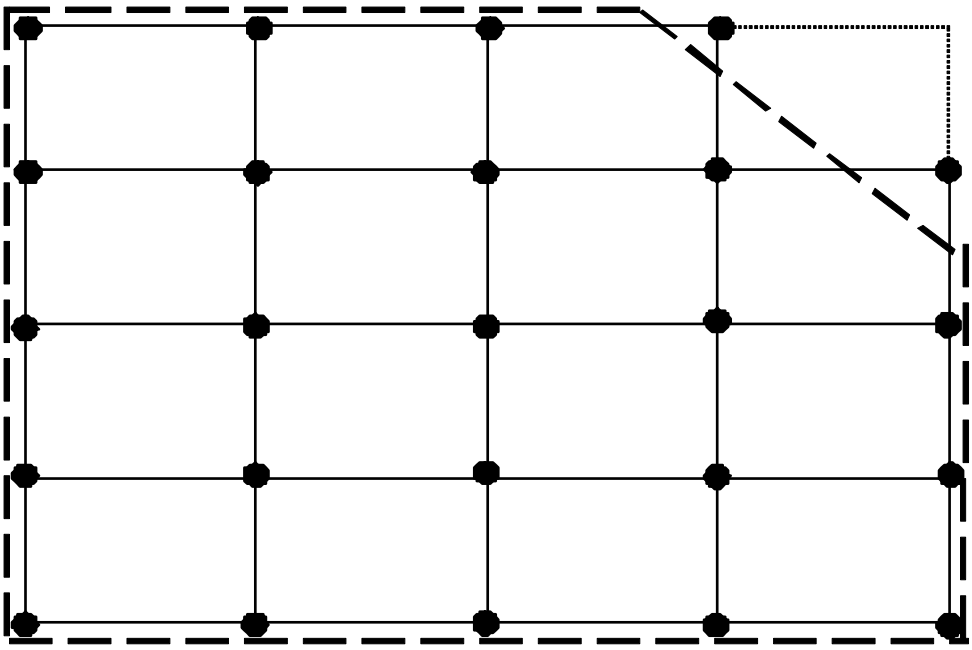


Fig. 7.2.4 Networks for an SDF. The dashed line shows the boundary of the “cropped” feature.

Boreholes

Boreholes, actually borehole sections (BHSs), are modelled as one-dimensional features. Each BHS is discretised with at least one node per intact grid cell and additional nodes where discrete features intersect. Nodes on the BHS are linked to nodes in the intact rock and discrete features, as described in detail in Section 7.8. The BHS nodes are considered to be part of the intact rock network, except that the connections between them (representing the borehole) are not cut by discrete features. Instead, they are linked to the discrete features by new connections.

7.2.4 Combining the Flow Networks

The final network that is used in FracAffinity, the Hybrid Medium Network, for solving the flow problem is created by merging the networks which form the Intact Rock, DDFs and SDFs and boreholes and then sorting out all the boundary conditions. This section discusses how the merging is done.

Merging is done one feature at a time. That is, we start with the Intact Rock network and add the DDF networks one by one. Then the SDF networks are added one by one. There is no creation of a combined network for all DDFs or SDFs.

Thus, the basic step is adding a two-dimensional network into an existing three-dimensional network. As the merging progresses, the three-dimensional network becomes increasingly unstructured, and so it is managed in the code it as a completely general network.

The steps in the merging algorithm are as follows:

1. Find all connections in the Three-D network that cut the surface that the Two-D network is constructed within.

2. For each cutting connection, find the nearest node on the Two-D network to the point of intersection.
3. Rewire the cutting connection to join up with the nearest node.
4. Modify the connection data to reflect the changes.

It is difficult to draw pictures of this in Three-D, but the algorithm is clear from considering a Two-D network to which One-D features are added, as shown in Fig. 7.2.5.

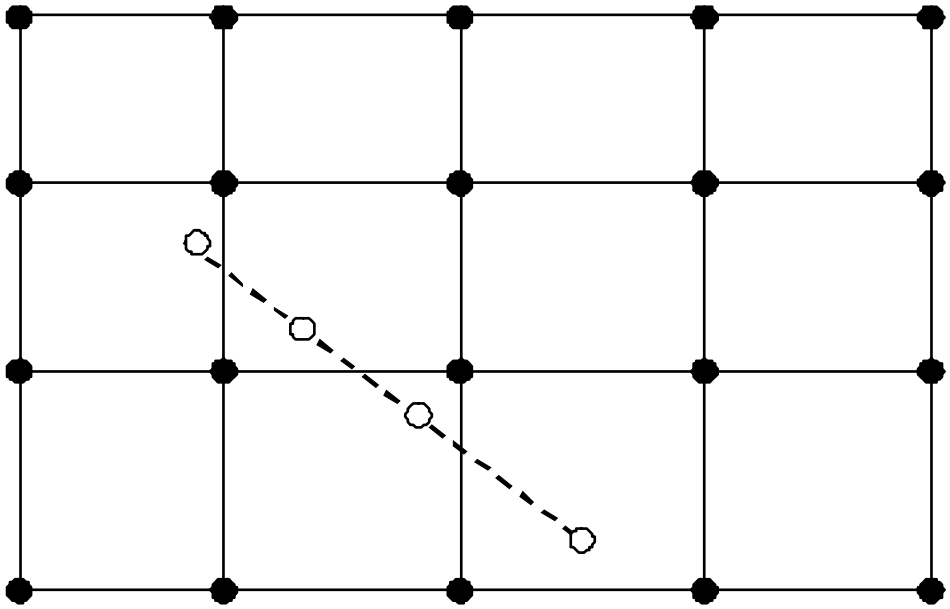


Fig. 7.2.5 Two-D Network with One-D feature to be merged.

There are three connections in the Two-D network that cross the One-D feature. Each of these is rewired as shown in Fig. 7.2.6. Notice that none of the nodes are moved. Each of the connections that cross the new feature is split in two and re-routed via the nearest node.

The final step in the merging process is to modify the data (and create new data) for the re-routed connections. Fig. 7.2.7 shows a single connection intersecting the added feature. Note that the intersection is in the half connection with Properties B.

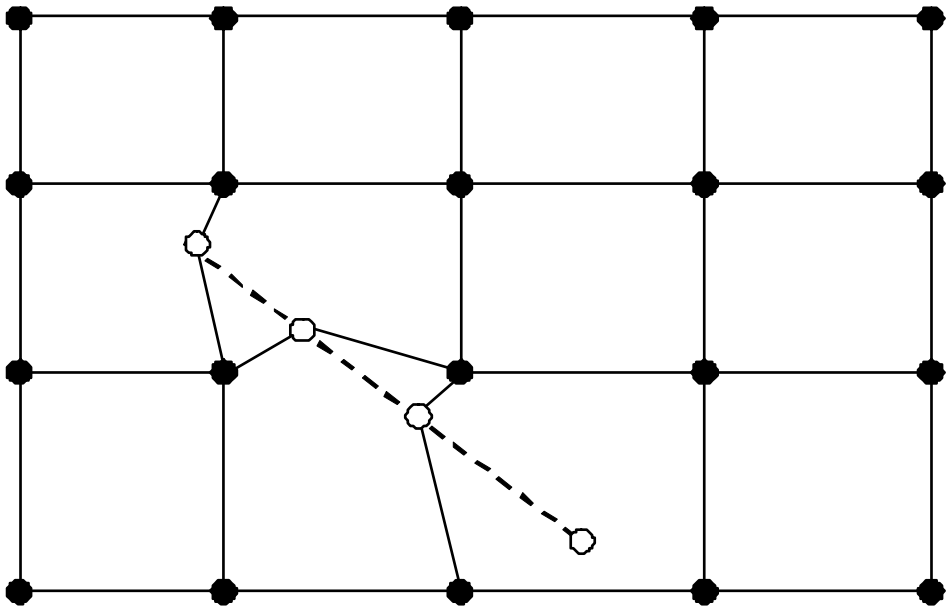


Fig. 7.2.6 Two-D Network after merging the One-D feature.

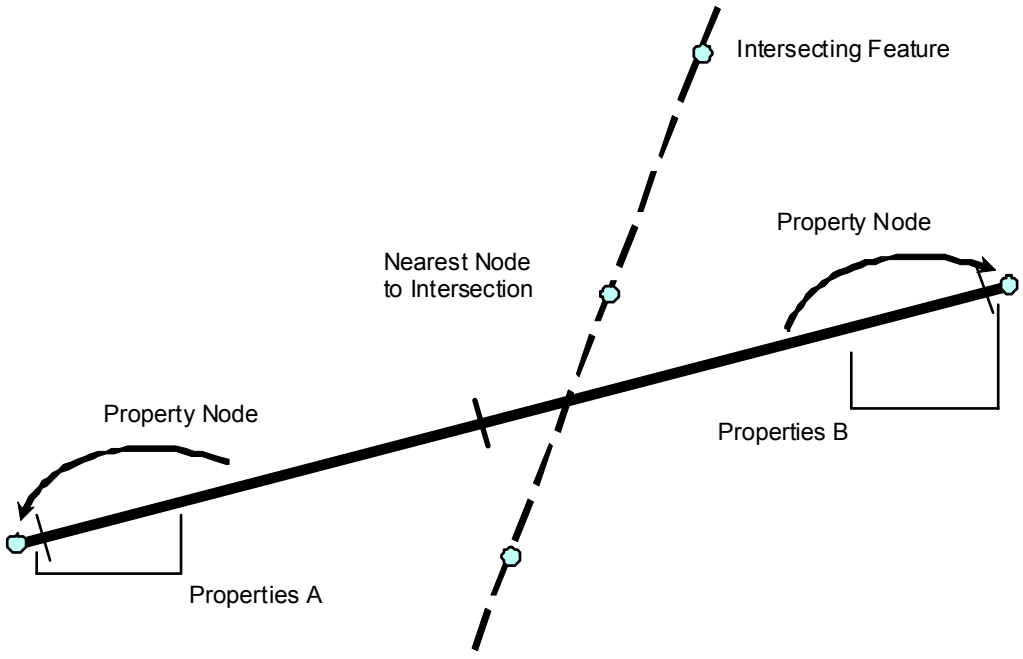


Fig. 7.2.7 Connection properties before merging.

Fig. 7.2.8 shows the situation after merging. The properties (representing area and aperture) are taken from the relevant half of the prior connection and the appropriate length calculated. More important is the way that the property node is assigned. The reasons for this become

clear if we imagine that the intersecting feature has very different physical properties from the pre-existing connection.

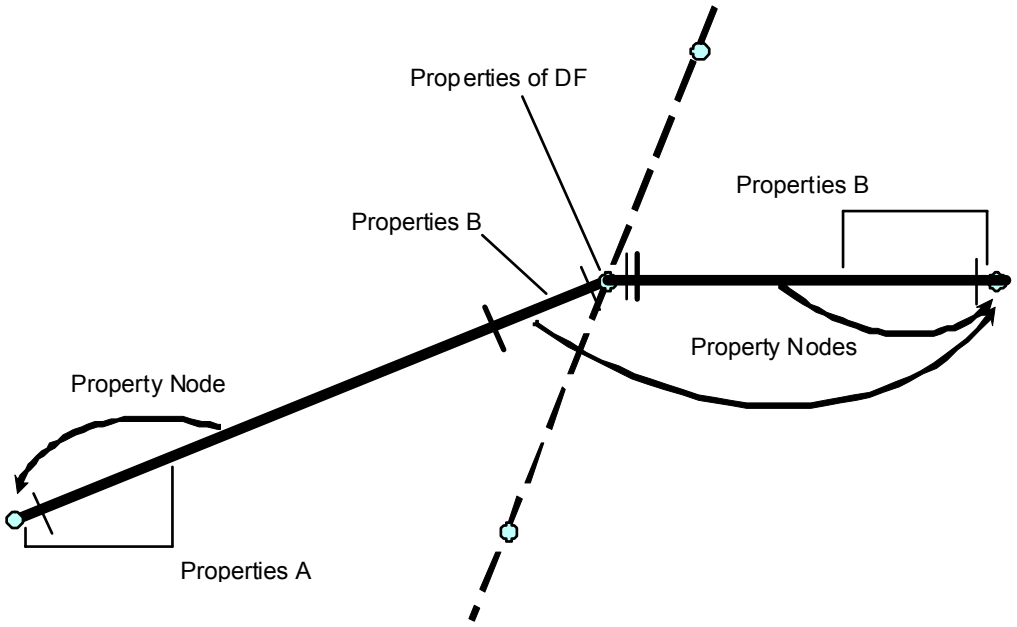


Fig. 7.2.8 Connection properties after merging.

Imagine that the intersecting feature is a high-conductivity fracture and the existing connection is from a low conductivity intact rock. Then, we want the two new connections to largely have the intact rock properties – the only part of them that is in the high-conductivity feature is the tiny section close to the feature (half of its aperture each side). To achieve this we use the old properties from the old property nodes for the far-sections of each connection (recall that the near-section always uses the properties of the node it is attached to). Without this approach, a major part of the old connection would become highly conductive, misrepresenting the true situation.

Because each new feature affects the connections that are already there, and cuts across them, the network depends on the order. In particular, features added later are more likely to form uninterrupted features in the final network. Thus, a low-conductivity feature added at the end will block flow. If it is added at the beginning it may be breached by later features. This capability can be very useful to control the properties of the FracAffinity network., but must be applied with reference to the known geological environment, or to investigate a particular conceptual model. An exception to this is the borehole section, which is added to the intact rock model but is not cut by discrete features.

7.2.5 Attaching Boundary Conditions

Assigning the correct boundary condition values is straightforward once the boundary nodes have been correctly positioned.

A boundary node is required wherever a connection crosses the model region boundary. Thus, we must first determine which connections do cross the boundary, and then replace the connection with a connection to a boundary node. In some cases the connection will already end in a boundary node, in which case it is simply moved; in others, a boundary node must be created, a new connection made and the old connection deleted.
Determining which connections cross the boundary is done by first determining which nodes are inside and which are outside the model region. Then all connections are checked. The location of the boundary node is found by repeated bisection of the crossing connection. In this process, the EarthVision surface grids are used with bilinear interpolation within each EarthVision grid cell. This gives a continuous surface that has the required z-coordinate at every EarthVision grid point.
As the old connections are deleted there will be parts of the network that become disconnected – these are handled as described in the next section.
Boundary nodes that are inside the model region are discarded – these arise on the edge of DDFs that do not cross the whole region.
Note that the algorithm described above applies to all connections in the network, regardless of their origin. Thus, discrete features are properly linked to the boundary as well as intact rock connections.

7.2.6 Removing Disconnected Parts of the Network

There are at least two ways that parts of the network can be disconnected. Firstly, they may be just outside the model region and become cut off when boundary nodes are derived. Secondly, the may have come from discrete features that failed to connect with anything else.
Note that we do not want to remove dead-ends (a dead-end is a node which is connected to the network by only one connection) – even if they take no flow in steady state, they can have a relevance to transient behaviour because of their storativity.
For the current purposes, a disconnected node is one that is not joined through network connections to boundary nodes.
Thus, the algorithm used marks all boundary nodes as wanted, and recursively extends the list of wanted nodes by adding the neighbours (via a connection). When no new nodes are added to the wanted list the recursion stops. Any nodes not on the wanted list at this stage are discarded (and connections between them also).
Note that the use of property nodes as described earlier complicates the picture somewhat – any node used as a property node cannot be deleted, instead it is moved onto a separate list so that it is not included in the calculations.
For the purposes of the solver, any sections of the networks that are only connected to flux-conditions at boundaries must be dealt with as they will otherwise lead to a singular (unsolvable) system. These are detected in a similar way to that described above. The nodes are not deleted, rather the boundary conditions are changed to zero head. This will not affect the behaviour in the rest of the system, but may show up in visualisation.

7.3 The Intact Rock

The intact rock model is intended to represent the porous rock within the geological model. The intact rock geometry and properties are closely related to the geological structure, referred to as the geological framework, exported from EarthVision. The geological framework exported from EarthVision also includes the information used to define DDFs, commonly faults, and the information on the formations is separated into the constituent fault-blocks. The stratigraphic formations are represented in the intact rock model and physical properties are associated with each stratigraphic formation.

The concepts of faults, fault-block and stratigraphic boundaries are illustrated in Fig. 7.3.1a and 7.3.1b, below. For a more detailed description of these concepts, the user should refer to the EarthVision User Guide[4].

In defining the FracAffinity model, the user specifies a Model Region, which provides the boundaries of the flow problem. The Model Region may be the same as the geological framework or may be smaller. It is a polygonally based region with vertical sides and is bounded above and below by user defined surfaces with variable elevation (see Fig. 7.3.1c).

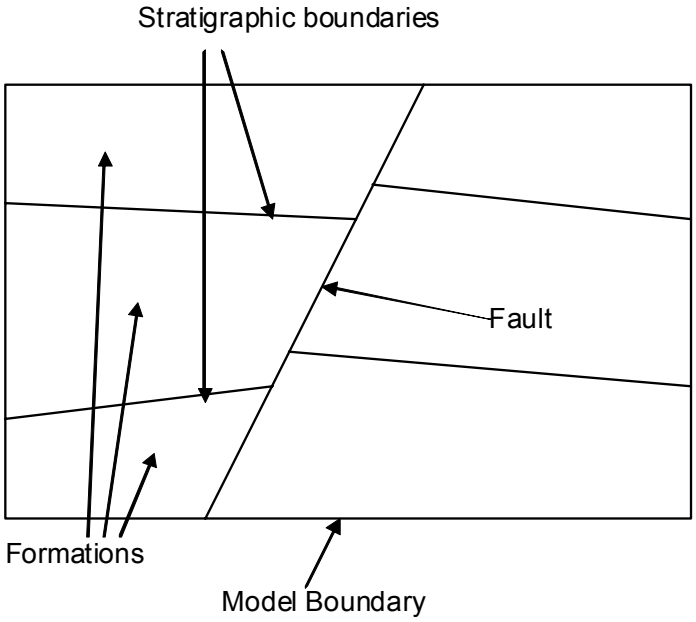


Fig. 7.3.1a Simplified Two-D representation of the stratigraphic information exported from EarthVision.

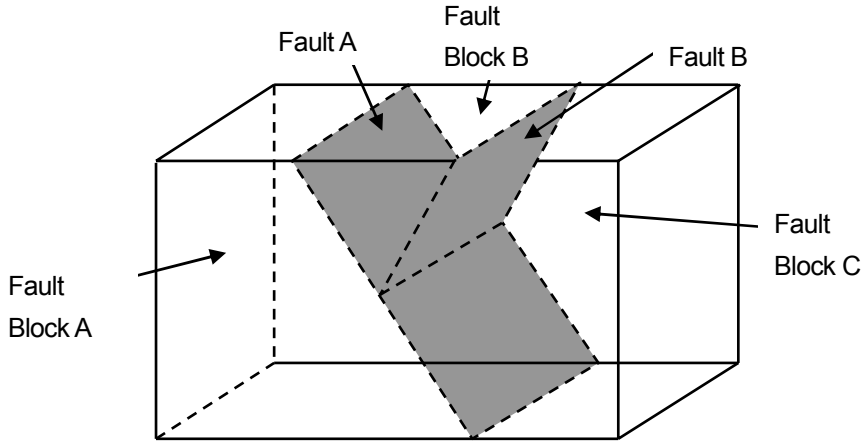


Fig. 7.3.1b Simplified fault block decomposition of a model region.

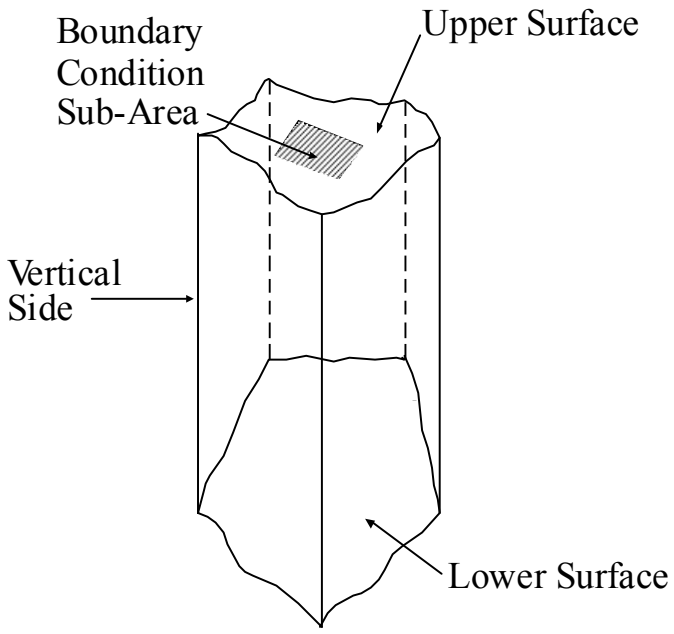


Fig. 7.3.1c The FracAffinity model region defines the boundaries of the flow problem. It is a polygonally based region with vertical sides and is bounded above and below by user defined surfaces with variable elevation.

The generation of the intact rock model consists of four phases:

1. Gridding of the model region to produce cells.
2. Conversion to a FracAffinity network.
3. Additional nodes to handle thin layers.
4. Assignment of physical properties to nodes

These stages are described in the remainder of this section.

7.3.1 Gridding the Intact Rock

The approach taken to gridding the intact rock was substantially modified in Version 3.0. The intact rock grid was previously rectilinear in three dimensions – with all cell faces exactly horizontal or vertical. Now, the grid has the same connectivity, but the top and bottom faces of each cell are no longer necessarily horizontal; the side faces remain vertical. This change was motivated by a need to have a better representation of the near-surface, particularly near a water table, without introducing very large numbers of cells. This change, couple with the new approach to handling thin layers, is intended to significantly reduce the number of nodes needed for a typical FracAffinity calculation.

The intact rock grid is generated in two stages. First, a rectilinear x-y grid is created. Subsequently, the top and bottom locations for each cell are determined at each of the x-y corners of the cells. The same number of cells in the vertical dimension is defined at each location, leading to a grid with non-horizontal surfaces but with a simple connectivity structure.

The two stages in creating the grid are described in the following sub-sections.

(1) The x-y grid

The x-y aspect of the intact rock grid generated by FracAffinity is rectilinear, with each vertical cell face being a parallelogram parallel to the x and y axes. The cells are defined by setting the cell boundaries on the x and y axes using three different stages. First, a series of control points are constructed from the geometric structures that define the model boundaries (Fig. 7.3.2). The control points define the corner points of cells. The location of the control points is determined from the vertices of the model region and from any shafts or excavations in the model.

Note that in earlier versions of FracAffinity, additional control points were added based on the locations of the maximum and minimum elevation of each of the stratigraphic surfaces defined by EarthVision. This was an attempt to handle complex stratigraphies. It generally resulted in large members of small cells but did not solve the basic problem of modelling thin layers. FracAffinity now includes specific algorithms to handle the thin layers directly (Section 7.3.3).

In the second stage, a user defined parameter “min_num” or minimum number is used to specify how many additional control points, should be placed between each pair of the automatic control points. These control points are equally spaced. To ensure that the number of control points is not excessive, the minimum distance between them is controlled by a user-specified resolution.

Finally, the user defines a regular spacing in each of the x and y directions (deltaX and deltaY) and further control points are added to each axis at these spacings, again subject to the minimum size implied by the resolution. The user may also specify points that they wish to force cell centres to match – these are managed by adjusting cell boundaries where necessary.

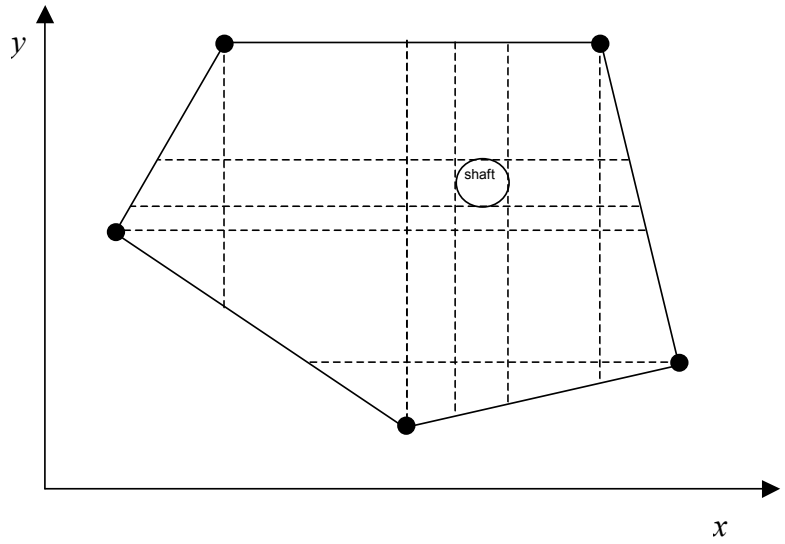


Fig. 7.3.2 Controls points on each axis for gridding the stratigraphy are obtained from the model boundaries, including internal boundaries such as a shaft.

The grid is then generated using the control points to specify the cell face boundaries.
The algorithm is repeated for sub-regions of the model region which the user may define. Here, the user may specify a sub-region within which the parameters deltaX and deltaY[1] and the resolution may be redefined. The effect of defining a sub-region with finer gridding is shown in Fig. 7.3.3. Sub-regions are most likely to be useful for increasing the model resolution near engineered features, such as a shaft excavation.
This sub-region refinement algorithm is limited to some extent by the fact that the grid is rectilinear. This means that the grid struggles to represent very narrow stratigraphic layers, whose surfaces vary significantly in elevation. If the variation in surface elevation is greater than the thickness of the layer then the grid may be unable to maintain the connectivity of the formation. This may be important where the layer is highly conductive or represents a significant barrier to flow. If this is case, it may be necessary to model the region as an equivalent deterministic fault. The grid for the fault will maintain the required connectivity since it is gridded independently (see the next section).
A second type of grid refinement, known as local refinement is also available. Local refinement specifies a cuboid in which a refined grid spacing is defined. This results in an unequal number of nodes in each cell, and is therefore generally more useful than sub-region refinement. The gridding of these structures is described in Section 7.3.2(2).

[1] deltaZ is also specified by the user and applied during gridding in the z-direction (Section 7.3.1(2)).

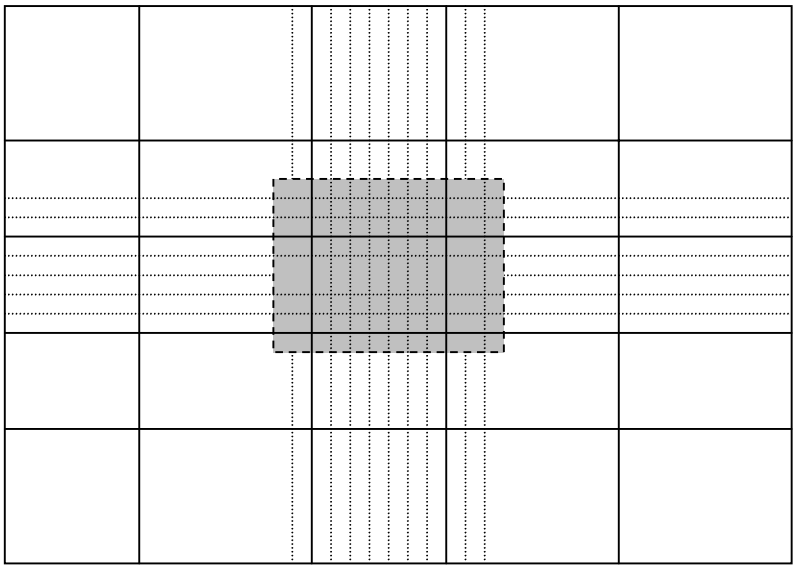


Fig. 7.3.3 A simple grid with a refined sub-region (greyed area) and the additional grid lines generated (dash lines). Note that additional cells are generated outside the sub-region as well as inside, since the grid is rectilinear, although this does not occur when local refinement is used.

(2) The z grid

Gridding in the vertical direction is controlled by the positions of the model's top and bottom surfaces and by various user-defined properties, in particular the location of an approximate water table surface. The number of cells generated in the vertical is constant across the region and so the gridding is fully defined by the cell boundaries at each x-y location. This is most simply described by a diagram. Fig. 7.3.4 shows the cells for a single location. Note that some cells may subsequently be discarded because they are outside the model region, so that the number of surviving cells in the vertical is not necessarily constant.

There are three distinct regions. Starting from the top surface there is a region down to the user-defined "water-table" surface. In this region, the cells get smaller as the water-table surface is approached. The user specifies the number of cells and the ratio of the sizes in adjacent cells. The user also specifies a minimum "respect" distance. If the top surface is less than this distance above the water table then the gridding starts from the respect distance above the water-table. Any cells above the surface are then discarded.

The second region is a layer with a user-defined thickness below the water table. This has a fixed number of equal-sized cells. It is intended that this region is used to model drawdown of the water table efficiently

Finally, the remaining model volume is gridded. Here the cells are equal-sized. The number of cells in this region is determined from the delataZ grid parameter. This is used in combination within the largest extent of the region to give a number of cells such that no cell exceeds the deltaZ size.

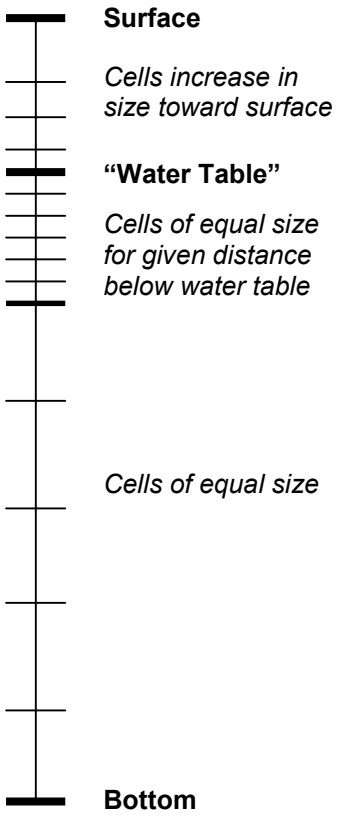


Fig. 7.3.4 Cell definition in the vertical direction.

7.3.2 Constructing the Initial Intact Rock Model

Once the grid has been constructed as described in the previous section, then the grid is converted to a network ready for construction of the hybrid medium.
First, the approach when there is no local refinement is discussed. Then, the more complicated situation when the cells are locally refined is described.

(1) Unrefined Cells

The conversion to the network is achieved by defining network nodes at the centroids of the grid cells and connecting the nodes for neighbouring cells (Fig. 7.3.5). Those cells which are next to the boundary of the model region will have no neighbouring cell for the cell face adjoining the boundary and in these cases a boundary node is defined at the centre of this face. The node at the cell centre is then joined to this boundary node.
The properties of the cell are assigned to the nodes and cell connections as described in Section 7.2. Thus, the volume of the cell and its hydraulic conductivity, porosity and storage coefficient are assigned to the node. The cell face area becomes the connection cross-sectional area and the distance from the separation between cell centroids becomes the connection length.

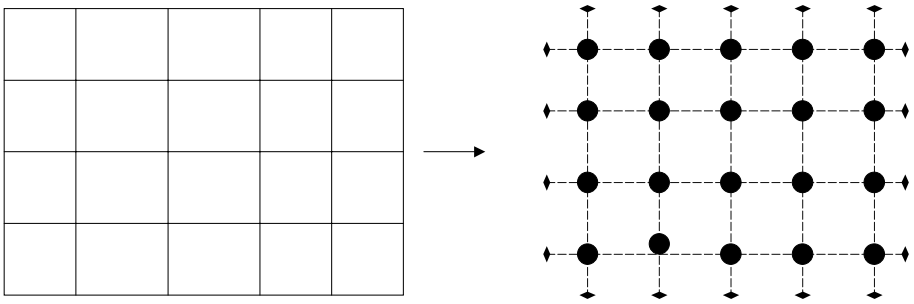


Fig. 7.3.5 An example (in Two-D) of an intact rock grid converted to a FracAffinity network. Boundary nodes are represented by diamonds.

(2) Locally Refined Cells

When a local refinement scheme is used, the affected cells will have a network grid spacing that is less than the cell size. They therefore generate a sub-network within the cell.

In each direction, the number of network nodes is chosen to satisfy the user-specified network size. Nodes are distributed evenly about the centre.

If one node is needed it will be at the ½ distance. Two nodes will be at ¼ and ¾, to give equal sizes when several cells have the same refinement.

Since the nodal properties are set later, each node can have different hydraulic conductivities.

When neighbouring cells have the same number of nodes in a direction, the linkage works on a simple one-to-one basis. Where the number of nodes varies, the larger number determines how many connections there will be (Fig. 7.3.6).

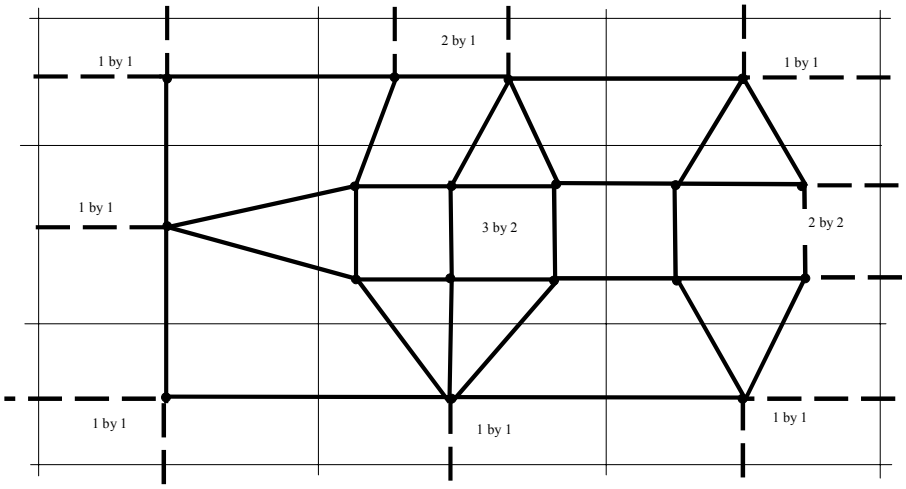


Fig. 7.3.6 An example (in Two-D) of connections between cells with different local refinement.

7.3.3 Handling Thin Layers

Special algorithms for handling thin layers were introduced in Version 3.0. They are described in this section. The user selects which layers are considered to be thin.

The objective of the special treatment of thin layers is twofold. First, to prevent any legs crossing over thin layers, since this has the potential to bypass a low conductivity layer. Second, to ensure that there are sufficient legs within a thin layer to allow flow within it, since this is necessary to correctly model high conductivity layers. Both objectives are treated together and the code makes no judgement on what constitutes low or high conductivity.
The basic approach is simply described. Starting with a network that has taken no special notice of the thin layers:

1. Identify legs that cross a thin layer.
2. Add new nodes within the thin layers and set suitable properties.
3. Add new legs joining nodes in the thin layer and set suitable properties.

The following subsections describe these steps in detail.

(1) Finding crossing legs and adding nodes

Overview

The identification of legs crossing a thin layer is relatively simple, but requires an efficient and unambiguous identification of which points lie in which stratigraphic zones (a stratigraphic zone is a stratigraphic layer confined to a particular fault block).
Given this identification, finding crossing legs is a simple matter. Any leg which crosses more than one stratigraphic boundary between zones in the stratigraphic sequence is a candidate for crossing. It's precise crossing points for the top and bottom of the zone are found. If these are where the zone is active (i.e. the zone is present) there will be a positive thickness. This thickness is then used in setting the volume of a new node centred within the thin layer. The volume of the node at the nearest end of the crossing leg is adjusted to preserve the total volume; both nodes may be adjusted if the new node is near the centre of the old leg.
Adding intermediate nodes deals with the case of low-conductivity thin layers, since it prevents any path crossing the thin layer without using the low conductivity properties. However, to handle high-conductivity thin layers it is also necessary to connect the nodes within the layer. This is discussed in Section 7.3.3(2). First, we describe the identification of which nodes are above, which are inside, and which are below the thin layer.

Relationship between nodes and the stratigraphic sequence

The key part of the algorithm is finding which nodes are above, which are inside, and which are below the thin layer. To this end, the top and bottom surfaces of each zone are calculated (Fig. 7.3.7). This is done on a common grid to avoid problems with conflicting interpolations.

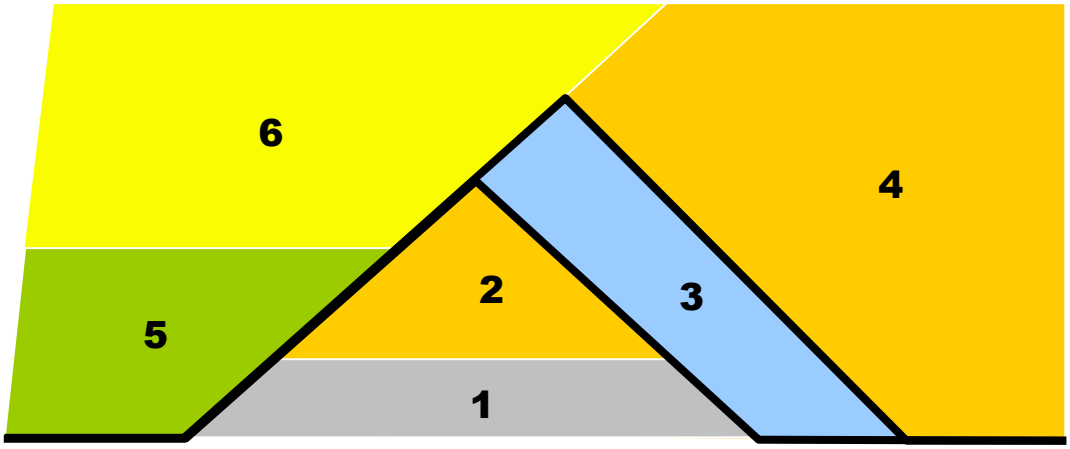


Fig. 7.3.7 Identification of top and bottom surfaces for zone 3, in a hypothetical model cross-section. The top and bottom surfaces are illustrated by thick black lines. Note that the two surfaces merge where the zone is not present.

Prior to the development of Version 3.0, the stratigraphic association of a node was undertaken using a linear interpolation of top and bottom surfaces. However, with such a linear interpolation sharp changes in the gradient of the surfaces resulted in smoothing of corners (Fig. 7.3.8). In order to recognise the sharp changes in gradient a new surface interpolation approach was implemented in Version 3.0. The interpolation used both the elevation at the grid points and its gradient. A piecewise linear interpolation was used to determine intersections – with the position where the gradient planes intersect defining the location of the change in gradient of resulting surface.

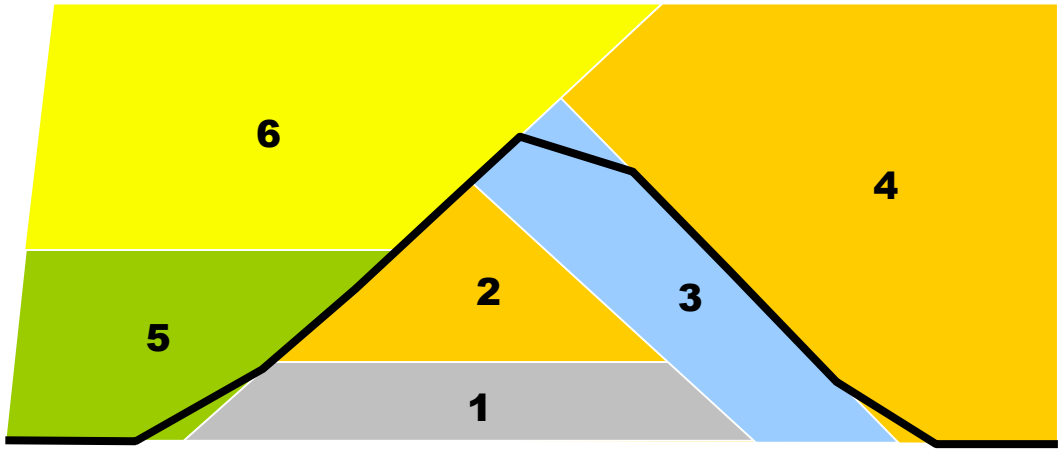


Fig. 7.3.8 Inappropriate interpolation for the top surface of zone 3 using linear interpolation. The surface grid spacing is exaggerated to emphasise the effect.

In Version 3.2, the approach was revised again, following the discovery of cases with very steeply dipping zone boundaries where the algorithm failed. The approach that is now taken is to store a list of the surfaces that jointly define the top and bottom surfaces. These lists are then used to determine the precise location of the top and bottom surfaces at any given (x,y) location. This approach directly relates to the specification of the stratigraphy and so gives precise results in all cases. It is somewhat less computationally efficient than the previous approaches that used a single merged surface, but not to a significant degree in terms of overall calculation times.

(2) Adding new legs

The objective of adding new legs within a thin layer is to create a connected network in that layer so that flow can occur within it. The basic approach is simply one of identifying neighbouring nodes, linking them with network legs, and setting the properties appropriately.

This is done as the new nodes are added and then for any nodes that were already in the thin layer.

A suite of algorithms are used to introduce the new legs based on basic and special-case Two-D scenarios (Fig. 7.3.9 – 7.3.14). The algorithms work in three dimensions since they are only based on the network connectivity.

Basic Cases

The basic scenarios consist of four geometries (Fig. 7.3.9):

(a) Corner crossing: in this case the thin formation cuts the corner of four existing nodes. Existing nodes are referred to as "old" nodes.

(b) Side-to-side: in this case the new formation cuts horizontally or vertically across four old nodes.

(c) Corner crossing – one old node: in this case the thin formation cuts across the corner of four old nodes but includes one of the four old nodes in its volume, and is therefore only represented by one new node.

(d) Diagonal crossing: in this case the thin formation cuts diagonally across four old nodes but no new nodes are added since no legs cut across the thin formation.

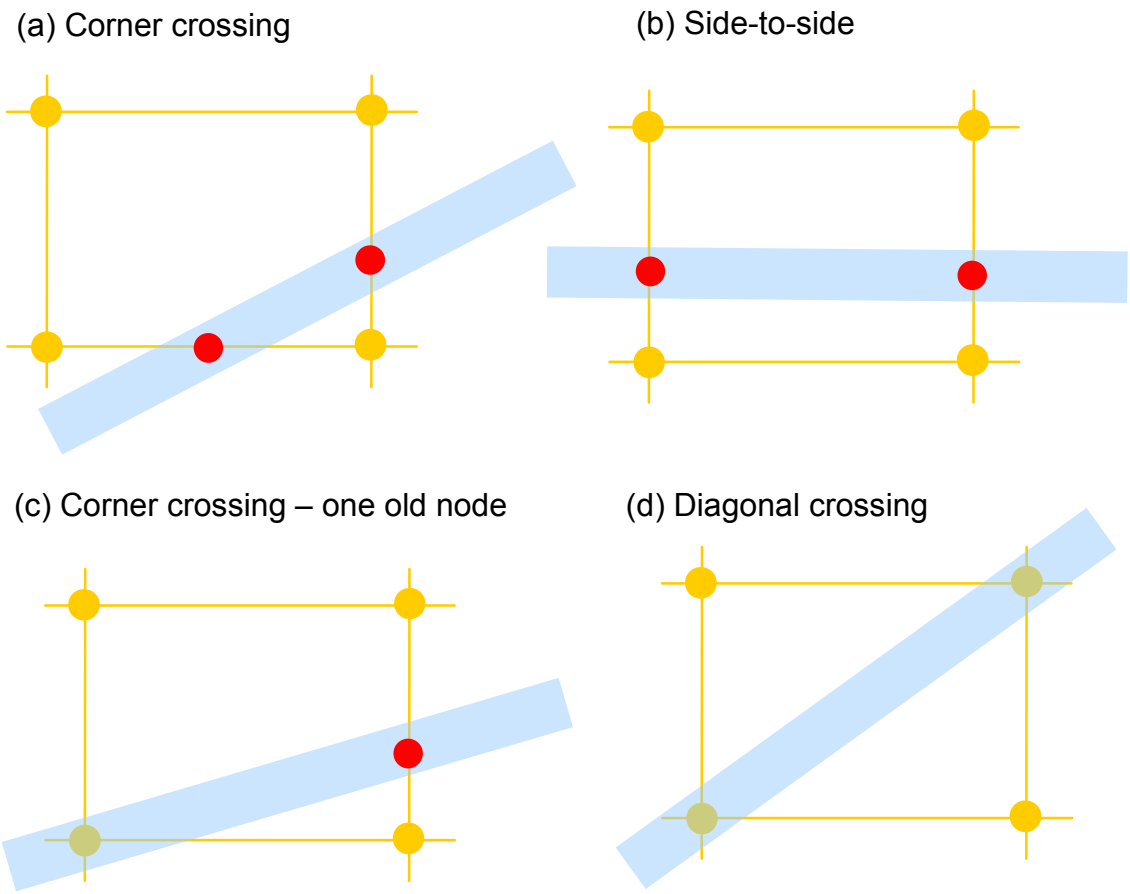


Fig 7.3.9 Basic situations that can arise where a new leg is required. The figure illustrates four "old" nodes, shown in yellow, being intersected by a thin formation, shown in blue. New nodes are added by the algorithm discussed in Section 7.3.3(1) but must be connected by network legs to account for the possibility that the thin formation has a high conductivity.

The algorithm detects these situations by looking at the network connectivity. Three of the cases (a, c, and d) can all be detected from a general rule.

- If two nodes in the thin layer are connected to the same node outside the layer then they should be connected.

The side-to-side case (b) is more difficult. However, the general rule used for the other three basic cases may be applied to the neighbouring legs both crossing the thin layer (Fig. 7.3.10).

- If N_1 and N_2 are two nodes in a thin layer which have been added to legs A_1B_1 and A_2B_2 respectively (where the A nodes are above the layer and the B nodes are below), then the nodes should be connected if there are legs A_1A_2 and B_1B_2 already in the network.

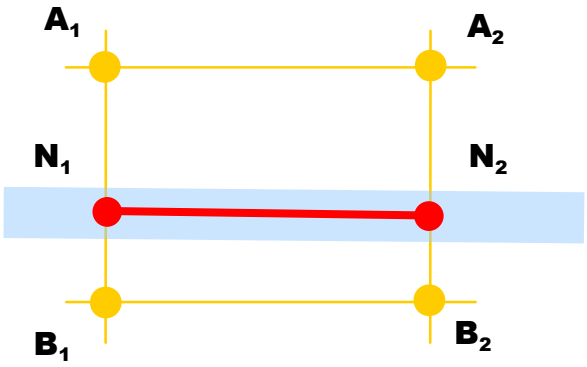


Fig. 7.3.10 Connection rule notation for side-to-side connections.

Special Cases

These basic rules need to be modified to account for two special-case situations:

- Two thin layers: where there are two adjacent thin layers; and
- Near-boundaries: where boundary nodes are involved.

For two thin layers, there are several cases that must be considered. The first two cases are shown in Fig. 7.3.11.

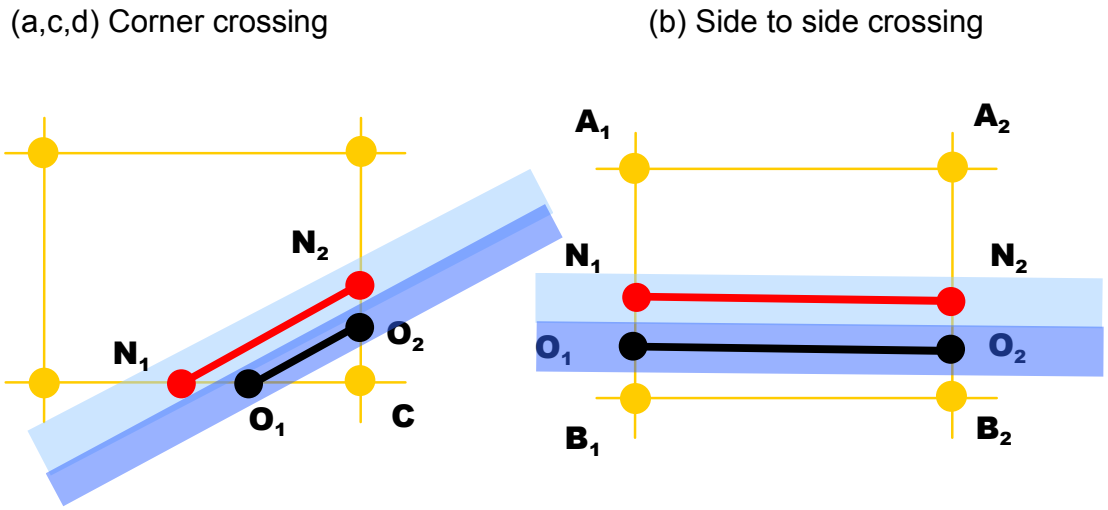


Fig. 7.3.11 Two thin layers, showing an old leg (O_1O_2) and a required new leg (N_1N_2).

For the side-to-side crossing no special action is needed, the O_1O_2 leg takes the place of B_1B_2 and the basic rule operates correctly. For the corner crossing, the O_1O_2 leg hides the corner node C and so the rule does not apply to N_1N_2. This is overcome by storing a list of corner connections and treating these as if they are the corner node.

There is one other case with thin layers that requires attention. This is when an old layer stops between nodes, as shown in Fig. 7.3.12. In this case, its is necessary to remember that the nodes A_1,N_1,O_1,B_1 are all on the same original leg, and to allow the O_1 node to be skipped in looking for connections.

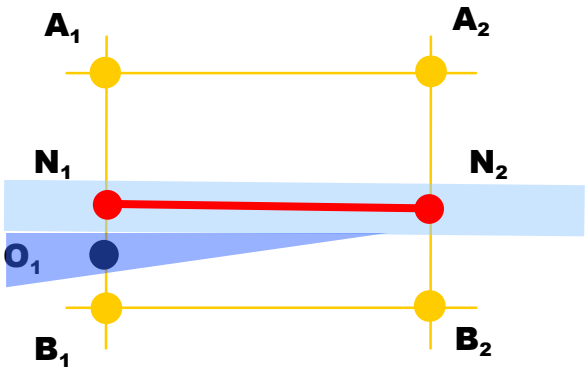


Fig. 7.3.12 Two thin layers, showing and old layer that terminated and a required new leg (N_1N_2).

For the near-boundaries special cases, two situations must be treated. These are shown in Fig. 7.3.13 and 7.3.14.

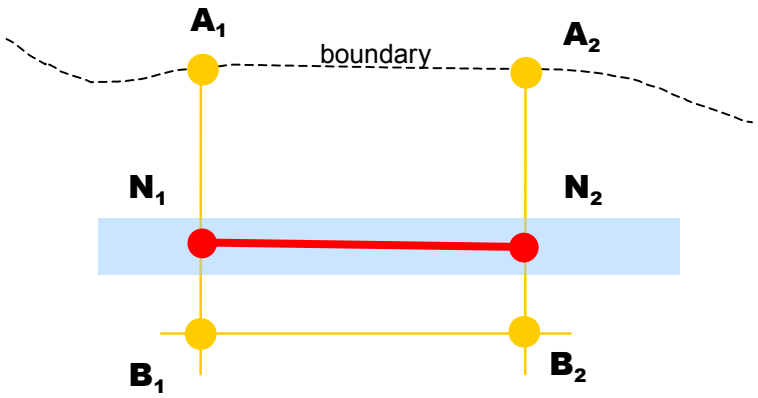


Fig. 7.3.13 Special case of crossing connection near a boundary.

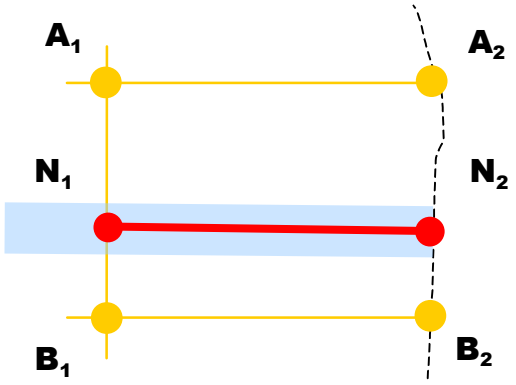


Fig. 7.3.14 Special case of a thin layer meeting a boundary.

To handle these cases, two new rules are introduced. The first is that nodes on a boundary are treated as if they are connected (Fig. 7.3.13). The second is that if the nodes at the end of the leg to which a new node has been added are both connected to the same boundary then a new boundary node is added (Fig. 7.3.14).

Specifying Properties

Once a new leg has been added its properties must be set. The length comes from the node positions and the area is set from the node volumes divided by this length. The areas for existing legs are adjusted to allow for this new path. This is done for the original paths that linked the nodes and defined the need for a leg. The "volume" of the legs is used as a basis (their area times length) – with the areas all scaled down so that the total leg-volume is preserved. This simple approach avoids double counting the available flow volume without being overly complicated. The accuracy of the volume calculations may be checked by comparing water balance reports when FracAffinity is run twice, first, with no thin formations, and second with thin formations.

7.3.4 Generation of Hydraulic Property Field

Once the network for the intact rock has been constructed, it is necessary to associate physical properties with the nodes. The stratigraphic information exported from EarthVision allows the identification of the formation and fault block within which a node lies. Using this information the porosity and storage coefficient (which have no heterogeneity within a formation) are easily assigned to nodes. The use of heterogeneous fractal hydraulic conductivity fields is more complex.

Two methods exist in FracAffinity to generate heterogeneous hydraulic conductivity properties: fractal generation and import of conductivity from a file. These are discussed in Sections 7.3.3 and 7.3.4.

In either case, anisotropy may be specified. The anisotropy factors are not heterogeneous and so can be assigned quite simply for each node. Each network connection may have a different anisotropy factor at each end, reflecting the fact that the nodes at each end may have originated from different zones.

Following the generation of the nodal conductivities, a scaling rule may be applied to some region – this simply multiplies conductivities by a given factor.

7.3.5 Generation of Fractal Hydraulic Conductivity Field

There are three main steps in the generation of the fractal hydraulic conductivity properties for the intact rock. The properties are defined according the formations defined in the EarthVision stratigraphy. Properties are generated for each stratigraphic unit in each fault block separately, thus the first step is to identify the region to generate properties for and the formation to which it belongs (to get the correct properties) (Fig. 7.3.15).

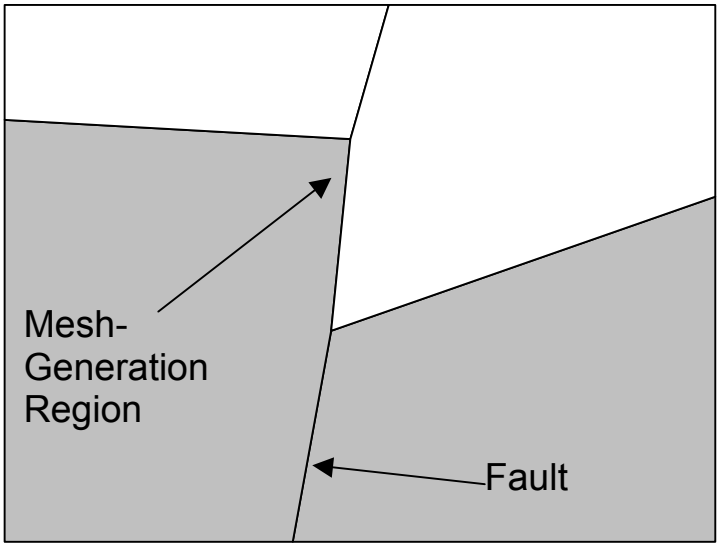


Fig. 7.3.15 A stratigraphic unit defined in a single fault block for which a mesh is generated and fractal properties calculated.

The stratigraphic units are generally rather irregular in shape with grids which do not have regular spacing. This makes it difficult for standard fractal generation algorithms to produce a hydraulic conductivity field that matches with the nodes of the intact rock model. To tackle this problem, FracAffinity makes use of an intermediate mesh that is regular in spacing. The fractal hydraulic conductivity properties are generated first on the regular mesh and then interpolated (or upscaled) onto the irregular structure of the intact rock model. The resolution of the intermediate mesh should preferably be higher than that of the intact rock grid.
Thus the stages are

1. Identify the cuboid which encompasses the required portion of the intact rock model.
2. Construct a regular mesh which covers the cuboid and generate a fractal hydraulic conductivity field on that mesh.
3. Interpolate the hydraulic conductivity values from the regular mesh back to the nodes of the intact rock model.

Step (1) is carried out by using the stratigraphic information exported from EarthVision. The method used to generate the fractal on a regular mesh is the Mid-Point Displacement Algorithm and this is described in detail in the next subsection. The following subsection describes the interpolation process which assigns appropriate hydraulic conductivity values to the rock model nodes, using the mesh.

(1) Mid-Point Displacement Algorithm

The Mid-Point displacement algorithm is well known as a rapid method of generating fractals on two-dimensional meshes. For FracAffinity, this method has also been extended to three dimensional meshes to model spatially heterogeneous permeability fields, conditioned on existing borehole measurements. All these previous methods generated only isotropic fields with a given fractal dimension and variance.

The isotropic fractal model of spatial heterogeneity assumes a variogram, $\gamma(h)$ defined by

$$\gamma(h) = \left\langle \left| \psi(\mathbf{x}) - \psi(\mathbf{x}-\mathbf{h}) \right|^2 \right\rangle = ah^{2p} \tag{7.3.1}$$

where $\psi(\mathbf{x})$ is the hydraulic conductivity field, $\mathbf{h}$ is a displacement vector (with magnitude h), p is the Hurst coefficient and a is a constant. The Hurst coefficient is related to the fractal dimension, D, by $p=n+1-D$ for a spatial dimension, n. Using this definition, the fractal dimension for a Three-D region will be in the range 3-4.

This can be modified to an anisotropic variogram by scaling the components of the vector h, to give

$$\gamma(h) = a\left[\left(\frac{h_x}{\alpha}\right)^2 + \left(\frac{h_y}{\beta}\right)^2 + \left(\frac{h_z}{\delta}\right)^2\right]^p \tag{7.3.2}$$

with, α β and δ giving scalings in each direction. Since the constant a has no physical significance, when each of the scale factors are equal, this expression reduces to an isotropic form.

Assuming that we have a unique correlation length in each direction and that the overall variance is an isotropic quantity, we can plot the variogram on a log-log plot to obtain the graph shown in Fig. 7.3.16. At distances greater than the correlation length (in this case h^0) the sill has a constant value corresponding to the variance of the overall field "at infinity", σ. Considering the point where the variogram meets the sill in the variogram for the x-direction, we have

$$\ln\sigma = \ln a + 2p(\ln h_x^0 - \ln\alpha) \tag{7.3.3}$$

Since the coefficient, a, is not physically significant, we can choose it to be equal to leading to the conclusion that $h^0 = \alpha$ and similarly that β and δ can be taken as the correlation lengths in the y and z directions respectively. From this, it is clear that the anisotropy of the fractal can be expressed in terms of different correlation lengths in each principal direction.

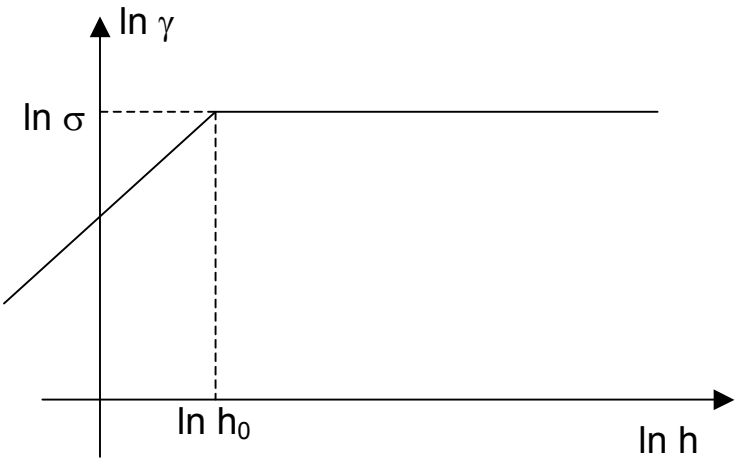


Fig. 7.3.16 A fractal variogram showing the sill effect.

The fractal algorithm developed from this analysis uses a single fractal dimension and variance and a fixed correlation length in each principal direction. The mid-point displacement method algorithm, is then applied using the scaled vector, h' when calculating separations where $h' = (h_x / h_x^0, h_y / h_y^0, h_z / h_z^0)$. This is equivalent to constructing an isotropic fractal with a correlation length of unity in the scaled coordinate system.

The basis of the Mid-Point Displacement Method is to iteratively generate fields which satisfy a fractal scaling law by successive refinement of a mesh. At each level of refinement point values are generated by taking the average of values generated at the previous levels of refinement and adding random values, sampled from a gaussian distribution whose variance depends on the level of refinement.

This basic approach has been extended to deal with conditioning data points, whose values must be honoured within the mesh, and anisotropic correlation lengths. The following description assumes for simplicity that the mesh has the same number of nodes in each direction and that the number of nodes is 2N + 1, where N is an integer.

The initial level of mesh refinement is generated by choosing random values for each corner of the mesh from a gaussian distribution of variance given by the variance of the field. This variance is specified by the user but should normally be consistent with the variance of any conditioning data.

Each subsequent level of refinement is carried out in three stages: filling in the cell centres, cell edges and cell faces, as shown in Fig. 7.3.17.

The first stage is carried out by setting the node at the centre of the mesh to have a value given by the mean of the values at the mesh corners plus a random addition from a gaussian distribution. The distribution's variance, is given by

$$\sigma' = \begin{cases} \sigma h^p & \text{if } h < 1 \\ \sigma & \text{if } h \geq 1 \end{cases} \tag{7.3.4}$$

Here h is the scaled distance from the mesh point to the corner points, and the scaled correlation length is isotropic and equal to unity. This ensures the correct behaviour for the variogram in normal space.

The second stage of refinement is achieved by filling in the values at the centres of the edges of the region. The new values are equal the mean of the values at each end of the edge plus the random addition, with variance determined using the scaled distance from the edge centre to the ends, in Equation 7.3.4.

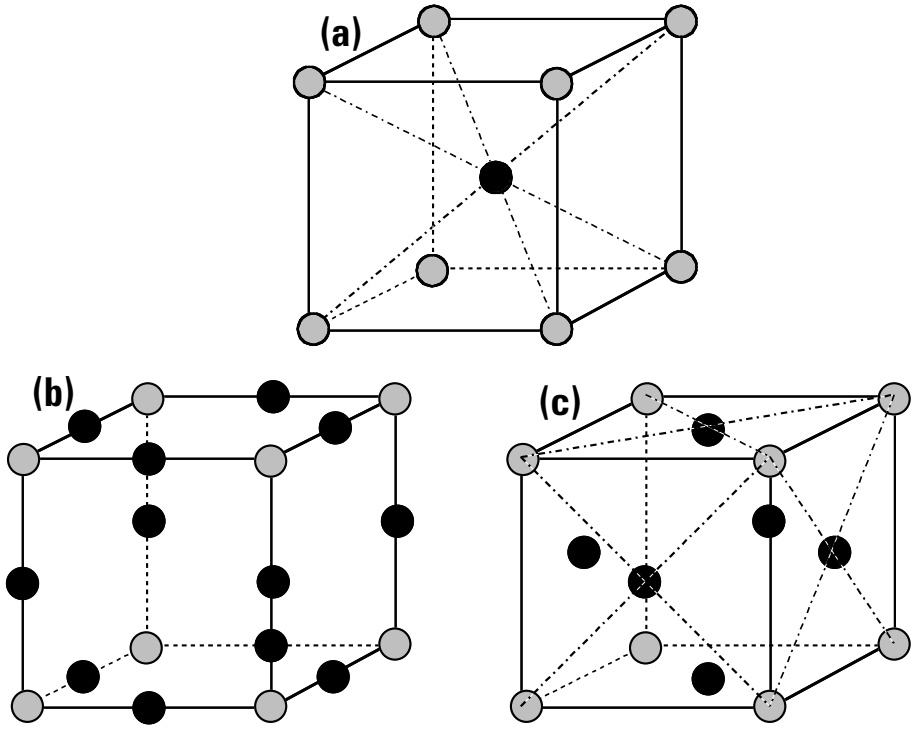


Fig. 7.3.17 The three stages of mesh refinement: (a) cell center (b) cell edges (c) cell faces. Grey spheres exist from the previous refinement level.

The third stage, involves setting the values for nodes at the centres of the cell faces. The starting value is chosen to the mean of the points which are closest in the scaled space. This can be an average of 2, 4, or 6 values, taken from neighbouring cell edges or cell centres as appropriate. After this third stage, a new cubic mesh has been constructed at a finer resolution and the process can be repeated until all the node values have been set.

As each node's value is set, a check is made for the nearest conditioning point to the node in question. If the conditioning point is closer than any of the averaging points then the value of the conditioning point is used instead of the average. The variance for the random addition is then calculated using the distance to the conditioning point in Equation 7.3.4. This ensures that the fractal will be consistent with the conditioning points, although a node will only have the same value as a conditioning point when the node and conditioning point coincide in space. (Note that where several conditioning points are equidistant from the node, the values are averaged).

(2) Interpolating Fractal Hydraulic Conductivity Fields

Once the mesh has been generated and the fractal hydraulic conductivity values assigned to the mesh cells, it is necessary to map the hydraulic conductivity properties of the regular mesh back to the irregular intact rock network. The approach used is relatively simple.

Each node in the intact rock network is considered in turn and the mesh cells whose centroids are close to the node are used to control the hydraulic conductivity. The $\log_{10}$ hydraulic conductivity of the grid node is calculated as an average of the $\log_{10}$ hydraulic conductivity of those mesh cells whose centroids are contained with a spherical region around the node with volume equal to that of the node.

In the case where there are no mesh cell within the nodal volume, the nearest mesh cell is found and it log hydraulic conductivity value used for the node. If the mesh resolution is chosen carefully (this means greater than the intact rock grid resolution) then this case should rarely occur. If the mesh resolution is too coarse, the fractal scaling law will be poor on length scales below the mesh grid spacing as many neighbouring network nodes will have the same hydraulic conductivity value.

7.3.6 Importing Field of Hydraulic Conductivity Values

The process of importing hydraulic conductivity data is very simple.

The user-specified file has a collection of $\log_{10}$ hydraulic conductivity values at particular locations.

For each node, the nearest point to the node position is determined and its value is used. Note that earlier versions did this on a grid cell basis which made nodes from local refinement all pick up the same value – the new version allows them to have different values.

7.4 Deterministic Discrete Features

7.4.1 Background

DDFs are used to represent mappable features within the flow network. DDFs most commonly represent faults (they are generally referred to as faults in this section) but can be used to model any mappable two-dimensional feature (e.g. a thin formation or a dyke). In the EarthVision model these are represented as surfaces, and the equivalent surface files are exported for use in FracAffinity. In addition polygon files may be exported and these are used, as described below, to model the limits of DDFs which do not cross the entire Model Region.

The surface files used by FracAffinity define the elevation of the surface as an array of (x, y) positions and elevations (Two-D grids) and are not allowed to be multi-valued because FracAffinity has no equivalent of the EarthVision Three-D grids[4]. Three-D grids are used in EarthVision to model recumbent folds or salt domes. This means that folded surfaces used by FracAffinity cannot be recumbent. Each fault in the EarthVision model is exported as a surface file for FracAffinity, but the user has control over which of the surfaces are actually to be gridded and used in the construction of the hybrid medium. Faults which would not be expected to have an impact on the hydraulic properties of the model may therefore be neglected.

If a fault is associated with a polygon, the polygon is used to define the extent of the fault in the Model Region. The polygon file defines a 2D polygon in x, y space inside of which the fault will be generated. It is important to note that the polygon will be used to restrict the surface that is used. Any surface point that is outside the polygon is ignored. It is therefore recommended that surface gird files for use with polygons use a relatively small x-y grid spacing.

Each surface is converted to a DDF by a similar process to that used for the intact rock. A grid is first calculated for the fault, properties are then associated with the grid and the grid is converted to a FracAffinity DDF network.

These processes are described in the remainder of this section.

7.4.2 Gridding

Despite the similarity in name, the grid of a surface for FracAffinity is independent of the EarthVision definition of a grid. The FracAffinity grid construction is intended to be used for the flow calculation and its resolution is controlled by the FracAffinity user. The grids exported from EarthVision are of fixed grid spacing which cannot be controlled by the FracAffinity user.

EarthVision surface grids define the elevation (z-coordinate) at each x-y position. In order to find the surface position at intermediate points, FracAffinity uses a bilinear interpolation, giving the EarthVision values at the EarthVision grid points.

The FracAffinity grid is constructed as a rectilinear grid in the x-y plane. The slope of the surface is taken into account when selecting the grid spacing. The user-specified grid size in the x-direction is used to limit the maximum distance along the surface between adjacent grid lines. The grid spacing in the x and y directions is controlled by the user through the specification of the maximum sizes and minimum number and resolution parameters.

The FracAffinity grid is generated as a two-dimensional grid (in Three-D space), but the cells also have an aperture, so there is a volume associated with each cell, as shown in Fig. 7.4.1. The closeness with which the FracAffinity grid follows the EarthVision surface therefore depends on the gridding parameters that have been selected. The nodes used will always be on the surface.

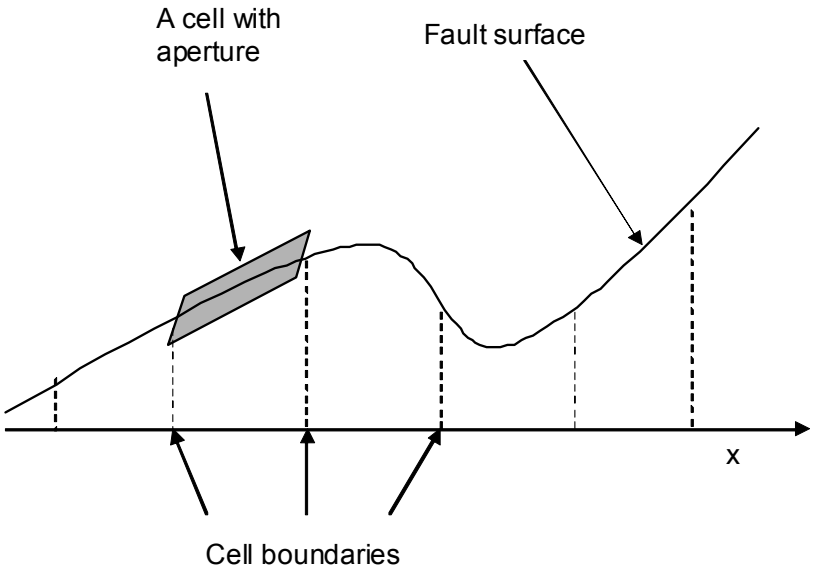


Fig. 7.4.1 An example of the construction of a DDF grid.

7.4.3 Hydraulic Properties

As for each formation in the intact rock, the porosity and specific storage coefficient of a DDF are uniform, while the hydraulic conductivity may be imported, uniform, anisotropic or fractal in nature.

Importing hydraulic conductivity was discussed in the previous section for Intact Rock. The situation for a DDF is very similar, with the nearest file location being used for each node.

For DDFs with uniform hydraulic conductivity a single value is applied to all property nodes.

If anisotropic hydraulic conductivity is specified, then the anisotropy factors for the in-plane and normal directions are used. The connections in the network all take the in-plane value. The normal value is only used when combining the DDF network into the Hybrid Medium Network, when it is applied to a short length (half of the DDF aperture) of for any connections that cross the DDF.

The approach used to generate fractal hydraulic conductivity values for the features is analogous to that used in the intact rock model. The fractal is first generated on a regular mesh and then interpolated onto the cells within the grid whilst the grid is projected onto the x, y plane. In the context of the DDF, the application of anisotropic conditions with fractal hydraulic conductivity is simple because there is no anisotropy in the plane of the DDF, and use of the normal anisotropy for an intersection is always associated with a specific node which has the hydraulic conductivity value recorded.

7.4.4 Constructing DDF Networks

The final step for constructing a DDF rock model is to turn the grid into a FracAffinity network. Again, this follows an analogous process to that used by the intact rock grid. Each cell has a node placed at its centroid. The location of the centroid (x,y,z) of the cell is calculated using linear interpolation from the (x, y, z) co-ordinates of the corners of the grid cell. Following this, neighbouring cells have connections made between them: porosity, hydraulic conductivity and storage are associated with nodes while connection lengths and cross sectional areas are associated with the connections. The only difference here is that the DDFs have an extra item of data, namely the aperture. The aperture is not used as part of the flow calculations, but is required for calculations of pathlines.

The aperture and normal anisotropy values are not stored for each node and connection. Instead, the DDF that is associated with the nodes and connections is recorded so that such information can later be retrieved.

7.5 Stochastic Discrete Features

7.5.1 Background

The SDFs within FracAffinity are designed to represent features that are considerably smaller than those that are modelled with deterministic features. In most cases, SDFs will be used to model fractures and will be generally referred to as such in this section. The precise locations and geometry of SDFs are unlikely to be well defined, but their statistical distributions can be sampled in boreholes, tunnels and shafts, or mapped at the surface. Given the statistical properties of these fractures, such as fracture density or mean size, it is possible to generate a network of SDFs stochastically, with the appropriate distribution for density, orientation, size and hydraulic parameters.

Given the uncertainty in specifying the structure of individual features and the large number of features which may be present, it is not practical to model the geometry of any individual features with a great degree of accuracy. Thus, the features generated within FracAffinity are all rectangular. Changing the details of the features shape (e.g. using ellipses, or higher order polygons) could affect the connectivity of a fracture network, but these effects are only significant for networks which are poorly connected. Given that no real features are likely to have such a convenient geometrical shape, the use of rectangles is a reasonable compromise between realism and simplicity.

7.5.2 Generating Stochastic Discrete Features

SDFs are defined by distributions which are associated with the formations, as provided by the EarthVision stratigraphic information. When a formation is known to contain small scale fractures, the user may specify that SDFs are to be generated in that formation with a given distribution.

The initial stage of the SDF generation is to identify the volume of the model for which fractures are to be generated. The method used is the same as that used to identify the volume to be used in generating a mesh for the fractal properties of the intact rock (see Fig. 7.3.7). Thus, a cuboid is constructed which contains the formation of interest in a single fault block.

Fractures are then generated one-by-one in this region of interest, as illustrated in Fig. 7.5.1. The fractures are generated according to the input parameters specified (see below) and the degree to which the fractures intersect will depend strongly on the fracture density (number of fractures per cubic metre of rock), size and the orientation distribution. Obviously, if the fractures are too sparsely distributed or all orientated in the same direction, the number of intersections will be very small.

Some care needs to be taken to ensure that the fracture size is compatible with the grid resolution of the intact rock model. If the intact rock grid is very coarse and the fractures very small and sparsely distributed, then it is possible that fractures will be generated that do not intersect any part of the intact rock network, or model region boundary. In this case, the fractures are removed from the final network (see Section 7.2.6), since they will have no impact on the properties of the region.

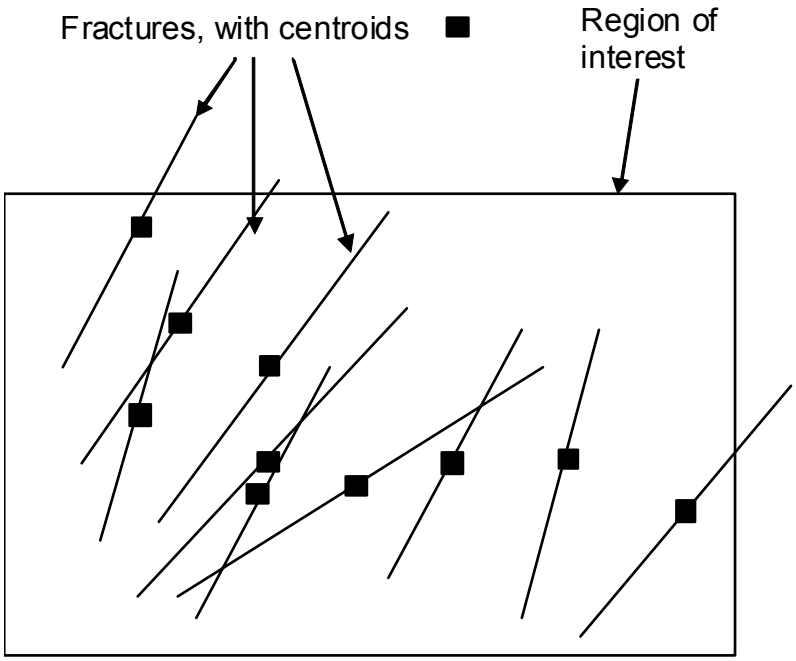


Fig. 7.5.1 An example (in Two-D) of fractures generated in a region of interest. Note that fracture may extend beyond the boundary, so long as their centroid lies in the region.

Notice from Fig. 7.5.1 that it is possible for a fracture to extend beyond the region of interest (but not beyond the model region) so long as the fracture's centroid lies within the region of interest. This behaviour is realistic for some geological formations but may not be for all. User control over the truncation behaviour of fractures at stratigraphic boundaries is a possible area of future development for FracAffinity.

Fractures which cross the boundaries of the Model Region are truncated by those boundaries. This means that the rectangular fractures may be converted to polygons at this stage. This truncation process is important since it prevents large numbers of nodes from being generated outside the Model Region.

This process is repeated for all the required distributions in all of the required regions and a given formation may have more than one fracture distribution. All of the fractures generated are then truncated by the Model Region to produce polygons. These polygons are turned into mini-networks of nodes and these mini-networks integrated into the final FracAffinity network, as described in Section 7.2.3.

The creation of networks for the SDFs is handled more directly than for the Intact Rock or DDFs. There is no intermediate grid generated. Instead a rectilinear network is directly created on the SDF. This has nodes at the corners and as many internal nodes as are needed to satisfy the user-specified maximum grid size. The spacing is uniform in each direction. An example was given in Fig. 7.2.4.

7.5.3 Location Distribution

The fractures are distributed randomly through the region of interest with a uniform spatial distribution for fracture centres and user specified fracture density (number of fractures per unit volume).

It is also possible to specify single "fixed" fractures that have a specified location and orientation. An SDF centroid will be placed at the location specified by the user and the remaining properties of the SDF will defined as described below. This can be useful for introducing specific features into the stochastic fracture network, such as a low conductivity barrier which is too small a region to be satisfactorily incorporated into the intact model, or too uncertain to be reasonably added to the geological model. This capability to rapidly add small localised features can be useful for sensitivity analysis or "what if" calculations.

7.5.4 Size and Shape Distributions

(1) Uniform Rectangles

There are three distributions available in FracAffinity that control the size and shape of the stochastic fractures. These are Uniform Rectangles, Power Law Rectangles and Power Law Squares.

It should be noted that the use of wide ranging distributions can make estimating the appropriate number density difficult. Thus, an iterative approach to deciding on appropriate density is recommended.

(2) Power Law Rectangles

This distribution generates rectangles whose side lengths are sampled from a uniform distribution. The user specifies the maximum and minimum sizes for the major and minor axes of the rectangles. Each SDF is then generated as a rectangle with side lengths sampled from distributions from these ranges.

(3) Power-Law Square Fracture

This method of generating fractures samples the major axis lengths for the rectangles from a power–law distribution:

$$f(L) = \frac{b-1}{L_{\min}}\left(\frac{L_{\min}}{L}\right)^{b}, \qquad L_{\max} > L \geq L_{\min},\ b > 1\ . \tag{7.5.1}$$

The user specifies $L_{\min}$, $L_{\max}$ and b, the minimum length, maximum length and exponent, respectively. They also specify a range of values for the aspect ratio of the rectangles. The minor axis of the rectangle is generated by multiplying the major axis length by an aspect ratio selected from a uniform distribution. The use of the aspect ratio distribution to define the minor axis length allows the user to prevent the generation of very long thin rectangles should this be necessary.

7.5.5 Orientation Distributions

The orientation of the fractures can be crucial in controlling the connectivity of a fracture network. There are three different methods available in FracAffinity for defining the orientation of networks of SDFs These are Uniform and Fisher distributions and the sampling of SDF orientations from an input file.

Dip angle and dip direction together specify the slope of the plane in which an SDF is situated. The centroid position places this at a definite location and the rotation angle allows the principal axes of the SDF to be rotated within the plane. All the angles are specified in degrees.

To visualise these angles, we start by drawing the plane in which the SDF is situated with a square grid on it. One set of lines are horizontal, while the other set is aligned with the steepest slope.

The dip angle is then the angle that the steeply dipping lines make to the horizontal. This is the same angle that the normal to the plane makes with the vertical.

The dip direction is the direction from North that these steeply dipping lines have when viewed from above.

To illustrate this we show a plane with a dip angle of 60º and a dip direction of 90º (that is East). The steeply dipping lines are thick with an arrow pointed down dip; the horizontal lines are thin with no arrow. Fig. 7.5.2a shows the plane as viewed from the South (the horizontal lines are not visible from this direction) and Fig. 7.5.2b shows the plane as viewed from above (the dip of the lines cannot been seen from this direction).

The rotation angle specifies how the SDF is oriented with respect to the steepest dip lines. The length of the SDF is taken to be broadly down the dip direction, but a random angle between –45º and 45º is taken for the rotation. Fig. 7.5.2c shows an SDF on the plane specified previously, with a rotation angle of 15º. The SDF is viewed down the normal.

(1) Uniform Orientation Distribution

For this distribution, the user specifies the minimum and maximum dip angle and minimum and maximum dip direction for the fractures. The actual dip angle and direction of each fracture is then sampled from these distributions.

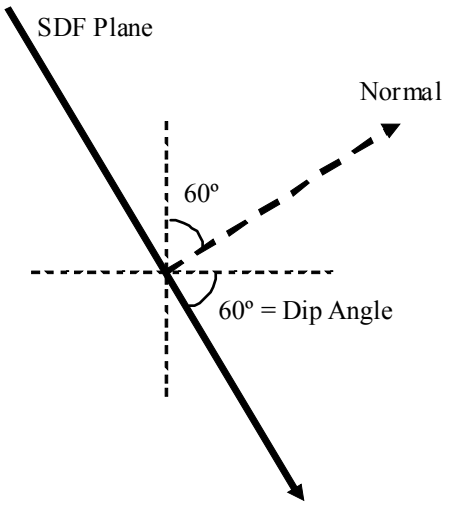


Fig. 7.5.2a An SDF with dip angle 60º and dip direction 90º viewed from the South.

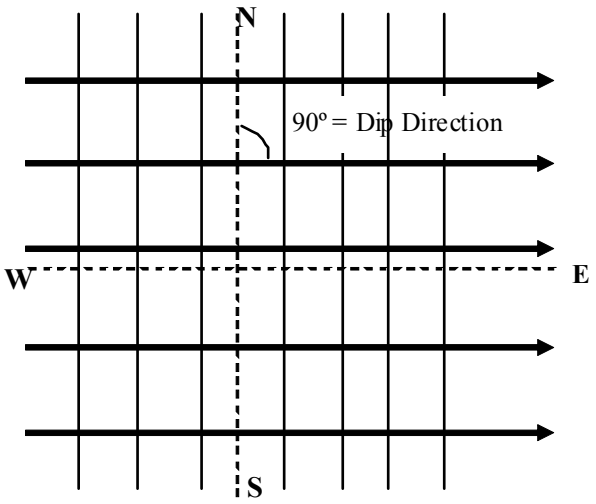


Fig. 7.5.2b An SDF with dip angle 60° and dip direction 90° viewed from above.

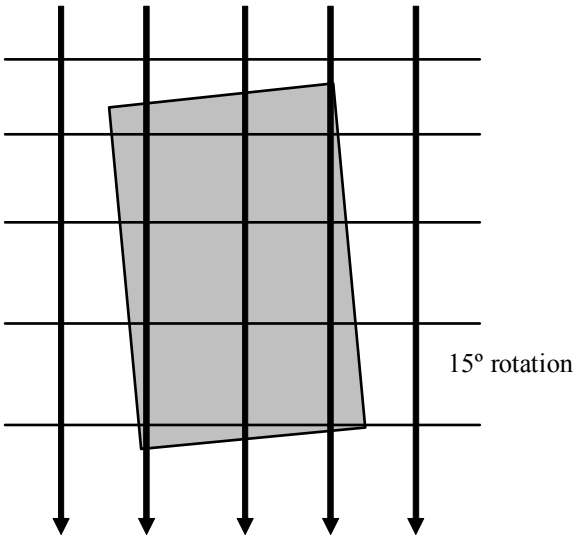


Fig. 7.5.2c An SDF with a 15° rotation angle, viewed down the normal to the plane.

(2) Fisher Distribution

When using the Fisher distribution, the user specifies the main dip angle and direction about which the distribution is made and the parameter, κ. The SDF planes are then distributed with normals at an angle θ to the main directions such that the probability density is given by:

$$f(\theta) = \frac{\kappa}{4\pi \sinh \kappa} \exp(\kappa \cos \theta). \tag{7.5.2}$$

The coefficient κ is positive and has a standard value in fractured granite of 10.

(3) Sampling of SDF Orientation from an Input File

When the orientations are sampled from a file, the user directly specifies a set of dip angles

and dip directions. These are read in and randomly sampled from (as a pair).

7.5.6 Hydraulic Conductivity Distributions

There are two hydraulic conductivity distributions available for SDFs in FracAffinity. These are uniform and truncated normal distributions. Both of these are actually distributions for the $\log_{10}$ value of hydraulic conductivity.

In addition, a tortuosity value is specified for each SDF set and anisotropy factors may be given. The anisotropy factors are used as for the DDF case, with the in-plane value used within the SDF and the normal value used when connections cross the SDF. The tortuosity is used to reduce the hydraulic conductivity within the plane (to simulate the effects of longer paths).

(1) Uniform Hydraulic Conductivity Distribution

To specify a uniform hydraulic conductivity distribution, the user must merely provided the minimum and maximum values of $\log_{10}$ hydraulic conductivity. Log_{10} hydraulic conductivities are then sampled from the distribution.

(2) Truncated Normal Hydraulic Conductivity Distribution

This more complicated hydraulic conductivity distribution assumes that the $\log_{10}$ values of hydraulic conductivity are distributed according to a normal distribution. Since the normal distribution allows for an infinite range of hydraulic conductivity values, the distribution is truncated by specifying an upper and lower bound for the $\log_{10}$ hydraulic conductivity values. Thus for this distribution, the user must specify the mean $\log_{10}$ hydraulic conductivity, the standard deviation of the distribution and the upper lower and lower bounds.

7.6 Enhanced Conductivity Intersections

In some situations, there is evidence that preferential flow paths occur along the intersections between fractures, or other discrete features. FracAffinity has the capability to represent the intersections between discrete features as regions of enhanced hydraulic conductivity. Modelling such situations in FracAffinity requires several steps that are discussed in turn below. We do not consider here the issue of how fractures intersect in real systems (e.g. does one fracture terminate against another). We focus initially on the intersection of SDFs, but similar logic is applied to DDFs too.

The steps that must be followed are:

- Finding the intersections;
- Assigning properties for each intersection feature;
- Gridding the features and linking to the rest of the system.

7.6.1 Finding Intersections

SDFs are finite, rectangular planar features with any orientation. Finding the intersections between such features is straightforward. FracAffinity does not explicitly find such intersections for its normal network creation – instead, the SDFs are gridded separately and the grid of one is linked into the grid of the other when they are added to the system network. The intersections are line segments. Obviously, each intersection is in the plane of the two features. In may cross the feature completely, just reach one edge, or be fully internal.

Fig. 7.6.1 shows a typical intersection, in the full Three-D view and from the plane of each participating feature. Of course, it is possible that several features may all intersect another one, and the intersections may cross. The scheme described below should handle this adequately.

7.6.2 Generation of Property Field for each Intersection Feature

The intersection features are intended to represent enhanced flow pathways. Hydraulic properties must be assigned appropriately: both an area and a conductivity value area required. The lengths and volumes associated with the intersection feature can be derived geometrically.

The area associated with the intersection is derived from the fracture geometry by considering the fracture apertures and angle of interception of the fractures. The hydraulic conductivity is derived from an enhancement factor for each DDF or SDF Fracture Set. Considering two features with enhancement factors f1 and f2 with conductivities k1 and k2, the conductivity of the intersection will be given by

$$K = \max(f1 * k1, \quad f2 * k2).$$

The use of enhanced conductivities is switched on by the adding the keyword ENHANCED_CONDUCTIVITY to either the DDF Conductivity input file or the SDF Conductivity Input file (or both). Note that enhanced conductivities will apply to both DDFs and SDFs intersections whichever input file the keyword is defined in.

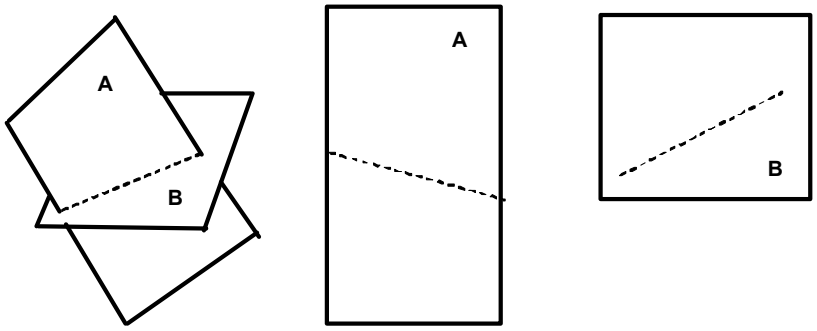


Fig. 7.6.1 Typical intersection between two SDFs in FracAffinity.

7.6.3 Gridding and Linking the Features

There are several things to bear in mind here. There must be network legs along the line of the intersection feature. These must be linked to the ordinary legs on each discrete feature. They must also link to other features that cross them.

The algorithm for achieving these objectives is described below. This relies on the general approach used in FracAffinity for merging networks on discrete features with those on the system network. This was described in Section 7.2.4 and is briefly recapped here.

Each discrete feature is gridded separately, and added to the system grid one by one. The intact rock grid is used as the starting point. As each new discrete feature is added, any legs on the system network that cross the feature are linked to it (by re-routing them via the nearest node of the feature). This has two important consequences: merging networks does not move any nodes; but, merging networks can move legs that arose from an SDF out of the plane of that SDF. This is difficult to draw, but take the case of the two fractures shown previously, with a coarse grid (Fig. 7.6.1). Fracture A is assumed to be added before fracture B. Fig. 7.6.2 below shows the amalgamated grid. It is clear that the intersection line remains in the plane of the legs for B, but not for A, which is no longer planar.

Thus, the following approach is used for enhanced intersections. The intersection feature is discretised and merged with the grid for B, before adding to the system network. Then, when B is added, the legs of A will be re-routed via the nodes on the intersection. With this scheme, the final combination network will look like the grid illustrated in Fig. 7.6.3. In this way, the intersection is properly connected to both A and B.

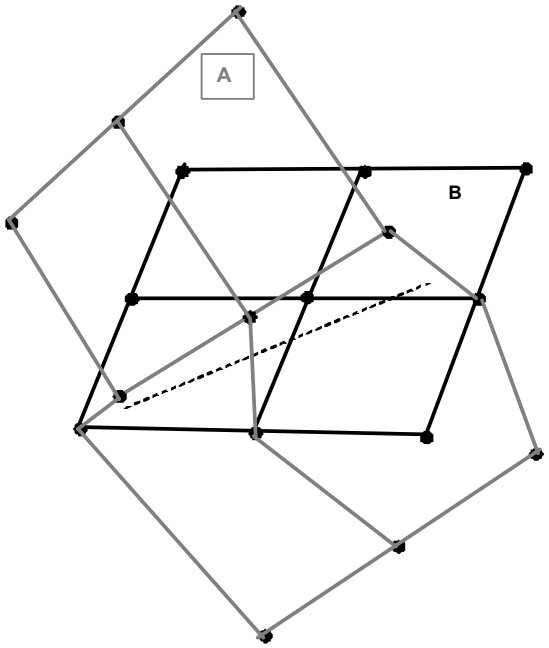


Fig. 7.6.2 Gridding of intersecting SDFs in FracAffinity.

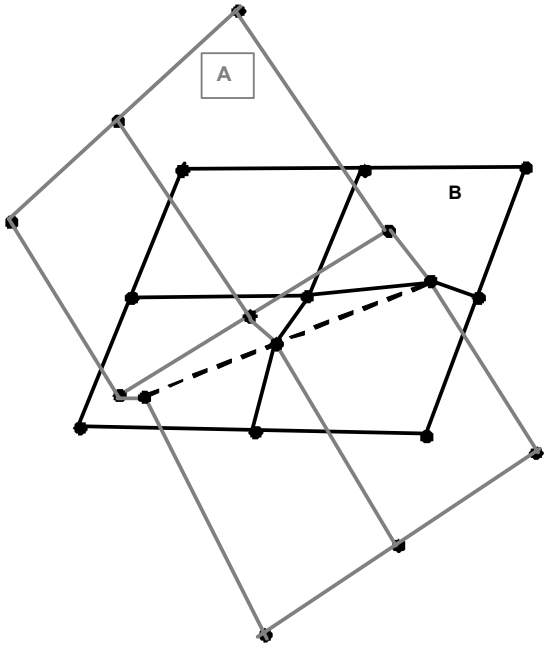


Fig. 7.6.3 Gridding of intersecting SDFs in FracAffinity with intersection explicitly represented.

When DDFs are considered, then the fact that they are not planar adds some complexity to the algorithm, but it remains fundamentally valid. The main difficulty is in determining the intersection itself. This is achieved by using a fine discretisation on the DDFs and determining which points on one surface are above the other surface. The boundary of this region is then the line of intersection.

7.7 Modelling Shaft Excavation

This section deals with the numerical implementation of the excavation procedure in a FracAffinity model. We start by first providing an overview of the excavation algorithms and progress by explaining the details of the gridding algorithms and how are they implemented.

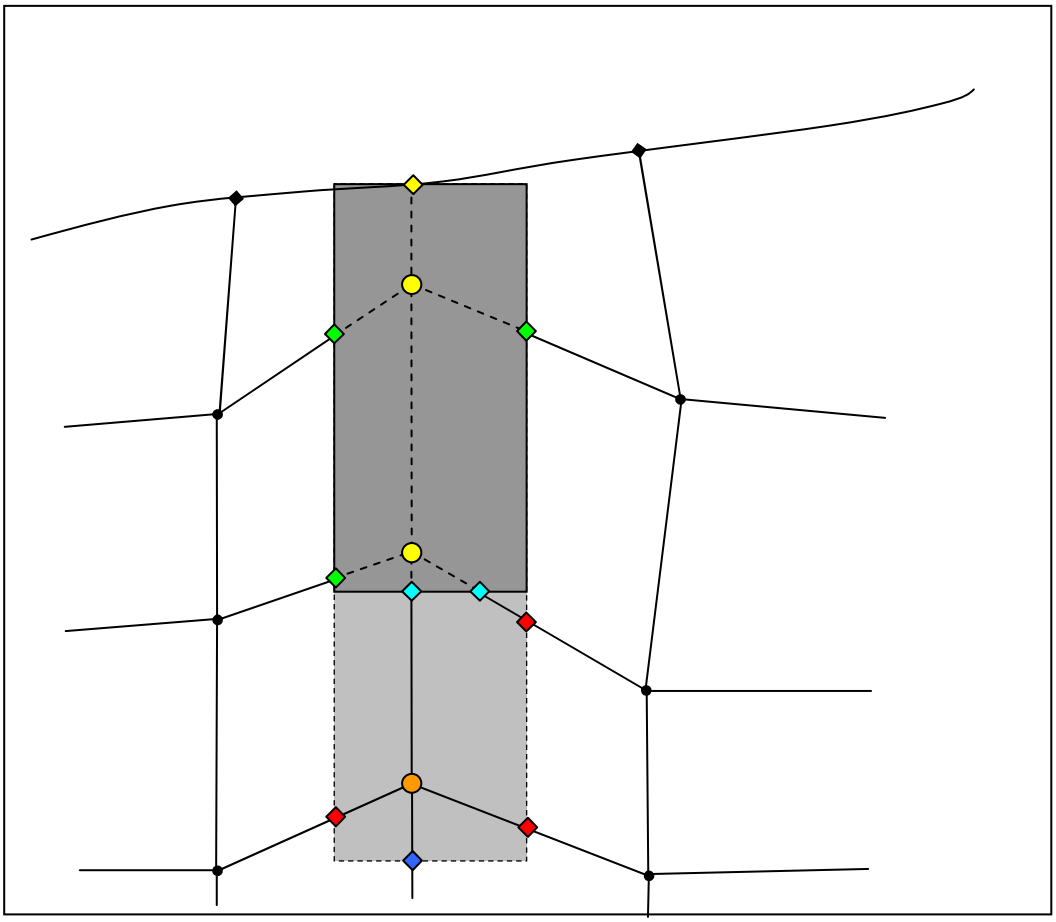
7.7.1 Overview

Time-dependency and dynamic shaft excavation has been a key feature of FracAffinity since its inception. In Version 3.2, this was extended to handle general excavations, but the basic approach remains unchanged. Here, we present an overview of the basic approach used in FracAffinity, indicating where the approach used for general excavations differs from that for vertical shafts.

The key steps in the approach are as follows (Fig. 7.7.1 illustrates this for a vertical shaft):

1. Step 1: The first step is to ensure that there are an adequate number of nodes in the excavated region for the excavation to be properly modelled. The approach differs between vertical shafts (which were the only option in FracAffinity 3.1 and earlier) or more general excavations.
    a. For vertical shafts, a column of grid cells is introduced around the shaft. In the absence of any local refinement this leads to a line of vertical nodes along the shaft centre. These lines of nodes form new planes of nodes across the model domain in both the x and y directions. If local refinement is used then there may be additional nodes in the excavated shaft but the basis of the algorithm is unchanged.
    b. For a general excavation, it is the user's responsibility to ensure that there is adequate refinement. To assist in this, the facility to refine the grid near an excavation is provided and warnings are produced if there are few nodes in an excavation (see Section 8.4.7).
2. Step 2: The nodes that are inside the excavation (Excavation Nodes) are identified. Given the typical grid spacings around shafts applied in FracAffinity models to date (5 m) and the radius of the shaft (3.5 m) the nodes inside a vertical shaft are typically the additional nodes added by FracAffinity in Step 1a. For a general excavation they will be some of the nodes added through the local refinement of Step 1b.
3. Step 3: Boundary Nodes are defined at the edge of the excavation. These Boundary Nodes link the nodes that lay inside the excavation and those that lie within the model domain (Internal Nodes) and can be of two types:
    a. Side Boundary Nodes, the position of these nodes does not change with time.
    b. End Boundary Nodes, the position of these nodes moves as the excavation progresses, and the position is calculated dynamically during the model run. As the excavation continues these Boundary Nodes are removed from the model domain and are referred to as inactive nodes.
4. Step 4: A list of nodes affected by the excavation is produced and the times at which they become inactive (Excavation Nodes and End Boundary Nodes) and active (all Boundary Nodes) is also listed.

5. Step 5: Transient groundwater flows are calculated using the transient solver and results are written to the FracAffinity Directory. For transient runs, monitor times entered into the Flow Solver Control File determine the times at which water balance reports are written to the FracAffinity Log File. During the run, a number of intervals are defined, with each interval representing the time between one Excavated Node becoming inactive and the next Excavated Node becoming inactive. The time step at the start of each interval is reduced to 100 seconds.



Legend	
◇	Top Boundary Node
○	Excavation Node Inactive
●	Excavation Node Active
◇	Side Boundary Node Active
◇	Side Boundary Node Inactive
◇	End Boundary Node Active
◇	End Boundary Node Inactive
／	Active Leg
┆	Inactive Leg

Fig. 7.7.1 Schematic illustration of excavated and boundary nodes for dynamic shaft excavation. The illustration shows an intermediate stage in shaft excavation, with the excavated shaft shown in dark grey and the shaft not yet excavated in light grey.

The rest of this section outlines the approach used to dynamically update the network as the excavation continues and solve the flow in such a dynamic network. Note that the diagrams use circles to represent internal nodes (where pressure is to be calculated) and diamonds to represent boundary nodes (at which boundary conditions are applied).
The ability to model any number of dynamic excavations was introduced in Version 3.3. Any number of static shafts and galleries may also be defined. However, the practical number of excavations, shafts and galleries is limited by the conceptual model applied within FracAffinity.

7.7.2 A Dynamic Network

A FracAffinity network is a complicated three-dimensional structure which requires considerable computational effort to construct. To allow flow calculations to take place on a sensible timescale, any attempt to dynamically update the position of the nodes and the connectivity of the FracAffinity network must be kept to the minimum.
The general approach is to set up as much information as possible about the nodes that will be affected before beginning a flow calculation and update the network accordingly as the solver steps forward in time. It cannot be assumed that the times at which the network needs modification will be monotonically increasing. Time-stepping solvers may attempt time-steps of a given size only to have the step rejected and try a smaller time-step. Thus the updating of the network cannot rely on any historical state (e.g. excavation length) of the network and must be able to calculate any required modification given only the current time.
The nodes which will be affected are those which will enter the excavation at some time up to the final excavation time. The first step of the algorithm is to identify all of the nodes which would be within the excavated region at any time during the simulation. The connections which will require dynamic updating are those connected to the nodes just identified.
Since the length of the excavation at a given time is known (it is defined by the user as a piece-wise linear function of time) it is clear at any time whether a node is inside or outside the excavation. Once a node is inside the excavation it must become inactive and its connection with the rest of the system must also become inactive. This is illustrated for the simplest case in Fig. 7.7.2.
The behaviour of linings is also modelled by defining the times at which various parts of the excavation are sealed. The relevant boundary nodes may then be made inactive at those times and sealed (no-flux) boundary conditions are applied.
In order to deal with connections that span different excavations, an initial analysis is performed and these connections are split at the point where they leave the earlier excavation. The separate parts can then be treated using the algorithms described here.

7.7.3 Connection Modifications

(1) Case 1

The first case to consider is illustrated in Fig. 7.7.2. This applies for connections where the node entering the excavation is the first point on the connection that the excavation reaches.
As the excavation reaches the node location, the node is made inactive and a previously computed boundary node becomes active. The connections to the inactive node are rejoined to

the boundary node, whose location varies with time. The actual condition to be applied at the boundary node will depend on whether the shaft is lined or not at that depth at the current time. A no flux condition is applied at lined segments of the shaft and dry air conditions otherwise.
In the example shown, the End Boundary Node moves and eventually becomes fixed and acts as a Side Boundary Node.

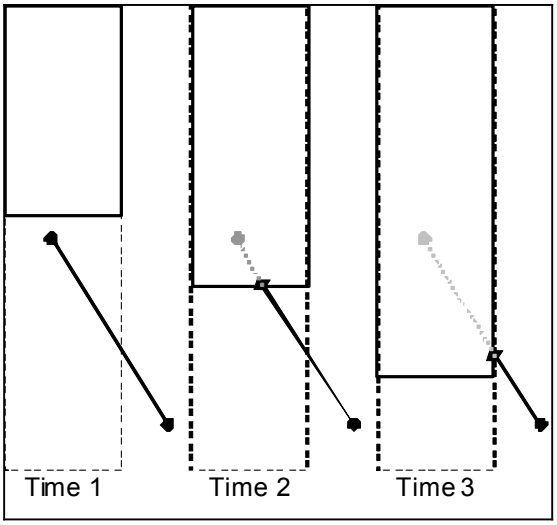


Fig. 7.7.2 A node becoming inactive (greyed out) as it enters the shaft. A new boundary node is generated at the interface with the shaft.

(2) Case 2

Where the first point on a connection that the excavation reaches is not at a node, a slightly different approach is needed, as demonstrated in Fig. 7.7.3. Here the connection is split into two and the original connection becomes inactive. Two new boundary nodes are made active, one of which is stationary (a Side Boundary Node) and the other (an End Boundary Node) moves towards its attached node as the excavation progresses. When the excavation reaches the attached node, the node and the connection become inactive.
Using these basic ideas, we can generate additional nodes and connections that can be made active or inactive as appropriate and with time dependent locations where needed.

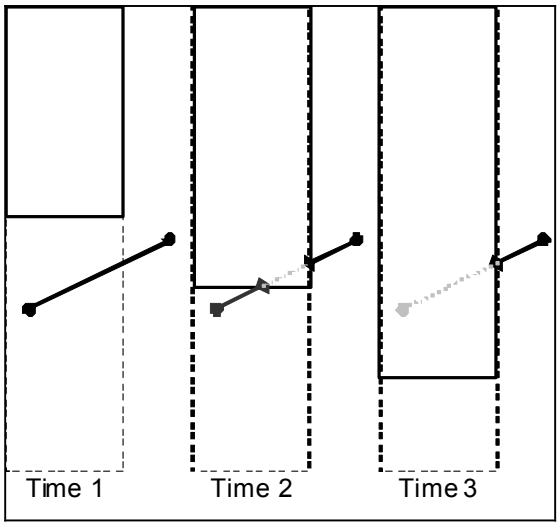


Fig. 7.7.3 A connection splits into two as it is crossed by the shaft. New boundary nodes are generated at the interfaces with the shaft (shown as diamonds). The lower connection becomes inactive as it enters the shaft.

7.7.4 The Algorithm

The notation used here labels internal network nodes where the pressure is to be solved for as i-nodes and boundary nodes as b-nodes.

The following assumptions are made about the state of the network nodes and connections entering or intersecting the shaft:

- All i-nodes start active and become inactive once they are within the excavation.
- Connections start inactive, become active at a given time and inactive again at a later time. The first time may be before the simulation start, the second after the simulation end.
- All b-nodes can be considered active at all times, but they will not be used when the connection with which they are associated becomes inactive.
- The position of b-nodes may be time-dependent, changing with every time step of the transient calculation.

The pseudo-code given in Fig. 7.7.4 then outlines the rules which are used to pre-process the network and calculate which nodes and connections are active at a given time. While this general approach is independent of the actual shape of the excavation zone, the details of how activity times and b-node locations are calculated are dependent on the shape and orientation of the zone.

The position of boundary nodes is relatively simple to calculate for the cylindrical shaft model. For each boundary node introduced, there is a starting location on the original connection and a final location on the connection. Boundary nodes on the bottom of the shaft are moving and those on the sides are stationary. Thus the relative position, *X(t)*, of the boundary node between its start and end points is proportional to the shaft depth such that

$$X(t) = \alpha X_{\text{start}} + (1-\alpha) X_{\text{end}} \;.....\; \alpha = \frac{z - z_{\text{start}}}{z_{\text{end}} - z_{\text{start}}}, \qquad (7.7.1)$$

where $z(t)$ is the vertical position of the shaft and the vertical positions of the start and end positions are z_{start} and z_{end}. The parameter α is restricted to values between zero and unity to limit the boundary node to points on the connection.
An example of how this algorithm will modify the structure of a network in the location of a node with several connections is illustrated in Fig. 7.7.5.

```
List all i-nodes that will enter the shaft during
excavation and set inactive times;

FOR each i-node (I) get a list of connections
{
  FOR each connection to other Node J
  {
    IF ( Node J is inside shaft and
          inactive time for J > inactive time for I )
    {
      Add new b-node at position of node I;
      Add inactive conn C from new b-node to J;
      Set C active time to be node I inactive time;
      Set C inactive time to inactive time for J;
      Calculate b-node position history;
    }
    ELSE IF ( Node I Excavated First(Case2) )
    {
      Add new b-node at position of node I;
      Add inactive conn C from new b-node to J;
      Set C active time to be node I inactive time;
      Set C inactive time to DBL_MAX;
      Set current conn inactive time to be C active time
      Calculate b-node position history;
    }
    ELSE IF ( Connection excavated first(Case1) )
    {
       Add 2 new b-nodes at initial intersection of
                 shaft and connection;
       Add inactive conn C1 from 1st b-node to J;
       Add inactive conn C2 from 2nd b-node to I;
       Set each C1, C2 active time to be b-node
                                   excavation time;
       Set current conn inactive time to be C1 active time
       Set each conn inactive time to be DBL_MAX;
       Calculate b-node position history;
    }
  }
}
```

Fig. 7.7.4 Pseudo-code describing the pre-processing of a network to generate all connections and boundary nodes needed for a calculation.

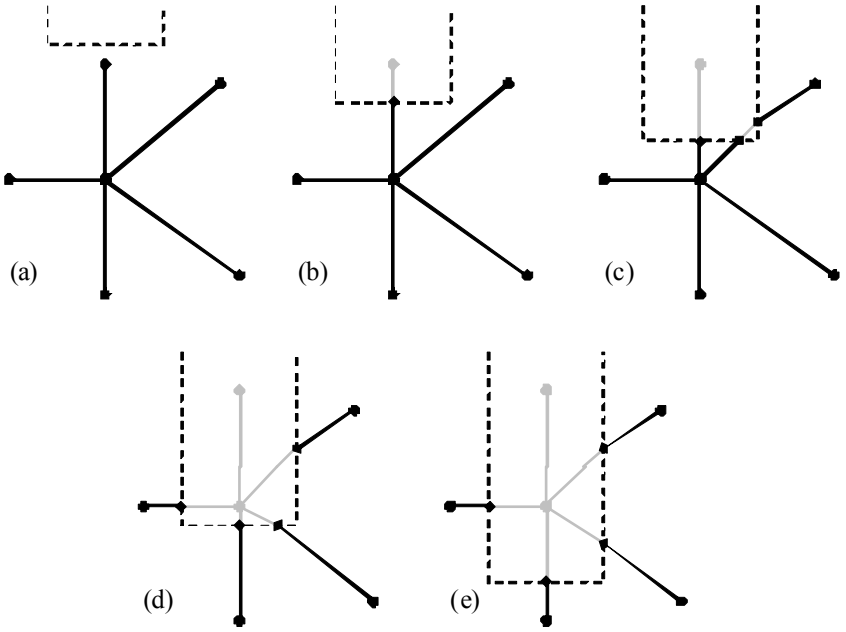


Fig. 7.7.5 The changes in a network as a cylindrical shaft is excavated. Inactive nodes and connections are greyed out.

Once the network has been pre-processed, it is relatively straightforward to update the network for any time required by the solver. As the solver steps forward in time, the activity of the nodes and connections is modified appropriately. The solver may then solve the problem for the active nodes and connections in the problem (see Sections 7.7 and 7.8). The total flux into the shaft is similarly calculated by summing the fluxes of the active (non-lined) connections that have been intercepted by the shaft. This sum is updated after every successful time-step.

For the solution methods used, the number of equations to be solved must be fixed. To prevent the total number of equations from changing as the nodes become inactive (there is one equation per node), the equations are modified on inactive nodes so that the head on inactive nodes is a constant. This does not affect the rest of the system, as the connections to the inactive node are also inactive.

7.7.5 Gradual Transitions

Prior to Version 3.3, spikes in the calculated excavation inflows occured when converging path legs become active (Fig. 7.7.6). In addition, the solver had difficultly in this regime – the sudden introduction of the two new connections with boundaries on the shaft replacing the old connection causes an abrupt transient response in the local flow system.

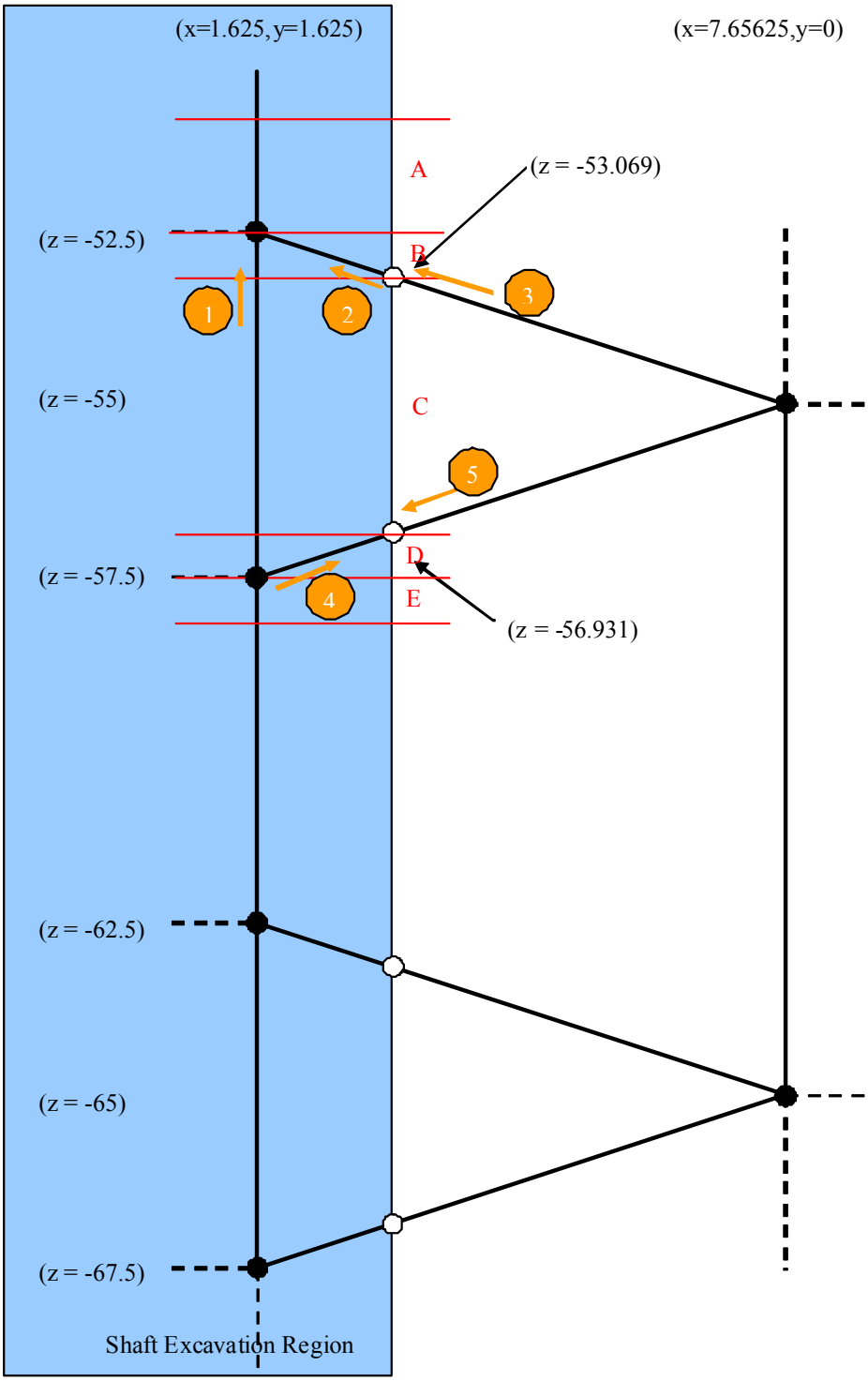


Fig. 7.7.6 Illustration of a flow network showing convergent path legs. The blue shaded area represents the shaft, and the orange numbers and arrows indicate the legs and flow directions of importance to understanding the occurrence of spikes. A, B, C, D and E are distinct regimnes for consideration of shaft inflows. Paths 4 and 5 are convergent.

This sudden change is an artefact caused by the way the system is discretised as a network. The connection that is intersected represents a volume of intact rock, and, therefore, the way

that the sudden change is modelled is unphysical because it implies excavation of a relatively long section of the shaft instantaneously. Representing a gradual change in a fully realistic fashion would be very difficult as the full details of the geometry are not stored during the model run (nodes have volumes and legs have areas but their precise spatial positions is not recorded).

Instead, from FracAffinity 3.3, we can represent the transition in an approximate way. We introduce a *Gradual Transition* while the excavation is in zone D (Fig. 7.7.6). The conductance of the existing internal connection (Path 5) is scaled down to zero, while the conductances of the new connections are scaled up from. The period over which this scaling is applied is the time that the excavation is in zone D.

This specific description of Gradual Transitions has been applied in the code in the following general sense:

- Information on shaft intersections is determined at the start of the calculation, and convergent legs that cross a shaft excavation are identified and tagged.
- The shaft excavation model defines intersections with legs and with nodes, and determines whether the shaft intersects the leg or a node first.
- If the shaft intersects the leg, then a Gradual Transition is used.

Note that Gradual Transitions are not used for discrete features, as the sudden intersection of these features by a shaft is more physically realistic. Also, sub-horizontal legs may still create a spike, since the vertical extent (and, therefore, the time) over which the Gradual Transition will be invoked will be small.

7.8 Borehole Sections

The treatment of boreholes in FracAffinity is separate from consideration of shafts and tunnels. Boreholes are considered to be of small radius and need not be purely vertical or horizontal. A borehole section is a hydraulically isolated section of a borehole and it is this that is the basis for the model in FracAffinity.

The basic geometry of a borehole section is a uniform radius cylindrical shape with centre-line made up from a sequence (maybe just one) of straight lines.

There are a number of steps involved in adding a borehole section to the FracAffinity networks. These are:

- Discretise the borehole section;
- Link the borehole section into the Intact Rock;
- Link the borehole section into the DDFs and SDFs;
- Assign properties to the borehole and link connections.

These steps are described in this Section. The boundary conditions that can be applied to a borehole section are described in Section 7.9.

7.8.1 Discretising a Borehole Section

Each borehole section is discretised by assigning nodes along its length. These are then connected together to allow flow along the borehole (unless the user specifies a non-conducting borehole in which case these connections are omitted). Before this is done, the borehole section is trimmed to be totally within the model region.

The user specifies a grid size which acts as a maximum separation between nodes. At least one node is created for each intact rock cell that the borehole section passes through (more if the cell has local refinement) and for each DDF and SDF that the borehole intersects. This is to ensure that the borehole section is properly linked to the rest of the system.

The connections along a borehole have the geometric area implied by the borehole radius and take their lengths from the nodal separations. The nodal volumes again come from the geometry, with the nodes for DDFs and SDFs having a small volume corresponding to the feature aperture times the borehole section area.

7.8.2 Linking a Borehole Section into the Intact Rock

Having discretised the borehole section, each node (except those at the end of the section) is connected to the Intact Rock, DDF or SDF. Recall that nodes were specifically added at DDF and SDF intersections. It is these nodes that connect to the relevant DDF or SDF, the rest connect to the intact rock (if there is any).

The basic approach is to link each node to four intact rock nodes. The aim is to link to the nearest surrounding nodes. For a simple geometry, with a vertical borehole and no local refinement, this is straightforward. Fig. 7.8.1 shows the links that would be added. Not that these links may not be single connections, see the later discussion.

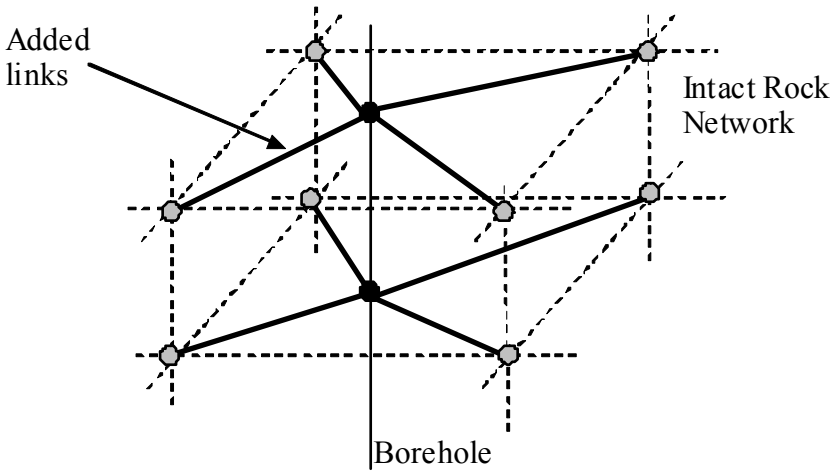


Fig. 7.8.1 The links for connecting borehole nodes to the Intact Rock Network for a simple situation.

To handle the general situation, we identify the four nodes to be linked to as being the nearest nodes in each of the four quadrants. The quadrants are selected according to the dominant direction of the borehole. For a vertical or near-vertical borehole section, the quadrants are: positive x, positive y; positive x, negative y; negative x positive y; and negative x, negative y. For borehole sections that are more nearly horizontal, quadrants defined by x and z or y and z are used.

Having decided on the nodes to link to, the links are split into a series of connections by inserting new nodes along their length. The user has control over the parameters for this. The first length, nearest the borehole, is specified. The growth factor (ratio of the length of one connection to the previous) is specified as is the maximum number of connections on a link. The maximum number takes priority is the given growth factor is too small.

There remains the question of assigning areas to these connections and volumes to the new nodes. This is addressed in Section 7.8.4. The hydraulic conductivity data is taken from the intact rock node to which the link is made.

7.8.3 Linking a Borehole Section into a DDF or SDF

The process of linking to a DDF or SDF is very similar to that described above. The borehole node will be placed at the intersection position. Generally, the search procedure described above, restricted to the nodes on the relevant discrete feature, will find the four surrounding nodes. If the borehole section intersection near the edge of a feature, it may be that there are no nodes in a particular direction – in such a case there are simply less than four links made.

7.8.4 Properties for the Borehole Section

For the connections within the borehole, the item not yet assigned is the hydraulic conductivity. This will be large compared to rock values. If Poiseuille flow in a tube is

assumed, then the effective hydraulic conductivity is given by $\frac{\rho g}{8\mu} r^2$ [17]. FracAffinity allows the user to specify the factor by which the square of the radius is multiplied. By default it is 1.25 x 10^6, which is obtained by taking typical values in SI units.
For a borehole of radius 3 cm, this give a hydraulic conductivity of 1125 m/s.
For the connections to the rest of the network, the hydraulic conductivity data is taken from the node to which the connection is made, since these legs represent the rock around the borehole (or the DDF or SDF around the borehole). In assigning areas to connections and volumes to the new nodes a number of things have to be considered.
For the volumes, these will be subtracted from the volume of the node to which the connection is made, to avoid double counting of volume. For the areas, it would be convenient is a suitable average area was found for each connection, since the geometric area increases with distance from the borehole. It is also important to recognise that the influence of the connecting node will increase if it is close to the borehole, and this is reflected by the fraction of the borehole circumference that is assigned to each link. Fig. 7.8.2 shows a typical situation.

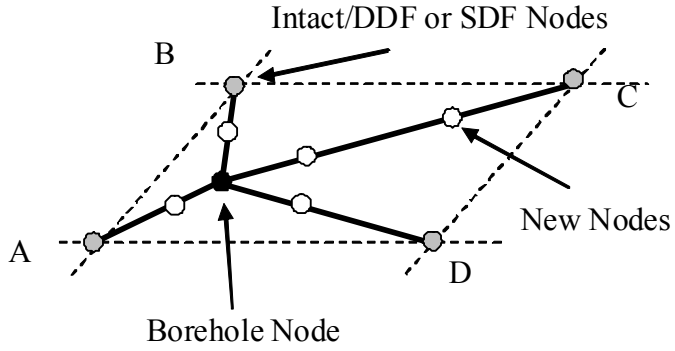


Fig. 7.8.2 Typical set of links for connecting a borehole node to the Intact Rock, DDF or SDF Network.

It is clear that connection A should be assigned more of the circumference than the others, with C having the least. The proportion will also depend on the volume of rock represented by each node. In FracAffinity, after some experimentation, the formula used for assigning fractions of the borehole circumference is that each link is assigned a fraction in proportion to the square root of its volume and the inverse square of its distance from the borehole.
Volumes for the nodes are then assigned from the geometry of the situation – taking account of the radial geometry and the length of borehole section represented. These volumes are deducted from the volume of the pre-existing node.

The areas assigned to the connections is a harmonic average of the geometric area along the connection (actually the two parts of the connection have separate averages). This can be justified on purely geometric grounds, but also ensures accurate representation of the pressure fall off away from a borehole.

Consider a long borehole with fixed head H_0 In steady-state the head falls logarithmically. In fact

$$H = H_0 - \frac{Q}{2\pi K} \ln \frac{r}{r_{bh}} \tag{7.8.1}$$

where Q is the flow (per unit length) from the borehole, K is hydraulic conductivity and r_{bh} is the borehole radius.

Now consider four equally spaced connections of length L from the borehole node representing a length B of the borehole. Each should take a quarter of the flow. In order to achieve this and have the correct head at the end of the connection, we need to find an area A such that

$$\frac{A}{L} \frac{Q}{2\pi K} \ln \frac{L}{r_{bh}} = \frac{QB}{4} \tag{7.8.2}$$

which implies that

$$A = B \frac{\pi}{2} \frac{L}{\ln L / r_{bh}} \tag{7.8.3}$$

Now the harmonic mean area is calculated as

$$\overline{A} = \frac{1}{\frac{1}{L} \int_{r_{bh}}^{L} \frac{2}{B\pi r} dr} \tag{7.8.4}$$

which can be evaluated and seen to be precisely the same as the area we deduced above in (7.8.3).

Thus, the formula (7.8.3) is used in FracAffinity, with the $\frac{\pi}{2}$ factor replaced by the appropriate fraction of 2π .

7.9 The Flow Equations

The previous sections have dealt with the implementation of complex geometries into networks composed of nodes, connected by "connections" of given properties. Once the physical models have been constructed, they are used in solving a system of equations that describe the groundwater flow. This section deals with the equations for saturated and partially saturated flow. The following section deals with the approach to solving these equations in FracAffinity.

The continuum governing equation for saturated flow can be written

$$\sigma \frac{\partial h}{\partial t} = \nabla.(K \nabla h) + q \tag{7.9.1}$$

where

h is the head (m)

σ is the specific storage coefficient (m^{-1})

t is the time (s)

K is the hydraulic conductivity (m/s)

q is a source of water ($m^3/s/m^3$)

For unsaturated flow there are various formulations. We choose the mixed from of Richard's equation, including storativity effects (when saturated)

$$\phi \frac{\partial s}{\partial t} + \sigma \frac{\partial p}{\partial t} = \nabla \cdot \left[K(p) \nabla \left(p + z \right) \right] + q \tag{7.9.2}$$

where

ϕ is the porosity,

s is the saturation,

p is pressure head (=h-z) (m),

$K(p)$ is the pressure (saturation) dependent hydraulic conductivity (m/s)

z is the coordinate in the vertical direction. (m).

The storativity term is not present when the saturation is not complete, which is when the pressure head is negative. To capture this behaviour it is convenient to introduce a new variable, ζ, and write

$$\begin{aligned} \zeta &= \phi s + \sigma p \quad \text{for } p \geq 0 \\ \zeta &= \phi s \quad \text{for } p < 0 \end{aligned} \tag{7.9.3}$$

so that

$$\frac{\partial \zeta}{\partial t} = \nabla \cdot \left[K(p) \nabla \left(p + z \right) \right] \tag{7.9.4}$$

The mixed form was used as it has the most suitable properties for numerical solutions [21]. These properties mean that numerical discretisations of the equation have good mass conservation properties. In each case, the steady-state version is obtained by requiring that the time derivatives are zero.

In FracAffinity, the domain is discretised as an unstructured network. The nodes of the network represent regions of space which have head and saturation values, while the connections between nodes represent flow paths. This network is equivalent to a finite-difference discretisation of the governing equations.

Thus, we reformulate the governing equations in the context of the FracAffinity network. The spatial derivatives are approximated by spatial differences (between nodes at the ends of a connection).

Thus the governing equation for saturated flow in the network is

$$\sigma_i \frac{\partial h_i}{\partial t} = \frac{1}{V_i} \sum_{\text{conn i-j}} A_{i,j} K_{i,j} (h_j - h_i) / d_{i,j} + q_i \tag{7.9.5}$$

where i and j are nodes. In terms of the conductance of the connections, this is written as

$$\sigma_i \frac{\partial h_i}{\partial t} = \frac{1}{V_i} \sum_{\text{conn i-j}} C_{i,j} (h_j - h_i) + q_i \tag{7.9.6}$$

The physical interpretation of this is clear. The right hand side represents the total water flowing into the node (per unit volume), which is balanced by the head increasing as determined by the storativity.

The steady state corresponds to the situation where all nodes have zero net inflow.

For unsaturated flow, the situation is very similar. The nodal equation is now

$$\phi_i \frac{\partial s_i}{\partial t} + \sigma_i \frac{\partial p_i}{\partial t} = \frac{1}{V_i} \sum_{\text{conn i-j}} C_{i,j} (p_i, p_j)(p_j - p_i + z_j - z_i) + q_i \tag{7.9.7}$$

but there are additional relationships between p and s and to derive the conductance given the nodal pressure heads (actually the dependence is on the saturation). These relationships can use the van Genuchten formulae or be user-specified and are discussed in the following section.

7.9.1 Water Retention Models

(1) Van Genuchten Relationship

The non-linear relationship between hydraulic conductivity and pressure head is controlled by two functions. The first relates the pressure head (usually referred to as suction for when the pressure head is negative) and the saturation or moisture content. The second relationship is that between the saturation and hydraulic conductivity. FracAffinity provides the ability to use the van Genuchten [10] model for these functions, as well as allowing the user to define their own functions as piece-wise continuous curves. For information on how the water retention models are defined see Chapter 8.

Van Genuchten [10], proposed the following analytic model,

$$s = \begin{cases} 1 & \text{for } p \geq 0, \\ \left[\dfrac{1}{1 + \left(A|p| \right)^B} \right]^C & \text{for } p < 0, \end{cases} \tag{7.9.8}$$

where A [L^{-1}] and B are fitting parameters, and $C = 1 - 1/B$ and

$$\frac{k(s)}{k_{sat}} = s^{\frac{1}{2}}\left\{1-\left(1-s^{\frac{1}{C}}\right)^{C}\right\}^{2}, \qquad (7.9.9)$$

The default value of B is taken to be 2 in FracAffinity but can be user-specified, and the parameter, A, is defined by the user (Fig. 7.9.1).

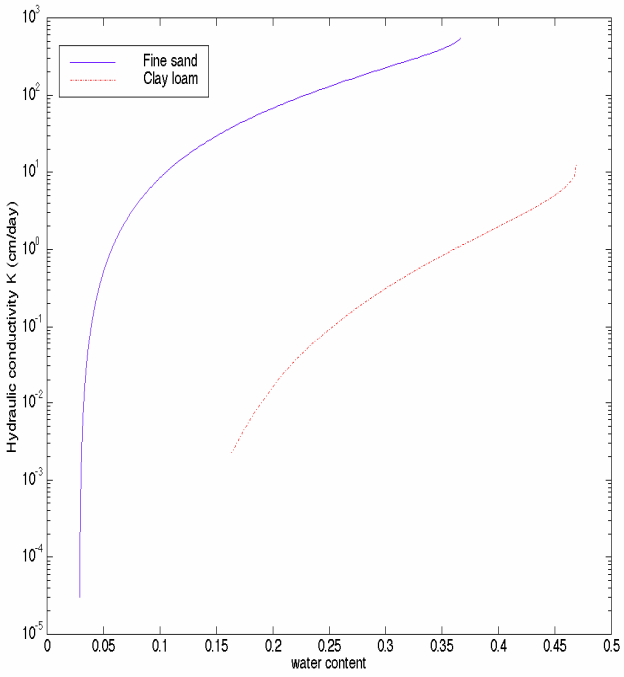


Fig. 7.9.1 Example plots of hydraulic conductivity against water content using the van Genuchten model.

In the network discretisation, it is necessary to derive a conductance for each connection given the pressure head, and hence saturation, at the end nodes. Given that the saturation implicitly varies between the nodes, the appropriate average is not obvious. In FracAffinity, we adopt the simple approach of taking the fully saturated conductance and scaling it by the harmonic mean of the scaling factors for the two nodal saturations.

(2) Modified Van Genuchten

Vogel, van Genuchten and Cislerova [11] discussed how the shape of the conductivity curve near saturation can strongly influence calculations. This was carried on by Ippisch, Vogel and Bastian [12] who also pointed out that improvements in the numerical behaviour is obtained when modified versions of the van Genuchten relationships are used. It is the numerical aspect that interests us here; discussion of true behaviour of the conductivity curve in deep rock is beyond the scope of the current project.

The modified version of the van Genuchten relations involves introduction of an Air Entry Pressure (AEP). We denote this pressure as $-P_e$, so P_e is specified as a positive value.
The van Genuchten relationships are used below this pressure, but are rescaled to given the required value at the air entry pressure.
So

$$s(P) = \begin{cases} 1 & P \geq -P_e \\ \dfrac{s_{VG}(P)}{s_{VG}(-P_e)} & P < -P_e \end{cases}, \tag{7.9.10}$$

and

$$k_{rel}(P) = \begin{cases} 1 & P \geq -P_e \\ \dfrac{k_{rel,VG}\left(s_{VG}(P)\right)}{k_{rel,VG}\left(s_{VG}(-P_e)\right)} & P < -P_e \end{cases} \tag{7.9.11}$$

This tends to make the gradients in the relationships less severe. When using this form, the storativity applies to pressures above $-P_e$.

7.9.2 Boundary Conditions

In addition to the governing equations, boundary conditions are needed to fully specify the system. There may also be internal sources or fixed head conditions and boreholes will have conditions attached. These are all discussed in this section.
Point and boundary conditions can be time-dependent through a specified piecewise-linear function. Except for the special case of borehole head conditions, discussed later, the functions of time are assumed constant outside the range of times given. Steady state calculations use the value at time zero. The time-dependent value can be discontinuous (if a time is repeated with a different value). The transient solvers identify this discontinuity and solve separately for the time intervals either side of it.
Boundary conditions can apply to a whole boundary or can be restricted by a polygon (for top and bottom boundaries). When there is a polygon specified, only boundary nodes with x- and y- coordinates within the polygon are given the condition.
When no boundary condition is given for a boundary node, a zero flux condition applies. This allows no flow across the boundary at this position.

(1) Point Source and Head Conditions

Point source conditions are attached to the node nearest the position specified. This may be a node on any feature or in the intact rock, and so this type of condition is rather unsatisfactory for normal use – the borehole section provides a better approach.
The specified flow rate (point sources are positive for flow into the model) is added to the water balance for the node (q_i in equations (7.9.5) and (7.9.7) is the flow divided by the volume of the node).

Point head conditions are also attached to the nearest node to the location specified. The equation for this node is replaced by setting the head as specified. For the steady-state calculations where the symmetry of the matrix is to be preserved, the entries for the connecting nodes are also changed. Note that specifying the head at a node allows flow in or out of the system. The water balance report (see Section 7.11) indicates the flow that arises.

(2) Boundary Head Conditions, Including Topographic

Boundary head conditions set the head at each node on the boundary. Two types exist: a head condition where all nodes have the same value (although it may be time-dependent); and a topographic condition where the nodal values are fixed for all time to the height (z-coordinate) of the surface above them. For the top surface, a topographic condition corresponds to atmospheric pressure at the surface and so is suitable for saturated cases. For side boundaries, the topographic condition sets all nodes in a column to the z-coordinate at the top. This allows flow through the boundary without imposing any vertical component.

(3) Boundary Flux Conditions

Boundary flux conditions set the flow at each node on the boundary. In FracAffinity a positive boundary flux is for flow out of the model region. The flux is specified as a flow per unit area, so the area of the connection to the boundary node is used to calculate the flow for each boundary node.

(4) Boundary Recharge Conditions

The boundary recharge condition acts differently in the saturated and unsaturated cases. In each case it is intended for use on the top surface.

For the saturated case, a recharge condition is the same as a topographic condition except that some additional flow is added. The recharge specified is the additional flux into the model (and so is scaled by the area of the connections). The way that this is implemented is illustrated in the Fig. 7.9.2.

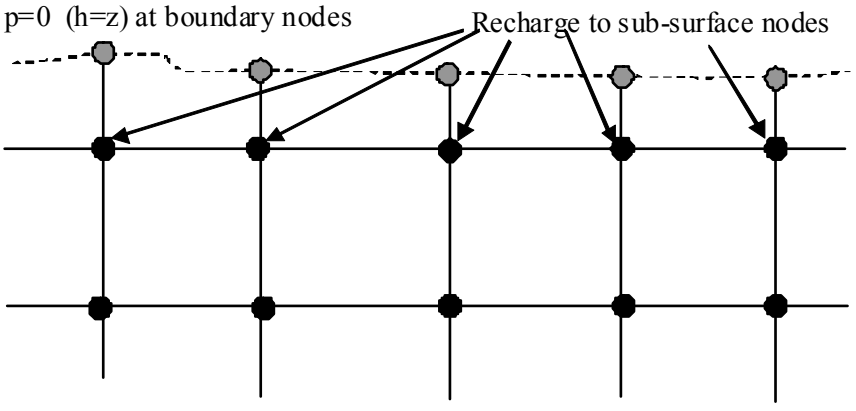


Fig. 7.9.2 Recharge for a saturated calculation.

Thus, the effect is that of having the topographic condition with a set of point sources just below the surface. The recharge may immediately flow out through the top surface and so is not guaranteed to add to the total flow in the deep system.
For an unsaturated case, the recharge acts as the maximum inflow that is allowed across the surface. The actual recharge depends on the local conditions. For recharge in unsaturated conditions, the maximum recharge that can be achieved is determined by the gravity term – corresponding to a condition of zero pressure-head gradient. Since this scales with the conductivity, the drier a boundary become, the less recharge it can take. The condition prevents the boundary being more saturated than the nearby interior. For saturated conditions, the boundary pressure head is limited to be zero, allowing a small inflow or an outflow. Thus the recharge condition can provide quite different behaviour from the equivalent flux condition, which does not depend on the local saturation state.

(5) Boundary Suction Conditions

A suction condition is simply a negative pressure condition for use in unsaturated cases. The value specified is positive, val, and is used to set a negative pressure, -val. A suction of zero corresponds to a topographic condition.

(6) Boundary Seepage Conditions

A seepage condition is intended for use on shaft walls in unsaturated cases although it can be used more widely. The condition switches between a no-flow condition and a fixed zero pressure condition according to the local saturation state of the rock. This follows an approach commonly adopted in groundwater modelling (see, for example, Anderson & Woessner [22]).
Where the rock next to the boundary is under-saturated the seepage condition act as a no-flow condition. This approximates the low flows that can occur once the rock is not fully saturated.
When the rock next to the boundary is saturated, the seepage condition acts as a fixed head condition, with a zero pressure value on the boundary. Note that this ensures that flow can only be from the rock into the shaft. An additional calibration parameter is used to scale the conductivity of the rock near the boundary (more specifically for FracAffinity, to scale the conductance of the leg attached to the boundary node). This parameter can be thought of as representing short length-scale effects near the boundary, such as desaturation or excavation damage.
With the introduction of the calibration factor, the flow equation used to solve the flow along the boundary legs (through the boundary nodes into the shaft) is written as follows:

$$Q = A \cdot \alpha \cdot K \frac{(H - h)}{L} \tag{7.9.12}$$

where:
Q – is the water flow through the boundary ($L^3 \cdot T^{-1}$)
A – is a surface area perpendicular to the flow (shaft wall) (L^2)
K – is the intact rock hydraulic conductivity ($L \cdot T^{-1}$)
H – is the prescribed head at the boundary node (=z) (L)

h – is the calculated head at the internal node connected to the boundary node (L)
L – is the distance between the boundary node and the internal connected node (L)
α – calibration factor (dimensionless)
The calibration parameter at a particular time is used to multiply the conductance of every leg from which flow enters the internal region. A value of less than 1.0 therefore reduces the flow, and can be used to represent wall effects caused by partial desaturation. A value greater than 1.0 could be used to represent the effect of a damaged zone. Since a single value is used for all legs regardless of their length or other properties, the calibration parameter should be regarded as a value that can be tuned to represent observations rather than something that can be calculated from a physical understanding.
The effect of using a seepage condition on a shaft will be to have no flow on the top part of the shaft, with flow into the shaft below some level. The level at which the flow starts is the local water-table level and it is this that the solver will be determining.

(7) Borehole Head Conditions

If a borehole head condition is set, all nodes in the borehole have the given head, which may vary in time. The exception to this is in the case of unsaturated flow when the MODEL_UNSAT_BH keyword has been specified.
In the normal case, the equations for the borehole nodes are replaced by assigning the given head. This allows water to flow in or out of the borehole, and this will be reported in the water balance reports, see Section 7.11.2.
If the MODEL_UNSAT_BH keyword has been specified in an unsaturated case, then the borehole is modelled differently if the top of it becomes unsaturated. In this case, only the heads in the saturated region are set, and the top of the borehole is modelled as an open system. In most cases this will give very similar behaviour to the standard approach. This is true as long as the borehole conductance is much greater than that of the surrounding rock.
The treatment of time-dependent borehole head conditions is different from the standard approach. In order to allow for controlled head pump tests to be modelled including a recovery phase, the head condition only applies between the first and last times specified. Outside these times the borehole is open. The flow solver treats these times as discontinuities.

(8) Borehole Flux Conditions

A borehole flux condition species flow into or out of the borehole. It is very similar in its effects to a point condition. In order to allow the water flows into the borehole to be properly calculated, the controlled flow is attached to the bottom node only. This node then has the appropriate source/sink term and the other nodes continue to be modelled as open.
Note that a borehole flux of zero does not isolate the borehole. It says that there is no net inflow or outflow. This is, of course, the normal situation and so a flux of zero corresponds to an open borehole. Because of this there is no need to employ the special treatment of the borehole head conditions.

7.9.3 Handling Non-linearity

(1) Pressure Transform

The highly non-linear dependence of the hydraulic conductivity, *K(p)*, means that the pressure solution will usually contain very sharp pressure gradients, especially near of saturated/desaturated boundaries. To address this, a pressure transform approach is used to try and reduce the steepness of the gradients in the solution variables.

By specifying the transformed variable, $f = f(p)$, which depends on the pressure head, in place of the pressure head itself and differentiating f,

$$\nabla f = \frac{\partial f}{\partial p} \nabla p \tag{7.9.13}$$

We can now replace the occurrence of ∇p in (9.2) to reveal,

$$\phi \frac{\partial s}{\partial t} + \sigma \frac{\partial p}{\partial t} = \nabla \cdot \left(D(p) \nabla f \right) + \frac{\partial K}{\partial z} \tag{7.9.14}$$

where $D(h) = (\partial f / \partial p)^{-1} K(p)$. The aim is to choose a suitable function $f(p)$ so that the non-linearities are easier to handle.

The transform used is based on the P_t-*based* transform, of Pan and Wierenga [23), 24)]:

$$f = P_t = \begin{cases} p, & p \geq 0, \\ \dfrac{p}{1 + \beta p}, & p < 0. \end{cases} \tag{7.9.15}$$

Here the parameter assumes values from –1 m^{-1} to –10 m^{-1}, although it is shown in [23)] that the performance of the P_t-*based* algorithm is fairly insensitive to the value of β, and a value of $\beta = -3 \mathrm{m}^{-1}$ is recommended.

It can be seen from this that *f* is constrained to be greater than 1/*β*.. However, the linear equations that are solved on each step can lead to values of *f* smaller (more negative) than this. The inverse transformation is simply

$$\begin{aligned} P &= f & f \geq 0 \\ P &= \frac{f}{1 - \beta f} & f < 0 \end{aligned} \tag{7.9.16}$$

and when this is applied to a value of *f* below 1/β it gives a positive value of *P*. This leads to convergence difficulties (although in some cases a false convergence is obtained with a negative *f* being returned because it is being wrongly associated with a saturated situation).

In order to correct this behaviour, the transform has been changed so that it uses a linear relationship for very negative *P* (and *f*) values:

$$\begin{aligned} f &= P & P \geq 0 \\ f &= \frac{P}{1+\beta P} & P < 0, \beta P < B \\ f &= \frac{P + B^2/\beta}{(1+B)^2} & P < 0, \beta P \geq B \end{aligned} \tag{7.9.17}$$

and the inverse:

$$\begin{aligned} P &= f & f \geq 0 \\ P &= \frac{f}{1-\beta f} & f < 0, \beta f < B/(1+B) \\ P &= f(1+B)^2 - B^2/\beta & f < 0, \beta f \geq B/(1+B) \end{aligned} \tag{7.9.18}$$

This form gives continuity of the transform value and derivative. The value of B is set to 100 in FracAffinity.

In terms of the nodal equations the transform gives a further scaling factor for the conductivity, somewhat countering the saturation dependence. This is combined with the saturation dependence and the harmonic average of the combination is used. Because the z-coordinate term does not have the transformed hydraulic conductivity, it retains the value accounted for the saturation effect alone.

Taking all of this into account the nodal equations that are solved are

$$\frac{\partial \zeta_i}{\partial t} = \frac{1}{V_i} \sum_{\text{conn i-j}} \mu_{i,j} C_{i,j} (f_j - f_i) + \nu_{i,j} C_{i,j} (z_j - z_i) + q_i \tag{7.9.19}$$

where

$$\frac{1}{\mu_{i,j}} = \frac{1}{2}\left(\left[\frac{\partial f}{\partial p} \frac{K_{sat}}{K(p)} \right]_i + \left[\frac{\partial f}{\partial p} \frac{K_{sat}}{K(p)} \right]_j \right), \tag{7.9.20}$$

$$\frac{1}{\nu_{i,j}} = \frac{1}{2}\left(\left[\frac{K_{sat}}{K(p)} \right]_i + \left[\frac{K_{sat}}{K(p)} \right]_j \right) \tag{7.9.21}$$

In practice, the pressure transform has often been found to be unhelpful in FracAffinity. The precise reason for this is not understood, but the current version has a default value of β of zero, disabling the transformation.

(2) Non-linear Fraction

An alternative approach to handling the highly non-linear dependence of the hydraulic conductivity, *K(p)*, is used in the steady-state unsaturated algorithms. There, a "non-liner fraction" is introduced to reduce the non-linearity. The idea is to gradually move to having the full non-linearity by solving a sequence of easier problem.

To do this, the conductivity used is modified to reduce the non-linearity. Let f_{nl} be the non-linear fraction, then we use

$$K(p;f_{nl}) = \frac{K_{sat}}{K_{red}}\left(f_{nl} + (1-f_{nl})K_{red}\right) \tag{7.9.22}$$

where K_{red} is the reduction factor for the full non-linear water retention model

$$K_{red} = \frac{K_{sat}}{K(p)}. \tag{7.9.23}$$

This reduces the non-linearity and constrains the minimum conductivity, for example, with a non-linear fraction of 0.98, the minimum conductivity is 1/50 of the saturated value.

Figures 7.9.3 and 7.9.4 show the effect of the non-linear fraction across the full range of saturation for a Van Genuchten and a user-defined model respectively. They show the relative conductivity which is the reciprocal of the reduction factor. A logarithmic scale is used to emphasis the effect. Each figure shows the full value and the 99%, 95%, 90% and 75% values (that is f_{nl} = 1.0, 0.99, 0.95, 0.9 and 0.75).

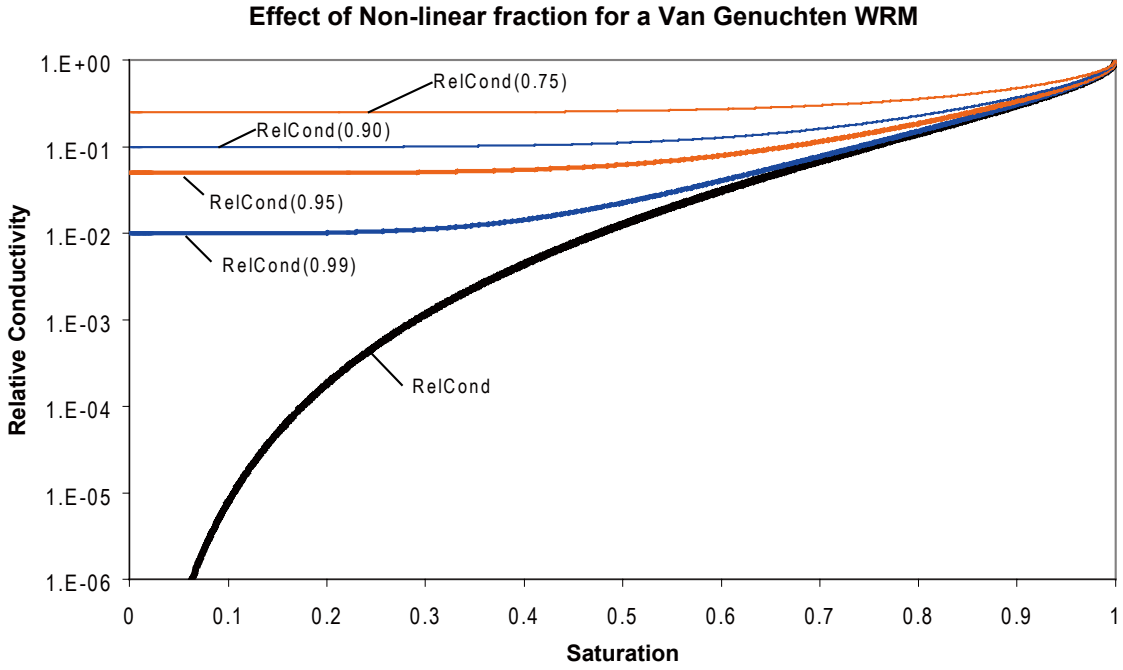


Fig. 7.9.3 The effect of the non-linear fraction on the relative conductivity for a typical Van Genuchten retention model (A=1, B=2).

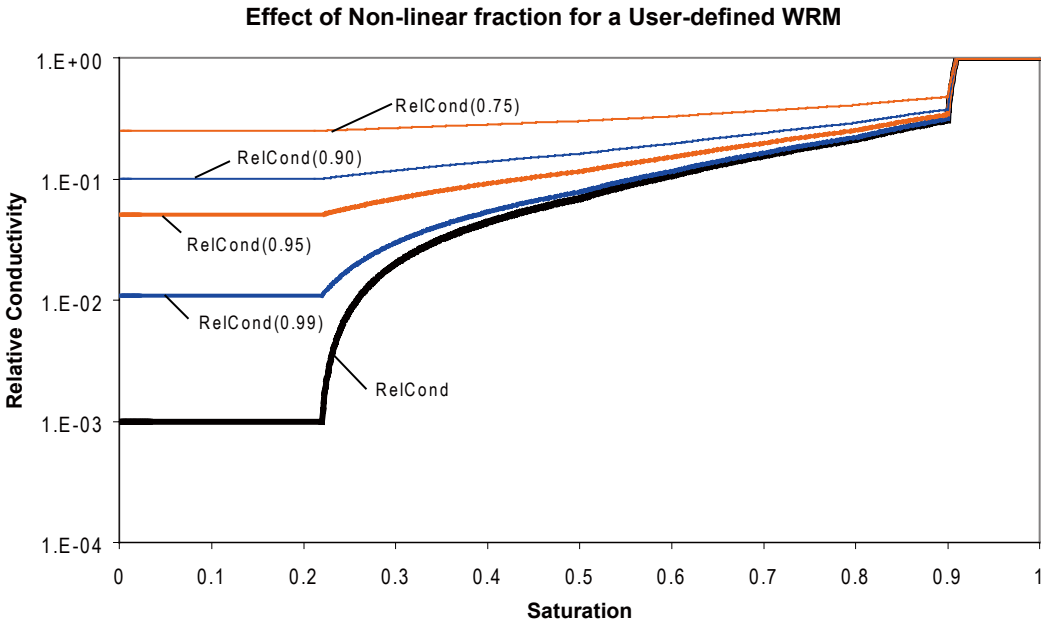


Fig. 7.9.4 The effect of the non-linear fraction on the relative conductivity for a typical user-defined retention model.

7.10 Solving the Flow Equations

In this section, the approaches used to solve the equations numerically are discussed.

7.10.1 Steady-state Flow

Steady-state flow calculations are useful in their own right, as a description of a system that is in equilibrium. They are also used in FracAffinity to obtain initial conditions for transient calculations.

The basic approach used for solving steady-state calculations in FracAffinity is at iterative one. An initial guess is gradually refined until the residuals (local mis-balanced flows) are small enough. The approaches used for saturated and unsaturated flow differ somewhat because the saturated flow system is linear whereas the unsaturated system is non-linear (because of the dependence of conductance on saturation).

In both cases, there is a need for a potentially large system of equations to be solved (equation 7.9.6 or 7.9.7 with zero time derivatives). For the saturated case, the linearity means that this requires solution of a single system of linear equations. For the unsaturated case, the iterative approach refines the equations to be solved, but at each step a linear system must be considered.

For saturated steady-state conditions the equations are symmetric, which allows more efficient algorithms to be used. The approach used in FracAffinity is discussed below.

(1) Iterative Solution of Symmetric Linear Equations

This sub-section describes the symmetric conjugate gradient approach that has been implemented. This has been shown to handle cases with well over 100,000 nodes in a few minutes of calculation time.

The basic approach taken is to use a standard pre-conditioned Conjugate Gradient (CG) algorithm, with the preconditioning matrix taken from a simple factorisation of the original matrix (not using any Lower/Upper Triangular (LU) or other decomposition). Because the pre-conditioning matrix is cheap to apply (the same cost as a matrix multiply), many iterations (several hundred) can be used.

It has been found that faster convergence is obtained if the Conjugate Gradient approach is augmented by some simple iterative refinement steps. These are undertaken after every 100 iterations of the conjugate gradient method.

The CG algorithm can be set out in various ways. The one given here is convenient as it does not require the pre-conditioning matrix to be split into symmetric factors.

The algorithm used to solve $Ax = b$ for a symmetric matrix A, given a symmetric preconditioning matrix M is given in Table. 7.10.1.

Table. 7.10.1 The sparse matrix algorithm to solve Ax = b.

Start with

$x = 0$

$r = b \ \ (= b - Ax)$

Iterate, for i=1,2,…

$z = M^{-1} r$

$\rho_{old} = \rho$

$\rho = r^T z$

if i=1 then

$p = z$

otherwise

$\beta = \rho / \rho_{old}$

$p = z + \beta p$

$q = Ap$

$\alpha = \rho / p^T q$

$x = x + \alpha p$

$r = r - \alpha q$

Loop until $\|r\| < \varepsilon \|b\|$

For the preconditioning matrix, we write

$$A = L + D + L^T \tag{7.10.1}$$

where D is block-diagonal with tridiagonal blocks, and L is strictly lower block-triangular. Then we choose M, as an approximation to A, to be

$$M = (L + D) D^{-1} (D + L^T) \tag{7.10.2}$$

The algorithm requires the application of the inverse of M, but this is simple, since

$$M^{-1} = (L^T + D)^{-1} D (D + L)^{-1} \tag{7.10.3}$$

which required inversion of a lower and an upper block triangular matrix and multiplication by the diagonal of the original matrix, all of which are cheap to do. The tri-diagonal nature of the blocks in D is important in this, since tri-diagonal matrices can be inverted cheaply.

The iterative refinement that is undertaken after every 100 steps is very simple. Using the same notation as above, is updates an approximation, x_n, to a new one, x_{n+1} by solving

$$Dx_{n+1} = b - (L + L^T)x_n \quad . \tag{7.10.4}$$

This is repeated 100 times. The combined effect of the conjugate gradient steps of the iterative refinement steps performs better than either of them separately.

The above scheme works quite well with small blocks (the previous versions essentially used a block-size of one), but is much more efficient if the blocks are as large as possible. It is also desirable for nodes that are strongly connected to be within the same block. The natural numbering of FracAffinity networks provides a satisfactory block structure. In particular, nodes in a borehole section have consecutive numbers. Nodes in the intact rock, DDFs and SDFs are numbered systematically and provide sensible blocks. These are split by the rewiring that occurs during the creation of the overall network, but remain large enough to be useful.

(2) Iterative Solution of the Non-linear Equations

When unsaturated flow is model, the conductances depend on the saturations, which in turn depend on the pressure head. In generic terms, this means that the system to be solved is now

$$A(x)x = b(x) \quad . \tag{7.10.5}$$

The approach used in FracAffinity is to solve this through a combination of Picard and Newton iterations. Picard iterations are cheaper than Newton but may not give full convergence.

A Picard step refines x_n to x_{n+1} by solving

$$A(x_n)x_{n+1} = b(x_n) \quad , \tag{7.10.6}$$

which is linear and symmetric in x_{n+1} and so can be solved in the same way as the saturated case. Because it is only part of a larger iterative scheme, the solution of the linear system need not be fully converged at each step. To enhance convergence, each step is modified slightly so that it generates a recommended change in x and then searches along that direction for the minimum residual (of the non-linear problem).

By default, a Newton step is performed every 4^{th} step or after a Picard step diverges. However, the user may select that only Newton steps are used or that all Newton steps are used after a few initial Picard steps.

A Newton step requires calculation of the derivatives of A and *b*. The new solution is given by

$$x_{n+1} = x_n - [b'(x_n) - A'(x_n)x_n - A(x_n)]^{-1}[b(x_n) - A(x_n)x_n] \quad . \tag{7.10.7}$$

The matrix arising here is non-symmetric, because of the derivative terms. The non-symmetric matrix inversion is handled using a the minimum residual iterative scheme utilising a non-symmetric version of the block preconditioner described for saturated flow..Again, full convergence is not necessary since this is only part of a larger iteration.

Problems with convergence in unsaturated regions can prevent overall convergence. In order to improve the convergence, nodes that are unsaturated are subjected to a local refinement step. This adjusts the pressure head (and saturation) to give a zero residual at the particular nodes.

7.10.2 Transient Flow

For transient flow situations, the storativity effects are most important in determining the timescales over which changes are propagated through the system. It is valuable to avoid very tiny storage coefficients since these give very rapid transients, but only if this is realistic for the model in question (for dynamic shaft excavation, where there are very large changes in pressure with excavation, it may be necessary to model all transients). FracAffinity has the capability to eliminate small transients by adjusting the storage coefficients to a user-defined time-scale, which must be much shorter than any timescale of interest.

(1) Rapid Transients

Equation (7.9.6) gives a natural time-scale for the response of each node in the system. This is important for the time-stepping algorithm. The time scale is simply given by the ratio of the coefficient of the time derivative and the nodal head:

$$\tau_i = \frac{\sigma_i V_i}{\sum_{\text{conn i-j}} C_{i,j}} \quad , \tag{7.10.8}$$

For systems where the nodal properties are highly variable it is possible that some nodes will have much shorter time-scales than others. This can adversely affect the behaviour of the time-stepping algorithm. It is found that the time-step can become unsatisfactorily small, while the transient behaviour of a single node is followed. There are two approaches that could be taken to this:

- Treat the rapidly changing node as being in equilibrium with its neighbours.
- Modify the nodal storativity to slow the transient down.

In FracAffinity, both approaches have been adopted. For nodes where a zero storativity has been set, the first approach is used. For nodes where the storativity is small but not zero, a user-specified time-scale can be used and the storativity is increased on nodes where this is violated.

(2) Time-stepping Solver

FracAffinity uses the DYLAN time-stepping solver for both saturated and unsaturated cases. DYLAN is a variable-order predictor-corrector method, based on the algorithm published by Byrne and Hindmarsh [13].

The general capabilities of DYLAN are presented in Appendix A.

FracAffinity uses both algebraic and differential variables. For the saturated case, the nodal heads are generally treated as differential, the exceptions being nodes that are controlled as point-conditions (where the head is a given function of time) and nodes where the storativity is zero (where the time-derivative is zero). For unsaturated flow, nodal vales for ζ and f (see Section 7.9) are calculated, with the ζ equations being differential and f being algebraic.

The inversion of the Jacobian matrix is handled by a general-purpose sparse matrix class. A partial LU decomposition is created and applied for each iteration of the Newton scheme that DYLAN uses.

7.11 Output Quantities

The main output quantities that FracAffinity reports are heads. These are in metres of water, as discussed previously. There are various other outputs produced, and this section describes what these correspond to. The outputs are:

- Velocities.
- Water Balance.
- Saturations.
- Flux plane information.
- Pathlines and particle clouds.
- SDF Data.

7.11.1 Velocities

Velocities are reported at nodes, and so are averages over the legs that connect to that node. Thus, the velocities calculated give a local approximation to a complete velocity field. The calculation that is made uses a flow-weighted average of the Darcy velocities. Thus at node I, the x-component of velocity that is reported is calculated from

$$\bar{v}_i^{(x)} = \frac{\sum_j \left| Q_{ij} \right| (x_j - x_i) / l_{ij}}{\sum_j \left| A_{ij} \right| (x_j - x_i) / l_{ij}} \tag{7.11.1}$$

with the y- and z-components calculated in a similar fashion. In this equation, Q_{ij} is a flow from i to j, l_{ij} is the distance from i to j and x_i is the x-coordinate of node i.

7.11.2 Water Balance

A water balance report is written to the FracAffinity log file for each monitor output time. This reports the instantaneous situation at the time. The format for the steady and transient and for the saturated and unsaturated cases are all basically the same, but some entries are not used in each case.

The total water in the system is reported, using the nodal volumes, porosities and saturation values. This is followed by a breakdown of water flows (volumes per unit time, m^3/s). This is presented by "boundary", which includes the external and internal physical boundaries but also includes storativity and resaturation effects. Where a physical boundary has more than one type of boundary condition, the different parts of the boundary are reported separately.

For each row of the table, the flow rate into the model and flow rate out of the model are presented separately. The net flow out is also reported. These flows are calculated simply by summing the flows in connections to the given boundary.

Note that for galleries, shafts and excavations, flow into a gallery, shaft or excavation is regarded as flow out of the model and is positive (in FracAffinity, flows into the model from boundaries are negative). For point-condition nodes, the specified flow is reported, or the flow that is calculated for a specified head condition.

For borehole sections, similarly, the specified flow is reported, or the flow that is calculated for a specified head condition.

For transient cases, the flows to and from storativity are reported. These are inferred from the non-zero net flows at each node. When point-condition and borehole conditions are applied, the flows will contribute to storativity initially – the rate at which this effect falls off will be determined form the storativity time-scales, either implied by the data or imposed by the user.
The final total would be zero in an exact calculation. For a steady-state calculation, a non-zero value indicates that the solution is not fully converged. For a transient calculation, the total should only differ from zero by rounding errors, since any local non-zero net flows are interpreted as being a storativity effect.

7.11.3 Saturation

Saturations reported by FracAffinity will be in the range zero to one. If the user-specified WRM file has a saturation value below one corresponding to zero pressure, then this will control the maximum saturation. For Van Genuchten, this maximum will be one.

7.11.4 Flux Plane Information

When a flux plane is specified, FracAffinity calculated the total flow across the plane.
Thus, all connections that cross the plane are included in full, with the sign of the flow being accounted for, so the reported value is the net flow rate.

7.11.5 Pathlines and Particle Clouds

Pathlines and particle clouds both use the implied flow velocities in the network to follow non-sorbed particles. The difference between them is in their behaviour at nodes. In both cases, a steady-state or transient flow field can be used. In the latter case, particle positions are updated after each timestep, as if flow field had been fixed for the previous step.
For a pathline, the particle follows the out-flowing connection with the largest flow from a particular node. Where two connections have the same flow, the connection chosen will be unpredictable.
For particle clouds, each particle selects randomly from the out-flowing connections, with the probability proportional to the flow rate. This corresponds to an assumption of complete mixing at each node.
A smoothed pathline can also be calculated as described in Section 7.11.5(2).
In either case, the particle track terminates when it arrives at a node with no out-flowing connections (usually a boundary node, a point-condition node or a borehole node).
The output files produced in pathline calculations provide detailed information about the path. If the keyword SUMMARY is included in the control file a simple summary output is produced. This keyword goes on the PATHLINE-POINT or PARTICLE-CLOUD header line.
The summary is simply a set of times and coordinates:

t, x, y, z

which is produced at each node and each leg mid-point.
The full output is produced at the same locations and reads:

t, x, y, z, ts, dist, len, v, Dv, flow, area, Mat, hcon, por, ap, cumF

where each item is as follows. See Fig. 7.11.1 also.

Item	Description
ts	is the time for the step [s] (=len/vel)
dist	is the total path-length so far [m]
len	is the length of the step (that lead to the given position) [m]
v	is pore-velocity for the step [m/s] (=flow/area)
Dv	is Darcy-velocity for the step [m/s]
flow	is the volumetric flow rate for the step [m^3/s]
area	is the cross-sectional area for the step [m^2]
Mat	is the material at location (as a name)
hcon	is the hydraulic conductivity for the step [m/s] (harmonic average)
por	is the porosity for the step [-] (average)
ap	is the aperture for the step [m] (0 if porous) (average)
cumF	Cumulative F-ratio. This is the time-integrated specific-surface-area (per unit volume of pore-water). Thus, for each step the cumulative F-ratio is increased by the travel-time divided by half the aperture. For porous legs there is no specific surface and so the cumulative F-coefficient is not changed. See further discussion below.

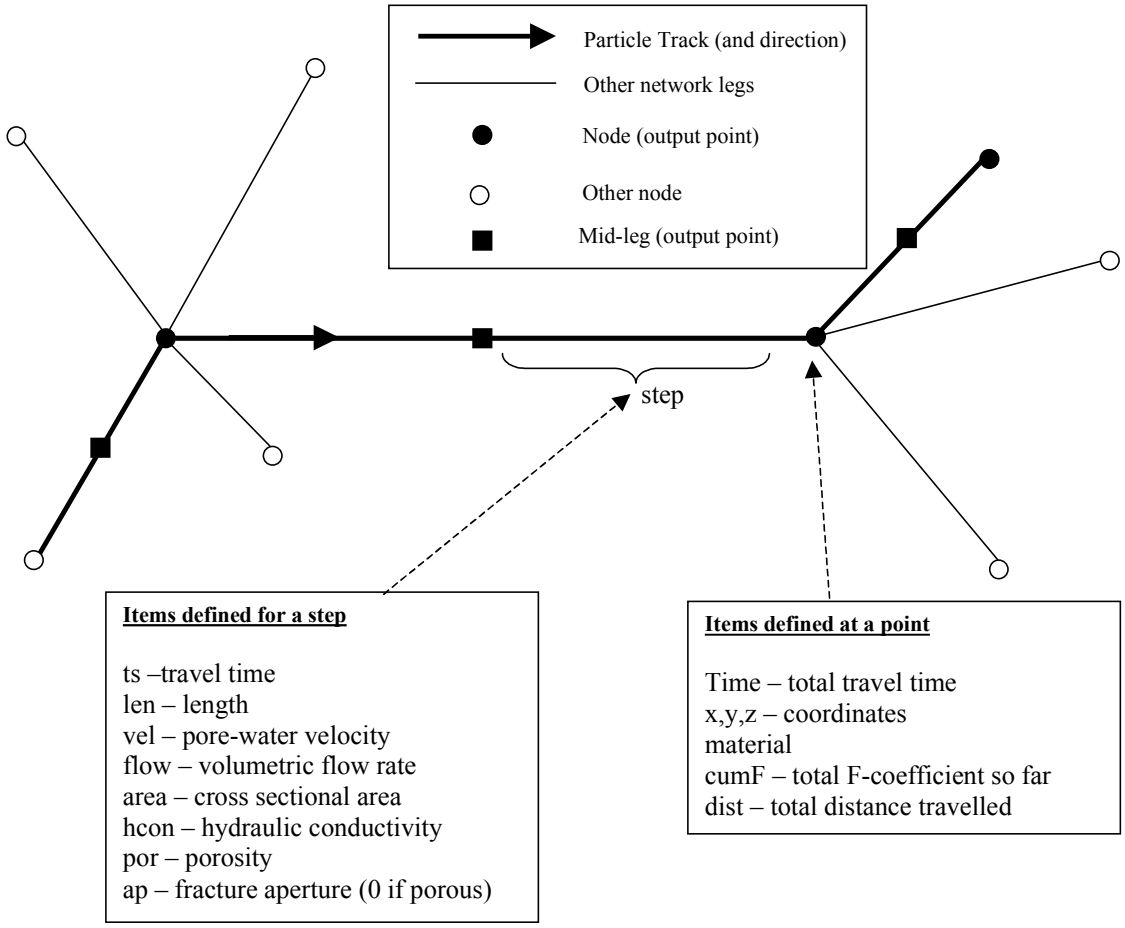


Fig. 7.11.1 Definition of items in the pathline file.

(1) Relationship between F-ratio and WL/Q

The F-ratio is equal to 2.WL/Q [25].

The WL/Q (transport resistance) used by Posiva arises from their analysis of transport through a fracture with no dispersion or sorption in the fracture, but with matrix diffusion (to an infinite matrix). It differs from SKI's F-ratio by a factor of 2 (F = 2. WL/Q). This factor arises because the F-ratio accounts for the surface area of both fracture surfaces.

The notation WL/Q can be misleading, as it has no explicit reference to the fracture properties. WL/Q could be calculated for a porous leg, but would be meaningless. For this reason, we prefer the F-ratio approach, which is explicitly define to be $a.t_w$ where a is the specific wet area for matrix diffusion and t_w is the water travel time.

For a fractured pathway, WL/Q can be simply recovered by dividing the cumulative F-ratio by two. For a porous path, the path length and travel time are the only pieces of information needed.

Note that "Q" here is the same as "flow" described earlier, although in the context of WL/Q, the Q is specifically referring to flow in a fracture channel.

Note that the files for EarthVision produced from the FAInterface produce just the time and position version of the pathline.

(2) Smooth Pathlines

The network-based solution approach in FracAffinity constrains what can be done with pathlines. There is no continuous velocity field available, and the flows in network legs must be the basis for any pathlines. Note that the derivation of a continuous velocity field could be contemplated in a homogeneous system with regularly oriented network legs, but the general FracAffinity network is highly disordered and heterogeneous.

The standard algorithm follows the strongest flow at each node on the path. There is a tendency for paths to continue to travel in one direction and then take a sharp turn when the dominant flow direction turns from, say, vertical to horizontal. A typical example is shown in Fig.7.11.2.

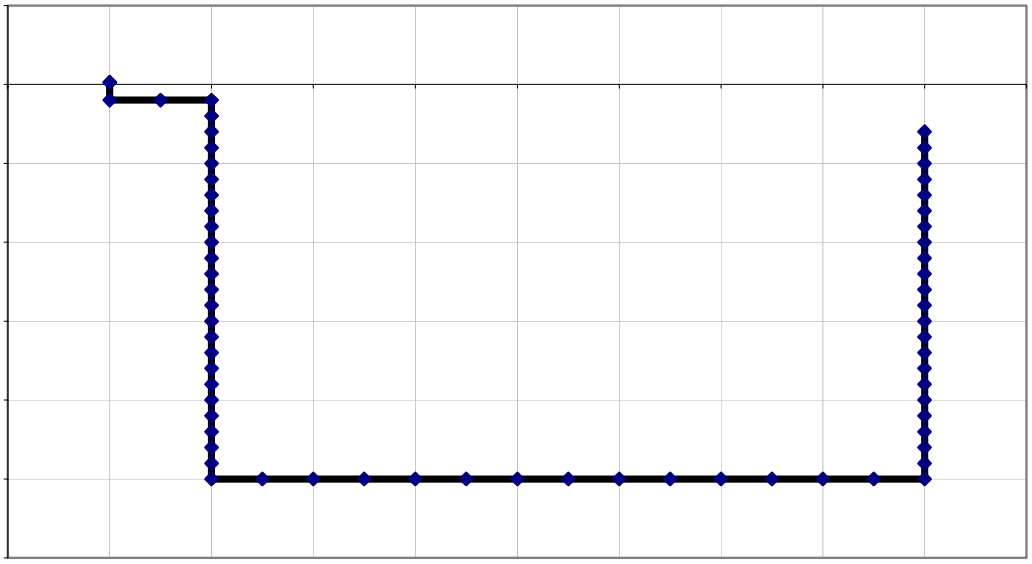


Fig. 7.11.2 Illustration of the behaviour of the standard pathline algorithm, in which the pathline follows the strongest flow at each node.

The path starts on the top right and runs vertically nearly to the bottom of the region before crossing and coming up on the other side. In this case there is only intact rock with a rectangular network to emphasise the problem.

It is possible in FracAffinity to calculate a local effective velocity at a node (see Section 7.11.1). This is used as the basis for the smooth pathline scheme. The basic approach behind the scheme is to use these nodal velocities to determine the direction along which the pathline moves, while allowing the pathline to move freely in space (i.e. not to be restricted to moving along network legs).

At each stage, n, the following information is stored:

- the true pathline position, $\boldsymbol{x}_n$;
- the nearby node, N_n;
- the current time, t.

The starting position is given by the user, and the nearby nodal position will be the nearest node to this. The initial time is also given by the user.
At each step, the following algorithm is applied:

1. find the nodal velocity at the nearby node, $\boldsymbol{v}_n$;
2. consider the pathline trajectory $\boldsymbol{x}_n + \delta t.\ \boldsymbol{v}_n$;
3. look at each downstream neighbour of N_n and determine how close the trajectory comes to it;
4. select the neighbour for which the trajectory comes closest;
5. set the pathline position to this closest point on the trajectory, set the nearby node to the selected neighbour and update the time by the necessary timestep.

The algorithm is terminated when the nearby node is a boundary node or when the pathline trajectory leaves the modelled region.
Essentially, this gives a "self-correcting" pathline. There is a path through the network, but the reported true pathline follows a local true velocity. In a simple case with a constant velocity direction the pathline follows the flow regardless to its orientation with respect to the network. In general, the accuracy of the new pathline can be expected to be of the order of the network spacing.
The pathline with the SMOOTH option for the case shown above is much more realistic. The path starts at the actual point requested (rather than at the nearest node) and takes a smooth path across the system (Fig.7.11.3).

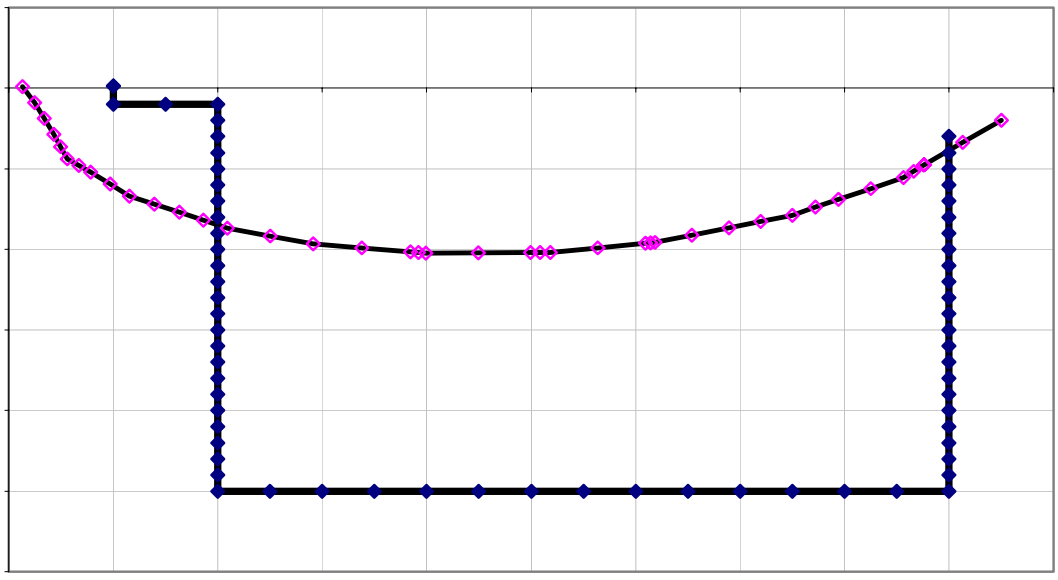


Fig.7.11.3 Comparison of the smooth pathline and the standard algorithm.

7.11.6 SDF Information

The SDF information that is written corresponds to the descriptions given earlier. Here we recap this information.

Coordinates of centroid: x, y, z	These are the coordinates used to generate the SDF, and may not be the centroid of the truncated shape.
Dip angle	In degrees below horizontal
Dip direction	In degrees from North
Rotation angle	In degrees
Length	The generated length of the SDF, i.e. the size of the SDF down dip.
Width	The generated width of the SDF, i.e. the size of the SDF across dip.
Hydraulic Conductivity	Log_{10} value
Porosity	
Tortuosity	Used to scale the conductivity
Storativity	
Aperture	
Anisotropy, in-plane and normal	Values for the SDF itself and for legs that cross it.
Enhancement Factor	Scaling factor for conductivity of enhanced intersections.

8. FracAffinity Version3.3 User Guide

8.1 Background Information

8.1.1 Overview of FracAffinity

FracAffinity is the flow simulator within the GEOMASS suite of applications. GEOMASS is an acronym for GEOlogical Modelling Analysis and Simulation Software. Within GEOMASS, EarthVision is used for the construction of geological models and for analysis and visualisation, and FracAffinity is used for constructing deterministic and/or stochastic models and for simulating flow. FracAffinity uses the concept of a hybrid medium. The flow solution is calculated on an irregular finite difference grid which explicitly represents the rock matrix (Intact Rock), mappable linear features within the rock mass (Deterministic Discrete Features, DDFs) and features which may not be mapped deterministically (Stochastic Discrete Features SDFs). Engineered features such as tunnels, shafts and boreholes can also be included. The operation and theory behind EarthVision is explained in the EarthVision User's Guide [4] and the background to FracAffinity is explained in the Chapter 7.

FracAffinity is accessed via an interface application (the FracAffinity Interface) and this chapter describes the use of the FracAffinity Interface (Version 3.3). The purpose of the FracAffinity Interface application is to allow the user to specify the data necessary to run the FracAffinity flow solver and to translate the output of FracAffinity into a form which can be used by EarthVision.

In the following sections we outline the system behaviour and the format of the ASCII input files which the FracAffinity Interface uses. The format of the files is intended to be user-friendly in that they should be easy to write and understand.

The user interaction with the FracAffinity Interface code is via specified data files and the command line. There are two main modes of operation. The first corresponds to setting up a problem for FracAffinity. This pre-processing mode is discussed in Sections 8.3 to 8.7. The second mode corresponds to processing FracAffinity output files for visualisation or analysis. The files are described in Section 8.8 and this post-processing mode is discussed in Section 8.9.

8.1.2 History of Changes since Version 1.0

This section summarises the changes made for each release. Note that Version 2.1 had many minor changes to the files that are required. Input files prepared for earlier versions will need to be updated for use with Version 2.1 or later.

A significant number of improvements were made in the transition between Version 1.0 and Version 1.2.4 of FracAffinity and the FracAffinity Interface. The modifications to FracAffinity from Version 1.0 include:

Improvements from 1.0 to 1.2, included the following:

- new orientation distributions for stochastic features,
- new size and shape distributions for stochastic features,
- new property distributions for stochastic features,
- improved behaviour at boundaries for all discrete features,
- improved testing for disconnected features,

- improved modelling of SDF intersections,
- improved flow solver performance,
- interface performance improved, and
- improved EarthVision exports.

Improvements from 1.2 to 1.2.4, included the following:

- the solver has been improved to handle a much larger number of nodes,
- polygons can be specified on the upper and lower boundaries to allow different regions of the boundaries to have different boundary conditions,
- the direct import of head boundary conditions from a previous model run, and
- the model network can be output as a file by the FracAffinity interface.

Improvements from 1.2.4 to 2.0, included the following:

- implementation of the capability to model unsaturated flow,
- implementation to model time dependent construction of a shaft,
- ability to refine grids in cuboid sub-regions of the model region, and
- an increase in the number of nodes in a FracAffinity model.

Additions in Version 2.0.1 were:

- recharge boundary conditions, for both saturated and unsaturated problems,
- user control over the setting of the Random Number Generator Seed and Storativity Time Scales.

Additions in Version 2.1 were

- enhanced flow on DDF and SDF intersections,
- extension of SDF orientation distributions,
- enhanced SDF output information,
- anisotropic conductivities,
- import conductivity fields,
- local refinement,
- modelling of boreholes,
- unnecessary information not prompted for,
- parameters are not time-dependent,
- boundary conditions time-dependence improved,
- save/restore options updated,
- water balance summaries are produced,
- point-by-point flows across the surface can be reported.

Additions in Version 2.2 were:

- additional information in pathline output,
- new general result output file gives geometry and solution information.

Additions in Version 2.3 were:

- enhanced performance on unsaturated cases, including updates to the recharge boundary condition and more solver options,
- boundary condition information included in general result file,
- improved estimation of nodal Darcy velocities,
- new options for pathlines.

Additions in Version 2.4 were:

- further improvements to the speed of the input and solver algorithms,
- seepage boundary condition,
- improved local refinement algorithm to properly localise the refinement,
- comment lines allowed in all input files,
- additional facilities to handle difficult non-linear cases,
- new shaft flow report,
- extended water balance report.

Additions in Version 2.4.1 were:

- property nodes bug fix,
- correction to FracAffinity interface information regarding fault polygons.

Additions in Version 2.4.2 were:

- allow multiple point conditions in point conditions file.

Additions in Version 3.0[2] were:

- correction to association between legs and DDFs that could result in DDFs being linked to incorrect boundaries;
- change to gridding algorithms for Intact Rock, incorporating thin layers and refinement near water table;
- fractal interpolation and interpolation from a file now on nodal rather than cell basis, allowing locally refined nodes to have different properties.

Additions in Version 3.0.1 were:

- correction to thin layer algorithm.

Additions in Version 3.1, 3.1.1 and 3.1.2 were:

- improvements to the transient flow solver,
- extension of dynamic shaft modelling to allow for multiple shafts,
- changes to specification of shafts and excavations,
- date and time now reported at start and end of a run,
- DDFs linked to shafts and excavations even when gridding is coarse,
- allowance for rescaling conductivities in a cuboidal region.

Additions in Version 3.1.3 were:

- allowance for rescaling conductivities in multiple cuboidal regions.

FracAffinity Versions 3.1.4 – 3.1.9 corrected some problems:

- smoothed pathlines are now properly truncated at the model boundary,
- the identification of the stratigraphic region for a node is now more accurate.

Additions in Version 3.2 were:

- extensions of dynamic excavation to allow general orientations and systems of excavations,
- additional option for local refinement near an excavation,
- ability to define dynamic conductivity scaling near excavations.

Additions in Version 3.3 were:

- introduction of Gradual Transitions to model the impact of converging legs in excavbations,

---

[2] Beta versions of, and draft documentation for, Version 3.0 were initially called Version 2.5.

- modification of the Dylan solver to provide faster performance,
    - introduction of the Air Entry Pressure to the van Genuchten relationship,
    - restarting the solver when nodes become inactive,
    - modified Newton iteration scheme,
- faster reading of initial conditions file,
- reporting flows to 8 significant figures in fluxshaft.fa file,
- reporting of total flow at each time in the shaftoutflow.faout file,
- intersection of intact rock grid and DDF grid no longer suffers from rounding problems.

8.1.3 Installation and Requirements of FracAffinity

Two executable programmes are required for FracAffinity. These are:

- the FracAffinity Interface which is used to input and output all data from FracAffinity,
- the FracAffinity application which is run automatically by the FracAffinity Interface.

FracAffinity may be loaded on any user defined location on the hard disk. To begin using FracAffinity first the environment variable FRAC_AFFINITY_PATH has to be set to provide the location of the FracAffinity executable for the FracAffinity Interface. If the FRAC_AFFINITY_PATH is not set then the interface will also search in the user's "path". This is controlled by the path environment variable.

8.2 Pre-Processing: Background

The user interaction with the FracAffinity Interface in pre-processing mode is via specified data files, and the command line.

Detailed control of the behaviour of FracAffinity Interface is through input files and/or in response to system prompts. No naming convention is imposed for the input files. However, a naming convention for files is suggested in the tables in this report which outline the file formats. The proposed format follows the following convention:

ProjectName_FileType.fai

This means that user is requested to provide the user defined file names in response to system prompts. The formats of all the input files required by the FracAffinity Interface are described in Sections 8.3 to 8.7. Section 8.8 lists all the output files that are produced. Since these correspond to FracAffinity input files knowledge of their format is not required by the user and they are not described here. It is not necessary for the user to construct or manipulate these files directly.

After a successful run and to speed up the process of preparing further calculations (during a sensitivity analysis for instance), a batch file can be prepared which can be directed as input to the FracAffinity Interface.

8.2.1 Overview of Input information

The order in which the Interface reads in information is:

- read in FracAffinity directory,
- read in stratigraphy,
- read in model region,
- specify flow solver controls,
- read in physical properties (depends on solver type),
  - intact rock parameters (K, θ, S and water retention models) for each formation,
  - parameters (geometry, K, θ, S and water retention models) for each DDF, and
  - parameters for SDFs in each formation,
- read in initial conditions, boundary conditions and point conditions, and
- specify outputs.

A full list of the prompts that will be encountered is shown in Table. 8.2.1.

Table. 8.2.1 Prompts produced by the FracAffinity Interface in input mode.

Prompt Number	Prompt	Response	Action
1	Name FracAffinity directory	Directory name	if directory is new go to 2
1.1	Do you want to overwrite existing directory [y/n]?	Y/N	
2	Name Stratigraphy sequence file	File name	

3	Name model region file	File name	
4	Name flow solver control file	File name	determines if flow, if transient and if unsat
5	Do you wish to use an existing hybrid medium file [y/n]?	Y/N	If 'N' go to 6
5.1	Name hybrid medium file	File name	Go to 10
6	Do you wish to include Intact Rock [y/n]?	Y/N	If 'N' go to 7
6.1	Do you wish to use an existing intact rock file [y/n]?	Y/N	If 'N' go to 6.3
6.2	Name intact rock model file (give full path)	File name	Go to 7
6.3	Name rock conductivity file	File name	
6.4	Name rock porosity file	File name	
6.5 (transient)	Name rock specific storage file	File name	
6.6 (unsat)	Name water retention model file	File name	
6.7	Name intact rock grid parameters file	File name	
6.8	Do you wish to save the intact rock model to a file [y/n]	Y/N	If 'N' go to 7
6.9	Name intact rock save file	File name	
7	Do you wish to add Deterministic Discrete Features [y/n]?	Y/N	If 'N' go to 8
7.1	Name a ddf Geometry file	File name	
7.2	Name ddf conductivity file	File name	
7.3	Name ddf porosity file	File name	
7.4 (transient)	Name ddf specific storage file	File name	
7.5 (unsat)	Name water retention model file	File name	
7.6	Name ddf grid parameters file	File name	
7.7	Do you wish to save the ddf rock model to a file [y/n]?	Y/N	If 'N' go to 8
7.8	Name ddf save file	File name	
8	Do you wish to add Stochastic Discrete Features [y/n]?	Y/N	If 'N' go to 9

8.1	Name a SDF Properties file	File name	
8.2 (unsat)	Name water retention model file	File name	
8.3	Name sdf grid file		
8.4	Do you wish to save the sdf rock model to a file [y/n]?	Y/N	If 'N' go to 9
8.5	Name sdf save file	File name	
9	Do you wish to save the hybrid rock model to a file [y/n]	Y/N	If 'N' go to 10
9.1	Name hybrid rock save file (this is an ASCII file – otherwise the default binary file properties.faout is written)	File name	
10 (transient)	Name Initial Conditions file	File name	
11 (flow)	Name Boundary Conditions file	File name	
12 (flow)	Name Point Conditions file	File name or dot (.) if none required	
13 (flow)	Name Output Definitions file	File name	
14	INPUT COMPLETE		

8.2.2 Units and Model Times

Before discussing the details of the input data required to run FracAffinity this section briefly discuses the use of units and time within FracAffinity. These two elements need to be considered when preparing input files. FracAffinity input files can use whatever standard set of units the user wishes to use. However, it is recommended that data is entered in SI units. Therefore, throughout this user guide the units for each parameter are provided as SI units. The conversion of units should be carried out prior to entering data into the input files, since computation is carried out assuming that units are consistent throughout.

FracAffinity uses ASCII format files for input. For structural information derived from EarthVision data are entered as column ordered fixed or free format. Therefore the user does not need to be concerned with the structure of grids. However, it is important that all z information is added as elevation and not depth.

The times for the model runs are entered in the flow solver control file. Times are also required when specifying time-dependent boundary conditions. If these are not within the times over which the calculation is made then they will be ignored.

8.2.3 Starting the FracAffinity Interface

The FracAffinity Interface is started for pre-processing by typing *fainterface -input* at the command line when within the required FracAffinity Interface directory. This directory will usually be the one containing the stratigraphic information (i.e. the sequence file and related surface files) exported from EarthVision.

The user is first prompted for a directory name:

```
==================================
FracAffinity Interface
Version 3.2
November 2006
==================================
Name FracAffinity directory :
```

On successful reading of this name, a sub-directory (referred to here as the FracAffinity directory) of the local directory will be opened with this name. If the directory already exists then the user is asked whether the existing directory should be overwritten or not. All FracAffinity input files are written by the FracAffinity Interface to this sub-directory. All the FracAffinity Interface input files should either be placed in the local directory, or full pathnames given in response to prompts for file names. The recommended directory structure is shown in Fig. 8.2.1.

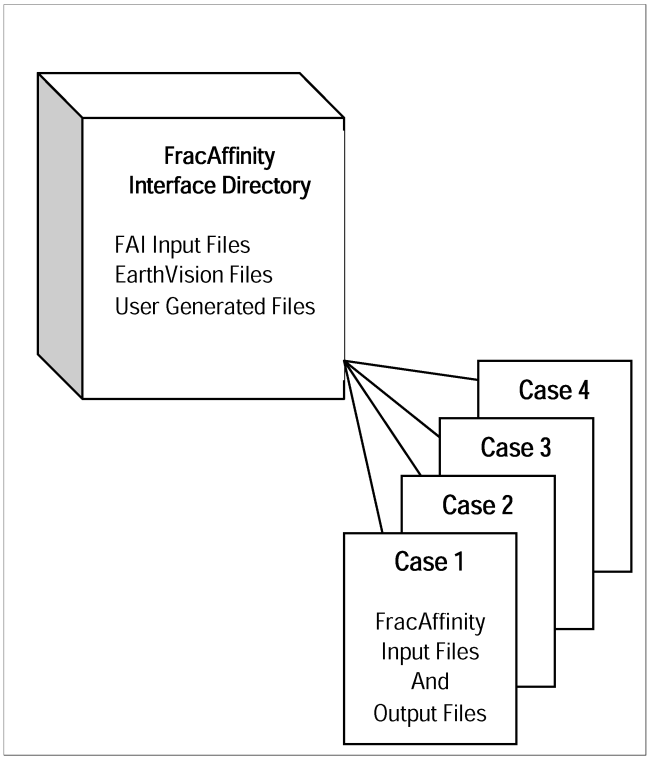


Fig. 8.2.1 Directory structure for FracAffinity.

The files which are required for the FracAffinity Interface and their formats are described in the following sections (Sections 8.3 to 8.7). In defining the formats the following conventions are used. Keywords that have to be specified are in UPRIGHT font, while user-specified values are in italic font. Optional entries are indicated by square brackets [ ]. Items should be separated with spaces; tab characters should be avoided as they are not always accepted by FracAffinity.

Comment lines can be included in any of the input files. This is achieved by using lines with a # character in the first column. FracAffinity then skips the whole line.

If an error occurs at any stage, such as a specified file not being found, the interface will ask how to continue, with the prompt:

"Try again or Quit [t/Q]?"

Entering "Q" followed by return will cause the interface to exit. Any other entry will allow the user to continue and correct the previous problem (for example by entering a new file name).

Note that when the interface is being driven from a batch file, an error will cause the code to lose its place in the inputs and subsequent errors may occur. The user should refer back to the original error.

8.3 Model Framework, Geometry and Flow Solver

8.3.1 Geological Framework

After the FracAffinity directory has been opened, the interface will prompt the user to

Name Stratigraphy sequence file:

Information on the geological framework of a region containing the FracAffinity model region is defined in three types of file (defined in Table. 8.3.1-8.3.3). The first file, the sequence file (Table. 8.3.1), defines fault blocks bounded by faults (which may also be DDFs) and the stratigraphic sequences in each fault block.

The format of these files corresponds to those produced by the FAExport macro, which uses sequence files from EarthVision Version 4 or EarthVision 5. The use of FAExport is covered in Appendix B of this document. FAExport only copies the relevant data from the EarthVision sequence file and makes copies of the files that are needed.

EarthVision sequence files can also be read directly. To allow this, the following applies when reading the FAInterface sequence file

- All lines starting with # are treated as comments.
- Blank lines are ignored.
- Keywords are case-insensitive.
- Open brackets ('{') are ignored.
- Close brackets ('}') are treated as 'END' keywords.
- Unrecognised keywords are ignored, including any block enclosed in curly brackets.

The GEOMETRY data specifies the maximum and minimum values in the model region (x_min, x_max, y_min, y_max, z_min, z_max). Each Zone is named and is given an Operation type (either Depositional or Unconformity) which controls how stratigraphic surfaces bounding the zones interact. Note that the zones refer to geological formations, which may appear in several Fault Blocks (although with the same properties in each).

There are two types of information for the fault blocks that form the model region (see Fig. 8.3.1). The first type specifies the fault block name, the name of the fault that divides it, the file containing the dividing fault surface, the bounding polygon and the names of the faults block above (hanging wall fault block) and below (footwall fault block) the dividing fault. This fault block information is naturally hierarchical.

The second type of information concerns the sequence of stratigraphic zones within each fault block. The list of names following the Sequence keyword defines the names of the p stratigraphic zones in the fault block, and the names of the files containing the data specifying the upper surfaces of the zone. The zones are listed from the lowest zone upwards. Note that p may be different for different fault blocks.

Table. 8.3.1 The format of a FAInterface sequence file. Note the use of '' '' symbols around fault block and sequence names is optional.

Name	Determined by FAExport/EarthVision - *name.seq.fai*

Format	GEOMETRY
	X x_min x_max
	Y y_min y_max
	Z z_min z_max
	END
	
	ZONE ''zone_1'' {
	OPERATION "zone_type"
	}
	… (repeat for each zone).
	FAULTBLOCK "block_name_1"
	FAULT "fault_name"
	FAULTFILE fault_filename
	[BOUNDINGPOLYGON poly_filename]
	ABOVEBLOCK above_block_name
	BELOWBLOCK below_block_name
	END
	… (repeat for each fault-type fault block).
	
	FAULTBLOCK "block_name_n-1"
	SEQUENCE
	"zone_n_1" surfacefilename_1
	…….
	"zone_n_p" surface_filename_p
	END
	END
	… (repeat for each sequence-type fault block).

Table. 8.3.2 The format of a FracAffinity input surface file.

Name	User-defined
Format	. …. .. Grid_size: *x_size* x *y_size* .. Grid_X_range: *xmin* to *x_max* .. Grid_Y_range: *ymin* to *y_max* . …. *x_1 y_1 z_1* *x_2 y_2 z_2* …. *x_N y_N z_N*

Table. 8.3.3 The format of a FracAffinity input polygon file.

Name	User-defined
Format	…. POLYGON *x_1 y_1* *x_2 y_2* …. *x_N y_N* *x_1 y_1*

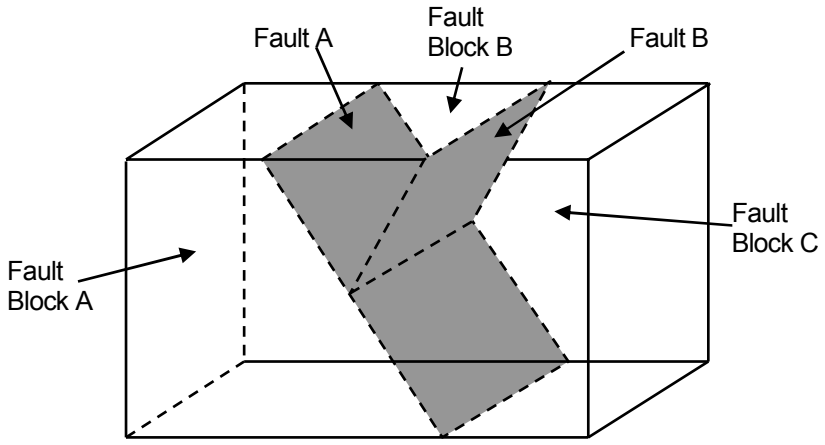


Fig. 8.3.1 A simple fault block decomposition of a region.

The sequence file (Table. 8.3.1) refers to additional files that specify fault grids (which may also be DDFs) which divide fault blocks, files that specify polygons defining the limits of faults and files that specify the top of geological zones (stratigraphic formations or lithological zones). The format of the surface files (fault_filename_1 and surface_filename_n) is detailed in Table. 8.3.2. The surface file includes the location of each point on the surface, defined on a regular grid, together with an index. Activity data may be included as an extra item in an EarthVision surface file, but is not used by FracAffinity. The format of the polygon files (polygon_filename) is detailed in Table. 8.3.3. Note that the last data point is the same as the first. If any points are duplicated then the duplicates are ignored.

Once the geological framework sequence file has been read in and checked for consistency, the location of the sequence file and associated surface files is stored for later use by FracAffinity. It is therefore important that once the sequence and surface files have been checked by the interface, they are not moved or deleted, otherwise the FracAffinity solver will be unable to locate them and fail to run correctly.

While the sequence file is being read, the interface will write information to the screen about the sequence file, finally listing the names all of the formations which are present in the stratigraphy. An example output is given below:

Name Stratigraphy sequence file : local_49x26.seq

Reading geometry information in stratigraphy

Name Stratigraphy sequence file : local_49x26.seq

Reading geometry information in stratigraphy

Reading Parent Fault Block ALL

Reading Fault Tsukiyoshi from Tsukiyoshi_v1.fa

Reading Base Fault Block HWtsu

Reading TokiGranite from TokiGraniteHWtsu.fa

Reading MizunamiGroup from MizunamiGroupHWtsu.fa

Reading SetoGroup from SetoGroupHWtsu.fa

Reading Base Fault Block FWtsu

Reading TokiGranite from TokiGraniteFWtsu.fa

Reading MizunamiGroup from MizunamiGroupFWtsu.fa

Reading SetoGroup from SetoGroupFWtsu.fa

Read fault block information in stratigraphy

Constructed fault block hierarchy in stratigraphy

Constructed formation information in stratigraphy

Completed checking of stratigraphy

Stratigraphy files have been read successfully

Formations in stratigraphy

0: TokiGranite

1: MizunamiGroup

2: SetoGroup

The interface will now ask for information about the model region.

8.3.2 Model Region Information

The Model Region defines the external boundaries of the region (see Fig. 8.3.2) that is to be modelled by FracAffinity. It is composed of a polygonal boundary defined in the x-y plane, and is bounded above and below (in the z direction) by surfaces. Therefore, the sides of the model region are vertical The model region may also include internal boundaries such as those modelling Shafts or Galleries, information on boreholes, and the excavation timetable for time dependent shafts. Regions that are to have their conductivity rescaled may also be specified.

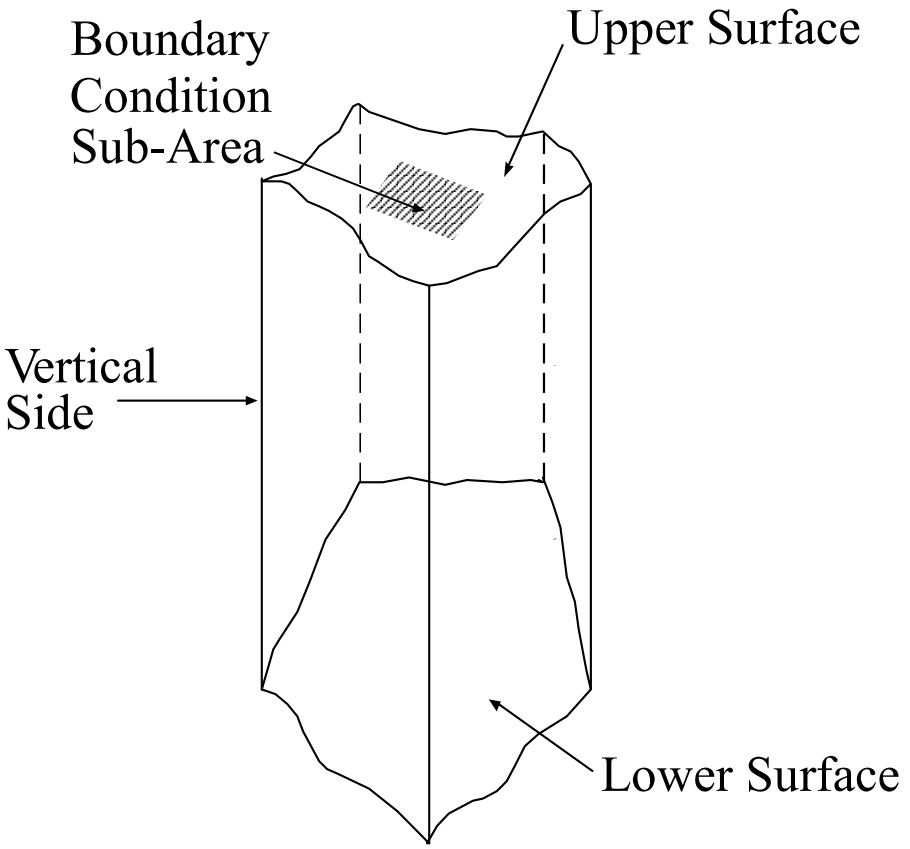


Fig. 8.3.2 An example model Region, consisting of five vertical planar sides with variable elevation upper and lower surfaces. Also shown is a shaded sub-region where a local boundary condition may be applied.

After reading the stratigraphy information the interface will prompt

"Name model region file :"

and the user should supply the name of a file with the format shown in Table. 8.3.4.

Table. 8.3.4 The format of the model region input file.

Name	User-defined (Suggested Name: *projectname*_model_region.fai)
Format	MODEL_REGION *region_name* TOP *name_upper.fai* BOTTOM *name_lower.fai* FACES *name_1 x_1 y_1* *name_2 x_2 y_2* …….

	name_N x_N y_N [GALLERY *gallery_name x_gs y_gs x_ge y_ge z_g radius*] [SHAFT *shaft_name x_s y_s z_top z_bottom radius*] [SHAFT *shaft_name x_s y_s DEPTH depth radius*] [SHAFT *shaft_name x_s y_s ELEVATION z_bottom radius*] [Excavation *groupname filename*] [ExcavationModel *name filename*] *(old style input)* [BOREHOLES *filename*] [ConductivityScale *filename*]

Example input file:

```
MODEL_REGION Tono_Sub_Region
TOP
TonoSR_upper.fai
BOTTOM
TonoSR_lower.fai
FACES
1 4172.30          -70938
2 5127.62          -70602.5341204
3 6498.41          -71272.3295122
4 7672.41          -69345.5272677
5 7820.39          -67804.4497947
6 7148.18          -66566.2441597
7 4890.34          -65481.9130847
8 3643.06          -66490.8476172
9 3699.01          -68628.978669
GALLERY Gallery1 4993 -68323 4750 -68289 135 1.25
GALLERY Gallery2 4993 -68168 4993 -68323 135 1.25
GALLERY Gallery3 4819.7 -68245 4819.7 -68289 135 1.25
SHAFT Shaft_One 4750.4 -68289 260.19 135    2
SHAFT Shaft_Two 4819.7 -68245 270.49 135    2
Excavation    TonoURL    TonoURL.fai
ConductivityScale CondScale.fai
```

Comments may be inserted at the beginning of this file, on lines starting with a # character.
The initial line of the file has the keyword MODEL_REGION which allows the user to specify a unique name for the model that is being run. This name is user defined.
In the model region file name_upper.fai and name_lower.fai are the names of surface files, of the same format as in Table 8.3.2, defining the model region upper and lower surfaces respectively. These surface may be created by hand or taken from EarthVision.
In addition to specifying files that define the top and bottom of the model region, an alternative is available to create flat surfaces at a specified elevation. The keywords TOP and BOTTOM can be replaced with TOP_FLAT and BOTTOM_FLAT respectively followed by the required elevation of the surface instead of the file name. The vertical faces of the model region are each named (name_1….., name_N). The values x_1 y_1,….., x_N y_N define the co-ordinates at which the faces intersect, such that name_n intersects the straight line (x_n, y_n) to (x_(n+1), y_(n+1)).

8.3.3 Information of Shafts and Galleries

The keyword GALLERY allows the user to define an internal model region in the form of a horizontal cylindrical gallery with centre line starting at (x_gs, y_gs, z_g) and finishing at (x_ge, y_ge, z_g). If more that one gallery is to be specified in the model region the GALLERY keyword must be entered again followed by the second gallery's geometric details. The keyword SHAFT allows the user to define an internal model region in the form of a vertical cylindrical with centre line starting at (x_s, y_s, z_top) and finishing at (x_s, y_s, z_bottom). In the usual case, when a shaft extends to the surface, two alternative formats are available in which the depth or elevation of the bottom must be specified but the top is worked out by the code to match the surface elevation. As for the galleries if more than one shaft is to be used in the model region then the keyword must be used again. Shafts and galleries can be combined, but not inclined. It should be noted that later versions of FracAffinity will enable the user to specify other, more complex internal boundaries.
Dynamic excavations can be specified using the Excavation keyword. Note that the old style input using ExcavationModel is still supported for backward compatibility but the new form is preferred.

(1) Excavation file

The keyword Excavation allows the user to specify a file in which a timetable for the construction and lining of a group of excavations with any orientations is defined. The name of this file is user defined but the suggested format is excavation_name.fai. The format of the excavation input file is detailed in Table. 8.3.5. Multiple excavations are specified within the single file; each must have a specific name that is used in boundary conditions and in reporting. Note that multiple excavations must be linked together so that each starts either at the surface or in a previously excavated section.

Table. 8.3.5 The format of the excavation input file.

Name	User-defined (Suggested Name: *excavation_name*.fai)
Format	Name *sectionname* Geometry FROM *xs ys zs* TO *xe ye ze* RADIUS *radius* Geometry FROM *xs ys zs* DEPTH *d* RADIUS *radius* Length *L_1 L_2 L_3 ..... L_N* Time *t_1 t_2 t_3 ..... t_N* Lining *L_1_start L_1_end t_1_start t_1_end* Lining *L_2_start L_2_end t_2_start t_2_end* ……. Lining *L_N_start L_N_end t_N_start t_N_end* Name *section_name … (for subsequent excavation sections)*

In the excavation file there is a block of input for each excavation section. An excavation section is straight cylinder. The name for the section is used in reporting to give a full name for a section as groupname (sectionname); this form can also be used in boundary conditions and in other input locations where an excavation name is required, but in these cases it is usually adequate to use the groupname so that the condition applies to all sections within that group.

There are two versions of the Geometry line. The first specifies the x, y and z coordinates of the two ends of the excavation and the radius. The second is for a vertical excavation and specifies the starting point (at the surface) and the final excavated depth and radius. In either case, the zs coordinate can be replaced by the keyword SURFACE to have FracAffinity calculate the relevant surface coordinate for the given (xs,ys) position.

The following two lines (Length and Time) provide a series of values for the length of the excavation at various times. The actual excavation length will be obtained at intermediate times by linear interpolation between these values. Lengths greater than the final depth value will be ignored. The number of lengths and times must be the same. Consecutive length values may not have the same time, and it is recommended that, should the user wish to input consecutive lengths at the same time, an increment of 1 second is used. The first Length must be zero and the last length can optionally be END to indicate the full excavated length (it is recommended that this is used to avoid problems with rounding when calculating the length of an inclined excavation).

The schedule for the lining of the shaft is defined on all subsequent lines. Each line corresponds to the portion of the shaft to be lined (between L_N_start and L_N_end) and the time between beginning lining (at the start) and completing the lining (at the end). These times may be identical if required. The Lining line of the file may be repeated as often as required to provide the desired lining schedule. Lining regions may overlap if desired because relining of a segment of shaft has no further effect. The L_N_end can be given as END or ENDFACE instead of a value. If END is used then the lining extends to the full excavated

length but the end wall is not lined. If ENDFACE is specified then the lining extends to the end of the excavation and includes the end wall.
The boundary conditions for the boundaries defined in the model region file are defined in the boundary condition file (Section 8.7.3). However, default boundary conditions are applied to the excavations. These are atmospheric pressure for an unlined excavation and no-flow for a lined excavation.

(2) ExcavationModel file

This approach to specifying a single excavated shaft is supported for backward compatibility. The FracAffinity Interface converts it into the Excavation file form for use in FracAffinity. The keyword ExcavationModel allows the user to specify a file in which a timetable for the construction and lining of a shaft defined. The name of this file is user defined but the suggested format is shaft_name.fai. The format of the excavation model input file is detailed in Table. 8.3.6. Multiple excavations are allowed and each must have a specific name that is used in boundary conditions and in reporting. Note that multiple excavations must not overlap or be too close together – an error message will be written if this is not respected.

Table. 8.3.6 The format of the excavation model input file.

Name	User-defined (Suggested Name: *shaft_name*.fai)
Format	Geometry *x y d radius* Depth *d_1 d_2 d_3 ..... d_N* Time *t_1 t_2 t_3 ..... t_N* Lining *d_1_top d_1_bot t_1_start t_1_end* Lining *d_2_top d_2_bot t_2_start t_2_end* ……. Lining *d_N_top d_N_bot t_N_start t_N_end*

In the excavation model file the x and y values give the location of the centre line of the shaft, while the final depth of the shaft is d. The shaft is assumed to be cylindrical with the specified radius. The following two lines (Depth and Time) provide a series of values for the depth of the shaft at various times. The actual shaft elevation will be obtained at intermediate times by linear interpolation between these values. Depths greater than the final depth value will be ignored. The number of depths and times must be the same. Consecutive times may not have the same value, and it is recommended that, should the user wish to input consecutive times, an increment of 1 second is used.
The schedule for the lining of the shaft is defined on all subsequent lines. Each line corresponds to the portion of the shaft to be lined (between d_N_top and d_N_bot) and the time between beginning lining (at the top) and completing the lining (at the bottom). These times may be identical if required. The Lining line of the file may be repeated as often as required to provide the desired lining schedule. Lining regions may overlap if desired because relining of a segment of shaft has no further effect. The bottom surface of the shaft will be

considered to be lined when a lining portion exceeds the final depth. If it is required that the bottom is lined then the final lining depth should be specified as a little more (1 m say) beyond the full depth..
The boundary conditions for the boundaries defined in the model region file are defined in the boundary condition file (Section 8.7.3). However, default boundary conditions are applied to the shaft excavation model. These are atmospheric pressure for an unlined excavation and no-flow for a lined excavation.

8.3.4 Boreholes Information

If a BOREHOLES input line has been given, the file that is named there must contain information about the geometry of the boreholes. The file also contains gridding information. The format of the file is:

```
[GRID_SIZE  size]
[CONDUCTIVITY_FACTOR  factor]
[SSC  storativity]
BH_SECTION name
        [GRID_SIZE  size]
        [CONDUCTIVITY_FACTOR  factor]
        [NON_CONDUCTING]
        [SSC  storativity]
        [FIRST_LENGTH  length]
        [GROWTH_FACTOR  factor]
        [MAX_NUMBER  num]
        RADIUS  radius
        X1  Y1  Z1
        …
        XN  YN  ZN
BH_SECTION next_name ...
```

The key information for each borehole section is its name, its radius and its location. Names must be unique. The radius is in metres, and the coordinates are used to specify a polyline. For a straight section only the end points need be specified.
Notice that the specification is for borehole sections. These are hydraulically separated, either by a packer or through being in different boreholes. It is recommended that the sections be given names that relate to the boreholes of which they are part.
The GRID_SIZE lines controls the spacing of nodes along the borehole section. If the GRID_SIZE is specified at the top of the file then it applies to all sections (unless they over-ride it). Otherwise each section must have a GRID_SIZE line. Note that the nodes along

the borehole will be spaced more closely than this if the local intact rock node spacing is smaller. Extra nodes are added for discrete feature intersections.
CONDUCTIVITY_FACTOR allows the user to change how the conductivity for the legs along the borehole is related to the radius. This would be necessary is units other than SI were used, e.g. times in years. The factor is the value by which the square of the radius is multiplied. By default it is 1.25E6, which corresponds to Poiseuille flow in a tube.
The NON_CONDUCTING keyword allows for a borehole to be specified without the nodes being connected. This is so that a pre-drilling situation can be modelled. Note that specifying CONDUCTIVITY_FACTOR 0 would have a similar effect but is not recommended as is results in the borehole nodes being connected with zero conductance.
SSC allows the storativity of the borehole to be specified. By default it is zero.
The FIRST_SIZE, GROWTH_FACTOR and MAX_NUMBER control how the borehole is linked to the rest of the system. All have defaults: the GROWTH_FACTOR defaults to 4.0; the MAX_NUMBER to 4; the FIRST_SIZE defaults to GROWTH_FACTOR times the borehole radius.
The code splits every link from a borehole section to the rest of the network according to the values given. The FIRST_LENGTH is used for the first leg, with subsequent legs growing by the GROWTH_FACTOR. If the number of legs would exceed MAX_NUMBER, then the GROWTH_FACTOR is increased. To avoid short legs, the length of a leg maybe up to 10% longer than the FIRST_SIZE and GROWTH_FACTOR imply.

8.3.5 Conductivity Scaling

Two options are provided for scaling conductivities. One provides for static rescaling in a cuboidal region and the other for dynamic rescaling near an excavation.
The option to rescale conductivities in cuboidal regions allows for modification of conductivity in any arbitrary region of the model. In this case the scaling is static, that is it is applied for all time. The option to rescale conductivities near to an excavation can be used to allow, for example, for the use of grout. In this case, the scaling is applied dynamically, as the excavation proceeds.
A scaling rule is provided in the file named on the ConductivityScale line of the model region file. There is no limit to the number of ConductivityScale lines that may be included in the model region file, and hence the number of sub-regions in which the conductivity can be modified. Note that if any node is in multiple scaling regions its conductivity will be scaled by each.
A conductivity scale file has just two lines, the first specifies the cuboid (sub-region in which the conductivity will be scaled) or the excavation and distance, and the second gives a factor by which all the nodal conductivities in that cuboid are to be multiplied. The factor is applied after all other grid generation has occurred.
The format of the file is:

```
CUBOID   xmin   xmax ymin   ymax   zmin   zmax
SCALE   factor
```

or:

NEAR_EXCAVATION name distance

SCALE factor

Here, the name can either be the name of a group of excavations, in which case the rule applies for all nodes near to any part of the excavtions in that group, or to a specific section, by using groupname (sectionname) as the name. The distance specifies the limit on distance from the node to the excavation over which the scaling applies..

Example

CUBOID 6000 7000 -66000 -65000 -500 100

SCALE 0.01

8.3.6 Flow Solver Controls

Before all the rock properties are input, the interface will request information related to the flow solver. This allows it to determine which property files will be needed.

The flow solver control file should be named in response to the following FracAffinity Interface response:

Name flow solver control file :

This file defines parameters controlling the flow solver in FracAffinity. It has the format detailed in Table. 8.3.7.

The first line will contain either the keyword NO_FLOW, STEADY, TRANSIENT, STEADY_UNSAT or TRANSIENT_UNSAT. The NO_FLOW keyword is used if the user wishes to construct the hybrid medium, but not solve for the flow. If the NO_FLOW keyword is used all other parameters in the file are ignored. The STEADY keyword instructs the Flow Solver to find the steady-state solution for the flow, the TRANSIENT keyword implies that a transient solution is to be found. The STEADY_UNSAT and TRANSIENT_UNSAT keywords cause steady-state and transient calculations to carried out using the unsaturated flow model.

The values t_start and t_end specify the start time and end time for the flow solver, while the values mt_1 to mt_M specify the output times at which the flow solver should write monitor information, using the monitor points, lines and planes specified in the flow solver output definition file (Section 8.7.5). A water balance report is also written to the log file at these times. The values ht_1 to ht_N are the times at which the entire flow field should be output to a file. Since the head output file can potentially be extremely large, it is expected that the head field will be output relatively infrequently during a solution, with the monitor points being used to give detailed information about the time-dependent behaviour of the system. The interface will warn the user if any of the output times are not in the range of times specified for the simulation. For a steady-state solution, the start and end times need not be specified, there are implicitly both zero.

Table. 8.3.7 The format of the flow solver control input file.

Name	User-defined (Suggested Name : *projectname*_control.fai)
Format	NO_FLOW *or* STEADY *or* TRANSIENT *or* STEADY_UNSAT *or* TRANSIENT_UNSAT START *t_start* END *t_end* MONITOR mt_1 ….. mt_M FLOW ht_1 …. ht_N TOLERANCE *tolerance* TIMESTEP_TOLERANCE *tolerance* [STORATIVITY_TIME_SCALE *timescale]* [RANDOM_SEED *seed]* [MAX_ITERATIONS *num]* [PT_BETA *beta]* [MODEL_UNSAT_BH] [SAVE_BEST_EVERY *num]* [MAX_PICARD_STEPS *num]* [MAX_ITERATIONS_PER_PICARD_STEP *num]* [NEWTON_ONLY] [PICARD_THEN_NEWTON *num]* [NON_LINEAR_RAMP *start step [end/AUTO]]* [STEADY_INITIAL_GUESS_FILE *filename]* [JACOBIAN_UPDATE_INTERVAL *num]* [EXCAVATION_GRID_PROMPT]

Example input file:

```
STEADY
START  1
END  1.57788e+08
MONITOR  1
FLOW  1
TOLERANCE  1e-04
```

The TOLERANCE and TIMESTEP_TOLERANCE keywords are used to specify the tolerance to be used by the Flow Solver. This gives the user control over the accuracy and speed of the flow solver. The tolerance is the value that the flow solver uses to assess when the residual errors from the calculations are small enough. The following gives some guidance on values that should be used: TIMESTEP_TOLERANCE applies in the time

stepping calculation, TOLERANCE is used in the steady-state and initial conditions calculations. If TIMESTEP_TOLERANCE is not specified, then TOLERANCE is used throughout.
For TOLERANCE it is recommended that the lowest possible value be used, for example 10^{-10} or less, and that values larger than this are only used if convergence proves difficult. For TIMESTEP_TOLERANCE, Table. 8.3.8 gives some guidance. By default, the TOLERANCE is set to 10^{-5} and the TIMESTEP_TOLERANCE is set equal to the TOLERANCE.
The TIMESTEP_TOLERANCE is used to control the size of timesteps taken in transient calculations. For each step, an estimate of the relative error is made and compared to the tolerance. If the error estimate is larger than the tolerance then the step is rejected and a smaller step is attempted; if the error is larger than the tolerance then the step is accepted and the next step is longer.

Table. 8.3.8 Guidance for setting the TIMESTEP_TOLERANCE.

10^{-3} to 10^{-4}	This range of values will give faster less accurate calculations. However, in some cases the calculations may become unstable and stop.
10^{-5} to 10^{-6}	This is the preferred range as it provides reasonable runtimes and accuracy.
10^{-8}	For more accurate runs when preliminary modelling has been carried out this value can be used, run times will be longer therefore use for final runs of the model only.
< 10^{-8}	Very small values may result in a failure of the solver to converge. This is due to the finite precision of the numerical calculations.

Finally there are some optional keywords available (shown in square brackets). The first optional keyword is STORATIVITY_TIME_SCALE. This allows the user to specify a minimum timescale, below which transient processes are ignored. However, this parameter should be set with caution, since it may result in unrealistic volumes of water being released from storage, and, therefore, it is recommended that values less than 900 seconds are used, and, where possible, the default value of 0 is applied.
The second optional keyword is RANDOM_SEED, which is used to set the seed for the Random Number Generator. This is used in the construction of the rock model for the stochastic feature generation and fractal conductivity regions. The seed value should be a positive integer.
The keyword MAX_ITERATIONS allows the number of iterations in the saturated steady-state solvers to be controlled. The default is 10000 which should be adequate for most purposes. The number specified must be less than 32000. Note that the equivalent control for the unsaturated steady-state solvers is provided by the MAX_PICARD_STEPS and MAX_ITERATIONS_PER_PICARD_STEP keywords.

The rest of the keywords are for controlling unsaturated calculations. Details are given in the Section 7.9.3. PT_BETA gives a value for the Pt transform that is used. The default is zero, and the value must be less than or equal to zero. A value of zero sometimes gives more rapid convergence but values from –0.001 down to –10 can be effective in particular cases.
MODEL_UNSAT_BH instructs the code to solve for the heads in the region above the water level in a desaturated borehole.
During the iterative solution of the steady-state unsaturated problem, the best solution so far is stored. In some cases this is achieved after a small number of iterations and no further convergence is achieved. However, if the run is terminated the solution is lost. To avoid this, the best solution is written to a standard head output file, called head_best.faout. By default this is done every 10 iterations (if there has been an improvement). The interval can be changed by including a line in the solver control file

SAVE_BEST_EVERY n

where n is the interval. A zero or negative value indicates that no file is to be written.
Within the iterative scheme for finding a unsaturated steady-state solution, the following keywords allow to control the iterations.

MAX_PICARD_STEPS nnn

controls the total number of iterations (note that this controls the total number, since the solver treats a Newton step as a "special" Picard step). The default is 100. After this, the best solution so far is treated as the final solution.

MAX_ITERATIONS_PER_PICARD_STEP nnn

Within both Picard and Newton steps, an iterative approach is used to solve the linear equations. The maximum number of iterations can be controlled with the above keyword. The default is 200.

NEWTON_ONLY

or

PICARD_THEN_NEWTON nnn

The default behaviour is for the unsaturated steady solver to use Picard iterations with a Newton step every 4^{th} iteration or after a divergent Picard step. Two alternative approaches are available, indicated by the above keywords. NEWTON_ONLY just uses Newton steps, while PICARD_THEN_NEWTON gives a number of Picard steps to be used initially. This can be effective when the Picard iterations get close to the solution but converge slowly.

NON_LINEAR_RAMP start step [end/AUTO]

In difficult unsaturated steady-state cases, the standard solver may fail to converge. In these cases, the non-linearity arising from the saturation-conductivity relationship can be introduced across a number of iterations to help convergence. The start and step relate to the fraction of the full non-linear effect that is included. Thus, NON_LINEAR_RAMP 0.4 0.2 introduces 40% of the non-linearity on the first iterations and adds a further 20% on the next three iterations. The start value can be zero. The default end value is unity, and in that case once the ramp goes above unity, the full non-linearity is used.

The non-linear fraction is used in the following way. If the full non-linear relationship gives $K(S) = K_{sat}/K_{red}$ then K_{red} (a potentially large reduction factor) is replaced by $K_{red}/(nlf+(1-nlf)*K_{red})$.

For large K_{red} this has the effect of capping the factor at $1/(1-nlf)$, so a 98% fraction is limited to a reduction factor of 50.

If an end value for NON_LINEAR_RAMP is specified, this acts as a maximum non-linear fraction that is attempted. This allows a converged solution to be obtained for a slightly less non-linear problem than the full situation. Water balance reporting uses the maximum fraction. Note that this only applies to steady-state calculations; transient calculations always use the full non-linearity.

If the keyword AUTO is used, an automated version attempts to step to full non-linearity but converges each intermediate step and saves the results. This allows a run to be undertaken with a fixed number of iterations to see how far up the ramp it is possible to get. Water balance reporting uses the final fraction.

Fig. 8.3.3 and 8.3.4 show the effect of the non-linear fraction across the full range of saturation for a Van Genuchten and a user-defined model respectively. They show the relative conductivity which is the reciprocal of the reduction factor. A logarithmic scale is used to emphasis the effect. Each figure shows the full value and the 99%, 95%, 90% and 75% values (that is f_{nl} = 1.0, 0.99, 0.95, 0.9 and 0.75).

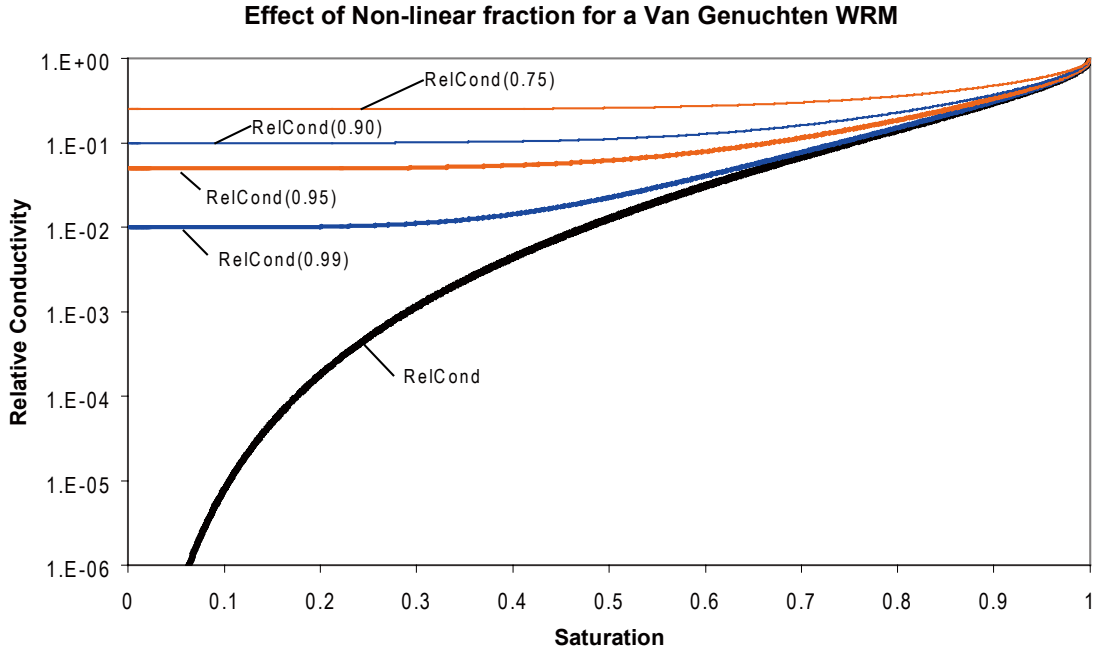


Fig. 8.3.3 The effect of the non-linear fraction on the relative conductivity for a typical Van Genuchten retention model (A=1, B=2).

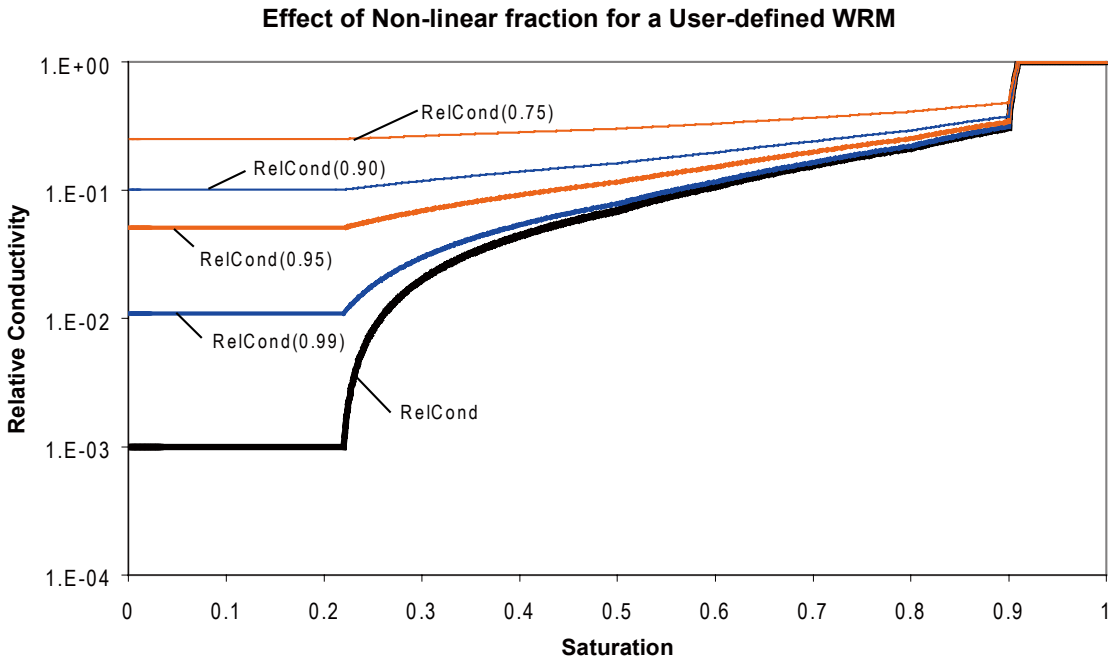


Fig. 8.3.4 The effect of the non-linear fraction on the relative conductivity for a typical user-defined retention model.

The NON_LINEAR_RAMP option requires some experimentation to find good values. It can be used in conjunction with the following option to attempt to get convergence in a series of calculations.

STEADY_INITIAL_GUESS_FILE filename

This provides an ability to load an initial guess for the solution from a file (expected to be the saved head_best file). This allows the user to carry on from where a previous attempt has reached, e.g. with a change of approach (using NEWTON_ONLY works well once a nearly correct solution is available). The filename should be a full pathname for the file to be used, in double quotes.

EXCAVATION_GRID_PROMPT

This asks for a prompt to be issued allowing cancellation of the run if there are judged to be too few nodes in the dynamic excavations. Warning messages are issued in any case if the excavated nodes are separated (in terms of distance along the excavation) by more than the diameter of the excavation. When the EXCAVATION_GRID_PROMPT keyword is present these warnings are followed by a prompt allowing the user to select whether to continue the run or terminate it. See Section 8.4.7 for a description of how to increase the refinement around an excavation.

8.4 Intact Rock Model Parameters

8.4.1 Reusing an Existing Hybrid Medium

At this stage the interface will ask:

Do you wish to use an existing hybrid medium file [y/n]?

An existing hybrid medium file contains a complete description of the hybrid medium used in a previous FracAffinity calculation. This file is named properties.faout (or may have been given a user specified name) and is written to the FracAffinity directory during a FracAffinity run. Using this file allows the user to rerun FracAffinity with the identical hybrid medium. Note that excavation information is not written to this file and so the option should not be used in that case.

If re-use of a hybrid medium is not required, the interface begins the construction of a new hybrid medium, starting with the intact rock as described here.

8.4.2 Reusing an Intact Rock Model

At this stage the interface will ask:

Do you wish to specify Intact Rock Properties?

Selecting "y" will continue to ask for information about the intact rock model. Selecting "n" will cause the interface to skip this section and ask about Deterministic Discrete Features (Section 8.5).

Information about the intact rock can be supplied either from previous FracAffinity Interface runs, or from input files. For the former case a single file is required whilst in the latter case a series of files, as described below, are needed.

At the beginning of the intact rock model specification, the interface will ask:

Do you wish to use an existing intact rock file [y/n]?

Selecting "y" for yes will cause the interface to prompt for a filename of an intact rock model saved from a previous calculation (see below) this file name will have been defined by the user in the FracAffinity run that generated the model. Using this file allows the user to re-run FracAffinity with the identical rock model. Any other reply will cause the interface to begin collecting the information needed to construct a new intact rock model.

Note that the properties are not time-dependent. The unused feature of earlier versions to specify two sets of properties has been removed.

8.4.3 Intact Rock Model: Hydraulic Conductivity

To start the process of creating an intact rock model, the interface will request the name of the rock conductivity file:

Name rock conductivity file :

This file defines the model parameters that are used in FracAffinity to generate conductivity values. It has the format shown in Table. 8.4.1. The most important feature of this input file specifies the spatial variability of the conductivity values for each stratigraphic zone (named zone_name_1,….., zone_name_N) in the model region. The keywords FRACTAL, UNIFORM and POINT_DATA refer to a scalar fractal interpolation conditioned on data (in files named below the CONDITIONING_DATA keyword), a uniform value in a zone, and a field read

from a file, respectively. Note that any conductivity rescaling (Section 8.3.5) is applied as a final step multiplying the conductivity specified here.

Table. 8.4.1 The format of the rock model conductivity input file.

Name	User-defined (Suggested Name: *projectname*_rock_cond_t*n*.fai)
Format	*zone_name_1* FRACTAL [ANISOTROPY *ax ay az*] MEAN *mean (log 10 value)* VARIANCE *variance (of the log 10 value)* DIMENSION *fractal dimension* CORRELATION *h_x h_y h_z* MESH_SIZE *delta_x delta_y delta_z* *zone_name_2* UNIFORM [ANISOTROPY *ax ay az*] VALUE *value (log 10 value)* CONDITIONING_DATA *name_1* *name_M* *zone_name_3* POINT_DATA [ANISOTROPY *ax ay az*] FILE *filename*

Example input file:

```
tsukiHWDZ
UNIFORM
VALUE -10
Felsic
FRACTAL
MEAN -10
VARIANCE 1.0
DIMENSION 3.5
CORRELATION 50 50 50
MESH_SIZE 50 50 50
```

BiotiteLWR
FRACTAL
MEAN -10
VARIANCE 0.5
DIMENSION 3.5
CORRELATION 100 100 100
MESH_SIZE 50 50 50
BioLowK
FRACTAL
MEAN -11.3
VARIANCE 2.0
DIMENSION 3.5
CORRELATION 100 100 100
MESH_SIZE 50 50 50
WeathAndCong
FRACTAL
MEAN -6
VARIANCE 2.0
DIMENSION 3.8
CORRELATION 25 25 25
MESH_SIZE 50 50 50
Sediments
FRACTAL
MEAN -7.3
VARIANCE 2.0
DIMENSION 3.7
CORRELATION 25 25 25
MESH_SIZE 50 50 50

8.4.4 Intact Rock Model: Conditioning of Hydraulic Conductivity

Conductivity conditioning data files define the data used to condition simulations of the conductivity field. They have the format specified in Table. 8.4.2.

Table. 8.4.2 The format of the rock model conductivity conditioning data input file.

Name	User-defined
Format	*x_1 y_1 z_1 LOG_10_K_1* *x_2 y_2 z_2 LOG_10_K_2* …….. *x_N y_N z_N LOG_10_K_N*

The values LOG_10_K_1,……, LOG_10_K_N specify conditioning data at = (x_1, y_1, z_1),…..,(x_N, y_N, z_N). The values specified are $\log_{10}$ of the conductivity value measured at that location. Note again that the division of conditioning data into separate files is not a requirement of the FracAffinity Interface and is independent of any stratigraphic data.
The same format is used for the POINT_DATA files, although these will generally have a three-dimensional distribution of values.

8.4.5 Intact Rock Model: Porosity and Specific Storage Coefficient

After the rock conductivity file has been successfully read, the files containing the porosity and storage coefficient parameters (if required) are read in turn.
These files define the model parameters that are used in FracAffinity to define porosity values and specific storage in different stratigraphic zones. The files have the same format for both properties. Each has the format shown in Table. 8.4.3. The storage coefficients are only needed for transient calculations.
The value of the parameter is set to value. The porosity is specified as a fraction (i.e. 10% is input as 0.1) and the specific storage coefficient is simply input in scientific notation (i.e. 1e-5). Data must be specified for each and every stratigraphic zone that is defined in the stratigraphy input file(s) defined in Section 8.3.1.
The spatial variability model for these physical properties is restricted to piecewise uniform distributions, with a single value per stratigraphic zone.

Table. 8.4.3 The format of the rock model porosity and specific storage coefficient input files.

Name	User-defined (Suggested Name: *projectname*_rock_por_t*n*.fai) (Suggested Name: *projectname*_rock_ssc_t*n*.fai)
Format	*zone_name_1 value_1* …….. *zone_name_N value_N*

Example input file:

tsukiHWDZ 0.0125

Felsic 0.0125

BiotiteLWR 0.0175
BioLowK 0.0125
BiotiteUP 0.0175
tsukiFWDZ 0.0125
WeathAndCong 0.3
Sediments
0.488

Example input file:

tsukiHWDZ 1e-09
Felsic 1e-09
BiotiteLWR 1e-10
BioLowK 1e-10
BiotiteUP 1e-10
tsukiFWDZ 1e-09
WeathAndCong 1e-08
Sediments
5e-08

8.4.6 Intact Rock model: Water Retention Models

After the porosity and storage coefficient parameters have been successfully read, the user is prompted for the name of the intact rock water retention model file (if required):

Name Intact Rock Zone - WRM file :

This file contains the information required by FracAffinity to model the water retention throughout the region. It is only required for an unsaturated calculation. It has the format described in Table. 8.4.4. For each zone a file is specified which contains the parameters for the water retention model.

Table. 8.4.4 The format of the intact rock water retention model input file.

Name	User-defined (Suggested Name: *projectname*_rock_wrm_t*n*.fai)
Format	*zone_name_1 wrmfile_1* …….. *zone_name_N wrmfile_N*

The water retention model can either be a Van Genuchten model (recommended) or a user-specified model. The formats of a water retention model parameters file for these two cases are described in Table. 8.4.5 and 8.4.6 respectively.

Table. 8.4.5 The format of the Van Genuchten water retention model file.

Name	User-defined (Suggested Name: none)
Format	*WRMVanGen* *Parameter_A [Parameter_B]* *WRMModVanGen* *Parameter_A [Parameter_B] [AEP]*

The Van Genuchten water retention model is specified as shown in Table 8.4.5 (first two lines). The model is a two parameter analytic model and is defined in Chapter 7 (Theoretical Background). The parameters for the model are given after the keyword *WRMVanGen*. If the second parameter is omitted it takes the default value of 2.0.
In addition, the capability to use a modified van Genuchten relationship was added in FracAffinity Version 3.3. The format of this file is also illustrated in Table. 8.4.5 (bottom two lines). The modified van Genuchten relationship allows for representation of a small capillary height, and less non-linearity in the hydraulic conductivity function, and is described in the Chapter 7 (Theoretical Background). The modified van Genucthen represents this capillary as an air entry pressure (AEP); if the AEP is omitted it takes the default value of 0.

Table. 8.4.6 The format of the user-defined water retention model file.

Name	User-defined (Suggested Name: none)
Format	*WRMSatCondP* N_s N_q $Cond_1$ $Cond_2$ … $Cond_{Nq}$ Sat_1 Sat_2 … Sat_{Nq} Sat_1 Sat_2 … Sat_{Ns} P_1 P_2 … P_{Ns}

For the user-defined water retention model file (Table. 8.4.6) the keyword WRMSatCondP specifies that the water retention model is a piecewise continuous function specified by pairs of data values. The first entry on line 2 of the water retention model file specifies the number of values on lines 3 and 4. The second entry gives the number of values on lines 5 and 6.

Lines 3 and 4 specify and the relationship between relative conductivity and saturation as a piece-wise linear function. Note that the values of relative conductivity must all be greater than zero or an error will be generated.
Lines 5 and 6 specify the relationship between saturation and pressure head (the negative of suction) and all the pressure values must be negative. It is assumed that pressures greater than zero correspond to the fully saturated state.

8.4.7 Intact Rock: Grid Parameters

When the conductivity, porosity, specific storage coefficient and water retention model parameter files have been read, the user is prompted for the name of the grid parameters file:

Name intact rock grid parameters file :

This file contains the parameters required in order for FracAffinity to generate the intact rock grid. It has the format defined in Table. 8.4.7, where delta_x, delta_y and delta_z are the global grid sizes in the x-direction, y-direction and z-direction, respectively. The parameter Min_N controls the minimum number of cells which should be constructed between control points. The resolution is used to limit the global resolution of the grid, preventing large numbers of very small cells from being constructed.

Table. 8.4.7 The format of the grid parameters input file.

Name	User-defined (Suggested Name : *projectname*_rock_grid.fai)
Format	X_SIZE *delta_x* Y_SIZE *delta_y* Z_SIZE *delta_z* MIN_NUM *Min_N* RESOLUTION *resolution* THIN_FORMATION *ALL or FormationName* SUB_REGION *x_min x_max y_min y_max z_min z_max* X_SIZE *delta_x* Y_SIZE *delta_y* Z_SIZE *delta_z* RESOLUTION *resolution* GRID_REFINEMENT GRID_SURFACE *grid file (usually water table grid)* ABOVE_NUM *number of nodes* ABOVE_RESPECT *minimum distance* ABOVE_RATIO *ratio of distance between nodes* BELOW_NUM *number of nodes added below grid*

	BELOW_THICK *distance below grid in which nodes are added* LOCAL_REFINEMENT CUBOID *x_min x_max y_min y_max z_min z_max* LOCAL_X *dx* LOCAL_Y *dy* LOCAL_Z *dz* LOCAL_REFINEMENT NEAR_BHS *bhs_nam distance* LOCAL_X *dx* LOCAL_Y *dy* LOCAL_Z *dz* LOCAL_REFINEMENT NEAR_EXCAVTION *name distance* LOCAL_X *dx* LOCAL_Y *dy* LOCAL_Z *dz* FORCE_X *z* FORCE_Y *z*

Example input file:

```
X_SIZE  50
Y_SIZE  50
Z_SIZE  50
MIN_NUM  4
RESOLUTION  0.01

SUB_REGION  5378 5578 -68628 -68428 -1000 400
X_SIZE  10
Y_SIZE  10
Z_SIZE 10
RESOLUTION  0.01

THIN_FORMATION WeatheredTokiGranite
```

```
GRID_REFINEMENT
    GRID_SURFACE watertable.fai
    ABOVE_NUM 5
    ABOVE_RESPECT 10
    ABOVE_RATIO 2
    BELOW_NUM 5
    BELOW_THICK 10

SUB_REGION   5100 5900 -69200 -67950 100 400
X_SIZE   10
Y_SIZE   10
Z_SIZE   10
RESOLUTION   0.01
```

The resolution works by limiting the size of cells in any direction to the value obtained when the resolution is multiplied by the total model size in that direction. Therefore, for a resolution of 0.001 and a model size in the X direction of 1,000 m, the minimum cell size in the x direction will be 1 m.

In order to ensure that the grid respects all of the strata within the geological model, the THIN_FORMATION keyword maybe used. Two options are available. The user may either specify each formation that FracAffinity should treat as a Thin Formation, or specify the ALL keyword, in which case every formation will be treated as a Thin Formation. The result of specifying Thin Formations is that FracAffinity will introduce intermediate nodes in the middle of formations that are completely crossed by network legs.

Sub-regions with a smaller grid spacing may be specified using the keyword SUB_REGION. The minimum and maximum x, y and z co-ordinates of sub-regions are defined using the x_min x_max y_min y_max z_min z_max parameters. Sub-regions may be repeated as often as required. If sub-regions overlap, then the values specified in later sub-region definitions will take precedence. Comments may be inserted in the file as desired. The first region covers the entire model region by default and does not require a SUB_REGION keyword. Sub-regions need not be wholly contained within the model region. Note that for a SUB_REGION, the cell size in each direction is affected across the model, not just in the region specified (this is where the three directions combine).

Refinement of the flow network around a Two-D grid may be specified using the GRID_REFINEMENT keyword. This was introduced into FracAffinity to allow for better resolution of region above and below the water table surface when running unsaturated flow models, especially in transient mode. However, the approach is generic and may be used to refine the grid for other reasons. The keyword requires the user to define a grid surface, normally the water table, either from site-specific data, or by running a steady-state

unsaturated flow calculation. The format of the grid is provided in Table. 8.3.2. The user then specifies the number of nodes to be added above and below the specified grid, and, above the grid the ratio of the distance between the grid nodes, and below the grid the regular spacing of the grid nodes. In addition, the user specifies a minimum distance over which the nodes are added above the specified grid. This is required to cope for cases where the water table intersects the ground surface (i.e., models with fully saturated regions).

Where no GRID_REFINEMENT block is present, vertically equal size cells are used throughout. The cell size is calculated for each x-y position, resulting in cells with tops and bottoms that are not horizontal.

The LOCAL_REFINEMENT keyword specifies a truly local effect. Rather than controlling the cell size, this controls the network spacing within the cells. Three types are available: a CUBOID region, which will affect all cells intersecting the cuboid; a NEAR_BHS, which affects all cells nearer to a borehole section than the specified distance; and a NEAR_EXCAVATION, which affects all cells nearer to the excavation than the given distance. A distance of zero can be used to affect those cells through which the borehole or excavation passes. When the CUBOID option is used, new cell boundaries are created in the intact rock grid so that the cells that are refined cover precisely the region specified. The name for NEAR_EXCAVATION can be the group name or a specific name in the form groupname (sectionname). Note that if multiple local refinements affect the same cell then the one requiring the most refinement will be used.

If a line of nodes is required at some specific x-, or y-coordinate, then the FORCE_X, and FORCE_Y keywords can be used. Note that the gridding algorithm cannot always honour these requests, particularly if there are several close together. The gridding algorithm is described in some detail in Chapter 7 (Theoretical Background). An additional keyword - FORCE_Z - was available in early versions of FracAffinity, but was not included from Version 3.0 onwards, since it is not compatible with the gridding approach introduced in Version 3.0.

8.4.8 Saving Intact Rock Models for Reuse

Once all the parameters for the intact rock model have been specified, the interface will give the user the opportunity to specify a file to which the intact rock information should be written:

Do you wish to save the intact rock model to a file [y/n]?

This file will be written by FracAffinity and stored in the FracAffinity directory. The file can then be re-used in later calculations (see Section 8.4.1).

8.5 Deterministic Discrete Feature Model

8.5.1 Adding Deterministic Discrete Features

When the intact rock parameters have been specified, the DDFs may be specified. The interface will ask

Do you wish to add Deterministic Discrete Features [y/n]?

Selecting "y" will continue to ask for information about the deterministic discrete features. Selecting "n" will cause the interface to skip this section and ask about Stochastic Discrete Features (Section 8.6).

8.5.2 Deterministic Discrete Feature Geometry

Input information on deterministic discrete features (DDFs) is split into six files: geometry; conductivity; porosity; specific storage coefficient, water retention model and grid. Additional files may also be used to condition conductivity simulations or specify point data.

The format of the DDF geometry file is provided in Table. 8.5.1. The geometric information is specified by reference to surface and polygon files in the same format as in Table. 8.3.2 and 8.3.3. Typically, EarthVision would generate these files, but the user may generate files of the same format manually.

The feature names will be used for reference in the property files described in Sections 8.5.4-8.5.6. Note that the use of the polygon file to limit the bounds of the fault surface is optional, and is not necessary for fault surfaces when the fault has lateral or vertical terminations. Note that the polygon is used to discard points in the surface file rather than the FracAffinity grid, and so cannot be used successfully if the surface is coarsely specified, e.g. by the corners only.

There is some flexibility in the way in which DDFs may be specified and modelled in FracAffinity:

- If the user wishes to grid the DDF in the same manner that it is treated in the sequence file, it should be given exactly the same name as in the sequence file. This will result in the truncations defined by the sequence file being honoured.

If the user wishes to grid the DDF independently of the sequence file, it should be given a different name to that defined in the sequence file. This will result in the truncations defined by the sequence file being ignored. This functionality also allows the user to introduce an additional DDF without the requirement for construction of a new model in EarthVision.

Table. 8.5.1 The format of the DDF geometry input file.

Name	User-defined (Suggested Name : *projectname*_ddf_geom.fai)
Format	*feature_name_1 surfacefile_1 [polygonfile_1]* ……. *feature_name_N surfacefile_N polygonfile_N*

Example input file:

```
Tsukiyoshi tsuki80_v1.fa          tsu.ply
```

Once the DDF geometry file has been read the Interface will echo to the user those features which it intends to model, e.g.:

DDF feature names

0: Tsukiyoshi

Next the Interface will begin to collect information about the properties of the features.

8.5.3 Deterministic Discrete Features: Hydraulic Conductivity

The interface requests DDF conductivity information with the prompt:

Name rock DDF conductivity file:

This file defines the model parameters that will be used to generate conductivity values in the DDFs. It also defines whether DDFs will be included in the generation of enhanced conductivity intersections, and what enhancement factor will be used. It has the format shown in Table. 8.5.2.

Table. 8.5.2 The format of the DDF conductivity input file.

Name	User-defined (Suggested Name: *projectname*_ddf_cond_t*n*.fai)
Format	[ENHANCED_CONDUCTIVITY] *feature_name_1* [ENHANCEMENT_FACTOR *f*] FRACTAL [ANISOTROPY *ai an*] MEAN *mean* VARIANCE *variance* DIMENSION *dimension* CORRELATION *h_x h_y h_z* MESH_SIZE *delta_x delta_y delta_z* APERTURE *aperture* *feature name2* [ENHANCEMENT_FACTOR *f*] POINT_DATA [ANISOTROPY *ai an*] FILE *filename* APERTURE *aperture* *feature_name_N* [ENHANCEMENT_FACTOR *f*] UNIFORM [ANISOTROPY *ai an*] VALUE *value* APERTURE *aperture*

	CONDITIONING_DATA *name_1* ……. *name_M*

Example input file:

Tsukiyoshi

FRACTAL

MEAN -12

VARIANCE 2

DIMENSION 3.8

CORRELATION 50 50 50

MESH_SIZE 50 50 50

APERTURE 1.61e-04

The most important feature of this input file specifies the spatial variability of the hydraulic conductivity for each deterministic discrete feature (named feature_name_1,….., feature_name_N) in the model region. The keywords FRACTAL, UNIFORM and POINT_DATA refer to a fractal interpolation conditioned on data (in files named below the CONDITIONING_DATA keyword), a uniform value in a zone, and data read from a file, respectively. It should be noted that the conditioning data files should be independent of the DDFs: it is simply (x, y, z, $\log_{10}$[k]) data, that would be available from a borehole, for example. The use of multiple conditioning data files allows for the use of multiple borehole data files.

These data give a scalar (single value) for the conductivity at each node. By default, the conductivity is isotropic, but the ANISOTROPY keyword can be used to specify different values for the in-plane and normal directions. The keyword is followed by the scaling factors for these two directions. These are used to multiply the base conductivity value when defining the conductance of legs in the appropriate direction. The normal-direction scaling factor is only used when there is a leg that intersects the feature; all the connections within the DDF use the in-plane value.

If the keyword ENHANCED_CONDUCTIVITY is specified, then DDF intersections will be explicitly modelled as enhanced conductivity features. The ENHANCEMENT_FACTOR for each DDF specifies the factor to be used to derive the intersection conductivity from the local DDF conductivity. A value of zero excludes a particular DDF from having enhanced conductivity intersections. Note that for DDF-SDF intersections to have enhanced conductivity, the ENHANCED_CONDUCTIVITY keyword must be specified here and in the SDF geometry file.

For the FRACTAL interpolation mean is the mean value, variance is its variance, dimension is the fractal dimension and h_x h_y h_z are the correlation lengths in each direction. The calculational mesh size for the generation of the fractal interpolation of conductivity is given

by the resolution delta_x delta_y delta_z. In the UNIFORM case value is the single value used to represent the conductivity of the specified DDF. The POINT_DATA option behaves exactly as in the intact rock case, with the file containing (x, y, z, $\log_{10}$[k]) data. Data must be specified for each and every feature that is defined in the DDF geometry input file(s) defined in Table. 8.5.1.
Note that the aperture of a DDF is fixed with a single value for the entire feature (using the APERTURE keyword). This applies even when the conductivity is spatially variable.

8.5.4 Deterministic Discrete Features: Conditioning of Hydraulic Conductivity

The conditioning data file(s) named in the DDF conductivity file define the data that will be used in FracAffinity to condition deterministic discrete features. The format of the DDF conditioning data file is the same as that used for intact rock conditioning data and is detailed in Table. 8.4.2.

8.5.5 Deterministic Discrete Features: Porosity and Specific Storage

Once the conductivity information has been specified for the deterministic features, the interface will prompt for the porosity and specific storage coefficient files (if required) in turn.

Name rock ddf porosity file :

Name rock ddf specific storage file :

These files define the model parameters that will be used in FracAffinity to define porosity values and specific storage in the DDFs. The files have the same format for both properties; each has the format shown in Table. 8.5.3. The specific storage file is only required for transient calculations.
The value of the parameter is set to value. The porosity is specified as a fraction (i.e. 10% is input as 0.1) and the specific storage coefficient is input in scientific notation (i.e. 1e-5).
Data must be specified for each and every feature that is defined in the DDF geometry input file defined in Table. 8.5.1. The format of the input files are as shown in Table. 8.5.3.

Table. 8.5.3 The format of the DDF porosity and specific storage coefficient input file.

Name	User-defined (Suggested Name : *projectname*_ddf_por_t*n*.fai) (Suggested Name : *projectname*_ddf_ssc_t*n*.fai)
Format	*feature_name_1 value_1* …….. *feature_name_N value_N*

Example porosity input file:

Tsukiyoshi 0.1

Example specific storage coefficient input file:

Tsukiyoshi 1e-12

8.5.6 Deterministic Discrete Features: Water Retention Models

Next, the interface will prompt the user to enter the name of the DDF water retention model input file (if required):

Name DDF Feature - WRM file :

The format of this file is described in Table. 8.5.4. For each DDF feature, the name of a file is specified which contains the water retention model parameters. This file is only required for unsaturated calculations.

Table. 8.5.4 The format of the DDF water retention model input file.

Name	User-defined (Suggested Name: *projectname*_ddf_wrm_t*n*.fai)
Format	*feature_name_1 wrmfile_1* …….. *feature_name_N wrmfile_N*

The format of the parameter files is the same as that described in Section 8.4.6 and Table. 8.4.5.

8.5.7 Deterministic Discrete Features: Grid Parameters

The final piece of information required are the grid parameters for the DDF gridder. The Interface will prompt:

Name ddf grid parameters file :

This file contains the parameters required in order for FracAffinity to generate a grid for each DDF. It has the format given in Table. 8.5.5, where delta_x and delta_y are the grid sizes in the x-direction and y-direction, respectively. The parameter MIN_NUM controls the minimum number of cells which should be constructed between corner points of the DDF grids. The resolution is used to limit the number of nodes in each DDF grid, preventing large numbers of very small cells from being constructed.

The resolution works by limiting the size of cells in any direction to the value obtained when the resolution is multiplied by the length of the DDF in that direction. Therefore, for a resolution of 0.01 and a DDF in the X direction of 100 m, the minimum cell size in the x direction will be 1m.

Table. 8.5.5 The format of the DDF grid parameters input file.

Name	User-defined (Suggested Name : project*name*_ddf_grid.fai)
Format	X_SIZE *delta_x* Y_SIZE *delta_y* MIN_NUM *min_num* RESOLUTION *resolution* LOCAL_REFINEMENT

	CUBOID *x_min x_max y_min y_max z_min z_max* LOCAL_X *dx* LOCAL_Y *dy* LOCAL_Z *dz* LOCAL_REFINEMENT NEAR_BHS *bhs_name distance* LOCAL_X *dx* LOCAL_Y *dy* LOCAL_Z *dz*

Example input file:

X_SIZE 50

Y_SIZE 50

MIN_NUM 1

RESOLUTION 0.01

The LOCAL_REFINEMENT keyword specifies a truly local effect. Rather than controlling the cell size, this controls the network spacing within the cells. Two types are available, a CUBOID region, which will affect all cells intersecting the cuboid, and a NEAR_BHS, which affects all cells nearer to a borehole section than the specified distance. A distance of zero can be used to affect those cells through which the borehole passes.

Where a DDF crosses a gallery, shaft or excavation, FracAffinity will ensure that it links to the feature by moving a DDF node into the feature if necessary. This will only be required when the grid spacing on the DDF is larger than the size of the feature.

8.5.8 Saving DDF Rock Models for Visualisation

Once all the parameters for the DDF rock model have been specified, the interface will give the user the opportunity to specify a file to which the intact rock information should be written:

Do you wish to save the ddf rock model to a file [y/n]?

Responding with the "y" will give the user the opportunity to specify a filename to store the DDF rock model.

Name ddf save file :

This file will be written by FracAffinity and stored in the FracAffinity directory. The name of the file is user specified. An EarthVision file may then be created using FracAffinity in output mode as described in Section 8.9.

8.6 Stochastic Discrete Features Model

8.6.1 Adding Stochastic Discrete Features

After any DDFs have been specified, the interface will ask:

Do you wish to add Stochastic Discrete Features [y/n]?

Selecting "y" will cause the interface to ask for information about the SDFs. Selecting "n" will cause the interface to skip this section and ask about the parameters for the flow solver.

8.6.2 Stochastic Discrete Features: Geometrical and Hydraulic Properties

If the user chooses to generate an SDF rock model, the interface will prompt for further files. The information required to generate SDFs is the geometry and physical properties (conductivity, porosity and specific storage coefficient).

The interface will first prompt:

Name SDF properties file name :

The geometric and physical property information is specified in a file with the format shown in Table. 8.6.1.

Table. 8.6.1 The format of the stochastic discrete feature input file.

Name	User defined (Suggested Name : *projectname*_sdf_*tn*.fai)
Format	[ENHANCED_CONDUCTIVITY] ZoneName1 [ENHANCEMENT_FACTOR f] *FeatureDist Parms* *OrientationDist Parms* *ShapeDist Parms* tortuosity aperture *ConductivityDist Parms* [ANISOTROPY *ai an*] porosity storage ZoneName2 … … …

Example input file:

TsukiHWDZ

RANDOM 1.37e-06

UNIFORM 76 86 -10 10

UNIFORM-RECT 50 100 50 100

1 3.16e-04

UNIFORM -9 -8
0.1 1e-07

Felsic
RANDOM 1.96e-07
UNIFORM 16 32 -116 -84
UNIFORM-RECT 50 100 50 100
1 5.1e-03
UNIFORM -6 -6
0.1 1e-07

Felsic
RANDOM 1.96e-07
UNIFORM 77 81 -2 4
UNIFORM-RECT 50 100 50 100
1 5.1e-03
UNIFORM -6 -6
0.1 1e-07

tsukiFWDZ
RANDOM 5.33e-07
UNIFORM 77 81 -2 4
UNIFORM-RECT 50 100 50 100
1 1.61e-03
UNIFORM -8 -6
0.1 1e-07

tsukiFWDZ
RANDOM 5.33e-07
UNIFORM 43 56 152 162
UNIFORM-RECT 50 100 50 100
1 1.61e-03
UNIFORM -8 -6
0.1 1e-07

```
tsukiFWDZ
FIXED 5433.29 -68552.4 -754.805
UNIFORM 43 56 152 162
UNIFORM-RECT 50 100 50 100
1 1.61e-03
UNIFORM -8 -6
0.1 1e-07

tsukiHWDZ
FIXED 5433.29 -68552.4 -675.605
UNIFORM 76 86 -10 10
UNIFORM-RECT 50 100 50 100
1 3.16e-04
UNIFORM -9 -8
0.1 1e-07
```

The geometric information is specified by formation name, the names of which must be consistent with the formations in the stratigraphy as defined in the sequence file (Section 8.3.1). The file also defines whether SDFs will be included in the generation of enhanced conductivity intersections, and what enhancement factor will be used.
After the name of the formation, the models or distributions to be used for the SDFs are defined. If the SDF density in a given zone is set to zero, or a zone name is omitted, no SDFs will be generated in that zone. Unlimited sets of SDFs may be defined for each zone. This is done by entering the name of the zone and the associated parameters, in the same way as for the first set of SDFs, each time a further set is to be generated.
The scalar (single value) conductivity gives a conductivity at each node. By default, the conductivity is isotropic, but the ANISOTROPY keyword can be used to specify different values for the in-plane and normal directions. The keyword is followed by the scaling factors for these two directions. These are used to multiply the base conductivity value when defining the conductance of legs in the appropriate direction. The normal-direction scaling factor is only used when there is a leg that intersects the feature; all the connections within the SDF use the in-plane value.
If the keyword ENHANCED_CONDUCTIVITY is specified, then SDF intersections will be explicitly modelled as enhanced conductivity features. The ENHANCEMENT_FACTOR for each SDF set specifies the factor to be used to derive the intersection conductivity from the local SDF conductivity. A value of zero excludes a particular SDF set from having enhanced conductivity intersections. Note that for DDF-SDF intersections to have enhanced

conductivity, the ENHANCED_CONDUCTIVITY keyword must be specified here and in the DDF geometry file.
The different types of distributions, the relevant keywords and the required parameters used in the input data file are defined as follows:

(1) Feature Distributions

The format used to define feature distributions is detailed in Table. 8.6.2. The following distributions are available for defining the location of the SDF centroids:

- RANDOM: The SDF centroids are located randomly according to the chosen distributions, with a specified number density (see Chapter 7 Theoretical Background for a full discussion);
- FIXED: A single SDF is generated centred on a specified location. Although the centroid of the SDF is defined deterministically the other parameters are still defined stochastically.

Table. 8.6.2 Parameters for feature position keywords.

Keyword	Parameters
RANDOM	*density*
FIXED	*x, y, z*

(2) Orientation Distributions

The format used to define orientation distributions is detailed in Table. 8.6.3. The following distributions are available for defining the orientation distribution of SDFs:

- UNIFORM: SDFs are oriented with angles sampled from a uniform distribution.
- FISHER: SDFs are oriented with angles sampled from a Fisher distribution.
- DATASET: SDFs are oriented according to values sampled from a specified dataset.

Table. 8.6.3 Parameters for feature orientation keywords.

Keyword	Parameters
UNIFORM	*min_da, max_da, min_dd, max_dd*
FISHER	*dip_angle, dip_dirn, kappa*
DATASET	*filename*

The parameters for the uniform distribution, min_da and max_da, are the minimum dip angle and the maximum dip angle respectively, these values range between -180˚ and 180˚ measured from the horizontal (the adequate and recommended range is 0˚ and 90˚). The parameters, min_dd and max_dd, are the minimum dip direction and maximum dip direction values respectively and are specified by angles ranging from -360˚ to 360˚ clockwise to north (the adequate and recommended range is 0˚ and 360˚). If the maximum is less than the minimum then the ranged is assumed to be "wrapped round", so 340˚ to 20˚ is interpreted as -20˚ to 20˚.
The Fisher distribution parameters are the central dip angle, the central dip direction and kappa.

For the DATASET option, the specified file has lines giving the dip-angle and dip-direction. For each SDF, a randomly chosen line from the file is used to give the orientation.

(3) Shape Distributions

The format used to define shape distributions is detailed in Table. 8.6.4. The following distributions are available for defining the lengths and widths of SDFs:

- UNIFORM RECTANGLES: SDFs are rectangular with side lengths and widths sampled from a uniform distribution,
- POWER-LAW RECTANGLES: SDFs are rectangular with side lengths and widths sampled from a power-law distribution, and
- POWER-LAW SQUARES: SDFs are square with side lengths sampled from a power-law distribution.

Table. 8.6.4 Parameters for feature shape keywords.

Keyword	Parameters
UNIFORM-RECT	*W_min, W_max, L_min, L_max*
POWER-RECT	*L_min, L_max, power, a_min, a_max*
POWER-SQUARE	*L_min, L_max, power*

In the shape distribution input file (Table. 8.6.4) *W_min*, *W_max*, *L_min* and *L_max* provide the minimum width, maximum width, minimum length and maximum length respectively. The quantity *power* is the exponent of the power law. The parameters *a_min*, and *a_max* give the minimum and maximum aspect ratio for the rectangles. This allows the user to prevent very long narrow rectangles from being generated.

(4) Tortuosity and Aperture

Tortuosity and aperture are specified as single values with no accompanying keyword. The value of tortuosity is a measure of how far a particular particle will have to travel between two points. For SDFs where the particle can travel in a straight line the value of tortuosity is equal to one. Where SDFs are partly filled and particles cannot flow in straight lines the value of tortuosity is increased. For example, a value of two would indicate that the path length is double that of the straight line distance between the points of interest. FracAffinity represents the effect of tortuosity by decreasing the effective conductivity by the tortuosity factor that is given.

The aperture is used when determining the velocity of particles moving through the SDFs (see Section 8.7).

(5) Conductivity Distributions

The format used to define conductivity distributions is detailed in Table. 8.6.5. The following distributions are available for defining the conductivity of SDFs:

- UNIFORM: conductivity values are sampled from a uniform distribution, and
- TRUNCATED NORMAL: conductivity values are sampled from a truncated normal distribution.

Table. 8.6.5 Parameters for feature conductivity distribution keywords.

Keyword	Parameters
UNIFORM	*min, max*
TRUNCATED-NORMAL	*mean, stdev, min, max*

The values used in these distributions should be entered as Log 10 values (e.g. 1e-5 m/s should be entered as -5). The min and max parameters define the minimum and maximum values for log10 conductivity for each distribution. The truncated-normal distribution has the extra parameters of mean and standard deviation for the normal distribution. A single value for the conductivity can be defined by using the same number for the min and max conductivity (in the UNIFORM case).

(6) Porosity and Storage Coefficient

After the conductivity has been specified, Porosity and Storage Coefficient are specified as single values with no accompanying keyword.

8.6.3 Stochastic Discrete Features: Water Retention Models

Next, the interface will prompt the user to enter the name of the SDF water retention model input file (if required):

Name SDF Feature - WRM file :

The format of this file is described in Table. 8.6.6. All SDF features use the same water retention model, so the file simply contains a keyword, ALLSDFs, followed by the name of the file containing the parameters for the model. This file is only required for unsaturated calculations. The format of the parameter file is the same as that described in Section 8.4.6 and Table. 8.4.5 and 8.4.6.

Table. 8.6.6 The format of the SDF water retention model input file.

Name	User-defined (Suggested Name: *projectname*_sdf_wrm_t*n*.fai)
Format	ALLSDFs *wrmfile*

8.6.4 Stochastic Discrete Features: Grid Parameters

The gridding of SDFs is controlled by parameters specified in the SDF grid file.
The interface will prompt:

Name SDF grid file name :

The grid file consists of information about the grid spacing (which is the same in both directions) of each SDF and allows local refinement, as shown in Table. 8.6.7.

Table. 8.6.7 File format for the SDF grid file.

Name	User defined (Suggested Name: *projectname*_sdf_grid.fai)
Format	GRID_SIZE *delta_x delta_y* MIN_NUM *min_num* RESOLUTION *resolution* LOCAL_REFINEMENT CUBOID *x_min x_max y_min y_max z_min z_max* LOCAL_SIZE *dist* LOCAL_REFINEMENT NEAR_BHS *bhs_name distance* LOCAL_SIZE *dist*

Example input file:

GRID_SIZE 40 40

MIN_NUM 1

RESOLUTION 0.01

The gridder always places a node at each corner of the SDF, before adding additional nodes using the parameters specified by the user. The internal nodes are placed so that they are separated by less than the given distance.

For SDFs, local refinement applies to the whole of an SDF. So, if any part of an SDF is in the specified cuboid or near to the named borehole section then the whole SDF uses the new grid size.

Where an SDF crosses a gallery, shaft or excavation, FracAffinity will ensure that it links to the feature by moving an SDF node into the feature if necessary. This will only be required when the grid spacing on the SDF is larger than the size of the feature.

8.6.5 Saving SDF Rock Models for Visualisation

Once all the parameters for the SDF model have been specified, the interface will give the user the opportunity to specify a file to which the intact rock information should be written:

Do you wish to save the sdf rock model to a file [y/n]?

Responding with the "y" will give the user the opportunity to specify a filename to store the SDF rock model.

Name sdf save file :

This file will be written by FracAffinity and stored in the FracAffinity directory. The file can then be used in the output processing step to generate an EarthVision file for viewing.

8.7 Other Solution Information

Once all the rock properties have been set up, the interface will request the remaining information related to the flow solution. The information required depend on whether a transient calculation is being perform. If there is no flow calculation being undertaken then no further input is required. Before asking for this information, the interface will ask whether the hybrid rock model is to be saved.

8.7.1 Save Hybrid Rock Model

Once all the parameters for the hybrid model have been specified, the interface will give the user the opportunity to specify a file to which the intact rock information should be written:

Do you wish to save the hybrid rock model to a file [y/n]?

Responding with the "y" will give the user the opportunity to specify a filename to store the hybrid rock model.

Name hybrid save file :

This file will be written by FracAffinity and stored in the FracAffinity directory. The file can then be used in the output processing step to generate an EarthVision file for viewing or can be reused in subsequent calculations (see Section 8.4.1).

Note that FracAffinity will automatically create a hybrid rock model file, called properties.faout, if the save option is not specified here. This automatically created file will be binary, whereas a file named here will be ASCII. The binary file will be written much faster than the ASCII file and so the ASCII file should only be created if its contents need to be accessed by the user.

8.7.2 Initial Conditions

If the flow solver control file specifies that a transient flow problem is to be solved, the interface will ask about the initial conditions for the problem:

Name Initial Conditions file :

This file defines initial conditions. It has two possible formats. The first corresponds to the choice of a steady-state initial condition and is shown in Table. 8.7.2.

Table. 8.7.2 The steady-state initial conditions input file.

Name	User-defined (Suggested Name : projectname_init_conditions.fai)
Format	STEADY_STATE

Example input file:

STEADY_STATE

The second possible format corresponds to initial head values specified at points in the model region. These values are interpolated to generate an initial value for every point in the model region. The format is detailed in Table. 8.7.3.

Table. 8.7.3 The format of the initial conditions specified at points in the model region.

Name	User-defined
Format	*time* *x_1 y_1 z_1 h_1 sat_1 vx vy vz vel_1* *x_2 y_2 z_2 h_2 sat_2 vx vy vz vel_2* …… *x_N y_N z_N h_N sat_N vx vy vz vel_N*

This file, referred to as an initial_conditions_file, may be an output of a previous FracAffinity calculation. The file may also be constructed by the user. The format of the file is the same as that of the head output file (head_n.faout) and this file is the one that should be specified when using the results of a previous calculation. The time at the top of this file is ignored for the purpose of the transient calculation and simply refers to the output time from a previous calculation. If a file is being constructed by hand simply enter the number one.
If the locations of the points in the initial conditions file are not the same as the FA rock model's nodes then they will be interpolated onto the model, if possible. If the initial conditions data is too sparse, or fails to overlap with the rock model, then the interpolation will fail and the flow calculation will be aborted.

8.7.3 Boundary Conditions

Various types of boundary conditions can be used: Head; Flux; Suction; Recharge; and Seepage. All are specified in the boundary conditions file. The boundary conditions file should be provided in response to the following FracAffinity Interface prompt:

Name Boundary Conditions file :

This file defines boundary conditions at specified faces and on engineered features. It has the format shown in Table. 8.7.4. If a face which exists in the model region file (Section 8.3.2) is not listed in the input file the default boundary condition of no-flow is applied.

Table. 8.7.4 The format of the boundary conditions input file.

Name	User-defined (Suggested Name : *projectname*_flow_bounds.fai)
Format	HEAD_FILE *file_location* *face_name_1* ALL *(not for borehole section)* FLUX/HEAD/RECHARGE/SUCTION/SEEPAGE *… values …* *face_name_2* *n*

	x_1 y_1 *.....* *x_n y_n* FLUX/HEAD/RECHARGE/SUCTION/SEEPAGE *… values …* *face_name_3* ALL HEAD TOPOGRAPHIC *face_name_4* ALL HEAD USE_FILE

The face names must correspond to one of the following:

- TOP or BOTTOM.
- The names following FACES in the model region file.
- The name of a borehole section.
- The names following GALLERY or SHAFT in model region file.
- The names following ExcavationModel in the model region file.

(Section 8.3.2 gives more information about the model region input file).

After specifying the face across which the boundary conditions are to be defined the user must first specify the boundary conditions across the whole of the named face by using the keyword ALL. Polygonal areas can also be defined within the top and bottom surfaces by the set of points (x_1, y_1),……,(x_n1, y_n1). The area within this polygon can then be assigned a different boundary condition to the rest of the face. Polygonal areas cannot be set on the vertical side boundaries or for borehole sections. Note that the polygon is automatically closed, so the first point should not be repeated at the end (unlike EarthVision polygon files). There are two principle options for the type of boundary condition on external boundaries, namely flux or head conditions. For unsaturated cases, the recharge, suction and seepage conditions can also be used.

Flux Condition

Fixed flux conditions are set using the keyword FLUX. A positive flux is one out of the model. Except for the borehole section, the specified flux is a volume per unit area per unit time, $m^3/m^2/s$ in SI units. The flux for a borehole in the total flow rate, m^3/s in SI units.

Following this keyword the time-varying flux can be given. The specification of time-dependent values for boundary conditions has a number of options. These apply to all boundary condition types. Details are given after the boundary condition types have all been discussed.

Head Condition

Head conditions are set using the HEAD keyword. Head conditions may be uniform, topographic or derived from a previous calculation. Uniform head conditions are the default and not associated with a keyword. Uniform heads may be time dependent and are specified as discussed later. A TOPOGRAPHIC condition sets the head value to the z-coordinate at the boundary location. Using a file requires that the filename be specified at the top of the boundary conditions file and the USE_FILE keyword specified for the particular face.

Suction Condition

The use of SUCTION boundary conditions (a negative pressure) is generally intended for internal regions such as shafts or galleries (including the excavation model), but may be applied to any boundary. The format is exactly the same as that used for HEAD or FLUX boundary conditions as described above, with the SUCTION keyword replacing HEAD or FLUX. The value specified for SUCTION should be a positive quantity and suction equal to this amount will be applied.

Recharge Condition

The use of RECHARGE boundary conditions is expected to be useful on the top surface of the model but may be applied to any boundary. The format is exactly the same as that used for HEAD or FLUX boundary conditions as described above, with the RECHARGE keyword replacing HEAD or FLUX. Unlike the flux condition a positive value for RECHARGE corresponds to a recharge into the Model Region.

The RECHARGE boundary condition has different behaviour in the saturated and unsaturated cases.

For the saturated case, it assumes that the pressure at the top surface of the Model Region is atmospheric, but adds the additional amount of recharge specified just below the surface. This may immediately discharge again if the model conductivity is too low to support the flow given.

For the unsaturated case, it restricts the amount of recharge to the value specified. This allows the region below the surface to become desaturated in the unsaturated case if the natural inflow is too low. The actual inflow may be less than the specified value in order to retain the condition that the boundary is less saturated than the interior (equivalently, the actual recharge is capped by the gravity flow). When the surface is saturated, the flow is determined by an atmospheric surface condition, and so may be outwards.

Seepage condition

The use of SEEPAGE boundary conditions is generally intended for internal regions such as shafts or galleries (including the excavation model), but may be applied to any boundary. The format is exactly the same as that used for HEAD or FLUX boundary conditions as described above, with the SEEPAGE keyword replacing HEAD or FLUX. The value specified for SEEPAGE is the calibration parameter for the condition. This should be a positive quantity.

The condition leads to no flow when the rock near the boundary is not fully saturated and to an outflow to the boundary which then has a zero pressure condition, when the rock is saturated.

The calibration parameter is used to scale the conductivity near the boundary. Note that the calibration parameter can be made time-dependent if required, as described in the following

paragraphs. With the introduction of the calibration factor, the flow equation used to solve the flow along the boundary legs (through the boundary nodes into the shaft) is written as follows:

$$Q = A \cdot \alpha \cdot K \frac{(H - h)}{L} \tag{8.7.1}$$

where:

Q – is the water flow through the boundary ($L^3 \cdot T^{-1}$)

A – is a surface area perpendicular to the flow (shaft wall) (L^2)

K – is the intact rock hydraulic conductivity ($L \cdot T^{-1}$)

H – is the prescribed head at the boundary node (=z) (L)

h – is the calculated head at the internal node connected to the boundary node (L)

L – is the distance between the boundary node and the internal connected node (L)

α – calibration factor (dimensionless)

The calibration parameter at a particular time is used to multiply the conductance of every leg from which flow enters the internal region. A value of less than 1.0 therefore reduces the flow, and can be used to represent wall effects caused by partial desaturation. A value greater than 1.0 could be used to represent the effect of a damaged zone. Since a single value is used for all legs regardless of their length or other properties, the calibration parameter should be regarded as a value that can be tuned to represent observations rather than something that can be calculated from a physical understanding.

Specifying the time-varying values

Time-dependent boundary conditions and point conditions can now be specified very flexibly. The format used in previous versions is still valid, but has been generalised as indicated here. The first line can have any number, indicating how many time values follow. Also, the first line can be a file name instead of a number. In this case the file contains the time and value lines (and the number is determined from the size of the file).

So, the following are all valid.

```
1
100.0
```

or

```
3
100    3.0
200    5.0
500    1.0
```

or

```
tsfile.txt
```

where tsfile.txt contains:

```
# this is a time-series file
# comment and blank lines are allowed
100    3.0
200    5.0
200    4.0
# note the repeated time for a discontinuous series
```

500 1.0

Note that the transient solver will be aware of discontinuities like that in the last example and will restart the solver to cope with them.

Example input file:

```
 TOP
ALL
FLUX
1
-3.24e-9

ONE
ALL
HEAD
1
140

TWO
ALL
HEAD
TOPOGRAPHIC

TOP
20
6615.138574032   -68850.9103331
6518.835977409   -69341.9928023
6702.50552514    -69656.8548841
7043.606113784   -70024.1939796
7142.797960652   -70173.6733308
4415.024141505   -69821.811034
4530.819637954   -69554.5906576
4325.950682699   -69198.2968224
4406.116795625   -69127.0380554
4584.263713238   -69340.8143565
4842.576743778   -69438.7951612
```

5270.12934605 -69278.4629353
5760.033369487 -68957.7984836
6214.308009401 -68200.6740838
6686.397341077 -67416.8276463
7135.239478833 -66573.859924
7246.238712116 -66780.0013573
6837.822221048 -67390.1056086
6855.63691281 -67710.7700603
6588.41653639 -68191.7667379
HEAD
TOPOGRAPHIC

8.7.4 Point Conditions

The interface will next ask the user to specify any point conditions such as source or sinks within the model region:

Name Point Conditions file :

This file defines point conditions (head or sources/sinks) which are to be applied during the flow calculation. It has the format shown in Table. 8.7.5. If no point conditions are required, then an empty file can be specified, or the prompt can be answered with a dot ('.').
The file may specify a number of points (for example, at x_1, y_1, z_1) at which either head or sink/source conditions are applied. If the keyword HEAD is chosen head values are set at the point to the time varying values specified. For setting sink/source values, use the keyword SOURCE, noting that a positive source is into the model region.
The format for specifying the time-varying values is as described for the boundary conditions.

Table. 8.7.5 The format of the point conditions input file.

Name	User-defined Suggested Name : *projectname_pt_conds.fai*
Format	*x_1 y_1 z_1* HEAD *… time-varying-values …* *x_2 y_2 z_2* SOURCE *… values …*

8.7.5 Flow Solver Output Definition

The flow solver output information is the last piece of data requested by the interface. The parameters defined in the flow solver output definition file are the locations at which outputs should be calculated. The times at which outputs should be calculated are defined in the flow solver control file (Section 8.3.6). The prompt provided is:

Name flow solver output file :

Since the head output file can potentially be extremely large, it is expected that the head field will be output relatively infrequently during a solution (at the times stipulated in the flow solver control file), with the monitor points/lines/planes and flux plane being used to give detailed information about the time dependent behaviour of the system and this file defines that output. Note that some outputs are automatically produced. In particular, a water balance report is written to the log file at each monitor output time. The flow solver output file has the format shown in Table. 8.7.6.

Table. 8.7.6 The format of the flow solver output file.

Name	User Defined (Suggested Name : *projectname*_output.fai)
Format	MONITOR_POINT *name x y z* MONITOR_LINE *name xs ys zs xe ye ze num* MONITOR_BHS *bhs_name num* MONITOR_PLANE *name vx vy vz wx wy wz* *x y z vn wn* FLUX_PLANE *name vx vy vz wx wy wz* x y z PATHLINE_POINT [SUMMARY] *name x y z t_start* PARTICLE_CLOUD [SUMMARY} *name x y z t_start num* REPORT_TOP_FLOWS REPORT_SHAFT_FLOWS GENERAL_REPORT SDF_DENSITY_LINE *name xs ys zs xe ye ze*

Example input file:

MONITOR_LINE

MIU-1 5488.83 -68629.36 220.07 5488.83 -68629.36 -779.93 10

MONITOR_PLANE

MIUsouth 800 0 0 0 0 1400

5100 -69200 -1000 16 28

```
FLUX_PLANE
MIUsouth 800 0 0 0 0 1400
5100 -69200 -1000
PATHLINE_POINT
1 5500 -68575 -400 0
```

A variety of flow solver outputs can be specified in FracAffinity. The main output types which can be specified are: monitor points, monitor boreholes, flux planes and pathlines. There are two additional types which can be specified: monitor lines and monitor planes, which are intended as quick ways of generating large numbers of monitor points in a line or a plane. Each output type has a keyword, which appears on a line by itself, followed by its relevant parameters on the next line. The parameters are as follows:

MONITOR_POINT

The monitor point name is followed by the x, y and z coordinates of the point.

MONITOR_LINE

The monitor line name is followed by the x, y and z coordinates of the start of the line (xs, ys, zs) and the x, y and z coordinates of the line end (xe, ye, ze) The final parameter is the number of monitor points which should be evenly distributed along the line.

MONITOR_BHS

The borehole section name is followed by the number of points required. The actual points used will be selected from the nodes along the borehole section at regular intervals. To select all nodes the number of points should be omitted.

MONITOR_PLANE

Each plane is restricted to a parallelogram with a corner at the position (x,y,z), and sides extending from the corner along the vectors (vx,vy,vz) and (wx,wy,wz). The parameters vn and wn are used to define the number of monitor points which are required in each of the directions specified by the first and second vector respectively.

FLUX_PLANE

The information provided is similar to the Monitor Plane. However, the parameters vn and wn are not required because the flux plane is not discretised into monitor points. The output of flux plane is the average flux through the plane.

PATHLINE_POINT

Each pathline has a name and a start location (x,y,z). Additionally, the time at which the pathline should start, t_start, must be provided. The optional keyword SUMMARY can be used to give simply a list of times and locations, otherwise details of the path are given – see Section 8.8.3(4). Two other optional keywords can be used, separately or together.

The SMOOTH keyword gives a pathline that is not restricted to follow the network legs. This will be smoother and more realistic than the default cases, but care should be taken in interpreting the properties along the path which are those of the nearby network leg. If SMOOTH is not specified a simple pathline is computed in which the pathline follows the strongest flow at each node.

The REVERSE keyword produces a path that goes back in time. The times given are negative with the flows reported being for actual flow field. Note that in the time-dependent case, the flow field at the start time is used.

PARTICLE_CLOUD

A particle cloud is a collection of paths. In a particle cloud, the route taken at each node is randomly selected in proportion to the flows. Each particle cloud has a name and a start location (x,y,z). Additionally, the time at which the pathline should start, t_start, and the number of particles to track, num, must be provided. The optional keyword SUMMARY can be used in the same way as for PARTICLE_POINT.

REPORT_TOP_FLOWS

No further data is required for this keyword. A file, TopOutflow.faout, giving the flows at each top-surface node is written with a block of data for each monitor time.

REPORT_SHAFT_FLOWS

No further data is required for this keyword. A file, ShaftOutflows.faout, giving the flows at each shaft-surface node is written with a block of data for each shaft at each monitor time. This also applies to excavations.

GENERAL_REPORT

If this keyword is present, a report is written for every out output time giving the geometric and solution information in a single file. The files are called result_n.faout with n starting at 0. The content of the files is described in Section 8.8.3(1).

SDF_DENSITY_LINE

The line name is followed by the x, y and z coordinates of the start of the line (xs, ys, zs) and the x, y and z coordinates of the line end (xe, ye, ze). A report is written to a file called sdf_density.faout, giving the number of intersection and density information. If there are several lines then each has its own SDF_DENSITY_LINE header and all are written to the same file.

Once the flow solver output file has been specified, the interface will automatically start FracAffinity and exit, leaving FracAffinity running in the background. The progress of the FracAffinity run can be assessed by looking at output that FracAffinity writes to the terminal and the FracAffinity log file (see Section 8.8).

Note that if FracAffinity fails to start because the FRAC_AFFINITY_PATH environment variable is not correctly set, then it can be run directly from the FracAffinity directory. This also allows FracAffinity to be rerun without rerunning the interface (for example to make a small change in an input file), but it should be remembered that files in the FracAffinity directory will be overwritten if the interface is run again later.

8.8 Output Files

The output files from the FracAffinity Interface must be of the correct format to be input files to FracAffinity. The FracAffinity Interface also generates a log file that is relatively easily read by the user, and is useful for checking and QA purposes.

8.8.1 FracAffinity Input Files

Table. 8.8.1 lists all the FracAffinity input files that may be generated by the FracAffinity Interface. This table also lists the names of output files from FracAffinity which are written to the FracAffinity directory after successfully running FracAffinity. A distinction is made between FracAffinity files with specified names and those with user-defined names referred to in FracAffinity input files (written in italic font).

8.8.2 Log File

The log file, called FA.log, which is stored in the FracAffinity directory, contains a record of the FracAffinity run. The information in this file includes an echo of the key input parameters, diagnostic information such as the number of nodes in the problem, the number of SDFs generated and water balance reports (see Section 8.8.2(1)). The start and end date and time are written to the file. An example of the information in the log file is given in Appendix A.

Most of the information in the log file is self-explanatory. An exception is the water balance report which is described in the following section.

(1) Water Balance Report

A water balance report is written for each monitor output time, as defined in the flow solver control input file (see Section 8.3.6). This reports the instantaneous situation at the time. The format of the water balance report for the steady and transient cases, and for the saturated and unsaturated cases are all basically the same, but some entries are not used in each case.

The total water in the system is reported, using the nodal volumes, porosities and saturation values. This is followed by a breakdown of water flows (volumes per unit time). This is presented by “boundary”, which includes the external and internal physical boundaries but also includes storativity and resaturation effects. Where a physical boundary has more than one type of boundary condition, the different parts of the boundary are reported separately.

For each row of the table, the flow rate into the model and flow rate out of the model are presented separately. The net flow rate out of the model is also reported. These flows are calculated simply by summing the flows in connections to the given boundary.

Note that for galleries, shafts and excavations, flow into a gallery, shaft or excavation is regarded as flow out of the model and is positive (in FracAffinity, flows into the model from boundaries are negative). An example log file is provided in Appendix A.

For point-condition nodes, the specified flow is reported, or the flow that is calculated for a specified head condition.

For borehole sections, similarly, the specified flow is reported, or the flow that is calculated for a specified head condition.

For transient cases, the flows to and from storativity and resaturation are reported. These are inferred from the non-zero net flows at each node. When point-condition and borehole conditions are applied, the flows will contribute to storativity initially – the rate at which this effect falls off will be determined form the storativity time-scales, either implied by the data or imposed by the user.
The final total would be zero in an exact calculation. For a steady-state calculation, a non-zero value indicates that the solution is not fully converged. For a transient calculation, the total should only differ from zero by rounding errors, since any local non-zero net flows are interpreted as being a storativity effect.

8.8.3 FracAffinity Output Files

The FracAffinity run will generate a number of output files depending on the type of run and outputs requested. In general these can be processed by the FracAffinity Interface as discussed in Section 8.9.

Table. 8.8.1 FracAffinity input files that may be generated by the FracAffinity Interface.

Name	Note
control.fa	General control file
zones.fa	Stratigraphic information
region.fa	Model region information
grid_parm.fa	Intact rock grid parameters
zone_cond.fa	Intact rock conductivity parameters
zone_por.fa	Intact rock porosity parameters
zone_ssc.fa	Intact rock storage parameters
conductivity.fa	Intact rock conditioning data
ddf_geom.fa	DDF geometry
ddf_grid_parm.fa	DDF grid parameters
ddf_cond.fa	DDF conductivity parameters
ddf_por.fa	DDF porosity parameters
ddf_ssc.fa	DDF storage parameters
ddf_conductivity.fa	DDF conductivity conditioning data
sdf.fa	SDF parameters
sdf_grid_parm	SDF Grid Parameters
flowsolver.fa	Flow solver control data
FlowSolverOutput.fa	Outputs required from flow solver
boundc.fa	Boundary conditions
sources.fa	Source/point conditions

initc.fa	Initial conditions
sdf_data.faout	SDF summary data
monpnt.faout	Monitor data
fluxplane_n.faout	Flux plane data
pathline.faout	Pathline data
fluxshaft.out	Cumulative flow to excavations
TopOutflow.faout	Point by point flow across the top surface
ShaftOutflows.faout	Point by point flow to shafts
properties.faout	Re-useable hybrid medium
roc.name	Re-useable intact rock model
head_n.faout	Re-useable head data for initial conditions
result_n.faout	General report on the solution
head_best.faout	Best head solution for unsaturated calculation
other copied files	Other user specified files (e.g. for water retention models and borehole definitions) are copied into the directory.

(1) General Report Files

In addition to the head_n.faout file, a general output file is now available that gives both data and results for all nodes and legs. This is written if the GENERAL_REPORT keyword is present in the flow output control file. The file is called result_n.faout, where n is 0, 1, 2 etc for the output times (or just 0 for a steady state run).

The file has three blocks of information, one for nodes, one for boundary nodes, and one for legs. These are preceded by a list of material names with numbers that are used in the rest of the output.

So the file has four types of line.

- For each material: Mnnn, MaterialName, Type (Intact/SDF/DDF).
- For each node: Nnnn, x, y, z, hcon, por, sto, vol, Mnnn, h, sat, dv, dvx, dvy, dvz, flow_n.
- For each boundary node: Bnnn, x, y, z, type, value.
- For each leg: Lnnn, Nnnn1, Nnnn2, len, area, hcon, por, ap, flow, dvel, vel.

If the leg connects to a boundary node then the second node will read Bnnn. Internal node numbers run from N0 and boundary node numbers run from B0. The information in the result file output is derived as shown in Table. 8.8.2 and Table. 8.8.3.

Table. 8.8.2 Information contained in the General Report File.

Item	**Description**
Nodes	
x, y, z	Node position as determined by the gridding process.
Hcon	Is the hydraulic conductivity at the node, as determined by the input data.
Por	Is the porosity at the node, as determined by the input data.

Item	Description
Sto	Is the storage coefficient at the node, as determined by the input data.
Vol	Is the volume associated with the node, as determined by the gridding process.
H	Is the calculated head at the node.
Sat	Is the calculated saturation at the node (1.0 for saturated cases).
Dv	Is the magnitude of the effective Darcy velocity at the node.
dvx, dvy, dvz	Is an effective Darcy velocity vector at the node. This is calculated from a flow-weighted average over all out-flowing legs.
flow_n	Is the flow through the node. This is calculated as half of the sum of the absolute flows in all legs connected to the node.
Boundary Nodes	
x, y, z	Node position as determined by the gridding process.
Type	Boundary condition type (see below for a full list)
Value	Value for associated boundary condition
Legs	
Len	Is the length of the leg as determined by the gridding process.
Area	Is the cross sectional of the leg as determined by the gridding process. This is an average over the two halves of the legs (weighted by their lengths).
Hcon	Is the effective hydraulic conductivity of the leg. This is a harmonic average over the near- and far-sections of two halves of the legs (weighted by their lengths).
Por	Is the effective porosity of the leg. This is an average over the two halves of the legs (weighted by their lengths).
Ap	Is the effective aperture of the leg. This is an average over the two halves of the legs (weighted by their lengths). Porous legs have zero aperture.
Flow	This is the calculated volumetric flow in the leg.
Dvel	This is the Darcy velocity, equal to the flow divided by the average area.
Vel	This is the pore velocity, calculated as Darcy velocity over average porosity. Note that the velocity in the pathline output is calculated using a different average and so will not always be the same as this.

Table. 8.8.3 Possible boundary condition types and the meaning of the value as written to the General Report File.

Boundary Condition Type	Value given
FLUX	Outgoing water volume per unit area per unit time [$m^3/m^2/s$]. For boreholes the total volume per unit time is given [m^3/s].
HEAD	Head [m].
TOPOGRAPHIC	Head [m].
ATMOSPHERIC	Head [m] – will always be 0.0.
REUSE_HEAD	Head [m].

SUCTION	Suction [m], a positive value corresponding to an imposed negative head value.
RECHARGE	Incoming water volume per unit area per unit time [$m^3/m^2/s$].
SOURCE	Incoming volume water per unit time [m^3/s].

(2) SDF Summary File

In addition to the files that can be post processed, a summary file for the SDFs is generated automatically. This file is called SDF_data.faout and in a comma-separated file (so that can be imported into Excel for example). The file has a header line and one line for each SDF, which provides the information shown in Table. 8.8.4.

Table. 8.8.4 Information on SDFs written to the SDF summary file.

Header line	Description
Xc	x-coordinate of the SDF centroid
Yc	y-coordinate of the SDF centroid
Zc	z-coordinate of the SDF centroid
dip-angle	The angle below the horizontal in degrees (0 to 90) for the steepest descent on the SDF plane
dip-direction	The angle clockwise from North in degrees (0 to 360) for the steepest descent on the SDF plane
rotation-angle	The angle by which the principal axis of the SDF is rotated from the line of steepest descent, in degrees (-45 to 45)
Length	The length of the principal axis of the SDF.
Width	The width of the SDF
log-cond	The logarithm base-10 of the conductivity
Porosity	The porosity
Tortuosity	The tortuosity
storage-coeff	The specific storage coefficient
Aperture	The aperture
Ai	The in-plane anisotropy factor
An	The normal anisotropy factor
Enhancement-factor	The enhancement factor for enhanced conductivity intersections

(3) Top Surface Outflow File

If requested, using the REPORT_TOP_FLOWS keyword (Section 8.7.5), a file, TopOutflow.faout, is written. A block is appended for each monitor time. The file lists every boundary point on the top surface and gives the location, flow and flux (flow/area) at that point. An example of the start of such a file is shown below.

Top Surface Outflows at time 0

x, y, z, total flow, flux (flow/area)

5708.27, -69067.4, 191.823, 4.83241e-05 , 4.24621e-09

5398.93, -69067.4, 215.753, 6.89448e-06 , 5.26334e-09

5377.89, -69067.4, 219.594, 3.17666e-06 , 4.36968e-09

5363.73, -69067.4, 223.207, 3.3609e-06 , 2.82743e-09

(4) Shaft Outflow File

If requested, using the REPORT_SHAFT_FLOWS keyword (Section 8.7.5), a file, ShaftOutflows.faout, is written. A block is appended for each shaft or excavation and monitor time. The file lists every active boundary point on the shaft surface and gives the location, flow and flux (flow/area). The location is provided as (x,y,z) coordinates and (r,theta) coordinates relative to the shaft axis. So, r is the distance from the shaft centre-line (equal to the radius of the shaft for the wall) and theta is the angle (in degrees clockwise from north) relative to the shaft centre-line. The total flow into the shaft or excavation is provided at the end of each block of data.

The "total flow" is the main result, giving the flow to the shaft at this position. "flux (flow/area)" gives this total divided by the area of the leg (note that this area is not adjusted for the geometry of the shaft and so this value is less accurate than the total flow. Finally a "type" is given, this is either an intact or a discrete feature type, indicating what the feature giving the flow was.

An example of part of such a file is shown below.

Excavation Shaft1(shaft1) Outflows at time 432000

x, y, z, total flow, flux (flow/area), type

-1.625 , -1.625 , -3 , 5.86817e-007 , 5.55567e-008 , rock1

-1.3 , -2.83125 , -3 , 8.03744e-007 , 4.94612e-008 , rock1

-2.83125 , -1.3 , -3 , 8.03744e-007 , 4.94612e-008 , rock1

-1.625 , 1.625 , -3 , 5.86817e-007 , 5.55567e-008 , rock1

-1.3 , 2.83125 , -3 , 8.03744e-007 , 4.94612e-008 , rock1

-2.83125 , 1.3 , -3 , 8.03744e-007 , 4.94612e-008 , rock1

1.625 , -1.625 , -3 , 5.86817e-007 , 5.55567e-008 , rock1

2.83125 , -1.3 , -3 , 8.03744e-007 , 4.94612e-008 , rock1

1.3 , -2.83125 , -3 , 8.03744e-007 , 4.94612e-008 , rock1

1.625 , 1.625 , -3 , 5.86817e-007 , 5.55567e-008 , rock1

2.83125 , 1.3 , -3 , 8.03744e-007 , 4.94612e-008 , rock1

1.3 , 2.83125 , -3 , 8.03744e-007 , 4.94612e-008 , rock1

Total Outflow at time 432000 is 8.77722e-006

(5) Pathline Files

The output files produced in pathline calculations provide detailed information about the path. If the keyword SUMMARY is included in the control file a simple summary output is produced. This keyword goes on the PATHLINE-POINT or PARTICLE-CLOUD header line. The summary is simply a set of times and coordinates:

t, x, y, z

which is produced at each node and each leg mid-point.
The full output is produced at the same locations and reads:

t, x, y, z, ts, dist, len, v, Dv, flow, area, Mat, hcon, por, ap, cumF

where each item provides the information specified in Table. 8.8.5 and illustrated in Fig. 8.8.1.
When the SMOOTH option is used, the coordinates reported will not be at a node, but the properties reported are taken from the associated (nearby) node and leg.

Table. 8.8.5 Information written to the pathline file.

Item	**Description**
Ts	is the time for the step [s] (=len/vel)
Dist	is the total path-length so far [m]
Len	is the length of the step (that lead to the given position) [m]
V	is pore-velocity for the step [m/s] (=flow/area)
Dv	is Darcy-velocity for the step [m/s]
Flow	is the volumetric flow rate for the step [m^3/s]
Area	is the cross-sectional area for the step [m^2]
Mat	is the material at location (as a name)
Hcon	is the hydraulic conductivity for the step [m/s] (harmonic average)
Por	is the porosity for the step [-] (average)
Ap	is the aperture for the step [m] (0 if porous) (average)
CumF	Cumulative F-ratio. This is the time-integrated specific-surface-area (per unit volume of pore-water). Thus, for each step the cumulative F-ratio is increased by the travel-time divided by half the aperture. For porous legs there is no specific surface and so the cumulative F-coefficient is not changed.

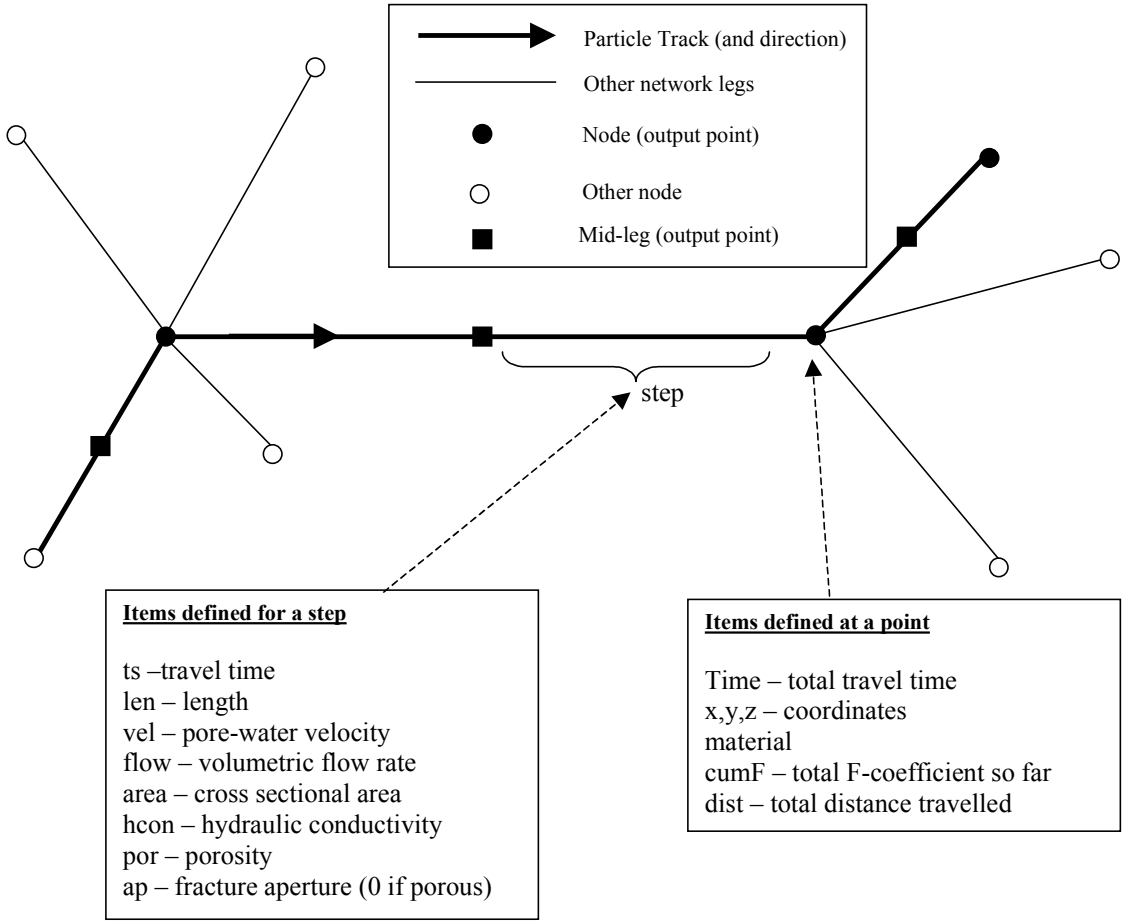


Fig. 8.8.1 Definition of items in the pathline file.

8.9 Post-Processing for EarthVision

The purpose of using the FracAffinity Interface in post-processing mode is to enable the user to select FracAffinity output files either for direct analysis or for visualisation in EarthVision, without the user having to understand the naming convention and format of FracAffinity output files. Information about how these files may be viewed in EarthVision itself is beyond the scope of this document, and the user is referred to the EarthVision documentation for such information [4].

8.9.1 Starting the post-processor

The FracAffinity Interface is started in post-processing mode by typing fainterface -output at the command line. The user is immediately prompted for the FracAffinity directory to which the user should supply a directory name. This is the FracAffinity directory inside which all the FracAffinity output files to be analysed are contained.

```
==================================
FracAffinity Interface
Version 3.2
November 2006
==================================
Name FracAffinity directory :
```

When the directory has been specified, the interface will give a message that is checking the output files in the named directory. The user will then be asked to select what form of FracAffinity output they wish to analyse:

```
============================
Properties          [1]
Head/Sat/Velocity   [2]
Monitor Points      [3]
Monitor Lines       [4]
Monitor Planes      [5]
Flux Planes         [6]
Pathlines           [7]
Network             [8]
SDFs                [9]
Monitor BHSs        [10]
Excavation Props    [11]
Quit                [0]
============================
Select Output [1]-[11]:
```

The selection is made by typing the number of the required choice and pressing return. The interface will then provide the user with additional selections depending on the type of output requested and what is available in the directory. The "Quit" option will cause the interface to exit immediately.
If an error occurs at any stage, such as file not being available for output, the interface will ask the user whether they would with to "Try again or Quit [t/Q]?". Entering "Q" followed by return will cause the interface to exit. Any other entry will allow the user to continue and correct the previous problem (for example by entering a new file name). If an output file is specified that already exists, the interface will ask if the file is to be replaced. Answer “y” to replace the file, “n” to specify a different name.

8.9.2 Rock Properties Output

If Properties is selected, the interface will inform the user of the data which is available for visualisation. This may be Intact Rock, Deterministic Discrete Features, Stochastic Features or the full Hybrid Medium.

```
============================
[0] Return to main menu
[1] Intact rock
[2] Deterministic discrete features
[3] Stochastic discrete features
[4] Hybrid rock
============================
Specify Output [0]-[4] :
```

The user is then shown the available output times and prompted for a time at which the conductivity, porosity and specific storage coefficient will be written to a named EarthVision file for visualisation.

```
========================================
Solver Start Time          :0
Solver End Time            :1e7
========================================
Specify Output Time            :1e5
Specify Output Filename   :projectname_run_props.pdat
```

The FracAffinity Interface will read the properties data file properties.faout and write the required information in a format suitable for EarthVision visualisation. Note that the properties.faout file is binary. An ASCII version can be produced by saving the hybrid rock model but this reports the data is a compact format that is not user-readable.
The option to return to the main menu is provided in case an option is selected accidentally. This allows the user to quickly correct the error and return to make the correct selection.

8.9.3 Head and Velocity Output

If the Head/Sat/Velocity option is selected, the user is first asked whether head, saturation or velocity vector output is required. The user is then prompted with a list of output times for which the data is available, is asked to select a suitable output time, and is asked to specify a name for the file where data is written.

```
Times
========================================
0
500
1000
========================================
Specify Output Time          : 500
Chosen Time 500

Specify Filename             :projectname_run_head.pdat
```

The FracAffinity Interface will read the appropriate head, saturation or velocity information (at the output time nearest to the user's selection) from the FracAffinity output file head_n.faout and write a file with the given name in a format suitable for EarthVision visualisation.

The velocity vectors may be used to visualise flow vectors in EarthVision using a binary script called ran2vec. The scripts are in the tar file arrowprogs99.tar.

The script and auxiliary files must be in the user directory to run. Although flow vectors for the full Three-D field may be visualised it is recommended that only sections (i.e. cross-sections or horizontal sections) are visualised. This requires editing the file output from the interface using the following steps:

1. Select a value for the cross-section by using the 3D Viewer visualising the file output from the FracAffinity Interface.
2. Use the EarthVision Formula Processor to give unwanted nodes NULL values (1e+20). This requires the use of the relational functions in the Formula Processor.
3. Re-save the file in the 3D Viewer, this removes the lines with the NULL nodes from the file.
4. Multiply the values so that they give values which may be interpreted by the visualisation script (i.e. > 1).

To begin the visualisation script type the following command at the command line interface:

ran2vec

The following is an example of the prompts written to the screen and the responses used to generate a binary grid file.

```
*********************************************
Enter the input location/component filename:
vect4x_BCA.pdat
Default output vector file name is bw-vec.3grd.
Do you want to use a different filename? (Y or N)
N
The default number of colors is 8. How many colors do
you want to represent the arrow length range (8 max)?
8
Current arrow colors (short to long) are:  6 7 2 5 8 4 1 3
Enter Y if you want to change them.
N
Shortest vector =          4.50176 Longest vector = 190874144.00000
Please enter the minimum and maximum arrow length in X,Y,Z data scale units
(with decimal points).
100 200
Bin   1, vectors from       4.50 to ********   units long, contains      604 vectors.
Bin   2, vectors from ******* to *******   units long, contains        3 vectors.
Bin   3, vectors from ******* to *******   units long, contains        2 vectors.
Bin   4, vectors from ******* to *******   units long, contains        0 vectors.
Bin   5, vectors from ******* to *******   units long, contains        0 vectors.
Bin   6, vectors from ******* to *******   units long, contains        0 vectors.
Bin   7, vectors from ******* to *******   units long, contains        0 vectors.
Bin   8, vectors from ******* to *******   units long, contains        2 vectors.
Do you want to use the even intervals reflected in _bins above,
or would you like to select irregular intervals?
Enter 1 for even Intervals, Enter 2 for irregular intervals.
1
Number of vectors written =      611
*************************************************
```

The output file can be visualised alongside a geological model by typing ctrl-z when the 3D Viewer is open and typing the binary 3D grid file name at the prompt.

8.9.4 Monitor Points

If the Monitor Points option is selected. The interface will provide a list of monitor points for which the data is available. An example is shown below.

```
Monitor Points are at:
========================================
[1] Point1 3000 -68000 0
[2] Point2 3000 -68000 -100
[3] Point3 3000 -68100 100
```

```
========================================

Select Monitor Point [1]-[3]:
```

The user is asked to select the monitor point(s) for which output is required, and then asked to specify a name for the EarthVision format file where the data is to be written. The suggested format is projectname_run_monitor_name.pdat. The monitor point selection may be a list of numbers separated by spaces or commas, or a single value, e.g. 1 or 1,3 or 1 2 3 would all be acceptable for the above example.
The FracAffinity Interface will read the FracAffinity monitor output file monpnt.faout and write the information for the selected monitor points in an EarthVision format file.

8.9.5 Monitor Lines or Borehole Sections

If the Monitor Lines or Monitor BHS[3] option is selected, the interface will provide a list of monitor points for which the data is available. An example is shown below.

```
Monitor lines are at:

========================================

[1] MonLine1 3500 -68000 -400 3600 -68000 -400 5


========================================

Select Monitor Lines [1]-[1]:
```

The user is asked to select the monitor line(s) for which output is required, and then asked to specify a name for the EarthVision format file where the data is to be written. The suggested format is projectname_run_monitor_name.pdat. As for the monitor points the monitor line selection may be a list of numbers separated by spaces or commas, or a single value. For monitor BHS(s) the borehole section names are given.
The FracAffinity Interface will read the FracAffinity monitor output file monpt.faout and write the information for the each of the monitor points which comprise the monitor line, in an EarthVision format file.

8.9.6 Monitor Planes

If the Monitor Planes option is selected, the interface will provide a list of monitor planes with their parameters (as specified in Section 8.7.5) for which the data is available. An example is shown below.

```
Monitor Planes are at:

========================================

[1] Plane1 100 0 0 0 100 0
3000 -68000 -1000 3 3
```

[3] Monitor BHS is Option 10 in the FracAffinity Interface.

==========

Select Monitor Planes [1]-[1]:

The user is asked to select the monitor plane(s) for which output is required, and then asked to specify a name for the EarthVision format file where the data is to be written. The suggested format is projectname_run_monitor_name.pdat. As for the monitor points and lines the monitor plane selection may be a list of numbers separated by spaces or commas, or a single value.

The FracAffinity Interface will read the FracAffinity monitor output file monpnt.faout and write the information for the each of the monitor points which comprise the monitor plane, in an EarthVision format file.

8.9.7 Flux Planes

If the Flux Planes option is selected, the interface will provide a list of monitor planes with their parameters (as specified in Section 8.7.5) for which the data is available. An example is shown below.

Flux Planes are at:

==========

[1] FP1 3000 -68000 -1000 3100 -68000 -1000 3000 -68000 -900

==========

Select Flux Planes [1]-[1]:

The user is asked to select the flux plane(s) for which output is required, and then asked to specify a name for the EarthVision format file where the data is to be written. The suggested format is projectname_run_flux_name.pdat. As for the monitor points, lines and planes the flux plane selection may be a list of numbers separated by spaces or commas, or a single value.

The FracAffinity Interface will read the appropriate FracAffinity flux plane output file(s) and write the information for the each of the flux planes in a form suitable for viewing in EarthVision.

8.9.8 Pathlines

If the Pathlines option is selected, the interface will provide a list of starting points of pathlines for which the data is available. An example is shown below.

Pathline starting points are at:

==========

[1] Path1 3000 -6800 -1000 0

[2] Path1 3000 -6800 -1500 0

[3] Path1 3000 -6800 -500 0

==========

Select Pathline start point [1]-[3]:

The user is asked to select the pathline(s) for which output is required, and then asked to specify a name for the EarthVision format file where the data is to be written. The suggested format is projectname_run_path_name.pdat. As for the previous output formats. the selection may be a list of numbers separated by spaces or commas, or a single value.
The FracAffinity Interface will read the appropriate FracAffinity pathline output file(s) and write the information for the each of the pathlines in a form suitable for viewing in EarthVision.

8.9.9 Network

If the Network option is selected, the interface will inform the user of the data which is available for visualisation. This may be Intact Rock, Deterministic Discrete Features, Stochastic Features or the full Hybrid Medium.

==========

[0] Return to main menu

[1] Intact rock

[2] Deterministic discrete features

[3] Stochastic discrete features

[4] Hybrid rock

==========

Specify Output [0]-[4] :

The FracAffinity will then prompt for an output file name and write the network information to a file. If there are excavations in the model, then FracAffinity will prompt for the time at which the network is to be produced. Connections that are inactive at that time will be given the null value of 1e-20 for the conductance so that they are easily visualised in EarthVision.
The network may be visualised in EarthVision following post-processing using the *frac.pl* post-processing perl script. The script is run from the command line using the following format:

frac.pl input_file output_file.pdat

The suggested format for the input file is projectname_network_run.faout. The suggested name for the output file is projectname_network_run.pdat. Once the programme has been run the FracAffinity network can be visualised in EarthVision.

8.9.10 SDFs

If the SDFs option is selected, FracAffinity will prompt for a filename and write the coordinates of the four corners of each SDF to an ascii file.
The SDF data may be visualised in EarthVision following post-processing using two awk scripts called fractFA2evfiles.awk and generate_ev_fractures.awk. The format for running the scripts is as follows:

fractFA2evfiles input_file

The suggested name for the input file is projectname_SDFs_run.faout. It is intended that the SDF model which is output from the script will be merged with a face file representing the geological model of the intact rock and DDFs. When the script is run the number of SDFs to be visualised will be listed and then the following information needs to be entered at the interface when prompted:

- Name output faces file: enter the name of the faces file to which the SDF model will be written (e.g. SDFs.faces)
- Enter xmin: Enter the minimum x value of the geological model with which the SDFs.faces will be merged.
- Enter xmax: Enter the maximum x value of the geological model with which the SDFs.faces will be merged.
- Enter ymin: Enter the minimum y value of the geological model with which the SDFs.faces will be merged.
- Enter ymax: Enter the maximum y value of the geological model with which the SDFs.faces will be merged.
- Enter zmin: Enter the minimum z value of the geological model with which the SDFs.faces will be merged.
- Enter zmax: Enter the maximum x value of the geological model with which the SDFs.faces will be merged.
- Enter number of x slices: Enter the number of x slices in the geological model with which the SDFs.faces will be merged.
- Enter number of y slices: Enter the number of y slices in the geological model with which the SDFs.faces will be merged.
- Enter number of z slices: Enter the number of z slices in the geological model with which the SDFs.faces will be merged.

The result of this script is a single faces file with the user defined name entered in response to further command line prompt. This single faces file can be merged with any other faces file (i.e. a geological model) with the same gridding parameters. The merge uses the Faces File Merging program in EarthVision. The geological model faces file should be entered first with the zone and block numbers given values of -1. The SDF faces file should be entered second the zone and block numbers given values of 1. A result from merging an SDF faces file with a geological model faces file is illustrated in Fig. 8.3.2.

8.9.11 Monitor BHSs

See Section 8.9.5.

8.9.12 Excavation Properties

The Excavation Properties option allows a file to be written to visualise the excavation against time.

If this is selected, the interface will inform the user of the data which is available for visualisation. This may be Intact Rock, Deterministic Discrete Features, Stochastic Features or the full Hybrid Medium.

```
============================
[0] Return to main menu
[1] Intact rock
[2] Deterministic discrete features
[3] Stochastic discrete features
[4] Hybrid rock
============================
Specify Output [0]-[4] :
```

The user is then asked whether all nodes should be reported or just those that are excavated. A filename is the requested and the information is written to this file. If the output file is saved with *pdat* as the file type, it can be visualised immediately using the Animation menu in the 3D Viewer.

This contains information of all the nodes, ordered by excavation (with unexcavated ones first if requested).

Within an excavation the nodes are ordered by excavation time.

For each node, the output line contains the following items:

- x y and z coordinates;
- porosity (as percentage);
- Hydraulic Conductivity (m/s);
- Specific storage coefficient;
- LogK;
- Relevant Excavtion;
- Time.

9. Summary

GEOMASS provides a powerful, efficient and flexible modelling system for modelling of the geological and hydrogeological environment. The system provides several novel approaches for integrated modelling of the geological and hydrogeological environment, including:

- The coupling of geological and hydrogeological modelling and visualisation in a fully integrated fashion, providing a sophisticated platform for investigating the impact of geological structure on groundwater flow.
- The hybrid medium approach to groundwater flow modelling adopted in FracAffinity.
- Routines in FracAffinity for modelling heterogeneity and time dependence (e.g. dynamic shaft excavation modelling).
- The ability to model unsaturated flow on regional scales.

GEOMASS has been developed by JAEA in the MIU Project, in which a URL is being constructed in a fractured granite. However, GEOMASS provides an approach to integrated geological and hydrogeological modelling and visualisation that is suitable for site investigations in all types of host rock, and for projects unrelated to radioactive waste disposal.

Acknowledgement

The GEOMASS system is developed with the support of Dr. Matthew J. WHITE (Galson SCIENCES Ltd.), Dr. Jordi GUIMERÀ (AMPHOS XXI Consulting, S.L.), Dr. Peter ROBINSON (Quintessa Ltd.), Dr. Kaoru INABA (TAKENAKA Corporation), Mr. Hironori ONOE (NEWJEC Inc.) and Mr. Hiroshi KOSAKA (Japan Atomic Energy Agency). The authors would like to acknowledge them. The authors also would like to thank the text-editing operations of this manual of Mr. Yoshinobu ENDO (NESI Inc.).

References

1) Long J. C. S., H. K. Endo, K. Karasaki, L. Pyrak, P. MacLean and P.A. Witherspoon, (1985). Hydrological Behaviour of Fracture Networks. Hydrol. of Rocks of Low Permeability. IAH Conf, Vol. XVII, pp.449-462.

2) Robinson P. C., (1981). Connectivity of Fracture Systems - A Percolation Theory Approach. J. Phys. A. 16. p.605.

3) Cacas M. C., E. Ledoux, G. de Marsily, A. Barbreau, P. Calmelld, B. Guilland and R. Margritta, (1990). Modelling Fracture Flow with Stochastic Discrete Fracture Network: Calibration and Validation 2. The Transport Model. Water Resour. Res. 26(3) 491-50.

4) Dynamic Graphics (1997). EarthVision User's Guide 5.0. Alameda, California.

5) White, M.J. (1998). Visualization of the El Berrocal Granite: Application to Rock Engineering. Engineering Geology, 49, pp.185-194.

6) Harding, T.P. and Lowell, D., (1979). Structural styles, their plate tectonic habits, and hydrocarbon traps in petroleum provinces. American Association of Petroleum Geologists Bulletin 63, pp.1016-1058.

7) Yamada, Y. and McClay, K.R., (2002). Analogue experiments of 3D inversion structures above listric fault systems. American Association of Petroleum Geologists Memoir. Thrust Tectonics and Petroleum Systems. (In press).

8) Buchanan, P.G. and Nieuwald, D.A., (1995). Modern Developments in Structural Interpretation, Validation and Modelling. Geological Society of London Special Publication 99, Geological Society, London.

9) Gibbs, A.D., and Griffiths, P.A., (1996). 3D Kinematic structural restoration and its impact on reduction of risk - new techniques arising from commercialisation of software developed for 3D volume balanced analysis of geological structure. In the Strategic Importance of Oil and Gas Technology, Edinburgh, pp.272 -282.

10) van Genuchten M., (1980). A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. Soil Sci. Soc. Am. J., 44, pp.892-898.

11) Vogel T, van Genuchten M.T., and Cislerova M. (2001). Effect of the shape of the soil hydraulic functions near saturation on variably-saturated flow predictions. Advances in Water Resources vol. 24 pp.133-144.

12) Ippisch O, Vogel H-J and Bastian P. (2006). Validity limits for the van Genuchten-Mualem model and implications for parameter estimation and numerical simulation. Advances in Water Resources vol. 29 pp.1780-1789.

13) Byrne G. D. and A.C. Hindmarsh, (1975). "A Polyalgorithm for the Numerical Solution of Differential-Algebraic Equations", ACM Trans. Math. Software 1, pp.71-96.

14) White, M.J. and del Olmo, C., (1996). The Application of Geological Computer Modelling

Systems to the Characterisation and Assessment of Radioactive Waste Repositories. Proceedings of the International Conference on Deep Geological Disposal of Radioactive Waste, Canadian Nuclear Society, Winnipeg, Manitoba, Canada, pp.3-235 - 3-244.

15) Peitgen, H.O. and Saupe, D., 1988. The Science of Fractal Images. Springer-Verlag.

16) White, M.J., Clark, K., Koide, K., Ogata, N., Takase, H. and Takeuchi, S. (1997). 3D Modelling of Heterogeneous Rock Mass Properties, Tono, Japan. In: Reeves, G. M. (ed.), Geo-Engineering of Hazardous and Radioactive Waste Disposal, Pre-prints of Papers from the 3rd European Engineering Geology Conference, Newcastle, 10-14 September.

17) de Marsily G., (1986). Quantitative Hydrogeology, Academic Press.

18) Saegusa, H. White, M.J., Robinson, P. and Guimerà, J. (2006). Development of a System for Integrated Geological Modelling and Groundwater Flow Simulation. In Proceedings of the International High-Level Radioactive Waste Management Conference, Las Vegas, May 2006.

19) Carrera J., L. Vives; P. Tume; M. Saaltink; G. Galarza; J. Guimerà and A. Medina, (1996). Interpretation of field tests in low permeability fractured media. Recent experiences. In: Duchatean, (ed) Parameter identification and inverse problems in hydrogeology, geology and ecology. Water Sci. and Tech. Library, 23:53-70 Kluwer Ac. Press.

20) Abelin H., L. Birgersson, H. Widén, T. Ägrén, L. Moreno and I. Neretnieks, (1994). Channelling Experiments in Crystalline Fractured Rocks. J. Cont. Hydrol. 15:129.

21) Humm J. P., S. J. Benbow, (1999). "A Partially Saturated Flow Module In FracAffinity, Final Report", QuantiSci Report MC-6142A-4 Version 1.1.

22) Anderson, M.P. and Woessner, W.W (1992), Applied Groundwater Modelling, Academic Press.

23) Pan L. and P. Wierenga, (1995). "A transformed pressure head-based approach to solve Richards' equation for variably saturated soils", Water Resour. Res., 31(4), pp.925-931.

24) Pan L. and P. Wierenga, (1997). "Improving numerical modelling of two-dimensional water flow in variably saturated, heterogeneous porous media", Soil Sci. Soc. Am. J., 61(2), pp. 335-346.

25) Posiva (1999). Safety assessment of spent fuel disposal in Hästholmen, Kivetty, Olkiluoto and Romuvaara TILA-99. Posiva Report 99-07.

Glossary

This glossary is provided to define terms as they are used within GEOMASS. Therefore, the glossary includes basic terms such as borehole, shaft and excavation, in order to explain the different way that these features are treated in the system.

Anisotropic	Hydraulic conductivity can take different values in different directions. In the intact rock, the x-, y- and z-direction values can be set. In DDFs and SDFs, the in-plane and normal components are specified. Anisotropy is one way in which heterogeneity is represented in FracAffinity.
Aperture	The small thickness of the flow paths in a DDF or SDF.
Borehole	A thin cylinder in the rock with internal boundary nodes.
Borehole Section	A hydraulically isolated part of a borehole. The isolation may be through packers or because of separate boreholes.
Boundary Condition	The specification of the hydraulic conditions at a boundary.
Boundary Node	A node lying on the boundary of the model region (in the Hybrid Medium Network) or at the extremities of the intact rock Network or a DDF network. Every boundary node has a boundary condition associated with it. Boundary Nodes are commonly referred to as Internal Boundary Nodes or External Boundary Nodes in GEOMASS. Internal Boundary Nodes lie within the model domain and External Boundary Nodes lie at the edge of the model domain, but they are treated in the same manner by FracAffinity.
Cell	A volume of rock with uniform properties. The intact rock and DDF networks are created by first generating a rectilinear grid of cells.
Conductivity	*See Hydraulic conductivity.*
Connection	A flow path between two nodes in a FracAffinity network. Connections are generated in the intact rock, DDF and SDF networks and can be split as these are merged.
DDF	Deterministic Discrete Feature. A hydraulically significant surface feature with known position.

Deterministic	Fully specified by the user.
EarthVision	Geological modelling system that the FracAffinity interface is designed to take data from and return data to for visualisation.
Excavation	A cylindrical internal boundary to the model defined by a piece-wise linear schedule, and used to represent dynamic shafts, galleries or access ways in transient calculations.
FracAffinity	The flow solver part of GEOMASS.
Fractal	The $\log_{10}$ hydraulic conductivity in FracAffinity for intact rock grids and DDF grids may be generated by a fractal scaling law. This gives a realistic representation of heterogeneity.
Gallery	A static horizontal feature represented by a series of internal boundary nodes.
GEOMASS	The system consisting of EarthVision and FracAffinity for modelling and visualising geological structure and groundwater flow in an integrated fashion.
Grid	A collection of cells for the intact rock or a DDF.
Head	The piezometric head in the water, equal to the pressure head plus z-coordinate.
Heterogeneity	Variation in properties from location to location. FracAffinity represents heterogeneity on different scales through different approaches. Small-scale heterogeneity is treated by averaging (e.g. at a pore scale). Larger-scale heterogeneity can be represented by fractal properties of rock and DDFs or by including SDFs. The largest scale heterogeneity is represented by the stratigraphic zones imported from EarthVision.
Hybrid Medium Network	The network that arises from merging networks for intact rock, DDFs and SDFs.
Hybrid Rock Model	The final representation of the system under consideration, including the hybrid medium network.
Hydraulic Conductivity	The constant of proportionality in Darcy's law when pressures are in metres of water.

Intact Rock Model	The representation of the intact rock part of the system under consideration, including the intact rock network.
Isotropic	When the hydraulic conductivity is the same in any direction.
Network	A FracAffinity network is a collection of nodes, boundary nodes and connections together with hydraulic data. Networks for intact rock, DDFs and SDFs are merged to give the Hybrid Medium Network.
Node	A point in a FracAffinity network where the head solution is to be calculated. Nodes have a volume (of rock, DDF or SDF) associated with them.
Pathline	A path across a model region indicating the direction of flow.
Pressure	Water pressure measured in Pascals, relative to a zero reference atmospheric pressure.
Pressure Head	Water pressure measured in metres of water.
Recharge	An inflow boundary condition that provides a limiting flow rate which, combined with the saturation condition, determines the actual inflow.
Rock Model	A software representation of part of the system being solved.
Saturated	When the whole system is assumed to be full of water.
Saturation	The fraction of the local volume filled with water.
SDF	Stochastic Discrete Feature. A hydraulically significant planar feature with known density from geological mapping. The positions and orientations of SDFs are randomly generated.
Seepage	A special boundary condition that acts as a no-flow condition when the local rock is under-saturated and a zero pressure condition when the local rock is saturated.
Shaft	A static horizontal feature represented by a series of internal boundary nodes.
Specific Storage Coefficient	A measure of the compressibility of the rock (mainly) and water. A low specific storage coefficient corresponds to rapid communication of pressure changes.

Storativity	The ability of the rock to compress under pressure and so accommodate more water.
Stochastic	Randomly generated.
Stratigraphic Formation	A geological layer used to define a FracAffinity zone.
Suction	Negative pressure head.
Unsaturated	When the whole system need not be full of water. The fraction of water present (saturation) is calculated along with the heads for each node.
van Genuchten Model	Analytic (empirical) relationships for deriving suction from saturation and relative hydraulic conductivity from saturation.
Water Retention Model	Relationships for deriving suction from saturation and relative hydraulic conductivity from saturation. Includes the van Genuchten Model and piecewise-linear interpolations provided by the user.
Zone	A stratigraphic region with distinct, but not necessarily uniform, properties.

Appendix A: Algorithms used in the DYLAN Time-stepping Solver

Summary

DYLAN is a package of C++ classes for the solution of stiff systems of ordinary differential equations (ODEs) and algebraic equations. Although firmly based on the Byrne and Hindmarsh[13] approach, the precise algorithms used in DYLAN differ in detail. This appendix presents a report that sets out the time-stepping algorithms that are used.

A1 Introduction

DYLAN is a collection of C++ subroutines. There has previously been no single reference to the algorithms used in DYLAN. This appendix sets out those algorithms.

This appendix is written in a generic fashion, consistent with the way that DYLAN is used. The individual codes provide specific methods for evaluating the functions that characterise their problem.

A2 Equations to be Solved

DYLAN is targeted at solving stiff systems of algebraic and differential equations. These equations arise is the solution of partial differential equations when these are discretised using finite differences or finite elements. The form of equations that occurs in such situations is not the most general form, but has the advantage of being of index 0 or 1. This is important because higher index systems (the index is a measure of complexity) are not amenable to the same solution approaches and can be essentially insoluble.

The basic form of equations that DYLAN tackles is the initial value problem

$$\begin{aligned} m(x,y,t)\dot{x} &= f(x,y,t) \\ 0 &= g(x,y,t) \end{aligned} \tag{A2.1}$$

with initial conditions

$$\begin{aligned} x(t_0) &= x_0 \\ y(t_0) &= y_0 \end{aligned} \tag{A2.2}$$

which are assumed to satisfy

$$0 = g(x_0, y_0, t_0) \tag{A2.3}$$

The variables x and y are both vectors, for example representing nodal values in a finite-difference calculation. The calculation of consistent sets of initial conditions, that satisfy (A2.3), is part of the DYLAN package and is discussed later.

The requirement is to calculate x and y for a range of times $t_0<t<t_{max}$. The algorithms assume that the functions are continuous and sufficiently differentiable within the domain of interest. The x-variables are known as differential variables and the y-variables as algebraic variables. The matrix M is included to cover the situation where a finite-element mass matrix is present, but can be taken to be the identity matrix in finite difference calculations.

In much of the discussions presented here, the distinction between the x- and y-variables is not important. For convenience, we therefore write the joint set of variables as u and consider the system

$$M(u,t)\dot{u} = F(u,t) \tag{A2.4}$$

where some rows of M, corresponding to the algebraic variables, are zero.

A3 Solving the Equations

A3.1 Time Stepping

The basic approach used is time stepping. We aim to create a sequence of results at a monotonically increasing set of times. These times are determined as part of the algorithm. We denote these times as $t_0, t_1, t_2, \ldots$, the (approximate) solutions there as $u_0, u_1, u_2, \ldots$ and the timesteps as $h_1, h_2, \ldots$ where $t_n + h_{n+1} = t_{n+1}$.

DYLAN employs a variable-order, variable-timestep backward difference formulation following Byrne and Hindmarsh[13]. This uses a predictor-corrector approach to control accuracy. Thus, at each step, the algorithm must:

- Predict the solution at the end of the step.
- Correct the prediction.
- Determine the error and decide whether to accept the step.
- Determine the timestep for the next step (or for a retaken step if the error was unacceptable).
- Determine the order of method (see later for a discussion of the order) for a new step.

Additionally, results much be captured for reporting back to the user. This is often done at predetermined output times. DYLAN makes no assumption about this and simply makes available an interpolation for the results within a successful step. This interpolation is in fact a fundamental part of the algorithm, and is presented first.

A3.2 Interpolating Polynomials and the Nordsieck Array

The algorithm is based on the idea of interpolating polynomials whose order varies as the solution progresses. The order at any stage is denoted as k and is between 1 and 5 (except on the first step when it could be thought of as zero). The objective at each stage is to create an interpolating polynomial that has the calculated values at the latest times and produces consistent derivatives at the new time. Thus, in stepping to t_n, we look for a polynomial, π_n, of order k, such that

$$\pi_n(t_{n-i}) = u_{n-i} \quad \text{for} \quad 0 \le i \le k \tag{A3.1}$$

and

$$M(u_n,t_n)\dot{u}_n = F(u_n,t_n) \quad \text{with} \quad \dot{u}_n = \dot{\pi}_n(t_n) \tag{A3.2}$$

Notice that (A3.1) is sufficient to determine the polynomial once u_n is known, and (A3.2) is therefore the equation that will determine u_n.

Rather than store the coefficients of powers of t in the polynomials, the Nordsieck array is used [Nordsieck, 1962]. This is obtained by expanding the polynomial about the latest time point. We write

$$\pi_n(t_n + \mu h_n) = Z_n^{(0)} + \mu Z_n^{(1)} + \mu^2 Z_n^{(2)} + \ldots + \mu^k Z_n^{(k)} \tag{A3.3}$$

which implies that

$$Z_n^{(r)} = \frac{h_n^r u_n^{(r)}}{r!} \tag{A3.4}$$

The array with the Z vectors as its columns is called the Nordsieck array. This is clearly useful for interpolating to times in the last step, by setting μ between 0 and –1, but is also useful for extrapolating in the prediction stage. The first two columns of the Nordsieck array are the solution and a scaled derivative at the latest time.

A3.3 Predictor

At a general point in the solution, we have evaluated the solutions to time t_{n-1} and obtained the interpolating polynomial, π_{n-1}, or the equivalent Nordsieck array, Z_{n-1} and want to calculate the result at the next time, t_n.

This is done in two stages, first a prediction is made (of the interpolating polynomial, π_n rather than just of the solution, u_n). This is subsequently "corrected", to ensure than equations (A3.1) and (A3.2) are satisfied at t_n.

The predictor, which we denote by a circumflex, is simply the interpolating polynomial at the last step

$$\hat{\pi}_n = \pi_{n-1} \tag{A3.5}$$

This already satisfies the conditions of (A3.1) since, as discussed later, the order k used on any step is at most one more than that used on the previous step. In terms of the solution, this is just taking the previous interpolating polynomial and using it to extrapolate to the new time. It is possible that the predictor includes infeasible solution values (e.g. negative saturations). In such cases the time step does not proceed and a step of half the size is attempted.

A3.4 Corrector

To correct the polynomial, we are constrained to add a polynomial that is zero at the previous k times, so that the condition of (A3.1) remain satisfied. Since we are looking for a polynomial of order k, this determines the correction up to a factor. This is crucial to the algorithm – it means that the correction in the derivative is a known factor times the correction in the value. So, we write

$$\pi_n = \hat{\pi}_n + e_n p_n \tag{A3.6}$$

where p_n is the polynomial that is unity at t_n and zero at the previous k times, specifically

$$p_n = \prod_{r=1}^{k} \frac{(t - t_{n-r})}{(t_n - t_{n-r})} \tag{A3.7}$$

So

$$\begin{aligned} u_n &= \hat{u}_n + e_n \\ \dot{u}_n &= \hat{\dot{u}}_n + \theta_n e_n \end{aligned} \tag{A3.8}$$

where θ_n is a constant,

$$\theta_n = \dot{p}_n(t_n) = \sum_{r=1}^{k} \frac{1}{(t_n - t_{n-r})} \tag{A3.9}$$

We can now substitute (A3.8) in (A3.2) to obtain a non-linear equation that is to be solved for e_n,

$$M(\hat{u}_n + e_n, t_n)(\hat{\dot{u}}_n + \theta_n e_n) = F(\hat{u}_n + e_n, t_n) \tag{A3.10}$$

In practice, it is convenient to divide through by θ_n and write the equation as

$$\Phi(e_n) = M(\hat{u}_n + e_n, t_n)(\Delta_n \hat{\dot{u}}_n + e_n) - \Delta_n F(\hat{u}_n + e_n, t_n) = 0 \tag{A3.11}$$

where

$$\Delta_n = \frac{1}{\theta_n} \tag{A3.12}$$

A3.5 Relationship to a Fully Implicit Time Step

The discussion above is rather abstract and it is useful to relate it to a simpler scheme. If with use an order 1 method throughout, take all the variables to be differential and take M to be the identity matrix, then the approach reduces to solving

$$\dot{u} = F(u, t) \tag{A3.13}$$

At t_{n-1} the polynomial will be

$$\pi_{n-1} = u_{n-1} + (t - t_{n-1})\dot{u}_{n-1} \tag{A3.14}$$

with

$$\dot{u}_{n-1} = F(u_{n-1}, t_{n-1}) \tag{A3.15}$$

So,

$$\begin{aligned} \hat{u}_n &= u_{n-1} + h_n \dot{u}_{n-1} \\ \hat{\dot{u}}_n &= \dot{u}_{n-1} \end{aligned} \tag{A3.16}$$

then

$$\Delta_n = h_n \tag{A3.17}$$

and the equation to solve is

$$\Phi(e_n) = (h_n \hat{\dot{u}}_n + e_n) - h_n F(\hat{u}_n + e_n, t_n) = 0 \tag{A3.18}$$

We can rearrange this in more familiar terms as

$$(u_n - u_{n-1}) = h_n F(u_n, t_n) \tag{A3.19}$$

which is simply the standard fully implicit method.

A3.6 Newton Iteration

The discussion above is rather abstract and it is useful to relate it to a simpler scheme. If with use an order 1 method throughout, take all the variables to be differential and take M to be the identity matrix, then the approach reduces to solving

$$\dot{u} = F(u, t) \tag{A3.20}$$

At t_{n-1} the polynomial will be

$$\pi_{n-1} = u_{n-1} + (t - t_{n-1})\dot{u}_{n-1} \tag{A3.21}$$

with

$$\dot{u}_{n-1} = F(u_{n-1}, t_{n-1}) \tag{A3.22}$$

So,

$$\begin{aligned} \hat{u}_n &= u_{n-1} + h_n \dot{u}_{n-1} \\ \hat{\dot{u}}_n &= \dot{u}_{n-1} \end{aligned} \tag{A3.23}$$

then

$$\Delta_n = h_n \tag{A3.24}$$

and the equation to solve is

$$\Phi(e_n) = (h_n \hat{\dot{u}}_n + e_n) - h_n F(\hat{u}_n + e_n, t_n) = 0 \tag{A3.25}$$

We can rearrange this in more familiar terms as

$$(u_n - u_{n-1}) = h_n F(u_n, t_n) \tag{A3.26}$$

which is simply the standard fully implicit method.

A3.7 Determining the Error

The error estimate used is DYLAN is precisely that derived by Byrne and Hindmarsh [13].

Before giving the formula, we need to introduce some additional notation. Recall the polynomial p_n, which is used in the correction step and was defined in equation (A3.7). If this is written in Nordsieck array form then the columns of the array are single values which we denote as $l_0(k)$, $l_1(k)$ etc. with the k notation acting as a reminder that the polynomial is order k.

We compute an error vector $E_n(k)$ as

$$E_n(k) = \frac{-1}{l_1(k)}\left[1 + \prod_{r=2}^{r=k}\left(\frac{t_n - t_{n-r}}{t_{n-1} - t_{n-r}}\right)\right]^{-1} e_n \qquad \text{(A3.27)}$$

and then calculate the norm

$$D_k = \left\|E_n(k)\right\| \qquad \text{(A3.28)}$$

This is compared to an error tolerance, ε, and the step is accepted if $D_k<\varepsilon$. In any case, the recommended scaling of the time step is given by

$$\eta_k = \left(\frac{\varepsilon}{D_k}\right)^{\frac{1}{k+1}} \qquad \text{(A3.29)}$$

A3.8 Controlling the Order

In order to decide whether to change the order, k, of the algorithm on the next step, DYLAN calculates error estimates for what the error would have been if k was one higher or one lower, again using the same formulae as Byrne and Hindmarsh[13].

$$D_{k\pm1} = \left\|E_n(k \pm 1)\right\| \qquad \text{(A3.30)}$$

where

$$E_n(k-1) = \frac{-1}{l_1(k-1)}\left[\prod_{r=1}^{r=k}\left(\frac{t_n - t_{n-r}}{t_n - t_{n-1}}\right)\right]^{-1} Z_n^{(k)} \qquad \text{(A3.31)}$$

and

$$E_n(k+1) = \frac{-1/(k+2)}{l_1(k+1)}\left[1 + \prod_{r=2}^{r=k}\left(\frac{t_n - t_{n-r}}{t_{n-1} - t_{n-r}}\right)\right]^{-1}\left(\frac{t_n - t_{n-k-1}}{t_n - t_{n-1}}\right)(e_n - Q_n e_{n-1}) \qquad \text{(A3.32)}$$

where Q_n is a constant derived from the earlier time step sizes and is given in Byrne and Hindmarsh[13].

If the time step factor implied by these formula is larger than the factor for leaving the step unchanged then the next step will be undertaken with the revised order.

A3.9 Enhancements for Algebraic Equations

When algebraic equations are present, their relationships are defined through the function g of equation (A2.1). Within the approach defined so far, the algebraic variables are determined along with the differential variables and are subject to the iterative approach on each step. This is a valid approach but can be inefficient in circumstances where the particular problem being solved has special knowledge of how to determine y to satisfy $g(x,y,t)=0$, given x and t. DYLAN allows for an improvement to the algebraic variables to be calculated in this way. This can be used at various points in the calculation: prior to calculating consistent initial conditions (see the next Section); to the predictor; to the corrector after each iteration; and to the corrector after convergence. Which are used depends on the type of problem being tackled,

for example if the improvement is expensive then it should not be applied after every iteration.

A3.10 Initial Conditions

In order to find a consistent set of initial values, DYLAN will solve the g equations from (A2.1) for a given set of differential variables to find the corresponding y variables.
This simply uses the same Newton iterative approach as on the time steps, but for only the algebraic variables. Thus an iterative scheme of the form

$$y_0^{I+1} - y_0^{I} = \delta y_0^{I} = -\left(\frac{\partial g}{\partial y}\right)^{-1} g(x_0, y_0^{I}, t_0)\,. \qquad \text{(A3.33)}$$

is executed to convergence.

Appendix B: An Example Log File

```
====================================
    FracAffinity
    Version 3.2
    November 2006
====================================

Started: Wed 22 Nov 2006 at 13:57:45
Current Directory:

C:\GEOMASS\FY2006\ShobaTestCases\4FiveExcavations_NrEx\FiveExcavations_NrEx

====================
Reading Control File
====================

====================================
Reading Model Region information
====================================
flowsolver.fa:: TRANSIENT_UNSAT
flowsolver.fa:: START 0
flowsolver.fa:: END 6.32e+008
flowsolver.fa:: MONITOR_TIMES 2.6e+006 5.2e+006 7.9e+006 1.05e+007 1.31e+007
1.58e+007 1.84e+007 2.11e+007 2.37e+007 2.63e+007 2.89e+007 3.16e+007 6.32e+007
9.48e+007 1.26e+008 1.58e+008 1.9e+008 2.2e+008 2.5e+008 2.84e+008 3.16e+008 6.32e+008
flowsolver.fa:: FLOW_TIMES 2.6e+006 5.2e+006 7.9e+006 1.05e+007 1.31e+007 1.58e+007
1.84e+007 2.11e+007 2.37e+007 2.63e+007 2.89e+007 3.16e+007 3.16e+008
flowsolver.fa:: TOLERANCE 1e-008
flowsolver.fa:: TIMESTEP_TOLERANCE 1e-005
flowsolver.fa:: MAX_ITERATIONS 10000
flowsolver.fa:: MAX_ITERATIONS_PER_PICARD_STEP 5
flowsolver.fa:: MAX_PICARD_STEPS 1000
flowsolver.fa:: SAVE_BEST_EVERY 10
flowsolver.fa:: JACOBIAN_UPDATE_INTERVAL 10
flowsolver.fa:: PICARD_THEN_NEWTON 5
flowsolver.fa:: NON_LINEAR_RAMP 0.7 0.1 0.8
flowsolver.fa:: RANDOM_SEED -1

====================================
Reading Stratigraphic information
====================================
```

```
test.seq.fai::
test.seq.fai:: # Version 4
test.seq.fai::
test.seq.fai::
test.seq.fai:: Geometry

Reading geometry information in stratigraphy
test.seq.fai::      X   5100   5900
test.seq.fai::      Y   -69200   -67950
test.seq.fai::      Z   -1000   400
test.seq.fai:: END
test.seq.fai::
test.seq.fai:: Zone   "tsukiHWDZ"
test.seq.fai::      Operation   "Depositional"
test.seq.fai:: END
test.seq.fai::
test.seq.fai:: Zone   "Felsic"
test.seq.fai::      Operation   "Depositional"
test.seq.fai:: END
test.seq.fai::
test.seq.fai:: Zone   "BiotiteLWR"
test.seq.fai::      Operation   "Depositional"
test.seq.fai:: END
test.seq.fai::
test.seq.fai:: Zone   "BioLowK"
test.seq.fai::      Operation   "Depositional"
test.seq.fai:: END
test.seq.fai::
test.seq.fai:: Zone   "BiotiteUP"
test.seq.fai::      Operation   "Unconformity"
test.seq.fai:: END
test.seq.fai::
test.seq.fai:: Zone   "tsukiFWDZ"
test.seq.fai::      Operation   "Unconformity"
test.seq.fai:: END
test.seq.fai::
test.seq.fai:: Zone   "WeathAndCong"
test.seq.fai::      Operation   "Depositional"
test.seq.fai:: END
test.seq.fai::
test.seq.fai:: Zone   "Sediments"
test.seq.fai::      Operation   "Unconformity"
test.seq.fai:: END
```

```
test.seq.fai::
test.seq.fai::
test.seq.fai:: FaultBlock   "ALL"
test.seq.fai::      Fault   "tsuki80"
test.seq.fai::      FaultFile   tsuki80.2grd.fai
Reading Fault tsuki80 from tsuki80.2grd.fai
test.seq.fai::      AboveBlock   "HWtsuki"
test.seq.fai::      BelowBlock   "FWtsuki"
test.seq.fai:: END

test.seq.fai::
test.seq.fai:: FaultBlock   "HWtsuki"
test.seq.fai::      Sequence
test.seq.fai::          "tsukiHWDZ"   tsuki_upDZ.2grd.fai
Reading tsukiHWDZ from tsuki_upDZ.2grd.fai
test.seq.fai::          "Felsic"   fel_HW_v80.2grd.fai
Reading Felsic from fel_HW_v80.2grd.fai
test.seq.fai::          "BiotiteLWR"   biotite_LowK_base_v1.2grd.fai
Reading BiotiteLWR from biotite_LowK_base_v1.2grd.fai
test.seq.fai::          "BioLowK"   biotite_LowK_top_v1.2grd.fai
Reading BioLowK from biotite_LowK_top_v1.2grd.fai
test.seq.fai::          "BiotiteUP"   WeathBase.2grd.fai
Reading BiotiteUP from WeathBase.2grd.fai
test.seq.fai::          "WeathAndCong"   CongTop.2grd.fai
Reading WeathAndCong from CongTop.2grd.fai
test.seq.fai::          "Sediments"   topo.2grd.fai
Reading Sediments from topo.2grd.fai
test.seq.fai::      END
test.seq.fai:: END

test.seq.fai::
test.seq.fai:: FaultBlock   "FWtsuki"
test.seq.fai::      Sequence
test.seq.fai::          "Felsic"   fel_FW_v80.2grd.fai
Reading Felsic from fel_FW_v80.2grd.fai
test.seq.fai::          "BiotiteLWR"   biotite_LowK_base_v1FW.2grd.fai
Reading BiotiteLWR from biotite_LowK_base_v1FW.2grd.fai
test.seq.fai::          "BioLowK"   biotite_LowK_top_v1FW.2grd.fai
Reading BioLowK from biotite_LowK_top_v1FW.2grd.fai
test.seq.fai::          "BiotiteUP"   tsuki_lwrDZ.2grd.fai
Reading BiotiteUP from tsuki_lwrDZ.2grd.fai
test.seq.fai::          "tsukiFWDZ"   WeathBase.2grd.fai
Reading tsukiFWDZ from WeathBase.2grd.fai
```

test.seq.fai:: "WeathAndCong" CongTop.2grd.fai
Reading WeathAndCong from CongTop.2grd.fai
test.seq.fai:: "Sediments" topo.2grd.fai
Reading Sediments from topo.2grd.fai
test.seq.fai:: END
test.seq.fai:: END

test.seq.fai::
test.seq.fai::
Read fault block information in stratigraphy
Constructed fault block hierarchy in stratigraphy
Constructed formation information in stratigraphy
Constructed upper and lower surfaces in stratigraphy
Completed construction of stratigraphy

Formations in stratigraphy
7: Sediments UNCONFORMITY
6: WeathAndCong DEPOSITIONAL
5: tsukiFWDZ UNCONFORMITY
4: BiotiteUP UNCONFORMITY
3: BioLowK DEPOSITIONAL
2: BiotiteLWR DEPOSITIONAL
1: Felsic DEPOSITIONAL
0: tsukiHWDZ DEPOSITIONAL

Time So Far 2.343
Action Time (Model Region and Stratigraphy) 2.343
Read zone_cond.fa
Read conductivity.fa
Read zone_por.fa
Read zone_ssc.fa
grid_parm.fa:: X_SIZE 100
grid_parm.fa:: Y_SIZE 100
grid_parm.fa:: Z_SIZE 100
grid_parm.fa:: MIN_NUM 1
grid_parm.fa:: RESOLUTION 0.001
grid_parm.fa:: GRID_REFINEMENT
grid_parm.fa:: LOCAL_REFINEMENT
grid_parm.fa:: NEAR_EXCAVATION excav 50
grid_parm.fa:: LOCAL_X 10
grid_parm.fa:: LOCAL_Y 10
grid_parm.fa:: LOCAL_Z 10
Read Local Refinement: OK

```
Read grid parameters in grid_parm.fa
Maximum grid size: 8 x 13 x 14 cells = 1456
Assigning grid cells
Assigning grid cell neighbours
Assigning grid cell vertices
Spliiting cells for local refinement
Generated regular rectilinear grid: 1456 cells 1890 vertices
With local refinement, number of nodes is: 23367
==========================================
Generating an intact rock model from grid
==========================================
------------------------------------------------------------------------
Populator Found 3 properties in Property Manager
Property 0 is conductivity
Property 1 is porosity
Property 2 is storage coefficient
------------------------------------------------------------------------
Populating Fault Zone 0 from Formation/Material tsukiHWDZ
   For property conductivity
   For property porosity
   For property storage coefficient
------------------------------------------------------------------------
Populating Fault Zone 1 from Formation/Material Felsic
   For property conductivity
   For property porosity
   For property storage coefficient
------------------------------------------------------------------------
Populating Fault Zone 2 from Formation/Material BiotiteLWR
   For property conductivity
   For property porosity
   For property storage coefficient
------------------------------------------------------------------------
Populating Fault Zone 3 from Formation/Material BioLowK
   For property conductivity
   For property porosity
   For property storage coefficient
------------------------------------------------------------------------
Populating Fault Zone 4 from Formation/Material BiotiteUP
   For property conductivity
   For property porosity
   For property storage coefficient
------------------------------------------------------------------------
Populating Fault Zone 5 from Formation/Material tsukiFWDZ
```

```
   For property conductivity
   For property porosity
   For property storage coefficient
---------------------------------------------------------------------------
Populating Fault Zone 6 from Formation/Material WeathAndCong
   For property conductivity
   For property porosity
   For property storage coefficient
---------------------------------------------------------------------------
Populating Fault Zone 7 from Formation/Material Sediments
   For property conductivity
   For property porosity
   For property storage coefficient

---------------------------------------------------------------------------
Populated Network with Cell data components
---------------------------------------------------------------------------
Intact Rock Network has been populated with data values
    Read Van Genuchten WRM
    Read Van Genuchten WRM
    Read Van Genuchten WRM
    Read Van Genuchten WRM
    Read Van Genuchten WRM
    Read Van Genuchten WRM
    Read Van Genuchten WRM
    Read Van Genuchten WRM
    Read Van Genuchten WRM
Time So Far 3.765
Action Time (Intact Rock Model) 1.422
Time So Far 3.765
Action Time (Shafts and Excavations) 0
===================================
Preparing for Deterministic features
===================================
Read grid parameters in ddf_grid_parm.fa
===================================
Gridding DDFs
===================================

Gridding feature : Tsukiyoshi
Maximum grid size: 8 x 10 cells = 80
Assigning grid cells
Assigning grid cell neighbours
Assigning grid cell vertices
```

```
Spliiting cells for local refinement
Generated surface grid: 80 cells 99 vertices
Number of grids = 1
=====================================
Get deterministic features properties
=====================================
Read ddf_cond.fa
Read ddf_conductivity.fa
Read ddf_por.fa
Read ddf_ssc.fa
=====================================
Populate deterministic features
=====================================
---------------------------------------------------------------------------
Populator Found 3 properties in Property Manager
Property 0 is conductivity
Property 1 is porosity
Property 2 is storage coefficient
---------------------------------------------------------------------------
Populating Fault Zone 0 from Formation/Material Tsukiyoshi
   For property conductivity
   For property porosity
   For property storage coefficient

---------------------------------------------------------------------------
Populated Grid with Cell data components
---------------------------------------------------------------------------
=====================================
Populated feature : Tsukiyoshi
=====================================
=====================================
Construct and Combine DDF Rock Grids
=====================================
DDF Model constructed
    Read Van Genuchten WRM
Time So Far 3.843
Action Time (DDF Model) 0.078
Time So Far 3.843
Action Time (SDF Model) 0
=====================================
Create Hybrid DDF Rock Grid
=====================================
=====================================
```

Preparing the Excavation Models
==========

****
Summary of Excavation Sections
Excavation excav(VerticalShaft) of radius 3.25
  Starts at time 0 from Point at (5478,-68528,237.876) (SURFACE)
  Ends at time 2.5e+006 at Point at (5478,-68528,42)
Excavation excav(Shaft2Section1) of radius 3.25
  Starts at time 0 from Point at (5478,-68428,252.864) (SURFACE)
  Ends at time 500000 at Point at (5600,-68100,100)
Excavation excav(Shaft2Section2) of radius 3.25
  Starts at time 500000 from Point at (5600,-68100,100) (LINKED TO excav(Shaft2Section1))
  Ends at time 1e+006 at Point at (5478,-68428,100)
Excavation excav(Shaft2Section3) of radius 3.25
  Starts at time 1e+006 from Point at (5478,-68428,100) (LINKED TO excav(Shaft2Section2))
  Ends at time 2.5e+006 at Point at (5478,-68428,42)
Excavation excav(Gallery) of radius 3.25
  Starts at time 2.51e+006 from Point at (5478,-68528,100) (LINKED TO excav(VerticalShaft))
  Ends at time 2.52e+006 at Point at (5478,-68428,100)
****

--> Remove Leg (793 to 792)
--> Remove Leg (793 to 763)
Time So Far 3.968
Action Time (Hybrid Model) 0.125
Hybrid Model has 23449 nodes
==========
Fit Rock Grid to boundaries and discard unconnected nodes
==========
Total number of nodes was 23449 of which 74 were outside the model region
Total number of boundary nodes was 1650 of which 23 were inside the model region
Total number of connections checked was 73247 of which 270 crossed the boundary
Total number of boundary nodes now 1920 of which 85 were not on the boundary of the model region
Total number of unconnected nodes was 74
Final number of nodes was 23375
Final number of boundary nodes was 1835
Final number of connections was 74705
Final number of nodes now outside region was 0

Time So Far 4.265
Action Time (Boundaries) 0.297
==========
Output hybrid Rock Grid for visualisation
==========
Time So Far 4.875
Action Time (Properties File) 0.61
==========
Pre-process Rock Grid for flow solver
==========
Time So Far 5.047
Action Time (Prepare for Solve) 0.172
==========
Solving the flow
==========
Finding nodes for all output positions requested

Applying boundary conditions to rock model
Warning: Some boundaries defined with no boundary condition
Assuming No Flux condition applies

Done

GRIDDING WARNING for Excavation excav(VerticalShaft)
No internal nodes between lengths 4.90744 and 14.7319
(Additional Lines Deleted)
GRIDDING WARNING for Excavation excav(Gallery)
No internal nodes between lengths 54.675 and 92.075
Solving for unsaturated steady flow initial conditions

Steady State Solver convergence target: 6.57468e-009

Solving with non-linear fraction set to: 0.7
==========
Taking a Picard iteration step
Linear solver failed to converge
Steady Unsaturated Iteration 1 Residual 0.142595

Solving with non-linear fraction set to: 0.8
==========
Taking a Picard iteration step
Linear solver failed to converge
Steady Unsaturated Iteration 2 Residual 0.0184677

Solving with non-linear fraction set to: 0.8

==========

Taking a Picard iteration step

Linear solver failed to converge

Steady Unsaturated Iteration 3 Residual 0.00526421

Solving with non-linear fraction set to: 0.8

(Additional Lines Deleted)

==========

Taking a Newton iteration step

Steady Unsaturated Iteration 29 Residual 7.98072e-009

Solving with non-linear fraction set to: 0.8

==========

Taking a Newton iteration step

Steady Unsaturated Iteration 30 Residual 1.45044e-009

Saving best result so far to head_best.faout

Num Iterations: 30 Num Divergences: 3

Done

Solving for transient flow

==========

Unsaturated Water Balance Report (Steady State)

==========

(Non-linear fraction is 0.8)

Total water in the model is 1.84749e+008

Boundary	In	Out	Net out
ReuseHead: west	0.0773763	0.0191823	-0.058194
ReuseHead: south	0.0293878	0.0147474	-0.0146404
Flux: BOTTOM	0	0	0
Recharge: TOP	0.234027	0.198091	-0.0359359
ReuseHead: north	0.0893609	0.0078122	-0.0815487
ReuseHead: east	0.000496332	0.190815	0.190319
TOTAL	0.430649	0.430649	9.43842e-008

- Unsaturated

==========

Unsaturated Water Balance Report (2.6e+006)

==========

Total water in the model is 1.84736e+008

Boundary	In	Out	Net out
ReuseHead: west	0.0776139	0.0191791	-0.0584348
ReuseHead: south	0.0293893	0.0147461	-0.0146432
Flux: BOTTOM	0	0	0
Recharge: TOP	0.220162	0.187733	-0.0324286
ReuseHead: north	0.0892116	0.00783681	-0.0813748
ReuseHead: east	0.000504385	0.189283	0.188779
Seepage: excav(VerticalShaft)	0	0.0600799	0.0600799
Seepage: excav(Gallery)	0	0.0202219	0.0202219
Seepage: excav(Shaft2Section3)	0	0.00105905	0.00105905
Seepage: excav(Shaft2Section2)	0	0.0191152	0.0191152
Seepage: excav(Shaft2Section1)	0	0.314083	0.314083
Resaturation	0.00259185	0.000323728	-0.00226812
Storativity	0.414216	2.7229e-005	-0.414189
TOTAL	0.430649	0.430649	9.43842e-008

(Additional Water Balance Reports Deleted)

==========

Unsaturated Water Balance Report (6.32e+008)

==========

Total water in the model is 1.83996e+008

Boundary	In	Out	Net out
ReuseHead: west	0.0834311	0.0172837	-0.0661474
ReuseHead: south	0.0294665	0.0147023	-0.0147642
Flux: BOTTOM	0	0	0
Recharge: TOP	0.230111	0.129389	-0.100722
ReuseHead: north	0.123429	7.16171e-005	-0.123358
ReuseHead: east	0.00244052	0.158189	0.155749
Seepage: excav(VerticalShaft)	0	0.0342281	0.0342281
Seepage: excav(Gallery)	0	0.0119297	0.0119297
Seepage: excav(Shaft2Section3)	0	0.000726561	0.000726561

Seepage: excav(Shaft2Section2)	0	0.0137604	0.0137604
Seepage: excav(Shaft2Section1)	0	0.0892643	0.0892643
Resaturation	8.39241e-005	6.57053e-005	-1.82188e-005
Storativity	0.00070842	5.99231e-005	-0.000648497
---	---	---	---
TOTAL	0.430649	0.430649	9.43842e-008

Writing out Path Lines
Time So Far 14563.9
Action Time (Solve) 14558.8
Completed: Wed 22 Nov 2006 at 18:00:29

Appendix C: Using FAExport with EarthVision

The GEOMASS EarthVision-to-FracAffinity export program is accessible through the GEOMASS menu of EarthVision, under the FA Export option. Upon selection of this option the dialogue box shown in Fig. C.1 is displayed. The user should then supply information into the text boxes detailing the EarthVision directory, the sequence file name and the FracAffinity input directory. The user is required to select a sequence file name, however, the other two text boxes may be left blank. If the directory boxes are left blank, the user will be requested to enter that information during the running of the export program.

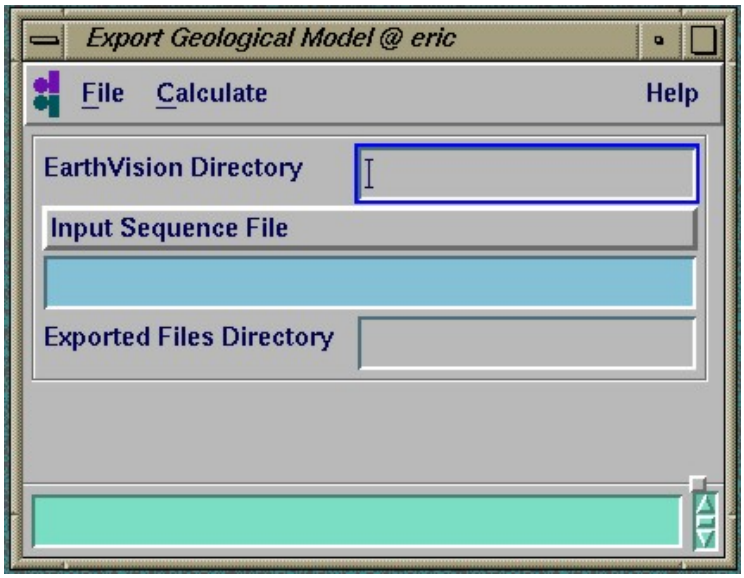


Fig. C.1 The EarthVision Graphical User Interface can be used to run the ev2fai programme by selecting GEOMASS>FracAffinity>Export-to-FracAffinity option from the EarthVision pull-down menu.

The following EarthVision files are written to the EarthVision export directory:

- Sequence file (*.seq), which describes the overall structure;
- Grid files (*.2grd), which describe a surface of fault.
- Polygon files (*.ply), which describe a polygon region.

Sequence Files

The sequence file is an ASCII file, and is processed by a new executable, EV2FAI. A shortened version of the sequence file is written for input to the FracAffinity Interface. The FAI sequence file is fname.seq.fai given the EV sequence file fname.seq.

The sections below indicate the lines where there is any processing. All other blocks in the EV sequence file are skipped, as described in the final subsection below.

Header Lines

The header lines (starting with #) are copied verbatim, but are not processed. These include the version number of EV used. Any blank lines are ignored.

Geometry Block

The geometry block X, Y and Z range information is required. Additional information on these lines or in the block is ignored. Thus, the processing looks for lines starting X, Y or Z and takes the following two numbers as minimum and maximum. The output sequence file therefore reads

```
Geometry {
X min-x max-x
Y min-y max-y
Z min-z max-z
}
```

Zone Blocks

The zone blocks contain various pieces of information that FracAffinity does not require. The only information copied over is the zone name and the Operation line. Thus an output sequence file will contain

```
Zone “zone-name” {
        Operation "Depositional"/"Unconformity"
}
```

The other information in the block (e.g. Horizon) is skipped.

FaultBlock Blocks

The FaultBlock inputs are of two types – Fault and Sequence. For the purposes of processing, it is sufficient to copy the information in the following lines: Fault, FaultFile, BoundingPolygon, AboveBlock, BelowBlock. In addition, the Sequence block is processed. It is the fault file and sequence block that contains the grids that need to be exported.

Thus

```
"name" pathname.2grd
```

will become

```
"name" fname.2grd.fai
```

with the appropriate call to ev_export and file renaming occurring.

Other blocks

Other blocks (e.g. History, PropertySummary) are skipped. This has been achieved by processing the input until a close bracket is discovered matching the open bracket that follows the header.

In order to make it easier to create a FracAffinity Interface sequence file by hand, the following applies.

All lines starting with # are treated as comments.

- Blank lines are ignored.
- Open brackets (‘{’) are ignored.
- Close brackets (‘}’) are treated as ‘END’ keywords.
- Keyword names are case-insensitive.

With this in mind, the FAI sequence file input syntax is as follows.

```
GEOMETRY
    X   x-min   x-max
    Y   y-min   y-max
    Z   z-min   z-max
END
ZONE   “zonename”
    OPERATION   "Depositional"/"Unconformity"
END

FAULTBLOCK   “blockname”
    FAULT “name”
    FAULTFILE fname.2grd.fai
    BOUNDINGPOLYGON fname.ply
    ABOVEBLOCK “blockname”
    BELOWBLOCK “blockname”
    SEQUENCE
                “name” fname.2grd.fai
                … repeated as necessary
        END
END
```

Grid Files

EarthVision grid files are binary, and must be processed using ev_export to produce an ASCII file. Given an input filename of fname.dat the result is named fname.2grd.fai. Note, that this implies a uniqueness in the grid-file names beyond what EarthVision might require. If there is already a file with this name then fname_n.2grd.fai is used, where n is 2, 3, 4 as required to give a unique name. At the start of processing, the fai-dirctory will have all *.2grd.fai files removed.

The EarthVision export program converts the EarthVision grid files referred to within a sequence file into an ASCII format which is readable by FracAffinity. The format of an exported file is as follows:

```
# Type: scattered data
# Version: 4
# Description: Exported from grid tsu.2grd
# Format: free
# Field: 1 x
```

Field: 2 y

Field: 3 z

Field: 4 column

Field: 5 row

Projection: Local Rectangular

Units: unknown

End:

Information from grid:

Grid_size: 100 x 100

Grid_X_range: 111.197000 to 12347.331000

Grid_Y_range: -73103.133000 to -60928.316000

Scattered_data: tsukiyoshi_v033.dat

Z_field: z

History: Calculated by ev_2grid

_History: Multiple data point gridding level: 4

_History: Extrapolated. (geomass, 23 Aug 1999)

Xi Yi Zi Col Row

……

Xn Yn Zn Col Row

The output file contains a header consisting of a number of lines beginning with a #. This header is generated by EarthVision's ev_export program and hence the details of this header are out of the control of the GEOMASS Development Team. Some aspects of the header provide useful information that may be used by FracAffinity (e.g. # Grid_size: # Grid_X_range: and # Grid_Y_range:). Other records such as the # History: are ignored. The grid data itself includes all records without the # in the first field. The data is listed from bottom left to top right (i.e. xmin, ymin to xmax, ymax). EarthVision identifies "NULL" nodes by the values 1E+20 or 1E-20.

Polygon Files

EarthVision polygon files are ASCII files. They are copied without modification for input into the FracAffinity Interface.

国際単位系（SI）

表１．SI 基本単位

基本量	SI 基本単位	
	名称	記号
長さ	メートル	m
質量	キログラム	kg
時間	秒	s
電流	アンペア	A
熱力学温度	ケルビン	K
物質量	モル	mol
光度	カンデラ	cd

表２．基本単位を用いて表されるSI組立単位の例

組立量	SI 基本単位	
	名称	記号
面積	平方メートル	m^2
体積	立法メートル	m^3
速さ，速度	メートル毎秒	m/s
加速度	メートル毎秒毎秒	m/s^2
波数	毎メートル	m^{-1}
密度，質量密度	キログラム毎立方メートル	kg/m^3
面積密度	キログラム毎平方メートル	kg/m^2
比体積	立方メートル毎キログラム	m^3/kg
電流密度	アンペア毎平方メートル	A/m^2
磁界の強さ	アンペア毎メートル	A/m
量濃度[(a)]，濃度	モル毎立方メートル	mol/m^3
質量濃度	キログラム毎立法メートル	kg/m^3
輝度	カンデラ毎平方メートル	cd/m^2
屈折率[(b)]	（数字の） 1	1
比透磁率[(b)]	（数字の） 1	1

(a) 量濃度（amount concentration）は臨床化学の分野では物質濃度（substance concentration）ともよばれる。
(b) これらは無次元量あるいは次元１をもつ量であるが、そのことを表す単位記号である数字の１は通常は表記しない。

表３．固有の名称と記号で表されるSI組立単位

組立量	SI 組立単位			
	名称	記号	他のSI単位による表し方	SI基本単位による表し方
平面角	ラジアン[(b)]	rad	1[(b)]	m/m
立体角	ステラジアン[(b)]	sr[(c)]	1[(b)]	m^2/m^2
周波数	ヘルツ[(d)]	Hz		s^{-1}
力	ニュートン	N		m kg s^{-2}
圧力，応力	パスカル	Pa	N/m^2	m^{-1} kg s^{-2}
エネルギー，仕事，熱量	ジュール	J	N m	m^2 kg s^{-2}
仕事率，工率，放射束	ワット	W	J/s	m^2 kg s^{-3}
電荷，電気量	クーロン	C		s A
電位差（電圧），起電力	ボルト	V	W/A	m^2 kg s^{-3} A^{-1}
静電容量	ファラド	F	C/V	m^{-2} kg^{-1} s^4 A^2
電気抵抗	オーム	Ω	V/A	m^2 kg s^{-3} A^{-2}
コンダクタンス	ジーメンス	S	A/V	m^{-2} kg^{-1} s^3 A^2
磁束	ウエーバ	Wb	Vs	m^2 kg s^{-2} A^{-1}
磁束密度	テスラ	T	Wb/m^2	kg s^{-2} A^{-1}
インダクタンス	ヘンリー	H	Wb/A	m^2 kg s^{-2} A^{-2}
セルシウス温度	セルシウス度[(e)]	℃		K
光束	ルーメン	lm	cd sr[(c)]	cd
照度	ルクス	lx	lm/m^2	m^{-2} cd
放射性核種の放射能[(f)]	ベクレル[(d)]	Bq		s^{-1}
吸収線量，比エネルギー分与，カーマ	グレイ	Gy	J/kg	m^2 s^{-2}
線量当量，周辺線量当量，方向性線量当量，個人線量当量	シーベルト[(g)]	Sv	J/kg	m^2 s^{-2}
酸素活性	カタール	kat		s^{-1} mol

(a)SI接頭語は固有の名称と記号を持つ組立単位と組み合わせても使用できる。しかし接頭語を付した単位はもはやコヒーレントではない。
(b)ラジアンとステラジアンは数字の１に対する単位の特別な名称で、量についての情報をつたえるために使われる。実際には、使用する時には記号rad及びsrが用いられるが、習慣として組立単位としての記号である数字の１は明示されない。
(c)測光学ではステラジアンという名称と記号srを単位の表し方の中に、そのまま維持している。
(d)ヘルツは周期現象についてのみ、ベクレルは放射性核種の統計的過程についてのみ使用される。
(e)セルシウス度はケルビンの特別な名称で、セルシウス温度を表すために使用される。セルシウス度とケルビンの単位の大きさは同一である。したがって、温度差や温度間隔を表す数値はどちらの単位で表しても同じである。
(f)放射性核種の放射能（activity referred to a radionuclide）は、しばしば誤った用語で"radioactivity"と記される。
(g)単位シーベルト（PV,2002,70,205）についてはCIPM勧告2（CI-2002）を参照。

表４．単位の中に固有の名称と記号を含むSI組立単位の例

組立量	SI 組立単位		
	名称	記号	SI 基本単位による表し方
粘度	パスカル秒	Pa s	m^{-1} kg s^{-1}
力のモーメント	ニュートンメートル	N m	m^2 kg s^{-2}
表面張力	ニュートン毎メートル	N/m	kg s^{-2}
角速度	ラジアン毎秒	rad/s	m m^{-1} s^{-1}=s^{-1}
角加速度	ラジアン毎秒毎秒	rad/s^2	m m^{-1} s^{-2}=s^{-2}
熱流密度，放射照度	ワット毎平方メートル	W/m^2	kg s^{-3}
熱容量，エントロピー	ジュール毎ケルビン	J/K	m^2 kg s^{-2} K^{-1}
比熱容量，比エントロピー	ジュール毎キログラム毎ケルビン	J/(kg K)	m^2 s^{-2} K^{-1}
比エネルギー	ジュール毎キログラム	J/kg	m^2 s^{-2}
熱伝導率	ワット毎メートル毎ケルビン	W/(m K)	m kg s^{-3} K^{-1}
体積エネルギー	ジュール毎立方メートル	J/m^3	m^{-1} kg s^{-2}
電界の強さ	ボルト毎メートル	V/m	m kg s^{-3} A^{-1}
電荷密度	クーロン毎立方メートル	C/m^3	m^{-3} sA
表面電荷	クーロン毎平方メートル	C/m^2	m^{-2} sA
電束密度，電気変位	クーロン毎平方メートル	C/m^2	m^{-2} sA
誘電率	ファラド毎メートル	F/m	m^{-3} kg^{-1} s^4 A^2
透磁率	ヘンリー毎メートル	H/m	m kg s^{-2} A^{-2}
モルエネルギー	ジュール毎モル	J/mol	m^2 kg s^{-2} mol^{-1}
モルエントロピー，モル熱容量	ジュール毎モル毎ケルビン	J/(mol K)	m^2 kg s^{-2} K^{-1} mol^{-1}
照射線量（X線及びγ線）	クーロン毎キログラム	C/kg	kg^{-1} sA
吸収線量率	グレイ毎秒	Gy/s	m^2 s^{-3}
放射強度	ワット毎ステラジアン	W/sr	m^4 m^{-2} kg s^{-3}=m^2 kg s^{-3}
放射輝度	ワット毎平方メートル毎ステラジアン	$W/(m^2 sr)$	m^2 m^{-2} kg s^{-3}=kg s^{-3}
酵素活性濃度	カタール毎立方メートル	kat/m^3	m^{-3} s^{-1} mol

表５．SI 接頭語

乗数	接頭語	記号	乗数	接頭語	記号
10^{24}	ヨタ	Y	10^{-1}	デシ	d
10^{21}	ゼタ	Z	10^{-2}	センチ	c
10^{18}	エクサ	E	10^{-3}	ミリ	m
10^{15}	ペタ	P	10^{-6}	マイクロ	μ
10^{12}	テラ	T	10^{-9}	ナノ	n
10^{9}	ギガ	G	10^{-12}	ピコ	p
10^{6}	メガ	M	10^{-15}	フェムト	f
10^{3}	キロ	k	10^{-18}	アト	a
10^{2}	ヘクト	h	10^{-21}	ゼプト	z
10^{1}	デカ	da	10^{-24}	ヨクト	y

表６．SIに属さないが、SIと併用される単位

名称	記号	SI 単位による値
分	min	1 min=60s
時	h	1h =60 min=3600 s
日	d	1 d=24 h=86 400 s
度	°	1°=(π/180) rad
分	’	1’=(1/60)°=(π/10800) rad
秒	”	1”=(1/60)’=(π/648000) rad
ヘクタール	ha	$1ha=1hm^2=10^4m^2$
リットル	L，l	$1L=11=1dm^3=10^3cm^3=10^{-3}m^3$
トン	t	$1t=10^3$ kg

表７．SIに属さないが、SIと併用される単位で、SI単位で表される数値が実験的に得られるもの

名称	記号	SI 単位で表される数値
電子ボルト	eV	$1eV=1.602\ 176\ 53(14)\times10^{-19}J$
ダルトン	Da	$1Da=1.660\ 538\ 86(28)\times10^{-27}kg$
統一原子質量単位	u	1u=1 Da
天文単位	ua	$1ua=1.495\ 978\ 706\ 91(6)\times10^{11}m$

表８．SIに属さないが、SIと併用されるその他の単位

名称	記号	SI 単位で表される数値
バール	bar	1 bar=0.1MPa=100kPa=10^5Pa
水銀柱ミリメートル	mmHg	1mmHg=133.322Pa
オングストローム	Å	1 Å=0.1nm=100pm=10^{-10}m
海里	M	1 M=1852m
バーン	b	$1\ b=100fm^2=(10^{-12}cm)2=10^{-28}m^2$
ノット	kn	1 kn=(1852/3600)m/s
ネーパ	Np	SI単位との数値的な関係は、対数量の定義に依存。
ベル	B	
デジベル	dB	

表９．固有の名称をもつCGS組立単位

名称	記号	SI 単位で表される数値
エルグ	erg	$1\ erg=10^{-7}$ J
ダイン	dyn	$1\ dyn=10^{-5}$N
ポアズ	P	$1\ P=1\ dyn\ s\ cm^{-2}=0.1Pa\ s$
ストークス	St	$1\ St=1cm^2\ s^{-1}=10^{-4}m^2\ s^{-1}$
スチルブ	sb	$1\ sb=1cd\ cm^{-2}=10^4cd\ m^{-2}$
フォト	ph	$1\ ph=1cd\ sr\ cm^{-2}\ 10^4lx$
ガル	Gal	$1\ Gal=1cm\ s^{-2}=10^{-2}ms^{-2}$
マクスウェル	Mx	$1\ Mx = 1G\ cm^2=10^{-8}Wb$
ガウス	G	$1\ G=1Mx\ cm^{-2}=10^{-4}T$
エルステッド[(c)]	Oe	1 Oe ≙ $(10^3/4\pi)$A m^{-1}

(c) ３元系のCGS単位系とSIでは直接比較できないため、等号「 ≙ 」は対応関係を示すものである。

表10．SIに属さないその他の単位の例

名称	記号	SI 単位で表される数値
キュリー	Ci	$1\ Ci=3.7\times10^{10}Bq$
レントゲン	R	$1\ R = 2.58\times10^{-4}C/kg$
ラド	rad	1 rad=1cGy=10^{-2}Gy
レム	rem	1 rem=1 cSv=10^{-2}Sv
ガンマ	γ	1γ=1 nT=10-9T
フェルミ		1フェルミ=1 fm=10-15m
メートル系カラット		1メートル系カラット = 200 mg = 2×10-4kg
トル	Torr	1 Torr = (101 325/760) Pa
標準大気圧	atm	1 atm = 101 325 Pa
カロリー	cal	1cal=4.1858J（「15℃」カロリー），4.1868J（「IT」カロリー）4.184J（「熱化学」カロリー）
ミクロン	μ	1 μ =1μm=10^{-6}m

（第8版，2006年改訂）