dGB-GDI
Concepts & Theory

Geoscientific software system for
Geology Driven Integration

Authors:

Dr. ir. Paul F.M. de Groot
Drs. Albertus (Bert) H. Bril

dGB; de Groot – Bril Earth Sciences B.V.
Boulevard 1945 nr. 24
7511 AE Enschede
The Netherlands
Tel. +31 53 4315155
Fax. +31 53 4315104
Email: info@dgb.nl
Website: http://www.dgb.nl

dGB
de Groot - Bril Earth Sciences

April, 2000

report number

Table of contents

1 General

In this document concepts and mathematical foundations are given on which some important dGB-GDI tools and functions are based. The document is organised in a number of completely independent chapters, each with its own reference list.

1.1 Acknowledgement

The Chapter on Monte Carlo statistics was published in Geophysics (de Groot et. al., 1996). The mathematical formulation originates from Dr. Frans Floris of NITG-TNO. The neural networks Chapter is taken from a PhD thesis (de Groot, 1995). The mathematical formulation is from Prof. dr. ir. Jacob Fokkema. The Chapter on fluid replacement is a compilation from various sources in the open literature.

2 Geology-Driven-Integration

2.1 Integration framework

In the geo-sciences interpretations are based on a combination of data and knowledge. Data is gathered in different dimensions with widely varying scales and accuracies. For a quantification all available information must be combined (Fig. 2.1).

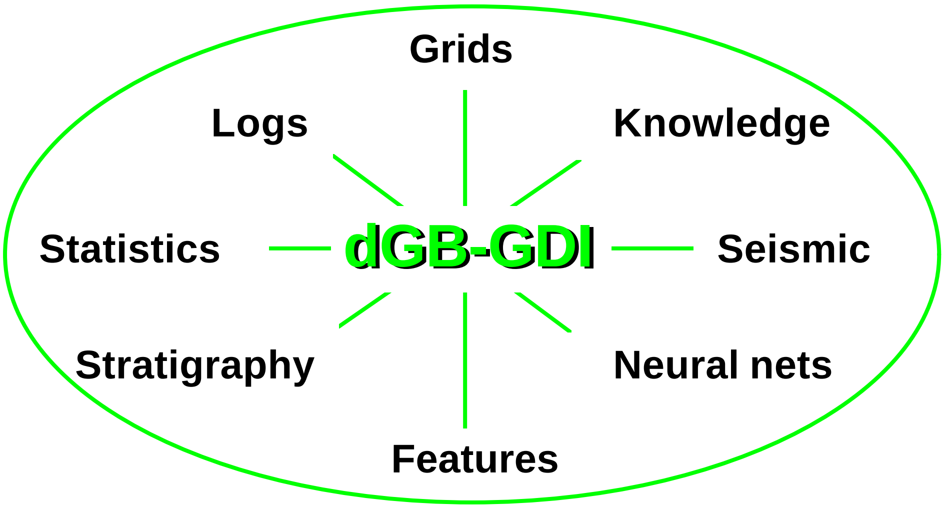


Figure 2.1: *dGB-GDI has been designed to quantify geo-scientific interpretations. Tools are available for integrating and manipulating various types of information.*

In dGB-GDI data and knowledge are combined via the **Integration framework**. This is a generic description of the subsurface in terms of geological objects with attached physical properties (non-numeric features such as seal, waste, or reservoir rock) and quantities (numeric features such as sonic, density, porosity, permeability etc.). The geological objects (or framework units) are the building blocks for describing different geological models (in this case wells). The models are called realisations of the framework (Figure 2.2).

The objects in a framework are ordered in a tree corresponding to a hierarchical ordering system (Figure 2.3). Each object is given a full name and code. Both are user-defined. The codes (UnitIDs) are used for identifying and manipulating the data at the natural (geological) scale levels. Any hierarchical system used in sedimentary geology can be projected into dGB-GDI and there be combined with any property and quantity of interest. There are no restrictions in the software with respect to the scale levels, nor to the properties and quantities. Both are completely user-defined.The integration framework is dGB-GDI's solution to the problem of integration of geo-scientific information. A stratigraphic model, in which geological objects have been defined in a hierarchical order, forms the basis. Realisations of the integration framework (wells) can be manipulated at the natural (stratigraphic) scale levels defined in the framework.

The concept is best explained by an example. Suppose we wish to study a geological setting with intercalating sands and shales. These rock-units can be grouped into a larger unit, say a member (which can also be grouped with other members into a formation and so on. For the sake of this argument we will restrict ourselves to the member level). Let us also assume that we are interested in studying two petrophysical quantities, e.g. sonic and density.

In a generic sense we are dealing with three objects in a hierarchical ordering system: member, sand and shale. If we attach the quantity thickness to all objects and the quantities sonic and density to the objects sand and shale, we have defined an integration framework. With this generic description we can now describe any sequence of sands and shales with associated sonic and density log responses in any detail required (Fig. 2.2).

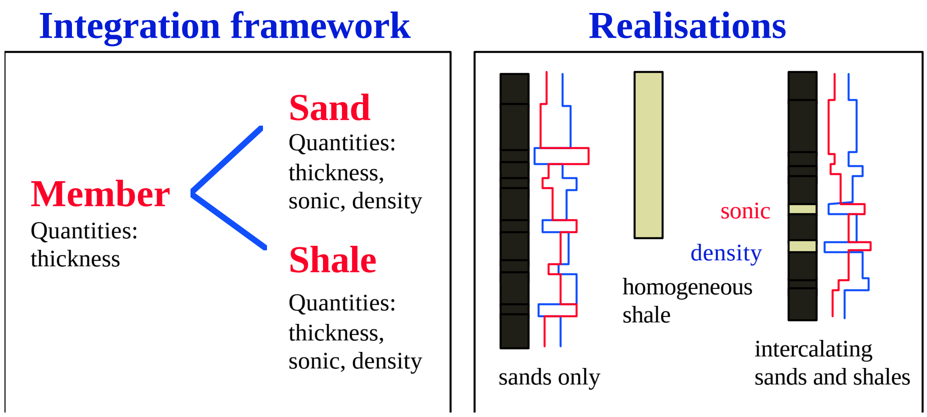
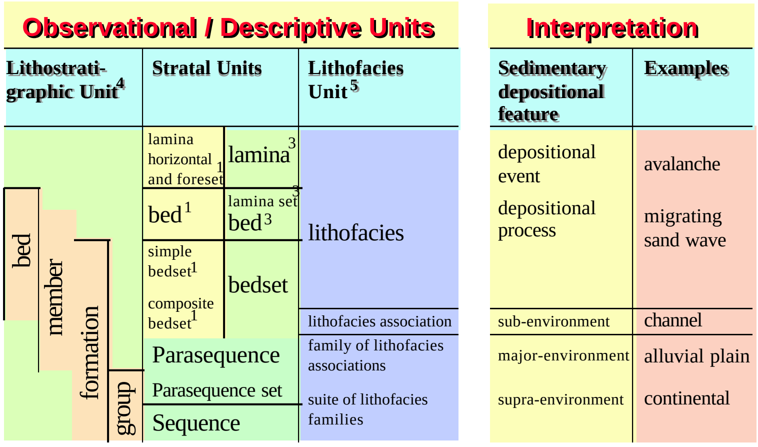


Figure 2.2: *Integration framework comprising three objects in a hierarchical system: member, sand and shale. The quantity thickness has been attached to all objects. Sonic and density quantities have been attached to the sand and shale objects. With this generic description any sequence of intercalating sands and shales with associated sonic and density logs can be described in any detail required. Some example 'realisations' are shown.*



1 Reineck & Singh, 1973
2 Van Wagoner et.al., 1990
3 Campbell, 1967
4 Hedberg, 1976
5 Reijers et.al., 1993

After Reijers, 1993

Figure 2.3: *Examples of hierarchical ordering systems used in sedimentary geology.*

The hierarchy in the integration framework corresponds to a stratigraphic hierarchy. Unfortunately, in sedimentary geology many different hierarchical ordering systems are used (Fig 2.3). The choice of ordering system depends of the type of geoscientific study. For a general-purpose tool like dGB-GDI this implies that for each study the user must be completely free to determine the ordering system as well as the kind of detail (scale levels) that is required. In the framework, the stratigraphic setting is defined in terms of geological objects called Framework units, which are ordered in a tree (Fig. 2.4).

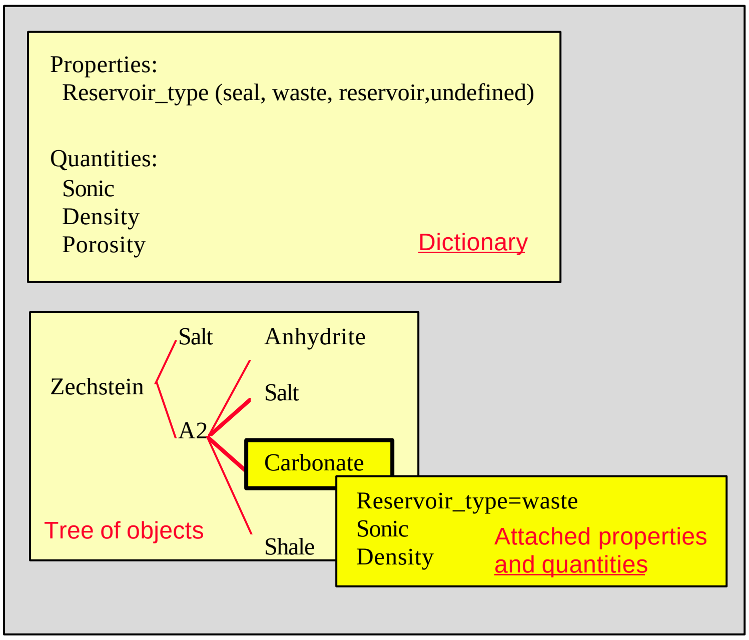


Figure 2.4: *Part of an integration framework. It is defined in three steps:*

- *a dictionary defines the properties and quantities to be studied*
- *geological objects are defined in a tree reflecting a hierarchical ordering system*
- *selected properties and quantities are attached to each of the objects.*

In the figure the attachments of the object 'Carbonate' are shown. The objects in the integration framework tree can be denoted nodes and leaves, or parents and children. Nodes and leaves refer to the position in the tree with a leaf always being at the smallest scale-level. Parents and children refer to inheritance with a parent always at a larger scale-level than its children.

A second requirement for a general-purpose quantitative interpretation tool is that also complete freedom must exist in the kind of data to be studied. In dGB-GDI the user defines himself which properties and quantities are to be studied. A distinction is made between *properties* and *quantities*. Properties are non-numeric features used to classify the data. For example a rock can be classified a 'seal', a 'reservoir' or a 'waste'. Quantities are features to which a value or a probability can be assigned. Quantities are, for example, the acoustic impedance, the thickness, the porosity, the permeability etc. Properties and quantities in dGB-GDI are defined in the "Framework Dictionary" window of the **Dictionary** option under **Geology menu**.

The integration framework is completely defined when the user has selected which properties and quantities from the dictionary must be attached to each of the geological objects defined in the tree. These geological objects are the building blocks for constructing stratigraphic models with attached physical properties and quantities. These models are called 'realisations' of the integration framework. They are either descriptions of deterministic data (factual wells) or they are simulated (see **2.1 Pseudo wells**).

Note that in the realisations, quantity values can be inherited from the geological objects at a larger scale levels. In order to use this feature the quantity must be attached in the framework to the parent unit and possibly to some of the children. If different values are given for a parent and a child, the child value will prevail. Each framework unit has a full reference name and a unit code (a user-defined mnemonic). In realisations of the integration framework the unit codes are used to give each object a unique identifier. This UnitID is a concatenation of unit codes separated by a dot ('.'), possibly followed by a comma (',') and a modifier value (see below). Each code can have an occurrence number attached between parentheses. Examples:

- lze.main.salt (equivalent to 'lze(1).main(1).salt(1)')
- lze.main(3).anh
- lze.main.ss(2),Gas

Unit IDs are used throughout the system for data identification and manipulation, see e.g. the **Features** module under **Wells menu**.

There are certain geological phenomena (such as fractured or fluid-filled zones in a stratigraphic column etc.) which cut right through the geometrical description in terms of framework units. In general, one can consider these phenomena as a result of circumstances which are not intrinsic to the framework unit. Very often the geoscientists are interested in the analysis and simulation of these phenomena e.g. in order to study the effects of hydrocarbons. In dGB-GDI such phenomena are described through a special type of features, called *Modifiers* and *Overlay-quantities*. They are defined in the framework dictionary (**Dictionary** module under **Geology menu**). Modifiers indicate that certain petrophysical quantities differ from their standard values due to the occurrence of a given geological phenomenon (e.g. fractured or fluid-filled zones etc.). Overlay-quantities describe the extent of the geological volume affected by the phenomenon.

The Modifiers and Overlay-quantities are used in the stochastic simulations of pseudo-wells in order to input corrected values of the above mentioned petrophysical quantities in the Gassmann Equation (**Apply Gassmann Equation** module under the **Utilities menu**).

The use of the *Modifiers* and the *Overlay-quantities* in the classification of the integration framework units and in the well simulations are given in Section 5.1 of this issue.

In realisations, modifier values are recognised by the Unit ID extension (e.g. ,Gas). Features can be extracted relative to the modifier values (e.g. average porosity gas-filled sands). When factual wells are entered into the system (**Edit** well module under **Wells menu**), the units that are affected by the modifier (e.g. units within the gas column) are specified.

2.2 Pseudo-wells

Pseudo-wells are primarily used in seismic applications. In general, they are used to quantify seismic measurements in terms of geological and/or petrophysical probabilities. For example, in lateral prediction studies pseudo-wells are used to find relationships between seismic features and underlying well features. In seismic pattern analysis they are used to relate the patterns to geological / petrophysical phenomena.

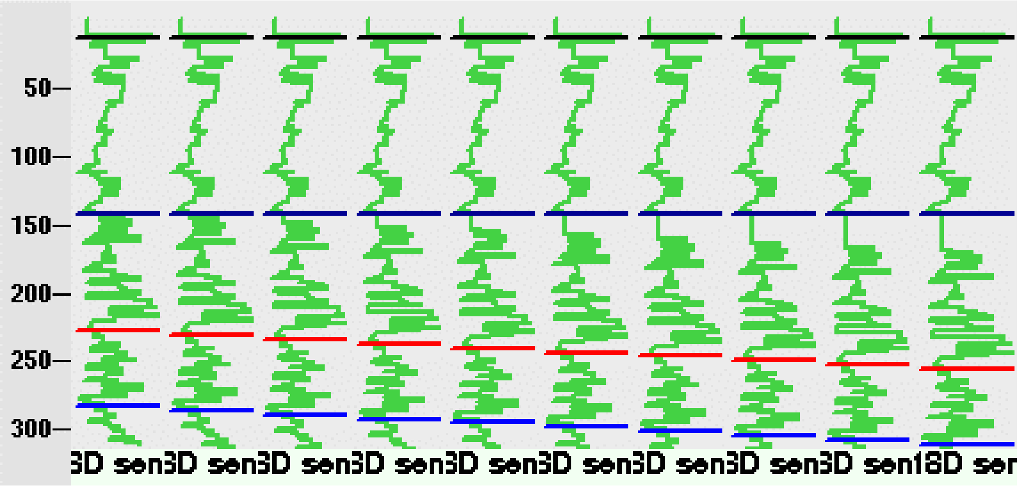


Figure 2.5: *Sensitivity analyses: ten pseudo-wells in which the thickness of one unit varies from 0 to 90 m with steps of 10 m. All other quantities remain unchanged. Only impdance logs are shown.*

Other applications are sensitivity analyses and feasibility studies. The objective in sensitivity analysis is to get a feel for the seismic response. Well information is varied in a controlled way and the corresponding seismic signals are visually inspected (Figs. 2.5 and 2.6). This is non-quantitative. Feasibility studies, in contrast are quantitative exercises. We want to establish how far the seismic data can be pushed, i.e. what information can be extracted from the seismic signals and what information is beyond the resolution.

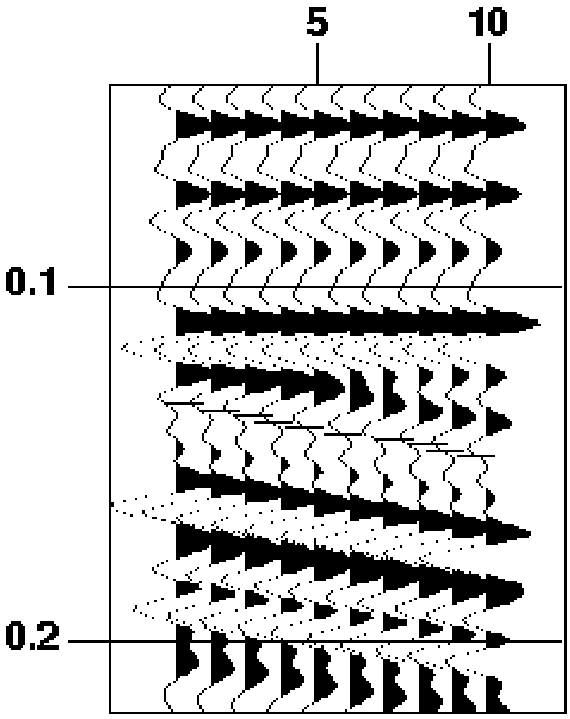


Figure 2.6: *Synthetic seismic traces corresponding to the pseudo-wells shown in Fig. 2.5. The tick mark indicates the position of the reference time. In this case the base of the unit that was varied.*

Some workers define pseudo-wells as stochastically simulated reflectivity sequences, others as simulated stratigraphic sequences and yet others as simulated logs. For the applications we have in mind we need to simulate all this information simultaneously. In dGB-GDI pseudo-wells are stochastic realisations of the integration framework. They are one-dimensional stratigraphic profiles with attached physical quantities (logs) but without spatial information. They are described in Unit IDs with attached values for thickness and other quantities (sonic, density etc.).

Pseudo-wells belong to a well group that was generated by the **Simulate** wells module (**Wells menu**). In general, synthetic seismic traces will be generated for the wells in a well group by the **Synthetics** module (**Seismic menu**). Operating in this way, high resolution well data and low(er) resolution seismic data is combined. The resulting integrated dataset can be used in various ways to analyse and quantify seismic data (de Groot, 1995).Input to the simulator consists of rules and constraints combined with stochastic information attached to integration framework units.

Pseudo-wells are constructed as follows:

- A stratigraphic profile is constructed by selecting framework units and their thicknesses.
- If modifiers and overlay-quantities were specified, these are simulated (e.g. Gas Column). The Unit IDs are updated accordingly (e.g. extension , Gas)
- Quantities are simulated for each of the units.
- Derived quantities (i.e. quantities specified in formulas) are calculated.

The results of generalised Markov chain analysis of real wells (**Sequences** module under **Analysis menu**) can be used by the pseudo-well simulator in order to generate pseudo-wells with similar stacking patterns.

The algorithm supports many different options for simulating quantities. For example, a value can be drawn from a specified probability distribution function, it can be calculated from other quantities, or it is varied in a controlled way. When the value is drawn from a probability distribution function, the quantity is a stochastic variable which can be correlated with any other stochastic variable in the simulator.

For example, suppose sonic and density logs have a negative correlation. In other words large density values correspond to small sonic values. In the algorithm this behaviour can be simulated by specifying a negative correlation coefficient between density and sonic quantities. Suppose, the density value is drawn first, then the algorithm updates the specified probability distribution function for the sonic quantity according to the specified correlation coefficient. The sonic value is now drawn from the updated distribution. Each quantity to be simulated is evaluated by the algorithm against the specified constraints. In this way only realistic values are accepted. For details on the mathematics, see de Groot et.al. (1996).

3 Artificial neural networks

3.1 General

Artificial neural networks, or connectionist models as they are sometimes referred to, have been inspired by what is known as the 'brain metaphor'. This means that these models try to copy the capabilities of the human brain into computer hardware or software. The human brain has a number of properties that are desirable for artificial systems (e.g. Schmidt, 1994):

- It is robust and fault tolerant. Even if nerve cells in the brain die (which is known to happen every day), the performance of the brain does not deteriorate immediately.
- It is flexible. This means that the human brain can adjust itself to new situations and can learn by experience.
- It can deal with information that is inconsistent, or contaminated with noise.
- It can handle unforeseen situations by applying knowledge from other domains and extrapolating this to new circumstances.
- It can deal with large amounts of input data and quickly extract the relevant properties from that data.
- It is highly parallel, hence it has a high performance.

Neural network research started in the forties. McCulloch and Pitts (1943) described the logical function of a biological neuron. They described that the transmission of neural signals is an all-or-nothing situation. A neuron fires only, if the cell has been stimulated above a certain threshold. The output signal will, in general, have a constant strength. In their paper, McCulloch and Pitts, described that networks consisting of many neurons might be used to develop the universal Turing machine (a kind of computer described by Turing (1937) that could, in principle, solve all mathematical problems). Research in neural networks was suddenly stopped following a publication by Minsky and Papert (1969). In this paper, it was shown that a relatively simple problem (the so-called XOR-problem) could not be solved by the linear algorithms used at the time. The major breakthrough which re-launched the interest in this technique has been the discovery in the eighties of a non-linear optimisation algorithm overcoming the previous limitations (Rumelhart et. al, 1986).

Neural networks have emerged in the last decade as a promising computing technique which enable computer systems to exhibit some of the desirable brain properties. Various types of networks have been applied successfully in a variety of scientific and technological fields. Examples are applications in industrial process modelling and control, ecological and biological modelling, sociological and economical sciences, as well as medicine (Kavli, 1992). Within the exploration and production world, neural network technology is now being applied to geologic log analysis (Doveton, 1994) and seismic attribute analysis (Schultz, 1994).

In dGB-GDI neural networks are used for pattern recognition. Three approaches can be recognised in neural network pattern recognition (Lippmann, 1989): supervised training, unsupervised training and combined supervised-unsupervised training. Supervised training approaches require the existence of representative datasets. Unsupervised techniques find structure in the data themselves, thereby extracting the relevant properties. In dGB-GDI Multi-Layer Perceptrons and Radial Basis Function networks are available for the supervised training approach. Unsupervised Vector Quantisers are available in the unsupervised mode. These networks are introduced in the following sections.

3.2 Multi-layer perceptrons (MLP)

The most general and most widely used neural network model is the 'multi-layer perceptron (MLP)'. The basic building block of this model is the perceptron (Fig. 3.1), a mathematical analogue of the biological neuron, first described by Rosenblatt (1962).

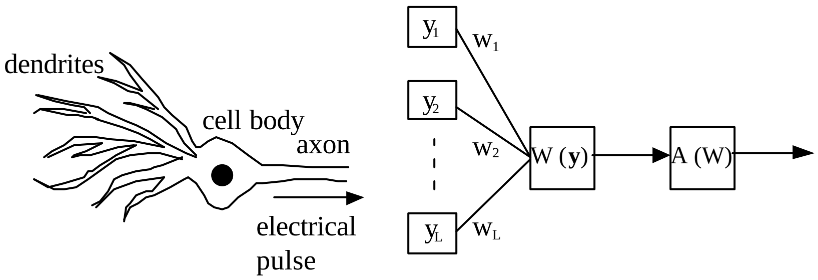


Fig. 3.1 *A biological neuron and a Perceptron*

The mathematical expression of the biological neuron can be written as an activation function A applied to a weighting function W, defined as:

$$W(\mathbf{y}) = \sum_{i=0}^{L} w_i y_i \,, \tag{3.1}$$

where:

$\mathbf{y}$ is the neural network input vector written as y_i with $i = 1,\ldots,L$ and weighting vector w_i with $i = 1,\ldots,L$.

The activation function of the classical perceptron (Fig. 3.2a) can now be written in the following form:

$$A(W) = \begin{cases} 1 & W > 0 \\ 0 & W \le 0 \end{cases}. \tag{3.2}$$

In MLPs the binary activation function is often replaced by a continuous function. The most widely used activation function is the sigmoid function (Fig. 3.2b). This function has the following form:

$$A(W) = \frac{2}{1 + \exp(-W)} - 1. \tag{3.3}$$

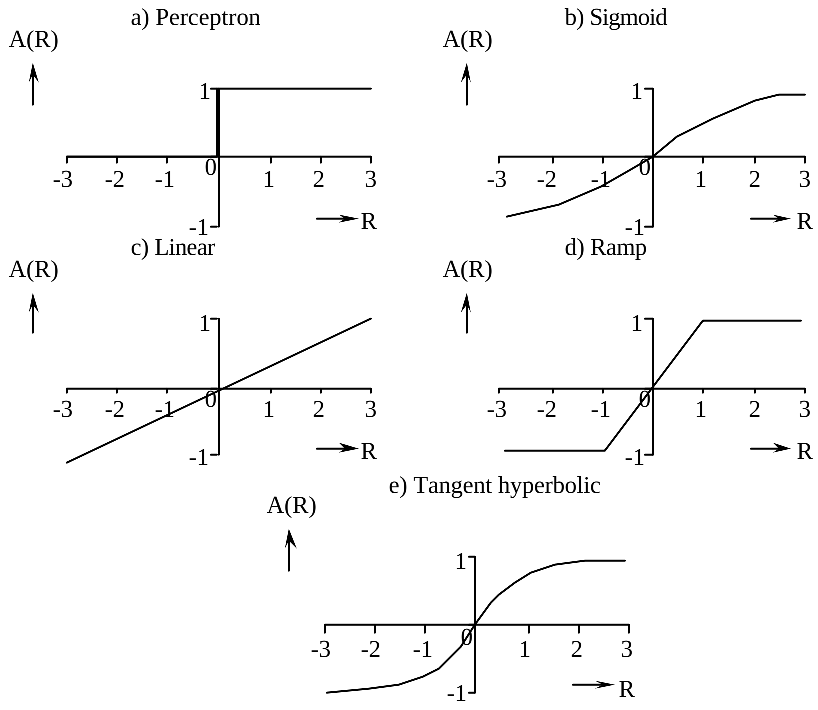


Fig. 3.2 *Different activation functions for MLP networks as supported in dGB's software. The prime-tangent hyperbolic function was used in this project. This function has the same mathematical expression as the tangent hyperbolic function but the update rules differ (see below).*

Other activation functions supported by the software are the linear, ramp and tangent hyperbolic functions. The linear function (Fig. 3.2c) is defined as:

$$A(W) = W . \tag{3.4}$$

The ramp function (Fig. 3.2d) is given by:

$$A(W) = \begin{cases} -1 & W < -1 \\ W & -1 \le W \le 1 \\ 1 & W > 1 \end{cases}. \tag{3.5}$$

The tangent hyperbolic function (Fig. 3.2e) is written as:

$$A(W) = \frac{\exp(W) - \exp(-W)}{\exp(W) + \exp(-W)}. \tag{3.6}$$

Two other activation functions are supported in dGB's software: the prime-sigmoid and prime-tangent hyperbolic. These functions have the same mathematical expressions as equations (3.3) and (3.6), respectively. The training algorithm treats the two types of functions differently. For the sigmoid and tangent hyperbolic functions, the derivative is used to update the weighting vector (Rich and Knight, 1991). For the prime-sigmoid and prime-tangent hyperbolic functions an offset is added to the absolute value of the derivative. This is done exclusively to avoid saturation problems during learning, where saturation means that continued learning does not lead to improved network performance. This modified procedure is used to update the weighting vector.

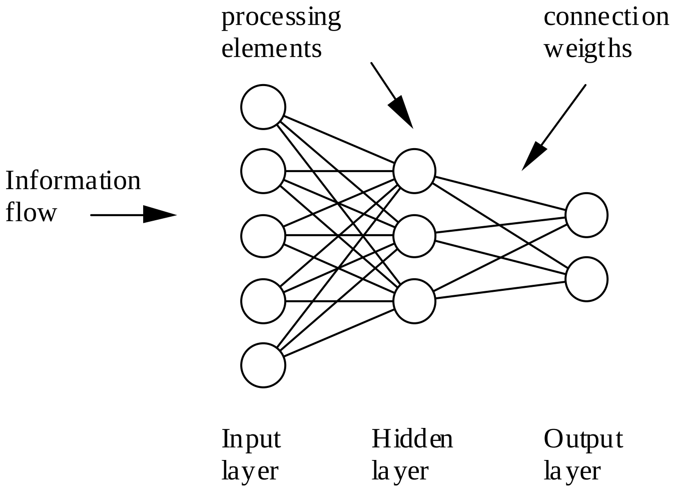


Fig. 3.3 *Schematic representation of a feed-forward layered neural network, such as a Multi-Layer Perceptron and a Radial Basis Function network.*

In a MLP the perceptrons are organised in layers (Fig. 3.3). In its simplest form, there are three layers; an input layer, a hidden layer and an output layer. There are no connections between neurons belonging to the same layer. The data flow between the layers is feed-forward. MLPs are trained on a representative dataset. This is a form of supervised learning. Known examples, consisting of input patterns and corresponding output patterns, are repeatedly offered to the network during the training phase. The 'back-propagation', learning, algorithm that is widely used to train this type of network attempts to minimise the error between the predicted network result and the known output by adjusting the weights of the connections. The algorithm was derived independently by a number of researchers. The modern form of back-propagation is often credited to Werbos (1974), LeCun (1985), Parker (1985) and Rumelhart et. al. (1986). A fast variation of backpropagation is given by Fahlman (1988).

MLPs have two properties of interest: abstraction and generalisation. Abstraction is the ability to extract the relevant features from the input pattern and discard the irrelevant ones. Generalisation allows the network, once trained, to recognise input pattern which were not part of the training set.

3.3 Radial Basis Function Neural Networks (RBF)

Radial basis functions have been used for data modelling (curve fitting) by many researchers, e.g. Powell (1987) and Poggio and Girossi (1989). Recently these functions have been put in a neural network paradigm in what is called Radial Basis Function (RBF) Neural Networks (Broomhead and Lowe (1988), Moody and Darken (1988), Lee and Kil (1988), Platt (1991)). Schultz et.al. (1994) applied RBF networks in a seismic reservoir characterisation study.

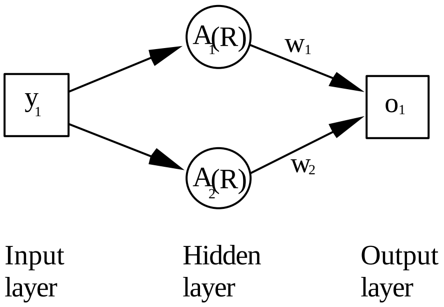


Fig. 3.4 *Schematic representation of a Radial Basis Function network for the case of a single input variable, two basis functions and one output variable.*

RBF networks have the same feed-forward layered architecture as MLP networks (Fig. 3.2), but the weighting function W and the activation function A are different. With RBF networks, there are only weights between output layer and hidden layer (Fig. 3.4). Each node in the hidden layer has a unique function, called the basis function. For the simple network of Fig. 3.3 with a single input, single output and two basis functions, the output is given by the sum of the two basis functions, each multiplied with its own weighting factor. In principle, any type of function can be used to act as basis function. For example, spline functions are used (Kavli, 1992), but the identification RBF network, applies only if radial basis functions are used.

Radial basis functions give local support to data points. The output of the hidden nodes, peaks when the input is near the centroid of the node, and then falls of symmetrically as the Euclidean distance between input and the centroid of a node increases (Fig. 3.5). The consequence of this behaviour is that RBF networks are good for data interpolation, but not good for data extrapolation.

Several different radial basis functions are in use, with the Gaussian function (Fig. 3.5a), being the most widely used. If the radial basis centre R is defined as:

$$R = \sqrt{\sum_{i=1}^{L} \frac{(y_i - m_i)^2}{s_i^2}} \, , \tag{3.7}$$

where:
m_i represents the centre location of each basis and s_i indicates a scaling of the width of each basis, then the Gaussian activation function is given by:

$$A(R) = \exp\left(-\frac{R^2}{2} \right). \tag{3.8}$$

Multiplication of the activation function $A(R)$ with a weighting factor w then yields the output o (Fig. 3.4).

Another widely used RBF function is the so-called Inverse Multi-Quadratic Equation (IMQE, Fig. 3.5b), defined as:

$$A(R) = \frac{1}{\sqrt{R + k^2}} \, , \tag{3.9}$$

where:
k is an empirically determined smoothing factor (default 0.5 in dGB’s software).

Note, that the widths in RBF functions are specified independently from each input dimension, making the functions elliptic rather than spherical. Note as well, that unlike the activation functions for MLPs no bias is included in the RBF functions.

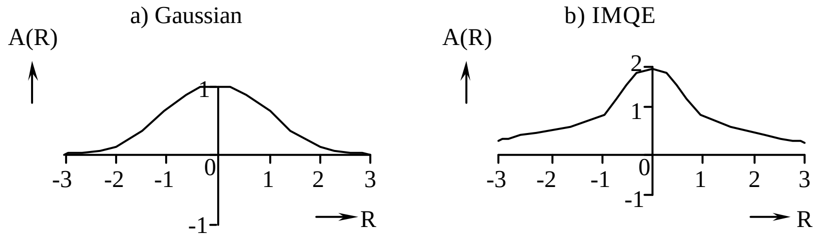


Fig. 3.5 *Activation functions supported in dGB's software for RBF networks. The Gaussian function has a m of 0 and a s of 1. The IMQE function has a m of 0, a s of 1 and a k of 0.5.*

Centre locations are typically determined by randomly selecting training examples from a large set of training data. The smoothing parameters and the number of nodes are typically adjusted empirically during training. RBF neural networks and MLPs have been compared by many workers. Kavli (1992) reported consistently better performance of RBF networks in five independent experiments. Another important aspect when comparing RBF networks and MLPs is the training speed. RBF networks can be trained within a fraction of the time that is required for training MLPs. RBF networks, however, generally require more nodes to obtain similar performances.

One of the training algorithms in dGB's software for RBFs is the so-called HSOL algorithm (Lee and Kill, 1989, Carlin, 1992). HSOL uses standard back propagation for updating the function parameters, but this learning algorithm also dynamically allocates new nodes in the hidden layer during training.

3.4 Unsupervised Vector Quantiser networks

In the preceding section Multi-Layer Perceptrons and Radial Basis Functions neural networks were introduced. These types of network belongs to the category of supervised learning approaches. Datasets with known input and target vectors are used to train and test these networks. In this section a type of network is introduced that belongs to the category of unsupervised, or competitive learning: the Unsupervised Vector Quantiser. The general aim of competitive learning is to find structure in the data themselves and thereby extracting the relevant properties or features. In the case of the UVQ the aim is to segment (cluster, classify) the data. Similar input vectors must be classified in the same category. The classes are found by the network itself from the correlations of the input data. Therefore, these networks are sometimes referred to as self-organising networks.

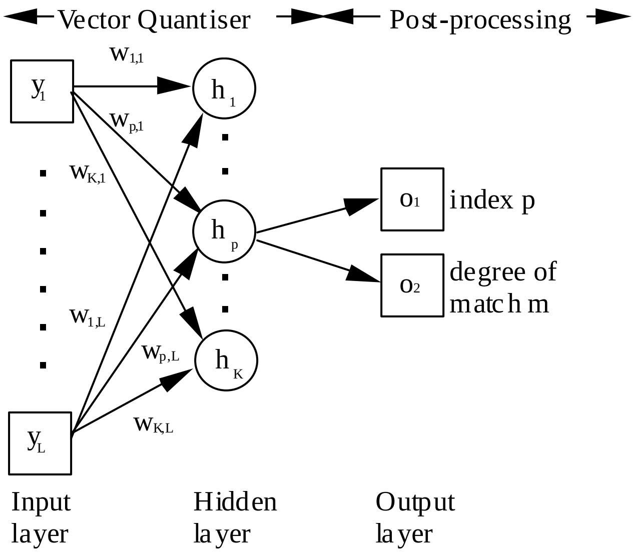


Fig. 3.6 *Schematic representation of the Unsupervised Vector Quantiser, as used in this study. The network consists of a vector quantiser part and a post-processing part. Two outputs are generated: the index of the winning hidden node (i.e. the class) and a degree of match, which indicates how close the input vector is located near the centre of the class.*

The UVQ that will be used is a modified version of a Learning Vector Quantiser (LVQ). Vector quantisation is an important application of competitive learning for data encoding and compression (Hertz et. al., 1991, and Haykin 1994). In vector quantisation an input vector is replaced by the index of the winning output unit. Vector quantisation requires a set of classes, or codebook to exist. Normally, a set of prototype vectors is used. The class is found by calculating the Euclidean distance to the prototype vectors. The nearest prototype vector is the winner. LVQ's are a supervised version of vector quantisation. In this case the prototype vectors are updated closer to the input, following a successful classification and further away from it when the classification is unsuccessful.

The unsupervised vector quantiser (UVQ) is quite similar to the LVQ. The prototype vectors are in the unsupervised case initialised as random vectors. The vector closest to the input vector is updated in the direction of the input vector.

The UVQ in this study consists of a two-layer vector quantiser followed by a post-processing output-layer (Fig. 3.6). In the vector quantiser part of the network, a single layer of hidden nodes h_i with $i = 1, \ldots, K$, where K indicates the number of classes, is fully connected with a set of input nodes y_j with $j = 1, \ldots, L$ via excitatory connections $w_{i,j}$. For each hidden node the net output is computed as the Euclidean distance to the input:

$$h_i(\mathbf{y}) = \sqrt{\sum_{j=1}^{L} (y_j - w_{i,j})^2} \quad i = 1, \ldots, K \cdot \tag{3.10}$$

In the learning phase the net outputs of all hidden nodes (classes) are compared in the post-processing layer. The hidden node with the smallest net output is designated the winner. The weighting vector $w_{p,j}$ associated with the winning node p is then updated according to:

$$w'_{i,j} = \begin{cases} w_{i,j} & i = 1, \ldots, p-1, p+1, \ldots K \quad j = 1, \ldots, L \\ w_{p,j} + h(y_j - w_{p,j}) & j = 1, \ldots, L, \end{cases} \tag{3.11}$$

where:

h is a empirically determined learning rate parameter and $w'_{i,j}$ is the updated weighting matrix. This update rule is known as the standard competitive learning rule. Updating is continued until no noticeable changes in the prototype vectors are observed.

In the application phase, the output layer consists of two nodes: one giving the index number of the winning node, and one giving a degree of match between the input vector and the prototype vector of this node. The degree of match m is computed as:

$$m = \left(1 - \frac{h_p(\mathbf{y})}{r\sqrt{L}}\right), \tag{3.12}$$

where r is the variation range for the training data.

In dGB's software, the input variables are rescaled so that they all fall in the range from -0.8 to 0.8 (therefore, r =1.6). The degree of match m can thus vary from 0 (minimum match) to 1 (perfect match).

The implication of rescaling is that all input variables will contribute equally to the classification result. In our application seismic signals are classified by feeding the UVQ network amplitudes at discrete sample positions. The samples are selected relative to a reference horizon. The rescaling procedure equalises the dynamic range at each sample position. It must be realised that some situations may exist where this approach does not yield an optimum result. For example, if, for the signals to be classified, a maximum amplitude and a zero-crossing always occur at the same sample positions, than the amplitude variations around the zero-crossing are relatively amplified.

This concludes the introduction to the integration framework and the type of neural networks that are available in dGB's software.

3.5 Neural network references

Broomhead, D.S., and Lowe, D., 1988. Multivariable functional interpolation and adaptive networks, Complex systems, 2:231-355, 1988.

Carlin, M., 1992. Radial Basis Function Networks and Nonlinear Data Modelling. Proceedings of Neuro-Nimes'92, Neural Networks and their Application, EC2, France, 1992, pp.623-633.

Doveton, J.H., 1994. Geologic Log Analysis Using Computer Methods. AAPG Computer Applications in Geology, No. 2. Association of American Petroleum Geologists.

Fahlman, 1988. An Empirical Study of Learning Speed in BackPropagation Networks. Technical Report CMU-CS-88-162, 1988. LeCun, Y., 1985. Une procedure d'apprentissage pour réseau à seauil asymétrique (A learning procedure for asymmetric threshold networks). Proceedings of Cognitiva 85, 599-604, Paris.

de Groot, P.F.M. and Bril, A.H., 1997. Quantitative interpretation generally needs no seismic attributes, 59th. EAGE conference, Geneva, 26-30 May 1997.

de Groot, P.F.M., 1996. Neural Network Experiments on Synthetic Seismic Data. Artificial Intelligence in the Petroleum Industry vol. 2 - Symbolic and computational applications. Éditions Technip, ed. Braunschweig, B. and Bremdal, B.A.

de Groot, P.F.M. and Bril, A.H., 1996. The Use of Pseudo-wells in Seismic Interpretation Studies. 58th. EAGE conference, Amsterdam, 3-7 June 1996.

de Groot, P.F.M, 1995. Seismic reservoir characterisation employing factual and simulated wells. PhD thesis, Delft University Press.

de Groot P.F.M. and Campbell A.E., 1995. Seismic reservoir characterisation in 'total space'; a Middle Eastern example. OAPEC/John Brown workshop 12-15 Sep. 1995, Delft.

de Groot P.F.M., 1995. 'Total Space Inversion'; Concept and Experiments. 57th. EAEG conference, June 1995, Glasgow.

de Groot, P.F.M. et.al., 1993. Seismic Reservoir characterisation using artificial neural networks and stochastic modelling techniques. 55th EAGE conference, June 1993, Stavanger.

Haykin, S., 1994. Neural Networks, A Comprehensive Foundation. Macmillan College Publishing Company, New York.

Hertz, J., Krogh, A. and Palmer, R.G., 1991. Introduction to the theory of neural computation, Lecture notes volume I. Santa Fe intsitute studies in the sciences of complexity, Addison-Wesley Publ. Comp.

Kavli, T.Ø., 1992. Learning Priciples in Dynamic Control. PhD.Thesis University of Oslo, ISBN no. 82-411-0394-8.

LeCun, Y., 1985. Une procedure d'apprentissage pour réseau à seauil asymétrique (A learning procedure for asymmetric threshold networks). Proceedings of Cognitiva 85, 599-604, Paris.

Lee, S. and Kil, R.M., 1988. Multilayer feedforward potential function network. IEEE International Conference on Neural Networks, I-161 - I-171, San Diego, 1988.

Lee, S. and Kil, R.M., 1989. Bidirectional Continuous Associator Based On Gaussian Potential Function Network. International Joint Conference on Neural Networks, vol.1, 1989, pp.45-53.

Lippmann R.P., 1989. Pattern Classification Using Neural Networks. IEEE Communications Magazine, November 1989.

McCulloch, W.S. and Pitts, W., 1943, A logical calculus of idea's immanent in nervous activity. Bulletin of Mathematical Biophysics 5, page 115-133. Reprinted in Anderson, J.A. and Rosenfield, E., 1988. Neurocomputing: Foundations of Research, Cambridge MIT Press.

Minsky, M. and Papert, S., 1969. Perceptrons: An Introduction to Computational Geometry. MIT Press, Cambridge, MA.

Moody, J., and Darken, C.J., 1988. Learning with localized receptive fields, in Proceedings of the 1988 Connectionist Models Summer School. pp. 133-143, editors: Touretzky et.al., Morgan-Kaufman.

Parker, D.B., 1985. Learning-Logic, Tech.Rep.TR-47. MIT Center for Computational Research in Economic and Management Science, Cambridge, MA. Rosenblatt, F., 1962. Principles of NeurodyWintershallics: Perceptrons and the Theory of Brain Mechanisms. Washington D.C., Spartan Books.

Platt, J., 1991. A resource-allocating network for function interpolation. Neural Computation, 3(2):213-225, 1991.

Poggio, T. and Girosi, F., 1989. A theory of networks for approximation and learning. Technical report, Artificial Intelligence Laboratory, Massachusets Institute of Technology, Jul. 1989.

Powel, M.J.D., 1987. Radial basis functions for multivariable interpolation: A review, in Algorithms for Approximation. editors: Mason, J.C., and Cox, M.G., Clarendon Press, London.

Rich, E. and Knight, K., 1991. Artificial Intelligence second edition. McGraw-Hill, Inc.

Rosenblatt, F., 1962. Principles of NeurodyWintershallics: Perceptrons and the Theory of Brain Mechanisms. Washington D.C., Spartan Books.

Rumelhart, D.E., Hinton, G.E., and Williams, R.J., 1986. Learning internal representations by error propagation, Parallel Distributed Processing. Editors: Rumelhart, D.E., McClelland, J.L. and the PDP Research group, 318-362, Cambridge, MA, MIT Press.

Schmidt, W.F., 1994, Neural Pattern Classifying Systems. PhD. thesis, TU Delft, CIP-DATA Koninklijke Bibliotheek, Den Haag, ISBN 90-9006716-7.

Schultz et.al.,1994. Seismic-guided estimation of log properties, Part 1: A data-driven interpretation methodology. The Leading Edge, May 1994; Part 2: Using artificial neural networks for nonlinear attribute calibration. The Leading Edge, June 1994; Part 3: A controlled study. The Leading Edge, July 1994.

Sinvhal A. and Sinvhal H., 1992. Seismic Modelling and Pattern Recognition in Oil Exploration. Kluwer Academic Publishers.

Turing A.M., 1937. On computable numbers, with an application to the Entscheidungsproblem. Proc. Lond. Math. Soc. (ser. 2), 42, 230-65; a correction 43, 544-6.

Werbos, P.J., 1974. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. PhD thesis, Harvard University, Cambridge, MA.

4 Monte Carlo statistics

4.1 Simulating correlated multi-variate stochastic variables

The following mathematical description is used in a simulation algorithm aimed at simulating wells, i.e. 1D-stratigraphic profiles with attached physical properties. In the algorithm, wells are constructed from so-called integration framework units (or entities). These entities are grouped at different scale levels. It is considered important that geological knowledge controls the selection of framework entities and that unrealistic realisations of variables can be redrawn. This implies that wells must be constructed one-by-one, entity-by-entity and variable-by-variable.

Variables in a computer are simulated using a (pseudo-) random number generator. When random variables are correlated, it is not simple, however, to simulate random draws using such a (pseudo-) random number generator. This is especially In the following discussion $\underline{X}$ is a stochastic vector. In our algorithm, $\underline{X}$ comprises all stochastic variables required for the simulation. A component of $\underline{X}$ is denoted by X_i. Examples of components are sonic, density, thickness and user-defined variables attached to framework entities. Each component X_i is assumed to be normally distributed with expectation m_i and variance s_i^2, symbolically written as: $X_i \sim N(m_i, s_i^2)$. The vector of expectation will be denoted $\underline{m}$. The components are assumed to be correlated. The covariance between components i and j is indicated by s_{ij}. Note, that the covariance between component i and itself, s_{ii} equals s_i^2. The matrix of covariances will be denoted as Σ. When the covariance s_{ij} is normalised with the standard deviations s_i and s_j, we obtain the correlation coefficient r_{ij}, symbolically written as: $r_{ij} = {s_{ij}} \big/ {\left(s_i * s_j\right)}$. The matrix of correlation coefficients will be denoted by C. Sets of components can be grouped into subvectors of $\underline{X}$ denoted by $\underline{X}^{(i)}$, An example of a subvector $\underline{X}^{(i)}$ is that part of stochastic vector $\underline{X}$ comprising correlated thicknesses of a set of layers. The theorems given hereafter apply to the general case of drawing entire subvectors. However, for design reasons, the variables are, drawn one-by-one, in the final implementation of the algorithm. In other words the subvector $\underline{X}^{(i)}$ to be drawn has only one component. This is illustrated by the example at the end of this Chapter.

We require two theorems for our algorithm to work. Theorem 4.1 is used for updating the expectation and covariance matrix of a variable to be drawn, given some already drawn correlated variables (Mardia, 1979). This theorem requires the covariance matrix to be specified completely. In general, the user will not be in a position to specify all coefficients. Therefore, the unspecified correlation coefficients must be approximated first. This is accomplished with Theorem 4.2 (Meeuwissen et.al., 1994).

In the following discussion, first the two theorems are given, followed by an illustration of their use with an example.

Theorem 4.1

First we introduce some notation. Let $\underline{X}$ be a n-dimensional stochastic vector which is partitioned as follows: true when the variables must be drawn one-by-one, as in our application. The realisations of already drawn variables will in that case influence the realisation of the variable to be drawn. For example, let us assume that a positive correlation exist between the thicknesses of two layers. When for the first layer a small thickness is drawn, then also for the second layer a small thickness must be drawn. In the case of normally distributed random variables, it is possible to draw the variables consecutively from the marginal distributions. Each time a variable is to be drawn, its marginal distribution must first be updated for the already drawn variables to which it is correlated.

$$\underline{X} = \begin{pmatrix} \underline{X}^{(1)} \\ \underline{X}^{(2)} \end{pmatrix}, \tag{4.1}$$

with expectation $\mathrm{E}[\underline{X}]$ equal to $\underline{m}$:

$$\underline{m} = \mathrm{E}[\underline{X}] = \begin{pmatrix} \underline{m}^{(1)} \\ \underline{m}^{(2)} \end{pmatrix}, \tag{4.2}$$

and a positive definite covariance matrix $Cov(\underline{X})$ given by:

$$\Sigma = Cov(\underline{X}) = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}. \qquad (4.3)$$

Suppose $\underline{X}$ is multivariate normally distributed with expectation $\underline{m}$ and covariance matrix Σ, which can be symbolically written as:

$$\underline{X} \sim MVN\left(\underline{m}, \Sigma\right). \qquad (4.4)$$

Here ~ denotes 'is distributed as' and MVN indicates multivariate normally distributed. Then the conditional distribution of $\underline{X}^{(1)}$ given a realisation $\underline{x}^{(2)}$ of $\underline{X}^{(2)}$ is multivariate normally distributed with expectation:

$$\hat{\underline{m}}^{(1)} = \underline{m}^{(1)} + \Sigma_{12}\Sigma_{22}^{-1}(\underline{x}^{(2)} - \underline{m}^{(2)}), \qquad (4.5)$$

where $\hat{\underline{m}}^{(1)}$ is the updated expectation. The updated covariance matrix $\hat{\Sigma}_{11}$ is given by:

$$\hat{\Sigma}_{11} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}. \qquad (4.6)$$

Theorem 4.2

Suppose X_1, X_2 and X_3 are correlated random variables which satisfy:

$$E[X_1 | X_2 = x_2] \quad \text{is linear in } x_2, \qquad (4.7)$$

and

$$E[X_1 | X_3 = x_3] \quad \text{is linear in } x_3. \qquad (4.8)$$

Then, given the correlation coefficients r_{12} between the pairs X_1 and X_2 and r_{13} between X_1 and X_3, the correlation coefficient r_{23} is given by:

$$r_{23} = r_{12} r_{13}. \tag{4.9}$$

The conditions in the theorem imply, say for X_1, X_2, that given a realisation x_2 of variable X_2, the expectation of X_1 shifts linearly towards x_2. For normal distributions this is always satisfied, as can be seen from theorem 4.1, equation (4.5).

Although this theorem applies to three variables with one missing correlation coefficient only, we are going to use it also, without strict theoretical justification, for more than three variables where several correlation coefficients may be missing. We must note here, that, for more then three correlated variables, the positive definiteness of the covariance matrix may be violated by this procedure. In practice we have seen this happen only in some rare cases.

We will illustrate the use of these theorems with the following example. Suppose the correlation matrix has been specified for five variables as follows:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & * & 0.8 & * \\ 0 & * & 1 & * & 0.6 \\ 0 & 0.8 & * & 1 & 0.4 \\ 0 & * & 0.6 & 0.4 & 1 \end{bmatrix}. \tag{4.10}$$

In this particular example, r_{24} , r_{35} and r_{45} are known coefficients and r_{34} , r_{25} and r_{23} are unknown, which is indicated in the matrix by the * symbol. Using 4.9 we can determine two of the unspecified correlation coefficients.

$$r_{34} = r_{35} r_{54} = 0.24, \tag{4.11}$$

and

$$r_{25} = r_{24} r_{45} = 0.32. \tag{4.12}$$

However, r_{23} cannot be determined by combination of two of the given correlation coefficients. In a second step, we can approximate it using the previously determined correlation coefficients:

$$r_{23} = r_{24} r_{43}, \tag{4.13}$$

which can be expanded using 4.11 to:

$$r_{23} = r_{24} r_{35} r_{54} = 0.192 \,. \tag{4.14}$$

Note, that we could also have used:

$$r_{23} = r_{25} r_{53} = r_{24} r_{45} r_{53} \,. \tag{4.15}$$

In this particular case, the same value for r_{23} will be obtained for (4.14) and (4.15). In general, however the approximation is not unique. If several combinations are possible, in which the number of initially specified correlation coefficients differs, then a selection is made from the combinations with the least number of initial coefficients. From these we, arbitrarily choose one of the possible combinations. Thus, if in a different example, r_{23}, r_{34}, r_{35}, r_{45} would have been specified, then we can obtain r_{25}, either from:

$$r_{25} = r_{23} r_{35}, \tag{4.16}$$

or, from:

$$r_{25} = r_{23} r_{34} r_{45} \,. \tag{4.17}$$

The former expression is favoured because it contains less specified correlation coefficients.

With respect to the approximate nature of the procedure, we emphasise that after multiplying correlation coefficients, the resulting number comes closer and closer to zero. Therefore, the effect of the resulting approximation of the correlation coefficient decreases rapidly. Hence, we argue that making an error in the approximation has little effect when many terms are involved.

After application of the above procedure, the correlation matrix of (4.10) can be approximated by:

$$\tilde{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0.192 & 0.8 & 0.32 \\ 0 & 0.192 & 1 & 0.24 & 0.6 \\ 0 & 0.8 & 0.24 & 1 & 0.4 \\ 0 & 0.32 & 0.6 & 0.4 & 1 \end{bmatrix}. \tag{4.18}$$

We can now draw samples for all variables. Suppose we would like to draw them in the order X_3, X_5, X_1, X_2, X_4. When selecting X_3, no other has been drawn, so we can just draw it from its marginal probability density function $X_3 \sim N(m_3, s_3^2)$. Now X_5 must be drawn, conditioned on the x_3 value. Using theorem 4.1, we find:

$$\hat{m}_5 = m_5 + s_{35}(s_3^2)^{-1}(x_3 - m_3), \tag{4.19}$$

and

$$\hat{s}_5^2 = s_5^2 - s_{35}(s_3^2)^{-1} s_{53}, \tag{4.20}$$

where

$$s_{35} = r_{35} s_3 s_5, \tag{4.21}$$

is the covariance between X_3 and X_5. Now X_5 can be drawn from $N(\hat{m}_5, \hat{s}_5^2)$.

Now X_1 is to be drawn. Since it is independent of X_2, X_3, X_4 and X_5 it can be drawn from its marginal distribution $N(m_1, s_1^2)$. Finally, for X_2 and X_4 we use:

$$\hat{m}_2 = m_2 + \begin{bmatrix} s_{23} s_{25} \end{bmatrix} \begin{bmatrix} s_3^2 & s_{35} \\ s_{35} & s_5^2 \end{bmatrix}^{-1} \begin{bmatrix} x_3 - m_3 \\ x_5 - m_5 \end{bmatrix}, \tag{4.22}$$

$$\hat{s}_2^2 = s_2^2 - \begin{bmatrix} s_{23} \, s_{25} \end{bmatrix} \begin{bmatrix} s_3^2 & s_{35} \\ s_{35} & s_5^2 \end{bmatrix}^{-1} \begin{bmatrix} s_{23} \\ s_{25} \end{bmatrix}, \tag{4.23}$$

and

$$\hat{m}_4 = m_4 + \begin{bmatrix} s_{24} \, s_{34} \, s_{54} \end{bmatrix} \begin{bmatrix} s_2^2 & s_{23} & s_{25} \\ s_{23} & s_3^2 & s_{35} \\ s_{25} & s_{35} & s_5^2 \end{bmatrix}^{-1} \begin{bmatrix} x_2 - m_2 \\ x_3 - m_3 \\ x_5 - m_5 \end{bmatrix}, \tag{4.24}$$

$$\hat{s}_4^2 = s_4^2 - \begin{bmatrix} s_{24} \, s_{34} \, s_{54} \end{bmatrix} \begin{bmatrix} s_2^2 & s_{23} & s_{25} \\ s_{23} & s_3^2 & s_{35} \\ s_{25} & s_{35} & s_5^2 \end{bmatrix}^{-1} \begin{bmatrix} s_{24} \\ s_{34} \\ s_{54} \end{bmatrix}, \tag{4.25}$$

respectively.

This allows us to draw the variables one by one in any order. Also, we can redraw any one of the variables when needed, and condition on the latest drawn value for each of the correlated variables.

4.2 Monte Carlo references

Davis, J.C., 1986. Statistics and Data Analysis in Geology. John Wiley & Sons.

Dubrule, O., 1989. A review of stochastic Models for Petroleum Industry. Geostatistics, M. Armstrong (ed.), 493-506, Kluwer, Dordrecht.

Mardia, K.V., Kent, J.T. and Bibby, J.M., 1979. Multivariate Analysis, Academic Press London.

Meeuwissen, A.M.H. and Cooke, R.H., 1994. Modelling joint probability distributions using graph and tree-dependence. submitted.

de Groot, P.F.M., Bril, A.H., Floris F.J.T, and Campbell, A.E. 1996. Monte Carlo simulation of wells. Geophysics, Vol. 61, No. 3 (May-June 1996); P.631-638.

de Groot, P.F.M. and Bril, A.H., 1996. The Use of Pseudo-wells in Seismic Interpretation Studies. 58th. EAGE conference, Amsterdam, 3-7 June 1996.

de Groot, P.F.M, 1995. Seismic reservoir characterisation employing factual and simulated wells. PhD thesis, Delft University Press.

Sinvhal A. and Sinvhal H., 1992. Seismic Modelling and Pattern Recognition in Oil Exploration. Kluwer Academic Publishers.

5 Fluid replacement

5.1 Basic equations

The velocity and density of a porous medium are influenced by the fluids that are present in the pore space. The bulk density r as a function of porosity f is formulated in the following equation:

$$r = (1 - f) r_S + f r_f , \tag{5.1}$$

where:
r_S denotes the density of the solid fraction and r_f the density of the pore fluid.

The relationship between velocity, porosity and fluid content is more complicated. Willie's time average equation, or (empirical) extensions to this formula have been used by many workers (e.g. de Haas, 1992). Willie's equation (Wyllie et.al., 1958) is formulated as:

$$t = (1 - f) t_S + f t_f , \tag{5.2}$$

where t denotes sonic travel time of the rock, t_S the travel time in the solid matrix (i.e. empty porous rock), t_f travel time for the pore fluid and f is the porosity.

This equation and the empirical extensions thereof, are not very reliable when used as fluid replacement algorithms, especially not for the gas-fill replacements.

The most widely used rock-physics models for studying wave propagation effects in porous media are the theoretical Gassman and Biot-Gassmann equations (see, e.g. Crans and Berkhout). The main difference between the two models is that Gassmann is applicable to low seismic frequencies only while Biot-Gassmann is also used for predicting frequency dependent velocities. The low frequency limit of Biot-Gassmann equals the Gassmann's equation which is formulated as:

$$c_p = \sqrt{\frac{k_s}{r}\left(3\left(\frac{1-s_b}{1+s_b}\right)b + \frac{(1-b)^2}{1-b+f\left(k_s/k_f-1\right)}\right)}, \qquad (5.3)$$

where:
c_p denotes the seismic velocity for compressional waves. For an explanation of the other symbols see Table 5.1.

Table 5.1 *Rock and pore parameter definitions.*

Parameter	Description	Unit
k_s	compressibility modulus solid	N/m^2
k_f	compressibility modulus fluid	N/m^2
s_b	Poisson ratio	-
f	Porosity	-
r_s	density solid	kg/m^3
r_f	density fluid	kg/m^3
r	density bulk	kg/m^3
c_p	P-wave velocity	m/s

Direct application of this equation to calculate rock velocities is of limited use since the Poisson ratio s_b and the compressibility moduli k_f , k_s are, in general unknown. If, however, the velocity of a rock with a given saturation is known, then the Gassmann equation can be used to calculate the velocity of the same rock with a different saturation, as follows.

Assume that k_w, k_{hc}, s_w, s_{hc}, s_b, r_s, r_w and r_{hc} are input parameters specified by the user. For a description see Table 5.1; the index *hc* denotes hydrocarbons. First calculate the porosity f using (5.1) for the brine-filled case. Then calculate r_f using s_w, s_{hc}, r_w and r_{hc}. Now calculate the density of the hydrocarbon-filled case using (5.1). Use Wood's law for the compressibility modulus of the fluid mixture k_f. Wood's law is formulated as:

$$1/k_f = s_w/k_w + s_{hc}/k_{hc}. \tag{5.4}$$

Now the Gassmann equation (5.3) can be employed to calculate the velocity of the hydrocarbon-filled rock. The Gassmann equation, as a fluid replacement algorithm is applied in two steps. In the first step, the frame strength, or Biot coefficient b defined as k_m/k_s where k_m is the compressibility modulus of the matrix, is derived from the sound velocity of the brine-filled rock. Defining g as:

$$g = 3(1-s_b)/(1+s_b)\ , \tag{5.5}$$

and B as:

$$B = f(k_s/k_f - 1)\ , \tag{5.6}$$

then b can be calculated as:

$$b = 1 - A \pm \sqrt{(A+B)^2 - (B^2/(1-g))}, \tag{5.7}$$

with:

$$A = ((rc_p^2/k_s) + g(B-1))/2(1-g). \tag{5.8}$$

In the second step of the fluid replacement algorithm the assumption is made that f, b and s_b are independent from the fluid properties. Substitution of these variables together with the properties of the fluid mixture in (5.3), yields the velocity of the hydrocarbon filled rock. The sonic travel time follows as the reciprocal of this velocity.

As stated above, the Gassmann equation assumes the velocity to be independent from frequency. Biot (1956b) has proved, however, that velocity does depend on frequency. At low (seismic) frequencies this effect can in general be ignored. Anderson (1984) proved that this effect can be significant in special cases, e.g. low permeability rocks with low saturation gas in the pores.

5.2 Fluid replacement references

Anderson, 1984. Fluid and Frequency effects on sonic velocities, SWPLA twenty fifth annual sonic logging symposium, June 10-13 1984.

Biot, M.A., 1956a. Theory of propagation of elastic waves in a fluid-saturated porous solid. I Low-frequency range. J. Acoust. Soc. Am., v. 28, p 168-178.

Biot, M.A., 1956b. Theory of propagation of elastic waves in a fluid-saturated porous solid. II High-frequency range. J. Acoust. Soc. Am., v. 28, p 179-191.

Bunch, A.W.H. and Dromgoole, P.W., 1995. Lithology and fluid prediction from seismic and well data. Petroleum Geoscience, Vol 1, No. 1, Jan. 1995.

Crans, W., and Berkhout, A.J., 1980. Assessment of seismic amplitude anomalies. Oil and Gas Journal, no. 11, pp. 156-168.

Doveton, J.H., 1994. Geologic Log Analysis Using Computer Methods. AAPG Computer Applications in Geology, No. 2. Association of American Petroleum Geologists.

de Groot, P.F.M, 1995. Seismic reservoir characterisation employing factual and simulated wells. PhD thesis, Delft University Press.

Mijnssen, F.C.J., Tyler, N. and Weber, K.J., 1993. Knowledge base developments for the estimation of reservoir rock properties in the interwell area: examples from the Texas Gulf coast. Special Publication Number 15. International Association of Sedimentologists. Editors: Flint, S.S, and Bryant, I.D.

Neidell, N.S., Beard, J.H., Cook, E.E., 1986. Use of Seismic-Derived Velocities for Exploration on Land: Seismic Porosity and Direct Gas Detection. Seismic Stratigraphy II - An Integrated Approach. AAPG Memoir 39. Editors: Berg, O.R. and Woolverton, D.G. Association of American Petroleum Geologists.

Wyllie, M.R., Gregory, A.R., and Gardner, G.H.F., 1958. An Experimental Investigation of Factors Affecting Elastic Wave Velocities in Porous Media. Geophysics, v. 23., p459-493.

6 Generalised Markov Chain Analysis

6.1 Markov chains

In many geological settings facies or sub-facies are genetically related to each other. This relationship is expressed as a certain degree of vertical ordering in parts of a stratigraphic sequence, and various stacking patterns may occur. Deposits can thus be thought of as laid down under the control of a Markovian process, a process where succeeding events in a sequence, in a probabilistic sense, are partially dependant on one or more immediately preceding events. The extent of the dependency in terms of preceding events is called the property or the order m of the Markov process.

Significant depositional stacking patterns in stratigraphic sequences can be detected and characterised with a Markov Chain Analysis (MCA). The sequence is classified according to a chosen level of stratigraphic detail. The level of stratigraphic detail gives a number of mutually exclusive stratigraphic states which then are indexed. For a given order m, n mutually exclusive states can be combined into n^m patterns. A certain order is assumed and the number of times that transition occurs from a possible pattern to a given state is tabulated in a transition frequency matrix. Based on this, a χ^2-test is performed where a null-hypothesis of random deposition is tested against a hypothesis of depositional dependency with the assumed order. The analysis yields a probability structure for how occurrence of a stratigraphic unit is related to preceding units. If any significant stacking relation between units in a sequence are found, this information can be used for interpretation of the depositional environment as well as for simulation of stratigraphic sequences.

Traditionally the occurrences have been tabulated in such a way that the lower state of a pattern indicated the row in the matrix and the state to which transition occurred indicated the column. Thus, for higher order (m>1), information on the intermediate states is lost in the analysis, and the resulting probability structure will not properly reflect the significant patterns and may even be skewed. Interpretation and simulation will be equally affected. To avoid this, the full pattern is used in the dGB-GDI approach which we call Extended Markov Chain Analysis (EMCA).

Furthermore in traditional MCA, transitions are recorded either, at fixed thickness intervals or, when there is a distinct state change. In dGB-GDI a deliberate choice has been made to do the analysis solely on unit ID and not involve properties like thickness. Our reasoning is that the sampling interval highly influences the probability structure and the test result. Thick units may be oversampled and/or thin units neglected, both effects skew a statistical test. This approach actually only works well if all units have similar thickness distribution.

6.2 Extended Markov Chain Analysis

In dGB-GDI the stratigraphic states corresponds to the framework units. EMCA can be performed at any parent level in the framework. The children present at this level supplies the base of mutually exclusive states at this level.

Two types of analysis are supported:

1. Immediate repetition of a state allowed. Thus in packages of e.g. sands which are characterised as being of the same state type, transitions between layers in the package are used in the analysis and included in the probability structure of the state transition. This is similar to sampling with a thickness sampling interval in standard MCA, but without the thickness element.
2. Immediate repetition of a state not allowed. Only transitions between different states are counted. If a e.g. a sand package is present, it will be counted as one distinct occurrence of a sand layer. This is called Embedded EMCA.

If immediate repetition is allowed but only few packages are found, this can influence the analysis. To avoid this, it is possible to re-do the analysis with the option to count the packages as one layer only.

Assuming a certain dependency between points in the sampled sequence, the assumed order m (length of patterns) gives a number of possible combinations of the n states. The occurrence of transition from a given pattern to a given state is tabulated in a transition frequency matrix which is a (n^m,n) matrix. Considering the full pattern, all the transition information is used and the matrix representation is a tensor representation of the available information : from the Grand Total (total number of transitions), which corresponds to no spatial information, to the m-th order transitions, which corresponds to information of m-th order spatial dependency.

The null-hypothesis matrix (the zero-order matrix) for a given assumed order is constructed assuming that the individual state transitions in a pattern and from a pattern are independent. This gives a basis for what information can be used to construct ‘expected’ random transition frequencies. The only available ‘random’ transition information between states is the total amount of transitions from any pattern to a given state and the Grand Total. These quantities are used to compute the unconditional probabilities for state transitions in a pattern and from a pattern to a given state. The product of these probabilities is an expression of the proportion of pattern-to-given-state transitions of the Grand Total that could be expected to occur in case of random transition. The ‘expected’ amount of occurrences is then simply this probability multiplied by the Grand Total. For Embedded EMCA, transition frequency for patterns involving self-transitions are set to zero and the allowed transition frequency patterns are normalised to the sum of the actual observed transitions to a given state.

A problem with the traditional χ^2-test is the strong dependency on the amount of actual transition observations and that the ‘expected’ random transition frequencies, as opposed to the observed transition frequencies, most often are not integer values. This introduces an unrealistic and uncontrolled error in the χ^2-test. The error diminishes with increasing number of observations, but it highly affects the chi contribution if e.g. an observation of 1 is tested against an ‘expected’ value of 0.1. Since the observed transition frequencies decrease rapidly for higher orders, these limitations would highly affect an EMCA. To counter this, we have introduced a correction to the traditional χ^2-test, a correction based on how the error is expected to influence the test.

6.3 Stationarity

Sedimentation can change in space and time, which may lead to space and time dependent transition count distributions. Non-stationary vertical or lateral effects on a MCA can be removed by splitting a sequence or a set of sequences into segments or subsets that are stationary. The χ^2-test can be used for vertical as well as lateral stationarity tests. Transition frequencies from a full sequence (vertical) or a full set of sequences (lateral) are tested against transition frequencies from a segment (vertical) or one sequence (lateral) which have been normalised to create ‘expected’ transition frequencies that are commensurable with the observed transition frequencies in the full sequence or sequence set.

6.4 Markov chain references

Allègre, C., 1964. Vers une logique mathémathique des series sédimentaires: Bulletin de Société Géologique de France, Series 7, v. 6, p. 214-218.

Davis, J.C., 1986. Statistics and Data Analysis in Geology. John Wiley & Sons.

Doveton, J.H., 1994. Theory and Applications of Vertical Variability Measures from Markov Chain Analysis. AAPG Computer Applications in Geology, No 3, The American Association of Petroleum Geologists.

Miall, A.D., 1973. Markov chain analysis applied to ancient alluvial plain succesion, Sedimentology, v. 20, 347-364.

Roux, J.P., 1994. Spreadsheet Procedure for Modified First-order Embedded Markov Analysis of Cyclicity in Sediments, Computers and Geoscience, v. 20, No 1, 17-22

7 Model Probability Module

7.1 Concept

One of the keys to successful use of dGB-GDI is the presence of dGB-GDI wells (i.e. 1-D geological models) with a corresponding (tied) seismic trace. If anywhere else in the area the geology is exactly the same, we should find the same seismic trace there, that is if processing has been successful.

Unfortunately, it does not work the other way round: identical seismic traces can be generated by completely different geology. Still, the amount of resemblance with a trace that has a known geology attached to it is valuable information that we want to use in our inversions.

Suppose we have two wells in the area. Let us assume well 1 has 10% porosity, and well 2 has 15% porosity. First, we correlate the two well traces with all other traces of the survey, so creating two correlation maps. At the well positions, we find that the other well trace correlates 0.5, whereas the well trace itself will of course have a correlation of 1. Let us consider a new position exactly between the two wells.

Consider the following possibilities:

1] Correlations are 0.9 and 0.6

2] Correlations are 0.6 and 0.6

3] Correlations are 0.9 and 0.9

4] Correlations are 0.6 and 0.9

Possibility 1] suggests a lower porosity than 2] and 3], which are all expected to be lower than 4].

The most important conclusions we can already make after this “evaluation” are:

- These exercises do uncover information on the subsurface
- We can not determine 'absolute' probabilities

Now consider a situation where we have a number of models which are known to be representative for the whole area. That is, the wells describe all possible geologies for the area and the occurrences reflect the abundances of the various components and quantitative properties. This property of the set of models is essential to take a next step in our analyses.

Each of the models will deliver a correlation of its seismic trace with all other traces in the seismic data cube. Consider the set of correlations at a selected position in the survey. The numbers represent the correlation of the actual seismic trace at this position with each of the model traces.

Imagine that all models whose traces have a high correlation also have a high porosity. Because the models are representative for the area, we expect the porosity at the position to be high. Exactly how high, and with which certainty is still a matter of discussion.

7.2 Scoring models

The central issue in the above procedures is the ranking of models at a certain position. From different sources of information we can put constraints upon the probability of finding a certain model at a certain position. Next to seismic correlation we have geostatistics as a constraint. Geostatistical information gives us another way to create a ranking in the probability of each of the models at a certain position. But it also gives more. Near a control point (existing well) we have less freedom in choosing a model than far away from any control point. This is expressed in the standard deviation which is determined during geostatistical analysis.

All possible constraints on the ranking of the different models must be unified to be able to get an integrated view on the subsurface parameters. Therefore, we have defined new concepts, 'Score' and 'Score significance'.

A measure of probability is called a Score, when the following conditions are met:

1. the maximum score corresponds to probability 1.
2. the minimum score corresponds to probability 0.

Score is a monotonous increasing function of Probability. In other words when (S_N, P_N) denotes a (score, corresponding probability) pair then for any (S_1, P_1) and (S_2, P_2):

1. if $S_1= S_2$ then $P_1= P_2$.
2. if $S_1> S_2$ then $P_1> P_2$

The prime importance of defining the concept is that we are allowed to not know the exact function Score -> Probability. This serves to incorporate seismic correlation, but probably also geostatistics, which may not be so exact as some people would like it to be.

Score significance is a value between 0 and 1 used to describe the importance of a certain Score. A Score significance of 0 means that no conclusions can be drawn from the Score at all, a value of 1 means that the score fully constrains the value. If the Score significance is 1, the score can only be the maximum score, or the minimum. For example, at a well position the score significance is 1: a model either exactly matches the actual model (score=max) or it is rejected (score=min). For scores originating from geostatistics, the significance will be defined as:

$$1-s\ /s_{\max}$$

Which is consistent with the fact that s =0 (only possible with variogram models with zero nugget effect) will result in a Score significance of 1. Outside the range of some variogram models the significance can become 0.

In dGB-GDI, scoring is done on basis of Feature values. As an example, let us follow the scoring of a Feature (e.g. 'Average Density [ato.low.vine]').

The first step is to extract this Feature from the real wells. With these Feature values, geostatistical analysis can be done which results in (expectation,standard deviation) pairs at each position in the survey. Then, the same Feature is extracted from a simulated well group. Each of the pseusdo-wells or 1-D models will so generate a value which can be compared at each position with the (expectation,standard deviation) at that position. This will result, at each position, in a Score for each model and a Score significance (because the significance is determined by the geostatistics only, the significance is the same for all models). The above process is called Score creation.

We could investigate these scores by themselves, but in many cases we will want to combine all the constraints on the models into a 'Final Score'. In this process the score and their significances are used to create a final score. An additional step would be to combine the geostatistical score also with a seismic score and correlation.

As a last step, we can use any score (geostatistical, sesimic but most likely a Final Score) to estimate the most probable value of a Feature at any position. If we extract a Feature for each model in the set of models, we can so create an expectation and a standard deviation for the Feature at each position. The values will be weighted on the score of the model at that position to get the expectation. The standard deviation is also weighted and therefore reflects mainly the spread in the values of the highest scoring models.

Each of the above results is useful in its own right. All steps should be carefully checked. For example, if some of the models score very low everywhere, they may be the result of an incorrect simulation of wells. Or if there is no prevalent value for the high scoring models, there may be no information in any of the constraints for this Feature.

7.3 Scoring geostatistical data

At a certain position, we have for a certain Feature, a pair (Expectation,Standard deviation). Each model will get a score depending on the model's value for this Feature: v. If standard deviation s is non-zero the score of value v on (e,s) will be defined using the normalized deviation:

$d_n = | (e-v) / s |$

The d_n cannot be used as Score for two reasons:

1. A higher sigma_n means a lower probability
2. d_n does not have a fixed maximum

There is an infinite number of transformations that will transform d_n into a Score. We have selected functions that are parameterizable with only one parameter. This parameter is the Half-score-distance (hsd), i.e. the value of the d_n for which the score is 0.5 . Currently supported are:

- **Linear:**

  Score = 1 - 0.5 * d_n / hsd (d_n < 2 * hsd)

  Score = 0 (d_n >= 2 * hsd)
- **Quadratic:**

  Score = 1 - (0.5 * d_n * d_n / hsd * hsd) (d_n < sqrt(2) * hsd))

  Score = 0 (d_n >= sqrt(2) * hsd))
- **Exponential:**

  Score = exp(-ln2 * d_n / hsd)
- **Gaussian:**

Score = exp(-ln2 * d_n * d_n / (hsd * hsd))

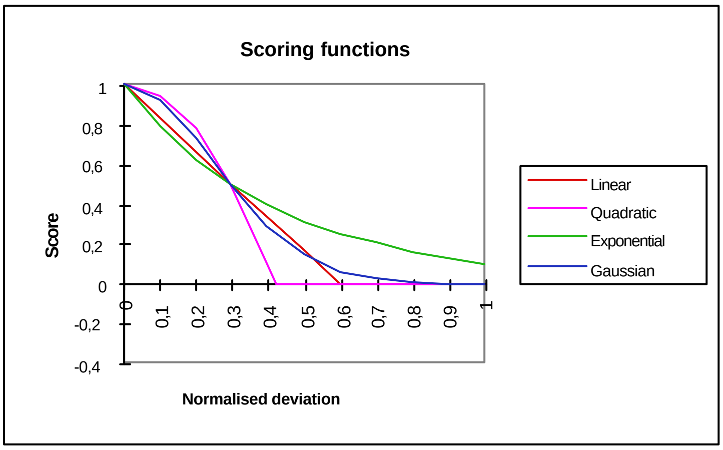


Figure 7.1: *Scoring functions for 'Half score relative deviation' equal to 0.3.*

8 GDI-PRESTACK

8.1 General

GDI-PRESTACK generates prestack gathers, stacked responses and AVO attributes. The package is used in combination with GDI's pseudo-well simulator for:

- unsupervised AVO inversion: matching AVO models to the seismic character using a UVQ neural network;
- supervised AVO inversion: inversion of offset stack data to rock properties using a MLP or RBF neural network trained on modelled data; and
- AVO scenario modelling: modelling the AVO response of simulated wells (generated in the GDI well simulation module).

Fig. 8.1 is a generalised workflow showing the required input, processing and generated output.

8.2 Forward modelling

Synthetic seismograms are generated according to wave propagation through a horizontally layered elastic isotropic earth model. The resulting CMP gather contains:

- *P*-wave reflections;
- no conversions to *S*-waves;
- no multiples; and
- no interface waves e.g. head waves or ground roll.

The algorithm is based on the equations of Zoeppritz (e.g. Aki & Richards, 1980). There are no restrictions to angles below the critical angle. Transmission effects are calculated for each interface separately rather than according to just the ray parameter for the lowermost reflector, as is the case in some other commercial applications.

Ray tracing is used to calculate the travel-times and incidence angles of reflected phases. Plane wave theory is used to calculate the amplitude and phase of these arrivals. The algorithm does not properly calculate the spreading behaviour away from a source point, which is accounted for by ray-based methods such as Gaussian beams, WKBJ and Maslov theory (Aki & Richards 1980).

The ray parameter (horizontal slowness) for a desired offset is determined by shooting rays so that the difference between the desired and calculated offsets is treated as a root (set equal to zero). The numerical method chosen to achieve this is the Van Wijngaarden-Dekker-Brent method taken from Numerical Recipes (Press et al 1992).

A discrete time series with a unit spike at the reflection arrival time does not retain the precision of the ray-traced time: the temporal position of the spike is truncated to the nearest sample point. The problem is compounded when there is tuning (the delay time between arrivals from adjacent reflectors is less than the dominant period). In order to avoid this problem, the equivalent unit spike signal is constructed directly in the frequency domain. The amplitude spectrum has a constant unit amplitude. The phase spectrum is a straight line passing through the origin whose gradient is the time-lag of the spike. Therefore, the precision of the ray-traced time is completely preserved.

In the frequency domain the impulse response (the frequency domain description of the unit spike signal) is multiplied by the appropriate product of the plane-wave (frequency-independent) reflection and transmission coefficients encountered on the ray-path (Aki & Richards 1980):

- *P*-wave reflection coefficient at the reflector;
- product of *P*-wave transmission coefficients on the downward leg; and
- product of *P*-wave transmission coefficients on the upward leg.

The resulting spectrum is produced for each reflector, at a particular offset, and summed.

In the next step the frequency domain representation of the source signal (i.e. the wavelet) is multiplied with the spectrum. The result is then transformed back into the time domain using an inverse FFT to arrive at a CMP gather that needs to be NMO corrected before stacking.

NMO correction is applied after convolution of the reflectivity series with the source wavelet as this is what happens in reality. In processing the NMO correction is estimated from Dix equation, and is a first order approximation. However, in GDI-

PRESTACK the NMO correction is known precisely from ray-tracing. The NMO correction is required not just at the reflection arrival times but at every sample point. The time shifts are therefore interpolated to every sample point using a 4-point, 3d. order La Grange polynomial interpolation algorithm. The NMO correction is applied at every sample. This results in a non-uniform sample interval as the NMO correction changes with time (becoming larger with time if there are velocity increases with depth). The NMO corrected time series is therefore resampled to the original sampling rate by the same 3d. order polynomial interpolation algorithm.

The NMO correction causes a distortion of the wavelet, which increases with offset and depth. Specifying a stretch mute percentage, which means that if the sample interval is stretched more than a specified percentage the signal is muted out, can control this stretching.

Where there is a velocity decrease with depth the travel-time for a shallow reflection can exceed that for a deep reflection beyond a certain offset. This phenomenon is called cross-over and causes a problem for applying NMO correction. In GDI-PRESTACK there is an option to prevent cross-over by always applying the smallest NMO correction where events cross-over. (An alternative approach would be to mute the data beyond this offset.)

The software offers a plane-wave option, which means that the NMO correction is applied before the reflectivity spectrum is multiplied with the source wavelet in the frequency domain. This means that there is no NMO stretch. The plane wave synthetic is completely unrealistic, however, it is instructive as any amplitude variation with offset is entirely due to the behaviour of the reflection and transmission coefficients with offset.

Source and receiver array effects can be included at source and receiver locations separately, or at both locations simultaneously. The calculation simply requires the number of array elements the sensor spacing, and is performed in the frequency domain.

An approximation to the effect of geometrical spreading may be applied to the synthetic seismograms according to either 1/(r) or $1/(V_{\mathrm{int}}^{2}\, t)$ (Sheriff & Geldart 1995).

The user may apply any number of linear mutes to the data. Any number of stacks, each being over a specified offset range, may be produced. The stacked data are normalized by the number of contributing samples.

The AVO gradient and intercept attribute stacks may be calculated (Sheriff & Geldart 1995). The incidence angle is required and may be used directly from the ray tracing or calculated as it would be from real data by using the ratio of the RMS velocity to the sample time and the local interval velocity.

8.3 Acknowledgement

GDI-PRESTACK was developed in co-operation with BG Technology. We are especially grateful to Steve Peel and Tim Pointer for their involvement in defining the functional specification, prototyping and documenting the work.

8.4 AVO references

Aki, K., and Richards, P., 1980. Quantitative seismology theory and methods. W.H. Freeman Company, New York.

Allen, J.L. and Peddy, C.P., 1993. Amplitude Variation with Offset: Gulf Coast Case Studies. Geophysical Developments Series, Vol. 4. Society of Exploration Geophysicists.

Castagna, J.P. and Backus, M.M., 1993. Offset-Dependent Reflectivity - Theory and Practice of AVO Analysis. Investigations in Geophysics No. 8. Society of Exploration Geophysicists.

Sherrif, R.E. and Geldart, L.P., 1995. Exploration Seismology. Cambridge University Press, New York.

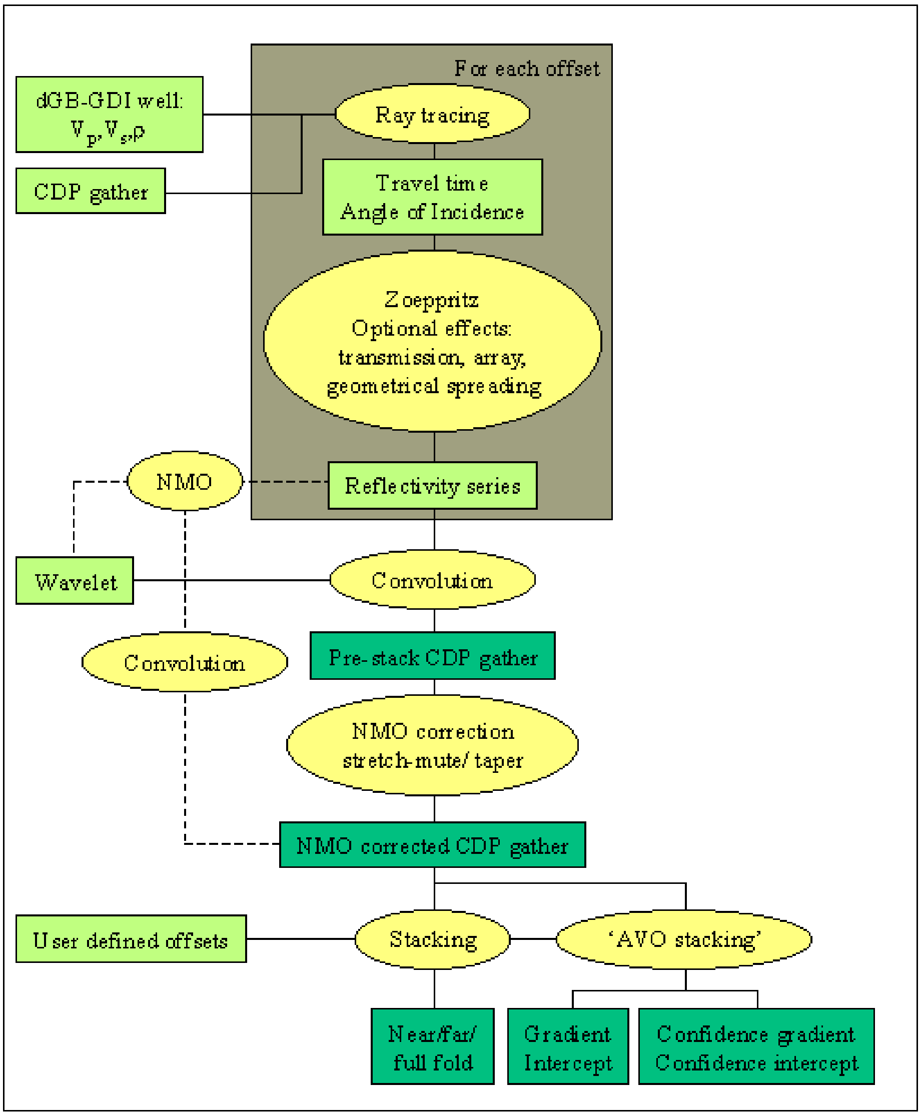


Figure 8.1: *AVO flow diagram.*