



#### **DISCLAIMER**

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government.

This report has been reproduced directly from the best available copy.

Application of Artificial Intelligence to Reservoir Characterization:  
An Interdisciplinary Approach

By  
B.G. Kelkar  
R.F. Gamble  
D.R. Kerr  
L.G. Thompson  
S. Sheno

January 2000

Work Performed Under Contract No. DE-AC22-93BC14894

Prepared for  
U.S. Department of Energy  
Assistant Secretary for Fossil Energy

Robert E. Lemmon, Program Manager  
National Petroleum Technology Office  
P.O. Box 3628  
Tulsa, OK 74101

Prepared by  
The University of Tulsa  
600 S. College Avenue  
Tulsa, OK 74104

## Table of Contents

List of Figures .....	v
List of Tables .....	xi
Executive Summary .....	xiii
Abstract .....	xv
1. Introduction .....	1
2. Decomposition of System .....	2
3. The Geological System .....	3
4. Cosimulation of Lithofacies and Petrophysical Properties .....	18
5. Reservoir Characterization Using Simulated Annealing with Dynamic Constraints.....	70
6. Well Model Identification System .....	100
References.....	107
Nomenclature.....	113
Appendix A Visualization System .....	115
Appendix B COSIM User's Guide.....	127
Appendix C Program Manual for Integration of Dynamic Data into Reservoir Description ....	175
Appendix References .....	186

## List of Figures

Figure 1: Overall System Components.....	3
Figure 2: Process for Determining Marker Beds.....	6
Figure 3: Sum-Difference Equation .....	8
Figure 4: Determining Cuts .....	10
Figure 5: Digital Filtering.....	10
Figure 6: Filtering parameters .....	12
Figure 7: Sample Segmentation Results.....	13
Figure 8: Explanation of the cut elimination rule.....	14
Figure 9: Results obtained with and without the cut elimination rule.....	15
Figure 10: Results obtained from an expert geologist.....	15
Figure 11: Schematic diagram of two stage approach.....	24
Figure 12: Typical horizontal variogram calculated using point to point correlation of well data.....	25
Figure 13: Horizontal variogram of rock type based on the thickness at each well.....	26
Figure 14: Common back transform procedure from Gaussian space into the original space for porosity and permeability .....	29
Figure 15: Category determination in sequential indicator simulation .....	31
Figure 16: Schematic diagram of the truncated Gaussian method in assigning facies at unsampled location for a system with 4 facies. In this example, Facies 2 is assigned .....	32
Figure 17: Schematic diagram of the co-simulation technique .....	35
Figure 18: Conditional distribution procedure for porosity .....	38
Figure 19: Example of the use of conditional distribution technique in the simulation.....	39
Figure 20: Conditional distribution technique for permeability.....	41
Figure 21: Illustration of aliasing effect in data sampling.....	44

Figure 22: Detail data configuration and exhaustive variogram .....	46
Figure 23: Well based data configuration and point-to-point variogram .....	46
Figure 24: Data average configuration and the average variogram .....	47
Figure 25: Comparison between exhaustive, point to point, and average variograms for 100 (10x10) well configuration.....	47
Figure 26: Comparison between exhaustive, point to point, and average variograms for 36 (6x6) well configuration.....	48
Figure 27: Comparison between exhaustive, point to point, and average variograms for 144 (12x12) well configuration.....	48
Figure 28: Comparison of indicator variogram calculated using traditional and modified technique.....	52
Figure 29: Program Structure .....	54
Figure 30: Geologist interpretation of facies distribution in the vicinity of Track-7 .....	61
Figure 31: Multiple realization of facies distribution of DGI A – Track 7 Unit.....	62
Figure 32: Porosity and permeability distribution of DGI A for Track-7 Unit from one of the realization.....	63
Figure 33: Computation time comparison for different grid block configurations and different number of realizations .....	64
Figure 34: North-South Cross Section of DGI distribution of the Shelf Unit data .....	65
Figure 35: North-South Cross Section of porosity distribution of the Shelf Unit data .....	65
Figure 36: North-South Cross Section of permeability distribution of the Shelf Unit Data ..	66
Figure 37: 3D view of rock type distribution .....	67
Figure 38: 3D view of porosity distribution.....	68
Figure 39: 3D view of permeability distribution.....	69
Figure 40: Comparison of <i>ECLIPSE</i> and LTFD Simulation Results.....	83
Figure 41: Comparison of Conventional & Modified Geometric Averaging Upscaling Results.....	86

Figure 42: Summarized Comparison of Conventional & Modified Geometric Averaging Upscaling Results.....	87
Figure 43: Modified geometric averaging upscaling performance related to the fine scale permeability contrast at the wellblock .....	88
Figure 44: (a) truth case (b) modified sa run results (c) variogram-only sa run results for dataset #1 .....	91
Figure 45: Pressure error comparisons between the modified sa approach and the variogram-only sa approach for dataset #1 .....	93
Figure 46: Comparisons of maximum absolute relative errors from variogram-only objective function and the composite objective function using the modified geometric averaging upscaling approach for dataset #1.....	93
Figure 47: Results from sensitivity tests on variogram component .....	95
Figure 48: Results from sensitivity test on well locations.....	96
Figure 49: Representative Symbols.....	102
Figure 50: Possible shapes and models .....	103
Figure 51: Hierachy diagram.....	105
Figure 52: Possible results.....	106
Figure 53: Main menu of COSIM .....	128
Figure 54: Example of GEOEAS Format.....	130
Figure 55: Pre-Simulation sub-menus.....	131
Figure 56: Indicator File Dialog .....	132
Figure 57: Output file from Indicator File menu.....	133
Figure 58: Normal Transform dialog .....	134
Figure 59: Global PDF dialog for normal transform of discrete variable .....	134
Figure 60: Vertical Prop. Curve Dialog .....	135
Figure 61: Vertical Proportion Curve File Format.....	136

Figure 62: Data Variogram and Model Variogram Sub-Menus.....	137
Figure 63: Example of Isotropic Variogram calculated using default lag parameters .....	140
Figure 64: Floating menu to modify variogram .....	141
Figure 65: Change Data-Variogram Parameter Dialog .....	141
Figure 66: Example of Anisotropic Variogram using default azimuth parameters.....	142
Figure 67: Program lay-out during variogram's modeling mode.....	143
Figure 68: Example of modeled variogram.....	145
Figure 69: Zonal anisotropy dialog .....	146
Figure 70: Simulation Parameter Dialog.....	148
Figure 71: Example of COSIM data format that contains soft data .....	149
Figure 72: Simulation Parameter Dialog for Grid Block Group .....	150
Figure 73: Simulation Parameter Dialog for Proportion Curves Group.....	151
Figure 74: Simulation Parameter Dialog for Kriging Group.....	152
Figure 75: Example of permeability correlation file .....	153
Figure 76: Simulation Parameter Dialog for Simulation Group .....	154
Figure 77: Simulation Parameter Dialog for Variogram Group.....	155
Figure 78: Variogram Model Dialog .....	155
Figure 79: The structure detail of variogram model.....	156
Figure 80: Beginning of simulation dialog.....	157
Figure 81: End of simulation dialog.....	158
Figure 82: Post-Simulation Sub-menus.....	158
Figure 83: 2D-Cross Section Window Dialog.....	159
Figure 84: Example of on-screen 2D-Cross Section .....	160
Figure 85: ASCII Format of the Output Result.....	161

Figure 86: 3D Spyglass-Slicer File Dialog.....	162
Figure 87: Statistics of the simulation result.....	162
Figure 88: Well map and facies distribution of the case study data.....	164
Figure 89: Vertical Proportion Curve for “cosim.dat”.....	165
Figure 90: Global PDF for “cosim.dat”.....	165
Figure 91: Parameters to create variogram for facies 1.....	168
Figure 92: Isotropic variogram for facies 1.....	169
Figure 93: Anisotropic variogram for facies 1 using default azimuth directions.....	169
Figure 94: Variogram model for facies 1.....	170
Figure 95: Zonal anisotropy dialog for variogram of facies 1.....	170
Figure 96: Variogram model for: (a) facies 2 and (b) facies 3.....	171
Figure 97: Gaussian variograms for (a) Facies and (b) Porosity.....	171
Figure 98: the 2D cross section of the simulation result for 4 different realizations.....	173
Figure 99: The 2D cross section of porosity and permeability distributions for one of the realizations.....	174
Figure 100: Modified SA algorithm for reservoir characterization.....	188



## List of Tables

Table 1:	Results of sum-difference equation on sample wells.....	8
Table 2:	Well Log Correlations with Well 7-87 .....	17
Table 3:	Well Log Correlations for Well 11-86.....	18
Table 4:	Facies at well location and its indicator variable .....	50
Table 5:	C++ Class Summary Used in the Program .....	56
Table 6:	Global PDF comparison between simulation and data for 5 realizations.....	61
Table 7:	Fine scale wellblock permeability in high contrast to surrounding permeability values .....	88
Table 8:	Fine scale wellblock permeability in low contrast to surrounding permeability values .....	89
Table 9:	Variogram's tool bar definition .....	144
Table 10:	Variogram requirement for simulation run.....	147



## **Executive Summary**

As the reservoir description technology matures, it is becoming increasingly evident that integration of knowledge from various disciplines is extremely critical for a representative reservoir description. The three primary disciplines, which are involved in constructing the reservoir description, are geology, engineering and geophysics.

The present work concentrates on integrating geological and engineering information in a consistent manner to construct a reservoir description. We use Artificial Intelligence (AI) based approach to articulate geological knowledge, and geostatistical based methodology to integrate engineering data.

Typically, the information available about the reservoir can be divided into two broad categories – static data and dynamic data. Static data include well core and log data and seismic data, and dynamic data include well test data and production data. The integration of dynamic data is much more challenging because it requires solving an inverse problem. The present work integrates both the static and dynamic information in constructing the reservoir description.

The overall report is divided into six sections. The first section, Introduction, lays out the framework for the need of integration and why AI based methodology provide us with a solution. The second section discusses the overall system and the individual components that form the system. It also explains the interrelationships among various components.

Section 3 discusses the methodology used in developing geological framework for the reservoir description. It is divided into three subsections – identification of marker beds, identification of sand types, and correlations between the wells. The identification of marker beds is done by the use of sum-difference equation. A log from a type well is compared with the adjacent wells. Knowing that marker bed is indicated by a strong contrast in gamma ray values, the sum-difference contrast is used to identify the marker bed. This method has been proved to be successful in more than 90% of the times it has been tested.

Once a marker bed is identified, neural network is used to identify the sand bodies. The sand shapes are divided into five categories – bell, funnel, blocky, symmetrical and linear. The cuts between the sand bodies are defined using well segmentation rule, and the sand body types are identified using neural networks. The success of neural network is within the uncertainties experienced by geologists.

The generated sand bodies are correlated between the two wells using several soft and hard parameters. These parameters include: position of the zone with respect to length of the well logs, the distance of the zone from the marker bed, thickness of the zone and the sand type within a zone. By using a weighted criterion, a matrix of values is developed, and used to identify a correlation. The method has been validated using field data sets. As in the case of sand identification, the correlations generated fall within the realm of uncertainties experienced by an expert geologist.

Section 4 of the report describes the technique that takes the information from the geological module and populates the geological facies and petrophysical properties at inter well locations. The procedure used for generating the facies and petrophysical properties is the conditional co-simulation technique. The conditional co-simulation method allows the multiple realization of various attributes with locally consistent relationships. For example, if geological facies “shale” is assigned at a particular grid block, then the porosity and permeability at that grid block should be zero. By generating multiple descriptions, we are able to quantify uncertainties in the descriptions, as well as rank various realizations according to different criteria.

Section 5 of the report explains the technique that uses the petrophysical properties as an input to flow simulate the reservoir description. This step is necessary to incorporate the production data. Through the use of simulated annealing, the petrophysical properties are perturbed till a match between the simulated data and the observed performance is obtained. To make the process computationally efficient, the flow simulation is conducted on a coarser scale using an appropriate upscaling technique. At present, the simulation is restricted to single-phase flow, and requires significant computational effort for a practical use for small operators.

Section 6 explains the evaluation of well testing data. Based on the shape of pressure-time data, an appropriate reservoir model is selected for evaluation. Using a non-linear regression, the match between the model and the observed data are obtained. This should provide us with estimation of reservoir parameters such as permeability and the skin factor. At present, the procedure is a stand-alone procedure, and is used as a qualitative tool to evaluate the geometric shape of the sand bodies.

The details of the computer programs are explained in the Appendix. Appendix A provides the overall structure of the computer program that conducts the geological evaluation and modeling. Appendix B provides the details about the co-simulation program, and the Appendix C provides the details about the inverse-modeling program that integrates the production data.

To summarize the achievements of the present work, we have been successful in integrating static data in constructing reservoir description that provides us with petrophysical properties that are consistent with the underlying geological structure. The integration of dynamic data has also been accomplished; however, the computational effort required to integrate the production data may not make it very practical. With increasing speed of computers, and better inverse modeling techniques, this process can be improved in the future.



## **Abstract**

The primary goal of the project is to develop a user-friendly computer program to integrate geological and engineering information using Artificial Intelligence (AI) methodology. The project is restricted to fluviially dominated deltaic environments.

The static information used in constructing the reservoir description includes well core and log data. Using the well core and the log data, the program identifies the marker beds, and the type of sand facies, and in turn, develops correlations between wells. Using the correlations and sand facies, the program is able to generate multiple realizations of sand facies and petrophysical properties at interwell locations using geostatistical techniques.

The generated petrophysical properties are used as input in the next step where the production data are honored. By adjusting the petrophysical properties, the match between the simulated and the observed production rates is obtained.

Although all the components within the overall system are functioning, the integration of dynamic data may not be practical due to the single-phase flow limitations and the computationally intensive algorithms. The future work needs to concentrate on making the dynamic data integration computationally efficient.



## 1. Introduction

The basis of this research is to apply novel techniques from Artificial Intelligence and Expert Systems in capturing, integrating and articulating key knowledge from geology, geostatistics, and petroleum engineering to develop accurate descriptions of petroleum reservoirs. The ultimate goal is to design and implement a single powerful expert system for use by small producers and independents to efficiently exploit reservoirs.

The main challenge of the proposed research is to automate the generation of detailed reservoir descriptions honoring all the available "soft" and "hard" data that ranges from qualitative and semi-quantitative geological interpretations to numeric data obtained from cores, well tests, well logs and production statistics. In this sense, the proposed research project is truly multi-disciplinary. It involves significant amounts of information exchange between researchers in geology, geostatistics, and petroleum engineering. Computer science (and artificial intelligence) provides the means to effectively acquire, integrate and automate the key expertise in the various disciplines in a reservoir characterization expert system. Additional challenges include the verification and validation of the expert system, since much of the interpretation of the experts is based on extended experience in reservoir characterization and these experts often have differing opinions.

The overall project plan to design the system to create integrated reservoir descriptions begins by initially developing an AI-based methodology for producing large-scale reservoir descriptions generated interactively from geology and well test data. Parallel to this task is a second task that develops an AI-based methodology that uses facies-biased information to generate small-scale descriptions of reservoir properties such as permeability and porosity. The third task involves consolidation and integration of the large-scale and small-scale methodologies to produce reservoir descriptions honoring all the available data. The final task is technology transfer. Using this plan, we have carefully allocated and sequenced the activities involved in each of the tasks to promote concurrent progress towards the research objectives. Moreover, the project duties are divided among the faculty member participants. Graduate students work in teams with faculty members.

The results of the integration are not merely limited to obtaining better characterizations of individual reservoirs. They have the potential to significantly impact and advance the discipline of reservoir characterization itself.

## **2. Decomposition of System**

We have decomposed the overall system development into smaller component parts to allow us to focus on the expert knowledge required for that component. In addition, the decomposition facilitates the implementation of the system and its validation and verification. The three component systems are representative of how each of the experts in geology, geostatistics, and engineering characterizes the reservoir. Figure 1 describes our overall system model. The concurrent development of these component systems fits into the development of the large and small-scale aspects of the system.

The geostatistical system includes the description of petrophysical properties that are consistent with underlying geology. The geological system computes well-log correlations by determining the placement of the marker bed, the discrete genetic intervals, and the facies information for each gamma-ray log. The well test interpretation system is complete as a stand-alone system. A simulation system provides a tool to generate consistent petrophysical properties with the production data. A graphical user-interface allows the user to direct the processing of the system through the various components.

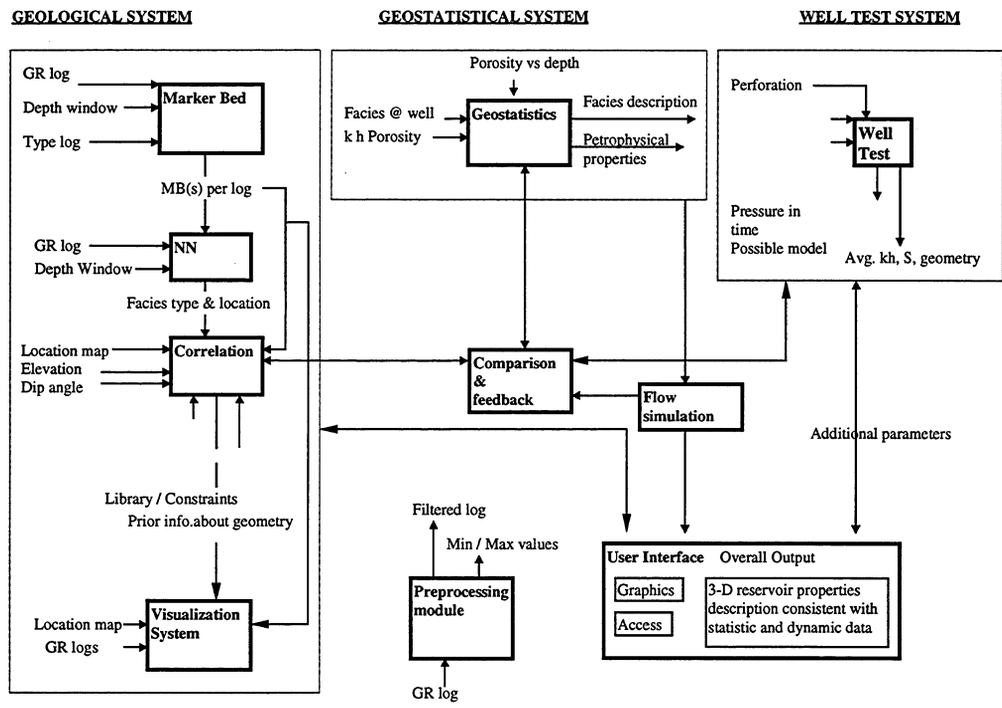


Figure 1: Overall system components

### 3. The Geological System

There are four major components of the geological system that result in the production of well-log correlation across discrete genetic intervals: (1) Marker Bed Determination Component, (2) Facies Determination Components, (3) Correlation Component, and (4) Visualization System. We describe each one in turn in the following section.

#### 3.1 Marker Bed Detection Component

In order to identify log facies and correlate stratigraphic units within wells across a reservoir, a reference point must be available that identifies the interval of observation for such analysis to take place. Marker bed identification provides the beginning and ending interval depths for this analysis.

Identification of marker beds is critical to the accurate correlation of reservoir architectural elements. In meandering fluvial and fluvial-dominated deltaic depositional systems, the stratigraphic elevation to (relative to a marker bed) the top of a facies is a key rule in the correlation of architectural elements.

Marker beds have two general characteristics: (1) Marker beds are comprised of distinctive lithologies. Such lithologies, in turn, have distinctive wireline log characteristics. A part of being distinctive is that such beds are usually thin (<5 ft.). Marker beds commonly are associated with other lithologies in the overlying and underlying strata. This tends to result in wireline log characteristics that vary in a subtle manner from well to well. (2) Marker beds are correlative at least over the geographic scale of most oil fields, and very commonly across entire sedimentary basins. Geologists can readily adapt to or allow for the subtle variations in marker bed characteristics when attempting to correlate from well to well.

Although the general characteristics of marker beds are simple and straightforward, the automated identification is not so simple. Over the course of this reporting term, we have tried several techniques, which have had varying degrees of success.

The use of heuristic rules covering the magnitude of combined log values was very successful in the identification of a known marker bed, but it also identified several other candidates for marker beds; thus, it was not very discriminating. The cross correlation approach was fairly successful in the identification of a known marker bed, but in many cases it selected other intervals as being better candidates than the known marker bed. Cross correlation is very sensitive to mutual changes in the slope of the data strings being compared. The approaches to this point attempted to identify the marker bed starting with any well and continuing on through well to well comparisons. The project team came to the understanding that such an approach was unrealistic. The present approach uses a type log, a common practice particularly in areas of mature development.

In areas of complex stratigraphy or covering large geographic areas, subsidiary type logs can also be established. A type log is a wireline log from a strategically selected well. In a type log, the depth ranges of major stratigraphic units are identified. This would include any marker beds that are in that are used in that geographic area. Other information that might affect wireline log characteristics

could also accompany the type log information. The type of log information, as well as other information/data shared across modules, is to be stored in the database module, which has yet to be developed. The user will be responsible for inputting the necessary information.

Given a type log, the characteristics of a particular marker bed can be evaluated. In many cases, marker beds are characteristics by large excursions in gamma ray or other log values. For example, the carbonaceous shale Inola marker bed used in Glenn Pool Field of eastern Oklahoma is noted for its extreme high gamma ray emission (about 5 times the natural gamma ray emissions of other mudstones, calcareous mudstones, and finely crystalline limestones at a similar stratigraphic level). In other cases, the log shape may be more distinctive, or some combination of the shape recorded on more than one log. The major contribution of a type log is that it provides a starting point or known case from which characteristics can be extracted and used to evaluate unknown cases.

### ***3.1.1 Sum Difference Equation***

The sum difference algorithm has proved, thus far, to be quite successful in the identification of marker beds characterized by strong wireline log excursions. The example tested is the Inola marker bed from Glenn Pool field of eastern Oklahoma.

Because the sum difference equation is sensitive to the magnitude of variation in the data strings being compared, log data must be scaled and type logs for open- and cased-holes should be considered separately. It is common for logs to vary from well to well, and even within a well for different log runs. In cased-holes, the presence of casing and annulus cement can alter the log characteristics from that found in open holes. Therefore, if cased-hole logs are to be used, then a type log, or at least a known marker bed segment, must be provided by the user. Thus each log needs to be first scaled in a similar fashion (as in Figure 2).

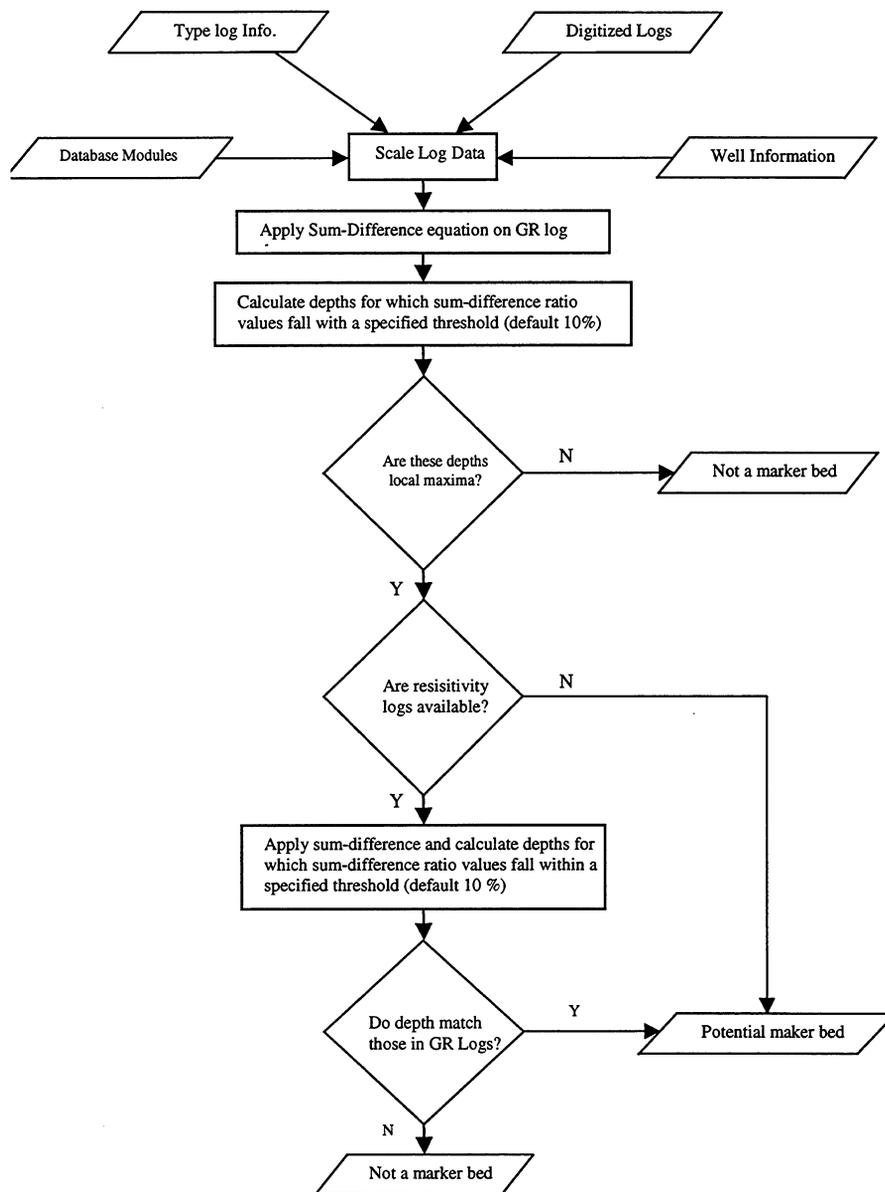


Figure 2: Process for Determining Marker Beds

The sum difference equation is next applied to gamma ray and resistivity logs separately (see Figure 2). The sum-difference equation evaluates the squared difference in log values between the type log interval of a known marker bed and the same log in offset wells. The difference is summed across the depth range of the marker bed interval, and scaled to the sum of the values in the marker bed interval (Figure 3). If the match is perfect, then the sum difference ratio is equal to zero. The stronger the mismatch, the greater the absolute value of the sum difference ratio.

The marker bed interval is progressively moved along the log string under evaluation with the sum difference ratio calculated at each comparison position or match position. Thus, producing sum difference ratio values over an interval of match positions.

Ideally, the sum difference ratio is at or near zero at only one match position, thus, providing an unambiguous identification of a marker bed. This is, however, not always the case. The next step is to rank the sum difference ratio values falling close to zero that meets some user-defined tolerance limit (Figure 2). With the tolerance the module can rank those sum difference ratio and their match positions.

An artifact of the sum difference algorithm is the shifting of the position of peaks. Thus, the module must return to the original log to identify the local maximum value corresponding to those match positions falling within the tolerance limit. If the selected depth does not correspond to a local maximum on gamma ray log, that selection is rejected as a marker bed. If the selected depth does not correspond to a local maximum then it is considered a potential marker bed (Figure 2). Potential marker beds will be further evaluated within the marker bed module using techniques to be developed in the next reporting period.

Of the 13 sample well logs first evaluated, the sum difference equation clearly ranked the known marker bed (Inola Marker) first with a success rate of 62%. This rate was improved to 90% over the larger test case discussed later in the correlation component when the marker bed at highest depth was given a weighted rank.

$$\frac{\sum_{i=0}^n (L_i - TL_i)}{\sum_{i=0}^n (TL_i)}$$

where  
L = gamma log for well under study  
TL =type log for field under study  
n =number of feet of marker bed in type log

Figure 3: Sum-Difference Equation

Well	Log Type Used	Identified Marker Bed?	Rank	Comments
Self 56	Open-hole	yes	1	
TRB 48	Open-hole	yes	1	
BG 11-87	Cased-hole	yes	2	Rank 1 within 2 ft. Inola marker
BG 11-86	Cased-hole	yes	2	Rank 1 below Glenn Sand
BG 18-32	Open-hole	yes	1	
BG 18-33	Open-hole	yes	1	
BG 11-85	Cased-hole	yes	1	
P 10	Cased-hole	yes	1	
BG 11-88	Open-hole	yes	1	
Self 78	Open-hole	yes	2	Rank 1 above Inola marker
Self 69	Pen-hole	yes	1	
BG 11-89	Open-hole	no	5	Rank 1-4 below Glenn Sand
P 12	Cased	no		Not in top 5 ranking

Table 1: Results of sum-difference equation on sample wells

### 3.2 Sand Body Identification

In order to analyze well log data, we solve the following two problems sequentially:

- *Well log segmentation problem*
- *Log facies identification problem*

**Well log segmentation.** Given a well log data file the system determines the endpoints, called *cuts*, of every sand body present in the log file. This is needed to divide the well log (gamma ray) into

discrete stratigraphic units. Such segmentation is for log facies identification and well-to-well correlation. A rule-based system is applied to the original data file to determine the cuts or segments. The resulting file is then fed to the neural network to solving the log facies identification problem.

**Log facies identification.** Given a well log data file and the predetermined cuts, the system determines which kind of facie or sand body is between any two cuts. A neural network is used to solve this problem. The input to the network is an intermediate file generated by the rule-based system.

Our neural network was previously trained with expert-classified well logs to recognize the following set of fundamental shapes:

*bell, funnel, blocky, symmetrical, linear*

### **3.2.1 Well Log Segmentation**

Well logs have to be scaled and normalized in order to set a common ground on which the problem can be solved. In consequence every log file is scaled in such a way that:

*maximum gamma ray value maps to 1*

*minimum gamma ray value maps to 0*

As a result of this process all the gamma ray values will be within this range (0-1). This is done before attempting to solve either the log segmentation or the facies identification problems.

Figure 4 shows how these rules are applied to a section of a log:

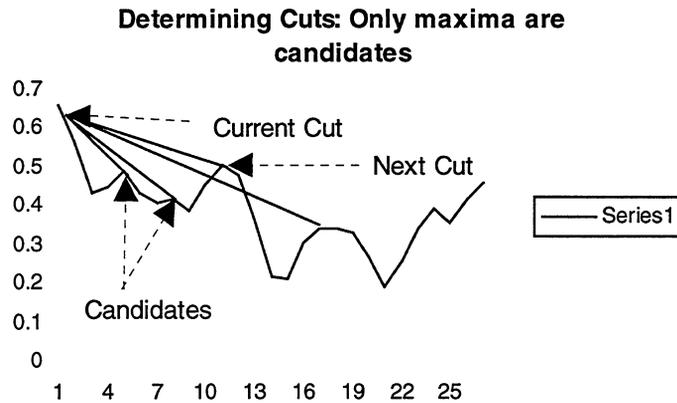


Figure 4: Determining Cuts

The original neural network had problems recognizing cuts between facies. This neural network only recognized about 70 percent of the actual cuts because it used high frequency information incorrectly. It was therefore necessary to create a new module to improve facies recognition. This module uses a low-pass digital filter to eliminate high frequency information.

The digital filter is represented by the block diagram below (Figure 5):

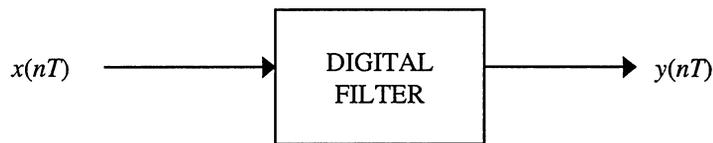


Figure 5: Digital Filtering

where  $x(nT)$  is the unfiltered or excitation data and  $y(nT)$  is the filtered data or the response of the filter. The response is related to the excitation by:

$$y(nT) = Rx(nT)$$

where  $R$  is an operator.

The type of filter used to filter the well log is *time-invariant*, *linear* and *nonrecursive*. *Time-invariant* means that the operator  $R$  does not depend on the time of the application of the excitation. *Linear* means that  $R$  satisfies the following conditions for all possible values of  $\alpha$  and all possible excitations  $x_1(nT)$  and  $x_2(nT)$ :

$$R\alpha x(nT) = \alpha Rx(nT)$$

$$R[x_1(nT) + x_2(nT)] = Rx_1(nT) + Rx_2(nT)$$

*Nonrecursive* means that the response to the filter at instant  $nT$  is of the form

$$y(nT) = f\{\dots, x(nT - 2T), x(nT - T), x(nT), x(nT + T), x(nT + 2T), \dots\}$$

Because the filter is *linear* and *time-invariant*,  $y(nT)$  can be expressed as

$$y(nT) = \sum_{i=-\infty}^{\infty} a_i x(nT - iT)$$

where  $a_i$  terms are constants.

The constants  $a_i$  used to filter the well logs are obtained from the function  $a(i)$  graphed below in Figure 6.

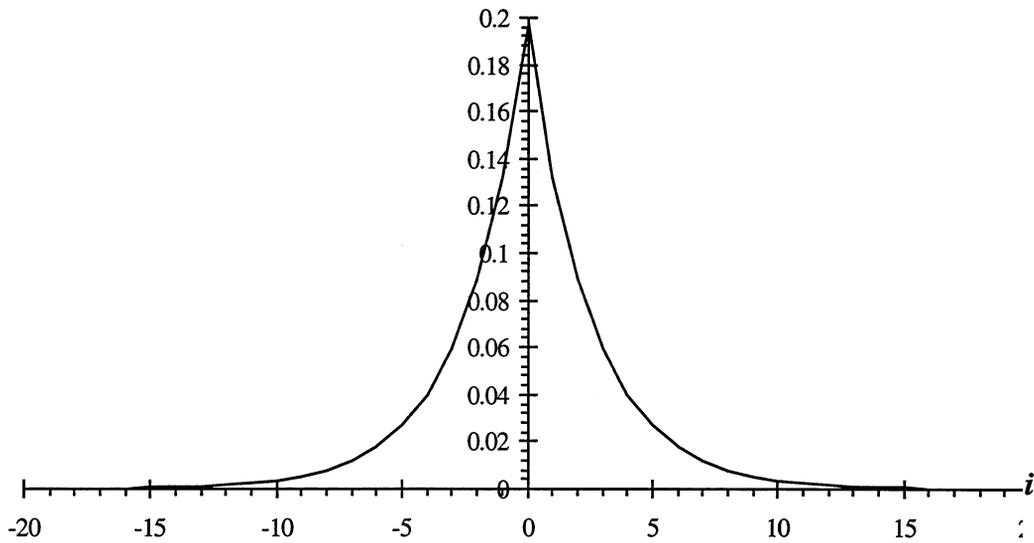


Figure 6: Filter parameters

The sum of all  $a_i$  constants is equal to 1, the maximum  $a_i$  is at  $i = 0$  and the assignment of  $a_i$  values is symmetric respect to the y-axis( $i$ ). Thus, the gain of the filter is 1 and there is no phase delay between the unfiltered and filtered well logs. The result of filtering is shown in Figure 6.

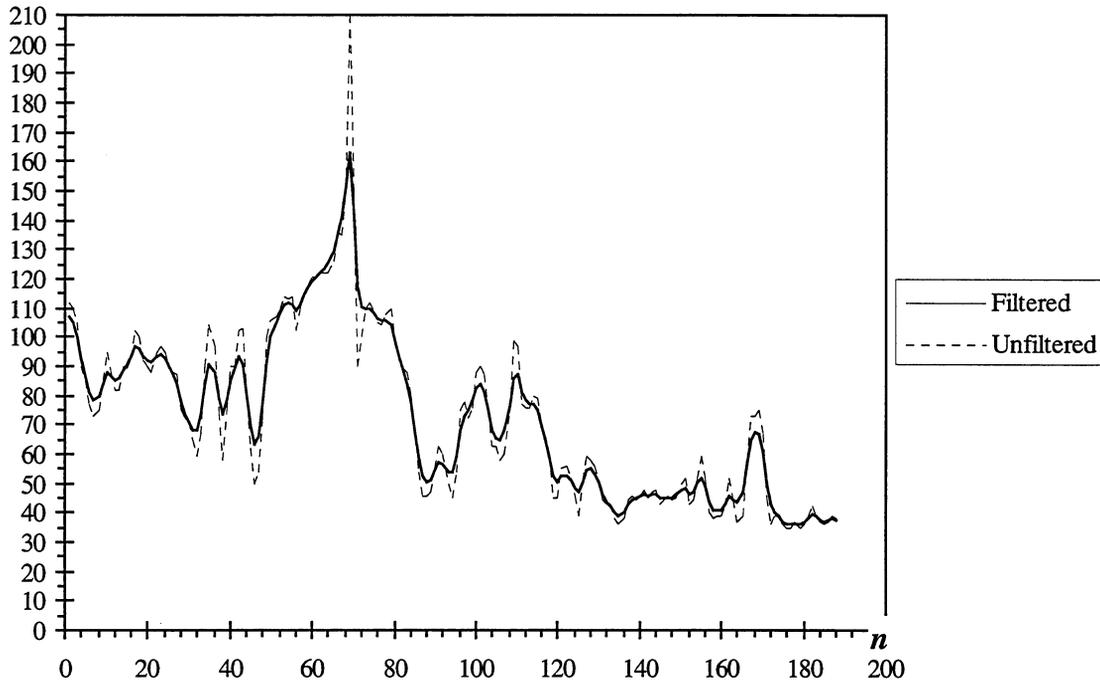


Figure 7: Sample Segmentation Results

We found that using the low-pass filter on the well log data improves its segmentation, but does not improve the facies recognition performance of the neural network. Therefore, only filtered well log data are used for segmentation.

We also found that it is necessary to include a new rule to the well log segmentation technique. This new rule calculates the shape distances between the cuts that are found using the former rules, if the distance is less than a prefixed valued, one of those must be a non-cut and it is eliminated. The actual cut is the one that has the larger gamma ray (Gr.) value. Figure 8 shows a portion of a well log. In this figure it is seen that the distance between two consecutive possible cuts is shorter than the prefixed value. Therefore, one of those possible cuts is not a cut. The cut selected by the rule is the one that has the larger gamma ray (Gr.) value.

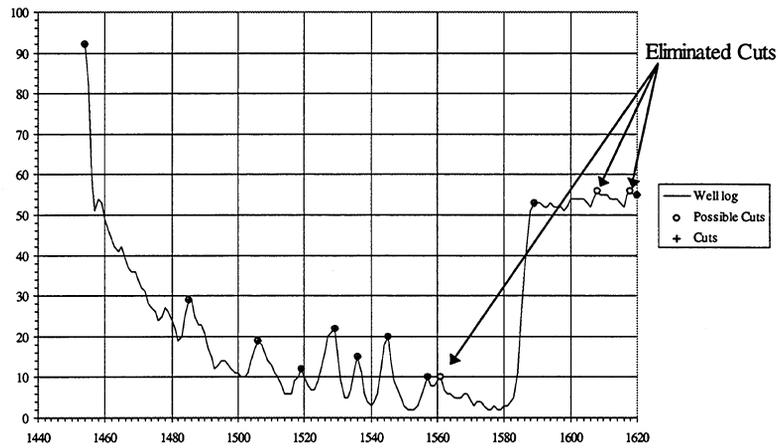


Figure 8: Explanation of the cut elimination rule

Figure 9 shows the result of applying the new rule. It is seen that the new rule eliminates 3 possible cuts and only retains the correct cuts. The eliminated cuts are located at depths of 1561, 1608, and 1618. Also, the eliminated cuts do not coincide with the cuts supplied by an expert geologist (shown in Figure 10).

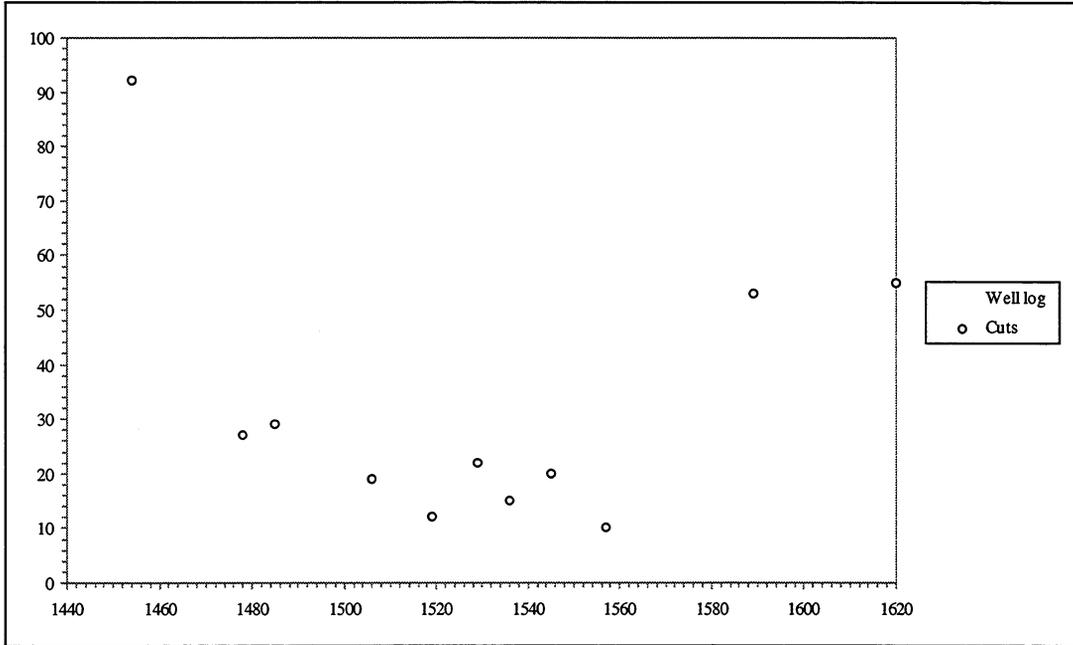


Figure 9: Results obtained with and without the cut elimination rule

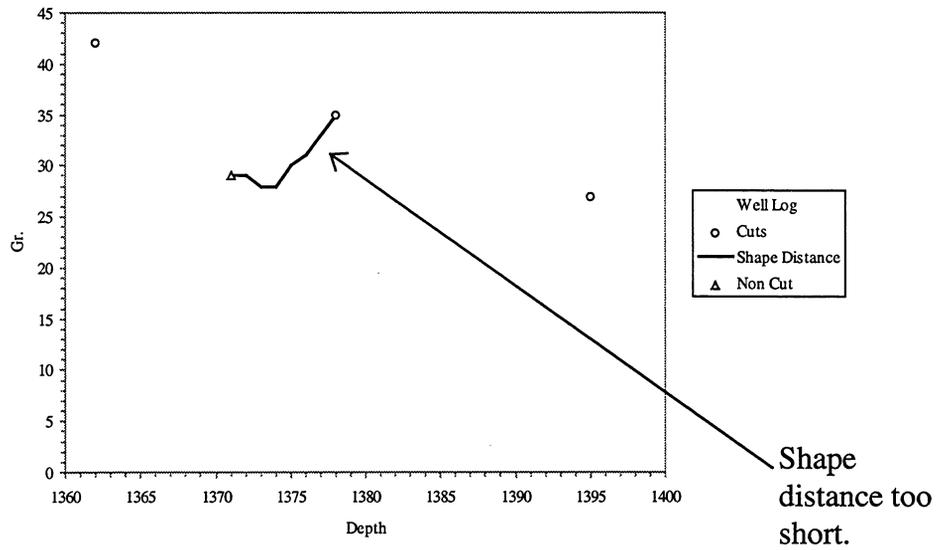


Figure 10: Results obtained from an expert geologist

The low-pass filtering technique and the new segmentation rule explained above improved the performance of well log segmentation from 70 percent to 90 percent. The probability of recognizing facies correctly is correspondingly increased.

### **3.2.2 *Facies Determination Component***

A neural network is used to identify the fundamental shapes of sand bodies. The log facies are normalized and scaled, then discretized and scaled so they can be fed to the network. To scale/normalize individual segments, a facies occurs between two consecutive cuts. All points between two consecutive breaks are scaled so that the maximum value maps to 1 and the minimum value maps to 0. To scale/discretized depth values for each facies, the original data is linearly interpolated at 16 equally spaced depth values and discretized accordingly.

To identify the sand bodies; the neural network was trained with multiple well logs to achieve a 97.2% recall rate or correct classification when compared to the expert geologist.

### **3.3 Correlation Component**

We have obtained sufficient correlation results from the Correlation Component. These results are passed to the geostatistical algorithms for simulation of the reservoir. The input to the Correlation Component consists of all the database information from the reference well and the well to which the reference well is correlated (henceforth, called the *basic well*). The database is built in MicroSoft Access. For the reference wells, the information is input by the user and remains stable. For the basic well, the database includes (1) the depths of the possible marker beds, (2) the depths (referred to as *cuts*) of each possible discrete genetic interval in the well log, and (4) facies information for each discrete genetic interval.

The expert system currently correlates with the closest reference well. The user interface allows additional correlations to be performed by the user dynamically. However, because the distance between wells is a factor considered by the system in the correlation and because neither well may be a stable reference well, the dynamic correlations may not provide a high degree of accuracy. We have compared the results of the system with that of the expert geologist on the cut values and the delineation of discrete genetic intervals with respect to the correlation.

The two tables below present the results of a pairwise correlation between wells in the Glenn Pool field. Two wells (labeled 11-86 and 7-87) are considered as reference wells with the characteristics given by the geologist. Table 2 is for the correlations with well 7-87 and Table 3 is for the correlations with well 11-86. All other wells were passed through the geological system to determine the marker bed, cuts, facies identification, and correlated intervals. The first column of each table has the labels of the wells correlated with the respective reference well. The second column indicates the total number of discrete genetic interval delineations for the correlation with respect to that well log. The third column indicates the number of acceptable delineations. The fourth column indicates the number of unacceptable delineations. The fifth column indicates the number of questionable delineations. The final column indicates how many of the delineations are missing.

In Table 2, the well logs are correlated with well 7-87. This well has 8 discrete genetic interval delineations as indicated by the expert geologist.

<b>Well</b>	<b>Total</b>	<b>Acceptable</b>	<b>Unacceptable</b>	<b>Questionable</b>	<b>Missing</b>
<b>7-109</b>	8	8	0	0	0
<b>7-110</b>	7	5	1	1	0
<b>7-89</b>	8	5	2	0	1
<b>d-9</b>	8	7	0	1	0
<b>e-9</b>	8	6	2	0	0
<b>e-8</b>	7	7	0	0	0
<b>f-8</b>	7	2	3	0	2
<b>6-83</b>	8	7	0	1	0
<b>g-8</b>	8	6	1	0	1
<b>g-85</b>	8	6	1	0	1
<b>7-113</b>	8	7	1	0	0
<b>7-107</b>	8	6	1	1	0
<b>7-99</b>	8	6	1	1	0
<b>7-100</b>	8	6	1	1	0
<b>7-103</b>	8	6	2	0	0
<b>6-85</b>	7	5	0	2	0
<b>6-79</b>	8	8	0	0	0
<b>6-84</b>	8	4	4	0	0
<b>h-85</b>	8	6	0	1	1
<b>7-97</b>	8	2	4	0	2

Table 2: Well log correlations with Well 7-87

Table 3 shows the analysis results for the well correlations with reference well 11-86. Well 11-86 has 8 discrete genetic interval delineations as given by the expert geologist.

<b>Well</b>	<b>Cuts</b>	<b>Unn. Cuts</b>	<b>Mis. Cuts</b>	<b>Intervals</b>	<b>Inc. Ints.</b>
<b>h-11</b>	7	6	1	0	0
<b>h-13x</b>	8	5	3	0	0
<b>h-12</b>	8	8	0	0	0
<b>h-10x</b>	8	7	1	0	0
<b>h-85</b>	9	1	0	7	1
<b>11-82</b>	8	7	1	0	0
<b>11-84</b>	7	7	0	0	0
<b>11-85</b>	7	6	0	1	0
<b>k-10</b>	7	5	1	1	0
<b>m-6</b>	8	5	3	0	0
<b>k-12</b>	8	5	3	0	0
<b>m-12</b>	8	8	0	0	0

Table 3: Well log correlations for Well 11-86

Overall the rate of acceptable discrete genetic intervals correlated with the reference wells is 70%. This figure rises to 74% if the questionable delineations are included as acceptable. The merging that occurs in the reference such that there is one less discrete genetic interval correlated than expected is correct 86% of the time. The missing delineations indicated in the tables are all due to cuts missing in the basic well.

#### **4. Cosimulation of Lithofacies and Petrophysical Properties**

##### **4.1 Summary**

This report presents the new procedure that simultaneously generates consistent distributions of categorical lithofacies and continuous petrophysical properties, i.e., porosity and permeability. The technique used is the conditional simulation method, which is capable of honoring the original distribution of the data and the associated spatial relationship. The simulation of lithofacies is conducted using the combination of indicator simulation and truncated Gaussian simulation techniques whereas the porosity simulation is conducted using the sequential

Gaussian simulation. Two options in generating permeability simulation are provided the sequential Gaussian simulation and the conditional distribution technique.

Previously, to generate petrophysical properties consistent with geological description, investigators have used two step approach where geological description are constructed first, and the petrophysical properties are described in the next step through a filtering process. In contrast, in the approach used in this report, each grid block is visited only once. Using the same search neighborhood, the geological facies is estimated first, followed by porosity and permeability values. The method accounts for correlations among these variables as well as the spatial relationships. This reduces the storage requirements and makes the process computationally efficient.

The method was successfully validated using the field data. Both sandstone and carbonate fields were used to generate the facies/rock type, porosity, and permeability distributions. The simulated geological descriptions matched well with the geologists' interpretation. Further the results were consistent with the observed production performance in terms of quality of rocks and petrophysical properties.

The implementation of this new procedure is done using the C++ language. The implementation includes the user interface program which operates in Windows 95 / NT system. Using this program, the user is able to perform variogram analysis, to prepare the required parameters, to run multiple realizations of the co-simulation, and to view the graphical results, i.e., the cross sections, directly from the screen.

## **4.2 Introduction**

Reservoir characterization or reservoir description is the process of generating reservoir properties, mainly porosity and permeability, by integrating many types of data. An ultimate goal of performing reservoir characterization is to better predict the future performance of the reservoir. Before we reach that goal a journey through various processes must come to pass. The more exhaustive the processes, the more accurate the prediction will be.

One of the most important processes in this journey is the incorporation of geological information.<sup>1-4</sup> The most common data types available for this purpose are in the form of well logs and/or well cores. The translation of these data into the petrophysical properties, i.e., porosity and permeability, at interwell locations that are consistent with the underlying geological description is a critical process. The study presented in this work provides the methodology and implementation to achieve the goal of this particular process. It further validates the method through various field applications. This methodology is based on the geostatistical technique of conditional simulation.

The step-by-step procedure to follow in this process starts with the work of geologist where the isochronal lines across the whole reservoir are determined. This step is followed by the assignment of facies and petrophysical properties at well locations for each isochronal intervals. The result of the process in this step is the hard data for the future processes.

Using the hard data described in the previous paragraph, the spatial analysis in both vertical and horizontal directions of the reservoir attributes, i.e., facies, porosity, and permeability, can be conducted. This step is very critical since it determines the final result. Due to the nature of how the data are distributed, i.e., abundant in the vertical direction but very sparse in the horizontal direction, this step becomes far from a simple task. Practitioners have used approximations to overcome this problem.<sup>5-7</sup>

A new technique in evaluating the horizontal spatial relationship is proposed in this work. The technique uses the average properties of the vertical data to infer the low frequency characteristics of the horizontal data. Additionally, a correction in calculating indicator variogram, which is used to capture the facies' spatial relationship, is provided. Once the spatial relationship of the reservoir attributes has been established, the generation of internally consistent facies and petrophysical properties at grid block level can be done through simulation process.

Common practice in the industry is to perform the conditional simulation of petrophysical properties adapting the two stage approach.<sup>8-11</sup> In the first stage, the geological description is simulated using a conditional simulation technique such as sequential indicator simulation or Gaussian truncated simulation to produce the detailed geological simulation. In the second stage,

petrophysical properties are simulated for each type of the geological facies/unit using a conditional simulation technique such as sequential Gaussian simulation or simulated annealing. The simulated petrophysical properties are then filtered using the generated geological simulation to produce the final description. The drawback of this approach is its inefficiency, since it requires several simulations, and hence intensive computation time.

Additionally, the effort to jointly simulate (or to co-simulate) interdependent attributes such as facies, porosity, and permeability, has been discussed by several authors.<sup>12-14</sup> The techniques used by these authors have produced reasonable results. Common disadvantages of these techniques are the requirement of tedious inference and modeling of covariances and cross covariances. Also large CPU-time to solve the numerical solution of a large co-kriging system is required.

Another co-simulation technique that eliminates the requirement of solving the full co-kriging system has been proposed by Almeida et al.<sup>15</sup> The technique is based on a collocated co-kriging and a Markov-type hypothesis. This hypothesis simplifies the inference and modeling of the cross covariances. Since collocated technique is used, an assumption of linear relationship among the attributes needs to be applied.

The new co-simulation technique developed in this work is mainly to avoid the two-stage approach described above. The technique is a combination of simultaneous sequential Gaussian simulation and conditional distribution technique. Using this technique there is no large co-kriging system to solve and there is no linear relationship among reservoir attributes to assume. The absence of co-kriging from the process also means that the user is free from developing the cross-variograms. This improves the practical application of the technique.

In summary, the step-by-step procedure of incorporating the geological information can be described as follows.

1. Generation of isochronal intervals across the whole reservoir based on geological input.
2. Assignment of facies and petrophysical properties at well locations for each isochronal interval.

3. Modeling spatial relationships in both vertical and horizontal directions.
4. Generation of internally consistent facies and petrophysical properties at grid block level.

The first two steps of this process are described in other parts of the report. The main objective of this work is to provide the framework that can be used to complete steps 3 and 4 of the above procedure.

Traditionally, most of the algorithms in geostatistics are implemented using the FORTRAN language. The problem that is encountered with this program is the difficulty in extending or reusing the previously developed code to implement the newly developed techniques. Following the progress achieved in computer science technology, the object-oriented paradigm is thought to be appropriate for implementation in this area. This is mainly due to the philosophy of the object-oriented paradigm, which is extendibility, and reusability of a computer program. The most common computer language that uses this paradigm is the C++ language. This language is used to implement the co-simulation technique developed in this work.

The other reason for using the C++ language instead of FORTRAN is the ease in the development of a user-friendly program. The necessity of user friendliness may be appreciated by the fact that many sophisticated techniques have not been frequently used because they are very difficult to use. Therefore, in addition to having a good technique, a computer program should also possess the characteristics of being easy to use.

This report is divided into 7 sections. Following short summary in the first section, an introduction to the problem is given in Section 2. Section 3 presents the background theory used in developing the technique and Section 4 presents the approach used in this work. The implementation of the technique into computer program is discussed in Section 5. Several examples of the case studies will be presented in Section 6 before closing with the conclusion in Section 7. To assist the user in using the software, a section of appendix about the user guide is attached to this report.

## 4.3 Background Theory

### 4.3.1 Two Stage Approach

Integrated multi-disciplinary approach has become the model in the industry in conducting reservoir studies during the last several years. This effort can be seen from several publications at various conferences.<sup>2,6,7,10,11</sup> An approach which is commonly used in the industry to incorporate the geological information is the two stage approach.<sup>6,8-11</sup> Even though this approach has been shown to produce good results, it is not computationally efficient.

The main goal of the two-stage approach is to produce consistent petrophysical properties, i.e., porosity and permeability, with the underlying geological description. This goal is achieved through two independent processes, which are later combined to produce the final result by a filtering process. The first process is the simulation of geological description. The second process is the simulation of porosity. The permeability description can be obtained through correlation between porosity and logarithmic of permeability once the final porosity description is produced.

To combine the results of these two processes, i.e., geological and porosity simulations, a filtering process is applied. Figure 11 shows the schematic diagram of this process. The filtered porosity at a grid block is obtained by first examining at the type of geological unit at that location and then selecting the porosity value from the corresponding realization. Once the filtered porosity is known, the permeability value at that location can be determined using the suitable correlation. As demonstrated by several authors,<sup>6,8-11</sup> this approach has produced consistent result, between the petrophysical properties and the underlying geological description.

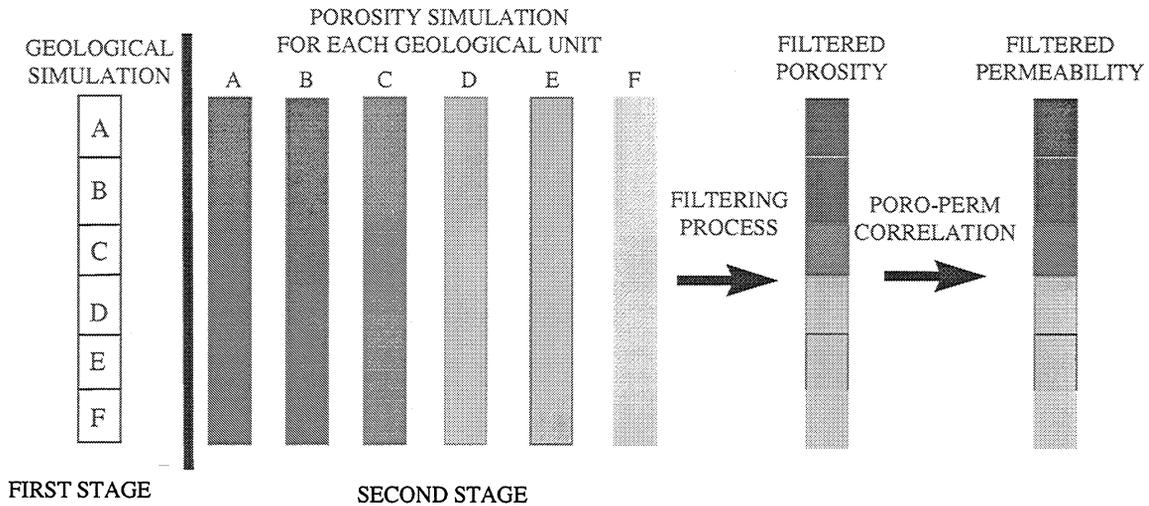


Figure 11: Schematic diagram of two-stage approach

Despite the good results produced using this approach, it suffers from efficiency problem. This technique requires intensive computation time, which not only includes the real CPU time, but also data preparation time. Additionally, this technique also requires a lot of static-computer storage.

The amount of time required to complete the study depends on the number of geological units to simulate. For example, if one realization of porosity simulation takes about 1 hour then for a system with 6 geological units it will take 6 hours. Also, if one realization needs 10 MB of hard-disk space, then it will take 60 MB to hold the temporary result that will be discarded after the filtering process. It is wise to remember that the amount of time and disk-space is very significant once the system reaches the order of millions of grid blocks which is the current trend in performing reservoir characterization. This example does not count the time required to prepare the input data, which could be significant, since all processes are conducted independently.

#### 4.3.2 *Horizontal Variogram Modeling*

The first step in constructing reservoir characterization using geostatistical methodology is to model the spatial relationship of the reservoir attribute through the calculation of variogram. In most cases, the calculation in the horizontal direction is never an easy task since it produces a pure nugget variogram. Figure 12 shows an example of this variogram. We can see that it will be

very difficult for us to model this variogram except to model it as a pure nugget model (dashed line). But, if we do this, we will end up with a realization that is very random. This would be unacceptable since in most cases the reality does show some continuity as predicted by the geologist.

A good variogram would show a clear correlation (structure) among the data points up to a certain lag distance, i.e., the variogram range. The dotted line of Figure 12 shows one of the typical structures that one wishes to have. The problem we encountered in this example is how can we build the variogram model for the horizontal direction based on the available data.

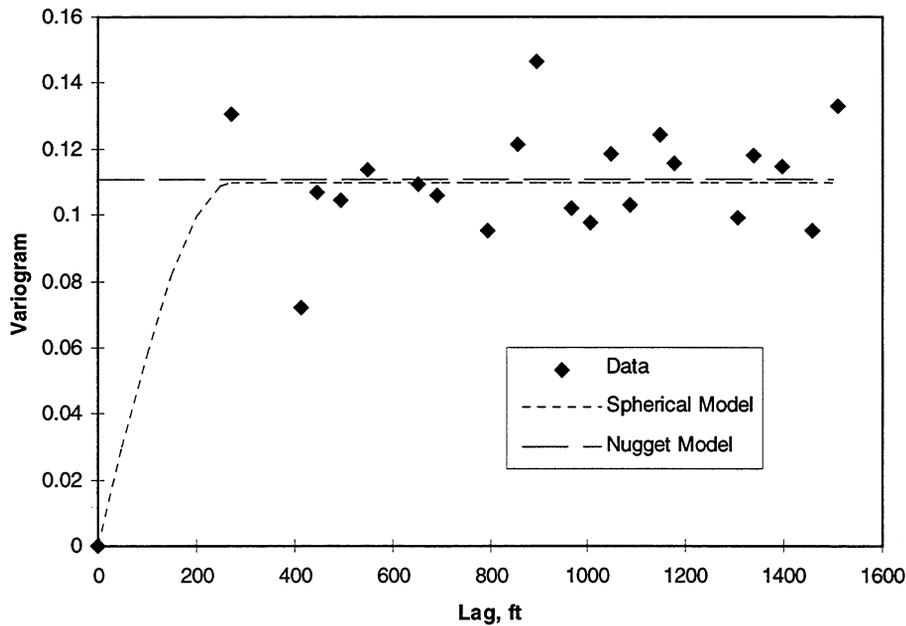


Figure 12: Typical horizontal variogram calculated using point to point correlation of well data

The problem presented in this example is due to the fact that the data are not close enough to each other. The closest distance between two horizontal data points is dictated by the closest distance between two wells.

In solving this problem, several approximation/short cut techniques have been used in the industry. Deutsch<sup>5</sup> used rule of thumb that depends on the geological deposition of reservoir of interest. Bahar<sup>6</sup> used the closest well distance as the variogram range in the Glenn Pool Field

study. Pande et al.<sup>7</sup> used the thickness of each rock type in each well and vertical average of porosity instead of the point value for the variogram calculation for the North Robertson Unit study. Significant improvement in the variogram structure was obtained but no justification is provided to validate the technique. Figure 13 shows an example of the calculated variogram using this technique. It can be seen that certain spatial structure exists in the system that enables us to model it with some confidence.

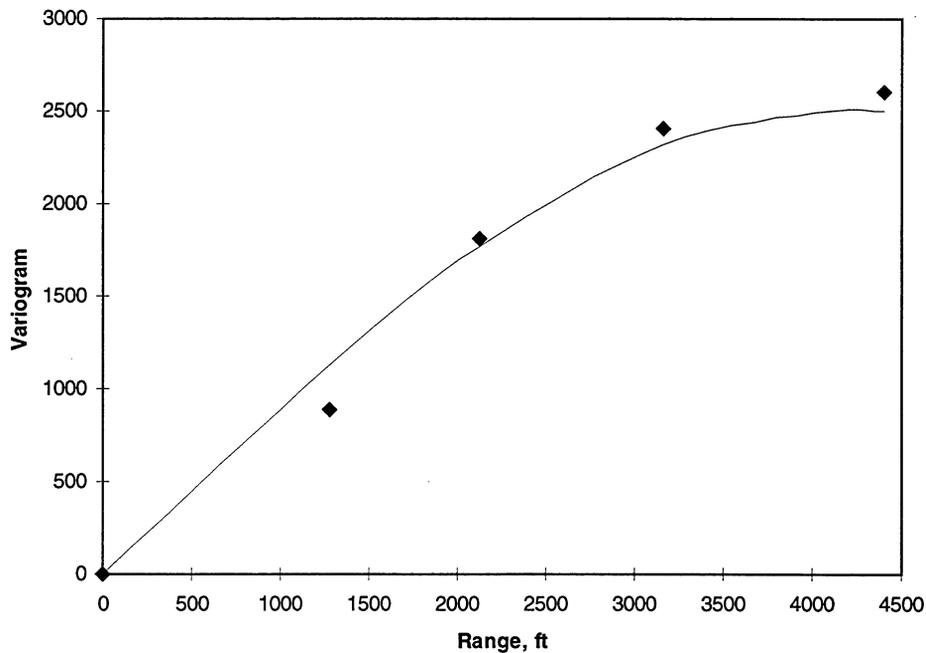


Figure 13: Horizontal variogram of rock type based on the thickness at each well

The solution to this problem has not yet been established. Recently Pizzaro and Lake<sup>15</sup> published a paper that estimates the interwell auto-correlation, i.e., range, using a simple method. The main idea in this technique is that the horizontal correlation must depend on the ratio of the areal-to-vertical variances and the correlation in the vertical direction. In this method, correlation range in horizontal direction is estimated using a type curve chart. The chart is generated from stochastic simulations using an assumed model (spherical, exponential, and truncated fractal). In this work, the approach used by Pande et al.<sup>7</sup> is further investigated and is proposed for implementation.

### ***4.3.3 Conditional Simulation of Single Attribute***

Conditional simulation techniques can be defined as geostatistical methods to generate the description of reservoir properties, which use the available quantitative and qualitative data. This method is a stochastic approach because reservoir properties are represented by random variables. The description of the properties generated by this method is conditional since the available data are honored at the sampled locations. The method simulates several equiprobable descriptions of the actual distribution of a property in the reservoir. In constructing the possible reservoir descriptions, the constraints imposed on the simulation process may include prior distribution of the simulated variables, spatial relationships in various directions, and geometry of geological shapes and sizes. As more constraints are incorporated in a conditional simulation process, more similar would be the equiprobable images.

Several conditional simulation techniques have been proposed in the literature. The techniques related to this work are :

- Sequential Gaussian Simulation.
- Sequential Indicator Simulation.
- Truncated Gaussian Simulation.
- Probability Field Simulation.

#### ***Sequential Gaussian Simulation***

Sequential Gaussian simulation (sGs) is an algorithm to generate realizations of a multivariate Gaussian field. Through the history of spatial stochastic modeling, Gaussian random fields are probably the oldest and the most commonly used models to represent spatial distribution of continuous variables.<sup>16</sup> Main reason for this is due to its extreme analytical simplicity. As can be seen from various publications, the use of multivariate Gaussian models have established record of successful applications.<sup>2,4,6,10</sup> Therefore, the choice of Gaussian model for modeling continuous model is very logical.

Sequential simulation approach is the technique to estimate an unsampled value from its conditional cumulative distribution function (ccdf) conditioned to all data within its neighborhood. In this case, simple kriging is a commonly used tool as the estimation technique. The neighborhood includes the original data as well as previously simulated values. For practical implementation, the number of conditioning data included in the estimation process is restricted to a certain nearest neighbors. This is due to fact that the number of previously simulated values becomes bigger and bigger as the simulation proceeds. The main advantage of this technique is computational speed.<sup>13,17</sup>

To use the sGs in simulating a reservoir attribute, e.g., porosity, permeability, facies, etc., we need to perform the normal score transform of the original data into the Gaussian data. Details about the transformation technique can be found elsewhere.<sup>18</sup>

The sequence of node visited during the simulation process normally follows a random path. Each realization of the simulation may or may not follow the same random path. At each unsampled location, simple kriging process<sup>18</sup> with the normal score variogram model is executed to determine the parameters (mean and variances) of the ccdf of the Gaussian random function. The estimated value of the attribute at the unsampled location for simple kriging technique can be evaluated using the following equation.<sup>18</sup>

$$x^*(\bar{u}_o) = \sum_{i=1}^n \lambda_i x(\bar{u}_i) + \left[ 1 - \sum_{i=1}^n \lambda_i \right] m \quad (1)$$

where, the kriging weight,  $\lambda$ , is obtained by solving the following system of linear equations.

$$\sum_{j=1}^n \lambda_j C(\bar{u}_i, \bar{u}_j) = C(\bar{u}_i, \bar{u}_o) \quad ; i = 1, \dots, n \quad (2)$$

For a system with small number of data points (as in the case discussed here), simulation through LU decomposition of the covariance matrix provides the fastest solution.<sup>19-20</sup> Therefore, this technique is appropriate for implementation to solve Eq. 2. Note that for this simulation we are using standard Gaussian/normal function with mean = 0. Thus, the right hand side of Eq. 1 reduces to the first term only.

In addition to the estimated value, the error variance is also estimated. Once the estimate and the variance are determined, a simulated value can be assigned to that location and becomes part of the data set for the next visited location. After visiting all the nodes, back transformation of the variable from the Gaussian field into its original space domain is required to assign values in the original domain.

Back transform procedure is generally based on the distribution of all data used in the simulation. Figure 14 shows the schematic diagram of the back transform commonly used in sGs procedure. In this example, this procedure is used to back transform either porosity or permeability. A different technique may be suitable for back transform of indicator variable/facies, which will be discussed later. Note that for permeability, the simulation is conducted using its logarithm value since in most cases permeability behaves as the normal distribution after log transformation.

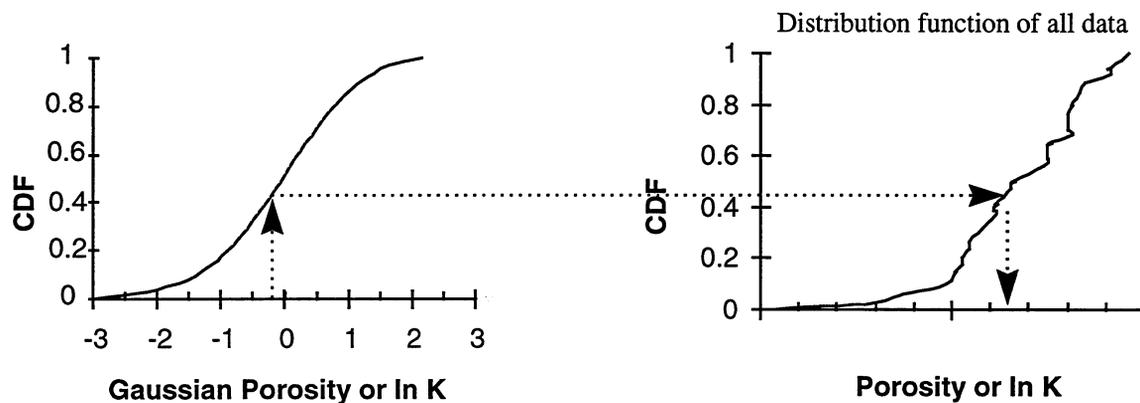


Figure 14: Common back transform procedure from Gaussian space into the original space for porosity and permeability

### ***Sequential Indicator Simulation***

Sequential Indicator Simulation (SIS) is the sequential simulation approach using discrete indicator variables. The main use of this technique for reservoir characterization is to simulate the geological facies. It is also being used for simulating continuous variables. Indicator simulation, however, is appropriate for facies. Facies, similar to indicators, can be described by discrete variables. That is, facies can only take finite number of values. Therefore, an indicator variable can be defined for each grid block by assigning a value of 1 for the present facies and 0

for the absent facies. For example, for a system with four facies an indicator value of [0,1,0,0] can be interpreted as facies 2 is present at that location and all the other facies are absent.

The sequential aspect of the simulation is the same as in the previous case, i.e., sGs. Thus, it will not be repeated here. The main difference is in the kriging process used, which is known as the indicator kriging. In indicator kriging, the goal is to estimate the indicator conditional expectation or conditional probabilities of existence of each indicator variable according to the following equation.<sup>18</sup>

$$\text{Prob}^* \{I_k(u) = 1\} = p_k + \sum_{i=1}^n \lambda_i [I_k(\bar{u}_i) - p_k] \quad (3)$$

The kriging weights,  $\lambda_i$ , can be calculated using the simple kriging equations, i.e., Eq. 2, whereas the marginal frequency of category k,  $p_k$ , can be inferred from the global proportion of the data of type k.

After probability value for each indicator variable is obtained for each unsampled location, an order relation correction for these probabilities distribution is required. This is done by defining a specific order of the indicator variable, such as 1, 2, ..., K, and re-scaling each of the probability value by cdf-type scaling, i.e., the probability value of each indicator is arranged in the cumulative form (with the maximum of 1.0) according to the order of the indicators. Once these probabilities are determined, facies can be assigned by drawing a random number,  $r$ , uniformly distributed in [0,1]. The interval in which  $r$  falls determines the simulated category (see Figure 15).

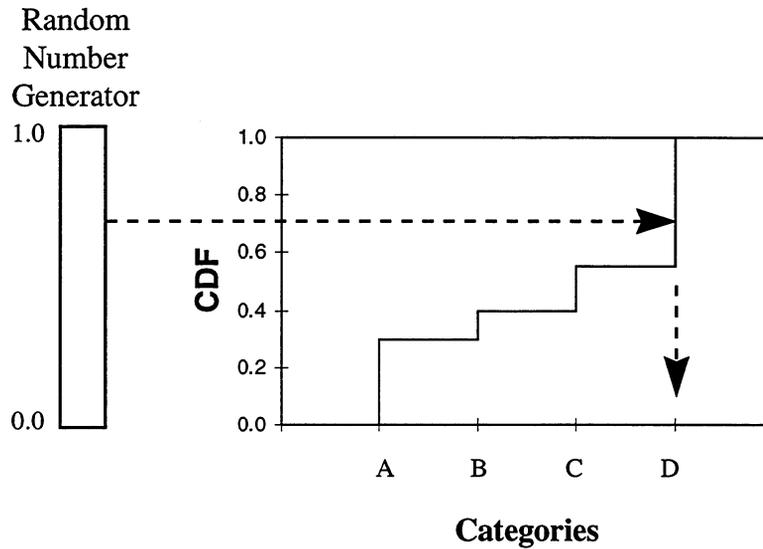


Figure 15: Category determination in sequential indicator simulation

### *Truncated Gaussian Simulation*

The truncated Gaussian technique has been widely used to simulate indicator variables, e.g., to simulate geological facies. Several software packages have been developed to implement this technique, e.g., HERESIM<sup>21</sup>, GTSIM<sup>22</sup>, etc. The method is simply a truncation of a continuous Gaussian random function at a number of truncation points. The truncation points is called the proportion curves that can be calculated using the indicator kriging followed by the order relation correction procedure as explained in the previous section. Also, normal score transform to each proportion curves is required before it can be used in the truncation process. One set of proportion curves is required for each grid block. See Figure 16 for an illustration of the method. In this example, there are 4 indicator/facies defined in the system, therefore there are 4 truncation points in each proportion curves used to truncate the Gaussian field as shown by the dotted lines (the last point is the maximum cdf, i.e., 1.0). From this figure we can see that the value from the sGs lies in the CDF that belong to Facies 2, therefore for this grid block, Facies 2 is assigned.

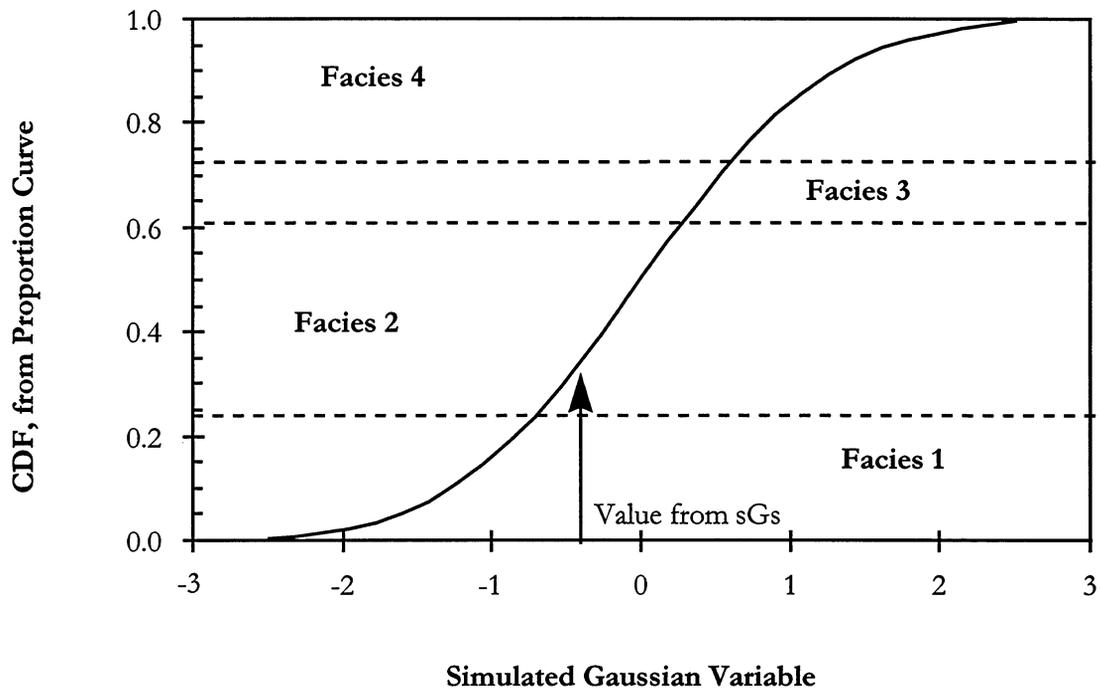


Figure 16: Schematic diagram of the truncated Gaussian method in assigning facies at unsampled location for a system with 4 facies - in this example, Facies 2 is assigned

As a summary, the complete procedure of this technique in simulating the geological facies (or other indicator variable) includes an estimation in the Gaussian domain and proportion curves for each unsampled location. The value in Gaussian domain can be produced by using the sequential Gaussian simulation, whereas the proportion curve can be obtained from the result of indicator kriging after transforming it into Gaussian space using normal score transform.

The main advantage of this method can be summarized as follows.<sup>22,23</sup>

- The algorithm is easy to use, fast, flexible and easy to constrain with external information like proportions.
- The vertical proportion curves, which can be incorporated into the algorithm as the source of the proportion curve, have proved to be a very simple but yet a powerful tool to simulate sequence stratigraphy.

#### *4.3.4 Conditional Simulation of Multiple Attributes*

The simulation of multiple attributes or commonly known as the joint simulation/co-simulation of several variables is the simulation of two or more variables that are spatially dependent. The main advantage of the co-simulation process is the ability to honor local relationships among various attributes simultaneously. One simple example of the importance of the honoring local relationship is the simulation of facies and porosity. If these two variables are simulated independently, it may happen that at certain location where shale is identified, high porosity is assigned. As discussed previously, the two-stage approach has been used to avoid this kind of problem. But, as it has been mentioned, this approach suffers a main drawback of inefficiency.

The joint simulation techniques that have been developed in the literature are few compared to the techniques for simulation of single attribute. The most logical way in conducting the joint simulation is to generalize the technique of single variable into multiple variables through modeling the cross covariances of the variables and using the co-kriging technique.<sup>12-14</sup> Common disadvantages of this technique is the requirement of tedious inference and modeling of covariances and cross covariances. Also large cpu-time to solve the numerical solution of a large co-kriging system is required.

A different approach has also been used by Almeida et al. in 1992.<sup>24</sup> The approach consists of drawing from successive conditional distributions with the most important variable is first simulated and the other variables are simulated by drawing from the specific conditional distribution. This technique is also used by Bahar<sup>25</sup> to generate permeability distribution from the simulated porosity distribution. The advantage of this technique is mainly its simplicity and the drawback is related to the indirect reproduction of the spatial correlation of the secondary variables. For example, if porosity is drawn from the conditional distribution between porosity and facies, then the spatial correlation of porosity is only reproduced through the correlation of facies meanwhile the spatial relationship of porosity itself is not honored.

In 1993, Almeida<sup>26</sup> presents a different approach to jointly simulate multiple variables. The approach is called sequential Gaussian co-simulation. It allows direct co-simulation of several interdependent random variables without tedious inference and modeling of a full cross-

covariance matrix as required by a full co-kriging-based algorithm. The algorithm can be seen as extension of sequential Gaussian simulation. It incorporates two key ideas, the collocated co-kriging and the Markov-type model of co-regionalization. This model relies on that the closest secondary-type datum screens the influence of all further away data of the same type. The procedure has been found to be computationally efficient. But, the implementation of collocated co-kriging requires that the relationships among variables be linear, which may not always be satisfied. Additionally, the back transform process does not guarantee that it only honors the distribution from certain conditional distribution since it is based on the distribution of all data.

## **4.4 Approach**

### ***4.4.1 Co-Simulation Technique***

The approach used in this work in performing the co-simulation procedure is the combination of sequential Gaussian simulation (sGs) and conditional distribution technique. Additionally, the truncated Gaussian technique and indicator kriging are also employed since it involves the simulation of indicator variable in Gaussian domain. Figure 17 presents the schematic diagram of the co-simulation technique used in this work.

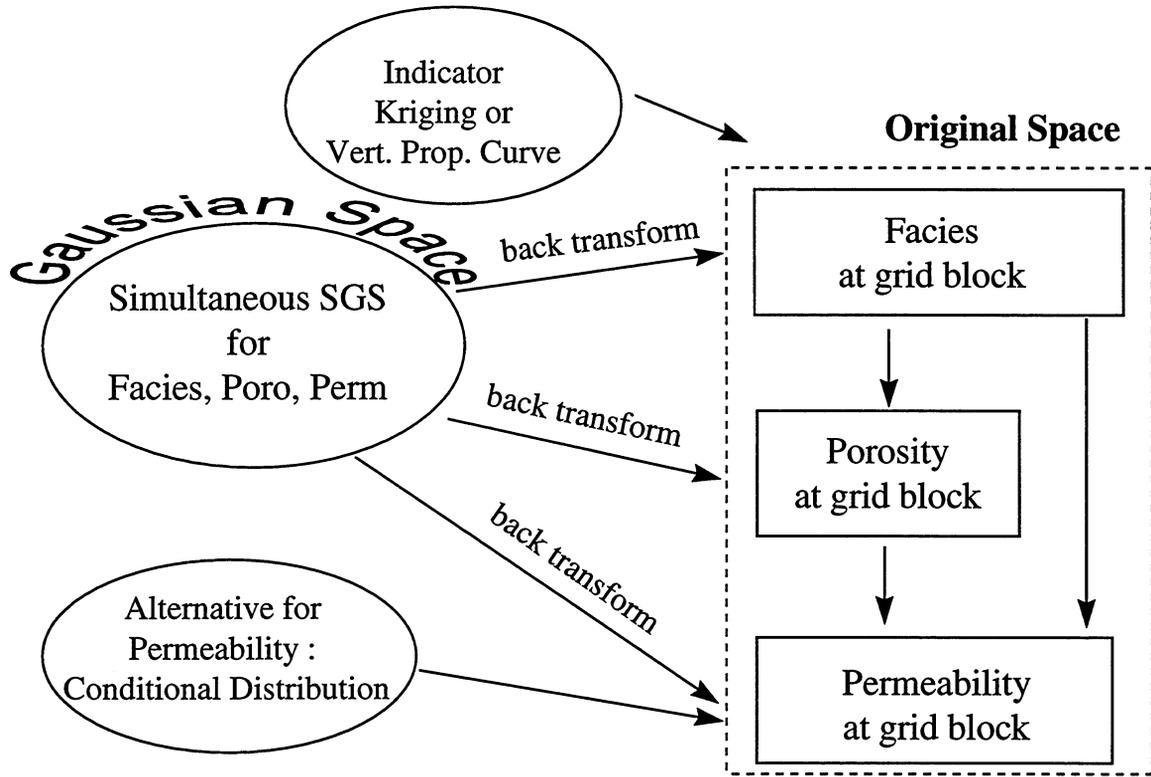


Figure 17: Schematic diagram of the co-simulation technique

The key techniques in this procedure are the implementation of the simultaneous sGs and the use of conditional distribution to back transform the secondary and tertiary variables, i.e., porosity and permeability. The back transform of the first variable, i.e., facies, is performed using the truncated Gaussian method as explained in the previous section. Using this technique, there will be no cross covariances/cross variograms required and no large co-kriging system to solve. This improves the practical application of the technique.

Theoretically, the technique is not limited to three variables only. We can extend it to as many variable as we want, but the implementation of this technique in this work is limited to these three variables only since these are the most important variables, based on the log and core data, that directly relate the geological information and petrophysical properties.

The first most important aspect about the co-simulation technique developed here is the use of sGs procedure simultaneously for facies, porosity, and permeability. It provides two advantages. First, the ability to incorporate the spatial relationship of each variable. Second, by selecting

same simulation procedure for all three variables, single search neighborhood can be applied where same data points are used in the kriging process of each variable. This technique makes the program computationally efficient. As in any sGs procedure, the sample points could either be the original data or the previously simulated nodes and each simulation node is visited only once using the selected random path.

The application of single search neighborhood during the simulation may not always be possible if some data for one or two variables are missing at one particular location. For example, consider the case where at one location facies has been identified but porosity and permeability are not available. For facies, this location is considered as the sampled point but for the other two variables this location needs to be treated as the unsampled location. So, when sampled points are searched during the search neighborhood process, this point can not be fully used to perform the kriging process of all three variables. It is only good for kriging of facies. Additional search is required for porosity and permeability until the requirement of minimum number of nodes to complete the kriging is satisfied.

In the most common situation, however, there will be porosity data for each location where facies is identified and limited number of permeability data. In this case, it is better to perform the single search neighborhood for facies and porosity simulations only and do separate search for permeability simulation.

In the case where the locations for permeability data are completely unknown then only conditional distribution technique can be applied for permeability simulation, i.e., without the sGs. This is the case where the “Alternative” procedure of Figure 17 is applicable.

Since the technique uses independent sGs for each variable, the spatial analysis required are the Gaussian variograms for facies, porosity, and permeability. There is no need to make the cross variogram between variables. Additionally, for the purpose of indicator kriging where its result is used in the truncated Gaussian procedure to back transform the Gaussian facies value, a set of indicator variograms is required. One variogram is needed for each facies.

The second most important aspect of this co-simulation procedure is the use of conditional distribution technique to perform the back transformation from the Gaussian space into the

original space. The idea of using conditional distribution is similar to the use of vertical proportion curves in the truncated Gaussian technique. The vertical proportion curves represents the probability or the percentage of geological facies at a particular depth. By using this proportion curve, it restricts the appearance of certain facies and allows other facies at certain depth with some probability. In this case, the use of conditional distribution technique will restrict the distribution of the simulated value into the distribution of the observed data only. This technique ensures that local relationship among the variables is honored.

In the standard sGs, the back transform procedure is conducted using the distribution of all data (see Figure 14). In this technique, the back transform of porosity, as the secondary variable, is constrained to the porosity distribution for certain facies, as the first variable, only. Figure 18 shows the schematic diagram of the conditional distribution procedure for porosity simulation. From this figure, we can see that, first, distribution functions or correlations between porosity and each facies are built (Figure 18a). Then, the cumulative distribution function (cdf) of porosity data in the original space (Figure 18b) as well as in the Gaussian space (Figure 18c) are calculated. The cdf in the Gaussian space can be calculated using normal score transform procedure.

Once a facies is assigned for a certain location, then the cdf for this facies is used to back transform porosity as shown in Figure 19. In the example shown in this figure, the facies for a certain location is identified as Facies A. Using the cdf of porosity for Facies A only, the Gaussian porosity obtained from the sGs procedure is back transformed to get the final porosity value. The dotted line on the cdf curve is calculated using linear interpolation of the data in the cdf curve. This line is used in assigning the final porosity instead of the discrete function. The dotted line with the arrowhead shows the direction of the process during back transform procedure.

The use of this technique can be extended to next variable, the tertiary variable, i.e., permeability. Since permeability is function of porosity and porosity is a function of facies then permeability need to be constrained to both of these variables. Figure 20 shows the schematic diagram of the back transform procedure using conditional distribution technique for permeability and its application in the back transform process. Figure 20 (a) shows the porosity-permeability

correlation for one of the facies. As we can see from this figure, several porosity classes are defined in this correlation. This is to minimize the variation of the permeability values. The histogram of permeability for one of the porosity class is shown in Figure 20 (b). Using this histogram, we can build the cdf and its corresponding value in Gaussian space as shown in Figure 20 (c) and (d).

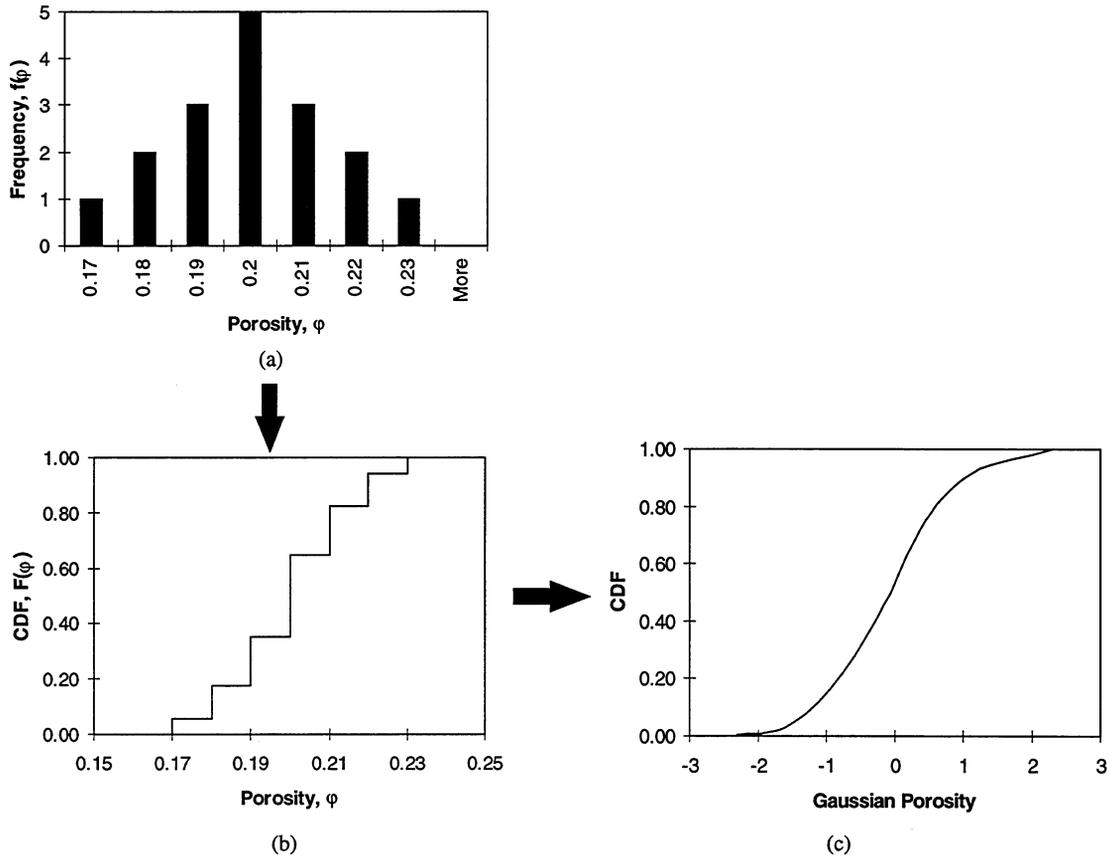


Figure 18: Conditional distribution procedure for porosity:

- (a) the frequency plot of porosity data for certain facies type
- (b) cumulative distribution function (cdf) of the corresponding porosity correlation
- (c) normal score transform of the cdf.

In the simulation procedure, once facies and porosity are assigned for a certain location, then we can assign permeability using the appropriate cdf, i.e., the cdf from the corresponding porosity class, and the Gaussian permeability value from the sGs procedure.

For the case where the sGs is not performed for permeability simulation due to the missing location information then random number between 0 and 1 can be used instead of the Gaussian transformed value.

In summary, the advantage of this co-simulation technique can be stated as follows. It honors the spatial relationship of each variable individually and at the same time it honors the local relationship among the variables through a procedure that is computationally efficient. The procedure requires less effort from the user. It avoids the requirement to perform the tedious work of cross-variograms analysis. The number of variograms required is equal to 3 (one variogram for each variable) plus  $n$  indicator variograms, where  $n$  is the number of facies.

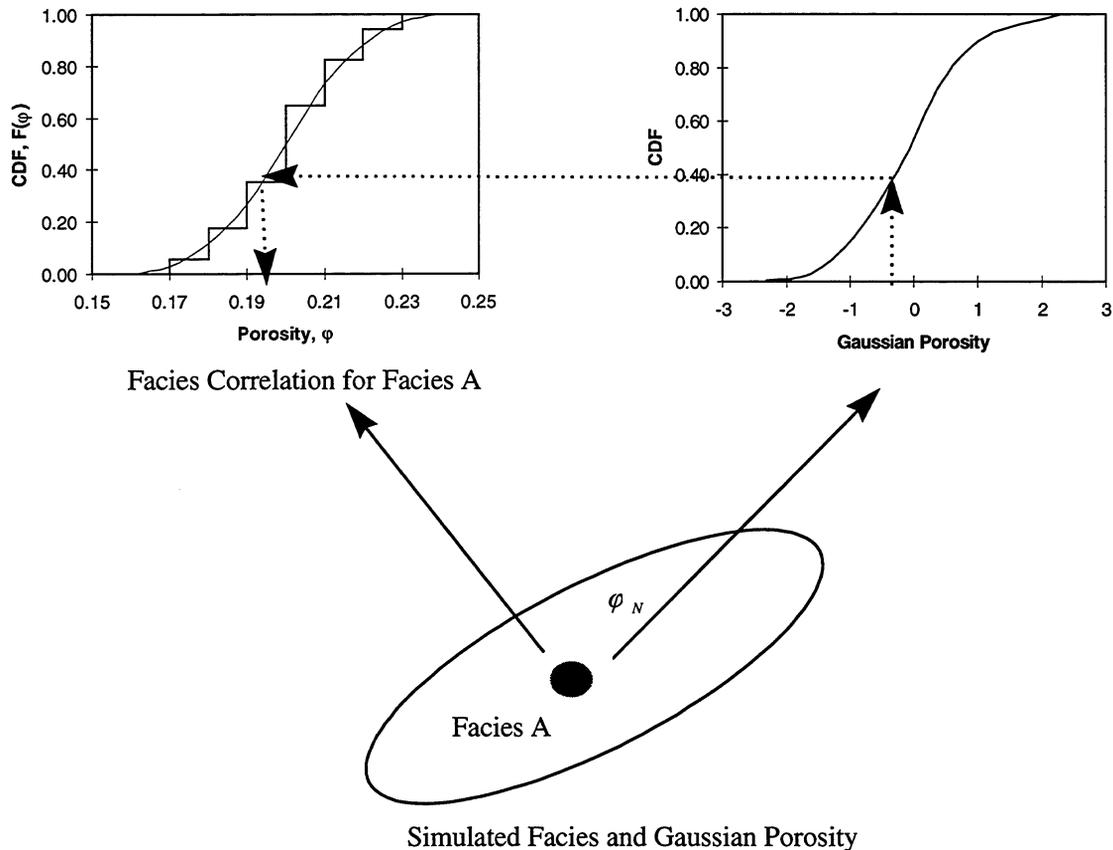


Figure 19: Example of the use of conditional distribution technique in the simulation

#### ***4.4.2 Soft Information***

The co-simulation technique described in the previous paragraphs use the data that originate mainly from well logs and well cores. Using these data and other information, the geologist will produce geological interpretation of the reservoir that includes the description of facies at well location. Facies distribution at, for example, 1-ft interval in each well could be generated based on this interpretation. This distribution becomes the hard data for the co-simulation technique. It is considered as hard data since it is based on the measured value.

The goal of the geological simulation is to closely replicate the geologist's interpretation. To achieve this goal, it may important that some other information, other than the hard data, to be incorporated in the geological simulation procedure. This information is commonly known as the soft data.

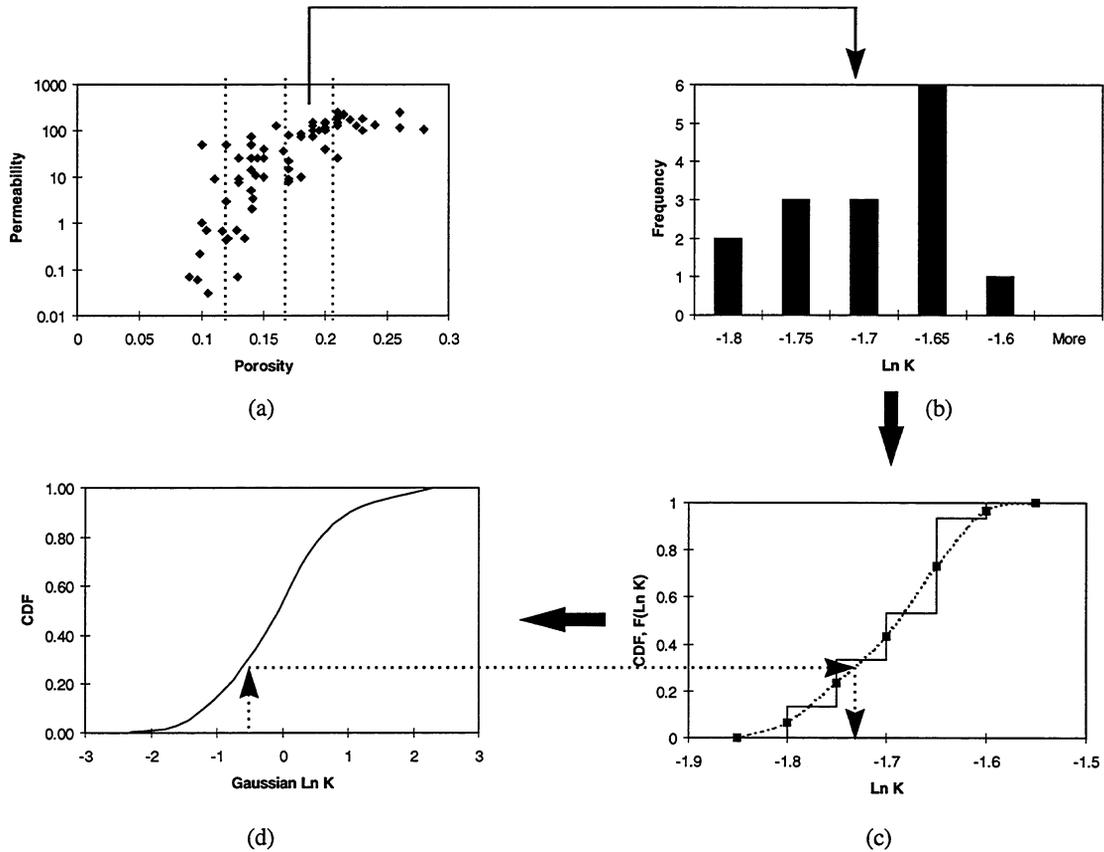


Figure 20: Conditional distribution technique for permeability

One type of information that can be considered as the soft data is the geologist's interpretation of facies distribution for the whole reservoir, e.g., the isopach map of the facies. This interpretation is considered soft information since it is not based on the measured value. Recall that, from the Background Theory section, hard data in the indicator simulation technique is represented by an indicator value that has the value of 1 if present and 0 if absent, e.g.  $I(\vec{x}) = [0 \ 1 \ 0 \ 0]$ , means facies 2 is present and others are absent at location  $\vec{x}$ . To incorporate the soft information, certain probability value can be assigned for each facies at one location. This information can be incorporated in the simulation procedure by modifying the definition of indicator function. For example, if there is a 50% chance of facies 1 and 50% chance for facies 2 to exist at certain location, then for a system with 4 facies, the indicator function becomes  $I(\vec{x}) = [0.5 \ 0.5 \ 0 \ 0]$ . A question mark can be given if the probability of one facies is not known with certainty. That is, if the probability of facies 1 is 25% but there is no certainty about the other 75% then we can assign a question mark for other facies, i.e.,  $I(\vec{x}) = [0.25 \ ? \ ? \ ?]$ .

The value specified in the soft data can be assigned as the result of indicator kriging for that location. In the case where the question mark is given for several facies then indicator kriging for those facies need to be performed. The cumulative value of the kriging results must be rescaled to (1 - sum probability of the known facies). So, the cumulative value of all facies becomes exactly 1.0. For example, if the soft data for a location  $\bar{x}$  is  $I(\bar{x}) = [0.25 \ ? \ ? \ ?]$ , the kriging for facies 2, 3, and 4 must result with the probabilities that satisfy the following condition :

$$\sum_i p_i = (1 - 0.25) = 0.75, \quad \text{where } 0 \leq p_i \leq 0.75, \text{ and } i = 2,3,4 \quad (4)$$

#### **4.4.3 Horizontal Variogram Modeling**

As discussed in the background theory section, the approach used in this work to solve the horizontal variogram problem is the approach used in the North Robertson Unit (NRU) study by Pande et al.<sup>7</sup> This is based on the field observation and physical reasoning of how the geologist makes the facies correlation in the field, i.e., by considering the average properties or the low frequency information of the data. Recall that, in this approach the horizontal variogram for rock type is calculated using the thickness of rock type and for porosity using vertical average of each well data. This approach will be referred as the average method in this report.

In the continuation of the NRU study, Freeman<sup>27</sup> calculated that the average of the permeability correlation is about 2418 ft. This calculation is based on the core analysis of the infill wells. Meanwhile, the average correlation range used in the geostatistical simulation to produce the petrophysical properties and rock type distribution of this study was 2882 ft. This result is very encouraging and becomes the motivation of the study presented in this work.

It is believed that this horizontal variogram problem is one example of the aliasing problem that occurs in many signal analysis problem. This effect will be discussed in this section before presenting the numerical experiment that is performed to support the hypothesis mentioned in the above paragraph.

### *Aliasing Effect*

In short, aliasing is the effect of lost information due to inadequate sampling. Figure 21 (a) through (d) represent signals with four different frequencies (continuous lines) and a similar sampling rate applied for each signal. From Figure 21(a) we can see that for the low frequency signal, the given sampling rate is adequate to reproduce the original signal. As the frequency of the signal becomes higher and higher (see Figure 21 (c) through (d)) it becomes more difficult to reproduce the original data. In the extreme case (see Figure 21(d)), the given sampling rate has completely lost the information of the high frequency signal and reproduces almost a flat line. The minimum-sampling rate required to reproduce the data is given by the Nyquist frequency.<sup>28</sup> That is, the illustration shown here that the Nyquist frequency for the given data is not satisfied by the sampling rate.

The illustration given in Figure 21(d) is believed to be the possible explanation to what happens with the horizontal variogram problem. That is, the typical porosity or facies distribution as given by well core or well log contain high frequency information whereas the sampling rate, which is represented by the distance of the wells, to reproduce the horizontal distribution of these two attributes is very small. Therefore, the aliasing effect occurs.

As suggested by the illustration presented in these figures, while restricted with the sampling rate, what we need to do is to somehow filter out the high frequency information of the data and keep the low frequency part of it. This process can be justified by observing the way the geologist makes the correlation of the reservoir attribute. In most cases, what the geologist trying to get is the big picture of the reservoir, which means to correlate the low frequency of the data.

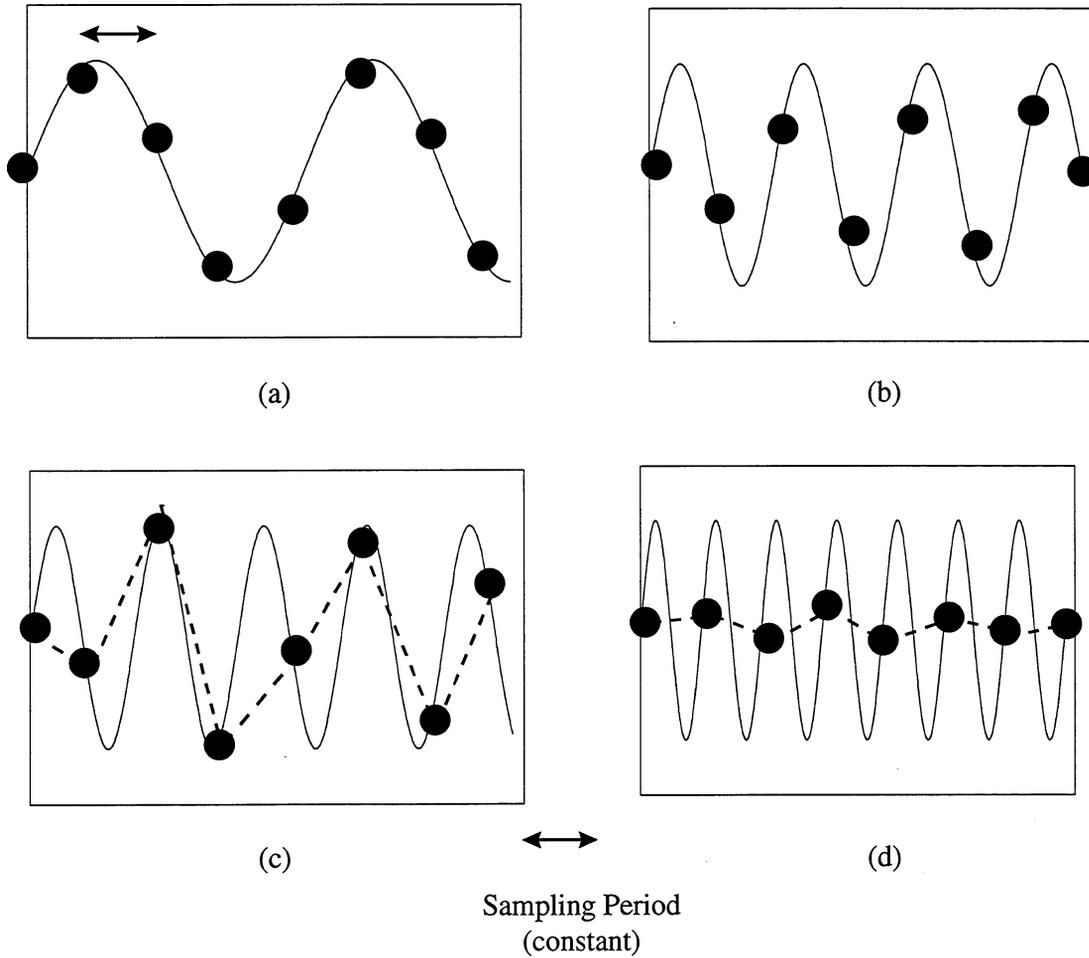


Figure 21: Illustration of aliasing effect in data sampling

***Reasons for Average Method***

The following facts are listed to support the approach of the average method.

- Vertical data as given by the well data contain high frequency information.
- Well spacing, which is fixed, determines the sampling rate.
- Point to point correlation would produce aliasing effect if the sampling rate is not high enough to satisfy the Nyquist frequency.

- Each well datum represents the point properties of the reservoir. Meanwhile, a point in numerical simulation represents properties of a grid block, which supports a much bigger area. That is, it represents average properties of the area it supported.
- Averaging is the process of smoothing the data, or in term of frequency, is the process of acquiring the low frequency information of the data.
- Geological information can be captured by the low frequency information.

### *Numerical Experiment*

The simplest way of getting the low frequency of the data is to take the average of them. Other sophisticated techniques such as Fourier Transform or Wavelet Transform can also be used. In this work, only the result of simple arithmetic average is conducted. The procedure used in the numerical experiment to support the theory presented above can be summarized as follows.

- Generate synthetic data for fine scale distribution using sequential Gaussian simulation.
- Calculate the variogram of the fine scale distribution, named as exhaustive variogram.
- Extract some data to represent well configuration, i.e., delete all data except those that are located at selected well locations.
- Calculate the variogram of the new system, named as point-to-point variogram.
- Calculate the average of the data at each well.
- Calculate the variogram of the average, named as the averaged variogram.
- Compare the three variograms.

In this report, the result of one of the case studies performed is presented. This case study is performed using the grid block configuration of  $75 \times 75 \times 100$ . Using the sGs technique, the grid block is populated with the synthetic porosity data. Several variogram models are used in creating the data. Variogram structure used in this case study is spherical structure without

nugget effect. Figure 22 shows the grid block configuration and the calculated exhaustive variogram. We can see that using exhaustive data we are able to get a very nice variogram structure.

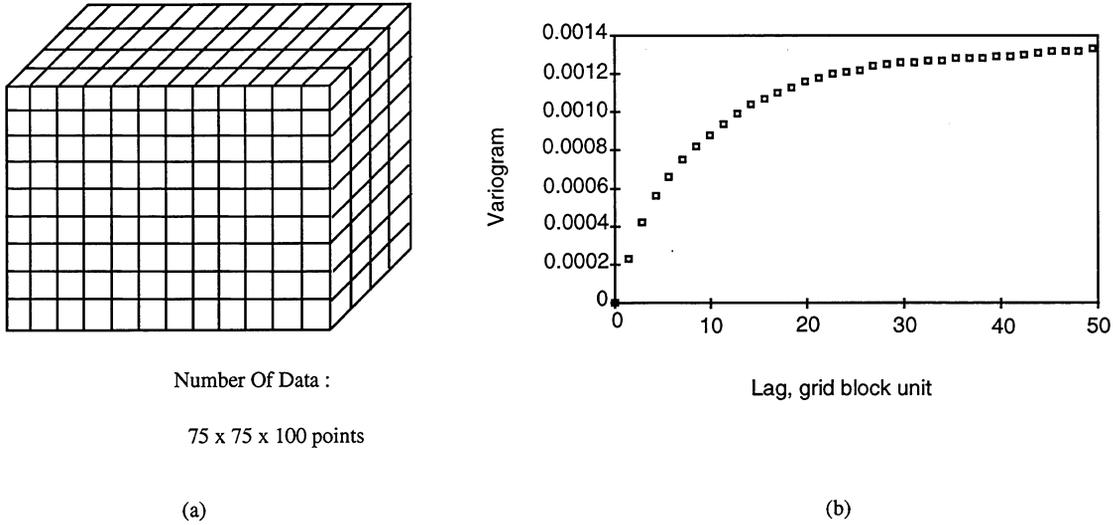


Figure 22: Detail data configuration and exhaustive variogram

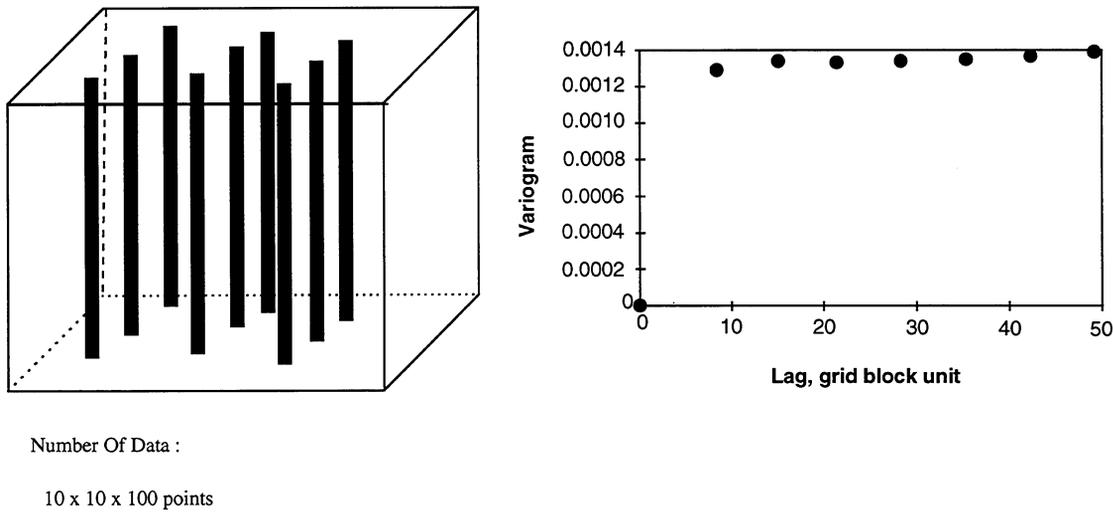


Figure 23: Well-based data configuration and point-to-point variogram

Following the numerical procedure presented previously, the synthetic well configuration can be generated and the corresponding variogram can be calculated using the point to point correlation as shown in Figure 23. We can see clearly that the pure nugget variogram is reproduced as

expected. As the last procedure of the numerical experiment, we need to make the average of the data in each well and calculate the corresponding variogram. This is shown in Figure 24. We can see a better variogram structure is obtained compared to the previous variogram.

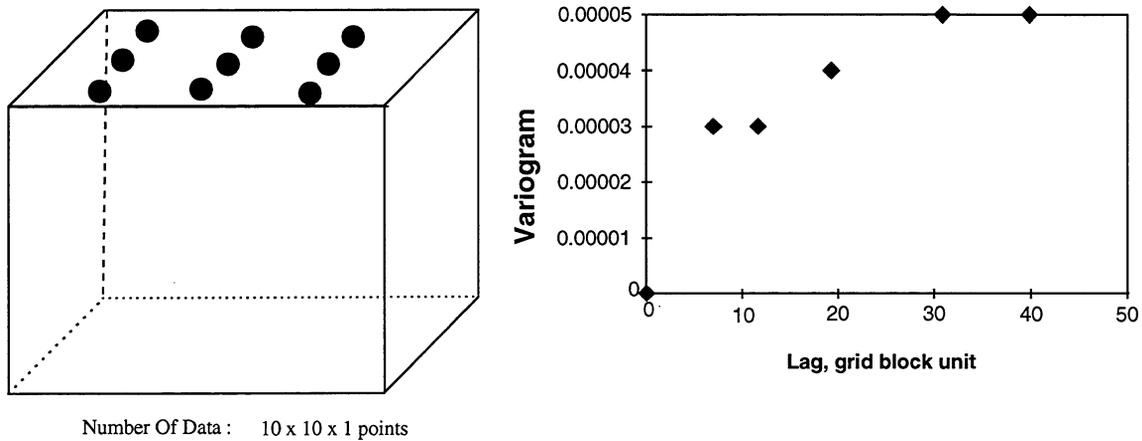


Figure 24: Data average configuration and the average variogram

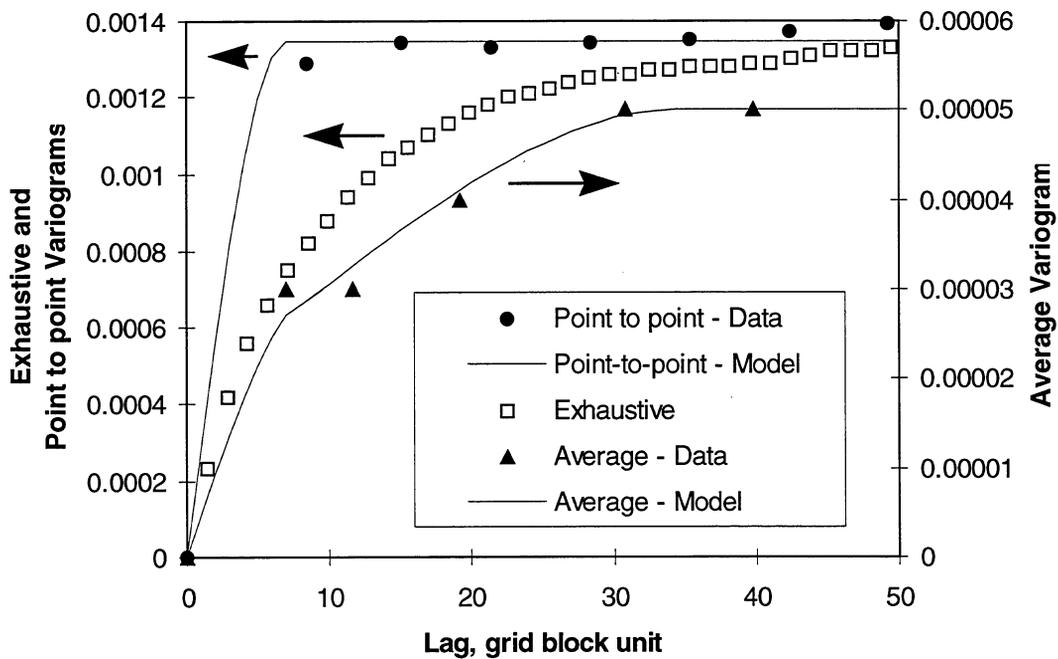


Figure 25: Comparison between exhaustive, point to point, and average variograms for 100 (10x10) well configuration

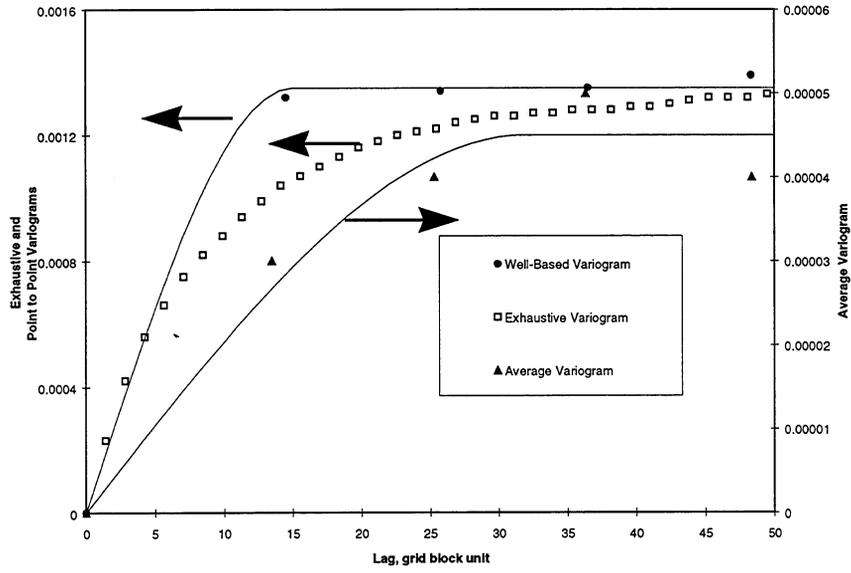


Figure 26: Comparison between exhaustive, point to point, and average variograms for 36 (6x6) well configuration

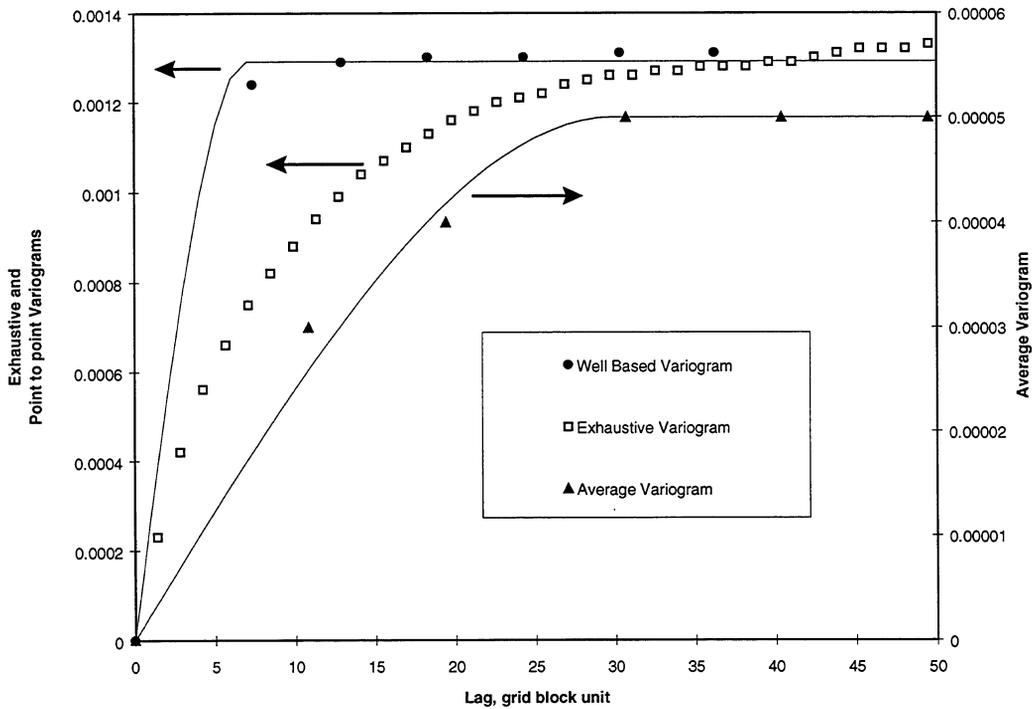


Figure 27: Comparison between exhaustive, point to point, and average variograms for 144 (12x12) well configuration

The comparison of all three variograms, i.e., exhaustive, well based, and average, for three different well configurations are shown in Figure 25 through Figure 27. The number of wells in each case is 100 (10x10), 36 (6x6), and 144 (12x12), respectively. From all of these figures, there are two important results that can be concluded.

1. The well-based variogram reproduces the sill of the exhaustive variogram very well, but the range and the structure is completely off. Meanwhile,
2. the average-based variogram reproduces the range and the structure type very well, but the sill is very far off.

Using this result, the solution to the horizontal variogram problem in this work is proposed as follows. The modeling of horizontal variogram can be done by taking the sill from the point to point correlation of the well data and the range and structure type from the variogram that is calculated using the average value of the vertical well data.

#### **4.4.4 Indicator Variogram Modification**

Indicator variogram is variogram calculated using indicator type variable. The formula to calculate variogram is given in the following equation,

$$\gamma(h) = \frac{1}{2N(h)} \sum_{i=1}^{i=N(h)} (x_i - x_{i+h})^2 \quad (5)$$

This equation is applicable for both continuous as well as indicator variable.

In the above equation, a pair of data points is selected as long as they satisfied the lag, h. This is satisfactory for continuous variable, but when it is applied to indicator variable, there are certain conditions that this equation contradicts with its physical interpretation. This contradiction, which will be explained below, inspires a modification to variogram calculation for indicator variable.

This modification is made to correct the way in which the data-pairs are selected. The effect of this modification is to reduce the number of data-pairs, which in turn will increase the sill of the variogram. This modification will be explained using numerical example as follows.

Consider well data that consists of 3 facies as given in Table 4. Generally, in calculating the variogram we need to collect all data into N number of pairs that are separated by certain lag distance and calculate the difference of the value at each pair according to Eq. 5. Applying Eq. 5 with  $h = 1$  into data of Table 4, we obtain :

For Facies A,

$$\gamma_A(1) = \frac{1}{2 \times 7} [(1-1)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2] = \frac{1}{14}$$

For Facies B,

$$\gamma_B(1) = \frac{1}{2 \times 7} [(0-0)^2 + (0-1)^2 + (1-1)^2 + (1-1)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2] = \frac{2}{14}$$

For Facies C,

$$\gamma_C(1) = \frac{1}{2 \times 7} [(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (1-1)^2 + (1-1)^2] = \frac{1}{14}$$

Depth	Facies	Indicator Variable		
		A	B	C
1	A	1	0	0
2	A	1	0	0
3	B	0	1	0
4	B	0	1	0
5	B	0	1	0
6	C	0	0	1
7	C	0	0	1
8	C	0	0	1

Table 4: Facies at well location and its indicator variable

In the above calculation we can see that when identical facies are present in a data pair, i.e., (1-1), the variogram value is equal to zero. This is logical. But, when the data pair does not contain the facies of interest, i.e., (0-0), the variogram value is also equal to zero, which means that the facies is correlated very well, or in other words the presence of this facies is very likely. This contradicts with the reality, which indicates that this facies does not even exist in that pair. Therefore, we need to exclude this pair from the calculation. By doing so, we will obtain different value of variogram for each unit as follows :

$$1. \quad \gamma_A(1) = \frac{1}{2 \times 2} [(1-1)^2 + (1-0)^2] = \frac{1}{4}$$

$$2. \quad \gamma_B(1) = \frac{1}{2 \times 4} [(0-1)^2 + (1-1)^2 + (1-1)^2 + (1-0)^2] = \frac{1}{8}$$

$$3. \quad \gamma_C(1) = \frac{1}{2 \times 3} [(0-1)^2 + (1-1)^2 + (1-1)^2] = \frac{1}{6}$$

From this example we can see that the modified indicator variogram should have a higher sill value compared to one that is calculated using the traditional method. Figure 28 presents the example of indicator variogram calculated using the traditional and the modified technique. We can see that the sill of the modified technique is higher than the traditional one as expected. The overall structure of the variogram may also be affected.

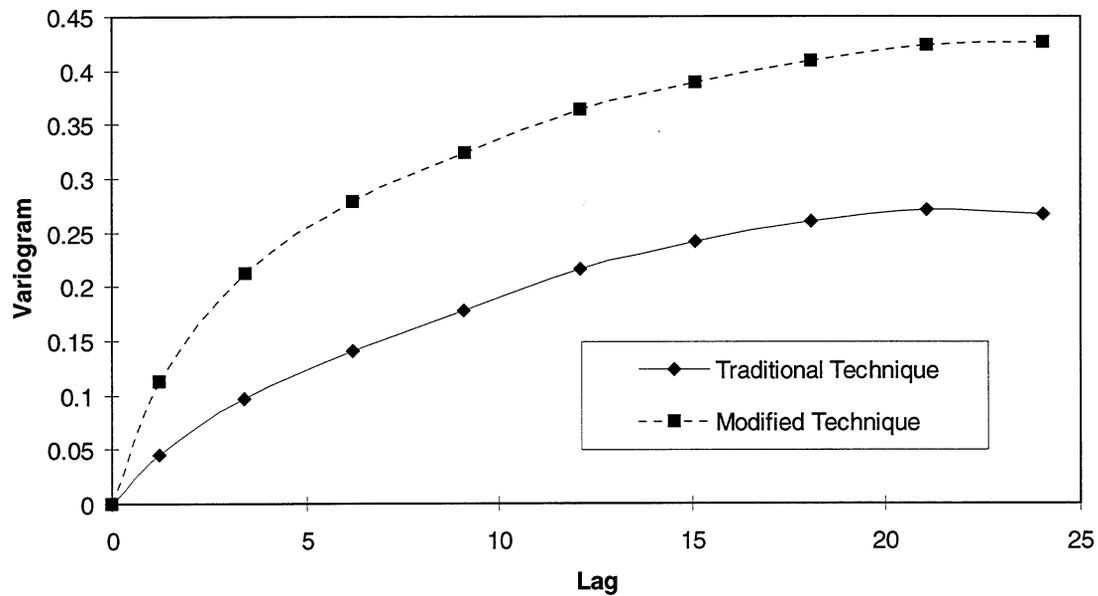


Figure 28: Comparison of indicator variogram calculated using traditional and modified technique

An important remark about this modification is the fact that the sill of the calculated variogram tends to be higher than the variance of data. Theoretically, it can be shown that for indicator variable the maximum value of the variance is equal to 0.25.<sup>23</sup> Ideally, the sill of the variogram will be close to its variance. This ideally situation will be hard to satisfy since it really depends on the way the facies occurs in the data. This modification, however, does not violate the limit that the indicator variogram can reach which is equal to 0.5. The limiting condition, which will happens when the appearance of a facies is only once, can happen to both traditional and modified techniques.

#### 4.5 Implementation

The implementation of the co-simulation technique presented in the previous section is conducted using the C++ language. This includes the calculation module as well as the interface module. The program is named COSIM. The interface of the program is targeted as a Windows-type program, which is the state of the art of program interface. An overview of the program structure, which mainly covers the structure of the calculation module is presented in this section.

Some of the calculation routines are either taken from the library of numerical recipe<sup>29</sup> or translation/modification into C++ from the FORTRAN source code of the GSLIB.<sup>18</sup>

The detail about the program interface and the user guide is given in the Appendix. The user guide includes an example of case study presented as a step-by-step procedure, from a raw data file, followed by the preparation of other data files, variogram analysis, preparation of simulation parameters, running the simulation, and closes by qualitative and quantitative evaluation of the result.

The program is designed to follow the object-oriented paradigm. That is, several classes are created in the program to initiate objects during program execution that will perform certain functionality. It is designed to be flexible for future modification as much as possible. The structure of the program is shown in Figure 29. From this figure we can see that a class, namely Application, is defined as the parent class. This class contains several member variables and member functions that are common to geostatistical simulation techniques. Another class, namely Simultaneous SGS, is defined as a derived class of Application. Therefore, all members variables and member functions declared as public or protected in the class Application, are available for Simultaneous SGS.

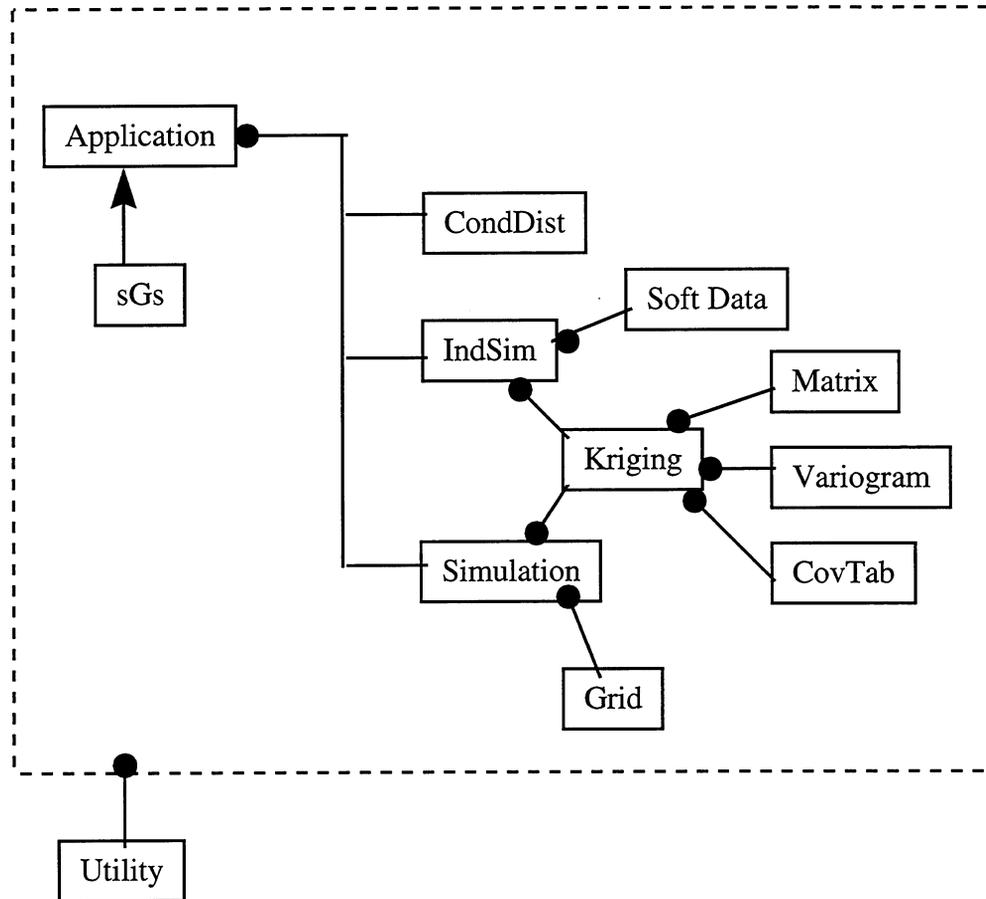


Figure 29: Program structure

Class Simultaneous SGS contains member functions that specifically are designed to perform the simultaneous sequential Gaussian simulation and conditional distribution for back transform, i.e., the co-simulation technique. If a new technique is planned in the future to use the overall program framework then another class, in the same level of sGs, can be defined to inherit from class Application. Or, it can also be derived from class sGs, i.e., one level below sGs. In this case, some or all facility of sGs can be used in the newly developed class.

The name and short description of classes defined in this program is summarized in Table 5. The detail description of each class will be the subject of the following paragraphs.

### ***4.5.1 Class Application***

As explained previously, Class Application is the parent class for the simulation technique. It is developed using the principal of polymorphism and dynamic binding where it contains several pure virtual functions, i.e., function which is declared but does not immediately implemented in the class where it is declared. One of the most important pure virtual function is called the cosimulation function. This function should be fully implemented inside each class that is inherited from it. Therefore, different implementations can be defined for this function, i.e., polymorphism. In current version, this function is defined in class Simultaneous SGS only, since this is the only class that derived from it. The polymorphism principal is implemented in this program to accommodate future development/extension of the program, i.e., by creating a new implementation of the cosimulation function inside a new class that is derived either from class Application or class Simultaneous SGS. This way, we do not have to modify the existing implementation of cosimulation function, while the new function still has fully access to all functions of the it is derived from. Additionally, this function will be dynamically binded during the execution of the program, which means that the program will dynamically determine which cosimulation function to select during an execution. This aspect is very important for application that runs under the message-driven mechanism, such as Windows application.

Several other functions in this class are declared as virtual function, but not as pure virtual function. In this case, the implementation of this function can be implemented directly in this class or it can be re-implemented in the derived class. Therefore, a newly developed program can either use this function as it is or it can make a completely new implementation.

Most of other classes are directly used in class Application as an object through client relationship. In client relationship, the object has access to member variables and member functions that are declared in the corresponding class as public members only. These objects are declared here mostly for dynamic initialization purposes only.

<b>Class Name</b>	<b>Description</b>
Application	Parent class of COSIM program
SimultaneousSGS	Define the simulation technique to be used
IndSim	Provide the procedure to calculate proportion curves using indicator kriging.
CondDist	Provide the procedure to perform the conditional distribution technique.
Kriging	Provide the procedure to estimate the node value either by Simple kriging or Ordinary kriging technique with or without covariance table.
Variogram	Provide the calculation of variogram and/or covariance value between any two points in 3D for a given Variogram model.
CovTab	Provide the calculation and storage for the covariance table.
Simulation	Provides the procedures to perform sequential simulation technique.
Grid	Provide the grid block network of the simulation system. Its main function is to store several global variables.
Point3D	Provide the structure to represent a 3D point.
Utility	Provide several utility functions that are common in geostatistical simulation technique.
MyMatrix	Provide the procedure related to matrix operation.

Table 5: C++ class summary used in the program

#### **4.5.2 Class Simultaneous SGS**

This class is derived from class Application. It is the main driver to perform the co-simulation procedure, i.e., through function cosimulation. Basically, this class contains some helper functions and one main loop that will generate multiple realizations of co-simulation technique. In each realization, two functions are called, namely cosimultaneous SGS and

EvaluateGridValue. From its name, the first function is self-explanatory whereas the second function is for the back transform procedure using the conditional distribution technique. The helper functions perform some calculations that are useful for either one of these two functions. The output of the simulation result is written to binary file inside the EvaluateGridValue function.

#### **4.5.3 Class IndSim**

The main purpose of this class is to provide the facility in calculating the proportion curves which in the co-simulation technique will be used to truncate the Gaussian data of the facies simulation result. As explained in background theory section, the proportion curve is created using indicator kriging. Therefore, this class mainly consists of one main loop that does indicator kriging followed by the correct order relation procedure for each grid block. The indicator kriging routine given in this class accommodates the soft data information.

#### **4.5.4 Class Simulation**

This class contains variables and functions that are commonly found in the sequential simulation technique such as selecting the random path, assigning data and simulation result into simulation nodes, searching the nodes for kriging (search neighborhood procedure), etc. Therefore, it can be classified as the helper class for the class Simultaneous SGS.

#### **4.5.5 Class Variogram**

This class stores and controls the spatial relationship information of an attribute. For each object declared, it stores variogram model and is capable of calculating the covariance based on the variogram model between two arbitrary points. This function is very useful in building the element of kriging matrix or the covariance table. The later is used for optimization in search neighborhood procedure.

#### **4.5.6 Class CondDist**

This class provides the facility to perform the full conditional distribution technique for permeability simulation. The conditional distribution procedure for porosity is not included in

this class but rather directly incorporated in class Simultaneous SGS. Class CondDist includes the procedure to divide porosity-permeability correlation for each facies into several porosity classes. The program automatically divides each correlation into 4 classes based on the quartile values. After properly initializing this class, a function named AssignPermeability can be used to sample the permeability value from the appropriate cdf.

#### **4.5.7 Class CovTab**

This class contains three big arrays, one each for facies, porosity, and permeability, of covariance values. The main purpose of this class is to provide a quick answer in getting the covariance value between two points which is used to build the element of kriging matrix. Therefore, before kriging is performed, a member function namely GenerateCovarianceTable has to be called first to initialize the array.

#### **4.5.8 Class Kriging**

The kriging class is used to perform the kriging of the unsampled value. In building the covariance matrix, this class can either use the covariance table, which is provided by the object of class CovTab, or directly calculate the covariance between two points through the object of class Variogram. Since the kriging matrix is a positive definite matrix, the Cholesky decomposition technique can be used in solving the linear equation of the kriging system. Also, the choice of this technique is based on the fact that the number of nodes that determines the size of the kriging matrix is relatively very small, i.e., around 10 - 20 points. For this case, this technique is very efficient.<sup>19-20</sup>

#### **4.5.9 Class Grid**

This class is a helper class to store most of the global variables typically required in the sequential simulation technique, such as the grid block configuration, i.e., number of blocks, size, and its origin, number of data, etc. Therefore, this class and class Simulation, will be very useful for new development program that uses the sequential simulation technique. This class also contains several dynamically allocated arrays to store various information, such as data values in

original space or in Gaussian space, the location of sample points, etc. and temporary result of the simulation variables, especially related to the search neighborhood.

#### ***4.5.10 Other Classes***

Several other classes defined in this program are class SoftData, class MyMatrix, class Point3D, class Utility. Class SoftData is used to help the class IndSim in performing the indicator kriging when soft data are defined for a particular location. Class MyMatrix is created to facilitate the Cholesky decomposition technique. It provides with the easy access to matrix operations. Class Point3D is a template structure definition of 3D point. Using this class, programming is made easier since it provides several operator overloading functions that gives the shortcut for several simple point operations such as addition and subtraction. Also it provides with the operator “=” that makes the program easier to read and to write. Both class MyMatrix and class Point3D are template classes. Therefore, they can be used to store data with any data type (int, float, double, etc.).

The most important helper class that is used by many other classes is the class Utility. It is basically the collection of several functions, which are useful for many applications. Some examples of the function available from this class are sorting arrays, creating index of an array, locating a point in array, calculating random number, reading GEOEAS titles, performing normal score transformation, etc.

## **4.6 Application Examples**

To demonstrate the capability of the program in simulating the consistent petrophysical properties with the underlying geological description, two cases are presented in this section. The first data set are taken from the Self Unit and Track-7 Unit, Glenn Pool field, Oklahoma, which are the sandstone reservoirs whereas the second data set are taken from the North Robertson Unit, West Texas, which is the carbonate reservoir. In both cases, the geologist has generated the geological description at well location. A set of porosity logs are available at several wells and a correlation of facies/rock type, porosity, and permeability are available from the core data.

#### 4.6.1 Sandstone Reservoir

The geological unit used by the geologist in describing the Glenn Sandstone in the Glenn Pool Field is the Discrete Genetic Interval (DGI).<sup>40</sup> Reference 31 and 32 provides detail information about this geological description. Kerr and Ye have defined 6 DGI's in the vicinity of Self Unit,<sup>31</sup> namely DGI A through F, and 7 DGI's in the vicinity of Track-7 Unit,<sup>32</sup> namely DGI A through G. Each DGI may consist of several facies such as channel fill, crevass splay, and floodplain mudstone.

The first result shown here is the 2D simulation of DGI A of Track-7 Unit. In the vicinity of this unit, three facies exist for the DGI A, namely channel-fill, crevass splay, and shale/mudstone. For simulation purposes, these three facies are named facies 1, 2, and 3, respectively. The dimension of the Track-7 Unit is 2640 x 2640 sq. ft. The simulation is conducted using 80x80 grid blocks in *x* and *y* directions or a total of 6,400 grid blocks.

These data set are used as the case study example in the Appendix B, i.e., the User Guide. Therefore detail variogram analysis and other simulation parameters can be seen directly in this Appendix B. Only the result will be presented in this section. Figure 30 presents the geologist interpretation of facies distribution for DGI A in the vicinity of Track-7. The simulation results for 4 different realizations are shown in Figure 31 (a) through (d). Comparing these figures, we can see that the simulation matches the geologist interpretation very well, especially in replicating the channel and splay distribution. The difficulty of the program to exactly match the shale distribution can be understood from the fact that there are only two data points available and both of them are located very close to each other (see Figure 30).

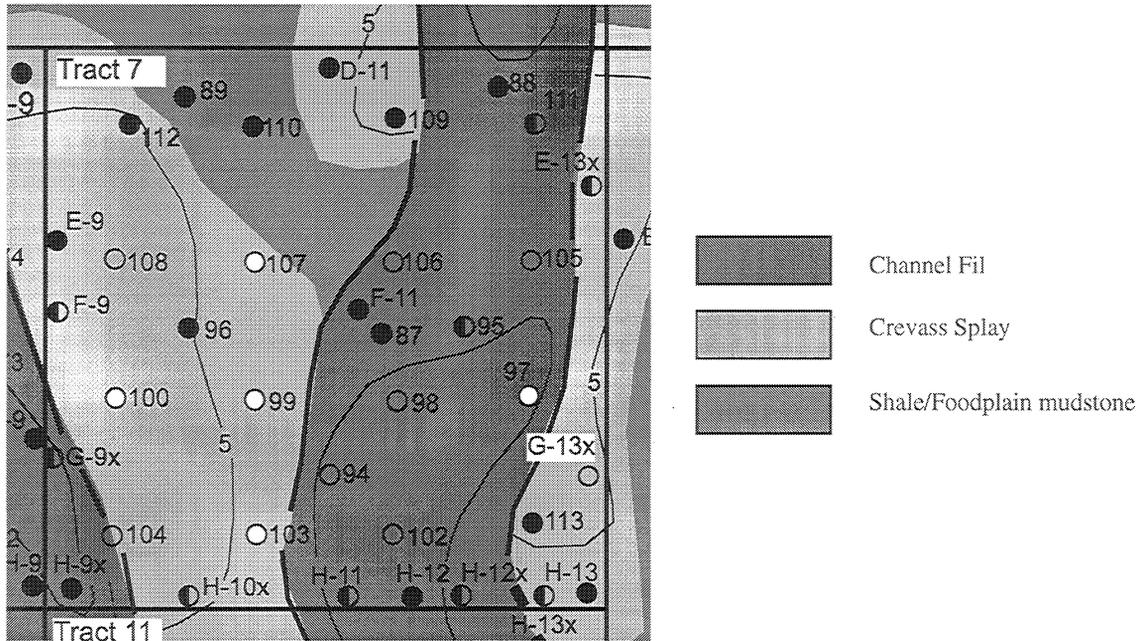


Figure 30: Geologist interpretation of facies distribution in the vicinity of Track-7  
(after Kerr and Ye)<sup>32</sup>

The global pdf comparison between the simulation and the conditioning data selected randomly is shown in Table 6. From this table we can see that the simulation produces proportion of each facies that matches the data very well.

Facies	Data	Real-3	Real-20	Real-18	Real-15	Real-1
1	0.4722	0.4606	0.4922	0.5072	0.5064	0.4828
2	0.4722	0.4727	0.4706	0.4353	0.4586	0.4833
3	0.0556	0.0667	0.0372	0.0575	0.035	0.0339

Table 6: Global PDF comparison between simulation and data for 5 realizations

The corresponding petrophysical properties distribution for this case is shown in Figure 32. From this figure we can see clearly that both porosity and permeability distribution follows facies description very well. The other realizations of the petrophysical properties, not shown in this report, have also been observed to produce similar result. This result shows that this program is capable in generating consistent petrophysical properties with the underlying geological description, which is the major technical aspect required from a co-simulation procedure. The other aspect that is important in this work is the efficiency of the program in replacing the

traditional technique, i.e., the two-stage approach. This aspect will be discussed in the next paragraph.

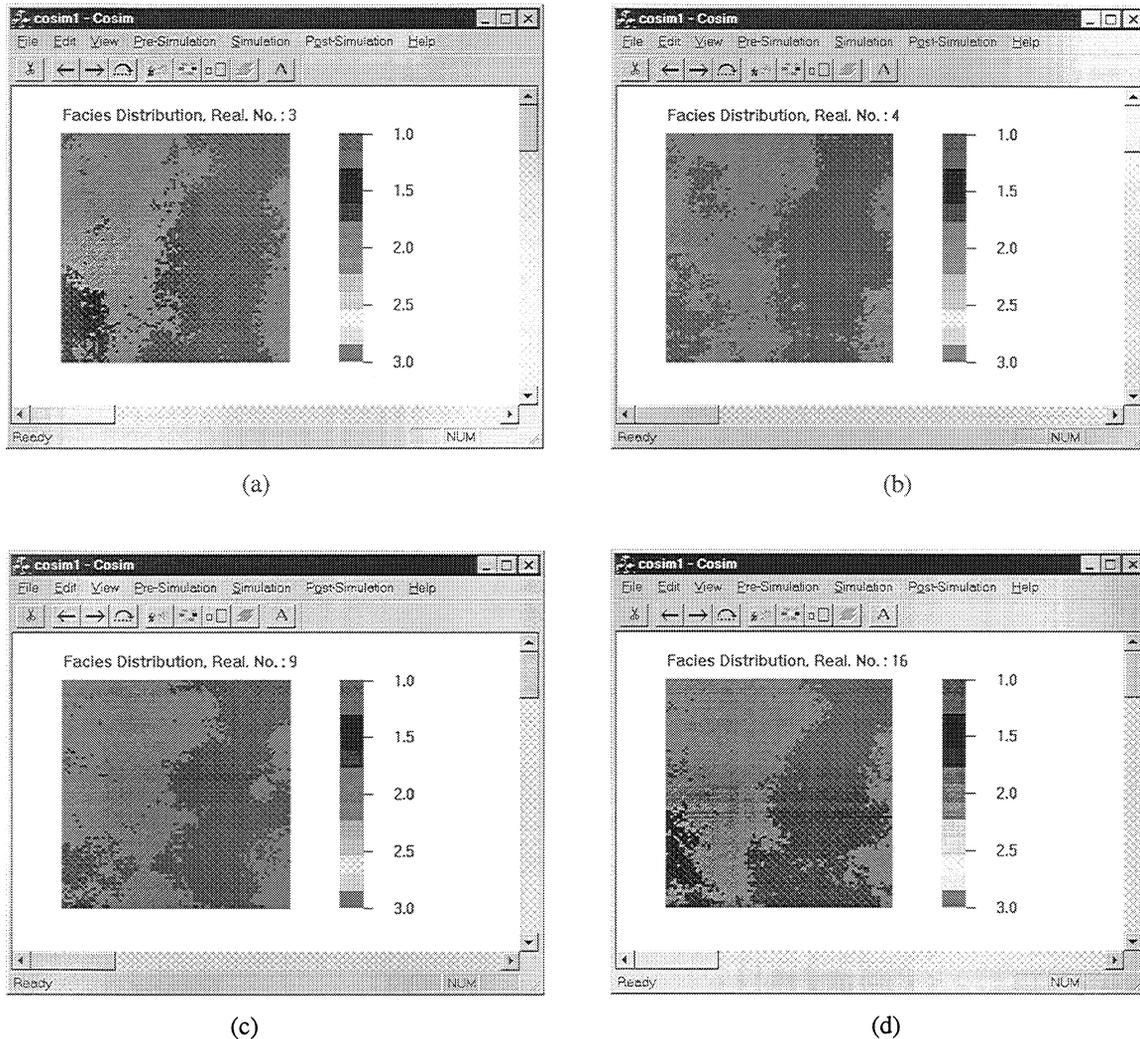


Figure 31: Multiple realization of facies distribution of DGI A - Track 7 Unit

The amount of time required to complete this simulation run can be considered very small. With the Pentium 200-Pro machine, it takes about 7 sec. The variation in simulation time using different grid block configuration and different number of realizations is shown in Figure 33. As expected, the amount of time vary significantly when the configuration is increased or decreased.

The efficiency of the program can be much more appreciated if we directly compare the actual time required to complete one case study starting from raw data until the distributions of facies,

porosity, and permeability, are obtained. This include data preparation and variogram analysis. Using the traditional approach and GSLIB tools, Paranji<sup>32</sup> estimates that, at least, it takes about 1 hr. and 10 min. for him to obtain similar result. The approach used by Paranji is not exactly the same as described in the background theory section, i.e., the two stage approach, but rather consists of a series of stochastic simulations, such as sequential indicator simulation followed by kriging with external drift. Meanwhile, using the COSIM package all operations take only about 7 minutes. We have to include the data preparation time in this comparison since the traditional approach requires a lot of human intervention to complete the work while the co-simulation technique has eliminated a big part of it.

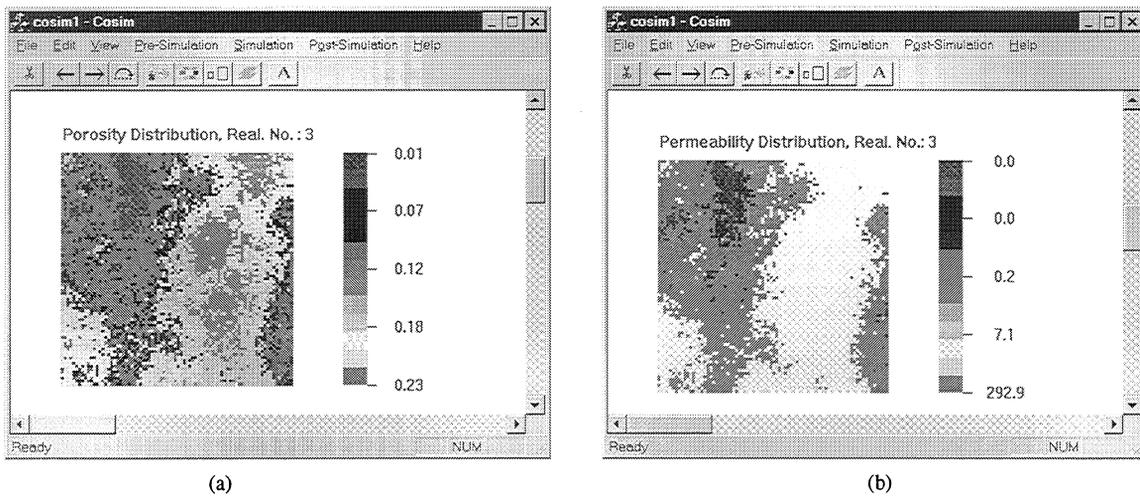


Figure 32: Porosity and permeability distribution of DGI A for Track-7 Unit from one of the realization

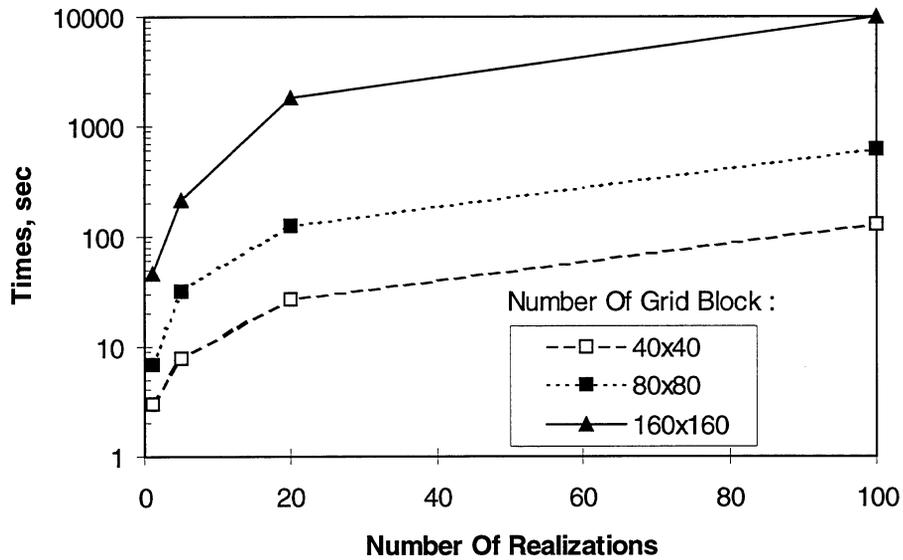


Figure 33: Computation time comparison for different grid block configurations and different number of realizations

The second result shown in this section is taken from the Self Unit data where the 3D simulation of DGI, not facies, are conducted. The purpose of this simulation is to show that the simulation is capable in simulating the sequence stratigraphic characteristic of the DGI. The number of grid blocks used in this example is 256,000 (40x40x160) grid blocks where the size of each block is 66 x 66 x 1 ft<sup>3</sup>. The North-South cross section of the DGI distribution at the middle of the reservoir is shown in Figure 34. The corresponding petrophysical distribution is shown in Figure 35 and Figure 36

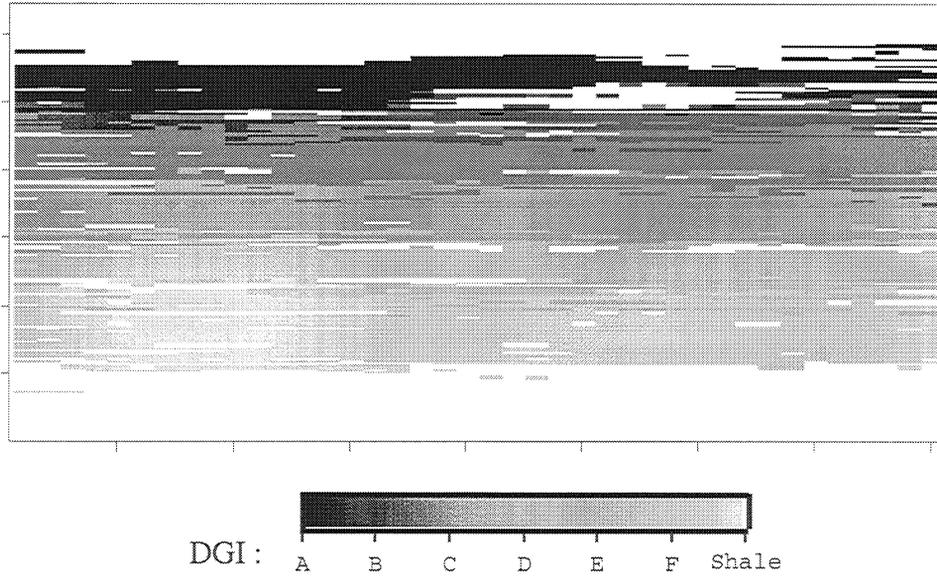


Figure 34 North-South Cross Section of DGI distribution of the Self Unit data

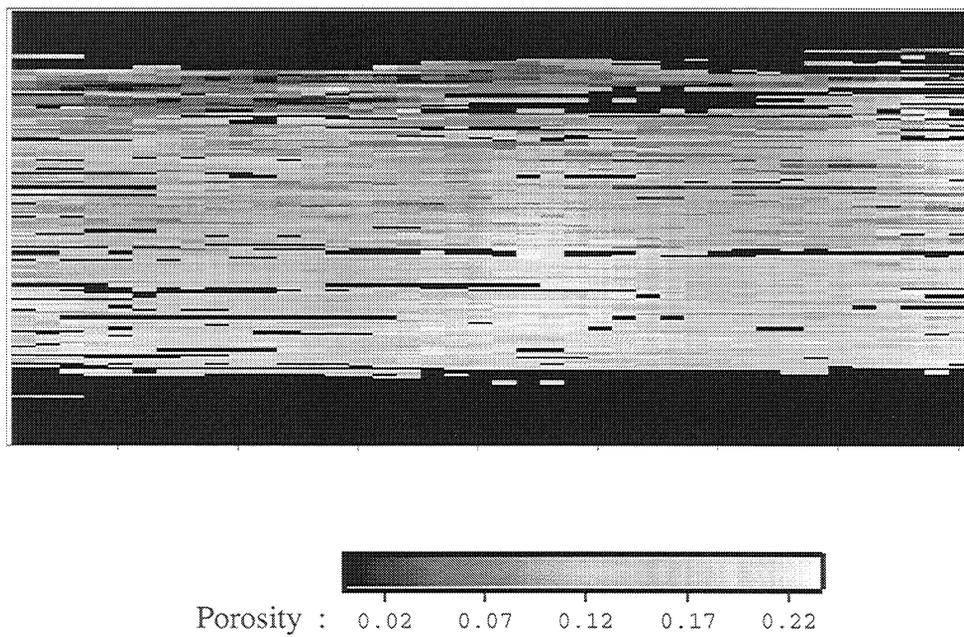


Figure 35: North-South Cross Section of porosity distribution of the Self Unit data

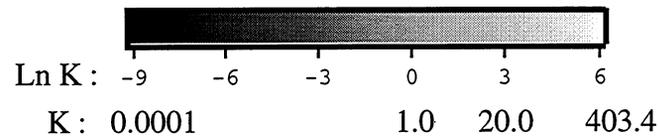
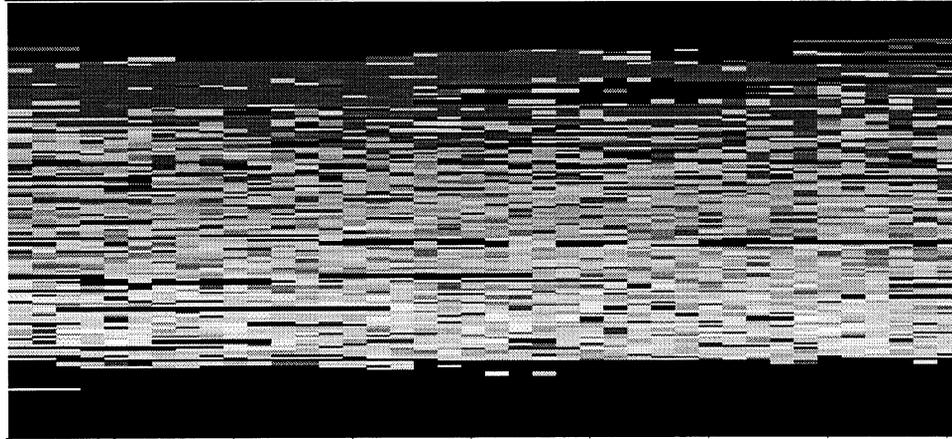


Figure 36: North-South Cross Section of permeability distribution of the Self Unit data

From these figures we can see clearly how the DGIs are varied from top to bottom with shale on the top and bottom borders. This distribution is followed very well by the petrophysical properties distribution as shown in Figure 35 and Figure 36. The increase in porosity and permeability from top to bottom are matched very well with the field observation, i.e., from well log and core data.

#### 4.6.2 Carbonate Reservoir

The data for this case study is obtained from the North Robertson Unit, West Texas.<sup>7</sup> The geological unit used by the geologist in describing the reservoir is rock type where the petrophysical properties depend mainly on the pore size. From co-simulation point of view, the rock type is considered as the facies, or a geological unit that can still be represented by an indicator value. Therefore, the term facies will be replaced by rock type in this case.

There are 8 rock types defined in the system, namely rock types 1 through 8, but only 4 of them are reservoir rocks, i.e., Rock Type 1, 2, 3, and 5. The other 4 rock types, i.e., Rock Type 4, 6, 7, and 8, are classified as non reservoir rock. In this simulation, all non reservoir rocks are

combined and renamed as Rock Type 4. Thus, for the simulation purposes, only 5 rock types are simulated, namely Rock Type 1 through 5.

The overall dimension of the field is 1 sq. mile with total formation thickness of about 1600 ft. For simulation purposes, this system is divided into 11 layers. Each layer is separated by a no flow unit defined by the geologist. In each layer, the reservoir is divided into  $99 \times 99 \times nz$  grid blocks, where  $nz$  is the vertical thickness of each layer. Using this configuration, the number of grid block in each unit range from 700,000 to 1,600,000.

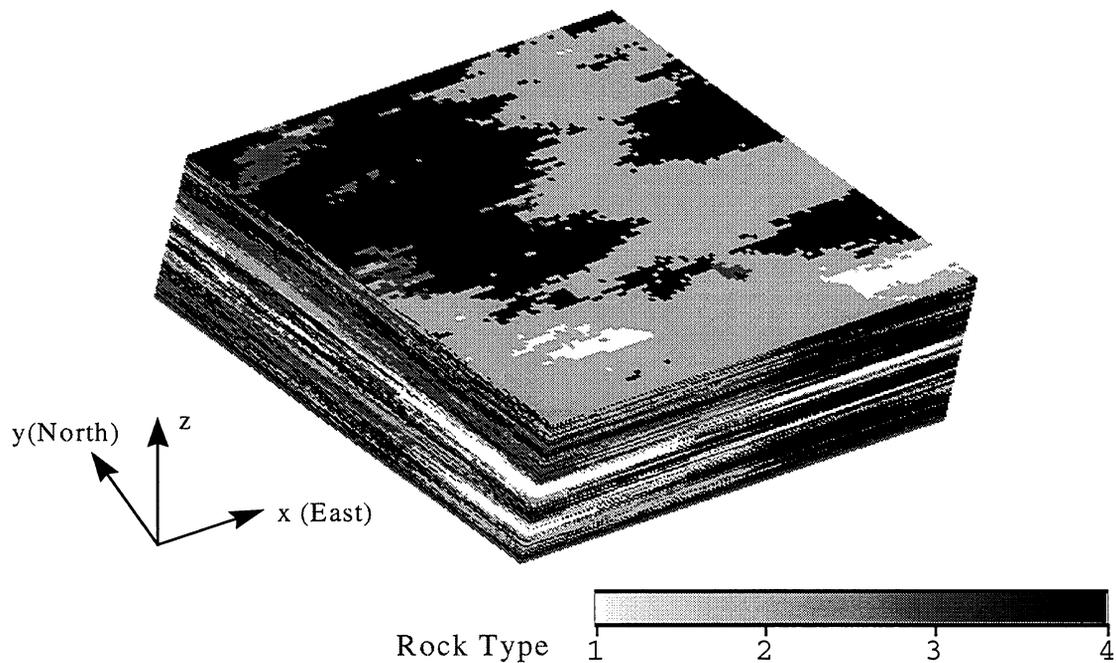


Figure 37: 3D view of rock type distribution

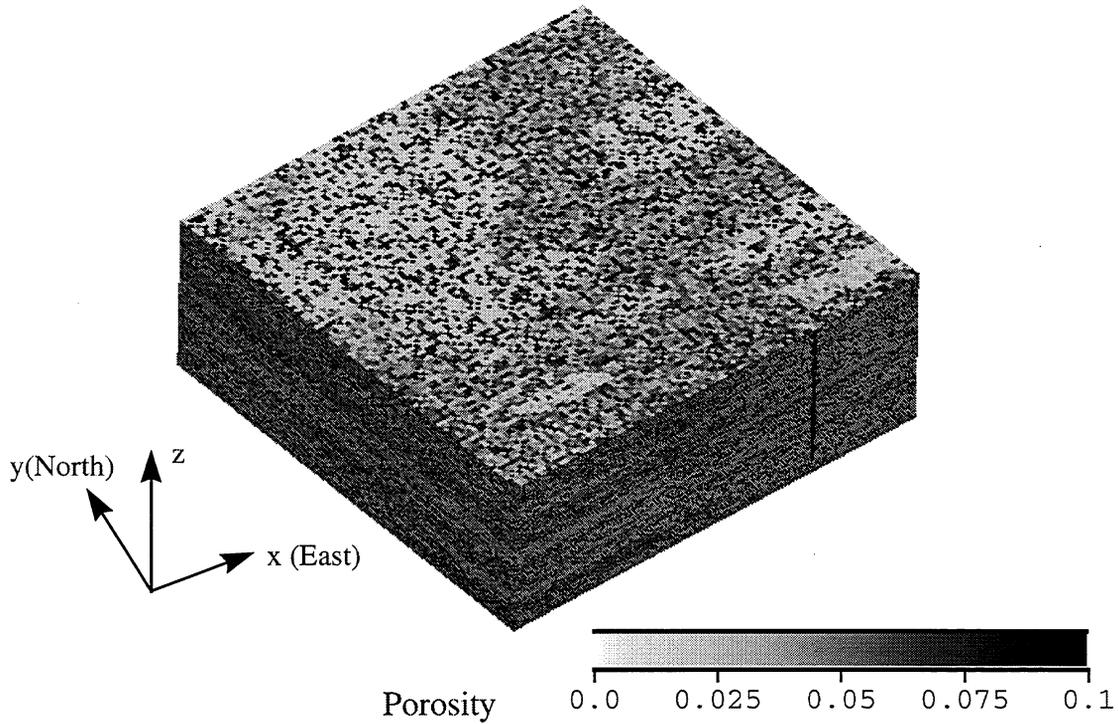


Figure 38: 3D view of porosity distribution

Figure 37 through Figure 39 show the 3D view of the Rock Type distribution with its associated porosity and permeability distribution. The number of grid blocks used in this simulation is 715,473 ( $99 \times 99 \times 73$ ) with the size of each grid block is  $55 \times 55 \times 1 \text{ ft}^3$ . As in sandstone case, we can also observe that petrophysical properties are consistent with the underlying geological description. In this particular example, Rock Type 5 is not part of the conditioning data. Therefore, none of the grid block has the property of Rock Type 5.

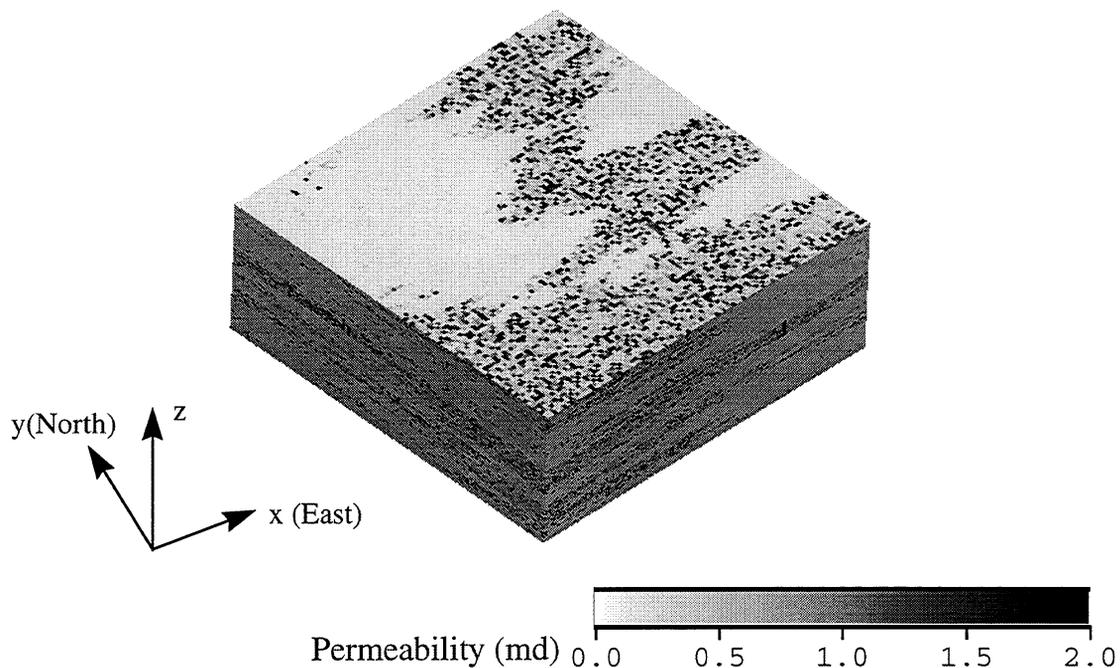


Figure 39: 3D view of permeability distribution

#### 4.7 Conclusions

Several conclusions that can be taken from this work are as follows.

1. Most of the currently available geostatistical tools are limited to single attribute simulation only. The number of techniques to jointly simulate multiple variable that are dependent to each other is very limited.
2. A new procedure to jointly simulate inter-related variables is proposed in this work. The development of this technique is driven by the practical application, i.e., the two stage approach that is felt to be inefficient.
3. The key technique developed here is the use of simultaneous sequential Gaussian simulation and conditional distribution principal. Additionally, truncated Gaussian technique is also used for indicator variable.

4. Theoretically, the technique can be used to simulate any number of variables, but the implementation performed in this work is limited to three variables only, i.e., facies, porosity, and permeability.
5. The technique honors the spatial relationship of each variable as well as local relationship among variables.
6. Several case studies have shown that the generated petrophysical properties consistent with the underlying geological description. The program is found to be very efficient and practical.
7. A new procedure in solving the horizontal variable problem is discussed. A solution based on the frequency analysis is proposed.
8. A modification in the way data pairs are selected is proposed for calculating indicator variogram.
9. A complete package of the co-simulation program, that includes the pre-simulation, simulation, and post-simulation facilities, as a PC software has been developed and tested.

## **5. Reservoir Characterization Using Simulated Annealing With Dynamic Constraints**

### **5.1 Introduction**

A petroleum reservoir is intrinsically deterministic; i.e., it has potentially measurable, deterministic properties and features. The question therefore arises as to why we should use stochastic methods in studying petroleum reservoirs.

Some of the major reasons we resort to stochastic methods for describing a deterministic reservoir are:<sup>34</sup>

- the incompleteness of information about the reservoir at all scales
- the very complex spatial disposition of reservoir facies

- the difficult-to-capture rock property variability and variability structure with spatial position and direction
- the unknown relationships between property values and the volume of rock over which we are averaging (the 'scale' problem -- of which more will be said later).

For these reasons the use of stochastic methods in reservoir characterization has gained widespread acceptance as a promising methodology for enhancing the engineer's ability to describe reservoir properties. This area of specialization in petroleum reservoir engineering represents some of the most recent technology being applied in the on-going effort to better manage petroleum reservoirs by creating models which are more representative of, or which more closely approximate, the complex reality of the reservoir. As we shall show, there are other benefits in using a stochastic approach, related to quantifying the uncertainty in projections of reservoir performance and parameter statistics.

## **5.2 Theory**

### **5.2.1 *Inverse Problem***

Reservoir characterization is an inverse problem, in that the objective is to develop descriptions of the spatial distribution of those reservoir parameters which dominate its flow behavior -- such as permeability and porosity -- from measurements of the parameters which characterize that flow behavior, namely, pressures and rates. Additionally, 'hard' data -- typically punctual observations or measurements of the porosity and permeability at the wellbores -- are used to condition the description process.<sup>35</sup> In the geostatistical approach to reservoir modeling, assumptions about the spatial statistics of the reservoir parameters in which we are interested are used as a means of guiding the resulting description. The spatial statistics are estimated by understanding the geologic processes responsible for the deposition of the reservoir, in conjunction with actual measurements of the reservoir parameters within the field.

### 5.2.2 Variography

The basic geostatistical tool used in describing the spatial correlation and (statistical) variability of reservoir properties is the *semi-variogram*, which, for simplicity, is usually referred to as the variogram. The variogram is used to capture the degree of spatial variability of a reservoir property. It is defined as the variance of the increment and is mathematically defined as

$$\gamma(\bar{h}) = \frac{1}{2} E \left\{ \left[ V(\bar{x}) - V(\bar{x} + \bar{h}) \right]^2 \right\} \quad (6)$$

where  $\bar{h}$  is the lag distance

$E\{ \}$  is the expected value operator and

$V(\bar{x}), V(\bar{x} + \bar{h})$  are the variable values at locations  $(\bar{x})$  and  $(\bar{x} + \bar{h})$  respectively.

There are then geostatistical methods, which utilize the variogram in creating a probabilistic model of the reservoir.

### 5.2.3 Stochastic Simulation vs Estimation

There are two basic approaches in the stochastic methodology and these relate to whether our objective is one of *estimation* or whether we are concerned with *simulation*.<sup>36, 37</sup> Estimation of reservoir properties refers to techniques that use linear interpolation to populate a reservoir grid with values. Conditioning data are used together with spatial statistics of the reservoir parameters to develop the estimate of the reservoir model. The geostatistical methodology of **kriging** is usually used for this.<sup>38, 39</sup> It should be noted that a single description, based on a given set of conditioning data and parameter statistics, results. Additionally, the resulting description is substantially ‘smoother’ than the true reservoir being modeled, since kriging utilizes the principle of minimum variance, unbiased estimate (MVUE) and so the error variance of the krigged description is less than that of the true reservoir.<sup>6</sup> Generally, and depending on the amount of conditioning data available, the image “resembles” the true image by capturing the highs and lows. In another sense, the kriging approach acts as a high-frequency filter by capturing the low frequency features of the image through the averaging process. It thus represents the best *estimate* of the true model, but on a finer scale, much of the detail in the description, i.e., the

spatial variability of the model, is lost. The conditioning values however are honored at their locations.

On the other hand, **stochastic** (conditional) **simulation** honors -- in addition to the conditioning data, if available -- the *a priori* reservoir parameter statistics. Hence the assumed variability of reservoir properties is preserved. In addition, multiple images may be generated, giving a better sense of the uncertainty in the reservoir description. Thus if the objective of the exercise is to develop models for flow simulation (for forecasting performance), then stochastic simulation is the methodology of choice.

While it is not the best estimator of the simulation variable (in fact the simulation variance is twice the kriging variance),<sup>36</sup> our preferred methodology is *simulation* for the purpose of reservoir characterization. Some of the advantages (of conditional simulation methods over interpolation algorithms) may be summarized as:<sup>6</sup>

1. They honor the sample data and retain the original distribution, while, as was mentioned, conventional kriging techniques reduce the variance of the distribution.
2. They honor the spatial relationships established using the sample data, unlike conventional kriging, which does not precisely honor these relationships.
3. They can create any number of equiprobable images (or realizations) of the region being studied. As alluded to above, this is a very useful feature of simulation methods, in that the variability among the different realizations gives an indication of the degree of uncertainty associated with the description. Traditional kriging allows estimation of the error variance, but may not be able to identify a local uncertainty.

In summary, the simulated values are almost certainly wrong in the sense that they are unlikely to match the unknown permeability (or porosity) values, but unlike the estimated fields, the simulated fields have the advantage of being *possibly* correct.<sup>37</sup> For these reasons, this is the methodology that will be used in our work.

#### ***5.2.4 Stochastic Simulation Methodology Selected***

Simulated annealing (SA) was selected as our methodology of choice for the following reasons:

- It is a non-parametric approach, in that, unlike the popular Gaussian-based approaches such as sequential Gaussian simulation, there is no assumption of any particular distribution function model (such as Gaussian) for the reservoir variable(s).
- This approach is flexible in ability to incorporate multiple constraints. Most other stochastic simulation approaches are more restrictive in the use of constraints.
- The methodology is robust, having a theoretical basis for asymptotic convergence. In addition, it is a well-established procedure, having been successfully applied as an approximation approach in a diverse range of disciplines such as the famous traveling salesman problem and computer chip placement on circuit boards.<sup>7</sup>

#### ***5.2.5 Overview of Algorithm***

SA is a combinatorial optimization algorithm. For practical applications it is used as an approximation algorithm so that a solution may be obtained in a reasonable timeframe. The goal of the algorithm is the minimization (or maximization) of an objective function. The traditional application of simulated annealing to reservoir characterization utilizes an objective function comprised of a mis-match of a measure of the reservoir's spatial statistics, captured by the variogram. Hence, we minimize the difference between the model variogram and the experimental variogram. Additionally, there may be conditioning data, which are honored and help to stabilize the problem. Thus a one-part objective function is used.<sup>41</sup>

In our modified approach, we use a two-part objective function in which the first part is the same as before, but a second constraint is introduced which involves the mis-match between observed flowing bottomhole pressures (BHPs) and BHPs flow simulated on the currently-developed reservoir description. In effect, automatic history-matching is done.<sup>42</sup> Special techniques are employed which cut down on the flow simulation time of the algorithm. These include:

- The use of representative upscaling for the flow simulation part of the objective function. The upscaling technique used enables approximate BHP matching even during the transient flow period. Thus flow simulation CPU time is reduced by using an upscaled grid.
- The flow simulation is performed in Laplace space using the Laplace Transform Finite Difference (LTFD) approach.<sup>10</sup> This effectively removes the time dependence from the partial differential equation and so reduces the truncation error of an equivalent conventional finite difference scheme. Moreover, the LTFD method yields a stable non-increasing material balance error. It should be noted however that the use of this approach limits the simulation to primary, single-phase oil production.
- Variogram pre-selection: the variogram component of the objective function is determined first. Should this objective function change be acceptable, only then is the flow simulation part of the objective function determined -- and hence the need for unnecessary flow simulation is substantially reduced.

### 5.3 Data Considerations

#### 5.3.1 Constraints and Conditioning Data

The constraints imposed on the simulation process may include a prior distribution function of the simulated variable, spatial relationships in various directions and the geometry of geologic shapes. In addition, conditioning data may be used to further assist in the development of a reservoir model, which better approximates the truth. These data are usually actual field measurements. The differences among the resulting equiprobable images -- as will be discussed below -- become smaller as more conditioning and constraining are applied. It is therefore advantageous to maximize these, once they are relevant and significant.

#### 5.3.2 Static and Dynamic Data

An important distinction that must be made is that between *static* and *dynamic* data. The two data types previously cited -- permeability and porosity -- represent static data; i.e., their values

do not (significantly) change over time. This facilitates their use as conditioning data. On the other hand, production data is an example of dynamic data, because it is a function of time. This complicates its use for conditioning, because it now becomes necessary to solve an *inverse problem*<sup>35, 44, 45</sup> (which is defined below). This is unfortunate, since production and pressure data are reservoir characteristics of primary interest in reservoir studies. Other than the economic necessity to predict present and future reservoir performance, there are various reasons for this. One reason is that they are a direct solution to the *forward problem* (also discussed below), that refers to determining the response of a reservoir from a definition of the reservoir parameters. Another is that they are representative of reservoir properties at the required scale level, i.e., the gridblock level, so no upscaling is necessary.

### 5.3.3 Scale

In reality a reservoir is a continuum of heterogeneities. That is, the variability in the reservoir properties does not normally change abruptly from one point to another (except of course for discontinuities such as faults and unconformities which, by their very names, emphasize that fact that they are sudden and exceptional events, and even these, at a micro-scale, are gradual). It is practically impossible however to be able to model such a system, firstly because available data are not complete and, secondly if they were, the model would be too unwieldy to be useful. Also, as Bear<sup>46</sup> and Lake<sup>47</sup> have pointed out, while in principle -- given all the relevant information -- we may formulate a mathematical system for all flow channels in the medium, the extremely tortuous phase boundaries in such a system prevents us from solving species conservation equations in a local sense for even the simplest microscopic permeable media geometry. To resolve these problems we resort to the concept of a *representative elementary volume (REV)*. This can be defined a volume below which local fluctuations in some primary property of the permeable medium, usually porosity, becomes large. It is a volume which is large with respect to the pore dimensions of the stationary phase but small compared to the dimensions of the permeable medium.<sup>43, 44</sup>

It should be noted however, that even these *REVs* are at a macroscopic (core level) scale; i.e. of the order of 1-100 cm., which is normally still too small for a full-field flow simulation run. For such applications therefore, the reservoir description is *discretized* whereby a 1-, 2- or 3-D grid is

overlain on the reservoir system. The gridblocks are then taken to represent homogenized units in which the reservoir properties are averaged out over the block volume. It can be easily appreciated that the finer the gridding applied, the closer the model becomes to the ‘true’ reservoir system. Of course the model size becomes larger with decreasing gridblock size (and a consequent increasing in the number of gridblocks). We therefore seek a compromise: on one hand we would like as fine a grid as possible but on the other hand we are constrained by the size of the model we are able to handle with the computer resources available. However, changing the scale at which we discretize the reservoir model means changing the degree of homogenization. This tends to change the behavior of the system, especially under transient conditions and even the later time behavior is not identical.<sup>b</sup> The issue of scale and the ability to change scale effectively is therefore a very important one.

## 5.4 The Approach

### 5.4.1 Flow Simulation

As stated previously, for full-field studies, the incorporation of dynamic constraints has been stymied because of the prohibitively-high cost in computing time required for the flow simulation phase. The major reason for this is the necessity to repeat the calculations in time-stepping between the initial time and the final flow period. It seems reasonable therefore, that the ability to utilize only *one* timestep over the same period would significantly reduce the computing time requirements.

This suggested an approach that could be used. It is based on the Laplace Transform Finite Difference (LTFD) method,<sup>43</sup> a procedure in which the problem is solved in Laplace space. This procedure removes the timestep size constraint as exists for an IMPES-type method, because the time variable is transformed into the Laplace variable, ' $u$ '. Thus it is semi-analytical in time and numerical in space. Hence, stability is conserved without having the timestep size limitation of explicit methods. Instead, one can use just *one* timestep between the initial time and the time of interest. As regards more implicit methods, e.g., Newton's method, the LTFD method has the advantage in that the calculation of gradients is not required, making it considerably less

---

<sup>b</sup> This latter problem though may be due to the inexactness of current upscaling techniques.

computationally-intensive. It is expected therefore that this approach, in terms of computational time, will be less expensive. Also, by using the flow simulation results in Laplace space, we obviate the need for numerical inversion (via the Stehfest algorithm<sup>51</sup>) to real time and space.

### ***Development of Model***

The diffusivity equation is the basic model for single-phase fluid flow in porous media. It is defined as:

$$\left\{ \nabla \cdot \left[ \frac{[k]}{\mu_o B_o} \nabla p \right] \right\} = \frac{\phi c_t}{B_o} \frac{\partial p}{\partial t} \quad (7)$$

To complete our problem statement, we need to define the initial and boundary conditions.

$$p(\bar{x}, t = 0) = p_{initial}, \quad (8)$$

where  $p_{initial}$  is the reservoir pressure at time,  $t = 0$ .

The most common outer boundary conditions are *no-flow* boundaries. The equations for these are,

$$\left. \frac{\partial p}{\partial x} \right|_{x=x_e} = 0 \quad (9)$$

where  $x_e$  represents the outer boundaries.

The inner boundary conditions are defined by the sink/source equation. These may be defined separately or included in the main *pde*. We opted to include the inner boundary conditions in the *pde*. Thus Eq. 4 becomes,

$$\left\{ \nabla \cdot \left[ \frac{[k]}{\mu_o B_o} \nabla p \right] \right\} = \frac{\phi c_t}{B_o} \frac{\partial p}{\partial t} + q, \quad (10)$$

where  $q$  is the rate. In solving this equation, the use of a particular convention for defining the units of the variables requires the definition of a conversion constant. This is thus included in the

equation. Further, to simplify calculations, we consider pressure changes instead of pressures in the equation. The pressure change is defined as,

$$\Delta p = p_{initial} - p, \quad (11)$$

and so the *pde* becomes,

$$C_1 \left\{ \nabla \cdot \left[ \frac{[k]}{\mu_o B_o} \nabla \Delta p \right] \right\} = \frac{\phi c_t}{B_o} \frac{\partial \Delta p}{\partial t} + q \quad (12)$$

where  $C_1$  is a conversion constant equal to  $0.006328 (=1.127 \times 10^{-3} \times 5.615)$  for the common oilfield units used and time in days. The units for the rate,  $q$ , are cubic feet per day per cubic foot or simply  $\text{day}^{-34}$ . To use the field unit of barrels per day per cubic foot, we would have to multiply the rate term by 5.615, or alternatively, we use  $1.127 \times 10^{-3}$  for  $C_1$  and divide the other terms by 5.615. The initial conditions simplify to,

$$\Delta p(\bar{x}, t = 0) = 0 \quad (13)$$

The simplifying assumptions are:

- single-phase, slightly-compressible fluid of constant compressibility
- formation volume factor is constant (=1)
- system compressibility is constant
- fluid viscosity is constant
- gravity term neglected,  $z$ -direction corresponds to vertical depth.

On the basis of these assumptions, the *pde* simplifies to,

$$C_1 \left\{ \nabla \cdot \left[ \frac{[k]}{\mu_o} \nabla \Delta p \right] \right\} = \phi c_t \frac{\partial \Delta p}{\partial t} + q \quad (14)$$

This problem can now be considered as a linear *pde* and we can thus take *Laplace transforms* as suggested by Moridis et al.<sup>43</sup> Thus we get,

$$L\left[C_1\left\{\nabla\cdot\left[\frac{[k]}{\mu_o}\nabla\Delta p\right]\right\}\right]=L\left[\phi c_t\frac{\partial\Delta p}{\partial t}+q\right] \quad (15)$$

This simplifies to,

$$C_1\left\{\nabla\cdot\left[\frac{[k]}{\mu_o}\nabla\overline{\Delta p}\right]\right\}=\phi c_t u\overline{\Delta p}+\overline{q} \quad (16)$$

where the ‘bar’ above a parameter indicates its Laplace transform and  $u$  is the Laplace variable. For constant rate, the equation becomes,

$$C_1\left\{\nabla\cdot\left[\frac{[k]}{\mu_o}\nabla\overline{\Delta p}\right]\right\}=\phi c_t u\overline{\Delta p}+\frac{q}{u} \quad (17)$$

Note that the time dependence is eliminated by this transform.

### ***Advantages of the LTFD Approach***

The principal advantage of the LTFD approach concerns the removal of the time dependence from the *pde* by use of the Laplace transform. The time-dependent discretization is the major source of instability and error in terms of its approximation in numerical models. For example, for a real-time numerical approximation via an implicit scheme, the truncation error is  $O((\Delta x)^2 + \Delta t)$ , so the time-based error dominates, limiting the timestep size. Using the LTFD approach means that we are not constrained to time-stepping via small, discrete time intervals from an initial state to the time of interest; we can use just *one* step without the penalty of instability as with explicit methods or truncation error as with implicit methods. Moreover, the LTFD method yields a stable non-increasing material balance error in addition to a more accurate solution than the conventional finite difference schemes.<sup>43</sup> This makes this approach well-suited for use in the SA algorithm since we are concerned with minimizing the flow simulation requirements. Thus in our approach we define a fixed number of time intervals for use in

matching the pressures and perform flow simulations for those times only. This theoretically gives us the freedom of using any and all production data and does not limit the simulation to the consideration of well test data only.

### ***Disadvantages of the LTFD Approach***

The first and obvious disadvantage of this approach is the fact that it is currently limited to a single-phase model because of the requirement of linearity for the Laplace transform. Fortunately, this does not invalidate the approach since there would usually be some of this type of data especially earlier in the life of the field.

It may be inferred that the coefficient matrix,  $A$ , is non-singular by being strictly diagonally-dominant,

$$|a_{i,i}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{i,j}| \quad \forall i = 1, \dots, n \quad (18)$$

This diagonal dominance can be easily seen since the absolute value of the main diagonal term is always greater than the sum of the absolute values of corresponding off-diagonal terms by  $\xi u$ . These terms are defined as,

$$\xi = \frac{\phi C_t \mu}{C_1} \text{ and } u = \text{Laplace variable} \quad (19a-b)$$

There is an inverse relationship between the Laplace variable,  $u$ , and time. Thus for longer times, the matrix moves away from strict diagonal dominance as the difference defined by Eq. 17 gets smaller. This means that the matrix becomes increasingly singular and the iterative solution takes many more iterations to converge. In fact in some instances, we do *not* get convergence to the required tolerance (a convergence tolerance of  $10^{-12}$  was normally used) within the maximum number of iterations set (we used a value of 500 for the maximum number of iterations).

This problem is most obvious when simulating relatively large grids (of the order of 10,000 gridblocks). For the smaller upscaled grids used (around 400 gridblocks) in the modified SA algorithm this is not a problem.

The *ECLIPSE-100* Black Oil simulator<sup>54</sup> is a commercial simulator with established credentials. Its accuracy has been validated, e.g. by Sagar<sup>55</sup> who compared its results with those of the analytical solution for a homogeneous reservoir. Additionally, its use is wide-spread in the petroleum industry and so we use *ECLIPSE-100* as a benchmark for validating the results obtained with the LTFD simulator using the same input dataset. A heterogeneous 100x100-gridblock model with 9 wells on production was used and the results presented graphically below. As can be seen in Figure 40, there is excellent agreement in the results within the time range considered (1 to 1000 days). It should be mentioned that similar agreement was obtained for other datasets.

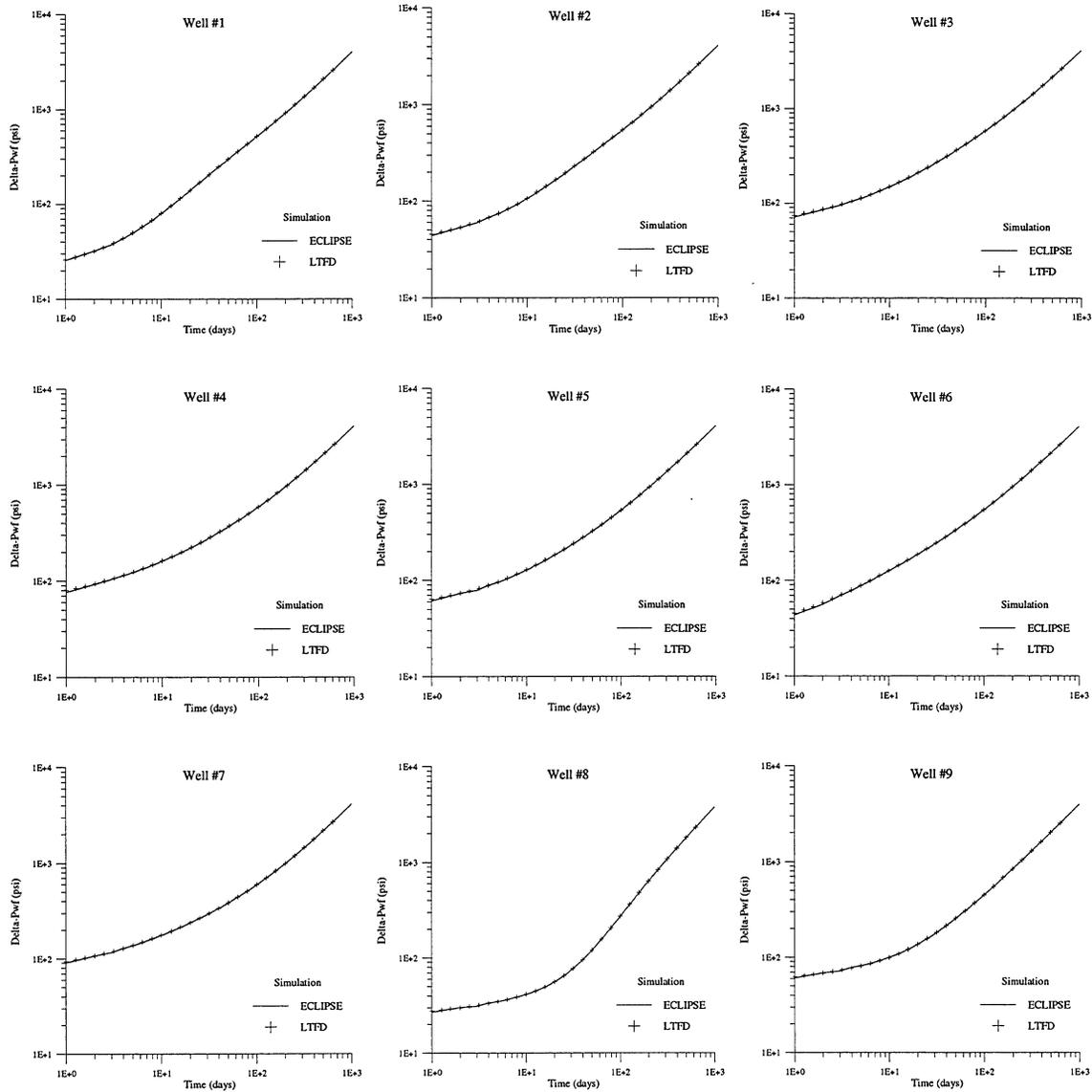


Figure 40: Comparison of *ECLIPSE* and LTFD simulation results

### 5.4.2 Upscaling of Grid Balance

In addition, the smaller the number of grids used for the flow simulation, the faster is the execution time. Thus another way of accelerating the execution time of the algorithm would be to perform the flow simulation on an *upscaled* grid. We have to ensure however, that the flow simulation performance of the coarser grid is representative of the “true” or fine scale grid within the time range used in the objective function of the simulated annealing algorithm. We will

present a modified upscaling approach, which does a reasonable job of matching the flow simulation performance between scales.

### ***Development of Procedure***

We have ascertained that the errors in the pressure match between scales from various conventional upscaling approaches for permeability were very similar.<sup>42</sup> We therefore decided to use the simplest methodology -- the geometric averaging approach -- and modify it in an attempt to better match the pressures. As shown, the major errors were obtained at early time, corresponding to the time at which the pressure transient is influenced by the near-well region,<sup>52, 53</sup> and, as was observed, dominated by the wellblock permeability. It seemed reasonable therefore to concentrate on improving the upscaling approach for this region to be able to better match the pressure behavior at early time. Further to this, the upscaling procedure should favor the fine scale wellblock permeability value. By numerical experiments, we determined that an appropriate procedure was to assign a weighted arithmetic average of the fine scale wellblock permeability and the mean of the remaining fine scale values making up the coarse scale wellblock permeability value. Thus,

$$\bar{k} = w_1 \cdot k_{wb} + w_2 \cdot \frac{1}{n-1} \sum_{\substack{i=1 \\ i \neq wb}}^n k_i \quad (20)$$

where  $w_1$  is the weighting applied to the fine scale wellblock permeability,  $k_{wb}$  is the fine scale wellblock permeability value,  $w_2$  (equal to  $1-w_1$ ) is the weighting applied to the mean permeability of the other fine scale, near-well gridblocks which are included in the coarse scale wellblock and  $\frac{1}{n-1} \sum_{\substack{i=1 \\ i \neq wb}}^n k_i$  represents that mean where  $n$  is the upscaling ratio. An appropriate

value for  $w_1$  was empirically determined to be 0.9 (thus  $w_2=0.1$ ). The upscaling procedure therefore consists of using conventional geometric averaging to upscale all coarse scale blocks except the coarse scale wellblocks. This equation is,

$$\bar{k} = \sqrt[n]{\prod_{i=1}^n k_i} \quad (21)$$

For wellblocks, Eq. 20 is used.

### **Analysis of Results**

Using the modified geometric averaging procedure, the image of the coarse scale grid is not much different from the conventional approach (since only the coarse scale wellblocks are different). However, there is a significant improvement in the behavior of the errors in the pressure matching between scales as seen in Figure 41 and Figure 42. Note that, unlike in Figure 41, the scales of the vertical axes in Figure 42 are the same to facilitate comparisons.

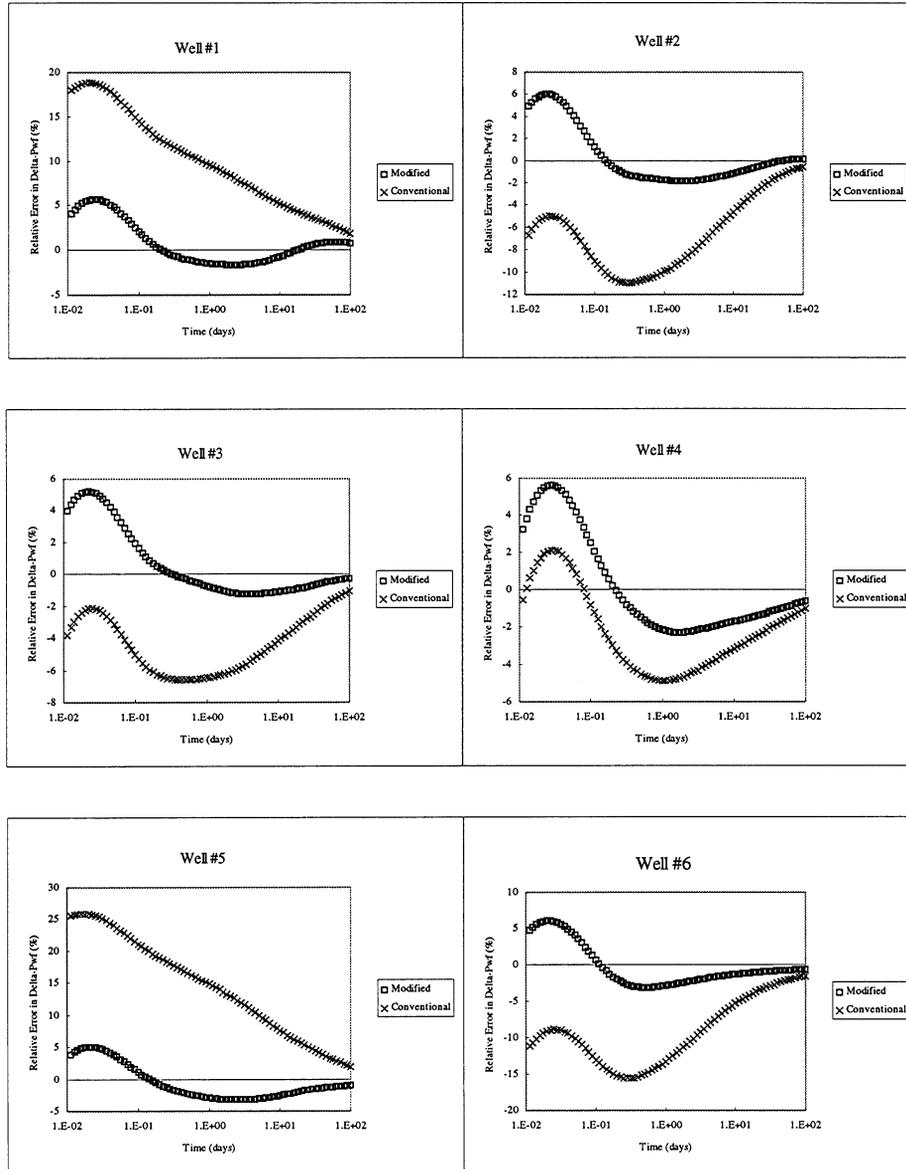


Figure 41: Comparison of conventional and modified geometric averaging upscaling results

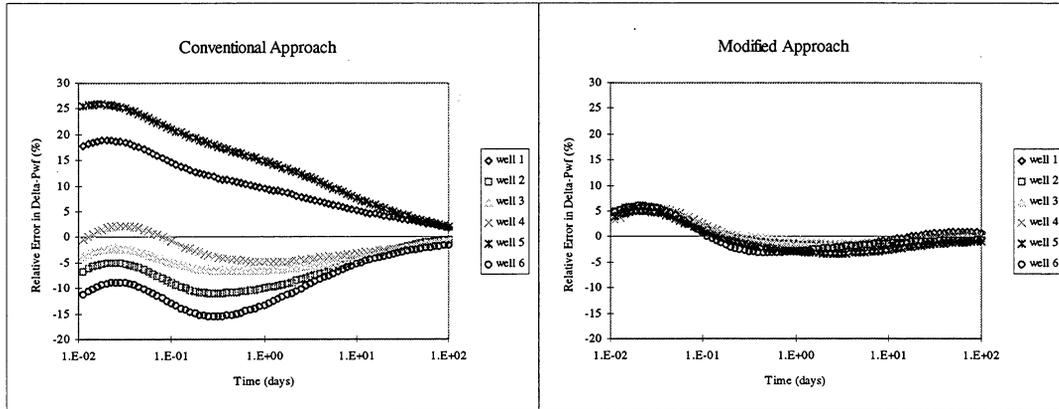


Figure 42: Summarized comparison of conventional and modified geometric averaging upscaling results

In all cases, except for well #4, the absolute errors are smaller with the modified approach for the entire time range. In the case of well #4, even though the errors are initially somewhat larger for the modified approach, they converge faster to zero. After 1 day in *all* cases, the modified approach produces errors that have a maximum absolute value of less than 5%. We can therefore consider using this upscaling approach if we constrain the matching of the pressures to a minimum time of one day.

### Limitations of Methodology

It was observed however, that this modified approach did not give consistent results. As Figure 43a shows, in some cases the errors remained relatively high with this approach.

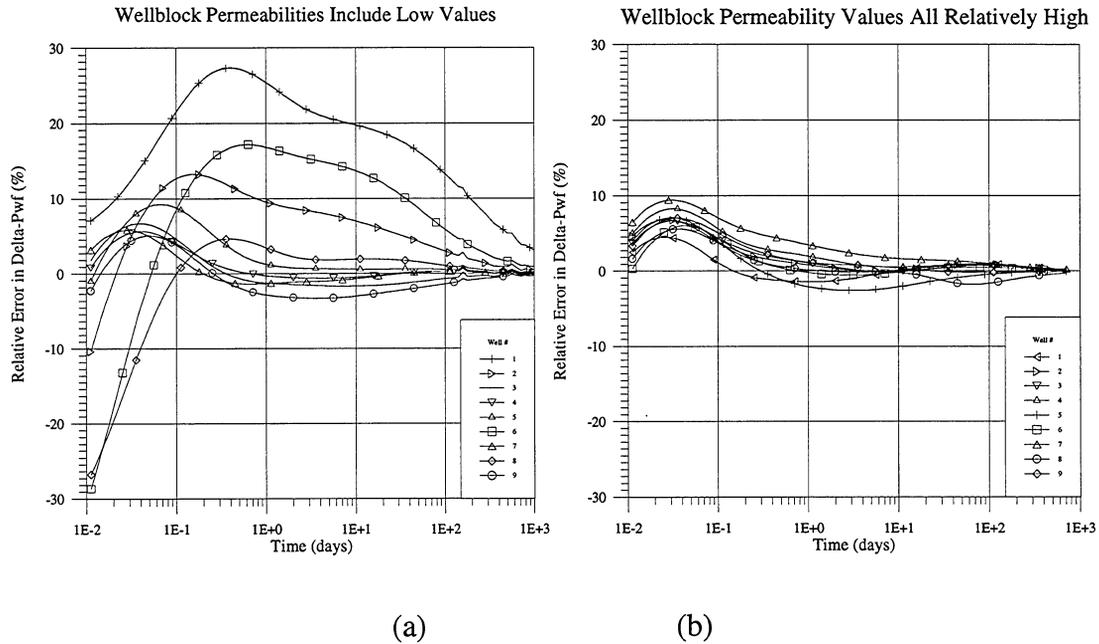


Figure 43: Modified geometric averaging upscaling performance related to the fine scale permeability contrast at the wellblock

As can be seen however, some of the wells were “well-behaved” while others were not. For example, well 9 displays low errors, but those for well 1 are high. It was determined that the magnitude of the fine scale wellblock permeability determines how good the modified geometric average upscaling approach works. This may be explained by considering that, when the contrast between the wellblock permeability and those of the nearby gridblocks is large, upscaling will be dominated by the permeability of the other blocks and so skewed away from the wellblock permeability. This will magnify the difference between the pressure responses observed at the fine and coarse scales.

well 1					
8.65	5.63	9.20	16.27	9.84	
10.62	1.45	3.86	8.31	10.12	
9.76	5.88	1.35	1.13	6.41	
12.51	10.53	2.18	9.23	10.92	
12.73	8.43	4.64	8.88	11.76	

Table 7: Fine scale wellblock permeability in high contrast to surrounding permeability values

well 9				
42.31	35.22	23.32	24.98	26.26
43.69	39.81	31.56	22.96	22.23
48.46	42.00	30.59	25.90	21.59
58.83	36.66	31.65	18.71	24.28
47.94	44.75	37.20	28.91	14.37

Table 8: Fine scale wellblock permeability in low contrast to surrounding permeability values

The above tables show the actual values of the 25 fine scale permeabilities in the near-well gridblocks for wells 1 and 9. The wellblock location and permeability value are denoted by the shaded value at the center. It can be seen that for well 1, the wellblock permeability value is relatively low, when compared to the surrounding values. In the case of well 9 however, this is not so. To verify whether this analysis was valid, another case was flow simulated in which all the fine scale wellblocks were set to permeability values comparable to those of the surrounding gridblocks. Figure 43b shows the results in this case, which supports our analysis.

Here again Figure 43a and b are plotted on the same scale to facilitate comparison. We may thus conclude that the modified geometric average upscaling technique may be used when there is not a high contrast between the fine scale wellblock permeability and the surrounding blocks' values.<sup>c</sup> It may be argued however, that in cases where the errors from the modified geometric averaging upscaling technique are greater, because these are relatively low permeability areas -- and hence areas of lower productivity -- the effects of these errors may be of small importance.

### 5.4.3 Simulated Annealing

The flow simulation is incorporated into the simulated annealing procedure as part of the objective function, in a general sense, as a difference between the observed and estimated flow rates or pressure values at discrete time intervals. Thus the reservoir model will be constrained by the production data in this manner. It is expected that reducing the computation time will result in a viable solution procedure in which production data -- or more generally one or more dynamic constraints -- may be incorporated.

---

<sup>c</sup> It should be noted that here we use a 100x100 grid upscaled to a 20x20 grid (while before we used a 90x90 grid upscaled to an 18x18 grid). This was done for generality.

## 5.5 Results

### 5.5.1 Comparison Between Time and Simulated Images

To evaluate the effectiveness of the new conditional simulation methodology of our modified SA algorithm that includes a dynamic constraint, the approach was to use this new algorithm to generate realizations of permeability fields for which the “truth cases” were known. Thus comparison both visually and in terms of the flow simulation performance could be made.

Three synthetic datasets were used, all 100x100x1 gridblocks. Sequential Gaussian simulation (sGs) was used to generate one set and SA for the other two. In two cases, 9 wells were located in the grid and in the 3rd case, 7 wells were used. All wells were produced at constant but different rates and the well locations coincided with the conditioning data points. Further details on the specification of the data for the synthetic datasets may be found in reference 42.

It should be pointed out that in the modified SA run, a flow simulation time range of 1-100 days was used in all cases. However, in these comparisons, we flow simulate to 150 days, i.e., we are no longer using history matching but are in fact observing how well the *future performance* of the results of both descriptions match that of the truth case.

The results for Dataset #1 are presented below. Similar results were obtained for the other datasets.

This synthetic image was generated using SA. Figure 44 presents a visual comparison of the results. It shows 3 maps: the first, (a), is the truth case, the second, (b), is the modified SA run in which modified geometric averaging upscaling is used to create the upscaled (flow simulation) grid and the third, (c), is the result for the variogram-only objective function SA run. It can be seen that the major features of the truth case description is captured by Figure 44(b) while Figure 44(c) does a relatively poorer job of doing so.

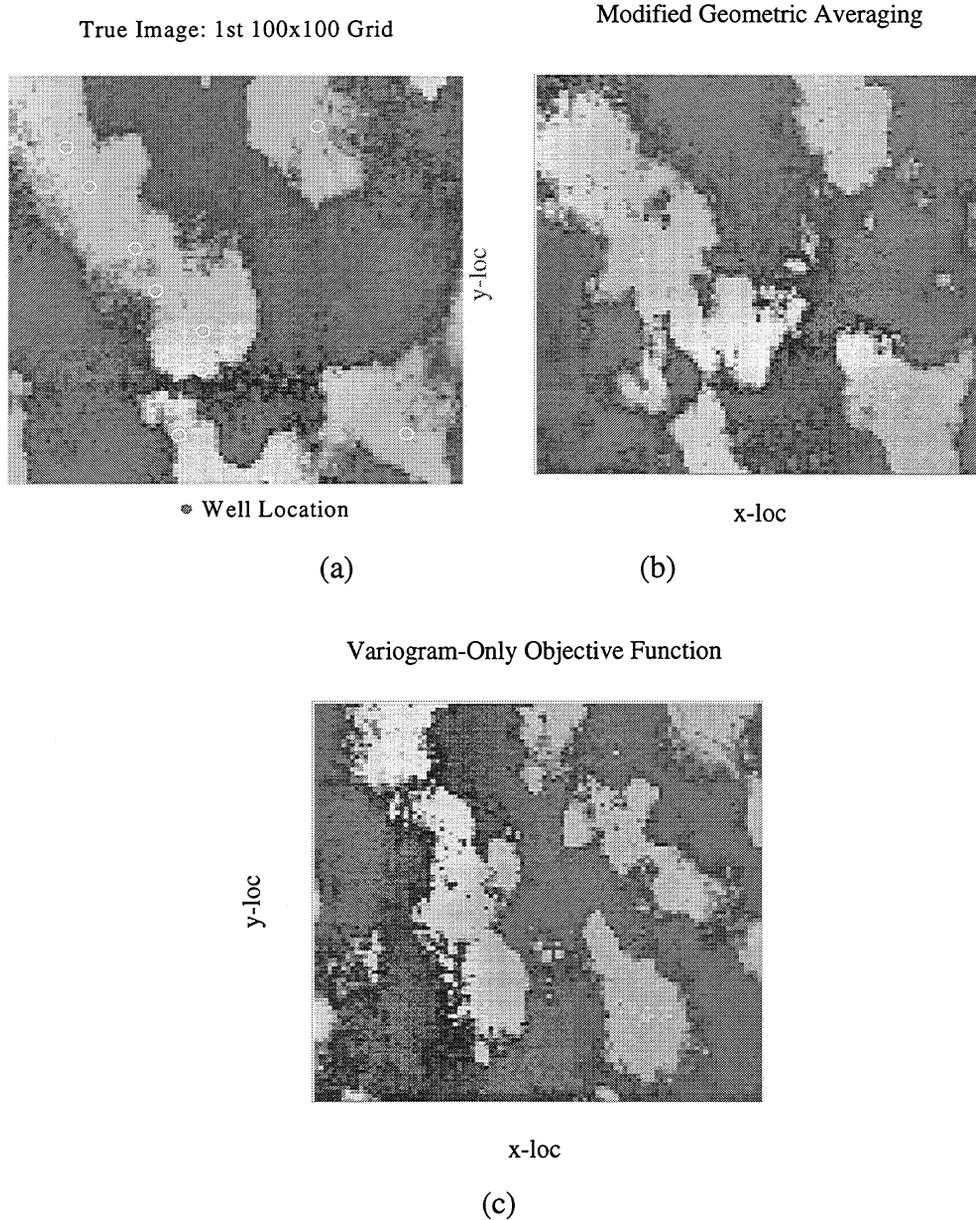


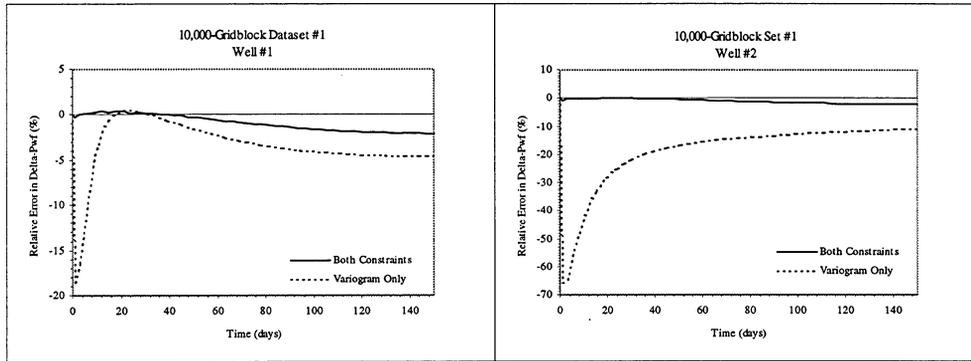
Figure 44: (a) truth case (b) modified sa run results (c) variogram-only sa run results for dataset #1

For this first dataset, nine wells were used for flow simulation. As stated above, all grids were flow simulated under the same set of conditions and the flowing *BHPs* compared. Figure 45(a)-(d) compare the errors in these pressures (actually in the *change* in the flowing *BHPs*) as a function of time for the variogram-only objective function SA run and that of the composite

objective function SA run for four of the wells. The relative errors are calculated as given in Eq. 22 below,

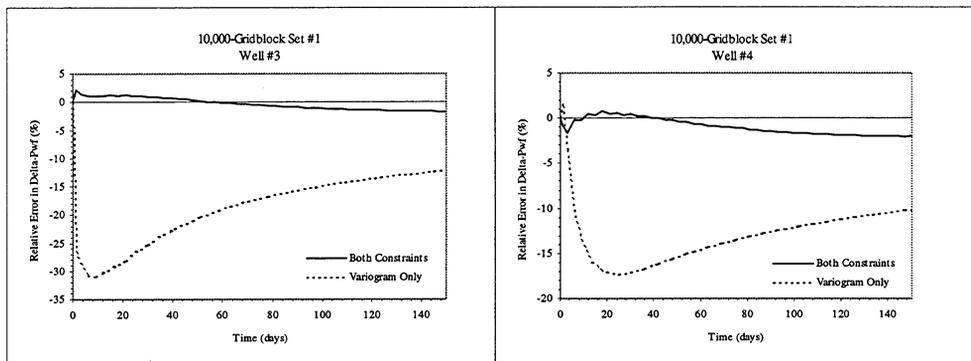
$$\text{Relative Error} = 100 \cdot \frac{\text{'Truth Case' } \Delta p_{wf} - \text{SA Run Result } \Delta p_{wf}}{\text{'Truth Case' } \Delta p_{wf}} \quad (22)$$

These plots clearly show that the errors are significantly smaller when the modified approach is used, even in the predictive part of the time scale (>100 days). One may argue that these results are not surprising since the SA objective function includes a flow simulation constraint that is minimized. However, it must be remembered that the SA objective function uses an *upscaled* grid for flow simulation. Hence, in addition to the validation conducted above, we may again conclude that the upscaling approaches used are adequate. Also Figure 46 shows that the maximum absolute relative errors from the variogram-only objective function results are larger than those of the results from the modified approaches.



(a)

(b)



(c)

(d)

Figure 45: Pressure error comparisons between the modified sa approach and the variogram-only sa approach for dataset #1

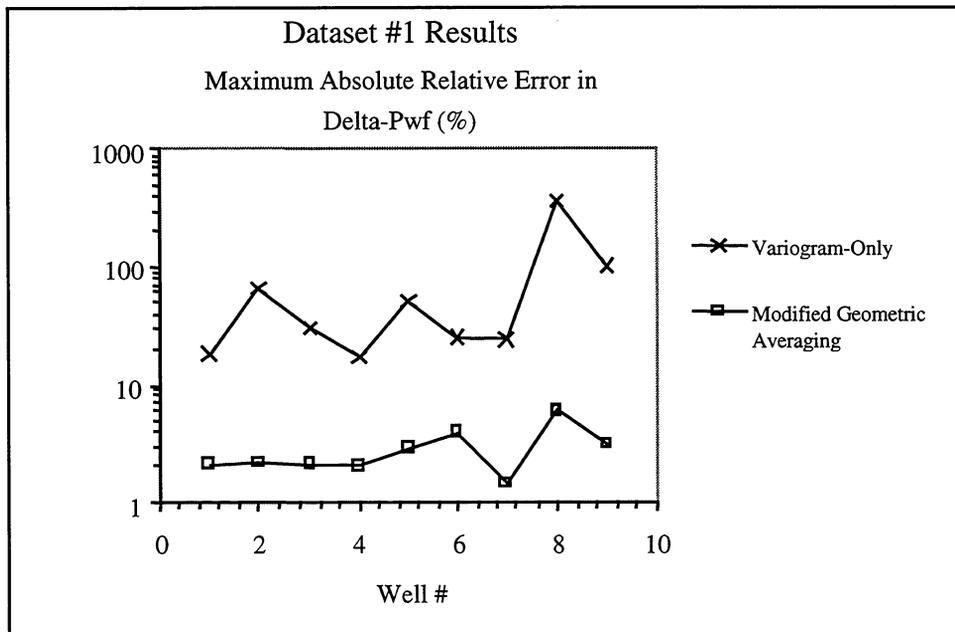


Figure 46: Comparisons of maximum absolute relative errors from variogram-only objective function and the composite objective function using the modified geometric averaging upscaling approach for dataset #1

### 5.5.2 *Sensitivities*

The modified algorithm was run for a case in which a zero weighting was applied to the variogram component yet variogram pre-selection was used. This was done to see how a flow simulation objective function would perform on its own, while still utilizing the variogram to ‘constrain’ the process. Figure 47b shows that there is still quite a good image reproduction when this approach is used, further supporting the conclusion that the flow simulation constraint is a significant contributor to the image-capturing process.

Previously the variogram models were based on the exhaustive dataset. Cases were run in which we attempted to model the variogram using the conditioning data (9 point values of permeability) only. This sparsity of data resulted in a variogram model that is very approximate and may even be inaccurate. Also while an exhaustive dataset allows anisotropy modeling, such a meager dataset is hard-pressed to give even an isotropic model. Figure 47c shows the resulting image obtained for the modified SA approach. It is obvious from these results that although there was insufficient information, we still get a reasonable image of the truth case. However, as shown in Figure 47d, the results for a variogram-only SA run -- using a conditioning data-based (isotropic) variogram model -- are unable to capture the image of the truth case. This last result is dramatic proof of the utility of the flow simulation constraint in the SA process.

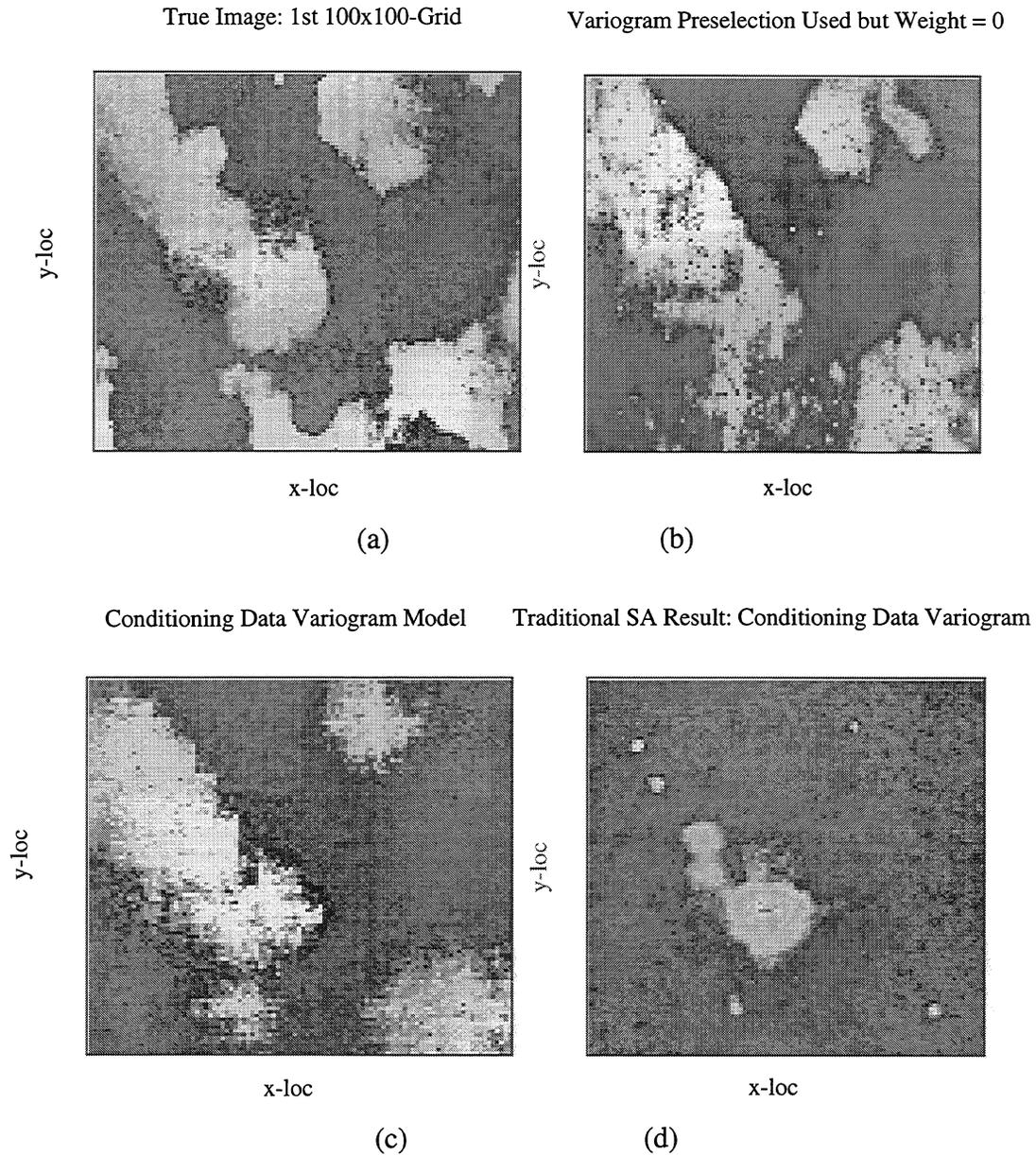


Figure 47: Results from sensitivity tests on variogram component

As noted before, the modified geometric averaging upscaling does not perform adequately when the permeability value of the fine scale wellblock is small and there is significant contrast between it and the permeabilities of the surrounding gridblocks. To illustrate this, a case was run in which the wells were located in both high and low permeability gridblocks. This was done by a uniform spacing of the well locations within the reservoir grid. Figure 48 shows that the results

obtained from this modified SA run do not match the truth case as well as the modified SA run results presented in Figure 44 above.

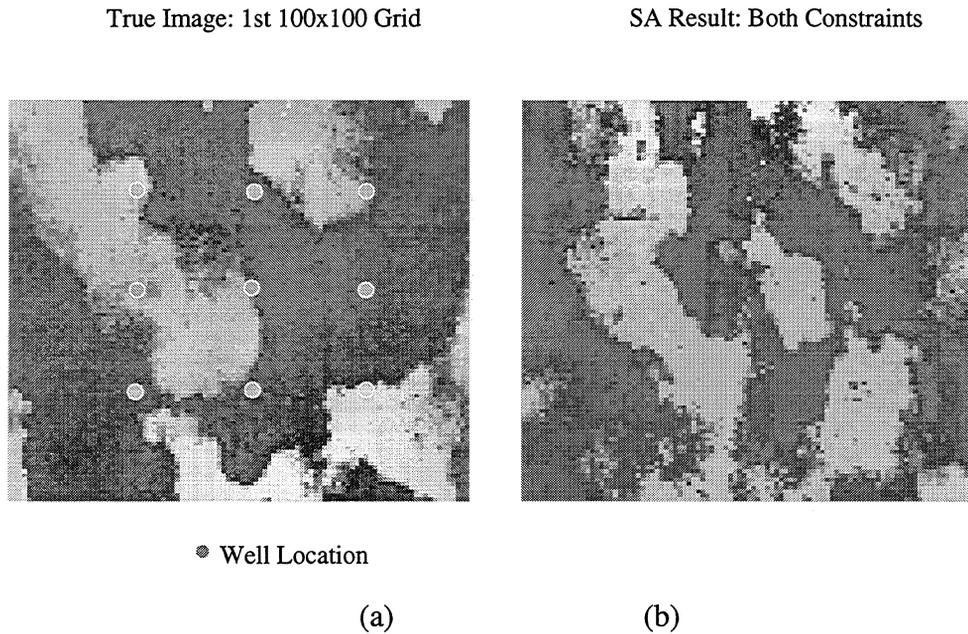


Figure 48: Results from sensitivity tests on well locations

## 5.6 Conclusions

### 5.6.1 Behavior of Objective Function Components

In testing the modified SA algorithm, the behavior of the individual components of the objective function was monitored. This was done by keeping track of the numbers of energy-reducing perturbations for both components. Thus we determined how many perturbations resulted in reducing the energy of the variogram component and how many reduced *both* components' energy. It was found that there was a divergence in the correspondence between the variogram component and that of the flow simulation as the temperature level lowered. In one case for example, the ratio of the number of flow simulation-component energy-reducing perturbations to the number of variogram-component energy-reducing perturbations was initially about 56%; at the end of the SA run however, this ratio dropped to about 45%. This suggests some degree of *incompatibility* between these constraints. When we consider how they are calculated, this does not seem unreasonable. The variogram component is based on mismatches between punctual

values in permeability. The flow simulation component is a summation of errors in the wells' flowing *BHPs*. These pressures at a particular time are driven by the *average* permeability over a certain region of the reservoir (which changes with time).<sup>53</sup> So they in fact represent different measures of the same variable (permeability).

It has been suggested that “conflicting” components are undesirable and may lead to adverse results in SA.<sup>56</sup> We submit however, that all *relevant* constraints carry some useful information about the variable we are attempting to model. Hence intrinsically, there should be no conflict among such constraints, because they are all either responses or measures of the same distribution of the variable. In fact, in some cases data which do not conflict may be redundant and so add nothing to the solution of the problem. Having ‘different’ constraints is therefore more useful because:

- they are present different information on the variable which we are trying to model, and
- because they capture different informational aspects of that variable, they may contribute to ‘constraining’ the behavior of the other component. This suggests symbiosis in having such constraints acting in concert. For example, we have established that the inverse problem is non-unique, thus alternate optima may exist; having the variogram constraint may help to ‘guide’ the flow simulation constraint towards a solution which is the ‘true’ optimum we seek and vice versa.

The symbiosis suggested above is however a two-edged sword. If a wrong variogram model is used to constrain the description, the overall description could be worse than if we used flow simulation as the only constraint.

### ***5.6.2 Effectiveness of Dynamic Constraint***

The results presented above suggest that the inclusion of dynamic constraints significantly improves the performance of SA in matching the reservoir description as well as the dynamic behavior of the reservoir, even with respect to its *future* behavior. It should be noted however, that the location of the wellblocks plays an important role in these results. For example, if the wells are located in the relatively low-permeability areas, the matches obtained are not as good as

when the wells were sited in the higher-permeability locations. One possible explanation for this can be related to the limitations of the upscaling methods used. As observed, when there is significant contrast between the wellblock permeability and the surrounding permeability values, these methods do not work well. Another explanation may be that, as Vasco et al.<sup>57</sup> concluded, *transient* pressure data seem to be most sensitive to structure in the immediate vicinity of the wells and do not seem to have as much resolving power as, e.g., tracer data. Thus it would be difficult for the wells in the low-permeability areas to resolve structural features of the high-permeability areas. Note however that our pressure data can include post-transient information, and so we should expect to capture some features which may be relatively remote from the wells, but which influence their pressure behavior at later times. In addition, it may be argued that from a practical standpoint we are primarily interested in the regions of higher potential in the reservoir, i.e., the higher permeability regions. The question arises therefore as to how do we obtain *beforehand*, knowledge on where these areas are. One source may be seismic information from which we may have, in a general sense, a feel for the 'lay of the land'. Another source of information may be existing well data. In this case we would consider only those data which satisfy the constraints identified above.

### ***5.6.3 Effectiveness of Modified Geometric Averaging Upscaling Approach***

This procedure has been shown to perform well in upscaling the reservoir description for flow simulation purposes. The flowing *BHPs* obtained at the coarse scale agree with those at the fine scale within about  $\pm 10\%$  even at very early times ( $< 0.1$  day) and better than  $\pm 5\%$  after 1 day flowing time. Thus this approach may be used in a 2-scale SA approach with reasonable accuracy if pressure data after about 1 day flowing time is used. There is however, a limitation on the degree of contrast, which may be accommodated by this approach.

### ***5.6.4 Effectiveness of the LTFD Approach<sup>43</sup>***

This algorithm is theoretically appealing in the sense that it suggests an approach for flow simulation which can efficiently timestep to given times of interest. However, there are severe limitations to the current implementation of this approach as stated above. The major ones were

related to the algorithm's limitation to single-phase flow and the trend towards matrix singularity at longer flow periods, especially for large grids.

### **5.6.5 Effectiveness of the Modified SA Algorithm**

The variogram-only SA algorithm for a grid size of 100x100 cells may be run in about 10-15 minutes while, as we have seen, the composite objective function run takes on the order of about one day.<sup>a</sup> This difference in run times may be acceptable only if the results obtained via the more lengthy run are much better than what is obtained from the variogram-only run. The results presented support this claim. Nevertheless, we perceive this work as a beginning. There is much improvement possible by refinements in the methodology. However, the *conceptual framework* has been developed and has also been shown to be a viable strategy. There may be tremendous benefits to be derived from a hybrid approach that considers this approach and, for example, a parallelized implementation.

## **5.7 Recommendations**

### **5.7.1 Use of Tracer Tests Information**

As alluded to above, the use of tracer tests may be more effective than pressure data in resolving the very small-scale reservoir features. It is therefore suggested that this information be considered as an additional constraint in a SA algorithm for reservoir characterization. It may be used as a replacement for the pressure information or as a third constraint. In this case the objective function would relate to differences between the observed and calculated tracer concentrations at the well locations. It should be noted however that such data are not as widely-available as pressure data.

### **5.7.2 3-D Upscaling and Multiphase Flow Simulation**

This work addresses only 2-D upscaling approaches. As shown above, even though there is a considerable improvement in the matching of the pressure behavior between scales when the

---

<sup>a</sup> The length of time for a run is a function of the rate of convergence. Thus it is difficult to state a more precise figure for the run times; this can only be done when quoting the run time for an *instance* of the SA problem.

modified upscaling techniques are used, these techniques are still approximate and improvements are possible.

An obvious -- but not as straightforward -- extension is to 3-D upscaling. A suggested basis for this is the work of Lee.<sup>58</sup> Additionally, multiphase simulation should be considered since this is much more sensitive to reservoir heterogeneity and because such conditions are more prevalent than single-phase flow. As a starting point, *ECLIPSE-100* may be considered for simulation since the features are already present for multiphase simulation runs.

### **5.7.3 LTFD Improvements**

This is a possible area for some further work because of the promise held by a Laplace transform approach. The focus should be to try and improve the stability of the matrix problem.

### **5.7.4 Annealing Schedule Optimization**

The use of a convergence rate factor, which was different from that suggested by Perez,<sup>41</sup> was mentioned above. By a limited amount of sensitivity analysis we selected a value of 0.1 instead of the 0.5 value used by Perez. However, optimum values for the other components of the annealing schedule for the modified approach were not investigated. Work in this area may yield improvements in the algorithm's efficiency

## **6. Well Model Identification System**

### **6.1 Subsystem-1**

When the Well Model Identification System is run, Subsystem-1 is invoked first, and it prompts the user for the name of the input file that contains the time/pressure data derived from well sites. From this data, we generate the derivative data using standard algorithms. Thus, the input to the system consists of time and derivative information, and the actual number of time and derivative data pairs that are used as input, depends upon the test data available. Once the system reads in the time and the derivative data, the next step is to analyze this data and come up with the simplified symbolic representation for the whole plot. This representation is done in terms of the following symbols:

*up, down, flat, maximum, minimum, plateau, valley*

The algorithm begins by calculating the slopes between the data points. This data is stored in a list. The algorithm proceeds by scanning this list and replacing each slope with symbolic representation. Then a second scan through the list is done to come up with a final representation of the whole plot. Here the algorithm uses rules that we developed to eliminate redundant symbols, or to identify new ones based on the sequence of primitive symbols (*up, down, flat*). Typical rules are as follows:

*up* followed by an *up* is *up*

*down* followed by *down* is *down*

*up* followed by *flat* followed by *down* is a *maximum* if the number of *flats* in between is sufficiently small, otherwise it is a *plateau*.

Using such rules, the algorithm produces a final list of representative symbols which describe the whole plot. Though it depends upon the particular data used, usually four to five symbols describe one complete graph. Consecutive identical symbols (e.g. *up, up*) are compressed into a single symbol representing several segments. Currently there are total of five possible shapes representing the models:

shape 1: [*up, maximum, down, flat*]

shape 2: [*up, maximum, down, flat, down, flat*]

shape 3: [*up, flat*]

shape 4: [*up, maximum, down, minimum, up, flat*]

shape 5: [*up, maximum, down, flat, down, minimum, up, flat*]

Once the representative symbols are produced, a conversion algorithm is invoked to convert the representative symbols (e.g. *up, maximum, flat*) into one or more paired integers (e.g. (-4 , 0), (-3 , 0)) and stored in a list Figure 49 below shows the 3 possible results in the list.

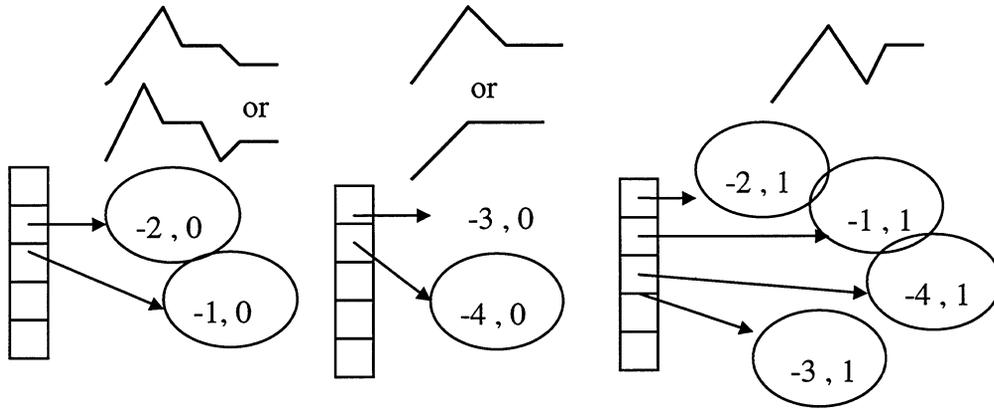
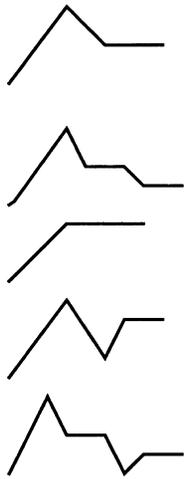


Figure 49: Representative Symbols

The conversion algorithm is necessary because several models may have the same shape and one model may have two shapes, but each model has a unique paired integers that represent it. These unique paired integers will be used in matching algorithm to match the models selected in Subsystem-1 and Subsystem-2. Figure 50 below shows the all possible shapes and the all possible models.

**ALL POSSIBLE SHAPES**



**ALL POSSIBLE MODELS**  
(DIFFERENT MODELS  
CAN HAVE SAME SHAPE)

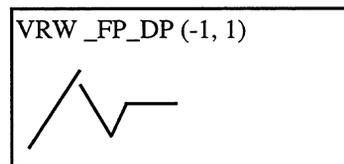
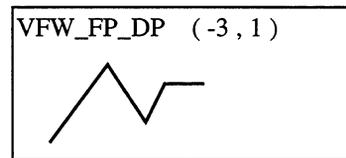
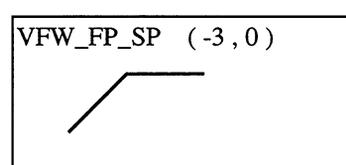
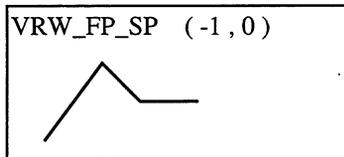
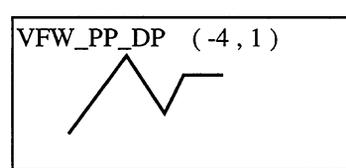
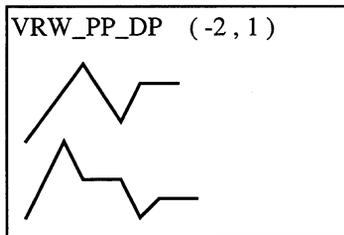
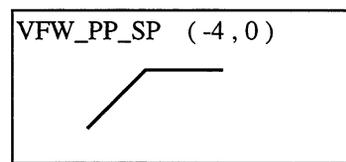
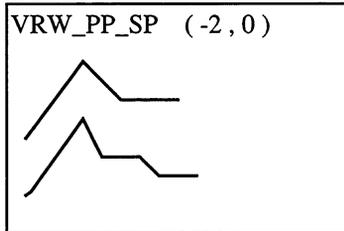


Figure 50: Possible shapes and models

**6.2 Subsystem-2**

This subsystem obtains information from geologist and uses anInference Engine for model identification and selection. This subsystem is invoked after the Subsystem-1 completed its task. When this subsystem is activated, it prompts the user for input. 3 questions will be asked, and the geologist must provide all 3 answers:

1. Type of well

2. Fully or Partially Penetrating

3. Single or Double Porosity

Then input information is interpreted. Interpreting this information involves evaluating conditions and rules from multiple sources. By Applying AI, especially knowledge based system techniques, it is easier to codify the experienced engineers' or analysts' expertise, and simulate the interpretation process. The well completion information gathered from experts in the Petroleum department was represented in the form of rules. The present rule base consists of 5 parameters and 17 rules. Inference in the rule base is made using *forward reasoning*. Forward reasoning is the process of working from a set of data towards the conclusion that can be drawn from this data. The input information is then used by the Inference Engine to fire the appropriate rules. The hierarchy diagram for the rule base is shown in Figure 51.

The Inference Engine, using the above information provided by the geologist, will always produce 1 answer which is the selected model in the form of, e.g. VRW\_PP\_SP (Vertical Radial Well, Partially Penetrating, Single Porosity). This result is then passed to a conversion function so the selected model can be represented in the form of unique paired integers, which will then be passed to the matching algorithm for final model matching and selecting. This conversion function will convert the result from (e.g. VRW\_PP\_SP) to paired integers (e.g. (-2 , 0) ).

### **6.3 Well Model Matching**

Matching : Result from Subsystem-2 is matched against the result from Subsystem-1. If no match, the algorithm terminate with no match message. Otherwise, the final matched model will be passed to the third subsystem which will calculate the nonlinear regression. Figure 52 below shows the 3 possible results from Subsystem-1, but only one of the 3 will be matched.

Horizontal well ( 10 , 10 ) will take the 1<sup>st</sup> answer from the array.

Otherwise, match them and store the answer.

These two subsystems are currently implemented using MSVC 1.51. The two subsystems are fully integrated and tested successfully.



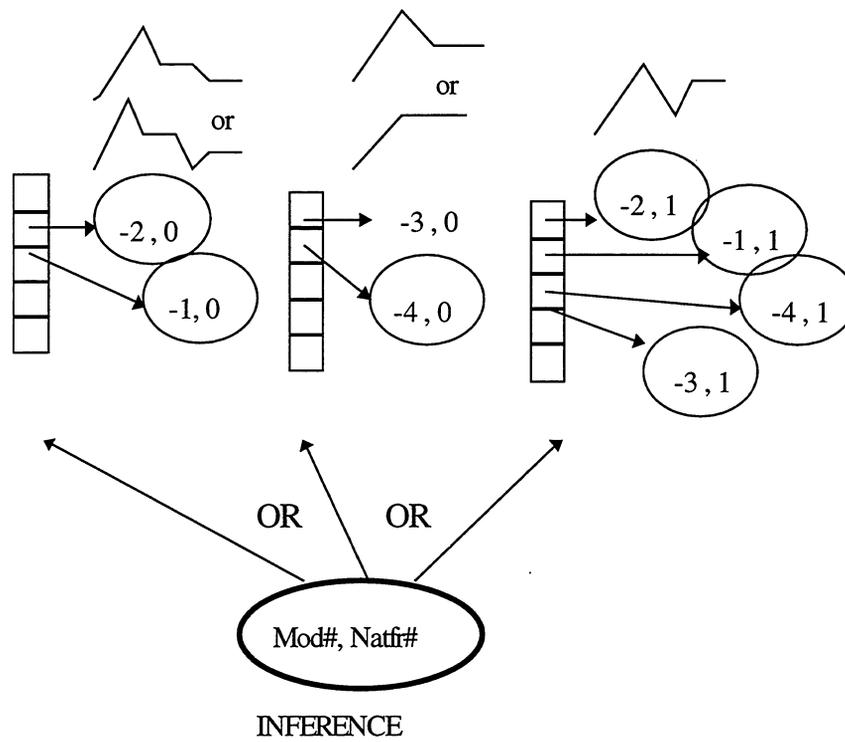


Figure 52: Possible results

The details of the non-linear regression algorithm for well testing were reported in the January 1996 annual report submitted for this project.

## References

1. Hand, J. L., et al., "Ability of Geostatistical Simulations to Reproduce Geology : A Critical Evaluation", Paper SPE 28414 presented at the 69<sup>th</sup> SPE Annual Meeting, New Orleans, L. A., Sept. 25-28, 1994.
2. Gawith, D. E., et al., "Integrating Geoscience and Engineering for Improved Field Management and Appraisal", paper SPE 29928, presented at the 70<sup>th</sup> SPE Annual Meeting, Dallas, Texas, Oct. 22-25, 1995.
3. Ametis, L. J., "Application of Sequence Stratigraphic Modeling to Integrated Reservoir Management at Aneth Unit, Greater Aneth Field, Utah.
4. Bashore, W. M., "Importance of a Geological Framework and Seismic Data Integration for Reservoir Modeling and Subsequent Fluid-Flow Predictions", in *Stochastic Modeling and Geostatistics : Practical Applications and Case Studies*, J. M. Yarus and R. L. Chambers (eds.) , American Association of Petroleum Geologists, Computer Applications in Geology, No. 3, 1994.
5. Deutsch, C. V., and Meehan, D. N., "Geostatistical Techniques Improve Reservoir Management", *Petroleum Engineer International*, March 1996, Page 24.
6. Bahar, A. et al., "Integrated Reservoir Description and Flow Performance Evaluation : Glenn Pool Field - Self Unit Study", Paper SPE 30622 presented at the 70<sup>th</sup> SPE Annual Meeting, Dallas, TX, Oct. 22-25. 1995.
7. Pande, P. K., et al., "Application of integrated reservoir management and reservoir characterization to optimize infill drilling", Continuation Application, DOE/BC/14989/Bartlesville Project Office, US DOE, OK, January 1996.
8. Jordan, D. L., "An Application of Categorical Indicator Geostatistics for Facies Modeling in Sand-Rich Turbidite Systems", Paper SPE 30603 presented at the 70<sup>th</sup> SPE Annual Meeting, Dallas, TX, Oct. 22-25. 1995.

9. Alabert, F. G., and Massonnat, G. J., "Heterogeneity in a Complex Turbiditic Reservoir: Stochastic Modelling of Facies and Petrophysical Variability", Paper SPE 20604 presented at the 65<sup>th</sup> SPE Annual Meeting, New Orleans, 1990.
10. Damsleth, Elvind, et al., "A Two Stage Stochastic Model Applied to a North Sea Reservoir", Paper SPE 20605 presented at the 65<sup>th</sup> SPE Annual Meeting, 1990.
11. Ahuja, B. K., "Integration of Geological and Petrophysical Information Using Geostatistical Methods - Self Unit Study", M.S. Thesis, The University of Tulsa, 1993.
12. Myers, Donald E., "Matrix Formulation of Co-Kriging", *Math. Geology*, 14, p. 249.
13. Verly, G., : "Sequential Gaussian cosimulation: a simulation method integrating several types of information", *Proceedings of 4-th International Geostatistical Congress*, 11p.
14. Carr, J., and Myers, D. : "COSIM: a Fortran IV program for co-conditional simulation program", *Computers & Geosciences*, 11(6): 675-705, 1985.
15. Pizzaro, J. O., and Lake, L., "A Simple Method to Estimate Interwell Autocorrelation", paper presented at Fourth International Reservoir Characterization Technical Conference, Houston, March 1997.
16. Damsleth, Elvind, "Mixed Reservoir Characterization Methods", paper SPE 27969 presented at the University of Tulsa Centennial Petroleum Engineering Symposium, Tulsa, August 1994.
17. Omre, H. et al., : "Simulations of Random Functions on Large Lattices", in *Geostatistics Troia '92*, A. Soares (ed.), Kluwer Academic Publishers, Dordrecht, Vol. 1 (1992), 179-201.
18. Deutsch, C. V., and Journel, A. G. : *Geostatistical Software and Library and User's Guide*, Oxford University Press, 1992.
19. F. Alabert. "The Practice of fast conditional simulation through the LU decomposition of the covariance matrix". *Math Geology*, 19(5): 369-386, 1987.

20. Davis., M. : “Production of conditional simulations via the LU decomposition of the covariance matrix. *Math Geology*, 19(2):91-98, 1987.
21. Rudkiewicz, J. L., et al. : “An Integrated Software for Stochastic Modeling of Reservoir Lithology and Property with an Example from the Yorkshire Middle Jurassic”, in *North Sea Oil and Gas Reservoirs - II*, A. T. Buller et al. (eds.) Graham and Trotman, London, 1990, 399-406.
22. Xu, Wenlong, and Journel A. G. : “GTSIM : Gaussian Truncated Simulations of Lithofacies”, Report-6 Stanford Center for Reservoir Forecasting, May 1993.
23. Galli A., et al., “The Pros and Cons of the truncated Gaussian method”, M. Armstrong and P. A. Dowd (eds.), *Geostatistical Simulations*, 217-233, Kluwer Academic Publishers, 1994.
24. Almeida, A. S., “Joint simulation of multiple petrophysical properties using a Markov-type coregionalization model: Stanford Center for Reservoir Forecasting”, Report 6, Stanford, 1993, 31p.
25. Bahar, A., “Integrated Reservoir Description and Flow Performance Evaluation - Glenn Pool Field, Self Unit Study”, M.S. Thesis, The University of Tulsa, 1994.
26. Almeida, A. S., et. al, “Generation of Stanford-1 reservoirs: Stanford Center for Reserovoir Forecasting, Report 5, Stanford, 1992, 81p.
27. Freeman, G. Personal Communication, Dept. of Petroleum Engineering, The University of Tulsa, Tulsa, Oklahoma, 1997.
28. Robinson, E. A., and Treitel, S. : *Geophysical Signal Analysis*, Prentice-Hall, Inc., Englewood Cliff, N. J., 1980.
29. Press, William H., et al. : *Numerical Recipes in C, The Art of Scientific Computing*, Second Edition, Cambridge University Press, 1992

30. Kerr, D. R., and Jirik, L. A. : “Fluvial Architecture and Reservoir Compartmentalization in the Oligocene Middle Frio Formation, South Texas”, paper from Transactions-Gulf Coast Association of Geological Sciences, Volume XL. 1990, pages 373-380.
31. Kerr, D. R., and Ye, L. M., “Geological Interpretation”, Second Quarterly DOE Report - 1994 - Contract No. DE-FC22-93BC14951, “Integrated Approach Towards the Application of Horizontal Wells to Improve Waterflooding performance”, The University of Tulsa, 1994.
32. Kerr, D. R., and Ye, L. M., “Geological Interpretation”, First Quarterly DOE Report - 1997 - Contract No. DE-FC22-93BC14951, “Integrated Approach Towards the Application of Horizontal Wells to Improve Waterflooding performance”, The University of Tulsa, 1997.
33. Paranjli, S., Personal Communication, Dept. of Petroleum Engineering, The University of Tulsa, Tulsa, Oklahoma, 1997.
34. Haldorsen, H.H., Brand, P.J., MacDonald, C.J.: "Review of the Stochastic Nature of Reservoirs", *Mathematics in Oil Production*, (S. Edwards and P. King, editors) Oxford Science Publications, Oxford (1988) 109-209.
35. Tarantola, Albert: *Inverse Problem Theory Methods for Data Fitting and Model Parameter Estimation*, Elsevier Science Publishers, Amsterdam (1987).
36. Journel, A.G. and Huijbregts, Ch.J.: *Mining Geostatistics*, Academic Press, Inc., London (1978).
37. Oliver, D.S.: “Multiple Realizations of the Permeability Field From Well Test Data”, paper SPE 27970 presented at the SPE & University of Tulsa Centennial Petroleum Engineering Symposium held in Tulsa, OK, August 29-31, 1994.
38. Deutsch, C.V. and Journel, A.G.: *GSLIB Geostatistical Software Library and User's Guide*, Oxford University Press, New York (1992).

39. Kelkar, B.G.: *Application of Statistics to Reservoir Characterization*, self-published, Tulsa, OK (1989).
40. Kirkpatrick, S., Gelatt, Jr., C.D., Vecchi, M.P.: "Optimization by Simulated Annealing", *Science* (May 13 1983) 671-680.
41. Perez, G.: "Stochastic Conditional Simulation for Description of Reservoir Properties", Ph.D. dissertation, University of Tulsa, Tulsa, OK (1992).
42. Gajraj, A: "The Incorporation of Dynamic Constraints in Stochastic Conditional Simulation", Ph.D. dissertation, University of Tulsa, Tulsa, OK (1996).
43. Moridis, G.J., McVay, D.A., Reddell, D.L. and Blasingame, T.A.: "The Laplace Transform Finite Difference (LTFD) Numerical Method for Simulation of Compressible Fluid Flow in Reservoirs", SPE 22888 presented at the 1991 Annual Technical Conference and Exhibition, Dallas, TX, Oct. 5-8.
44. Ouenes, A., Bréfort, B., Meunier, G. and Dupéré, S.: "A New Algorithm for Automatic History-Matching: Application of Simulated Annealing Method (SAM) to Reservoir Inverse Modeling".
45. Oliver, D.S.: "Incorporation of Transient Pressure Data into Reservoir Characterization", *In Situ*, 18(3), 243-275 (1994).
46. Bear, Jacob: *Dynamics of Fluids in Porous Media*, American Elsevier Publishing Company, Inc., New York (1972).
47. Lake, Larry: *Enhanced Oil Recovery*, Prentice-Hall, Inc., Englewoods Cliffs, New Jersey (1989).
48. Ouenes, A.: "Application of Simulated Annealing to Reservoir Characterization and Petrophysics Inverse Problems", PhD Dissertation, The New Mexico Institute of Mining and Technology, Socorro, NM (1992).

49. Long, J.C.S., Doughty, C., Hestir, K., Martel, S.: "Modeling Heterogeneous and Fractured Reservoirs with Inverse Methods Based on Iterated Function Systems", paper presented at the Third International Reservoir Characterization Technical Conference, Tulsa, OK Nov. 3-5, 1991.
50. Fasanino, G., Molinard, J.E.: "Inverse Modeling of Gas Reservoirs", paper SPE 15592 presented at the 1986 Annual Technical Conference and Exhibition, New Orleans, LA, Oct. 5-8.
51. Stehfest, H.: "Numerical Inversion of Laplace Transforms", Algorithm 368, *Communications of ACM* (Jan. 1970) **13**, No. 1. 47-49.
52. Lee, John: *Well Testing*, SPE Textbook Series Volume 1, Society of Petroleum Engineers of AIME, New York (1982).
53. Oliver, D.S.: "The Averaging Process in Permeability Estimation From Well-Test Data", *SPEFE* (Sept. 1990) 319-324.
54. *ECLIPSE 100 - Black Oil Simulator*, ECL-Bergeson Petroleum Technologies, Inc., Oxfordshire, England (1990).
55. Sagar, R.K: "Reservoir Description by Integration of Well Test Data and Spatial Statistics", Ph.D. dissertation, University of Tulsa, Tulsa, OK (1993).
56. Deutsch, Clayton V. and Cockerham, Perry W.: "Practical Considerations in the Application of Simulated Annealing to Stochastic Simulation", *Mathematical Geology*, Vol. 26, No. 1, pp 67-82, Jan 1994 (ISSN 08828121).
57. Vasco, D. W., Datta-Gupta, A., Long, Jane C.S.: "Resolution and Uncertainty in Hydrologic Characterization," submitted to *Water Resources Research*, 1994.
58. Lee, Jaedong: "Analytical Upscaling of Permeabilities for Reservoir Simulation Grid Blocks," Ph.D. Dissertation, The University of Tulsa, Tulsa, Oklahoma (1996).

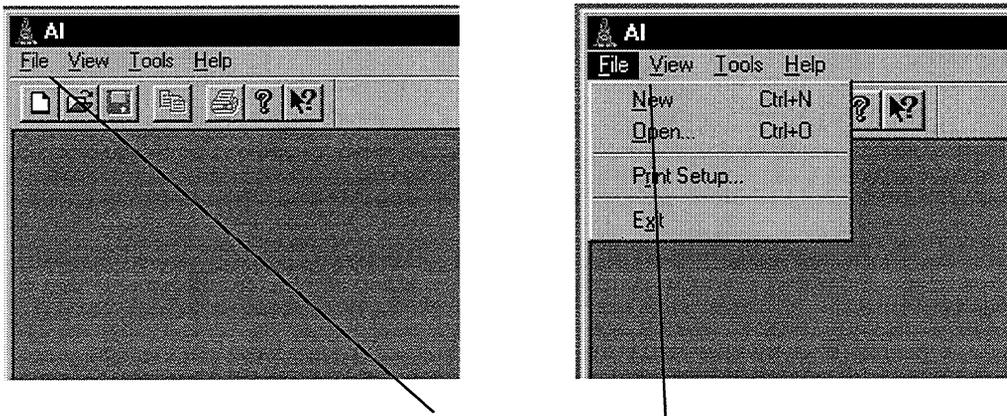
## Nomenclature

$x^*$	=	Estimated value of variable at unsampled location.
$x$	=	Value of variable at sampled location.
$\bar{u}_o$	=	Unsampled vector location.
$\bar{u}_i$	=	Data sampled vector location.
$\lambda$	=	Kriging weight.
$m$	=	Mean of the sampled data.
$C(\bar{u}_i, \bar{u}_j)$	=	Covariance between two sampled locations.
$C(\bar{u}_i, \bar{u}_o)$	=	Covariance between sampled location and un-sampled location.
$p_k$	=	Marginal frequency of category k.
$I_k(\bar{u})$	=	Indicator value at location $\bar{u}$ .
$\gamma(h)$	=	variogram at lag distance h.
$N(h)$	=	number of pair that satisfy the lag distance h.



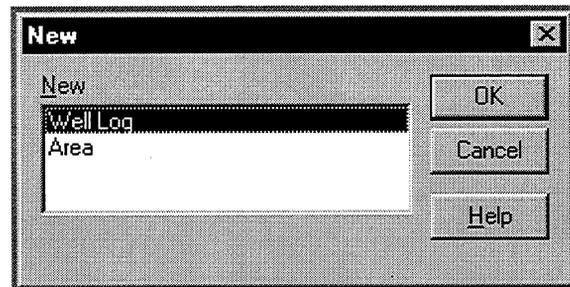
## Appendix A Visualization System

Initially, the application does not open any window. To read data from the database it has to create a new window. To do so, the user must press the bottom *new* (  ) on the toolbar or select the command *new* in the menu *File*.



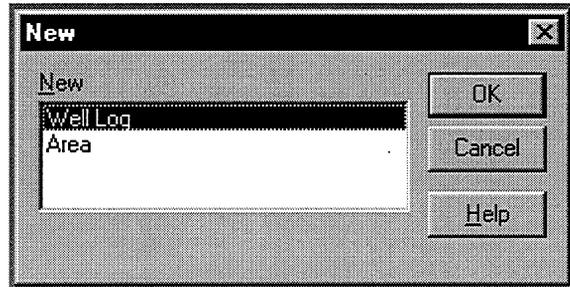
Select “New” using the button *new* or the menu *File*.

Users can select two types of window to be created. They can select to create a new window to display a well log or an area where oil well are located.

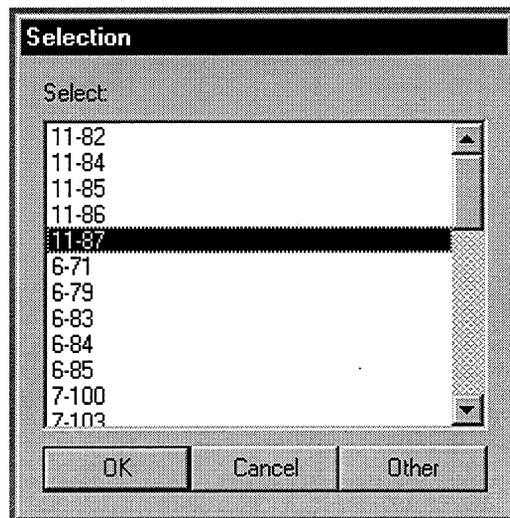


### A.1 Displaying and Working with Well Logs.

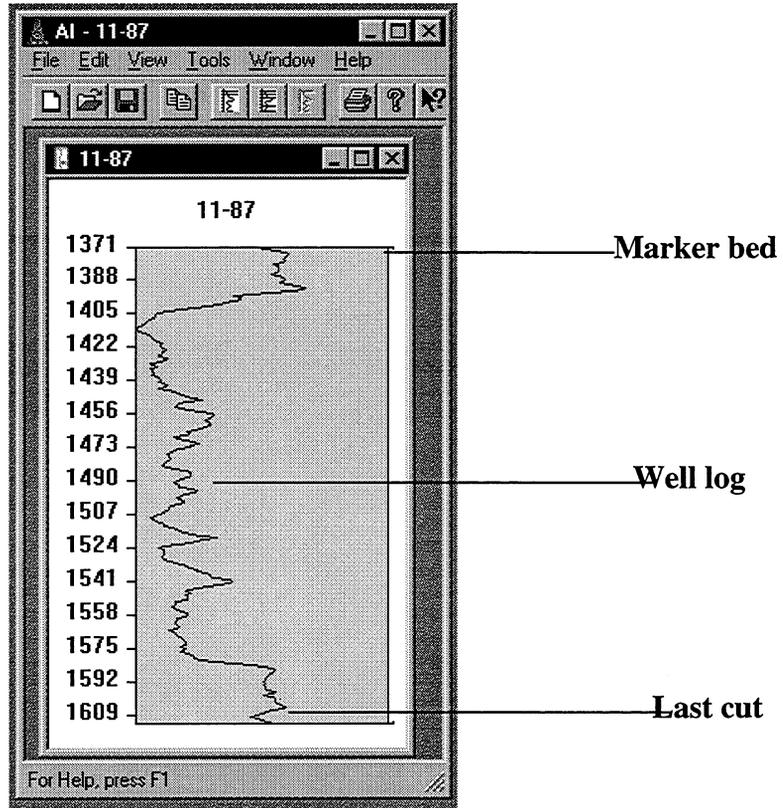
In order to open a window for displaying and working with a well log, user must select the option “*Well Log*”, and click “*OK*” .



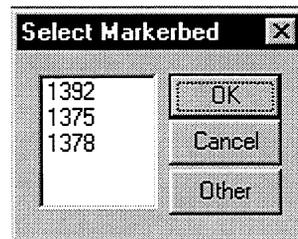
Then, the user must select which well log will be read from the database to be displayed. To do so, the user must select one of the well logs in the list and click *OK*.



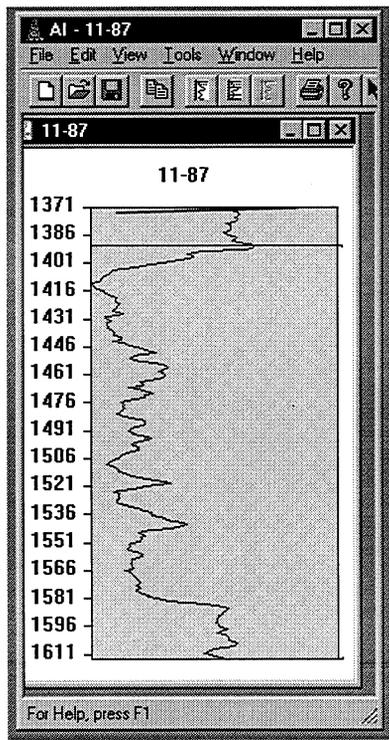
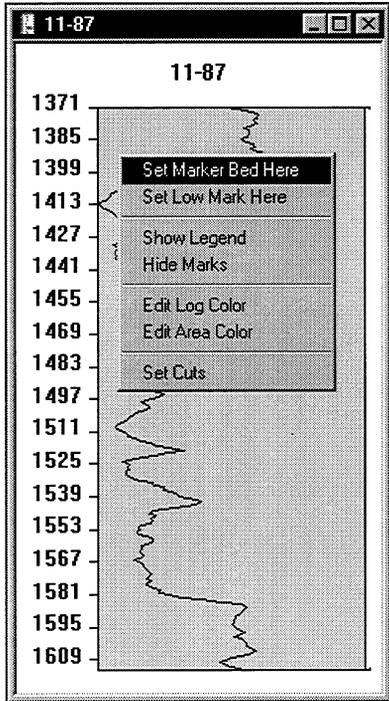
If the selected well log has never been modified (or saved modified data into the database), the well log in the window would look like the graphic in the following figure.



The user may want to change the position of the Marked bed. It can be modified semi-automatically using a tool or manually using the mouse. To modify the marker bed using the tool, the user must select the button “*Set Marked Bed*” (  ) in the toolbar or select the command “*Set Marker Bed*” in the menu *Tools*. In most cases for basic wells, the application will show three possible marker beds from which the user can select.

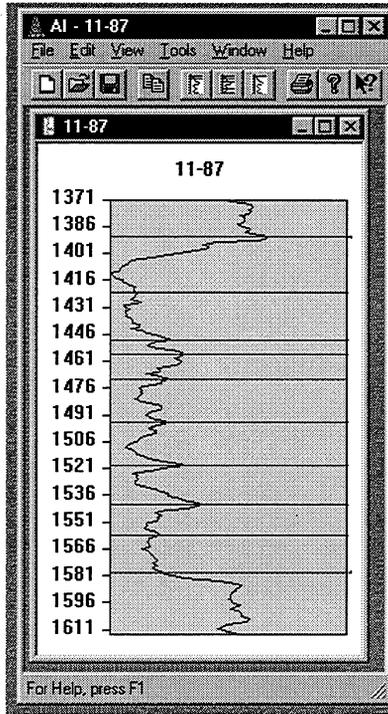


The user is allowed to select another marker bed position in addition to the ones proposed by the tool. This selection can be performed in the dialog box shown above or by using the mouse. To select the marked bed using the mouse, the user must click the right mouse button on the preferred position and select “*Set Marker bed here*” on the popup menu.



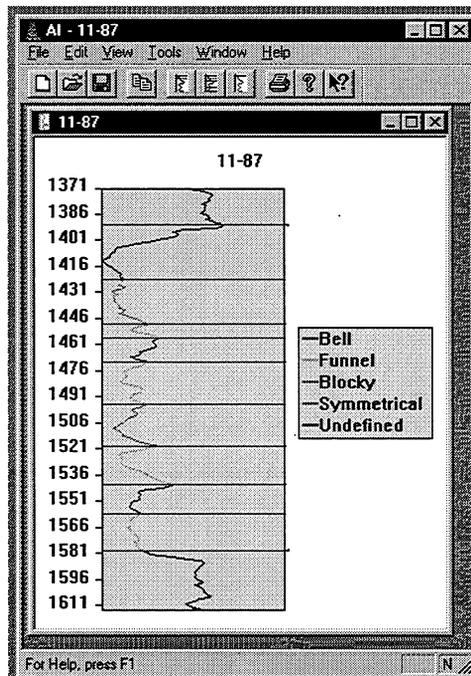
New marker bed

To determine the cuts into the well log, the user invoke the component “Set Cuts”. This component is run using the button (  ) on the toolbar, or using the command “Set Cuts” in the menu *Tool*.



Cuts Calculated by the Tool

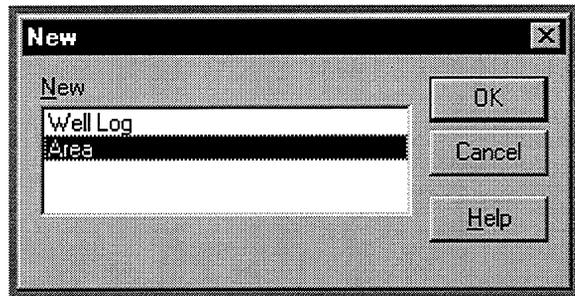
Before calculating the facies types for each discrete genetic interval, the cuts must first be set. When determining the facies type between two cuts (or discrete genetic intervals), the user must invoke the component “*Set Facies*”. This component is started via the button (  ) on the toolbar, or using the command “*Set Facies*” in the menu *Tool*.



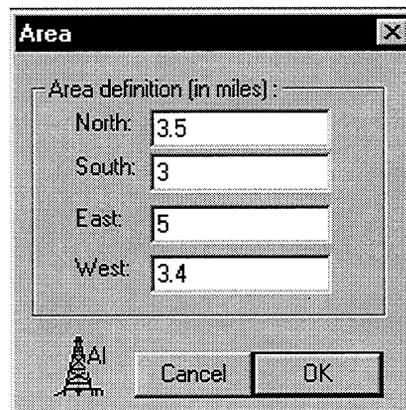
Using the command “*Update*”, the user can save the information into the database in any moment. This command can be called using the button (  ) on the toolbar or the Menu *File*.

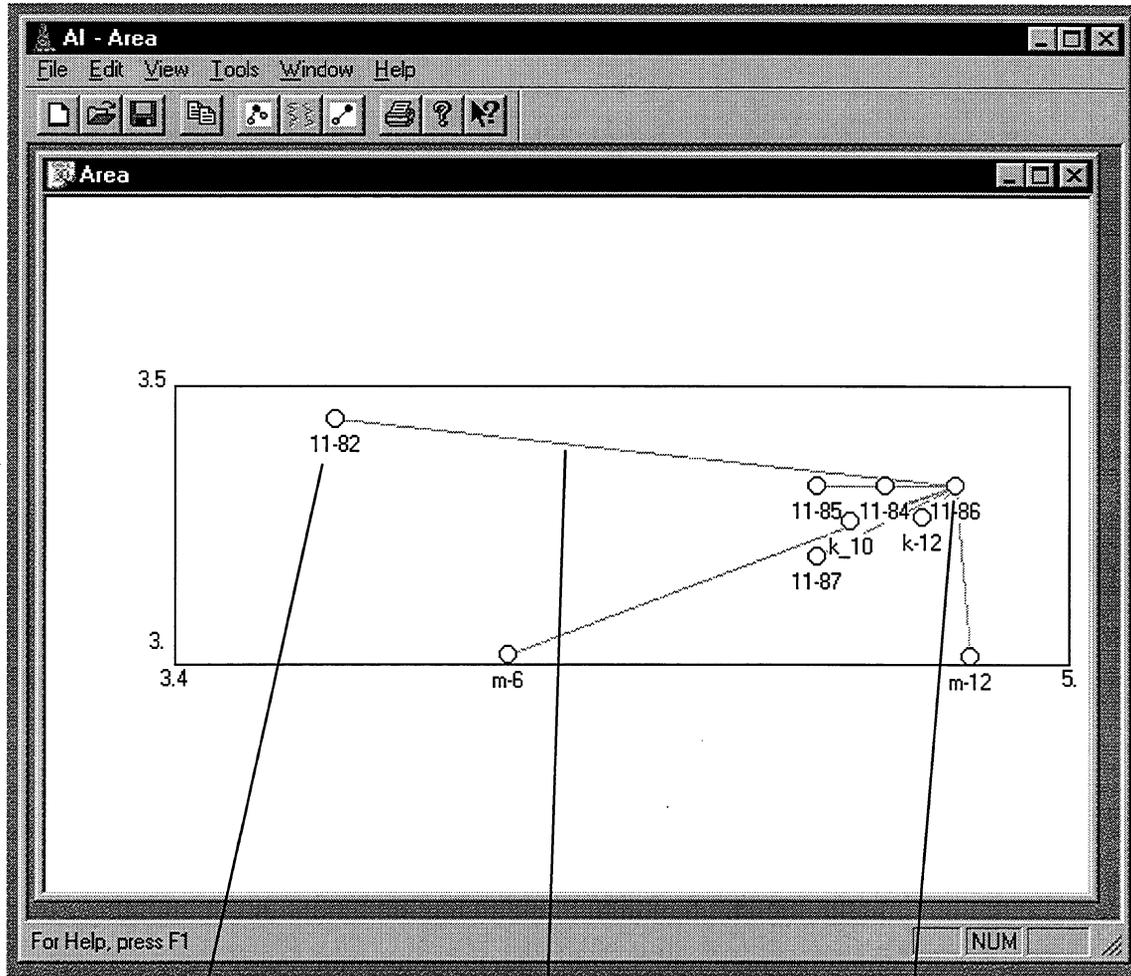
## A.2 Displaying and Working with Area

In order to open a window for displaying and working with an area, user must select the option “*Area*”, and click “*OK*” .



The application will ask the user to define the area that has to be shown in the new window. By default, it presents parameters to show all of the wells currently stored in the database.





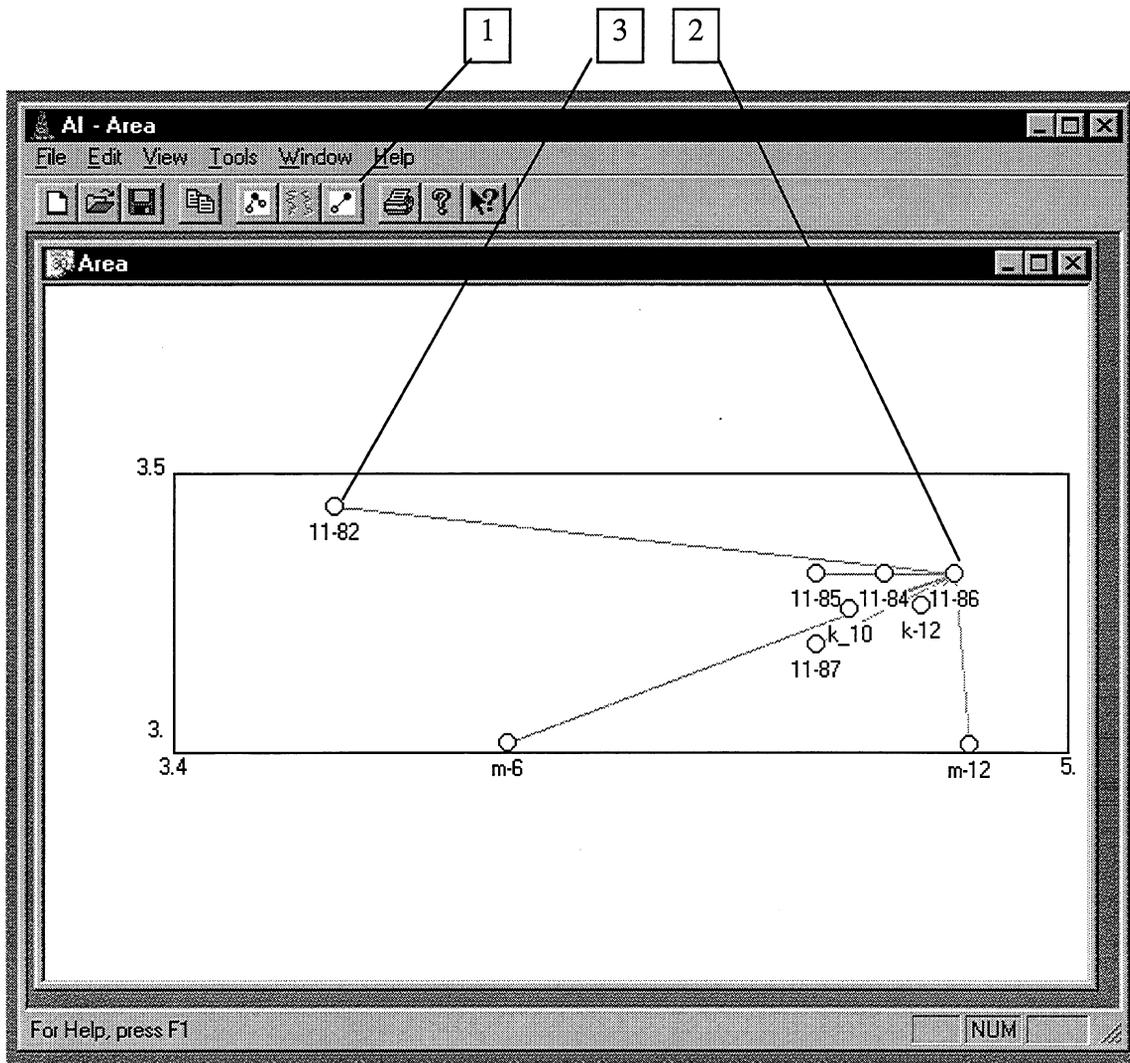
A black circle represents an oil well.

The line represent that the correlation between the oil well in each end has been calculated and stored in the database.

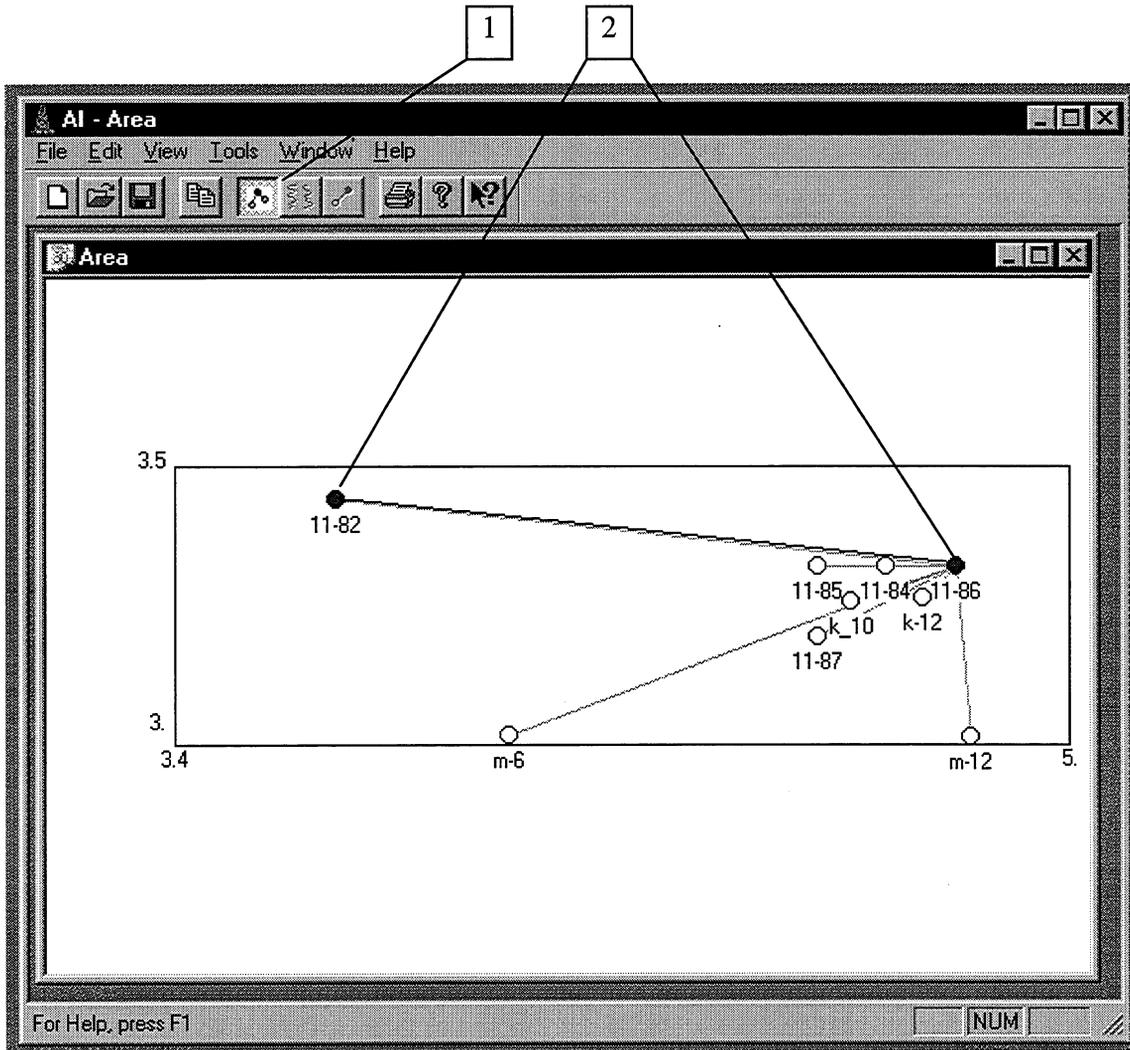
A blue circle represents a reference oil well.

To display well log information, the user can double-click on the desired well in the area map.

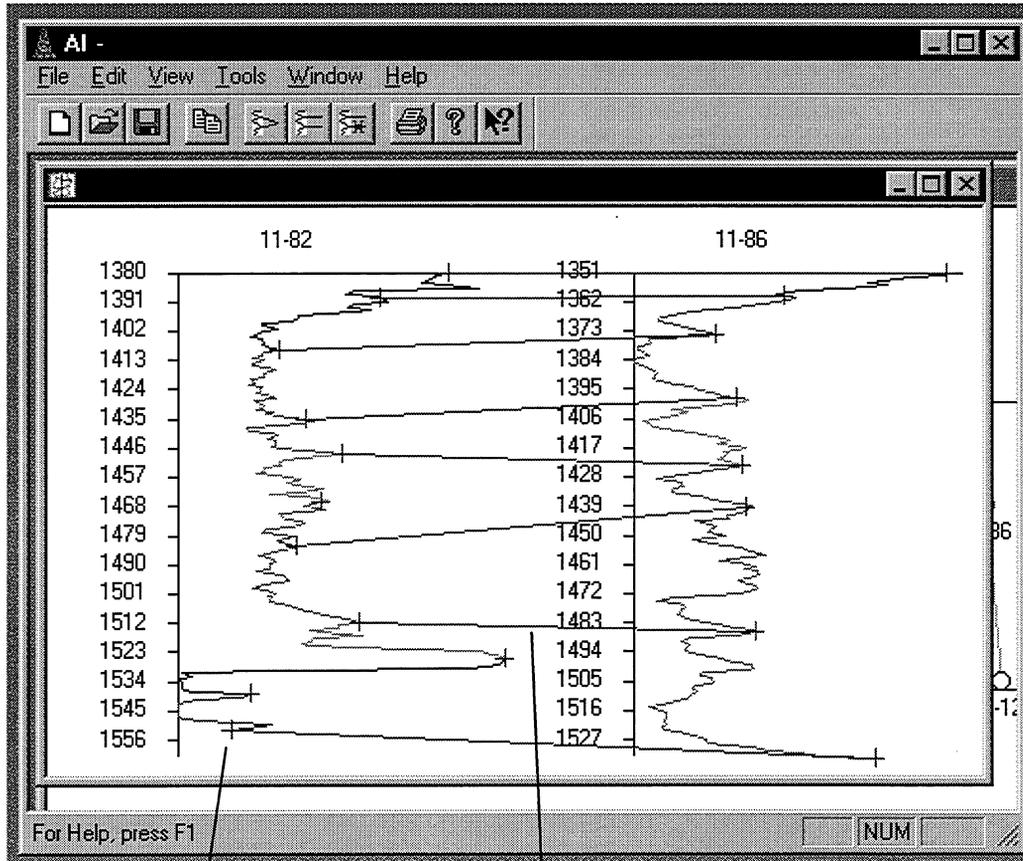
To correlate two well logs, the user must invoke the Correlation Component. In order to do so, the user has to select the tool “Correlate”, using the button “Correlate” (  ) in the toolbar or menu *Tools* [1]. Then, the user has to select (using the mouse) a reference well log [2] and a basic well log [3] (Note: One of the well logs involves in the correlation has to be a reference well log.). Facies must already be identified in both well logs before correlation can be performed.



To display a correlation, the user has to select the command “*Select*” on the toolbar (  ) [1] or on the menu *Tools*. Then, the user has to select all connected oil well that she/he wants (at least 2) [2], and select the command “*Select*” again [1].



At this moment the button “*Show Correlation*” (  ) in the toolbar becomes able. The user has to select this command to create a new window that displays the correlation.



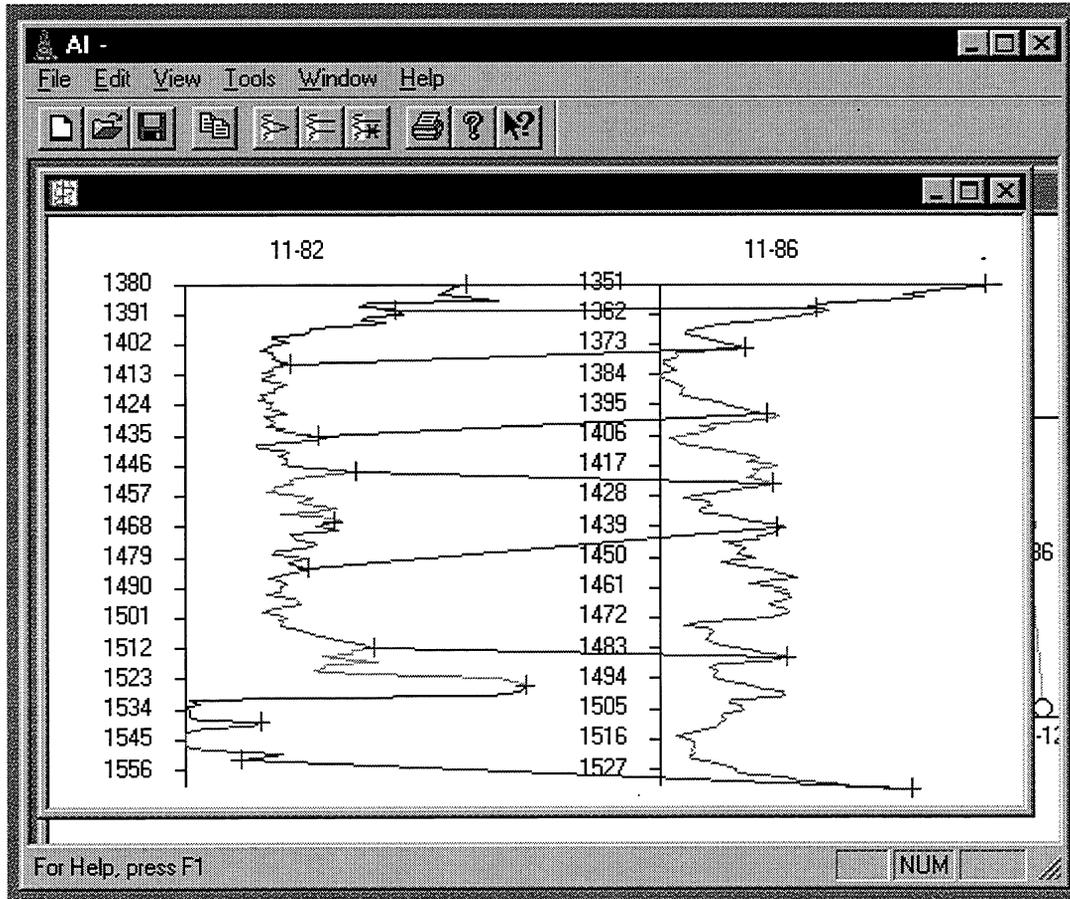
Cut (+)

### Correlation Lines

Using the command “*Save*” the information of area is saved in such a way that the program “*cosim*” can read it.

### A.3 Modifying the Correlation

The application allows users to modify the correlation between any two well logs manually. They are three commands to modify correlation, “*Move Correlation*”, “*Add Correlation*”, and “*Delete Correlation*”. These commands can be found in the toolbar (☒, ☒, and ☒ respectively) or in the menu *Tools*.



To move a correlation line, the user has to select the command “*Move Correlation*” and then to drag one of the ends of any line from a cut (+) to another in the same well log.

To add a correlation line, the user has to select the command “*Add Correlation*”, and then to drag from a cut (+) in a well log to a cut in another well log.

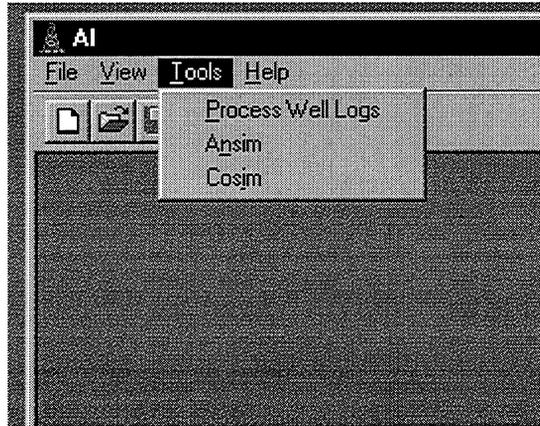
To remove a correlation line, the user has to select the command “*Delete Correlation*” and then to click on the line to be deleted. (Note: if all correlation lines are deleted, the correlation is delete from the database, but it will not be shown in any area window unless the command “*Update from database*” is called).

#### A.4 Others Commands

The application has three other commands independent of any window. They are “*Process Well Logs*”, “*Ansim*”, and “*Cosim*”.

“*Process Well Log*” command computes marker bed, calculates cuts and facies, and correlates to the closest reference well log, for all well logs in the database, but reference well log.

“*Ansim*” and “*Cosim*” commands run programs *Ansim* and *Cosim* respectively.



## **Appendix B COSIM User's Guide**

The program package developed in this work is called COSIM. This program requires IBM-PC/Compatible with Windows-95 or Windows-NT operating system. The memory requirement depends on the size of the model, i.e., total number of grid blocks. Small model can be practically run in any machine (as long as Windows 95 or NT is installed). For a big model, i.e., in the order of millions of grid blocks, high virtual memory for swapping between CPU and hard-disk space may be required. This requirement depends on computer's random access memory (RAM).

In most cases, Windows 95 uses automatic virtual memory management. Therefore, user does not need to allocate specific virtual memory size in running the model. On the other hand, Windows NT requires the administrator to set the size of the virtual memory. Therefore, for running a big model on NT machine, the program may not run unless the size of virtual memory is set appropriately. It is wise to remember that the performance of the program will reduce significantly when swapping process is involved.

The user interface is developed using Microsoft Visual C++ 4.0. This program can be executed in the same manner as any Windows program, i.e., by double clicking left mouse' button at the program name/icon or at the document file name/icon, or by executing the executable code, i.e., `cosim.exe`, from the Run menu of Windows.

As in any Windows-based program, all setting variables and data values of one particular case is stored in a document file. The default extension of COSIM's document file is "csm". For example, a document named "test" will be automatically saved as "test.csm". COSIM is developed using the Single Document Interface (SDI) principle. This means that only one document file can be opened at one time. A default setting is automatically loaded if the user run the program by double clicking the mouse' left button at the program name (not at the document file name) or selecting the File New menu after the program is executed.

Figure 53 shows the main menu of the program once it is executed. As we can see from this figure, in addition to the Windows' standard menus, e.g., File, Edit, View, and Help, COSIM consists of 3 other main menus, namely Pre-Simulation, Simulation, and Post-Simulation. The

function of each main menu is self explanatory. Some tool bars are provided for direct access to the sub-menus. The meaning of each tool bar can be found by moving the mouse close to it. For example, when the user moves the mouse close to the “010” tool bar, a small window contains the text “Indicator” appears below the mouse’ arrow and a remark at the bottom left corner prompts the function of this tool bar.

Throughout the program, the communication between user and the program is performed via a dialog window. In many instances, user is asked to provide the program with the file name either as data file or output file. A general remark about the file name will be given in the following paragraph.

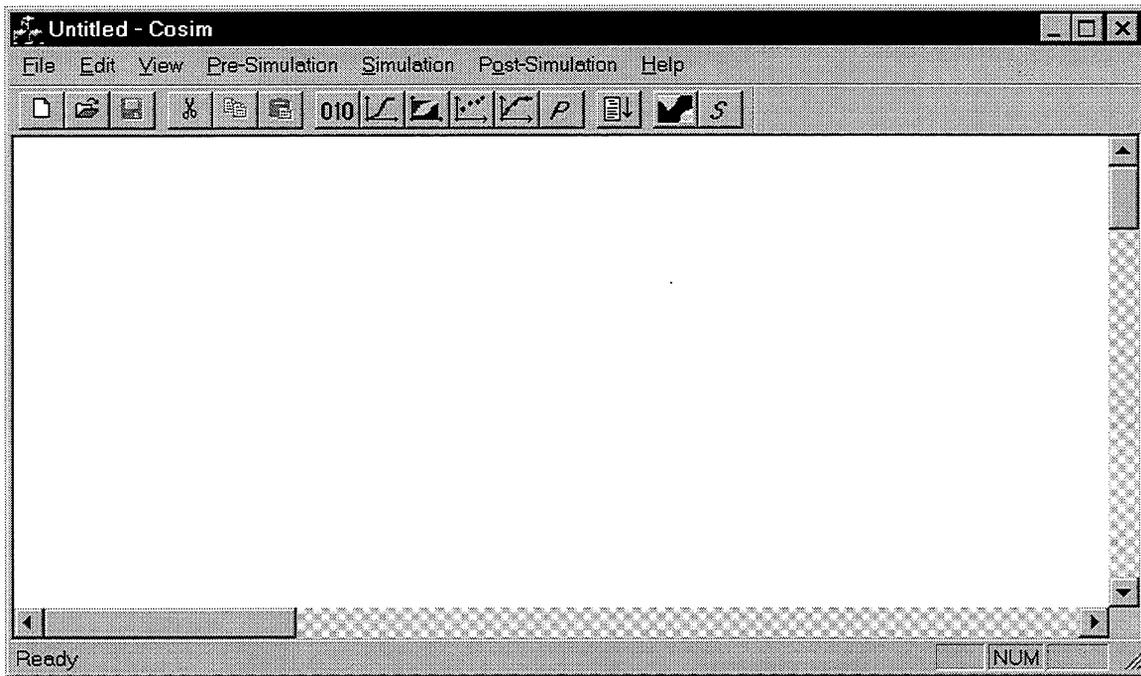


Figure 53: Main menu of COSIM

When file name is shown on a dialog box, the path of the file has been hidden from user’s view to avoid writing long name, but if the user needs to change this file, a complete path is required. To avoid the mistake in the path name, the program does not allow the user to manually type the data file name, but rather provides a standard Windows’ File-Dialog that enables the program to extract the path of the data file correctly. This is to ensure that the program always reads an existing file. For an output file name, user is allowed to input the file name directly, with or

without path. If the path is supplied the file will be written at that path otherwise it will be written at current directory.

In all calculation routines program uses the GEOEAS format for data files. This is true except when soft information is incorporated in the simulation routine. In this case, a different format, namely COSIM format, is required. This format is explained in Section A.2.5. Figure 54 shows an example of GEOEAS format. As we can see from this figure, the GEOEAS format contains  $n + 2$  lines of text where  $n$  is the number of variables followed by as many lines of data as the user wants to provide arranged column-wise. The first line is reserved for the title of the data. The second line provides the number of columns of variables,  $n$ . The third until the  $n + 2$  line provide the title of each variable. In, Figure 54 it is also shown data the data are arranged well-wise. The program does not require this arrangement but it will help the user in the organization of the data.

Self Unit Data					
5					
x					
y					
z					
DGI					
Porosity					
5.000	1968.750	1.0	7.0	0.0001	
5.000	1968.750	2.0	7.0	0.0001	
...	...	...	...	...	
...	...	...	...	...	
...	...	...	...	...	
...	...	...	...	...	
15.625	656.250	48.0	3.0	0.2000	
15.625	656.250	51.0	3.0	0.1600	
...	...	...	...	...	
...	...	...	...	...	
...	...	...	...	...	
...	...	...	...	...	
328.125	328.125	31.0	1.0	0.1200	
328.125	328.125	32.0	1.0	0.0900	
328.125	328.125	33.0	1.0	0.1000	
...	...	...	...	...	
...	...	...	...	...	
...	...	...	...	...	

Figure 54 Example of GEOEAS Format.

To edit a data file, the program provides an access to the Microsoft Wordpad or Microsoft Notepad program which are commonly installed when Windows is installed. This program can be accessed via the Pre-Simulation menu or from various dialogs that require the editor. The Wordpad program is used as the default program due to its flexibility in the size of the file. If none of these program are installed then the user needs to edit data file independently.

A difficulty may occur in accessing one of these programs if they are located in a path that does not follows the DOS naming system, i.e., path name with more than 8 characters. One solution to this problem is to make the copy (or at least the short-cut) one of these editor programs in any directory that follows the DOS naming system, e.g., C:\Windows. Similar problem may also occur when opening data file after the editor is loaded. A message about the absence of data file

may appear on the screen even though the file itself does exist. The solution to this problem is to open the file directly after the editor is loaded, i.e., through File-Open menu.

This appendix is divided into 5 sections. The first section presents the general information about the program. The next 3 sections present the guide in using the sub-menus found in the Pre-Simulation, Simulation, and Post-Simulation menus whereas in the last section an example of case study is presented. It is assumed that the user is familiar with the standard Windows' menus (File, Edit, View, and Help). Therefore, they will not be discussed here.

## B.2 Pre-Simulation

The Pre-Simulation is the facility to help the user in preparing the required parameters needed in simulation process. There are 6 menus available for this facility, namely Indicator File, Normal Transform, Vertical Prop. Curve, Variogram Analysis, Simulation Parameter, and Edit Text File. Figure 55 shows the lay-out of the Pre-Simulation with its sub-menus. The information contains in one menu is directly available for the other menus. That is, the program takes care of all inter-related information. For example, variogram modeled using Variogram Analysis menu will be directly stored as the information for Simulation Parameter menu. Consequently, the user needs to be aware that changes made in one place will automatically propagate to the other place.

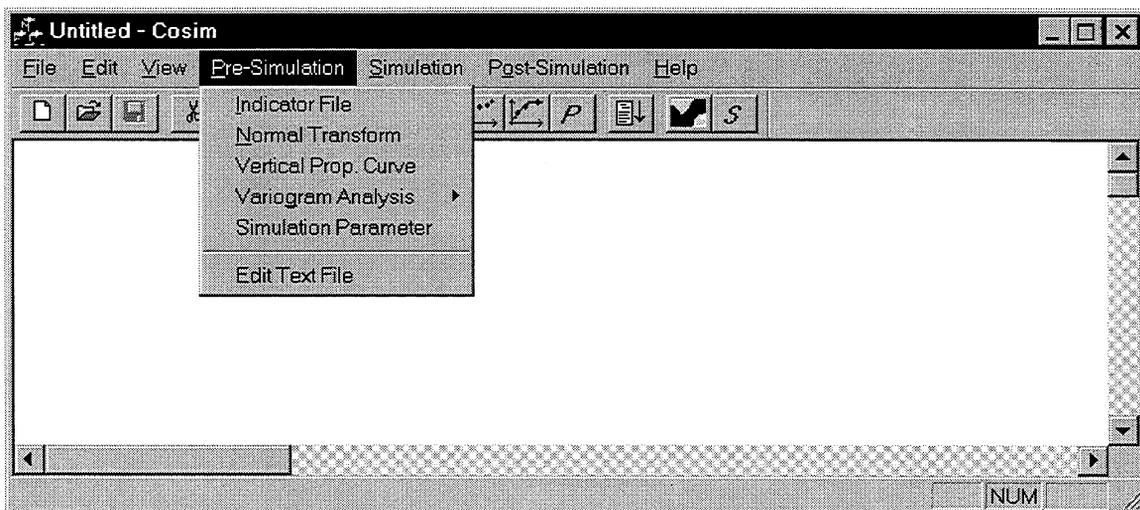


Figure 55: Pre-Simulation sub-menus

### B.2.1 Indicator File

The Indicator File menu is the facility to create a file of indicator values from the facies data at well locations. The purpose of this file is as input file in variogram analysis. Typically, facies will be named as an integer, with the first facies named as facies 1 and the last facies named as facies  $n$ , where  $n > 1$ . The conversion into indicator values is based on the presence or absence of a facies at a particular location. A value of 0 is assigned if the facies is absent and 1 if it is present. For example, if at a particular location facies 3 is present than if the number of facies,  $n$ , is equal to 4, the indicator value for this location becomes : 0 0 1 0.

This facility is not intended to create indicator values from a continuous variable, such as porosity or permeability using several cut-offs. This is due to the fact that the simulation of continuous variable is not conducted using indicator technique, but rather using Gaussian technique. Therefore, the indicator value of porosity or permeability is not required.

Figure 56 presents the dialog that appears after the Indicator File menu is selected. The only information required besides the file names is the column number of the facies data and the total number of facies defined in the system. Note that the number of facies is one of the global variable. That is, if changes are made here, it will propagate throughout the program. Figure 57 shows the example of the output result with  $n = 7$  and column number of facies in the original file is 4.

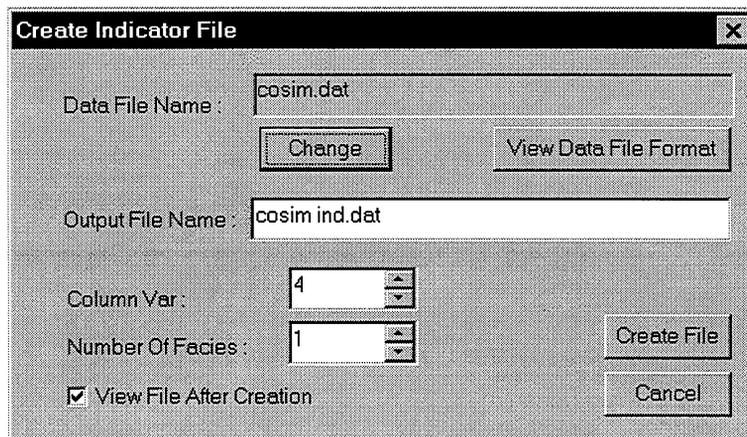


Figure 56: Indicator File Dialog

```

Indicator Variable from Data File : C:\Users\ASNUL\COSIM\Data\selfunit.dat
12
x
y
z
DGI
Porosity
Indicator : 1
Indicator : 2
Indicator : 3
Indicator : 4
Indicator : 5
Indicator : 6
Indicator : 7
5.0000 1968.7500 11.0000 7 0.0001 0 0 0 0 0 0 0 1
5.0000 1968.7500 12.0000 7 0.0001 0 0 0 0 0 0 0 1
5.0000 1968.7500 49.0000 3 0.0300 0 0 1 0 0 0 0 0
5.0000 1968.7500 50.0000 3 0.0450 0 0 1 0 0 0 0 0
5.0000 1968.7500 51.0000 3 0.0550 0 0 1 0 0 0 0 0
5.0000 1968.7500 68.0000 4 0.0900 0 0 0 1 0 0 0 0
5.0000 1968.7500 69.0000 4 0.1625 0 0 0 1 0 0 0 0
5.0000 1968.7500 70.0000 4 0.1350 0 0 0 1 0 0 0 0
5.0000 1968.7500 91.0000 5 0.1900 0 0 0 0 1 0 0 0
5.0000 1968.7500 92.0000 5 0.1800 0 0 0 0 1 0 0 0
5.0000 1968.7500 93.0000 5 0.1600 0 0 0 0 1 0 0 0
5.0000 1968.7500 94.0000 5 0.1500 0 0 0 0 1 0 0 0
5.0000 1968.7500 95.0000 5 0.1750 0 0 0 0 1 0 0 0
5.0000 1968.7500 96.0000 5 0.1800 0 0 0 0 1 0 0 0
5.0000 1968.7500 97.0000 6 0.1700 0 0 0 0 0 1 0 0
5.0000 1968.7500 98.0000 6 0.1600 0 0 0 0 0 1 0 0
5.0000 1968.7500 99.0000 6 0.1600 0 0 0 0 0 1 0 0

```

Figure 57: Output file from Indicator File menu

## B.2.2 Normal Transform

The Normal Transform menu is the facility to transform variable from its original space into Gaussian space using standard normal function, i.e., mean = 0.0 and standard deviation = 1.0. It works for both continuous and discrete variables. The purpose of this menu as a separate facility of the simulation routine is the same as the Indicator File menu, i.e., to create an output file that can be used as an input for variogram analysis.

Figure 58 shows the dialog to perform the normal transform process. As in the Indicator File menu, the only information, other than input and output filenames, required to perform normal transform for continuous variable is the column number of variable to be transformed.

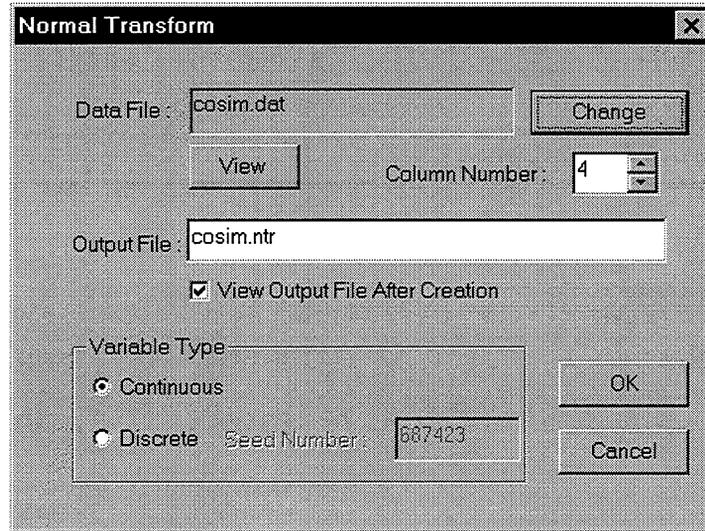


Figure 58: Normal Transform dialog.

To transform discrete variable into Gaussian space, random despiking procedure is required. This procedure requires the global probability density function (pdf) information of each indicator variable and a seed number to initiate random number generator. Therefore, for discrete variable a new dialog is required.

Figure 59 shows the global pdf dialog used in random-despiking procedure. The pdf of each variable can be input manually or if it has been entered previously it will be written automatically. The program simultaneously calculates the pdf of a data set while executing the Vertical Prop. Curve (VPC) menu (see Section A.2.3). Therefore, it is better to execute the VPC menu before executing this menu so we can avoid entering the gpdf manually. The result of the transformed variable is written in the output file as an additional column of the original data file.

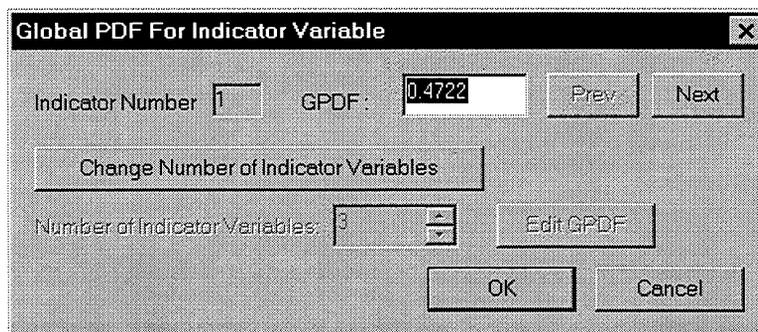


Figure 59: Global PDF dialog for normal transform of discrete variable.

### B.2.3 Vertical Prop. Curve

The Vertical Prop. Curve (VPC) menu is the facility to calculate the vertical proportion curve of a data set. Additionally, this facility also simultaneously calculates the global pdf of each facies. Therefore, this menu should be executed before the executing the process that requires the global pdf information.

Figure 60 shows the dialog of the VPC menu. We can see that the information required in this dialog is very basic, such as input-output filenames and column number of variables. After execution of the menu, a new text file is created. Figure 61 shows the example of the VPC file. Note that the sum of variables in each line is equal to 1.0.

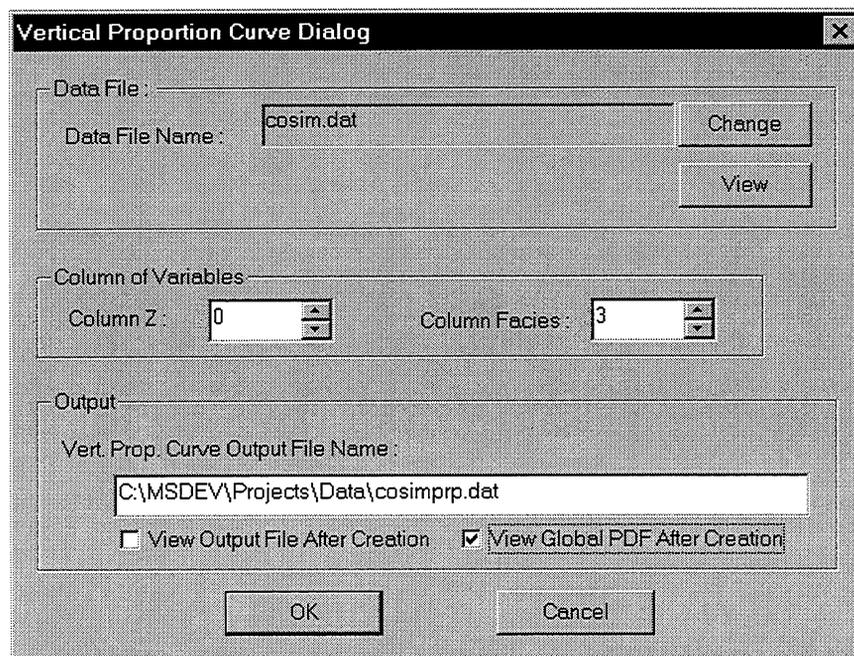


Figure 60: Vertical Prop. Curve Dialog

```

Vertical Proportion Curve of File : C:\Users\ASNUL\COSIM\Data\selfunit.dat
7
Facies : 1
Facies : 2
Facies : 3
Facies : 4
Facies : 5
Facies : 6
Facies : 7
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000
0.1667 0.0000 0.0000 0.0000 0.0000 0.0000 0.8333
0.2000 0.0000 0.0000 0.0000 0.0000 0.0000 0.8000
0.2500 0.0000 0.0000 0.0000 0.0000 0.0000 0.7500
0.6087 0.0000 0.0000 0.0000 0.0000 0.0000 0.3913
0.5556 0.0000 0.0000 0.0000 0.0000 0.0000 0.4444
0.3077 0.2308 0.0385 0.0000 0.0000 0.0000 0.4231
0.2800 0.2000 0.0400 0.0000 0.0000 0.0000 0.4800
0.0000 0.1364 0.5000 0.0000 0.0000 0.0000 0.3636
0.0000 0.1053 0.6316 0.0000 0.0000 0.0000 0.2632
0.0000 0.1000 0.7500 0.0000 0.0000 0.0000 0.1500
0.0000 0.1000 0.8000 0.0000 0.0000 0.0000 0.1000
0.0000 0.0952 0.8095 0.0000 0.0000 0.0000 0.0952
0.0000 0.0000 0.5652 0.1304 0.0000 0.0000 0.3043
0.0000 0.0000 0.5417 0.2083 0.0000 0.0000 0.2500
0.0000 0.0000 0.0000 0.0000 0.0000 0.3636 0.5000 0.1364
0.0000 0.0000 0.0000 0.0000 0.3043 0.5217 0.1739
0.0000 0.0000 0.0000 0.0000 0.3182 0.5000 0.1818
0.0000 0.0000 0.0000 0.0000 0.2381 0.5714 0.1905

```

Figure 61: Vertical Proportion Curve File Format

### B.2.4 Variogram Analysis

Variogram Analysis is the most important Pre-Simulation facility incorporated in this program. The ultimate goal of this analysis is to obtain variogram model that will later be used for simulation purposes. To make variogram model, the user can either use a raw data set followed by calculation of data variogram or use a previously calculated data variogram. Figure 62 shows the sub-menus, namely Data Variogram and Model Variogram, available in performing variogram analysis. The difference between the two menu lies on the type of data used in the modeling process as mentioned above.

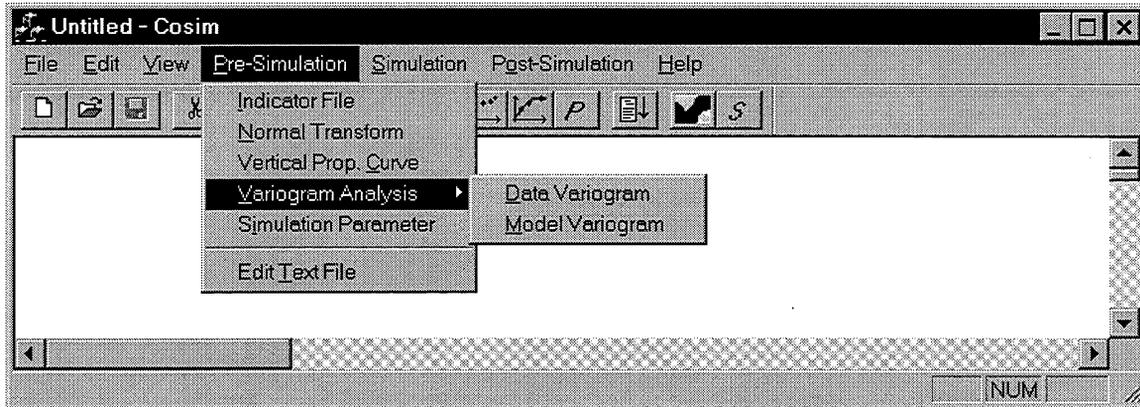


Figure 62: Data Variogram and Model Variogram Sub-Menus

Experience has shown that obtaining a good data variogram, i.e., variogram that shows some spatial structure, is not an easy task. There are several variogram parameters, such as lag distance, lag tolerance, number of lag, azimuth, azimuth tolerance, bandwidth, dip, dip tolerance, that are needed to combine to produce a good variogram. The problem is how can the user find this combination quickly to get a good variogram structure (if there is any).

In this program, the user is guided through a procedure that may help in obtaining a good data variogram quickly. The procedure is as follows.

1. Default isotropic variogram. This calculation is performed by the program using default lag parameters, i.e., lag distance, lag tolerance, and number of lags, and default bandwidth. The program automatically calculates maximum lag distance as half of the maximum distance among the data points for a given data file and uses number of lags equal to 10 as a default. Also, the bandwidth is defaulted to 2 times of maximum lag distance.
2. Isotropic Variogram. Based on default variogram, the user may need to find good isotropic variogram (if any) by changing lag parameters and bandwidth.
3. Anisotropic variogram. Calculate anisotropic variogram using lag parameters found in step 2 and default azimuth parameters, i.e., azimuth, azimuth tolerance. The program always calculates 4 azimuth directions at a time. The default azimuth directions are  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , and  $135^\circ$ , and the default azimuth tolerance is  $45^\circ$ .

4. Direction of maximum continuity. For simulation purposes, we only need to have variogram at maximum and minimum continuity directions. The maximum direction can be found from the variogram that gives the longest range. The minimum direction is assumed to be perpendicular to the maximum direction. To find these directions, user need to recalculate the variogram using different azimuth parameters. The program always calculates variogram in the set of 4 directions. The variance of the data is shown as a dotted line in each variogram as a guide in determining the maximum range.
5. Modeling the maximum and the minimum variograms. Modeling can be done interactively in the program.
6. Vertical Variograms. For 3D case, user needs to calculate vertical data variogram and model it.
7. Zonal Anisotropy. This is to convert the model from zonal anisotropy into the equivalent geometric-anisotropy.

The first dialog window that appears on the screen when user selects Data Variogram menu is shown in Figure 62. It is assumed that by selecting this menu, the user is going to calculate the isotropic variogram unless the Vertical dip plane is chosen to indicate that analysis is for vertical variogram.

Every time the user changes data file, a new lag parameters will be calculated. It is important to note here that the calculation of lag parameters takes longer if the number of data points in the file is big, e.g., in the order of thousands. Therefore, the user may notice that the process is idle for a period of time if this is case.

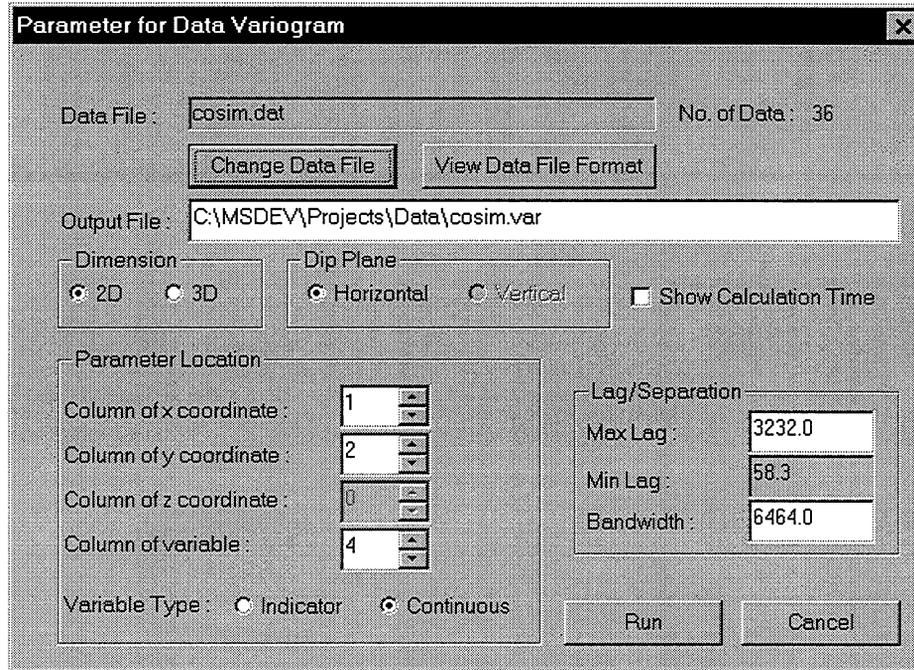


Figure 62: Data variogram dialog

The minimum lag distance reported in this dialog is the minimum distance among data points. This distance is calculated to eliminate variogram point that has lag less than the minimum distance of data points. Other parameters shown in Figure 62 are self explanatory, therefore it will not be discussed further.

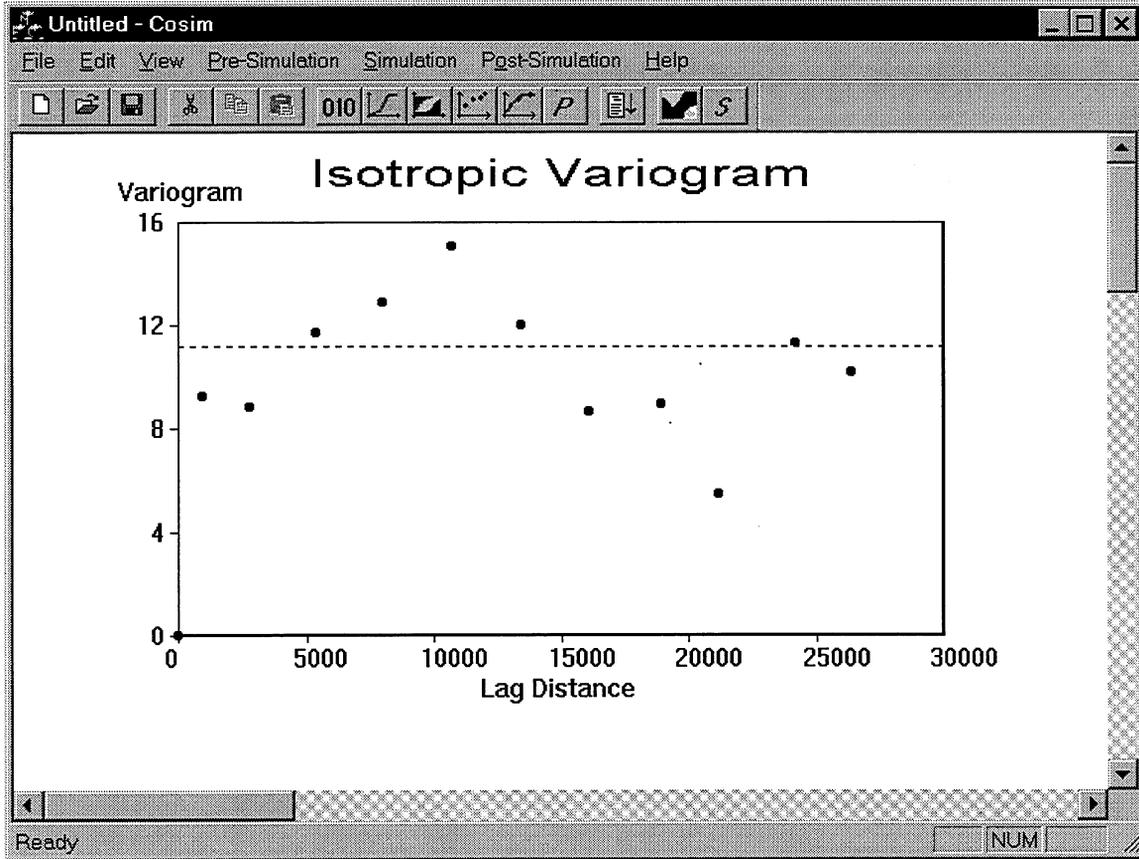


Figure 63: Example of Isotropic Variogram calculated using default lag parameters

Figure 63 shows an example of isotropic variogram calculated using default lag parameters. To improve the structure of this variogram, the user needs to make several trial-and-error procedures by changing lag parameters. To change lag parameters, the user needs to select the Change Parameters menu from the floating menu. The floating menu can be accessed by pressing the mouse's right button as shown in Figure 64. Once the Change Parameters menu is selected a dialog window shown in Figure 65 appears on the screen.

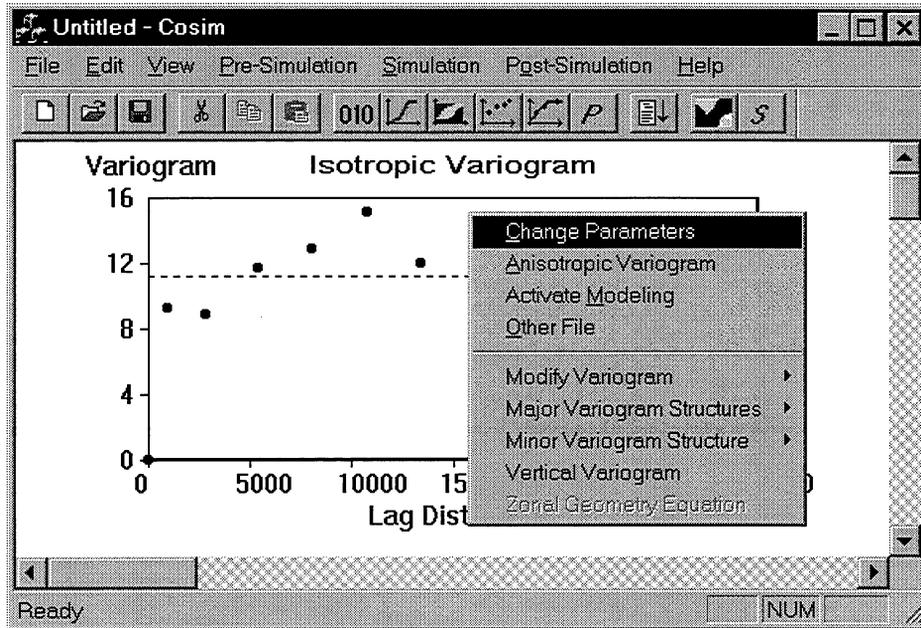


Figure 64: Floating menu to modify variogram

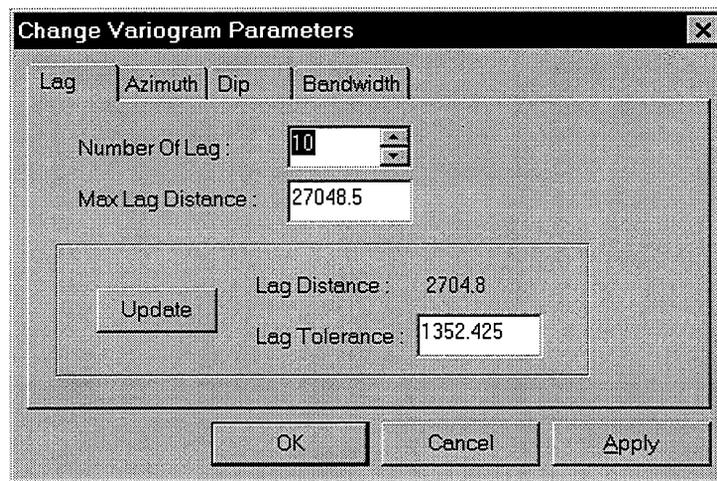


Figure 65: Change Data-Variogram Parameter Dialog

As we can see from Figure 65, this dialog allows the user to apply the changes made immediately without leaving the dialog, i.e., via the Apply button. We can see that the dialog is built using the Property Sheet/Group.<sup>Ref</sup> One characteristic of Property Group is that every time the user moves between two groups, e.g., from Lag group to Bandwidth group, the variable is automatically updated. Therefore, once the user moves to the other group, the previous information can not be

retrieved. Only the original information of the current group that can be restored when the user presses the Cancel button.

When the user presses the OK button, all information will be updated and variogram is recalculated. If the user has pressed the Apply button prior to selecting the OK button, the variogram is calculated twice for the same parameters. Therefore, in this case, it is better to select the Cancel button instead of the OK button to leave the dialog. This is especially true when calculating variogram with big data.

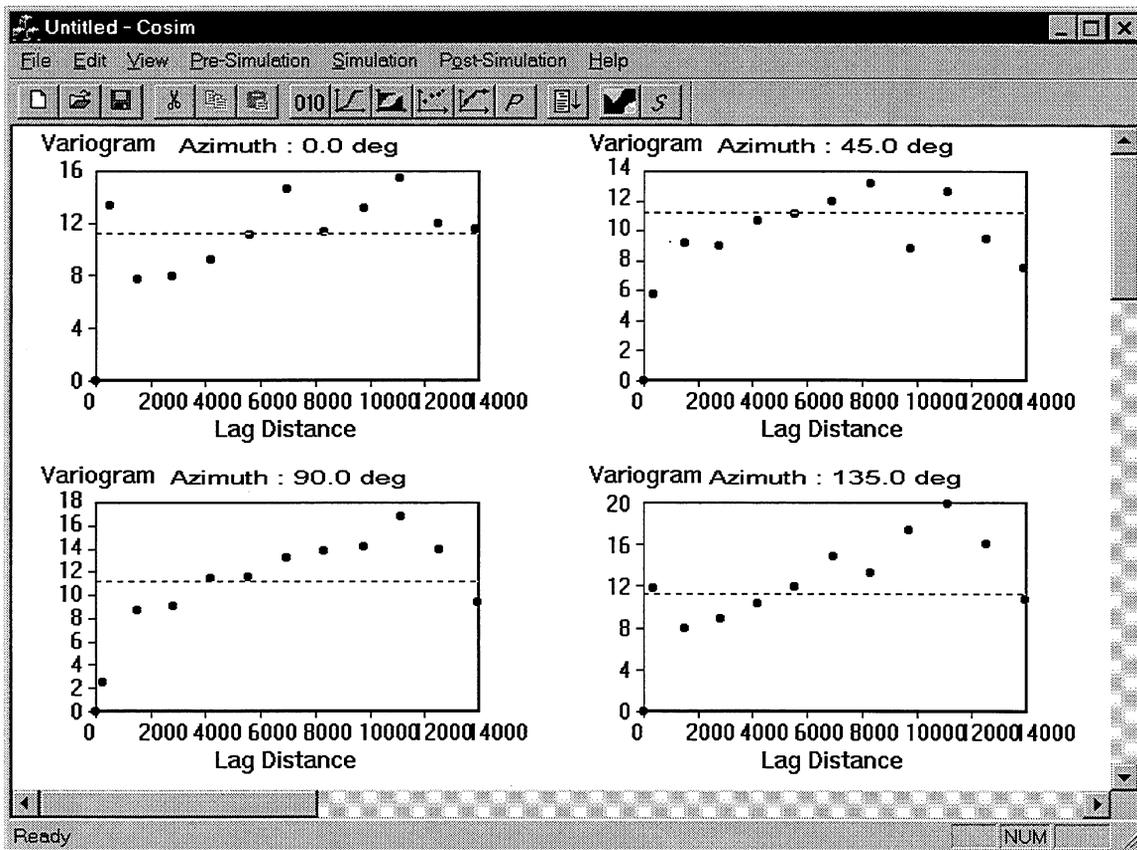


Figure 66: Example of Anisotropic Variogram using default azimuth parameters.

To calculate the anisotropic variogram, the user again needs to activate the floating menu and to select the Anisotropic menu (see Figure 64). Figure 66 shows the example of anisotropic variogram calculated using the default azimuth parameters. To obtain maximum continuity direction, user can exercise to make as many variograms (in a set of 4) as he/she wants to see by changing the azimuth parameters. The procedure to change these parameters is the same as the

procedure to change the lag parameters discussed in the previous paragraph. Once the maximum direction is found, modeling data variogram can be started by activating the floating menu and selecting the modeling menu (see Figure 64). The user needs to inform the maximum direction to the program by selecting one of the variograms. After selecting the maximum continuity direction, two variograms (maximum and minimum directions) appear on the screen as shown in Figure 67. During the modeling mode, the main's tool bars are replaced by the variogram's tool bar. Table 9 gives the description of each tool bar.

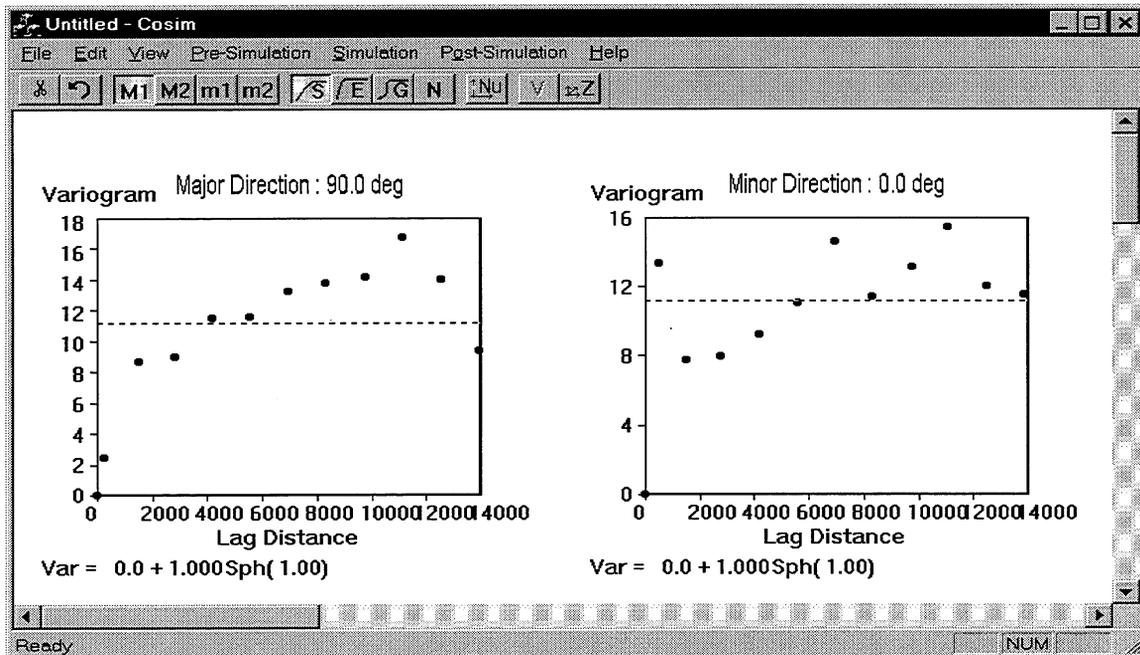


Figure 67: Program lay-out during variogram's modeling mode.

No	Tool Bar Sign	Description
1	Scissors	Clear the screen
2	Curved Arrow	Redraw variogram
3	M1	Structure 1 of Major Variogram
4	M2	Structure 2 of Major Variogram
5	m1	Structure 1 of Minor Variogram
6	m2	Structure 2 of Minor Variogram
7	S	Spherical structure
8	E	Exponential structure
9	G	Gaussian structure
10	N	No structure
11	Nu	Nugget effect
12	V	Vertical variogram
13	Z	Zonal anisotropy variogram

Table 9: Variogram's tool bar definition

To create a model, user needs to select a point inside the variogram plot area by pressing the left mouse' button. The program interprets this point as the range and sill values of the currently selected structure. Two active tool bars, one located in the second group and the other in the third group of tool bars, indicate what type of structure and which variogram structure is being selected. As we can see from Figure 67 the tool bar options for the structure type are S (Spherical), E (Exponential), G (Gaussian), or N (No Structure), and the options for the active variogram structure are M1 (structure 1 of major variogram), M2 (structure 2 of major variogram), m1 (structure 1 of minor variogram), and m2 (structure 2 of minor variogram). If user moves the mouse inside the plotting area while pressing the mouse's left button, transition model will be drawn as dashed line.

To change the nugget value, user can either carefully select a point along the vertical axis or activate the Nugget's tool bar and select a point where its vertical distance from the horizontal

axis determines the nugget value. Nugget can only be changed from major variogram to satisfy the equivalent zonal geometric requirement.

In each variogram model, the program allows the combination of a nugget and a maximum of two structures. The limitation of two structures is for simplicity (rule of parsimony) which is assumed to satisfy many cases. The structure combinations are not restricted to similar structure type only. Figure 68 shows an example of modeled variograms. Note that the program automatically presents the equation of each variogram model shown below each plot.

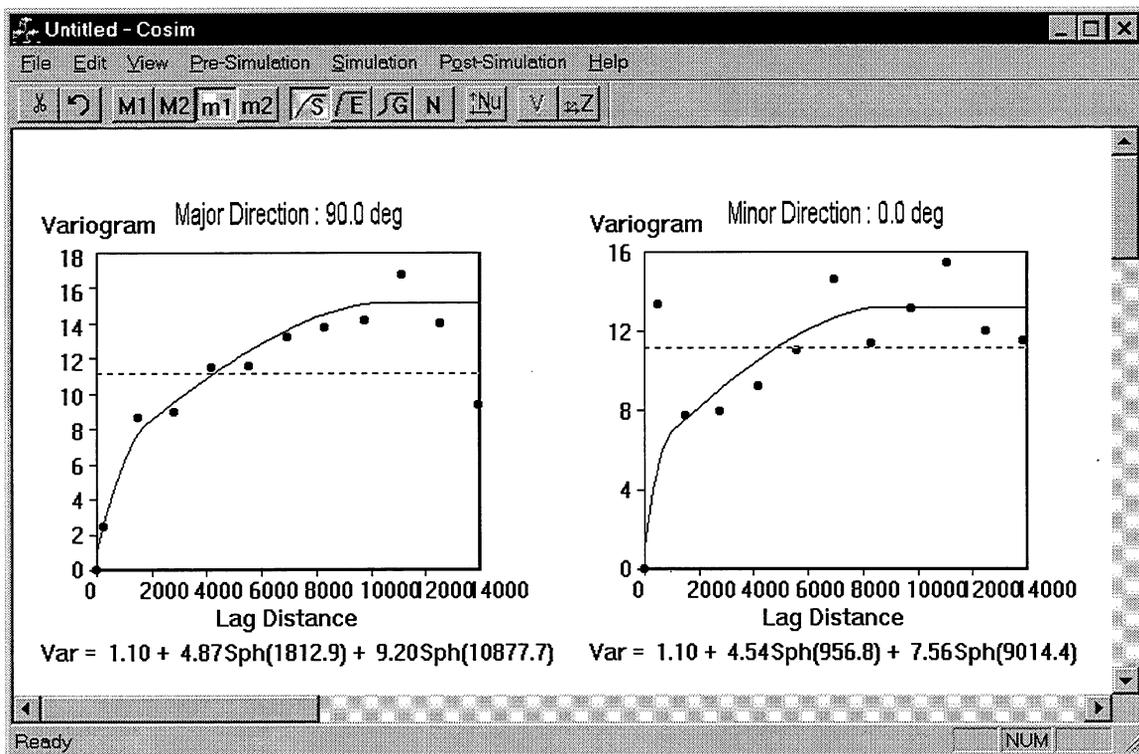


Figure 68: Example of modeled variogram.

For a 3D case, the user can proceed with the vertical variogram analysis. This can be done by selecting the vertical variogram tool bar and following the given instruction. In most cases, vertical variogram is much easier to model. This is due to the abundance of the data vertically, i.e., every 1 ft interval. The way to model the variogram is the same as discussed in the previous paragraphs.

As shown in Figure 68, the result of the analysis presented in the previous paragraphs are two independent, or three for 3D case, variograms, i.e., major, minor, and vertical variograms. These variograms satisfy the zonal anisotropic characteristics. The simulation requires that the model satisfy the geometric anisotropy characteristic only. To convert variograms model from zonal anisotropic into equivalent geometric-anisotropy, a procedure named as Zonal Anisotropy is implemented in the program. To perform procedure, user needs to select the Z tool bar. Figure 69 shows the dialog for zonal anisotropy procedure. The user needs to formally accept the model by pressing the “Accept Model” button and provides the name of the variogram so it can be assigned properly into simulation parameter. The name of the variogram is explained in the following paragraph.

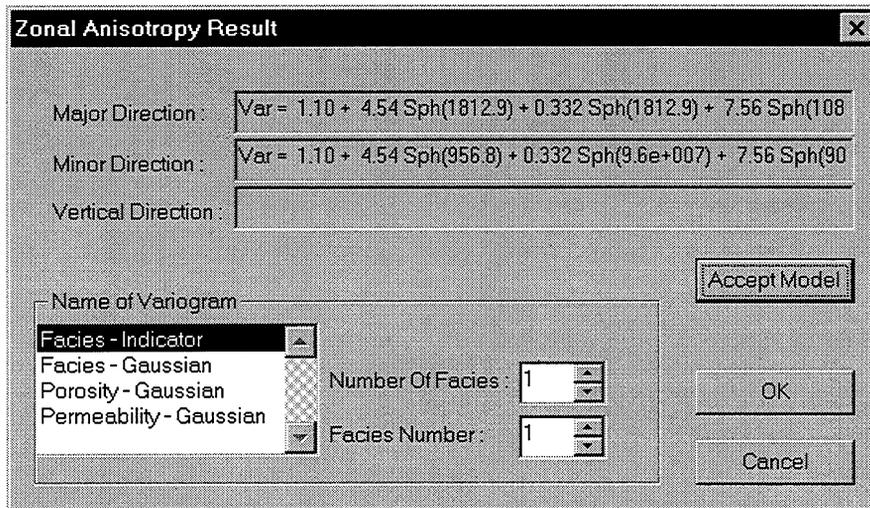


Figure 69: Zonal anisotropy dialog.

The number of variograms required in the simulation routine depends on the source of the proportion curves and how the permeability is simulated. Table 10 presents the simulation mode that determines the number and name of variograms needed based on the source of proportion curves. From this table, we can see that there are 3 variograms type required, namely Facies-Indicator, Facies-Gaussian, Porosity-Gaussian. Facies-Gaussian can be replaced by deducing it from one of the Facies-Indicator. Therefore, for Mode-2 (see Table 10), the number of variogram can be reduced by 1, by specifying in the simulation parameter menu that Facies-Gaussian variogram is to be deduced from one of the Facies-Indicator variogram.

Mode	Source of Proportion Curves	Variogram Needed/Name
1	Vertical Proportion Curve only or From an Input File.	Facies and porosity Gaussian variograms.
2	Indicator Kriging, calculated in the simulation run.	<ul style="list-style-type: none"> <li>• Each facies variogram in its original space, named as facies-indicator variogram</li> <li>• Facies and porosity Gaussian variograms.</li> </ul>

Table 10: Variogram requirement for simulation run.

Additional Gaussian variogram, named as Permeability-Gaussian, is also required if permeability to be simulated using the sequential Gaussian simulation. This can be incorporated only if the location of the permeability data are known and are specified in the data file.

### B.2.5 Simulation Parameter

The Simulation Parameter menu is the facility to compile all parameters required for simulation process. The dialog window for this facility is built using the property group as shown in Figure 70. As can be seen from this figure, there are 8 groups of parameters that the user need to provide, namely Data File, Grid Block, Vertical Proportion, Kriging, Output File, Permeability File, Simulation, and Variogram.

The default of the data file format is the GEOEAS format unless it contains the soft data. Soft data is the a priori assumption, assigned as probability, of facies type at certain location. If the data consists of soft data, then the format of the data file has to follow the format, namely COSIM format, described in the following paragraph.

In COSIM format, only 6 variables are allowed, namely x-coordinate, y-coordinate, z-coordinate, facies, porosity, and permeability. If the facies at a location is a hard data then it should be input normally. But, if it is a soft data then a negative value should be assigned for facies. Porosity and permeability values at this location is irrelevant since this location will be assumed as unsampled

location during the sequential Gaussian simulation. Following the permeability value, a set of probability values need to be assigned for each facies. If the probability is only known for part of the facies then a negative value should be assigned for the unknown facies. Figure 71 shows an example of COSIM data format. Note, that permeability data are absent. This can be informed to the program by specifying 0 as the column number for permeability as in the GEOEAS format.

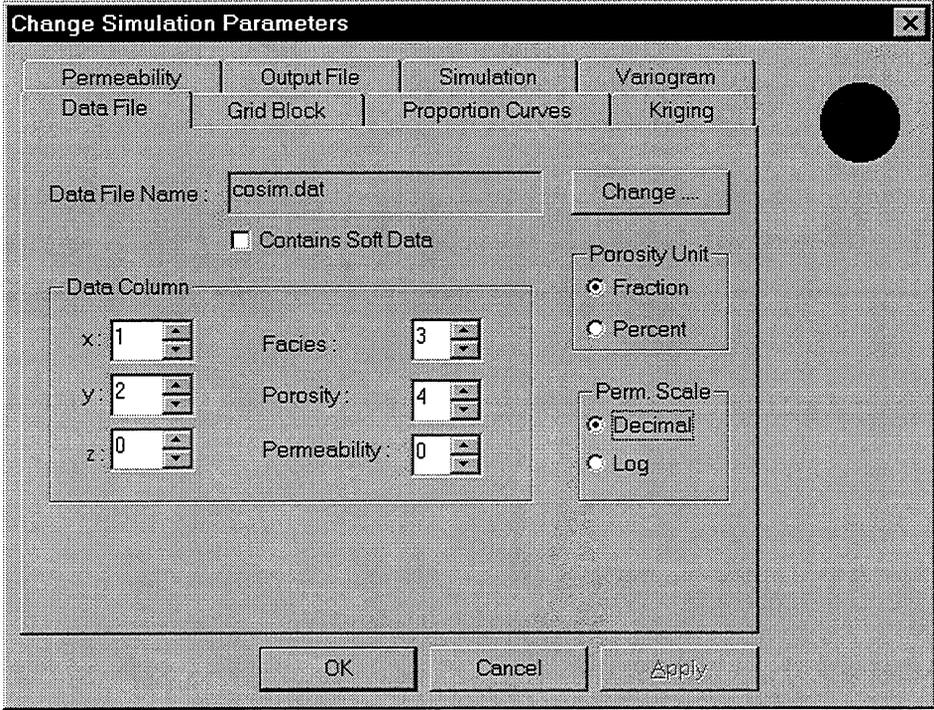


Figure 70: Simulation Parameter Dialog

Soft Data Example							
5	x	y	z	Facies	Porosity		
2583829	358331	122	1	0.1291			
2583829	358331	122	1	0.1023			
2583829	358331	123	1	0.1215			
2583829	358331	124	1	0.1176			
2583829	358331	125	1	0.1397			
2583829	358331	125	2	0.1401			
2583829	358331	126	2	0.1892			
2583829	358331	125	3	0.0492			
2583829	358331	125	3	0.0282			
2583829	358331	126	3	0.0591			
2583829	358331	142	-1.0	0.1000	0.50	0.50	0.00
2583829	358331	143	-1.0	0.1000	0.50	0.30	0.20
2583829	358331	144	-1.0	0.1000	0.50	-1.00	-1.00
2583843	357851	132	-1.0	0.0600	0.70	-1.00	-1.00
2583843	357851	133	-1.0	0.0600	0.70	-1.00	-1.00

Figure 71: Example of COSIM data format that contains soft data

In the example shown in Figure 71, the number of facies is 3. Using this format, we can quickly identify which data are hard data and which are soft data as there are additional columns when they are soft data. Also, note how the probability values should be assigned when the probability is only known for part of the facies.

For a location where facies is identified but either porosity or permeability or both are not known then a negative values should be used. This is to maintain the consistency of the data file.

The Grid Block group specifies the dimension of the model. The required information is number, size, and origin of the grid blocks. The origin of the grid block or the minimum coordinate is defined as the middle point of the first grid block. For example, if the origin of the coordinate system for the whole model is located at (0,0,0) and the size of the grid blocks in x, y, and z directions are 66, 66, and 1 ft, then the minimum coordinate is (33, 33, 0.5). Figure 72 shows the Simulation Parameter Dialog for the Grid Block group.

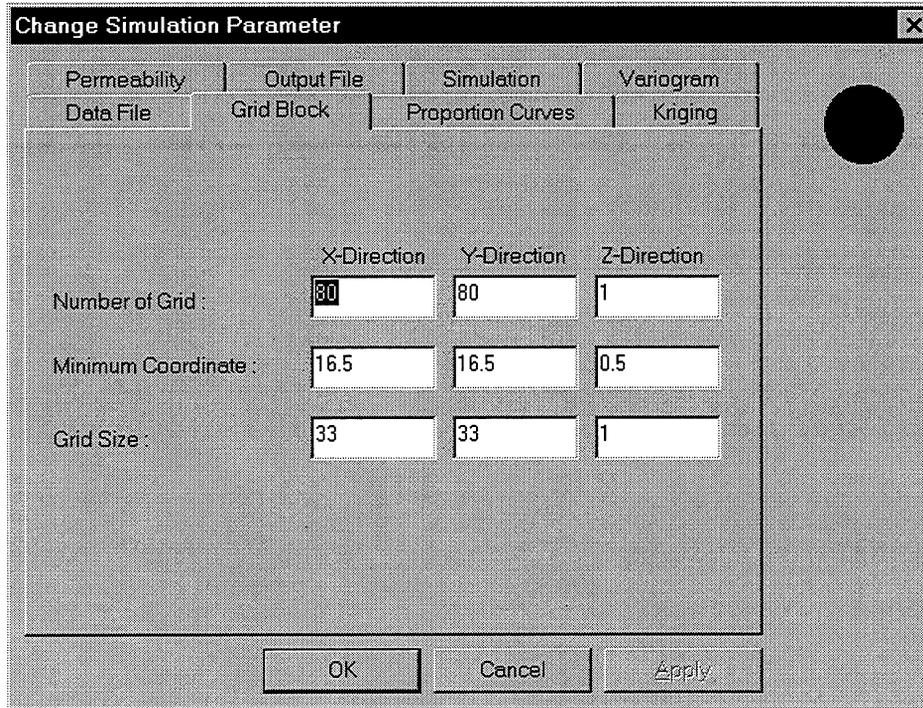


Figure 72: Simulation Parameter Dialog for Grid Block Group

The third group in the Simulation Parameter dialog is the Proportion Curves group as shown in Figure 73. There are 3 options that can be used as the source for the proportion curves. They are

1. direct calculation using indicator kriging,
2. use vertical proportion only, or
3. input from a file.

If the user chooses the first or the second option, the vertical proportion file is required. This file is required for first option to overcome the singularity problem during kriging. This file can be generated using the Vertical Prop. Curve facility (see Section A.2.3). The last option is provided so the user can either use independent method in calculating proportion curves or use proportion curves from the previous run. The program stores the proportion curves created using indicator kriging in a binary file named as “prpcrv.tmp”. User can use this file for future simulation run. This file contains the cumulative distribution function (cdf) of the kriging result for the first

$n - 1$  facies, where  $n$  is the number of facies, for each grid block. The grid block cycles fastest in  $x$ , then in  $y$ , then in  $z$  directions.

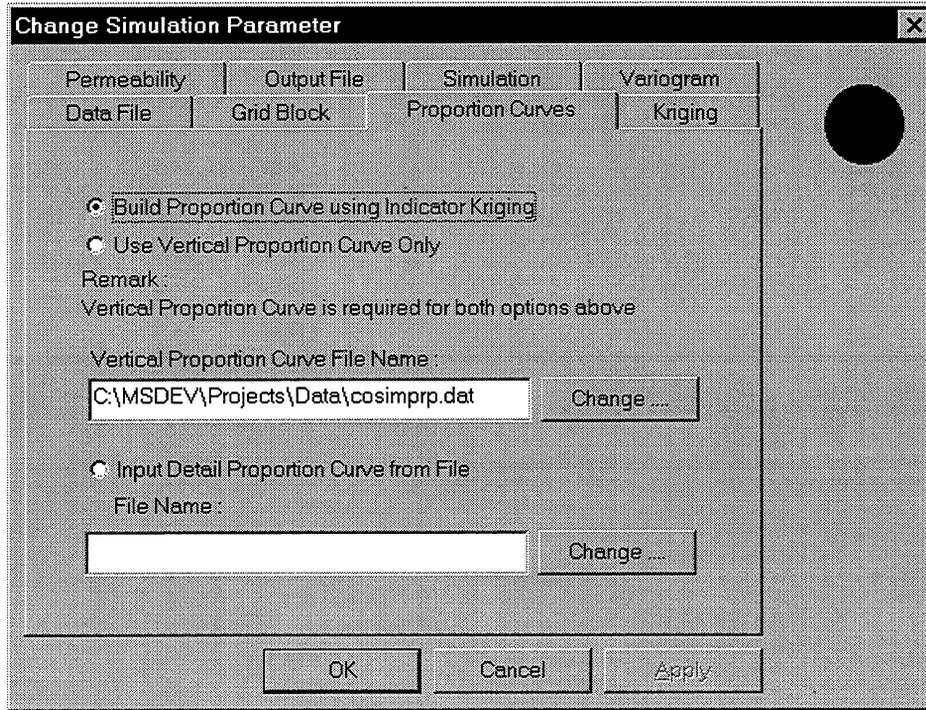


Figure 73: Simulation Parameter Dialog for Proportion Curves Group

The fourth group in the Simulation Parameter Dialog is the Kriging group. This group contains the parameters that control the kriging process, such as search neighborhood definition and the number of minimum and maximum points for kriging. Figure 74 shows the group dialog. The search neighborhood defines the ellipsoid around the unsampled location that is used to find the sampled point or the previously simulated point.

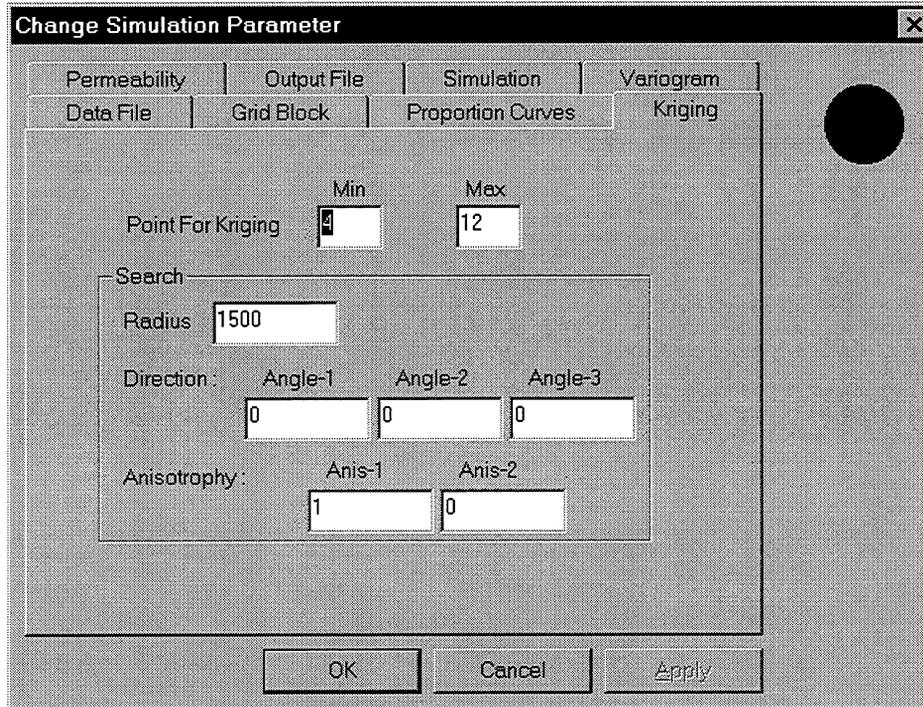


Figure 74: Simulation Parameter Dialog for Kriging Group

The fifth group in the Simulation Parameter Dialog is the Permeability Group. Using this group, the user can identify whether or not to perform the simulation of permeability. If permeability data are not written together with the data file then a separate permeability correlation is required. This file should be written in GEOEAS. Figure 75 shows an example of the correlation file for permeability simulation purposes.

The sixth group in the Simulation Parameter Dialog is the Output File group. The only information required is the output file name. The program generates  $n + 1$  output files, where  $n$  is the number of realizations. In each of the  $n$ -files, the simulation result for realization number  $i$ , where  $i \leq n$ , is written in the binary format in the following order, facies, porosity, and permeability (if simulated). The ASCII version of the simulation output can be generated using the Post-Simulation facility (see Section 5.4). The extra file is created to store the grid block information and the value range (max. and min.) of each attribute in each realization. This file is useful for Post-Simulation purposes, e.g., drawing the 2D cross section.

The given output filename will be used as the name of the extra file. Meanwhile, the name for each realization file will be given as the input file name plus an additional two letters that determines the realization number. The letters added are 'R' and 'i', where  $i$  is an integer  $\leq n$ . For example, if the input file name is "cosim.ocm", then the file name for the output of realization number 1 is "cosimR1.ocm". The extension "ocm", short for output cosim, is used as a default extension.

Facies-Porosity-Permeability		
Correlation		
3		
DGI		
Porosity		
Permeability		
1	0.1700	25.00
1	0.1500	25.00
1	0.1600	10.00
1	0.1500	20.00
1	0.1800	40.00
2	0.0900	30.00
2	0.1400	15.00
2	0.1900	55.00
2	0.1800	35.00
2	0.0700	10.00
3	0.1400	75.00
3	0.1400	50.00
3	0.1450	25.00
3	0.1500	25.00
3	0.1500	40.00

Figure 75: Example of permeability correlation file

The seventh group in the Simulation Parameter dialog is the Simulation Group. The parameters given in this group are used in the sequential Gaussian simulation part of the co-simulation procedure. The required parameters can be seen in the group dialog as shown in Figure 76. Most of the default parameters in this group should work very well in many cases. Therefore, it may left unchanged, unless multiple realizations are required.

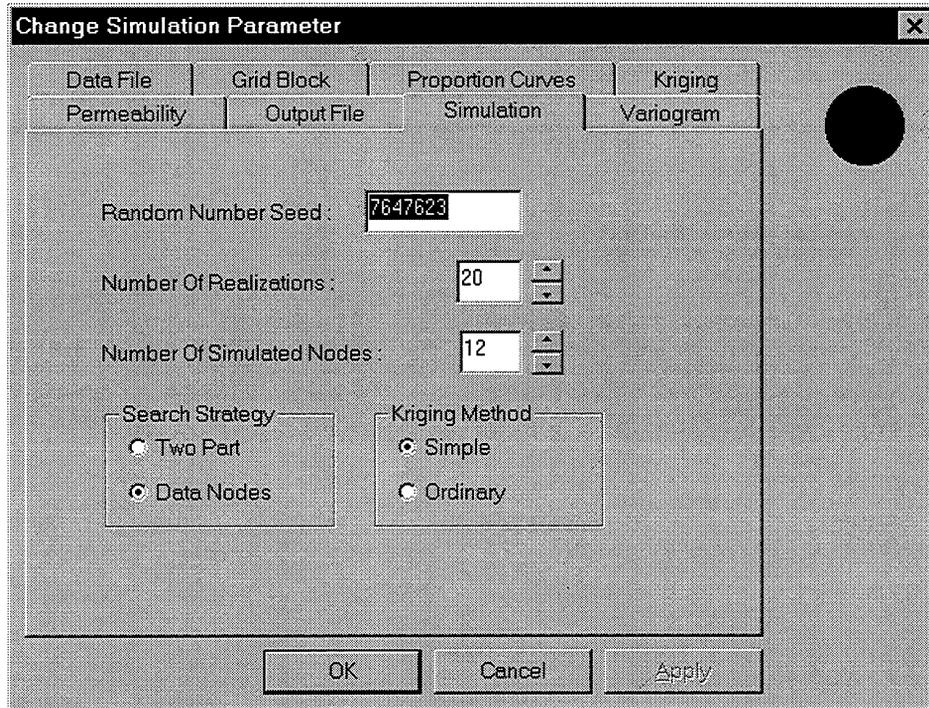


Figure 76: Simulation Parameter Dialog for Simulation Group

The last group in the Simulation Parameter dialog is the Variogram group. If Variogram Analysis menu and Vertical Prop. Curve menu, which simultaneously calculates the global pdf, are successfully performed prior to executing this dialog, all parameters should be correct already. Therefore, the user needs only to verify the results. However, any changes can be made independently through the dialog of this group if necessary. Figure 77 shows the dialog of the Variogram group.

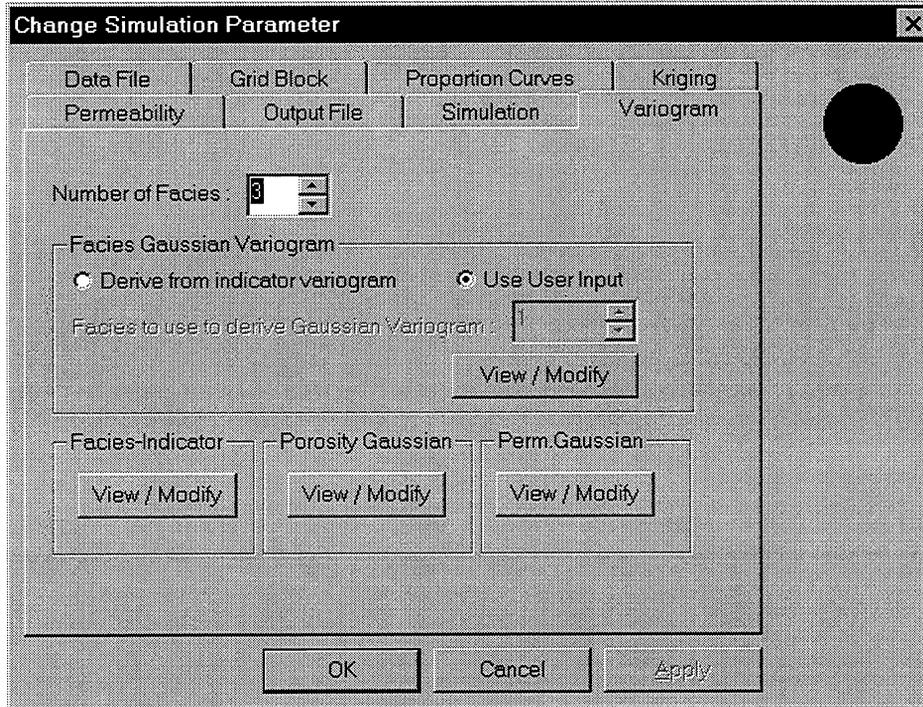


Figure 77: Simulation Parameter Dialog for Variogram Group

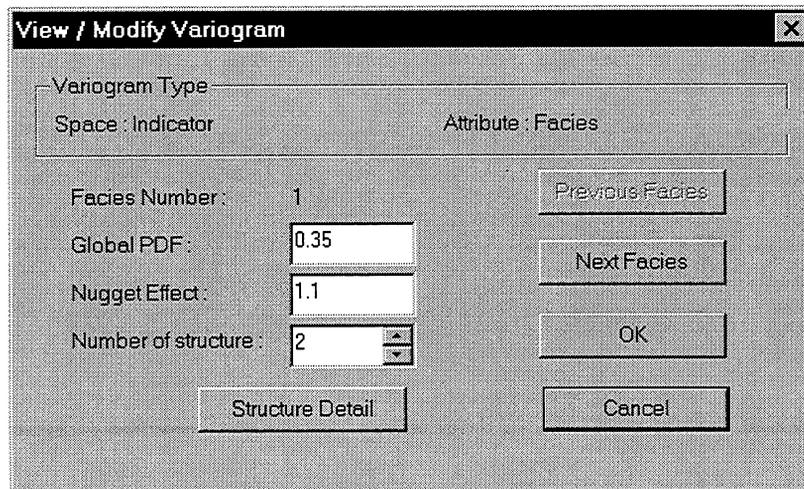


Figure 78: Variogram Model Dialog

As shown in Table 10 there are 3 or 4 variogram types required in the program, namely Facies-Indicator variogram, Facies-Gaussian variogram, Porosity-Gaussian variogram, and Permeability variogram (if necessary). Additional dialog window will appear on the screen when the user wants to view the detail of the model of each of these variograms. The example of this dialog is

shown in Figure 78. The “Previous” and “Next” button are provided on the dialog to view the model of each facies. Furthermore, to see structure detail of each model, another dialog window is required as shown in Figure 79. To navigate from one structure to the other structure, the user needs to press the “Previous Structure” and/or “Next Structure” buttons.

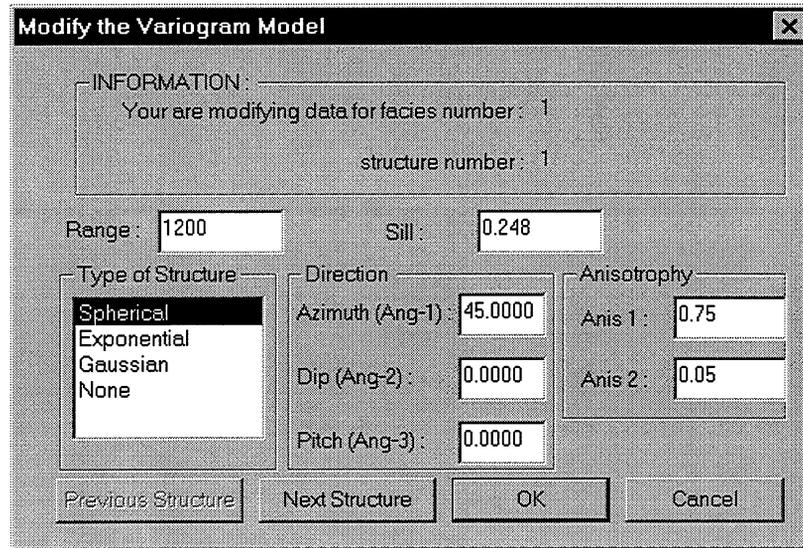


Figure 79: The structure detail of variogram model

### B.3 Simulation

The Simulation is the menu to use when the user is ready to execute the simulation routine using the parameters specified in the Parameter File menu (see section A.2.5). After selecting this menu, a dialog shown in Figure 80 appears on the screen. The user needs to specifically press the Execute button to run the simulation. This is to provide the user with the option of canceling the simulation, i.e., by pressing the Windows’ exit button at the upper right corner of the dialog window.

Once the simulation is executed, the program does not have any way to cancel the operation except to use the Windows’ Close Program/End Tasks mechanisms. To avoid the lost of information set previously, it is advised that the user save the document file (\*.csm) prior to executing simulation run.

During simulation run, the program shows the progress of the simulation. There are 5 major progress steps shown. The first progress is shown when the program finishes with program initialization and data reading. The second progress is shown while the program is building the proportion curves. The third progress is shown when the program finishes with performing the normal transform routines and preparing the back transformation tables. The fourth step is shown while the sequential Gaussian simulation is being performed. The last progress is shown while the program is performing the back transformation procedure.

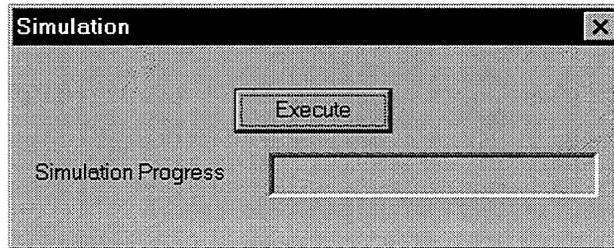


Figure 80: Beginning of simulation dialog.

Once the simulation is finished, the message box appears on the screen to inform the user that the simulation is complete. This dialog also shown the total simulation time. Figure 81 shows an example of the End of Simulation Dialog.

The output of each realization is written out into a binary file specified in the parameter file (see Section A.2.5). In this file, a stream of an integer and 2 floating numbers (or 1 if permeability is not simulated) are written for each grid block. The output are cycling fastest on  $x$ , then  $y$ , then  $z$ .

The statistics, i.e., global pdf and porosity/permeability averages, of each realization is calculated simultaneously. These information are saved in the drawing information file. Additionally, for multiple realizations, the rank of these realizations is calculated based on the comparison between the simulated and data pdf of the main facies. Main facies can be user input or automatically determined by the program from the highest data pdf.

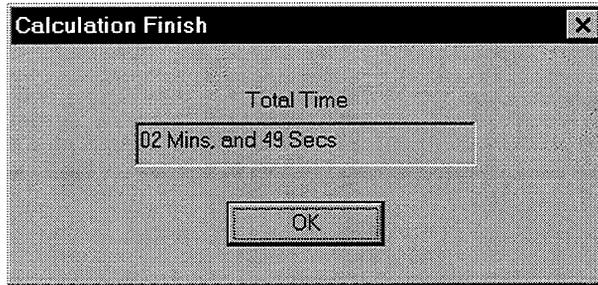


Figure 81: End of simulation dialog.

#### B.4 Post-Simulation

The Post-Simulation is the facility to help the user in evaluating the simulation result. There are 3 sub-menus available from this menu, namely 2D-Cross Section, 3D-File, and Statistics. Figure 82 shows the appearance of these sub-menus.

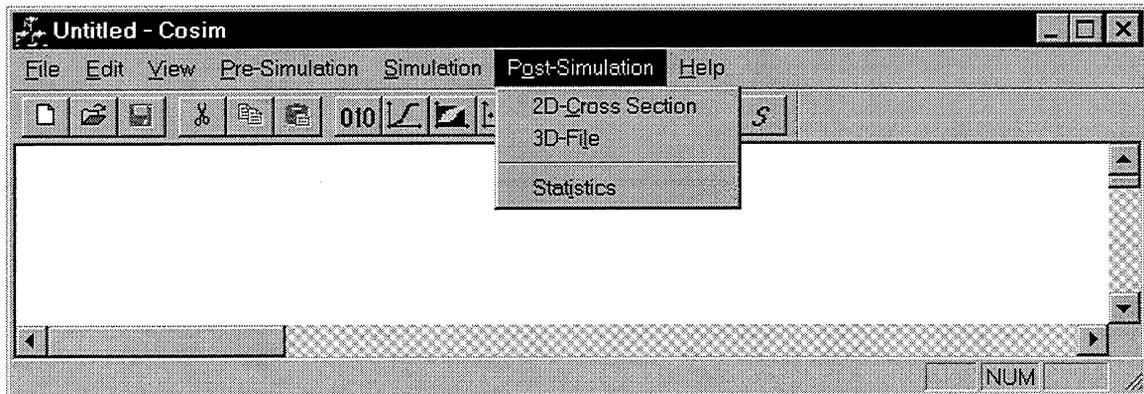


Figure 82: Post-Simulation Sub-Menus

The 2D-Cross Section enables the user to view any cross section of the simulation result directly from the screen. A dialog window that contains the parameter to be used to draw the cross section will appear on the screen upon the selection of this menu. Figure 83 shows this dialog.

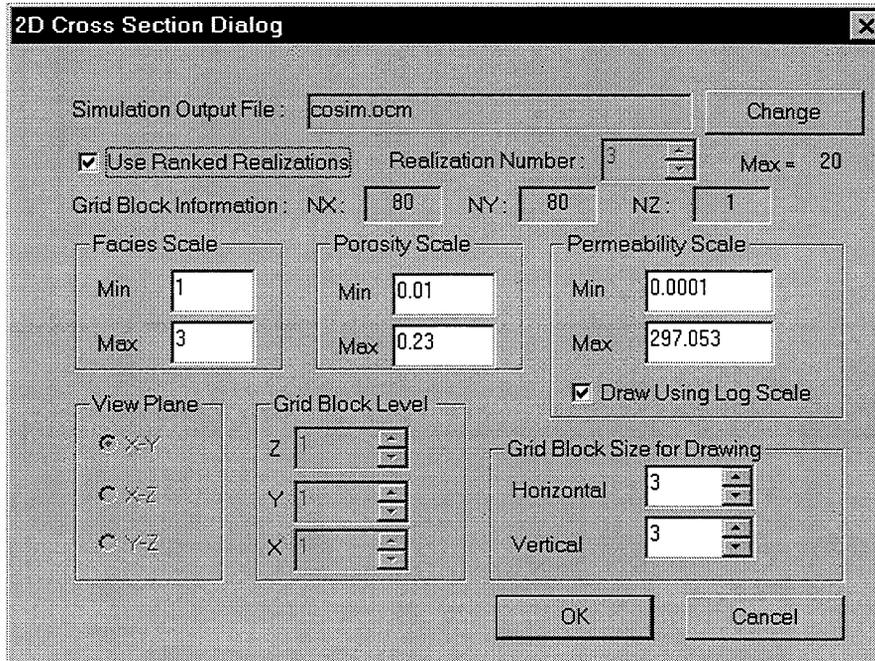


Figure 83: 2D-Cross Section Window Dialog

The first step in using this dialog is to assure that the output filename is correct. As explained in Section A.2.5 for each output file, the program has automatically generated a drawing information file and  $n$ -realization files. The drawing information file contains the grid block information, i.e.,  $nx$ ,  $ny$ , and  $nz$ , of the model used to produce the simulation result. This information should be fixed for each set of output files and should not be changed. User also needs to select the realization to view. It could be any realization number or the ordered according to the rank of the realizations.

Additionally, the drawing information file also contains the maximum and minimum values of each attribute (facies, porosity, and permeability), followed by the statistics of the output. All of these informations are given for each realization. The statistics reported are the global pdf and the average values of porosity and permeability, for each facies. The maximum and minimum values of each attribute may be used as the minimum and maximum scale of the drawing. The user may give different set of values for the scale of the drawing by changing the appropriate edit boxes.

The cross section that can be viewed using this program is the XY, XZ, or the YZ plane at any level/depth. The level/depth of the plane is measured from the origin of the coordinate system.

The unit size of each grid block (horizontal and vertical) drawn on the screen is given as the size of screen pixel. For example, for an XY plan view with  $n_x = 40$  and grid block size = 5, the total horizontal length of the drawing will be 200 pixels. The program will cut portion of the drawing if the total length at each side (horizontal and vertical) exceeds the available drawing area. User needs to customize this size to match with the application. Figure 84 shows an example of 2D-Cross Section.

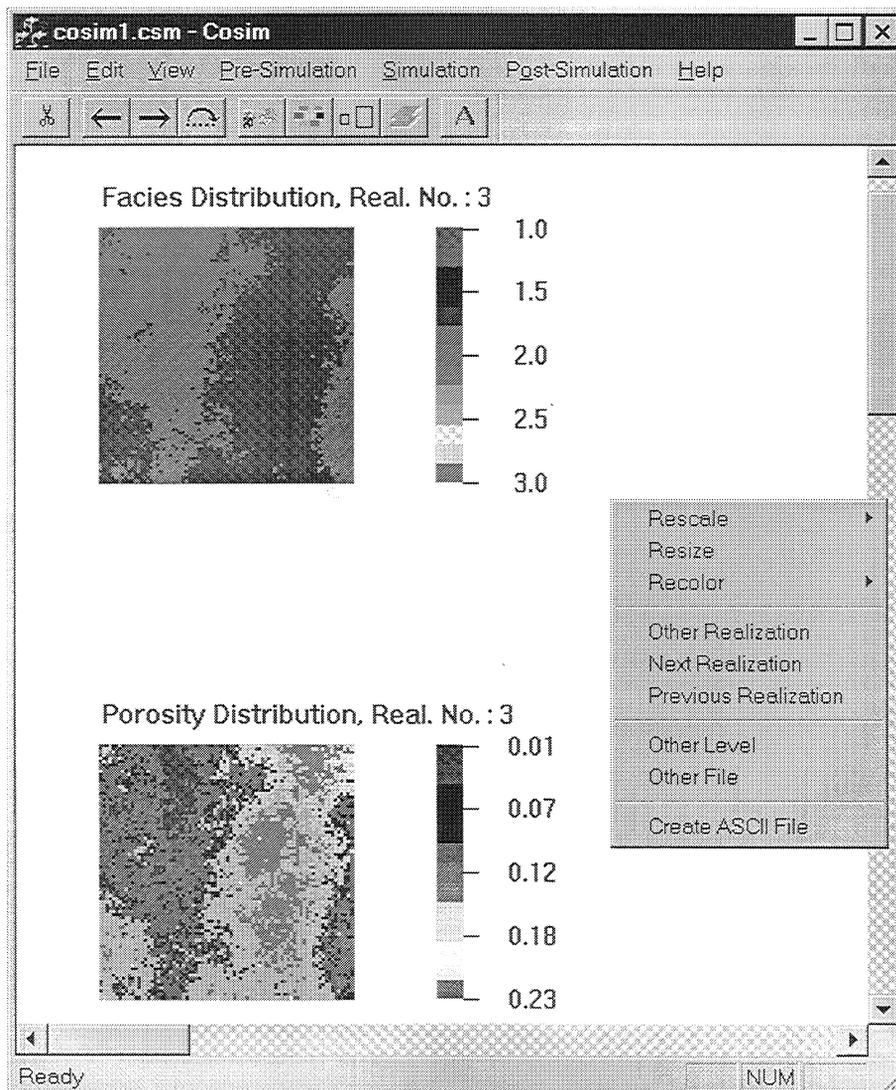


Figure 84: Example of on-screen 2D-Cross Section.

After an image is drawn on the screen, the user has the flexibility to change several drawing attributes, such as color scheme, plot scales, screen-grid block size. To change this attribute, the user first needs to open the floating menu as shown on the right side of Figure 84 or use the available tool bars. The floating menu can be activated by pressing the mouse's right button. Similarly, to see other realization, user can use the provided tool bar or accessing through the floating menu.

Every time a cross section is shown, user can create the ASCII text file of the image generated. This file may be useful to use with other drawing package. The file contains 4 or 5 (depends on whether the permeability is simulated or not) columns of data in the format shown in Figure 85.

x-coordinate	y-coordinate	facies	porosity	permeability
--------------	--------------	--------	----------	--------------

Figure 85: ASCII Format of the Output Result

The direct on-screen view of the 3D image has not been implemented in the latest version of COSIM. Therefore, the facility provided in this menu is limited only to the creation of the secondary output file that is suitable for viewing with other commercial package, such as the Spyglass-Slicer program, or the regular output file in GEOEAS format.

The dialog to create the secondary output file is shown in Figure 86. As we can see that all parameters required in this dialog are self explanatory. Therefore, it will need not further explanation. The program will automatically generate separate output file when the Slicer format is selected. In this case, the output file names are “\*f.out”, “\*p.out”, and “\*k.out”, for facies, porosity, and permeability, respectively, where ‘\*’ is the output file name given by the user. For GEOEAS format, only one output file will be created with file name equal to the simulation output file name as given by the user in the Simulation Parameter but with different extension. The extension for this file is “out”.

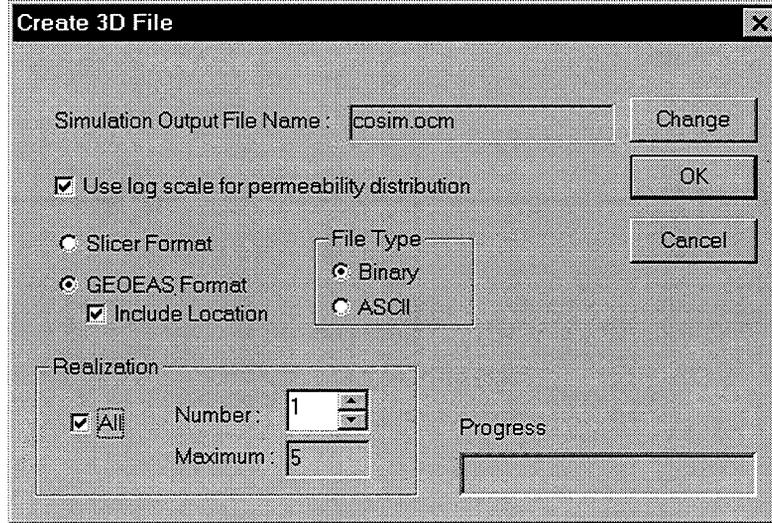


Figure 86:3D Spyglass-Slicer File Dialog

The Statistics option of the Post-Simulation menu provides the user with the statistics of the simulation result. This includes the comparison of global pdf of each facies between conditioning data and simulation result, and the average of porosity and permeability of the simulation result for each facies. Figure 87 shows an example of the statistics output. For multiple realizations, the results are shown for the first 5 realizations only. It is ordered according to the rank (see Section A.3).

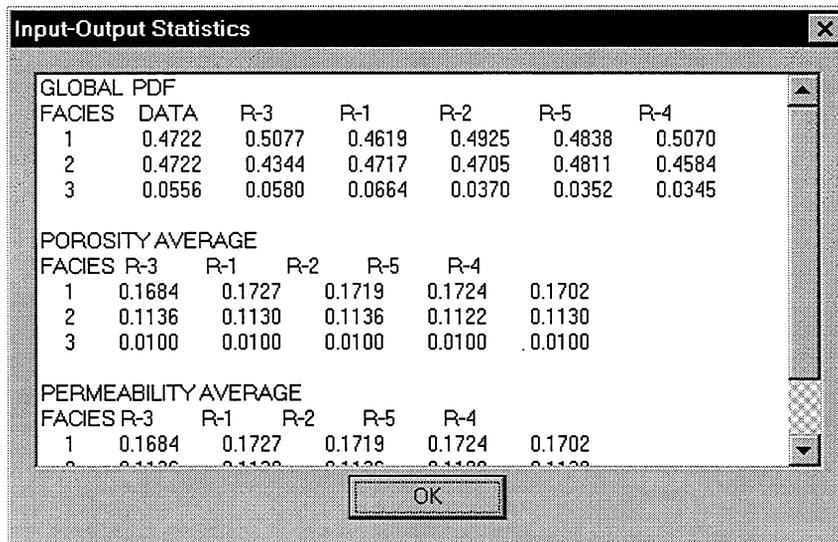


Figure 87: Statistics of the simulation result.

## **B.5 Case Study - Example**

In this section a case study as a complete example of using the program is presented. For simplicity, the example shown here is limited to the 2D case so user can follow it without waiting for too much time for the computer to perform the simulation. A 3D case example is also included in the distribution disk, but it will not be discussed here. The example is shown as the step-by-step procedure in the sections that follows.

### ***B.5.1 Data Preparation***

The main input data required is the facies, porosity, and permeability values, at well locations. These data can be combined into a single file using the GEOEAS format. In this example, the location of the permeability data are not known so the correlation is written as a separate file. This also means that we can not calculate the variogram for permeability. The program will use conditional distribution technique with pure random number to generate permeability simulation. It is recommended that all data file be named with extension “dat”, since this is the default extension that the program uses for data file.

Files “cosim.dat” and “cosimfpk.dat” in the distribution disk contain the data needed in this case study. The first data file contains the well data for facies and porosity whereas the second data contains the correlation among facies, porosity, and permeability, without location information. There are 3 facies defined in the data file named as facies 1, 2, and 3. By opening file “cosim.dat” using the “Edit Text File” menu found in the Pre-Simulation facilities, we can see that

1. there is no soft data in the file, since all facies are positive,
2. column numbers for  $x = 1$ ,  $y = 2$ ,  $z = 0$  (since it does not exists in the file), facies = 3, and porosity = 4, and
3. porosity is expressed as fraction, not as percentage.

Meanwhile, from file “cosimfpk.dat” we can also see that column numbers for facies, porosity, and permeability are 1, 2, and 3, respectively.

Prior to executing the COSIM program, the user needs to know the dimension of the reservoir and plans the appropriate grid block model. Well map of the reservoir may help in planning this model. Figure 88 shows the well map and facies distribution used in this case study. From this figure we can see that a model with the dimension of 2640 x 2640 ft<sup>2</sup> is appropriate. The number of grid blocks to be used in this example are  $n_x = n_y = 80$ . Since this is the 2D case, then  $n_z = 1$ . Using these number we can calculate the grid block size to be 33 x 33 ft<sup>2</sup>. For this problem, the origin of the coordinate system is located at (0,0), therefore the minimum coordinate becomes (16.5,16.5), i.e., half of the grid block size measured from the origin.

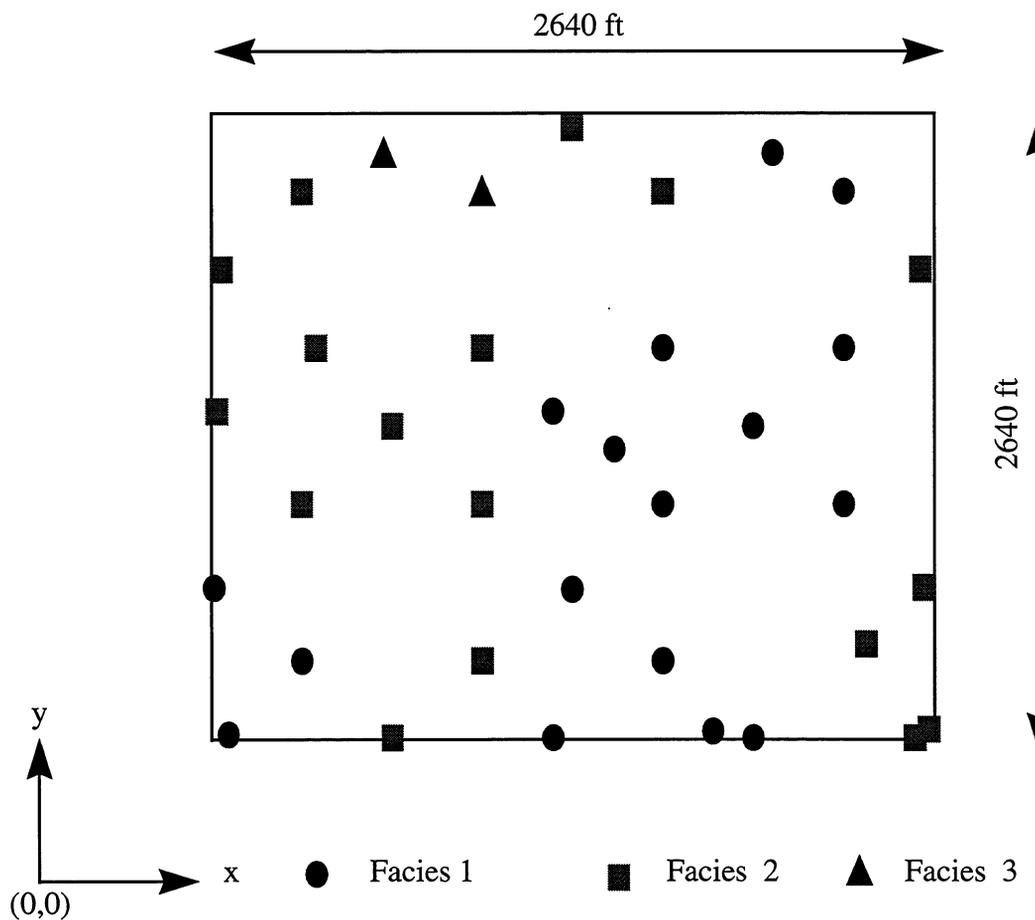


Figure 88: Well map and facies distribution of the case study data

### B.5.2 Vertical Proportion Curves and Global PDF Calculations

These calculations are performed simultaneously in the program via the Vertical Prop. Curve menu. Applying this menu for the “cosim.dat”, we will obtain the results as shown in Figure 89 and in Figure 90. Note that, for 2D case, both Global PDF and Vertical Proportion Curves are exactly the same as shown in these two figures. Note that a file named “cosimprp.dat” is created as the default output file name for the vertical proportion curves file and it will be automatically stored into the simulation parameter.

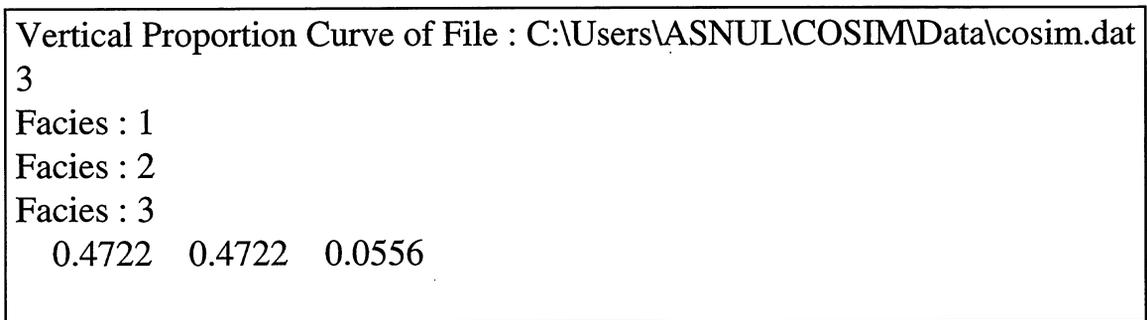


Figure 89: Vertical Proportion Curve for “cosim.dat”

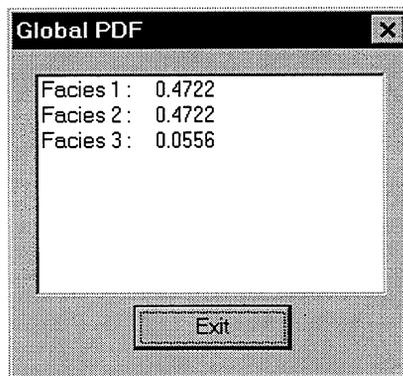


Figure 90: Global PDF for “cosim.dat”

### B.5.3 Indicator File and Normal Transform

As explained in Sections A.2.1 and A.2.2, before we can proceed with variogram analysis we need to have the data file of the indicator function and of the normal transform. Therefore, we need to execute these two menus to obtain these files. To create the indicator file, the only information required is the column number of the facies data. From file “cosim.dat” we know

that facies is located on column number 3. Therefore, we can proceed with this information and produce the required file. The default output file name for the indicator file is the original data file plus a word “ind” added to the end of the file name (before the extension name). In this case, the output indicator file name will be “cosimind.dat”. To see the result of the output indicator file immediately, the user needs to check the “View File After Creation” box before hitting the “Create File” button (see Figure 56). Note that the column numbers of the indicator variable in file “cosimind.dat” of facies 1, 2, and 3, are 5, 6, and 7, respectively. We will need these column number in variogram analysis.

Similar procedure can be followed to produce the normal transform file. If the Gaussian variogram of the facies is to be modeled directly from the Gaussian data, we need to perform the normal transform procedure twice, one for facies and the other for porosity. Otherwise one process is enough (porosity only). The other way to generate Gaussian facies’ variogram is to deduce it from one of the Facies-Indicator variogram (see Section A.2.4).

The column number of porosity in “cosim.dat” is 4. Using this information together with the “Continuous” as the variable type (see Figure 58), we can get the normal score transform of the porosity. The default extension of the normal score transform file name is “ntr”. In this example, the output file for normal score transform of porosity is “cosimporo.ntr”. The column number of transformed variable in the output file is equal to 5, since the data file contains 4 columns of data.

The normal transform of the facies can be generated using column number = 3, and “Discrete” as the variable type. As explained previously, for discrete variable the program needs the global pdf information of the data. But, since we have done this calculation in the previous step, the program already knows these values. Therefore, we do need to re-enter them. The output file of the normal score transform of facies data is named “cosimfacies.ntr”. As in the porosity case, the column number of the transformed facies in the output file is equal to 5.

#### ***B.5.4 Variogram Analysis***

In this example, we are going to assume that the source of proportion curves is from the indicator kriging run simultaneously in the program, therefore we are going to follow the variogram

requirement specified as Mode 2 in Table 10. That is, we need to build 5 variograms, 3 for facies variogram in its original space and 2 for Gaussian variograms (facies and porosity).

For each variogram analysis, we need to follow the 7 step-by-step procedures described in Section A.2.4. The analysis of facies 1 will be presented in detail in the following paragraphs but for other variograms only the results will be shown.

1. Default Isotropic Variogram. To create a default isotropic variogram for facies 1, user needs to select the Data Variogram menu and update the parameters as shown in Figure 91. Note that the program automatically calculates the lag parameters for the given data file.

2. Isotropic Variogram. Using different lag parameters, we can recalculate the isotropic variogram. User may make several trials before obtaining the best isotropic structure (if any). In Figure 92 the final isotropic variogram is obtained using maximum lag of 2500 ft and number of lags equal to 10.

3. Anisotropic Variogram. After activating the floating menu and selecting the anisotropic variogram menu, we can obtain the default anisotropic variogram as shown in Figure 93. To get variogram at other directions, user needs to select the Change Parameters menu and to modify the azimuth group dialog.

4. Direction of maximum continuity. From Figure 93, we can see that the variogram with azimuth equal to  $0^\circ$  shows the maximum range, therefore we are going to select this variogram as the variogram of maximum continuity. The program will automatically assume that  $90^\circ$  is the minimum continuity direction.

5. Modeling the maximum and minimum continuities variogram. To model the variogram, first we need to activate the modeling mode by selecting the Activate Modeling of the floating menu. Figure 94 shows the final model of the two variograms.

6. Vertical variogram. Since this is the 2D case we can skip this step.

7. Zonal Anisotropy. After selecting the Zonal Anisotropic menu we can get the final variogram model as shown in the dialog as shown in Figure 95. To enter this model into the

simulation parameter dialog, the user needs to press the “Accept Model” button so the variogram name window is highlighted. The name of this variogram should be Facies-Indicator, with facies number equal to 1.

The other 3 variograms created for this example are shown in Figure 96 (b) through (d). These variograms need to be entered into the zonal anisotropy dialog so the simulation parameter can be automatically updated appropriately.

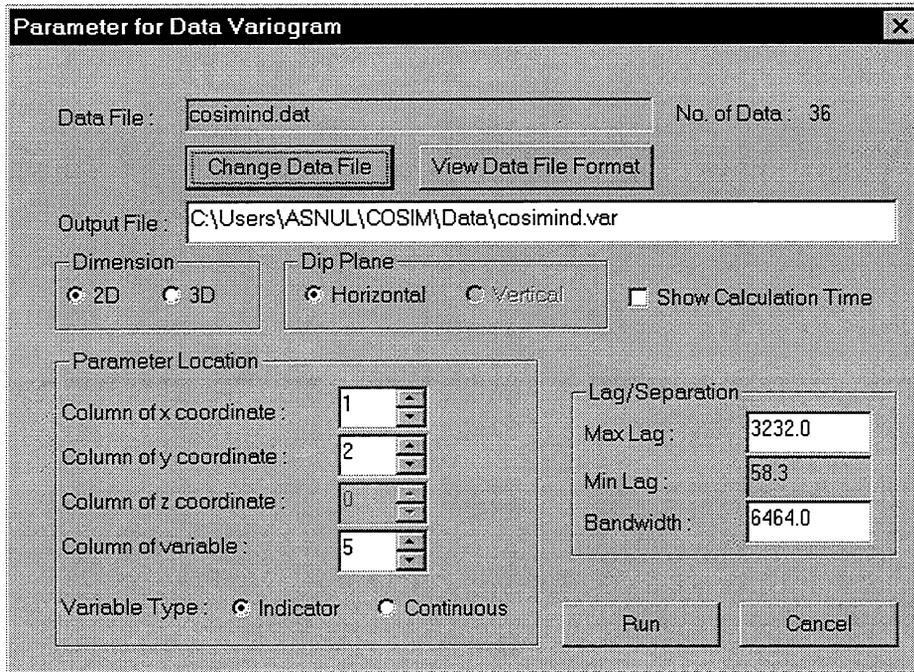


Figure 91: Parameters to create variogram for facies 1

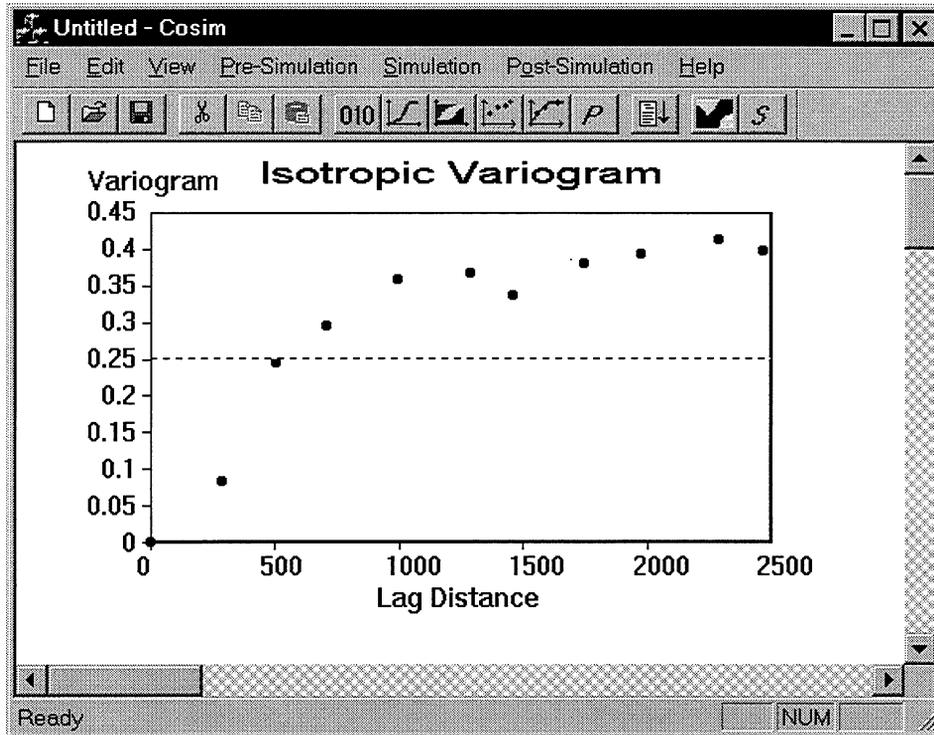


Figure 92: Isotropic variogram for facies 1

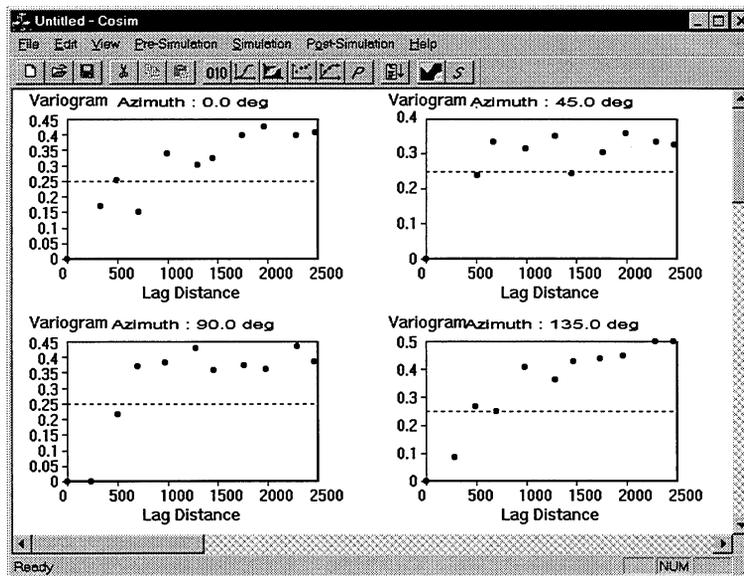


Figure 93: Anisotropic variogram for facies 1 using default azimuth directions

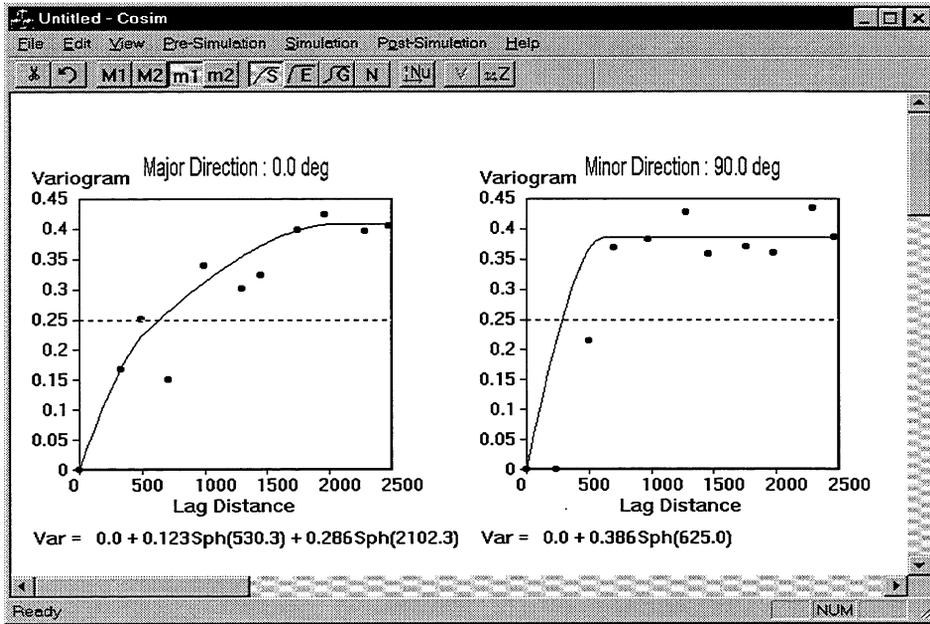


Figure 94: Variogram model for facies 1.

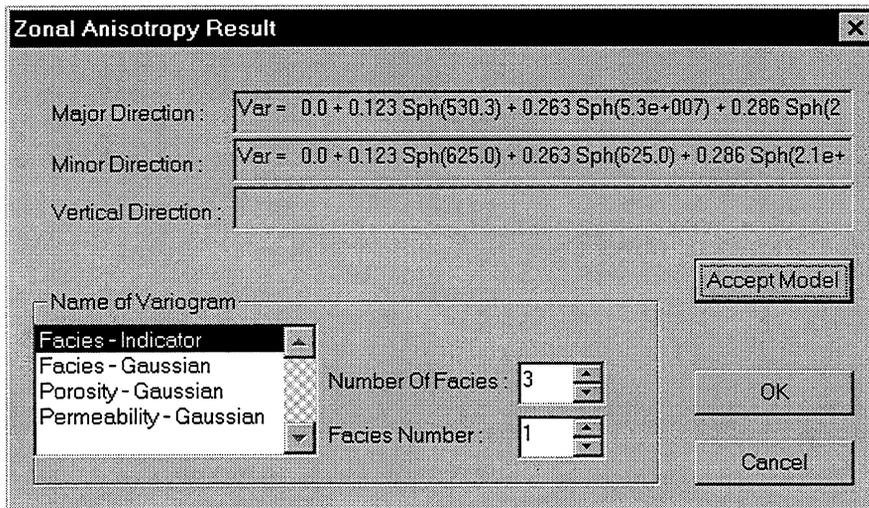
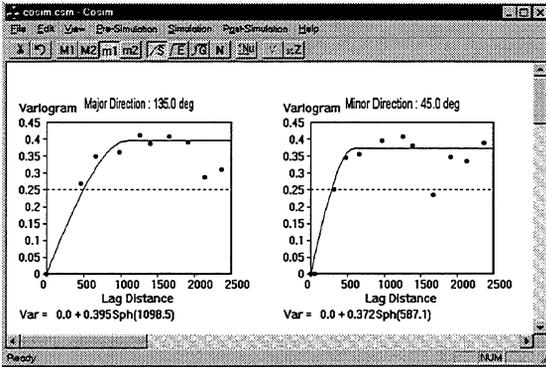
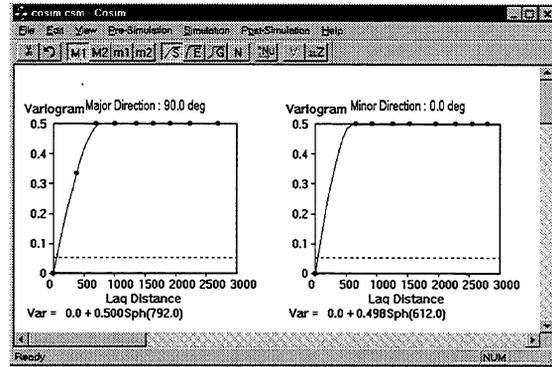


Figure 95: Zonal anisotropy dialog for variogram of facies 1.

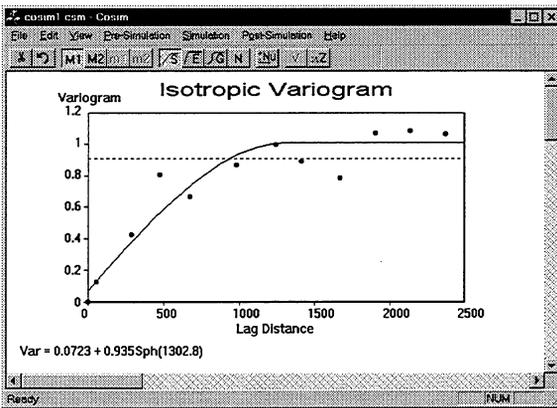


(a)

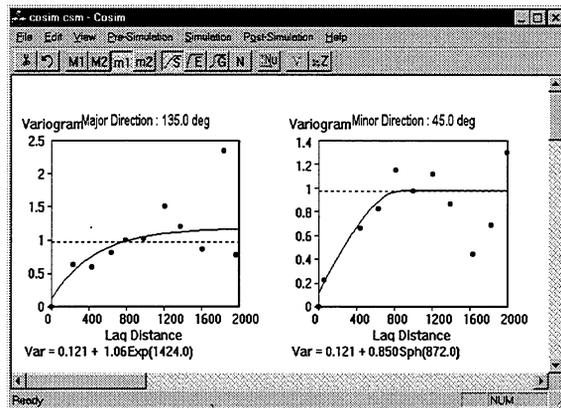


(b)

Figure 96: Variogram model for : (a) facies 2 and (b) facies 3.



(a)



(b)

Figure 97: Gaussian variograms for (a) Facies and (b) Porosity.

### B.5.5 Simulation Parameter

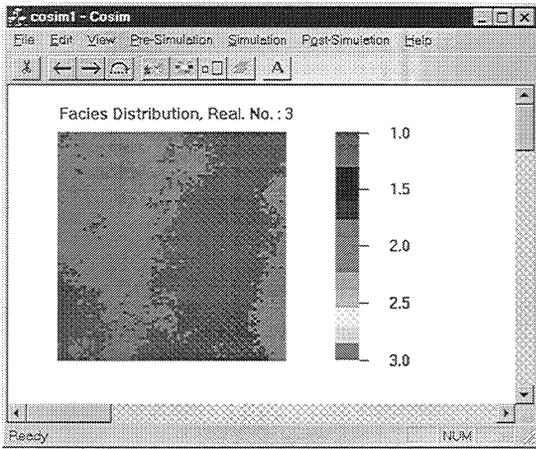
After performing all of the Pre-Simulation menu, some of the required simulation parameters should be automatically correct in the Simulation Parameter dialog. This includes the parameters in the following groups, Data File, Proportion Curves, and Variogram groups. User can verify this by browsing these groups. The other groups that still need modification are Grid Block, Kriging, Output File, Permeability, and Simulation Groups.

The grid block parameters in this example are as follows,  $n_x = n_y = 80$ ,  $n_z = 1$ , grid block size in  $x =$  size in  $y = 66$  ft, in  $z = 1$  ft, and minimum coordinates in  $x =$  in  $y = 33$  ft, in  $z = 0.5$  ft. For the Kriging group, the minimum and maximum points used in kriging are 4 and 12, respectively. The

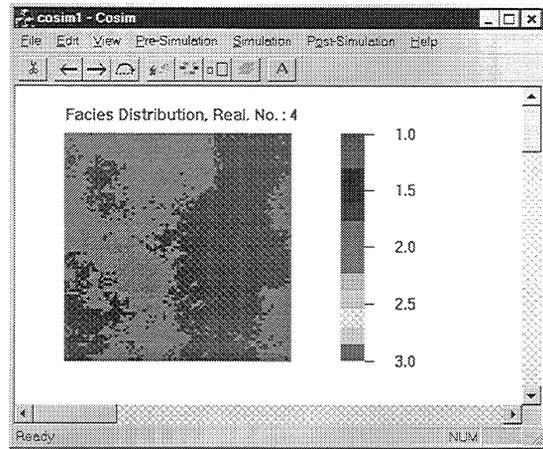
search neighborhood is defined as a circle, i.e., isotropic search neighborhood, with radius of 1500 ft. The Output file name will be given as “cosim.ocm”. The correlation file name for permeability simulation is “cosimfpk.dat” with the column number for facies, porosity, and permeability in this file are 1, 2, and 3, respectively. The only information to be changed from the Simulation group is the number of realizations, from 1 to 20.

Once these parameters are correctly entered in the Simulation Parameter dialog, the simulation is ready to run. It is advised that before the simulation is executed, the document file (\*.csm) should be saved. This is to avoid losing the information, since the program does not allow to cancel the simulation except to use the Windows’ End Tasks mechanism that will stop the program completely without giving the chance to save the document file. The document file in this example is named “cosim.csm”.

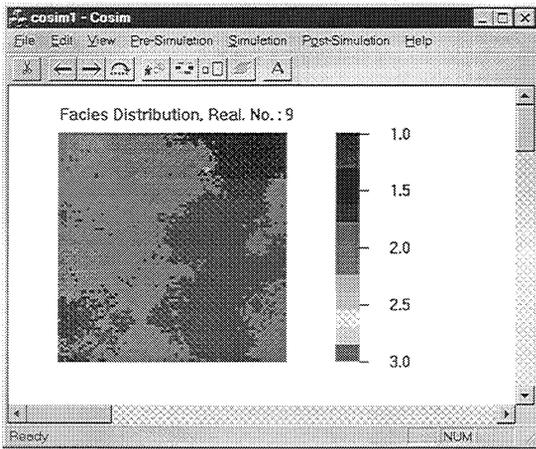
After the simulation is completed, the output simulation result can be viewed using the 2D Cross Section menu. Figure 98 shows the example of 4 realizations of the facies distributions and Figure 99 shows porosity and permeability distributions for one of the realizations.



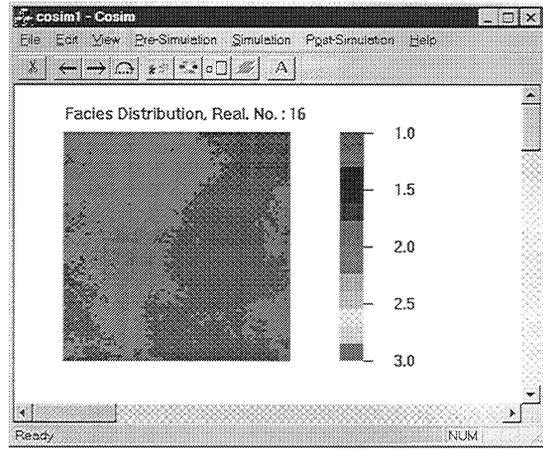
(a)



(b)

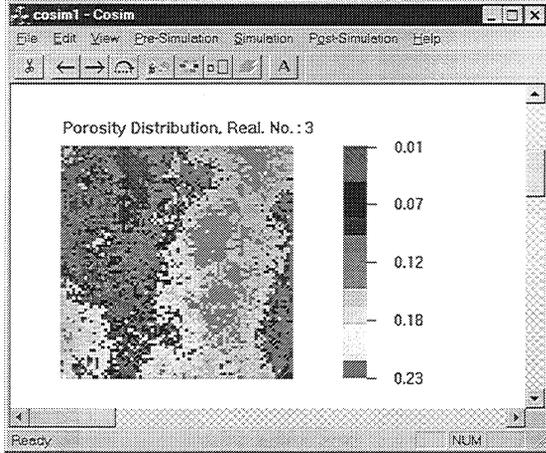


(c)

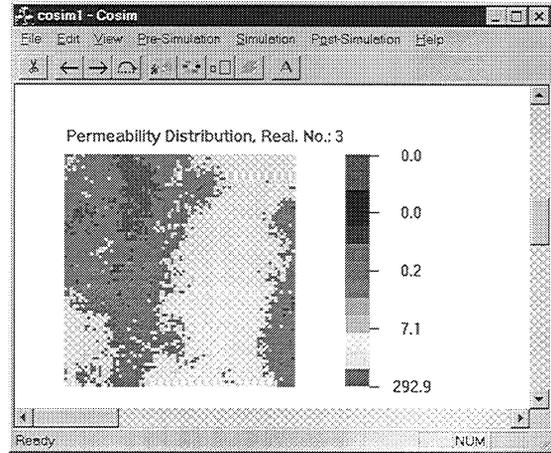


(d)

Figure 98: The 2D cross sections of the simulation result for 4 different realizations



(a)



(b)

Figure 99: The 2D cross section of porosity and permeability distributions for one of the realizations

## **Appendix C Program Manual for Integration of Dynamic Data into Reservoir Description**

### **C.1 Introduction**

Reservoir characterization is an inverse problem, in that the objective is to develop descriptions of the spatial distribution of those reservoir parameters which dominate its flow behavior -- such as permeability and porosity -- from measurements of the parameters which characterize that flow behavior, namely, pressures and rates. Additionally, 'hard' data -- typically punctual observations or measurements of the porosity and permeability at the wellbores -- are used to condition the description process.<sup>1</sup> In the geostatistical approach to reservoir modeling, assumptions about the spatial statistics of the reservoir parameters in which we are interested are used as a means of guiding the resulting description. The spatial statistics are estimated by understanding the geologic processes responsible for the deposition of the reservoir, in conjunction with actual measurements of the reservoir parameters within the field.

This report presents the manual for a stochastic simulation approach to reservoir characterization called simulated annealing (SA).<sup>2</sup> Because of our incomplete knowledge of the earth's subsurface, heuristic approaches -- as typified by stochastic simulation methods -- have grown in popularity. In addition, by presenting us with alternative yet equi-probable reservoir descriptions, these approaches enable quantification of the uncertainty of reservoir performance forecasts. This is a lot more useful for risk and economic evaluation than the traditional deterministic methodology of layer-cake modeling.

### **C.2 Theory**

#### ***Stochastic Simulation vs Estimation***

Estimation of reservoir properties refers to techniques which use linear interpolation to populate a reservoir grid with values. Conditioning data are used together with spatial statistics of the reservoir parameters to develop the estimate of the reservoir model. The geostatistical methodology of kriging is usually used for this.<sup>3,4</sup> It should be noted that a single description, based on a given set of conditioning data and parameter statistics, results. Additionally, the resulting description is substantially 'smoother' than the true reservoir being modeled, since

kriging utilizes the principle of minimum variance, unbiased estimate (MVUE) and so the error variance of the kriged description is less than that of the true reservoir.<sup>4</sup> On the other hand, stochastic (conditional) simulation honors -- in addition to the conditioning data, if available -- the a priori reservoir parameter statistics. Hence the assumed variability of reservoir properties is preserved. In addition, multiple images may be generated, giving a better sense of the uncertainty in the reservoir description. Thus the objective of the exercise is to develop models for flow simulation (for forecasting performance), then stochastic simulation is the methodology of choice.

### ***Stochastic Simulation Methodology Selected***

SA was selected as our methodology of choice for the following reasons:

- It is a non-parametric approach, in that, unlike the popular Gaussian-based approaches such as sequential Gaussian simulation, there is no assumption of any particular distribution function model (such as Gaussian) for the reservoir variable(s).
- This approach is flexible in ability to incorporate multiple constraints. Most other stochastic simulation approaches are more restrictive in the use of constraints.
- The methodology is robust, having a theoretical basis for asymptotic convergence. In addition, it is a well-established procedure, having been successfully applied as an approximation approach in a diverse range of disciplines such as the famous traveling salesman problem and computer chip placement on circuit boards.<sup>5</sup>

### ***Overview of Algorithm***

Simulated annealing is a combinatorial optimization algorithm. For practical applications it is used as an approximation algorithm so that a solution may be obtained in a reasonable timeframe. The goal of the algorithm is the minimization (or maximization) of an objective function. The traditional application of simulated annealing to reservoir characterization utilizes an objective function comprised of a mis-match of a measure of the reservoir's spatial statistics, captured by the variogram. Hence, we minimize the difference between the model variogram and

the experimental variogram. Additionally, there may be conditioning data which are honored and help to stabilize the problem. Thus a one-part objective function is used.<sup>6</sup>

In our modified approach, we use a two-part objective function in which the first part is the same as before, but a second constraint is introduced which involves the mis-match between observed flowing bottomhole pressures (BHPs) and BHPs flow simulated on the currently-developed reservoir description. In effect, automatic history-matching is done.<sup>8</sup> Special techniques are employed which cut down on the flow simulation time of the algorithm. These include:

- The use of representative upscaling for the flow simulation part of the objective function. The upscaling technique used enables approximate BHP matching even during the transient flow period. Thus flow simulation CPU time is reduced by using an upscaled grid.
- The flow simulation is performed in Laplace space using the Laplace Transform Finite Difference (LTFD) approach.<sup>7</sup> This effectively removes the time dependence from the partial differential equation and so reduces the truncation error of an equivalent conventional finite difference scheme. Moreover, the LTFD method yields a stable non-increasing material balance error. It should be noted however that the use of this approach limits the simulation to primary, single-phase oil production.
- Variogram pre-selection: the variogram component of the objective function is determined first. Should this objective function change be acceptable, only then is the flow simulation part of the objective function determined -- and hence the need for unnecessary flow simulation is substantially reduced.

In essence, the algorithm follows the following steps:

- The input data are read-in. These data define the 3-D Cartesian reservoir grid, the model variogram(s), the conditioning data values and locations, the non-parametric cumulative distribution function (*cdf*), the petrophysical properties such as porosity and compressibility, the production and pressure history and the SA parameters.

- An initial permeability distribution is generated. This is usually a random distribution constrained by the input *cdf* and honoring the conditioning data.
- The initial variogram and flow simulation components of the objective function (energies) are calculated. These are termed the initial energies and are used as normalizing factors for the annealing objective function.
- The initial ‘temperature’ is calculated. Analogous to physical annealing, a relatively high temperature, equivalent to a state of randomness or high energy in the distribution, is needed to start the annealing process.
- The temperature reduction loop is now performed. Within this loop, an inner loop over the maximum allowable swaps set for that temperature level is performed. In this loop a gridblock is selected for perturbation -- note that we use single-point perturbations instead of two-point swaps -- and the change in the variogram energy is determined. The transition probability defined by the Metropolis condition is applied:

$$P\{acceptance\} = \begin{cases} 1 & \text{if } \Delta E \leq 0 \\ \exp\left(\frac{-\Delta E}{T}\right) & \text{if } \Delta E > 0 \end{cases} \quad (C.23)$$

where  $\Delta E$  is the change in the objective function (energy) value, and  $T$  is the control parameter or temperature.

Simply stated, this tests whether to accept the perturbation. If the energy is lowered, then the perturbation is accepted with a probability of one. Should the energy be increased, then there is a small non-zero probability that the perturbation will be accepted. This is the heart of a stochastic hill-climbing algorithm, since it allows random uphill moves, ensuring that the solution avoids local minima and converges to the globally-optimal solution (or nearly so). If the variogram energy change is unacceptable, we simply select a new location for perturbation after checking that we have not exceeded the maximum number of perturbations set for this temperature level.

- If, however, the variogram energy change is accepted, we then determine the flow simulation energy change and then calculate the overall change in the objective function. Note that the two

components have to be weighted. We have determined that equal weightings was an acceptable procedure to apply.<sup>8</sup> If the overall change in the objective function is rejected by the Metropolis condition, we simply select a new location to perturb. As previously mentioned, using variogram pre-selection cuts down on the number of flow simulations required.

- If the overall objective function change is accepted, the distribution is updated. That is, the permeability at the location perturbed is updated to the new value. Additionally, the objective function components are also updated. We then test to see if the tolerances set on the two objective components have both been met. If they have (i.e. each component is equal to or less than its set tolerance), or if the number of successful perturbations is equal to the maximum set for this temperature level, then we output the statistics for this temperature level and re-initialize the parameters for the new temperature level (i.e. the inner loop is terminated). If the objective function tolerance is met (or exceeded) then the algorithm terminates. If not, then we step to the next temperature level (once the maximum number of temperature levels or the acceptance ratio tolerance is not yet exceeded).

Figure 1 is a flowchart of the new SA procedure.

### **C.3 Inputs**

#### *SA Inputs*

##### *sanio.dat*

Figure 2 is a sample of this input data file.

- **Grid Configuration:** This defines the 3-D grid origin and its uniform gridblock dimensions. Following the geoEAS convention, the origin is usually the center of the first gridblock. Thus it is defined as one-half of the gridblock dimensions.
- **Conditioning Data:** This section defines the number of conditioning data points (if available) and the file in which they are to be found. Note that this conditioning data file is laid out as a typical geoEAS data file. Figure 3 is a sample conditioning data file. The conditioning data file has six columns of data:

- rock type (an indicator value)
  - porosity (porosity units)
  - permeability (md)
  - x-location (ft.)
  - y-location (ft.)
  - z-location (ft.)
- **Distribution Function:** Here we define the source for the *cdf*; i.e. whether it is input or calculated from the conditioning data. Also defined is whether we assume a continuous *cdf* -- and so interpolate linearly between the class boundaries -- or a discrete one in which case the class mark *cdf* value is used for generating the initial distribution. The file name of the *cdf* is defined and the lower boundary of the first class is defined for the case of a continuous *cdf*. Figure 4 is a sample distribution function input file. This file is not in the geoEAS format (i.e. no file header). Further, the data values represent equi-quantile locations; e.g. if we define 20 quantiles, then the 1st value represents where the 1st one-twentieth of the ranked distribution of permeability values lie, while the last value should be the maximum value expected.
- **Semi-Variogram Models:** This section defines the variogram model(s) to be used. Up to nine directions of anisotropy are catered for. For each search direction, the following parameters are defined:
- the direction cosines in the x-, y- and z-directions
  - the search radius as a fraction of the maximum lag along the specified direction for which the variogram is to be simulated; this value must be between zero and one and is used to calculate the number of lags to be simulated along a specific direction
  - the search increment index to calculate successive lags along each variogram direction; for most simulations this parameter should be equal to one

- the number of variogram structures

Additionally, the nugget is defined and for each variogram structure the following are defined:

- the structure number (not used except to keep track of the number of structures being used)
- the type of variogram structure (1: spherical, 2: exponential, etc., all specified in the file)
- sill value
- variogram range
- if the model is an *fGn* (fractional Gaussian noise) model, then the delta value (smoothing factor) is defined next. Note however that this data field should not be left blank regardless of the model defined.
- Weighting to apply to the variogram(s): Typically a value of 0.5 is used. This means that the flow simulation component is also given a 0.5 weight.
- Simulation Parameters:
  - random number seed: This is an initialization value for the random number generator. Changing the value of the seed will result in another equi-probable realization of the distribution.
  - convergence rate factor: This is the factor to apply in determining the temperature schedule. An exponential decline rate for the temperature is used, such that:

$$T^r = \alpha T^{r-1} \quad (\text{C.24})$$

where  $T$  is the ‘temperature’,  $r$  is the level and  $\alpha$  is the convergence rate factor. Perez had determined via numerical experiments that a value of 0.5 is optimum for a variogram-only objective function<sup>6</sup>

- maximum number of accepted iteration cycles: Usually set at 5, this is the upper limit for the number of accepted perturbations in the inner loop discussed above. For the 1st temperature level, it is also the maximum *total* number of perturbations for that inner loop.

- variogram tolerance: This is the convergence tolerance set for the variogram component of the objective function.
- flow simulation tolerance: This is the convergence tolerance set for the flow simulation component of the objective function.
- acceptance ratio tolerance: This is the convergence tolerance set for the acceptance ratio. This tolerance is used as one option for terminating the algorithm's execution. Sometimes at the lower temperatures the acceptance ratio may fall to a very low value before the solution has annealed to its tolerance level. In such a case, there would be very little benefit to continue the simulation, since very few of the perturbations are being accepted and so the distribution is changing at an extremely slow rate. This tolerance is therefore used as a termination criterion.
- Output Files: Two output files are defined here. One is the distribution -- in a simplified geoEAS format and the other is an output file for the statistics of the simulation. Figures 5 and 6 give examples of the distribution and statistics files respectively. In addition, the output mode is defined to specify whether to overwrite the output files or not.

### ***Conditioning Data File***

The specification of this file is given in the section above.

### ***Distribution Function File***

The specification of this file is given in the section above.

### ***Flow Simulation Inputs***

#### ***data1.dat***

An example of the data1.dat file is given in Figure 7. This file defines the petrophysical and fluid properties: total system compressibility, porosity and oil viscosity. The porosity may be input as a constant value, in which case the *iphi* switch is set to 0, or read-in from a porosity input file, porosity.dat (see Figure 8), in which case *iphi* is set to 1. Also a permeability input switch is defined. If *iperm* is set to 0, then no permeability values are read-in; instead the initial

distribution is created as a random distribution constrained by the *cdf*. If *iper*m is set to 1, then the initial permeability distribution is read-in from file, perm.dat (see Figure 9). data1.dat also defines the initial reservoir pressure (assumed uniform at time zero), the *itzswitch* is used to set the cross-flow option; if 0, cross flow allowed, in which case the z-direction transmissibilities are non-zero, whereas a value of 1 will set all z-direction transmissibilities to zero and only 2-D flow is allowed. Additionally, the directional permeabilities are assumed as related to each other in that the factors *xfac*, *yfac* and *zfac* relate these tensorial values to the gridblock permeability which we are simulating. The remaining inputs in this file are the well locations and the linear solver inputs. Typically, a value of  $10^{-10}$  is set for the iterative solver's convergence tolerance, *tol*. The *itol* switch determines which type of convergence criterion is used. *itmax* sets the maximum number of iterations to undertake before abandoning the iterations and *isym* specifies whether the data storage is symmetric -- in which case only the main diagonal and the upper (or lower) triangular part of the matrix is stored -- or asymmetric -- in which case the entire matrix is stored. This storage type obviously depends on whether the matrix is symmetric or not. For the 3-D case the matrix is not symmetric, primarily because of the partitioning of the rates in the z-direction. For a 2-D system however, this matrix is symmetric.

#### ***data2.dat***

This file contains the parameter settings for the Laplace transform of the pressure and rate data. Typically, these values should remain as set, except for the *tmin* and *tmax* values. As defined in the file header (see Figure 10), these inputs are the wellbore storage coefficient (set at zero), the extrapolation option (set at zero for linear extrapolation using the chord slope of the last pressure interval), the number of log cycles of data used for extrapolation -- if the pressure derivative is used, the number of pressure derivative points for extrapolation (again if the pressure derivative extrapolation option is selected) and a switch for deciding whether to use all the data for extrapolating or whether to detect outliers (standard deviation greater than 2.5).<sup>9</sup>

#### ***data3.dat***

This file (see Figure 11) contains the time-pressure and rate data for each well. Note that we assume constant rates -- which however may be different for the different wells. We have found

that at least 10 pressure values per log cycle are needed for a ‘decent’ numerical Laplace transform -- especially during the transient flow period.<sup>10</sup>

### ***sim3d.inc***

This ‘include file’ defines the 3-D Cartesian grid dimensions. The 2-D upscaling factor in x- and y-directions is defined next. Thus a value of ‘5’ means that the number of fine scale x-direction gridblocks is five times number of coarse scale gridblocks and the same holds for the y-direction grid. Next the *isize* parameter is defined. This equals 19 for a 3-D matrix if an asymmetric matrix storage format is used and 13 if symmetric storage is used. The coarse scale grid dimensions are calculated next from the fine scale values and the upscaling factor. The number of wells is next specified followed by the *nval* value. This value defines the number of matching points to use for each well in the flow simulation component of the SA objective function.

The other required spatial parameters are determined next. Note that there is no need for input for these values -- they are automatically calculated. The next parameter to be calculated is the rank of the coefficient matrix for the flow simulation part of the algorithm. Next the linear solver (SLAP) parameters are set based on the grid specification. Again, no input for these values is required. The *ndiff* value is then calculated based on the value input for the number of wells and the *nval* value. Finally, the input, work and output file unit numbers are defined. A sample of this file is shown in Figure 12.

### ***permea.inc***

This one-line parameter file (shown in Figure 13) is required to define the number of categorical variables being considered; i.e. the number of discrete genetic intervals (DGIs) or facies units being modeled. Also the maximum number of permeability or porosity values per DGI is set.

## C.4 Code

### *Overview*

The code is written in Fortran 77 in double precision. It is contained in six files: `sansp1.f`, `sansp2.f`, `sansp3.f`, `sansp4.f`, `sansp5.f` and `sansp6.f`. The simplest way to compile on a workstation would be as follows:

```
f77 -o ansim sansp1.f sansp2.f sansp3.f sansp4.f sansp5.f sansp6.f
```

where “ansim” is the name of the executable. For a personal computer, there are various compilers which may be used. A brief outline of the function of each file follows.

### *sansp1.f*

This file is the main driver for the routine. It first calls the data input routine then does some initializations and quality control on the input data. The initial distribution is generated and its statistics are calculated and output. The main SA routine is then called after which the results are output and the program terminated.

### *sansp2.f*

This file contains the data input routines, routines to open files and perform initializations and also routines for data output.

### *sansp3.f*

This file contains the core SA routines. The main simulation routine, *simxyz*, is called from the main driver (in `sansp1.f`) and proceeds to calculate the initial energies for the variogram and the flow simulation components. The initial temperature is then calculated and the (nested) perturbation loops, as described above, are performed. The final distribution is created before control is returned to the main driver for final output and program termination.

### *sansp4.f*

This file contains the core flow simulation routines. The routine, *matinv*, is called by *simxyz* (in *sansp3.f*) to perform the flow simulation at each perturbation of the simulation grid. The flow simulation component of the objective function is also determined and the value passed back to the *simxyz* routine.

### *sansp5.f*

This file contains the linear solver code. The SLAP (Sparse Linear Algebra Package) code, a public domain package, is used.

### *sansp6.f*

This file contains the Laplace transform code. A numerical Laplace transform methodology is used.<sup>9</sup> The file also contains the scaling routines, as well as the code to convert from the block permeabilities to the permeability tensor values. In addition, there is a routine to determine the DGI from the location selected for perturbation, then the *cdf* quartile to use based on the porosity value at that location. Each DGI has its *cdf* decomposed into four quartiles and the perturbation is performed within a quartile.

## **References**

- C.1. Aarts, E. and Korst, J.: *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*, John Wiley & Sons Ltd., Chichester (1989).
- C.2. Tarantola, Albert: *Inverse Problem Theory Methods for Data Fitting and Model Parameter Estimation*, Elsevier Science Publishers, Amsterdam (1987).
- C.3. Deutsch, C.V. and Journel, A.G.: *GSLIB Geostatistical Software Library and User's Guide*, Oxford University Press, New York (1992).

- C.4. Kelkar, B.G.: *Application of Statistics to Reservoir Characterization*, self-published, Tulsa, OK (1989).
- C.5. Kirkpatrick, S., Gelatt, Jr., C.D., Vecchi, M.P.: "Optimization by Simulated Annealing", *Science* (May 13 1983) 671-680.
- C.6. Perez, G.: "Stochastic Conditional Simulation for Description of Reservoir Properties", PhD dissertation, University of Tulsa, Tulsa, OK (1992).
- C.7. Moridis, G.J., McVay, D.A., Reddell, D.L. and Blasingame, T.A.: "The Laplace Transform Finite Difference (LTFD) Numerical Method for Simulation of Compressible Fluid Flow in Reservoirs", paper SPE 22888 presented at the SPE Annual Technical Conference and Exhibition, Dallas, TX, Oct. 5-8, 1991.
- C.8. Gajraj, A.: "The Incorporation of Dynamic Constraints in Stochastic Conditional Simulation", PhD dissertation, University of Tulsa, Tulsa, OK (1996).
- C.9. Carvalho, Renato de Souza: "Nonlinear Regression: Application to Well Test Analysis", PhD dissertation, University of Tulsa, Tulsa, OK (1993).
- C.10. Thompson, L.G.: Personal Communication (1995).

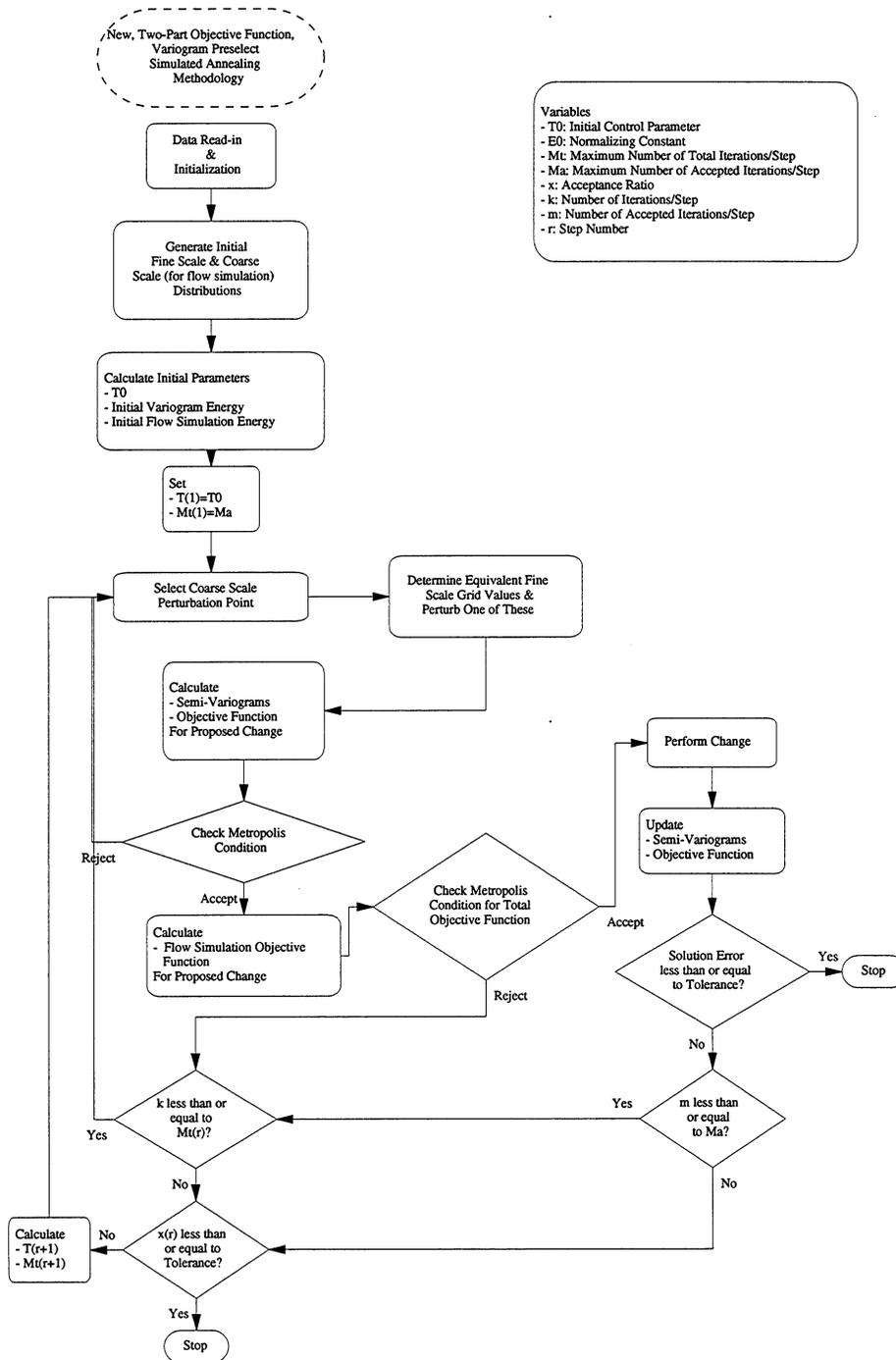


Figure 100: Modified SA algorithm for reservoir characterization



6  
7  
8  
9

```

Nugget                                ->0.d0
|
Type                                  |
= 1 Spherical                          |
= 2 Exponential                         |
= 3 Gaussian                            |
= 4 fBm ()                              |
= 5 fGn ()                              |
|
#      Type      Sill      Range      Delta      |
      (Scalng)   (H)       (fGn)      |
---      ----      -+-----      -+-----      +
1        1        800.d0   983.6d0   1.d0
2        1        800.d0  1379.6d0  1.d0
3        1        800.d0   983.6d0   1.d0
4        1        800.d0  1302.9d0  1.d0
5
6
7
8
9
|
Weighting to apply to variogram(s)     ->0.5d0
|
Simulation Parameters -----+
(old seed value 7868704) Seed           ->1079602
(alpha) Convergence rate factor         ->0.5d0
(Ma) Maximum number of accepted iterations cycles->5.d0
(ev) Variogram tolerance                 ->0.01d0
(tldp)Flow simulation tolerance          ->0.05d0
(ea) Acceptance ratio tolerance          ->0.025d0
|
Output Files -----+
File name for summary of simulation      ->1500s.out
simulated variable                       ->1500v.out
Output mode                              ->2
= Mode 1, Do not overwrite files         |
=      2, Overwrite files                 |
|
+-----+

```

Figure 2: Sample sanio.dat input data file

Conditioning Permeability - Data

6

Rock Type

Porosity (p.u)

Permeability (md)

x-loc

y-loc

z-loc

1	0.1210	69.25d0	500.d0	1500.d0	20.d0
1	0.1130	65.81d0	500.d0	1500.d0	60.d0
1	0.1150	65.11d0	500.d0	1500.d0	100.d0
1	0.2600	82.17d0	500.d0	1500.d0	140.d0
1	0.0880	48.58d0	500.d0	1500.d0	180.d0
2	0.0910	48.58d0	2500.d0	500.d0	20.d0
2	0.1050	67.70d0	2500.d0	500.d0	60.d0
2	0.2180	86.04d0	2500.d0	500.d0	100.d0
2	0.0750	48.58d0	2500.d0	500.d0	140.d0
2	0.1050	74.77d0	2500.d0	500.d0	180.d0
3	0.0980	65.81d0	1500.d0	1500.d0	20.d0
3	0.0840	48.58d0	1500.d0	1500.d0	60.d0
3	0.1320	67.70d0	1500.d0	1500.d0	100.d0
3	0.0990	48.58d0	1500.d0	1500.d0	140.d0
3	0.1130	74.77d0	1500.d0	1500.d0	180.d0

Figure 3: Sample Conditioning Data File

7.237819  
9.09045  
10.863065  
12.250621  
12.866299  
14.671041  
17.232455  
20.145452  
23.67143  
25.52434  
31.087451  
34.514231  
40.26343  
48.58  
49.576224

55.187266  
65.81  
69.25  
74.77  
93.543966

Figure 4: Sample Distribution Function Input Data File

TLEVEL: 29 # OF SWAPS: 12000

1  
 perm  
 0.231600  
 0.277669  
 0.215000  
 0.279056  
 0.235158  
 .....  
 144.622000  
 185.512000  
 0.297788  
 0.237353  
 0.173856  
 0.223275  
 0.181880  
 0.211615

Figure 5: Sample Output Distribution File (truncated)

.....  
 Initial Pattern Variography  
 .....

Search Direction #: 1  
 x-index...= 0 -> x-angle...= 90.00 deg.  
 y- ...= 1 -> y- ...= 0.00 deg.  
 z- ...= 0 -> z- ...= 90.00 deg.

Error Function = (Sample/Model - 1)\*\*2

#	Lag		Semi-variogram		# of	Error
	Relative	Actual	Model	Sample	Pairs	Function
...	.....	.....	.....	.....	.....	.....
1	1	200.0	0.24064D+03	0.24145D+03	1425	0.11291D-04
2	2	400.0	0.46110D+03	0.45238D+03	1350	0.35779D-03
3	3	600.0	0.64121D+03	0.63730D+03	1275	0.37132D-04
4	4	800.0	0.76079D+03	0.75591D+03	1200	0.41113D-04
5	5	1000.0	0.80000D+03	0.79771D+03	1125	0.81728D-05
6	6	1200.0	0.80000D+03	0.80540D+03	1050	0.45573D-04
7	7	1400.0	0.80000D+03	0.80111D+03	975	0.19395D-05
8	8	1600.0	0.80000D+03	0.80452D+03	900	0.31948D-04
9	9	1800.0	0.80000D+03	0.79019D+03	825	0.15030D-03

Search Direction #: 2

x-index...= 1 -> x-angle ...= 45.00 deg.  
 y- ...= 1 -> y- ...= 45.00 deg.  
 z- ...= 0 -> z- ...= 90.00 deg.

Error Function = (Sample/Model - 1)\*\*2

#	Lag		Semi-variogram		# of	Error
	Relative	Actual	Model	Sample	Pairs	Function
10	1	282.8	0.24257D+03	0.24746D+03	1330	0.40506D-03

Search Direction #: 3

x-index...= 1 -> x-angle ...= 0.00 deg.  
 y- ...= 0 -> y- ...= 90.00 deg.  
 z- ...= 0 -> z- ...= 90.00 deg.

Error Function = (Sample/Model - 1)\*\*2

#	Lag		Semi-variogram		# of	Error
	Relative	Actual	Model	Sample	Pairs	Function
11	1	200.0	0.24064D+03	0.24145D+03	1400	0.11413D-04
12	2	400.0	0.46110D+03	0.45584D+03	1300	0.12995D-03
13	3	600.0	0.64121D+03	0.64245D+03	1200	0.37282D-05
14	4	800.0	0.76079D+03	0.76287D+03	1100	0.74788D-05
15	5	1000.0	0.80000D+03	0.80006D+03	1000	0.55802D-08
16	6	1200.0	0.80000D+03	0.80137D+03	900	0.29130D-05

E1FC= 9.9132091973375D-03

E2FC= 0.11556740839121

.....  
 Simulation Statistics  
 .....

Tem	Total #	Acctpc	St	Energy	Change	Mean	Std. Dev.
Lev	Temp	of Cycls	Ratio	#			
0	0.112D+02	0.000	1.000000	0	0.1000D+01	0.0000D+00	0.1000D+01 0.0000D+00
1	0.112D+02	5.000	0.983333	1	0.1480D+02	0.1380D+02	0.7480D+01 0.3873D+01
2	0.558D+01	10.083	0.983607	1	0.1981D+02	0.5009D+01	0.1774D+02 0.1444D+01
29	0.416D-07	717.983	0.015500	17	0.2031D+02	-0.1988D-02	0.2031D+02 0.4310D-03

\* Acceptance Ratio = Fraction of accepted iterations

of total iterations per step (level)  
 \* St # = Number of steps reaching iterations limit

Initial normalized energy = 0.10000D+01  
 Final = 0.20312D+02  
 Initial r.m.s. error = 0.99132D-02  
 Final = 0.20136D+00  
 Total number of swaps = 43079  
     accepted = 8261  
     rejected = 34818  
 Number of temperature levels = 29  
 Initial temperature = 0.11159D+02  
 Final = 0.41572D-07

.....  
 Final Pattern Variography  
 .....

Search Direction #: 1  
 x-index...= 0 -> x-angle ...= 90.00 deg.  
 y- ...= 1 -> y- ...= 0.00 deg.  
 z- ...= 0 -> z- ...= 90.00 deg.

Error Function = (Sample/Model - 1)\*\*2

#	Lag		Semi-variogram		# of	Error
	Relative	Actual	Model	Sample	Pairs	Function
...	.....	.....	.....	.....	.....	.....
1	1	200.0	0.24064D+03	0.29453D+03	1425	0.50157D-01
2	2	400.0	0.46110D+03	0.37215D+03	1350	0.37216D-01
3	3	600.0	0.64121D+03	0.41336D+03	1275	0.12627D+00
4	4	800.0	0.76079D+03	0.44615D+03	1200	0.17104D+00
5	5	1000.0	0.80000D+03	0.46602D+03	1125	0.17428D+00
6	6	1200.0	0.80000D+03	0.49618D+03	1050	0.14423D+00
7	7	1400.0	0.80000D+03	0.50347D+03	975	0.13739D+00
8	8	1600.0	0.80000D+03	0.44446D+03	900	0.19752D+00
9	9	1800.0	0.80000D+03	0.45523D+03	825	0.18573D+00

.....  
 Simulated Annealing Terminated Normally

Figure 6: Sample Output Distribution Statistics File (truncated)

```

c file name: data1.dat
c
c ct          = total compressibility, psi^-1
c iperm       = flag for input permeability
c             0: permeability values not read in
c             1: array of values to be read-in from file, perm.dat
c iphi       = flag for input porosity
c             0: constant value to be read-in after iphi value
c             1: array of values to be read-in from file, porosity.dat
c isym       = switch for type of data storage in coefficient matrix
c             0:non-symmetric
c             1:symmetric
c itmax      = maximum number of iterations for SLAP
c itol       = flag to determine type of convergence criterion
c             itol=2 is strongly recommended for 'bcgs'
c itzswtch   = flag for no cross-flow case
c             0: cross-flow allowed
c             1: no cross-flow
c mu         = oil viscosity, cp
c p_in       = initial reservoir pressure, psi
c phic       = constant value porosity
c rw         = wellbore radius, ft
c tol        = SLAP-required convergence tolerance
c wellx      = x-distance of well location, ft
c welly      = y-distance of well location, ft
c xfac       = multiplicative factor for perm -> kx
c yfac       = multiplicative factor for perm -> ky
c zfac       = multiplicative factor for perm -> kz
c

```

```

c=====

```

```

c      data values

```

```

c=====

```

```

c
c      ct          mu          rw
c      1.5d-6      5.d0       0.25d0

```

```

c
c      p_in        itzswtch
c      5000.d0     1

```

```

c
c      xfac        yfac        zfac
c      1.d0        1.d0        1.0

```

```

c
c

```

```

c The following parameters are required by the SLAP routines...

```

```

c-----

```

```

c      tol      itol
      1.d-10    2
c
c      itmax    isym
      500      0
c-----
c
c      wellx()  welly()
      500.d0    1500.d0
      2500.d0   500.d0
      1500.d0   1500.d0
c
c      iphi     iperm
      1         1
c
c      phic
      0.3
c=====

```

Figure 7: Sample Input Data File, data1.dat

Combination of porperm.dgi and porperm.por

2

DGI

porosity

```

3 .2438
2 .2419
2 .2174
2 .2757
2 .2206
2 .2077
1 .2277
1 .1959
2 .1847
1 .1227
2 .1971
2 .2532
2 .2707
3 .2310
3 .2305
2 .2676
2 .2663
2 .2144

```

```
2 .2193
2 .2405
1 .2328
2 .1819
1 .1808
1 .2280
2 .2372
2 .2410
.....
2 .2055
2 .1824
1 .2318
2 .2402
1 .2429
1 .2470
1 .2490
1 .2139
3 .2205
2 .2758
3 .2670
3 .2804
3 .2319
1 .2415
2 .1885
1 .2220
2 .2000
2 .2307
```

Figure 8: Sample Porosity Data Input File (Truncated), porosity.dat

perm.dat input permeability data file

```
1
perm
93.880000
33.148000
23.264000
54.056000
24.372000
20.195000
 9.833000
 7.149000
14.478000
 3.428000
```



```

% IEXTR- estrapolation option:
%           = 0 -> linear extrapolation in pressure using the chord slope of the
%               last pressure interval
%           = 1 -> linear extrapolation of the logarithmic pressure derivative
%               on a cartesian plot
% XLD - span of the data (in terms of log cycle) used to compute the best
%       straight line for extrapolating the pressure derivative (usually
%       XLD is between 0.1 and 0.3). It is used only for the case IEXTR=1.
%       If XLD is negative, the program will ask the user to input the
%       equation of the straight line  $tdDP/dt = A t + B$ .
% NDD - number of pressure derivative data points (from the last one)
%       that should not be considered for extrapolation purpose (IEXTR=1)
% IOUT - option for deleting derivative outliers (IEXTR=1)
%           = 0 -> use all derivative data within XLD
%           = 1 -> delete derivative data which std. deviation is greater than 2.5
% TMIN - minimum input time value (4.167d-2=>1 hr)
% TMAX- maximum input time value
%
% EST      IEXTR      XLD      NDD      IOUT
% 0.d0     0          0.175d0  1        0
%
% TMIN     TMAX
% 1.d0     100.d0

```

Figure 10: Sample Input Data File, data2.dat

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% INPUT FILE OF PROGRAM LTRANSF.FTN
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% University of Tulsa
% Renato Carvalho 03-15-93
%
% modified by Allyson Gajraj 12-04-93:
%
% The objective of LTRANSF.FTN is:
% Numerically transform tabulated data to Laplace space by using the
% piecewise linear method.
%
% Input data:
% -----
% NTP - number of pressure values
% NTQ - number of rate values
% P - pressure (psi)

```



```

%
%      T          P
.00000E+00 .00000E+00
.10000E-01 .86670E+01
.10593E-01 .86920E+01
.11220E-01 .87180E+01
.....
.74993E+02 .67012E+02
.79433E+02 .69252E+02
.84143E+02 .71604E+02
.89133E+02 .74074E+02
.94413E+02 .76667E+02
.10000E+03 .79390E+02
%
%      Rate Data
%      -----
%      NTQ
%      1
%
%      T          RATE
%      1.d-5      30.d0
%=====
.....

```

Figure 11: Sample Input Data File, data3.dat

```

c Note: This version is for use with SLAP gmres method
c
c file name = sim3d.inc
c
c this file contains the parameter statements for the program, sansp*.f*
c
c-----
c ifilein1      = static data input file, data1.dat
c ifilein2      = static Laplace transform data file, data2.dat
c ifilein3      = time-pressure-rate data file, data3.dat
c ifilewk1      = static data work file, temp1 (data from data1.dat)
c ifilewk2      = work file, temp2 (data from data2.dat)
c ifilewk3      = dynamic data work file, temp3 (data from data3.dat)
c ifileout1     = final results output file, results.out
c ifileout2     = intermediate results output file, slap.out
c ifileout3     = Laplace results output file, ltf.out
c isize        = dimension for the dummy vectors for changing the coefficient

```

```

c          matrix values in coefchange & coefmod
c leniw   = length of the 'iwork' array
c lenw    = length of the 'rwork' array
c ndiff   = number of summation terms for the dynamic constraint
c         part of the simulated annealing objective function
c nel     = number of non-zeros in the lower triangle of the matrix
c         (including the main diagonal)
c nozero  = maximum number of non-zeros expected
c ntrans  = number of transmissibility value per well (used for
c         upscaled grid runs)
c nu      = number of non-zeros in the upper triangle of the matrix
c         (including the main diagonal)
c nval    = number of time-pressure-rate data points for each well
c nwell   = number of wells
c nx      = number of grid blocks in the x-direction
c nxx     = dimension size for the x-direction spatial variable
c nxyz    = total number of internal grid blocks
c ny      = number of grid blocks in the y-direction
c nyy     = dimension size for the y-direction spatial variable
c nz      = number of grid blocks in the z-direction
c nzz     = dimension size for the z-direction spatial variable
c=====
c USER-DEFINED PARAMETERS (CHANGE THESE ONLY)...
c
c     parameter (nx=15,ny=20,nz=5)
c     parameter (igrfac=5, isize=19)
c     parameter (nx1=nx/igrfac,ny1=ny/igrfac)
c     parameter (nwell=3,nval=10)
c
c=====
c Spatial parameters...
c
c     parameter (nxyz =nx*ny*nz)
c     parameter (nxx =nx+2,nyy=ny+2,nzz=nz+2)
c     parameter (nxyz1=nx1*ny1*nz)
c     parameter (nxx1 =nx1+2,nyy1=ny1+2)
c-----
c Additional values in matrix for rate discretization...
c     parameter (msize=nxyz1+nwell*(nz+1))
c     *****1*****2*****3*****4*****5*****6*****7**
c SLAP-required parameters...
c
c     'nozero, nel, nu' specifications if x-flow allowed:
c     parameter (nozero = 7*nxyz1-2*(nx1*ny1+nx1*nz+ny1*nz)+5*nwell*nz)
c     parameter (nel   = 4*nxyz1- (nx1*ny1+nx1*nz+ny1*nz)+3*nwell*nz)

```

```

c      parameter (nu    = 4*nxyz1- (nx1*ny1+nx1*nz+ny1*nz)+3*nwell*nz)
c=====
c      `nozero, nel, nu' specifications if no x-flow allowed:
      parameter (nozero = 7*nxyz1-2*(nx1*ny1*nz+nx1*nz+ny1*nz)+
1          5*nwell*nz)
      parameter (nel    = 4*nxyz1- (nx1*ny1*nz+nx1*nz+ny1*nz)+
1          3*nwell*nz)
      parameter (nu    = 4*nxyz1- (nx1*ny1*nz+nx1*nz+ny1*nz)+
1          3*nwell*nz)
c=====
      parameter (lenw  = nel+nu+8*msize)
      parameter (leniw = nel+nu+4*msize+12)
c-----
c Flow simulation-required iteration parameter...
c
      parameter (ndiff=nwell*nval)
c-----
c Parameters for files...
c
      parameter (ifilein1 =101,ifilein2 =102,ifilein3 =103,ifilein4=110,
+      ifilein5 =111)
      parameter (ifilewk1 =104,ifilewk2 =105,ifilewk3 =106)
      parameter (ifileout1=107,ifileout2=108,ifileout3=109)
c-----

```

Figure 12: Sample Include File, sim3d.inc

```

*****1*****2*****3*****4*****5*****6*****7**
*** permea.inc ***
*
* maxdatk=maximum number of porosity/permeability values per dgi
* ncat =number of categorical variables (dgi's)
*
      parameter (ncat=3,maxdatk=1000)
*
*****1*****2*****3*****4*****5*****6*****7**

```

Figure 13: Sample Include File, permea.inc

