DOE/BC/14894--12 R 5/10/96

APPLICATION OF ARTIFICIAL INTELLIGENCE TO RESERVOIR CHARACTERIZATION: AN INTERDISCIPLINARY APPROACH

(DOE Contract No. DE-AC22-93BC14894)

Submitted by

The University of Tulsa
Tulsa, OK 74104

Contract Date:	October 1, 1993
Anticipated Completion Date:	September 30, 1996
Government Award:	$240,540
Program Manager:	B.G. Kelkar R.F. Gamble
Principal Investigators:	D.R. Kerr L.G. Thompson S. Shenoi
Reporting Period:	January 1 - March 31, 1996

Contracting Officer's Representative

Mr. Robert E. Lemmon
Pittsburgh Energy Technology Center
P.O. Box 10940, M/S 141-L
Pittsburgh, PA 15236-0940

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, express 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 or any agency thereof.

Objectives

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 amount 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 are the verification and validation of the expert system, since much of the interpretation of the experts is based on extended experience in reservoir characterization.

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 will be technology transfer. With 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 will 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.

Summary of Technical Progress

1. 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 will facilitate the implementation of the system and its validation and verification. The three component systems will be representative of how each of the experts in geology, geostatistics, and engineering characterizes the reservoir. Figure 1 describes a model for this breakdown. The concurrent development of these component systems fits into the development of the large and small scale aspects of the system as originally stated in the proposal.

The geostatistical system continues to be tested and updated. This sytem includes the use of wavelet transforms to determine the effect of compression to some part of the original data on the overall performance of the reservoir. Concentration on the geology system has been placed on upgrading the neural network output for log facies recognition. In addition, we have developed an automated system for correlation of zones among wells. The marker bed recognition system is considered complete at this time, though later enhancements may be added. The individual components (completion rules, type curve matching, and linear regression components) are currently being integrated to form a complete well test interpretation system. The graphical system is currently being designed for implementation to visualize correlations between wells. This system will be augmented as the other system components mature. The designing of the overall user interface to integrate all of the systems will begin in the following quarter.

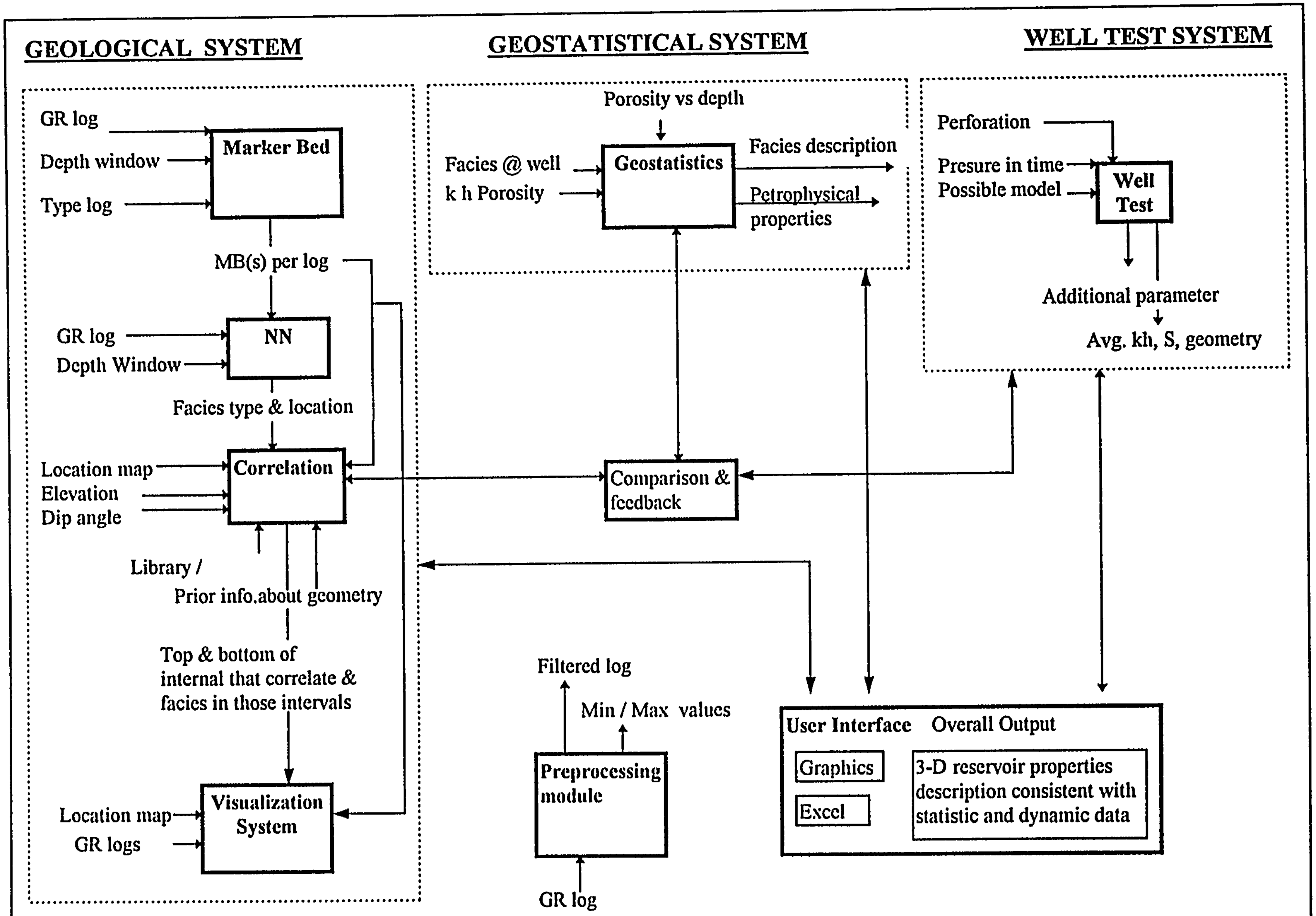


Figure 1: Overall System Components

2. Geostatistical System: Incorporation of Dynamic Constraints in a Reservoir Description Process

Three synthetic 100x100-gridblock datasets were generated: two using the *Ingen/makecdf/SA* suite of algorithms (described in the report for the last quarter for 1994) and one using the sequential Gaussian simulation (*sGs*) algorithm from *GSLIB*.[1] In this report, we present the results obtained using these datasets for evaluating the performance of the proposed composite objective function SA algorithm. The approach used in evaluating the results is to compare the 'truth case' description to that obtained by our algorithm and also to that obtained when a variogram-only objective function is used. Note that in testing our modified approach, we use both upscaling approaches (modified geometric averaging and 'Ding'[2]) in the runs for the first two synthetic images. Note also that the *same* variogram models are used for the composite objective function SA run (our modified SA algorithm) and for the variogram-only objective function SA run. Hence the only difference between these SA algorithms is that the flow simulation constraint is included in our modified approach. The comparisons made are:

- Visual comparisons of the images.
- Comparison of the dynamic behavior. This is accomplished by flow simulating all descriptions using *ECLIPSE-100*[3] and calculating the relative (percentage) errors in the flowing *BHP*s using the truth case flow simulation results as the standard or basis of comparison.

The purpose, of course, is to validate that

- the realizations generated by the modified SA algorithm more closely 'resemble' the truth case image than that generated by the variogram-only objective function, and that
- the dynamic behavior of the distribution obtained from the modified approach also matches that of the truth case more closely than that of the variogram-only objective function SA run.

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.

2.1 Visual Comparisons for Dataset #1

Figure 1 presents a visual comparison of the results. It shows 4 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, the third, (c), is the modified SA run in which Ding's upscaling approach is used and the fourth, (d), is the result for the variogram-only objective function SA run. It can be seen that the major features of the truth case description are captured by both Figures 1 (b) and (c), while Figure 1(d) does a relatively poor job.

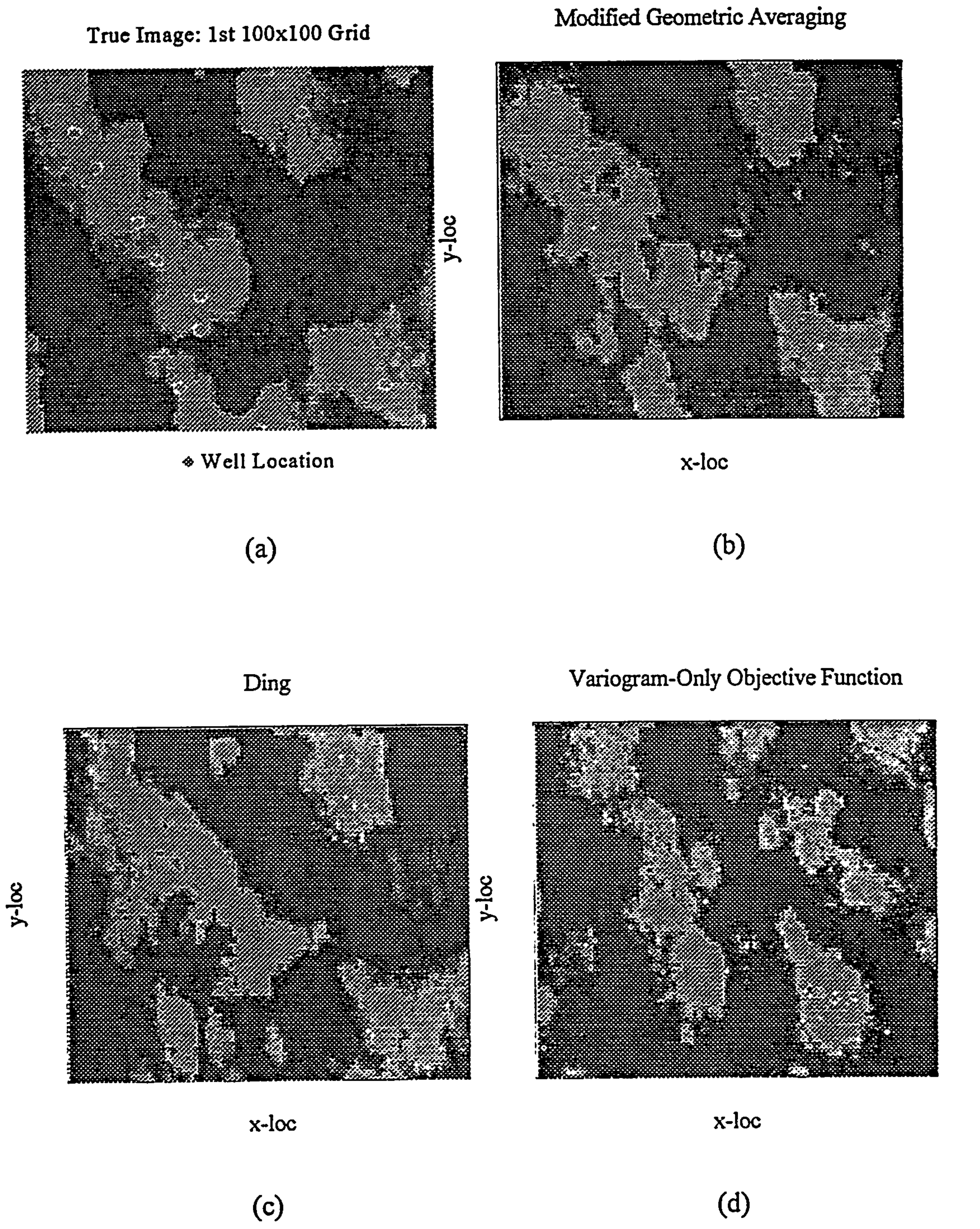


Figure 1: (a) Truth Case (b) Modified SA Run Results - 1
(c) Modified SA Run Results - 2 (d) Variogram-Only SA Run Results for Dataset #1

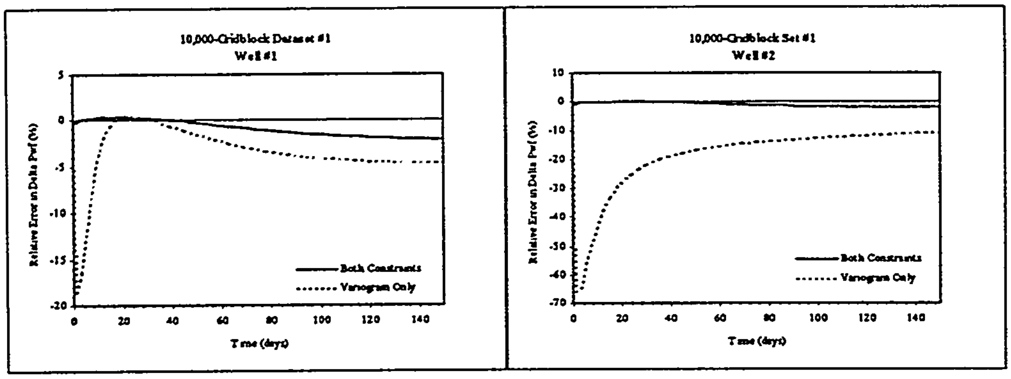
2.2 Flow Performance Comparisons for Dataset #1

2.2.1 Composite Objective Function vs Variogram-Only Objective Function

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 *BHP*s compared. Figures 2(a)-(b) compare the errors in these pressures (actually in the *change* in the flowing *BHP*s) as a function of time for the variogram-only objective function SA run and that of the composite objective function SA run. In these plots we have used the geometric averaging approach. The relative errors are calculated as given in Equation (1) below:

$$\text{Relative Error} = 100 \cdot \frac{\text{'Truth Case'}\ \Delta p_{wf} - \text{SA Run Result}\ \Delta p_{wf}}{\text{'Truth Case'}\ \Delta p_{wf}}. \quad (1)$$

Other wells show a similar behavior. These plots show that the errors are significantly smaller when the modified approach is used, even in the predictive part of the time scale (>100 days). These plots further illustrate that the upscaling approaches used are adequate.

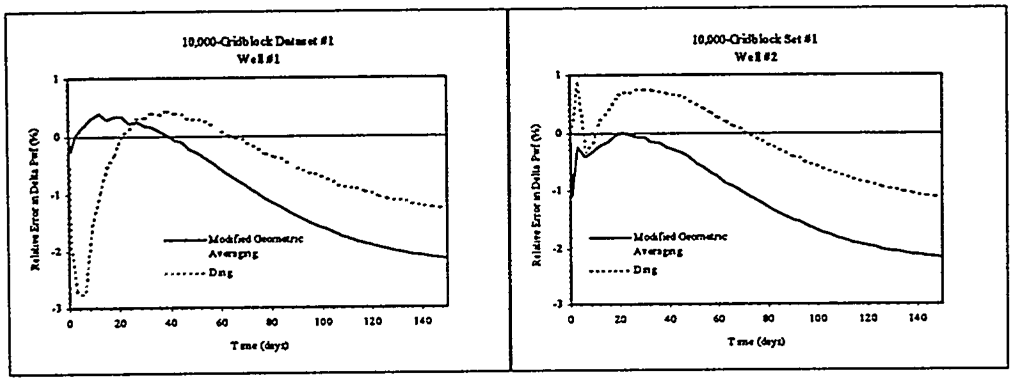


(a) (b)

Figure 2: Pressure Error Comparisons Between the Modified SA Approach and the Variogram-Only SA Approach for Dataset #1

2.2.2 Comparison of Upscaling Approaches

Figures 3(a)-(b) give the results of the analyses conducted to ascertain whether one upscaling approach was better than the other. It can be seen from these plots that the methods performed about the same, thus validating the use of either. Figure 4 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. Figure 4 also shows that when the maximum absolute relative errors are compared, the two approaches appear to be on par, and in addition, these results are at least an order of magnitude better than those of the variogram-only results.



(a)

(b)

Figure 3: Pressure Error Comparisons Between the Modified Geometric Averaging and 'Ding' Upscaling Approaches for Dataset #1

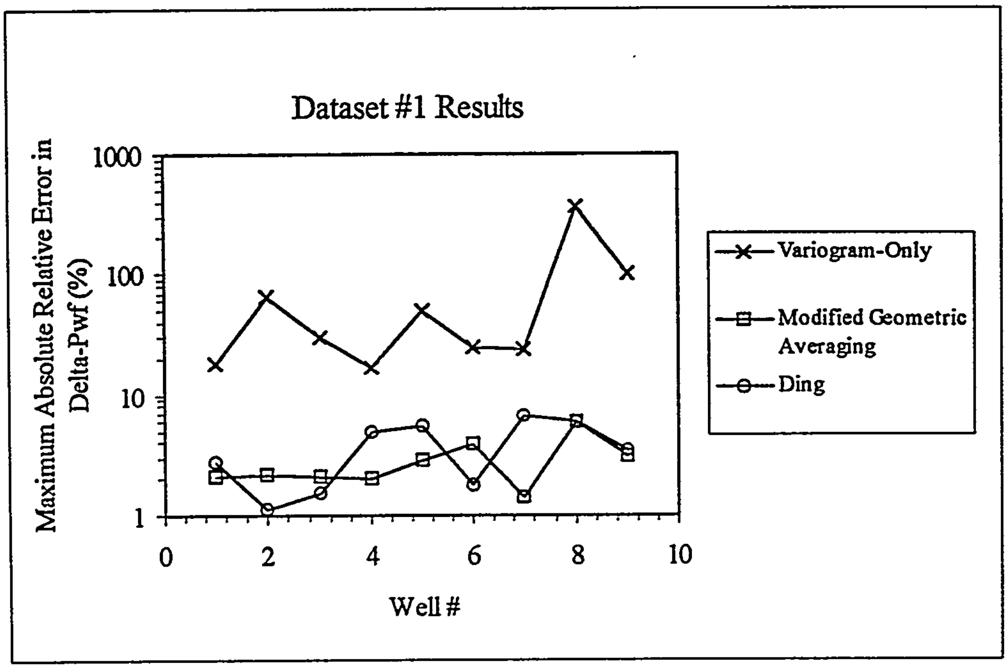


Figure 4: Comparisons of Maximum Absolute Relative Errors from Variogram-Only Objective Function and the Composite Objective Function Using the Two Upscaling Approaches For Dataset #1

2.3 Visual Comparisons for Dataset #2

Figure 5 below shows the true image for dataset #2 -- also generated using the *lngen/makecdf/SA* code -- in (a), the SA composite objective function run results using modified geometric averaging upscaling and Ding upscaling in (b) and (c) respectively, and the SA variogram-only objective function run result in (d). As for dataset #1, it can be seen that the composite objective function SA runs results match the true image much better than that of the variogram-only SA run. Although not shown, the bottomhole pressure comparisons are very similar to Dataset # 1.

2.4 Visual Comparisons for Dataset #3

Having verified that the two upscaling approaches are equally good, we focussed on a comparison of the modified SA algorithm (in which upscaling is performed via modified geometric averaging) and the variogram-only algorithm for dataset #3. For this case also it can be seen in Figure 6 that the composite objective function results are better in matching the true image visually.

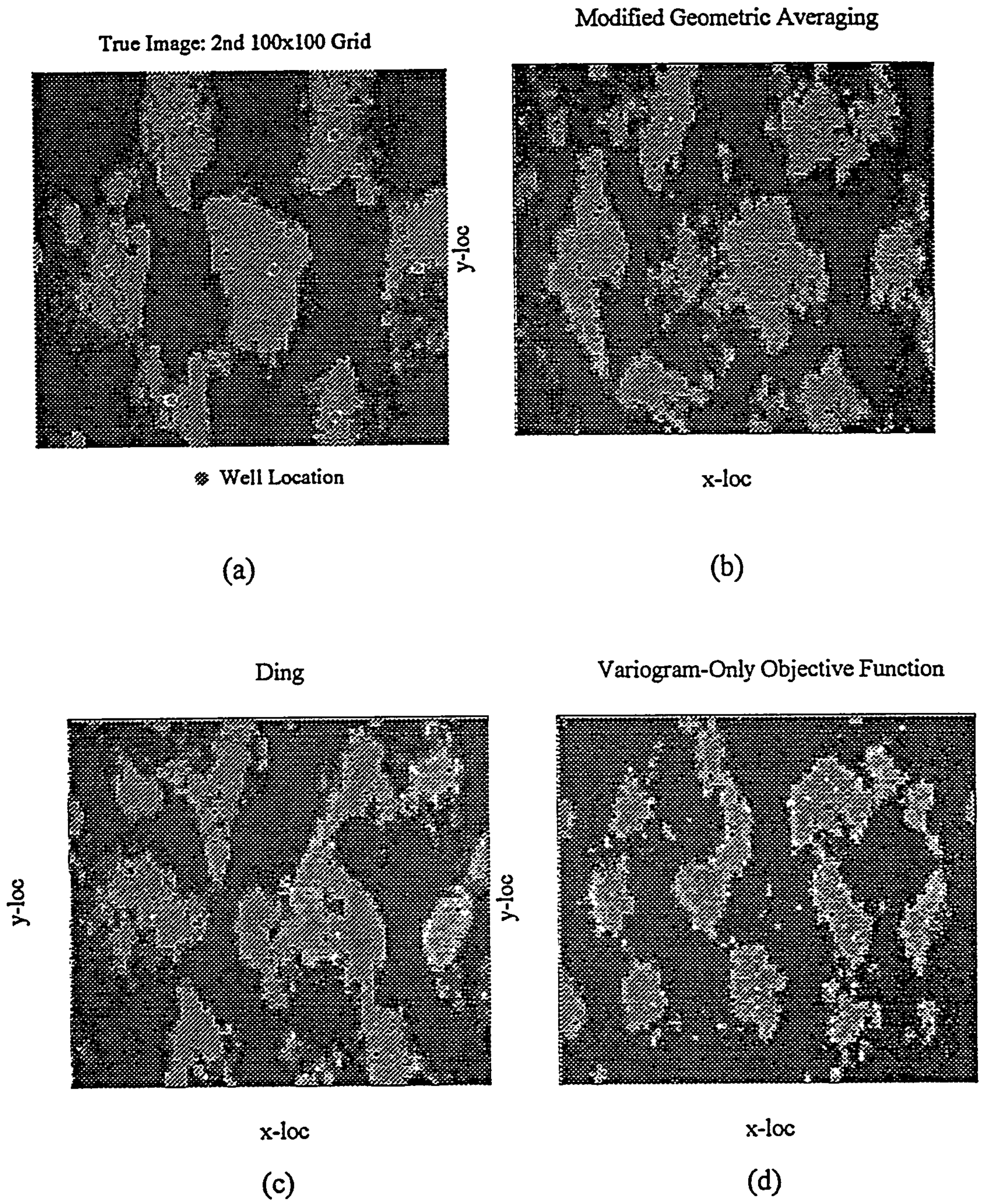


Figure 5: (a) Truth Case (b) Modified SA Run Results - 1

(c) Modified SA Run Results - 2 (d) Variogram-Only SA Run Results for Dataset #2

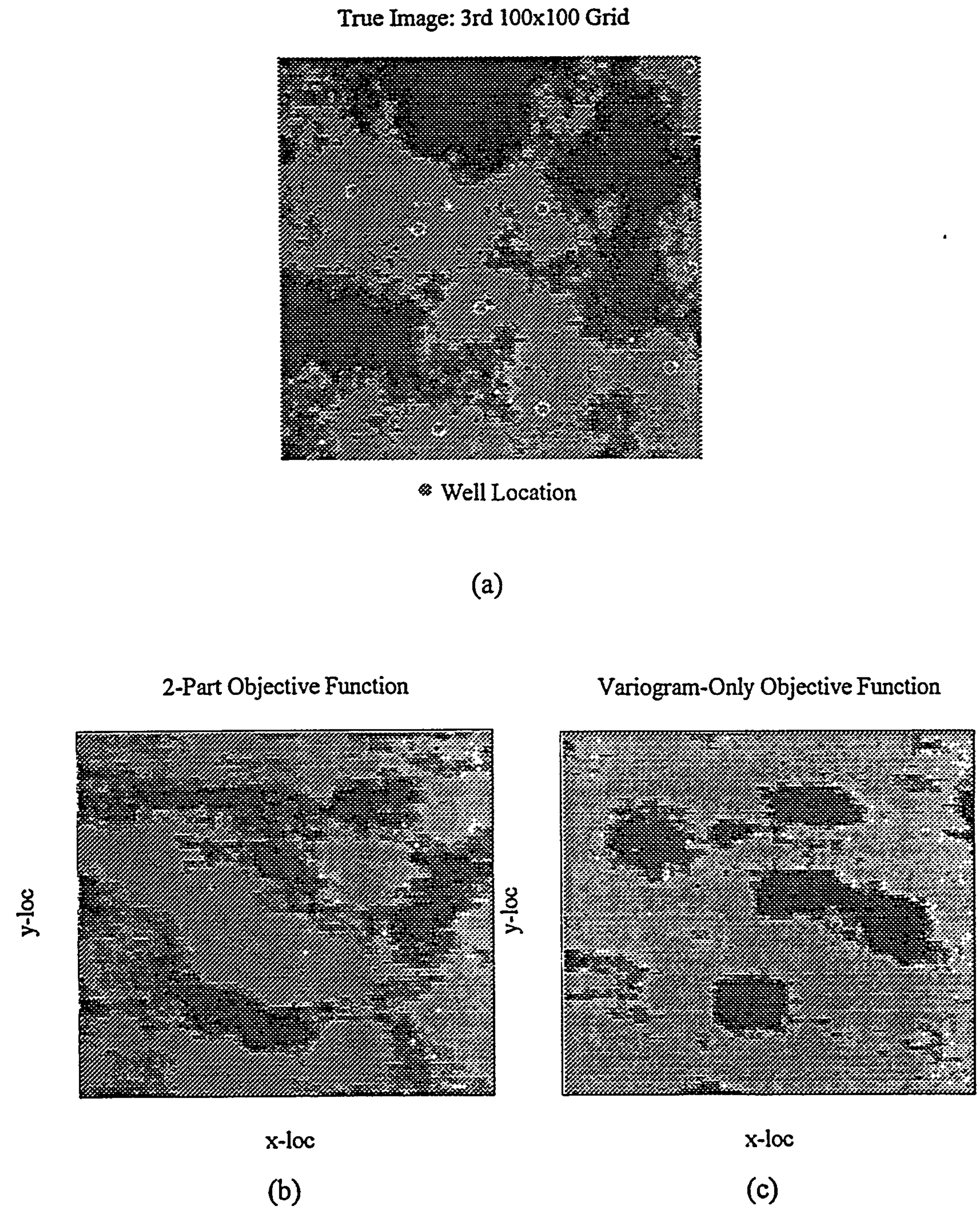


Figure 6: (a) Truth Case (b) Modified SA Run Results - 1
(c) Variogram-Only SA Run Results for Dataset #3

2.5 Sensitivites on Variogram Component

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 sparcity of data resulted in a variogram model which 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 7(b) 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 7(c), 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.

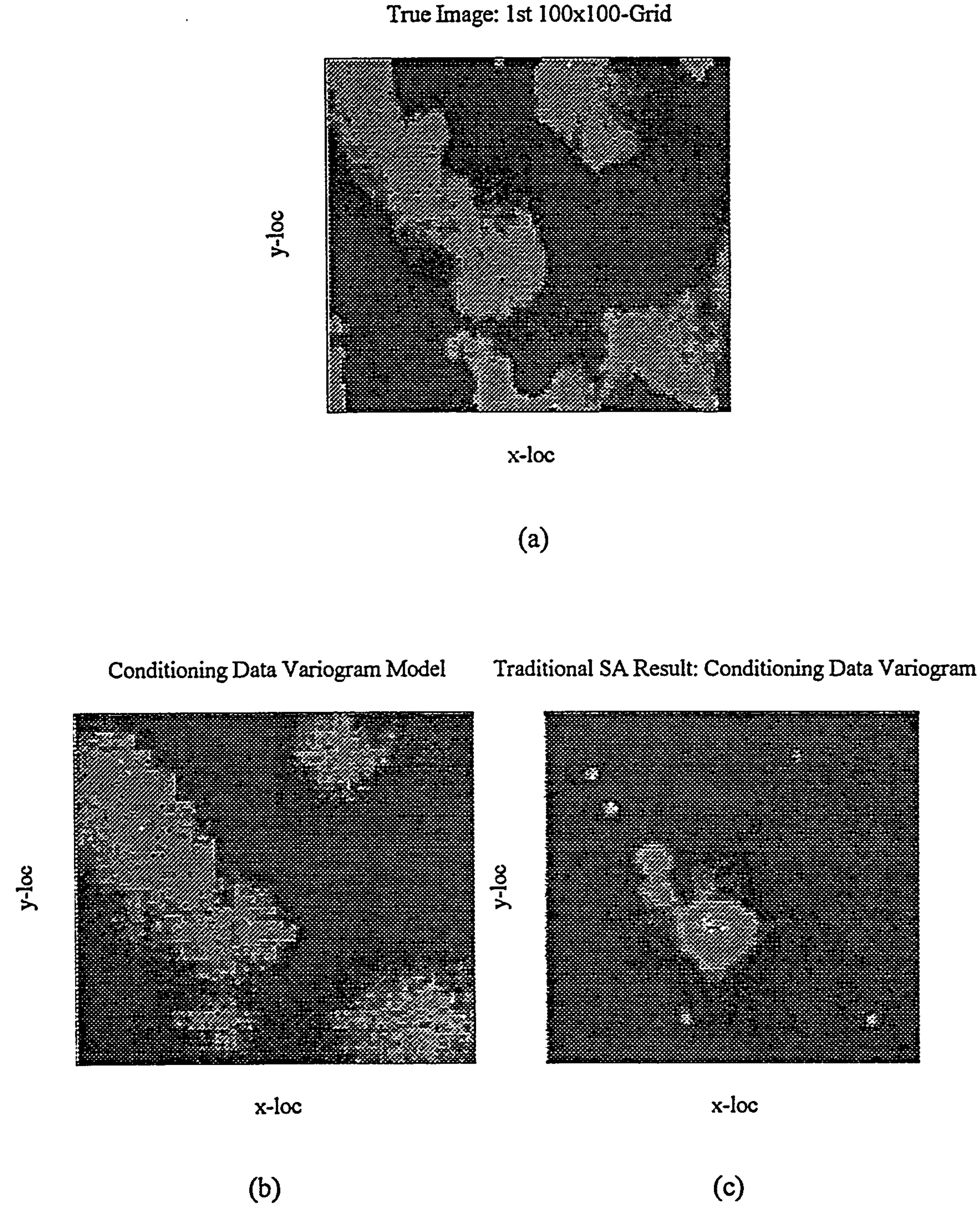


Figure 7: Results from Sensitivity Tests on Variogram Component

3. Integrated Lithofacies and Petrophysical Properties Simulation

This section presents the progress for the new procedure developed to generate reservoir models by simultaneously simulating the lithofacies and 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 main driver of the program has been developed and compiled with the previously-generated classes to obtain a working program. Much of the time was spent in debugging the code. A preliminary result, which is the reproduction of the results from the original Fortran version, has been obtained. The future work will emphasize the modification of the simulation technique and the development of the program interface.

3.1 Co-Simulation Program

Background theory used in developing the simulation technique was presented in the previous quarterly report. This report presents the description of the program structure followed by a discussion of the preliminary results.

3.1.1 Program Structure

The cosimulation program is written in the C++ language. The main purpose of writing the program in this language is to obtain reusable and extendable code while maintaining the ease in creating the interface in the spirit of a user friendly program package. The list of the classes used in the program is summarized in Table 1. The inter-relationship among these classes is shown in Figure 8. The arrowed line indicates the inheritance relationship, e.g., between class Application and class Cosim, whereas the line ending with a circle indicates where the class is being used. For example, class Variogram is used in class Kriging, and class Kriging is used in class GausSim.

Class Application is the main driver for the program. It is developed using the principal of polymorphism and dynamic binding where it contains a virtual function called DoSimulation. This virtual function is defined inside some other classes which are inherited from the Application class and dynamically binded during the execution of the program. Therefore, common features of the Application class, such as development of the grid system, variogram definition, correlations among variables, etc., can be used for different techniques of simulation. At this time, there is only one class derived from the Application class, i.e., the Cosim class. Future extension of this program would take advantage of this structure.

The Cosim class consists of the functions to perform the cosimulation of lithofacies and petrophysical properties. There are 3 main classes which are used in this class, i.e., the IndSim for Indicator Simulation, the GausSim for sequential Gaussian simulation, and the CondDist for conditional distribution technique for generating the permeability distribution. The class GaussSim can be used for either generating porosity distributions alone or for both rock type and porosity in which the technique of truncated Gaussian simulation (GTSIM) is applied. In the case where it is used for GTSIM, the IndSim is used for generating the proportion curve which is required in transforming the Gaussian simulated result back to its indicator value. At this time, the program allows the GTSIM technique only.

The kriging class is used to performed the kriging of the unsampled value. In developing the covariance matrix, this class can either use the covariance table which is provided by class CovTab, or directly calculate the covariance between two points given the variogram model. The class stores the information about the variogram model input by the user and provide the routine to calculate the variogram or covariance value between two given points in three dimensional space. The CovTab class stores the covariance value between two points which is defined by the super block searching technique.

Several other classes which are grouped together as the Miscellaneous class are used by almost all of other classes. This is due to its functionality that basically provides the utility function and common data structure.

Class Name	Description
Application	Main driver for the cosim program
Cosim	Define the simulation technique to be used
IndSim	Provide the procedure to perform indicator simulation.
GausSim	Provide the procedure to perform the simulation Gaussian.
CondDist	Provide the procedure to perform the conditional distribution technique in generating the permeability distribution and the storage of the related correlation between porosity and permeability
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.
Grid	Provide the grid block network of the simulation system that includes the neighborhood searching technique such as super block search.
Point3D	Provide the structure to represents a 3D point. This class is generated using a template that can accept any data-type.
Utility	Provide several utility functions that are common in geostatistical simulation technique such as random number generator, inverse of Gaussian data, normal transformation, etc.
listClass	Provide the link list of the data to store variable with unknown size. This class is templated to accept any data-type.
Matrix	Provide the procedure related to matrix operation. This class is templated.

Table 1: C++ Class Summary Used in the Cosim Program

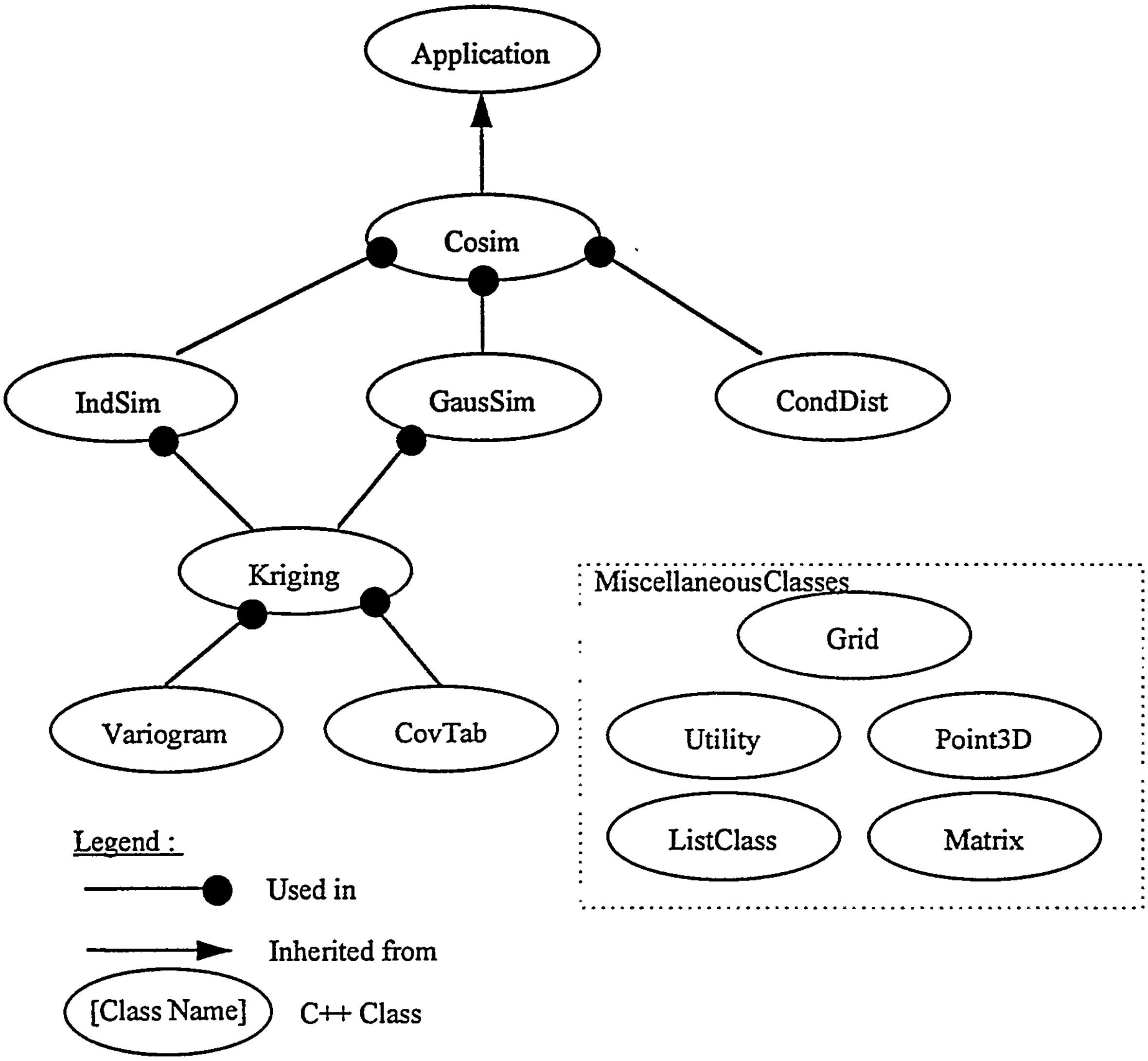


Figure 8: Program Structure

3.1.2 Simulation Results

The progress that has been achieved at this point is the simulation using the truncated Gaussian technique as it was written in the original Fortran version. The result from the new program matches very well with this Fortran version. That signifies the reproducibility of the program.

Figures 9A through 9C show cross sections of rock type, porosity, and permeability, respectively, of the simulation result using the data from North Robertson Unit, West Texas. The number of the grid blocks used in creating this pictures is 19,602 (99x99x2). Based on the geological description, there are five rock types defined, namely rock type 1, 2, 3, 4, and 5. But, for the cross section presented in these figure rock type 5 is absent. The formation type is carbonate with a low porosity and permeability. From these three figures we can see how each variable is connected to each other areally.

Comparing these three figures we can see how the generated petrophysical properties (porosity and permeability) are consistently follows the underlying rock type which is the main feature of this program. Figure 10 shows 3D view of the rock type distribution to demonstrate the capability of the program in generating the 3D problem.

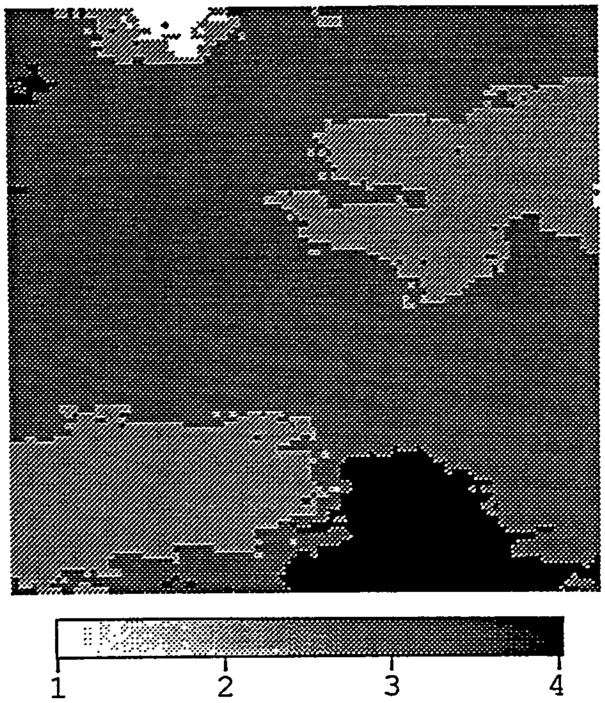


Figure 9A **Rock Type Cross Section**

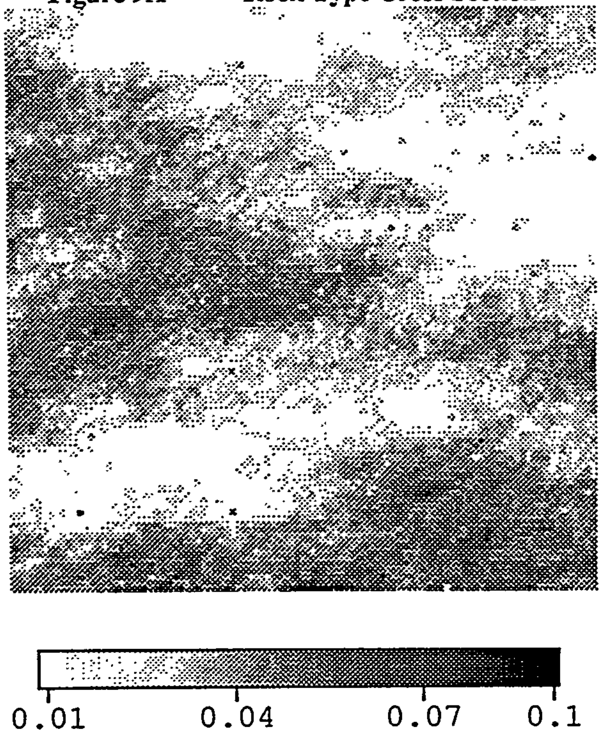


Figure 9B **Porosity Cross Section**

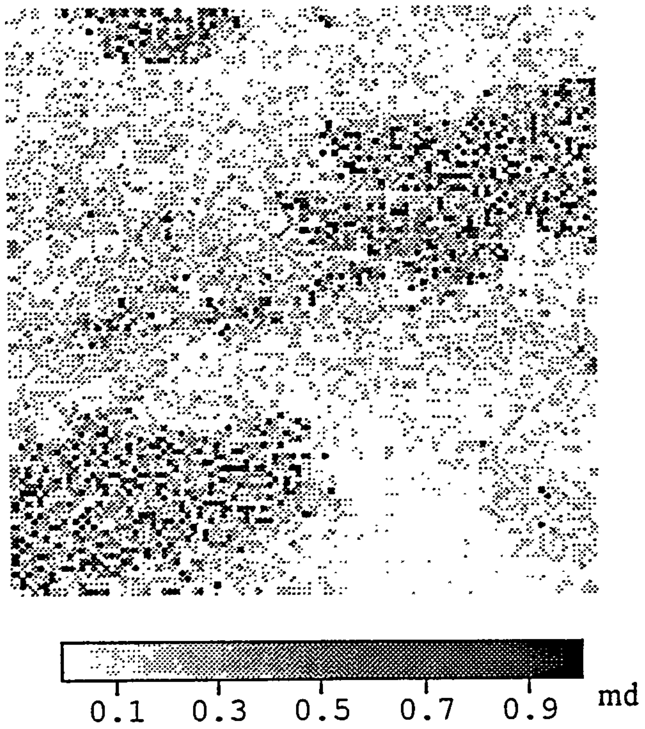


Figure 9C Permeability Cross Section

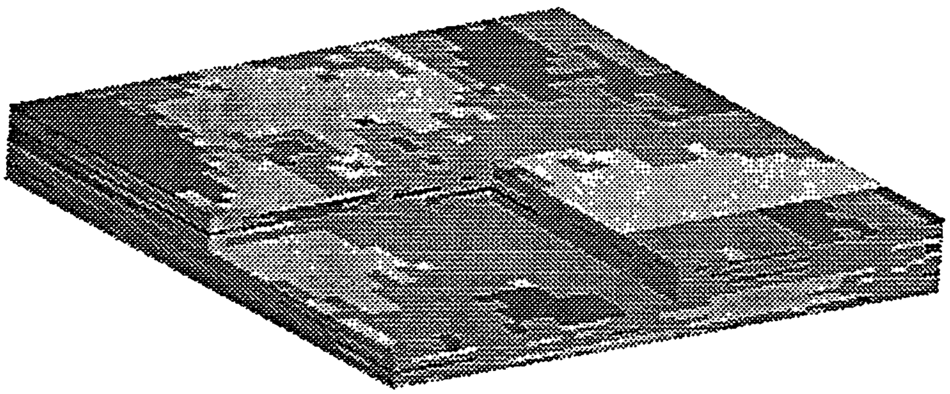


Figure Asnul 10 Three dimensional view of rock type distribution

3.1.3 Future Work

The first modification that is going to be done in the near future is the incorporation of indicator simulation technique in generating the rock type distribution. This modification includes the incorporation of soft data to accommodate the uncertainty in the data itself. Using this modified program, an interface program will be built to make the program user friendly.

4. Geological System: 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

4.1 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.

4.1.1 Applying Well Log Segmentation Rules

Figure 11 shows how this rules are applied to a section of a log:

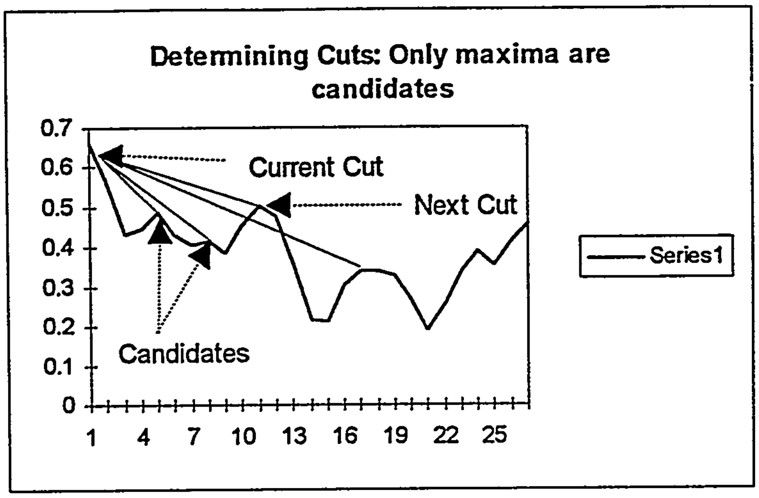


Figure 11

4.1.2 Segmentation Results

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 (Figure12):

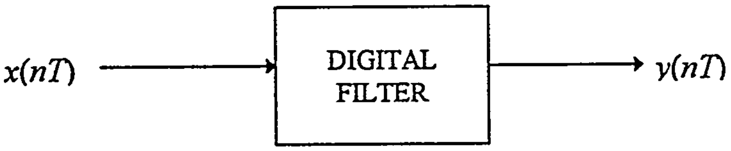


Figure 12: Digital Filtering.

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 α 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

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

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 13.

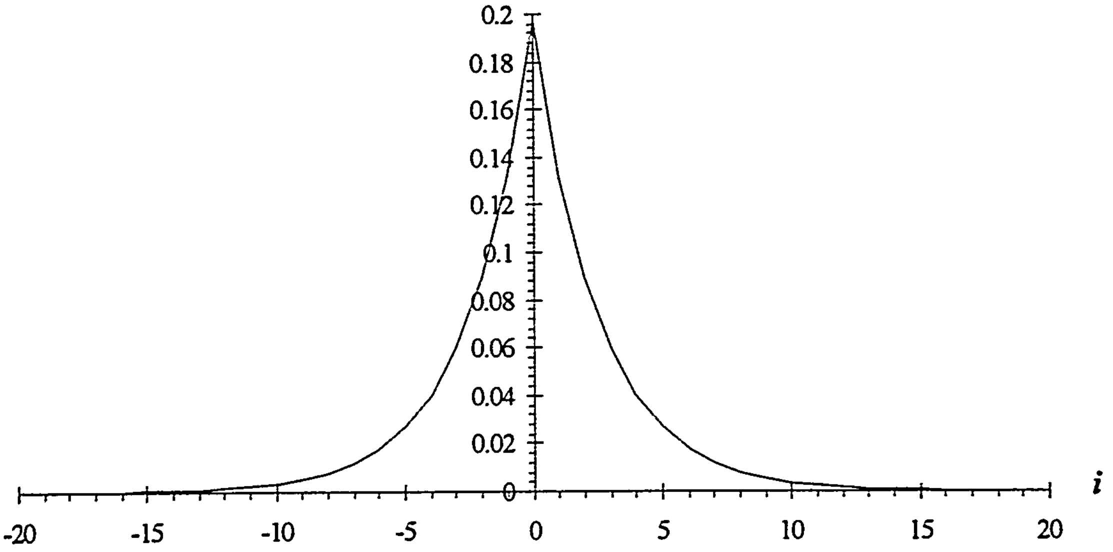


Figure 13: Filter parameters.

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 14:

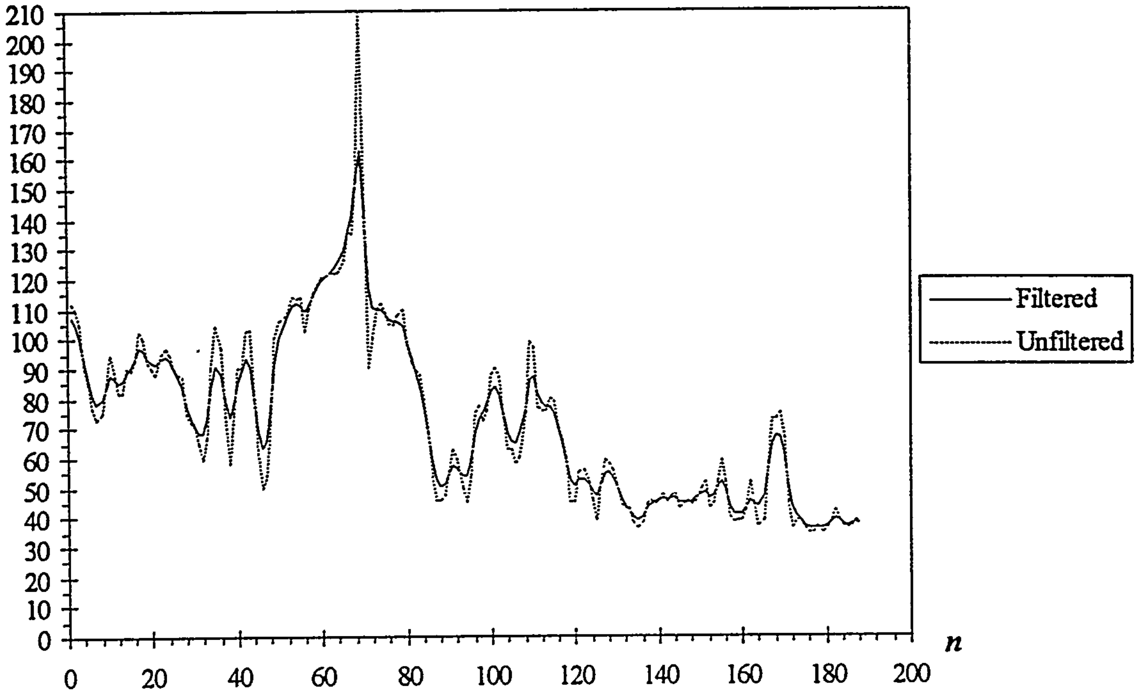


Figure 14: Results of filtering.

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 which has the larger gamma ray (Gr.) value. Figure 15 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.

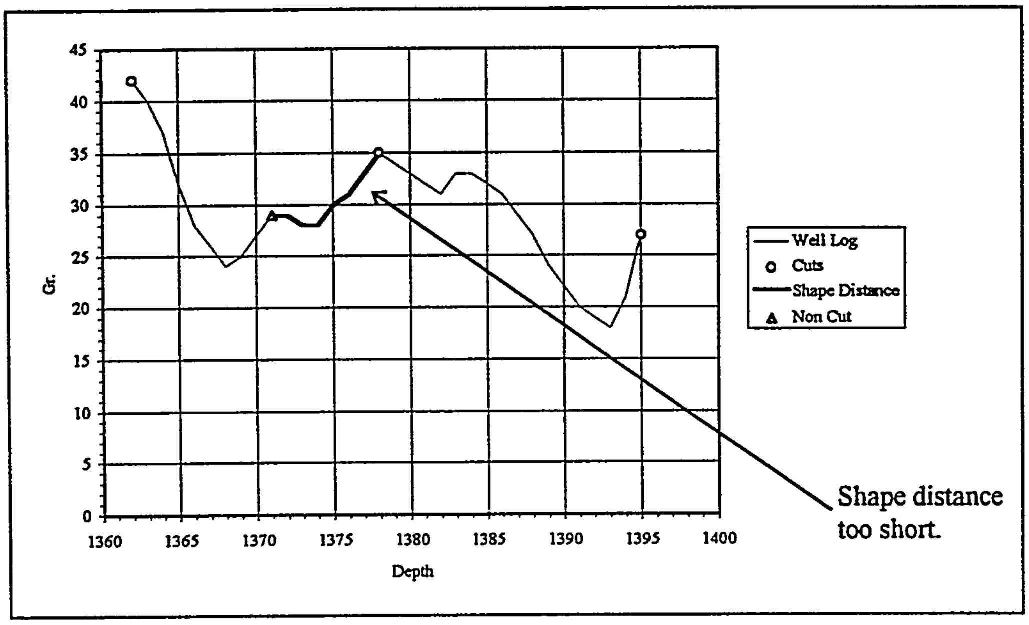


Figure 15: Explanation of the cut elimination rule.

Figure 16 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 supplies by an expert geologist (shown in Figure 17).

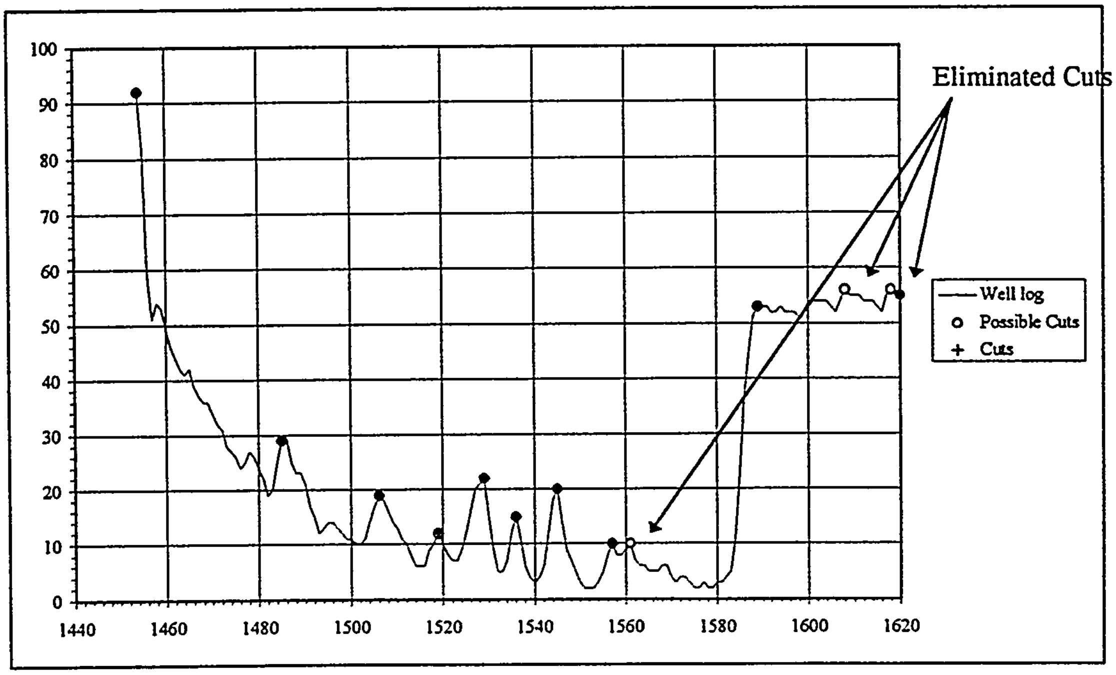


Figure 16: Results obtained with and without the cut elimination rule.

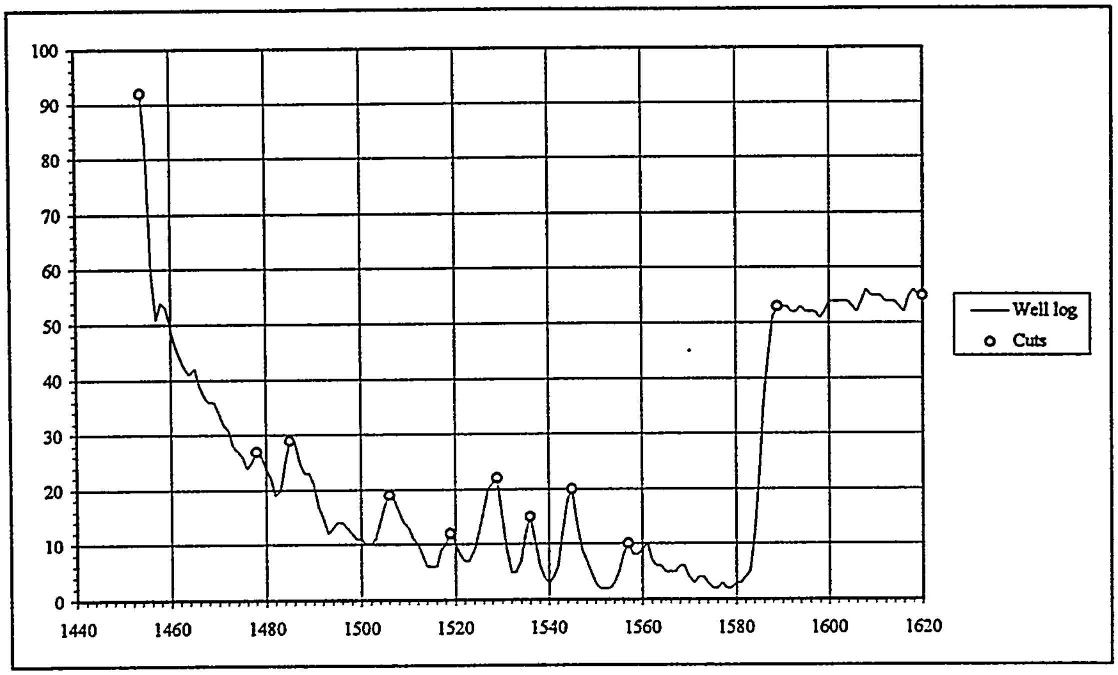


Figure 17: Results obtained from an expert geologist.

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. Table 2 summarizes the results.

Well Log	Number of Cuts			Missing Cuts		Extra Cuts	
	Right	Before	After	Before	After	Before	After
Self 56	13	16	14	1	1	4	2
Self 78	18	20	17	1	3	3	2
Self 81	14	11	12	3	3	1	0
Self 82	11	13	10	1	1	3	0

Table 2: Summary of Results.

4.2 Expert Poll for Log Segmentation and Log Facies Identification:

The ability of the neural network to identify log facies accurately is largely dependent on the accuracy of the log segmentation algorithm. Although the results have improved (80% correct) substantially with the new algorithm, the level of accuracy could still be improved. To date, the assessment of accuracy of the neural network module output is based on a single expert. Before embarking on additional changes or modifications, the research group decided that the diversity of opinion among experts should be polled.

The polling experiment is based on standard wireline logs from 3 wells (2 Glenn Pool; 1 Frio). The log displays GR, SP (Glenn Pool logs only) and 3 resistivity curves at a 1-in. = 20-ft. scale over about 200 ft. of thickness. The location of the wells and log depths are withheld from the subjects of the experiment. The subjects are given a brief overview of the purpose of the neural network module and how the polling results are to be used. They are given instructions for completing the experiment. The subjects are asked a few questions regarding their background. The subjects are located in the Tulsa area; this was mainly done to expedite the polling. The subjects, however, come from a variety of backgrounds and some have never worked either of the areas covered by the logs used in the experiment.

The results are expected back by June 1. A tally will be made to quantify the diversity of selection among the subjects. A cross tabulation between experience/background factors and selection outcome will also be compiled to determine any biases.

5. Geological System Components: Correlation of Log Curves

In this section, we detail the current approach to the correlation of log curves.

5.1 Overview of Approach

The approach has been to develop a rule-based correlation of Gamma Ray logs of two wells. The rules are based on similarities in well log trace shapes, thickness and vertical position of the zones. The segmentation of the well logs and log-facies identification by the neural network and depths of identified marker beds will be given as input.

As mentioned in the previous approach, we formulated the rules based on the four criteria listed below, with the zones and the log-facies being automatically determined by the well-log segmentation and neural network programs described earlier.

1. position of the zones with respect to the length of the logs;
2. distance of the zones from the marker beds;
3. thickness of the zones;
4. log-facies of the zones;

To further improved our resolutisn of the correlation matrix, we are constraining the distance of the zones from the marker bed and thickness of the zones criteria that are considered for formulating the rules. The variation in distance of the zones from the marker bed is a function of the distance between two wells and stratigraphic dip angle. The key idea for considering vertical distance from the marker bed is that as the difference in distance from marker bed increases, their correlatibility ranking goes down. Two stratigraphic units are not considered correlative if the absolute value of the difference in distance from the marker bed is greater than $x\tan\theta$, where x = distance between two wells, and θ = stratigraphic dip angle which is set at 5 degrees by default. Similary, in order to constrain the thickness difference between two zones begin compared, we impose that when the difference in thickness goes beyond $x\tan\theta$, the zones under consideration are not correlative.

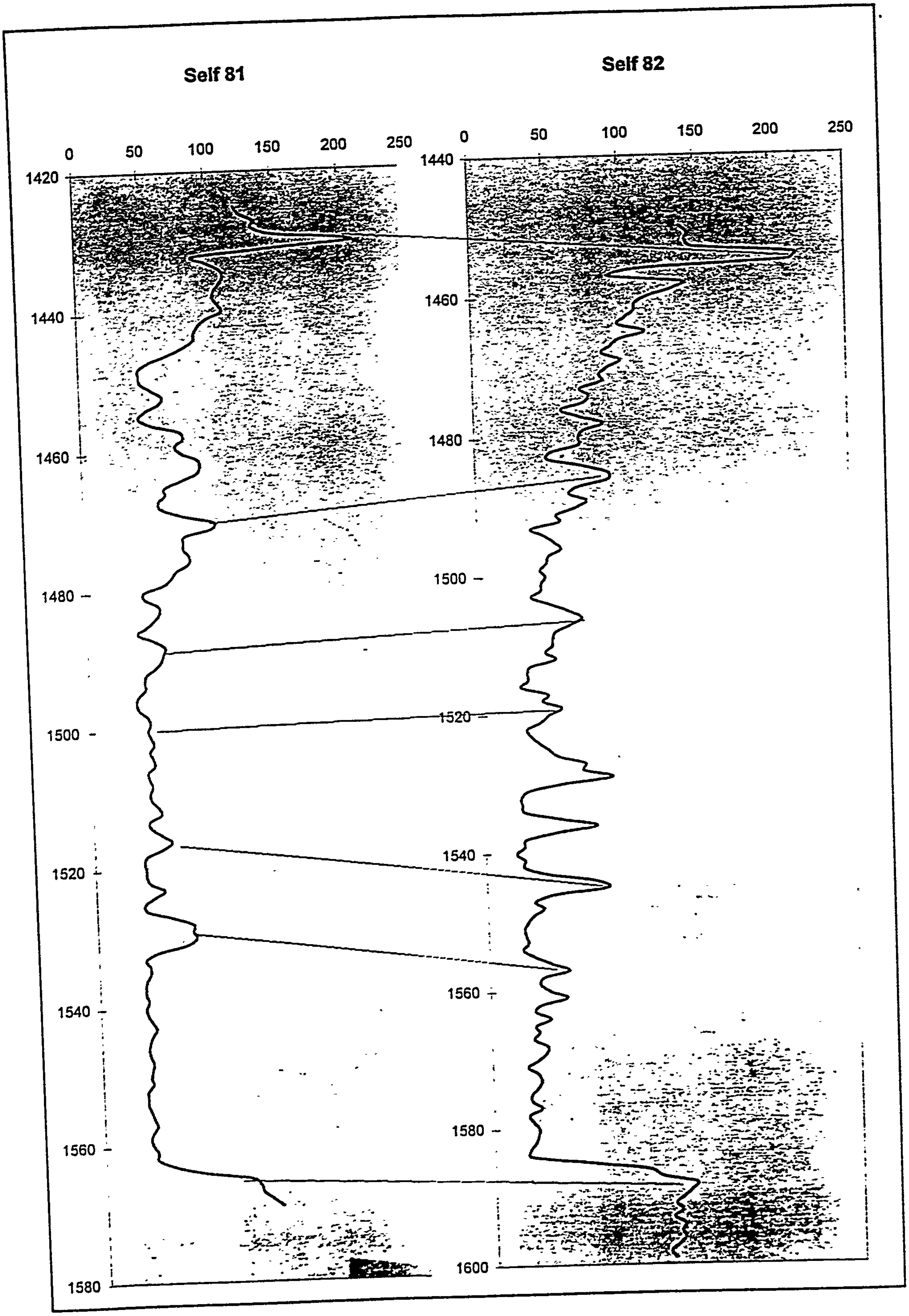
5.2 Performing the Correlation of Two Wells from the Matrix

Once the rules have determined the correlation values, the constructed matrix is analyzed to establish where the zones actually correlate. This analysis is performed by a small expert system. The analysis is based on the following facts.

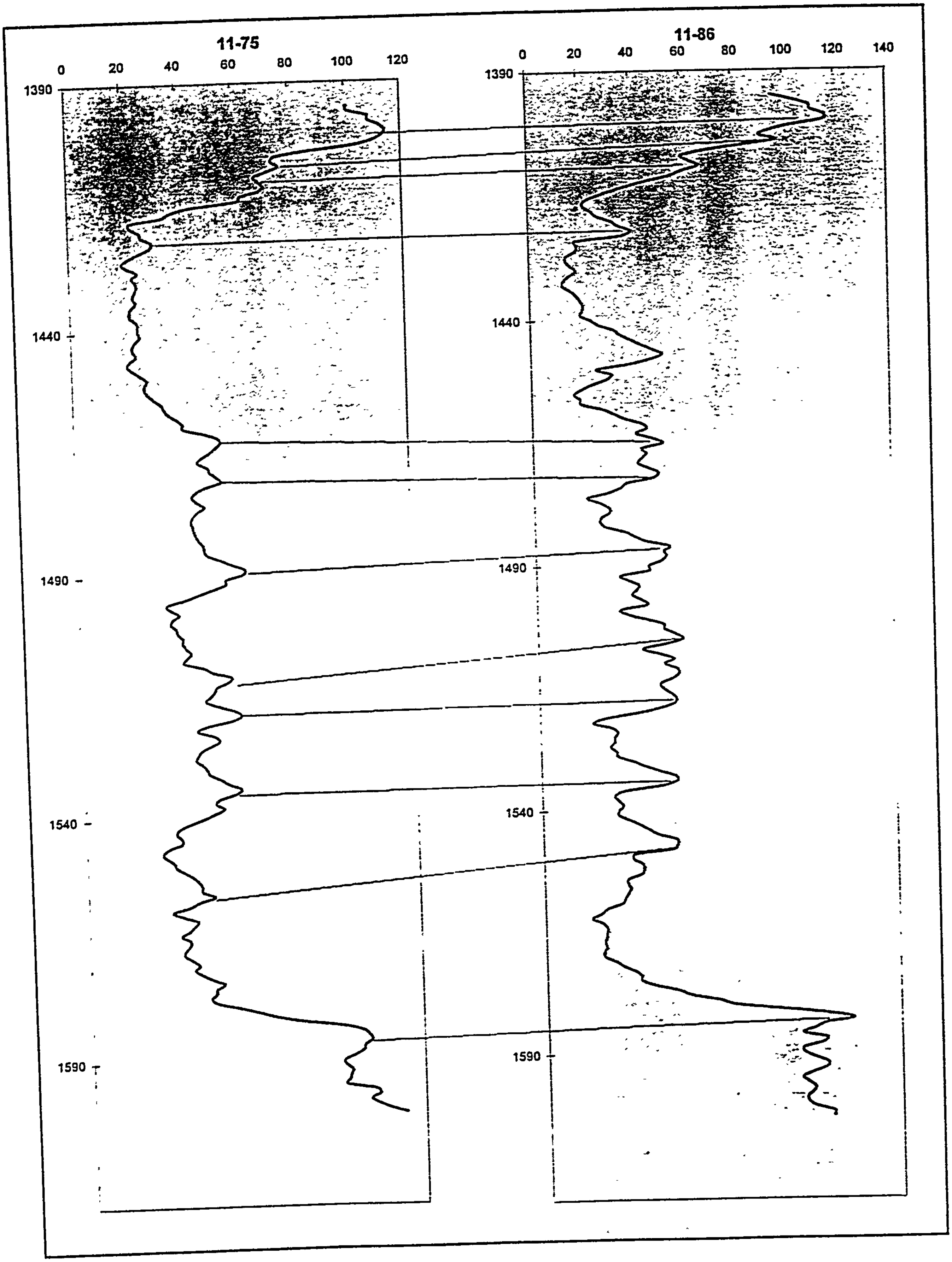
1. the ranking of the rules is strictly linear, i.e., a one point difference counts
2. a maximum value for a row and column is the first choice for correlation
3. the neural network may miss cuts making zones appear larger
4. large grain correlation, as in combining zones, is better than pinch outs

We have tested the complete correlation between two sets of wells from Glenn Pool field: (1) Self 81 with Self 82 and (2) 11-75 with 11-86. The logs were manually zoned and the log facies of each zone was identified for comparison with the automated approach. The correlation by a geologist is show in the graphs below for Self 81 and Self 82 and for 11-75 and 11-86, respectively.

Correlation by a geologist



Correlation by a geologist



5.3 Expert System Correlation Results

In this section we detail the expert system approach to correlating the wells depicted above using the martix information. Table 3 shows the values of the matrix for Self 81 and Self 82 when the zones and facies are obtained by the well-log segmentation and neural network programs.

		Self 81				
	Zone	a1	a2	a3	a4	a5
Self 82	b1	186	21	128	16	4
	b2	132	156	141	16	16
	b3	133	51	186	22	34
	b4	40	54	34	184	132
	b5	39	135	33	147	36
	b6	39	135	33	24	159
	b7	21	132	51	42	182
	b8	35	20	47	52	40

Table 3

The zones in the above table correspond to the following depths.

a1 : 1430 - 1462
a2 : 1462 - 1471
a3 : 1471 - 1488
a4 : 1488 - 1516
a5 : 1516 - 1529

b1 : 1454 - 1485
b2 : 1485 - 1506
b3 : 1506 - 1518
b4 : 1518 - 1529
b5 : 1529 - 1536
b6 : 1536 - 1545
b7 : 1545 - 1557
b8 : 1557 - 1589

Several passes of the expert system rules were required over the matrix to obtain a full correlation. The first pass generates all of the values that maximums for rows and columns. Those values that are both a maximum for a row and a column results in a correlation of those zones. The results of the first pass on the matrix in Table 3 are as follows.

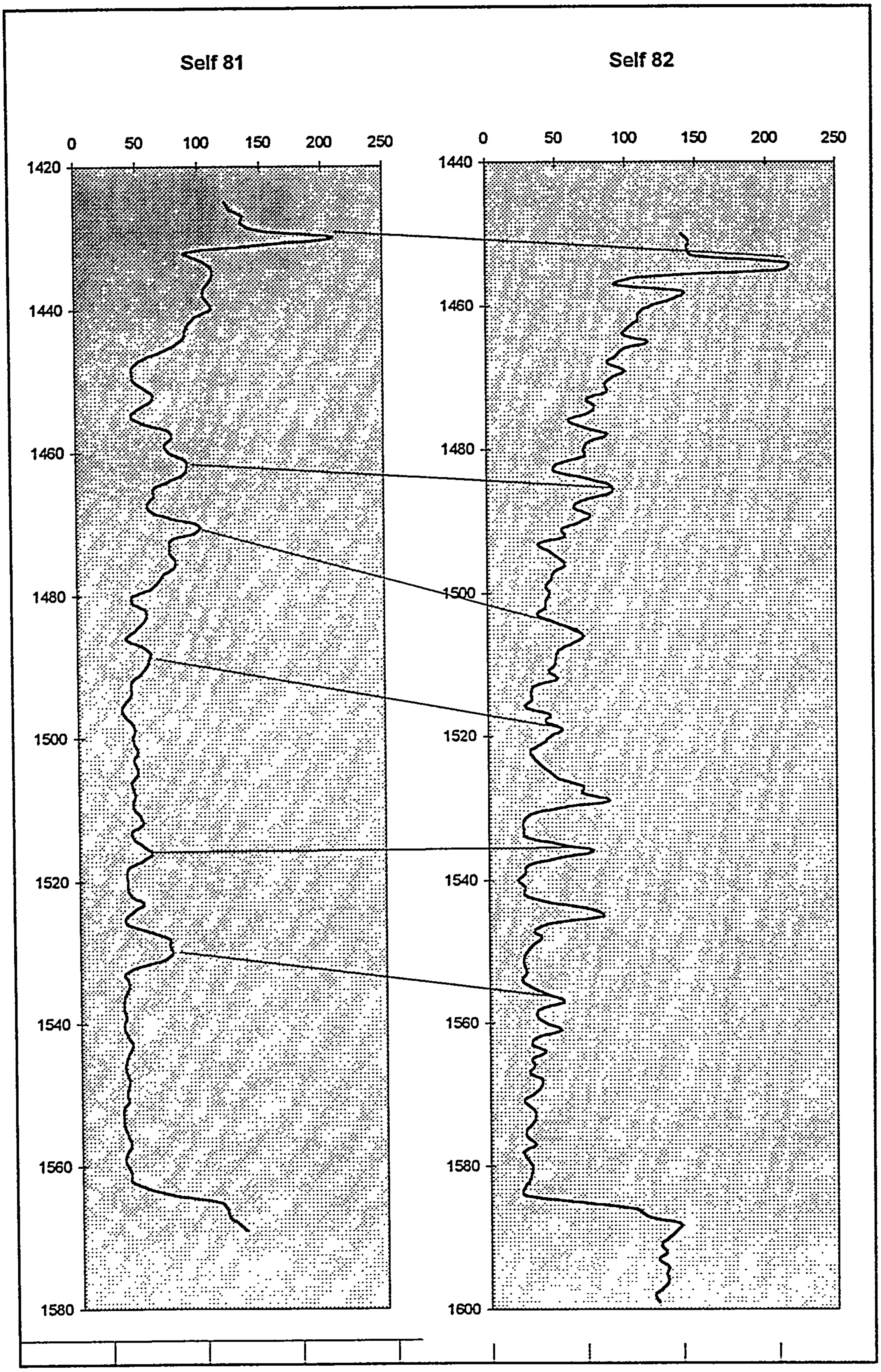
First Pass Correlated
a1 - b1
a2 - b2
a3 - b3
a4 - b4
a5 - b7

If all zones are correlated, the system will then analyze the correlation (discussed below). If there are some zones that are not correlated, the system attempts to "fit" them into the currently correlated zones. This fitting is performed by examining a neighbor zone to see if its correlated zone is compatible. The results of the second pass are as follows.

Second pass
a1 - b1
a2 - b2
a3 - b3
a4 - b4
a4 - b5
a5 - b6
a5 - b7

An analysis of the values in the second pass determines that all Self 81 zones are correlated and all Self 82 zones, except zone 8 are correlated. Because zone 8 in Self 82 has values under 100 and it is the lowest depth zone in Self 82, it is deemed uncorrelatable. Thus, it is assumed to be correlated with the remaining well log information in Self 81. This situation occurs because the well-log segmentation program missed a cut in Self 81.

Once all zones are correlated, the system analyzes the results for any problems, such as a cross-over between zones. It also reevaluates the values to determine if a maximum could be used in place of a current value. The above correlation is determined to be complete. The resulting correlation is seen in the graph below, which can be compared with the earlier graph depicting the geologist's interpretation.


Self 81
Self 82
0
50
100
150
200
250
1420
1440
1460
1480
1500
1520
1540
1560
1580
0
50
100
150
200
250
1440
1460
1480
1500
1520
1540
1560
1580
1600

The determination of the correlation for 11-75 and 11-86 is similar to the Self 81 and Self 82 correlation, except that more passes are required. The extra passes are due to a cross over appearing early in the correlated zones. The cross over causes remaining zones for be "fit" improperly. Thus, when the correlation is analyzed and the cross over is detected, some initial correlations are deleted and must be redone. The matrix is in Table 4.

		11-75				
	Zone	a1	a2	a3	a4	a5
11-86	b1	133	35	33	3	16
	b2	168	34	36	12	129
	b3	21	50	134	130	102
	b4	130	181	51	21	106
	b5	22	52	182	102	162
	b6	127	35	105	111	124
	b7	3	26	160	187	114
	b8	5	162	104	122	180

Table 4

where the zones correspond to the following depths

a1 : 1400 - 1463
a2 : 1463 - 1489
a3 : 1489 - 1519
a4 : 1519 - 1556
a5 : 1556 - 1586

b1 : 1399 - 1422
b2 : 1422 - 1447
b3 : 1447 - 1465
b4 : 1465 - 1487
b5 : 1487 - 1511
b6 : 1511 - 1533
b7 : 1533 - 1547
b8 : 1547 - 1583

Determining the correlation from the maximum values for each row and column, the initial results appear below. It should be noted that all zones for 11-75 are correlated.

First pass
a1 - b2
a2 - b4
a3 - b5
a4 - b7
a5 - b8

Given the above correlation in the first pass, a second pass is made in which to "fit" those zones not initially correlated. The results of this pass appear below.

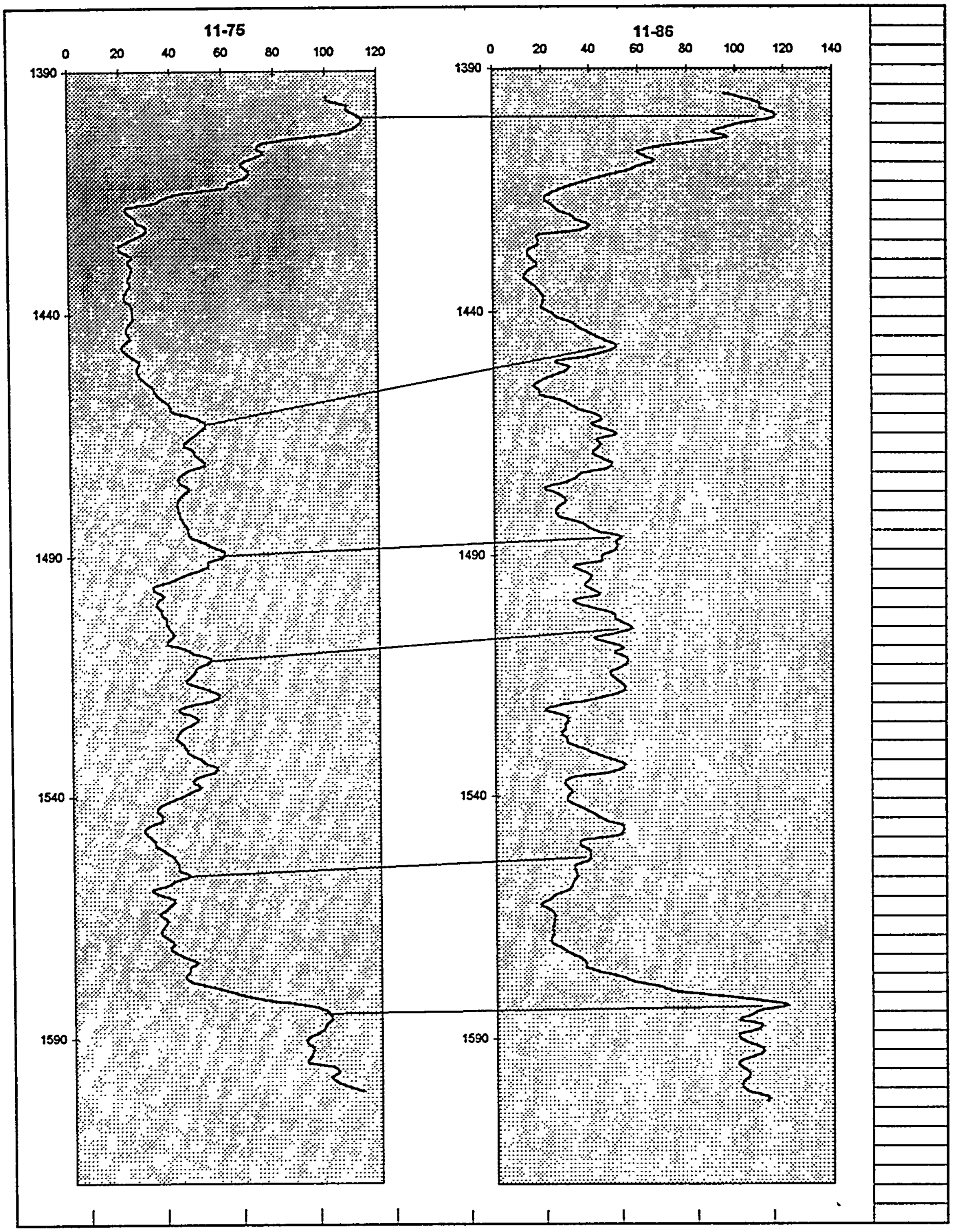
Second pass
a1 - b1
a1 - b2
a2 - b4

a3 - b5
a4 - b7
a5 - b8

After two more passes the final correlation appears below.

a1 - b1
a1 - b2
a2 - b3
a2 - b4
a3 - b5
a4 - b6
a4 - b7
a5 - b7

The results are shown in the graph below which can be compared with the earlier geologist's results from manual correlation. The system's results are fairly consistent with the geologist, though additional improvement in all aspects of the automation is needed.


11-75
0
20
40
60
80
100
120
1390
1440
1490
1540
1590
11-86
0
20
40
60
80
100
120
140
1390
1440
1490
1540
1590

6. Well Model Indentification System

The Well Model Identification System consists of two subsystems in well model identification. The first subsystem (Subsystem-1) uses the time/pressure data derived from well sites for model interpretation, and the second subsystem (Subsystem-2) obtains information from geologist and uses Inference Engine for model identification and selection.

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 18 below shows the 3 possible results in the list.

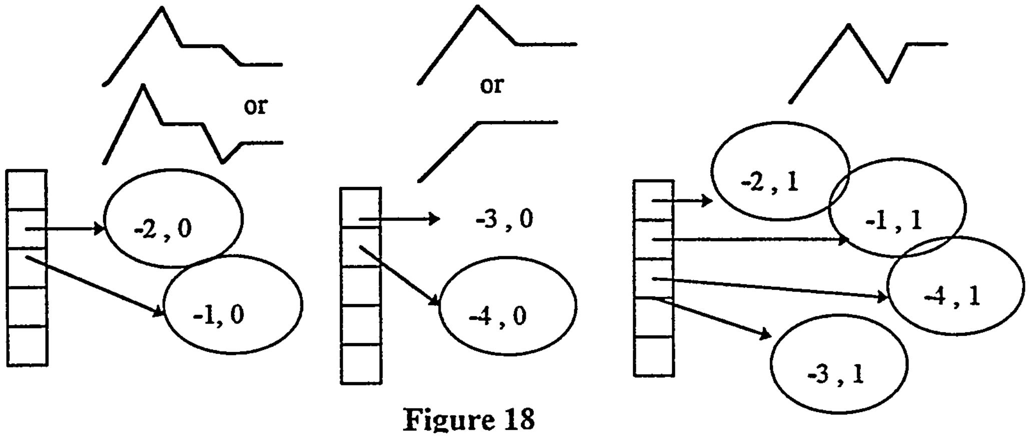


Figure 18

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 19 below shows the all possible shapes and the all possible models.

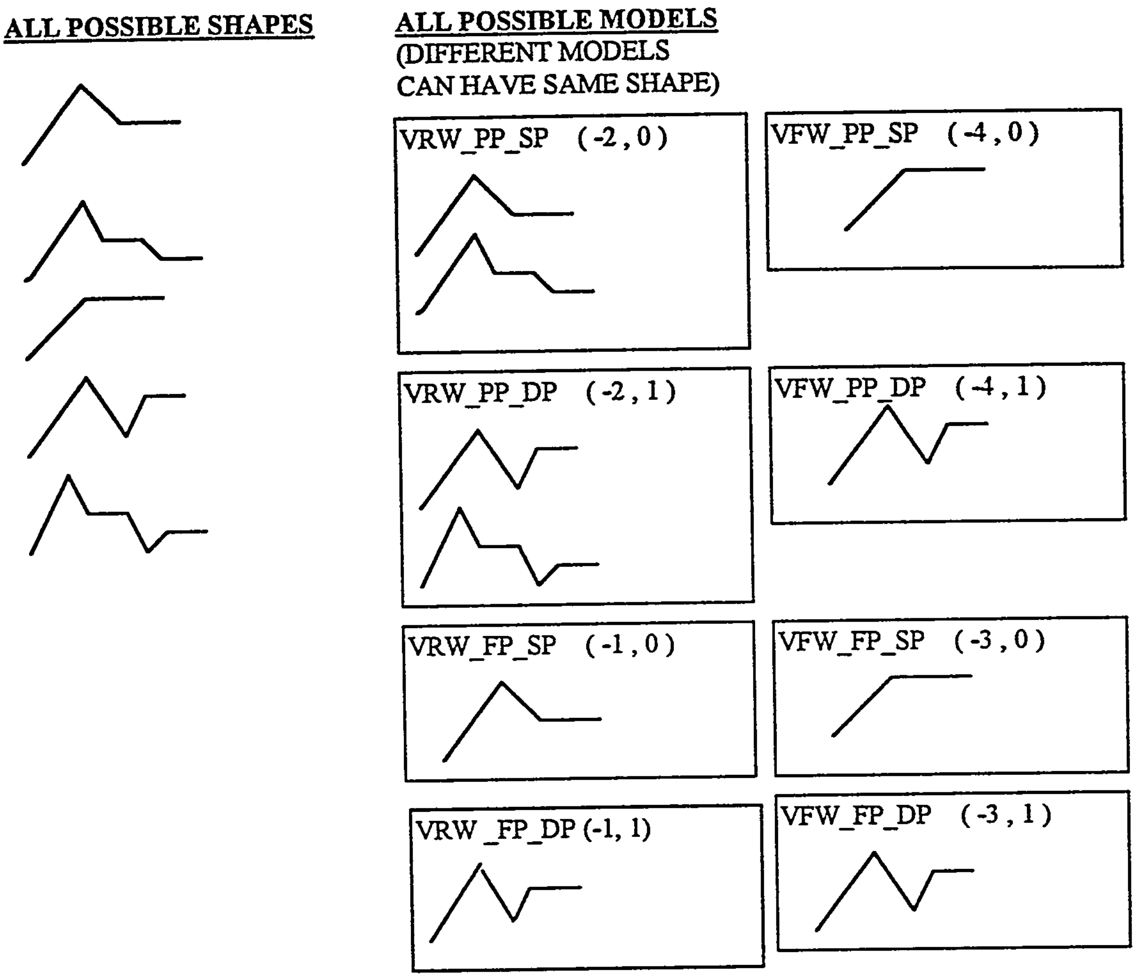


Figure 19

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 20. 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 21 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 1st 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. Further work is being done to convert the Well Model Identification System into MSVC 4.0

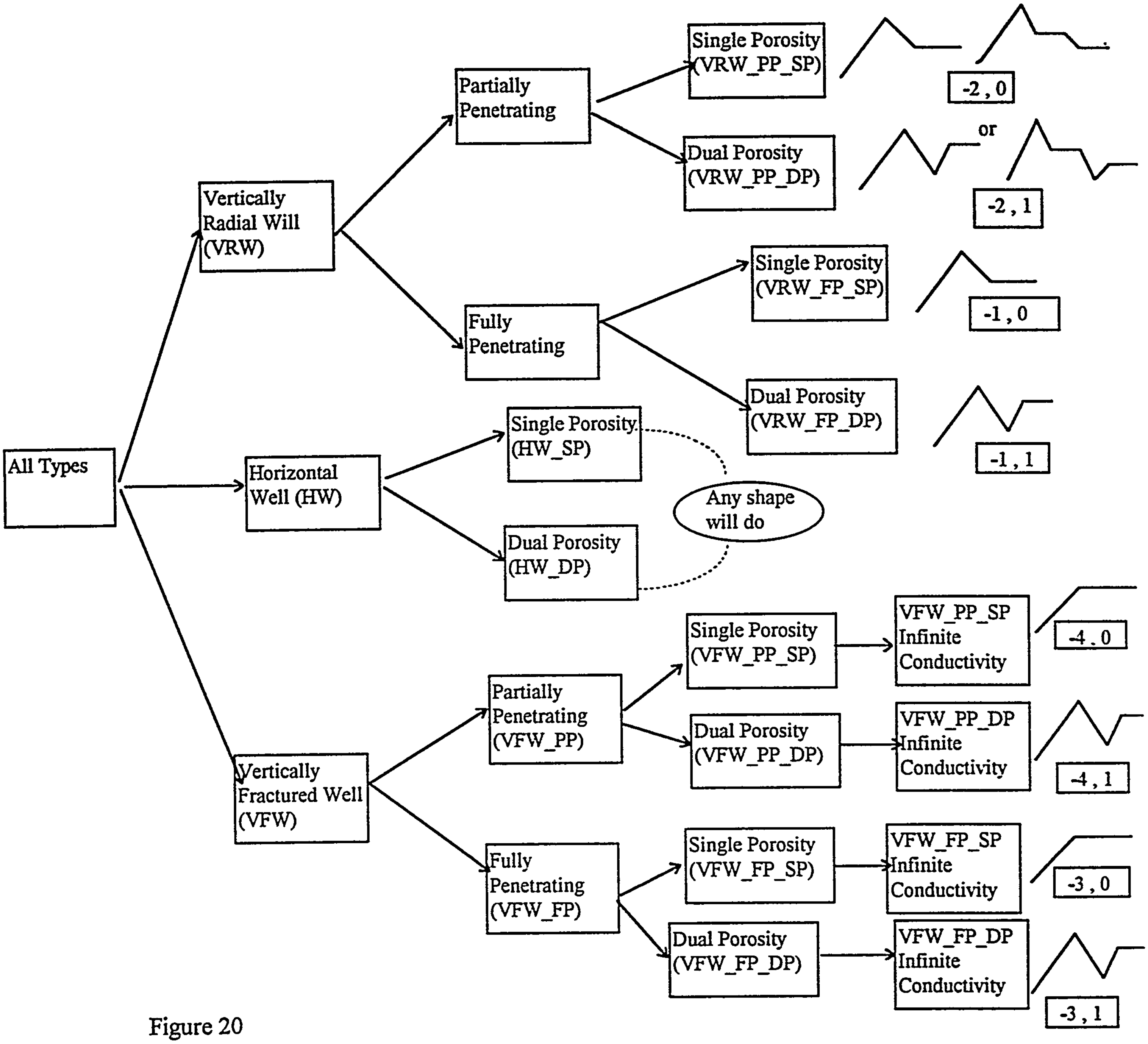
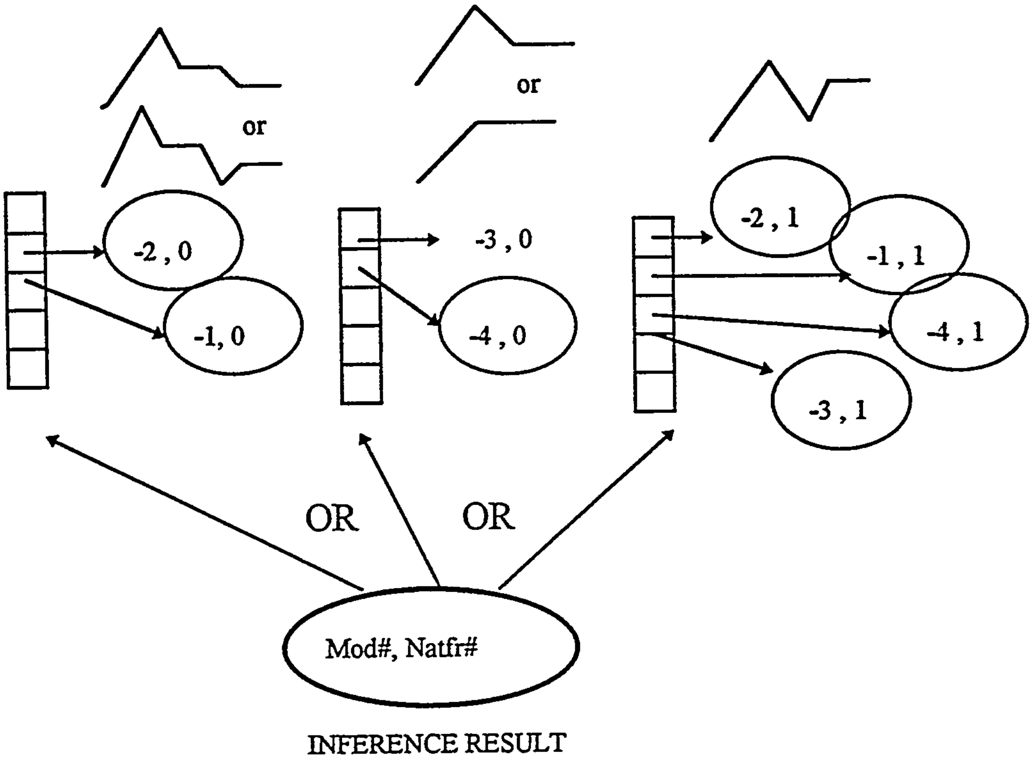

All Types
Vertically Radial Will (VRW)
Partially Penetrating
Single Porosity (VRW_PP_SP)
-2 , 0
or
Dual Porosity (VRW_PP_DP)
-2 , 1
Fully Penetrating
Single Porosity (VRW_FP_SP)
-1 , 0
Dual Porosity (VRW_FP_DP)
-1 , 1
Horizontal Well (HW)
Single Porosity (HW_SP)
Dual Porosity (HW_DP)
Any shape will do
Vertically Fractured Well (VFW)
Partially Penetrating (VFW_PP)
Single Porosity (VFW_PP_SP)
VFW_PP_SP Infinite Conductivity
-4 , 0
Dual Porosity (VFW_PP_DP)
VFW_PP_DP Infinite Conductivity
-4 , 1
Fully Penetrating (VFW_FP)
Single Porosity (VFW_FP_SP)
VFW_FP_SP Infinite Conductivity
-3 , 0
Dual Porosity (VFW_FP_DP)
VFW_FP_DP Infinite Conductivity
-3 , 1

Figure 20

Figure 21



7. References

1. Deutsch, C.V. and Journel, A.G.: *GSLIB Geostatistical Software Library and User's Guide*, Oxford University Press, New York (1992).
2. Ding, Yu: "Scaling-up in the Vicinity of Wells in Heterogeneous Field", paper SPE 29137 presented at the 13th SPE Symposium on Reservoir Simulation held San Antonio, TX, U.S.A., 12-15 February 1995.
3. *ECLIPSE 100 - Black Oil Simulator*, ECL-Bergeson Petroleum Technologies, Inc., Oxfordshire, England (1990).