

**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. Sheno
Reporting Period:	Jan. 1, 1997 - Mar. 31, 1997

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 is nearing completion. This system 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 correlation results and integrating the various components of the system under one interface. The well test interpretation system is nearly complete with the exception to a graphical display of the models and its integration with the geostatistical system. The designing of the overall user interface to integrate all of the systems is continuing.

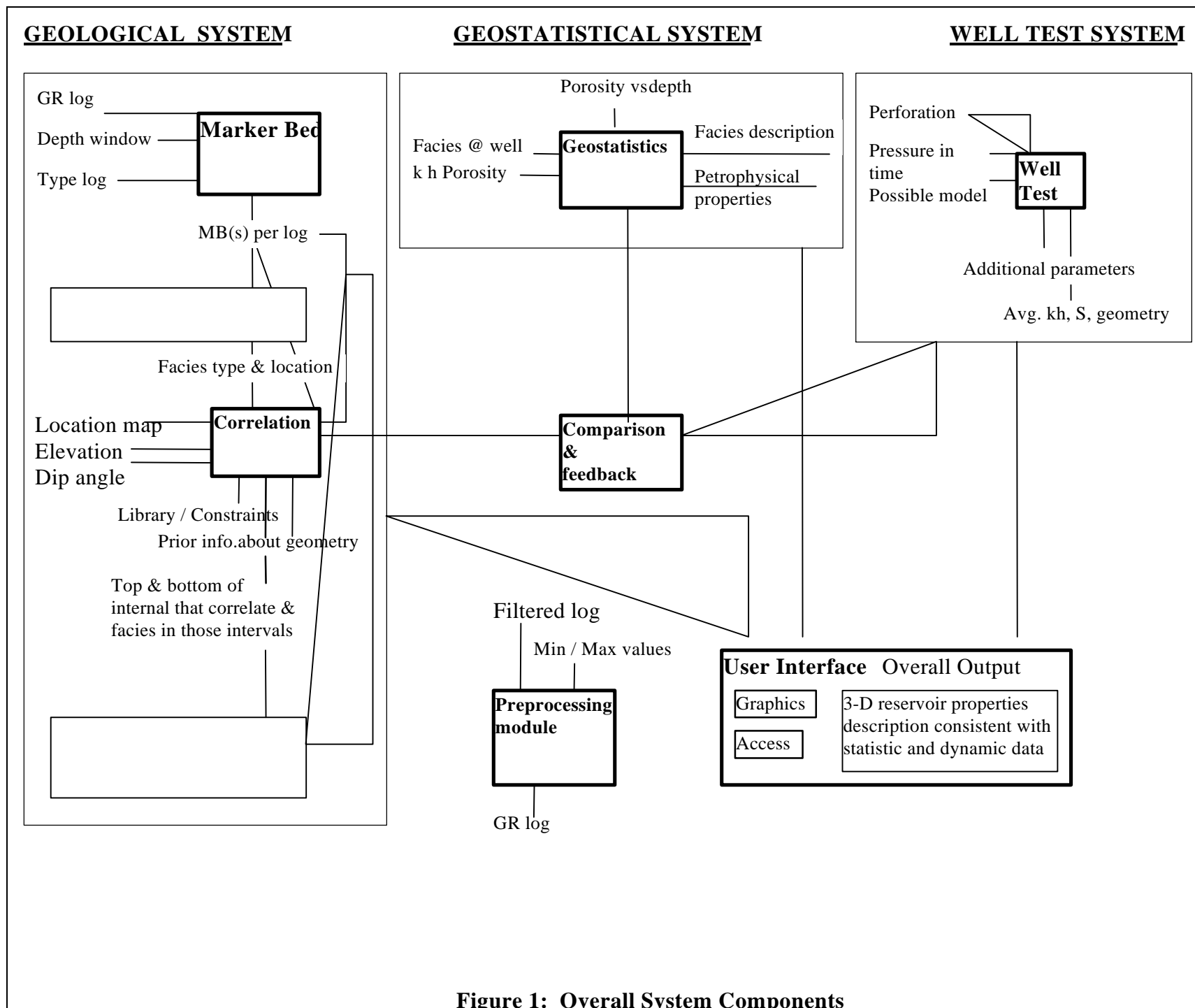


Figure 1: Overall System Components

2. Geostatistical System: Horizontal Variogram Modeling

Modeling the spatial relationship of the reservoir attribute is one of the most important tasks in constructing reservoir characterization using geostatistical methodology. In the last quarterly report, new technique is proposed in modeling the horizontal variogram. This technique is now being implemented as part of the Cosimulation program.

The work done during this quarterly report is concentrated into the development of the user interface of the variogram modeling that includes the modeling of horizontal variogram. Using this interface the user is able to perform the calculation of conditioning data variogram and subsequently modeling it interactively. The final variogram model selected by the user is automatically input into the parameter file of the Cosimulation program.

In addition to the development of the user interface, a modification in calculating the indicator variogram is proposed. This modification is made to correct the way in which the data-pair are selected. The effect of this modification is to reduce the number of data-pair which in turn will increase the sill of the variogram.

2.1 Variogram Modeling Interface

The interface built for variogram modeling is intended as part of the Cosimulation program, i.e., COSIM, described in the last annual report. This facility can be accessed from the Pre-Simulation menu where 2 sub-menus are available, Data Variogram menu and Model Variogram menu. Using Data Variogram menu the user is allowed to first calculate the variogram from the raw data before modeling it. On the other hand, the Model Variogram menu is only available for modeling of previously calculated variogram.

The interface is built in such a way that the user is guided to follow certain procedure which is considered to be correct and efficient in performing variogram modeling. For the case where the user selects to perform Data Variogram analysis, the procedure is as follows :

1. Calculate the isotropic variogram.
The purpose of this step is to find the best lag, lag tolerance, and bandwidth for the set of data of interest. This can be achieved by the fact that using the isotropic assumption, the program can fix several parameters related to the direction parameters, such as azimuth, dip, and their associated tolerances, and let the user concentrate with the lag parameters. During this step the user is allowed to make sensitivity study of the lag parameters until a good variogram structure (if there is any) is obtained.
2. Calculate the anisotropic variogram.
Once lag parameters are fixed, the user can perform the anisotropy study where variograms at several directions are calculated. The program uses a set of default directions, i.e., 0, 45, 90, and 135 degrees. Once calculated, the user

can see all 4 variograms on the screen. To see other directions, the user can change one or more directions (up to 4) and recalculate the variograms.

3. Select variogram with maximum continuity direction.

Based on the anisotropic variograms, the user should be able to decide the direction of maximum continuity and inform it to the program by pressing the mouse button on the appropriate variogram plot. Subsequently, the program will assume that the minimum continuity will be perpendicular to the maximum direction and shows the two variograms (maximum and minimum directions) on the screen.

4. Modeling the variogram.

Using the mouse, user is able to draw variogram model on the two variograms (maximum and minimum continuity) that he/she thinks fit the data best. The program provides the feed back of the model equation selected by the user (the implementation is still underway). Internally, the program also ensures that the model selected satisfies the condition of positive definiteness which is required for kriging process. Figure 2 presents the example of variogram modeling described in this step.

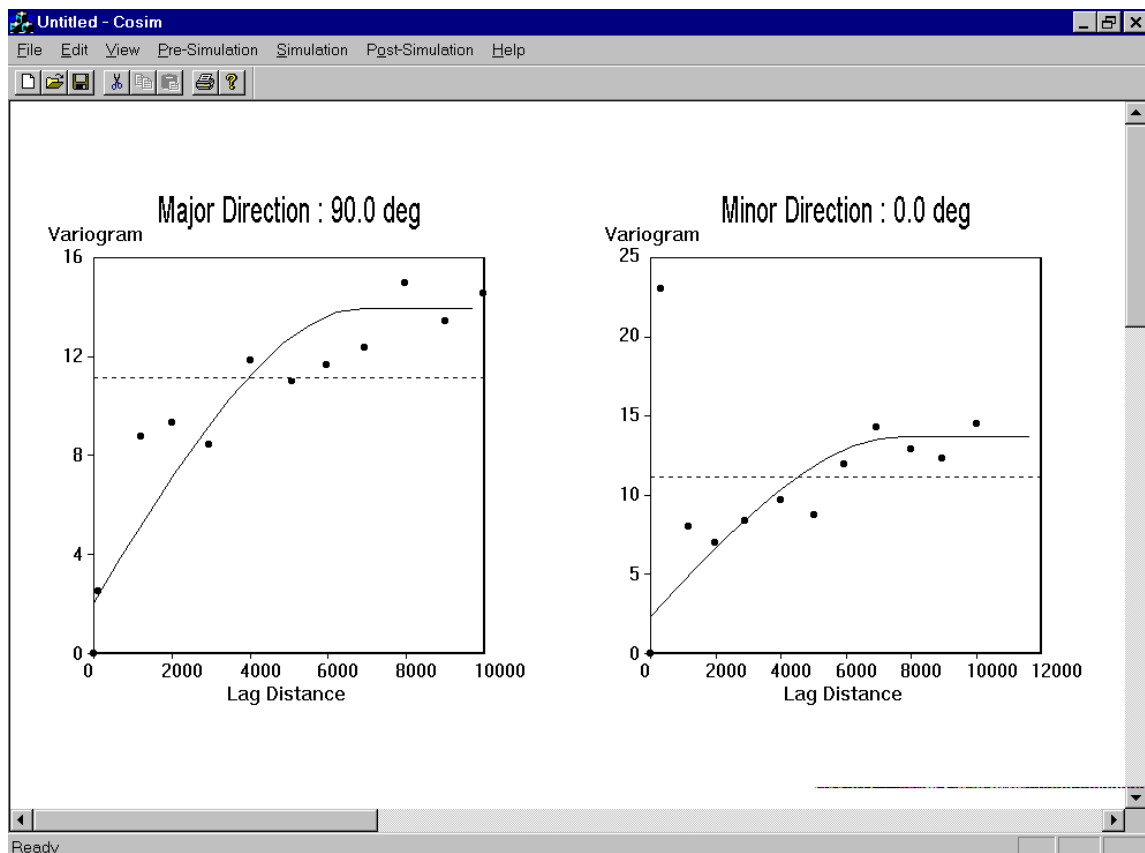


Figure 2: Interactive Variogram Modeling using COSIM program

For each variogram model, the user is allowed to use a nugget effect and 4 combinations of variogram structures, i.e., spherical, exponential, and gaussian. In the example presented in Figure 2, only nugget effect and spherical structure is shown.

2.2 Indicator Variogram Modification

Indicator variogram is variogram calculated using indicator type variable. Typically, indicator variable is used to simulate the geological description where the presence of a geological unit, e.g. facies, at a particular location is assigned as a value of 1 and the absence of that unit as a value of 0. For example, consider well data that consists of the following facies as given in Table 1.

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 1: Facies at well location and its indicator variable

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 the following equation :

$$g(h) = \frac{1}{2N(h)} \sum_{i=1}^{i=N(h)} (x_i - x_{i+h})^2 \quad \text{Eq. (1)}$$

where :

$g(h)$ = variogram at lag distance h

$N(h)$ = number of pair that satisfy the lag distance h

x_i = value of variable at location i

x_{i+h} = value of variable at location i + h

Applying Eq. (1) with h = 1 into data of Table-Asnul 1, we obtain :

1. For Unit A :

$$\begin{aligned}
g_A(1) &= \frac{1}{2 \times 7} [(1-1)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2] \\
&= \frac{1}{14}
\end{aligned}$$

2. For Unit B :

$$\begin{aligned}
g_B(1) &= \frac{1}{2 \times 7} [(0-0)^2 + (0-1)^2 + (1-1)^2 + (1-1)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2] \\
&= \frac{2}{14}
\end{aligned}$$

3. For Unit C :

$$\begin{aligned}
g_C(1) &= \frac{1}{2 \times 7} [(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (1-1)^2 + (1-1)^2] \\
&= \frac{1}{14}
\end{aligned}$$

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 unit is correlated very well. In reality, this unit does not even exists at 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. $g_A(1) = \frac{1}{2 \times 2} [(1-1)^2 + (1-0)^2] = \frac{1}{4}$
2. $g_B(1) = \frac{1}{2 \times 4} [(0-1)^2 + (1-1)^2 + (1-1)^2 + (1-0)^2] = \frac{1}{8}$
3. $g_C(1) = \frac{1}{2 \times 3} [(0-1)^2 + (1-1)^2 + (1-1)^2] = \frac{1}{6}$

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 Asnul-2 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. Due to the differences for each pair, the overall structure of the variogram may also be affected. Further study about this effect is still required.

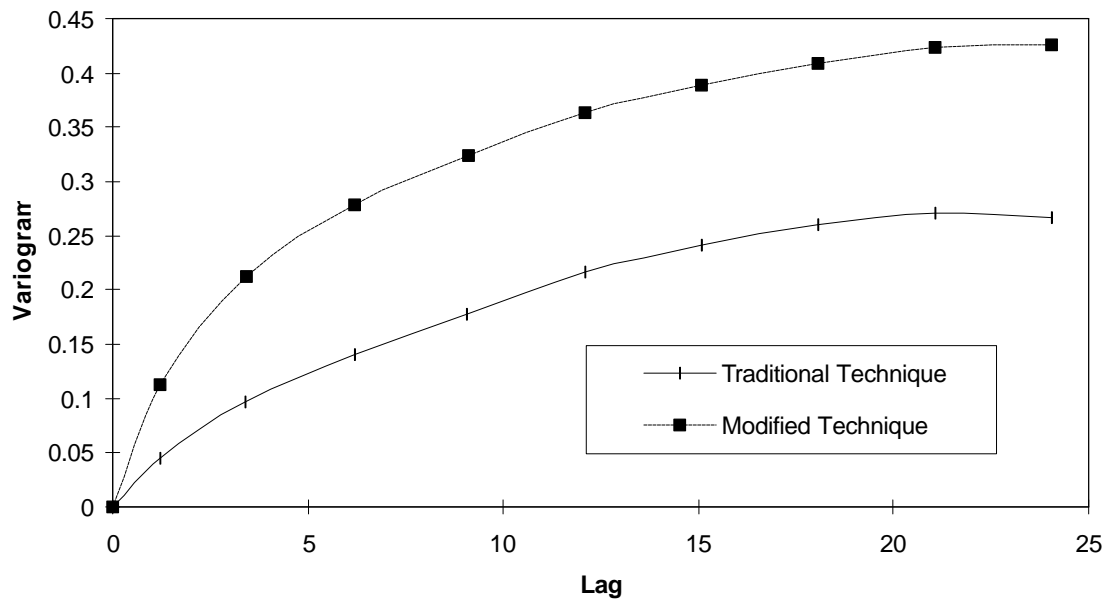


Figure 3: Indicator Variogram Comparison between Traditional and Modified Techniques.

3. Testing the Geological System Components

3.1 Improving the Correlation Results

As the integration effort continues, we have attempted to obtain better correlation results to pass to the geostatistical algorithms for simulation. One of the major changes was to more fully utilize the reference well as having stable characteristics. This improved the correlation results dramatically. 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 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 a high degree of accuracy.

The two tables below present the results of a pairwise correlation between wells in the Glenpool field. Two wells (labeled 11-86 and 7-87) are considered as reference wells with the characteristics given by the geologist. There is one table for the correlations with well 11-86 and one table for the correlations with well 7-87. 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 cuts found by the system for that well log. The third column indicates the number of

unnecessary cuts within that total. The fourth column indicates the number of missed cuts. Both unnecessary and missed cuts can alter the correlation. The fifth column indicated that number of intervals determined by the system. The final column indicates how many of those intervals are incorrect.

In Table 2, the well logs are correlated with well 7-87. This well has 8 cuts and 7 intervals as indicated by the expert geologist. Merging of the intervals in 7-87 only occurs in the case that less cuts appear in the well to which it is being correlated. Through the analysis, a pattern for the incorrect interval delineation emerged. In many cases the well log information for the well being correlated with well 7-87 was not complete, resulting in a “shorter” log. Our correlation algorithm is strongly directed to finding all 7 intervals as given in the reference well. Therefore, the system attempts to correlate the ending curves of the reference wells to the ending curves of the other well, resulting in intervals where there should not be any. We have placed an asterisk (*) by the well name for those wells in which this mistake occurs. We are now determining what the reason is behind the lack of data and if, in fact, the data is valid we need to add intelligence to the system to notice the “shorter” logs and resist correlating for the sake of finding the key number of intervals. However, with the existing algorithms, we are achieving about a 70% correct correlation rate, which we are sure we can improve upon in the * wells.

Well	Cuts	Unn. Cuts	Mis. Cuts	Intervals	Inc. Ints.
7-109	9	1	1	7	1
7-110	6	0	1	6	1
7-89*	7	0	1	6	2
d-9*	9	1	0	7	1
e-9*	8	0	0	7	2
e-8	9	1	0	6	0
f-8*	5	0	2	4	2
6-83*	9	0	0	7	1
g-8*	8	0	1	6	2
g-85	10	0	1	6	1
7-113	11	2	0	7	2
7-107*	9	1	0	7	2
7-99*	10	3	0	7	2
7-100	12	1	0	7	1
7-103*	9	0	0	7	2
6-85	12	1	1	6	2
6-84*	8	0	0	7	2
7-97*	10	2	0	7	2

Table 2 Well Correlations with Well 7-81

Table 3 shows the analysis results for the well correlations with reference well 11-86. Well 11-86 has 8 cuts and 7 intervals as given by the expert geologist. A similar problem with the “shorter” well logs arose in these correlations as well. An asterisk (*)

marks those wells whose incorrect intervals are due to this problem. Overall, the results for well 11-86 correlations were better than those for well 7-87.

Well	Cuts	Unn. Cuts	Mis. Cuts	Intervals	Inc. Ints.
h-11*	10	1	1	6	1
h-13x*	12	0	0	7	1
h-12	12	1	0	7	0
h-10x*	8	1	2	7	2
h-85	9	1	0	7	1
11-82	11	0	0	7	0
11-84	9	1	1	6	0
11-85	10	1	0	5	1
k-10	10	0	1	6	0
m-6*	9	1	1	7	2
11-87	10	0	1	6	2
k-12	15	0	0	7	2
m-12*	11	0	0	7	3

Table 3: Well Correlations with Well 11-86

3.2 Future Work

In addition to increasing the intelligence of the system, we must be concerned with conflict resolution. Some wells can be correlated with more than one reference well. This may lead to different results for the DGI. Therefore a problem is how to resolve the conflict. One approach is to correlate the well again with itself, using the DGI information obtained from the two previous correlations. This would involve re-identification of the facies and may not resolve the problem completely since the correlation may lead to different DGI again.

Another problem we are having is with the access database. The well information is currently hardcoded into the database. We now have to make the system more flexible for user input. In order to do this, the user will have to input the well log information directly into the access database with tables made to our specifications for the system to work correctly. We are investigating how this can be done within a user interface.

4. Integrating Subsystems

To facilitate the construction of the reservoir characterization system, we decomposed the system into smaller parts as described in Section 2. This decomposition allowed us to apply multiple artificial intelligence techniques, such as expert systems and neural networks, as well as utilize many numerical techniques. Because of the interdisciplinary nature of the project multiple languages were used in the development,

along with multiple platforms. The language and platform decisions were also made to expedite development and testing. Because the target architecture is the PC, it was necessary to port all code from workstations for PCs.

We had chosen C++ as the final language in which all the systems would be encoded. However, some of the very computationally intensive code was written in FORTRAN. This gave us three choices in integration: (1) manually convert all of the code to C++, (2) use a conversion tool to convert all of the code to C++, or (3) find software that claims to integrate C++ and FORTRAN without any conversion. Manual conversion would be extremely difficult because there are thousands of lines of code in FORTRAN. Automatic conversion is available, but it produces poor quality and hard to read code, some of which will not compile. The final option seemed most feasible because of Microsoft's claims that their Visual C++ and PowerStation FORTRAN were fully integratable. We discuss the integration of the whole system using this approach in the next section.

Independent component systems were constructed for each part of the generation of the reservoir characterization software. Each system was either written in FORTRAN or C++. The objective of this work was to develop a fast and effective method to integrate the various subsystems. More over, the task was further complicated by the incorporation of all codes in the Microsoft Visual C++ Integrated Development Environment (IDE) generated main application. The concept of using Microsoft Foundation Class (MFC) Libraries to code a driver or main application has now been established. However a few problems remain these will be elaborated later in the report.

4.1 The Aiinteg.exe (Driver Routine)

As of now, five different methods for integration have been experimented and utilized. Out of the five, the three most important methods are enumerated below:

4.2 CreateProcess Command :

This is a Windows Application Programming Interface (API) command by which applications of individual subsystems can be called and executed. This syntax being specific for console type applications had little use in the driver code. Another point worth mentioning is the cumbersome and bulky nature of the final compiled executable.

The *CreateProcess* was utilized in the initial steps of code generation but as the command required a number of system specific arguments and complicated pointer manipulations its use in the driver routine was limited.

4.3 _Spawnl Command :

This command is a standard call command for applications supported by Win32. *_Spawnl* performs essentially the same function as the *CreateProcess* command but its

implementation is much more straight forward. Since the command allows ‘off-the-fly’ generation of external application calls with very little memory leakage this command was preferred over *CreateProcess*. However *_Spawnl* like the *CreateProcess* requires arguments in its call process and is therefore limited to applications (*.exe files) which pass arguments at run-time. Even with this limitation *-Spawnl*, works more efficiently than the *CreateProcess* application calling command.

4.4 *System* Command :

Due to the limitations of the *_Spawnl* command an external application calling *System* command has been incorporated in the *Aiinteg.exe* driver routine. This it was observed, is the simplest and the most portable way of linking executables in our application. The *System* command works well in almost all classes generated by the Microsoft Developer Studio (MSDEV), Application Wizard. The *System* call also works well with the Microsoft Foundation Class (MFC) based derived classes. If the system call is made to a MFC based application developed in the MSDEV, the *System* command has no problem in running all existing applications simultaneously. However a probable problem of memory leakage during the execution of the console based application causes only one of the application to run at one time, thus hampering interactions with the rest of the applications. This phenomena is observed only during some calls, the solution to this problem is being sought.

4.5 Approach

Having developed an effective method of calling all applications the final step was to integrate all the subsystems together. The main task in the integration is to configure outputs generated from one program (i.e., COSIM, the Co-simulation program) as inputs to another application (i.e., ANSIM, the Annular simulation code). Also, as all the interactions are to be done in windows, separate dialog boxes and views are being constructed. The main objective of this effort is to provide a more ‘user-friendly’ approach to data input. This it is felt would allow the user to spend less time providing input files and more time on running the applications and gaining useful results. The first attempt to provide input dialog boxes and views is being done for the well testing application. ANSIM will be the next application for this input data configuration process.