APPLICATION OF ARTIFICIAL INTELLIGENCE TO RESERVOIR CHARACTERIZATION: AN INTERDISCIPLINARY APPROACH

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

Submitted by DOE/BC/14894--13

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:	April 1 - June 30, 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

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 has begun.

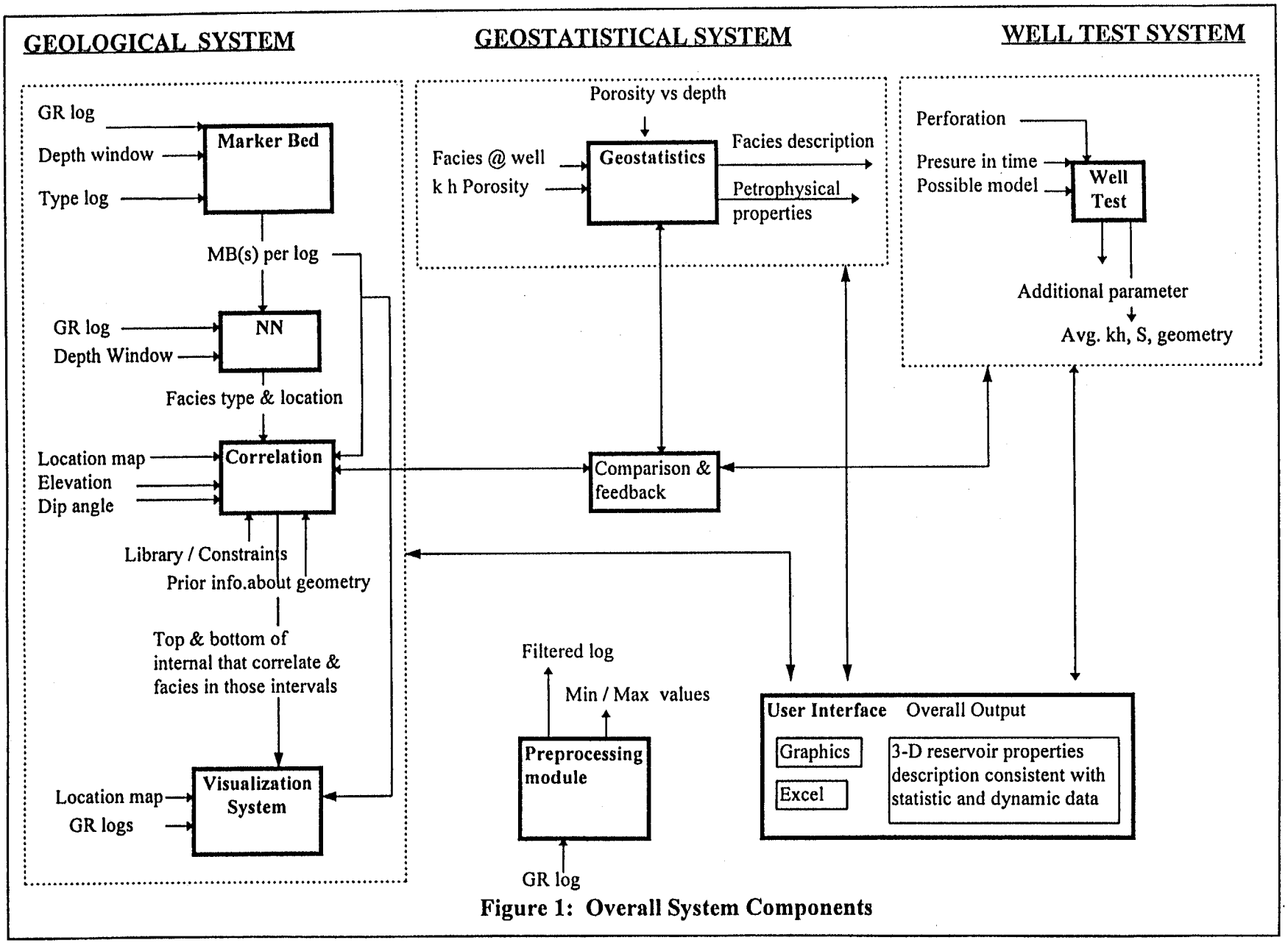


Figure 1: Overall System Components

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

2.1 Overview

Modification of the code to enable full 3-D characterization capability was undertaken during this period. Note, however, that the upscaling techniques were not modified from 2-D to 3-D. Discussions and work on integrating this module into the overall framework of the project were also started.

2.2 Code Modification

Although the flow simulation code was initially designed for 3-D, it was implemented as a 2-D model, primarily because the upscaling technique used was for a 2-D system. Thus all modifications of the code in the past were made within a 2-D setting. It was necessary therefore to undertake a modification of the code so that 3-D models may be considered. The major effort concerned the re-design of the code to initialize, modify and update the coefficient matrix which must be inverted for the solution of the flow equations. A major consideration in this was the fact that, whereas for a 2-D system the matrix is symmetric (for no-flow boundary conditions), for a 3-D system this matrix is not. This meant that an assymetric solver had to be used and also that the entire matrix had to be stored (instead of using only the upper triangular or lower triangular part, as is possible with a symmetric system). The end result was that the order of the matrix increased from the number of upscaled gridblocks in the system, *nxyz1*, to *nxyz1+nwell(nz+1)*, where *nwell* is the number of wells in the system and *nz* is the number of z-direction gridblocks in the 3-D Cartersian grid system being used. These extra equations are required for the Peaceman[1] corrections used to calculate the flowing bottomhole pressures from the gridblock pressures (*nwell*nz* equations) and, for the gridblocks penetrated by each well, to equate the sum of the rates from those blocks to the overall well rate (*nwell* equations). Additionally it was assumed that there was no cross-flow between the layers. This, which is necessary because of the upscaling technique used, is a fair approximation since in reality the vertical permeabilities are typically an order of magnitude smaller than the horizontal values, hence cross-flow is usually not appreciable. Also, while the simulated annealing (SA) code was general enough to handle 3-D, some changes had to be made to the code linking the SA code to the flow simulation code. These changes were completed and the system successfully tested using a 3-D system as well as a quality-control check using a 2-D system for which the results were already known.

2.3 Code Integration

Some work was initiated on the integration of the code into the overall characterization package. This code will be a module which outputs the end-result realization for the permeability distribution. The inputs will be the dynamic (rate/pressure well) data and the porosity-permeability correlation. The porosity distribution will be considered as the 'truth case' distribution from which the initial permeability distribution is generated and then perturbed to match the dynamic information. Simple arithmetic averaging for the porosity data will be used and modified geometric averaging in 2-D for the permeability data for the flow simulation part of the SA algorithm. Note that the upscaling remains in 2-D such that a fine scale grid, with *nx* gridblocks in the *x*-direction, *ny* in the *y*-direction and *nz* in the *z*-direction, is upscaled to a coarse scale grid with *nx1* gridblocks in the *x*-direction, *ny1* in the *y*-direction and *nz* in the *z*-direction, where *nx1* and *ny1* are respectively *nx/factor* and *ny/factor*, where *factor* is the upscaling factor used.

3. Integrated Lithofacies and Petrophysical Properties Simulation

3.1 Overview

This report presents the progress report 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 progress achieved during this reporting period is the development of the program interface. The interface is built using the Microsoft Visual C++ 4.0. Using this interface, the user is able to built the parameter file, run the simulation, and see the 2D cross sectional image of simulation result, in one single PC-based program. Future version of this interface will include the variogram analysis and other statistics tools to help the user evaluate the data as well as the simulation result.

In addition to building the program interface, some effort has also been spent to modify the program while transforming the program from the UNIX-based system to the PC-based system. After the modifications, the program has been tested successfully to run with 1.5 million grid blocks. Theoretically, this number is limited only by the computer memory.

3.2 Co-Simulation Program

Background theory used in developing the simulation technique was presented in the previous quarterly report. This report is concentrated in presenting the overview of the program interface.

3.2.1 Main Window

As in any Windows program, the first window that appears when the user executes the program is called the main window. For simplicity, the first version of this program is built using the Single Document Interface (SDI) technique. This means that the program has only one window and only one document can be loaded at a time. Even though this technique restricts the number of windows that can be opened at one time, its performance will not be significantly affected. The future version of this program is planned to be developed using the Multiple Document Interface (MDI) technique.

The appearance of the Cosim's Main Window is shown in Figure 2. Some of the menus are fully operational but some are not. The first three and the last menus, i.e., File, Edit, View, and Help, are standard menus from Microsoft Foundation Class (MFC) program. The fourth, fifth, and sixth menu, i.e., Pre-Simulation, Simulation, and Post-Simulation, are Cosim's specific menus. From its name, its function is self explanatory.

The view of the window is built using the Scroll View window that enables the user to see some portions of the image which is out of the current view. All tool bars are standards MFC tool bars. No specific tool bar are defined at the moment.

The File menu consists of several sub-menus that controls the loading and unloading of the document. This document is saved in the binary format. At this time, there is no default extension defined for the Cosim's file. Therefore, the user has to supply the correct file extension. The extension "*.csm" is recommended to be used. The Print command, which is part of File menu, is fully operational, but it has not been configured to give the "good" size. Some work is needed to obtain good hard copy.

The only command that is fully operational from the Edit menu is the Clear Screen command. This command is used to clear the screen. When the program is first executed, blank white screen appears. After the user creates the image on the screen, the Clear Screen command or the corresponding toolbar, i.e., the scissors, can be used to clear the entire screen to obtain the blank white screen.

The View menu consists of two fully operational commands; the View Tool Bar and the View Status Bar. These commands can be used to enable or disable the tool bar and/or the status bar. Disabling these tool bars will generate bigger view area and vice versa.

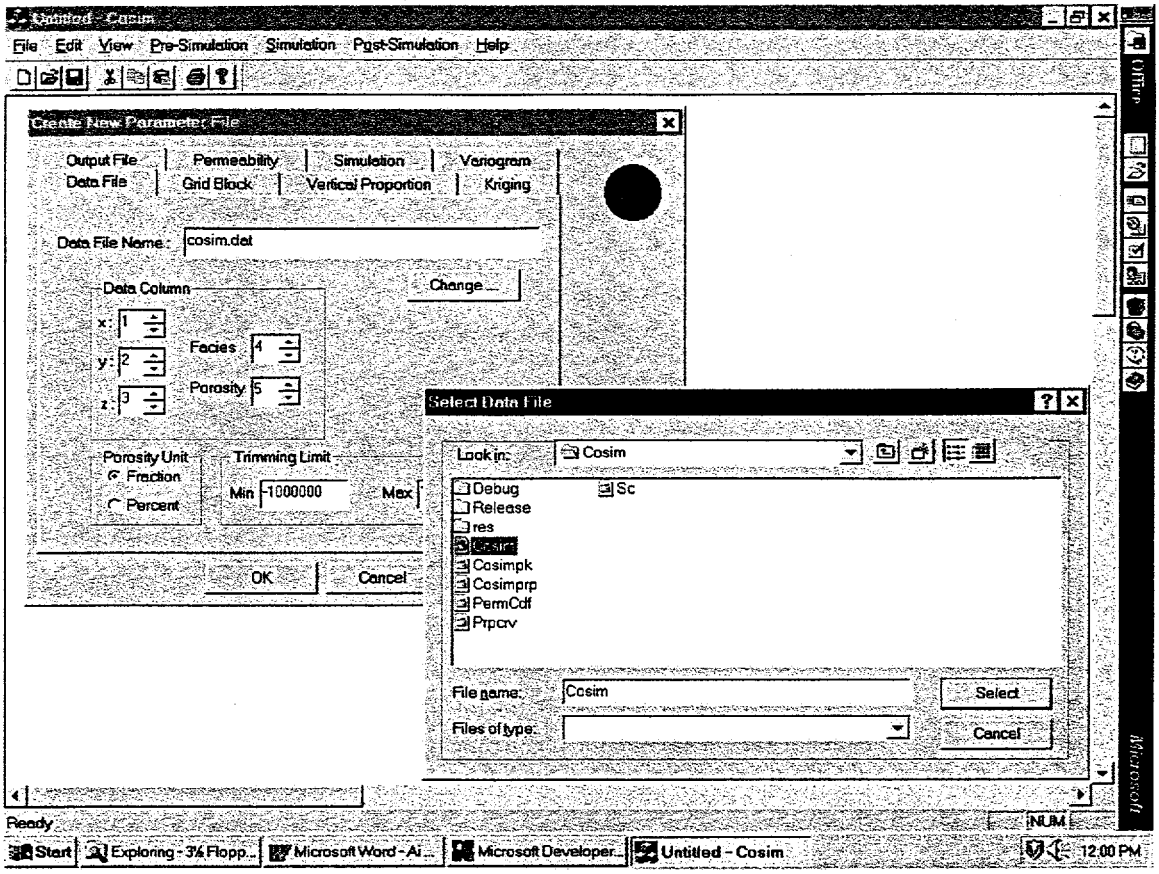


Figure 2: Dialog for building parameter file

3.2.2 Pre-Simulation

The Pre-Simulation menu consists of two sections. The first section is related to general geostatitstics tool such as the variogram analysis, the statistical analysis of the data, and the normal transformation which is used for Gaussian type simulation. The second section is specifically related to the Cosim program, i.e., the building of parameter file. This parameter file is the main input for the Cosim calculation. The progress that has been achieved at this stage is the development of second section. This is due to the priority that was set to obtain a single simulation program that runs simultaneously from generating parameter file to viewing the result. It is true that the incorporation of variogram analysis will make the program much more powerful. But considering the amount of time that is required to build this interface, variogram analysis is considered a secondary priority.

In building the parameter file, the user can either open the previously generated file or create the new one. The dialog is designed using the property sheet where there are 8 properties defined for the whole parameters. Effort has been made to make the use of this facility as easy as possible. Figure 2 presents the dialog that guides the user in building the parameter file. This figure also shows the standard Windows Open/Save File facility that is used to select a file.

3.2.3 Simulation

The Simulation menu is the menu to use when the user is ready to execute the simulation calculation. The default or the latest parameter file selected by the user is prompted by the program to be used for simulation. The user can change this file as required. At this stage the program does not have the capability of showing the progress of the simulation. Once the simulation is initiated it can not be interupted. An effort is being made to make the program more flexible.

3.2.4 Post-Simulation

The post simulation menu is designed to help the user in evaluating the simulation results. The design includes qualitative evaluation, such as the presentation of the cross section on the screen, as well as quantitative evaluation, such as statistics calculation. The progress achieved so far is for the qualitative evaluation only. To obtain the screen image of a 2D cross section, the program provides the dialog to guide the user in selecting specific cross section that he/she wants to see.

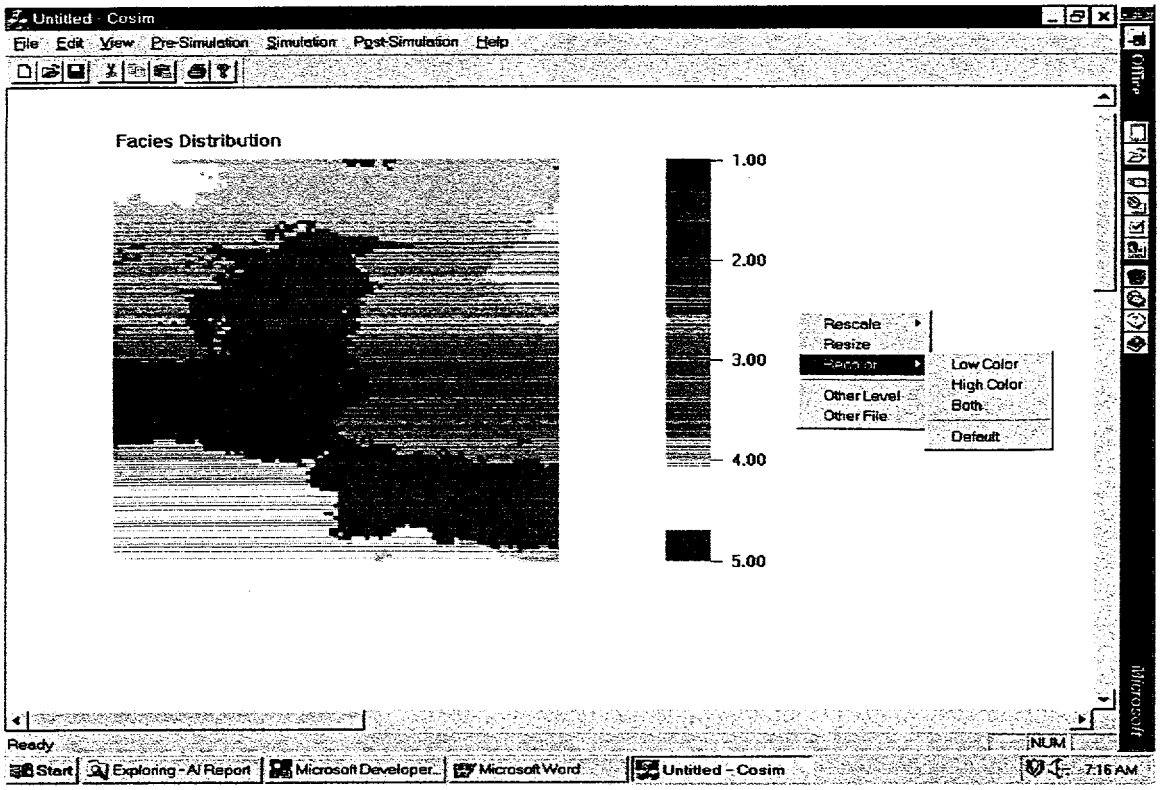
Figure 3 presents the example of rock type cross section which is generated using the simulation result of the Carbonate field data. To give the flexibility to the user in selecting the drawing attributes such as plot scale, color scheme, or grid block size, the program provides the floating menu that can be accessed by clicking the mouse's right button. This floating menu is also shown in Figure 3.

The corresponding porosity and permeability distribution for the rock type distribution shown in Figure 3 is presented in Figure 4. This figure is generated using different color scheme and different grid block size. Comparing these two figures we can

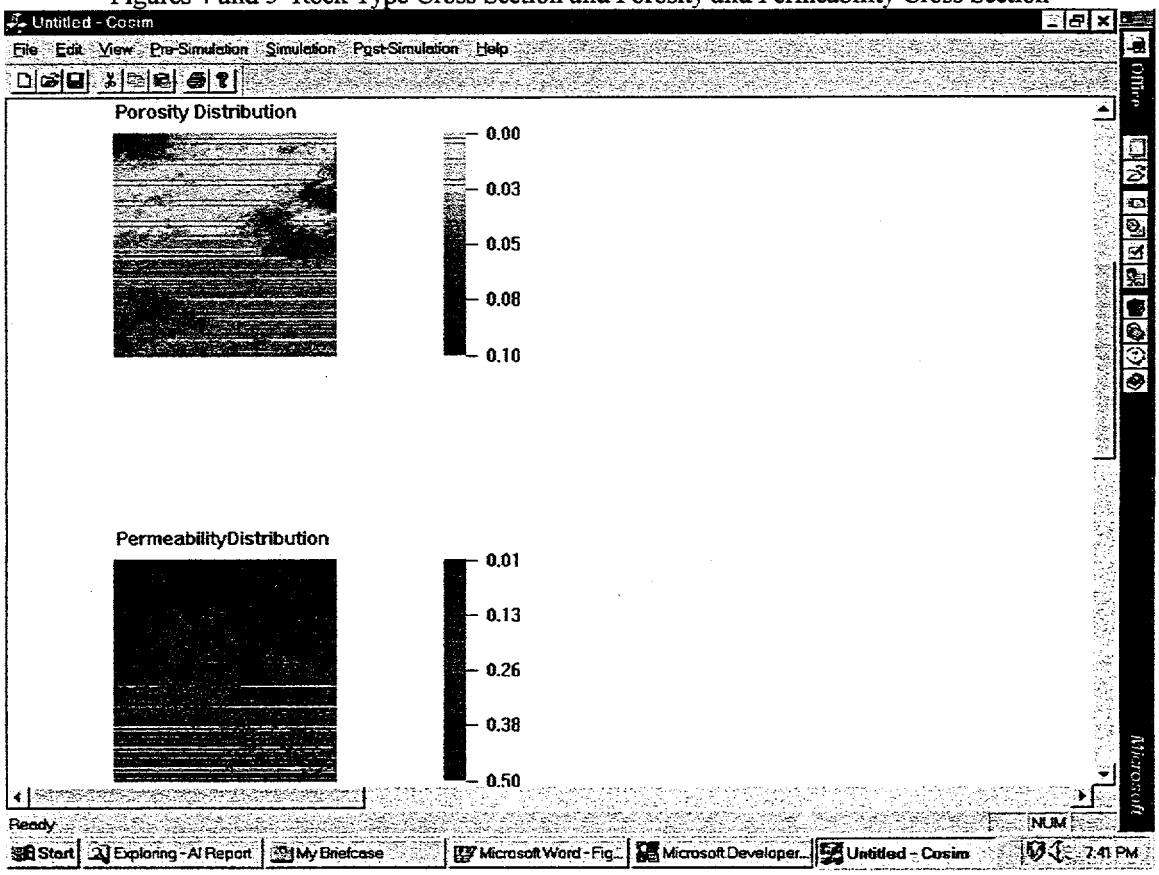
observe that the generated petrophysical properties are consistent with the underlying geological description.

3.3Future Work

The future work that is planned for this program involves the improvement in the interface program as well as the improvement in the simulation technique. The improvement in the simulation technique will be focused for the facies simulation. As mentioned in the previous report, the facies simulation is generated based on the truncated Gaussian simulation, where the future version is planned to be pure indicator simulation.



Figures 4 and 5 Rock Type Cross Section and Porosity and Permeability Cross Section



4. Testing the Geological System Components

We have completed reasonable testing of the subsystems comprising the geological system. This tesing resulted in the fine tuning of the subsystems. The results are promising and we plan to continue the fine-tuning as integration of the subsystems continues and as we build the 3-D model of the information. We begin the next section by providing updates to the subsystem. The bulk of the section presents the testing of the results against two geologoy expert interpretations.

4.1 Updating the Well Log Segmentation Process

Previously, we discussed the filtering approach to detecting the cuts in the gamma ray logs. Figure 6 shows a basic result of the approach. Figure 7 shows the gologist interpretation of the same log showing that the filtering approach for this log detected a high percentage of the correct cuts.

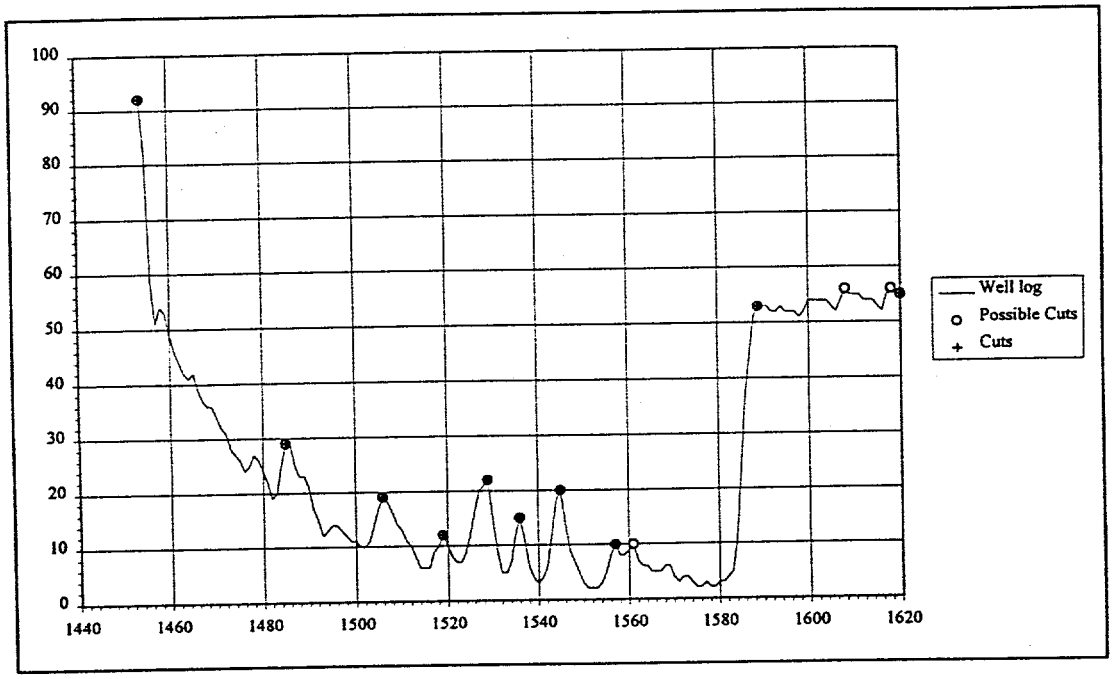


Figure 6: Results obtained from previous filtering approach

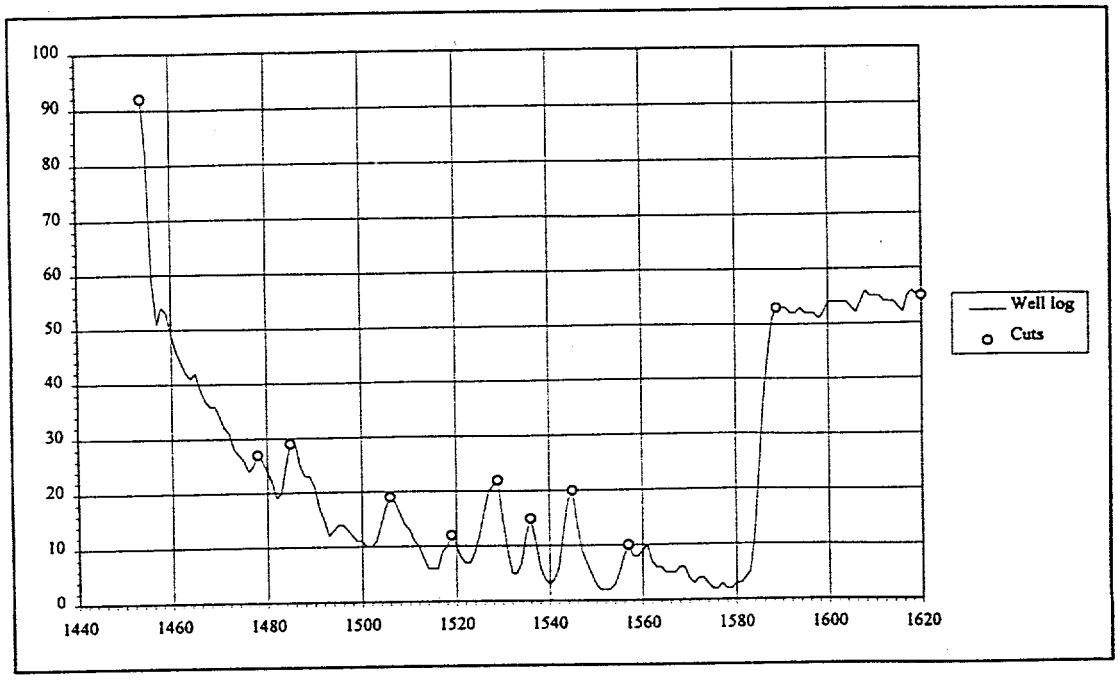


Figure 7: Results obtained from an expert geologist.

With the additional testing, it was determined that the approach used in some logs can leave big gaps without a cut. Therefore, a new rule had to be included in the filtering algorithm. This new rule calculates the shape distance between the cuts. If a distance is longer than a prefixed valued, it is possible that a cut is missed. Therefore, all the rules used to find cuts are applied again over this gap, but using the unfiltered log. In Figure 8 is shown an well log with a big gap between depth 1454 and 1486, thus the new rule determines any cuts could be missed. Figures 6 and 7 show a missing cut at depth 1478. In the well log shows in Figure 9 the new rule was applied and the missing cut was found.

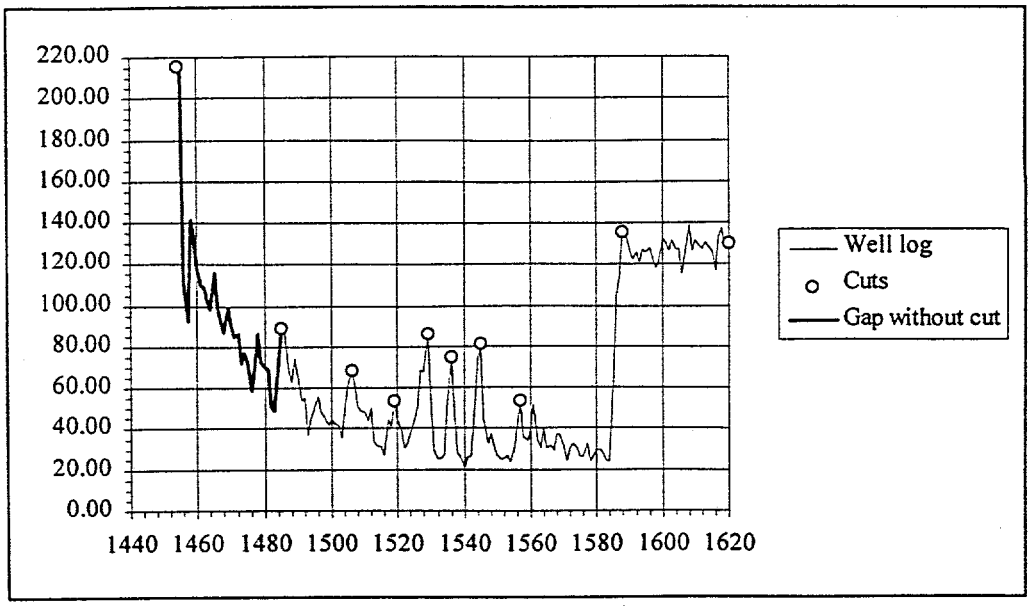


Figure 8: Results obtained without gap detection.

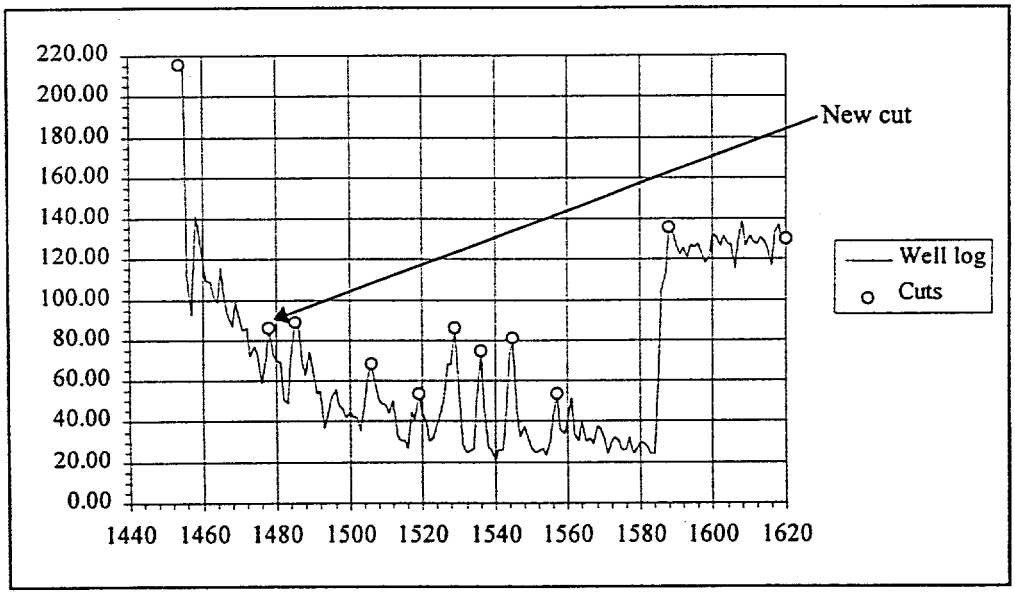


Figure 9: Results obtained with gap detection.

4.2 Updating the Correlation Process

In this section, we describe the expert system that performs the correlation given the matrix values obtained by the previous ranking approach to the correlation of log curves. A rule set, described in the March 1996 report has been used to generate a matrix of compatibilities of zones in wells. In addition, we introduce a visualization module to depict and manipulate the system's correlations.

4.2.1 The Expert System for Correlation

The system for correlating and zones in wells and visualizing them can be considered as three modules each performing distinct tasks. They are the correlation rank matrix module, matrix analysis module and the visualization module as depicted in Figure 10.

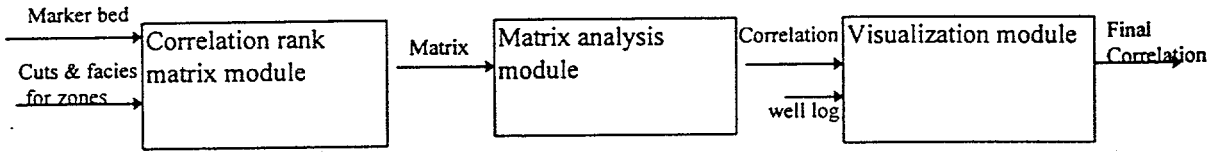


Figure 10 The system model for correlation of wells

Correlation rank matrix module. This module uses a set of rules based on similarities in well log trace shapes, thickness and vertical position of zones. The segmentation of well logs and the log-facies identification by the neural network and depths of identified marker beds will be given as input. The matrix obtained from this module is used as input for the module analyzing the matrix.

Matrix analysis module. Using the matrix obtained from the correlation rank matrix module, this module reports the correlation of the zones. A set of rules are used to analyze the ranks in the matrix to decide which zones correlate and which ones merge. These rules are given below.

1. Find the row and column maximums for each of the rows and columns in the matrix.
2. If a particular entry in the matrix is a row and column maximum, then the zones corresponding to that entry in the matrix are correlated. This is done since if two zones being compared have the highest rank both on the row and column, then there is a obvious correlation between them.
3. If there is a row/column maximum and if the respective column/row corresponding to the maximums is not correlated then the zones are correlated. This allows the freedom of correlating rows and columns in which there is either a row or column maximum but the corresponding entry may not be the maximum by a couple of ranks.

4. For each of the remaining rows/columns that have not been correlated, going down from the first maximum to each of the successive lower maximums, it is checked to see if any of the adjacent rows or columns have been correlated. If they have then the corresponding zones are merged and the correlation extended to cover that zone. Since the neural network comes up a varying number of cuts for the two wells being correlated, there is a likelihood that zones merge before they can be correlated.
5. A rule which checks if an exterior row/column has been correlated and if it has does not permit that correlation with an interior row/column. This is done to limit the number of cross-overs at the end, and to allow the fact that there may be some zones that won't be correlated at the very end probably because of one of the logs being longer than the other.
6. Detection and removal of cross-over: A cross-over occurs when the correlation of the zones intersect.

The above steps are repeated until all the zones have been correlated with no cross-over. Initially all the cross-overs were detected and removed and only then were the zones recorrelated. This led to the problem that several zones would remain uncorrelated. To remove this problem, cross-overs were detected one at a time and zones recorrelated after every removal. Another problem was that the system was picking certain cross-overs which were in fact zones merging with others. To handle this problem, rules were added to check for merging before testing for cross-overs. Finally, there may still be zones in which have not been correlated which probably occurred because of a shorter log data which will have to be resolved by the user during visualization of the correlation.

4.2.2 Visualization module.

To allow the user the freedom to modify the correlation obtained by the system, a 2-dimensional view of the gamma-ray logs of two wells which are being correlated is plotted against depth and is displayed in a window. In order to do that, a window was created and the log values were rescaled so that they fit the window dimensions. This is done by finding the minimum depth and gamma ray values for the log, assigning them minimum X and Y axis values for the window and for the remaining values of the log the window values are calculated relative to the minimum values. The correlation of the zones is displayed by drawing lines between the corresponding depths again taking into account the rescaling. The user can use the mouse to click on the end-points of the line to delete and add lines where they think there should be a correlation. The depth values of the logs are displayed along the y-axis.

A problem was that to delete inappropriate correlation, the user had to click exactly on the pixel corresponding to the end point of the line for the system to recognize the line chosen. Clicking on a pixel in the vicinity was not sufficient. To provide a better interface, a rectangular region was defined around the end points and drawn. So the user

now has to click anywhere inside that rectangle for that correlation to be deleted. A print option is provided in the form of a button which allows the user to view the hard copy as well. Also, once the user has decided on a correlation, the data is written back to a file with the changes, since some of the correlation between the zones may have been deleted or modified.

4.3 Comparison of the system with geology experts

4.3.1 Cut comparison

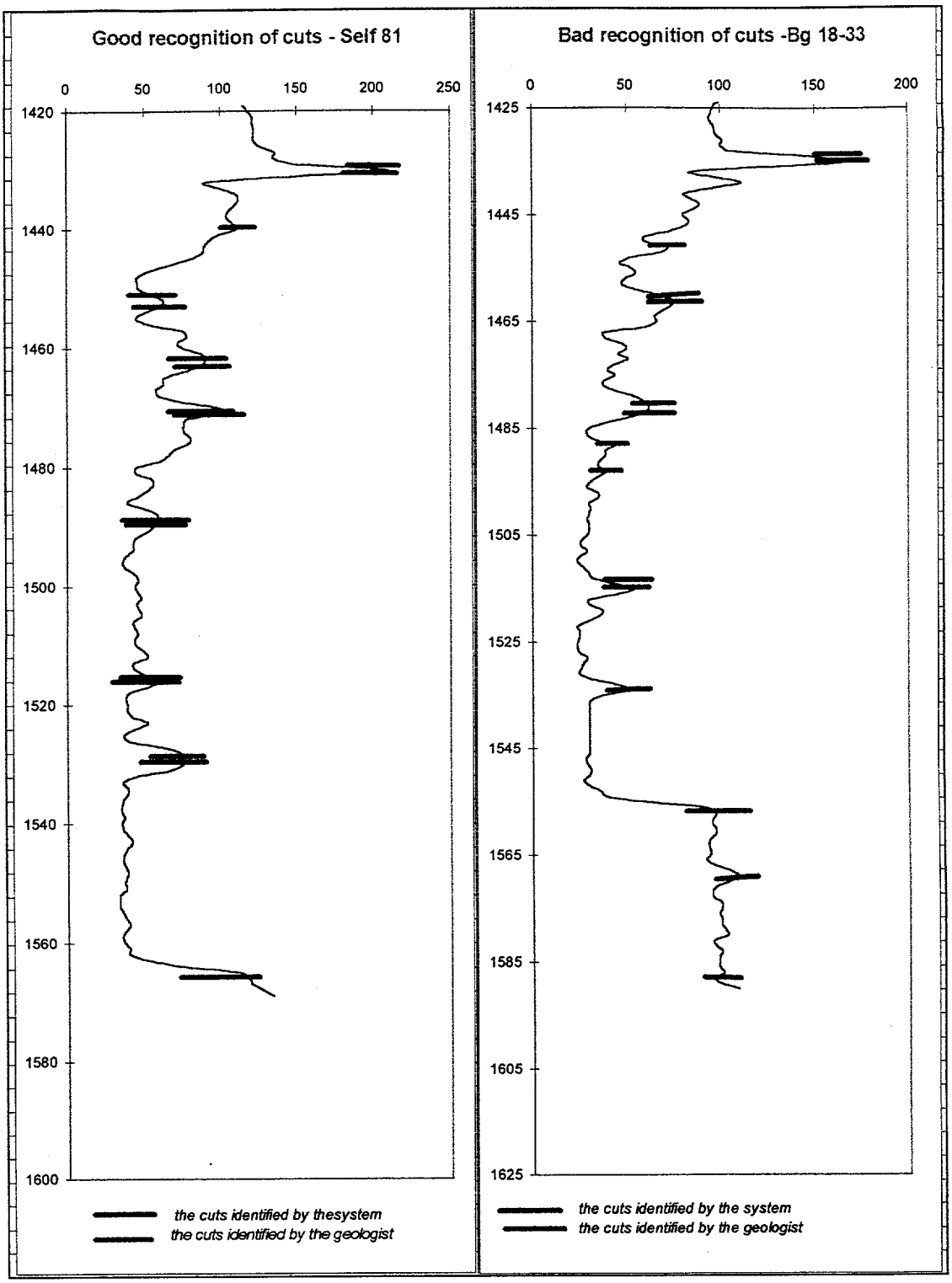
The cut recognition of the system has been compared with the cuts identified by the geologists. They are compared on the following basis: (1) the number of cuts identified by the system that correspond exactly with the cuts identified by the geologist is counted, (2) the number of cuts that the geologist has marked but which the system did not identify is noted, (3) the number of additional cuts added by the system is also noted as it will affect the correlation of well logs, and (4) the percentage is calculated based on the number of cuts identified correctly by the system to the total number of cuts identified by the geologist. The results appear in Table **1**.

Well name	Compared to Geologist 1				Compared to geologist 2			
	correct	Missed	Added	%	Correct	Missed	Added	%
Well 11-75	8	5	1	62%	8	3	1	73%
Well 11-86	8	4	2	67%	8	3	2	73%
Well 11-89	12	4	3	75%	10	4	3	71%
Well 679	13	3	1	81%	10	3	4	77%
Well 672	9	2	0	82%	9	4	0	69%
Well 1051	10	5	2	67%	9	4	3	69%
Self 82	8	1	3	89%	9	1	2	90%
Self 78	8	6	0	57%	7	2	1	78%
Bg18-32	8	6	1	57%	6	3	3	67%
Bg18-33	5	4	3	55%	6	1	2	86%
Self 81	8	2	0	80%	7	1	1	88%
Well 11-88	12	4	1	75%	9	3	4	75%

Table 1: The comparison of cut recognition

On average, the number of cuts correctly identified is about 74%. the number of cuts missed is approximately 26% and the number of cuts added by the system is about 14%. The best recognition of cuts is for Self 81 with 1 missed cut and 1 added. The worst is Bg 18-33 with 4 cuts missed out of 9. The best and worst recognition of cuts based on the number missed is shown in Figure 11.

Figure 11: Comparison of cuts



4.3.2 Comparison of the facies

To analyze the facies recognition of the neural network system, the facies identified by the network was compared to the facies identified by the geologists considering the zones identified by the system, as this would allow a direct comparison. The results are presented in Table 2.

Well name	Compared to Geologist 1		Compared to geologist 2	
	Correct facies	Percentage	Correct facies	Percentage
Well 11-75	4/7	57%	3/7	43%
Well 11-86	3/8	38%	6/8	75%
Well 11-89	5/12	41%	8/12	67%
Well 679	9/13	69%	4/12	33%
Well 672	3/8	38%	4/8	50%
Well 1051	7/12	58%	7/10	70%
Self 82	7/9	78%	7/9	78%
Self 78	4/7	57%	5/6	83%
Bg18-32	3/7	43%	3/7	43%
Bg18-33	2/6	33%	3/6	50%
Self 81	5/7	71%	7/7	100%
Well 11-88	6/12	50%	6/8	75%

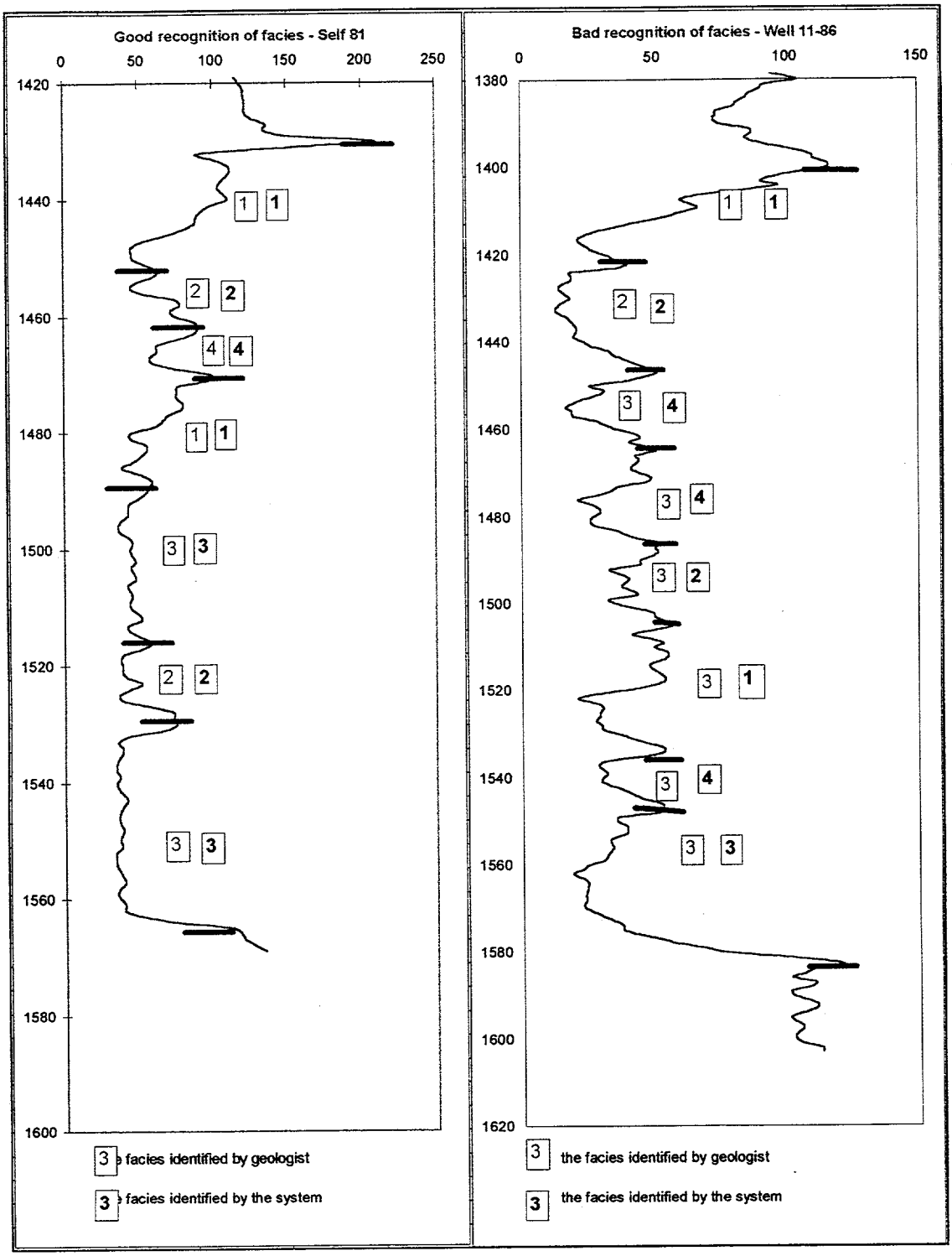
Table 2: The facies recognition comparison

As with the cut comparison, the best and worst facies recognition is shown in Figure 12. The average recognition of the facies is about 58%.

4.3.3 Correlation Comparison

To evaluate the performance of the system's correlation, the system's correlation was presented to a geologist who classified the correlation of zones under 3 categories: (1) A *good* correlation: A correlation which agrees with the expert's correlation of the wells, (2) An *acceptable* correlation: A correlation which does not exactly match with the expert but which is a reasonable correlation, (3) A *bad* correlation: A correlation which

Figure 12: Comparison of facies



cannot be justified by the expert. Based on the above 3 categories, the system's correlation was analyzed and the results are presented in Table 3.

Wells being correlated	Geologist1				Geologist 2			
	Total	Good	Acceptable	Bad	Total	Good	Acceptable	Bad
1175 vs. 1188	7	2	3	2	7	6	0	1
1186 vs. 1189	5	1	2	2	5	3	2	0
1175 vs. 1186	5	4	1	0	5	5	0	0
self 81 vs. self 78	4	3	0	1	4	2	2	0
self 81 vs. self 82	6	2	2	2	6	5	1	0
672 vs. 1051	5	0	2	3	5	5	0	0
1833 vs. 1832	6	2	0	4	6	5	0	1
679 vs. 672	4	0	2	2	4	3	0	1
Total	42	14	12	16	42	34	5	3

Table 3: Comparison of Correlation

On an average the system comes up with 78% of good and acceptable correlation. One problem that needs to be modified is the merging of zones. Since the number of zones for one of the wells may be considerably greater than the other, the zones have to be merged while correlating. Incorrect merging is the cause of some problems when the correlation is bad. Also with the input to the system is from other modules which perform automated recognition of marker bed, cuts and facies which may not be the most suitable choices, the correlation may be bad because of such inconsistencies. Figures **13** and **14** present good and bad correlations performed by the expert system

Figure 13: A good correlation by the system: Wells 11-75 and 11-86

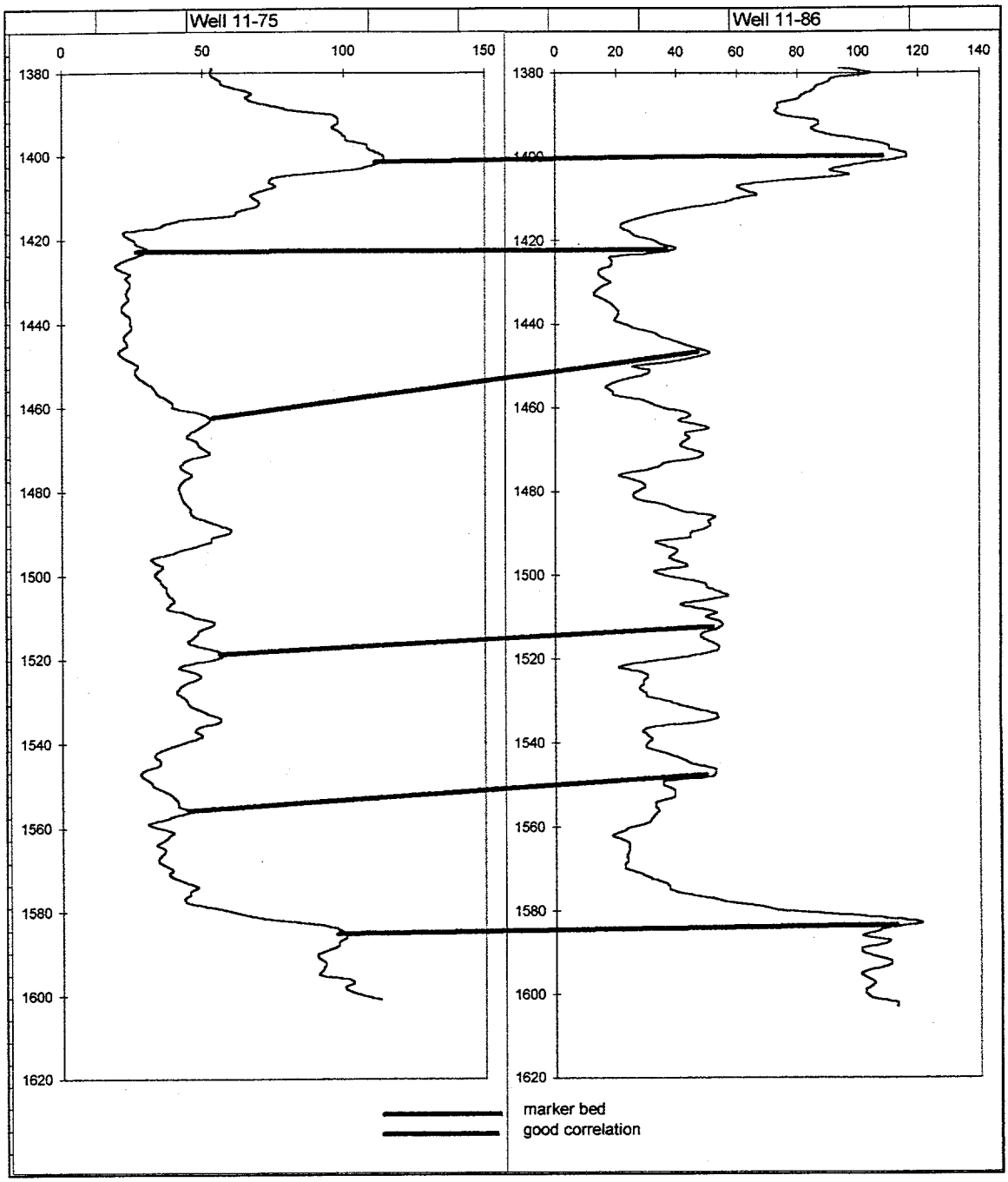
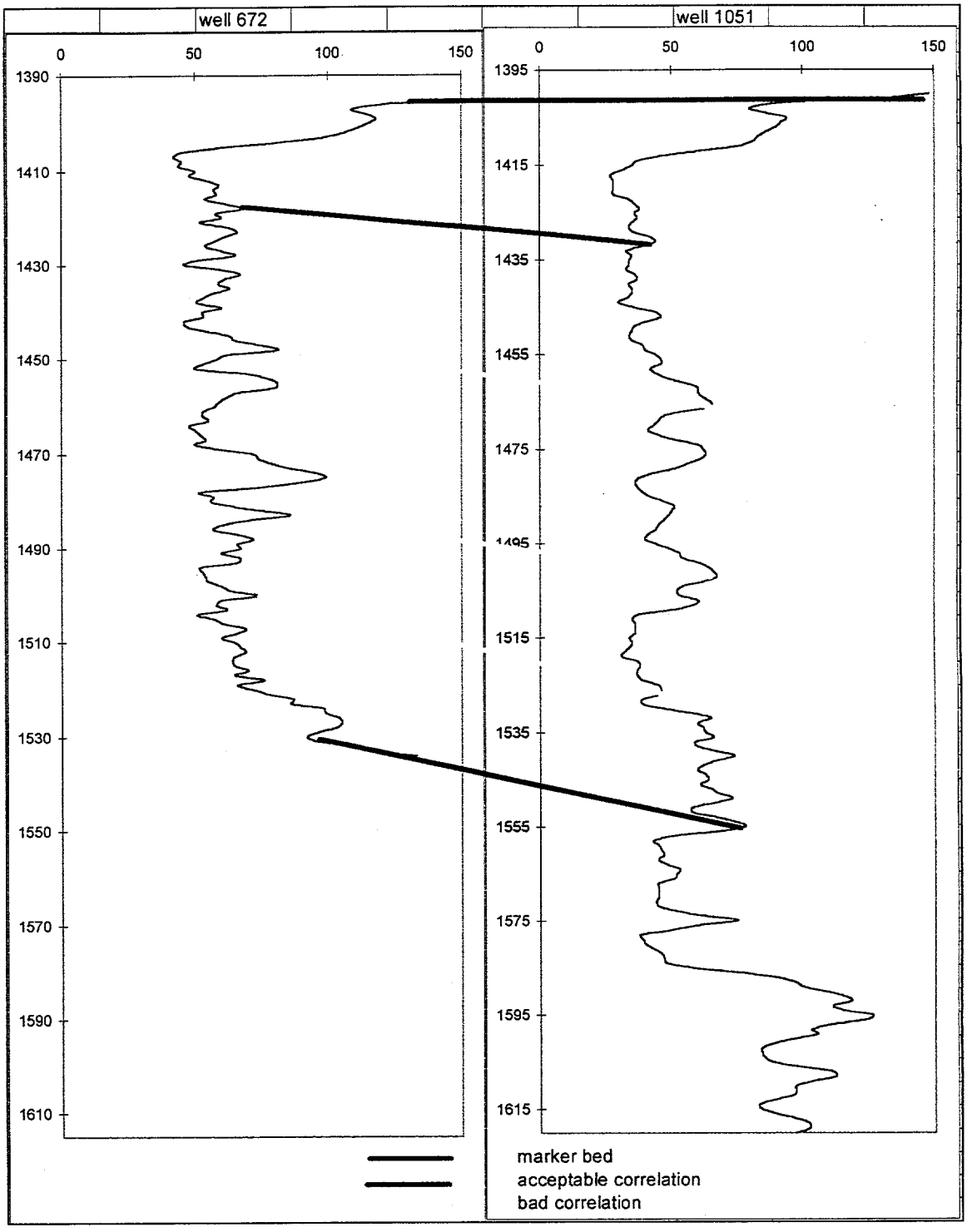


Figure 14: A bad correlation by the system: Wells 672 and 1051



5. Well Model Indentification System

The Well Model Identification System reported in the March 1996 quarterly report has integrated successfully all of the subcomponents which were encoded using both C++ and Fortran. The remaining parts to be developed within this system are:

- the incorporation and integration of a graphing facility to display the information
- the automatic transfer of data between the subsystems and the non-linear regression algorithm
- the integration of the well model identification system with the whole reservoir integration syste.

The final task involves using information obtained from the geological interpretation system as input to the well model identification system. In addition, the output of the well model identification system will be used to with the results from the geostatical system to determine the final reservoir description.

6. Integrating the Component Systems into a Single Reservoir Characterization System

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 converstion 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 intergration of the whole system using this approach in the next section.

6.1 Integration of the Subsystems

Despite the fact that the code was written in both FORTRAN and C++, integrating the components of the geological, geostatistical, and well test systems should be achieved using the following approach.

1. Any needed modifications are done to the files to allow them to be used together.
2. A driver is created to call the individual systems.
3. The code is compiled and debugged until working object modules are produced.
4. A single executable files is created by linking the object files.

Due to problems with the compilers, libraries, and code itself, integration was not as easy as it should have been. In the above steps, modification needed to be made to avoid the errors associated with each step.

Step 1. Making the modifications to the files to allow compilation is not a complicated matter. Errors in this step are easily located and fixed. Errors that fall into this category are problems such as duplicate function names in the code of different components, error messages that terminate execution prematurely, and path names to data files that are not compatible between machines.

Step 2. Coding the driver is also a relatively simple step. Because some of the code is written in C++ and some in FORTRAN, a special declaration statement is needed for the C++ compiler to be able to recognize FORTRAN subroutines. If this statement is coded correctly, it will appear that the FORTRAN and C++ are seamlessly integrated.

Step 3. It is during compilation and linking that most of the problems with integration appear. Using Microsoft FORTRAN PowerStation and Microsoft Visual C++ with the Microsoft Developer Studio, the programmer can write code in either language, and during compilation, the Developer Studio will call the needed compiler. In some cases, however, using both compilers together can cause the FORTRAN object files to become corrupted. There can also be conflicts between the required FORTRAN libraries and the required C++ libraries resulting in a link error. These problems can be resolved by turning off the optimization on the FORTRAN compiler and removing the conflicting sections of the libraries from the C++ libraries.

Additional problems result from using the Developer Studio itself. The Developer Studio contains two platforms through which you can write and compile code. The Win 32 - Debug platform creates files that record execution and make debugging the code easier. These files must be run within the Developer Studio. The Win - 32 Release platform creates no such files. It simply creates the object files from the code. The libraries and programs used by the Debug platform can cause problems. For example, when executing C++ code that contains linked constructs and has been compiled under Debug, there can appear run-time memory access violations. The easiest solution to this problem, assuming that the code has been thoroughly tested to be correct, is to use the Release platform to compile C++ code containing linked constructs.

Step 4. Once the code has been successfully compiled and linked, a executable is created that can have problems at run-time. Any time a potential divide by zero is encountered in a piece of code, a run time math error is produced and the computer stops execution. The compiler does not necessarily locate this error in the code because the values of the divisors may not be set until execution. When the FORTRAN and C++ code are compiled and linked, a divide by zero error occurs during execution that does not happen when the pieces of code are compiled and run separately. Apparently, the C++ compiler can handle this problem and allows an executable to be created that prevents this run-time math error. The FORTRAN compiler, however, does not overlook the potential divide by zero. When executed together, the FORTRAN libraries linked to the code cause an error statement to be produced on the C++ code. The problem, though difficult to identify originally, can be easily fixed with a conditional statement in the C++ code wherever potential divide by zeros exist.

An unusual problem that had been encountered during integration also relates to the use of the Developer Studio. FORTRAN code that compiles, links, and executes fine under the Debug platform will not execute properly under the Release platform. Though much debugging has been done on the code, no improvements have been made. The code still fails to execute under Release. Because the C++ code contains linked structures and must use Release, it is important that this problem be solved before any integration can take place. At this time, the solution to this problem requires further research.

Most of the problems that have arisen thus far during integration have been solved. It is likely that more problems will be encountered as programming continues, but the experience gained thus far should make solving future problems easier.

7. Graphical Interface

This section presents the initial development of a graphical user interface. We are experimenting with the use of Microsoft Visual C++ interface capabilities, along with basic MS Windows programming. The graphical interface will allow users to handle different graphic formats to present data, and to process the data through the individual tools. Figure 15 shows how it looks in MS Windows95 (it also can be used on MS Windows NT 3.51). In Figure 15 also shows the current graphical formats that are allowed: well log representation (at right) and two dimension representations (at left).

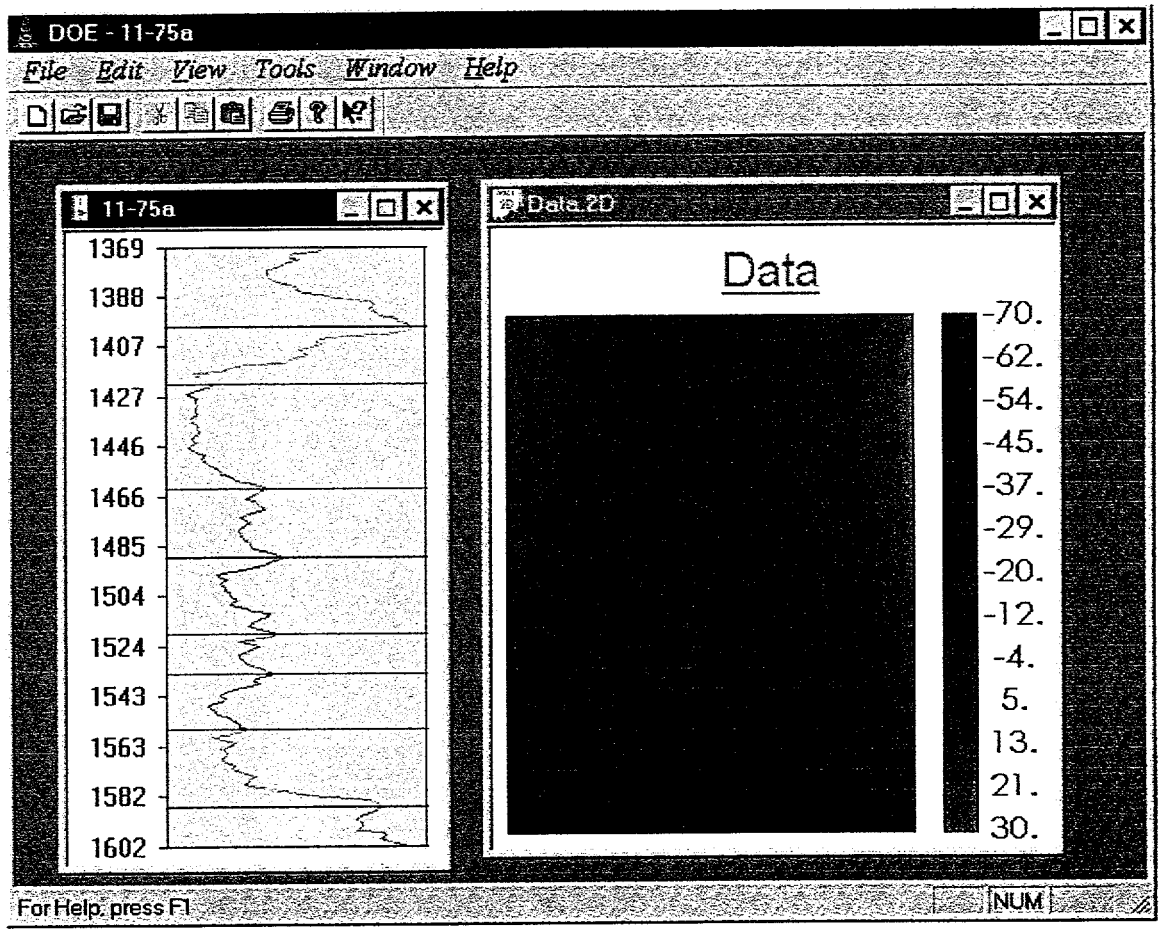


Figure 15: The graphic interface.

Figure 16 shows the application allows one to load and save data in multiple formats. The current formats are:

.log	only use for well log data.
.2d	only use for 2d graphics.
.txt	use for well log and 2d data but using text format.

Figure 17 shows the application and the well log graphic representation. On this representation is possible to use two tools. One of them is *Set Cuts*, used to identify cuts in the log. The other is *Set Facies* used for facies recognition. Figure 18 shows the same well log with cuts and identified facies.

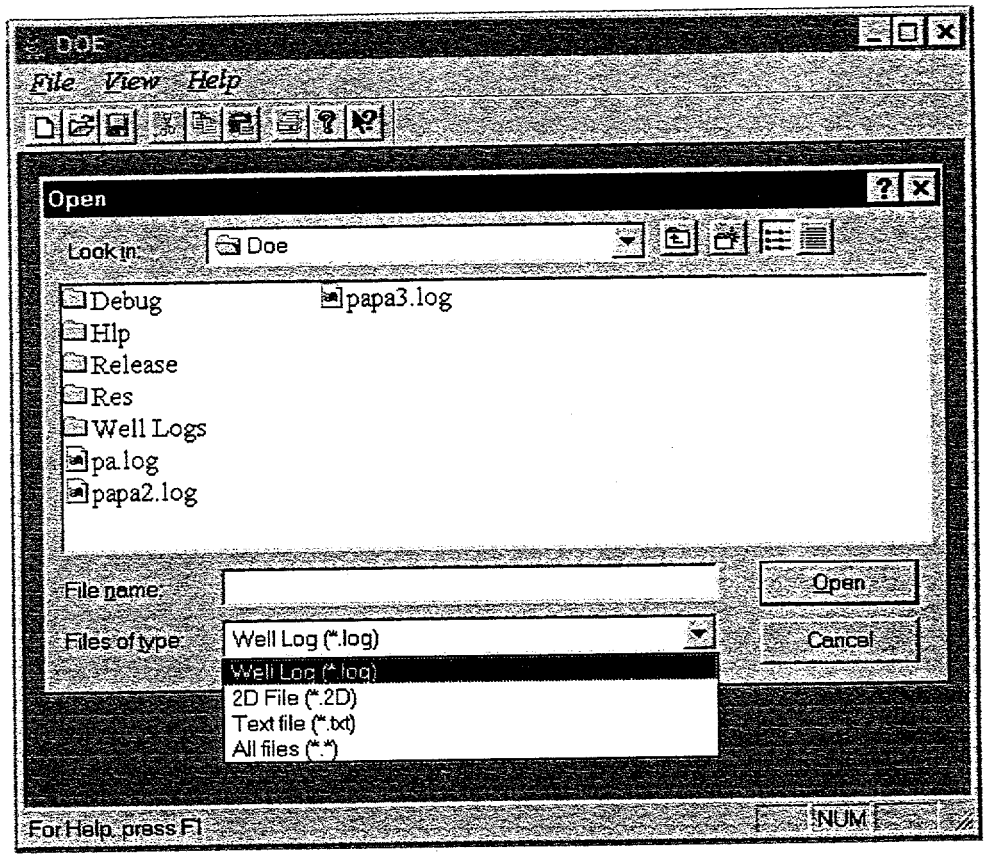


Figure 16: The application and the *Open file* dialog box.

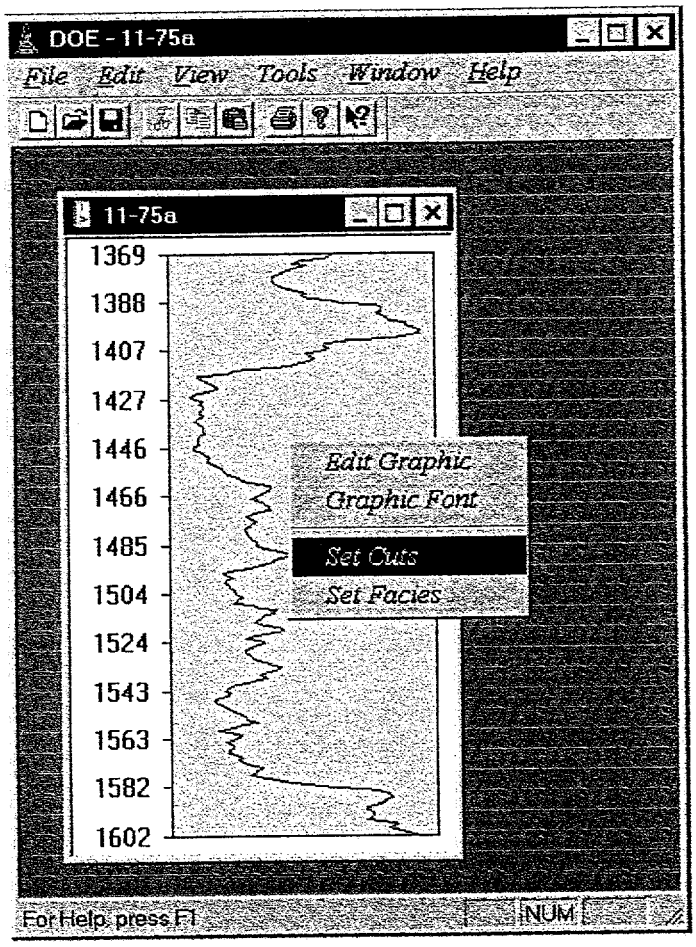


Figure 17: The application and well log graphic with unidentified facies.

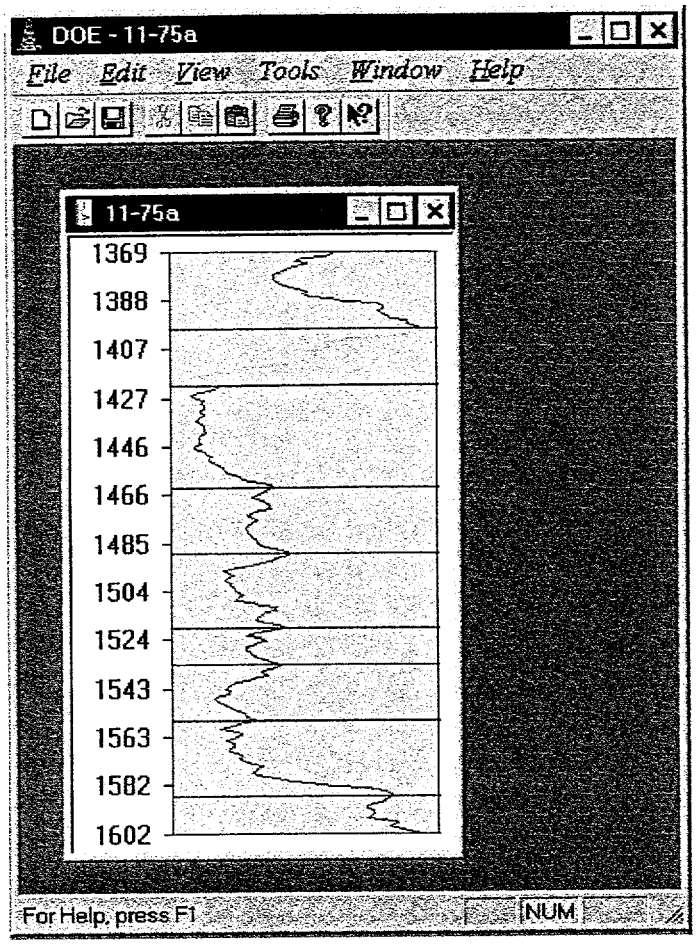


Figure 18: The application and well log graphic with identified facies.

8. References

1. Peaceman, D.W.: "Interpretation of Well-Block Pressures in Numerical Reservoir Simulation with Nonsquare Grid Blocks and Anisotropic Permeability", paper SPE 10528 presented at the Sixth SPE Symposium on Reservoir Simulation of the Society of Petroleum Engineers of AIME, New Orleans, LA, Jan 31-Feb 3, 1982.