DOE/BC/14894-10
Distribution Category UC-122

Application of Artificial Intelligence to Reservoir Characterization: An Interdisciplinary Approach

Annual Report for the Period
October 1994 to October 1995

By
D. Kerr
L. Thompson
S. Shenoi

January 1996

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

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

Robert Lemmon, Project Manager
Bartlesville Project Office
P.O. Box 1398
Bartlesville, OK 74005

Prepared by
The University of Tulsa
Tulsa, OK 74104

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

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 a 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 note merely limited to obtaining better characterization of individual reservoirs. They have the potential to significantly impact and advance the disciplines 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 1.1 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. In Figure 1.1, each component system in the model is depicted as interfacing (through the bi-directional links) with a central repository of reservoir descriptions. Though, portions of these description will essentially be passed from component to component as more information is gathered (as shown by the bi-directional links), the model of a central repository is an accurate account of how the components are integrated, i.e., the final descriptions in the repository are consistent will all of the information given by the components systems. This system model allows us to develop the system using an Artificial Intelligence technique called a *blackboard system*, in which information is centrally located, i.e., on a blackboard, and experts take their turn to update, change, and correct the information on the blackboard.

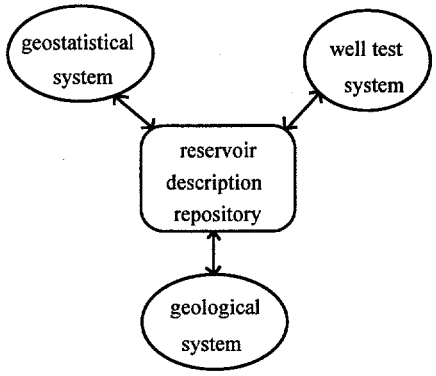


Figure 1.1: Expert System Decomposition

All implemented code has been portedinto C/C++ for integration. This includes both Fortran and Kappa-PC implementations of the various component systems that have been developed to this point.

2. Geostatistical System: Incorporating Dynamic Constraints in a Reservoir Description Process"

2.1 Introduction

The primary objective of this work is to develop a procedure for implementing dynamic constraints, such as well test and production data, in a reservoir description process. This should improve the description because:

i. It is a generally-accepted principle that the addition of constraints reduces uncertainty and thus a closer approximation of the reservoir is obtained.

ii. Reservoir parameter estimation from point measurements is an ill-posed problem; stabilization is thus required in order to obtain a meaningful estimate. The addition of prior information as a constraint will enhance our ability to stabilize the ill-posedness of our problem.[1]

iii. Dynamic constraints, e.g. production and pressure data, provide the only observable response which is representative of reservoir-scale behavior. They also provide "in-situ" information. Other available data, including log and core data, represent reservoir data which is limited to the near-wellbore region.

iv. In the proposed approach, geological constraints can be subsequently included to create a truly integrated reservoir description.

2.2 Approach

Conditional simulation, because of its ability to honor the sample data and create multiple equiprobable reservoir descriptions if required, has been selected as the stochastic method which will be used for this study. Specifically, the conditional simulation algorithm to be employed is the *Simulated Annealing Method.*[2,3] This method was selected because of its robustness and the facility and flexibility in incorporating conditioning data and additional constraints.

Simulated annealing is an example of adaptive heuristics for multivariate or combinatorial optimization.[3] Its implementation requires the definition of an objective function, the magnitude of which defines how well our estimated model "fits" the input model. A parameter which can characterize the model is used in the objective function (variogram models are normally used). The goal of the algorithm is to optimize

(minimize) the objective function. Note that more than one model parameter may be used, in which case we attempt to minimize the combined effect of these parameters.

Our approach involves the coupling of a numerical flow simulator to the simulated annealing algorithm. The flow simulator will constitute a dynamic constraint as part of the simulated annealing objective function. The objective function will then consist of a variogram constraint and a flow simulation constraint.

The *Laplace Transform Finite Difference* (LTFD) method[4] will be used for the flow simulation. This method is based on solving the single-phase flow equations in Laplace space. It has the advantage of using only one timestep between the initial time and the time of interest. As a result this method is computationally far less expensive than other numerical methods.

2.3 Areas of Research Focus

2.3.1 Input Data

Analysis of the behavior of the composite objective function led to the conclusion that the rate of convergence was too slow. Additionally, the terminal objective function value of about 0.1 was too high for one to conclude that convergence to a globally-optimal solution had occurred.

For the flow simulation part of the objective function, we compare observed pressures to pressures calculated by the flow simulator as a means of judging how closely the performance of the estimated model approximates the "true" reservoir description. This comparison is done in Laplace space, and so a routine to convert the real time and space data to Laplace space is used.

We determined that the precision of the Laplace transform of the input data could be improved by the inclusion of more data values for better interpolation and by choosing the appropriate data extrapolation procedures (which depends on the flow regime characteristics at the time from which we have to extrapolate).[5] This improvement in precision produced better results in that convergence was faster and the terminal objective function value reached about one order of magnitude lower at approximately 0.01. Also the maximum absolute percentage error between the 'observed' and calculated Laplace space pressure values was reduced to about 0.8%. It is thus necessary to undertake some pre-analysis of the input data to ensure that the data quality and quantity are adequate and also that the appropriate algorithms are used.

2.3.2 Training Images

In recognition of the direct relationship between the time required for running the simulated annealing algorithm (especially when the flow simulation is included) and the the level of grid discretization, it was decided to undertake runs with smaller grids (upscaled grids). The algorithm would then run faster, enabling speedier analysis of the optimization schemes. The use of various approaches was considered for generating the

"training images" used in testing our algorithm. The turning bands method,[6] sequential Gaussian simulation (sGs)[7] and simulated annealing (SA)[2] were some of these.

The turning bands method has two major drawbacks, which will be mentioned but not discussed here:

(i) artifact banding in the realizations,
(ii) difficulty in handling anisotropy directions other than the coordinate axes.

sGs, for the grid sizes with which we worked, also did not perform to our satisfaction, by requiring more conditioning data than we wanted to use.

Because of the shortcomings of the first two, we developed an approach using SA. Code was developed to create a cumulative distribution function (cdf) for log-normal data. This was then used as input to a variogram-only objective function SA run to obtain the synthetic image used as a training set. In this approach we can control the variance and mean of the model we use, as well as the variography.

2.3.3 Algorithmic Considerations

By far, the major effort was expended in developing and improving the algorithm. The three main packages analyzed were the flow simulation code, the matrix solver and the SA code. Changes to the SA algorithm were considered only in terms of modifications made to implement the flow simulation constraint. The basic, variogram constraint-handling SA code was considered sufficiently optimized. Changes to the matrix solver were made, not by modifying the code, but by considering alternate solvers. The flow simulation code, however, was closely scrutinized. From the above comments, it can be seen that all these efforts were directed at improving and optimizing the flow simulation performance, since this was recognized as the critical area needing attention.

2.3.3.1 Flow Simulation Code

(a) *Coefficient Matrix Code*

The heart of the flow simulation code is the development and solution of the matrix of coefficients for the gridblock pressures. We did a detailed study of the structure of that matrix and determined that, for the given boundary conditions, the matrix is a 7-band, sparse, *symmetric* system. There is little scope for improving the matrix formulation code, but we were able to optimize the code which *updates* that matrix (which is required when we perturb the description). We determined that on update only 19 matrix elements are affected (13 for a 2-D grid), thus we need to change only those elements. Initial testing has shown that the changes have resulted in some gain in efficiency, but not of the magnitude hoped for.

(b) *Variable Rates or Constant Rates*

When variable rates are used -- i.e. rates which are step functions of time -- it has been observed that the LTFD method results in pressure changes, coincident with the rate

change, which are "smoother" than they should be. This smoothing problem is confirmed when we compare the results from the LTFD simulator with the results from the *ECLIPSE-100* (ECL)[8] simulator. In fact, *in general* the use of variable rates with Laplace transforms is problematic. One solution[9] has been to use a constant rate solution in Laplace space and use superposition in real space to correct for the variable rates. This is not a viable approach for our case, since we solve our problem entirely in Laplace space. Recently a procedure[10] was presented for correcting this anomaly. This has led to our consideration of two approaches:

(i) the use of the Chen et al approach[10] to correct the variable rate problem,

(ii) the use of constant rates to avoid the Laplace transform problems and the problem of the introduction of new transients at each rate change.

The equation given in reference 10 for the "corrected" bottomhole pressure in Laplace space is:

$$\bar{p}_{wD}(s) = \frac{q_{D_1}}{s_1^2\left[C_D + \frac{1}{s_1 G(s_1)}\right]} + \sum_{j=2}^{n} \frac{t_D}{t_D - t_{D_{j-1}}} \frac{q_{D_j} - q_{D_{j-1}}}{s_j^2\left[C_D + \frac{1}{s_j G(s_j)}\right]} \qquad (1)$$

where

$\bar{p}_{wD}(s)$ is a Laplace space approximation for a variable rate p_{wD}

s_j is the Laplace variable based on $t_D - t_{D_{j-1}}$

C_D is the wellbore storage coefficient

t_D is dimensionless time

q_D is dimensionless rate

$G(s)$ is defined by the equation: $\bar{p}_{wD} = \frac{1}{sG(s)} = p(s)$.

Firstly, it seems that this equation may be incorrect, so we will have to verify its accuracy before any further consideration of its usage. Secondly, the structure of the equation suggests that, in order to determine the variable rate, Laplace space, bottomhole pressure for a particular value of *s*, we have to determine the constant rate, Laplace space solutions for a number of *s* values equal to the number of times the rates are changed. This would mean introducing a significant extra computing "load" which would add to the algorithm's inefficiency. For this reason, we have suspended further consideration of this approach and for the moment we are concentrating on the **constant rate approach.**

2.3.3.2 Matrix Solver

We considered matrix solvers from the SLAP,[11] ITPACK[12] and Templates[13] packages. Some solvers promise faster execution times; however, because of the specialized nature of those approaches, they have not yet been tested. For example, red-black ordering implies parallelization of the code -- something which we are not

considering at present. We did not test the Templates codes because of time constraints and there seemed to be minimal advantage in exploring this alternative. Table 2.1 summarizes the results obtained with the packages tested. A 224-block dataset was used for the testing.

Package	Method	Run Time (CPU Sec)
ITPACK	Jacobi Conjugate Gradient (JCG)	14096
	Symmetric Successive Overrelaxation Semi-Iteration (SSORSI)	44084
SLAP	Conjugate Gradient (CG)	13824
	Generalized Minimum Residual (GMRES)	22045
	Bi-Conjugate Gradient Squared (BCGS)	14283

Table 2.1: Comparison of Run Times for Various Matrix Solvers

Based on the above results, we have selected the **SLAP CG** code for use. Consultation with the Mathematical and Computer Sciences Department[14] at the University of Tulsa has suggested that our choice was the appropriate one.

2.3.3.3 Simulated Annealing Code

(a) *NVAL Parameter*

NVAL is the parameter used for the number of pressure values per well used for comparison in the flow simulation part of the objective function of the simulated annealing algorithm. This is defined in the following equation:

$$\text{Flow Simulation Part for Iteration, k} = wgt2 * \frac{1}{E_2^0}\sqrt{\sum_{i=1}^{NDIFF}\left[\frac{L[\Delta p(i)]_s^k}{L[\Delta p(i)]_b} - 1\right]^2} \qquad (2)$$

where

wgt2 =weighting applied to the flow simulation part

$$E_2^0 = \sqrt{\sum_{i=1}^{NDIFF}\left[\frac{L[\Delta p(i)]_s^0}{L[\Delta p(i)]_b} - 1\right]^2} = \text{Initial Flow Simulation Energy} \qquad (3)$$

NVAL = Number of summation terms per well

NWELL = Number of wells

NDIFF = *NWELL***NVAL*=number of terms used in the summation

$L[\Delta p$ = the Laplace transform of the change in the flowing bottomhole pressure

Ten was the *NVAL* value being used. The run time of one iteration in the algorithm is proportional to *NVAL*, so we analyzed the effect of the value of *NVAL* on the

performance of the code. Values of *NVAL* ranging from five to twenty were used. The observations so far are:

(i) The lower the value of *NVAL* the shorter the cycle time, but, in some cases, more cycles are needed for convergence. This suggests that there is/are optimum values for *NVAL*. It seems that this value is about six or seven. However, these results are for just one dataset so far.

(ii) The relative "goodness of fit" of the distributions obtained using the different *NVAL* values is difficult to determine by visual inspection (i.e. whether one description is better than another) and, since we are still researching an appropriate *objective* measure of goodness of fit, has not yet been determined.

(b) *Variogram Pre-Selection Approach*

It was suggested[15] that we could significantly improve the execution time of the SA algorithm if we could determine *before running the flow simulation* whether a perturbation would be "beneficial". We subsequently determined that using the variogram-based objective function change as a criterion for deciding whether a perturbation would be accepted, was a good approach since the acceptance of the variogram-only change was highly correlated with that of the two-part objective function change (conservatively over 60%).

(c) *Two-Scale Approach*

If we could use an upscaled grid for the flow simulation part of the algorithm, then we would be able to reduce the execution time of the algorithm considerably. In addition, this approach would allow the consideration of grid sizes (at the fine scale) which were previously prohibitively large for flow simulation. Note that we would still perform the variogram analysis at the finer scale. The upscaling methodology is the major problem to be solved for this approach to be viable. While the upscaled distribution "resembles" the true distribution, matching the flowing bottomhole pressures obtained from flow simulation at the different scales is a problem.

(d) *Two-Scale, Variogram Pre-Check Approach*

This is a combination of the ideas in (b) and (c) above. It is the approach on which we are currently focusing. In this approach the objective function consists of two parts: a variogram component and a flow simulation component. The perturbation strategy is single-point. The algorithm proceeds as follows:

i. An initial distribution is generated which honors the spatial statistics and the conditioning data.

ii. The initial temperature level is determined by the procedure suggested by Aarts and Korst.[16]

iii. A *coarse scale* gridblock location is randomly selected. Based on the upscaling factor, the corresponding *fine scale* grid locations are

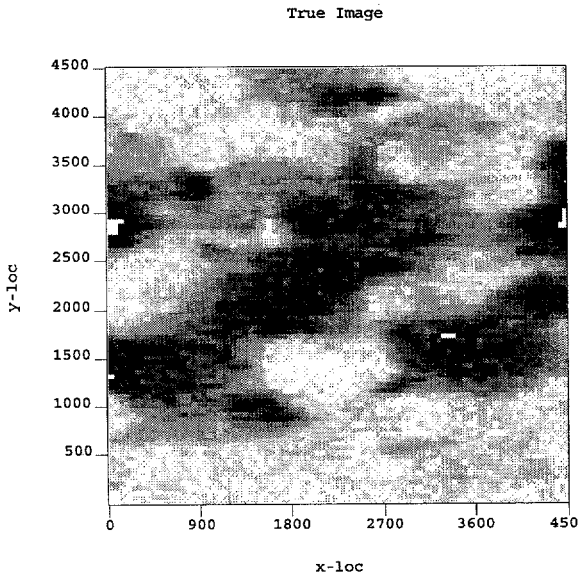
determined, and *one* of these is perturbed. We evaluate the *variogram part* of the objective function on the fine scale grid system and test the energy change via the Metropolis condition. This is the pre-selection stage.

iv. If the perturbation is accepted then the average of the permeabilities of the fine scale blocks -- which make up the upscaled block selected in part iii above -- is determined. In this way the upscaled grid system is perturbed and this modified system is flow simulated. The flow simulation-derived change in the objective function (energy) value is thus determined *on the upscaled grid system.* The approach is thus a 2-scale one.

v. The composite (combined variogram and flow simulation) change in the objective function is determined. Again the Metropolis condition test is used and if *this* test is passed then the distribution is modified by updating with the new permeability value at the perturbed fine scale location. The energy (objective function) value is also updated.

vi. The convergence tests are conducted. If not satisfied, another location is selected and we loop again.

This approach was used on a 8100-block (90x90) grid system. The results are shown in Figure 2, comparing the true distribution with those obtained with a variogram-only objective function and the composite objective function.



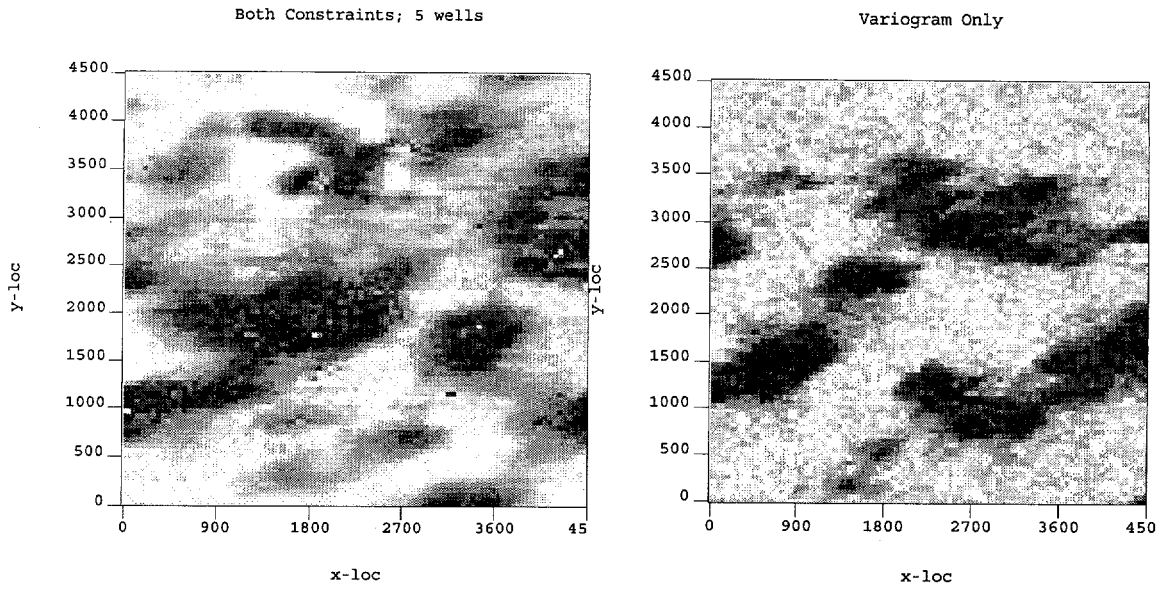


Figure 2.1: Comparison of the "Training Image" with Images Obtained Using a Two-Part Objective Function and a Variogram-Only Objective Function

The image generated using both constraints (variogram and flow simulation) seems to better capture the major features of the true distribution. As noted at (c) above however, the flowing bottomhole pressures from the upscaled model do not match the true scale (fine scale) values very well, especially at early times, thus we expect that our match would be better if we could formulate a methodology which enabled the flowing bottomhole pressures obtained at the different scales to match.

The SA run for the composite objective function took about 18 hours of CPU time on an HP RISC workstation with an objective function tolerance of 0.05. It should be noted that this does not represent the limits of the algorithm's efficiency since better or more sophisticated matrix solvers (for example the *ECLIPSE-100* solver: conjugate gradient accelerated by orthomin) can improve the execution time required. Should we attempt to use a 1-scale, no pre-select approach for a grid of this size, it would take a prohibitively long time -- on the order of several days.

(e) *Time Scale Used for Flow Simulation Pressure Comparisons in SA*

In a strict sense, upscaling results are valid only after the onset of pseudo-steady state flow. In the SA algorithm not much attention was paid to this constraint, and the comparison of pressure values was made at times over the entire range of the flow period. However, as alluded to at (c) and (d) above and shown in Figure 2.2 below, when we compare the pressure response obtained from flow simulating a fine scale description and that of its equivalent coarse scale model (obtained, e.g. by geometric averaging upscaling), there are significant errors in the pressure values, especially at the early times. These results are for a 33x33-block reservoir (the upscaled grid is 11x11) with three wells flowing at constant (but different) rates.

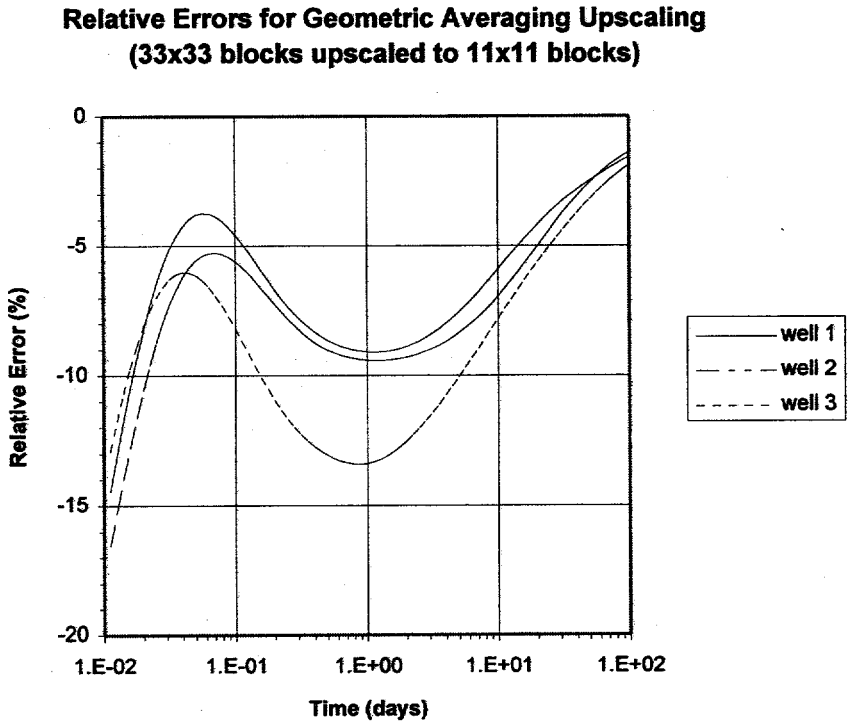


Figure 2.2: Example of Relative Errors in Upscaled Flowing Bottomhole Pressure as a Function of Time

These results prompted further investigation into upscaling, including an analysis of alternative approaches used for permeability upscaling.

2.3.4 Upscaling

2.3.4.1 Comparison of Upscaling Methods

(a) *Methods and Dataset Used*

The three upscaling methods used were: renormalization, geometric averaging and tensor upscaling as displayed in Figure . A fine scale, 8,100 (90x90) blocks permeability distribution was upscaled using two factors: 9-to-1 and 25-to-1, resulting in 900 (30x30) and 324 (18x18) blocks coarse scale distributions respectively. The fine scale and all the upscaled distributions were flow simulated for approximately 300 days with 5 wells producing at constant (but different) rates. The resulting profiles of flowing bottomhole pressures (pwf) over time for all the upscaled grids were compared to the results for the fine scale grid, which we consider as the "truth case". Thus plots of relative error in pwf

error in pwf (actually the error in the *change* in pwf) versus time were developed for all the upscaled distributions.

(b) *Error Profiles from Different Upscaling Methods*

The behavior of the relative errors for the distributions obtained from the three upscaling methods were quite similar for both the 9-to-1 and 25-to-1 cases. Thus one upscaling method does not seem to have any clear advantage over the others when the flowing bottomhole pressures are compared.

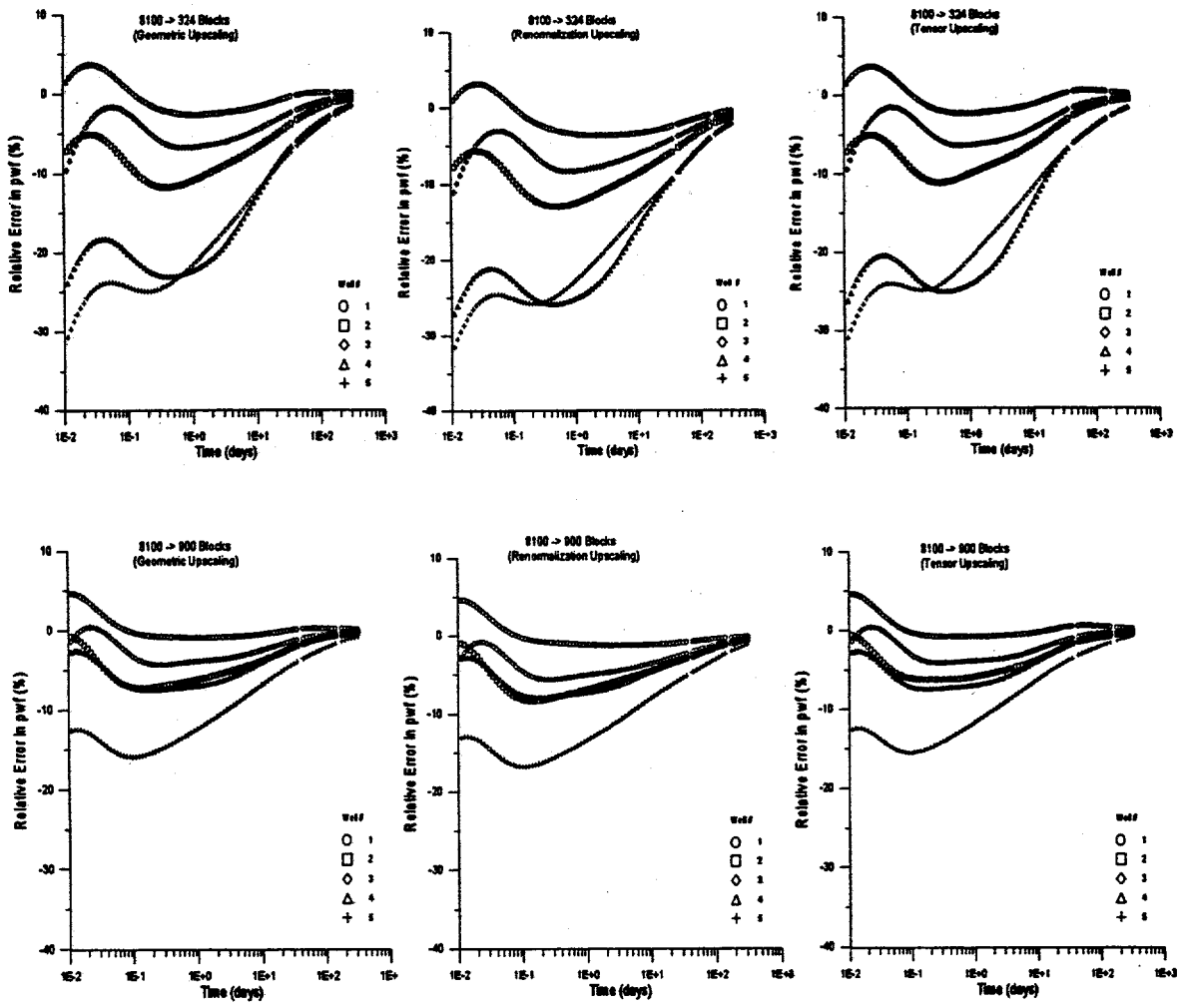


Figure 2.3B: Relative Errors in Upscaled Flowing Bottomhole Pressure as a Function of Time for Three Upscaling Methods & Two Scaling Factors

(c) *Coarse Scale Wellblock Permeability Values From Different Upscaling Methods*

The upscaled wellblock permeability values obtained from the three methods were found to be quite close. Table 2.2 and Table 2.3 summarize these results. This agrees with the observation in (b) above, and supports the conclusion that, in terms of this analysis, one upscaling method has no apparent advantage over the others.

well #	8100-blks	900-blks				
		tensor-x	tensor-y	renorm-x	renorm-y	geometric
1	69.005	68.681	68.652	68.656	68.636	68.665
2	83.943	78.152	79.077	78.860	78.143	78.623
3	33.898	32.148	32.924	32.400	31.813	32.535
4	66.844	59.330	60.741	60.650	59.271	60.045
5	82.171	65.291	64.947	65.032	64.618	65.121

Table 2.2: Coarse Scale Values for 9-to-1 Upscaling Factor

well #	8100-blks	324-blks				
		tensor-x	tensor-y	renorm-x	renorm-y	geometric
1	69.005	67.511	67.885	67.327	67.325	67.682
2	83.943	73.867	74.509	74.610	72.985	74.222
3	33.898	31.055	32.512	32.081	30.536	31.736
4	66.844	43.803	56.304	56.205	43.251	50.629
5	82.171	55.495	57.768	57.638	55.154	56.783

Table 2.3: Coarse Scale Values for 25-to-1 Upscaling Factor

2.3.4.2 Analysis of Relative Errors in Flowing Bottomhole Pressures

(a) *Error Definition*

The error in pwf is defined as:

$$E(\%) = 100\left(\frac{\Delta p_{wf_{fs}} - \Delta p_{wf_{us}}}{\Delta p_{wf_{fs}}}\right) \tag{4}$$

where

$\Delta p_{wf_{fs}}$ = fine scale Δp_{wf}

$\Delta p_{wf_{us}}$ = upscaled Δp_{wf}

Referring to Figure , it can be seen that all the graphs show similar trends: increasing to a "hump", decreasing to a "minimum" value, then converging asymptotically to zero. The early-time errors in the flowing bottomhole pressures are defined as the errors up to the hump in the plots. Because of the times at which these occur, it seems that these humps correspond to the times at which the upscaled wellblock boundaries are reached by the pressure transient. An analysis of the plots shows that the early-time error is a function of the magnitude of the difference between the wellblock permeabilities at the fine and upscaled grids. Darcy's Law defines an inverse relationship between pressure change and permeability, thus when the fine scale wellblock permeability is *higher* than the upscaled wellblock permeability (which means that the pressure change at the upscaled wellblock should initially be *higher* than that at the fine scale wellblock) we observe that the error tends to lie on the *negative* side of the error (vertical) axis. This behavior is shown in the graphs and further to this, we see that the larger the difference in the permeabilities at the different scales, the larger is the early-time error. In addition, we had found that the wells' proximity did not seem to have any significant effect on the errors.

These results suggested two approaches for dealing with the early-time errors:

(i) find a better upscaling algorithm
(ii) use an alternative approach -- other than upscaling -- for handling the near-well permeabilities.

2.3.4.3 Modified Upscaling Approach

(a) Single Phase Flow

Results from the first approach are shown in Figure 2.4. A 9-to-1 upscaling factor is used with a 33x33 (1089) blocks grid at the fine scale and 11x11 (121) blocks at the coarse scale. Figure 2.4 shows the errors in the pressures when the conventional geometric averaging upscaling technique is used and when a modified technique is used. In the modified approach, we set the upscale wellblock permeabilities as a weighted average of the arithmetic average of the 8 fine scale blocks surrounding the wellblock and the fine wellblock permeability value. The weights applied are 0.1 and 0.9 respectively. Conventional geometric averaging is used elsewhere. It can be seen that there is substantial improvement in the results, since for the modified upscaling approach, the errors lie within ±7% and converge to 0% faster than the conventional results. Similar results were obtained for a 45x45 (2025) grid upscaled to a 15x15 (225) coarse scale grid.

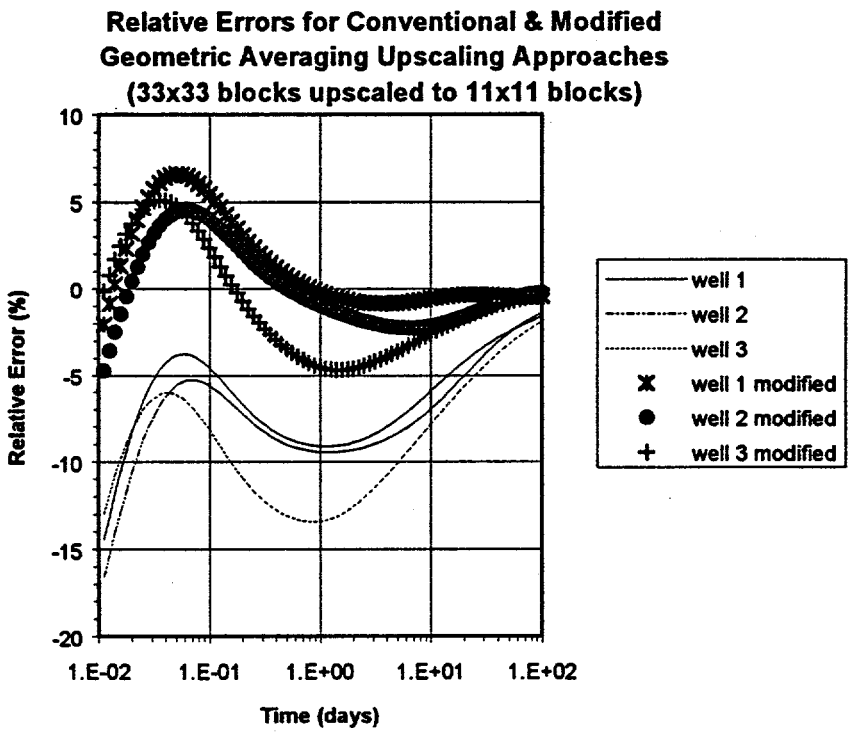


Figure 2.4: Comparison of Results Between the Conventional & Modified Geometric Averaging Upscaling Methods

(b) Two Phase Flow

A model for two phase flow system is developed after upscaling the fine scale grid by a factor of 5 in each direction. The fine scale grid consists of 90 x 90 (8100) grid blocks and the coarse scale consists of 18 x 18 (324) grid blocks. In applying the modified upscaling technique for the wellblock, three weight values are required; one for the fine wellblock permeability, one for the arithmetic average of the first layer (8 fine scale blocks surrounding the wellblock), and one for the arithmetic average of the second layer (16 fine scale blocks surrounding the first layer blocks). The weights applied are 0.9, 0.08, and 0.02. Figure and Figure show the relative errors for the geometric average and modified upscaling technique respectively. As in the single phase flow case, significant improvement is obtained after applying this modified technique. The errors are reduced and converged to zero much faster.

This technique suggests that the fine scale wellblock permeabilities should be much more heavily weighted in the upscaling process. The results also show some promise that this may lead to effective upscaling technique.

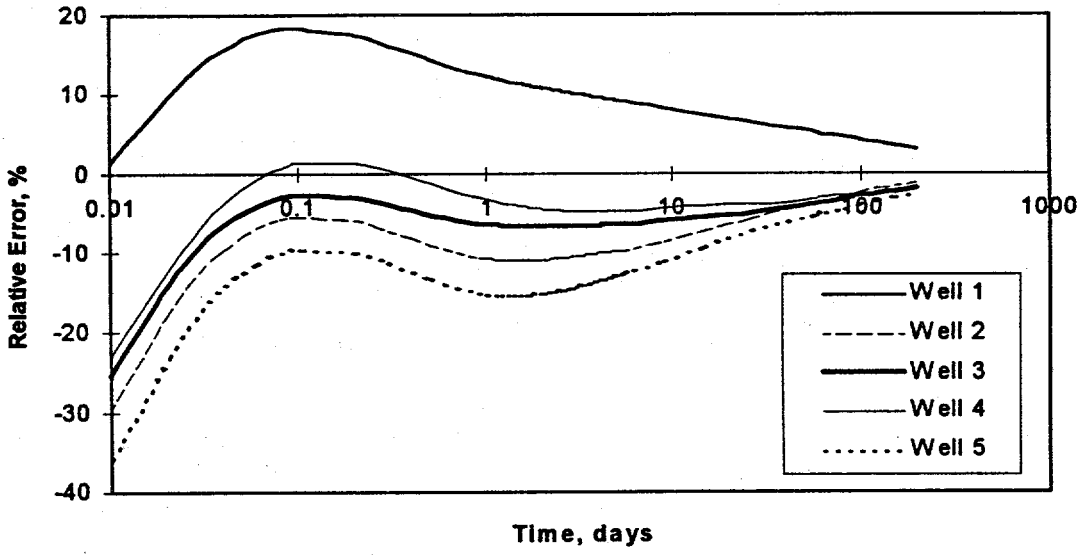


Figure 2.5: Relative Errors for Two Phase Flow Model using Geometric Average Upscaling Technique

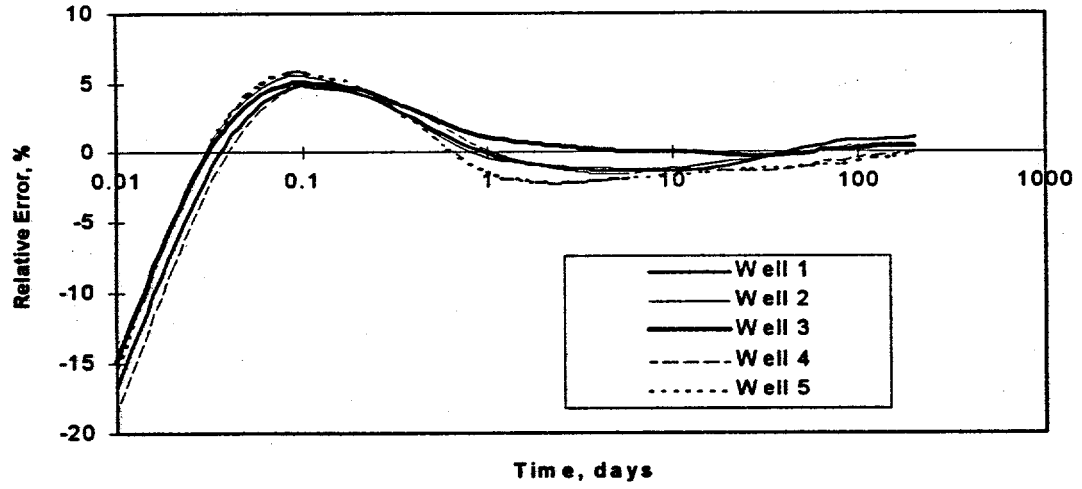


Figure 2.6: Relative Errors for Two Phase Flow Model using Modified Upscaling Technique

2.3.4.4 Radius of Investigation Concepts and Approach

The second approach of dealing with the early-time errors involved consideration of the radius of investigation concept. Since the *transient* behavior of the pressures seemed most problematic, we considered using an analytical approach to obtain the upscaled average permeability for the near-wellbore region by estimating the time at which

the pressure transient reaches the upscaled wellblock boundaries and iteratively solving for the average permeability.

(a) *Definition of Radius of Investigation*

Various authors have discussed this concept. In all cases the radius is defined as:

$$r_{D_i} = A\sqrt{t_D} \tag{5}$$

where

$$r_{D_i} = \frac{r_i}{r_w} \tag{6}$$

r_i = radius of investigation, (ft.)

r_w = wellbore radius, (ft.)

$$t_D = \frac{\bar{k}t}{\phi\mu c_t r_w^2} \tag{7}$$

$\bar{k}$ = average permeability, (md.)

t = time, (days)

ϕ = porosity (assumed constant in our case)

μ = viscosity, (cp.)

c_t = system compressibility, (psi^{-1})

A = a numeric constant

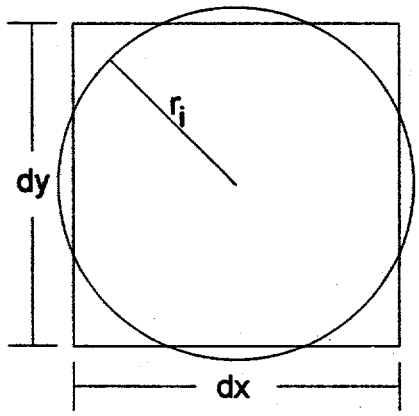
The A value has been defined as 0.03248 by Lee[17] for a homogeneous reservoir, 0.038 by Oliver[18] for a heterogeneous reservoir in which the permeability variations are small about a reference permeability and 0.01 by Alabert[19] as an 'optimum' value for a heterogeneous reservoir. Using the different A values and geometric averages for the upscaled block permeabilities, we attempted to verify which A value gave the best match to the geometric-averaged case. The results were inconclusive, since in some cases one A value showed better agreement, while in other cases the others performed better.

(b) *Approximations Used for Cartesian Grid System*

For convenience in using a Cartesian gridding system, we used the following approximation in this radius of investigation analysis:

$$r_i \approx \sqrt{\frac{dx\,dy}{\pi}} \tag{8}$$

where r_i, dx, dy are defined in the figure below.



For the case where $dx=dy$, there is an approximate 9% error in this assumption, in that the area lying outside of the grid is about 9% of the square grid area.

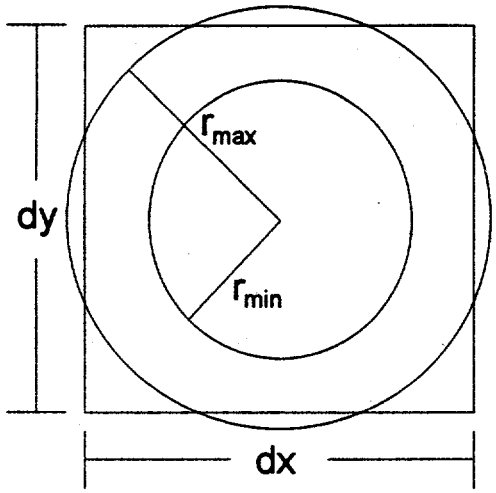
(c) *Oliver's Weighting Function*

In trying to determine the 'best' A value, we assumed that the 'effective' permeability is averaged over an approximately *circular* area (in 2-D) defined by the radius of investigation. However, as Oliver[18] has shown, the permeability affecting the change in the flowing bottomhole pressure is centered in an *annular* region. The location of this region relative to the wellbore is a function of time. In fact, the 'kernel' or 'weighting' function he derived, determines the relative contributions of the reservoir permeabilities at a given time on the 'effective' permeability at that time. It should be noted that, because of the shape of the weighting function, the permeabilities within the annular band do not all contribute equally to the effective permeability. Of interest to our problem are the definitions for the radii at which 1% and 99% respectively of the annular area are located:

$$r_{D_{1\%}} = 0.002\sqrt{t_D} \tag{9a}$$

$$r_{D_{99\%}} = 0.038\sqrt{t_D} \tag{9b}$$

where t_D is as defined before.



Note that the maximum (99%) radius is equivalent to the radius of investigation of other authors. This means that for our approach, the error in using a circular region as opposed to the more correct annular region is about 0.3% -- the ratio of the smaller area to the larger area. We thus conclude that the circular approximation, which is simpler to determine, is adequate for our purposes.

(d) *Average Permeability Determination*

The procedure for using this approach is as follows:

i. Using equation (8), calculate the equivalent radius to the block boundary.
ii. Initialize the estimate for the wellblock permeability (the fine scale wellblock value can be used). Call this k1.
iii. Using equation (5) with the estimated permeability value, calculate the time at which the transient would reach the block boundary.
iv. Assuming transient flow, calculate the semi-log slope by using a regression fit of the pressure values up to the time calculated above.
v. Calculate the effective permeability from this slope. Call this k2.
vi. If the absolute value of the difference between k1 and k2 is greater than the tolerance, replace k1 by k2 and go back to step iii. If not, stop. This is the value to use for the uspaceld wellblock permeability.

(e) *Results and Comments*

The results using this approach were mixed. In some cases, there was a reduction in the error, while in others where was not. Part of the reason seems to be that the permeability calculation is very sensitive to that of the semi-log slope, and we have found that within the time estimated to reach the boundary of the wellblock, there can be significant variations in the magnitude of that slope. For example, assuming that there are 10 pressure-time values within that time range, we would get different slopes if we were to use e.g. 5, 7, 10 of these values. In all cases also, the regression statistics would

indicate that the fits were good. So while there may be some promise to this approach, more work needs to be done to stabilize the results.

2.3.5 Wavelet Approach

The main purpose of applying the wavelet approach in this study is in the aim of reducing the flow simulation computing costs. Before we can actually implement this approach in solving the computation time problem, an investigation about the feature of the wavelet- base flow simulation is required. The subsequent paragraphs discuss this feature.

Wavelets are a family of basis function similar to Fourier transforms that have orthogonal basis functions called the *scaling functions* and the *wavelets*. The potential eficacy of the wavelet transform lies in two of its properties :

(a) the scaling functions and the wavelets have quite compact support, i.e., it is only locally-defined thus making it suitable for segregating information in data into local intervals, and

(b) the wavelet transform provides the flexibility to choose a particular basis (wavelet) function that is customized to the specific application.

Using the wavelet transform we can convert the original permeability distribution into frequency domain space which results into coefficients. These coefficients are related to the variation of the original distribution. The small coefficient values may contribute very little to the overall characteristic of the original distribution so we can discard them. It is the purpose of this study in finding out the level where we can still discard these coefficients while maintaining the overall flow performance of the reservoir. The process of discarding certain percentage of the small coefficients will be referred to the compression process in the subsequent paragraphs. For example, a process with 75% compression factor means after transforming the original distribution via wavelet transform, 75 % of the small coefficients are discarded and only the remaining 25 % is used in transformating back into the original space.

Figure presents the original permeability distribution and the permeability distribution after being compressed with 50%, 75%, and 95% compression factor. We can observe that the overall feature is retained in all cases. But, whether or not this is really the case, flow simulation is required to investigate the overall flow performance. The relative error of the bottom hole pressure as used in the previous section is also used in this analysis.

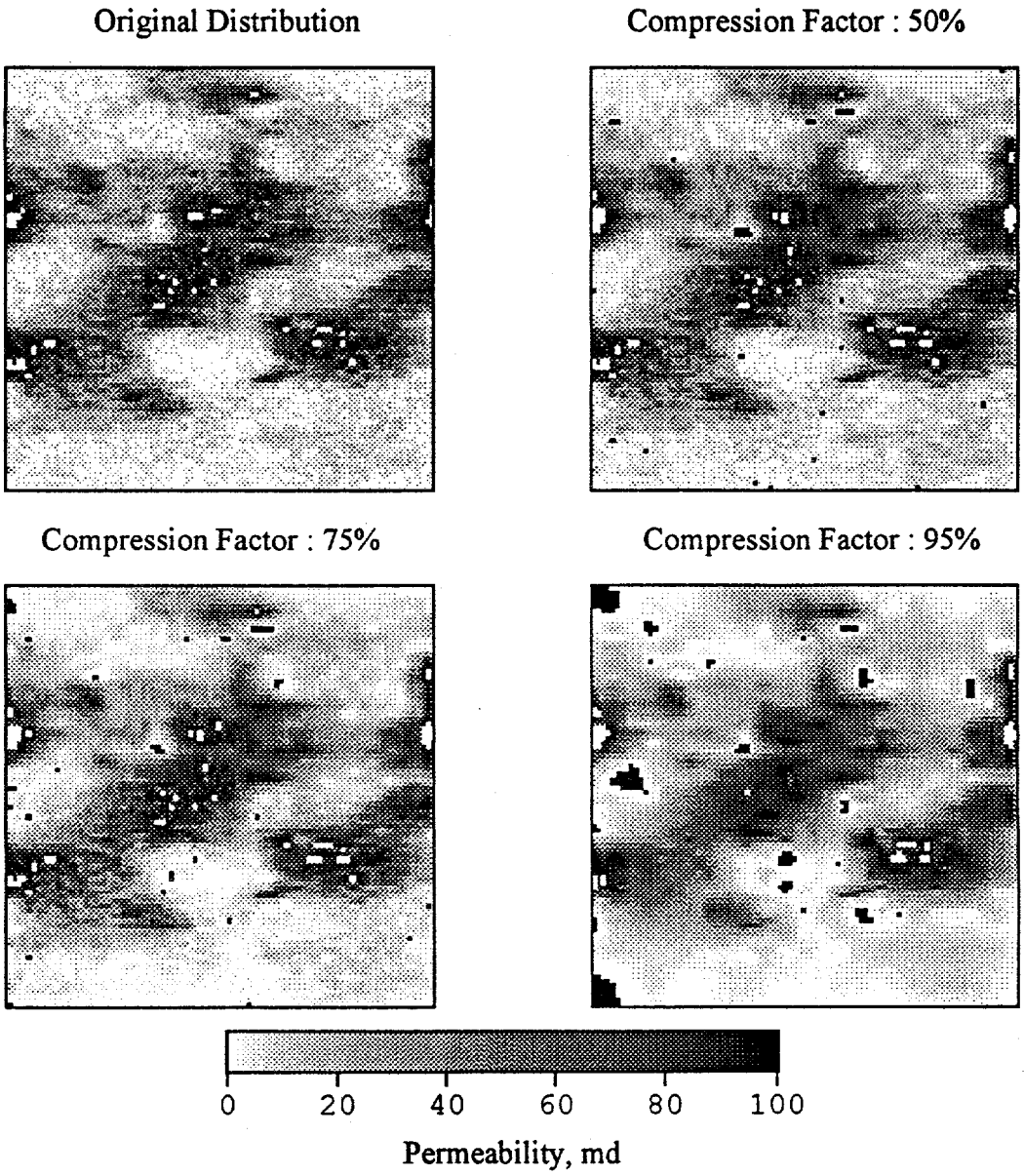


Figure 2.7: Permeability Distribution for Several Compression Factors

Figure show the results of the relative errors for the two phase flow model. From Figure A-C we can see that the absolute value of the relative errors are less than 8 % and are converged to zero at a certain time for the cases with compression factor of 25%, 50%, and 75% whereas from Figure D which is the case of 95% compression factor the relative errors become bigger and do not really converge to zero after long time. Thus, these results suggest that we can perform the compression process up to a factor of 75% and still maintaing the overall flow performance. Similar results are also observed from the single phase flow model.

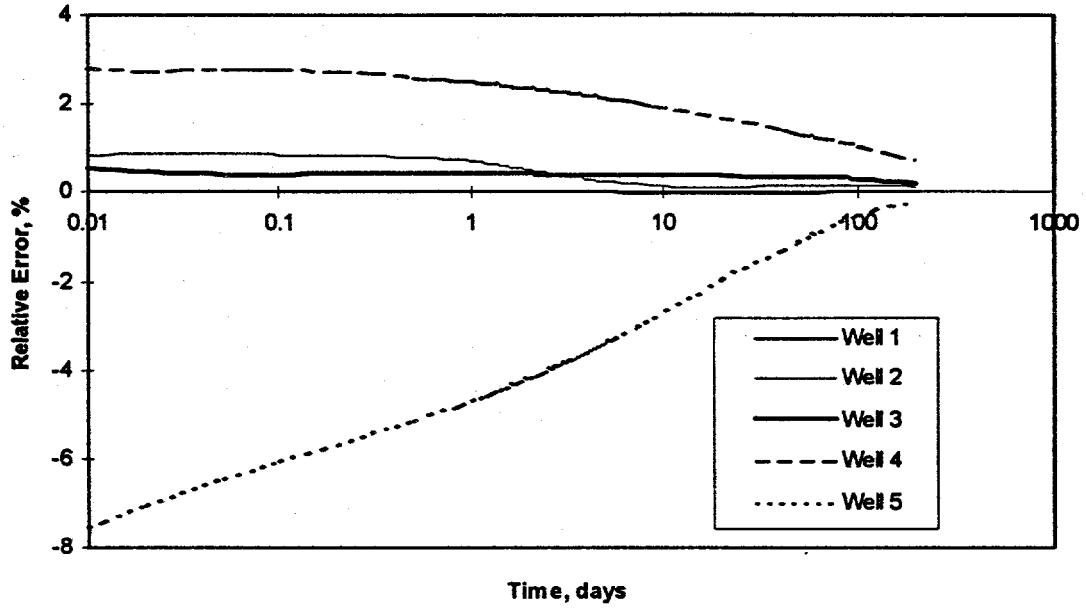


Figure A: Relative Errors for 25 % Compression Factor.

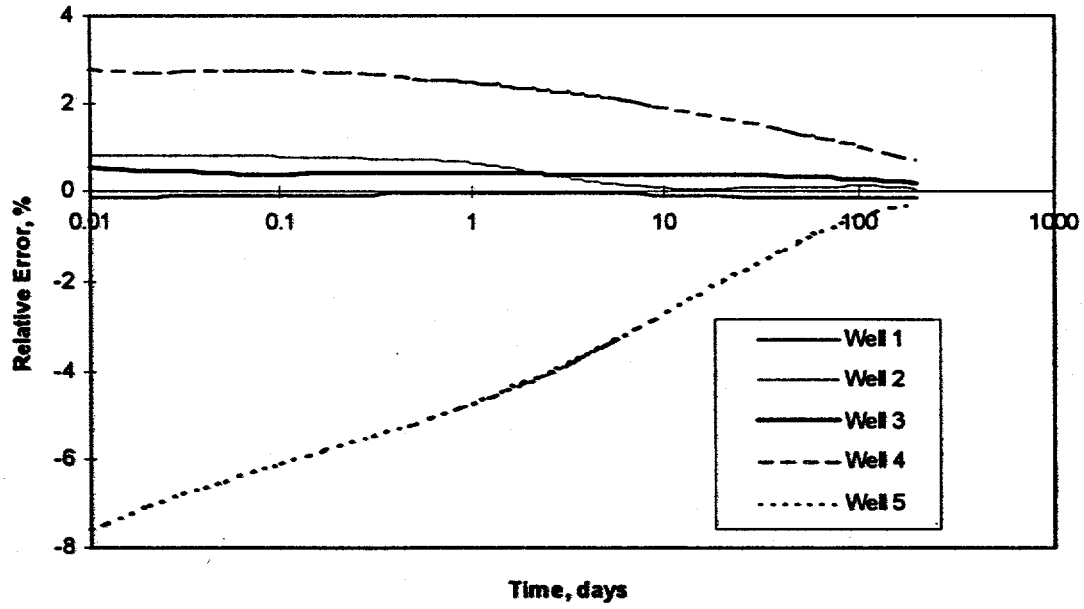


Figure 2.8B – Relative Errors for 50 % Compression Factor.

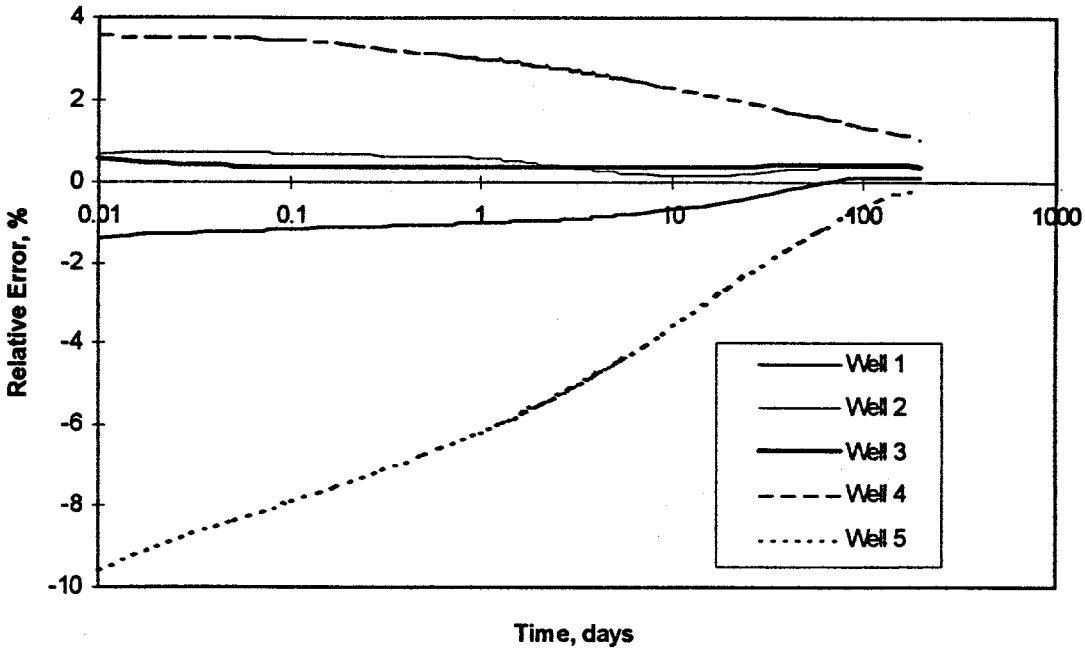


Figure C: Relative Errors for 75 % Compression Factor.

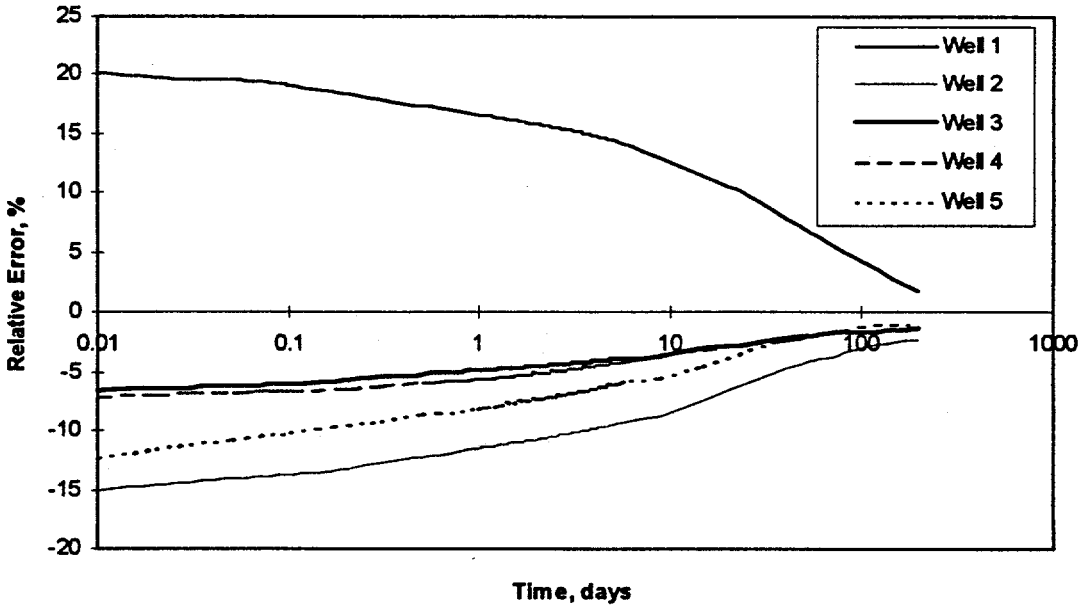
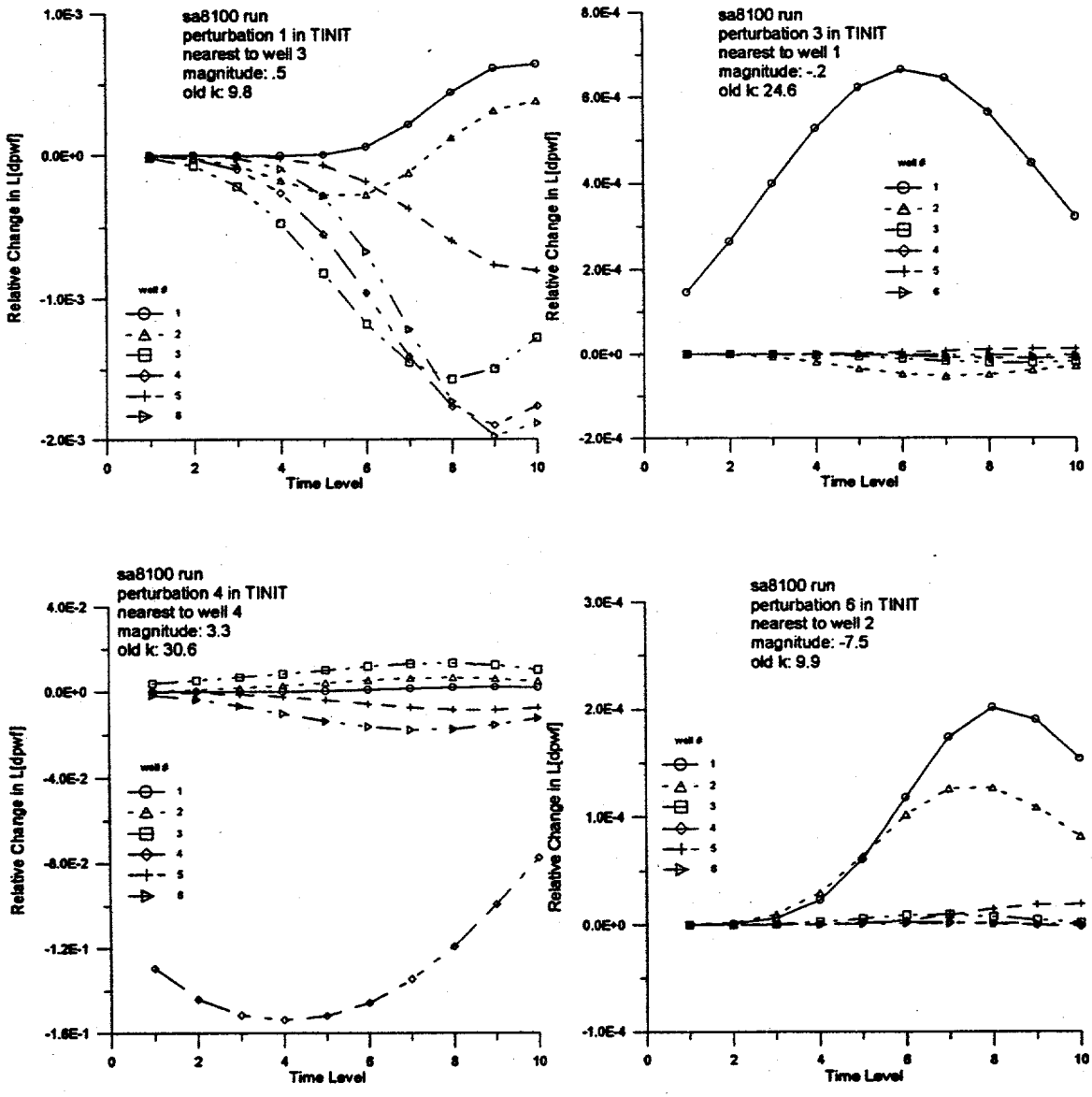


Figure D: Relative Errors for 75 % Compression Factor

Figure 2.8: Relative Errors for Compression Factors

2.3.6 Analysis of Single-Point Perturbations

This study was undertaken to get better insights into the behavior of the flowing bottomhole pressures of wells when a perturbation of the permeability field is made. We measured the changes in the pwfs of the wells (actually the Laplace transform of the change in the pwf) as a function of time for a perturbation at a specified location in the field. Some of these results are presented Figure and Table 2.4.



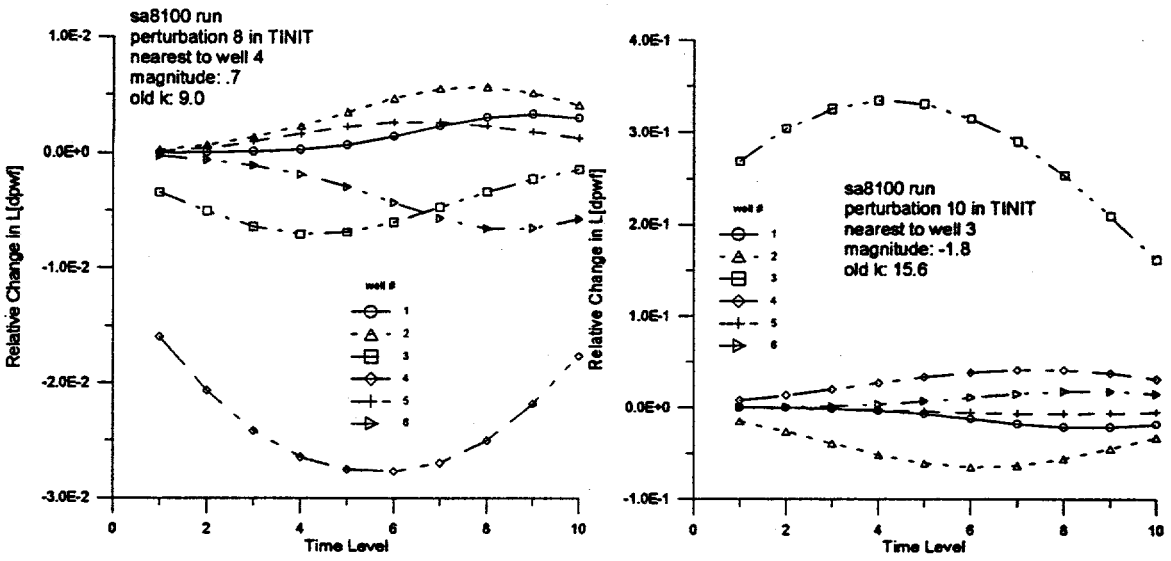


Figure 2.9: Time Behavior of Well Pressure Changes with Single-Point Perturbations

perturb #	distance to well 1	distance to well 2	distance to well 3	distance to well 4	distance to well 5	distance to well 6
1	3132	1820	1275	1601	2915	2358
2	3162	2000	1677	2121	3400	2915
3	559	1677	2574	3250	3500	4191
4	2704	1346	500	250	1458	1250
5	2693	1500	1250	1803	3010	2693
6	1768	1458	2016	2761	3579	3758
7	3717	3363	3260	2926	1581	2750
8	2574	1275	559	354	1250	1275
9	1275	2372	3132	3553	3132	4257
10	2151	791	250	791	1601	1768

Table 2.4: Locations of Perturbations Relative to Wells

The relative change is defined in a manner analogous to equation (4). From this analysis, we have empirically determined the sign of the change at the well nearest to the perturbation -- for the perturbations we studied -- is *always* correct as expected from Darcy's Law. That is, if the permeability at the wellblock is increased, then the pressure change at the well is negative and vice versa. This seems to suggest a possible strategy for manipulating the magnitude and direction of the perturbations. The idea is that by determining the location of the well nearest to a proposed perturbation, we can decide *beforehand* in which direction to perturb the permeability value at that point. This idea is yet to be tested.

2.4 Conclusions

1. Input data quality control is important to determine whether the appropriate quality and quantity of data is being used.
2. A procedure was developed to generate synthetic images which are used in testing the methodology.
3. The coefficient matrix code of the flow simulation program has been optimized.
4. A constant rate (as opposed to a variable rate) approach is the more appropriate option at this time.
5. The pre-conditioned conjugate gradient matrix solver is the method of choice for our application.
6. All of the upscaling approaches tested produced similar results on flow simulation of the upscaled grids.
7. Visual inspection suggests that the addition of the dynamic constraint to the objective function of the simulated annealing algorithm *does* improve the match between the training image and the generated image over that obtained when only a variogram objective function is used. However, an objective measure of this relative "goodness of fit", the *linear correlation coefficient*, performed inconsistently, giving a higher correlation for the model from the variogram-only objective function in some cases and for that from the composite objective function in others. This raises questions about the utility or suitability of that coefficient.
8. The simulated distributions were flow-simulated using the ECLIPSE simulator. Comparison of the pressure-time behavior showed a consistently better agreement with the training image by the distribution generated with a 2-part objective function over the distribution obtained from a variogram objective function.
9. The variogram pre-selection, 2-scale approach is a viable methodology which would produce satisfactory results if the problem of pressure matching between scales is resolved.
10. There is promise in some of the methodologies being studied for improving the upscaling approach to address the problem mentioned at 3 above.
11. A strategy using the well nearest to the perturbation to determine the magnitude and sign to use for the perturbation has been proposed.

3. 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 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 system in charge of 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

The log files used for training must undergo a scaling and normalization process to compensate for well-to-well variations in gamma ray values as well as variations in thickness. Also, the same shape may have different sizes which, again, requires some kind of normalization.

3.1 Advantages of Neural Networks

Neural networks are particularly well suited to solve the sand body recognition problem for several reasons. Among its most important features are:

- Human -brain-like processing
- Abilily to recognize patterns for systems such as:
    1. pattern recognition systems
    2. handwriting recognition systems
    3. classification systems

Models for neurons and neural networks are available extensively in related literature as well as several techniques to train them which have been thoroughly tested.

3.2 Neural Network Architecture

Despite all the knowledge available on neural networks, no standard procedure exists to determine an optimal network configuration to solve a given problem. Such procedures exist only for the most simple kind of problems which are known as linearly-separable problems requiring very simple structures (single layer configurations).

However some guidelines exist to solve more complex problems (non linear-separable problems) and they require multiple layer structures. Tests must be made to ensure that a network structure will be capable to solve the problem up to some level of confidence.

In order to solve the log facies identification problem the following three-layer structure showed to give acceptable results:

- Input Layer (16 neurons): every shape is discretized so it will consist of 16 points. Each one of these points is fed to a neuron in this layer. In consequence, the network is looking at a whole shape at any given time.
- Hidden Layer (100 neurons): this layer is used by the network to store what is known as the internal representation.
- Output Layer (5 neurons): this layer is the output of the network which is designed and trained to recognized five fundamental shapes. Each neuron in this layer correspond to a different shape (bell, funnel, blocky, symmetrical and linear)

The neural network is fully connected, that is every neuron in a layer is connected to every neuron in the following layer.

3.2.1 Supervised learning algorithm

A supervised training algorithm is applied to train the neural network. The algorithm is known as *error-back propagation algorithm*. The output of the network is compared against that of an expert and errors at the output layer are computed. Errors are then propagated from the output layer to the input layer and weights are updated. It is an iterative process. The network is considered to be trained when the maximum number of iterations is exceeded or when the error is below some value. The algorithm is based on the well known gradient-descent technique and it tries to minimize a quadratic error measure.

3.3 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.

To segment the well log, we use a rule based system that is structured as a two-step system:

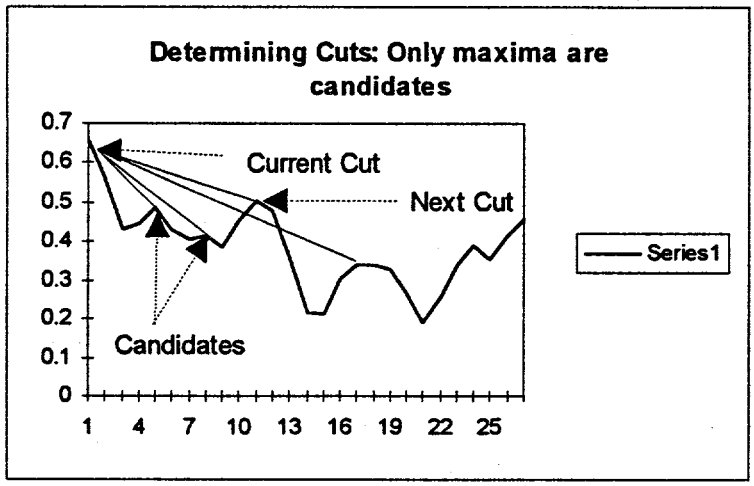
1. *Identify local maxima*
2. *Apply rules to obtain breaks*

The following rules are used by the system:

- Rule 1: a local maximum is a candidate break. This rule requires local maxima to be previously identified and is the reason for step 1 in our system.
- Rule 2: the local maximum following a candidate must not be seen from the previous identified break. By *not seen* we mean that the *candidate* is below the straight line joining the *previous identified break* and *the local maximum following the candidate*.

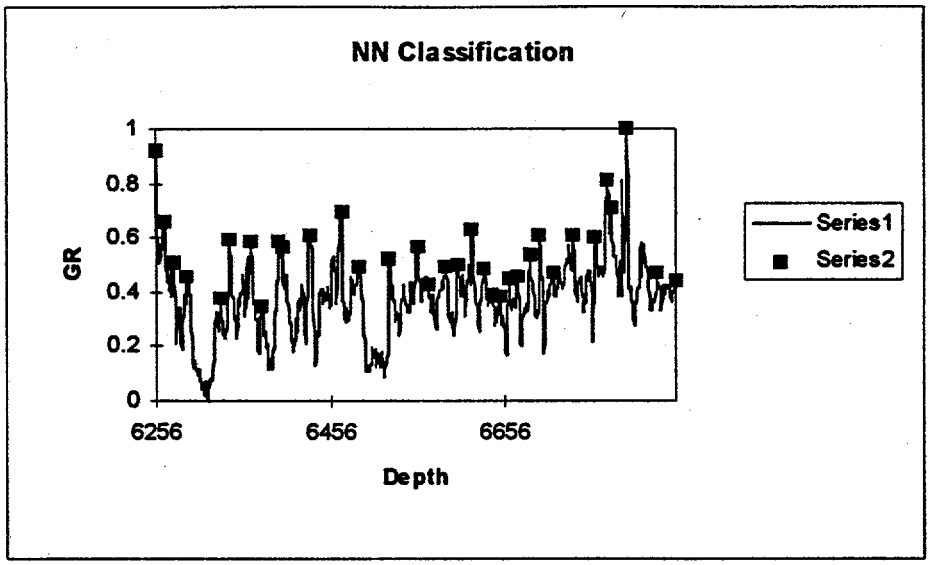
3.3.1 Applying Well Log Segmentation Rules

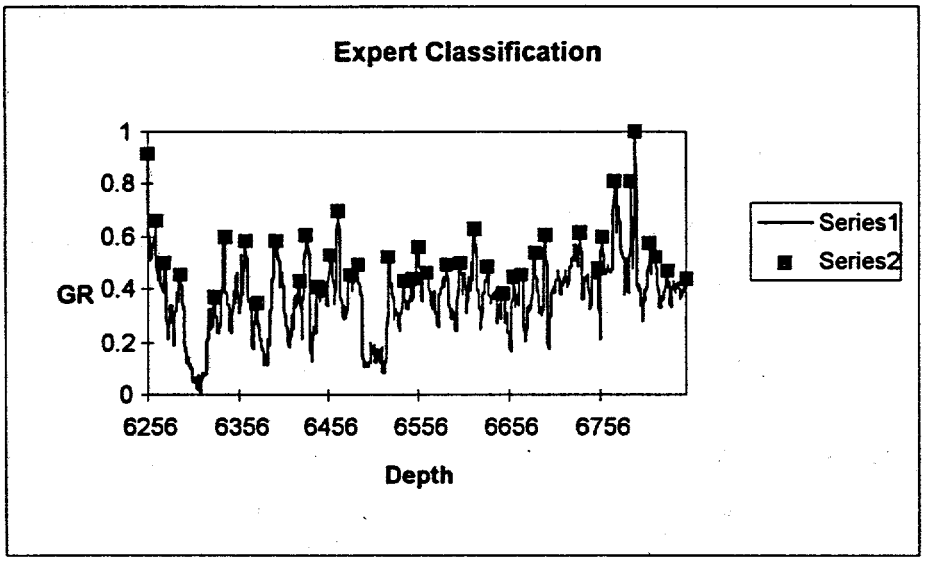
This figure shows how this rules are applied to a section of a log:



3.3.2 Segmentation Results

Figures comparing the neural network performance against expert classification for one testing log (gamma ray) is presented here. Squares are cuts.





The output of this system is then applied to the neural network.

3.4 Log Facies Identification

A neural network is also used to identify the fundamental shapes of sand bodies. The log facies must be normalized and discretized scaled so they can be fed to the network.

- Scale/normalize individual segments: a facie occurs between two consecutive breaks. All points between two consecutive breaks are scaled so:
    1. maximum value maps to 1
    2. minimum value maps to 0

- Scale/discretize depth values for each facie: the original data is linearly interpolated at 16 equally-spaced depth values and discretized accordingly. Points 1 and 16 are breaks.

3.4.1 Identification Results

Table 3.1 illustrates the interpretations obtained from the neural network as compared with the human expert for identifying the well log segments or cuts.

	Breaks (NN)	Breaks (Expert)	Percentage difference	Matching Breaks	Matching Breaks (%)
well6l	34	40	15.0	29	72.5
well6u	31	29	6.9	22	75.9
well7l	32	39	17.9	24	61.5
well7u	27	23	17.4	17	73.9
well8l	39	43	9.3	29	67.4
well8u	32	30	6.7	22	73.3
Total	**195**	**204**	**4.4**	**143**	**70.1**

Table 3.1

3.5 Neural Network Training and Testing

To identify the sand bodies, the neural network was trained using 11 different well logs with a 97.2% recall rate or correct classification when compared to that of the expert. The performance is described in Table 3.2 below.

Facies	Actual	Identified	Percentage
Bell	49	49	100.0
Funnel	54	54	100.0
Blocky	81	79	97.5
Symmetrical	80	80	100.0
Linear	25	19	76.0
Total	**289**	**281**	**97.2**

Table 3.2

4. Geological System: Marker Bed Identification

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

4.1 Introduction

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

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

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

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

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

be stored in the *database module*, which has yet to be developed. The user will be responsible for inputting the necessary information.

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

4.2 Sum Difference Equation

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

Because the sum difference equation is sensitive to the magnitude of variation in the data strings being compared, log data must be scaled and type logs for open- and cased-holes should be considered separately. It is common for logs to vary from well to well, and even within a well for different log runs. In cased-holes, the presence of casing and annulus cement can alter the log characteristics from that found in open holes. Therefore, if cased-hole logs are to be used, then a type log, or at least a known marker bed segment, must be provided by the user. Thus, each log needs to be first scaled in a similar fashion (Figure 4.1). Here, the gamma ray log data range (maximum value - minimum value) is arbitrarily scaled from 0 to 100 for each well.

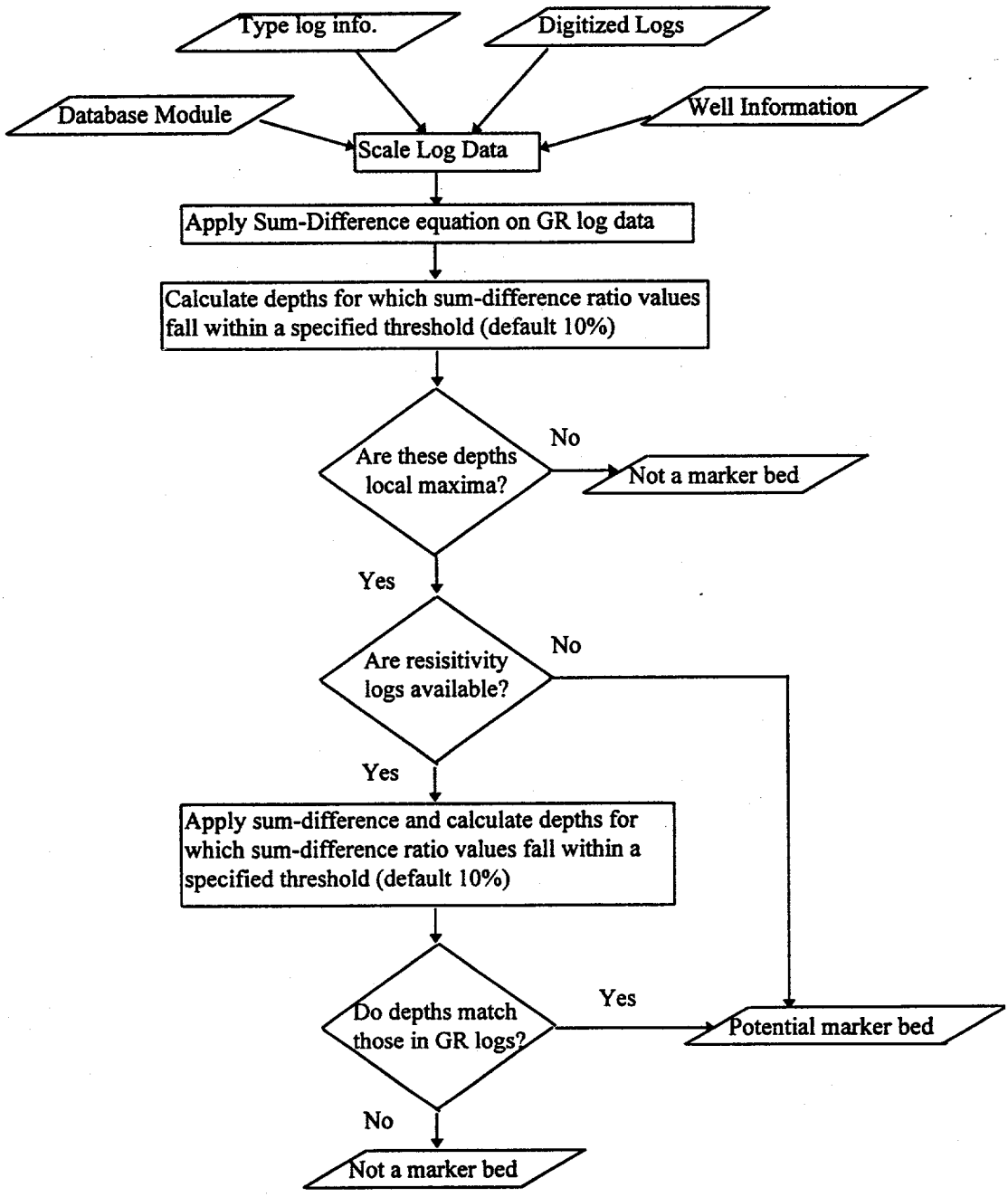


Figure 4.1: Determination of Marker Beds

The sum difference equation is next applied to gamma ray and resistivity logs separately (Figure 4.1). The sum difference equation evaluates the squared difference in log values between the type log interval of a known marker bed and the same log in offset wells. The difference is summed across the depth range of the marker bed interval, and scaled to the sum of the values in the marker bed interval (Figure 4.2). If the match is perfect, then the sum difference ratio is equal to zero. The stronger the mismatch, the greater the absolute value of the sum difference ratio.

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

where

L = gamma log for well under study

TL = type log for field under study

n = number of feet of marker bed in type log

Figure 4.2: Sum-Difference Equation

The marker bed interval is progressively moved along the log string under evaluation with the sum difference ratio calculated at each comparison position or match position. Thus, producing sum difference ratio values over an interval of match positions (Figure 4.3).

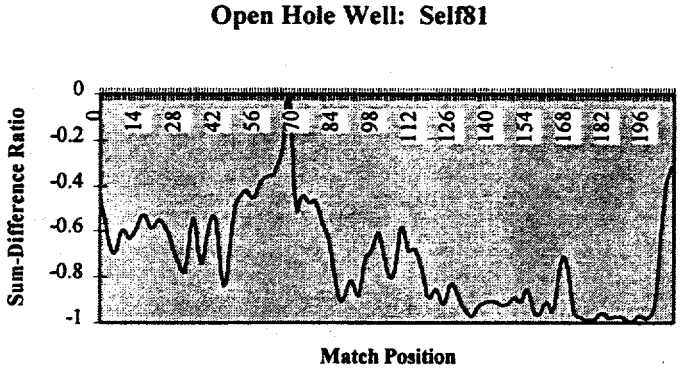


Figure 4.3

Ideally, the sum difference ratio is at or near zero at only one match position, thus, providing an unambiguous identification of a marker bed. This is, however, not always the case. The next step is to rank the sum difference ratio values falling close to zero that

meet some user-defined tolerance limit (Figure 4.1). With the tolerance the module can rank those sum difference ratio and their match positions.

An artifact of the sum difference algorithm is the shifting of the position of peaks. Thus, the module must return to the original log to identify the local maximum value corresponding to those match positions falling within the tolerance limit. If the selected depth does not correspond to a local maximum on gamma ray log, that selection is rejected as a marker bed. If the selected depth does correspond to a local maximum then it is considered a *potential marker bed* (Figure 4.1). Potential marker beds will be further evaluated within the marker bed module using techniques to be developed in the next reporting period.

Results of the sum difference algorithm are summarized in Table 4.1. Of the 13 well logs evaluated, the sum difference equation clearly ranked the known marker bed (Inola marker) as number 1, a success rate of 62%. With the use of the local maxima routine, the success rate is 69%. We would like the success rate to be better than 90%. In two of the cases (BG 11-86 and Self 78) a marker bed well above and well below the Glenn Sand were ranked ahead of the known marker bed (Table 4.1). These two failures should be picked up if additional heuristic rules are used to establish the depth relationship of marker beds to known reservoir intervals. With the application of such heuristic rules, the performance should improve to 92%.

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

Table 4.1: Results of sum-difference equation

In Figure 4.4, we present the type log used to detect marker beds, along with a well log in which the marker bed was found and a well log in which the marker bed was not ranked high enough in the list of potentials to be chosen. The reasons why the marker bed in well BG 11-89 was not successfully detected are twofold.

(i) There are very high gamma-ray peaks below the Glenn Sand which are being picked up as potential marker beds, thus causing a ranking higher than the Inola marker bed.

(ii) The Inola marker does not show a characteristic high gamma ray in this well.

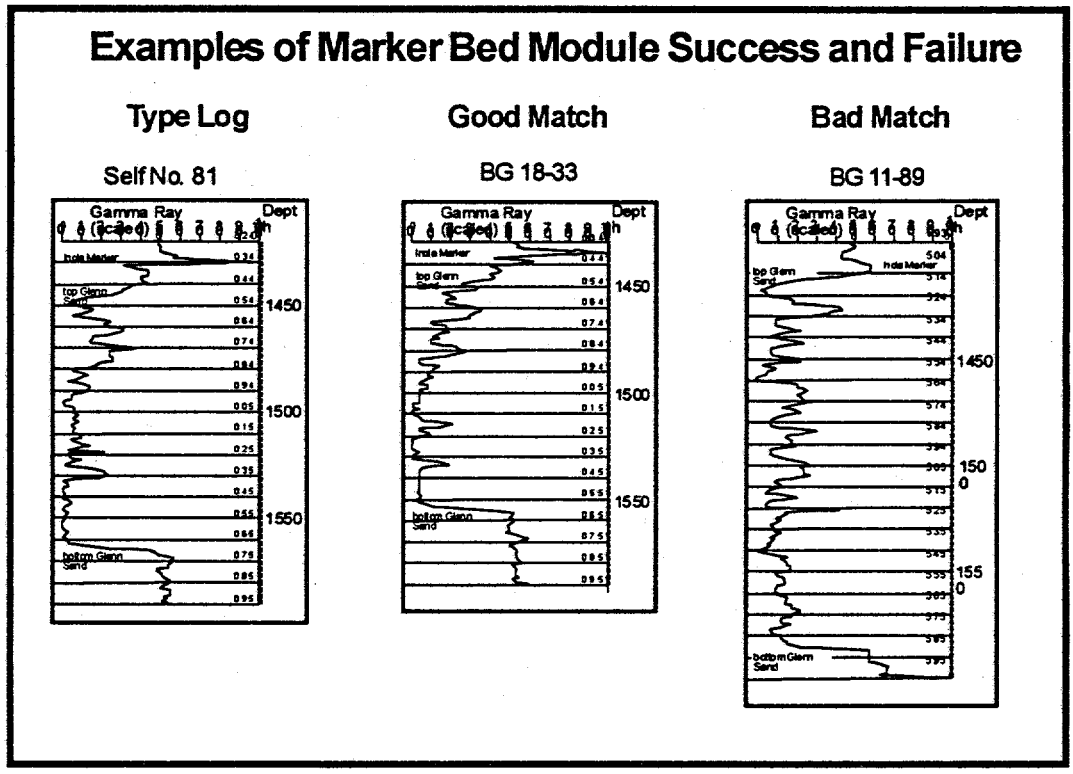


Figure 4.4: Example well logs for marker bed module

4.3 Future Work

The approach of the sum difference algorithm has provided a great deal of improvement in the recognition of a marker bed having wireline log characteristic of extreme log values. We are confident that with some additional heuristic rules, we could achieve identification of greater than 90%. Any logs in which a marker bed can not be resolved will have to either be sent to the user for identification, or not be used in the development of an architectural model.

In recognizing that not all marker beds are characterized by strong excursions in log values, a pattern recognition algorithm is to be considered for inclusion in the marker bed module. This algorithm will be developed from the same approach used in the automated recognition of a personal signature. We manually tested a simple set of attributes over the last reporting quarter, but with this set we were not satisfied that the set could sufficiently discriminate a known marker bed from other potential marker beds. Knowing that marker bed log patterns can have subtle variants, this algorithm will not likely work on its own, but must work in conjunction with the other algorithms contained in the module.

To satisfy the lateral continuity (correlation) of and with allowing for subtle variation in characteristic of marker beds a case-based reasoning approach will be employed. Case based reasoning suggests a model of reasoning that incorporates problem solving, understanding and learning. Case based reasoning can mean adapting old situations to meet new demands, using old cases to explain new situations or reasoning from precedents and interpret a new situation or create an equitable solution to a new problem.

5. Geological System Components: Correlation Program

The approach is to develop a set of rules for correlation of two log curves. 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. This approach is modified after Starzman and Kuo[20].

For correlating two zones from two different wells the following four criteria have been chosen:

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;

5.1 Position of the zones

The *position* of a zone indicates where this zone appears in the log. A zone can be in the *upper, middle* or *lower* section of a log trace. Determination of *position* is based on the ratio of two parameters: (1) length from top of the log trace to the middle of the zone and (2) total length of the log trace. If the ratio of the former to the latter is less than 1/3, then the *position* is given as *upper section.* If this ratio is greater than 2/3, then the *position* is given as *lower section.* Otherwise the position is *middle section.*

The rules for correlating two zones based on their position similarity can be specified as follows:

1. If the two zones being correlated have the same position (e.g. upper zone with upper zone, middle zone with middle zone, etc.) in respective logs, the position similarity is *same.*
2. If a zone is correlated with another zone that belongs to either one position higher or lower (e.g. upper zone with middle zone , lower zone with middle zone etc.) , in respective logs, then the position similarity is *close.*
3. Otherwise the position similarity is *apart.*

5.2 Distance from marker bed

The rules for correlating two zones based on their distance from marker bed can be specified as follows:

1. If $|d1-d2| <= x \tan\theta$, the zones are *same,*
2. If $x \tan\theta < |d1-d2| <= x \tan\theta$ + 10% of (d1+d2)/2, then the zones are *close,*
3. Otherwise, the zones are *apart,* where, d1 and d2 = distances of the mid points of the zones being correlated, x = horizontal distance between the two wells to which the zones belong, and θ = stratigraphic dip angle.

The user can be queried about the average stratigraphic dip angle or it can be set to a default value of 15 degrees.

5.3 Thickness of zones

The rules for correlating two zones based on their thickness can be specified as follows:

1. If $|t1-t2| <= x \tan\theta$, then there is *no difference,*
2. If $x \tan\theta < |t1-t2| <= x.\tan\theta$ + 10% of t1(where t1 < t2), then there is *small difference,*
3. Otherwise, there is *large difference,* where, t1 and t2 = thicknesses of the zones being correlated, x = horizontal distance between the

two wells to which the zones belong, and θ = stratigraphic dip angle.

5.4 Log-facies of zones

The rules for correlating two zones based on their log-facies can be specified as follows:

1. If the log facies of the two zones are same, then the two zones are *same*.
2. Otherwise, they are *different*.

5.5 Comprehensive Rule Set

With these basic rules in hand, we need to formulate a comprehensive set of rules for different combinations of above different situations. A tentative comprehensive rule set based on the above four criteria is given as Figure 4.5. This rule set is to be referred by the system every time two zones are compared and a correlation rank is established. Two zones are finally correlated with highest correlation rank.

Comprehensive Rules Rule no.	Criteria log facies	mbed_dist	rel_pos	thick_diff	Rank
1	same	same	same	no_diff	54
2	same	same	same	small_diff	53
3	same	same	same	large_diff	52
4	same	same	close	no_diff	51
5	same	same	close	small_diff	50
6	same	same	close	large_diff	49
7	same	close	same	no_diff	48
8	diff	same	same	no_diff	47
9	diff	same	same	small_diff	46
10	diff	same	same	large_diff	45
11	same	close	same	small_diff	44
12	same	close	same	large_diff	43
13	same	close	close	no_diff	42
14	same	close	close	small_diff	41
15	same	close	close	large_diff	40
16	diff	same	close	no_diff	39
17	diff	same	close	small_diff	38
18	diff	same	close	large_diff	37
19	same	same	apart	no_diff	36
20	same	same	apart	small_diff	35
21	same	same	apart	large_diff	34
22	same	close	apart	no_diff	33
23	same	close	apart	small_diff	32
24	same	close	apart	large_diff	31
25	same	far	same	no_diff	30
26	same	far	same	small_diff	29
27	same	far	same	large_diff	28
28	same	far	close	no_diff	27
29	same	far	close	small_diff	26
30	same	far	close	large_diff	25

31	same	far	apart	no_diff	24
32	same	far	apart	small_diff	23
33	same	far	apart	large_diff	22
34	diff	close	same	no_diff	21
35	diff	close	same	small_diff	20
36	diff	close	same	large_diff	19
37	diff	close	close	no_diff	18
38	diff	close	close	small_diff	17
39	diff	close	close	large_diff	16
40	diff	same	apart	no_diff	15
41	diff	same	apart	small_diff	14
42	diff	same	apart	large_diff	13
43	diff	close	apart	no_diff	12
44	diff	close	apart	small_diff	11
45	diff	close	apart	large_diff	10
46	diff	far	same	no_diff	9
47	diff	far	same	small_diff	8
48	diff	far	same	large_diff	7
49	diff	far	close	no_diff	6
50	diff	far	close	small_diff	5
51	diff	far	close	large_diff	4
52	diff	far	apart	no_diff	3
53	diff	far	apart	small_diff	2
54	diff	far	apart	large_diff	1

Figure 4.5: Rule Set for Correlation Ranking

5.6 Future Work

We need to test this approach first and we shall also need to consider additional. Presently we are only considering information regarding the two individual zones that are being correlated with each other without any data regarding the zones above and zones below them. In the future, we shall include some parameters about the neighboring zones also while correlating two individual zones in order to reinforce the correlation system.

6. Visualization of Architectural Elements

6.1 Introduction

Analysis and design for the project's visualization module has begun. The functions of this module is to display and possibly manipulate architectural elements (e.g., channel-fill facies) in a 3D data volume. Geologists have highly developed visualization capabilities. They much prefer to investigate a problem through the use of graphic illustrations, either in map view, cross section, or 3D perspective images. The purpose of the visualization module is to assist the geologist in this type of investigation.

Project members have visited exhibits at the Mid-Continent Section Meeting of the American Association of Petroleum Geologists (AAPG). We have also been in

contact with geologists working at Amoco's Tulsa Research Center. They have been very helpful, and they have suggested that we study a World Wide Web site that contains information about state-of-the-art geological visualization techniques. The web address is http://sepwww.stanford.edu/seg/research/3Dmodel /SALTHOME/segsalt.html. We have also consulted Yarus and Chambers[21]. It contains several papers, with color images, that have been helpful in selecting the appropriate graphics techniques for this project. The information that has been gathered is being evaluated.

6.2 Module Concept

At the present time, the initial capabilities of the module will include the visualization and manipulation of architectural elements in three dimensions. Wells will be represented as thin cylinders, or perhaps more simply as a line. After correlation is performed on the sand bodies, the module will display facies as simple primitive shapes based on a list of rules for the expected geometry of facies. For example, channel-fill facies are represented by sinuous, hemicylinders, and splay facies low-profile hemicones. The resolution of architectural elements will varying depending on the magnification of the graphic image. For example, individual lateral accretion bar deposits within a channel-fill could be represented at a sufficient magnification, or at higher magnification the lateral accretion surface drapes within a lateral accretion bar between wells.

As more information is processed, the expert system will refine the description of the geometry of an architectural element and the location of a given well within that element. For example, well test data can constrain the location of a well with respect to changes in permeability-thickness that could be regarded as a facies contact or some other architectural element boundary.

6.3 Graphics Techniques and Tools

Currently, we are favoring the use of polygonal models. That is, each three dimensional object will be represented as a polygonal mesh. A polygonal mesh is a set of polygons that form a closed surface in three dimensional space. The field of computer graphics has many techniques for rendering images from these type of models. These techniques generally fall under the title of scanline graphics. Scanline graphics techniques perform some type of operation at each pixel in the image to determine the appropriate color value for that pixel. One of the most popular of these operations is the z-buffer hidden surface removal algorithm. The z-buffer algorithm considers each polygon in the scene. It projects the polygon onto a two dimensional viewport, the pixel values in this viewport become the pixel values of the image. The z-buffer keeps track of the depth of the polygon that created the current color value at every pixel. If the polygon that is being considered is closer to the viewpoint than the one that created the current color value, then the color value of the current polygon replaces the previous color value.

The advantage of polygonal models and scanline graphics is that rendering, the process of creating an image from a geometric description or model, can be done quickly.

We are also evaluated a graphics library called OpenGL. This is a platform and operating system independent graphics library. It is designed to interface with C and C++. Therefore, it should integrate with the rest of the software from the project. OpenGL is provided as a part of Microsoft Window NT. The next version of Windows 95 is supposed to include support for OpenGL. A third party vendor, Template Graphics, will be releasing a Win32s version of OpenGL that will run under Windows NT and Windows 95 at the beginning of next year. This version is also going to have the capability to generate an executable module that will run under Windows 3.1. At the present time we are considering using Windows NT and a beta version of OpenGL for Windows 95.

6.4 Near Term Objectives

The next task that must be completed is a thorough specification of the interface between the expert system and the visualization module. A solid specification will accomplish two goals. It will enable us to finalize the selection of graphics techniques and tools. It will also allow work to be performed in parallel with the develop of the expert system.

In the near term we want to accomplish two things. First, we want to obtain a version of the OpenGL library and begin generating test images. Secondly, we want to develop the specification of the interface between the visualization tool and the expert system. At that point it will be time to assess the impact that the features of OpenGL and the specification have upon one another.

6. Well-Test Interpretation

Transient testing is a major source of vital information about reservoir parameters like permeability, reservoir pressure, wellbore conditions, reservoir discontinuities and other information that is essential for reservoir studies. The accuracy of these properties estimated from well tests depends on prior identification of a model that describes the reservoir accurately. This model is known as the 'well test interpretation model'.

The procedure for finding an appropriate model can be quite complex. It usually resides deep in the expert's mind. The failure of mathematical models to solve the problem can be attributed to the nature of the problem itself since the procedure is not completely quantitative and relies on experience. Computerized well testing involves interpreting the various forms of the time vs. pressure data and other well data and to determine the well model and calculate the various parameters. Basically this can be grouped into two parts: the qualitative analysis, and the quantitative analysis. Qualitative analysis deals with selecting the appropriate well model from the input data. Quantitative analysis involves calculating various well parameters like permeability, skin factor (if applicable) etc. Estimation of well properties depends upon the selection of the right model and hence qualitative analysis is a crucial part of the problem.

The expert system component developed for the well test interpretation consists of rules and facts for buildup test analysis gathered from an expert in this field. These carefully extracted rules simulate the reasoning process used by an expert to identify the appropriate interpretation model for a well test. In this approach, the system is designed to use a description of the shapes of the derivative plot. When needed, the system seeks information in addition to test data from relevant sources such as known reservoir and fluid properties, production statistics, well logs and geological data.

This section is divided into two parts. The first part discusses the procedures used for quantative identification of reservoir paramenters using non-linear regression methods. The second part presents the well test interpretation package.

6.1 Non-linear Regression Analysis

6.1.1 Introduction

One of the key requirements in creating an integrated reservoir description is to use well test data to constrain geological information. As part of the process, it is critical that we are able to analyze the well test pressure data to determine various reservoir paramenters. The automatic determination of the reservoir parameters will allow us to compare these paramaters with the geological model in terms of resrvoir boundaries. By integrating two pieces of information, we will be able to define the reservoir parameters in a consistent manner.

This part of the report presents a new method for imaging the objective function across all boundaries imposed to satisfy physical constraints on the parameters. The algorithm is extremely simple and reliable. The method uses an equivalent unconstrained objective function to impose the physical constraints required in the original problem. Thus, it can be used with standard unconstrained least squares software without reprogramming and provides a viable alternative to penalty functions for imposing constraints when estimating well and reservoir parameters from pressure transient data.

In addition, two new methods are presented for implementing the penalty function approach for imposing parameter constraints in a general unconstrained least squares algorithm. Comparing the methods, the new imaging method always converges to a feasible solution in less time than the penalty function methods.

6.1.2 Background

In recent years, well test parameter estimation using nonlinear regression techniques[22-28] has become a popular and widely used tool for analyzing pressure transient data. Assuming that a reservoir/wellbore model has been selected, the basic idea behind nonlinear regression analysis is the determination of the set of model parameters, α, such that measured and model pressure transient data (e.g., pressure, pressure derivative or pressure/pressure derivative ratio) match each other as closely as possible. Commonly used approaches to ensuring a close match between measured and model data include minimization of the sum of the squares of the residuals (i.e., the difference between corresponding measured and model data points) and minimization of the sum of the absolute values of the residuals. The former approach is referred to as LS regression, while the latter is denoted LAV regression. Carvalho, *et al.*[28], describes in detail practical methods of applying widely available unconstrained least-squares algorithms such as LMDER[29] to LS or LAV regression analysis for determination of well test parameters.

A major difficulty in applying unconstrained regression algorithms to analyze practical well test data is the possibility of the selection of physically unreasonable parameter estimates (e.g., negative permeabilities) during the course of searching for the optimum set of model parameters. To overcome this problem, three new algorithms are presented: two implementations of the widely-used penalty function approach suitable for easy use in unconstrained algorithms, and a new robust imaging extension procedure. The latter approach offers the advantage of always yielding physically reasonable parameter estimates, regardless of the initial guesses for the parameters.

6.1.3 Handling of Parameter Constraints

a) Penalty Functions

In an unconstrained optimization algorithm, it is possible that non-physical values of the parameters may be selected in the process of minimizing the objective function. In order to keep the parameters physically meaningful, some type of constraint should be used. Abbaszadeh and Kamal[22] and Rosa and Horne[23] employ *penalty functions* which transform the constrained problem into an unconstrained problem by adding terms (penalty functions) to the objective function. The penalty functions (one for each

constraint) are defined in such a way that the new objective function remains almost unchanged in the interior of the feasible region of the parameters, but increases rapidly as a parameter approaches the boundary. Thus, if we constrain each parameter α_i, i = 1, 2,...p, to lie between some minimum value and maximum value, i.e., $\alpha_{\min,i} < \alpha_i < \alpha_{\max,i}$, then the new unconstrained objective function, $E^+(\alpha)$, can be written as[1]

$$E^+(\alpha^k) = E(\alpha^k) + F_c \sum_{i=1}^{p} \left[\frac{\delta_{\min,i}}{\alpha_i^k - \alpha_{\min,i}} + \frac{\delta_{\max,i}}{\alpha_{\max,i} - \alpha_i^k} \right], \qquad (1)$$

where α_i^k is the current value of parameter α_i, i.e., the estimate of α_i at the kth iteration, and $E(\alpha^k)$ is the original unconstrained objective function. F_c, $\delta_{\min,i}$ and $\delta_{\max,i}$ are positive values that govern the amount of contribution from the penalty functions. The superscript k refers to the current iteration. One choice for a set of values F_c, $\delta_{\min,i}$ and $\delta_{\max,i}$ is, respectively[1],

$$F_c = \varepsilon\, E(\alpha^{k-1}), \qquad (2)$$

where ε is a small number, e.g., $\varepsilon = 10^{-4}$,

$$\delta_{\min,i} = \alpha_{0i} - \alpha_{\min,i}, \qquad (3)$$

and

$$\delta_{\max,i} = \alpha_{\max,i} - \alpha_{0i}, \qquad (4)$$

where α_{0i} is the initial estimate of parameter α_i. Note that, in this formulation, $\delta_{\min,i}$ and $\delta_{\max,i}$ are computed only at the beginning of the regression, but F_c changes at each iteration.

Rosa and Horne[23] shows that the use of penalty functions improves the convergence of the iterative procedure; however, they also report that the penalty function method may not avoid non-physical parameter values when the initial estimates are far from the solution.

To overcome the non-physical values problem, two methods are offered for implementing the penalty function method. In the first method, (Method 1), we apply Eq. (1) with the following modification to the definition of F_c in Eq. (2):

$$F_c = \varepsilon\, E(\alpha^k); \qquad (5)$$

that is, we write F_c in terms of the objective function at the current parameter estimates rather than at the previous estimate. This change is important because now the modified objective function of Eq. (1) can be written as

$$E^+(\alpha^k) = \left(\beta(\alpha^k)\right)^2 E(\alpha^k), \qquad (6)$$

where,

$$\beta(\alpha^k) = \sqrt{1 + \varepsilon \sum_{i=1}^{p} \left[\frac{\delta_{\min,i}}{\alpha_i^k - \alpha_{\min,i}} + \frac{\delta_{\max,i}}{\alpha_{\max,i} - \alpha_i^k} \right]}. \qquad (7)$$

Suppose we were performing a least squares regression on pressure, so that our original objective function $E(\alpha^k)$ was given by

$$E(\alpha^k) = \sum_{i=1}^{n} \left(p_i - p_{model,i}(\alpha^k) \right)^2 = \sum_{i=1}^{n} r_i^2, \qquad (8)$$

where r_i denotes the ith residual; then our modified objective function incorporating the penalty functions is given by:

$$E^+(\alpha^k) = \sum_{i=1}^{n} \rho_i^2 ; \quad \text{.....(9)}$$

where

$$\rho_i = \beta(\alpha^k) r_i ; \quad \text{.....(10)}$$

that is, the penalty function approach can be incorporated by a simple modification of the definition of each residual. Note that when $\varepsilon = 0$, $\beta(\alpha^k) = 1$, and the modified objective function of Eq. (9) reverts to the original objective function given in Eq. (8). Also, the partial derivative of the modified ith residual (Eq. (10)) with respect to any model parameter α_j is given by the following simple relationship:

$$\frac{\partial \rho_i}{\partial \alpha_j} = \beta(\alpha^k) \frac{\partial r_i}{\partial \alpha_j} + r_i \frac{\partial \beta(\alpha^k)}{\partial \alpha_j} ; \quad \text{.....(11)}$$

where, from Eq. (7),

$$\frac{\partial \beta(\alpha^k)}{\partial \alpha_j} = \frac{\varepsilon}{2\,\beta(\alpha^k)} \left[\frac{\delta_{\max,j}}{\left(\alpha_{\max,j} - \alpha_j^k\right)^2} - \frac{\delta_{\min,i}}{\left(\alpha_j^k - \alpha_{\min,j}\right)^2} \right] . \quad \text{.....(12)}$$

Again, note that when $\varepsilon = 0$, the modified objective function and its partial derivatives revert to the original unconstrained objective function. Also, using the approach outlined above, penalty functions can just as easily be applied to LAV regression; in Carvalho, *et al.*[28], we showed that LAV regression can be closely approximated by a least squares optimization problem by a simple redefinition of the residuals. Thus, for an LAV regression on pressure, the ith residual in Eq. (8) assumes the form

$$r_i = \left(\left(p_i - p_{model,i}\right)^2 + \zeta \right)^{1/4} . \quad \text{.....(13)}$$

where ζ is a small positive number. The procedure for constraining the LAV parameter estimates using penalty functions becomes a special case of penalty-function-constrained LS regression.

In the second penalty function method, (Method 2), we incorporate Eq. 1 directly into the LS objective function as follows. Define a new penalty function $\Pi(\alpha)$ as

$$\Pi(\alpha) = \frac{F_c}{n} \sum_{i=1}^{p} \left[\frac{\delta_{\min,i}}{\alpha_i^k - \alpha_{\min,i}} + \frac{\delta_{\max,i}}{\alpha_{\max,i} - \alpha_i^k} \right] , \quad \text{.....(14)}$$

where n is the number of measured data points. Using Eqs. 8 and 14, we can write Eq. 1 as

$$E^+(\alpha) = \sum_{i=1}^{n} \left(r_i^2 + \Pi(\alpha) \right) = \sum_{i=1}^{n} \tilde{\rho}_i^2 , \quad \text{.....(15)}$$

where

$$\tilde{\rho}_i = \sqrt{r_i^2 + \Pi(\alpha)} . \quad \text{.....(16)}$$

The partial derivative of the modified residual defined in Eq. 16 with respect to a model parameter α_i, is given by

$$\frac{\partial \beta_i}{\partial \alpha_i} = \frac{1}{2\,\beta_i}\left(2\,r_i \frac{\partial r_i}{\partial \alpha_i} + \frac{\partial\, \Pi(\alpha)}{\partial \alpha_i}\right), \quad \text{.....(17)}$$

where

$$\frac{\partial \Pi(\alpha)}{\partial \alpha_i} = \frac{F_c}{n}\left(\frac{\delta_{\max,\,i}}{\left(\alpha_{\max,\,i} - \alpha_i\right)^2} \frac{\delta_{\min,\,i}}{\left(\alpha_i - \alpha_{\min,\,i}\right)^2}\right) + \frac{1}{F_c}\frac{\partial F_c}{\partial \alpha_i}\Pi(\alpha), \quad (18)$$

Equations 14 - 18 are general; F_c may be any suitable user-specified function. In the case where F_c is independent of the model parameters, the last term in Eq. 18 disappears. As with the first method, the second method is completely general: it can be applied to any LS problem.

In the application of Method 2, we define F_c as follows. Initially, we set F_c to a large number; i.e., $F_c = 10^{10}\,\varepsilon$. Thereafter, we set $F_c = \min(F_c, E^+(\alpha^{k-1}))\varepsilon$. (This ensures that F_c decreases as we approach convergence.) For both penalty function methods, we set $\varepsilon = 10^{-4}$.

b) Imaging Extension Procedure

As an alternative approach for keeping the parameters physically meaningful, a new constraining algorithm which is referred to as *the imaging extension procedure* is devised. The idea behind the method is to extend the constrained objective function in such a way that the new objective function is defined everywhere (except at the boundaries) and where the solution of this new unconstrained problem is equivalent to the solution of the original constrained problem.

Let us illustrate the proposed method with a model of the form, $F(\alpha,t)$, where α is the single parameter of the model (say permeability, k) and t is the independent variable (say time, t). Since permeability can only assume positive values, the original problem should be constrained through the inequality $k > 0$. If we reflect the objective function through the ordinate axis, the new objective function is now defined for positive values of k as well as for negative values of k (see Figure 6.1). Clearly, if k^* minimizes the new objective function, then $|k^*|$ will also minimize the original problem. Thus, we choose to minimize the modified unconstrained problem. Sometimes, we also refer to the modified problem as an *equivalent problem*. We also use E^+ to denote the objective function for this equivalent problem.

An algorithm to handle a single constraint of the form, $\alpha > \alpha_{\min}$, can be written as

1) Is the current parameter value of the equivalent problem, α_{eq}^c, greater than $\alpha_{\min}$? If YES, compute the objective function and the parameter gradient in the usual way. If NO, GO TO step 2.
2) Is α_{eq}^c equal to $\alpha_{\min}$? If YES, STOP. If NO, compute the corresponding parameter value of the original problem as

$$\alpha^c = \left|\alpha_{eq}^c - \alpha_{\min}\right| + \alpha_{\min} \quad \text{.....(19)}$$

3) Compute the current value of the unconstrained objective function, E^+, as

$$E^+\left(\alpha_{eq}^c\right) = E\left(\alpha^c\right) \text{.....(20)}$$

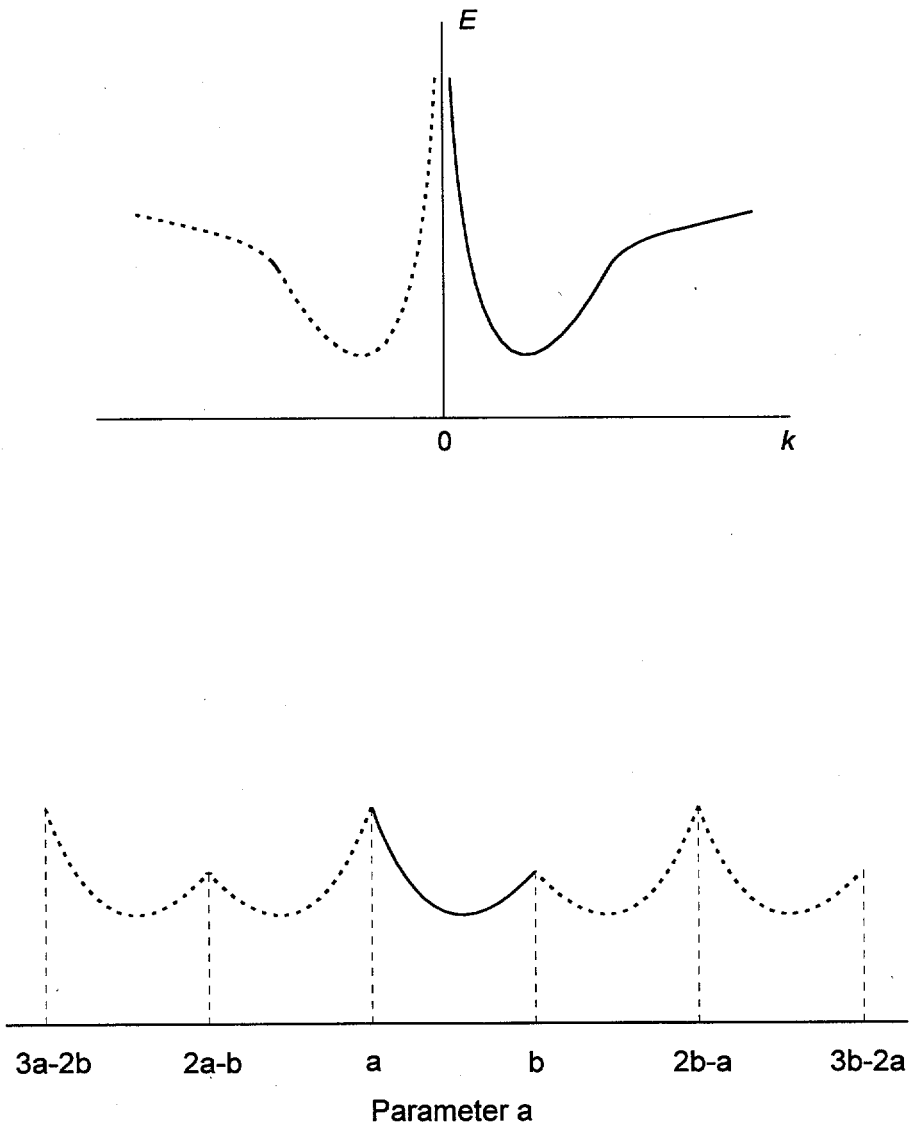
4) Compute the value of the gradient of the unconstrained objective function at α_{eq}^{c} as

$$\left.\frac{\partial E^{+}(\alpha)}{\partial \alpha}\right|_{\alpha_{eq}^{c}} = \left.\frac{-\partial E(\alpha)}{\partial \alpha}\right|_{\alpha^{c}} \quad \text{.....(21)}$$

Once a minimizer, α_{eq}^{*}, of the equivalent unconstrained problem is found, the corresponding minimizer of the original problem is computed by using Eq. 19. The above algorithm can be easily extended for problems with more than one parameter subject to the same type of constraint $\alpha_i > \alpha_{\min,i}$. As shown in detail in Carvalho[30], many applications of the proposed method for different functional models (e.g., horizontal well and double-porosity models) indicate that the algorithm works effectively for well test applications.

The imaging procedure can be generalized to handle constraints of the type $\alpha_{\min,i} < \alpha_i < \alpha_{\max,i}$. In this case, the objective function is periodically repeated by a sequence of images that extend to $\pm\infty$ (see Figure 6.1, top). The continuous curve in Figure 6.1 (bottom) is the original objective function, which is defined only in the interval (a,b), and the dashed curves represent the images. The minimizer of the original problem, α^{*}, can be computed once the minimizer α_{eq}^{*} of the extended objective function is found. Thus, first we apply the iterative minimization procedure to the modified (unconstrained) objective function, and then we find the corresponding solution to the original problem which is geometrically related to the solution of the modified (unconstrained) problem. Note that, by construction, at the boundaries of each image, the new (unconstrained) objective function, E^{+}, is continuous; however, its partial derivative with respect to the parameter that is at the boundary may not exist. In fact, its partial derivative exists (and is zero) at a "boundary" only when the corresponding derivative of the original objective function is zero, i.e., when

$$\left.\frac{\partial E(\alpha)}{\partial \alpha}\right|_{\text{boundary}},$$



Figures 6.1 (top) Illustration of the imaging of an objective function and (bottom) Illustration of the objective function for a constraint of a < b.

at the corresponding boundary, is zero. In the event a boundary is hit during the iterative process, we assume, for that iteration, that the partial derivative with respect to the parameter at the boundary is zero.

An algorithm to handle constraints of the form,

$$\alpha_{\min,i} < \alpha < \alpha_{\max,i},$$

can be written as

1) Set the parameter index, $i = 1$.

2) For the current value of the ith parameter of the equivalent problem, $\alpha^c_{eq,i}$, compute α^c_i via,

$$\alpha^c_i = \left|\alpha^c_{eq,i} - \alpha_{\min,i}\right| + \alpha_{\min,i} \quad \text{..... (22)}$$

3) Is $\alpha^c_{eq,i}$ greater than $\alpha_{\min,i}$? If YES, set $SIGN_i = 1$. If NO, set $SIGN_i = -1$.

4) Compute

$$\alpha^c_i = \alpha^c_i - 2\left(\alpha_{\max,i} - \alpha_{\min,i}\right) INT\left[\frac{\alpha^c_i - \alpha_{\min,i}}{2\left(\alpha_{\max,i} - \alpha_{\min,i}\right)}\right], \quad \text{..... (23)}$$

where $INT(x)$ is the largest integer less than or equal to x.

5) Is α^c_i (given by Eq. 23) greater than $\alpha_{\max,i}$? If NO, GO TO step 6. If YES, set $SIGN_i = -SIGN_i$, and compute

$$\alpha^c_i = \alpha_{\max,i} - \left(\alpha^c_i - \alpha_{\max,i}\right) \quad \text{..... (24)}$$

6) Is α^c_i equal to $\alpha_{\min,i}$ or $\alpha_{\max,i}$? If YES, set $SIGN_i = 0$.

7) Repeat steps 2 to 6 for all parameters, i.e., for $i = 1, 2, \ldots p$.

8) Compute the current value of the unconstrained objective function, E^+, as

$$E^+\left(\alpha^c_{eq}\right) = E\left(\alpha^c\right) \quad \text{..... (25)}$$

9) Compute the value of the gradients of the unconstrained objective function at α^c_{eq} as

$$\left.\frac{\partial E^+\left(\alpha_{eq}\right)}{\partial \alpha_{eq,i}}\right|_{\alpha^c_{eq}} = SIGN_i \left.\frac{\partial E(\alpha)}{\partial \alpha_i}\right|_{\alpha^c} \quad \text{for } i = 1, \ldots, p. \qquad (26)$$

In step 2, the current parameter estimate $\alpha^c_{eq,i}$ is reflected through the left boundary of the original objective function (i.e., the boundary $\alpha_i = \alpha_{\min,i}$) when $\alpha_{eq} < \alpha_{\min,i}$. Note that every time a reflection occurs the variable *SIGN* (which controls the sign of the derivative) changes sign. In step 4, the estimate is transformed to lie in the first repetitive element of the new objective function (which consists of the original objective function and its right reflection), without changing either the value of the objective function or the value of its partial derivative. In step 5, if the estimate corresponds to the reflection of the original objective function, it is transformed (i.e., reflected through the boundary at $\alpha_{\max,i}$) to obtain an α^c_i value within the feasible interval. At the end, $SIGN = 1$ means that, in terms of the unconstrained problem, we are in a replica of the original objective function, and $SIGN = -1$ means that we are in a replica of the reflection. When a minimizer, α^*_{eq}, of the equivalent unconstrained problem is found, the corresponding minimizer of the original problem is given by the coordinates of α^c (Eqs. 23 or 24) at the last iteration.

The proposed method is illustrated with a dual-porosity example where the parameter ω, storativity ratio is regressed. By definition, the feasible interval for the storativity ratio is $0 \leq \omega \leq 1$. The minimization procedure (with our constraining algorithm) was applied to synthetic pressure data which was generated with $\omega = 0.10$. Figure 6.3 presents the changes in the parameter ω during the iterative process, for five different initial estimates, $\omega = 0.9$, 0.70, 0.50, 0.25 and 0.05, respectively. In these figures, the abscissa represents ω and the ordinate axis represents the square root of the

objective function, so one can see how the objective function decreases from iterate to iterate.

The solid continuous lines in Figure 6.3 refer to the path of iterates for the equivalent (unconstrained) problem, and the dashed lines refer to the equivalent path for the original problem. All five regressions converged to the right solution, i.e., $\omega = 0.1$. For example, Figure 6.3 (top left) shows that for the initial estimate $\omega = 0.95$, the regression applied to the equivalent problem converged to $\omega_{eq} = -1.90$ (i.e., the algorithm converged to a minimum in the second image to the left), which corresponds to $\omega = 0.1$. Note that for the two cases where the initial estimates were closer to the solution, i.e., $\omega = 0.25$ and 0.05 (Figure 6.3, top right and Figure 6.4, top left, respectively), the constraining algorithm was not necessary since ω always remained in the feasible region.

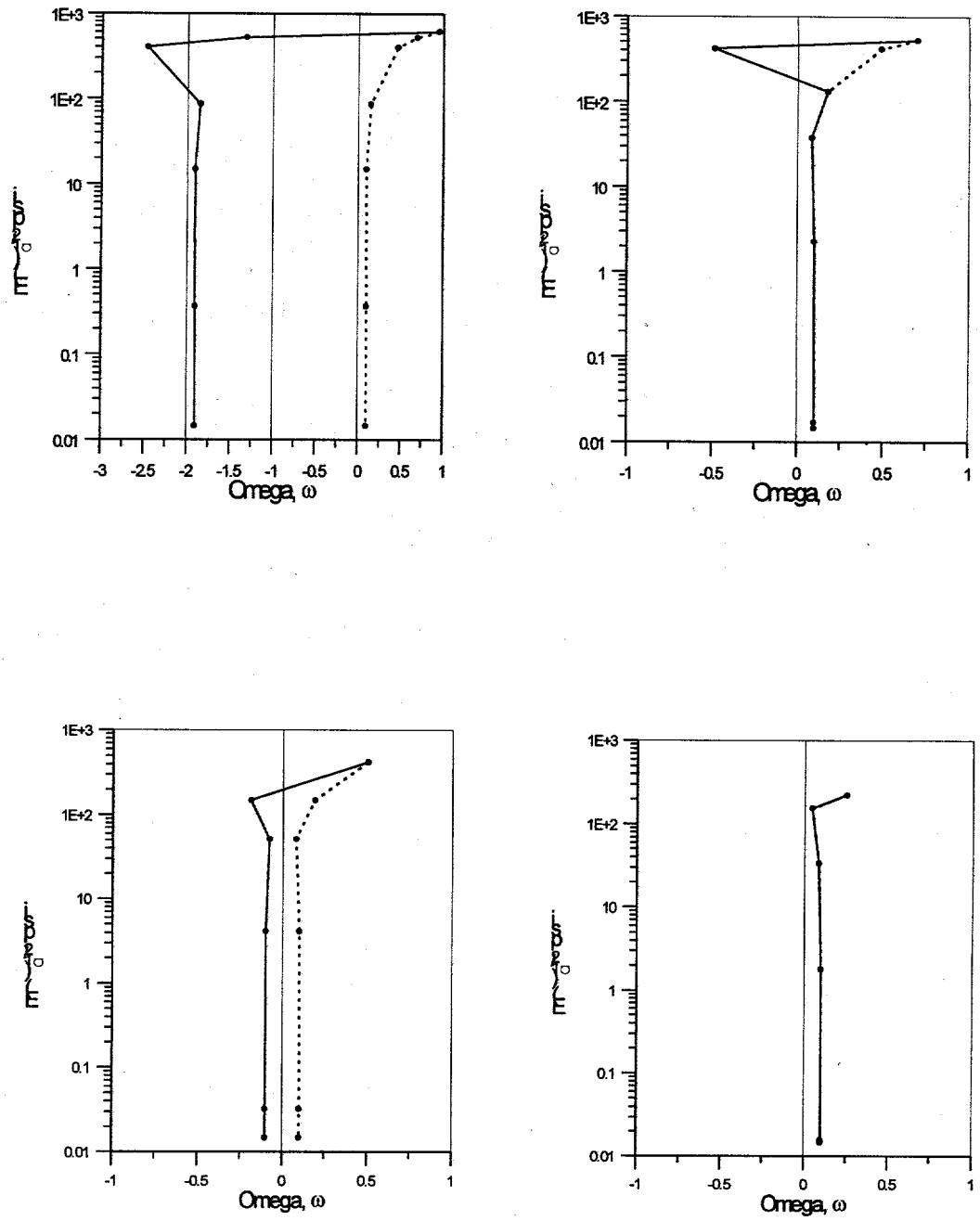


Figure 6.3: Change in the estimate of omega during the iterative process. Initial estimates are as follows - (top left) omega = 0.95, (top right) omega = 0.70, (bottom left) omega = 0.50, and (bottom right) omega = 0.25.

6.1.4 Comparison of Constraining Algorithms

In order to compare the performance of the presented constraining algorithms, we applied them to the set of synthetic dual porosity drawdown pressure data shown in Figure 6.4 (top right). The input model data used to generate Figure 6.4 are presented in Table 6.1, while Table 6.2 presents the initial estimates of the parameters we were trying to optimize using LS regression, along with the applied constraints.

All of the methods successfully converged to the correct answer. However, the penalty function methods required the extra constraint of returning large residual values when parameter estimates (λ and C for this problem) became negative during the first few iterations. The remaining parameters always remained physically meaningful. The imaging extension procedure is of course immune to the problem of non-physical parameter values, and, in fact, converged to an unconstrained estimate of λ that was the negative of the actual value.

Figure 6.4 shows how each modified objective function behaved at each step of the optimization process. This reflects our experience with working with the imaging extension procedure versus the penalty function methods. The imaging extension procedure converges more rapidly than penalty function methods. Figure 6.4 shows the penalty function behavior ($\beta(\alpha) - 1$) (Eq. 7) and $\Pi(\alpha)$, (Eq. 14).

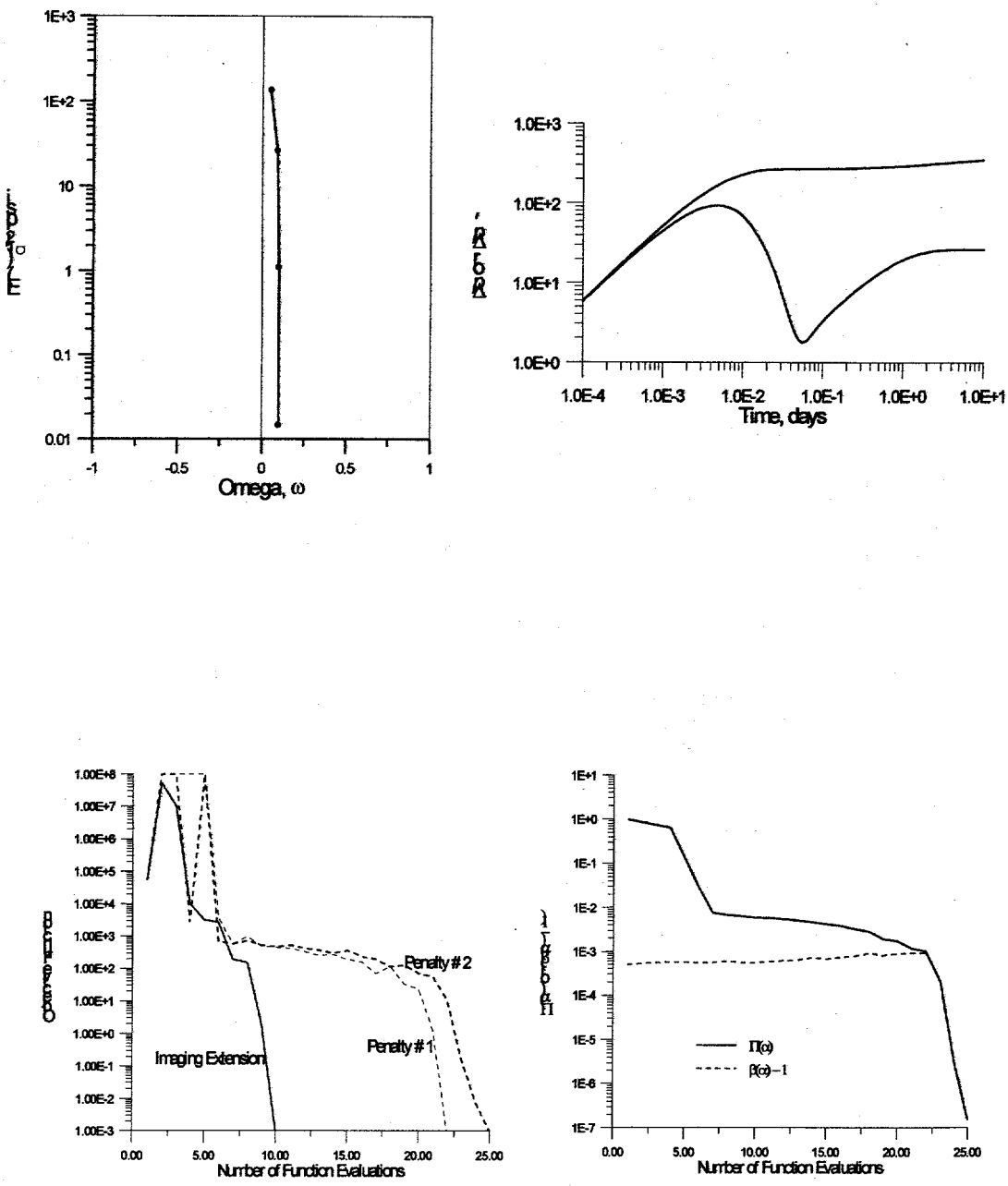


Figure 6.4: (top left) Change in the estimate of omega during the iterative process with initial estimate omega = 0.05, (top right) Synthetic draw down data, (bottom left) Behavior of the objective function during the optimization procedure, and (bottom right) Behavior of the penalty.

6.1.5 Conclusions

In this work, we have presented three alternative procedures for constraining parameter estimates obtained during the course of nonlinear regression to remain within physically meaningful limits. Two methods are based on the classical penalty function approach. The third method is a new procedure for extending the constrained-parameter objective function so that it is defined for unconstrained parameter estimates by an imaging extension procedure. All of the methods are easily applied in public domain unconstrained LS algorithms (such as LMDER) without any modification to the algorithm code.

We showed that whereas it is possible for the penalty function methods to provide non-physical parameter values depending on the initial guesses, (and we discussed how this problem may be circumvented), the new imaging extension procedure will always provide feasible model parameters regardless of initial guess.

Finally, we demonstrated via a synthetic example, that the new imaging extension procedure obtains optimized parameter values faster than the penalty function approaches.

6.1.6 Nomenclature

C	Wellbore Storage Coefficient, bbl/psi
c	Fluid compressibility, psi^{-1}
E	Objective Function that is minimized by nonlinear regression.
E^+	Modified Objective Function that incorporates penalty functions or imaging extension
F_c	Part of the penalty function (see Eq. 1)
h	formation thickness, ft
k	permeability, md
k_f	fracture permeability, md
n	number of measured data points (e.g., pressures).
Δp_{su}	unit rate skin pressure drop, psi/bbl/day
p	number of model parameters
p_i	ith measured pressure, psi
p_{in}	initial pressure, psi
$p_{model,i}$	ith model pressure, psi
q	Producing rate, bbl/day
r_e	reservoir radius, ft
r_i	ith residual.
r_w	wellbore radius, ft
t	time

Greek Symbols

α	the set of model parameters
α_{0i}	initial parameter estimate
α_i	ith model parameter
α_{max}	maximum parameter value
α_{min}	minimum parameter value
β	coefficient of the residuals that incorporates penalty functions; see Eq.

7

$\delta_{min,i}$, $\delta_{max,i}$	scaling constants for ith penalty function; see Eqs. 3 and 4
ε	small number
ζ	small positive number
λ	Dual Porosity parameter
Π	penalty function; see Eq. 14
ρ_i	*i*th LS residual, modified to incorporate penalty functions
ϕ	porosity
ω	storativity ratio

Initial Pressure, p_{in} psi	5000.0
Well Radius, r_w ft	0.364
Reservoir Radius, r_e ft	∞
Formation Thickness, h ft	20.0
Wellbore Storage Coefficient, C bbl/psi	0.025
Unit Rate Skin pressure Drop, Δp_{su} psi/bbl/day	− 0.0706
Fluid Viscosity, μ cp	1.
Fluid Compressibility, c psi^{-1}	9.4×10^{-6}
Fracture Permeability, k_f md	200.0
Fracture Porosity, ϕ_f	0.20
λ	1×10^{-6}
ω	0.01
Producing Rate, q bbl/day	1500.

Table 6.1: Naturally Fractured Reservoir Data (Pseudosteady State Interporosity Flow) for Synthetic Drawdown Example

Parameter	Initial Estimate	Lower Limit	Upper Limit
k_f	150.	0.	1000.
λ	$1.\times10^{-5}$	0.	100.
ω	0.001	0.	1.
C	0.05	0.	1.
Δp_{su}	− 0.1	− 0.5	5.0

Table 6.2: Parameter Initial Estimates

6.2 Expert System Overview

The well test interpretation system can be divided into two main modules. The model identification model deals with analyzing the input data to characterize the well. The other module is the parameter estimation model that used the model information to calculate the parameters required. Figure 6.5 shows the components in the system.

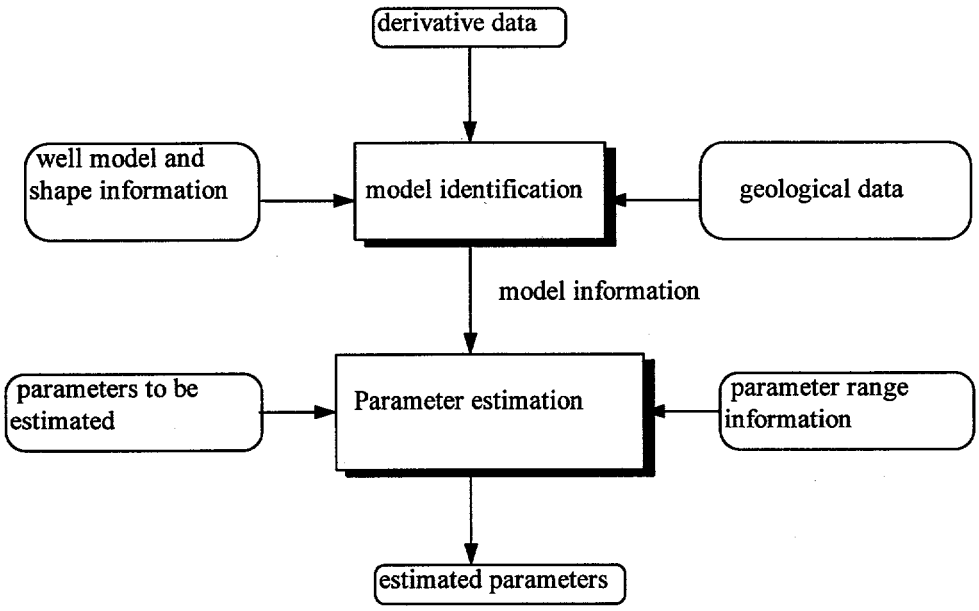


Figure 6.5 : System modules

The input to the system is the time and pressure derivative. This information is provided to the system in an input file. The model identification module analyses this input and creates a description of the shape of the plot in terms of primitives. This shape is then compared with the available model and shape information to select a model. In case of multiple choices, the selection is narrowed down using geological information provided by the user. Once the module is able to classify the input, the model information is then used to estimate the parameters. The parameters that are to be estimated is specified by the user. The user may also specify the range of values that each parameter can have. This helps in making the estimation process more efficient. The output from the parameter estimation module is the final output required from the system. The following section describes each module in greater detail.

The following are the main modules used in the system.

6.2.1 Model identification

The model identification process is made up of two distinct steps. These are: creation of the internal representation of the plot using symbols, and matching this representation with the shape information stored in the system. Each of these steps are discussed below.

The input data is analyzed and converted into an internal representation for further analysis. This provides a simple and efficient method of matching shapes. This representation is done in terms of the following symbols:

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

A flat segment is a segment with slope smaller than 0.1 in absolute value. An *up* segment is one which has slope greater than 0.1. *Plateau, valley, maximum and minimum* are determined from the primitives *up, down* and *flat*. This convention is similar to that of Startzman and Kuo[4] who first observed the usefulness of the symbolic representation of log data.

The algorithm begins by calculating the slopes between the data points. This data is stored in a list. Then the algorithm proceeds by scanning this list and replacing each slope with a symbolic representation. Heuristic rules are applied to create the correct representation. The algorithm does a second scan through the list to come up with a final representation of the whole plot. Here the algorithm uses rules which have been developed to eliminate redundant symbols, or identify new ones based on the primitive symbols (*up, down, flat*). Small disturbances due to noise which slightly distort the plot is also eliminated here. 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.

An interpretation model is usually obtained by combining several components which produce the observed shapes on different parts of the derivative. The expert system is provided with the models and a representation of their derivative.

The initial implementation includes the description of two models which are:

1. Finite wellbore radius well; Infinite isotropic reservoir; Single porosity system.
2. Fully penetrating vertically fractured well; Anisotropic Single porosity system.
3. Finite wellbore radius well; Infinite isotropic reservoir; Dual porosity system.

4. Fully penetrating vertically fractured well; Anisotropic Dual porosity system.
5. Fully penetrating line source well; Anisotropic Dual porosity system

The characteristic derivative shapes associated with these models were determined and are as follows:

1. model1: [*up, maximum, down, flat*]
2. model2: [*up, flat*]
3. model3: [*up*]

Model3 is an alternate representation of model2. All the above models are a representation of the plots that show the time vs. derivative curve.

The matching algorithm is designed along the lines of the work done by Allain and Horne[3]. The aim is to find the model that qualitatively matches the data. The algorithm starts with selecting all the models that match with the input data at the first symbol. It then proceeds to the next symbol and eliminates all the models which do not match with the input symbols at any stage. Finally at the end of the process, the algorithm ends up with the correct model, or no model in case there is no match. In certain cases, the system cannot narrow down the search to a single model. This may be due to the nature of the input where it matches more than one model. The geological information may be used to proceed further.

Our matching algorithm has been modified to work correctly with the symbolic representation of models in our system. A particular input data matches a model if its sequence of symbols completely match with those of a particular model. Since the model information is complete and has not been broken down to different regimes, a complete match of all the symbols of the input data with those of a model is considered. This approach is simpler than considering the different regimes (as done by Allain and Horne[3]), and works well when complete model information has been coded in the system.

The fact that one can hardly select an appropriate interpretation model from the pressure transient data alone emphasizes the need for incorporation of external data into well test analysis. This information can help in narrowing down the choice of models. The system presently uses 5 parameters to make its decisions. The parameters and their possible values are shown in Table 6.3.

Parameter	Possible values
Geometry	Vertically Radial Well (VRW) Horizontal Well (HW) Vertically Fractured Well (VFW) Radially Heterogeneous Well (RHW) Layered System (LS)
Penetration	Fully Penetrating Partially Penetrating
Porosity	Single porosity Dual porosity
Conductivity	Finite Conductivity Infinite Conductivity
Group	This parameter can be any of the possible values in the above groups or any of their possible combinations. Eg: Vertically Radial Well-Fully Penetrating

Table 6.3

The parameter *Group* is used by the system to select the set of interpretation models. All the other parameters are taken as input from the user. Information about *Geometry* and *Penetration* is usually available. *Porosity* and *Conductivity* on the other hand may not always be known.

6.2.2 Parameter estimation

Parameter estimation is carried out after a model has been selected by the system. Though they are not intended to be the final result of the interpretation, these estimates constitute the starting point for an automated type curve matching analysis. The system was thus extended to include a very powerful parameter estimation algorithm. Figure 6.6 shows the parameter estimation module in relation to the other modules in the system:

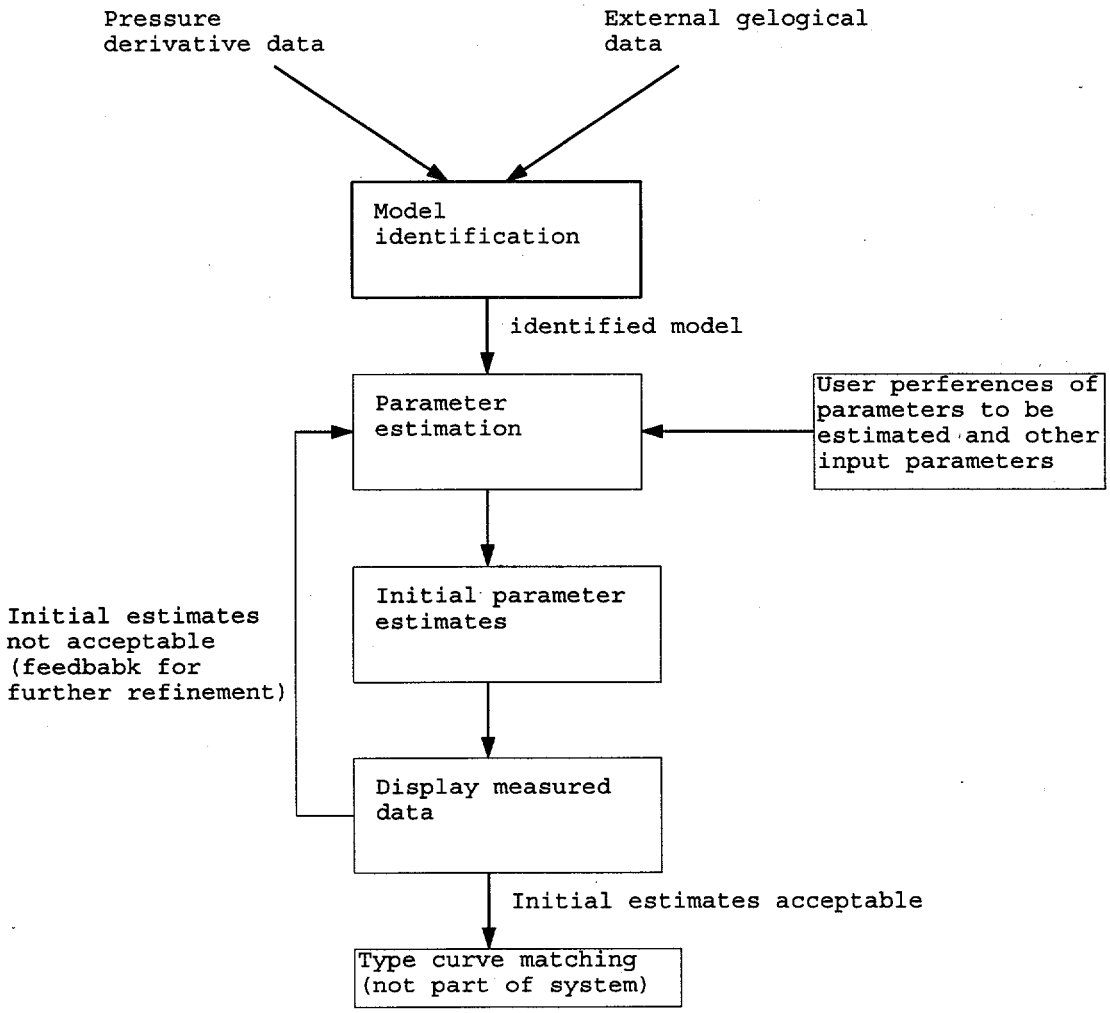


Figure. 6.6: Parameter estimation module

Presently, the system carries out the estimation process only after a single model is identified. In case multiple models are selected, the user needs to narrow down the choice to only one model. The user then selects the parameters that are to be estimated. The user is provided with an interface for this purpose. Based on the model identified and the parameters specified, the system builds up an input to the estimation program. The estimation module is then executed. The output from this module is obtained in an output file which gives the values of the required parameters.

The parameter estimation module also produces the calculated pressure and derivative values based on the model and the parameter values. This data is displayed to the user and helps in judging the correctness and the accuracy of the estimated values. The user may decide to accept these values, or run the estimation procedure again. Refining the estimated values may require changing certain parameters used by the estimation program.

6.2.3 Using the system

The system provides the user with an interface that makes it easy to provide input and analyze the results. Using the well test interpretation system is described using the following example. The interpretation process is started by the user by clicking on the 'Interpret' button. The user is then prompted for the input file name. This file is loaded by the system and the plot is displayed to the user. The system then matches the plot to the stored model information and prints out the model identified.

A single model needs to be selected before the parameter estimation algorithm can be executed. The user can provide the model to the system using the interface's *Enter model* button. External information can also be to possibly narrow down the search. This information is entered into the system using the *External Data* button in the interface. In this case, the knowledge that this is a fractured well was is entered by choosing the corresponding entry in the *Geometry* field in external data entry screen.

The parameter estimation module is executed using the *Estimate Parameters* button in the interface. The options that need to be specified, like the parameters to be estimated, and their range, is entered by clicking on the *Set Parameters* button. This invokes a screen to enter these values.

6.2.4 Further extensions

The program as shown in the previous section was coded in KAPPA. For greater efficiency and scalability, this is being converted to C++ code. Though the internal representation and some of the interfaces have been enhanced for better performance, the basic functionality provided by the system remains the same. The following extensions are required in the system:

Extending the model information in the system: The system needs to be extended to handle all possible well models. This process does not require any major change to the system. Only the shape information for new models need to be provided. Entering the shape information into the system is all that is required to extend the system. This can be done by the following steps:

Eliminating noise: The system needs to be extended to handle noisy data. This can be achieved by simply adding a smoothing routine to smooth the plot before analyzing it.

6.2.5 Conclusion

The well test interpretation system is an efficient tool to analyze well data. The system starts with the minimum information available (derivative data), and tries to classify the well. Other information is used if required. Once the well is classified, the system can then be used to estimate the parameters for that well. This system has been developed as a stand-alone tool to be used by an expert. It frees the user of the routine and tedious task of classifying the wells and estimation of parameters. The information obtained from the system is also useful, and can be used by other groups in the project.

8. References

1. Oliver, D.S.: "Incorporation of Transient Pressure Data into Reservoir Characterization", In Situ 18(3), pp 243-275 (1994)
2. Perez, Godofredo: "Stochastic Conditional Simulation for Description of Reservoir Properties", Ph.D. Dissertation, The University of Tulsa, Tulsa, OK (1991)
3. Kirkpatrick, S., Gelatt Jr., C.D. and Vecchi, M.P.: "Optimization by Simulated Annealing", Science (May 13 1983) pp 671-680.
4. Moridis, G.J., McVay, D.A., Reddell, D.L. and Blasingame, T.A.: "The Laplace Transform Finite Difference (LTFD) Numerical Method for Simulation of Compressible Fluid Flow in Reservoirs", SPE 22888 presented at the 1991 Annual Technical Conference and Exhibition, Dallas, TX, Oct. 5-8
5. Carvalho, Renato de Souza.: "Nonlinear Regression: Application to Well Test Analysis", Ph.D. Dissertation, The University of Tulsa, Tulsa, OK (1993)
6. Yang, A.-P.: "Turning Bands Method to Generate 2-D Random Field with Autocorrelation," University of Texas at Austin (April 1987)
7. Deutsch, C.V. and Journel, A.G.: *GSLIB Geostatistical Software Library and User's Guide*, Oxford University Press Inc., NY (1992)
8. *ECLIPSE 100 - Black Oil Simulator*, ECL-Bergeson Petroleum Technologies, Inc., Oxfordshire, England (1990)
9. Thompson, Leslie: Personal Communication (Apr. 1995)
10. Chen, Chih-Cheng and Raghavan, Rajagopal: "An Approach To Handle Discontinuities by the Stehfest Algorithm", SPE 28419 presented at the SPE 69th Annual Technical Conference and Exhibition held in New Orleans, LA, U.S.A., 25-28 September 1994
11. Greenbaum, A. et al.: *Sparse Linear Algebra Package Version 2.0* Lawrence Liverpool National Laboratory, Liverpool Computing Center (1986)
12. Kincaid, David R., Respess, John R., Young, David M. and Grimes, Roger G.: *ITPACK 2C: A FORTRAN Package for Solving Large Sparse Linear Systems by Adaptive Accelerated Iterative Methods* freeware and documentation from netlib.
13. Barrett, R., Berry, M., Chan, T., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C. and van der Vorst, H.: *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods* freeware and documentation from netlib.
14. Hensley, Jeff: Personal Communication (Jan. 1995)
15. Redner, Richard: Personal Communication (Aug. 1994)
16. Aarts, Emile and Korst, Jan: *Simulated Annealing and Boltzmann Machines A Stochastic Approach to Combinatorial Optimization and Neural Computing*, John Wiley and Sons, New York (1989)
17. Lee, J: *Well Testing,* SPE, Dallas (1982).
18. Oliver, D.S.: "The Averaging Process in Permeability Estimation From Well-Test Data", SPE 19845 in *SPE Formation Evaluation*, September 1990.

19. Alabert, F.G: "Constraining Description of Randomly Heterogeneously Reservoirs to Pressure Test Data: A Monte Carlo Study", paper SPE 19600 presented at the 64th Annual Technical Conference and Exhibition of the Society of Petroleum Engineers held in San Antonio, TX., October 8-11, 1989.
20. Starzman R. A. and Kuo T. B., 1987, An artificial intelligence approach to well log correlation: Log Analyst, March-April, p.175-183.
21. Yarus, J.M. and Chambers, R.L, eds., *Principles, Methods, and Case Studies*, AAPG, Tulsa, 1994.
22. Abbaszadeh, M. and Kamal, M.M.: "Automatic Type-Curve Matching for Well Test Analysis," *SPEFE* (Sept. 1988) 567–577.
23. Rosa, A.J. and Horne, R.N.: "Automated Type-Curve Matching in Well Test Analysis Using Laplace Space Determination of Parameter Gradients", paper SPE 12131 presented at the 58th SPE Annual Technical Conference and Exhibition, San Francisco, CA (October 5–8, 1983).
24. Barua, J., Horne, R.N., Greenstadt, J.L. and Lopez, L.: "Improved Estimation Algorithms for Automated Type Curve Analysis of Well Tests," *SPEFE* (March, 1988) 186–196.
25. Nanba, T. And Horne, R.N.: "An Improved Regression Algorithm for Automated Well Test Analysis," paper SPE 18161 presented at the 63rd SPE Annual Technical Conference and Exhibition, Houston, TX (October 2-5, 1988).
26. Rosa, A.J. and Horne, R.N.: "Automated Well Test Analysis Using Robust (LAV) Nonlinear Parameter Estimation," paper SPE 22679 presented at the 66th SPE Annual Technical Conference and Exhibition, Dallas, TX (October 6-9, 1991).
27. Horne, R.N.: "Advances in Computer-Aided Well Test Interpretation," paper SPE 24731 presented at the 1992 SPE Annual Technical Conference and Exhibition, Washington D.C., Oct. 4-7.
28. Carvalho, R.S., Redner, R., Thompson, L.G. and Reynolds, A.C.: "Robust Procedures for Parameter Estimation by Automated Type Curve Matching," paper SPE 24732 presented at the 67th SPE Annual Technical Conference and Exhibition, Washington, D.C. (Oct. 4-7, 1992).
29. Garbow, B.S., Hillstrom, K.E. and More, J.J.: Subroutine LMDER, Argonne National Laboratory, MINPACK project, March, 1980.
30. Carvalho, R.S.: *Nonlinear Regression: Application to Well Test Analysis*, Ph.D. Dissertation, The University of Tulsa, Petroleum Engineering Department, 1993.

☆U.S. GOVERNMENT PRINTING OFFICE: 1996 - 760-632/20071