History Matching In Parallel Computational Environments

Annual Report

Reporting period start date: 1 Oct. 2004
Reporting period end date: 30 Sept. 2005

Authors: Dr. Steven Bryant, Dr. Sanjay Srinivasan, Alvaro Barrera and Sharad Yadav

Date Issued: October 2005

DOE Award Number: DE-FC26-03NT15410

Submitting organization: Center for Petroleum and Geosystems Engineering
The University of Texas at Austin
Austin, Texas 78712-0228

DISCLAIMER

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

ABSTRACT

A novel methodology for delineating multiple reservoir domains for the purpose of history matching in a distributed computing environment has been proposed. A fully probabilistic approach to perturb permeability within the delineated zones is implemented. The combination of robust schemes for identifying reservoir zones and distributed computing significantly increase the accuracy and efficiency of the probabilistic approach.

The information pertaining to the permeability variations in the reservoir that is contained in dynamic data is calibrated in terms of a deformation parameter r_D. This information is merged with the prior geologic information in order to generate permeability models consistent with the observed dynamic data as well as the prior geology. The relationship between dynamic response data and reservoir attributes may vary in different regions of the reservoir due to spatial variations in reservoir attributes, well configuration, flow constrains etc. The probabilistic approach then has to account for multiple r_D values in different regions of the reservoir.

In order to delineate reservoir domains that can be characterized with different r_D parameters, principal component analysis (PCA) of the Hessian matrix has been done. The Hessian matrix summarizes the sensitivity of the objective function at a given step of the history matching to model parameters. It also measures the interaction of the parameters in affecting the objective function. The basic premise of PC analysis is to isolate the most sensitive and least correlated regions. The eigenvectors obtained during the PCA are suitably scaled and appropriate grid block volume cut-offs are defined such that the resultant domains are neither too large (which increases interactions between domains) nor too small (implying ineffective history matching).

The delineation of domains requires calculation of Hessian, which could be computationally costly and as well as restricts the current approach to some specific

simulators. Therefore a robust technique to evaluate a covariance matrix, which is analogous to 'Hessian matrix', from a set of equi-probable realizations has also been developed. This technique is easy to implement and provides the domains, which could be intuitively justified.

Since the domain delineation process yields zones that are least correlated with each other, each r_D parameter can be optimized independently and simultaneously using individual nodes of a cluster of computers. Further least correlation criteria help in retaining the simplicity of 1-D optimization during the history matching. Upon convergence, the perturbed regions are put together and the history match is verified. The proposed approach results in a set of independent tasks of equal magnitude and thus is particularly suited for distributed computing. The methodology has been successfully tested on various synthetic cases.

TABLE OF CONTENTS

DISCLAIMER ii

ABSTRACT iii

TABLE OF CONTENTS v

LIST OF TABLES vii

LIST OF FIGURES viii

EXECUTIVE SUMMARY xv

1. INTRODUCTION 1

2. VALIDATING PROBABILISTIC APPROACH FOR DYNAMIC DATA INTEGRATION 5

3. APPLICATION OF PARALLEL AND DISTRIBUTED COMPUTING 30

Concepts in parallel and distributed computing applied to reservoir simulation 31
1) Parallel computing 31
Amdahl's law 33
2) Distributed computing 36
Choice of the reservoir simulator and its important features 38
Eclipse (General) 38
Fully implicit technology (Black Oil) 38
Adaptive Implicit and IMPES (Compositional) 39
Parallel Option 39
Flux Boundary Conditions 40
Selection of a parallel and distributed computing scheme considering the specific requirements of the project 41
Preliminary verification of the availability of the required features in the selected simulator 43
Test Cases 43
Parallel Simulation 43
Case1 and Case 2 43
Flux Boundary Conditions 47

4. SENSITIVITY ANALYSIS 49

Example 1 50
Example 2 57

5. DESCRIPTION OF METHOD 61

6. TEST CASES 72

1. Alternate ways of Domain Decomposition 73
   a) Domain decomposition based on porosity and thickness 73
   b) Domain decomposition based on permeability covariance matrix 77
2. Computational cost associated with sensitivity analysis 82
3. Robustness of Domain Delineation 83
   Case 1 83
   Case 2 86
4. History matched synthetic cases 88
   Case 1 89
   Case 2 93
   Comparison of Case 1 and 2 95

7. SENSITIVITY COEFFICIENTS AND CORRELATION COEFFICIENTS FROM A SET OF EQUI-PROBABLE REALIZATIONS 97

Case 1 98
Case 2 101
Case 3 103
Case 4 107
Inferences 109

EXPERIMENTAL METHODS/APPROACH 112

RESULTS AND DISSCUSSIONS 112

CONCLUSIONS 117

COST AND SCHEDULE SECTION 118

SUMMARY OF SIGNIFICANT ACCOMPLISHMENTS DURING REPORTING PERIOD 119

FUTURE WORK 119

REFERENCES 120

LIST OF TABLES

Table 2.1: Description of the synthetic simulation case used to evaluate the probabilistic dynamic data integration algorithm. 24

Table 3.1: Bandwidth and Latency for various switches of a PC cluster 33

Table 3.2: Influence of aspect ratio on computation speed using ECLIPSE 100 (1-D domain decomposition). 44

Table 3.3: Computation time against number of grid blocks. 45

Table 3.4: Computation time versus ratio x to y grid block sizes 46

Table 6.1: Profile of the eigenvalues for the two cases discussed 81

Table 6.4: Reservoir model parameters for case 1 of History matched examples 89

Table 6.5: Showing the objective function for 1-well case 91

Table 6.6: Reservoir model parameters for case 2 of History matched examples 93

Table 6.7: Showing the objective function for 6-well case 95

LIST OF FIGURES

Fig 2.1: Spatial interpolation obtained by: a) Kriging; b) Sequential Gaussian simulation and c) Sequential indicator simulation..... 7

Fig 2.2: Example of gradual deformation of geological models by probability perturbation method – First scheme. A single parameter, r_d determines the magnitude of the perturbation in this model. 12

Fig 2.3: Effect of deformation parameter on local distribution for third perturbation scheme. For rd = 0.5, the local distribution conditioned to geological information is recovered. 14

Fig 2.4: Example of gradual deformation of geological models by probability perturbation scheme four. A single parameter, r_d determines the transition between an initial realization and two proposals. The range of variation of the geological model is enlarged under this scheme..... 16

Fig 2.5: Effect of varying the r_D parameter on: a) Oil production rate at Producer 1; b) Water cut at Producer 1; c) Oil production rate at Producer 2; d) Water cut at Producer 2. 17

Fig 2.6: Sensitivity of water saturation distribution in the reservoir at 4000 days and 5200 days to r_D parameter..... 18

Fig 2.7: Example of convergence of an objective function for a single-parameter problem using the Dekker-Brent inverse parabolic interpolation algorithm. 19

Fig 2.8: Field Pressure History Match obtained with the probabilistic dynamic data integration algorithm for the synthetic case study. Results after forty flow simulation runs, distributed on 5 outer iterations of 8 inner Dekker-Brent iterations each. 26

Fig 2.9: Field Production History Match of synthetic case using the probability perturbation method. Final model is obtained after 40 simulation runs. 26

Fig 2.10: Example of Gradual Deformation of geological models, obtained with the probability perturbation method. Variations in the third layer of the geological model through 40 flow simulation runs are presented. 27

Fig 2.11: Third layer of the reference geological model used in the synthetic case to generate the historical production data. 27

Fig 2.12: Example of convergence of the objective function applying the probabilistic approach to history matching. 28

Fig 3.1: Influence of aspect ratio on computation speed using ECLIPSE 100 (1-D domain decomposition) 44

Fig 3.2: Plot of computation time against number of grid blocks. 45

Fig 3.3 : Full field flow simulation results: a) Pressure variations in the reservoir at 500 days obtained by full field simulation; b) Pressure variations in a domain obtained corresponding to boundary conditions derived from the full field simulation; c) Oil saturation values obtained by full field simulation, and d) Oil saturation values obtained for sub-domain simulation using boundary conditions derived from full field simulation. 47

Fig 4.1: Reservoir model utilized to generate the base case production history 51

Fig 4.2: (a) Permeability realization for Layer 5 of STAN 5 51

(b) Permeability realization for Layer 3 of STAN5 51

Fig 4.3: Global sensitivities for: a) 3rd layer, and b) 5th layer. The permeability corresponding to the 4th layer is used as the base model for calculating the sensitivities. 52

Fig 4.3(c): Permeability for base case (d) Permeability for 3rd layer (e) Global sensitivity coefficient on percentile map for layer 3 with respect to layer 4 of STAN5. 53

Fig 4.4: Sensitivity coefficients for the layer 3 permeability model calculated on the basis of: a) Well 1 oil production rate; b) Well 2 oil production rate; c) Well 3 oil production rate; d) Well 1 water production rate; e) Well 2 water production rate; and f) Well 3 water production rate. 54

Fig 4.5: Sensitivity coefficients for the layer 3 permeability model calculated on the basis of: a) Well 1 gas oil ratio, b) Well 2 gas oil ratio, c) Well 3 gas oil ratio. 55

Fig 4.6: Permeability field with perm x very different from the base case. The corresponding global sensitivity coefficients computed on the basis of the base case model are shown in the right. 58

Fig 4.7: Global sensitivities corresponding to a homogeneous permeability field. 58

Fig 4.8: Typical objective function profile and the importance of a good initial guess. 59

Fig 5.2: (a) Actual permeability field used to generate the history; 64

(b) Initial guess to be history matched; (c) show the domains obtained without scaling of eigen vectors with thresholds of 30% for the five eigen vectors; (d) Domains with scaled eigen vectors with thresholds 30% 64

Fig 5.3: Showing the sensitivity coefficients on the percentile map for the case '5.2(a)' as reference for creating history and '5.2 (b)' as the initial guess. Highest values in red, intermediate in green and the lowest in blue, negative sensitivity coefficients regions are shown with shaded lines; (a) With respect to well-1 oil production rate (b)

With respect to well-2 oil production rate (c) With respect to well-3 oil production rate..... 66

Fig 5.4: Showing the sensitivity coefficients on the percentile map for the realization shown in Fig. '5.2(a)' as reference for creating history and Fig. '5.2(b)' as the initial guess. Highest values in red, intermediate in green and the lowest in blue, negative sensitivity coefficients regions are shown with shaded lines; (a) With respect to well-1 oil production rate b) With respect to well-1 water production rate (c) With respect to well-3 Gas oil ratio..... 66

Fig 5.5: Flow chart showing the work flow; step 6 has been described in detail Fig 5.6 70

Fig 5.6: Flow chart showing the work-flow for r_d optimization 71

Fig 6.1: a) Permeability for base case model (Layer 4 of Stan5) used to generate the production history; b) Geobodies computed using the porosity and layer thickness information for layer 4; c) Geobodies computed using a modified definition that focuses on a search radius of 16 units around well locations. 74

Fig 6.2: a) Sensitivity coefficients for constant 1000 md (in X, Y & Z directions) permeability realization b) Sensitivity coefficients for 3rd Layer of Stan5 c) Permeability realization used for the sensitivity analysis for the sensitivity coefficients shown in fig. 6.2d d) Corresponding sensitivity coefficients for 6.2c 75

Fig 6.3: Eigen domains based on covariance matrix of permeabilities. The following is the color scheme employed: First eigenvector positive components ■, second eigenvector positive components ■ and third eigenvector positive components ■. 77

Fig 6.4: Reservoir regions for the STAN5 data obtained using the permeability covariance matrix: a) Applying a 60% cut-off for the first two eigenvectors of, b) Applying a 50% cut off on the first 3

eigenvectors, c) Applying a 50% cut-off on the first 5 eigenvectors, d) Applying a 30% cut off on the first 5 eigenvectors..... 79

Fig 6.5: Reservoir regions delineated corresponding to a suite of permeability models obtained using p-field simulation. The eigen decomposition of the permeability covariance was performed and the reservoir regions were obtained by: a) Applying a 60% cut-off for the first two eigenvectors, b) A 50% cut off applied to the first 5 eigenvectors, and c) A 30% cut off applied to the first 5 eigenvectors. 80

Fig 6.6: Sensitivity coefficients on percentile plot for the 3rd layer of STAN5 (with 4th layer as base case for generating history) at a) 100 days, b) 500 days, and c) 1500 days. 82

Fig 6.7 (a) Actual permeability field used to generate the history; 84

(b) Initial guess to be history matched (c) Sensitivity coefficients on percentile map. Highest values in red, intermediate in green and the lowest in blue, negative sensitivity coefficients regions are shown with shaded lines; (d), (e), (f), (g) show the domains obtained after scaling of eigen vectors with thresholds of 20%, 40%, 60% and 80% respectively; (h), (i), (j) and (k) show the domains with unscaled eigen vectors with thresholds of 20%, 40%, 60% and 80% respectively for all the five eigen vectors. 84

Fig 6.8: (a) Showing the actual permeability field used to generate the history; (b) Initial guess to be history matched; (c) show the domains obtained without scaling of eigenvectors with thresholds of 30% for the five eigenvectors; (d) show the domains with scaled eigenvectors with thresholds 30%. 87

Fig 6.9: (a) Reference realization for creating history (b) Domains for 80% threshold (c) Initial Guess (d) Realizations after perturbing

domain wise (e) Merged realization (f) Realization after perturbing the merged realization by one deformation parameter 90

Fig 6.10: The red line is the initial guess, green is the target response and blue is the response from history matched realization 92

Fig 6.11: (a) Domains for 50% threshold (b) Realizations after perturbing domain wise (c) Merged realization (d) Realization after perturbing the merged realization by one deformation parameter 94

Fig 7.1: (a) to (d) are the four of total 50 realizations used for the analysis 98

Fig 7.2: Averaged sensitivities calculated for the set of realizations with respect to a) Oil Production rate b) Well production rate c) Gas oil ratio d) Total oil production e) Total water production 99

Fig 7.3: Correlation coefficients calculated over the set of realizations with respect to: a) Oil Production rate at 500 days b) Well production rate at 500 days c) Gas oil ratio at 500 days d) Oil Production rate at 2000 days e) Well production rate at 2000 days f) Gas oil ratio at 2000 days 100

Fig 7.4: Permeability realization corresponding to different r_d values $r_d = 0$ b) $r_d = 0.5$ c) $r_d = 1$ 101

Fig 7.5: Correlation coefficients calculated over the set of realizations at 2000 days with respect to a) Oil Production rate b) Well production rate c) Gas oil ratio 102

Fig 7.6: Correlation coefficients calculated over the set of realizations (with correlation lengths 20, 20) at 2000 days with respect to a) Oil Production rate b) Well production rate c) Gas oil ratio 103

Fig 7.7: Correlation coefficients calculated over the set of realizations (with correlation lengths 10, 10) at 2000 days with respect to a) Oil Production rate b) Well production rate c) Gas oil ratio 104

Fig 7.8: Correlation coefficients calculated over the set of realizations (with correlation lengths 5, 5) at 2000 days with respect to a) Oil Production rate b) Well production rate c) Gas oil ratio 105

Fig 7.9: Correlation coefficients calculated over the set of realizations (with correlation lengths 5, 10 in x and y directions respectively) at 2000 days with respect to a) Oil Production rate b) Well production rate c) Gas oil ratio 106

Fig 7.10 (a): Variance of the pressure or the sensitivities from a set of realizations; The figure has been scaled down to 50*50 so that the final covariance matrix used for PCA is manageable. (b) Most sensitive and least correlated regions obtained after subjecting the covariance matrix to PCA. 108

Fig 7.11: Variance of the pressure or the sensitivities 109

EXECUTIVE SUMMARY

The objective of history matching is to modify a prior model for the reservoir such that the updated model reflects the available production data and the uncertainties in production forecasts are reduced. This project approaches the problem of developing a robust scheme for history matching with two guiding ideas. First, the reservoir model that results from the scheme should be geologically plausible. The motivation for this condition is that the inverse problem represented by history matching – choosing a large set of parameters (local values of permeability) so that a small set of data (flow rates at wells as a function of time) is matched – is under-constrained. A solution to this inverse problem that makes geological sense is more likely to provide reliable forecasts. The second guiding idea is that it must be possible to obtain insight from a computer implementation of the history matching in a practical length of time (e.g. overnight). The time scale for decision-making in many industrial applications does not allow for lengthy calculations.

We have implementing these ideas in this project by taking advantage of two technologies: a probabilistic approach for integrating production data into the reservoir model which maintains geological consistency and distributed computing for making the algorithm computationally efficient.

Our probabilistic approach to dynamic data integration is based on conditional probability distributions that account for the uncertainty in permeability at any given location in the reservoir. At the beginning of the history matching process, these distributions embody the statistical properties of the reservoir heterogeneity as inferred from well logs, core samples, etc. The key idea being tested in this project is that the conditional probability distributions are iteratively updated during the history matching procedure to account for the additional information contained in the dynamic response data (the production histories). We have implemented this update with a perturbation parameter r_D, that controls the magnitude of the deformation applied to the values of permeability in the model. Scheme for this perturbation has been developed, yielding an approach that ensures a gradual deformation of the model. This has proved to be a good way to maintain consistency between the model and the geological reality. To date we have validated this scheme on some test cases.

We have extended the notion of a single perturbation parameter r_D to consider a set of such parameters for a given reservoir. Each parameter applies to a particular domain in the reservoir; the domains are non-overlapping. This extension provides a basis for decomposing the flow simulation onto distributed computing platforms as well as for increasing the effectiveness of the perturbation scheme. For the latter, we have used a sensitivity analysis. The influence of the value of permeability at any given location upon the production data predicted from the forward model can be extracted from the simulator. We have applied a principal components analysis on these sensitivities to identify domains on the basis of sensitivity and least correlation. Sensitive regions

manner increases the effectiveness of the history matching process. The least correlation criterion makes the problem amenable to distributed computing and hence imparting efficiency in terms of computational time. Least correlation also helps in retaining the simplicity of 1-D optimization. The full algorithm has been successfully tested on 2D reservoirs.

The delineation of domains requires calculation of Sensitivities, which could be computationally costly and as well as restricts the current approach to some specific simulators. Therefore we have developed a robust technique to evaluate sensitivities from a set of equi-probable initial realizations. This technique is easy to implement and provides the domains, which could be intuitively justified.

We have successfully implemented an integration of all these ideas into a single history-matching algorithm. The next phase of the research would concentrate on testing the full algorithm on complex 3D reservoirs within a practical length of computational time.

1. INTRODUCTION

The objective of history matching is to modify a prior model for the reservoir such that the updated model reflects the available production data and the uncertainties in production forecasts are reduced. The resultant geological model must therefore not only reproduce production data by numerical simulation but it must also be consistent with the geological knowledge of the reservoir.

The history matching process mainly consists of:

1) Identifying model parameters that could be modified to effect history match.
2) Defining a suitable objective function for the optimization procedure.
3) Proper selection and design of an optimization technique for reducing the objective function to a minimum.
4) Tracking computational cost associated with the flow simulations used within the selected optimization technique.

Many sets of parameter estimates may yield nearly identical matches of the data. A decision therefore has to be taken during the history matching procedure to determine the set of parameters estimates that are most appropriate given prior knowledge about the geology.

The data obtained from the field can be classified as static data or dynamic data. The static data do not vary with time e.g. permeability, porosity etc. while the dynamic data do vary with time e.g. production rates, well bore flowing pressures etc. The reservoir model is developed based on the available static data using geostatistical simulation or a geological model. History matching techniques attempt to constrain the reservoir models to the dynamic data. The goal of the history matching is to minimize the objective function that measures difference between simulator response and the observed field data. The objective function may be defined as follows

$$\mathbf{Q} = (½)\sum_i \sum_j \sum_k \frac{(z_{ijk}^{observed} - z_{ijk}^{simulated})^2}{\sigma_{ijk}^2} \quad (1.1)$$

Z^{obs} is the measured value of a flow response such as bottom-hole pressure, gas oil ratio, oil production rates etc. $Z^{simulated}$ is the corresponding value obtained by performing a flow simulation on the reservoir model synthesized using the available data. σ_{ijk} is the measurement error. The index **k** means the type of observation data, **j** is the index for the number of wells, **i** is the index of the measurements dates for each well. The quantity $1/\sigma^2{}_{ijk}$ can be viewed as the weight assigned to the response Z_{ijk} – the larger the measurement error, the less the contribution of the mismatch to the overall objective function. In the cases studied in this report, the measurement error is assumed to be the same for all the observations used in the calculation of the objective function. Despite this simplification, the behavior of the objective function is strongly non-linear and small perturbation to model parameters can result in large fluctuations of the objective function.

Some authors have advocated that a prior information term (or regularization) should be added to the objective function formulation so that geological consistency is maintained during history matching process. Recently it has been proposed to add a 4D seismic data mismatch term in the objective function (Gosselin et al. 2000).

The combined objective function can then be written as:

$$Q = a^* (½)\sum_i \sum_j \sum_k \frac{(z_{ijk}^{observed} - z_{ijk}^{simulated})^2}{\sigma_{ijk}^2} +$$

$$b^* ½ (y_k - y_k^{mean})^t C_p^{-1} (y_k - y_k^{mean}) \quad + \quad ½ (s(m) - e)^t W_s (s(m) - e) \quad (1.2)$$

Where

a, b are the user's defined weight constants, W_s is the weight matrix assigned to seismic data. y_k is the history matching parameter (e.g. Permeability at a given grid cell), y_k^{mean}

is the mean of the parameter. C_p^{-1} is the inverse of the covariance matrix of permeability and s(m) is the seismic derived predicted values and 'e' are the observed seismic values. The first two terms in the objective function measure the production mismatch and the deviation from the prior geological model while the last term quantifies the deviation from the seismic information. During this course of study only the production mismatch term was used in the objective function.

Multiple approaches have been considered for conditioning reservoir models to dynamic flow data. Trial and error optimization algorithms follow an iterative process to update the reservoir model applying a perturbation scheme for converging to a set of model parameters that yield a match to the observed flow response. Some of the most popular methods include gradient based methods (Yeh, 1986; Anterion, F. et al., 1989), pilot point methods (La Venue el al., 1992; de Marsily et al., 1995), sequential self-calibration methods (Gomez-Hernandez et al., 1997), Markov chain Monte Carlo methods (Omre and Tjelmeland, 1996) and gradual deformation methods (Roggero and Hu, 1998). There are other geostatistical methods that presuppose that the permeability fields could be conditioned to both hard data as well as dynamic data in a probabilistic sense by using a Bayesian formulation for the integration.

Optimization approaches upon convergence yield a deterministic reservoir model. In contrast, a probabilistic approach was proposed by Kashib and Srinivasan (CIM paper 2002-125, Kashib, T. and Srinivasan, S) and was applied successfully to history-match several 2-D and 3-D reservoir examples. A conditional distribution representing the local uncertainty in permeability at a location is derived on the basis of available static and geologic information. The uncertainty in permeability value at the same location as indicated by the dynamic data is calibrated using an update parameter r_D. These conditional probability distributions are merged using an approach proposed by Journel (Mathematical Geology, V.34, N.5: 573-596, 2002). An updated value of permeability is

obtained by sampling from the merged probability distribution and the process is repeated at all locations.

A methodology is developed that increases the accuracy of the probabilistic approach by using multiple deformation parameters in different domains of the reservoir. Since the deformation parameter contains the dynamic information, the definition of domains is based on the flow sensitivities. Furthermore, domains are defined in such a way that makes the probabilistic approach amenable to distributed computing resulting in significant savings in computational cost. The evaluation of sensitivity coefficients is computationally expensive. In order to make the algorithm efficient, sensitivities are calculated only at a few locations and interpolated to all remaining locations in the reservoir. The proposed algorithm is demonstrated on a synthetic example.

2. VALIDATING PROBABILISTIC APPROACH FOR DYNAMIC DATA INTEGRATION

Reservoir models are generally constructed considering subsurface geological data obtained from different sources (such as seismic, well logging, well tests, sequence stratigraphy, etc), and a geological model of heterogeneity. The prior geological model for heterogeneity is commonly represented in the form of a variogram model that is inferred from the same conditional subsurface information. These two components are combined within a simulation/interpolation framework to generate geological models conditioned to static information.

Geostatistics as a geological modeling technique and its two original basis: i) the variogram model and ii) the kriging interpolation methodology, were initiated by the work of Daniel Krige (1951) and later developed by Georges Matheron (1962-1963, 1965), with the purpose of providing locally accurate grade estimates of mining blocks; however, its application has extended from the mining industry to many other related disciplines, including the oil industry.

The simple kriging (SK) estimator k_{SK}^{*} at each location $\boldsymbol{u}_j$ of the target geological model $k(\boldsymbol{u})$ (such permeability or porosity field) is the best linear unbiased estimator and can be written as:

$$k_{SK}^{*} - m(\boldsymbol{u}_j) = \sum_{i=1}^{N} \lambda_i(\boldsymbol{u}_j)\left[k(\boldsymbol{u}_i) - m(\boldsymbol{u}_i)\right] \tag{2.1}$$

where, $m(\boldsymbol{u}_j) = E\left\{k(\boldsymbol{u}_j)\right\}, j = 1,...,J$, are the known stationary means of the random function $k(\boldsymbol{u}_j)$ at the locations $\boldsymbol{u}_j$; J is the size of the model; $k(\boldsymbol{u}_i), i = 1,...,N$ are the conditional data; and $\lambda_i(\boldsymbol{u}_j)$ are the kriging weights for each conditional data at each location for the estimation at location $\boldsymbol{u}_j$. The weights are calculated from the following system of equations:

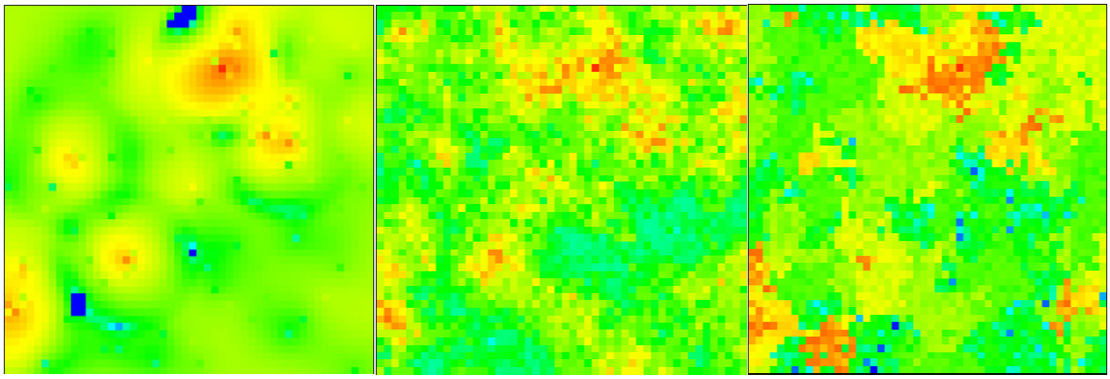
$$\sum_{k=1}^{N} \lambda_k(\boldsymbol{u}_j) C\left(\boldsymbol{h}_{ik}\right) = C\left(\boldsymbol{h}_{ij}\right), \forall i = 1,\ldots,N \tag{2.2}$$

where $C\left(\boldsymbol{h}_{ij}\right)$ or $C\left(\boldsymbol{u}_i - \boldsymbol{u}_j\right)$ is the covariance at the lag $\boldsymbol{h}_{ik} = \boldsymbol{u}_i - \boldsymbol{u}_j$, considering stationarity; and is related to the variogram by: $\gamma\left(\boldsymbol{h}_{ij}\right) = C(0) - C\left(\boldsymbol{h}_{ij}\right)$ and $C(0) = Var\left\{k(\boldsymbol{u}_{i)}\right\}$ .The corresponding minimum estimation (error) variance σ_{SK}^2 is:

$$\sigma_{SK}^2(\boldsymbol{u}_j) = C(0) - \sum_{i=1}^{N} \lambda_i(\boldsymbol{u}_j) C\left(\boldsymbol{h}_{ij}\right) \tag{2.3}$$

Stochastic simulation was introduced by Matheron (1973) and Journel (1974) to correct for the smoothing effects and other artifacts of kriging (See Figure 1a.), and to enable the reproduction of the spatial variance predicted by the variogram model. Different algorithms have been developed including sequential simulation (Journel, 1983, Isaaks, 1990; Srivastava, 1992; Goovaerts, 1997; Chiles and Delfiner, 1999), which has become the workhorse for many current geostatistical applications.

The stochastic simulation approach is based on the calculation of probability distributions at individual locations, considering the conditional information and the spatial heterogeneity model. There are different methods for the construction of the local probability distributions, including the Gaussian approach where the kriging estimation and the estimation variance are used as the mean and the variance of the local normal conditional distribution. In another approach, the use of indicator transforms allows modeling multivariate distributions without relying on Gaussian assumption, to generate models that exhibit more connected and well-defined geological bodies. Figure 1 compares the results of the original kriging interpolation technique with the Gaussian and the Indicator Sequential Simulations, considering the same conditional information.



a b c

Fig 2.1: Spatial interpolation obtained by: a) Kriging; b) Sequential Gaussian simulation and c) Sequential indicator simulation.

Stochastic simulation also provides the capability to generate multiple equiprobable realizations, giving birth to the idea of assessing spatial uncertainty (Journel and Huijbregts, 1978) on reservoir models.

Sequential Indicator Simulation

Spatial distributions can be modeled following a non-parametric approach, where the local probability distributions $F(\boldsymbol{u}; z_i)$ can be calculated from the available conditional information, by defining a set of thresholds $z_i, i = 1, ..., NT$ to discretize the range of variability of the spatial variable, and subsequently performing indicator kriging using the indicator transformed variables.

The indicator transform of a random variable is simply a binary transform: the value one is assigned if the value at a location is less than the threshold and zero if not. The expected value of an indicator random variable is therefore equivalent to the probability of that particular threshold. Hence, the probability distribution can be calculated by sequentially calculating the expected value of the indicator random variable

corresponding to different thresholds. A multi-valued indicator variable can be defined as:

$$I(\boldsymbol{u}, z_i) = \begin{cases} i & if\ k(\boldsymbol{u}) \le z_i\ \forall i = 1,...,NT \\ NT+1 & if\ k(\boldsymbol{u}) > z_i \end{cases} \tag{2.4}$$

Hence the indicator transform discretizes a continuous variable (such as permeability) into classes or categories. The expected value of the indicator corresponding to a particular category is:

$$E\left\{I(\boldsymbol{u}, z_i\right\} = \text{Prob}\left\{k(\boldsymbol{u}) \le z_i\right\} = F_k(z_i) \tag{2.5}$$

The indicator coded data is used to infer the experimental variogram at each threshold, allowing the usage of different heterogeneity models (variograms) for different thresholds. Then, at a particular location, the conditional expectation of the indicator random function for each threshold is determined by applying indicator kriging with the available indicator coded conditional information.

$$I^*\left(\boldsymbol{u}_\alpha; z_i \middle| (n)\right) = F_k^*\left(z_i \middle| (n)\right) = \sum_{i=1}^{n} \lambda_\alpha(\boldsymbol{u}).i(\boldsymbol{u}_\alpha; z_i) \tag{2.6}$$

where $i(\boldsymbol{u}_\alpha; z_i)$ is the indicator coded data at location $\boldsymbol{u}_\alpha$, with n conditional data; and the weights $\lambda_\alpha(\boldsymbol{u})$ are obtained by solving a kriging system that utilizes indicator covariances:

$$\sum_{\beta=1}^{n} \lambda_\beta(\boldsymbol{u}, z_i) C_I\left(h_{\alpha\beta}; z_i\right) = C_I\left(h_{\alpha o}; z_i\right), \forall \alpha = 1,...,n \tag{2.7}$$

The probabilities (conditional expectations) for the local conditional distributions are evaluated at a limited set of thresholds. Therefore, interpolation and extrapolation methods are required to obtain a continuous conditional cumulative distribution function. Interpolation between the thresholds and tail extrapolations can be obtained by applying different approaches such linear or hyperbolic interpolation/extrapolation or using tabulated values.

Following the sequential simulation approach, a realization of the target reservoir model is generated by sequentially sampling from the local conditional distributions following a

random path, where the previously sampled values become conditional information for the construction of subsequent local conditional distributions. The process is repeated until all the uninformed locations in the model are populated. Monte Carlo or other sampling technique can be applied on the local conditional distributions. Multiple realizations of the target reservoir can be obtained by altering the random path and/or changing the random draw from the local conditional distributions.

The application of the Indicator sequential simulation approach can be summarized by the following:

1. Select appropriate thresholds consistent with the spatial phenomena.
2. Indicator code the data corresponding to different thresholds
3. Infer indicator variogram/covariance model(s) for different thresholds.
4. Define a random path to visit all uninformed locations. On each subsequent location of the random path apply the following sub-procedure:

4.1. Calculate the conditional expectation of the indicator random function for all the thresholds, applying indicator kriging with the available indicator coded conditional information.

4.2. Correct for order relations (non-monotonicity of the distributions) on evaluated probabilities (conditional expectations).

4.3. Randomly sample a value from the local conditional distribution. In the sampling process, use interpolation/extrapolation methods to model a continuous ccdf from the discrete probabilities evaluated at the thresholds.

4.4. Include the sampled value in the list of conditional information for subsequent estimations.

5. A single realization of the target reservoir is obtained after all the uninformed locations have been visited following the random path. To generate multiple realization repeat step 4 with different random paths.

An indicator sequential simulator has been implemented on C++ language, and validated with other algorithms available on public domain. This algorithm is the base code for the probability updating method utilized in the research project. In subsequent sections, additions including modules for gradual deformation of geological models, interface with flow simulators, and optimization schemes are also developed.

Gradual deformation of geological models using dynamic data

Honoring the geological model is an important objective during the generation of static geological models; however, it is commonly forgotten during the integration of dynamic information. During the final stages of reservoir modeling, the history matching process, the perturbations or modification in the model should be performed to ensure a match to the flow history, while preserving the geological model of heterogeneity. In this research, that goal is accomplished by applying a probabilistic approach for gradual deformation of geological models. The gradual deformation is obtained by systematically perturbing the local conditional distributions with a deformation parameter, r_d that is calibrated using the available dynamic information.

Perturbation of Local Conditional Distributions

The gradual deformation starts with a particular realization of the target reservoir, such that all locations in the reservoir, cells or nodes have attribute values. In the case of continuous variables, the initial realization is transformed into an indicator random field such that the value at a particular location falls within an indicator category referred to as the initial class, z_I. The initial realization is obtained by following a particular random path through the reservoir model, sequentially generating the local conditional distribution at each location conditioned to the original data and previously simulated values and randomly sampling values from the constructed distributions.

During the gradual deformation of the geological model, the local conditional distributions are perturbed using a deformation parameter r_d. The optimum value of rd is

calibrated in such a fashion as to minimize the deviation from the observed production history. In addition, the random path along which the visited as well as the random draws from local conditional distributions are modified in order to search for a global optimum for r_d. Different perturbation schemes for establishing the optimal r_d have been evaluated in order to define the methodology that better fits the objectives of the research.

In the first perturbation scheme, the deformation parameter, r_D reduces the probabilities of all indicator categories in the local conditional distribution, except that of the initial class, z_I, which is proportionally increased. In this case the deformation parameter multiplies the probabilities of the off-class indicator categories and the probability of the initial class always increases (or remains the same for $r_d = 1$). This perturbation can be written in terms of conditional probabilities as:

$$F_k^{'}(z_i|(n)) = \begin{cases} r_D \cdot F_k(z_i|(n)) & i \neq I \\ 1 - \sum_{\substack{j=1 \\ j \neq I}}^{NT+1} r_D \cdot F_k(z_j|(n)) & i = I \end{cases} \tag{2.8}$$

where $F_k^{'}(z_i|(n))$ is the perturbed local conditional probability. A deformation parameter of value zero will generate a distribution with probability 1 for the initial class, z_I, ensuring the reproduction of the initial realization. A deformation parameter of value one will recover the local distribution conditioned to geological information and hence an independent realization from the original conditional distribution $F_k(z_i \mid (n))$ is sampled. Since the deformation parameter only increases the probability of the initial class, the maximum deformation of the model is rather small, slowing the process of gradual deformation. This drawback encouraged the evaluation of other perturbation schemes. Figure 2 shows the effect of the deformation parameter on the geological model.

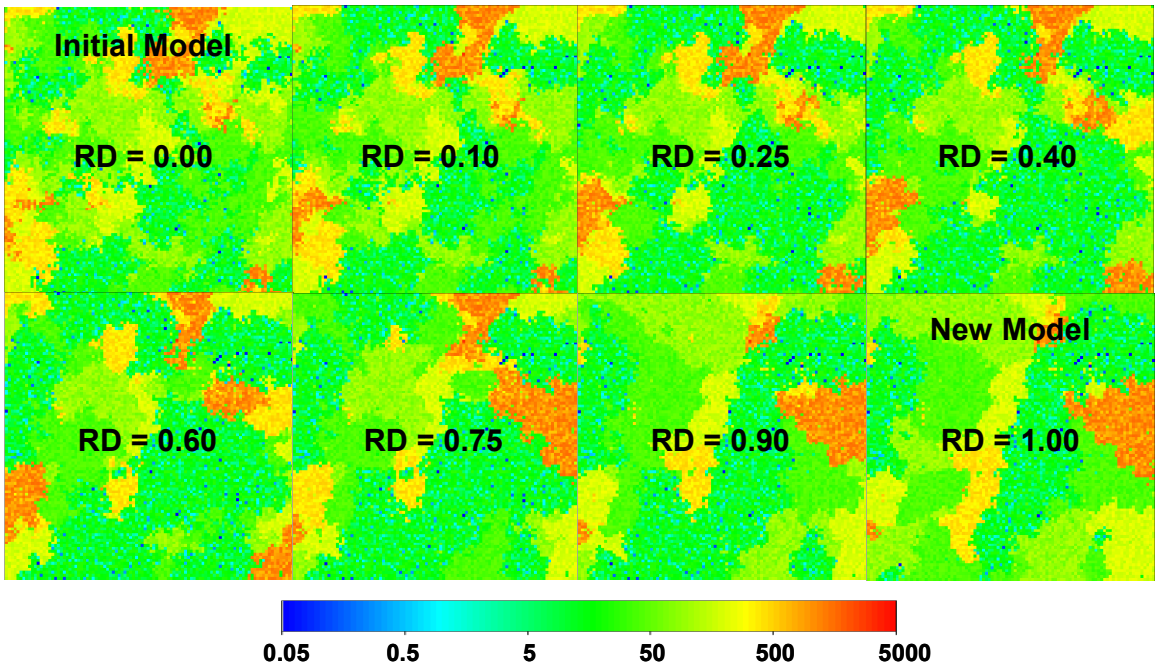


Fig 2.2: Example of gradual deformation of geological models by probability perturbation method – First scheme. A single parameter, r_d determines the magnitude of the perturbation in this model.

In the second perturbation scheme, the deformation parameter reduces the probability of the initial class in the local distributions, while the probabilities of the other indicator categories increase proportionally. In this case the probability of the initial class is always reduced (or remains the same for $r_d = 1$). This perturbation scheme can be written as:

$$F_k^{'}(z_i|(n)) = \begin{cases} r_D \cdot F_k(z_i|(n)) & i = I \\ \dfrac{F_k(z_i|(n))\left(1 - r_D \cdot F_k(z_I|(n))\right)}{\sum_{\substack{j=1 \\ j \neq I}}^{NT+1} F_k(z_j|(n))} & i \neq I \end{cases} \tag{2.9}$$

The perturbed conditional probability for $i \neq I$ is written such that the perturbed probability for a particular class is scaled according to its initial value. A deformation parameter value of zero will generate a distribution with probability 0 for the initial class, producing a notable deformation of the geological model. A deformation parameter of

value one will recover the local distribution conditioned to geological information. Now, the deformation parameter only decreases the probability of the initial class, ensuring a large deformation of the geological model, speeding the process of gradual deformation. However, this perturbation scheme does not allow the reproduction of the initial realization and consequently the deformation process is no longer systematic and controlled.

In the third perturbation scheme, the two previous schemes were combined to ensure a more controlled, but at the same time fast gradual deformation of the geological model. In this case the probability of the initial class has a range of variation from 1 to 0 (for r_d = 0 to r_d = 1), ensuring the reproduction of the initial realization for r_d = 0.0, the recovery of the distribution conditioned to geological information for r_d = 0.5, and rejecting the initial class, generating a new realization and corresponding large deformation for r_d = 1.0. In this approach the range of variation of the deformation parameter, r_d, is divided in two intervals, values below and above 0.5. For values of r_d below 0.5 the first perturbation scheme is applied using a transformed value of r_d, $r_{d'} = 2 \times r_d$. For r_d values above 0.5, the second perturbation scheme is applied with the transformed value of r_d, $r_{d'} = (2 - 2 \times r_d)$. The perturbation scheme can be written as:

$$r_D^{'} = \begin{cases} 2r_D & r_D \le 0.5 \\ 2 - 2r_D & r_D > 0.5 \end{cases}$$

$$F_k^{'}(z_i|(n)) = \begin{cases} r_D^{'} \cdot F_k(z_i|(n)) & i \ne I; r_D \le 0.5 \\ 1 - \sum_{\substack{j=1 \\ j \ne I}}^{NT+1} r_D^{'} \cdot F_k(z_j|(n)) & i = I; r_D \le 0.5 \\ r_D^{'} \cdot F_k(z_i|(n)) & i = I; r_D > 0.5 \\ \dfrac{F_k(z_i|(n))\left(1 - r_D^{'} \cdot F_k(z_I|(n))\right)}{\sum_{\substack{j=1 \\ j \ne I}}^{NT+1} F_k(z_j|(n))} & i \ne I; r_D > 0.5 \end{cases} \tag{2.10}$$

Consequently, the probability of the initial class is increased from the original value for r_d values below 0.5; and decreased for rd values above 0.5, ensuring a wider but controlled range of variation in the perturbation of the local distribution. Figure 3 shows the effect of the deformation parameter r_D on the local conditional distribution under the perturbation scheme 3.

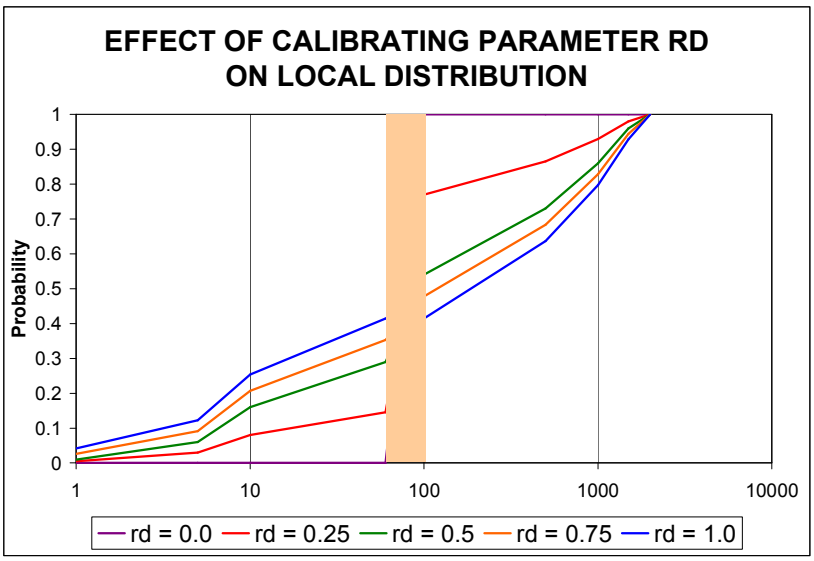


Fig 2.3: Effect of deformation parameter on local distribution for third perturbation scheme. For rd = 0.5, the local distribution conditioned to geological information is recovered.

The above three perturbation schemes focus on gradually transitioning an initial realization to a new realization or proposal. Multiple equiprobable proposals can be generated using sequential indicator simulation, simply by modifying the random path and/or the set of random drawings for the local conditional distributions. However, in the three previous schemes, starting from an initial realization, the transition to only one new realization is evaluated at a time. This implies that only the range of variability between these two limiting realizations is searched for an optimal value of r_d. This limits the range of variation during the gradual deformation of the geological model. A fourth perturbation scheme is proposed to circumvent this drawback.

In the fourth perturbation scheme the transition to two different new realizations are evaluated at the same time. In this case the perturbation introduced by the deformation parameter in the local distributions is controlled in order to generate a gradual transition between the initial realization and two different equiprobable realizations obtained with different random paths and different sampling draws. In this scheme the initial realization is reproduced for r_d = 0.5 (probability of the initial class is 1), and the distributions conditioned to geological information corresponding to the two proposals are recovered for r_d values of 0.0 and 1.0. Consequently, in this approach the range of variation of the deformation parameter, r_d, is divided in two intervals, values below and above 0.5. The transitions between the initial realization and the first and second proposals are obtained with values of r_d below and above 0.5 respectively. These transitions follow the first perturbation scheme using a transformed value of r_d, $r_d' = 1 - 2\, r_d$, for $r_d \leq 0.5$; and $r_d' = 2\, r_d - 1$, for $r_d > 0.5$. The perturbation scheme can be written as:

$$r_D^{'} = \begin{cases} 1 - 2r_D & r_D \leq 0.5 \\ 2r_D - 1 & r_D > 0.5 \end{cases}$$

$$F_k^{'}(z_i \,|\, (n)) = \begin{cases} r_D^{'} \cdot F_k^a(z_i \,|\, (n)) & i \neq I; \quad r_D \leq 0.5 \\ 1 - \sum_{\substack{j=1 \\ j \neq I}}^{NT+1} r_D^{'} \cdot F_k^a(z_j \,|\, (n)) & i = I; \quad r_D \leq 0.5 \\ r_D^{'} \cdot F_k^b(z_i \,|\, (n)) & i \neq I; \quad r_D > 0.5 \\ 1 - \sum_{\substack{j=1 \\ j \neq I}}^{NT+1} r_D^{'} \cdot F_k^b(z_j \,|\, (n)) & i = I; \quad r_D > 0.5 \end{cases} \tag{2.11}$$

Where F_k^a and F_k^b are the probabilities corresponding to the two proposals a and b.

The fourth scheme compared to the first three, evaluates twice the number of proposals, increasing the rate of convergence and reducing the probability of getting trapped on local minima.

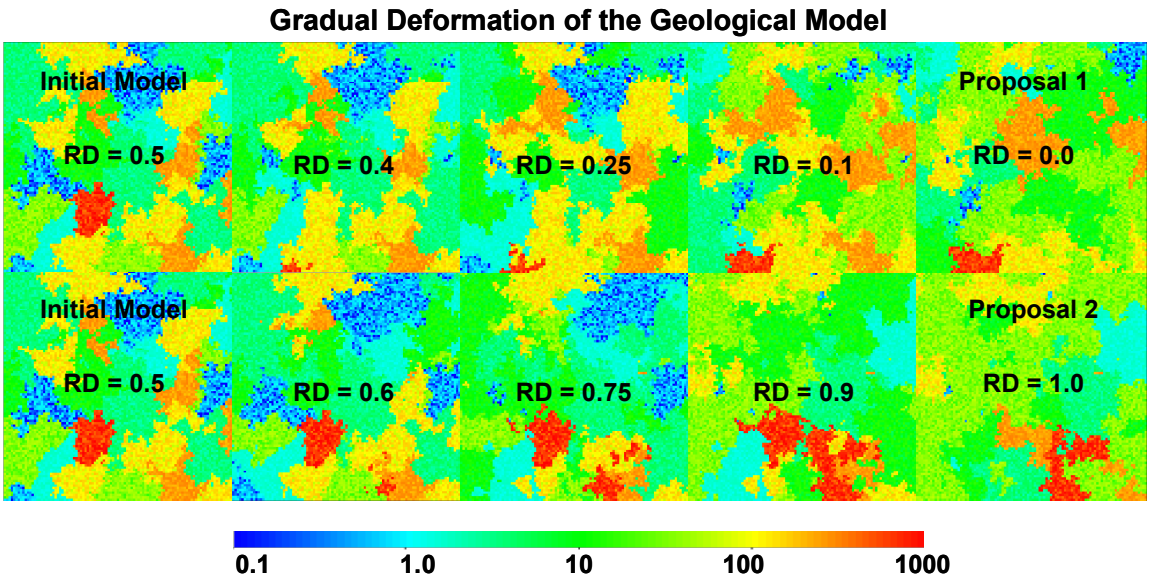


Fig 2.4: Example of gradual deformation of geological models by probability perturbation scheme four. A single parameter, r_d determines the transition between an initial realization and two proposals. The range of variation of the geological model is enlarged under this scheme.

Conditioning of Local Distribution to Dynamic Information

Now that the methodology for the perturbation of local conditional distributions in geological models has been defined, the next step is to calibrate the deformation parameter on the basis of the observed production history of the reservoir. A primary goal is to estimate the local probability distributions of the geological event A, (permeability, porosity, etc) conditioned to the dynamic information, C, i.e. P(A|C). This requires the implementation of an optimization scheme to calibrate the deformation parameter, and the development of interfaces between the geological modeling algorithm and a flow simulator. Figure 5 shows the impact of the deformation parameter on the flow response of the perturbed geological model.

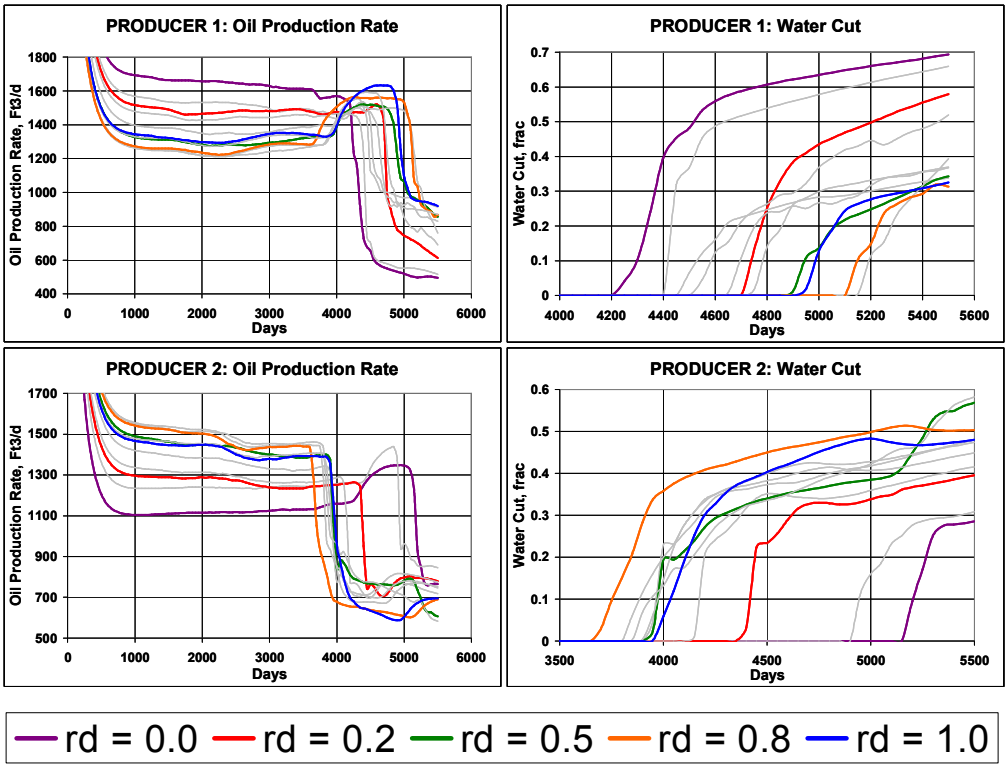


Fig 2.5: Effect of varying the r_D parameter on: a) Oil production rate at Producer 1; b) Water cut at Producer 1; c) Oil production rate at Producer 2; d) Water cut at Producer 2.

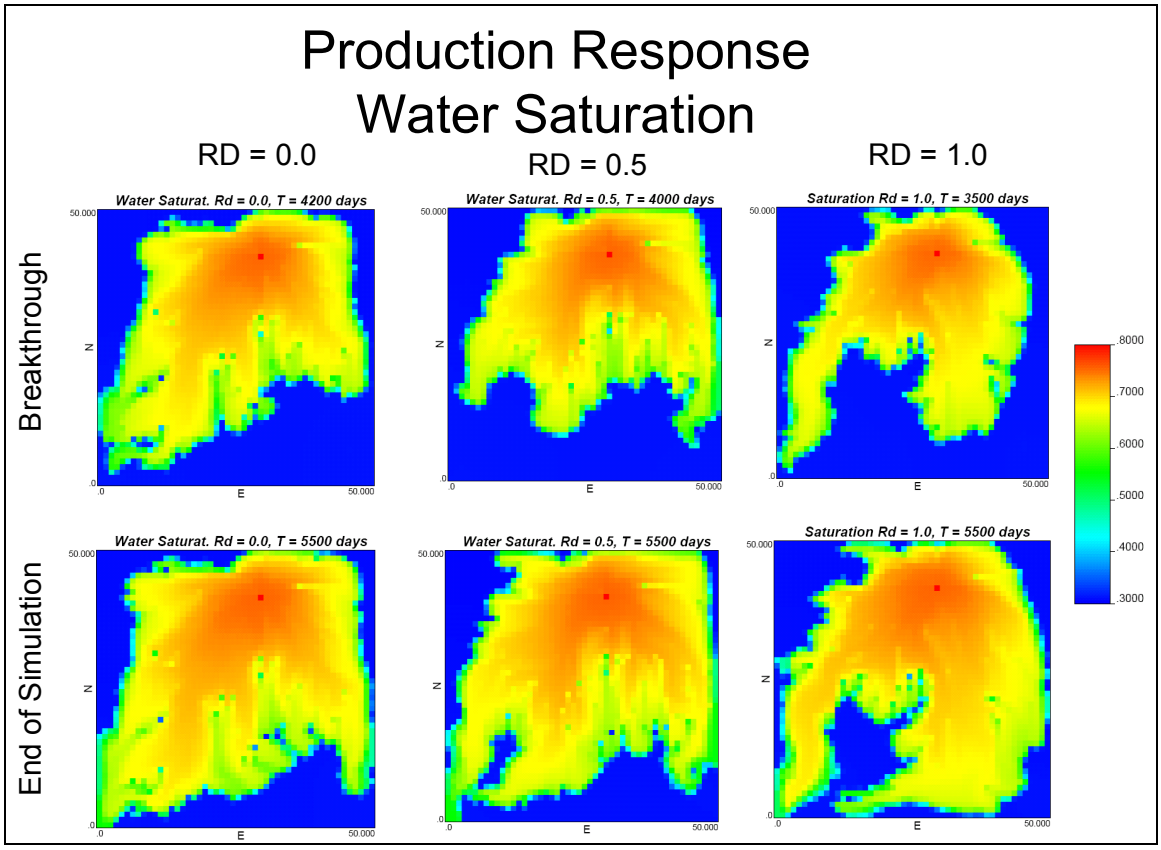


Fig 2.6: Sensitivity of water saturation distribution in the reservoir at 4000 days and 5200 days to r_D parameter.

The deformation parameter r_D is calibrated using the Dekker-Brent iterative optimization procedure where the objective is to improve the fit between the flow response of the model (from the simulator) and the production history. The Dekker-Brent algorithm is an inverse parabolic interpolation method that has the advantage of being a non-gradient based approach that only requires the calculation of the objective function corresponding to different values of the deformation parameter. The algorithm yields an optimal value of the deformation parameter, r_D^* (abscissa), corresponding to a minimum value of the objective function, $\Delta O = f(r_D^*)$ (ordinate). Three abscissa values are required, a, b and c with the corresponding values of the objective function f(a), f(b) and f(c); and b chosen such that a < b < c and f(a) > f(b) < f(c). The estimated location of the abscissa (r_D^*) with

the apparent minimum ordinate (objective function) is calculated by fitting a parabola through these three points.

$$r_D^* = \frac{\dfrac{(b+c)f(a)}{(a-b)(a-c)} + \dfrac{(a+c)f(b)}{(b-a)(b-c)} + \dfrac{(a+b)f(c)}{(c-a)(c-b)}}{2\left(\dfrac{f(a)}{(a-b)(a-c)} + \dfrac{f(b)}{(b-a)(b-c)} + \dfrac{f(c)}{(c-a)(c-b)}\right)}$$

The next figure illustrates the process of the inverse parabolic interpolation.

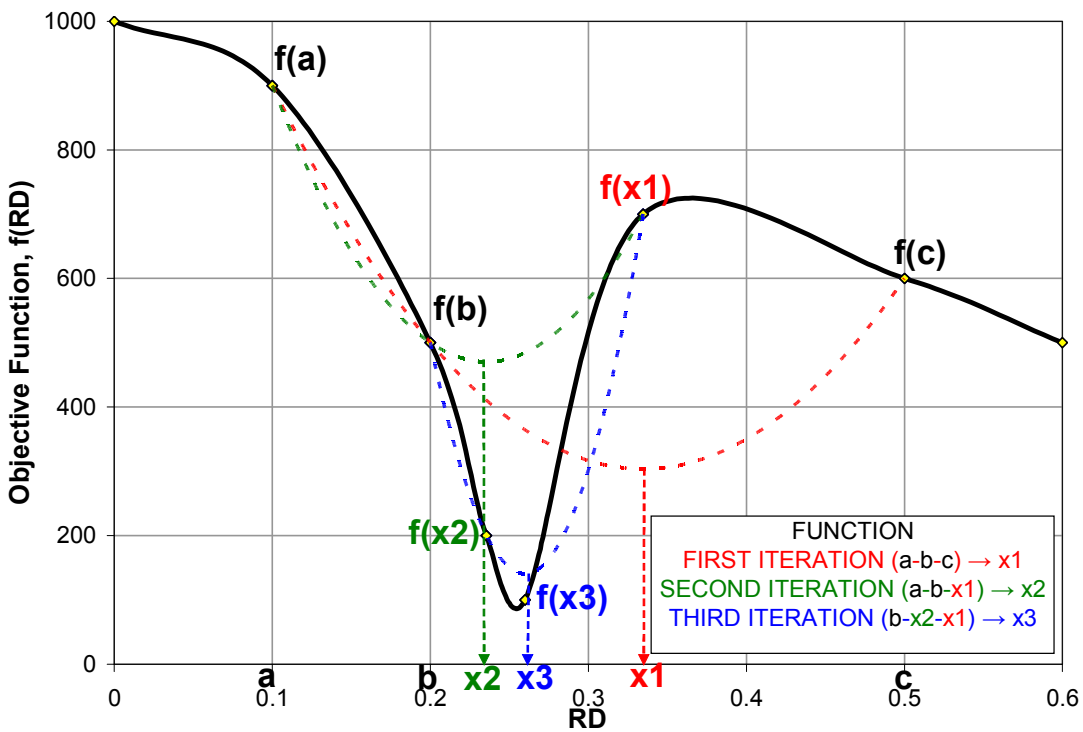


Fig 2.7: Example of convergence of an objective function for a single-parameter problem using the Dekker-Brent inverse parabolic interpolation algorithm.

An inverse parabola (red) is fitted through the initial values of rd a, b and c and the correspondent objective functions for these values, f(a), f(b) and f(c), resulting in the first apparent optimal value of rd = x1. Then, the real objective function correspondent to x1, f(x1) is calculated and based on the result, the three points for the next parabolic

interpolation (green) are selected (points a, b and x1). This iterative procedure continues until a minimum in the objective function is reached

In this case, the objective function to be minimized is a measurement of the deviation between the simulated production response and the production history. Different production variables can be included in the objective function, including field and well pressures, single phase rates, two phase ratios, and basically any other variable for which a historical record is available. The mismatches of individual variables are normalized in order to level their influence on the objective function. However, in some cases it might be useful to assign higher weights to some production variables in order to emphasize the relevance of their reproduction in the target model. The proposed objective function for N production variables over T time steps is:

$$Obj\ Fun = \sum_{t=1}^{T} \sum_{i=1}^{N} \left[\frac{\left(Sim_{i,t} - Hist_{i,t} \right)^2}{Var\left(Hist_i \right)} \right]$$

Where $Sim_{i,t}$ represents the simulated value of the production variable i at time t, and $Hist_{i,t}$ represents the correspondent historical value for the same variable at the same time. The square of the difference between the simulated and the historical values at a particular time is normalized by the variance of the historical values over time, in order to control the influence of each variable on the objective function. Other objective functions can be easily implemented applying different norms and normalization methods.

Merging the information from dynamic and static sources

At this point, the methodology to estimate the local distribution conditioned to dynamic information, P(A|C), has been presented, i.e. perturbing the local conditional distributions with a parameter calibrated with the production history data. In order to ensure consistency with the geological model during all stages of the history matching procedure, the conditional distribution P(A|C) has to be merged with the distribution inferred from static information, P(A|B). Realizations sampled from the resultant merged distribution P(A|B,C) will honor both the static information (well data and geologic interpretation) as well as the historic production data.

The Permanence of Ratio Hypothesis (Journel, 2002) is the methodology applied to combine the individual distributions conditioned to dynamic and static information. The following distance or information measures a, b, c and x are defined:

$$a = \frac{1 - P(A)}{P(A)} \quad b = \frac{1 - P(A \mid B)}{P(A \mid B)} \quad c = \frac{1 - P(A \mid C)}{P(A \mid C)} \quad x = \frac{1 - P(A \mid B, C)}{P(A \mid B, C)}$$

The quantity a, for example, denotes the relative distance to the event A occurring given P(A). If P(A) is one, the relative distance a is zero. The relative distance a is infinity, if P(A) is zero. The measures b, c and x can be interpreted similarly.

According to the permanence of ratios hypothesis, the relative updating of a simulation event (A) due to a dynamic event (C) remains the same irrespective of the presence of the static event (B). This can be written in terms of a,b,c and x as:

$$\frac{x}{b} = \frac{c}{a}$$

Consequently, the joint probability of the simulation event given the dynamic and static information can be calculated from the elemental probabilities – the prior probability for A : P(A), the conditional probability of A given the dynamic information : P(A|C), and the conditional probability of A given the static information B : P(A|B).

$$P(A \mid B, C) = \frac{a}{a + bc}$$

In this approach, the marginal probabilities of the static data P(B), and that of the dynamic data P(C), which are difficult to estimate in practice, are not required.

Putting things together – a gradual updating procedure for history matching

The calibration of the dynamic parameter r_D and the subsequent merging of conditional probability distributions requires the implementation of an interface between the geological modeling algorithm and the flow simulator. In this project, the geological modeling algorithm (Sequential Indicator Simulation) is the main program that has been expanded to include all the tasks in the probabilistic approach for dynamic data

integration. The flow simulator (@Eclipse) is also executed within the program. The task of combining the conditional probability distributions P(A|B) and P(A|C) into a joint-conditional distribution P(A|B,C) is also implemented within the main program. The geological model used in the flow simulator to calibrate the deformation parameter r_D is sampled from this jointly conditioned distribution, P(A|B,C). The complete probability updating procedure therefore consists of:

- Performing indicator kriging and obtaining the conditional distributions P(A|B) at each unsampled location in the reservoir. Sample a realization from the P(A|B) by sequential simulation
- Corresponding to that realization of the permeability field and making an initial guess for r_D obtain the corresponding P(A|C)
- Merge P(A|C) with P(A|B) to obtain P(A|B,C) and sample a realization from P(A|B,C)
- Perform flow simulation and obtain objective function. Revise estimate of r_D using Dekker-Brent approach and repeat until objective function is minimized.

The reproduction of historical production data is a complex non-linear inverse problem. This implies that the probability updating cannot be accomplished in a single perturbation loop starting from an initial realization of the permeability field; calibrating an optimal r_D value and obtaining an updated probability distribution reflecting the dynamic characteristics of the reservoir. Instead, a multi-loop iterative process is required to update the geological model using the dynamic data.

A Markov-Chain is a stochastic updating procedure where the parameter state at any step of the procedure is assumed to be dependent only on the state immediately prior to that step. Thus the proposed realization at any stage of the process depends only on the preceding realization in the sequence, and the convergence towards the desired target depends on carefully specifying the transition from one realization to the next one, i.e. the

methodology for the new proposed realization. In this case, the parameter r_D controls the transition of the permeability value at a location from one category to the next.

In the implemented Markov chain approach, at every updating step, from iteration step l to step $l+1$ of an outer loop, the probability distributions conditioned to dynamic and static information, $P(A|B,C)^l$, is obtained by applying the permanence of ratio hypothesis to combine distributions conditioned to static and dynamic information. The distribution $P(A|B)^l$ is obtained from geological data and heterogeneity model and the distribution conditioned to dynamic information, $P(A|C)^l$ is estimated knowing the indicator category at each location from the realization sampled from $P(A|B,C)^\ell$, the prior distribution P(A) and the deformation parameter, r_D, calibrated using the Dekker-Brent iterative optimization procedure. At the end of each inner Dekker-Brent loop the converged distribution $P(A|B,C)^l$ is used to sample the initial realization for the next outer loop, $l+1$, until global match to the historic data is attained. The calibration of the deformation parameter r_D to honor the available dynamic information thus represents the internal optimization scheme. The converged model and realization at the end of the inner loop is used as the starting realization for the next sequence of inner Dekker-Brent optimization runs to determine the conditional distribution P(A|C) and that constitutes the outer optimization loop.

Even though the two-loop Markov chain procedure ensures global convergence, the introduction of multiple sets of inner optimization schemes requires multiple evaluations of the flow response. However, the dynamic-calibrated gradual deformation methodology renders the history match process faster and more controlled, increasing the consistency between the initial and the proposed realizations at every step, and improving the rate of convergence of the objective function.

The implemented interface between the geological modeling algorithm and the flow simulator is summarized in the following steps:

1. The geological modeling algorithm generates a file with a realization of the target permeability model in the appropriate format. This file will be used by the flow simulator as an include file.
2. The modeling algorithm invokes the flow simulator. The flow simulator (ECLIPSE®) is run, generating an output file with the flow response.
3. The simulated flow response is read from the output file and the objective function is evaluated.
4. The Dekker-Brent optimization is implemented in order to generate a new realization of the geological model.

Applications

An approach that uses a probability perturbation method for gradual deformation of geological models conditioned to dynamic information has been presented. This approach, compared to other perturbation methods, offers the important advantages of preserving the prior geological heterogeneity model and simplifying the history match process to a single (or few) parameter(s) optimization problem.

Preliminary evaluations of the algorithm for dynamic data integration using the proposed probabilistic approach have been pursued with the synthetic case described in the following table.

Table 2.1: Description of the synthetic simulation case used to evaluate the probabilistic dynamic data integration algorithm.

SIMULATION PROPERTY/DESCRIPTION	VALUE
Simulation Model	Black Oil
Solution	Implicit

Simulation Period, years	2
Grid (Cartesian)	50x50x5
Active Grid blocks	12500
Grid block dimensions, ft^3	80x80x4
Porosity	0.22
Kx = Ky (Mean – Std Dev), md	200 - 250
Kz/Kx	0.15
Saturation Pressure, psi	5064
Water-Oil Contact, ft	9000
Gas-Oil Contact, ft	4000
Reference Depth, ft	7300
Initial Pressure @ 7300 ft, psi	6000
Residual Water Saturation	0.18
Residual Oil Saturation	0.24
Oil Relative Permeability Endpoint	0.7
Water Relative Permeability Endpoint	0.5
Oil Gravity (API)	35
Water Injectors	1
Injection – Control Rate, Stb/day	5000
Injection – BHP upper Limit, psi	8000
Oil Producers	2
Production – Control BHP Lower limit, psi	2000
Production – Minimum rate, Stb/day	10

Figures 8-12 describes the convergence of the simulated field pressures to the historic data.

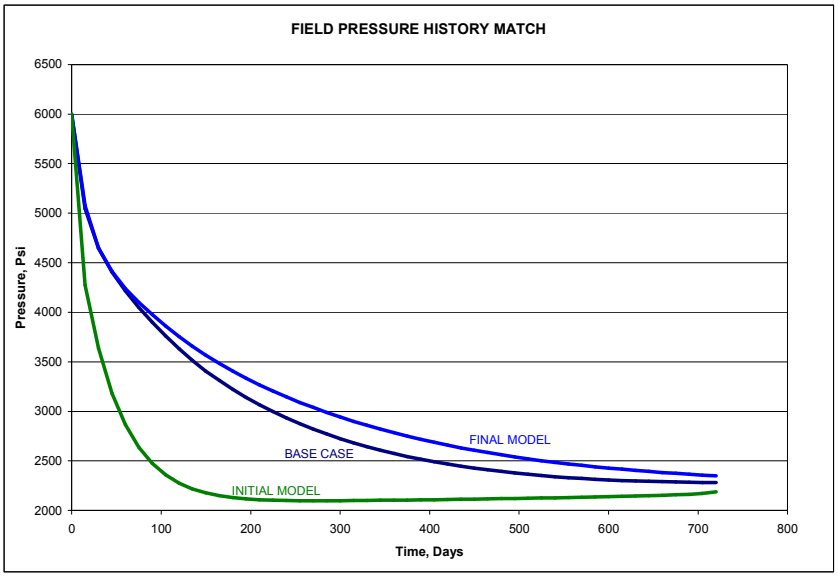


Fig 2.8: Field Pressure History Match obtained with the probabilistic dynamic data integration algorithm for the synthetic case study. Results after forty flow simulation runs, distributed on 5 outer iterations of 8 inner Dekker-Brent iterations each.

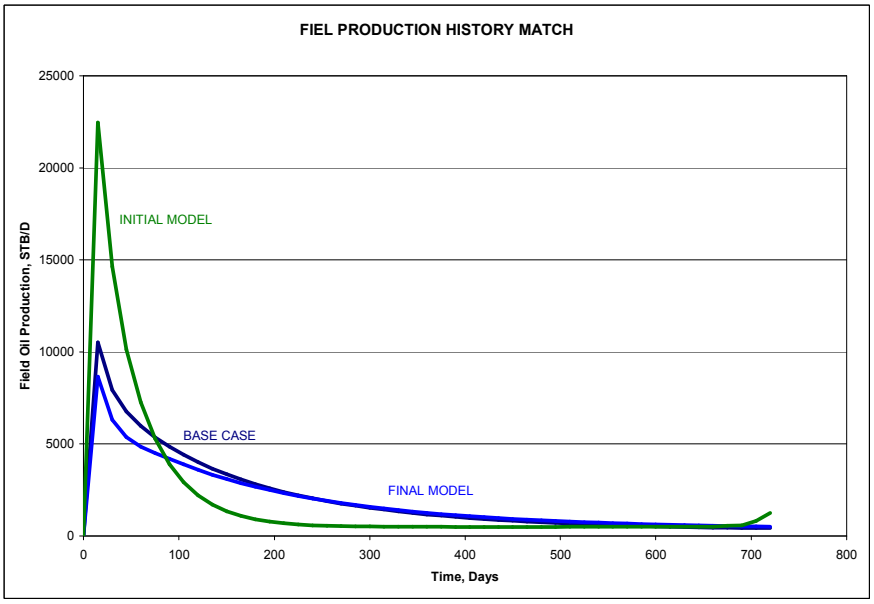


Fig 2.9: Field Production History Match of synthetic case using the probability perturbation method. Final model is obtained after 40 simulation runs.

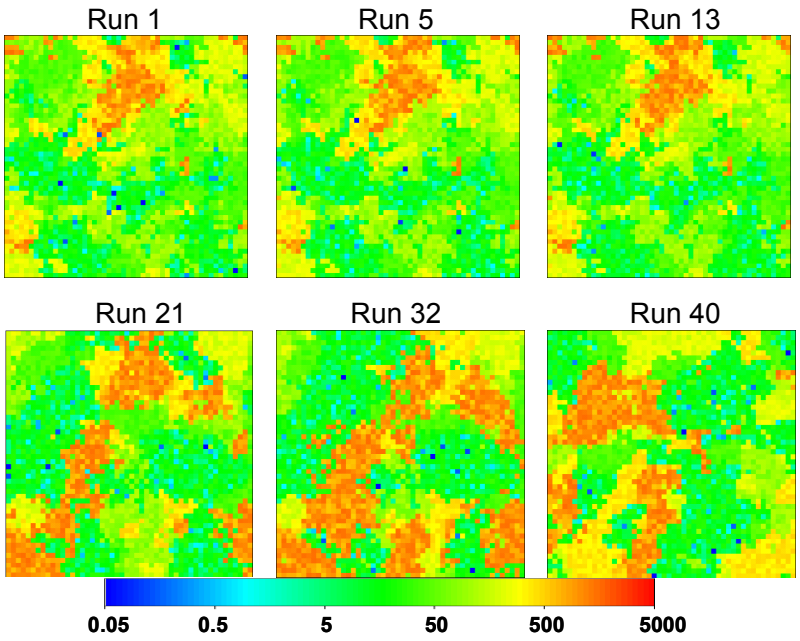


Fig 2.10: Example of Gradual Deformation of geological models, obtained with the probability perturbation method. Variations in the third layer of the geological model through 40 flow simulation runs are presented.

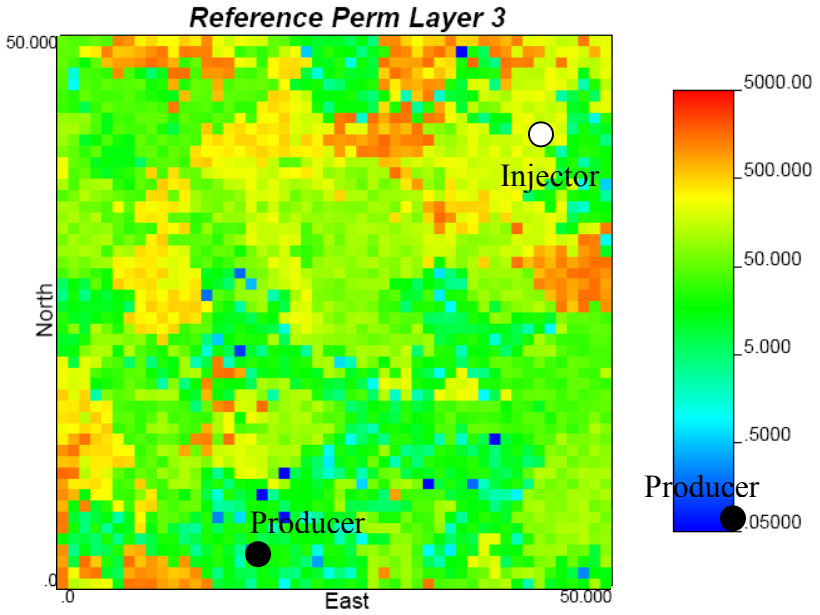


Fig 2.11: Third layer of the reference geological model used in the synthetic case to generate the historical production data.

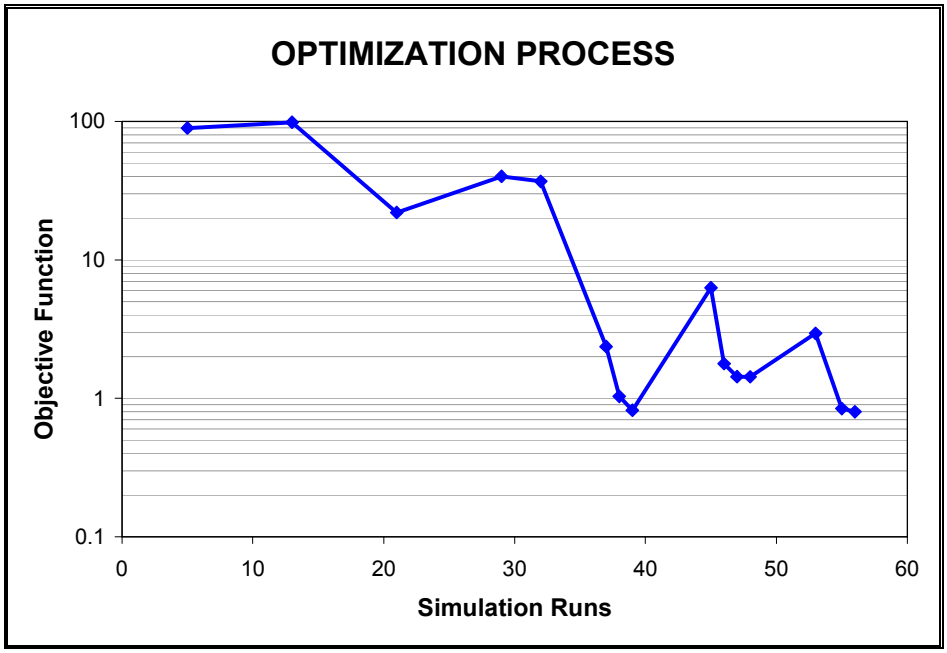


Fig 2.12: Example of convergence of the objective function applying the probabilistic approach to history matching.

A computer code implementing the following steps for history matching has been developed:

1. An indicator sequential simulator is used to generate an initial stochastic realization of the target reservoir model and calculate the local probability distributions conditioned to static information.
2. A Markov chain iterative updating process is started with the initial realization. The Markov chain forms the outer loop of the procedure and every outer step or outer iteration includes the following sub procedures:

2.1. Generate new random paths and sets of random draws to sample from the local conditional distributions. The random paths and the sampling draws are fixed during each outer iteration, but changes from one outer iteration to the next.

2.2. Evaluate the Objective function at different values of the deformation parameter, r_d, screening the whole range of variability [0, 1]. Usually 5 different values are enough to get started. For each value of r_d, a different permeability model is

obtained by merging with P(A|B) and that model is run in the flow simulator; in order to obtain the objective function.

2.3. Pick the value of the deformation parameter with the minimum objective function and start the calibration process of r_d with the dynamic data using the Dekker-Brent iterative algorithm. This calibration process is called the inner loop, and the number of inner steps or inner iterations can be fixed or controlled by a tolerance in the change of the objective function in consecutive steps.

2.4. Use the best model (with the minimum objective function) to update the stochastic realization. When the best model is obtained with a deformation parameter of 0.5, no updating is required (the realization remains invariant). This is the final step of the outer loop.

3. Repeat step 2 (outer loop) until a tolerance in the objective function (history match) has been reached or for a fix number of outer iterations.

4. Print out final permeability realization with the corresponding flow response.

3. APPLICATION OF PARALLEL AND DISTRIBUTED COMPUTING

The following section describes

1) Some concepts in parallel and distributed computing applied to reservoir simulation.
2) Choice of the reservoir simulator and the important features of the chosen simulator.
3) Selection of a parallel and distributed computing scheme considering the specific requirements of the project & preliminary verification of the availability of the required features in the selected simulator.

Concepts in parallel and distributed computing applied to reservoir simulation

1) Parallel computing

In parallel computing, a task is subdivided among many computer processors such that if done efficiently the total computational time scales with the number of CPUs assigned. Parallel computer architecture have generally been based on the following major concepts:

a) SIMD – Single instruction multiple paths.

b) MIMD – Multiple instruction multiple data paths

The MIMD architecture has emerged recently and is based on the principle that each CPU can operate independently, with frequent synchronization among the CPUs.

MIMD architecture has three basic memory configurations:

- Flat shared memory: Allows access to all memory by each of the CPUs. This is costly and cannot scale well above 10 CPUs.
- Multi-level shared memory: Based on the concept of cache i.e. sophisticated software or hardware keep track of the current location of data within the global memory of the system.
- Distributed memory. Has become the most popular for massively parallel computers. These machines are based on the use of RISC-based CPUs, some with attached vector processors. Performance of each CPUs varies from a few million floating-point operations per second to more than 100 Mflops.

Recently the parallelization of reservoir simulators has been accomplished on distributed memory parallel computers. This parallelization has been accomplished on both MIMD and SIMD. A critical analysis of parallel computing with respect to reservoir simulation has been done by Killough (SPE 26634, 1993).

Message passing between the processors is critical to the performance of the distributed memory parallel computing. Two aspects of communication are important: **latency** and **bandwidth.** Latency may be defined as the time taken for setting up pathways between the processors and identifying locations that are involved in the transfer of data. In order to reduce the latency, several messages may be packed together, so that communication between the processors is reduced. In that case the bandwidth becomes critical.

Theoretically, for an N*N grid solved on P processors :

1) Total work time is proportional to N*N/P (Assuming complete parallelizing of the simulator)
2) Communication time is proportional to N

As the problem size increases :

1) Latency is affixed cost
2) Communication time increases with N
3) Work time increases with N*N

For inter-node connections, there are in general four options

a) Fast Ethernet
b) Gigabit
c) Myrinet
d) Quadrics

Table 3.1: Bandwidth and Latency for various switches of a PC cluster

Switch	**Bandwidth (Mbytes/sec)**	**Latency (microsecond)**
Fast Ethernet	12	150
Gigabit	128	26-12
Myrinet	421	7
Quadrics	400	5

The Myrinet and Quadrics have high bandwidth and are most suited for reservoir simulation. The research cluster used for this project has a gigabit switches between the different processing nodes. A reduction in parallel performance is therefore likely.

Amdahl's law

The efficiency of a parallel program can be assessed by comparing the performance speed of a parallel algorithm with that of a program run on a single node. That measure known as speedup can be written as:

$$Speedup = \frac{1.0}{\left(s + \frac{p}{n}\right)} \tag{3.1}$$

s = serial fraction of program work including the communication overhead

p = parallel fraction of program work

n = number of parallel processors

To achieve better efficiency the subdivided tasks among the processors should be large as compared to the inter-processor communication. Amdahl's law implies that even the smallest portion of the model must be parallelized to achieve reasonable parallel

efficiencies for a massively parallel processor with one hundred or more processors. In the notation of the above equation, making the fraction part to be higher is better than keeping s to be high since the quantity p is divided by n the number of processors and the net gain in speedup may be considerable. Serial code sections become significant if a high number of CPUs (i.e. 100 + CPUs) are used in a single run.

It must be noted that parallel algorithms *are generally less optimized* than their serial counterparts and hence the performance of a parallel code running on a serial machine may often not be as good as the serial code running on the same serial machine.

There are certain challenges using parallel computation for reservoir simulation:

a) Firstly the non-recursive nature of existing linear solutions techniques for solving the sparse matrices encountered in reservoir simulation renders them unacceptable for massively parallel architectures. Considerable research is currently being focused on developing robust parallel linear equation solutions. Killough (SPE 26634, 1993) proposed a complex preconditioning scheme for conjugate residual type iterative methods such as ORTHOMIN.
b) The trade-off between load balancing and global data structure has yet to be thoroughly investigated.
c) Well and facility constraints and production optimization are generally implemented using serial algorithms and these may lead to severe serial bottlenecks.

One of the key issues affecting the performance of parallel computing is **load balancing**. Load imbalances can severely reduce the efficiency of parallel computing. If one of the processor within a parallel job is multi-tasking or has not finished the allotted load, then other processors within the parallel job will end up waiting. Hence parallel jobs **must** have dedicated resource (i.e. the processors to be used for parallel processing should not

be multi-tasking or processing other jobs). The point being that if the speed of one processor slows down, other processors would be waiting reducing the efficiency of the tasks that are spawned on the other processors as well. Two common load-balancing methods are

- Static, data structures are allocated before beginning the computation
- Dynamic, restructuring of data structures as the computation proceeds

Wheeler and Smith (SPE 19804, 189) dealt with load imbalances brought about by irregularly shaped grids through a redistribution of active cells among the processors. This technique works well in cases where the solution depends linearly on the number of grids blocks. But for cases where the solution is related non-linearly to the sub dimensions associated with the processor arbitrary redistribution of active cells can further exacerbate the problem. Killough (SPE 26634, 1993) distributed the computation work at each grid block within the simulation domain to all the processor nodes and assigned to each node its proportion of the calculations. A substantial improvement appears to be possible but unresolved issues still exist:

a) The number of grid blocks will seldom be a perfect multiple of the number of nodes.
b) The number of iteration will vary for each grid block. This implies that to achieve better load balancing, a prior idea of number of iterations and hence allocation of work to the computer nodes is necessary.
c) The communication overhead may overshadow the performance of the algorithm.

Killough (SPE 29102, 1995) later proposed a receiver-initiated dynamic load-sharing algorithm to achieve high parallel efficiencies. It adapts the workload on each node in a dynamic way and can react to any sudden perturbation that can appear during the simulation. The algorithm suffers from the fact that there is significant loss in performance for a small number of processors since one processor is dedicated for supervising and monitoring the computation at the remaining nodes.

2) Distributed computing

In a distributed configuration, the processors do not share memory or clocks. Distributed computing systems group individual processors together via network connections and pool the computing resources in order to accomplish CPU intensive computation. The network provides a means by which client and host machines communicate, sharing computed information or passing information that is required by the computations. Local area networks (LAN) of PCs connected by Ethernet connections are ubiquitous and are good examples of distributed computing. Distributed computing requires coordinating the efforts of a collection of processing nodes linked together by a network. The common tool used for the external parallelization is PVM (Parallel Virtual Machine). PVM allows the development of the programs that can send "slave" tasks to different cpus in a network.

The use of distributed computing/external parallelization to speed up history matching procedures has been studied by Schiozer and Sousa (SPE 39062, 1997). They showed that sensitivity analysis could be performed efficiently, as well as various direct search optimization techniques can be improved, by the use of external parallelization (or distributed computing). The choice of optimization techniques to be used is quiet important. Methods that use derivatives may have convergence problems. For this reason, Leitao and Schiozer (SPE 53977, 1999) recommends direct search methods that are most robust. Quenes et al (SPE 29107, 1995) applied parallelization techniques to history matching reporting good results. The benefits of external parallelization could be summarized as follows:

- To calculate sensitivities
- To determine the best search directions for direct optimization techniques.
- To launch several simulations in direct search methods.

There are some advantages of using external parallelization (or distributed computing) over parallel computing

- Efficiency of current commercial simulator codes is maintained.
- An existing network of workstations can be used without the need of high investment in parallel computers or in communication.

The main disadvantage is slow data transfer imposed by the LAN communication protocols, but in this kind of work, actual computations are so time consuming that communication time can be neglected for practical applications.

Choice of the reservoir simulator and its important features

Two commercial simulators Eclipse® and VIP® were tested for their parallel computation capabilities. Both have the parallel as well as the flux boundary options available. The flux boundary option is important from the standpoint of distributed computing. The choice of simulator was dictated by the availability of options (outputs) such as the Hessian matrix of the objective function that would help in identifying the sensitive regions. Simopt® - a history-matching module in Eclipse, has the option to report sensitivities. The commercial simulator VIP doesn't have similar options. Keeping in view this limitation of VIP, the following section dwells on important features of the Eclipse flow simulator that facilitates parallel computation and retrieval of flux boundary conditions.

Eclipse (General)

The Eclipse simulator suite consists of two separate simulators: Eclipse 100 Specializing in black oil modeling, and Eclipse 300 specializing in compositional modeling. Eclipse 100 is a fully implicit, three phase, three dimensional, general-purpose black oil simulator with gas condensate options. Eclipse 100 can be used to simulate 1, 2 or 3 phase systems. Two-phase options (oil/water, oil/gas, gas/water) are solved as two component systems saving both computer storage and computer time. Eclipse 300 is a compositional simulator with cubic equation of state, pressure dependent K-value and black oil fluid treatments. Eclipse 300 can be run in fully implicit, IMPES and adaptive implicit (AIM) modes.

Fully implicit technology (Black Oil)

Eclipse (Black oil) uses the fully implicit finite difference method to ensure numerical stability over long time steps. The non-linear fully implicit equations are solved precisely by reducing all residuals to user set tolerances. Newton's method is used to solve the non-linear equations. The Jacobian matrix is fully expanded in all variables to ensure quadratic (fast) convergence. Most simulators cannot apply fully implicit methods to

large problems. In Eclipse, these restrictions are removed by nested factorization, which solves large problems efficiently and reliably.

Adaptive Implicit and IMPES (Compositional)

In a compositional model where the number of components and hence equations to be solved is greater than say 5 or 6, the cost of performing fully implicit simulations can become prohibitive in terms of both memory and CPU time. In Eclipse 300 this problem is tackled by using an adaptive implicit (AIM) scheme, making cells implicit only where necessary. In Eclipse 300 adaptive implicit, fully implicit or IMPES solution techniques may be selected. For larger compositional studies AIM can be selected.

Parallel Option

The Parallel option allows a single simulation job to be distributed across a number of processors. This allows large simulations to be carried out in shorter time than would normally be the case with the standard simulators. The results obtained using a number of processors will generally agree with single processor solutions within limits of engineering accuracy. The scalability is poorer than one might expect, as the linear solver becomes less efficient when larger numbers of processors are utilized. Eclipse 100 first partitions the reservoir either in the x or the y direction, depending on the outer solver direction. The code is optimized so as to automatically divide the reservoir into domains with approximately equal numbers of active cells.

For reservoirs with a significant number of inactive cells, the default partitioning is not sufficient to load-balance the domains. It is possible to control further the way in which cells are assigned to domains by applying the following formula:

$$\text{Active cell load balance } \% = \frac{\text{total active cells in reservoir}}{max(\text{active cells per partition}) \times \text{number of procs}} \times 100$$

The one important difference between Parallel Eclipse 100 and Parallel Eclipse 300 is in the approach used for the parallelization of the linear solver. Eclipse 100 uses a modified

nested factorization approach in which a 1-dimensional domain decomposition is performed, full coupling of the solution across the reservoir for each linear iteration is maintained, at the expense of increasing the work required per linear iteration by 1.3.
Eclipse 300 allows a 2-dimensional domain decomposition (For large numbers of domains, a two-dimensional decomposition is desirable, **to avoid thin strip domains)** approach.

Flux Boundary Conditions

Flux boundary conditions enable to runs to be performed on a small section of a field using the boundary conditions established from a full field run. These smaller field simulations can be distributed to the different nodes of a cluster. Flows across the boundary of the reduced field are written to a FLUX file at each min-time step of the full field run. The flux file is input to the smaller field simulation as the boundary condition for that simulation.

Instead of using the flows of each phase from the full field run as boundary conditions on the reduced run, an alternative treatment (i.e. alternative set of boundary conditions) using pressure and saturations is available in Eclipse.

Selection of a parallel and distributed computing scheme considering the specific requirements of the project

The proposed approach for history matching utilizes reservoir regions determined by the sensitivity analysis procedure and subsequent perturbation of the local conditional distributions describing the uncertainty in permeability values in that region in order to effect a history match. The perturbation to the probability distributions is performed using an updating parameter termed r_D. There can be two plausible ways of handling multiple "r_Ds" corresponding to multiple domains i.e. parallel or distributed computing. Each has its merits. For parallel computing the first requirement is that the domains be of comparable size. This problem can be addressed by having two different definitions of domains i.e. that updating is done on the domains defined as per the sensitivity criteria while during the running of simulator the domains divided are defined such that they are equal in size. This implies that the flow simulation can consist of multiple r_D regions, and so the problem of r_D optimization of multi-parameter optimization. The resultant algorithm will therefore be complicated, defeating the very objective of the proposed history matching approach.

Using distributed computing, the same problem of r_D optimization can be approached using the 1-D optimization methodology that renders the proposed method efficient.
Each domain can be perturbed at different node of the cluster while targeting the global objective function. Finally the perturbed regions can be put together since the domains perturbed were least correlated. The proposed approach is particularly suited for distributed computing since independent tasks of equal magnitude need to be performed. This would amount to flow simulations of the same realization at different nodes while perturbing different regions. Rather than doing the flow simulation on whole realizations at different nodes, certain low sensitive regions may be excluded by using boundary conditions. These low sensitivity boundary regions would not need frequent updating since the there would be minimal flow across those regions. *However when more severe changes are made to permeability field, the solver may not converge resulting in a much*

larger simulation time. For such cases full field simulations at different nodes while perturbing locally is advisable. In the distributed environment, any commercially available serial flow simulator can be utilized at each of the computational node and furthermore, the high efficiency of serial algorithms can be put to use. Two scripts for distributing simulations to various nodes have been attached in 'Chapter-3' in the appendix. The scripts are very simple and use the 'rsh' utility. The scripts are well documented and self-explanatory.

In the first script 'rshtemp', the macro '/opt/MPI/ecl/macros/@eclipse' is distributed to the different nodes of the cluster i.e. MPI_1 to MPI_8. The purpose of this script is just to illustrate the use of 'rsh' utility to distribute the applications.

The second script 's.pl' is a perl script. This script is much more generalized script. This script requires an input file, which has the name of the nodes to which the application needs to be distributed. The macro or the application that needs to be distributed is stored in the variable 'macro_name'. For example if the temp file has the name of the nodes of the cluster then 'perl s.pl temp' should be typed on the command prompt. And the macro name should be accordingly changed as per the requirement.

Preliminary verification of the availability of the required features in the selected simulator

Test Cases

Parallel Simulation

In order to demonstrate the parallel flow simulation capabilities of Parallel ECLIPSE 100, the following cases were run.

Case1 and Case 2

A 4000 ft. x 4000 ft. x 20 ft reservoir model was assumed (with constant permeability of 500 md, constant porosity of 30%). The reservoir is assumed to have one vertical producer in the middle of the reservoir. Four parallel processors were utilized for the parallel simulation. Parallel Eclipse 100 was used that allows the reservoir to be divided in either X or Y dimension. Three cases were run, with identical reservoir dimensions and well locations. The simulation was run for 500 days. All parallel simulations utilized four processors. In our case the definition **of aspect ratio is the number of grid blocks in x to y** directions. The whole point of doing the following cases was to demonstrate that certain schemes of domain decomposition could be computationally efficient.

Case 1: Aspect ratio changed while the number of grid blocks are kept same

The effect of the aspect ratio of the reservoir on the performance of the parallel simulation was assessed.

Table 3.2: Influence of aspect ratio on computation speed using ECLIPSE 100 (1-D domain decomposition).

Grids	Aspect Ratio	Time (Sec)	Producer Coordinates
4*4000*3	0.001	184	(2,2000)
16*1000*3	0.016	75	(8,500)
20*800*3	0.025	56	(10,400)
40*400*3	0.1	144	(20,200)
80*200*3	0.4	236	(40,100)
160*100*3	1.6	322	(80,50)
400*40*3	10	491	(200,20)

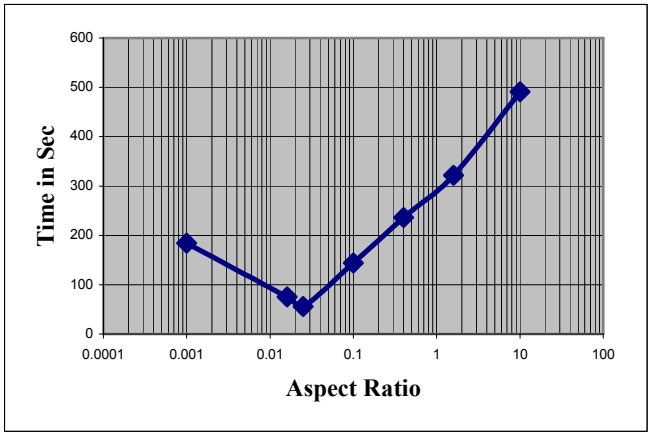


Fig 3.1: Influence of aspect ratio on computation speed using ECLIPSE 100 (1-D domain decomposition)

Case 2: Aspect ratio constant while the number of grid blocks were changed

The same reservoir scenario was simulated but this time with different grid resolution and keeping the aspect ratio constant. The grids employed are summarized below:

Table 3.3: Computation time against number of grid blocks.

Grids	No. of grids	Time (Sec)
4*4*3	48	4
40*40*3	4800	21
80*80*3	19200	76
160*160*3	76800	530

The results shown in Figure 3.2 indicate that maximum gain in computational speed is achieved when the grid size is increased by a factor of 100 from 100 grid blocks to 10000 blocks. Subsequently, the computation slows down when the grid size is increased. This could be attributed to the increased communication between processors due the larger interacting surface area among the domains.

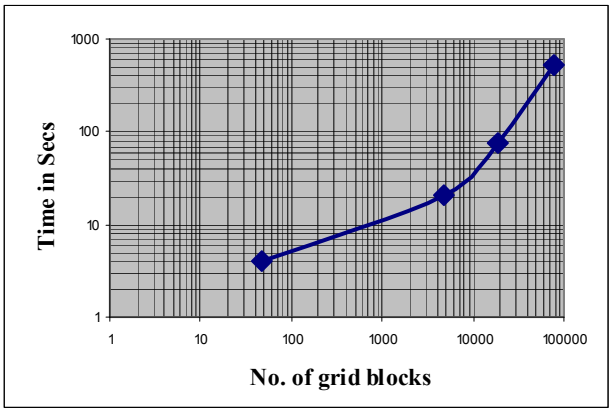


Fig 3.2: Plot of computation time against number of grid blocks.

Case 3: Number of grid blocks are kept same, the ratio of x to y gridblock dimensions has been changed and the physical dimensions of reservoirs are varied.

In this case physical dimensions of the reservoir are varied unlike the earlier two cases. The reservoir is assumed to have constant permeability of 500 md, constant porosity of 30%. The reservoir is assumed to have one vertical producer in the middle of the reservoir. Four parallel processors were utilized for the parallel simulation. Parallel Eclipse 100 was used that allows the reservoir to be divided in either X or Y dimension. Three cases were run, with identical reservoir dimensions and well locations. The simulation was run for 500 days. Parallel simulations utilized four processors. Physical dimension of grid blocks were kept same i.e. 500 ft, 500 ft and 50 ft in x, y and z directions respectively.

Table 3.4: Computation time versus ratio x to y grid block sizes

Aspect Ratio	Time in sec
0.001	598
0.025	213
0.1	100
10	82
1000	413

The results indicate that simulations performed with approximately the same number of blocks in the x and y directions and with square grid blocks are the most efficient.

Flux Boundary Conditions

The following are the results obtained while simulating a small section of the field using the boundary conditions obtained from the full-scale simulation. The full field has 50*50 gridblocks while the smaller sections have 25*25 grid blocks. Fluxes (flows of each phase from the full field run as a function of time) are used as boundary conditions. The full field simulation took 21 seconds while the smaller section took 9 seconds. Figure 18 shows the pressures and oil saturations at the end of 5000 days. Comparing the grid bloc pressures and saturations obtained for the sub domain simulations against the full field simulation values, the accuracy of the sub-domain simulation results was ± 1 psi. The saturations were alike up to three places of decimal.

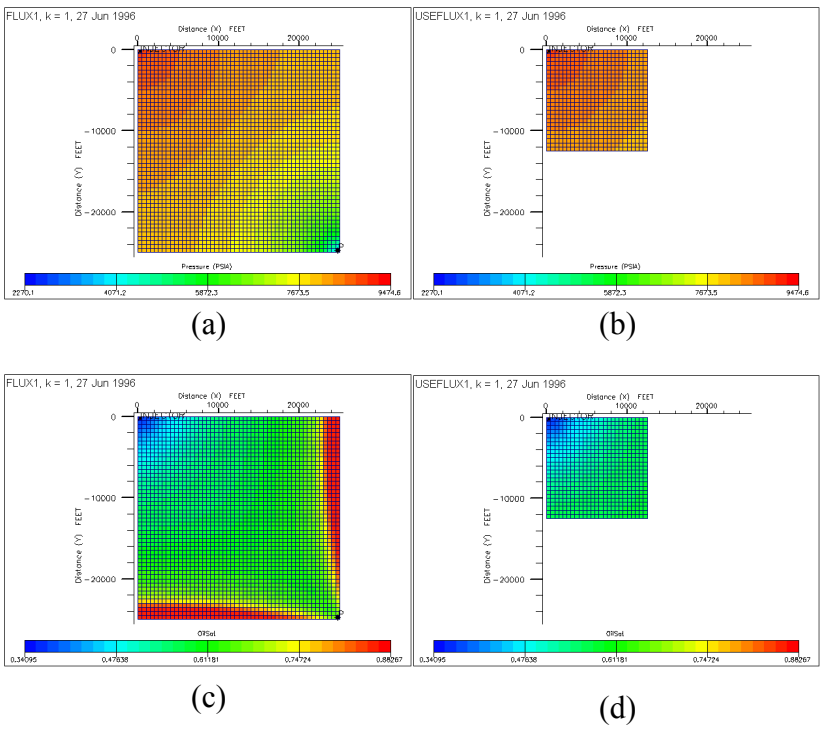


Fig 3.3 : Full field flow simulation results: a) Pressure variations in the reservoir at 500 days obtained by full field simulation; b) Pressure variations in a domain obtained corresponding to boundary conditions derived from the full field simulation; c) Oil saturation values obtained by full field simulation, and d) Oil saturation values

obtained for sub-domain simulation using boundary conditions derived from full field simulation.

The choice of simulator was dictated by the availability of the calculation options that facilitate identification of sensitive regions and provide Hessian matrix of the objective function. Various Parallel and Boundary flux cases were run for the preliminary checks on Eclipse. A scheme for using parallel and distributed computing, specifically from the project's point of view has been formulated.

4. SENSITIVITY ANALYSIS

For the purpose of defining the domains the obvious choice would be to identify the sensitive regions to the objective function. Jacquard et al. (1965) first investigated the use of sensitivity coefficients. Sensitivity coefficients are defined as derivatives of the target simulation output with respect to parameters being adjusted to get a history match. The sensitivities may be calculated for one of the following:

1) Global objective function
2) Well specific objective function
3) Field phase (oil, water or gas flow rates) objective function.
4) Well specific phase (flow rates) objective function.

Mathematically sensitivity coefficient may be defined as $\frac{\partial f}{\partial x}$ where, f is one of the objective function listed above and x is the history matching parameter at a grid block, which is permeability for the cases presented in this report. In case the sensitivity were calculated with respect to multiple parameters x_i (e.g. porosity, permeability, thickness etc.), then normalization of the parameters would have to be performed in order to render the resultant sensitivities comparable on the same scale.

The program module **Simopt** in the ECLIPSE® suite has been used for the sensitivity analysis done in the report. All the flow sensitivity coefficients mentioned in this thesis are with respect to permeabilities in X direction. All the figures show permeability in X direction and permeability in Y & Z directions are equal to Permeability in X direction, unless otherwise is stated. The results can be easily generalized to permeabilities in other directions.

In order to demonstrate the sensitive analysis procedure, a deterministic reservoir model representative of a fluvial depositional environment has been used. The model has 120,000 (100*120*10) grid blocks with high permeability channels. The channel distribution and orientation exhibit small changes from one layer to the next. The process for generating the synthetic data set including the calculation of pseudo-seismic responses has bee discussed in Mao, 2000. This particular reservoir model was used for many of the cases discussed later in this report and is referred to as the ***STAN5*** data set throughout the report.

Example 1

Computing the sensitivity coefficient corresponding to the permeability in each grid block is practically not feasible due to the associated computational cost. The sensitivity analysis is done at 180 uniformly distributed pilot points. The sensitivity coefficients are then spatially interpolated to all other grid blocks. Since the different layers of the reservoir exhibit strong resemblance to one another, the strategy adopted for the analysis was to utilize one of the layers as the reference layer and the other layers as the incremental realizations for calculating the required sensitivities. For this case, the fifth (Fig 4.2 (a)) and the third (Fig 4.2 (b)) layers of the reservoir were used for computing sensitivities while the history was generated using the fourth layer (Fig 4.1). The history is generated for 500 days. Adjacent layers are chosen deliberately so that the initial guess is near to the actual model. There are three producers and two injectors. The full STAN5 rock property data & surfaces data is attached as 'STAN5' in the appendix. The eclipse data files for layer 3, 4 & 5 are also attached in the appendix as 'STAN5-Layer3-Eclipse file', 'STAN5-Layer3-Eclipse file' & 'STAN5-Layer3-Eclipse file' respectively.

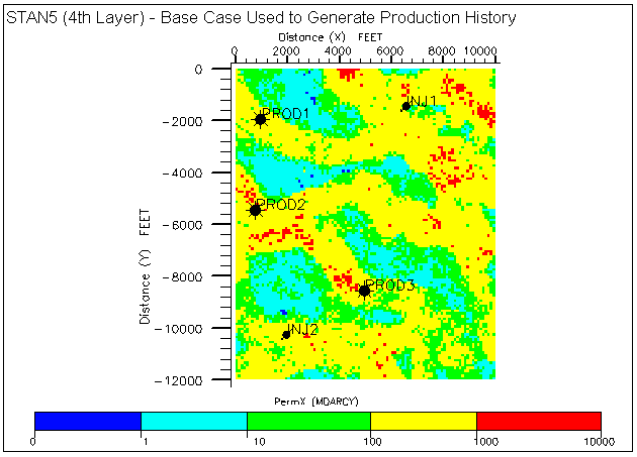
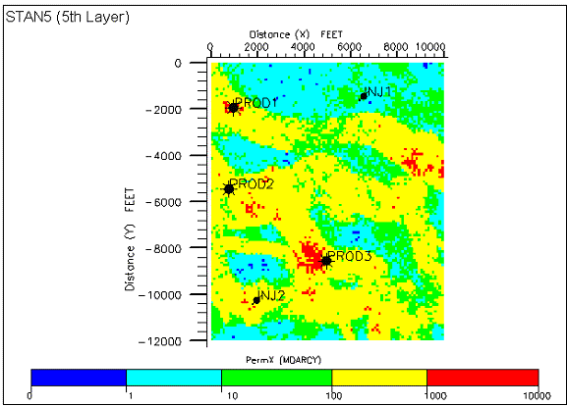
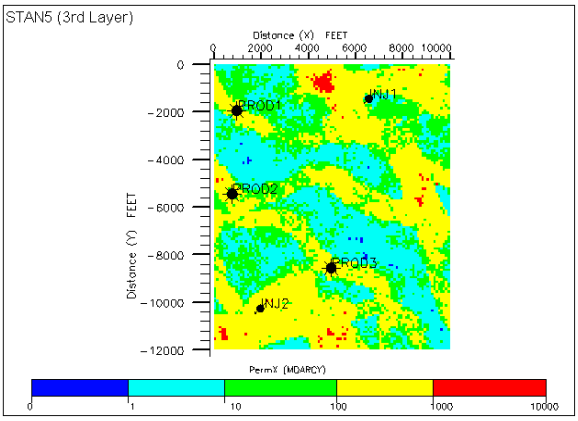


Fig 4.1: Reservoir model utilized to generate the base case production history



(a)

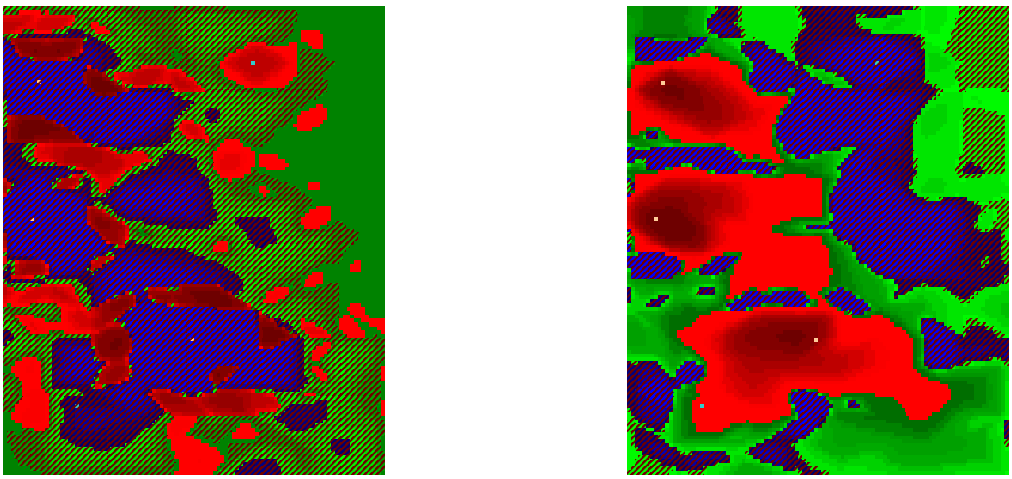


(b)

Fig 4.2: (a) Permeability realization for Layer 5 of STAN 5

(b) Permeability realization for Layer 3 of STAN5

The following Fig 4.3 shows the global (derivative of global objective function with respect to permeabilities) sensitivity coefficients. In all the cases discussed in this report the objective function is calculated using only flux terms with equal weight attached to them. The degree of change is proportional to absolute magnitude while the direction of change is governed by the sign of sensitivity coefficients. The highest values are represented in red, the middle ones in green and the lowest in blue. The negative values are shown with the shaded lines. Negative sensitivity implies that permeability values in those regions need to be increased in order to match the flow response for the base case.



(a) (b)

Fig 4.3: Global sensitivities for: a) 3rd layer, and b) 5th layer. The permeability corresponding to the 4th layer is used as the base model for calculating the sensitivities.

Sensitivity coefficient can also be calculated on the basis of individual well response. The following Fig 4.4 and 4.5 depict the sensitivity of different well responses to the permeability variations in layer 3, given the 4th (STAN5) layer as the base case.

The following figure qualitatively illustrates the concept of sensitivity coefficients. The blue shaded region implies that permeability needs to be increased in the region while red

implies that permeability needs to be decreased in the region. It is evident form fig 4.3 (c) & (d) that permeability needs to be increased in third layer near the produces wells in order to approach the base case. This is exactly what is being derived from the sensitivity coefficients in fig 4.3(e).

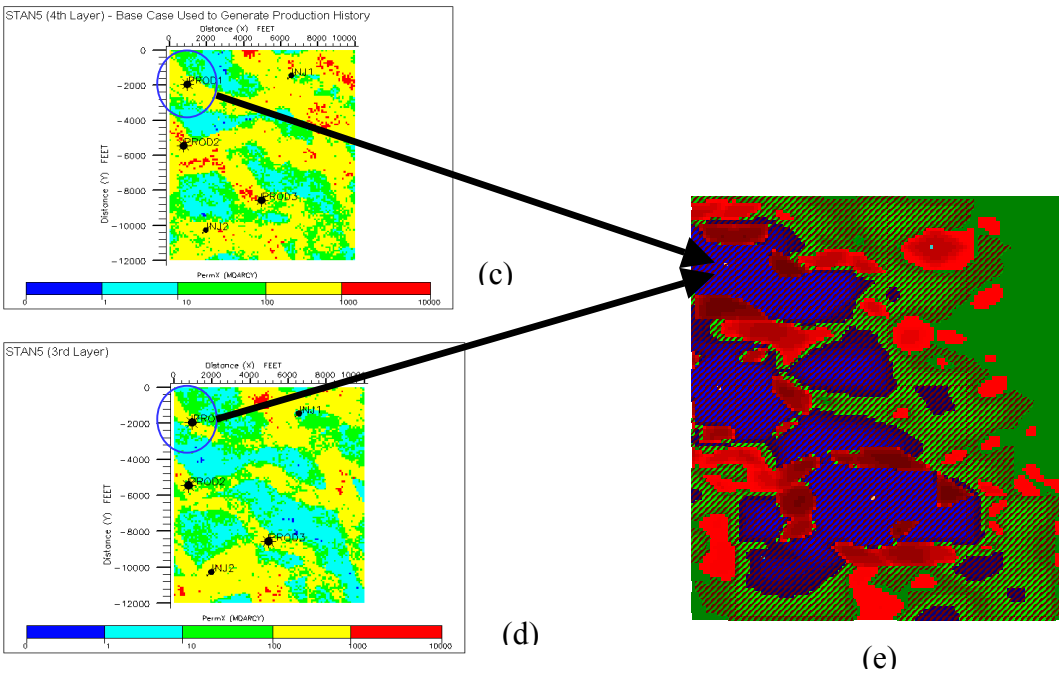


Fig 4.3(c): Permeability for base case (d) Permeability for 3rd layer (e) Global sensitivity coefficient on percentile map for layer 3 with respect to layer 4 of STAN5.

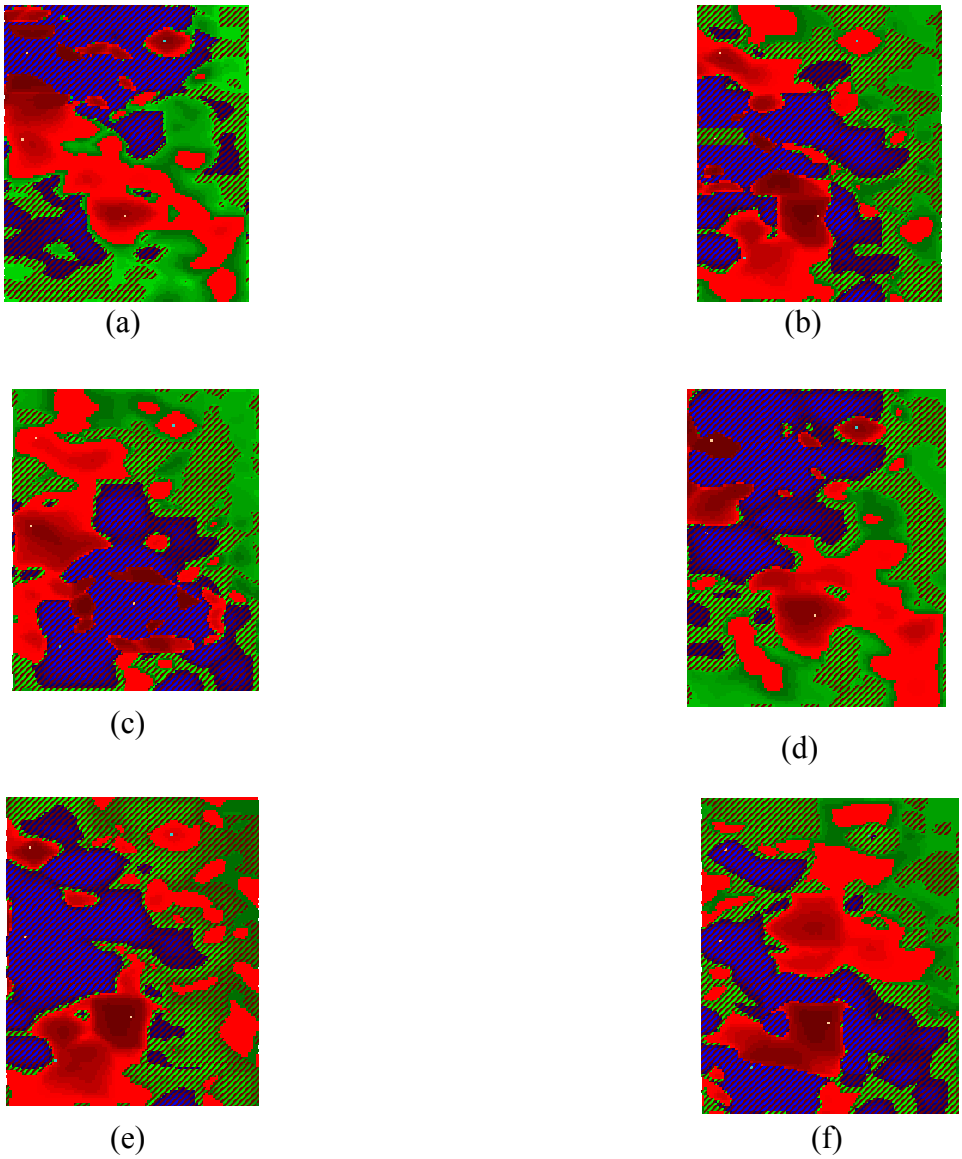
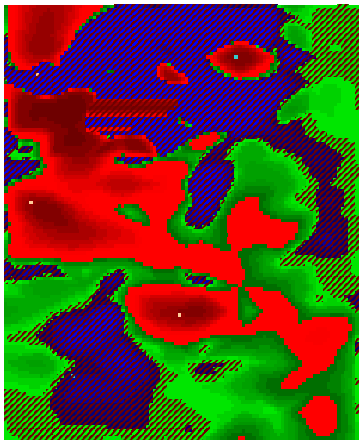
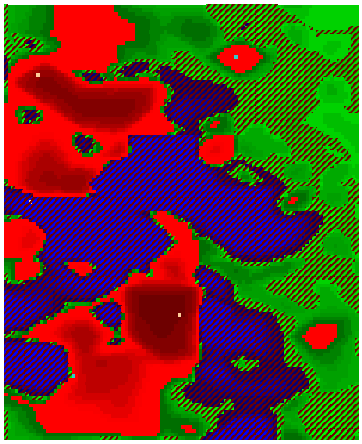


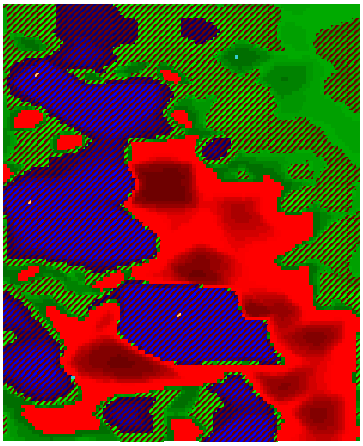
Fig 4.4: Sensitivity coefficients for the layer 3 permeability model calculated on the basis of: a) Well 1 oil production rate; b) Well 2 oil production rate; c) Well 3 oil production rate; d) Well 1 water production rate; e) Well 2 water production rate; and f) Well 3 water production rate.



(a)



(b)



(c)

Fig 4.5: Sensitivity coefficients for the layer 3 permeability model calculated on the basis of: a) Well 1 gas oil ratio, b) Well 2 gas oil ratio, c) Well 3 gas oil ratio.

The following observations are based on the results in Fig 4.4 and 4.5:

- Decreasing objective function defined on a per well basis may not be a wise approach for history matching. There are regions in the reservoir, that show positive sensitivities for one well and negative sensitivity for another well. This implies that decreasing the permeability in one region may decrease the objective function for one well but it may increase the objective function for another well.
- Even for a particular well it is possible to decrease the mismatch for particular phase production rate at the cost of the increasing the mismatch for another phase.

Based on the results and observations made above, it is proposed that a better approach to defining reservoir regions may be using the global objective function. The global objective function would be a weighted sum of individual phase production rates from all wells and would hence represent a compromise between competing flow response characteristics. Hence the better approach would be to target the global objective function while perturbing the permeability locally in a domain.

Example 2

In this case the permeability realization is very different from the base case unlike the earlier case where the initial guess closely resembled the base case (reference) model. Since an iterative, permeability updating procedure for history matching is proposed in this research, the objective is to gauge the frequency with which the reservoir zones designation would have to be updated during the iterative procedure.

The base model is still the **4th layer of STAN5** and all other inputs remain the same. Sensitivity coefficients are calculated for two models. The first model has perm-y and perm-z equal to the base case while the perm-x is very different from the base case and is shown below. The second model is a constant permeability field with permeability being 1000 md in all directions.

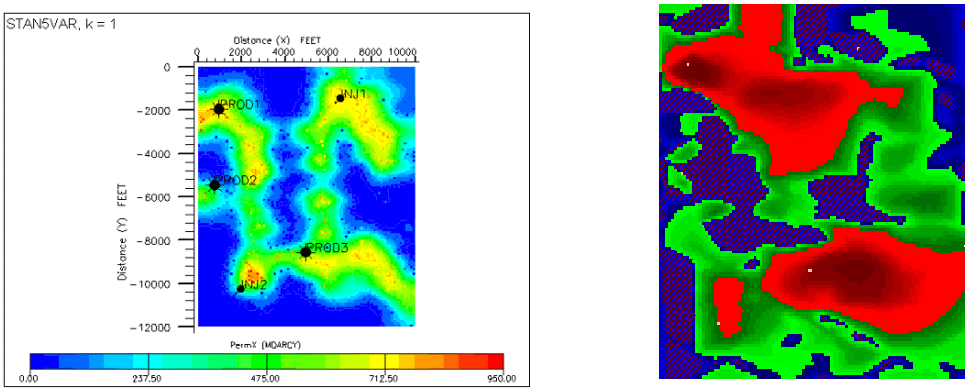


Fig 4.6: Permeability field with perm x very different from the base case. The corresponding global sensitivity coefficients computed on the basis of the base case model are shown in the right.

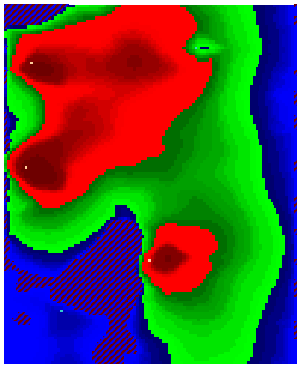


Fig 4.7: Global sensitivities corresponding to a homogeneous permeability field.

In both these cases, the results seem to indicate that the sensitivity coefficients are not so reliable if the initial guess is not near the actual model.

This important observation can be understood on the basis of the following conceptual plot.

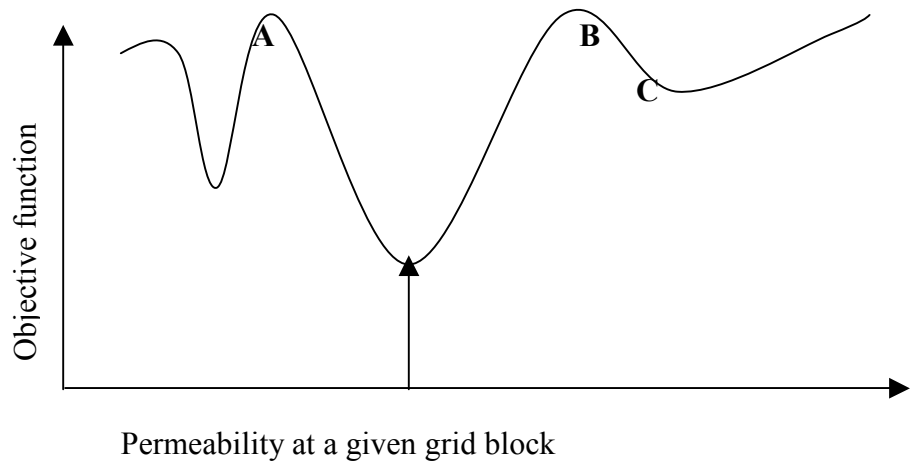


Fig 4.8: Typical objective function profile and the importance of a good initial guess.

The sensitivity analysis would give reliable information if the initial guess is between points A and B. The negative slope of the objective function from A to the global minima would suggest a negative sensitivity i.e. the permeability value in the grid block would have to be increased. If the permeability value becomes too high, the positive slope of the objective function profile from the global minimum point to B would suggest a positive sensitivity coefficient value and hence a decrease in permeability value to result in a history match. However, if the initial guess had been C, the negative slope of the objective function profile at C would suggest a negative sensitivity value and hence an increase in permeability and hence a further drift away from the true minimum value.

The application of sensitivity analysis for domain identification and some of the issues that have to be considered prior to defining reservoir zones have been discussed in this section. Prior to refining the procedure for zone identification, a brief review of the underlying theory is presented in the next section and subsequently techniques to improve the robustness of the zone identification procedure are discussed.

5. DESCRIPTION OF METHOD

The initial permeability realization is generated using p-field simulation (Srivastava, R.M., 1992, Reservoir characterization with probability field simulation, SPE 24753, 67th SPE Annual Technical Conference and Exhibition, 927-938). The local conditional distribution at each location is derived using indicator kriging (Deutsch, C.V., Journel, A.G.(1998): GSLIB: Geostatistical Software Library and Users Guide, Oxford University Press Inc., Oxford, New York 1992). Indicator kriging utilizes the available data and a model for spatial correlation (indicator variogram). A realization of the permeability field is obtained by sampling from the respective conditional distributions using spatially correlated random draws. The resultant permeability realization reflects the desired geological correlation and honors the available data. The history is created from a reference realization. All other parameters except the permeabilities are kept same while generating history from the reference model. The resulting realization is then subjected to sensitivity analysis.

Sensitivity coefficients are calculated using permeability values at certain pre-selected pilot-points (or the grid blocks). Ideally the points should be located at the points of maximum uncertainty from a flow perspective and also the number points should be kept as low as possible without compromising much on the accuracy. In current application the pilot points are distributed uniformly through out the reservoir. The sensitivity analysis is only done with respect to permeabilities since only permeabilities are perturbed for the present case in order to history match the synthetic field. During the sensitivity analysis both sensitivity coefficients and the Hessian of the objective function are evaluated at pilot points.

Principal Component analysis is then performed on the Hessian Matrix of the objective function. The gradient zone concept first proposed by Robert Bissel (Bissell, R. C. “Calculating Optimal Parameters for History Matching”, ECMOR IV, 4th European Conference on the Mathematics of Oil Recovery, Roros, Norway, 1994) has been

suitably modified to meet the requirements of the proposed approach. The basic premise of PC analysis is to isolate the most sensitive and least correlated regions. In the original application of PCA, the most sensitive regions were needed to make the history matching efficient while the least correlated property is needed so that regression works efficiently. Regression works best when the independent variables are least correlated with each other in terms of their effect on the dependent variables. In the present case the independent variables are the permeabilities and the dependent variable is objective function. Full independence of domains within a reservoir is not possible unless there are regions separated by zero permeability streaks or zero transmissibility faults. In the proposed probabilistic history matching approach, least correlated regions are needed so as to make the problem suitable for distributed computing and to retain the simplicity of 1D optimization.

The proportion of the variance extracted by the eigenvectors has been used as the deciding criteria to determine the number of eigenvectors (or the domains) to be extracted That proportion can be obtained by dividing the corresponding eigenvalue with the trace of the Hessian matrix and a suitable threshold may be applied on the amount of total variance (or the total information) extracted.

The following procedure is followed for obtaining the domains based on principal component analysis of the Hessian matrix

1) Obtain the eigen values and the corresponding eigenvectors of the Hessian matrix. Eigenvectors thus evaluated would be in n dimensional space, where n is the number of the pilot points (or the rank of the Hessian matrix). Hence every eigenvector would have a component associated to all pilot points
2) Interpolate the component of the eigenvectors throughout the reservoir from the component values at the pilot points.
3) Rank the eigen values based on their magnitude.

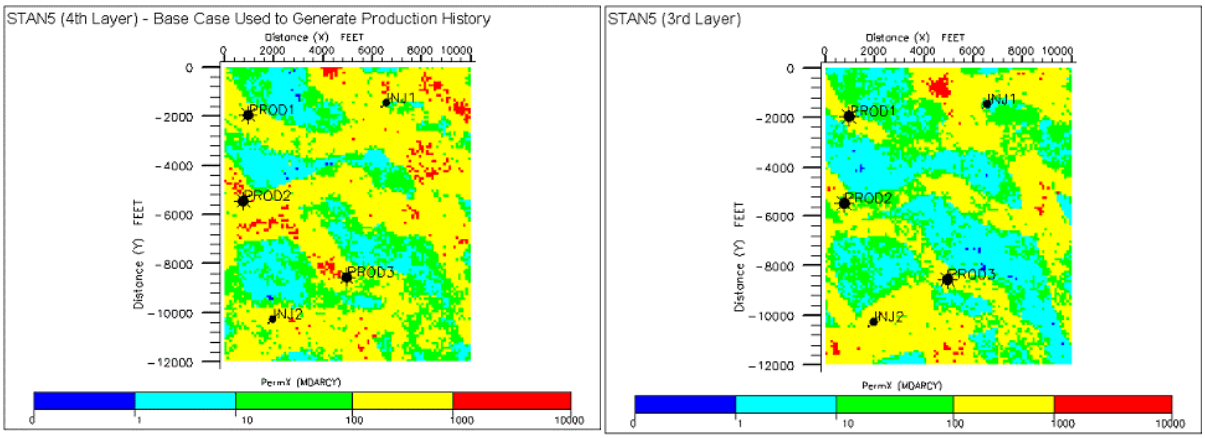
4) Apply a threshold on the size of domain covered by first eigenvector. It could be say 40% of the total grid blocks. In other words all the grid blocks at which the *absolute value* of the eigen component corresponding to the first eigenvector exceeds the P60 value (of the eigen components) are grouped as the first domain.
5) Follow step 4 for the subsequent eigen vectors with respective thresholds. During this step the grid block cells covered by the earlier (higher in rank) are excluded.

The commercial simulator used for this research already has the capabilities to evaluate the Hessian and the eigen vectors.

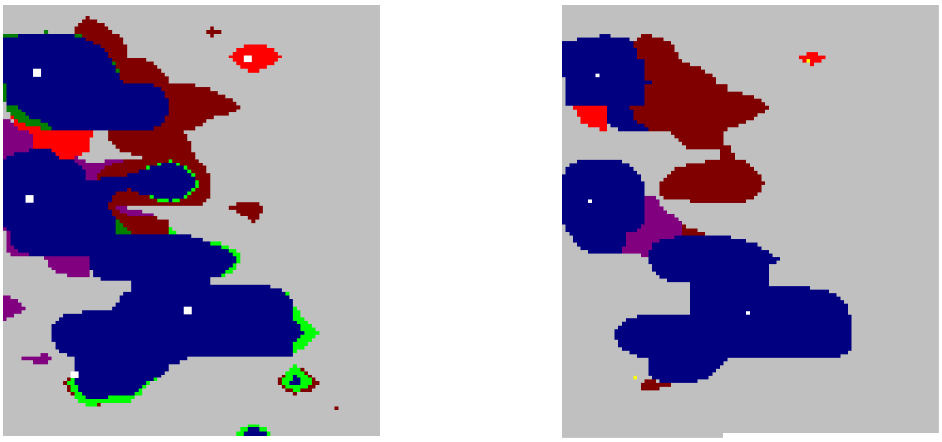
In the above scheme the magnitude of the eigenvalue is used to rank and retain a reduced set of eigenvectors. In the subsequent step, the eigenvectors are scaled such that the magnitude of a vector is equal to the corresponding eigenvalue. The merits for this sort of scaling can be summarized as follows:

1) There is no need for different thresholds for each eigenvector in order to define regions that span a certain volume of the reservoir. Instead one common threshold may be applied for all eigenvectors since the eigen components are directly comparable once they are scaled.
2) It might be possible that a component of lower eigen value exceeds the eigen component of the larger eigen value at a particular grid block. In that case, if proper scaling is not done, the grid block may end up being erroneously assigned to the larger eigenvector domain.
3) In practical implementations, it has been observed that if the eigenvectors are used without scaling then the regions covered by the lower rank eigenvectors form rings around the region covered by higher eigenvector (Figures 5.2(c) & (d)). This is institutive from sensitivity criteria perspective. But this is not physically possible since a region corresponding to a lower eigen vector can not be least correlated and yet form an annular region around the region corresponding to a higher eigen

vector. The code for domain decomposition subsequent to scaling has been attached as 'Code 5.1' in the appendix.



(a) (b)

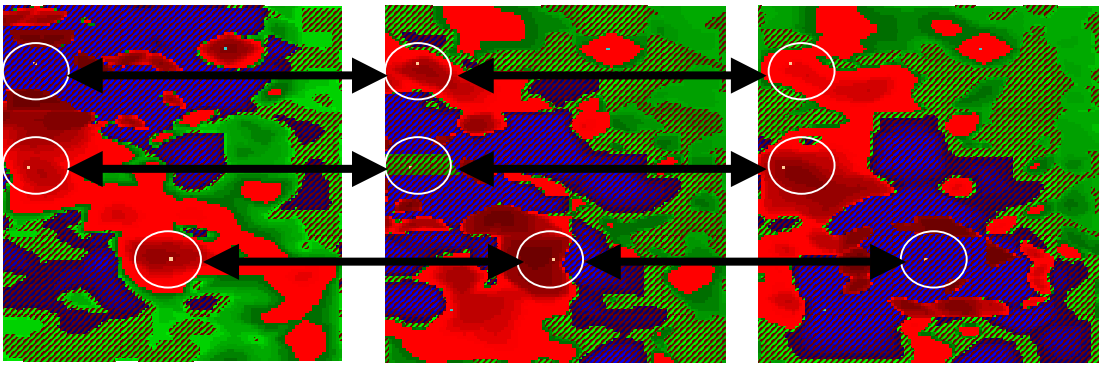


(c) (d)

Fig 5.2: (a) Actual permeability field used to generate the history; (b) Initial guess to be history matched; (c) show the domains obtained without scaling of eigen vectors with thresholds of 30% for the five eigen vectors; (d) Domains with scaled eigen vectors with thresholds 30%.

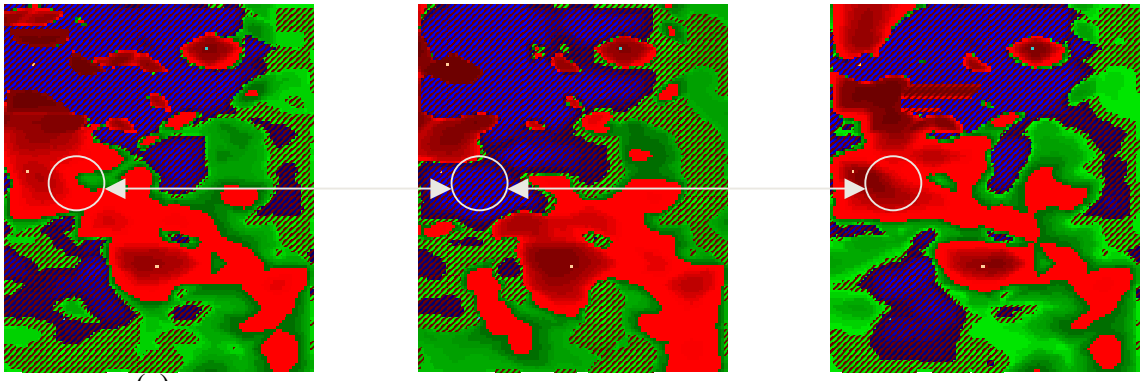
Each domain is then perturbed using the probabilistic approach at different nodes of the cluster while targeting the *global objective function.* The global objective function is a weighted sum of individual phase production rates from all wells in the reservoir.

This point has been mentioned in chapter 4.The following discussion explains it in more detail, that it is important to consider the global objective function since otherwise the objective function for one well may be reduced at the expense of increasing the objective function for another well. Even for a particular well it is possible to decrease the mismatch for one particular phase production rate at the expense of increasing the mismatch for other phase. This point has been illustrated in Fig 5.3 & 5.4, which show that same region could be in either positive sensitivity region (implying that the permeability has to be increased to minimize mismatch) or in the negative sensitivity zone depending upon the well or the phase, for which sensitivity coefficients are being calculated.



(a) (b) (c)

Fig 5.3: Showing the sensitivity coefficients on the percentile map for the case '5.2(a)' as reference for creating history and '5.2 (b)' as the initial guess. Highest values in red, intermediate in green and the lowest in blue, negative sensitivity coefficients regions are shown with shaded lines; (a) With respect to well-1 oil production rate (b) With respect to well-2 oil production rate (c) With respect to well-3 oil production rate.



(a) (b) (c)

Fig 5.4: Showing the sensitivity coefficients on the percentile map for the realization shown in Fig. '5.2(a)' as reference for creating history and Fig. '5.2(b)' as the initial guess. Highest values in red, intermediate in green and the lowest in blue, negative sensitivity coefficients regions are shown with shaded lines; (a) With respect to well-1 oil production rate b) With respect to well-1 water production rate (c) With respect to well-3 Gas oil ratio.

Once the reservoir zones have been delineated, the perturbation of permeability values within individual zones is controlled by the updating parameter r_D for that zone. Starting from an initial permeability model, the permeability value at each location is coded into an indicator category by selecting suitable permeability thresholds. The indicator category at each location is then perturbed within a L-step Markov chain. The probability of transitioning from indicator category *k* at step ℓ to the category *k'* at step $\ell+1$ is written as:

$$
\begin{aligned}
&P\left\{I^{\ell+1}(\mathbf{u})=k' \mid I^{\ell}(\mathbf{u})=\mathrm{k},\, C\right\}=r_{\mathrm{D}}\cdot P\{I(\mathbf{u})=k'\} \quad \forall k'\neq k \\
&P\left\{I^{\ell+1}(\mathbf{u})=k \mid I^{\ell}(\mathbf{u})=\mathrm{k},\, C\right\}=1-\sum_{k'\neq k} r_{\mathrm{D}}\cdot P\{I(\mathbf{u})=k'\}
\end{aligned}
\tag{5.9}
$$

The probability $P\{I(\mathbf{u})=k'\}$ is the same as the prior probability P(A) and represents the marginal probability corresponding to class *k'* . This prior probability can be calculated on the basis of available data. The parameter r_D is an optimization parameter that is selected such that the mismatch between the predicted dynamic response and the target dynamic response is minimized. Upon convergence, the probability on the left hand side of Eq. (5.9) is the conditional probability $P(A\,|\,C)$ that quantifies the information contained in dynamic data pertaining to the permeability variations in the reservoir. At any step of the iterative calibration procedure, a higher value of r_D implies a higher probability for the current category *k* to change to a new category *k'* from step ℓ to step $\ell+1$. As a result the realization sampled at the step $\ell+1$ may be significantly different from that at step ℓ. Conversely, a low value of r_D signifies a higher probability to remain in the same category *k* over the step. This gives rise to a realization that is quite similar to that at the previous step.

Starting with an initial guess for r_D^*, the probability on the left-hand side of Eq. (5.9) corresponding to different indicator categories $k=1,\ldots,K+1$ is calculated. As noted from Eq. (5.9), a starting image $I^o(\mathbf{u})$ is required in order to compute the requisite

probabilities. A starting image conditioned to available hard data and depicting the target spatial correlation structure is used. The probability distribution obtained using Eq. (5.9) has to be combined with the conditional probability $P(A|B)$ that quantifies the prior geological knowledge and is obtained from indicator kriging. The merging of probabilities $P(A|B)$ and $P(A|C)$ is done using the permanence of ratio hypothesis. Under this hypothesis, the relative updating of the permeability event A due to the observed dynamic data C remains unaffected by the occurrence of B (the prior geological information). The joint conditional probability $P^*(A|B,C)$ can then be written as:

$$P^{\ell}(A|B,C) = \frac{a}{a+bc} \tag{5.10}$$

.

The superscript in the probability expression indicates that the joint conditional distribution is not yet the converged distribution and corresponds to the step ℓ of the Markov chain. The quantities:

$a = \dfrac{1-P(A)}{P(A)}$, $b = \dfrac{1-P(A|B)}{P(A|B)}$ and $c = \dfrac{1-P(A|C)}{P(A|C)}$ can be interpreted as the relative distance to event *A* occurring given that events *B* or *C* occur. For example, if occurrence of *B* guarantees the occurrence of event *A*, then $b=0$, i.e. the relative distance to *A* is zero. If on the other hand, occurrence of *B* guarantees non-occurrence of *A*, then the relative distance $b \to \infty$.

The updated probability distributions at all locations **u** within the domain are derived using Equation (5.10). A spatially correlated field of random numbers is used to sample permeability values from the updated conditional distributions as in p-field simulation. The updated permeability field at the iteration step ℓ is thus obtained.

The model is then flow simulated and the deviation of the simulated production response from the target response is computed in a global sense:

$$\Delta O^i(r_D) = \left\| f_{target}(t) - f^i(r_D{}^*, t) \right\|^2 \tag{5.11}$$

The objective function is minimized by iteratively adjusting the deformation parameter r_D. A 1-D, non-gradient based optimization procedure such as the Dekker-Brent method (Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P.: "Numerical Recipes in C: The art of scientific computing", Cambridge University Press, 1992) is utilized to obtain the optimal value of the parameter r_D. The model $I^1(\mathbf{u})$ obtained corresponding to the optimal r_D value is still a locally optimal realization, since it represents the optimal corresponding to the starting realization $I^O(\mathbf{u})$. Several outer iterations $\ell = 1,...,L$ are performed until a global match to the production history data is obtained. Thus the procedure described above consists of two iterative loops – an inner loop that corresponds to Dekker-Brent iterations for establishing the optimal r_D and an outer loop that corresponds to updating the reservoir realizations $I^\ell(\mathbf{u})$ for computing the LHS probability in Eq. (5.9). The two loops are repeated for all the domains.

Finally the perturbed regions are put together since the domains perturbed were least correlated and the geological consistency is always maintained while perturbing locally. The proposed approach is particularly suited for distributed computing since independent tasks of equal magnitude need to be performed. This would amount to performing flow simulations of the full reservoir model at different nodes while perturbing different regions. At the end of r_D optimization for individual domains, the merged permeability field is then subjected to full field perturbation. This step is needed because certain regions in the reservoir may never get perturbed if they don't fall within the delineated zones. Secondly, the full field perturbation would smooth out any artifact in the merged permeability due the fact that domains are not perfectly independent. The full methodology has been described through the flow chart in Fig. 5.5 & 5.6.

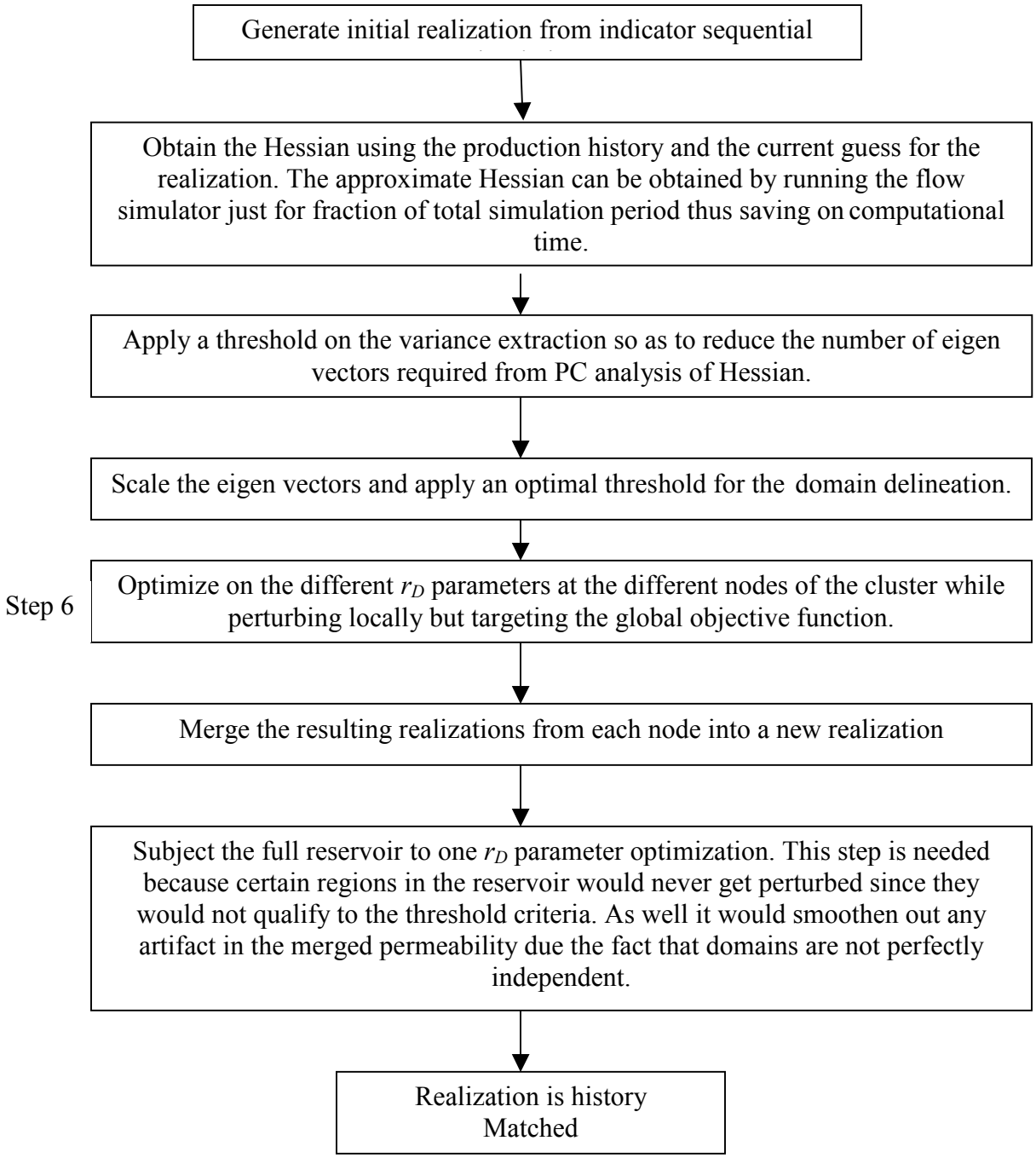


Fig 5.5: Flow chart showing the work flow; step 6 has been described in detail Fig 5.6

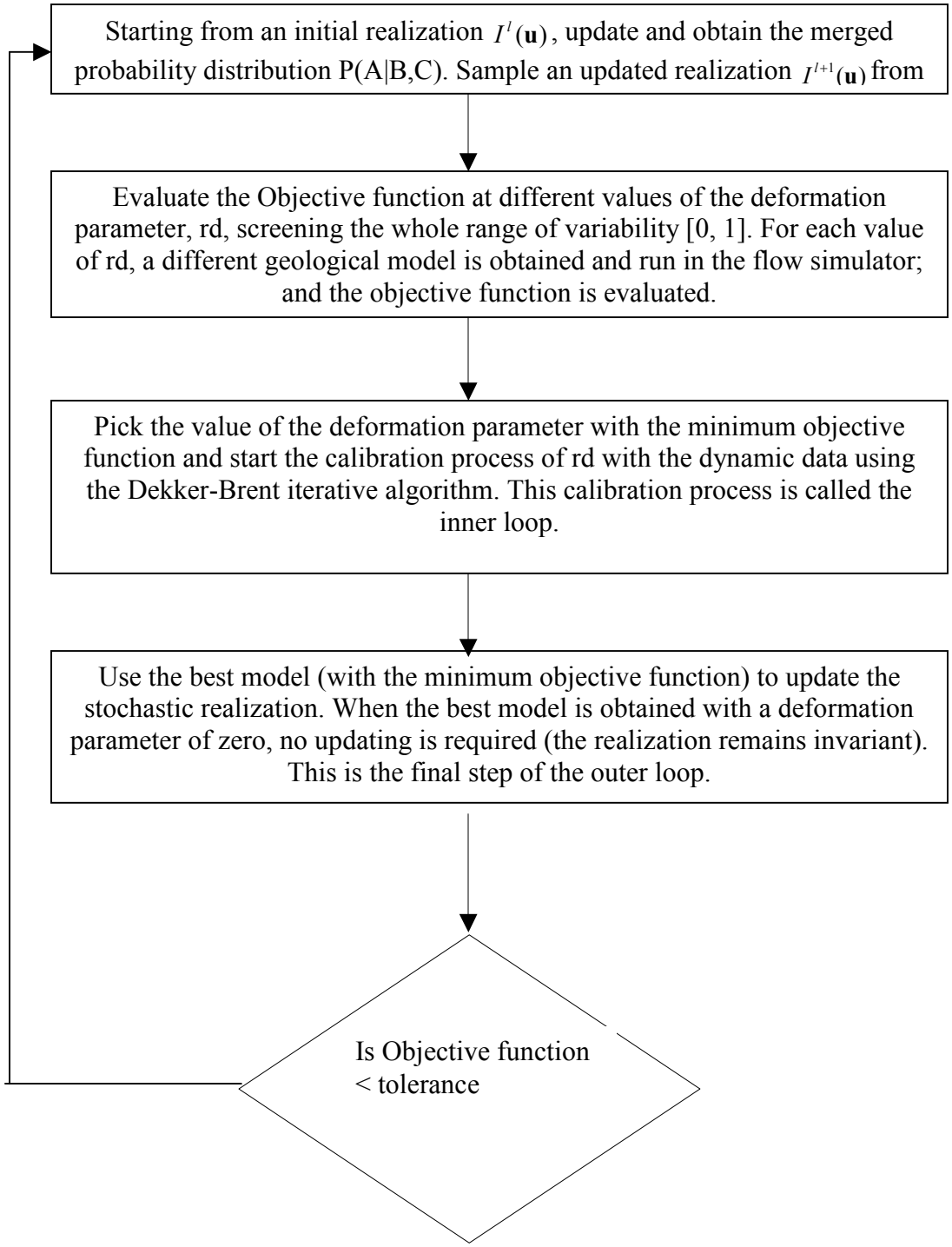


Fig 5.6: Flow chart showing the work-flow for r_d optimization

6. TEST CASES

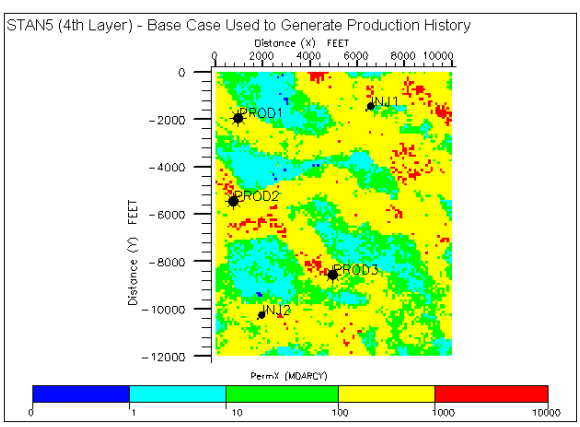
This section dwells on the following topics

1. Alternate ways of domain decomposition
2. Computational cost associated with sensitivity analysis
3. Robustness of domain delineation
4. History matched synthetic cases

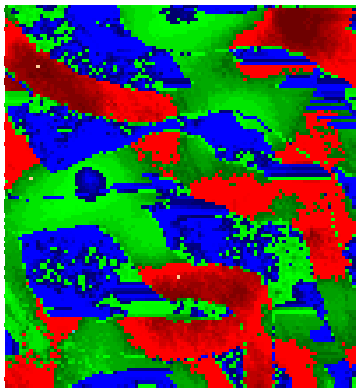
1. Alternate ways of Domain Decomposition

a) Domain decomposition based on porosity and thickness

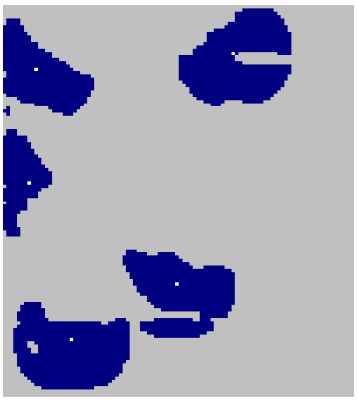
Domain delineation based on principal component analysis of the Hessian matrix is computationally expensive. In most situations, porosity and reservoir thickness can be reliably ascertained using auxiliary data such as seismic. In order to test this conjecture, an attempt was made to delineate the domains based just on the porosity and thickness at the grid blocks. Geobodies were defined on the basis of the product of porosity and thickness (Fig. 6.1(b)). Geobodies represent connected volumes of the reservoir that have storativity ($\phi \cdot h$) values greater than a threshold. The definition of geobodies was later modified in order to take into account the connectivity of storativity to a well location. In this revised scheme (Fig 6.1(c)), all regions within a search radius around wells where the product of porosity and thickness exceeds a threshold are grouped in a domain. The search radius is made proportional to the upper flow rate constrains on the wells.



(a)



(b)

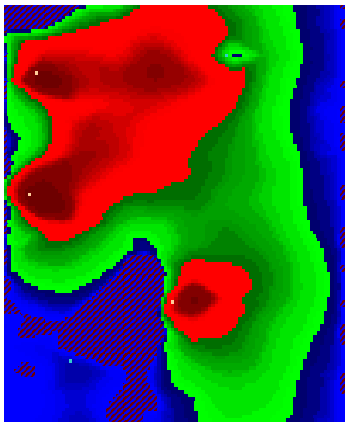


(c)

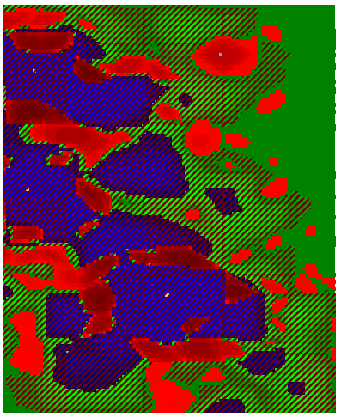
Fig 6.1: a) Permeability for base case model (Layer 4 of Stan5) used to generate the production history; b) Geobodies computed using the porosity and layer thickness information for layer 4; c) Geobodies computed using a modified definition that focuses on a search radius of 16 units around well locations.

For the same set of geobodies but different permeability realizations sensitivity analysis was performed. The base case was always kept the fourth layer of Stan5. Different sensitivity maps are generated for different permeability realizations. The results show that sensitivity coefficients are a strong function of the permeability realization under consideration. In a way this can be predicted since the permeability sensitivity coefficients are more influenced by the permeability realization under consideration rather than the product of porosity and thickness.

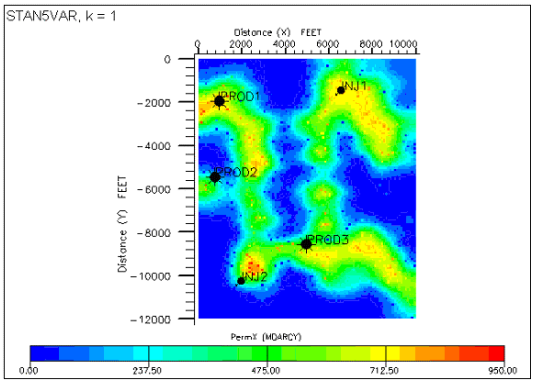
Fig 6.2 shows the sensitivity maps on percentile scale for different permeability realizations but same geobodies. It can be easily appreciated that they look quite different despite sharing same set of porosities and thickness values.



(a)



(b)



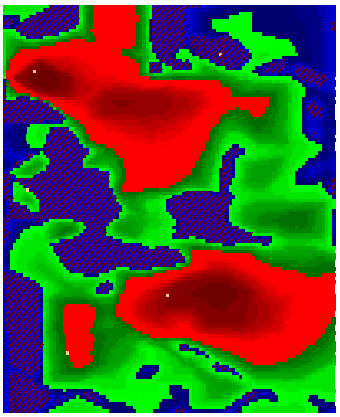


Fig 6.2: a) Sensitivity coefficients for constant 1000 md (in X, Y & Z directions) permeability realization b) Sensitivity coefficients for 3rd Layer of Stan5 c) Permeability realization used for the sensitivity analysis for the sensitivity coefficients shown in fig. 6.2d d) Corresponding sensitivity coefficients for 6.2c

A second approach could be that using permeabilities for domain definition instead of geobodies. This approach suffers from the fact that domains defined on the basis of permeabilities would lack the information of the reference. It is to be noted that the reference permeability field is indirectly represented by production history in case of sensitivity coefficients.

Geology is very important for history matching. It can be used to guide the degree perturbation to the existing guess in order to approach the reference. This would be a very wise way to get to physically realistic initial model. But in the present case the stochastic methodology dictates the perturbation, which does, has P(A/B) component at the end of each outer loop forcing the geological consistency.

b) Domain decomposition based on permeability covariance matrix

Principal component analysis can also be applied to the permeability covariance matrix obtaining the domains. Once the eigenvectors are obtained appropriate thresholds in the manner described earlier may be applied to get the domains. It is expected that the domain delineation procedure using the covariance matrix would identify regions that exhibit similar permeability structure (geological facies). A potential drawback could be that the analysis is just based on the static data. It is possible that for particular boundary (well) conditions, there might be significant flow across facies boundaries, resulting in connected flow-based sensitivity regions but un-connected permeability-based regions.

A synthetic 100*120*1 field is used to illustrate the point outlined above. The covariance matrix is generated in such a way there are three facies (100*40*1, 100*40*1, 100*40*1) in the reservoir. This is deliberately done to evaluate the efficacy of eigenvectors in identifying the facies. The exponential model is used such that variances within three facies are 800, 600 and 1000 respectively. The correlation length is 170 units within the facies. The correlation length is 20 units across the facies with variance being 100 units. A cut off 33% was applied to first three eigenvectors. The eigenvectors could resolve the three facies perfectly (Fig 6.3).

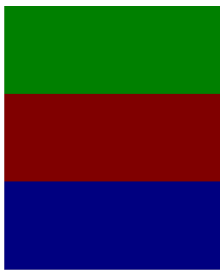
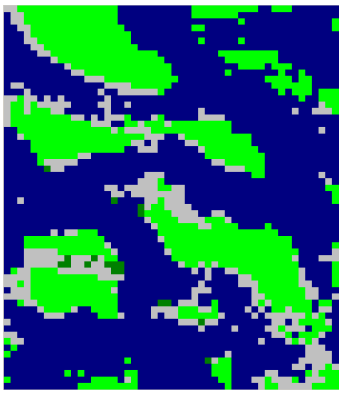


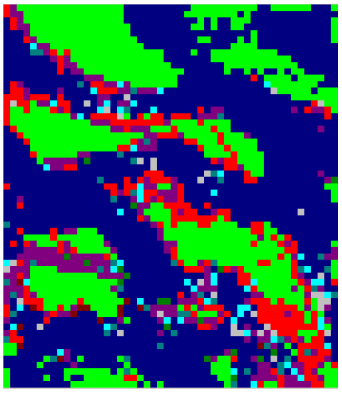
Fig 6.3: Eigen domains based on covariance matrix of permeabilities. The following is the color scheme employed: First eigenvector positive components ■, second eigenvector positive components ■ and third eigenvector positive components ■.

The facie with highest variance is being identified as most sensitive, the facie with the second highest variance as the next most sensitive and so on. The covariance matrix is attached as 'Covariance_6.2' the appendix. The code for generating this matrix is attached as 'Code_6.3' in the appendix.

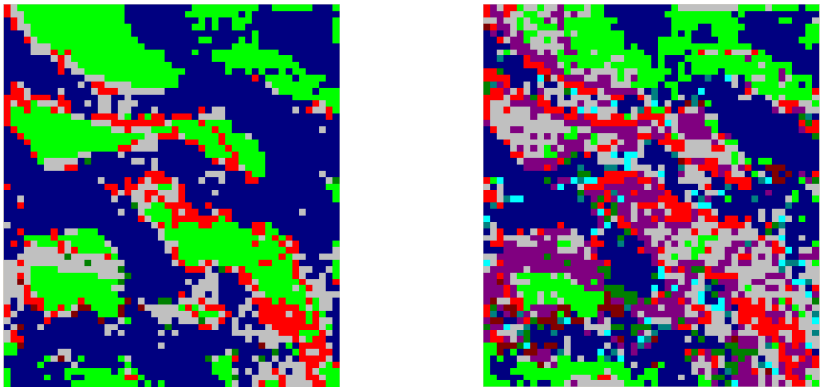
A similar exercise was carried using 2nd, 3rd, 4th, 5th and 6th layer of STAN5. The covariance matrix was calculated from the above-mentioned five layers. In other words at a particular grid block the permeabilities from the five layers are taken as multiple observations at that particular grid block. Stationarity is not assumed while calculating the covariance matrix since we have multiple observations for each grid block. Covariance matrix from a single permeability realization may also be used but then stationarity needs to be assumed. The problem may be visualized as evaluating interdependencies from six plausible geological models of the same 2D field. It must be noted that a covariance value calculated over a sample set of five is likely non-robust, nevertheless the calculation was performed to demonstrate the methodology.



(a)



(b)



(c) (d)

Fig 6.4: Reservoir regions for the STAN5 data obtained using the permeability covariance matrix: a) Applying a 60% cut-off for the first two eigenvectors of, b) Applying a 50% cut off on the first 3 eigenvectors, c) Applying a 50% cut-off on the first 5 eigenvectors, d) Applying a 30% cut off on the first 5 eigenvectors.

As seen in Figure 6.4, the delineated zones retain the imprint of the channel feature common to all the reservoir layers. Increase in the number of eigenvectors retained to represent the variability and/or reduction in the cut off threshold, adds more noise to the resultant maps.

In order to evaluate whether the lack of adequate statistical mass influences the calculation of the covariance matrix and hence the determination of eigen domains, the same exercise was performed on 50 realizations of a permeability field generated by the p-field simulation technique. The reservoir size is 50 x 50 grid blocks. The results are shown in Fig. 6.5.

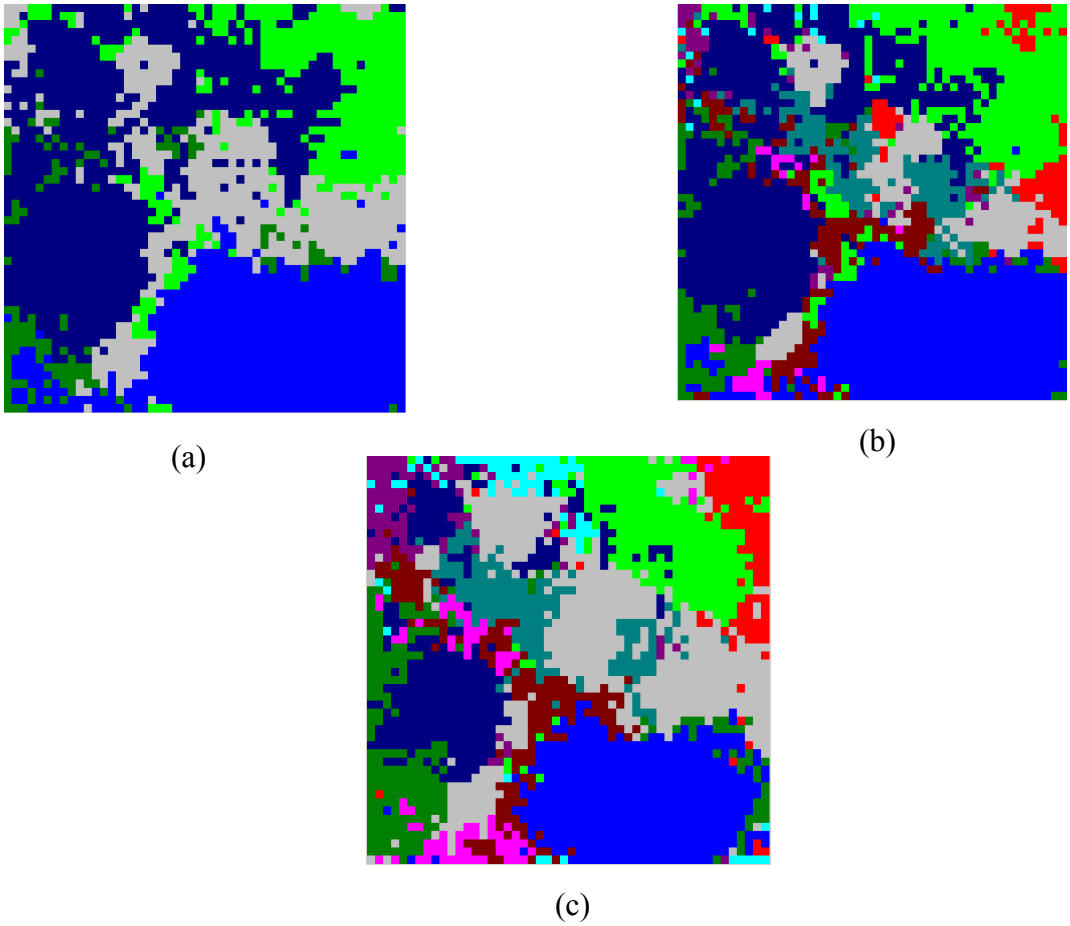


Fig 6.5: Reservoir regions delineated corresponding to a suite of permeability models obtained using p-field simulation. The eigen decomposition of the permeability covariance was performed and the reservoir regions were obtained by: a) Applying a 60% cut-off for the first two eigenvectors, b) A 50% cut off applied to the first 5 eigenvectors, and c) A 30% cut off applied to the first 5 eigenvectors.

The following is a comparison table for the two cases discussed above.

Table 6.1: Profile of the eigenvalues for the two cases discussed

Eigenvalues of the covariance matrix generated from the 50 Realizations generated by p-field simulations.	10409609	8356552	5564055	5162116	4260301	3972610
Ratio of the Eigenvalues to the fourth largest eigenvalue.	2.016539	1.618823	1.077863	1	0.825301	0.76957
Eigenvalues of the covariance matrix generated from the five layers of STAN5.	2.5E+08	6.9E+07	3.7E+07	10000	0	0
Ratio of the Eigenvalues to the fourth largest eigenvalue	24820	6853	3731	1	-	-

It can be seen from the table above that eigenvalues are almost of the same magnitude for the 50 Realizations generated by p-field simulation unlike STAN5 layer case. This implies that redundancy level among the 5 layers is much higher as compared to 50 realizations. This is due the fact that permeability realizations from the five layers are quiet similar to each other.

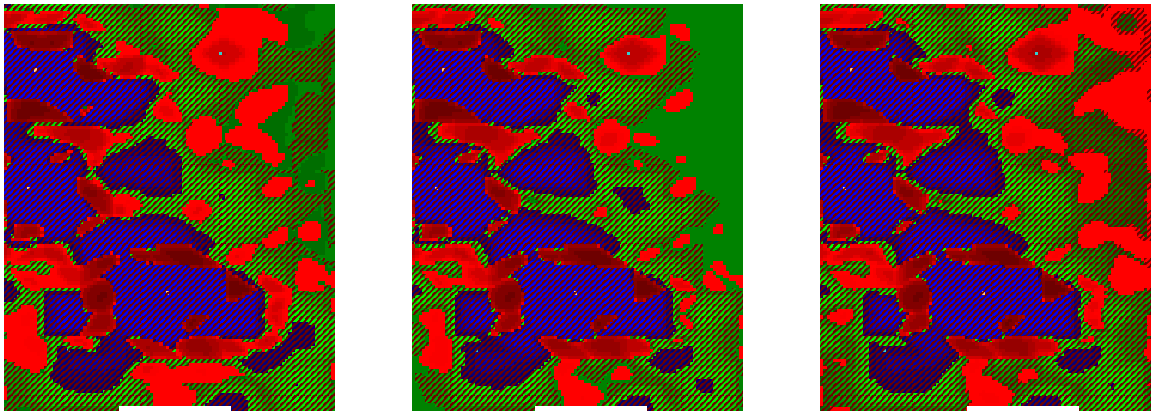
It should be noted that static sensitivity analysis based on many permeability realizations while sensitivity coefficients from flow are calculated for a given realization.

There are two aspects considering during history matching – Uncertainty (Geostatistical) and Flow Sensitivities. The domains defined based on the Principal component analysis of permeability covariance matrix identifies the most uncertain region to be most sensitive. The ***most uncertain parameters may not be very sensitive to the objective function***. For example the permeabitities at particular reservoir regions may be most uncertain but due to lesser flow may not be consequential to the objective function. The regions near the wells may be most certain but still may be most sensitive. Any difference in the actual and estimated model permeability near the well would be exaggerated in the low sensitivity analysis. ***Targeting most uncertain regions would be more accurate but would not be as effective.***

2. Computational cost associated with sensitivity analysis

As earlier mentioned calculation of one sensitivity coefficient cost as much as 20% of the total run time. Some approximations are necessary in order to keep the computational cost reasonable. The feasibility and robustness of reservoir regions defined by restricting the duration of the flow simulation runs was evaluated. The results in Fig 6.6 indicate that while the sensitivity coefficients do change in the magnitude with respect to time, their relative magnitudes remain the same provided the permeability, well constraints, number of wells remains the same. The sensitivity coefficients defined on percentile scale remain almost the same. This implies that reservoir regions defined by performing the flow simulation for a fraction of total simulation period would serve the purpose of defining reservoir regions.

It has to be emphasized that in this project, the objective of sensitivity analysis is only to identify reservoir regions. History matching would ultimately be performed using a probabilistic approach only. The absolute magnitudes of the sensitivity coefficients are not important since they are not used in the history matching process.



(a) (b) (c)

Fig 6.6: Sensitivity coefficients on percentile plot for the 3rd layer of STAN5 (with 4th layer as base case for generating history) at a) 100 days, b) 500 days, and c) 1500 days.

3. Robustness of Domain Delineation

In this section the robustness of domains delineation based on the PC analysis has been demonstrated on two cases. For the two cases discussed below, global objective function has been used for the evaluation of Hessian. The objective function comprises of flux terms from all wells weighted equally. Each well also has unit weight as well in the global objective function. Measurement errors are assumed to be the same for all responses.

Case 1

Fig 6.7 (a) & (b) shows the two realizations used for this case. One realization was used for creating history while the other was used for history matching. The model has 2500 (50*50*1) grid blocks. Each grid block has dimension 100ft x 100ft x 30ft. The simulation parameters are summarized in Table 6.2. The history is generated for 450 days. There are four producers and three injectors. Fig. 6.7 (c) shows the sensitivity analysis. Fig. 6.7 (d)-3(g) show the domains based on scaled eigenvectors with different thresholds. Fig. 6.7 (h) –3(k) show the domains based on eigenvectors (not scaled) with different thresholds. For eigenvectors (not scaled) the individual thresholds for each eigenvector is kept same. The total variance extracted by first five eigenvectors is 96%. The regions near the wells are the most sensitive regions as expected. More importantly, the example illustrates the least correlation concept since the injector-producer couples fall in one domain. In other words the couple does not have much tendency to interact with other regions in reservoir. This is a particular case in which the injector-producer couples are situated in close proximity to one another.

Table 6.2: Reservoir model parameters for Case 1

Phases Present	Oil, Water and Gas
Reservoir Dimensions	50*50*1
Grid block Dimensions (ft)	100*100*30
Porosity	0.32
Number of wells	7
Well coordinates - Producers	(17,32); (17,7); (32,17); (42,27)
Well coordinates - Injectors	(7,12); (27,22); (12,42)

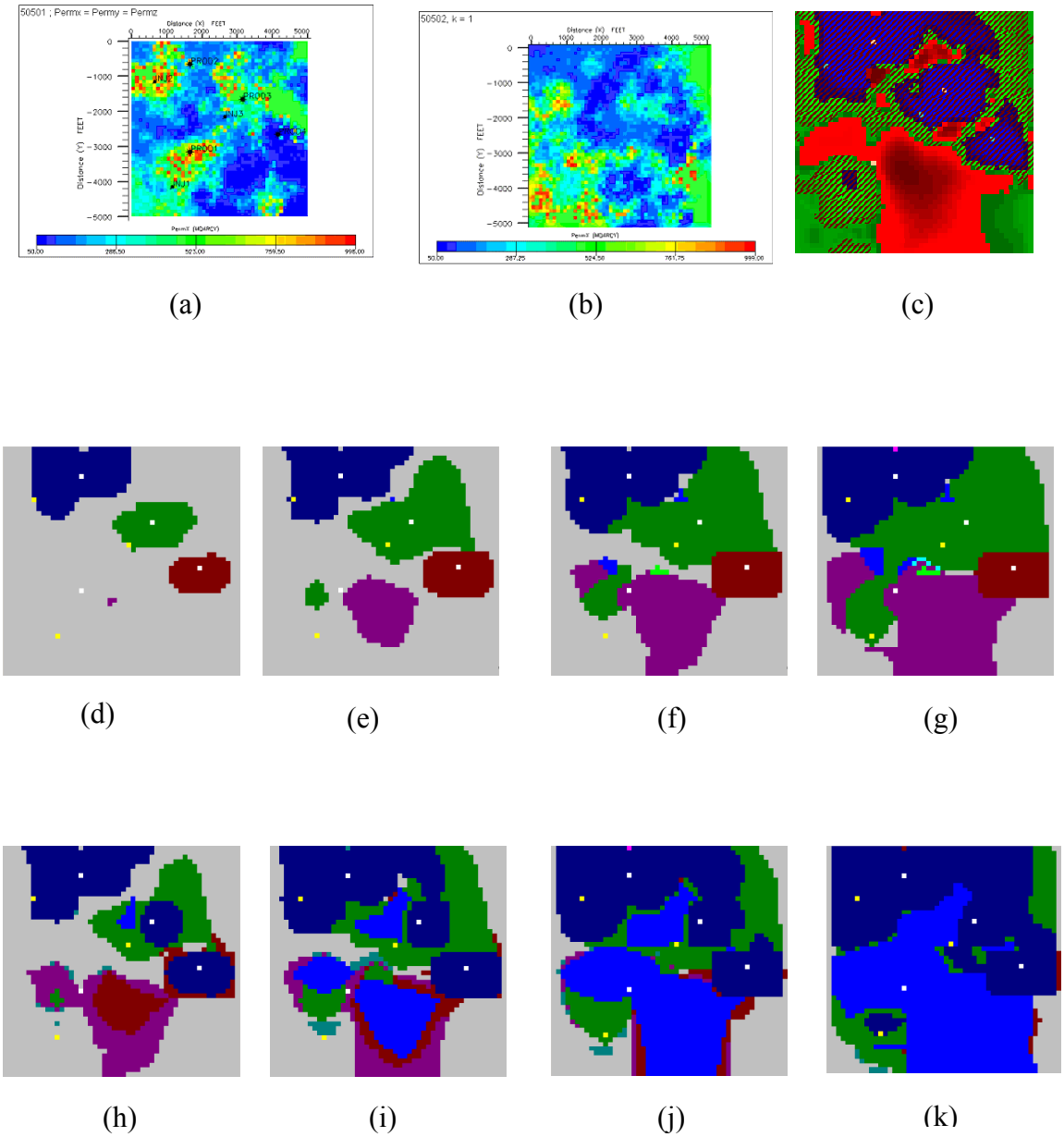


Fig 6.7 (a) Actual permeability field used to generate the history;

(b) Initial guess to be history matched (c) Sensitivity coefficients on percentile map. Highest values in red, intermediate in green and the lowest in blue, negative sensitivity coefficients regions are shown

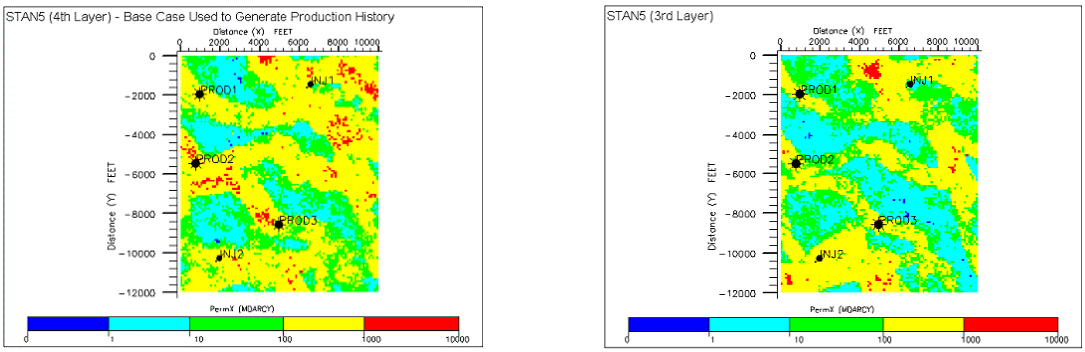
with shaded lines; (d), (e), (f), (g) show the domains obtained after scaling of eigen vectors with thresholds of 20%, 40%, 60% and 80% respectively; (h), (i), (j) and (k) show the domains with unscaled eigen vectors with thresholds of 20%, 40%, 60% and 80% respectively for all the five eigen vectors.

Case 2

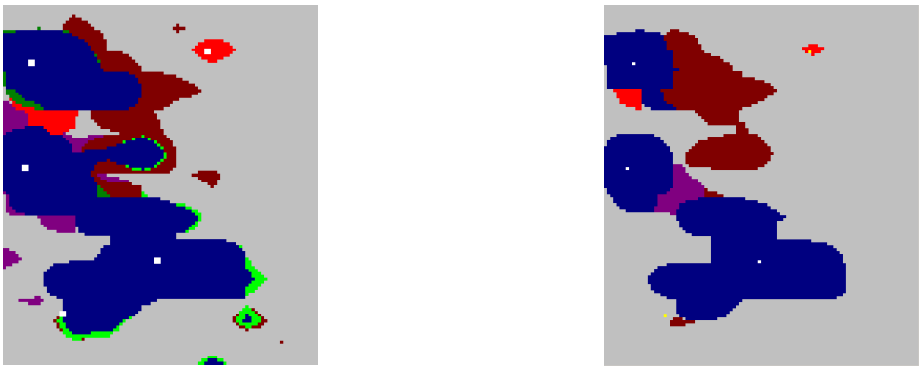
Fig. 6.8 (a) & (b) show the realizations used for history matching as well as for creating the history. The history is generated for 450 days. The model in this case has 12000 (100*120*1) grid blocks. Each grid block is 100ft in length, 100 ft wide, while the thickness is variable with the average being 16 ft. There are three injectors and two producers. Total variance extracted by the first five eigenvectors is 80%. Other details are provided in Table 6.3. The injector and producers are far apart in this case as compared to case 1. But still the concept of least correlated domains based on scaled eigenvectors can be readily appreciated from this case. It can be seen that the regions corresponding to the first eigenvector are disjoint and near each producing well. The disjointed domains near producer wells imply that any perturbation near one well would highly influence the performance of other producer wells. In other words the regions near the three producer wells should not be perturbed with different r_D parameters.

Table 6.3: Reservoir model parameters for Case 2

Phases Present	Oil, Water and Gas
Reservoir Dimensions	100*120*1
Grid block Dimensions (ft)	100*100*Variable (Average 16.07)
Porosity	Variable (Average 0.21)
Number of wells	5
Well coordinates - Producers	(10,20); (8,55); (50,86)
Well coordinates - Injectors	(66,15); (20,103)



(a) (b)



(c) (d)

Fig 6.8: (a) Showing the actual permeability field used to generate the history; (b) Initial guess to be history matched; (c) show the domains obtained without scaling of eigenvectors with thresholds of 30% for the five eigenvectors; (d) show the domains with scaled eigenvectors with thresholds 30%.

4. History matched synthetic cases

Two realizations generated by p-field simulation have been used. One realization was used for creating history while other was used for history matching. The model has 10000 (100*100*1) grid blocks. The grid blocks have dimension 100ft x 100ft x 30ft. History is generated for 450 days. Figures 6.9 (a) & (c) show the permeability realizations. Two extreme well configurations cases were tested on the same realization sets. The first case has just one well while the second case has four producers and two injectors. Global objective function has been used for the evaluation of Hessian. The objective function comprises of flux terms from all wells weighted equally.

Case 1

This case has one producer well. Details of the simulation are provided in Table 6.4.

Fig. 6.9 (b) shows the domains pertaining to 80% threshold after scaling the eigenvectors. Three domains are delineated i.e. the regions corresponding to the 1st positive, 2nd negative and 3rd positive eigenvectors. The 3rd positive domain is barely visible. The total variance extracted by first five eigenvectors is 99.6%. Different domains are perturbed at different nodes of the cluster. Six outer loops and six inner loops have been performed on each domain. The perturbed realizations are then merged. The merged realization has significantly less objective function value as compared to the starting guess. The merged realization is then subjected to a full field parameter perturbation. The objective function values are listed in Table 6.5. Fig. 6.9(d) shows the realizations after perturbing different domains at different nodes of the cluster. Fig. 6.9(e) shows the merged realization.

Fig. 6.9(f) shows the final history matched realization after perturbing globally using one deformation parameter. Fig. 6.10 shows the flow responses from the history-matched realization compared to the production history.

Table 6.4: Reservoir model parameters for case 1 of History matched examples

Phases Present	Oil, Water and Gas
Reservoir Dimensions	100*100*1
Grid block Dimensions (ft)	100*100*20
Porosity	0.25
Number of wells	1
Well coordinates - Producers	(41, 71)
Well coordinates - Injectors	-

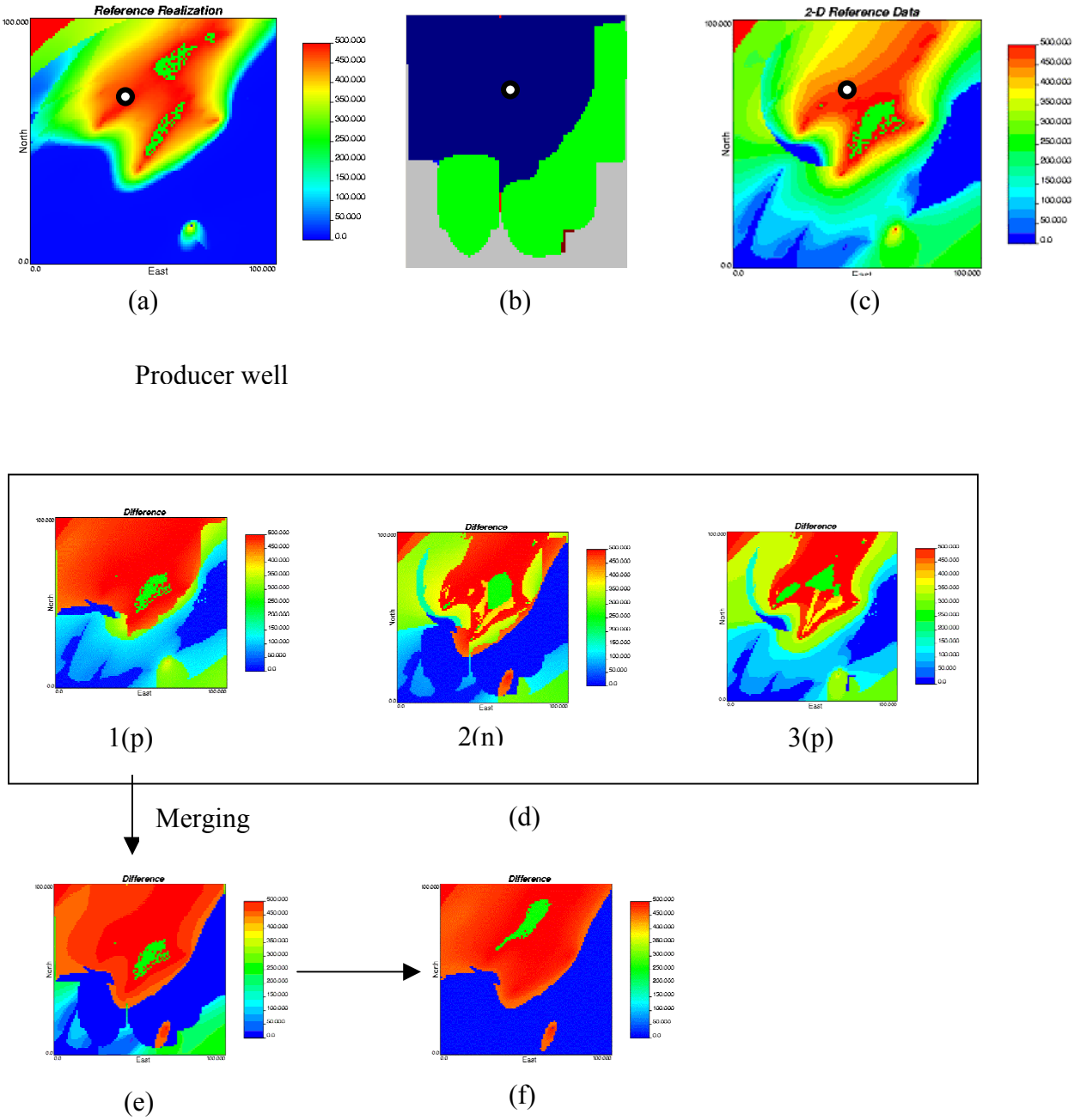
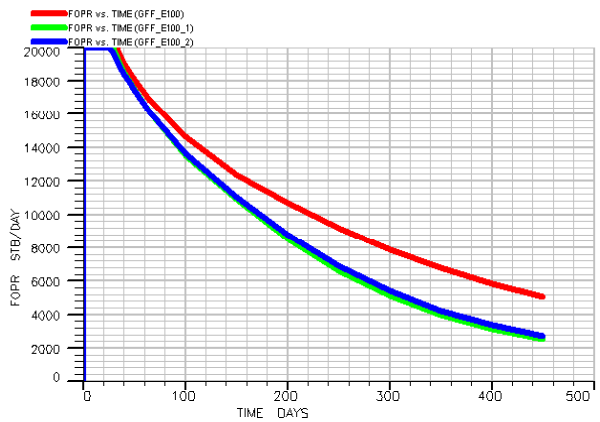


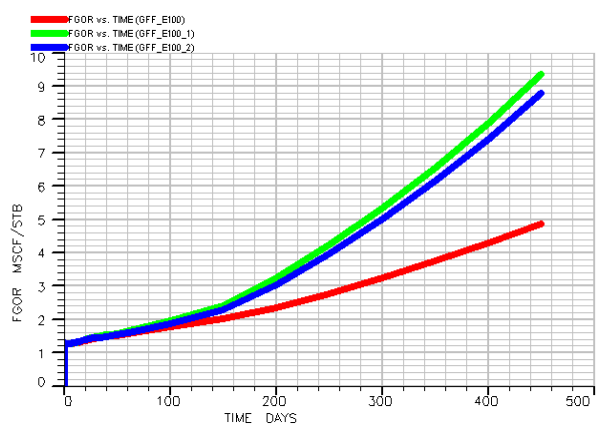
Fig 6.9: (a) Reference realization for creating history (b) Domains for 80% threshold (c) Initial Guess (d) Realizations after perturbing domain wise (e) Merged realization (f) Realization after perturbing the merged realization by one deformation parameter

Table 6.5: Showing the objective function for 1-well case

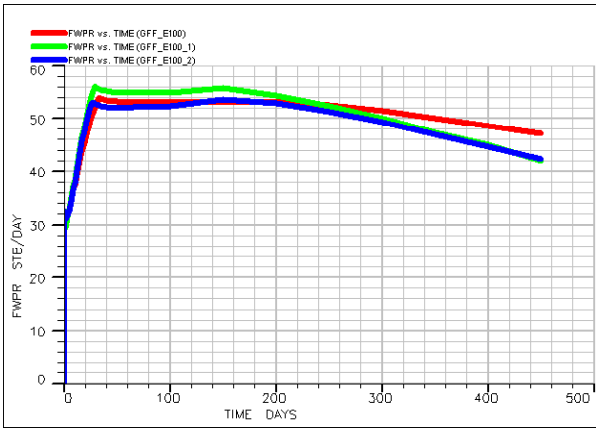
	Objective function
Initial guess	2.88
After preturbing 1p	2.19
After preturbing 1n	-
After preturbing 2p	-
After preturbing 2n	0.17
After preturbing 3p	2.18
After preturbing 3n	-
After preturbing 4p	-
After preturbing 4n	-
After preturbing 5p	-
After preturbing 5n	-
Merged Realization	0.25
After preturbing merged realization with one rd parameter	0.12



(a) Well oil production rate



(b) Well gas oil ratio



(c) Well water production rate

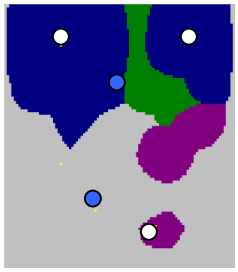
Fig 6.10: The red line is the initial guess, green is the target response and blue is the response from history matched realization

Case 2

This case has four producer wells and two injector wells. Simulation details are provided in Table 6.6. Fig. 6.11(a) shows the domains pertaining to 50% threshold after scaling the eigenvectors. Three domains are delineated i.e. regions corresponding to the 1st positive, 2nd positive and 4th positive eigenvectors. The total variance extracted by first five eigenvectors is 99.4%. Different domains are perturbed at different nodes of the cluster. As in case 1, six outer loops and six inner loops have been performed on each domain. The perturbed realizations are then merged. The merged realization has smaller objective function compared to the starting guess. The merged realization is then subjected to full field parameter perturbation. The objective function values are listed in Table 6.7. Objective function was reduced to 20% of the original after 36 iterations in the merged 50 % case. Fig. 6.11(b) shows the realizations after perturbing different domains at different nodes of the cluster. Fig. 6.11(c) shows the merged realization. Fig. 6.11(d) shows the final history matched realization after global perturbation.

Table 6.6: Reservoir model parameters for case 2 of History matched examples

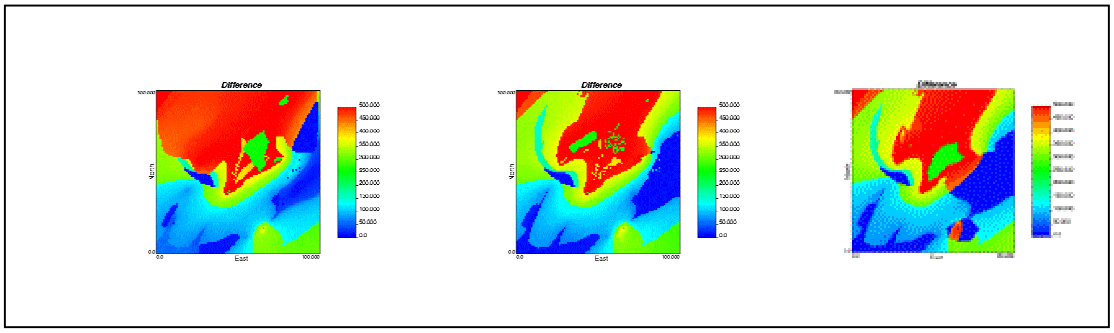
Phases Present	Oil, Water and Gas
Reservoir Dimensions	100*100*1
Grid block Dimensions (ft)	100*100*20
Porosity	0.25
Number of wells	6
Well coordinates - Producers	(25, 85); (80,90); (66,16); (25,40)
Well coordinates - Injectors	(40,22); (50,70)



Injector

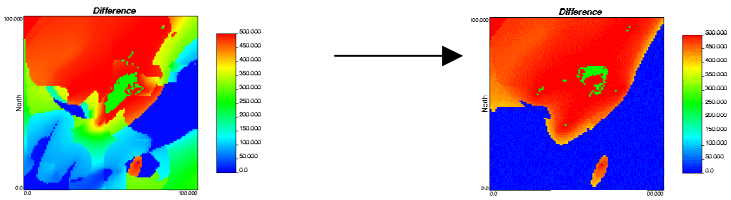
Producer

(a)



Merging

(b)



(c) (d)

Fig 6.11: (a) Domains for 50% threshold (b) Realizations after perturbing domain wise (c) Merged realization (d) Realization after perturbing the merged realization by one deformation parameter

Table 6.7: Showing the objective function for 6-well case

6 well Case	Objective function
Initial Guess	95.52
After perturbing 1p (50% case)	47.26
After perturbing 1n (50% case)	-
After perturbing 2p (50% case)	38.18
After perturbing 2n (50% case)	-
After perturbing 3p (50% case)	-
After perturbing 3n (50% case)	-
After perturbing 4p (50% case)	28.01
After perturbing 4n (50% case)	-
After perturbing 5p (50% case)	-
After perturbing 5n (50% case)	-
Merged Realization (50% case)	18.72
Merged Realization (80% case)	1195.75
After perturbing merged realization from 50% case with one rd parameter	19.35

Comparison of Case 1 and 2

The flow sensitivities at a grid block are influenced by the amount of flow through the grid block and the difference between the initial guess and the reference realizations of the permeability field. For the 1-well case the difference between the initial guess and the actual permeability field dominates the domain delineation. It can be seen from Fig. 6.9(b) that the transition between high to low permeability regions is generally identified as a separate domain. For the 6-well case the boundary conditions at the wells become the dominant influence in delineating zones (Fig. 6.11(a)). In the 1-well case the merged realization for 80% threshold has significantly less objective function than the initial realization. While for 6-well case, the objective function does not decrease as much as Case-1 in the merged case for 50% threshold. This is because of more interactions between nearly located wells. The objective function value is sensitive to the threshold value used for delineating the zones. The objective function increases by a factor of 12.5, if 80% threshold is taken for 6-well case. This is due to the fact that higher threshold increases the correlation among the domains.

Ideally domains should be recalculated as soon as some changes are induced in the permeability field. That would imply recalculation of costly Hessian matrix. Since there

is a gradual deformation of permeability field in the approach discussed, frequent updating of domains is not needed.

In order to further speed up the domain delineation process, the feasibility of using a reduced duration of production record for establishing the domains was assessed. A test case was run and sensitivity analysis was performed corresponding to different time durations of production record. The sensitivity coefficients do change in magnitude as the duration changes. But the relative values remain same (if all other factors i.e. permeability, well constrains, number of wells etc. remains the same). Hence the sensitivity coefficients on a percentile scale remain almost the same. This is shown in Fig. 6.6. This implies that sensitivity coefficients obtained after a shorter duration would be sufficient for delineating the domains, thereby expediting the process.

In all cases after scaling of eigenvectors, domain corresponding to either positive or negative component for a particular eigenvector is minimal and can be safely neglected. In case the regions corresponding to positive and negative eigen components for a particular eigenvector, cover significant grid block volume, they should not be perturbed on different nodes since they are not orthogonal. As well they cannot be perturbed on the same node using one-dimensional optimization since the regions correspond to opposite sensitivity regions (Fig.6.7 (c) & (g)). Therefore 2D optimization routine may be needed for theses special cases.

7. SENSITIVITY COEFFICIENTS AND CORRELATION COEFFICIENTS FROM A SET OF EQUI-PROBABLE REALIZATIONS

Calculating sensitivity coefficients is costly. Therefore an attempt was made to get the same information from flow simulations carried on a set of realization. In the conventional sensitivity analysis all other parameters except the one to which sensitivity is sought is kept same. In the cases discussed in this section the approach is to allow all the parameters to vary simultaneously such that parameters honor the prior variogram.

In all the cases discussed in this chapter there is

- One injector at (1,1,1) and a producer at (99,99,1)
- 100*100*1 gridblocks
- Three phases are present
- Injector is injecting water at a constant rate of 7500 bbl/day

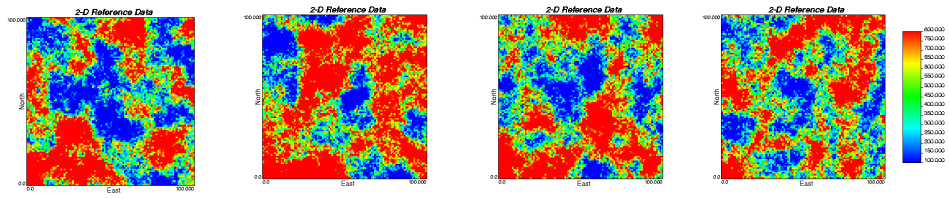
Sensitivity coefficients and correlation coefficients are calculated using the following procedure

a) Flow simulations are performed on the multiple realizations.
b) For every pair of realization, calculate the difference between the flow response (e.g. OPR, GOR etc.) parameter, **Δf**. Also calculate the difference between the permeabilities, **Δx**, for the same pair for respective grid blocks. Calculate the sensitivity, **Δf/Δx,** for the pair of realization in consideration.
c) Perform step (b) for each pair of realizations. Calculate the mean of the sensitivities thus calculated.
d) Correlation coefficients may also be obtained between the permeabilities at a particular grid block across all the realizations and the corresponding flow responses for the set of realizations.

The visual basic code for calculating sensitivities is attached as ‘Sensitivity Code 7.1’ in the appendix.

Case 1

The realizations for this case and the flow response are attached as ‘Case1_7.2’ & ‘Case1_7.3’ respectively in the appendix. The eclipse data file with full details is attached as ‘CASE1.DATA’ in the appendix.



(a) (b) (c) (d)

Fig 7.1: (a) to (d) are the four of total 50 realizations used for the analysis

The following are the plots for sensitivities and the correlation coefficients obtained using the 50 realizations.

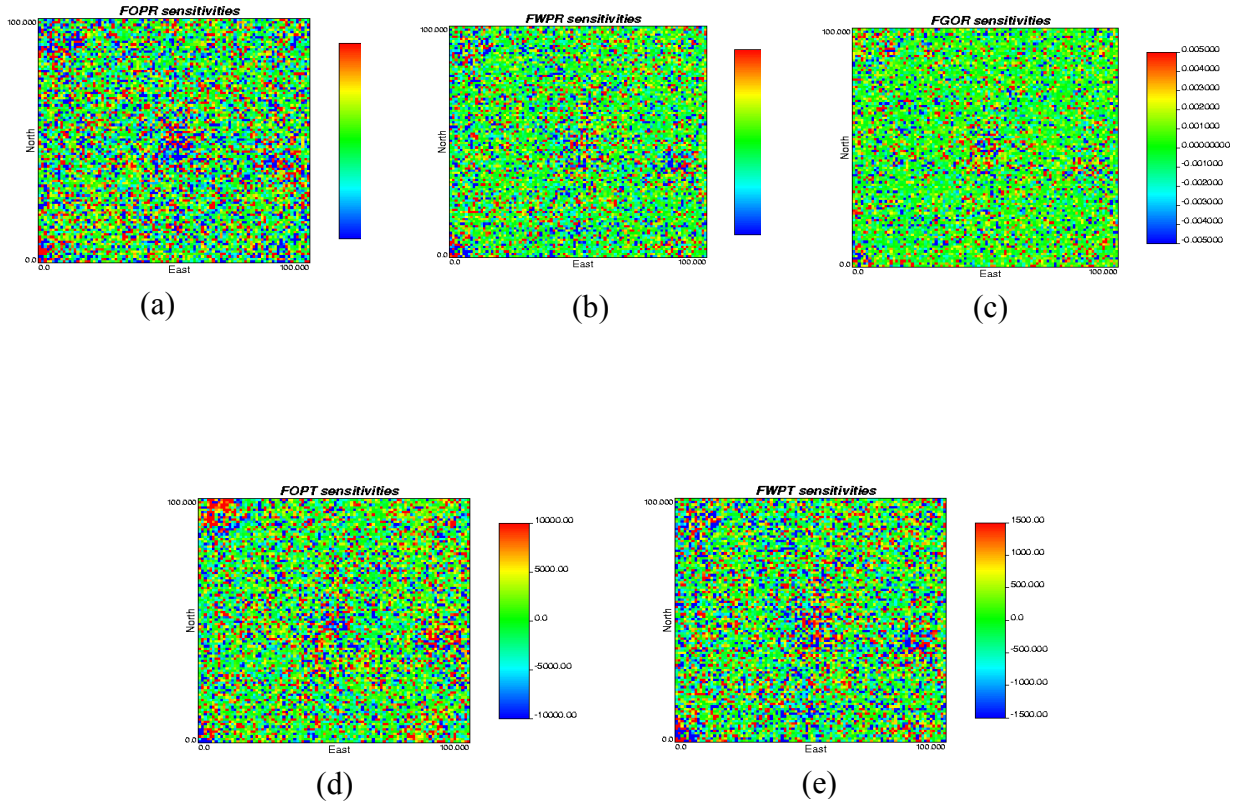


Fig 7.2: Averaged sensitivities calculated for the set of realizations with respect to a) Oil Production rate b) Well production rate c) Gas oil ratio d) Total oil production e) Total water production

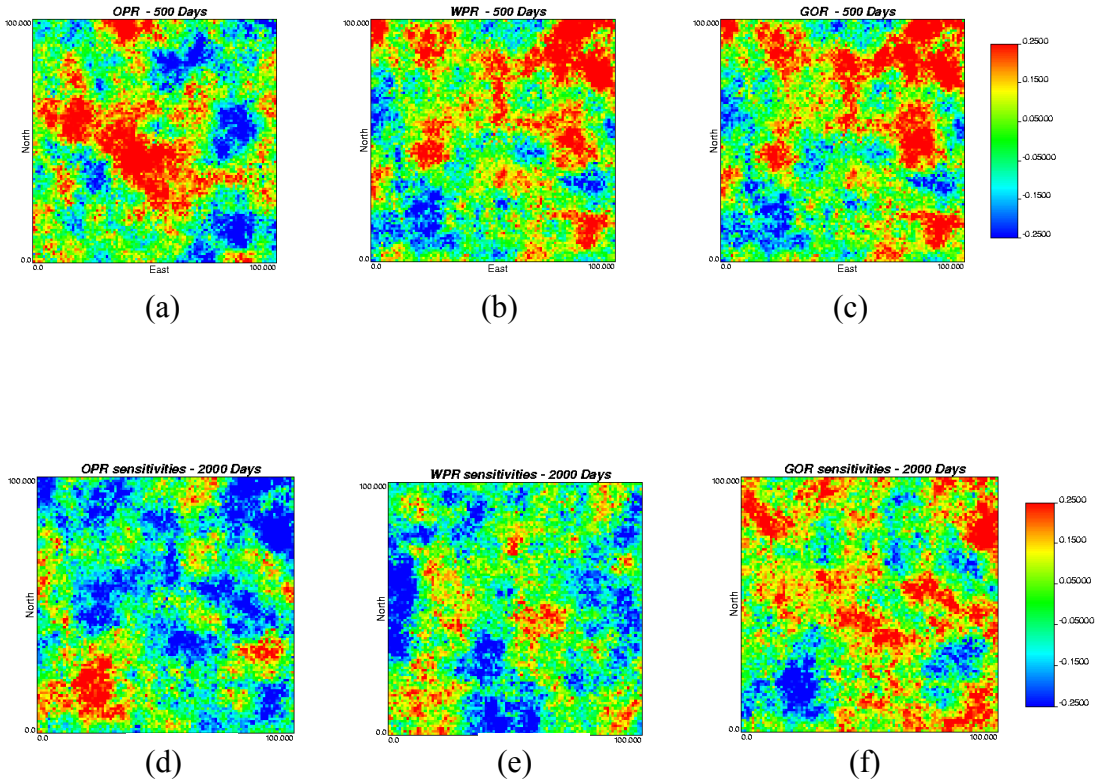
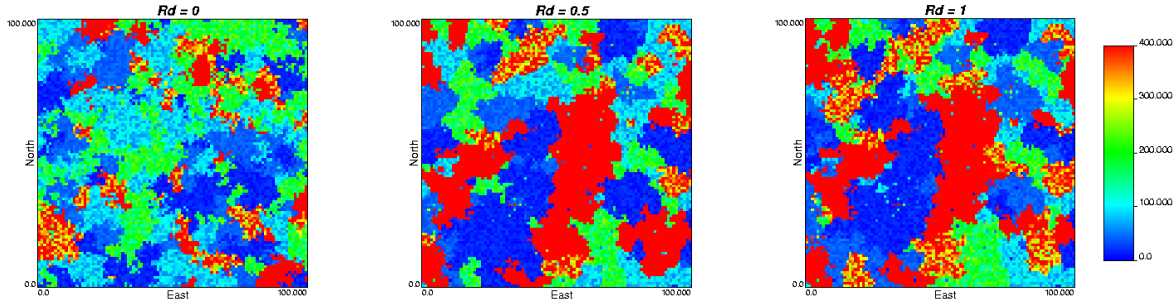


Fig 7.3: Correlation coefficients calculated over the set of realizations with respect to: a) Oil Production rate at 500 days b) Well production rate at 500 days c) Gas oil ratio at 500 days d) Oil Production rate at 2000 days e) Well production rate at 2000 days f) Gas oil ratio at 2000 days

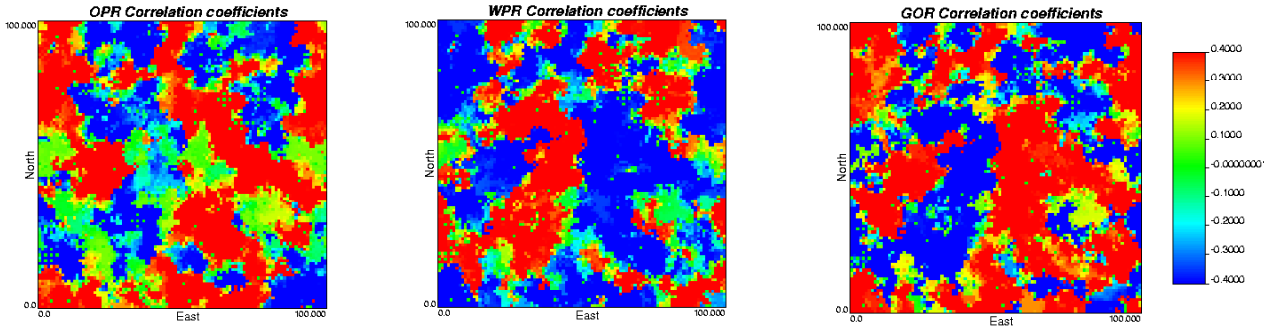
Case 2

The realization used for this case is constructed from different $\mathbf{r}_d$ guesses with a seed realization. A total of 51 realizations were used for the analysis. The realizations for this case and the flow response are attached as 'Case2_7.4' & 'Case2_7.5' respectively in the appendix. The eclipse data file with full details is attached as 'CASE2.DATA' in the appendix.



(a) (b) (c)

Fig 7.4: Permeability realization corresponding to different $\mathbf{r}_d$ values
$\mathbf{r}_d = 0$ b) $\mathbf{r}_d = 0.5$ c) $\mathbf{r}_d = 1$

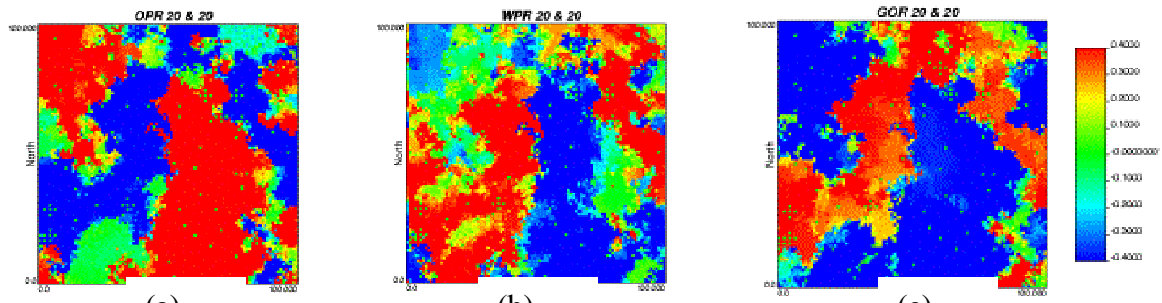


(a) (b) (c)

Fig 7.5: Correlation coefficients calculated over the set of realizations at 2000 days with respect to a) Oil Production rate b) Well production rate c) Gas oil ratio

Case 3

In this case the effect of correlation lengths on the correlation coefficients was investigated. The realization sets used for this case is constructed from different $\mathbf{r}_d$ values with a seed realization. A total of 51 realizations were used for the analysis. The realization set and the corresponding flow responses are attached as 'Case3_7.6' in the Appendix. The flow responses after 2000 days are provided at the end in the excel file 'Case3_7.6' in the respective sheets. The eclipse data file is the same as used for Case 2.



(a) (b) (c)

Fig 7.6: Correlation coefficients calculated over the set of realizations (with correlation lengths 20, 20) at 2000 days with respect to a) Oil Production rate b) Well production rate c) Gas oil ratio

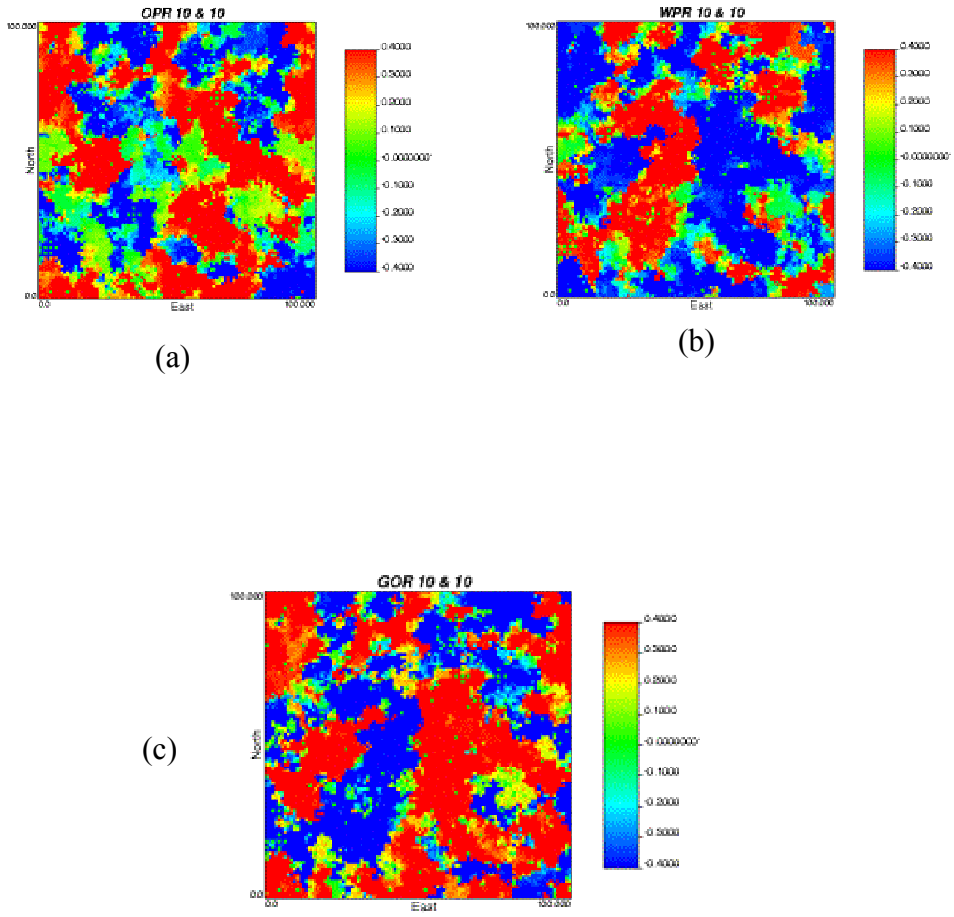
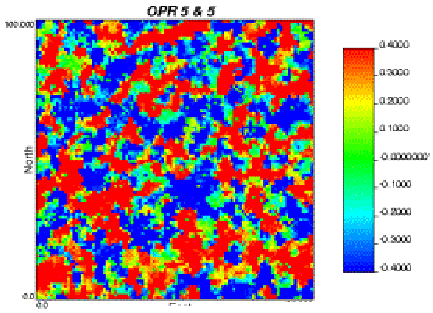
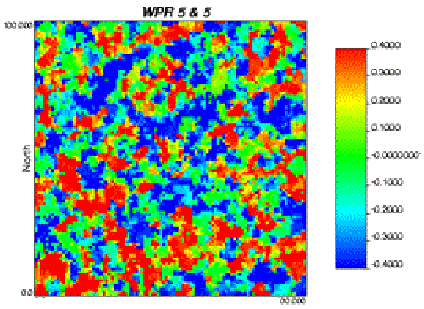


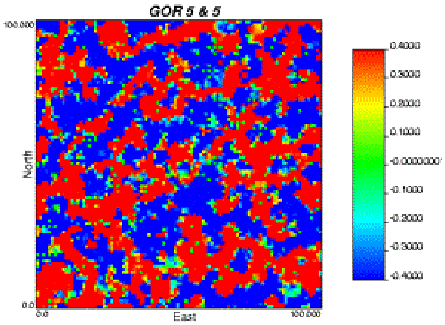
Fig 7.7: Correlation coefficients calculated over the set of realizations (with correlation lengths 10, 10) at 2000 days with respect to a) Oil Production rate b) Well production rate c) Gas oil ratio



(a)

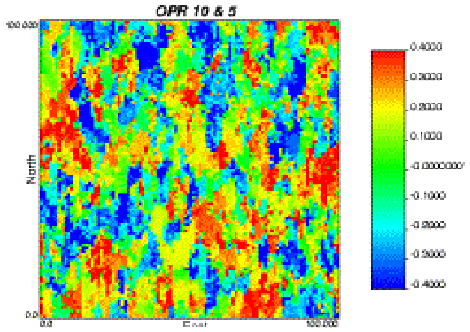


(b)

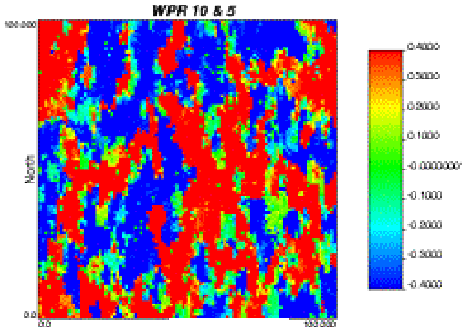


(c)

Fig 7.8: Correlation coefficients calculated over the set of realizations (with correlation lengths 5, 5) at 2000 days with respect to a) Oil Production rate b) Well production rate c) Gas oil ratio



(a)



(b)

(c)

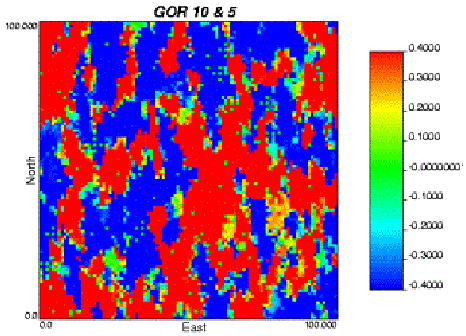


Fig 7.9: Correlation coefficients calculated over the set of realizations (with correlation lengths 5, 10 in x and y directions respectively) at 2000 days with respect to a) Oil Production rate b) Well production rate c) Gas oil ratio

Case 4

In the three cases discussed above there is no second derivative information, which is needed to delineate the least correlated regions. Also the regions obtained were not near the wells, which is counterintuitive.

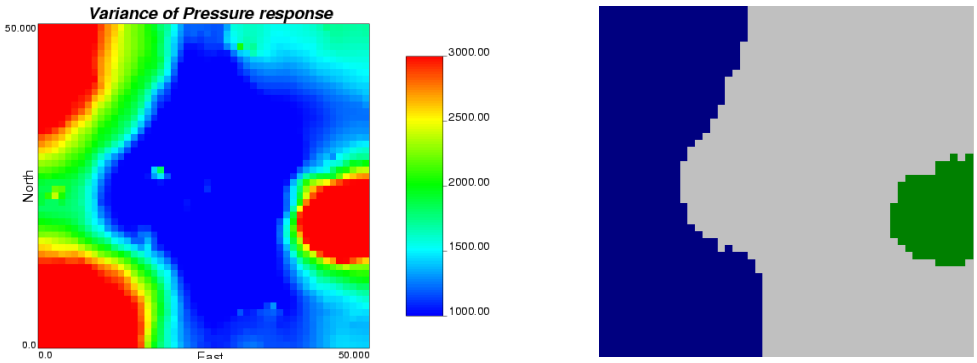
So a different methodology was adopted. The method is simple and is described below:

1) Simulate a flow parameter (Pressure, Saturations etc.) at different gridblocks from the set of realizations
2) Obtain the variance of the flow parameter across different realizations at each gridblock. The variance is intuitively analogous to sensitivity.
3) Calculate the covariance matrix for the simulated flow parameters from the multiple simulated values at each gridblock.
4) Subject the covariance matrix to Principal Component Analysis and obtain the most sensitive and least correlated regions.

For this case there are

- Three producers at (10,10,1), (10,90,1), (90,40,1) and injectors at (20,10,1), (20,90,1), (90,60,1)
- 100*100*1 gridblocks
- Three phases are present
- Simulation is run for 500 days
- Injector is injecting water at a constant rate of 3000 bbl/day
- Producers are Oil rate controlled at 3000 bbl/d and a lower BHP limit of 600 psia

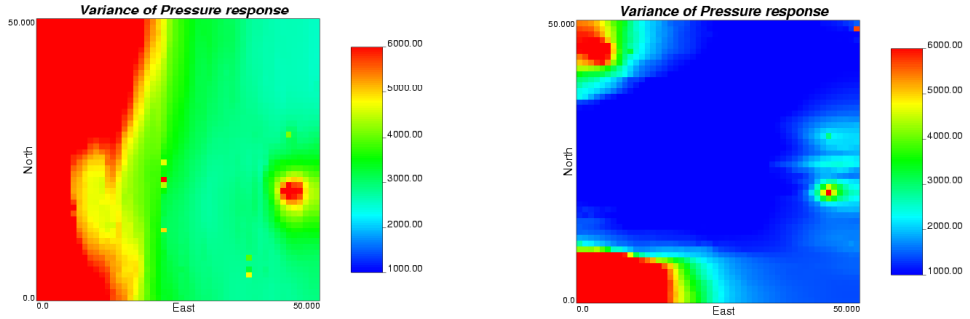
This method gives extremely good results and provides the regions, which could be intuitively justified.



(a) (b)

Fig 7.10 (a): Variance of the pressure or the sensitivities from a set of realizations; The figure has been scaled down to 50*50 so that the final covariance matrix used for PCA is manageable. (b) Most sensitive and least correlated regions obtained after subjecting the covariance matrix to PCA.

The following figures demonstrate the effect of change in underlying permeability realizations to the sensitive regions. The right figure (fig 7.11(a)) corresponds to realizations with correlation lengths 30 and 5 units in y and x directions respectively. The left figure (fig 7.11(b)) corresponds to realizations with correlation lengths 5 and 30 units in y and x directions respectively.



(a)

Fig 7.11: Variance of the pressure or the sensitivities

The following inferences can be made from the cases discussed above

1. Sensitivity coefficient map obtained as per the procedure mentioned earlier is very smooth with no information content in it. The map seems to be white noise with no structure. A reason could be that the averaging process wipes out the correlation structure of the sensitivity coefficients.

2. Correlation coefficients do have some information stored in them. Further investigation is needed to in order to formulate a scheme to use this information for history matching.

3. Correlation coefficients appear to change as a function of time. Certain regions, which were highly correlated at earlier times, may become less correlated to the flow response with time.

4. Correlation coefficient maps appear to be become less continuous as the correlation lengths decreases in the set or realizations used to calculate the correlation lengths. There is high degree of association between the correlation lengths of the underlying set realizations and the structure obtained on the correlation coefficient map.

5. The correlation coefficient approach provides the degree to which the permeability at a particular gridblock is correlated with the flow response from various realizations. It might give an insight to sensitivities, but these are inadequate to get least correlated regions, which are instrumental for the distributed computing application. In other words second derivative information is needed for distributed computing. A plausible approach may be calculating the covariance among the correlation coefficients. This could be done by having multiple sets of realization giving multiple sets of correlation coefficients. Theses

multiple correlation coefficient map may then be used for calculating the correlation among the correlation coefficients.

6. There is little resemblance between the sensitivity coefficients obtained for the Simopt® and the correlation coefficients obtained from the set of realizations. This might be attributed to the fact the sensitivity coefficients calculated using Simopt are based on one realization. Hence it is realization specific, while the correlation coefficient gives a generic parameter quantifying how the flow is affected if the permeability is changed at particular grid block across the set of realization. The other point to be noted is that permeability at other locations are also getting changed in set of realizations used for calculating the correlation coefficients.

7. **The most sensitive and least correlated regions defined based on the PCA on the covariance matrix of a flow parameter (which was Pressure in Case 4 in Chapter 7) simulated over a set of realizations, gives good results and provides the regions which could be intuitively justified. One of the weak point of this methodology is that it doesn't take into account the history (or the reference). In the conventional sensitivity analysis, which takes into account the history, one can ascertain whether the model parameter needs to increased or decreased from the sign of the sensitivity coefficient. But this approach would only provide absolute sensitivity of the model parameter, which serves well the requirements of the current project, because the perturbation is not guided by the sign of sensitivity coefficient.**

EXPERIMENTAL METHODS/APPROACH

In this project we are developing algorithms and implementing them on computers. The scope of work does not include experiments.

RESULTS AND DISSCUSSIONS

1. A computer code implementing the Probabilistic approach of History matching has been developed and validated on various test cases.

2. **Global rather than well specific** objective function should be minimized by perturbing permeability values locally within domains.

3. Reservoir regions based on sensitivity coefficients calculated for an initial guess of the permeability field that is far from the "truth" **may not be very reliable**. The results with the ensemble of realizations indicate that it may be better to base the reservoir regions on the average response of the ensemble rather than use any particular initial guess.

4. These domains are delineated on the basis of sensitivity and least correlation. Principal component analysis of the Hessian matrix and **subsequent scaling** provide the domains that are sensitive and least correlated. This has been verified from test cases. Regions defined on the basis of sensitivity increases the effectiveness of the history matching process. The least correlation criteria make the problem amenable to distributed computing and hence imparting efficiency in terms of computational time. Least correlation also helps in retaining the simplicity of 1-D optimization.

5. The grid block volume cut-offs should be reasonable such that the regions covered by eigenvectors are not too large (which increases the correlation) nor too small (implying ineffective history matching). If the wells are located near

to each other then it becomes difficult to isolate least correlated regions and the merged realization may have higher objective function. In these cases the smaller thresholds for domain delineation should be used.

6. In case there is high degree of overlap among the eigen components of different eigenvectors, rotation of eigenvectors by relaxing the constraint of sensitivity but maintaining orthogonality has been suggested. This approach has not yet been tested on history matching examples.

7. Alternate ways of domain decomposition

a. Principal component analysis of permeability covariance matrix
    i. Regions with the highest variance are identified as the most sensitive. Regions with less covariance (different facies) among them are identified as least correlated. The drawback being that the analysis is just based on the static data. It is very much possible that there is very high flow across the facies due to particular well locations and facies are correlated from flow perspective but not statically. Also high variance (or high sensitivity statically) regions may have not much flow associated with them and thus would not affect the objective function in any significant way.

b. Domain decomposition based on porosity and thickness
    i. From the test cases it appeared that sensitivity coefficients are a strong function of the permeability realization in consideration. Geobodies defined on the basis on porosity and thickness were inadequate to define the domains.

8. In order to counter the high computational cost associated with sensitivity analysis an approximation based on analysis of limited duration of production data has been proposed.

9. The following codes are ready to be used
    i. A visual basic code for domain delineation. This code is very user friendly and requires minimal inputs like number of grid blocks and the volume threshold. Domains can be visualized layer by layer. The code provides the indicator -5 to + 5 at each gridblock, depending upon which eigenvector dominates at that particular gridblock. The problem of having the negative and positive regions for a particular eigenvector is not encountered in any of the cases till tested. In those cases this problem arises, the two dimensional optimization may be needed.

    ii. A visual basic code for the modified geobodies as mentioned in chapter.

    iii. A script to distribute the simulation over on different nodes of the cluster

10. Sensitivity coefficients and Correlation coefficients from a set of realizations
    - Sensitivity coefficient map obtained as per the procedure mentioned is very smooth with no information content in it. The map seems to be white noise with no structure. A reason could be that the averaging process wipes out the correlation structure of the sensitivity coefficients.

    - Correlation coefficients do have some information stored in them. Further investigation is needed to in order to formulate a scheme to use this information for history matching.

- Correlation coefficients appear to change as a function of time. Certain regions, which were highly correlated at earlier times, may become less correlated to the flow response with time.

- Correlation coefficient maps appear to be become less continuous as the correlation lengths decreases in the set or realizations used to calculate the correlation lengths. There is high degree of association between the correlation lengths of the underlying set realizations and the structure obtained on the correlation coefficient map.

- The correlation coefficient approach provides the degree to which the permeability at a particular gridblock is correlated with the flow response from various realizations. It might give an insight to sensitivities but these are inadequate to get least correlated regions, which are instrumental for the distributed computing application. In other words second derivative information is needed for distributed computing. A plausible approach may be calculate the covariance among the correlation coefficients. This could be done by having multiple sets of realization giving multiple set of correlation coefficients. Theses multiple correlation coefficient map may then be used for calculating the correlation among the correlation coefficients.

- There is little resemblance between the sensitivity coefficients obtained for the Simopt® and the correlation coefficients obtained from the set of realizations. This might be attributed to the fact that the sensitivity coefficients calculated using Simopt are based on 1 realization. Hence it is realization specific while the correlation coefficient gives a generic parameter quantifying how the flow is affected if the permeability is changed at particular grid block across the set of realization. The other

point to be noted is that permeability at other locations are also getting changed in set of realizations used for calculating the correlation coefficients.

- **The most sensitive and least correlated regions defined based on the PCA on the covariance matrix of a flow parameter (which was Pressure in Case 4 in Chapter 7) simulated over a set of realizations, gives good results and provides the regions which could be intuitively justified. One of the weak point of this methodology is that it doesn't take into account the history (or the reference). In the conventional sensitivity analysis, which takes into account the history, one can ascertain whether the model parameter needs to increased or decreased from the sign of the sensitivity coefficient. But this approach would only provide absolute sensitivity of the model parameter, which serves well for the requirements of the current project, because the perturbation is not guided by the sign of sensitivity coefficient.**

CONCLUSIONS

A novel methodology for delineating multiple reservoir domains, for the purpose of history matching in a distributed computing environment has been proposed and tested. The domains are delineated on the basis of sensitivity and least correlation. Sensitive regions manner increases the effectiveness of the history matching process. The least correlation criteria make the problem amenable to distributed computing and hence imparting efficiency in terms of computational time. Least correlation also helps in retaining the simplicity of 1-D optimization.

Since delineation of domains would require calculation of Hessian which could be computationally costly and as well as would have restricted the current approach to some specific simulators, therefore a robust technique to get those regions from a set of equi-probable realizations has been developed. This technique is easy to implement and provides the regions, which could be intuitively justified.

The new thing in this work is scaling of eigenvectors, defining optimal thresholds based on grid block volumes and using the thus defined domains for history matching in a distributed environment using probabilistic approach. Also a new and robust technique has been developed to delineate most sensitive and least correlated regions without evaluating the Hessian of the objective function.

The use of flow sensitivities for guiding the permeability perturbation by constraining the random number draw such that perturbation is according to flow sensitivity sign has been deliberately avoided. The constraint would have improved the convergence but could have taken away the freedom in the probabilistic approach to escape the local minima. Sampling from a PDF using a random number helps in escaping local minima. But the down side is that even higher objective function values may be accepted during the iteration.

COST AND SCHEDULE SECTION

Categories	Yr. 1- 9/1/03- 3/31/05 Budget- 1st Increment Funding	Funds Redirected	Adjusted Balance	2nd Increment Funding 4/1/05- 3/31/06	Total Funding to Date thru 8/31/05	Expenditures 9/1/03-8/31/04	Expenditures 9/1/04 thru 8/31/05	Total Expenditures to Date 8/31/05	Balance Remaining for period 9/1/05-3/31/06
Salaries (12)	$82,074	$5,800	$87,874	$83,112	170986	$88,664	$7,007.00	$95,671	$75,315
Fringe Benefits (14)	$18,944	$3,500	$22,444	$19,183	41627	$17,946	$647.98	$18,594	$23,033
MO&E	$1,875		$1,875	$1,875	3750	2604.17	2571.85	$5,176	-$1,426
Subcontract (67)	$0	$0	$0	$12,400	$12,400	$2,914	0	$2,914	$9,486
Tuitition & Fees (71)	$13,200	$12,450	$25,650	$13,199	38849	$27,269	$3,253.49	$30,523	$8,326
Travel (75)	$2,600	$0	$2,600	$4,941	7541	$3,180	$177.60	$3,357	$4,184
Special Equipment (80)	$22,500	-$17,600	$4,900	-$4,900	0	$0	$0	$0	$0
Overhead Expense (90)	$63,997	-$4,150	$59,847	$58,306	118153	$55,361	$4,351.53	$59,713	$58,440
INCOME	$205,190	$0	$205,190	$188,116	393306	$197,937	$18,009.45	$215,947	$177,359
UT's Cost Share Breakdown by Account				**Total**	**for Period 9/1/03-8/31/05**				
Steven Bryant	Salaries		$17,030						
"	FB		$4,358						
		Subtotal	$21,388						
	OH	at 50%	$10,694	**$32,082**	**Acct # 14-3085-20XX)**				
Total									
Sanjay Srinivasan	Salaries		$16,947						
"	FB		$3,909						
		subtotal	$20,856						
	OH	at 50%	$10,428						
Total				**$31,284**	**Acct 14-3085-20XX**				
Computer Cluster			19558	**19558**	**Accts 20-3085-2457 & 20-3085-2557**				
UT-Portion - Total Cost Share to date (8/31/05)				**$82,924**					
Cost Share Required on $215,947 (expenditures thru 8/31/05)				**$63,305**					

SUMMARY OF SIGNIFICANT ACCOMPLISHMENTS DURING REPORTING PERIOD

1) Robustness of the PCA for domain delineation has been successfully tested.
2) The full algorithm has been tested on 2-D synthetic cases.
3) A new and robust technique has been developed to delineate most sensitive and least correlated regions without evaluating the Hessian of the objective function. This would eliminate the need of costly Hessian, which could be computationally costly, and as well as wouldn't restrict the current approach to any some specific simulator.

FUTURE WORK

1. Test the methodology on complex 3-D cases.
2. Integrate the three pieces of code i.e.
    a) Domain delineation code
    b) History matching code
    c) Script used for distributing the simulations

REFERENCES

1. Richard Reyment, K.G. Joreskog, "Applied Factor Analysis in the Natural Sciences"

2. Bissell, R. "Calculating Optimal Parameters for History Matching": 4th European Conference on the Mathematics of Oil Recovery, Roros, Norway

3. R.C. Bissel, Yogeshwar Sharma, J.E. Killough, "History Matching Using the Method of Gradients: Two Case Studies", SPE 28590

4. Richard L. Gorsuch, "Factor Analysis", Second Edition.

5. King, M.J. and Dutta-Gupta, A., and Yoon, S.: "Streamline Simulation: A current Perspective," In Situ (1988) 22, No. 1, 91.

6. Oliver, D.S., Renyolds, A.C., Bi, Zhuoxin and Abacioglu, Y., "Integration of Production data into Reservoir models", Petroleum Geoscience, (7), S65-S73, 2001.

7. Iterative Updating of Reservoir Models Constrained to Dynamic Data CIM paper 2002-125, Kashib, T. and Srinivasan, S

8. Journel, A.G.: " Combining knowledge from diverse sources: An alternative to traditional data independence hypotheses ", Mathematical Geology, V.34, N.5: 573-596, 2002.

9. Srivastava, R.M., 1992, Reservoir characterization with probability field simulation, SPE 24753, 67th SPE Annual Technical Conference and Exhibition, 927-938

10. Deutsch, C.V., Journel, A.G.(1998): GSLIB: Geostatistical Software Library and Users Guide, Oxford University Press Inc., Oxford, New York 1992.

11. Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P.: "Numerical Recipes in C: The art of scientific computing", Cambridge University Press, 1992

12. Mao, S. and Journel, A.: 1999, Generation of a reference petrophysical/seismic data set: the stanford v reservoir, Report 12, Stanford Center for Reservoir Forecasting, Stanford, CA.

13. Eclipse® Technical Manual, 2003A, Schlumberger

14. Simopt® Manual, 2003A, Schlumberger

15. R.C. Bissel, O. Dubrule, P. Lamy, P. Swaby and O. Lepine, "Combing Geostatistical Modelling With Gradient Information for History Matching: The Pilot point Method", SPE 38730

16. Leitao, H.C.; Schiozer, D.J., "A new automated History Matching Algorithm improved by parallel computing", SPE 53997, SPE Latin American and Carribbean Petroleum Engineering Conference held in Venezuela, 1999.

17. W.A. Habiballah, M.E. Hayder, M.S. Khan, K.M. Issa, S.H. Zahrani, R.A. Shaikh, A.H. Uwaiyedh, T.P. Tyraskis, M.A. Baddourah. "Parallel reservoir simulation Utilizing PC Clusters", SPE 84065, SPE Annual Technical Conference held in Colorado, 2003.

18. D.J. Schiozer, S.H.G. Sousa, "Use of Reservoir simulation, Parallel computing and Optimization techniques to accelerate History Matching and Reservoir management decisions", SPE 53979, Fifth Latin American and Caribbean Petroleum Engineering Conference and Exhibition, Brazil, 1997.

19. D.J. Schiozer, S.H.G. Sousa, "Use of external parallelization to improve history matching", SPE 39062, Fifth Latin American and Carribean Petroleum Engineering Conference and Exhibition held in Rio De Janeiro, 1997.

20. Ahmed Quenes, William Weisss, A.J. Sultan, Abu Dhabi Nati, Jaber Anwar, "Parallel Reservoir Automatic History matching using a network of Workstations and PVM", SPE 29107, SPE Symposium on Reservoir Simulation held in San Antonio, 1995.

21. Mengjiao Yu, Phani B. Gadde and Mukul M. Sharma, "Process based Decomposition on Distributed Systems", SPE 77721, Annual Technical Conference and Exhibition held in San Antonio, 2002.

22. J.E. Killough, "Is parallel computing ready for reservoir simulation", SPE 26634, 68th Annual Technical Conference and Exhibition of Petroleum Engineers held In Houston, 1993.

23. Ahmed Quenes, Naji Saad, "A New Fast Parallel Simulated Annealing Algorithm for Reservoir Characterization", SPE 26419, 68th Annual Technical Conference and Exhibition of SPE held in Houston, 1993.

24. Luc Anguille, J.E. Killough, T.M.C. Li and J.L. Toepfer, "Static and Dynamic Load Balancing Strategies for Parallel Reservoir Simulations", SPE 29102, 13th SPE Symposium on Reservoir Simulation held in San Antonio, 1995.

25. Jorge L. Landa, Sebastien Strebelle, "Sensitivity analysis of Petrophysical Properties Spatial Distributions and Flow Performance Forecasts to Geostatistical Parameters

using Derivative Coefficients", SPE 77430, SPE Annual Technical conference in San Antanio, 2002

26. H.A. Tchelepi, L.J. Durlofsky, W.H. Chen, A. Bernath and M.C.H. Chien, "Practical use of scale up and Parallel Reservoir Simulation Technologies In Field studies", SPE 38886, SPE Annual Technical Conference and Exhibition in San Antonio, 1997.

27. C.E. Romero, J.N Carter, R.W. Zimmerman, A.C. Gringarten, "Improved Reservoir Characterization through Evolutionary Computation", SPE 62942, SPE Annual Technical Conference and Exhibition in Dallas, 2000.

28. F. Roggero, L.Y. Hu, "Gradual deformation of Continuous Geostatistical Models for History Matching", SPE 49004, SPE Annual Technical conference in Louisiana, 1998.

29. Bertrand Brun, Olivier Gosselin, John W. Barker, "Use of prior information in Gradient-Based History Matching", SPE 66353, SPE Resevoir Simulation Symposium held in Houston, 2001.

30. Suasana Gomez, Olivier Gosselin, John W. Barker, "Gradient Based History Matching with Global Optimization Method", SPE 56756, Annual Technical Conference of SPE held in Houston, 1999.

31. P.C. Shah, G.R. Gavalas, J.H. Seinfeld, "Error analysis in history matching: The optimum level of Parameterization", SPE 6508, 1977.

32. Sophie Verdiere, Lisette Quettier, Pierre Samier, Alan Thombson, "Application of Parallel simulator to industrial cases", SPE 51887, SPE Reservoir Simulation Symposium held in Houston, 1999.

33. Iterative Updating of Reservoir Models Constrained to Dynamic Data. CIM paper 2002-125, Kashib, T. and Srinivasan, S

34. Ahmed Quenes, B. Brefort, Gilbert Meunier, Simon Dupere, "A New Algorithm for Automatic History Matching: Application of Simulated Annealing Method to Reservoir Inverse Modeling, SPE 26297.

35. Schiozer, D.J., "Use of Reservoir Simulation, Parallel Computing and Optimization Techniques to accelerate History Matching and Reservoir Management Decisions", SPE 53979, SPE Latin American and Carribean Petroleum Engineering Conference held in Venezuela, 1999.

36. Blanc, G., Hu, L.Y., Noetinger, B., "Gradual deformation and iterative cal liberation of sequential stochastic simulation", Mathematical Geology, V.33, N.4:475-489, 2001.

37. Caers, J: "Markov chain theory for spatial stochastic simulation". Report No. 12, Volume 1, May 1999, Stanford for Reservoir Forecasting, Stanford University.

38. Caers, J: "Methods for history matching under geological constrains", 8th European Conference on Mathematics of Oil Recovery, 2002.

39. Chu, L., Renyolds, A.C. and Oliver, D.S.: " Computation of sensitivity coefficients for conditioning the permeability field to well test pressure data", In Situ, V.19, N.2:179-223, 1995.

40. J.E. Killough, M.F. Weeler, "Parallel Iterative Linear Equation Solvers: An investigation of Domain Decomposition Algorithms for Reservoir Simulation", SPE 16021, Ninth SPE symposium on reservoir simulation held in San Antonio, 1987.

41. Sanjay Srinivasan, Steven Bryant, "Integrating Dynamic Data in Reservoir Models Using a Parallel Computational Approach", SPE 89444

42. R.W. Schulze-Riegert, O. Haase, A. Nekrassov, "Combined Global and Local Optimization Techniques Applied to history Matching", SPE 79668

43. Jef Caers, "Methods for History Matching under Geological Constraints"

44. Th. Grussaute, P. Gouel, " Computer Aided History Matching of a Real Field Case", SPE 50642

45. Ning Liu, Dean S. Oliver, "Automatic History Matching of Geological Facies" SPE 84594

46. Matlab, The MathWorks Inc.