IMPERIAL COLLEGE LONDON

Department of Earth Science and Engineering

Centre for Petroleum Studies

Construction of the Midge History Match Model

By

Lukman Wahab

A report submitted in partial fulfilment of the requirements for the MSc and the DIC

September 2010

DECLARATION OF OWN WORK

I declare that this thesis [CONSTRUCTION OF THE MIDGE HISTORY MATCH MODEL] is entirely my own work and that where any material could be construed as the work of others, it is fully cited and referenced, and/or with appropriate acknowledgement given.

Signature: .....

Name of student: **Lukman Wahab**

Name of supervisor: **Dr. Jonathan Carter**

ABSTRACT

This paper attempts to tackle the inverse problem of History Matching (HM) on a simplified version of a real reservoir model (Midge Reservoir Model donated by BP) that has been used to make reservoir management decisions. The ability to make correct reservoir management decisions relies mainly on being able to predict their consequences, which in turn depends on the achieved accuracy in the description of the reservoir's internal structure (Gringarten, 1998). This reservoir model is parameterised in 18 regions or compartments, each having homogeneous properties. This parameterisation approach is intended to reduce the number of model parameters. These parameters are mainly the porosities (PORO) and permeability (PERMX, PERMY, PERMZ) in the various compartments, the transmissibility across the faults and the allocation of water to the injection wells (Carter et al. 2006).

Different realisations of the Midge model were generated by searching the parameter space in a systematic way using a set reservoir parameters range (first varying single parameters in each region: uni-variate cases, then varying two different parameters for each region: bi-variate cases). Synthetic production parameters (Field Oil Production Rate – FOPR, Field Water Production Rate – FWPR, Field Gas Production Rate – FGPR) were then generated from each of the models; these were compared with the results from the true model which is the base case. The optimization technique used was to find the minimum of an objective function that best represents the quality of the model. For multivariate parameters variations, a hill climbing algorithm was used to search for all possible 'best models'.

In most of the studies, independent of the method used for the history matching, there is usually an assumption that there exists a simple unique solution at the "correct" model. They therefore neglect the inherent non-uniqueness of the solution of the underlying inverse problem. This, consequently, leads to the assumption that a good history-matched model is a good representation of the reservoir and therefore gives a good forecast (Carter et al. 2004). One of the aims of this study is to challenge these assumptions. Some of the results show the existence of multiple solutions for some set values of objective functions, but produce different results at the prediction phase.

ACKNOWLEDGEMENTS

I would like to thank Dr Jonathan Carter for initiating this project and for all the assistance provided during the cause of this study.
Special thanks to Fahad Dilib (Ph D student in the Petroleum Engineering Dept. Imperial College London) for his precious help and his contributions to the success of the project.

Finally to my family, many thanks for the enduring love.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Imperial College
London

Construction of the Midge History Matching Model

Student name: Lukman Wahab

Imperial College supervisor: Dr. Jonathan Carter

Abstract

This paper tackles the inverse problem of History Matching (HM) on a simplified version of a real reservoir model (Midge Reservoir Model donated by BP) that has been used to make reservoir management decisions. The ability to make correct reservoir management decisions relies mainly on being able to predict their consequences, which in turn depends on the achieved accuracy in the description of the reservoir's internal structure (Gringarten, 1998). This reservoir model is parameterised in 18 regions or compartments, each having homogeneous properties. This parameterisation approach is intended to reduce the number of model parameters. These parameters are mainly the porosities and permeability in the various compartments, the transmissibility across the faults and the allocation of water to the injection wells (Carter et al. 2006).

Different realisations of the Midge model were generated by searching the parameter space in a systematic way using a set reservoir parameters ranges (first varying single parameters in each region – univariate cases, then varying two different parameters for each region- bivariate cases). Synthetic production parameters (Field Oil Production Rate – FOPR, Field Water Production Rate – FWPR, Field Gas Production Rate – FGPR) were then generated from each of the models; these were compared with the synthetic production data from the true model which is the base case. The optimization technique used was to find the minimum of an objective function that best represents the quality of the model. For multivariate parameters variations, a hill climbing algorithm was used to search for all possible 'best models'.

In most of the studies, independent of the method used for the history matching, there is usually an assumption that there exists a simple unique solution at the "correct" model. One of the conclusions of this study is that HM produce multiple 'best' solutions and by comparing the forecast from these 'best' solutions, we can reduce the number of possible solutions. And some of the results show the existence of multiple solutions for some set values of objective functions, but produce different results at the prediction phase.

Introduction

History Matching (HM) is used in reservoir modelling for estimating unknown reservoir properties such as porosity and permeability from measured data. This is usually an inverse problem with non-unique solutions. Some of the synthetic production data used for this analysis are pressures (e.g. Bottom Hole Pressure – BHP, Well Head Pressures – WHP) and rate data (e.g. Field Oil Production Rate – FOPR, Field Water Production Rate – FWPR, Field Gas Production Rate – FGPR, Field Injection Rate – FIJR) at the field level or at each of the producing wells (e.g. Well Oil Production Rate – WOPR, Well Water Producing Rate – WWPR). In other to manage the reservoir optimally, the heterogeneity of the reservoir must be properly characterized, thereby improving the predictive capability of the reservoir model.

HM is a difficult inverse problem and the challenge is to find a model that closely produces the performance of the actual reservoir. This can be carried out either manually by trial and error (usually for small number of parameters) or automatically by a simulator (usually for more than 10000 parameter values). The automatic method is usually time-saving and varies the model parameters until a set matching criteria is achieved. Since the flow equations involved in the simulation are too complex, they cannot be analytically solved in general and therefore the reservoir model behaviour is computed with a flow simulator integrating solvers for these equations. The ECLIPSE flow simulator is used for the analysis. The key objectives of the study are:

1. Decide the parameters that will be considered as uncertain for the history matching problem
2. Write a computer program the will read from a file the values of the uncertain parameters, create the corresponding eclipse data set, run the model and finally extract the data that is used in the history match
3. Write an optimisation code to search for a good quality history match and search for multiple history match solutions.
4. Show, if possible, that any two high quality solutions cannot be joined in parameter space by a path that only includes other high quality solutions.

Comparing this study to the IC Fault Model Approach (Carter et al. 2005)

The simple model used in (Carter et al. 2005) has a 2D cross section of a reservoir with a sequence of alternating good and poor quality sands and a simple fault. The three unknown parameters were high and low permeability of the sands, and the throw of the fault. Carter (2006) attempted to capture the effects of modelling errors in the inverse problems using Bayesian statistics. He then used a similar cross-sectional model of a reservoir to test the results. It appeared that the proposed error model yields a multi modal objective function that leads to multiple acceptable solutions. The approach used (Carter et al. 2005) was to generate a large number of models, which could be searched to find the best match according to the criteria chosen.

For this study, a more complex reservoir model (36 x 35 x 20 cells, parameterised in 18 regions, each having homogeneous properties, 4 producer wells and 2 injection wells, 30yr production parameters generated - first 4000 days production used for the HM problem while production from 4000 to 10000 days was used for the prediction phase) was used but with similar approach to the one used by Carter (2006) to generate a large number of realizations of the reservoir. We then compared all these realizations with our base model and use the sum of squares as the measure of fit between the performance of the base case and each realization. By obtaining both the best production and the best parameter-matched models, we then study the history matching problem. The result shows that a good fit for the production data at the HM phase does not necessarily give the best match at the prediction phase. The best models may in fact give a bad forecast.

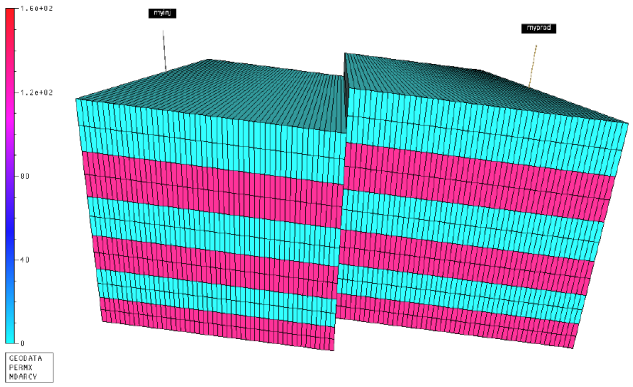

Fig. 1: IC Fault Model with fault throw of h, green and pink represent poor and good quality sands (Kathrada, 2009).

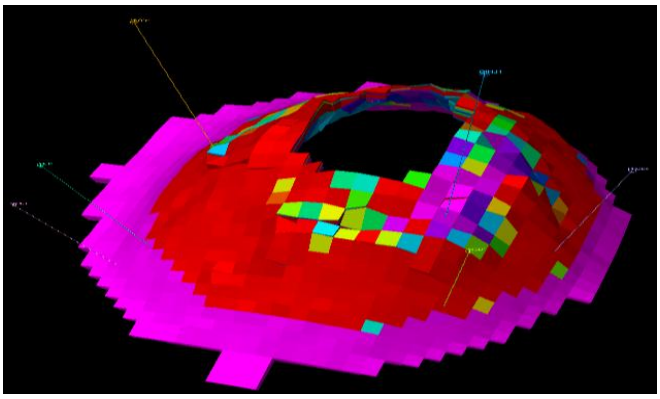

Fig. 2: Midge Model with 25200 cells, 18 compartments, with a total of 72 free model parameters.

Objective Function and Data

To measure the extent to which a reservoir model compares to the available data, an objective function was defined quantifying the misfit between the model output and the measurements at each well. The history covers 360 months of production and contains 1440 data points. These data are used to control the volumes of produced and injected fluid in the model. There are 6 drilled wells (4 producers and 2 injectors).The objective function has been formulated as a combination of the sum-of-squares error on production rates (Oil, Water & Gas) measured synthetic data from 'true' model and compared with rate generated from over 300 'similar' models.

General History Matching Approaches

From a mathematical standpoint, the history matching process reduces to an optimisation problem for which a large number of numerical algorithms are available (Kathrada, 2009). Generally, optimisation algorithms are from two distinct classes:-

1. Techniques that use derivatives like Levenberg-Marquardt and Quasi-Newton (Kathrada, 2009). They have relatively fast convergence but are capable of only finding local minima.
2. Techniques that do not use derivative information like genetic algorithms and particle swarms. They are slower to converge since they search a wider area of the parameter space but are capable of finding multiple minima. They also lend themselves to distributed processing and treating the simulator as a black box (Kathrada, 2009).

The first method was selected for this study because it is adequate and shows an ensemble of 'best' solutions around the global optimal of the both the univariate and multivariate function used.

Deviation between simulated and observed data is given by;

$$D = \sum_{k=0}^{n} (S_i - O_i)^2 \qquad (1)$$

Root Mean Square Error (RMSE)

$$\mathrm{RMSE} = \frac{\sum_{k=0}^{n}(\mathrm{w_s})(\mathrm{S_i} - \mathrm{O_i})^2}{\sum_{k=0}^{n}\mathrm{w_s}} \quad (2)$$

Ws is weighs for each data set

Likelihood Function / Probability (P)

$$\mathrm{P} = \frac{\exp(-\sum_{k=0}^{n}(\mathrm{w_s})(\mathrm{S_i}-\mathrm{O_i})^2)}{\mathrm{Constant}} \quad (3)$$

Multivariate Normal Distribution Function (MVNPDF)

The probability density function of the *d*-dimensional multivariate normal distribution is given by

$$\mathrm{MVNPDF} = \mathrm{f}(\mathrm{x}, \mu, \Sigma) = \frac{1}{\sqrt{|\Sigma|(2\pi)^{\mathrm{d}}}} \mathrm{e}^{-\frac{1}{2}(\mathrm{x}-\mu)\Sigma^{-1}(\mathrm{x}-\mu)'} \quad (4)$$

The multivariate normal distribution is a generalization of the univariate normal to two or more variables. It is a distribution for random vectors of correlated variables, each element of which has a univariate normal distribution. In the simplest case, there is no correlation among variables, and elements of the vectors are independent univariate normal random variables.
The multivariate normal distribution is parameterized with a mean vector, μ, and a covariance matrix, Σ. These are analogous to the mean μ and variance σ^2 parameters of a univariate normal distribution. The diagonal elements of Σ contain the variances for each variable, while the off-diagonal elements of Σ contain the covariances between variables.
The multivariate normal distribution is often used as a model for multivariate data, primarily because it is one of the few multivariate distributions that are tractable to work with (Mathworks statistical tool-box, 2010).

Local Search Algorithm: The Hill Climbing (HC) algorithm is an optimization technique which belongs to the family of local search. It is relatively simple to implement and can be used to solve problems that have many solutions, some of which are better than others. It starts with a random (potentially poor) solution, and iteratively makes small changes to the solution, each time improving it a little. When the algorithm cannot see any improvement anymore, it terminates. HC attempts to maximize (or minimize) a function *f*(*x*), where *x* are discrete states. Hill climbing will follow the graph from vertex to vertex, always locally increasing (or decreasing) the value of *f*, until a local maximum (or local minimum) *xm* is reached (Rusell and Norvig, 2010 - Wikipedia).

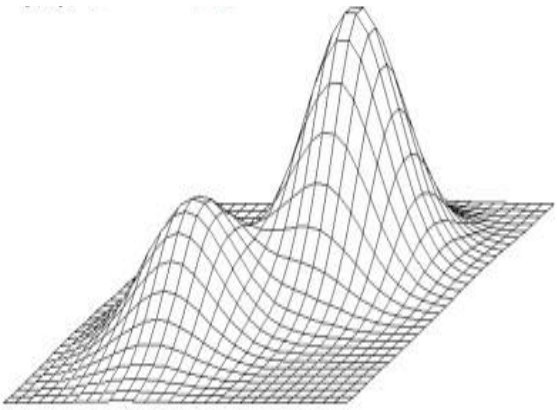

Fig. 3: Multivariate normal probability density function

(Rusell and Norvig, 2010 - Wikipedia)

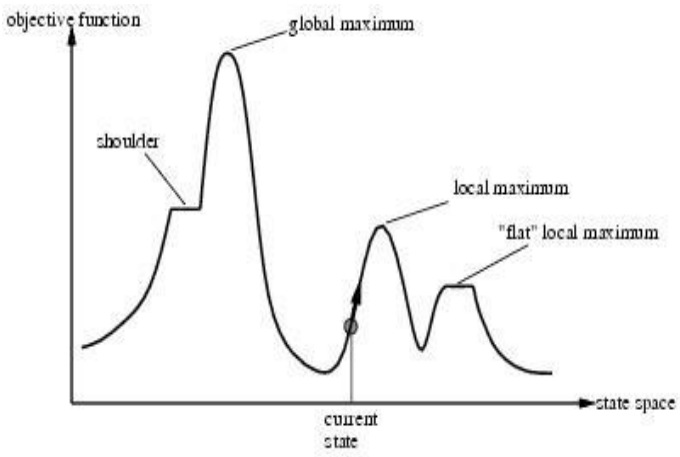

Fig. 4: Simple Hill Climbing Algorithm (Rusell and Norvig, 2010 - Wikipedia)

Simulations Results & Analysis

For each model generated, a full field simulation was run and synthetics production data were generated from each well and at the field scale. The complete simulation results can be found in Appendix D. To simplify the analysis process, a few cases have been selected for review.

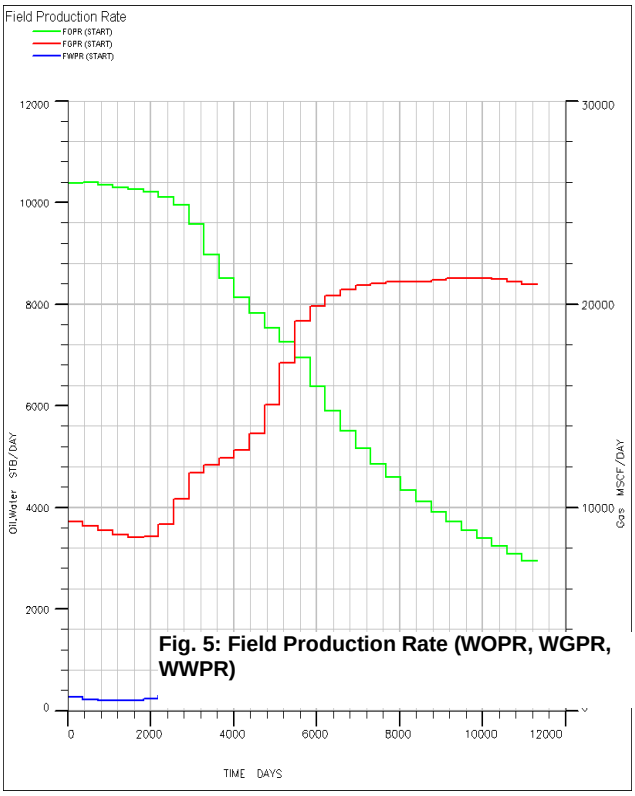

Fig. 5: Field Production Rate (WOPR, WGPR, WWPR)

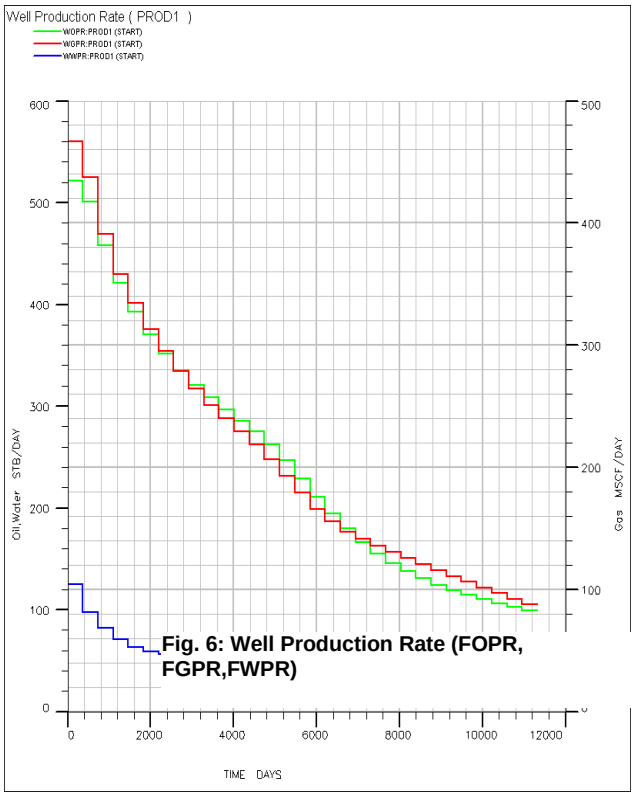

Fig. 6: Well Production Rate (FOPR, FGPR,FWPR)

Fig. 5 & 6 above shows the production parameters that were used in the HM analysis at the field and well scales respectively. These data were generated for each model using the univariate and multivariate approach discussed earlier. And Table 1 below shows a typical production parameters generated for the true model (START) and a univariate model (START1) generated by varying the PORO in region 2 of the true model.

Table 1: Field Production Rate (FOPR, FGPR and FWPR) for the 'true model' and 'trial model'

TIME (days)	FOPR (stb/day)	FOPR (stb/day)	FWPR (stb/day)	FWPR (stb/day)	FGPR (mscf/day)	FGPR (mscf/day)
'START'	'START'	'START1'	'START'	'START1'	'START'	'START1'
365	10382	10026	266	206	9293	8757
730	10390	10069	210	150	9069	8852
1096	10345	10072	196	137	8864	8818

Fig. 7-10 shows the RMSE obtained by comparing the production parameters (e.g. FOPR) obtained from each model generated using the univariate approach with the true model. Fig. 7 for example shows that the best parameter-match models are models 8-12 because they meet the criteria of RSME less than 0.15 set to measure the quality of the match. And Fig. 10 shows the best parameter-match models are models 3-7.

Cases – univariate

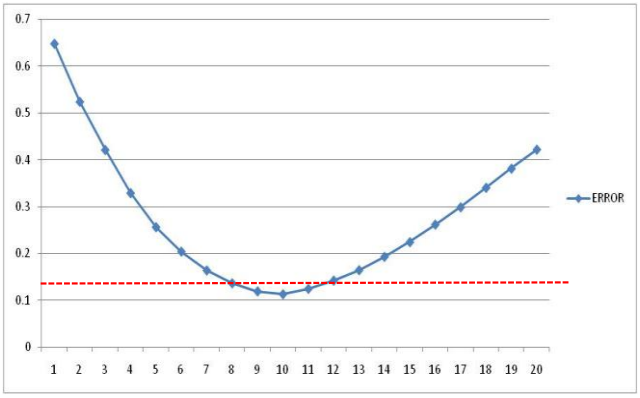

Fig. 7: RMSE obtained by varying the PORO [0.1:0.01:0.3] in region 2 and comparing the FOPR for different realizations with the 'true' case

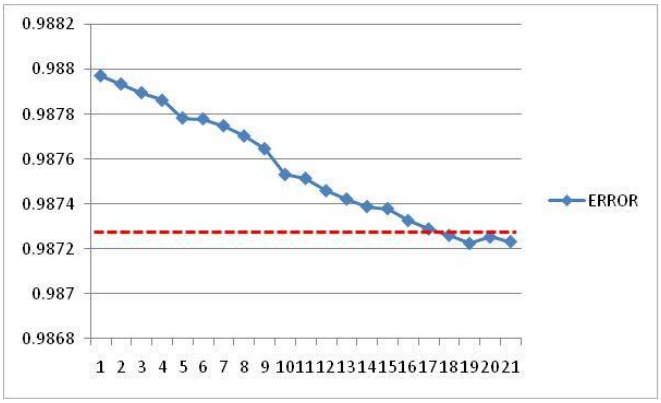

Fig. 8: RMSE obtained by varying the PERMX [1:0.1:40] in region 2 and comparing the FWPR for different realizations with the 'true' case

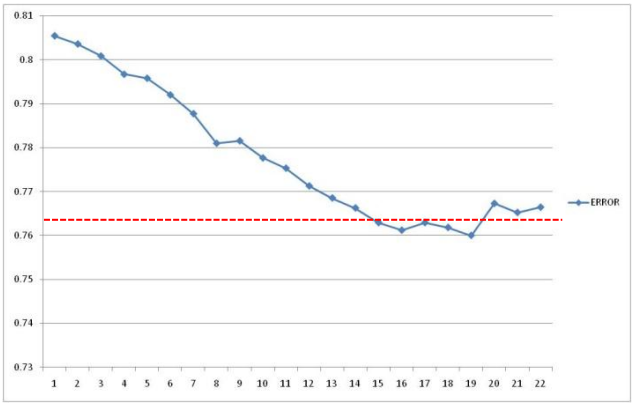

Fig. 9: RMSE obtained by varying the PERMY [1:0.1:40] in region 4 and comparing the FOPR for different realizations with the 'true' case

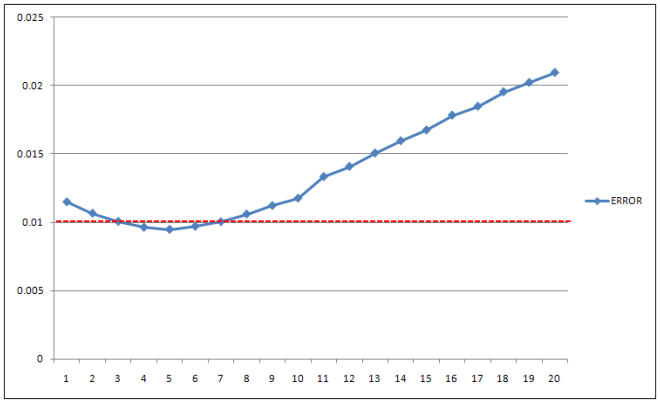

Fig. 10: RMSE obtained by varying the PORO [0.1:0.01:0.3] in region 4 and comparing the FGPR for different realizations with the 'true' case

Figs. 11, 12, 14, 15, 17, and 18 show the MVNPDF plotted for different cases obtained by varying multiple parameters in different regions. And Fig. 13, 16 & 19 shows the results of the HC algorithm identifying the best models for each case investigated. Fig. 16 was obtained by varying both PORO [0.1:0.01:0.3] and PERMX [1:0.1:40] in region 4 and comparing the FOPR for different realizations with the 'true' case. The result shows that the best models for this case are 12, 15, 16, 17 & 19.

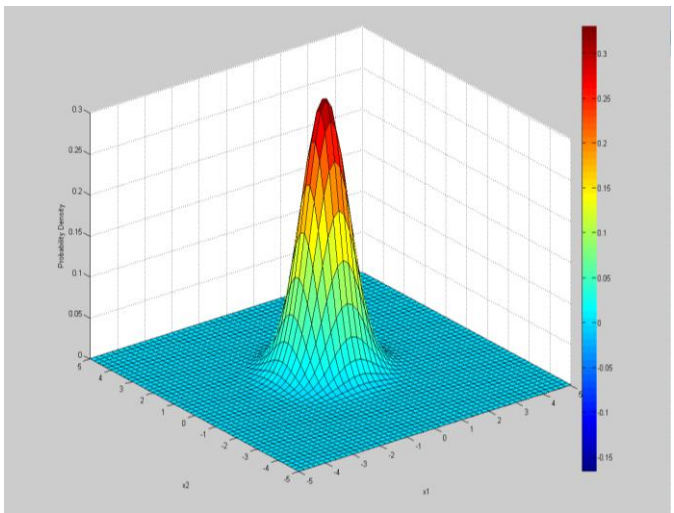

Fig. 3: – Multivariate Normal Probability Density Function obtained by varying both PORO [0.1:0.01:0.3] and PERMX [1:0.1:40] in region 2 and comparing the FOPR for different realizations with the 'true' case

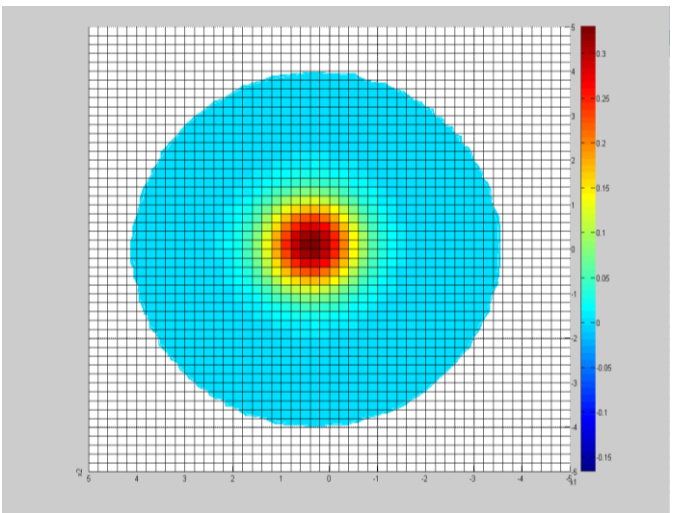

Fig. 12: Contour Map for Multivariate Normal Probability Density Function obtained by varying both PORO [0.1:0.01:0.3] and PERMX [1:0.1:40] in region 2 and comparing the FOPR for different realizations with the 'true' case

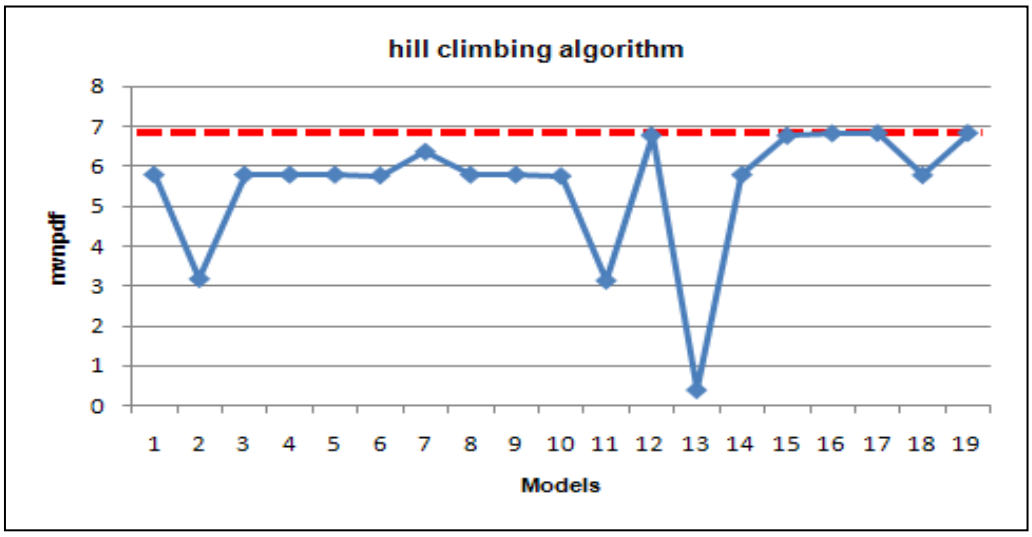

Fig. 4: 10 best models were identified around peak of the Multivariate Normal Probability Density Function (MVNPDF) plot shown on Fig. 11

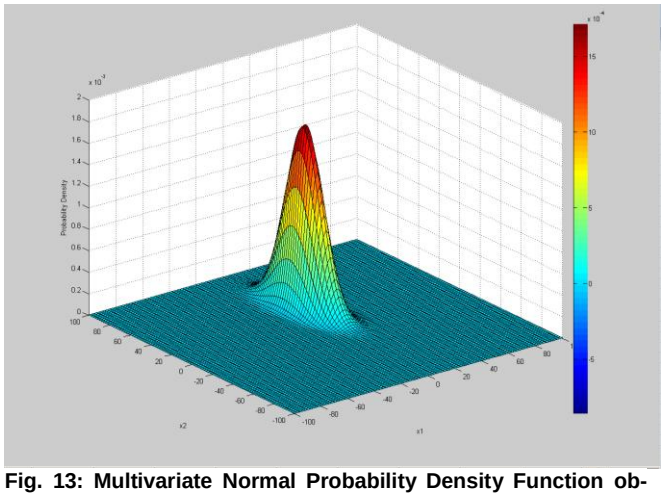

Fig. 13: Multivariate Normal Probability Density Function obtained by varying both PORO [0.1:0.01:0.3] and PERMX [1:0.1:40] in region 4 and comparing the FOPR for different realizations with the 'true' case

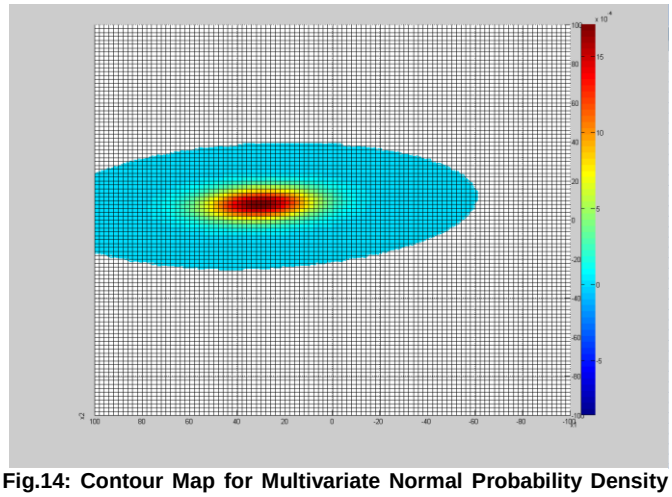

Fig.14: Contour Map for Multivariate Normal Probability Density Function obtained by varying both PORO [0.1:0.01:0.3] and PERMX [1:0.1:40] in region 4 and comparing the FOPR for different realizations with the 'true' case

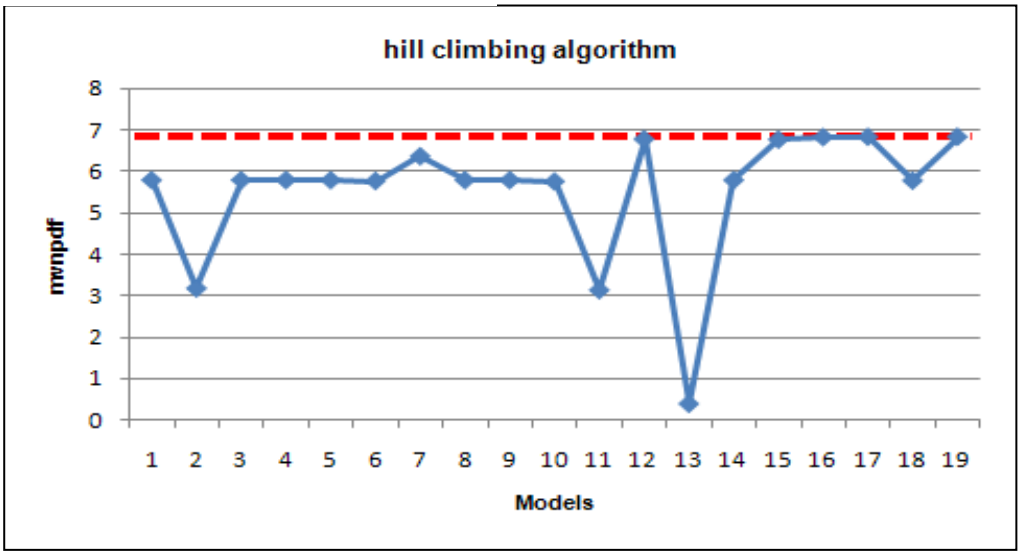

Fig. 5: 5 best models were identified around peak of the Multivariate Normal Probability Density Function (MVNPDF) plot

shown on Fig. 14

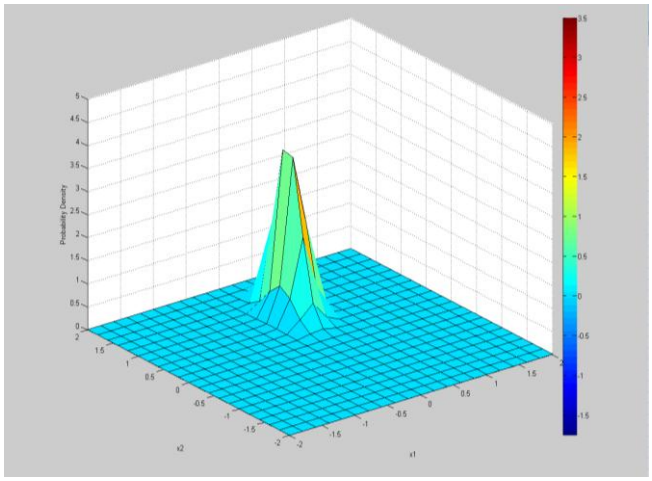

Fig. 7: MVNPDF obtained by varying both PORO [0.1:0.01:0.3] and PERMY [1:0.1:40] in region 4 and comparing the FOPR for different realizations with the 'true' case

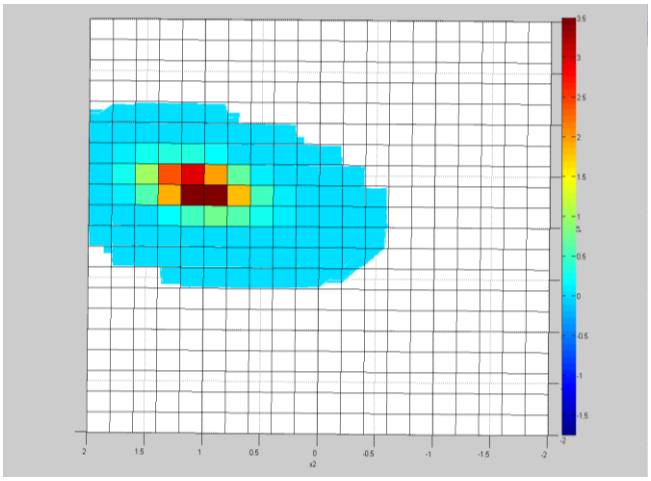

Fig. 6: Contour Map for MVNPDF obtained by varying both PORO [0.1:0.01:0.3] and PERMY [1:0.1:40] in region 4 and comparing the FOPR for different realizations with the 'true' case

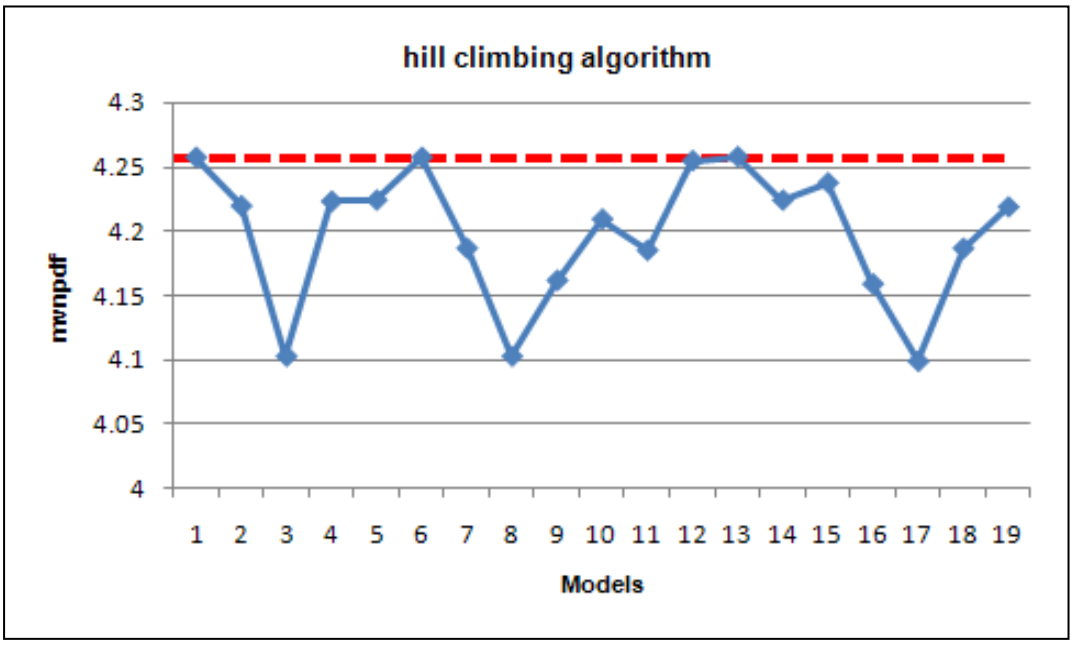

Fig. 8: 4 best models were identified around peak of the Multivariate Normal Probability Density Function (MVNPDF) plot shown on Fig. 14

Forecast from the Best Models

The best parameter-match and production-match models were identified from the results discussed above. These models will test further at the prediction phase by comparing the forecasts from these wells with the true model. Production data obtained up to 4000 days of production were used for the HM analysis. While the production data obtained after this period were assessed during the prediction phase. Fig. 20 and Fig. 21 shows the FWPR and FOPR predictions respectively from 3 best models – START6, START12 & START13 and compares these with the prediction from the true model (START).

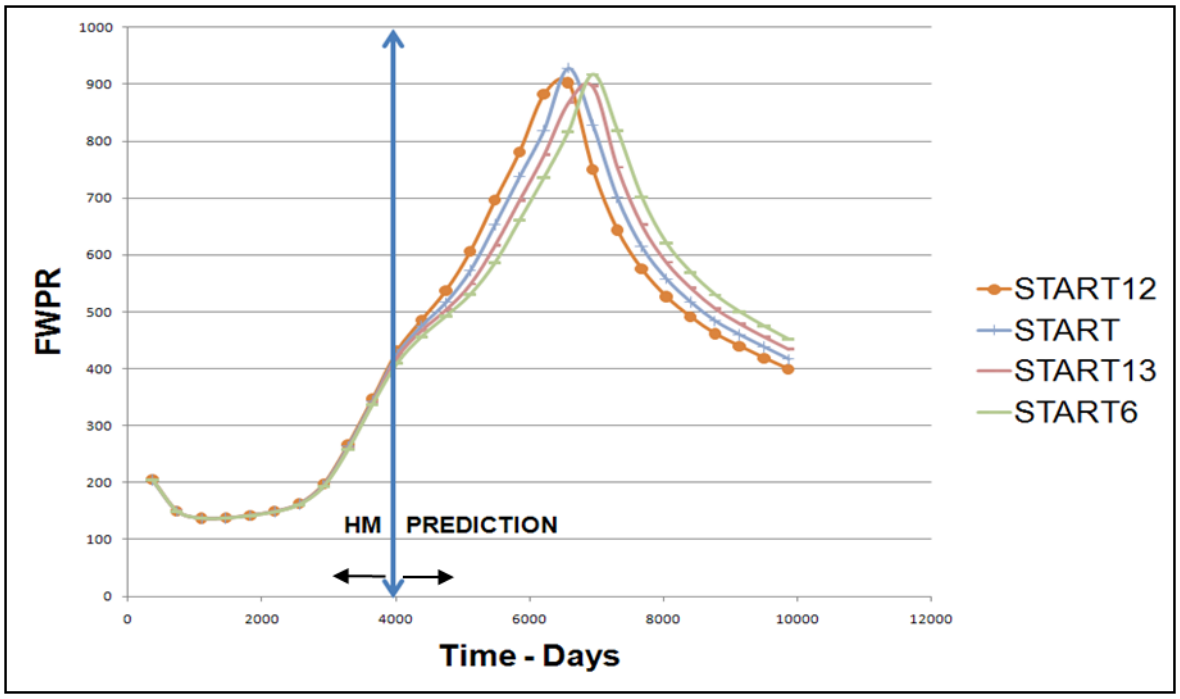

Fig. 9: Plot showing the FWPR predictions from 3 best models and the true model for the case shown in Fig. 18

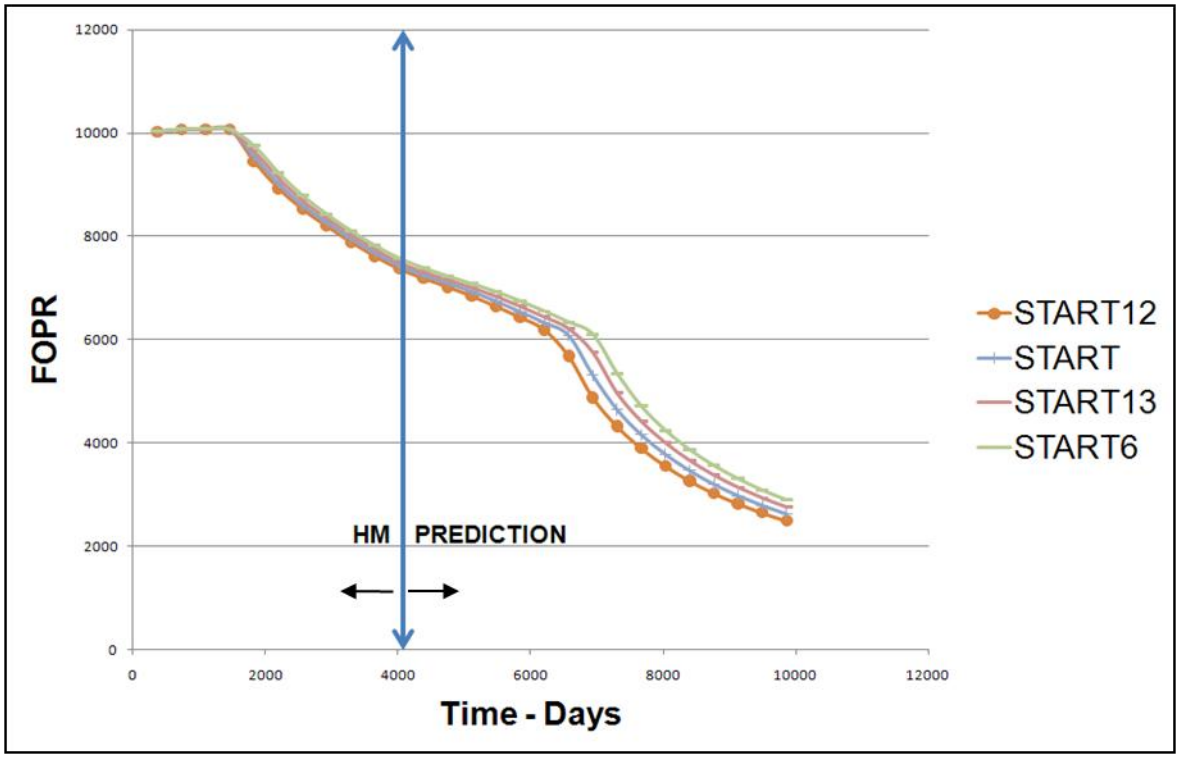

Fig. 10: Plot showing the FOPR predictions from 3 best models and the true model for the case shown in Fig. 18

Table 2: Shows the best parameters Models described in Fig. 20 & 21: 6, 12 & 13

Note: Four parameters were considered unknown (PORO, PERMX, PERMY & PERMZ) for each region, making a total of 72 free model parameters each having a lower and upper bound. Only two parameters were varied for a particular region

	REGION 4	REGION 4
START	PORO=0.2	PERMY=40
START01	0.1	40
START02	0.11	38
START03	0.12	36
START04	0.13	34
START05	0.14	32
START06	0.15	30
START07	0.16	28
START08	0.17	26
START09	0.18	24
START10	0.19	22
START11	0.2	20
START12	0.21	19
START13	0.22	18
START14	0.23	17
START15	0.24	16
START16	0.25	15
START17	0.26	14
START18	0.27	13
START19	0.28	12
START20	0.29	11
START21	0.3	10

Fig. 21 and Fig. 22 also compare the Oil saturations (So) predictions from two of the best models – START12 and START13.

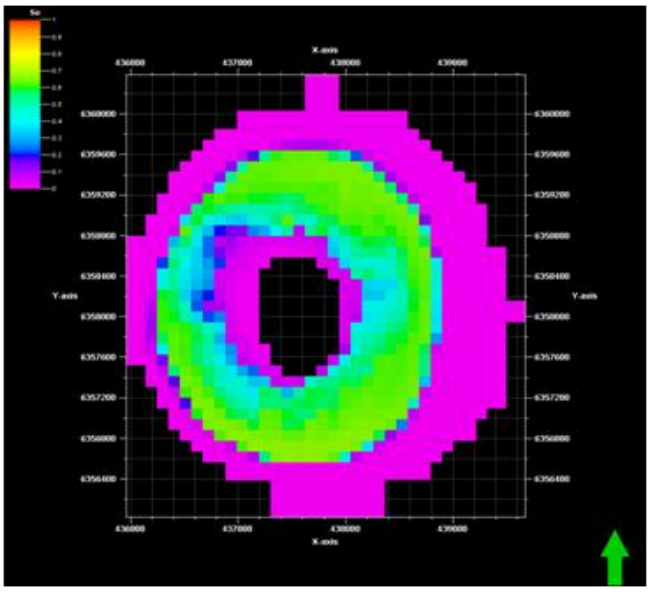

Fig. 21: Oil Saturation (So) plots at the end of the prediction period for 'START12' model

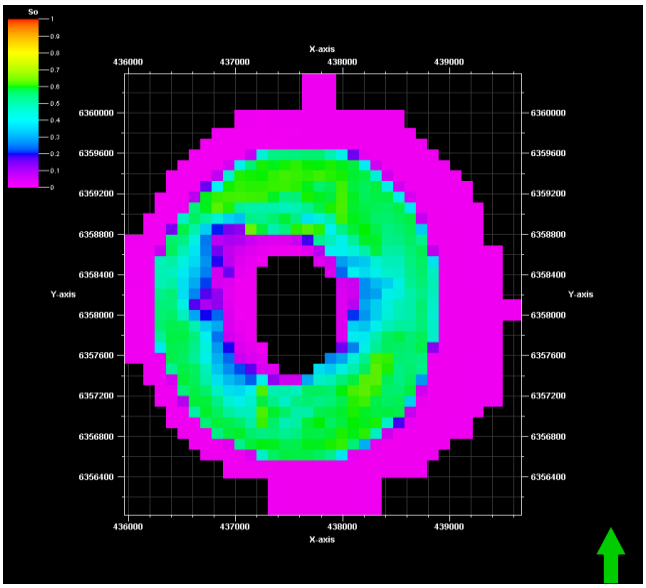

Fig. 22: Oil Saturation (So) plots at the end of the prediction period for'START13' model

Discussion

This study used both the univariate and multivariate analysis technique to search the parameter space of a simplified model (BP Midge Reservoir Model) for all possible distinctly different solutions in a systematic way, thus creating an ensemble of all possible history match solutions. For the univariate analysis a RSME value was set by visual inspection of the matches and most of the results show more than one best solution for different cases examined. For the multivariate analysis, we searched the MVNPDF plot using a simple hill climbing algorithm for all solutions around the maximum points (points with minima error between the true case and the trials cases, the results show a set of solutions around single minima. Multiple 'best' solutions were located around the top of the hill.

The ensemble of 'best' solutions was carried forward for the prediction phase (Fig 19 & 20) to test the forecast from these models against the true model. The productions parameters up to 4000 days were used for the HM analysis while the data beyond this point was used for the prediction phase. To compare the predictions, Field Oil Production Rate, Field Water Production Rate and Oil Saturations were examined. And the result shows that even though these three models (6, 12 & 13) in Fig. 19 & 20 gave a 'perfect' match during the HM stage, their predictions were different.
Oil Saturation (So) plots shown in Fig. 21 & 22 for Models 12 & 13 also indicate different So predictions for different regions.

Conclusions

The Bayesian method is usually adopted with an objective function which is assumed to have a single 'best' model. This study used a simplified model donated by BP (Midge Reservoir Model) that is parameterized into 18 regions each having homogeneous properties to show that this is may not be the case. Four parameters were considered unknown (PORO, PERMX, PERMY & PERMZ) for each region, making a total of 72 free model parameters each having a lower and upper bound. For each of the regions we generated different realisations by systematically searching the parameter space and ran simulations for all these realizations. Table 2 shows the parameters of the best models that were considered.

Using a weighted sum of squares for the objective function, we found the best production and parameter-matched models. The optimization technique used was to find the minimum of an objective function that best represents the quality of the model. For multivariate parameters variations, a hill climbing algorithm was used to search for all possible 'best models'. From the results, we can deduce the followings:

- The HM produce multiple 'best' solutions which can be carried forward into the forecasting stage
- By comparing the forecast from the 'best', we can reduce the number of possible solutions
- The result further support the claim that the HM solutions are not unique
- The HM exercise requires a good optimisation algorithm and detailed result processing identify all possible 'best models'
- This study only identified solutions that are joined in parameter space by a path that only includes other high quality solutions

Suggestions for further work

- Run a simulated annealing algorithm and compare results obtained from the simple hill climbing algorithm
- Implement the Particle Swarm Optimisation (PSO) to search a wider area of the parameter space for the existence of multiple minima
- Use the original complex model supplied by BP to solve the same HM problem and compare the results

Nomenclature

Σ	=	Covariance matrix
K_x	=	PERMX: Horizontal permeability (x-direction)
K_y	=	PERMY: Horizontal permeability (y-direction)
K_z	=	PERMZ: Vertical permeability
μ	=	Mean Vector
M_n	=	Number of models corresponding to the level n
N	=	Number of observed data
OF	=	Objective function
O_i	=	Observed data
Ø	=	PORO: Porosity
RMSE	=	Root Mean Square Error between observed and simulated data
S	=	Number of data series in the objective function composition
S_i	=	Simulated data
σ^2	=	variance
So	=	Oil saturations
MVNPDF	=	Multivariate Normal Probability Distribution Function
MVCPDF	=	Multivariate Cumulative Probability Distribution Function

References

1. J.N. Carter, P.J. Ballester, Z. Tavassoli and P.R. King: ‘‘Our Calibrated Model has No Predictive Value: An Example from the Petroleum Industry’’, Reliability Engineering and System Safety 91 (2006) 1373–1381
2. J.N. Carter, Z. Tavassoli and P.R. King: ‘‘Errors in History Matching’’, September 2004 SPE Journal (SPE 86883)
3. J.N. Carter and Bush, M.D.: “Application of a modified genetic algorithm to parameter estimation in the petroleum industry,” paper presented at the 1996 Conference on Artificial Neural Networks in Engineering, St. Louis, Missouri, 1–13 November
4. Yamada, T.: “Nonuniqueness of History Matching,” paper SPE 59434 presented at the 2000 SPE Asia Pacific Conference on Integrated Modeling for Asset Management, Yokohama, Japan, 25–26 April.
5. J.N. Carter, Tavassoli, Z. and King, P.R.: ‘‘An analysis of history matching errors’’, Computational Geosciences (2005)
6. J.N. Carter, C.E. Romero, A.C. Gringarten, and R.W. Zimmerman: ‘‘A Modified Genetic Algorithm for Reservoir Characterisation’’, paper SPE 64765. Proceedings of SPE International Oil and Gas Conference and Exhibition in China held in Beijing, China, 7–10 November 2000.
7. Muhammad Kathrada: ‘‘Uncertainty Evaluation of Reservoir Simulation Models using Particle Swarms and Hierarchical Clustering’’, PhD thesis, Institute of Petroleum Engineering, Heriot-Watt University, June 2009
8. Fanchi, John R.: ‘‘Principles of Applied Reservoir Simulation (3rd Edition)’’, 2006 Elsevier
9. S. A. Ghoniem, S. Abdel Aliem, M. H. Sayyouh and M. El Salaly: ‘‘A simplified method for petroleum reservoir history matching’’, Department of Petroleum Engineering, Cairo University, Cairo, Egypt (Received January 1983)
10. ZHU Yan, XIE Jin-zhuang, YANG Wei-hua and HOU Lian-hua: ‘‘Method for improving history matching precision of reservoir numerical simulation’’, PETROL. EXPLOR. DEVELOP., 2008, 35(2): 225–229.
11. Paulo Camargo Silva, Célio Maschio and Denis J. Schiozer: ‘‘Use of Neuro-Simulation techniques as proxies to reservoir simulator: Application in production history matching’’, Journal of Petroleum Science and Engineering 57 (2007) 273–280
12. Richard Baker: ‘‘Streamline Technology: Reservoir History Matching and Forecasting = Its Success, Limitations, and Future’’, Journal of Canadian Petroleum Technology, April 2001, Volume 40, No. 4
13. Jens-Petter Nørgård, MEPO, Scandpower Petroleum Technology AS: ‘‘Revolutionising History Matching and Uncertainty Assessment’’, GEO ExPro September 2006
14. Alexandre Castellini, Chevron ETC; Arman Vahedi, Updesh Singh, Ramzy Shenouda Sawiris and Thomas Roach, Chevron Australia Pty. Ltd.: ‘‘Reconciling History Matching and Assessment of Uncertainty in Production Forecasts: A Study Combining Experimental Design, Proxy Models and Genetic Algorithms’’, Presentation at the International Petroleum Technology Conference held in Kuala Lumpur, Malaysia, 3-5 December 2008, IPTC 12745.
15. F.J.T. Floris, M.D. Bush, M. Cuypers, F. Roggero and A.R. Syversveen: ‘‘Methods for quantifying the uncertainty of production forecast: a comparative study’’, Petroleum Geoscience, Vol. 7 2001, pp. S87-S96
16. L.C. Reis, L.E. dos Reis, L.C. da Silva, GG. Becerra, Petrobras: ‘‘History Matching: Is it Necessary to Optimize?’’, Presentation at the 2009 SPE Latin American and Carribean Petroleum Engineering Conference held in Cartagena, Colombia, 31 May – 3 June 2009, SPE 122826
17. Mike Christie, Vasily Demyanov and Demet Erbas: ‘‘Uncertainty quantification for porous media flows’’, Journal of Computational Physics 217 (2006) 143-158
18. S. Subbey, M. Christie, M. Sambridge: ‘‘Prediction under uncertainty in reservoir modeling’’, Journal of Petroleum Science and Engineering 44 (2004) 143– 153
19. Oswaldo Velez-Langs: ‘‘Genetic algorithms in oil industry: An overview’’, Journal of Petroleum Science and Engineering 47 (2005) 15– 22
20. P.J. Ballester and J.N. Carter: ‘‘Real-parameter Genetic Algorithms for Finding Multiple Optimal Solutions in Multimodal Optimisation’’, Genetic and Evolutionary Computation Conference 2003:706, Lecture Notes in Computer Science 2723, Pub Springer-Verlag, Heidelberg,2003.
21. M.D. Bush and J.N. Carter: ‘‘Applications of a Modifed Genetic Algorithm to Parameter Estimation in the Petroleum Industry’’, Intelligent Engineering Systems through Artificial Neural Networks 6:397, 1996.
22. The MathWorks: ‘‘http://www.mathworks.es/access/helpdesk/help/toolbox/stats/’’ statistical tool-box, 1994-2010.
23. Rusell and Norvig: ‘‘Artificial Intelligence: A Modern Approach’’, Wikipedia, 2010: ‘‘The hill climbing algorithm’’– Third Edition (Modified: Jan 13, 2010)

APPENDICES

APPENDIX A: CRITICAL LITERATURE REVIEW

MILESTONES IN HISTORY MATCHING STUDY

Source	Paper n°	Year	Title	Authors	Contribution
SPE	SPE 64765	2000	A Modified Genetic Algorithm for Reservoir Characterisation	C.E. Romero,J.N. Carter, A.C. Gringarten, and R.W. Zimmerman, SPE Members, Imperial College, London, UK	This work discusses a functional history matching approach where an optimization process is no longer necessary
SPE	86883	2004	Errors in History Matching	Z. Tavassoli, Jonathan N. Carter, and Peter R. King, Imperial College, London	The results show that a good fit for the production data does not necessarily have a good estimation for the parameters of the reservoir, and therefore it may lead to a bad forecast for the performance of the reservoir
Journal of Petroleum Science and Engineering	44 (2004) 143– 153	2004	Prediction under uncertainty in reservoir modelling	S. Subbeya, M. Christiea, M. Sambridgeb	This paper presents a new approach for generating uncertain reservoir performance predictions and quantifying the uncertainty associated with forecasting future performance
Journal of Petroleum Science and Engineering	47 (2005) 15–22	2005	Genetic algorithms in oil industry: An overview	Oswaldo Velez-Langs	The study presented here is directed to accumulate the body of knowledge which is up to now built around the techniques of Evolutionary Computation in the Oil Industry, particularly in the Exploration and Production business.
Journal of Computational Physics	217 (2006) 143–158	2006	Uncertainty quantification for porous media flows	Mike Christie , Vasily Demyanov, Demet Erbas	The approach was demonstrated on a simple three parameter sampling problem that had proved a difficult problem in previous studies. Objective of the paper: To quantify the uncertainties involved in predicting flows of oil and water through oil reservoirs
Journal of Petroleum Science and Engineering	59 (2007) 157–168	2007	A parallel real-coded genetic algorithm for history matching and its application to a real petroleum reservoir	Pedro J. Ballester, Jonathan N. Carter	A new methodology using Real-coded Genetic Algorithm (GA) was presented to tackle History Matching problems
International Petroleum Technology Conference	IPTC 12745	2008	Reconciling History matching and assessment of Uncertainty in production forecasts: A study combining experimental design, proxy models and genetic algorithms.	Alexandre Castellini, Chevron ETC; Arman Vahedi, Updesh Singh, Ramzy Shenouda sawiris and Thomas Roach, Chevron Australia Pty. Ltd.	This paper addresses the limitations of conventional techniques and provides a practical, structured workflow to reconcile the processes of data integration and uncertainty assessment.
SPE	SPE 122826	2009	History Matching: Is it Necessary to Optimize?	L.C. Reis, SPE, L.E. dos Reis, LC. da Silva, GG. Becerra, Petrobras	This work discusses a functional history matching approach where an optimization process is no longer necessary

Paper 1:

SPE 64765
A Modified Genetic Algorithm for Reservoir Characterisation

Authors: C.E. Romero,J.N. Carter, A.C. Gringarten, and R.W. Zimmerman, SPE Members, Imperial College, London, UK

Contribution to the understanding of the Midge HM model:
This work discusses a functional history matching approach where an optimization process is no longer necessary

Objective of the paper:
This paper describes the implementation of a Genetic Algorithm (GA) to carry out hydrocarbon reservoir characterisation by conditioning the reservoir simulation model to production data (history matching) on a predefined geological and structural model.

Methodology used:
Genetic Algorithms are a feasible technique for generating reservoir descriptions using production data. The method is capable of handling many parameters, which is critical when dealing with large full-field reservoir simulation models.

Conclusion reached:
This paper has presented results on the application of a modified Genetic Algorithm to a realistic, synthetic model, with respect to main issues of its formulation. This paper describes in detail the formulation of a modified GA using non-standard genome and genetic operators. The method is computationally efficient, in that it requires only a modest number of forward simulations.

Comments:
The proposed technique combines the advantages of the pilot point method for the description of petro physical properties, with the advantages of GAs for global optimisation.

Paper 2:

Paper (SPE 86883) peer approved 28 May 2004.
Errors in History Matching

Authors: Z. Tavassoli, Jonathan N. Carter, and Peter R. King, Imperial College, London

Contribution to the understanding of the Midge HM model:
The results show that a good fit for the production data does not necessarily have a good estimation for the parameters of the reservoir, and therefore it may lead to a bad forecast for the performance of the reservoir.

Objective of the paper:
To shows the errors in the common procedure in history matching where we adopt a Bayesian approach with an objective function that is assumed to have a single simple minimum at the "correct" model.

Methodology used:
Generated a large number of realizations of the reservoir and choose one of them as a base case. Using the weighted sum of squares for the objective function, the best production- and best parameter-matched models were found.

Conclusion reached:
In summary, all the results seem to suggest that in using the conventional history-matching methods, one cannot practically guarantee to recover the true model, which represents the real geological structure of the reservoir.

Comments:
This paper discusses the idea that the "true" model (base case model) is not necessarily the most likely to be obtained using conventional history-matching methods.

Paper 3:

Journal of Petroleum Science and Engineering 44 (2004) 143– 153
Prediction under uncertainty in reservoir modelling

Authors: S. Subbeya, M. Christiea, M. Sambridgeb

Contribution to the understanding of the Midge HM model:
This paper presents a new approach for generating uncertain reservoir performance predictions and quantifying the uncertainty associated with forecasting future performance.

Objective of the paper:
Usually, a single history-matched model, conditioned to production data, is obtained. The model is then used to forecast future production profiles. Because the history match is non-unique, the forecast production profiles are therefore uncertain, although this uncertainty is not usually quantified. This paper aim to demonstrate a methodology to quantify these uncertainties

Methodology used:
Firstly, multiple reservoir realizations are generated using a new stochastic algorithm. This involves adaptively sampling the model parameter space using an algorithm, which biases the sampling towards regions of good fit. Using the complete ensemble of models generated, the posterior distribution is resampled in order to quantify the uncertainty associated with forecasting reservoir performance in a Bayesian framework. The strength of the method in performance prediction is demonstrated by using an upscaled model to history match fine scale data. The maximum likelihood model is then used in forecasting the fine grid performance, and the uncertainty associated with the predictions is quantified.

Conclusion reached:
This paper has demonstrated an approach for generating uncertain history-matching models, and quantifying uncertainty in model performance prediction using the Neighbourhood Approximation algorithm. The approach is able to incorporate both data and model errors in quantifying the degree of model fit to the observed data, and in defining the model likelihood.

Comments:
To quantify uncertainty in model predictions, the Neighbourhood Approximation algorithm was employed in a Bayesian framework and that the true solution lies within the uncertainty bounds predicted by the algorithm.

Paper 4:

Journal of Petroleum Science and Engineering 47 (2005) 15– 22
Genetic algorithms in oil industry: An overview

Authors: Oswaldo Velez-Langs

Contribution to the understanding of the Midge HM model:
The study presented here is directed to accumulate the body of knowledge which is up to now built around the techniques of Evolutionary Computation in the Oil Industry, particularly in the Exploration and Production business.

Objective of the paper:
The techniques aim at the incorporation into the reservoir models of all data available so that more realistic models can be generated for improved prediction capabilities.

Methodology used:
Intelligent techniques such as neural computing, fuzzy reasoning, and evolutionary computing for data analysis and interpretation are an increasingly powerful tool for making breakthroughs in the science and engineering fields by transforming the data into information and information into knowledge. The process makes use of measurements made on the field to restrict the range of values that the parameters might take.

Conclusion reached:
The work described here presents the state of the art in engineering applications from a point of view of the computational theory of the adaptation and the evolution for applications in Oil industry.

Comments:
The use of this technique offers a true benefit in exploration and production business.

Paper 5:

Journal of Computational Physics 217 (2006) 143–158
Uncertainty quantification for porous media flows

Authors: Mike Christie , Vasily Demyanov, Demet Erbas

Contribution to the understanding of the Midge HM model:
The approach was demonstrated on a simple three parameter sampling problem that had proved a difficult problem in previous studies.
Objective of the paper: To quantify the uncertainties involved in predicting flows of oil and water through oil reservoirs

Methodology used:
This paper examines a Bayesian Framework for uncertainty quantification in porous media flows that uses a stochastic sampling algorithm to generate models that match observed data. Machine learning algorithms are used to speed up the identification of regions in parameter space where good matches to observed data can be found.

Conclusion reached:
The best results were obtained using direct prediction of misfit with a trained multi-layer perceptron. This method provided an estimate of uncertainty that was very close to the estimate provide using a steady state genetic algorithm which involved a significantly higher number of forward model evaluations.

Comments:
By using the neural network to guide sampling within the context of a stochastic search algorithm, and running the expensive forward model principally in regions of good fit, they were able to easily generate a significant number of models that match history well.

Paper 6:

Journal of Petroleum Science and Engineering 59 (2007) 157–168
A parallel real-coded genetic algorithm for history matching and its application to a real petroleum reservoir

Authors: Pedro J. Ballester, Jonathan N. Carter

Contribution to the understanding of the Midge HM model:
A new methodology using Real-coded Genetic Algorithm (GA) was presented to tackle History Matching problems

Objective of the paper:
Adjust the model parameter values until an optimal between simulated and measured history is achieved

Methodology used:
Real-coded Genetic Algorithm (GA)

Conclusion reached:
An improvement in the quality of result and HM model variability. While different models can be generated the study shows the importance of applying optimisation methods capable of identifying all possible realizations.

Comments:
In other to save computation time, different solutions generated from the GA were evaluated in parallel on different computers. The best solutions were analysed using clustering algorithm

Paper 7:

International Petroleum Technology Conference - IPTC 12745
Reconciling History matching and assessment of Uncertainty in production forecasts: A study combining experimental design, proxy models and genetic algorithms.

Authors: Alexandre Castellini, Chevron ETC; Arman Vahedi, Updesh Singh, Ramzy Shenouda sawiris and Thomas Roach, Chevron Australia Pty. Ltd.

Contribution to the understanding of the Midge HM model:
This paper addresses the limitations of conventional techniques and provides a practical, structured workflow to reconcile the processes of data integration and uncertainty assessment.

Objective of the paper:
To efficiently find combinations of parameters that minimize the objective function.
Methodology used: In order to reduce the number of actual simulations and to accelerate the overall procedure, non-linear response surfaces, built with kriging interpolants at each iteration of the optimization routine, filter out unnecessary combinations of parameters. The models that reasonably honour the historical data are selected via cluster analysis techniques and provide an estimate of future production. The final distribution of the prediction variables defines the range of uncertainty conditioned to production history.

Conclusion reached:
The strategy ensures multiple and significantly different history-matched models that provide estimation of the future performance of the reservoir.

Comments:
The paper presents a method to tackle complex inverse problems where highly non-linear responses are involved.

Paper 8:

SPE 122826
History Matching: Is it Necessary to Optimize?

Authors: L.C. Reis, SPE, L.E. dos Reis, LC. da Silva, GG. Becerra, Petrobras

Contribution to the understanding of the Midge HM model:
This work discusses a functional history matching approach where an optimization process is no longer necessary

Objective of the paper:
To discuss the history matching process in the way it is being applied in most cases.

Methodology used:
This work discusses the functional history matching approach coupled with uncertainty analysis. Two methodologies were proposed: ''Risk analysis with filtered RSM'' and ''Risk Analysis with Filtered Neural Network''. Both use tolerance criteria for the objective function to select (or filter) the possible models. They differ from each other only in the proxy model used.

Conclusion reached:
The methodology discussed in this work proposes a different approach for history matching.

Comments:
The paper questions the need for optimization in decision making by managers.

APPENDIX B: MIDGE MODEL DATA FILE

INLUDED IN THE CD COPY

APPENDIX C: SYNTETIC PRODUCTION FROM 'TRUE MODEL'

TIME 'START1' (DAYS)	FOPR 'START' (STB/DAY)	FWPR 'START' (STB/DAY)	FGPR 'START' (MSCF/DAY)
0	0	0	0
365	10381.819	266.44409	9293.1709
730	10390.167	210.27602	9069.1699
1096	10345.456	196.45242	8864.3145
1461	10300.184	193.15553	8659.1328
1826	10259.078	199.10921	8506.835
2191	10205.668	225.72102	8571.916
2557	10107.381	302.28046	9177.2188
2922	9943.1621	448.4996	10413.41
3287	9565.5762	594.81335	11703.373
3652	8962.627	703.11243	12076.069
4018	8503.0811	755.54272	12438.756
4383	8133.8262	780.74115	12826.513
4748	7818.6602	807.33795	13618.858
5113	7532.5273	846.92407	15050.591
5479	7256.6606	894.797	17094.832
5844	6940.5083	928.8374	19156.137
6209	6379.9507	878.63544	19906.213
6574	5903.9526	829.91718	20408.561
6940	5506.874	785.65253	20720.65
7305	5160.4976	744.45923	20915.617
7670	4858.2451	706.32977	21021.406
8035	4585.4448	672.00555	21083.297
8401	4332.2319	640.24194	21084.707
8766	4105.4609	611.67664	21102.377
9131	3907.1436	586.4118	21197.061
9496	3722.3494	563.33276	21246.785
9862	3550.3547	542.13538	21277.75

Find attached CD copy containing all other production parameters generated for different cases

APPENDIX D: SIMULATION RESULTS

CASES- BIVARIATE PROBABILITY DISTRIBUTION FUNCTION

CASE1

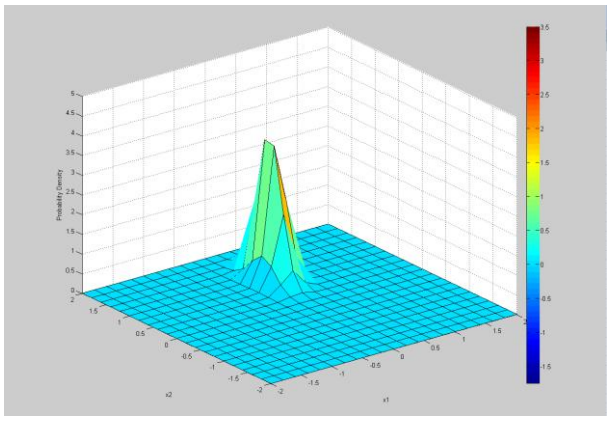

Fig. 11: MVNPDF obtained by varying both PERMX [1:0.1:40] and PERMY [1:0.1:40] in region 4 and comparing the FOPR for different realizations with the 'true' case

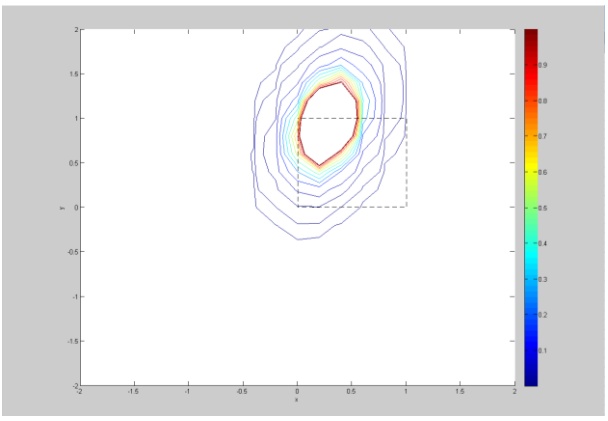

Fig. 12: Contour Map for MVNPDF obtained by varying both PERMX [1:0.1:40] and PERMY [1:0.1:40] in region 4 and comparing the FOPR for different realizations with the 'true' case

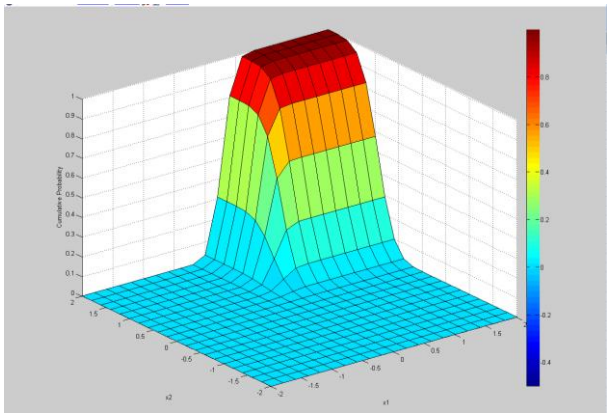

Fig. 13: MVCPDF obtained by varying both PERMX [1:0.1:40] and PERMY [1:0.1:40] in region 4 and comparing the FOPR for different realizations with the 'true' case

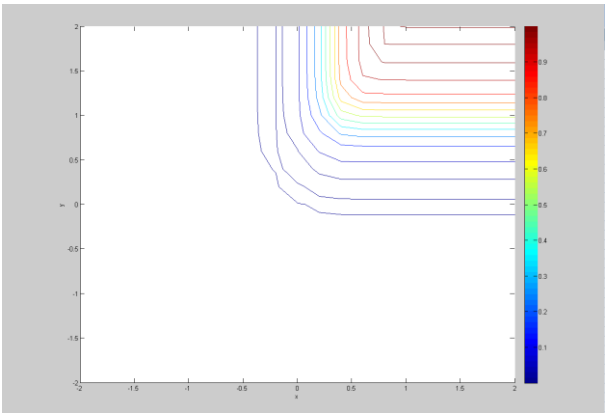

Fig. 14: Contour Map for MVCPDF obtained by varying both PERMX [1:0.1:40] and PERMY [1:0.1:40] in region 4 and comparing the FOPR for different realizations with the 'true' case

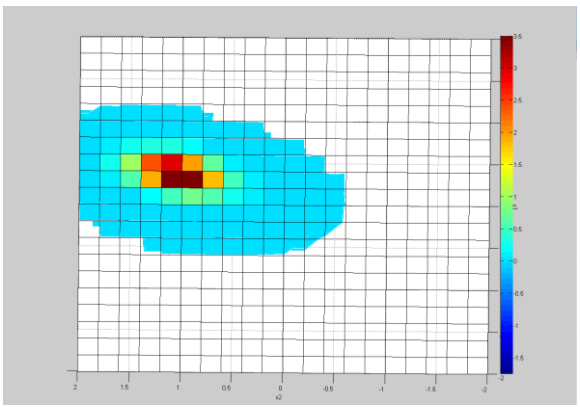

Figure 15: Contour Map for MVNPDF obtained by varying both PERMX [1:0.1:40] and PERMY [1:0.1:40] in region 4 and comparing the FOPR for different realizations with the 'true' case

CASE 2

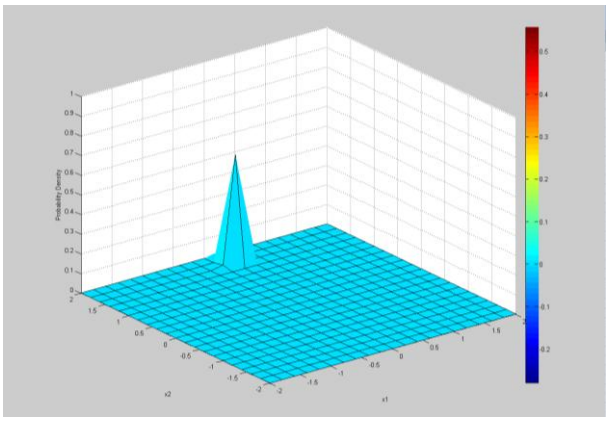

Fig. 16: MVNPDF obtained by varying both PORO [0.1:0.01:0.3] and PERMY [1:0.1:40] in region 3 and comparing the FWPR for different realizations with the 'true' case

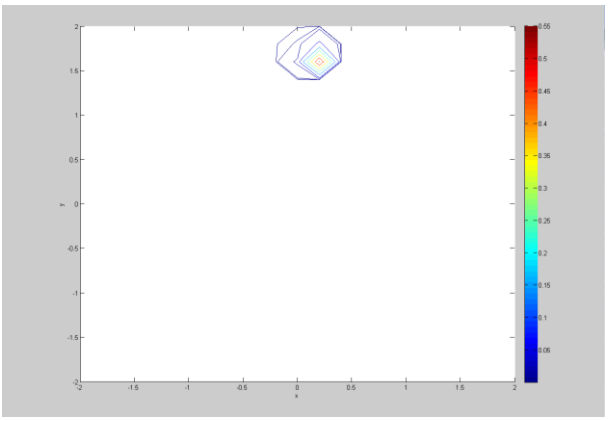

Fig. 17: Contour Map for MVNPDF obtained by varying both PORO [0.1:0.01:0.3] and PERMY [1:0.1:40] in region 3 and comparing the FWPR for different realizations with the 'true' case

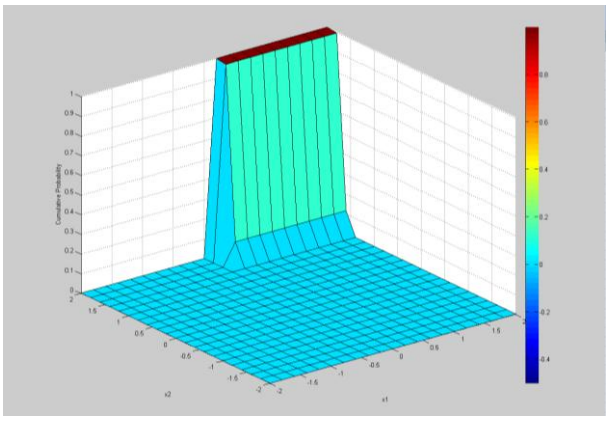

Fig. 18: MVCPDF obtained by varying both PORO [0.1:0.01:0.3] and PERMY [1:0.1:40] in region 3 and comparing the FWPR for different realizations with the 'true' case

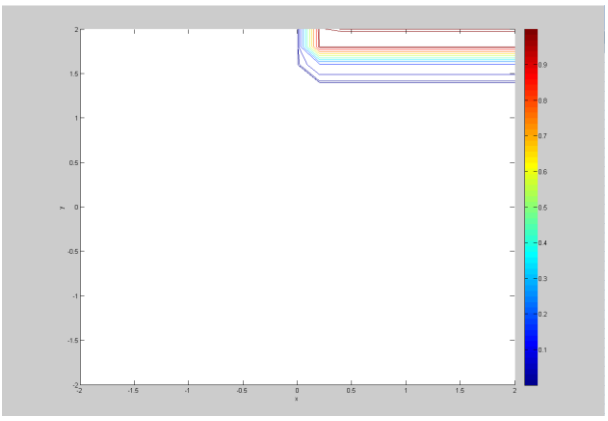

Figure 19: Contour Map for MVCPDF obtained by varying both PORO [0.1:0.01:0.3] and PERMY [1:0.1:40] in region 3 and comparing the FWPR for different realizations with the 'true' case

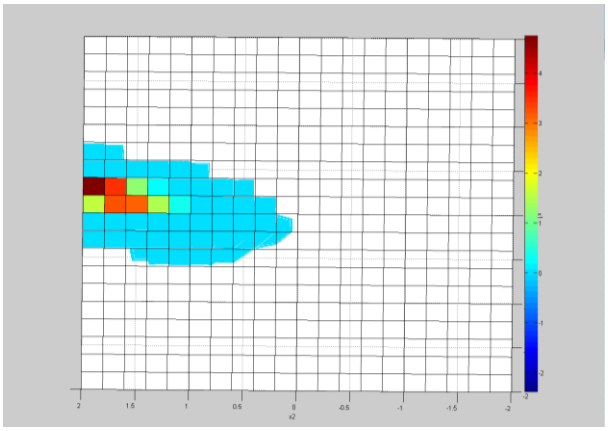

Figure 20: Contour Map for MVNPDF obtained by varying both PORO [0.1:0.01:0.3] and PERMY [1:0.1:40] in region 3 and comparing the FWPR for different realizations with the 'true' case

CASE 3

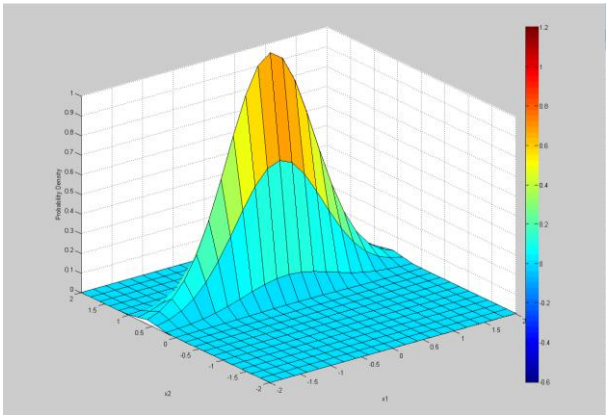

Fig. 21: MVNPDF obtained by varying both PORO [0.1:0.01:0.3] and PERMX [1:0.1:40] in region 5 and comparing the FWPR for different realizations with the 'true' case

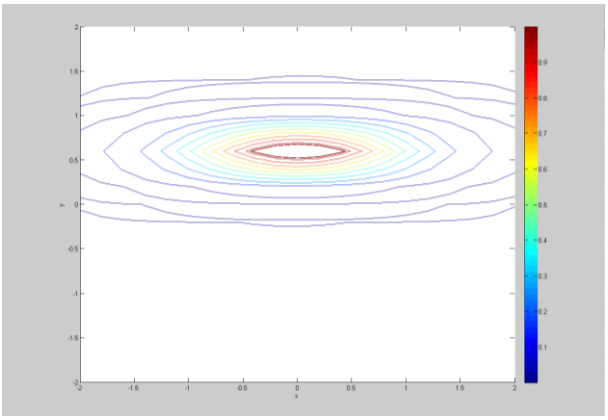

Fig. 22: Contour Map for MVNPDF obtained by varying both PORO [0.1:0.01:0.3] and PERMX [1:0.1:40] in region 5 and comparing the FWPR for different realizations with the 'true' case

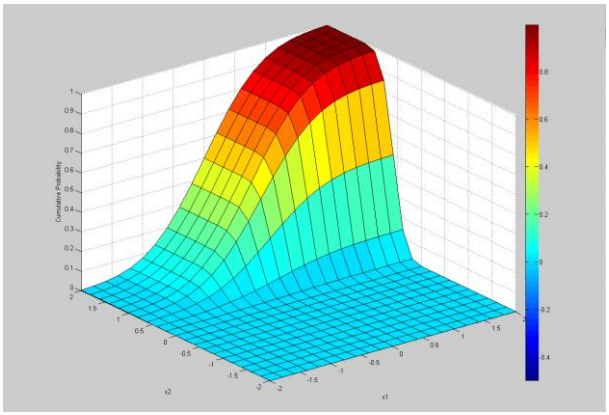

Fig. 23: MVCPDF obtained by varying both PORO [0.1:0.01:0.3] and PERMX [1:0.1:40] in region 5 and comparing the FWPR for different realizations with the 'true' case

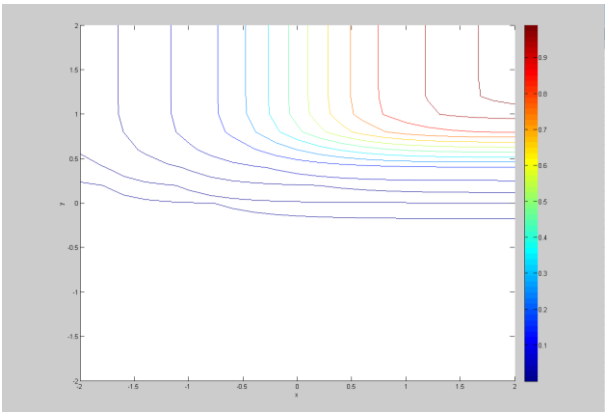

Fig. 24: Contour Map for MVCPDF obtained by varying both PORO [0.1:0.01:0.3] and PERMX [1:0.1:40] in region 5 and comparing the FWPR for different realizations with the 'true' case

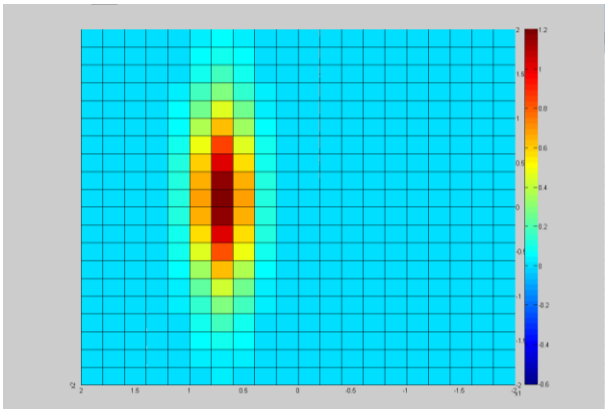

Figure 25: Contour Map for MVNPDF obtained by varying both PORO [0.1:0.01:0.3] and PERMX [1:0.1:40] in region 5 and comparing the FWPR for different realizations with the 'true' case

CASE 4

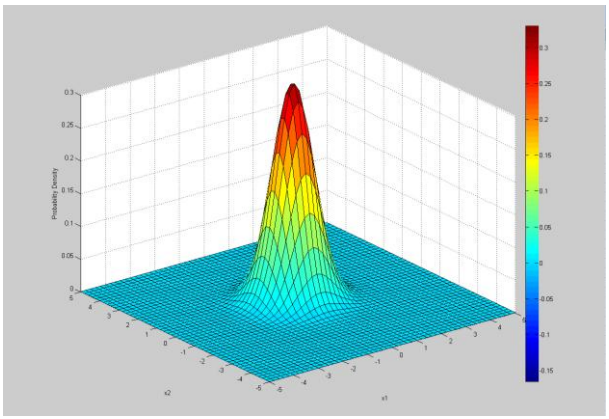

Fig. 26: MVNPDF obtained by varying both PORO [0.1:0.01:0.3] and PERMX [1:0.1:40] in region 2 and comparing the FOPR for different realizations with the 'true' case

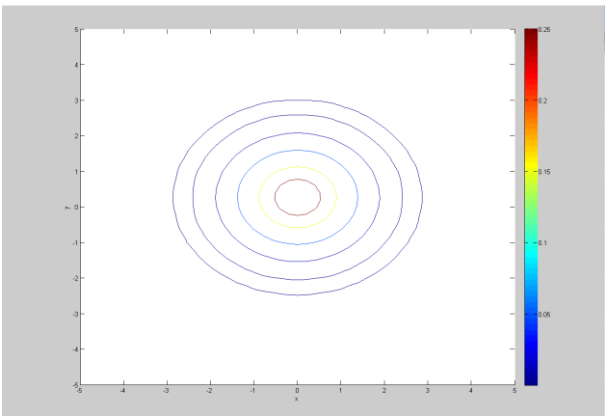

Fig. 27: Contour Map for MVNPDF obtained by varying both PORO [0.1:0.01:0.3] and PERMX [1:0.1:40] in region 2 and comparing the FOPR for different realizations with the 'true' case

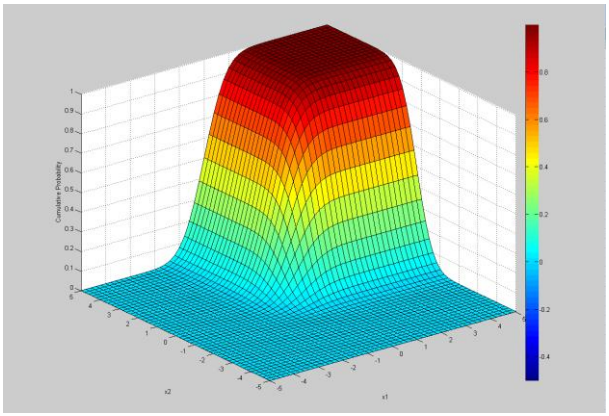

Fig. 28: MVCPDF obtained by varying both PORO [0.1:0.01:0.3] and PERMX [1:0.1:40] in region 2 and comparing the FOPR for different realizations with the 'true' case

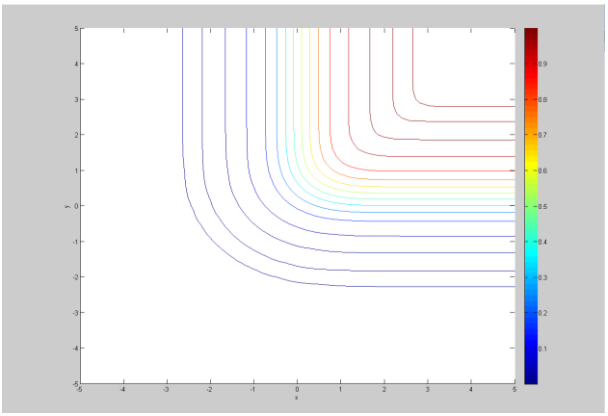

Fig. 29: Contour Map for MVCPDF obtained by varying both PORO [0.1:0.01:0.3] and PERMX [1:0.1:40] in region 2 and comparing the FOPR for different realizations with the 'true' case

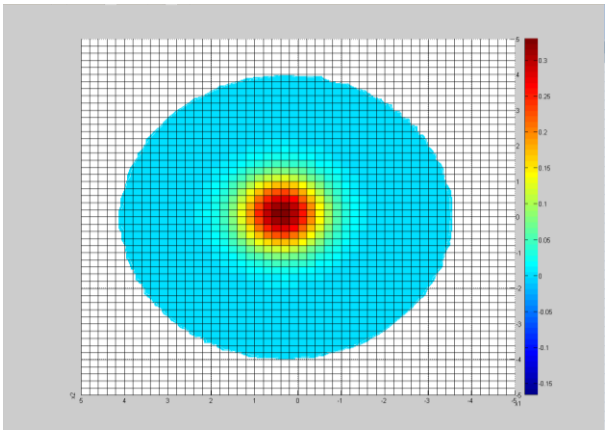

Fig. 30: Contour Map for MVNPDF obtained by varying both PORO [0.1:0.01:0.3] and PERMX [1:0.1:40] in region 2 and comparing the FOPR for different realizations with the 'true' case

CASE 5

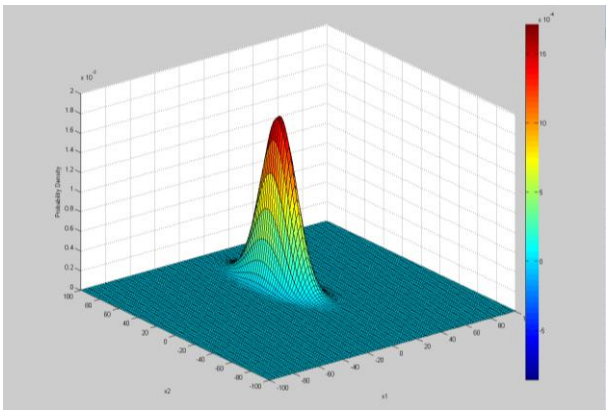

Fig. 31: MVNPDF obtained by varying both PORO [0.1:0.01:0.3] and PERMX [1:0.1:40] in region 4 and comparing the FOPR for different realizations with the 'true' case

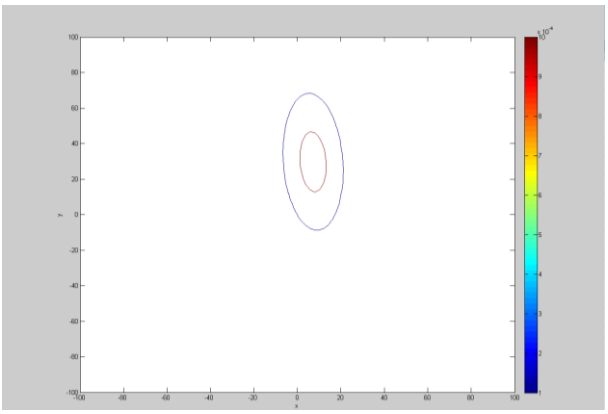

Fig. 32: Contour Map for MVNPDF obtained by varying both PORO [0.1:0.01:0.3] and PERMX [1:0.1:40] in region 4 and comparing the FOPR for different realizations with the 'true' case

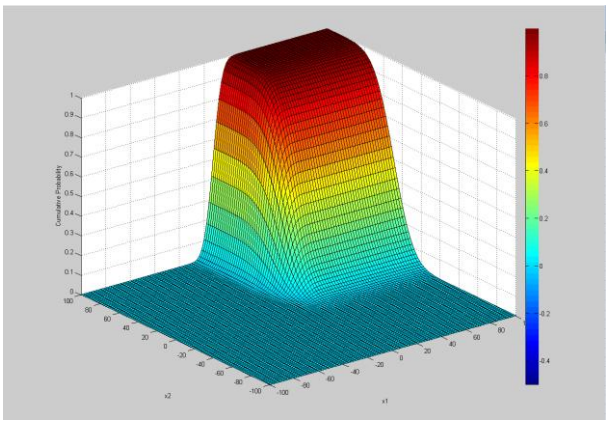

Fig. 33: MVCPDF obtained by varying both PORO [0.1:0.01:0.3] and PERMX [1:0.1:40] in region 4 and comparing the FOPR for different realizations with the 'true' case

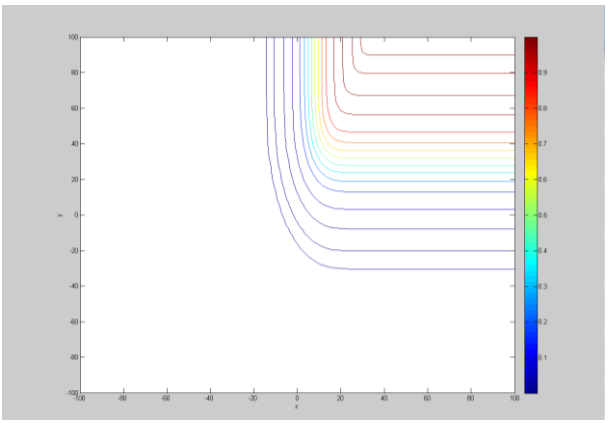

Fig. 34: Contour Map for MVCPDF obtained by varying both PORO [0.1:0.01:0.3] and PERMX [1:0.1:40] in region 4 and comparing the FOPR for different realizations with the 'true' case

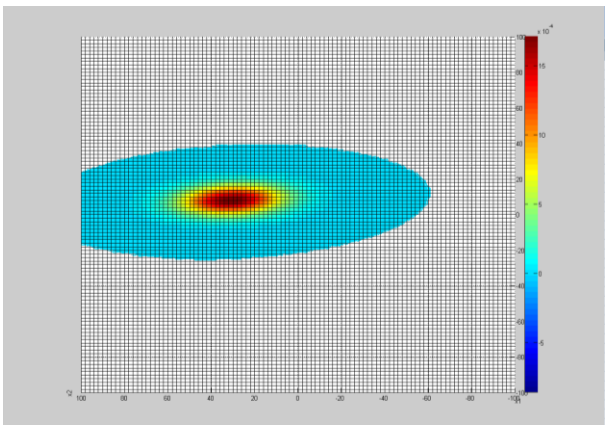

Figure 35: Contour Map for MVNPDF obtained by varying both PORO [0.1:0.01:0.3] and PERMX [1:0.1:40] in region 4 and comparing the FOPR for different realizations with the 'true' case

CASE 6

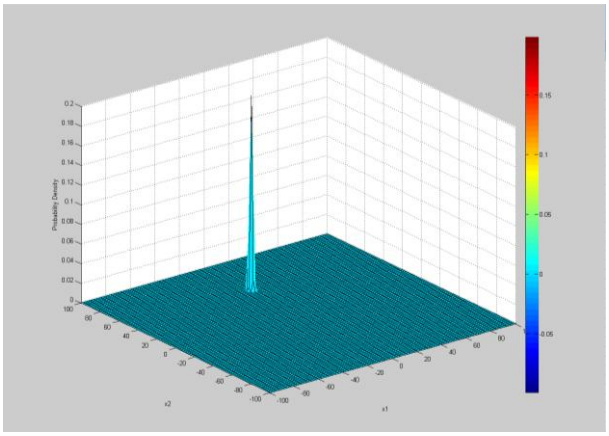

Fig. 36: MVNPDF obtained by varying both PERMZ [0.001:0.0001:0.01] and PERMX [1:0.1:40] in region 6 and comparing the FGPR for different realizations with the 'true' case

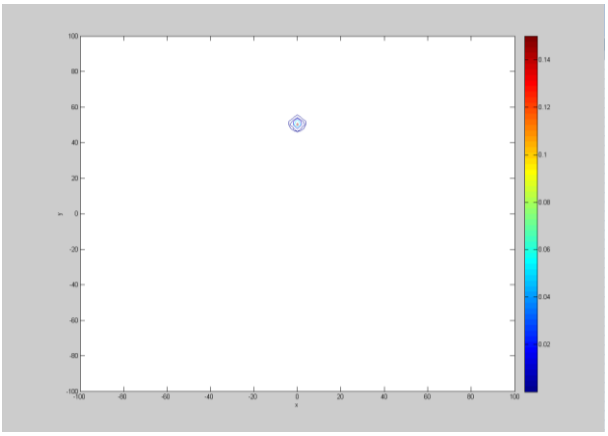

Fig. 37: Contour Map for MVNPDF obtained by varying both PERMZ [0.001:0.0001:0.01] and PERMX [1:0.1:40] in region 6 and comparing the FGPR for different realizations with the 'true' case

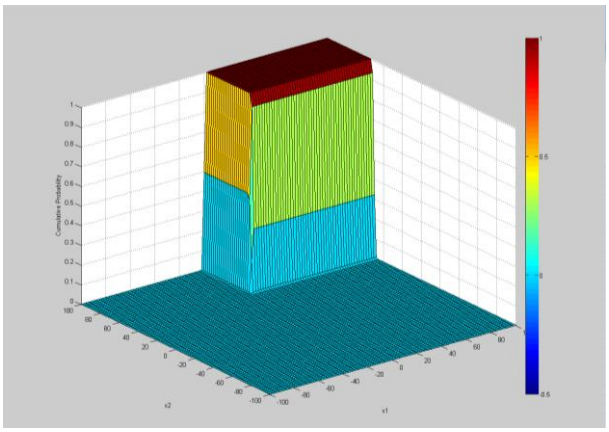

Fig. 38: MVCPDF obtained by varying both PERMZ [0.001:0.0001:0.01] and PERMX [1:0.1:40] in region 6 and comparing the FGPR for different realizations with the 'true' case

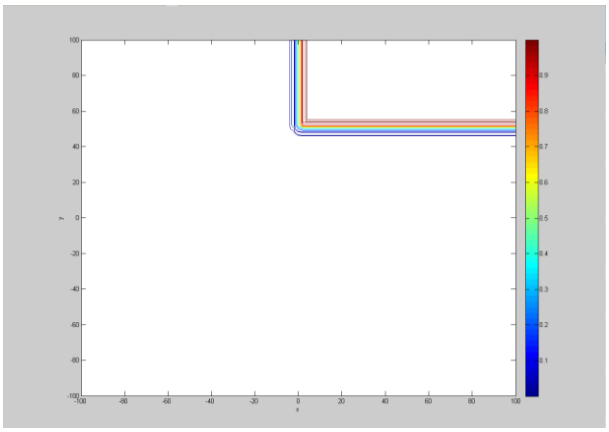

Fig. 39: Contour Map for MVCPDF obtained by varying both PERMZ [0.001:0.0001:0.01] and PERMX [1:0.1:40] in region 6 and comparing the FGPR for different realizations with the 'true' case

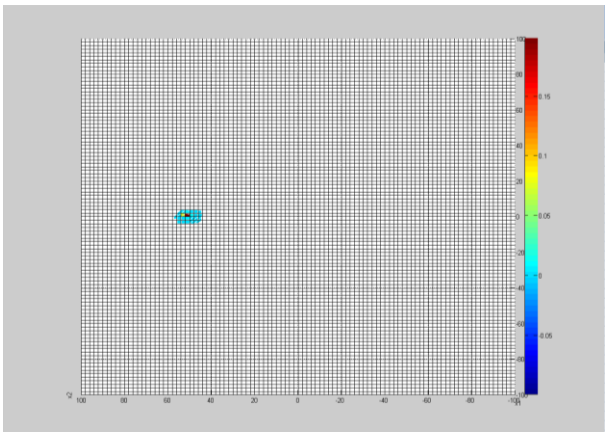

Fig. 40: Figure 47: Contour Map for MVNPDF obtained by varying both PERMZ [0.001:0.0001:0.01] and PERMX [1:0.1:40] in region 6 and comparing the FGPR for different realizations with the 'true' case

CASE 7

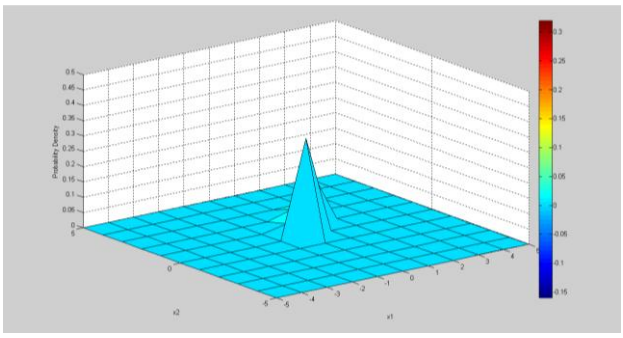

Fig. 41: MVNPDF obtained by varying both PERMZ [0.001:0.0001:0.01] and PERMY [1:0.1:40] in region 6 and comparing the FGPR for different realizations with the 'true' case

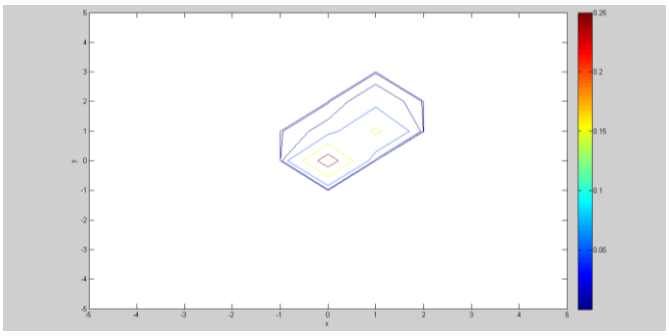

Fig. 42: Contour Map for MVNPDF obtained by varying both PERMZ [0.001:0.0001:0.01] and PERMY [1:0.1:40] in region 6 and comparing the FGPR for different realizations with the 'true' case

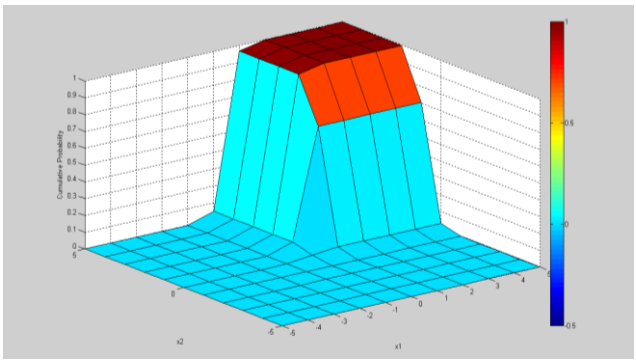

Fig. 43: MVCPDF obtained by varying both PERMZ [0.001:0.0001:0.01] and PERMY [1:0.1:40] in region 6 and comparing the FGPR for different realizations with the 'true' case

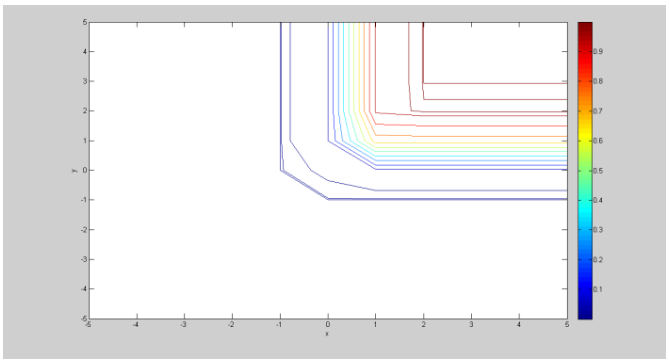

Fig. 44: Contour Map for MVCPDF obtained by varying both PERMZ [0.001:0.0001:0.01] and PERMY [1:0.1:40] in region 6 and comparing the FGPR for different realizations with the 'true' case

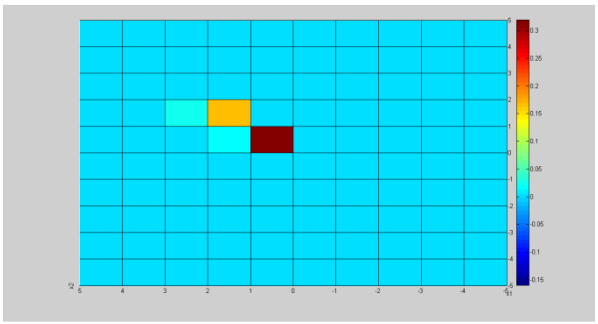

Fig. 45: Contour Map for MVNPDF obtained by varying both PERMZ [0.001:0.0001:0.01] and PERMY [1:0.1:40] in region 6 and comparing the FGPR for different realizations with the 'true' case

CASE 8

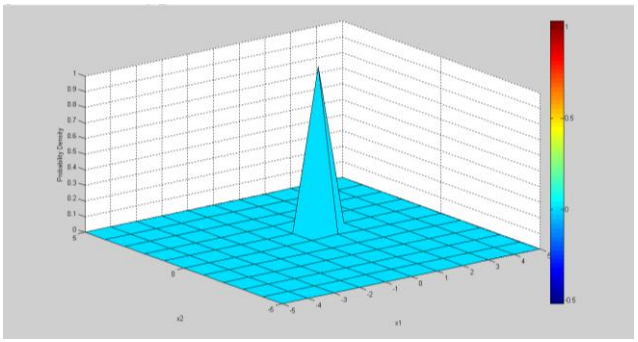

Fig. 46: MVNPDF obtained by varying both PERMZ [0.001:0.0001:0.01] and PERMY [1:0.1:40] in region 7 and comparing the FGPR for different realizations with the 'true' case

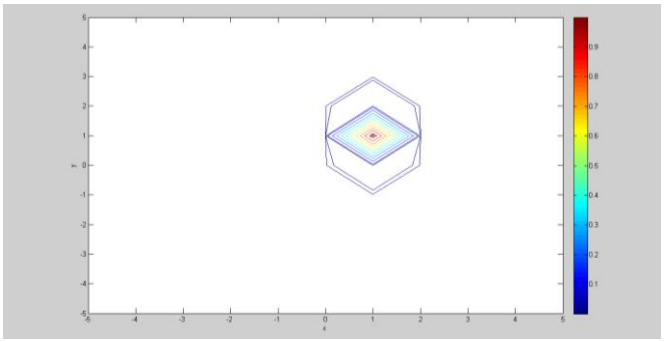

Fig. 47: Contour Map for MVNPDF obtained by varying both PERMZ [0.001:0.0001:0.01] and PERMY [1:0.1:40] in region 7 and comparing the FGPR for different realizations with the 'true' case

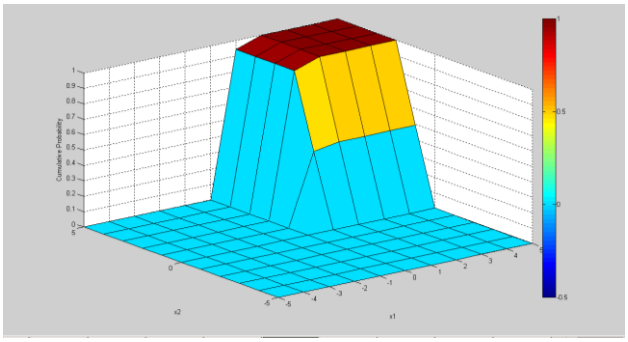

Fig. 48: MVCPDF obtained by varying both PERMZ [0.001:0.0001:0.01] and PERMY [1:0.1:40] in region 7 and comparing the FGPR for different realizations with the 'true' case

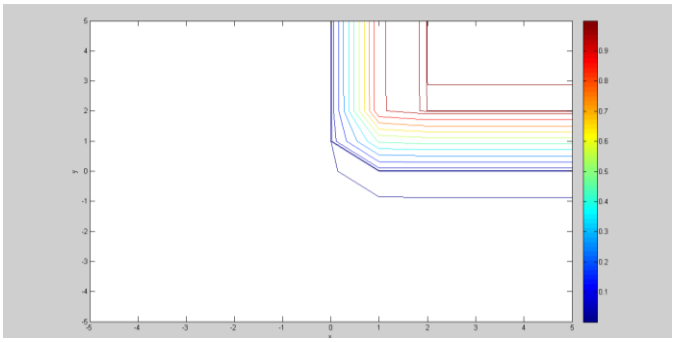

Fig. 49: Contour Map for MVCPDF obtained by varying both PERMZ [0.001:0.0001:0.01] and PERMY [1:0.1:40] in region 7 and comparing the FGPR for different realizations with the 'true' case

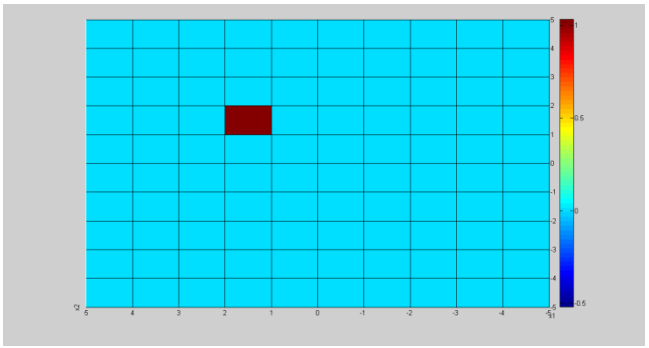

Figure 50: Contour Map for MVNPDF obtained by varying both PERMZ [0.001:0.0001:0.01] and PERMY [1:0.1:40] in region 7 and comparing the FGPR for different realizations with the 'true' case

CASE 9

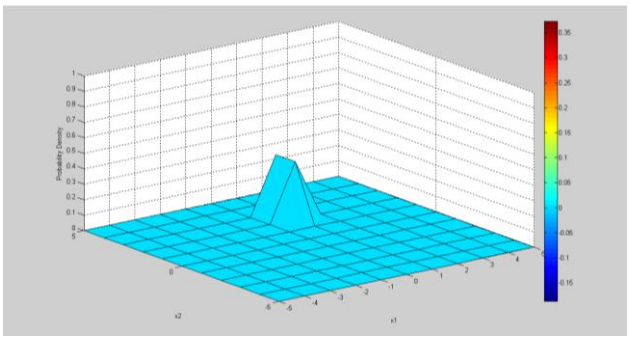

Figure 51: MVNPDF obtained by varying both PERMZ [0.001:0.0001:0.01] and PERMY [1:0.1:40] in region 8 and comparing the FGPR for different realizations with the 'true' case

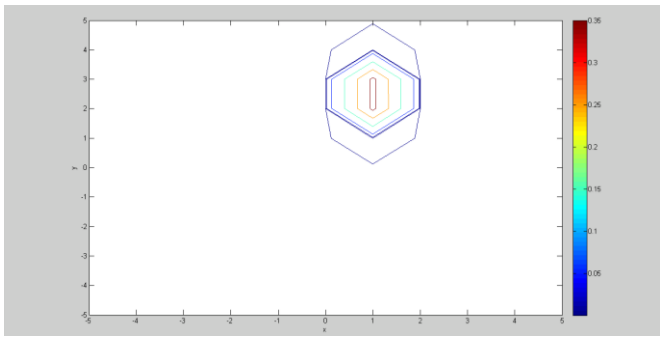

Figure 52: Contour Map for MVNPDF obtained by varying both PERMZ [0.001:0.0001:0.01] and PERMY [1:0.1:40] in region 8 and comparing the FGPR for different realizations with the 'true' case

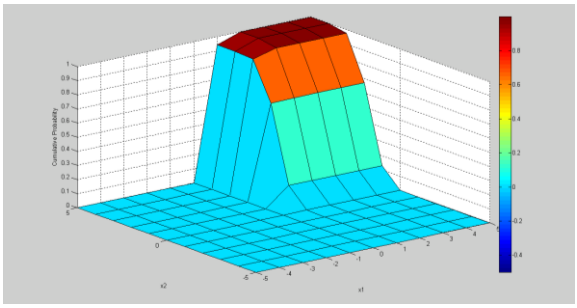

Figure 53: MVCPDF obtained by varying both PERMZ [0.001:0.0001:0.01] and PERMY [1:0.1:40] in region 8 and comparing the FGPR for different realizations with the 'true' case

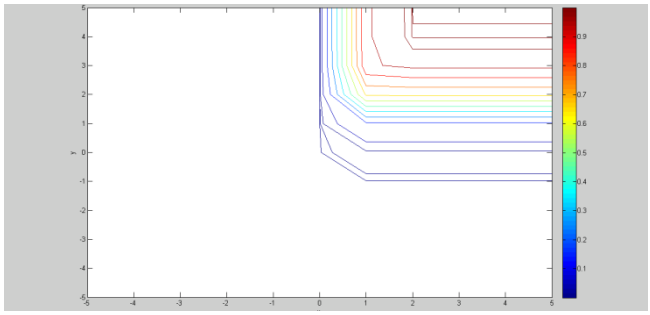

Figure 54: Contour Map for MVCPDF obtained by varying both PERMZ [0.001:0.0001:0.01] and PERMY [1:0.1:40] in region 7 and comparing the FGPR for different realizations with the 'true' case

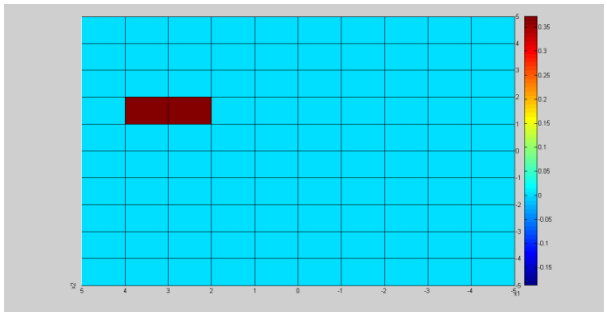

Figure 55: Contour Map for MVNPDF obtained by varying both PERMZ [0.001:0.0001:0.01] and PERMY [1:0.1:40] in region 7 and comparing the FGPR for different realizations with the 'true' case

CASE 10

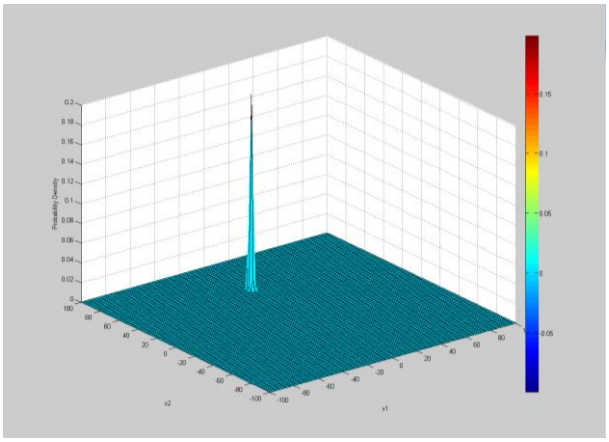

Figure 56: MVNPDF obtained by varying both PERMX [1:0.1:40] and PERMY [1:0.1:40] in region 8 and comparing the FGPR for different realizations with the 'true' case

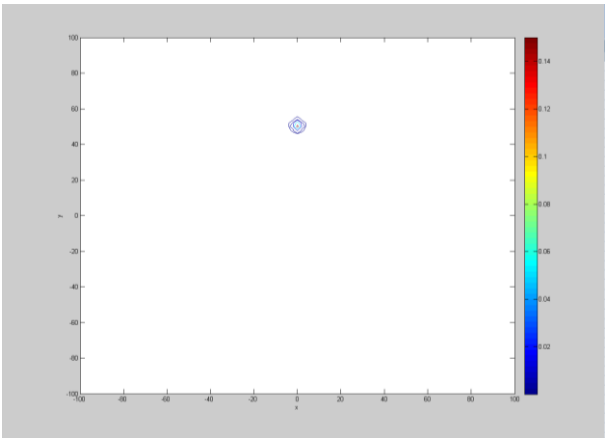

Figure 57: Contour Map for MVNPDF obtained by varying both PERMX [1:0.1:40] and PERMY [1:0.1:40] in region 8 and comparing the FGPR for different realizations with the 'true' case

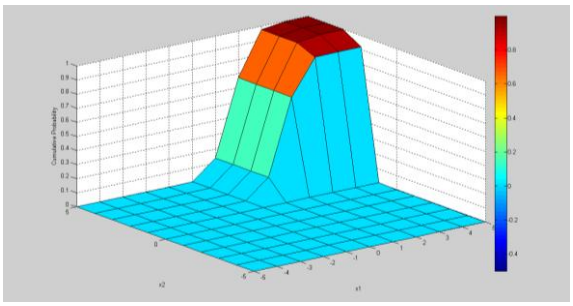

Figure 58: MVCPDF obtained by varying both PERMX [1:0.1:40] and PERMY [1:0.1:40] in region 8 and comparing the FGPR for different realizations with the 'true' case

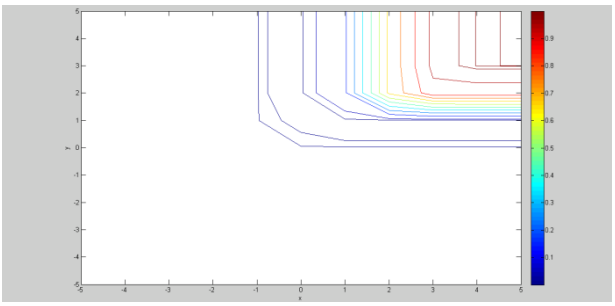

Figure 59: Contour Map for MVCPDF obtained by varying both PERMX [1:0.1:40] and PERMY [1:0.1:40] in region 8 and comparing the FGPR for different realizations with the 'true' case

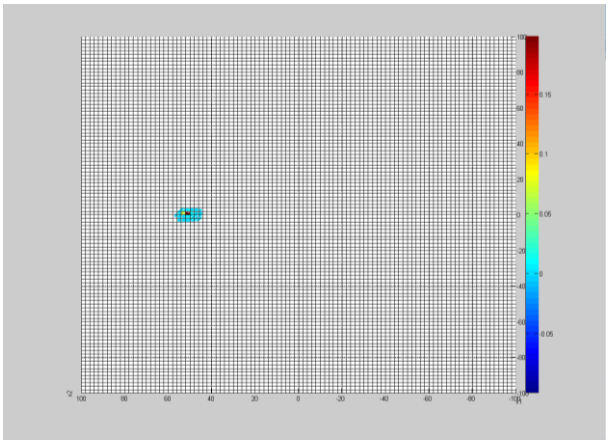

Figure 60: Contour Map for MVNPDF obtained by varying both PERMX [1:0.1:40] and PERMY [1:0.1:40] in region 8 and comparing the FGPR for different realizations with the 'true' case

CASES – UNIVARIATE PROBABILITY

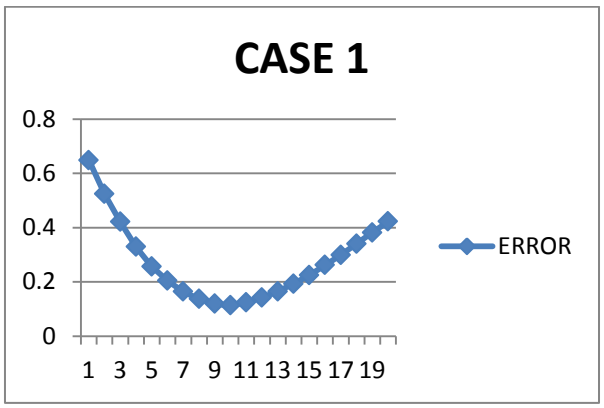

Fig. 61: RMSE obtained by varying the PORO [0.1:0.01:0.3] in region 2 and comparing the FOPR for different realizations with the 'true' case

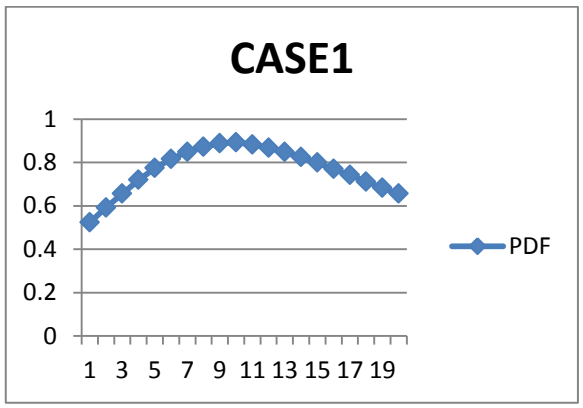

Fig. 62: PDF obtained by varying the PORO [0.1:0.01:0.3] in region 2 and comparing the FOPR for different realizations with the 'true' case

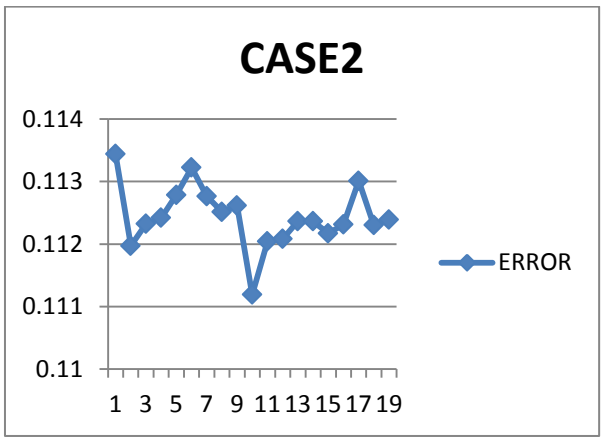

Fig. 63: RMSE obtained by varying the PERMY [1:0.1:40] in region 3 and comparing the FWPR for different realizations with the 'true' case

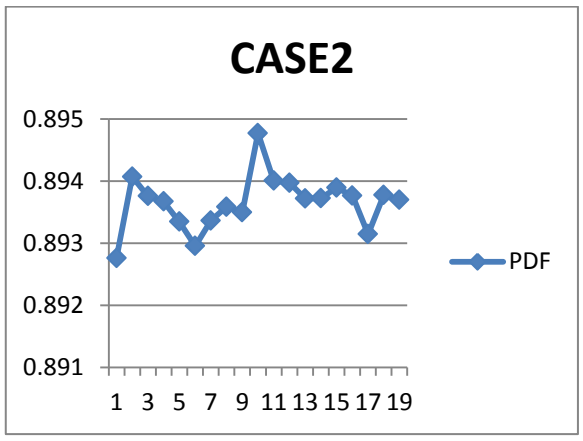

Fig. 64: PDF obtained by varying the PERMY [1:0.1:40] in region 3 and comparing the FWPR for different realizations with the 'true' case

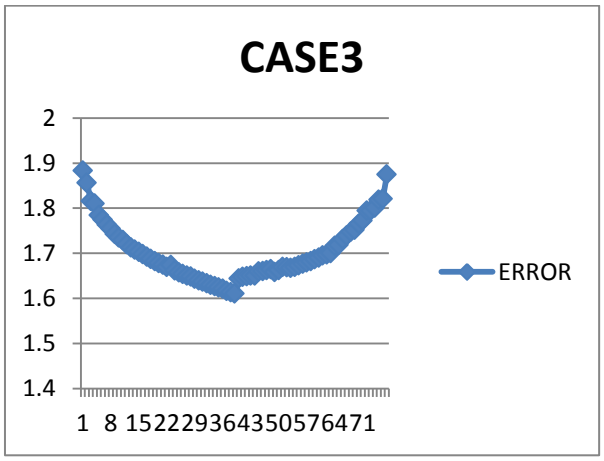

Fig. 65: RMSE obtained by varying the PORO [0.1:0.01:0.3] in region 3 and comparing the FWPR for different realiza-

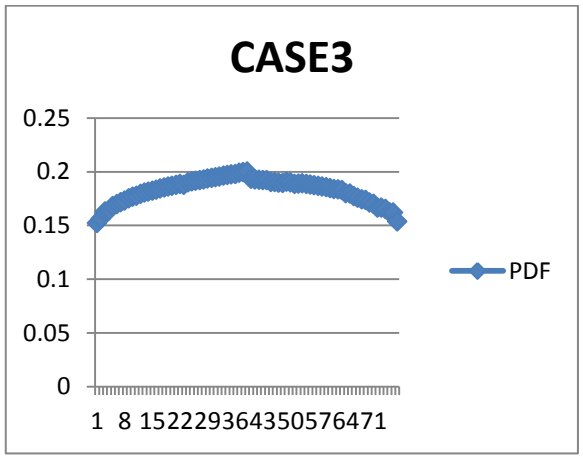

Fig. 66: PDF obtained by varying the PORO [0.1:0.01:0.3] in region 3 and comparing the FWPR for different realiza-

tions with the ‘true’ case

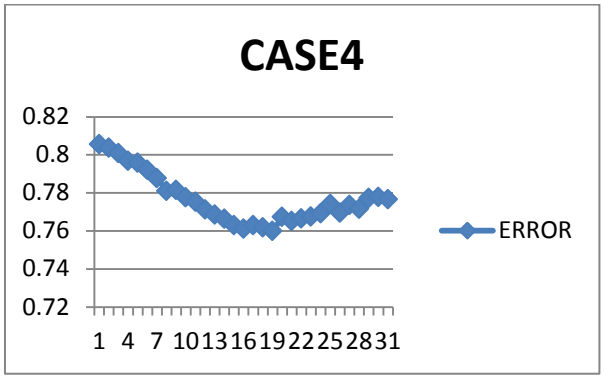

Fig. 67: RMSE obtained by varying the PORO [0.1:0.01:0.3] in region 5 and comparing the FWPR for different realizations with the ‘true’ case

tions with the ‘true’ case

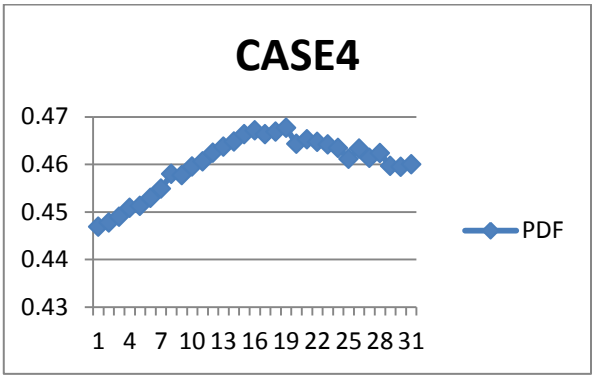

Figure 68: PDF obtained by varying the PORO [0.1:0.01:0.3] in region 5 and comparing the FWPR for different realizations with the ‘true’ case

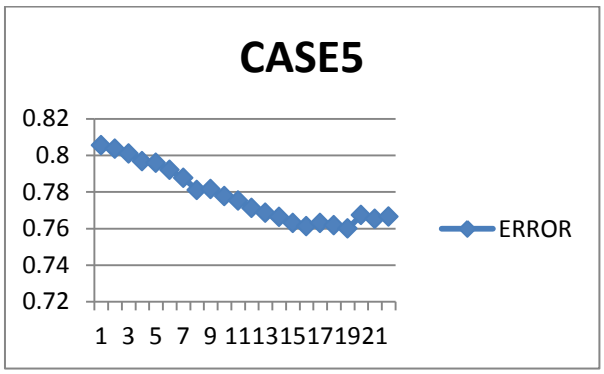

Fig. 69: RMSE obtained by varying the PERMX [1:0.1:40] in region 2 and comparing the FWPR for different realizations with the ‘true’ case

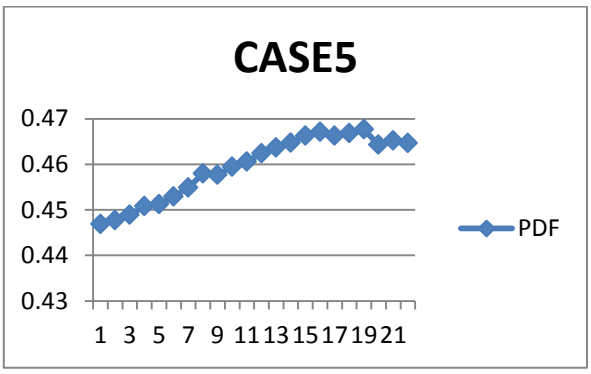

Fig. 70: PDF obtained by varying the PERMX [1:0.1:40] in region 2 and comparing the FWPR for different realizations with the ‘true’ case

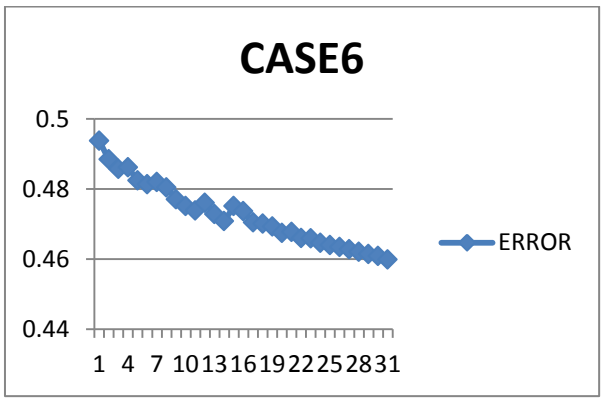

Fig. 71: RMSE obtained by varying the PERMX [1:0.1:40] in region 5 and comparing the FGPR for different realizations with the ‘true’ case

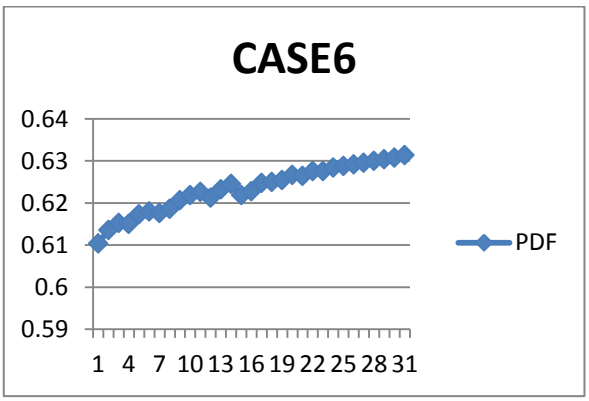

Fig. 72: PDF obtained by varying the PERMX [1:0.1:40] in region 5 and comparing the FGPR for different realizations with the ‘true’ case

APPENDIX E: COMPUTER PROGRAM RESULTS

BIVARIATE PROBABILITY DISTRIBUTION FUNCTION

CASE1

```
mu = [0.2736 0.9504];
Sigma = [0.0217 0.0136; 0.0136 0.0729];
x1 = -2:.2:2; x2 = -2:.2:2;
[X1,X2] = meshgrid(x1,x2);
F = MVNPDF([X1(:) X2(:)],mu,Sigma);
F = reshape(F,length(x2),length(x1));
surf(x1,x2,F);
caxis([min(F(:))-.5*range(F(:)),max(F(:))]);
axis([-2 2 -2 2 0 5])
xlabel('x1'); ylabel('x2'); zlabel('Probability Density');

F = mvncdf([X1(:) X2(:)],mu,Sigma);
F = reshape(F,length(x2),length(x1));
surf(x1,x2,F);
caxis([min(F(:))-.5*range(F(:)),max(F(:))]);
axis([-2 2 -2 2 0 1])
xlabel('x1'); ylabel('x2'); zlabel('Cumulative Probability');

contour(x1,x2,F,[.0001 .001 .01 .05:.1:.95 .99 .999 .9999]);
xlabel('x'); ylabel('y');
line([0 0 1 1 0],[1 0 0 1 1],'linestyle','--','color','k');
mvncdf([-2 -2],[2 2],mu,Sigma)
ans = 0.9999
```

```
 0. F= -4.257540 x= -1.269000 y=  0.797000
 1. F= -4.219874 x=  0.150000 y= -0.921000
 2. F= -4.103055 x=  0.705000 y=  1.186000
 3. F= -4.223490 x= -1.753000 y= -2.008000
 4. F= -4.224394 x=  0.721000 y= -0.801000
 5. F= -4.257540 x= -1.894000 y=  0.535000
 6. F= -4.186955 x=  0.350000 y=  1.529000
 7. F= -4.103055 x= -1.465000 y=  0.130000
 8. F= -4.162080 x=  0.462000 y= -0.638000
 9. F= -4.209596 x=  1.152000 y=  1.832000
10. F= -4.185335 x=  0.455000 y=  1.533000
11. F= -4.254905 x=  0.860000 y=  0.771000
12. F= -4.258041 x= -0.977000 y=  1.955000
13. F= -4.224344 x= -1.001000 y= -2.005000
14. F= -4.237827 x= -2.038000 y=  0.907000
15. F= -4.159221 x= -2.017000 y= -1.890000
16. F= -4.099121 x=  0.414000 y= -0.019000
17. F= -4.186955 x=  1.086000 y= -1.527000
18. F= -4.219072 x= -1.127000 y=  0.986000
19. F= -1.149257 x=  0.520000 y= -1.729000
Best Found
F= -4.258041 x=  0.273000 y=  0.952000
```

CASE2

```
mu = [0.1153 1.6663];
Sigma = [0.01 0.0136; 0.0136 0.0729];
x1 = -2:.2:2; x2 = -2:.2:2;
[X1,X2] = meshgrid(x1,x2);
F = MVNPDF([X1(:) X2(:)],mu,Sigma);
F = reshape(F,length(x2),length(x1));
surf(x1,x2,F);
caxis([min(F(:))-.5*range(F(:)),max(F(:))]);
axis([-2 2 -2 2 0 5])
xlabel('x1'); ylabel('x2'); zlabel('Probability Density');

F = mvncdf([X1(:) X2(:)],mu,Sigma);
F = reshape(F,length(x2),length(x1));
surf(x1,x2,F);
caxis([min(F(:))-.5*range(F(:)),max(F(:))]);
axis([-2 2 -2 2 0 1])
xlabel('x1'); ylabel('x2'); zlabel('Cumulative Probability');

contour(x1,x2,F,[.0001 .001 .01 .05:.1:.95 .99 .999 .9999]);
xlabel('x'); ylabel('y');
line([0 0 1 1 0],[1 0 0 1 1],'linestyle','--','color','k');


mvncdf([-2 -2],[2 2],mu,Sigma)


ans = 1.0000
```

```
 0. F= -6.823430 x=  0.739000 y=  1.723000
 1. F= -5.783104 x= -1.839000 y= -1.498000
 2. F= -3.177750 x=  0.171000 y=  0.117000
 3. F= -5.783104 x= -0.241000 y= -0.394000
 4. F= -5.783104 x= -0.244000 y=  1.477000
 5. F= -5.782063 x=  1.075000 y= -1.344000
 6. F= -5.762378 x= -1.463000 y=  0.065000
 7. F= -6.356151 x=  0.430000 y= -0.872000
 8. F= -5.782063 x= -0.712000 y=  1.081000
 9. F= -5.783104 x= -1.962000 y=  1.197000
10. F= -5.747011 x=  1.487000 y=  1.481000
11. F= -3.134999 x=  1.723000 y=  2.002000
12. F= -6.768634 x= -0.400000 y=  0.609000
13. F= -0.399312 x= -1.690000 y= -0.990000
14. F= -5.782063 x= -1.204000 y=  0.306000
15. F= -6.757206 x=  1.444000 y= -0.990000
16. F= -6.816367 x= -0.878000 y=  0.552000
17. F= -6.823290 x= -0.069000 y=  0.663000
18. F= -5.770792 x= -0.470000 y=  1.114000
19. F= -6.823430 x=  0.823000 y= -0.574000
```

Best Found
F= -6.823430 x= 0.115000 y= 1.666000

CASE3

```
mu = [0.0047 0.6002];
Sigma = [0.5 0.002; 0.002 0.035];
x1 = -2:.2:2; x2 = -2:.2:2;
[X1,X2] = meshgrid(x1,x2);
F = MVNPDF([X1(:) X2(:)],mu,Sigma);
F = reshape(F,length(x2),length(x1));
surf(x1,x2,F);
caxis([min(F(:))-.5*range(F(:)),max(F(:))]);
axis([-2 2 -2 2 0 1.5])
xlabel('x1'); ylabel('x2'); zlabel('Probability Density');

F = mvncdf([X1(:) X2(:)],mu,Sigma);
F = reshape(F,length(x2),length(x1));
surf(x1,x2,F);
caxis([min(F(:))-.5*range(F(:)),max(F(:))]);
axis([-2 2 -2 2 0 1])
xlabel('x1'); ylabel('x2'); zlabel('Cumulative Probability');

contour(x1,x2,F,[.0001 .001 .01 .05:.1:.95 .99 .999 .9999]);
xlabel('x'); ylabel('y');
line([0 0 1 1 0],[1 0 0 1 1],'linestyle','--','color','k');


mvncdf([-2 -2],[2 2],mu,Sigma)


ans = 0.9953
```

```
 0. F= -1.073955 x= -0.459000 y= -1.290000
 1. F= -1.073955 x=  0.604000 y= -2.016000
 2. F= -1.073955 x= -1.763000 y=  1.297000
 3. F= -1.073955 x=  0.251000 y= -1.416000
 4. F= -0.122651 x=  1.087000 y= -1.068000
 5. F= -1.203211 x= -1.181000 y= -0.085000
 6. F= -0.122651 x=  1.393000 y= -1.023000
 7. F= -1.073955 x=  1.353000 y= -0.500000
 8. F= -1.176313 x=  1.819000 y= -0.662000
 9. F= -1.202190 x=  0.799000 y=  2.048000
10. F= -1.073955 x= -1.415000 y= -0.427000
11. F= -1.073955 x= -0.553000 y= -0.337000
12. F= -1.192369 x= -1.563000 y=  1.950000
13. F= -1.073955 x=  2.004000 y= -1.356000
14. F= -1.203211 x= -1.764000 y=  0.965000
15. F= -1.203211 x=  0.679000 y=  2.003000
16. F= -1.203211 x= -1.477000 y= -1.275000
17. F= -1.176312 x=  0.480000 y=  0.167000
18. F= -1.192369 x=  0.083000 y=  0.949000
19. F= -1.073955 x= -0.532000 y=  1.300000
```

Best Found

F= -1.203211 x= 0.005000 y= 0.599000

CASE4

```
mu = [0 0.2751];
Sigma = [0.5 0; 0 0.4584];
x1 = -5:.2:5; x2 = -5:.2:5;
[X1,X2] = meshgrid(x1,x2);
F = MVNPDF([X1(:) X2(:)],mu,Sigma);
F = reshape(F,length(x2),length(x1));
surf(x1,x2,F);
caxis([min(F(:))-.5*range(F(:)),max(F(:))]);
axis([-5 5 -5 5 0 0.4])
xlabel('x1'); ylabel('x2'); zlabel('Probability Density');

F = mvncdf([X1(:) X2(:)],mu,Sigma);
F = reshape(F,length(x2),length(x1));
surf(x1,x2,F);
caxis([min(F(:))-.5*range(F(:)),max(F(:))]);
axis([-5 5 -5 5 0 1])
xlabel('x1'); ylabel('x2'); zlabel('Cumulative Probability');

contour(x1,x2,F,[.0001 .001 .01 .05:.1:.95 .99 .999 .9999]);
xlabel('x'); ylabel('y');
line([0 0 1 1 0],[1 0 0 1 1],'linestyle','--','color','k');
```

mvncdf([-5 -5],[5 5],mu,Sigma)

ans = 1

```
 0. F= -0.312700 x=  0.294000 y= -1.525000
 1. F= -0.332439 x=  1.271000 y= -2.008000
 2. F= -0.332440 x= -1.611000 y= -0.551000
 3. F= -0.332278 x= -0.024000 y=  1.654000
 4. F= -0.332293 x=  0.725000 y= -0.981000
 5. F= -0.332439 x=  0.995000 y=  0.950000
 6. F= -0.332440 x=  0.703000 y=  2.007000
 7. F= -0.332439 x=  0.608000 y=  1.772000
 8. F= -0.332440 x=  1.414000 y= -1.871000
 9. F= -0.332293 x= -0.861000 y= -0.181000
10. F= -0.332440 x=  1.854000 y= -0.569000
11. F= -0.332293 x=  0.389000 y=  1.736000
12. F= -0.332379 x=  1.440000 y=  1.034000
13. F= -0.332293 x=  1.701000 y=  1.166000
14. F= -0.332439 x=  1.239000 y=  2.048000
15. F= -0.312700 x= -0.443000 y=  0.532000
16. F= -0.332379 x=  1.682000 y=  1.291000
17. F= -0.332434 x=  1.311000 y= -2.046000
18. F= -0.332278 x= -1.856000 y= -1.242000
19. F= -0.332293 x= -1.580000 y= -0.228000
```

Best Found
F= -0.332440 x= -0.000000 y= 0.275000

CASE5

```
mu = [7.2309 29.7057];
Sigma = [33.1234 -12.499; -12.499 260.4879];
x1 = -100:2:100; x2 = -100:2:100;
[X1,X2] = meshgrid(x1,x2);
F = MVNPDF([X1(:) X2(:)],mu,Sigma);
F = reshape(F,length(x2),length(x1));
surf(x1,x2,F);
caxis([min(F(:))-.5*range(F(:)),max(F(:))]);
axis([-100 100 -100 100 0 .002])
xlabel('x1'); ylabel('x2'); zlabel('Probability Density');

F = mvncdf([X1(:) X2(:)],mu,Sigma);
F = reshape(F,length(x2),length(x1));
surf(x1,x2,F);
caxis([min(F(:))-.5*range(F(:)),max(F(:))]);
axis([-100 100 -100 100 0 1])
xlabel('x1'); ylabel('x2'); zlabel('Cumulative Probability');

contour(x1,x2,F,[.0001 .001 .01 .05:.1:.95 .99 .999 .9999]);
xlabel('x'); ylabel('y');
line([0 0 1 1 0],[1 0 0 1 1],'linestyle','--','color','k');
```

```
mvncdf([-100 -100],[5 5],mu,Sigma)

ans = 0.0161
```

```
 0. F= -0.000360 x=  1.284000 y=  0.721000
 1. F= -0.000360 x=  1.221000 y=  2.029000
 2. F= -0.000360 x=  0.181000 y= -1.685000
 3. F= -0.000360 x= -2.008000 y= -0.133000
 4. F= -0.000360 x= -1.676000 y=  0.981000
 5. F= -0.000360 x=  0.749000 y= -1.632000
 6. F= -0.000360 x= -1.917000 y= -0.981000
 7. F= -0.000360 x= -0.638000 y= -1.551000
 8. F= -0.000360 x=  1.327000 y= -1.644000
 9. F= -0.000360 x=  0.207000 y= -1.241000
10. F= -0.000360 x=  1.227000 y= -0.335000
11. F= -0.000360 x= -1.790000 y=  0.235000
12. F= -0.000360 x=  1.377000 y=  0.180000
13. F= -0.000360 x= -0.919000 y=  0.595000
14. F= -0.000360 x= -1.021000 y= -0.635000
15. F= -0.000360 x= -1.995000 y= -0.278000
16. F= -0.000360 x= -0.051000 y=  1.080000
17. F= -0.000360 x= -1.929000 y= -1.117000
18. F= -0.000360 x= -0.142000 y=  1.488000
19. F= -0.000360 x= -0.223000 y= -0.565000
```

```
Best Found
F= -0.000360 x=  4.023000 y=  4.023000
```

CASE6

```
mu = [0.0107 50.4777];
Sigma = [0.5 0.0017; 0.0017 1.0285];
x1 = -100:2:100; x2 = -100:2:100;
[X1,X2] = meshgrid(x1,x2);
F = MVNPDF([X1(:) X2(:)],mu,Sigma);
F = reshape(F,length(x2),length(x1));
surf(x1,x2,F);
caxis([min(F(:))-.5*range(F(:)),max(F(:))]);
axis([-100 100 -100 100 0 .2])
xlabel('x1'); ylabel('x2'); zlabel('Probability Density');

F = mvncdf([X1(:) X2(:)],mu,Sigma);
F = reshape(F,length(x2),length(x1));
surf(x1,x2,F);
caxis([min(F(:))-.5*range(F(:)),max(F(:))]);
axis([-100 100 -100 100 0 1])
xlabel('x1'); ylabel('x2'); zlabel('Cumulative Probability');

contour(x1,x2,F,[.0001 .001 .01 .05:.1:.95 .99 .999 .9999]);
xlabel('x'); ylabel('y');
line([0 0 1 1 0],[1 0 0 1 1],'linestyle','--','color','k');
```

```
mvncdf([-100 -100],[100 100],mu,Sigma)

ans = 1
```

```
 0. F= -0.000401 x=  0.600000 y= -1.210000
 1. F= -0.000401 x= -1.216000 y=  0.186000
 2. F= -0.000401 x= -1.910000 y= -1.866000
 3. F= -0.000401 x=  1.615000 y= -0.220000
 4. F= -0.000401 x= -0.412000 y= -1.621000
 5. F= -0.000401 x=  1.631000 y= -0.282000
 6. F= -0.000401 x=  0.094000 y= -2.000000
 7. F= -0.000401 x=  1.634000 y=  0.030000
 8. F= -0.000401 x= -0.694000 y= -1.431000
 9. F= -0.000401 x= -1.440000 y=  0.101000
10. F= -0.000401 x= -0.642000 y= -0.293000
11. F= -0.000401 x=  0.177000 y= -0.085000
12. F= -0.000401 x=  0.911000 y=  0.340000
13. F= -0.000401 x= -0.702000 y=  0.453000
14. F= -0.000401 x=  1.214000 y=  0.704000
15. F= -0.000401 x= -0.862000 y=  1.441000
16. F= -0.000401 x= -0.742000 y=  0.752000
17. F= -0.000401 x= -1.922000 y=  0.855000
18. F= -0.000401 x=  0.823000 y=  1.009000
19. F= -0.000401 x=  1.979000 y= -0.788000

Best Found
F= -0.000401 x=  4.023000 y=  4.023000
```

CASE7

```
mu = [0.4726      0.7764];
Sigma = [0.0812 0.0875; 0.0875            0.1733];
x1 = -5:1:5; x2 = -5:1:5;
[X1,X2] = meshgrid(x1,x2);
F = MVNPDF([X1(:) X2(:)],mu,Sigma);
F = reshape(F,length(x2),length(x1));
surf(x1,x2,F);
caxis([min(F(:))-.5*range(F(:)),max(F(:))]);
axis([-5 5 -5 5 0 .5])
xlabel('x1'); ylabel('x2'); zlabel('Probability Density');

F = mvncdf([X1(:) X2(:)],mu,Sigma);
F = reshape(F,length(x2),length(x1));
surf(x1,x2,F);
caxis([min(F(:))-.5*range(F(:)),max(F(:))]);
axis([-5 5 -5 5 0 1])
xlabel('x1'); ylabel('x2'); zlabel('Cumulative Probability');

contour(x1,x2,F,[.0001 .001 .01 .05:.1:.95 .99 .999 .9999]);
xlabel('x'); ylabel('y');
line([0 0 1 1 0],[1 0 0 1 1],'linestyle','--','color','k');
```

```
mvncdf([-100 -100],[100 100],mu,Sigma)

ans = 1
```

```
 0. F= -0.000401 x=  0.600000 y= -1.210000
 1. F= -0.000401 x= -1.216000 y=  0.186000
 2. F= -0.000401 x= -1.910000 y= -1.866000
 3. F= -0.000401 x=  1.615000 y= -0.220000
 4. F= -0.000401 x= -0.412000 y= -1.621000
 5. F= -0.000401 x=  1.631000 y= -0.282000
 6. F= -0.000401 x=  0.094000 y= -2.000000
 7. F= -0.000401 x=  1.634000 y=  0.030000
 8. F= -0.000401 x= -0.694000 y= -1.431000
 9. F= -0.000401 x= -1.440000 y=  0.101000
10. F= -0.000401 x= -0.642000 y= -0.293000
11. F= -0.000401 x=  0.177000 y= -0.085000
12. F= -0.000401 x=  0.911000 y=  0.340000
13. F= -0.000401 x= -0.702000 y=  0.453000
14. F= -0.000401 x=  1.214000 y=  0.704000
15. F= -0.000401 x= -0.862000 y=  1.441000
16. F= -0.000401 x= -0.742000 y=  0.752000
17. F= -0.000401 x= -1.922000 y=  0.855000
18. F= -0.000401 x=  0.823000 y=  1.009000
19. F= -0.000401 x=  1.979000 y= -0.788000
```

Best Found

F= -0.000401 x= 4.023000 y= 4.023000

CASE8

```
mu = [0.7778      1.0062];
Sigma = [0.02 0.001; 0.001 0.1001];
x1 = -5:1:5; x2 = -5:1:5;
[X1,X2] = meshgrid(x1,x2);
F = MVNPDF([X1(:) X2(:)],mu,Sigma);
F = reshape(F,length(x2),length(x1));
surf(x1,x2,F);
caxis([min(F(:))-.5*range(F(:)),max(F(:))]);
axis([-5 5 -5 5 0 1])
xlabel('x1'); ylabel('x2'); zlabel('Probability Density');

F = mvncdf([X1(:) X2(:)],mu,Sigma);
F = reshape(F,length(x2),length(x1));
surf(x1,x2,F);
caxis([min(F(:))-.5*range(F(:)),max(F(:))]);
axis([-5 5 -5 5 0 1])
xlabel('x1'); ylabel('x2'); zlabel('Cumulative Probability');

contour(x1,x2,F,[.0001 .001 .01 .05:.1:.95 .99 .999 .9999]);
xlabel('x'); ylabel('y');
line([0 0 1 1 0],[1 0 0 1 1],'linestyle','--','color','k');
```

```
mvncdf([-100 -100],[100 100],mu,Sigma)

ans = 1
```

```
 0. F= -0.000401 x=  0.600000 y= -1.210000
 1. F= -0.000401 x= -1.216000 y=  0.186000
 2. F= -0.000401 x= -1.910000 y= -1.866000
 3. F= -0.000401 x=  1.615000 y= -0.220000
 4. F= -0.000401 x= -0.412000 y= -1.621000
 5. F= -0.000401 x=  1.631000 y= -0.282000
 6. F= -0.000401 x=  0.094000 y= -2.000000
 7. F= -0.000401 x=  1.634000 y=  0.030000
 8. F= -0.000401 x= -0.694000 y= -1.431000
 9. F= -0.000401 x= -1.440000 y=  0.101000
10. F= -0.000401 x= -0.642000 y= -0.293000
11. F= -0.000401 x=  0.177000 y= -0.085000
12. F= -0.000401 x=  0.911000 y=  0.340000
13. F= -0.000401 x= -0.702000 y=  0.453000
14. F= -0.000401 x=  1.214000 y=  0.704000
15. F= -0.000401 x= -0.862000 y=  1.441000
16. F= -0.000401 x= -0.742000 y=  0.752000
17. F= -0.000401 x= -1.922000 y=  0.855000
18. F= -0.000401 x=  0.823000 y=  1.009000
19. F= -0.000401 x=  1.979000 y= -0.788000

Best Found
F= -0.000401 x=  4.023000 y=  4.023000
```

CASE9

```
mu = [0.7764 1.7165];
Sigma = [0.02 0.07; 0.07 0.41];
x1 = -5:1:5; x2 = -5:1:5;
[X1,X2] = meshgrid(x1,x2);
F = MVNPDF([X1(:) X2(:)],mu,Sigma);
F = reshape(F,length(x2),length(x1));
surf(x1,x2,F);
caxis([min(F(:))-.5*range(F(:)),max(F(:))]);
axis([-5 5 -5 5 0 1])
xlabel('x1'); ylabel('x2'); zlabel('Probability Density');

F = mvncdf([X1(:) X2(:)],mu,Sigma);
F = reshape(F,length(x2),length(x1));
surf(x1,x2,F);
caxis([min(F(:))-.5*range(F(:)),max(F(:))]);
axis([-5 5 -5 5 0 1])
xlabel('x1'); ylabel('x2'); zlabel('Cumulative Probability');

contour(x1,x2,F,[.0001 .001 .01 .05:.1:.95 .99 .999 .9999]);
xlabel('x'); ylabel('y');
line([0 0 1 1 0],[1 0 0 1 1],'linestyle','--','color','k');
```

```
mvncdf([-100 -100],[100 100],mu,Sigma)

ans = 1
```

```
  0. F= -0.000401 x=  0.600000 y= -1.210000
  1. F= -0.000401 x= -1.216000 y=  0.186000
  2. F= -0.000401 x= -1.910000 y= -1.866000
  3. F= -0.000401 x=  1.615000 y= -0.220000
  4. F= -0.000401 x= -0.412000 y= -1.621000
  5. F= -0.000401 x=  1.631000 y= -0.282000
  6. F= -0.000401 x=  0.094000 y= -2.000000
  7. F= -0.000401 x=  1.634000 y=  0.030000
  8. F= -0.000401 x= -0.694000 y= -1.431000
  9. F= -0.000401 x= -1.440000 y=  0.101000
 10. F= -0.000401 x= -0.642000 y= -0.293000
 11. F= -0.000401 x=  0.177000 y= -0.085000
 12. F= -0.000401 x=  0.911000 y=  0.340000
 13. F= -0.000401 x= -0.702000 y=  0.453000
 14. F= -0.000401 x=  1.214000 y=  0.704000
 15. F= -0.000401 x= -0.862000 y=  1.441000
 16. F= -0.000401 x= -0.742000 y=  0.752000
 17. F= -0.000401 x= -1.922000 y=  0.855000
 18. F= -0.000401 x=  0.823000 y=  1.009000
 19. F= -0.000401 x=  1.979000 y= -0.788000
Best Found
F= -0.000401 x=  4.023000 y=  4.023000
```

CASE10

```
mu = [1.7139 1.6559];
Sigma = [0.42 0.14; 0.14 0.06];
x1 = -5:1:5; x2 = -5:1:5;
[X1,X2] = meshgrid(x1,x2);
F = MVNPDF([X1(:) X2(:)],mu,Sigma);
F = reshape(F,length(x2),length(x1));
surf(x1,x2,F);
caxis([min(F(:))-.5*range(F(:)),max(F(:))]);
axis([-5 5 -5 5 0 0.5])
xlabel('x1'); ylabel('x2'); zlabel('Probability Density');

F = mvncdf([X1(:) X2(:)],mu,Sigma);
F = reshape(F,length(x2),length(x1));
surf(x1,x2,F);
caxis([min(F(:))-.5*range(F(:)),max(F(:))]);
axis([-5 5 -5 5 0 1])
xlabel('x1'); ylabel('x2'); zlabel('Cumulative Probability');

contour(x1,x2,F,[.0001 .001 .01 .05:.1:.95 .99 .999 .9999]);
xlabel('x'); ylabel('y');
line([0 0 1 1 0],[1 0 0 1 1],'linestyle','--','color','k');
```

```
mvncdf([-100 -100],[100 100],mu,Sigma)

ans = 1
```

```
 0. F= -0.000401 x=  0.600000 y= -1.210000
 1. F= -0.000401 x= -1.216000 y=  0.186000
 2. F= -0.000401 x= -1.910000 y= -1.866000
 3. F= -0.000401 x=  1.615000 y= -0.220000
 4. F= -0.000401 x= -0.412000 y= -1.621000
 5. F= -0.000401 x=  1.631000 y= -0.282000
 6. F= -0.000401 x=  0.094000 y= -2.000000
 7. F= -0.000401 x=  1.634000 y=  0.030000
 8. F= -0.000401 x= -0.694000 y= -1.431000
 9. F= -0.000401 x= -1.440000 y=  0.101000
10. F= -0.000401 x= -0.642000 y= -0.293000
11. F= -0.000401 x=  0.177000 y= -0.085000
12. F= -0.000401 x=  0.911000 y=  0.340000
13. F= -0.000401 x= -0.702000 y=  0.453000
14. F= -0.000401 x=  1.214000 y=  0.704000
15. F= -0.000401 x= -0.862000 y=  1.441000
16. F= -0.000401 x= -0.742000 y=  0.752000
17. F= -0.000401 x= -1.922000 y=  0.855000
18. F= -0.000401 x=  0.823000 y=  1.009000
19. F= -0.000401 x=  1.979000 y= -0.788000

Best Found
F= -0.000401 x=  4.023000 y=  4.023000
```

APPENDIX F: HILL CLIMBING ALGORITHM

Algorithm: Simple Hill Climbing:

Step 1: Evaluate the initial state. It it is also a goal state, then returns it and quit.
Otherwise continue with the initial state as the current state.
Step 2: Loop until a solution is found or until there are no new operators left to be applied in the current state:

(a) Select an operator that has not yet been applied to the current state and apply it to produce a new state.
(b) Evaluate the new state.
(i) If it is a goal state, then return it and quit .
(ii) If it is not a goal state, but it is better than the current state, then make it the current state.
(iii) If it is not better than the current state, then continue in the loop (Stuart Rusell, 2010)

```
function [ F ] = myfunc( str )
% Function to be MINIMIZED
% By Kyriakos Tsourapas
% You may contact me through the Mathworks site
% University of Essex 2002

[x, y] = myconvert( str );
mu = [0.0895  0.0760];
Sigma = [0.0022   0.0023
0.0023    0.0025];
x1 = -10:.2:10; x2 = -10:.2:10;
[X1,X2] = meshgrid(x1,x2);
F = MVNPDF([X1(:) X2(:)],mu,Sigma);
F=-F
F = reshape(F,length(x2),length(x1));
surf(x1,x2,F);
caxis([min(F(:))-.5*range(F(:)),max(F(:))]);
axis([-10 10 -10 10 0 100])

%caxis([min(F(:))-.5*range(F(:)),max(F(:))]);
%axis([-2 10 -2 10 0 -1])
xlabel('x1'); ylabel('x2'); zlabel('Probability Density');
%F = x^2 + y^2;
End
% START THE HILL CLIMBING
 % LOOK IN start_points STARTING POINTS
while t < start_points
   local = FALSE;

   % CREATE A NEW STRING AT RANDOM, WITHIN THE LIMITS
   num1 = 10; % just to get in the loop
   num2 = 10; % just to get in the loop
   while num1 < lowlimit | num1 > uplimit | num2 < lowlimit | num2 > uplimit
      str = rand(26,1);

      for i=1:size(str,1)
         if str(i) < 0.5
            str(i) = 0;
```

```
        else
            str(i) = 1;
        end
    end
    [num1, num2] = myconvert(str);
end

% SEARCH UNTIL LOCAL OPTIMUM IS REACHED
while ~local
    k = 1;
    F = myfunc(str);
    % CREATE 26 DIFFERENT STRINGS AND KEEP THE BEST
    % BY FLIPPING A SIGLLE BIT AT A TIME
    while k < 27
        new_str = newstr(str, k);
        newF = myfunc(new_str);

        if k == 1
            bestStr_sofar = new_str;
        else
            bestF_sofar = myfunc(bestStr_sofar);
            if newF < bestF_sofar
                bestStr_sofar = new_str;
            end
        end

         k = k + 1;
    end
     % COMPARE THE BEST OF THE 26 STRINGS WITH
    % THE STARTING STRING
    if F > bestF_sofar
        str = bestStr_sofar;
    else
        local = TRUE;
    end

end

F = myfunc(str);

if (t == 0) | (F < bestF)
    bestF = F;
    best_str = str;
end

disp( sprintf('%4d. F=%10f x=%10f y=%10f', t, F, num1, num2) );
t = t + 1;
end

[num1, num2] = myconvert(best_str);
bestF = myfunc(best_str);
disp( sprintf('\nBest Found\nF=%10f x=%10f y=%10f', bestF, num1, num2) );
```