

Novel Sampling Techniques for Reservoir History Matching Optimisation and Uncertainty Quantification in Flow Prediction

Lina Mahgoub Yahya Mohamed

Submitted for the
Degree of Doctor of Philosophy
Institute of Petroleum Engineering
Heriot-Watt University
January 2011

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that the copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the prior written consent of the author or the University (as may be appropriate).

Abstract

Modern reservoir management has an increasing focus on accurately predicting the likely range of field recoveries. A variety of assisted history matching techniques has been developed across the research community concerned with this topic. These techniques are based on obtaining multiple models that closely reproduce the historical flow behaviour of a reservoir. The set of resulted history matched models is then used to quantify uncertainty in predicting the future performance of the reservoir and providing economic evaluations for different field development strategies. The key step in this workflow is to employ algorithms that sample the parameter space in an efficient but appropriate manner. The algorithm choice has an impact on how fast a model is obtained and how well the model fits the production data. The sampling techniques that have been developed to date include, among others, gradient based methods, evolutionary algorithms, and ensemble Kalman filter (EnKF).

This thesis has investigated and further developed the following sampling and inference techniques: Particle Swarm Optimisation (PSO), Hamiltonian Monte Carlo, and Population Markov Chain Monte Carlo. The inspected techniques have the capability of navigating the parameter space and producing history matched models that can be used to quantify the uncertainty in the forecasts in a faster and more reliable way. The analysis of these techniques, compared with Neighbourhood Algorithm (NA), has shown how the different techniques affect the predicted recovery from petroleum systems and the benefits of the developed methods over the NA.

The history matching problem is multi-objective in nature, with the production data possibly consisting of multiple types, coming from different wells, and collected at different times. Multiple objectives can be constructed from these data and explicitly be

optimised in the multi-objective scheme. The thesis has extended the PSO to handle multi-objective history matching problems in which a number of possible conflicting objectives must be satisfied simultaneously. The benefits and efficiency of innovative multi-objective particle swarm scheme (MOPSO) are demonstrated for synthetic reservoirs. It is demonstrated that the MOPSO procedure can provide a substantial improvement in finding a diverse set of good fitting models with a fewer number of very costly forward simulations runs than the standard single objective case, depending on how the objectives are constructed.

The thesis has also shown how to tackle a large number of unknown parameters through the coupling of high performance global optimisation algorithms, such as PSO, with model reduction techniques such as kernel principal component analysis (PCA), for parameterising spatially correlated random fields. The results of the PSO-PCA coupling applied to a recent SPE benchmark history matching problem have demonstrated that the approach is indeed applicable for practical problems. A comparison of PSO with the EnKF data assimilation method has been carried out and has concluded that both methods have obtained comparable results on the example case. This point reinforces the need for using a range of assisted history matching algorithms for more confidence in predictions.

Dedication

To my family ...

Acknowledgements

First of all, I thank Allah for his grace, blessings and kindness showered upon me. I would like to heartily express my sincere gratitude to my supervisor, Professor Mike Christie, for giving me the opportunity to study for a PhD and introducing me to a timely research topic. His invaluable guidance, encouragement, and constructive comments have helped me throughout the duration of my studies and have been of great value for me. I owe my deepest gratitude to my second supervisor Dr. Vasily Demyanov for his helpful discussions and continuous support.

I am grateful for the time and financial support provided by the sponsors of the Uncertainty Quantification Project at Heriot-Watt University: BG, BP, ConocoPhillips, and JOGMEG. Many thanks go to Schlumberger-GeoQuest for providing the ECLIPSE and PETREL software. My thanks and gratitude go to Richard Rivenq for offering me the internship opportunity at Total E&P UK. Thanks to Dick Kachuma for being supportive during that time and thereafter.

I wish to express my warm thanks to Monika, Dan, Gillian, Alannah, Temistocles, Lorna, Mohammad and Asaad. I also would like to thank the IT staff Chis, Alan, and Chris.

I would like to thank my PhD examiners Prof. Peter King and Prof. Gary Couples for their invaluable comments and feedback which have significantly improved the thesis.

I sincerely thank my parents, brothers, and sisters for their unconditional love and their support. I owe my sincere gratitude to Ashraf and Mohammed for being so supportive and helpful throughout my studies and their continuous encouragement.

Table of Contents

Table of Contents	i
List of Publications from Thesis	iii
List of Figures	v
List of Tables.....	xii
Chapter 1 – Introduction	1
1.1 Reservoir Model History Matching as an Inverse Problem	1
1.2 Thesis Objectives	3
1.3 Thesis Structure.....	4
Chapter 2 – Reservoir Simulation and Characterisation: Literature Review	7
2.1 Reservoir Analysis Techniques.....	7
2.2 Flow Equations	13
2.3 Gridding and Upscaling Techniques	16
2.4 Reservoir Characterisation and Modelling.....	18
2.5 Parameterisation.....	24
2.6 Chapter Summary.....	34
Chapter 3 – Reservoir Model History Matching and Uncertainty Quantification	
Techniques: Literature Review	35
3.1 Objective Function	35
3.2 History Matching	37
3.3 Assisted History Matching Algorithms.....	39
3.4 Quantifying the Uncertainty.....	44
3.5 Comparison of Uncertainty Quantification Methods.....	50
3.6 Chapter Summary.....	51
Chapter 4 – History Matching and Uncertainty Quantification of Reservoir Simulation	
Models with Particle Swarm Optimisation	52
4.1 Particle Swarm Optimisation (PSO)	54
4.2 Efficiency Criteria for Evaluating the Algorithms.....	62
4.3 Petroleum Test Studies.....	63
4.4 Chapter Summary	118
Chapter 5 – History Matching and Uncertainty Quantification: Multi-Objective Particle	
Swarm Optimisation Approach.....	121
5.1 Multi-Objective Optimisation in Petroleum	121
5.2 Basic Multi-Objective Optimisation Concepts	125
5.3 Objectives and Mechanisms of MOPSO	128
5.4 Categorisation of MOPSO Techniques	135
5.5 Multi-Objective Particle Swarm Optimisation Variant.....	136
5.6 Numerical Experiments.....	141
5.7 Use of Multi-Objective Particle Swarm Optimisation in Calibration of	
Reservoir Models	146
5.8 Field Application Test 1: IC Fault Model	146

5.9	Field Application Test 2: PUNQ-S3 Model.....	154
5.10	Chapter Summary.....	169
Chapter 6 – Advanced MCMC Techniques for History Matching Uncertainty		
Quantification – Part I: Hamiltonian Monte Carlo		
6.1	Inverse Problems: A Bayesian Perspective.....	172
6.2	Markov Chain Monte Carlo (MCMC) Methods	174
6.3	Hamiltonian Monte Carlo (HMC).....	177
6.4	Numerical Experiments.....	184
6.5	Computation of the Gradient.....	195
6.6	HMC Procedure Recap	197
6.7	Numerical Example Test.....	205
6.8	HMC Sampling Strategies	206
6.9	HMC Application in Reservoir Modelling	209
6.10	Chapter Summary.....	212
Chapter 7 – Comparison of Stochastic Sampling Algorithms for History Matching and Uncertainty Quantification.....		
7.1	Field Application Test 1: The Teal South Reservoir	218
7.2	Field Application Test 2: The IC Fault Model.....	219
7.3	Chapter Summary.....	230
Chapter 8 – Advanced MCMC Techniques for History Matching and Uncertainty Quantification – Part II: Population MCMC Methods.....		
8.1	Population Markov Chain Monte Carlo (Pop–MCMC)	240
8.2	Numerical Experiments.....	241
8.3	Petroleum Field Application	247
8.4	Chapter Summary.....	252
Chapter 9 – Brugge Reservoir Model History Matching: Comparison of Particle Swarm Optimisation and Ensemble Kalman Filter		
9.1	Brugge Reservoir	268
9.2	Parameterisation.....	269
9.3	History Matching Application – Brugge Reservoir	273
9.4	Chapter Summary.....	279
Chapter 10 – Summary and Conclusions		
10.1	Thesis Summary.....	306
10.2	Thesis Contributions	307
10.3	Overall Conclusion.....	312
10.4	Future Perspectives	312
Appendix A – Brugge Reservoir History Matching Results.....		316
Appendix B – Brugge Reservoir Best Fitting Model Results		332
References		338

List of Publications from Thesis

- Mohamed, L., Calderhead, B., Filippone, M., Christie, M., and Girolami, M. (2011) Population MCMC methods for history matching and uncertainty quantification. *Computational Geosciences*, to appear.
- Mohamed, L., Christie, M., and Demyanov, V. (2010) Comparison of stochastic sampling algorithms for uncertainty quantification. *SPE Journal*, SPE 119139–PA, 15(1), 31–38.
- Mohamed, L., Christie, M., and Demyanov, V. (2011) History matching and uncertainty quantification: Multiobjective particle swarm optimisation approach. In: Proceedings of the SPE EUROPEC 2011, SPE 143067, 23–26 May, Vienna, Austria.
- Mohamed, L., Christie, M., Demyanov, V., Robert, E., and Kachuma, D. (2010) Application of particle swarms for history matching in the Brugge reservoir. In: Proceedings of the SPE Annual Technical Conference and Exhibition (ATCE), SPE 135264, 19–22 September, Florence, Tuscany, Italy.
- Mohamed, L., Calderhead, B., Filippone, M., Christie, M., and Girolami, M. (2010) Population MCMC methods for history matching and uncertainty quantification. In: Proceedings of the 12th European Conference on the Mathematics of Oil Recovery, B012, 6–9 September, Oxford, UK.
- Mohamed, L., Christie, M., and Demyanov, V. (2010) Reservoir model history matching with particle swarms: Variants study. In: Proceedings of the Oil and Gas India Conference and Exhibition (OGIC), SPE 129152, 20–22 January, Mumbai, India.

Mohamed, L., Christie, M., and Demyanov, V. (2009) Comparison of stochastic sampling algorithms for uncertainty quantification. In: Proceedings of the SPE Reservoir Simulation Symposium, SPE 119139, 2–4 February, The Woodlands, Texas, USA.

Hajizadeh, Y., Demyanov, V., Mohamed, L., and Christie, M. (2011) Comparison of evolutionary and swarm intelligence methods for history matching and uncertainty quantification in petroleum reservoir models. In Koeppen et al. (Eds.) *Intelligent Computational Optimisation in Engineering: Techniques and Applications*. Springer–Verlag.

List of Figures

Figure 1.1: The forward problem and the inverse problem	3
Figure 2.1: Effect of pressure drop causes to fluid expansion in material balance approach	9
Figure 2.2: Reservoir simulation approach	14
Figure 2.3: Compartmentalisation technique principle where gridblocks are grouped to create zones with constant permeability values (Source: Le Ravalec-Dupin (2005))	26
Figure 2.4: SVMs kernel trick, SVM does classification (c) or regression (e,f) or characterisation of a dataset in one-class SVM (d) (Source: (Loshchilov et al. (2010)).....	31
Figure 3.1: Likelihood function definition.....	36
Figure 3.2: Number of papers on history matching prepared each year for SPE conferences and journals (Source: Oliver and Chen (2010)).....	38
Figure 3.3: NA optimisation workflow (Source (Erbas, 2007))	44
Figure 3.4: Bayesian framework for uncertainty quantification (Source: Christie et al., 2006)	46
Figure 3.5: NAB resampling (Source: Sambridge (1999b))	47
Figure 4.1: Velocity component construction	56
Figure 4.2: PSO common topologies	58
Figure 4.3: Boundary strategies for particle swarms	61
Figure 4.4: The Teal South Field Location	64
Figure 4.5: The Teal South 4500-ft sand structure map and the $11 \times 11 \times 5$ simulation grid	64
Figure 4.6: Production history for Teal South reservoir	65
Figure 4.7: Two different 2D projections of initial population of 30 randomly generated models in 8D parameter space	66
Figure 4.8: Misfit reduction and corresponding global best history match for variants	67
Figure 4.9: Misfit reduction after removing two outlier points (the 3rd and 4th).....	69
Figure 4.10: Random strategy sampling history for each inertial weight choice.....	70
Figure 4.11: Reflecting strategy sampling history for each inertial weight choice	71
Figure 4.12: Damping strategy sampling history for each inertial weight choice	72
Figure 4.13: Absorbing strategy sampling history for each inertial weight choice	73
Figure 4.14: Random boundary strategy measures	75
Figure 4.15: Reflecting boundary strategy measures	76
Figure 4.16: Damping boundary strategy measures.....	77
Figure 4.17: Absorbing boundary strategy measures.....	78
Figure 4.18: Bayesian credible intervals generated with random strategy variants.....	81
Figure 4.19: Bayesian credible intervals generated with reflecting strategy variants.....	82
Figure 4.20: Bayesian credible intervals generated with damping strategy variants	83
Figure 4.21: Bayesian credible intervals generated with absorbing strategy variants	84
Figure 4.22: Diversity per generation and relative uncertainty.....	85
Figure 4.23: Misfit reduction for variants	87
Figure 4.24: Diversity per generation and relative uncertainty.....	88
Figure 4.25: Bayesian credible intervals of total recovery prediction for PSO variants.....	89
Figure 4.26: IC Fault Model (2010) (Tavassoli et al., 2004)	91

Figure 4.27: IC Fault model truth case production data (oil and water rates)	91
Figure 4.28: 2D saturation map for the IC Fault model truth case	91
Figure 4.29: The truth value cross-section along each parameter.....	92
Figure 4.30: Database models with $M \leq 25$	93
Figure 4.31: Initial population of 20 starting points obtained with Latin Hypercube Sampling in 3D parameter space.....	94
Figure 4.32: LHS for two parameters with normal distribution. For each parameter the cumulative probability is split into five equal strata, and a random sample is drawn at each strata. Five samples from each parameter are then paired randomly forming a Latin square (Source: Minasny and McBratney (2006)).....	95
Figure 4.33: Random strategy sampling history for each inertial weight choice.....	98
Figure 4.34: Reflecting strategy sampling history for each inertial weight choice	99
Figure 4.35: Damping strategy sampling history for each inertial weight choice	100
Figure 4.36: Absorbing strategy sampling history for each inertial weight choice	101
Figure 4.37: 3D view – random and reflecting boundary strategies	102
Figure 4.38: 3D view – damping and absorbing boundary strategies.....	103
Figure 4.39: Random boundary strategy measures	104
Figure 4.40: Reflecting boundary strategy measures.....	105
Figure 4.41: Damping boundary strategy measures.....	106
Figure 4.42: Absorbing boundary strategy measures.....	107
Figure 4.43: Random boundary strategy measures – generational diversity	109
Figure 4.44: Reflecting boundary strategy measures – generational diversity	110
Figure 4.45: Damping boundary strategy measures – generational diversity.....	110
Figure 4.46: Absorbing boundary strategy measures – generational diversity.....	111
Figure 4.47: Bayesian credible intervals generated with the database.....	112
Figure 4.48: Bayesian credible intervals generated with random strategy variants.....	113
Figure 4.49: Bayesian credible intervals generated with reflecting strategy variants...	114
Figure 4.50: Bayesian credible intervals generated with damping strategy variants	115
Figure 4.51: Bayesian credible intervals generated with absorbing strategy variants ..	116
Figure 4.52: Bayesian credible intervals of total recovery prediction for PSO variants	117
Figure 5.1: Matching one objective worsens the other when solution perturbed	122
Figure 5.2: Pictorial view of dominance concept (Source: (Engelbrecht, 2005)).....	126
Figure 5.3: Pictorial view of ϵ -dominance concept (Source: (Engelbrecht, 2005)).....	127
Figure 5.4: The parameter space and the corresponding objective space pictorial (Source: (Deb, 2009)).....	128
Figure 5.5: For each particle's niche, the particle whose niche is less crowded is favoured (Source: (Reyes and Coello, 2006)).....	130
Figure 5.6: Use of ϵ -dominance in an external archive (Source: (Reyes and Coello, 2006))	134
Figure 5.7: Crowding distance computation for two objective functions. Particles with a larger value are preferred	138
Figure 5.8: Result of example test 1	142
Figure 5.9: Result of example test 2	143
Figure 5.10: Result of example test 3.....	144
Figure 5.11: Result of example test 4.....	145
Figure 5.12: Sampling history of SOPSO and MOPSO	148
Figure 5.13: SOPSO vs. MOPSO Sampling	148
Figure 5.14: Pareto front for MOPSO1 and MOPSO2	149
Figure 5.15: Convergence speed of SOPSO approach with a swarm of 50 particles for 40 generations using 20 seeds – median of M around 0 at iteration 28.....	151

Figure 5.16: Convergence speed of MOPSO approach with a swarm of 50 particles for 40 generations using 20 seeds – median of M around 0 at iteration 12.....	151
Figure 5.17: Best history matches for SOPSO and MOPSO approaches	152
Figure 5.18: Comparison of Bayesian credible intervals for a single run of each scheme – database vs. SOPSO and MOPSO	153
Figure 5.19: Prediction uncertainties derived from the database, SOPSO, and MOPSO	154
Figure 5.20: Top structure map of PUNQ-S3 (Floris et al., 2001)	155
Figure 5.21: Well 1 pressure for the best history matched model	159
Figure 5.22: Well 15 pressure for the best history matched model	159
Figure 5.23: Well 1 GOR for the best history matched model	160
Figure 5.24: Well 11 watercut for the best history matched model.....	160
Figure 5.25: Pareto front for MOPSO.....	161
Figure 5.26: Sampling history for SOPSO.....	163
Figure 5.27: Sampling history for MOPSO – choice 1	163
Figure 5.28: Sampling history for MOPSO – choice 2.....	164
Figure 5.29: Speed of convergence for SOPSO approach with a population of 30 individuals for 100 generations – 20 runs.....	164
Figure 5.30: Speed of convergence for MOPSO – choice 1 approach with a population of 30 individuals for 100 generations – 20 runs.....	165
Figure 5.31: Speed of convergence for MOPSO – choice 2 approach with a population of 30 individuals for 100 generations – 20 runs.....	165
Figure 5.32: FOPT Bayesian credible intervals for SOPSO: P90–P10 = 201.0 STB... ..	166
Figure 5.33: FOPT Bayesian credible intervals for MOPSO-choice1: P90–P10 = 407.5 STB	166
Figure 5.34: FOPT Bayesian credible intervals for MOPSO-choice2: P90–P10 = 297.5 STB	167
Figure 5.35: Total Recovery for the SOPSO, MOPSO-choice1, and MOPSO-choice2	167
Figure 5.36: CDF for FOPR.....	168
Figure 5.37: CDF for FOPT	169
Figure 6.1: Different sampling conditions affected an infill well decision (Erbas and Christie, 2007a).....	173
Figure 6.2: Metropolis-Hastings random walk behaviour using simple Gaussian proposals to navigate 2D Gaussian target distribution (Choo, 2000)	183
Figure 6.3: Dynamic representation for choosing adapted Gaussian proposals	183
Figure 6.4: The HMC steps algorithm for $k = 1$ iteration.	189
Figure 6.5: 3D plot for the probability distribution of interest $\pi(p_1, p_2)$ (a), position versus its augmented momentum coordinate (b), and HMC samples (c)	195
Figure 6.6: 3D plot for the probability distribution of interest $\pi(x, y)$ (a), with a 3D plot magnified in (b), and the 2D plot for one side of distribution (c).....	196
Figure 6.7: HMC method uses leapfrog simulation where in (a) three accepted states are drawn and in (b) two accepted states are drawn	197
Figure 6.8: Samples drawn in (a) 3D plot, (b) 3D larger scale plot, and (c) 2D plot ...	197
Figure 6.9: In the learning from example scheme, we learn a function f from input-output pairs (x_i, M_i) called the training set.	199
Figure 6.10: Gaussian kernel centred at 10.....	201
Figure 6.11: Pattern layer: consists of N kernels – one for every available data point.	202
Figure 6.12: GRNN estimate based on $N = 3$ samples with kernel width $\sigma = 3$	202
Figure 6.13: Different kernel widths change controlling the smoothness of the surface	203

Figure 6.14: The HMC-GRNN algorithm steps for $k = 1$ iteration (1 forward simulation run).....	206
Figure 6.15: (a) Initial training dataset and (b) cross-validation error vs. kernel width.....	207
Figure 6.16: GRNN surface using the 36 kernels with kernel width = 0.04.....	207
Figure 6.17: GRNN surface using the 1108 kernels and kernel width = 0.01.....	208
Figure 6.18: GRNN surface and the gradient vectors in red.....	208
Figure 6.19: HMC samples using grad GRNN shown in pink pluses (a) 3D plot, (b) 3D larger scale.....	209
Figure 6.20: Three possible MCMC strategies for obtaining nine samples in a fixed amount of computer time.....	210
Figure 6.21: HMC Samples drawn for two runs.....	211
Figure 6.22: HMC samples for Rastrigin function drawn for 2 runs (b) Run 1, and (c) Run 2.....	212
Figure 6.23: Initial population composed of 50 models generated randomly.....	213
Figure 6.24: All models generated.....	214
Figure 6.25: HMC samples generated.....	214
Figure 6.26: HMC samples generated – forecast period.....	214
Figure 6.27: Bayesian credible intervals for Run 1.....	215
Figure 6.28: Bayesian credible intervals for Run 2.....	215
Figure 6.29: Bayesian credible intervals for Run 3.....	216
Figure 7.1: Comparison of the best history matches (left) and the corresponding permeability estimates (right) obtained from NA and PSO.....	220
Figure 7.2: Two alternative sets of parameter values providing almost equivalent match qualities to the maximum likelihood model.....	221
Figure 7.3: Evolution of the mean generational minimum misfit for NA and PSO.....	222
Figure 7.4: Sampling history of NA for each of the 8 unknown parameters.....	223
Figure 7.5: Sampling history of PSO for each of the 8 unknown parameters.....	224
Figure 7.6: Sampling history of HMC for each of the 8 unknown parameters.....	224
Figure 7.7: Autocorrelation of HMC for each of the 8 unknown parameters.....	226
Figure 7.8: Autocorrelation length of HMC for each of the 8 unknown parameters.....	227
Figure 7.9: Power spectrum, $P(k)$, of the 8 unknown parameters obtained from the HMC.....	227
Figure 7.10: Marginalised distributions of the 8 unknown parameters Teal South model obtained from the HMC.....	228
Figure 7.11: Bayesian credible intervals generated by NA.....	229
Figure 7.12: Bayesian credible intervals generated by PSO.....	229
Figure 7.13: Bayesian credible intervals generated by HMC.....	229
Figure 7.14: Relative uncertainty of the three methods.....	229
Figure 7.15: Bayesian credible intervals for total recovery.....	229
Figure 7.16: Cumulative Distributions from NA, PSO and HMC at (left) and after (right) the end of history matching.....	230
Figure 7.17: Best history matches for all methods.....	232
Figure 7.18: Particle swarm and neighbourhood optimisers' ensembles.....	233
Figure 7.19: Sampling history for two variants of PSO and NA.....	234
Figure 7.20: Stochastic sampling algorithms ensembles.....	234
Figure 7.21: Bayesian credible intervals of oil and water rates and total recovery prediction for PSO and NA variants.....	237
Figure 7.22: Bayesian credible intervals for total recovery prediction PSO and NA variants.....	238
Figure 8.1: A sequence of 320 million realisations shows misfits correlation length of over 100 million iterations (Liu and Oliver, 2003).....	241

Figure 8.2: Representation of Population MCMC where π_N is the prior distribution with the highest temperature and π_1 the posterior distribution with the lowest temperature (Kim et al., 2009)	245
Figure 8.3: Pop-MCMC application on example test 1	248
Figure 8.4: Pop-MCMC application on example test 2	248
Figure 8.5: Pop-MCMC application on 2-parameter Rastrigin function	249
Figure 8.6: Pop-MCMC application on 3-parameter Rastrigin function	250
Figure 8.7: Multidimensional representation of complex biological system posterior distribution (a) and Metropolis Hastings sampling (b)	251
Figure 8.8: Population MCMC Sampling for biological system example (Source: Calderhead and Girolami (2009))	251
Figure 8.9: Pop-MCMC-500 sampling plot at various temperatures	255
Figure 8.10: Pop-MCMC-1500 sampling plot at various temperatures	256
Figure 8.11: DB ensemble in comparison to Pop-MCMC ensembles	256
Figure 8.12: Pop-MCMC parameter width per temperature at the end of sampling	256
Figure 8.13: Sampling history of the chains with the 10 temperatures – Pop-MCMC-500	257
Figure 8.14: Sampling history of the chains with the 10 temperatures – Pop-MCMC-1500	258
Figure 8.15: ACF for Temp10 temperatures sampled chains	259
Figure 8.16: Comparison of Bayesian credible intervals for oil rate (MSTB/D), water rate (MSTB/D), and cumulative oil produced (MSTB) – database vs. Pop-MCMC-500 and Pop-MCMC-1500	260
Figure 8.17: Comparison of history match quality and prediction quality guide	262
Figure 8.18: Forecast misfit (M_f) vs. history misfit (M_h): DB benchmark constrained in x and y ranges	262
Figure 8.19: Forecast misfit (M_f) vs. history misfit (M_h) for two runs for each technique	263
Figure 8.20: Prediction uncertainties derived from the database, NA1, NA2, PSO1, PSO2, Pop-MCMC-500, and Pop-MCMC-1500	264
Figure 8.21: The 1D marginal posterior distribution estimates for Pop-MCMC-500 ..	264
Figure 8.22: The 1-D marginal posterior distribution estimates for Pop-MCMC-1500	264
Figure 8.23: The 1-D marginal for DB	265
Figure 8.24: The calibration curve for weather forecasts (Source: Christie et al., 2005)	265
Figure 9.1: 3D Top structure map, OWC and fault (Source: Peters et al., 2010)	270
Figure 9.2: Principal Components in 2D	274
Figure 9.3: Simple example of how to form feature vector from 104 prior images with Linear PCA by applying a polynomial “kernel” of order 1 (see Section 9.2.1.1) ..	275
Figure 9.4: y_1 and y_2 correlations for multi-Gaussian y (Source: Sarma et al. (2008b))	277
Figure 9.5: Kernel trick idea in kernel PCA approach (Source: Sarma et al. (2008b)) ..	278
Figure 9.6: Kernel PCA principle (linear PCA in feature space)	279
Figure 9.7: The history matching workflow for the Brugge reservoir	280
Figure 9.8: Misfit reduction for PSO	282
Figure 9.9: Sampling evolution for the first 52 parameters	282
Figure 9.10: Sampling evolution for the 53-104th parameters	283
Figure 9.11: Well pressure history matches for producer 13 for EnKF (a) and PSO (b)	286

Figure 9.12: Well pressure history matches for producer 15 for EnKF (a) and PSO (b)	286
Figure 9.13: Well water production rate history matches for producer 2 for EnKF (a) and PSO (b)	286
Figure 9.14: Well water production rate history matches for producer 5 for EnKF (a) and PSO (b)	287
Figure 9.15: Well water production rate history matches for producer 17 for EnKF (a) and PSO (b)	287
Figure 9.16: Well water production rate history matches for producer 18 for EnKF (a) and PSO (b)	287
Figure 9.17: Recap of wells history matches for EnKF (a) and PSO (b)	288
Figure 9.18: NTG for the 9 layers for EnKF (a) and PSO (b)	289
Figure 9.19: Natural logarithm of permeability in X direction for the 9 layers for EnKF (a) and PSO (b)	290
Figure 9.20: OWC optimisation adding value to misfit reduction	292
Figure 9.21: OWC addition	292
Figure 9.22: Well pressure history matches for producer 5 for PSO with 104 parameters (a) and with OWC parameter added (b)	294
Figure 9.23: Well pressure history matches for producer 10 for PSO with 104 parameters (a) and with OWC parameter added (b)	294
Figure 9.24: Well pressure history matches for producer 15 for PSO with 104 parameters (a) and with OWC parameter added (b)	294
Figure 9.25: Well water production rate history matches for producer 5 for PSO with 104 parameters (a) and with OWC parameter added (b)	295
Figure 9.26: Well water production rate history matches for producer 6 for PSO with 104 parameters (a) and with OWC parameter added (b)	295
Figure 9.27: Well water production rate history matches for producer 10 for PSO with 104 parameters (a) and with OWC parameter added (b)	295
Figure 9.28: Well water production rate history matches for producer 16 for PSO with 104 parameters (a) and with OWC parameter added (b)	296
Figure 9.29: Field oil and water rates of the initial run of the ensemble and the observations	296
Figure 9.30: Field oil rate, field water rate, and total oil recovery of the two PSO runs – 104 and 105 parameters	297
Figure 9.31: OWC result for the 9 layers with NTG (a) and natural logarithm of permeability in X direction (b)	299
Figure 9.32: Misfit reduction using different number of parameters	300
Figure 9.33: Misfit reduction using 105 parameters, 40 parameters (with fixed OWC parameter), and 41 (including variable OWC parameter). The first 64 parameter values are set from the best fitting model obtained from the 105 parameter case	300
Figure 9.34: Misfit reduction using different swarm sizes	301
Figure 9.35: Misfit reduction comparison of KPCA1, KPCA2, and KPCA3	302
Figure 9.36: NTG for the 9 layers for KPCA2 (a) and KPCA3 (b)	303
Figure 9.37: Natural logarithm of permeability in X direction for the 9 layers for KPCA1 (a) and KPCA3 (b)	304
Figure A.1: EnKF p10-p50-p90 results	317
Figure A.2: EnKF p10-p50-p90 results	318
Figure A.3: EnKF p10-p50-p90 results	319
Figure A.4: EnKF p10-p50-p90 results	320
Figure A.5: EnKF p10-p50-p90 results	321
Figure A.6: PSO p10-p50-p90 results	322

Figure A.7: PSO p10-p50-p90 results	323
Figure A.8: PSO p10-p50-p90 results	324
Figure A.9: PSO p10-p50-p90 results	325
Figure A.10: PSO p10-p50-p90 results	326
Figure A.11: PSO p10-p50-p90 result – adding OWC parameter	327
Figure A.12: PSO p10-p50-p90 result – adding OWC parameter	328
Figure A.13: PSO p10-p50-p90 result – adding OWC parameter	329
Figure A.14: PSO p10-p50-p90 result – adding OWC parameter	330
Figure A.15: PSO p10-p50-p90 result – adding OWC parameter	331
Figure B.1: Porosity for the 9 layers for EnKF (a) and PSO (b).....	333
Figure B.2: Natural logarithm of permeability in Z direction for the 9 layers for EnKF (a) and PSO (b).....	334
Figure B.3: PSO with OWC addition result for the 9 layers with porosity (a) and natural logarithm of permeability in Z direction (b)	335
Figure B.4: Porosity for the 9 layers for KPCA2 (a) and KPCA3 (b)	336
Figure B.5: Natural logarithm of permeability in Z direction for the 9 layers for KPCA1 (a) and KPCA3 (b)	337

List of Tables

Table 2.1: Data required for a simulation study (Source: Fanchi (2001))	12
Table 2.2: Porosity values depend on rock type. (Source: Fanchi (2001))	12
Table 3.1: Comparison of advantages and disadvantages of five history matching methods (adapted and modified from Source: (Oliver and Chen, 2010)).....	51
Table 4.1: Acronyms used to denote the PSO variants	62
Table 4.2: Parameterisation and prior ranges for the Teal South model.....	64
Table 4.3: Measures summary – 10 Seeds	74
Generational minimum.....	76
Table 4.4: Measures summary – 10 Seeds	79
Table 4.5: Best history and forecast misfit values of PSO variants	80
Table 4.6: Measures summary – 1 Seed.....	86
Table 4.7: Measures summary – 1 Seed.....	89
Table 4.8: Parameterisation for the IC Fault model and prior ranges	90
Table 4.9: Measures summary per strategy – 10 Seeds	108
Table 4.10: Measures summary per inertial weight choice – 10 Seeds	109
Table 4.11: Measures summary – 1 Seed	117
Table 4.12: Measures summary – 1 Seed	117
Table 5.1: MOPSO variants (adapted from source: Reyes and Coello (2006)).....	136
Table 5.2: SOPSO and MOPSO algorithm setup	146
Table 5.3: Results of the performance of SOPSO and MOPSO with a population of 50 particles	147
Table 5.4: Summary of best 10 non-dominated elements (ranked by global misfit out of 179) stored in the archive and their respective objectives for MOPSO1	149
Table 5.5: Best model history and forecast misfits	152
Table 5.6: Model uniform prior ranges	156
Table 5.7: Algorithms parameter setups for PUNQ-S3	158
Table 5.8: Results of the performance of SOPSO and MOPSO with a population of 30 particles for 100 iterations.....	158
Table 5.9: Summary of best 10 non-dominated elements stored in the archive and their respective objectives (MOPSO-choice 1)	161
Table 5.10: Summary of best 10 non-dominated elements stored in the archive and their respective objectives (MOPSO-choice 2)	162
Table 6.1: HMC algorithm parameters	210
Table 6.2: HMC algorithm parameters	212
Table 6.3: HMC algorithm parameters	213
Table 7.1: Best history and forecast misfits for all stochastic algorithms	221
Table 7.2: Statistical measures	226
Table 7.3: Best misfits values obtained with the three stochastic methods and their corresponding parameter values.....	231
Table 9.1: Fluid properties in the Brugge Field (Source: Geel, 2008).....	271
Table 9.2: Stratigraphy used in the Brugge Field with the main characteristics (Source: Geel, 2008).....	272
Table 9.3: Best misfits achieved both with and without adding OWC parameter	292
Table 9.4: Best misfits achieved with two swarm sizes – 105 parameters	301

Chapter 1 – Introduction

1.1 Reservoir Model History Matching as an Inverse Problem

To conduct analysis and attain an understanding of subsurface system such as oil flows underground in oil reservoirs, aquifers underground porous media, and waste deposits, a numerical simulation model is required. The simulation model is constructed to characterise the spatial correlations of the physical properties such as porosities, permeabilities and facies that govern the flows of fluid in porous media. The models are then used to make approximations of reserves and fluid displacements.

Simulation models are three dimensional mathematical illustration of the reservoir involving millions of gridcells. Each gridcell centre is given values for their respective physical property. The data are noisy, limited, obtained from sparse precise locations, and unreliable. Yet, the conceptual model of geological history is needed to make models of the entire reservoir and utilise these models to determine the value of the reservoir by predicting the size, shape and the amount of oil or gas that the field may produce.

All available data then has to be integrated and modelled in 3D to simulate the flow. There are several sources of information that are visualised in different computerised platforms and which participate in constructing the reservoir model. The data will include the static data, which refers to the hard data coming from the wells and is gathered ahead of production, and the dynamic data, which refers to the production data recorded after the start of production and is called the historical data. Constructing a reservoir simulation model that is consistent with historical dynamic data is called the history matching problem.

History matching is an important and challenging task in any reservoir engineering study, even though solving this problem is not the ultimate objective but rather the prediction of oil or gas that will be produced. The aim of history matching is to adjust unknown reservoir properties such as permeabilities and porosities in order to obtain a model that can be utilised to reproduce a match with historical production data in a minimum manner observed at wells. Forecasts are then made on the basis of the matched model. The anticipation is that incorporating all the information will deliver reservoir models that are more predictive and reliable, and thus have more confidence in the predicted uncertainty. It is known that uncertainty pervades all phases of reservoir modelling and therefore statistical analysis is essentially required to account for different scenarios.

History matching problem is a difficult task for two reasons. Firstly, there are a large number of uncertain parameters that have to be estimated. Secondly, there is a nonlinear relationship between the model parameters and the model misfit surface which is a least square formula consisting of the sum of squared differences between model response and simulated data. The nonlinearity of the problem leads to the presence of multiple local minima while the linear least squares problem one would have a parabolic curve structure, convex and a closed-form solution that is unique, given that the number of data points used for fitting equals or goes over the number of unknown parameters, apart from special degenerate circumstances, thus leading to a single minimum. The existence of multiple minima means that global optimisation methods may get entrapped in local minima in which case the model solution is not the global minimum. In other words, different values of the model parameters may be consistent with the data. History matching problem is known to belong to a family of mathematical problems referred to as inverse problems (Tarantola, 1987). In Figure 1.1 the forward problem represents finding the result of a given model. Inverse problems, by contrast, of which history matching is considered a member, involve finding a model for a given output. Solving inverse problems has challenges due to ill-posedness that may occur (Sun, 1994). A well-posed problem as introduced by Jacques Hadamard should meet all the following conditions

- A solution should exist,
- The solution should be unique,
- The solution should continuously depend on the data.

If any of these conditions is not met, the problem is called ill-posed. Usually the third condition is hard to meet. In history matching special case there is no unique solution. Furthermore, solutions are very sensitive to small perturbations in input data, particularly noise that may force large errors in the solution. First efforts to solve this problem were initiated by Nelson (1960) and Jacquard and Jain (1965).

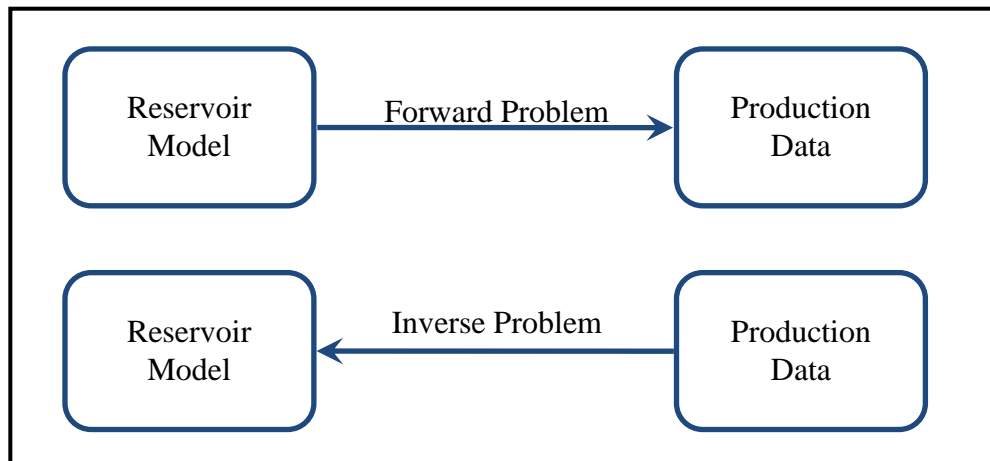


Figure 1.1: The forward problem and the inverse problem

In brief, history matching and uncertainty quantification are important and challenging tasks in reservoir engineering studies. The aim of history matching methods is to obtain a match with historical production data. An ensemble of good fitting models can then be used to quantify the uncertainty of the forecasts. The quality of such forecasts is affected by the selection of the estimated reservoir properties, as well as the accuracy of the reservoir model itself.

1.2 Thesis Objectives

The goal of this thesis is to investigate and develop reliable techniques that can produce models, capable of navigating the parameter space quickly, able to overcome local minima by avoiding entrapment, and have a high probability of locating all optimal regions of the parameter space without exhaustive sampling of the parameter space. Achieving this goal will improve confidence in our predictions in petroleum industry in a faster and more reliable way. The thesis has the following specific objectives:

- To explore new developments of novel state-of-the-art promising optimisation and inference methods from computer science background such as Particle Swarm Optimisation with its different variants.

- To optimally balance the different objectives (individual least squared differences for all production data types) simultaneously while maintaining solutions diversity, and to examine the trade-offs, called Pareto optimal front. This requires investigation of innovative Multi-Objective Optimisation schemes in the field of petroleum engineering.
- To inspect Hamiltonian Monte Carlo and to introduce new Population MCMC in petroleum engineering.
- To compare these developed methods with the population-based Neighbourhood Algorithm and study the impact of different methods on history matching
- To investigate these techniques on simple and synthetic examples and examine the influence of sampling performance of the different methods
- To study how to handle complex nonlinear response surfaces with these methods
- To tackle the inverse history matching problem with both a huge number of unknown parameters and very costly forward simulations runs that hinder the use of global optimisation algorithms.
- To examine the Particle Swarm Optimisation for addressing large problems and compare the results with the Particle Filter Data Assimilation method.

1.3 Thesis Structure

The outline of this thesis is as following

This chapter has introduced the parameters estimation, inverse history matching and uncertainty quantification problems. Thus, assisted history matching and inference methods will be the theme of the thesis.

Chapter 2 presents a review of the literature on reservoir modelling fundamentals including reservoir performance prediction techniques and a review of recent research on reservoir characterisation and modelling topics.

Chapter 3 reviews history matching and uncertainty quantification concepts and techniques. The Bayesian methodology and other topics set the basis for the next chapters.

Chapter 4 introduces the novel Particle Swarm Optimisation (PSO) for solving nonlinear ill-posed reservoir history matching problems. The chapter starts with definitions of the main components of the algorithm which serves as the

foundation for understanding the technique. The main adaptations and variants investigated in this thesis are presented. The technique is applied to two petroleum examples. Comparisons of the performance of the different variants of the method are studied for better employment on large problems.

Chapter 5 extends the application of Particle Swarm Optimisation algorithm to handle multi-objective optimisation in reservoir history matching context. The chapter introduces Multi-Objective Particle Swarm Optimisation (MOPSO). The application of the technique to two challenging synthetic petroleum examples is shown.

Chapter 6 reviews basic concepts and then introduces the Hamiltonian Monte Carlo (HMC) algorithm for uncertainty quantification. The key elements of the method and implementations are discussed. Analysis is drawn for some numerical examples, and then an application of the method on reservoir example is demonstrated.

Chapter 7 investigates the efficiency of three stochastic sampling algorithms for generating history matched reservoir models: Hamiltonian Monte Carlo algorithm, Particle Swarm Optimisation algorithm, and the Neighbourhood Algorithm. The comparative analysis is presented for the two case studies. The effects of the different sampling methods are examined and analysed.

Chapter 8 presents the application of Population MCMC (Pop-MCMC) method to history matching and uncertainty quantification. The technique is tested on analytical examples followed with application on the IC Fault model. Comparisons with other methods are shown.

Chapter 9 focuses on the handling of large history matching problems with the Particle Swarm Optimisation on more realistic large Brugge field (a recent SPE benchmark case study) with the insights gained from previous chapters. The history matching results are compared with the ones obtained with Ensemble Kalman Filter (EnKF) data assimilation method. A comparative analysis and details of the results are then provided.

Chapter 10 concludes with the main research contributions and achieved results of the thesis. The thesis has suggested a number of areas that require further investigation and identified future research directions.

Chapter 2 – Reservoir Simulation and Characterisation: Literature Review

The present chapter provides a review of the literature on reservoir modelling fundamentals including reservoir performance prediction techniques and a review of recent progress on reservoir characterisation and modelling topics. A demonstration of the background for reservoir simulation is given.

2.1 Reservoir Analysis Techniques

The responsibilities associated with fundamental reservoir analysis present information that is needed to prepare input data for a simulation study. These tasks include volumetric and reservoir performance prediction techniques. Fluid volumes in a reservoir are values that can be obtained from multiple sources and therefore serve as quality control. For instance geoscientists use static information to determine volume in a procedure that is referred to as volumetric analysis (Fanchi, 2001). Material balance and reservoir simulation approaches use dynamic data to obtain the same information. An accurate reservoir characterisation should obtain consistent estimates of the initial reservoir fluid volumes in place regardless of the approach selected to find out the fluid volumes.

2.1.1 Volumetric Analysis

The equation for volumetric estimates of initial oil and gas in place is given by Eq. (2.1) (Craft et al., 1991; Dake, 1978; Fanchi, 2001, 2006).

$$N = \frac{7758\phi Ah_r S_{ri}}{B_{ri}} [\text{STB}] \quad \text{Eq. (2.1)}$$

where N is the initial oil/gas in place, ϕ is the reservoir porosity, A is the reservoir area, h_r is the net thickness of oil/gas zone, S_{ri} is the initial reservoir oil/gas saturation, and B_{ri} is the initial oil/gas formation volume factor. To convert from acre-feet to stock tank barrels, the constant value of 7758 is used. An acre of reservoir 1 foot thick would include 7758 barrels of oil in a situation of 100% porosity, zero water saturation and no oil shrinkage.

2.1.2 Reservoir Performance Prediction Techniques

Reservoir performance prediction techniques can be used to determine a number of designs, operational and troubleshooting problems during all phases in the development of a field. Several techniques exist in the petroleum community for estimating the reservoir performance forecasts: material balance analysis, decline curve analysis and reservoir simulation.

2.1.2.1 Material Balance

Material balance describes material entering or leaving a physical system in which the law of conservation of mass and energy is the foundation for computations. The amount of material leaving a control volume is equal to the amount of material entering the volume minus the amount of material accumulated in the volume. This is demonstrated in Eq. (2.2) together with Figure 2.1 in which the volume balance is evaluated in the general form of the material balance for a hydrocarbon reservoir, provided below (Dake, 1978).

$$\begin{aligned} \text{withdrawal} = & \text{expansion of oil + originally dissolved gas} \\ & + \text{expansion of the gascap gas} \\ & + \text{reduction in hydrocarbon pore volume due to connate water} \\ & \quad \text{expansion} \\ & + \text{reduction in pore volume due to rock compressibility} \\ & + \text{aquifer influx} \end{aligned}$$

Eq. (2.2)

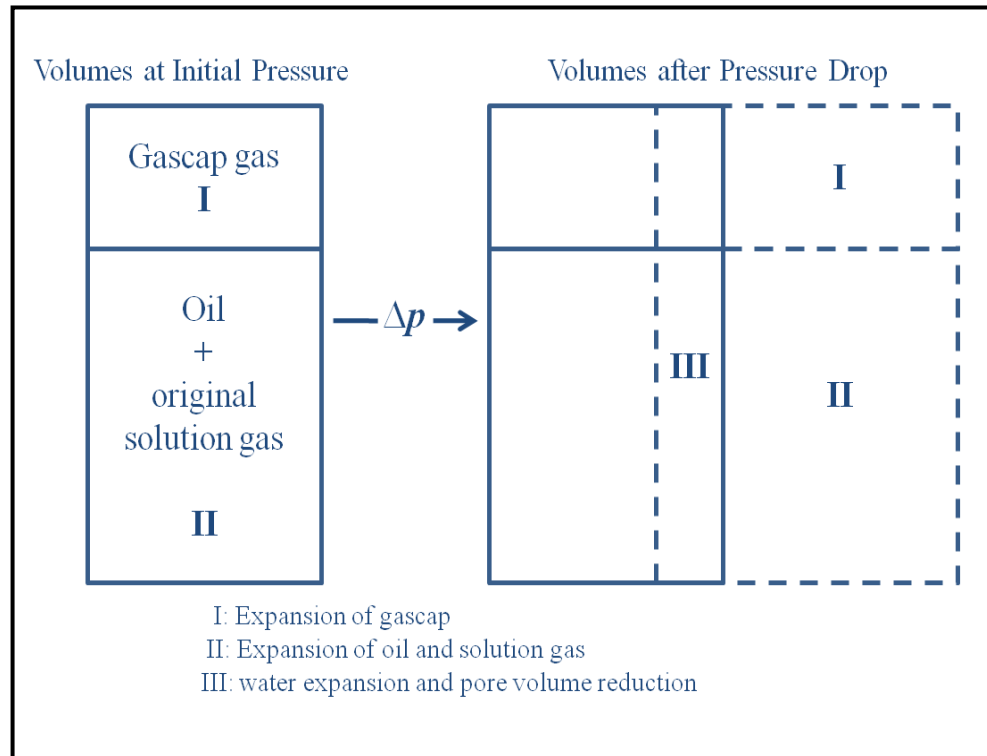


Figure 2.1: Effect of pressure drop causes to fluid expansion in material balance approach

Material balance calculations treat the reservoir as a large tank of material and uses measurable quantities to investigate the amount of a material that cannot be directly measured such as the volume of hydrocarbons in place where measurable quantities include cumulative fluid production volumes for oil, water, and gas phases, accurate reservoir pressures measured over time and fluid property data from samples of produced fluids. Several goals can be achieved with material balance as it provides an independent approach of estimating the volume of oil, water, and gas in the reservoir for comparison with volumetric estimates. The magnitude of different factors in the material balance equations indicates the relative contributions of different reservoir working drive mechanisms. Material balance can be used to predict future reservoir performance and assist in estimating cumulative recovery efficiency. Derivation of the material balance equation and discussions of these subjects can be found in Craft et al. (1991), Crichlow (1977), Dake (1978), and Mattax and Dalton (1990). Since material balance ignores the reservoir heterogeneity it has a limited prediction performance and more sophisticated approaches like reservoir simulation need to be utilised alongside for more reliable predictions.

2.1.2.2 *Decline Curve Analysis*

Decline Curve Analysis is a graphical technique for estimating the reservoir's ultimate oil or gas recovery and predicting production in oil reservoirs and oil fields. The decline curve depicts how a cumulative production curve decreases as a function of time, usually as a result of loss of reservoir pressure or the changing relative volumes of the produced fluids. The basis for the decline curve analysis concept is fitting a line through the reservoir performance history and assuming this same line will take a similar trend into the future forms. The curve is extrapolated to an end point. However no pressure data can be extrapolated (Arps, 1945; Arnold, 2008; Fanchi, 2001).

Historically, Arps (1945, 1956) collected these ideas into a comprehensive set of line equations defining three curves. Later, Fetkovich (1980) developed a broad set of type curves to enhance the application of decline curve analysis. The advent of the computer revolutionised the decline curves analysis by making the process less time consuming.

Arps investigated the relationship between flow rate and time for producing wells. He employed the equation of a hyperbola to define three general equations to model production declines. These models are exponential, hyperbolic and harmonic equations. In order to locate a hyperbola in space the following three parameters are needed: the starting point on the production represented by the y-axis, the initial decline rate, and the degree of curvature of the line. The main decline curves analysis characteristics are:

- All production rate-time curves must trend in a downward behaviour.
- The semilog rate-time decline lines are curved for the hyperbolic and harmonic equations, while the exponential equation decline line is a straight line.
- The Cartesian rate-cumulative recovery plots are curved for the hyperbolic and harmonic cases, and a straight line for the exponential case.
- The semilog rate-cumulative production plots for the exponential and hyperbolic declines are curved, while the harmonic equation results in a straight line.

Decline curves generally tend to have limitations such as underestimating reserves, underestimating production rates, or overestimating reservoir performance. Even though, the Arps equations were developed in the mid-1900s, they are still used (Towler, 2002) despite their shortcomings. This is because data curves can still be used as an adequate useful analysis tool for more valid predictions in steady reservoirs when

no change is observed in reservoir properties, driving strategies, states of the wells, or the number of wells (MacKay, 1994).

2.1.2.3 Reservoir Simulation

Reservoir simulation is an essential and more powerful tool for reservoir management and forecasting the reservoir performance. Reservoir simulation entails solving the partial differential equations (PDEs) of heat and mass transfer which describe the flow of fluids in petroleum reservoirs numerically, subject to appropriate initial and boundary conditions. There are two ways to develop discrete equations either by writing balance equations directly over control volumes (or grid blocks) or by developing general PDEs and then introducing appropriate discretisation for a particular coordinate system. The mass balance principle is achieved in each gridlock by equating the accumulation of mass in the block with the difference between the mass leaving the block and the mass entering the block. The simulation model is different from the reservoir engineering material balance in the ability of the simulator to account for flow between blocks. The simulation model can be enlarged to include position-dependent effects by modifying the grid representing the reservoir architecture.

Any reservoir simulation study consists of two parts, the input data to be acquired and evaluated and the model. This data consists of the density at surface conditions, PVT relations (volume factors, viscosity), constant gas resolution factor, relative permeabilities as functions of water saturation, and water – oil capillary pressure. In the following we briefly highlight some of these data and a general list of the types of data that are required in a model study is given in Table 2.1 Fanchi (2001).

One of the most significant properties of rock that must be included in a reservoir model is porosity. Porosity (ϕ) is the fraction of a porous medium that is void space. Porosity values have dependence on rock type as shown in Table 2.2. Two basic techniques for measuring porosity are core analysis in the laboratory and well logging.

Table 2.1: Data required for a simulation study (Source: Fanchi (2001))

Property	Source
Porosity, rock compressibility	Core analyses, Well logs
Permeability	Pressure transient tests, Core analyses, Correlations, Well performance
Saturations	Well logs, Core analyses, Pressure cores, Single-well tracer tests
Relative permeability and capillary pressure	Laboratory core flow tests
Fluid property (PVT) data	Laboratory analyses of reservoir fluid samples
Faults, boundaries, fluid contacts	Seismic, Pressure transient tests
Aquifers	Seismic, Material balance calculations, Regional exploration studies
Fracture spacing, orientation, connectivity	Core analyses, Well logs, Seismic, Pressure transient tests, Interference testing, Wellbore performance
Rate and pressure data, completion and work over	Field performance history

Table 2.2: Porosity values depend on rock type. (Source: Fanchi (2001))

Rock Type	Porosity Range (%)	Typical Porosity (%)
Sandstone	15–35	25
Unconsolidated sandstone	20–35	30
Carbonate		
– Intercrystalline limestone	5–20	15
– Oolitic limestone	20–35	25
– Dolomite	10–25	20

Permeability is a physical constant defined by Darcy's law describing flow in a given sample for a given fluid with a set of experimental conditions. It represents the ability of a rock to let fluid flow through it. Generally, permeability of oil reservoirs is of the order of hundreds of mD and that of an aquifer is of the order of a Darcy and it may contrast inside a single reservoir by several orders of magnitude. The Permeability distribution is usually highly non-symmetrical and is usually approximated by a lognormal distribution (Freeze, 1975). Permeability is a scalar if the medium is isotropic while it is a second order tensor if the medium is anisotropic. Thus, permeability has a directional component in which permeability may be larger in one direction than another in bed scale. Vertical permeabilities are usually assumed rather

than measured. A rule of thumb is assuming vertical permeability is approximately one tenth of horizontal permeability. These are reasonable assumptions when there is no available data to suggest the contrary (Fanchi, 2001). In the existence of some data, complex functions can be used to describe the relationship between vertical and horizontal permeabilities.

Multiphase flow combinations like, gas-oil, gas-water, and gas-oil-water, are modelled by including relative permeability curves in the simulator. Saturation end points for the relative permeability curves are used to establish initial fluid in place in addition to modelling fluid behaviour. Relative permeability curves represent flow mechanisms, such as drainage or imbibition, or fluid wettability. Each phase curve is used to calculate an effective permeability.

Reservoir models usually include capillary pressure data primarily to determine the initial fluid contacts, transitions zones and controlling the flow of fluid between the fracture and the rock matrix in fractured reservoir models. The relationship between capillary pressure and elevation is used to establish the initial transition zone in the reservoir.

2.2 Flow Equations

The general equations for describing fluid flow in a porous media include a dispersion term, a convection term, a source/sink term representing wells, the time varying accumulation term, and Darcy's law. Darcy's law is the basic elliptic equation relating the flow to the gradient of pressure and thus describing the fluid flow in a simulator. The establishment of Darcy's Law (Darcy, 1856) was based on the results of Henry Darcy's experiments on the flow of water through beds of sand. For single-phase flow, Darcy's equation as given in Eq. (2.3) states that the flow rate, q , is equal to cross-sectional area, A , times permeability, k , and pressure difference, ΔP , across distance length, L , and is inversely proportional to the viscosity of fluid, μ . The direction of flow is opposite to the direction of increasing pressure; fluids flow from high pressure to low pressure in a horizontal (gravity-free) system. A one-dimensional single phase flow horizontal form is shown in Figure 2.2. In Darcy's law the fluid flow rate is proportional to pressure gradient.

$$q = \frac{kA}{\mu} \frac{\Delta P}{L}; \quad \Delta P = P_2 - P_1; \quad P_2 > P_1 \quad \text{Eq. (2.3)}$$

Dividing the flow rate, q , in the differential form by the cross-section area of the rock sample, A , leads to Darcy or filtration velocity as written in Eq. (2.4). Darcy velocity represents the volume of fluid that perpendicularly crosses a surface unit of porous medium per time unit.

$$u = -\frac{k}{\mu} \text{grad}(P) \quad \text{Eq. (2.4)}$$

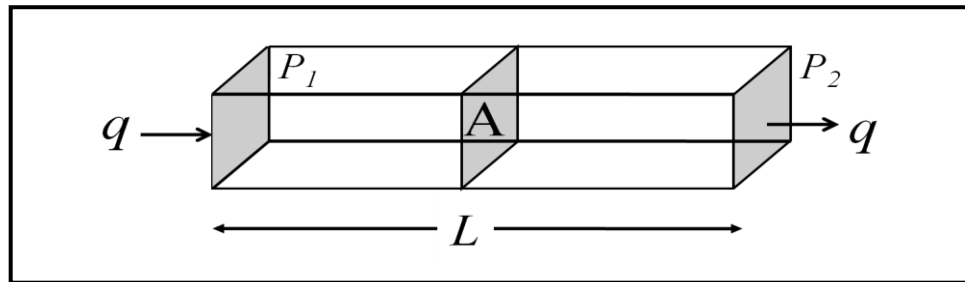


Figure 2.2: Reservoir simulation approach

2.2.1 Simulator Selection

The selection of a reservoir simulator depends on factors like the objective of the study, fluid type, and dimensionality of the system. Standard black oil and compositional simulators assume isothermal flow and mass transport is instantaneous through a gridcell. The most reservoir simulators assume reservoir temperature is constant throughout the life of the field and the equilibrium is established instantaneously which usually are sounded assumptions. Hence, temperature gradients or the time needed for a mixture to get to equilibrium is not considered in the majority of simulators. A black oil simulator represents oil, water, and gas phases, while, a compositional simulator represents the fluid as a mixture of hydrocarbon components. Black oil simulator may be viewed as a compositional simulator with two components, where they can have gas dissolved in the oil phase as well as oil dissolved in the gas phase.

Constructed fluid flow equations are a set of nonlinear partial differential equations that must be solved with computer simulators. The partial derivatives can be substituted with finite differences which are derived from Taylor's series (Fanchi, 2006). The gridcell length is the spatial finite difference interval while the timestep is the temporal finite difference. The finite difference representations of the partial derivatives substituted in the original flow equations can be rearranged algebraically and then solved. The regular solution schemes are IMplicit Pressure-EXplicit Saturation (IMPES), Newton-Raphson and fully implicit advanced techniques. Detailed technical background can be found in

Aziz and Settari (1979), Fanchi (2006), Gerritsen and Durlofsky (2005), Mattax and Dalton (1990), Peaceman (1977), Rosenberg (1977), Thomas (1982), and Thomas and Thurnau (1983).

2.2.2 Streamlines Simulation

A method which is used for more accurate transport calculations is the streamline simulation. Streamline simulations have been in the literature since first application by Muskat and Wyckoff (1934) and have received much attention and popularity with an extensive discussion in the literature (Baker et al., 2002; Batycky et al., 1997; Blunt et al., 1996; Datta-Gupta, 2000; Datta-Gupta and King, 2007; King and Datta-Gupta, 1998; Lolomari et al., 2000; Thiele, 2001; Thiele, 2010; Thiele and Batycky, 2001). The streamline is a line or path that is everywhere tangent to the local velocity distribution at a given instant in time. In streamline simulations, the pressure equations are solved once using total mobilities and streamlines are calculated. Fluids are transported over a timestep along streamlines rather than from gridcell to gridcell as in conventional finite-difference approaches. Streamlines reflect the immediate velocity distribution. Consequently, fluids are advanced to move with the total velocity distribution, obtained from the newly obtained spatial pressure distribution, the static petrophysical description, and Darcy's law, along the streamlines until the velocity distribution is updated at a later time to account for its changing behaviour. The spatial distribution of the static petrophysical properties (e.g., permeability, porosity, and relative permeability regions) and the volumes produced/injected at the wells is reflected directly in the constructed geometry of the streamlines and the velocity at which fluids move along each individual streamline (Thiele, 2010). The approach can computationally be much faster than conventional finite-difference simulations when applied to large and heterogeneous models. Current streamline models are suitable for modelling tracer transport and waterflooding where the velocity field are fairly static and the streamlines require updating occasionally. In these circumstances streamline models can be orders of magnitude faster than conventional finite difference simulators (Datta-Gupta and King, 1995; Batycky et al., 1997). Because the pressure equations are only solved a small number of times during the simulation rather than possibly thousands of times. While this approach is less accurate than conventional simulations schemes, it is feasible to carry out streamlines simulations on the grids containing millions of blocks.

Streamline simulation also has advantages in its ability to screen highly detailed geological models, allowing the engineer to visualise fluid flow paths in the reservoir (e.g. FrontSim Software), identifying regions and parameters in the reservoir that need to be history matched, as well as its rapid production data integration, speed and versatility which have led to many novel applications. Streamline models are not considered as a replacement to grid-based simulators but can play an important role in bridging the gap between geologic modelling and flow simulation (Datta-Gupta, 2000).

2.2.3 Top-down Reservoir Modelling

The Top-Down Reservoir Modelling (TDRM) is an integrated approach developed by British Petroleum (BP) to incorporate historical data into reservoir simulation models to facilitate fast uncertainty investigation. The Top-Down philosophy is “*to start investigations with the simplest possible model and simulator appropriate for the business decision*”. (Williams et al. 2004).

In the TDRM approach a search is made over a large number of reservoir parameters, an uncertainty space with dozens of dimensions. The parameter values controlling properties in the simulation model are changed to match the observed data. These parameters are the same ones that might be adjusted in a manual history match, such as reservoir pore volumes and/or permeabilities, aquifer strength or fault transmissibilities (Walker and Lane, 2007).

2.3 Gridding and Upscaling Techniques

Several gridding techniques have been developed to build a simulation model (Arnold, 2008). Most models used in this thesis are Cartesian grids where the cell size is constant and the cells are oriented in the same general direction.

Reservoir models built with geostatistical techniques are an integration of several measurement scales. The gridcell of the geostatistical model inferred from well log data can be of order 10^3 m^3 in size while the gridcell size of the geostatistical model that has been inferred from core data may be of order 20 cm^3 . Thus, high level of resolution of reservoir models involves very large number of gridcells, and is computationally expensive usually for numerical flow simulators. This limitation increases with history matching since a significant number of flow simulations are often required. Fine gridcells are aggregated into coarse ones in order to reduce the model size. The

problem that arises is deciding on the equivalent transport properties (permeability and porosity) of the coarse gridblocks. The equivalent properties rely on the properties simulated at the fine scale ones. The procedure of converting fine scale into coarse scale one is called upscaling.

Permeability upscaling in hydrocarbon reservoirs is a challenge to engineers, because of the multiphase nature of the fluids. Upscaling of absolute permeability is easier and several methods ranging from simple averaging of the permeabilities to pressure solution methods have been developed (Renard and de Marsily, 1997). In Renard and de Marsily (1997), the various methods used to calculate the equivalent permeability of a heterogeneous reservoir are discussed in an excellent review.

Upscaling methods can be categorised with respect to the types of parameters being upscaled (single or multi-phase systems) and the approach in which these parameters are computed (local and global calculations) (Durlofsky, 2003; Farmer, 2002). In single-phase systems, the upscaled parameter is usually the absolute permeability, and the goal is to maintain the gross features of flow on the coarse grid. In multiphase systems (Christie, 2001) for instance two-phase upscaling of relative permeability and capillary pressure must also be considered (Christie, 1996; Sablok and Aziz, 2008). Available single-phase upscaling techniques include renormalisation technique (King, 1989, 1996; King et al., 1993, 1995), effective medium theory, pressure-solver methods, homogenization theory, and harmonic/arithmetic averaging methods (Christie, 1996). A comparison of the performance of different upscaling methods on a test problem, the 10th SPE Comparative Solution Project on Upscaling, is provided in Christie and Blunt (2001). Recent comparison between wavelet and renormalisation upscaling methods and adaptive iterative upscaling-downscaling approach is provided in Babaei and King (2011).

Pickup et al. (2004) suggested a new way of generating coarse-scale permeabilities without upscaling by history matching a two-phase flow simulation. By comparing coarse-scale permeabilities with the conventional upscaled permeabilities and the fine-scale results, the resulting errors obtained from the history matched coarse-scale permeabilities were close to the ones acquired with the fine-scale models.

Note that a history match in production data does not automatically mean a match in reservoir parameters, and a model that gives a good fit in one respect could do very badly in the other (Tavassoli et al., 2004).

2.4 Reservoir Characterisation and Modelling

Definition 2.1. Stationarity Assuming X is a variable with property value $x(u)$ at location u , X is stationary of order two when (1) the expected value $E(X(u))$ exists and invariant with the study area and (2) the variance and the covariance between $X(u)$ and $X(u+h)$ exist and depend only on the separation vector h . This is an empirical decision determined to be accepted or not based on the data.

Definition 2.2. Experimental (semi)variogram: measures the dissimilarity between data separated by vector h , named the *lag*. The semivariogram is the average squared difference between every data pair separated by h as in Eq. (2.5). Plotting $\gamma(h)$ vs. h , while taking each h in the same direction θ , pictures the increase in variability between two property values with increasing $|h|$ in that direction θ . The direction θ is characterised by two angles the azimuth and dip angles (Caers, 2005).

$$\gamma(h) = \frac{1}{2N(h)} \sum_{i=1}^{N(h)} (x(u) - x(u+h))^2 \quad \text{Eq. (2.5)}$$

The number of pairs is $n(n-1)/2$ for n data pairs. Once the semivariogram of the geological data has been determined, three main characteristics are inspected (Caers, 2005).

- **Sill:** the maximum value of the semivariogram for the parameter u . it is also the variance of the measured data. Usually, while the separation distance increases, the variogram increases reaching a plateau implying a stationarity. The asymptotic value achieved by the variance is called the sill. The presence of non-stationarity or multi-scale structure can be detected by certain behaviour of the variogram. For example the variogram will increase indefinitely or continually oscillate.
- **Range:** is an estimate of the maximum correlation length between two points at a defined separation distance, that is, the distance from which the variogram arrives at the sill asymptotically.
- **Nugget effect:** is the value of the semivariogram at zero lag. This discontinuity or vertical jump at the origin of the variogram is caused by the dissimilarity which is in turn due to the sampling errors and short scale variability at

extremely short distances. The term nugget derived from the observation that the lag for a finite size gold nugget has non-zero value.

Usually the sample variogram is approximated via a variogram model or the experimental variogram. The models usually used are nugget effect model, spherical, exponential, Gaussian, and power. In addition variograms are often built as linear combinations of basic models which characterises the nest model.

By using the available data, a prediction image for the reservoir, i.e. reservoir properties can be built. This is achieved through the use of kriging which is named in honour of the innovative work of Krige (1951). There are several kriging techniques including simple kriging, ordinary kriging, and kriging with a trend simply containing linear interpolation with weighting coefficient factors. In these techniques the variance of the kriging estimator is unbiased and its variance is as small as possible. Sometimes the kriging estimator is referred to as the Best Linear Unbiased Estimator (BLUE). The simple kriging estimator for example is a linear combination of the differences between data and mean obtained for each reservoir gridblock as shown in Eq. (2.6) where α_i is the weight which depends on the location u . The kriging weights depend on the variogram or the covariance matrix adapted by the sample variogram, the location of the sample gridblocks and the location of the gridblock to be estimated.

$$X_k(u) = m + \sum_{i=1}^n \alpha_i(u) [(X(u_i) - m)] \quad \text{Eq. (2.6)}$$

The major goal of estimating reservoir properties is to minimise the risk of estimates varying significantly from the unknown values. Nevertheless kriging smoothes the spatial variability and leads to a realisation whose histogram is not consistent with the sample realisation (Le Ravalec-Dupin, 2005).

Simulating reservoir properties techniques can be used to simultaneously generate a set of geological representations (realisations) constrained to static data. These include Cholesky decomposition method (Albert, 1987; Davis, 1987) and the sequential simulations algorithms (Chilès and Delfiner, 1999; Goovaerts, 1997; Strebelle, 2002). The generated realisations are equiprobable (having the same probability), and honour the sample histogram and variogram, and as kriging estimators, simulators reproduce all data values at data locations.

A commonly used simulation algorithm for modelling continuous realisations is based upon a sequential procedure called Sequential Gaussian Simulation (SGS) (Chilès and Delfiner, 1999; Goovaerts, 1997). In this method assuming the reservoir has N gridblocks, the sequential principle idea involves minimising the problem of simulating an N -dimensional random vector into a sequence of N univariate simulation problems. A random path is defined at first to visit each grid block only once. At each grid block, the mean and the variance of the Gaussian cumulative density function constrained to the known well data are determined as well as the values drawn for previously visited gridblocks to preserve the spatial structure of the data. Computations are carried out based on simple kriging, and therefore are dependent on the well data used and the spatial correlation of that data. A value attributed to the relevant specific gridblock is then drawn from the identified conditional cumulative density function. This procedure is repeated along the random path until all gridblocks have been visited.

Conditioning continuous realisations to static data through kriging is achieved as follows: assuming there are n static well data points and y a realisation of the continuous random field, Y , and only reproduces the n well data points by chance. Then the expression in Eq. (2.7) accomplishes conditioning realisation y to the n static data (Chilès and Delfiner, 1999). That is, the conditional realisation which honours both the variogram inferred from the data, to respect the spatial structure, and the static data values at the n well locations, y_c , in terms of the unconditional realisation, y , and the kriging estimators, y_{dK} and y_K , for the known data and the y simulated at the n static points respectively.

$$y_c(u) = y_{dK}(u) + [(y(u) - y_K(u))] \quad \text{Eq. (2.7)}$$

This procedure iterated for n different unconditional realisations produces n different conditional realisations where the mean and the variance of the conditional realisations approach the kriging estimator, y_{dK} , and kriging variance respectively as n approaches infinity. Note that at the well location in Eq. (2.7), $y(u) - y_K(u) = 0$, therefore $y_c = y_{dK}$ at the well location which in turn is the value of the static property data at the well. SGS can be considered one of the most popular methods for simulating reservoir properties in reservoir modelling workflows (Doyen, 2007). The approach has simple implementation and is flexible and suitable for integrating seismic data.

2.4.1 Discrete Realisations

Similar stochastic modelling techniques have been developed for mapping facies realisations in the reservoir using discrete or categorical realisations due to the diversity of complex geological environments and connectivity structures. Two main categories of stochastic model that generate discrete realisations outlined here are object-based models and indicator models.

2.4.1.1 Object-Based Simulation Models

These models are constructed from geometrical objects spread in space and are often used to represent fracture networks or channels in the reservoir. The locations, shapes and orientations of the objects are characterised with probability laws and portions relationships. The Boolean model (Matheron, 1967) is a specific form of object-based model. Object-based simulation models are well recognised and have been successfully applied for many years particularly for fluvio-deltaic reservoirs (Doyen, 2007). These models are constructed from the union of independently generated elementary objects, i.e. sinuous channels, ellipse, and barchans, where object locations are defined from a Poisson point process of constant density in the stationarity state. The shapes and sizes of the objects are independent of their locations. They use iterative optimisation algorithms like simulated annealing to place objects with predetermined shapes in the model with location constraints provided by well data. The main problem with the object-based models which needs to be overcome is with regard to conditioning to well data (Lantuéjoul, 1997).

Arnold (2008) utilised an object-based methodology framework where the IRAP RMS™ geological modelling commercial software was employed successfully to model complex object-based geological structures. He indicated that even though there is difficulty conditioning to well data and considerable CPU demand with the long iterations caused by slow convergence rates because of the optimisation procedure within the framework, it is still a useful tool for generating realistic facies distribution and was the only sourced software available at the time. The developed framework methodology and object-based models examples are thoroughly detailed in Arnold (2008).

2.4.1.2 Indicator Simulation Models

Indicator models differ from object-based models, in the need for the simulation of a value for each gridblock. This characteristic is similar to continuous Gaussian models. Indicator realisations can be constructed using (1) Truncated Gaussian Simulation (TGS) (Galli et al., 1994; Hu, Ravalec, and Blanc, 2001; Matherson et al., 1987; Xu and Journel, 1993) or its extended form, the Pluri-Gaussian Simulation (PGS) (Lantuéjoul, 2002; Le Loc'h and Galli, 1997; Le Loc'h et al., 1994), and (2) the Sequential Indicator Simulation method (Journel and Gomez-Hernandez, 1993).

The basic idea of truncated Gaussian simulation (TGS) is to generate realisations of normalised Gaussian random field and to truncate them using a threshold to generate facies realisations. The threshold value establishes the facies proportions whereas the spatial covariance structure of the Gaussian field determines the spatial continuity of the facies distribution. Because only one Gaussian random field needs to be simulated in order to simulate multiple facies it has a better speed than SIS in addition to the possibility of controlling the association between different facies by careful ordering of the Gaussian classes. The limitation of TGS lies in the usage of a single spatial covariance function of the Gaussian field to control the spatial relationship of multiple facies which makes it impossible to compel different anisotropy characteristics on the individual facies. The generalised version of the TGS, called the Pluri-Gaussian Simulation (PGS) has been developed to give more flexibility on anisotropy modelling and more control on the spatial correlation relationship between facies. This involves the simultaneous truncating of several Gaussian random functions. Each facies field imposes its spatial correlation relationship to one or more of the facies in accordance with the identified threshold. Another possibility is to use non-stationary thresholds, i.e. thresholds that are location-dependent.

The Sequential Indicator Simulation (SIS) is a modelling facies approach. An indicator variable is defined based on variogram models for each facies indicator property since each facie may have different variograms with different correlation lengths and anisotropy characteristics reflecting the difference in spatial continuity of the various existing facies. Kriging, as a least square error estimator, allows calculation of the probabilities needed in sequential simulation by performing an estimation of the indicator values. A random path is defined at first to visit sequentially each grid block

only once. At each grid block, a local sand/shale probability distribution by Indicator Kriging is calculated, based on a weighted linear combination of indicator data, using the original known well data and previously visited simulated gridblocks values as control points. In the next step, a simulated value (1 or 0) attributed to the relevant specific gridblock is then drawn at random from this identified local distribution. The simulated binary value is then utilised as an additional control point for the remaining simulation steps. A complete lithology simulation is obtained by repeating this process alongside the random path until all gridblocks have been visited.

2.4.2 Multiple-Point Statistics (MPS)

The pixel-based approaches outlined earlier reproduce a sample variogram estimated from data pairs and therefore preserve two point statistics. Nevertheless, two-point statistics has limited control on pattern geometry and connectivity, thus, cannot be used to characterise complex structures such as channels. Guardiano and Srivastava (1993) proposed to consider more than two locations at a time and introduced multiple-point statistics to extend the technique in an attempt to merge the flexible data conditioning achieved by pixel-based approaches with realistic shape information captured by Boolean methods and assist in capturing complex geological structures such as channels and curvilinear structures. Later the approach is efficiently implemented by Strebelle and Journel (2000), called single normal equation simulation (snesim) (Strebelle, 2002). A reservoir image with objects manually drawn from outcrop observation for instance or simulated from object-based simulation algorithms produced by geologists could be used. The basic idea is to learn multi-point statistics from this initial geological image called a training image and analyse the occurrence of different possible patterns. A new image or realisation is subsequently generated with a pixel-based sequential simulation procedure demonstrated earlier to create facies simulations that are conditioned to well data and approximately reproduce the MPS inferred from the training image. Assuming the reservoir has N gridblocks, a random path is defined at first to visit each grid block only once. At each gridblock, the conditional cumulative density function is determined based on the training image. In the next step, a uniform deviate is randomly drawn and transformed into facies indicator using the conditional cumulative density function. This procedure is iterated until all gridblocks are visited. The resulted simulated image realisation reproduces the probabilities inferred from the training image. The approach has two features: it reproduces realisations with realistic geological features and it

makes conditioning to static data very easy. MPS simulation techniques have gained popularity with many recent advanced schemes presented (Caers et al., 1998; Caers et al., 2003; Daly and Caers, 2010; Strebelle, 2002; Strebelle and Journel, 2001; Strebelle et al., 2003).

Simulated annealing (SA) optimisation approach is investigated in stochastic simulation where SA generates numerical models or realisations by formulating an optimisation problem to be solved. By optimising an objective function such as minimising a weighted sum of mismatch terms with respect to observed values, SA tries to reproduce different spatial statistics and respect constraints imposed in that objective function (Besag, 1986; Farmer, 1992; Geman and Geman, 1984; Kirkpatrick et al., 1983; Metropolis, 1953; Rothman, 1985). The applications of SA on the geostatistical topics are investigated by Deutsch and Cockerham (1994) and Datta-Gupta, et al. (1995).

2.5 Parameterisation

One of the key uncertainties in reservoir characterisation and simulation is populating with multiphase flow properties. For instance, relative permeabilities are measured on only a few core samples and then populated in reservoir simulation to large heterogeneous gridblocks that may contain many rock types. As a consequence, relative permeabilities are rarely considered reliable and are often modified with no physical justification during history matching. However, to predict recovery and to design an improved oil recovery strategy, it is required to input the relationships between relative permeability and saturation during multiphase fluid flow in porous media. Consequently, the prediction of relative permeability and saturation has been the subject of many studies in the past years using pore-scale modelling pioneered by Bryant et al. (1993a, 1993b). More extensions and experiments investigated in Adler and Thovert (1998), Bakke and Øren (1997), Bryant and Blunt (1992), Blunt et al. (2002), Øren and Bakke, (2002), and Øren et al. (1998).

Discretisation of reservoir simulation models commonly involves sizes of more than 10^6 gridblocks and even with upscaling generally the number of gridblocks is large. A resumption of interest in geostatistical inversion and practical needs have brought to research and industry problems with a huge number of parameters (permeability or porosity values). A key aspect in solving the optimisation problem is selecting a proper

parameterisation and so perturbing the initial reservoir model with a fewer number of parameters as well as honouring the prior spatial variability structure conditions.

2.5.1 Number of Model Parameters

Increasing the number of model parameters characterising the reservoir increases the number of degrees of freedom and therefore a larger number of possible directions to navigate the parameter space will need to be explored during the optimisation process. At present there is no specific definitive measure to tell how many parameters should be used in the optimisation. Increasing the number of parameters can facilitate reducing the objective function characterising the mismatch between the dynamic data and the observed history. However, overparameterisation may slow the exploration procedure as well. For example, adjusting hundreds of parameters in order to history match dozens of dynamic production data would be unfruitful, may complicate the model, and could generate parameters with no physical meaning. In addition, if this optimal set of parameters would enable the dynamic data to be perfectly reproduced, it may still have no predictive capacity. Thus, a balance of choices will need to be considered.

Visualising high dimensions is also an issue since it is difficult to view higher dimensional spaces. Al-Dossary (2004) investigated different ways to be able to analyse the large number of parameters.

2.5.2 Compartmentalisation

A common approach utilised in hydrology and petroleum engineering to reduce the number of parameters is the compartmentalisation (zonation or regionalisation) technique (Stallman, 1956a). An abbreviated version of the paper can be found in Stallman (1956b) which was later reprinted as a benchmark paper in hydrology (Freeze and Back, 1983). This paper pioneered the development of methods for ground water inverse problem and introduced the inverse problem to hydrogeologists and undoubtedly stimulated much of the research that led to numerous recent advances in this research area. The work involved the application of numerical methods for the solution of the finite-difference equations governing ground water flow, and led to his innovative work on the solution of the inverse problem. The compartmentalisation technique as adopted in petroleum reservoirs models involves grouping gridblocks to create subregions with constant permeability values (Figure 2.3 for a realisation with a spherical variogram). The optimisation is then achieved by adjusting these values.

Although the compartmentalisation technique does not handle the spatial variability model inferred from the static data, it is a standard method in petroleum practices (Floris et al., 2001). Section 5.9.1.1 in the thesis shows an application example of the compartmentalisation technique.

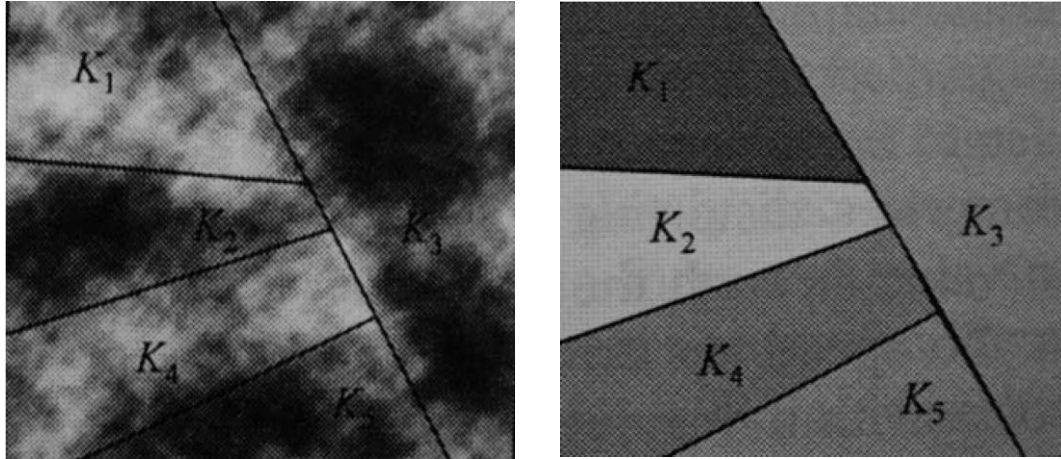


Figure 2.3: Compartmentalisation technique principle where griddblocks are grouped to create zones with constant permeability values (Source: Le Ravalec-Dupin (2005))

This method has an extensive history in petroleum engineering application. It started with early history matching studies of Jacquard and Jaïn (1965) and Jahns (1966). Gavalas et al. (1976) and Shah et al. (1978) studied the use of zonation approach for history matching, but they have concluded that Bayesian history matching approach has been more accurate and has quick convergence. It is however noted that the misfit obtained with the zonation approach is larger than desired because of the small number of degrees of freedom and the predefined prior ranges. Furthermore, the method may not approximate geology accurately and has discontinuities at the boundaries (Oliver and Chen, 2010).

2.5.3 Gradual Deformation Method (GDM)

The gradual deformation method is a stochastic parameterisation technique introduced by Roggero and Hu (1998) which allows narrowing of the parameter space. The gradual deformation method was initially developed for gradually changing Gaussian stochastic reservoirs models while maintaining their spatial variability (Hu, 2000). It was then extended to non-Gaussian reservoir models simulated from sequential indicator simulation (Hu, Blanc, and Noetinger, 2001) and Boolean techniques (Le Ravalec-Dupin and Hu, 2004).

The working mechanism for gradual deformation to construct a realisation from $Q+1$ independent realisations characterised by mean y_0 and the same covariance as the

combined realisations, is given in Eq. (2.8) where θ_i are the deformation parameters (Roggero and Hu, 1998).

$$\left[y(\theta_i, i \in [1, Q]) - y_0 \right] = \prod_{i=1}^Q [y_1 - y_0] \cos(\theta_i) + \sum_{i=1}^Q \sin(\theta_i) \prod_{j=i+1}^Q \cos(\theta_j) [y_{i+1} - y_0] \quad \text{Eq. (2.8)}$$

The major characteristics of the GDM are the reduction of the number of parameters and the preservation of the spatial variability while changing continuously the reservoir model, thus the entire reservoir model is simply modified by changing a few deformation parameters for any number of gridblocks. More details can be found in Le Ravalec-Dupin (2005).

2.5.4 Pilot Point Method (PPM)

The pilot point method also called master point method (de Marsily, 1978; de Marsily et al., 1984) was first developed for prediction purpose, then extended to calibrate permeability field to dynamic data (Cuypers et al., 1998; Gomez-Hernandez et al., 1997; RamaRao et al., 1995).

To adjust a realisation discretised over a grid, a set of gridblocks or points called pilot points is chosen with the goal of modifying their values. The resulting perturbation is mapped to the entire realisation using the kriging equation. Here, the conditional realisation respects the static data and the pilot points in addition to the variogram inferred from static data. The pilot points are assimilated to static data, whose values can be adjusted to oblige the realisation to respect the dynamic data similar to the gradual deformation method. As the adjustments are propagated to the entire realisation with kriging the spatial variability structure model is maintained. This method is applied to locally modify realisations from a limited number of parameters, while preserving the spatial variability of the studied attribute (permeability, porosity, etc) where the spatial variability is approximated by a variogram model inferred from static data.

2.5.5 Gradual Pilot Point Method

The two techniques demonstrated earlier, the gradual deformation method and the pilot point method were designed to reduce the number of parameters and to honour the inferred spatial structure. The weaknesses for the pilot point method have been revealed that pilot points can be assigned unreasonably extreme values and possible correlations among the pilot points are neglected. The gradual pilot point method was introduced (Le Ravalec-Dupin and Hu, 2007) to overcome these limitations by combining the pilot

point and gradual deformation methods. The crucial difference compared to the original pilot point method is that pilot points are changed via gradual deformation rather than optimisation. In the proposed approach the method does not produce extreme variations compared to the original pilot point method. In the proposed approach intermediate gradual deformation parameters are introduced which govern the pilot point values. In addition, the correlations among the pilot points are respected when modifying simultaneously the entire set of pilot points from a single deformation parameter. Hence, many pilot points can be placed on the random field, irrespective to their locations and they can produce local and global deformation. The approach applied for calibrating permeability fields was a two-step approach where firstly the gradual deformation method is used to globally deform the permeability fields. When the permeability fields had been globally improved, they were locally refined using the gradual pilot point method.

2.5.6 Facies Proportions Calibration Technique

Enchery et al. (2010) proposed a parameterisation technique to automatically adjust facies proportions during a history matching process. This method depends on the ratio of average proportions between facies classes with a priori poorly known proportion. Two different schemes were introduced depending on the geological environment. The first generates discontinuity between the target area and the embedding environment. This approach does not ensure the continuity of the facies proportions between the transformation region and the rest of the reservoir and it helps to increase the contrast in the proportions between different geological objects like channels. On the other hand, the second scheme ensures continuity of proportions at the boundary of the modified region and aims more at reproducing an average trend in the proportions of the identified architectural elements. The methods were tested for improving the calibration of 4D seismic data (Enchery et al., 2010) on a faulted turbidite field located in offshore Angola.

2.5.7 Stochastic Partial Differential Equations

Potsepaev and Farmer (2010) proposed a coordinate free approach called stochastic elliptic partial differential equations for modelling stochastic textures in reservoirs. Often reservoir properties are modelled using a Gaussian stationary random field then mapped back to the physical space. The approach avoids the construction of a map from physical to parametric space resulting in stratigraphic coordinates, (x, y, z) , where it is possible

to generate realisations directly in the physical space in the presence of deformations and faults. The method is tested in simple reservoir property modelling cases and generalisations involving nonlinear terms are to be considered.

2.5.8 *MP Simulations without Computing MP Statistics*

Mariethoz et al. (2010) presented a method that produces conditional realisations honouring the high-order statistics of univariate or multivariate training images. The method is simple and easy to parallelise and therefore it can produce very large and complex realisations. It is based on a sampling method introduced by Shannon (1948), however, it does not require conditional probabilities computations and to store them. A distance metric between data configurations is used in the sampling process to simulate both discrete and continuous parameters. The method does not require storage needs, thus, neighbourhoods can have virtually any size and neighbourhoods are not restricted to a template, making multiple-grids unnecessary. The method was applied to simple cases and it has promising preliminary results.

2.5.9 *Machine Learning Methods*

Recent machine learning methods have been applied in reservoir modelling for predicting reservoir properties and reproducing complex geological structures including Principal Component Analysis (PCA) (Mohamed et al., 2010a; Sarma et al., 2008b), Support Vector Machines (SVM) (Alanazi, 2009; Demyanov et al. 2008; Gallardo and Leuangthong, 2009), Multiple Kernel Learning (Demyanov, Foresti, Christie, and Kanevski, 2011; Demyanov, Foresti, Kanevski, and Christie, 2010), and Kernel Ridge Regression (KRR) (Sætrum and Omre, 2010). The methods have demonstrated their capacity in analysing and describing different geological scenarios (Caers, 2008). Next we briefly review the working mechanisms for some of these methods. Section 9.2.1 provides a demonstration of principal component analysis technique applied in the thesis in detail.

2.5.9.1 *Support Vector Regression (SVR)*

Support vector machines (SVM) were first introduced by Vapnik. There are two main classes for support vector machines: support vector classification (SVC) and support vector regression (SVR). Support Vector Regression is a kernel-based non-linear rigorous technique for regression prediction introduced by Vapnik et al. (1997). Support Vector Regression (SVR) is the most common application form of SVMs. The basic

ideas underlying support vector machines for regression and function estimation has been overviewed in Smola and Schölkopf (1998) and Schölkopf and Smola (2002) with a summary of currently used algorithms for training SVMs. The basic idea in kernel methods, called the kernel trick, is to map the data into a high-dimensional feature space via a nonlinear mapping, known as a kernel function, k , and to do linear regression in this space (Vapnik, 1999) as in Eq. (2.9) where Φ is the map in the feature space for data vector x .

$$k(x, x_i) = \Phi(x) \cdot \Phi(x_i) \quad \text{Eq. (2.9)}$$

SVR controls the complexity of the model and provides accurate results with high-dimensional and noisy data by constructing sparse kernel models. The SVR model is based on the use of an ε -insensitive loss function, which preserves the sparseness property. The SVR model is built by kernel functions linear expansion, $k(x, x_i)$ as in Eq. (2.10) for predicting new inputs expressed in terms of kernel functions where N is the number of data points. The function encapsulated weighted dot products in the high-dimensional feature space between x and the support vectors x_i obtained by the SVR model which are the closest samples to decision boundary. The weights of the expansion have non-zero values to contribute to the maximum marginal solution and they are produced by quadratic programming, thus as the problem is convex, its solution is unique. The SVR generated model depends only on a subset of the training data, because the cost function for building the model ignores any training data close to the model prediction within a threshold ε .

$$y(x) = \sum_{i=1}^N (\alpha_i - \hat{\alpha}_i) k(x, x_i; \theta) + b \quad \text{Eq. (2.10)}$$

The effectiveness of SVR lies in the selection of kernels and what is called soft margin parameters (C, ε). Soft Margin approach selects a hyper plane that splits the training data as cleanly as possible, whilst still maximising the distance to the nearest cleanly split examples. The method introduces slack variables, ξ_i , which measure the degree of misclassification of the datum x_i . $C > 0$ refers to the parameter that controls the trade-off between minimising training errors and controlling model complexity and ε is called the hyper parameter utilised for constructing sparse models with a reduced set of the training data only. For kernels, different pairs of (C, ε) values could be used, however, in practice the one with the best cross-validation accuracy is usually picked. Attempting exponentially growing sequences of C is a practical method to identify good parameters. The one with the best accuracy can be selected for classification and prediction which

can be then used in future for testing and prediction. Figure 2.4 illustrates the problems that SVM can tackle.

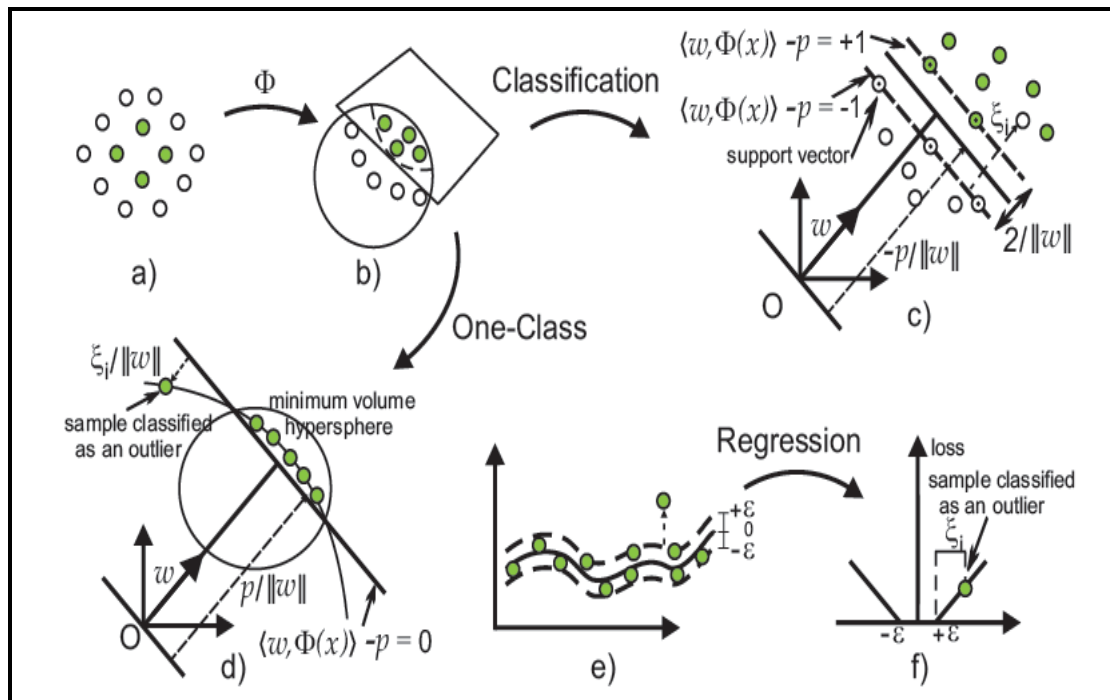


Figure 2.4: SVMs kernel trick, SVM does classification (c) or regression (e,f) or characterisation of a dataset in one-class SVM (d) (Source: (Loshchilov et al. (2010)))

SVR application in modelling petrophysical properties distribution of a fluvial environment has been investigated in Demyanov et al. (2008) and has shown promising results in modelling complex structures. In this application approach, a geomanifold is identified using two classes of features. One comes from well data and the seismic attribute acoustic impedance (for correlating with porosity), called labelled data, and the second represents the spatial geological structure, and called unlabelled data. The SVR model computes linear regression in high dimensional space using a single unique kernel in a feature space as a correlation model for all input variables. Properties are then populated into the gridblocks whilst maintaining the spatial structure and preserving the seismic and static well data.

2.5.9.2 Multiple Kernel Learning (MKL)

The kernel used in the Demyanov et al. (2008) application is a unique kernel which they indicated is a limitation of the SVR technique because reproducing multi-scale non-stationary structure is difficult even while using a semi-supervised learning approach with unlabelled data, since the correlation on multiple scales is not explicitly reflected in the model. Recent advances have been investigated to deal with this aspect where a

multi-kernel SVR (Gert et al., 2004) has been applied for a spatial regression problem in Pozdnoukhov and Kanevski (2008).

The use of multiple kernels gives flexibility and adaptability to the data and can enhance the performance of the prediction model. Multiple kernel learning can be achieved by defining a convex combination of basis kernels. A simple kernel can be constructed by substituting $k(x, x')$ as in Eq. (2.11) in which d_i refers to the weights augmented to each kernel and β_i is the corresponding kernel parameter.

$$k(x, x') = \sum_{i=1}^Q d_i k_i(x, x'; \beta_i); \text{ where } \sum_{i=1}^Q d_i = 1 \text{ and } d_i \geq 0 \quad \forall i \quad \text{Eq. (2.11)}$$

Different multiple kernel learning variations differ in simultaneously optimising the SVR weights α and d . Among these variants Rakotomamonjy et al. (2008) introduced a simple two-step optimisation approach, called Simple MKL. During the first step in the approach the weight vector d is optimised and then followed by solving the SVR problem in the second step.

The main strength of the MKL technique is its ability to determine the relevance of particular features or groups of features, i.e. permeability distribution to yield a better understanding of the problem and hence enhance the performance as a result of noisy features elimination. MKL can be a successful predictive model which serves as an exploratory tool. The method works by incorporating features obtained from data at difference scales such as seismic static data, well data, and reservoir images in a statistical consistent way in order to provide proper predictions by analysing and detecting the important features. The MKL formulation is flexible and can be used in a variety of ways: the features, combination of features spaces accounting for different features, different scales (same features, but different kernel parameters) or all can be considered. The performance of the conventional SVR and its MKL extension is compared in Foresti et al. (2009; 2010) for spatial wind speed mapping which is a similar problem to reservoir modelling. The MKL method obtained superior results compare to the SVR approach. Demyanov, Foresti, Christie, and Kanevski (2011), and Demyanov, Foresti, Kanevski, and Christie (2010) applied the MKL approach on the same example outlined earlier to model reservoir petrophysical properties of a fluvial system which features multi-scale structures of a meandering channel and a broken fading channel with highly noisy data. The resulting MKL application showed a

potential in detecting and modelling complex structured geobodies in addition to the ability to interpret the results in comprehensive fashion.

2.5.10 Other Methods

Farmer (1989, 1992) introduced simulated annealing technique based on Two-point Histogram to create patterns of discrete rock types representing different lithologies (King, 1992). Incorporation of large-scale soft data will be conditioned via an objective function in the simulated-annealing method. Two-point histogram can not encapsulate sufficient information to reflect complex geological structural models, and the results can not capture spatial relationships. Deutsch and Journel (1992) then proposed the Multipoint Histogram that captures much more information than the two-point histogram. Le Ravalec et al. (2000) proposed the Fast Fourier Transform-Moving Average (FFT-MA) generator approach to produces unconditional Gaussian fields with stationary covariance functions based upon the moving average framework. In Bayesian image analysis, data are most usually available as a degraded image, corrupted by noise and smoothed in some ways (Besag, 1986; Geman and Geman, 1984; Ripley, 1988). Sætrom and Omre (2010) used Ensemble Kalman filtering with shrinkage regression techniques and Ensemble Kalman filtering for non-linear likelihood models using kernel-shrinkage regression techniques also called Kernel Ridge Regression which belongs to the kernel methods family. The probability perturbation method was introduced by Caers (2003; 2004) and applied by Hoffman and Caers (2003, 2004) in the North Sea reservoir. Jafarpour and McLaughlin (2007, 2009) used the discrete cosine transform and presented good results. More recently, Caers et al. (2010) developed multi-dimensional scaling (MDS) modelling techniques in metric space, which means that processes are reformulated and achieved in metric space, where the location of any model is determined entirely by the mutual differences in responses as defined by a distance term.

2.5.11 Parameterisation Summary

Several techniques, suitable for building fine or high-resolution reservoir models, have been reviewed in this section. The methods are investigated thoroughly in the specialised literature and some are utilised in current petroleum community software. A comprehensive description can be found in Beucher and Renard (2005), Caers (2005), Dubrule (2003), Doyen (2007), and Le Ravalec-Dupin (2005) in which excellent illustrative examples are demonstrated.

Generally, the aim of these methods is to produce geologically realistic reservoir models which preserve the spatial variability model inferred from the available static data for meaningful results. The geostatistical techniques described are often used to populate a reservoir model with porosity values. A permeability-porosity relationship as in Eq. (2.12), derived from a petrophysical study, can then be used to attribute permeability values to all the reservoir gridblocks.

$$\log_{10}(k) = \alpha\phi + \beta \quad \text{Eq. (2.12)}$$

where ϕ is the porosity and k permeability. α and β are static coefficients estimated per facies. Since, ϕ distribution is normal, k distribution is lognormal.

Another technique called collocated cokriging (Xu et al., 1992) can be used for populating permeability in gridblocks where the permeability variogram, describing the spatial continuity that may have different correlation lengths and anisotropy characteristics, is available. In this technique the porosity attribute is called a covariable where a simple linear correlation between porosity and log permeability is evaluated by a correlation coefficient which can be computed by scatter plotting logpermeability versus porosity acquired from the core data. The basic idea is to produce a permeability model which reflects the continuity modelled in its variogram and maintain the permeability data from wells in addition to having the observed correlation with the porosity. The applicability of the technique has been widely used to interpolate the depth in structural modelling (Doyen, 2007; Caers, 2005).

The dependencies between covariates, e.g. between porosity and permeability, can also be co-estimated using some more flexible and efficient geostatistics based methods indicated earlier such as SVM or the more general MKL and KRR.

2.6 Chapter Summary

This chapter presented an overview of the literature on reservoir modelling fundamentals, including reservoir performance prediction techniques and a review of the recent development on reservoir characterisation and modelling topics.

Chapter 3 – Reservoir Model History Matching and Uncertainty Quantification Techniques: Literature Review

This chapter presents history matching and uncertainty quantification concepts with a review of the assisted history matching techniques and uncertainty quantification methods. The Bayesian methodology and uncertainty quantification framework used in this thesis and other topics that set the basis for the next chapters are provided.

3.1 Objective Function

Conditioning to dynamic data, the production data, well pressures, water cuts, etc, is a nonlinear problem which may be solved using linearisation or optimisation. The optimisation approach is favoured among engineers because they have to deal with complex geological structures. Zimmerman et al. (1998) indicated that optimisation provides better results than linearisation.

Optimisation algorithms use an objective function, also called a cost function or misfit function. It is a function of unknown parameters and measures the discrepancies between simulated and observed data. The objective of the optimisation process is to minimise the objective function by fine tuning the unknown parameters. Defining an appropriate objective function and quantifying the mismatch are critical challenges for any reservoir engineering study. Objective function values define a multidimensional

surface that may have many twisting valleys. Often, the objective function is the negative log of the likelihood as defined in Eq. (3.1).

$$M = -\log(p(O|m)) \quad \text{Eq. (3.1)}$$

where m is the model parameters and O is the observed data. $p(m|O)$ is the probability of the model given the data. Assuming the measurement errors are Gaussian, independent and identically distributed with zero mean and assume that there is no error in the numerical solution of the model, the posterior can be defined as in Eq. (3.2). Figure 3.1 shows the likelihood definition based on this assumption. The misfit can be computed using the conventional least squares formula in Eq. (3.3).

$$p(O|m) = \prod_{t=1}^T \left(\frac{1}{\sqrt{2\pi\sigma}} \right) e^{-\frac{(q^{obs} - q^{sim})_t^2}{2\sigma^2}} \quad \text{Eq. (3.2)}$$

$$M = -\log(p(O|m)) = \sum_{t=1}^T \frac{(q^{obs} - q^{sim})_t^2}{2\sigma^2} \quad \text{Eq. (3.3)}$$

where T is the number of observations, q is the rate, *obs* and *sim* refer to observed and simulated, and σ^2 is the variance of the observed data. Other statistical models for observational noise give rise to different expressions for the misfit.

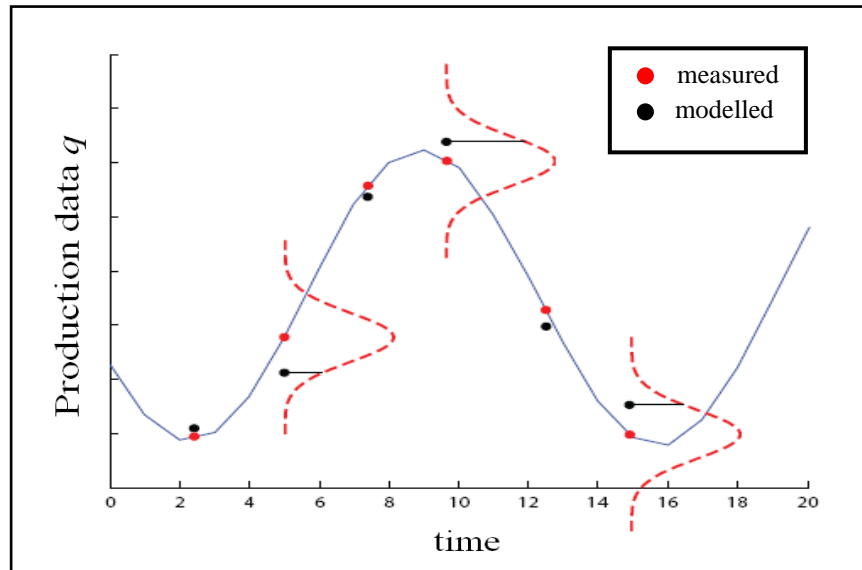


Figure 3.1: Likelihood function definition

It is noted that refining the grid size in reservoir models does not guarantee that the model will capture all the physics. This is termed the model inadequacy (Kennedy and O'Hagan, 2001). Simulation error model can be included in the history matching procedure where fine grid models are compared with the coarse model (Christie et al., 2008; O'Sullivan, 2004; Park et al., 2010).

The measurement errors can be estimated from the data with statistical data fitting approaches. Erbas (2007) used the best history matched model while Arnold (2008) utilised a simple polynomial curve to estimate errors in a similar approach. Valjak (2008) used Wiener filtering approach to estimate the errors from the production data directly. The common recommended practice (Roggero and Hu, 1998) is to allow 5% error tolerance from observed data for oil and gas production rates.

3.2 History Matching

A reservoir is characterised by a set of model parameters. History matching is the process of adjusting the value of these model parameters to produce a model that matches the production data, as well as honouring the static data from wells and any seismic survey. Extensive advances have been made during the last few decades in inverse problem theory to integrate dynamic data into reservoir models resulting in the growth of a broad list of history matching approaches. The time-lapse or 4D seismic data also needs to be integrated quantitatively into reservoir characterisation at the history matching stage to update reservoir models (Aanonsen et al. 2002; Castro, 2007). Figure 3.2 shows the development of research on the subject since the first attempts to solve the inverse problem were initiated by Nelson (1960) and Jacquard and Jain (1965).

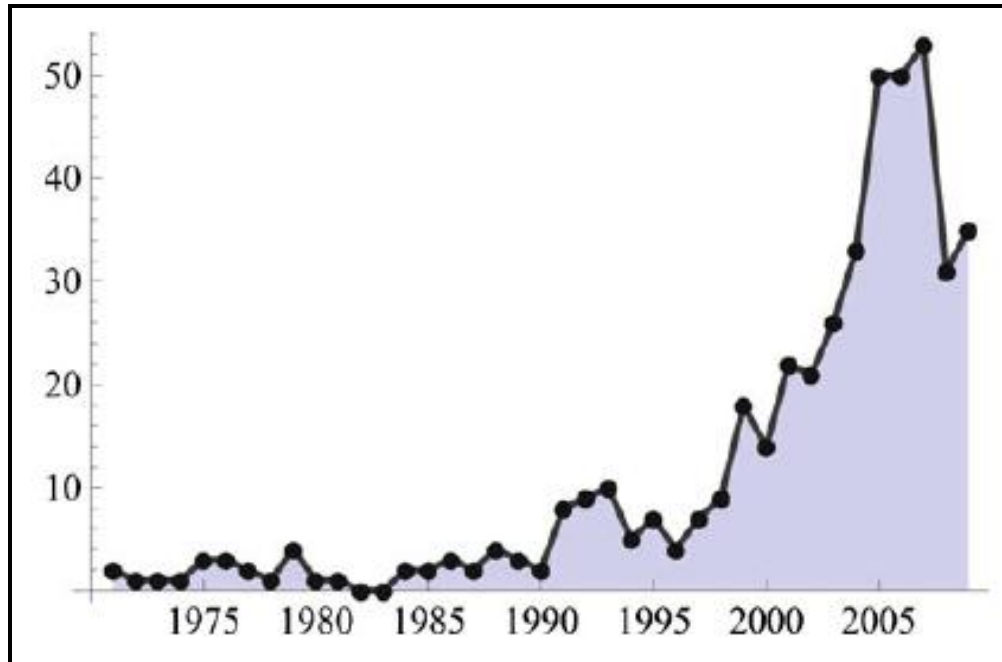


Figure 3.2: Number of papers on history matching prepared each year for SPE conferences and journals (Source: Oliver and Chen (2010))

3.2.1 Manual History Matching

Traditional manual history matching is achieved through a deterministic trial-and-error process in which the engineer's knowledge, judgment, and expertise are exploited. The process is tough, very time-consuming and only provides a single forecast and may not be feasible for large reservoirs.

There is no single universal method for achieving a history match but there are some guiding principles to aid the history matching procedure considered in the literature such as Crichlow (1977), Fanchi (2001), Mattax and Dalton (1990), Saleri et al. (1992), and Thomas (1982) in the following steps.

1. Match the volume measurements with material balance and recognise aquifer support
2. Match the energy of the reservoir energy – the pressure both globally and locally for each well. Global model quality can be initiated by matching the average field pressure as overall material balance. The pressure distribution displays the spatial variation linked to local variability of reservoir performance. Thus, local pressure matches can be examined with well test plots at certain locations in time.
3. Match variables that are dependent on saturation. These variables like gas-oil ratio (GOR) and water-oil ratio (WOR) are often the most sensitive production

variables since they have an impact on both breakthrough time and the profile of the WOR or GOR curve.

4. Match flowing pressure for wells.

As a result of ensuring material balance is used correctly, the pressure profiles should match the observed data, and the matching of water rates then follows. The first two steps need to be accomplished before the final two steps since if they cannot be achieved, then there is a strong possibility that the model is inadequate as a result of: inaccurate or incomplete field data, poor characterisation of the model, or the wrong model was selected to name a few examples. Reviews of the model will be essential.

3.3 Assisted History Matching Algorithms

Even though all history matching approaches aim generally at minimising the objective function (misfit), there are variations on how to do the minimisation and quantify the uncertainty. This arises from the fact that some optimisation algorithms suit some history matching problems better than others, and some algorithms also have simpler implementations than others. Choosing the optimisation algorithm on specific history matching optimisation problems can be facilitated by the objective function structure.

There are a number of algorithms that have been used in the petroleum literature to generate history matched models and quantify uncertainties, and these algorithms fall into two principal types: data assimilation methods and calibration methods (Christie et al., 2005). Data assimilation methods calibrate a number of estimates of model parameters sequentially to points in a time-series of observed data. In calibration methods on the other hand, a complete run of the simulation is carried out and the match quality to the historical production data is used to move the estimates of the model parameters towards a better solution.

This section summarises some of the calibration algorithms that have been employed in the specialised history matching literature, examples of where it was performing well and their feasibility to work in complex real life problems. More details can be found in Oliver and Chen (2010).

3.3.1 History Matching Optimisation with Deterministic Techniques

Gradient methods are highly efficient and have been widely used for history matching problems. These methods require the calculation of the derivative of the objective function with respect to the reservoir model parameters as either gradients or sensitivity coefficients. This computation requires a lot of computational time when a large number of parameters is included. An effective method in obtaining the gradients is the adjoint approach (Chavent et al., 1973). Sensitivities (gradients) can also be estimated or approximated using a single flow simulation, the so-called streamline approach (Datta-Gupta, 2000).

Techniques available include: steepest descent, Newton, quasi-Newton, Gauss-Newton, conjugate gradient, and Levenberg-Marquardt (Press et al., 1988; Ranganathan, 2004; Sun, 1994; Tarantola, 1987) which can be found in some modern commercial history matching software such as SimOpt (2005). Early work by Bissell et al. (1994) history matched two real case studies using gradient methods and found that the results were comparable with hand matches. Lepine et al. (1999) estimated the uncertainty of production forecasts using linear perturbation analysis. A range of possible future production profiles were obtained and the confidence intervals for the future production performance were derived to quantify the uncertainty. The main problem of using gradient based methods is that they can easily get trapped in local minima due to the complexity of the flow problem.

In recent years, research has involved around quantifying uncertainty by generation of multiple history matched reservoir models, rather than just seeking the best history matched model. A practical reason for using multiple history matched models is that a single model, even if it is the best history matched model, may not provide a good prediction (Tavassoli et al., 2004). Stochastic methods that have other mechanisms that produce multiple models are demonstrated next.

3.3.2 History Matching Optimisation with Stochastic Techniques

Modern reservoir management has an increasing focus on predicting the likely range of field recoveries and consequently providing economic evaluations of different field development strategies. The favoured route to such uncertainty quantification is by

obtaining multiple history matched simulation models and using them to estimate ranges in likely recovery factors.

Recently stochastic sampling algorithms have gained some popularity and have been among the most rapidly developing with more oil and gas companies adopting them as a strong component of the reservoir history matching process. This is due to their relatively simple implementations and capacity for parallelisation capacity. With an increased availability of computing resources, it is possible to obtain multiple models. The methods do not require the computation of the gradients. They are equipped with various heuristics for randomising the search and hence exploring the global space and preventing entrapment in local minima as well as the sequence of parameter values generated generally improve the history match as time evolves. However, they are slow and computationally require a significant number of objective function evaluations.

Examples of stochastic methods include the golden section that deals with only single parameter, simulated annealing, the Metropolis-Hastings method, chaotic approach, simplex method, genetic algorithms and evolutionary strategies, which deal with high-dimensional problems.

The Golden section method (Press et al., 1988) is usually utilised to solve optimisation problems which include a single parameter where the parameter space is partitioned off into two to bracket the optimum. The produced ranges decrease over time and goes towards the optimal parameter value. The name of the method is derived from the way the method works which is partitioning the parameter space according to the Golden number. A similar approach, interval halving, works by taking an interval $[a,c]$ and dividing it into two intervals of equal size. Then checking a point in the middle, b . If the function altered sign in the interval $[a,b]$, that is $f(a)f(b) < 0$, then by the intermediate value theorem there is a solution in this interval. If not, then the solution is in the interval $[b,c]$. This is repeated until a sufficiently small interval is obtained.

Simulated annealing (SA) (Kirkpatrick et al., 1983; Černý, 1985; Press et al., 1988) is a method inspired by thermodynamics for the solving global optimisation problems by trying to find the global optimum of a given function in a large parameter space. The approach is an adaptation of the Metropolis-Hastings algorithm (Metropolis et al.,

1953). The name and inspiration come from annealing in metallurgy, an approach consisting of heating and controlled cooling of a material to increase the size of its crystals and reduce their deficiencies. With high temperature values, the molecules of a liquid have a free movement according to one another. When the temperature decreases slowly, the thermal mobility vanishes. If cooling is very fast, the final state can be a shapeless state whose energy is fairly higher than the energy of the crystallised state. The energy here is understood as the objective function where temperature is the control parameter. Starting from a point in parameter space, a random update is accepted or rejected according to the energy difference. That is, the state is accepted if it has improved the objective function. In this method, there is a possibility that the system state gets out of local minimum for a global optimum. Ouenes et al. (1993) show an application in petroleum engineering.

Mantica et al. (2002) used a method where an optimisation problem can benefit from an analogy with electric forces. The optimisation procedure called chaotic dynamic begins with a set of possible solutions. Zabalza-Mezghani (2000) extended a method called the simplex method by incorporating gradient information

Genetic algorithm (GA) (Goldberg, 1953; Holland, 1975) based on natural processes has been used widely in history matching, and is available in a variety of forms. The method is inspired by the probabilistic change rules motivated by Darwin's theory of evolution to solve optimisation problems with an evolutionary process. The method simply starts with a population of solutions or individuals who produce new population based on their objective function values. The evolution procedure includes three simple operators: reproduction, crossover and mutation. The evolutionary optimisation procedure is repeated until an acceptable individual solution is produced. Variations of the method include binary coded GAs and real-coded GAs. Romero et al. (2000a) applied a modified genetic algorithm to a realistic synthetic reservoir model and studied the main issues of the algorithm formulation. Yu et al. (2008) used genetic programming to construct proxies for reservoir simulators while Carter and Ballester (2004), Erbas and Christie (2007a), and Romero et al. (2000b) also investigated the method on real field applications and have shown how this method can find an ensemble of solutions. An overview of genetic algorithm in the oil industry is provided in Velez-Langs (2005).

Some innovative global optimisation approaches that have had a good track record of successful application within the petroleum industry are: evolutionary strategies (Schulze-Riegert et al., 2001; Schulze-Riegert and Ghedan, 2007; Schulze-Riegert et al., 2009), population-based incremental learning (Petrovska and Carter, 2007), estimation of distribution algorithms (Petrovska, 2009; Petrovska and Carter, 2006; 2010), differential evolution (Jahangiri, 2007; Hajizadeh et al., 2011), ant colony optimisation (Razavi and Jalali-Farahani, 2008a, 2008b; Hajizadeh et al., 2009), and neighbourhood algorithm (Christie et al., 2002). The latter will have a particular focus in the next section since it is used in the thesis for comparisons.

3.3.2.1 The Neighbourhood Algorithm

The Neighbourhood Algorithm (NA) is a stochastic sampling algorithm that was originally developed by Sambridge (1999a) for solving geophysical inverse problems. It is a derivative-free method that aims at finding an ensemble of acceptable models rather than seeking for a single solution. NA is a sampling technique that uses the properties of Voronoi cells in high dimensions to achieve multiple history matched models. The key approximation in NA is that the misfit surface is constant in the Voronoi cell surrounding a sample point in parameter space. Quantifying the uncertainty using the NA involves two phases: a search phase, in which we generate an ensemble of acceptable solutions of the inverse problem, and an appraisal phase, in which NA-Bayes (NAB) (Sambridge, 1999b) computes the posterior probability based on the misfits of the sampled models and the Voronoi approximation of the misfit surface. NA has been used in a number of reservoir history matching studies (Christie et al., 2002; Christie et al., 2011; Erbas and Christie, 2007a; Rotondi et al., 2006; Subbey et al., 2003; Subbey et al., 2004).

The search phase can be summarised as follows:

1. The NA algorithm is initialised with a population of an initial set of n_{init} models randomly generated in the search space by a quasi random number generator; for each model the forward problem is solved and the corresponding misfit value is obtained.
2. Determine the n_r models having the lowest misfit values, among the previously generated models n_s .
3. Generate a total of n_s new models in the n_r Voronoi cells previously selected.

4. NA returns to Step 2 and the process is repeated until it reaches the user defined number of iterations.

Thus a total of $N = n_{init} + n_s \times \text{no. of iterations}$ models is generated by the algorithm. The ratio n_s/n_r controls the behaviour of the algorithm: the lowest value of $n_s/n_r = 1$ aims to explore the space and find multiple regions of good fitting models; as the value of n_s/n_r is increased, the algorithm tends to improve the matches obtained at the expense of finding multiple clusters of good fitting models. A general guideline is to start with a value of $n_s/n_r = 2$ to obtain a balance between exploration and exploitation. Figure 3.3 summarises the NA workflow.

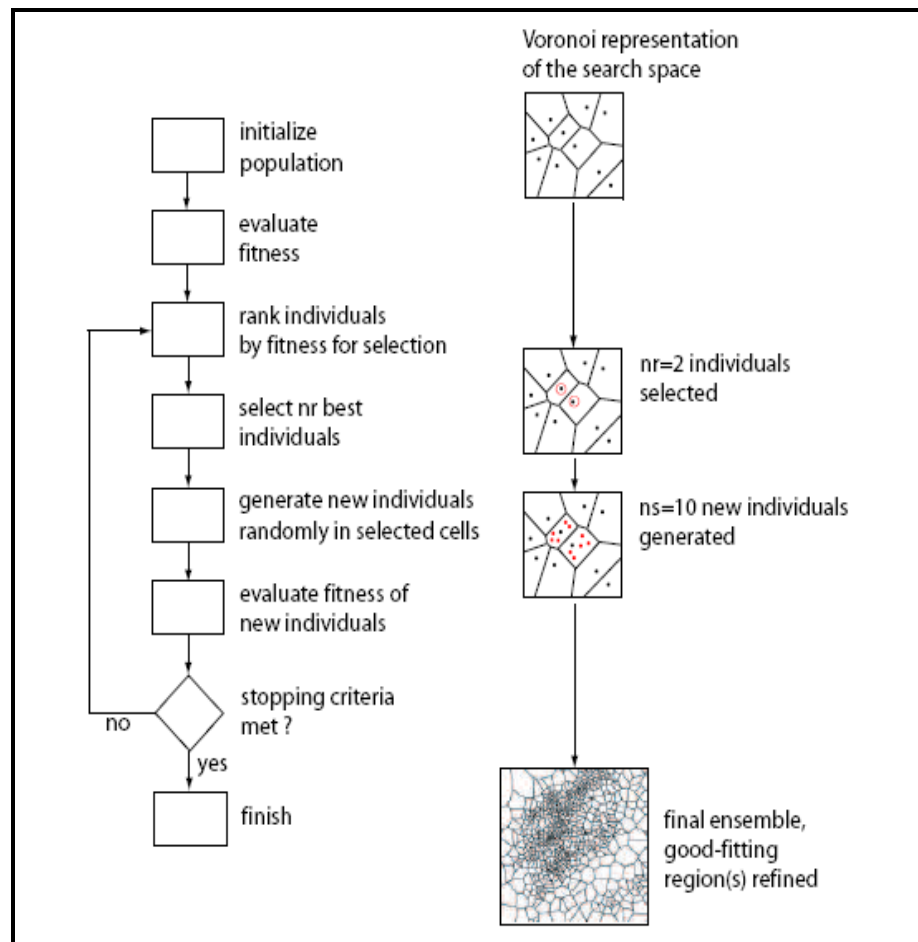


Figure 3.3: NA optimisation workflow (Source (Erbas, 2007))

3.4 Quantifying the Uncertainty

Uncertainties are present in any scientific application including simulations due to: inaccurate representation of the initial boundary conditions, inexact model parameters, and incomplete knowledge of the modelling system. The two main sources of error influencing uncertainty in our predictions in history matching context, and which need to be accounted for are the model and observation errors. The data errors encompass lack

of knowledge of the subsurface geology and errors inherent in the time-dependent dynamic data. The model errors on the other hand are based on incorrect physics, impact of non-linearity, and the choice of numerical simulator we pick such as mass balance or streamline, finite element or finite difference. These errors occur from the approximation of the initially continuous conservation and flow equations with discrete analogues, and the inability to capture sub-grid details. The major error however, is the result of being unable to capture these sub-grid details (O’Sullivan, 2004; Okano, 2006). Concern must be taken with errors caused by numerical diffusion and cell-aspect ratio. Numerical diffusion dictates that front resolution is impossible over fewer than three grid cells, while cell-aspect ratio errors are caused by the sensitivity of simulation to cell height-thickness ratio. Simulation errors are also correlated in time. This time correlation has to be accounted for in evaluating model fit to data, otherwise acceptable models are rejected when many points are contained in the time series of production data and would have a major effect on any method that needed real time rate and pressure data.

Integrating uncertainty information in the simulation representation, propagating these uncertainties through the history matching framework, and finally putting together the uncertainties in the model prediction is of practical importance. The uncertainties in the model outputs have major effects in reservoir plan optimisation and facilitate policy-making. Through increasing model complexity and the amount of computation entailed, several traditional uncertainty quantification techniques have their strengths and limitations. Remarkable algorithms which have been developed for handling uncertainty over the past decade includes Bayesian approaches (Andrieu, 2004; Clyde and George, 2003; Oden et al., 2010a, 2010b), and particle filter methods (Evensen, 1986). Examples of algorithms for uncertainty quantification are discussed below.

3.4.1 Uncertainty Quantification with Global Optimisation-Based Algorithms

Because the sampling density of the misfits obtained with global optimisation algorithms is not related to the posterior probability density, a separate calculation has to be carried out to compute probabilities of the models with the NA-Bayes algorithm (Christie, 2011). The NA-Bayes (NAB) algorithm, developed by Sambridge (1999b), evaluates the posterior probability density function, based on the ensemble of models generated during the automated history matching. This calculation assumes that the

misfit is constant in each Voronoi cell, and calculates the probability as the exponential of the negative misfit times the volume of the Voronoi cell. Consequently, the forward simulation are only required for the models that are resampled by NAB. The NAB resampling procedure is described below. The resulting ensemble of models with their posterior probabilities can then be used to estimate the p_{10} - p_{50} - p_{90} uncertainty envelopes. Figure 3.4 presents the Bayesian framework for the uncertainty quantification used in this thesis.

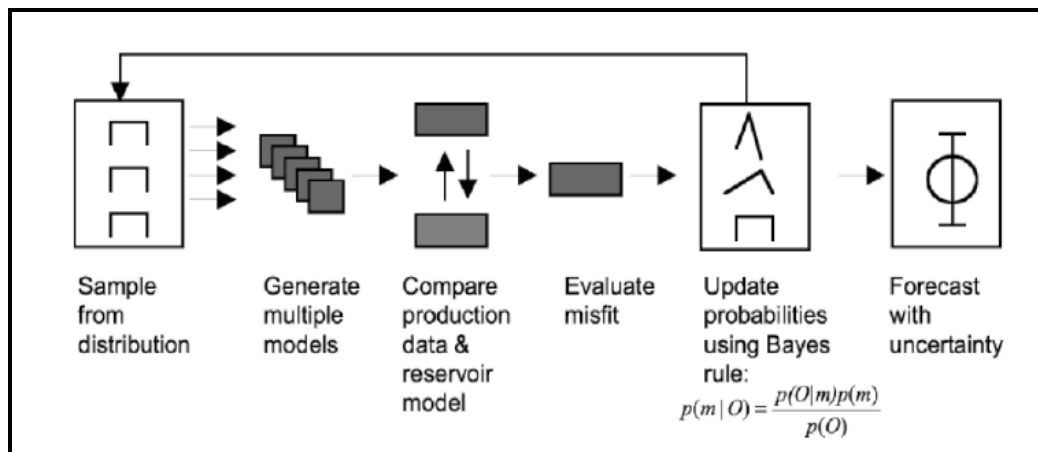


Figure 3.4: Bayesian framework for uncertainty quantification (Source: Christie et al., 2006)

3.4.1.1 NA-Bayes

This is the appraisal stage of the Neighbourhood Algorithm referred to as the NA-Bayes or NAB. The NAB procedure can be summarised as follows.

1. A starting point is selected, usually the minimum misfit corresponding to the most likely model, represented by point B in Figure 3.5.
2. Random steps are then performed along each parameter as shown by the two steps in Figure 3.5.
3. A range is defined for each parameter axis and denoted as l_i to u_i in which a conditional probability distribution function, such as $P_{NA}(x_i | x_{-i})$ as shown in Figure 3.5, is constructed based on intersection points of the range with the Voronoi cells of the ensemble.
4. A new step is proposed by random deviation from uniform distribution through the range.

- The proposed step x_i^p is accepted or rejected according to Eq. (3.4) in which $P_{NA}(x_i^{max} | x_{-i})$ represents the conditional maximum value throughout along the range and r refers to a second random deviation in the interval (0,1).

$$r \leq \frac{P_{NA}(x_i^p | x_{-i})}{P_{NA}(x_i^{max} | x_{-i})} \tag{Eq. (3.4)}$$

- This procedure is repeated until a proposal step is accepted.

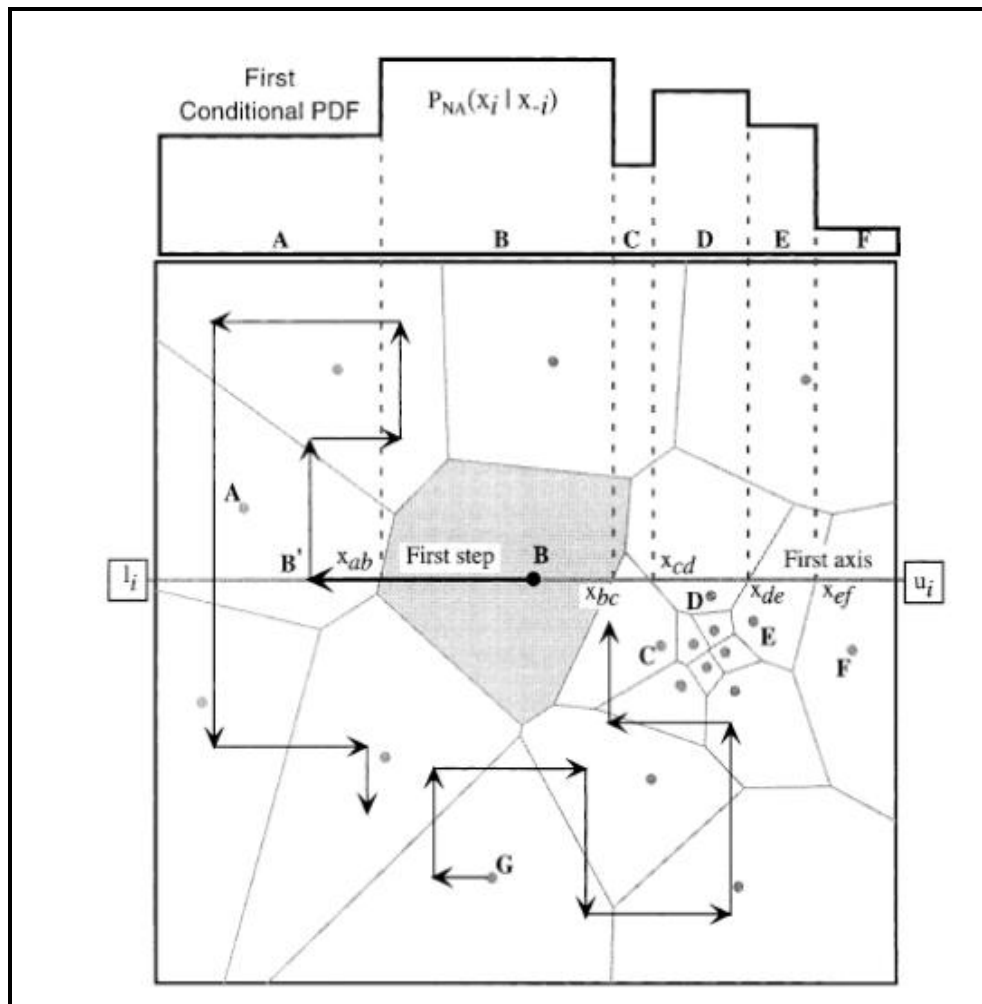


Figure 3.5: NAB resampling (Source: Sambridge (1999b))

3.4.2 Markov Chain Monte Carlo (MCMC)

The Metropolis-Hastings algorithm (Hastings, 1970) is an extension of simulated annealing via the Metropolis algorithm (Metropolis et al., 1953). It is a Markov chain Monte Carlo (MCMC) method that constructs a Markov chain that converges to the posterior probability thus producing an ensemble of model solutions rather than a single one. Accelerated by progress in approaches and technology for posterior computation, the capacity of these MCMC approaches has broadened significantly. The main driving

forces of these advances have involved novel approaches for semiautomatic prior realism and posterior exploration. The Markov chain Monte Carlo approach for uncertainty quantification is provided in detail in Chapter 6.

3.4.3 Ensemble Kalman Filter (EnKF)

Data assimilation methods calibrate a number of estimates of model parameters sequentially to points on a time-series of observed data. In calibration methods, on the other hand a complete run of the simulation is carried out and the match quality to the historical production data is used to move the estimates of the model parameters towards a better solution. The main data assimilation method used for history matching in the oil industry is the Ensemble Kalman Filter (EnKF). Evensen (2007) explained the theory of EnKF and shows a number of applications of EnKF to history matching field examples. Liu and Oliver (2005) showed that the EnKF compared favourably with gradient based methods when applied to history match a truncated Gaussian geostatistical model of facies distributions. The number of applications of EnKF is growing rapidly, with several papers presented at the 2009 and 2011 SPE Reservoir Simulation Symposium. Examples of developed methods include Particle Filtering Methods (Chen and Oliver, 2010; Evensen, 1994).

The Ensemble Kalman Filter (EnKF) is a Monte Carlo data assimilation method that was introduced by Evensen (1994) as an extension to the earlier developed Kalman Filter (Kalman, 1960). EnKF has gained popularity as an efficient filtering technique and has seen much success in many applications including atmospheric and ocean modelling, weather forecasting, and petroleum. The first application to reservoir modelling was attributed to Nævdal et al. (2002). It has been since then extensively employed as a reservoir characterisation and history matching tool in the literature (Aanonsen et al., 2009). A brief overview of the approach is provided below.

The EnKF approach consists of a forecast step (stepping forward in time) and an assimilation/analysis step, in which variables characterising the state of the system are corrected to honour the observations. EnKF is a sequential data assimilation method where observations are incorporated in time.

In reservoir modelling applications, we can represent all the reservoir model parameters that are uncertain as a state vector which contains all static uncertain parameters such as

porosity, permeability and net-to-gross and dynamic data predictions like pressure, water and gas saturations, and solution gas-oil ratio in each grid cell. Predicted data may include well bottom-hole pressures, water cuts, gas-oil ratio, and water-oil ratio values. Assume the state vector is denoted as in Eq. (3.5).

$$\mathbf{y} = (Y_1, Y_2, Y_3, \dots, Y_N) \quad \text{Eq. (3.5)}$$

where each member Y_i stands for a state parameter like porosity or pressure in a specified grid cell. The error covariance matrix is defined by

$$C_{YY} = \overline{(\mathbf{y} - \mathbf{y}_t)(\mathbf{y} - \mathbf{y}_t)^T} \quad \text{Eq. (3.6)}$$

where \mathbf{y}_t is the state vector corresponding to the truth. Suppose $\mathbf{d} = (d_1, d_2, d_3, \dots, d_m)^T$ refers to the m observations of the system and suppose H is a matrix containing zeros and ones defined as following

$$d = HY^t + \varepsilon \quad \text{Eq. (3.7)}$$

where ε is the measurement error. The assimilation equation minimising the error covariance C_{YY} based on the assumption that the measurements are Gaussian and independent is given by

$$Y^a = Y^f + K(d - HY^f) \quad \text{Eq. (3.8)}$$

In Eq. (3.8), the matrix $K = C_{YY}H^T(HC_{YY}H^T + C_D)^{-1}$ is called the Kalman gain and with the EnKF, the covariance matrix C_{YY} is estimated from the ensemble of N_R realisations $Y = (\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \dots, \mathbf{y}_{N_R})$. The matrix $C_D = \overline{\varepsilon\varepsilon^T}$ represents the covariance matrix of the observation data. The finite size of the ensemble limits the performance of the EnKF to the initial realisations due to the inherent uncertainty in the prior geological description. The change in covariance shows the models' sensitivity to the individual parameters. In Eq. (3.8), Y^f refers to the model state before assimilation while Y^a stands for the model state after the assimilation step. The improvement of the history matches can be observed by a comparison with the initial realisations forecast of pressure and flow rates.

The practical EnKF workflow can be summarised as follows:

1. Generate an ensemble of model realisations e.g. 100 initial realisations represent the prior knowledge of the initial state of the system and its probability distribution

2. Run the simulator for each realisation up to a timestep where the first observation is available
3. Approximate the correlation between the model parameters and model predictions
4. Correct the parameters in each realisation proportionally to:
 - a. The difference between predicted and observed data
 - b. The correlation between each parameter and the prediction
5. Run until the next observation is available.
6. Repeat steps 3–5.

One of the main reasons the EnKF is appealing is that since each member of the ensemble can be simulated independently, the forecast step is naturally parallel. Furthermore, the linear algebraic equations defined above are computationally inexpensive compared to the forward reservoir simulation (Evensen, 2007).

3.5 Comparison of Uncertainty Quantification Methods

A number of papers discussed comparisons of methods including some MCMC methods (Oliver et al., 2008), ensemble Kalman filter data assimilation method (Aanonsen et al., 2009) and stochastic sampling methods in Mohamed et al. (2010b). A review of methods from a statistical viewpoint can be found in Oliver and Chen (2010). Table 3.1 shows a comparison of advantages and disadvantages of five history matching methods utilised in petroleum literature.

Table 3.1: Comparison of advantages and disadvantages of five history matching methods (adapted and modified from Source: (Oliver and Chen, 2010))

Class	Advantages	Disadvantages
Manual	Great flexibility for parameters and data	Poor assessment of uncertainty. Large manpower requirement. Not suitable for large numbers of variables. Cannot get detailed matches.
Adjoint/Gradient	Rapid convergence near minimum. Relatively efficient for a single history match	Uncertainty assessment requires multiple history matches. Difficult to adapt to different simulators or variables.
MCMC	Statistically sound and accurate methods if worked.	Slow mixing and convergence. Simple MCMC variants are prohibitively expensive due to random walk making it difficult to use in practical applications. Chapter 6 and 8 show advanced MCMC methods to overcome these aspects.
Evolutionary algorithms	Highly parallelisable. Suitable for discrete parameters as well as continuous ones. Suitable for highly non-Gaussian distributions. Easily adaptable to various simulators	Slow convergence. Not suitable for large numbers of variables (possible solutions discussed in Chapter 9). Need separate calculations to computer probability of model obtained.
EnKF	Highly parallelisable. Suitable for large numbers of variables. Uncertainty assessment is a by-product of assimilation. Easily adaptable to different simulators and variables.	Generally underestimates uncertainty. Requires additional parameterisation to adapt to discrete variables. Not well suited for variables multimodal distributions unless transformations are possible.

3.6 Chapter Summary

In this chapter we have carried out an overview of objective function definitions, history matching and uncertainty quantification concepts with a review on the assisted history matching techniques and uncertainty quantification methods which have included global optimisation methods and the particle filter approach. The Bayesian methodology and uncertainty quantification framework used in the thesis and other topics that set the basis for the next chapters have been provided. A comparison of methods, their advantages and their limitations have been briefly outlined as motivation for this research.

Chapter 4 – History Matching and Uncertainty Quantification of Reservoir Simulation Models with Particle Swarm Optimisation

History matching optimisation in a Bayesian framework is a fairly recent approach (Valjak, 2008) to quantifying uncertainty in oil industry practices in which multiple history matched simulation models are obtained and used to estimate ranges in likely recovery factors. While some innovative global optimisation approaches have gained popularity in research among oil companies for tackling history matching problems, some of the existing assisted history matching methods have limitations in how fast they can obtain these models, how realistic the models are, and how reliable the model forecasts are.

This chapter is devoted to one of the techniques that belong to the so-called swarm intelligence algorithms for solving nonlinear global optimisation problems. Swarm intelligence (SI) is an innovative distributed intelligent paradigm in which the cooperative social behaviours of simple individual particles or agents cooperate locally and with their environment. SI integrates swarming intelligent behaviours by simulations of the social behaviour of swarms of bees, flocks of birds, colonies of ants, schools of fish, and human social behaviour (Bonabeau et al., 1999; Holland, 1998; Kennedy and Eberhart, 2001). The individual particles follow very simple rules, and interactions between them result in the emergence of "intelligent" global behaviour.

Swarm intelligence natural search algorithms try to find optimal regions of complex parameter spaces through the communication of individuals in a swarm of particles (Clerc, 2006). Swarm intelligence-based techniques include among others: stochastic diffusion search, bee colony optimisation, ant colony optimisation, and particle swarm optimisation. The methods have exhibited a good performance across a wide range of applications (Abraham et al., 2006; Chang et al., 2004; Du et al., 2005; Parsopoulos and Vrahatis, 2002b; Schutte and Groenwold, 2005; Sousa, et al., 2004; Ursem and Vadstrup, 2004).

This chapter introduces the Particle Swarm Optimisation (PSO) (Kennedy and Eberhart, 1995) as one of the swarm intelligence algorithms for solving nonlinear ill-posed inverse problems. The chapter shows the development of simple useful variants of novel Particle Swarm Optimisation (PSO) to address the question of how to tune PSO and adjust it to make it efficient in the history matching optimisation context. The variants have the flexibility in converging quickly towards good solutions as well as carrying out global exploration depending on the choice of the task and the variant choice. Part of the work carried out here in particle swarm optimisation (PSO) is reported in Mohamed et al. (2009, 2010d).

Contemporary and paralleled work on PSO has been applied by others. Kathrada and Carter (2010) tested a variant of the PSO called Flexi-PSO (Kathrada, 2009a, 2009b), coupled with hierarchical clustering on synthetic history matching problem though the uncertainty was not quantified in the study. García-Gonzalo and Fernández-Martínez (2010) applied PSO to environmental geophysics and petroleum reservoir engineering. Onwunalu and Durlofsky (2010) have tested PSO for a well placement problem and have compared the results with the Genetic Algorithm (GA). Ravalec-Dupin et al. (2010) used the PSO approach on a synthetic application to modify spatial distributions of petrophysical properties from the gradual deformation method and identify facies reservoir models.

The chapter is organised as follows. We begin with definitions of the main components of the algorithm. We will outline the original particle swarm optimisation and workflow described by Kennedy and Eberhart (1995). The main adaptations and variants investigated in this thesis will also be presented. A discussion of the implementation of the algorithm for generating reservoir models follows and we will show the application

of PSO for reservoir model history matching and uncertainty quantification on two petroleum examples: a simple real reservoir in the Gulf of Mexico and a complex synthetic example. In these examples, it is demonstrated that algorithms based on swarm intelligence concepts have the potential to be effective tools in uncertainty quantification in the oil industry. A comparison of results in terms of the model diversity and quality of history match obtained on the example problems will be presented. Finally, conclusions and guidelines from the examples to facilitate deployment on the algorithm will be commented on.

4.1 Particle Swarm Optimisation (PSO)

The particle swarm optimisation algorithm (PSO) is a swarm intelligence technique originally introduced by Kennedy and Eberhart (1995). PSO is a population-based metaheuristic stochastic optimisation technique inspired by social behaviour of bird flocking or fish schooling. PSO has proven to be a powerful contender to other population based evolutionary algorithms for global optimisation problems (Matott et al., 2006). PSO has been successfully applied in a variety of fields, research and application areas, which include fuzzy computing, chaos theory, and engineering applications (Kennedy and Eberhart, 1995; Eberhart and Shi, 2001). It is demonstrated that PSO gets better results in a faster and cheaper way compared with other stochastic techniques like GA for problems investigated in Mouser and Dunn (2005) and was much easier to configure and more likely to produce an acceptable model. Another reason that PSO is attractive is that it has a small number of parameters to adjust. PSO has both simple formulation and computer implementation.

PSO models the exploration of a parameter space by a population of particles or agents that fly through the search space by following the current optimum particles. The position of a particle is a candidate solution to the optimisation problem. The particles' history of success affects their own exploration pattern and those of their peers. The search is focused toward promising regions by biasing each particle's velocity vector towards their own remembered best positions as well as the best swarm position.

4.1.1 Basic PSO Algorithm

The basic PSO algorithm starts with the random initialisation of a swarm of particles in the search space. Each particle is considered as a candidate solution to a problem in d -dimensional space, with the position of particle i represented by x_i . Each particle

maintains a memory of its previous best position, $pbest_i$, and a velocity along each dimension, represented as v_i . The $pbest$ vector of the particle with the best fitness in the neighbourhood is designated $gbest$. The importance of these two positions, $gbest$ and the $pbest_i$, is weighted by two factors known as the cognitive and social scaling factor parameters at each iteration (Shi and Eberhart, 1998). These two elements are among the main governing parameters of swarm behaviour and algorithm efficiency, and have been the subject of extensive studies (Kennedy, 1997, 1998; Suganthan, 1999). The algorithm convergence has been investigated by a number of authors (Engelbrecht, 2005).

4.1.1.1 Velocity Update

In the basic PSO algorithm, at each iteration k , particle i 's velocity v_i^k is updated using Eq. (4.1).

$$v_i^{k+1} = v_i^k + c_1 r_1 (pbest_i^k - x_i^k) + c_2 r_2 (gbest^k - x_i^k) \quad \text{Eq. (4.1)}$$

in which x_i^k refers to the current position of a particle i at iteration k . c_1 and c_2 are user-defined non-negative constant real parameters which weight the particle's attraction towards its own best known position $pbest_i^k$ and the global best known position of the entire swarm $gbest^k$ up to iteration k , respectively. r_1 , and r_2 are two random vectors with each component corresponding to a uniform random number between 0 and 1. The introduction of such random elements into the optimisation is intended to simulate the slightly stochastic unpredictable component of natural swarm behaviour. In addition to this, the user also chooses the swarm size N .

The velocity update in Eq. (4.1) has three major components (Engelbrecht, 2005). Figure 4.1 illustrates the velocity update mechanism in PSO.

1. The first component of the velocity update equation, referred to as inertia, models the tendency of the particle to continue in the same direction it has been moving.
2. The second component of the velocity update equation, referred to as memory, is a linear attraction towards the best position ever found by the particle.
3. The third component of the velocity update equation, referred to as social knowledge, is a linear attraction towards the best position found by any particle.

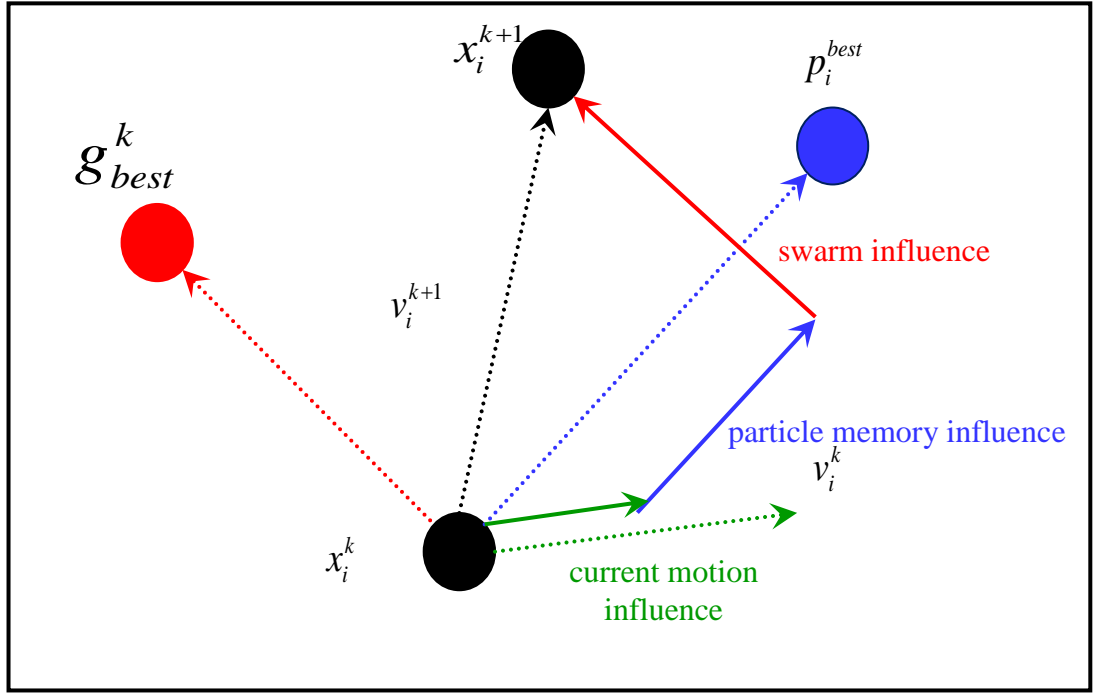


Figure 4.1: Velocity component construction

4.1.1.2 Position Update

The particle's position is added to the particle's velocity once it has been calculated to determine the new position of the particle.

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad \text{Eq. (4.2)}$$

The particle's position is therefore updated regardless of progress to its objective function.

The update equation of the personal best position $pbest_i$ is presented in Eq. (4.3), assuming a minimisation problem where f denotes the objective function that is being minimised and k is the iteration (generation) number.

$$pbest_i^{k+1} = \begin{cases} pbest_i^k & \text{if } f(x_i^{k+1}) \geq f(pbest_i^k) \\ x_i^{k+1} & \text{if } f(x_i^{k+1}) < f(pbest_i^k) \end{cases} \quad \text{Eq. (4.3)}$$

The main computational PSO workflow is described in the following steps:

1. Initialise the swarm of n_{init} models by assigning at locations randomly generated in parameter space. Each particle is also assigned a plausible random velocity.
2. For each model (particle) the forward problem is solved and the relevant objective function value M is obtained.

3. For each particle, update the position and value of $pbest$ – the best solution the particle has seen. If the current objective function value of one particle is better than its $pbest$ value, then its $pbest$ value and the corresponding position are replaced by the current objective function value and position, respectively as in Eq. (4.3).
4. Find the global best objective function value and the corresponding best position $gbest$ across the whole swarm's $pbest$ and update if appropriate.
5. Update the velocities and positions of all the particles using Eqs. (4.1) and (4.2) where c_1 is a weighting factor, termed the cognition component which represents the acceleration constant which changes the velocity of the particle towards $pbest_i$, c_2 is a weighting factor, termed the social component which represents the acceleration constant which changes the velocity of the particle towards $gbest^k$.
6. Repeat steps 2–5 until a stopping criterion is met (e.g. the maximum number of iterations is reached or a sufficiently good objective function value).

4.1.2 Topology of the Particle Swarm

There are two general neighbourhood topologies used commonly in PSO: 1) global best (gbest) and 2) local best (lbest). In the gbest neighbourhood, each particle is influenced by the best solution found from the entire swarm. It is a star topology, in which each particle has access to the information of all other particles in the swarm, as shown in Figure 4.2(a). In the local best approach, each particle is influenced only by particles in its local neighbourhood and has access only to the information corresponding to its direct neighbours. The two most common topologies are the wheel topology in which the individuals are isolated from one another as information has to be communicated through a focal individual (see Figure 4.2(b)), and the ring topology in which each particle connected to two neighbours, as shown in Figure 4.2(c). In the application of the thesis a global neighbourhood is used when exchanging information about swarm best values and positions. Furthermore, the synchronous parallelisation scheme is used in which the swarm best value, particle best remembered positions, velocities and fitness values are updated on a per swarm basis, rather than a per individual basis. Therefore, fitness evaluations involving reservoir simulations are concurrently performed on different processors. The particle positions are updated after synchronising results from the participating processors at the end of each iteration. The solution characteristics, parallel speed-up and efficiency as well as maintaining load

balance between processors can be further improved with other parallelisation schemes (Kalivarapu et al., 2009; Koh et al., 2006; Schutte et al., 2004; Venter and Sobieszczanski-Sobieski, 2006).

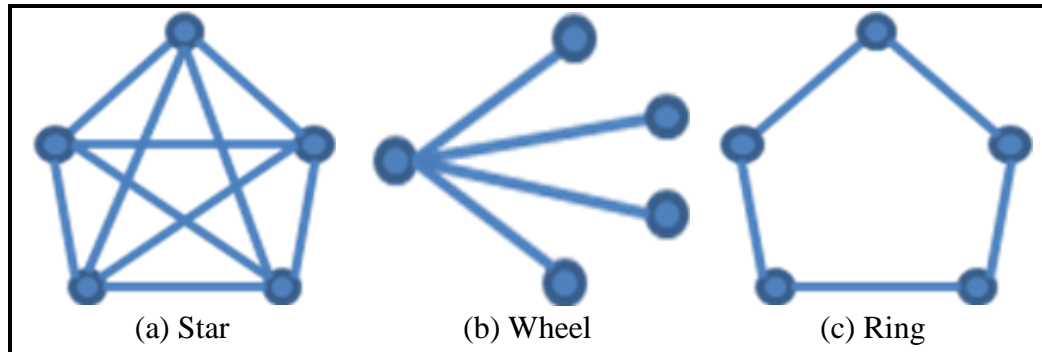


Figure 4.2: PSO common topologies

4.1.3 Particle Swarm Variants

Ever since the start of PSO in 1995 a substantial number of adaptations have been made to the basic algorithm despite its wide use and popularity for realising performance improvements. For example, one of these variations involves looking at the presence of problem-dependent algorithm parameters. Some researchers have attempted to establish universal values for the PSO parameters (Carlisle and Dozier, 2001) primarily on analytical optimisation problems. As a result the PSO has undergone rapid development, with several modifications to improve the speed of convergence and enhance the performance of the algorithm. These variants include the introduction of velocity clamping, an inertia weight, velocity constriction coefficient type, different approaches of finding out the personal best and global best (or local best) positions as well as with dynamic neighbourhood topologies, different velocity update rules, enhanced diversity variants, and with components from other optimisation techniques. Sophisticated variants ideas and approaches have been described in detail within the particle swarm literature (Eberhart and Shi, 2007; Engelbrecht, 2005). Pedersen and Chipperfield (2009) have recently shown another research trend by simplifying the original PSO method. They have reported its success over others methods. This section discusses the basic modifications concerning the inertia weight choices and various strategies to control any particles which fly outside the predefined feasible regions by constraint handling strategies.

4.1.3.1 Velocity Clamping

To reduce excessively large stepsizes which lead to particles leaving the boundary of the search space in the position update Eq. (4.2), Eberhart and Shi (2001) applied an imposed upper limit on the maximum velocity v_{max} for particles. Initially the values of the velocity vectors are randomly generated with $v_i^{k=0} \in [-v_{max}, v_{max}]$. Usually v_{max} is problem-dependent and chosen to be a fraction of the domain of each dimension of parameter space. In our implementation we used 0.5 consistent with Schutte et al. (2004).

4.1.3.2 Inertial Weight Choices

A significant aspect that determines the efficiency and accuracy of an optimisation algorithm is the exploitation-exploration trade-off. The inertial weight was introduced by Shi and Eberhart (1998) in the velocity update equation Eq. (4.4), as a mechanism to control the exploration and exploitation ability of the swarm. It monitors the way in which a great deal of memory of the past flight vector will affect the new velocity and influences the convergence of the algorithm. Large values for ω facilitate exploration of different regions of the search space in order to locate good optimum, with increased diversity. A small ω enhances local exploitation on a promising area in order to refine a potential solution. With respect to $\omega \geq 1$, velocities rise up gradually over time, increasing towards the upper limit velocity causing the swarm to diverge. Particles cannot change direction so as to move backwards towards potential regions. For $\omega < 1$, particles slow down until their velocities reach zero.

$$v_i^{k+1} = \omega v_i^k + c_1 r_1 (pbest_i^k - x_i^k) + c_2 r_2 (gbest^k - x_i^k) \quad \text{Eq. (4.4)}$$

The optimal value for the inertial weight is problem-dependent. Some implementations of the inertial weight use a static value for all particles for each dimension for the entire course of optimisation. Some static inertia weight examples being used (Birge, 2003) are Trelea Set Type I' (denoted T1) in which $c_1 = c_2 = 1.494$, $\omega = 0.729$ and Trelea Set Type I'' (denoted T2) in which $c_1 = c_2 = 1.7$, $\omega = 0.6$ (Trelea, 2003). Alternatively, dynamic approaches which vary the inertial weight during the search duration could be used such as linear decreasing (denoted LD) in which an initially large inertia weight (usually 0.9) is linearly decreased over time to a small value (usually 0.4). In this way, particles can explore in the initial optimisation course, and then refine potential regions as time increases (Shi and Eberhart, 1998). On the other hand, in the linear increasing

inertia weight variant (denoted LI) the inertia weight is linearly increased from 0.4 to 0.9. The selection of value for ω has to be made in combination with the selection of the values for c_1 and c_2 . (Kennedy and Eberhart, 1995) suggested using $c_1 = c_2 = 2$ to allow the product $c_1 rand_1$ or $c_2 rand_2$ to have a mean of 1. In this case the particles overshoot the target half the time, thus maintaining partition within the grouping and allowing for a greater area to be searched. If $c_1 + c_2 \geq 4$, velocities and positions will go towards infinity (Schutte, 2001). Other variants for tuning the inertial weight or the controlling weights include constant inertial weight, random adjustments like Gaussian and Peng options, adaptive inertial weight motivated by Clerc, and fuzzy adaptive inertia. Clerc developed a dynamic approach equivalent to inertial weight which differs in the constriction parameters, does require velocity clamping and guarantees convergence under given constraints (see Engelbrecht, 2005). In this thesis, we focused on the application of investigated variants that seem to provide plausible and more stable results on our petroleum examples.

The pseudo-code for particle swarm is given in Algorithm 4.1. The ‘‘For’’ loop is achieved in parallel mode to speed up the computation processing in reservoir modelling applications.

Algorithm 4.1: Particle Swarm Optimiser

1. Initialise $t = 0$, Swarm, $SwarmSize$, $pbest_i$, $gbest$, c_1 , c_2 , ω
2. For $k = 1$ to $NumberOfIterations$ do
3. Update ω # Update inertial weight
4. For $i = 1$ to $SwarmSize$ do
5. # Update position and velocity
6. $v_i^{k+1} = \omega v_i^k + c_1 r_1 (pbest_i^k - x_i^k) + c_2 r_2 (gbest^k - x_i^k)$ #Update velocity
7. $x_i^{k+1} = x_i^k + v_i^{k+1}$ # Update position
8. Evaluate Objective Function
9. If $f(x_i^{k+1}) \geq f(pbest_i^k)$ then
10. $pbest_i^{k+1} = pbest_i^k$ # Update personal best $pbest$
11. Else
12. $pbest_i^{k+1} = x_i^{k+1}$
13. End If
14. End For
15. Update global best $gbest$
16. $k++$
17. End For
18. Return $\{x_i, v_i\}_{i=0}^{Nmodels}$

4.1.3.3 Handling Boundary Strategies for Particle Swarm

Various boundary conditions are proposed to keep particles within the search domain. The four boundary strategies that are used in literature (Birge, 2003; Huang and Mohan, 2005; Schutte et al., 2004) and we currently found useful among others are:

1. Respawn strategy (random): when the particle moves outside the feasible parameter space in one of the dimensions, it is randomly reinitialised with a sensible random velocity component in that dimension, as shown in Figure 4.3(a).
2. Reflecting strategy (bouncing): when the particle moves outside the feasible parameter space in one of the dimensions, it is repositioned at the boundary of parameter space in that dimension, and the sign of the velocity component in that dimension is changed, as shown in Figure 4.3(b).
3. Damping strategy: when the particle moves outside the feasible parameter space in one of the dimensions, it is repositioned at the boundary of the parameter space in that dimension, and the velocity component in that dimension is damped in the opposite direction with a fraction of velocity that can be obtained by multiplying velocity by a random number between 0 and v_{max} , as shown in Figure 4.3(c).
4. Absorbing strategy (saturation at limit): when the particle moves outside the feasible parameter space in one of the dimensions, it is repositioned at the boundary of parameter space in that dimension, and the velocity component in that dimension is zeroed, as shown in Figure 4.3(d).

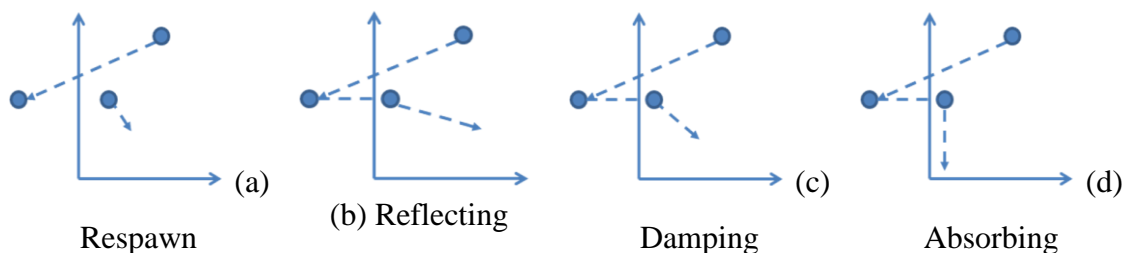


Figure 4.3: Boundary strategies for particle swarms

The acronyms in Table 4.1 are the different variants that will be used in the analysis and discussion in the thesis to differentiate between them. Each is composed of three letters. The first and second letters represent the inertial weight choice. The last letter represents the boundary strategy.

Table 4.1: Acronyms used to denote the PSO variants

Acronym	Name
LDR	Linear decrease inertia weight with random boundary strategy.
LIR	Linear increase inertia weight with random boundary strategy.
T1R	Trelea Set Type I' with random boundary strategy.
T2R	Trelea Set Type I'' with random boundary strategy.
LDB	Linear decrease inertia weight with bouncing boundary strategy.
LIB	Linear increase inertia weight with bouncing boundary strategy.
T1B	Trelea Set Type I' with bouncing boundary strategy.
T2B	Trelea Set Type I'' with bouncing boundary strategy.
LDD	Linear decrease inertia weight with damping boundary strategy.
LID	Linear increase inertia weight with damping boundary strategy.
T1D	Trelea Set Type I' with damping boundary strategy.
T2D	Trelea Set Type I'' with damping boundary strategy.
LDS	Linear decrease inertia weight with absorbing boundary strategy.
LIS	Linear increase inertia weight with absorbing boundary strategy.
T1S	Trelea Set Type I' with absorbing boundary strategy.
T2S	Trelea Set Type I'' with absorbing boundary strategy.

4.2 Efficiency Criteria for Evaluating the Algorithms

The plots which will be used to measure and compare the performance of the optimisation course for the PSO variants include: the generational minimum, the global best per generation, sampling history trails, and the diversity defined by Eq. (4.5) (Engelbrecht, 2005).

$$diversity(k) = \frac{1}{n_s} \sum_{i=1}^{n_s} \sqrt{\sum_{j=1}^d (x_{ij}^k - \bar{x}_j^k)^2} \quad \text{Eq. (4.5)}$$

In this equation k is the generation number, n_s is the swarm size, d is the dimensionality of the problem, and \bar{x}_j is the average of the j^{th} dimension over all particles. Large diversity values mean that a larger area of the parameter space has been explored (Engelbrecht, 2005). Note that in our applications in the thesis the diversity is calculated for the scaled parameter values. The notation ‘>’ will be used for comparison to denote: more diverse, faster convergence or wider uncertainty ranges

while ‘<’ will be used for the opposite, that is less diverse, slower convergence, or narrower uncertainty ranges.

4.3 Petroleum Test Studies

The algorithm has been tested on benchmark test suite functions to test the convergence of the simple variants investigated and ensure the consistency of the results in comparison with the results obtained previously in these benchmark problems. Tests are then being carried out on history matching problems. In the next section we report the results of obtaining multiple history matched models with the particle swarm optimisation variants indicated earlier. A comparative study of the four boundary strategies with each of the four inertial weight choices is conducted on two reservoirs. Firstly, we will compare the inertial weight choices and the four boundary strategies from a history matching application perspective. Uncertainty evaluation and prediction results then follow. Comparisons of simulation results are presented from two different points of view: the diversity of the models obtained and the efficiency of the algorithm and the impact on the forecasts in the two studies.

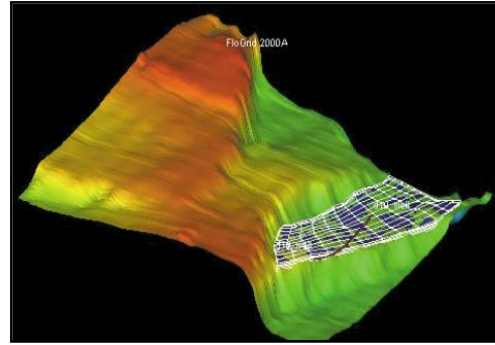
Two petroleum applications are used for testing the algorithms. Teal South is a real reservoir with one producing well. The second model, the IC Fault model, is a synthetic challenging benchmark case study that has been widely used for testing different optimisation algorithms for reservoir history matching. IC Fault has a single injector and single producer with fault affecting the fluid flow.

4.3.1 Teal South Reservoir

Teal South is a reservoir located in the Gulf of Mexico approximately 144km south west of Morgan City, Louisiana, shown in Figure 4.4. A 4500ft sand deposit is bounded on three sides by faults and closed by a dip to the north, shown in Figure 4.5. Fluids are produced from a single horizontal well through solution gas drive, aquifer drive and compaction drive. The Fetkovich model is used for describing water influx in the aquifer. Production started in late 1996 and data is available in the form of monthly oil, water and gas measurements, and two pressure measurements of 3096 psi initially and 2485 psi after 570 days of production (Christie et al., 2002). A number of automated history matching techniques have been investigated on Teal South reservoir (Christie, 2011; Christie et al., 2002; Mohamed et al., 2010b, 2010d) and also time-lapse studies by Texaco.



Figure 4.4: The Teal South Field Location

Figure 4.5: The Teal South 4500-ft sand structure map and the $11 \times 11 \times 5$ simulation grid

4.3.1.1 Teal South Model Uncertain Parameters

Teal South is a small reservoir located in the Gulf of Mexico. We used a relatively coarse simulation model with a grid size of $11 \times 11 \times 5$. We set up the model with 8 uncertain parameters: horizontal permeabilities of the five geological layers, a single value for k_v/k_h , rock compressibility and aquifer strength. Parameters are denoted from P1 to P8 respectively. We chose uniform priors in the logarithms of the variables as shown in Table 4.2.

Table 4.2: Parameterisation and prior ranges for the Teal South model

Parameter	Units	Prior range
k_h (for each of the 5 layers)	mD	$10^{[1,3]}$
k_v/k_h	–	$10^{[-2,-1]}$
Rock compressibility	psi^{-1}	$10^{[-4.096,-3.699]}$
Aquifer strength	$MMstb$	$10^{[7,9]}$

Figure 4.6 shows the observed production rates for oil and water, as a function of time for 1247 days. The oil rate peaked after 80 days of production and then declined rapidly. Water production started after the oil rate peaked and stayed steady for the majority of the time. The first 181 days of production data were used in the history matching (6 measurements out of 41 measurements) and the remaining 3 years is used as prediction data to measure the predictive quality of the history matches. The simulator production was controlled to match total liquid rate, and history matching was carried out by matching the field oil rate.

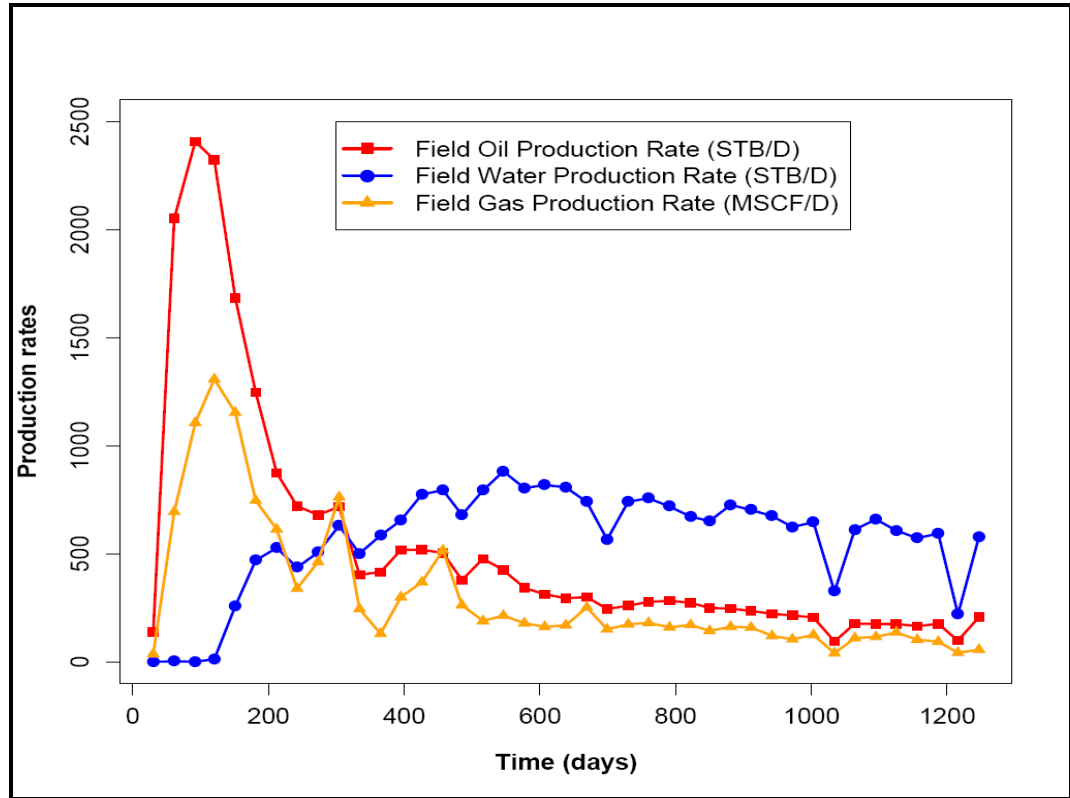


Figure 4.6: Production history for Teal South reservoir

4.3.1.2 Objective Function Definition

A least squares misfit is commonly used in history matching as the objective function to measure the goodness of fit of a specific set of reservoir model parameters as defined in Eq. (4.6).

$$M = \sum_{t=1}^T \frac{(q^{obs} - q^{sim})_t^2}{2\sigma^2} \quad \text{Eq. (4.6)}$$

Here T is the number of observations, q is the flow rate for observed and simulated data, and σ^2 is the variance of the observed data. This definition is based on the assumption that the measurement errors are Gaussian and independent. The standard deviation of the oil production measurement errors was set to 100 STB/D as an estimation of the oil production measurement errors on our reservoir model case study.

4.3.1.3 Teal South Setup Specifications

An initial population comprised of 30 models has been generated randomly in the parameter space. Figure 4.7 shows an example of two different 2D projections of the initial models. On the left we show the models plotted against the scaled values of two layer permeability multipliers (P1 and P2), and on the right the models are plotted against the scaled aquifer strength and the scaled rock compressibility multipliers. The

points are colour coded according to the misfit where the blue points represent the models of misfits of 6 or less. This setup is going to be used throughout the thesis for testing different methods unless stated otherwise.

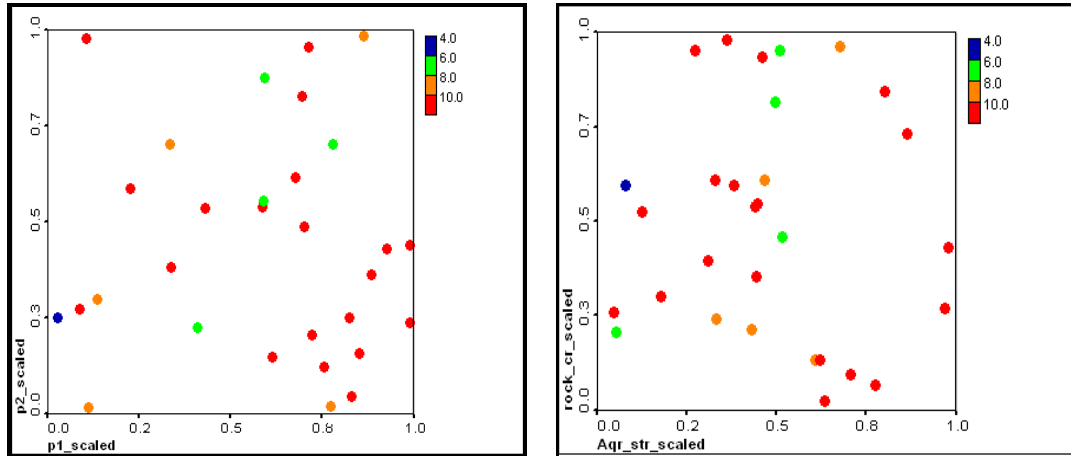


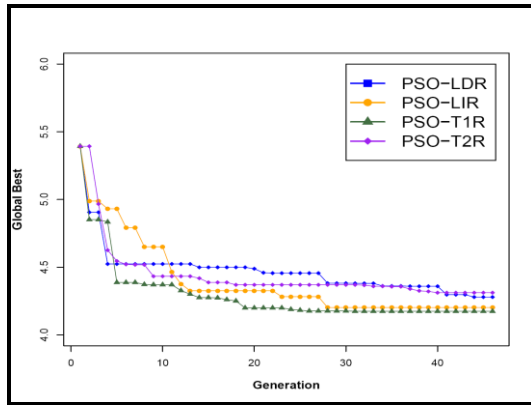
Figure 4.7: Two different 2D projections of initial population of 30 randomly generated models in 8D parameter space

4.3.1.4 History Matching of Teal South Reservoir

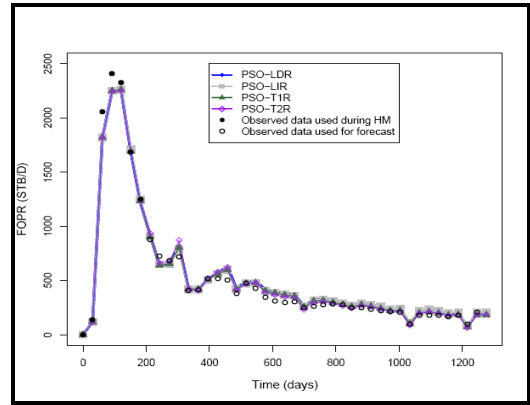
For all the variants we start from a fixed initial population comprising 30 models generated randomly in parameter space. The optimisation is done for 45 iterations. The total number of reservoir model simulations is 1380 for all the performed tests. The performed tests were repeated for 10 seeds.

4.3.1.4.1 Inertial Weight Choices

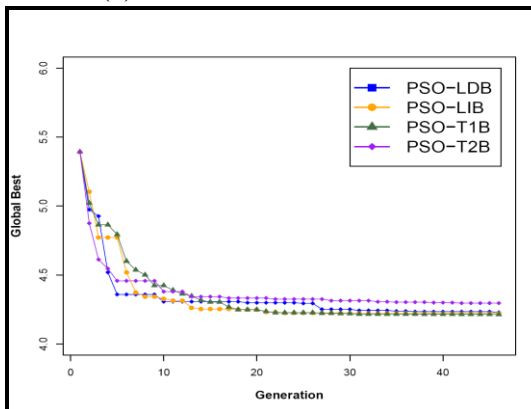
We compare the four inertial weight choices per each boundary strategy. In the application, the cognition and social components were chosen to be 2 for the dynamic inertial weight variants. The best fitting models obtained by the four variants in this history matching have a misfit value of around 4.2 (see Table 4.5: Page 82) for three strategies except for the random strategy variants: PSO-LDR and PSO-T2R (refer to acronyms in Table 4.1: Page 63 for description) which have a slightly higher misfit value of 4.3. Figures 4.8(a), (b), (c), and (d) show the misfit reduction of the global best for each boundary strategy per generation starting from the same initial population (we will call this seed hereinafter). Figures 4.8(e), (f), (g), and (h) show the corresponding best history matches obtained at the end of the optimisation course for each strategy. The plots show that the obtained models match the observed data before and after the history matching period plausibly well.



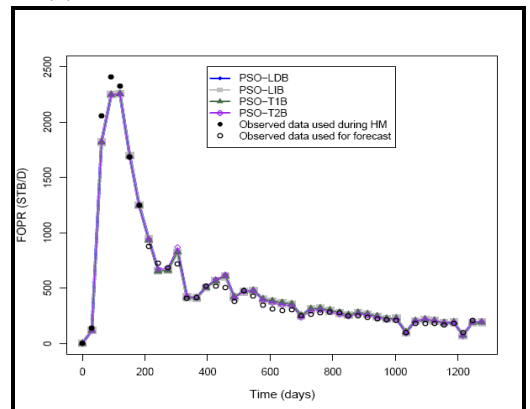
(a) Misfit reduction – Random



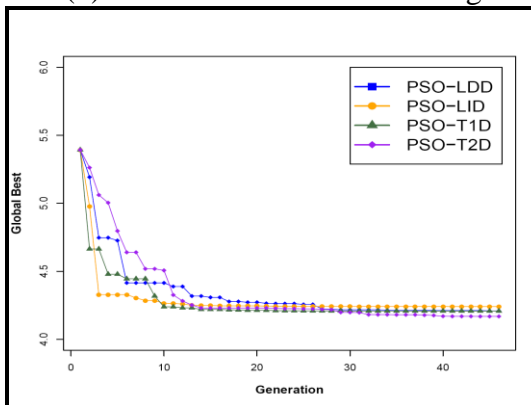
(e) Lowest misfit model – Random



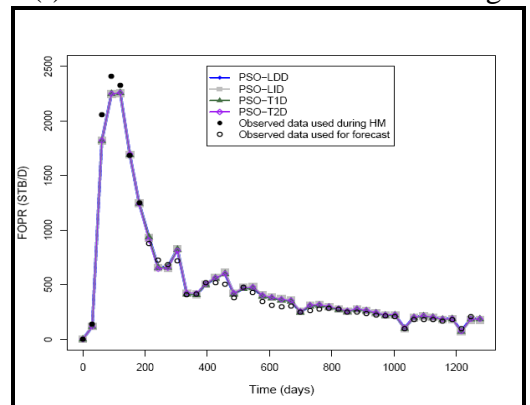
(b) Misfit reduction – Reflecting



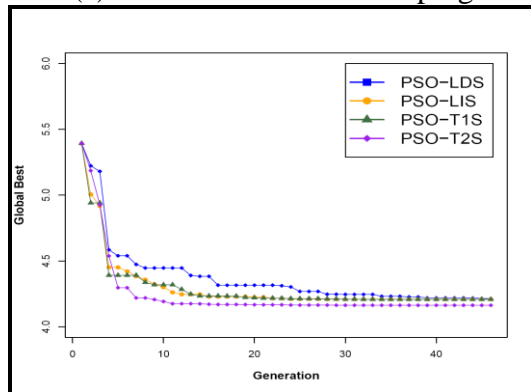
(f) Lowest misfit model – Reflecting



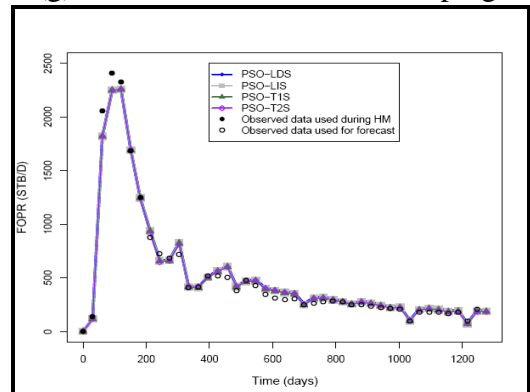
(c) Misfit reduction – Damping



(g) Lowest misfit model – Damping



(d) Misfit reduction – Absorbing

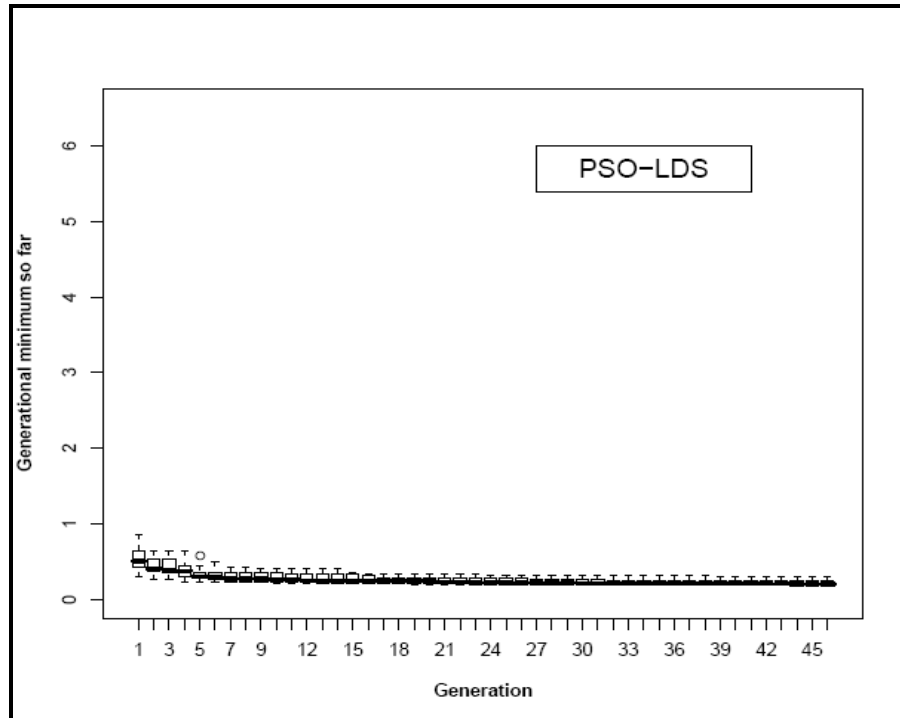


(h) Lowest misfit model – Absorbing

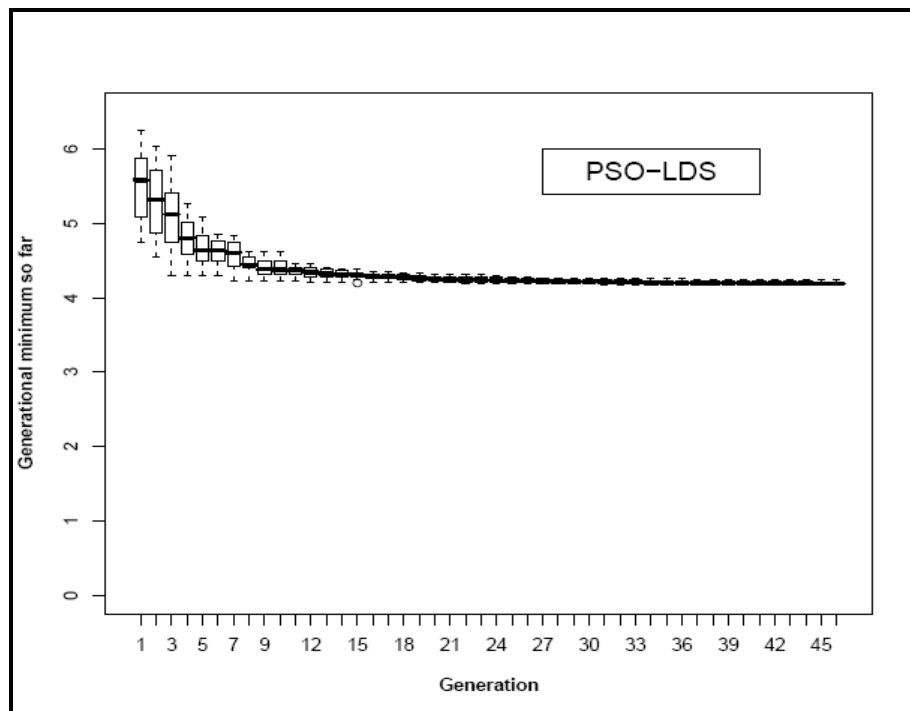
Figure 4.8: Misfit reduction and corresponding global best history match for variants

We note that around 93% of the errors between simulated and observed are obtained from the outliers in the 3rd (day 61 around 60%) and 4th (day 92 around 33%) timesteps, which seem to be unrealistic measurements, as shown in Figures 4.8(e), (f), (g), and (h) (note the index of the observation counts the initial step at zero to be the first observation). If we remove the two timesteps (3rd and 4th) for PSO–LDS, our optimisation process for the same setup obtains a misfit of around 3.0 (computed with the 4 measurements) as well as improving the speed of convergence to low misfit models as shown in Figure 4.9(a) where 10 runs were performed and convergence achieved where simulations could be stopped at around iteration three while the first case convergence achieved around iteration ten as depicted in Figure 4.9(b).

Sampling history plots to look at the performance during the optimisation course are shown in Figures 4.10, 4.11, 4.12, and 4.13 per strategy. Each plot has 8 panes, showing the evolution of the parameter sampling as we sample in time. The horizontal axis is the model number and the vertical axis shows the scaled values of each parameter multiplier between 0 and 1. The points are colour coded according to the misfit where blue points indicate the low misfit models of 4 or below. The red points have misfit 10 or above, and include many models that do not match at all well. The green points indicate models with misfit in the range [6,4) and the orange points indicate models with misfit in the range [8,6). As sampling advances in time, the concentration in promising regions with low misfit models is observed. The good fitting models generated have misfits of 6 or below – corresponding to an average deviation from observed values of 1.4 standard deviations or below.

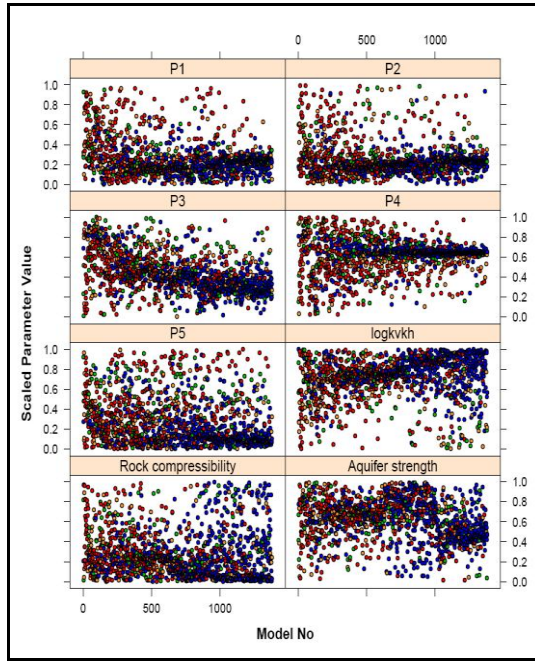


(a) Four measurements used in history matching after removing outliers

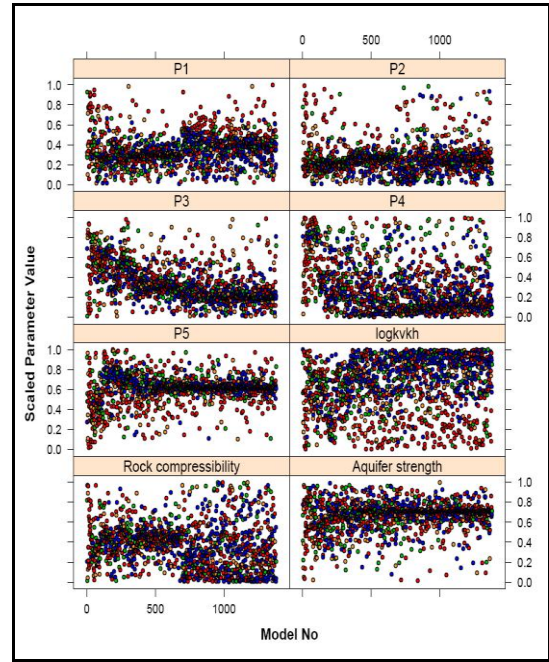


(b) Six measurements used in history matching

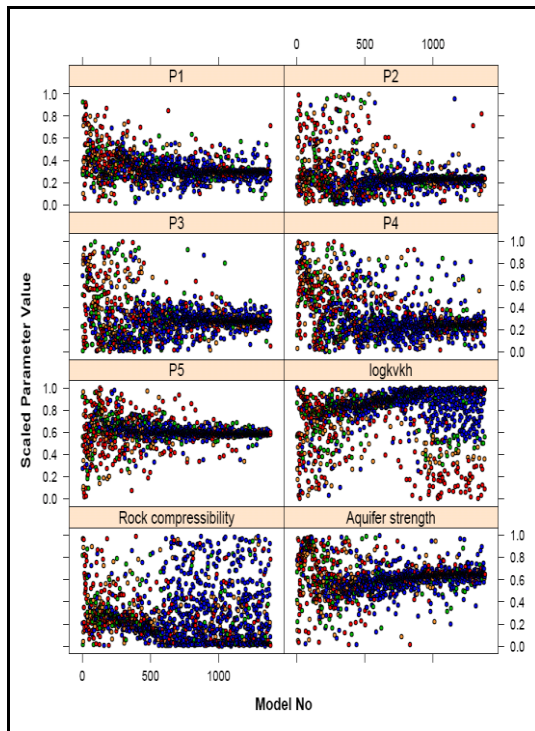
Figure 4.9: Misfit reduction after removing two outlier points (the 3rd and 4th)



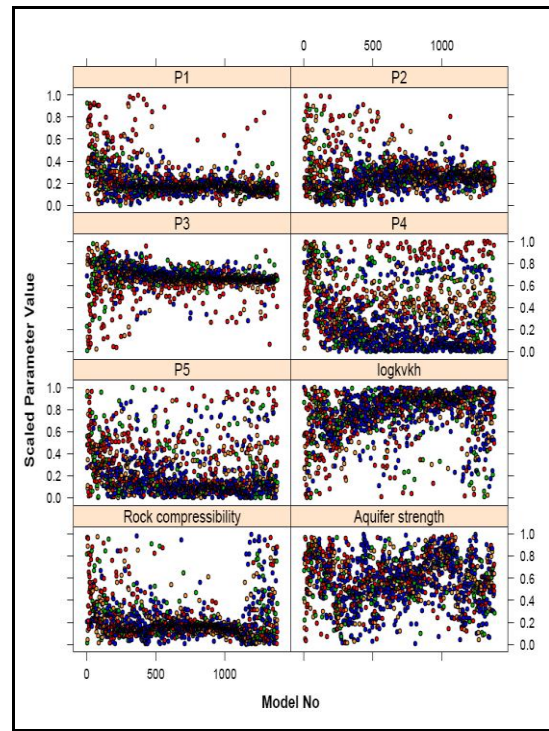
(a) Sampling history of PSO-LDR



(b) Sampling history of PSO-LIR

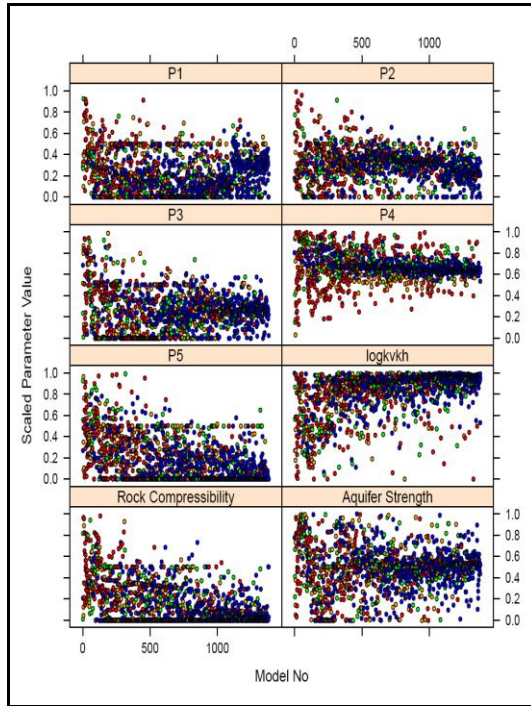


(c) Sampling history of PSO-T1R

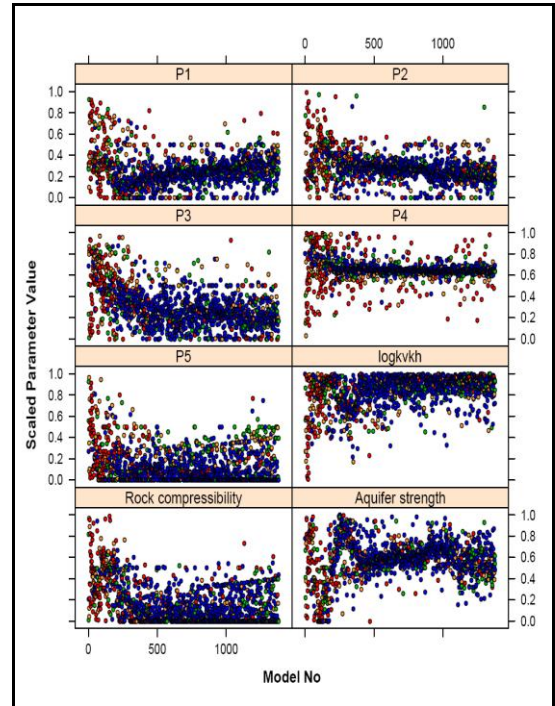


(d) Sampling history of PSO-T2R

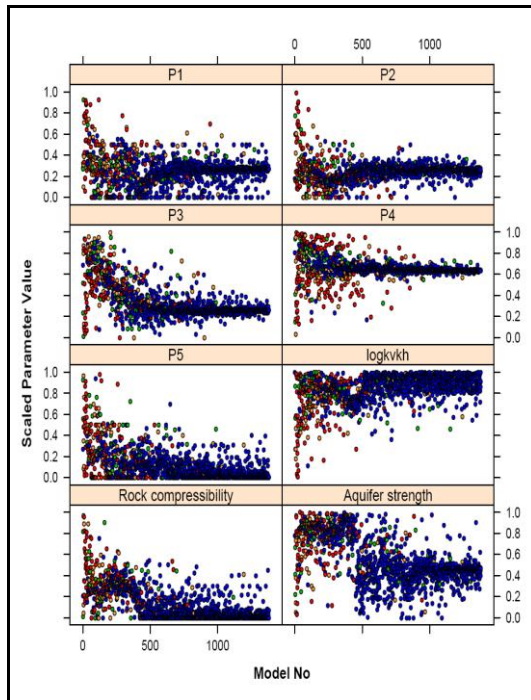
Figure 4.10: Random strategy sampling history for each inertial weight choice



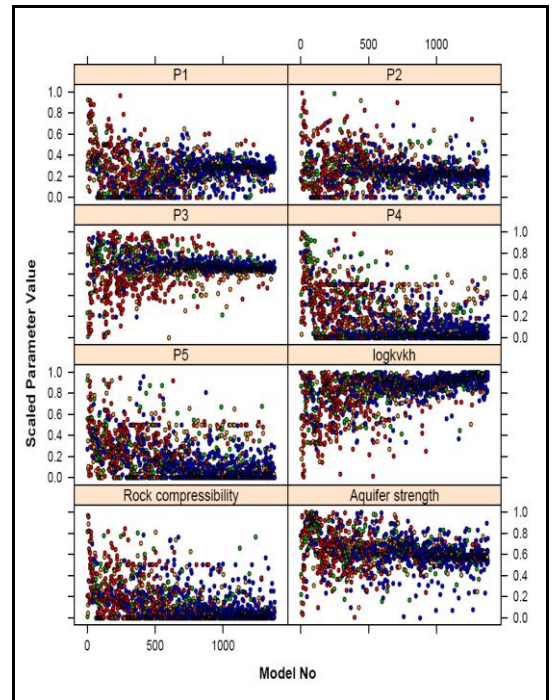
(a) Sampling history of PSO-LDB



(b) Sampling history of PSO-LIB

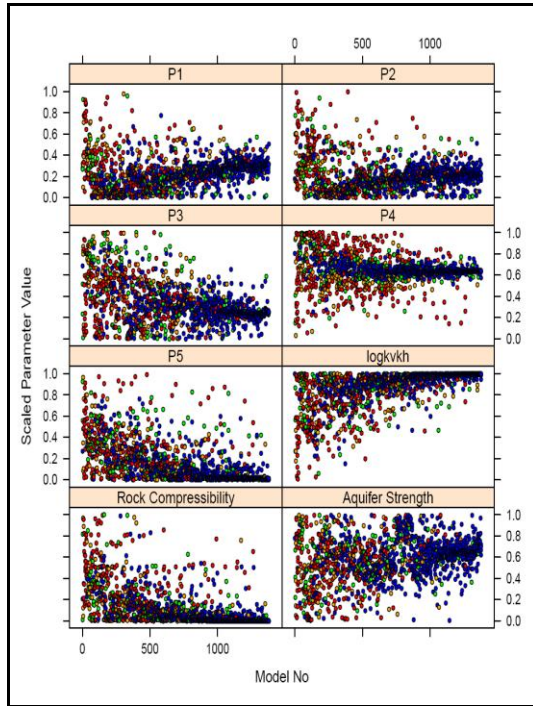


(c) Sampling history of PSO-T1B

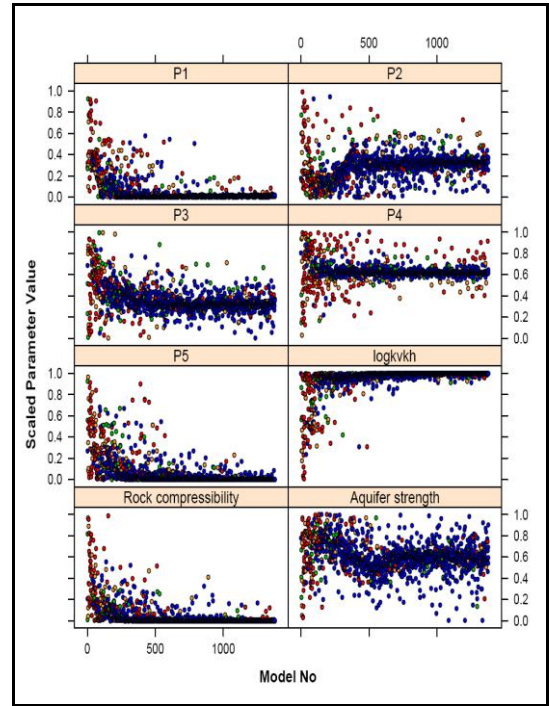


(d) Sampling history of PSO-T2B

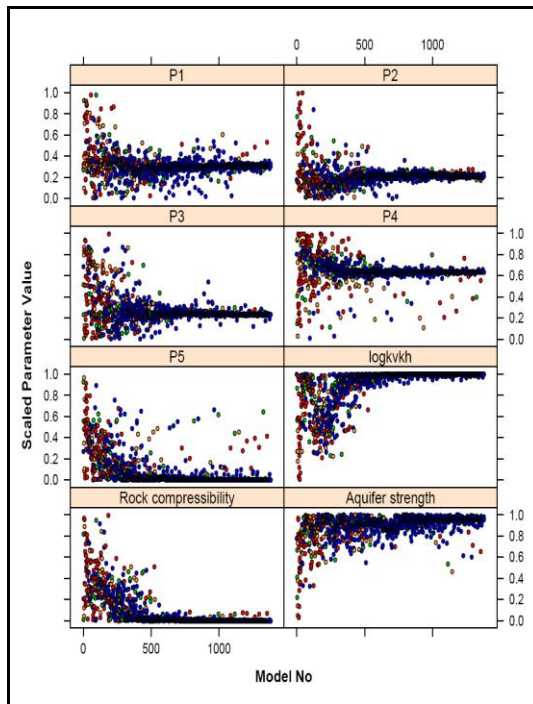
Figure 4.11: Reflecting strategy sampling history for each inertial weight choice



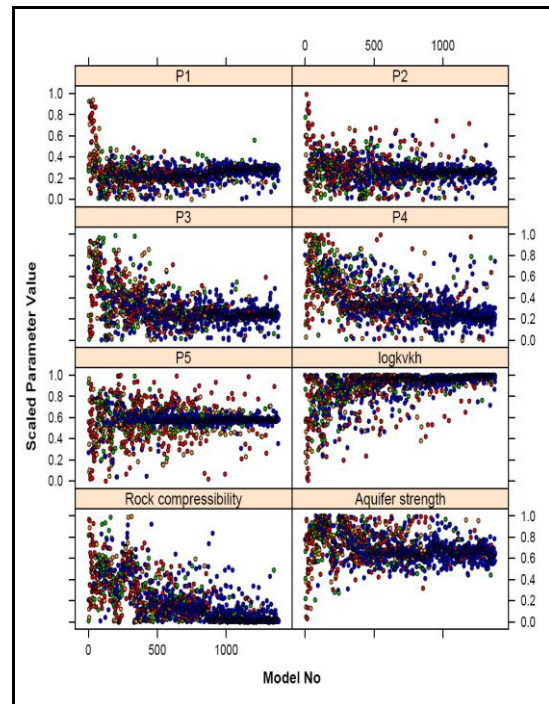
(a) Sampling history of PSO-LDD



(b) Sampling history of PSO-LID

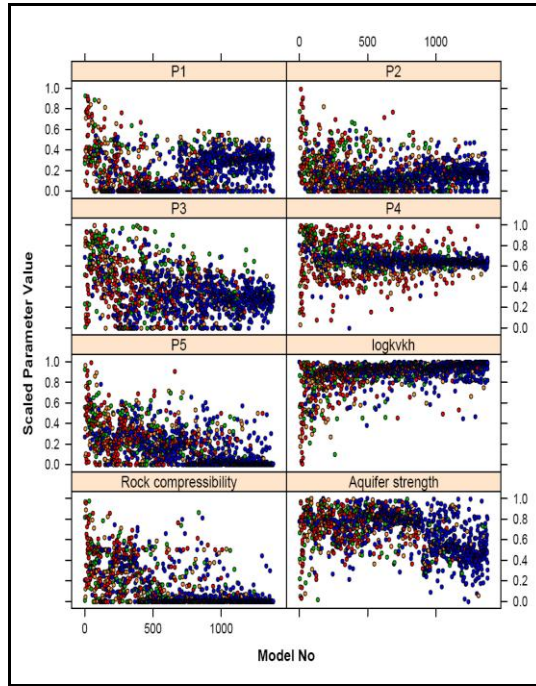


(c) Sampling history of PSO-T1D

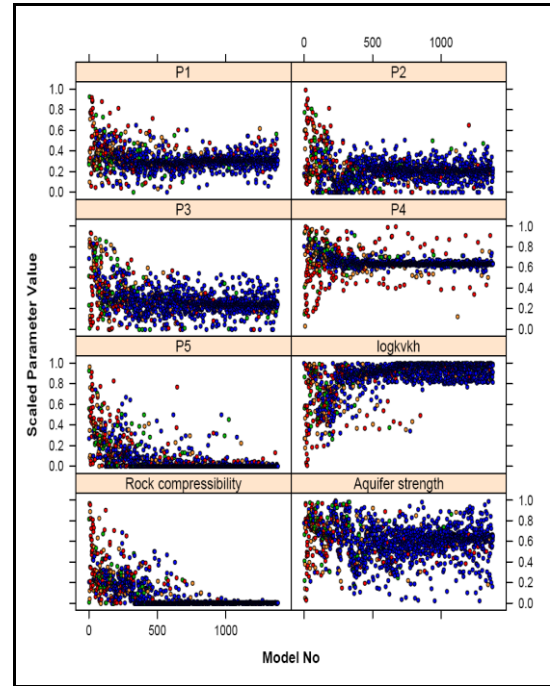


(d) Sampling history of PSO-T2D

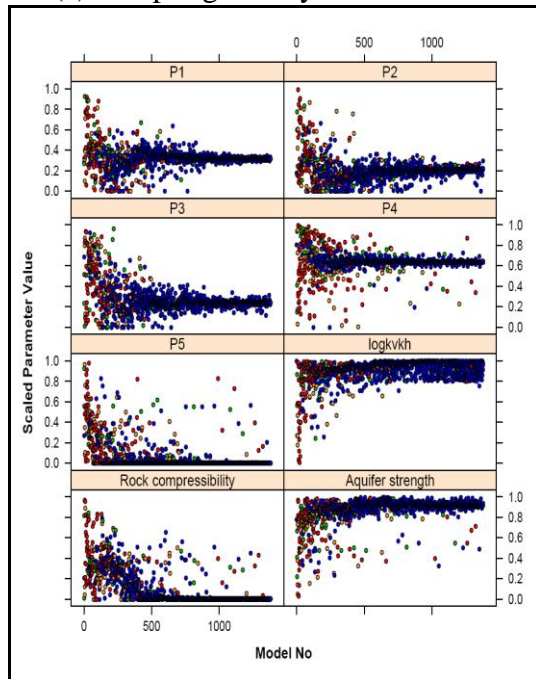
Figure 4.12: Damping strategy sampling history for each inertial weight choice



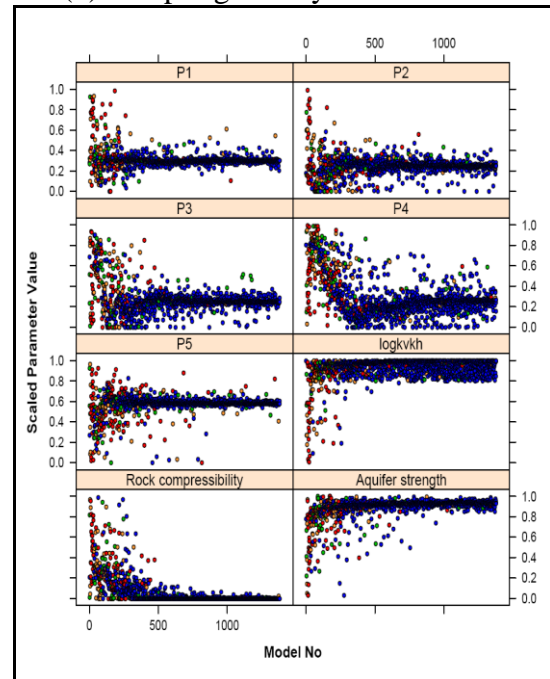
(a) Sampling history of PSO-LDS



(b) Sampling history of PSO-LIS



(c) Sampling history of PSO-T1S



(d) Sampling history of PSO-T2S

Figure 4.13: Absorbing strategy sampling history for each inertial weight choice

For a meaningful comparison and to test the reliability for each corresponding variant we performed 10 runs. Each run started from identical sets of 30 points with a new random seed. The mean generational minimum misfit evolution, along with the standard deviation around each point is shown in plots (a), (b), (c), and (d) of Figures 4.14, 4.15, 4.16, and 4.17. Similarly, the mean generational global misfit per generation, along with the standard deviation around each point is shown in plots (e), (f), (g), and (h) of Figures 4.14, 4.15, 4.16, and 4.17. For instance, both static PSO-T1B and dynamic

PSO–LDB variants reach the same misfit values, however, we can see on average the static PSO–T1B reduces the misfit in each generation more quickly than PSO–LDB. The two set of plots (the left and middle corresponding to mean generational minimum and the mean generational minimum so far) in the previous figures are close as efficiency plot measures for testing convergence speed. However, due to the stochastic nature of the algorithm they may differ in some other examples (see Section 4.3.2).

Plots of (i), (j), (k), and (l) in Figures 4.14, 4.15, 4.16, and 4.17 show the corresponding calculated diversity of the swarm per generation along with one standard deviation around each point for the 10 performed runs starting from the same 10 identical points calculated using Eq. (4.5). We can see that the diversity is reducing during the optimisation process which is to be expected. For example, we can observe that the degree of dispersion of particles in the dynamic PSO–LDB, in Figure 4.15(i), is more than in the PSO–T1B variant, in Figure 4.15(k). This could also be supported by Figure 4.11(a) and (c) in which we use sampling history plots to show the performance during the course of the optimisation. The parameter values for the good history matched models can be seen by looking at the range of the blue points. Both variants appear to concentrate on sampling for similar zones, although dynamic PSO–LDB is able to maintain population diversity more while PSO–T1B improves the sampling (shown in the blue points) as the optimisation progresses. Table 4.3 summarises these efficiency plot measures for each of the strategies per each of the inertial weight choices. There is a noticeable trend for diversity which is that the diversity decreases in the direction: Trelea Set Type I', Trelea Set Type I'', linear increase, linear decrease, $T1 < T2 < LI < LD$. A less solid tendency is observed for convergence of global best. If we compare the two measures, diversity and convergence of global best, per strategy there is no specific trend observed.

Table 4.3: Measures summary – 10 Seeds

Measure Strategy	Convergence of global best	Diversity
Random	T2R>LDR>LIR>T1R	T1R<T2R<LIR<LDR
Reflecting	LIB>T2B>T1B>LDB	T1B<T2B<LIB<LDB
Damping	LDD>LID>T2D>T1D	T1D<T2D<LID<LDD
Absorbing	T1S>T2S>LDS>LIS	T1S<T2S<LIS<LDS

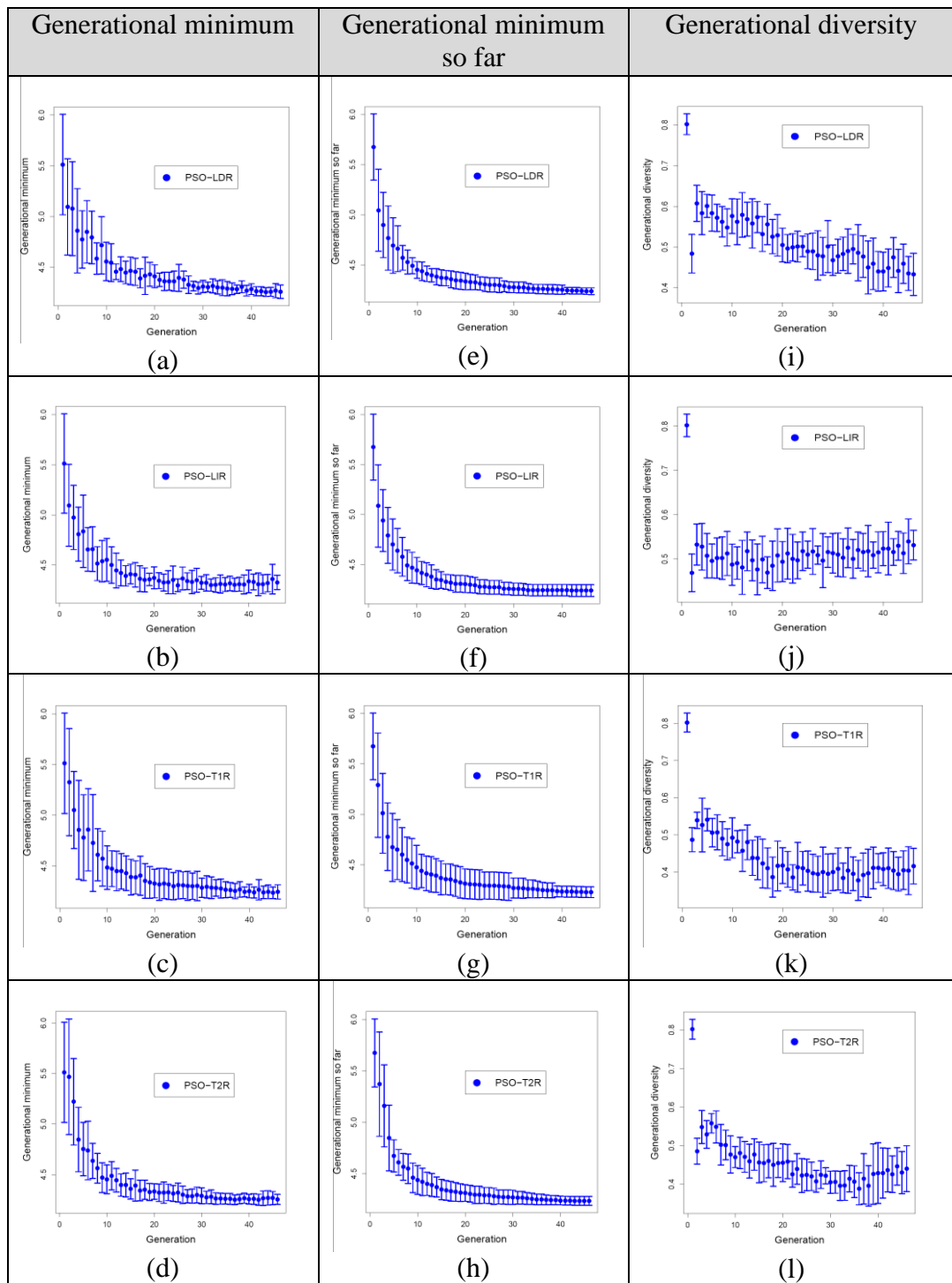


Figure 4.14: Random boundary strategy measures

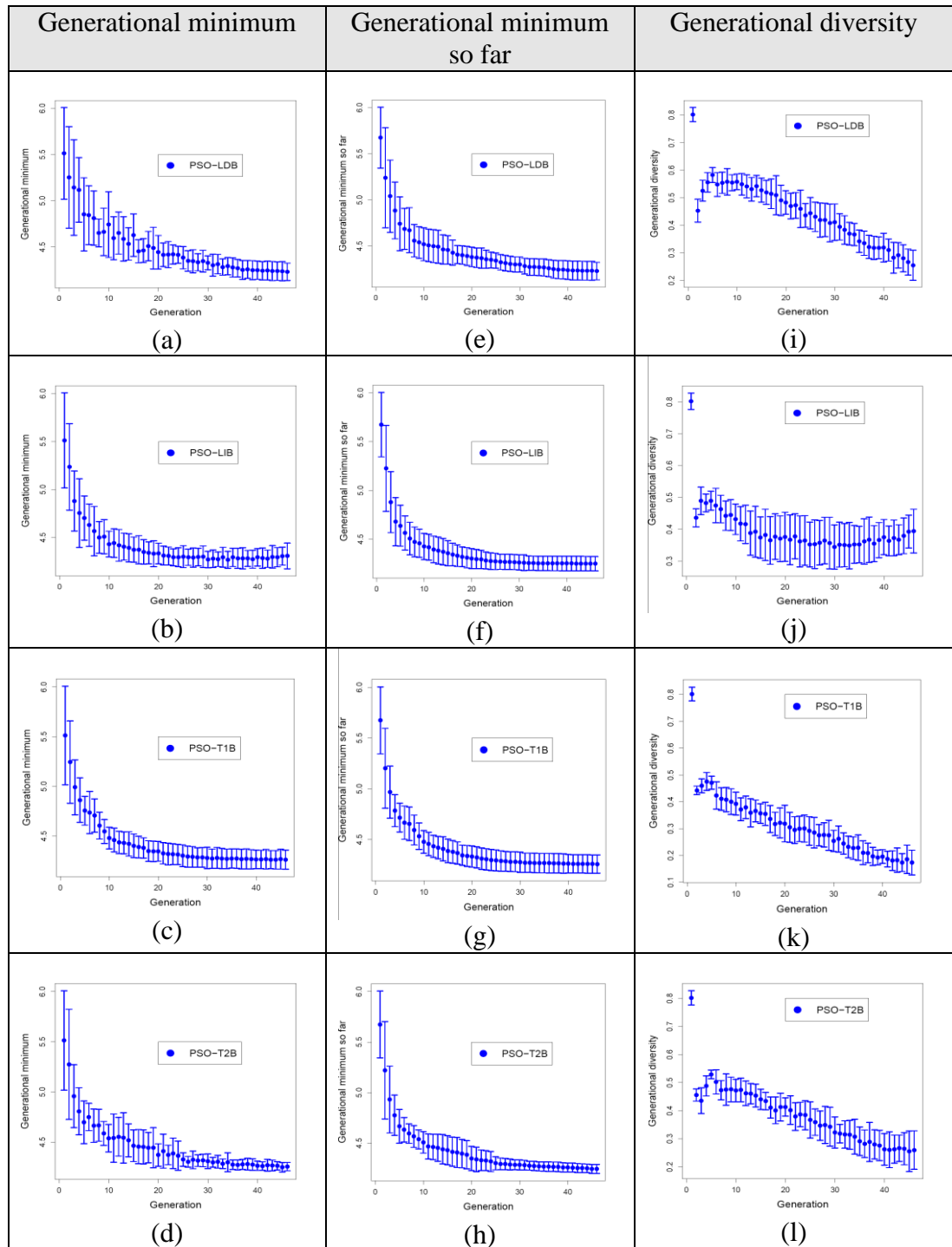


Figure 4.15: Reflecting boundary strategy measures

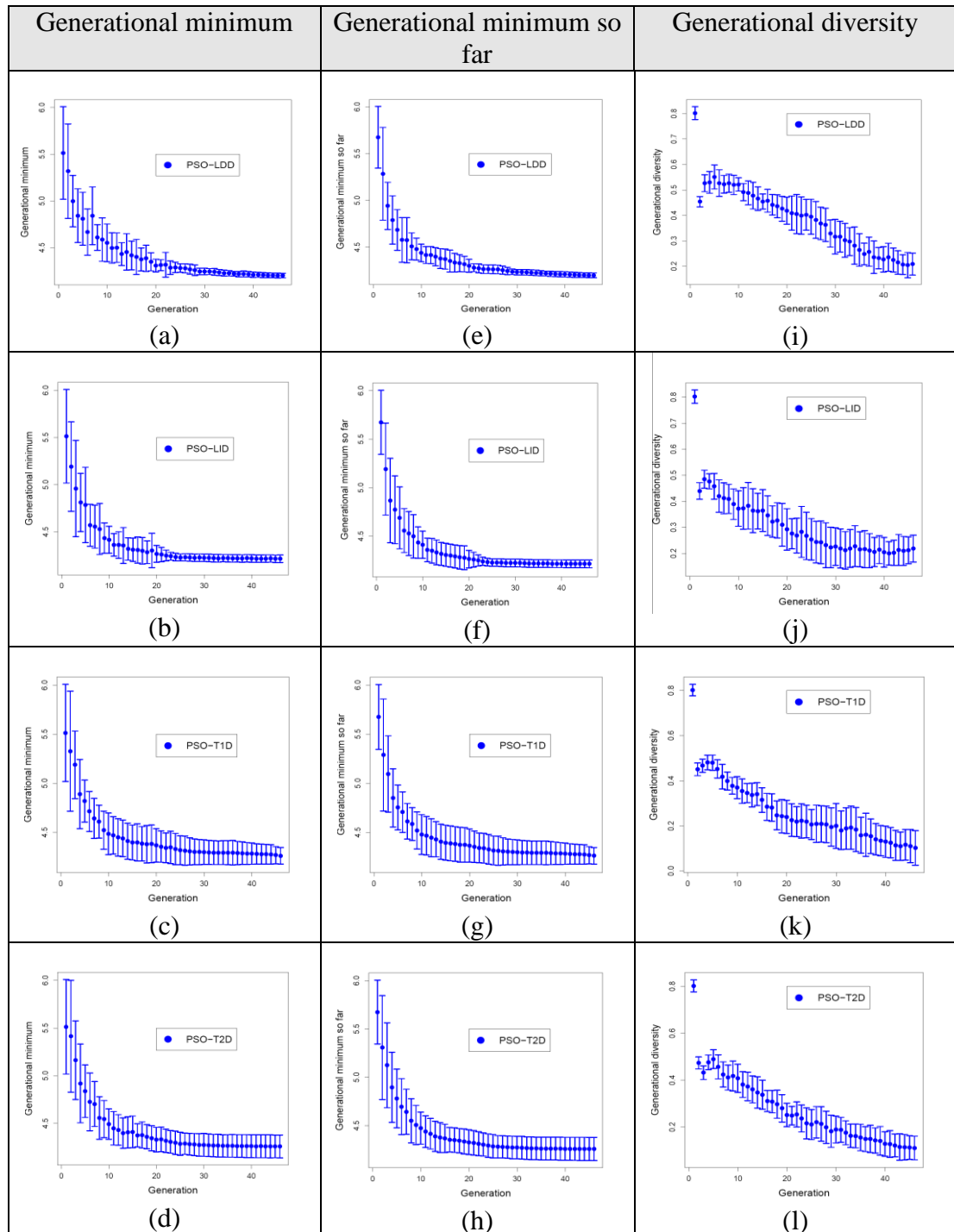


Figure 4.16: Damping boundary strategy measures

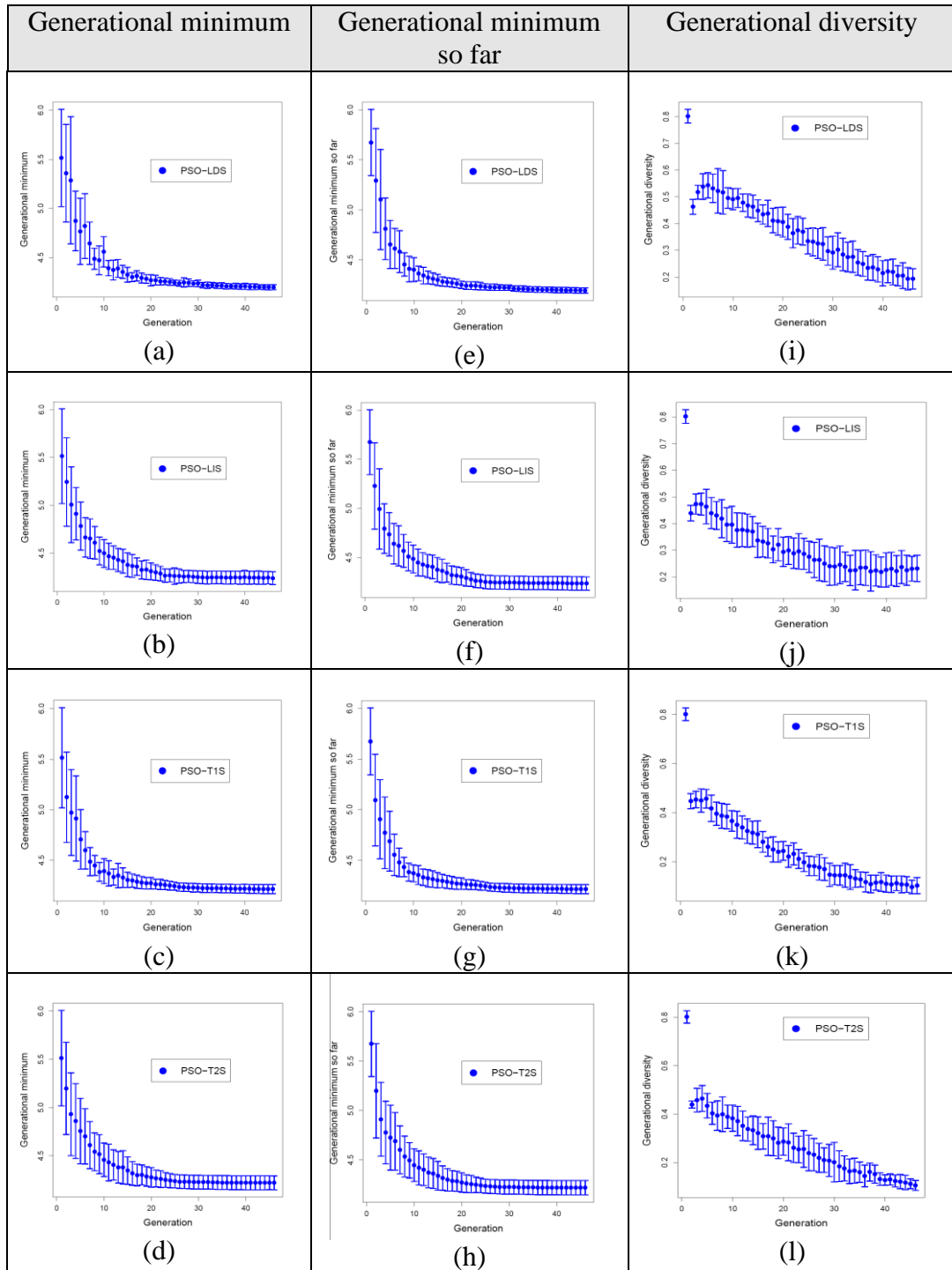


Figure 4.17: Absorbing boundary strategy measures

4.3.1.4.2 Handling Boundary Strategies for Particle Swarm Optimisation

Here we comment on the results of generating multiple history matched models with the four PSO boundary strategies (refer to Table 4.1 for acronyms). The comparison is based on the four inertial weight choices.

It is notable that the damping and absorbing strategies reduce the global best misfit quickly; however when no use is made of the old position and velocity, particularly

when the particle moves outside the boundary as in the random PSO–LDR strategy, the misfit values do not decline as fast as others during the search procedure. The reflecting strategy seems to have similar process to the random one. The previous plots in Figures 4.14, 4.15, 4.16, and 4.17 show the mean generational minimum misfit, the mean global best, diversity per generation evolution plus and minus one standard deviation around each point for the 10 achieved runs as shown in Table 4.4 which summarises the performance measures. In the comparison between the measures per inertial weight choice, a rough trend can be picked between diversity and convergence speed. Absorbing or damping strategies have faster convergence than random or reflecting strategies while the random or reflecting solutions have more diversity than absorbing or damping.

Table 4.4: Measures summary – 10 Seeds

Measure	Convergence	Diversity
IW Choice		
Linear Decrease	LDS>LDD>LDR>LDB	LDS<LDD<LDB<LDR
Linear Increase	LID>LIS>LIB>LIR	LID<LIS<LIB<LIR
Type I'	T1S>T1D>T1B>T1R	T1S<T1D<T1B<T1R
Type I''	T2S>T2D>T2R>T2B	T2D<T2S<T2B<T2R

4.3.1.5 *Uncertainty Assessment and Predictions*

In the previous section the history matching optimisation performances for the PSO variants were compared in terms of misfit reduction and diversity of models obtained. In this section we show how that influenced our uncertainty predictions using the ensemble of models obtained. As PSO was developed as an optimisation tool, we have to extend the algorithm to quantify uncertainty in reservoir modelling. We choose to extend the algorithm using the same concepts as the Neighbourhood Algorithm (NA), by running the NAB resampler code which computes the posterior probability. The NAB resampler (Sambridge, 1999b) is employed for the posterior uncertainty analysis purpose of the ensemble of models generated in the search stage by PSO variants, rather than making inferences from the single, best–fitting model with the lowest misfit value obtained. NAB utilises this ensemble to approximate the values of various Bayesian integrals with the standard Gibbs sampler. The misfit surface is assumed to be constant over each Voronoi cell surrounding a particle which is a property that is used by the NAB resampler. As a result, no forward modelling is carried out. The posterior

probability distribution (PPD) is approximated everywhere in parameter space using the Neighbourhood approximation of the PPD from the input ensemble. Inertial weight uncertainty evaluation is demonstrated next, followed by similar comparison for boundary strategies used.

4.3.1.5.1 *Inertial Weight Choices*

The forecast misfit calculated for the best history matched models with the same objective function definition indicated earlier in Eq. (4.6) including the corresponding number of observations used in the forecast and their associated history misfits for the PSO variants starting from a single seed are shown in Table 4.5.

Table 4.5: Best history and forecast misfit values of PSO variants

PSO Variant	Best History Misfit	Forecast Misfit
PSO-LDR	4.28	8.80
PSO-LIR	4.20	7.73
PSO-T1R	4.17	7.14
PSO-T2R	4.31	7.87
PSO-LDB	4.23	7.61
PSO-LIB	4.22	7.32
PSO-T1B	4.21	7.61
PSO-T2B	4.30	7.75
PSO-LDD	4.21	7.33
PSO-LID	4.24	7.75
PSO-T1D	4.21	7.01
PSO-T2D	4.17	7.32
PSO-LDS	4.21	8.04
PSO-LIS	4.21	7.32
PSO-T1S	4.21	7.10
PSO-T2S	4.16	7.14

Figure 4.18, 4.19, 4.20, and 4.21 show the Bayesian credible intervals ($p10-p50-p90$) for oil rate after history matching to the first 181 days of production. All variants capture the observed measurements for the history matching period and predictions. However, the uncertainty bounds are however wider in the dynamic inertial weight than the static ones. We prefer to have realistic reliable wider ranges of uncertainty in oil industry history matching applications. The reason is that in real-life case studies the predictions tend to underestimate the uncertainty (Valjak, 2008). The exploitation of search in the PSO-T1B to obtain models with low misfit values for instance, although it required fewer reservoir simulations, shown in Figure 4.15(c), resulted in local approximation of the posterior by NAB leading to narrower ranges of uncertainty than

the case with the PSO–LDB, in Figure 4.15(a). The predictions are affected by the individual contributions of the models. This could clearly be seen in the right plots (e), (f), (g), and (h) of Figure 4.22 in which the relative uncertainty is shown increasing over time at each simulated timestep. The corresponding diversity plots for these single runs are shown in (a), (b), (c), and (d) of same figures. In dynamic PSO–LDB the relative uncertainty is larger than in the case of PSO–T1B, in Figure 4.22(f). The diversity of models plot shows a larger dispersion of models in the dynamic PSO–LDB than in the static PSO–T1B in Figure 4.22(b), which is reflected in the relative uncertainty figure. Table 4.6 summarises the relative uncertainty with their corresponding global best and diversity evolutions for the single seed.

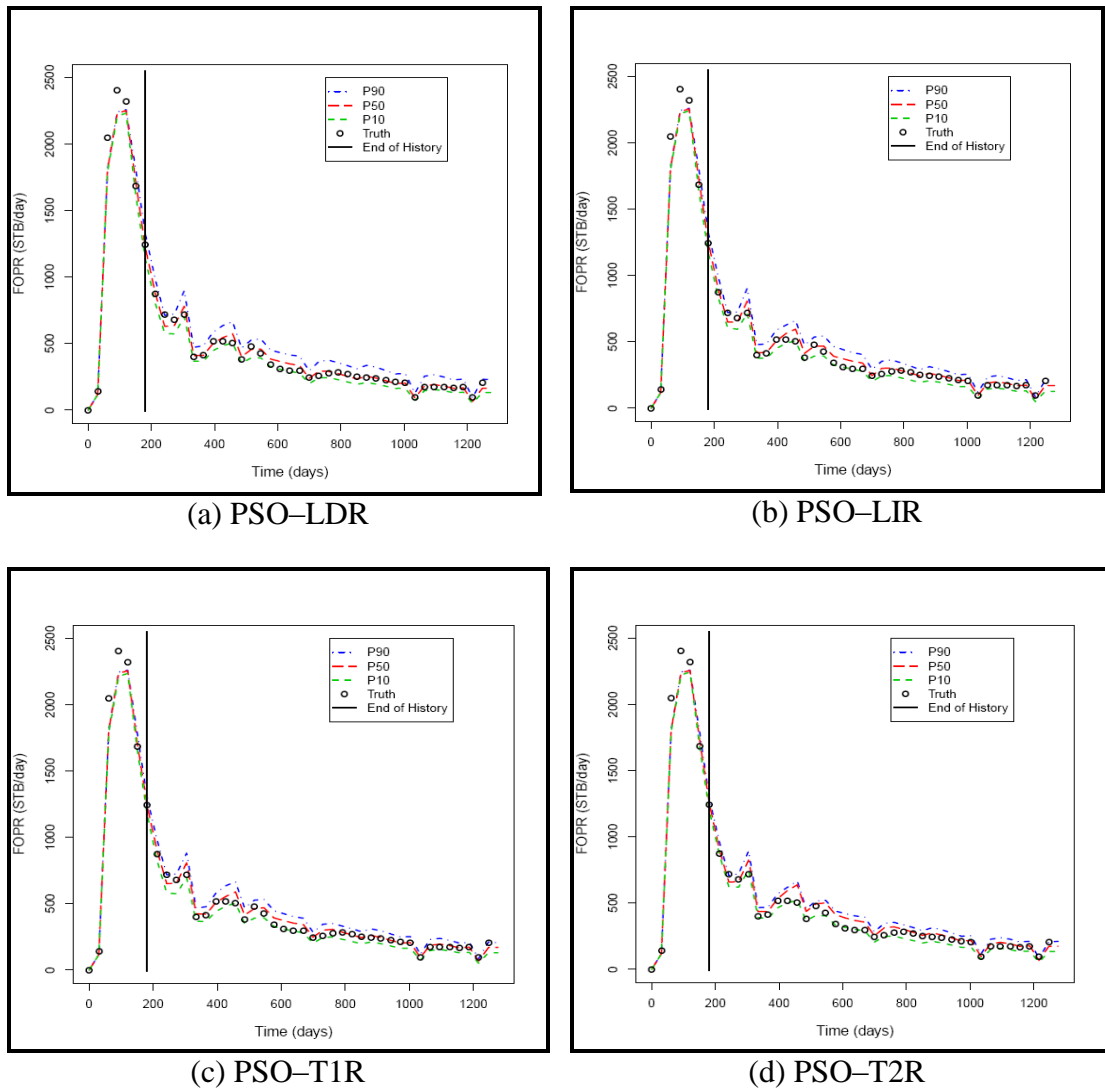
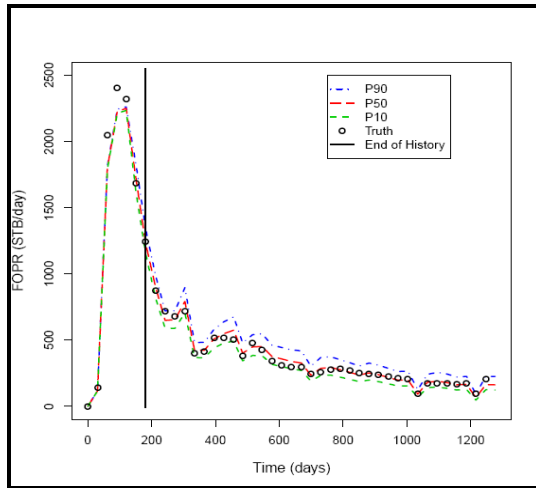
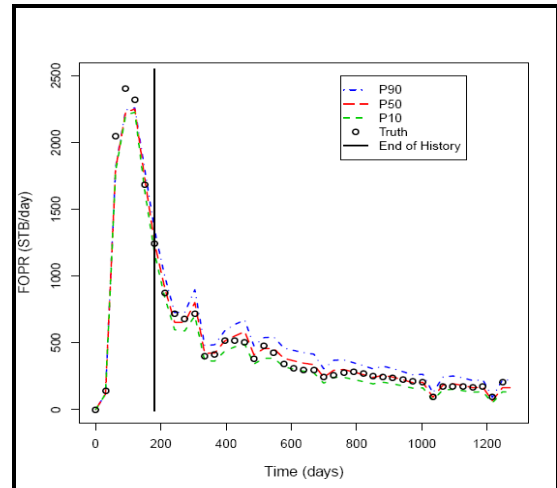


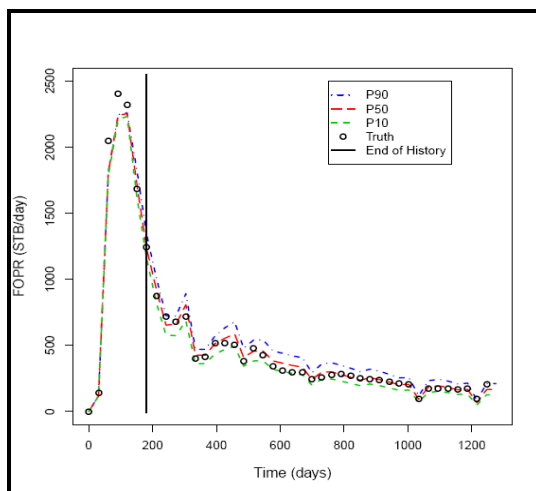
Figure 4.18: Bayesian credible intervals generated with random strategy variants



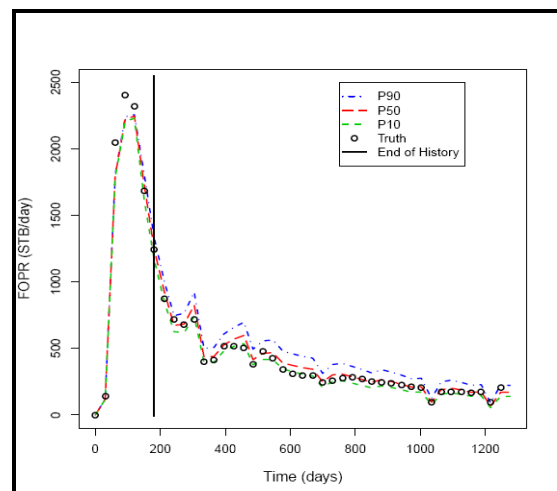
(a) PSO-LDB



(b) PSO-LIB

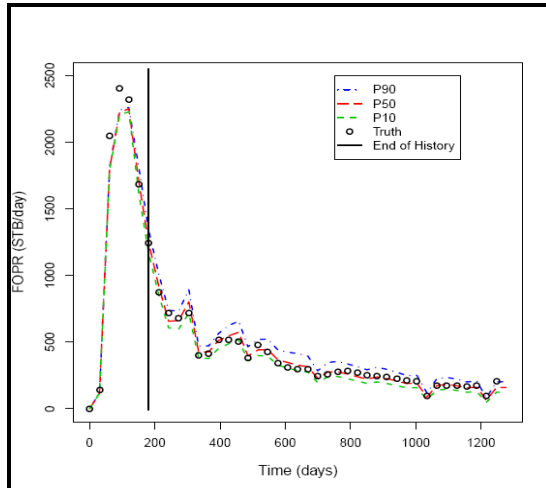


(c) PSO-T1B

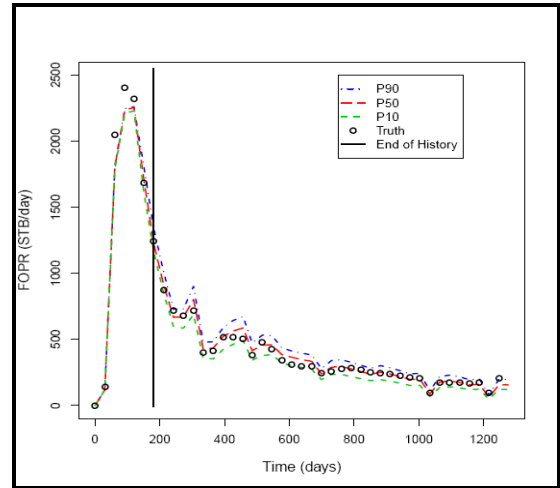


(d) PSO-T2B

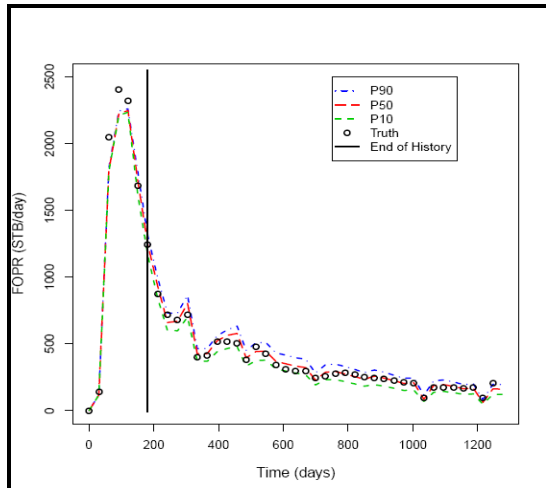
Figure 4.19: Bayesian credible intervals generated with reflecting strategy variants



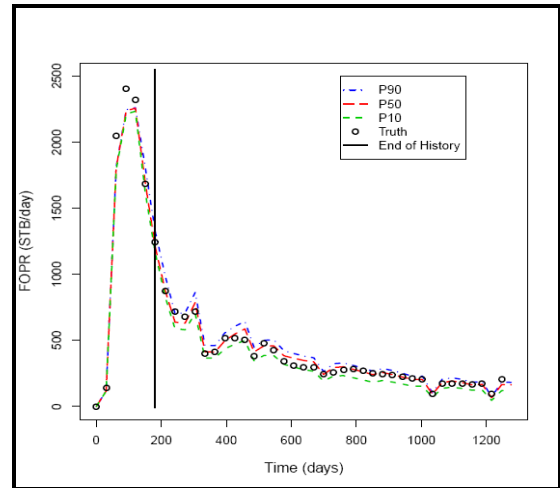
(a) PSO-LDD



(b) PSO-LID

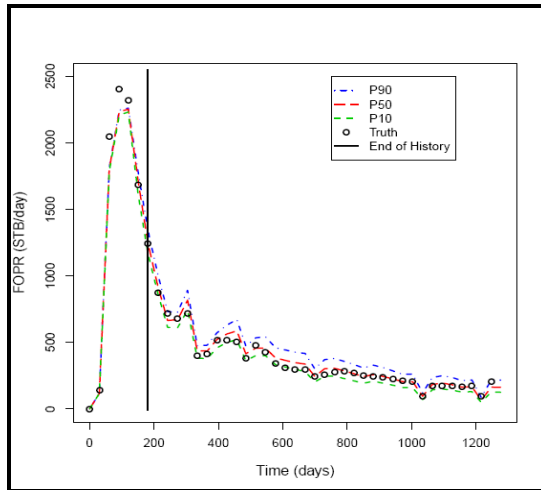


(c) PSO-T1D

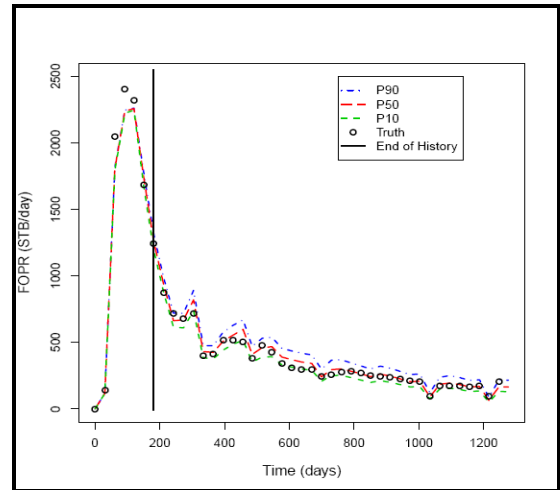


(d) PSO-T2D

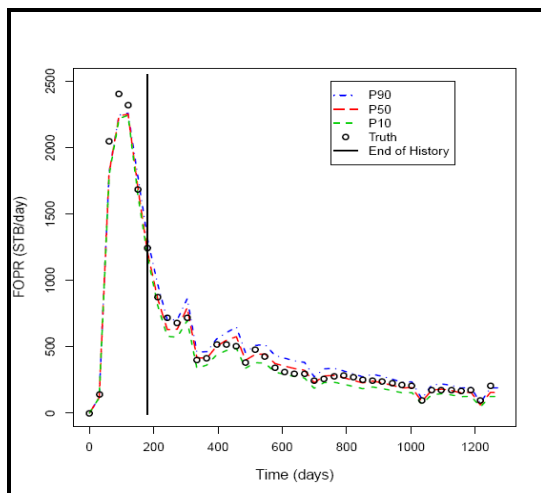
Figure 4.20: Bayesian credible intervals generated with damping strategy variants



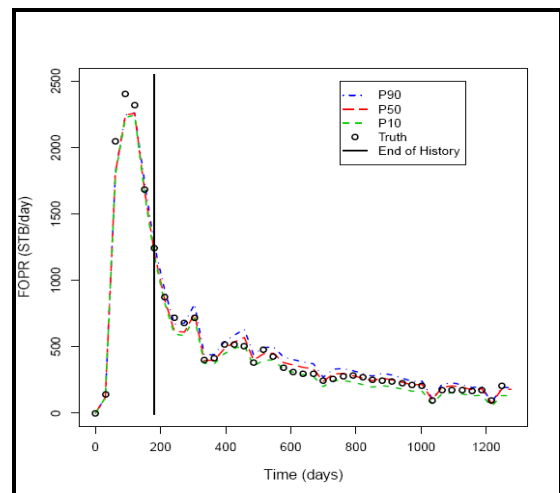
(a) PSO-LDS



(b) PSO-LIS

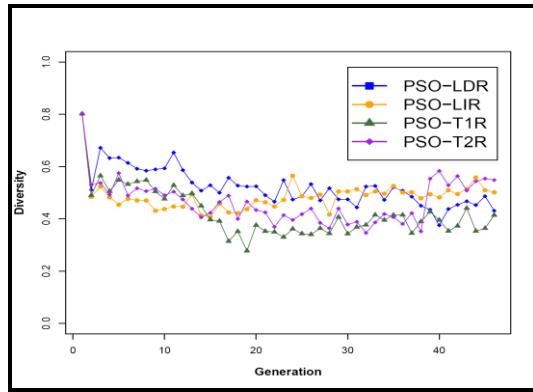


(c) PSO-T1S

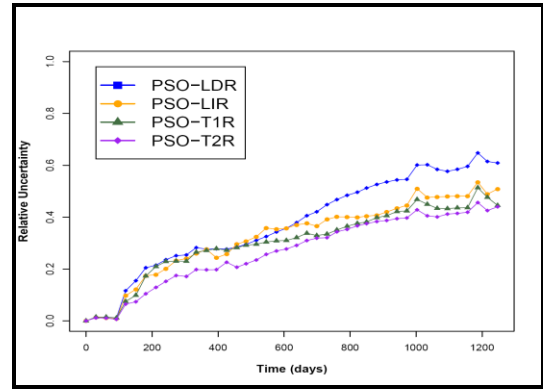


(d) PSO-T2S

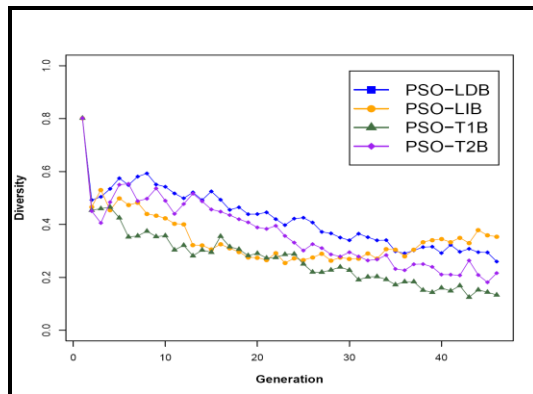
Figure 4.21: Bayesian credible intervals generated with absorbing strategy variants



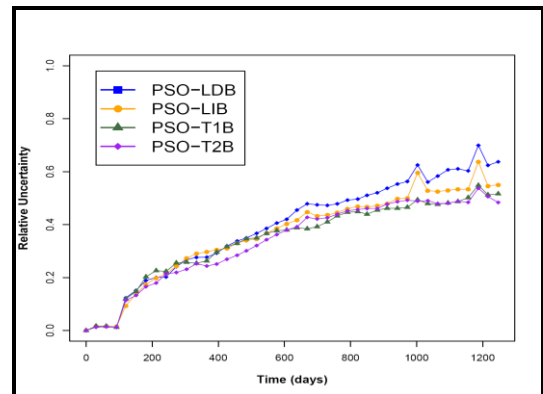
(a) Random – Diversity



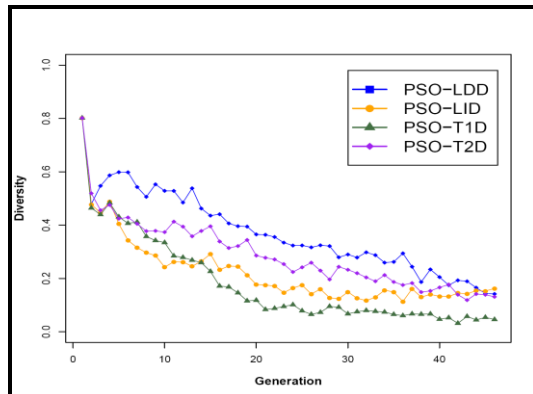
(e) Random – Relative uncertainty



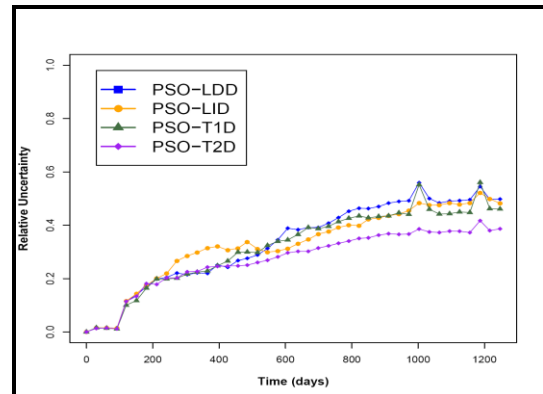
(b) Reflecting – Diversity



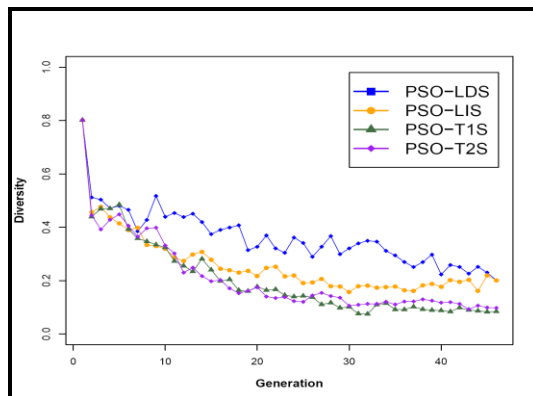
(f) Reflecting – Relative uncertainty



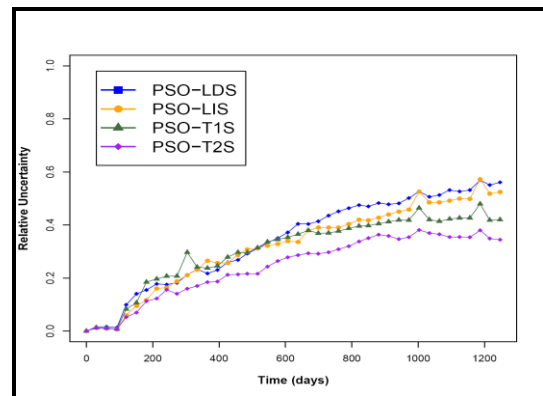
(c) Damping – Diversity



(g) Damping – Relative uncertainty



(d) Absorbing – Diversity



(h) Absorbing – Relative uncertainty

Figure 4.22: Diversity per generation and relative uncertainty

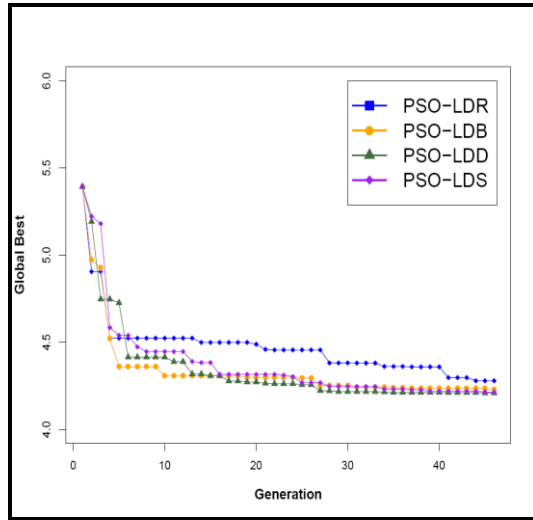
Table 4.6: Measures summary – 1 Seed

Measure Strategy	Convergence	Diversity	Uncertainty estimation
Random	T1R>LIR>T2R>LDR	T1R<T2R<LIR<LDR	T2R<T1R<LIR<LDR
Reflecting	T1B>LIB>LDB>T2B	T1B<T2B<LIB<LDB	T2B<T1B<LIB<LDB
Damping	T2D>T1D>LDD>LID	T1D<LID<T2D<LDD	T1D<T2D<LID<LDD
Absorbing	T2S>T1S>LIS>LDS	T1S<T2S<LIS<LDS	T2S<T1S<LIS<LDS

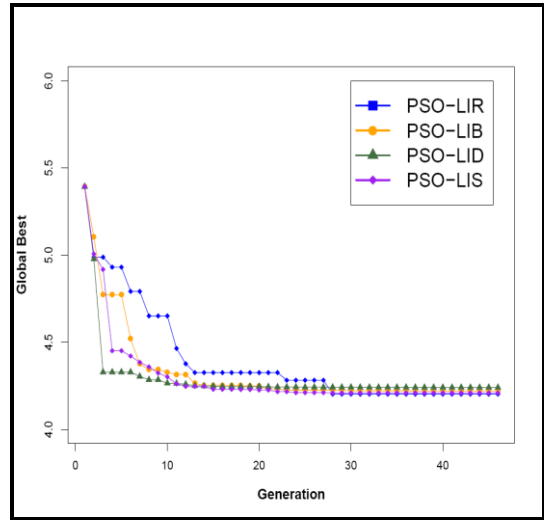
4.3.1.5.2 Handling Boundary Strategies for Particle Swarm

Similar comparison plots are carried out with the four strategies. As indicated earlier, the comparison here shows a trend between diversity and convergence speed. The Bayesian credible intervals ($p10$ – $p50$ – $p90$) for oil rate have captured the observed measurements in the history matching period and predictions for all the PSO strategies as shown previously. Figure 4.24 shows the relative uncertainty (ϵ) depicted in Figure 4.24 with the corresponding diversity plots. If we look at the linear decrease choice for example with all the strategies we see that the refinement behaviour observed in the absorbing and damping strategies, PSO–LDS, and PSO–LDD, that entailed less reservoir simulation models to converge to low misfit models, have the effect that the uncertainty envelopes for PSO–LDS and PSO–LDD are relatively narrower than in PSO–LDB and PSO–LDR with the first element in each group having the largest range over than the second element in the same group, as shown in Figure 4.24(a) and (e). The PSO–LDB and PSO–LDR have more diverse models which encompass the good fitting models with low misfit values as well as the poor–fitting ones. As a result, the NAB inferences reflected the two different sampling behaviours in the two concentrative (observed in PSO–LDS and PSO–LDD) and explorative sampling (observed in PSO–LDR and PSO–LDB). For the same example (LD) the relative uncertainty is higher for reflecting PSO–LDB than a random PSO–LDR even though the random PSO–LDR has a noticeably larger diversity of particles compared to the reflecting PSO–LDB. The reason for this is that the good fitting models for the random PSO–LDR are relatively as good as that for the reflecting PSO–LDB, as noted earlier in the sampling history in Figures 4.10(a) and 4.11(a) and the mean generational minimum in Figures 4.14(a) and 4.15(a). Subsequently, the uncertainty estimations are reflected by two factors: the quality of the sampling and the dispersion of the low misfit reservoir models obtained in parameter space. The strategies PSO–LDS and PSO–LDD have similar performances as they both move near the boundaries of promising regions. However, we note that the

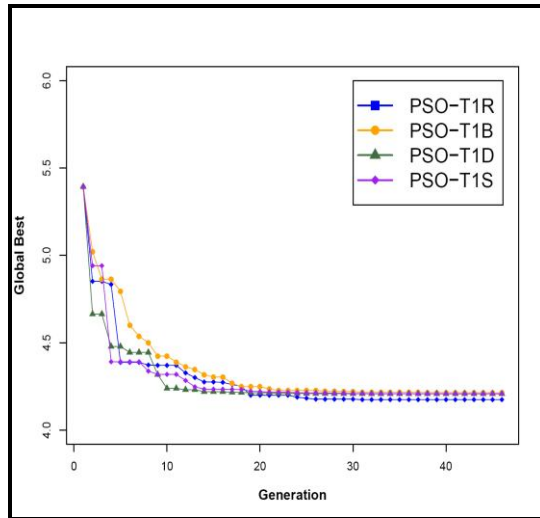
performance of the PSO-LDS has the wider ranges of uncertainty than the PSO-LDD with fast convergence, as shown in Figures 4.24(e) and 4.23(a). Table 4.7 summarises the relative uncertainty with their corresponding global best and diversity evolutions for the different tuning of inertial weight choices for a single seed.



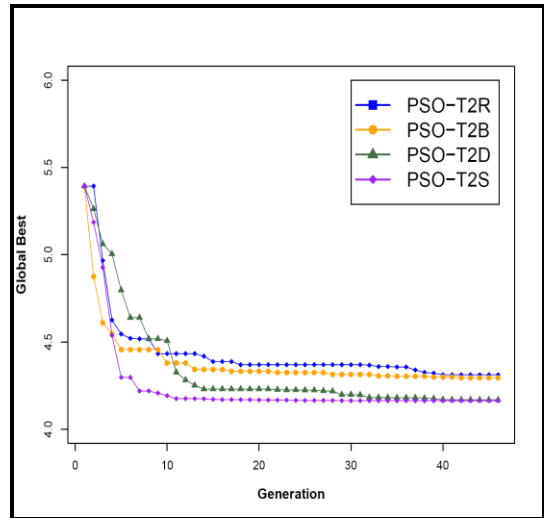
(a) LD – Misfit reduction



(b) LI – Misfit reduction

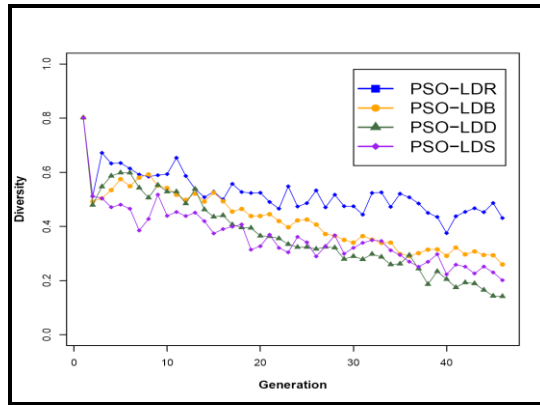


(c) T1 – Misfit reduction

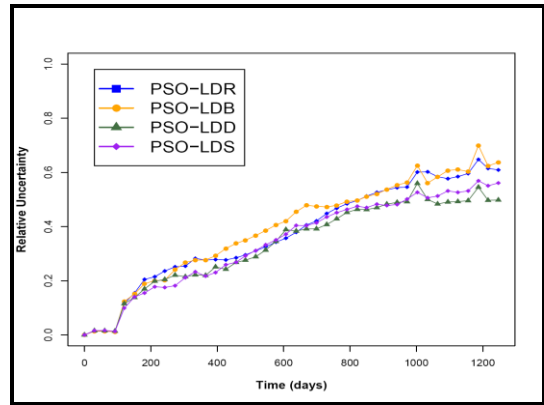


(d) T2 – Misfit reduction

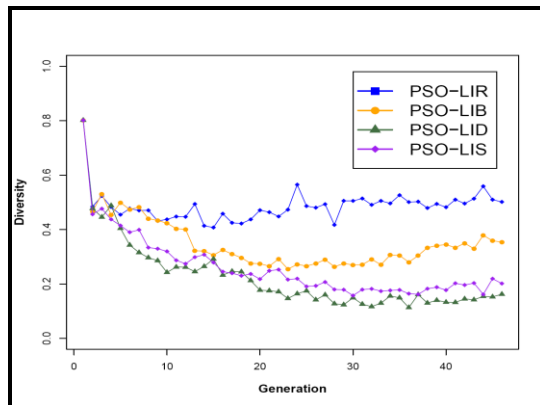
Figure 4.23: Misfit reduction for variants



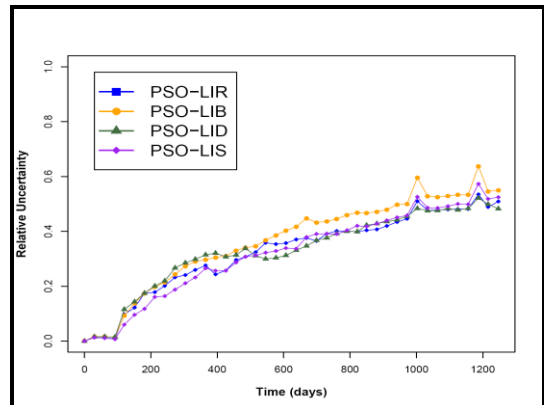
(a) LD – Diversity



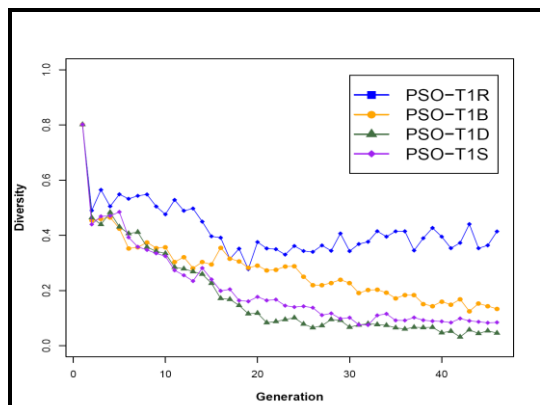
(e) LD – Relative uncertainty



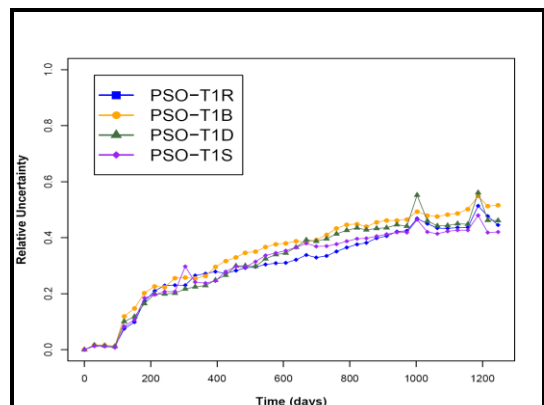
(b) LI – Diversity



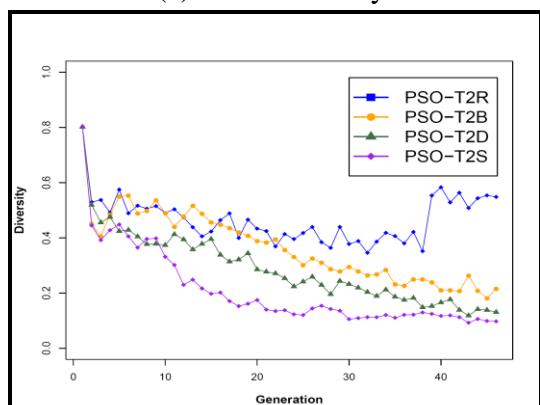
(f) LI – Relative uncertainty



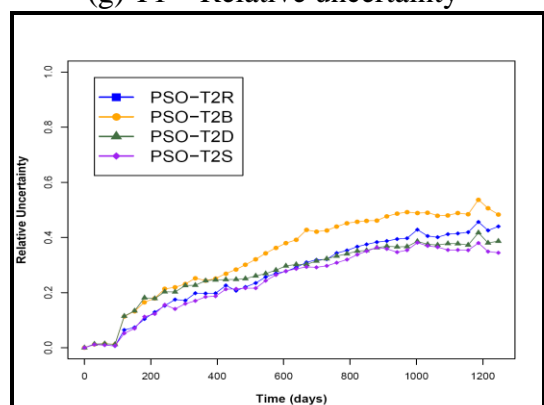
(c) T1 – Diversity



(g) T1 – Relative uncertainty



(d) T2 – Diversity



(h) T2 – Relative uncertainty

Figure 4.24: Diversity per generation and relative uncertainty

Table 4.7: Measures summary – 1 Seed

Measure IW Choice	Convergence	Diversity	Uncertainty estimation
Linear Decrease	LDD>LDS>LDB>LDR	LDD<LDS<LDB<LDR	LDD<LDS<LDR<LDB
Linear Increase	LIS>LID>LIB>LIR	LID<LIS<LIB<LIR	LID<LIR<LIS<LIB
Type I'	T1R>T1D>T1S>T1B	T1D<T1S<T1B<T1R	T1R<T1S<T1D<T1B
Type I''	T2S>T2D>T2B>T2R	T2S<T2D<T2B<T2R	T2B<T2R<T2D<T2S

Finally, the total oil recovery prediction after 1247 days for Teal South reservoir is shown in Figure 4.25. All PSO variants captured the measured observed value which is shown as a horizontal dashed line.

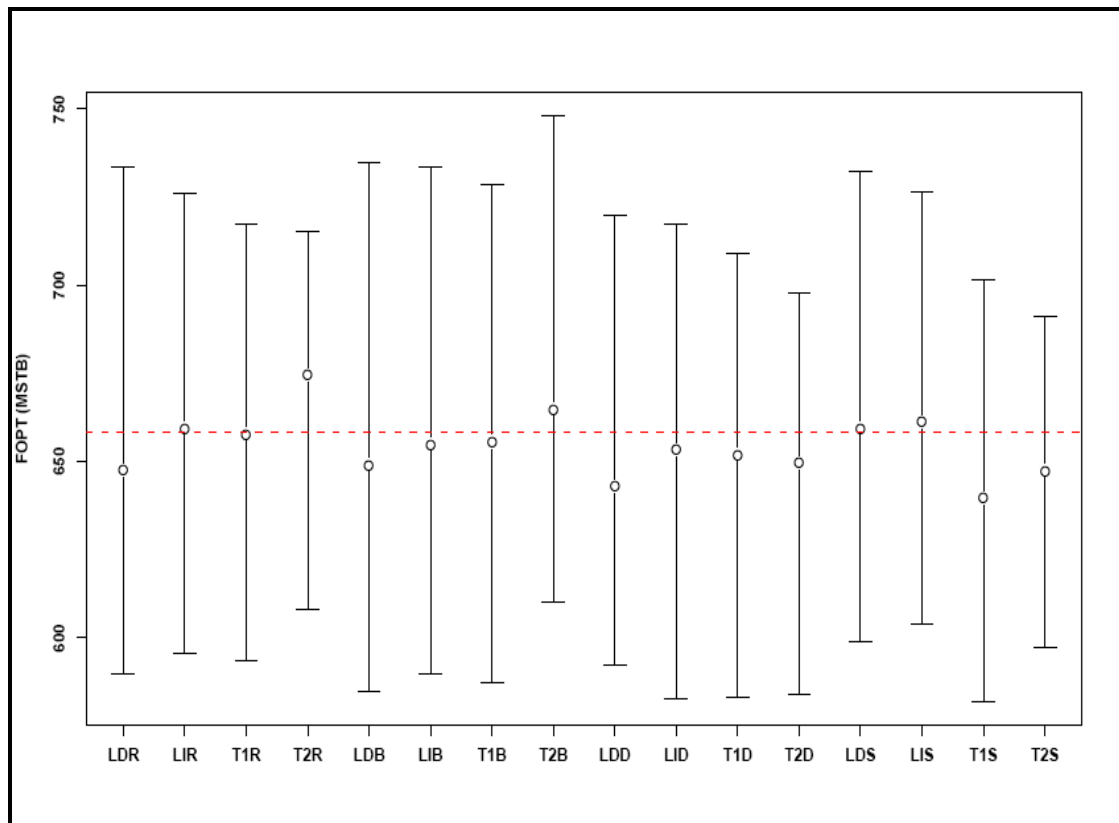


Figure 4.25: Bayesian credible intervals of total recovery prediction for PSO variants

4.3.2 Imperial College Fault Model

The Imperial College Fault model (Tavassoli et al., 2004) is a simple synthetic test case which is known to be a very challenging test example (Busby and Feraille, 2008; Bush and Carter, 1996; Carter and Ballester, 2004; Carter et al., 2006; Erbas and Christie, 2007b; Tavassoli et al., 2004) where standard methods for history matching often fail and can be very unreliable. The IC Fault model response surface is a complicated

surface with many local minima. Examples of methods which have been tested include genetic algorithms (Ballester and Carter, 2006; Carter and Ballester, 2004), estimation of distribution algorithms (Petrovska and Carter, 2006, 2010) and support vector machines (Demyanov, Pozdnoukhov, Christie, and Kanevski, 2010).

The geological model consists of six layers of alternating high and low permeability sands. The three good quality layers have identical properties, and the three poor quality layers have a different set of identical properties. The thickness of the layers gradually decreases from top (thickness 12.5 ft) to bottom (7.5 ft) with a total reservoir thickness of 60 ft. The width of the model is 1000 ft, with a simple vertical fault in the middle, which affects the connectivity between layers. A water injector well is located at the left-hand edge, and a producer well at the right-hand edge. Both wells are completed in all layers, and operated at fixed bottom hole pressures to control the production from the reservoir. There are no oil-water or gas-oil contacts. The simulation model is 100×12 grid blocks. Each geological layer is divided into two equal simulation layers as shown in Figure 4.26. A more detailed description of the model is provided in Carter (2004), and Tavassoli et al. (2004).

4.3.2.1 IC Fault Model Uncertain Parameters

The simplified reservoir of the IC Fault model is characterised by three uncertain input parameters corresponding to the fault throw thickness (h), and the values of good (k_{high}) and poor permeability (k_{low}). The porosities of the high quality sand are set to 0.30 and the poor quality sand to 0.15. The uniform prior ranges and the truth case values used are shown in Table 4.8 following Christie et al. (2006), Erbas and Christie (2007b), and Tavassoli et al. (2004).

Table 4.8: Parameterisation for the IC Fault model and prior ranges

Parameter	Units	Prior range	Truth case	Truth case scaled to [0,1]
k_{high}	mD	[100,200]	131.6	0.1316
k_{low}	mD	[0,50]	1.3	0.0260
h	ft	[0,60]	10.4	0.1733

Figure 4.27 shows the true production rates for oil and water, as a function of time. The observed data consists of the first three years of monthly oil and water rates obtained from the truth case simulation. The next seven years are used as the forecast data to

measure the predictive quality of the history matches. The 2D saturation map of oil/water for the truth case simulation at the end of the history matching period of 3 years is shown in Figure 4.28.

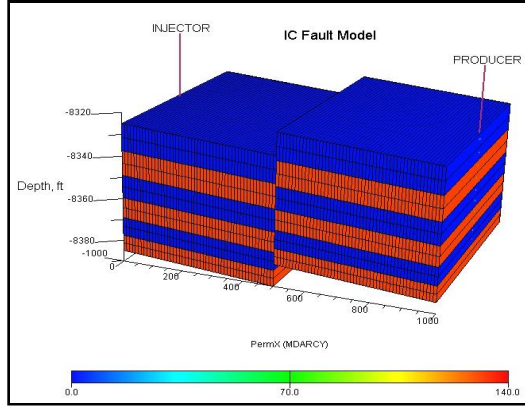


Figure 4.26: IC Fault Model (2010)
(Tavassoli et al., 2004)

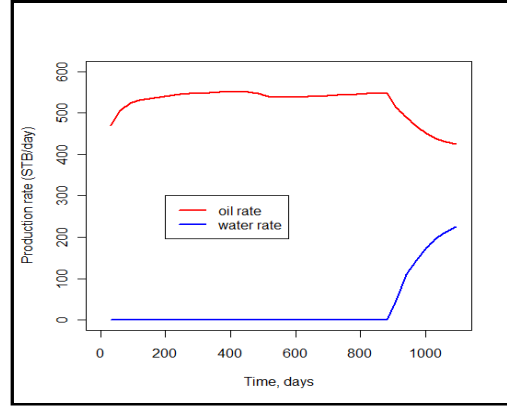


Figure 4.27: IC Fault model truth case
production data (oil and water rates)

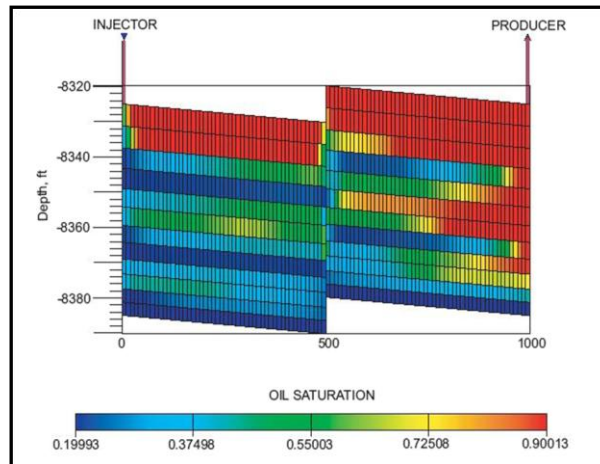


Figure 4.28: 2D saturation map for the IC Fault model truth case

4.3.2.2 Objective Function Definition

The misfit M – the negative log of the likelihood – is calculated using the least squares formula Eq. (4.7).

$$M = \frac{1}{T} \sum_{p \in \{o, w\}} \sum_{t=1}^T \frac{(q_p^{obs} - q_p^{sim})^2}{2\sigma_t^2} \quad \text{Eq. (4.7)}$$

where T is the number of observations equal to 36, q_p represents oil and water production rates, and obs and sim refer to observed (truth case) and simulated respectively. $\sigma = 0.03 \times (q_p^{obs} + w)$ is the standard deviation of the observed data (the

added random Gaussian noise) where $w = 10^{-6}$ is an offset that has an impact on the misfit surface smoothness (Ballester, 2005). The summation for the water rate starts from when the water is produced.

The challenge in sampling with search algorithms is due to the complexity in sampling the response surface of the model. This can be illustrated by the minimum located in a sharp region in the response surface in Figure 4.29 where 1D cross-sections were depicted for the three uncertain parameters by fixing two parameters to their true values (refer to Table 4.8) and changing the third each in turn. Along the throw parameter three regions with steep minima can be identified. The response surface is influenced by the water flow from injector to producer, the production rates and water breakthrough time. Faults affect flow in reservoir simulation, altering the connectivity of the sedimentological layers and displacements across the faults possibly connecting stratigraphically disconnected high permeability layers as well as juxtaposing high against low permeability units. These effects are conventionally incorporated in flow simulators using the fault transmissibility multipliers. Thus, changing the throw parameter values has an impact on the placement on low and high permeability geological layers leading to different fault transmissibility values. This presents a challenge to assisted history matching techniques in finding models that match the historical data well and to be able quantify the uncertainty in the forecasts.

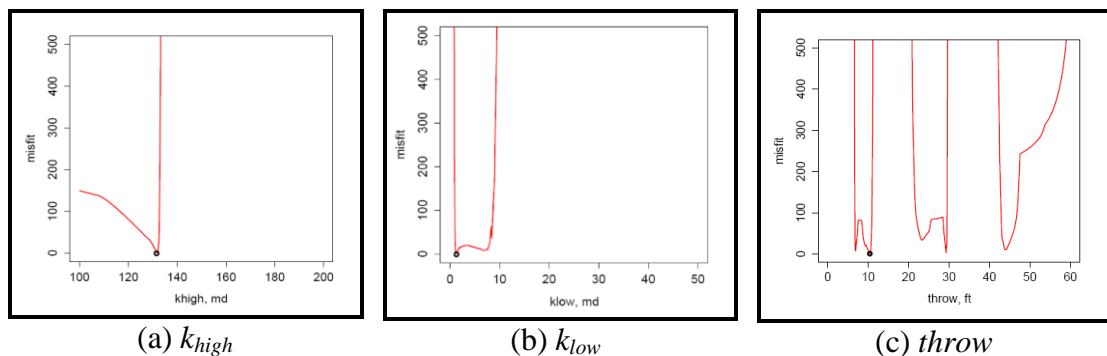


Figure 4.29: The truth value cross-section along each parameter

Because the IC Fault Model is simple and quick, it is possible to generate a large number of samples using Uniform Monte Carlo sampling to act as a benchmark result. Bush and Carter (1996) demonstrated in a previous work that it is difficult to obtain a history match using simple optimisation and thus generated a large number of realisations as shown in Tavassoli et al. (2004). Although this method does not

guarantee finding the global optimum it may get sufficiently close as well as identifying if local optima exist. A database of 159,661 uniform sampled models following the original study of Tavassoli et al. (2004) was regenerated. Figure 4.30 shows all models in the database with misfit $M \leq 25$. Note that this hard cutoff of $M \leq 25$ is only used for plotting purposes to illustrate the complex twisting, ribbon-like shape of the low misfit region in three dimensions. The database result shows a complex, twisting, ribbon-like structure of models that match history.

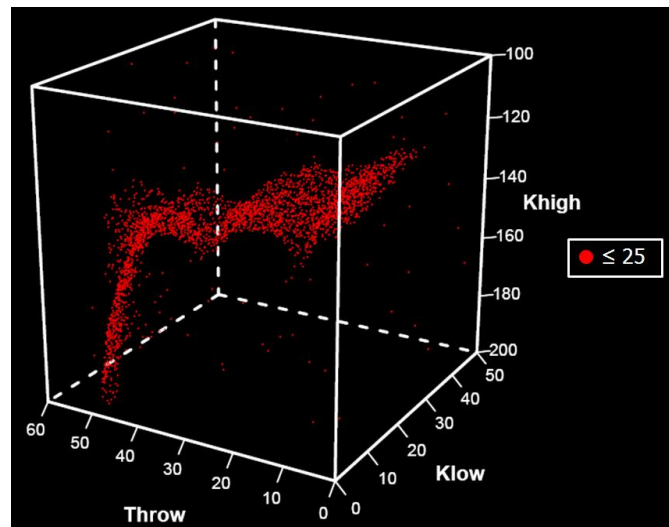


Figure 4.30: Database models with $M \leq 25$

4.3.2.3 IC Fault Model Setup Specifications

The previous section compared the performance of PSO variants on Teal South, a simple single well field in the Gulf of Mexico with 8 parameters. In the second example we will compare PSO variants on the IC Fault Model, which exhibits a complex misfit surface.

To compare on this example we used a set of 20 initial starting points obtained with Latin Hypercube Sampling (LHS) for its characteristics. Figure 4.31 shows an example of a set of initial models with a minimum misfit of 57. The LHS is briefly illustrated next. For comparison purposes PSO variants were set up to be as similar as possible by using the same initial points. This setup will be used within the thesis for testing different methods unless otherwise stated.

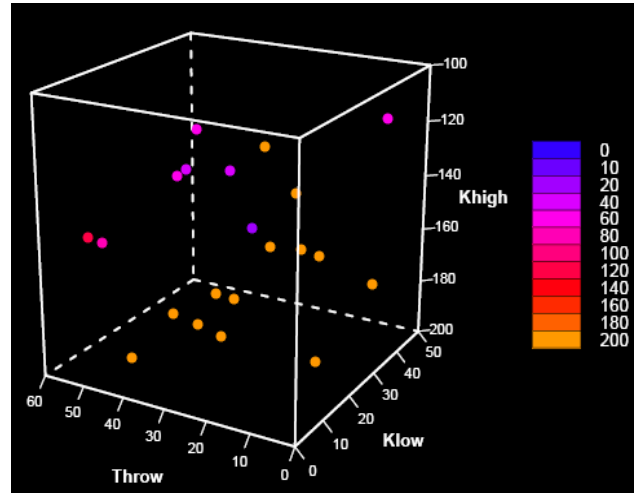


Figure 4.31: Initial population of 20 starting points obtained with Latin Hypercube Sampling in 3D parameter space

4.3.2.3.1 Latin Hypercube Sampling (LHS)

Latin Hypercube Sampling (LHS) was originally used for selecting parameter values in computer models for Monte Carlo simulation purposes (Iman and Conover, 1980; McKay et al., 1979). It has been applied in environmental studies and soil science for predictions in uncertainty assessment (Minasny and McBratney, 2002), as well as in geostatistics for simulation of Gaussian random fields (Pebesma and Heuvelink, 1999; Zhang and Pinder, 2003). LHS is a stratified-random sampling technique that performs sampling efficiently from parameter's distributions (Iman and Conover, 1980). The LHS can be looked at as a sampling technique that lies between simple random sampling, which entails no stratification, and stratified sampling, which stratifies on sample space (Wahanani et al., 2009). In contrast to simple random sampling, this technique guarantees a full coverage of each parameter's range since it stratifies each marginal distribution to maximum.

Considering d parameters x_1, x_2, \dots, x_d , LHS sampling samples m values from their distributions by dividing the cumulative distribution for each into N equiprobable intervals. From each interval a random value is drawn as in Figure 4.32. In the next step the N values acquired for each parameter are paired randomly with the other parameters. The LHS procedure is summarised in the following steps:

- The cumulative distribution of each parameter is divided into N equiprobable intervals.
- A value is selected randomly from each interval. Then the sampled cumulative probability for interval i , is as in Eq. (4.8) (Wyss and Jorgensen, 1998) where r_u refers to a uniformly distributed random number in the range $[0,1]$.

$$\text{Prob}_i = (1/N)r_u + (i-1)/N \quad \text{Eq. (4.8)}$$

- By using the inverse of the distribution function F^{-1} as in Eq. (4.9), the probability values sampled are transformed into the values x :

$$x = F^{-1}(\text{Prob}) \quad \text{Eq. (4.9)}$$

- For each parameter x , the N values acquired are paired in a random manner with the m values of the other parameters, in equally likely combinations.

The technique assumes that the parameters are independent. This may be considered as a limitation since in practical applications that may not be the case and the parameters may be correlated. Note that independent parameters tend to bias uncertainty. In addition, random pairing of correlated parameters could lead to impossible or infeasible combinations (Minasny and McBratney, 2006).

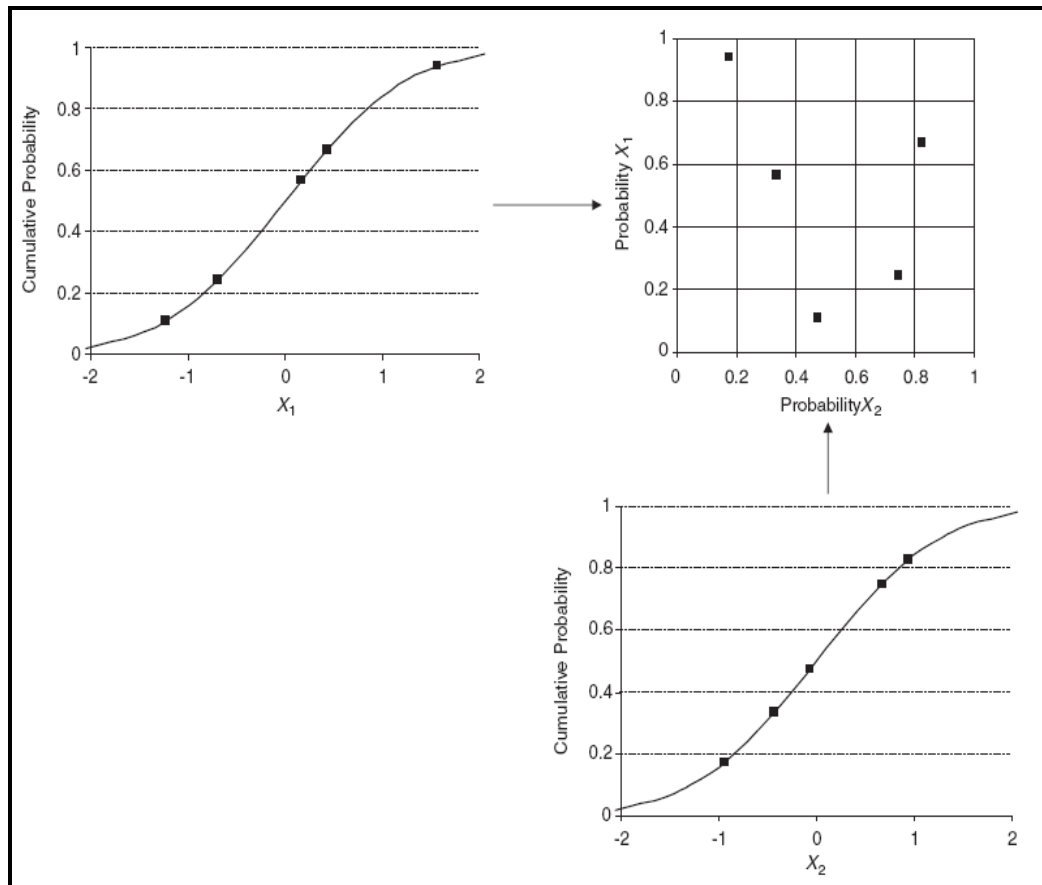
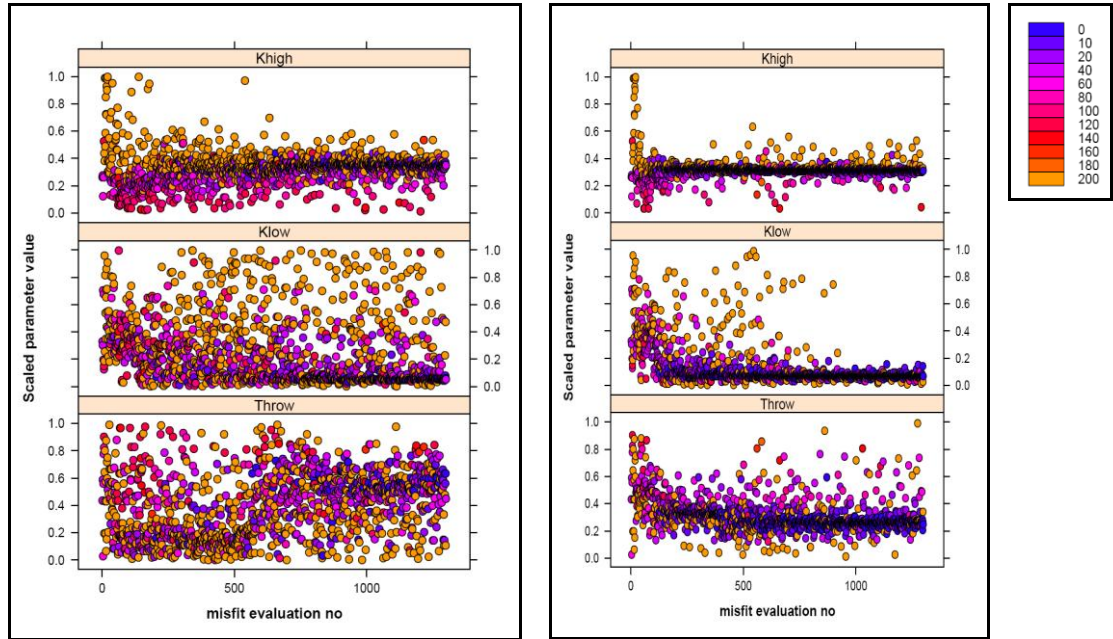


Figure 4.32: LHS for two parameters with normal distribution. For each parameter the cumulative probability is split into five equal strata, and a random sample is drawn at each strata. Five samples from each parameter are then paired randomly forming a Latin square (Source: Minasny and McBratney (2006))

4.3.2.4 *History Matching of IC Fault Model*

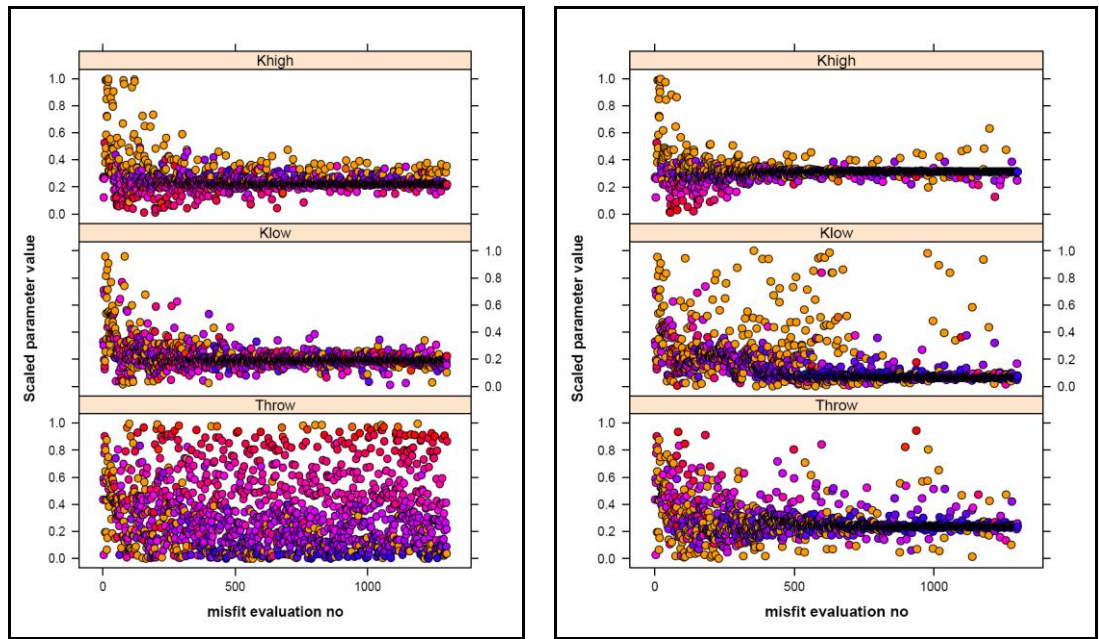
For all the variants we start from a fixed initial population comprising 20 models generated with LHS in parameter space. The optimisation is done for 65 generations where each generation consists of these 20 models. The total number of reservoir model simulations is 1300 for all the tests performed. Each test was repeated for 10 seeds. We compare the four inertial weight choices per each boundary strategy. The cognition and social components were chosen to be 2 for dynamic inertial weight choices. Sampling history plots to look at the performance during the course of the optimisation are shown in Figures 4.33, 4.34, 4.35, and 4.36 per strategy. Each plot has 3 panes, showing the evolution of the parameter sampling as we sample in time. The horizontal axis is the model number and the vertical axis shows the scaled values of each parameter multiplier between 0 and 1. The points are colour coded according to the misfit where levels of blue points indicate the low misfit models. The orange points have a misfit of 200 or above, and include many models that do not match at all well. As sampling advances in time, the concentration in promising regions with low misfit models is observed. Due to the complexity of the response surface the colour varies for each variant. We aim to find good models particularly exploring different areas along the throw axis to capture as many multiple minima as possible. Using all the strategies with linear decrease inertial weight in particular achieves this objective to some extent. This is also supported by the representation of samples in the 3D view depicted in Figures 4.37 and 4.38 where points in red are the models which have misfits equal to 25 or below. Note that the predictions are influenced by two factors the good quality models as well as the poor ones (shown in pink) and the density in sampling in the parameter space. Finding models in the same cluster will probably have the same prediction profile. Thus, we need to find as many distinct minima as possible to have different divergent descriptions. In particle swarm increasing the size of the swarm when the response surface is very complex can help even more in finding these models. Our test here aims to come up with initial guesses and ready options for achieving certain objectives and tasks, such as faster convergence or more diverse models, or a balance between the two, by investigating how different variants influence the sampling. In real-field applications where we have sharp minima we need to consider the linear decrease inertial weight as starting test.

For a meaningful comparison and to test the reliability of each corresponding variant we performed 10 runs. Each run started from an identical set of 10 points with a new random seed. Since our good models have misfits of below 1 or above 200 we used the box plots to show the performance results for the plots. We present box plots of generational minimum per generation and global best per generation for the 10 runs similar to previous Teal South model for all the PSO variants as illustrated in Figures 4.39, 4.40, 4.41, and 4.42. This is done by calculating the generational minimum in 10 runs starting from the same starting samples. In the box plot the line in the box refers to the median, the lower and upper bound points to the 25% and 75% quartiles respectively, and the “circle” sign refers to the outliers. The box plot for generational minimum misfit evolution, along with the standard deviation around each point is shown in plots (a), (b), (c), and (d) of Figures 4.39, 4.40, 4.41, and 4.42. Similarly, the mean generational global misfit per generation, along with the standard deviation around each point is shown in plots (e), (f), (g), and (h) of Figures 4.39, 4.40, 4.41, and 4.42. For instance, we can see on average the static PSO–T1S reduces the median misfit in each generation more quickly as shown in Figure 4.42(c) and (g) than PSO–LIS in Figure 4.42(b) and (f) .



(a) Sampling history of PSO-LDR

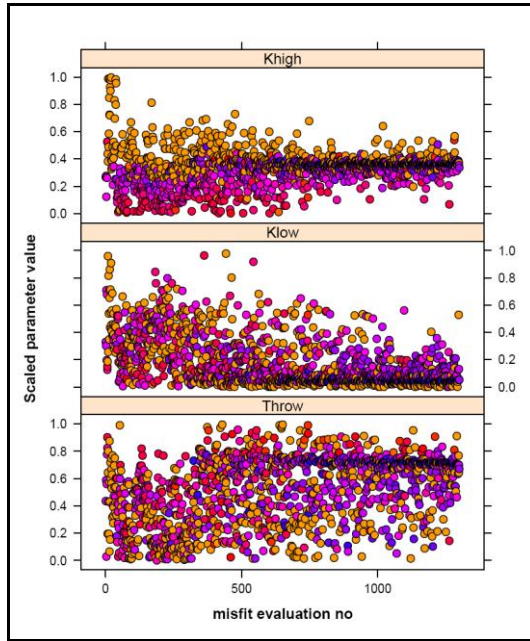
(b) Sampling history of PSO-LIR



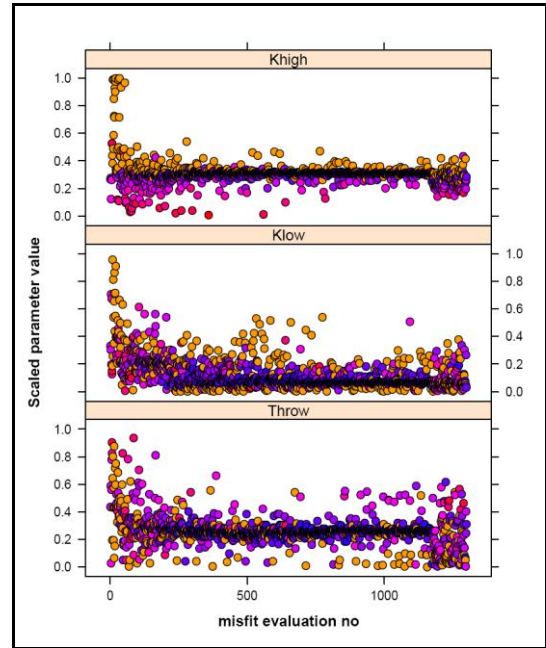
(c) Sampling history of PSO-T1R

(d) Sampling history of PSO-T2R

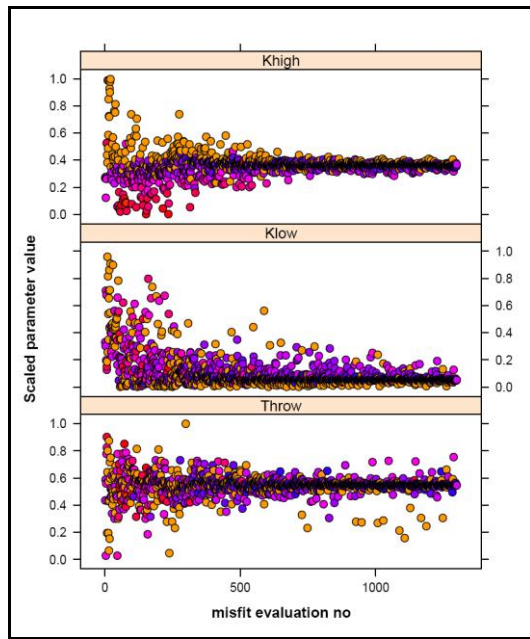
Figure 4.33: Random strategy sampling history for each inertial weight choice



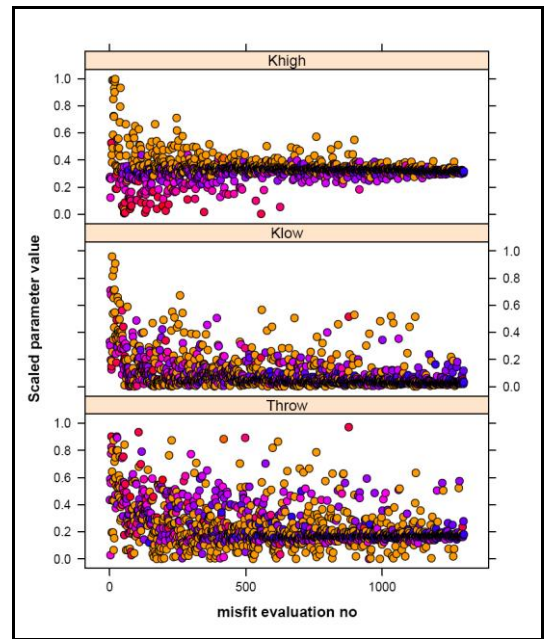
(a) Sampling history of PSO-LDB



(b) Sampling history of PSO-LIB

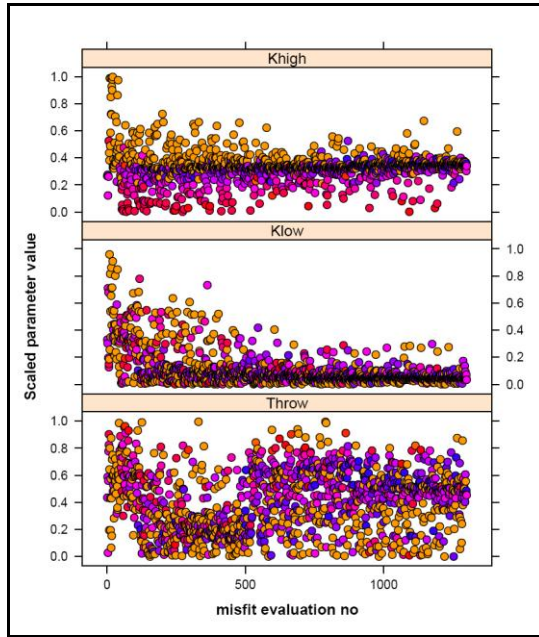


(c) Sampling history of PSO-T1B

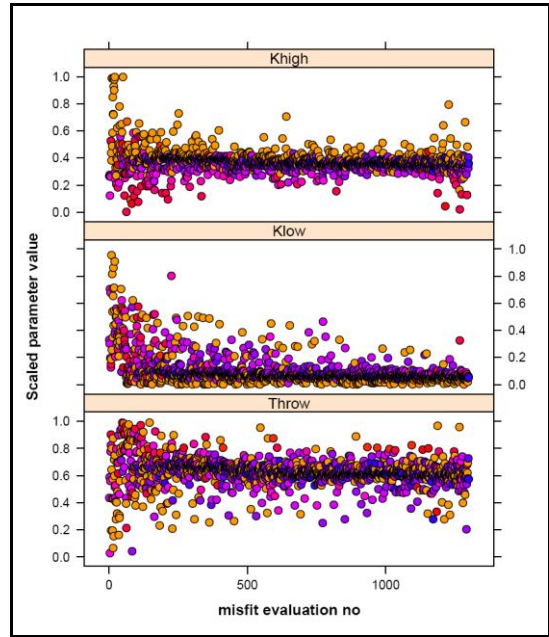


(d) Sampling history of PSO-T2B

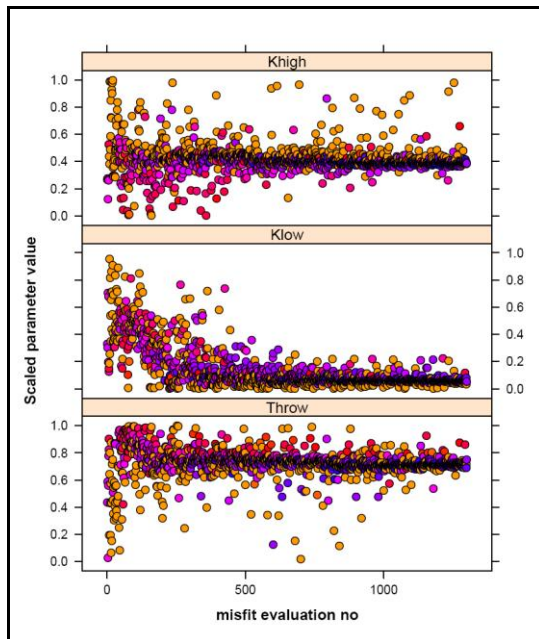
Figure 4.34: Reflecting strategy sampling history for each inertial weight choice



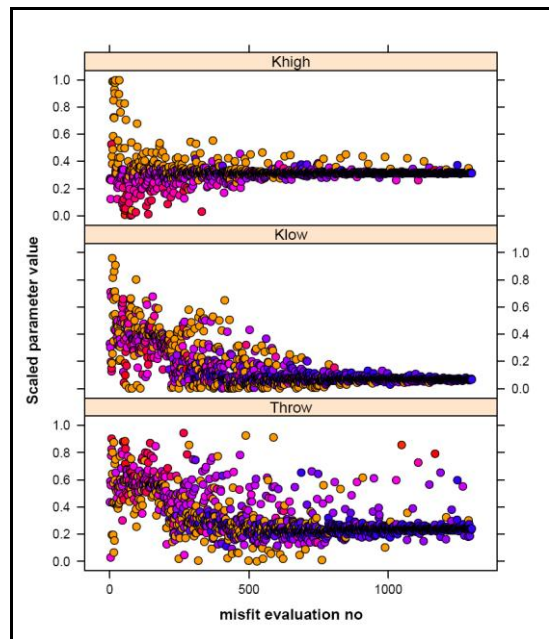
(a) Sampling history of PSO-LDD



(b) Sampling history of PSO-LID

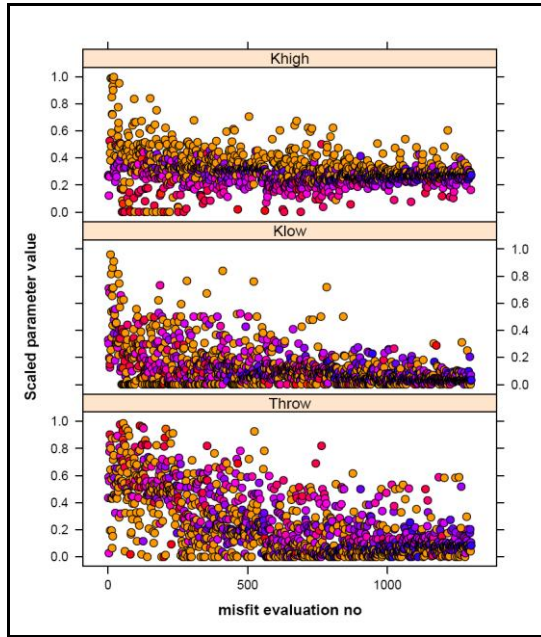


(c) Sampling history of PSO-T1D

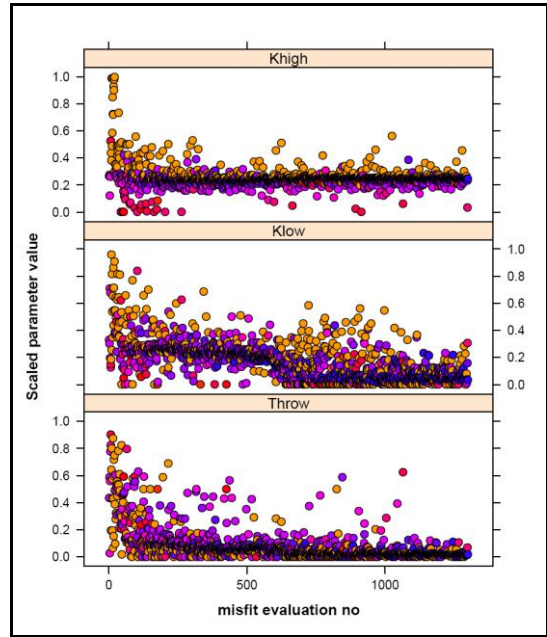


(d) Sampling history of PSO-T2D

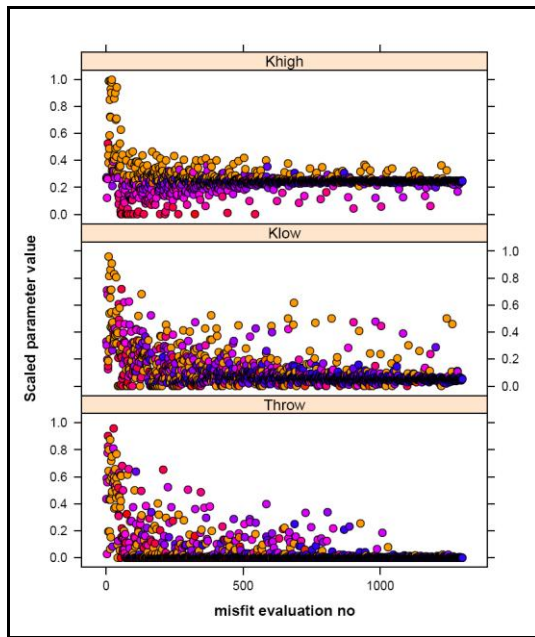
Figure 4.35: Damping strategy sampling history for each inertial weight choice



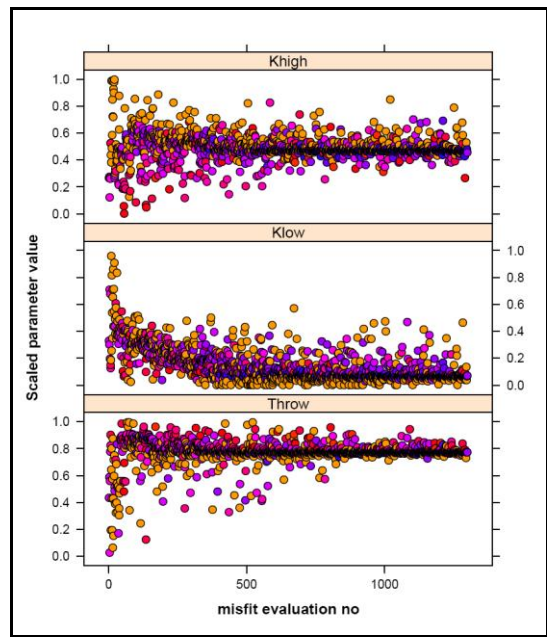
(a) Sampling history of PSO-LDS



(b) Sampling history of PSO-LIS

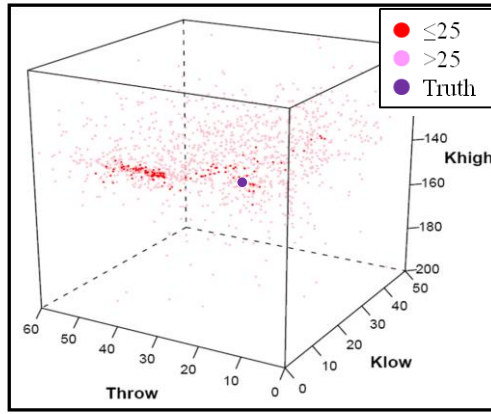


(c) Sampling history of PSO-T1S

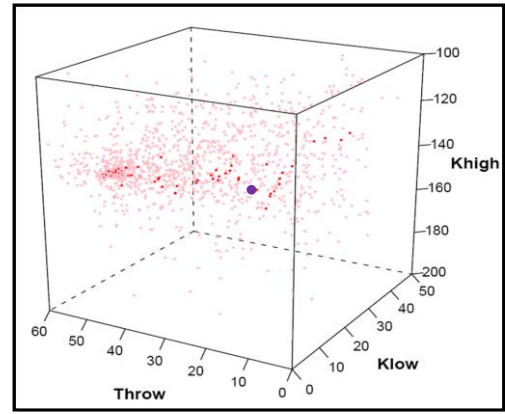


(d) Sampling history of PSO-T2S

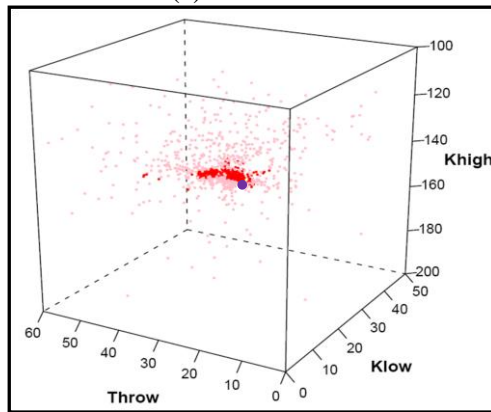
Figure 4.36: Absorbing strategy sampling history for each inertial weight choice



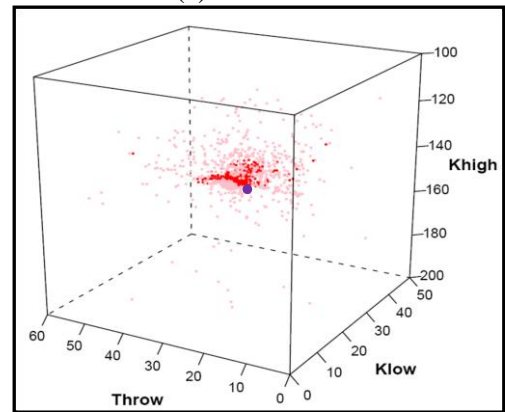
(a) PSO-LDR



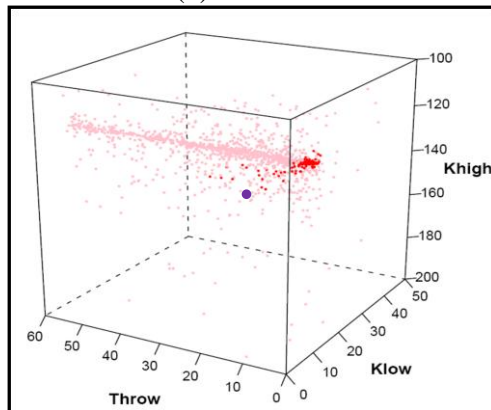
(e) PSO-LDB



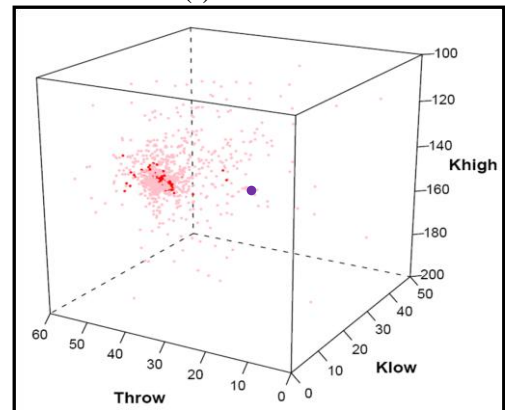
(b) PSO-LIR



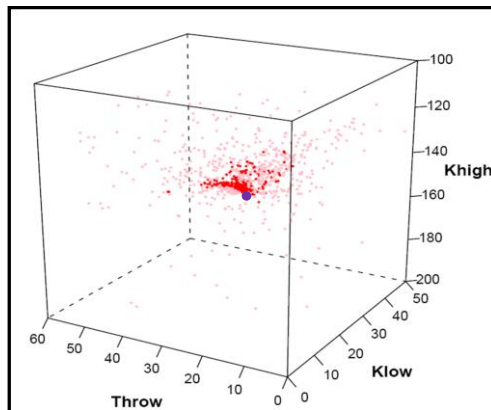
(f) PSO-LIB



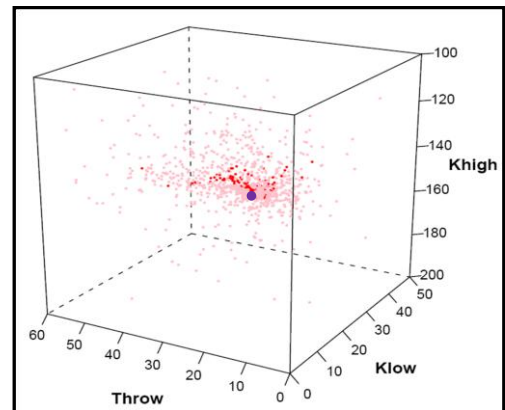
(c) PSO-T1R



(g) PSO-T1B



(d) PSO-T2R



(h) PSO-T2B

Figure 4.37: 3D view – random and reflecting boundary strategies

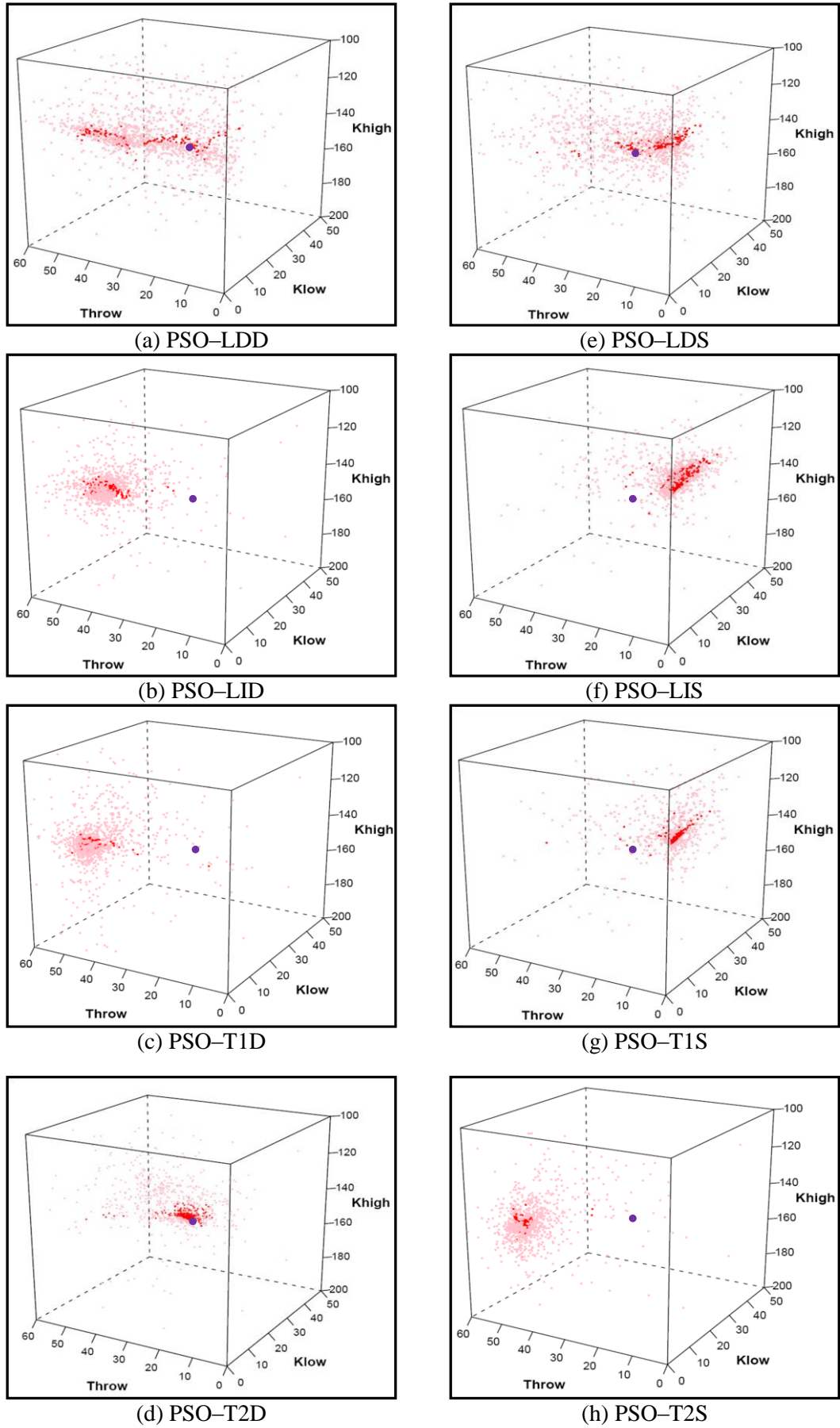


Figure 4.38: 3D view – damping and absorbing boundary strategies

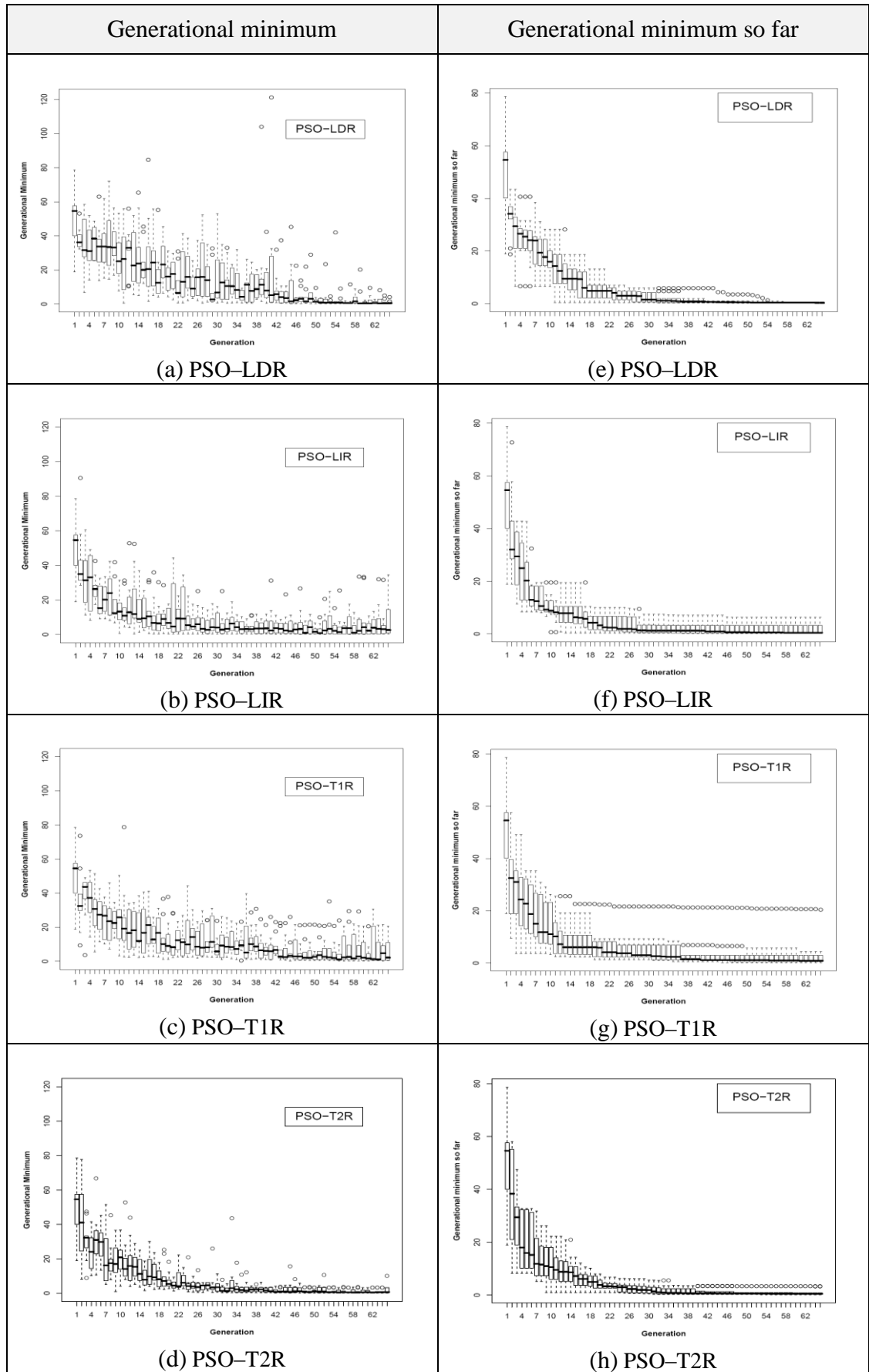


Figure 4.39: Random boundary strategy measures

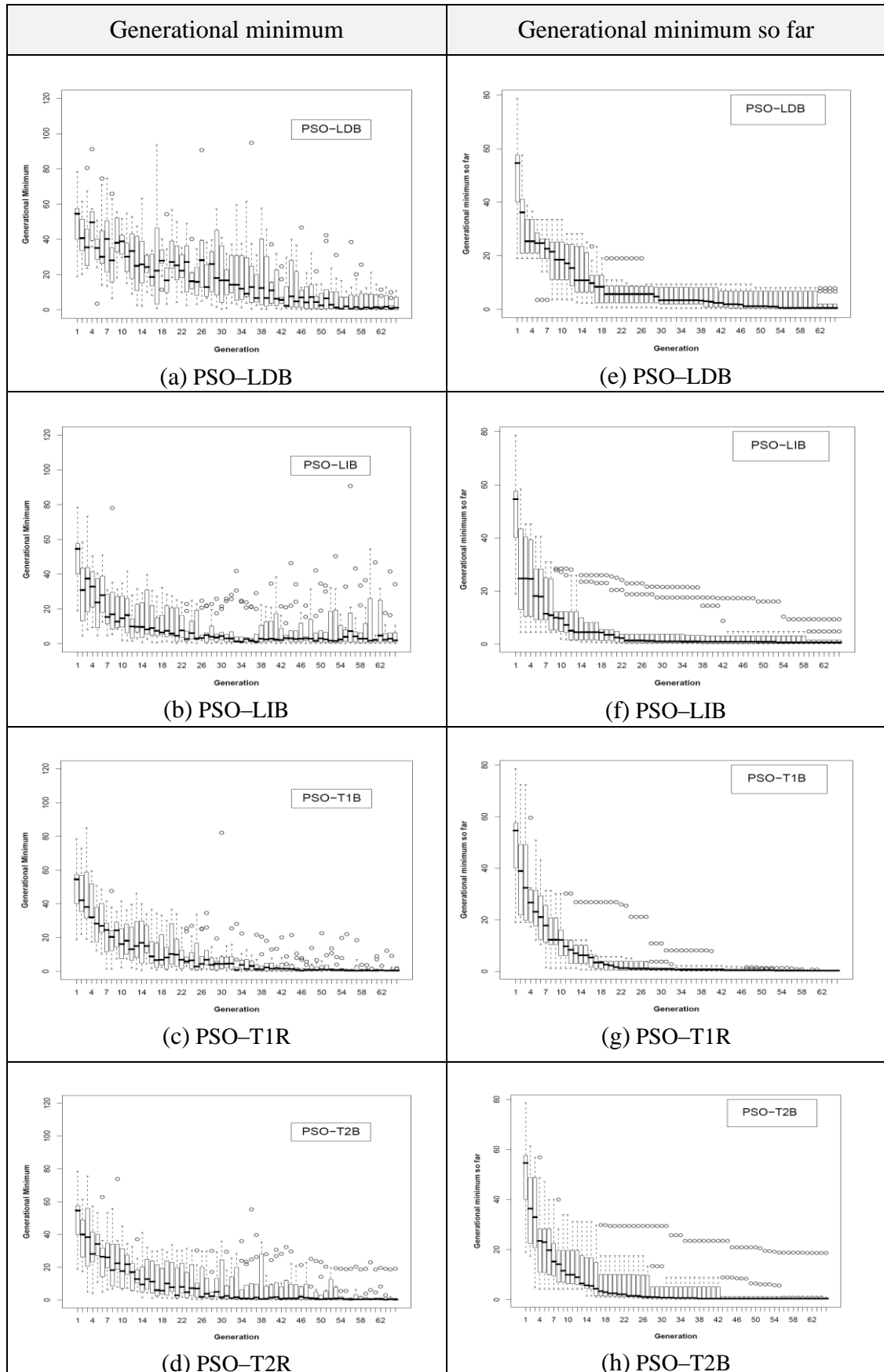


Figure 4.40: Reflecting boundary strategy measures

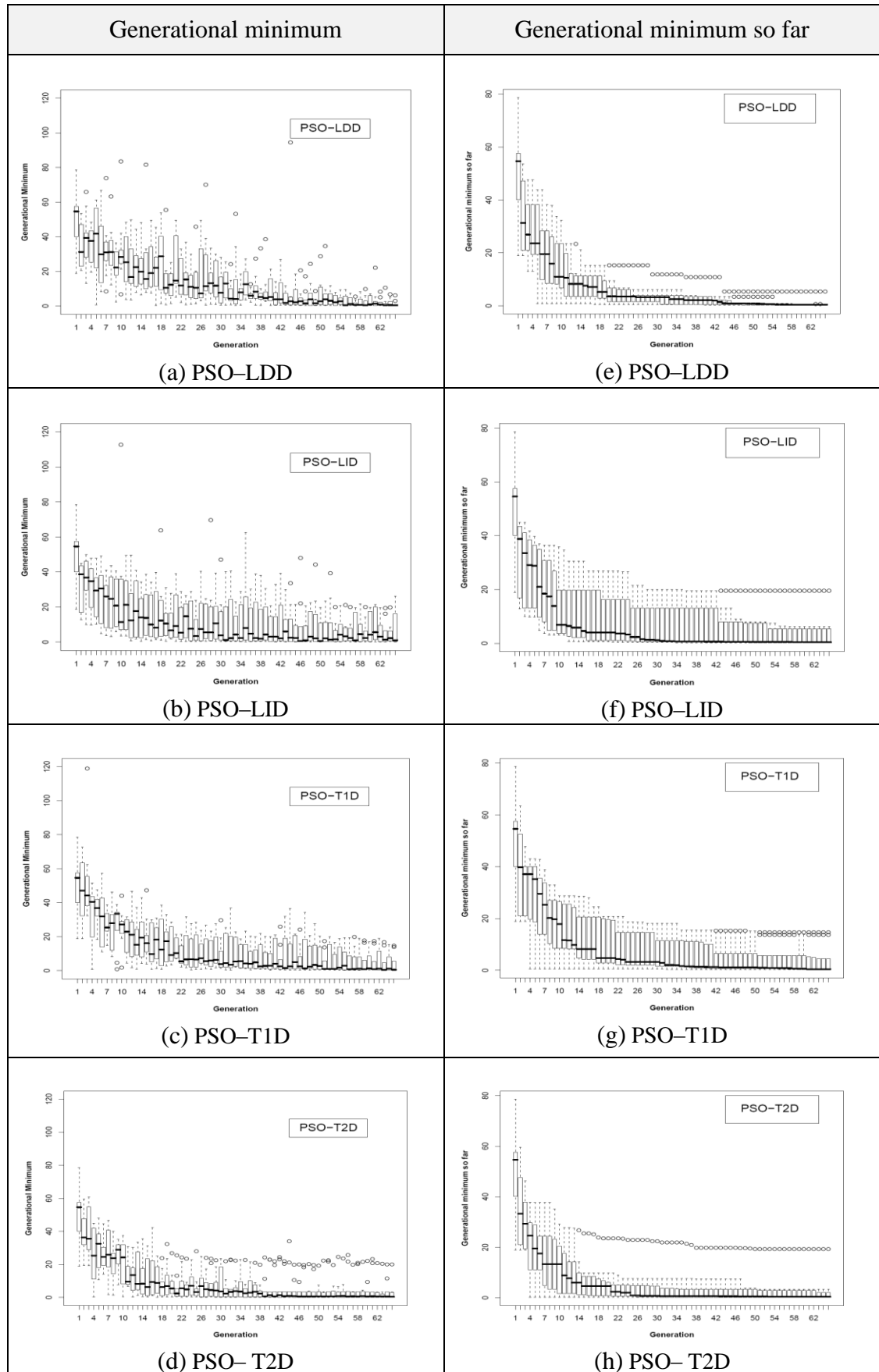


Figure 4.41: Damping boundary strategy measures

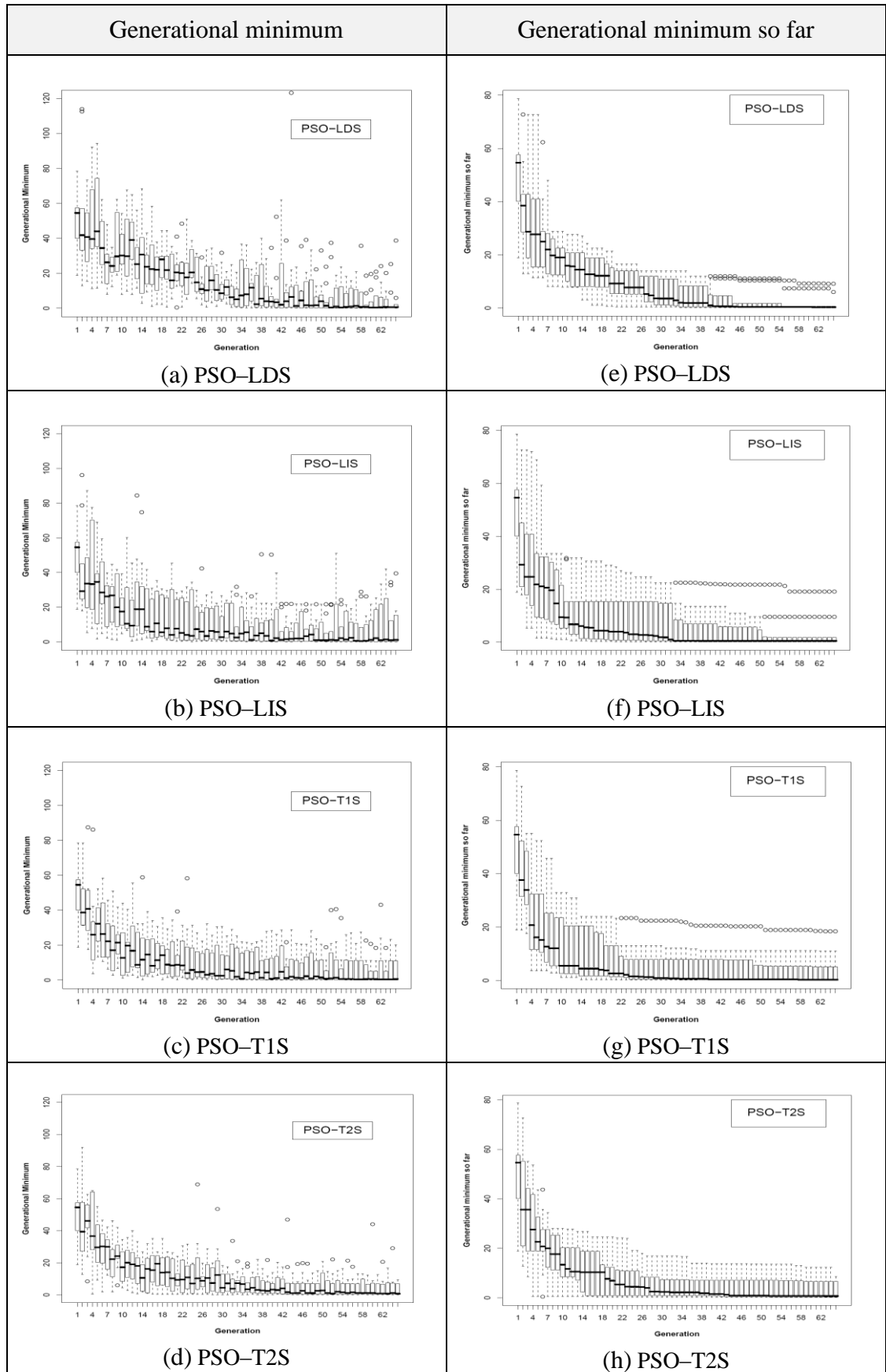


Figure 4.42: Absorbing boundary strategy measures

Plots of (a), (b), (c), and (d) in Figures 4.43, 4.44, 4.45, and 4.46 show the corresponding box plot calculated diversity of the swarm per generation for the 10 performed runs calculated with Eq. (4.5). We can see that the diversity is reducing during the optimisation process which is to be expected. We can also observe that for example that the degree of dispersion of particles in the dynamic PSO–LDB is more than in the PSO–T1B variant as shown in (a) and (c) of Figure 4.44. This could also be supported by Figure 4.34(a) and (c) in which we used sampling history plots to show the performance during the course of the optimisation. The parameter values for the good history matched models can be seen by looking at the range of the blue points. Both variants appear to concentrate on sampling for similar zones, although the dynamic PSO–LDB is able to maintain population diversity more while PSO–T1B improves the sampling (shown in the blue points) as the optimisation progresses. Table 4.9 and Table 4.10 summarise the performance measures per each of the strategies and per each of the inertial weight choices respectively for the 10 seeds. There is a trend in results between diversity and convergence speed observable in the particular per strategy as shown in Table 4.9 where static inertial weight choice has faster convergence compared to the dynamic ones, regardless of boundary strategy used, while the dynamic choices have more diverse solutions compared to the static ones for this example (the comparison is noticeable if we look vertically in the tables to compare both measures). Random and reflecting boundary strategies have more diverse models in comparison to the damping and absorbing boundary strategies (Table 4.10).

Table 4.9: Measures summary per strategy – 10 Seeds

Measure Strategy	Convergence	Diversity
Random	T2R>LDR>T1R>LIR	T2R<T1R<LIR<LDR
Reflecting	T1B>T2B>LIB>LDB	T2B<T1B<LIB<LDB
Damping	T2D>T1D>LDD>LID	T2D<T1D<LID<LDD
Absorbing	T2S>T1S>LDS>LIS	T2S<T1S<LIS<LDS

Table 4.10: Measures summary per inertial weight choice – 10 Seeds

Measure \ IW Choice	Convergence	Diversity
Linear Decrease	LDD > LDR > LDS > LDB	LDD < LDS < LDB < LDR
Linear Increase	LIB > LIS > LIR > LID	LIS < LID < LIB < LIR
Type I'	T1S > T1B > T1D > T1R	T1S < T1D < T1B < T1R
Type I''	T2D > T1S > T2B > T2R	T2D < T2S < T2B < T2R

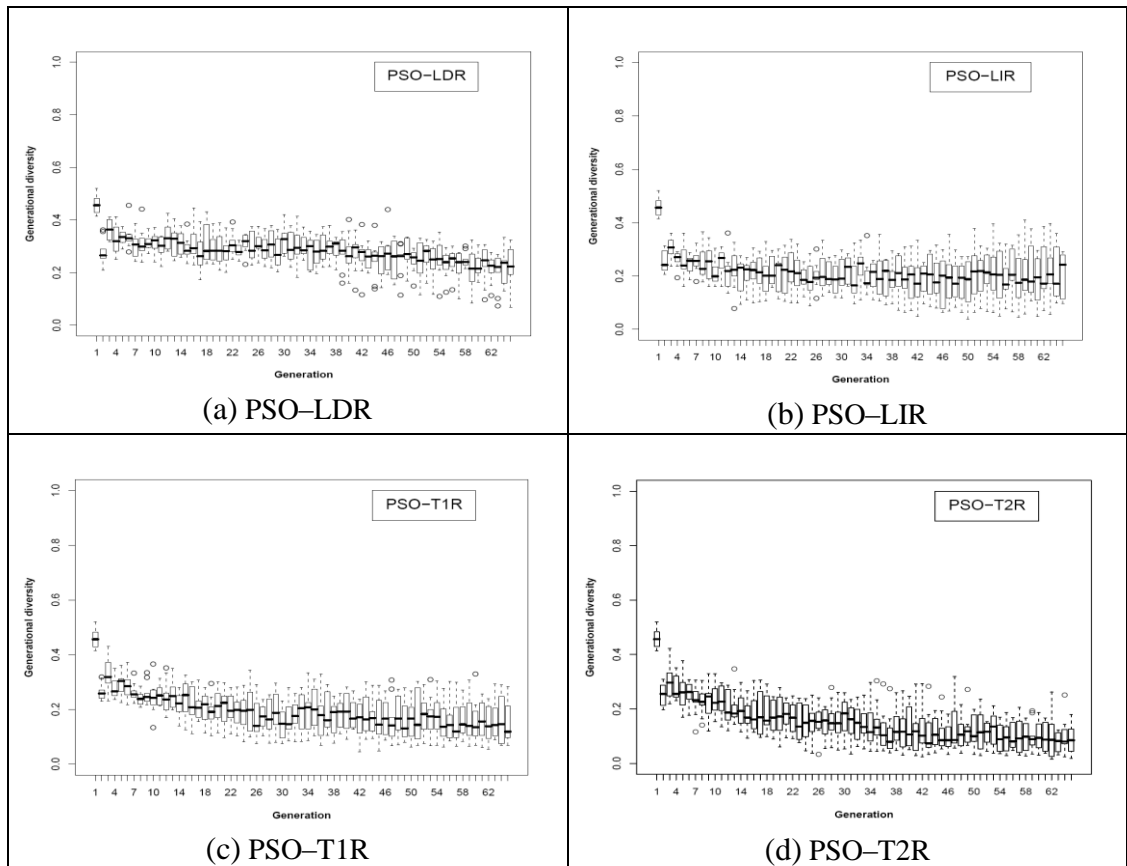


Figure 4.43: Random boundary strategy measures – generational diversity

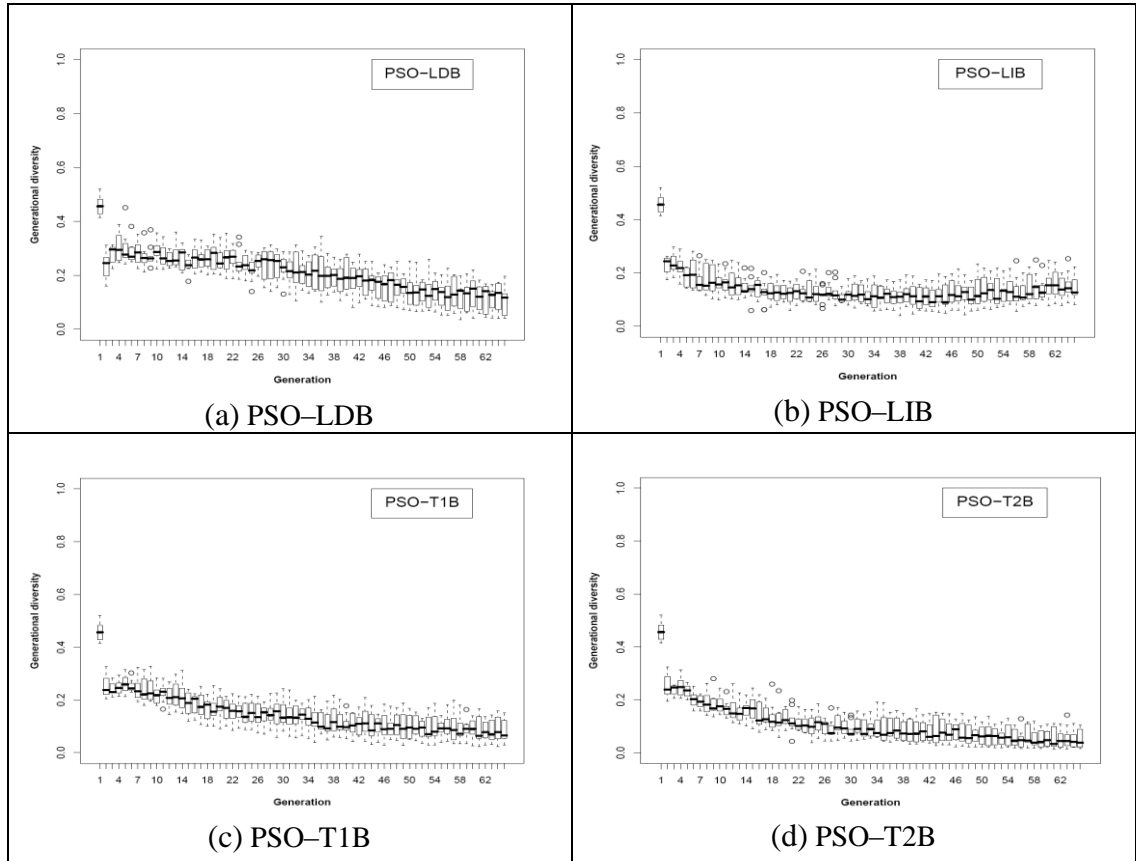


Figure 4.44: Reflecting boundary strategy measures – generational diversity

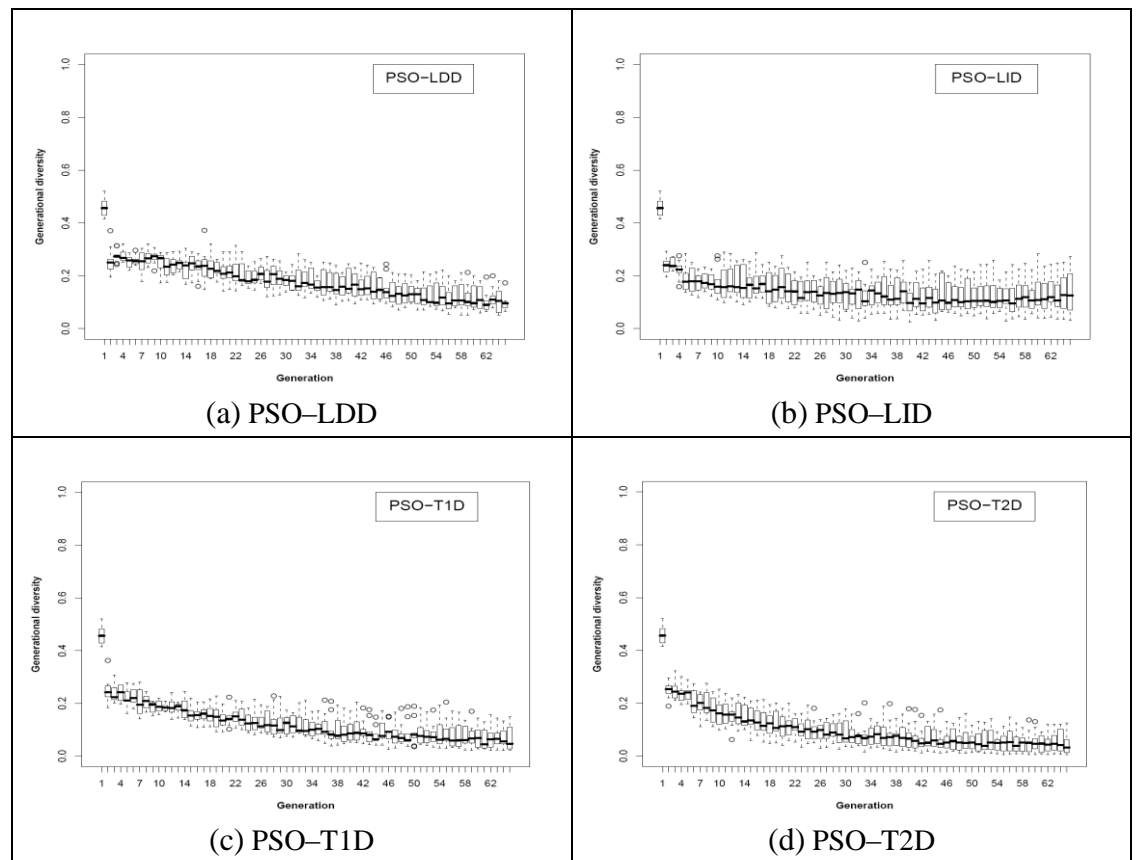


Figure 4.45: Damping boundary strategy measures – generational diversity

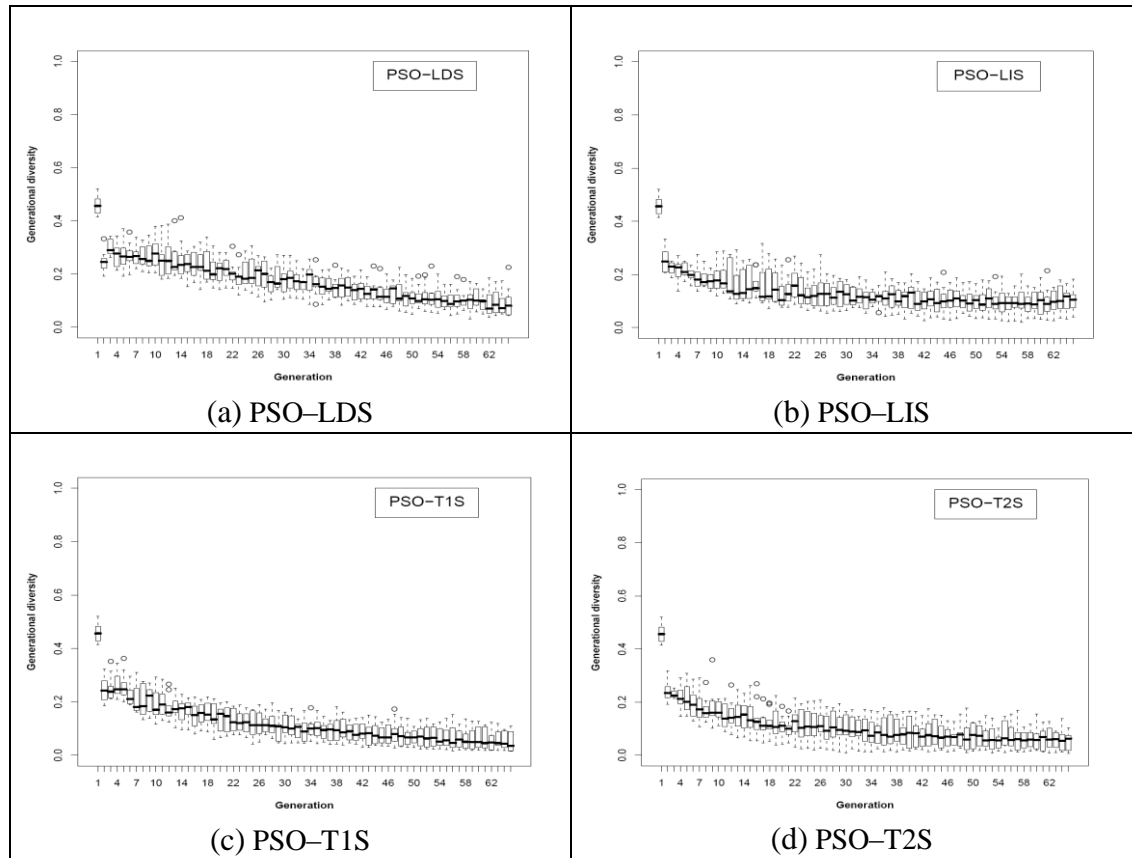


Figure 4.46: Absorbing boundary strategy measures – generational diversity

4.3.2.5 Uncertainty Assessment and Predictions: Comparison of PSO Variants

In the previous section the history matching optimisation performance for the PSO variants was compared in terms of misfit reduction and diversity of models obtained. In this section we show how that influenced our uncertainty predictions using the ensemble of models obtained in the test example and resampled with NAB routine.

Figure 4.47 presents the Bayesian credible intervals ($p10$ – $p50$ – $p90$) of the database benchmark case for oil rate, water rate and total oil recovery after history matching to the first three years of production history. The forecast is carried out for the next seven years to measure the predictive capability of the history matches. The FOPT profile simulated with the truth is below $p10$ of database prediction as depicted in Figure 4.47(c).

Figures 4.48, 4.49, 4.50, and 4.51 show the Bayesian credible intervals ($p10$ – $p50$ – $p90$) for oil rate, water rate and total oil recovery after history matching for all the PSO variants. The uncertainty bounds vary in this example where wider uncertainty estimates are obtained with PSO-LIB, PSO-LDS, PSO-LIS, and PSO-T1S. The total oil recovery prediction after 10 years for IC Fault model is shown in Figure 4.52 in

comparison with the database prediction which is shown in purple. The PSO variants which have obtained relatively close results to the database are: PSO-LDB, PSO-T2B, PSO-LDD, PSO-LID, PSO-T1D, and PSO-T2S as illustrated in Figure 4.52. The PSO variants which have obtained relatively wider uncertainty estimates results are: PSO-LDB, PSO-LDS, PSO-LIS, and PSO-T1S as illustrated in Figures 4.49(i), 4.51(i), 4.51(j), and 4.51(k) respectively. The dashed horizontal line shows the truth equivalent value. Table 4.11 and Table 4.12 show the convergence speed and diversity of models for the particular single runs and how it influences the uncertainty estimation based on how closely it matches the database result and how wide the uncertainty ranges are as shown in Figure 4.52. However, this table result is based on a single run only for each variant.

It is noted that the comparisons from the plots are based on the overall sampling since each sampling point is a simulation that may take hours to run and so the overall performance is important. If the comparison is based solely on the last iteration, the tables may have different behaviour.

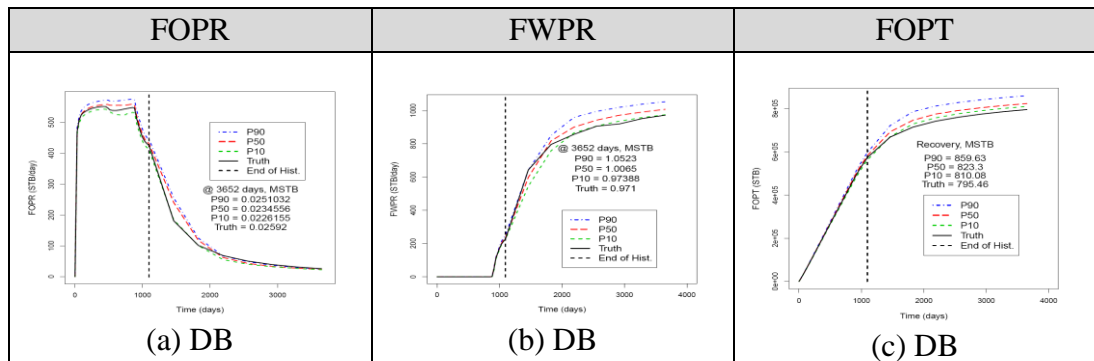


Figure 4.47: Bayesian credible intervals generated with the database

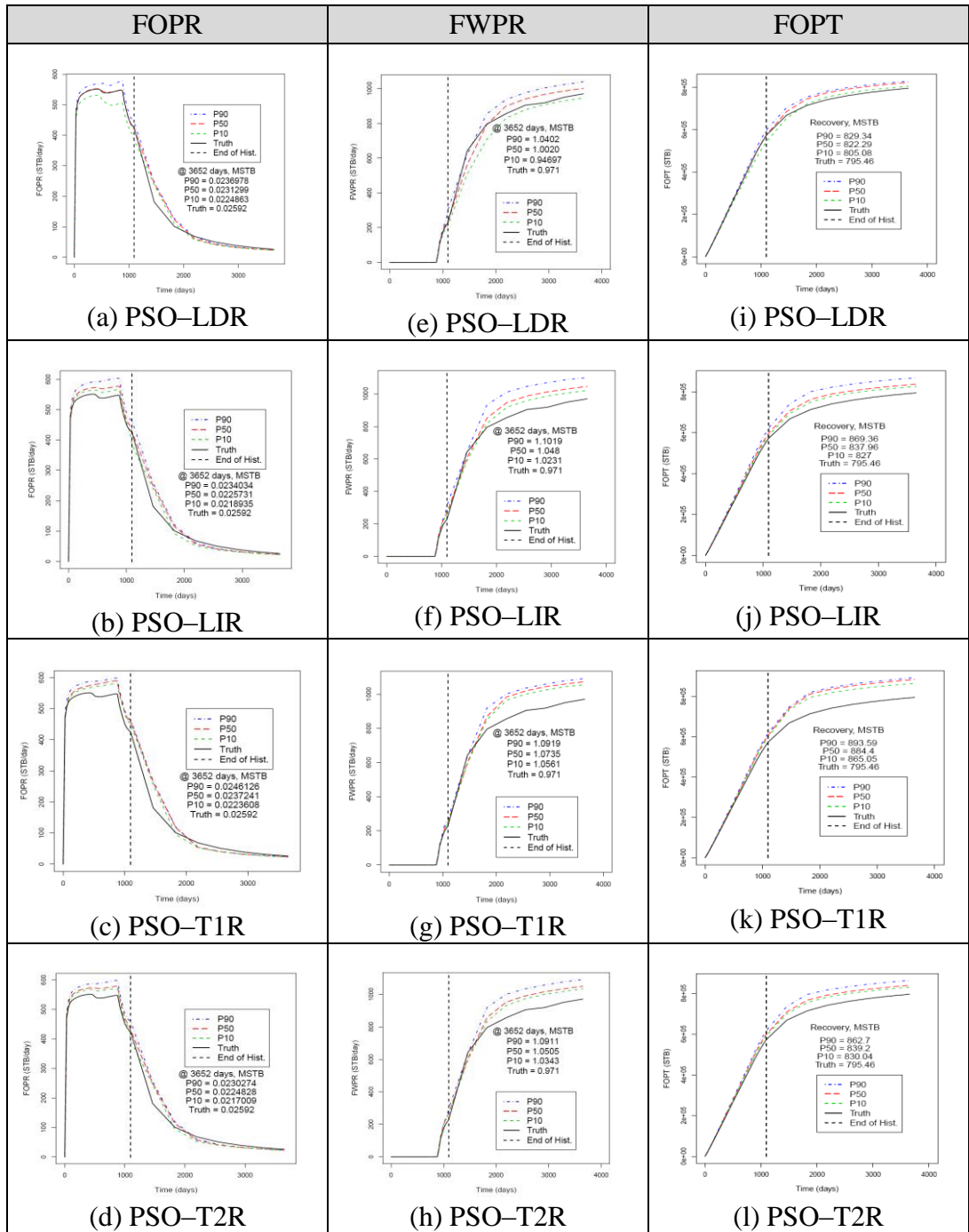


Figure 4.48: Bayesian credible intervals generated with random strategy variants

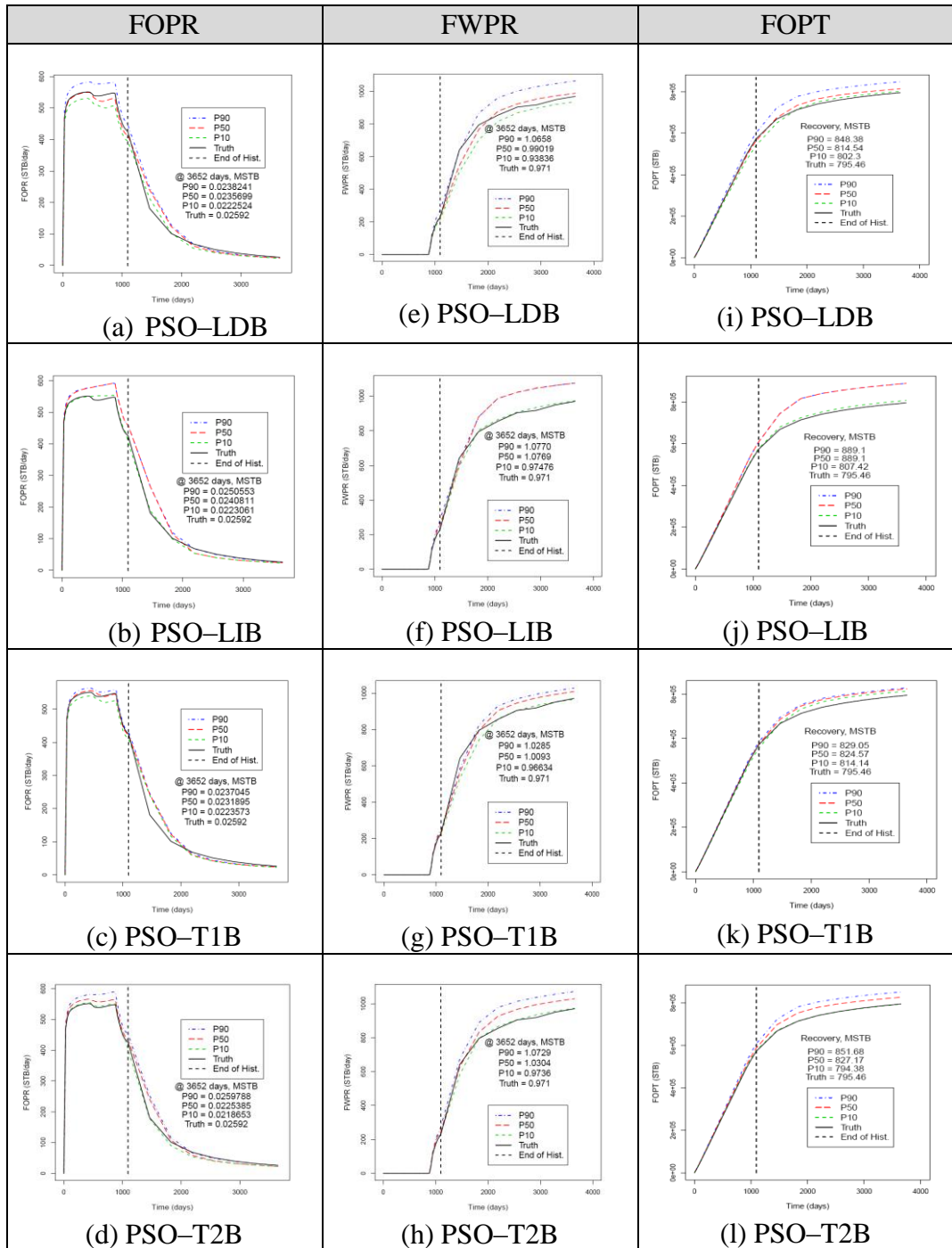


Figure 4.49: Bayesian credible intervals generated with reflecting strategy variants

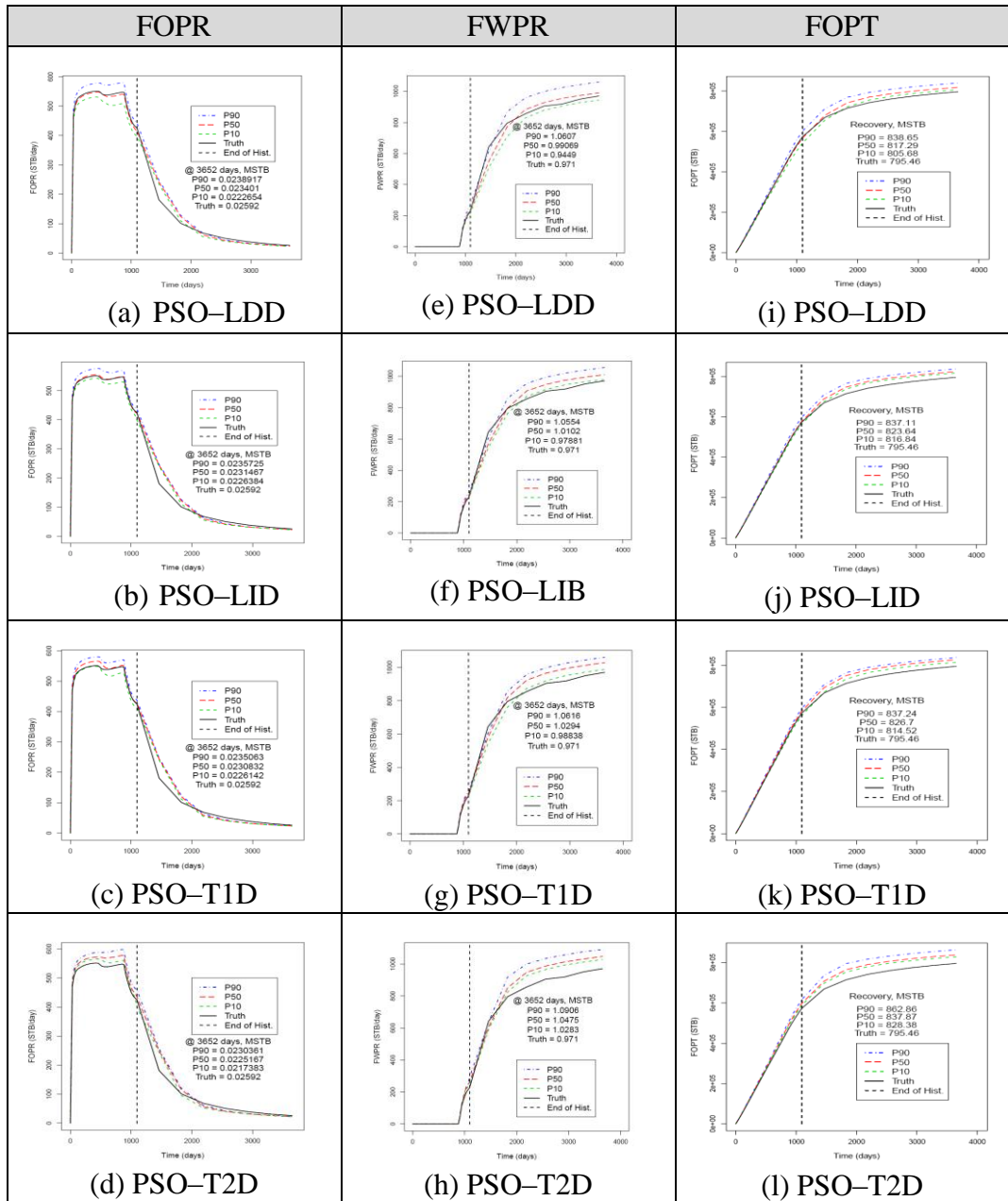


Figure 4.50: Bayesian credible intervals generated with damping strategy variants

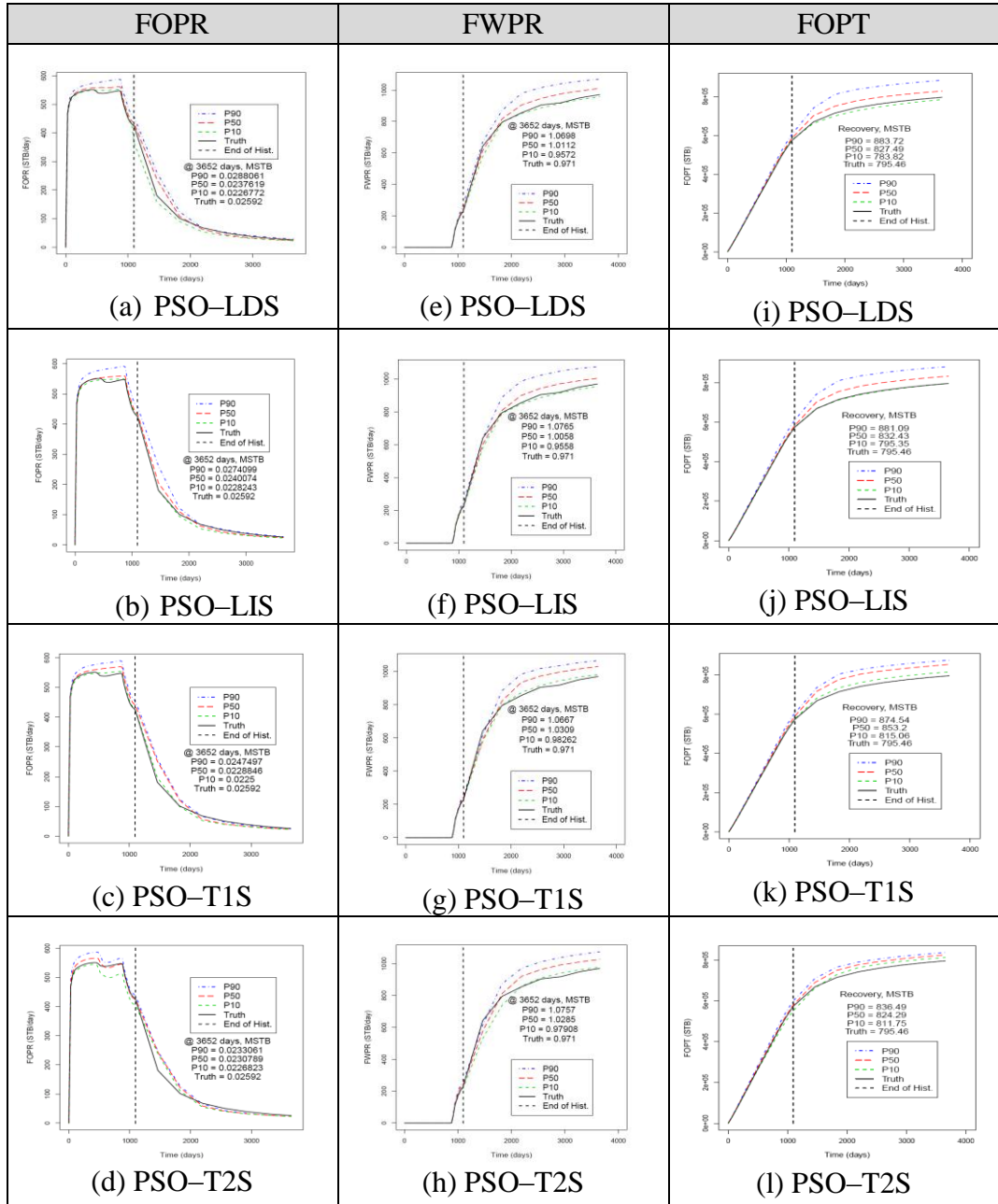


Figure 4.51: Bayesian credible intervals generated with absorbing strategy variants

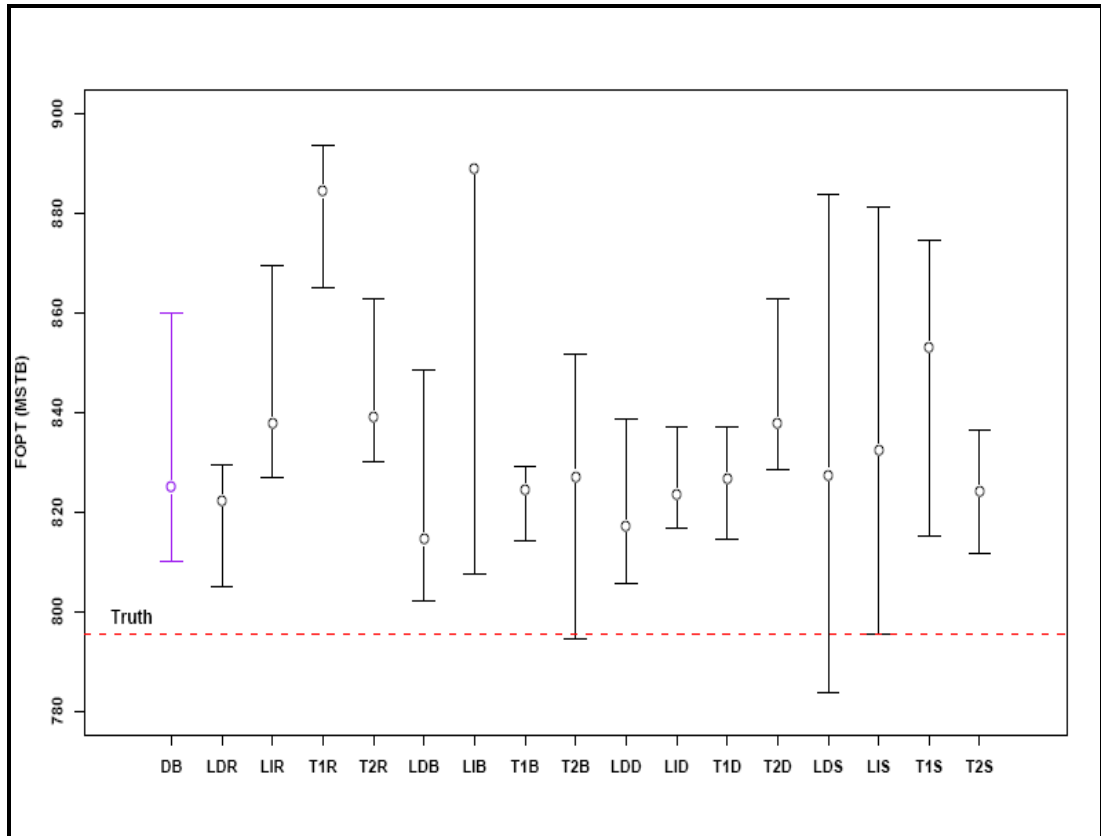


Figure 4.52: Bayesian credible intervals of total recovery prediction for PSO variants

Table 4.11: Measures summary – 1 Seed

Measure \ Strategy	Convergence	Diversity	Uncertainty estimation
Random	T2R>LDR>T1R>LIR	T2R<T1R<LIR<LDR	T1R<T2R<LIR<LDR
Reflecting	T1B>T2B>LIB>LDB	T2B<T1B<LIB<LDB	LIB<T1B<T2B<LDB
Damping	T2D>T1D>LDD>LID	T2D<T1D<LID<LDD	T2D<LID<T1D<LDD
Absorbing	T2S>T1S>LDS>LIS	T2S<T1S<LIS<LDS	T2S<T1S<LIS<LDS

Table 4.12: Measures summary – 1 Seed

Measure \ IW Choice	Convergence	Diversity	Uncertainty estimation
Linear Decrease	LDD>LDR>LDS>LDB	LDD<LDS<LDB<LDR	LDR<LDD<LDB<LDS
Linear Increase	LIB>LIS>LIR>LID	LIS<LID<LIB<LIR	LIB<LIR<LID<LIS
Type I'	T1S>T1B>T1D>T1R	T1S<T1D<T1B<T1R	T1R<T1B<T1D<T1S
Type I''	T2D>T1S>LDB>LIR	T2D<T2S<T2B<T2R	T2R<T2D<T2S<T2B

4.4 Chapter Summary

In this chapter the novel particle swarm optimisation (PSO) algorithm has been introduced in reservoir modelling to generate multiple history matched reservoir models. The results show that algorithms based on swarm intelligence concepts have the potential to be effective tools in uncertainty quantification in the oil industry. The technique has been tested on two reservoir examples. The first example is the Teal South reservoir with 8 unknown parameters and the second example is the 3-parameter model, the IC Fault model. The technique has been applied on the more realistic Brugge model and this is discussed in Chapter 9.

The question of how to tune PSO and adjust it to make the algorithm efficient in solving history matching problem is addressed. This is achieved through the development of simple useful variants of PSO. The variants have flexibility in converging quickly towards good solutions as well as in global exploration depending on the choice of the task and the variant choice. The efficiency of some of the basic modifications of the PSO has been investigated to test the reliability of the technique. Different variants of the method involving four boundary strategies and four inertial weight choices have been studied. It is seen that the forecasted uncertainty envelopes are influenced by two factors: the goodness of fit and diversity of the models in the ensemble obtained. It is shown that PSO could be improved by optimising the PSO control parameters. Some variants converge faster to good fitting regions in parameter space leading to a fewer number of reservoir simulation runs, though others better maintain the diversity of the reservoir models for performed reservoir examples tests. Recommendations and guidelines from the studies help facilitate deployment on the PSO algorithm and how to better employ the algorithm for complex response surface in reservoir model history matching and uncertainty quantification.

It was reported in the literature that many variants obtain good results. However, when tested on the two petroleum studies those conclusions were not supported. For instance in a boundary strategy called the wraparound (Birge, 2003), some of Clerc's formulations that are mentioned to avoid any boundary considerations and some inertial weight choice like nonlinear decreasing, randomly sampled from Gaussian or Peng version (see Engelbrecht (2005)) inertial weight choices have led to divergence of particles. The discussed variants used here are the ones which have obtained good

convergence speed, diversity and quality of history match among them based on the task chosen.

Particular conclusions drawn from the results of the two datasets used could be summarised as follows:

- The use of static inertial weight choices: $c_1 = c_2 = 1.494$, $\omega = 0.729$ (Trelea Set Type I) and $c_1 = c_2 = 1.7$, and $\omega = 0.6$ (Trelea Set Type I") yields low misfit models faster than the use of the dynamic choices: linear decreasing and linear increasing inertial weights. Yet, the dynamic versions obtained more diverse set of models and wider envelopes of uncertainty. This behaviour was consistently repeated for a set of 10 seeds and initial conditions. The results obtained are robust.
- The absorbing and damping boundary strategies obtained similar low misfit models faster with narrower uncertainty bounds in the Teal South example (generally in high dimensional cases as shown for a high dimensional case in Chapter 9, they are fast) and wider uncertainty envelopes in the IC Fault model example. The location of minima in the parameter space has an impact on the differences in the variants results.
- Teal South has a relatively smooth misfit surface though in 8-parameters as it produces good convergence where the difference of the uncertainty estimations results for PSO variants is marginal. The IC Fault model has a much more complex misfit surface with multiple diverse local minima, thus the different behaviour of PSO variants which were studied in terms of convergence and diversity analysis. It is also worth noting that the minima in the Teal South example appear to be located in boundaries and that seems to be true for most high-dimensional cases. There are multiple local minima in the Teal South model but they are not as extreme as for the IC Fault case.
- There is a trade-off between an accurate estimation of model parameters, diversity and quick convergence. The results show that PSO variants used here could all produce reasonably reliable forecasts in the Teal South real field example while most obtained relatively reasonable estimates in the IC Fault model case compared with the database one.

- Since these results have been obtained on simple and synthetic field datasets, further studies on more complex fields will be needed to establish definitive guidelines.

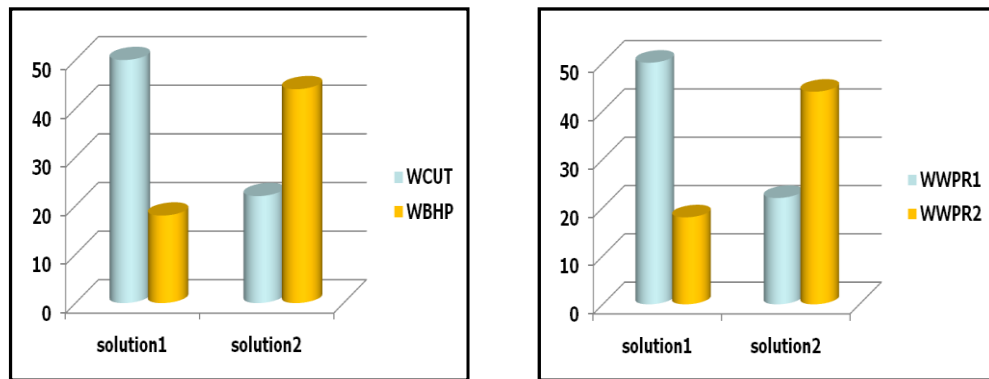
Chapter 5 – History Matching and Uncertainty Quantification: Multi- Objective Particle Swarm Optimisation Approach

5.1 Multi-Objective Optimisation in Petroleum

Research studies in assisted history matching techniques, such as genetic algorithms (Romero et al., 2000a), neighbourhood algorithm (Christie et al., 2006; Nicotra et al., 2005), chaotic approach (Mantica et al., 2002), evolutionary strategies (Schulze-Riegert et al., 2001), and particle swarm optimisation (Mohamed et al., 2010b), primarily focused on a specific optimisation method using a single aggregated objective function. The goals of these methods are to navigate the parameter space for multiple good fitting models quickly and identify as many different optima as possible. Obtaining multiple optima can result in an ensemble of history matches that has divergent prediction profiles for more accurate and reliable predictive uncertainty estimates. Assisted history matching techniques which have been proposed use nonlinear optimisation to minimise the objective function and produce the best least square fit of the historical observed data. The objective function can be defined in least square sense as in Eq. (5.1) in which T is the number of observations, q^{sim} and q^{obs} is the simulated and observed data respectively representing production rates, cumulative or pressure measurements from a reservoir. The term $(q^{sim} - q^{obs})_t$ is called the residual. w_t is the weighting factor and can be estimated with various data analysis and filtering techniques introduced. However, the standard approach or the best choice for each field study still remains an open question.

$$r_e = \sum_{t=1}^T (w(q^{sim} - q^{obs}))_t^2 \quad \text{Eq. (5.1)}$$

Little attention in history matching reservoir simulation research has been drawn to multi-objective optimisation schemes (Schulze-Riegert et al., 2007; Ferraro and Verga, 2009) in which the aim is to find a set of solutions which optimally balance the different objectives rather than the single best solution as in the single aggregated one. History matching problem, as many real-world optimisation problems, is a multi-objective in the sense that well measurements like fluid rates, water rates or water breakthroughs vary in time, space and type. Figure 5.1 shows an example where either one objective is well matched or the other depending on different choices of the weights. Defining the conflicting objectives of complex real field in order to simultaneously optimise more than one objective is a challenging task in reservoir simulation studies and needs the engineer's expertise and careful judgement of results.



(a) WBHP is better matched with solution 1 than in solution 2 while WCUT is better matched with solution 2 than in solution 1

(b) WWPR is better matched for well 1 with solution 1 than in solution 2 while for well 2 is better matched with solution 2 than solution 1

Figure 5.1: Matching one objective worsens the other when solution perturbed

In the multi-objective optimisation scheme, multiple objectives can be defined representing each or some of the weighted squared differences of a production type. By constructing multiple objectives that measure the contribution of each objective in the multi-objective optimisation scheme, it can be possible to find a set of solutions which optimally balances the different objectives simultaneously while maintaining solution diversity. The advantage of this construction is that the tradeoffs between the objectives can be explored and utilised explicitly in the optimisation procedure to find all possible combination of good fitting model solutions that have similar match quality. In history matching, it is desirable to have various solutions that map to relatively similar low misfit values that can represent all the possible geological scenarios.

De Almeida et al. (2001) applied multi-objective approach to production scheduling of petroleum refinery. They reviewed a multi-objective fitness evaluation method called energy minimisation and presented an analysis of the method's behaviour while using genetic algorithm. The numerical results were presented and analysed, leading to an overall assessment of the benefits provided by the multi-objective approach. First attempts on multi-objective history matching optimisation have been tested in reservoir modelling application by Schulze-Riegert et al. (2007) and Schulze-Riegert and Ghedan (2007). Schulze-Riegert et al. (2007) used Pareto-based method known as Strength Pareto Evolutionary Algorithm (SPEA) that utilises the popular Pareto concept among computational intelligence researches. In Schulze-Riegert et al. (2011) multiple optimisation algorithms are used to optimise partial objectives individually worked well. The methodology has been tested on the well placement optimisation problem in a gas condensate field. The goal in this was to find the optimum well trajectories by maximising cumulative gas production. One of the early attempts using the Pareto concept and comparing it with the single objective approach in history matching context is attributed to Ferraro and Verga (2009). They applied the multi-objective genetic algorithm and evolution strategies for history matching and uncertainty quantification of the PUNQ-S3 synthetic case study and compared the results with the single objective scenarios of the algorithms. They showed the benefits of using the multi-objective scheme over the single objective aggregated function. Recently, Busby and Sergienko (2010) have combined probabilistic inversion and multi-objective optimisation for the production development of a reservoir, in which the approach defines different weights for the multiple objective functions by considering independent measurement errors for each time-dependent output. It is noted that in such an approach the posterior distribution is sensitive to the choice of the objective function weights: the higher the assumed measurement errors the wider the posterior distribution. Ciaurri et al. (2010) have used versions of direct search methods (generalised pattern search and Hooke-Jeeves direct search) in combination with nonlinear constraint handling techniques called filtering method in which borrowed ideas from multi-objective optimisation schemes (Pareto) were utilised in order to determine optimum well controls to maximise the net present value as the objective function for generally-constrained production optimisation problems.

Kathrada (2009b) applied a variant of niching in reservoir modelling called sequential niching, to locate multiple solutions. Sequential niching finds and isolates niches so that future explorations do not duplicate sampling in niches that have already been identified, usually by modifying the fitness landscape around the niche. Multi-objective optimisation (MOO) is similar to niching, since both approaches identify multiple solutions to an optimisation problem. The difference is that MOO algorithms have to find a set of solutions which optimally balance the different objectives, whereas niching algorithms locate multiple solutions to a single objective. In brief, there are few multi-objective reservoir history matching application studies using a small number of algorithms and thus the area warrant further investigation.

The goal of this thesis is to develop algorithms for history matching and uncertainty quantification that would be able to find multiple solutions to be used in uncertainty modelling. This chapter tries to narrow the gap between theory and practice in the context of reservoir history matching optimisation. To this end multi-objective particle swarm optimisation (MOPSO) variations have been looked at. In the previous chapter we have reviewed the main concepts of particle swarm optimisation and showed its applications in reservoir modelling. The present chapter extends the application of particle swarm optimisation algorithm to handle MOO in the reservoir history matching. This study provides the first application of extending PSO in petroleum history matching and uncertainty quantification, attempting to fill the research gaps in our knowledge in this respect and to better employ the innovative computing tools and technology available. Part of the work carried out here is reported in Mohamed et al. (2011a).

The chapter is organised as follows. Firstly, the chapter introduces the goals of MOO and reviews different approaches such as aggregation methods, criterion-based methods, and Pareto-optimality or dominance, with a focus later on Multi-Objective Particle Swarm Optimisation (MOPSO) describing the current applications in the literature of MOPSO. The definitions and the overview are based on surveys by Engelbrecht (2005), Deb (2009) and Reyes and Coello (2006) where the popular optimisation techniques have been discussed. The application of a developed novel variant is introduced and we show the application of the multi-objective particle swarm optimisation technique to two challenging synthetic petroleum examples and the results

has been compared with the single objective methodology. Analyses of history matching quality and predictive uncertainty estimation based on the resulted models have been conducted to obtain the uncertainty predictions envelopes for both strategies. The comparative results suggest that the multi-objective particle swarm approach obtains better history matches and has achieved over twofold faster convergence speed than the single objective approach in the first example while in the second the results has depended on how the objective functions are constructed.

5.2 Basic Multi-Objective Optimisation Concepts

The goal in single objective optimisation (SOO) where only one objective is optimised is to find the global optimum. The definition of optimality in MOO is not simple. The main problem is the presence of conflicting objectives, where improvements in one objective may cause deterioration in another. Trade-offs exist between such conflicting objectives, and the task is to find solutions which balance these trade-offs. Such a balance is achieved when a solution cannot improve any objective without degrading one or more of the other objectives. These solutions are called the non-dominated solutions, of which many may exist. A number of definitions are provided below for MOO demonstration. Assuming the minimisation problems:

Minimise

$$f(x) = [f_1(x), \dots, f_k(x)] \quad \text{Eq. (5.2)}$$

Subject to:

$$\begin{aligned} g_i(x) &\leq 0 & i = 1, 2, \dots, m \\ h_i(x) &= 0 & i = 1, 2, \dots, p \end{aligned} \quad \text{Eq. (5.3)}$$

where x is the parameter vector where $x \in [x_{min}, x_{max}]$, g_i, h_i are the inequality and equality constraints of the problem, and $f_i, i = 1, \dots, k$ are the objective functions. To describe the concept of optimality, we will introduce the next few definitions from Deb (2009), Engelbrecht (2005), and Reyes and Coello (2006).

Definition 5.1. Domination: A solution, x_1 , dominates a solution, x_2 (denoted by $x_1 \prec x_2$) if and only if

- x_1 is not worse than x_2 in all objectives, i.e. $f_i(x_1) \leq f_i(x_2), \forall i = 1, \dots, k$, and
- x_1 is strictly better than x_2 in at least one objective, i.e. \exists some $i \in \{1, \dots, k\}$ for which $f_i(x_1) < f_i(x_2)$.

An objective vector, f_1 , *dominates* another objective vector, f_2 , (denoted by $f_1 \prec f_2$) if f_1 is not worse than f_2 in all objective values, and f_1 is better than f_2 in at least one of the objective values. The striped area depicted in Figure 5.2 illustrates the concept of dominance for a two-objective function, $f(x) = (f_1(x), f_2(x))$ in which the area of objective vectors is dominated by f .

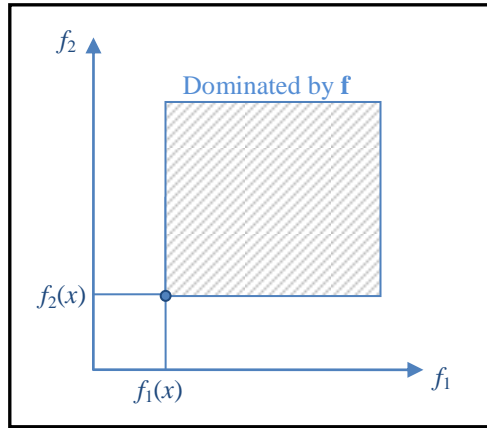


Figure 5.2: Pictorial view of dominance concept (Source: (Engelbrecht, 2005))

Definition 5.2. Weak domination: A solution vector x_1 , weakly dominates a solution vector x_2 (denoted by $x_1 \preceq x_2$) if and only if

- x_1 is not worse than x_2 in all objectives, i.e. $f_i(x_1) \leq f_i(x_2), \forall i = 1, \dots, k$.

Definition 5.3. Pareto-optimal: A solution vector $x^* \in F$ (where F is the feasible region) is Pareto-optimal if it is non-dominated with respect to F . That is there does not exist a solution vector, $x \neq x^* \in F$, that dominates it ($\nexists x: f_i(x) < f_i(x^*)$). An objective vector $f^*(x)$, is Pareto-optimal if x is Pareto-optimal.

The concept of Pareto-optimality was firstly brought up by Francis Ysidro Edgeworth. The Italian economist mathematician, Vilfredo Pareto, generalised the concept in his work *Manual of Political Economy*. The concept has been named after him (Engelbrecht, 2005). In economics terminologies, Pareto-optimality stands for the event when a society enjoys maximum *ophelimity* when no one can be made better off without making someone else worse off.

Definition 5.4. Pareto-optimal set: The set containing the solutions or balanced trade-offs for the multi-objective problem and could be defined as Eq. (5.4).

$$P^* = \{x^* \in F \mid \nexists x \in F : x \prec x^*\} \quad \text{Eq. (5.4)}$$

Definition 5.5. Pareto-optimal front: It is defined by the set which contains all the objective vectors corresponding to parameter vectors which are not dominated by any other parameter vector, $PF^* = \{f = (f_1(x^*), f_2(x^*), \dots, f_k(x^*)) \mid x^* \in P\}$.

The MOO tries to determine the Pareto optimal set from the set F of all the parameter vectors. In some applications, though, the whole of the Pareto optimal set is not normally attainable or sought-after as it may not be desirable to have various solutions that map to the same values in objective function space (Reyes and Coello, 2006).

Definition 5.6. ε -Domination: A solution vector, x_1 , ε -dominates a solution vector, x_2 (denoted by $x_1 \prec_\varepsilon x_2$) for some $\varepsilon > 0$, if and only if

- $f_i(x_1)/(1+\varepsilon) \leq f_i(x_2), \forall i = 1, \dots, k$, and
- \exists some $i \in \{1, \dots, k\}$ for which $f_i(x_1)/(1+\varepsilon) < f_i(x_2)$

Figure 5.3 illustrates the concept of ε -dominance for a two-objective function in which the shaded area being dominated has been extended by a value proportional to the parameter ε which is to be defined by the user.

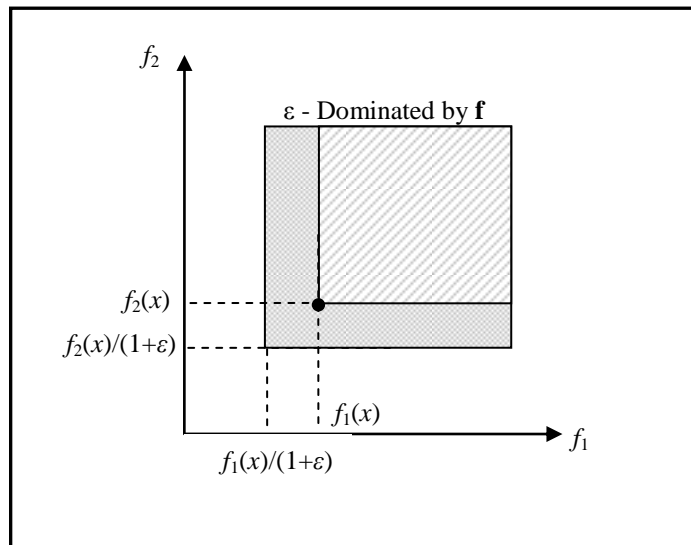


Figure 5.3: Pictorial view of ε -dominance concept (Source: (Engelbrecht, 2005))

Definition 5.7. Objective space: One of the differences between multi-objective and single objective optimisation is that the former contains a multi-dimensional space known as the objective space, O , besides the common parameter space (Deb, 2009). For each solution x in the parameter space, there exists a point in the objective

space $f(x) = o = [o_1, o_2, \dots, o_k]^T \in O \subseteq \mathbb{R}^k$. Figure 5.4 depicts the two spaces and mapping between a d -dimensional solution vector and a k dimensional objective vector.

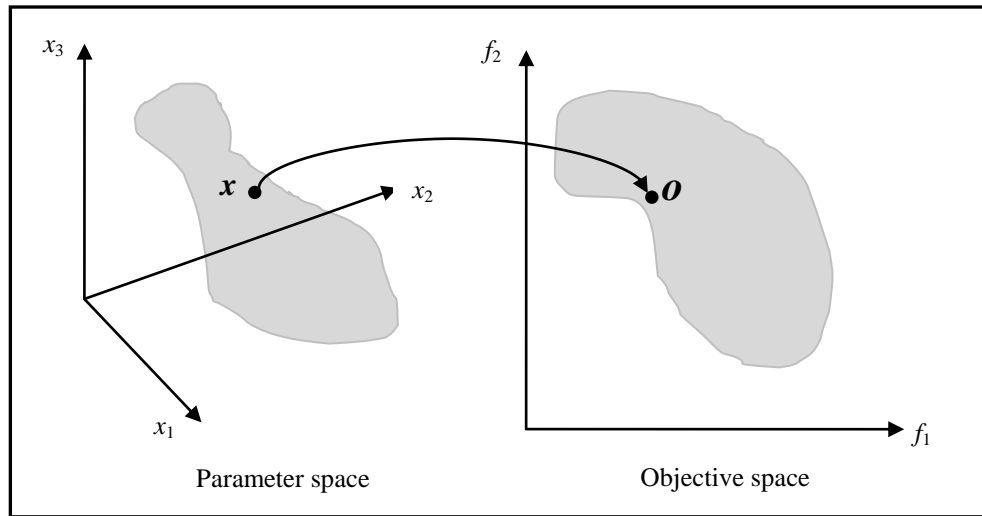


Figure 5.4: The parameter space and the corresponding objective space pictorial (Source: (Deb, 2009))

The term vector optimisation is sometimes used interchangeably to multi-objective optimisation since a vector of objectives rather than a single objective is optimised.

5.3 Objectives and Mechanisms of MOPSO

The main objective of MOO is to find a set of solutions which optimally balances the trade-offs among the different objectives. That is to find the set of non-dominated solutions or the Pareto-optimal set. Solving multi-objective problems entails achieving three main goals for obtaining all the solutions that estimate the Pareto front (Zitzler et al., 1999):

1. Minimise the distance between solutions and the Pareto front (if known).
2. Maximise the diversity and spread of the non-dominated solutions to represent as much as possible of the Pareto front.
3. Maximise the number of elements of the Pareto optimal set found and maintain already found ones.

Research studies addressed these objectives mainly with the following.

1. The first objective is addressed by defining a fitness function (so called *leader*) to quantify the quality of a solution with respect to the multiple objectives. Three types of leaders have been utilised in the literature to favour the selection of non-dominated solutions over dominated ones:

- a. Aggregation-based methods: These are the simplest approaches that can handle MOO problems by aggregating the objective functions into a single objective as a weighted sum and then applying the single objective optimisation algorithm.

In the aggregation approaches the algorithm is designed to obtain the optimum. Therefore, the algorithm has to be repeated many times in order to obtain different solutions and even with that there is no guarantee that different solutions will be found. Some authors use niching strategies to locate multiple solutions. Furthermore, the class of aggregation methods can only be applied when the Pareto front is concave to produce solutions of the Pareto-optimal set despite the weights values choice (Das and Dennis, 1997; Jin et al., 2001).

- b. Criteria-based methods: These techniques do not handle all objectives simultaneously but rather different phases of the optimisation course operate on different objectives.
- c. Pareto dominance-based methods: These methods make use of leader selection techniques based on Pareto preference to find a set of non-dominated solutions and maintain diversity of the estimated Pareto front. This is achieved usually by utilising a repository or archive of all located non-dominated solutions. Several variations developed by different researchers as a result of various schemes exist.

Additional criteria and quality measures have been proposed in the MOO area for leader selection based on density measures promoting diversity and closeness of particles within the swarm. The most important measures are:

- a. Crowding distance

The crowding distance of a non-dominated solution provides an estimate of the density of solutions surrounding it. It is also known as the nearest neighbour density estimator (Deb et al., 2002). It is estimated by the size of the largest cuboid enclosing each particle without including any other. The areas with a larger crowding distance are preferred for selection of local leader in particular. This measure is detailed in Section 5.5.5.

- b. Niche count

This quality measure defines a neighbourhood of each particle sharing resources with others in terms of a parameter called σ_{share} which indicates

the radius of the neighbourhood. A niche count, η_i , is computed for each particle as the number of other particles within a σ_{share} distance from i . These neighbourhoods of particles are termed *niches*. The fitness of a particle is deteriorated relatively to the number and closeness to particles which enclose it within a certain perimeter (Deb and Goldberg, 1989; Goldberg and Richardson, 1987; Reyes and Coello, 2006). That is to say when the niche of a particle is less crowded, with a lower niche count, that particle is favoured as depicted in Figure 5.5. The efficiency of the strategy depends on the value of σ_{share} selected by the user. An adaptive version of niche radius has also been suggested (Fonseca and Fleming, 1993). This measure is also known as kernel density estimator.

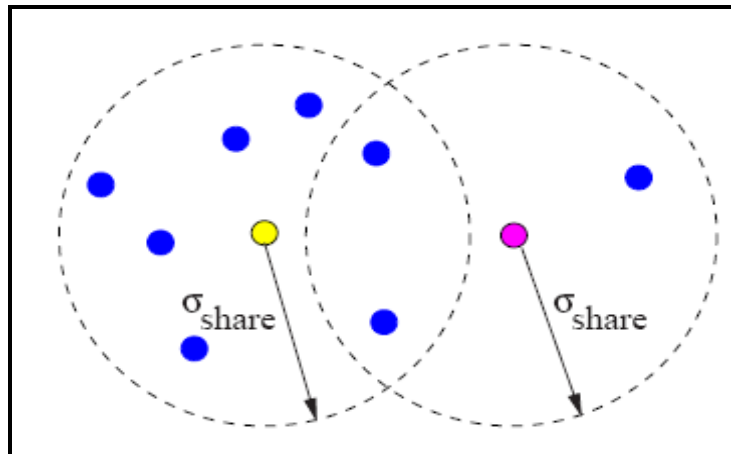


Figure 5.5: For each particle's niche, the particle whose niche is less crowded is favoured (Source: (Reyes and Coello, 2006))

c. Random replacement

In this mechanism, overcrowded regions with particles of the front are deleted and replaced with randomly generated particles. Either crowding distance (choosing particles with smallest crowding distance) or niche count (choosing particles with largest niche count) can be used for selecting particles for replacement. New particles will then select a solution as its leader from the least populated regions of the front. The leaders are selected by sorting all the non-dominated solutions in decreasing order of crowding distance or increasing order of niche count. The leader is then chosen randomly from the top of this list.

2. The second objective is addressed by promoting diversity mechanisms of non-dominated solutions to increase the probability of attracting new solutions

towards low-density areas of the Pareto front. The rapid loss of diversity within the swarm during the optimisation process is the main cause of premature convergence and entrapment into a single solution. Diversity within the swarm could be preserved through the selection of leaders. Nevertheless, it can also be achieved through two main schemes for constructing new solutions:

a. Updating the positions

- i. Neighbourhood topology of the swarm determines how rapidly the information will be exchanged within the particles in the swarm as the leader is determined as soon as the topology is chosen. A star (or full-connected) topology, as mentioned in Section 4.1.2 will lose the diversity faster since it has faster rate of exchanges through the global best of the swarm. Smaller neighbourhoods therefore facilitate diversity of solutions for a longer period.
- ii. Inertia weight, defined in Eq. (4.4), can assist in enhancing diversity and spread of solutions in the swarm since it influences the amount of the previous flight taken to the current one. Large values facilitate exploration while a small value encourages local exploitation.

b. Mutation operator usage

The purpose of this strategy is to add some mutation (craziness or unconsciousness) to a particle. When a swarm stagnates, that is, when the velocities of the particles are almost zero causing the whole swarm to be trapped in local optima, the mutation reflects the change in a particle's flight. Mutation is achieved by randomly changing each component of a single particle position (or velocity) with certain probability. This mechanism potentially presents a way of escaping local optima and speeding up the search if the new leader attracting particle is the mutated particle (Stacey et al., 2003). Several mutation operators have been introduced that mutate components of either the position or the velocity of a particle.

3. The third objective is addressed by maintaining an external archive to retain the non-dominated solutions previously found along the entire course of the optimisation. This is similar to elitist strategy in EAs in which a repository is

used to store best solutions. A solution goes into the archive either if it is non-dominated solution with regard to the solutions stored in the archive or if it dominates all solutions in the archive in which case the dominated solutions may be eliminated from the archive. In addition, to ensure that the found solutions are maintained, the archive is used to select the global best and personal best positions, which are non-dominated solutions for each particle. In this case they are called the global best and local best leaders.

Different design aspects of archives have been proposed. The most important aspect is determining the size of the archive. Permitting the archive to grow indefinitely has the advantages that a good diversity of solutions is facilitated through unlimited archives and the number of non-dominated solutions that can be located is not bounded. This means archiving approaches can use smaller swarms than non-archiving algorithms. However, the drawback of the unrestricted archives is that the computational complexity increases substantially as the archive size and number of objectives increase. This is largely due to non-dominance ranking and leader selection computations required. If all particles go into the archive, at each iteration as the worst case scenario, each update will have a computational complexity of $O(kN^2)$, where k is the number of objectives and N is the size of the swarm.

Several approaches have been developed to address the computational complexity of archiving algorithms. Different data structures can be used for faster comparison of the stored non-dominated solutions and the clustering scheme in which non-dominated solutions are replaced by each cluster centroid. The selection of non-dominated solutions is made from the merging of two consecutive swarms. The truncation approach which limits the archive size is one of the common approaches being used in MOPSO algorithms. Even though, the truncation approach imposes an upper limit on the computational complexity, additional aspects arise: the non-dominated solutions already found may get lost because of the deletion required when the archive reaches its limit capacity, the loss of diversity in the Pareto-optimal set, selection of the non-dominated solution to be deleted, and the selection of the non-dominated solutions of the current swarm to get into the archive. Different archiving

approaches have different answers to these questions. A summarised survey of the deletion and selection schemes is presented in Bartz-Beielstein et al. (2003).

Relaxed forms of dominance have also been suggested and the major one has been ε -dominance, depicted in Figure 5.3. The size of the final external archive in this approach is dynamic and depends on a predefined parameter ε . Laumanns et al. (2002) proposed using ε -dominance as a way of filtering solutions in the external archive. The main idea is to define a set of boxes of size ε where only one non-dominated solution is retrieved from each box. This is illustrated in Figure 5.6, for two objective functions in which the one closest to the lower left hand corner is retrieved. In the plot solution 1 dominates solution 2, thus, solution 1 is favoured. Solutions 3 and 4 are incomparable (in terms of dominance as $f_1(3) < f_1(4)$ and $f_2(4) < f_2(3)$). However, solution 3 is favoured over solution 4, because solution 3 is the nearer to the lower left-hand corner corresponding to the location $(2\varepsilon, 2\varepsilon)$. Solution 5 dominates solution 6, thus, solution 5 is favoured. Solution 7 is rejected because the enclosing box, corresponding to the location $(2\varepsilon, 3\varepsilon)$ is dominated by the box corresponding to the location $(2\varepsilon, 2\varepsilon)$. This procedure ensures that the retrieved solutions are non-dominated with regards to all solutions produced during the search. As this places a limit on the computational complexity, the ε -dominance concept also improves diversity. This is shown in a comparative study with existing clustering approaches for fixing the archive size in Mostaghim and Teich (2003b). They obtained comparable diversity and convergence to the clustering strategies.

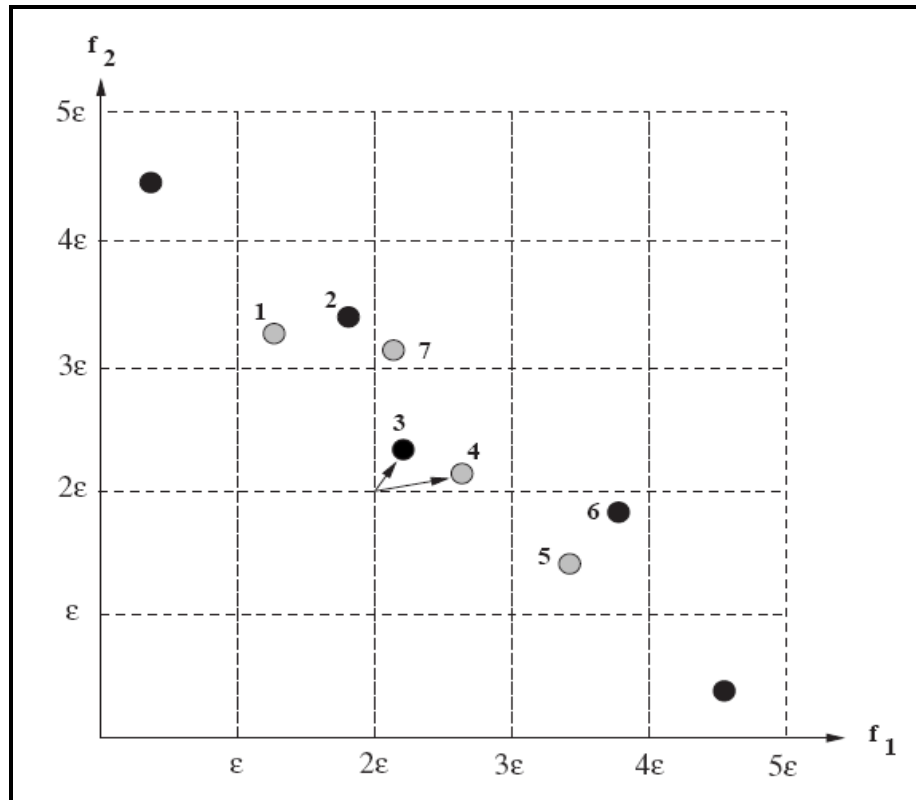


Figure 5.6: Use of ε -dominance in an external archive (Source: (Reyes and Coello, 2006))

Algorithm 5.1 and Algorithm 5.2 show the pseudo-codes for the general PSO and MOPSO algorithms respectively. The significant changes to the PSO to solve MOO problems are *italicised* in Algorithm 5.2. The MOPSO algorithm starts by initialising the swarm and the leaders with non-dominated particles of the swarm being stored in the external repository. In each generation, for each particle, a leader is selected by computing a selection criterion measure which quantifies the quality of the leader (crowding distance or niche count, etc) and the update of velocity and position is carried out as in the general PSO algorithm. The mutation procedure may then be performed on the produced solutions. The particle objective functions are then evaluated and its corresponding personal best *pbest* is updated. The replacement of a particle's *pbest* by its new self is generally carried out either when they are both non-dominated with regard to each other or when the new particle is non-dominated. At the end of the entire swarm update, the leaders are updated. The final step in the process is re-computing the selection criterion measure of the leaders and a replacement procedure is carried out as required if archive is full. The procedure is repeated for either a specified number of iterations or a specified objective functions values.

Algorithm 5.1 General PSO algorithm pseudo-code	Algorithm 5.2 General MOPSO algorithm pseudo-code
<pre> Begin iter = 0 Initialise the swarm Select leader Repeat For each particle Update velocity Update position Evaluate Misfit Update personal best <i>pbest</i> End For Update leader Next iteration (<i>iter++</i>) Until stopping criteria is true End </pre>	<pre> Begin iter = 0 Initialise the swarm Select leaders <i>in an external archive</i> Evaluate(<i>leader</i>) Repeat For each particle Select leader Update velocity Update position Mutation Evaluate Misfit Update personal best <i>pbest</i> End For Update leaders <i>in the external archive</i> Random replacement when archive is full Evaluate(<i>leader</i>) Next iteration (<i>iter++</i>) Until stopping criteria is true Report results <i>in the external archive</i> End </pre>

5.4 Categorisation of MOPSO Techniques

We highlighted earlier the point that history matching problems are multi-objective in nature, since they normally have several (possibly conflicting) objectives that must be satisfied simultaneously. Currently, there is a large amount of mathematical programs of multi-objective optimisation techniques, each one corresponding to a different understanding of the term “optimum” depending on the designer’s choice of which one better fits to the application. We have, however, reviewed earlier the general platform for implementing MOPSO that allows the testing and comparison of existing and future MOO techniques. A categorisation of various MOPSO techniques based on the leader selection (or fitness function) is provided in Table 5.1 where their major characteristics are highlighted (Engelbrecht, 2005; Reyes and Coello, 2006).

Table 5.1: MOPSO variants (adapted from source: Reyes and Coello (2006))

MOPSO Variant	Neighbourhood Topology	Leaders Selection	External Archive	Dynamic ω	Mutation Operator
Aggregation-based methods					
Parsopolous and Vrahatis (2002a)	<i>fully connected</i>	single-objective	no	yes (1.0 \rightarrow 0.4)	no
Baumgartner et al. (2004)	<i>fully connected</i>	single-objective	no	no	no
Criterion-based methods					
Hu and Eberhart (2002)	<i>ring</i>	single-objective	no	yes <i>rnd</i> (0.5, 1.0)	no
Parsopoulos et al. (2004)	<i>fully connected</i>	single-objective	yes	no	no
Zhang et al. (2003)	<i>fully connected</i>	composite leader (relative to each other)	no	yes (0.8 \rightarrow 0.4)	no
Pareto-Based approaches					
Moore and Chapman (1999)	<i>ring</i>	dominance	no	no	no
Ray and Liew (2002)	<i>fully connected</i>	density estimator	yes	no	no
Fieldsend and Singh (2002)	<i>fully connected</i>	dominance & closeness	yes	no	yes
Coello and Salazar-Lechuga (2002) and Coello et al. (2004)	<i>fully connected</i>	density of solutions	yes	no	yes
Pulido and Coello (2004)	<i>fully connected</i>	randomly	no	no	no
Mostaghim and Teich (2003a)	<i>fully connected</i>	<i>sigma</i> value	yes	no	yes
Mostaghim and Teich (2003b)	<i>fully connected</i>	<i>sigma</i> value	yes	no	yes
Mostaghim and Teich (2004)	<i>fully connected</i>	<i>sigma</i> value	yes	no	yes
Li (2003)	<i>fully connected</i>	niche count; density estimator	yes	yes (1.0 \rightarrow 0.4)	yes
Hu et al. (2003)	<i>ring</i>	single-objective	yes	yes <i>rnd</i> (0.5, 1.0)	no
Srinivasan and Seow (2003)	<i>fully connected</i>	niche count & dominance	no	no	yes
Bartz-Beielstein et al. (2003)	<i>fully connected</i>	density of solutions; success	yes	no	no
Raquel and Naval (2005)	<i>fully connected</i>	density estimator	yes	no	yes
Reyes and Coello (2005)	<i>fully connected</i>	density estimator	yes	yes <i>rnd</i> (0.1, 0.5)	yes
Alvarez-Benitez et al. (2005)	<i>fully connected</i>	dominance	yes	no	yes
Ho et al. (2005)	<i>fully connected</i>	fitness & age	yes	yes proposed	yes
Villalobos-Arias et al. (2005)	<i>fully connected</i>	<i>stripes</i>	yes	no	yes
Salazar-Lechuga and Rowe (2005)	<i>fully connected</i>	niche count	yes	no	no
Zhao and Cao (2005)	<i>fully connected</i>	fuzzy membership	yes	no	no
Janson and Merkle (2005)	<i>fully connected</i>	random	yes	no	no

5.5 Multi-Objective Particle Swarm Optimisation Variant

Among successful MOPSO variants is one which uses the crowding distance developed by Raquel and Naval (2005). In the next subsections we will focus on the details of the implementation of this adapted approach.

5.5.1 Particle Swarm Optimisation with Crowding Distance Implementation

The original version of the approach developed by Raquel and Naval (2005), called Particle Swarm Optimisation with Crowding Distance (MOPSO), which extends the algorithm of the SOPSO to handle MOO problems. In this approach the crowding distance technique is employed for selecting the leader and also for eliminating particles from the restricted external archive of non-dominated solutions to preserve the diversity

of the swarm. The approach uses the mutation operator proposed in Parsopoulos and Vrahatis (2002a) and the constraint-handling technique from the NSGA-II (Zitzler et al., 2000). These mechanisms are detailed below. In the implementation we have added the use of the linear decrease inertia weight from 0.8 to 0.4 as well as cognition and social weighing factors c_1 and c_2 which provided a better performance than the original version in the performed reservoir modelling applications.

5.5.2 Leader Selection

The selection of the global guide, the leader, of the particle swarm is fundamental in MOPSO algorithm. The leader choice influences diversity of the non-dominated solutions within the swarm and the convergence rate. The approach uses an external archive in order to promote the swarm attraction towards the less crowded (sparse) areas of the search space. The archive is sorted based on the decreasing crowding distance computed for non-dominated solutions so as to facilitate the swarm particles to be generated in the less populated regions of the non-dominated solutions in the objective space. The leader is then chosen randomly from the top of this list.

5.5.3 Random Replacement

A random replacement mechanism is employed when the archive reaches its full capacity. Overcrowded regions in the front are deleted and replaced with randomly generated particles. The crowding distance is computed in order to select the particle which is to be eliminated by choosing particles with the smallest crowding distance for replacement. New particles will then select a solution as its leader from the least populated regions of the front.

5.5.4 Mutation

A mutation operator that acts on the entire swarm initially is employed in order to facilitate the exploration ability of the swarm and avoid premature convergence to local Pareto fronts existing in several multi-objective optimisation problems. Then over time fewer particles are mutated to allow rapid convergence. This version was an adaption of an earlier MOPSO considered by Coello et al. (2004).

5.5.5 Crowding Distance Computation

The crowding distance value of a non-dominated solution provides an estimate of the density of solutions surrounding that solution (Deb et al., 2000). Figure 5.7 illustrates the calculation of the crowding distance of point i which is an estimate of the size of the largest cuboid enclosing i without including any other point. For a specific point, i , the two points on either side of point i are selected. The crowding distance is the average of the distances between each of these points and i and are computed with respect to each objective. Areas with a larger crowding distance are particularly favoured for the selection of a local leader in particular (Engelbrecht, 2005).

The procedure for computing the crowding distance is as follows. For each objective function, the set of particles is ordered in descending order of objective function values at first. A certain particle's crowding distance is the average distance of its two neighbouring particles. The solution particles located at the boundaries which have the smallest and largest objective function values are given infinite crowding distance values with the intention of always being selected. The final crowding distance value of a particle is obtained by summing over all the particle's crowding distance values for each objective function. This is demonstrated in the pseudo-code presented in Algorithm 5.3 (Raquel and Naval, 2005).

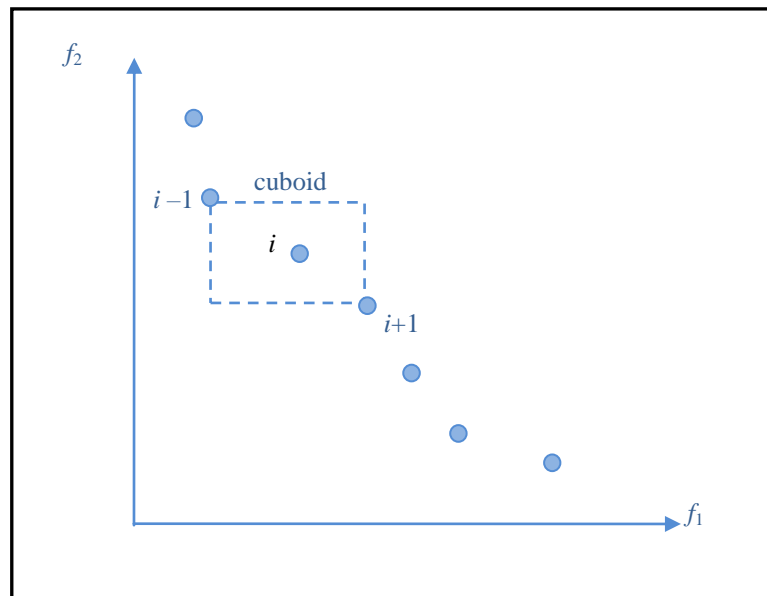


Figure 5.7: Crowding distance computation for two objective functions. Particles with a larger value are preferred

Algorithm 5.3 Crowding Distance Algorithm

1. Find the number of non-dominated solutions in the external archive S
 - a) $n = |S|$
2. Initialise distance
 - a) For $i = 0$ to $n + \text{swarm size}$
 $S_i.\text{distance} = 0$
3. Calculate the crowding distance for each particle
 - a) For each objective m from k
 Sort using each objective value $S = \text{sort}(S, m)$
 For $i = 1$ to $(n - 1)$
 $S_i.\text{distance} = S_i.\text{distance} + (S_{i+1}.m - S_{i-1}.m)$ # The '.' means in respect to m
 - b) Set the maximum distance to the boundary points so that they are always selected as
 $S_0.\text{distance} = S_n.\text{distance} = \text{maximum distance}$

5.5.6 Constraint Handling in MOPSO

MOPSO adapted the constraint handling technique employed by NSGA-II because of its simplicity. Feasibility and non-dominance of solutions are the considered criteria when comparing solutions to handle particles that fly outside the feasible regions in constrained optimisation problems. A constrained-dominance concept is used to compare between two particles.

1. If both particles are feasible, the non-dominated particle is chosen.
2. If one of the particles is feasible, that particle is chosen.
3. If both particles are infeasible, the particle with the least constraints violated is chosen.

Thus, the particle which dominates the other particle when assessing two feasible particles is considered a better solution and if both particles are infeasible, the particle with the least constraints violated is the better solution.

5.5.7 The Time Complexity of MOPSO

The contribution in the computational complexity of the approach is originated from the objective function computation, crowding distance computation, and the non-dominated comparisons of the swarm's particles in the predefined restricted archive. Suppose the swarm has N particles and the problem has k objective functions, then the computations of sorting the objective functions have computational complexity of $O(kN)$. The main expensive component in crowding distance computation for leader selection operation is based on ranking the solutions in each objective function. This can be evaluated as following: assuming the archive has h elements, ordering the elements in the archive has a computational complexity of $O(kh \log h)$ (with merge or heap sorting). With the equal number of solutions, N , in the swarm and the archive, ordering comparisons of the non-dominated solutions has the computational complexity

$O(kN^2)$ for inserting a new particle from the swarm. Thus, each update in MOPSO will have a computational complexity $O(kN^2)$. The overall computational complexity is therefore of $O(kN^2)$. The complete pseudo-code of MOPSO adapted from Raquel and Naval (2005) is provided in Algorithm 5.4.

Algorithm 5.4 MOPSO Algorithm

1. Initialise swarm
 - a. For $i = 1$ to N (N is the swarm size)
 - i. Initialise the position of particle i denoted x_i^k randomly in the swarm of particles X
 - ii. Initialise the velocity of particle i denoted $v_i^k = 0$
 - iii. Evaluate fitness of x_i^k
 - iv. Initialise the personal best of each particle as $pbest_i^k = x_i^k$
 - b. End For
2. Initialise leaders in an external archive S that stores non-dominated solutions originated from X
3. Quality Measure (leaders)
4. $k = 0$ – initialise the iteration counter
5. While $k < T_{max}$ # (T_{max} is the maximum number of iterations)
 - a. Compute the crowding distance values of each non-dominated solution in the archive S
 - b. Sort the non-dominated solutions in S in descending crowding distance values
 - c. For $i = 1$ to N
 - i. Select the global best leader randomly for x_i^k from a specified top portion (e.g. top 10%) of the sorted archive S and store its position to $gbest^k$.
 - ii. Update the velocity:

$$v_i^{k+1} = \omega v_i^k + c_1 \times rand_1 \times (pbest_i^k - x_i^k) + c_2 \times rand_2 \times (S(gbest^k) - x_i^k)$$

Eq. (5.5)

 (ω is the inertia weight, c_1 and c_2 are the cognition and social components respectively, $rand_1$ and $rand_2$ are selected at random from the range $[0,1]$) ($pbest_i^k$ is the best position that the particle i has seen) ($S(gbest^k)$ is the leader)
 - iii. Update the position:

$$x_i^{k+1} = x_i^k + v_i^{k+1}$$

Eq. (5.6)
 - iv. If x_i^{k+1} flies outside the boundaries, then it is repositioned using one of the strategies discussed in Section 4.1.3.3.
 - v. If ($t < (T_{max} \times p_{mut})$), then perform mutation on x_i^k where p_{mut} is the probability of mutation
 - vi. Evaluate fitness of x_i^k
 - d. End For
 - e. Insert all new non-dominated solution into S if they are not dominated by any of the stored solutions. All dominated solutions in the archive by the new solution are removed from the archive. If the archive is full, the solution to be replaced is determined by the following steps:
 - i. Compute the crowding distance values of each non-dominated solution in the archive S
 - ii. Sort the non-dominated solutions in S in descending crowding distance values
 - iii. Randomly select a particle from a specified bottom portion (e.g. lower 10%) which comprise the most crowded particles in the archive then replace it with the new solution
 - f. Update the personal best solution of each particle in X . If the current $pbests$ dominates the position in memory, the particle best position is updated using

$$pbest_i^k = x_i^k$$
 - g. Quality Measure(leaders)
 - h. $k++$
6. End While
7. Report results in the external archive

5.6 Numerical Experiments

The performance of the MOPSO technique is evaluated using four test functions designed for testing multi-objective optimisation problems. The MOPSO results are compared with the SOPSO. For all the test functions and both SOPSO and MOPSO approaches, the setup used is as following:

- Number of particles in the swarm = 20
- Maximum number of generations = 200
- For MOPSO, the capacity of the archive = 500
- Total number of function evaluations = 4000

5.6.1 Numerical Example Test 1

The first test function proposed by Kita (Kita et al., 1996) is a multi-objective maximisation function which has three constraints. It is an example of two objective functions, two parameters, and concave Pareto optimal front shape. The maximisation problem function is given in Eq. (5.7).

Maximise

$$f(x) = (f_1(x, y), f_2(x, y))$$

where

$$f_1(x, y) = -x^2 + y$$

$$f_2(x, y) = \frac{1}{2}x + y + 1$$

Eq. (5.7)

subject to

$$0 \geq \frac{1}{6}x + y - \frac{13}{2}; 0 \geq \frac{1}{2}x + y - \frac{15}{2}; 0 \geq 5x + y - 30$$

with a range of $0 \leq x, y \leq 7$.

The result for applying MOPSO is shown in Figure 5.8(b) where the Pareto front is highlighted in red. The SOPSO result obtained by aggregating the two objectives into a single one is shown in Figure 5.8(a) where only part of the Pareto front is produced. However, we note that the result is obtained without the three constraints since our implementation does not handle it.

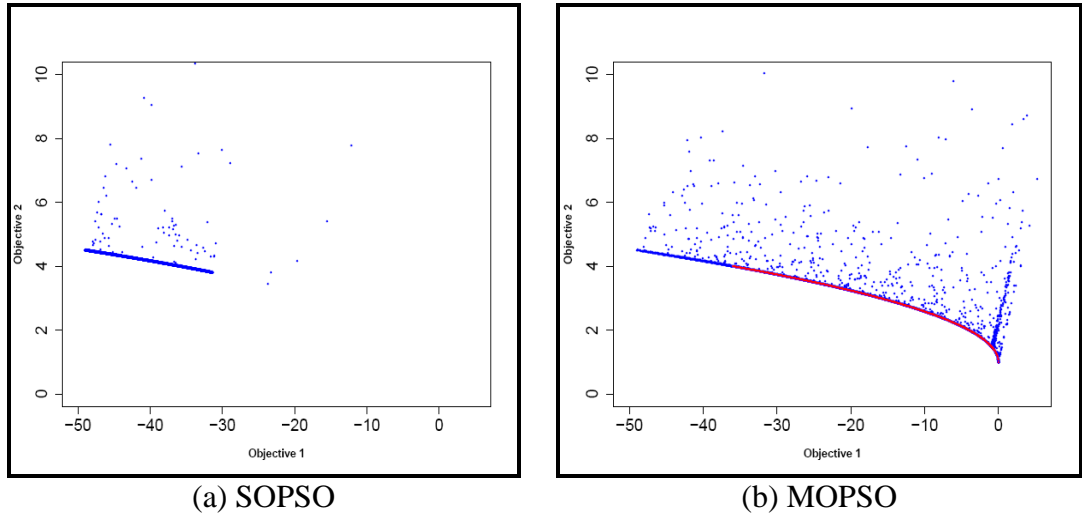


Figure 5.8: Result of example test 1

5.6.2 Numerical Example Test 2

The second test function has been introduced by Kursawe (1991) (Li, 2003) which has three nonconvex disconnected Pareto curves. Its solution mapping in the objective space is very convoluted. It is an example of two objective functions and three parameters. The minimisation problem function is given in Eq. (5.8) where $k = 3$.

$$\begin{aligned} \text{Minimise} \quad & f_1(\vec{x}) = \sum_{i=1}^{k-1} \left(-10 \exp\left(-0.2\sqrt{x_i^2 + x_{i+1}^2}\right) \right) \\ \text{Minimise} \quad & f_2(\vec{x}) = \sum_{i=1}^k \left(|x_i|^{0.8} + 5 \sin(x_i^3) \right) \\ \text{where} \quad & -5 \leq x_1, x_2, x_3 \leq 5. \end{aligned} \quad \text{Eq. (5.8)}$$

The result for applying MOPSO is shown in Figure 5.9(b) where the disconnected Pareto front is highlighted in red. The SOPSO result obtained by aggregating the two objectives into a single one is shown in Figure 5.9(a). Here, we see that while MOPSO focuses on balancing the two objectives, SOPSO obtained more points in a corner near -15 . Yet, both approaches perform well for this example.

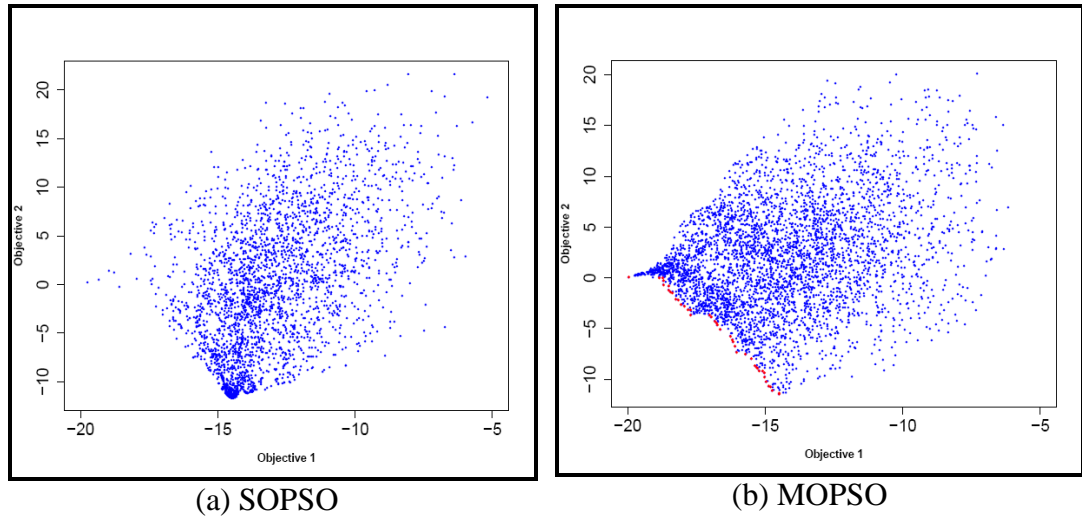


Figure 5.9: Result of example test 2

5.6.3 Numerical Example Test 3

The third test function used has been introduced by Deb (Deb, 1999) and has a bimodal function $g(x_2)$ which has local and global minimum values. It is an example of two objective functions, two parameters, and convex Pareto optimal front shape. The minimisation problem function is given in Eq. (5.9).

$$\begin{aligned}
 &\text{Minimise} && f_1(x, y) = x \\
 &\text{Minimise} && f_2(x, y) = \frac{g(y)}{x} \\
 &\text{where} && g(y) = 2.0 - \exp\left\{-\left(\frac{y-0.2}{0.004}\right)^2\right\} - 0.8 \exp\left\{-\left(\frac{y-0.6}{0.04}\right)^2\right\} \\
 &\text{and} && 0.1 \leq x, y \leq 1.0.
 \end{aligned} \tag{5.9}$$

The result for applying MOPSO is shown in Figure 5.10(b) where the convex estimated Pareto front is clearly covered in MOPSO than in SOPSO as depicted in Figure 5.10(a). The Pareto front in MOPSO is closer to the lower-left corner than in SOPSO indicating that the models with the lowest misfit obtained by the sum of the two objectives in this case is smaller in the MOPSO case.

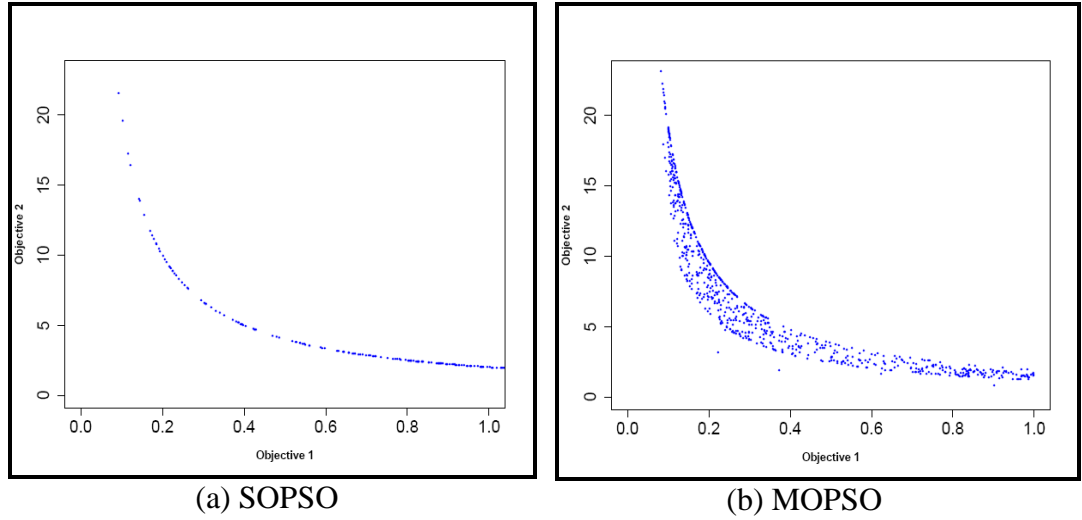


Figure 5.10: Result of example test 3

5.6.4 Numerical Example Test 4

The fourth test function used to test the performance of MOPSO is from DTLZ (Deb-Thiele-Laumanns-Zitzler) test suites. The numerical test function is DTLZ6 that has been introduced by Deb (Deb et al., 2001; 2002; 2005). It is an example of three objective functions and 22 parameters. The minimisation problem function is given in Eq. (5.10) for the general case with M objective functions.

$$\begin{aligned}
 & \text{Minimise} && f_1(x_1) = x_1 \\
 & && \vdots \\
 & \text{Minimise} && f_{M-1}(x_{M-1}) = x_{M-1} \\
 & \text{Minimise} && f_M(x) = (1 + g(x_M))h(f_1, f_2, \dots, f_{M-1}, g), \\
 & \text{where} && g(x_M) = 1 + \frac{9}{|x_M|} \sum_{x_i \in X_M} x_i, \\
 & && h = M - \sum_{i=1}^{M-1} \left[\frac{f_i}{1+g} (1 + \sin(3\pi f_i)) \right], \\
 & \text{subject to} && 0 \leq x_i \leq 1; \text{ for } i = 1, 2, \dots, d.
 \end{aligned} \tag{5.10}$$

The minimisation problem for the three objective functions is given in Eq. (5.11). DTLZ6 has $2^2 = 4$ disconnected Pareto-optimal regions.

Minimise $\{f_1(x), f_2(x), f_3(x)\}$
 subject to $0 \leq x_i \leq 1$ for $i = 1, 2, \dots, 22$.

where the objective functions defined as

$$\begin{aligned} f_1(x_1) &= x_1, \\ f_2(x_2) &= x_2, \\ f_3(x) &= (1 + g(x_3))h(f_1, f_2, g), \end{aligned} \tag{5.11}$$

and the functionals are

$$\begin{aligned} g(x_3) &= 1 + \frac{9}{|x_3|} \sum_{x_i \in x_3} x_i, \\ h &= 3 - \sum_{i=1}^2 \left[\frac{f_i}{1+g} (1 + \sin(3\pi f_i)) \right]. \end{aligned}$$

Similarly, the result for applying MOPSO is shown in Figure 5.11(b) while SOPSO one is depicted in Figure 5.11(a). The Pareto front obtained with MOPSO is shown as red colour points in Figure 5.11(c), and the pink coloured points shows all the models sampled. It is clearly shown that MOPSO tried to find all diverse models that balanced between the three objectives focused on some boundaries that have lower misfit values (since it is the sum). The SOPSO, on the other hand, focused on the area in which the models have the lowest misfit values that MOPSO found as one of the Pareto-optimal regions. Note that in this case the models in other areas found by MOPSO were good fitting models, but they were not the models with the lowest misfit values. Since the target in SOPSO is the finding of the optimum, the optimisation is concentrating on the lowest misfit part while in the MOPSO the optimum is defined by all the points in the Pareto set, which are all of the non-dominated solutions that balance between the three objectives.

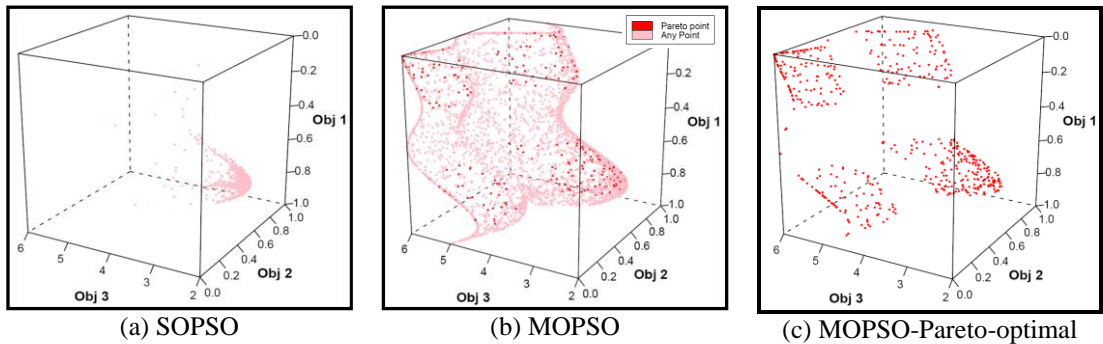


Figure 5.11: Result of example test 4

5.7 Use of Multi-Objective Particle Swarm Optimisation in Calibration of Reservoir Models

In the previous section we have shown the application of MOPSO in testing multi-objective optimisation problems. In this section, we report the results of generating multiple history matched models with MOPSO on two reservoir models: IC Fault Model (Tavassoli et al., 2004), and PUNQ-S3 (Floris et al., 2001). The posterior probabilities for the models obtained are calculated using the NAB routine (Sambridge, 1999b).

5.8 Field Application Test 1: IC Fault Model

The IC Fault Model with the 3 unknown parameters: k_{high} , k_{low} , and $throw$ in Table 4.8, is used with the uniform priors indicated.

5.8.1 Algorithm Setup Specifications

A set of 50 initial starting points obtained with Latin Hypercube Sampling (LHS) for SOPSO and MOPSO (refer to Section 4.3.2.3.1) was used. We ran 40 iterations for both PSO and MOPSO algorithms leading to a total of 2000 reservoir model simulations. For single objective particle swarm (SOPSO) we used absorbing boundary strategy where the normal component of a particle's velocity is zeroed. The setup for both SOPSO and MOPSO is shown in Table 5.2.

Table 5.2: SOPSO and MOPSO algorithm setup

Algorithm	Number of particles	Generations	Total number of simulations	ω	c_1	c_2	v_f	Mutation
SOPSO	50	40	2000	0.8–0.4	2.0	2.0	0.5	–
MOPSO	50	40	2000	0.8–0.4	1.0	1.0	–	0.5

5.8.1.1 Objective Function Definition

For MOPSO we consider the least square weighted residuals of oil rate at the production well as the first objective and the least square weighted residuals of water rate as the second objective. The aggregated global objective function to be minimised was defined as the sum of the two objective functions defined in Eq. (5.12) equivalent to Eq. (4.7) for the SOPSO approach.

$$M = Obj_1 + Obj_2 \quad \text{Eq. (5.12)}$$

5.8.2 History Matching Results

The history matching summary results for IC fault model for SOPSO and MOPSO are shown in Table 5.3 which shows the best misfit obtained for two runs of each approach in the entire algorithm run. In both runs, the best misfit obtained was for MOPSO, yet they are close. The number of NAB models in the first case is larger in SOPSO while in the second case it is larger in MOPSO.

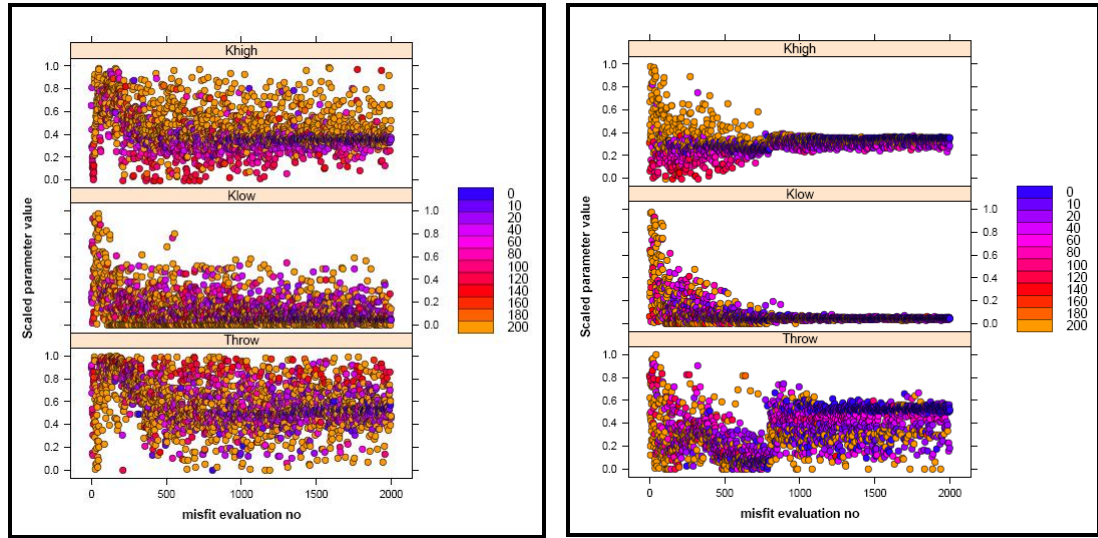
Table 5.3: Results of the performance of SOPSO and MOPSO with a population of 50 particles

Algorithm	Generational minimum objective function (last iteration)	Minimum objective function	Number of models below objective function ($M \leq 25$)	Number of NAB models	Size of Pareto = Pareto Front Set
SOPSO1	0.19	0.16	343	373	–
MOPSO1	0.47	0.11	1299	235	179
SOPSO2	0.32	0.31	150	61	–
MOPSO2	0.30	0.30	733	248	18

Sampling history plots for each parameter for the first run are shown in Figure 5.12 where the models are colour coded according to the misfit. While SOPSO was exploring more parts in space, MOPSO found a larger region of good quality models along the throw axis. The number of models below the threshold of 25 (indicating the good quality models) is larger in the MOPSO case as shown in Table 5.3 and depicted in the 3D representation in Figure 5.13 where these models are shown as red points. Thus, the multi-objective particle swarm approach have explored the parameter space and obtained a well distributed set of good fitting reservoir models more representative to the models with low misfits in the ribbon-like structure of the benchmark database case resulting in more divergent prediction qualities.

Stochastic sampling evolutionary algorithms sample a large portion of the search space avoiding local minima, but unfortunately their convergence to the global optimum is slow. On the other hand, the employment of a multi-objective optimisation scheme contributed more knowledge to the optimisation when compared to the single-objective scheme. A set of non-dominated solutions populating the repository were reported when the optimisation runs finished, the Pareto optimal front as shown in Figure 5.14 for two single runs of MOPSO with different seeds. In the single objective scheme, on

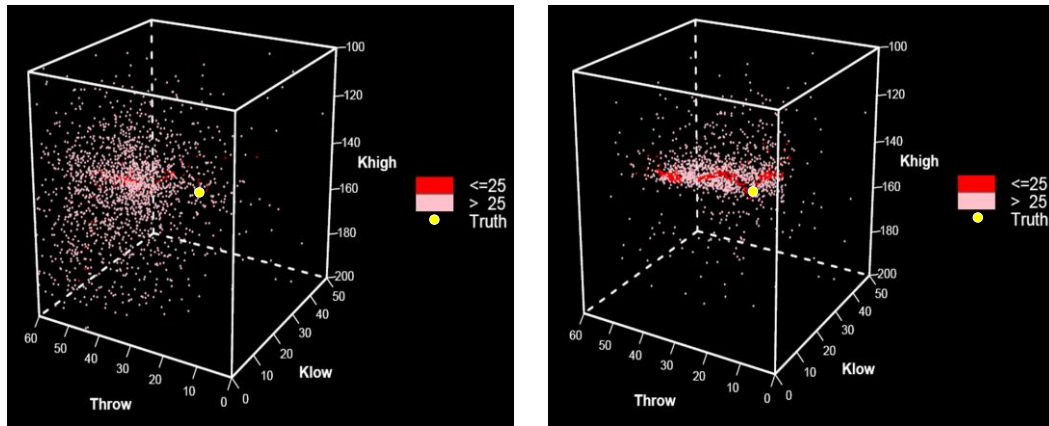
the other hand, only one global solution is targeted. The optimal Pareto set may also involve a number of solutions equivalent to different calibrations of the reservoir model.



(a) Sampling History of SOPSO

(b) Sampling History of MOPSO

Figure 5.12: Sampling history of SOPSO and MOPSO



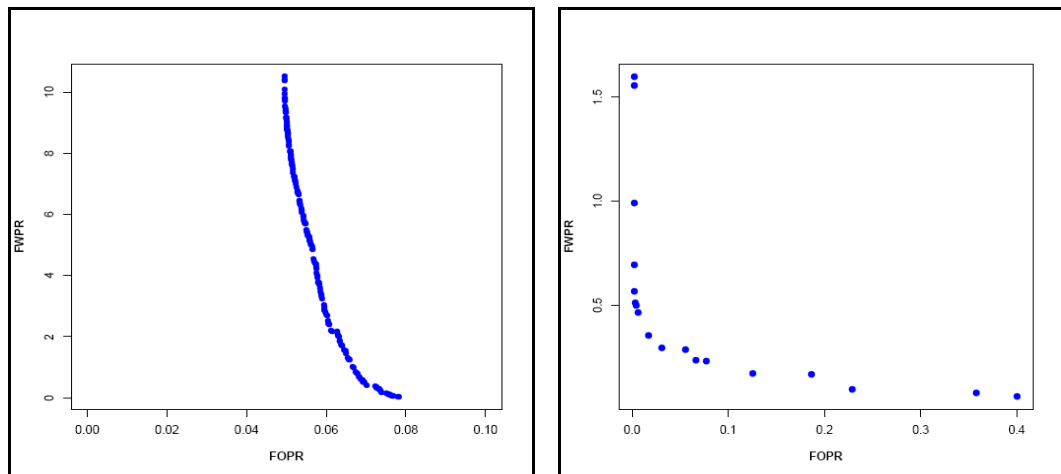
(a) SOPSO

(b) MOPSO

Figure 5.13: SOPSO vs. MOPSO Sampling

A summary of the best ten non-dominated solutions is given in Table 5.4 which shows the two objective values. The ten solutions correspond to the best objective functions ranked out of the total number in the archive. The models in the full archive are shown in the Pareto front plot obtained with the two runs (MOPSO1 and MOPSO2 in Figure 5.14). It is noticeable that while MOPSO2 has focused on the parts towards south-west corner, MOPSO1 has focused in finding more diverse models with higher FWPR in the Pareto front. One reason that may explain this performance is that in the IC Fault model case the water production mismatch is the main contributor to the global misfit function. As MOPSO tries to balance the objectives with more diverse solutions, the crowding

distance does that role by giving the boundary points infinite crowding distances so that they are always selected (see Section 5.5.5). Thus, we can relax the behaviour in Figure 5.14(a) by making the choice to stay in the archive controlled by some other factor such as ranking based on the global best when the models obtain a misfit value larger than predefined thresholds for each objective when a reasonable level of diversity is reached in such cases. The ranking idea could be coupled and that would be interesting to investigate.



(a) Pareto front MOPSO (179 models) (b) Pareto front MOPSO2 (18 models)

Figure 5.14: Pareto front for MOPSO1 and MOPSO2

Table 5.4: Summary of best 10 non-dominated elements (ranked by global misfit out of 179) stored in the archive and their respective objectives for MOPSO1

Model ID	Objective 1 (FOPR)	Objective 2 (FWPR)	Global objective function
1	0.078021	0.034099	0.112120
2	0.078176	0.033957	0.112133
3	0.076751	0.044862	0.121613
4	0.076716	0.048693	0.125409
5	0.076455	0.056497	0.132952
6	0.076091	0.072707	0.148798
7	0.075951	0.074662	0.150613
8	0.075835	0.115798	0.191633
9	0.075609	0.11751	0.193119
10	0.075518	0.125434	0.200952

For checking the robustness and sensitivity of the both single objective SOPSO and multi-objective MOPSO schemes, we performed 20 runs for each scheme starting from different initial seed values. Two plots that can be used to evaluate the performance and assessment of the global optimisers' efficiency are the global best and generational

minimum. The box plot for the two efficiency measures can aggregate the information gained in the 20 runs.

Using the global best (minimum so far) box plots for 20 runs is not necessarily a good indicator for the comparison between the SOPSO and MOPSO schemes, since in MOPSO the leader for each particle is chosen randomly from the top 10% of the non-dominated solutions stored in the external archive that has been sorted in descending crowding distance. In SOPSO, on the other hand, the leader used is the typical global best of the entire swarm. Thus, only the generational minimum per generation for 20 runs is used here. We show the overall performance of SOPSO and MOPSO algorithms during the course of the optimisation with box plots demonstrated in Figures 5.15 and 5.16. This is achieved by calculating the generational minimum in 20 runs starting from the same 50 starting points for both schemes. In the box plot the line in the box refers to the median, the lower and upper bound points to the 25% and 75% quartiles respectively and the circle sign refers to the outliers. Box plots are frequently used to represent meaningful statistical results.

The box plots demonstrate the benefit of using the multi-objective approach by looking at the boundary points of the maximum and the minimum value of the box plots where faster convergence of MOPSO has been observed. A better anticipation of the minimum objective function in the multi-objective approach is apparent in the first explorative runs. In particular, the objective function accomplished at generation 12 was remarkably better in MOPSO than in the single-objective procedure. The study of the performance of the MOPSO reflected that the objective function decreased sharply at the early stages of the evolution while it flattened later on. The difference between the 25% and 75% quartiles indicates how the values of the misfit vary along the corresponding generation. It is noticeable how the multi-objective schemes has obtained smaller ranges compared to the single objective implying that history matches produced with the multi-objective approach are less sensitive to difference in seed value and provide more good fitting models. The MOPSO has achieved better than the SOPSO and this result is robust while using the 20 seeds.

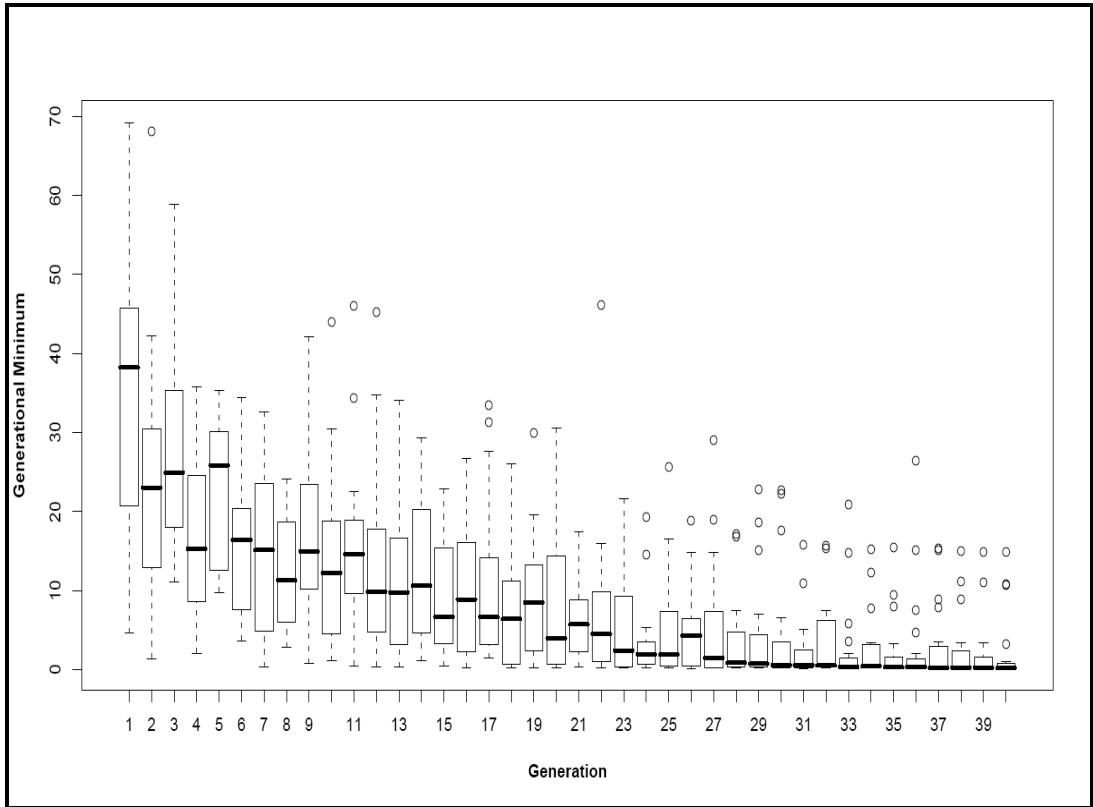


Figure 5.15: Convergence speed of SOPSO approach with a swarm of 50 particles for 40 generations using 20 seeds – median of M around 0 at iteration 28

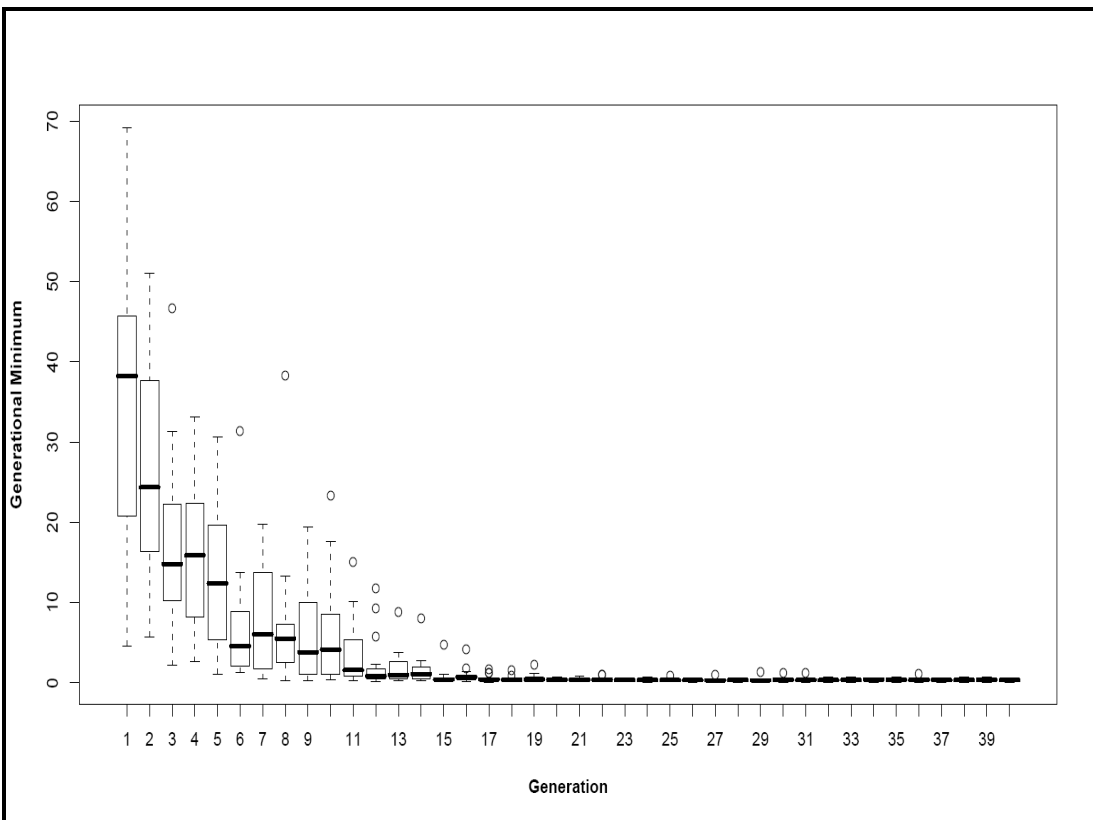


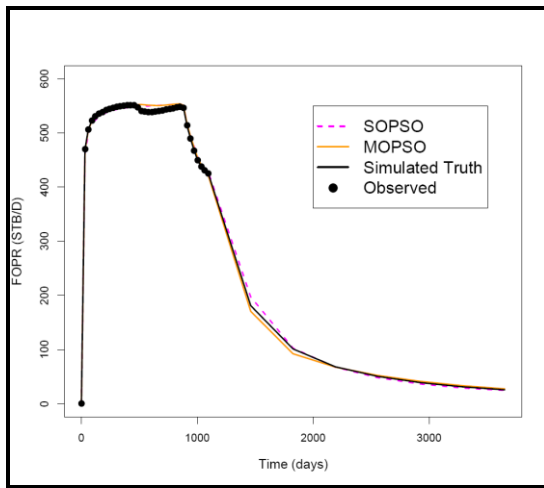
Figure 5.16: Convergence speed of MOPSO approach with a swarm of 50 particles for 40 generations using 20 seeds – median of M around 0 at iteration 12

5.8.3 Uncertainty Quantification

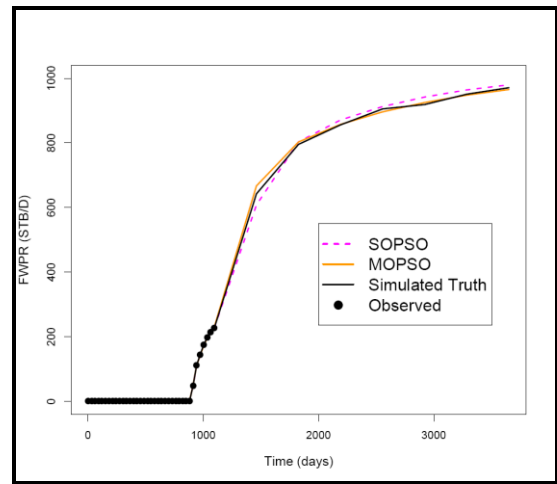
The best history matched models from the two schemes were run forward for the prediction period for a single run of each scheme. Table 5.5 shows the result of the forecast misfit for the best model. Both schemes obtain a good forecast for their runs. The corresponding history matched figures are shown in Figure 5.17 for oil rate, water rate, total recovery, and water injection rate.

Table 5.5: Best model history and forecast misfits

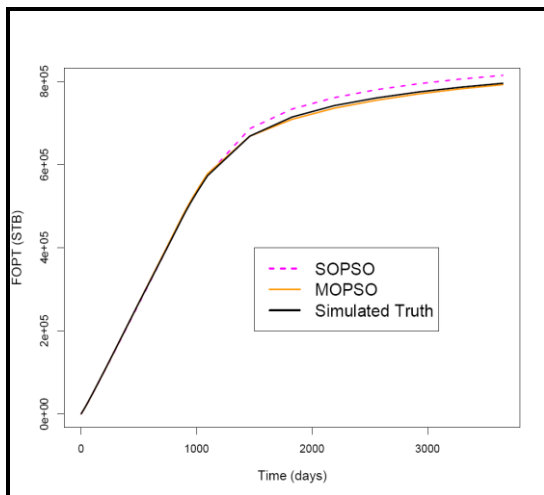
Algorithm	History misfit	Forecast misfit
SOPSO	0.16	2.04
MOPSO	0.11	2.10



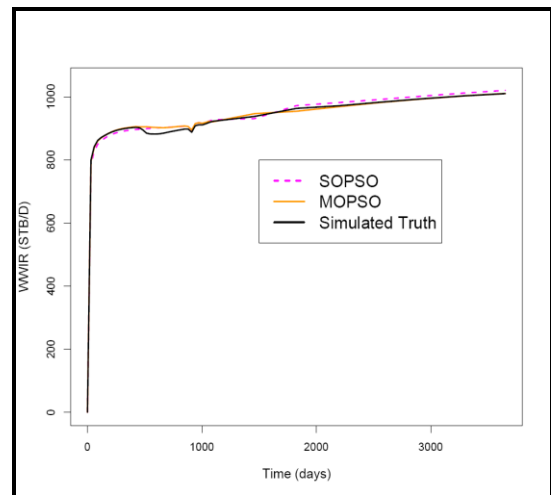
(a) Oil rate



(b) Water rate



(c) Total recovery



(d) Water injection rate

Figure 5.17: Best history matches for SOPSO and MOPSO approaches

The Bayesian credible intervals for the two schemes compared to the database are shown for a single run of each scheme in Figure 5.18 indicating 30% error difference in water rate prediction of SOPSO in Figure 5.18(e) compared to the database result in

Figure 5.18(d) while obtaining 5% error difference with MOPSO in Figure 5.18(f) compared to the database indicating an improvement in water prediction by 25%. The error difference with the SOPSO in total oil prediction in Figure 5.18(h) is 20% from the database result in Figure 5.18(g) while obtaining a 12% error difference with MOPSO in Figure 5.18(i) from the database result indicating an improvement in total oil prediction by 8%.

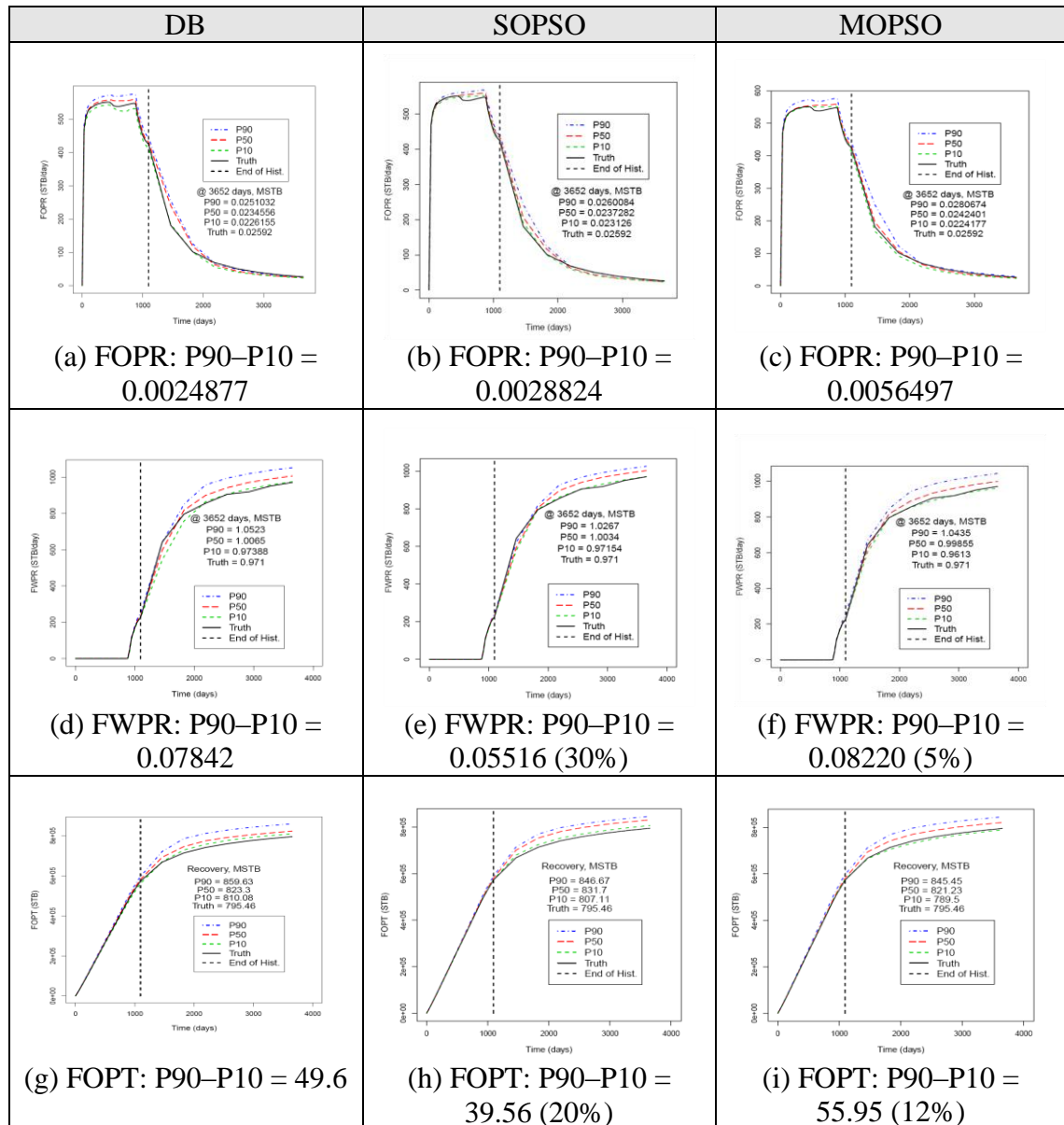


Figure 5.18: Comparison of Bayesian credible intervals for a single run of each scheme – database vs. SOPSO and MOPSO

Finally, the total recovery at the last step is depicted in Figure 5.19 where both have obtained good results relatively close to the database result. MOPSO has obtained a wider estimate of uncertainty that captured the truth, while SOPSO has produced narrower uncertainty estimates.

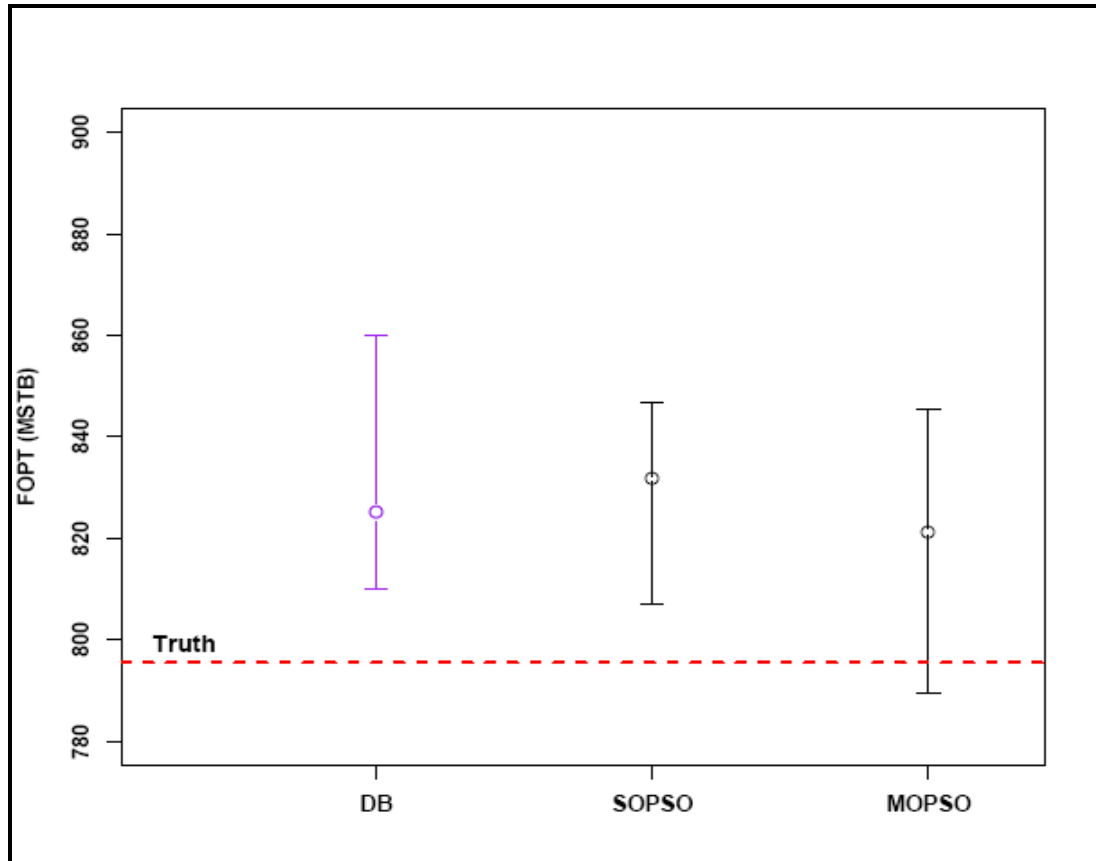


Figure 5.19: Prediction uncertainties derived from the database, SOPSO, and MOPSO

5.9 Field Application Test 2: PUNQ-S3 Model

The particle swarm algorithm two schemes has also been applied to a complex three-phase synthetic petroleum field, prepared as part of the PUNQ (production forecasting with uncertainty quantification) project sponsored by the European Community, which is a small-size industrial reservoir model drawn from a reservoir engineering study on a real field carried out by Elf Exploration & Production (Bos, 2000). Ten partners from universities, research institutes, and industry collaborated on investigating uncertainty quantification methods in the PUNQ project for oil production forecasting (Floris et al., 2001).

The PUNQ-S3 model has an oil reservoir with a gas cap and five geological layers. The reservoir is surrounded to the east and south by a fault, and to the north and west by

quite a strong aquifer maintaining the pressure in the field. The small gas cap is positioned in the centre of the dome-shaped geological structure (Barker et al., 2001). The reservoir was produced through six wells positioned around the gas-oil contact. The model dimension is $19 \times 28 \times 5$ gridblocks (including 1761 active ones). Figure 5.20 shows a representation of the PUNQ-S3 reservoir model.

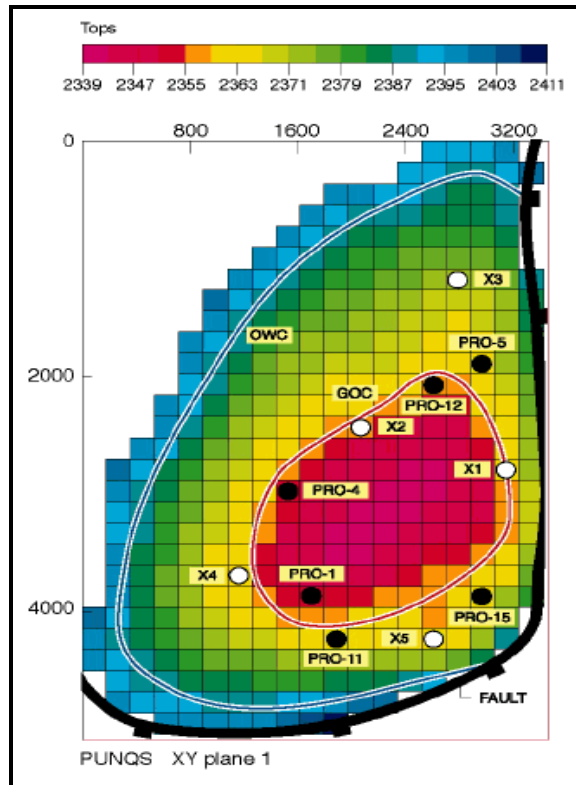


Figure 5.20: Top structure map of PUNQ-S3 (Floris et al., 2001)

5.9.1.1 PUNQ-S3 Model Uncertain Parameters

The model setup used as provided in Hajizadeh et al. (2011) includes the division of the five layers in PUNQ-S3 model into nine regions leading to 45 unknown parameters for porosity that is used in the history matching optimisation process. The horizontal and vertical permeability are then computed with the correlations (Bos, 2000) from fitting to the hard well data as provided in Eq. (5.13). Table 5.6 shows the uniform prior ranges for uncertainty parameters.

$$\begin{aligned} \ln(k_h) &= 0.77 + 9.03\phi \\ k_v &= 3.124 + 0.306k_h \end{aligned} \quad \text{Eq. (5.13)}$$

Table 5.6: Model uniform prior ranges

Layer	Porosity	Horizontal Permeability (mD)	Vertical Permeability (mD)
1	0.15 – 0.30	133 – 3013	44 – 925
2	0.05 – 0.15	16 – 133	8 – 44
3	0.15 – 0.30	133 – 3013	44 – 925
4	0.10 – 0.20	47 – 376	17 – 118
5	0.15 – 0.30	133 – 3013	44 – 925

5.9.1.2 Objective Function Definition

The global objective function – the misfit M represented by the negative log of the likelihood – is calculated using the least squares formula in Eq. (5.14) for the SOPSO approach.

$$M = \frac{1}{n_w} \sum_i \frac{1}{n_p} \sum_j \frac{1}{n_t} \sum_k \left(w_{ijk} \frac{O_{ij}^{obs}(t^k) - O_{ij}^{sim}(t^k; p)_t}{\sigma_{ijk}} \right)^2 \quad \text{Eq. (5.14)}$$

where n_w represents the number of wells, n_p represents the number of observed production data types, n_t represents the number of data points, O^{obs} refers to observed and O^{sim} refers to the simulated data for model. i stands for gas-oil-ratio, water cut, or well pressures, and k runs overall number of timesteps. w represents the weighting factor assigned to normalise the contributions to the objective function while σ refers to the standard deviation of measurement errors which both are given (PUNQ-S3 Model, 2010) to participate to the objective function calculations.

The production history in terms of bottom hole pressure (WBHP), water cuts (WWCT) and gas-oil ratios (WGOR) for each well from the six wells for the reservoir is available for an 8-year period. Gaussian noise was added to both the historical data and the well observations before the data sets with uncertainties were accessible to the partners. After 8 years of production, the recovery strategy considered for production is to keep on producing with the original six wells for another 8.5 years. The total simulation period was 16.5 years. The simulator is controlled to match oil rate.

We consider two choices for defining two objective functions for the MOPSO application as following.

1. Sum the individual least square weighted residuals of pressures measured at the wells from its simulated value for all the wells as the first objective and the individual least square weighted residuals of gas-oil-ratios and watercuts

measured at the production wells from its simulated value for all the wells as the second objective as given in Eq. (5.15).

$$\begin{aligned} Obj_1 &= [WBHP]_i; & i &= \{1, 4, 5, 11, 12, 15\} \\ Obj_2 &= [WGOR + WWCT]_i; & i &= \{1, 4, 5, 11, 12, 15\} \end{aligned} \quad \text{Eq. (5.15)}$$

2. Sum the individual least square weighted residuals of all production data types from its simulated value for wells 1, 4, and 5 as the first objective and the individual least square weighted residuals of all production data types from its simulated value for wells 11, 12, and 15 as the second objective as provided in Eq. (5.16).

$$\begin{aligned} Obj_1 &= [WBHP + WGOR + WWCT]_i; & i &= \{1, 4, 5\} \\ Obj_2 &= [WBHP + WGOR + WWCT]_i; & i &= \{11, 12, 15\} \end{aligned} \quad \text{Eq. (5.16)}$$

The aggregated global objective function to be minimised was defined as the sum of the two objective functions defined as in Eq. (5.17) equivalent to Eq. (5.14).

$$M = Obj_1 + Obj_2 \quad \text{Eq. (5.17)}$$

It is noted that using an objective function constructed by the sum of the least squared residuals of three objective functions: the gas-oil ratio, the water cut and the bottom hole pressures gives high misfit values with MOPSO ($M > 4$) for the global best. In this MOPSO variant diversity increases as number of objectives increases, in other words, it gives diversity at the cost of performance trade-off. This definition of objectives was not further pursued since it did not provide results as good as the two choices provided above.

5.9.2 Algorithm Setup Specifications

The algorithm parameter setup is shown in Table 5.7. For a better comparison of the performance of the schemes we have carried out 20 different seeds each starting from the same initial population generated with Latin Hypercube Sampling (LHS) for both SOPSO and MOPSO.

Table 5.7: Algorithms parameter setups for PUNQ-S3

Algorithm	Number of particles	Generations	Total number of simulations	ω	c_1, c_2	Top %	Mutation
SOPSO	30	100	3000	0.8–0.4	2,2	–	–
MOPSO-choice 1	30	100	3000	0.8–0.4	1,1	10	0.5
MOPSO-choice 2	30	100	3000	0.8–0.4	1,1	10	0.5

5.9.3 History Matching Results

The history matching summary results for the PUNQ-S3 model for SOPSO and MOPSO two choices are shown in Table 5.8 which shows the results of the performance with a population of 30 particles for 100 iterations. The best misfit obtained for a single run for each approach is reported in the entire algorithm run. Both SOPSO and MOPSO-choice 2 obtain comparable good quality models. The number of models below the threshold of 5 is larger in SOPSO which is double the MOPSO-choice2 while MOPSO-choice1 is smaller than the other two. Yet, the number of NAB models in MOPSO-choice2 is larger than both SOPSO and MOPSO-choice1.

Table 5.8: Results of the performance of SOPSO and MOPSO with a population of 30 particles for 100 iterations

Algorithm	Generational Minimum objective function (last iteration)	Minimum objective function	Number of models below objective function ($M < 5$)	Number of NAB Models	Size of Pareto = Pareto Front Set
SOPSO	1.578510	1.578510	2153	1068	–
MOPSO-choice1	4.267193	2.295731	100	451	15
MOPSO-choice2	1.511936	1.459408	1061	1596	14

The best history matched models from the two schemes were run forward for the prediction period. Both schemes obtained a good forecast for their runs. The corresponding history matching plots are shown in Figure 5.21 and Figure 5.22 for well 1 and 15 bottom hole pressures. Well 1 gas-oil ratio is shown in Figure 5.23 while well 11 watercut is shown in Figure 5.24.

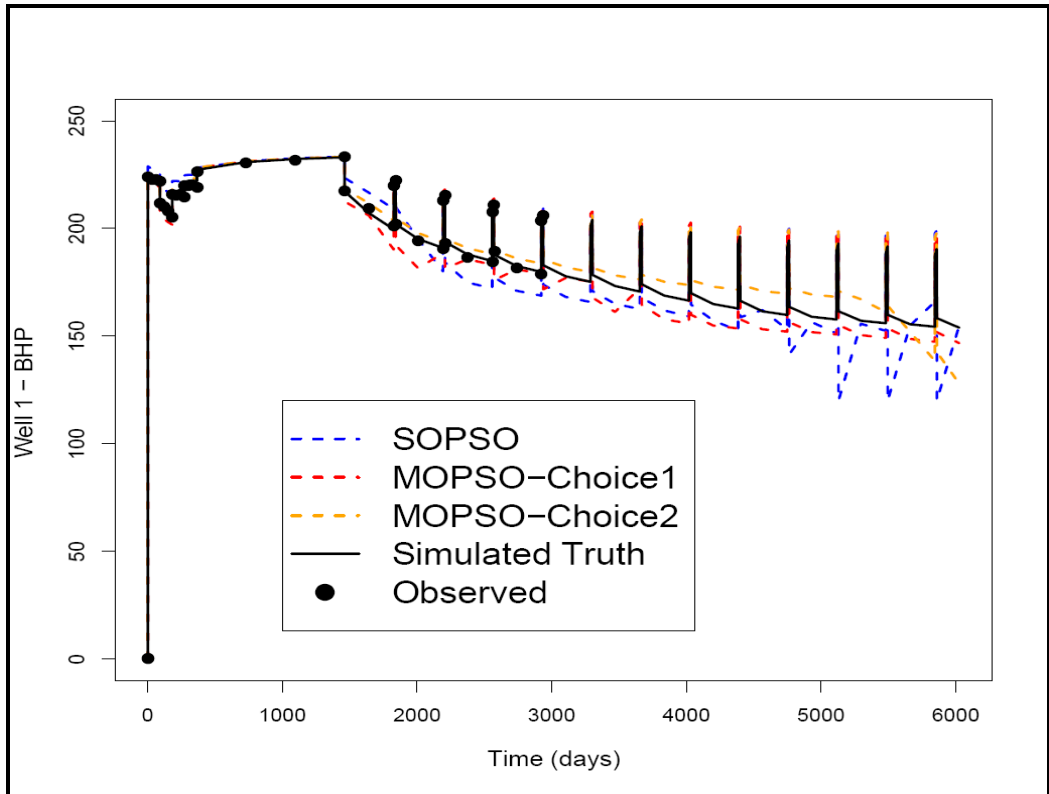


Figure 5.21: Well 1 pressure for the best history matched model

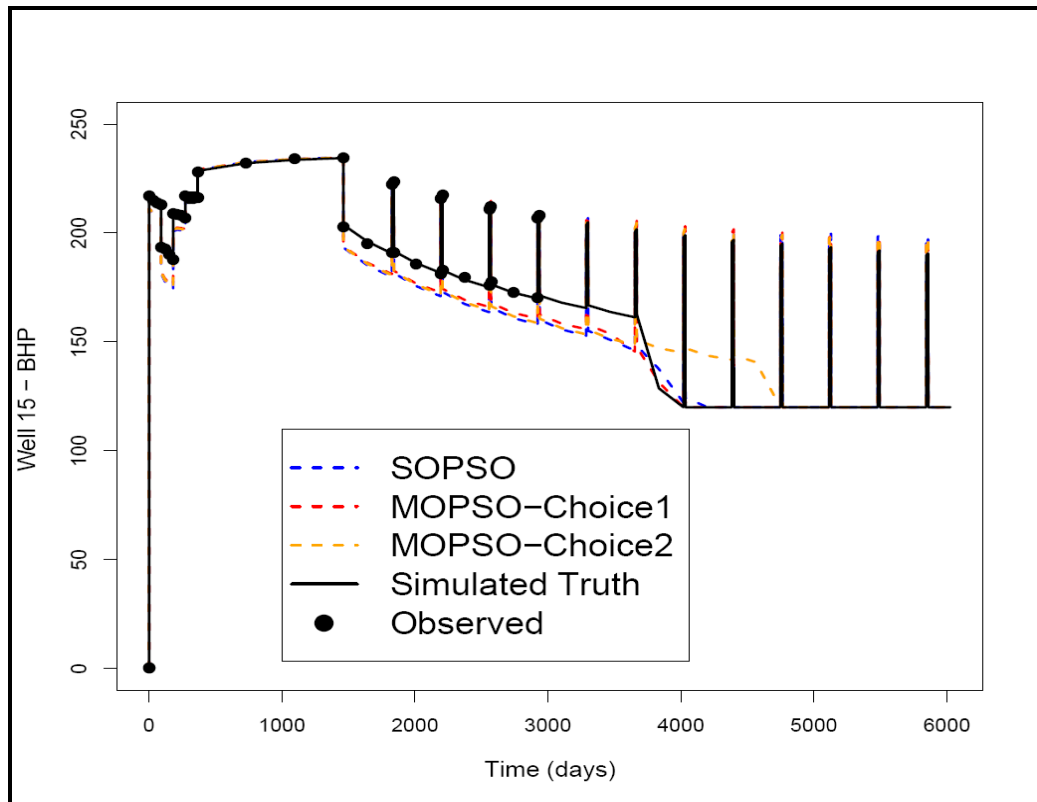


Figure 5.22: Well 15 pressure for the best history matched model

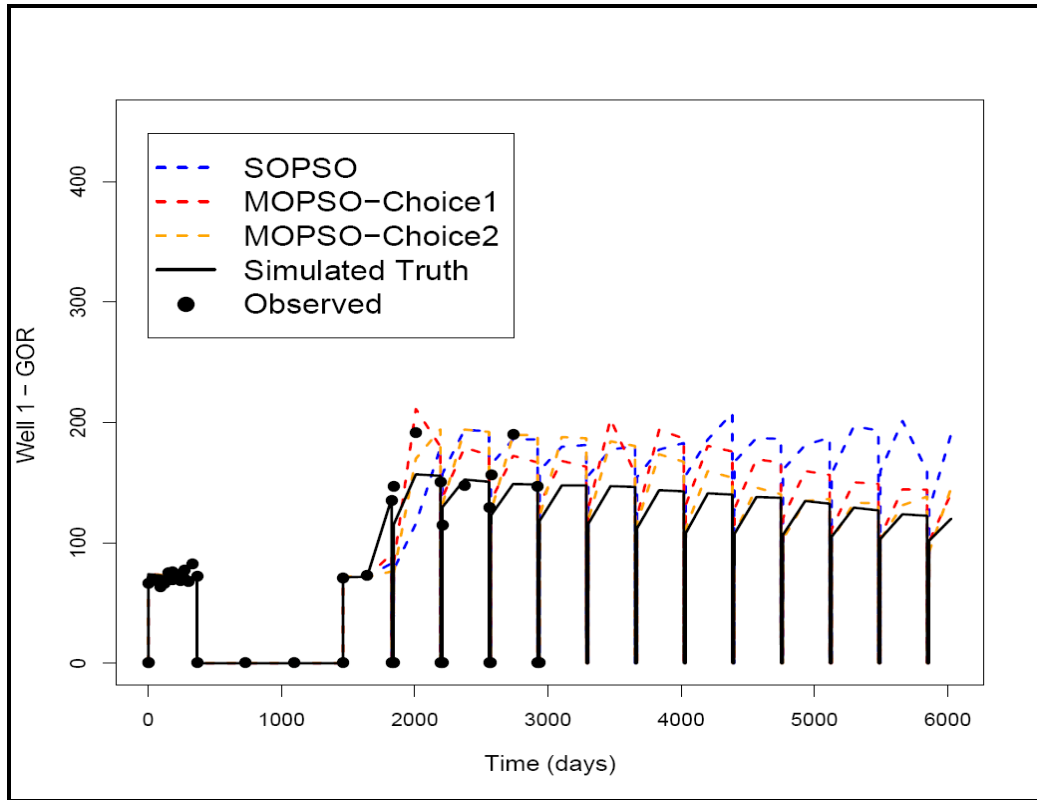


Figure 5.23: Well 1 GOR for the best history matched model

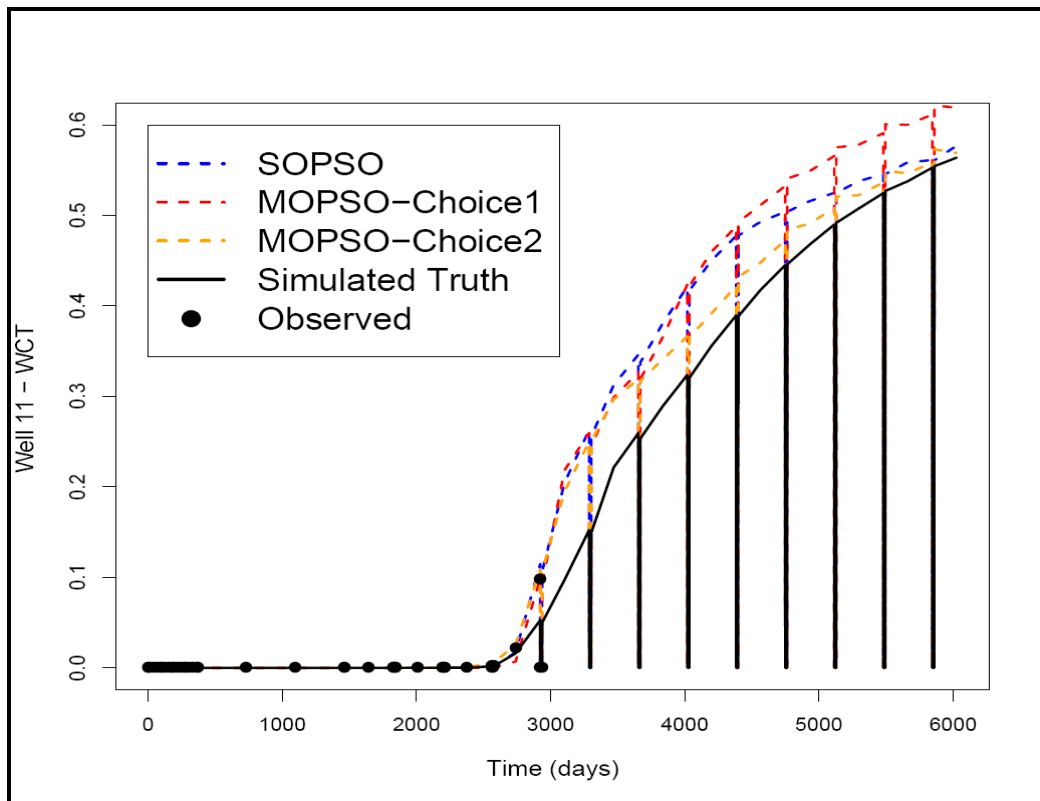


Figure 5.24: Well 11 watercut for the best history matched model

A summary of the best ten non-dominated solutions for both MOPSO choices are given in Table 5.9 and Table 5.10 which show the two objective values and the global best

misfit. While solution 1 corresponds to the minimum global objective function, MOPSO-choice1 found 14 Pareto optimal solutions. The history match of the field WBHP, particularly, was better for solutions 4, 5, 6, 7, 8, and 10, which in contrast achieved a worse match of the field WWCT and WGOR. Similarly, MOPSO-choice2 found other 14 Pareto optimal solutions other than the global one. The history match of the wells 11, 12, and 15 production data was better, particularly, for solutions 8 (plus the other 4 in the archive) which in contrast achieved a worse match of the production data for wells 1, 4, and 5. We can interpret that the conflicting multiple objectives' solutions as multiple history matches. The models in the full archive are shown in the Pareto front plot Figure 5.25 obtained with the two choices.

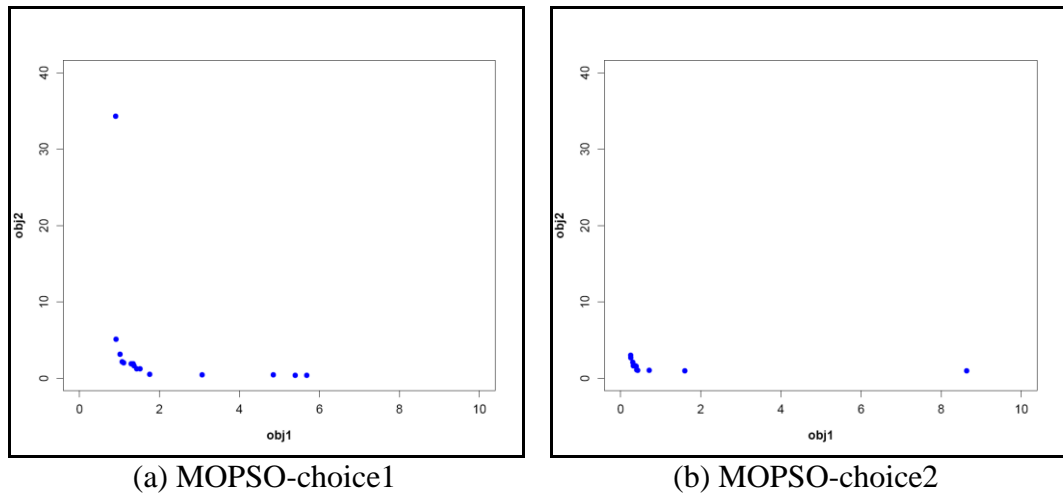


Figure 5.25: Pareto front for MOPSO

Table 5.9: Summary of best 10 non-dominated elements stored in the archive and their respective objectives (MOPSO-choice 1)

Model ID	Objective 1 (WBHP)	Objective 2 (WGOR+WWCT)	Global objective function
1	1.7603	0.5355	2.2958
2	1.4379	1.2550	2.6929
3	1.5145	1.2494	2.7639
4	1.3682	1.6192	2.9874
5	1.1071	2.0246	3.1317
6	1.2949	1.9160	3.2109
7	1.3495	1.8733	3.2228
8	1.0757	2.1666	3.2423
9	3.0687	0.4430	3.5117
10	1.0143	3.1019	4.1162

Table 5.10: Summary of best 10 non-dominated elements stored in the archive and their respective objectives (MOPSO-choice 2)

Model ID	Objective 1 for Wells {1,4,5}	Objective 2 for Wells {11,12,15}	Global objective function
1	0.425539	1.033870	1.459409
2	0.405824	1.106112	1.511936
3	0.712924	1.029459	1.742383
4	0.324444	1.639111	1.963555
5	0.390050	1.584124	1.974174
6	0.369850	1.616830	1.986680
7	0.310960	2.106286	2.417246
8	1.601236	0.980622	2.581858
9	0.258055	2.698885	2.956940
10	0.252916	2.982797	3.235713

Sampling history plots for all the schemes are shown in Figures 5.26, 5.27, and 5.28 in which the x -axis is the simulation index and the y -axis is the scaled parameter value in the range $[0,1]$. SOPSO was concentrating on good quality models indicated by the blue coloured points (refer to Table 5.8), while MOPSO-choice1 and MOPSO-choice2 were sampling different parts of the parameter space and larger region of good quality models indicated by the blue and green coloured points representing different range of misfits for example P1, P25, P26, and P37. MOPSO-choice2, in particular have larger spread of good fitting solutions as shown in Figure 5.28 with the colour code indicates the quality of the solution.

The box plot for 20 runs generational minimum per generation for both SOPSO and MOPSO two choices are shown in Figures 5.29, 5.30, and 5.31. SOPSO and MOPSO-choice2 have fast convergence speed compared to MOPSO-choice1. This suggests that the grouping of the wells and production data have an impact on the speed of convergence and demonstrates the benefit of using the multi-objective approach in MOPSO-choice2 where a diverse set of models has been obtained as was highlighted in Figure 5.28.

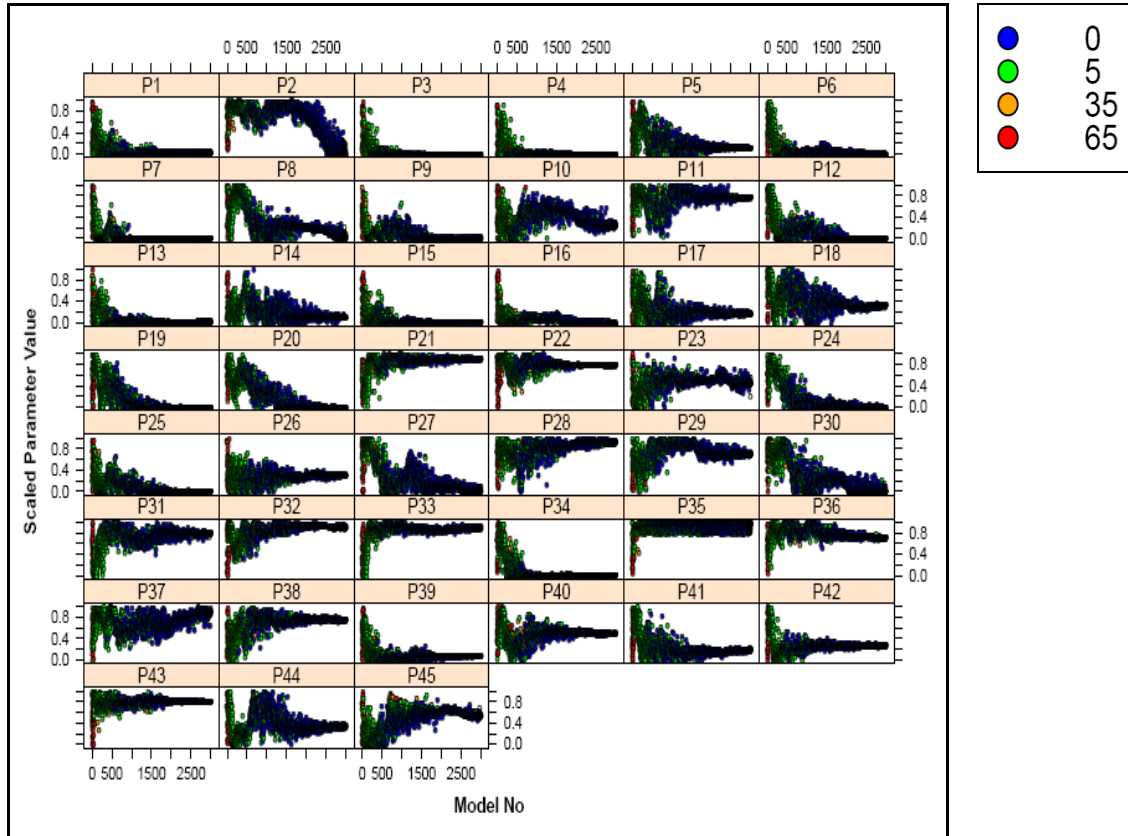


Figure 5.26: Sampling history for SOPSO

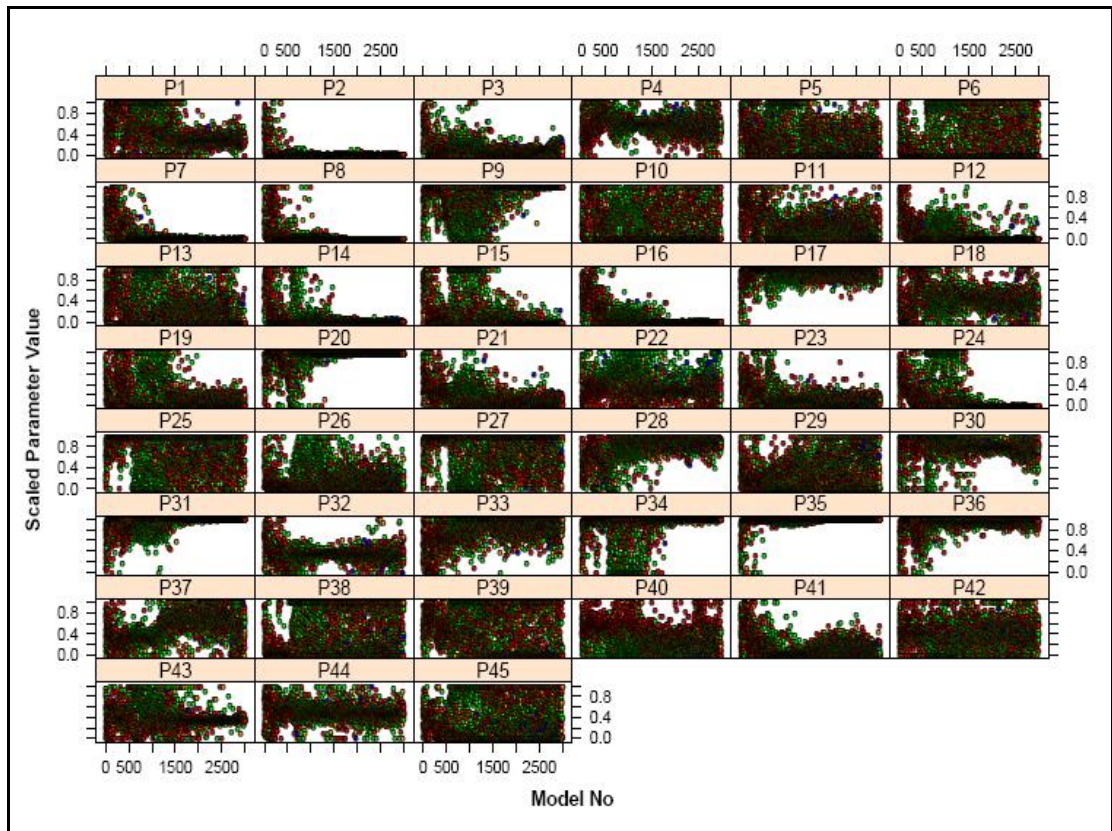


Figure 5.27: Sampling history for MOPSO – choice 1

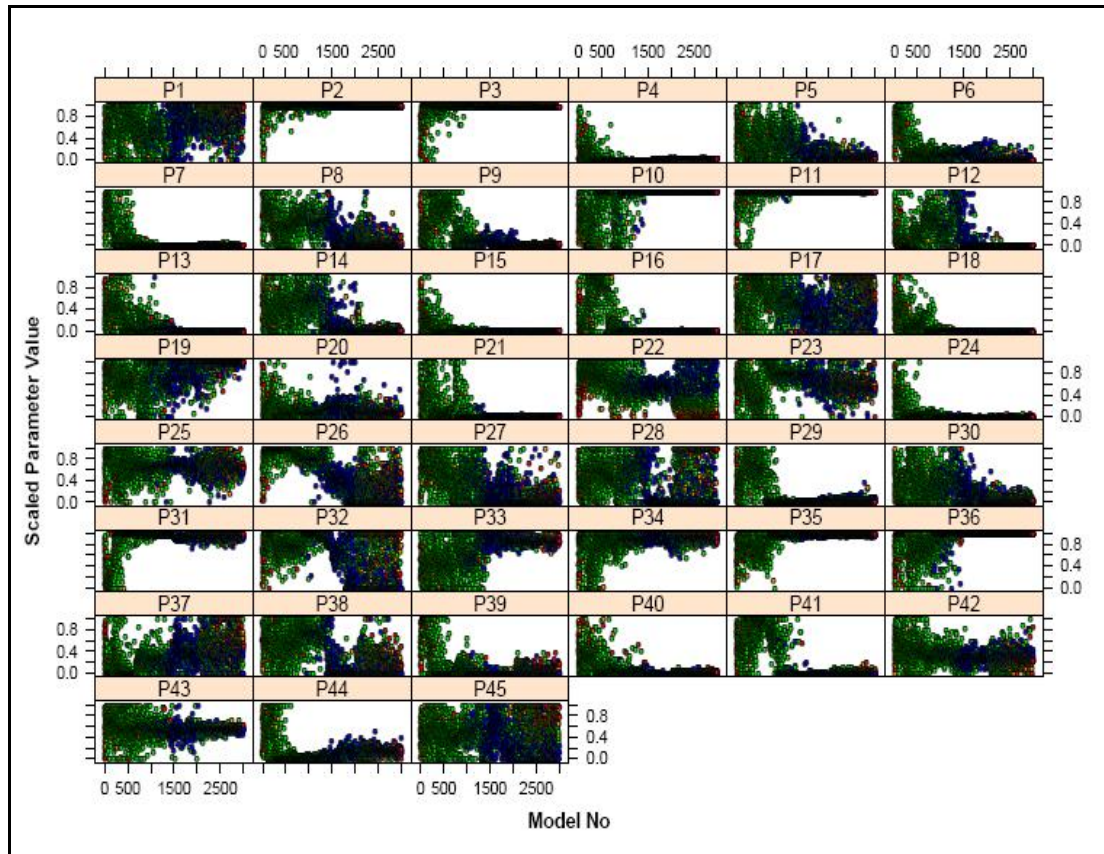


Figure 5.28: Sampling history for MOPSO – choice 2

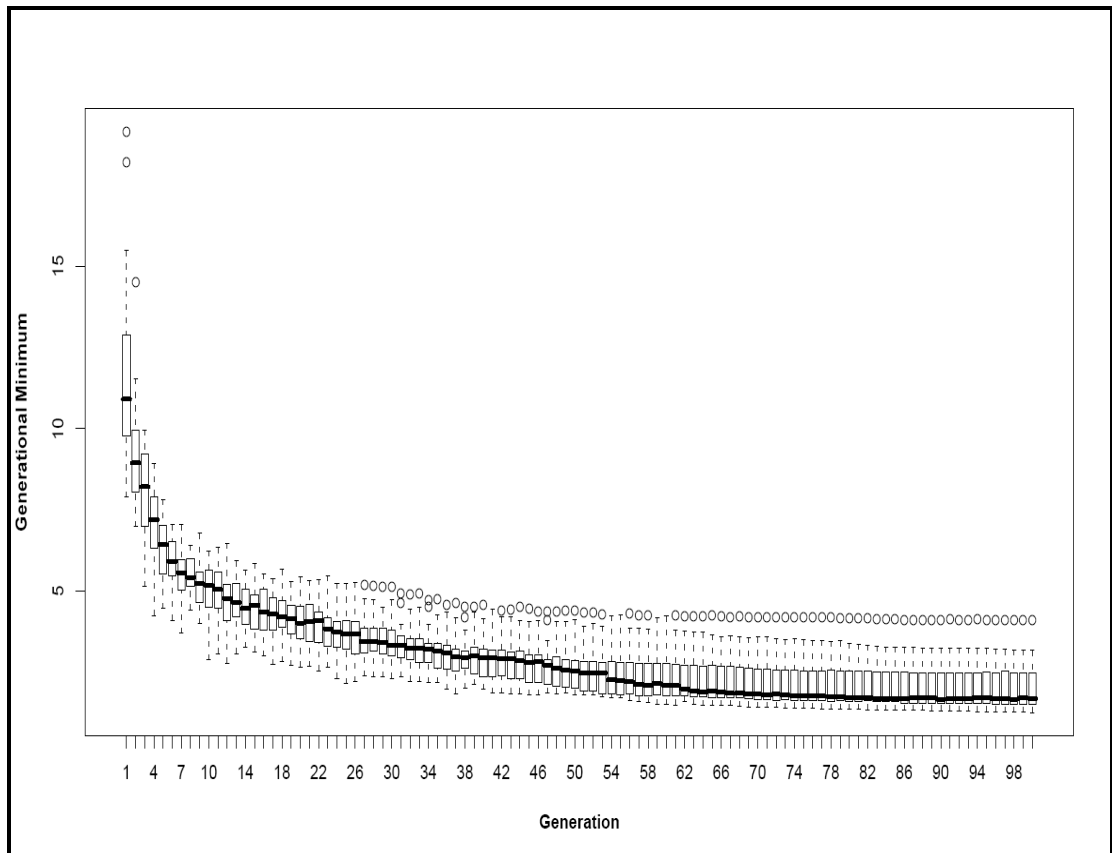


Figure 5.29: Speed of convergence for SOPSO approach with a population of 30 individuals for 100 generations – 20 runs

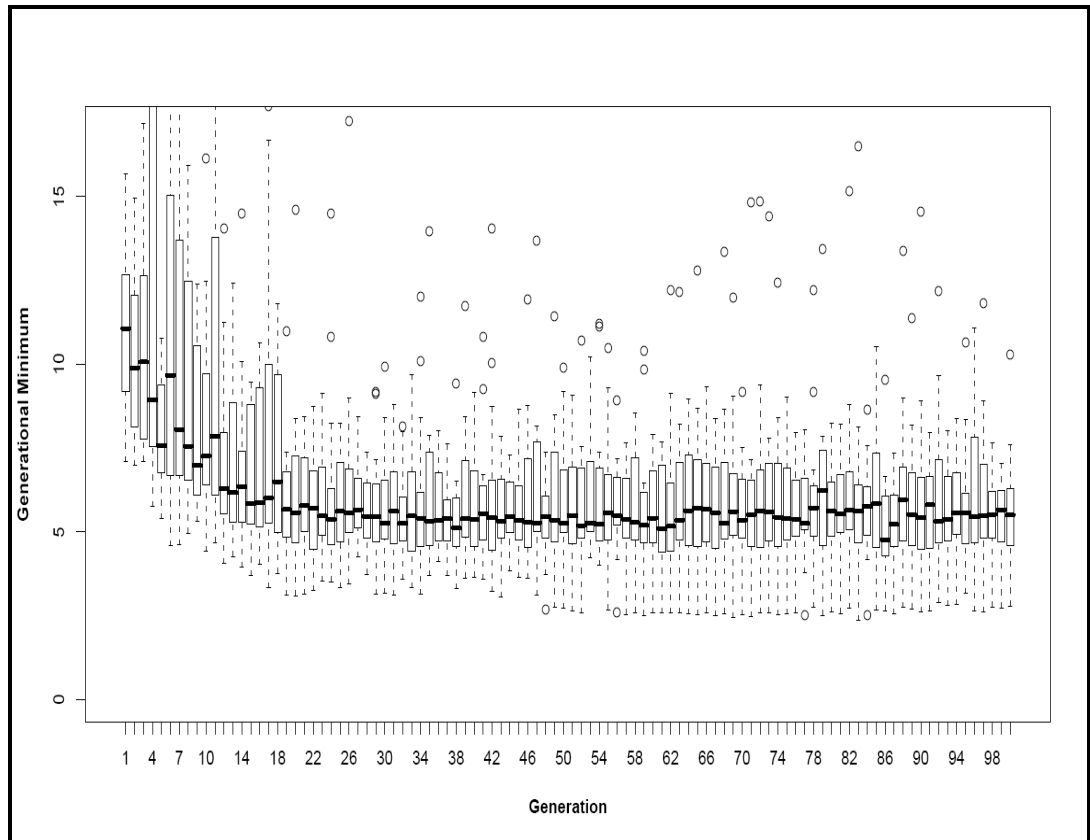


Figure 5.30: Speed of convergence for MOPSO – choice 1 approach with a population of 30 individuals for 100 generations – 20 runs

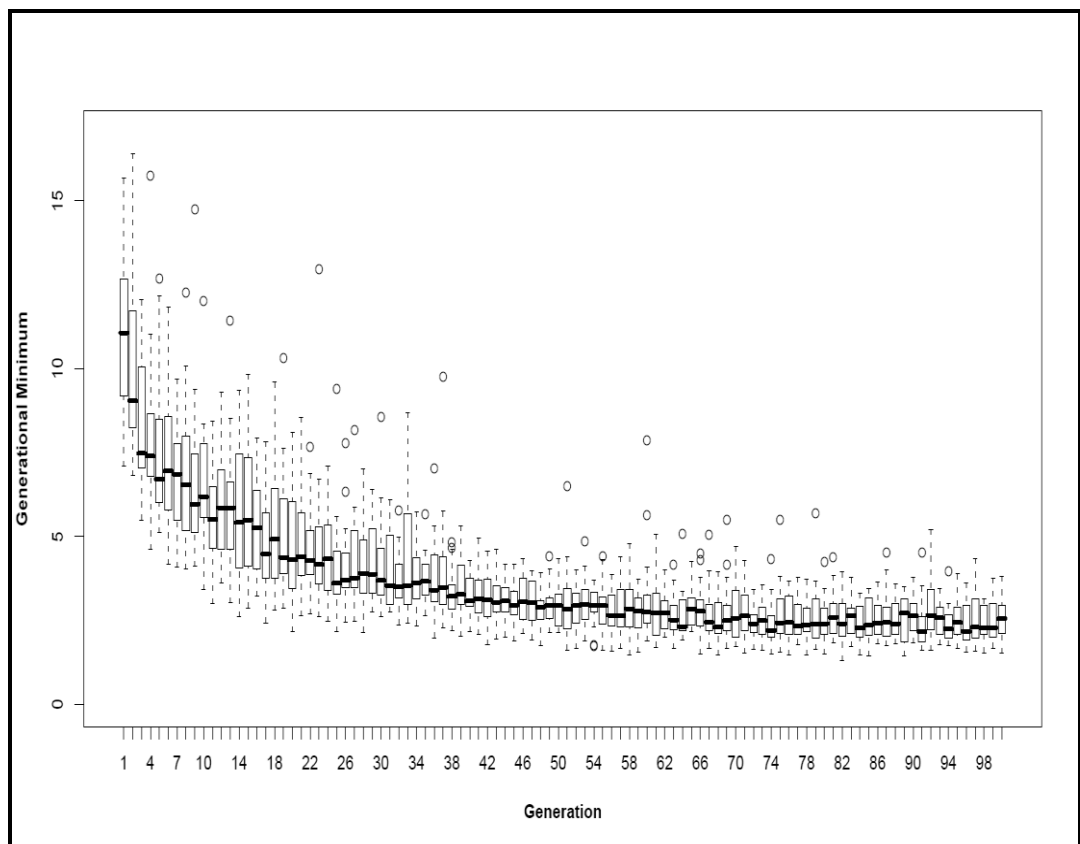


Figure 5.31: Speed of convergence for MOPSO – choice 2 approach with a population of 30 individuals for 100 generations – 20 runs

5.9.4 Uncertainty Quantification

The Bayesian credible intervals for the three schemes are shown in Figure 5.32, 5.33, and 5.34 where all methods captured the simulated truth estimate.

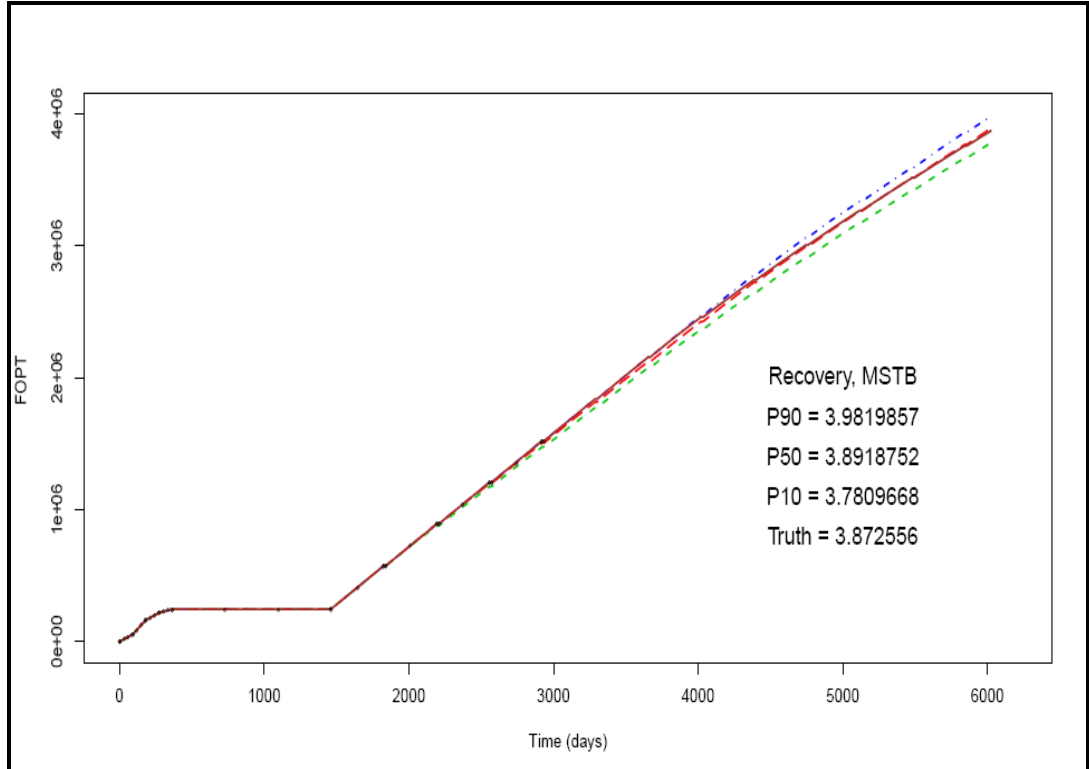


Figure 5.32: FOPT Bayesian credible intervals for SOPSO: P90–P10 = 201.0 STB

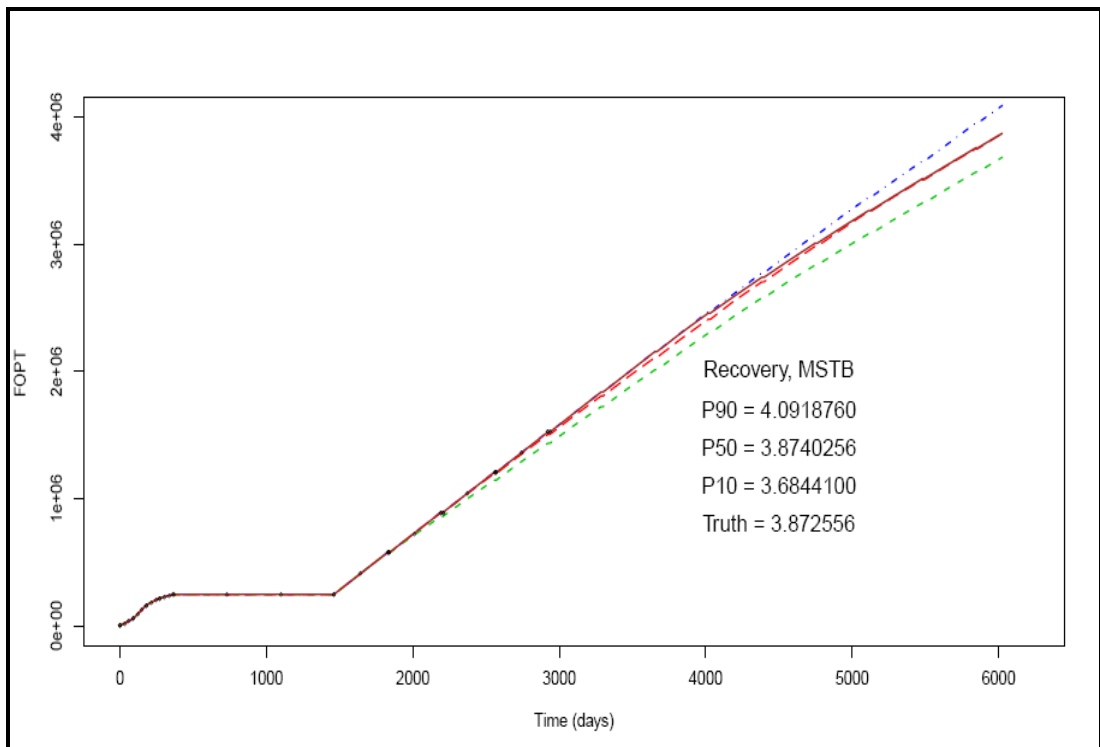


Figure 5.33: FOPT Bayesian credible intervals for MOPSO-choice1: P90–P10 = 407.5 STB

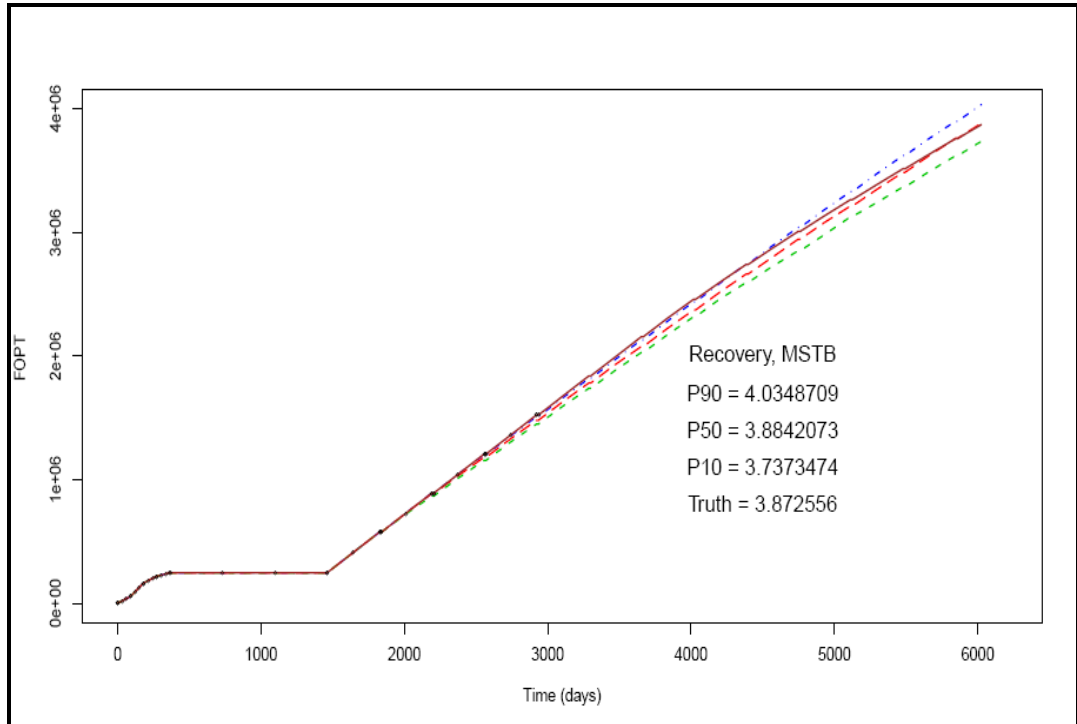


Figure 5.34: FOPT Bayesian credible intervals for MOPSO-choice2: P90–P10 = 297.5 STB

The total recovery at the last step is depicted in Figure 5.35 where all methods captured the true value in the Bayesian credible intervals and both multi-objective MOPSO schemes obtain wider ranges of uncertainty than SOPSO.

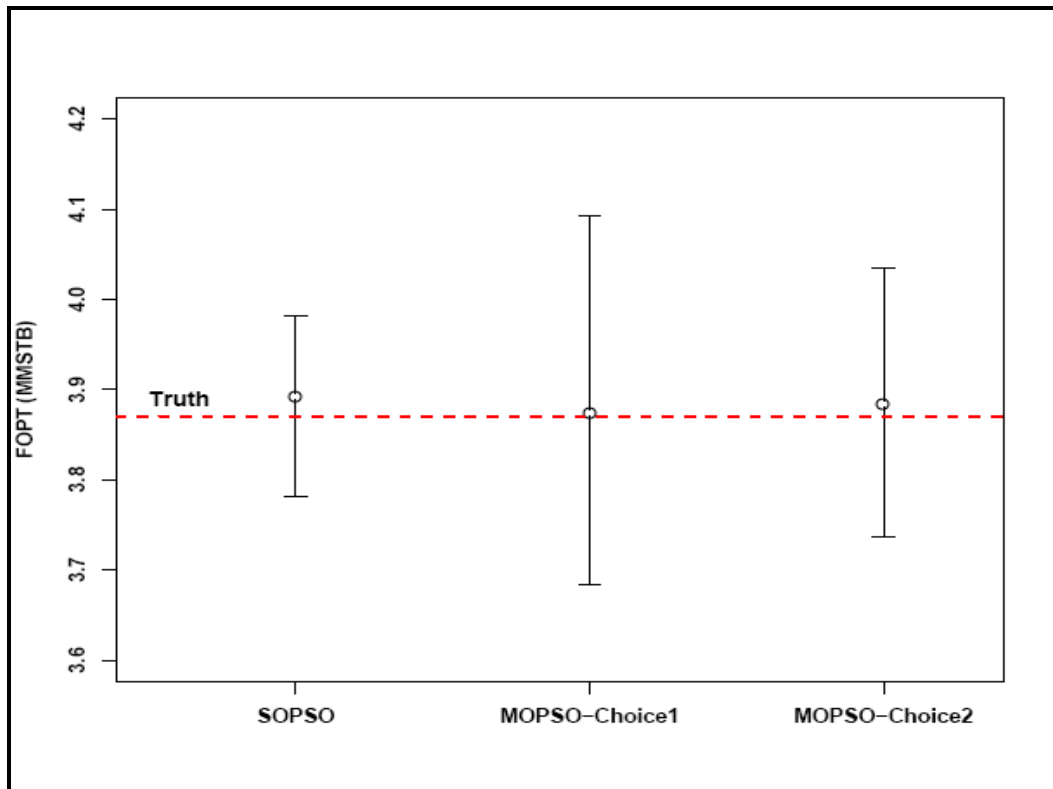


Figure 5.35: Total Recovery for the SOPSO, MOPSO-choice1, and MOPSO-choice2

Finally, the cumulative distribution function for oil rate and the incremental oil production after 16.5 years for the schemes are shown in Figure 5.36 and 5.37. For oil rate the largest uncertainty estimates were obtained with the two MOPSO choices while SOPSO obtained narrower ranges in comparison. The uncertainty estimate for the incremental oil production after 16.5 years obtained with the MOPSO-choice1 is slightly larger than MOPSO-choice2 and SOPSO even though the sampling performance and the quality of the history match were not as good as in the others, followed by MOPSO-choice2, then SOPSO, yet, these estimates are comparable.

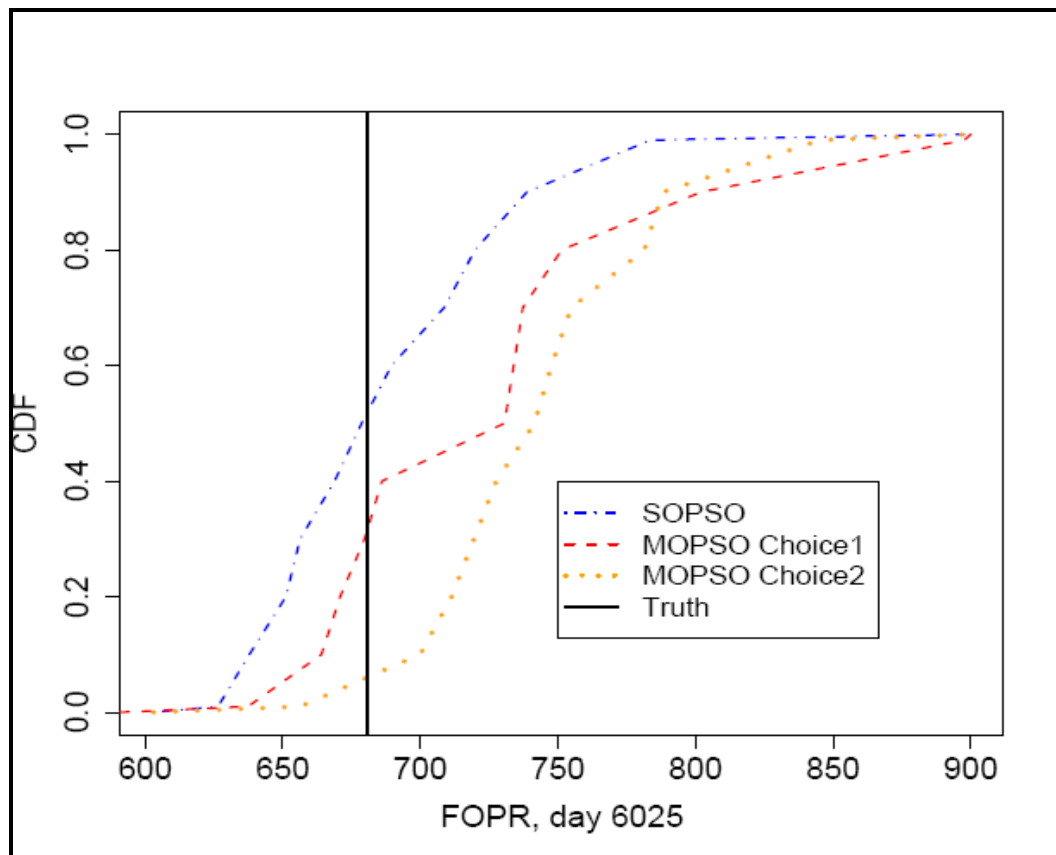


Figure 5.36: CDF for FOPR

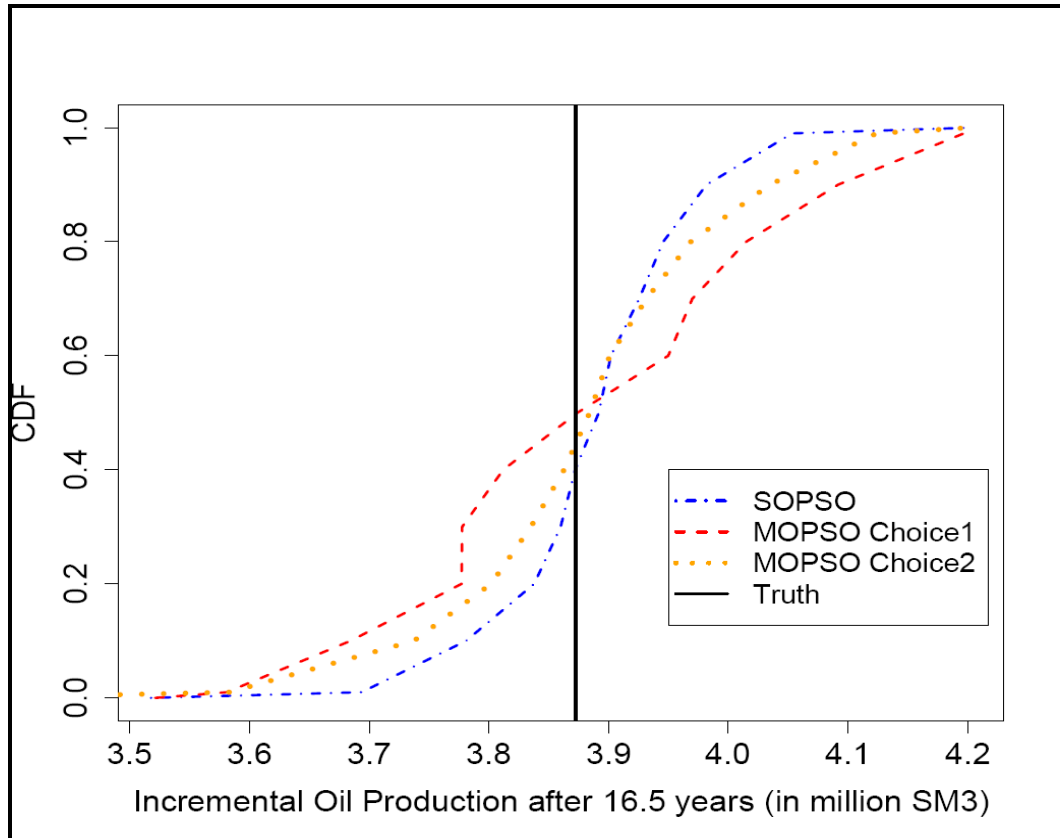


Figure 5.37: CDF for FOPT

5.10 Chapter Summary

Assisted history matching algorithms in previous research studies have been widely utilised and have primarily focused on optimising a single objective function in which all the objective function components are aggregated into a single one. Yet, the nature of the history matching process is multi-objective where we have multiple types of measurements being collected at different times from different wells. An alternative technique is the multi-objective approach which has seen successful applications and large growth replacing the single objective approach among engineering communities by finding a set of solutions which optimally balance the different objectives simultaneously while maintaining diversity. The success of these techniques is due to the characteristics of the multi-objective approach to individually improve the multiple objectives relevant to the different matching quantities. Multi-objective optimisation techniques aim to find multiple minima and to characterise a theoretical method for finding multiple scenarios as the input for reliable uncertainty estimations.

This chapter has reviewed the main aspects of MOPSO variants used in the literature, investigated, and modified a multi-objective particle swarm optimisation scheme that

uses a crowding distance mechanism in cooperation with a mutation operator to preserve the diversity of solutions. This technique was applied on two well-known synthetic reservoir simulation models and the results were compared with the single objective methodology. Analysis of matching quality and predictive uncertainty based on the resulting models was conducted to obtain the uncertainty predictions envelopes for both strategies. The comparative results suggest that, for the reservoirs under consideration, with the specified definition of the objective functions, that, the multi-objective particle swarm approach is highly competitive in obtaining a well distributed set of good fitting reservoir models.

In the first IC Fault model example MOPSO has been more than twice faster than the single objective approach to achieve a similar match quality for more robust future predictions. In the example the speed efficiency result is robust using 20 seeds. MOPSO scheme has provided more accurate estimation of uncertainty in predictions in comparison to the benchmark database results than the single-objective SOPSO scheme. This study showed how the MOPSO algorithm could efficiently improve the history matching. The benefits of using a multi-objective scheme are to enhance obtaining a diverse set of history matches while if possible reducing the number of simulations required for achieving a similar matching performance in comparison with the single scheme as shown in the IC Fault model example.

In PUNQ-S3 model, two ways of defining the objective functions have been tested. The second choice, MOPSO-choice2, where the wells are divided into two groups, has obtained better match than the first choice, MOPSO-choice1, where the type of the production data as way of defining the objectives has been used as in the IC Fault model. The second MOPSO-choice2 scheme has provided a slightly slower convergence than the SOPSO with more diverse good fitting models which has led to larger uncertainty estimates.

The conclusions drawn from the two case studies are that the MOPSO scheme gives flexibility in optimising different objectives simultaneously, obtains better history matches, has faster convergence speed, and gives a more accurate estimation of uncertainty. To answer the question of how many objectives to use: the rule of thumb is to start simple with fewer objectives and then add more objectives as required. Multi-

objective optimisation can also be used to understand, analyse the simulations, and group the conflicting objectives in different ways, leading to the identification of more possible scenarios and then use them in predictions to facilitate decision-making.

Chapter 6 – Advanced MCMC

Techniques for History Matching

Uncertainty Quantification – Part I:

Hamiltonian Monte Carlo

Assisted history matching and uncertainty quantification techniques for reservoir simulation broadly reviewed in Chapter 3 involve approaches based on optimisation and those based on Bayesian inference.

Global optimisation methods such as Genetic Algorithms (Carter and Ballester, 2004; Erbas and Christie, 2007a), Neighbourhood Algorithm (Sambridge, 1999a), Estimation of Distribution (Petrovska and Carter, 2006), and Particle Swarm Optimisation (Mohamed et al., 2010a, 2010b, 2010d), find better models through an automated process which changes unknown parameter values. These algorithms generate a sequence of parameter values that generally improve the history match as time evolves. However, they do not provide any statistically valid method of assessing uncertainty without additional calculations – the reason for this is that the distribution of parameter values is mainly controlled by the algorithm settings; an example of the variability in estimates can be seen in Figure 6.1 from Erbas and Christie (2007a) as well as our results using particle swarm optimisation in Chapter 4. This can be corrected by running a second code to compute probabilities associated with each set of parameters, for example the NAB code developed by Sambridge (1999b).

Approaches based on Bayesian inference (Bayes and Price, 1763; Jaynes, 2003), on the other hand, aim at estimating the posterior probability over the reservoir properties, and

are based on, for example, particle filters such as the Ensemble Kalman Filter (EnKF) (Aanonsen et al., 2009; Evensen, 2007) or Markov Chain Monte Carlo (MCMC) approaches, such as the Metropolis Algorithm, Langevin MCMC (Ma et al., 2008), Hamiltonian Monte Carlo (Mohamed et al., 2010b) and combinations of evolutionary algorithms with MCMC (Holloman et al., 2006; Vrugt et al., 2009).

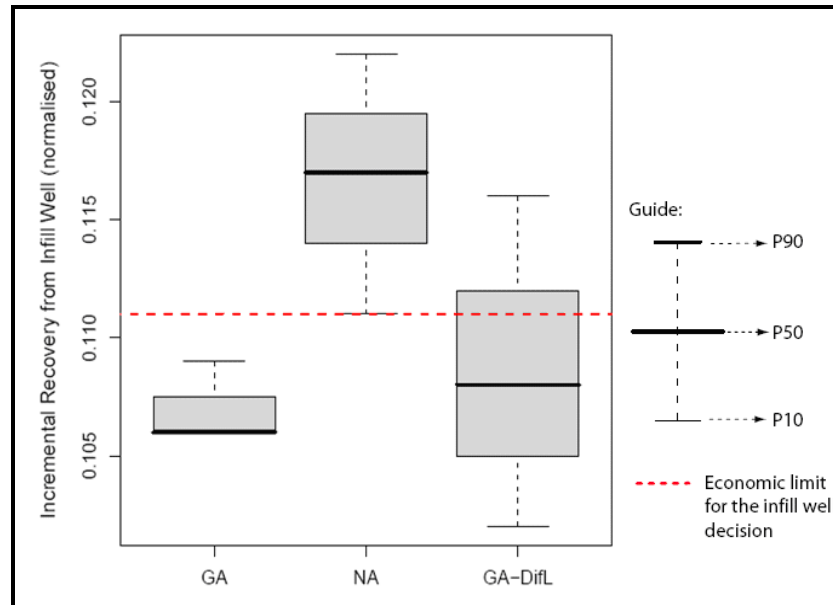


Figure 6.1: Different sampling conditions affected an infill well decision (Erbas and Christie, 2007a)

This chapter presents the application of the Hamiltonian Monte Carlo method to generate history matched reservoir models. The Hamiltonian Monte Carlo algorithm was originally developed in a landmark paper by Duane et al. (1987) to model physical systems. The method originally called “Hybrid Monte Carlo” abbreviates to “HMC”, the later “Hamiltonian Monte Carlo” is commonly used being more specific and descriptive. Hamiltonian Monte Carlo unites the MCMC and molecular dynamics approaches. The method has seen many applications early on in quantum chromodynamics. Ever since then it has become popular in many applications including statistical physics (Akhmatskaya and Reich, 2010; Akhmatkaya et al., 2009; Gupta et al., 1988; Gupta et al., 1990; Hasenbusch, 2001; Sexton and Weingarten, 1992), neural network models (Choo, 2000; Neal, 1996a; Zlochinn and Baram, 2001), Bayesian neural networks (Neal, 1996a, 1996b), data assimilation (Alexander et al., 2005), computational chemistry (Hansmann et al., 1996; Schütte, 1999; Tuckerman et al., 1993), statistical problems (Ishwaran, 1999; Liu, 2008; Schmidt, 2009), and Bayesian statistical inference (Chen et al., 2000; Girolami and Calderhead, 2011; Neal, 1993).

Firstly, a review of the Bayesian approach to tackle inverse problems and parameter estimation is provided. An overview of some classical Markov chain Monte Carlo (MCMC) methods along with the basic concepts and definitions then follows based on materials from Alfaki (2008), Choo (2000), Neal (2011), Robert and Casella (2011) and others. In Section 6.3, Hamiltonian Monte Carlo and its use in constructing a Markov chain Monte Carlo is presented. It discusses the main components of the algorithm and the implementation of the algorithm. Numerical testing is carried out on analytical functions experiments first and intended as a proof of the concept of how HMC could be used in the field of history matching by establishing a specific formalism and implementing it for simple examples. This step allows us to explore the underlying workflow of the technique which is an essential step for further treatment. This is then followed by an application in a reservoir model history matching problem.

6.1 Inverse Problems: A Bayesian Perspective

The Bayesian approach to solve inverse problems differs fundamentally from the conventional deterministic optimisation approach in the nature of the solution. Rather than obtaining the single best solution in the deterministic approach we obtain a probability distribution over the solutions instead in the Bayesian approach (Aster et al., 2005). The interpretation is due to data uncertainty e.g. as a result of incomplete information or imprecise measurements. Bayes theorem allows higher levels of interpretation like the selection or rejection of particular models (Hanson et al., 1997).

The Bayesian approach has the following advantages: it can easily represents the insufficiency of the data in terms of the probability, it provides a unified way for model uncertainty in a single framework, and it can conveniently update the degree of uncertainty for example increasing confidence by adding more data to prior information.

6.1.1 Bayesian Inference

Bayes Theorem (Bayes and Price, 1763) is the formal rule by which probabilities are updated given new data (Feller, 1966; Jaynes, 2003). Bayes Theorem is given by Eq. (6.1):

$$p(m|O) = \frac{L(O|m)\pi(m)}{p(O)} \quad \text{Eq. (6.1)}$$

$p(m/O)$ is the posterior probability inferred – that is the probability of the model has the values $m \in \mathbb{R}^M$, of M variables, m_1, \dots, m_M given the observed data O . The

conditional probability distribution $L(O/m)$ is the likelihood term – it is the probability of the data assuming a particular setting if the model is true. The normalising constant $p(O) = \int_{\mathbb{R}^M} L(O|m)\pi(m)dm$ is referred to as the evidence of plausibility; if this term is small, it suggests that the model does not fit the data well. This term allows comparison of all models in the set for consistency with the data O . Choosing the best models for given data among classes of models is an active area of research in modern Bayesian statistics (Robert, 2007). $\pi(m)$ is the prior probability over parameters m expressing our prior belief about m , and can be given as a sum of independent probabilities for model parameters, or as some more complex combination of model inputs. It is noted that it is difficult to guess a good setting for the parameters; however it is easier to guess a prior distribution for parameters, in terms of its mean and some measure of its spread. The priors may be based on problem-specific attributes such as ancillary modelling predictions, empirical evidence, scientific knowledge, or expert judgement (Taylor, 1993). Priors may alternatively signify default priors, based on a more generic problem characterisation, for example uniform priors or may even symbolise the results of a previous iteration of updating. In the sequential outlook of Bayesian analysis each posterior distribution becomes a prior for the next stage of assessment. Such a sequential analysis is complementary to the iterative conventional optimisation approach discussed in Section 3.4.1. Recent research to determine a realistic prior based on geological data and geological characteristics has been investigated by Arnold (2008) and Mittermeir et al. (2010). More recent work in obtaining richer informative 3D priors for channel systems with novel machine learning methods has also been explored by Rojas (2010).

The posterior probability distribution represents the basis for computing useful statistical quantities of interest. The marginal probability of a certain parameter m_i , given all other parameters m_j , is inferred by integrating the posterior probability over the other parameters as in Eq. (6.2). Similarly, two-parameter interactions may be obtained as for $p(m_i, m_j | m_k; m_k \in \{m \setminus m_i, m_j\})$.

$$p(m_i | m_j; m_j \in \{m \setminus m_i\}) = \int \cdots \int p(m | O) \prod_{\substack{j=1 \\ j \neq i}}^d dm_j \quad \text{Eq. (6.2)}$$

In order to compute the term in Eq. (6.2), the generic Bayesian integral, $\langle J \rangle$, in Eq. (6.3) has to be firstly calculated in which $J(m)$ is a function such that the expectation values $E[J(m)]$ fully characterises the distribution of the random variable m such as the distribution of model parameters. This is known as the (generalised) problem of moments. The term $\langle J \rangle$ in Eq. (6.3) similarly needs the evaluation of the normalising constant $\int_{\mathbb{R}^M} L(O|m)\pi(m)dm$ in Eq. (6.1). The normalising constant is usually intractable even for very low values of d .

$$\langle J \rangle = \int_{\mathbb{R}^N} J(m)p(m|O)dm \quad \text{Eq. (6.3)}$$

A Monte Carlo strategy provides a way for evaluating the integral in Eq. (6.1) by generating a representative ensemble $\{m_1, m_2, \dots, m_N\}$ of samples from $p(m|O)$ that are in some way random but rather than coming from regions of low contribution to the integral, they are more likely to come from regions of high contribution to the integral. The integral of $J(m)$ can then be approximated without explicit evaluation of the normalising constant. Therefore, the integral in Eq. (6.3) is approximated with the Monte Carlo approach as given in Eq. (6.4) where N represents the number of models in the ensemble.

$$\langle J \rangle = \frac{1}{N} \sum_{i=1}^N \frac{J(m_i)p(m_i|O)}{h(m_i)} \quad \text{Eq. (6.4)}$$

The posterior $p(m|O)$ in most practical cases can be quite a non-standard distribution and thus direct drawing of independent samples may be impossible. In a relaxed formulation, the ensemble $\{m_{(i)}\}$ may be obtained by any process that draws samples by the support of $p(m|O)$ in proper proportions so that the integral of $J(m)$ with respect to the distribution $p(m|O)$ is approximated by the sample mean. Thus, if $p(m_i|O) \approx h(m_i)$ where the samples $\{m_{(i)}\}$ do not necessitate independency then Eq. (6.5) holds.

$$\begin{aligned} \langle J \rangle &= \frac{1}{N} \sum_{i=1}^N \frac{J(m_i)p(m_i|O)}{h(m_i)} \\ &\approx \frac{1}{N} \sum_{i=1}^N J(m_i) \quad \text{for } h(m_i) \approx p(m_i|O) \\ m_i &\approx m(m_i) \end{aligned} \quad \text{Eq. (6.5)}$$

In the Markov chain Monte Carlo (MCMC) methods, generating an ensemble of models, m_1, m_2, \dots, m_N , whose distribution $h(m)$, approximates $p(m|O)$, can reduce the task of evaluating the generic integrals to computing averages over $J(m_i)$. This is achieved using a Markov chain which has its stationary distribution defined by $p(m|O)$ (Gilks et al., 1996; Kolehmainen, 2001). This expression accounts for probability density rather than probability.

Once an independent and identically distributed set of samples from the chain ($\{m_1, m_2, \dots, m_N\}$ from $p(m|O)$) have been drawn, we can then calculate posterior quantities of interest and some statistics: e.g. the posterior mean of field oil recovery as in Eq. (6.6), and the variance which quantifies the uncertainty as in Eq. (6.7).

$$\text{Mean}_t = \left[\frac{1}{N} \sum_{i=1}^N q^{obs}(m_i)_t \right] \quad \text{Eq. (6.6)}$$

$$\text{Var}_t = \left[\frac{1}{T} \sum_{t=1}^T (q^{obs}(m_i) - \text{Mean})^2 \right] \quad \text{Eq. (6.7)}$$

In the following Section 6.2 the basic principles of Markov chain Monte Carlo (MCMC) methods and the most common one, the Metropolis-Hastings algorithm are presented.

6.2 Markov Chain Monte Carlo (MCMC) Methods

This section will present necessary background and review of some classical Markov chain Monte Carlo (MCMC) methods at a level of technical detail sufficient to explain the work in this thesis. More detailed presentations may be found elsewhere (e.g., Choo, 2000; Feller, 1968; Gamerman, 1997; Gilks et al., 1996; Mackay, 2003; Neal, 1993, 2011; Robert and Casella, 2000).

6.2.1 Basic Definitions and Background

Definition 6.1. (Markov chain) A Markov chain, named after Andrey Markov, is a series of random variables $\{X_t | t \in T\}$ called a stochastic process such that the distribution of X_t given all the earlier states, X_0, \dots, X_{t-1} , depends only upon the given preceding one X_{t-1} . That is, given X_{t-1} , X_t is independent of all earlier X 's. This is stated in Eq. (6.8) and known as the Markov property.

$$P(X_t | X_{t-1}, X_{t-2}, \dots, X_0) = P(X_t | X_{t-1}) = T(X_t | X_{t-1}) \quad \text{Eq. (6.8)}$$

A Markov chain is defined by three components: the state space in which all X_i 's live, the distribution over the initial state $P(X_0)$, and the transition probability function or so called transition kernel T . The Markov chain is called time-homogeneous if the transition matrix is the same after each step meaning that the transition kernel does not depend on the time $(t-1)$.

Definition 6.2. (Ergodicity) A Markov chain is called an ergodic chain if the Markov chain satisfies the following properties:

1. Stationarity: As $t \rightarrow \infty$ the Markov chain converges to its invariant (stationary or equilibrium) distribution.
2. Irreducibility: Starting from any starting point, the Markov chain has positive probability to reach a non-empty set of states within a finite number of steps. That is to reach any state from any other state.
3. Aperiodicity: The state can return to itself at irregular times. This guarantees that the chain will not oscillate between different sets of states.

The ergodicity condition of the Markov chain is sufficient for the existence of the invariant distribution $\pi(x)$ independent of the initial probability at the initial state. If a Markov chain has a proper invariant distribution $\pi(x)$, and it is both irreducible and aperiodic, then π is the unique invariant distribution and is also the equilibrium distribution of the chain (Tierney, 1994; Rosenthal, 2001).

Definition 6.3. The reversibility condition expressed in the detailed balance equation in Eq. (6.9) is sufficient, but not a necessary condition for a distribution $\pi(x)$ to be stationary distribution.

$$\pi(x)T(x, x') = \pi(x')T(x', x) \quad \text{Eq. (6.9)}$$

A Markov chain that converges to a desired distribution can be constructed by ensuring that it is aperiodic, irreducible, and has the target distribution as an invariant distribution. The first portion of a Markov chain is typically not representative of the invariant distribution, and is usually discarded. These samples can be used to estimate usefully required statistical properties with a large number of samples generated to improve the estimation accuracy.

Definition 6.4. (Markov Chain Monte Carlo) A Markov chain Monte Carlo (MCMC) for simulating a target distribution $\pi(x)$ is any method that constructs an ergodic Markov

chain $\{X_t\}_{t=1}^N$ with probability transition kernel T that converges to stationary distribution $\pi(x)$ in the long run, independent of the initial state of the chain.

6.2.2 Basic MCMC Techniques

Many strategies are available for constructing Markov chains with a specified invariant distribution. Gibbs sampling and Metropolis-Hastings algorithms are two commonly used basic MCMC techniques and are outlined in the following subsections.

6.2.2.1 The Gibbs Sampler Algorithm

The Gibbs sampling algorithm, named after the physicist Josiah Gibbs, is a Markov chain with invariant distribution is constructed with conditioning (Casella and George, 1992). The Gibbs sampler (Geman and Geman, 1984) has its roots in image processing and is usually used for simulating multivariate distributions.

In this approach only one parameter is drawn from the conditional distribution at a time, holding all others fixed. At each update the parameter is replaced by a sample point drawn from its distribution conditioned on all the other parameters with their most recent values. The new candidate point is always accepted. The Markov chain constructed with Gibbs sampling is an ergodic chain in which the three properties hold. The new value of the parameter is selected regardless of the old value it replaces, thus the desired distribution is invariant and the resulting multivariate state is drawn according to the target distribution. Moreover, since all values of X_t have non-zero probability to be drawn, the Markov chain constructed is irreducible and aperiodic provided that all parameters are updated with time t . Algorithm 6.1 presents the pseudo-code for Gibbs sampler. Gibbs sampler is simple, yet requires sampling from the full conditional distributions, i.e. the conditional distribution of each parameter, which may not be possible for complex distributions and makes it less applicable in practice (Martinez and Martinez, 2002). There are other schemes which do not require this.

Algorithm 6.1: The Gibbs Sampler

1. Initialise $t = 0, N_{\text{samples}}$, Initial point $X_0 = (x_1^{(0)}, \dots, x_d^{(0)})$
2. For $t = 0$ to $(N_{\text{samples}} - 1)$ do
3. Draw $x_1^{(t+1)} \sim \pi_1(x_1 | x_2^{(t)}, \dots, x_d^{(t)})$
4. Draw $x_2^{(t+1)} \sim \pi_2(x_2 | x_1^{(t+1)}, x_3^{(t)}, \dots, x_d^{(t)})$
5. \vdots
6. Draw $x_d^{(t+1)} \sim \pi_d(x_d | x_1^{(t+1)}, \dots, x_{d-1}^{(t+1)})$
7. $X_{t+1} = (x_1^{(t+1)}, \dots, x_d^{(t+1)})$
7. End For

6.2.2.2 The Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm was originally introduced by Metropolis et al. (1953) and generalised by Hastings (1970). It is a well-known algorithm for constructing a Markov chain with a desired invariant distribution. The algorithm has been used extensively in statistical physics (Hammersley and Handscomb, 1964) which provided later the seed to use these methods in the statistics literature (Bernardo and Smith, 1994; Gelfand and Smith, 1990; Müller, 1993; Smith and Roberts, 1993; Tierney, 1994), as well as in many other fields. The Metropolis-Hastings algorithm has been cited as being among the top ten algorithms having the greatest influence on the development and practice of science and engineering in the 20th century (Beichl and Sullivan, 2000; Cipra, 2000). The general algorithm framework is demonstrated in Meyn and Tweedie (1993). In Roberts and Tweedie (1996) an application of Metropolis-Hastings methods is provided.

The Metropolis-Hastings algorithm with simple proposals requires only that we are able to evaluate the target probability density $\pi(x)$ at a given state. In this algorithm, all the parameters can be changed at once. Thus, the Gibbs sampler can be viewed as a particular case of one of the Metropolis algorithm variants (multiple-block Metropolis-Hastings). In the simplest Metropolis-Hastings, the parameter vector is perturbed from the current sequence point by adding a trial step drawn randomly from an easy-to-sample symmetric proposal distribution, $q(\cdot | x'_t)$, that has the same property as the target distribution. A convenient choice for a proposal distribution is the normal distribution centred at the current state, x'_t , in the simulation and with a fixed covariance matrix. The proposal distribution should satisfy the irreducibility and aperiodicity conditions which can be achieved if the proposal distribution has a positive

density on the same support as the target distribution. The trial proposed state/position is either accepted or rejected for the next state in the chain on the basis of the probability at the trial proposed position relative to the current one given by Eq. (6.10).

$$x_{t+1} = \begin{cases} x'_t, & \text{with probability } \alpha(x_t, x'_t) \\ x_t, & \text{with probability } 1 - \alpha(x_t, x'_t) \end{cases} \quad \text{Eq. (6.10)}$$

where

$$\alpha(x_t, x'_t) = \min\left(\frac{q(x_t | x'_t)\pi(x'_t)}{q(x'_t | x_t)\pi(x_t)}, 1\right). \quad \text{Eq. (6.11)}$$

If the trial proposed state is rejected, the state x_{t+1} is set to be the previous state x_t . the ratio in Eq. (6.11) does not require computing the normalising constants ratio if unknown since the ratio is 1. Thus, it is a good attribute of the Metropolis-Hastings.

Algorithm 6.2 presents the variant of Metropolis-Hastings pseudo-code called Global Metropolis-Hastings. In this variant all parameters of the current state x_t are perturbed. Choosing a spherically and symmetric proposal distribution is preferred in this variant (Neal, 1993). In the local Metropolis-Hastings variant, on the other hand, the perturbation is achieved for each parameter one after another. Therefore any appropriate proposal distribution can be used.

Algorithm 6.2: Metropolis-Hastings Sampler

1. Initialise $t = 0$, $N_{samples}$, Initial point $X_0 = (x_1^{(0)}, \dots, x_d^{(0)})$
2. For $t = 1$ to $N_{samples}$ do
3. Propose $x'_t \sim q(\cdot | x_{t-1})$
4. Draw $u \sim Uniform[0,1]$
5. Compute $\alpha(x_t, x'_t) = \frac{q(x_t | x'_t)\pi(x'_t)}{q(x'_t | x_t)\pi(x_t)}$
6. If $u < \min(\alpha(x_t, x'_t), 1)$ then
7. $x_t = x'_t$
8. Else
9. $x_t = x_{t-1}$
10. End If
11. End For

It can be shown that the Metropolis algorithm ensures that the target distribution is an invariant distribution of the Markov chain by satisfying the detailed balance in Eq. (6.9)

(Walsh, 2004). Nevertheless, this does not ensure that the Markov chain is aperiodic and irreducible (Choo, 2000).

The sampling efficiency of the Metropolis algorithm can be tested in sampling a target distribution while using a simple Gaussian distribution, in Eq. (6.12), as the proposal with a fixed covariance Σ centred at the current state x_t in which d represents the number of parameters.

$$p(x'_t) = \frac{1}{(2\pi|\Sigma|)^{d/2}} \exp\left[-\frac{1}{2}(x'_t - x_t)^T \Sigma^{-1}(x'_t - x_t)\right] \quad \text{Eq. (6.12)}$$

In this example if the variance of the Gaussian proposals is too large in comparison with the width of the target distribution, the proposed samples often lie in regions with low probabilities. Thus, the Metropolis algorithm roughly rejects all the time causing slow exploration of the parameter space if smaller variances are not reset to obtain a higher acceptance rate. The acceptance rate is the fraction of proposed samples accepted in the last S samples after reaching the stationary distribution. The shape of the distribution needs to be taken into account for choosing stepsizes. The existence of a long thin region in a target distribution restricts the stepsize that could be taken for the new sample to fall in a high probability region and that is the first problem. The choice of the stepsize must be small enough if the distribution is thin in one direction and long in the other. So the standard deviation choice for the proposal distribution should be comparable in size to the thinnest cross-section of the target distribution and that can be very small if compared with the overall distribution. The second problem, known as the random walk behaviour, results from the fact that the choice of the second step independently of the first leads to the possibility of walking back on the first step as shown in Figure 6.2. In the Metropolis algorithm, the simulation will usually move a distance proportional to \sqrt{N} (Neal, 1993) where N is number of iterations. The Metropolis-Hastings algorithm must be tailored to individual problems by dynamically adapting Gaussian proposal distributions as shown in Figure 6.3 for different kinds of scaling and rotating treatments.

In high dimensional problems traditional MCMC methods suffer from a number of issues: low acceptance rate, high correlations between the consecutive steps, slow mixing, slow convergence and low efficiency. It is advised that the stepsize is chosen so

that the acceptance rate is about 20% to 40% (Gilks et al., 1996). The efficiency of the sampling is determined by the number of steps in the sequence needed to effectively provide a statistically independent sample from the target distribution. The efficiency of the Metropolis-Hastings with optimal choice of the proposal distribution falls as $0.33/d$ (Dunkley et al., 2005; Gelman et al., 1996; Hanson, 2001). With the traditional MCMC, a huge number of samples are needed in order to adequately explore the parameter space, visit all high density regions, converge to the target distribution, and provide plausible and accurate estimates for some statistical properties of interest (Calderhead and Girolami, 2007; Gilks et al., 1996; Robert and Casella, 2000). More details describing the MCMC algorithms and related concepts can be found in Gamerman (1997), Gilks et al. (1996), Mackay (2003), Murray (2007), Neal (1993), Ripley (1981, 1987, 1988), and Robert and Casella (2000).

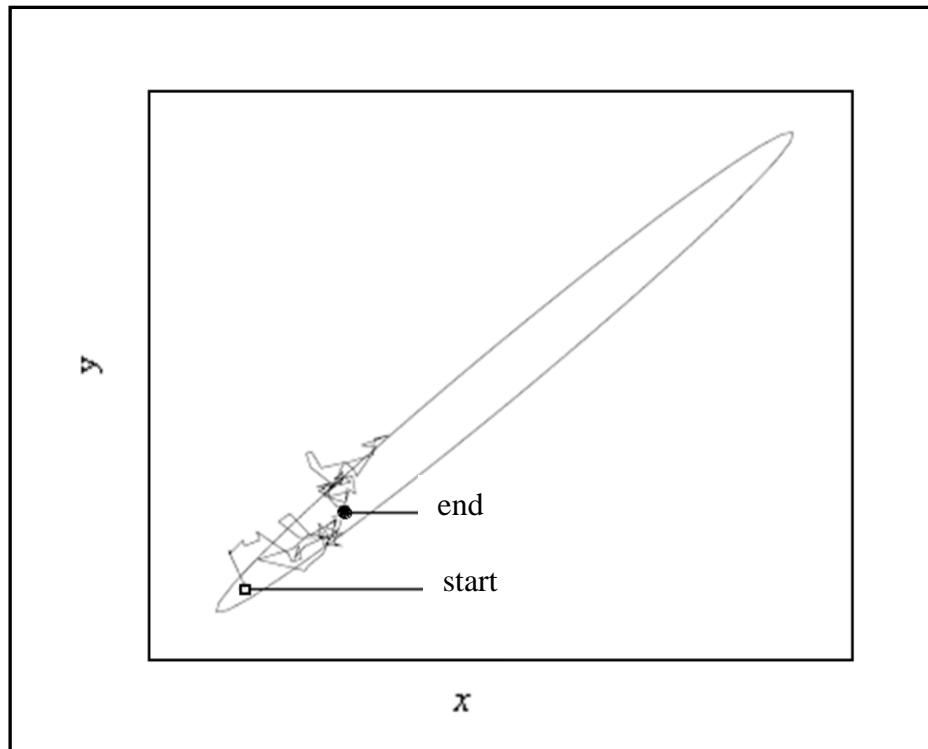


Figure 6.2: Metropolis-Hastings random walk behaviour using simple Gaussian proposals to navigate 2D Gaussian target distribution (Choo, 2000)

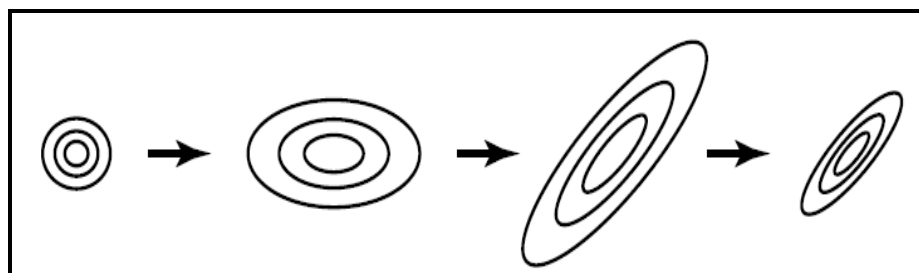


Figure 6.3: Dynamic representation for choosing adapted Gaussian proposals

History matching is an inverse problem that involves a large number of parameters and it is expected that the posterior distribution is complicated under most circumstances and may potentially have long narrow regions. Although, the Metropolis algorithm with simple proposals is a rigorous method frequently employed because of its simplicity, to sample from such complex posterior distribution with Metropolis-Hastings using Gaussian proposals that involve small variances may result in an inefficient random walk as described above. For these reasons we may need to run MCMC for a long time before converging to the target distribution. Consequently, that requires running thousands of flow simulations for the sampling which is described in the literature as random walk behaviour. The weaknesses illustrated make the use of Metropolis-Hastings in sampling high dimensional space unsuitable and motivate a more sophisticated and appropriate method for sampling complicated posteriors, the hybrid Monte Carlo method, which is investigated in this thesis and described in the next section.

6.3 Hamiltonian Monte Carlo (HMC)

Hamiltonian Monte Carlo (HMC) is a Markov Chain Monte Carlo (MCMC) method introduced by Duane et al. (1987) for sampling. HMC merges the benefits of Hamiltonian dynamics (Anderson, 1980; Rossberg, 1983) and Metropolis algorithm (Hanson, 2001; Metropolis et al., 1953), to sample from complex posterior distributions. HMC aims to fix some of the MCMC problems. For example, the Metropolis algorithm (Metropolis et al., 1953) can suffer from slow exploration of the probability distribution if the stepsize is too small, or can suffer from excessive rejection of proposed locations if the stepsize is too high. The main components of the HMC algorithm are derived from the classical mechanics. In this section we describe the basic definitions and equations followed by a demonstration of the implementation of the algorithm as a method for parameter estimation and uncertainty quantification.

The idea of HMC is to regard each set of parameters as a point in parameter space and to introduce a set of auxiliary momentum variables u . The potential energy is defined as the negative logarithm of the posterior probability $U(x) = -\log p(m/O)$, and a kinetic energy as in Eq. (6.13) with the set of masses $\{m_i\}_{i=1}^d$.

$$K(u) = \sum_{i=1}^d u_i^2 / 2m_i \quad \text{Eq. (6.13)}$$

For $m_i = 1$, $K(u) = u^T u/2$. The Hamiltonian or total energy is then $H(x,u) = U(x) + K(u)$. The extended probability distribution is the joint density. Sampling is achieved at two steps: firstly the momenta are sampled from a normal distribution with mean zero and variance 1 while leaving the state x unchanged. This proposal can be considered as a Gibbs sampling update and always accepted. The joint probability distribution is given by Eq. (6.14).

$$p(x,u) \propto e^{-H(x,u)} = e^{-U(x)} e^{-K(u)} \propto p(m|O) N(u;0,1) \quad \text{Eq. (6.14)}$$

By sampling from the extended distribution in the second step and discarding the momentum variables, we can obtain samples from the posterior probability distribution. In the Hamiltonian Monte Carlo, new samples are generated by simulating the time evolution of the physical system with the defined Hamiltonian. The simulation is achieved by solving Hamilton's equations of motion.

6.3.1 Hamiltonian Dynamics

Hamiltonian dynamics are given by Eqs. (6.15) and (6.16):

$$\dot{u} = -\frac{\partial H(x,u)}{\partial x} = -\nabla U(x) \quad \text{Eq. (6.15)}$$

$$\dot{x} = \frac{\partial H(x,u)}{\partial u} = \frac{u}{m_i} \quad \text{Eq. (6.16)}$$

The first equation Eq. (6.15) is Newton's second law of motion (the change of momentum is determined by the gradient of the potential energy $U(x)$). The second equation Eq. (6.16) decides where the state x goes. In this system the total energy H is conserved only if the dynamics are done accurately. Hamiltonian dynamics are time-reversible and preserve volume in state space and total energy. If we simulate the dynamics exactly, we will leave the extended density invariant (Bishop, 2006; Neal, 2011). On the other hand, a Markov chain involving only the change in (x,u) is not irreducible since all samples produced from an initial sample do not part a hypershell of constant H , which breaches the irreducibility constraint of MCMC (Choo, 2000). Thus, the momentum variables are adjusted in such a way that the Markov chain has some chance of reaching all the other values of H after a number of iterations to remedy the constraint. This is achieved through the replacement of momentum variables with new values drawn from the distribution $\exp(-K(u))$. The step leaves the joint density

invariant since u is drawn from the conditional distribution that happens to be independent of x . If the dynamical trajectory of a state is followed for a long time, a state that is less correlated with the initial state might be produced compared to the Metropolis with simple proposals samples. It is worth noting that Neal (1995, 1998) discussed an improvement of the HMC algorithm to further suppress the random walk in drawing the momentum variables by using ordered overrelaxation strategies in the Gibbs sampling stage. This is not considered in this thesis.

6.3.2 Leapfrog Proposals – Discretising Hamilton’s Equations

In reality we are unable to simulate the Hamiltonian dynamics perfectly. Thus, for computer implementation, Hamilton’s equations are simulated by discretising time with a small stepsize Δt . From an initial state at time zero, the state at times Δt , $2\Delta t$, $3\Delta t$ is approximately computed iteratively.

A well-known method to approximate the solution to a system of differential equations is Euler’s method. Neal (2011) indicated that Euler’s method has the tendency to produce a trajectory that diverges to infinity, so some care is required to ensure that the discretised version is reversible, since the Hamiltonian dynamics formulation is time reversible. In practice, the system is simulated with much better results using a leapfrog approach where positions and momentums ‘leap frog’ each other. The leapfrog algorithm of size Δt works as follows:

$$u\left(t + \frac{\Delta t}{2}\right) = u(t) - (\Delta t/2) \times \nabla U(x(t)) \quad \text{Eq. (6.17)}$$

$$x(t + \Delta t) = x(t) + (\Delta t/m) \cdot u\left(t + \frac{\Delta t}{2}\right) \quad \text{Eq. (6.18)}$$

$$u(t + \Delta t) = u\left(t + \frac{\Delta t}{2}\right) - (\Delta t/2) \times \nabla U(x(t + \Delta t)). \quad \text{Eq. (6.19)}$$

Eq. (6.17)– Eq. (6.19) can be abbreviated to Eqs. (6.20) and (6.21):

$$x(t + \Delta t) = x(t) + (\Delta t/m) \left[u(t) - \frac{\Delta t}{2} \times \nabla U(x(t)) \right] \quad \text{Eq. (6.20)}$$

$$u(t + \Delta t) = u(t) - (\Delta t/2) \times \left[\nabla U(x(t)) + \nabla U(x(t + \Delta t)) \right] \quad \text{Eq. (6.21)}$$

Although the leapfrog algorithm simulation is time-reversible and volume-preserving of region of the phase space (Bishop, 2006), finite Δt does not keep H constant and it only preserves energy to second order in the time stepsize Δt , $O(\Delta t^2)$ (Neal, 1993). The total trajectory time $\tau = L \times \Delta t$ where L represents the number of leapfrog steps and τ

represents the trajectory length. This error is obtained after simulating for a fixed time interval τ and is called, the global error. The error produced after one step is called local error which resulted from time t to $t + \Delta t$ and is of order Δt^3 (Neal, 2011). The effect of the global error can be eliminated by applying a Metropolis accept/reject step at the end of the trajectory. The proposal step is accepted/rejected according to the Metropolis rejection test with acceptance probability given by Eq. (6.22), where $H(x', u')$ is the total energy at the proposed step (x', u') .

$$\alpha = \min(1, \exp(-\delta H)) = \min\left(1, \exp\left[-H(x', u') - H(x, u)\right]\right) \quad \text{Eq. (6.22)}$$

In a perfect simulation $\delta H = 0$ leads to $\alpha = 1$ for the proposed new state. That ensures the joint density is kept invariant and permits persistent, rapid movement across a state space while avoiding damaging the equilibrium distribution of the Markov chain through discretisation errors. A pseudo-code for the HMC algorithm using the leapfrog in the dynamics simulation is given in Algorithm 6.3 with Figure 6.4 illustrating the main components.

Leapfrog proposals satisfy the reversibility condition since the negation of the momentum at the end of the leapfrog trajectory of many steps implies that a trajectory from point A to point B is taken back from B to A. However, the momentum negation is not shown in the Algorithm 6.3 implementation since the replacement of the momentum is achieved with resampling before starting new leapfrog trajectory.

The HMC algorithm utilises long leapfrog trajectory lengths, thus avoids the random walk behaviour. As the parameters x travel in the direction of the momentum u during each dynamical proposal, the state of the system tends to move linearly with time, that is, L steps will thus tend to be proportional to L (Neal, 2011). In comparison, Metropolis-Hasting often travels distance proportional to \sqrt{L} for L steps with stepsize Δt (Neal, 1993).

The HMC algorithm utilises long leapfrog trajectory lengths, thus avoids the random walk behaviour. As the parameters x travel in the direction of the momentum u during each dynamical proposal, the state of the system tends to move linearly with time, that is, L steps will thus tend to be proportional to L (Neal, 2011). In comparison,

Metropolis-Hasting often travels distance proportional to \sqrt{L} for L steps with stepsize Δt (Neal, 1993).

Algorithm 6.3: Hamiltonian Monte Carlo Sampler

1. Initialise $t = 0, \Delta t, N_{\text{samples}}, L, x_0$
2. For $i = 1$ to N_{samples} do
3. Draw $u \sim N[\text{mean} = 0, \text{variance} = m]$ # Gibbs Sampling step
4. Set $(x^{(0)}, u^{(0)}) = (x_{i-1}, u)$
5. For $j = 1$ to L do
6. # Alternate full steps for position and momentum
7. $u^{(j-\frac{1}{2})} = u^{(j-1)} - (\Delta t/2) \times \nabla U(x^{(j-1)})$ # Make a half step for momentum
8. $x^{(j)} = x^{(j-1)} + (\Delta t/m) \cdot u^{(j-\frac{1}{2})}$ # Make a full step for position
9. $u^{(j)} = u^{(j-\frac{1}{2})} - (\Delta t/2) \times \nabla U(x^{(j)})$ # Make another half step for momentum
10. End For
11. Set $(x', u') = (x^{(L)}, u^{(L)})$
12. # Evaluate Hamiltonian (potential plus kinetic energies) at start and end of the trajectory
13. Evaluate $H(x^{(0)}, u^{(0)}) = U(u^{(0)}) + \sum_{k=1}^d u_k^{(0)2} / (2m_k)$
14. Evaluate $H(x', u') = U(u') + \sum_{k=1}^d u_k'^2 / (2m_k)$
15. Evaluate $\delta H = H(x', u') - H(x^{(0)}, u^{(0)})$
16. Draw $\alpha \sim \text{Uniform}[0, 1]$
17. If $\alpha < \min\{1, \exp(-\delta H)\}$ then # Metropolis algorithm step
18. $(x_i, u_i) = (x', u')$ # accept
19. Else
20. $(x_i, u_i) = (x_{i-1}, u_{i-1})$ # reject
21. End If
22. End For
23. Return $\{x_i, u_i\}_{i=0}^{N_{\text{samples}}}$

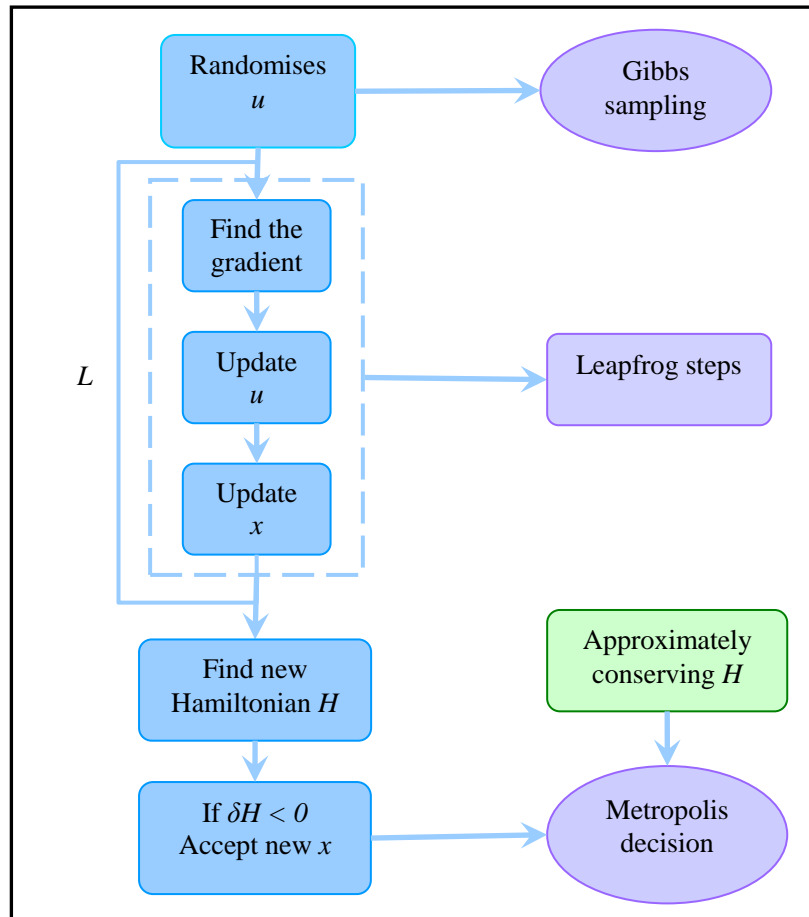


Figure 6.4: The HMC steps algorithm for $k = 1$ iteration.

HMC combines stochastic and deterministic sampling characteristics. The deterministic approach allows large steps in the parameter space to be taken with only a few evaluations of U and the gradient of U . The ability to take large steps is the essential feature of the Hamiltonian method that makes it attractive.

6.3.3 Algorithm Parameter Effect

Choosing the appropriate tuning parameters for MCMC algorithms is a major issue within the MCMC community (Roberts and Rosenthal, 2001). Although some choices can be set via human ingenuity, a long-standing objective in MCMC is to construct these choices and the application of MCMC in an automatic way (Green and Murdoch, 1998). Updating tuning parameters of MCMC algorithms automatically as the algorithm runs is one of the recent advances in the adaptive MCMC algorithm. Carefully designed adaptation for the parameter values during the run can be valid and effective and have potential for a wider set of MCMC applications in the future (Andrieu and Atchadé, 2005; Andrieu and Moulines, 2003; Atchadé and Rosenthal, 2005; Giordani and Kohn, 2010; Haario, Laine, Mira, and Saksman, 2005; Haario,

Saksman, and Tamminen, 2005; Roberts and Rosenthal, 2007; Roberts and Rosenthal, 2009), whereas other naive adaptations can destroy ergodicity (Rosenthal, 2004). Atchadé et al. (2009) provides an introduction to adaptive MCMC and particularly reviews the methodological and theoretical aspects of the method.

There are three parameters to tune while performing HMC, namely: m , stepsize Δt , and the number of leapfrog steps L where $\Delta t L$ determines the trajectory length to be taken. A careful selection of stepsizes is crucial for HMC to perform well and provide accurate estimates with the best choices that would be obtained by monitoring the acceptance rate and the efficiency of the resulting chains. Large stepsizes will bring errors into the simulation and the energy H at the end of the trajectory will differ significantly from the energy at the start of the trajectory that may result in repeated rejections caused by the Metropolis step, hence the acceptance rate declines. Therefore, Δt should be chosen small enough to carry out the Hamiltonian dynamics correctly, so that the average rejection rate resulting from the Metropolis step is not too large but not too small that effective exploration of the high probability region is hindered. Theoretical analysis of optimal stepsizes Δt and acceptance rates is presented in Beskos et al. (2010).

For each dynamic progression in the deterministic step, the number of leapfrog steps L , the second parameter to be tuned, should be large enough to take the walker far from the starting point. Too many steps may increase the computational cost, and a small number of steps may result in correlation between the samples. Thus, a trade-off between short and long trajectories is needed for the studied problem. The values for the third parameter, the mass vector m , can be selected where an element m_i is 1 for each i if the elements are of comparable scale. This can be guaranteed by normalising the uncertain parameters in the beginning. It turns out that selecting the masses m_i is equivalent to selecting different stepsizes Δt in different dimensions of the parameter (Choo, 2000; Neal, 2011).

6.3.4 Algorithm Parameter Selection

In evaluating how well the parameters, stepsize Δt and the number of leapfrog steps L , are selected, preliminary runs need to be performed and trial values have to be selected. However, this may be time-consuming and impractical particularly in higher dimensions. Adaptive stepsize is usually recommended over the constant stepsize (Neal, 2011; Roberts and Rosenthal, 2007; Roberts and Rosenthal, 2009) where the

dynamics evolve with different speeds in different regions of the trajectory based on the characteristics of the region such as the shape and orientation of the potential energy function. This reduces the computational cost and simulation errors significantly (Choo, 2000). The masses could be considered the extra degrees of freedom that allow leapfrog updates with different stepsizes for different components whilst still simulating Hamiltonian dynamics and approximately keeping H constant. This is achieved by substituting $\tilde{u} \equiv u/\sqrt{m}$ in the leapfrog updates in Eqs. (6.17)– (6.19), resulting in Eqs. (6.23)– (6.25).

$$\tilde{u}\left(t + \frac{\Delta t}{2}\right) = \tilde{u}(t) - \left(\frac{\Delta t}{2\sqrt{m}}\right) \times \nabla U(x(t)) \quad \text{Eq. (6.23)}$$

$$x(t + \Delta t) = x(t) + \left(\frac{\Delta t}{\sqrt{m}}\right) \cdot \tilde{u}\left(t + \frac{\Delta t}{2}\right) \quad \text{Eq. (6.24)}$$

$$\tilde{u}(t + \Delta t) = \tilde{u}\left(t + \frac{\Delta t}{2}\right) - \left(\frac{\Delta t}{2\sqrt{m}}\right) \times \nabla U(x(t + \Delta t)) \quad \text{Eq. (6.25)}$$

Neal (1996a) optimised stepsizes for a quadratic Hamiltonian by estimating the local second derivative of the potential energy. He concluded that H stays bounded under the leapfrog discretisation if $\Delta t < 2\sigma$ while H diverges if a stepsize $\Delta t > 2\sigma$ is used. Choo (2000) generalised this result for non-quadratic H and suggested taking into account the local length scales of the distribution based on the width of potential energy bowl in the direction i , Δt_i as in Eq. (6.26) in which η represents the stepsize adjustment factor. The properties outlined in Section 6.2.1 are satisfied for the Markov chain resulting from simulating $H(x,u)$ under the new updates (Choo, 2000).

$$\Delta t_i \approx \eta \times \left(\frac{\partial^2 U}{\partial^2 x_i}\right)^{-\frac{1}{2}} \quad \text{Eq. (6.26)}$$

In this thesis, we estimated the stepsizes by the simple equation $\Delta t_i \approx \eta \times \beta_i$ where β_i is the averaged distance for each dimension from the current values of the parameters and η is the stepsize adjustment factor adjusted during iterations while taking into account the appropriate scale in various dimensions. The setting of the stepsizes is problem-dependent. Neal (1996a) makes a selection depending on the current location in a neural network model application where stepsizes have not been updated during a leapfrog trajectory course and thus the leapfrog trajectories are reversible.

Once a proper stepsize is established, the number of leapfrog steps can be selected based on experimenting with different stepsizes and monitoring the autocorrelation and

efficiency of the chain. L may be randomly selected from a discrete uniform distribution from 1 to some preselected L_{max} to avoid getting into a resonance condition (Mackenze, 1989). The resonance condition occurs when trajectories in the leapfrog loop circle the same closed trajectory for a number of cycles, but it seldom happens in practice. Autocorrelation is one of the diagnostic tools often used to analyse MCMC methods performance and it is discussed in Section 6.3.6.

A variety of Hamiltonian based Monte Carlo methods discussed in significantly more detail and background material can be found in Andrieu et al. (2003), Chen et al. (2000), Cheung and Beck (2009), Choo (2000), Goldstein (1980), Hanson (2001), Leimkuhler and Reich (2004), MacKay (2003), Murray (2007), Neal (1993), Rasmussen (2003), Rossberg (1983), and Torby (1984).

6.3.5 *Hamiltonian Monte Carlo Convergence*

For HMC to converge asymptotically to an invariant unique distribution, it should be ergodic (aperiodic and irreducible) which relies upon the Hamiltonian, the leapfrog trajectory length, and the stepsize adjustment factor. Periodicity in the majority of complex nonlinear problems is improbable. Varying the stepsize adjustment factor randomly over a small range should eliminate any periodicities (Choo, 2000; Mackenze, 1989). Satisfying the irreducibility constraint is based on the exact shape of the potential energy surface in the underlying problem. If for finite values of x , the potential energy does not become infinite, HMC should be irreducible. A positive probability exists for obtaining large enough kinetic energy from momentum replacement to avoid entrapment and to explore other areas of the parameter space when sampling near a local minimum. However, this can fail if the local minimum is located in very steep regions and bounded by walls of infinite potential minimum energy or very steep gradients such as presented in the petroleum example in Section 7.2.2. Thus, we will have irreducibility. Choo (2000) noted that, even though HMC only simulates Hamiltonian dynamics approximately, it is possible that a finite potential may be entrapped like an infinite one. Even so, he argued that we can regard HMC simulation of the Hamiltonian dynamics as a justification to consider that the irreducibility condition holds. For the history matching problem the Hamiltonian is finite for finite values of model parameters x . In this case we do not face the convergence problem or a very complex surface. In the first case we may adjust the prior ranges for the parameters to feasible realistic ones or consider another parameterisation approach like

the ones demonstrated in Section 2.5. In the later case some MCMC methods, called Population MCMC, with the coupled HMC principles could be utilised to illustrate how to handle such cases. This is motivated in IC Fault model example in Section 6.2.2.

6.3.6 Diagnostic Tools for Evaluating HMC Sampling Performance

Diagnostic tools are usually used for evaluating the sampling performance of the chains drawn by MCMC methods. In this thesis, a few diagnostic tools are selected to carry out tests, namely the autocorrelation length, and the power spectrum to estimate efficiency. The methods are illustrated in Cowles and Carlin (1996), Hajian (2007), and Wolff (2004). Adapted MATLAB and R packages (Coda) are used to conduct the tests.

6.3.6.1 Autocorrelation Function (ACF)

The autocorrelation (Box and Jenkins, 1976) function of a random process describes the general dependence between the values of the samples at different points in time, as a function of the two times or the time difference. The autocorrelation function (ACF) can be used to check correlations of successive steps of a chain. The autocorrelation function at lag l is defined by Eq. (6.27).

$$\begin{aligned} \rho(l) &= \frac{\text{Cov}(x_i, x_{i+l})}{\text{Var}(x_i)} \\ &= \frac{\sum_{i=1}^{n-l} (x_i - \bar{x})(x_{i+l} - \bar{x})}{\sum_{i=1}^n (x_i - \bar{x})^2} \end{aligned} \quad \text{Eq. (6.27)}$$

High autocorrelations within chains signify slow mixing and slow convergence. In this case thinning out a chain with high autocorrelations before calculating summary statistics is a useful common practice since a thinned chain may hold most of the information, but occupy less space in memory. An ideal sampler with no correlations between successive components of the chain will have an autocorrelation which wanes rapidly.

6.3.6.2 Autocorrelation Length

The autocorrelation length of a chain is described in Eq. (6.28) in which l_{max} represents the maximum lag. The $\rho(l)$ is aggregated until the length of the series is less than l_{max} which is identified by monitoring noisy autocorrelation to limit the errors in the

autocorrelation length estimate. Achieving a value $L = 1$ indicates an ideal sampler, while higher values represent more relationships between the data.

$$L(l) = 1 + 2 \sum_{l=1}^{l_{\max}} \rho(l) \quad \text{Eq. (6.28)}$$

6.3.6.3 Power Spectral Analysis

The power spectrum of the finite MCMC chain is described by Eq. (6.29) in which α_k are ratios between the discrete (inverse) Fourier transform of the chain and the square root of the number of samples in the MCMC chain, N . Here, $k = 2\pi j/N$ for $j = 1, \dots, (N/2 - 1)$.

$$P(k) = |\alpha_k|^2 \quad \text{Eq. (6.29)}$$

MCMC chains often have correlations on small scales and as a result their power spectra will have curvature on small scale levels (Dunkley et al., 2005). Achieving a flat power spectrum indicates an ideal sampler (Hajian, 2007). The spectral density at frequency zero, $P(k)$ at $k = 0$, estimates the variance of sample mean sample, σ_x , in Eq. (6.30).

$$\sigma_x^2(N) = \frac{P_0}{N}, P_0 = P(k=0) \quad \text{Eq. (6.30)}$$

6.3.6.4 Efficiency

The statistical efficiency of an MCMC chain is described by Eq. (6.31), in which σ_0 is the variance of the target distribution and σ_x is the variance of the sample mean from the MCMC chain. It is the ratio of the number of independent samples drawn from the target distribution to the number of MCMC iterations needed to attain the equivalent variance in an estimated quantity of interest.

$$E = \lim_{N \rightarrow \infty} \frac{\sigma_0^2/N}{\sigma_x^2(N)} \quad \text{Eq. (6.31)}$$

Substituting Eq. (6.30) in Eq. (6.31) yields the efficiency of an MCMC chain in terms of P_0 as in Eq. (6.32). To determine the fraction that makes MCMC chain longer than an ideal chain, E^{-1} can be computed.

$$E = \frac{\sigma_0^2}{P_0} \quad \text{Eq. (6.32)}$$

6.4 Numerical Experiments

The performance of HMC technique in sampling posterior distributions is investigated on two numerical examples where an analytical function can be visualised and the gradient can be readily evaluated.

6.4.1 Numerical Example Test 1

We consider sampling from two variables target distribution defined by Eq. (6.33) (Kaipio and Somersalo, 2005) with HMC algorithm.

$$\pi(p_1, p_2) \propto \exp\left(-10(p_1^2 - p_2^2)^2 - \left(p_2 - \frac{1}{4}\right)^4\right) \quad \text{Eq. (6.33)}$$

Figure 6.5(a) shows the scaled contour of the target density where the curved white area represents the high probability region. We consider the (p_1, p_2) position parameters and the augmented corresponding momentum variables are drawn from Gaussian distribution with means of zero, standard deviations of one and zero correlation. Figure 6.5(b) shows a sequence of four simulated leapfrog trajectories where position versus momentum for a single parameter, p_1 , is depicted. The chain converges from an initial condition that is not close to the distinctive set of the target distribution. The first trajectory turns out to finish in a state closer to the bottom of the potential energy landscape. As the potential energy U is smaller, the kinetic energy $K = u^2/2$ is essentially larger than it was in the beginning of the leapfrog trajectory. As soon as the momentum is randomised ahead of the second trajectory, the kinetic energy reaches greatly smaller value. Following the simulation of the fourth leapfrog trajectory, the sampled state lies in an area typical of the target density. HMC samples drawn are shown in Figure 6.5(c) and the plot clearly shows that the samples came from the high probability region in the target density.

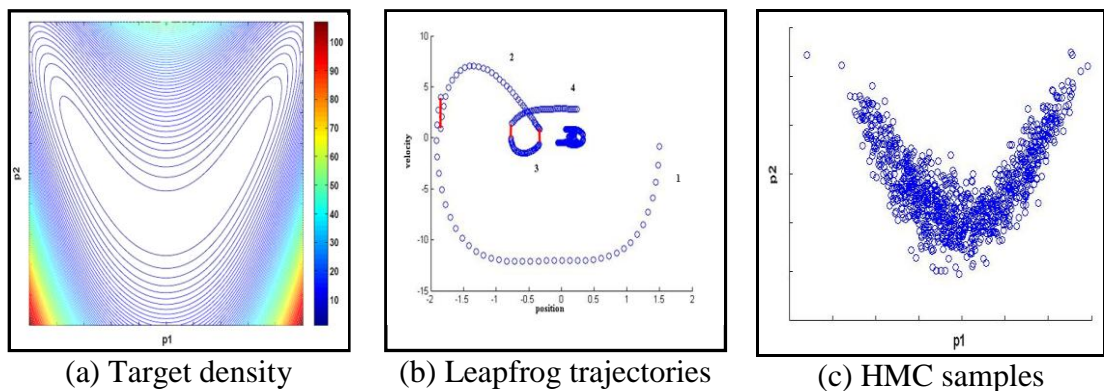


Figure 6.5: 3D plot for the probability distribution of interest $\pi(p_1, p_2)$ (a), position versus its augmented momentum coordinate (b), and HMC samples (c)

6.4.2 Numerical Example Test 2

Here we consider sampling with the HMC algorithm from the distribution given by Eq. (6.34) (Dunkley et al., 2005; Hajian, 2007). The target distribution is an example of a thin, curved, non-Gaussian distribution.

$$\pi(x, y) = \frac{\left((2x-1)^2 + (2y-1)^2 - 1 \right)^2}{8\sigma_1^2} + \frac{(2y-1)^2}{2\sigma_2^2} \tag{Eq. (6.34)}$$

The 3D plot of the negative log probability distribution is shown in Figure 6.6(a) in which $\sigma_1 = 0.035$, $\sigma_2 = \sqrt{5\sigma_1}$ and x, y are scaled in the range $[0, 1]$ in Eq. (6.34) (Figure 6.6(b) and (c) are in the original scale). Sampling from non-Gaussian or curved distributions can be very difficult and inefficient with common Metropolis-Hastings MCMC algorithms (Dunkley et al., 2005). That is because for a number of dimensions $d \geq 2$ a potential problem arises for these distributions in which the region of high probability is elongated and curved as shown in Figure 6.6(c). In this plot, the connected region of high probability has the shape of a thin crescent. This is also illustrated in the magnified 3D plot in Figure 6.6(b).

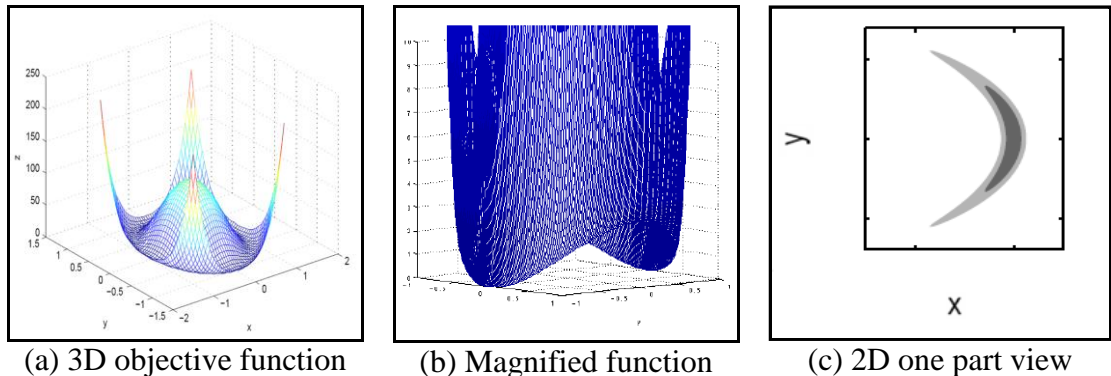


Figure 6.6: 3D plot for the probability distribution of interest $\pi(x,y)$ (a), with a 3D plot magnified in (b), and the 2D plot for one side of distribution (c)

The leapfrog algorithm is illustrated in Figure 6.7(a) and (b). Starting from the state $(0, -1)$, the three coloured curves formed by crosses or dots represent the first three trajectories generated by the Hamiltonian dynamics coloured by red, green and blue. The circles at the end of each trajectory show the end points. Each trajectory consists of $\tau = 40$ leapfrog steps with stepsize $\epsilon = 0.025$. These steps are indicated by the crosses on each trajectory. Similarly to the previous example, after each trajectory, the momentum is randomised, the arrows indicate the direction of the gradient where the potential energy U is smaller as in the point numbered 2 in Figure 6.7(a), and hence, the kinetic energy $K(u) = \sum_{i=1}^d u_i^2 / 2$ is larger than it was at the start of the next trajectory to

preserve the Hamiltonian H . When the momentum is randomised before the third trajectory, its kinetic energy becomes much smaller (from the stochastic nature) and so converges. As a contrast in Figure 6.7(b) the Hamiltonian H is not approximately conserved for the rejected state indicated by ‘R’. In Figure 6.7(a) all trajectories are accepted, while in Figure 6.7(b) two are accepted and one is rejected.

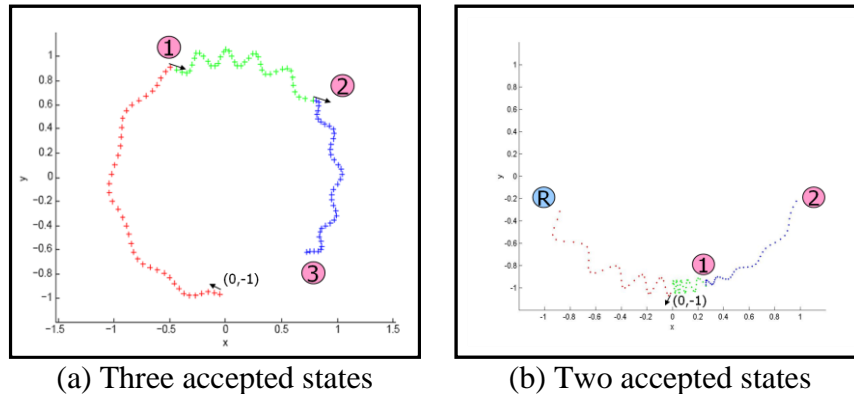


Figure 6.7: HMC method uses leapfrog simulation where in (a) three accepted states are drawn and in (b) two accepted states are drawn

HMC sampling from this distribution is shown in Figure 6.8(a) where the first 1000 samples are depicted in 3D. The gradient is calculated at every point in the parameter space directly from the analytical function. The widths are chosen such that $\sigma_2^2/\sigma_1 = 5$. Figure 6.8(b) and (c) show the same samples on a larger scale and 2D plot.

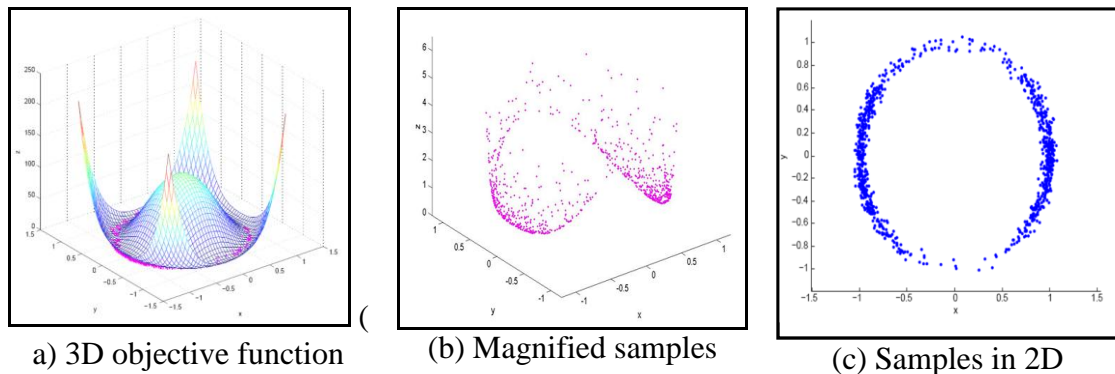


Figure 6.8: Samples drawn in (a) 3D plot, (b) 3D larger scale plot, and (c) 2D plot

6.5 Computation of the Gradient

In theory, gradients are a much richer source of information and only cost a constant multiple of the computer time needed to calculate a function (Bischof and Bücker, 2000; Murray, 2007). However, the difficulty with simulation tasks is that inaccuracies accumulate quickly unless time is discretised very finely, which may lead to a large computational cost.

Implementing Hamiltonian Monte Carlo requires us to compute the gradient of the negative logarithm of the posterior distribution, which is the sum of the misfit and the negative log of the prior. In general, this term cannot be computed analytically, so numerical approaches have to be used. There are two approaches which can be followed to compute this term. If the gradients of the solution with respect to uncertain variables are available, for example from an adjoint code, then this term can be computed directly. Computing the gradient using finite differences for instance requires either d or $2d$ evaluations of U , where d represents the dimensionality of the uncertainty parameters. Alternatively, it can be computed less accurately from a proxy to approximate the misfit surface obtained from a fast interpolation model instead of the exact one based on the flow simulation result. The multi-layer perceptron (MLP) neural network is one of the efficient interpolation models that could be used (Christie et. al., 2006).

In the application presented in the thesis, we use a General Regression Neural Network (GRNN) (Specht, 1991; Nadaraya, 1964; Watson, 1964) with Gaussian Kernels to approximate the misfit surface, and then compute the gradient from the approximated surface. It is important to note that there are some restrictions on updating the gradient surface to ensure consistency with the MCMC assumptions (Rosenthal, 2007, 2011). This is discussed in significant detail with illustrative examples in Gilks et al. (1998) and Roberts and Rosenthal (2007, 2009) and fall under adaptive MCMC topics. A theoretical review of a machine learning approach, the general regression neural network, is provided next to develop an approximation that resembles the misfit surface, and to estimate the gradients.

In the second equation in the leapfrog algorithm which involves computing the gradient of the negative logarithm of the posterior distribution, Eq. (6.17), is composed of the misfit term and the minus log (prior) term from the relation Eq. (6.35). Using an auxiliary distribution that mimics the negative logarithm of the posterior distribution is fast to compute. The leapfrogs can then be taken with no major additional cost of time.

$$\begin{aligned} U(x) &= -\log(\text{likelihood}) - \log(\text{prior}) \\ &= M(x) - \log(\text{prior}) \end{aligned} \quad \text{Eq. (6.35)}$$

Substituting in Eq. (6.17) we get Eq. (6.37) in which $M(x)$ is the misfit at x .

$$u\left(t + \frac{\Delta t}{2}\right) = u(t) - (\Delta t/2) \times (\nabla M(x(t)) - \nabla \log(\text{prior})) \quad \text{Eq. (6.36)}$$

$$u(t + \Delta t) = u\left(t + \frac{\Delta t}{2}\right) - (\Delta t/2) \times (\nabla M(x(t + \Delta t)) - \nabla \log(\text{prior})) \text{ Eq. (6.37)}$$

6.5.1 Machine Learning Background

The machine learning approach includes a wide selection of data-driven algorithms based on the concept of learning from examples (supervised learning). The so called, lazy learning approach provides a way to modify the model by means of continuous adaptive training once more information becomes available.

In supervised learning, systems are built to compute the output from inputs without explicit programming but rather using examples as shown in Figure 6.9. This implies that our machines are trained instead of programmed as they usually are to perform a specific task.

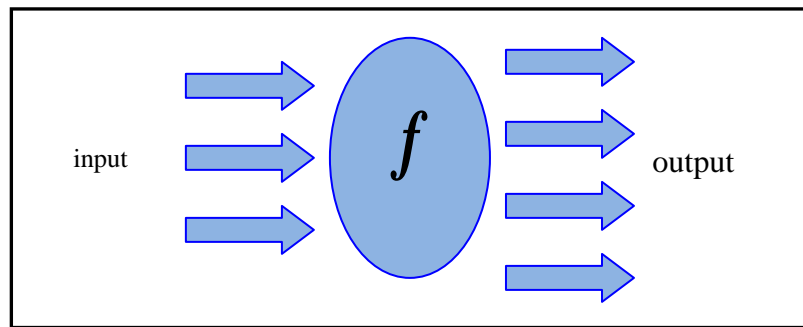


Figure 6.9: In the learning from example scheme, we learn a function f from input-output pairs (x_i, M_i) called the training set.

Training is the process whereby we choose the best function that describes the relation between each input (model) and the corresponding output (misfit). By *best function*, we mean a function that not only performs well on the training data (interpolation), but also generalises well on unseen input data (extrapolation). More formally, if the data are in the form (x_i, M_i) , $i = 1, \dots, N$ the learning from examples scheme is to fit a multivariate function to the data so that this function predicts on unseen data. That is the entire space.

General Regression Neural Networks (GRNN) is one of the machine learning approaches that was tested previously to guide the stochastic sampling algorithms in assessing the uncertainty of predictions, and it has shown reasonable results (Demyanov, 2006).

6.5.2 General Regression Neural Network (GRNN)

General Regression Neural Network (GRNN) is Donald Specht's term (Specht, 1991), for Nadaraya-Watson kernel regression (NWKRE) (Nadaraya, 1964; Watson, 1964). It was recreated in the neural network literature by Schiøler and Hartmann (Tomandl and Schober, 2001) (kernels are also called parzen windows, and they are usually probability density functions). It is based on established statistical principles and converges with an increasing number of samples asymptotically to the optimal regression surface.

Consider a nonlinear regression problem, described by a model whose observable output M_i in response to an input vector x_i is defined by

$$M_i = f(x_i) + \varepsilon_i \quad \text{Eq. (6.38)}$$

where $f(x_i)$ is a smooth function, $\varepsilon_i \sim \mathcal{N}(0, \sigma_{x_i}^2)$, and M_i is the i^{th} misfit defined by:

$$M_i = \sum_{t=1}^T \frac{(\text{obs} - \text{sim}(x_i))_t^2}{2\sigma^2}; \forall i \quad \text{Eq. (6.39)}$$

The aim is to construct the underlying function, $f(x_i)$, given the training data $\{(x_i, M_i)\}_{i=1}^N$. The key idea is to compute a linear combination (in M_i) of local kernel functions centred on the training data. The Nadaraya-Watson kernel regression NWKRE estimator is then given by Eq. (6.40).

$$\hat{M}(x) = \sum_{i=1}^N M_i \frac{K(x_i, x)}{\sum_{i=1}^N K(x_i, x)} \quad \text{Eq. (6.40)}$$

where $\hat{M}(x)$ is the kernel density estimate at x and M_i are the weights (misfits). The most commonly used kernel function is the Gaussian kernel in Eq. (6.41).

$$K(x_i, x) = \exp\left(-\frac{(x_i - x)^2}{2\sigma^2}\right) \quad \text{Eq. (6.41)}$$

σ is called the smoothing parameter or bandwidth which represents the width of the kernel as in Figure 6.10.

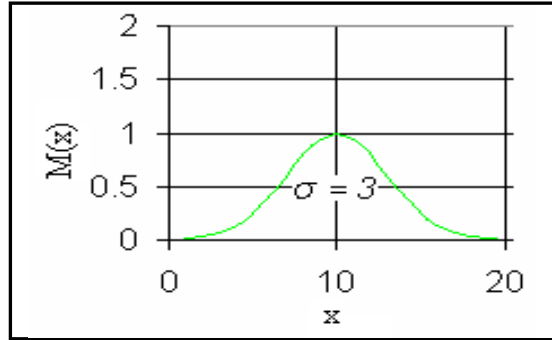


Figure 6.10: Gaussian kernel centred at 10

NWKRE can be rewritten as in Eq. (6.42).

$$\hat{M}(x) = \frac{\sum_{i=1}^N M_i \exp\left(-\frac{(x_i - x)^2}{2\sigma^2}\right)}{\sum_{i=1}^N \exp\left(-\frac{(x_i - x)^2}{2\sigma^2}\right)} = \frac{U}{V} \quad \text{Eq. (6.42)}$$

6.5.2.1 *NWKRE as a Neural Network: GRNN*

GRNN consists of four layers: input, pattern, summation and an output layer. Figure 6.11 is a graphical representation of GRNN in terms of a neural network. The input layer transfers an input signal vector x , into the next pattern layer. The number of neurons (kernels) in the pattern layer is equal to the number of training samples, N . For an input vector all pattern layer neurons compute the Euclidian distance between the input vector and the corresponding neuron location. These Gaussian kernel distances are the activation function, and passed out to the next summation layer. The summation layer consists of two neurons that calculate the numerator and denominator respectively in Eq. (6.42). Each of these neurons computes a weighted sum of the output from a previous layer. The weights correspond to the arrows between the neurons. Each arrow from the pattern layer neuron to the nominator neuron is a target value associated with the corresponding neuron location. All arrows between pattern layer neurons and a denominator neuron are equal to a unit. The output layer neuron carries out the division operation and transfers the output value.

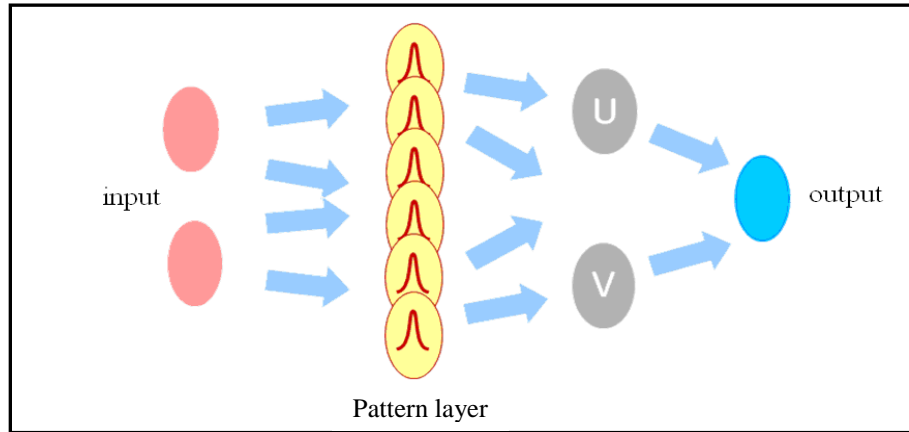


Figure 6.11: Pattern layer: consists of N kernels – one for every available data point

In Figure 6.12 the GRNN estimator is shown in red based on $N = 3$ samples with kernel width $\sigma = 3$ computed using Eq. (6.43).

$$\hat{M}(x) = M_1 \frac{\exp\left(-\frac{(x_1-x)^2}{2\sigma^2}\right)}{\sum_{i=1}^3 \exp\left(-\frac{(x_i-x)^2}{2\sigma^2}\right)} + M_2 \frac{\exp\left(-\frac{(x_2-x)^2}{2\sigma^2}\right)}{\sum_{i=1}^3 \exp\left(-\frac{(x_i-x)^2}{2\sigma^2}\right)} + M_3 \frac{\exp\left(-\frac{(x_3-x)^2}{2\sigma^2}\right)}{\sum_{i=1}^3 \exp\left(-\frac{(x_i-x)^2}{2\sigma^2}\right)} \quad \text{Eq. (6.43)}$$

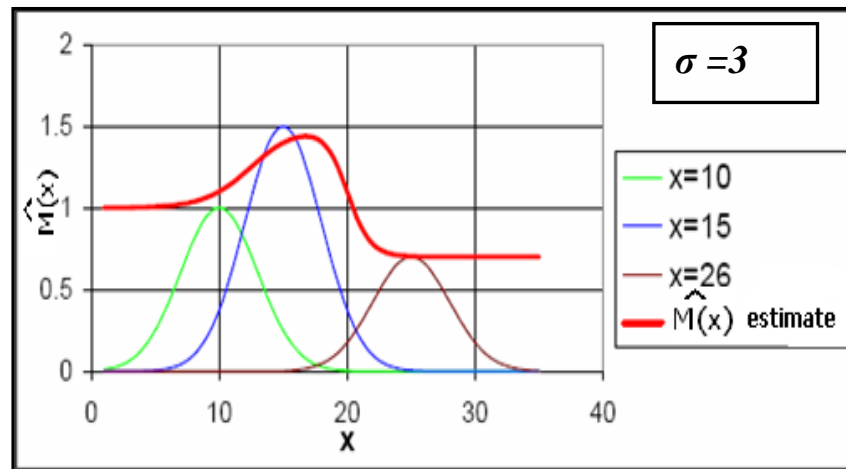


Figure 6.12: GRNN estimate based on $N = 3$ samples with kernel width $\sigma = 3$

6.5.2.2 Kernel Width Optimisation

The kernel width σ is a very crucial parameter as it controls the smoothness of the estimate which in turn determines the smoothness of the boundaries (this affects the probability of error).

- As σ decreases towards 0, the number of modes increases to the number of data points and the kernel density estimator is very noisy, as we have a sum of delta functions (very spiky approximation) as in Figure 6.13(a).

- As σ increases towards one (depends on the scale of the problem), the number of modes drops to 1, so that any interesting structure has been smeared away and the kernel density estimator displays a unimodal pattern, as we have a sum of constant functions (approximated by a constant equal to the sample mean of the observations). Figure 6.13(c) shows the GRNN approximation using kernel width = 0.9.
- In between we get approximations that are gradually smoother as shown in Figure 6.13(b).

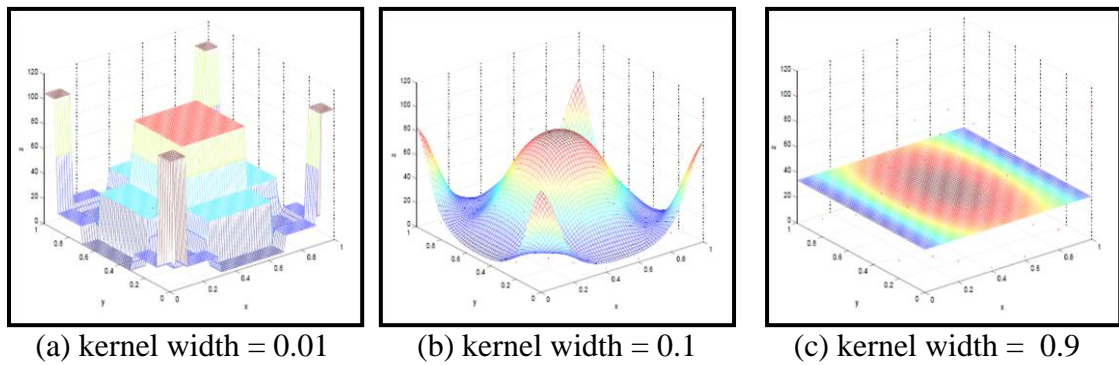


Figure 6.13: Different kernel widths change controlling the smoothness of the surface

There are several ways to tune the kernel widths parameter σ_i (in the case of the anisotropic kernels). These widths are usually chosen by cross-validation (leave-one-out) procedures or by more complex methods that might use the gradient (Demyanov, 2006). Cross-validation procedures to choose the optimal parameters are iterative with each iteration costing $O(N^2)$. So adding a new kernel costs $O(N^3)$. The basic idea can be summarised in the following steps

- Leave some data out of the training set (cross-validation set)
- Train with different kernel widths
- Evaluate performance on cross-validation set by calculating the root mean square deviation (RMSD) – the residuals
- Pick best kernel width configuration (σ_{opt}) corresponds to CV_{Error} given by Eq. (6.44) in which \hat{M}_i^{NET} is the estimate of the cross-validation set.

$$CV_{Error} = \min \left\{ \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{M}_i^{NET} - M_i)^2} \right\} \quad \text{Eq. (6.44)}$$

GRNN is a universal approximator for smooth functions, so it should be able to solve any smooth function approximation problem given enough data. The main drawback of

GRNN is that like kernel methods in general, it suffers badly from the curse of dimensionality. A comprehensive study of other kernel width optimisation choices and more sophisticated methods are available in Kanevski and Maignan (2004) which also contains an exhaustive bibliography of the literature.

As illustrated a critical step in the regression is computing the kernel width parameter. In our application, we used a fraction of the average distance between points in each dimension and cross-validation training only in the early stages for reasonable initial guesses.

6.5.3 Gradient of the Regression Surface by the GRNN

Yet, another feature of the GRNN is the ability to compute the gradient of the regression surface directly without the need for further numerical approximation. Considering the selected training data allows the gradient to be defined and computed, then for the special case of the isotropic Gaussian kernel (σ) the gradient is computed by Eq. (6.45).

$$\nabla \hat{M}(x) = \frac{\hat{M}(x)}{\sigma^2} \left[\sum_{i=1}^N (x_i - x) e^{\left(-\frac{(x_i-x)^2}{2\sigma^2}\right)} \left[\frac{M_i}{\sum_{i=1}^N M_i e^{\left(-\frac{(x_i-x)^2}{2\sigma^2}\right)}} - \frac{1}{\sum_{i=1}^N e^{\left(-\frac{(x_i-x)^2}{2\sigma^2}\right)}} \right] \right] \quad \text{Eq. (6.45)}$$

Other variants for anisotropic kernel widths case can be found in Tomandl and Schober (2001).

6.5.4 Lower Boundary of Kernel Width

The lower boundary is designed to guarantee numerical stability for preventing division by zero in Eq. (6.42) and Eq. (6.45) corresponding to the GRNN estimate and its gradient respectively. Each term of the sum has to return a value larger than zero. In the case of the Gaussian kernel Eq. (6.41) leads to the following condition in Eq. (6.46).

$$\exp\left(-\frac{D_{i,\min}^2}{2\sigma_i^2}\right) = \exp\left(-\frac{(x_i - x)_{i,\min}^2}{2\sigma_i^2}\right) \geq \varepsilon \quad \text{Eq. (6.46)}$$

where ε is the smallest number larger than zero (hardware dependent), and $D_{i,min}$ is distance of x_i to its nearest neighbour. Hence, the lower boundary for the i^{th} kernel width in Eq. (6.47) (anisotropic case) holds:

$$\sigma_{i,min} = \sqrt{-\frac{D_{i,min}^2}{2 \ln \varepsilon}} \quad \text{Eq. (6.47)}$$

If the training data include two identical samples x_i and x_j , the lower boundaries $\sigma_{i,min}$ and $\sigma_{j,min}$ become zero. This undesirable behaviour can be prevented by eliminating all double samples.

6.6 HMC Procedure Recap

The sampling procedure for generating multiple history matched models is summarised in the following steps with diagrammatic illustration in Figure 6.14.

1. The HMC algorithm is initialised with a population of an initial set of n_{init} models randomly generated in the search space by a random generator.
2. For each model the forward problem is solved and the relevant misfit value M is obtained. The population of models with their misfits will be used to approximate the surface and obtain the gradients.
3. Train GRNN on the initial population from an exploratory run to get the optimal sigma.
4. Generate a new momentum vector u from the Gaussian distribution $p(u) \propto \exp(-K(u))$. This step is considered as the stochastic part of the algorithm, which ensures the whole phase space is explored.
5. Starting from the current state, perform L leapfrog steps with a stepsize Δt , resulting in the new state $(x(\Delta t L), u(\Delta t L))$
6. Employ the Metropolis rule, make the next sample $(x^{k+1}, u^{k+1}) = (x(\Delta t L), u(\Delta t L))$ with probability $\min(1, \exp(-(H(x(\Delta t L), u(\Delta t L)) - H(x^k, u^k))))$, where k is the iteration number.

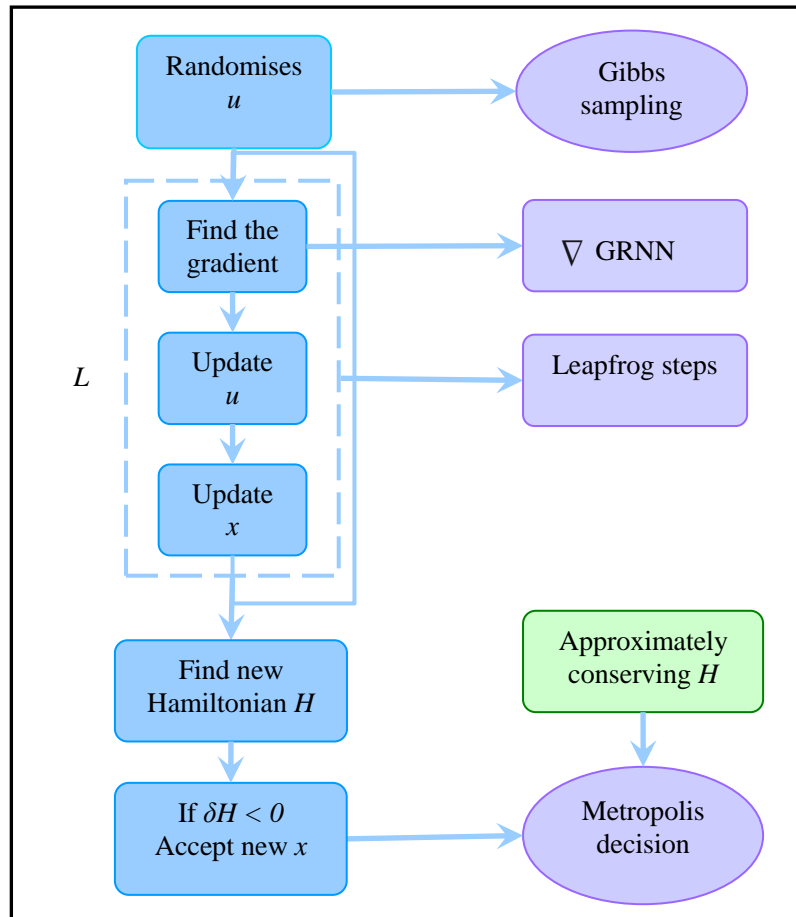


Figure 6.14: The HMC-GRNN algorithm steps for $k = 1$ iteration (1 forward simulation run)

Some common methods for evaluating MCMC convergence is based on observing whether the sample estimation of a certain $E(f(x))$ stabilises for some chosen function f . However, this may provide misleading results because the stabilisation can be a result of the chain of samples being trapped in some region of the parameter space, even though, the Markov chain has not yet converged to the stationary distribution. Another main weakness of this approach is that if it is the only one, is that it is difficult to evaluate how far away the Markov chain is away from reaching stationarity or convergence given that it is not known a priori what value the estimate for $E(f(x))$ should converge to. Thus this approach needs to be complementary to other plots to monitor sampling efficiency by testing the reduction in the value of the misfit as well as the trace plot to monitor whether the chain is exploring different parts in the parameter space.

6.7 Numerical Example Test

We consider the thin, curved, non-Gaussian distribution defined by analytical target density example in Section 6.4.2 to demonstrate and test the algorithm coupled with

gradients estimated with GRNN in this case. The initial training dataset composed of 36 kernels is shown in pink points in Figure 6.15(a), and the optimal kernel width obtained for this dataset is 0.04 using the cross-validation approach as shown in Figure 6.15(b).

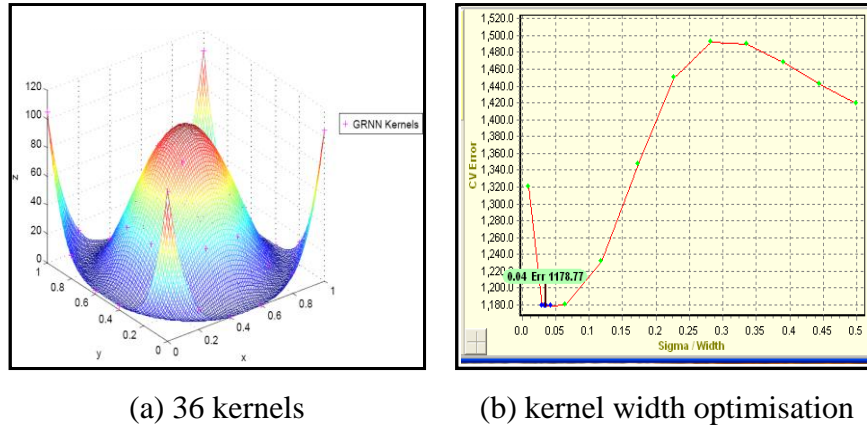


Figure 6.15: (a) Initial training dataset and (b) cross-validation error vs. kernel width

Using a kernel width = 0.04, the approximated GRNN surface is shown in Figure 6.16. The gradient for such surface is not a good approximation to the exact gradients computed at the 36 kernels.

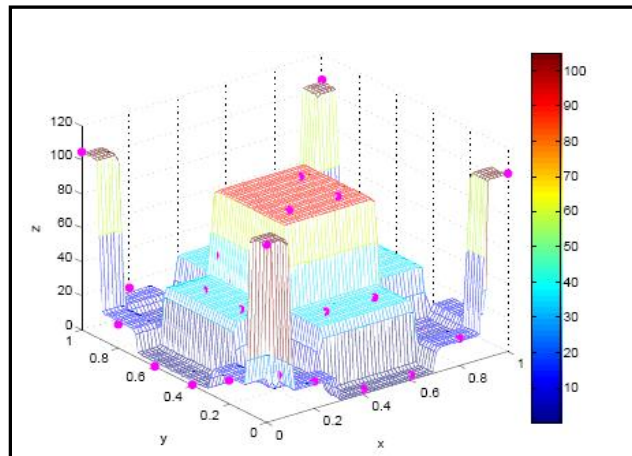
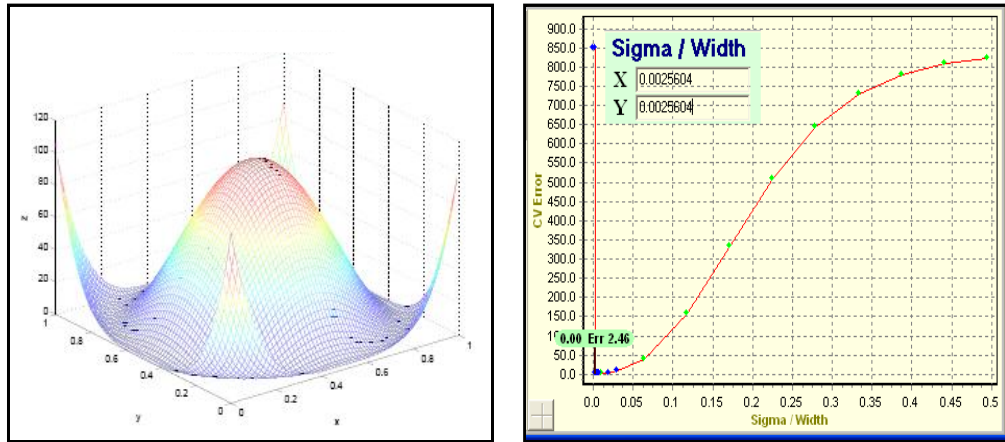


Figure 6.16: GRNN surface using the 36 kernels with kernel width = 0.04

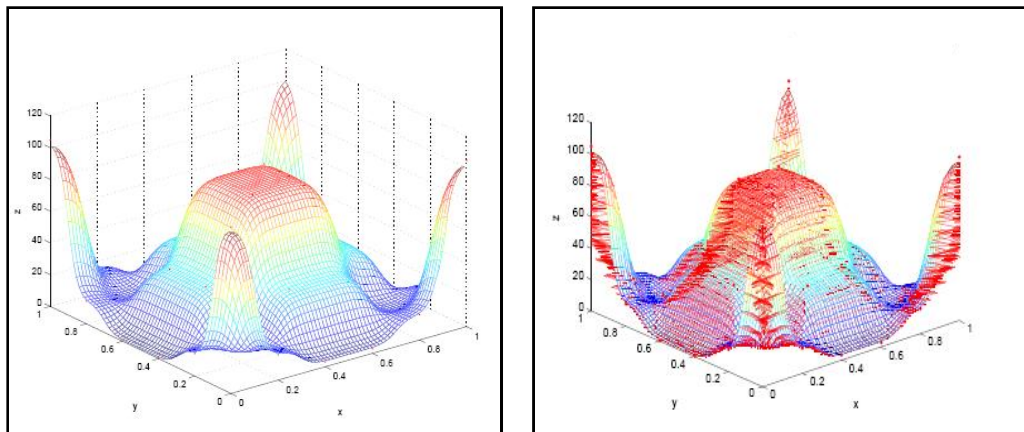
The optimal kernel width using the cross-validation for 1156 kernels is 0.0025604 which gives the GRNN surface shown in Figure 6.17. Since our objective is to carry out as few simulations as possible we used the 36-kernels surface and we increased the value of the kernel width as an alternative approach to obtain a smoother surface as the cross-validation is not suited when the dataset is small. The cross-validation can then

be employed after a number of iterations to check if there is a benefit to using either of the two approaches at any stage in the optimisation.



(a) GRNN surface (b) Optimal kernel width 0.0025604
Figure 6.17: GRNN surface using the 1108 kernels and kernel width = 0.01

Using the 36-kernels with kernel width = 0.07, the GRNN surface with the gradients are shown in Figure 6.18 which is a smoother surface.



(a) GRNN surface using kernel width = 0.07 (b) The scaled GRNN gradients using kernel width = 0.07
Figure 6.18: GRNN surface and the gradient vectors in red

The HMC acceptance rate is 68%. Each HMC trajectory consists of $L = 40$ leapfrog steps each with stepsize $\Delta t = 0.0025$. Figure 6.19(a) shows the first 680 samples drawn from this distribution and Figure 6.19(b) shows the same samples in a larger scale. In a previous study which used the exact gradients, the acceptance rate was 80%. This means that the gradients obtained using the approximated misfit surface by the GRNN give a lower acceptance rate and the better the surface is approximated the higher the acceptance rate is.

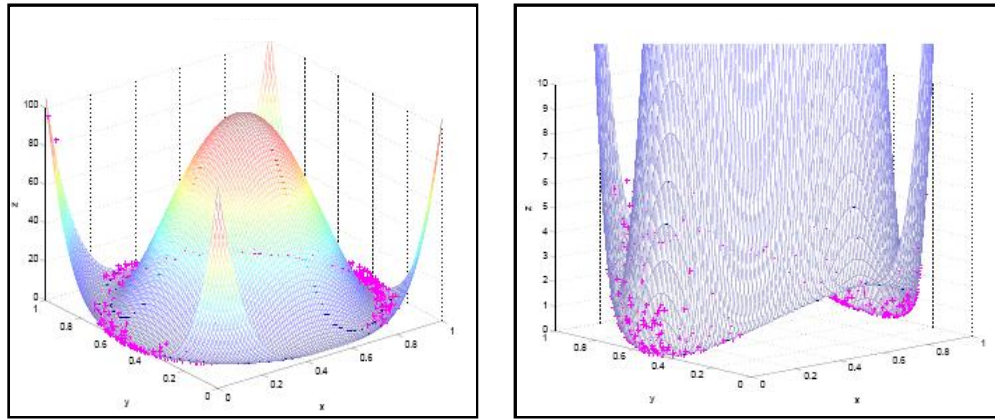


Figure 6.19: HMC samples using grad GRNN shown in pink pluses (left) 3D plot, (right) 3D larger scale

6.8 HMC Sampling Strategies

An experiment with MCMC involves an initial period in which the stepsizes and any control parameters are adjusted. This is followed by a burn-in period in which we check that the simulation converges to the desired distribution. Finally, we record the sample vector occasionally to create a list of samples that we hope are roughly independent samples from the desired distribution. Figure 6.20 shows three possible MCMC strategies to obtain N defined samples where time is represented by horizontal lines and samples by yellow circles (MacKay, 2003).

1. Make one long run consisting of one long burn in period followed by a sampling period obtaining all N samples from it.
2. Make a few medium-length runs with different initial conditions, obtaining some samples from each.
3. Make N short runs, each starting from a different random initial condition, with the only sample that is recorded being the final state of each simulation.

The first strategy has the best chance of achieving convergence. The last strategy may have the advantage that the correlations between the recorded samples are smaller. The middle strategy is popular among MCMC experts (Gilks et al., 1996) because it avoids the waste of discarding burn-in iterations in many simulation runs, while still being able to detect a lack of convergence problems that would not be noticeable from a single run. Note that it is possible to average over dependent states. This will not lead to any bias in the estimates, but estimating the accuracy of the estimate is harder when the states are dependent (MacKay, 2003).

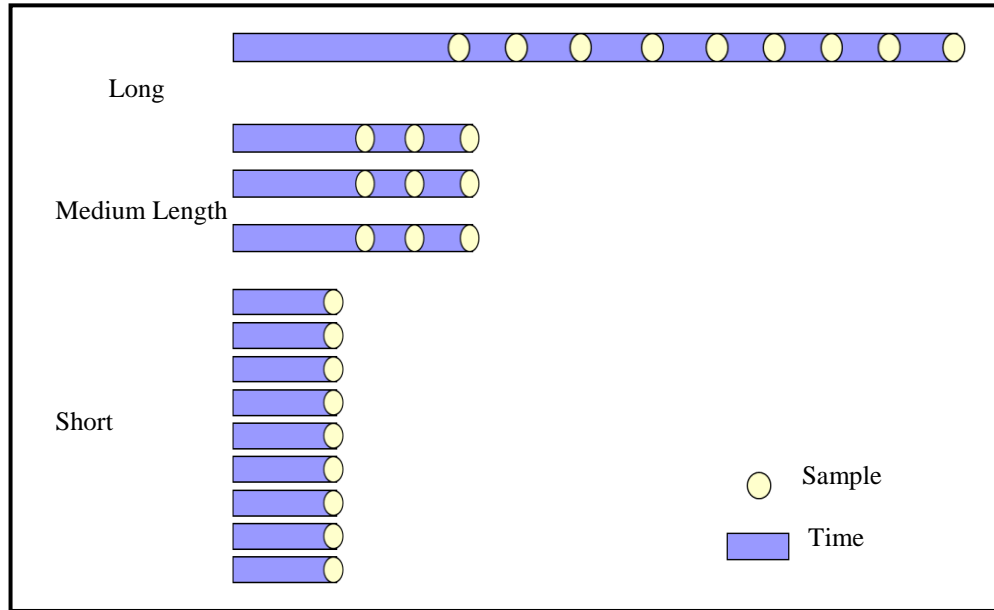


Figure 6.20: Three possible MCMC strategies for obtaining nine samples in a fixed amount of computer time

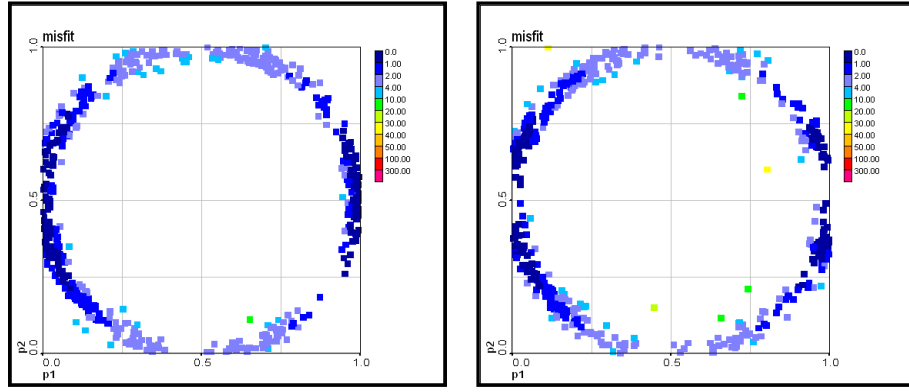
6.8.1 Numerical Example Test 1

We test medium-length runs for the same example in Section 6.7 with GRNN approximation of gradients. The setup used for this case is shown in Table 6.1.

Table 6.1: HMC algorithm parameters

HMC Parameter	Sampling-1	Sampling-2
The method used for generating n_{init}	Latin Hypercube Sampling	Latin Hypercube Sampling
n_{init}	36	36
No. of runs (R)	20	36
Upper limit for no. of leapfrog steps (L_u)	30	30
Lower limit for no. of leapfrog steps (L_l)	10	10

The obtained HMC samples for the two medium length runs are shown in Figure 6.21. The samples are colour coded according to the misfit value. The samples are obtained with the same total number of simulations but with Sampling-2 using shorter chains than Sampling-1. As can be seen in the plot, the results are comparable with no significant differences. Samples obtained with the two sampling lengths obtained good quality models indicated with the blue colour for low misfit values.



(a) Sampling-1

(b) Sampling-2

Figure 6.21: HMC Samples drawn for two runs

6.8.2 Numerical Example Test 2

We test HMC sampling using medium-length runs on the Rastrigin function with two variables. The Rastrigin function is a non-convex, non-linear, multimodal function, and has a large number of local optima. It is used as a performance test problem for optimisation algorithms. The 2-dimensional function was first introduced by Rastrigin and has been generalised by Mühlenbein et al. (1991). Because of the large parameter space and the large number of local minima it is considered to be one of the fairly difficult optimisation problems. The Rastrigin function is defined by Eq. (6.48).

$$\pi(x) \propto \exp\left(-\left(Ad + \sum_{i=1}^d (x_i^2 - A \cos(2\pi x_i))\right)\right) \quad \text{Eq. (6.48)}$$

The parameter range is: $-5.12 \leq x_i \leq 5.12$, $i = 1, 2, \dots, d$. For two independent parameters, Rastrigin's function is defined as in Eq. (6.49) in which $A = 10$. It has a global minimum of zero at zero.

$$\pi(x, y) \propto \exp\left(-\left(20 + x^2 + y^2 - 10(\cos 2\pi x + \cos 2\pi y)\right)\right) \quad \text{Eq. (6.49)}$$

The setup used for this case is shown in Table 6.2. The quality of the samples obtained is colour coded according to the misfit and shown in Figure 6.22 with blue coloured points.

Table 6.2: HMC algorithm parameters

HMC Parameter	Setup
The method used for generating n_{init}	Regular grid
n_{init}	36
No. of runs (R)	36
Upper limit for no. of leapfrog steps (L_u)	30
Lower limit for no. of leapfrog steps (L_l)	10
Number of iterations	30
Total no. of simulations	1080

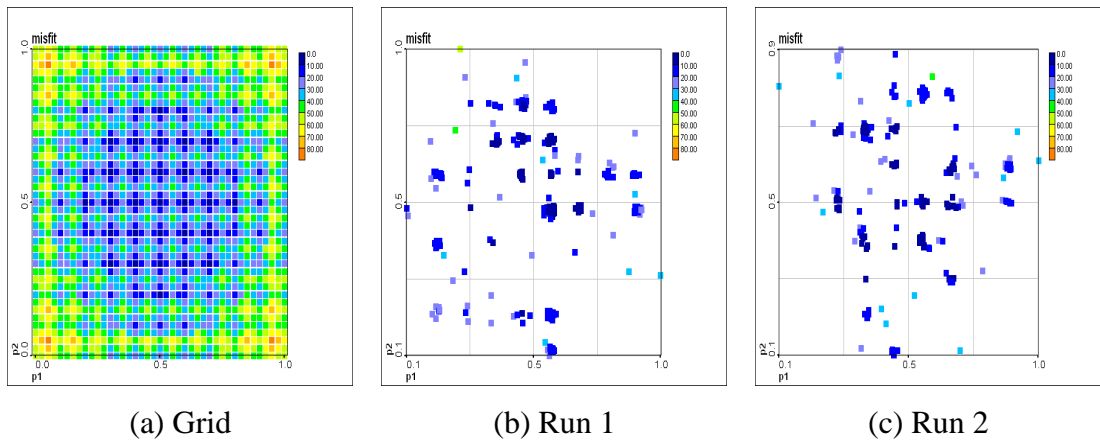


Figure 6.22: HMC samples for Rastrigin function drawn for 2 runs (b) Run 1, and (c) Run 2

6.9 HMC Application in Reservoir Modelling

This section presents the results of producing history matched models used to quantify the uncertainty using HMC for the Teal South petroleum application parameterised with 8 uncertain parameters as presented in Section 4.3.1.

We used 607 days of production data for history matching (20 measurements out of 41 measurements), and the remaining 3 years for all cases were used as prediction data to measure the predictive quality of the history matches. The variance of oil rate is set to 100. Assuming an average error between observed and simulated data of about 75 with 20 points, Eq. (6.50) suggests that a misfit of about 11.8 is reasonable.

$$\text{Misfit} \approx \frac{N \times (\text{average errors})^2}{\sigma_{qop}^2} \quad \text{Eq. (6.50)}$$

The initial population comprised of 50 models generated randomly in parameter space are used to construct the misfit surface with GRNN as shown in Figure 6.23. We ran three HMC chains each with 1350 reservoir model simulations starting from a random

one of the 50 initial points used. We started the HMC sampling using gradients estimated from the GRNN with kernel width adjustment factor equal to 0.02. The leapfrog stepsizes were chosen to be $\Delta t = 0.02 \times \sigma_i$, where $\sigma_i = (C_{ii})^{1/2}$ and C is the covariance matrix. At each HMC iteration, the number of leapfrog steps taken was randomly drawn from a uniform distribution from 10 to 25. Table 6.3 represents the setup of HMC runs. The best misfits obtained were 11.18, 12.55, and 12.1.

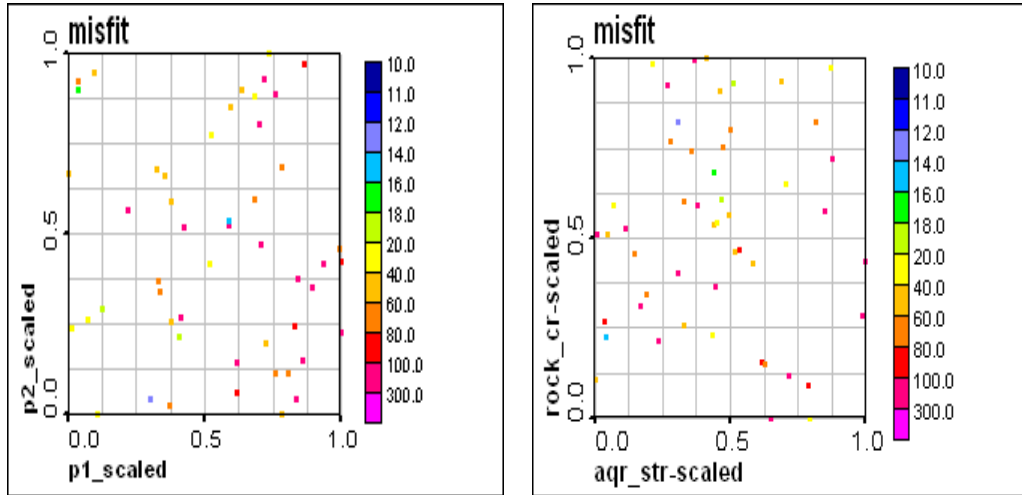


Figure 6.23: Initial population composed of 50 models generated randomly

Table 6.3: HMC algorithm parameters

The HMC Strategy	HMC-Long Run
n_{init}	50
No. of independent runs	3
Upper limit for no. of leapfrog steps (L_u)	25
Lower limit for no. of leapfrog steps (L_l)	10
Stepsize adjustment factor (η)	0.02
σ_{GRNN} adjustment factor	0.80
Total number of simulations	1350

All samples generated for the three runs starting from different location are shown in 2D projections in Figure 6.24 where the crosses represent the initial starting points. HMC samples are shown in Figure 6.25. Finally, the HMC samples used for the forecast period to estimate oil rate are depicted in Figure 6.26. The Bayesian credible intervals are shown in Figures 6.27, 6.28, and 6.29 which captures the truth value. The ranges are narrower than what we have seen in the Section 4.3.1 results for PSO. The reason is that here we used more observed data points. Thus, adding more information restricts

and reduces the uncertainty in the estimates. Chapter 7 shows more tests on the study in comparison with other stochastic methods.

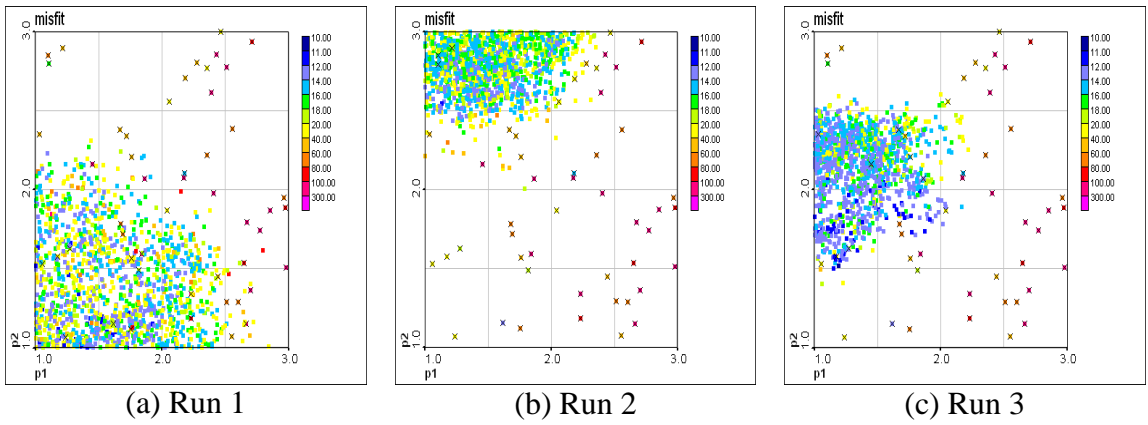


Figure 6.24: All models generated

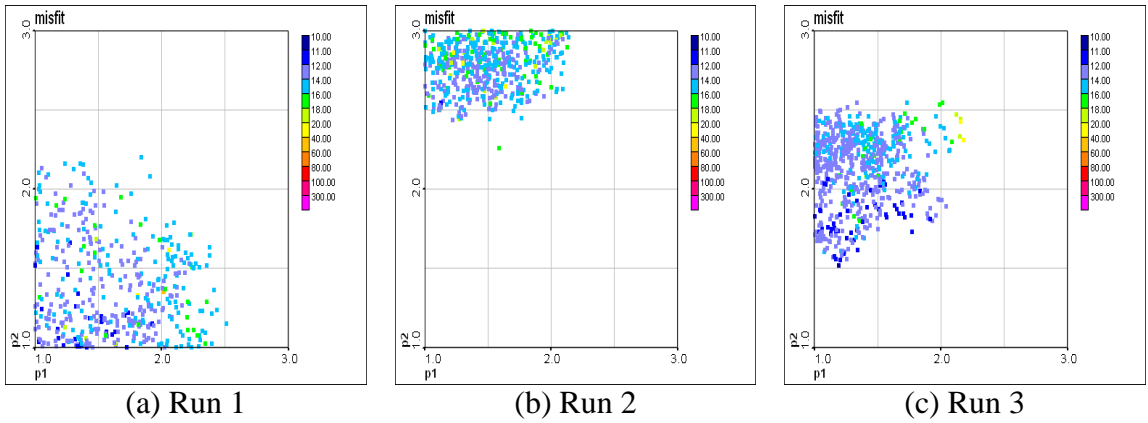


Figure 6.25: HMC samples generated

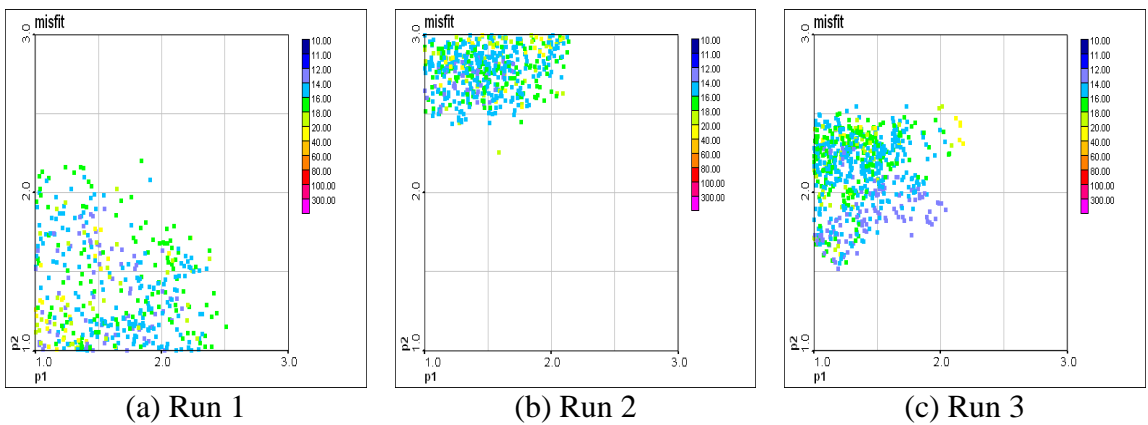


Figure 6.26: HMC samples generated – forecast period

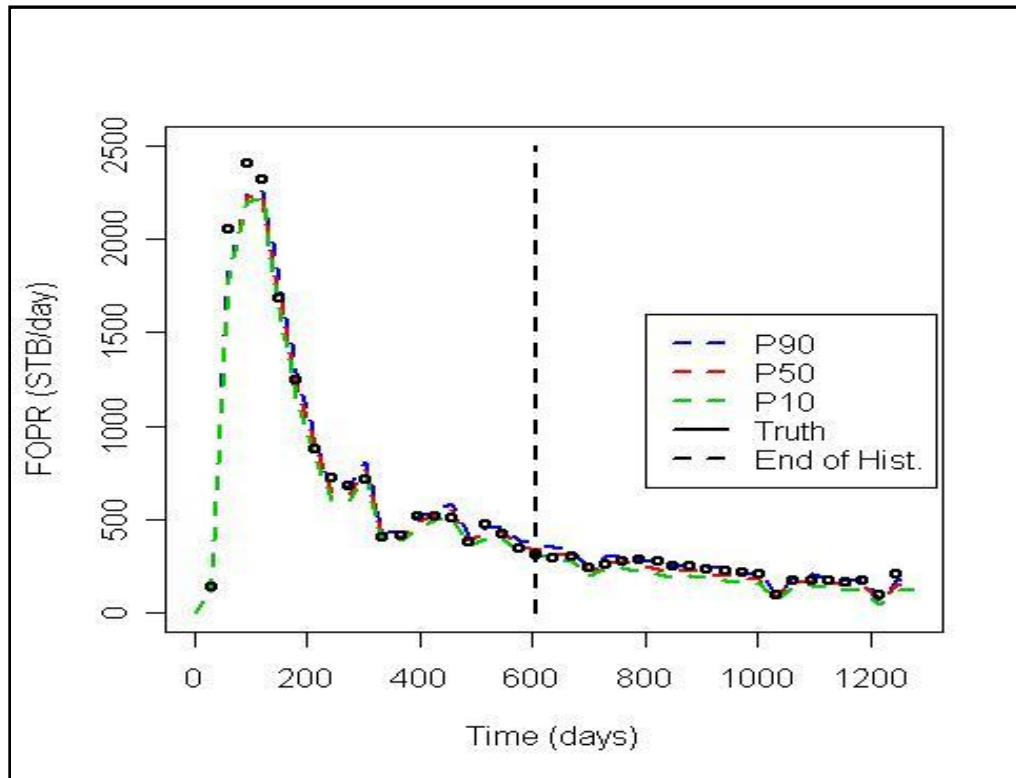


Figure 6.27: Bayesian credible intervals for Run 1

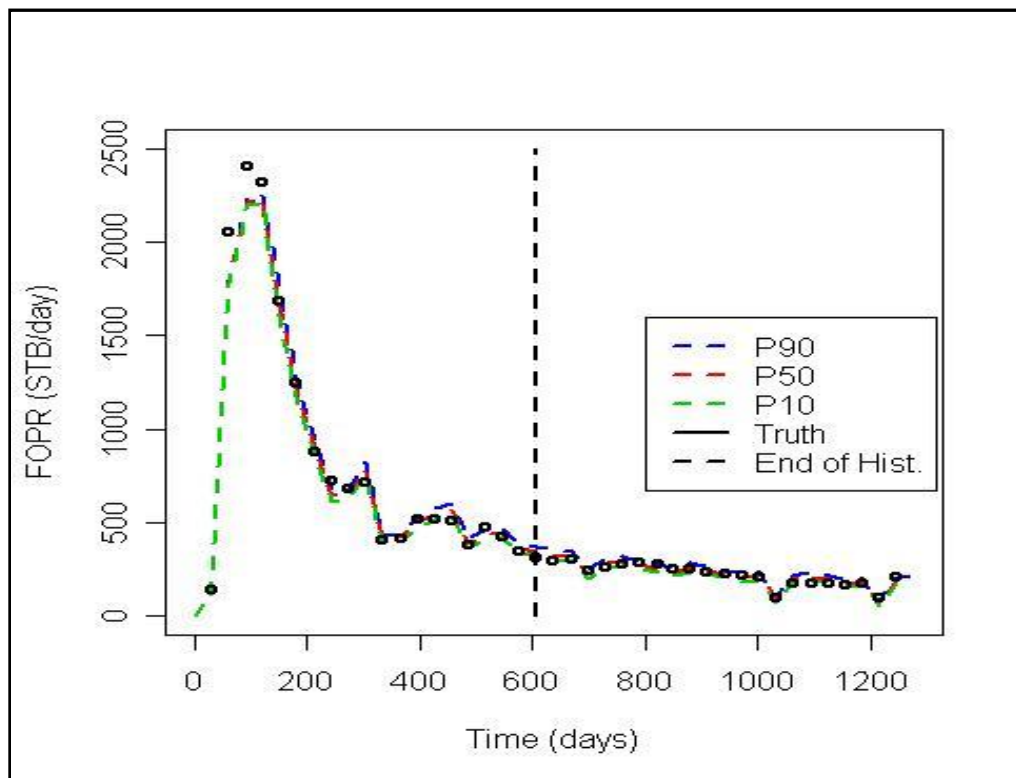


Figure 6.28: Bayesian credible intervals for Run 2

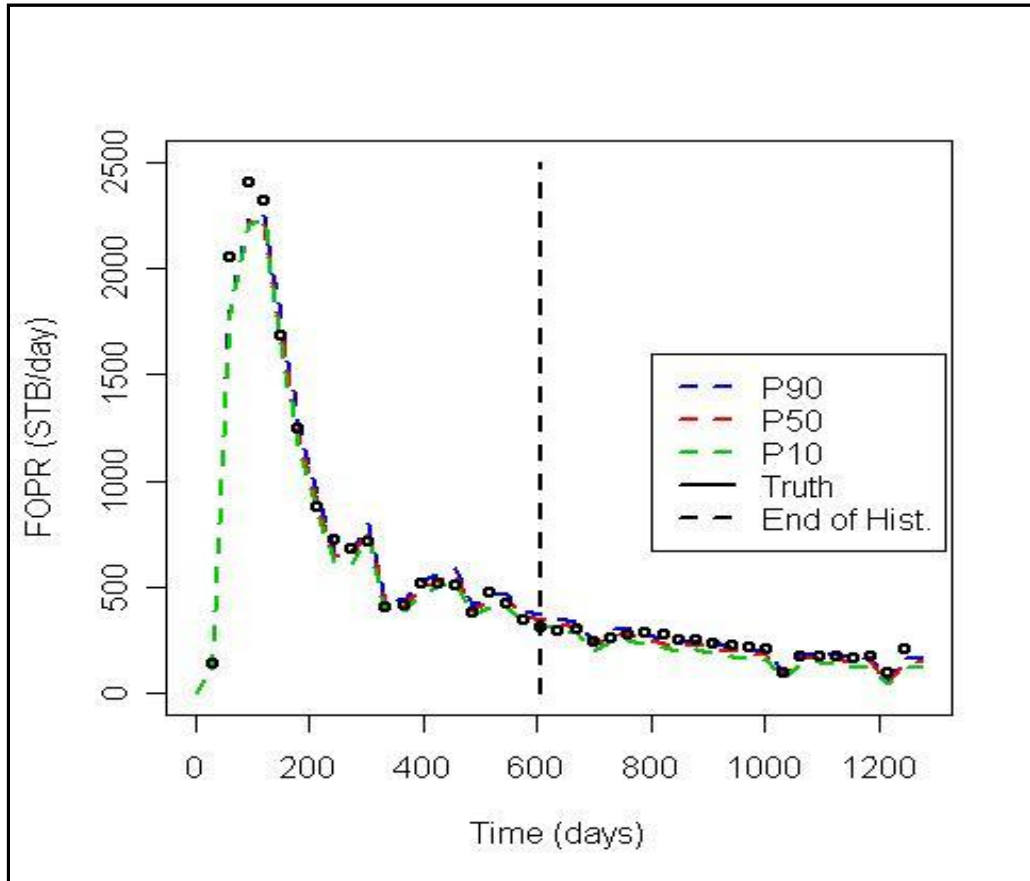


Figure 6.29: Bayesian credible intervals for Run 3

6.10 Chapter Summary

In this chapter we presented the Hamiltonian Monte Carlo (HMC) algorithm as a significant potential tool for rapid sampling of high dimensional parameter spaces which can be utilised for reservoir history matching, parameter estimation, and uncertainty quantification. HMC is a Markov chain Monte Carlo (MCMC) technique that combines the characteristics of the Hamiltonian dynamics and the Metropolis algorithm to sample complex distributions. The HMC approach integrates gradient information to address the random walk problem in the classical Metropolis algorithm by having auxiliary momentum variables that allow it to continue in the same direction for many steps followed by the Metropolis rejection test. The leapfrog simulation has been used for discretisation of the Hamiltonian dynamics equations.

An application for rapid generation of stochastic realisations, particularly of the permeability field, is reported in Bonet-Cunha et al. (1998). The applications of the HMC algorithm to parameter estimation and uncertainty quantification in petroleum history matching examples have not been reported in petroleum literature. This chapter

demonstrated our technical contribution of the work carried out and tested on Teal South field study using the HMC technique. We have shown that algorithms based on Hamiltonian dynamics have the potential to be effective tools in uncertainty quantification in the oil industry. In some complex cases as in the IC fault model case, more work needs to be carried out as we will outline in Chapter 8.

Chapter 7 – Comparison of Stochastic Sampling Algorithms for History Matching and Uncertainty Quantification

The purpose of this chapter is to investigate the efficiency of three stochastic sampling algorithms for generating history matched reservoir models: Hamiltonian Monte Carlo (HMC) algorithm, Particle Swarm Optimisation (PSO) algorithm and the Neighbourhood Algorithm (NA). As we described in previous chapters, HMC is a Markov chain Monte Carlo (MCMC) technique that uses Hamiltonian dynamics to achieve larger jumps than are possible with other MCMC techniques. PSO is a swarm intelligence algorithm that uses similar dynamics to HMC to guide the search, but incorporates acceleration and damping parameters to provide rapid convergence to possible multiple minima. The NA is a sampling technique that uses the properties of Voronoi cells in high dimensions to achieve multiple history matched models.

The comparative analysis in the next sections in this chapter is done for the two case studies in Section 4.3: the simple real Teal South and the complex IC Fault model. The algorithms are compared by generating multiple history matched reservoir models, and comparing the Bayesian credible intervals ($p10-p50-p90$) produced by each algorithm. We show that some algorithms are able to find models that fit the data well quickly, whereas others are able to find a more diverse set of models in parameter space. The effects of the different sampling of model parameter space are compared in terms of the $p10-p50-p90$ uncertainty envelopes. For the Teal South example we show that all algorithms are able to find equivalent match qualities and uncertainty estimates

(Mohamed et al., 2010b), while in the IC Fault example, the results vary (Mohamed et al., 2010c).

7.1 Field Application Test 1: The Teal South Reservoir

The same Teal South model setup with 8 uncertain parameters: horizontal permeabilities of all the five layers, a single value for k_v/k_h , rock compressibility and aquifer strength is used in the comparison with the uniform priors in the logarithms of the variables as indicated in Table 4.2. The first 181 days of production data were used in the history match and the remaining 3 years is used as prediction data to measure the predictive quality of the history matches (6 measurements out of 41 measurements). The simulator production was controlled to match the total liquid rate, and history matching was carried out by matching the field oil rate. A least squares misfit was used to measure how well a specific set of reservoir model parameters fits the observed data. The standard deviation of the oil production measurement errors was set to 100 STB/D based on previous works estimations (Christie et al., 2002; Valjak, 2008).

7.1.1 Algorithm Setup Specifications

For all the methods we started from the same initial population comprised of 30 models generated randomly in parameter space with 2D projections examples shown in Figure 4.7.

We ran 45 iterations for both NA and PSO algorithms. For NA the parameters we chose were: $n_s/n_r = 30/15$ for balancing exploration and exploitation capabilities. For PSO the parameters we chose were: $c_1 = c_2 = 2$ with a linear decrease in the inertial weight w from 0.9 to 0.4 (Eberhart and Shi, 2007). The random boundary strategy is used for handling particles flying outside of the feasible region, PSO-LDR (see Section 4.1.3.3). Although, this is not the best choice to handle the boundaries, it is useful to test the PSO performance with the NA in speed terms. The total number of reservoir model simulations was 1380 in NA and PSO.

We ran a single HMC chain of 1350 reservoir model simulations starting from a random one of the 30 initial points used for NA and PSO. We started the HMC sampling using gradients estimated from the GRNN with kernel width adjustment factor equal to 0.4 using the 30 initial points to construct the GRNN approximated misfit surface. Thus, the total number of simulations used is 1380. The leapfrog stepsizes were chosen to be $\Delta t =$

$0.02 \times \sigma_i$, where $\sigma_i = (C_{ii})^{1/2}$ and C is the covariance matrix of the data points. At each HMC iteration, the number of leapfrog steps taken was randomly drawn from a uniform distribution from 10 to 25.

7.1.2 History Matching Results

We will first compare the performance of the two sampling algorithms NA and PSO. Both NA and PSO generate multiple history matched models. Uncertainty quantification is then carried out using the separate NAB code which converts the posterior probability density at each sampled location to a posterior probability (equal to density times the volume of Voronoi cell surrounding a point).

Figure 7.1 (left) shows the best history match obtained by NA and PSO. There is little to choose between the two history matched models. Figure 7.1 (right) shows the optimal values for the 5 horizontal permeabilities – the best fitting parameters found by NA and PSO are different (although the differences are not large). Two other sets of parameter values providing almost equally good matches are shown in Figure 7.2.

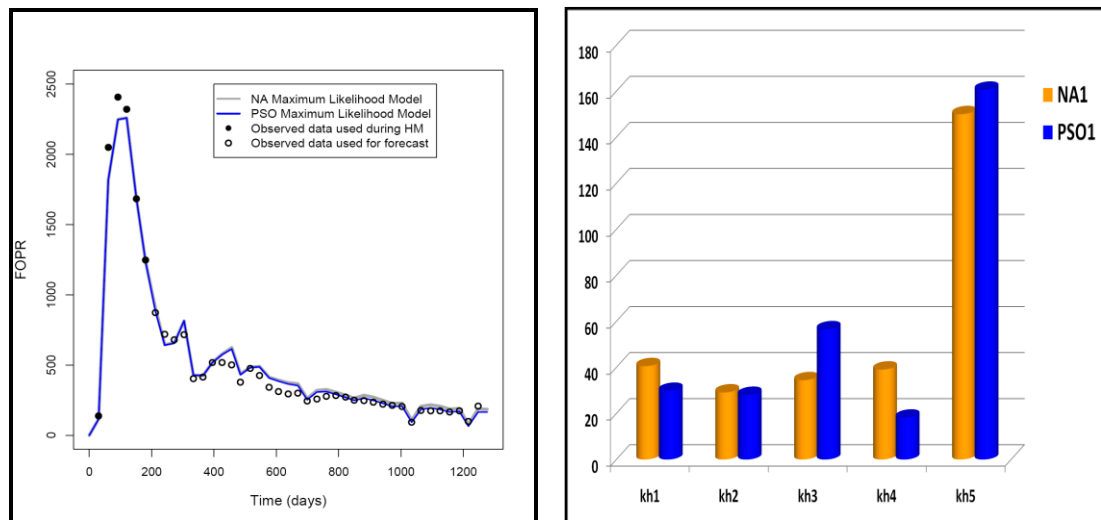


Figure 7.1: Comparison of the best history matches (left) and the corresponding permeability estimates (right) obtained from NA and PSO

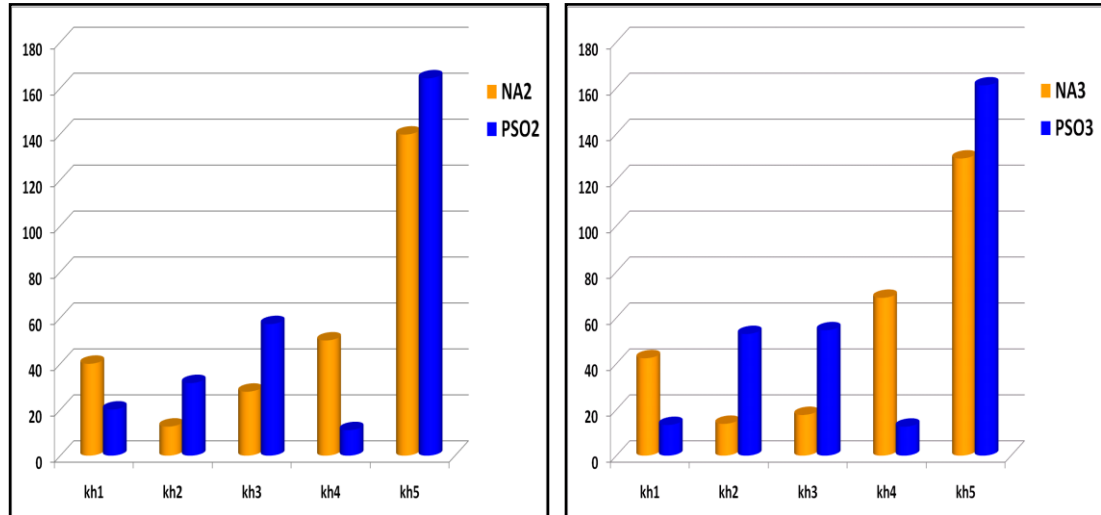


Figure 7.2: Two alternative sets of parameter values providing almost equivalent match qualities to the maximum likelihood model

Table 7.1 shows the best history matching misfits obtained by each method in both historical and forecast periods. The history misfits for all methods are around 4.2. The corresponding forecast misfit calculated with the same misfit definition in Eq. (4.6), in which the number of observations used is $T = 35$, is presented in the table with the best match in the forecast being obtained by HMC followed by PSO then NA. The values show that there are small differences in the probability values of these models.

Table 7.1: Best history and forecast misfits for all stochastic algorithms

Algorithm	Number of Simulations	Best Misfit	Forecast Misfit
PSO	1380	4.2	7.45
NA	1380	4.2	8.33
HMC	1380	4.2	6.91

Figure 7.3 shows the progress of the mean generational minimum misfit of NA and PSO. To generate this plot we ran 5 runs of NA and PSO. NA and PSO used identical sets of points for each run, with a new set of random starting conditions generated for each run. We then plotted the mean generational minimum misfit, along with the standard deviation around each point. We can see that NA and PSO reach the same misfit, but that on average PSO reduces the misfit in each generation more rapidly than NA.

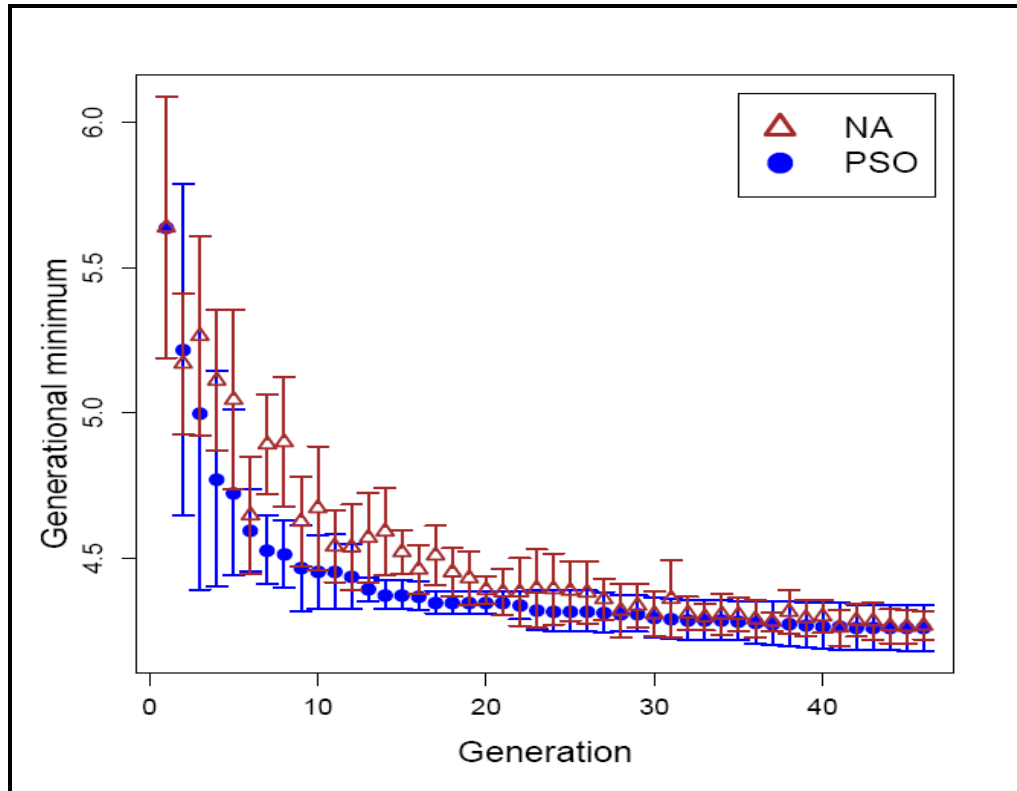


Figure 7.3: Evolution of the mean generational minimum misfit for NA and PSO

The sampling history of NA is shown in Figure 7.4, and that of PSO in Figure 7.5. Each plot consists of 8 plots, showing the evolution of the parameter sampling as a function of sampling. The points are colour coded according to the misfit, showing the concentration of sampling at low misfit values as the algorithm sampling evolves. Note that the red points have misfit 10 or above, and include many models that do not match well at all.

The parameter values for the good history matched models can be seen by looking at the range of the blue points. Both algorithms appear to concentrate sampling for the permeability values in similar zones, although NA appears to be holding onto two possible minima for $\log(kh_2)$ (upper right plot), whereas PSO has homed in on the higher value. The sampling for rock compressibility, aquifer strength, and $\log(k_v/k_h)$ evolves differently for both algorithms. Nonetheless, the best matches obtained are very comparable in quality as shown in Figure 7.1 (left).

Figure 7.6 shows the sampling history for HMC. Note that in HMC, the algorithm is not continually trying to improve the degree of match; rather it is constantly sampling models that are acceptable history matches. Most of the models generated have misfits

of 8 or below – corresponding to an average deviation from observed values of 1.5 standard deviations or below.

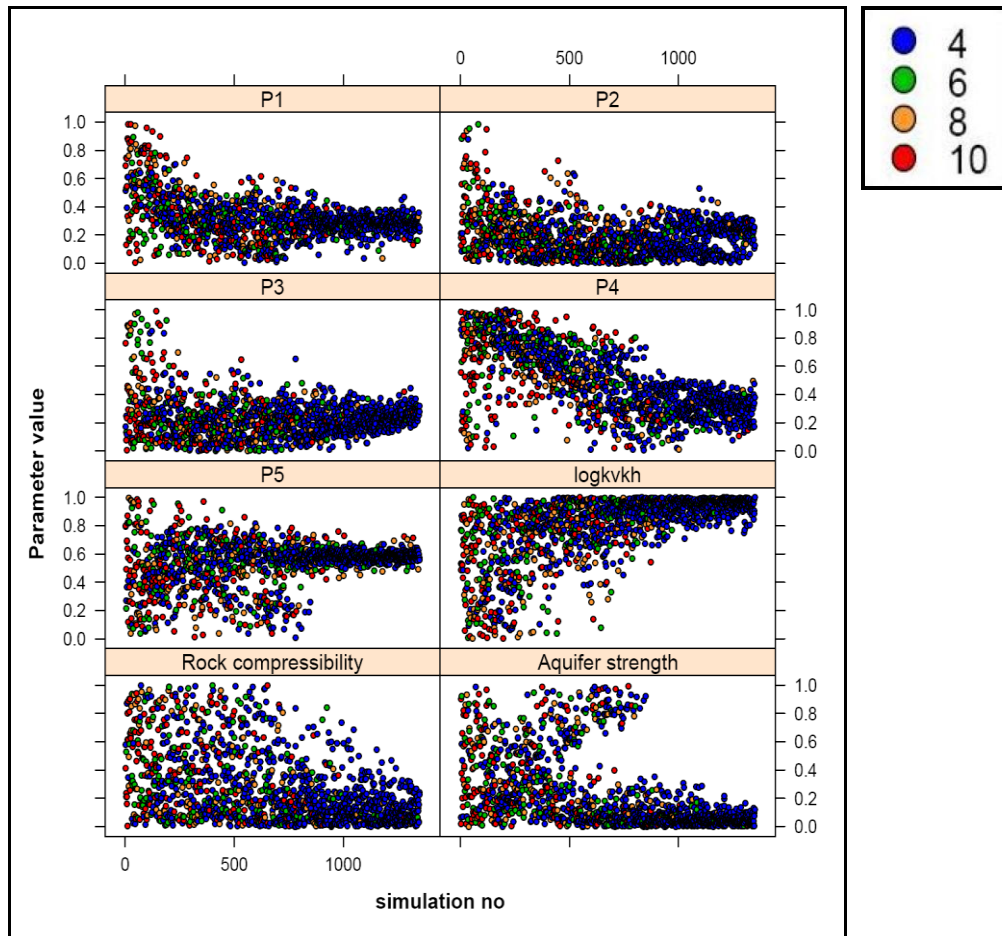


Figure 7.4: Sampling history of NA for each of the 8 unknown parameters

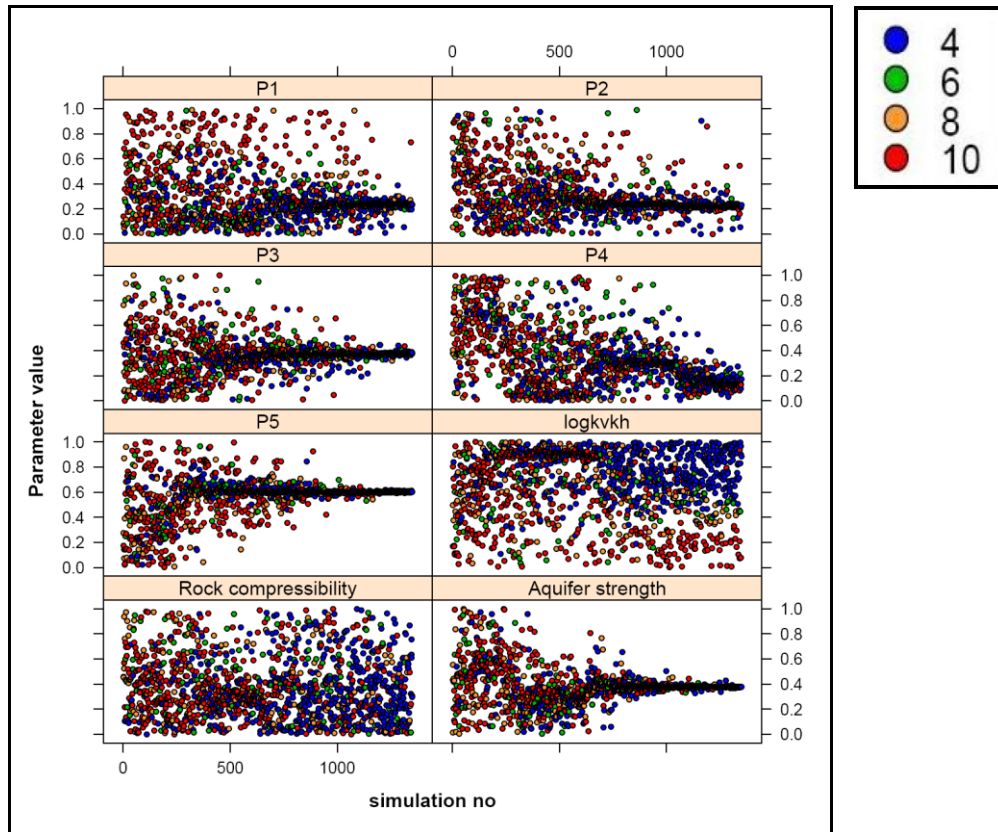


Figure 7.5: Sampling history of PSO for each of the 8 unknown parameters

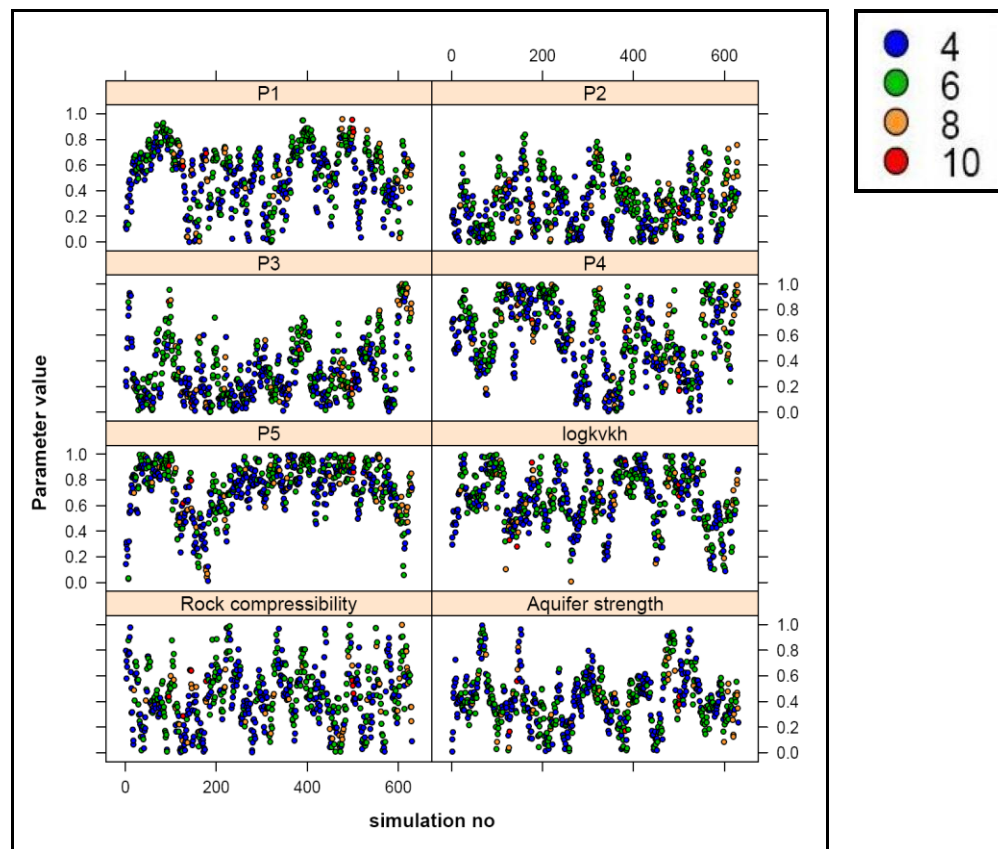


Figure 7.6: Sampling history of HMC for each of the 8 unknown parameters

7.1.3 *Uncertainty Quantification: Comparison of NA, PSO, HMC*

Hamiltonian Monte Carlo is designed so that the models generated sample correctly from the posterior distribution (within sampling error), which means that $p10$, $p50$, $p90$ predictions can be generated from an appropriate sum of the generated models. Neither PSO nor NA has this property, so a separate assessment has to be made to determine the posterior probability of each of the sampled models. We used the NAB code (Sambridge, 1999b) to determine these probabilities for PSO and NA. NAB works by running a Gibbs sampler on the approximate misfit surface generated by assuming the misfit is constant over each of the Voronoi cells surrounding a sampled point.

Diagnostic tools for HMC introduced in Section 6.3.6 are used to analyse HMC samples before computing predictions. Some statistics are shown in Table 7.2 where the effective size for each parameter corresponds to the effective sample size for the parameter. The effective size is calculated on the basis of the autocorrelation: the less the autocorrelation the higher the effective size. The predicted value of the spectral density at frequency zero is denoted by $P_0 = P(k=0)$. Finally, the order is the order of the fitted model. The plots in Figures 7.7, 7.8, 7.9, and 7.10 show the autocorrelation, the autocorrelation length, the power spectrum, and the marginal density per parameter respectively. The effective size tells how many independent samples to get out of the 600 accepted samples. As shown in the table the minimum across all the parameters is 24 meaning that it is allowed to use 24 for computing statistics of predictions. If all the samples for predictions are used, slightly biased results may be obtained which was not the case in this example. Thus, all samples have been used. In practice if biased results are obtained, then sub-sampling 1 sample every $600/24$ is required.

Table 7.2: Statistical measures

Parameter	Effective size	$P_0=P(k=0)$	Order
P1	35.43	0.89	1
P2	49.00	0.48	2
P3	35.85	0.92	4
P4	24.47	1.91	2
P5	33.63	0.75	2
$\log k_v k_h$	32.53	0.85	3
Rock compressibility	56.91	0.59	4
Aquifer strength	32.06	0.79	1

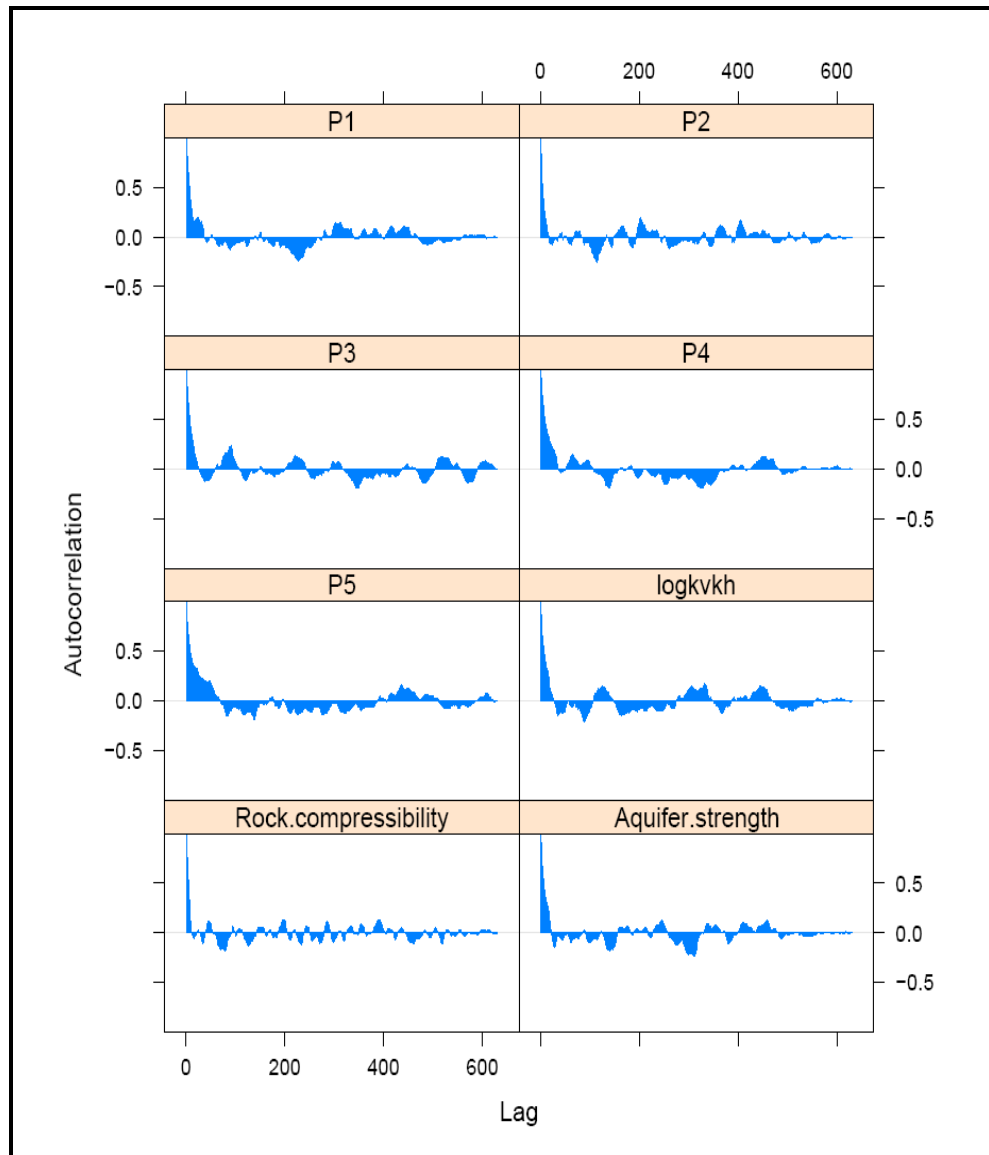


Figure 7.7: Autocorrelation of HMC for each of the 8 unknown parameters

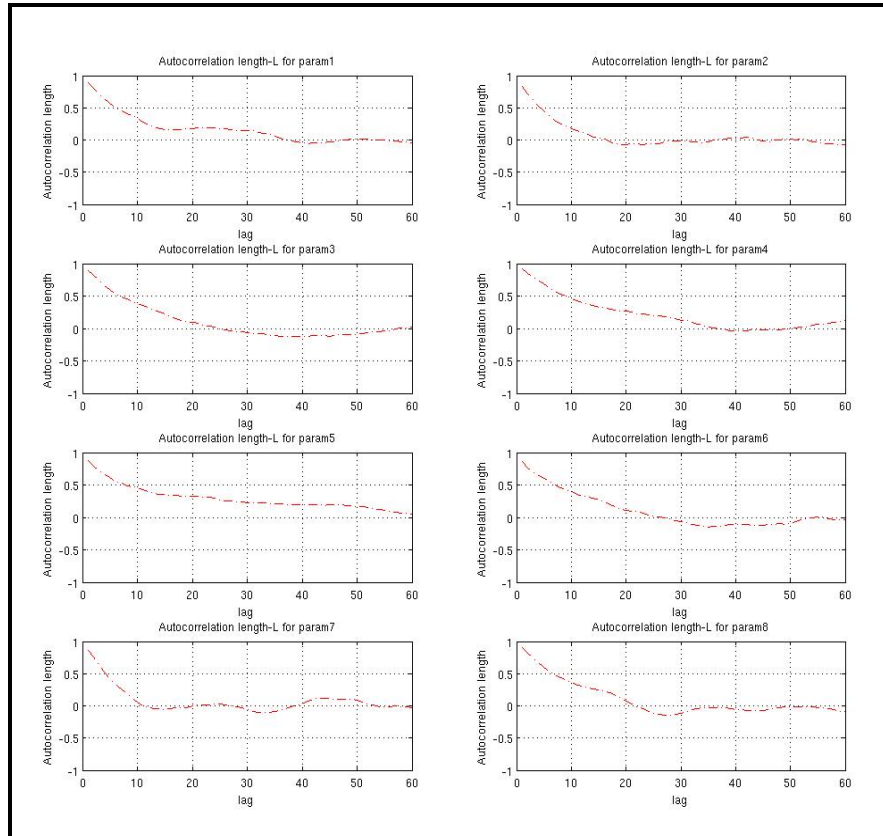


Figure 7.8: Autocorrelation length of HMC for each of the 8 unknown parameters

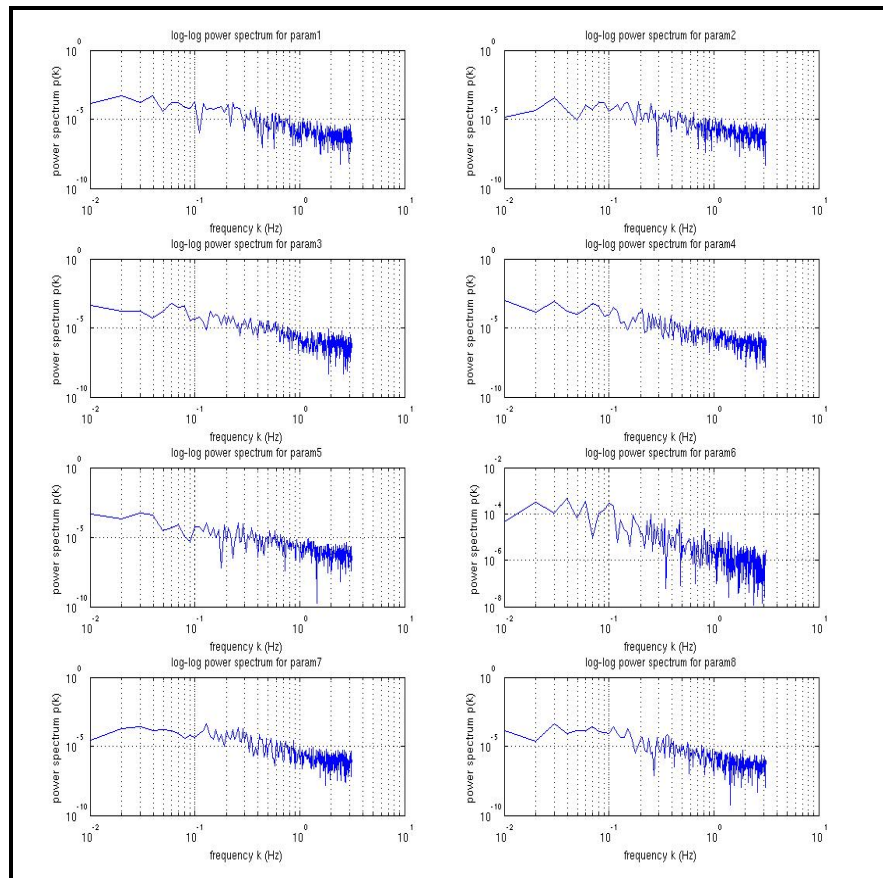


Figure 7.9: Power spectrum, $P(k)$, of the 8 unknown parameters obtained from the HMC

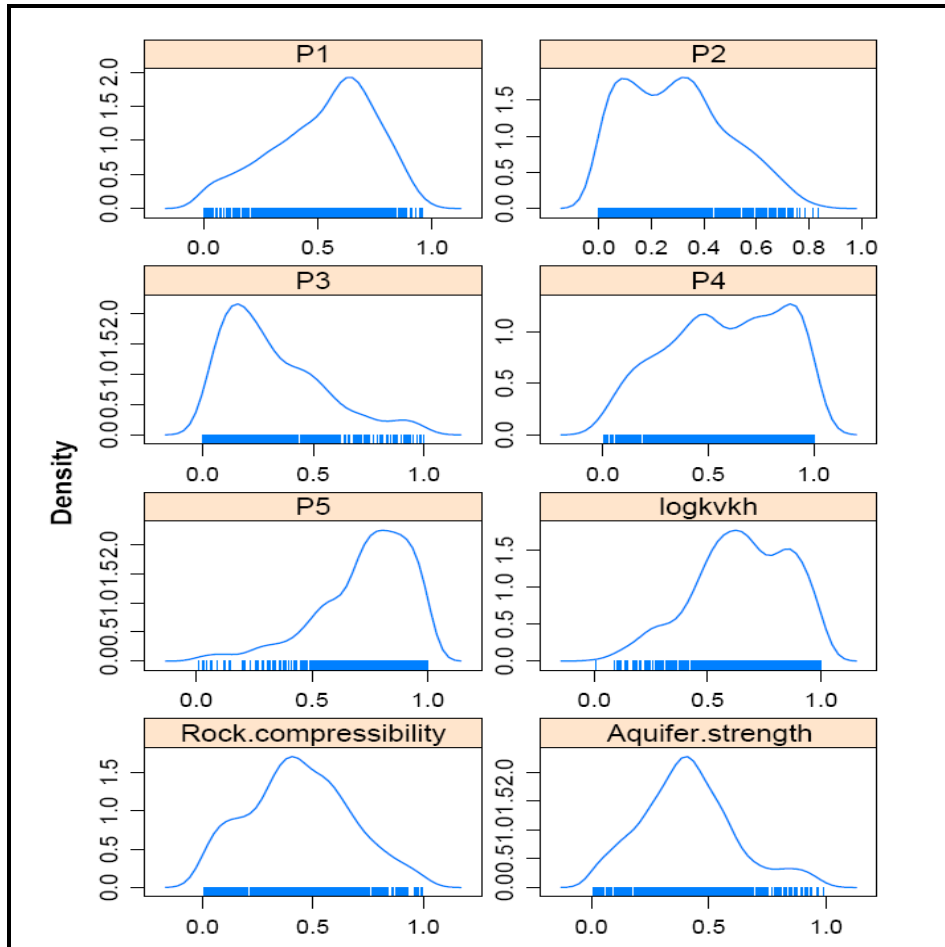


Figure 7.10: Marginalised distributions of the 8 unknown parameters Teal South model obtained from the HMC

Figure 7.11 and Figure 7.12 show the Bayesian credible intervals (p_{10} - p_{50} - p_{90}) for oil rate after history matching to the first 181 days of production. Both NA and PSO have produced similar ranges. The equivalent plot for HMC is shown in Figure 7.13.

Figure 7.14 shows the relative uncertainty, $\left[\frac{(p_{90} - p_{10})}{p_{50}}\right]$, for each timestep. PSO have wider uncertainty ranges after around 400 days while NA and HMC have narrower uncertainty estimates than the PSO one. The overall relative uncertainty estimates for the three methods have little differences between them in this example.

Figure 7.15 presents the Bayesian credible intervals for total recovery where all methods captured the observed value.

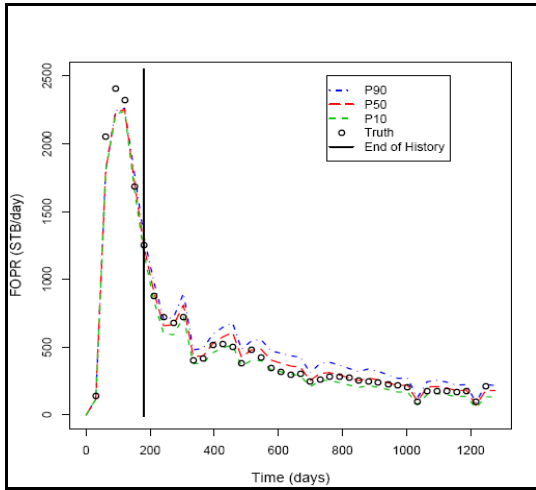


Figure 7.11: Bayesian credible intervals generated by NA

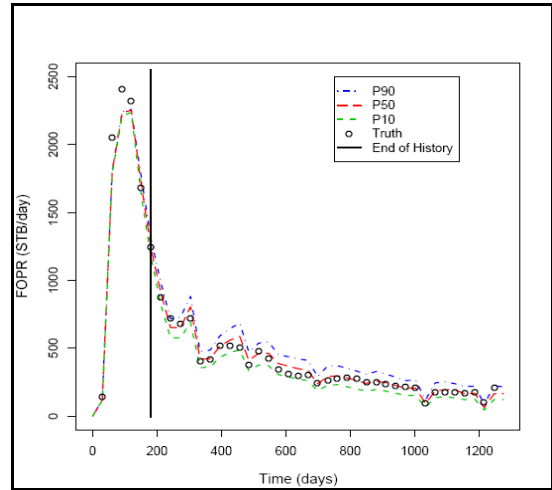


Figure 7.12: Bayesian credible intervals generated by PSO

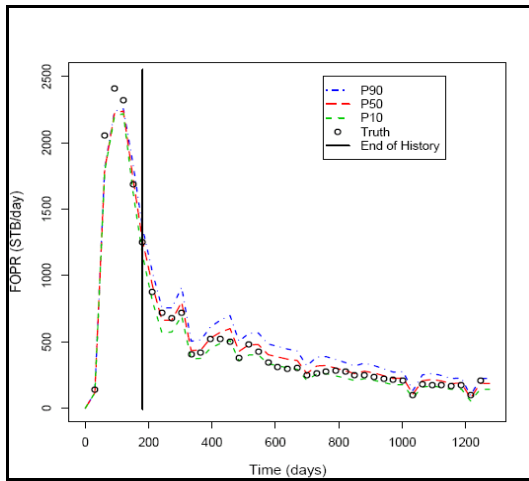


Figure 7.13: Bayesian credible intervals generated by HMC

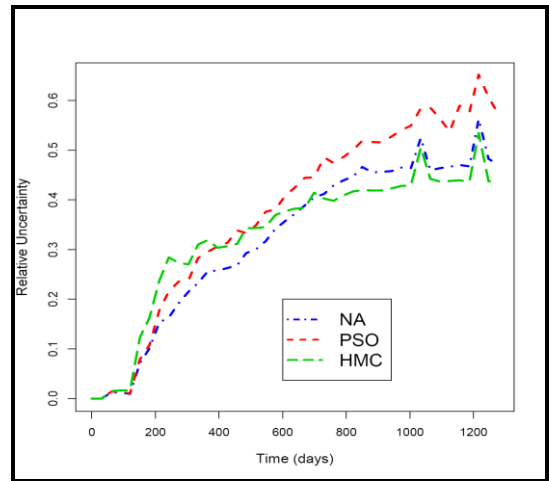


Figure 7.14: Relative uncertainty of the three methods

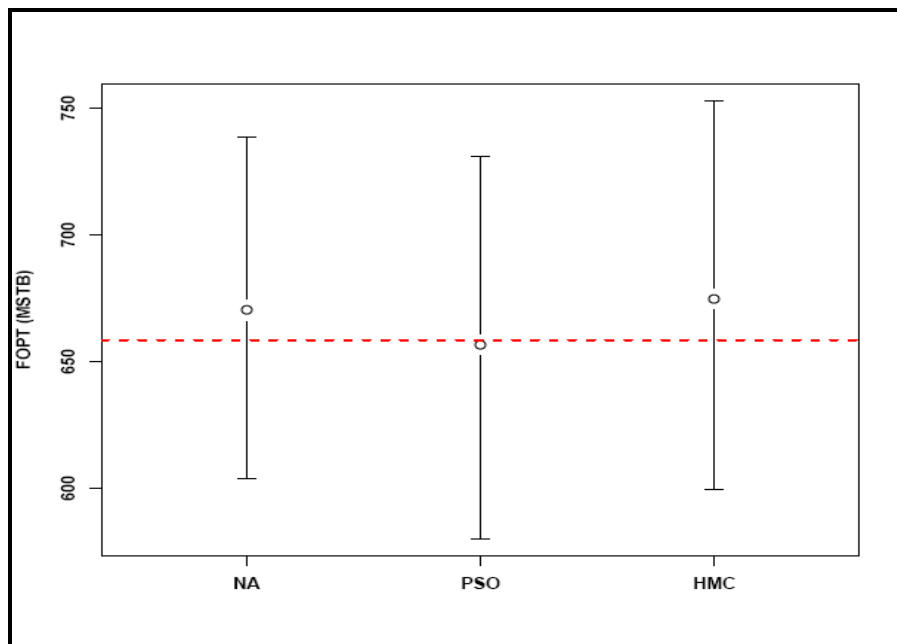


Figure 7.15: Bayesian credible intervals for total recovery

Figure 7.16 shows the cumulative density function of the oil rate at two different times. The left hand plot is the CDF for the oil rate after 181 days at the end of the history matching period. The uncertainty plots for all three algorithms are very similar. The right hand plot shows the CDF after 1187 days. In this case there is a greater difference between the algorithms, although they are still similar. HMC provides a slighter wider $p10-p90$ range after the end of the history matching period than NA and PSO. The solid black vertical lines show the observed value at the end of history match time (left) and at the forecasted time (right).

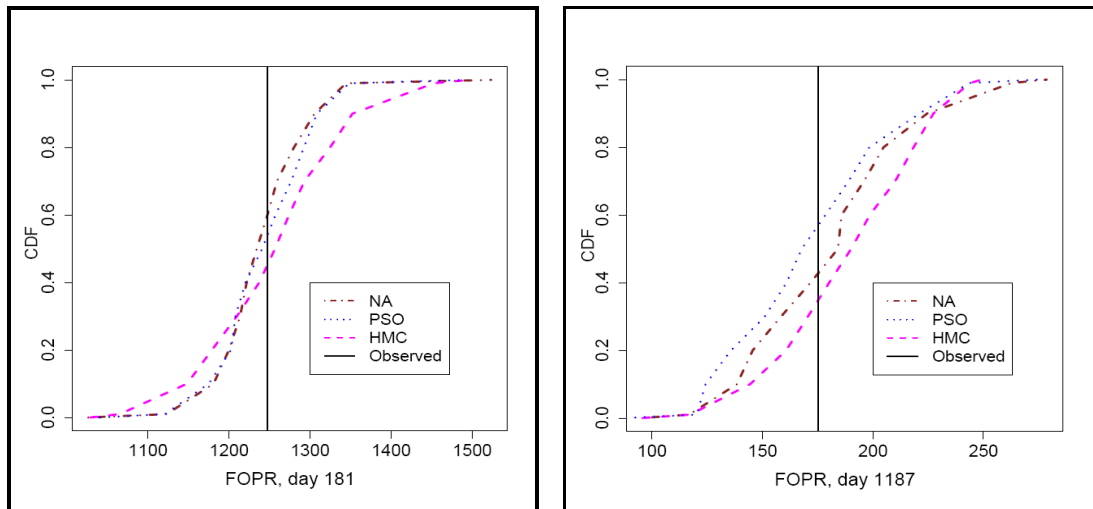


Figure 7.16: Cumulative Distributions from NA, PSO and HMC at (left) and after (right) the end of history matching

7.2 Field Application Test 2: The IC Fault Model

In the previous section we compared the performance of Hamiltonian Monte Carlo with two stochastic optimisers (Particle Swarm Optimisation and the Neighbourhood Algorithm) on Teal South, a simple example. In this section we compare it on IC Fault Model with the complex misfit surface.

7.2.1 Algorithm Setup Specifications

The stochastic sampling algorithms were set up to be as similar as possible. We used the set of 20 initial starting points obtained with Latin Hypercube Sampling (LHS) for PSO (refer to Section 4.3.2.3). We ran 65 iterations for PSO leading to a total of 1300 reservoir model simulations. The two versions of PSO used differed only in their handling of boundary effects: the first variant uses an absorbing boundary strategy (PSO1) where the normal component of a particle's velocity is zeroed at the boundary; the second variant used a reflecting strategy (PSO2), where a particle hitting the

boundary is reflected from the boundary. The PSO parameters we chose were: $c_1 = c_2 = 2$ with a linear decrease in the inertial weight w from 0.8 to 0.4. We also ran two cases with the Neighbourhood Algorithm (NA) each with 1300 simulations in total. The first case, NA1, had $n_s = n_r = 50$ for 25 generations starting from 50 points. The second case, NA2, had $n_s = n_r = 100$ for 11 generations starting from a larger population of 200 models.

HMC sampling requires gradients, which were estimated from a general regression neural network (Specht, 1991). We started twenty independent chains each with 1000 simulations.

7.2.2 History Matching Results

Table 7.3 shows the misfit values obtained with the two optimisers and HMC and their corresponding parameter values. The best misfit value was obtained with PSO1 followed by HMC, then NA misfits, and finally PSO2. The corresponding history matches for all methods is shown in Figure 7.17.

We compared the sampling performance with the previous identified structure, the twisting, ribbon-like structure in Figure 4.30 for the samples obtained with Uniform Monte Carlo sampling to act as a benchmark result. Figure 7.18 shows the samples with misfit ≤ 25 generated the stochastic sampling algorithms. Both NA and PSO are able to sample from parts of the region of good fitting models, but neither captures the whole structure.

Table 7.3: Best misfits values obtained with the three stochastic methods and their corresponding parameter values

Algorithm	Number of Simulations	k_{high}	k_{low}	$throw$	Best Misfit	Corresponding Forecast Misfit
PSO1 (LDS)	1300	126.94	1.59	4.45	0.11	1.90
PSO2 (LDB)	1300	137.67	2.50	42.45	1.27	20.94
NA1 (50/50)	1300	135.19	2.80	36.54	0.45	22.71
NA2 (100/100)	1300	126.24	3.06	4.18	0.40	6.93
HMC	1000×20	126.59	1.58	3.66	0.12	2.07

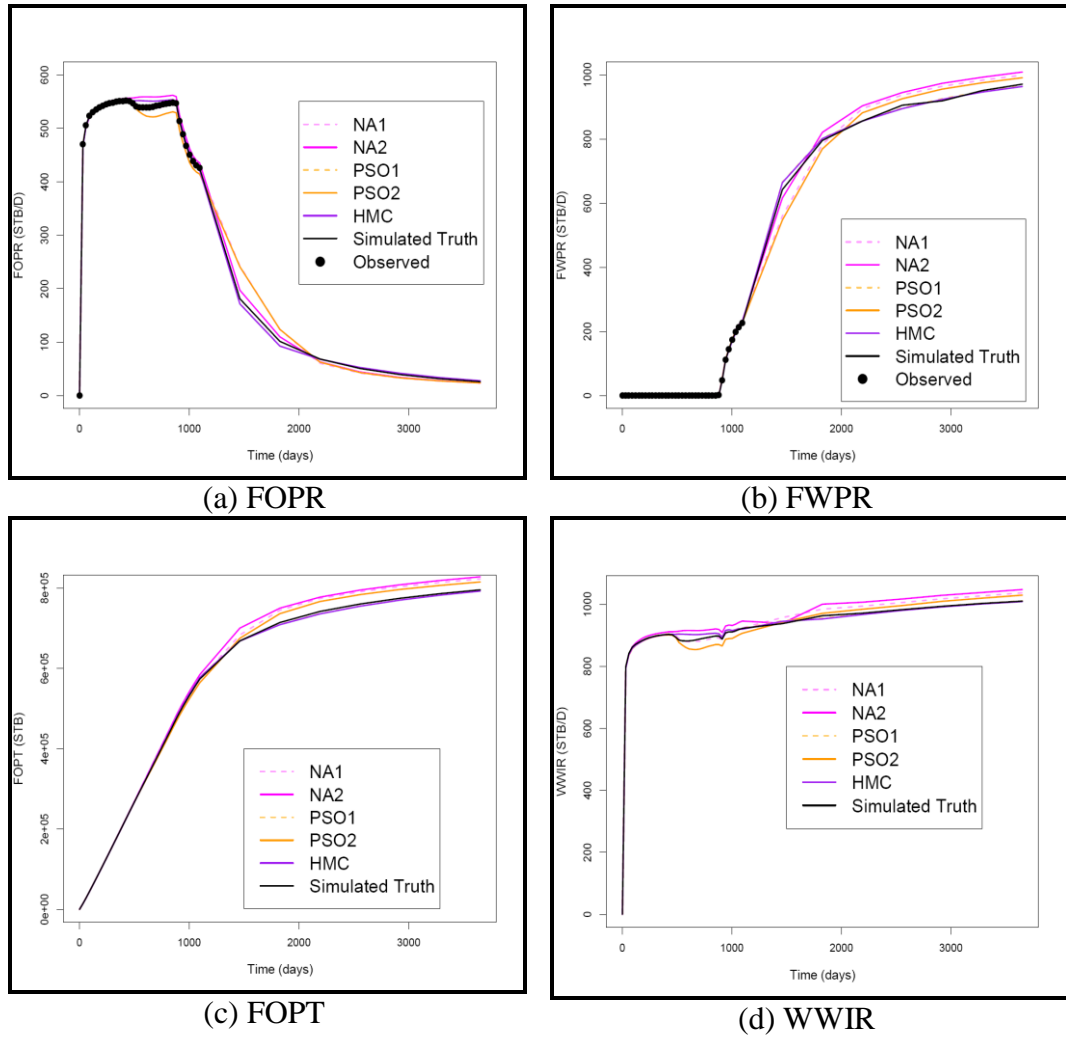


Figure 7.17: Best history matches for all methods

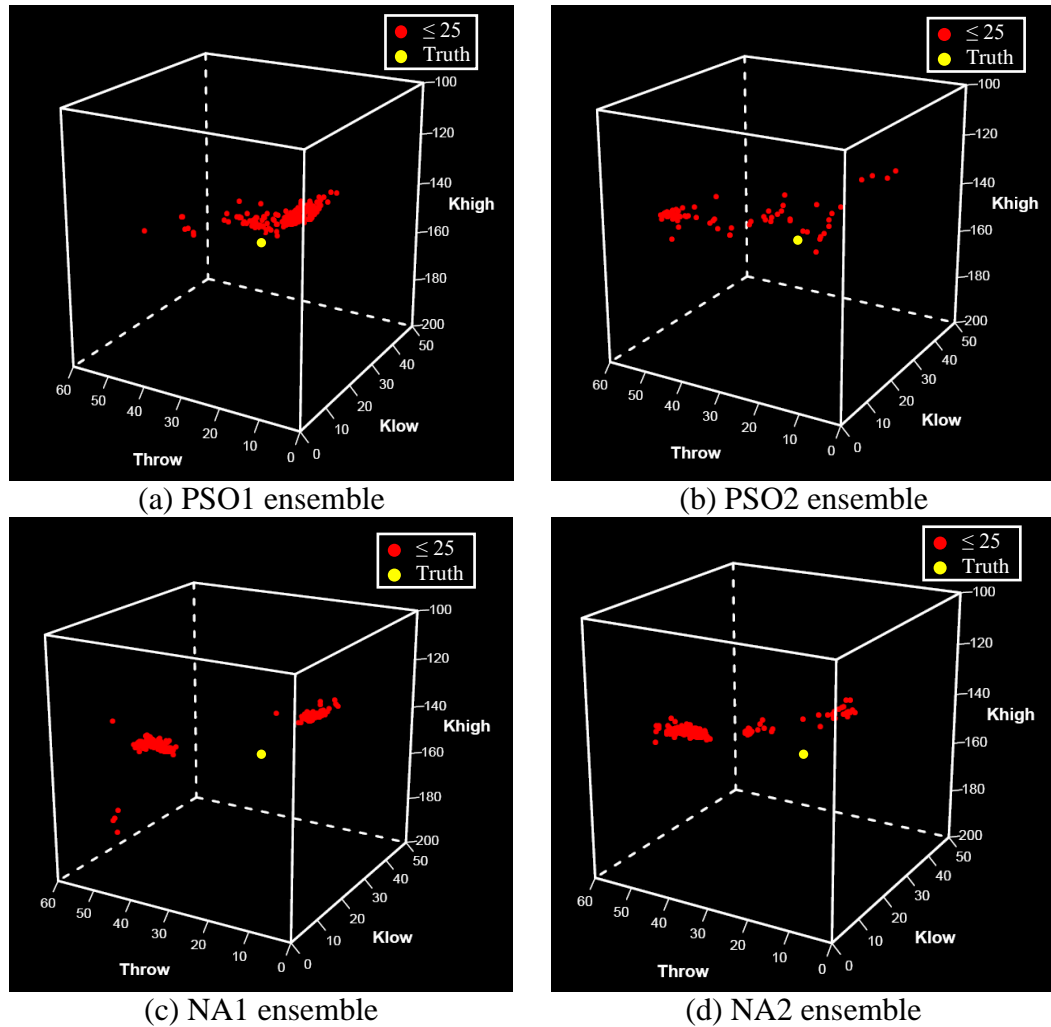
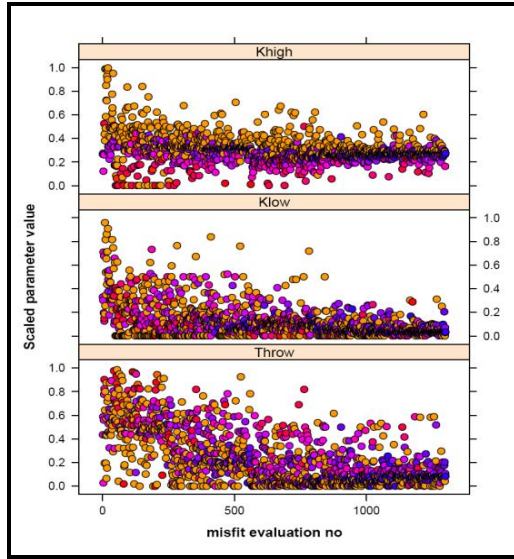


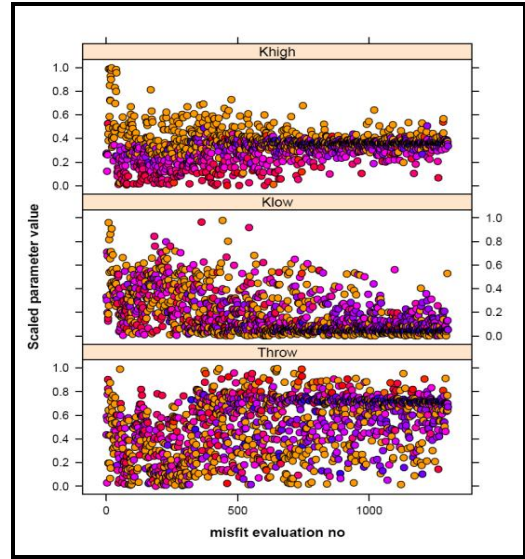
Figure 7.18: Particle swarm and neighbourhood optimisers' ensembles

Figure 7.19 shows the sampling history for the two PSO and NA variants. The points are colour coded according to misfit. While PSO two runs exploring parameter space, both NA runs concentrate on good regions particularly with the most explorative mode for this example

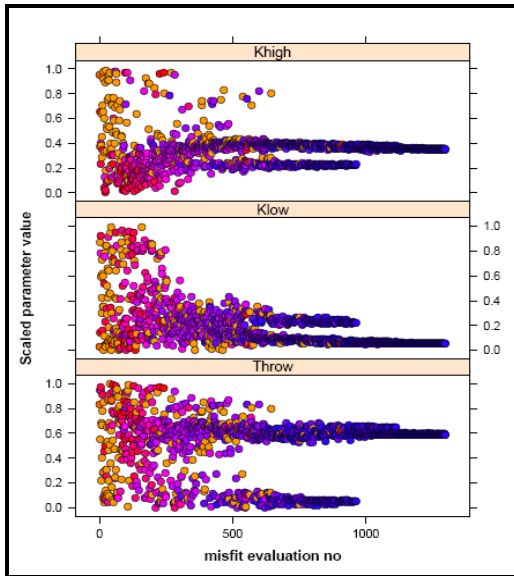
Figure 7.20 also shows two plots describing the performance of HMC. The twenty independent chains of HMC are shown in Figure 7.20(b) where each chain is represented with a different colour. We can see that each of the chains has explored a localised area near the random starting point. Since HMC satisfies the Markov chain requirements, we know that it will eventually sample from the posterior, but for this complex surface it has not achieved that in a limited number of samples. From these results, we can see that the initialisation and the choice of tuning parameters used in the sampling algorithms are crucial for achieving an effective exploration of the misfit surface.



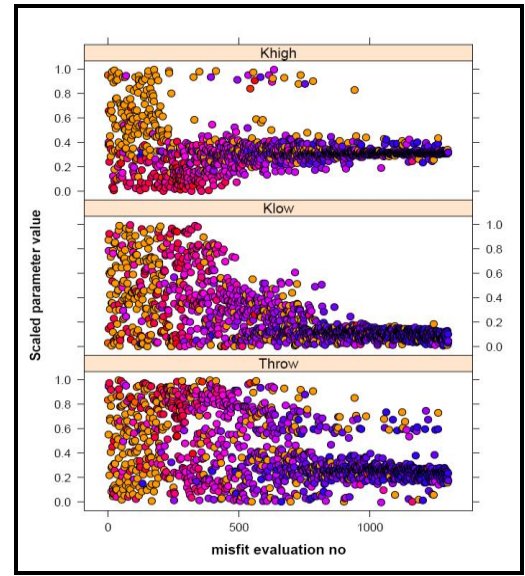
(a) Sampling history of PSO1



(b) Sampling history of PSO2

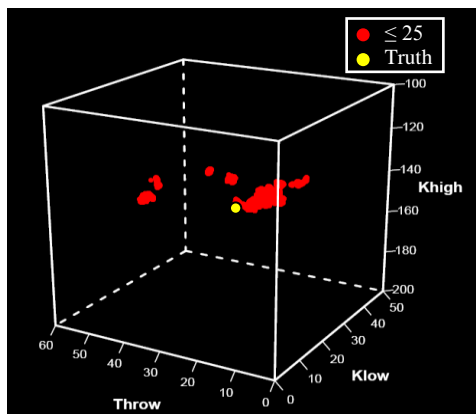


(c) Sampling history of NA1

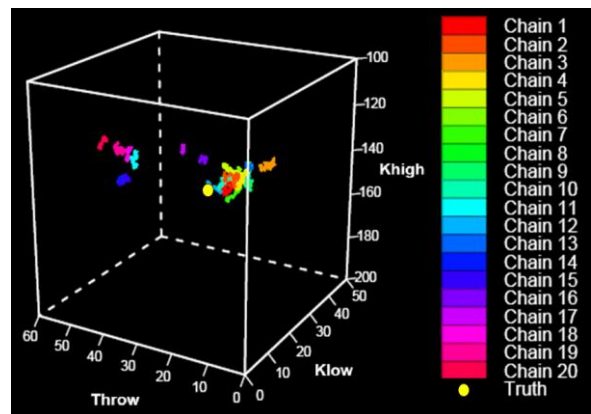


(d) Sampling history of NA2

Figure 7.19: Sampling history for two variants of PSO and NA



(a) HMC – 20 chains ensemble



(b) HMC – 20 chains ensemble

Figure 7.20: Stochastic sampling algorithms ensembles

We indicated early on that MCMC algorithms propose small changes in the state vector in each iteration resulting in rare moves between modes that in turn would lead to slow convergence and high correlations between successive states. In multimodal distributions, large proposal states with high acceptance probability are sought. The medium-length runs and short runs strategies discussed in Section 6.8 need to be employed with caution by assuring that modes are visited according to their probability. The reason behind this is that the frequencies of which the modes are represented are given by their “basin of attractions” not by the total probability within each mode. Thus, using local optimisation by running several chains with initial states given by optimisations from different starting points to locate the modes may not provide accurate results. This was seen in the 20 medium-length chains obtained with HMC for IC Fault model application.

Several remedies have been investigated to produce Markov chains with faster convergence speed. Tjelmeland and Hegstad (2001) suggested using mode jumping proposal in which they specified how optimisation for local optima of the target distribution can be incorporated in the specification of the Markov chain leading to a chain with frequent jumps between modes. A generalised scheme is later investigated in Tjelmeland and Eidsvik (2004). Neal (1996b), Marinari and Parisi (1992) and Geyer and Thompson (1995) suggested two approaches to cope with multimodal distributions where a series of distributions between a “cold” distribution, equivalent to the desired distribution, and a “warm” one with no isolate modes, are defined and the chain moves between modes through the warmer distributions. Neal (1996b) proposed a tempered transitions approach whereby the chain moves between the different distributions systematically while Marinari and Parisi (1992) and Geyer and Thompson (1995) proposed simulated tempering in which the chain moves stochastically. Geyer (1991) employed Metropolis-coupled Markov chains variant where a single parameter of one state is used for all the distributions and a swap of values is proposed between neighbouring states.

To address the desirable characteristics of the HMC chain we have proposed a generalised variant of the later method of Geyer for tackling multimodal complex distributions as the IC Fault model example. This is discussed in Chapter 8 for designing an MCMC. The method could incorporate the HMC mechanism to tackle multimodal distributions more effectively.

7.2.3 Uncertainty Quantification: Comparison of NA and PSO

Figure 7.21 shows the Bayesian credible intervals ($p10$ - $p50$ - $p90$) for oil and water rates and total oil production after history matching to the first three years of production. The widest range is obtained with PSO1. PSO2 obtained uncertainty ranges very close to the database ones in Figure 4.47. The two NA runs have different results, NA1 obtained narrower ranges than NA2 which had larger population size per generation and larger initial population size. It is noted that we used larger initial NA samples than PSO for reasonable results.

Figure 7.22 presents the Bayesian credible intervals for total recovery where PSO1 produced the larger ranges and captured the observed value while the PSO2 obtained similar results to the database one.

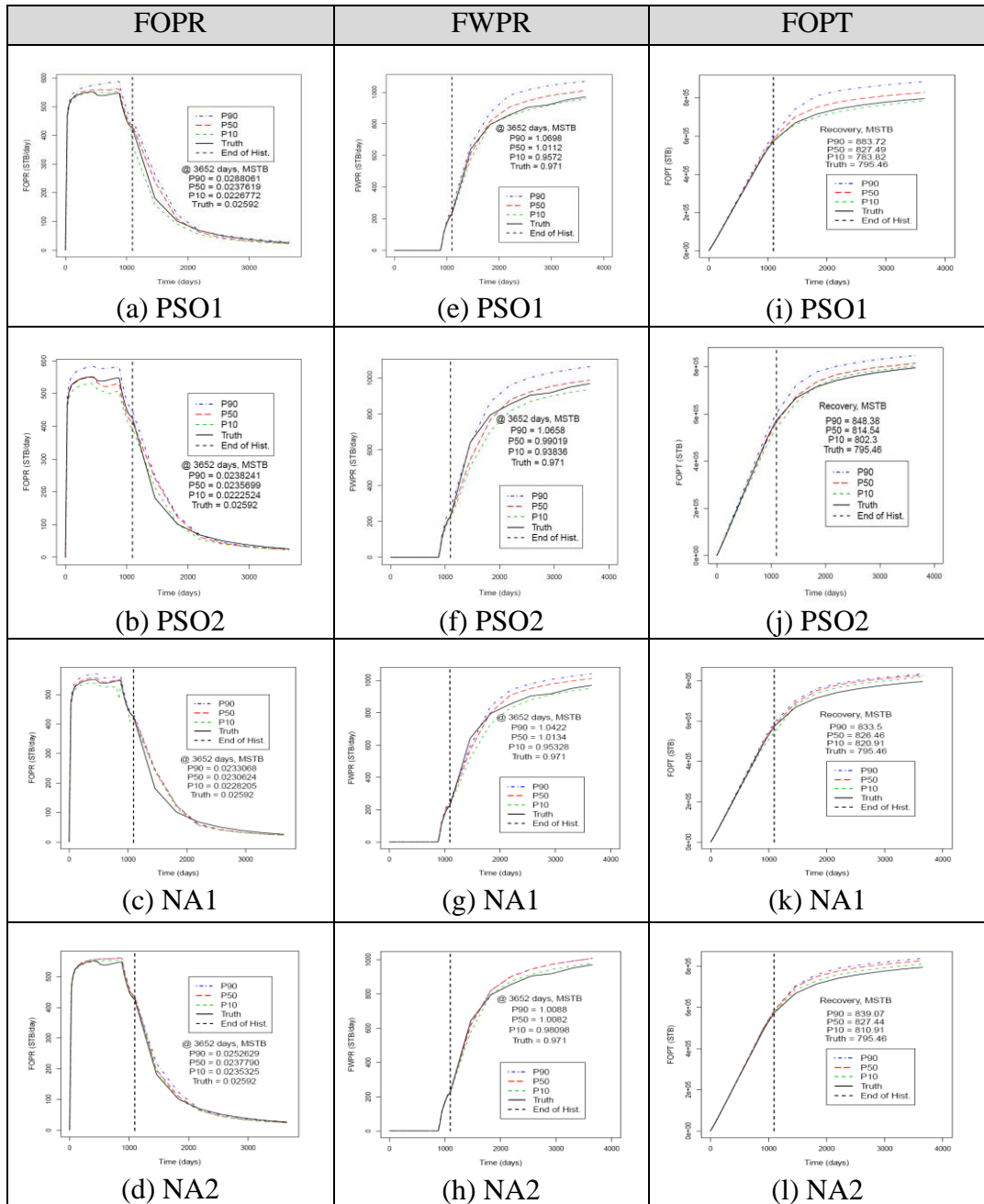


Figure 7.21: Bayesian credible intervals of oil and water rates and total recovery prediction for PSO and NA variants

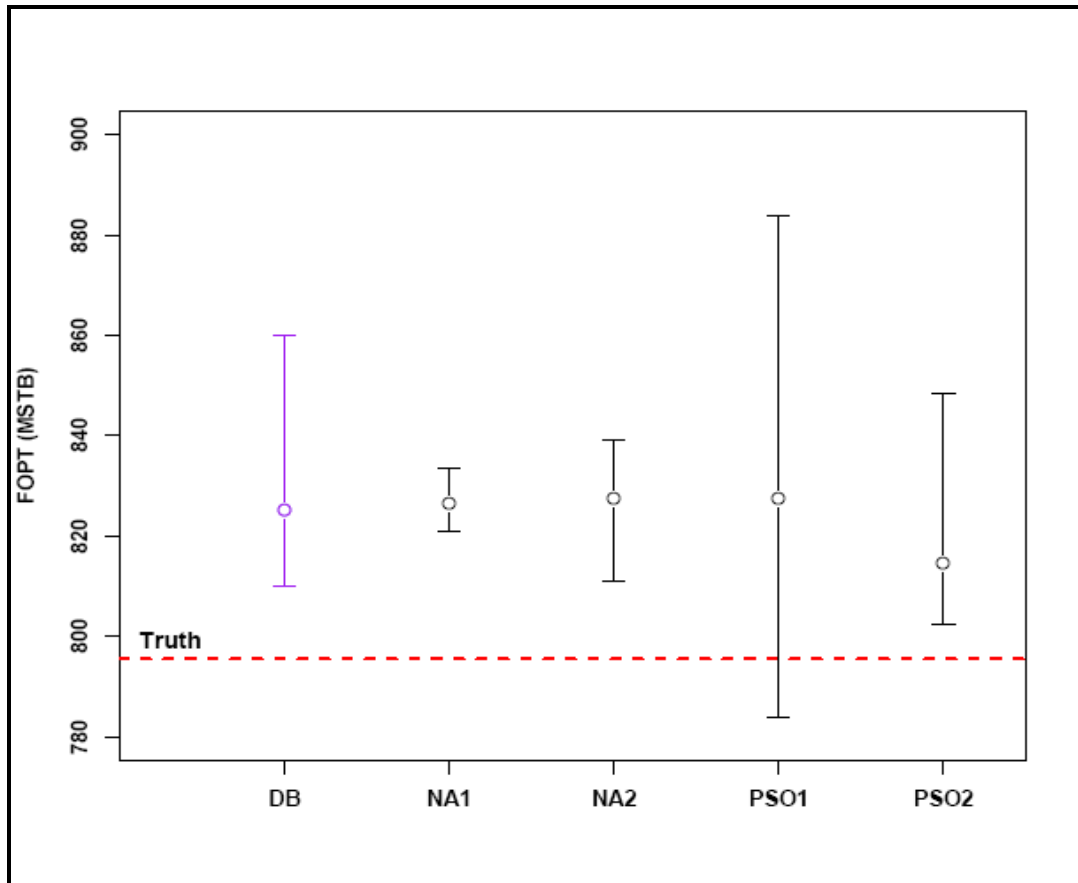


Figure 7.22: Bayesian credible intervals for total recovery prediction PSO and NA variants

7.3 Chapter Summary

This chapter investigated the efficiency of three stochastic sampling algorithms: Hamiltonian Monte Carlo (HMC) algorithm, Particle Swarm Optimisation (PSO) algorithm and the Neighbourhood Algorithm (NA). We compared the algorithms by generating multiple history matched reservoir models for the Teal South reservoir with the 8 unknown parameters and the complex IC Fault model with 3 unknown parameters.

Specific conclusions have been drawn from the Teal South study:

- NA, PSO and HMC are able to find equivalent match qualities for this example
- PSO is able to obtain a good history match in a fewer number of iterations than NA, and this behaviour is robust to changing the initial random starting conditions.
- PSO tends to concentrate sampling more in the low misfit regions than NA. For each algorithm, this behaviour depends on the algorithm parameter setting.
- NA and PSO need a separate calculation to go from sampled models to forecasts of uncertainty.

- HMC is able to generate samples from the posterior in one step.
- NA, PSO and HMC are all able to produce equivalent forecasts of uncertainty.

Specific conclusions have been drawn from the IC Fault study:

- PSO obtained good history matches, and a similar conclusion has been drawn from Figures 4.39, 4.40, 4.41, and 4.42.
- NA and PSO produced different forecasts of uncertainty depending on the tuning of algorithms and seed.
- Both NA and PSO have good convergence speed. The focus on this study which has multiple local minima is the ability to explore the entire parameter space and analyse the spread of good fitting models along with the speed since that has a direct impact on the inferences.
- In this example diversity of models is a very important factor in obtaining reliable forecasts. We have seen that PSO is able to obtain diverse models, in which the variations in sampling can be determined with large population size and choice of more diversity modes of PSO.
- HMC samples are localised in regions which are close to the initial state. A proposed solution is discussed in Chapter 8.

Chapter 8 – Advanced MCMC

Techniques for History Matching and Uncertainty Quantification – Part II: Population MCMC Methods

Assisted history matching and uncertainty quantification techniques approaches presented in the literature encounter the problems that lie with the extremely complex form of the misfit surface. This provides a very challenging task for optimisation methods to find solutions of low misfit value, and it is very difficult to exhaustively explore the posterior distribution for MCMC methods. As an example, consider the work by Liu and Oliver (2003) where they designed a highly non-linear 1D heterogeneous reservoir problem where a single-phase transient flow problem was chosen. MCMC simulations were performed to generate a sequence of realisations that are samples from the target probability density accepted with the Metropolis-Hastings criterion. Even with a sequence of 320 million realisations, a correlation length of over 100 million iterations (shown by vertical grey dotted lines in Figure 8.1) was observed indicating a slow mixing rate of the Markov chain. It typically took at least 100 million iterations (perturbations) to go from high to low misfit values. Thus, it is important to design MCMC approaches that are able to deal efficiently with the complex and multimodal form of posterior distributions. It can be envisaged that more complex models would lead to greater multimodalities and standard Metropolis-Hastings samplers simply cannot cope with them.

The aim of this chapter is to address the issue of multimodality in complex geophysical models. This chapter presents the application of the Population MCMC (Pop-MCMC)

method to generate history matched reservoir models. The technique has been developed by Liang and Wong (2000) and successfully adopted in challenging domains such as computational biology (Calderhead and Girolami, 2009) and stereo matching (Kim et al., 2009; Park et al., 2007). The main feature of the algorithm is its ability to explore a complex multimodal posterior efficiently, as it can easily escape local optima due to the global large steps. From an optimisation perspective, this means that Pop-MCMC is able to locate the global optimum efficiently. Part of the work carried out here is presented in Mohamed et al. (2010c, 2011b).

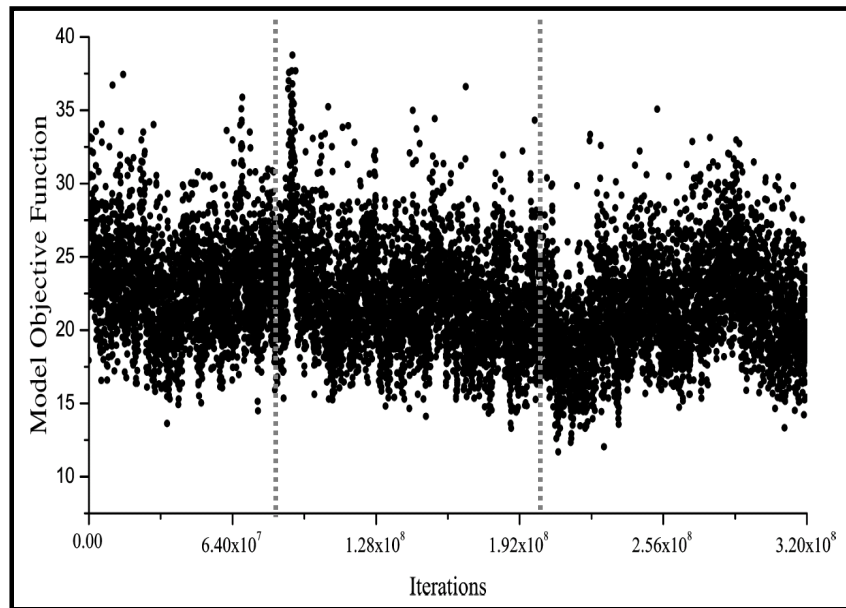


Figure 8.1: A sequence of 320 million realisations shows misfits correlation length of over 100 million iterations (Liu and Oliver, 2003)

8.1 Population Markov Chain Monte Carlo (Pop-MCMC)

Population MCMC (also known as evolutionary Monte Carlo) is a method for efficiently exploring and drawing samples from a complex, multimodal probability distribution by means of a series of smoothed intermediate distributions with varying “temperatures”. Separate chains are run for each tempered distribution and these are able to interact by exchanging positions, thus allowing chains to more easily escape local optima. Population-Based MCMC is a variant of a method originated by Swendsen and Wang (1986), and later developed by Geyer (1991). Geyer’s method, called parallel tempering (PT), aims to overcome the slow mixing problems of traditional MCMC using a Metropolis-Hastings update. This algorithm is derived from the parallel tempering method developed by Geyer (1991) and aims to overcome the problem of slow mixing associated with the use of traditional single-chain Metropolis-

Hastings algorithms. Geyer (1991) defined a new target density on the product space $\pi^* = \pi(x_1) \pi_1(x_2)$, where π_1 is different but linked to π and swapped x_1 and x_2 through an exchange step. In contrast to optimisation methods such as Genetic Algorithms and Particle Swarm, which aim to find the single “best” fit to the data, MCMC algorithms are built upon solid statistical foundations and aim to sample probabilistically all sets of parameters with which the model output most likely describes the observed data.

A standard method of implementing Population Markov Chain Monte Carlo within the Bayesian framework is as follows. We assume we wish to sample from a posterior distribution defined on the real space

$$p(\theta | y) \propto L(y | \theta)\pi(\theta) \quad \text{Eq. (8.1)}$$

where $L(y | \theta)$ is the likelihood of the observed data, y , conditioned on the parameters, θ , that represents the product of all the possible measurements (refer to Eq. (3.2) and Figure 3.1). In other words how well the data supports the model. $\pi(\theta)$ is the prior distribution over the parameters. We first define an N -step temperature schedule, $t = (t_1, \dots, t_N)$ with $t_1 < \dots < t_N = 1$. Note that for the metaphor of temperature to make sense, the parameter schedule t is actually inversely proportional to temperature, with t_1 considered a high temperature and $t_N = 1$ considered a low temperature. A sequence of distributions, corresponding to each step $i = 1, \dots, N$ on the temperature schedule, is then constructed

$$p(\theta_i | y) = \frac{L(y | \theta_i)^{t_i} \pi(\theta_i)}{Z_{t_i}} \quad \text{Eq. (8.2)}$$

where θ_i will be considered the position of the Markov chain running at temperature, t_i , and Z_{t_i} is some, usually intractable, normalising constant called the partition function

$$Z_{t_i} = \int L(y | \theta_i)^{t_i} \pi(\theta_i) d\theta_i \quad \text{Eq. (8.3)}$$

We therefore have a series of probability distributions which runs from a posterior distribution, which is possibly difficult to explore, to the prior distribution, which is smooth and easy to explore. One can picture a multimodal target distribution at $t_N = 1$, which melts at higher temperatures so that the distributions at $t_i < 1$ are easier to draw samples from. The resulting distribution at each temperature is explored using an individual Markov chain, so that the total number of Markov chains running simultaneously is N . In Population Markov Chain Monte Carlo a product distribution is

considered when moving individual chains, thus taking the entire population of chains throughout the temperature schedule into account. We therefore sample from

$$p(\Theta | y) = \prod_{i=1}^N p(\theta_i | y) = \frac{1}{Z_t} \prod_{i=1}^N L(y | \theta_i)^{t_i} \pi(\theta_i) \quad \text{Eq. (8.4)}$$

where Θ is the population of Markov chains, $\theta_1, \dots, \theta_N$ at the temperatures, t_1, \dots, t_N respectively. The (intractable) normalising constant is now

$$Z_t = \prod_{i=1}^N Z_{t_i} \quad \text{Eq. (8.5)}$$

Although we make no direct use of the normalising constant in this thesis, it is relevant for model comparison. Calderhead and Girolami (2009) show how to use the concepts of thermodynamic integration through population MCMC to evaluate this constant.

We note that other sequences are possible, but in this Bayesian setting we fix a geometric path between the prior and the posterior, by raising the likelihood term to a power between 0 and 1. An investigation into the optimal path is presented in Calderhead and Girolami (2009). Markov chains explore the distributions induced by the temperature schedule and these chains may also interact with one another by swapping positions across temperatures. The algorithm proceeds as in Algorithm 8.1 if we use the componentwise Metropolis-Hastings.

There is a trade-off between the exploration and convergence of the algorithm. The rates between global moves and local moves (mutation) can be controlled by adjusting mutation rate p_m . The chain with the lowest temperature is used for parameter estimation and inferences while other chains are useful for calculating Bayes factors for Bayesian model comparisons of a statistical hypothesis to be able to rank a set of plausible model structures based on the experimental evidence available. Thus, there is no waste in the algorithm (Calderhead and Girolami, 2009).

Algorithm 8.1 Pop-MCMC Algorithm (Componentwise Metropolis-Hastings Version)

1. Assign starting positions to each chain in a population, $\Theta = (\theta_1, \dots, \theta_N)$
2. Define a temperature ladder attached to the population, $(\Theta, t) = (\theta_1, t_1, \dots, \theta_N, t_N)$
3. *iteration number* = 0
4. Repeat (one iteration)
 - a. If $U \sim [0,1] < p_m$ (where p_m is sometimes known as the mutation rate) then
 - i. Apply local move (mutation) to each chain in the population
 - For $i = 1$ to N do
 - Select a Markov chain θ_i for the i^{th} chain from the population Θ
 - For $j = 1$ to d (perform componentwise Metropolis-Hastings)
 - Propose a new position θ'_{ij} for the i^{th} chain and j^{th} component of vector θ'_i and determine whether accept it or not with probability p_m (Metropolis-Hastings rule)
 - End for
 - End for
 - Else
 - ii. Apply crossover to each chain in the population
 - b. Try to exchange θ_i and θ_j for N pairs (i,j) , with i sampled uniformly on $(1, \dots, N)$ and $j = i \pm 1$ with probability $p_e(\theta_j, \theta_i)$ where $p_e(\theta_{i+1}, \theta_i) = p_e(\theta_{i-1}, \theta_i) = 0.5$ and $p_e(\theta_1, \theta_2) = p_e(\theta_N, \theta_{N-1}) = 1$
 - c. *iteration number* = *iteration number* + 1
5. Until chains converge or *iteration number* = *maximum number of iterations*
6. Keep the chain with the lowest temperature

Figure 8.2 illustrates the idea in which $\pi_{N-i+1} = L(y|\theta_i)^i \pi(\theta_i)$ is the distribution for chain i . The chain with the lowest temperature, the target distribution, is shown as $\pi_1 = L(y|\theta_N)^{t_N} \pi(\theta_N) = L(y|\theta_N) \pi(\theta_N) = p(\theta_N|y)$ since $t_N = 1$. For the chain with the highest temperature, the prior distribution, $\pi_N = L(y|\theta_1)^{t_1} \pi(\theta_1) = \pi(\theta_1)$, is nearly flat as represented, where the heights of boundaries between local optima are very low. Consequently, the samples in such a chain can freely move by comparison to the samples in a chain with low temperature. By swapping these higher temperature configurations with the configuration of a low temperature of the desired density, the low temperature simulation can be assisted to sample configurations much more efficiently than with local Metropolis updates only. This results in faster mixing rate between samples, and allows escape from local optima. Usually, the Metropolis step in each chain is performed with local steps trying to achieve high acceptance rates. The crossover operator is done for example by choosing different chains and swap over part of them (Calderhead and Girolami, 2009). The three types of moves in Pop-MCMC are mutation, exchange and crossover and are described below. These moves stem from the

genetic algorithm and are modified to fit the MCMC framework (Del Moral and Doucet 2003; Kim et al., 2009; Liu 2001; Liang and Wong, 2000, 2001; Park et al., 2007).

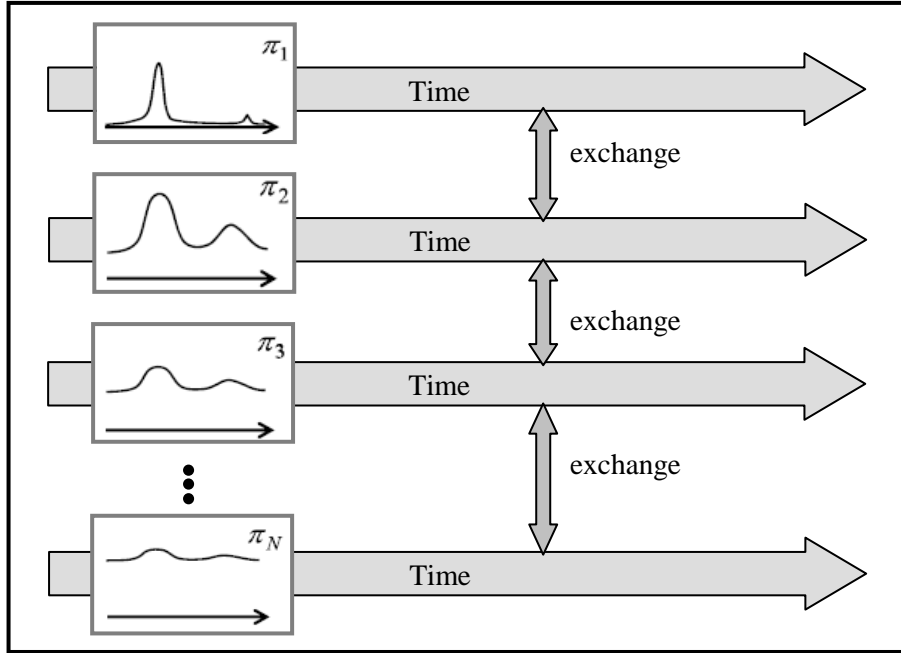


Figure 8.2: Representation of Population MCMC where π_N is the prior distribution with the highest temperature and π_1 the posterior distribution with the lowest temperature (Kim et al., 2009)

8.1.1 Local Metropolis Move (Mutation)

A random Markov chain position, θ_i , is selected from the population Θ , and a random vector is added to it to create a new proposed position, θ'_i . Thus a new population is defined as $\Theta' = (\theta_1, \dots, \theta'_i, \dots, \theta_N)$, which is then accepted with probability $\min(1, r_m)$ according to the Metropolis-Hastings rule,

$$\begin{aligned}
 r_m &= \frac{p(\Theta' | y)T(\Theta | \Theta')}{p(\Theta | y)T(\Theta' | \Theta)} \\
 &= \frac{\frac{1}{Z_t} \left[L(y | \theta_1)^{t_1} \pi(\theta_1) \times \dots \times L(y | \theta'_i)^{t_i} \pi(\theta'_i) \times \dots \times L(y | \theta_N)^{t_N} \pi(\theta_N) \right]}{\frac{1}{Z_t} \left[L(y | \theta_1)^{t_1} \pi(\theta_1) \times \dots \times L(y | \theta_i)^{t_i} \pi(\theta_i) \times \dots \times L(y | \theta_N)^{t_N} \pi(\theta_N) \right]} \times \frac{T(\Theta | \Theta')}{T(\Theta' | \Theta)} \\
 &= \frac{L(y | \theta'_i)^{t_i} \pi(\theta'_i)}{L(y | \theta_i)^{t_i} \pi(\theta_i)} \times \frac{T(\Theta | \Theta')}{T(\Theta' | \Theta)}
 \end{aligned} \tag{8.6}$$

where $T(\cdot | \cdot)$ denotes the probability of transition from one population to another. A common choice for the transition density T is a Gaussian centred on the current position

of the chain, which is symmetric and thus allows the transition densities in the above equation to be cancelled.

8.1.2 Exchange Move

This is similar to a standard exchange move in temperature based Monte Carlo methods. A new population Θ' is created by swapping the positions of two chains, θ_i and θ_j on the temperature ladder so that

$$(\Theta', t) = (\theta_1, t_1, \dots, \theta_j, t_i, \dots, \theta_i, t_j, \dots, \theta_N, t_N) \quad \text{Eq. (8.7)}$$

The new population is accepted with probability $\min(1, r_e)$ according to the Metropolis-Hastings rule:

$$\begin{aligned} r_e &= \frac{p(\Theta' | y)T(\Theta | \Theta')}{p(\Theta | y)T(\Theta' | \Theta)} \\ &= \frac{\left[L(y | \theta_j)^{t_i} \times L(y | \theta_i)^{t_j} \right]}{\left[L(y | \theta_i)^{t_i} \times L(y | \theta_j)^{t_j} \right]} \times \frac{T(\Theta | \Theta')}{T(\Theta' | \Theta)} \end{aligned} \quad \text{Eq. (8.8)}$$

where many of the terms, including the normalising constants, have conveniently cancelled out as shown previously for a local Metropolis step. Usually, the two selected chains are chosen to be direct neighbours in the temperature ladder to increase the likelihood of the interaction being accepted.

8.1.3 Crossover Move

This step is used as an alternative to the Metropolis local update. There are a few variations on the crossover operator. A chain, θ_i , is selected uniformly from a population, Θ . A second, different chain, θ_j , is also selected, either at random or for example with a probability proportional to its current likelihood. Two new chain positions, θ'_i and θ'_j , are then produced by the so-called one-point, k-point or adaptive crossover. The positions of the new chains replace the old positions to form a new population, Θ' , which is then accepted or rejected according to a standard acceptance probability. The one-point crossover takes place by uniformly selecting a crossover point, c , from $(1, \dots, (d-1))$ where d is the dimensionality and then swapping all the values in the vectors θ_i and θ_j which occur after position c as shown in Eq. (8.9).

$$\begin{bmatrix} \theta_i = (p_{i1}, \dots, p_{ic}, p_{i(c+1)}, \dots, p_{iD}) \\ \theta_j = (p_{j1}, \dots, p_{jc}, p_{j(c+1)}, \dots, p_{jD}) \end{bmatrix} \rightarrow \begin{bmatrix} \theta'_i = (p_{i1}, \dots, p_{ic}, p_{j(c+1)}, \dots, p_{jD}) \\ \theta'_j = (p_{j1}, \dots, p_{jc}, p_{i(c+1)}, \dots, p_{iD}) \end{bmatrix} \quad \text{Eq. (8.9)}$$

In the k -point crossover operator, k points are chosen randomly (two-point crossover, three-point crossover, etc). The k -point crossover is similar except there are multiple uniformly selected crossover points, dictating which parts of the vector should be swapped. The adaptive crossover is more complicated and the reader is referred to Liang and Wong (2000) for the details.

More details about Population MCMC methods can be found in Laskey and Myers (2003), Iba, Y. (2001), Hukushima and Iba (2003), Hukushima and Nemoto (1996), Del Moral et al. (2006), and Cappé et al. (2004).

8.2 Numerical Experiments

In this thesis, a MATLAB code developed at the Inference Group within the Department of Computing Science at Glasgow University (Calderhead and Girolami, 2009) is used to carry out this work. The code is adjusted and coupled with a reservoir simulator and misfit calculation routine for objective function evaluation. Numerical tests are firstly performed on simple analytical examples presented in previous chapters before examining the Population MCMC method on reservoir application for consistency purposes. For all the analytical tests the burn-in period was set to 2000 iterations with a further 1500 iterations to collect the posterior samples for parameter estimates.

8.2.1 Numerical Example Test 1

In this example, we test Pop-MCMC algorithm with the target density defined in Section 6.4.1. Pop-MCMC sampling runs chain with 10 different temperature schedules. Pop-MCMC samples are shown as purple points in the contour plot in Figure 8.3(a). Kernel density estimate using the sampled points produced estimates shown in Figure 8.3(b) and which shows samples effectively came from target density since it is similar to the shape of target density. Figure 8.3(c) shows the parameter width for each of the two parameters and we can see that the parameter widths decrease as the temperature schedule increases (moving to a cool state).

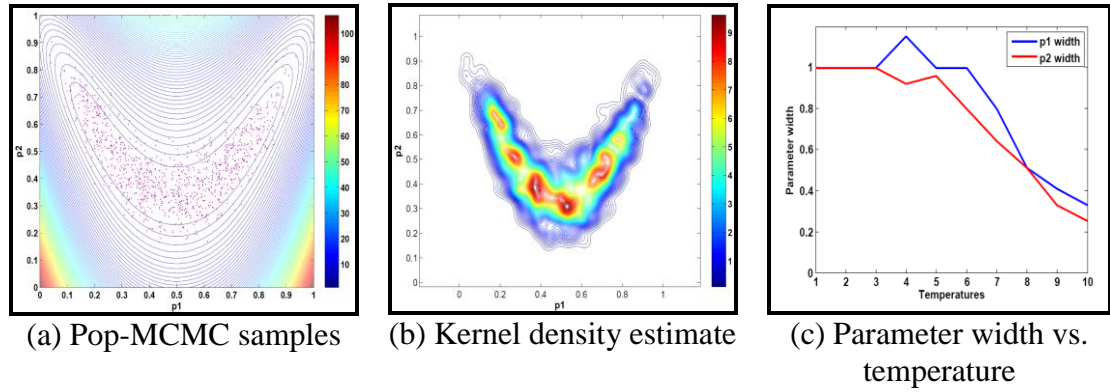


Figure 8.3: Pop-MCMC application on example test 1

8.2.2 Numerical Example Test 2

In the second example, we test the Pop-MCMC algorithm with the target density defined in Section 6.4.2. Pop-MCMC sampling runs chains with 10 different temperature schedules. Similarly to the previous example, Pop-MCMC samples are shown as purple points in the contour plot in Figure 8.4(a). Kernel density estimate using the sampled points produced is shown in Figure 8.4(b). This shows samples effectively came from target density suggesting that sampling with Pop-MCMC was effective and contained reasonable information to estimate quantities of interests and produce the connected region of high probability that has the shape of a thin crescent. Figure 8.4(c) shows the parameter width for each of the two parameters and again we can see that the parameter widths decrease as the temperature schedule increases. We note that the number of temperatures only affects the acceptance rate. Increasing the number of members in each population within a chain increases only the number of samples.

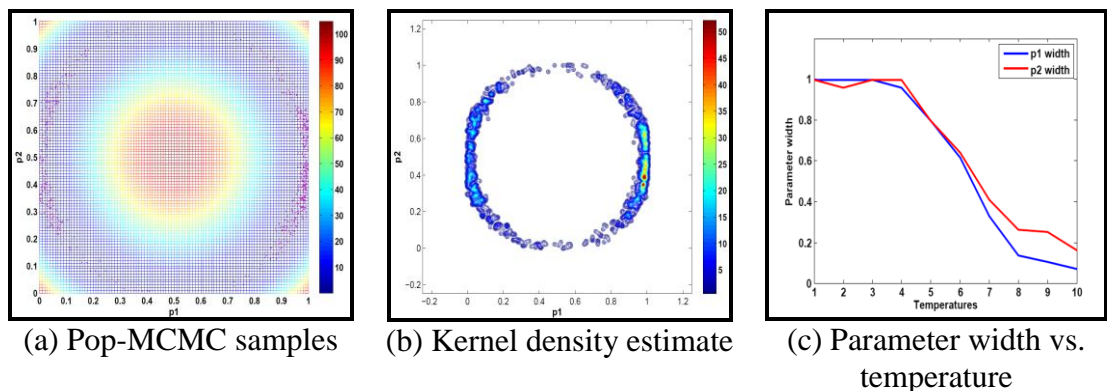


Figure 8.4: Pop-MCMC application on example test 2

8.2.3 Numerical Example Test 3

In the third example, we test the Pop-MCMC algorithm with the target density defined as the negative log of the objective function defined by the non-convex, non-linear, multimodal Rastrigin function detailed in Section 6.8.2. The application of Pop-MCMC for the 2-parameter case is shown in Figure 8.5 where the purple points shows 1500 posterior samples. It is clear that the points are sampled from the many minima shown clearly in the contour plot. This is a useful characteristic since in reservoir history matching we need to locate as many minima as possible in complex response surfaces to be able to produce reliable forecasts. The 3D view of the sampling in Figure 8.5(b) shows this complexity of surface. The similar decrease in the parameter width is shown in Figure 8.5(c) where small steps are needed for the coolest temperature (temperature number 10) for higher acceptance rates in narrow regions while in the hottest one the free movement of indicated with large parameter widths.

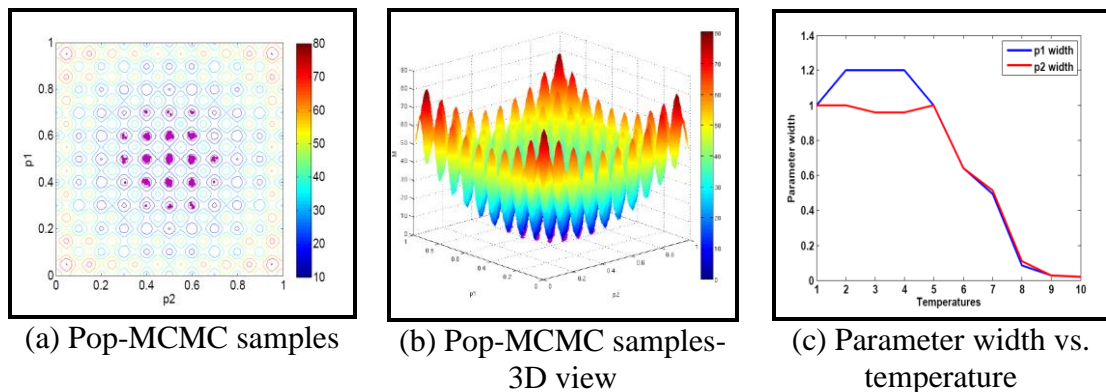


Figure 8.5: Pop-MCMC application on 2-parameter Rastrigin function

The application of Pop-MCMC for the 3-parameter case is shown in Figure 8.6 where the points are the 1500 posterior samples as shown in Figure 8.6(a). The red colour points in the 3D view shows the models with misfits below or equal to 3 and pink points are models where misfits > 3 . It is clear that the points are clustered around minima shown as red points in different locations. The 3D view of the sampling in Figure 8.6(b) shows this complexity of surface. The corresponding decrease for the three parameters widths for the last iteration per temperature is shown in Figure 8.6(b).

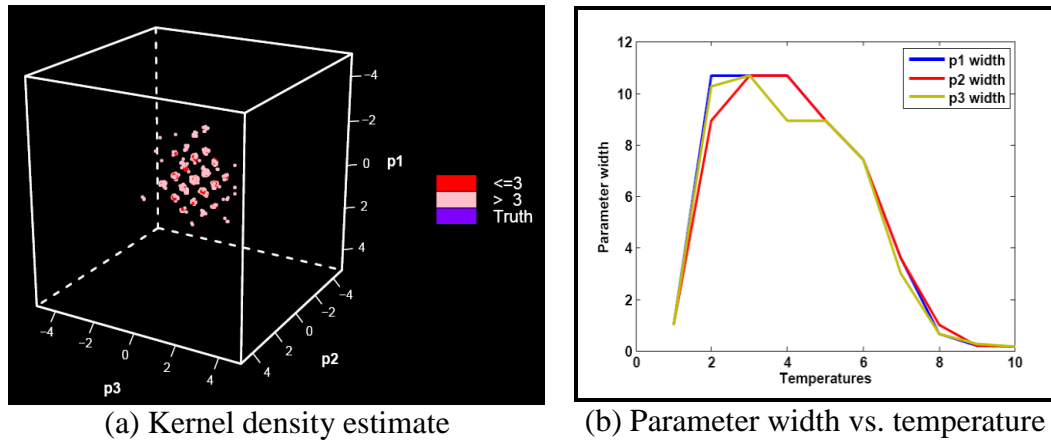


Figure 8.6: Pop-MCMC application on 3-parameter Rastrigin function

8.2.4 Numerical Example 4

An application for a biological system is noted here from Calderhead and Girolami (2009) where a multidimensional representation of a complex posterior distribution is shown in Figure 8.7(a). The same posterior distribution shown from the top is in Figure 8.7(b) where the blue colour represents the low density regions and the red colour represents the high density regions. Running the standard Metropolis-Hastings samplers with 20 independent chains is shown in the same plot where the cross symbol denotes the start of the chains then follows a path to circles denoting the end positions of the chains. When a chain starts from a high density region position near the global maximum it is plausible to climb up to the hill. If sampling start somewhere else it tends to follow the ridges to higher density areas and if it starts far away, then it goes somewhere else in parameter space, so the chains do not converge as seen in the plot. It is noted that adapted parameter widths (stepsizes) were used to get high acceptance rate (Gilks et al., 1996; Gelman et al., 2004). In order to obtain an accurate sample from the posterior distribution, the algorithm is required to adequately explore the parameter space, visit all high density regions and converge to the target distribution. In the advanced Pop-MCMC technique instead of sampling single independent chains a population of chains that communicate with each other are used. In this case if one chain finds a high density region than another, then it can move towards that region. This is achieved by introducing a temperature ladder and thus different chains exploring different distributions in parallel rather than sampling just a single posterior distribution. Thus, a range of intermediate temperatures with a typical easy-to-sample prior distribution are usually used. In the application uniform starting positions from prior distribution are drawn. Pop-MCMC results are shown in Figure 8.8 where samples for

prior distribution have free movement shown and the chain with the lowest temperature is quickly focused on sampling the absolute maximum due to the interactions of samples. The samples for the chain with the lowest temperature can be used for parameter estimation, uncertainty quantification, and inferences.

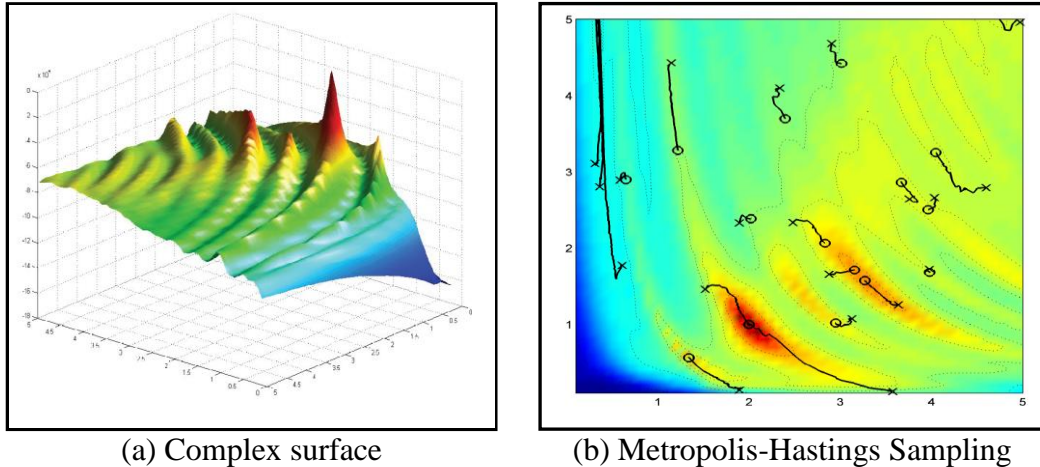


Figure 8.7: Multidimensional representation of complex biological system posterior distribution (a) and Metropolis Hastings sampling (b)

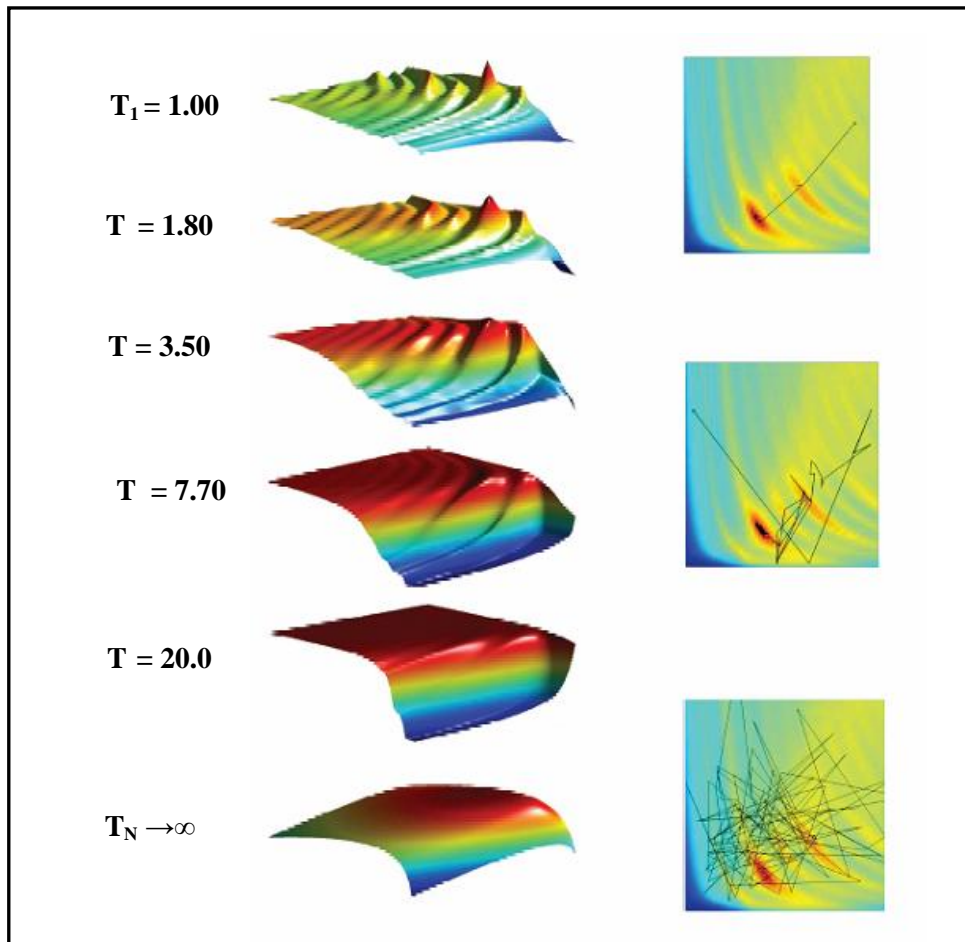


Figure 8.8: Population MCMC Sampling for biological system example (Source: Calderhead and Girolami (2009))

8.3 Petroleum Field Application

In previous section we have shown the motivation to apply this technique while testing the method in analytical examples and applications in other areas. In this section, we test the application of Pop-MCMC on the IC Fault Model (Tavassoli et al., 2004). This model characterises the kind of complexities that can arise in the posterior distributions of geological models. We use this model as a pilot study and intend to apply this methodology to more complex geologically realistic models in the future. As was shown in Section 7.2.2 with the standard HMC the sampling focused in areas close to or near starting points as depicted in Figure 7.20 similar to the results obtained in the biological system example with the Metropolis-Hastings in Figure 8.7(b), indicating ineffective sampling and therefore our forecasts will be questioned. Although, the idea is to couple the Hamiltonian dynamics within the Pop-MCMC structure, we only present in this thesis the application of Pop-MCMC without the coupling as a proof of concept in this example due to limited time. Further research on combining the HMC within the Pop-MCMC framework on challenging real field examples is undergoing.

8.3.1 Population MCMC Algorithm Setup Specifications for IC Fault Model

We ran a Pop-MCMC algorithm on the IC Fault model to sample a ladder of ten temperatures with spacing $t_i = (i/10)^5$, to allow for good mixing of chains. Each one of the ten chains has a single member in the population. The componentwise Metropolis-Hastings algorithm was used for sampling. Within one iteration, there are N evaluations of the likelihood, if all the model parameters are updated jointly, where N is the number of chains. In our case, given that each chain uses a componentwise version of Metropolis-Hastings, also called Metropolis within Gibbs (Ntzoufras, 2009), each iteration requires $d \times N$ (d is the number of model parameters) evaluations of the likelihood. We have carried out two runs. In the first run the burn-in period was set to 1000 iterations with a further 500 iterations to collect the posterior samples for parameter and prediction estimates. Each iteration consists of generating a new proposal for each temperature, and since we are using a Gibbs sampler, each proposal requires a function evaluation for each dimension, leading to 30 function evaluation per iteration. This is clearly not efficient, nor it is achievable for anything other than the smallest reservoir models. However, we can use the Gaussian process to build emulators for the simulation outputs, which lead to a dramatic reduction in expensive calls to the reservoir simulator and vastly improves efficiency, see for example Fillipone et al.

(2010). The total number of simulation is 45,000 (1500×30) computed with Eq. (8.10). In the second run the burn-in period was set to 2000 iterations with a further 1500 iterations to collect the posterior samples for parameter and prediction estimates. The total number of simulation is 105,000. We decided to run the chains with 1000 or 2000 burn-in samples to ensure full convergence of the chains to all potential modes. The 500 and 1500 posterior samples were adequate for illustrative purposes, showing full coverage of the posterior distribution. The number of posterior samples to be drawn depends on the statistical estimators of interest and accuracy of the estimate required. Standard Metropolis-Hastings methods generally cannot sample adequately from multimodal posteriors and their use could lead to extremely biased inferences and predictions. The proposal widths (proposal steps) are sampled from a normal distribution with mean of the current parameter and its variance. These parameters widths are adapted automatically, so that sampling improves with the number of iterations. The parameter width adapts depending on the current values of the chain. The initial widths used are (5, 1, 1) for $(k_{high}, k_{low}, throw)$.

$$\begin{aligned} \text{The total number of simulations} &= \text{the number of members in the population} \times \\ &\text{the number of chains (i.e. number of temperatures in the tempering scheme)} \times \\ &\text{the number of parameters } (d) \times \text{the number of iterations (burn-in and posterior)} \end{aligned}$$

Eq. (8.10)

8.3.2 *History Matching Results*

In Chapter 7 we compared the performance of Hamiltonian Monte Carlo with two stochastic optimisers (Particle Swarm Optimisation and the Neighbourhood Algorithm) on Teal South, a simple single well field in the Gulf of Mexico (Mohamed et al., 2010b).

We will compare Pop-MCMC with HMC and the two stochastic samplers (PSO and NA) on the IC Fault Model that exhibits a complex misfit surface as has been analysed in Chapter 7. The stochastic sampling algorithms were set up to be as similar as possible (refer to Section 7.2.1).

Bayesian inference allows us to obtain posterior distributions on parameters and predictive distributions. In order to compare results from a Bayesian method with non-Bayesian ones, some sort of threshold on the likelihood (negative misfit) based on

intervals obtained from the Bayesian method, is needed. In our case, we chose 25 as the threshold.

Figure 8.9 shows an equivalent set of results to Figure 7.18 and Figure 7.20 from Pop-MCMC with 500 posterior samples for the first run. Each plot represents the 500 samples obtained after burn-in for each temperature. Temp_1 corresponds to the prior, and clearly shows essentially uniform sampling over the parameter space. As the temperature decreases, we can see that the sampling localises onto the ribbon-like structure found from the database. Figure 8.10 shows an equivalent set of results from Pop-MCMC with 1500 posterior samples (Pop-MCMC-1500) for the second run. We can see that in this run the posterior samples were able to capture more or less the database ribbon-like structure shown in Figure 4.30 where it captures the samples at the corner that was not seen in Figure 8.9. The small differences that can be seen between the database models with $M \leq 25$ plot shown in Figure 4.30 and the Temp_{10} plot shown in Figure 8.9 and Figure 8.10 occur because Pop-MCMC importance samples, so that there are many lower misfit points in Temp_{10} in Figure 8.9 and Figure 8.10 than in Figure 4.30.

Similarly, if we restrict the misfits shown for the database, Pop-MCMC-500, and Pop-MCMC-1500 results to $M \leq 3$ as shown in Figure 8.11, we can see that Pop-MCMC is able to reach configurations of very low misfit value particularly apparent in Figure 8.11(c). The reason that Temp_{10} in Pop-MCMC obtains part of the ribbon-structure compared to the other intermediate higher temperatures is because model samples of misfits equal to 3 or less with lower misfits were given importance weights since these models will have more impact on the parameter being estimated than others. Since, these "important" values are emphasised by sampling more frequently, the estimator variance can be reduced. Note that, the simulation results are weighted to correct for the use of the biased distribution, and this guarantees that the new importance sampling estimator is unbiased where the weight is given by the likelihood ratio.

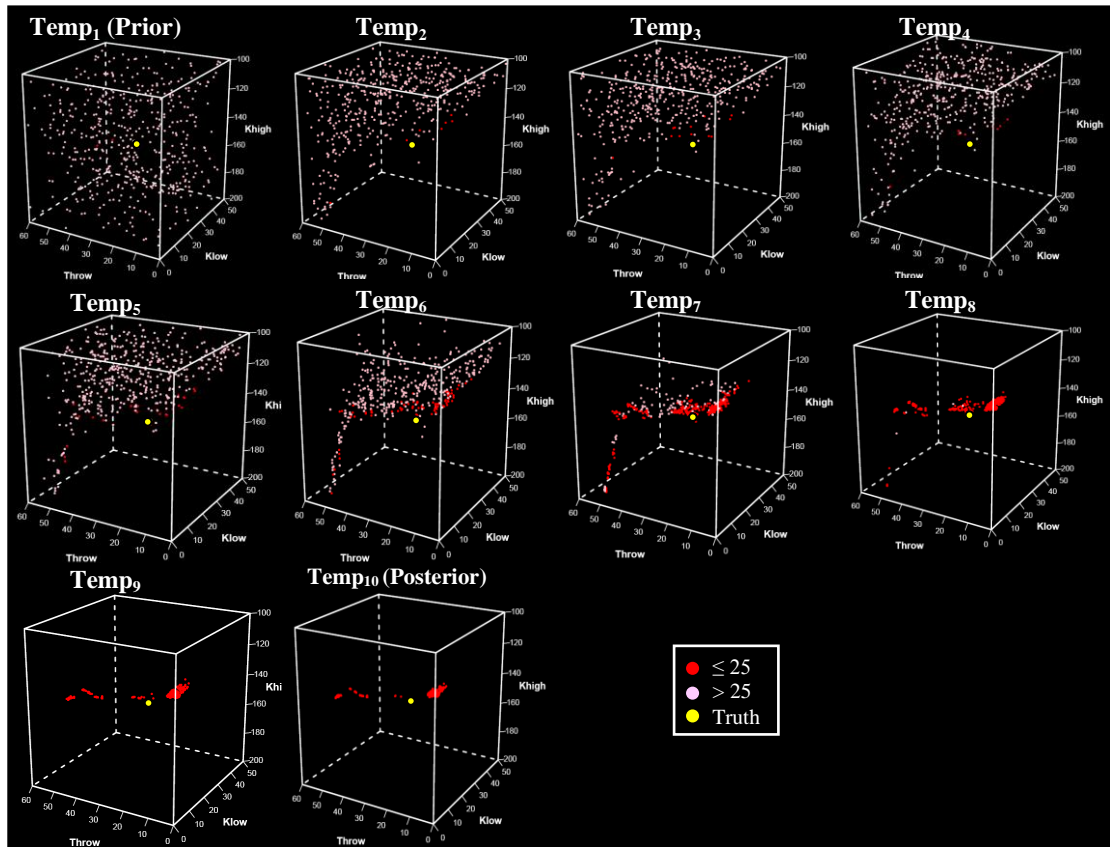


Figure 8.9: Pop-MCMC-500 sampling plot at various temperatures

The local Metropolis step proposals within each tempered distribution were selected using a general recommendation to achieve good mixing [see for example Gilks et al. (1996)], that is keeping the acceptance rate between 20% and 40%. Monitoring the exchange acceptance rate, which tells us whether the Markov chains are successfully jumping between the tempered distributions, shows high acceptance values indicating a good mixing of chains. Figure 8.12 shows the parameter widths for Pop-MCMC of 500 samples (Pop-MCMC-500) in Figure 8.12(b) and of 1500 samples (Pop-MCMC-1500) in Figure 8.12(c) using 10 temperature schedules. The jumps in the longer state have large parameter widths decreasing over the temperature value. This is clearly noticeable in the case similar to Figure 8.12(b) of 500 samples but with only 5 temperatures as shown in Figure 8.12(a). The jumps are larger in Figure 8.12(c) compared to Figure 8.12(b) since each temperature here has 1500 samples to use for estimating the mean of the width of the parameter with, rather than the 500 in the second. Yet, that result is not very different in terms of sampling performance as both obtain similar structure as shown in Figure 8.11 with differences seen in Figure 8.9 and Figure 8.10.

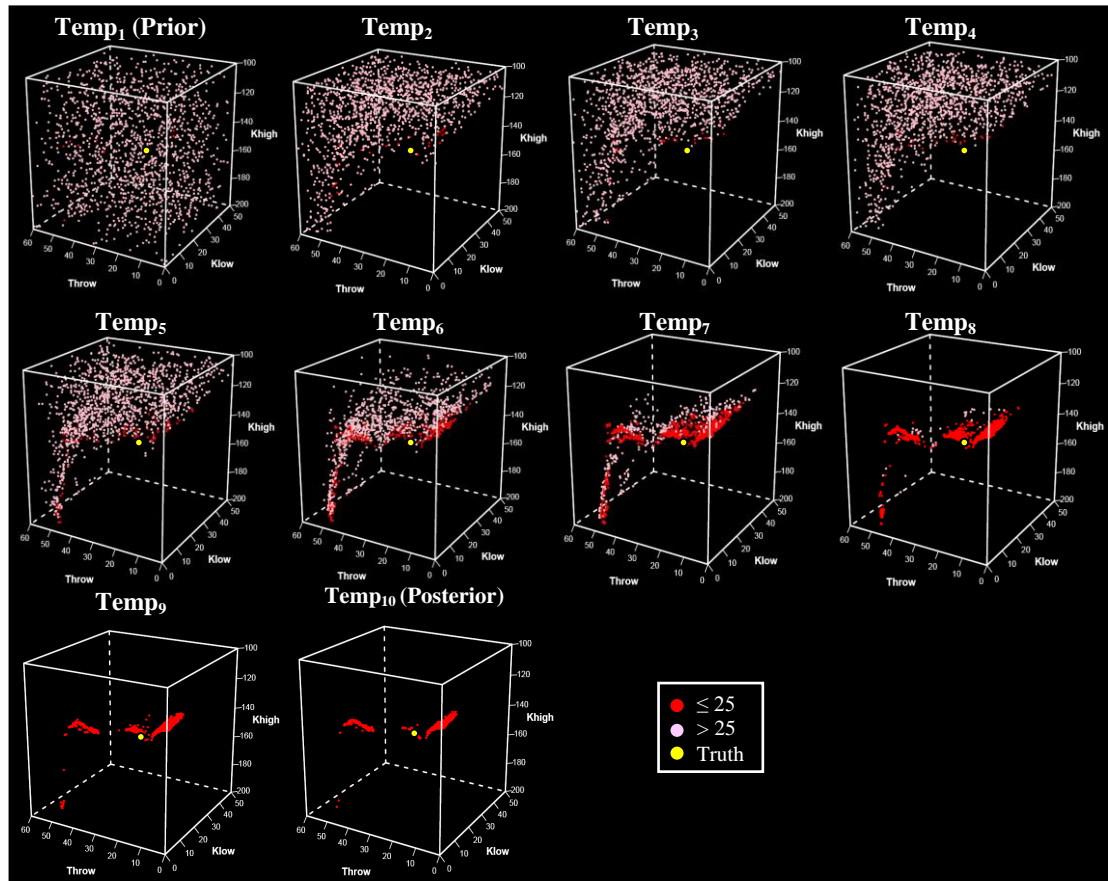
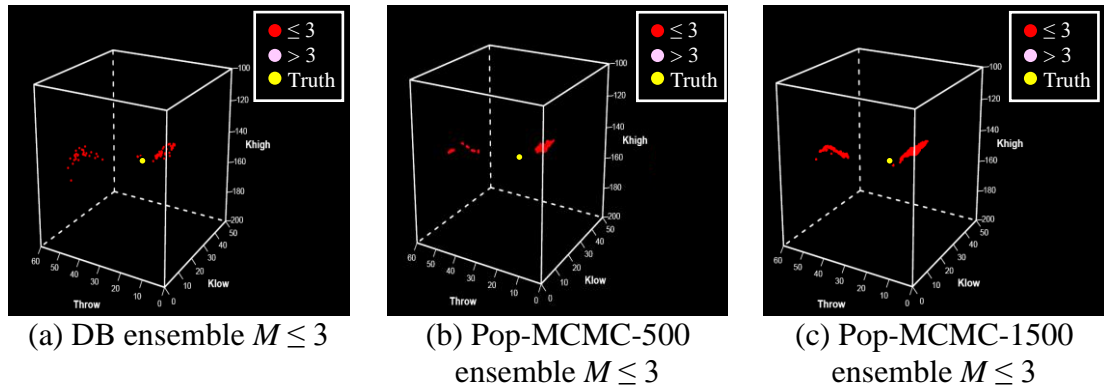


Figure 8.10: Pop-MCMC-1500 sampling plot at various temperatures

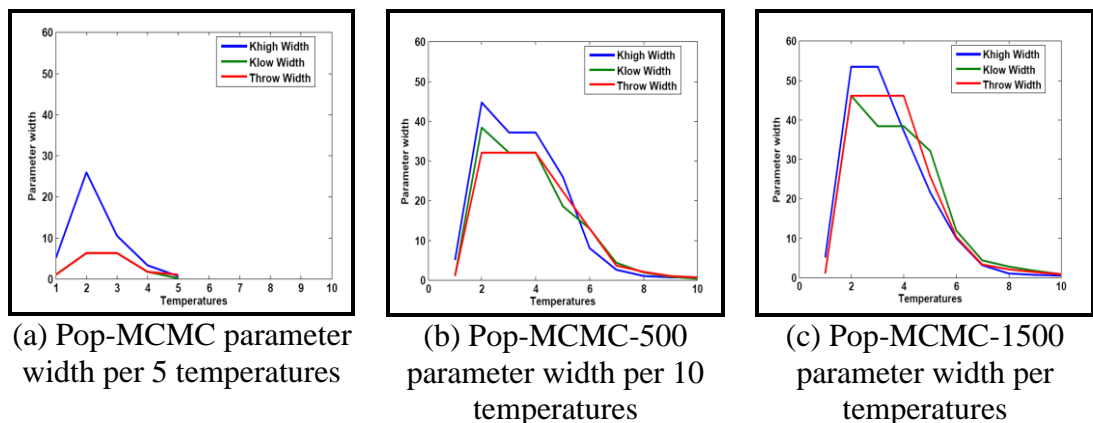


(a) DB ensemble $M \leq 3$

(b) Pop-MCMC-500 ensemble $M \leq 3$

(c) Pop-MCMC-1500 ensemble $M \leq 3$

Figure 8.11: DB ensemble in comparison to Pop-MCMC ensembles



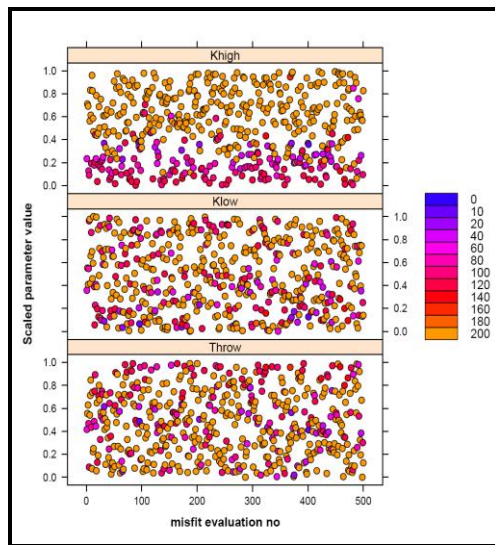
(a) Pop-MCMC parameter width per 5 temperatures

(b) Pop-MCMC-500 parameter width per 10 temperatures

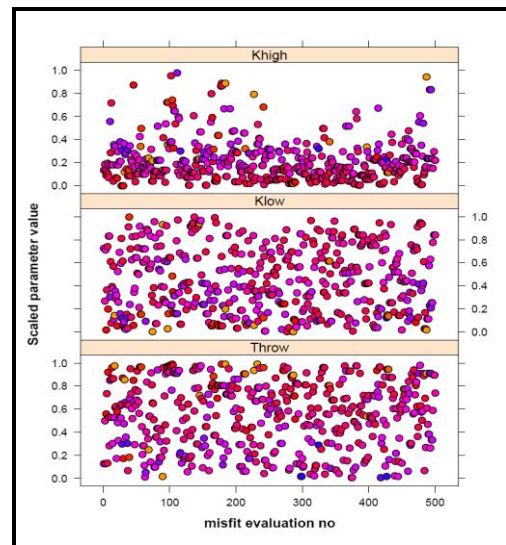
(c) Pop-MCMC-1500 parameter width per 10 temperatures

Figure 8.12: Pop-MCMC parameter width per temperature at the end of sampling

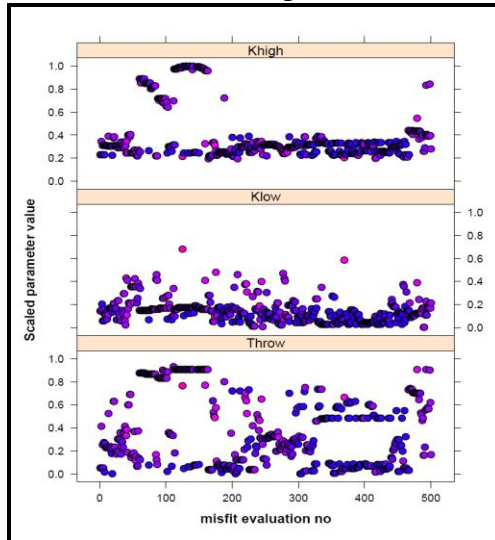
Sampling history evolutions of the three parameters are shown in Figure 8.13 for Pop-MCMC-500 and Figure 8.14 for Pop-MCMC-1500 for the selected four temperatures. Comparing the evolution for each run only we can see the performance of sampling and the quality of models (how well they fit the data) obtained during the course of posterior sampling in which models are colour coded according to the misfit. In the Pop-MCMC-1500 run there are many good quality models found as depicted as blue points in Figure 8.14(d) in comparison to Pop-MCMC-500 one in Figure 8.13(d)).



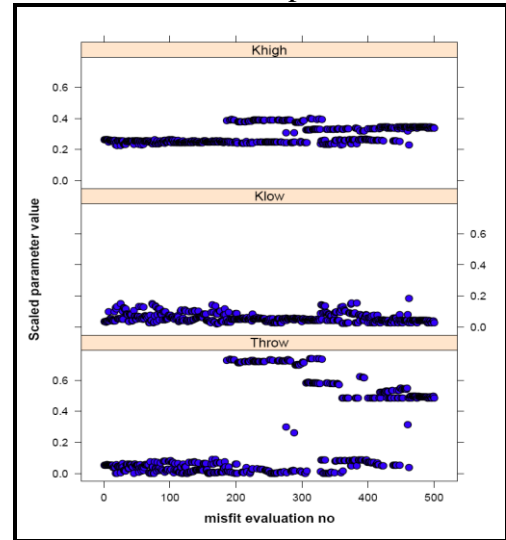
(a) Sampling history of Pop-MCMC-500 – Temp₁



(b) Sampling history of Pop-MCMC-500 – Temp₄



(c) Sampling history of Pop-MCMC-500 – Temp₇



(d) Sampling history of Pop-MCMC-500 – Temp₁₀

Figure 8.13: Sampling history of the chains with the 10 temperatures – Pop-MCMC-500

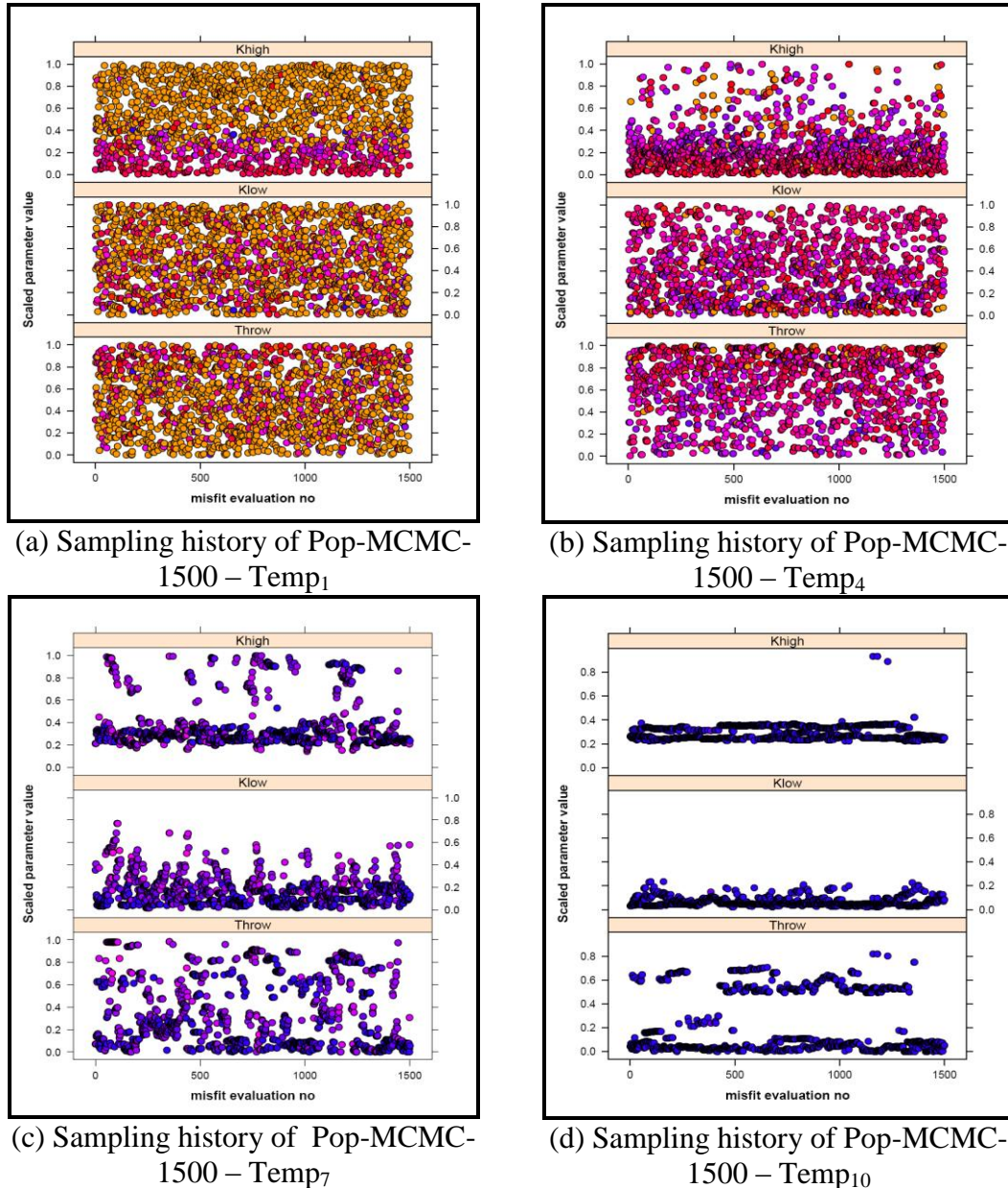
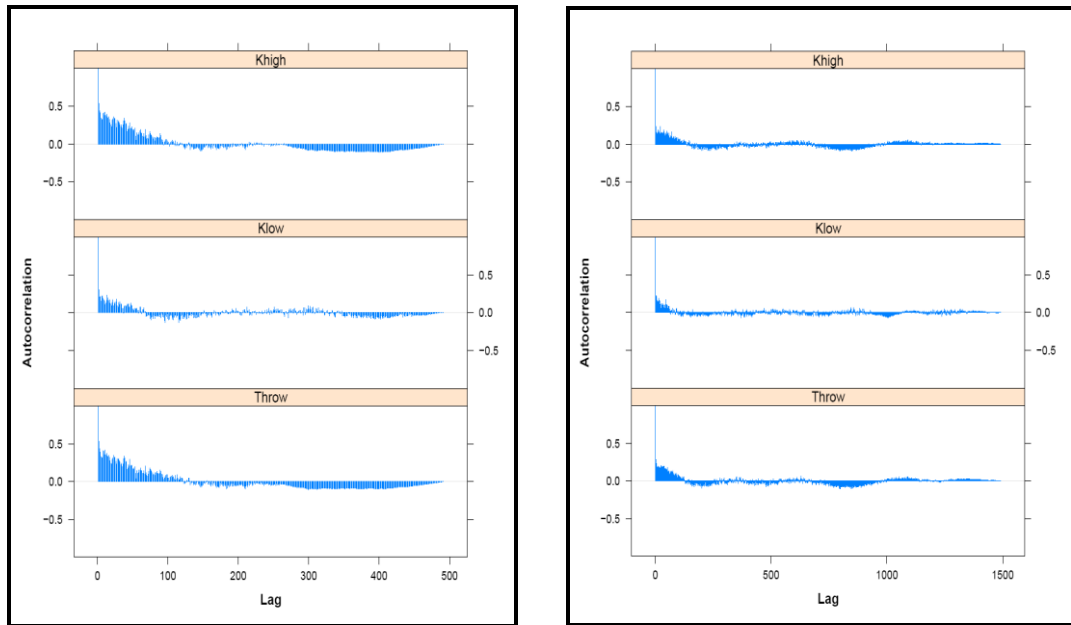


Figure 8.14: Sampling history of the chains with the 10 temperatures – Pop-MCMC-1500

8.3.3 Uncertainty Quantification

Before using the ensemble of models for quantifying the uncertainty and making predictions, we can look at the autocorrelation function within Temp₁₀ chains (the posterior samples that will be used in predictions) along the three parameters. Figure 8.15(a) illustrates the autocorrelation function (ACF) for Pop-MCMC-500 and Figure 8.15(b) shows the ACF for Pop-MCMC-1500. The chains have low correlation which indicates no random walk and thus the estimation of predictions is based on these samples.



(a) ACF for Pop-MCMC-500 – Temp₁₀ (b) ACF for Pop-MCMC-1500 – Temp₁₀

Figure 8.15: ACF for Temp₁₀ temperatures sampled chains

The Bayesian credible intervals for cumulative oil production from both Pop-MCMC runs and the database are shown in Figure 8.16. The results in particular Pop-MCMC-500 one, have obtained estimates for the uncertainty envelopes of total recovery that are very close to exhaustive UMC sampling in the database.

The true model has the minimal cumulative oil production. The whole point of quantifying uncertainty is to obtain confidence intervals for parameters and predictions. Data were generated as explained in Tavassoli et al. (2004); inference on the parameters, assuming that the model is correct, would converge to the true parameters in the limit of infinite data. In this case there is a slight mismatch between models, as the permeabilities used in the simulation to obtain the historical data are not fixed to the values used in the misfit evaluations.

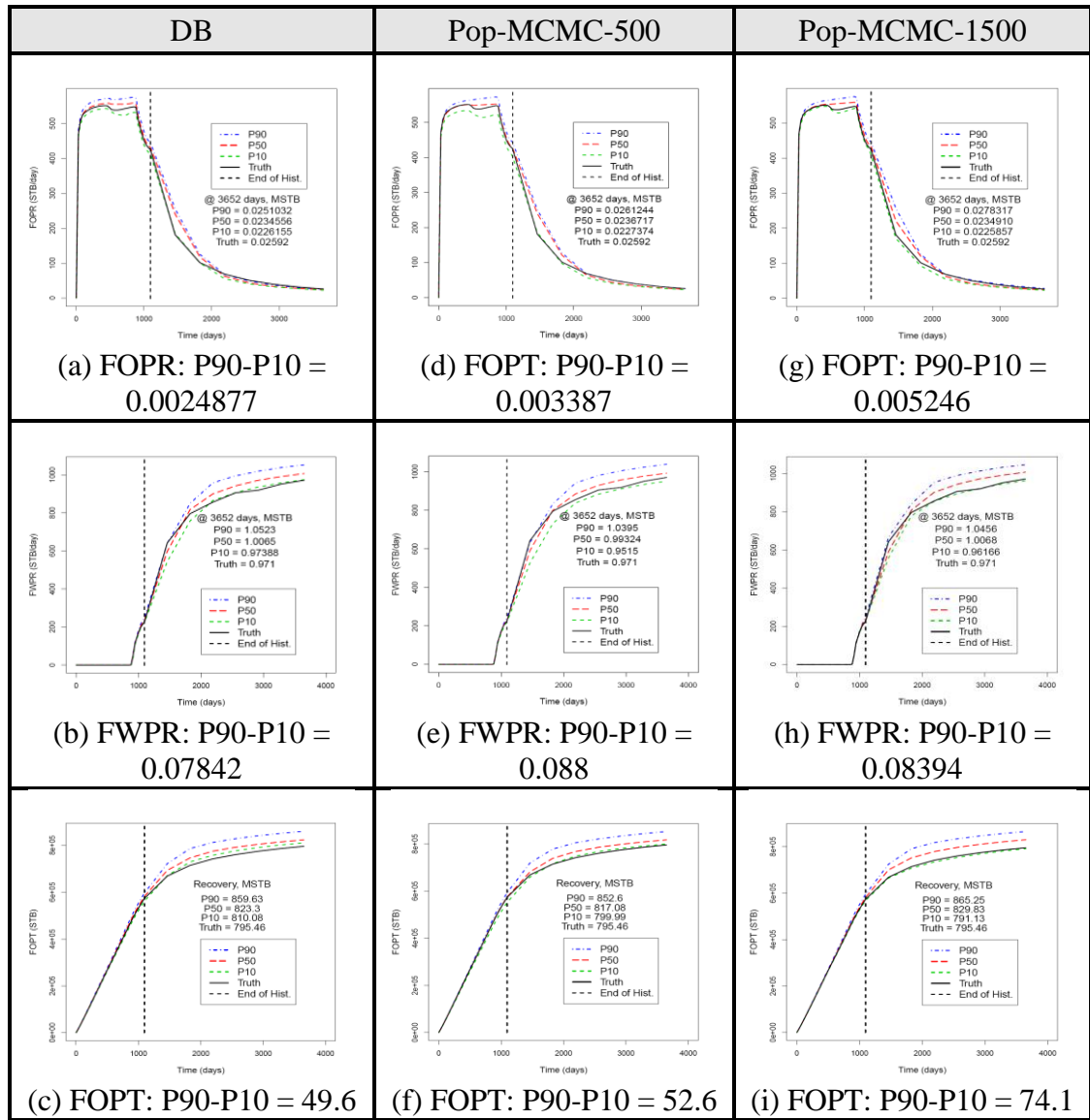


Figure 8.16: Comparison of Bayesian credible intervals for oil rate (MSTB/D), water rate (MSTB/D), and cumulative oil produced (MSTB) – database vs. Pop-MCMC-500 and Pop-MCMC-1500

8.3.4 Uncertainty Assessment Comparison with Other Techniques

We also show a comparison of the 'forecast misfit' and history misfit for each algorithm. The forecast misfit is computed using the known production for years 4 to 10 using the same expression used to evaluate the misfit during the history period in Eq. (4.7) while using a number of observations equal to 7 for the forecast period. For the stochastic sampling algorithms, the corresponding figure in the restricted ranges including the points that are resampled using the NAB algorithm (Sambridge, 1999b) is shown. Figure 8.17 is a comparison of history match and prediction quality guide where the lower-left corner represents the region where the models match historical data well and are good predictors. We note that the two clusters of models in this plot fall inside the ellipse indicating a set of reservoir models for which the prediction quality is similar to

the history match quality. However, the other cluster of models falling outside of the ellipse represents the ensemble of reservoir models for which the prediction quality significantly diverges from the history quality. Such unreliable reservoir models could yield very poor predictions in the forecast period (Walker and Lane, 2007). Thus, we want the predictions to be based on both good and poor history matched models for more accurate assessment of uncertainty.

The picture in Figure 8.18 shows the database forecast misfit versus history misfit with the restricted ranges for history misfit ≤ 25 and forecast misfit ≤ 50 . Figure 8.19 clearly shows how certain combinations of input parameters for the stochastic sampling algorithms leads to sampling that matches well (low history misfit) but forecasts poorly (NA1, Figure 8.19(a), PSO2, Figure 8.19(d)). Different choices of algorithm parameters lead to much better sampling and more reliable forecasts (NA2, Figure 8.19(b), PSO1, Figure 8.19(c)). The results from Pop-MCMC show very good history matches (most misfits less than 4 in Pop-MCMC-500 as depicted in Figure 8.19(e) and less than 10 in Pop-MCMC-1500 as shown in Figure 8.19(f)) and a very good spread of forecast misfits, demonstrating that the algorithm has been able to capture all the models that contribute to the forecast uncertainty.

Figure 8.20 shows a comparison of the Bayesian credible intervals for cumulative oil produced at the end of the forecast period for all the methods in which PSO2 and Pop-MCMC-500 obtain FOPT estimates close to the database one.

Figures 8.21, 8.22 and 8.23 show the 1D marginal distribution of the 3-parameter model obtained from the posterior samples ($Temp_{10}$) for Pop-MCMC-500 and Pop-MCMC-1500 (scaled to the range [0,1]) compared to the database marginal of the 3-parameter model where it is shown the truth value lies between low probability valleys. The results of k_{low} and throw parameters are very close to the database ones indeed and particularly so for k_{high} . The Pop-MCMC-1500 posterior density is very comparable to the database density one.

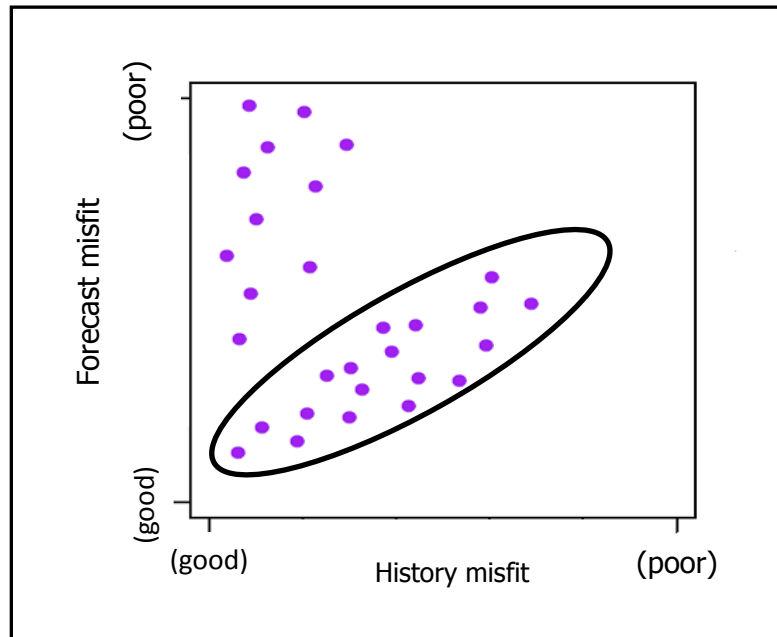


Figure 8.17: Comparison of history match quality and prediction quality guide

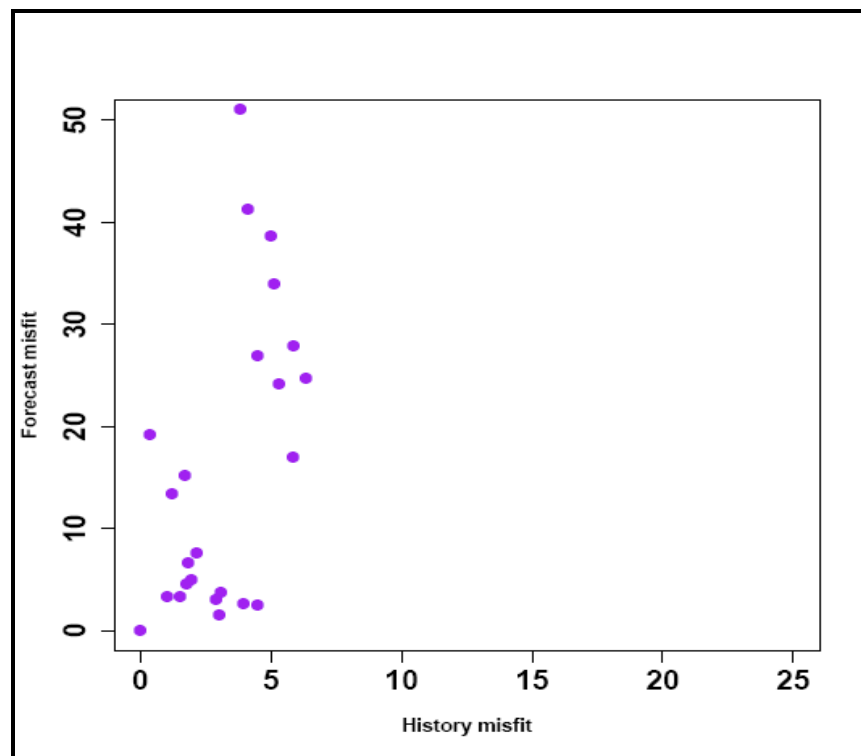
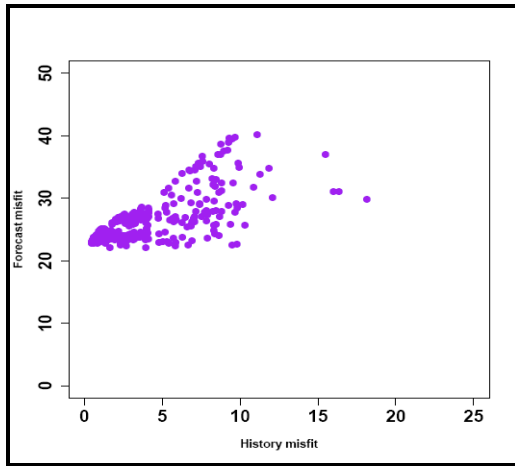
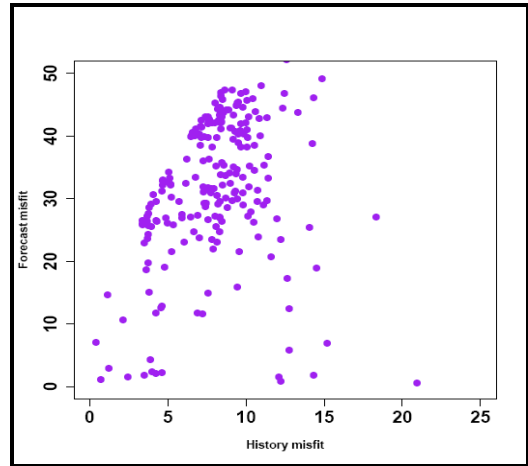


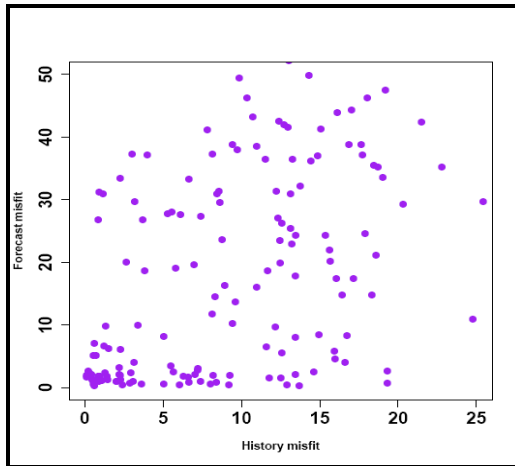
Figure 8.18: Forecast misfit (Mf) vs. history misfit (Mh): DB benchmark constrained in x and y ranges



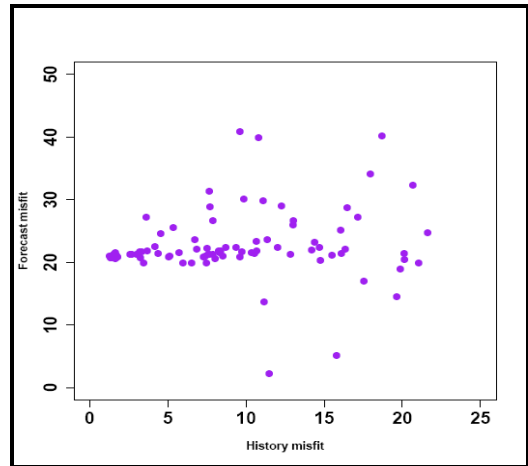
(a) NA1



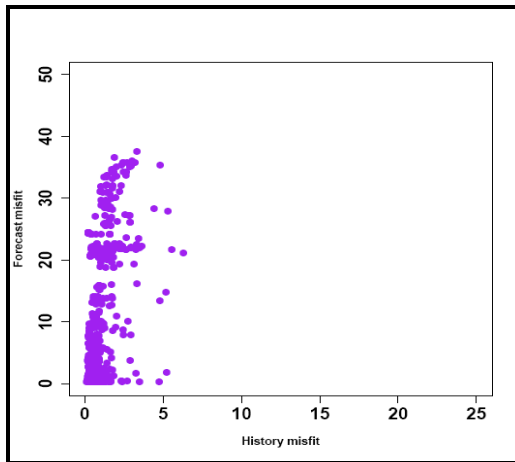
(b) NA2



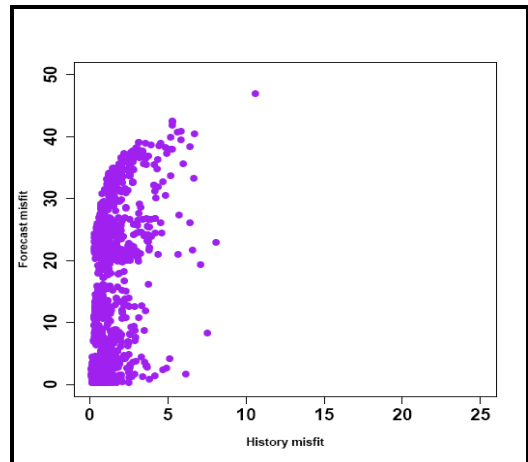
(c) PSO1



(d) PSO2



(e) Pop-MCMC-500



(f) Pop-MCMC-1500

Figure 8.19: Forecast misfit (M_f) vs. history misfit (M_h) for two runs for each technique

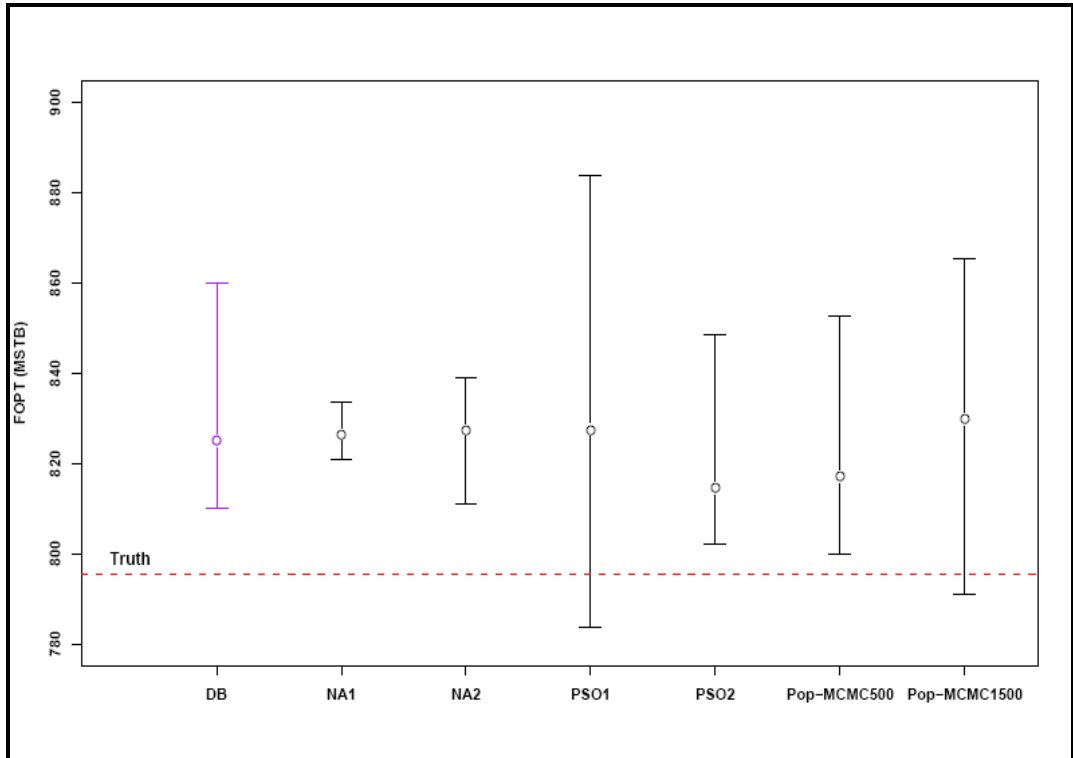


Figure 8.20: Prediction uncertainties derived from the database, NA1, NA2, PSO1, PSO2, Pop-MCMC-500, and Pop-MCMC-1500

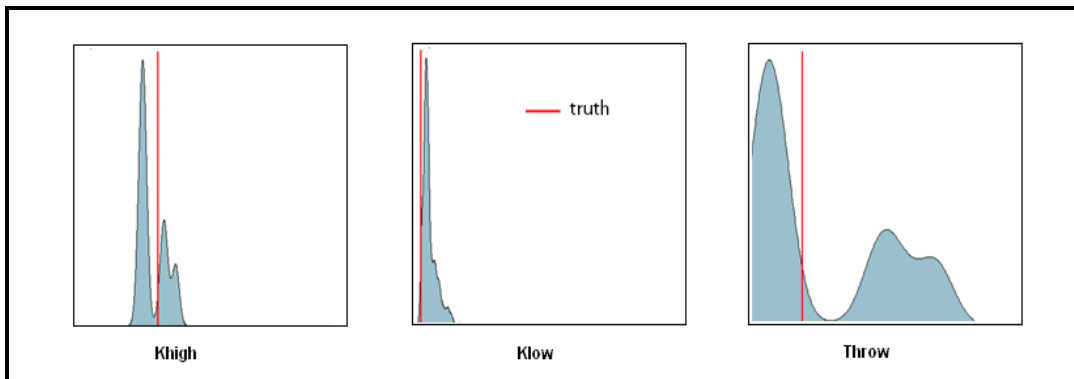


Figure 8.21: The 1-D marginal posterior distribution estimates for Pop-MCMC-500

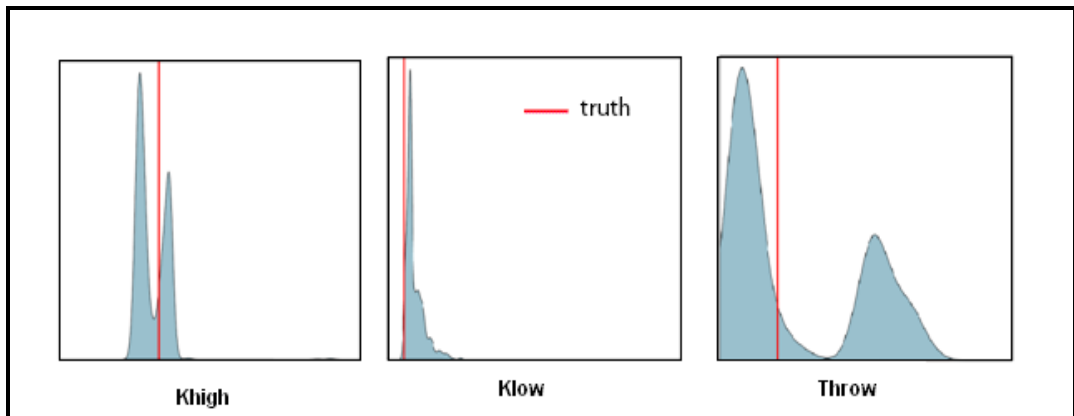


Figure 8.22: The 1-D marginal posterior distribution estimates for Pop-MCMC-1500

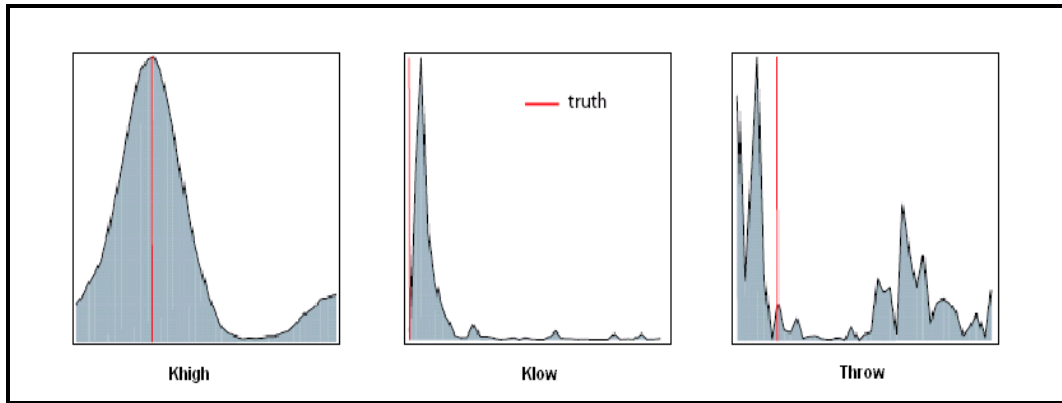


Figure 8.23: The 1-D marginal for DB

To check the accuracy of the forecasts, plotting the forecast probability estimates versus observed frequency of an event is a useful tool (Christie et al., 2005) as shown in Figure 8.24 for a similar problem in weather forecasting with a large number of observations where each data point represents the number of observations for that forecast indicated at that point.

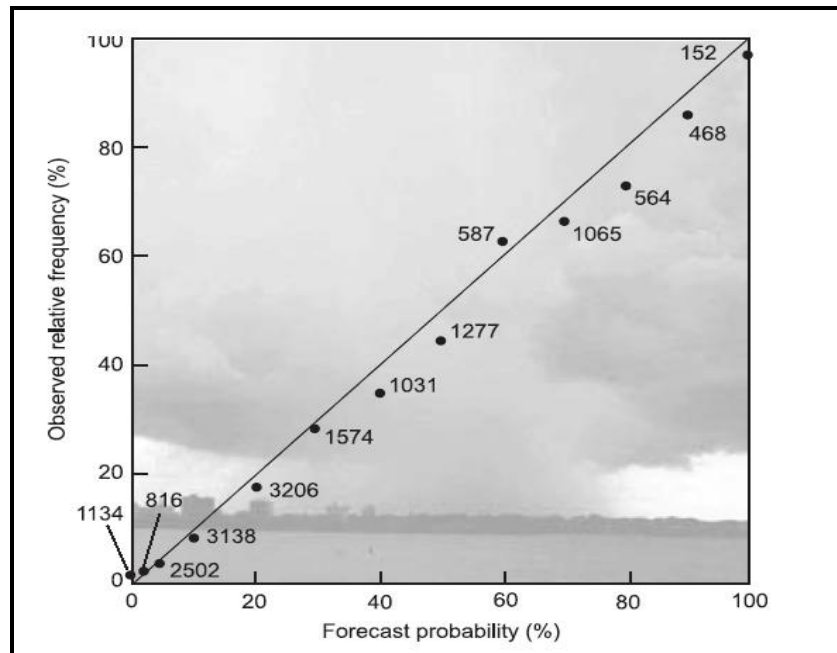


Figure 8.24: The calibration curve for weather forecasts (Source: Christie et al., 2005)

Figure 8.25 shows the equivalent calibration curve plot for selected runs of each technique. The plot shows illustrations for oil rate, water rate, water injection rate, water cut, and total oil recovery. Pop-MCMC results are shown in purple colour lines which are close to the black line. This indicates a close match to the database estimation particularly in oil rate and water cuts. In most other cases, PSO shows more accurate results while NA seems to overestimate uncertainty on three occasions.

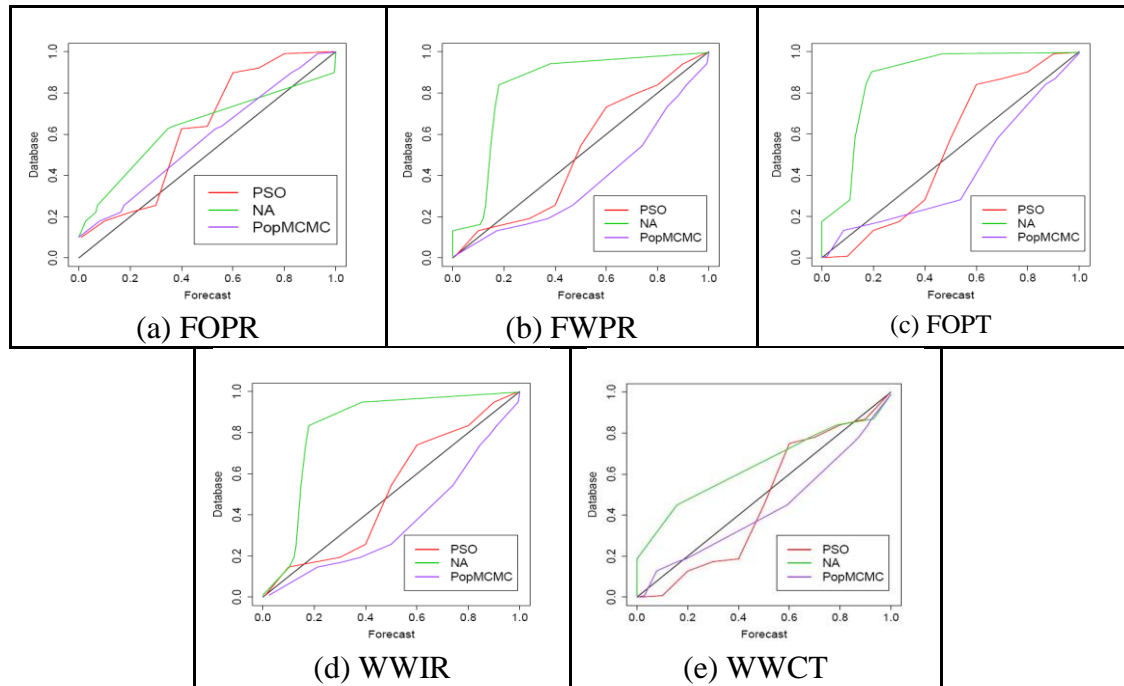


Figure 8.25: Calibration curve for selected runs for each of the stochastic methods in comparison with Pop-MCMC-500 at the end of the forecast period

For this study, we have not worried about the efficiency of the process as the IC Fault model is fast to run. However, there are a number of ways to reduce the number of forward simulations. Firstly, the use of a different MCMC sampler – for example Hamiltonian Monte Carlo (Duane et al, 1987) or Riemann Manifold HMC (Girolami and Calderhead, 2011) – in place of the componentwise Metropolis-Hastings sampler would reduce the number of simulations needed by a factor of 3 for the IC Fault Model. Secondly, it is relatively straightforward to use a Gaussian Process (GP) to emulate the simulator and reduce the number of expensive full simulations for Pop-MCMC particularly that the intermediate temperatures’ chains can use an estimated value for the misfit and may not require an exact simulation result. This work is under investigation with promising preliminary results as published in Filippone et al. (2010). Busby (2009), Busby and Feraille (2008), and Busby et al. (2007a, 2007b) tested a similar approach on the IC Fault model and PUNQ model (refer to the synthetic case study in Section 5.9) where they proposed a sequential strategy called hierarchical adaptive experimental design (HAED) to obtain an accurate emulator while using the least possible number of simulations (285 in Busby and Feraille (2008)). They concluded that the HAED method is superior to other standard state-of-the-art methodologies as well as providing a plausibly accurate approximation of the emulator

accuracy and an effective stopping criterion. Our preliminary results and theirs suggest that this is an interesting line of further research in combination with Pop-MCMC.

8.4 Chapter Summary

This chapter presents the application of a population MCMC technique to generate history matched models. The technique has been developed and successfully adopted in challenging domains such as computational biology and stereo matching, but has not yet seen application in reservoir modelling. In population MCMC, multiple Markov chains are run on a set of response surfaces that form a bridge from the prior to posterior. These response surfaces are constructed from the product of the prior with the likelihood raised to a varying power less than one. The chains exchange positions, with the probability of a swap being governed by a standard Metropolis accept/reject step, which allows for large steps to be taken with high probability.

We show results of Population MCMC on the IC Fault Model – a simple 3 parameter model that is known to have a highly irregular misfit surface and hence be difficult to match. Our results show that population MCMC is able to generate samples from the complex, multimodal posterior probability distribution of the IC Fault model very effectively. By comparison, previous results from stochastic sampling algorithms often focus on only part of the region of high posterior probability depending on algorithm settings and starting points.

Chapter 9 – Brugge Reservoir Model

History Matching: Comparison of Particle Swarm Optimisation and Ensemble Kalman Filter

In the present chapter we focus on testing the applicability of the novel Particle Swarm on history matching an SPE benchmark reservoir example from the Netherlands Organisation for Applied Scientific Research (TNO), and compare the results with the ensemble Kalman filter data assimilation method. The study has been conducted on the Brugge reservoir where the setup was particularly tailored for particle filter methods and the challenge for evolutionary and swarm intelligence methods lies in using an ensemble of prior geological realisations in which the more-global optimisation methods appeared to be at a disadvantage (Denney, 2009). We show that PSO can history match while honouring the data from the wells and can obtain comparable results with the EnKF in this study. This chapter has made two contributions (Mohamed et al., 2010a). Firstly, the combined use of particle swarm optimisation and model reduction techniques, such as kernel principal component analysis (PSO-PCA), has helped in tackling large number of uncertainty parameters and parameterising spatially correlated random fields. Secondly, we show that by using Principal Component Analysis, Particle Swarm Optimisation is able to obtain a diverse set of good fitting models comparable to the EnKF ones.

In the next subsections we present the Brugge model that has been built as part of this study in Eclipse to carry out the work from the data provided by TNO. The application

and the comparison results of both ensemble Kalman filter technique (discussed in Section 3.4.3) and particle swarm optimisation then follow.

9.1 Brugge Reservoir

The Brugge reservoir is a unique SPE benchmark synthetic reservoir designed by the TNO from an existing waterflooded oil field (dead oil) to test a closed loop waterflooding optimisation and history matching methods. Nine groups took part and showed their results during the SPE Applied Technology Workshop (ATW) (Closed-Loop Reservoir Management Workshop) held in Bruges, Belgium, in June 2008.

The sandstone reservoir consists of an east-west elongated half-dome formation enclosed within a fault-bounded structure at its northern edge. Figure 9.1 shows the Brugge reservoir top structure map. The lateral flow is impacted with the only internal fault with a modest throw at an approximate angle of 20° to the Northern Boundary Fault (NBF). The reservoir model is divided into 9 geological layers where its properties and thickness are typical for a North Sea Brent-type reservoir. The estimated STOIP is about 755 MMSTB. There is no pressure support from the inactive aquifer on the edge of the reservoir. The field is depleting with voidage replacement. The Brugge field has in total 30 smart wells with three perforation intervals per well and vertical flow control. Smart wells are wells equipped with smart completion that can mitigate unexpected water production due to fractures and therefore enhance the ultimate recovery accomplished by selectively controlling production from multiple laterals. Three downhole inflow control valves (ICVs) were installed in all producers and injectors to enable controlling separately the rates for the Schelde, Waal, and Maas formations. There are 20 vertical oil producers and 10 vertical peripheral water injectors as shown in Figure 9.1.

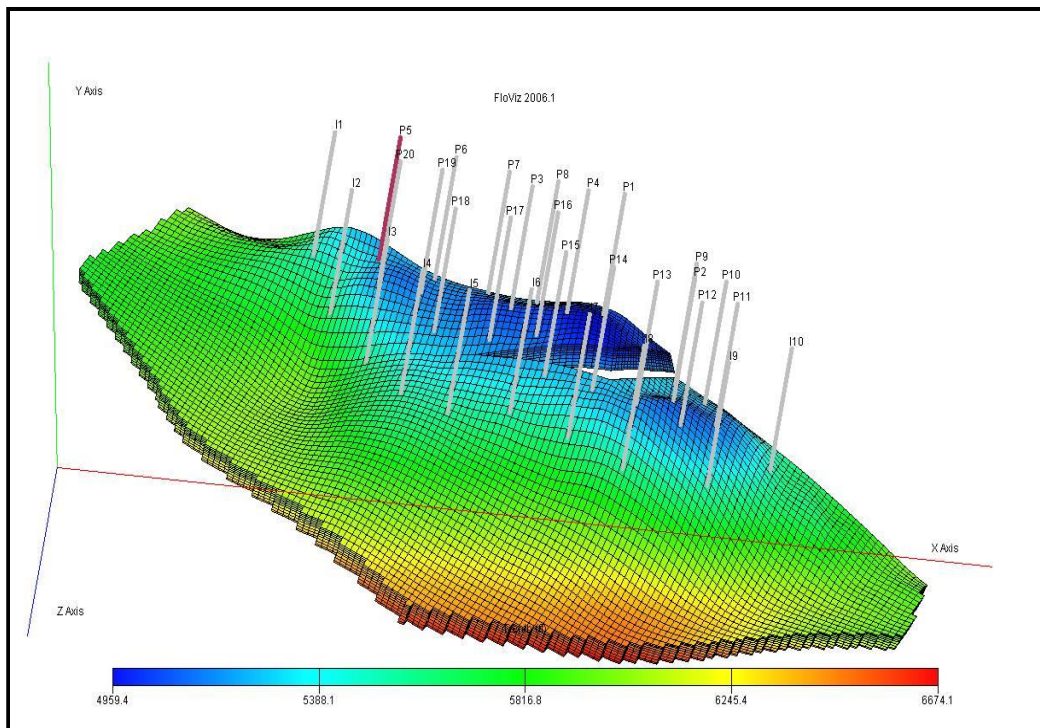
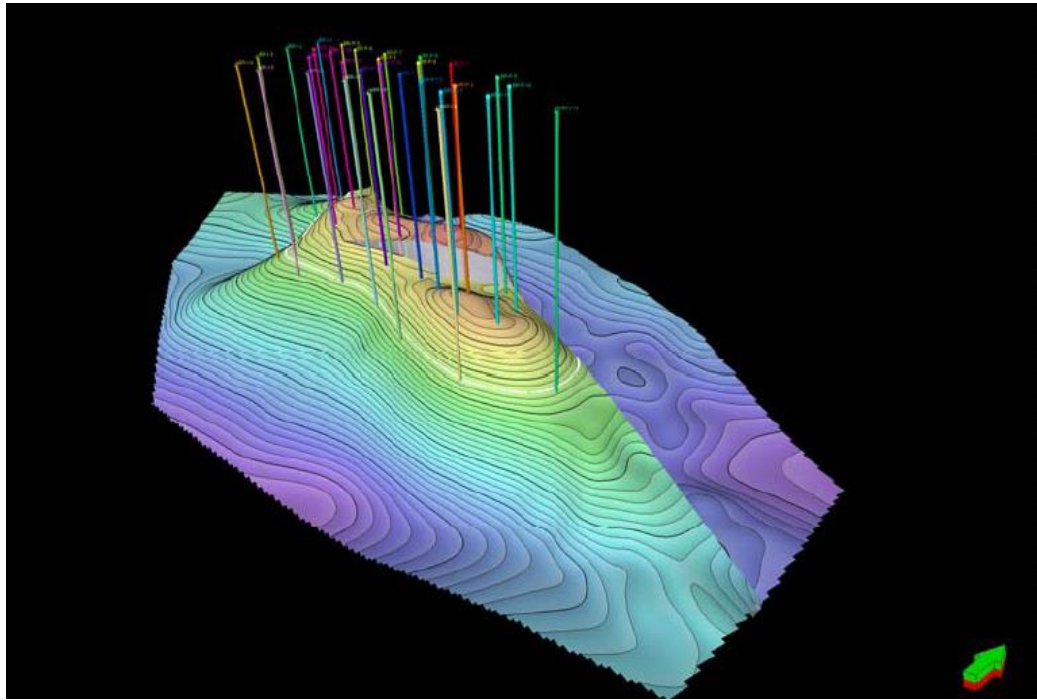


Figure 9.1: 3D Top structure map, OWC and fault (Source: Peters et al., 2010)

The 10-year reservoir history production consists of well oil and water rates, in addition to well pressures for producers and injectors. The wells' historical data includes measurement errors created by adding noise. The organisers prepared a high-resolution reservoir model containing 20 million grid cells which has been upscaled to 450,000 grid cells later exploited to simulate the truth case. The "truth" reservoir model has a size of $75 \times 75 \times 2.5$ m and a total of 327,067 active gridblocks. It was modelled in the

proprietary Shell’s new Modular Reservoir Simulator (MoReS) (Peters et al., 2010). A total of 104 upscaled geological realisations of a 3D geological model have been created representing the prior knowledge about the field containing 60,048 grid blocks. Brugge upscaled model dimensions are 139×48×9. The data provided for the participants included wells perforations, well logs, relative perms, free water level, PVT, economic parameters for oil and water, and discount rate. In addition, an inverted time-lapse seismic survey in terms of (uncertain) pore pressures and saturations on the coarse scale is provided twice: initially and after 10 years (Geel, 2008; Peters et al., 2010). The original objective of the study is to history match using any technique and to come up with an optimal production control strategy for the next 20 years to estimate the cumulative oil production and optimise the net-present-value (NPV) with no further wells drilled. The Brugge field includes dead oil (no gas) and water characterised in Table 9.1. The field is a considerably oil-wet field.

Table 9.1: Fluid properties in the Brugge Field (Source: Geel, 2008).

Fluid	Surface density (lb/ft ³)	Viscosity (cp)	Compressibility (psi ⁻¹)
Oil	56	1.29	9.26.10 ⁻⁶
Water	62.6	0.32	3.10 ⁻⁶

In this thesis we focus on history matching of the reservoir model only. The description of the realisations as noted by Geel (2008) is provided below.

9.1.1 Stratigraphy of the Brugge Reservoir

The properties and thicknesses of the reservoir zones are typical for a North Sea Brent-type field, i.e. a delta plain/barrier type of reservoir, however, with the vertical sequence of the formations altered with respect to the general Brent stratigraphy (consisting of the Broom–Rannoch–Etive–Ness–Tarbert formations) to improve the exercise’s attractiveness. Thus, the highly permeable reservoir zone changed places with the underlying less-permeable, heterogeneous zone. Brent is an acronym for the members of the Jurassic Brent formation that make up the field: Broom, Rannoch, Etive, Ness and Tarbert (in turn named after features in the Scottish Highlands). The Brugge reservoir consists of a stratigraphy as presented in Table 9.2 (Geel, 2008).

Table 9.2: Stratigraphy used in the Brugge Field with the main characteristics (Source: Geel, 2008)

Formation (Reservoir Zone)	Average Thickness (m)	Average Porosity (%)	Average Permeability (mD)	Average N/G (%)	Depositional Environment	Remarks
Schelde Fm	10	20.7	1105	60	Fluvial	Discrete sand bodies in shale
Waal Fm	26	19.0	90	88	Lower Shoreface	Contains loggers: carbonate concretions
Maas Fm	20	24.1	814	97	Upper Shoreface	–
Schie Fm	5	19.4	36	77	Sandy Shelf	Irregular carbonate patches

9.1.2 Realisations of Reservoir Properties

A total of 104 upscaled (60,048 grid cells with 44,550 active cells) realisations of the field were created to reflect the prior distribution. The selected control parameters fell into the following four classes:

1. Facies: subdivide the reservoir into facies classes with associated poroperm characteristics (FY), or alternatively, ignore facies (FN)
2. Fluvial: if facies modelling is enabled (FY), then the fluvial reservoir zone can be modelled either as channel objects in a shale background (SF), or it can be modelled as a Sequential Indicator Simulation (SS)
3. Porosity: porosity is generated stochastically using Sequential Gaussian Simulation (PS)
4. Permeability: permeability can be generated deterministically, as a single poroperm regression (KS), or as a poroperm regression per facie (KM) or it can be generated stochastically, with co-Kriging on porosity (KP)

Porosity is generated by PS where sequential simulation is used to maintain the spatial structure (honour the histogram and variogram inferred from the known data). Kriging condition models to static hard data at the wells (see Section 2.4).

A total of 104 realisations were constructed by combining the different options. Each modelling technique has 13 realisations. Each of the realisations has the following properties: Facies, Porosity, NTG, Water Saturation, and Permeability in X, Y, and Z directions. There are 7 regions distinguished in the reservoir based on the porosity distribution with each one having its own saturation table. Different rock types exist with 7 different relative permeability curves, but only one relative permeability curve is

used in this study. It is important however to further integrate the capillary pressure data for modelling initial water saturation and pressure with the 7 equilibration regions.

9.2 Parameterisation

Compartmentalisation (zonation/regionalisation) parameterisation approach is a common method usually used in evolutionary and swarm intelligence techniques to adjust the uncertainty parameters of the reservoir to reduce the number of parameters. This form of parameterisation has been the standard in the Petroleum Industry (Floris et al., 2001). Schulze-Riegert et al. (2009) have previously tested coupling an Evolutionary Strategy (ES) with the ensemble Kalman filter (EnKF) in order to benefit from and overcome the weaknesses of both methods using Brugge as a test case. However, in real field applications geologists can create more than one scenario to account for the underestimated uncertainty introduced by using only one realisation. The Brugge case could be considered as a good candidate to test the applicability in using stochastic techniques as a stand-alone optimiser to generate multiple history matched models that honour the geology and the spatial correlation features. The available data provided by TNO does not incorporate any defined regions, but 104 realisations that describe the spatial correlation of the properties. Recent parameterisation techniques, which integrate geostatistical information and tries to preserve geological consistency and continuity of an ensemble of models created, include, to name a few, Principal Component Analysis (Sarma et al., 2008a, 2008b), Gradual Deformation (Hu, 2000), Discrete Cosine Transform (Jafarpour and McLaughlin, 2009), Kernel Ridge Regression (Sætrum and Omre, 2010), Discrete Wavelet Transform (Kind and Quinteros, 2007), and Multiple Kernel Learning (Demyanov, Foresti, Christie, and Kanevski, 2011; Demyanov, Foresti, Kanevski, and Christie, 2010) (refer to Section 2.5 for further details). In our application here we have picked the Principal Component Analysis strategy to predict the reservoir properties of this study. This work is presented in Mohamed et al. (2010a). Very recent work by Fernández-Martínez et al. (2010) adopted in a similar way our PSO-PCA approach on a synthetic history matching problem, extracted from the Stanford VI (Castro et al., 2005) sand and shale synthetic reservoir. Below is a brief theoretical demonstration upon which this work draws.

9.2.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. Since patterns in data can be hard to find in higher dimensions, where the luxury of graphical representation is not available, PCA is a powerful tool for analysing data (Hastie et al., 2009). Once these patterns in the data have been found, the data can be compressed by reducing the number of dimensions without much loss of information. This technique is popular and has been extensively used in the literature in several areas such as reservoir engineering, fluid dynamics, turbulence, signal processing, operational oceanography, image compression, , dimensionality reduction, feature extraction, data visualisations, weather prediction, statistics, and in many machine learning applications (Bishop, 2006; Jolliffe, 2002). Figure 9.2 shows the principal components for dataset represented in 2D. The largest principal component is the direction that maximises the variance of the projected data and the smallest principal component is the one which minimises that variance (Hotelling, 1933). PCA is also defined as the linear projection that minimises the mean squared distance between the data points and their projections (Pearson, 1901). The method is also known as Orthogonal Empirical Bases, Karhunen-Loève Transform and Proper Orthogonal Decomposition (Bishop, 2006).

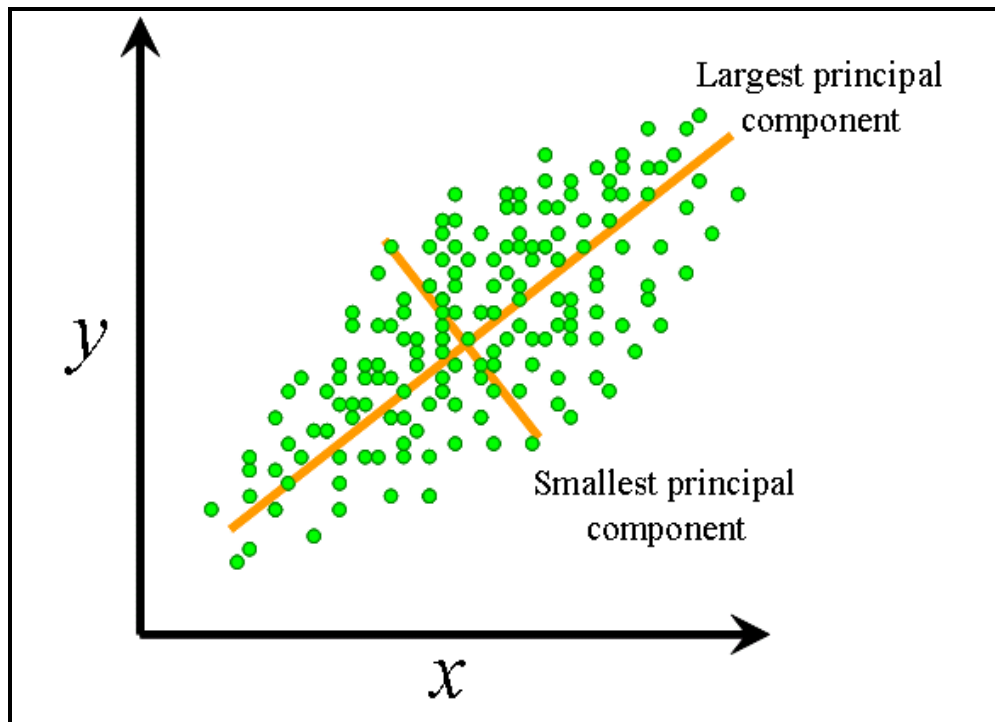


Figure 9.2: Principal Components in 2D

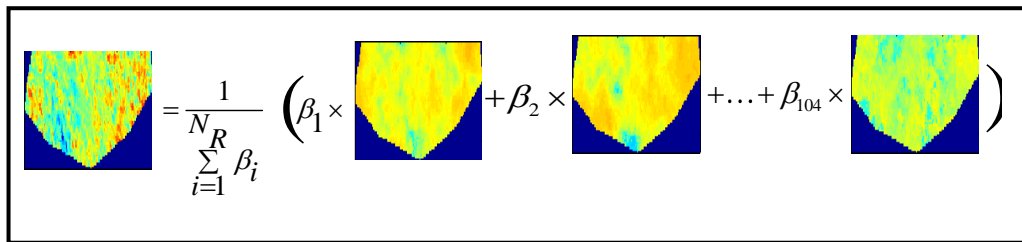
The common steps for applying the PCA:

1. Obtain the geological data.
2. Calculate the covariance matrix C .
3. Calculate the eigenvectors (E) and eigenvalues (λ) of the covariance matrix C .
4. Choose principal components (PCs) and form a feature vector Eq. (9.1) – as illustrated in Figure 9.3.

$$y = \frac{\sum_{i=1}^{N_R} \beta_i y_i}{\sum_{i=1}^{N_R} \beta_i} \quad \text{Eq. (9.1)}$$

5. Derive the new data set

We adopted the methodology used by Sarma et al. (2008b) which uses the standard Karhunen-Loeve (K–L) expansion or Linear PCA. Linear PCA only preserves the two-point statistics of the random field.



$$\text{Image} = \frac{1}{\sum_{i=1}^{N_R} \beta_i} \left(\beta_1 \times \text{Image}_1 + \beta_2 \times \text{Image}_2 + \dots + \beta_{104} \times \text{Image}_{104} \right)$$

Figure 9.3: Simple example of how to form feature vector from 104 prior images with Linear PCA by applying a polynomial “kernel” of order 1 (see Section 9.2.1.1)

9.2.1.1 Linear PCA Application

PCA application involves searching an orthogonal base of the experimental covariance matrix estimated with the prior 104 geological models, and then selecting a subset of the most important values with their associated eigenvectors that are used as a reduced model space base.

The K–L expansion is a linear functional relationship as $\beta = f(\xi)$. The random field y is parameterised in terms of N_m independent random variables ξ where N_m is the largest N_c total eigenvalues maintained. The largest N_m eigenvalues capture the general field features whilst the $(N_c - N_m)$ smallest eigenvalues represent the detailed geological features.

For getting a new realisation, the diagonalisation of the covariance matrix C needs to be worked out by solving the equation:

$$\lambda v = C v \quad \text{Eq. (9.2)}$$

Solving this with singular value decomposition (SVD) is very expensive, with a computational complexity of $O(N_C^3)$, where N_C is the size of the vector y . The exact equivalent alternative formulation of the same problem demonstrated in the previous equations is called the kernel eigenvalue problem in which a “kernel” is defined. Hence, the problem can now be solved much more efficiently to determine the non-zero eigenvalues λ . The kernel matrix K where $k_{ij} = (y_i, y_j)$ is the dot product of realisations i and j is of size $N_R \times N_R$ matrix and is called polynomial kernel of order 1. It can be shown that the eigenvalues and eigenvectors of the eigenvalue problem are equivalent of the following kernel eigenvalue problem in Eq. (9.3) in which γ is a coefficient vector.

$$N_R \lambda \gamma = K \gamma \quad \text{Eq. (9.3)}$$

The solution of Eq. (9.3) is equivalent to solving Eq. (9.2) since the non-zero eigenvalues of Eq. (9.2) are of that of Eq. (9.3) scaled by N_R . Thus, the eigenvalue problem of this is given by $N_R \lambda$ and the eigenvectors are given by γ .

For the application in this work, the following procedure has been followed:

1. We collect the uncertain variables of interest into a y_i vector composed of Porosity, NTG, and Permeability in X , Y , and Z directions at each grid block.

$$y_i = [\phi^T, NTG^T, \ln k_x^T, \ln k_y^T, \ln k_z^T] \quad \text{Eq. (9.4)}$$

where $k_x = k_y$, $i \in \{1, 2, \dots, 104\}$. The components of y_i are all vectors containing the static variables at every grid block. For the application of Linear PCA, an ensemble of vectors is collected in a matrix Y

$Y = [y_1; y_2; \dots; y_{N_R}]$; $N_R = 104$ where Y is $N_R \times 5N_C$ matrix. $N_C = 60048$ is the model size.

2. Calculate kernel matrix as in Eq.(9.5) where K is $N_R \times N_R$ matrix and k_{ij} is defined as in Eq. (9.7):

$$K = [k_{ij}] \quad i, j = 1, \dots, N_R \quad \text{Eq. (9.5)}$$

3. Construct the eigenvectors matrix E ($N_R \times N_R$) of the matrix K and Λ ($N_R \times N_R$) the diagonal matrix of the eigenvalues of K . The PCA can generate realisations

$$\text{as } y = \sum_i \beta_i y_i / \sum_i \beta_i \text{ where}$$

$$\beta = E\Lambda^{1/2}\xi / \sqrt{N_R} \quad \text{Eq. (9.6)}$$

9.2.1.2 Kernel PCA Application for Preserving Multipoint Statistics

The K–L expansion can be employed for geostatistical simulation to produce a realisation having the same covariance as the random field acquired with the training image. However, the outcome realisations do not reproduce complex structures like channelised models regardless of the amount of energy maintained. That is because the K–L expansion is a linear combination of Gaussian random parameters maintaining the two-point statistics of the original realisations and two-point statistics do not define channels. In non-Gaussian realisation $y = (y_1, y_2)$, y_1 and y_2 may be nonlinearly correlated in \mathbb{R}^2 such as values of log permeability in two particular grid blocks as shown in Figure 9.4 (left) in which the linear PCA application will give the right plot.

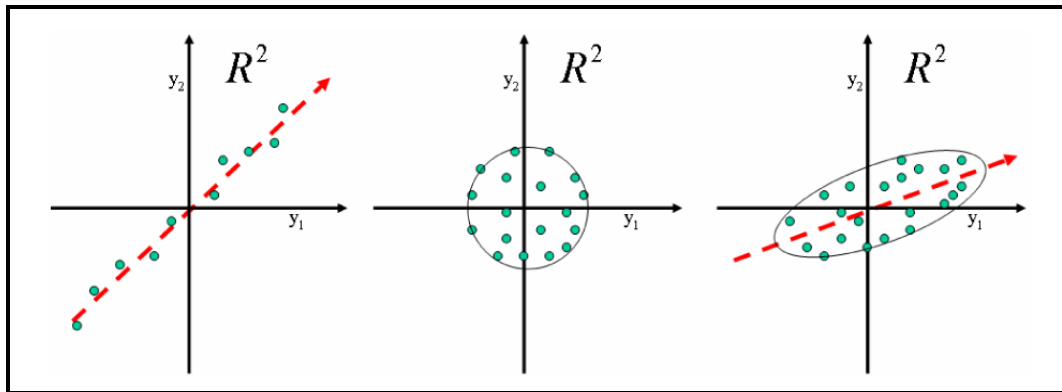


Figure 9.4: y_1 and y_2 correlations for multi-Gaussian y (Source: Sarma et al. (2008b))

By applying a nonlinear map Φ which links the realisation space \mathbb{R}^{N_R} to feature space F that has a large dimensionality as shown in Figure 9.5 we can capture these nonlinearities.

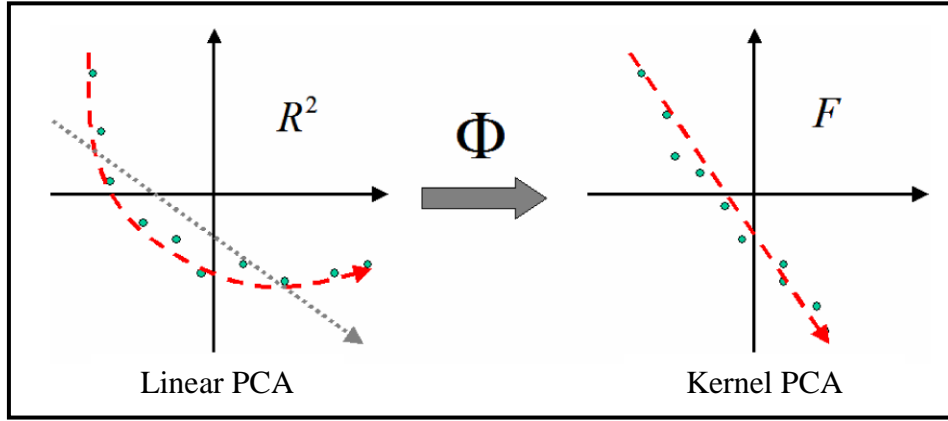


Figure 9.5: Kernel trick idea in kernel PCA approach (Source: Sarma et al. (2008b))

The map is the kernel function that after the transform can make the realisations that were nonlinearly uncorrelated in realisation space be linearly correlated in feature space where linear PCA could be applied. The application of linear PCA in feature space regarding to the polynomial kernel of $(x \cdot y)^d$ maintains the $2d^{\text{th}}$ order moment or $2d$ -point statistics of parameter space. The kernel in Eq. (9.7) maintains up to the $2d^{\text{th}}$ order moment.

$$(\Phi(x) \cdot \Phi(y)) = k(x, y) = \sum_{i=1}^d (x \cdot y)^i \quad \text{Eq. (9.7)}$$

9.2.1.3 The Pre-Image problem

Now we assume that we have a virtual realisation $Y \in F$ and we want to come back into the parameter space in order to have the real new realisation $y \in \mathbb{R}^{N_c}$ such that $y = \Phi^{-1}(Y)$. This problem is called the “Pre-Image problem”. However, it is noted that such y may not exist or indeed many of them may exist because the problem is ill-posed and thus the pre-image problem is an optimisation problem that need to be solved. The following optimisation problem need to be solved so as to find this y

$$\min_y \rho(y) = \|\Phi(y) - Y\|^2 = \langle \Phi(y) \cdot \Phi(y) \rangle - 2\langle Y \cdot \Phi(y) \rangle + \langle Y \cdot Y \rangle \quad \text{Eq. (9.8)}$$

The fixed-point approach in Eq. (9.9) will be used to solve this problem

$$y^{k+1} = \frac{\sum_{i=1}^{N_R} \beta_i \cdot k'(y_i, y^k) y_i}{\sum_{i=1}^{N_R} \beta_i \cdot k'(y_i, y^k)} \sum_{j=1}^{N_R} \gamma_{i,j} \delta_j^{1/2} \xi_j \quad \text{with} \quad \beta_i = \frac{1}{\sqrt{N_R}} \quad \text{Eq. (9.9)}$$

where δ_i is the i^{th} eigenvalue of the matrix K and γ_i is the i^{th} eigenvector of the matrix K , for $j = (1, \dots, N_R)$. A recap of the kernel approach is given in Figure 9.6. Model reduction techniques are described in detail in Bishop (2006), Cheng (2010), Hastie et al. (2009), Sarma et al. (2008b), Soismier (2009), and machine learning literature.

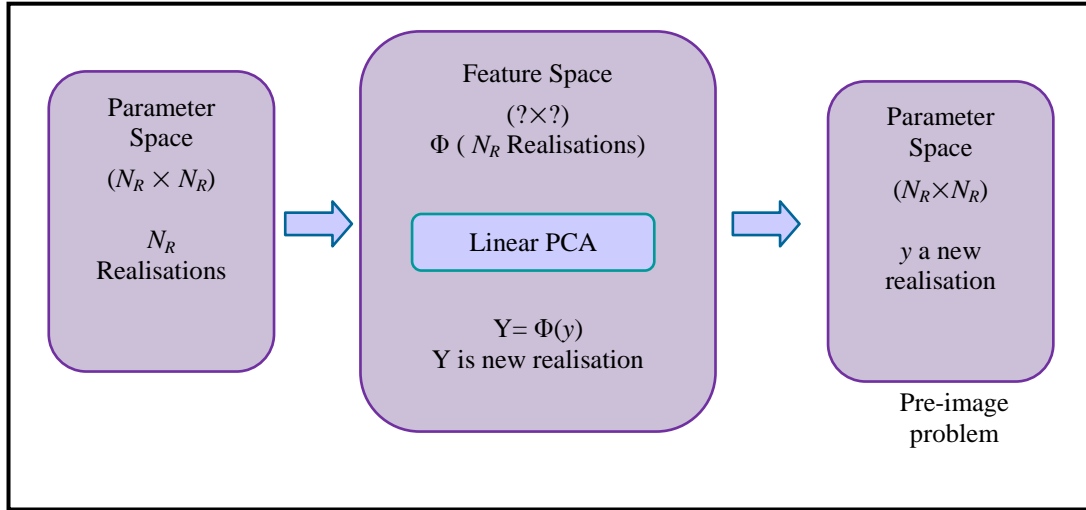


Figure 9.6: Kernel PCA principle (linear PCA in feature space)

9.3 History Matching Application – Brugge Reservoir

So far, Chapter 2 has presented the Ensemble Kalman Filter (EnKF) data assimilation method while Chapter 4 has reviewed the Particle Swarm Optimisation (PSO). This section will present the objective functions definitions and then the comparison of PSO and EnKF follows.

9.3.1 Brugge History Matching Uncertain Parameters

The PSO objective is to find the best particle represented by the set of the 104 parameters ordered from the largest eigenvalue to the lowest eigenvalue represented by $(\xi_1, \xi_2, \dots, \xi_{104})$ which leads to the minimum misfit value of the corresponding realisation. No explicit physical meaning is linked to these parameters. The prior ranges for the parameters are uniform priors sampled from the ranges obtained by the minimum and maximum values for each parameter computed from Eq. (9.10).

$$\xi = \beta \left(E\Lambda^{1/2} / \sqrt{N_R} \right)^{-1} \quad \text{Eq. (9.10)}$$

By choosing $\beta = I$ (the identity matrix), the prior realisations can be obtained where Eq. (9.11) are the parameter values that can be chosen.

$$\xi_0 = \left(E\Lambda^{1/2} / \sqrt{N_R} \right)^{-1} \quad \text{Eq. (9.11)}$$

Figure 9.7 summarises the history matching workflow developed for the Brugge field with linear PCA equations for demonstration. A similar procedure for kernel case is applied.

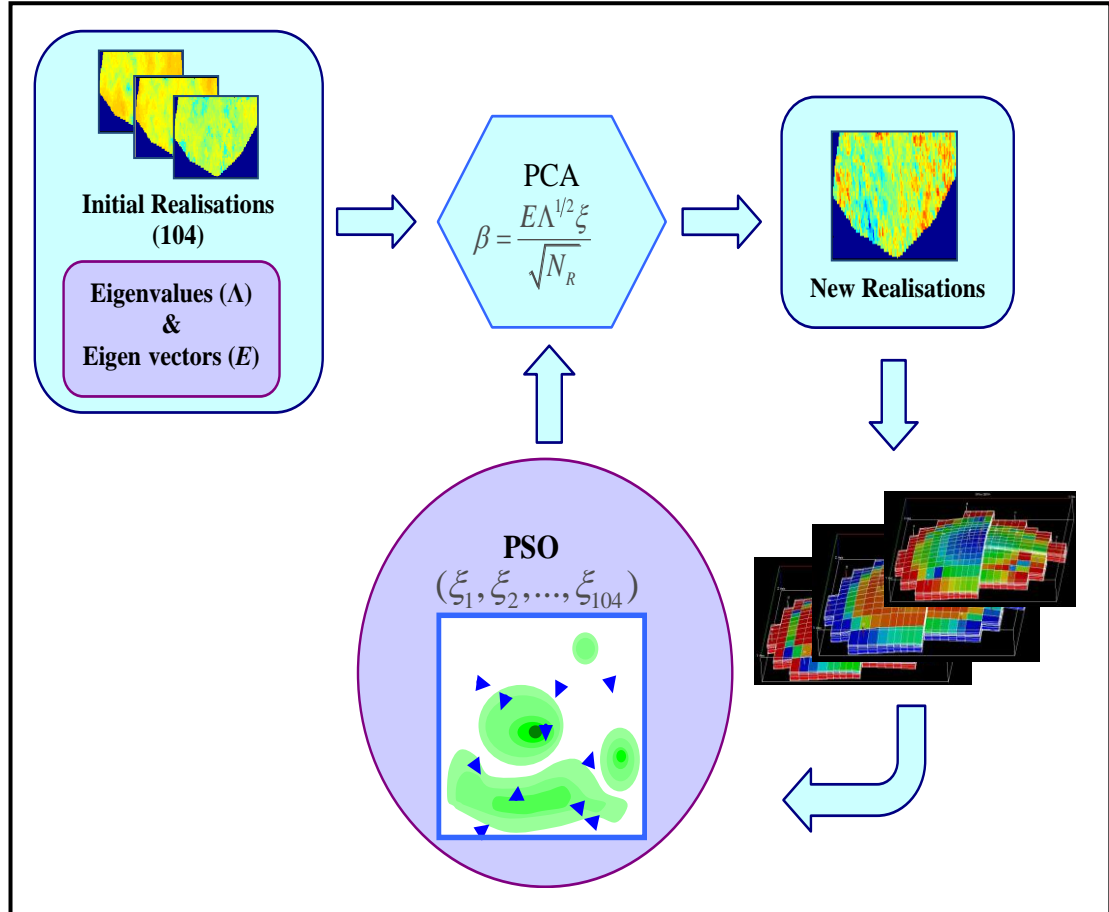


Figure 9.7: The history matching workflow for the Brugge reservoir

9.3.2 Objective Function Definition

The objective function for PSO is defined as in Eq. (9.12).

$$Misfit = \sum_{t=1}^T \left(\frac{(q_w^{obs} - q_w^{sim})_t^2}{2\sigma_{qwp}^2} + \frac{(P^{obs} - P^{sim})_t^2}{2\sigma_{WBHP}^2} \right) \quad \text{Eq. (9.12)}$$

where T is the number of observations, q is the water rate, P is the well pressure for observed (denoted as *obs*) and simulated data (denoted as *sim*) respectively, and σ^2 is the variance of the observed data.

This definition is based on the assumption that the measurement errors are Gaussian and independent. White noise was added to the observations to make the data more representative. A standard deviation of $\sigma_{qwp} = \sigma_{qop} = 30$ STB/D for oil and water

production rates measurement errors and $\sigma_{WPHP} = 7.35$ PSI for pressures were used (Peters et al., 2010). The injection rate is constant at 4000 STB/D. The producers are shut if the water cut exceeds 0.9. The simulator is controlled to match well oil rates.

9.3.3 Results

In this section we report the results of obtaining multiple history matched models with the particle swarm optimisation using the linear PCA parameterisation approach. History matching results are compared with EnKF. The result shown here are the best results among the couple of initial runs with different settings for both the EnKF and PSO.

In PSO, we started from an initial population comprising 100 models generated randomly in parameter space. The optimisation is done for 30 iterations, each generation comprising 100 models. The total number of reservoir model simulations is 3000 for the performed tests. A fair match obtained around 1000 forward simulations with 10 generations as shown in Figure 9.8. Runtime with the Brugge field has taken around 11.5 hours (roughly 13.8 minutes per simulation) with 25 nodes (50 CPUs) in the in-house Heriot-Watt University cluster. Figure 9.8 monitors the global best misfit versus the generation number where the misfit is reducing over time. After 1000 simulations (generation 10) the misfit is not reducing much, so we may stop the run at that point. Sampling history evolution is shown in Figure 9.9 and Figure 9.10 in which we plot 104 parameters ordered according to the corresponding 104 coefficients associated with the highest eigenvalues. The algorithm is sampling different parts of the parameter space as we see in some of the parameters such as P6, P7, and P30.

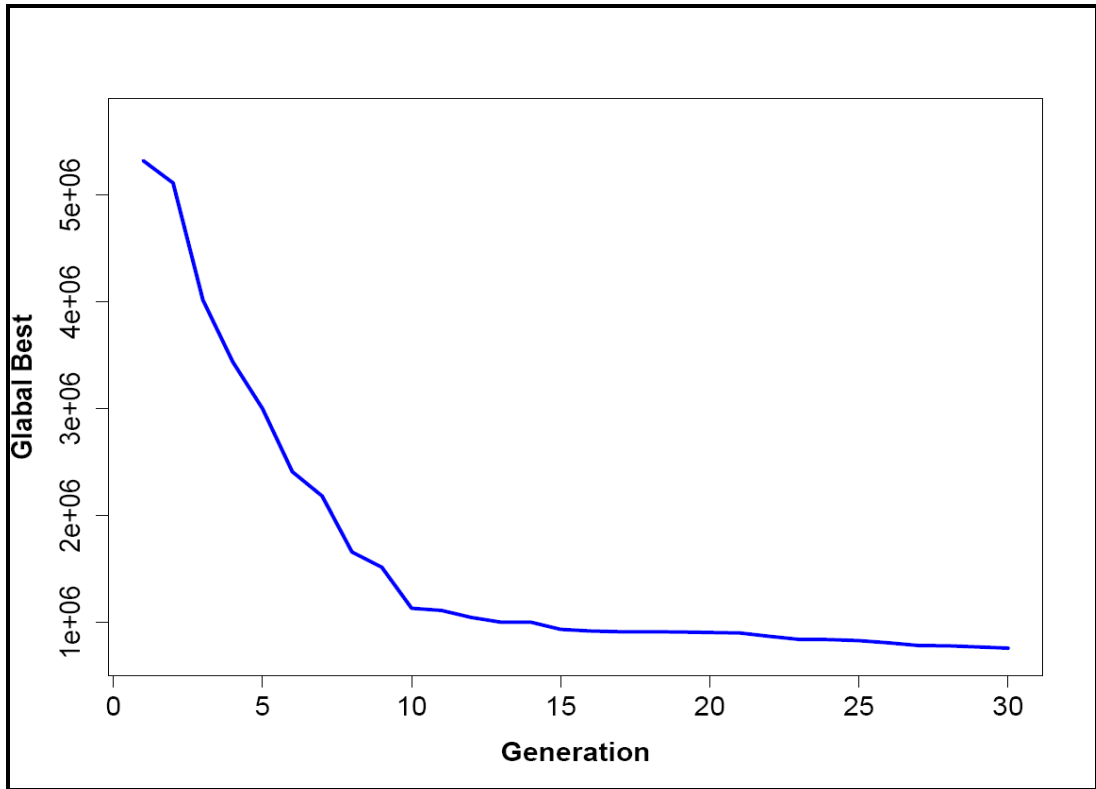


Figure 9.8: Misfit reduction for PSO

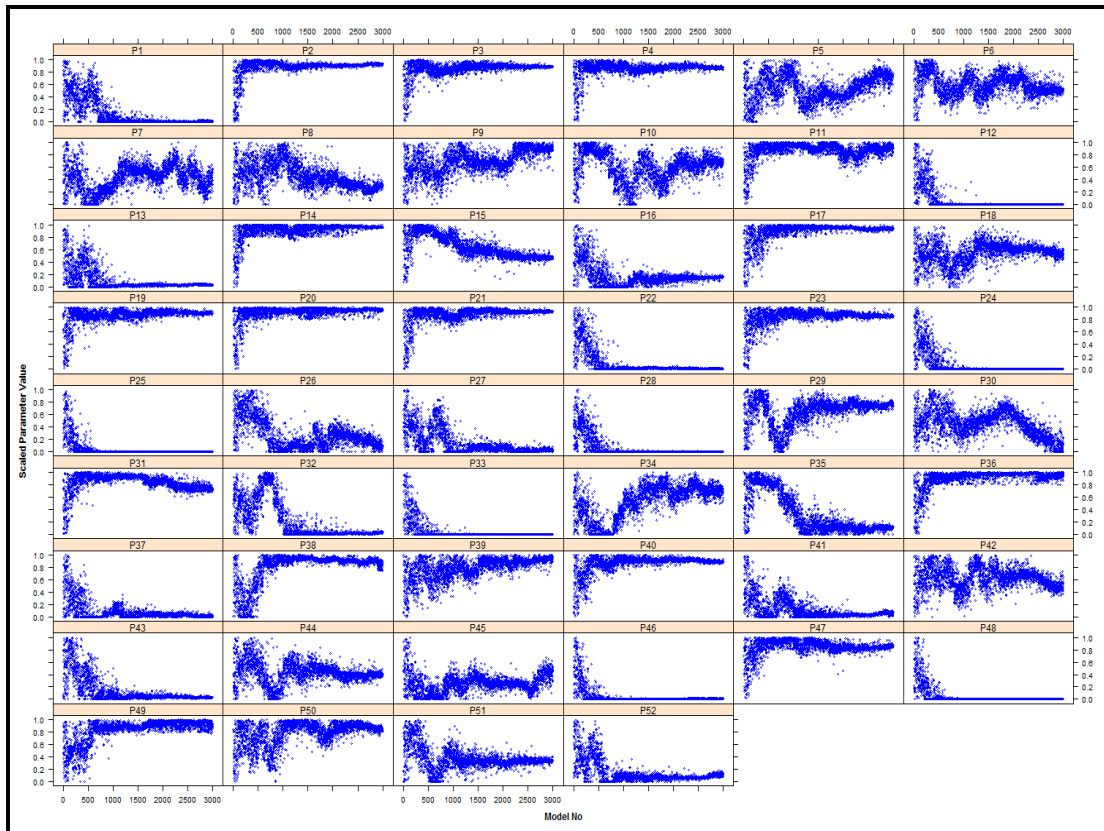


Figure 9.9: Sampling evolution for the first 52 parameters

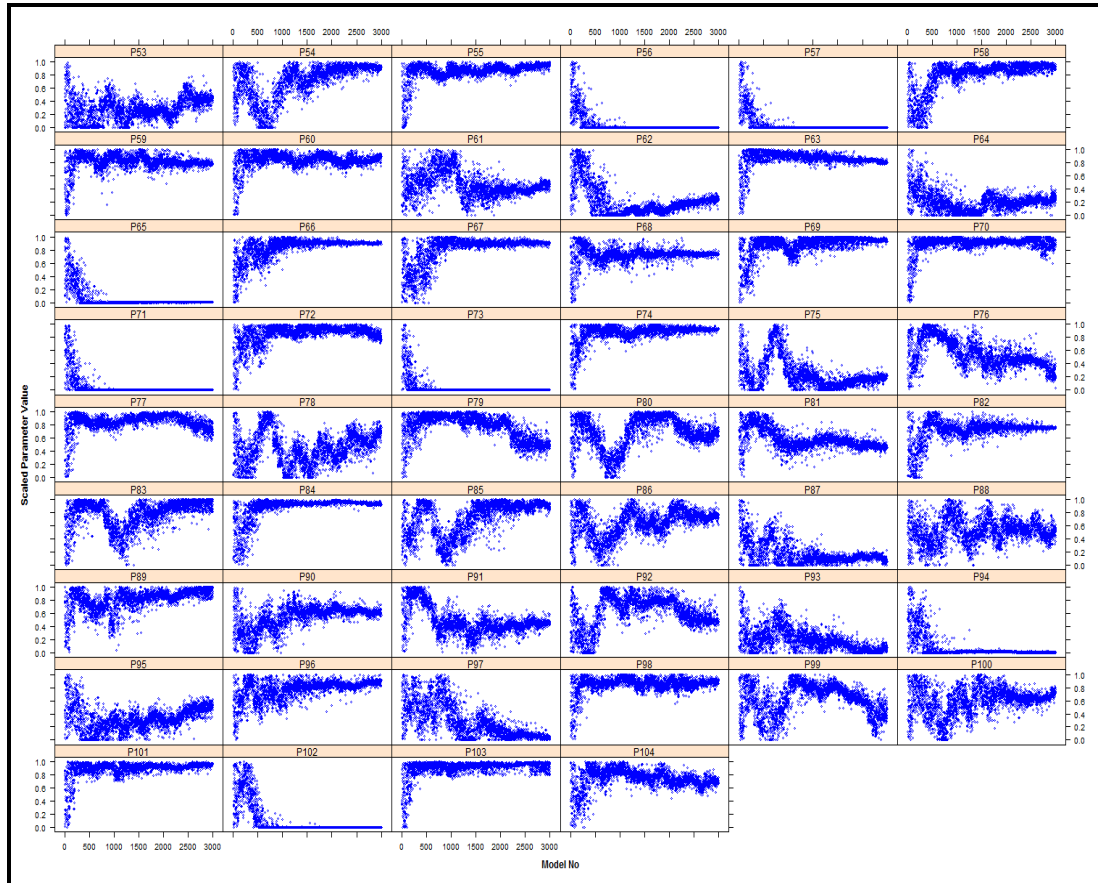


Figure 9.10: Sampling evolution for the 53-104th parameters

9.3.4 Comparison with EnKF

The work carried out in this comparison used an in-house Total E&P UK Standard Ensemble Kalman Filter MATLAB code. It is noted that the application of EnKF here does not use localisation of covariance, a technique introduced to EnKF to tackle the limitation of the EnKF standard approach in having sparse values such as high permeability. Perturbation of these unknowns in locations far away from the wells would have different impact on the results of the technique. Primarily the intention has been to test the standard stand-alone versions. Nevertheless, the technique has been investigated recently with localisation step in Chen et al. (2010) and it is reported that it mitigates the sampling errors for data assimilation in EnKF and this improved the results over the canonical EnKF for the same Brugge study. Thus, it will be interesting research to compare the methods with the recommended localisation step.

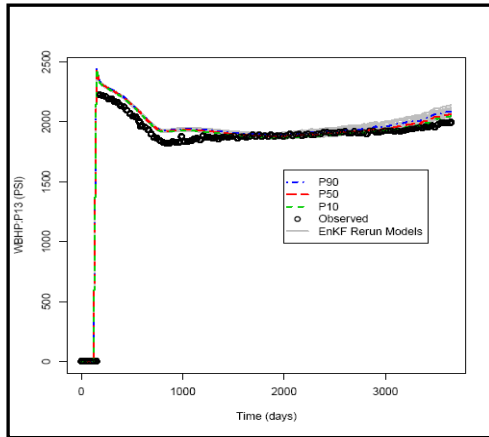
The EnKF setup used for this preliminary comparison involves a different misfit definition by allowing higher noise level values per well for every timestep while it is kept constant in PSO. With the same misfit definition the EnKF has not obtained a better history matching results than the ones reported here. Therefore, this was not achieved and misfit definitions choices which suit each technique best were adopted to make the comparison. Furthermore, the oil rate error term from Eq. (9.12) has been included in the EnKF misfit definition. The measurement errors for oil rate, water rate and pressure used were as follows: 10% of oil production, 15% of water production, and 5% of well pressure respectively. In addition one observation is used per each five observations for a total of 32 timesteps for computational reasons. Since the data points have a specified trend, this should not affect the results as much as the variance used to estimate the measurement errors. The total number of complete simulations in EnKF is equal to the number of realisations which is 104. Since there are differences in the distribution of misfit values for the prior 104 realisations with the EnKF misfit definition in comparison with the distribution of misfits obtained with the definition used for PSO and EnKF, the optimisation procedures will be achieved differently. For comparing the history matching results with PSO, all data points originally provided in the study were restored in the plots for achieving the comparative analysis of both methods' performance.

For the comparative study purpose we used the best 100 fitting models of the PSO to compare with the EnKF models obtained. We assumed that all the best 100 fitting models of the PSO were equally likely then we plotted the P10-P50-P90 uncertainty envelopes. The grey colour shows the models found by each method. Here are examples for a few wells selected as shown in Figures 9.11 to 9.16 for the comparison:

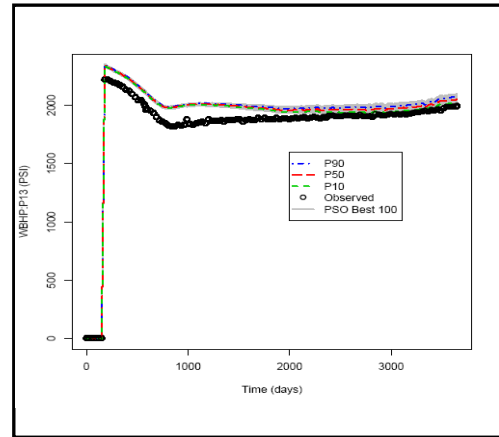
- Well pressure for producer 13 is better matched by EnKF as shown in Figure 9.11(a), compared to PSO as shown in Figure 9.11(b).
- Well pressure for producer 15 is better matched by PSO as shown in Figure 9.12(b), compared to EnKF as shown in Figure 9.12(a).
- Well water production rate for producer 2 is well matched by both as shown in Figure 9.13(a) and (b).
- Well water production rate for producer 5 is not matched by both as shown in Figure 9.14(a) and (b).

- Well water production rate for producer 17 is better matched by PSO as shown in Figure 9.15(b), compared to EnKF as shown in Figure 9.15(a).
- Well water production rate for producer 18 is better matched by EnKF as shown in Figure 9.16(a), compared to PSO as shown in Figure 9.16(b).

The complete history matched performance for both EnKF and PSO are given in Figures A.1 to A.10 in Appendix A. The EnKF obtained better history matches of pressures for producers for some wells while PSO obtained wider ranges and better captured the observed values in water production rates for some wells. Figure 9.17 summarises the overall performance of both techniques. We can see comparable results achieved by both methods with PSO having a slight improvement. Wells which were difficult to match with EnKF have a better match with PSO, however some wells have not obtained a satisfactory match with PSO as indicated earlier.

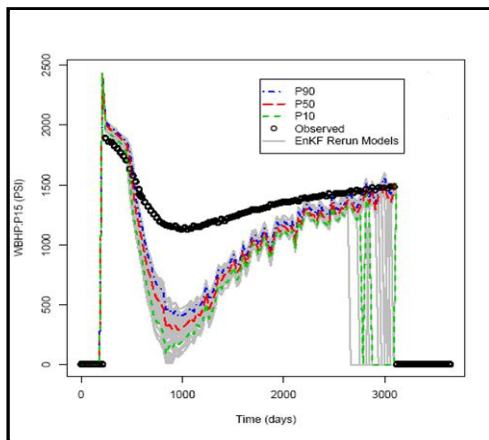


(a) EnKF

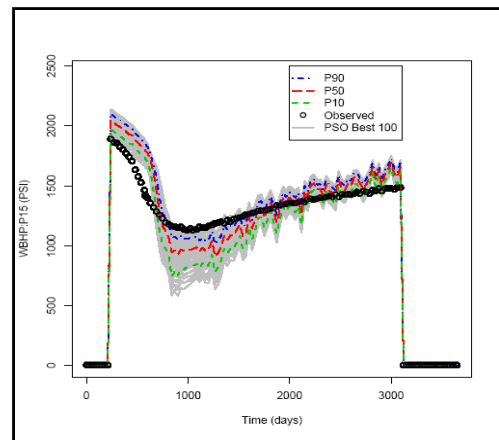


(b) PSO

Figure 9.11: Well pressure history matches for producer 13 for EnKF (a) and PSO (b)

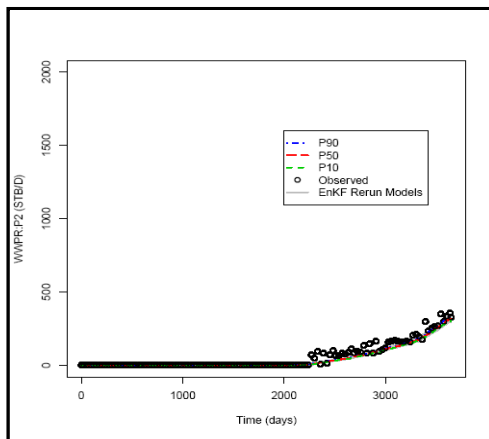


(a) EnKF

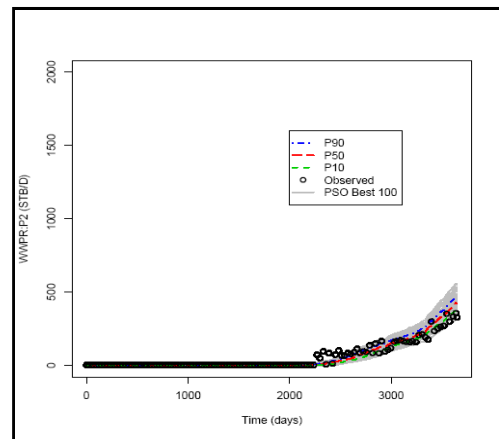


(b) PSO

Figure 9.12: Well pressure history matches for producer 15 for EnKF (a) and PSO (b)

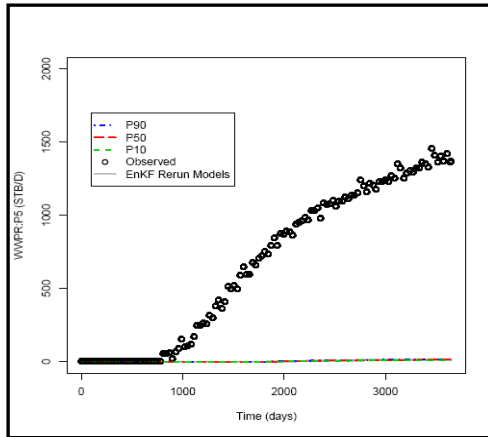


(a) EnKF

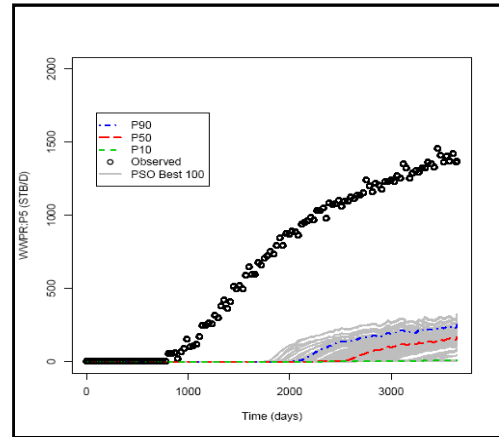


(b) PSO

Figure 9.13: Well water production rate history matches for producer 2 for EnKF (a) and PSO (b)

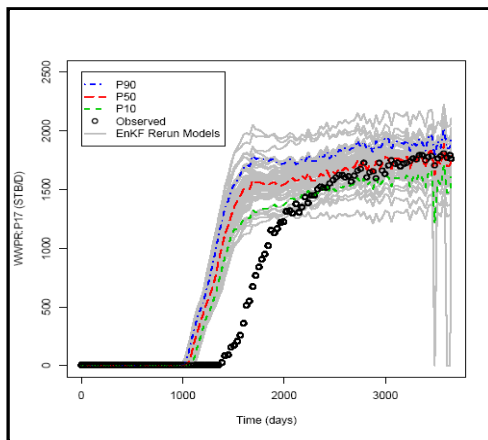


(a) EnKF

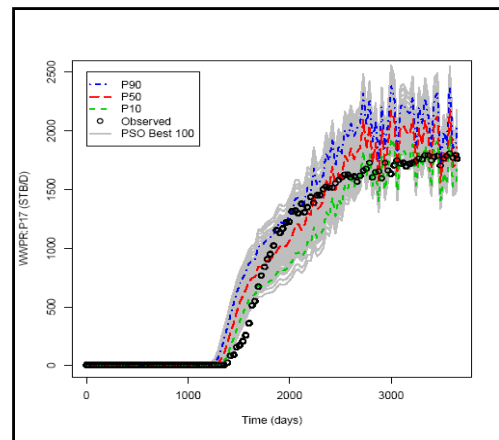


(b) PSO

Figure 9.14: Well water production rate history matches for producer 5 for EnKF (a) and PSO (b)

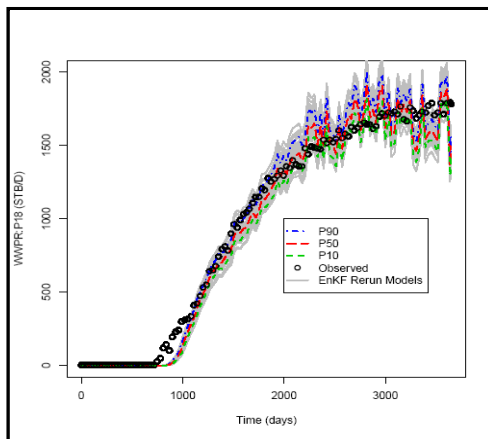


(a) EnKF

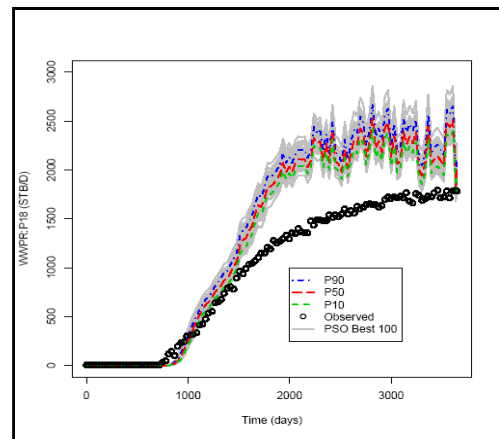


(b) PSO

Figure 9.15: Well water production rate history matches for producer 17 for EnKF (a) and PSO (b)

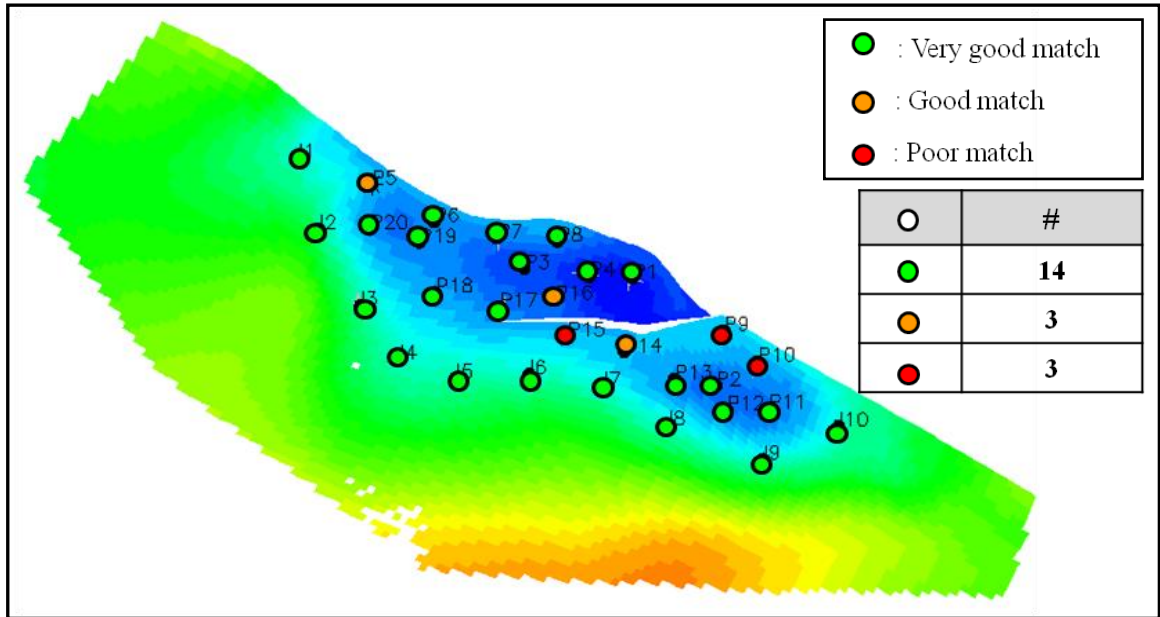


(a) EnKF

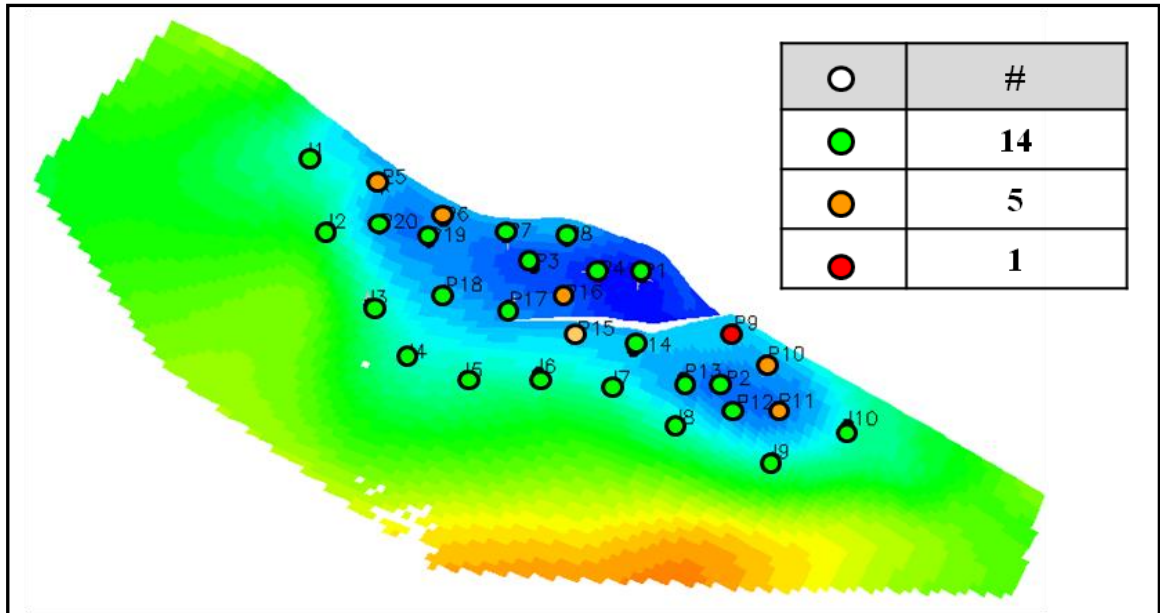


(b) PSO

Figure 9.16: Well water production rate history matches for producer 18 for EnKF (a) and PSO (b)



(a) EnKF

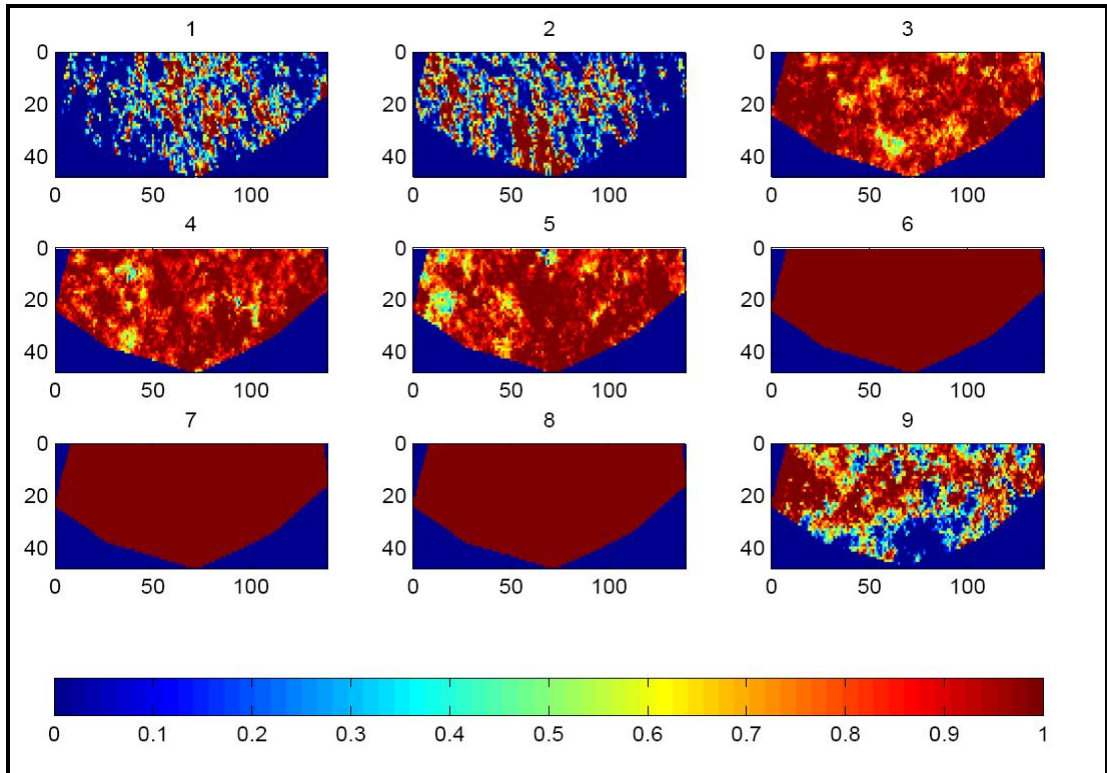


(b) PSO

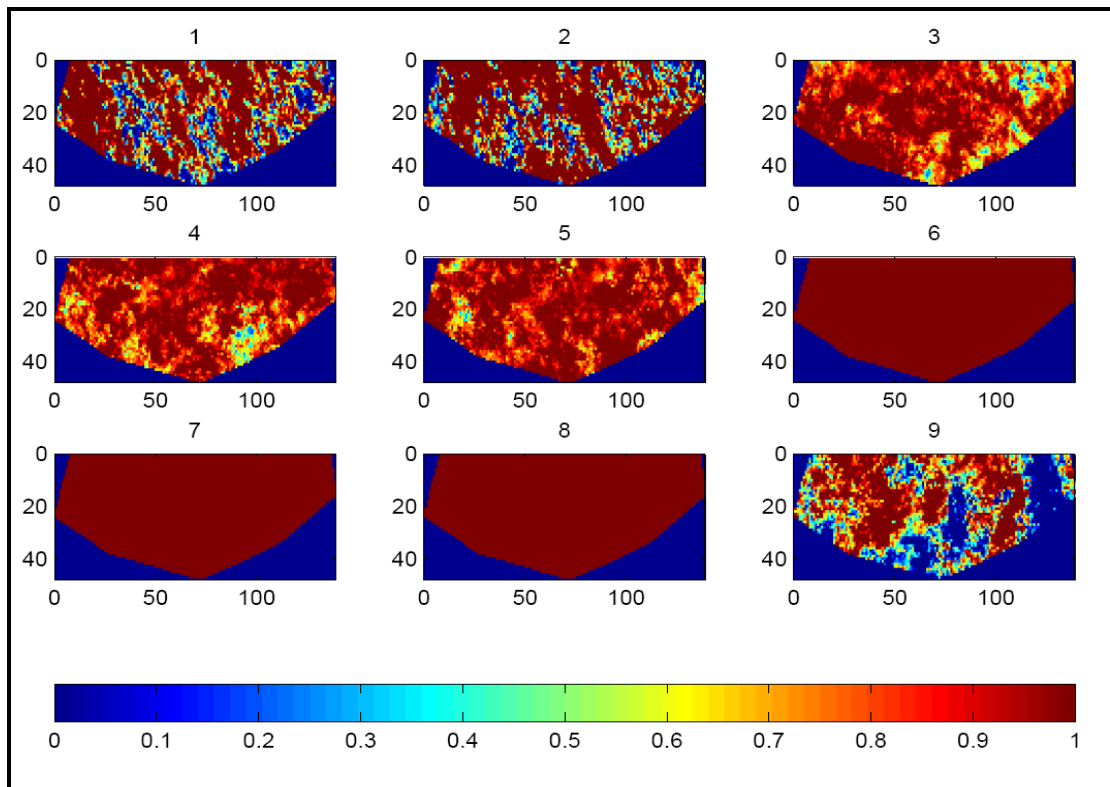
Figure 9.17: Recap of wells history matches for EnKF (a) and PSO (b)

The petrophysical properties for the nine layers: the NTG and permeability in X direction (in mD) for the best fitting model obtained by EnKF and PSO are shown in Figures (a) and (b) of 9.18, and 9.19 respectively. The porosity for the best fitting model obtained by EnKF and PSO is shown in (a) and (b) of Figure B.1 respectively in Appendix B. The permeability in Z direction (in mD) for the best fitting model obtained by EnKF and PSO is shown in (a) and (b) of Figure B.2 respectively in Appendix B. Generally, continuity is observed in PSO best-fitting realisation compared to the EnKF best-fitting one, where heterogeneity is observed. Geological knowledge of

the field may help evaluate and modify the realisations consistently with the observed reality.

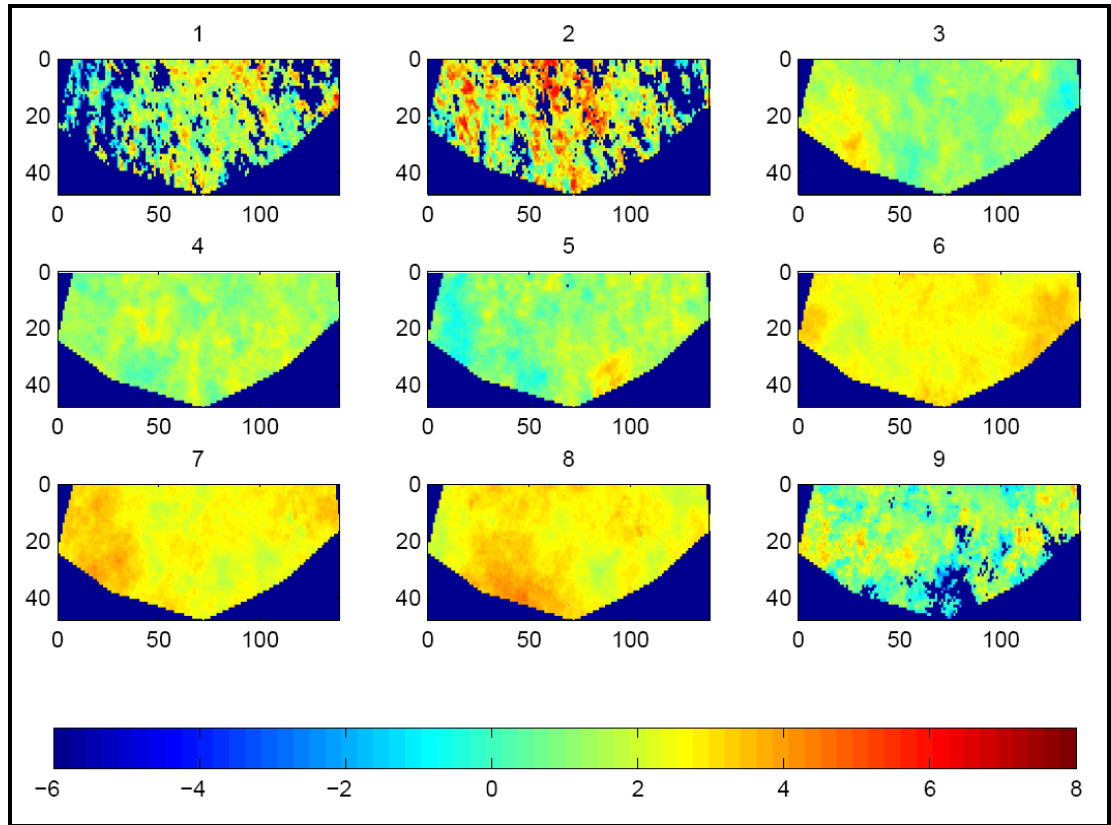


(a) EnKF result

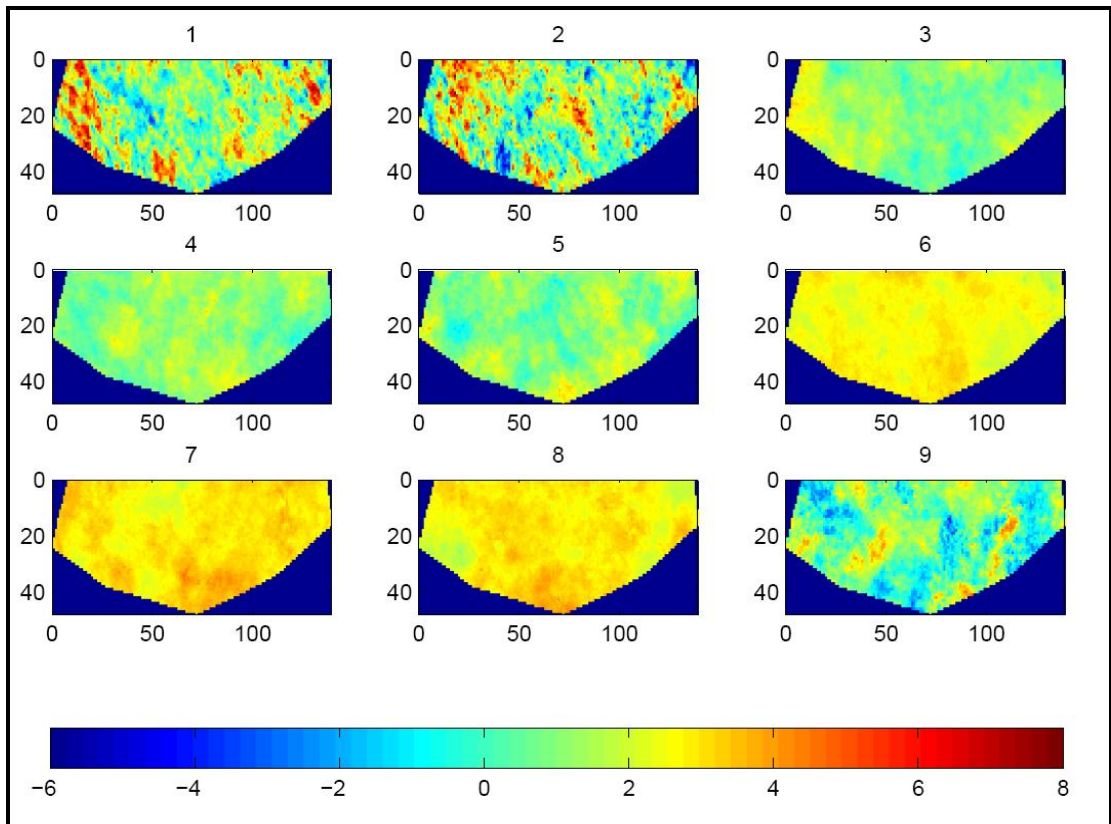


(b) PSO result

Figure 9.18: NTG for the 9 layers for EnKF (a) and PSO (b)



(a) EnKF result



(b) PSO result

Figure 9.19: Natural logarithm of permeability in X direction for the 9 layers for EnKF (a) and PSO (b)

9.3.5 *Location of Oil-Water Contact*

The location of the oil-water contact (both abbreviations OWC and WOC are used in the literature) can often be interpreted from 4D time-lapse seismic image surveys with some confidence. From the reservoir model, the location OWC can be obtained by inspecting the water saturation trend in each vertical column of cells in the model and using a threshold saturation value to point out the depth at which a transition takes place (Walker and Lane, 2007). In poorly porous intervals, the oil-water, gas-water or gas-oil contacts can be concealed, which makes the assessment of hydrocarbon reserves difficult, noticeably showing the uncertainties included. The well's petrophysical information frequently used to verify and describe these further. Brugge reservoir reference pressure at a depth of 1700 m is 170 bar. The free water level is indicated to be positioned at 1678 m and OWC at 1670 (Geel, 2008).

There were two observations with the PSO results (as well as EnKF) that showed

- High pressures with low water rates for some wells or no water at others.
- Low pressures with high water rates.

This triggered a potential water elevation level problem in this study. We therefore have examined adding a new OWC parameter to be optimised with the prior range [5430,5480] ft. Improved PSO results are obtained by adding an OWC parameter with the initial 104 PCA parameters. OWC optimisation is shown in Figure 9.20 compared with the original optimisation where the dashed black line represents the 105 parameters misfit reduction plot and the blue line represents the one obtained with 104 parameters. Sampling history is depicted in Figure 9.21. A difference of about 17 meters (55 ft) is obtained from the one indicated in the study. Table 9.3 shows clearly the best misfit achieved with 105 parameters reduced the misfit by around a factor of two.

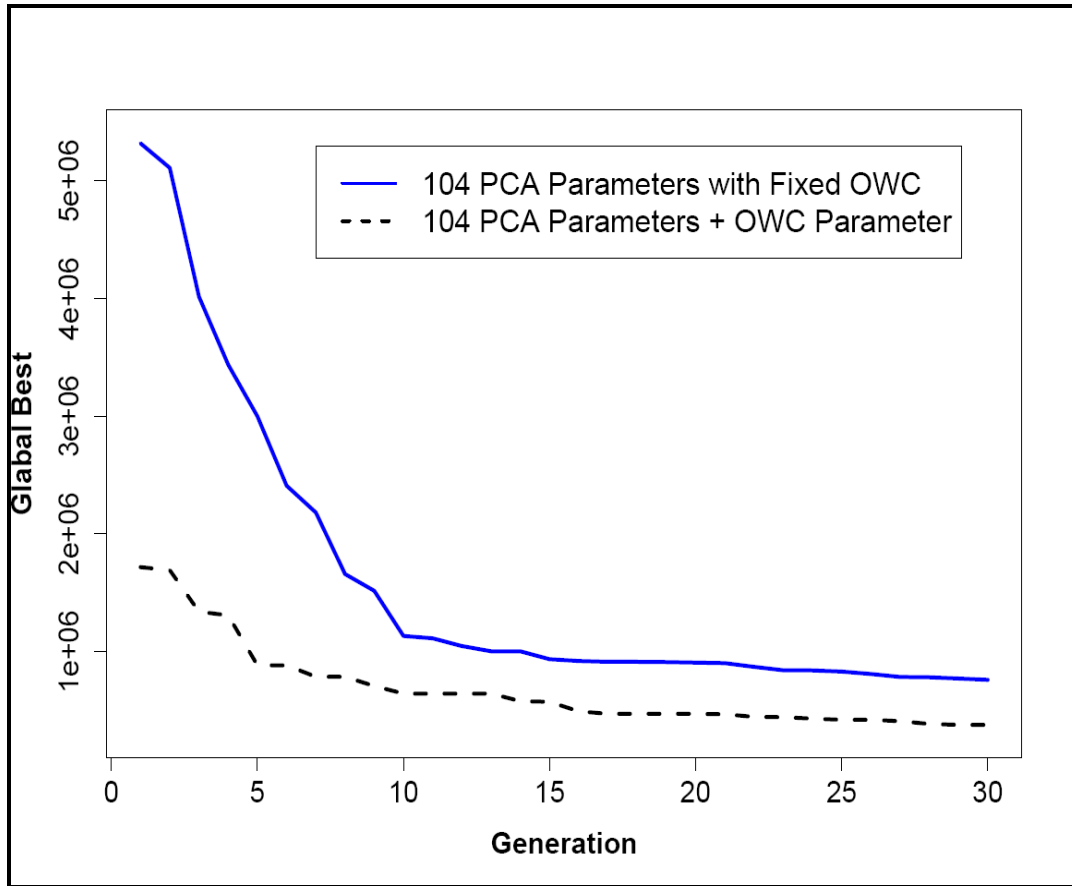
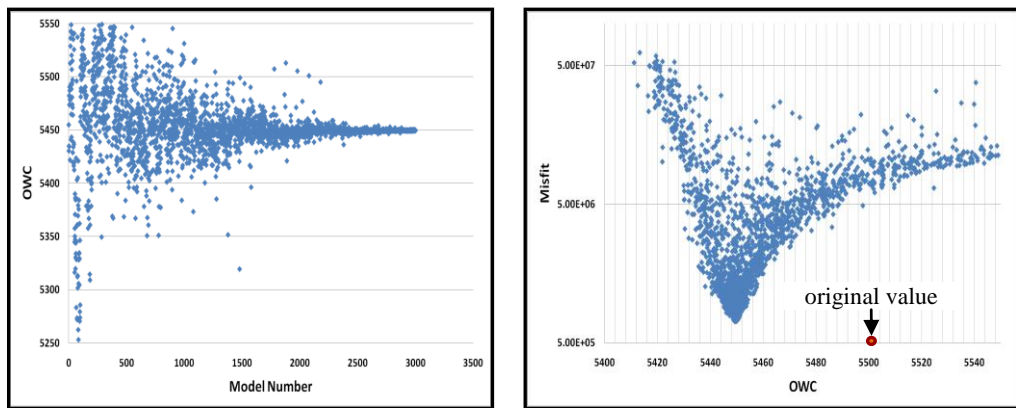


Figure 9.20: OWC optimisation adding value to misfit reduction



(a) Sampling History

(b) Misfit vs. OWC

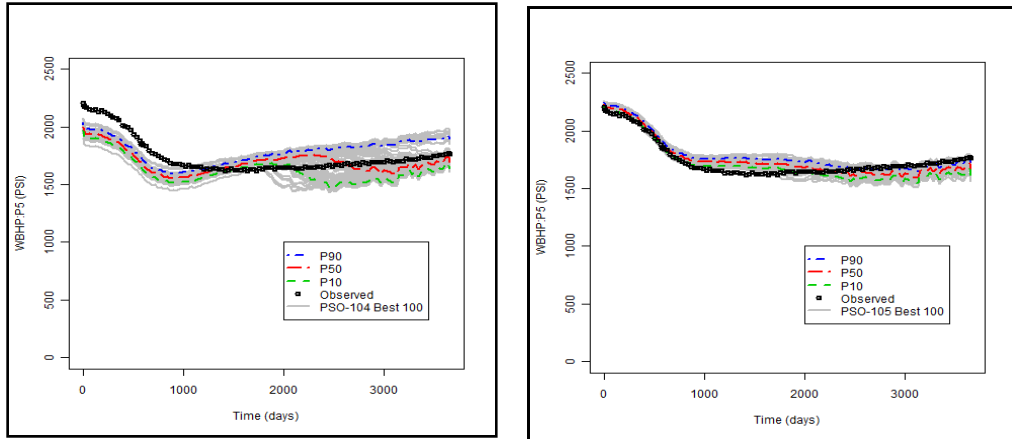
Figure 9.21: OWC addition

Table 9.3: Best misfits achieved both with and without adding OWC parameter

Number of parameters	Best misfits
104 parameters (PCA parameters)	758060
105 parameters (PCA parameters and OWC parameter)	375313

For examples of improvements, the p_{10} - p_{50} - p_{90} results obtained by adding the OWC parameter is shown in (b) of Figures 9.22 to Figure 9.28 compared to the results obtained with the 104 parameter earlier in (a). A complete list of history matching results is shown in Figures A.11 to A.15 in Appendix A.

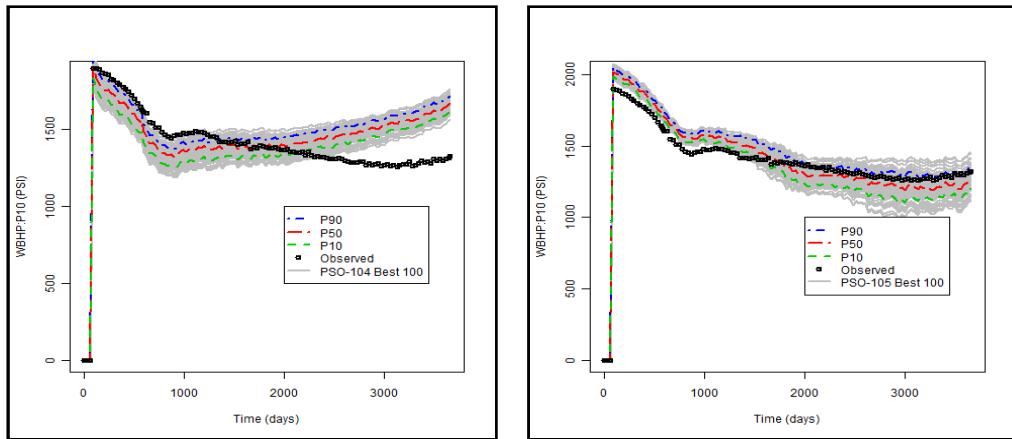
- Well pressure for producer 5, as shown in Figure 9.22(b) compared to previous in Figure 9.22(a).
- Well pressure for producer 10, as shown in Figure 9.23(b) compared to previous in Figure 9.23(a).
- Well pressure for producer 15, as shown in Figure 9.24(b) compared to previous in Figure 9.24(a).
- Water production rate for producer 5, as shown in Figure 9.25(b) compared to previous in Figure 9.25(a).
- Water production rate for producer 6, as shown in Figure 9.26(b) compared to previous in Figure 9.26(a).
- Water production rate for producer 10, as shown in Figure 9.27(b) compared to previous in Figure 9.27(a).
- Water production rate for producer 16, as shown in Figure 9.28(b) compared to previous in Figure 9.28(a).



(a) PSO – 104

(b) PSO – 105

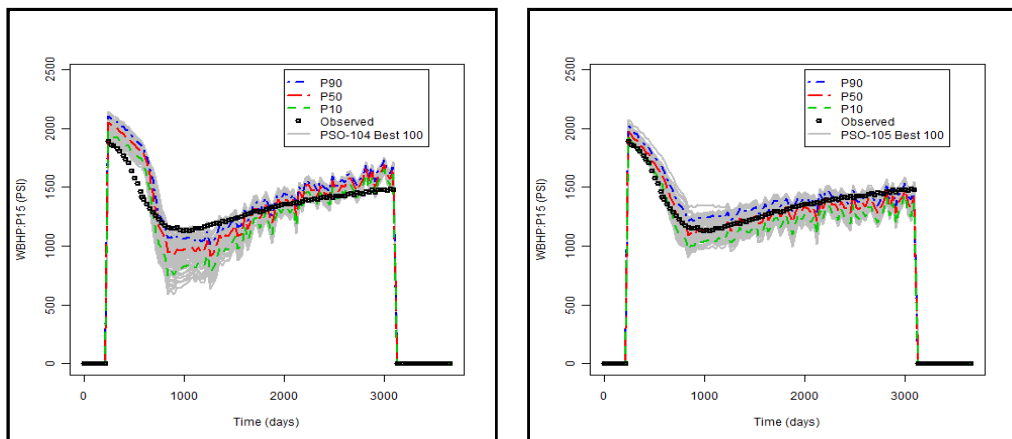
Figure 9.22: Well pressure history matches for producer 5 for PSO with 104 parameters (a) and with OWC parameter added (b)



(a) PSO – 104

(b) PSO – 105

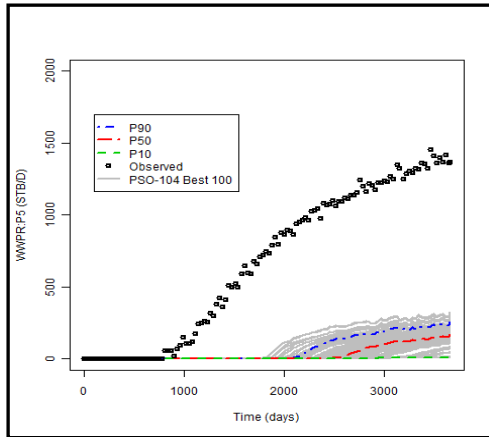
Figure 9.23: Well pressure history matches for producer 10 for PSO with 104 parameters (a) and with OWC parameter added (b)



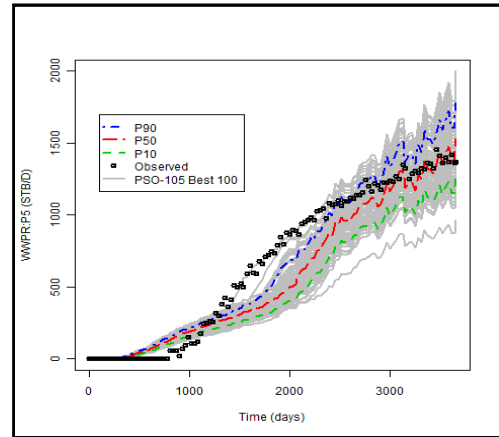
(a) PSO – 104

(b) PSO – 105

Figure 9.24: Well pressure history matches for producer 15 for PSO with 104 parameters (a) and with OWC parameter added (b)

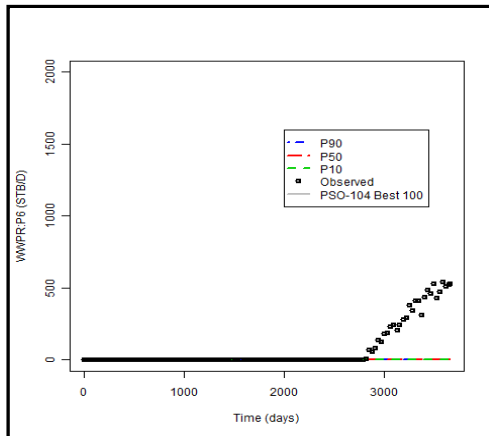


(a) PSO – 104

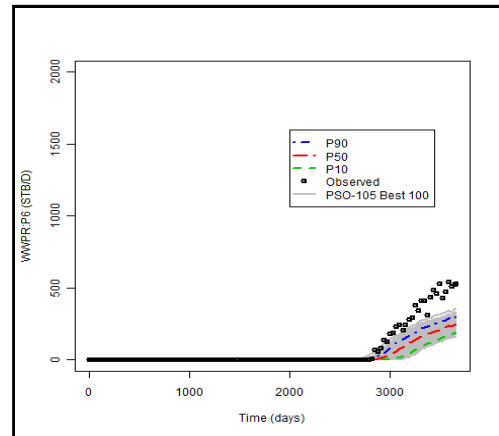


(b) PSO – 105

Figure 9.25: Well water production rate history matches for producer 5 for PSO with 104 parameters (a) and with OWC parameter added (b)

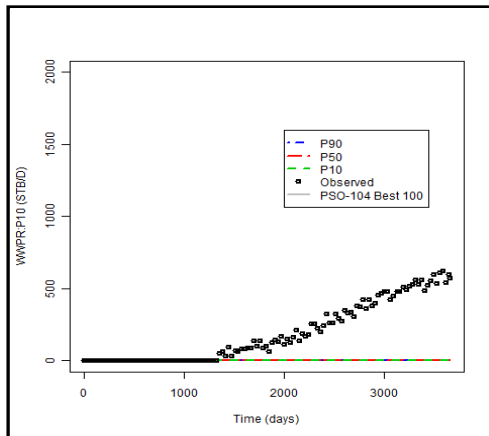


(a) PSO – 104

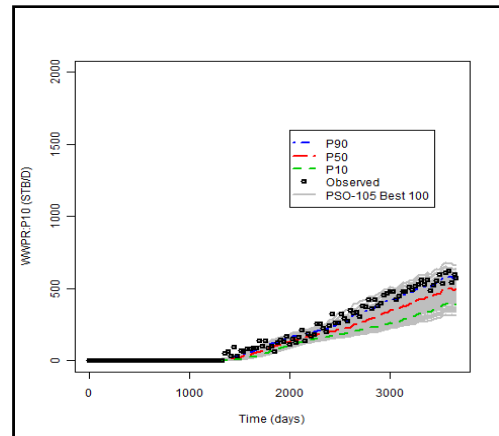


(b) PSO – 105

Figure 9.26: Well water production rate history matches for producer 6 for PSO with 104 parameters (a) and with OWC parameter added (b)



(a) PSO – 104



(b) PSO – 105

Figure 9.27: Well water production rate history matches for producer 10 for PSO with 104 parameters (a) and with OWC parameter added (b)

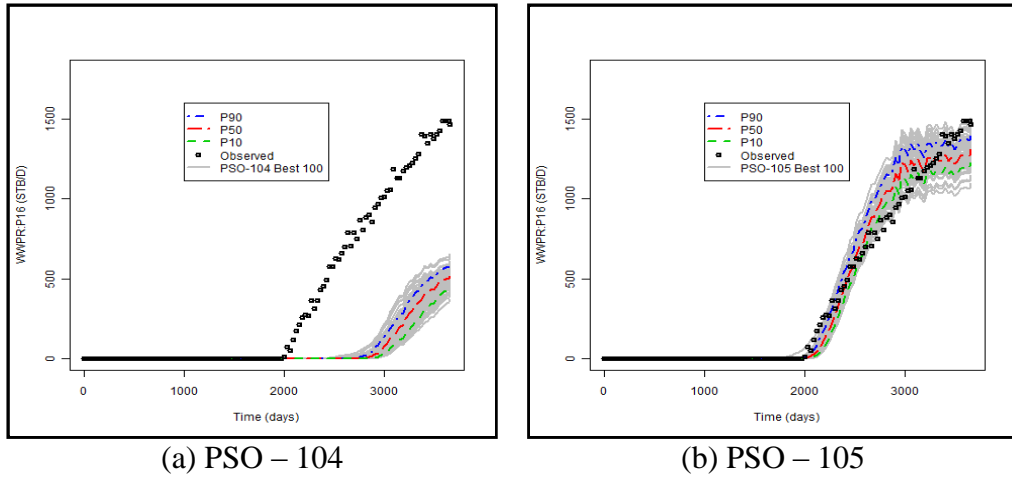


Figure 9.28: Well water production rate history matches for producer 16 for PSO with 104 parameters (a) and with OWC parameter added (b)

Figure 9.29 shows the initial field oil and water rates of the initial run of the 104 ensemble and the observations. The p10-p50-p90 uncertainty estimate equivalent plots are shown in Figure 9.30 for the models obtained with PSO (for both 104 parameters in (a),(b), and (c) and 105 parameters in (d),(e), and (f)). The field oil rate is well matched while the field water rate is captured particularly at the end of the historical period.

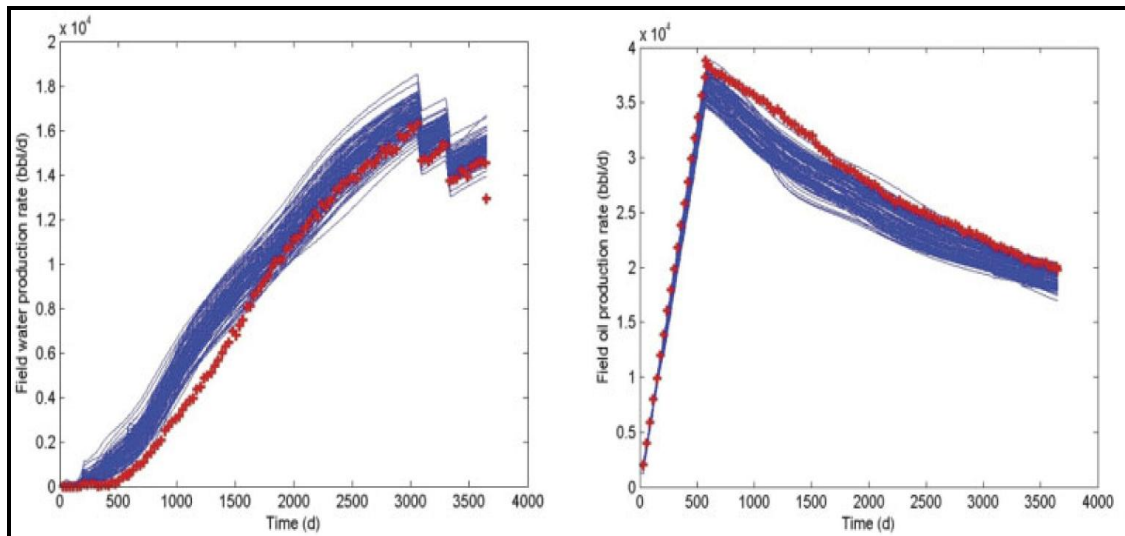
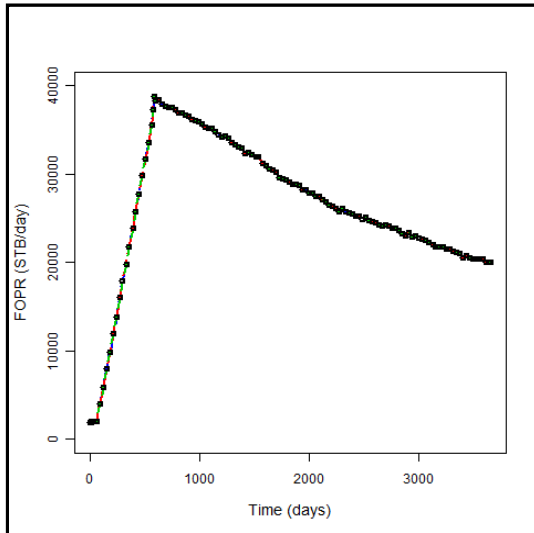
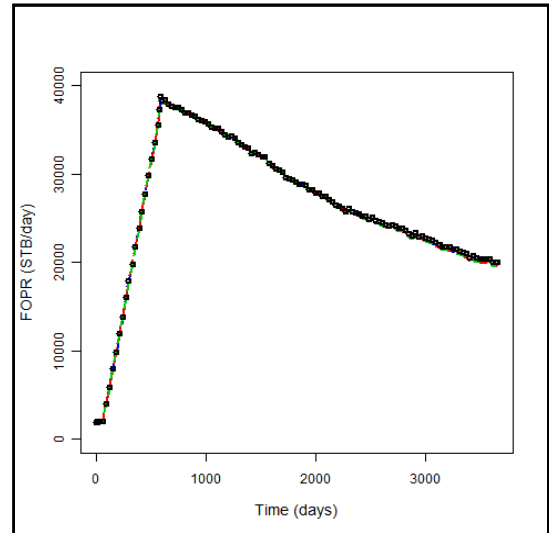


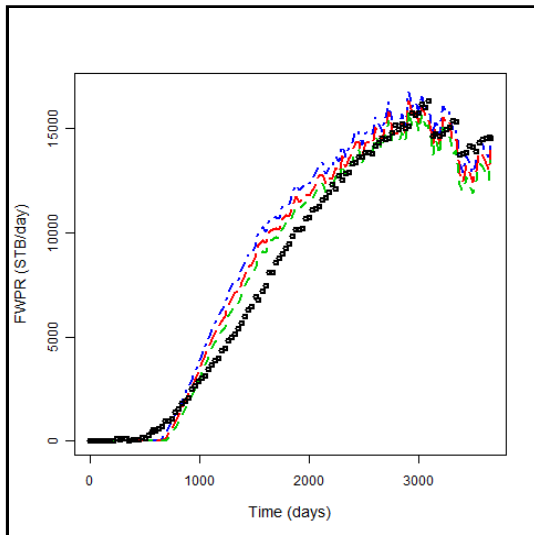
Figure 9.29: Field oil and water rates of the initial run of the ensemble and the observations



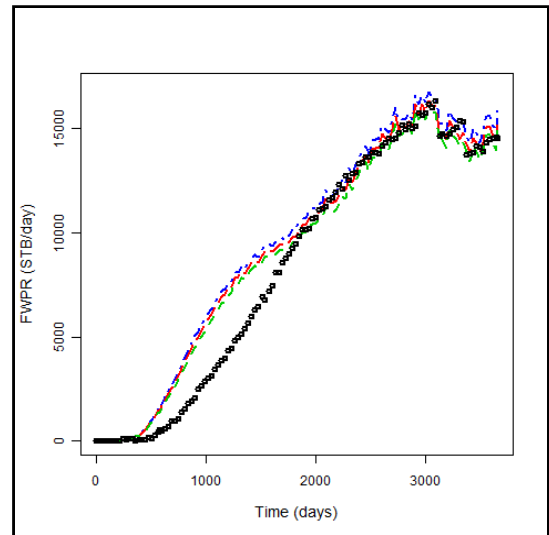
(a) PSO – 104



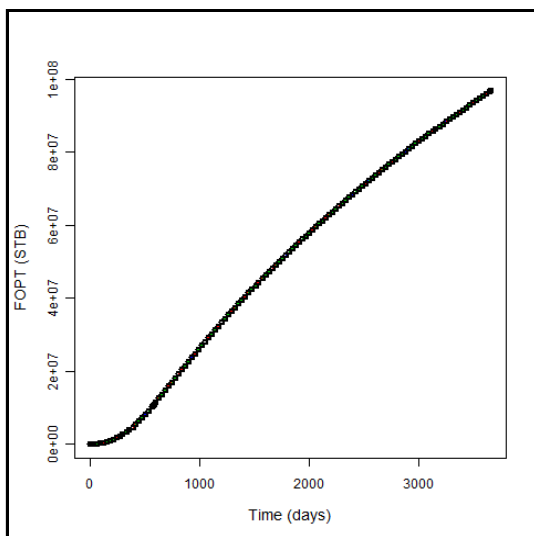
(d) PSO – 105



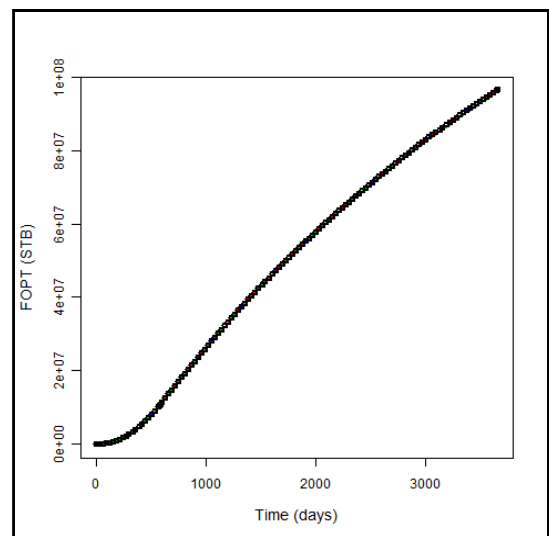
(b) PSO – 104



(e) PSO – 105



(c) PSO – 104



(f) PSO – 105

Figure 9.30: Field oil rate, field water rate, and total oil recovery of the two PSO runs – 104 and 105 parameters

The petrophysical properties for the nine layers: the NTG and permeability in X direction (in mD) for the best fitting model obtained for 105 parameters are shown in Figures (a) and (b) of 9.31. The corresponding porosity and permeability in Z direction (in mD) for the best fitting model are shown for completeness in (a) and (b) of Figure B.3 in Appendix B.

Figure 9.32 shows the misfit reduction when we changed the number of parameters to be the first 10, 20, 30, 40, 50, 60, and 104 PCA parameters plus an OWC parameter making total of 11, 21, 31, 41, 51, 61, and 105 respectively. The first PCA parameters, which correspond to the highest eigenvalues coefficients, would control the general features while the remaining PCA parameters control the geological details. Note that the values will not be ordered since these are the coefficients associated with the 104 eigenvalues. The best results were obtained when we used all the 104 PCA parameters indicating that the small scale geological details have a useful input in the optimisation procedure. When we have fixed the first 64 parameters properties by the values associated with the best misfit value obtained from the best model of the 104 parameters to study how that influenced the optimisation, the plot in Figure 9.33 has been obtained. This shows that the optimisation is sensitive to the number of parameters used since the misfit values are higher than the one original with 104 parameters. In addition, the misfit has been reduced during the optimisation procedure.

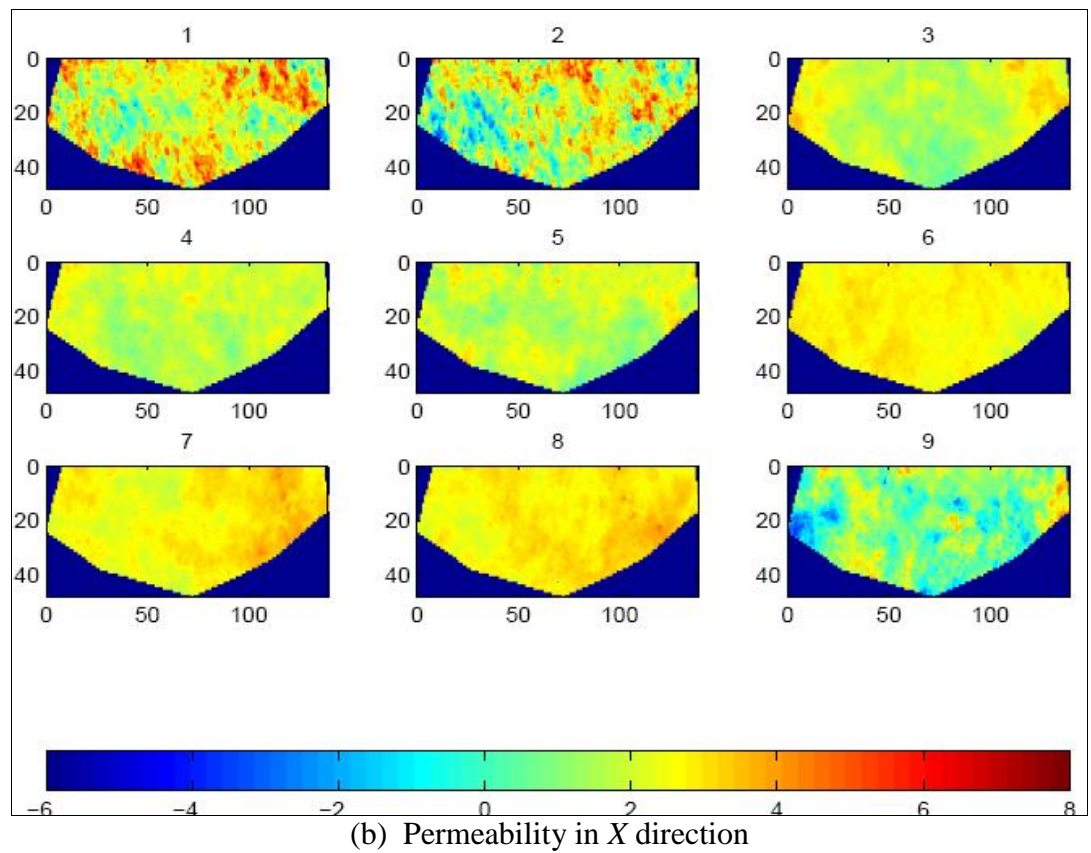
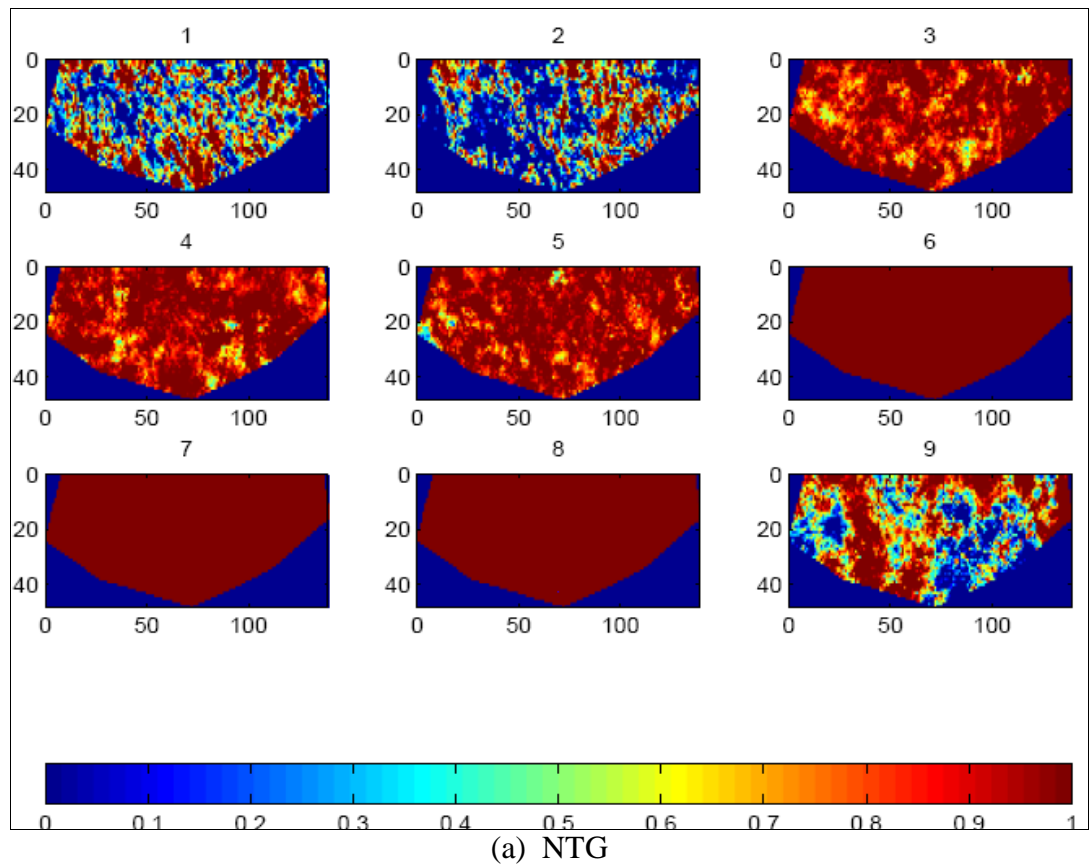


Figure 9.31: OWC result for the 9 layers with NTG (a) and natural logarithm of permeability in X direction (b)

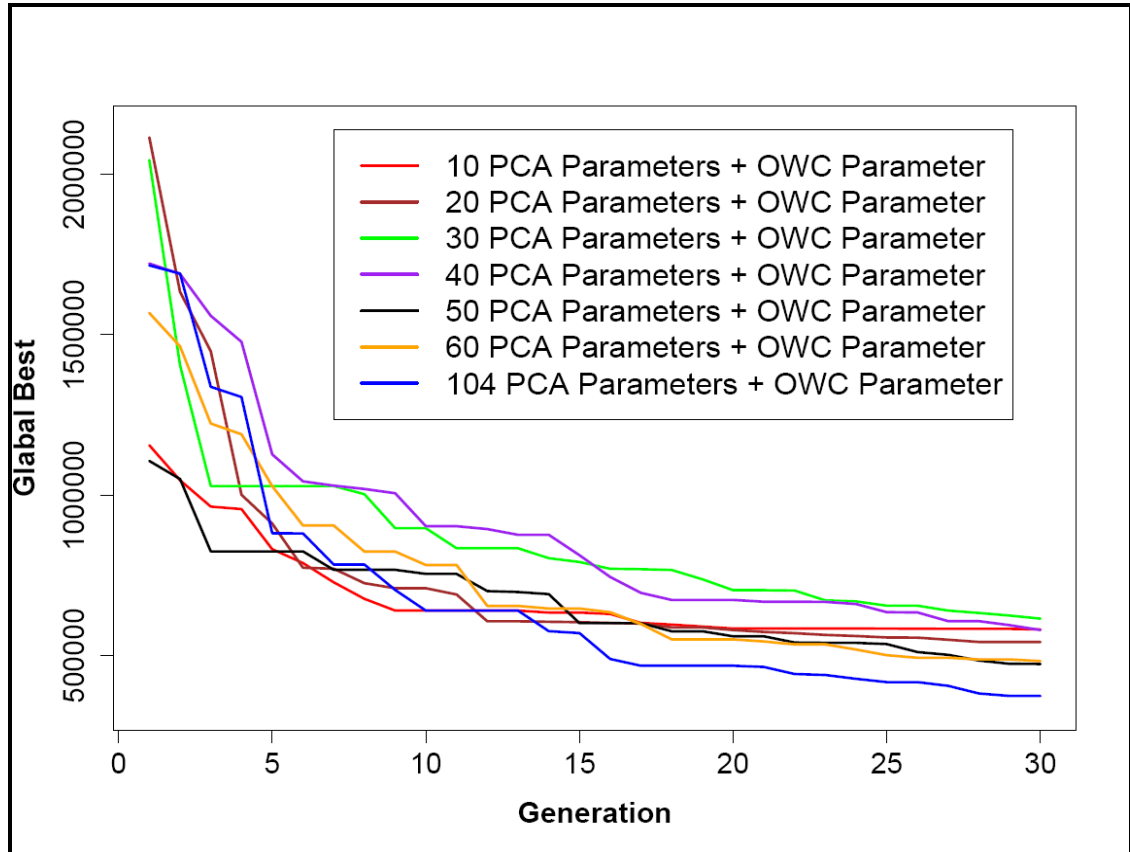


Figure 9.32: Misfit reduction using different number of parameters

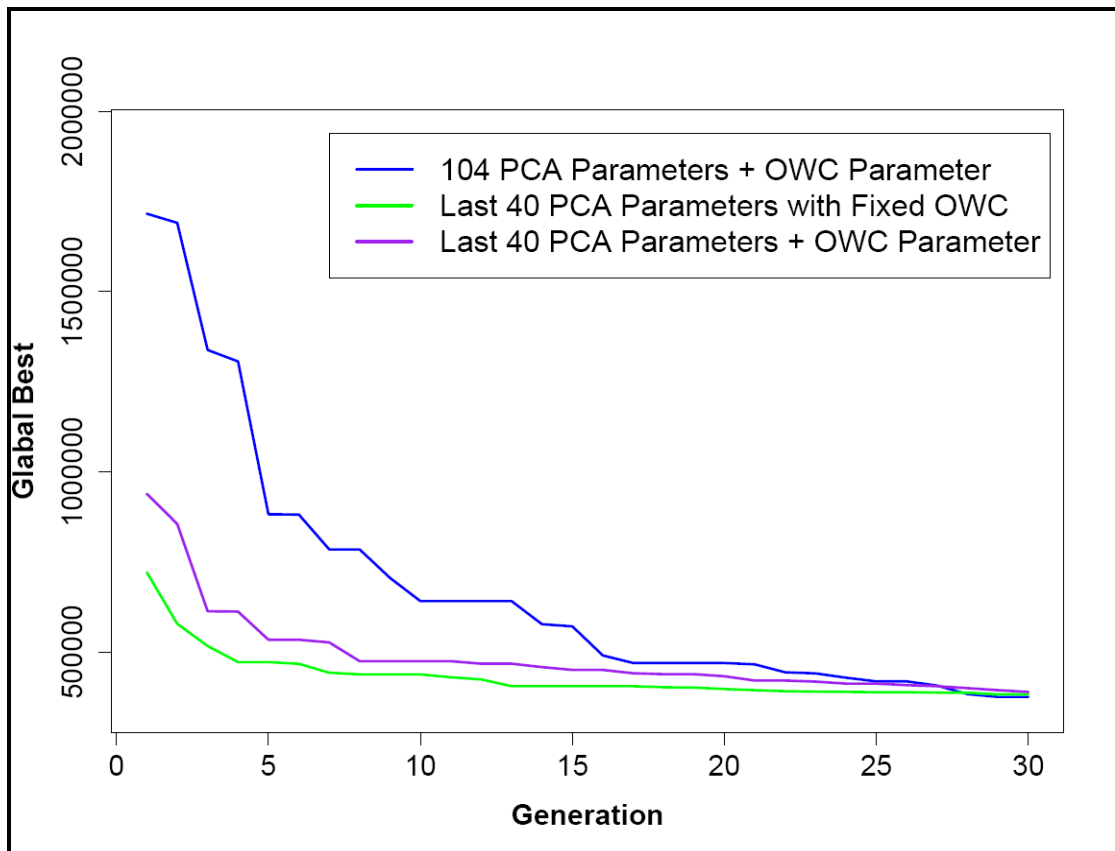


Figure 9.33: Misfit reduction using 105 parameters, 40 parameters (with fixed OWC parameter), and 41 (including variable OWC parameter). The first 64 parameter values are set from the best fitting model obtained from the 105 parameter case

We also showed the results of using different swarm sizes: 10, 20, 40, 50, 60, and 100. Figure 9.34 shows the misfit reduction results of changing the swarm size for the PSO while using the same total number of simulations. The best misfit achieved was obtained with 30 particles per generation shown in Table 9.4.

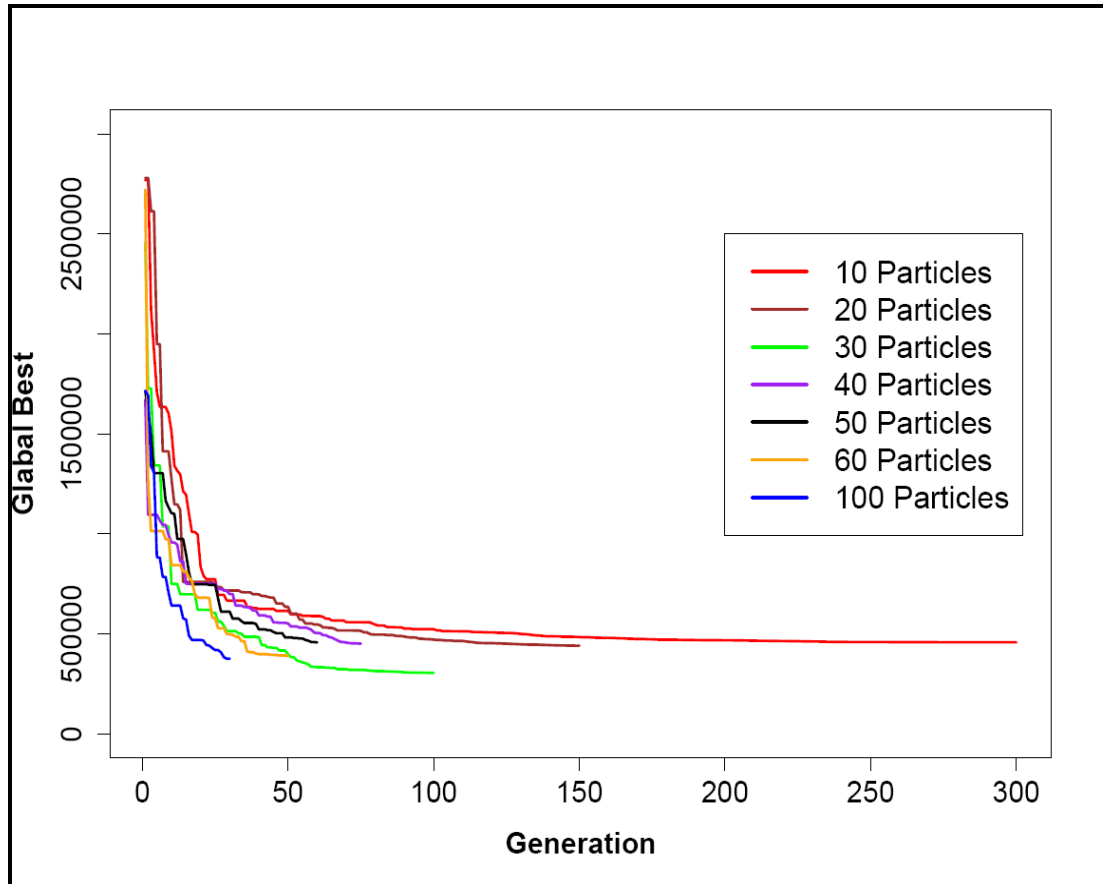


Figure 9.34: Misfit reduction using different swarm sizes

Table 9.4: Best misfits achieved with two swarm sizes – 105 parameters

Number of particles	Best misfits
100	375313
30	303948

9.3.6 Kernel Principal Component Analysis Application

In this section we report the results of applying nonlinear PCA with high order polynomial kernels. The kernel PCA parameters are optimised with PSO. We used a second (KPCA2) and third (KPCA3) order polynomial kernels. Figure 9.35 shows the misfit reduction comparison when applying the Linear PCA (KPCA1), kernel PCA with

order two and three. KPCA3 has reduced faster, although the best misfit achieved has been obtained with the KPCA1. KPCA 2 started from a higher value than both KPCA1 and KPCA3 and reducing sharply at the first stages of optimisation although it has obtained a higher misfit value at the end than the other two.

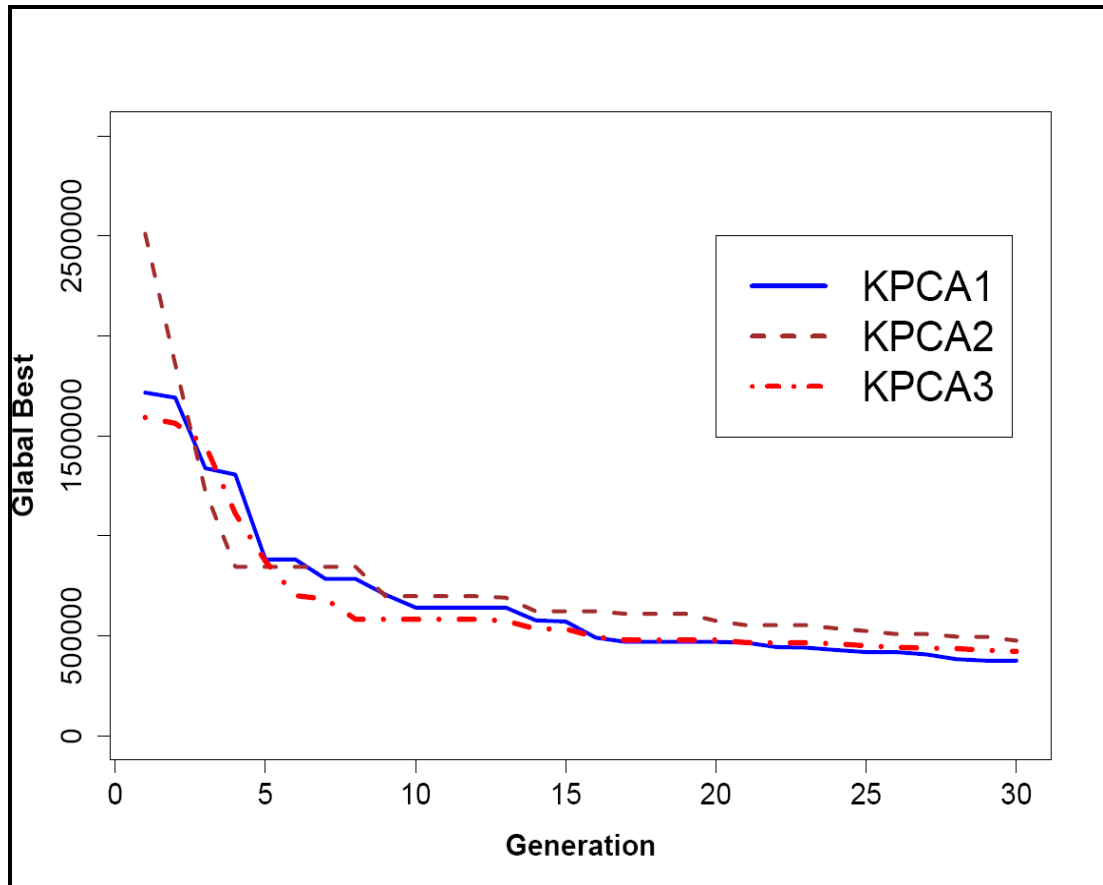
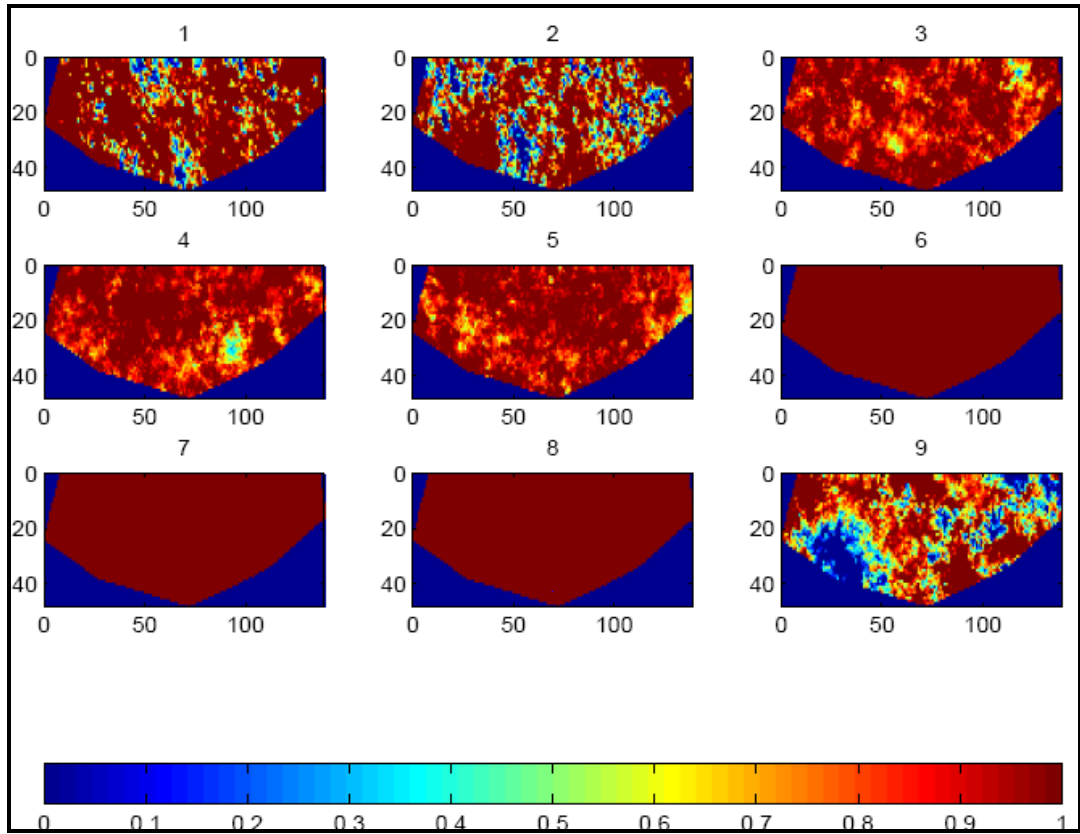
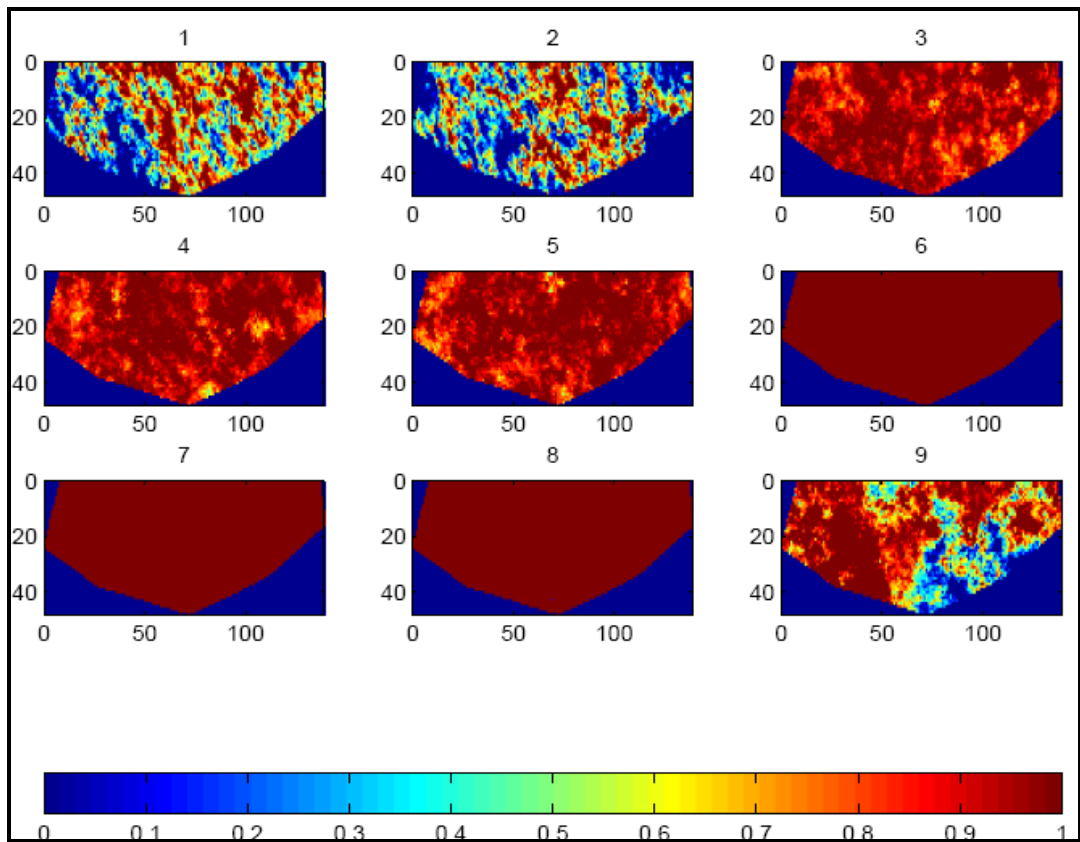


Figure 9.35: Misfit reduction comparison of KPCA1, KPCA2, and KPCA3

The petrophysical properties for the nine layers: the NTG and permeability in X direction (in mD) for the best fitting model obtained for KPCA2 and KPCA3 are shown in Figures (a) and (b) of 9.36 and 9.37. The porosity maps for the best fitting model obtained for KPCA2 and KPCA3 are shown in (a) and (b) of Figure B.4 in Appendix B. The corresponding permeability fields in Z direction (in mD) for the best fitting model obtained for KPCA2 and KPCA3 are shown in (a) and (b) of Figure B.5 in Appendix B. Heterogeneity is observed in KPCA2 and KPCA3 in comparison to KPCA1 in Figure Figures 9.18(b), 9.19(b) and 9.31.

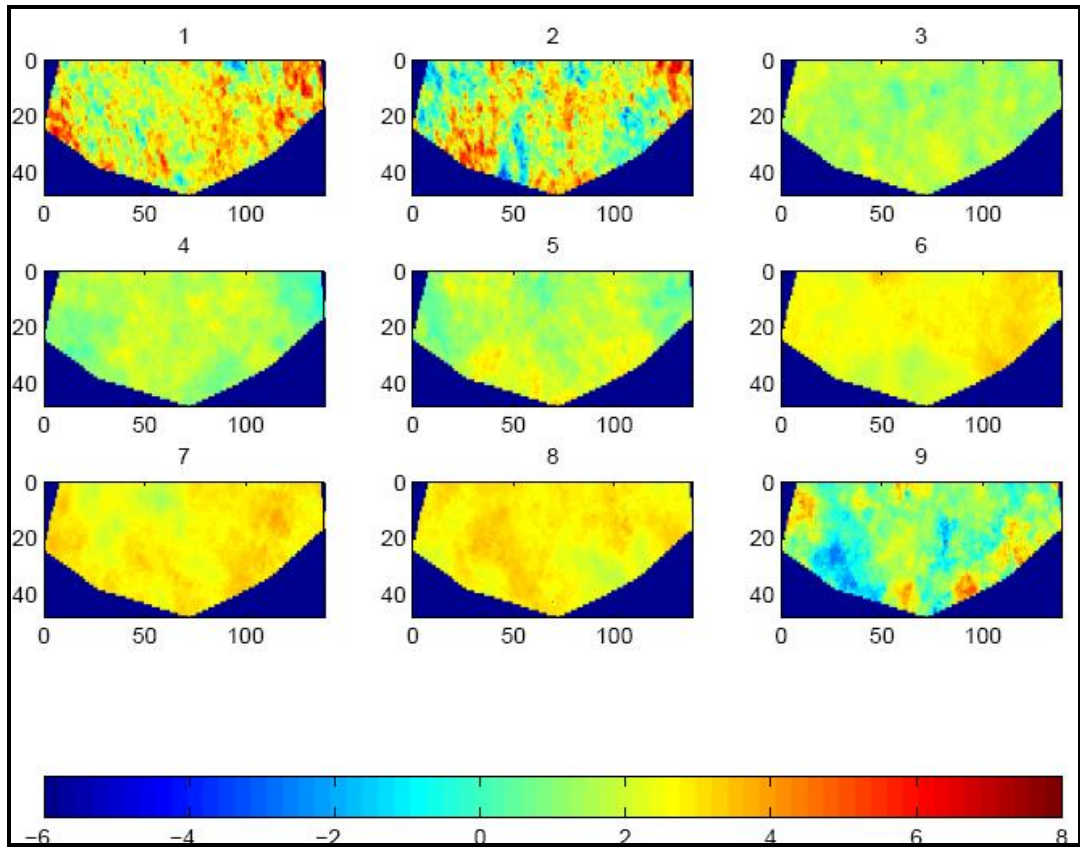


(a) KPCA2 result

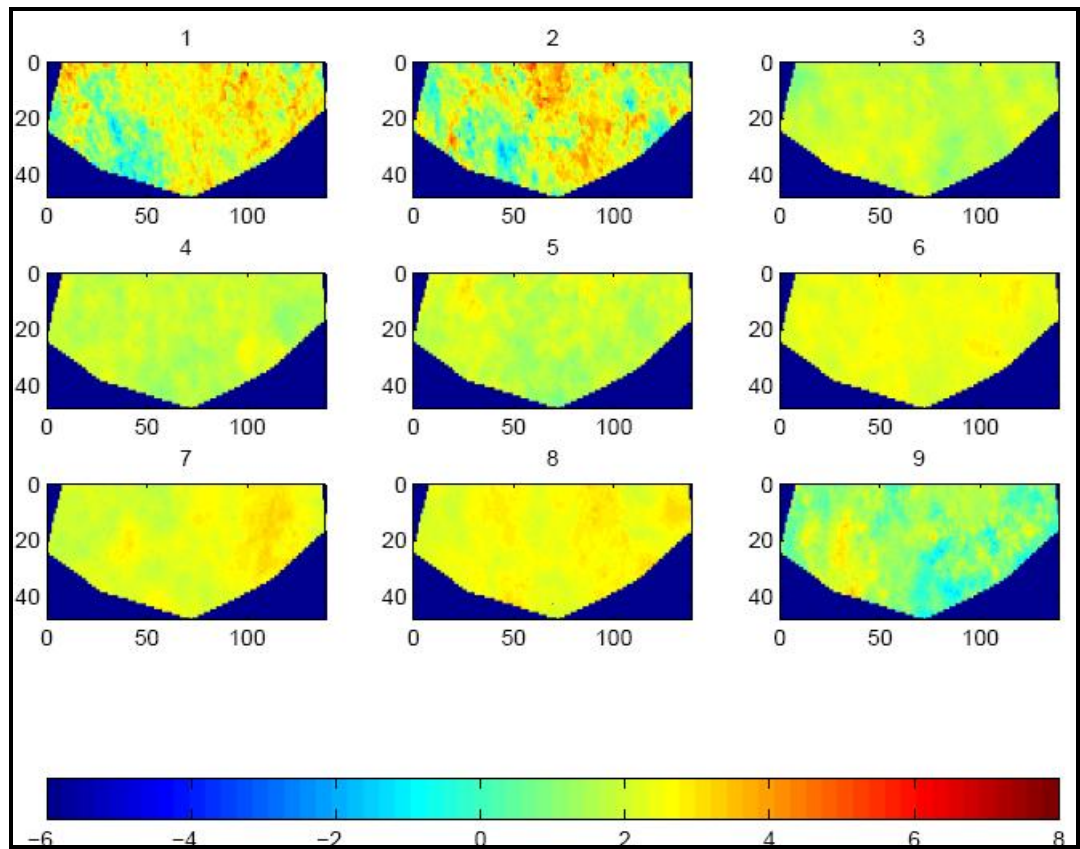


(b) KPCA3 result

Figure 9.36: NTG for the 9 layers for KPCA2 (a) and KPCA3 (b)



(a) KPCA2 result



(b) KPCA3 result

Figure 9.37: Natural logarithm of permeability in X direction for the 9 layers for KPCA1 (a) and KPCA3 (b)

9.4 Chapter Summary

In this chapter we have applied Particle Swarm Optimisation (PSO) on a large synthetic test case with 104 parameters and have obtained good matches with successful parameterisation using Principal Component Analysis (PCA). Further improvements for PSO results were obtained when adding an OWC parameter. Principal Component Analysis is successfully applied to deal with large number of parameters. PSO optimisation was faster particularly at early generations when kernel PCA is used although the results were more or less comparable to the linear PCA one.

Based on comparative analysis with EnKF the following specific conclusions could be drawn:

- PSO has produced good history matching results comparable with EnKF.
- EnKF has obtained better pressure matches for some producers.
- PSO has obtained wider ranges and captured the observed water rates better for some wells.
- EnKF has obtained the good history matching results with 104 simulations while PSO has obtained similar history matching performance in a higher number of simulations.
- Continuity was observed in PSO realisations compared to EnKF ones where heterogeneity is observed.

It is hard to argue that either algorithm is a clear winner for this problem, which suggests that each algorithm has its own strengths and weaknesses, and will work well for specific problems that suit it and may not work on other problems.

While our study does not show that PSO outperforms EnKF in all cases, it does illustrate the need for a range of algorithms to be available for history matching.

Chapter 10 – Summary and Conclusions

The main aim of this thesis is to investigate and develop effective techniques that can produce models capable of improving the statistics of oil predictions in petroleum industry in a faster and more reliable way. Advancement in computer technology has encouraged researchers to develop novel computer science techniques and software for history matching and uncertainty quantification. An emphasis in this study has been on how different optimisation and inference techniques affect the recovery from petroleum systems. This has led to the examination and development of novel state-of-the-art optimisation and inference methods from computer science, statistical and mathematical background. Importantly, examining the techniques working mechanisms can provide clear insights into what problems suit them best. Synthetic and simple field studies have been used to inspect systematically the techniques and to make sound reasonable concluding remarks. This chapter summarises the main contributions of this thesis. Furthermore, key findings from this research have suggested a number of areas which warrant further investigation and these will be provided towards the end.

Four petroleum case studies served and contributed as test examples to carry out the research and experiments for the thesis. The Teal South reservoir presented in Chapter 4 is an oil field located in the Gulf of Mexico. Because of realistic nature, existence of measurement errors, and the model simplicity in having only one well, it was used to examine the efficiency of different optimisation techniques. The model represents an eight-parameter optimisation problem, a reasonable high-dimensional problem for testing the performance of the techniques and realistic setup to devise and develop visualisation tools to help in analysing results for the rest of the work performed.

The second case study is the IC Fault Model introduced in Chapter 4. While the synthetic model is a low-dimensional model having only three parameters, it represents a challenging optimisation problem because of it has highly irregular misfit surface and thus be difficult to match for assisted history matching evolutionary and swarm intelligence optimisation algorithms. This model is a significant platform for investigating how different optimisation mechanisms in sampling complex surfaces can influence the predictions and studying how to handle such cases with developed methods.

The third study is PUNQ-S3, presented in Chapter 5, which is a synthetic model having a number of wells sufficient to identify multiple objectives whilst involving simplicity in terms of time required to carry out simulations for the intention of confirming the effectiveness of multi-objective history matching approach. It was used to serve that purpose in this thesis; to give insights prior to tackling complex models with huge number of wells and large historical data. The model is a high-dimensional problem involving 45 parameters.

Finally, the last model is the Brugge model, introduced in Chapter 9, which is a synthetic model based on real field representing what we see in a real field as a fairly complex model having 30 smart wells with their oil rates, water rates and pressures recorded for 10 years. The Brugge model is not too complex model to prevent progress in carrying out experimental tests nor too small to invalidate developed methodologies and conclusions. The study was developed by the Netherlands Organisation for Applied Scientific Research (TNO). Brugge field was originally constructed and tailored to test close-loop particle filter methods, and recognised to represent a challenge for evolutionary optimisation algorithms to be at a disadvantage. Thus, it was used to illustrate how such cases can be adapted to be used with evolutionary methods. The developed model is high-dimensional problem including 105 parameters.

10.1 Thesis Summary

The thesis has been structured into ten chapters to carry out the study. Chapter 4 has introduced the high performance global particle swarm optimisation (PSO) algorithm to navigate the parameter space and to find an ensemble of history matched reservoir models that are used to probabilistically quantify the uncertainty in Bayesian framework with the use of the NAB routine for resampling and postprocessing. PSO is a swarm

intelligence algorithm uses a swarm of particles composed of positions representing the models and velocities. PSO search mechanism incorporates acceleration and damping parameters to accelerate towards multiple solutions. The technique has very interesting exploration – exploitation capabilities. The thesis has studied basic modifications of the PSO that greatly influence the performance of the method in order to experiment and understand the reliability of the technique. The examined different variants of PSO include four boundary strategies dealing with particles which fly outside the boundary of the search space and four inertial weight choices in which an inertia parameter, representing the fraction of previous flight that could be integrated in the next one, is modified (sometimes with the cognition and social components). The technique has been tested on three reservoir examples. The first example is the Teal South reservoir with 8 unknown parameters, the second one is the 3-parameter multimodal IC Fault model, and the third is the Brugge model. It is seen that the forecasted uncertainty envelopes are influenced by two factors: both the quality and diversity of the models in the PSO ensemble obtained. It is shown that PSO has the flexibility in converging fast towards good solutions as well as in carrying out global exploration depending on the task. The use of static inertial weights like Trelea Set Type I' ($c_1 = c_2 = 1.494$, $\omega = 0.729$) and I'' ($c_1 = c_2 = 1.7$, and $\omega = 0.6$), converges faster to good fitting regions in parameter space provided that the boundary strategy is fixed in most cases. This leads to a fewer number of reservoir simulation runs while the dynamic versions (linear decreasing and linear increasing inertial weights) maintain diversity of the reservoir models. The absorbing and damping boundary strategies in most cases have faster convergence speed while random and reflecting boundary strategies maintain the diversity. This behaviour is robust. In the Teal South model the smoothness of the surface leads to similar predictions while in the low-dimensional IC Fault model the results varied according to the diversity of models found. Testing different variants for different problems is a way forward to ensure the reliability of the produced inferences and choosing the variant that is more suitable to a particular problem. For instance, linear decrease inertial weight with absorbing boundary strategy was shown to be suitable for high-dimensional problems because there is a large possibility that many local minima are located at boundaries in high spaces.

Assisted history matching algorithms in previous research studies primarily focused on optimising a single objective function in which all the objective function components

are aggregated into a single number. Whereas, history matching process is physically multi-objective since there are multiple wells and measurements are recorded at different times. An alternative innovative methodology proposed is the multi-objective approach where we aim at optimally balancing the different objectives simultaneously while maintaining solutions diversity. The advantageous of such representation is that it is possible to examine the trade-offs between the objectives and construct a reservoir model that makes an equivalent level of match for all the reservoir objectives. In other words, none of the multiple objectives is dominating, in which case the set of non-dominated solutions is called Pareto optimal front.

Chapter 5 has investigated Multi-Objective Particle Swarm Optimisation (MOPSO) scheme on the IC Fault and PUNQ-S3 synthetic models. The conclusions from the two case studies have been that the MOPSO scheme gives flexibility in optimising different objectives simultaneously and obtains better history matches with faster convergence speed (based on the way the objectives are aggregated that has been seen in PUNQ-S3). Consequently, the number of simulations required for achieving a similar matching performance has been reduced with faster estimation of uncertainty. In addition, the multi-objective optimisation approach can also be used to understand and analyse the simulations and group the conflicting objectives in different ways. Geological spatial correlations, the energy balance of the reservoir, and calculating the contribution of each objective to the misfit function can be employed as ways of grouping the objectives. The observations from the PUNQ-S3 example where two aggregation approaches were used have an impact on the speed convergence and quality of history matches. Yet, the differences in uncertainty estimates were insignificant. Since, the results of aggregation have been obtained on the single field PUNQ-S3 dataset, further studies on more complex fields will be needed to establish definitive guidelines. The implementation of PSO and MOPSO is achieved in a synchronous master/slave parallelisation scheme that can be easily coupled with any other method.

Chapter 6 has presented the application development of the novel Hamiltonian Monte Carlo (HMC) algorithm for uncertainty quantification. HMC is a Markov Chain Monte Carlo (MCMC) inference technique that combines the characteristics of the Hamiltonian dynamics and the Metropolis algorithm to sample complex posterior distributions. The developed HMC approach integrates gradient information estimated with General

Regression Neural Network to address the random walk problem in the classical Metropolis. The method could use adjoint code directly for gradients if available. HMC generates samples from the posterior with one step. The work carried out on Teal South as a realistic case showed that for its smooth surface HMC results were comparable to that of other stochastic methods. In some complex cases such as the IC fault model case, more work need to be carried out as outlined in Chapter 8.

Chapter 7 investigated the efficiency of the three stochastic sampling algorithms. The algorithms are compared by generating multiple history matched reservoir models for the Teal South and IC Fault model. The conclusion drawn from the smooth Teal South study was that all three methods NA, PSO and HMC are able to find equivalent match qualities and equivalent forecasts of uncertainty. PSO with the random boundary strategy (not the fastest variant) had faster convergence than NA for the example. Furthermore, in the complex IC Fault example, the NA, with different tuning parameters and the same population size used for PSO, failed to obtain a close estimate of uncertainty to the benchmarked database one and got trapped in local minima, while in PSO some variants were able to do so. Using a large size of population for those variants can help in finding even more diverse solutions. On the other hand, HMC samples are localised in regions which are close to the initial states. Proposed solutions to deal with these cases have been discussed in Chapter 8.

Chapter 8 has presented the first application of a Population MCMC (Pop-MCMC) technique to generate multiple history matched models. Pop-MCMC combines techniques from evolutionary algorithms and parallel MCMC algorithms to design new algorithms for sampling or optimising complex distributions. The sampling efficiency of Population MCMC has been tested on the IC Fault Model that has a highly irregular misfit surface caused by the nonlinearity between model solutions and simulated reservoir response, and hence is difficult to match. The fundamental idea of Population MCMC parallel tempering is to enable the system to “exchange” configurations corresponding to differently “tempered” distributions, allowing the sampler to explore the state space in a more flexible way. Population MCMC was able to generate samples very effectively from the complex, multimodal posterior probability distribution. By comparison, previous results from stochastic sampling algorithms often focus on only part of the region of high posterior probability depending on algorithm settings and

starting points as shown in Chapter 7. Pop-MCMC algorithm provides better sampling and more robust inference than standard MCMC or previous evolutionary algorithms. Clearly, the merger of the evolutionary algorithms and MCMC paradigms represent a rich source for future development of powerful sampling and optimisation algorithms. A further study on combining the GP emulator with Pop-MCMC is ongoing with promising initial results.

Global evolutionary or swarm intelligence optimisation methods are often employed to solve global optimisation history matching problems with a small number of parameters to adjust around hundreds which hinders the use of global optimisation algorithms in large problems. Chapter 9 has illustrated that the sampling and optimisation can be achieved in a reduced model space through the combined use of high performance global algorithms such as Particle Swarm Optimisation and model reduction techniques or kernel methods such as Principal Component Analysis (PCA). PCA has been performed on a set of prior scenarios that were constructed from integrating prior geological information via stochastic-based simulation approaches. PCA benefits from the assumption, which is realistic, that there exist correlations between model parameters originated from the physics of the forward problem. Thus, through PCA application, a new good fitting ensemble of geomodels has been constructed to preserve and integrate in correspondence with the prior realisations. This merged PSO-PCA facilitates tackling thousands of parameters encountered in real world applications and uncertainty analysis around the minimum objective function solution. The developed approach can be utilised with any other global optimisation technique.

Chapter 9 has also investigated Particle Swarm Optimisation for history matching and uncertainty on a large synthetic test case with 105 parameters and compared history matching performance results with Ensemble Kalman Filter. PSO has obtained good matches with successful parameterisation using Principal Component Analysis (PCA). Further improvements for PSO history matching results were obtained when adding more parameters. Based on our comparison with EnKF it is hard to tell which algorithm is the better method of the two suggesting that each algorithm has its own strengths and weaknesses, and will work well for problems that suit them.

10.2 Thesis Contributions

The contributions of the thesis can be summarised as follows

1. Developing efficient simple variants of Particle Swarm to history matching optimisation. The variants have the flexibility in converging fast towards good solutions and carrying out global exploration.
2. Developing Hamiltonian Monte Carlo algorithm for uncertainty quantification that uses approximated gradients.
3. Comparison of PSO, HMC, and Neighbourhood Algorithm (NA) for history matching and showing that PSO variant obtained by coupling the random strategy with linear decreasing inertial weight choice, PSO–LDR variant, has faster convergence than NA. The algorithms based on Hamiltonian dynamics and swarm intelligence concepts have the potential to be effective tools in history matching and uncertainty quantification.
4. Introduction of Population MCMC that provides better sampling and more robust results.
5. Introduction of innovative Multi-objective Particle Swarm Optimisation (MOPSO) to petroleum engineering. MOPSO can obtain a diverse set of better history matches with faster convergence speed than the standard single objective case (SOPSO). Thus, the number of simulations required for achieving a similar matching performance has been reduced with faster estimation of uncertainty.
6. Coupling Particle Swarm Optimisation and Principal Component Analysis (PSO-PCA) for reservoir modelling and demonstrating that the approach is applicable to practically large problems.
7. Comparison of Particle Swarm Optimisation (PSO) and Ensemble Kalman Filter (EnKF).

10.3 Overall Conclusion

The performance of sampling algorithms has an impact on the predictive inferences in reservoir simulation, and therefore careful uncertainty assessment analysis is required to enhance confidence in the predictions. In this thesis, the efficient particle swarm optimisation has been proposed for history matching and uncertainty quantification in a Bayesian framework. The algorithm is extended to enhance diversity of models for more confidence in oil recovery predictions with the introduction of a multi-objective particle swarm scheme. It has been shown that with the global exploration characteristics of the particle swarm, the uncertainty can be quantified effectively. In

addition, the history matching optimisation with particle swarm optimisation is faster than the neighbourhood algorithm and can obtain comparable uncertainty estimates compared with ensemble Kalman filter approaches.

Advanced MCMC techniques such as Hamiltonian Monte Carlo and Population MCMC, which have been introduced here, provide much better sampling than conventional sampling methods, including MCMC and evolutionary methods, and hence more accurate statistics of oil predictions. They both have the ability to sample directly from the posterior distribution. Population MCMC merges the advantages of MCMC and evolutionary algorithms by using multiple chains in parallel and establishes faster convergence rate by exchanging information between the parallel chains. Population MCMC provides much better sampling than conventional sampling methods including MCMC and evolutionary methods.

The applicability of these methods has been demonstrated using synthetic and real field applications. Furthermore, prior geological models can be integrated to obtain new possible geomodels through the use of kernel methods. Coupling particle swarm with model reduction techniques for predicting reservoir properties can further produce realisations that not only encapsulate prior geological knowledge from different sources about the field but also significantly match the dynamic production data for more improved inferences and better understanding of the problem.

The thesis has demonstrated the value and benefits to the petroleum industry of integrating and testing the most recent and efficient computer science algorithms in petroleum engineering problems. The employment of the new algorithms significantly speeds up history matching procedures and improves the accuracy of uncertainty estimations substantially.

10.4 Future Perspectives

I conclude this thesis with the following suggestions for further research

- The particle swarm optimisation algorithm is a population-based algorithm and thus it has parallel nature characteristics that can take full advantage of the scalability of computer clusters. The application in the thesis used a global neighbourhood topology when exchanging information about swarm best values and positions, (the star topology scheme). The synchronous parallelisation

scheme is used in which the swarm best value, particle best remembered positions, velocities and fitness values are updated on a per swarm basis, rather than on a per individual basis. Other parallelisation schemes exist such as finely and coarsely-grained parallelisation as indicated in Chapter 4, as well as the ring, wheel and other topologies that can be implemented. A careful study is required with the aim to investigate the solution characteristics, parallel speedup and efficiency as well as maintaining the load balance between processors when the other new parallelisation schemes are used.

- As highlighted in Chapter 5, there is a vital need to see how the developed Multi-Objective Particle Swarm Optimisation (MOPSO) approach can be exploited and validated in realistic complex field examples with a large number of wells to come up with guidelines to identify multiple objectives. Novel and interesting methods which need to be investigated and introduced to the petroleum industry and to be compared with MOPSO are Multi-Objective Honey-Bees Mating Optimisation (MOHBMO) and Multi-objective Shuffled Complex Evolution Metropolis (MOSCEM). These two innovative techniques have performed good results in water resources engineering inverse problems (Barros et al., 2008; Barros et al., 2010) which are similar to history matching inverse problems.
- Chapter 8 has introduced the novel advanced Population MCMC to deal with complex response surfaces as a proof of concept where the number of simulations was not taken into account. There are many ways to reduce the number of forward simulations required in the study for practical application in industry that have not been achieved yet. Firstly, Hamiltonian Monte Carlo (introduced in Chapter 6) or Riemann Manifold HMC (Girolami and Calderhead, 2011) can be incorporated in place of the componentwise Metropolis-Hastings sampler. Secondly, a Gaussian Process (GP) could be used to emulate the simulator and reduce the number of expensive full simulations for Population MCMC indicated. This is an interesting line of further research in combination with Population MCMC. The Population MCMC technique also needs to be investigated in real examples as the ultimate goal.
- New hybrids methods that merge the evolutionary algorithms and MCMC paradigms as well as with other evolutionary or swarm intelligence algorithms represent a rich source for future development of powerful sampling and

optimisation algorithms. Other combinations of MCMC with evolutionary algorithms are available as well as combinations with the evolutionary algorithms that take the strengths of different techniques.

- As mentioned in Chapter 9, the results of Ensemble Kalman Filter and Particle Swarm Optimisation may further be improved by coupling the localisation step with EnKF. Applying this step with the PCA coupling may be a fruitful line to be examined for improving efficiency.
- The development of geomodels based on kernels like Kernel PCA and Partial Least Squares Regression while coupling with global optimisation algorithm is one of the prospective, interesting and challenging areas for future research. Kernel methods are advanced methods that have the capacity of integrating geological features at different scales, better understanding, and consistency to the problem. Thus, incorporating more geological knowledge can help to provide more confidence in the predictions. There are also still gaps that warrant further investigation related to the pre-image problem highlighted in Chapter 9 that maps back to the realisation space from feature space. Gaps in these areas have been articulated to affect kernel methods generally, and there is a need to invest effort in tackling this problem.

Appendix A – Brugge Reservoir History Matching Results

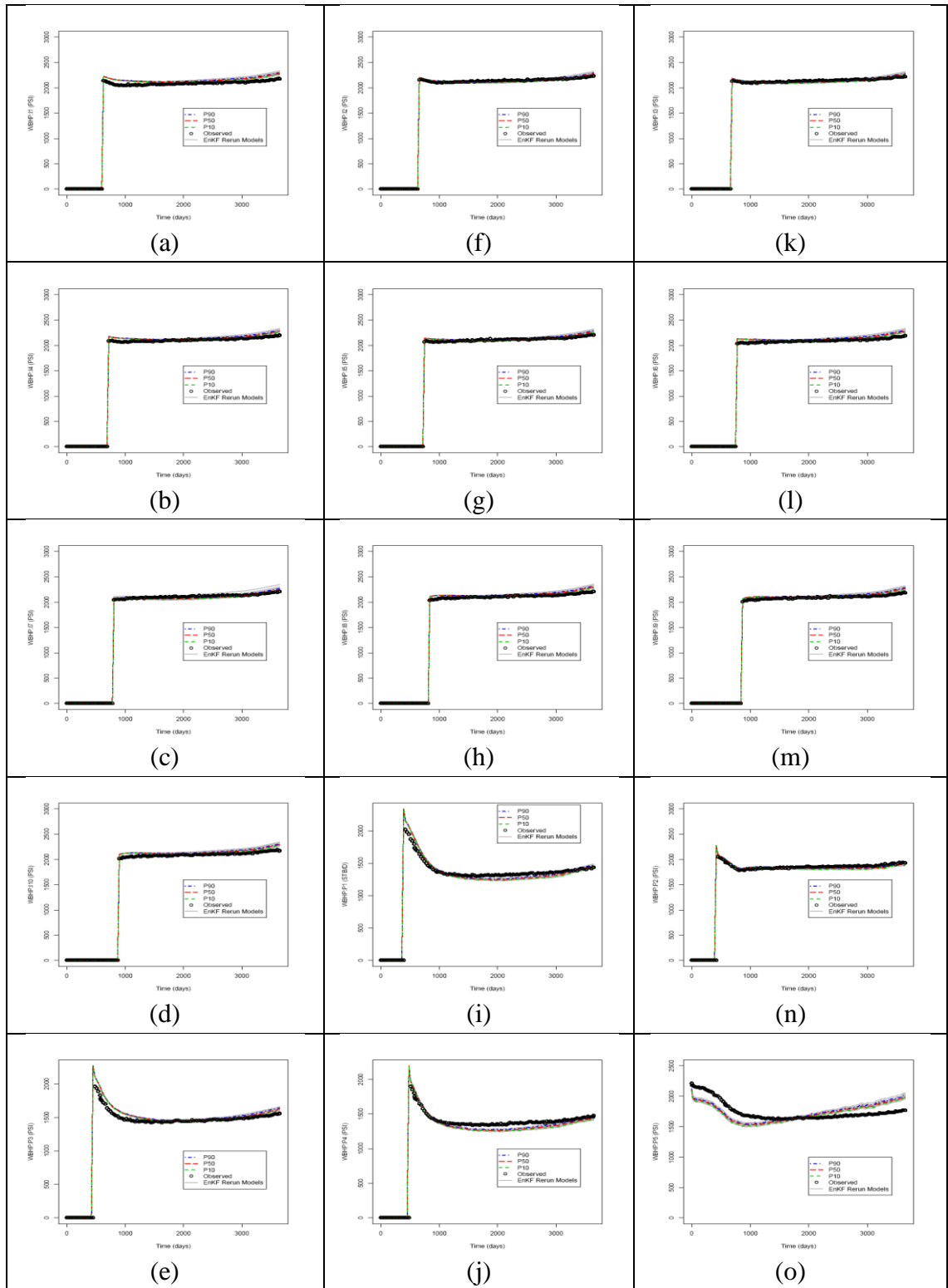


Figure A.1: EnKF p10-p50-p90 results

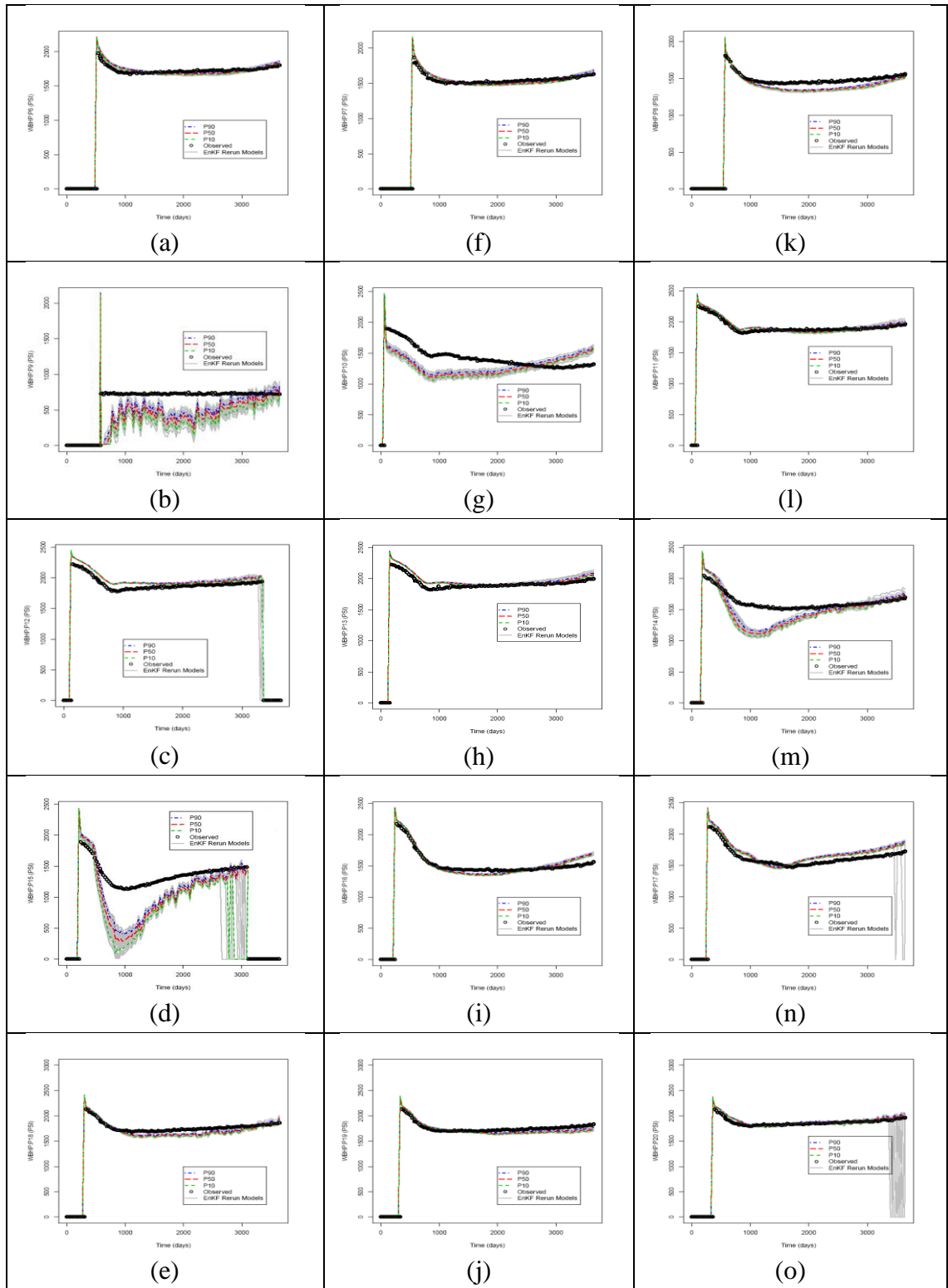


Figure A.2: EnKF p10-p50-p90 results

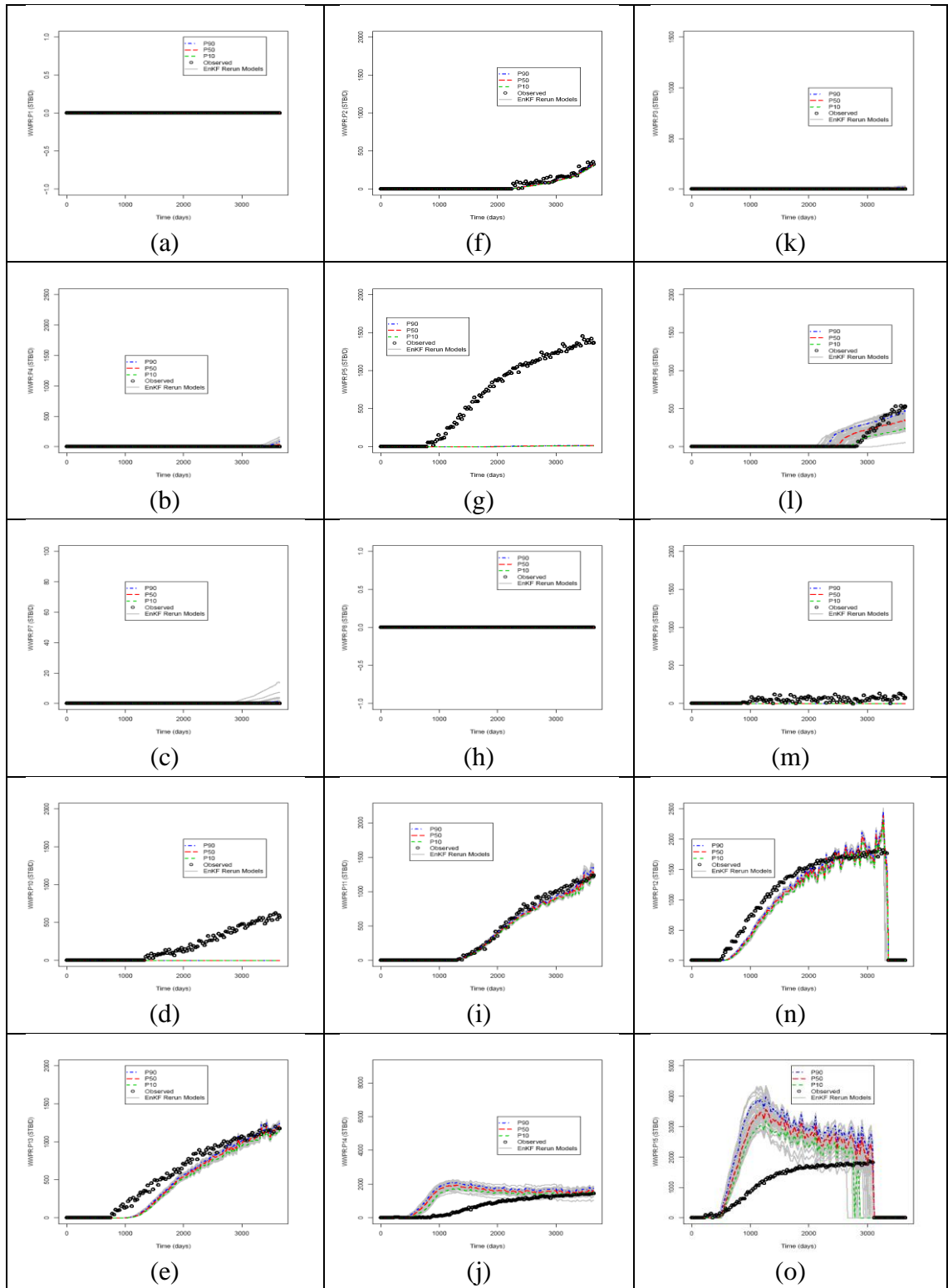


Figure A.3: EnKF p10-p50-p90 results

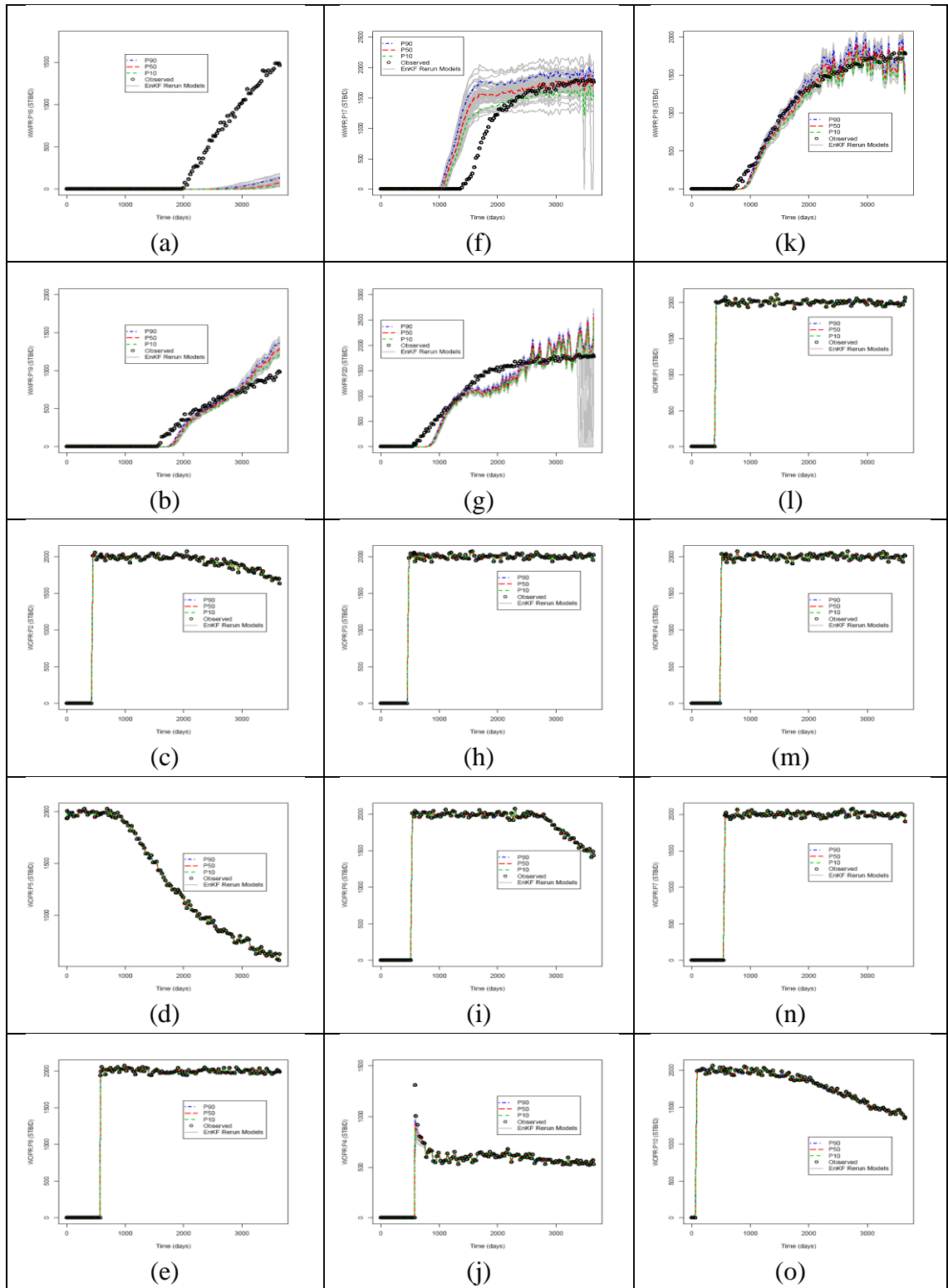


Figure A.4: EnKF p10-p50-p90 results

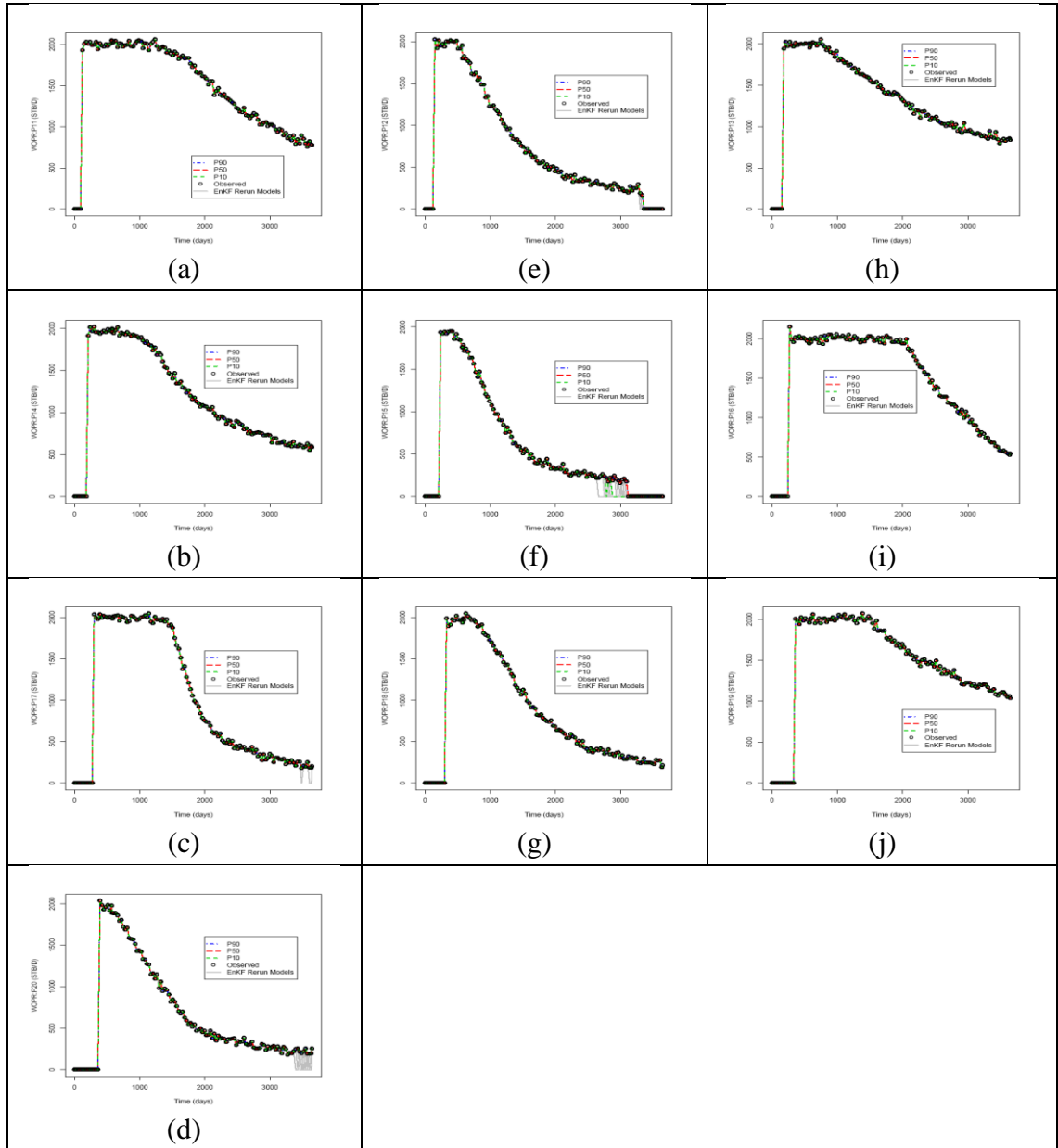


Figure A.5: EnKF p10-p50-p90 results

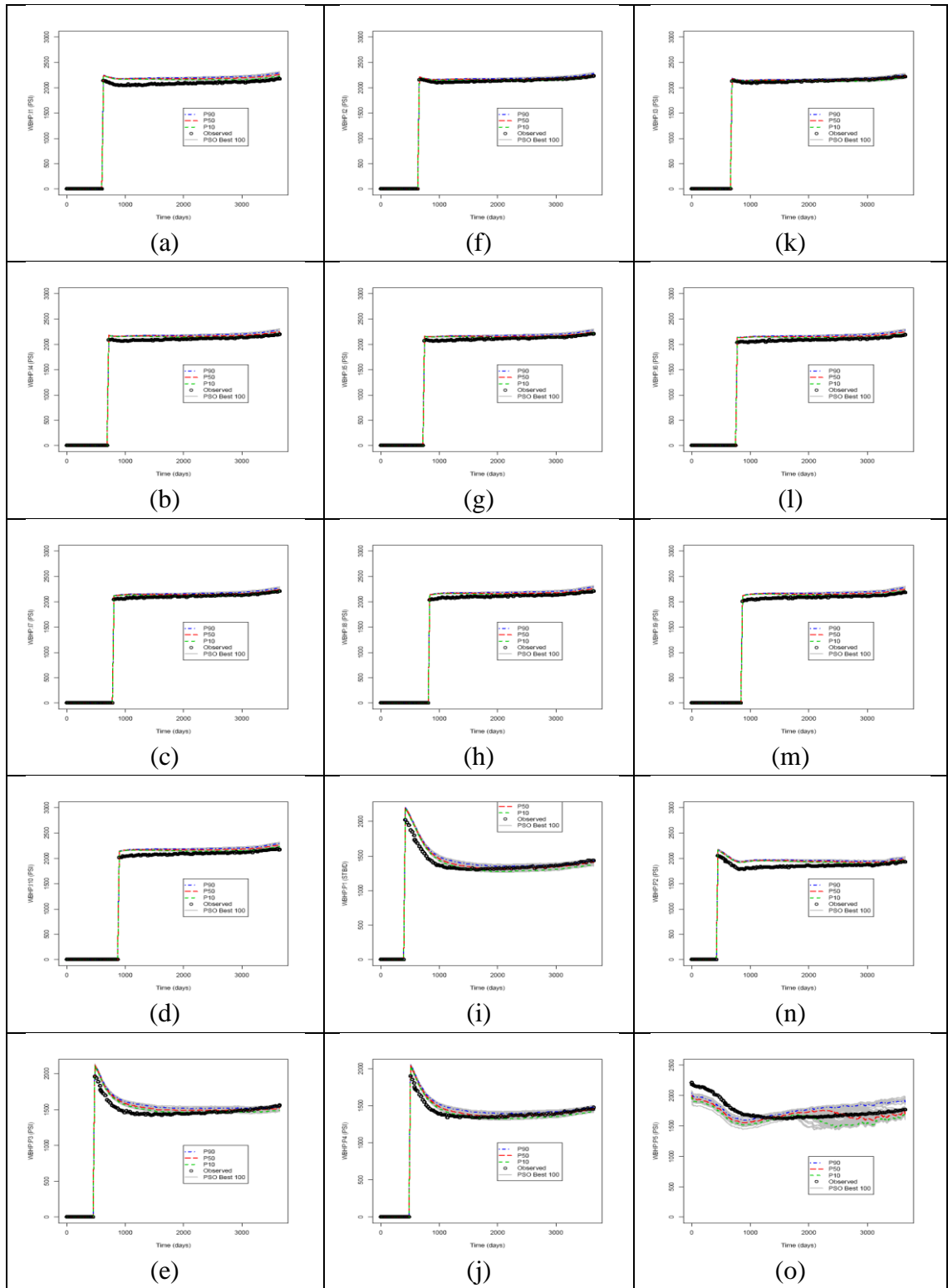


Figure A.6: PSO p10-p50-p90 results

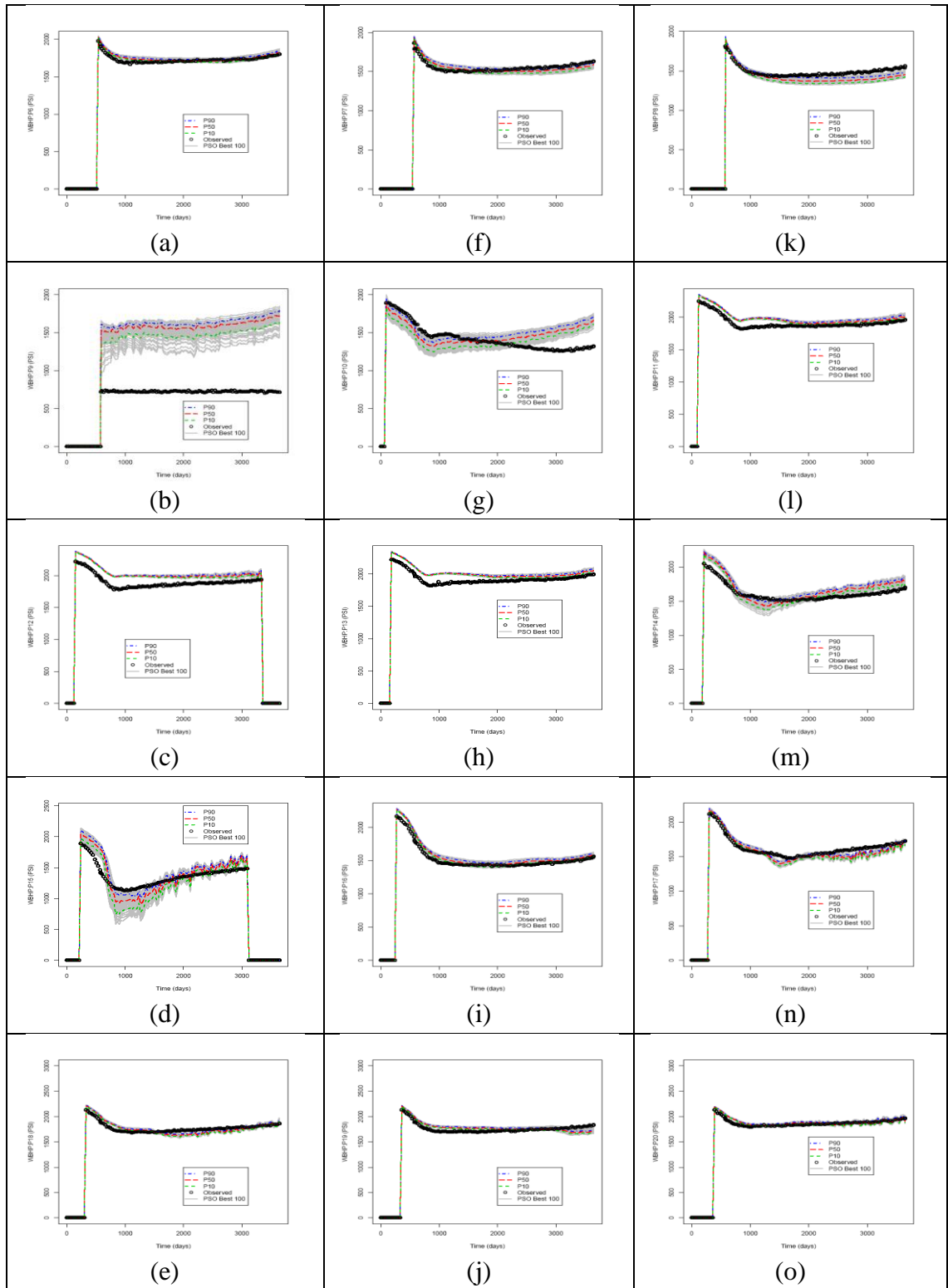


Figure A.7: PSO p10-p50-p90 results

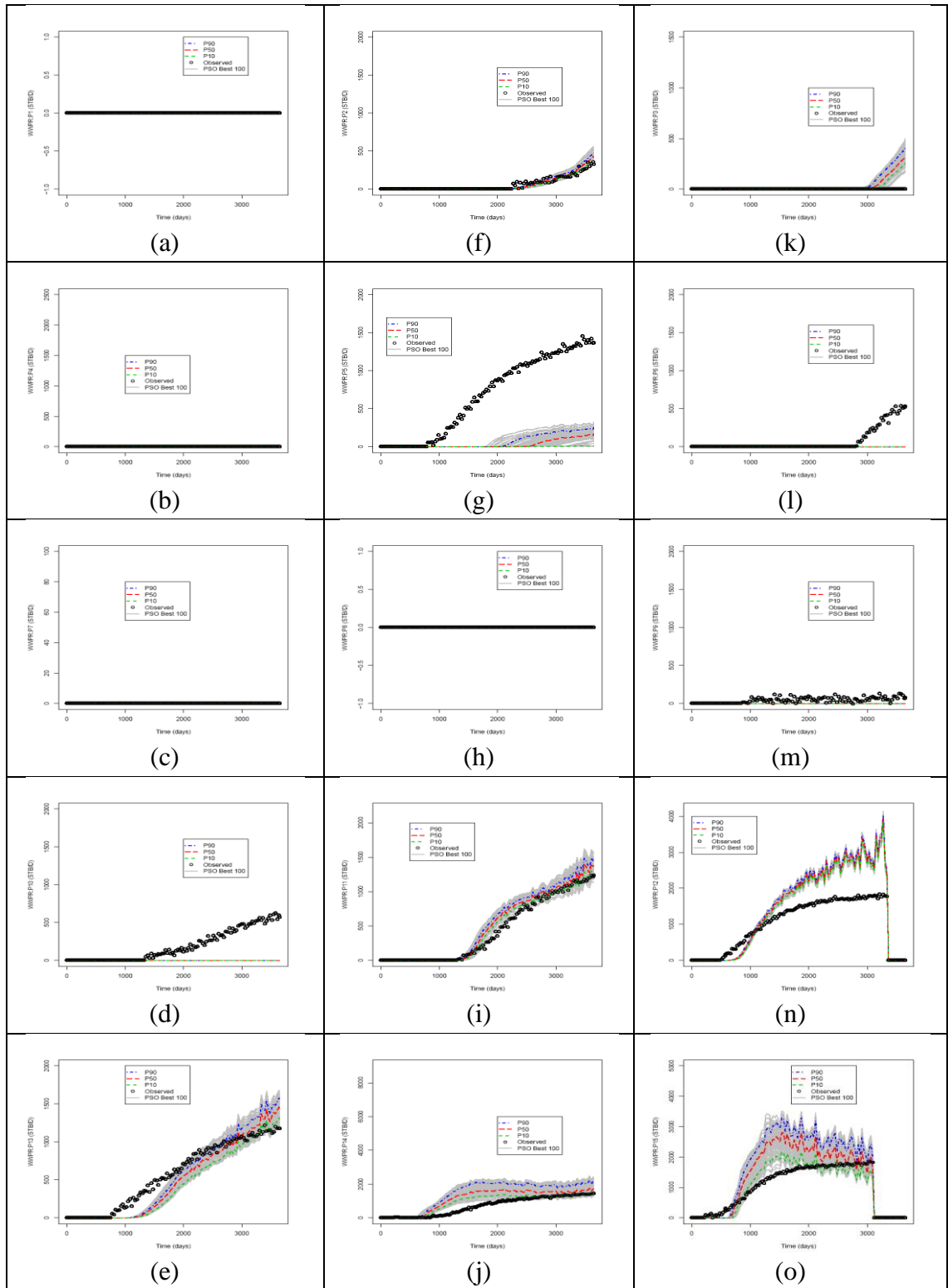


Figure A.8: PSO p10-p50-p90 results

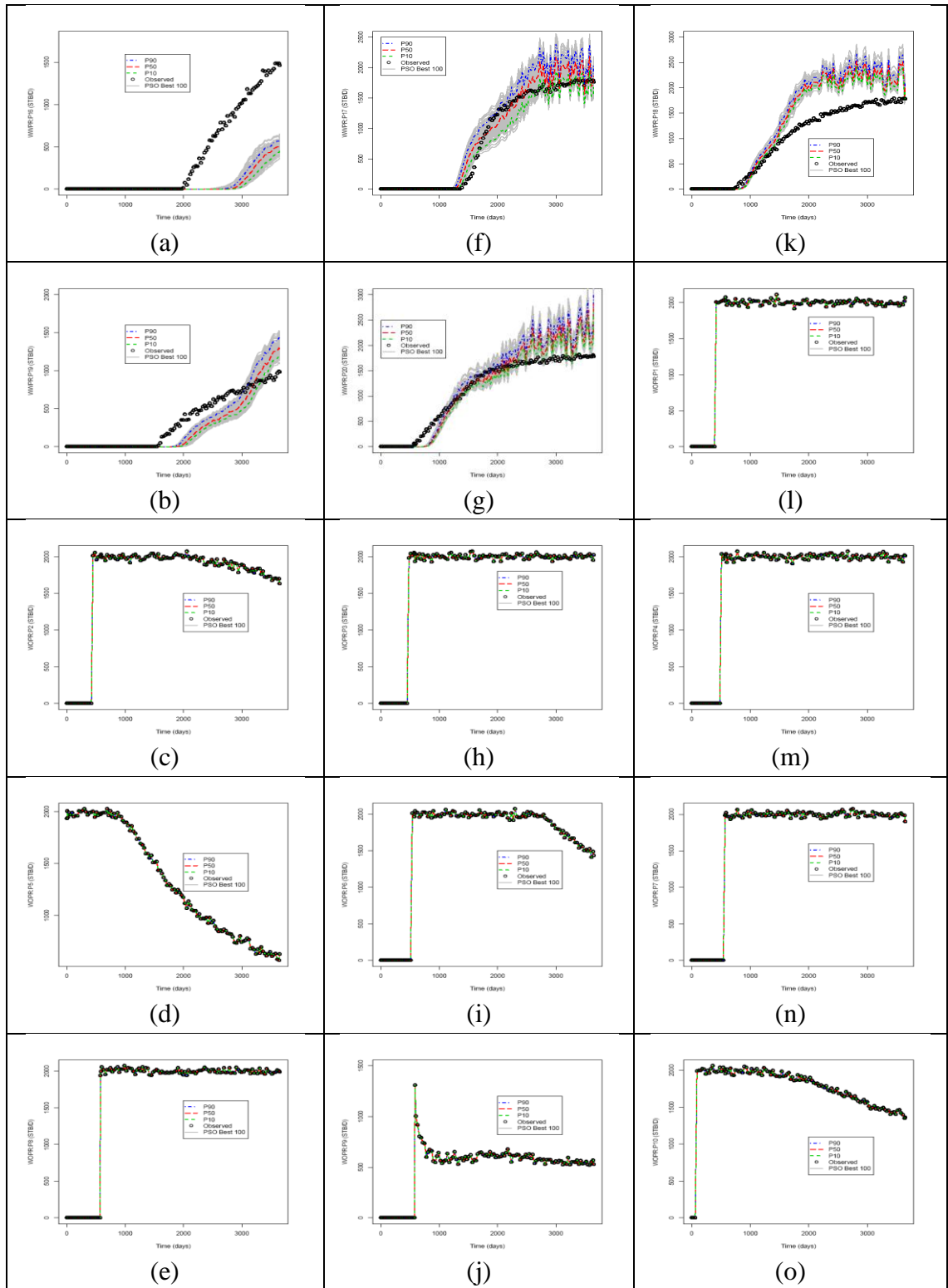


Figure A.9: PSO p10-p50-p90 results

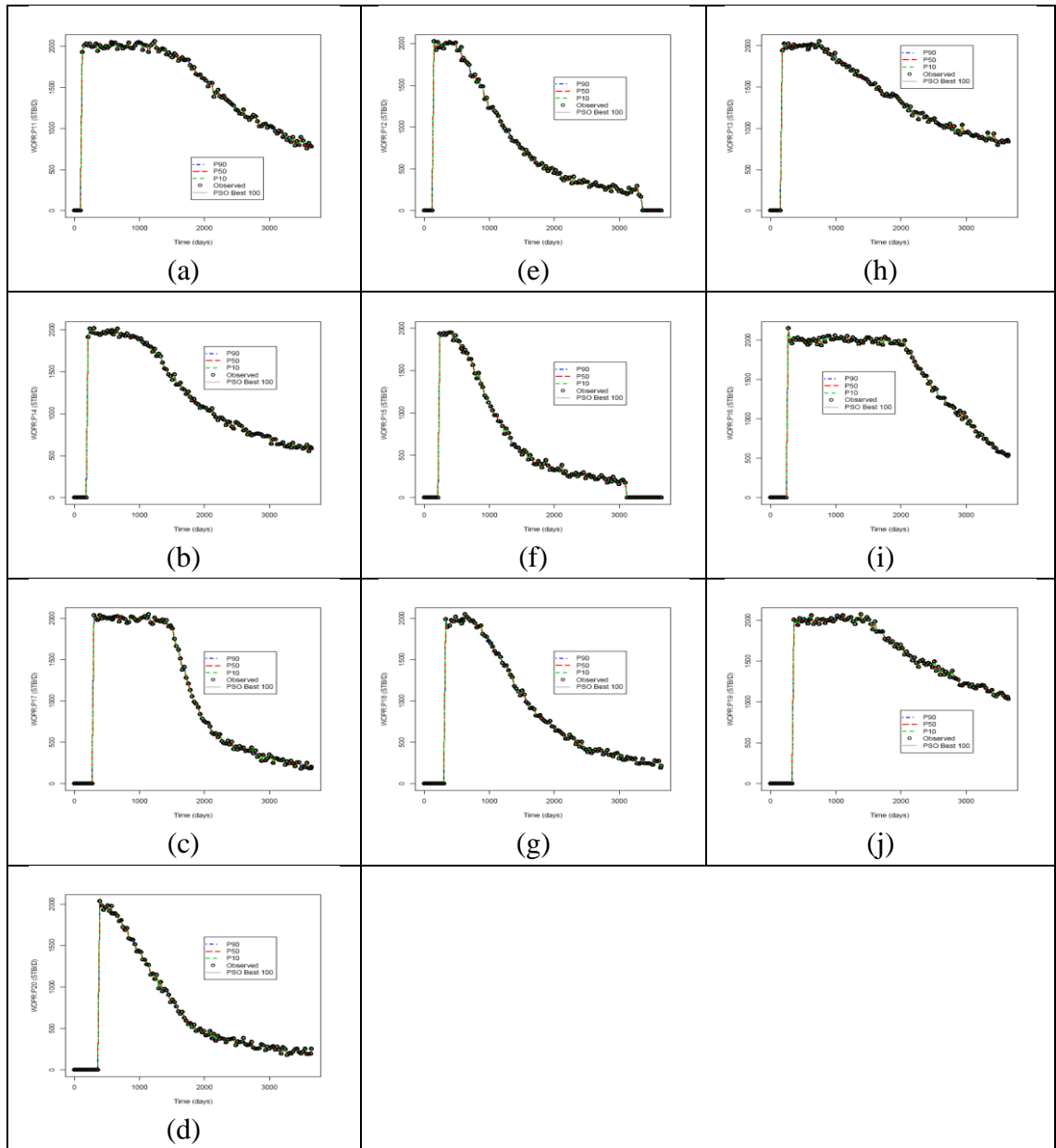


Figure A.10: PSO p10-p50-p90 results

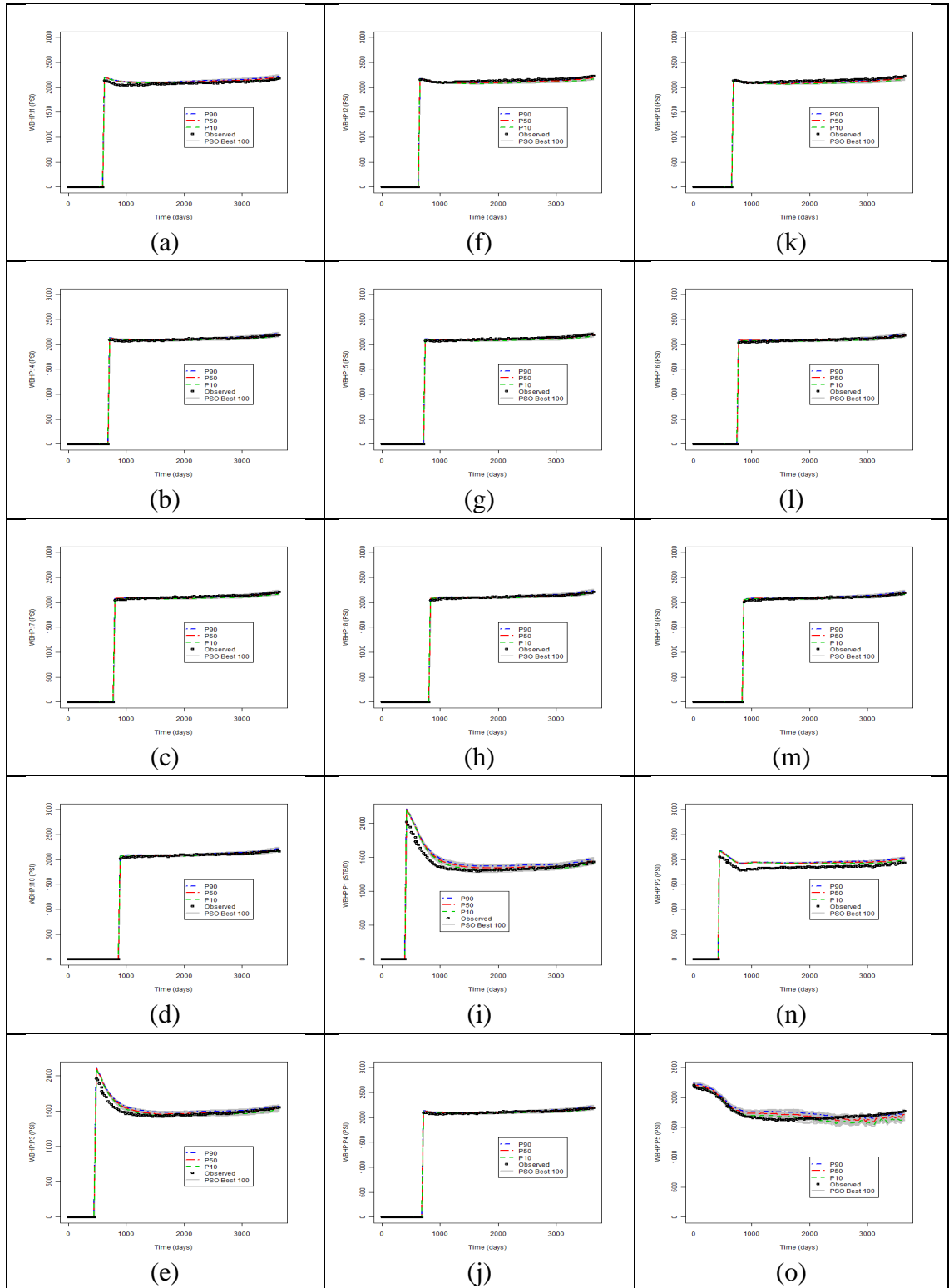


Figure A.11: PSO p10-p50-p90 result – adding OWC parameter

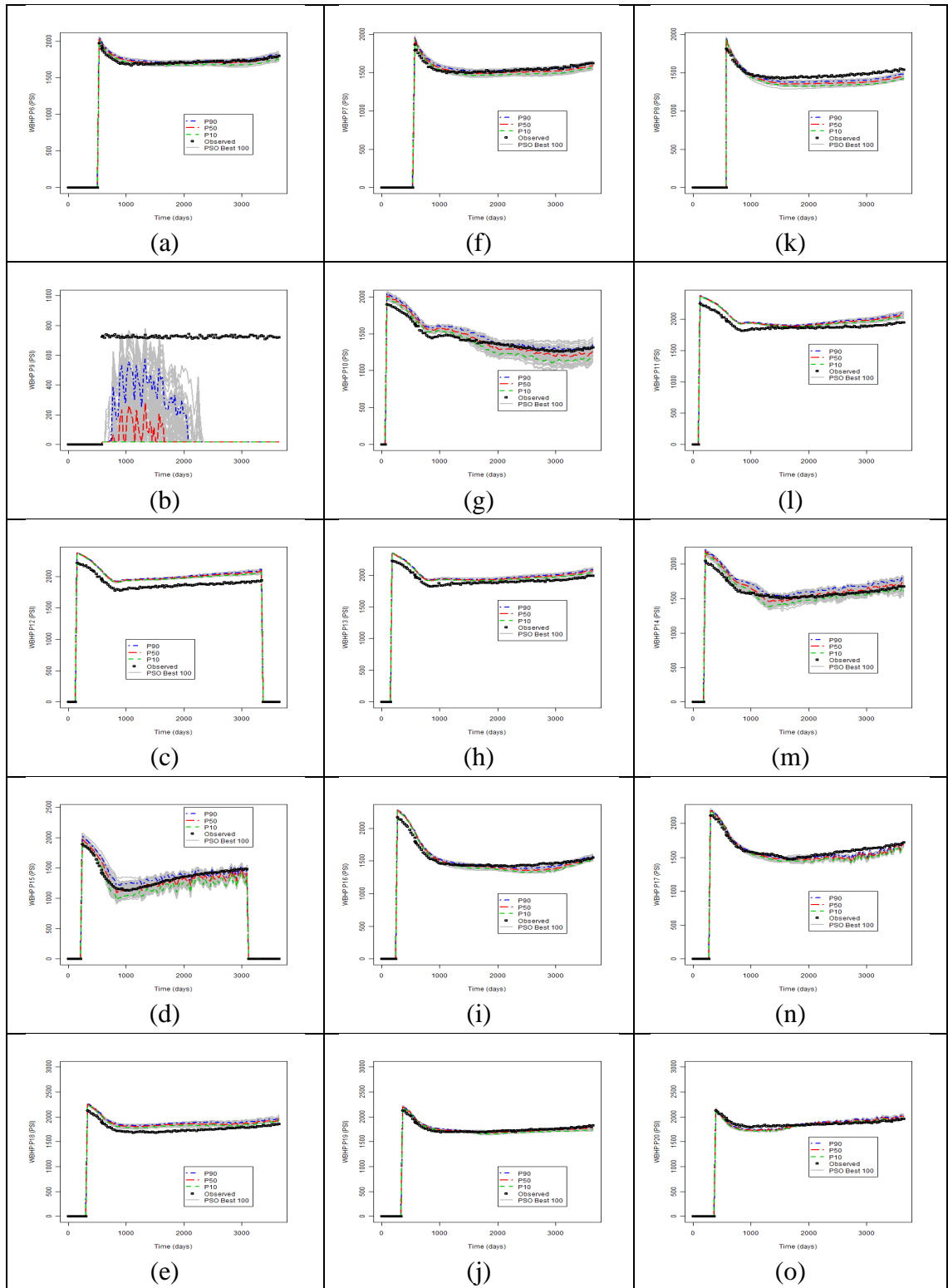


Figure A.12: PSO p10-p50-p90 result – adding OWC parameter

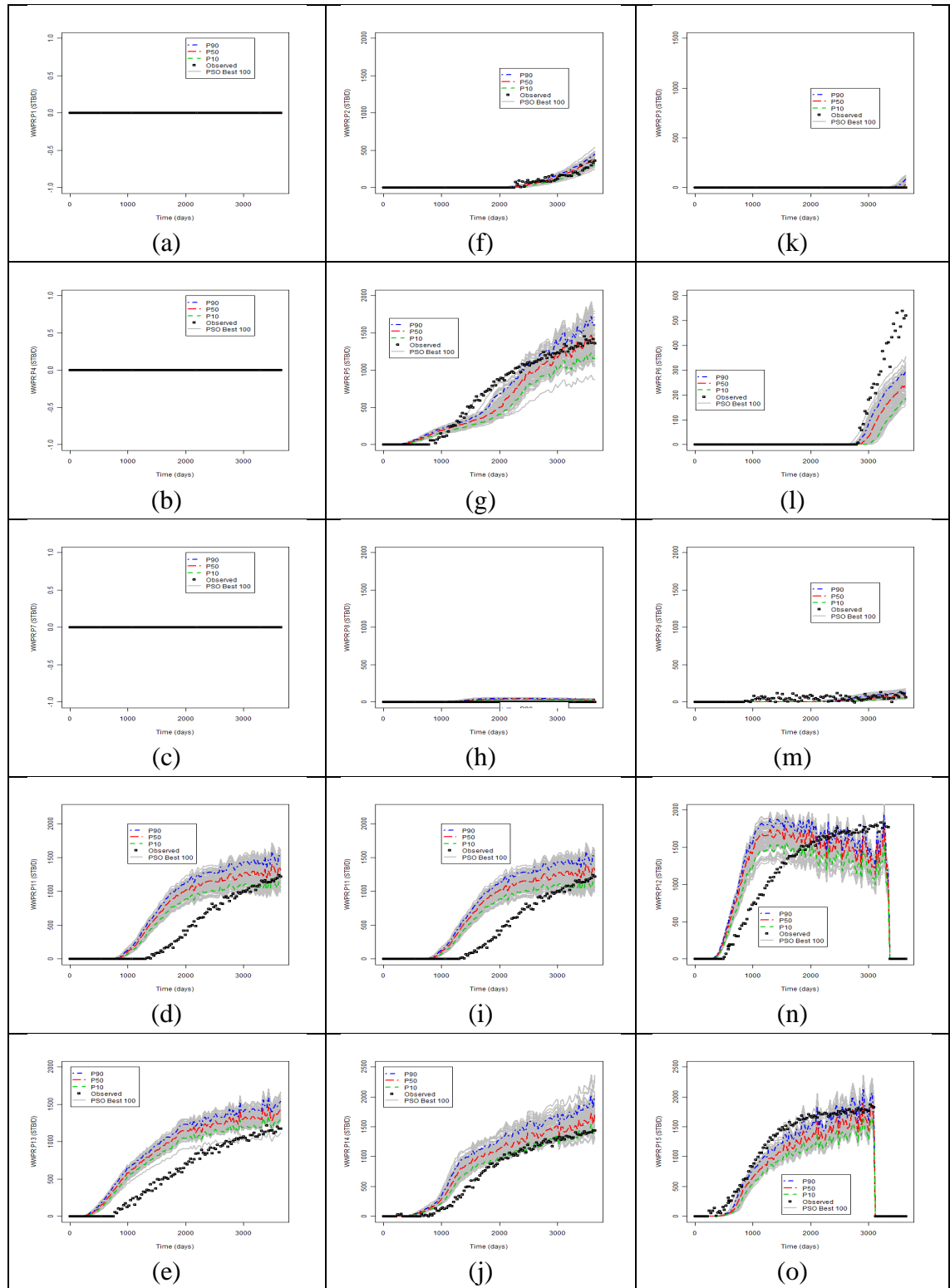


Figure A.13: PSO p10-p50-p90 result – adding OWC parameter

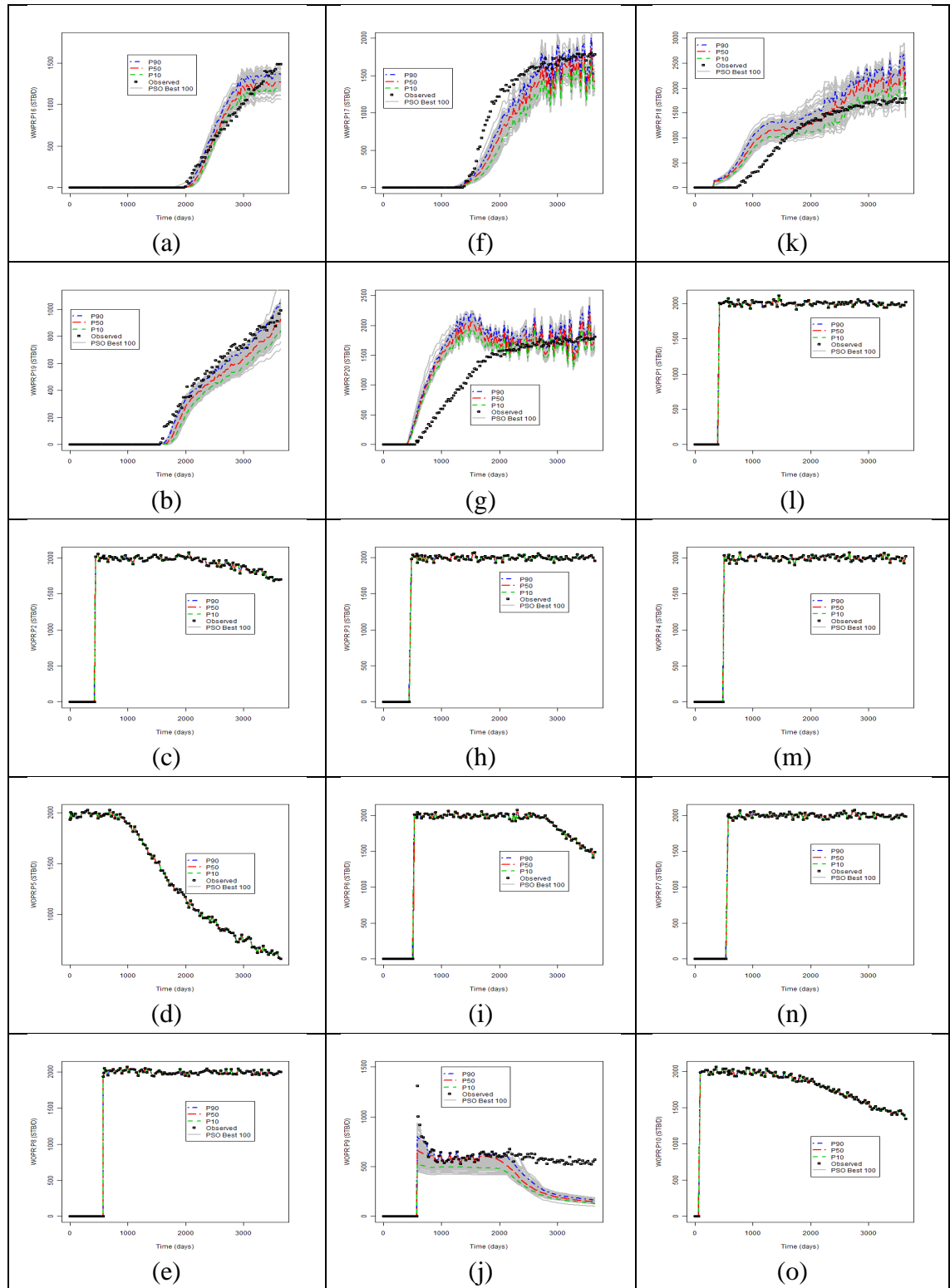


Figure A.14: PSO p10-p50-p90 result – adding OWC parameter

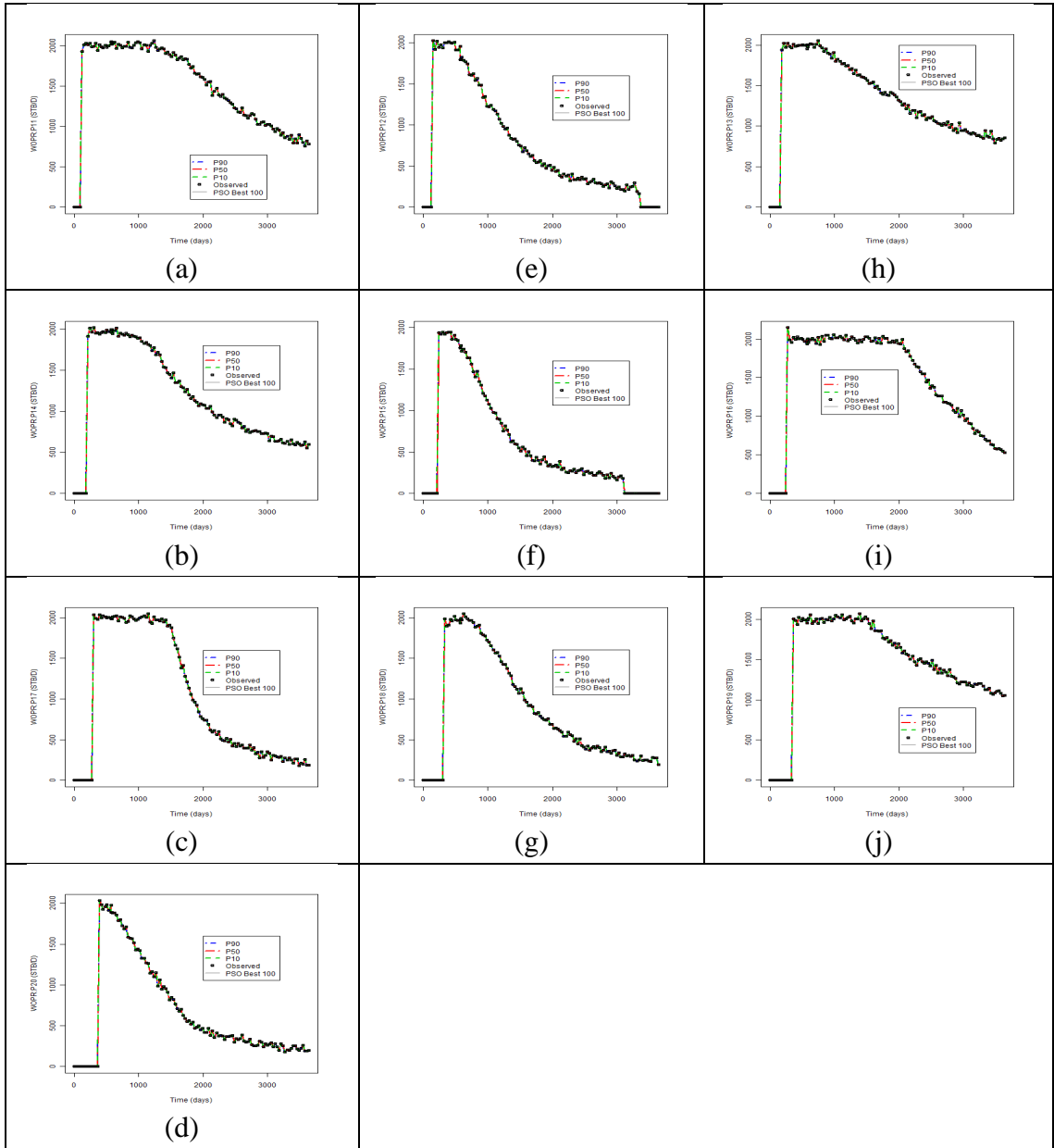


Figure A.15: PSO p10-p50-p90 result – adding OWC parameter

Appendix B – Brugge Reservoir Best Fitting Model Results

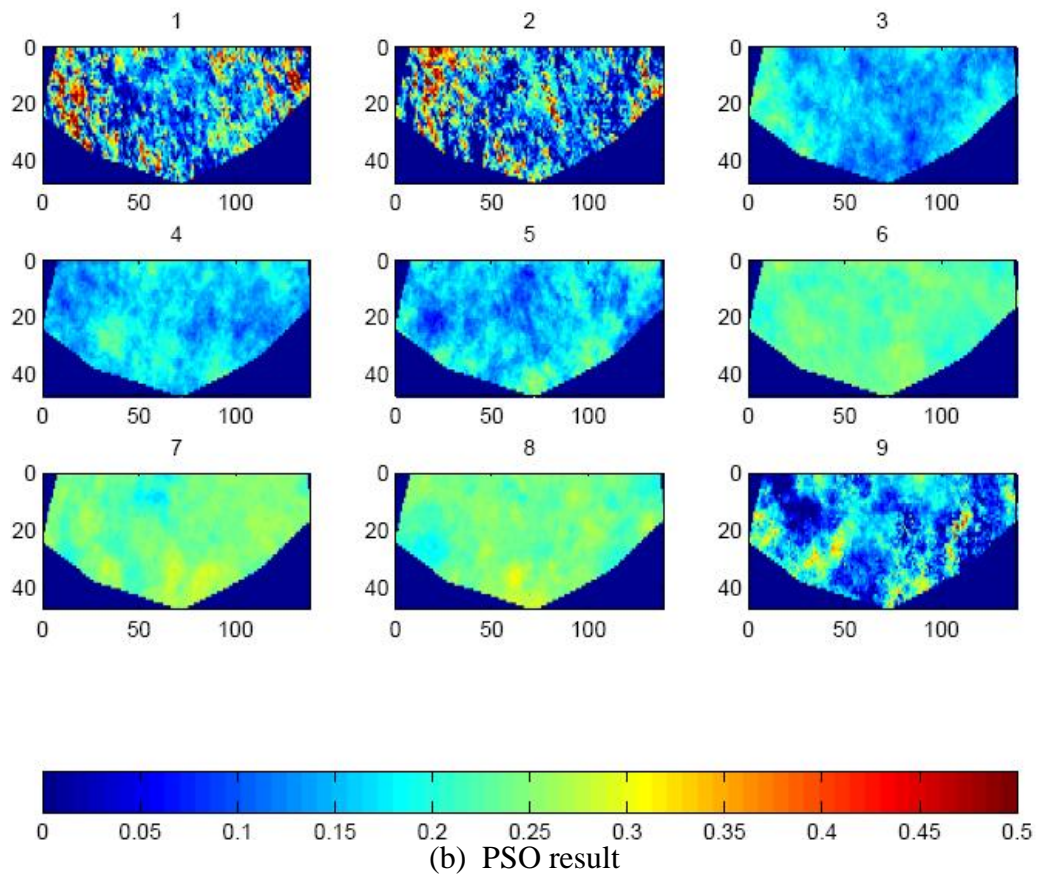
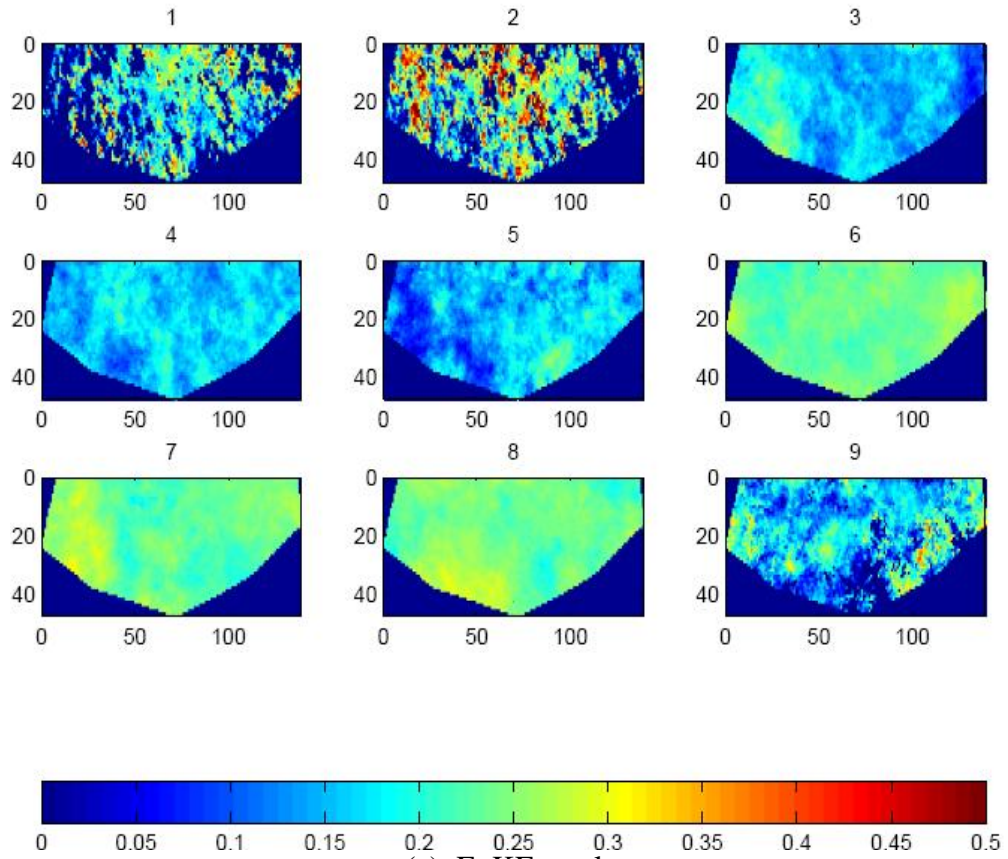


Figure B.1: Porosity for the 9 layers for EnKF (a) and PSO (b)

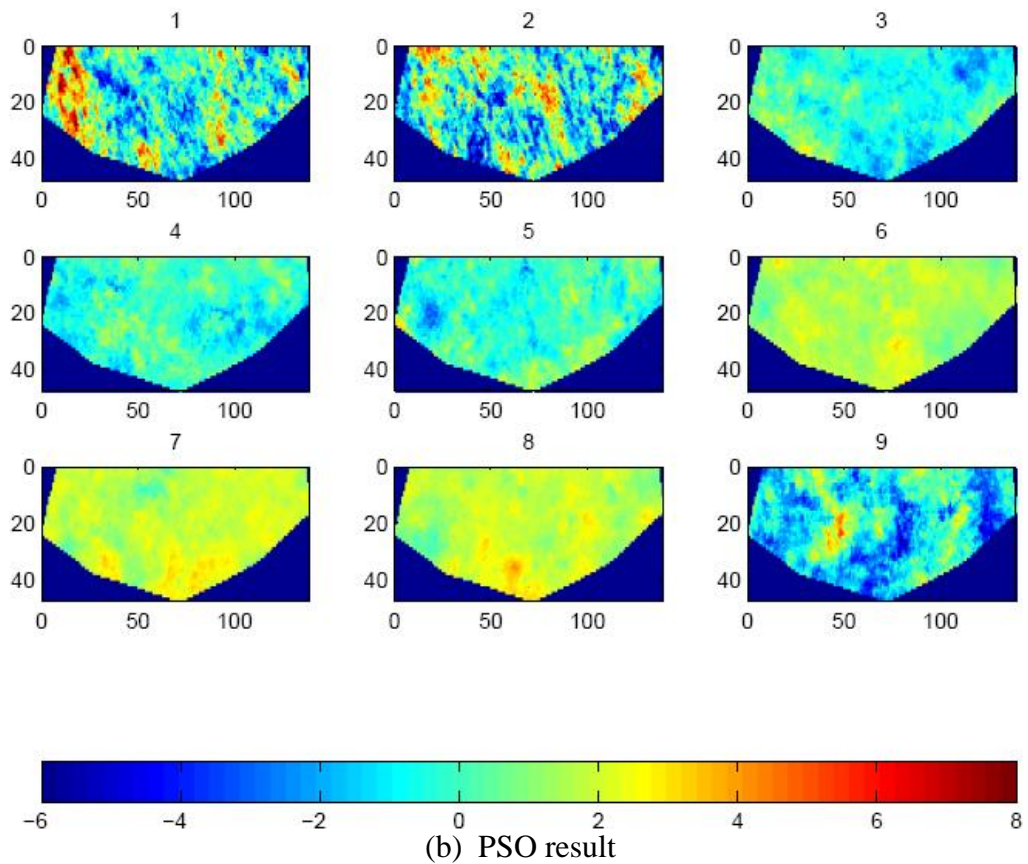
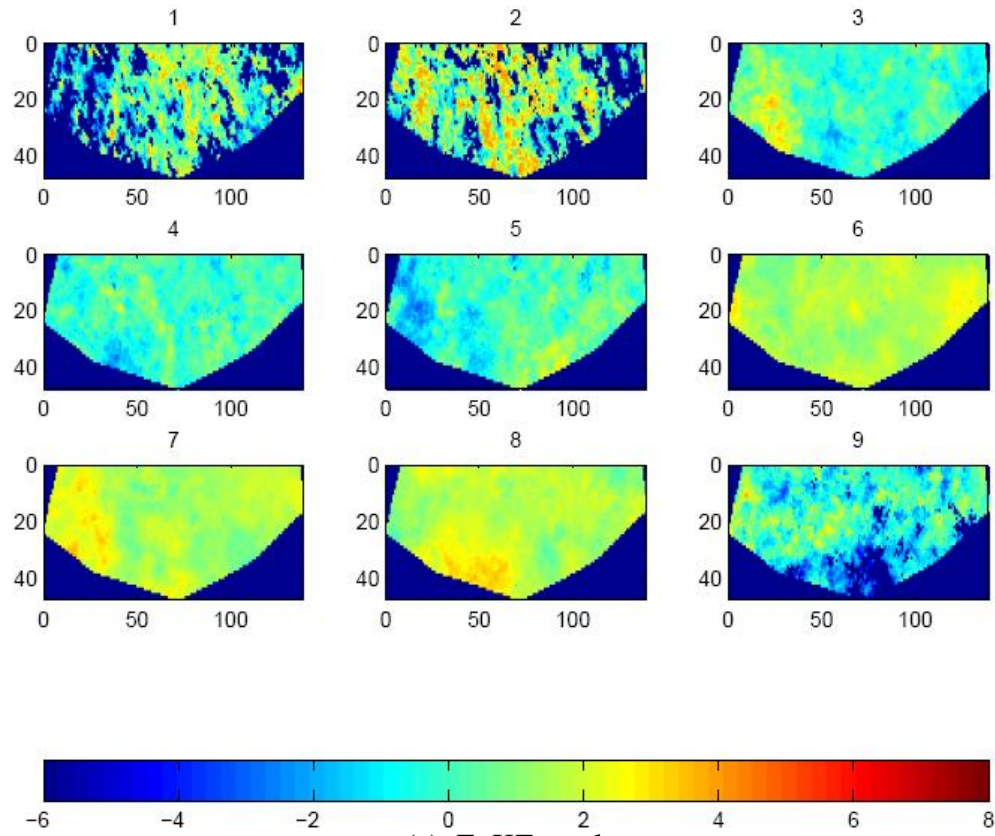


Figure B.2: Natural logarithm of permeability in Z direction for the 9 layers for EnKF (a) and PSO (b)

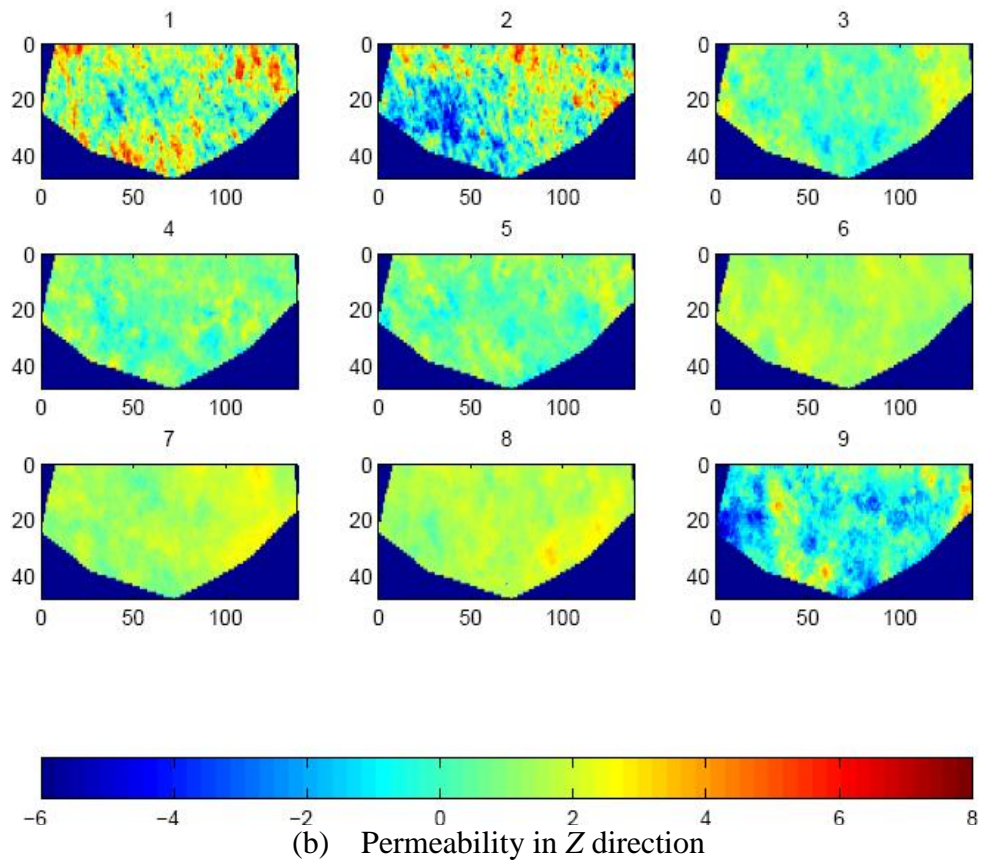
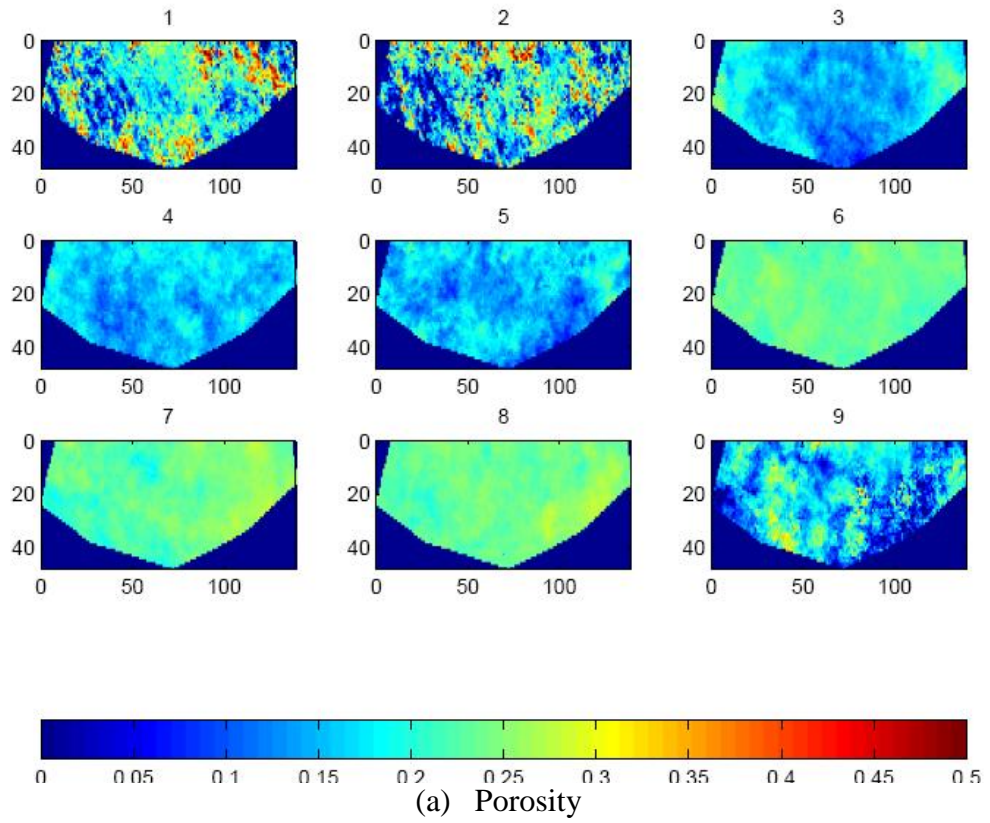


Figure B.3: PSO with OWC addition result for the 9 layers with porosity (a) and natural logarithm of permeability in Z direction (b)

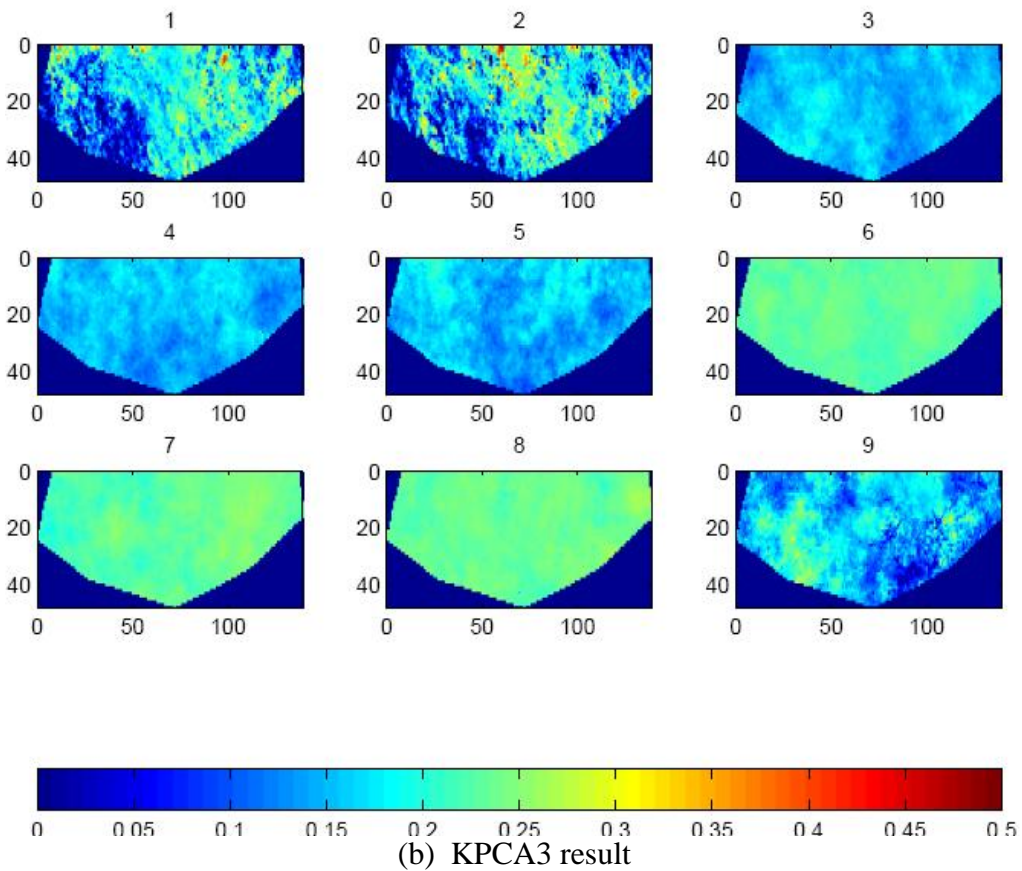
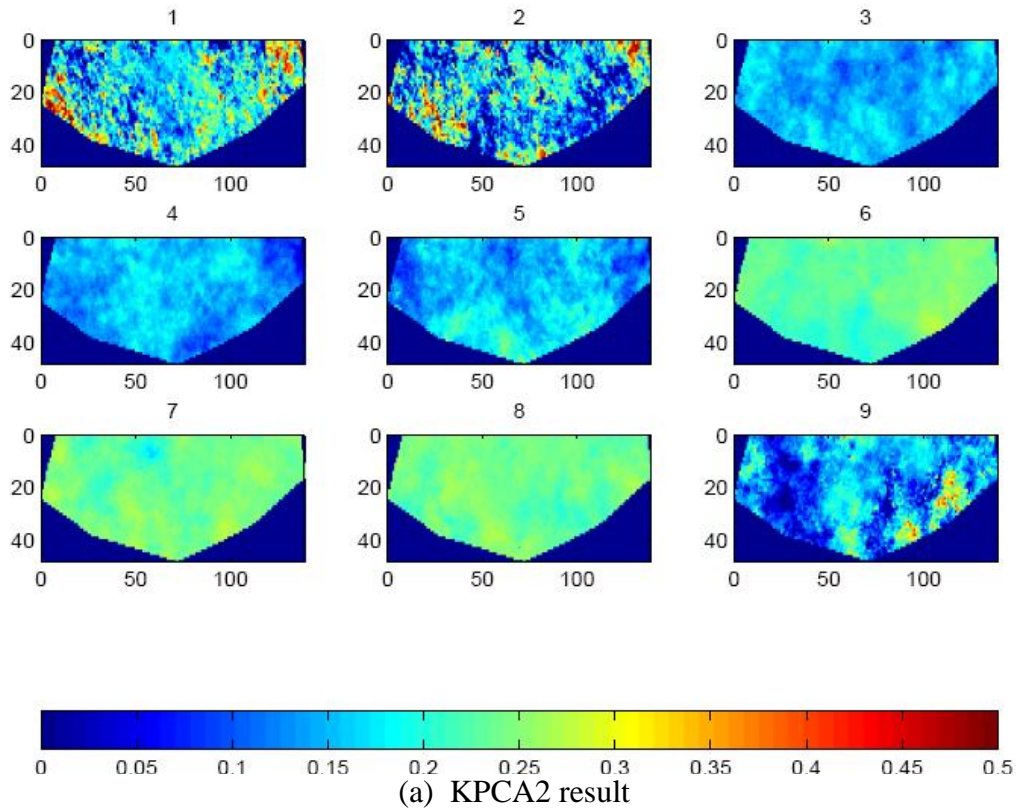


Figure B.4: Porosity for the 9 layers for KPCA2 (a) and KPCA3 (b)

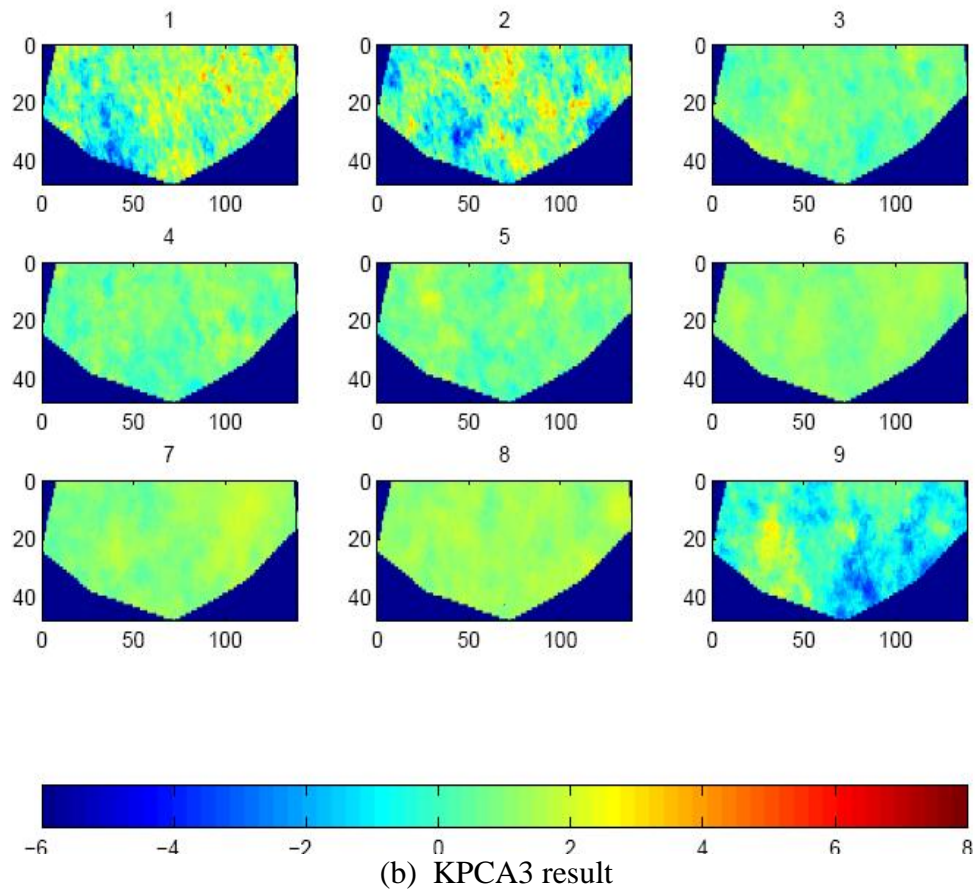
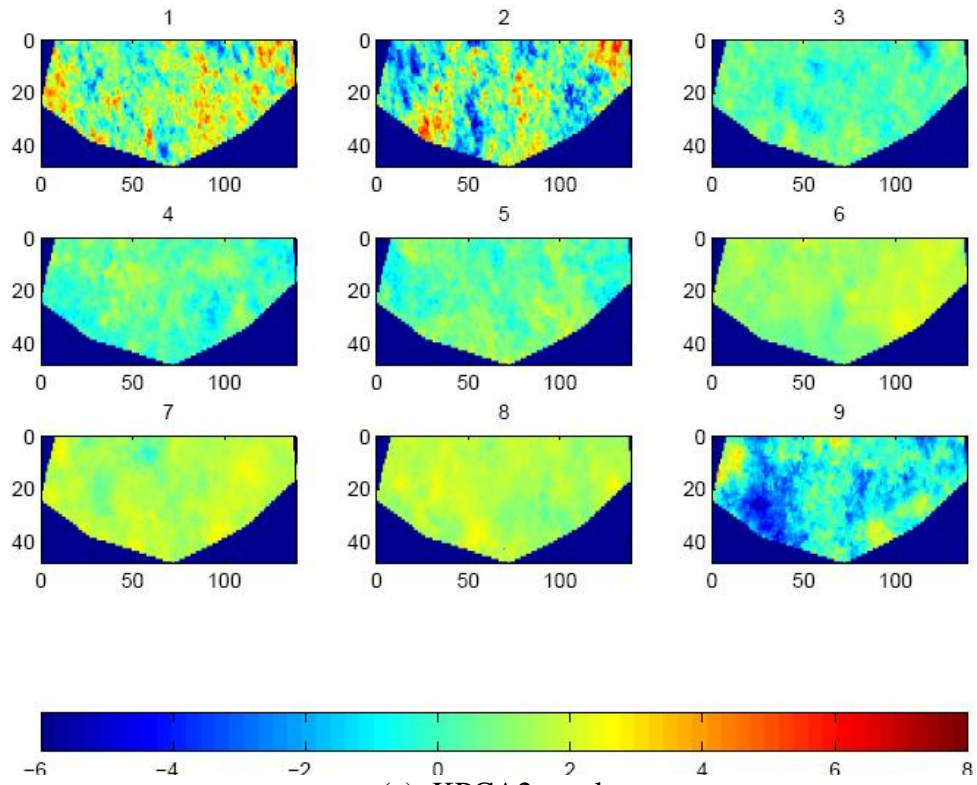


Figure B.5: Natural logarithm of permeability in Z direction for the 9 layers for KPCA1 (a) and KPCA3 (b)

References

- Aanonsen, S., Cominelli, A., Gosselin, O., Aavatsmark, I., and Barkve, T. (2002) Integration of 4D data in the history match loop by investigating scale dependent correlations in acoustic impedance cube. In: Proceedings of the 8th European Conference on the Mathematics of Oil Recovery (ECMOR VIII), E59, 3–6 September, Freiberg, Germany.
- Aanonsen, S., Naevdal, G., Oliver, D., Reynolds, A., and Valles, B. (2009) The ensemble Kalman filter in reservoir engineering – a review. *SPE Journal*, SPE 117274–PA, 14(3), 393–412.
- Abraham, A., Guo, H., and Liu, H. (2006) Swarm intelligence: foundations, perspectives and applications. *Studies in Computational Intelligence*, 26, 3–25.
- Adler, P., and Thovert, J. (1998) Real porous media: Local geometry and macroscopic properties. *Applied Mechanics Reviews*, 51(9), 537–585.
- Akhmatskaya, E., Bou-Rabee, N., and Reich, S. (2009) A comparison of generalised hybrid Monte Carlo methods with and without momentum flip. *Journal of Computational Physics*, 228(6), 2256–2265.
- Akhmatskaya, E., and Reich, S. (2010) New hybrid Monte Carlo methods for efficient sampling: From physics to biology and statistics. In: Proceedings of the Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2010 (SNA + MC2010), 17–21 October, Tokyo, Japan.
- Alanazi, A. (2009) Support vector machines-based porosity prediction model in heterogeneous reservoir. In: Proceedings of the International Association for on Mathematical Geosciences (IAMG) Conference, 23–29 August, Stanford, California, USA.

- Albert, F. (1987) The practice of fast conditional simulations through the LU decomposition of the covariance matrix. *Mathematical Geology*, 19(5), 369–386.
- Al-Dossary, M. (2004) Multidimensional visualisation in reservoir simulation uncertainties. MSc Thesis, Heriot–Watt University, Edinburgh, UK.
- Alexander, F., Eyink, G., and Restrepo, J. (2005) Accelerated Monte Carlo for optimal estimation of time series, *Journal of Statistical Physics*, 119(5–6), 1331–1345.
- Alfaki, M. (2008) Improving efficiency in parameter estimation using the Hamiltonian Monte Carlo algorithm. MSc Thesis, University of Bergen, Bergen, Norway.
- Alvarez-Benitez, J., Everson, R. and Fieldsend, J. (2005) A MOPSO algorithm based exclusively on Pareto dominance concepts. In: Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimisation, EMO 2005, 459–473, Guanajuato, Mexico, LNCS 3410, Springer–Verlag.
- Andrieu, C. (2004) Discussion on the meeting on ‘Statistical approaches to inverse problems’. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(3), 627–652.
- Andrieu, C., and Atchadé, Y. (2005) On the efficiency of adaptive MCMC algorithms. Preprint.
- Andrieu, C., de Freitas, N. Doucet, A., and Jordan, M. (2003) An introduction to MCMC for machine learning. *Machine Learning*, 50(1–2), 5–43.
- Andrieu, C., and Moulines, É. (2003) On the ergodicity properties of some adaptive Markov Chain Monte Carlo algorithms. Preprint.
- Arnold, D. (2008) Geological parameterisation of petroleum reservoir models for improved uncertainty quantification. PhD Thesis, Heriot–Watt University, Edinburgh, UK.
- Arps, J. (1945) Analysis of decline curves. *Trans. AIME*, 160, 228–247.
- Arps, J. (1956) Estimation of primary reserves. *Trans. AIME*, 207, 24–33.
- Atchadé, Y., Fort, G., Moulines, E., and Priouret, P. (2009) Adaptive Markov chain Monte Carlo: Theory and methods. In: *Inference and Learning in Dynamic Models*. Cambridge, UK: Cambridge University Press. Preprint.
- Atchadé, Y., and Rosenthal, J. (2005) On adaptive Markov chain Monte Carlo algorithms. *Bernoulli*, 11(5), 815–828.
- Aziz, K., and Settari, A. (1979) *Petroleum Reservoir Simulation*. London, UK: Applied Science Publishers.

- Babaei, M., and King, P. (2011) A comparison between wavelet and renormalisation upscaling methods and iterative upscaling-downscaling scheme. In: Proceedings of the SPE Reservoir Simulation Symposium, SPE 141418, 21–23 February, The Woodlands, Texas, USA.
- Baker, R., Kuppe, F., Chug, S., Bora, R., Stojanovic, S., and Batycky, R. (2002) Full-field modelling using streamline-based simulation: Four case studies. *SPE Reservoir Evaluation & Engineering*, SPE 77172–PA, 5(2), 126–134.
- Bakke, S., and Øren, P. (1997) 3-D pore-scale modelling of sandstones and flow simulations in the pore networks, *SPE Journal*, SPE 35479–PA, 2(2), 136–149.
- Ballester, P., and Carter, J. (2006) Characterising the parameter space of a highly nonlinear inverse problem. *Inverse Problems in Science and Engineering*, 14(2), 171–191.
- Ballester, P. (2005) New computational methods to address nonlinear inverse problems. PhD Thesis, Imperial College London, London, UK.
- Barker, J., Cuypers, M., and Holden, L. (2001) Quantifying uncertainty in production forecasts: Another look at the PUNQ-S3 problem. *SPE Journal*, SPE 74707–PA, 6(4), 433–441.
- Barros, F., Martins, E., Nascimento, L., and Reis, D. (2008) Calibration of hydrogic models using multi-objective evolutionary algorithms. In: Proceedings of the 13th IWRA World Water Congress, 1–4 September, Montpellier, France.
- Barros, F., Martins, E., Nascimento, L., and Reis, D. (2010) Use of multi-objective evolutionary algorithms in water resources engineering. In: N. Nedjah et al. (Eds.), *Multi-Objective Swarm Intelligent Systems*. SCI 261, (pp. 45–82). Berlin, Germany: Springer–Verlag Berlin Heidelberg.
- Bartz-Beielstein, T., Limbourg, P., Mehnen, J., Schmitt, K., Parsopoulos, K., and Vrahatis, M. (2003) Particle swarm optimisers for Pareto optimisation with enhanced archiving techniques. In: Proceedings of the Congress on Evolutionary Computation (CEC’2003), 3, 1780–1787, Canberra, Australia, December, IEEE Press.
- Batycky, R., Blunt, M., and Thiele, M. (1997) A 3D field-scale streamline-based reservoir simulator. *SPE Reservoir Engineering*, SPE 36726–PA, 12(4), 246–254.
- Baumgartner, U., Magele, C., and Renhart, W. (2004) Pareto optimality and particle swarm optimisation. *IEEE Transactions on Magnetics*, 40(2), 1172–1175.

- Bayes, T., and Price, R. (1763) An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53, 370–418.
- Beichl, I., and Sullivan, F. (2000) The Metropolis algorithm. *Computing in Science & Engineering*, 2(1), 65–69.
- Bernardo, J., Smith, A. (1994) *Bayesian Theory*. New York, USA: John Wiley & Sons.
- Besag, J. (1986) On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society B*, 48(3), 259–302.
- Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J., and Stuart, A. (2010) Optimal tuning of the hybrid Monte Carlo algorithm. Technical Report. Department of Statistical Science, University of College London, London.
- Beucher, H., and Renard, D. (2005) Reservoir characterisation. Technical Report N-03/05/G, the Geosciences department, Mines ParisTech Graduate School, http://cg.ensmp.fr/bibliotheque/public/BEUCHER_Rapport_00318.pdf (accessed 24 December 2010).
- Birge, B. (2003) PSOt – a particle swarm optimisation toolbox for use with MATLAB. In: Proceedings of the 2003 IEEE, Swarm Intelligence Symposium, 182–186.
- Bischof and Bücker (2000) Computing derivatives of computer programs. In: J. Grotendorst (Ed.): *Modern Methods and Algorithms of Quantum Chemistry*, 1, (pp. 287–299), John von Neumann Institute for Computing: Jülich, NIC Series.
- Bishop, C. (2006) *Pattern Recognition and Machine Learning*. New York, USA: Springer.
- Bissell, R., Sharma, Y., and Killough, J. (1994) History matching using the method of gradients: Two case studies. In: Proceedings of the SPE Annual Technical Conference and Exhibition, SPE 28590, 275–282, New Orleans, Louisiana, USA.
- Blunt, M., Jackson, M., Piri, M., and Valvatne, P. (2002) Detailed physics, predictive capabilities and macroscopic consequences for pore-network models of multiphase flow. *Advances in Water Resources*, 25(8–12), 1069–1089.
- Blunt, M., Liu, K., and Thiele, M. (1996) A generalised streamline method to predict reservoir flow. *Petroleum Geoscience*, 2(3), 259–269.
- Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999) *Swarm Intelligence: From Natural to Artificial Systems*. New York, USA: Oxford University Press.

- Bonet-Cunha, L., Oliver, D., Redner, R., and Reynolds, A. (1998) A hybrid Markov chain Monte Carlo method for generating permeability fields conditioned to multiwell pressure data and prior information. *SPE Journal*, SPE 50991-PA, 3(3), 261–271.
- Bos, C. (2000) Production forecasting with uncertainty quantification. Final report of EC project, NITG-TNO report NITG 99-255-A, January.
- Box, G., and Jenkins, G. (1976) *Time Series Analysis: Forecasting and Control*. Oakland, California: Holden-Day.
- Bryant, S. and Blunt, M. (1992) Prediction of relative permeability in simple porous media. *Physical Review A*, 46(4), 2004–2011.
- Bryant, S., King, P., and Mellor, D. (1993a) Network model evaluation of permeability and spatial correlation in a real random sphere packing. *Transport in Porous Media*, 11(1), 53–70.
- Bryant, S., Mellor, D., and Cade, C. (1993b) Physically representative network models of transport in porous media. *AIChE Journal*, 39(3), 387–396.
- Busby, D. (2009) Hierarchical adaptive experimental design for Gaussian process emulators. *Reliability Engineering and System Safety*, 94(7), 1183–1193.
- Busby, D., Farmer C., and Iske, A. (2007a) Hierarchical nonlinear approximation for experimental design and statistical data fitting. *SIAM Journal on Scientific Computing*. 29(1), 49–69.
- Busby, D., Farmer C., and Iske, A. (2007b) Uncertainty Evaluation in Reservoir Forecasting by Bayes Linear Methodology. In A. Iske, and J. Levesley, (Eds.), *Algorithms for Approximation: Proceedings of the 5th International Conference XIV* (pp. 187–196). Chester, UK: Springer-Verlag Berlin Heidelberg.
- Busby, D., and Feraille, M. (2008) Adaptive design of experiments for calibration of complex simulators – An application to uncertainty quantification of a mature oil field. *Journal of Physics: Conference Series*, 135(1), 012026.
- Busby, D., and Sergienko, E. (2010) Combining probabilistic inversion and multi-objective optimisation for production development under uncertainty. In: Proceedings of 12th European Conference on the Mathematics of Oil Recovery, B018, 6–9 September, Oxford, UK.
- Bush, M., and Carter, J. (1996) Applications of a modified genetic algorithm to estimation in the petroleum industry. In: Proceedings of the 6th Conference on Artificial Neural Networks in Engineering.

- Caers, J. (2003) History matching under training-image based geological model constraints. *SPE Journal*, SPE 74716-PA, 8(3), 218–226.
- Caers, J. (2004) The probability perturbation method – an alternative to a traditional Bayesian approach for solving inverse problems. In: Proceedings of the 9th European Conference on the Mathematics of Oil Recovery (ECMOR IX), 30 August–02 September, Cannes, France.
- Caers, J. (2005) *Petroleum Geostatistics*. TX, USA: Society of Petroleum Engineers.
- Caers, J. (2008) Distance-based random field models and their applications. In: Proceedings of the 8th International Geostatistical Congress, Santiago, Chile.
- Caers, J., Scheidt, C., and Park, K. (2010) Modelling uncertainty of complex Earth systems in metric space. *Handbook of Geomathematics*, in press.
- Caers, J., Srinivasan, S., and Journel, A. (1998) Stochastic reservoir simulation using neural networks trained on outcrop data. In: Proceedings of the SPE Annual Technical Conference and Exhibition, SPE 49026, 27–30 September, New Orleans, Louisiana.
- Caers, J., Strebelle, S., and Payrazyan, K. (2003) Stochastic integration of seismic data and geologic scenarios: A West Africa submarine channel saga. *The Leading Edge*, 22(3), 192–196.
- Calderhead, B., and Girolami, M. (2007) System identification of Enzymatic control processes using Population Monte Carlo methods. In: Proceedings of the PESB Workshop, 17th Annual Mathematical and Statistical Aspects of Molecular Biology (MASAMB) Conference, 29–30 March, Manchester, UK.
- Calderhead, B., and Girolami, M. (2009) Estimating Bayes factors via thermodynamic integration and population MCMC. *Computational Statistics and Data Analysis Archive*, 53(12), 4028–4045.
- Cappé, O., Guillin, A., Marin, J. M., and Robert, C. (2004) Population Monte Carlo. *Journal of Computational and Graphical Statistics*, 13(4), 907–929.
- Carlisle, A., and Dozier, G. (2001) An off-the-shelf PSO. In: Proceedings of the Workshop on Particle Swarm Optimisation, 1, 1–6, Indianapolis.
- Carter, J. (2004) Using Bayesian statistics to capture the effects of modelling errors in inverse problems. *Mathematical Geology*, 36(2), 187–216.
- Carter, J., and Ballester, P. (2004) A real parameter genetic algorithm for cluster identification in history matching. In: Proceedings of the 9th European

- Conference on the Mathematics of Oil Recovery (ECMOR IX), 30 August–02 September, Cannes, France.
- Carter, C., Ballester, P., Tavassoli, Z., and King, P. (2006) Our calibrated model has poor predictive value: An example from the petroleum industry. *Reliability Engineering and System Safety*, 91(10–11), 1373–1381.
- Casella, G., and George, E. (1992) Explaining the Gibbs sampler. *The American Statistician*, 46(3), 167–174.
- Castro, S. (2007) A probabilistic approach to jointly integrate 3D/4D seismic, production data and geological information for building reservoir models, PhD Thesis, Stanford University, California, USA.
- Castro, S., Caers, J., Mukerji, T. (2005) The Stanford VI reservoir. Technical Report 18th Annual Technical Report (May), Stanford Centre for Reservoir Forecasting (SCRF), Stanford University, California, USA.
- Černý, V. (1985) A thermodynamical approach to the travelling salesman problem: An efficient simulation algorithm. *Journal of Optimisation Theory and Applications*, 45(1), 41–51.
- Chang, B., and Ratnaweera, A., and Halgamuge, S. (2004) Particle swarm optimisation for protein motif discovery. *Genetic Programming and Evolvable Machines*, 5(2), 203–214.
- Chavent, G., Dupuy, M., and Lemonnier, P. (1973) History-matching by use of optimal control theory. In: Proceedings of the Annual Technical Conference and Exhibition (ATCE), SPE 4627, Las Vegas, Nev., USA.
- Chen, C., Wang, Y., Li, G., and Reynolds, A. (2010) Closed-loop reservoir management on the Brugge test case. *Computational Geosciences*, 14(4), 691–703.
- Chen, L., Qin, Z. and Liu, J. (2000) Exploring hybrid Monte Carlo in Bayesian computation. In: Proceedings of the 6th World Meeting of the International Society for Bayesian Analysis (ISBA), 28 May–1 June, Hersonissos, Heraklion, Crete, Retrieved from <http://www-personal.umich.edu/~qin/hmc.pdf> (accessed 30 December 2010).
- Cheng, H. (2010) Uncertainty quantification and uncertainty reduction techniques for scientific simulations. Tutorial, Computer Science Department, Willamette University, Salem, Oregon, USA.

- Chen, Y., and Oliver, D. (2010) Ensemble-based closed-loop optimisation applied to Brugge field. *SPE Reservoir Evaluation & Engineering*, SPE 118926-PA, 13(1), 56–71.
- Cheung, S., and Beck, J. (2009) Bayesian model updating using hybrid Monte Carlo simulation with application to structural dynamic models with many uncertain parameters. *Journal of Engineering Mechanics*, 135(4), 243–255.
- Chilès, J., and Delfiner, P. (1999) *Geostatistics – Modelling Spatial Uncertainty* (first edition). Wiley series in probability and statistics, New York, USA: Wiley – Interscience.
- Choo, K. (2000) Learning hyperparameters for neural network models using Hamiltonian dynamics. Master Thesis. University of Toronto, Toronto, Canada.
- Christie, M. (1996) Upscaling for reservoir simulation. *Journal of Petroleum Technology*, SPE 37324, 48(11), 1004–1010.
- Christie, M. (2001) Flow in porous media – scale up of multiphase flow. *Current Opinion in Colloid and Interface Science*, 6(3), 236–241.
- Christie, M. (2011) Uncertainty quantification and oil reservoir modelling. In M. Christie, A. Cliffe, P. Dawid, and S. Senn (Eds.), *Simplicity, Complexity and Modelling*. England, UK: John Wiley & Sons.
- Christie, M., and Blunt, M. (2001) Tenth SPE comparative solution project: A comparison of upscaling techniques. *SPE Reservoir Evaluation & Engineering*, SPE 72469-PA, 308–317.
- Christie, M., Cliffe, A., Dawid, P., and Senn, S. (2011) (Eds.) *Simplicity, Complexity and Modelling*. England, UK: John Wiley & Sons.
- Christie, M., Demyanov, V., and Erbas, D. (2006) Uncertainty quantification for porous media flows. *Journal of Computational Physics*, 217(1), 143–158.
- Christie, M., Glimm, J., Grove, J., Higdon, D., Sharp, D., and Wood-Schultz, M. (2005) Error analysis and simulations of complex phenomena. *Los Alamos Science*, 29, 6–25.
- Christie, M., MacBeth, C., and Subbey, S. (2002) Multiple history-matched models for Teal South. *The Leading Edge*, 21(3), 286–289.
- Christie, M., Subbey, S., and Sambridge, M. (2002) Prediction under uncertainty in reservoir modelling. In: Proceedings of the 8th European Conference on the

- Mathematics of Oil Recovery (ECMOR VIII), E053, 3–6 September, Freiberg, Germany.
- Christie, M., Pickup, G., O’Sullivan, A., and Demyanov, V. (2008) Use of solution error models in history matching. In: Proceedings of the 11th European Conference on the Mathematics of Oil Recovery, (ECMOR XI), B19, 8–11 September, Bergen, Norway.
- Ciaurri, D., Isebor, O., and Durlofsky, L. (2010) New approaches for generally constrained production optimisation with an emphasis on derivative-free techniques. In: Proceedings of 12th European Conference on the Mathematics of Oil Recovery, B029, 6–9 September Oxford, UK.
- Cipra, B. (2000) The best of the 20th century: Editors name top 10 algorithms. *SIAM News*, 33(4), 1–2.
- Clerc, M. (2006) Particle Swarm Optimisation, London, UK: Wiley-Blackwell ISTE Publishing Company.
- Clyde, M., and George, E. (2003) Model uncertainty. Technical Report #2003-16, December, Statistical and Applied Mathematical Sciences Institute (SAMSI), North Carolina, USA.
- Coello, C., Pulido, G., and Salazar-Lechuga, M. (2004) Handling multiple objectives with particle swarm optimisation. *IEEE Transactions on Evolutionary Computation*, 8(3), 256–279.
- Coello, C., and Salazar-Lechuga, M. (2002) MOPSO: A proposal for multiple objective particle swarm optimisation. In: Proceedings of the Congress on Evolutionary Computation (CEC’2002), 2, 1051–1056, Piscataway, New Jersey, May, IEEE Service Centre.
- Cowles, M., and Carlin, B. (1996) Markov chain Monte Carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association*, 91(434), 883–904.
- Craft, B., Hawkins, M., and Terry, R. (1991) *Applied Petroleum Engineering* (second edition). Englewood Cliffs, NJ: Prentice Hall.
- Crichlow, H. (1977) *Modern Reservoir Engineering – A Simulation Approach*. Englewood Cliffs, NJ: Prentice Hall.
- Cuypers, M., Dubrule, O., Lamy, P., and Bissell, R. (1998) Optimal choice of inversion parameters for history matching with the pilot point method. In: Proceedings of

- the 6th European Conference on the Mathematics of Oil Recovery (ECMOR VI), B028, 8–11 September, Peebles, Scotland.
- Dake, L. (1978) *Fundamentals of Reservoir Engineering*. Amsterdam: Elsevier.
- Daly, C., and Caers, J. (2010) Multipoint statistics – An introductory review. *First Break*, 28(9), 39–47.
- Darcy, H. (1856) *Les fontaines publiques de la ville de Dijon*. Paris, France: Victor Dalmont.
- Das, I., and Dennis, J. (1997) A closer look at drawbacks of minimising weighted sums of objectives for Pareto set generation in multicriteria optimisation problems. *Structural Optimisation*, 14(1), 63–69.
- Datta-Gupta, A. (2000) Streamline simulation: A technology update. *Journal of Petroleum Technology*, SPE 65604, 52(12), 68–73.
- Datta-Gupta, A., and King, M. (2007) *Streamline Simulation: Theory and Practice*. Richardson, Texas: Society of Petroleum Engineers.
- Datta-Gupta, A., Lake, L., and Pope, G. (1995) Characterising heterogeneous permeable media with spatial statistics and tracer data using sequential simulated annealing. *Mathematical Geology*, 27(6), 763–787.
- Davis, M. (1987) Production of conditional simulations via the LU decomposition of the covariance matrix. *Mathematical Geology*, 19(2), 91–98.
- De Almeida, M., Hamacher, S., Pacheco, M., Vellasco, M. (2001) The energy minimisation method: a multi-objective fitness evaluation technique and its application to the production scheduling in a petroleum refinery. In: Proceedings of the 2001 Congress on Evolutionary Computation, 1, 560–567, Seoul, Korea.
- Deb, K. (2009) *Multi-objective Optimisation Using Evolutionary Algorithms* (reprinted version). Chichester, UK: John Wiley & Sons.
- Deb, K., and Goldberg, D. (1989) An investigation of niche and species formation in genetic function optimisation. In J. Schaffer (Ed.), *Proceedings of the 3rd International Conference on Genetic Algorithms*, pp. 42–50, San Mateo, California, June. George Mason University, Morgan Kaufmann Publishers.
- Deb, K. (1999) Multi-objective genetic algorithms: problem difficulties and construction of test problems. *Evolutionary Computing*, 7, 205–230.
- Deb, K., Agrawal, S., Pratab, A., and Meyarivan, T. (2000) A fast elitist nondominated sorting genetic algorithm for multi-objective optimisation: NSGA-II. In:

- Proceedings of Parallel Problem Solving From Nature VI* (pp. 849–858): Lecture Notes in Computer Science.
- Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2005). Scalable test problems for evolutionary multi-objective optimisation. *Evolutionary Multi-objective Optimisation*, Springer, 105–145.
- Deb, K., Thiele L., and Laumanns, M. and Zitzler, E. (2002) Scalable multi-objective optimisation test problems. In: Proceedings of the Congress on Evolutionary Computation (CEC'2002), 825–830.
- Del Moral, P., and Doucet, A. (2003) On a class of genealogical and interacting Metropolis models. In J. Azéma, M. Emery, Ledoux, and M. Yor (Eds.), Séminaire de Probabilités XXXVI, Lecture Notes in Mathematics, LNM 1832 (pp. 415–446). Berlin, Germany: Springer–Verlag.
- Del Moral, P., Doucet, A., and Jasra, A. (2006) Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society Series B*, 68(3), 411–436.
- Demyanov, V. (2006) Lazy learning sampling for uncertainty assessment. SGM Report (April), Institute of Petroleum Engineering, Heriot-Watt University, UK.
- Demyanov, V., Foresti, L., Christie, M., and Kanevski, M. (2011) Reservoir modelling with feature selection: A kernel learning approach. In: Proceedings of the SPE Reservoir Simulation Symposium, SPE 141510, 21–23 February, The Woodlands, Texas, USA.
- Demyanov, V., Foresti, L., Kanevski, M., and Christie, M. (2010) Multiple kernel learning approach for reservoir modelling. In: Proceedings of the 12th European Conference on the Mathematics of Oil Recovery (ECMOR XII), B001, 6–9 September, Oxford, UK.
- Demyanov, V., Pozdnoukhov, A., Christie, M., and Kanevski, M. (2010) Detection of optimal models in parameter space with support sector machines. In P. Atkinson and C. Lloy (Eds.), *GeoENV VII – Geostatistics for Environmental Applications: Proceedings of the 7th European Conference on Geostatistics for Environmental Applications* (pp. 345–358), Chester, UK: Springer–Verlag Berlin Heidelberg.
- Demyanov, V., Pozdnoukhov, A., Kanevski, M. and Christie, M. (2008) Geomodelling of a Fluvial system with semi-supervised support vector regression. In: *Proceedings of the VII International Geostatistics Congress*, Gecamin, Chile, 627–636.

- Denney, D. (2009) Paper Review: Peters, E., Arts, R.J., Brouwer, G.K., Geel, C.R. Results of the Brugge benchmark study for flooding optimisation and history matching. Reservoir simulation and visualisation. *Journal of Petroleum Technology*, 61(7), 48–49.
- Deutsch, C., and Cockerham, P. (1994) Practical considerations in the application of simulated annealing to stochastic simulation. *Mathematical Geology*, 26(1), 67–82.
- Deutsch, C., and Journel, A. (1992) Annealing techniques applied to the interpretation of geological and engineering data. In: (Editions Technip Ed.), Report for Stanford Center for Reservoir Forecasting, Stanford University.
- Doyen, P. (2007) *Seismic Reservoir Characterisation: An Earth Modelling Perspective*. DB Houten, Utrecht, The Netherlands: EAGE Publications BV.
- Du, F., Shi, W., Chen, L., Deng, Y., and Zhu, Z. (2005) Infrared image segmentation with 2-D maximum entropy method based on particle swarm optimisation (PSO). *Pattern Recognition Letters*, 26(5), 597–603.
- Duane, S., Kennedy, A., Pendleton, B., and Roweth, D. (1987) Hybrid Monte Carlo. *Physics Letters B*, 195(2), 216–222.
- Dubrule, O. (2003) *Geostatistics for Seismic Data Integration in Earth Models*. Tulsa, OK, USA: SEG/EAGE Distinguished Instructor Short Course, 226E.
- Dunkley, J., Bucher, M., Ferreira, P., Moodley, K., and Skordis, C. (2005) Fast and reliable MCMC for cosmological parameter estimation. *Monthly Notices of the Royal Astronomical Society*, 356, 925–936, arXiv,astro-ph., 0405462.
- Durlofsky, L. (2003) Upscaling of geocellular models for reservoir flow simulation: A review of recent progress. In: Proceedings of the 7th International Forum of Reservoir Simulation, 23–27 June, Bühl/Baden–Baden, Germany.
- Eberhart, R., and Shi, Y. (2001) Particle swarm optimisation: Developments, applications and resources. In: Proceedings of IEEE Congress on Evolutionary Computation, IEEE Service Centre, 1, 81–86, Seoul, South Korea.
- Eberhart, R., and Shi, Y. (2007) *Computational Intelligence: Concepts to Implementations*. Burlington, USA: Morgan Kaufmann.
- Enchery, G., and Roggero, F., and Ravalec, M., and Tillier, E., and Gervais, V. (2010) A new parameterisation technique for the calibration of facies proportions in history matching processes. In: Proceedings of the 12th European Conference

- on the Mathematics of Oil Recovery (ECMOR XII), B009, 6–9 September, Oxford, UK.
- Engelbrecht, A. (2005) *Fundamentals of Computational Swarm Intelligence*. Chichester, England, UK: John Wiley & Sons.
- Erbas, D. (2007) Sampling strategies for uncertainty quantification in oil recovery prediction. PhD Thesis, Heriot–Watt University, Edinburgh, UK.
- Erbas, D., and Christie, M. (2007a) Effect of sampling strategies on prediction uncertainty estimation. In: Proceedings of the SPE Reservoir Simulation Symposium, SPE 106229, 26–28 February, Houston, Texas, USA.
- Erbas, D., and Christie, M. (2007b) How does sampling strategies affect uncertainty estimations?, *Oil and Gas Science and Technology – Rev. IFP*, 62(2), 155–167.
- Evensen, G. (1994) Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99(C5), 10143–10162.
- Evensen, G. (2007) *Data Assimilation: The Ensemble Kalman Filter*. New York, USA: Springer.
- Fanchi, J. (2001) *Principles of Applied Reservoir Simulation* (second edition). Houston, TX, USA: Gulf Professional Publishing.
- Fanchi, J. (2006) *Principles of Applied Reservoir Simulation* (third edition). Burlington, MA, USA: Gulf Professional Publishing.
- Farmer, C. (1992). Numerical rocks. In: King, P. (Ed.), *The Mathematics of Oil Recovery*, (pp. 437–447). Oxford University Press: Oxford, UK.
- Farmer, C. (1989) The mathematical generation of reservoir geology. In: Proceedings of the Joint IMA/SPE European Conference on The Mathematics of Oil Reservoir, Robinson College, July, Cambridge University, UK.
- Farmer, C. (2002) Upscaling: A review. *International Journal for Numerical Methods in Fluids*, 40, 63–78.
- Feller, W. (1968) *An Introduction to Probability Theory and Its Applications* (third edition). New York, USA: John Wiley & Sons.
- Fernández-Martínez, J., Mukerji, T., and García-Gonzalo, E. (2010) Particle swarm optimisation in high dimensional spaces. In M. Dorigo, M. Birattari, G. Di Caro, R. Doursat, A. Engelbrecht, D. Floreano, L. Gambardella, R. Groß, E. Sahin, H. Sayama, and T. Stützle (Eds.), *Swarm Intelligence: Lecture Notes in*

- Computer Science, 6234, (pp. 496–496). Berlin, Germany: Springer–Berlin Heidelberg.
- Ferraro, P., and Verga, F. (2009) Use of evolutionary algorithms in single and multi-objective optimisation techniques for assisted history matching. In: Proceedings of the 10th Offshore Mediterranean Conference and Exhibition RES 04/01 (79), 25–27 March, Ravenna, Italy.
- Fetkovich, M. (1980) Decline curve analysis using type curves. *Journal of Petroleum Technology (JPT)*, SPE 4629–PA, 32(6), 1065–1077.
- Fieldsend, J., and Singh, S. (2002) A multi-objective algorithm based upon particle swarm optimisation, an efficient data structure and turbulence. In: Proceedings of the 2002 UK Workshop on Computational Intelligence, 37–44, September, Birmingham, UK.
- Filippone, M., Calderhead, B., Girolami, M., Mohamed, L., and Christie, M. (2010) Inference for Gaussian Process emulation of oil reservoir simulation codes. In: Proceedings of the ISBA 2010 World Meeting/9th Valencia Meeting, Benidorm, Spain, 3–8 June.
- Floris, F., Bush, M., Cuypers, M., Roggero, F., and Syversveen, A-R. (2001) Methods for quantifying the uncertainty of production forecasts: a comparative study. *Petroleum Geoscience*, 7(S), S87–S96.
- Fonseca, C., and Fleming, P. (1993) Genetic Algorithms for multi-objective Optimisation: Formulation, discussion and generalisation. In: Proceedings of the 5th International Conference on Genetic algorithms, 355–365.
- Foresti, L., Tuia, D., Kanevski, M. and Pozdnoukhov, A. (2010) Learning wind fields with multiple kernels. *Stochastic Environmental Research and Risk Assessment*, 1–16, in press.
- Foresti, L., Tuia, D., Pozdnoukhov, A., and Kanevski, M. (2009) Multiple kernel learning of environmental data. Case study: analysis and mapping of wind fields. In C. Alippi, M. Polycarpou, C. Panayiotou, and G. Ellinas (Eds.), *Artificial Neural Networks – Proceedings of the 19th International Conference on Artificial Neural Networks (ICANN), Part II* (pp. 933–943). Berlin, Germany: Springer–Verlag.
- Freeze, R. (1975) A stochastic-conceptual analysis of one-dimensional groundwater flow in nonuniform homogenous media. *Water Resources Research*, 11(5), 725–741.

- Freeze, R., and Back, W. (1983) Physical hydrogeology. In R. Freeze and W. Back, Stroudsburg (Eds.), *Benchmark Papers in Geology*, 72 (pp. 193–197), Stroudsburg, PA: Hutchinson Ross.
- Gallardo, E., and Leuangthong, O. (2009) A hybrid approach to model selection for support vector classification for facies modelling. In: Proceedings of the International Association for on Mathematical Geosciences (IAMG) Conference, Stanford.
- Galli, A., Beucher, H., Le Loc'h, G., Doligez, B., and Group, H. (1994) The pros and cons of the truncated Gaussian method. In M. Armstrong, and P. Dowd (Eds.), *Geostatistical Simulations*, (PP. 217–233). Dordrecht, The Netherlands: Kluwer Academic Publishers.
- Gamerman, D. (1997) *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. Boca Raton, Florida, USA: Chapman & Hall/CRC.
- García-Gonzalo, E., and Fernández-Martínez, J. (2010) Particle swarm optimisation and inverse problems. In C. Borgelt, G. González-Rodríguez, W. Trutschnig, M. Lubiano, M. Gil, P. Grzegorzewski and O. Hryniewicz (Eds.), *Combining Soft Computing and Statistical Methods in Data Analysis: Advances in Soft Computing*, 77, (pp. 289–296). Berlin, Germany: Springer– Berlin Heidelberg.
- Gavalas, G., Shah, P., and Seinfeld, J. (1976) Reservoir history matching by Bayesian estimation. *SPE Journal*, SPE 5740–PA, 16(6), 337–350.
- Geel, K. (2008) Description of the Brugge field and property realisations, white paper for the Comparative Case Study on Closed-loop Reservoir Management, SPE ATW on Closed-loop Reservoir Management, Bruges, Belgium.
- Gelfand, A., and Smith, A. (1990) Sampling based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410), 398–409.
- Gelman, A., Carlin, J., Stern, H., and Rubin, D. (2004) *Bayesian Data Analysis* (second edition). Boca Raton, USA: Chapman & Hall/CRC, CRC Interdisciplinary Statistics Series.
- Gelman, A., Roberts, G., and Gilks W. (1996) Efficient Metropolis jumping rules. In: J. Bernardo, J. Berger, A. David and A. Smith (Eds.), *Bayesian Statistics* (pp. 599–607). Oxford, UK: Oxford University Press.
- Geman, S., and Geman, D. (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6), 721–741.

- Gerritsen, M., and Durlafsky, L. (2005) Modelling fluid flow in oil reservoirs. *Annual Review of Fluid Mechanics*, 37(1), 211–238.
- Gert, R., Lanckriet, G., De Bie, T., Cristianini, N., Jordan, M., Noble, W. (2004) A statistical framework for genomic data fusion. *Bioinformatics*, 20(16), 2626–2635.
- Geyer, C. (1991) Markov chain Monte Carlo maximum likelihood. In E. Keramidas (Ed.), *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, (pp. 156–163). Berlin, Germany: Springer–Verlag Berlin Heidelberg.
- Geyer, C., and Thompson, E. (1995) Annealing Markov chain Monte Carlo with applications to ancestral inference. *Journal of the American Statistical Association*, 90(431), 909–920.
- Gilks, W., Richardson, S., and Spiegelhalter, D. (1996) *Markov Chain Monte Carlo in Practice*. Boca Raton, Florida, USA: Chapman & Hall/CRC Interdisciplinary Statistics Series.
- Gilks, W., Roberts, G., Sahu, S. (1998) Adaptive Markov chain Monte Carlo through regeneration. *Journal of the American Statistical Association*. 93(443), 1045–1054.
- Giordani, P., and Kohn, R. (2010) Adaptive independent Metropolis-Hastings by fast estimation of mixtures of normals. *Journal of Computational and Graphical Statistics*, 19(2), 243–259.
- Girolami, M., and Calderhead, B. (2011) Riemann manifold Langevin and Hamiltonian Monte Carlo methods (with discussion). *Journal of the Royal Statistical Society, Series B*, 73, Part 2, 1–37.
- Goldberg, D. (1953) *Genetic Algorithms in Search, Optimisation, and Machine Learning*. Addison, Wesley Longman.
- Goldberg, D., and Richardson, J. (1987) Genetic algorithms with sharing for multimodal function optimisation. In: Proceedings of the 2nd International Conference on Genetic Algorithms, 41–49, Lawrence Erlbaum Associates.
- Goldstein, H. (1980) *Classical Mechanics*. MA, Reading: Addison-Wesley.
- Gomez-Hernandez, J., Sahuquillo, A., and Capilla, J. (1997) Stochastic simulation of transmissivity fields conditional to both transmissivity and piezometric data. 1. Theory. *Journal of Hydrology*, 203(1), 162–174.
- Goovaerts, P. (1997) *Geostatistics for Natural Resources Evaluation*. New York, USA: Oxford University Press.

- Green, P., and Murdoch, D. (1998) Exact sampling for Bayesian inference: Towards general purpose algorithms. In J. Bernardo and et al. (Eds.), *Bayesian Statistics 6* (pp. 301–321). Oxford, UK: Oxford University Press.
- Guardiano, F., and Srivastava, R. (1993) Multivariate geostatistics: Beyond bivariate moments. In A. Soares (Ed.), *Geostatistics –Troia* (pp. 133–144). Dordrecht, The Netherlands: Kluwer Academic Publishers.
- Gupta, R., Kilcup, G., and Sharpe, S. (1988) Tuning the hybrid Monte Carlo algorithm. *Physics Review D*, 38(4), 1278–1287.
- Gupta, S., Irb ac, A., Karsch, F., and Petersson, B., (1990) The acceptance probability in the hybrid Monte Carlo method. *Physics Letters B*, 242(3–4), 437–443.
- Haario, H., Laine, M., Mira, A., and Saksman, E. (2005) DRAM – efficient adaptive MCMC. In: Proceedings of the Adap’SkI, the satellite meeting to MCMC’ski 1, 9–11 January, Bormio, Italy.
- Haario, H., Saksman, E., and Tamminen, J. (2005) Componentwise adaptation for high dimensional MCMC. *Computational Statistics*, 20(2), 265–274.
- Hajian, A. (2007). Efficient cosmological parameter estimation with Hamiltonian Monte Carlo technique. *Physical Review D*, 75(8), 083525–1–11.
- Hajizadeh, Y., Christie, M., and Demyanov, V. (2009) Ant colony optimisation for history matching. In: Proceedings of EUROPEC/EAGE Conference and Exhibition, SPE 121193, 8–11 June, Amsterdam, The Netherlands.
- Hajizadeh, Y., Demyanov, V., Mohamed, L., and Christie, M. (2011) Comparison of evolutionary and swarm intelligence methods for history matching and uncertainty quantification in petroleum reservoir models. In Koeppen et al. (Eds.) *Intelligent Computational Optimisation in Engineering: Techniques and Applications*. Springer–Verlag.
- Hammersley, J., and Handscombd, C. (1964) *Monte Carlo Methods*. London, UK: Chapman and Hall/CRC.
- Hansmann, U., Okamoto, Y. and Eisenmenger, F. (1996) Molecular dynamics, Langevin and hybrid Monte Carlo simulations in a multicanonical ensemble. *Chemical Physics Letters*, 259(3–4), 321 – 330.
- Hanson, K. (2001) Markov chain Monte Carlo posterior sampling with Hamiltonian methods. In M. Sonka and K. Hanson (Eds.), *Medical Imaging: Image Processing, Proceedings of Sensitivity Analysis of Model Output*, SPIE 4322, LA-UR-01-1016, 456–467.

- Hanson, K., Cunningham, G., and Mckee, R. (1997) Uncertainty assessment for reconstructions based on deformable geometry. *International Journal of Imaging Systems and Technology*, 8(6) 506–512.
- Hasenbusch, M. (2001) Speeding up the hybrid Monte Carlo algorithm for dynamical fermions. *Physics Letters B*, 519(1–2), 177–182.
- Hastie, T., Tibshirani, R., and Friedman, J. (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (second edition). New York: Springer.
- Hastings, W. (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1), 97–109.
- Ho, S., Shiyou, Y., Guangzheng, N., Lo, E., and Wong, H. (2005) A particle swarm optimisation based method for multi-objective design optimisations. *IEEE Transactions on Magnetics*, 41(5), 1756–1759.
- Hoffman, B., and Caers, J. (2003) Geostatistical history matching using a regional probability perturbation method. In: Proceedings of the 78th Annual Fall Meeting of SPE, SPE 84409, 5–8 October, Denver, Colorado, USA.
- Hoffman, B., and Caers, J. (2004) History matching with the probability perturbation method: Application to a North Sea reservoir. In: Proceedings of the 9th European Conference on the Mathematics of Oil Recovery (ECMOR IX), 30 August–02 September, Cannes, France.
- Holland, J. (1975) *Adaptation in Natural and Artificial Systems*. Ann Arbor: The University of Michigan Press.
- Holland, J. (1998) *Emergence: From Chaos to Order*. Oxford, UK: Oxford University Press.
- Holloman, C., Lee, H., and Higdon, D. (2006) Multiresolution genetic algorithms and Markov chain Monte Carlo. *Journal of Computational and Graphical Statistics*, 15(4), 861–879.
- Hu, L. (2000) Gradual deformation and iterative calibration of Gaussian-related stochastic models. *Mathematical Geology*, 32(1), 87–108.
- Hu, L., Blanc, G., and Noetinger, B. (2001) Gradual deformation and iterative calibration of sequential stochastic simulations. *Mathematical Geology*, 33(4), 475–489.
- Hu, X., and Eberhart, R. (2002) Multi-objective optimisation using dynamic neighbourhood particle swarm optimisation. In: Proceedings of the Congress

- on Evolutionary Computation (CEC'2002), 2, 1677–1681, Piscataway, New Jersey, May, IEEE Service Centre.
- Hu, X., Eberhart, R., and Shi, Y. (2003) Particle swarm with extended memory for multi-objective optimisation. In: Proceedings of the 2003 IEEE Swarm Intelligence Symposium, 193–197, Indianapolis, Indiana, USA, April, IEEE Service Centre.
- Hu, L., Ravalec, M., and Blanc, G. (2001) Gradual deformation and iterative calibration of truncated Gaussian simulations. *Petroleum Geoscience*, 7(S), S25–S30.
- Huang, T, and Mohan, A. (2005) A hybrid boundary condition for robust particle swarm optimisation. *IEEE Antennas and Wireless Propagation Letters*, 4, 112–117.
- Hukushima, K., and Iba, Y. (2003) Population annealing and its application to a spin glass. *The Monte Carlo Method in the Physical Sciences: Celebrating the 50th Anniversary of the Metropolis Algorithm*, AIP conference proceedings, 690 (1), 200–206.
- Hukushima, K., and Nemoto, K. (1996) Exchange Monte Carlo method and application to spin glass simulations. *Journal Physics Society*, 65, 1604–1608.
- Iba, Y. (2001) Population Monte Carlo algorithms. *Transactions of the Japanese Society for Artificial Intelligence*, 16(2), 279–286.
- IC Fault Model (2010) London: Department of Earth Science and Engineering, Imperial College London, <http://www3.imperial.ac.uk/earthscienceandengineering/research/perm/icfaultmodel>, (accessed July 2010).
- Iman, R., and Conover, W. (1980) Small sample sensitivity analysis techniques for computer models, with an application to risk assessment. *Communications in Statistics Theory and Methods*, A9, 1749–1874.
- Ishwaran, H. (1999) Applications of hybrid Monte Carlo to Bayesian generalised linear models: Quasicomplete separation and neural networks. *Journal of Computational Graphical Statistics*, 8(4), 779–799.
- Jacquard, P., and Jaïn, C. (1965) Permeability distribution from field pressure data, *SPE Journal*, SPE 1307–PA, 5(4), 281–294.
- Jafarpour, B., and McLaughlin, D. B. (2007) Efficient permeability parameterisation with the discrete cosine transform. In: Proceedings of the SPE Reservoir Simulation Symposium, SPE 106453, 26–28 February, Houston, Texas, USA.
- Jafarpour, B., and McLaughlin, D. (2009) Reservoir characterisation with the Discrete Cosine Transform. *SPE Journal*, SPE 106453–PA, 14(1), 182–201.

- Jahangiri, H. (2007) Production optimisation using smart well technology with differential evolution algorithm. In: Proceedings of the Student Research Symposium, 16 October, University of Southern California, USA.
- Jahns, H. (1966) A rapid method for obtaining a two-dimensional reservoir description from well pressure response data. *SPE Journal*, SPE 1473–PA, 6(4), 315–327.
- Janson, S., and Merkle, D. (2005) A new multi-objective particle swarm optimisation algorithm using clustering applied to automated docking. In: M. Blesa, C. Blum, A. Roli, and M. Sampels (Eds.), *Hybrid Metaheuristics*, Lecture Notes in Computer Science, 3636, *Second International Workshop, HM 2005* (pp. 128–142), Barcelona, Spain, August, Springer.
- Jaynes, E. (2003) *Probability Theory: The Logic of Science (Vol. 1)*. Cambridge, UK: Cambridge University Press.
- Jin, Y., Olhofer, M., and Sendhoff, B. (2001) Dynamic weighted aggregation for evolutionary multi-objective optimisation: Why does it work and how? In: Proceedings of the Genetic Algorithm and Evolutionary Computation Conference, 1042–1049.
- Jolliffe, I. (2002) *Principal component analysis* (2nd edition). Springer Series in Statistics: New York, USA.
- Journel, A., and Gomez-Hernandez, J. (1993) Stochastic imaging of the Wilmington clastic sequence. *SPE Formation Evaluation*, SPE 19857–PA, 8(1), 33–40.
- Kaipio, J., and Somersalo, E. (2005) *Computational and Statistical Methods for Inverse Problems*. Monograph: Springer–Verlag.
- Kalivarapu, V., Foo, J. L., and Winer, E. (2009) Synchronous parallelisation of particle swarm optimisation with digital pheromones. *Advances in Engineering Software*, 40(10), 975–985.
- Kalman, R. (1960) A new approach to linear filtering and prediction problems, *Transactions of the ASME – Journal of Basic Engineering*, 82(Series D), 35–45.
- Kanevski, M., and Maignan, M. (2004) *Analysis and Modelling of Spatial Environmental Data*. Lausanna, Switzerland: EPFL Press.
- Kathrada, M. (2009a) The flexi-PSO: Towards a more flexible particle swarm optimiser. *OPSEARCH*, Springer India, 46(1), 52–68.

- Kathrada, M. (2009b) Uncertainty evaluation of reservoir simulation models using particle swarms and hierarchical clustering. PhD Thesis, Heriot–Watt University, Edinburgh, UK.
- Kathrada, M., and Carter, J. (2010) Case studies of successfully history matched reservoir simulation models using a powerful optimisation algorithm being of limited predictive value. In: Proceedings of the Abu Dhabi International Petroleum Exhibition and Conference (ADIPEC), SPE 136659, 1–4 November, Abu Dhabi, United Arab Emirates.
- Kennedy, J. (1997) The particle swarm: Social adaptation of knowledge. In: Proceedings of the International Conference on Evolutionary Computation, IEEE Service Centre, 303–308, Piscataway, USA.
- Kennedy, J. (1998) The behaviour of particles. In: Proceedings of the 7th International Conference on Evolutionary Programming VII, 1447, 581–589, San Diego, CA. Berlin: Springer–Verlag.
- Kennedy, J., and Eberhart, R. (1995) Particle swarm optimisation. In: *Proceedings of the IEEE International Conference on Neural Networks*, IEEE Service Centre, 4, 1942–1948, Piscataway, NJ, USA.
- Kennedy, J., and Eberhart, R. (2001) *Swarm Intelligence*. San Francisco, CA: Morgan Kaufmann Publishers.
- Kennedy, M., and O’Hagan, A. (2001) Bayesian calibration of computer models. *Journal of Royal Statistics Society B*, 63(3), 425–464.
- Kim, W., Park, J., and Lee, K. (2009) Stereo matching using population–based MCMC. *International Journal of Computer Vision*, 83(2), 195–209.
- Kind, M., and Quinteros, J. (2007) History matched reservoir model validation based on Wavelets methods. In: Proceedings of the Latin American & Caribbean Petroleum Engineering Conference, SPE 108124, 15–18 April 2007, Buenos Aires, Argentina.
- King, M., and Datta-Gupta, A. (1998) Streamline Simulation: A Current Perspective. *Situ*, 22(1), 91–140.
- King, P. (Ed.) (1992) *The Mathematics of Oil Recovery*. Oxford University Press: Oxford, UK.
- King, P. (1989) The use of renormalisation for calculating effective permeability. *Transport in Porous Media*, 4(1), 37–58.

- King, P. (1996) Upscaling permeability: Error analysis for renormalisation. *Transport in Porous Media*, 23(3), 337–354.
- King, M., King, P., McGill, C., and Williams, J. (1995) Effective properties for flow calculations. *Transport in Porous Media*, 20(1–2), 169–196.
- King, P., Muggeridge, A., and Price, W. (1993) Renormalisation calculations of immiscible flow. *Transport in Porous Media*, 12(3), 237–260.
- Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983) Optimisation by simulated Annealing. *Science*, New Series, 220(4598), 671–680.
- Kita, H., Yabumoto, Y., Mori, N., and Nishikawa, Y. (1996) Multi-objective optimisation by means of the thermodynamic genetic algorithm. *Parallel Problem Solving From Nature—PPSN IV*. In: H.-MVoigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, (Eds.), Berlin, Germany: Springer-Verlag/Heidelberg, Lecture Notes in Computer Science, 504-512, Sept.
- Koh, B., George, A., Haftka, R., and Fregly, B. (2006) Parallel asynchronous particle swarm optimisation. *International Journal for Numerical Methods in Engineering*, 76(4), 578–595.
- Kolehmainen, V. (2001) Novel approaches to image reconstruction in diffusion tomography. PhD thesis, Kuopio University Publications C. Natural and Environmental Sciences, University of Kuopio, Kuopio, Finland.
- Krige, D. (1951) A statistical approach to some mine valuations and allied problems at the Witwatersrand. Master Thesis, University of Witwatersrand, Johannesburg, South Africa.
- Kursawe, F. (1991) A variant of evolution strategies for vector optimisation. In: H. Schwefel and R. Männer (Eds.) *Proceedings of Parallel Problem Solving From Nature* (pp. 193–197): Lecture Notes in Computer Science, 1st Workshop, October, 496, PPSN I, Berlin, Germany: Springer-Verlag.
- Lantuéjoul, C. (1997) Iterative algorithms for conditional simulations. In E. Baafi and N. Schofield (Eds.), *Geostatistics Wollongong 96*, 1, 27–40, Dordrecht, The Netherlands: Kluwer Academic Publishers.
- Lantuéjoul, C. (2002) *Geostatistical Simulation: Models and Algorithms*. Berlin, Germany: Springer-Verlag Berlin Heidelberg.
- Laskey, K., and Myers, J. (2003) Population Markov chain Monte Carlo. *Machine Learning*, 50 (1–2), 175–196.

- Laumanns, M., Thiele, L., Deb, K., and Zitzler, E. (2002) Combining convergence and diversity in evolutionary multi-objective optimisation. *Evolutionary Computation*, 10(3), 263–282.
- Le Loc’h, G., Beucher, H. Galli, A. Doligz, B. and Group, H. (1994) Improvement in the truncated Gaussian method: Combining several Gaussian Functions. In: Proceedings of the 4th European Conference on the Mathematics of Oil Recovery (ECMORE IV), 7–10 June, Røros, Norway.
- Le Loc’h, G. and Galli, A. (1997) Truncated plurigaussian method: Theoretical and practical points of view. In E. Baafi and N. Schofield (Eds.), *Geostatistics Wollongong* (pp. 211–222). Dordrecht, The Netherlands: Kluwer Academic Publishers.
- Le Ravalec, M., Noetinger, B. and Hu, L. (2000) The FFT moving average (FFT-MA) generator: An efficient numerical method for generating and conditioning Gaussian simulations. *Mathematical Geology*, 32(6), 701–723.
- Le Ravalec-Dupin, M. (2005) *Inverse Stochastic Modelling of Flow in Porous Media: Application to Reservoir Characterisation*. Paris, France: Editions Technip.
- Le Ravalec-Dupin, M., and Hu, L. (2004) Gradual deformation of Boolean simulations. In: Proceedings of the 7th International Geostatistics Congress, 26 September–1 October, Banff, Alberta, Canada.
- Le Ravalec-Dupin, M. and Hu, L. (2007) Combining the Pilot Point and Gradual Deformation Methods for Calibrating Permeability Models to Dynamic Data. *Oil & Gas Science and Technology – Rev. IFP*, 62 (2), 169–180.
- Leimkuhler, B., and Reich, S. (2004) *Simulating Hamiltonian Dynamics*. Cambridge, UK: Cambridge University Press.
- Lepine, O., Bissell, R., Aanonsen, S., Pallister, I., and Barker, J. (1999) Uncertainty analysis in predictive reservoir simulation using gradient information. *SPE Journal*, SPE 57594-PA, 4(3), 251–259.
- Li, X. (2003) A non-dominated sorting particle swarm optimiser for multi-objective optimisation. In: E. Cantú-Paz, J. Foster, K. Deb, L. Davis, R. Roy, U. O’Reilly, HG. Beyer, R. Standish, G. Kendall, S. Wilson, M. Harman, J. Wegener, D. Dasgupta, M. Potter, A. Schultz, K. Dowsland, N. Jonoska, and J. Miller (Eds.): *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2003)*: Lecture Notes in Computer Science (pp. 37–48), 2723, Part I, Berlin, Germany: Springer Berlin–Heidelberg.

- Liang, F., and Wong, W. (2000) Evolutionary Monte Carlo: Applications to C_p Model sampling and change point problem. *Statistica Sinica*, 10, 317–342.
- Liang, F., and Wong, W. (2001) Real parameter evolutionary Monte Carlo with applications to Bayesian mixture models. *Journal of the American Statistical Association*, 96(454), 653–666.
- Liu, J. (2008) *Monte Carlo Strategies in Scientific Computing* (corrected edition) New York, USA: Springer–Verlag.
- Liu, N., and Oliver, D. (2005) Critical evaluation of the Ensemble Kalman Filter on history matching of geologic facies. In: Proceedings of the SPE Reservoir Simulation Symposium, SPE 92867, Houston, Texas, USA.
- Liu, N., and Oliver, D. (2003) Evaluation of Monte Carlo methods for assessing uncertainty. *SPE Journal*, SPE 84936–PA, 8(2), 188–195.
- Lolomari, T., Bratvedt, K., Crane, M., Milliken, W., Tyrie, J. (2000) The use of streamline simulation in reservoir management: Methodology and case studies. In: Proceedings of the SPE Annual Technical Conference and Exhibition, SPE 63157, 1-4 October, Dallas, Texas, USA.
- Loshchilov, I., Schoenauer, M., and Sebag, M. (2010) Mono surrogate for multi-objective optimisation. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), 7–11 July, Portland, Oregon, USA.
- Ma, X., Al-Harbi, M., Datta-Gupta, A., and Efendiev, Y. (2008) An efficient two-stage sampling method for uncertainty quantification in history matching geological models. *SPE Journal*, SPE 102476–PA, 13(1), 77–87.
- MacKay, D. (2003) *Information Theory, Inference and Learning Algorithms*. Cambridge, UK: Cambridge University Press.
- MacKay, V. (1994) *Determination of Oil and Gas Reserves*. Petroleum Society SPEC Monograph, 1: The Petroleum Society of the Canadian Institute of Mining, Metallurgy and Petroleum.
- Mackenze, P. (1989) An improved hybrid Monte Carlo method. *Physics Letters B*, 226(3–4), 369–371.
- Mantica, S., Cominelli, A., and Mantica, G. (2002) Combining global and local optimisation techniques for automatic history matching production and seismic data. *SPE Journal*, SPE 78353–PA, 7(2), 123–130.

- Mariethoz, G., Renard, P., Straubhaar, J., and Caers, J. (2010) MP simulations without computing MP statistics. In: Proceedings of the 12th European Conference on the Mathematics of Oil Recovery (ECMOR XII), B002, 6–9 September, Oxford, UK.
- Marinari, E., and Parisi, G. (1992) Simulated tempering: A new Monte Carlo scheme. *Europhysics Letters (EPL)*, 19(6), 451–458.
- Marsily, G. de (1978) De l'identification des systèmes hydrologiques, PhD Thesis, Université Paris VI.
- Marsily, G. de, Lavedan, G., Boucher, M., and Fasanino, G. (1984) Interpretation of interference tests in a well field using geostatistical techniques to fit the permeability distribution in a reservoir model. In: G. Verly et al., D. Reidel, Norwell, Mass. (Ed.): *Geostatistics for Natural Resources Characterisation 2'*, Part 2, (pp. 831–849).
- Martinez, W., and Martinez, A. (2002) *Computation Statistics Handbook with MATLAB*. Boca Raton, Florida, USA: Chapman & Hall/CRC.
- Matheron, G. (1967) *Éléments pour une théorie des milieux poreux*. Paris, France: Masson et Cie.
- Matott, L., Rabideau, A., Craig, J. (2006) Pump-and-treat optimisation using analytic element method flow models. *Advances in Water Resources*, 29(5), 760–775.
- Mattax, C., and Dalton, R. (1990) *Reservoir Simulation*. SPE Monograph #13, Richardson, TX, USA: Society of Petroleum Engineers.
- McKay, M., Beckman, R., and Conover, W. (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1), 55–61.
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953) Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6), 1087–1092.
- Meyn, S., and Tweedie, R. (1993) *Markov Chains and Stochastic Stability*. London, UK: Springer–Verlag.
- Minasny, B., and McBratney, A. (2002) Uncertainty analysis for pedotransfer functions. *European Journal of Soil Science*, 53, 417–430.
- Minasny, B., and McBratney, A. (2006) A conditioned Latin hypercube method for sampling in the presence of ancillary information. *Computers & Geosciences*, 32(9), 1378–1388.

- Mittermeir, G., Amiry, M., and Heinemann, Z. (2010) A new method for the updating and assessing validity of prior reservoir models. In: Proceedings of the 12th European Conference on the Mathematics of Oil Recovery (ECMOR XII), B010, 6–9 September, Oxford, UK.
- Mohamed, L., Calderhead, B., Filippone, M., Christie, M., and Girolami, M. (2010c) Population MCMC methods for history matching and uncertainty quantification. In: Proceedings of the 12th European Conference on the Mathematics of Oil Recovery, B012, 6–9 September, Oxford, UK.
- Mohamed, L., Calderhead, B., Filippone, M., Christie, M., and Girolami, M. (2011b) Population MCMC methods for history matching and uncertainty quantification. *Computational Geosciences*, to appear.
- Mohamed, L., Christie, M., and Demyanov, V. (2009) Comparison of stochastic sampling algorithms for uncertainty quantification. In: Proceedings of the SPE Reservoir Simulation Symposium, SPE 119139, 2–4 February, The Woodlands, Texas, USA.
- Mohamed, L., Christie, M., and Demyanov, V. (2010b) Comparison of stochastic sampling algorithms for uncertainty quantification. *SPE Journal*, SPE 119139–PA, 15 (1), 31–38.
- Mohamed, L., Christie, M., and Demyanov, V. (2010d) Reservoir model history matching with particle swarms. In: Proceedings of the SPE Oil and Gas India Conference and Exhibition, SPE 129152, 20–22 January, Mumbai, India.
- Mohamed, L., Christie, M., and Demyanov, V. (2011a) History matching and uncertainty quantification: Multi-objective particle swarm optimisation approach. In: Proceedings of the SPE EUROPEC 2011, SPE 143067, 23–26 May, Vienna, Austria.
- Mohamed, L., Christie, M., Demyanov, V., Robert, E., and Kachuma, D. (2010a) Application of particle swarms for history matching in the Brugge reservoir. In: Proceedings of the SPE Annual Technical Conference and Exhibition (ATCE), SPE 135264, 20–22 September, Florence, Italy.
- Moore, J., and Chapman, R. (1999) Application of particle swarm to multi-objective optimisation. Department of Computer Science and Software Engineering, Auburn University.
- Mostaghim, S., and Teich, J. (2003a) Strategies for finding good local guides in multi-objective particle swarm optimisation (MOPSO). In: Proceedings of the 2003

- IEEE Swarm Intelligence Symposium, 26–33, Indianapolis, Indiana, USA, April. IEEE Service Centre.
- Mostaghim, S., and Teich, J. (2003b) The role of ε -dominance in multi-objective particle swarm optimisation methods. In: Proceedings of the Congress on Evolutionary Computation (CEC'2003), 3, 1764–1771, Canberra, Australia, December, IEEE Press.
- Mostaghim, S., and Teich, J. (2004) Covering Pareto optimal fronts by subswarms in multi-objective particle swarm optimisation. In: Proceedings of the Congress on Evolutionary Computation (CEC'2004), 2, 1404–1411, Portland, Oregon, USA, June, IEEE Service Centre.
- Mouser, C., and Dunn, S. (2005) A comparison of particle swarm optimisation and the genetic algorithm. *ANZIAM Journal*, 46(E), C89 –C101.
- Mühlenbein, H., Schomisch, D., and Born, J. (1991) The parallel genetic algorithm as function optimiser. *Parallel Computing*, 17(6–7), 619–632.
- Müller, P. (1993) A generic approach to posterior integration and Gibbs sampling. Technical Report 91-09, Department of Statistics, Purdue University, West Lafayette, Indiana.
- Murray, I. (2007) Advances in Markov chain Monte Carlo methods. PhD Thesis, University College London, London, UK.
- Muskat, M., and Wyckoff, R. (1934) A theoretical analysis of waterflooding Networks. *Trans. AIME*, 107, 62–77.
- Nadaraya, E. (1964) On estimating regression. *Theory of Probability and its Applications*, 9(1), 141–142.
- Nævdal, G., Mannseth, T., and Vefring, H. (2002) Near-well reservoir monitoring through ensemble Kalman filter. In: Proceedings of the SPE/DOE Improved Oil Recovery Symposium, SPE 75235, 13–17 April, Tulsa, Oklahoma, USA.
- Neal, R. (1993) Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto, Toronto, Canada.
- Neal, R. (1995) Suppressing random walks in Markov chain Monte Carlo using ordered overrelaxation. In M. Jordan (Ed.), *Learning in Graphical Models*, (pp. 205–228). The Netherlands: Kluwer Academic Publishing.
- Neal, R. (1998) Suppressing random walks in Markov chain Monte Carlo using ordered overrelaxation. In M. Jordan (Ed.), *Learning in Graphical Models* (pp. 205–228). Dordrecht, The Netherlands: Kluwer Academic Publishers.

- Neal, R. (1996a) *Bayesian Learning for Neural Networks*. New York, USA: Springer.
- Neal, R. (1996b) Sampling from multimodal distributions using tempered transitions. *Statistics and Computing*, 6(4), 353–366.
- Neal, R. (2011) MCMC using Hamiltonian dynamics. In B. Brooks, A. Gelman, G. Jones and X-L Meng (Eds.), *Handbooks of Modern Statistical Methods: Handbook of Markov Chain Monte Carlo*. Boca Raton, Florida, USA: Chapman & Hall/CRC.
- Nelson, R. (1960) In-place measurement of permeability in heterogeneous media: 1.Theory of a proposed method. *Journal of Geophysical Research*, 65(6), 1753–1758.
- Ntzoufras, I. (2009) *Bayesian Modelling using WinBUGS*. New Jersey, USA: John Wiley & Sons.
- Oden, T., Moser, R., and Ghattas, O. (2010a) Computer predictions with quantified uncertainty, Part I. *SIAM News*, 43 (9), 1 & 12.
- Okano, H. (2006) Quantification of uncertainty in coarse-scale relative permeability for reservoir production forecast. PhD Thesis, Heriot–Watt University, Edinburgh, UK.
- Oliver, D., and Chen, Y. (2010) Recent progress on reservoir history matching: a review. *Computational Geosciences*, 1–37, in press.
- Oliver, D., Reynolds, A., and Liu, N. (2008) *Inverse Theory for Petroleum Reservoir Characterisation and History Matching*. Cambridge, UK: Cambridge University Press.
- Onwunalu, J., and Durlofsky, L. (2010) Application of a particle swarm optimisation algorithm for determining optimum well location and type. *Computational Geosciences*, 14(1), 183–198.
- O’Sullivan, A. (2004) Modelling simulation error for improved reservoir prediction. PhD Thesis, Heriot–Watt University, Edinburgh, UK.
- Ouenes, A., Brefort, B., Meunier, G., and Dupere, S. (1993) A new algorithm for automatic history matching: application of simulated annealing method (SAM) to reservoir inverse modeling. *SPE Journal*, SPE 26297–PA, 3, 251–259.
- Park, J., Kim, W., and Lee, K. (2007) Stereo matching using population–based MCMC. In Y. Yagi and et al. (Eds.), *Computer Vision – ACCV 2007, Part II*, LNCS 4844 (pp. 560–569). Berlin, Germany: Springer–Verlag Berlin Heidelberg.

- Park, K., Scheidt, C., and Caers, J. (2010) Mathematical reformulation of highly nonlinear large-scale inverse problem in metric Space. In: Proceedings of the 12th European Conference on the Mathematics of Oil Recovery, (ECMOR XII), B11, 6–9 September, Oxford, UK.
- Parsopoulos, K., Tasoulis, D., and Vrahatis, M. (2004) Multi-objective optimisation using parallel vector evaluated particle swarm optimisation. In: Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA 2004), 2, 823–828, Innsbruck, Austria, February, ACTA Press.
- Parsopoulos, K., and Vrahatis, M. (2002a) Particle swarm optimisation method in multi-objective problems. In: Proceedings of the 2002 ACM Symposium on Applied Computing (SAC'2002), 603–607, Madrid, Spain, ACM Press.
- Parsopoulos, K., and Vrahatis, M. (2002b) Recent approaches to global optimisation problems through particle swarm optimisation. *Natural Computing*, 1(2–3), 235–306.
- Peaceman, D. (1977) *Fundamentals of Numerical Reservoir Simulation*. Amsterdam, The Netherlands: Elsevier Scientific Publishing Company.
- Pearson, K. (1901) On lines and planes of closest fit to systems of points in space. *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science*, Sixth Series 2, 559–572.
- Pebesma, E., and Heuvelink, G. (1999) Latin hypercube sampling of Gaussian random fields. *Technometrics*, 41(4), 303–312.
- Pedersen, M. and Chipperfield, A. (2009) Simplifying particle swarm optimisation. *Applied Soft Computing*, 10(2), 618–628.
- Peters, E., Arts, R., Brouwer, G., and Geel, C. (2010) Results of the Brugge benchmark study for flooding optimisation and history matching. *SPE Reservoir Evaluation & Engineering*, SPE 119094–PA, 13(3), 391–405.
- Petrovska, I. (2009) Estimation of distribution algorithms for reservoir history matching optimisation. PhD Thesis, Imperial College London, London, UK.
- Petrovska, I., and Carter, J. (2006) Estimation of distribution algorithms for history matching. In: Proceedings of the 10th European Conference on the Mathematics of Oil Recovery, 4–7 September, Amsterdam, The Netherlands.

- Petrovska, I., and Carter, J. (2007) Using population-based incremental learning algorithm to quantify the uncertainty in model parameters, In: Proceedings of the 69th EAGE Conference & Exhibition, 11–14 June, London, UK.
- Petrovska, I., and Carter, J. (2010) Evolutionary algorithms with pairwise conditional sampling for history matching optimisation. In: Proceedings of the 12th Conference on the Mathematics of Oil Recovery (ECMOR XII), B022, 6–9 September, Oxford, UK.
- Pickup, G., Monfared, H., Zhang, P., and Christie, M. (2004) A new way of looking at upscaling. In: Proceedings of the 9th European Conference on the Mathematics of Oil Recovery (ECMOR IX), A018, 30 August–2 September, Cannes, France.
- Potsepaev, R., and Farmer, C. (2010) Application of partial stochastic differential equations to reservoir property modelling. In: Proceedings of the 12th European Conference on the Mathematics of Oil Recovery (ECMOR XII), B003, 6–9 September, Oxford, UK.
- Pozdnoukhov, A., and Kanevski M. (2008) Multi-scale support vector regression for hot spot detection and modelling. *Stochastic Environmental Research and Risk Assessment (SERRA)*, 22, 647–660.
- Press, W., Teufolsky, S., Vetterling, W., and Flannery, B. (1988). *Numerical Recipes in C*. New York, USA: Cambridge University Press.
- Pulido, G., and Coello, C. (2004) Using clustering techniques to improve the performance of a particle swarm optimiser. In: Kalyanmoy Deb et al., (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2004)*: Lecture Notes in Computer Science (pp. 225–237), Seattle, Washington, USA, June, 3102, Springer-Verlag.
- PUNQ-S3 Model (2010) London: Department of Earth Science and Engineering, Imperial College London, <http://www3.imperial.ac.uk/earthscienceandengineering/research/perm/punq-s3model>, accessed September.
- Rakotomamonjy, A., Bach, F., Canu, S., and Grandvalet, Y. (2008) Simple MKL. *Journal of Machine Learning Research*, 9, 2491–2521.
- RamaRao, B., LaVenue, A., de Marsilly, G., and Marietta, M. (1995) Pilot point methodology for automated calibration of an ensemble of conditionally simulated transmissivity fields. 1. Theory and computational experiments. *Water Resources Research*, 31(3), 475–493.

- Ranganathan, A. (2004) The levenberg-marquardt algorithm. Technical Report. Retrieved from <http://www.ananth.in/docs/lmtut.pdf> (accessed 1 October 2010).
- Raquel, C., and Naval, P. (2005) An effective use of crowding distance in multi-objective particle swarm optimisation. In: Proceedings of the Genetic and Evolutionary Computation (GECCO 2005), 257–264, June, Washington, DC, USA.
- Rasmussen, C. (2003) Gaussian processes to speed up hybrid Monte Carlo for expensive Bayesian integrals. In: J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith and M. West (Eds.), *Bayesian Statistics, 7*, (pp. 651–659), Oxford University Press.
- Ravalec-Dupin, M., Enchery, G., Baroni, A., and Da Veiga, S. (2010) Pre-selection of reservoir models from a geostatistics-based petrophysical seismic inversion. In: Proceedings of the SPE EUROPEC/EAGE Annual Conference and Exhibition, SPE 131310, 14–17 June, Barcelona, Spain.
- Ray, T. and Liew, K. (2002) A swarm metaphor for multi-objective design optimisation. *Engineering Optimisation*, 34(2),141–153.
- Razavi, F., and Jalali–Farahani, F. (2008a) Ant colony optimisation: a leading algorithm in future optimisation of petroleum engineering processes. In: L. Rutkowski, R. Tadeusiewicz, L. Zadeh, and J. Zurada (Eds.), *Artificial Intelligence and Soft Computing – ICAISC 2008: Lecture Notes in Computer Science* (pp. 469–478), 5097.
- Razavi, F., and Jalali–Farahani, F. (2008b) Oil production optimisation in petroleum reservoirs by ant algorithm. *Chemical Product and Process Modelling*, 3(1) (Article 14).
- Renard, P., and Marsily, G., de (1997) Calculating equivalent permeability: A review. *Advances in Water Resources*, 20(5–6), 253–278.
- Reyes, M., and Coello C.(2006) Multi-objective particle swarm optimisers: A survey of the state-of-the-art. *International Journal of Computational Intelligence Research*, 2(3), 287–308.
- Ripley, B. (1981) *Spatial Statistics*. Wiley: New York, USA.
- Ripley, B. (1987) *Stochastic Simulation*. Wiley: New York, USA.
- Ripley B. (1988) *Statistical Inference for Spatial Processes*. Cambridge: Cambridge University Press.

- Robert, C. (2007) *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. New York, USA: Springer.
- Robert, C., and Casella, G. (2000) *Monte Carlo Statistical Methods* (second edition). Springer.
- Robert, C., and Casella, G. (2011) A short history of Markov chain Monte Carlo-subjective recollections from incomplete data. In S. Brooks, A. Gelman, G. Jones, X-L Meng (Eds.), *Handbook of Markov Chain Monte Carlo*. Florian, USA. Chapman and Hall/CRC.
- Roberts, G., and Rosenthal, J. (2001) Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, 16(4), 351–367.
- Roberts, G., and Rosenthal, J. (2007) Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms. *Journal of Applied Probability*, 44(2), 458–475.
- Roberts, G., and Rosenthal, J. (2009) Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics*, 18(2), 349–367.
- Roberts, G., and Tweedie, R. (1996) Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4), 341–363.
- Roggero, F., and Hu, L. (1998) Gradual deformation of continuous geostatistical models for history matching. In: Proceedings of the SPE Annual Technical Conference and Exhibition, SPE 49004, 27–30 September, New Orleans, Louisiana.
- Rojas, T. (2010) Realistic prior information for modelling fluvial channels. SGM Report (October), Institute of Petroleum Engineering, Heriot-Watt University, UK.
- Romero, C., Carter, J., Gringarten, A., and Zimmerman, R. (2000a) A modified genetic algorithm for reservoir characterisation. In: Proceedings of the SPE International Oil and Gas Conference and Exhibition, SPE 64765, Beijing, China.
- Romero, C., Carter, J., Zimmerman, R., Gringarten, A. (2000b) Improved reservoir characterisation through evolutionary computation. In: Proceedings of the SPE Annual Technical Conference and Exhibition, SPE 62942, 1–4 October, Dallas, Texas, USA.
- Rosenberg, D. (1977) *Methods for the Numerical Solution of Partial Differential Equations*, Tulsa, Oklahoma, USA: Farrar and Associates.
- Rosenthal, J. (2001) A review of asymptotic convergence for general state space Markov chains. *Far East Journal of Theoretical Statistics*, 5(1), 37–50.

- Rosenthal, J. (2004) Adaptive MCMC Java Applet. Retrieved from <http://probability.ca/jeff/java/adapt.html> (accessed 01 July 2010).
- Rosenthal, J. (2007) AMCMC: An R interface for adaptive MCMC. *Computational Statistics and Data Analysis*, 51(12), 5467–5470.
- Rosenthal, J. (2011) Ergodicity of adaptive MCMC algorithms. In: Proceedings of the Adap'SkIII, the satellite meeting to MCMC'ski 3, 3–4 January, The Canyons, Park City, Utah.
- Rossberg, K. (1983) *A First Course in Analytical Mechanics*. New York: USA: John Wiley & Sons.
- Rothman, D. (1985) Nonlinear inversion, statistical mechanics and residual statics estimation. *Geophysics*, 50(12), 2784–2796.
- Rotondi, M., Nicotra, G., Godi, A., Contento, F., Blunt, M., and Christie, M. (2006) Hydrocarbon production forecast and uncertainty quantification: A field application. In: Proceedings of SPE Annual Technical Conference and Exhibition, SPE 102135, San Antonio, Texas, USA.
- Sablok, R., and Aziz, K. (2008) Upscaling and discretisation errors in reservoir simulation. *Petroleum Science and Technology*, SPE 93372, 26(10–11), 1161–1186.
- Sætrom, J., and Omre, H. (2010) Ensemble Kalman filtering with shrinkage regression techniques. *Computational Geosciences*, 1–22, in press.
- Salazar-Lechuga, M., and Rowe, J. (2005) Particle swarm optimisation and fitness sharing to solve multi-objective optimisation problems. In: Proceedings of the Congress on Evolutionary Computation (CEC'2005) (pp. 1204–1211), September, Edinburgh, Scotland, UK: IEEE Press.
- Saleri, N., Toronyi, R., and Snyder, D. (1992) Data and data hierarchy. *Journal of Petroleum Technology*, 1286–1293.
- Sambridge, M. (1999a) Geophysical inversion with a Neighbourhood algorithm – I. Searching a parameter space. *Geophysical Journal International*, 138(2), 479–494.
- Sambridge, M. (1999b). Geophysical inversion with a Neighbourhood algorithm – II. Appraising the ensemble. *Geophysical Journal International*, 138(3), 727–746.
- Sarma, P., Durlafsky, L., and Aziz, K. (2008a) Computational techniques for closed-loop reservoir modelling with application to a realistic reservoir. *Petroleum Science and Technology*, 26(10–11), 1120–1140.

- Sarma, P., Durlafsky, L., and Aziz, K. (2008b) Kernel principal component analysis for efficient, differentiable parameterisation of multipoint geostatistics. *Mathematical Geosciences*, 40(1), 3–32.
- Schmidt, M. (2009) Function factorisation using warped Gaussian processes. In: Proceedings of the 26th International Conference on Machine Learning (ICML), 14–18 June, Montreal, Canada.
- Schölkopf, B., and Smola, A. (2002) *Learning with Kernels: Support Vector Machines, Regularisation, Optimisation and Beyond*. Cambridge, UK: MIT Press.
- Schulze-Riegert, R., Axmann, J., Haase, O., Rian, D., and You, Y. (2001) Optimisation methods for history matching of complex reservoirs. In: Proceedings of the SPE Reservoir Simulation Symposium, SPE 66393, 11–14 February, Houston, Texas, USA.
- Schulze-Riegert, R., and Ghedan, S. (2007) Modern techniques for history matching. In: Proceedings of the 9th International Forum on Reservoir Simulations, 9–13 December, Abu Dhabi, United Arab Emirates.
- Schulze-Riegert, R., Bagheri, M., Krosche, M., Kück, N., and Ma, D. (2011) Multiple-objective optimisation applied to well path design under geological uncertainty. In: Proceedings of the SPE Reservoir Simulation Symposium, SPE 141712, 21–23 February, The Woodlands, Texas, USA.
- Schulze-Riegert, R., Krosche, M., Abul F., and Ghedan, S. (2007) Multi-objective optimisation with application to model validation and uncertainty quantification. In: Proceedings of the SPE Middle East Oil and Gas Show and Conference, SPE 105313, 11–14 March, Kingdom of Bahrain.
- Schulze-Riegert, R., Krosche, M., and Pajonk, O. (2009) Data assimilation coupled to evolutionary algorithms – a case example in history matching. In: Proceedings of the 2009 SPE/EAGE Reservoir Characterisation and Simulation Conference, SPE 125512, 19–21 October, Abu Dhabi, United Arab Emirates.
- Schütte, C. (1999) Conformational dynamics: Modelling, theory, algorithm, and application of biomolecules. Habilitation Thesis, Department of Mathematics and Computer Science, Free University Berlin, Retrieved from <http://publications.mi.fu-berlin.de/89/1/SC-99-18.pdf> (accessed 24 December 2010).
- Schutte, J., (2001) Particle swarms in sizing and global optimisation. MSc Thesis, University of Pretoria, South Africa.

- Schutte, J., and Groenwold, A. (2005) A study of global optimisation using particle swarms. *Journal of Global Optimisation*, 31(1), 93–108.
- Schutte, J., Reinbolt, J., Fregly, B., Haftka, R., and George, A. (2004) Parallel global optimisation with the particle swarm algorithm. *International Journal for Numerical Methods in Engineering*, 61(13), 2296–2315.
- Sexton, J., and Weingarten, D. (1992) Hamiltonian evolution for the hybrid Monte Carlo algorithm. *Nuclear Physics B*, 380(3), 665–677.
- Shah, P., Gavalas, G., and Seinfeld, J. (1978) Error analysis in history matching: The optimum level of parameterisation. *SPE Journal*, SPE 6508–PA, 18(3), 219–228.
- Shannon, C. (1948) A mathematical theory of communication. *The Bell System Technical Journal*, 27(3), 379–423 & 623–656.
- Shi, Y. and Eberhart, R. (1998) A modified particle swarm optimiser. In: Proceedings of the IEEE International Conference on Evolutionary Computation, IEEE Press, pp. 69–73, Piscataway, NJ, USA.
- SimOpt (2005) Schlumberger GeoQuest reservoir technologies. SimOpt User Guide, 2005A.
- Smith, A., and Roberts, G. (1993) Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society B*, 55(1), 3–23.
- Smola, A., and Schölkopf, B. (1998) A tutorial on support vector regression. NeuroCOLT, Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK.
- Soismier, M. (2009) Ensemble selection and re-parameterisation with EnKF. Technical Report, Geoscience Research Centre, Total E&P, Aberdeen, UK.
- Sousa, T., Silva, A., and Neves, A. (2004) Particle swarm based data mining algorithms for classification tasks. *Parallel Computing*, 30(5–6), 767–783.
- Specht, D. (1991) A general regression neural network. *IEEE Transactions on Neural Networks*, 2(6), 568–576.
- Srinivasan, D., and Seow, T. (2003) Particle swarm inspired evolutionary algorithm (PS-EA) for multi-objective optimisation problem. In: Proceedings of the Congress on Evolutionary Computation (CEC'2003), 3, 2292–2297, Canberra, Australia, December, IEEE Press.

- Stacey, A., Jancic, M., and Grundy, I. (2003) Particle swarm optimisation with mutation. In: Proceedings of the Congress on Evolutionary Computation, pages 1425–1430, Camberra, Australia, 2003. IEEE Press.
- Stallman, R. (1956a) Numerical analysis of regional water levels to define aquifer hydrology. *Transactions American Geophysical Union*, 37(4), 451–460.
- Stallman, R. (1956b) Use of numerical methods for analysing data on ground water levels. *International Association of Scientific Hydrology Publications*, 41, Tome II, 227–231.
- Strebelle, S. (2002) Conditional simulation of complex geological structures using multiple-point statistics. *Mathematical Geology*, 34(1), 1–21.
- Strebelle, S., and Journel, A. (2001) Reservoir modelling using multiple-point statistics. In: Proceedings of the SPE Annual Technical Conference and Exhibition, SPE 71324, 30 September–3 October, New Orleans, Louisiana.
- Strebelle, S., and Journel, A. (2000) Sequential simulation drawing structures from training images. In: Proceedings of the 6th International Geostatistics Congress, April, Geostatistical Association of Southern Africa, Cape Town, South Africa.
- Strebelle, S., Payrazyan, K., and Caers, J. (2003) Modelling of deepwater turbidite reservoir conditional to seismic data using principal component analysis and multiple-point geostatistics. *SPE Journal*, SPE 85962–PA, 8(3), 227–235.
- Subbey, S., Christie, M., and Sambridge, M. (2003) A strategy for rapid quantification of uncertainty in reservoir performance prediction. In: Proceedings of the SPE Reservoir Simulation Symposium, SPE 79678, Houston, Texas, USA.
- Subbey, S., Christie, M., and Sambridge, M. (2004) Prediction under uncertainty in reservoir modeling. *Journal of Petroleum Science and Engineering*, 44(1–2), 143–153.
- Suganthan, P. (1999) Particle swarm optimiser with neighbourhood operator. In: Proceedings of the Congress of Evolutionary Computation, IEEE Press, 3, pp. 1958–1962, Washington D.C., USA.
- Sun, N-Z. (1994) *Inverse Problems in Groundwater Modelling (Theory and Applications of Transport in Porous Media)*. Dordrecht, The Netherlands: Kluwer Academic Publishers.
- Swendsen, R., and Wang, J. (1986) Replica Monte Carlo simulation of Spin-Glasses. *Physics Review Letter*, 57(21), 2607–2609.

- Tarantola, A. (1987) *Inverse Problem Theory, Methods for Data Fitting and Model Parameter Estimation*. Amsterdam, The Netherlands: Elsevier Science Publishers.
- Tavassoli, Z., Carter, C., and King, P. (2004) Errors in history matching. *SPE Journal*, SPE 86883-PA, 9(3), 352–361.
- Taylor, A. (1993) Using objective and subjective information to develop distributions for probabilistic exposure assessment. *Journal of Exposure Analysis and Environmental Epidemiology*, 3(3), 285–298.
- Thiele, M. (2001) Streamline simulation. In: Proceedings of the 6th International Forum on Reservoir Simulation, 3–7 September, Schloss Fuschl, Austria.
- Thiele, M. (2010) Streamline simulation for modern reservoir-engineering workflows. *Journal of Petroleum Technology (JPT)*, SPE 118608, 62(1), 64–70.
- Thiele, M., and Batycky, R. (2001) Discussion of SPE 65604 – Streamline simulation: A technology update. *Journal of Petroleum Technology*, 53(5), 26–27.
- Thomas, G. (1982) *Principles of Hydrocarbon Reservoir Simulation*. Boston, MA, USA: International Human Resources Development Corporation.
- Thomas, G., and Thurnau, D. (1983) Reservoir simulating using an adaptive implicit method. *SPE Journal*, 23(5), 759–768.
- Tierney, L. (1994) Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22(4), 1701–1728.
- Tjelmeland, H., and Eidsvik, J. (2004) On the use of local optimisations within Metropolis-Hastings updates. *Journal of the Royal Statistical Society Series B*, 66(2), 411–427.
- Tjelmeland, H., and Hegstad, B. (2001) Mode jumping proposals in MCMC. *Scandinavian Journal of Statistics*, 28(1), 205–223.
- Tomandl, D., and Schober, A. (2001) A modified general regression neural networks (MGRNN) with new, efficient training algorithms as a robust ‘black box’ – tool for data analysis. *Journal of Neural Networks (Elsevier Science)*, 14(8), 1023–1034.
- Torby, B. (1984) *Advanced dynamics for Engineers*. Austin, Texas, USA: Holt Rinehart & Winston.
- Towler, B. (2002) *Fundamental Principles of Reservoir Engineering*. SPE Text Book Series, 8: Society of Petroleum Engineers.

- Trelea, I. (2003) The particle swarm optimisation algorithm: Convergence analysis and parameter selection. *Information Processing Letters*, 85(6), 317–325.
- Tuckerman, M., Berne, B., Martyna, G., and Klein, M. (1993) Efficient molecular dynamics and hybrid Monte Carlo algorithms for path integrals. *The Journal of Chemical Physics*, 99(4), 2796–2808.
- Ursem, R., and Vadstrup, P. (2004) Parameter identification of induction motors using stochastic optimisation algorithms. *Applied Soft Computing*, 4 (1) 49–64.
- Valjak, M. (2008) History matching and forecasting with uncertainty: Challenges and proposed solutions for real life field applications. PhD Thesis, Heriot–Watt University, Edinburgh, UK.
- Vapnik, V. (1999) *The Nature of Statistical Learning Theory* (second edition). Berlin, Germany: Springer–Verlag Berlin Heidelberg.
- Vapnik, V., Golowich, S., and Smola, A. (1997) Support vector method for function approximation, regression estimation, and signal processing. In M. Mozer, M. Jordan, and T. Petsche (Eds.), *Neural Information Processing Systems*, 9 (pp. 281–287). Cambridge, MA: MIT Press.
- Velez-Langs, O. (2005) Genetic algorithms in oil industry: An overview. *Journal of Petroleum Science and Engineering*, 47(1–2), 15–22.
- Venter, G., and Sobieszcanski-Sobieski, J. (2006) A parallel particle swarm optimisation algorithm accelerated by asynchronous evaluations. *Journal of Aerospace Computing, Information, and Communication*, 3(3), 123–137.
- Villalobos-Arias, M., Pulido, G., and Coello, C. (2005) A proposal to use stripes to maintain diversity in a multi-objective particle swarm optimiser. In: Proceedings of the 2005 IEEE Swarm Intelligence Symposium, pp. 22–29, June, Pasadena, California, USA: IEEE Press.
- Vrugt, J., Braak, C., Diks, C., Robinson, B., Hyman, J., and Higdon, D. (2009) Accelerating Markov Chain Monte Carlo simulation by differential evolution with self-adaptive randomised subspace sampling, *International Journal of Nonlinear Sciences and Numerical Simulation*, 10(3), 273–290.
- Wahanani, N., Purwaningsih, A., and Setiadipura, T. (2009) Latin hypercube sampling for uncertainty analysis. *Journal of Theoretical Computational studies*, 8(0408), 1–3.
- Walker, G., and Lane, H. (2007) Assessing the accuracy of history matching predictions and the impact of time-lapse seismic data: A case study of the Harding reservoir.

- In: Proceedings of the SPE Reservoir Simulation Symposium, SPE 106019, 26–28 February, Houston, Texas, USA.
- Walsh, B. (2004) Markov chain Monte Carlo and Gibbs sampling. Lecture Notes for EEB 581.
- Watson, G. (1964) Smooth regression analysis. *Sankhya*, 26(1), 359–372.
- Williams, G., Mansfield, M., MacDonald, D., and Bush, M. (2004) Top-down reservoir modelling. In: Proceedings of the SPE Annual Technical Conference and Exhibition, SPE 89974, 26–29 September, Houston, Texas, USA.
- Wolff, U. (2004) Monte Carlo errors with less errors. *Computer Physics Communications*, 156(2), 143–153.
- Wyss, G., and Jorgensen, K. (1998) A user's guide to LHS: Sandia's Latin hypercube sampling software. Technical Report SAND98–0210. Sandia National Laboratories, Albuquerque, NM.
- Xu, W., and Journel, A. (1993) GTSIM: Gaussian truncated simulations of reservoir units in a West Texas carbonate field. SPE 27412.
- Xu, W., Tran, T., Srivastava, R., and Journel, A. (1992) Integrating seismic data in reservoir modelling: The collocated cokriging alternative. In: Proceedings of the SPE Annual Technical Conference and Exhibition, SPE 24742, 4–7 October, Washington, D.C., USA.
- Yu, T., Wilkinson, D., and Castellini, A. (2008) Constructing reservoir flow simulator proxies using genetic programming for history matching and production forecast uncertainty analysis. *Journal of Artificial Evolution and Applications*, 1–13.
- Zabalza-Mezghani, I. (2000) Analyse statistique et planification de'expériences en Ingénierie de réservoir, Mémoire de these, Universite de Pau, Pau, France.
- Zhang, L., Zhou, C., Liu, X., Ma, Z., and Liang, Y. (2003) Solving multi-objective optimisation problems using particle swarm optimisation. In: Congress on Evolutionary Computation (CEC'2003), 3, 2400–2405, Canberra, Australia, December, IEEE Press.
- Zhang, Y., and Pinder, G. (2003) Latin-hypercube sample-selection strategies for correlated random hydraulic-conductivity fields. *Water Resources Research*, 39(8), 1226.

- Zhao, B., and Cao, Y. (2005) Multiple objective particle swarm optimisation technique for economic load dispatch. *Journal of Zhejiang University Science*, 6A(5),420–427.
- Zimmerman, D., Marsily, G. de, Gotway, C., Marietta, M., Axness, C., Beauheim, R., Bras, R., Carrera, J., Dagan, G., Davies, P., Gallegos, D., Galli, A., Gómez-Hernández, J., Grindrod, P., Gutjahr, A., Kitanidis, P., Lavenue, A., McLaughlin, D., Neuman, S., RamaRao, B., Ravenne, C., and Rubin, Y. (1998) A comparison of seven geostatistically based inverse approaches to estimate transmissivities for modelling advective transport by groundwater flow. *Water Resources Research*, 34(6), 1373–1413.
- Zitzler, E. (1999) Evolutionary algorithms for multi-objective optimisation: Methods and applications. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.
- Zitzler, E., Deb, K., Thiele, L. (2000) Comparison of multi-objective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2), 173–195.
- Zlochin, M., and Baram, Y. (2001) Manifold stochastic dynamics for Bayesian learning. *Neural Computation*, 13(11), 2549–2572.
- Øren, P., and Bakke, S. (2002) Process based reconstruction of sandstones and prediction of transport properties. *Transport in Porous Media*, 46(2–3), 311–343.
- Øren, P., Bakke, S., and Arntzen, O. (1998) Extending predictive capabilities to network models. *SPE Journal*, SPE 38880–PA, 3(4), 324–336.