

DOE/PC/91008-16
Distribution Category UC-122

Reservoir Characterization Modeling: Nonstationary Models for Quantitative Geological
Characterization

By
Dennis Kerr
Duryodan Epili
Mohan Kelkar
Richard Redner
Al Reynolds

December 1998

Work Performed Under Subcontract No. 93-0001/47 Contract DE-AC22-94PC91008

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

Rhonda P. Lindsey, Technology Manager
National Petroleum Technology Office
P.O. Box 3628
Tulsa, OK 74101

Prepared by:
The University of Tulsa
600 South College Avenue
Tulsa, OK 74104

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

MASTER

DISCLAIMER

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

DISCLAIMER

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

TABLE OF CONTENTS

PART I

EXECUTIVE SUMMARY	xi
--------------------------------	----

PART II

PROJECT OBJECTIVES AND ACCOMPLISHMENTS	1
Facies Architecture	1
Seismic Modeling and Interpretation.....	2
Markov Random Field and Boolean Models for Geologic Modeling of Facies Distributions.....	3
Estimation of Geological Architecture Using Bayesian/Maximum Entropy Approach	5

PART III

FACIES ARCHITECTURE (<i>D.R. Kerr, Kexian Yang and Amy Richardson</i>).....	7
SUMMARY	7
INTRODUCTION	8
FLUVIAL FACIES ARCHITECTURE DIMENSIONS	9
Meandering Channel Dimensions.....	9
Geomorphology Approach.....	9
Sand-Body Geometry Approach.....	10
Selected Literature Compilation	12
Crevass Splay Dimensions.....	13
General Assessment.....	16
PROCESSED-BASED 3D FLUVIAL ARCHITECTURE SIMULATIONS:	
MIDDLE FRIO FORMATION SOUTH TEXAS	16
Overview of Frio Formation and Stratton Field Study Area	17
Geological Characterization.....	19
Simulation Model and Program	32
Description of the Simulation Model.....	36
Program Design and Implementation	48
Simulation and Realizations	51
Considerations in Defining Input Parameters	51
Suggestions and Further Improvements	
REFERENCES	

APPENDIX I: Computer Programs	71
Description.....	71
Instructions.....	71

PART IV

SEISMIC MODELING AND INTERPRETATION (<i>D. Epili, C. Chung, B.M. Al-moughraby and I Arpandi</i>)	73
SYNTHETIC SEISMOGRAM (1-D, ZERO OFFSET, NO MULTIPLES)	73

Procedure to generate synthetic seismogram (vertical ray path).....	74
Fluvial Channels and Their Seismic Response.....	74
Vertical Resolution	75
Horizontal Resolution	75
AMPLITUDE	76
D Simulation (Amplitude with transmission loss and multiples).....	77
Reflections coefficient at normal incidence.....	77
Pressure amplitude	78
Particle velocity amplitude	78
Transmission coefficient at normal incidence	78
Pressure amplitude	79
Particle velocity amplitude	79
LATTICE FORMULATION	80
Simulation examples.....	81
Earth's geological model	81
Use of amplitude for seismic interpretation.....	82
SEISMIC VELOCITY	82
Introduction.....	82
Seismic velocity and its errors	83
Procedure	85
Synthetic examples	86
SEISMIC INTERPRETATION	87
Introduction.....	87
Interpretation procedure.....	87
Interpretation.....	88
CONCLUSION	89
REFERENCES	91
APPENDIX A	118
APPENDIX B	119
APPENDIX C	123

PART V

MARKOV RANDOM FIELD AND BOOLEAN MODELS FOR GEOLOGIC MODELING OF FACIES DISTRIBUTION (<i>Z.X. Bi, R.A. Redner, D.S. Oliver, Y. Abacioglu and A.C. Reynolds</i>).....	127
SUMMARY	127
INTRODUCTION	128
LITERATURE REVIEW	129
Markov Random Fields.....	132
Neighborhood System and Cliques.....	133
Markov chain Monte Carlo sampling	140
MCMC Algorithm	144
MODELING GEOLOGICAL FACIES DISTRIBUTION BY MRFS	145
MLL Model.....	146
Configuration model	151
Phase transition and penalty term	154

MCMC realizations of facies distribution.....	155
MODEL EXPERIMENTS FOR TWO-DIMENSIONAL PROBLEMS	156
Boundary effects and treatment	157
Experimental results on second-order models	159
Stability of the model and the penalty term	162
Conditioning of the model to observations.....	169
EXPERIMENTS FOR THREE-DIMENSIONAL PROBLEMS	172
Boundary treatment.....	179
Two facies examples.....	179
Three facies case.....	182
Three facies examples.....	184
Model conditioning.....	193
ESTIMATION OF MODEL PARAMETERS	194
Example 1	201
Remarks	208
GENERATING RANDOMLY DISTRIBUTED CHANNELS IN FLUVIAL SYSTEMS.....	209
INTRODUCTION	209
Generating a random cluster of chanel.....	210
Generating random cluster centers.....	211
The deconvolution data problem	215
Nonparametric density estimation	218
Estimation of parameters for the Oligocene Frio Sandstone of South Texas	219
CLOSING REMARKS	222
BIBLIOGRAPHY	227
APPENDIX A	233
APPENDIX B.....	237

PART VI

ESTIMATION OF GEOLOGICAL ARCHITECTURE USING BAYESIAN/MAXIMUM ENTROPY APPROACH (<i>R Li, M. Kelkar and E. Ozkan</i>)	245
SUMMARY.....	245
INTRODUCTION	245
APPROACH	246
An Application Example for Two-Point Connectivity Information.....	250
High Order Moments	253
A Nonlinear Estimation Procedure	257
Application Examples for Four-Point Connectivity Information	260
REFERENCES	263
APPENDIX A	265
APPENDIX B.....	275
APPENDIX C	281
APPENDIX D	282

LIST OF TABLES

Table 1.	Facies architecture characterization based on cross section.....	33
Table 2.	Input Parameters for the Example Realizations	53
Table 3.	Summary of avulsion increment information of concentrated architecture	58
Table 4.	Simulation results for the concentrated architecture	59
	Table 4.1 Earth model.....	93
	Table 4.2 Estimated stacking velocities (V_{stk}) compared to a average Velocities (V_{ave})	93
	Table 4.3 Summary of error observation	94
Table 5.	Summary of avulsion increment information for dispersed architecture	60
	Table 5.3.1 Potential values for realizations (a) through (f) in Fig. 3.4.....	163
	Table 5.4.1 Proportions of facies (percentage)	185
	Table 5.5.1a Results after 1 iteration of the algorithm	203
	Table 5.5.1b Comparison of statistics.....	203
	Table 5.5.2a Results after 2 iterations of the algorithm.....	204
	Table 5.5.2b Comparison of statistics.....	204
	Table 5.5.3 Translated maximum likelihood estimate.....	205
	Table 5.5.4 Initial Guess by Derin and Elliott method	207
	Table 5.5.5 Maximum likelihood estimate	207
Table 6.	Simulation results for the dispersed architecture.....	63

LIST OF FIGURES

Figure 3.1	Cross plot of meandering channel deposits selected from the literature	12
Figure 3.2	Crevasse splay deposit width vs thickness	13
Figure 3.3	Atchafalaya delta dimensions	15
Figure 3.4	Atchafalaya delta dimensions for $1/w$ less than 1	15
Figure 3.5	Atchafalaya delta dimensions for $1/3 >$ than 1	15
Figure 3.6	Location map of the study area (from Hardge et al. 1994 and Levey et al. 1993)	18
Figure 3.7	Type log from the Union Production Company Driscoll No. 7A showing The reservoir nomenclature for Stratton (left) and Agua Dulce (right) Fields (from Kerr 1990)	20
Figure 3.8	Fluvial sedimentological model of middle Frio Formation (from Galloway, 1977; Kerr and Jirik, 1990)	21
Figure 3.9	Facies interpretation on cores cut through part of the middle Frio in Wardner 184 well, Stratton field (from Kerr and Jirik, 1990)	22
Figure 3.10	Well location and cross section index map	23
Figure 3.11	Type well log from Wardner 195 showing reservoir subdivision, markers And facies architecture type of the study interval	24
Figure 3.12	3D view of E41m structure	26
Figure 3.13	3D view of F11m structure	26
Figure 3.14	Interval thickness map between E41m and F11m	27
Figure 3.15	Facies architecture and controlling factors of middle Frio Formation (modified from Kerr and Jirik, 1990)	29
Figure 3.16	Channel belt to gross thickness ratio in different type of facies architecture	30
Figure 3.17	Histogram showing number of channel belts and channel belt cluster width Variation with cross section location	31
Figure 3.18	Channel belt width distribution	34
Figure 3.19	Floodplain configuration (from Bridge and Leeder, 1979)	35
Figure 3.20	Floodplain initialization with a constant slope	37
Figure 3.21	Tectonic sketch of the simulated area	39
Figure 3.22	Diagram showing tectonic subsidence elements calculated in the model	40
Figure 3.23	Diagram showing subsidence rate variation across a local structure Depression	41
Figure 3.24	Channel belt aggradation increase as fault fall compensation	42
Figure 3.25	Diagram showing floodplain aggradation (modified from Mackey and Bridge, 1995)	45
Figure 3.26	Avulsion hierarchy in the model	47
Figure 3.27	Diagram showing channel belt connections definition	52
Figure 3.28	Sample realization of concentrated architecture	56
Figure 3.29	Cross section view of the sample realization for concentrated architecture	57
Figure 3.30	Sample realization of dispersed	61
Figure 3.31	Cross section view of the sample realization for dispersed architecture	62

Figure 4.1	Hypothetical subsurface model with two well log information	95
Figure 4.2	Zero phase Ricker wavelet with 30 Hz frequency.....	95
Figure 4.3	Reflectivity series of Figure 4.1	96
Figure 4.4	Synthetic Seismogram, after convolution of reflectivity (Figure 4.3) with Ricker wavelet (Figure 4.2)	96
Figure 4.5	Subsurface fluvial channel	97
Figure 4.6	Reflectivity series for a single cross section shown in Figure 4.5.....	98
Figure 4.7	Synthetic Seismogram for a fluvial channel, obtained after convolving Ricker wavelet with reflectivity shown in Figure 6.....	98
Figure 4.8	Waves propagation for marine survey (left sketch) and land survey (right sketch)	99
Figure 4.9	Lattice configuration for reflected and transmitted waves in layer k and at interface k	99
Figure 4.10	Numbering scheme for traveltime formulation	99
Figure 4.11	Schematic diagram of downgoing and upgoing waves at equal time for Robinson's model (Robinson, 1983)	100
Figure 4.12	Schematic diagram of downgoing and upgoing waves at equal time for	
Figure 4.13	Depth model	102
Figure 4.14	Dip model (11 degree dip).....	104
Figure 4.15	lateral velocity-variation model.....	106
Figure 4.16	Vertical velocity-variation model.....	108
Figure 4.17	Trap model	110
Figure 4.18	Near-surface anomaly model.....	112
Figure 4.19	A base map of the seismic data area showing the location of the VSP well...	113
Figure 4.20	Line 1 showing the F11 horizon at approximately 1600 ms and E41 at 1555 ms.....	113
Figure 4.21	A prospective view of the target horizon between 1400ms (top) and 1900ms.....	114
Figure 4.22	Crossline (trace) #89 showing the VSP well position.....	114
Figure 4.23	(a) Time slice at 1592ms showing the spot indicated by the arrow in Figure 4.23. Below (b) the same spot became clearer and large at 1594ms time slice	115
Figure 4.24	Fill reservoir time structure map	116
Figure 4.25	E41 reservoir time structure	116
Figure 4.26	Amplitude map for F11	117
Figure 4.27	Amplitude map for E41	117
Figure 5.1.1	(a) first order neighborhood of pixel I consists of the dotted 4 pixels; (b) pixel I, has only three pixels as neighbors.....	134
Figure 5.1.2	A hierarchial sequence of neighborhoods of pixel in a two-dimensional Lattice.....	134
Figure 5.13	Clique types for (a) first-, (b) second-, (c) third-order	135
Figure 5.2.1	Realizations from MLL model	149
Figure 5.2.2	Two-dimensional, 2-pixel clique configurations for 2 facies A and B	153

Figure 5.2.3	Examples of 5-pixel clique configurations for a 2D third-order Neighborhood system.....	154
Figure 5.3.1	Illustration of boundary effect	157
Figure 5.3.2	Periodic boundary and guard region for boundary pixel.....	158
Figure 5.3.3	Clique configurations and the corresponding potentials for three facies A,B and C	160
Figure 5.3.4	Realizations of 2D, second-order, clique configuration models	164
Figure 5.3.5	Realizations from the model (a) of Table 3.1 show similarity in features	165
Figure 5.3.6	The change of proportions of the facies with the number of iterations.....	166
Figure 5.3.7	Realizations of the model listed in Table 3.1(d), with penalty term	167
Figure 5.3.8	Fractions of facies versus the number of iterations, from the model with penalty term (straight line) and without penalty term.....	168
Figure 5.3.9	Variations of facies fractions with the number of iterations for the model With strong penalty, relaxed penalty and without penalty	169
Figure 5.3.10	Realizations from a degenerated model	170
Figure 5.3.11	Facies fractions before and after model conditioning	171
Figure 5.3.12	Comparison of facies fractions as the number of iterations	172
Figure 5.4.1	The cut layers of the three dimensional 3 rd -order neighborhood system	173
Figure 5.4.2	The 7-pixel cliques in the neighborhood system.....	174
Figure 5.4.3	The 7 configurations specified with facies A in the center pixel	176
Figure 5.4.4	Clique configurations and alternatives.....	179
Figure 5.4.5	A realization of 3D	181
Figure 5.4.6	A realization of 3D	181
Figure 5.4.7	The specified 9 configurations with facies A in the center pixel	183
Figure 5.4.8	Configuration 8 with three facies present and facies A in the center pixel.....	182
Figure 5.4.9	A realization of 3D three facies model, 3000 iterations.....	186
Figure 5.4.10	A realization of 3D three facies model, 4000 iterations.....	186
Figure 5.4.11	Examples of 5-pixel clique configurations.....	187
Figure 5.4.12	The cut layers of three-dimensional 2 nd -order neighborhood.....	187
Figure 5.4.13	The 2-pixel cliques in the neighborhood system.....	188
Figure 5.4.14	Vertical clique configurations	188
	Figure 5.4.14(a) A realization of the 3D, 2 nd -order model 400 iterations	190
	Figure 5.4.14(b) A cut view of Fig. 5.4.14a cut at x=y=15, z=10	190
Figure 5.4.15	A realization from this model.....	191
	Figure 5.4.15(a) A realization of the 3D model, 500 iterations.....	192
	Figure 5.4.15(b) View of Fig. 5.4.15a from another position.....	192
Figure 5.4.16	Fractions of facies versus the number of iterations models with and without conditioning data.....	194
Figure 5.5.1a	Third order neighborhood in 2D	196
Figure 5.5.1b	Five-pixel cliques for third order neighborhood	196
Figure 5.5.2	all configurations for a five-pixel clique with facies A in the center pixel of the clique	196
Figure 5.5.3	Alternative configurations for five pixel cliques, two pixels occupied by facies A and A in the center	
Figure 5.5.4	Realizations from the probability function.....	206

Figure 5.5.5	Comparison of statistic n_1 for 3000 samples from the probability functions Based on the true parameters as well as estimated parameters.....	206
Figure 5.5.6	Realizations from the probability functions based on true θ and on the MLE of θ from initial guess obtained by Derin and Elliott	208
Figure 5.5.7	Comparison of statistics n_3 from the samples of the probability functions.....	208
Figure 5.6.1	100 random points in a rectangular region.....	211
Figure 5.6.2	Graph of f using $a=5$, $b=.07$, $c=1$ and $d=b$	212
Figure 5.6.3	A graph of the correction function with $a=3$ and $b=.07$	213
Figure 5.6.4	A picture of 100 well spaced points generated using the MCMC method.....	214
Figure 5.6.5	Distribution of channel thickness	220
Figure 5.6.6	Distribution of width to thickness ratio	220
Figure 5.6.7	Distribution of angle magnitudes	221
Figure 5.6.8a	An oblique view of the simulated reservoir	223
Figure 5.6.8b	A side view of the simulated reservoir	224
Figure 5.6.8c	A top view of the simulated reservoir	225
Figure 6.1	Location of Sampled and Unsampled Points	250
Figure 6.2	Facies Distribution Estimated by Well Data	252
Figure 6.3	Facies Distribution Estimated by Seismic Data	252
Figure 6.4	Facies Distribution Estimated by Using Both Well and Seismic Data	253
Figure 6.5	The Training Image	255
Figure 6.6	Example of Tri-Variance	255
Figure 6.7	Example of Quadri-Variance.....	256
Figure 6.8	Spherical Model for Quadri-Variance Function.....	256
Figure 6.9	Exponential Model for Quadri-Variance Function.....	257
Figure 6.10	Gaussian Model for Quadri-Variance Function	257
Figure 6.11	Comparison of Nonlinear Regression and Kriging Results	260
Figure 6.12	Comparison of Nonlinear Regression and Kriging Results	261
Figure 6.13	Training Image for the Second Example.....	262
Figure 6.14	Comparison of Nonlinear Regression and Kriging Results.....	262
Figure 6.15	Difference Between the Training Image and the Estimates by the Two Methods	263

PART I

EXECUTIVE SUMMARY

Fluvial facies architecture dimensions were investigated through a literature search. Dimensional relationships were considered for meandering channel, crevasse splay and fluvial-dominated delta channels. Although this effort provided insight into the appropriate scale for such facies elements, the usefulness for estimating modeling parameters was very limited.

A three-dimensional process-based stratigraphy simulation model is specified and used to predict the channel sandstone distribution and facies architecture within E and F reservoir units of Middle Frio Formation, Stratton field. The process based fluvial facies architecture simulation model is modified from the model developed by others. Geological processes and features can be simulated in this model include floodplain and channel belt configuration, tectonism, aggradation, compaction and avulsion. Based on the specific geological characteristics in the Middle Frio Formation, modifications are made to incorporate the growth faulting and rollover mechanism, avulsion hierarchy and local subsidence rate variation.

The program is written in C++ language. Each geological process is coded into individual functions and the lower order processes are nested into higher order processes. 35 input parameters are needed to define the geological processes. Simulation output include calculation information, 2D and 3D graphic view of the simulated facies distribution and facies architecture statistic parameters. Multiple realizations are generated to simulate the dispersed and concentrated architecture and the stacking pattern variation within each type of architecture. The major facies architecture features can be successfully simulated.

Geophysics efforts focused on seismic amplitude and velocity are two important attributes for reservoir characterization. The study of seismic amplitude involves the complete field of geophysics due to several contributing factors. Here, we studied the seismic amplitude contribution with and without consideration to transmission losses and with and without incorporating multiples. Also, the contribution to seismic amplitude for marine and land environment are analyzed.

Markov random field and boolean models have been directed at generating realizations of facies distributions constrained to observations (well or outcrop) and geological interpretation with specific emphasis on fluvial systems. The work has emphasized the development of two very different types of statistical models, Markov Random Field (MRF) models and Boolean models.

MRF models allow us to define a joint probability function for the distribution of facies by specifying local characteristics, or more mathematically, conditional probabilities based on local neighborhood systems. This is an extremely convenient feature as it is usually intuitively obvious how to specify the conditional probabilities to

incorporate desired geologic features, continuity, anisotropy (or more generally orientation), and ordering and to some extent overall structure. Once the statistical parameters in the joint probability function have been determined realizations of the facies distribution are generated by a Markov chain Monte Carlo (MCMC) sampling procedure. A penalty term can be incorporated in the probability function to ensure that the fraction of facies in each realization is within some specified range. Conditioning to facies observed at wells is a trivial task. The main model we developed is extremely simple in that it is based on a second order neighborhood system and controls geologic features using only two-pixel cliques. Because of this, we are able to simply use a free boundary condition and avoid the unsolved problem of determining an appropriate definition of potential functions (statistical parameters) on cliques adjacent to the actual three-dimensional reservoir. This simplicity also makes it much easier to define conditional probabilities to simulate desired geologic features. Despite this simplicity, this MRF model can be applied to generate a rich variety of facies configurations. However, use of the model and code developed requires some understanding and intuitive insight on the way conditional probabilities affect geologic features, but much of this insight can be developed simply by reading this report. It would of course be preferable to develop an automatic procedure to generate the statistical parameters which define the probability function for the MRF model directly from geologic interpretation. For this purpose, we have investigated a number of algorithms, e.g., maximum likelihood estimation, coding and histogramming. Many of these methods have been successfully applied in the image analysis literature. Thus, we tried to apply various modifications of these procedures to our problem. Unfortunately, none of them proved generally reliable. The maximum likelihood estimation procedure is reliable only when a good initial guess is available for the statistical parameters, and even then, it is computationally expensive.

Although we do believe that the MRF model developed will prove to be a useful tool for the generation of many facies distributions of interest to geologists, it is not easily applied to generate facies as geometrical objects having a desired structure, e.g., channels and splays in a fluvial system. Thus, we have investigated the application of Boolean methods. With Boolean methods, or more generally, Marked Point Process techniques, distinct objects are inserted into the reservoir. With this approach, well-organized structures are easily obtained. By working closely with geologists, detailed information on the statistics of the parameters in the model can be obtained. For example in our study of the Middle Frio Stratton Field, we have accumulated data on channel thickness, channel width to thickness ratios, the angles that the channel makes with respect to some reference point and the natural clustering of channels. With these and other pieces of data we can make relatively realistic simulations of channel belts. Even more realistic simulations are possible if additional data are incorporated into the model.

The inclusion of specific well and outcrop information can be used to generate statistics to help understand the nature of the field and can also be used to constrain the stochastic realizations that are generated. At this time we have only used the data from the study areas to generate statistics which are used to generate simulations. We have not, at this time, used this type of information to constrain the simulations.

Bayesian maximum entropy (BME) method investigates the use of this method to generate geological facies as part of the reservoir description. The advantages of the proposed method over the conventional approaches are first, the method can account for non-stationarity in geological properties and second, it is general enough so that various constraints can be accommodated as part of the description.

During the first year of the project, we developed a generalized procedure, which accounts for the presence of seismic data as part of the geological description. These functions include variogram, covariances, and cross-covariances among the neighboring values. The method can also account for local probability distribution of facies, which can be obtained from the seismic information. The results indicate that the use of seismic data can improve the facies estimation compared to using simple kriging.

In the second year, we examined the feasibility of extending the method for multipoint connectivity function. Our studies in the second year comprised three main investigations. First we modeled higher order moments. We showed that the higher order moments bear similar characteristics to that of the second order moments. Second, based on a nonlinear regression scheme, we developed an independent means of incorporating multi-point connection into reservoir architecture modeling. This proves to be a robust procedure and improves our ability to estimate the reservoir architecture. It also provides the basis for checking the results obtained by the BME method. In the final phase of the study, we applied the BME method to the four-point connectivity case and showed the improvements of reservoir architecture modeling and thus the potential of the BME method.

PART: II

PROJECT OBJECTIVES AND ACCOMPLISHMENTS

Facies Architecture

An early task/objective for the facies architecture component of the RAM Project was to develop and present tutorials. Tutorial sessions covered the subjects of: stratigraphic and facies architecture principles and concepts; meandering fluvial process and facies; process based simulation model of fluvial facies architecture; and application of process based fluvial facies architecture simulation. The tutorials helped to familiarize the various project members with the fluvial deposition process, facies architecture produced and preserved, geological process based simulation technique and its application. More importantly, these sessions served as a catalyst for developing modeling efforts.

From literature search and compilation, fluvial facies architecture dimensions were assessed. Accomplishments of this investigation include: literature data subsurface data compilation; compilation of other quantitative facies architecture information; and development of synthetic data volume. Relevant information and data are provided to other team members working on different modeling techniques.

Subsurface case study of the fluvial facies architecture on the middle Frio Formation, Stratton field was completed. This study is mainly based on digital log data of 350 wells from Stratton field. A facies architecture model is established according to variations of discrete genetic interval stacking styles. Stacking style is related to other geological processes including aggradation rate change and structure activities.

The fluvial facies architecture of the middle Frio formation was successfully simulated using process-based modeling technique. Achievements include: modified previous simulation models to incorporate the specific geological processes in Stratton area; implemented the simulation model into a C++ program; and applied this simulation model to the middle Frio formation and successfully captured the major facies architecture features.

Some issues that require additional inquiry or improvements are briefly discussed. We were unable to adequately quantify aspects of the facies architecture. Conventional techniques are not very useful in providing the estimation of the modeling parameters. Work needs to address this point in order to obtain realistic modeling output. None of the efforts achieved an approach conditioned on the subsurface data; this point, although stated as a project goal, was considered long-range objective in the project proposal. We were not able to achieve realistic models through a wide range of spatial scales. For example, object-based methods were used to lay down channels or channel belts, but these objects were not filled with any realistic material. One approach discussed (but not fully considered, due to the limited project schedule) is to use hybrid modeling: object-based, controlled in part by process-based routines, are laid down and then the objects are filled using Markov random field process.

Seismic Modeling and Interpretation

Reservoir characterization involves interdisciplinary study of complex geologic features such as boundaries, faults, or facies bodies. The accuracy of imaging these features is a challenging geophysical problem. Also, the accuracy in seismic amplitudes and velocity field will give valuable insight into the observability of pertinent geological and reservoir properties. So, the main objective of the geophysical component of this research is to model and interpret seismic data in the light of the depositional model and utilize this interpretation for conditioning data for the predictive methods that can utilize nonstationary models.

The following tasks were defined:

- a) Literature survey on factors that contribute to seismic amplitude.
- b) Development of software for synthetic seismogram.
- c) Synthetic data for geologic models.
- d) Processing and interpreting seismic data.

In this period of research, we have taken a few more key approaches to better understand a reservoir along with the ones outlined above.

The achievements are as follows:

- a) Delivered 3 tutorials to the group to familiarize the geophysical concepts in reservoir models
- b) Performed literature survey on seismic amplitude and velocity analysis.
- c) Developed algorithm and software for synthetic seismogram; developed the fundamental concept of generating particle velocity synthetic seismogram (for land seismics) and pressure synthetic seismogram (for offshore seismics).
- d) Developed a ray tracing software for computing the travel times over geological models of various degrees of complexities and obtained the uncertainty estimates for velocity calculations.
- e) We have obtained a 3-D seismic data from the Stratton Field, South Texas, loaded on Landmark Interpretation System, and interpreted the E41 and F11 horizons

We have learned that seismic amplitude and velocities are important attributes for reservoir characterization. We have performed a systematic study on the contributing factors and uncertainties. Also, 3-D seismic data provides an extremely good interpretable subsurface image if constrained with the VSP, well log information. However, the main disadvantage of the seismic techniques (if used alone) are the resolution (both vertical and horizontal), and uncertain depth estimation. We have addressed this problem with the field data.

We have not noted any failures in this research, but given more time we would have constrained the geologic model obtained for the Stratton Field data which was mainly produced from the well log correlation.

Markov Random Field and Boolean Models for Geologic Modeling of Facies Distributions

Our over-riding objective was to develop stochastic models for generating facies distributions conditioned to well data with a particular emphasis on nonstationary models. In relation to this primary objective, the main sub-objectives were to (a) to quantify geologic information in order to generate estimates of the statistical parameters which determine the probability function for the distribution of facies, (b) to distribute petrophysical properties within facies.

Regarding the primary objective, we have developed both Markov Random Field (MRF) models and models based on marked point processes (specifically Boolean models) to generate facies distributions. With the Markov random field models developed, we can generate a rich variety of facies distributions. Specifically, based on geologic interpretation and information, one can select the statistical parameters defining the Gibbs probability function for the MRF to generate realizations of the facies distribution that display the desired geologic features, orientation, anisotropy, continuity, ordering, facies proportion and to some extent structure. Moreover, the MRF models can be easily conditioned to facies observed at wells. However, if one wishes to generate facies as large-scale objects, e.g., channels and splays in a fluvial environment, the Boolean model we have developed is preferable. With this model, one can generate channels and splays within a background facies. Conceptually, both models can be applied to nonstationary problems, by defining the statistical parameters as functions of position. However, at this point, this can not be easily done with the code we have developed. Instead to implement nonstationarity, one must use zonation, i.e., divide the volume of interest into specific regions and apply the stochastic algorithm to individual regions. This approach is particularly useful for fields such as the Oligocene Frio Sandstone of South Texas where in the top part of the geologic interval, channels tend to cluster, whereas, the lower portion, channels are randomly distributed with no apparent clustering.

A significant amount of work was devoted to the investigation of procedures to estimate statistical parameters directly from observations, e.g., outcrop data or geologic interpretation. For MRF field models, we modified and experimented with several procedures that have been used successfully in image analysis research. However, we were unable to develop any reliable method. On the other hand, for Boolean models, we were able to generate a stochastic model from well data and geologic interpretation for the Oligocene Frio Sandstone of South Texas using a combination parametric and nonparametric density estimation. Then, by applying a Markov chain Monte Carlo simulation procedure, we are able to generate realizations of channel distributions.

The fruitless time invested in attempting to estimate statistical parameters from geologic information and interpretation precluded an investigation of procedures for distributing petrophysical properties within facies. However, there exist many procedures for doing so, ranging from simple procedures such as assigning the same value at each

gridblock occupied by the same facies to more complex procedures such as Gaussian cosimulation and indicator methods.

In summary, we delineate our successes, failures and lessons learned.

Successes: We have developed Markov Random Field models which can be applied to model a rich variety of facies distributions and capture pertinent geologic features, e.g., orientation, continuity, anisotropy, ordering and facies proportions. With this model, it is easy to condition realizations to facies observed at wells. Boolean models have been developed for generating channel distributions in a fluvial environment. Statistical methods have been developed for estimating the statistical parameters defining the Boolean model from geologic observations and interpretations and applied to a specific field example.

Failures: Procedures to distribute petrophysical properties within facies have not been investigated. We were unable to develop a generally reliable model to estimate the statistical parameters defining the Gibbs probability mass function directly from geologic interpretation.

Lessons Learned: Contrary to previous thought, a rich variety of pertinent geologic features can be captured using MRF models using only two-pixel cliques. High order neighborhood systems are not required. Boolean models for the distribution of channels and crevasse splays within a background facies can be generated using a very simple polygonal parameterization in which all objects lie within intersecting planes. This characterization should prove extremely useful for distributing petrophysical properties within the facies and in conditioning Boolean models to production data. In fact we have already begun work on conditioning Boolean models to well-test pressure data under the auspices of the Tulsa University Petroleum Reservoir Exploitation Projects.

Final Comments: The ultimate goal of reservoir characterization should be to evaluate the uncertainty in predicted reservoir performance under proposed operating conditions. By doing so, one can make more informed reservoir management decisions which will lead to reduced operational and developmental costs. To reduce uncertainty in simulated reservoir descriptions, one clearly should condition models to production data. We fully expect that many of the results obtained in this research will prove useful in our work on generating reservoir realizations conditioned to production data and geologic data and interpretation.

Estimation of Geological Architecture Using Bayesian/Maximum Entropy Approach

The overall project objective was to develop a more flexible technique to describe geological properties. Ideally, the technique should have the ability to capture non-stationarity, typically observed in geological descriptions. Further, the technique should also have the ability to capture geological constraints, which may not be captured using conventional two-point connectivity functions such as variogram and co-variance.

We chose the technique of Bayesian Maximum Entropy to apply it for geological description process. Based on the publications available, the technique had the ability to capture the non-stationarity in description. The technique is also flexible enough to accommodate various geological constraints.

We started implementing Bayesian Maximum Entropy (BME) method by first developing the necessary analytical solutions for two point connectivity functions. These functions include variogram and co-variance. The development of solutions is provided in the Appendix. The method has the flexibility to also incorporate cross-covariance among geological facies. We used the procedure to develop geological facies description in the presence of seismic data, and showed that the method works well in incorporating soft information.

Unfortunately, when we tried to extend the technique to multi-point connectivity functions, we realized that we were not able to obtain analytical solutions. We tried using discrete variables instead of continuous variables, but still were not able to resolve the problem of obtaining analytical solutions. To overcome the problem, we decided to develop a numerical procedure, the details of which are provided in the Appendix. The procedure proved to be very robust, and was able to converge to a solution independent of the starting values. However, since it involved numerical integration, it proved to be time consuming. As the size of multi-point connectivity function increased, the integral size increased proportionately. Computationally, it became more demanding. Although we were able to solve the problems of four-point connectivity function, its practical implications are unclear. The program can be optimized to improve the integration; however, it still seems impractical with current computational resources. Further, as the number of neighboring values increase, the numerical solution becomes more cumbersome.

To briefly summarize the BME method, the method is extremely flexible and robust. It has the ability to accommodate various constraints. We have shown that it can be used in cases where traditional interpolation techniques may not be used. In practical terms, however, the method cannot be easily used because of the requirement of numerical solution. The solution, although robust, is computationally intensive, and cannot be applied in practice at present time.

As an alternative to BME method, we investigated a method that can provide us with flexibility similar to BME method, but, at the same time, provide us with an analytical solution. In the report, we present this new regression technique. The

technique can account for four-point connectivity function – a substantial improvement over conventional kriging technique. The procedure provides an analytical solution, which results in estimation of weights assigned to the neighboring values. The weights are estimated using an iterative scheme; however, the iterative procedure is extremely efficient, and converges very quickly. By comparing the proposed method with conventional kriging technique, we have shown that the proposed method always produces better or as good results as kriging technique. We believe that this technique has the requisite flexibility to incorporate higher point connectivity functions; at the same time, has the computational efficiency to apply it in practice.

PART III

FACIES ARCHITECTURE

By: Dennis R. Kerr, Kexian Yang and Amy Richardson

SUMMARY.

Fluvial facies architecture dimensions were investigated through a literature search. Dimensional relationships were considered for meandering channel, crevasse splay and fluvial-dominated delta channels. Although this effort provided insight into the appropriate scale for such facies elements, the usefulness for estimating modeling parameters was very limited.

The Oligocene Middle Frio Formation is one of the major reservoir interval in the Gulf Coast Basin. The succession is characterized by a sediment-supply dominated fluvial system with rapid deposition and high subsidence. In the study area, the Middle Frio is the major subunit of Frio Formation which is composed of interbedded mudstone and lenticular channel and splay sandstones. Previous geological studies reveals that channel sandbodies are arranged into different stack patterns as response to the varying aggradation rate of the coastal plain. A structural and facies analysis in this study suggests that the growth fault zone normal to the depositional axis and associated rollover structure play an very important role on the distribution of reservoir sandbodies. A quantitative geological characterization is carried out to provide information of input parameters, and also served as a comparison model for the simulation results.

A three-dimensional process-based stratigraphy simulation model is specified and used to predict the channel sandstone distribution and facies architecture within E and F reservoir units of Middle Frio Formation, Stratton field. The process based fluvial facies architecture simulation model is modified from the model developed by Bridge and Leeder (1978), and Mackey and Bridge (1992; 1995). Geological processes and features can be simulated in this model include floodplain and channel belt configuration, tectonism, aggradation, compaction and avulsion. Based on the specific geological characteristics in the Middle Frio Formation, modifications are made to incorporate the growth faulting and rollover mechanism, avulsion hierarchy and local subsidence rate variation.

The program is written in C++ language. Each geological process is coded into individual functions and the lower order processes are nested into higher order processes. 35 input parameters are needed to define the geological processes. All the input parameters are taken according to the quantitative geological characterization, earlier studies of the middle Frio Formation and general geological knowledge. Simulation output include calculation information, 2D and 3D graphic view of the simulated facies distribution and facies architecture statistic parameters.

Multiple realizations are generated to simulate the dispersed and concentrated architecture and the stacking pattern variation within each type of architecture. Major

considerations in the simulation include aggradation rate variation, tectonic subsidence rate variation and probability of different type of avulsions. The major facies architecture features can be successfully simulated.

Process based facies architecture simulation has good potential to produce geologically reasonable realizations in predicting spatial fluvial facies distribution. However, geological study of the simulated area is necessary to understand the geological processes involved and to define the input parameters.

Based on the model used in this study, simulation effects show that the periodic variation of regional aggradation rate caused repeated change of channel fill facies architecture from concentrated to dispersed. This simulation effect is in agreement with the explanation by Kerr and Jirik (1990).

The channel belt avulsion hierarchy, especially the effect of nodal avulsion in response to growth faulting has significant impact on facies stacking pattern variation in the field scale. Channel belt avulsion and deviation from the location of growth fault can results in: 1) increase of channel belt sandbody number and 2) change of channel belt facies stacking pattern from vertical to more lateral.

Depression along deposition direction can produce the effect of channel belt shifting towards the depression axis. As a result, channel belt facies density along the axis is higher than adjacent areas.

INTRODUCTION.

The facies architecture component of the RAM Project pursued the following efforts in the course the project: 1) Developed tutorials for establishing a foundation from which modeling efforts could proceed. 2) Investigated the literature for data and empirical relations of facies architecture element geometry. 3) Developed a synthetic meandering fluvial system volume. 4) Developed and applied a process-based model for meandering fluvial architecture. Items 2 and 4 are presented in this report.

The decision to develop a process-based model came from: 1) The literature investigation of fluvial facies architecture dimensions pointed to a gap in the incorporation of geological processes in controlling fluvial sand-body geometry. This point is particularly important when considering elements of differing scale and potential for sand-body connectedness. 2) This endeavor bore more fruit when working with the other object-based modeling efforts of the RAM Project. However, the limited time of this project did not permit the full integration of these different lines of modeling. 3) The Stratton data set was available and familiar to the investigators.

Details of the processed based modeling can be found in a Masters of Science thesis by K. Yang. The thesis is available through the McFarlin Library at The University of Tulsa. Also Appendix provides instructions for using the program included on disk with

this report. Subsurface maps prepared by D. Kerr (unpublished 1989-1990) were used for parts of this study. Kerr's subsurface maps are archived at The University of Texas Bureau of Economic Geology in Austin, Texas.

FLUVIAL FACIES ARCHITECTURE DIMENSIONS.

An early project objective was to collect fluvial facies architecture geometry information from the literature. This objective was deemed vital to any attempts at realistic simulation of fluvial architecture. This section summarizes the results of this effort.

The most widely available stratal dimension available from subsurface data is thickness. The thickness of a stratal unit is readily measured from wireline logs. The width of a stratal unit is generally unknown, except perhaps in the case of high-resolution 3D seismic image. Thus, stratal width is estimated from some relationship tied to thickness. Following is a summary of relationships between thickness and width of sand bodies found in the literature. Given the depositional systems emphasis stated in the original proposal, meandering channel, crevasse splay, and fluvial-dominated delta dimensions were the focus of the literature search.

Meandering Channel Dimensions

Several empirical equations have been suggested for the dimensional relationships of meandering channel strata.

Geomorphology Approach

Schumm (1963; 1972) developed an equation that relates width and depth of fluvial channels:

$$F = 225 M^{-1.08}$$

where:

F = width:depth ratio

M = weighted mean percent silt and clay

Schumm doesn't give an error for his formula, but Cotter (1971) uses a standard error of 0.20. Schumm (1972) related meander wavelength to F with the equation:

$$l = 18 (F^{0.53} w^{0.69})$$

or

$$\log l = 1.27809 + 0.52822 \log F + 0.68774 \log w.$$

where:

l = meander wavelength

w = bankfull width

The standard error is 0.21 log units.

One of the more commonly used formulas is from Leeder (1973):

$$w = 6.8h^{1.54}$$

where:

w = bankfull width

h = bankfull depth

This relationship is based on 57 modern meandering rivers with sinuosity (channel length to valley length ratio) greater than 1.7. The correlation between w and h is strong (r = 0.91). This equation is widely used (Bridge and Mackey, 1993; Lorenz et al., 1985; Stancliffe and Adams, 1986; Collinson, 1978; and Williams, 1988).

Williams (1986) offers two equations for width and depth relation:

$$w = 15.5 d^{1.4} \quad (\text{for 30 sites with sinuosity} > 1.7)$$

$$w = 21.3 d^{1.45} \quad (\text{for 67 sites irregardless of sinuosity})$$

where:

w = bankfull width

d = bankfull depth

Sand-Body Geometry Approach

Collinson (1978) compiled sand-body shapes for fluvial deposits regarded as being of a meandering channel origin. Thus, this approach uses the depositional geometry, and results in the relationship:

$$W_m = 64.6 t^{1.54}$$

$$\lambda = 74.1 t^{1.54}$$

where:

W_m = channel belt width

t = sand body thickness

λ = meander wavelength

However, there is a lot of variation in the data compiled, and Collinson (1978) offers no measure of the strength of the correlation.

Fielding and Crane (1987) considered Collinson's (1978) equation, but developed a best-fit equation from their own data compilation:

$$W_m = 12.1 t^{1.85}$$

where:

W_m = channel belt width

t = sand body thickness

Again, Fielding and Crane's (1987) data are highly variable and include the deposits from fluvial systems other than meandering. They suggest that meandering fluvial channel belts likely reside between:

$$W_m = 0.95 t^{2.07} \quad \text{upper bounding line}$$

$$W_m = 64.6 t^{1.54} \quad \text{fully developed meanders (Collinson, 1978)}$$

where:

W_m = channel belt width

t = sand body thickness

Fielding and Crane (1987) continue to consider a simple descriptive model for relating thickness of channel-belt sand bodies to assumed widths (i.e. best-fit line equation) in a given well. Next they compute a cross sectional area of all sandstones that is intersected by a well, excluding any that are less than 2 meters thick:

$$\text{cross sectional area} = \sum_{i=1}^n t_i W_i$$

where:

t = thickness

W = channel-belt width

i = sand-body index, from 1 to n (=total number of beds)

The procedure is extended to consider the success rate of an offset well located a distance W (or shorter) from the first well:

$$\text{success rate} = \frac{\sum_{i=1}^n t_i \text{minimum}(w_i, W)}{\sum_{i=1}^n t_i W}$$

where:

t = thickness

W = channel-belt width

i = sand-body index, from 1 to n (=total number of beds)

For secondary or tertiary development the success rate equation was rewritten as

$$\text{success rate} = \frac{\sum (w_i \geq W) t_i W}{\sum_{i=1}^n t_i W}$$

Selected Literature Compilation

Using 20 data points compiled from Williams (1986), Puigdefabregas (1973), Schumm (1972) a graph was made plotting thickness vs. width meandering channel deposits (figure 3.1).

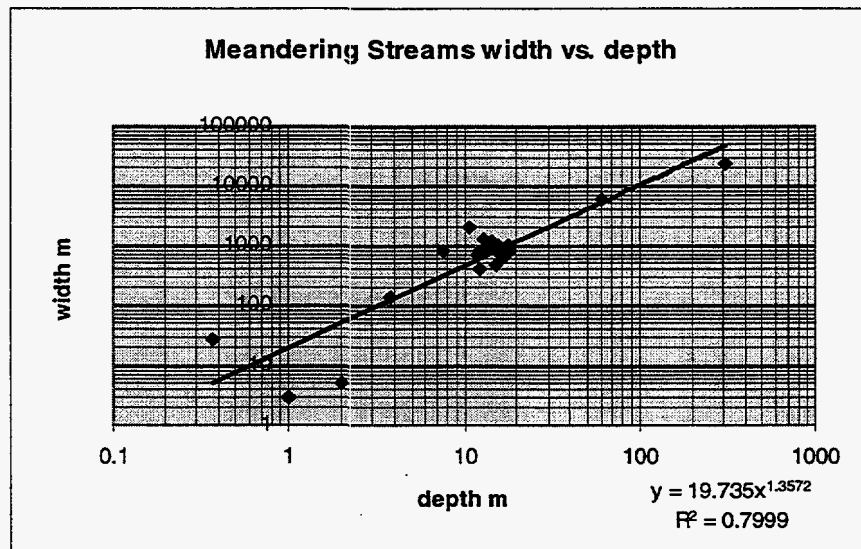


Figure 3.1. Cross plot of meandering channel deposits selected from the literature.

A best-fit line is:

$$W_m = 19.735 t^{1.3572}$$

where:

W_m = channel-belt width

t = thickness

This result is reasonably comparable to the other width to thickness relationships discussed above; the equation is very similar to Leeder's relationship. However, each

width to thickness (or depth) relationship given above offers markedly different results as the thickness increases. This draws into question which to choose, or perhaps is this a useful approach at all.

Crevasse Splay Dimensions

Crevasse splays are overbank deposits that occur when a river flooded breaches its levee. A crevasse channel forms, and fan- or lobate-shaped splay of sand and mud is deposited on the floodplain. Deposits can range from a few inches (cm) in thickness to 10's feet (m). Smith et al. (1989) article related stages of crevasse splays to geometry. Stage I is an immature splay that is small and produces lense or wedge shaped deposits. Stage II forms less tabular deposits that are disconnected by channelization. The cross sections are irregular. Stage III produces linear and more isolated sand bodies as channelization is even more common. With each stage the size of the splay increases.

Few empirical formulas were offered in the literature that related the thickness and width of crevasse splay deposits. Mjos et al. (1993) indicated maximum length:thickness ratio is 2000. More typically the ratio ranges between 150-1500, and thicknesses range 0.3 to 4.5 m. Crevasse channel sandstones have width/thickness ratios of 5-60, and thickness are usually less than 4m, but can range up to 7m.

With data given in Rhee and Chough (1993), O'Brien and Wells (1986), Flores (1984), and Chrzastowski et al. (1994), a graph relating width to thickness came up with a slope of 426.87 (figure 3.2). This fits in with Mjos et al. assumptions, but is not conclusive because of the few data points. Larger splays gave ratios that were closer to the published norms.

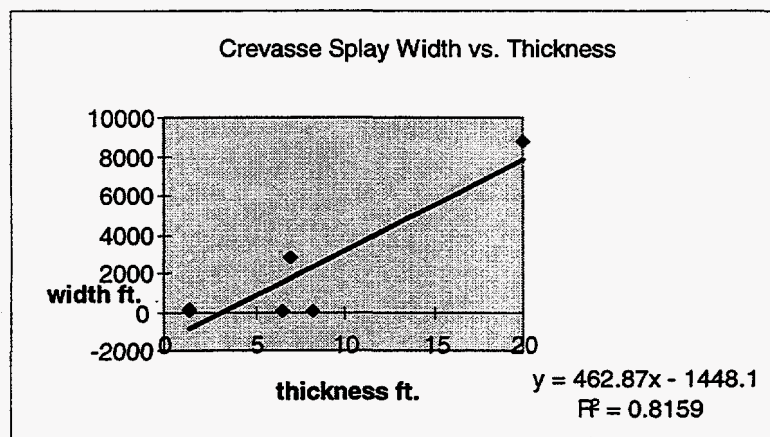


Figure 3.2. Crevasse splay deposit width vs thickness.

Fluvial Delta Dimensions

Despite the voluminous literature on delta systems, the literature seems to contain little factual information on the geometry of entire deltas and channel-fill deposits. There is some research on deltaic channels and bar deposits.

Lowry and Jacobson (1993) give an empirical formula for channel width to thickness ratio for isolated channels that incise mouth-bars:

$$w = 12.7 t^{1.2}$$

where:

w = channel width

t = channel-fill thickness

They also found distributary channel width to thickness ratios range from 10 to 20.

Tyler et al. (1991) found that in the Ferron Sandstone basal channels have fairly consistent width to thickness ratios. The average is 9.5. These are small features at the base of the distributary complex. Falkner and Fielding (1993) found in their study area that proximal mouth bars have width to thickness ratios of 5 to 25.

From study of the Atchafalaya River delta, Wells et al. (1984) mapped five deltas in different stages of development. Each were measured to find length perpendicular to the main river channel, width parallel to the main river channel, and the channel width of the main distributary channel to the delta. The length (l) to width (w) ratio for all deltas is charted in figure 3.3. There seems to be two main groups in this data. The deltas where the l/w ratio is greater than 1 and the group where it is less than 1. The cause of this difference seems to be delata size -- smaller deltas have ratios greater than 1, and the larger deltas have l/w ratios smaller than 1. Compiled data offer a best-fit line:

$$c = -0.0302 l/w + 0.1995 \quad (\text{deltas} < 10 \text{ km in length; figure 3.4})$$

$$c = 0.3663 l/w + 0.3458 \quad (\text{deltas} > 10 \text{ km in length; figure 3.5})$$

where:

c = main distributary channel width

l = length measured perpendicular to main channel

w = width measured parallel to main channel

The correlation coefficient is so low that it is difficult to say whether any real relationship exists among the variables considered. However, the outlier with >1 km channel width is likely distorting the correlation coefficient.

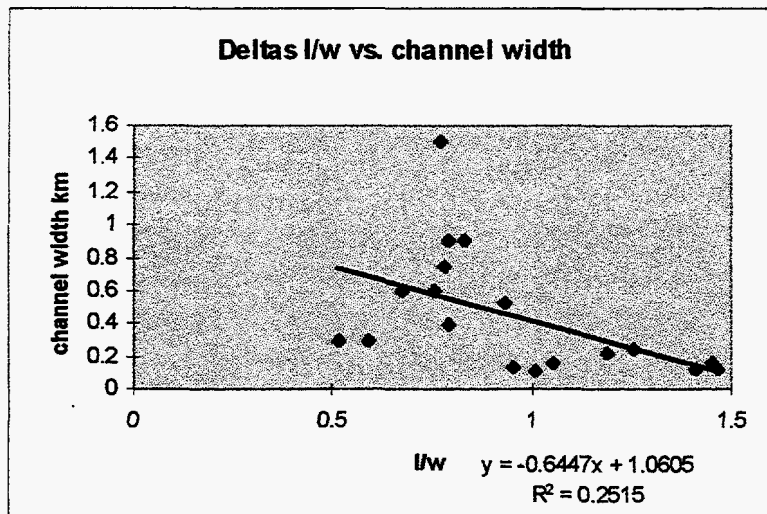


Figure 3.3 Atchafalaya delta dimensions.

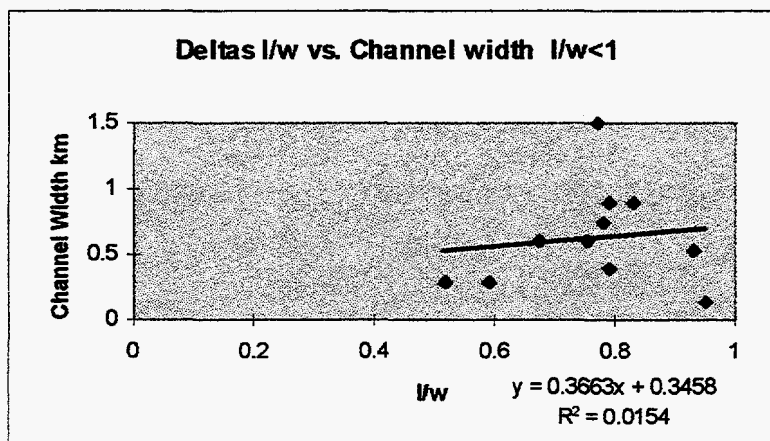


Figure 3.4 Atchafalaya delta dimensions for l/w less than 1.

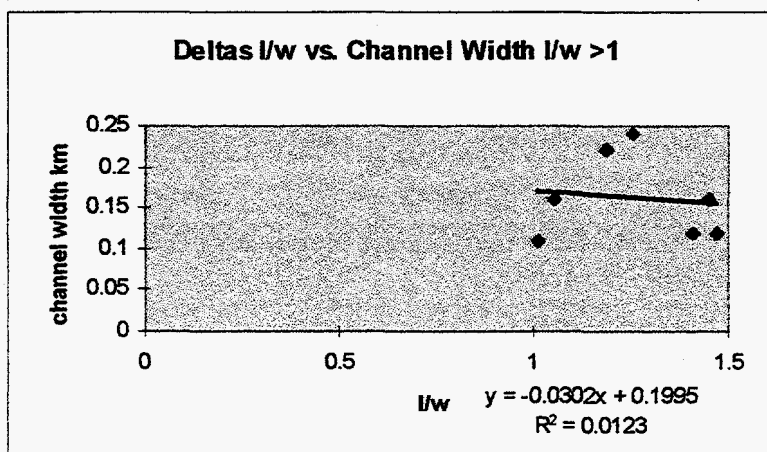


Figure 3.5 Atchafalaya delta dimensions for l/w > than 1.

General Assessment.

The literature is rich with attempts to relate dimensions of fluvial facies architectural elements. Unfortunately, the usefulness for modeling efforts is questionable.

Meandering channel dimensions seem to present reasonable correlations in their dimensional character. The geomorphology based approach provides highly variable results. The problem appears to be that channel width is defined differently, or not clearly defined at all. Channel sand-body relationships are also reasonably correlated. However, other geological processes come in to play in the deposition and preservation of rock volumes. This include, but are not limited to, accommodation space and alluvial plain aggradation.

The data for crevasse splay deposits is interesting. But one needs to keep in mind that crevasse splay sand-bodies are approximated by low-profile cone segments.

The fluival deltaic data is difficult to decipher. It was not used in our work.

The dimensional data and relationships are interesting, but their usefulness to modeling architectural elements is limited. The greatest limitation comes from the fact that individual data sets are small and not well documented in terms of how and exactly what was collected. This is an area the sedimentology community needs to address. For the RAM Project, these relationships were not helpful in estimating parameters for the other modeling efforts.

PROCESSED-BASED 3D FLUVIAL ARCHITECTURE SIMULATION: MIDDLE FRIO FORMATION SOUTH TEXAS.

Motivation and Objectives

Geological process-based 2D and 3D quantitative models of fluvial stratigraphy have been developed to simulate the distribution, proportion and connectedness of coarse grained channel-belt deposits within alluvial strata (Bridge and Leeder, 1972; 1992, Mackey and Bridge, 1995). Geological processes considered in current models usually include: floodplain aggradation; compaction; tectonism; and channel avulsion. Process-based simulation approach has a good potential to predict facies distribution in fluvial successions. Though some preliminary efforts have been made to apply these models to modern natural cases (Mackey and Bridge, 1995), application to ancient subsurface cases in the published literature was not found. Difficulties of the application mainly come from: 1) the models are too generalized or simplified to simulated natural cases with more specific and complicated geological processes; 2) the input parameters needed by the models are difficult to define in real-world cases, especially subsurface cases since data density are not sufficient, and some of the geological processes are not obviously recorded in the rock volume.

Efforts are needed to apply this approach to real-world cases, especially subsurface cases, to understand for specific geological setting, how these processes can influence facies architecture. Motivated by this idea, an object-oriented model will be built based on a real subsurface case -- the middle Frio Formation of south Texas.

Overview of Frio Formation and Stratton Field Study Area

The case chosen to simulate is the middle Frio Sandstone in south Texas (according to the stratigraphic subdivision by Kerr and Grigsby, 1991), which is one of the major reservoir intervals in the Gulf Coast Basin (Galloway et al., 1982; Galloway et al., 1983; Kosters et al., 1989). The study area includes Stratton Field and adjacent area as shown in Fig. 3.6. Earlier geological studies suggest a fluvial-deltaic depositional system (Galloway, 1982). Growth faults and associated structures that cross depositional trends impact sandstone accumulation. The effects of varying aggradation rate with time are thought to be recorded by changing of channel sandbody stacking patterns (Kerr and Jirik, 1990).

Stratton field is located within the onshore Gulf Coast basin of south Texas along the Vicksburg fault zone (Fig. 3.6), near the northern end of FR-4 gas play (Kosters et al., 1989). Geographically, it is in area overlapping Nueces, Kleberg and Jim Wells Counties.

The area of detailed analysis comprises about 31 mi² (80 km²) which covers the major part of Stratton field. The detailed study is mainly based on logs from about 400 wells, mostly in digital format. A rich source of previously published and unpublished studies in this area are available and can serve as good references and foundation to this study. These works include both regional and detailed deposition, structure, reservoir and hydrocarbon resources investigation (Galloway, 1977; Galloway et al., 1982, Galloway et al., 1983; Kosters et al., 1989; Jackson and Galloway, 1984; Ewing, 1986; Kerr, 1989-1990 unpublished maps; Kerr and Jirik, 1990; Kerr, 1990; Kerr and Grigsby, 1991; Levey et al., 1993).

The Oligocene Frio Formation is a major progradational offlapping stratigraphic unit in the Northwest Gulf of Mexico Basin. The succession is characterized by a sediment-supply dominated fluvial system with rapid deposition and high subsidence rate (Galloway and others, 1982; Morton and Galloway, 1991).

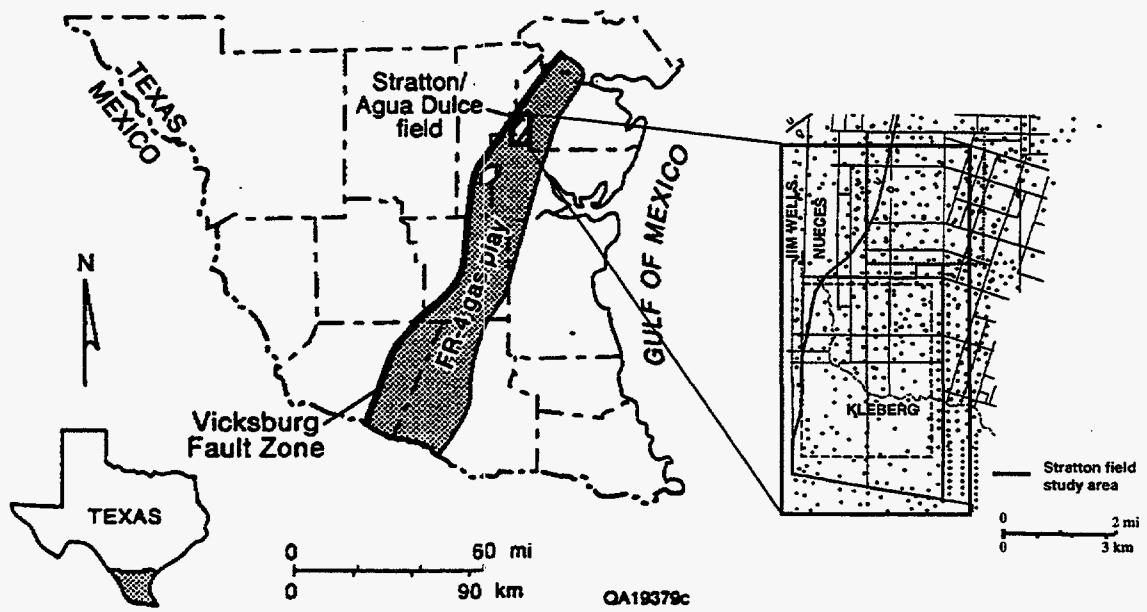


Fig. 3.6. Location map of the study area (from Hardge et al.1994 and Levey et al. 1993).

The Frio Formation is informally subdivided into three subunits in the region along Vicksburg fault zone referred to as the lower, middle and upper. The total thickness of Frio Formation varies from less than 2,000 ft (610 m) to great than 9,000 ft (2740 m). The middle Frio is 2000 to 2500 ft (610 to 760 m) thick and composed of interbedded floodplain mudstone and fluvial channel-fill sandstones and splay sandstones (Kerr and Jirik, 1990; Kerr and Grigsby, 1991). In Stratton Field, the middle Frio Formation is subdivided into F to B reservoirs (Fig. 3.7). The interval of detailed analysis in this study includes E41 to F11 with a total thickness of about 300 ft (91 m).

Stratton field lies within the Rio Grande Embayment structural province (Levey et al., 1993). The structural style in this region is characterized by discontinuous belts of growth faults parallel to the striking direction. A series of normal faults are developed in the Vicksburg and lower Frio Formation and sole out into the Vicksburg detachment zone within the Jackson Shale. The structure framework of the middle and upper Frio Formation is much simpler than the underlying formations and the strata are relatively undeformed and flat-lying. At the study interval from E41 - F11, The structure attitude is mostly affected by the master growth fault while all the antithetic and faults are diminished at this level.

Middle Frio reservoir in Stratton field are part of the Gueydan fluvial system with the general depositional direction from north west to south east which is perpendicular to the growth faults. Middle Frio strata consist of fluvial depositional systems that contain channel-fill and splay reservoir facies (Galloway, 1977; Kerr and Jirik, 1990).

Kerr and Grigsby (1991) show that the volcanic glass present in Frio sandstones at Stratton is absent in sandstones at Seeligson field. This variation is explained by differences in source terrain. That is, different fluvial systems with separate catchment basins may have coexisted and delivered sediments to these two different area.

The established depositional model (Galloway, 1977; Kerr, 1990) suggests a meandering river system composed of floodplain, levee, splay and channel fill/point bar facies (Fig. 3.8).

Criteria for identifying channel-fill and splay facies from cores and well logs were described by Kerr and Jirik (1990), Kerr et al. (1991). Channel fill facies maybe vertically stacked with the upper part of the underlying channel fill being eroded by the overlying channel fill (Fig. 3.9).

Geological Characterization

Field scale structure and facies architecture study is based on well data within the study area. Fig. 3.10 shows the well locations and stratigraphic cross section grid. Two major markers named E41m and F11m are used for stratigraphic correlation. Each corresponds to the top of a extensive and continuous shale (paleosol ?) above E41 and F11 sand respectively (Fig. 3.11).

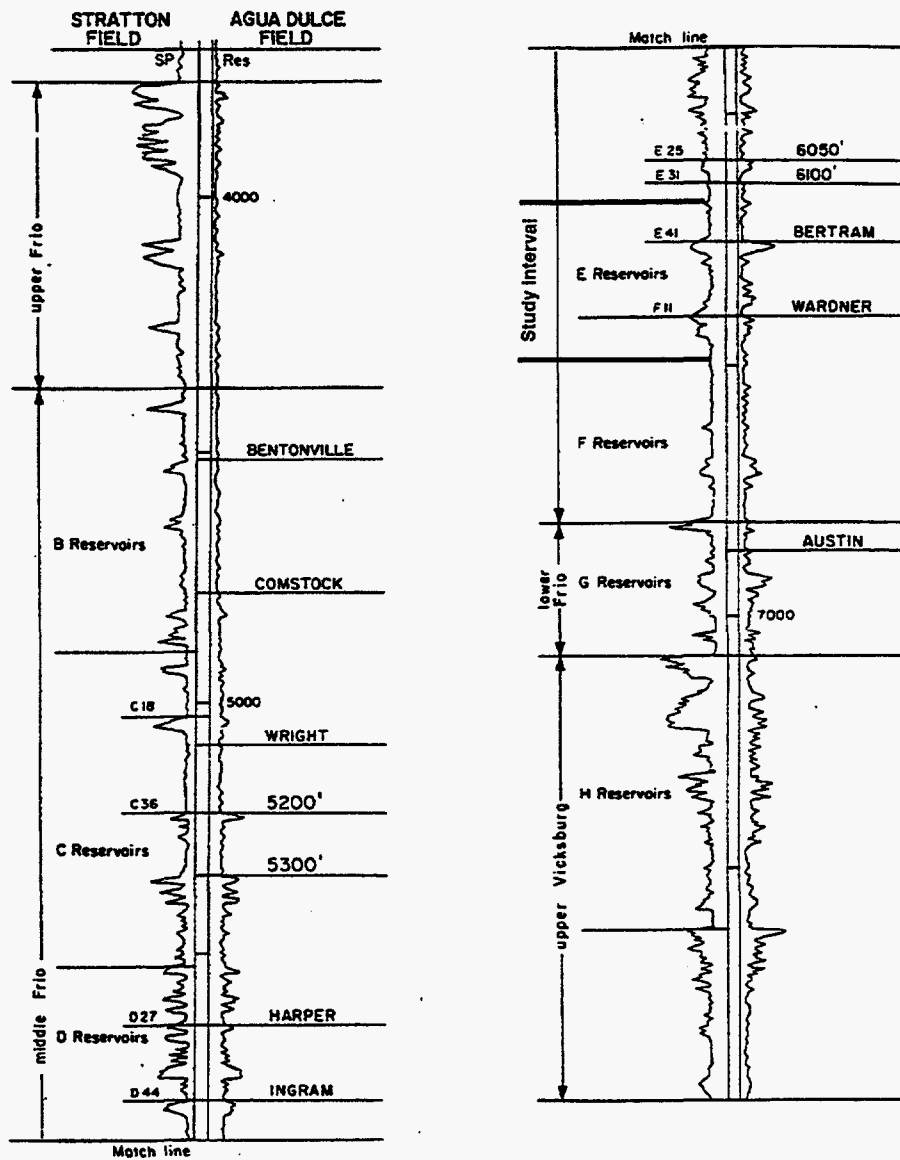


Fig. 3.7. Type log from the Union Production Company Driscoll No. 7A showing the reservoir nomenclature for Stratton (left) and Agua Dulce (right) fields (from Kerr 1990).

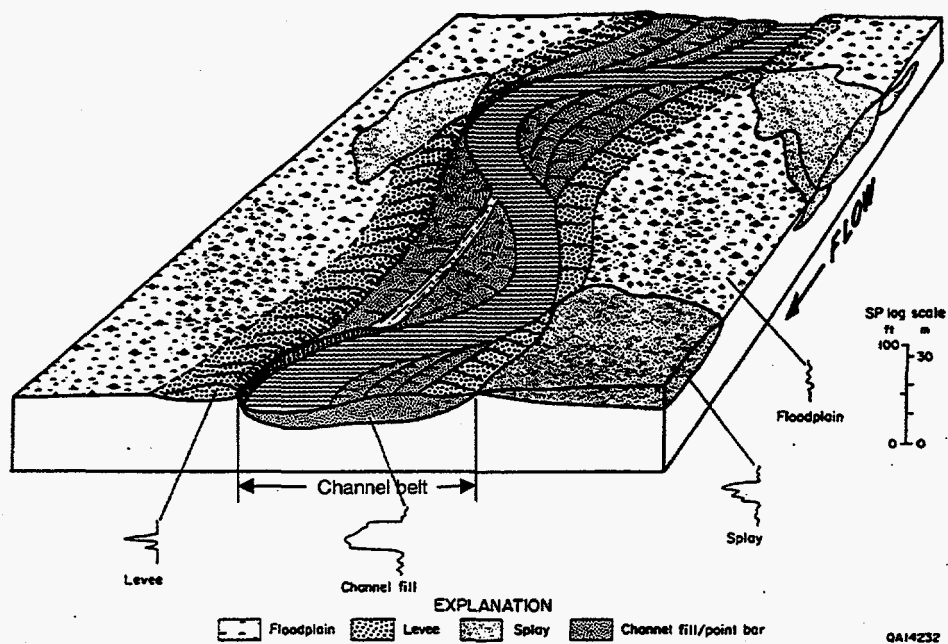


Fig. 3.8. Fluvial sedimentological model of middle Frio Formation (from Galloway, 1977; Kerr and Jirik, 1990).

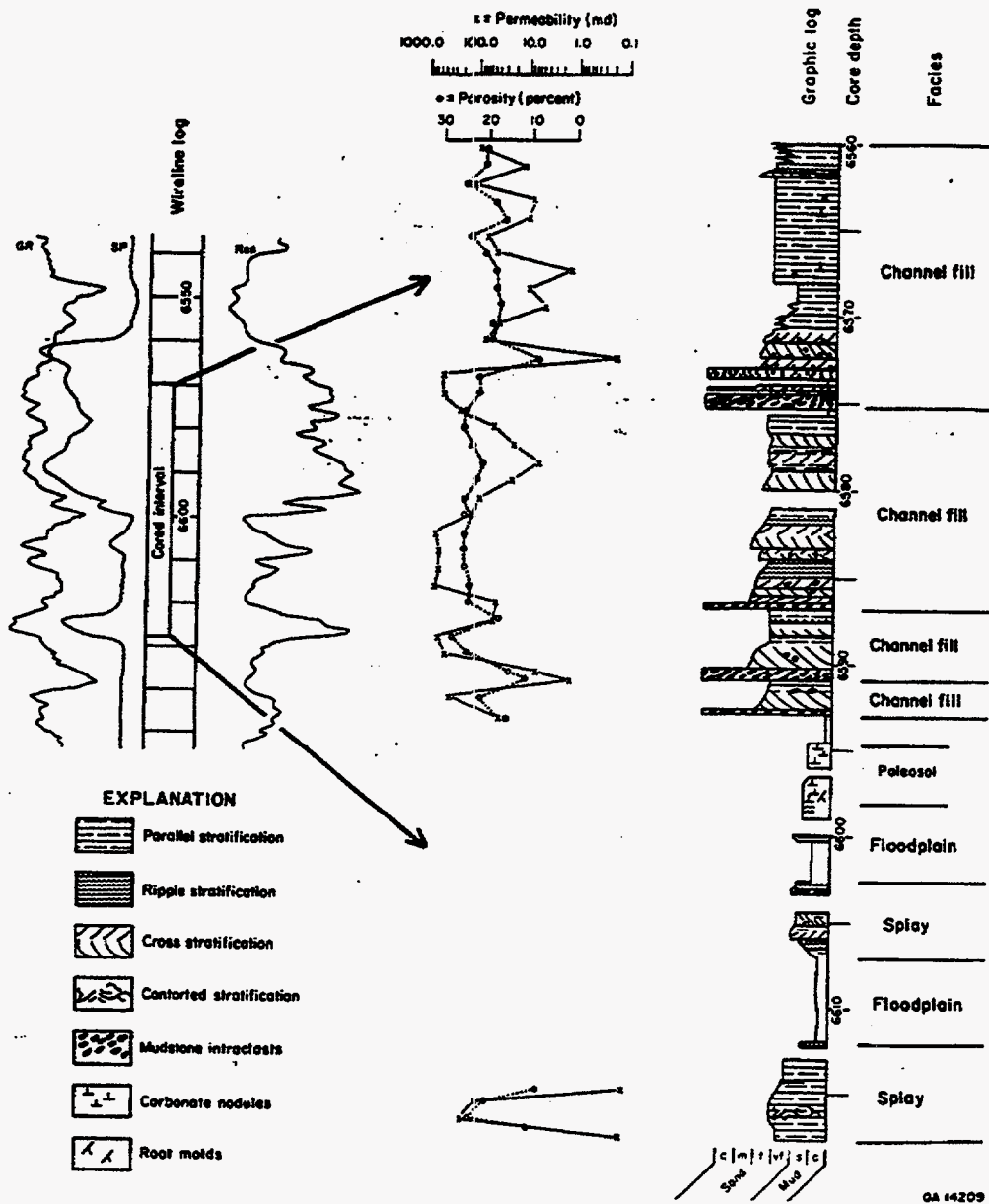
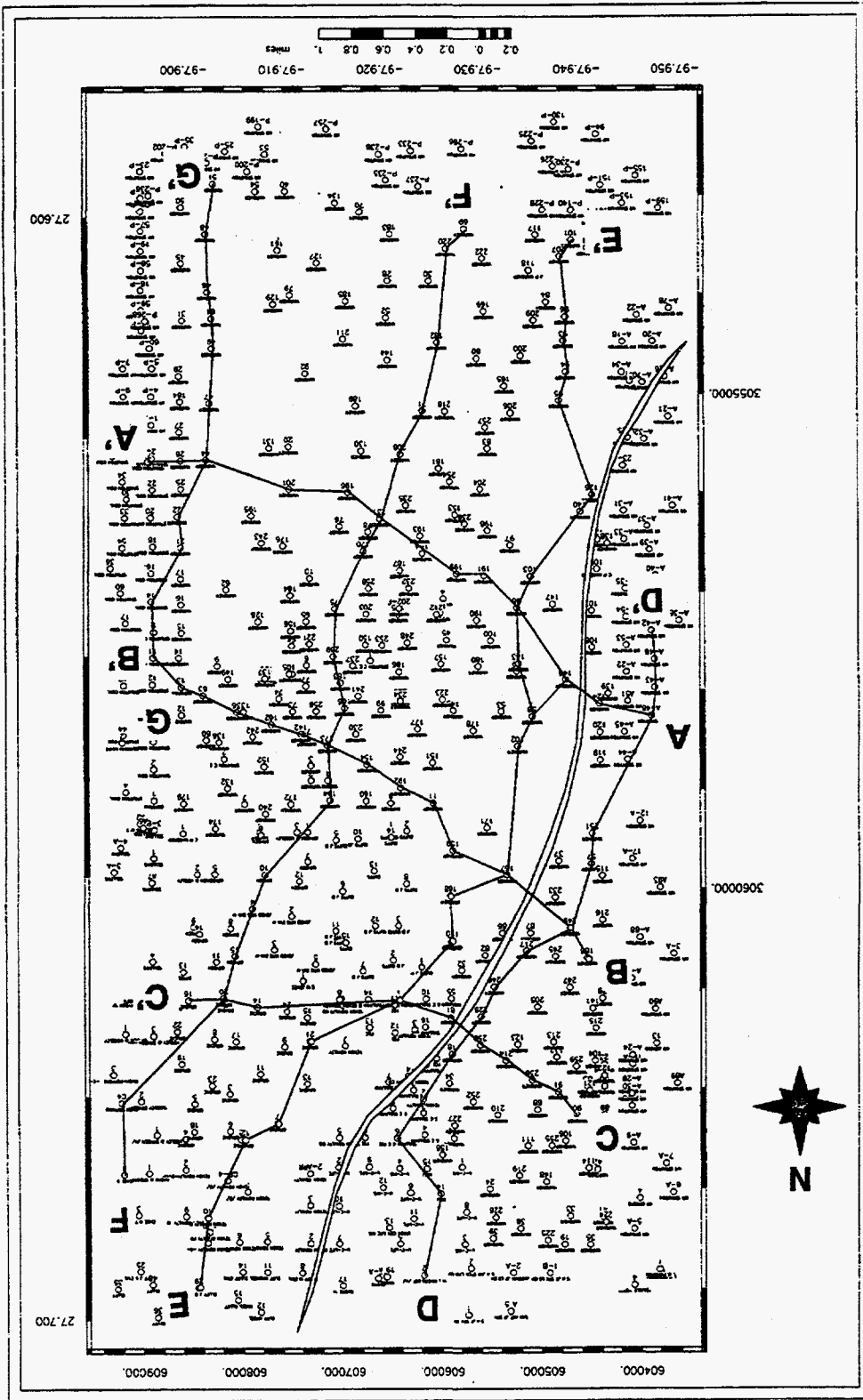


Fig. 3.9. Facies interpretation on cores cut through part of the middle Frio in Wardner 184 well, Stratton field (from Kerr and Jirik, 1990).

Fig. 3.10. Well location and cross section index map.



Wardner 195

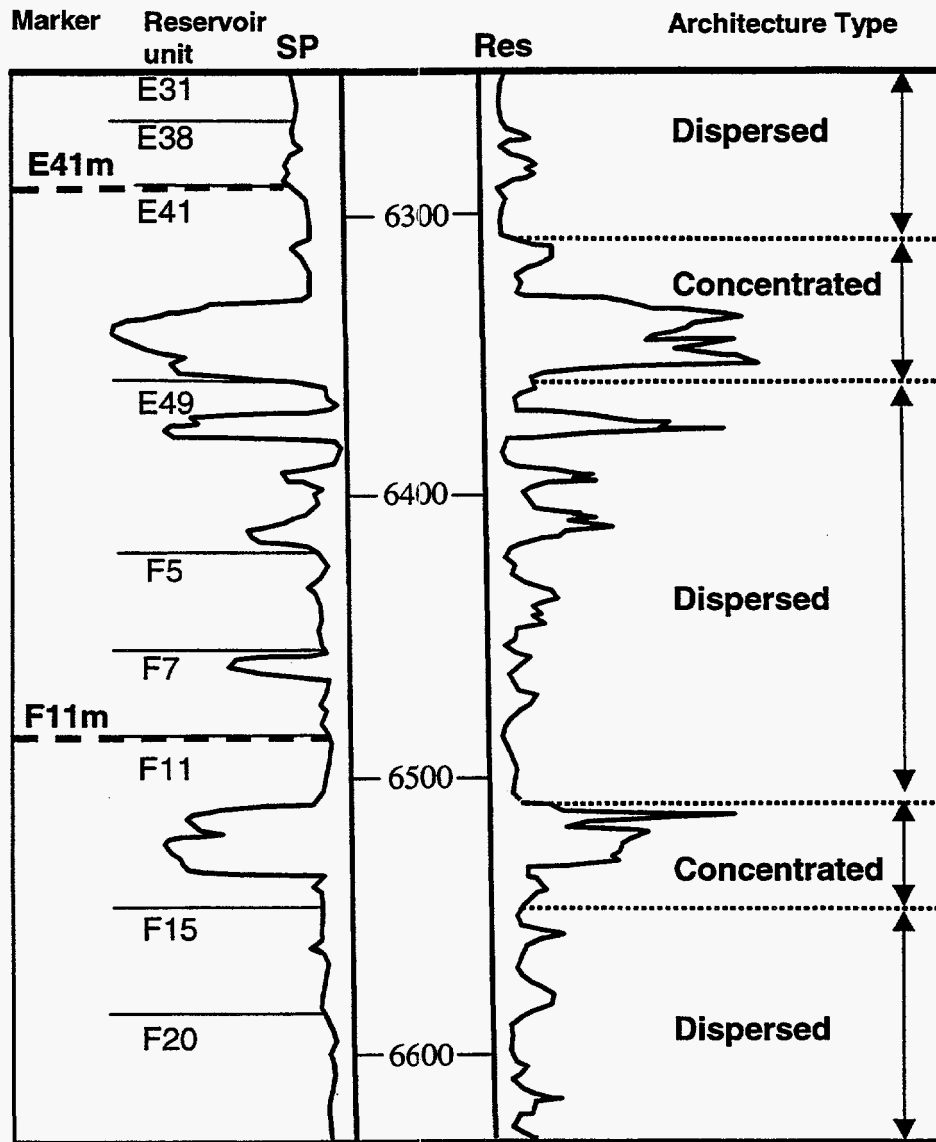


Fig. 3.11. Type well log from Wardner 195 showing reservoir subdivision, markers and facies architecture type of the study interval.

The structure maps of the two markers show that the master Vicksburg growth fault is the dominant feature of the structure framework (Fig. 3.12 and 3.13). Parallel to the master fault, a rollover anticline is developed about 1.5 mi (2.4 km) to the east in the hanging wall block. The fault throw and folding amplitude decrease upward at different stratigraphic level. The structure of F11m shows the fault throw from 150 - 350 ft (45 - 107 m) and roll over amplitude from 100 - 250 ft (30 - 76 m). Up to the level of E41m, the fault throw decreases to 80 - 230 ft (25 - 70 m) and the rollover amplitude decreases to 50 - 100 ft (15 - 30 m).

Since the growth fault was active during deposition, it has significant control over gross thickness variation at different locations with respect to the growth fault. The interval thickness between E41m and F11m illustrates the variation pattern (Fig 3.14). In the footwall block, the interval thickness is about 150 ft (46 m). As more accommodation space created by growth faulting in the hanging wall block, the thickness increases to about 200 ft (61 m). A wedge shaped space is create by the rollover effect. As a result, the interval thickens towards the growth fault and the total thickness can increase up to 250 ft (76 m).

The interval thickness variation indicates that structural subsidence resulted in increase of aggradation, which may suggest a rich sediment supply during the deposition of middle Frio Formation and the capability of the fluvial system to compensate the accommodation space created by the structure activities.

Previous reservoir studies on the middle Frio Formation have been carried out at different hierarchical levels. Based on regional stratigraphic analysis, middle Frio consists of deposits of Gueydan fluvial system (Galloway et al., 1982). Discrete genetic interval (DGI) is define and used in reservoir compartmentalization study by Kerr and Jirik (1990). Each DGI is composed of facies including channel fill, levee splay and floodplain. The interest of this study is to model the facies distribution at the level of channel belt, which is approximately the same level of DGI and equivalent to Group 8 according to the facies hierarchical scheme of Miall (1991).

Based on the established facies identification criteria (Kerr and Jirik, 1990; Kerr, et al., 1991), a field scale facies correlation is completed to analyze facies architecture and its variation. The interval of detailed study comprises reservoir units from E41 to F11. The two marker beds E41m and F11m are the essential control for stratigraphic correlation. Correlation of markers was accomplished using logs from 350 wells. Seven cross-sections through 89 wells were used to establish facies architecture. Efforts have been made to quantify facies architecture variation based on observation of the seven stratigraphic cross-sections generated across Stratton field.

The cross-section network was set up to evaluate architecture and its variation relative to the growth fault coursing through the study area. Three sections are parallel to regional depositional dip and perpendicular to the growth fault . Four sections are in the perpendicular to the growth fault . Major features of facies architecture observed include:
1) Channel-fill facies show a higher lateral

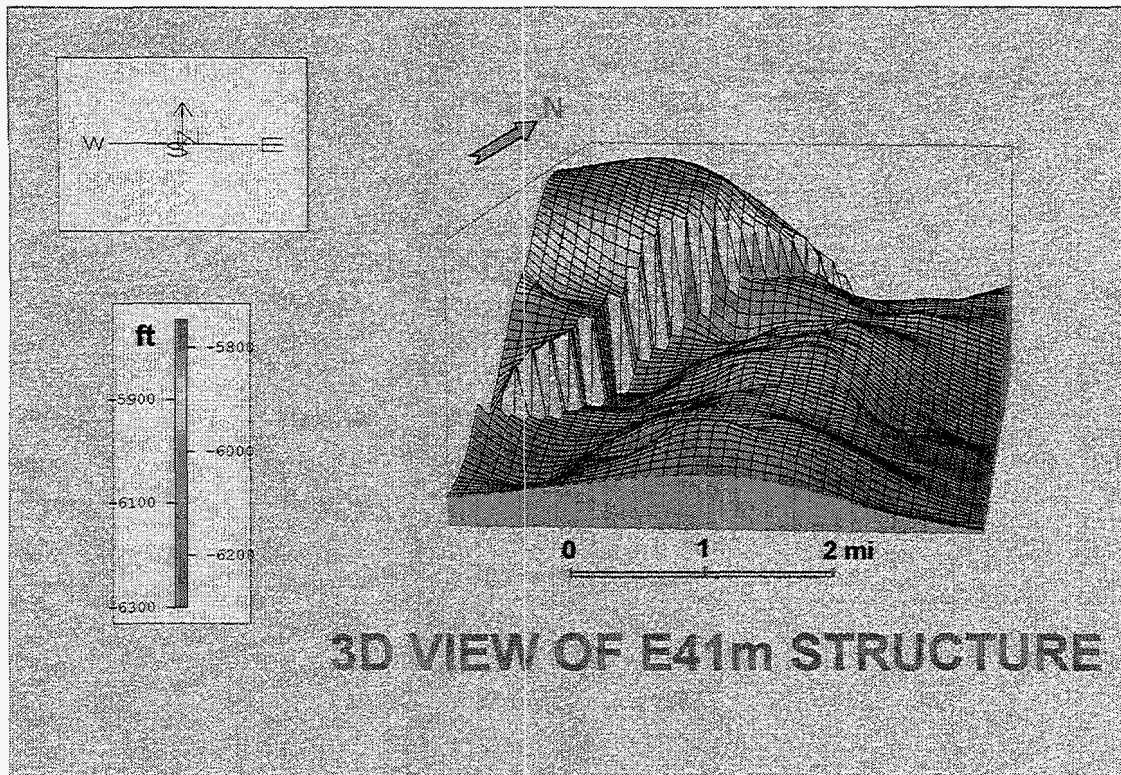


Fig. 3.12. 3D view of E41m structure.

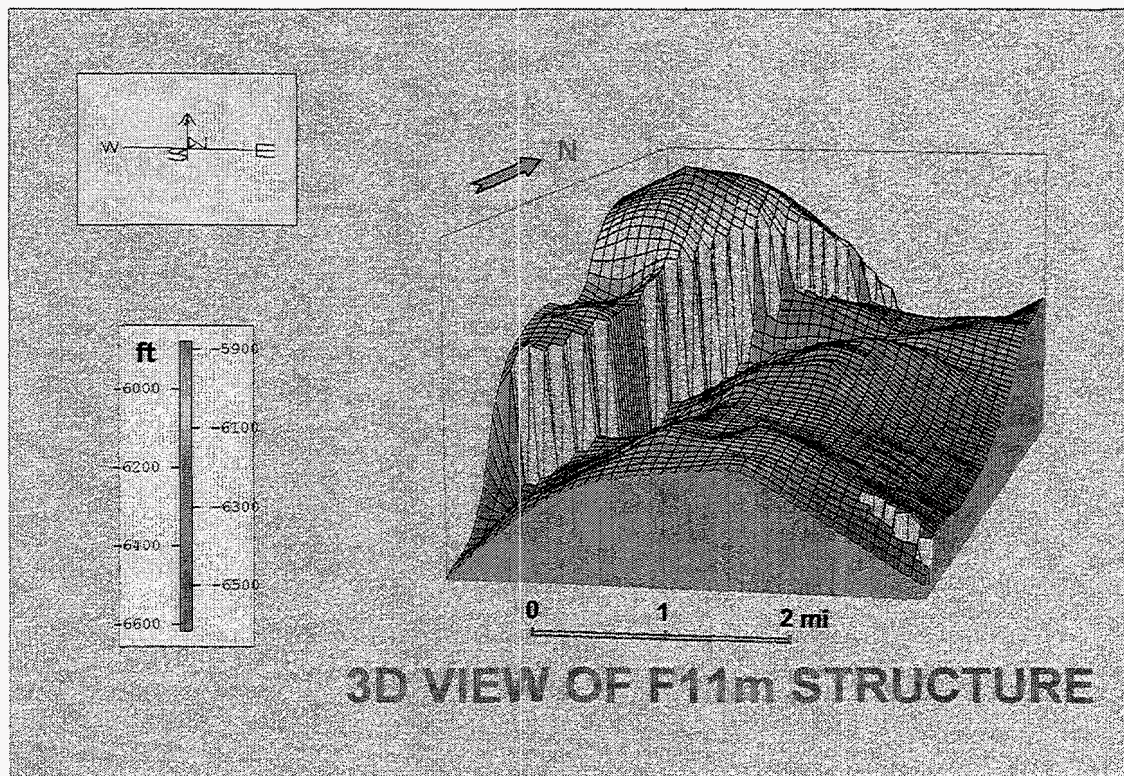


Fig. 3.13. 3D view of F11m structure.

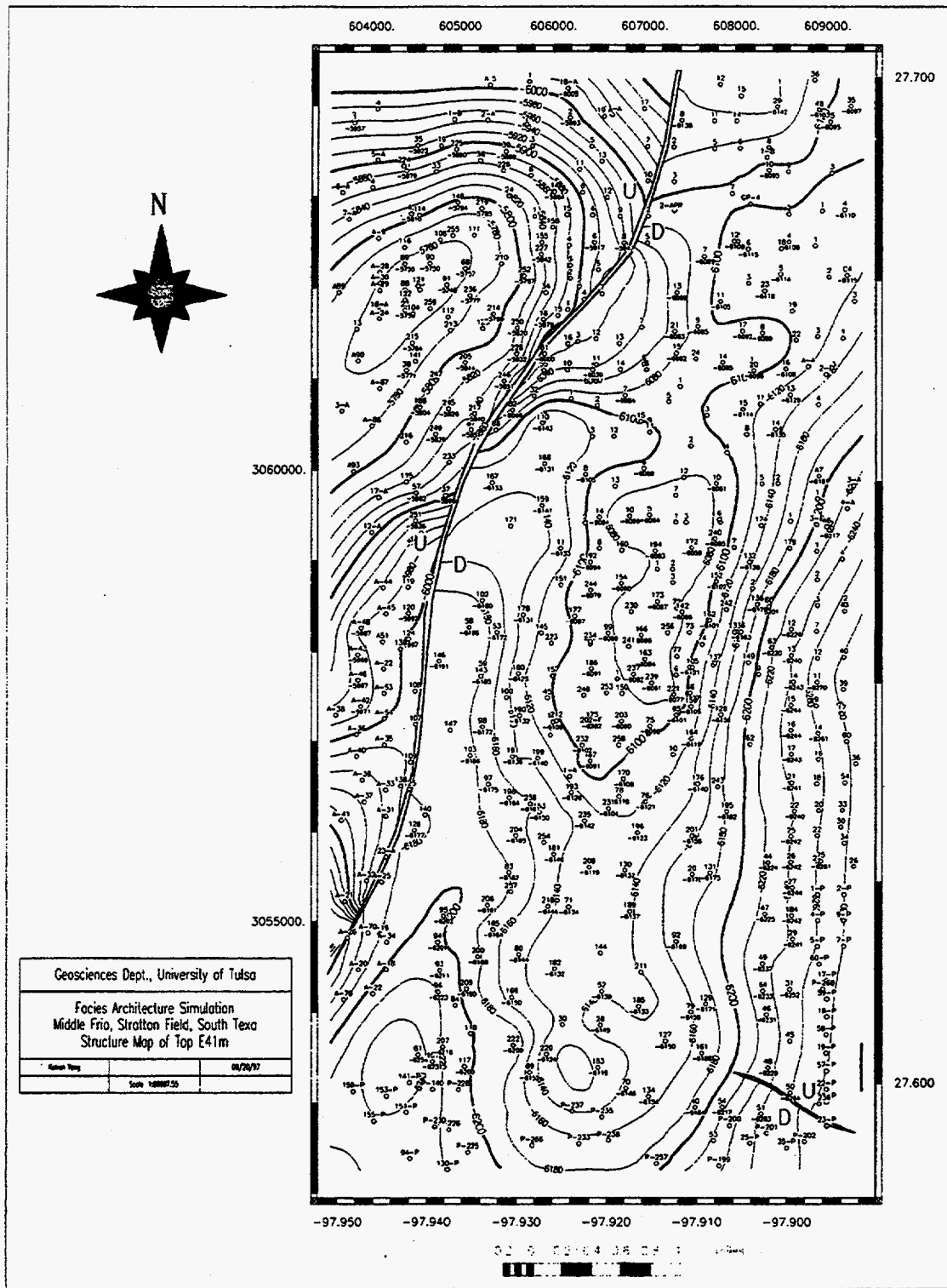


Fig. 3.14. Interval thickness map between E41m and F11m.

continuity in the direction of depositional dip compared to perpendicular to depositional dip. This observation is not surprising for meandering fluvial systems. 2) Channel-fill facies repeat vertically from concentrated architecture to dispersed architecture (Figure 3.15). This variation was noted earlier by Kerr and Jirik(1990) and named as lateral stacking and vertical stacking respectively. 3) Channel-fill facies in the major reservoir units such as E41 and F11 (concentrated architecture) transition from laterally to vertically stacked toward the growth fault. In the vertically stacked pattern, channel belts are gathered into clusters with relative small lateral extent and more layers overlapped vertically. While in the laterally stacked pattern, channel belts become more laterally spread out and evenly distributed. 4) The number of channel fill facies is increased on the hanging wall block side across the growth fault. This observation is more obvious in the units of dispersed architecture.

The explanation for the variation from concentrated to dispersed channel-fill facies stacking is a change of aggradation rate. As the floodplain aggradation rate increases, more fine grained material will be deposited onto the floodplain and channel-fill facies will become more dispersed (Figure 3.16 and 3.17). The increase of channel facies number across the growth fault and stacking pattern variation within the concentrated architecture unit may be caused by growth faulting activity. The events of faulting are likely to increase the probability of river avulsion. As a result, nodal avulsion may occur at the fault location and river start to deviate from the avulsion point. The effect of channel deviation may have caused more even distribution of channel facies laterally.

Preliminary quantitative characterization of facies architecture is made based on three cross-sections (D-D', E-E' and F-F') parallel to the growth fault. The objective of this study is to characterize facies architecture variation both vertically and laterally. The study interval can be vertically classified into units with architecture of concentrated channel-fill (as E41 and F11) and units with architecture of dispersed channel-fill (as from E49 – F7). Since the three cross sections have different length and cut through different portion of the field, standardization was made in calculating architecture parameters to get comparable results. Cross section E-E' has the most extensive coverage and is chosen as the standard cross section. Cross sections D-D' and F-F' are artificially extended to have the same projected length and coverage. Architecture parameter values calculated based on the actual cross sections need to be adjusted after the artificial extension. This adjustment is made for each extended cross section accordingly in considering extended length of the cross section and facies distribution across the field. Statistics show difference between these two types of architecture.

Concentrated architecture has a channel belt to gross thickness ratio ranging from 0.54 to 0.60. Dispersed stacking architecture has a ratio ranging from 0.14 to 0.18 (Table 1 and Fig. 3.16, numbers used are after standardization). The increase in number of channel-fill facies across the growth fault can be estimated on the cross sections. In the unit of concentrated architecture, the standardized number of channel fill layers based on interpretation is 17 in section D-D', 26 in section E-E' and 25 in F-F'. While in the unit of dispersed architecture, the standardized number of channel fill layer is 14 in section D-

D', and 22 in section E-E' and F-F' (Fig. 3.16). In concentrated architecture, channel belts are gathered in to clusters with small lateral extent at the upstream location close to the

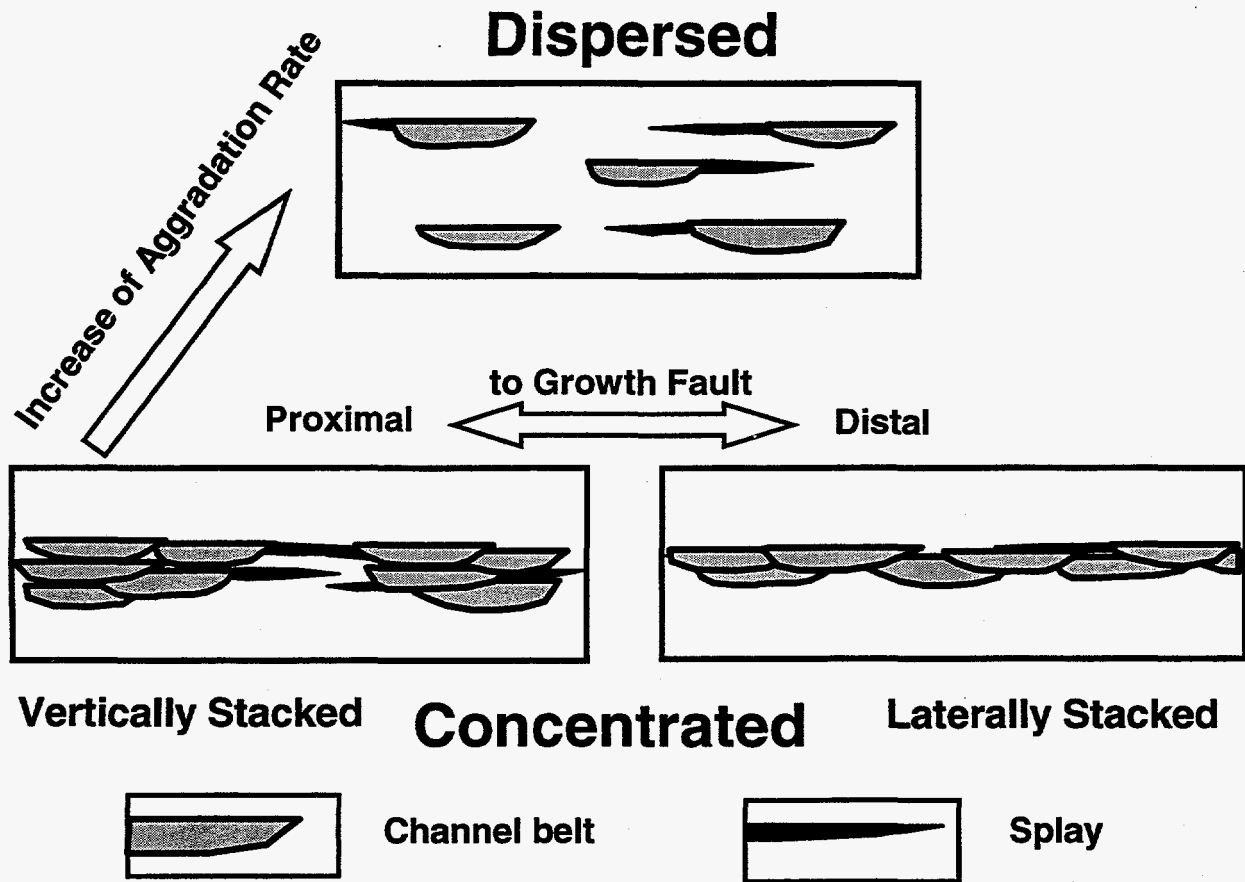


Fig. 3.15. Facies architecture and controlling factors of middle Frio Formation (modified from Kerr and Jirik, 1990).

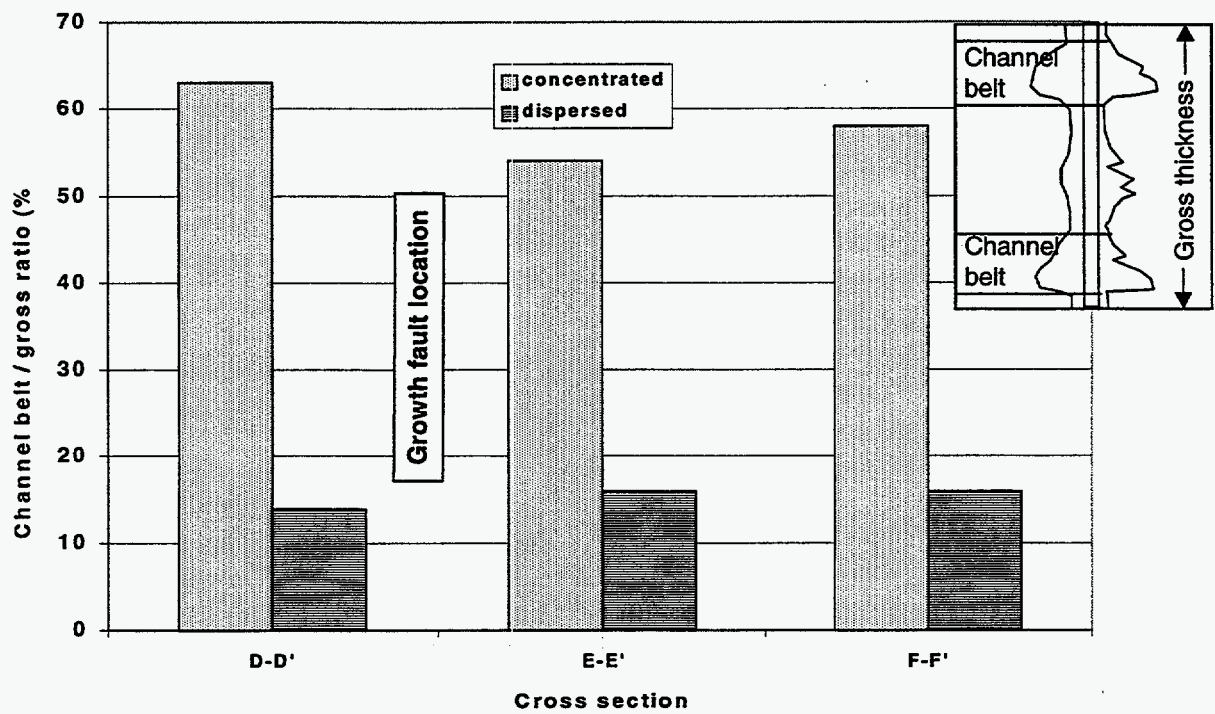


Fig. 3.16. Channel belt to gross thickness ratio in different type of facies architecture. The definition of channel belt and gross thickness is shown in the diagram at the upper right.

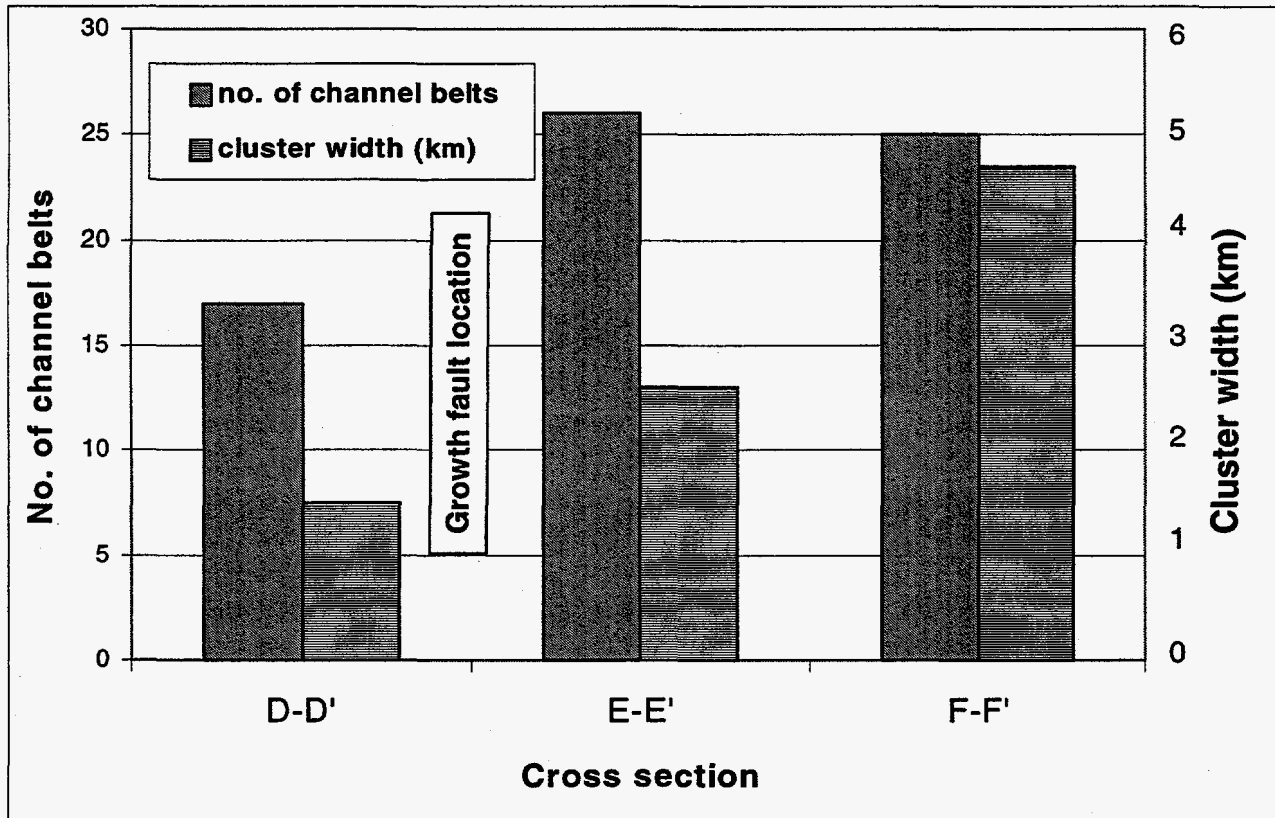


Fig. 3.17. Histogram showing number of channel belts and channel belt cluster width variation with cross section location.

growth fault and become spread out at more downstream locations (Table 1 and Plate 4, 5, 6 and 7). Channel facies connectedness is high and has an increasing trend in downstream direction. Channel belts are usually low to unconnected in the dispersed architecture. However, relatively high connectedness is observed in cross section E-E'. This is probably because that it is closer to the growth fault where nodal avulsion occurs. Channel belts are more likely to be connected.

Based on measurements from the unpublished facies mapping by Kerr (1989-1990), the width of single story channel fill sandbody is usually 2000 to 3600 ft (600 - 1100 m), averaging 2700 ft (820 m) (Fig. 3.22). The single story thickness measurement from the cross sections is usually 10 to 20 ft (3 to 6 m), averaging 16 ft (4.9 m). However, within the concentrated stacking architecture, channel belt sands may be stacked up to miles wide and over 50 ft (15 m) thick.

Simulation Model and Program

Various simulation models of general alluvial stratigraphy have been developed to understand and predict the spatial distribution of channel-belt sandstone bodies in alluvial succession. Of these approaches, process-based models, which are based on the simulation of geological processes, have the best potential to predict realistically the architecture of alluvial stratigraphy (e.g., Leeder 1978; Allen 1978, 1979; Bridge and Leeder 1979; Bridge and Mackey, 1993; Mackey and Bridge, 1992; 1995).

A comprehensive two-dimensional alluvial stratigraphic model was developed by Bridge and Leeder (1979). Geological effects considered in this model include channel-belt dimension, floodplain width, variable floodplain aggradation across channel valley, channel avulsion and tectonism. Bridge and Mackey modified this two-dimensional model to simulate alluvial process more accurately and to predict more aspects of alluvial architecture (Bridge and Mackey, 1993).

A process-based three-dimensional model of alluvial stratigraphy was published by Mackey and Bridge (1995). The three-dimensional model made it possible to simulate down-valley variation in alluvial architecture in response to tilting of the floodplain due to faulting and realistically simulate channel avulsion.

To simulate the specific features of the geological processes involved in the case of the middle Frio Formation, the model is modified from the early models and is described below.

Table 1. Facies architecture characterization based on cross sections.

cross section	D-D'		E-E'		F-F'	
number of wells	14		21		17	
actual length(km)	8.2		13		11.4	
projected length(km)	7.2		11.3		10.1	
location to fault(km)	0.3-1.1		0.5-1.3		2.9-3.5	
architecture	concentr	dispers.	concentr	dispers.	concentr.	dispers.
gross thickness(m)	11.5	28.9	17.7	45.1	12.5	40
actual c/g ratio (%)	60	11	54	18	58	16
std c/g ratio(%)	60	14	54	18	58	16
actual no. of chnls	11	8	26	22	22	20
std no. of chnls	17	14	26	22	25	22
stacking pattern	vertical	scattered	vertical	scattered	lateral	scattered
connectedness	high	unconnected	high	low	high	very low
chnl cluster width(km)	1.1-1.9, ave. 1.5		2.4-3.2, ave.2.6		3.7-5.6, ave.4.7	
cluster distance(km)	1.8-2.2, ave.1.9		3.2-4, ave. 3.5		5.6	
no. of chnl in cluster	2-4, ave. 3		6-7, ave. 6	<3	8-14, ave. 11	<2
no. of lyr stacked	2-3, ave. 2.5	1	3-6, ave. 4.2	<2	1-3, ave. 2	1

c/g ratio: channel belt to gross thickness ratio.

Std: values after adjustment of standardization.

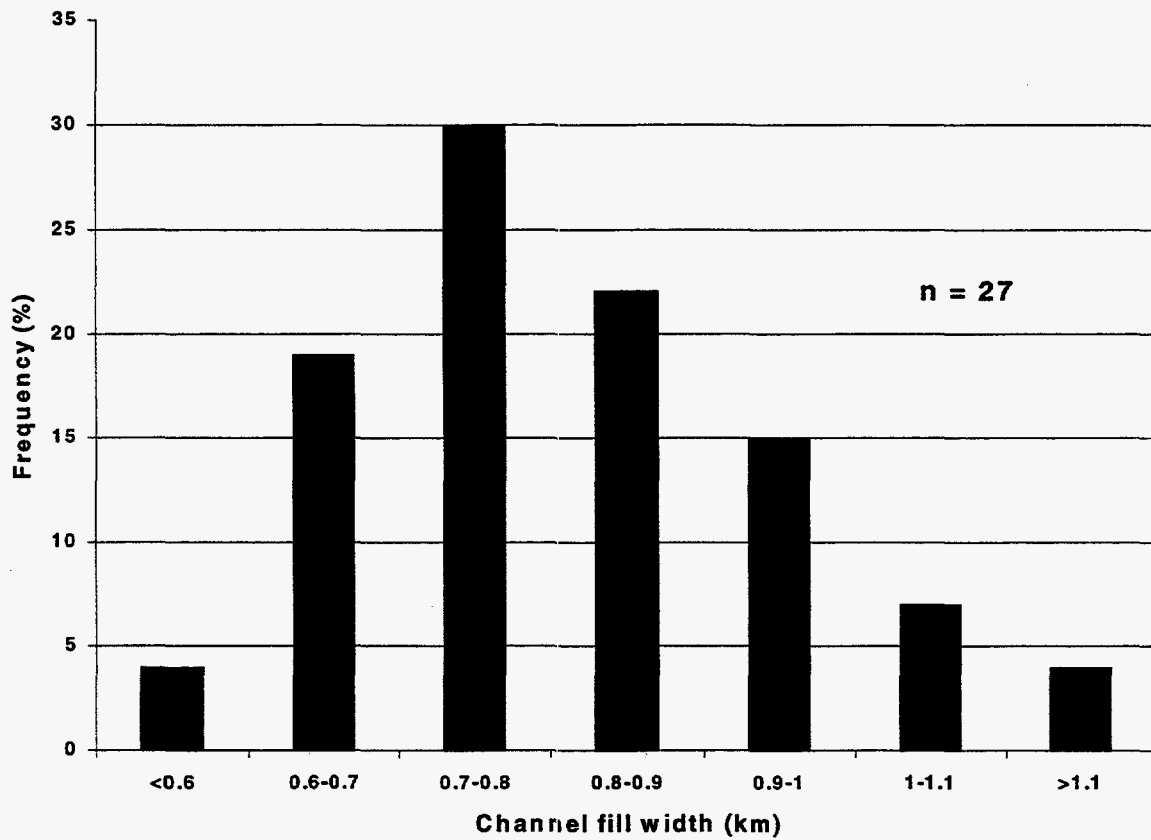


Fig. 3.18. Channel belt width distribution. Data is sampled from facies maps of unpublished study by Dennis Kerr (1989).

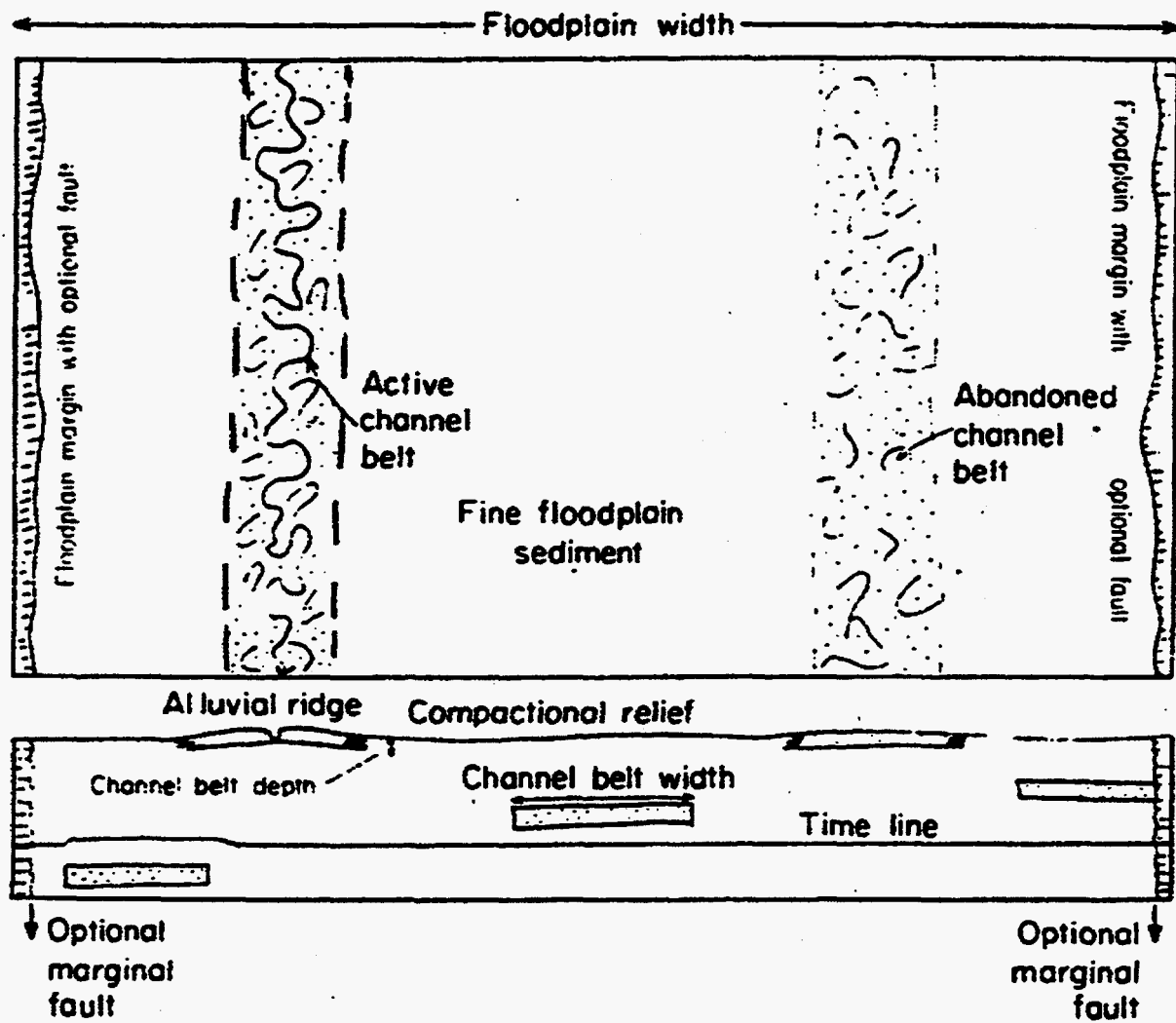


Fig. 3.19. Floodplain configuration (from Bridge and Leeder, 1979).

Description of the Simulation Model

Floodplain. In the model, the floodplain (Fig. 3.19) is defined as having finite width, confined by obstacles to river migration such as terrace or fault scarps, and traversed by a single river which is free to migrate laterally within the channel-belt (Bridge and Leeder, 1979). Floodplain width in natural rivers is often of the order of ten times channel belt width (Fisk, 1947; Russell, 1954; Bernard et al., 1970). Within the channel belt, deposits are considered to be mainly sand whereas adjacent overbank deposits are considered to be predominantly silt and clay. Crevasse splay is included in the overbank deposits in this model.

In this study, the simulation area is a segment of a floodplain that covers Stratton field.

Initial floodplain surface. In nature, a floodplain usually has varying width and down-valley slope which is represented by an exponential or linear function in the model developed by Mackey and Bridge (1995). Since only a segment of the floodplain is to be simulated in this model, the initial floodplain surface is considered to have a rectangular shape and a constant downvalley slope (Fig. 3.20). Initial floodplain elevation is calculated by:

$$y_z = S_i(L - z) \quad (1)$$

y_z : floodplain surface elevation at down valley location z .

S_i : initial floodplain slope.

L : floodplain (simulated area) length.

Local relief can be added as user defined by center, width and amplitude. Trigonometric function is used to model the local relief surface. For example, the cross section of a down valley striking local relief can be defined as:

$$\text{Rel}_x = \{1 + \cos[2\pi(x-C)/W]\} \text{Relmax} \quad (2)$$

Rel_x : local relief magnitude at cross valley location x .

C : local relief center in x coordinate.

W : width of relief.

Relmax : maximum magnitude of local relief.

Channel-belt dimension. A single channel belt is first incised into the floodplain surface with a certain width and initial bankful depth, and then start to aggrade (see

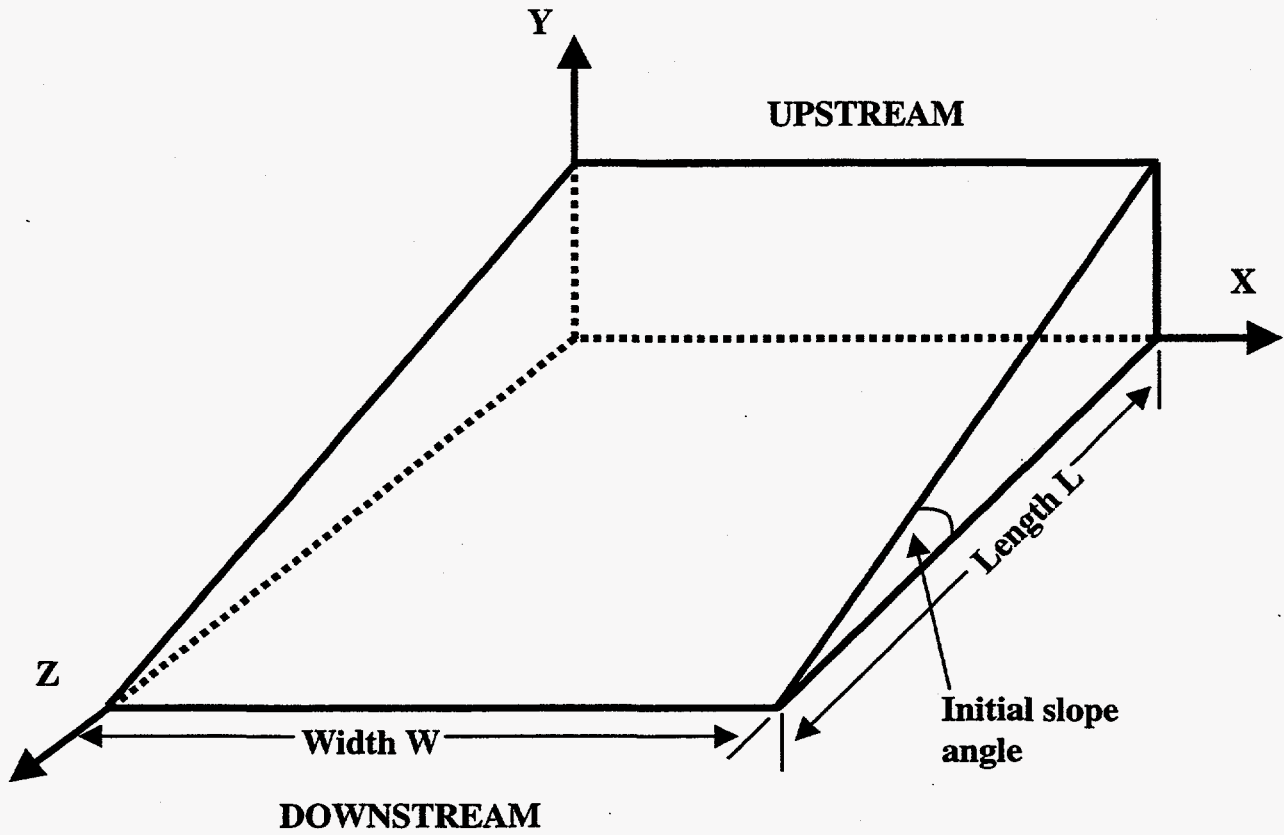


Fig. 3.20. Floodplain initialization with a constant slope. X: axis parallel to depositional strike; Z: axis parallel to depositional dip; Y: axis parallel to aggradation direction.

floodplain aggradation for more detailed discussion). The channel-belt is generated using a fixed user defined width or can have variable width sampled from a user provided distribution. Channel depth is obtained by a user given width-depth ratio. Channel-belt width and depth are constant within the simulated area at each single increment (Mackey and Bridge, 1992).

Tectonism

Tectonism (faulting, tilting and folding) can cause changes in the rate and direction of floodplain subsidence and result in floodplain topographic variation. These changes and variation in turn have significant impact on channel-belt location and distribution fluvial sediments. For example, avulsion may be the direct response to an individual tectonic event (Fisk 1944). Channel geometry may be forced to change by a change in gradient and result in incision and reduced probability of avulsion, or result in aggradation and increased probability of avulsion (Mackey and Bridge, 1995).

The major tectonic activities involved in the study area include 1) growth faulting, 2) rollover structure associated with the growth fault and 3) local variation of subsidence rate. Growth fault is the predominant tectonic event and controls the structural framework. The nature of these tectonic activities may be different and complicated. However, for convenience and simplicity of computation in the model, these activities are treated as gradual and continuous events.

The simulated interval in this study is approximately from the reservoir unit E41 to F11 in middle Frio, which is near the upper end of the growth fault. Since we are only modeling the topmost of the growth fault where it becomes fairly steep, the growth fault is approximately treated as a vertical plane striking in the cross-valley direction. The axis of the rollover structure is parallel to the growth fault at a user provided distance on the hanging wall block (see Fig. 3.21).

In the simulation model, the total tectonic subsidence rate of the simulated floodplain surface is subdivided into four elements: regional subsidence rate R_{reg} , faulting subsidence rate R_{fault} , rollover subsidence rate R_{roll} , and local subsidence rate R_{loc} . In the footwall block, the floodplain surface subsides at the basic regional rate R_{reg} . At the downstream side of the rollover axis in the hanging wall block, subsidence of the floodplain surface is the combined effect of both R_{reg} and R_{fault} . In the area between growth fault and rollover axis, in addition to R_{reg} and R_{fault} , the rollover subsidence rate R_{roll} should also be taken into consideration.

The effect of rollover is simulated by a parabolic function where the subsidence rate varies with the position relative to the rollover axis (Fig. 3.22). It can be expressed as:

$$R_{roll_z} = (R_{rollmax}/D^2)d^2 \quad (3)$$

R_{roll_z} : rollover subsidence rate at down valley location z .

$R_{rollmax}$: the maximum rollover subsidence rate near the fault.

D: the distance between the growth fault and rollover axis.

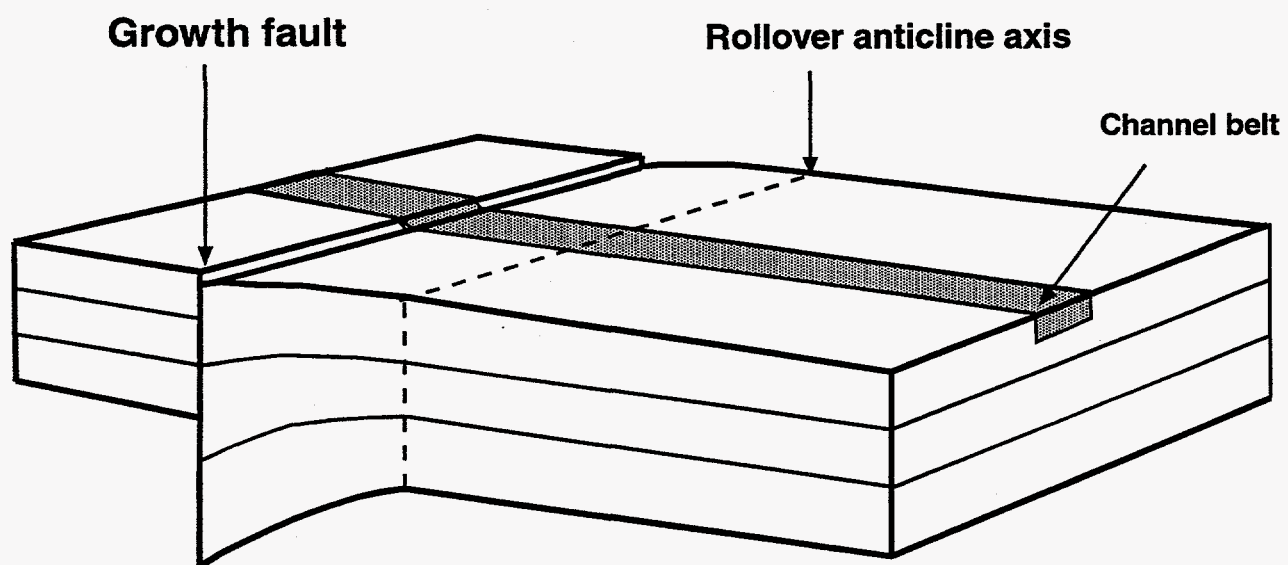


Fig. 3. 21. Tectonic sketch of the simulated area.

d: the distance from the current point to the rollover axis.

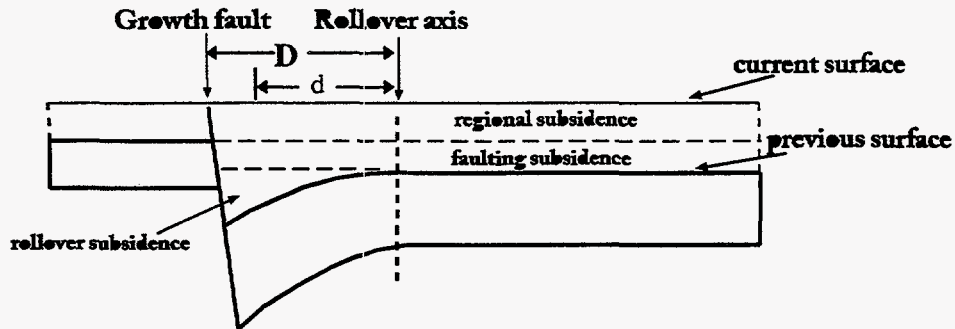


Fig. 3.22. Diagram showing tectonic subsidence elements calculated in the model.

Local structure activities can also be simulated in the model. These local variations of subsidence rate may be in different pattern and magnitude. The most important local tectonism observed in the study area that can remarkably influence the fluvial architecture is the down valley oriented depression. Trigonometric function is used to model the effect of this type (Fig. 3.23) :

$$Rloc_x = -\{1 + \cos [2\pi(x - \text{Axis})/\text{Width}]\} Rlocmax/2 \quad (4)$$

$Rloc_x$: local depression rate at cross valley location x

Axis: cross valley location of local depression axis

Width: local depression width

$Rlocmax$: maximum subsidence rate (along the axis of depression center)

Aggradation

Floodplain aggrades in the form of channel belt and overbank deposition as accommodation space created by various geological processes. Responding to subsidence rate changes, the aggradation rate varies both with time and relative position in the floodplain. The major consideration of this variation in the study area include: 1) periodic variation of aggradation rate responding to regional tectonism and baselevel fluctuation, 2) aggradation rate changes with location responding to local tectonism and 3) decrease of aggradation rate on the floodplain with increasing distance from the edge of channel belt.

Periodic variation of aggradation rate is modeled by varying the basic channel belt aggradation rate for different simulated interval. The magnitude of this variation is in the range of a few times. Based on the assumption that rich sediment supply of the fluvial

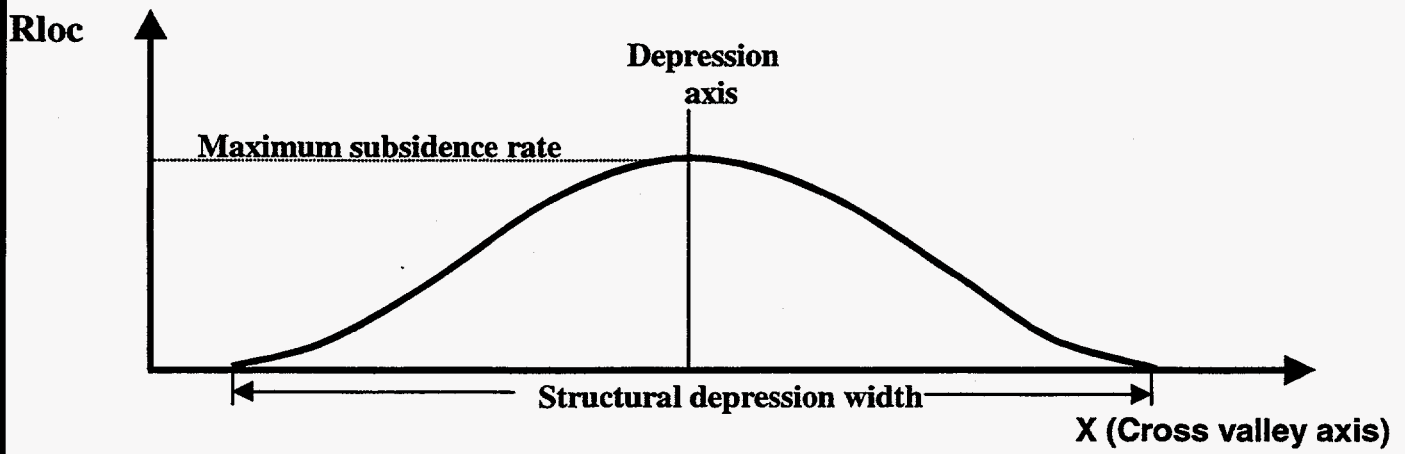


Fig.3.23. Diagram showing subsidence rate variation across a local structure depression.

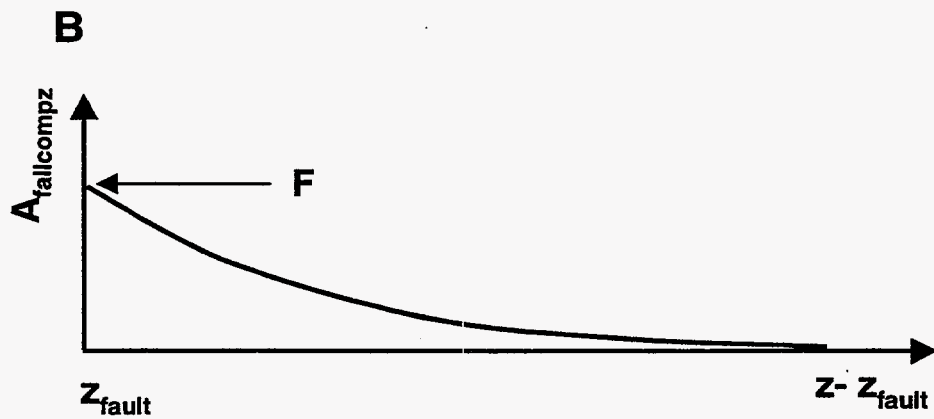
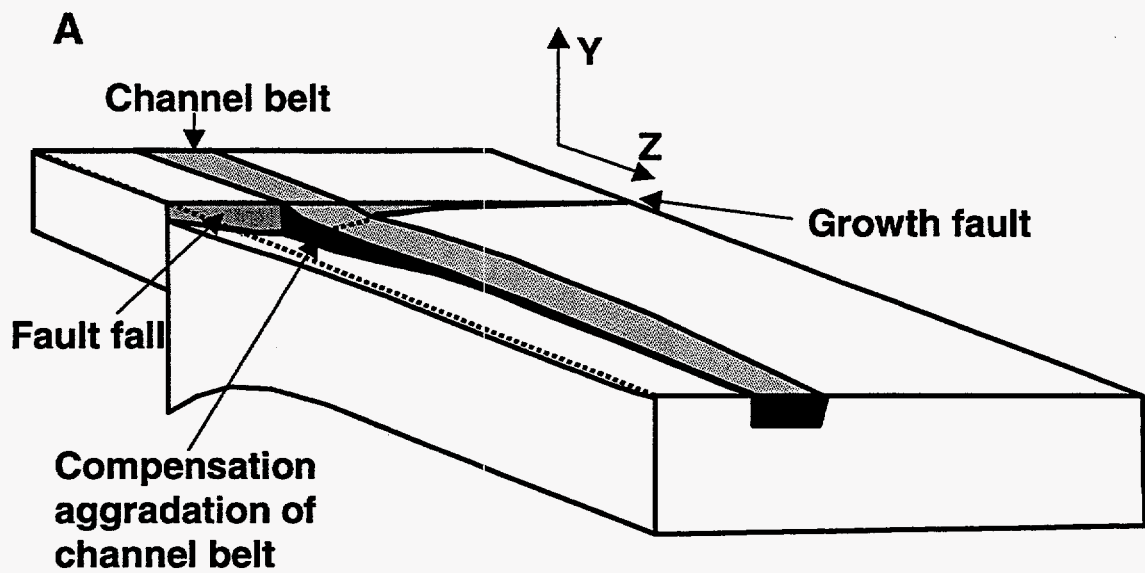


Fig. 3.24. Channel belt aggradation increase as fault fall compensation.
A: Increase of channel belt aggradation caused by fault fall.
B: Fault fall compensation aggradation decreases exponentially with distance in down valley direction. Explanation for F see equation 5.

system was capable to compensate the accommodation space created by growth fault and rollover structure, channel belt aggradation rate will be increased by the same amount as the subsidence rate caused by these local tectonism. However, at distal locations from the channel belt where less sediments are deposited and overbank aggradation rate is low, the subsidence created by growth faulting and rollover may not be fully compensated. As a result, a topographic fall will be developed across the fault. While a river coursing across the fault at such locations, the probable responses will be deeper incision on the footwall block and increased aggradation on the hanging wall block. In the model, this compensation rate is calculated by a exponential function:

$$A_{\text{fallcompz}} = F[1 + (z - z_{\text{fault}})/(L - z_{\text{fault}})]^n \quad (5)$$

$$a_{\text{fallcompz}} = A_{\text{fallcompz}} / T_{\text{avul}} \quad (6)$$

$A_{\text{fallcompz}}$: channel belt aggradation increment responding to fault fall compensation at location z

$a_{\text{fallcompz}}$: channel belt aggradation rate increment responding to fault fall compensation at location z

F : topographic fall at the fault location

x_{fault} : down valley location of the growth fault

L : length of floodplain

T_{avul} : avulsion period

Total channel belt aggradation rate can be expressed as:

$$a_z = a_{\text{basic}} + a_{\text{fault}} + a_{\text{rollz}} + a_{\text{fallcompz}} \quad (7)$$

a_z : channel belt aggradation rate at down valley location z

a_{basic} : basic channel belt aggradation rate, varying with time

a_{fault} : channel belt aggradation rate increment corresponding to growth faulting, equal to R_{fault}

a_{rollz} : channel belt aggradation rate increment corresponding to rollover structure at location z , equal to R_{rollz}

$a_{\text{fallcompz}}$: channel belt aggradation rate increment corresponding to fault fall compensation at location z

n : fault fall compensation exponential

Overbank deposition on floodplain occurs just about every year during floods. Generally, the amount and grainsize of sediments carried by and deposited from the flood water decrease with increasing distance from the edge of channel belt (Fig. 3.25). In the model, overbank aggradation rate is calculated by:

$$r_{zx} = a_z (1 + d/w)^b \quad (\text{modified from Mackey and Bridge, 1995}) \quad (8)$$

r_{zx} : overbank aggradation rate at down valley location z and cross valley location x

d : distance from edge of the channel belt

w : floodplain width

b : overbank aggradation exponent, a parameter describes the rapidity with which deposition rate decreased with increasing distance

Realistic value of b is estimated 0.5 - 1.8 for multiple flood events from Brandywine Creek, Pennsylvania (Pizzuto, 1987) and 5 - 10 for a single flood from the Mississippi River (Kesel et al., 1974).

Compaction

Compaction of all sediments starts immediately after burial. Compaction of sediments will cause reduction of porosity and consequently, reduction of gross thickness. For a given layer of recently deposited sediments of thickness T , the solid thickness is

$$T_s = \int_0^T [1 - P(y)] dy \quad (9)$$

T : layer thickness

$P(y)$: porosity at the depth y

Assuming that no mineral material is added and removed or otherwise transformed in density, the solid thickness T_s remains constant irrespective of depth of burial, which can be expressed as

$$T_s = \int_{y_1}^{y_2} [1 - P(y)] dy \quad (10)$$

where $P(y)$ is the porosity function with burial depth y . y_1 and y_2 are the burial depth of the bottom and top of the sediment layer respectively (Perrier and Quiblier, 1974).

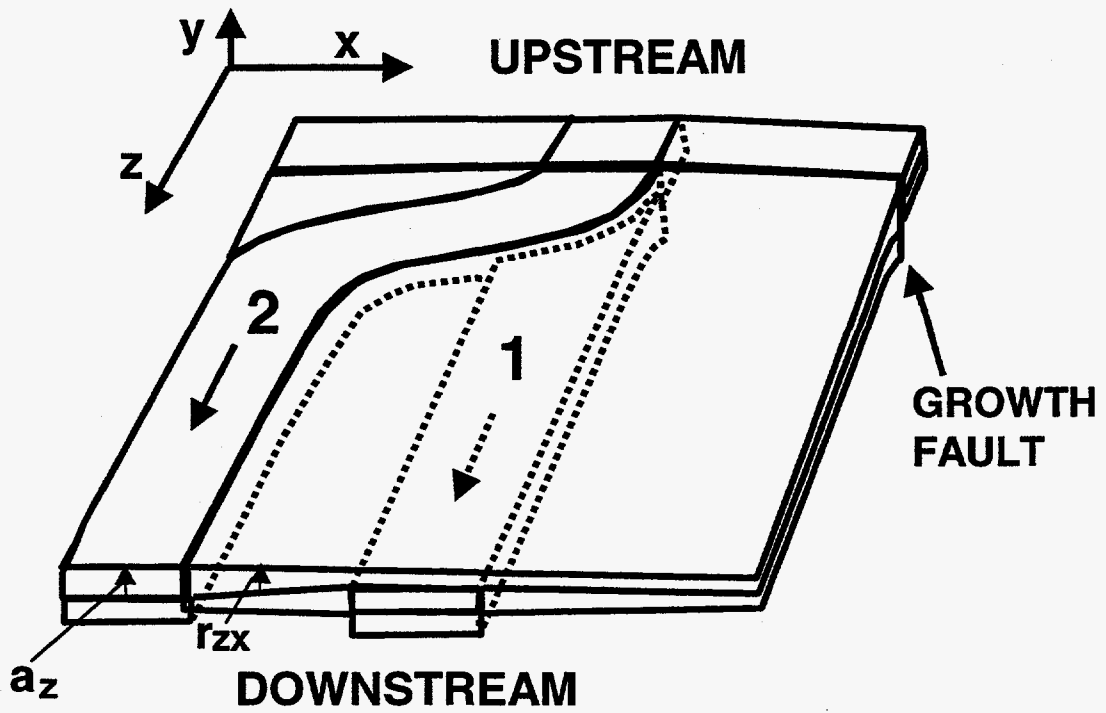


Fig. 3.25. Diagram showing floodplain aggradation (modified from Mackey and Bridge, 1995). See text and equation 7 and 8 for explanation of a_z and r_{zx} .

For mud sediments, the porosity can be calculated regarding burial depth according to the empirical function

$$P(y) = 0.50 + 0.0222 \ln(y+1) - 0.0092 [\ln(y+1)]^2 \quad (11)$$

Solid thickness can be obtained by:

$$T_s = 0.54y^{-(y+1) \ln(y+1) * [0.0406 - 0.0092 \ln(y+1)]} \Big|_{y_1}^{y_2} \quad (12)$$

(Mackey and Bridge, 1992).

The porosity-depth relationship for sandstone can be represented as:

$$P(y) = 0.49e^{-y/3700} \quad (\text{Baldwin and Butler, 1985}) \quad (13)$$

The solid thickness for a given sand layer can be expressed as

$$T_s = y + 1813e^{-y/3700} \Big|_{y_1}^{y_2} \quad (14)$$

T_s : solid thickness

Thus, with known T_s and the depth of one layer boundary, the depth of the other boundary can be obtained.

Avulsion

Avulsion is defined as the process where a channel-belt suddenly changes its course. This process is recorded in geological history by abandoned channel-belt preserved in floodplain. In nature, the mean period of avulsion is usually of the order of $10^1 - 10^3$ years, often about 10^3 years (Bridge and Leeder, 1979; Mackey and Bridge, 1995). In the models developed by Bridge and Leeder (1979), and Mackey and Bridge (1992), it is considered that the probability of occurrence of avulsion as a function of at least 1) relative elevation of alluvial ridge and floodplain, and 2) a measure of the severity and persistence of the seasonal flood. However, in Stratton area during the deposition of middle Frio, growth faulting perpendicular in cross valley direction may significantly increase the probability of river avulsion and diversion at the faulting location.

All of these factors are time dependent, avulsion would become more likely as time progressed (Bridge and Leeder, 1979). As quantitative description of this process in this models, avulsion recurrence times are obtained by Monte Carlo sampling from a presumed distribution. Considering its time dependent nature, a Weibull distribution is used to model this process (Bridge and Leeder, 1979).

Three types of avulsion are incorporated in the simulation model: regional avulsion, upperstream avulsion and nodal avulsion. The regional avulsion occurs at very upstream location and causes regional shift of channel belt from one part of the floodplain to another. It is observed in the nature that each of this regional avulsion may followed by a series of more downstream avulsion (Fig. 3.26). This process is described by Mackey

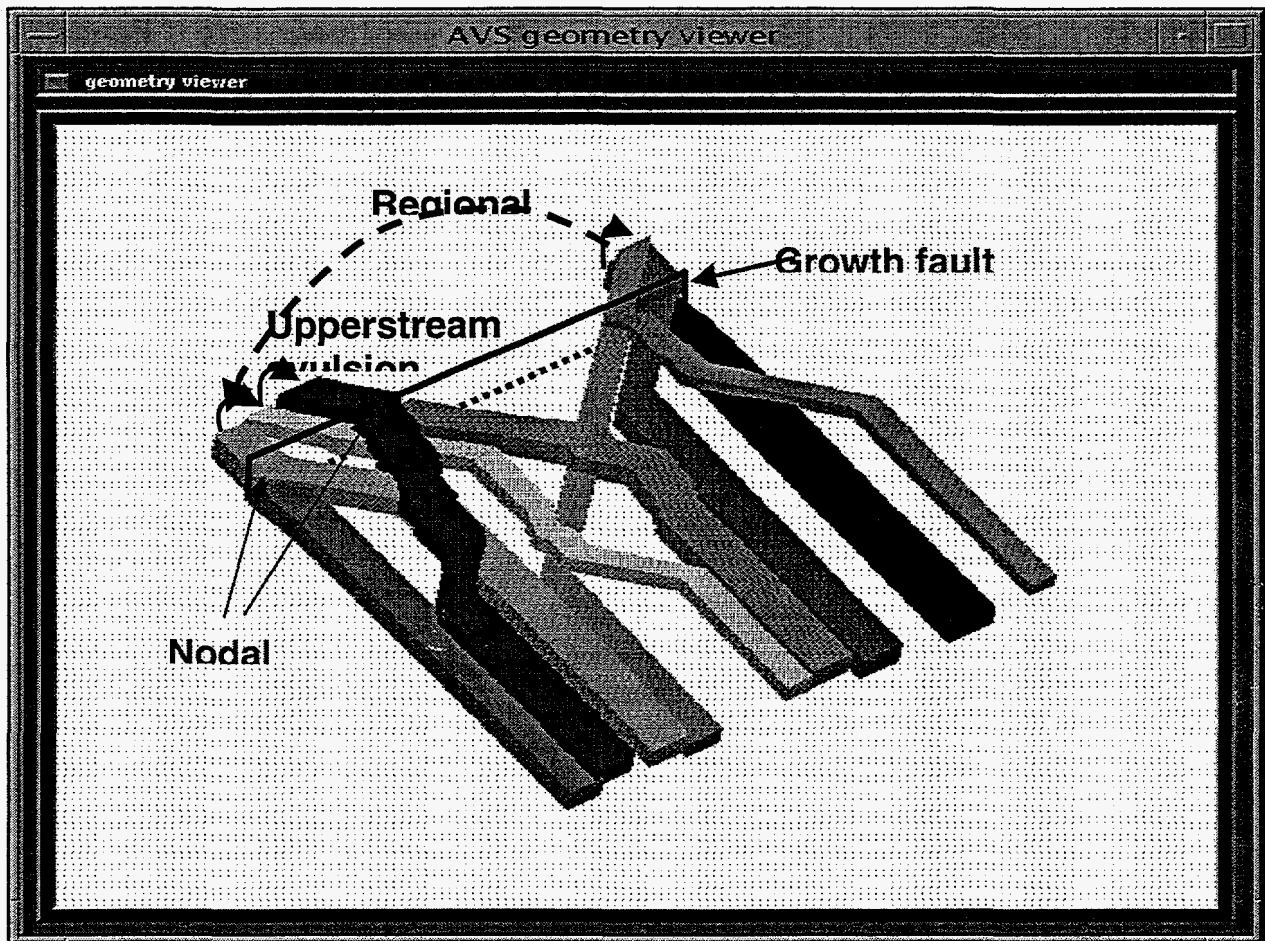


Fig. 3.26. Avulsion hierarchy in the model. Three types of avulsion are incorporated in a hierarchical order. See text for explanation of avulsion styles.

and Bridge (1995 as avulsion sequence). The term upperstream avulsion refers to the more localized avulsions following a regional avulsion and occurs at a upperstream location from the simulated area. The avulsions occurred at more downstream location are not simulated in the model. Nodal avulsions are considered to be caused by the activity of growth faulting.

In the case of regional avulsion, it is assumed that avulsion occurs far enough upstream to allow free migration of the channel to enter the first upstream section at position of the lowest elevation of floodplain surface. Each regional avulsion is followed by one or more nodal avulsions and the number is calculated by Monte Carlo sampling from a presumed distribution. The location at which the river enters the first upstream transection of the simulated floodplain segment after a upperstream avulsion is deviated from the course of preceding regional avulsion and somewhat restricted by the regional avulsion. In the simulation model, this offset at the first upstream transection is calculated by uniformly sampling from a given range, usually about one third to one half of the floodplain width. Nodal avulsion is considered to be caused by growth faulting and occurs at the faulting location. The number of nodal avulsion following each upperstream avulsion a upstream avulsion is also sampled from a presumed distribution.

The initial phase of avulsion in terms of spatial and temporal changes is complicated and poorly understood. However, for convenience of calculation, it is generally assumed that immediately after the avulsion, the river will erode its course down to the initial depth and fill by coarse material up to the banks, and the new course will follow the direction of maximum slope.

Program Design and Implementation

The simulation program is written to have high readability, portability and can be easily modified. Each major geological process is coded into individual functions and the lower order processes are nested into higher order processes. User input parameters can be imported from a data file or interactively typed in. The computer program is written in C++ Language and runs on SUN workstations (Appendix). Two computer graphic programs are also written to visualize the simulation results. One is used to render the 3D realization volume and the other is to create cross section images at chosen locations (Appendix). The underlying graphic interface for both the visualization programs is OpenGL and the driving language is C.

The simulation is started with initialization of the floodplain. After the initial floodplain is initialized, a loop is used to control the maximum number of increments to be generated in the particular realization. Avulsion period is the basic calculation time step and a genetic stratigraphic increment is generated during each avulsion period. Within each of the basic time step, a cycle is executed in the following order: 1) calculation of the current avulsion period, type and location, 2) calculation of channel belt dimension, 3) placement of the channel belt onto the current floodplain surface, 4) calculation of tectonism and adjusting floodplain surface, 5) aggradation of channel belt and overbank and 6) compaction of all the sediments previously deposited. This cycle is repeated until the desired number of increments is reached. After the realization is

generated, the results will be exported into output files for computer graphic imaging. Architecture parameters are then calculated based on the realization (Appendix).

Algorithm, input and output

The function InitialFPS is used to initialize floodplain surface (Appendix). Input parameters needed to define the size, slope and local relief of the floodplain include floodplain length (LFP), floodplain width (WFP), initial slope (SlopeIni), local depression center (Lcenter), width (Lwidth) and depth (Ldepth). A subfunction LocalRelief is called within the initialization function to calculate local topographic relief. The floodplain is subdivided into equal sized gridblocks. Gridblock size is defined by the interval length in both X (IntervX) and Z(IntervZ) direction. A three dimensional array FPS[x][z][increment] is used to store the elevation of every increment top surface at every grid point. After initialization, elevation of floodplain surface is calculated at each grid point and stored as FPS[x][z][0].

Each avulsion period is calculated by sampling from a predefined Weibull distribution. Expected avulsion period (ExpeAvul) and the exponential factor (AvulExpo) are required to define this distribution. The number of upperstream avulsions following a regional avulsion, number of nodal avulsions following a upperstream avulsion and the offset of the current upperstream avulsion from preceding regional avulsion at the location of first entering the simulated area are calculated by sampling from predefined distributions (normal distribution or uniform distribution). In case of normal distribution, a mean value, maximum value, minimum value and standard deviation are provided to define it. While in case of uniform distribution, a range is needed. Before each regional avulsion, the subroutine LowFPS is called to calculate the location of the lowest elevation point on the floodplain along the first upstream transection. The river will then enter the simulated area at this point (Appendix).

For each avulsion, the function ChannelDimension will be called to calculate the channel belt width by sampling a given normal distribution. This distribution is defined by user input mean (WcMean), maximum (WcMax), minimum (WcMin) and standard deviation (WcStdv) values. Channel belt depth (Dc) is obtained by a width/depth ratio (WDRatio). An array is used Wc[increment] is used to record the channel belt width of each increment.

After the channel dimension is calculated, the function PlaceChannel will place the channel belt onto the floodplain. The new channel belt follow its old course down to its new avulsion point if the avulsion point is within the simulated area, and then start to deviate. At each cross valley transection location, the function ChanErosion will enable the channel belt to erode the floodplain down to its initial depth (Dc) and the function PointSlope will calculate the floodplain slope at different directions and chose the maximum slope as channel belt flow direction. The left boundary and right boundary of the channel belt at each transection in each increment will be assigned respectively into two two dimensional arrays named ChanLeft[increment][transection] and ChanRight[increment][transection] to record its location.

A three dimensional array `Lith[x][z][increment]` is used to record the lithology for each increment at each grid point location. Its values are assigned by functions `Lithology` and `ChannelErosion`. The value 0 represents overbank facies, 1 represents channel fill facies, 2 represents partially eroded channel fill facies and 3 represent totally eroded overbank or channel fill facies (Appendix).

The function `Tectonism` will calculate different tectonism subsidence rates at each grid point location on the floodplain and adjust the elevation at each location. Input parameters needed to define tectonism include growth fault location in terms of distance from the first upstream transection (`FaultDist`), faulting rate (`Rfault`), rollover axis location (`AxisDist`), maximum rollover rate (`Rrm`), local depression center (`Dcenter`), depression width (`Dwidth`) and maximum depression rate at the center (`Drate`). Rollover folding subsidence rate is calculated by a parabolic function and local depression is defined by a trigonometric function. The total tectonism subsidence during each increment is the product of total subsidence rate and avulsion time period. The elevation of the current floodplain surface and all previous increment surfaces are adjusted by reassigning the array `FPS[x][z][increment]` according to total subsidence (Appendix).

The process of floodplain aggradation is modeled by the function `Aggradation`. The channel belt aggradation rate is calculated prior to overbank aggradation rate. The channel belt is first filled up to the current floodplain surface and start to aggrade upward. The total channel belt aggradation rate is the sum of the user given basic aggradation rate (`Rbasic`) and aggradation rates increments responding tectonic activities. These increments include faulting compensation, rollover compensation and fault fall compensation. The increment of aggradation rate corresponding to faulting compensation and rollover compensation equal the tectonic subsidence rate respectively. The fault fall compensation rate is calculated by function `FaultFallComp` which first determines the elevation difference between footwall block and hangingwall block at the fault location along channel belt course, and then calculate the compensation rate using a exponential function (6). After the channel belt aggradation rate is determined at each location along its course, overbank aggradation rate can be obtained by using function (8).

Total aggradation is the product of total aggradation rate and the avulsion period. Again, elevation of the current floodplain surface will be adjusted by reassign `FPS[x][z][increment]` array according to total aggradation (Appendix).

The compaction procedure used in this program is basically modified from Mackey and Bridge (1992) and translated from FORTRAN into C++. The purpose of the function `Compaction` is to compact the whole sediment volume and respectively adjust each increment surfaces by reassign array `FPS[x][y][increment]` according to the compaction result.

Right after each new increment is deposited, its solid thickness is calculated at every grid point location and recorded in the array `SolidT[x][z][increment]` by functions `SandSolid` and `ShaleSolid`. This solid thickness will remain constant unless it is eroded by a later channel belt. During the burial history, if we the solid thickness, lithology and burial depth of a given layer, we can restore its real thickness by adding back the porosity

calculated according to the given porosity - depth relationship. This work is done by functions SandComp and ShaleComp and Newton-Raphson method is used to solve equation (12) and (14). Parameters needed to define porosity - depth relationship are hard coded (Appendix).

The numerical results of realization are exported into two external files named "results" and "cross" by function Output. These two files are later used as input to graphic program to generate 3D visualization of facies architecture and 2D cross-sections respectively.

Simulated architecture parameters including number of channel belts, channel belt to gross thickness ratio, channel belt connectedness ratio and other parameters are calculated by function ArchParaCal at each chosen cross section and output by function Display.

Since channel belt start to deviate at the avulsion point, a new channel belt is generated in the area down stream to the avulsion point after each avulsion. At the area upstream to the avulsion point, the river will remain in the same course and no new channel belt is formed. As a result, the number of channel belt may vary from one cross section location to another. Channel belt connectedness ratio is calculated by using a connectedness indicator correspondent to each channel belt. When a new channel belt erodes into an preexisting channel belt at a given cross section location, the indicator for the new channel belt will be assigned a 1. Otherwise it will be assigned a 0. The final connectedness ratio is defined as the sum of indicator values divided by the total number of channel belts in the specific cross section (Fig. 3.27 and Appendix).

Simulation and Realizations

Considerations in Defining Input Parameters

The input parameters for simulation of middle Frio Formation facies architecture are defined according to subsurface geological study and characterization, modern sedimentation study and general knowledge of fluvial deposition. Some of the parameters are adjusted during the simulation process and finalized after a number of realizations.

Two sets of realizations are generated. One is for the concentrated stacking and the other is for dispersed stacking. The input parameters for each set of realizations are listed in Table 2.

The total number of increments to be generated in each set of realization is approximately equal to the number of channel belts observed in the stratigraphic cross sections perpendicular to depositional direction (Table 1). The simulation area covering Stratton field is a segment of floodplain and is defined to have a square shape. The exact width of floodplain for the river system that was delivering sediments to Stratton area during the deposition of middle Frio is difficult to define. For practical purpose, the width

of the simulation area is chosen as 12000 m which is about the size of Stratton field in the cross valley direction. The length of the simulation area is also 12000 m covering both sides of the major growth fault.

Channel belt width distribution is obtained from early detailed facies mapping in the scale of individual DGI by Kerr (1989-1990 unpublished maps) which shows a normal distribution with a mean value of 820 m and standard deviation of 180 m (see Chapter II for more information). The initial channel depth is calculated by a given channel belt width and depth ratio. A single storey channel fill thickness is the result of channel erosion and fill, aggradation and later compaction. Given a certain river system,

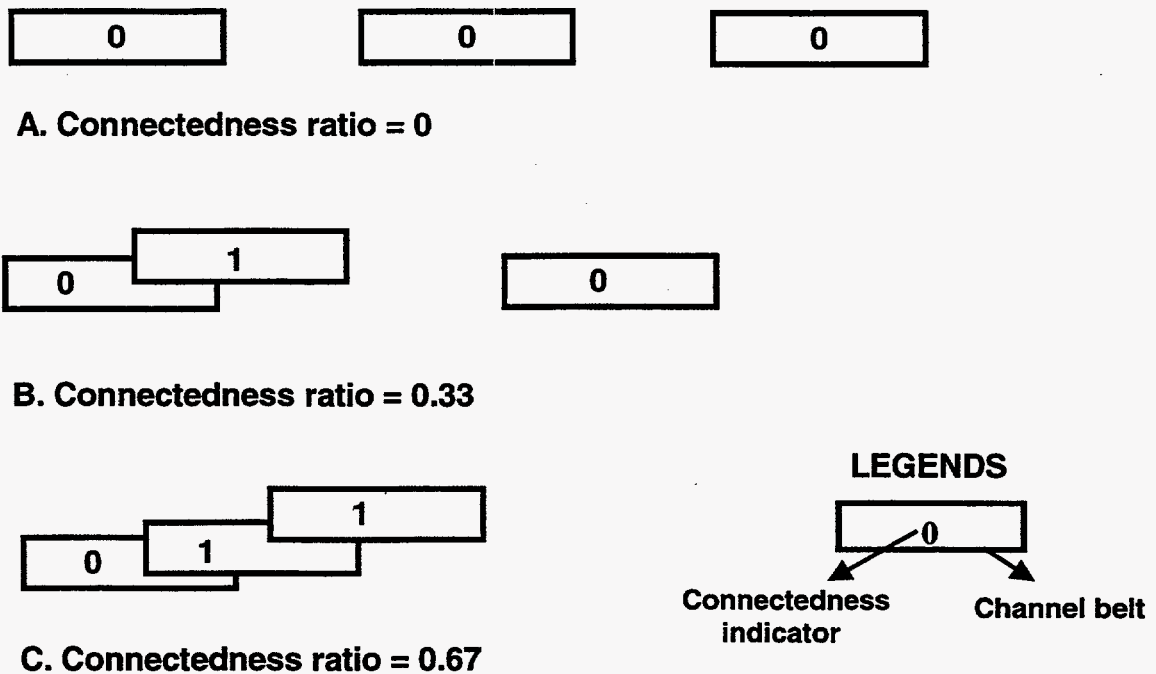


Fig. 3.27. Diagram showing channel belt connectedness definition.

Table 2. Input Parameters for the Example Realizations

Process/features	Parameter	Input Set - 1	Input Set -2
General	Max. increment number	26	22
Floodplain	Length (L: m)	12000	12000
	Width (W: m)	12000	12000
	Initial slope	0.0012	0.0005
	Grid size in W direction (m)	120	120
	Grid size in L direction (m)	240	240
	Local relief center (m)	3000	3000
Channel-belt	Local relief width (m)	3000	6000
	Local relief amplitude (m)	0	0.1
	Width mean (m)	820	820
	Width maximum (m)	1050	1050
	Width minimum (m)	600	600
Avulsion	Width standard deviation (m)	180	180
	Width – depth ratio	180	350
	Expected period (year)	1000	500
	Exponential	0.2	0.2
	Nodal avulsion no. mean	1.5	1.7
	Nodal avulsion no. max.	2	2
	Nodal avulsion no. min.	1	1
	Nodal avulsion no. Stdev	0.5	0.5
	Upstream avulsion no. mean	3.5	2
	Upstream avulsion no. max.	4	3
Tectonism	Upstream avulsion no. min.	3	1
	Upstream avulsion no. Stdev	0.5	0.5
	Max. offset (m)	1000	3000
	Growth fault location (m)	1200	1200
	Faulting rate (m/year)	0.00008	0.0013
	Rollover axis location (m)	3600	3600
	Max. rollover rate (m/year)	0.00025	0.0007
	Structure folding axis (m)	4000	4000
Aggradation	Structure folding width (m)	25000	16500
	Structure folding rate (m/year)	0.0001	0.0009
	Basic rate (m/year)	0.0011	0.0042
Compaction	Overbank exponential b	5.5	1.0
	Hard coded		

the initial channel erosion depth and aggradation rate usually have a reversed relationship according to baselevel fluctuation. A low baselevel may result in deeper channel erosion and slow aggradation and a high baselevel can cause shallow channel erosion and high aggradation rate. These parameters are estimated from channel fill facies thickness and adjusted during simulation process.

Channel belt avulsion period is estimated according to modern deposition study and its value usually falls into the order of $10^1 - 10^3$ years. Since avulsion frequency is also affected by the baselevel fluctuation, a shorter average avulsion period of 500 years is used during high baselevel and a longer average avulsion period of 1000 years is used during low baselevel. The distribution of upperstream avulsion number following each regional avulsion and nodal avulsion following each upperstream avulsion is estimated according to channel facies distribution and stacking pattern variation observed from subsurface cross section of Stratton field (Table 1). These parameters are also adjusted during simulation. The value usually falls into the range of 1 to 4.

The growth fault is treated as a straight line in cross valley direction, located 1200 m down stream from the upperstream boundary of the simulated area. According to subsurface structure study on markers F11m and E41m, the rollover axis is about 2400 m downstream from the major growth fault (3600 ft from the upstream boundary of the simulation area). Growth faulting rate and maximum rollover rate are estimated from interval thickness variation observed from subsurface geological characterization. These tectonic activities also seems related to baselevel changes. The growth fault appears less active during low baselevel period and more active during high baselevel period. The possible explanation for this variation is that when the baselevel drops, the thick delta-front deposition moves farther away from the growth fault location and less material is deposited on the floodplain, the effects of differential loading, differential compaction and differential gravity gliding that cause growth faulting (Jackson and Galloway, 1984) are reduced. Other type of local structural subsidence variation is also observed from isopach map and cross sections and the relatively high subsidence rate in the south part of Stratton field seems to cause high density of channel belt facies.

Since the facies architecture characterization of the middle Frio Formation is mainly based on observation of stratigraphic cross sections, cross section view of the simulation results is also made at three locations roughly correspondent to cross section D-D', E-E' and F-F' for comparison.

Simulation of the Concentrated Architecture

The concentrated stacking pattern is considered to be resulted from low aggradation rate (Kerr and Jirik, 1990). This deposition period may have a relatively low baselevel. Consequently, rivers will cut deeper down to the floodplain during the initial period. Avulsion probability is decreased and avulsion frequency is relatively low. As results, less overbank deposition events occur and floodplain aggradation is slow. These geological characteristics is reflected in the input parameters by a low channel belt width/depth ratio (180), long avulsion period (1000 years), low aggradation rate (0.0012 m/yr) and higher overbank exponent value (5.5) (Table 2).

Totally 26 increments are generated in the simulation of concentrated stacking which is about the same number of channel belts observed in the cross-sections (Table 3). Realization produced using the above input parameters shows a high sand/gross ratio of 54 - 67%, close to the ratio of 54 - 64% estimated from real subsurface cross-sections. The gross thickness of the realization also shows a good match with the subsurface cross sections at the correspondent locations (Table 4).

The stacking pattern variation observed from the subsurface cross sections are also captured in the simulation (Fig. 3.28 and Fig. 3.29). The effect of facies architecture variation from vertical stacking near the growth fault to lateral stacking at more distal location is produced by avulsion hierarchy including regional avulsion, avulsion sequence and nodal avulsion.

Simulation of the Dispersed Architecture

The dispersed architecture is considered to be developed during a period of high aggradation rate which may correspondent to a high baselevel phase. The river cut shallower down to the floodplain. Since more accommodation space is created and sufficient sediments supply is available, floodplain aggrades relatively faster during this phase. The frequency of overbank events as well as river avulsion are likely to increase. A relative thickening of this interval in the midsouth of Stratton field generally parallel to deposition direction can be observed both from isopach map and cross section. This may reflect a relatively high local subsidence rate. This down valley structural feature is simulated as a local depression.

Totally 22 increments are generated in this realization (Table 5). Resulted from growth fault and rollover structure subsidence, significant gross thickness variation is produced at different location relative to the growth fault. The gross thickness of the simulation result matches the subsurface data at correspondent cross section locations (Table 6). Simulation effects also show that the increased aggradation rate resulted in the low sand/gross ratio of 18 - 19%, close to the subsurface calculation of 14 - 18%. Channel belt facies become more dispersed (Fig. 3.30 and Fig. 3.31).

As a result of nodal avulsion which is caused by the growth fault activity, a higher number of channel belt are encountered in cross sections on the down stream side of the fault than on the upstream side. Immediately across the growth fault where rivers start to deviate from its old course after nodal avulsion, channel belt facies are more connected than at other locations (Table 6). The simulation local depression generally along deposition direction in the midsouth of Stratton field caused the effect that channel belts tend to shifting towards the depression axis and the channel belt density close to the depression center is relatively high. These variations fit the observation of subsurface cross sections.

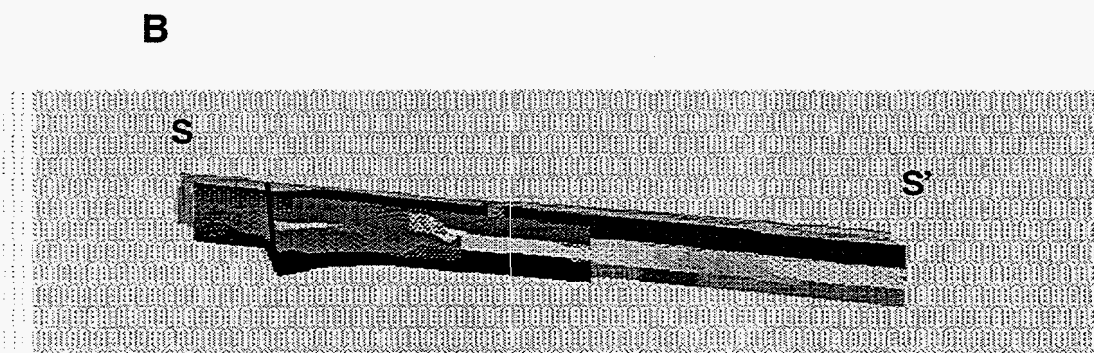
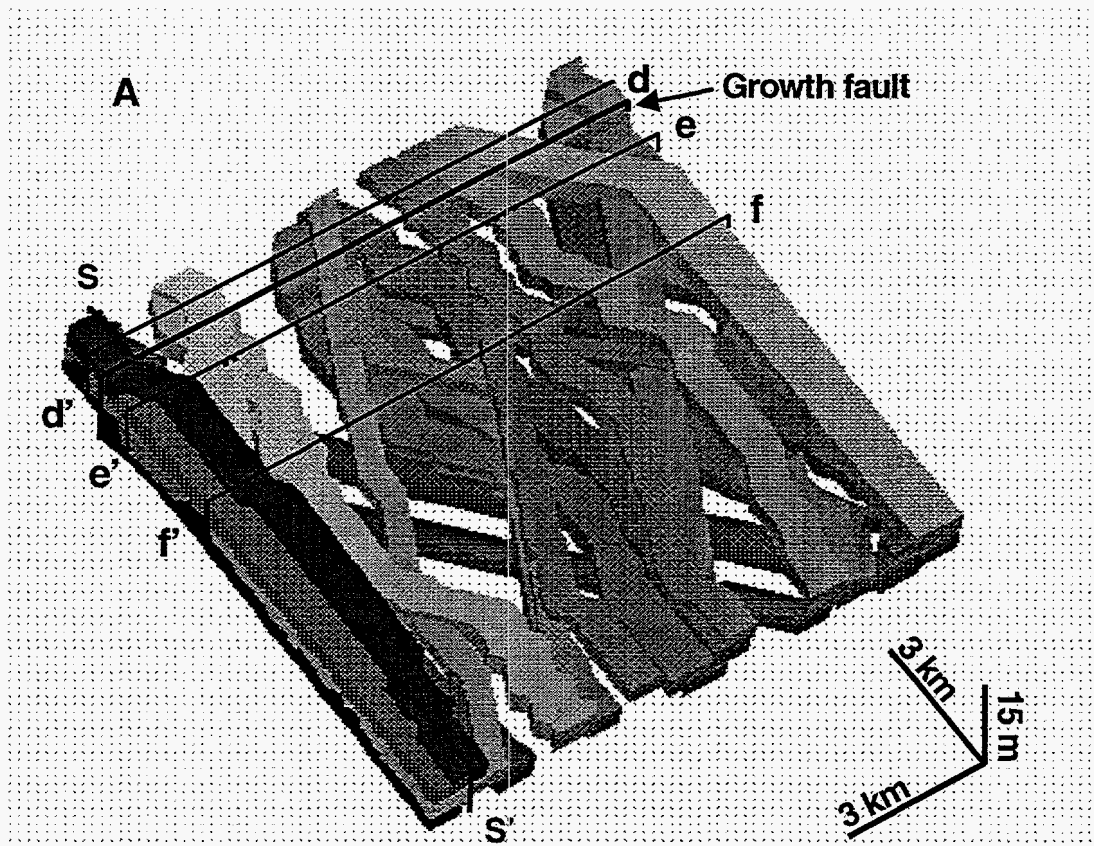


Fig. 3.28. Sample realization of concentrated architecture.
A. 3D visualization of concentrated architecture realization with cross section index.
B. Sliced view of the concentrated realization (sliced at location S-S' and viewed from left).

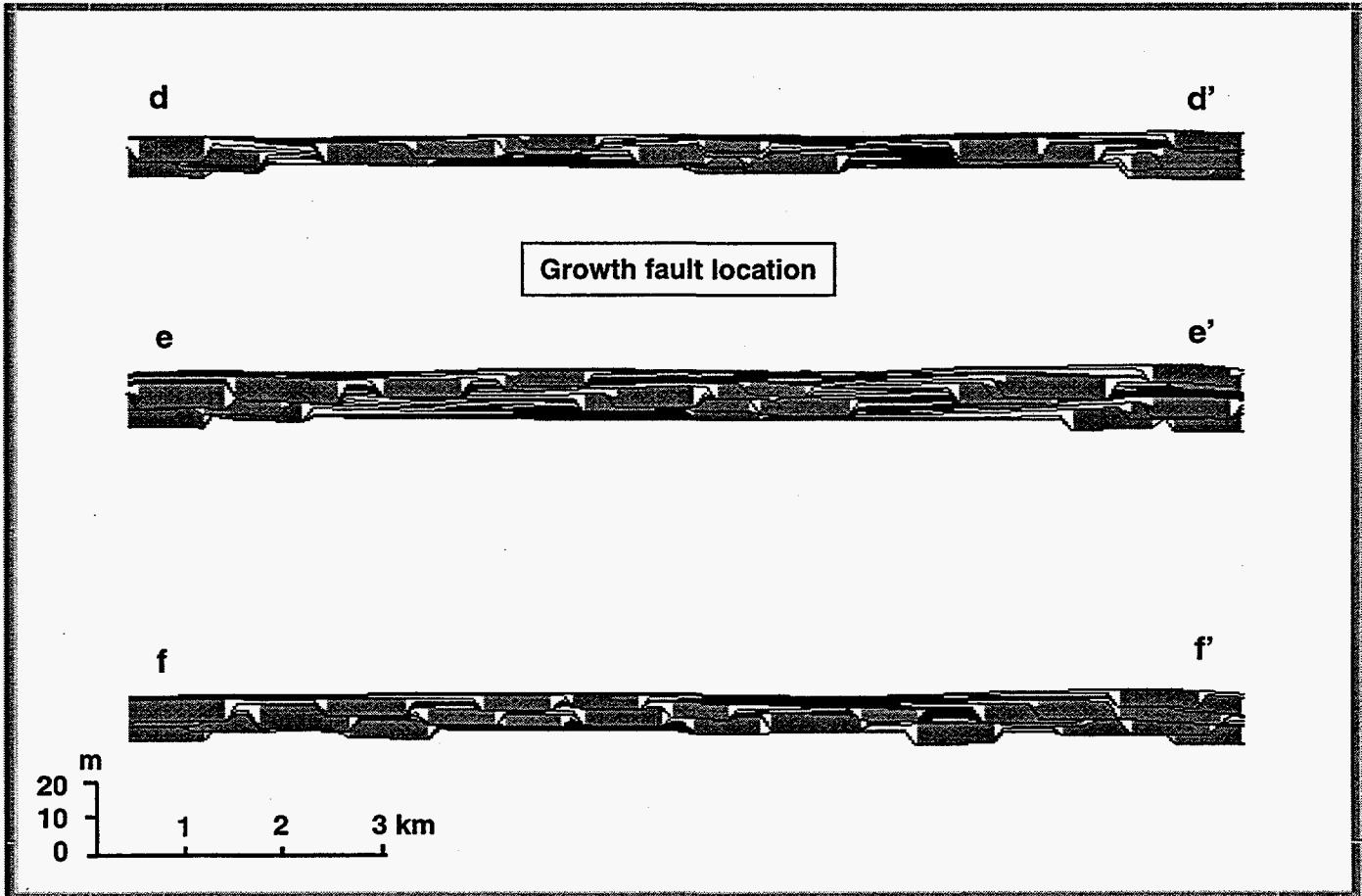


Fig. 3.29. Cross section view of the sample realization for concentrated architecture. The number of channels is increased from d-d' to e-e' and f-f' and become more laterally stacked at f-f'. Upstream to the avulsion point, deposition in the channel belt is considered to be continuous during consecutive avulsion period and form one channel belt.

Table 3. Summary of avulsion increment information for concentrated architecture

avulsion increment	avulsion type	down valley location(m)	cross valley location(m)	avulsion period(yrs)	channel belt width (m)	initial depth (m)
1	R	0	0	832	721	4.0
2	U	0	0	1099	793	4.4
3	N	1200	960	813	677.8	3.8
4	U	0	240	1015	877.6	4.9
5	R	0	11640	1263	845.2	4.7
6	U	0	11640	662	830.8	4.7
7	N	1200	10680	813	812.8	4.5
8	U	0	11520	1020	694	3.9
9	N	1200	11640	1245	1012.6	5.6
10	R	0	5520	1216	688.6	3.8
11	N	1200	5520	398	641.8	3.6
12	U	0	5280	552	937	5.2
13	U	0	5640	1073	715.6	4.0
14	N	1200	6360	1357	821.8	4.6
15	U	0	5880	1245	627.4	3.5
16	N	1200	5160	398	649	3.6
17	R	0	2520	705	874	4.9
18	U	0	2400	1009	906.4	5.0
19	U	0	2280	862	863.2	4.8
20	N	1200	2280	1067	1048.6	5.8
21	R	0	8520	850	1032.4	5.7
22	U	0	8520	764	836.2	4.6
23	U	0	7920	1113	692.2	3.8
24	U	0	7080	723	683.2	3.8
25	R	0	0	838	667	3.7
26	N	1200	360	714	791.2	4.4

*Avulsion type: R - regional, U - upperstream, N - nodal.

Table 4. Simulation results for the concentrated architecture

cross section	realization			subsurface		
	d-d'	e-e'	f-f'	D-D'	E-E'	F-F'
gross thickness (m)	10.9	17.1	13.0	11.5	17.7	12.5
c/g thickness ratio(%)	61	54	67	63	54	58
no. of channel belts	18	26	26	17	26	25
stacking pattern	vertical	vertical	lateral	vertical	vertical	lateral
connectedness ratio	0.66	0.62	0.65	high	high	high

Table 5. Summary of avulsion increment information for dispersed architecture

avulsion increment	avulsion type	down valley location(m)	cross valley location(m)	avulsion period(yrs)	channel belt width (m)	initial depth (m)
1	R	0	3000	416	1036	3.0
2	R	0	12000	510	928	2.7
3	N	1200	11520	347	730	2.1
4	R	0	840	304	838	2.4
5	N	1200	840	386	604	1.7
6	U	0	3240	347	1018	2.9
7	N	1200	3480	403	730	2.1
8	R	0	9000	311	784	2.2
9	N	1200	9000	361	946	2.7
10	U	0	7200	318	946	2.7
11	N	1200	6600	530	748	2.1
12	R	0	1440	437	892	2.5
13	N	1200	2640	403	622	1.8
14	U	0	3960	556	1018	2.9
15	R	0	12000	502	874	2.5
16	N	1200	11520	453	874	2.5
17	U	0	9480	464	766	2.2
18	N	1200	8880	459	748	2.1
19	R	0	2400	304	1018	2.9
20	U	0	3000	199	784	2.2
21	R	0	2400	590	1018	2.9
22	U	0	2400	470	622	1.8

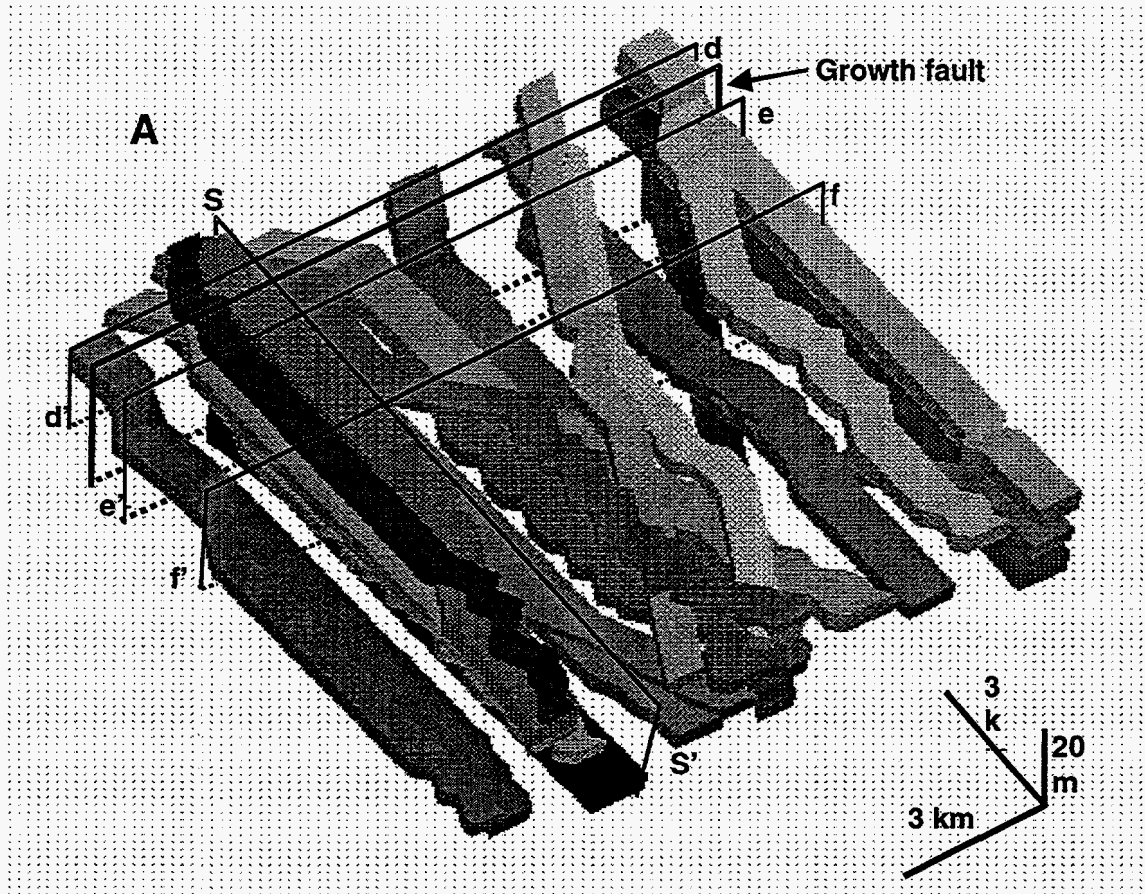


Fig. 3.30. Sample realization of dispersed

A. 3D visualization of dispersed architecture realization with cross section index.

B. Sliced view of the dispersed realization (sliced at location S-S' and viewed from left).

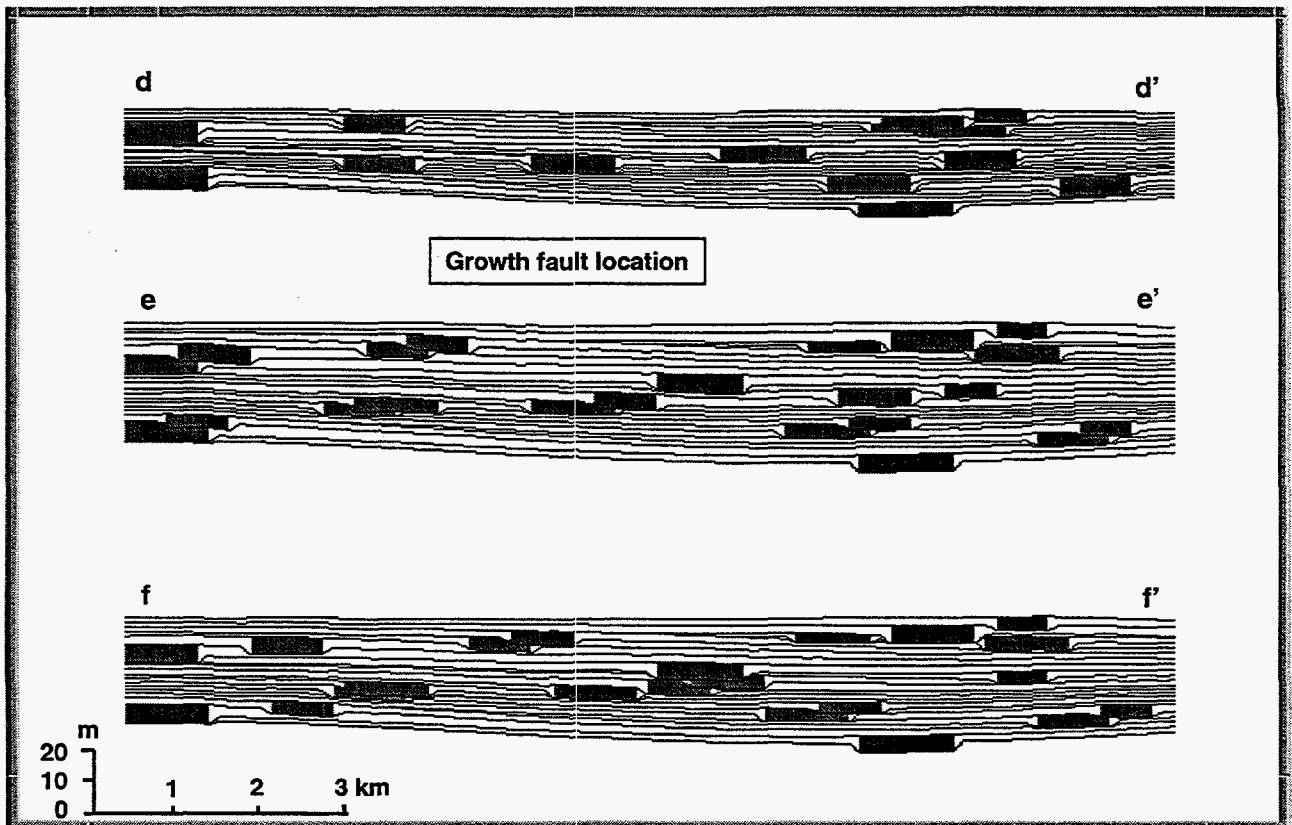
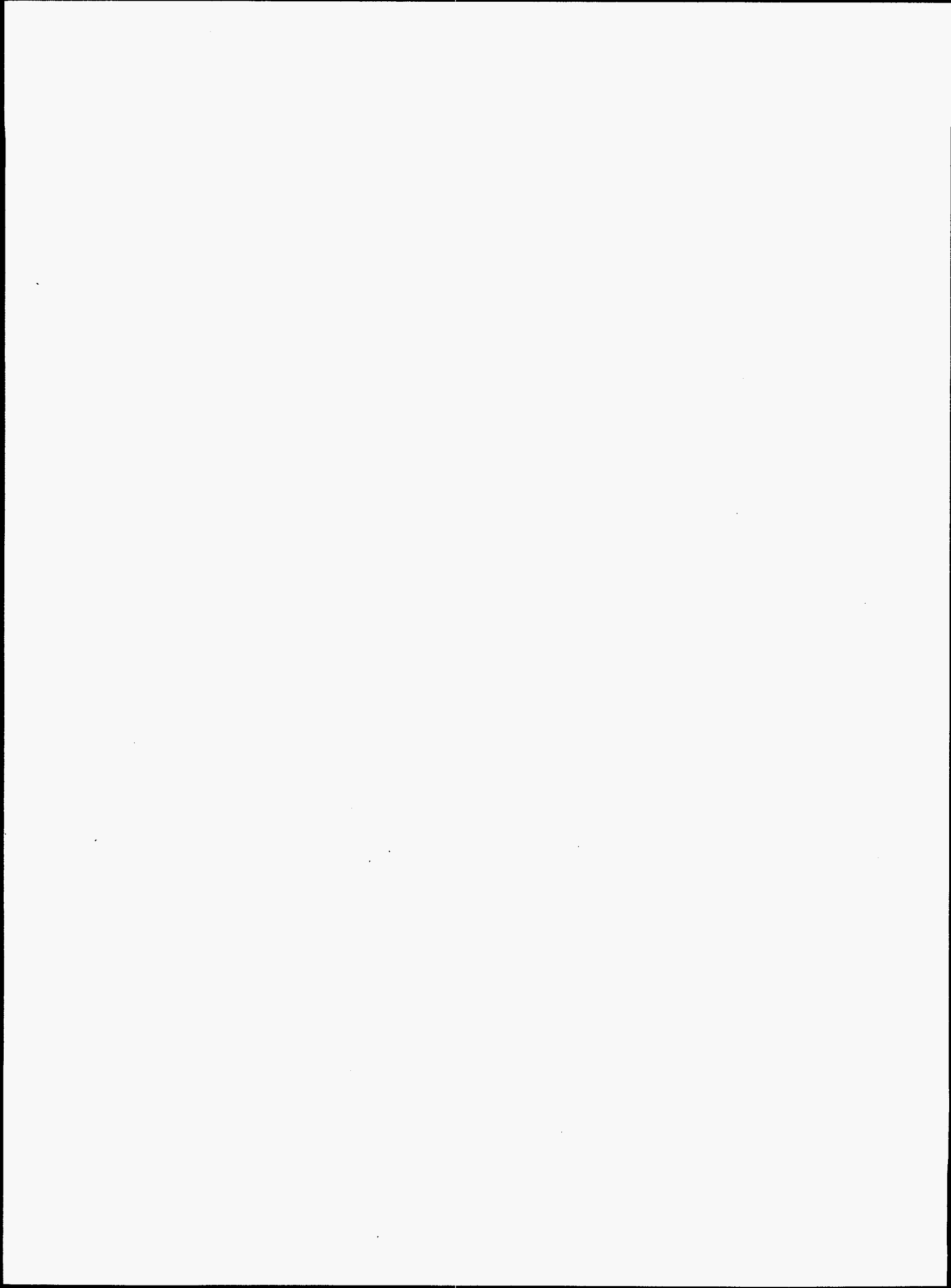


Fig. 3.31. Cross section view of the sample realization for dispersed architecture. The number of channel belts is increased from d-d' to e-e' and f-f' and channel belts are more concentrated towards the depression center. Upstream to the avulsion point, channel belt deposition is considered to be continuous during consecutive avulsion period and form one channel belt.

Table 6. Simulation results for the dispersed architecture

cross section	realization			subsurface		
	d-d'	e-e'	f-f'	D-D'	E-E'	F-F'
gross thickness (m)	30.7	45.6	40.7	28.9	45.1	40
c/g thickness ratio(%)	19	18.7	18	14	18	16
no. of channel belts	14	22	22	14	22	22
connectedness ratio	14	36	27	uncnct.	low	very low

Note: c/g: channel belt to gross
 uncnct.: unconnected



Suggestions for Further Improvements

Even though three dimensional process based model can produce realistic results and capture some of the major architecture features, it is still very difficult to match different quantitative statistic parameters and to honor local data. Further development is needed in the following areas:

1) Better understanding of physical mechanisms of the interaction between growth faulting and river behavior. Aspects of study include growth faulting fashion, fluid flow, channel incision, sedimentation and avulsion. Studies in similar modern environment is especially important.

2) Geological parameterization. Some of the input parameters are difficult to define from direct geological characterization. Techniques need to be developed to help define these parameters based on characterization of real geological data.

3) Incorporate conditional modeling techniques in this forward modeling procedure in order to honor local data and statistic parameters.

REFERENCES

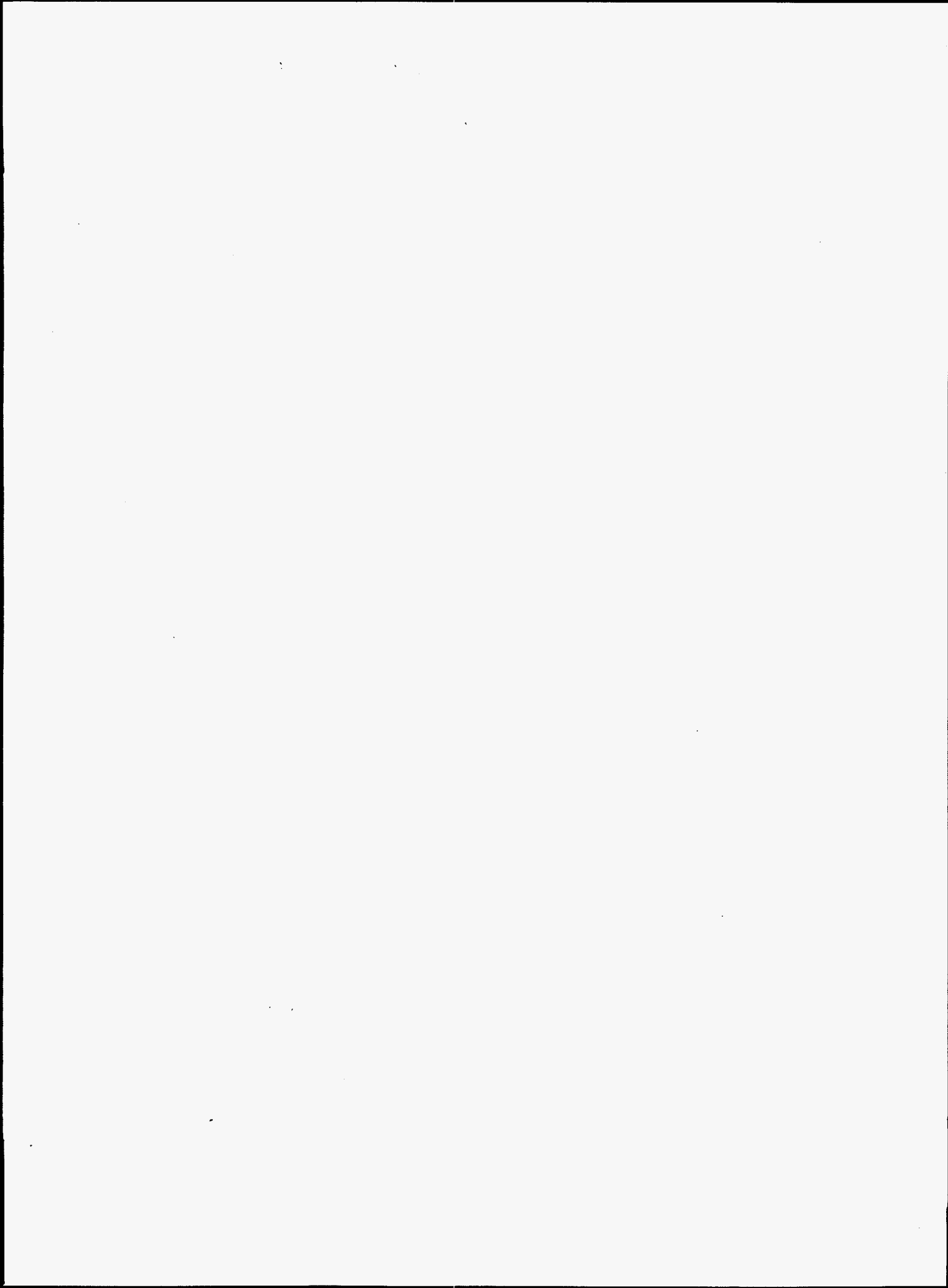
- Allen, J. R. L., 1978, Study in fluvial sedimentation: An exploratory quantitative model for the architecture of avulsion-controlled alluvial suites: *Sedimentary Geology*, v. 21, p. 129-147.
- Allen, J. R. L., 1979, Study in fluvial sedimentation: An elementary model for the connectedness of avulsion-related channel sand bodies: *Sedimentary Geology*, v. 24, p. 253-267.
- Baldwin, B., and Butler, C. O., 1985, Compaction curves: *American Association of Petroleum Geologists Bulletin*, v. 69, p. 622-626.
- Bernard, H. A., Major, C. F., Parrot, B. S., and Le Blanc, R. J., 1970, Recent sediments of the southeast Texas: A field guide to Brazos alluvial and deltaic plains and the Galveston Barrier Island complex: Bureau of Economic Geology, University of Texas-Austin, Guidebook 11, 16 p.
- Bridge, J. S., and Leeder, M. R., 1979, A simulation model of alluvial stratigraphy: *Sedimentology*, v.26, p. 617-644.
- Bridge, J. S., and Mackey, S. D., 1993a, A revised alluvial stratigraphy model, in Marzo, M., and Puigdefabregas, C., eds., *Alluvial Sedimentation: International Association of Sedimentologists Special Publication 17*, p. 319-336.
- Bridge, JS and Mackey, S. D., 1993b, A theoretical study of fluvial sandstone body dimensions, in Flint, S., and Bryant, I.D., eds., *The Geological Modeling of Hydrocarbon Reservoirs: International Association of Sedimentologists Special Publication 15*, p. 213-236.

- Chrzastowski, M.J. et al., 1994, The Great Flood of 1993 geologic perspectives on flooding along the Mississippi River and its tributaries in Illinois: Illinois State Geological Survey Dept. of Energy and Natural Resources, Special Report 2.
- Collinson, J.D., 1978, Vertical sequence and sand body shape in alluvial sequences, *in* Miall A.D., ed., *Fluvial Sedimentology*: Canadian Society of Petroleum Geologists Memoir 5, p. 577-586.
- Cotter, E., 1971, Paleoflow characteristics of a Late Cretaceous river in Utah from analysis of sedimentary structures in the Ferron Sandstone: *Journal of Sedimentary Petrology*, v. 41, p. 129-138.
- Ewing, T. E., 1986, Structural styles of the Wilcox and Frio growth-fault trends in Texas: Constraints on geopressed reservoirs, US DOE report, 90p.
- Falkner, A.J. and Fielding, C.R., 1993, Geometrical facies analysis of a mixed influence deltaic system: the Late Permian German Creek Formation, Bowen Basin, Australia, *in* Marzo, M., and Puigdefabregas, C., eds., *Alluvial Sedimentation*: International Association of Sedimentologists Special Publication 17, p. 195-209.
- Fielding C.R. and Crane, R.C., 1987, An application of statistical modeling to the prediction of hydrocarbon recovery factors in fluvial reservoir sequences *in* Etheridge, F.G., Flores, R.M., and Harvey, M.S., eds., *Recent Developments in Fluvial Sedimentology*: SEPM Special Publication 39, p. 321-327.
- Fisk, H. N., 1944, Geological investigation of fluvial valley of the Lower Mississippi River: United States Army Corps of Engineers, Mississippi River Commission Report, Vicksburg, Mississippi, 78p.
- Fisk, H. N., 1947, Fine grained alluvial deposits and their effects on Mississippi River activity: United States Army Corps of Engineers, Mississippi River Commission Report, Vicksburg, Mississippi, 82p.
- Flores, R. M., 1984, Comparative analysis of coal accumulation in Cretaceous alluvial deposits, southern United States Rocky Mountain basins, *in* Statt, D.F, and Glass, D.J., eds., *The Mesozoic of Middle North America*: Canadian Society of Petroleum Geologists Memoir 9, p. 373-385.
- Galloway, W. E., 1977, Catahoula Formation of the Texas Coastal Plain - depositional system, composition, structural development, groundwater flow history, and uranium distribution: The University of Texas at Austin, Bureau of Economic Geology Report of Investigations No. 87.
- Galloway, W. E., Ewing, T. E., Garrett, C. M., Tyler, N. and Bebout, D. G., 1983, , Atlas of major Texas oil reservoirs: The University of Texas at Austin, Bureau of Economic Geology, 139p.
- Galloway, W. E., Hobday, D. K. and Magara, K., 1982, Frio Formation of the Texas Gulf Coast Basin - Depositional systems, structural framework, and hydrocarbon origin, migration, distribution, and exploration potential, the University of Texas at Austin, Bureau of Economic Geology Report of Investigations No. 122, 78P.
- Hardage B. A., Levey, R. A., Pendleton. V., Simmons, J., and Edson, R., 1994, A 3-D seismic case history evaluating fluvially deposited thin-bed reservoirs in a gas-producing property: *Geophysics*, v. 59, No. 11, p. 1650-1665.
- Jackson, M. P. A., and Galloway, W. E., 1984, Structural and Depositional styles of Gulf Coast Tertiary continental margins: Application to hydrocarbon exploration, AAPG Continuing education course note series #25, 226p.

- Kerr, D. R., 1990, Reservoir heterogeneity in the middle Frio Formation: case study in Stratton and Agua Dulce fields, Nueces County, Texas, Gulf Coast Association of Geological Societies Transactions, v. 11, p. 363-372.
- Kerr, D. R., and Grigsby, J. D., 1991, Recognition and implications of volcanic glass detritus in the fluvial deposits of the middle Frio Formation, South Texas: Gulf Coast Association of Geological Societies Transactions, v. 12, p. 353-358.
- Kerr, D. R., and Jirik, L. A., 1990, Fluvial architecture and reservoir compartmentalization in the Oligocene Middle Frio Formation, South Texas: Gulf Coast Association of Geological Societies Transactions, v. 11, p. 373-380.
- Kerr, D. R., Grigsby, J. D., and Scott, A. R., 1991, Fluvial deposits of the Frio Formation along the Vicksburg fault zone: Examples from Stratton and Seeligson fields, *in* Levey R. A., Grigsby, J. D., Langford, R. P., Kerr, D. R., Scott, A. R. and Finley, R. J.: Core and log analysis of depositional systems and Reservoir Properties of Gulf Coast Natural gas reservoirs: An integrated approach to infield reserve growth in Frio, Vicksburg, and Wilcox sandstones, Notes for a core workshop prepared for the October, 1991 Gulf Coast Association of Petroleum Geologists Convention, Houston, Texas, p. 7-26.
- Kesel, R. H., Dunne, K. C., McDonald, K. R., and Spicer, B. E., 1974, Lateral overbank deposition on the Mississippi River in Louisiana caused by 1973 flooding: *Geology*, v. 1, p. 461-464.
- Kosters E. C., Bebout, D. G., Seni, S. J., Garrett, C. M., Brown, L. F., Hamlin, H. S., Dutton, S. P., Ruppel, S. C., Finley, R. J. and Tyler, N., 1989, Atlas of major Texas gas reservoirs: The University of Texas at Austin, Bureau of Economic Geology Special Publication, 160p.
- Leeder, M. R., 1978, A quantitative stratigraphic model of alluvium, with special reference to channel deposit density and interconnectedness, *in* Miall, A. D., ed., *Fluvial Sedimentology: Canadian Society of Petroleum Geologists Memoir 5*, p. 587-596.
- Leeder, M.R., 1973, Fluvial fining-upwards cycles and the magnitude of palaeochannels: *Geology Magazine*, v. 110, p. 265-276.
- Levey, R. A. et al, 1993, Secondary natural gas recovery: Targeted technology applications for infield reserve growth in fluvial reservoirs, Stratton Field, South Texas, topical report, 244p.
- Lorenz, J.C., Heinze, D.M., Clark, J.A., and Searls, C.A., 1985, Determination of widths of meander-belt sandstone reservoirs from vertical downhole data, Mesaverde Group, Piceance Creek Basin, Colorado: *AAPG Bulletin*, v. 69, p. 710-721.
- Lowry, P., and Jacobsen, T., 1993, Sedimentological and reservoir characteristics of a fluvial-dominated delta-front sequence: Ferron Sandstone member (Turonian), East-central Utah, USA, *in* Ashton, M., ed., *Advances in Reservoir Geology: Geological Society Special Publication 69*, p. 81-103.
- Mackey, S. D., and Bridge, J. S., 1992, A revised FORTRAN program to simulate alluvial stratigraphy, *Computer and Geosciences* v.18, No. 2/3, p. 119-181.
- Mackey, S. D., and Bridge, J. S., 1995, Three-dimensional model of alluvial stratigraphy: theory and application, *Journal of Sedimentary Research* v. B65, No. 1, p. 7-31.

- Mial, D. A., 1991, The three-dimensional facies architecture of terrigenous clastic sediments and its implication for hydrocarbon discovery and recovery: Concepts in sedimentology and paleontology, SEPM special publication, v.3, 309 p.
- Mjos, R. et al., 1993, Crevasse splay sandstone geometries in the Middle Jurassic Ravenscar Group of Yorkshire, UK in *Alluvial Sedimentation: International Association of Sedimentologists Special Publication 17* ed., M.Marzo and C. Puigfabregas, P.167-184.
- Mjos, R., Walderhaug, O., and Prestholm, E., 1993, Crevasse splay sandstone geometries in the Middle Jurassic Ravenscar Group of Yorkshire, UK, in Marzo, M., and Puigfabregas, C., eds., *Alluvial Sedimentation: International Association of Sedimentologists Special Publication 17*, p. 167-184.
- Morton, R. A., and Galloway, W. E., 1991, Depositional, tectonic and eustatic controls on hydrocarbon distribution in divergent basins: Cenozoic Gulf of Mexico case history: *Marine Geology*, v. 102, p. 239-263.
- O'Brien, P. E. and Wells, A. T., 1986, A small, alluvial crevasse splay: *Journal of Sedimentary Petrology*, V.56, No. 6, P.876-879.
- O'Brien, P.E., and Wells, A. T., 1986, A small alluvial crevasse splay: *Journal of Sedimentary Petrology*, v. 56, p. 876-879.
- Perrier, R., and Quiblier, J., 1974, Thickness changes in sedimentary layers during compaction history: Methods for quantitative evaluation: *American Association of Petroleum Geologists Bulletin*, v. 58, p. 507-520.
- Pizzutto, J. E., 1987, Sedimentary defusion during overbank flows: *Sedimentology*, v. 34, p. 301-317.
- Puigdefabregas, C., 1973, Miocene point-bar deposits in the Ebro Basin, Northern Spain: *Sedimentology*, v. 20, p. 133-144.
- Rhee, C. W., and Chough, S.K., 1993, The Cretaceous Pyonghae Basin, southeast Korea: sequential development of crevasse splay and avulsion in a terminal alluvial fan: *Sedimentary Geology*, v. 83, p. 37-52.
- Russell, R. J., 1954, Alluvial morphology of Anatolian rivers. *Ann. Ass. Am. Geog.* V. 44, p. 363-391.
- Salinas-Rincones, L., 1996, Architecture of Middle Frio gas reservoir F21 - F25, Southern Stratton Field, Texas, MS thesis of the University of Tulsa, 108p.
- Schumm, S.A., 1963, Sinuosity of alluvial rivers on the Great Plains: *Geological Society of America Bulletin*, v. 74, p. 1089-1099.
- Schumm, S.A., 1978, Fluvial paleochannels, in Rigby, J. K., and Hamblin, W. K., eds., *Recognition of Ancient Sedimentary Environments: Society of Economic Paleontologists and Mineralogists Special Publication 16*, p. 98-107.
- Smith, N. D., Cross, T. A., Dufficy, F.P., and Clough, S. R., 1989, Anatomy of an avulsion: *Sedimentology*, v. 36, p. 1-23.
- Stancliffe, R. J., and Adams, E.R., 1986, Lower Tuscoloosa fluvial channel styles at Liberty Field, Amite County, Mississippi: *Transactions of the Gulf Coast Association of Geological Societies*, v. 36, p. 305-313.
- Tyler, N. et al., 1991, Architecture and permeability-structure analysis of retrogradational deltaic sandstones: A Field Guide to Selected Outcrops of the Ferron Sandstone, East-Central Utah: The University of Texas at Austin, Bureau of Economic Geology, 80 p.

- Van Vark, W., Gorge, P. M., Paardekam, A. H. M., Brint, J. F. and Van Lieshout, J. B., 1992, The construction and validation of a numerical model of a reservoir consisting of meandering channel: SPE Europe Petroleum Conference. PROC V. 2, P. 407-417.
- Vendeville, B., and Cobbold P. R., 1988, How normal faulting and sedimentation interact to produce listric fault profiles and stratigraphic wedges: Journal of Structural Geology. Vol. 10. No. 7. p 649-659.
- Wells, J.T. et al., 1984, The Atchafalya River delta, Report 4, generic analysis of delta development: prepared for the U.S. Army Engineer District, New Orleans, monitored by Hydraulics Laboratory, 89 p.
- Williams, G. P., 1986, River meanders and channel size: Journal of Hydrology, v. 88, p. 147-164.
- Williams, G. P., 1988, Paleofluvial estimates from dimensions of former channels and meanders, *in* Baker, V. R. et al., eds., Flood Geomorphology: John Wiley and Sons, Inc., p. 321-334.



APPENDIX I: Computer Programs

Description

The directory "appendix" contains both the source code and executable of the computer programs generated and used in the study of 3D Fluvial Facies Architecture Simulation of the Middle Frio Formation, Stratton Field, South Texas (see readme file under appendix). The program file names are listed and described as following:

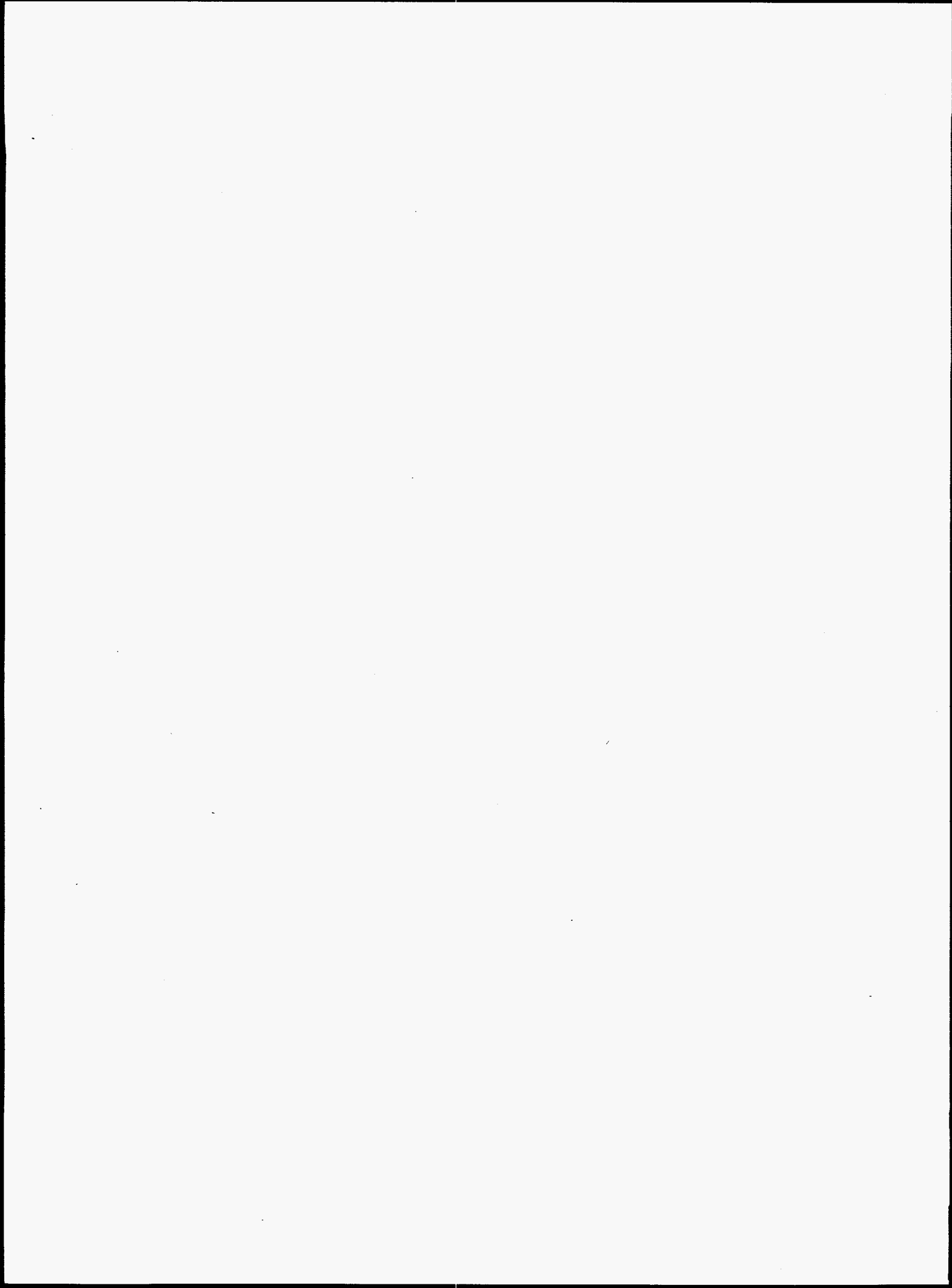
1. facsim.cpp: source code of facies simulation program written in C++;
2. facsim: executable of the facies simulation program "facsim.cpp", compiled at a UNIX Sun workstation of Tulsa University euler system;
3. strav.c: source code of 3D visualization program written in OpenGL and using C as driven language;
4. strav: executable of "strav.c", compiled at a UNIX HP workstation of Tulsa University hpserv system;
5. crossv: source code of cross section visualization program written in OpenGL and using C as driven language;
6. crossv: executable of "crossv", compiled at a UNIX HP workstation of Tulsa University hpserv system;

Instructions

The procedure to use this program package is listed as following:

1. Prepare the input file to "facsim"
A input file named "param.data" must prepared before **facsim** can be used. Values of all the 34 parameters needed for the facies architecture simulation should be defined in this file. All the parameter values are delimited by a space or end of line. The order of the parameters is:
IncreMax
LFP WFP SlopeIni IntervX IntervY
Lcenter Lwidth Ldepth
WcMean WcMax WcMin WcStdv WDRatio
ExpeAvul AvulExpo
NudMean NudMax NudMin NudStdv
SeqMean seqMax seqMin seqStd offMax
FaultDist Rfault AxisDist Rrm
Dcenter Dwidth Drate
Rbas b
See report text for explanation of the parameters.
2. Run the **facsim** program by simply type in facsim and press enter key. Some statistical results will appear on the screen as the program running and two output files named "results" and "cross" will be produced. The file "results" contains the data for 3D visualization and "cross" contains data for cross section visualization.
3. Run the **strav** and **crossv** program by simply typing in their names and pressing enter key respectively. Note that the input files "results" and "cross" must be under the same directory as the program **strav** and **crossv**.

Note: a different computer system environment may cause some unexpected problems in using these programs.



PART IV

SEISMIC MODELING AND INTERPRETATION

By: Durydhan Epili, Chong Chung, Bassel M. Al-moughraby and Inta Arpandi

SUMMARY.

Seismic amplitude and velocity are two important attributes for reservoir characterization. The study of seismic amplitude involves the complete field of geophysics due to several contributing factors. Here, we studied the seismic amplitude contribution with and without consideration to transmission losses and with and without incorporating multiples. Also, the contribution to seismic amplitude for marine and land environments are analyzed.

SYNTHETIC SEISMOGRAM (1-D, ZERO OFFSET, NO MULTIPLES)

The basic concepts of the reflection seismograph method are relatively simple and straightforward. A pulse from a seismic source travels outwards gets reflected when it encounters an impedance (product of velocity and density) change, travels upwards towards the surface and gets recorded by seismometer. The energy with which the reflected wave travels is dependent on the acoustic impedance contrast at the boundary separating 2 media.

Although the process is simple, a number of complicating factors enter into the process. In order to understand and interpret the data on seismograms it is necessary to look all the processes closely. A complete analysis of these factors would encompass the entire field of reflection seismology. We will limit our analysis on multi-layered rocks where accurate velocity information from well information is available.

Several approaches to generate a synthetic seismogram have been reported in geophysical literature; they range in some simple convolution models to more complicated viscoelastic finite difference models. Here for understanding the concepts, we present a simple one-dimensional convolutional model.

We have generated a one-dimensional synthetic seismogram for a simple geologic model (Figure 4.1). It is simply a zero phase Ricker wavelet shown in Figure 4.2 convolved with a reflectivity sequence assuming zero offset and horizontal layering. Reflectivity is the response of a seismic wavelet to an acoustic impedance change within the earth. In equation form,

$$s(t) = w(t) * r(t) + n(t)$$

where

$s(t)$ = the seismic trace

$w(t)$ = a seismic wavelet,

$r(t)$ = earth reflectivity

$$r(t) = \frac{\rho_2 v_2 - \rho_1 v_1}{\rho_2 v_2 + \rho_1 v_1} \quad (4.1)$$

Where ρ_1 , ρ_2 are the densities, and v_1 , v_2 are the velocities of media one and two respectively.

$n(t)$ = additive noise,

* implies convolution

In this seismogram, we assumed that noise is zero.

Procedure to generate synthetic seismogram (vertical ray path)

The steps for generating the seismogram are as follows: (Sheriff and Gildart, 1995; Gadallah, 1994; Dobrin and Savit, 1988)

- (a) Digitize the sonic and density log at uniform depth.
- (b) Convert the depth values of the velocity and density to time values.
- (c) Combine transit time and density samples to produce acoustic impedance. The acoustic impedance is derived as a function of one way time by multiplying velocity and density values.
- (d) Compute the reflectivity series as a function of time.
- (e) Convolute the reflectivity series with a wavelet of the same bandwidth of theseismic section with which the synthetics will be compared.

The reflectivity sequence and the synthetic seismogram (after convolution with the wavelet) are shown in Figures 4.3 and 4.4.

Fluvial Channels and Their Seismic Response

Fluvial refers to process related to rivers. Channel patterns are related to controlling factors that include suspension of the load, sinuosity, and discharge (Schumm, 1981). In Figure 4.5, we are looking at a 3-D model of a fluvial system. The channel is meandering but we are looking at a straight part of the channel. The model has no velocity variations in the X, Y direction. The seismic response generated for this model is very similar to the response generated for a 1-D synthetic seismogram. The acoustic impedance was derived; the reflectivity series was computed as a function of time (Figure 4.6). At last the reflectivity series was convolved with a zero phase Ricker wavelet

(Figure 4.7). In this model, the facies changes do not occur in a horizontal lines, but rather in a synclines. The seismic response also takes a similar shape and we can clearly see the boundaries between the facies.

In seismic processing, we deal with digital data that is sampled at a constant time interval. If we consider the reflectivity to consist of a reflection coefficient at each time sample (some of which could be zero), and the wavelet to be a smooth function in time, convolution can be thought of as replacing each reflection coefficient with a scaled version of the wavelet and summing the results. One of the important aspects of this convolution to notice is that the wavelet smears the reflection coefficients. The outcome is the loss of resolution, which is the ability to resolve closely spaced reflectors.

Vertical Resolution

One of the fundamental equations governing the wave propagation is

$$\lambda = \frac{V}{f}$$

Where

λ = wave length,

V = velocity

f = frequency of the wavelet

The vertical resolution limit, $\lambda/4$ is the thin bed that can be seen in the data from distinct top and bottom reflections. This assumes that top/base of the reflections from the bed are the same polarity. If the top/base reflections are of opposite polarity, then $\lambda/4$ bed thickness causes high amplitude interference effect called tuning. For beds thinner than $\lambda/4$, the beds become unresolvable; the wavelet alignment is lost and the amplitude falls off rapidly.

Horizontal Resolution

Another fundamental concept in seismology is that a reflection does not arise from a single point on a reflector. It arises from an area called Fresnel zone. The Fresnel zone is considered as the horizontal resolution. The Fresnel diameter (F) is defined as

$$F = V \sqrt{\frac{t}{f}}$$

where

t = two way travel time,

V = wave velocity,

f = dominant frequency.

AMPLITUDE

We see a lot of amplitude variations in seismic data, most of which do not give any information about the subsurface. The information we are interested in are the primary waves reflected from the reflectors. We usually think of reflection as dependent on density and velocity. The factors contribute to the reflection coefficients are lithology, porosity and fluid content. If we want to attribute lithologic or hydrocarbon accumulation to amplitude variations, we must reduce the distracting factors as much as possible. Some factors are multiple reflection, array attenuation, coherent and random noise, spherical spreading, reflector curvature, spherical wavefronts, transmission coefficients, instrumentation/processing, inelastic attenuation, and diffraction (Castagna, 1993; Ostrander, 1984).

In a simplified form, the seismic amplitude is given approximately by:

$$A = \frac{S \cdot D_S \cdot D_R \cdot R}{G} \prod_j T_j e^{-\alpha_j l_j} \quad (4.2)$$

where,

- A = received amplitude
- S = source amplitude
- D_S, D_R = source and receiver directivities (radiation patterns)
- G = geometrical spreading function
- R, T_j = reflection and transmission coefficients (angular)
- Π = Product symbol
- α_j = attenuation factor
- l_j = raypath length
- j = counter for each layer transverse by the ray.

This formula ignores several factors including, source/receiver coupling (land data), instrument performances, interference from multiple reflections, interference from ground roll, refractions and other source-generated noise, random noise, data processing, and reflector curvature. A few manipulations are performed on seismic wiggles to correct the amplitude losses.

It is not easy to estimate or compensate for each contribution to amplitude. Spherical divergence correction, attenuation, and loss due to reflection and transmission is usually compensated and rest are all put in some sort of normalization. Thus, conventionally a relative amplitude is preserved for interpretation.

On the following section, we study the reflection and transmission characteristic of seismic response.

D Simulation (Amplitude with transmission loss and multiples).

A seismic wave is an acoustic wave in a solid rock. It represents the motion of particles and the effect of the inter-particle or elastic forces. It has been questioned what should be measured in a seismic wave. Therefore the knowledge of the environment that we conduct the seismic survey needs to be understood.

There is a significant confusion in literature on seismic amplitude. Main part of the confusion is whether our measurements is particle velocity response or particle pressure. Unless this is clear, the physics behind the problem can be inaccurate.

Amplitude signifies the magnitude of the acoustic pressure for marine survey and magnitude of the particle velocity for land survey. When we work on land with geophones at the free surfaces, it is essential that we use geophones sensitive to motion in the form of particle velocity not to pressure. When we work at sea, the free surface is too noisy, in order to avoid this noise we place the detectors deep in the water, which allow us to use hydrophones that are sensitive to pressure not to motion.

When source initiates a seismic wavefield, that wavefield interacts with the earth. The wave is partially transmitted and reflected. Only receivers at the earth's surface measure the reflected wave. The measured field at the surface is known as the seismic response of the earth at that particular location.

Reflection coefficient at normal incidence

By making displacement and stress continuous across an interface, a simple relationship for the ratio of the amplitude of the reflected wave compared to the amplitude of the incident wave, (called as reflection coefficient (R_0)) can be calculated as (Sheriff, 1975)

$$R_0 = \text{Reflected amplitude} / \text{Incidence amplitude} \quad (4.3)$$

Two ways of calculating the reflection coefficient depend on the seismic acquisition: a land geophone measures particle velocity; a marine streamer hydrophone measures acoustic pressure.

Pressure amplitude

If a plane wave of pressure amplitude, p_i , propagating in a material of acoustic impedance I_1 , is incident on the interface into a material of acoustic impedance I_2 , then a wave of pressure amplitude p_r , is reflected from the interface, where:

$$I = V\rho \quad (4.4)$$

$$R_p = \frac{p_r}{p_i} = \frac{I_2 - I_1}{I_2 + I_1} \quad (4.5)$$

R_p is the reflection coefficient, V is the acoustic velocity, ρ is the density of the material and 1 and 2 are the upper and lower layer.

Particle velocity amplitude

A wave of particle velocity amplitude v_i , is propagating through the similar material of acoustic impedance as in pressure amplitude then v_r is reflected from the interface, where:

$$R_v = \frac{v_r}{v_i} = \frac{I_1 - I_2}{I_1 + I_2} \quad (4.6)$$

There are two limiting values implicit in the reflection coefficient formula; when I_2 is zero for a free surface and when I_2 is infinite for a fixed surface. The first limitation is approximated by the interface between earth or sea and the air. Substantially the reflection coefficient is equal to -1 , which implies that an upcoming seismic wave is transmitted back down to the earth by the free surface.

Transmission coefficient at normal incidence

The proportion of the signal reflected is defined by the reflection coefficient R , and the proportion transmitted is defined by the transmission coefficient T .

Pressure amplitude

The transmission coefficient T_p , for pressure amplitude is represented as follows:

$$T_p = \frac{p_t}{p_i} = \frac{2I_2}{I_2 + I_1} \quad (4.7)$$

Where p_i is the incident pressure and p_t is the transmitted pressure.

Particle velocity amplitude

The transmission coefficient, T_v , for particle velocity amplitude is written as:

$$T_v = \frac{v_t}{v_i} = \frac{2I_1}{I_1 + I_2} \quad (4.8)$$

The transmission coefficient, for both of the pressure and particle velocity amplitude can be written as:

$$T_p = 1 + R_p \quad (4.9)$$

$$T_v = 1 + R_v \quad (4.10)$$

If the reflection coefficient is positive then transmitted signal is larger than the incident signals, which seems incorrect. However, it is mathematically correct and physically reasonable. If there is no reflected signal, all the incident energy gets transmitted into the lower layer. The pressure amplitude in the lower layer increases to a large value, which correspond to reduction of particle velocity (see Figure 4.8).

Figure 4.8 shows the wave propagation through a horizontal layer. The sketch at the left is for the pressure measurement and that at the right is for the particle velocity measurement. Note that from the right sketch the reflection recorded by a velocity-sensitive geophone appears to be negative, not because the reflection coefficient is negative but due to the wave has changed direction.

LATTICE FORMULATION

The fundamental problem addressed in seismic modeling or simulation is calculation of the seismic response (i.e. traveltime and amplitude) for a given earth model. The seismic simulation can lead to generate the synthetic seismograms that are essential to identify events for seismic interpretation and analysis.

The stratified earth model is based on the fact that wave motion in each layer is traveling up-going as reflected wave and down-going as transmitted wave for the upper layer. By applying the lattice formulation, we can obtain a complete time and amplitude profile of the up-going and down-going wave in multi-layered media.

In this analysis, it is assumed to have the same one-way traveltime for a pulse propagating from one interface to the other within several layers. This one-way traveltime corresponds to one-half unit time. Figure 4.9 shows the details of assigning the up-going and down-going waves for each traveltime.

In Figure 4.9 we set the down-going and up-going waves at the top of layer k respectively as $d_k(t)$ and $u_k(t)$. At the bottom of layer k , set $d_k'(t)$ and $u_k'(t)$ as the down-going and up-going waves respectively. As the waves propagate through the medium there is no absorption in the layer. The down-going and up-going waves are formulated as:

$$d_k'(t) = d_k\left(t - \frac{1}{2}\right) \quad (4.11)$$

$$u_k'(t) = u_k\left(t + \frac{1}{2}\right) \quad (4.12)$$

At the interface k , the waves correspond to the up-going, $u_{k+1}(t)$, and down-going, $d_{k+1}(t)$, in the next layer.

The wave $u_k'(t)$ consists of a contribution due to the reflection of $d_k'(t)$ and contribution due to the transmission of $u_{k+1}(t)$, then $u_k'(t)$ yields:

$$u_k'(t) = R_k d_k'(t) + (1 - R_k) u_{k+1}(t) \quad (4.13)$$

Where R_k is reflection coefficient for the k^{th} layer.

We can also arrange the equation for $d_{k+1}(t)$ which consists of a contribution due to the reflection of $u_{k+1}(t)$ and a contribution due to the transmission of $d_k'(t)$;

$$d_{k+1}(t) = -R_k u_{k+1}(t) + (1 + R_k) d_k'(t) \quad (4.14)$$

Eqs. (4.13) and (4.14) are valid for pressure amplitude and particle velocity amplitude conditions.

4.4.4. Traveltime formulation

Traveltime is the time spent traveling from source to receiver, (Figure 4.10). For zero offset, traveltime can be calculated from layer thickness and velocity. For many layers the traveltime is calculated as

$$t_k = t_{k-1} + \frac{2h_k}{V_k} \quad (4.15)$$

Where t is traveltime, h is layer thickness, and V is layer velocity. Figure 4.10 also shows the layer interface numbering scheme.

Simulation examples

This study shows two 1-D seismic simulation examples. The first example is taken from Robinson's analysis and the second example is obtained by setting the arbitrary earth's geological model. Both of the examples present the synthetic seismogram resulted by applying the lattice formulation.

Robinson's model (1983)

Robinson generated the synthetic seismogram at the surface and in the subsurface by inputting the reflection coefficients for four interfaces. Those reflection coefficients are assigned as c_1 , c_2 , c_3 , and c_4 with values 0.1, -0.2, 0.3, and -0.4 respectively. Assuming the reflection coefficient at the surface, c_0 , is -1 (i.e., for the interface between the earth and sea), he calculated each upgoing and down-going particle velocity amplitude at equal time, (Figure 4.11). The computer code is presented in Appendix A.

Earth's geological model

The purpose of this study is to verify Robinson's method and apply the method for the pressure amplitude conditions. Earth's model is set-up by giving different velocity and thickness for each layer (see Table 4.1).

Following the lattice formulation two synthetic seismograms are obtained for up-going and downgoing waves for pressure amplitude conditions. The detail of each up-going and down-going wave calculation is presented in Appendix B and the diagrams of both the results are shown in Figure 4.12. The spikes of the synthetic seismogram are convolved with Ricker wavelet as illustrated before.

Use of amplitude for seismic interpretation

The energy recorded is a function of the spatial attenuation factors, the partition of energy at the reflection boundaries, and the geometrical spreading. If a proper correction can be made, the amplitude is useful for subsurface interpretation.

Currently, large 3-D seismic surveys are routinely used to assist in the development of the hydrocarbon reservoirs. Seismic attributes, especially amplitude, are now easily mapped. It allows visualizing subtle tectonic and structural elements, which are not obviously seen on time-structure maps.

Stratigraphic, structural, tectonic, and fluid distribution information may be obtained by displaying the amplitude variation $A = A(x, y, t)$ (Enachescu, 1991) associated with a particular reflective surface. Horizon amplitude map is the most powerful tool available to the amplitude interpreter. Dependence of the amplitude response on the shape of the reflector provides important geometrical constraints on the interpretation.

Amplitude interpretation can reveal useful information concerning reservoir properties such as porous intervals, gas-filled reservoir, fluid contacts, distribution and continuity of reservoirs, reservoir heterogenites and abnormal reservoir pressure. The nature of the reservoir boundaries (unconformity, faults and stratigraphic barriers) can also be inferred from the amplitude analysis.

SEISMIC VELOCITY

Introduction

The measurement or estimation of velocity is a major task in the processing and interpretation of seismic data. The velocity can be estimated from the reflection traveltimes data because the velocity is implicitly contained within the acquired reflection traveltimes data. However the estimated velocity is not exactly the same as the true velocity of seismic propagation in the subsurface medium. In the interpretation stage, velocity itself contains information about lithology and stratigraphy. Marsden (1993) emphasized velocity as a common element for the following processes: depth map, pressure prediction, lithology prediction, and depth migration.

Velocity analysis based on normal moveout (NMO) equation has serious limitations when using the parameter in the processing and interpretation. The main reason is that the velocity analysis using the hyperbolic moveout formula has conditions, which are too

restrictive to yield useful information. The main factors affecting velocity determinations are:

- 1) Acquisition error (due to irregular survey geometries and uncontrolled acquisition parameters).
- 2) Processing error (all processings, which influence the resolution, related to velocity analysis).
- 3) Noise (coherent and random noises).
- 4) Errors related to wavelet form (not to exceed 0.5-1.0 %).
- 5) Errors related to wave propagation (refracted path, anisotropy, multiples, dipping interfaces, diffractions, etc.).
- 6) Velocity and structure variations in the ground (statics, velocity heterogeneity within a layer, dipping layer).
- 7) Subjective error (interpreter's experience in identifying seismic events).

There is no general mathematical expression for representing errors in estimating velocity. However, accuracy studies of velocity estimation have been done by several researchers with their own specifications. Landa et al. (1991) mentioned that the error in velocity estimation is weakly dependent upon the form of the interfaces. For example, when the dip changed from 10 to 40 degrees, the value of velocity error increased by 20 percent. Also Sorin (1995) studied the accuracy of velocity estimation in 3D layered structures formed by plane and curved interfaces. When he used the coherency inversion for 3-D velocity estimation, he found that the layer velocity resolution is affected by the refractor shape. Blackburn (1980) used various geological model to show errors in stacking velocity due to geological complexities including dip and velocity structure. In this report, we analyze the errors related to velocity and structure variations which are main factors, affecting the velocity estimations.

Seismic velocity and its errors

Before discussing the error in velocity estimation, the terminology of velocity will be reviewed briefly because of the confusion in terminology and concepts, even in the geophysics world. Seismic velocity is an intrinsic physical property of material in which seismic waves propagate. Different wave types propagate through the same medium with different velocities (Marsden, 1993). Various wave types are air waves (P-waves), surface waves (Rayleigh, hydrodynamic, Love), interface waves (Stoneley), body waves (P- and S-waves), and channel waves (quasi-Rayleigh, quasi-Love). In our exploration seismic, we have conventionally confined ourselves to using the P-waves. Seismic velocities are affected by other physical properties such as lithology, pore size, fluid, depth of burial, and different pressure. The seismic velocities are obtained from laboratory measurements, sonic logs, and analysis of seismic reflection data. A statistical

representation for velocity is needed due to the different scale of measurement, even though there are mathematical complexities of our treatment of velocity. For example, interval velocities are measured within a given interval from core samples in the laboratory and/or by sonic logging in the well with a measurement scale of less than 2 ft, which is much less than the wavelet scale of seismic exploration (≈ 50 feet).

Seismic velocities described here are interval velocity (V_i), average velocity (V_a), normal moveout velocity (V_n), stacking velocity (V_s), and rms velocity (V_r). Detailed mathematical definitions and relationships between these velocities can be found in Al-Chalabi (1994), Amery (1993), and Byun (1990).

Interval velocity (V_i) is the interval-time-weighted average of the velocities of all the lamina that compose the unit. Interval velocity can be obtained from a sonic log usually with 2 ft spacing, although the V_i is obtained by averaging the sonic log velocity.

Average velocity (V_a) is the average of all the interval velocities from the surface to the depth of a particular horizon. It can be measured during a velocity survey, which is performed by measuring traveltime shot from the surface to the geophone in the borehole in the vertical direction. Therefore V_a , a true vertical velocity, is used for depth conversion rather than NMO calculation.

NMO velocity (V_n) is defined by the reciprocal of the square root of the slope of the t^2 - x^2 curve at zero offset (where t is 2-way traveltime and x is offset distance). V_n is a function of subsurface geologic parameters in a certain form (Shah, 1973). The parameters are the emergence angle and wavefront curvature of the normal incidence ray at the zero-offset point.

Stacking velocity (V_s) is an apparent velocity chosen to fit the data using the NMO equation, so no corresponding mathematical expression can be given. V_s is often used synonymously with NMO velocity. V_s is some average moveout velocity which gives optimum stack. With short width of gathering, V_s may serve as a good approximation to NMO velocity (Shah, 1973). V_s increases with increased spread length or with increased dip of the reflector. V_s is greater than V_a because raypaths follow a minimum time path (Fermat's principle) rather than a minimum distance path for the multilayer model.

RMS velocity (V_r) is a mathematical quantity with no physical meaning. It is not a measurable parameter. It is the truncated product of expansion of the squared traveltime in the infinite series of squared offset distance.

Errors in estimating the RMS velocity from seismic data can be attributed to statistical and nonstatistical sources (Schneider, 1971).

Statistical errors: The more accurate RMS velocity is obtained when a data set has

- 1) higher S/N ratio (improvement by square root of number of folding traces)
- 2) wider frequency bandwidth
- 3) longer spread length (depends on the square of the spread length)

Minimum statistical error in estimating RMS velocity from seismic data occurs when offset approximately equals depth (called X/D concept).

Nonstatistical errors: The sources of the nonstatistical error may be the limiting factor.

- 1) Dip effect (correction factor fails when inflecting dip occurs)
- 2) Geometric errors (feather, not straight survey-line; can be reduced in processing stage)
- 3) Complex horizon sequence (due to thin layers)
- 4) Multiple reflection (interference with primaries)
- 5) Near-surface velocity (smoothing the estimated RMS velocity)

Procedure

In this report, we use the NMO equation to estimate velocity, which is a stacking velocity (or NMO velocity), based on the definition discussed earlier. The procedure for estimating errors in stacking velocity are followed:

- 1) Set up the model.
- 2) Forward modeling using a ray tracing method.
- 3) Apply the Least-Square method to estimate velocity.
- 4) Estimate errors correspond to different degrees of heterogeneity and non-flatness.

The method of Langan et al. (1985) is used for the ray-tracing because of its efficiency, flexibility, and compatibility. The Langan method used here can be replaced by any other ray-tracing methods for the purpose of the forward modeling. Rays are traced through a velocity field discretized into cells. Each cell has a velocity gradient calculated from given velocities. Velocity gradients are used algebraically to produce traveltimes, orientation vectors, and position vectors. To find the ray path between two fixed points, a shooting method is employed. Topographic geometries for shots and receivers are expressed by smoothed line segments, and arbitrary locations of shots and receivers are allowed. Reflecting interfaces are defined by a cubic spline within each cell because the reflecting interfaces should be smooth and continuous across cell boundaries. For the continuity of the reflecting interfaces, the polynomial extrapolation and interpolation are applied where reflecting points do not exist during the forward modeling.

The NMO equation is applied to synthetic traveltime data obtained through the ray tracing:

$$t^2(x) = t^2(0) + x^2/v^2, \quad (4.16)$$

Where x is distance (offset) between the source and receiver positions, v is the velocity of the medium above the reflecting interface, $t(0)$ is twice the zero-offset traveltime. The Least-Square method is used to estimate the velocity.

An error factor is adopted from Blackburn (1980). According to his definition, the error factor to ascertain the variations in stacking velocity is the ratio of the difference in stacking velocity as derived from ray modeling and the average velocity at the location to the stacking velocity. The error factor is expressed as a percentage. Variations in error factors and their magnitudes will represent the area having velocity problems.

The Fortran code is described in Appendix C

Synthetic examples

Simplified models are set up to isolate the problems associated with geological complexities in estimating stacking velocities. In this project, six models are designed based on the factors affecting velocity determinations discussed before. All models have 100 x 40 cells and a cell has 10 x 10 m in dimension except depth model (10 x 20 m). Distances of far- and near-offset are 300 and 20 m, respectively. Each cmp has 100 m interval and a cmp has fifteen traces. The raypaths and velocities for our six models are shown in the Figures 4.13- 4.18 (a). The models and their configurations are followed:

1. *Depth model*: 7 different flat beds (55, 155, 255, 355, 455, 555, 655 m); constant velocity (1000 m/s); 1 cmp per each depth model; Figure 4.13-(a).
2. *Dip model*: a dipping bed (11°); constant velocity (1000 m/s); 7 cmps; Figure 4.14-(a).
3. *Lateral velocity-variation model*: a flat bed (355 m); lateral gradient k^* (0, 1, 2, 3, 4); 1 cmp; Figure 4.15 -(a).
4. *Vertical velocity-variation model*: a flat bed (355 m); vertical gradient k (0, 1, 2, 3, 4); 1 cmp; Figure 4.16 -(a).
5. *Trap model*: a flat bed (355 m); pinching-out velocity anomaly; 7 cmps; Figure 4.17 (a).
6. *Near-surface anomaly model*: a flat bed (355 m); near-surface velocity anomaly; 6 cmps; Figure 4.18 -(a).

* $\text{velocity}(x) = \text{initial velocity}(x=0) + k*x$

Two observation methods are considered for our specific purpose of estimating errors in determining velocities: t-tnmo and error factor. The t-tnmo is a RMS value of residuals which is discrepant values between traveltimes from a ray tracing and traveltimes obtained from stacking velocity through NMO equation. High value of t-tnmo represents that the NMO equation does not work well for a given model. However it does not mean that a small t-tnmo always implies well estimated velocity. For example, dip model and velocity-variation models (lateral and vertical both) have relatively small range of t-tnmo values. Estimated stacking velocities are not well matched with true and average velocities as shown Table 4.2. 3-D diagrams showing t-tnmo against offset, depth, k , and cmp # can be found in Figures 4.13-4.18 (b) for each models. And t-x

curves showing hyperbolic fitting are in Figures 4.13-4.18 (c). The most reliable method for examining the errors is an error factor defined earlier. Large fluctuations of the error factor always indicate the areas having problem to estimate velocities as shown in Figures 4.13-4.18 (d).

Our models can be grouped into three based on their characteristics: (1) *interface change* (depth model and dip model), (2) *gradient velocity change* (lateral and vertical velocity-variation model), and (3) *abrupt velocity change* (trap and near-surface anomaly model). I summarized the results with following table 4.3 showing averaged numbers of t-tmo and error factor.

The group of interface is very reliable to estimate stacking velocity even with a dipping layer. In the group of gradient velocity change, lateral velocity changes are more serious in deterring stacking velocities than the vertical change. The worst case occurs in the group of abrupt velocity change. Considering that lateral velocity change with large gradient number is the special case of the group of abrupt velocity change, abrupt velocity change is the most significant factor affecting velocity determinations.

SEISMIC INTERPRETATION

Introduction

Horizon interpretation of the seismic data combined with the well information and the geological information of the area can help in developing a unified model of the reservoir system in the study area. The seismic data provide considerable amount of extra information in the interwell areas and can give a more accurate information on the location of the boundaries of the thin-bed compartment.

The 3-D seismic data under study comes from Stratton field, South Texas. It is confined to 2 square miles. It has 3 seconds of data with a sampling rate of 2 ms. It comprises 100 inlines (oriented east-west) and 200 crosslines (oriented north-south) with a trace spacing of 55 ft. A larger seismic 3-D data has been acquired in this area but is not available to us yet. The seismic sections show a smooth, non-faulted area almost throughout the whole seiscube above 1700 ms. Below that area and on some of the sections we can detect a few faults. The VSP calibration is very important to identify the faulted horizons. We see from (Hardage et al, 1994) that the undeformed area is the upper Frio and a large part of the middle Frio formation. Few faults extend from lower Frio to middle Frio formation. The quality of the data set is good in general and the target horizons appear to be clear on the seismic sections. The interpretation has been performed using Landmark interactive interpretation system.

Interpretation procedure

The main challenge that we are facing in this area is the thickness of the target horizons. The Frio reservoirs that we are interested in are very thin (less than 15 feet). Also the separation between them is very small (10 to 15 feet). These conditions require extra attention to get the precise calibration of the stratigraphic depth with seismic reflection time. For the two main target reservoir which are E41 and F11 the *events* on the seismic sections appears to be clear throughout most of the 3-D seismic set. As for the E 49 reservoir it appears to be discontinuous over the area and that makes it untrackable. The fact that E49 can not be tracked in 3D dose not mean that it is useless in the seismic sense. Actually the information that 3D provide on this reservoir can prove to be useful in the context of the comprehensive model, taking into consideration the resolution of the seismic data.

The procedure that we followed to get the product maps for E41 and F11 reservoir is as follows:

- a. Identify the target horizons on the seismic section using the VSP data from well #9. The location of this well is defined on the base map of the seismic data as the intersection of line 80 (E-W) and trace 89 (N-S). Figure 4.19 shows the location of the well on the base map.
- b. Track the target horizons in 2D sections using autodip-tracking mode, which following the actual dip of the horizon. This procedure aims basically to define seed points for the target horizons over the whole area to improve the 3D tracking.
- c. Perform 3D auto tracking for the target horizons.
- d. Produce time structure maps and amplitude maps for both horizons.

Interpretation

The seismic section that shows the target horizons is in Figure 4.20. This seismic section is for line #1 on the base map and shows clearly the time interval that separates the base of reservoir F11 from the base of reservoir E41. We can see these horizons in prospective view in Figure 4.21 which represent a 500 ms section from the 3D seismic cube between 1400 and 1900 ms. In Figure 4.20 we can see a *tuning effect* which is a signature mark for thin beds.

The tuning effect can be summarized as follows: if the bed is thick enough to be resolved by seismic, we can see distinct positive and negative reflections proportional to the reflection coefficient. If the thickness of this bed decreases, the amplitude of the reflection from the top of the bed will interfere with the reflection from the bottom of the bed causing the tuning effect. The amplitude of the result of the interference reaches a maximum for a thickness of $\frac{1}{4}$ of the wavelength, which is called the tuning thickness. For beds thinner than the tuning thickness the shape of the wavelet is the same but with less amplitude content. The tuning thickness depends solely on the wavelet frequency.

The black rectangular in Figure 4.22 highlights an example of the tuning effect on the sections.

Even though we can not map these spots that we believe that it is the E41 reservoir, we can see these *patches* all over the 3D cube (Figure 4.22).

The bright spot indicated by the arrow in Figure 4.22 is emphasized in the two time slices in Figure 4.23. We can see clearly how this *compartment* stands out in the time slice where we can determine the spatial diameters for this compartment.

These results can prove to be very helpful if we put into our consideration the following considerations and restrictions:

- (a) The seismic data covers only a small section of the field, and we might need to interpret a larger data set to generalize our conclusions.
- (b) Most of the time the compartments thickness is thinner than the seismic resolution. That means any visible compartments are only the ones that are thicker than the seismic resolution and the result achieved can not be extended to the smaller compartments through the conventional seismic interpretation.
- (c) A better mapping of the channels can be achieved using the procedure proposed by Hardage et al. (1994) to get the original depositional surface. The seismic flattening technique is not available to us.
- (d) Reservoirs E41 and F11 appear to be continuous over most of the area from the time structure map (Figures 4.24 and 4.25).

The high values are in dark blue. A possible local dome closure is indicated in Figure 4.24.

The empty patches on the time structure maps are due to the tracking technique of the software we are using; the empty points did not meet the tracking parameters defined by the interpreter. Figure 4.25 shows a time structure map for the E41 reservoir. We need to keep in mind that these are time structure maps and not depth maps. Without an independent source of velocities that can enable us to perform a time-depth conversion, these horizon maps can be misleading.

The amplitude maps give a good idea about the amplitude content of the horizons, lithology and possible hydrocarbon accumulations. Figure 4.26 shows the amplitude map of F11 reservoir. The high amplitude values are in red and the low values are in blue.

The lateral changes in amplitude can mean a potential area of prospect that needs to be studied closely with the integration of other available data.

The amplitude map for the E41 horizon is shown Figure 4.27. This map shows less variation than the F11 horizon, but it has some promising areas, especially the ones that has the bright spots possibly associated with hydrocarbon accumulations.

CONCLUSIONS

Estimation of seismic amplitude and velocity are very important to characterize a reservoir and these are the main topics considered here. Seismic amplitudes is a result of several factors some of those are coupled with data acquisition and processing. They can

not be estimated correctly. Thus, traditionally 1-D synthetic seismogram is generated in x, y space and compared to the seismic processed real data. Although it is important to study the seismic amplitude and its uncertainties, we limited our analysis to study the amplitude due to transmission and multiples only.

A significant confusion arises in the literature as to the seismic measurements, seismic particle velocity or pressure response. The physics behind these two is different, thus, a different response is expected for both measurements. For land data, the geophons respond to particle velocity and for marine data the sensors respond to the change in pressure. In order to avoid this confusion, a simple modeling study was carried out, and total response of the simple earth model is obtained.

Seismic velocity estimation is another area studied extensively. Again, the terminology is vast and confusing. If there is no well control, the seismic velocity obtained measurements is the stacking velocity and is far from accurate. Also, the accuracy changes from simple to complex subsurface geometry and for lateral velocity variation. Using simple subsurface models with varying degree of complexity, we obtained the stacking velocities and the error inherent to the estimation. The error in stacking velocity computation could be as much as 25% for complex models. The only way to reduce this uncertainty is with log correlation and migration velocity analysis.

Finally an attempt was made to interpret a 3-D seismic data from the Stratton field, south Texas, taking resolution into consideration. The dominant frequency of this set is 60 Hz. For the formation under study here t, the vertical resolution is about 10m, and the horizontal resolution is about 20 meters after 3-D migration. With this kind of resolution the smaller reservoir compartments are not well resolved.

The interpretation that has been performed on the seismic data is only a basic qualitative interpretation. The seismic data at hand provided us with some valuable information about the subsurface structure especially in the areas where we do not have any well information or geological data. The results are promising, and can be enhanced with well log correlation and depth inversion.

REFERENCES

- Al-Chalabi, M., 1994, Seismic velocities-a critique: *First Break*, v. 12 no. 12, p. 589-596.
- Amery, G. B., 1993, Basics of seismic velocities, v. 12, no. 11, p. 1087-1091.
- Blackburn, G., 1980, Errors in stacking velocity-True velocity conversion over complex geologic situations: *Geophysics*, v. 45, p. 1465-1488.
- Byun, B. S., 1990, Velocity analysis on multichannel seismic data: *SEG Geophysics reprints series* No. 12.
- Castagna, J. P., 1993, Petrophysical Imaging using AVO, *The Leading Edge*, 12, 172-179.
- Dobrin, M. B., and Savit, C. H., 1988, *Introduction to Geophysical Prospecting*, Fourth Edition, McGraw-Hill, Inc., New York, 867p.
- Enachescu, M. E., 1991, Amplitude interpretation of three-dimensional reflection data: 61th Ann. Internat. Mtg., Soc. Expl. Geophy., *Expanded Abstracts*, 275-278.
- Gadallah, M. R., 1994, *Reservoir Seismology*, PennWell Publishing Company, Tulsa, 384p.
- Hardage, B. A., Levley, R. A., Pendleton, V., Simmons, J., and Edson, R., 1994, A 3-D seismic case history evaluating fluviially deposited thin-bed reservoirs in a gas-producing property. , *Geophysics* vol. 59, no.11, P. 1650-1665.
- Landa, E., Thore, P., Sorin, V., and Koren, Z., 1991, Interpretation of velocity estimates from coherency inversion: *Geophysics*, v. 56, p. 1377-1383.
- Marsden, D., 1993, The velocity domain: *The Leading Edge*, v. 12, no 7, p. 747-749.
- Marsden, D., 1993, Velocity and the exploration cycle: a new paradigm: *The Leading Edge*, v. 12, no 7, p. 770-772.
- Ostrander, W. J., 1984, Plane-wave reflection coefficients for gas sands at normal angle of incidence, *Geophysics*, 49, 1637-1648.
- Robinson, E. A., 1983, Multichannel time series analysis with digital computer programs: Goose Pond Press, 117-148.
- Shah, P. M., 1973, Ray tracing in three dimensions, *Geophysics*, 38, 600-604.
- Shah, P. M., 1973, Use of wavefront curvature of relates seismic data with subsurface parameters: *Geophysics*, v. 38, p. 812-825.

Schneider, W. A., 1971, Developments in seismic data processing and analysis (1968-1970): *Geophysics*, v. 36, p. 1043

Schumm, S., 1981, Evolution and response of the fluvial system, sedimentological implications, in Recent and ancient nonmarine depositional environments: Models for exploration: SEPM Special Publication No. 31, p. 19-30.

Sherrif, R. E., and Geldart, L. P., 1995, *Exploration Seismology*, Second Edition, Cambridge University Press, New York, 592p

Sheriff, R. E., 1975, Factors affecting seismic amplitudes: *Geophys. Prosp.*, v. 23, p. 125-138.

Sorin, V., 1995, Accuracy study of velocity estimation in 3-D layered models: *Geophysics*, v. 60, p. 1567-1574.

Table 4.1. Earth model.

No.of layers	h(ft)	V(ft/s)
1	800	10000
2	4000	11000
3	2000	17000
4	3000	10000

Table 4.2 Estimated stacking velocities (Vstk) compared to average velocities (Vave).

(1) Depth model

depth (m)	Vstk (m/s)	Vave (m/s)
55	995	1000
155	999	1000
255	999	1000
355	1000	1000
455	1000	1000
555	998	1000
655	1001	1000

(2) Dip model

cmp #	Vstk (m/s)	Vave (m/s)
20	1018	1000
30	1016	1000
40	1019	1000
50	1018	1000
60	1018	1000
70	1018	1000
80	1016	1000

(3) Lateral velocity-variation model

Horiz. k	Vstk (m/s)	Vave (m/s)
0	500	500
1	971	1030
2	1414	1581
3	1859	2137
4	2304	2692

(4) Vertical velocity-variation model

<i>Verti. k</i>	<i>Vstk (m/s)</i>	<i>Vave (m/s)</i>
0	500	500
1	669	662
2	831	804
3	988	936
4	1138	1059

5) Trap model

<i>cmp #</i>	<i>Vstk (m/s)</i>	<i>Vave (m/s)</i>
20	974	915
30	960	914
40	999	920
50	1030	922
60	999	930
70	1018	939
80	1027	950

(6) Near-surface anomaly model

<i>cmp #</i>	<i>Vstk (m/s)</i>	<i>Vave (m/s)</i>
20	1000	1000
30	1000	1000
40	1000	1000
50	787	992
60	862	958
80	862	958

Table 4.3. Summary of error observations

MODEL	depth	dip	lateral k=4	vertical k=4	trap	near-surface
t-tnmo	1.5	0.4	1.5	0.5	5	23
error factor (%)	0.3	1.8	16	6	10	25

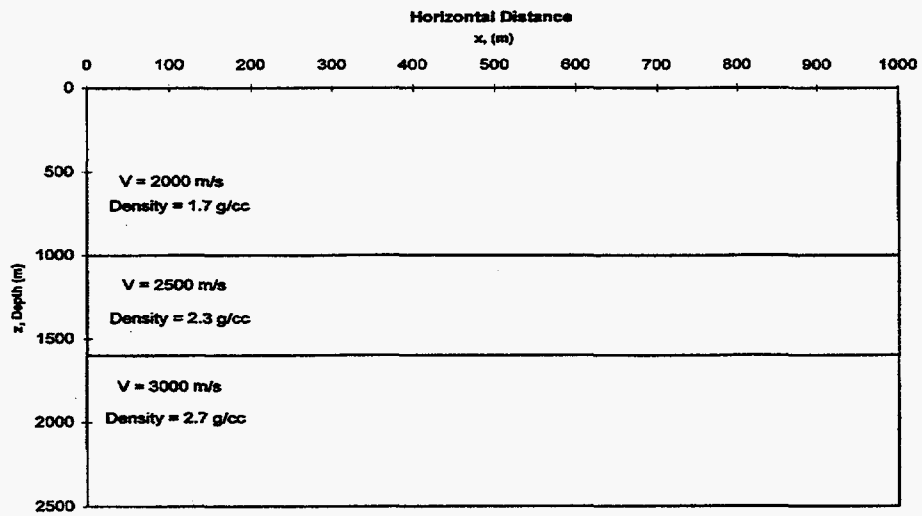


Figure 4.1. Hypothetical subsurface model with two well log information.

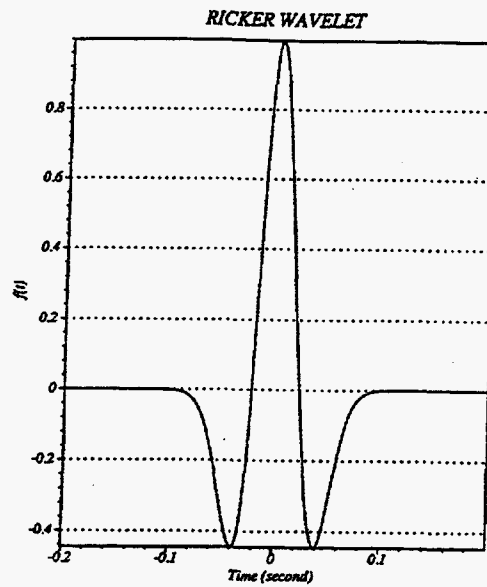


Figure 4.2. Zero phase Ricker wavelet with 30 Hz frequency.

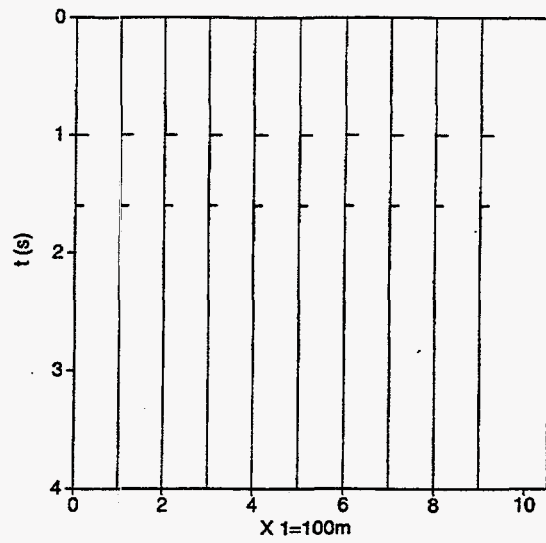


Figure 4.3. Reflectivity series of Figure 4.1.

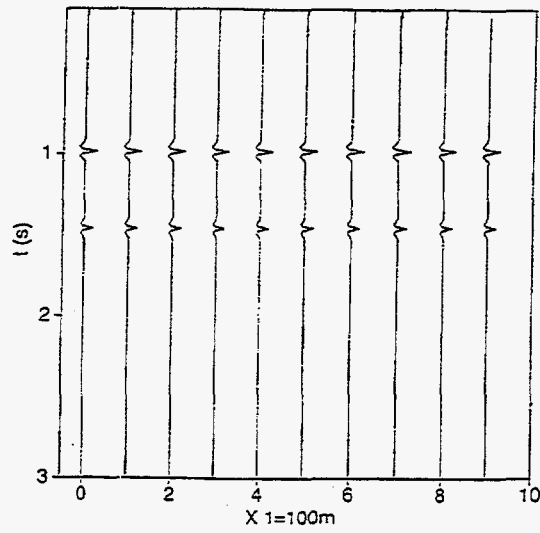


Figure 4.4. Synthetic Seismogram, after convolution of reflectivity (Figure 4.3) with Ricker wavelet (Figure 4.2).

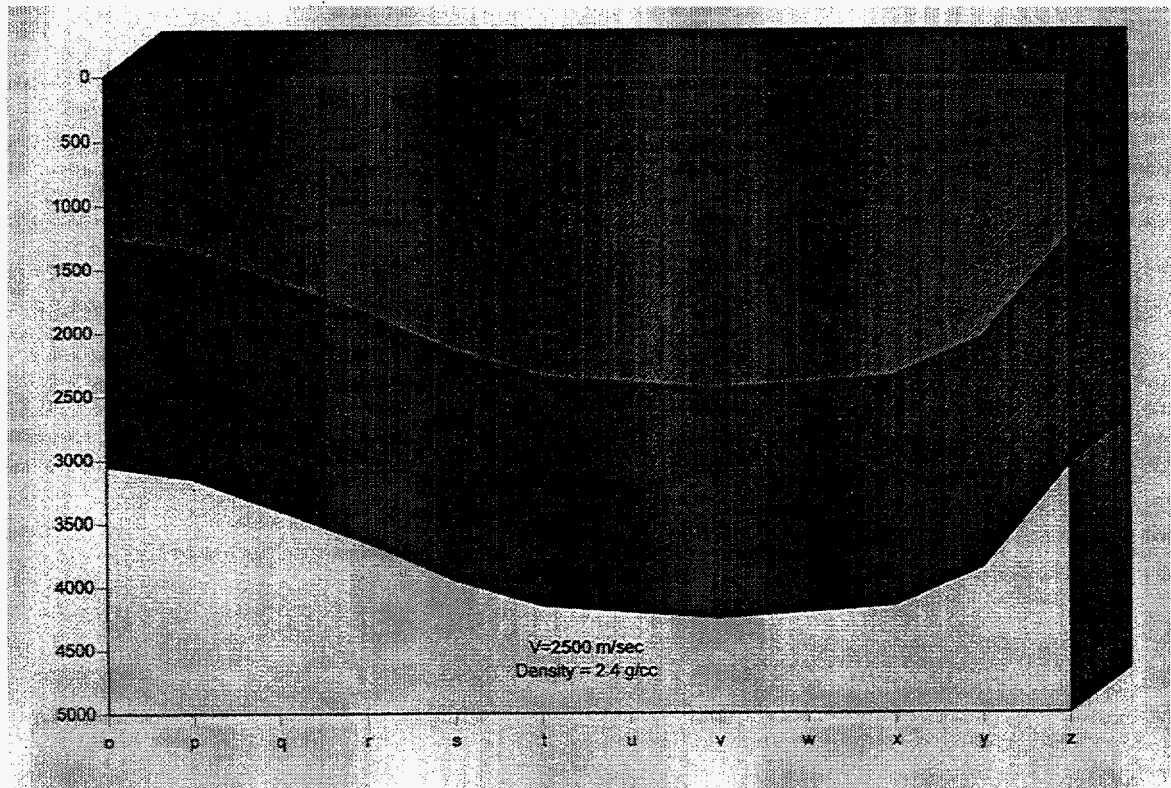


Figure 4.5. Subsurface fluvial channel.

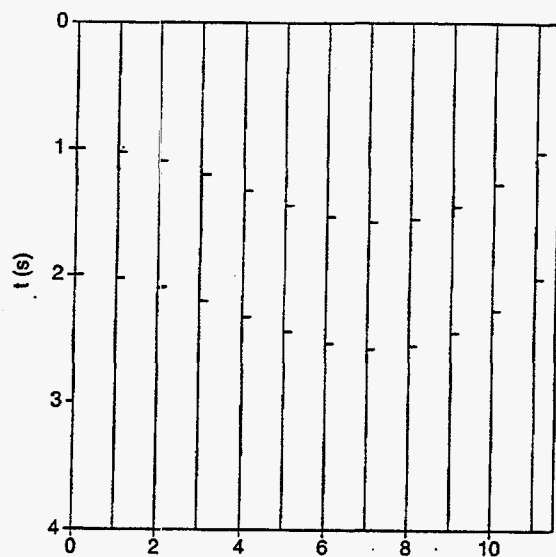


Figure 4.6. Reflectivity series for a single cross section shown in Figure 4.5.

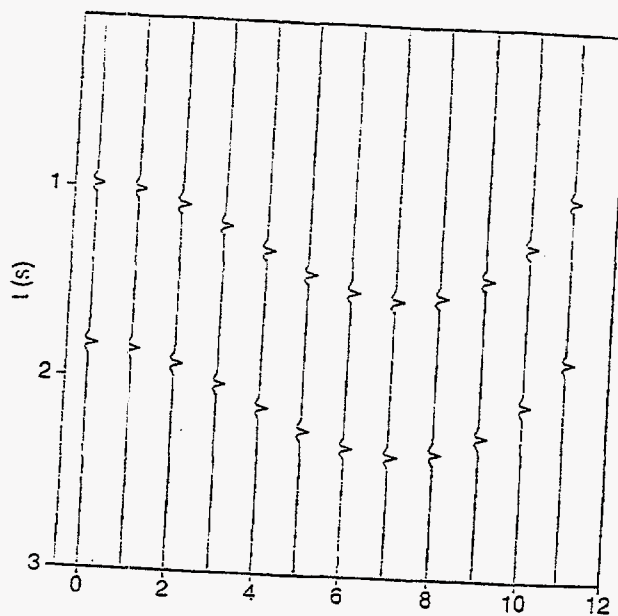


Figure 4.7. Synthetic Seismogram for a fluvial channel, obtained after convolving Ricker wavelet with reflectivity shown in Figure 6.

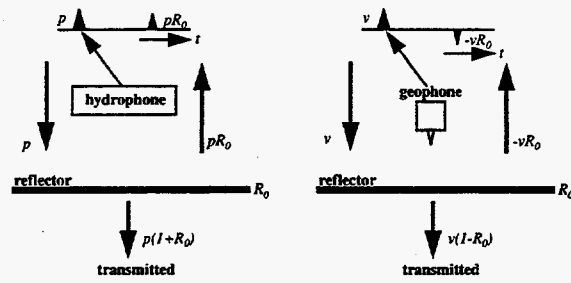


Figure 4.8: Waves propagation for marine survey (left sketch) and land survey (right sketch).

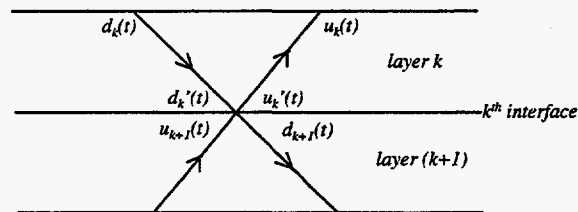


Figure 4.9: Lattice configuration for reflected and transmitted waves in layer k and at interface k .

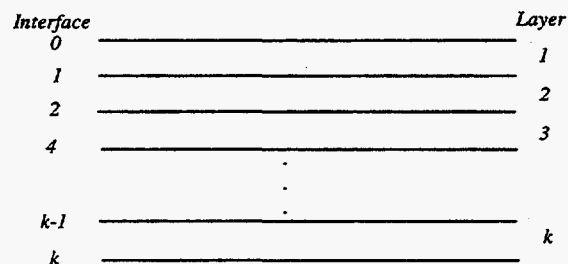


Figure 4.10: Numbering scheme for traveltime formulation.

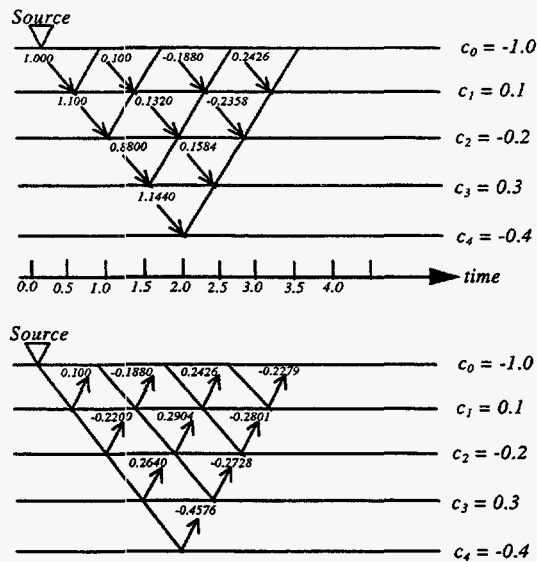


Figure 4.11: Schematic diagram of downgoing and upgoing waves at equal time for Robinson's model (Robinson, 1983).

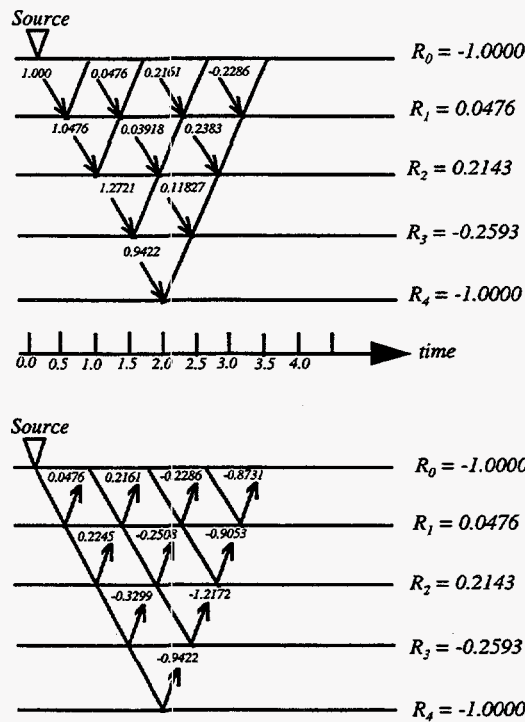
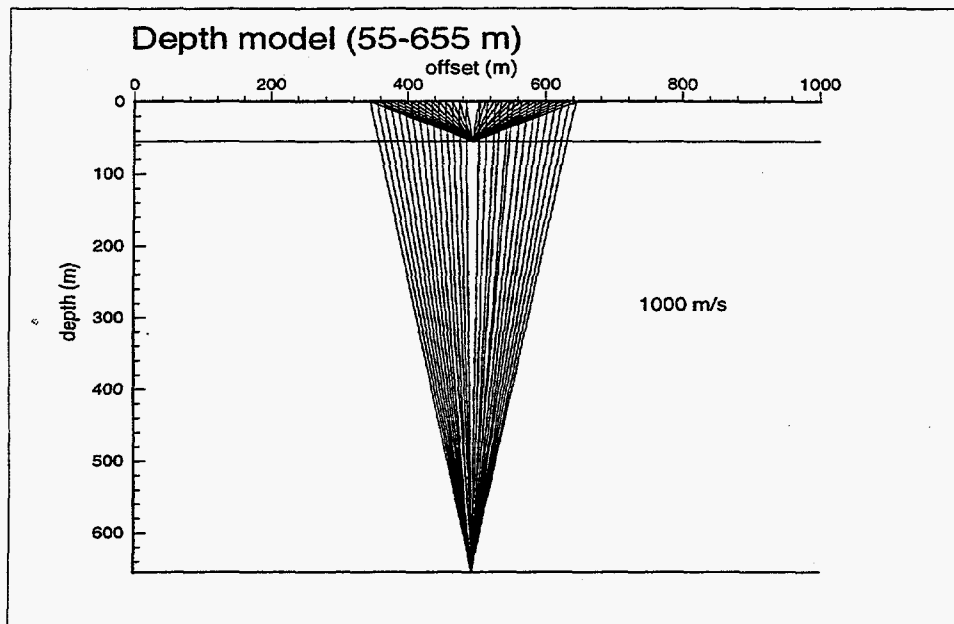
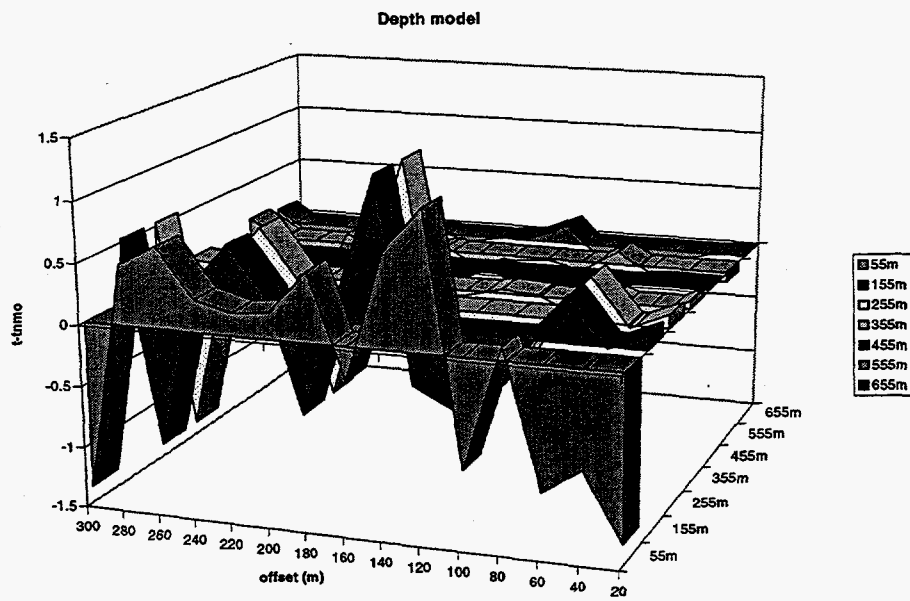


Figure 4.12 Schematic diagram of downgoing and upgoing waves at equal time for earth's model conditions.

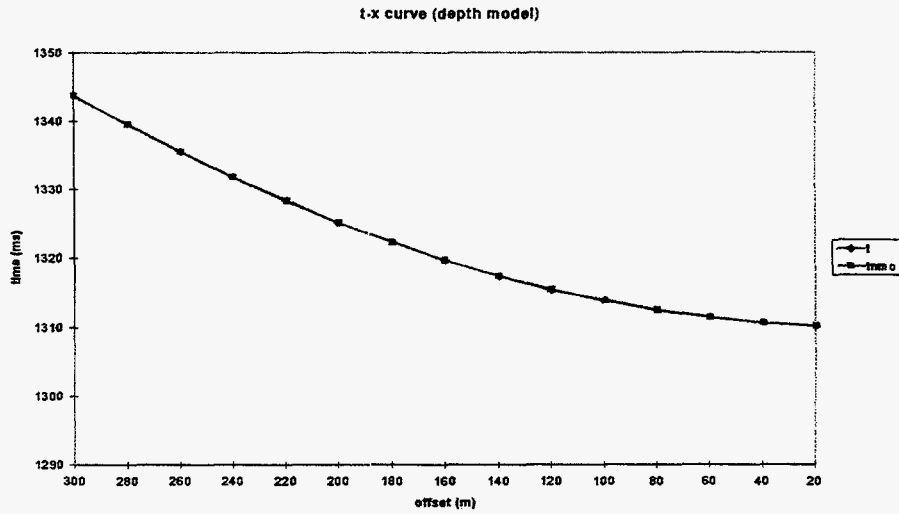
a) Depth model



(b) t-tmo (depth model)



(c) t-x curve (depth model, depth = 655 m)



(d) Error factor (depth model)

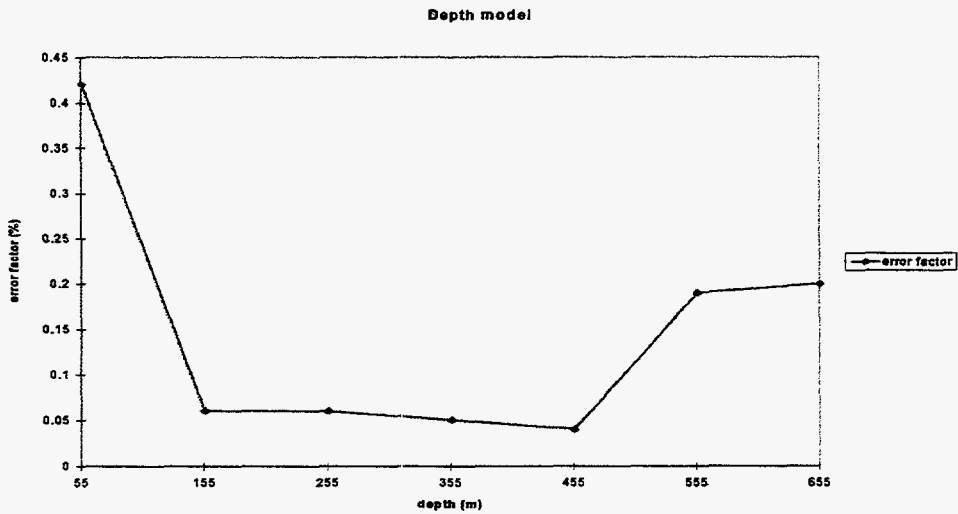
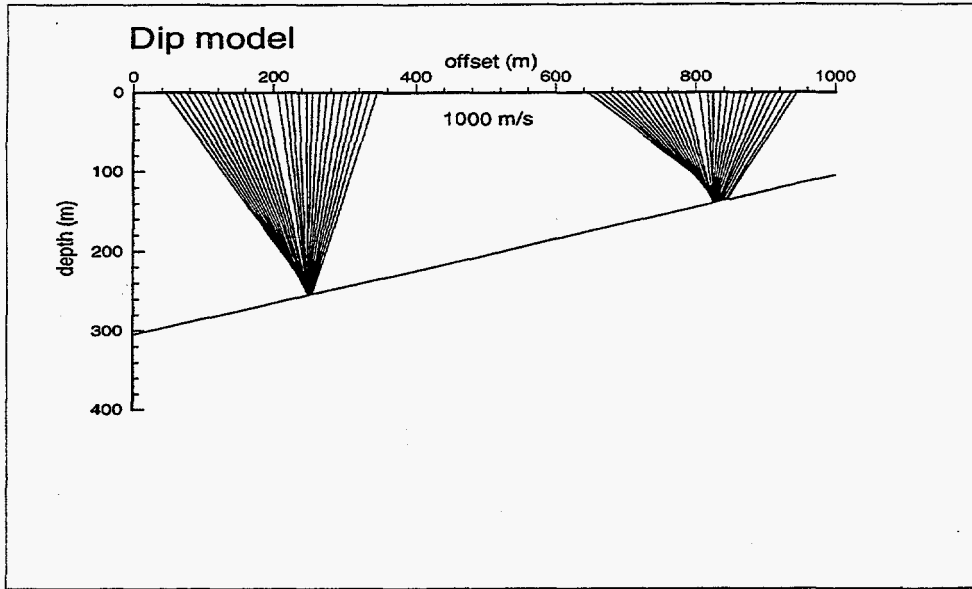
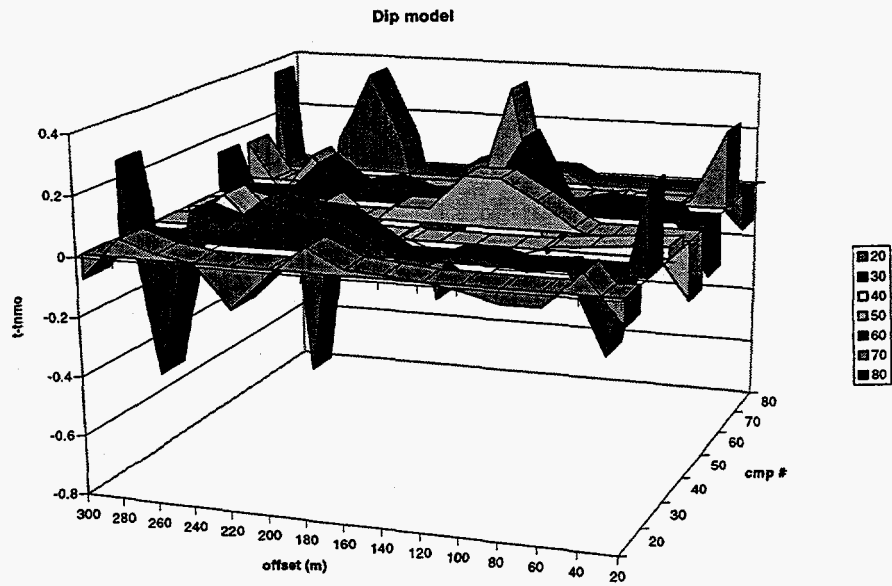


Figure 4.13. Depth model. (a) seven different depths used (55 - 655 m), (b) t-t_{nmo} for different depths, (c) t-x curve between true traveltimes and traveltimes obtained NMO equation (depth = 655 m), and (d) error factors for different depths.

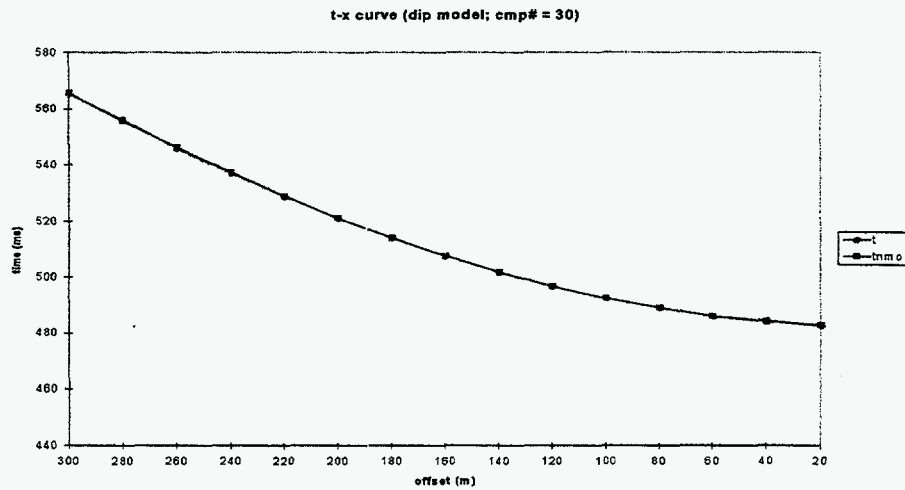
(a) Dip model



(b) t-tmo (dip model)



(c) t-x curve (dip model, cmp # = 30)



(d) Error factor (dip model)

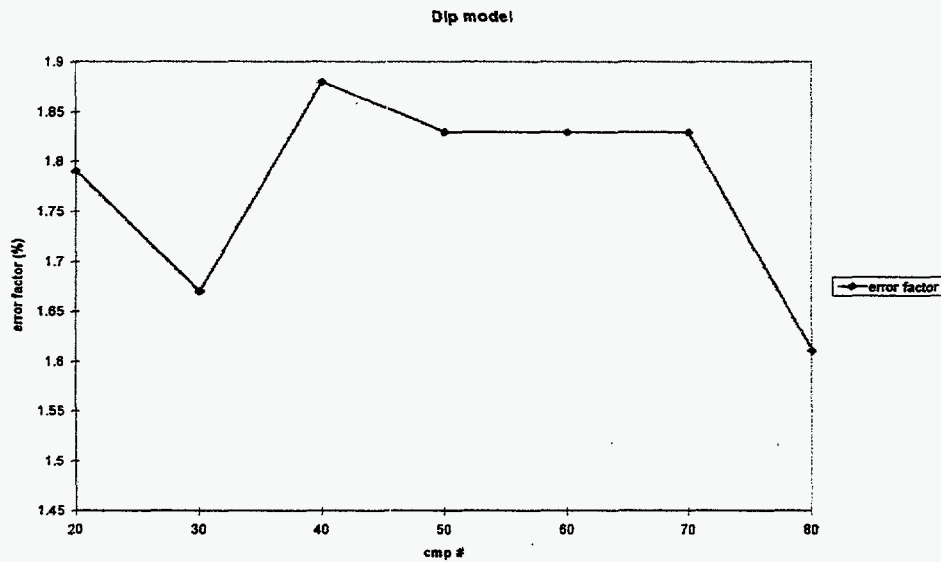
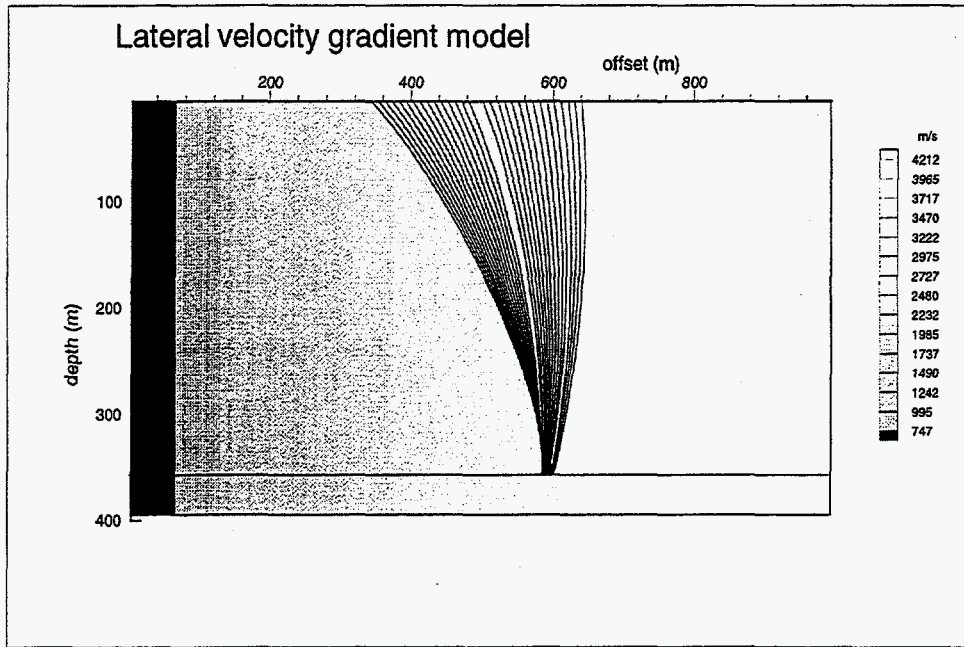
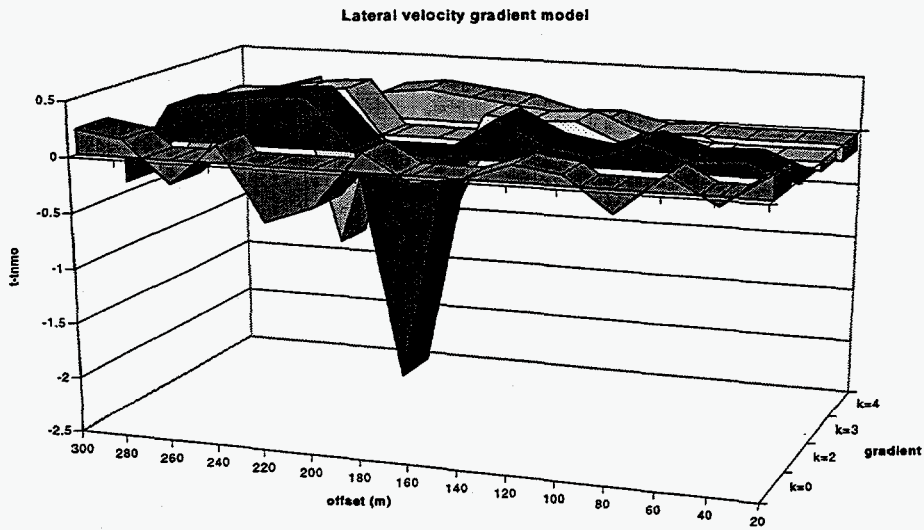


Figure 4.14. Dip model (11 degree dip). Larger number of cmp represents shallower depth. (a) seven cmp data sets used, (b) t-t_{nmo} for different cmp, (c) t-x curve between true traveltimes and traveltimes obtained the NMO equation, and (d) error factors for different cmp numbers.

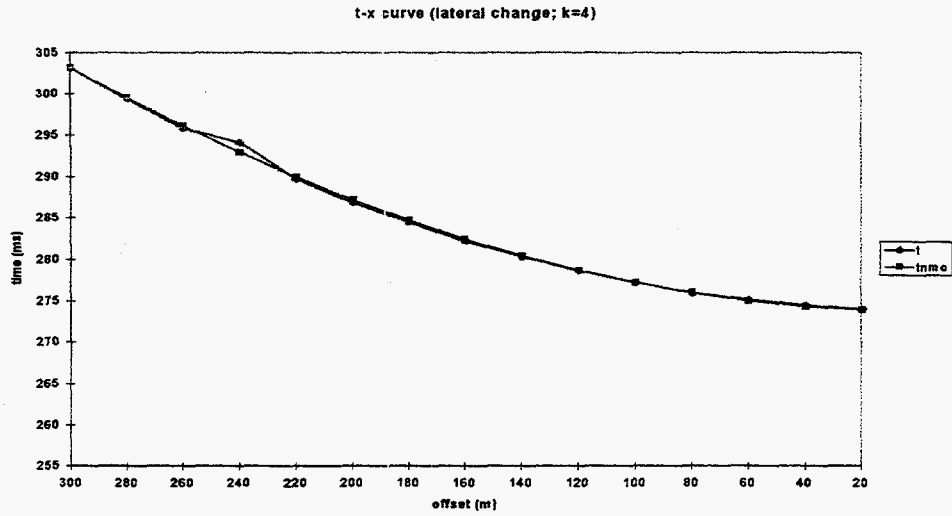
(a) Lateral velocity-variation model ($k=0-4$)



(b) t - $\ln t$ (lateral velocity-variation model)



(c) t-x curve (lateral velocity-variation model, k=4)



(d) Error factor (lateral velocity-variation model)

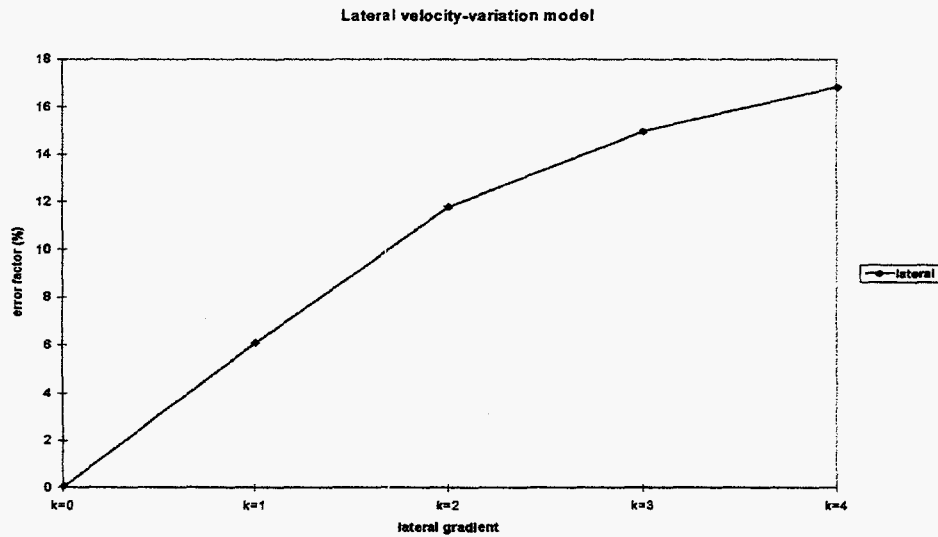
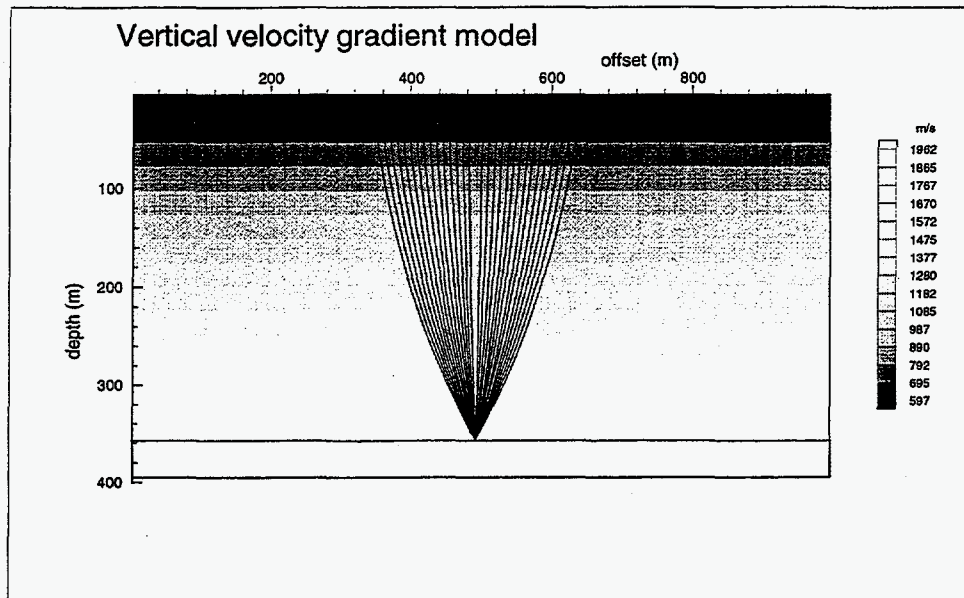
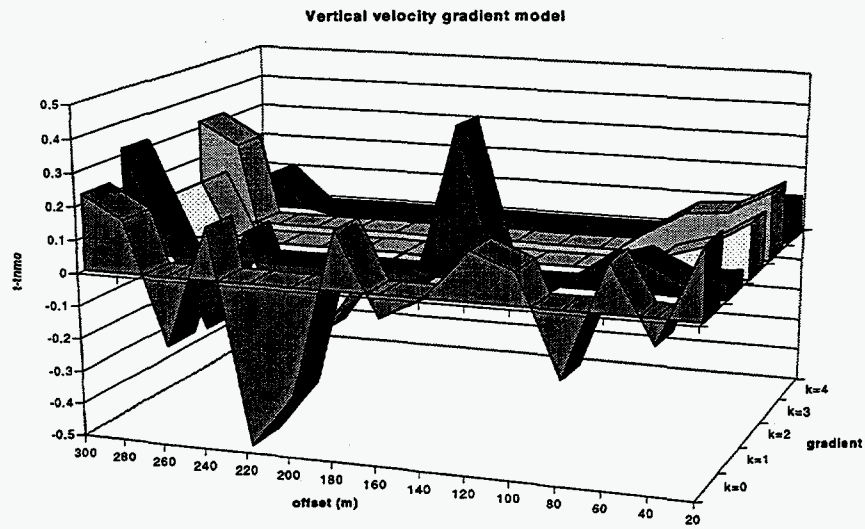


Figure 4.15. Lateral velocity-variation model. Five different lateral gradients (k) are used ($k=0, 1, 2, 3, 4$). (a) lateral velocity-variation model and curved raypaths for $k=4$, (b) $t-t_{nmo}$ for different k , (c) $t-x$ curve between true traveltimes and traveltimes obtained the NMO equation, and (d) error factors for lateral-velocity variations along different k s.

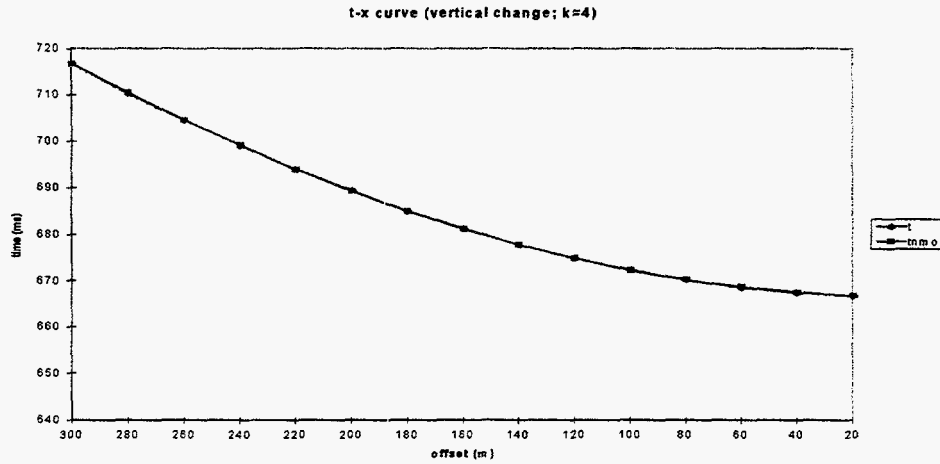
(a) Vertical velocity-variation model ($k=0-4$)



(b) $t-t_{\text{rms}}$ (vertical velocity-variation model)



(c) t-x curve (vertical velocity-variation model, k=4)



(d) Error factor (lateral and vertical velocity-variation models)

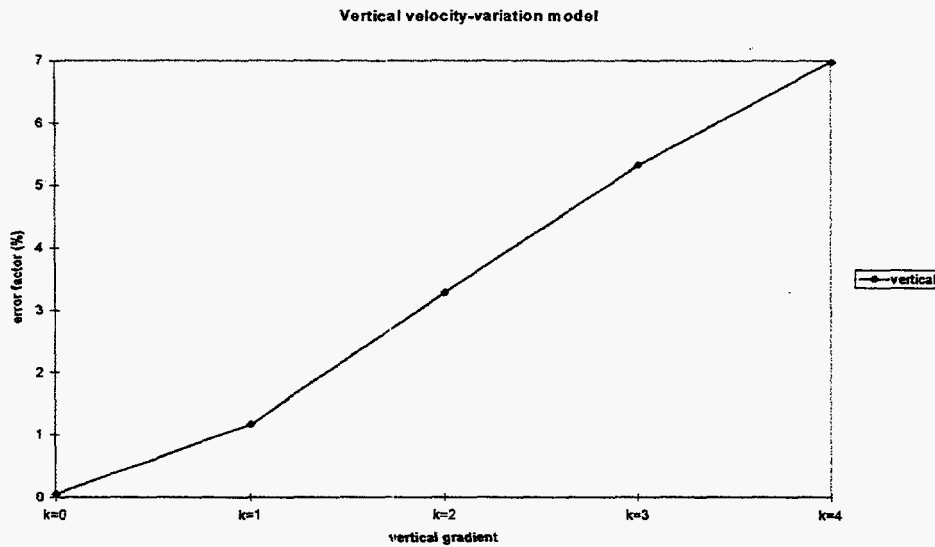
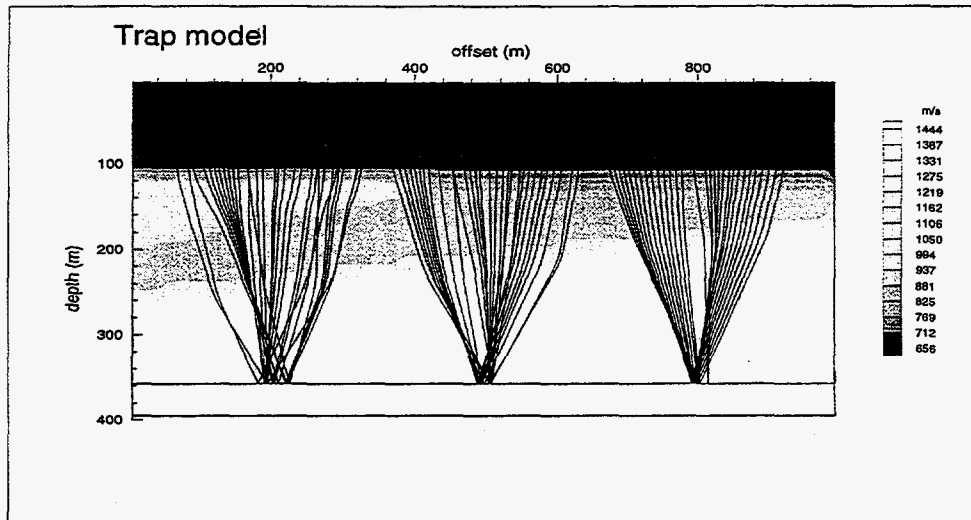
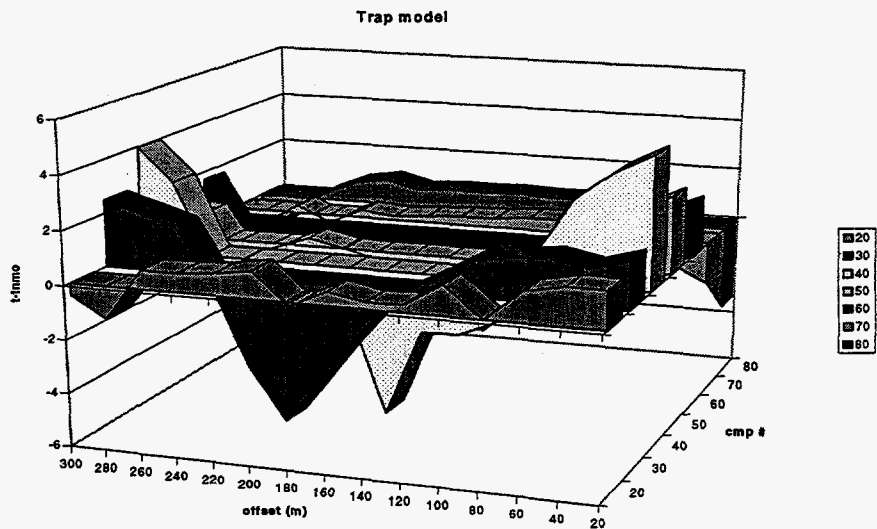


Figure 4.16. Vertical velocity-variation model. Five different vertical gradients (k) are used ($k=0, 1, 2, 3, 4$). (a) vertical velocity-variation model and curved raypaths for $k=4$, (b) $t-t_{mo}$ for different k s, (c) $t-x$ curve between true traveltimes and traveltimes obtained the NMO equation, and (d) error factors for vertical-velocity variations along different k s.

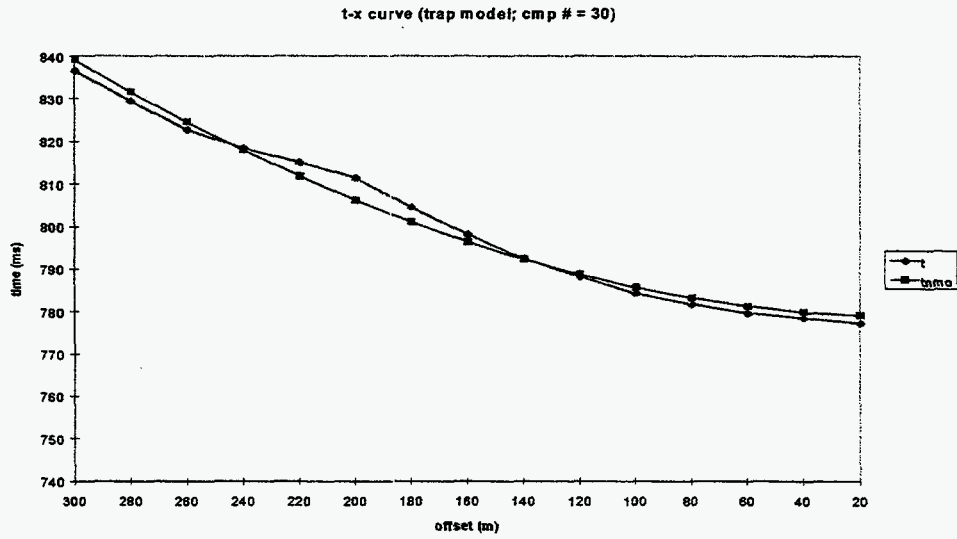
(a) Trap model



(b) t-tmo (trap model)



(c) t-x curve (trap model, cmp# = 30)



(d) Error factor (trap model)

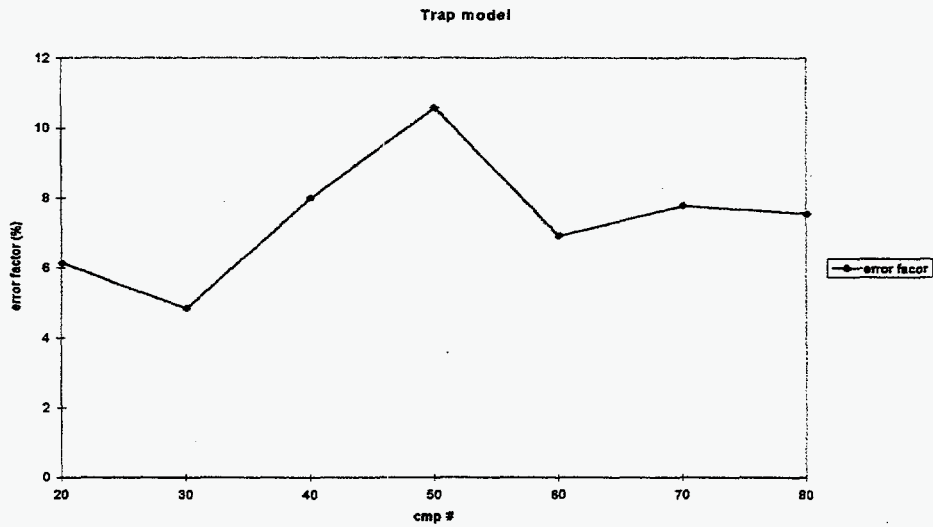
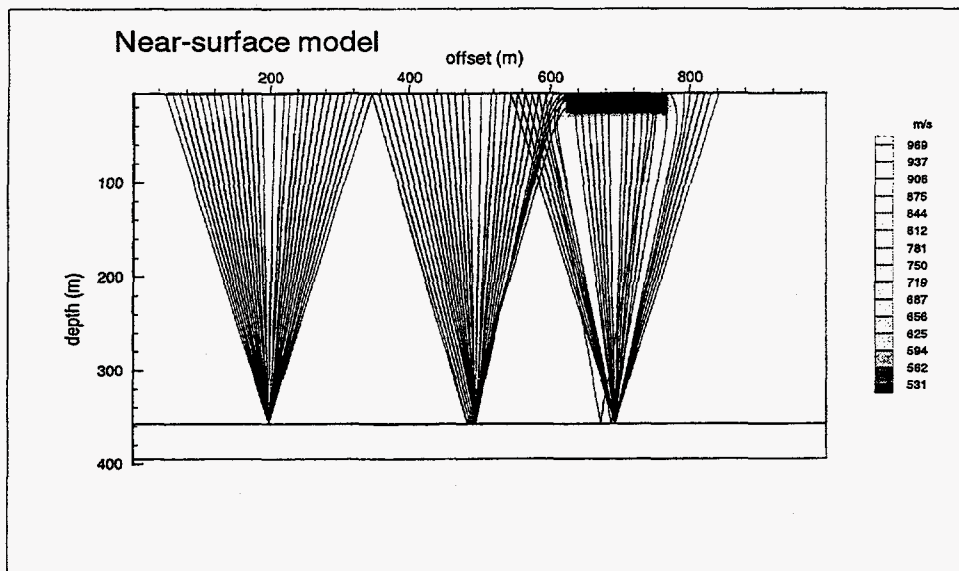
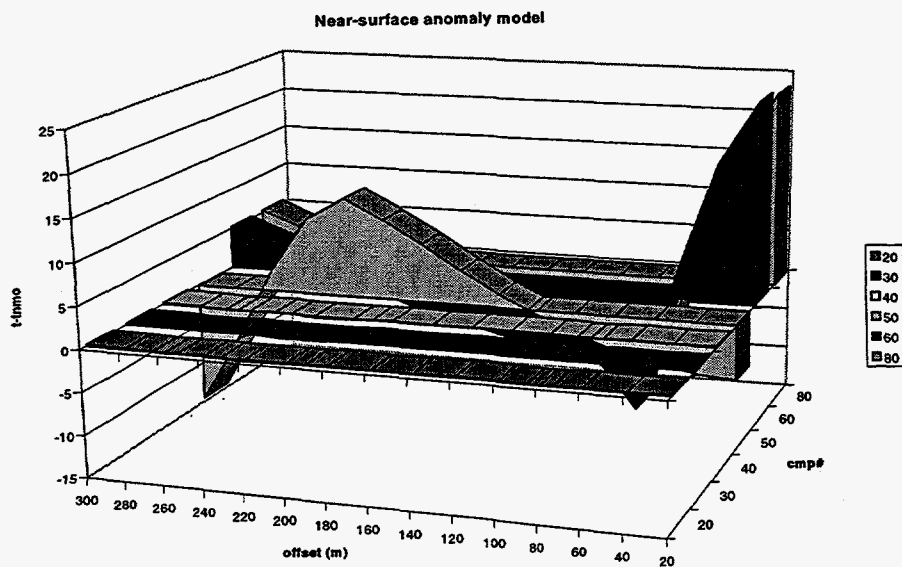


Figure 4.17. Trap model. Seven cmp data sets are used. (a) geological trap and curved raypaths, (b) t-t_{nmo} for different cmp numbers, (c) t-x curve between true traveltimes and traveltimes obtained the NMO equation, and (d) error factors for different cmp numbers.

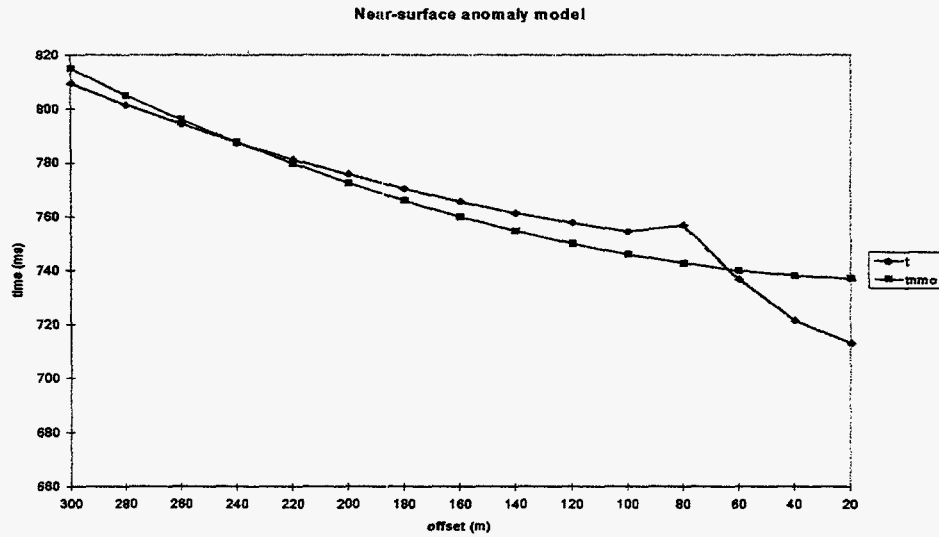
(a) Near-surface anomaly model



(b) t-tmo (near-surface anomaly model)



(c) t-x curve (near-surface anomaly model, cmp# = 60)



(d) Error factor (near-surface anomaly model)

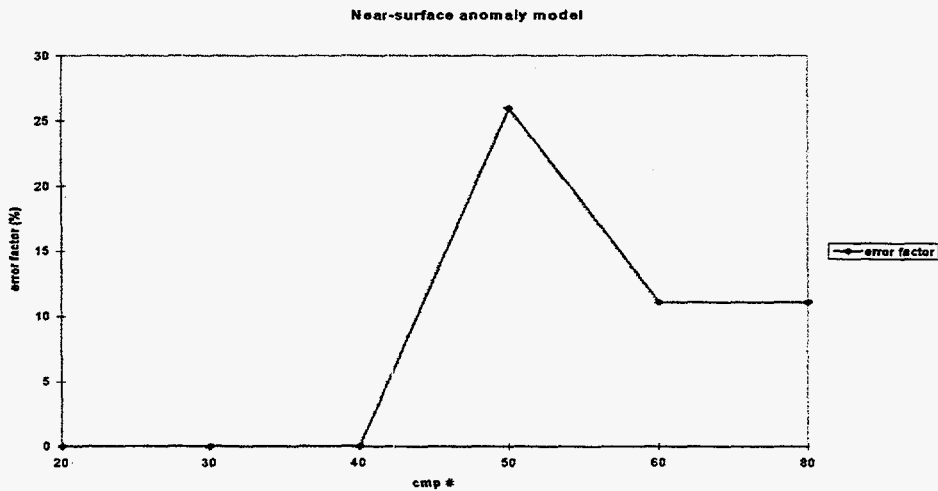


Figure 4.18. Near-surface anomaly model. Six cmp data sets are used. Background and anomaly velocities are 1000 and 500 m/s, respectively. (a) low-velocity anomaly and curved raypaths, (b) t-t_{nmo} for different cmp numbers, (c) t-x curve between true traveltimes and traveltimes obtained the NMO equation, and (d) error factors for different cmp numbers.

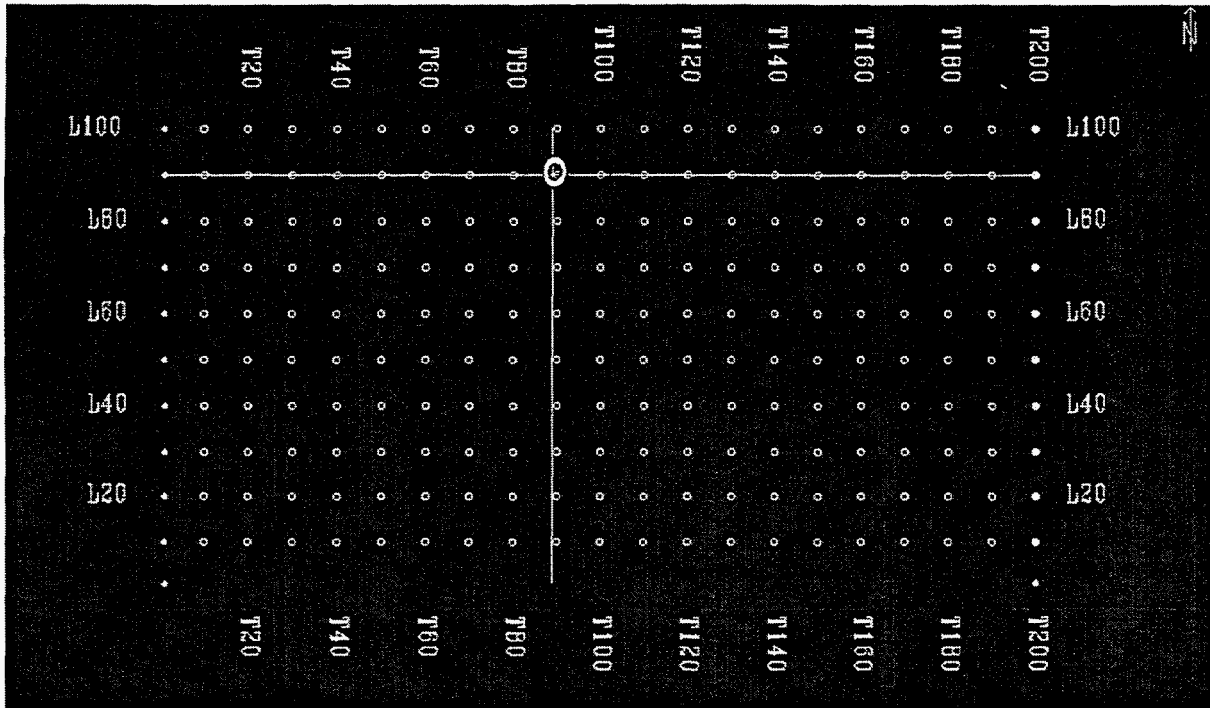


Figure 4.19: A base map of the seismic data area showing the location of the VSP well

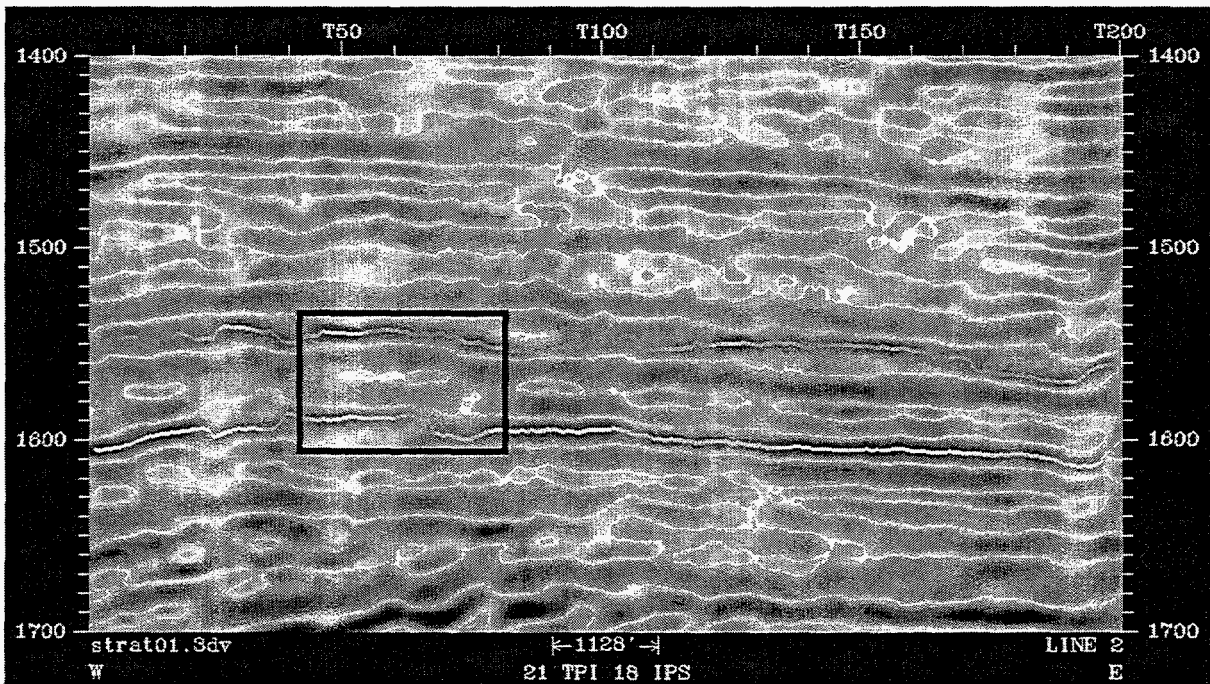


Figure 4.20: Line 1 showing the F11 horizon at approximately 1600 ms and E41 at 1555 ms

Figure 4.22: crossline (trace) #89 showing the VSP well position.

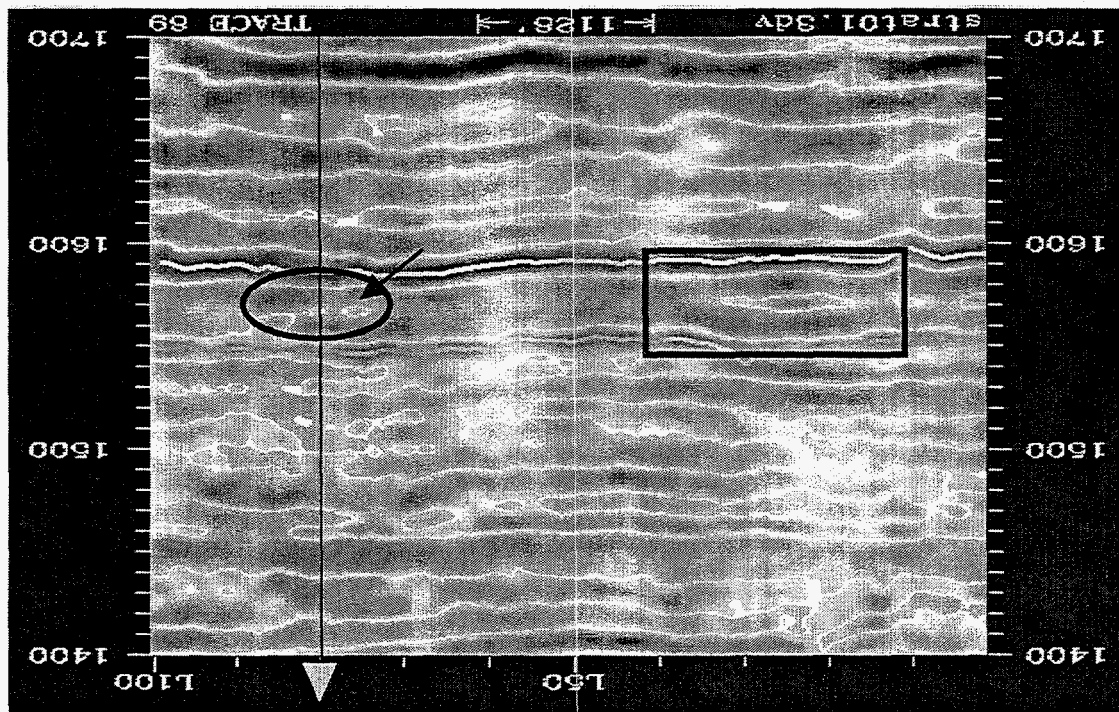
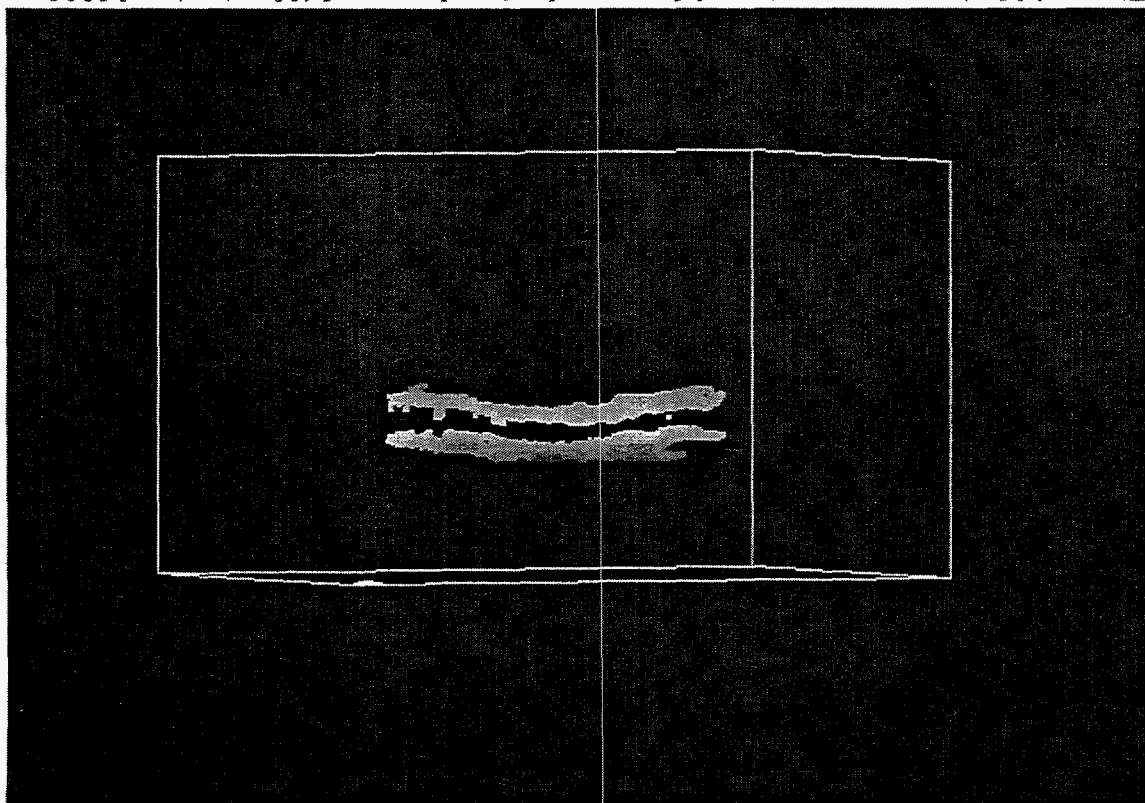
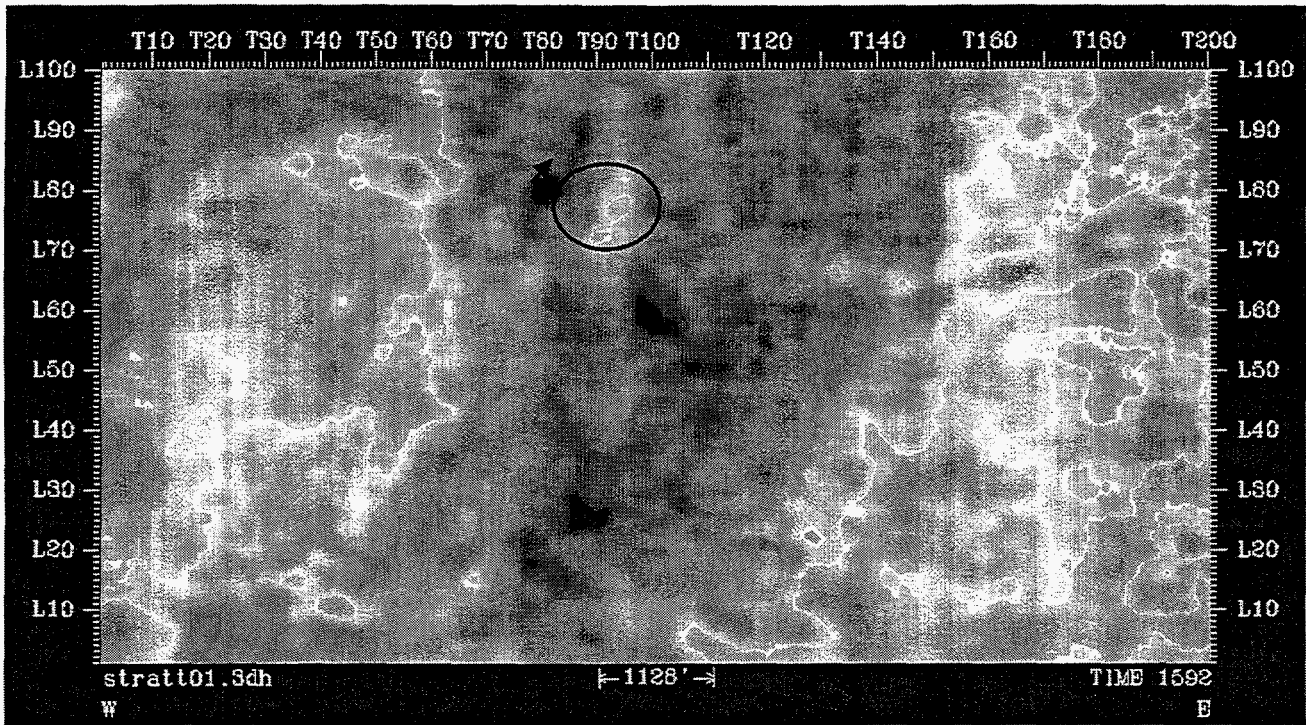
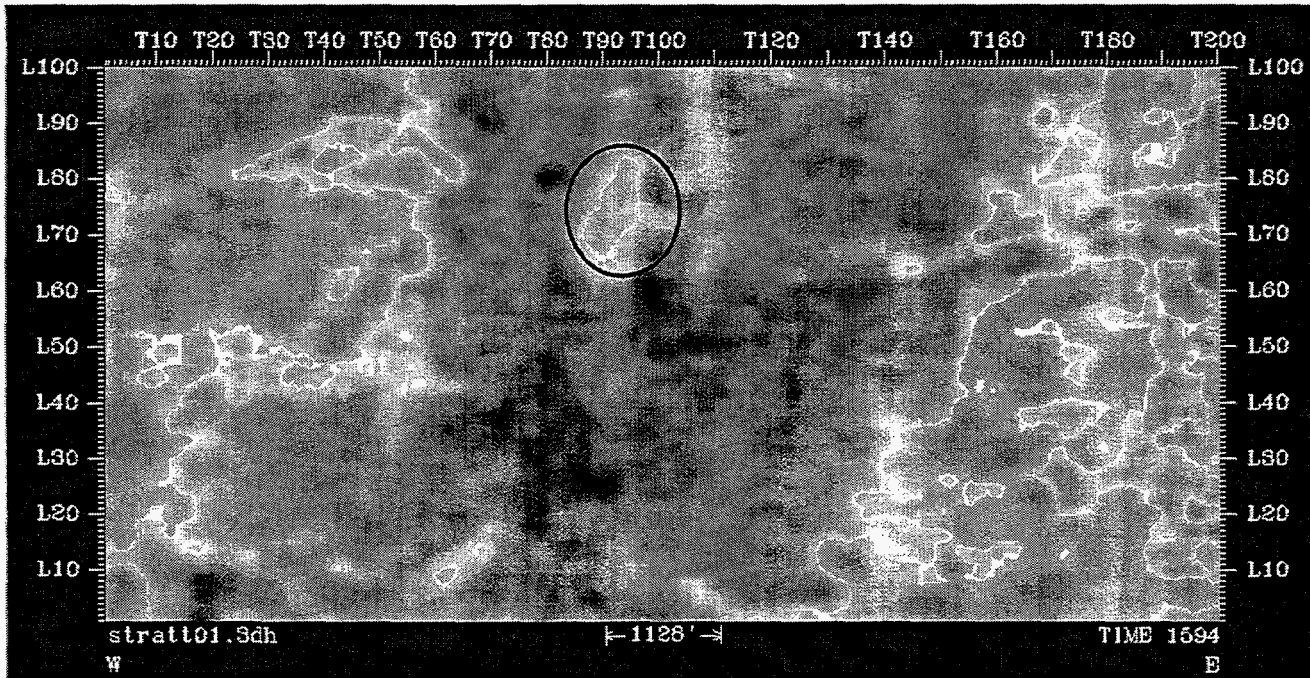


Figure 4.21: A prospective view of the target horizon between 140ms(top) and 190ms.





(a)



(b)

Figure 4.23: (a) Time slice at 1592ms showing the spot indicated by the arrow in Figure 4.23. Below (b) the same spot became clearer and large at 1594 ms. time slice.

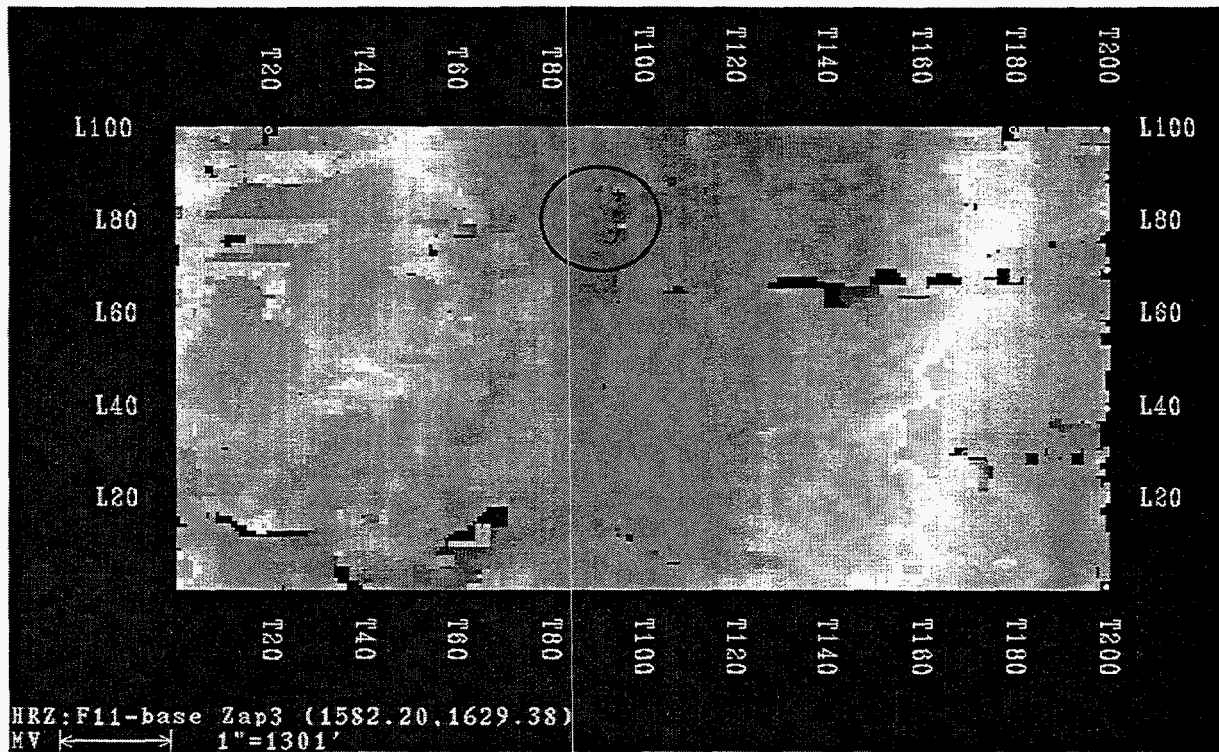


Figure 4.24: F11 reservoir time structure map.

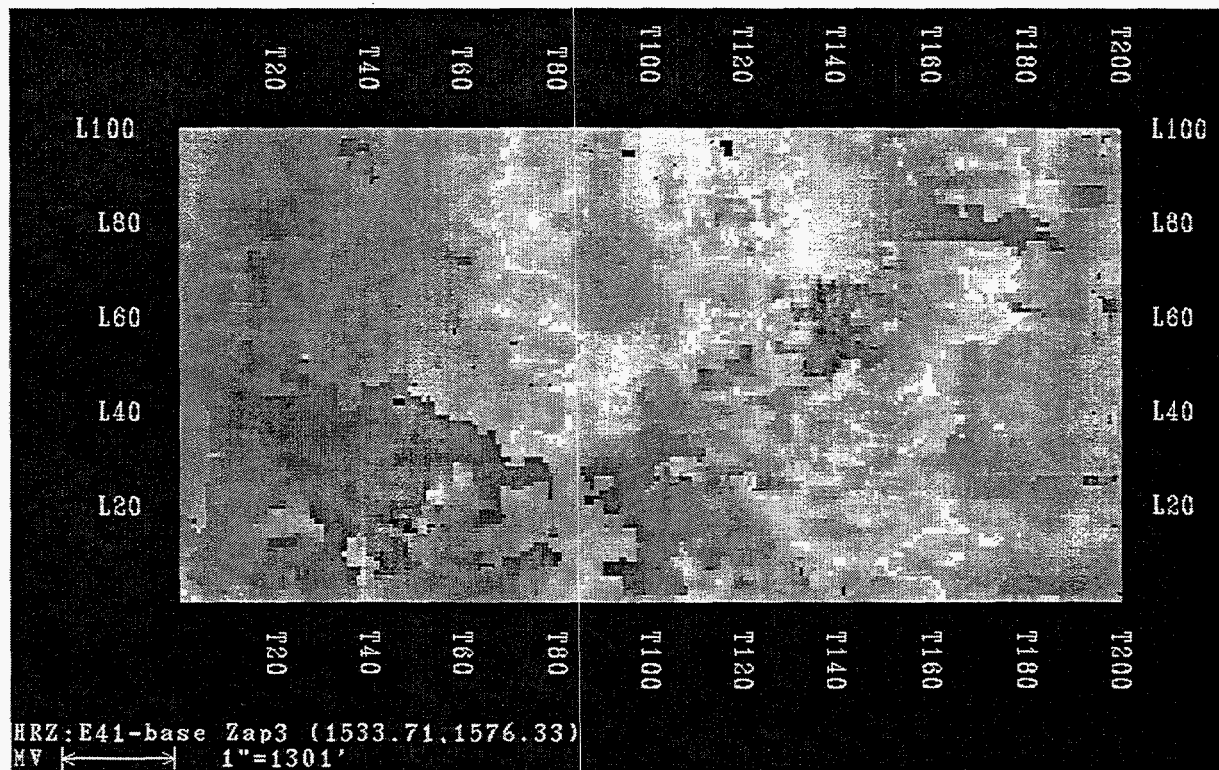


Figure 4.25: E41 reservoir time structure.

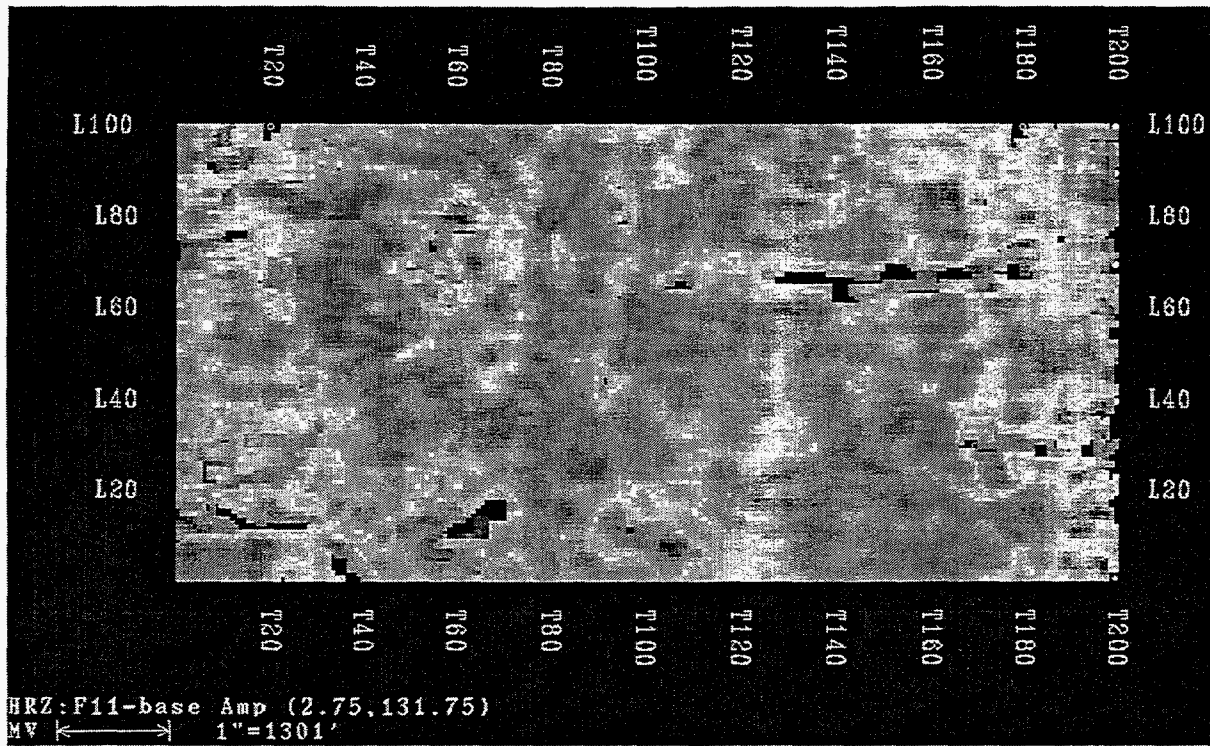


Figure 4.26: amplitude map for F11

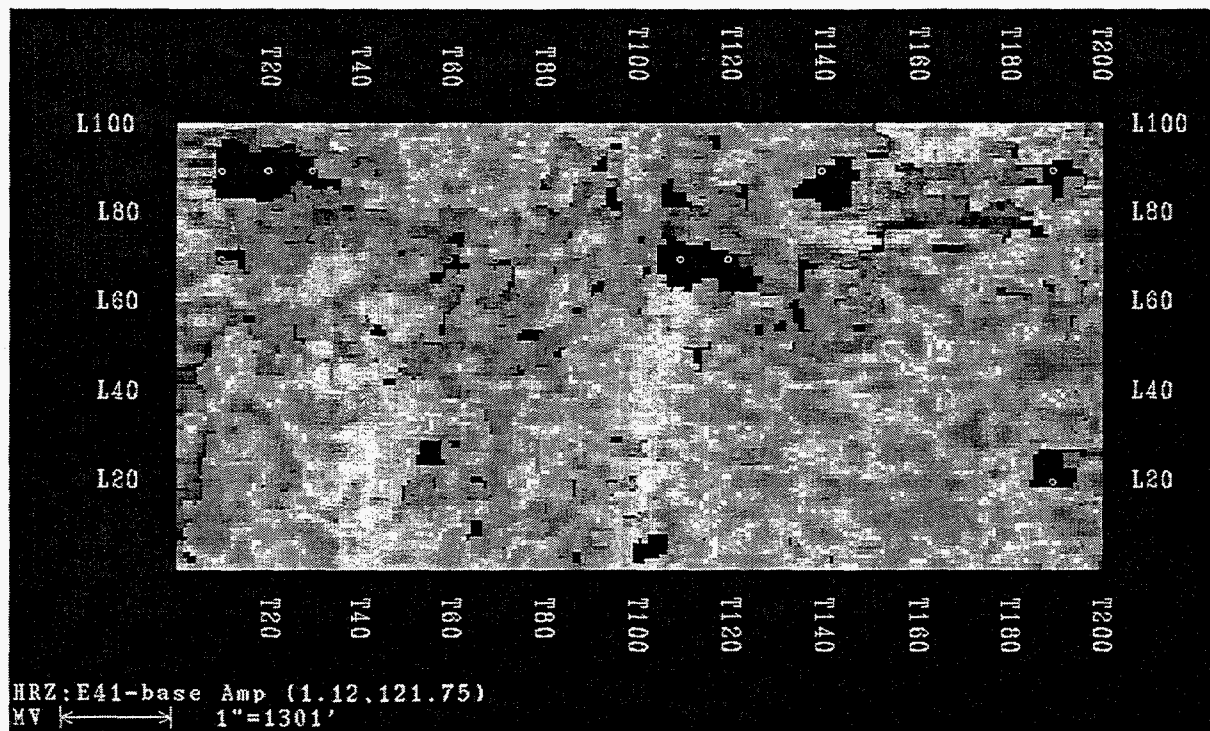


Figure 4.27: amplitude map for E41

Appendix A

Program Marine:

Synthetic seismogram from reflection coefficients.
Program is written in Fortran.77 and reside in a separate disk.

Subroutines:

read-data
twotime
time
amplitude
ricker
fold
zero
convolve

Variable Discription:

Ld = number of layers.
V = velocity.
H= thickness.
T = two way travel time.
D = down going response.
R0 = reflection coefficient

Appendix B

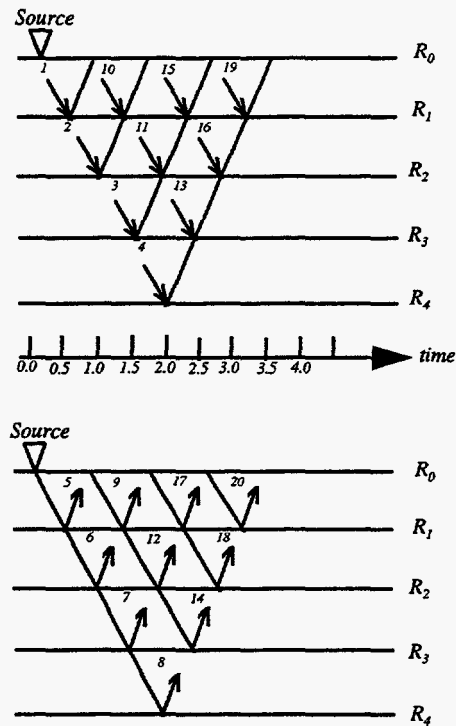


Figure B1: Numbering scheme diagram for downgoing and upgoing waves at equal time.

Pressure amplitude

Calculate the reflection coefficient

$$R = \frac{p_r}{p_i} = \frac{I_2 - I_1}{I_2 + I_1}$$

Assumed the densities are constant, then reflection coefficients for each layer are calculated as:

$$R_1 = \frac{V_2 - V_1}{V_2 + V_1} = \frac{11000 - 10000}{11000 + 10000} = 0.0476$$

$$R_2 = \frac{V_3 - V_2}{V_3 + V_2} = \frac{17000 - 11000}{17000 + 11000} = 0.2143$$

$$R_3 = \frac{V_4 - V_3}{V_4 + V_3} = \frac{10000 - 17000}{10000 + 17000} = -0.2593$$

$$R_4 = \frac{V_5 - V_4}{V_5 + V_4} = \frac{0 - 10000}{0 + 10000} = -1.0000$$

Calculate the downgoing and upgoing waves

$$d_{k+1}(t) = -R_k u_{k+1}(t) + (1 + R_k) d_k'(t)$$

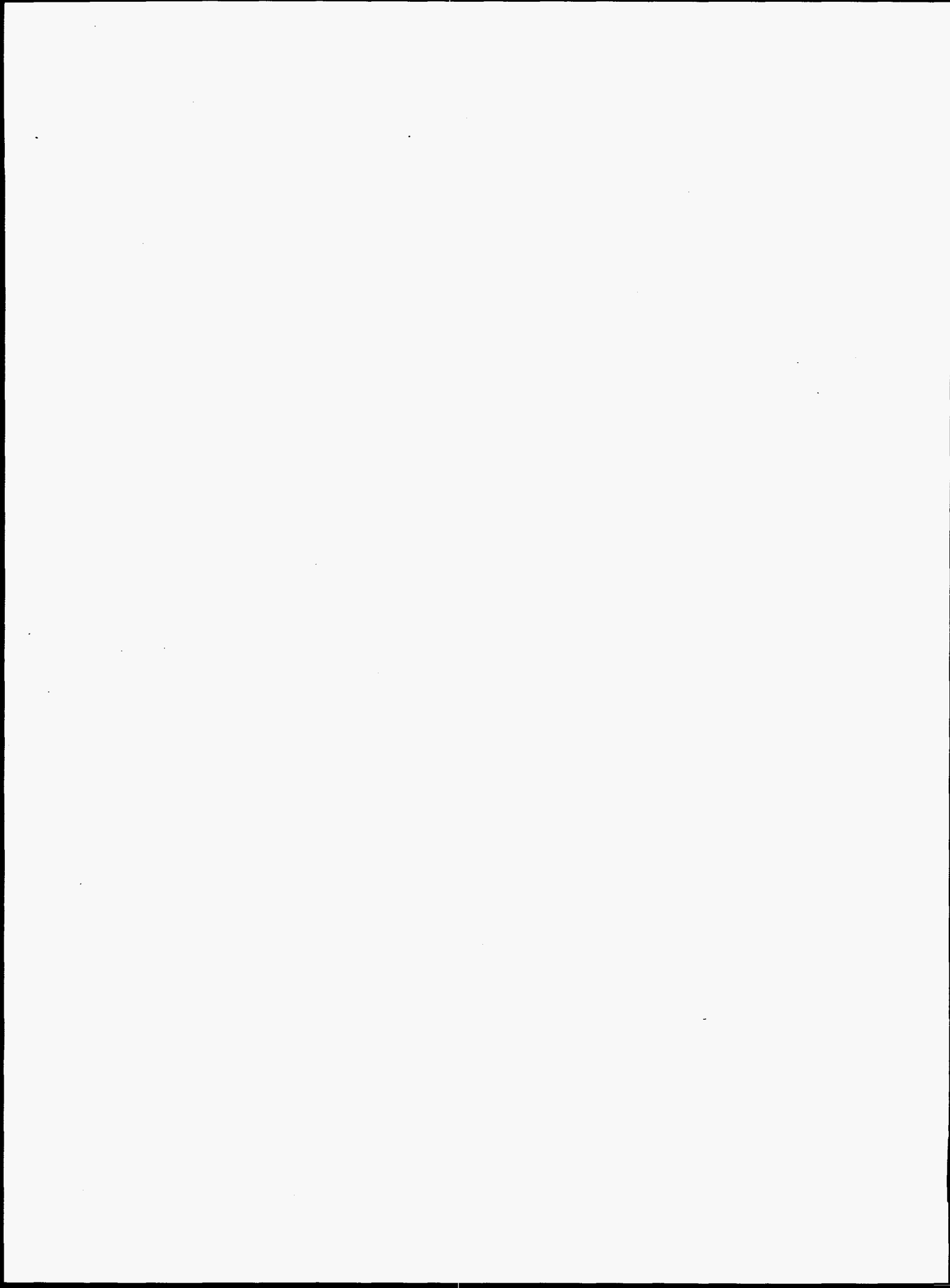
$$u_k'(t) = R_k d_k'(t) + (1 - R_k) u_{k+1}(t)$$

1. $d_1^{0.5} = 1.000$
2. $d_2^1 = 1.000(1.000 + 0.0476) + (0 \times (-0.0476)) = 1.0476$
3. $d_3^{1.5} = 1.0476(1.000 + 0.2143) + (0 \times (-0.2143)) = 1.2721$
4. $d_4^2 = 1.2721(1.000 - 0.2593) + (0 \times 0.2593) = 0.9422$
5. $u_1^{0.5} = 0(1.000 - 0.0476) + (1.000 \times 0.0476) = 0.0476$
6. $u_2^1 = 0(1.000 - 0.2143) + (1.0476 \times 0.2143) = 0.2245$
7. $u_3^{1.5} = 0(1.000 + 0.2593) + (1.2721 \times (-0.2593)) = -0.3299$
8. $u_4^2 = 0(1.000 + 1.000) + (0.9422 \times (-1.000)) = -0.9422$
9. $u_1^{1.5} = 0.2245(1.000 - 0.0476) + (0.0476 \times 0.0476) = 0.2161$
10. $d_1^{1.5} = 0(1.000 + 0.0476) + (0.0476 \times (-1.000)) = -0.0476$
11. $d_2^2 = 0.0476(1.000 + 0.0476) + (0.2245 \times (-0.0476)) = 0.03918$
12. $u_2^2 = -0.3299(1.000 - 0.2143) + (0.03918 \times 0.2143) = -0.2508$
13. $d_3^{2.5} = 0.03918(1.000 + 0.2143) + (-0.3299 \times (-0.2143)) = 0.11827$
14. $u_3^{2.5} = -0.9422(1.000 + 0.2593) + (0.11827 \times (-0.2593)) = -1.2172$
15. $d_1^{2.5} = 0(1.000 + 0.0476) + (0.2161 \times (-1.000)) = -0.2161$
16. $d_2^3 = 0.2161(1.000 + 0.0476) + (-0.2508 \times (-0.0476)) = 0.2383$
17. $u_1^{2.5} = -0.2508(1.000 - 0.0476) + (0.2161 \times 0.0476) = -0.2286$

$$18. u_2^3 = -1.2172 (1.000 - 0.2143) + (0.2383 \times 0.2143) = -0.9053$$

$$19. d_1^{3.5} = 0 (1.000 + 0.0476) + (-0.2286 \times -(-1.000)) = -0.2286$$

$$20. u_1^{3.5} = -0.9053 (1.000 - 0.0476) + (-0.2286 \times 0.0476) = -0.8731$$



Appendix C.

This velocity-analysis software package consists of 3 main parts:

- Model setup: to make cell velocity distributions and geometries (reflectors, shot and receivers).
- Ray tracing: to calculate traveltimes with a velocity model given by user.
- Velocity estimation: to estimate velocity from given traveltimes.

The programs are in Fortran 77 and reside in a separate disk.

Model setup

- 1) vmod.for --- to make a cell velocity model for a true model or a starting model.
Option: horizontal and/or vertical velocity gradient or constant velocity.
- 2) ref.for --- to create reflection interface.
Option: line segment with different dipping and/or sine function.
- 3) ele.for --- to make a shot-receiver elevation file (used linear interpolation with known heights for every station numbers, st #). It creates boundary coordinates of each pixel and coefficients of straight line for an elevation function.

input: random elevation data from elevation survey

```
~
      ex) st #  elevation
~
           0      79.8
~
           11     100.2
```

output: output3—coordinates transformed (inline offset, shooting depth employed)

```
      ex) st #, shx, shy, recx, recy
      output 4-- boundary x, y and a, b for y=ax+b
      ex) boundx, boundy, a, b
```

*note: start ix(i)=first st # and finish ix(i)=last st # for linear interpolation
(that is, extrapolation not allowed)

- 4) shst.for --- to generate the file for shot and receiver station number.
Option: different spreads geometry (end spread, split spread, and CMP sorted).

Ray tracing

- 5) ray.for --- to trace curved rays for 2-D seismic data using cells based on a shooting method. All algorithms for the code can be found in Langan's paper (1985, Geophysics, 50, 1456-1465). The program can be used for borehole, VSP (Vertical Seismic Profile), RVSP (Reverse Vertical Seismic Profile), refraction (Turing wave),

and reflection. It uses cubic spline reflection interface. Stations for shot and receiver can be located anywhere for reflection geometries.

```
input source x,y and ray definition data nput infoin(2): example

1 0 1          istrflag (0 = curv), ismooth (0=no), ireflect (0=no reflect)

1 10. 45 20 20 iautoang (1 = auto), angmin, angmax, nangle, niter

1             itflag (0=last iter. ray; 1 = 0 & tomo.out; 2 = every itrr. ray)
0             idata (0 = for borehole/vsp ; 1 = for reflection/refraction)

shot and station information depend on idata
filename (shstin)

if,          idata = 0          idata = 1
2 (nshots)   2 100. 1. (nshots, shotx1, shspace)
1 100 0.     24                (nstations = channels)
2 110 0.     1 -6. 0.          (ist, ichx, ichy)
1 (nstations) 2 -----
1 150. 0.    24 8. 0.

line 1****
istrflag: flag for straight or refracting rays
0 = normal mode, use snell's law at each boundary
1 = straight ray mode, ignore snell's law

ismooth: flag for smoothing velocity.
0 = no smoothing
1 = smoothing
ireflect: # of reflecting interface
0 = no reflecting interface (ex: borehole)
1 - ? = how many interfaces

line 2****
iautoang: angle given automatically
0 = not auto
1 = auto (ignore angmin,angmax)
angmin: minimum takeoff ang for 1st iteration (deg)
angmax: maximum takeoff ang for 1st iteration (deg)
nangs : number of ray angs tried for each iteration
niter : number of iteration steps

line 3****
```

```
itflag: flag for operating mode
  0 = write out only rays from last iteration
      don't create the tomo.out file for the tomography program
  1 = write out only rays from last iteration, but
      create the tomo.out file
  2 = write out all rays to check # of iterations
      don't create the tomo.out file
```

```
~
line 4****
~
```

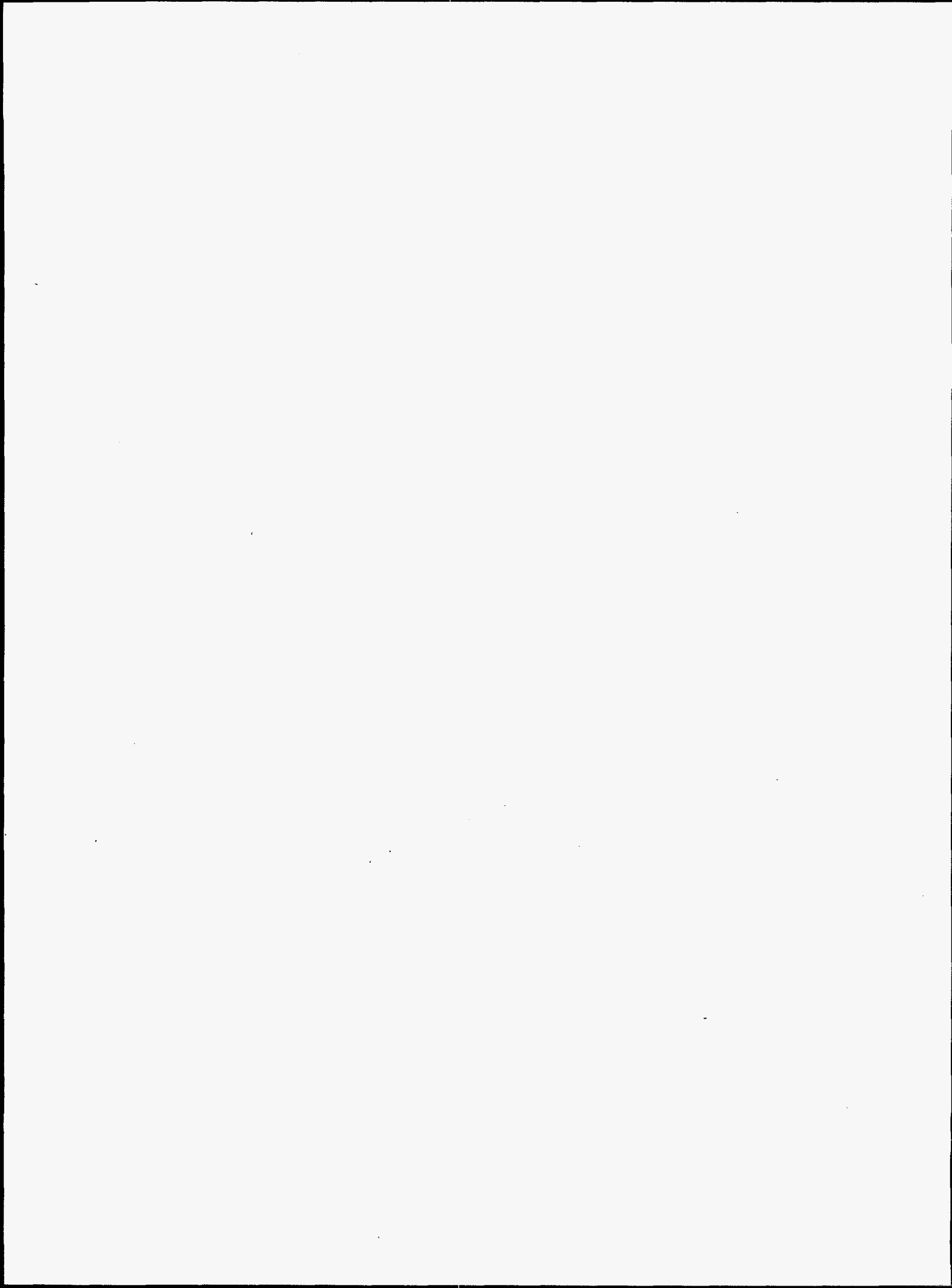
```
data: flag for data type
~
```

```
  0 = for borehole/vsp
~
```

```
  1 = for reflection/refraction
~
~
```

Velocity estimation

- 6) lsqv.for --- to solve $X = G^{-1}b$ where G^{-1} denotes the generalized inverse matrix. This routine takes the equation $Ax = b$ and solves the associated least-squares problem $A'Ax = A'b$ where prime denotes transpose. In the code, matrix $A(n,m)$ n is # of columns, m is # of rows, and $b(m)$ and $x(n)$ have one column. g and bb are work spaces for the intermediate arrays and vectors but g ends up holding the generalized inverse matrix G^{-1} at the end.



PART V

MARKOV RANDOM FIELD AND BOOLEAN MODELS FOR GEOLOGIC MODELING OF FACIES DISTRIBUTIONS

By: Z. X. Bi, R. A. Redner, D. S. Oliver, Y. Abacioglu and A. C. Reynolds

Summary

The objective of this work has been to generate realizations of facies distributions constrained to observations (well or outcrop) and geological interpretation with specific emphasis on fluvial systems. The work has emphasized the development of two very different types of statistical models, Markov Random Field (MRF) models and Boolean models.

MRF models allow us to define a joint probability function for the distribution of facies by specifying local characteristics, or more mathematically, conditional probabilities based on local neighborhood systems. This is an extremely convenient feature as it is usually intuitively obvious how to specify the conditional probabilities to incorporate desired geologic features, continuity, anisotropy (or more generally orientation), and ordering and to some extent overall structure. Once the statistical parameters in the joint probability function have been determined realizations of the facies distribution are generated by a Markov chain Monte Carlo (MCMC) sampling procedure. A penalty term can be incorporated in the probability function to ensure that the fraction of facies in each realization is within some specified range. Conditioning to facies observed at wells is a trivial task. The main model we developed is extremely simple in that it is based on a second order neighborhood system and controls geologic features using only two-pixel cliques. Because of this, we are able to simply use a free boundary condition and avoid the unsolved problem of determining an appropriate definition of potential functions (statistical parameters) on cliques adjacent to the actual three-dimensional reservoir. This simplicity also makes it much easier to define conditional probabilities to simulate desired geologic features. Despite this simplicity, this MRF model can be applied to generate a rich variety of facies configurations. However, use of the model and code developed requires some understanding and intuitive insight on the way conditional probabilities affect geologic features, but much of this insight can be developed simply by reading this report. It would of course be preferable to develop an automatic procedure to generate the statistical parameters which define the probability function for the MRF model directly from geologic interpretation. For this purpose, we have investigated a number of algorithms, e.g., maximum likelihood estimation, coding and histogramming. Many of these methods have been successfully applied in the image analysis literature. Thus, we tried to apply various modifications of these procedures to our problem. Unfortunately, none of them proved generally reliable. The maximum likelihood estimation procedure is reliable only when a good initial guess is available for the statistical parameters, and even then, it is computationally expensive.

Although we do believe that the MRF model developed will prove to be a useful tool for the generation of many facies distributions of interest to geologists, it is not easily applied to generate facies as geometrical objects having a desired structure, e.g., channels and splays in a fluvial system. Thus, we have investigated the application of Boolean methods. With Boolean methods, or more generally, Marked Point Process techniques, distinct objects are inserted into the reservoir. With this approach, well-organized structures are easily obtained. By working closely with geologists, detailed information on the statistics of the parameters in the model can be obtained. For example in our study of the Middle Frio Stratton Field, we have accumulated data on channel thickness, channel width to thickness ratios, the angles that the channel makes with respect to some reference point and the natural clustering of channels. With these and other pieces of data we can make relatively realistic simulations of channel belts. Even more realistic simulations are possible if additional data are incorporated into the model.

The inclusion of specific well and outcrop information can be used to generate statistics to help understand the nature of the field and can also be used to constrain the stochastic realizations that are generated. At this time we have only used the data from the study areas to generate statistics which are used to generate simulations. We have not, at this time, used this type of information to constrain the simulations.

Introduction

The distribution of geological facies in petroleum reservoirs is a key element for reservoir characterization and flow simulation because it controls the heterogeneity of physical properties of the reservoir over multiple scales. In practice, some general information about this distribution is usually available from outcrop studies or geological interpretation. Although various stochastic models have been proposed to generate facies distributions, this section of the report focuses on Markov Random Field (MRF) models and Boolean models for generating facies distributions.

MRFs provide a powerful tool for modeling spatial interactions and have been used extensively to define prior distributions in Bayesian image analysis and statistical physics. It has also been suggested that one could apply a MRF to generate geological facies distributions, but many of the special issues related to this problem have not been previously explored. The MRF models developed in this work can be applied to generate realizations of facies distributions that display desirable characteristics determined from geologic interpretation, for example, continuity, orientation and ordering. Boolean methods are more applicable for generating facies as large, geometric, geologic objects having a specified shape, e.g., channels and splays in a fluvial environment. Norwegian researchers have been responsible for developing many of the stochastic algorithms for generating facies distributions. They have proved especially adept at developing and applying stochastic simulation techniques based on Boolean models, or more generally marked-point processes.

In Boolean methods, facies are distributed in space as geometrical objects according to probability laws developed from well data, geological data and interpretation. The process of distributing objects in space by applying a probability law, which governs the distribution of the centers of the objects, is normally referred to a "point process" since the distribution of objects is determined explicitly by the distribution of the points describing the centers. In a marked point process, the point process defining the center of objects is "marked" (associated) with other probability distributions which may be used to describe properties of the object, e.g., size, shape and type.

Generally, there are three basic problems involved in the application of any probabilistic model: (a) application of geological knowledge to propose an underlying stochastic model; (b) use of information, observations and geologic interpretation to estimate the parameters needed to define the probability function; (c) simulation or sampling of the probability function to generate realizations. This last part includes conditioning realizations to actual observations. In this research, we investigate these problems systematically. Unfortunately, the parameter estimation problem is one that we have been able to solve only in special cases.

Literature Review

When Markov random fields are used in stochastic modeling, one must first divide the region (surface or volume) into an array of "grid-blocks" or pixels, which are labeled 1 to n . Following the notation of Ref. [54], we let $S = \{1, 2, \dots, n\}$ and let $X = (X_1, X_2, \dots, X_n)$ be a random vector with X_j representing a random variable defined on pixel j . For example, as in Ref. [54], X_j could represent the facies located at grid-block j if we wish to generate realizations of the facies distributions on the grid-blocks. Alternately, X_j could represent shale thickness in a grid-block, or the shale thickness at a pixel located on a correlation surface [31, 32]. Ref. [54] gives a formal definition of what it means for X to be a Markov random field. The main characteristic of a Markov random field is that the conditional probability of X_j given values of all X_i for $i \neq j$ depends only on the values of those X_i located in some neighborhood of X_j , i.e., in a "local neighborhood". This property results in significant computational speedup when one samples X by starting with an initial distribution, proposes a new distribution by some algorithm and accepts the new distribution with a specified probability (e.g. Gibbs sampler or Metropolis algorithm). As in a Markov chain Monte Carlo procedure, iteration is continued until the distribution of X converges. This type of procedure was used very successfully in the classical paper of Geman and Geman [21] on image reconstruction. However, when Tjelmeland and Holden [54] used the procedure to generate facies distributions with little or no well conditioning data, they found that the realization obtained either looks like independent noise or is completely dominated by one facies. In certain simple cases, justification of why a realization may be dominated by one facies is provided by the theoretical results of Kinderman and Snell [35]. Tjelmeland and Holden recommended using a semi-Markov random field to overcome the apparent inability of Markov random fields to capture large-scale features. In this procedure, one can ensure that realizations contain at least a prescribed number of "bodies" where the same facies is assigned at each gridblock within a particular body. Although we have experimented extensively with this

type of approach, it appears to be highly computationally inefficient due to the extensive computation required to keep track of the number of bodies.

The trend setting work of Haldorsen and Lake [29] represents one of the earliest attempts to use a stochastic algorithm to generate facies distributions. They presented a procedure to distribute shales within a vertical cross section of a reservoir. Their work may be viewed as distributing shale units within a background facies (sandstone). They considered both "deterministic" and stochastic shales, where deterministic shales are continuous between observation points. Stochastic shales are randomly located within the sand body. In a two dimensional cross-section, shales are represented by rectangles. (In three dimensions, shales are assumed to be rectangular parallelepipeds.) They assume that the thickness and length of rectangular shale bodies are independent and that the locations of shale bodies are independent so that shales may overlap. The probability distribution of shale thicknesses is formulated based on observations at wells. The probability distribution for the length is built from outcrop studies or by general correlation on lateral continuity based on depositional environment, see Fig. 5 of Ref. [29] which was reproduced from Ref. [57].

The papers [31,32] by Hoiberg, Omre and Tjelmeland are more complex, but ones which make more thorough use of geological information. In particular, the geological setting for these papers is the North Sea, and the model is a permeable sandstone background with good flow properties with nearly impermeable shale barriers lying within this sandstone matrix. These shale barriers are assumed to be of two types. The first shale type is that of large lateral shale bodies that may intersect multiple wells. The second shale type is smaller shale-units which are not correlated between wells. The procedure assumes that shale within core data can be delineated from sandstone and also assumes that shale corresponding to large lateral shale barriers and shale of small units can be distinguished from the analysis of core samples. We note, as do the authors of this paper, that the strength of this approach is that all realizations reproduce specified data about the reservoir. In particular, the occurrence and thickness of shales observed in wells will be reproduced and the probabilities that wells are connected by shale sheets are satisfied. Furthermore, the percentage of shale versus sand will be as specified and the thickness distribution of shale thicknesses in each simulation will match the shale thickness distribution which was specified.

The geological model used by Syversveen and Omre [49,50] is much simpler than the model of [31]. In this paper, the authors consider a reservoir which has a sandstone background. But within the background matrix, only a single type of shale barrier can exist and the paper is written for only two dimensions (although it could easily be extended to a third dimension). Within the sandstone background, they consider the problem of introducing impermeable shale barriers using a marked point process.

In contrast to [31] and [49], the geological setting for the paper by Egeland, Georgsen and Knarud [19] is a fluvial reservoir and hence the background is assumed to be a nearly impermeable floodplain matrix that contains permeable sand-filled channels. Sheetsplay sand and barriers are added to the model to produce a more realistic and complex

geology. This paper provides only a broad overview of the stochastic models but underlying theoretical details can be found in Georgsen and Omre[22]. In Ref. [19], sand channels are modeled by marked point process. Each channel is initially identified with a straight line that provides the main direction for the channel. A joint probability distribution for these lines is used to provide correlation between channels. The second part of the model gives additional information on the size and shape of these individual channels. A finite number of points are selected along each line, and at each of these points, random values for the location (as a perturbation from the original line), thickness and width of the channels is generated. Gaussian random functions are used to generate these random values. Additionally, sheets are deposited along the edges of the channel. Finally elliptic-shape barriers are placed randomly along the line defining the channel. These are perturbed using the location information that was used to perturb the channel from the initial reference line.

Papers of interest that deal with the general problem of facies distribution include Ref. [53, 55] presented by Tjelmeland. This approach is based on defining probabilities for special configurations that exist within cliques arising from higher order neighborhoods. The procedure is sufficiently general to incorporate a known ordering of facies. Our implementation of this basic idea, as well as the study of the related issues, is the main subject of this section of the report. The basic method used in Ref. [55] is actually an extension of an image segmentation model, which was introduced and called a Multi-Level Logistic (MLL) model by Derin and Elliott [14]. In the MLL model, clique potentials can take on only two values, one when all the pixels in the clique are identical (for our problem, this means all occupied by the same facies) and one when they are not. Tjelmeland noted that some realizations that have been generated by this model are not truly representative of the underlying MRF model, but instead represent results generated during the transient part of a Markov chain. Our experiments confirm this conclusion. Thus, Tjelmeland proposed a MRF with higher order interactions, i.e., larger neighborhood systems, with clique potentials dependent on the specific configuration in the clique.

Regarding the problem of parameter estimation, Beseg [2] proposed a procedure called the coding method. This procedure may be viewed as a maximum likelihood estimation method that yields parameter estimates that maximize the product of the conditional probabilities of a subset of the random variables in a field, conditioned to the rest of the field. The codings depend on the specified neighborhood. The advantage of this method is that the maximization is computationally efficient because the normalizing constant can be calculated exactly. However there exist different possible codings for a specified neighborhood system and the estimates of parameters obtained depend on the specific coding used. It has also been shown [53] that the estimates obtained by coding may be highly inaccurate for some probabilistic models.

In view of the difficulties involved in the coding method, Derin and Elliott [14] introduced a histogram approach. This method is also relatively easy to implement and is efficient, but unlike the maximum likelihood estimate, it does not have a clear probabilistic interpretation. For some of the problems, the method is not applicable

because the number of equations that can be obtained from the observation is less than the number of parameters to be estimated.

Because of the unknown normalizing constant in the MRF model, some statisticians have presented procedures to estimate this normalizing constant. Geyer and Thompson [23, 24, 25] proposed a procedure called Markov chain Monte Carlo Maximum Likelihood Estimation in which the normalizing constant is estimated by samples from a Markov chain Monte Carlo run based on an initial guess of the parameters. It turns out that when the initial guess is close enough to the truth, this method works very well, but for problems where the MRF probability function involves a large number of statistical parameters, the computational cost is very high.

The preceding papers were reviewed in detail because they contain basic ideas that motivated our interest in the general problem of generating facies distributions. Important background on statistical theory related to these subjects is given in Cressie [9], Ripley [45], Stoyan et al. [47], Besag [2] and Besag and Green [3]. References [9], [45] and [47] are especially important as they provide information on the Hammersley-Clifford Theorem which gives conditions such that conditional probability distributions used in Markov random field models yield a consistent statistical model. We discuss the generation of facies as three-dimensional geometical objects within a background facies using Boolean models (or more generally, marked point processes) in another part of the report. We note that marked point processes have been used in other related applications, e.g., to model the distribution of faults [38]. We also note that various forms of marked point processes have actually been applied to model geological architecture field cases, see, for example, Refs. [12] and [56]. Both of these papers use a two step procedure. In the first stage, a marked-point process is used to distribute facies and in the second stage, a continuous Gaussian model is used to distribute petrophysical properties within each facies. Finally, we note that there exist numerous papers on non-Boolean types procedures for generating facies distributions, e.g., truncated Gaussian fields and indicator methods, which are not discussed here. Good overviews of stochastic models and the reasons for applying stochastic models are provided in Refs. [13], [28] and [39]; Ref. [58] presents a geologist's viewpoint.

Markov Random Fields

Basic Notation

When Markov random fields are applied for modeling geological facies distributions, one must first divide the region of interest (2D surface or cross-section or 3D volume) into an array of regular gridblocks, pixels or sites. We use S as the set of subscripts or indices of pixels or sites. For example, for a two-dimensional $M \times N$ lattice, $S = \{(i, j), 1 \leq i \leq M, 1 \leq j \leq N\}$, and for a three-dimensional system with $N_x \times N_y \times N_z$ gridblocks, $S = \{(i, j, k), 1 \leq i \leq N_x, 1 \leq j \leq N_y, 1 \leq k \leq N_z\}$. Here, each pair (i, j) or triple (i, j, k) is called a pixel or a site or a grid-block. In theoretical developments, the notation is significantly simpler if we use a single index for the random variables, i.e., $S = \{i, 1 \leq i \leq n\}$. Here, n is the

number of pixels in the region of interest, e.g., $n = M \times N$ or $n = N_x \times N_y \times N_z$. Although S could be infinite, our discussion will be restricted to the cases where S is finite.

Throughout, capital letters such as X and Y , are used for random variables or random fields and the corresponding lower case letters are used for realizations of these random variables. Capital letters used without any subscripts are used to denote the random fields from which the random variables are taken. For example, $X = \{X_{ij}, (i, j) \in S\}$ represents a two-dimensional random field; it is a collection of random variables, one defined at each pixel (i, j) . We use X_C to represent a subset of the entire random field X where C is a subset of S . The notation $P(x)$ refers to a probability mass function. Our objective is to generate facies distribution on the system of pixels and consequently the random variable can take only discrete values representing facies so $P(x)$ is a probability mass function, not a probability density function. Note we use $P(x)$, instead of $P(X = x)$, to denote the probability that the random field X takes on a value x .

In our application, the random variable defined on pixel i ($i \in S$), X_i , represents the facies value located at pixel i since we wish to generate realizations of the facies distribution on the system of pixels. For convenience, we use alphabetic letters, A, B, C, \dots to represent different facies, but we use numbers, $1, 2, 3, \dots$ as the facies values in the corresponding computer program. The set of facies that each X_i can take on is denoted by F , i.e., $F = \{A, B, C, \dots\}$ or $F = \{1, 2, \dots, K\}$, where K is the number of facies or values to be considered. A few basic definitions are in order.

Neighborhood System and Cliques

Definition 1: A collection of sets of subscripts $\partial = \{\partial_i \mid i \in S\}$ of S is called a neighborhood system if

- (a) for each $i \in S$, $\partial_i \subset S$
- (b) for each $i \in S$, $i \notin \partial_i$
- (c) for each $i \in S$ and $j \in S$, $j \in \partial_i$ implies $i \in \partial_j$.

This definition indicates that neighborhoods must lie within S , a pixel i is not its own neighbor and if i is a neighbor of j , then j is a neighbor of i . Consider for example the traditional first order neighborhood system in two dimensions. In this case, the neighbors of a pixel i are the pixels immediately above, below, to the left and right of pixel i ; see Fig. 5.1.1a. Of course, this must be suitably modified in the obvious way when i is on the edge of the pixel system, depending on what type of boundary treatment is applied. For example, if pixel i is on the right edge of a two-dimensional lattice and a free boundary is adopted, then the neighbors of i are only the pixels above, below and to the left of pixel i ; see Fig. 5.1.1b. Thus, pixel i has fewer neighbors than the interior pixels. (From the experiments that we have done, it appears that this free boundary procedure is the best way to avoid visual boundary effects.) Similar adjustments should also be made around the entire border.

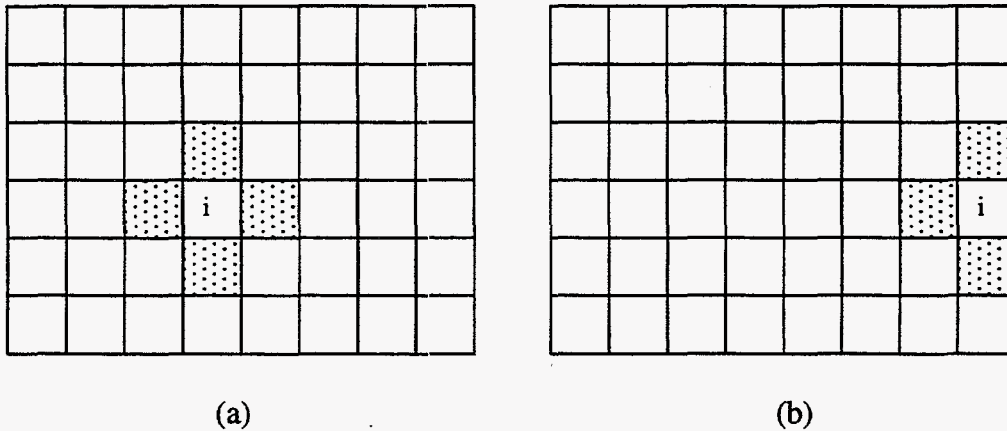


Fig. 5.1.1(a) first order neighborhood of pixel i consists of the dotted 4 pixels;
 (b) pixel i , has only three pixels as neighbors.

5	4	3	4	5
4	2	1	2	4
3	1	i	1	3
4	2	1	2	4
5	4	3	4	5

Fig. 5.1.2 – A hierarchical sequence of neighborhoods of pixel i in a two-dimensional lattice.

Actually, there is a hierarchical sequence of neighborhood systems in a two-dimensional lattice; see Fig. 5.1.2. The second-order neighborhood consists of the first-order neighborhood plus the 4 nearest neighbors in the diagonal directions and the j^{th} order ($j=1,2,\dots,5$) neighborhood contains all pixels that are labeled with a number less than or equal to j . Of course, neighborhood systems are not limited to the ones in the above sequence, any set of pixels satisfying the definition qualifies as a neighborhood system. Obviously the highest order neighborhood consists of the entire lattice. Notice also that ∂_i 's does not have to be the same for each pixel i . However, in our applications, ∂_i is chosen to be the same for every i in the lattice (homogeneity) except near the edges where we use a free boundary.

Given a neighborhood system, a clique is defined as follows.

Definition 2: A clique c is a subset of S such that

- (a) c contains only a single pixel, or in the case that c contains two or more pixels then
- (b) if $i \in c$ and $j \in c$ then i and j are neighbors of each other.

By this definition, individual pixels are cliques. The set of all cliques in a neighborhood system is denoted by C . sometimes we use C_1 to denote the set of single pixel cliques, C_2 the set of two pixel cliques and so on. Clique types for the first-, second- and third-order neighborhood systems are depicted in Fig. 5.1.3.

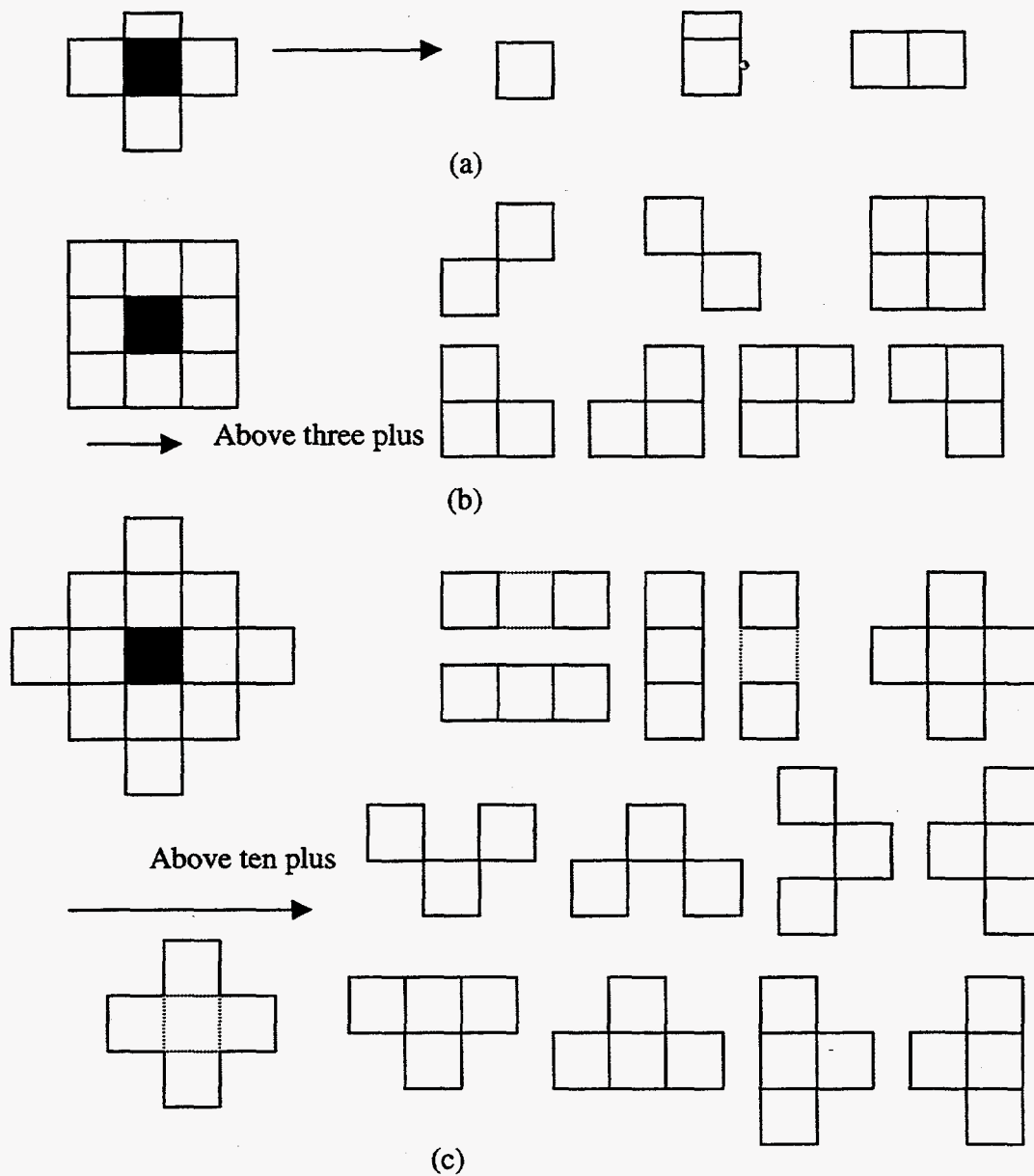


Fig. 5.1.3 – Clique types for (a) first-, (b) second-, (c) third-order

neighborhood systems in two dimensions.

Markov Random Fields

Definition 3: A random field X defined on S with probability function $P(x)$, is said to be a Markov random field with respect to a neighborhood system ∂ if the following two conditions are satisfied

$$(a) \text{ (Positivity) } P(X = x) > 0 \text{ for all } x \in \Omega \quad (5.1.1)$$

$$(b) \text{ (Markovianity) } P(x_i | x_j, j \neq i) = P(x_i | x_j, j \in \partial_i) \quad i, j \in S \quad (5.1.2)$$

where Ω is called the sample space or state space. In our application, the size of Ω depends on the number of pixels as well as the number of values that each random variable can assume, i.e., the number of elements in the set F . For example, if F contains K elements and the lattice contains n pixels, then $|\Omega| = K^n$. The positivity condition requires that every x in Ω has a nonzero probability of occurrence. The second property, called Markovianity, states that the conditional probability of X_i taking on a value x_i , given values of all other X_j , depends only on the values of those X_j located in the neighborhood of i . This condition provides a way to compute conditional probabilities easily and efficiently if the neighborhood is not very large. But, defining a MRF by specifying a local property or conditional probability can make it difficult to determine a joint probability function which is consistent with the conditional probabilities unless care is taken. Ensuring a consistent probabilistic model is the objective of the Hammersly-Clifford Theorem.

Gibbs Random Fields (GRF)

Definition 4: A random field X defined on S has a Gibbs distribution, or equivalently is a Gibbs random field with respect to a neighborhood system ∂ on S , if and only if its joint probability mass function $P(x)$ is of the form

$$P(X = x) = \frac{1}{Z} \exp\{-U(x)/T\}, \quad (5.1.3)$$

where

$U(x) = \sum_{c \in C} V_c(x_c)$ is referred to as energy function and $V_c(x_c)$ is called the potential

function associated with clique c ; the sum is over the set of all cliques C .

$Z = \sum_{x \in \Omega} \exp\{-U(x)/T\}$ is simply a normalizing constant, also known as partition function,

which ensures that the sum of the probabilities over all possible realizations x is unity. This is necessary in order for $P(x)$ to be a probability function. T is a constant (often called temperature) which will be assumed to be unity unless otherwise stated

To calculate the Gibbs distribution, it is necessary to evaluate the partition function. Since the size of the state space Ω of X is K^n , even for a problems where K is small (two for the case where there are only two possible facies) and the number of elements in S is only a few hundred, the evaluation of the partition function is computationally intensive. This problem adds to the complexity of deriving procedures for sampling and/or estimating the statistical parameters defining the probability function.

Note also that $V_c(x_c)$ is dependent only on the pixel values within a clique c , i.e., x_c . For our application, a clique potential $V_c(x_c)$ can be specified by parameters which quantify the influence of different configurations of x_c on the characteristics of the entire random field. This aspect will be discussed later in detail.

A GRF is said to be homogenous if $V_c(x_c)$ is independent of the relative position (location) of clique c in S . It is said to be isotropic if $V_c(x_c)$ is independent of the orientation of c . In this work, homogeneity is always assumed unless we explicitly state otherwise. Anisotropy will be applied to describe the direction-dependent characteristics of the facies distribution.

The temperature T controls the sharpness of the distribution. When the temperature is very high, all of the possible facies distributions have essentially the same probability and consequently there is no pattern or structure in a realization generated, i.e., the realizations looks much like random noise. If the temperature is near zero, realizations the will concentrate around the global minima of $U(x)$. This phenomenon will be discussed later.

The origins of Gibbs distributions lie in the physics and statistical mechanics literature. Much of the research results pertain to the Ising model (see Refs. [16, 35]) which is a special case of the Gibbs distribution. Unfortunately, the analysis of even this very simple Gibbs distribution is difficult. In the application of modeling facies distributions, the analysis is basically prohibitive because of the huge state space or the intractable normalizing constant. The advantage of the Gibbs distribution is that it provides an effective probability model by specifying a joint (or global) probability mass function. The celebrated Hammersley-Clifford theorem indicates that a unique GRF exists for every MRF and conversely as long as the Gibbs random field is defined based on the cliques in a neighborhood system.

The Hammersley-Clifford Theorem

Theorem 1. (The Hammersley-Clifford Theorem) Assume that $P(x_1, x_2, \dots, x_n)$ is a probability function which satisfies the positivity condition and let $\{\partial_i \mid i \in S\}$ be a neighborhood system. If for each i , markovianity of Eq. 5.1.2 holds, then

$$\begin{aligned}
P(x_1, x_2, \dots, x_n) = & \frac{1}{Z} \exp\{-[\sum_i x_i G_i(x_i) + \sum_{i < j} x_i x_j G_{ij}(x_i, x_j) \\
& + \sum_{i < j < k} x_i x_j x_k G_{ijk}(x_i, x_j, x_k) + \dots + x_1 \dots x_n G_{1, \dots, n}(x_1, x_2, \dots, x_n)]\} \quad (5.1.4)
\end{aligned}$$

where each of the G functions is zero if the indices for that G function do not form a clique. Subject to this restriction, the G functions may be chosen arbitrarily.

The last statement of the theorem should of course be interpreted as saying that the G functions can be chosen arbitrarily subject to the restriction that they are zero if the indices do not form a clique and then Z is chosen as a normalizing constant so that the total probability is one.

Now clearly the formula for P can be written more compactly as the Gibbs distribution

$$P(x_1, x_2, \dots, x_n) = \frac{1}{Z} \exp(- \sum_{c \in C} V_c), \quad (5.1.5)$$

where the sum is taken over all cliques. So the question which we now address is "what restrictions must be placed on the functions V_c in order to get a well defined probability function." In the discrete case, we require that the sum over all states of X be a finite number, i.e., $Z = \sum_{x \in \Omega} \exp(- \sum_{c \in C} V_c(x_c)) < \infty$. As long as the function can be properly normalized, then it defines a probability function.

So the next question is whether or not joint probability function of Eq. 5.1.5 has the type of conditional probability structure we desire. It turns out that it does and we might attempt to solve this problem by expressing the relationship between the " V_c " functions and the G 's. But, this is not the easiest way to proceed. Instead, we will prove directly that the V_c functions can be chosen arbitrarily.

Theorem 2. Given that the sum over all states of $\exp(- \sum_{c \in C} V_c)$ is a finite positive number (Z), then

$$P(x_1, x_2, \dots, x_n) = \frac{1}{Z} \exp(- \sum_{c \in C} V_c) \quad (5.1.6)$$

is a probability function and

$$P(x_i | x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i | x_j, j \in \partial_i). \quad (5.1.7)$$

Proof: We begin by reemphasizing that there are no restrictions on the values of the probability function except that each probability must be non-negative and that the sum of all probabilities must be exactly one. Since the exponential function is non-negative, then P is a probability function.

To continue, we recall that

$$P(x_i | x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = \frac{P(x_1, x_2, \dots, x_n)}{P(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)} \quad (5.1.8)$$

Since $P(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$ is the marginal probability, it follows that

$$\begin{aligned} P(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) &= \sum_{x_i} P(x_1, x_2, \dots, x_n) \\ &= \sum_{x_i} \left\{ \frac{1}{Z} \exp(-\sum_{c \in C} V_c) \right\} \end{aligned} \quad (5.1.9)$$

where \sum_{x_i} is the sum over all possible values of x_i . Using the above equations,

$$P(x_i | x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = \frac{\frac{1}{Z} \exp(-\sum_{c \in C} V_c)}{\sum_{x_i} \left\{ \frac{1}{Z} \exp(-\sum_{c \in C} V_c) \right\}} = \frac{\exp(-\sum_i V_c)}{\sum_{x_i} \exp(-\sum_i V_c)}, \quad (5.1.10)$$

where \sum_i in the exponent refers to the sum over all cliques containing x_i . The last equality holds since all of the other terms factor out and cancel since the cliques for these terms are independent of the summation over x_i . That this expression is the same as $P(x_i | x_j, j \in \partial_i)$ is intuitively reasonable but will be proven below, as it is not obvious.

In order to establish this final fact, we will need additional notation. Let $\sum_{-\partial_i}$ denote the sum over all of the possible values of x_j for all j such that j is not i and is not a neighbor of i . As above, let \sum_i denote the sum over all cliques which depend on i and let \sum_{-i} denote the sum over all cliques which do not depend on i . It follows that

$$\begin{aligned}
P(x_i | x_j, j \in \partial_i) &= \frac{P(x_i, x_j, j \in \partial_i)}{P(x_j, j \in \partial_i)} = \frac{\sum_{c \in C} \exp(-\sum V_c)}{-\partial_i} \\
&= \frac{\sum_{x_i - \partial_i} \exp(-\sum V_c) \exp(-\sum V_c)}{\sum_{x_i - \partial_i} \sum_{-i} \exp(-\sum V_c) \exp(-\sum V_c)} \\
&= \frac{\exp(-\sum V_c) \sum_{-i} \exp(-\sum V_c)}{\sum_{x_i} \exp(-\sum V_c) \sum_{-i} \exp(-\sum V_c)} \\
&= \frac{\exp(-\sum V_c)}{\sum_{x_i} \exp(-\sum V_c)} \tag{5.1.11}
\end{aligned}$$

Note that the positivity condition is automatically satisfied since the exponential function is positive. Note that we have also shown that there are no restrictions on the choice of the potential functions (the V_c 's) except that they must depend only on values in cliques and that they sum to a finite value. Also note that Eq. 5.1.6 expresses the joint probability function as a Gibbs distribution.

The equivalence between a MRF and a GRF was also proved by Beseg[2] using a factorization theorem. Through his proof of the Hammersley-Clifford theorem, Beseg gives the most general form for the conditional probability such that it is consistent with the joint probability given by Eq. 5.1.4 in Theorem 1.

The preceding theorem indicates that the global characterization of a GRF by its joint probability function (Eq. 5.1.6) encapsulates the local characteristics of a MRF; see Eq. 5.1.7. This allows us to capture the global characteristics in modeling facies distributions while using the local characteristics to sample the distribution. Theorem 2 also shows that to define a MRF on S , it suffices to define the neighborhood system ∂ , specify the associated cliques and then define the clique potential functions. How to choose the forms and parameters of the potential function is the focus of much of the remainder of this work. It should be pointed out that for neighborhood system, there are different types of cliques, but we do not need to define explicitly potential functions for all of them to capture structural features. We can specify nonzero values on some of them to capture structural properties and simply define the potential to be zero on the other cliques.

Markov chain Monte Carlo sampling

Here, we assume that we have partitioned the domain of interest into n pixels. To each pixel s , there is an associated random variable X_s . We define the random field X by

$$X = (X_1, X_2, \dots, X_n) \tag{5.1.12}$$

where the random variable X_s , which can take on any one of the K values from the facies set F , represents the facies in pixel s . As in our previous discussion, S denotes the entire set of pixels (grid-blocks). We also assume that X is a Markov random field (MRF) with respect to some neighborhood system ∂ defined on S and that its probability mass function is a Gibbs distribution. Thus, we can define conditional probability distributions consistent with the joint probability function for X using the results of the Hammersley-Clifford theorem as presented in Theorem 2. In this section, we use s and r to index pixels, and i and j to represent states of a Markov chain for convenience.

The objective is to apply a Markov chain Monte Carlo (MCMC) method to sample X , which we also refer to as sampling the probability distribution for X . This probability distribution is again denoted by $P(x)$. For our application, a Markov chain may be thought of as a sequence of random vectors, X^ℓ , $\ell=0, 1, 2, \dots$. Here each X^ℓ has the form,

$$X^\ell = (X_1^\ell, X_2^\ell, \dots, X_n^\ell), \quad (5.1.13)$$

and the random variable X_s^ℓ represents the facies at pixel s in state ℓ . Each X^ℓ will have its own probability function, $P_\ell(x)$, with the specific form of this probability determined by how the stochastic variables (the X^ℓ) in the Markov chain are defined. The theory of Markov chains is summarized below. Additive details can be found in Refs. [26] and [51]. Applications to petroleum engineering problems can be found in Refs. [5] and [6]. Our objective is to generate realizations of X by constructing a sequence x^ℓ where x^ℓ is a sample of X^ℓ . If the Markov chain is constructed properly, eventually, the sequence of samples we construct will represent samples of X . We refer to any specific sample x^i generated in the Markov chain as state i . For the problem of interest here, the state space Ω refers to all possible realizations of the random vector X . For the problem of interest to us, at each of the s pixels, we can have one of K facies so the dimension of the state space Ω is K^n . A Markov chain with a finite state space is referred to as a finite Markov chain.

The objective is to construct the Markov chain such that

$$P_\ell(x) \approx P(x), \quad (5.1.14)$$

for all $\ell \geq L$ regardless of how $P_0(x)$ (the probability function for X^0) is defined. More formally, if a Markov chain satisfies

$$\lim_{\ell \rightarrow \infty} P_\ell(x) = P(x), \quad (5.1.15)$$

for any $P_0(x)$ defined on X^0 , we say that the Markov chain converges to the correct probability function. If Eq. 5.1.14 holds and x^ℓ is a sample of X^ℓ (or $P_\ell(x)$) then the set of realizations x^ℓ , $\ell=L, L+1, \dots$ represents an approximate sampling of X .

$$\alpha_{i,j} = \min \left[1, \frac{q_{j,i}P(x^j)}{q_{i,j}P(x^i)} \right], \quad (5.1.16)$$

which is often referred to as the Hastings-Metropolis condition. If $q_{ij} = q_{ji}$, then Eq. 5.1.16 reduces to

$$\alpha_{i,j} = \min \left[1, \frac{P(x^j)}{P(x^i)} \right], \quad (5.1.17)$$

which is the well known Metropolis [37] condition. The product of the q_{ij} and α_{ij} gives the transition probabilities,

$$\pi_{i,j} = q_{i,j}\alpha_{i,j}, \quad (5.1.18)$$

and the $|\Omega| \times |\Omega|$ matrix

$$\pi = [\pi_{i,j}], \quad (5.1.19)$$

is referred to as the transition probability matrix.

To construct a sequence of states in the Markov chain, we generate a sample x^i of X^0 by sampling $P_0(x)$. Subsequent states are generated by the following procedure. If $x^\ell = x^i$ is a realization of X^ℓ (ℓ^{th} state generated in the Markov chain), we obtain a realization of $X^{\ell+1}$ as follows: propose a new state x^j (realization of X) with probability q_{ij} , and accept this new state with probability $\alpha_{i,j}$. If the new state is accepted, set $x^{\ell+1} = x^j$, otherwise, set $x^{\ell+1} = x^i$. The transition probability $\pi_{i,j}$ gives the probability that we will obtain state x^j at step $\ell+1$ (i.e., obtain x^j as a realization of $X^{\ell+1}$) given that $X^\ell = x^i$. If, as in our application, the transition probabilities are independent of ℓ , the Markov chain is said to be homogeneous.

We let $\pi_m = [\pi_{ij}^{(m)}]$ denote the m^{th} power of the transition probability matrix, i.e., $\pi_m = \pi^m$. If for any i and j , there exists a finite m such that $\pi_{ij}^{(m)} > 0$, then the Markov chain is said to be irreducible. If a chain is irreducible, then there is a nonzero probability of obtaining any state in the state space in a finite number of steps. If there exists at least one i such that $\pi_{i,i} > 0$, the state space is said to be aperiodic. The following theorem is well known (Refs. [21], [26]).

Theorem 3: A finite, homogeneous, irreducible, aperiodic Markov chain with acceptance probabilities defined by Eq. 5.1.16 is convergent, i.e.,

$$\lim_{\ell \rightarrow \infty} P_\ell(x) = P(x), \quad (5.1.20)$$

for any $P_0(x)$ defined on X^0 .

Next, a Markov chain Monte Carlo (MCMC) method for generating facies distributions is considered. Assume that there are K facies in the set $F=\{1,2,\dots,K\}$, and the "prior" probability of drawing facies k is given by T_k , $k=1, 2,\dots,K$, where we require that

$$\sum_{k=1}^K T_k = 1. \quad (5.1.21)$$

Let $x=(x_1,x_2,\dots,x_n)$ denote any state of X (see Eq. 5.1.13) and let $n_k(x)$ be the number of pixels that are occupied by facies k in the realization x . Then to honor this prior probability distribution, it is desirable that

$$\frac{n_k(x)}{n} \approx T_k. \quad (5.1.22)$$

Since we wish to use the results of the Hammersley-Clifford Theorem to define conditional probabilities, we define the probability function on X by

$$P(x) = P(X = x) = \frac{1}{Z} \exp\{-\sum_{c \in C} V_c(x_c)\}. \quad (5.1.23)$$

As previously, the MRF hypothesis implies that the conditional probability of the random variable X_s is given by

$$P(X_s = x_s | X_r = x_r, 1 \leq r \leq n, r \neq s) = \\ P(X_s = x_s | X_r = x_r, r \in \partial_s) = \frac{1}{Z_s} \exp\{-\sum_{c:s \in C} V_c(x_c)\}, \quad (5.1.24)$$

where Z_s is a normalizing constant and the last sum in Eq. 5.1.24 is over all cliques that contain the pixel s , i.e., over all cliques contained in the neighborhood of s . As before, it is simpler to rewrite Eq. 5.1.24 as

$$P(x_s | x_r, r \neq s) = \frac{1}{Z_s} \exp\{-\sum_{c:s \in C} V_c(x_c)\}. \quad (5.1.25)$$

Now suppose $x^{(1)}$ and $x^{(2)}$ are two realizations of X which differ only at pixel s , i.e.,

$$x^{(1)} = (x_1, x_2, x_{s-1}, x_s^{(1)}, x_{s+1}, \dots, x_n) \quad (5.1.26)$$

and

$$x^{(2)} = (x_1, x_2, x_{s-1}, x_s^{(2)}, x_{s+1}, \dots, x_n). \quad (5.1.27)$$

Now we have,

$$P(x^{(1)}) = P(x_s^{(1)} | x_r, 1 \leq r \leq n, r \neq s) P(x_1, \dots, x_{s-1}, x_{s+1}, \dots, x_n) \quad (5.1.28)$$

where the last probability function in Eq. 5.1.28 is the marginal probability for the remaining $n-1$ random variables. Similarly,

$$P(x^{(2)}) = P(x_s^{(2)} | x_r, 1 \leq r \leq n, r \neq s) P(x_1, \dots, x_{s-1}, x_{s+1}, \dots, x_n). \quad (5.1.29)$$

Dividing Eq. 5.1.29 by Eq. 5.1.28 and using Eq. 5.1.25, it follows that

$$\frac{P(x^{(2)})}{P(x^{(1)})} = \frac{\exp\{-\sum_{c:s \in C} V_c(x_c^{(2)})\}}{\exp\{-\sum_{c:s \in C} V_c(x_c^{(1)})\}} \quad (5.1.30)$$

MCMC Algorithm

Here, we present the Metropolis-Hastings algorithm for generating facies distributions on a system of pixels S . To generate the first state (realization of X^0) in the Markov chain, one simply cycles through the pixels $s = 1, 2, \dots, n$ and at pixel s , one assigns facies k with probability T_k . This gives the initial facies distribution generated from the prior probability distributions $\{T_k, k=1, 2, \dots, K\}$. Given that we have generated state x^i as a sample of X^ℓ at step ℓ , the next state (sample of $X^{\ell+1}$) is generated by the following procedure.

(i) Randomly choose a pixel s in S . Choose facies k' with probability $T_{k'}$ from the set of facies and propose replacing the current facies located at pixel s by facies k' . If we choose the same facies that already occupies pixel s , there is no point in continuing so we actually exclude this probability and modify the T_k 's appropriately. The state (overall facies distribution) that would be obtained by making this change at pixel s represents the proposed state x^j . Note that the probability of proposing a transition from state i to state j is then given by

$$q_{ij} = T_{k'} / n, \quad (5.1.31)$$

where $1/n$ is the probability of choosing pixel s , and $T_{k'}$ is the probability of selecting facies k' from the prior probability function. (If all facies are equally probable, then $T_{k'} = 1/K$.) Also note that if pixel s is occupied by facies k in state x^i , then the probability of proposing the reverse transition from x^j to x^i is given by

$$q_{ji} = T_k / n. \quad (5.1.32)$$

(ii) We have left to define the probability, α_{ij} . From Eq. 5.1.30, we have

$$\frac{P(x^j)}{P(x^i)} = \frac{\exp\{-\sum_{c:s \in C} V_c(x_c^j)\}}{\exp\{-\sum_{c:s \in C} V_c(x_c^i)\}} \quad (5.1.33)$$

Once the potential functions have been defined, using Eqs. 5.1.33, 5.1.31 and 5.1.32 in the Hastings-Metropolis condition (Eq. 5.1.16) defines α_{ij} .

(iii) Accept the new state x^j with probability α_{ij} . This is done by generating a random number with a random number generator (RAN2) first. If α_{ij} is greater than or equal to the random number generated, then replace the old state x^i with the proposed new state x^j . Otherwise, keep the old state x^i as the new state.

To assure that x^l is the right sample from the probability distribution $P(x)$, steps (i) through (iii) may have to be repeated many times, say for L iterations, where here, one iteration actually represents n attempted replacements, one for each pixel on average. According to Theorem 3, once L is sufficiently large (formally as L goes to infinity), the set of samples from that point on will represent a correct sampling of $P(x)$.

Note that the only difference between x^i and x^j is in the selected pixel s . Since the limiting distribution $P(x)$ appears only in the ratio $P(x^j) / P(x^i)$ and the Markov hypothesis applies, so only those cliques containing pixel s have to be included when evaluating Eq. 5.1.16; see Eq. 5.1.33. This is the computational advantage of using MRF models. Also note that Eq. 5.1.33 does not require knowledge of the normalizing constant (Z).

This basic MCMC procedure is done in the subroutine SIMULATION in the FORTRAN program. (see Appendix 5A)

MODELING GEOLOGICAL FACIES DISTRIBUTIONS BY MRFS

As discussed in the introduction, our goal is to generate geological facies distributions in a particular depositional system by using a MRF model that encompasses knowledge obtained from geological information or interpretation. We seek a MRF model that can generate realizations of facies distributions that exhibit the expected geologic features. The first assumption in stochastic modeling is that a realization of the facies distribution can be generated as a sample from a probability distribution, in our case, a Gibbs distribution. We have seen that the Markov random field model defines a Gibbs joint probability distribution on a set of random variables representing facies values. Of course, we want the MRF model to impose statistical dependence in a spatially meaningful way. This requires an appropriate definition of the conditional probabilities or the Gibbs clique potential functions. Our general knowledge about facies distribution includes the proportion, continuity, orientation and ordering of the facies. Therefore, we must consider how to impose this "real-world knowledge" on the definition of clique potential functions. It should be noted that MRF models have been extensively and successfully used in image analysis for at least two decades. Models that are relevant to our problem include the so-called multi-level logistic model of Derin and Elliott [14] and the models of Strauss [47], and Beseg [4]. Although the body of literature concerning MRF models in image analysis is vast, application of MRFS to the modeling of geological facies distribution has only been considered recently; see Refs. [52], [53] and [55]. Thus, problems such as the selection of the order of the neighborhood system, the choice of cliques, the definition of potential functions, phase transitions, boundary effects, and estimation of the model parameters, need to be studied systematically. Below, we concentrate on appropriate definitions of the clique potential functions and associated issues.

MLL Model

As described previously, the joint probability mass function of a Gibbs distribution with $T=1$ is

$$P(x) = \frac{1}{Z} \exp\{-U(x)\} \quad (5.2.1)$$

and the most general form of energy $U(x)$, as shown in the Hammersley-Clifford Theorem, is

$$U(x) = \sum_{1 \leq i \leq n} x_i G_i(x_i) + \sum_{1 \leq i < j \leq n} x_i x_j G_{i,j}(x_i, x_j) + \dots + x_1 x_2 \dots x_n G_{1,2,\dots,n}(x_1, x_2, \dots, x_n). \quad (5.2.2)$$

Since we consider only homogeneous MRFs, i.e., assume that the clique potential functions are independent of the locations of the cliques, we can rewrite the most general form of $U(x)$ as the sum of potential functions over all types of cliques, i.e.,

$$U(x) = \sum_{i \in C_1} V_C(x_i) + \sum_{(i,j) \in C_2} V_C(x_i, x_j) + \sum_{(i,j,k) \in C_3} V_C(x_i, x_j, x_k) + \dots \quad (5.2.3)$$

where C_1 , C_2 and C_3 simply represent the sets of one-, two-, and three-pixel cliques respectively. We will see that there are different types of cliques in each set. For example, in a two-dimensional first order neighborhood system, there are two types of two-pixel clique in set C_2 , e.g., horizontal cliques and vertical cliques.

The widely used Multi-Level Logistic (MLL) model for image processing applications defines clique potential functions in terms of whether all pixel values in a clique are the same or not. A clique potential function does not depend on exactly how pixel values vary within the clique. Tjelmeland and Holden [54] tried this model. They defined all $V_c = 0$ except on two-pixel cliques in a 2D, second-order neighborhood system. Thus, the energy function is

$$U(x) = \sum_{(i,j) \in C_2} V_C(x_i, x_j). \quad (5.2.4)$$

On cliques containing exactly two pixels, say pixels i and j , define

$$V_c(x) = V_{ij} = -\beta \quad (5.2.5a)$$

if $x_i = x_j$ and

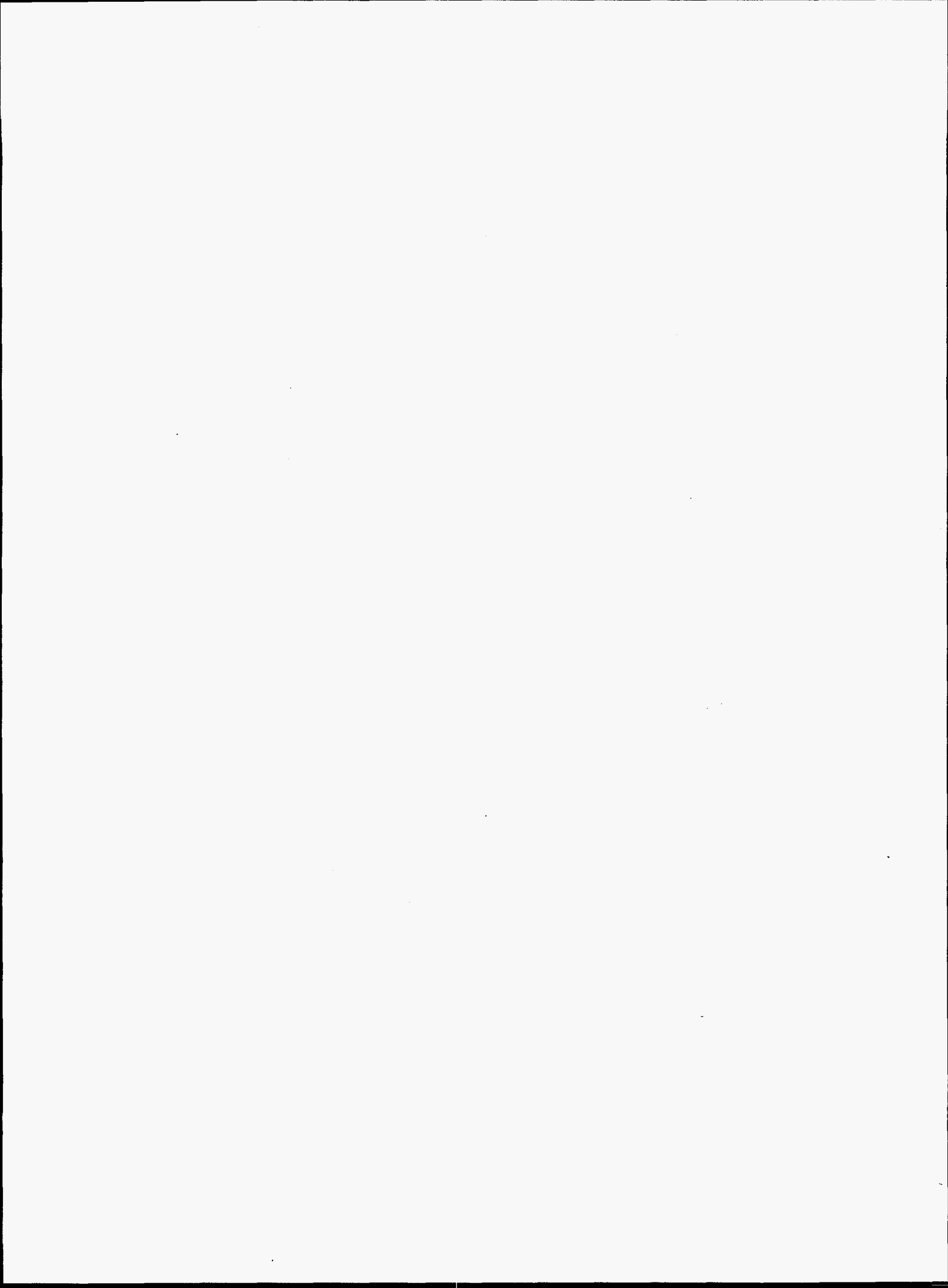
$$V_c(x) = V_{ij} = \beta \quad (5.2.5b)$$

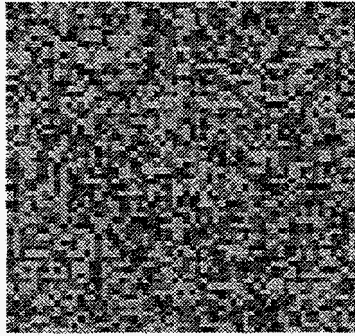
for some $\beta > 0$ if $x_i \neq x_j$. Note with this definition, the most probable realizations of X are those that have the same facies at all pixels. In fact, unless β is very small, Tjelmeland and Holden found that after a sufficiently large number of iterations, the states generated in the Markov chain are all equal to a realization x , where one facies occupies all pixels. On the other hand if β is sufficiently small, the realizations obtained look like random noise.

Fig. 5.2.1(a) through 5.2.1(e) illustrate the case where one facies totally dominates after a large number of iterations have been executed. In this case, we applied the Markov chain Monte Carlo algorithm presented in the last chapter to a five facies example with $T_k = 0.2$ for $k=1,2,\dots,5$, and with $\beta = 0.6$. Fig. 5.2.1(a) shows the initial distribution of facies on a 60×60 grid, i.e., there are $n = 3,600$ pixels. Again, in our terminology, one iteration consists of randomly selecting 3600 pixels in the MCMC simulation procedure. Each time a pixel s is selected, we consider changing the facies at that pixel, i.e., consider changing the value of x_s using the perturbation mechanism described in the MCMC sampling procedure. Fig. 5.2.1(b) through 5.2.1(d), respectively, show the facies distributions obtained after 10, 5,000 and 20,000 iterations. Note that at iteration 5,000 only 3 facies remain and by iteration 20,000, except at a few "stray pixels", only one facies exists.

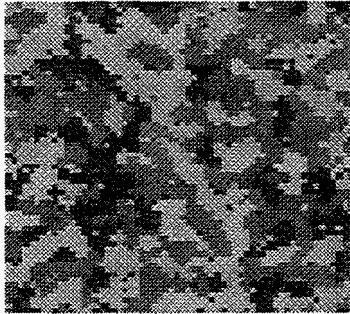
Hjelmeland and Holden [54] noted that some scenes ("realizations") that have been shown in the image processing literature, e.g., the one generated after 10 iterations in the example of Fig. 5.2.1, are not truly representative of the underlying MRF model, but instead represent results generated during the transient part of a Markov chain, i.e., represent scenes generated before the Markov chain has converged to its stationary distribution. The problem of obtaining either a realization that is random noise or one that contains only one facies occurs because of the "phase transition" problem; see Refs. [16], [35] and [40].

Since, one would like most, if not all, realizations to contain all five facies and would also like the prior probability distribution to hold approximately, it is clear the results obtained in this example are inappropriate. To obtain an appropriate procedure, it is necessary to modify the neighborhood system and the associated potential functions.

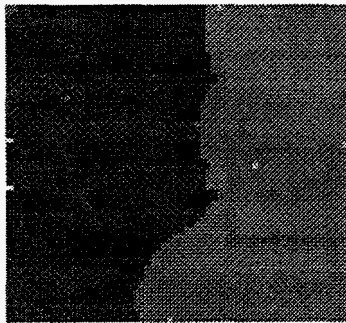




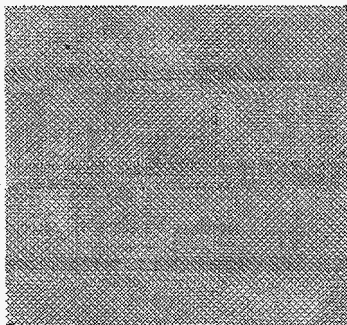
(a) Initial distribution of facies



(b) Facies distribution after 10 iterations.

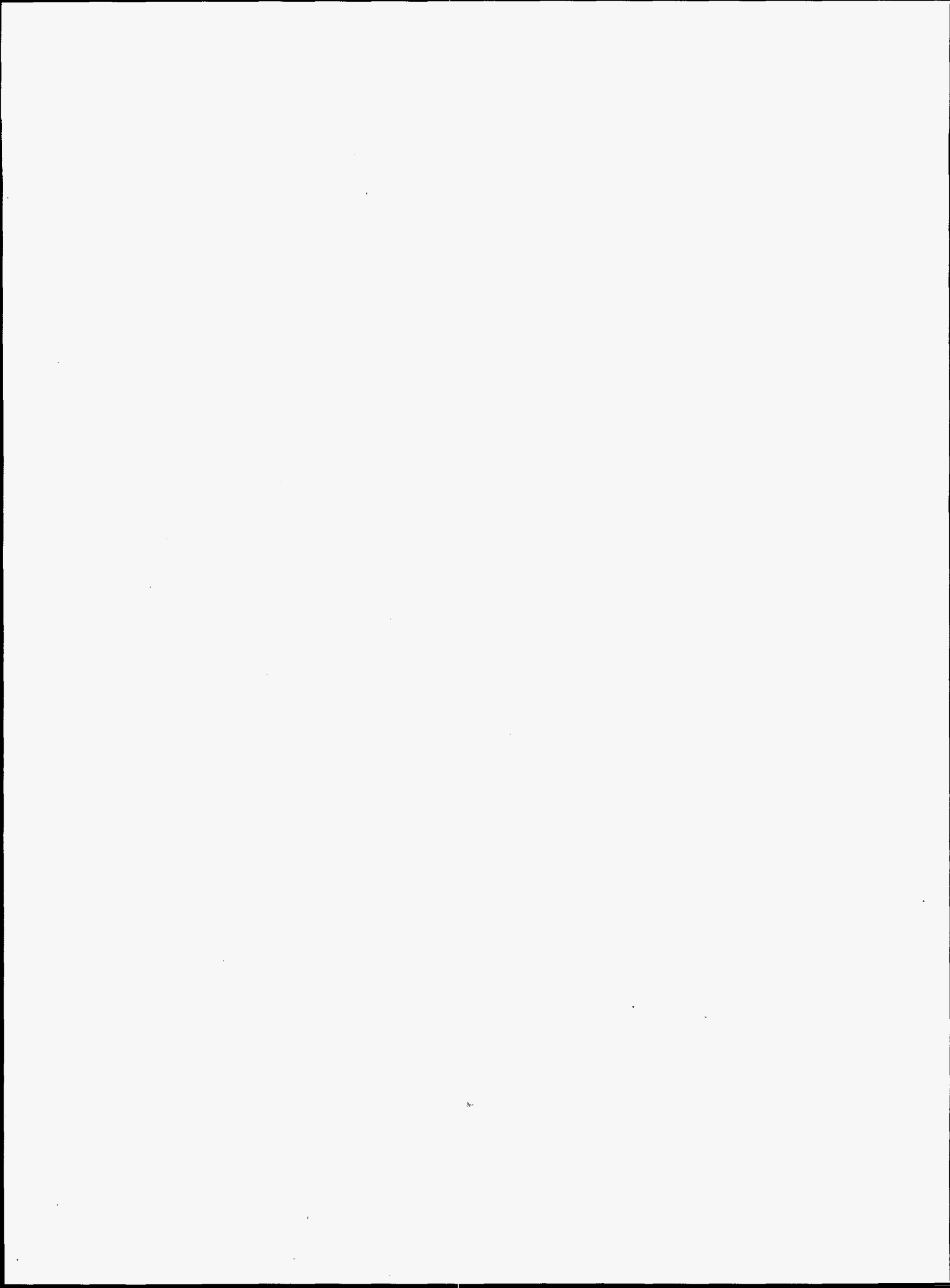


(c) Facies distribution after 5000 iterations.



(d) Facies distribution after 20000 iterations.

Fig. 5.2.1 – Realizations from MLL model



Configuration model

Tjelmeland [52] also noted that semi-MRF's have been proposed (see, for example Ref. [54]) to generate more realistic structures, but the resulting models suffer from slow convergence and may still be unable to control large-scale structures. Thus, following the methodology of Besag[4], Strauss [48], Tjelmeland and Besag [52], he proposed a MRF with higher order interactions, i.e., larger neighborhood systems. In this model, the conditional probabilities or the clique potential function is defined based on specific facies configurations that exist within cliques. A configuration is defined in terms of the facies that exist at each pixel within a clique including the structure of the overall facies distribution within a clique. We refer to this model as a clique configuration model to avoid confusion with the general term configuration that is used in other fields to simply represent a possible x in Ω . This model is sufficiently general to incorporate continuity,

orientation and ordering. If the parameters are appropriate, the proportions of each facies theoretically could also be controlled.

As before, we denote x_c as the facies configuration on a particular type of clique C_p ($p=1,2,\dots,n$) in a neighborhood system, i.e., x_c gives the set of facies occupying the pixels of clique type C_p . C_p is a subset of C . Generally, we use only one or two types of cliques in a neighborhood system, e.g., C_2 or C_3 . So we define $V_c(x_c) = 0$ for other types of cliques. For example, if we wish to control the features of the realization by using only single- and two-pixel cliques in a second-order neighborhood system, we simply consider the potentials for all other cliques (3- and 4-pixel cliques) to be zero. Therefore $V_c(x_c)$ is defined by

$$V_c(x_c) = \begin{cases} \theta_\ell & \text{if } c \in C_p \text{ and } x_c \in D_\ell, \ell = 1, 2, \dots, N_c \\ 0 & \text{if } c \notin C_p \end{cases} \quad (5.2.6)$$

where $D_\ell, \ell=1, 2, \dots, N_c$, are the specified configuration sets for cliques C_p , and N_c is the number of configurations which depends on C_p and the number of facies K . The θ_ℓ 's represent values which define the potential functions. In general we refer to the θ_ℓ 's as the model or statistical parameters. Apparently, the bigger the θ_ℓ , the less likely the corresponding clique configuration since there is a negative sign in front of the summation of potential functions; see Eq. 5.2.1. Conceptually, the values of the θ_ℓ 's control both the structure of any realization obtained as well as the expected value of the relative number of pixels occupied by each facies. However, how to choose the θ_ℓ 's to control the structure of realizations or the number of expected number of pixels occupied by each facies is not an easy problem. If the values of θ_ℓ 's are not appropriate, then the sequence of states generated in the Markov chain may converge to a realization which contains only one facies. This is the well known "phase transition" phenomena which will be discussed in more detail latter.

Possible clique configurations for two-dimensional lattice are shown below. We define a second-order neighborhood system on this lattice as shown in Fig. 5.1.3 and assume potential functions are zero except on two pixel cliques. Also suppose that there are only two possible facies, facies A and B. All possible two-pixel clique configurations as well as the notation used for the corresponding potentials are shown in Fig. 5.2.2.

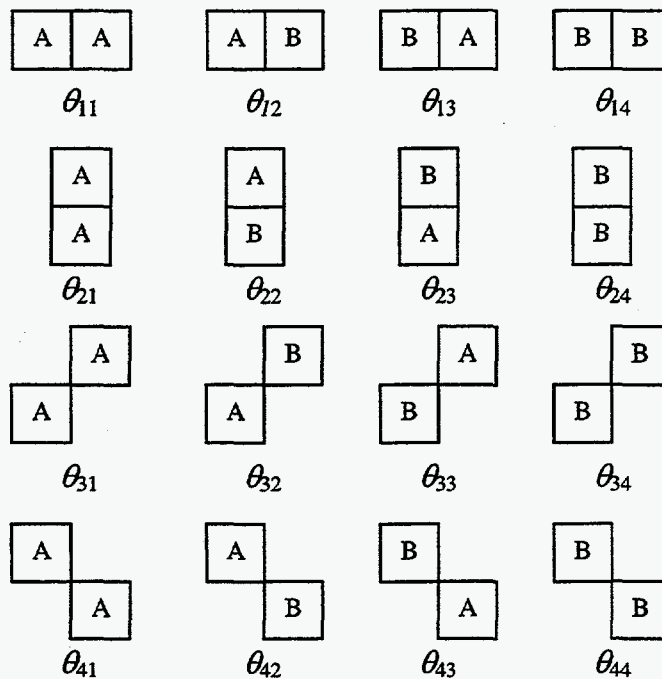


Fig. 5.2.2 – Two-dimensional, 2-pixel clique configurations for 2 facies A and B.

The first configuration with potential θ_{11} affects the continuity of facies A in the horizontal direction. Larger negative values of θ_{11} tend to promote continuity of facies A in the horizontal direction. If we want to promote continuity of facies A in the horizontal direction, we can assign a smaller value to θ_{11} than to other potential functions, otherwise, we increase the value of θ_{11} . Similarly, a low value (or larger negative value) of the potential θ_{43} tends to promote a transition from facies B to A in the northwest to southeast direction, or, more specifically to increase the probability that facies A will occur to the right and below facies B. Intuitively, the clique configurations are very simple, but all parameters have an effect on the realizations generated so choosing appropriate values for a specific geologic model is not always easy. However, we can generate a lot of relatively complicated facies structures by appropriate modification of the potential values. This is a major advantage of the MRF model, i.e., it is possible to determine the global structure, or more mathematically, the joint probability function by using conditional probabilities with local characteristics. It should be pointed out that in general, the above configurations are not considered rotation-free, e.g. $\theta_{11} \neq \theta_{21} \neq \theta_{31} \neq \theta_{41}$. Thus, in general, the Markov random field is not isotropic.

More complicated clique configurations can be obtained by cliques with more pixels as shown in Fig. 5.2.3 in which 5-pixel cliques in a two-dimensional, third-order neighborhood system are used, and 3 facies A, B, C are considered. We could interpret the first configuration (a) as a 2D corner for facies A and the second configuration (b) as straight edge for facies B with facies C located above. However, we prefer not to use cliques containing several pixels because this increases the computational cost, the boundary effect problem tends to become more severe, and the complexity of the implementation increases and intuitive understanding decreases.

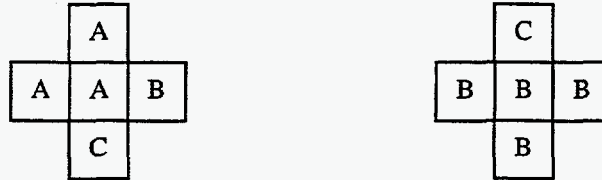


Fig. 5.2.3 – Examples of 5-pixel clique configurations for a 2D, third-order neighborhood system.

Applying the Hammersley-Clifford theorem with the definition of Eq. 5.2.6 gives

$$\begin{aligned}
 P(x) &= \frac{1}{Z} \exp\{-U(x)\} = \frac{1}{Z} \exp\left\{-\sum_{c \in C_p} V_c(x_c)\right\} \\
 &= \frac{1}{Z} \exp\left\{-\sum_{\ell=1}^{N_c} \theta_\ell n_\ell(x)\right\}
 \end{aligned} \tag{5.2.7}$$

where $n_\ell(x)$, $\ell=1, 2, \dots, N_c$, is the number of cliques in x with configuration D_ℓ .

To impose an ordering between facies, Tjelmeland [55] modified the above definition and added additional parameters in the definition of potential functions. For example, he used

$$V_c(x_c) = \theta_\ell^E + \beta_\ell^d(E, E') + \beta_\ell^u(E, E'') \tag{5.2.8}$$

for imposing vertical ordering in three dimensions. Here E is the facies value of the current pixel, E' denotes the facies value located below the current pixel of a clique and E'' denotes the facies located above the center pixel (in vertical direction). β_ℓ^d 's and β_ℓ^u 's are used to impose an ordering on the facies, e.g., facies E' below facies E and facies E'' above facies E . If no ordering is required, those parameters are just set to be zero. This modification appears to introduce additional difficulty in the parameter estimation procedure. Moreover, our experiments showed that this modification is not necessary because we can honor the ordering directly by appropriately defining the θ_ℓ 's. Using only 2-pixel cliques, we will see this clearly from the results presented later. Therefore, we will define potential functions using Eq. 5.2.6.

Phase transition and penalty term

A MRF exhibits a phase transition problem (see Dubes and Jain [17], Pickard [40], Derin [15]), that is, as parameters vary over critical values, abrupt change in statistics occur and the realizations generated may contain only one facies as shown in Fig. 5.2.1.

Pickard [40] interpret the phase transition problem as a specification of parameters such that a short-term correlation between neighboring pixels develops into long-term correlation. From another point of view, the Gibbs probability mass function for the MRF in phase transition has several sharp peaks. Of course, models exhibiting phase transition are not physically reasonable. To avoid the phase transition problem and assure the validity of the models, we propose a modified configuration model based on a global neighborhood system.

For a global neighborhood system and a given pixel i on S , all pixels j with $j \neq i$ are neighbors of i , i.e., $j \in \mathcal{A}_i$, for $j \neq i$. Moreover, all subsets of the complete set of n pixels are cliques. However, we define $V_c = 0$ on cliques unless the clique is contained in C_p , (e.g. except on two-pixel cliques) or the clique is the one containing all pixels. On the clique containing all pixels, V_c is defined as

$$V_c(x) = \sum_{k=1}^K \left| \frac{\bar{n}_k(x) - T_k \times n}{\sigma_k} \right| \quad (5.2.9)$$

where $\bar{n}_k(x)$ represents the number of pixels in the realization x that correspond to facies k . Since this clique contains all pixels of S , we drop the subscript c on x . The σ_k parameter is a scale factor and essentially specifies a term like the standard deviation for the variable $\bar{n}_k(x)$. This V_c function is referred to as a penalty term which penalizes realizations which do not contain a reasonable (in a probabilistic sense) percentage of each facies as prescribed by the prior probability function T_k .

The definition of clique potential function $V_c(x_c)$ for cliques in C_p is similar to Eq. 5.2.6.

Therefore the probability of obtaining realization x is given by

$$\begin{aligned} P(x) &= \frac{1}{Z} \exp \left\{ - \sum_{c \in C_p} V_c(x_c) - \sum_{k=1}^K \left| \frac{\bar{n}_k(x) - n \times T_k}{\sigma_k} \right| \right\} \\ &= \frac{1}{Z} \exp \left\{ - \sum_{\ell=1}^{N_c} n_\ell(x) \theta_\ell - \sum_{k=1}^K \left| \frac{\bar{n}_k(x) - n \times T_k}{\sigma_k} \right| \right\} \end{aligned} \quad (5.2.10)$$

Note that $n_\ell(x)$ and $\bar{n}_k(x)$ are different. The former represents the number of cliques in x with clique configuration D_ℓ , and the later is the number of pixels occupied by facies k in realization x .

MCMC for realizations of facies distribution

To generate a facies distribution, we generate a Markov chain. To do this, we begin by generating an initial distribution of the facies by cycling through all pixels and assigning a facies in each pixel by sampling from the prior distribution, i.e, the probability of selecting facies k in pixel k is T_k , $k=1,2,\dots,K$, where K is the number of possible facies. At each subsequent step, we randomly select one of the n pixels, say pixel s , which is currently occupied by facies k . We let x^i denote the facies distribution obtained as the i^{th} state in the Markov chain and here, also refer to x^i as the current configuration. We propose a new facies k' for this pixel by sampling from remaining $K-1$ facies and let x^j

denote the new distribution of facies obtained by replacing facies k in pixel s by the new facies k' we have selected. In the language of Markov chains, x^i is the current state or i^{th} state and x^j is the proposed state or j^{th} state, where as noted previously, x^i represents the facies distribution determined at the i^{th} step of the Markov chain. The probability of proposing the transition from state i to state j is given by

$$q_{ij} = \frac{T_k}{n}, \quad (5.2.11a)$$

and the probability of proposing the reverse transition is given by

$$q_{ji} = \frac{T_{k'}}{n}. \quad (5.2.11b)$$

Note since we exclude choosing the same facies as the one currently occupied the T_k 's actually must be modified to reflect the probability of drawing a potential facies from the remaining $K-1$ facies. Thus, the Hastings-Metropolis acceptance criterion is simply

$$\alpha_{ij} = \min\left(1, \frac{q_{ji} P(x^j)}{q_{ij} P(x^i)}\right). \quad (5.2.12)$$

Since x^i and x^j differ only at pixel s and the MRF hypothesis applies, we have

$$\frac{P(x^j)}{P(x^i)} = \frac{\exp\left(-\sum_{c:s \in C_p} V_c(x_c^j) - \sum_{k=1}^K \left| \frac{\bar{n}_k(x^j) - n \times T_k}{\sigma_k} \right| \right)}{\exp\left(-\sum_{c:s \in C_p} V_c(x_c^i) - \sum_{k=1}^K \left| \frac{\bar{n}_k(x^i) - n \times T_k}{\sigma_k} \right| \right)}, \quad (5.2.13)$$

where the first sum in each exponential term of the exponents is over cliques that contain pixel s and are in C_p . The \bar{n}_k 's in the second sums are evaluated on all the pixels of S . However, in the implementation of MCMC sampling, we compute all the \bar{n}_k 's when generating the initial distribution and then we simply keep track of how these numbers change from iteration to iteration of the MCMC procedure. This significantly enhances computational efficiency.

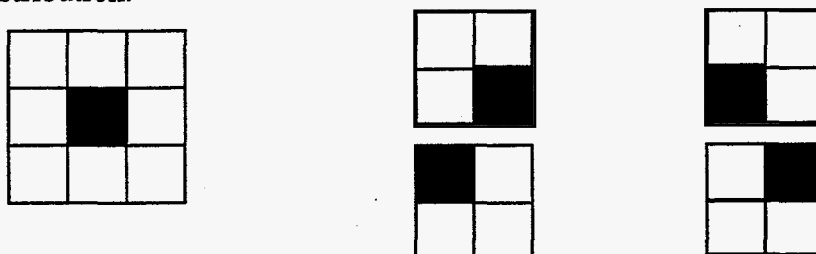
MODEL EXPERIMENTS FOR TWO-DIMENSIONAL PROBLEMS

Here, we present examples of the application of Markov Random Field models to the modeling of geological facies distribution. Particular relevant issues, e.g., boundary effects, clique types, order of the neighborhood system, and conditioning to observed data are also addressed in this section.

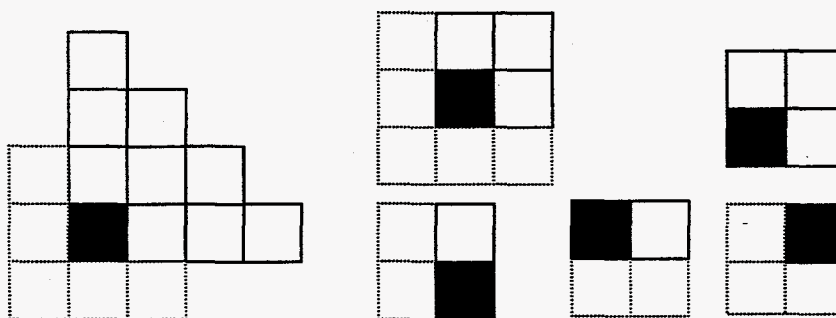
Boundary effects and treatment

As discussed previously, the application of MRFs in practice inevitably deals with a finite region, i.e., S , contains a finite number of pixels. This means that a border or region boundary must be present. Depending on the order of the neighborhood system, the pixels adjacent to the boundary or pixels at some distance away from the boundary have missing neighbors. Fig. 5.3.1 illustrates the boundary effect where a two-dimensional, second-order neighborhood system is used and only 4-pixel cliques are considered. Fig. 5.3.1a shows a complete second-order neighborhood for an interior black pixel and the four 4-pixel cliques that contain this black pixel. Fig. 5.3.1b shows a boundary pixel (black) located in the lower left corner of the two-dimensional region of interest, its incomplete second-order neighborhood and the missing pixels (with dashed line edges) for 4-pixel cliques that contain this pixel.

The boundary problem may be summarized as follows. Consider an isolated system of pixels, S , over which the MRF random variable X , is distributed. The finite region S consists of the set of interior pixels, S_I , which contains those without missing neighbors and the set of boundary pixels denoted by S_B , i.e., $S=S_I \cup S_B$ and $S_I \cap S_B=0$. The values of random variables outside of S are not defined or available. If a pixel i is an interior pixel, that is, $i \in S_I$, then the conditional probability $P(x_i|x_j, j \in \partial_i)$ is completely defined in terms of the potentials of the cliques contained in ∂_i . However, if i is a boundary pixel, that is, $i \in S_B$, then it is not clear how one should define $P(x_i|x_j, j \in \partial_i)$ because, some of the pixels that should be contained in ∂_i are not within the region of interest. We can not expect that $P(x_i|x_j, j \in \partial_i)$ to be equal to $P(x_i|x_j, j \in \partial'_i)$ where $\partial'_i = \partial_i \cap S'$ and $S' = S \setminus i$. Unfortunately, how conditional probabilities are defined at boundary pixels will affect the joint probability distribution.



(a) The second-order neighborhood of interior black pixel and its 4-pixel cliques.



(b) Boundary pixel (black), its second-order neighborhood and 4-pixel cliques, pixels with dashed-line edges are missing.

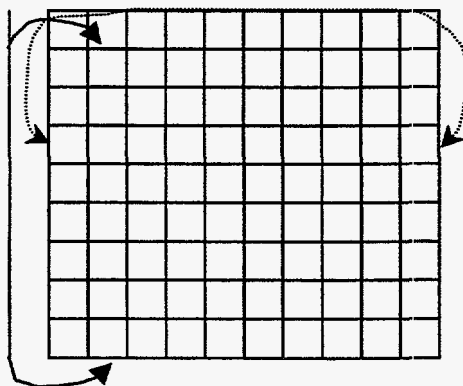
Fig. 5.3.1 – Illustration of boundary effect

Griffith [27] provides both an extensive overview of this problem and a survey of methods for dealing with it. Basically, there are three commonly used approaches in the image analysis literature. In the free boundary treatment, missing pixels are simply ignored. This treatment appears to be most applicable if cliques with less than three pixels are used. If the cliques contain more than two pixels, then only some of the pixels in a clique are missing and it appears that one must still define a conditional probability on the reduced clique. As shown in Fig. 5.3.1 (b), there is only one complete 4-pixel clique and one clique with all the neighboring pixels missing. For the other two 4-pixel cliques, there are two pixels missing. If the free boundary is used, then either we must ignore the clique potential for the later two cliques or define it in some way.

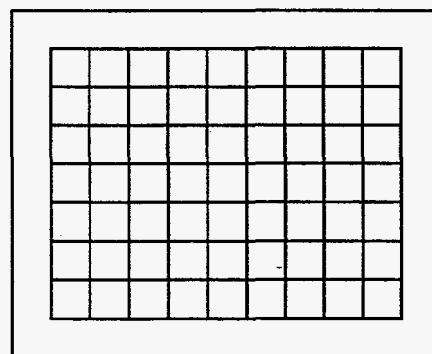
Using a periodic boundary is another alternative that has been proposed for treating boundary pixels. This is illustrated in Fig. 5.3.2 (a). In this procedure, we treat the row of pixels along the top boundary as if they occur immediately below the bottom row of pixels with a similar treatment at other boundaries. The main drawback to this treatment is the introduction of an unexpected and undesirable correlation between boundaries, e.g., between the bottom row of pixels and the top row of pixels.

The final procedure for dealing with boundary pixels is to introduce an artificial guard region or extra border. This is a very easy procedure to implement and guarantees that every pixel in region S has a complete neighborhood and a complete set of cliques. But, this introduces the problem of dealing with pixel values in the guard region, especially at the boundaries of the guard region. If the guard region is large, computational costs are increased significantly. How large a guard region should be used appears to be an unsolved problem.

All three of these methods for dealing with boundary pixels attempt to provide a medium for removing the bias introduced into spatial statistical analysis, especially statistical inference, by the presence of boundaries. Tjelmeland [55] actually proposed a procedure to define potentials on reduced cliques that occur near the boundary and we have tried his and various other methods for defining potentials on reduced cliques, but ultimately, we adopted the free boundary treatment. Since, we use only two pixel cliques, this free boundary treatment is very convenient since reduced cliques never occur.



(a). Torus boundary treatment.



(b). Construction of guard region.

Fig. 5.3.2 – Periodic boundary and guard region for boundary pixels.

Experimental results on second-order models

As discussed before, geological facies distributions, e.g., in a fluvial depositional environment, may exhibit continuity, orientation and variation in facies body size, anisotropy and large-scale facies structures. We first present results for experiments that we have done for 2D problems using a second-order neighborhood. We illustrate how model parameters control the pertinent characteristics of realizations of the facies distribution. Three-dimensional problems are considered latter.

We consider a three facies case, so that we can impose special ordering on the realizations. The 2D, second-order neighborhood system and the type of cliques are depicted in Fig. 5.1.2 (a). If three facies labeled A, B and C are considered and only 2-pixel cliques are used, then the only possible clique configurations are those shown in Fig. 5.3.3. In our discussion, the first subscript on the parameters represents the clique type and the second subscript indicates configuration type. We will refer to horizontal 2-pixel cliques in two dimensions as clique type 1, vertical 2-pixel cliques as type 2, 2-pixel cliques in southwest-northeast direction as clique type 3 and cliques in northwest-southeast direction as type 4.

From Fig. 5.3.3, we note that there are a total of 36 configurations, or parameters, for this three facies problem.

All the realizations assume a 64×64 grid, i.e., 64 gridblocks in each of the two directions. Thus, our domain contains 4,096 pixels. The number of iterations for the MCMC algorithm is 1000 and the free boundary treatment is applied.

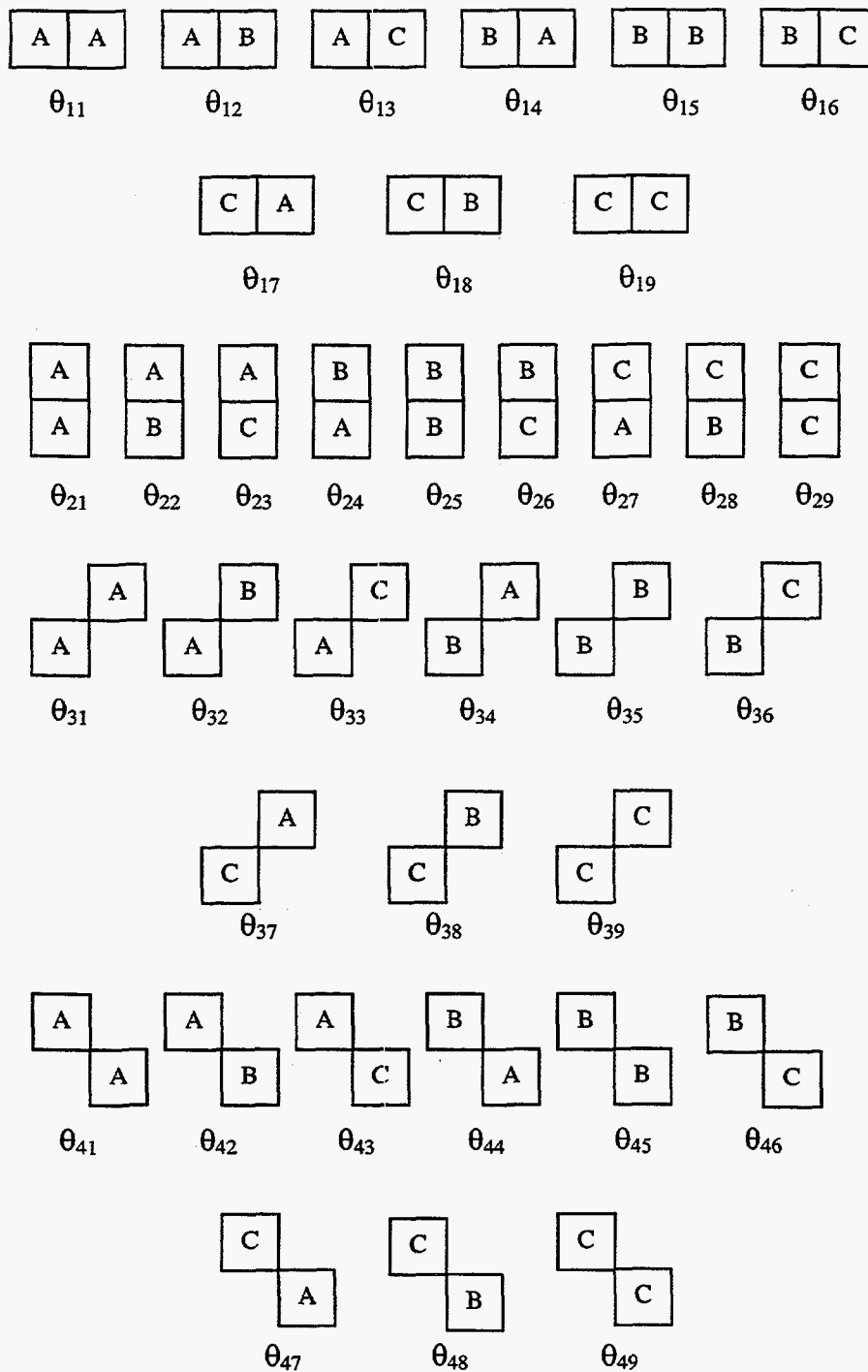


Fig. 5.3.3 – Clique configurations and the corresponding potentials for three facies A, B and C.

We can see from Fig. 5.3.3 that there are a total of 36 configurations or statistical parameters for this three facies case, nine for each type of clique. Each parameter has its own physical interpretation. For example, θ_{31} represents the continuity of facies A in the southwest to northeast direction in the sense that a large negative value of θ_{31} promotes continuity of facies A in this direction that should be exhibitedly in realizations of the facies distribution obtained by sampling the resulting Gibbs probability function. Similarly, the value of θ_{48} has a strong influence on the transition (or ordering) between facies C and A in the northwest to southeast direction. If θ_{48} is a relatively large negative number, we expect to see transitions from C to A in this direction and there will be a tendency for facies C to appear "above" facies A in this diagonal direction. If, for example, more continuity in the horizontal direction is desired in the realization, this can be accomplished by using larger negative values of θ_{11} , θ_{15} and θ_{19} . Although, these simple intuitive arguments are valid and can normally be used to select parameter values so that realizations have the desired features, all parameters influence the joint probability function and the effect that the interaction of all 36 parameters will have on realizations is not always obvious.

All realizations shown in Fig. 5.3.4 again pertain to a 64x64 grid system. However, each realization was generated with a different set of parameter values that were selected to promote a particular type of structure in the realizations. Each realization represents the one obtained after 500 iterations of the MCMC algorithm, and in all cases, the free boundary condition was applied. Facies A is blue, facies B is green and facies C is red. All the potential values for each realization are listed in Table 5.3.1. The purpose of this example is to show that a variety of facies structures (orientation, continuity, ordering and shape of the facies body) can be captured by appropriate selection of the values of potentials, i.e., the θ values.

Note that in Fig. 5.3.4a, each of the facies A, B and C is highly continuous in the horizontal direction because the most negative values were assigned to θ_{11} , θ_{15} and θ_{19} ; all were set equal to -1.5 ; see Fig. 5.3.3 and Table 5.3.1. Also note that in Fig. 3.4(a), facies B (green) tends to be found above facies C (red) and below facies A (blue). These features were promoted by the following strategy. We assigned a relatively large negative number to θ_{26} which gives a high probability to the occurrence of the green facies above the red facies while assigning $\theta_{28} = 0.5$ which gives a much lower probability to the occurrence of the green facies below the red facies. Note we also set $\theta_{24} = 0.5$ to discourage the appearance of the green facies above the blue facies. By using this same type of strategy to assign values for the other θ 's, it was easy to promote the ordering shown in Fig. 5.3.4a. Also note that relatively high values (0.5) were assigned to all θ 's pertaining to diagonal cliques as we did not wish to promote either continuity or transitions in the diagonal directions. Note that we set $\theta_{21} = \theta_{25} = \theta_{28} = -0.5$ as we do wish to have some continuity in the vertical direction.

To obtain the results shown in Fig. 5.3.4 (b)-(d), we assigned relatively large negative values to the potentials that control continuity in particular diagonal directions. It is instructive to compare Fig. 5.3.4b and c. In both cases, we assigned $\theta_{41}=\theta_{45}=\theta_{49}=-1.5$ to promote continuity in the southeast to northwest direction. However, in (b), except for θ_{47} , we used smaller theta values than in (c) for the configurations that promote transitions in the southeast to northwest direction. Thus, in (b), we see more transitions between facies in this direction. In both cases, in the southwest to northeast direction, we see that transitions occur from red (facies C) to blue (facies A) (this transition is controlled by $\theta_{37}=-0.5$ and $\theta_{33}=0.5$), from blue to green (facies B) (this transition is controlled by θ_{32} and θ_{34} and from green (B) to red (C); this last transition is controlled by θ_{36} (negative in both cases) and $\theta_{38}=0.5$. Case b was generated using a penalty term (see Eq. 5.2.9 and the discussion in the next subsection) to ensure that the proportion of each facies in any realization is close to the prescribed value noted in the figure caption.

The results of Fig. 5.3.4d are similar to those of Fig. 5.3.4c except that to generate the realization shown in Fig. 5.3.4d, we specified θ values to promote continuity in the southwest to northeast direction. The most interesting feature of Figs 5.3.4e and 5.3.4f is that there is never a transition from red (facies C) to the green (facies B). This was accomplished by setting the values of θ that control transitions between these two facies to unity to discourage any such transitions.

Conceptually, the values assigned to the statistical parameters should determine the relative proportion of each facies appearing in any realization, however, we have been able to tightly control these proportions only by using a penalty term. The penalty term also causes the model to be stable. In many cases, if the θ 's are not chosen carefully, then after a large number of iterations of the MCMC procedure, the states generated all contain only one facies. As discussed in more detail in the next subsection, the penalty term prevents this.

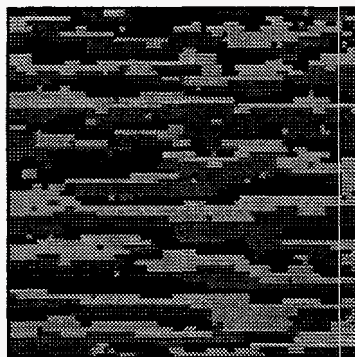
Stability of the model and the penalty term

As noted previously, MRF models suffer from the phase transition problem. In this report, the "phase transition" refers to a selection of model parameters that tend to cause realization to contain only single facies. In other words, the MCMC sampling procedure always generates a realization that contains only one facies if a large number of iterations are done. More specifically, there exists a range of values of the θ 's such that the MCMC sampling procedure will generate realizations which have similar features. But if all parameters are divided by a sufficiently small constant, then the MCMC will generate realizations from some point on such that all states contain only one facies. On the other hand, if the θ values are all divided by a sufficiently large constant, then realizations will always have the appearance of random noise. If the MRF model parameters are such that we generate reasonable realizations, then the model is said to be or referred to a stable or an equilibrium model. If the values of θ 's specified result in a stable model, then from the point in the Markov chain where we begin sampling the probability function correctly, the relative proportion of each facies will oscillate within some reasonable range as the

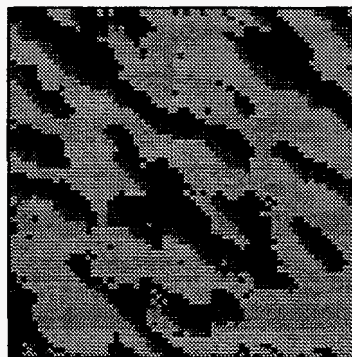
iterations proceed. Fig. 5.3.5 shows 4 realizations from the same model used to generate the results of Fig. 5.3.4 (a).

Table 5.3.1 – Potential values for realizations (a) through (f) in Fig. 3.4

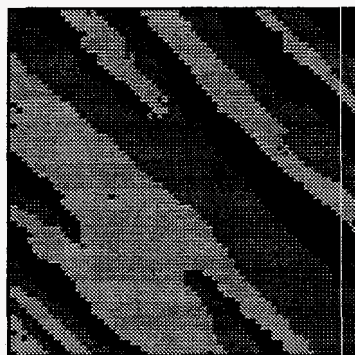
Parameter	(a)	(b)	(c)	(d)	(e)	(f)
θ_{11}	-1.5	-0.3	-0.3	-0.3	-0.6	-0.3
θ_{12}	0.5	-0.2	-0.2	-0.2	1.0	1.0
θ_{13}	0.5	0.5	0.5	0.5	1.0	1.0
θ_{14}	0.5	0.5	0.5	0.5	1.0	1.0
θ_{15}	-1.5	-0.3	-0.3	-0.3	-0.6	-0.3
θ_{16}	0.5	-0.3	-0.2	-0.2	1.0	1.0
θ_{17}	0.5	-0.5	-0.2	-0.2	1.0	1.0
θ_{18}	0.5	0.5	0.5	0.5	1.0	1.0
θ_{19}	-1.5	-0.3	-0.3	-0.3	-0.6	-0.3
θ_{21}	-0.5	-0.3	-0.3	-0.3	-0.5	-0.5
θ_{22}	-0.4	0.5	-0.2	0.5	1.0	1.0
θ_{23}	0.5	-0.3	0.5	-0.2	-0.7	-0.7
θ_{24}	0.5	-0.3	0.5	-0.2	-0.7	-0.7
θ_{25}	-0.5	-0.3	-0.3	-0.3	-0.5	-0.5
θ_{26}	-0.4	0.3	-0.2	0.5	1.0	1.0
θ_{27}	-0.4	0.5	-0.2	0.5	1.0	1.0
θ_{28}	0.5	-0.3	0.5	-0.2	1.0	1.0
θ_{29}	-0.5	-0.3	-0.3	-0.3	-0.5	-0.5
θ_{31}	0.5	-0.5	-0.5	-1.5	-0.3	-0.6
θ_{32}	0.5	-0.2	-0.5	0.5	0.5	0.5
θ_{33}	0.5	0.5	0.5	0.5	0.5	0.5
θ_{34}	0.5	0.5	0.5	0.5	0.5	0.5
θ_{35}	0.5	-0.5	-0.5	-1.5	-0.3	-0.6
θ_{36}	0.5	-0.3	-0.5	0.5	1.0	1.0
θ_{37}	0.5	-0.5	-0.5	0.5	0.5	0.5
θ_{38}	0.5	0.5	0.5	0.5	1.0	1.0
θ_{39}	0.5	-0.5	-0.5	-1.5	-0.3	-0.6
θ_{41}	0.5	-1.5	-1.5	-0.5	-0.4	-0.4
θ_{42}	0.5	0.2	0.5	-0.2	0.5	0.5
θ_{43}	0.5	-0.3	0.5	0.5	0.5	0.8
θ_{44}	0.5	-0.3	0.5	0.5	0.5	0.5
θ_{45}	0.5	-1.5	-1.5	-0.5	-0.4	-0.4
θ_{46}	0.5	0.3	0.5	-0.3	1.0	1.0
θ_{47}	0.5	0.5	0.5	-0.5	0.5	0.5
θ_{48}	0.5	-0.3	0.5	0.5	1.0	1.0
θ_{49}	0.5	-1.5	-1.5	-0.5	-0.4	-0.4



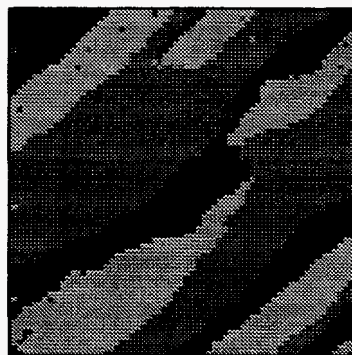
(a) Uniform prior distribution for facies, without penalty term.



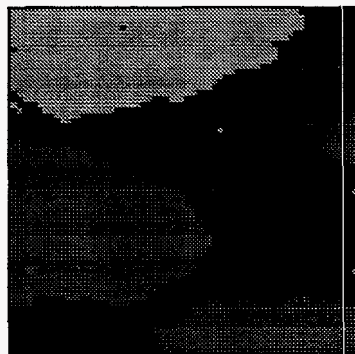
(b) A=30%, B=60% and C=10%, with penalty term and $\sigma_k=1.0$, $k=1,2,3$.



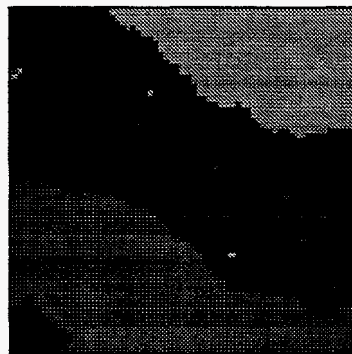
(c) Uniform prior distribution for facies, without penalty term.



(d) Uniform prior distribution for facies, with penalty term and $\sigma_k=1.0$, $k=1, 2, 3$.

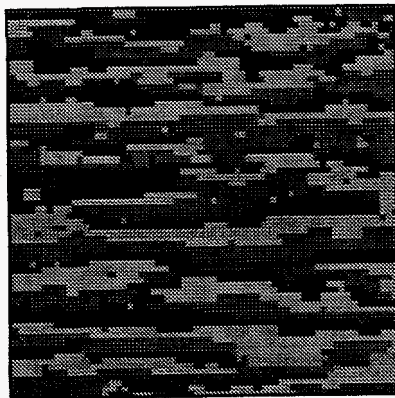


(e) A=50%, B=20% and C=30%, with penalty term and $\sigma_k=1.0$, $k=1,2,3$.

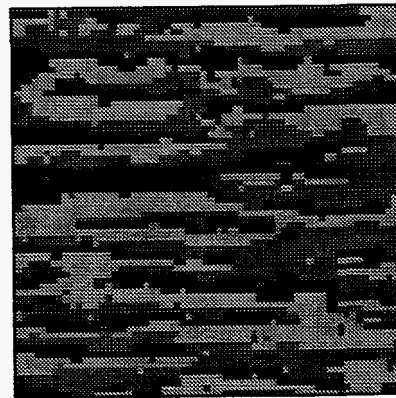


(f) A=50%, B=20% and C=30%, with penalty term and $\sigma_k=1.0$, $k=1,2,3$.

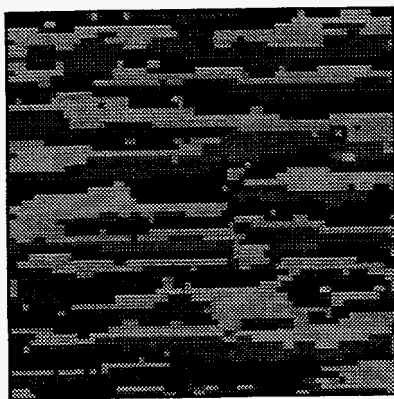
Fig. 5.3.4 – Realizations of 2D, second-order, clique configuration models.



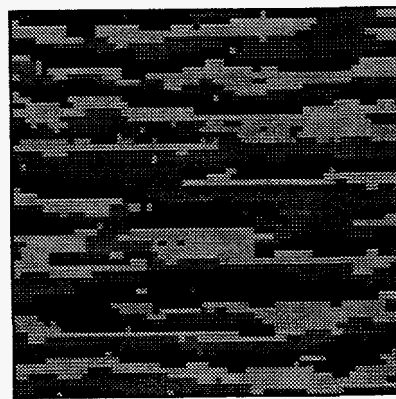
(a) Realization 500 iterations.



(b) Realization after 600 iterations.



(c) Realization after 700 iterations.



(d) Realization after 800 iterations.

Fig. 5.3.5 – Realizations from the model (a) of Table 3.1 show similarity in features.

These realizations are from a long run of the MCMC procedure. Realization (b) occurs 100 iterations after realization (a) and so on. We can see very clearly that the structures in all four realizations are similar; all exhibit continuity in the horizontal direction and the same ordering between facies in the vertical direction. But, the relative proportion of the facies varies from realization to realization as shown in Fig. 5.3.6. The percentage of each facies contained in realizations of Fig. 5.3.5 is as follows: in the realization of Fig. 5.3.5 (a), A (blue) = 30.4%, B (green) = 33.7%, C (red) = 35.9%; in (b), A = 29.2%, B = 35.5%, C = 35.2%; in (c), A = 33.1%, B = 34.6%, C = 32.3% and in (d), A=33.9%, B = 32.4%, C=33.7%.

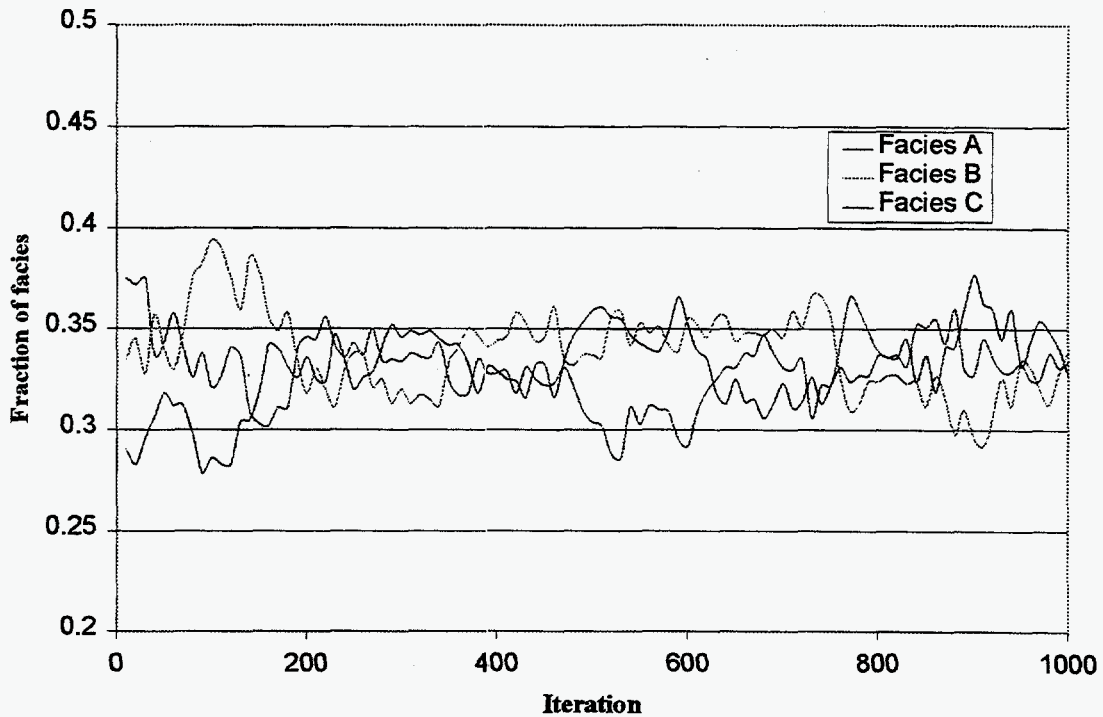


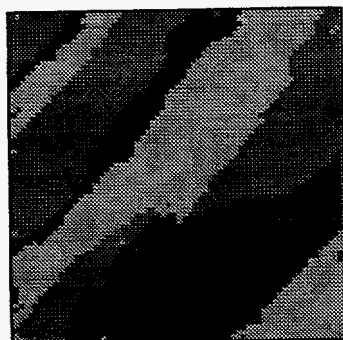
Fig. 5.3.6 – The change of proportions of the facies with the number of iterations.

Although we obviously wish to choose values of the statistical parameters (values of the θ 's) to obtain a stable MRF model, for complicated models with a large number of statistical parameters, it is difficult to estimate θ values that will ensure stability. It is also difficult to estimate which will yield realizations that contain relative proportions of facies within some specified range. Both of these problems can be overcome by using a penalty term. After a number of experiments, we ultimately choose the penalty term given by Eq. 5.2.9 which is repeated here as

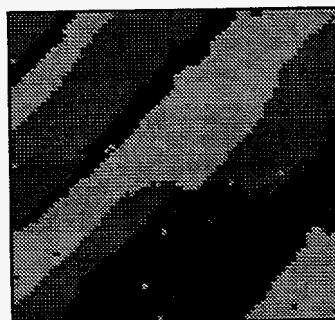
$$V_c(x) = \sum_{k=1}^K \left| \frac{\bar{n}(x) - n \times T_k}{\sigma_k} \right| \quad (5.3.1)$$

In general, if we use Eq. 5.3.1 with $\sigma_k=1$, then the states generated in the MCMC procedure converge to a realization which does not change at subsequent iterations. The results of Fig. 5.3.7 exhibit this phenomenon. The model used to generate the results of Figs. 5.3.7 and 5.3.8 are the same as the model with the set of potentials listed in Table 5.3.1 column d. As can be inferred from the four realizations shown in Fig. 5.3.7, all states generated between iteration 500 and iteration 800 are for all practical purposes identical. Since we wish to generate a set of independent or at least uncorrelated realizations, this is not a desirable feature. However, the problem can be avoided by

simply starting with a different initial random field and constructing another Markov chain.



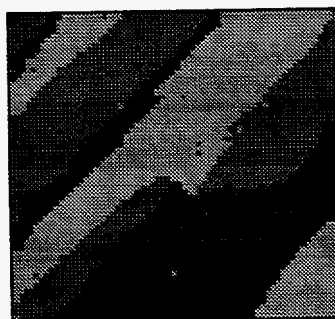
(a) Realization after 500 iterations.



(b) Realization after 600 iterations.



(c) Realization after 700 iterations.



(d) Realization after 800 iterations.

Fig. 5.3.7 – Realizations of the model listed in Table 3.1(d), with penalty term and $\sigma_k=1.0$, for $k=1, 2, 3$.

Fig. 5.3.8 presents the fraction of each facies occurring with a realization versus the number of iterations of the MCMC algorithm. The set of results shown by a solid line pertains to the results of Fig. 5.3.7. Note that after a relatively small number of iterations all realizations contain 33.33% of each facies. The other curves in Fig. 5.3.8 show results obtained when a penalty term is not applied. In this case, the number of pixels occupied by Facies C tends to continually increase. Although not shown, after 3000 iterations, all realizations generated contained only facies C

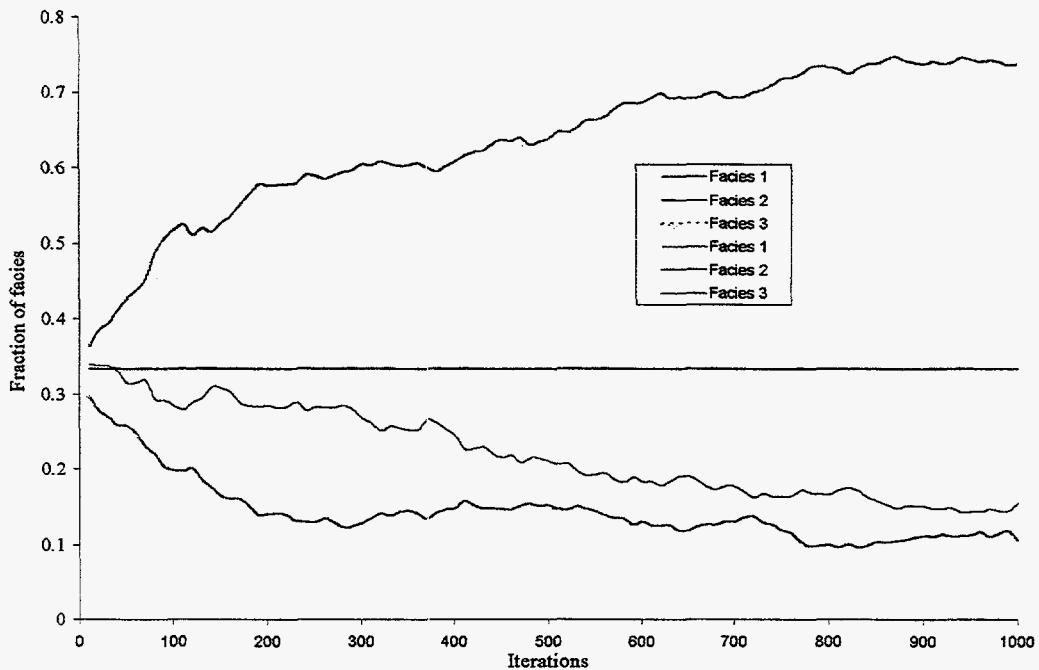


Fig. 5.3.8 – Fractions of facies versus the number of iteration, from the model with penalty term (straight line) and without penalty term.

By relaxing the penalty term, i.e., increasing σ_k , we obtain a model such that the fraction of pixels occupied by each facies is not fixed, but varies within some reasonable range. This relaxation factor is chosen to attempt to specify a reasonable variation in the desired relative proportions of each facies. For example, if we wish to have a 6% variation in the fraction of each facies, then we set (for $k=1, 2, 3$)

$$\sigma_k = 0.06 \times T_k \times n = 0.06 \times \frac{1}{3} \times (64 \times 64) = 72. \quad (5.3.2)$$

where T_k is the prior probability of facies k and n is the number of pixels in the region S . Essentially, σ_k represents a term somewhat similar to the standard deviation for the variable n_k where the mean of each n_k is $n T_k$ and the set of variables $n_k, k=1,2,\dots,K$, have a multivariate Gaussian distribution.

The results of Fig. 5.3.9 pertain to the same model considered in Figs. 5.3.7 and 5.3.8. Note in Fig. 5.3.9, we compare the variation in facies from state to state in the Markov chain generated. Note when the relaxed penalty term is used, the fraction of each facies obtained in the realizations varies within a narrow range. For example, from iteration 300 on, the fraction of facies B obtained varies from 0.27 to 0.39 approximately as we expected from the chosen $\sigma_k=72$. While with $\sigma_k=1$, the fraction of facies B was 0.33 and basically no variation.

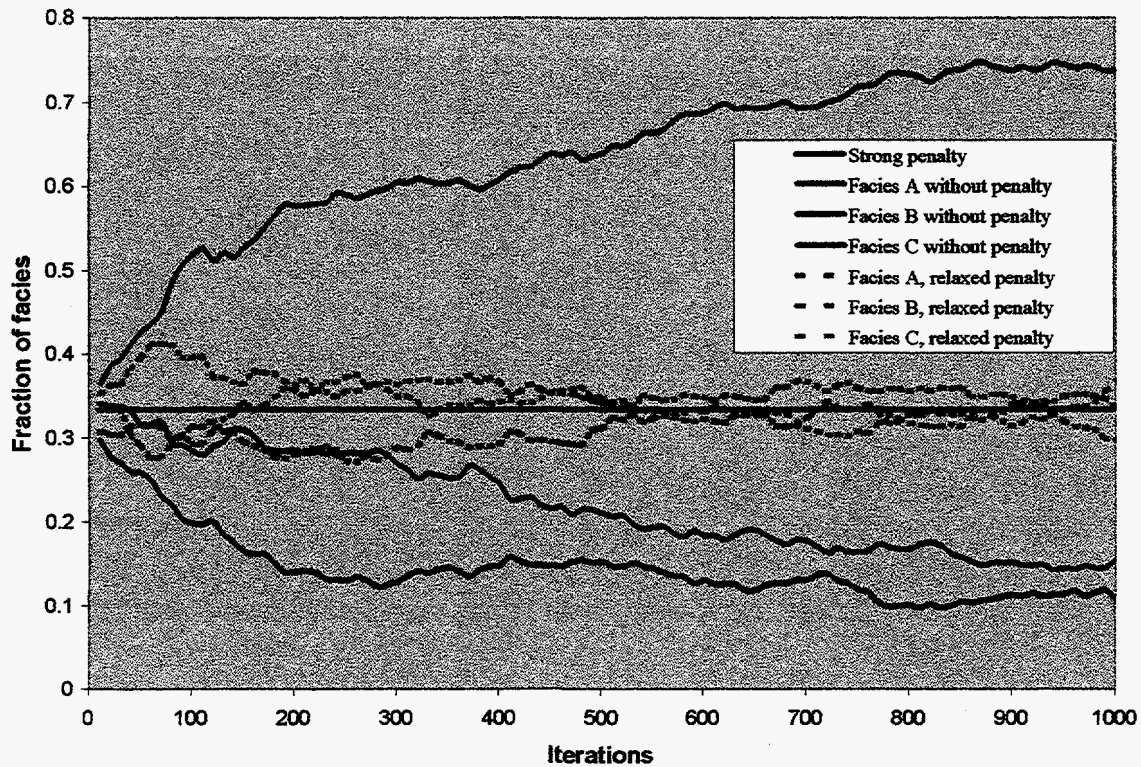


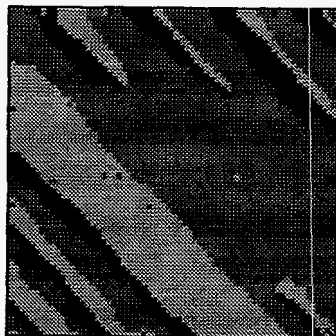
Fig. 5.3.9 – Variation of facies fractions with the number of iterations for the model with strong penalty, relaxed penalty and without penalty.

Conditioning of the model to observations

In generating realizations of the facies distribution with a MRF model, we would like to condition the realizations to well observations. This conditioning is easy to implement. We simply fix the facies (at gridblocks penetrated by wells) to their observed values and do not perturb them during the MCMC sampling procedure. It might also be expected that conditioning realizations to observed data will tend to make the model more stable and reduce the phase transition problem. If all facies are observed at well locations, then all facies but one can not disappear since the observed values are fixed throughout the MCMC simulation. The two examples presented in this subsection indicate that conditioning data reduces the phase transition problem.

In the first example, we apply the model with potential values listed in column d of Table 5.3.1 except that we interchange the potential values in the two diagonal directions to obtain continuity in the southeast to northwest direction. Without a penalty term, this model is unstable; see Fig. 5.3.8. Fig. 5.3.10 shows realizations obtained at iteration 400, 600, 800 and 1000 of the MCMC procedure. As we can see from the realizations, the proportion of facies C found in a realization tends to increase fairly rapidly as the iterations proceed even though at iteration 1000 the basic structure promoted by our choice of the statistical parameters is still maintained. The results of Fig. 5.3.10 were generated without using conditioning data and no penalty term was used. To clearly

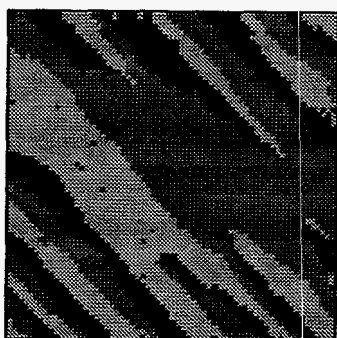
illustrate the effect of conditioning data, we did the same run as in Fig. 5.3.10 up to 400 iteration, but from iteration 400 on we fixed the values in the 32nd column of gridblocks to the values obtained at iteration 400. Thus after 400 iterations, facies values on this column serves as conditioning data.



(a) Realization after 400 iterations.



(b) Realization after 600 iterations.



(c) Realization after 800 iterations.



(d) Realization after 1000 iterations.

Fig. 5.3.10 – Realizations from a degenerated model.

Fig. 5.3.11 shows a plot of facies fractions in realizations obtained by the MCMC method versus the number of iterations. Results are shown both for the case where conditioning data were used and for the case where conditioning data were not used. It is apparent that if no conditioning data is used, realizations will eventually contain only

facies C. However, when conditioning data is used the fraction of pixels occupied by facies C tends to stabilize, or at least its rate of increase is retarded.

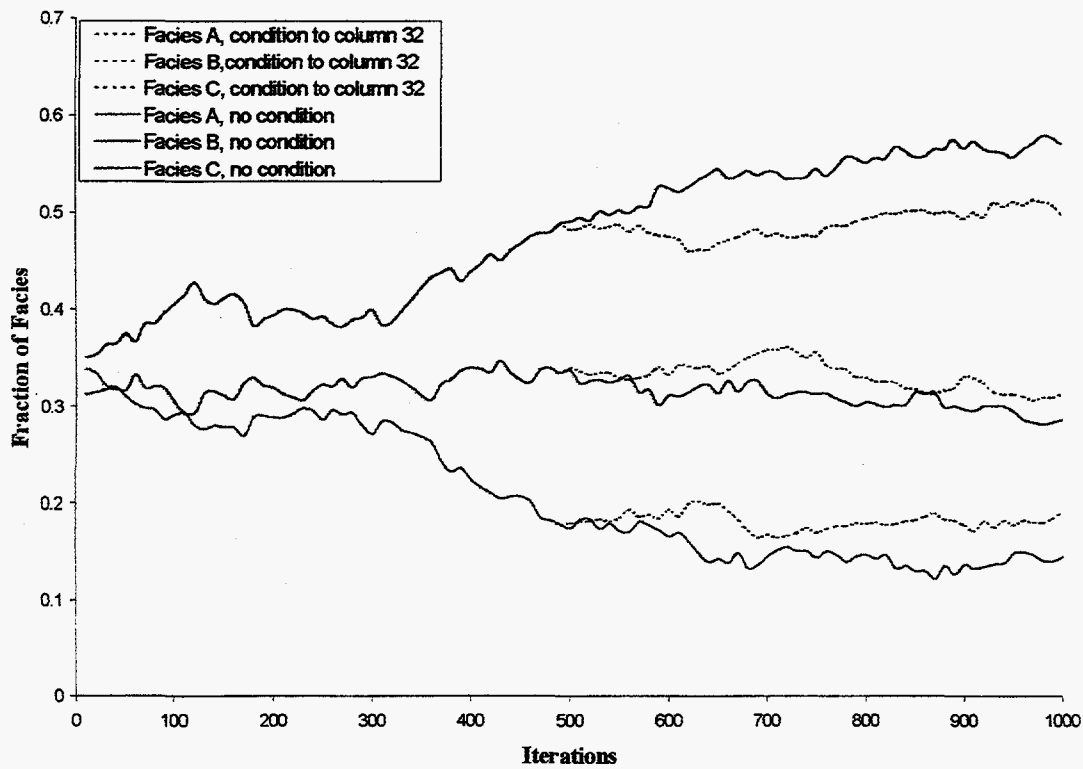


Fig. 5.3.11 – Facies fractions before and after model conditioning.

The results of Fig. 5.3.12 are similar to those shown in Fig. 5.3.11 except in the results of Fig. 5.3.12, we applied the conditioning data beginning with the very first iteration of the MCMC algorithm. Again, although the fraction of pixels occupied by facies C tends to increase, the conditioning data tends to modulate the rate of increase. One would of course expect that adding additional conditioning data with slow the rate of increase even more.

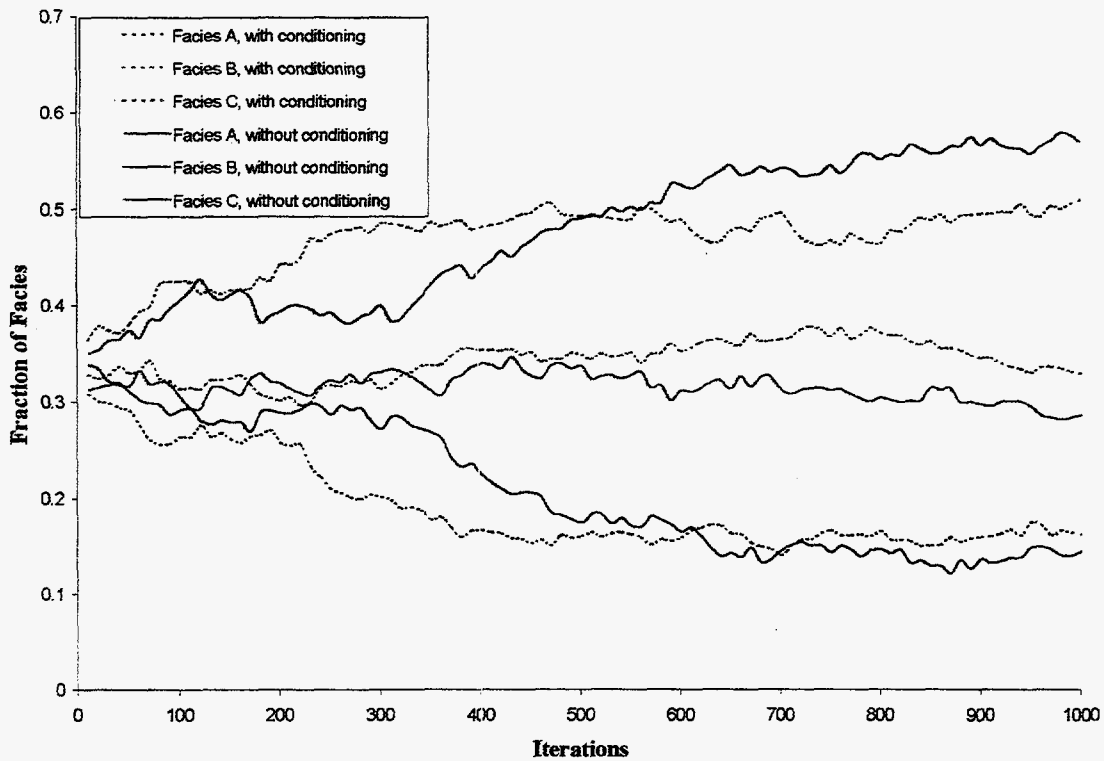


Fig. 5.3.12 – Comparison of facies fractions as the number of iterations.

EXPERIMENTS FOR THREE-DIMENSIONAL PROBLEMS

In this section, we present results for the case where we wish to generate facies distributions in a three-dimensional volume partitioned into pixels with N_x pixels in the x-direction, N_y in the y-direction and N_z in the z-direction. We first consider the 3D model used by Tjelmeland [55]. We use the same three dimensional third order neighborhood system that he used. This neighborhood system, which contain 25 pixels, is shown in Fig. 5.4.1 in terms of layers (vertical or z-direction pixels). The pixels in the neighborhood system are labeled with three indices where the first index as the x-direction index, the second as the y-direction index and the third as the z-direction index. Thus, the center pixel of the neighborhood is (i, j, k) and, for example, pixel $(i, j, k+2)$ presents a pixel two pixels directly above the center pixel in the z-direction. The neighborhood shown in Fig. 5.4.1 contains seven cliques containing exactly seven pixels. These cliques are shown in Fig. 5.4.2.

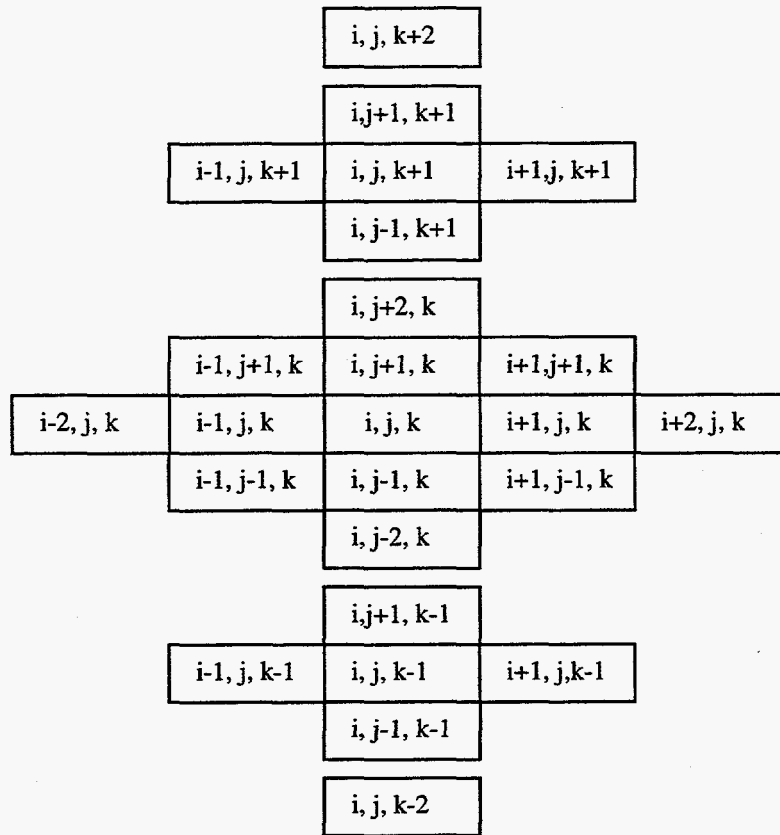


Fig. 5.4.1 - The cut layers of the three dimensional 3rd-order neighborhood system, 24 neighbors, on five layers.

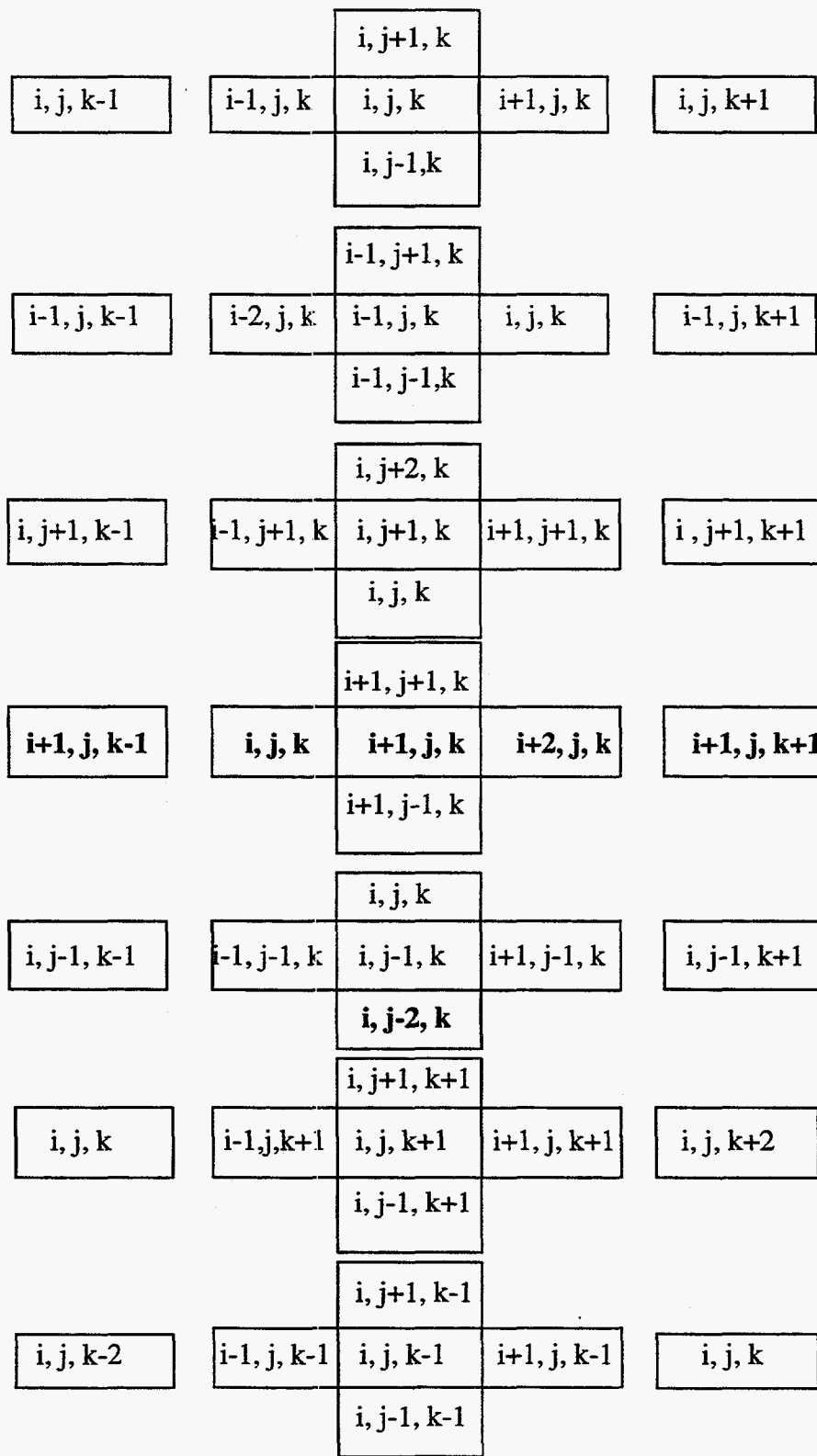


Fig. 5.4.2 - The 7- pixel cliques in the neighborhood system

Tjelmeland defines the potential functions (the V_c 's in the Hammersley-Clifford Theorem) based on the configuration that exists in a particular clique. Potential functions are defined to be equal to zero on all cliques except the 7-pixel cliques shown in Fig. 5.4.2. Fig. 5.4.3 shows all the configurations specified for the case where we wish to distribute only two facies, facies A and facies B and the center pixel in the clique is occupied by facies A. Configuration 1 refers to the case where all pixels are occupied by facies A. Configuration 2 represents the case where any one non-central pixel in the clique is occupied by facies B and all other pixels are occupied by facies A. Configurations are assumed to be rotationally invariant so the location of the white pixel is immaterial as long as it does not occupy the center pixel.

Fig. 5.4.4 shows all possible equivalent alternatives for configuration 2 through 6. Again, we emphasize that Fig. 5.4.3 shows the 7 possible configurations where the center pixel is occupied by facies A. There also exist seven configurations with facies B occupying the center pixel. These can be obtained from Fig. 5.4.3 simply by interchanging facies A and facies B, i.e., replacing all white pixels by shaded pixels and all shaded pixels by white pixels. Thus, there are actually 14 specified possible configurations.

Returning to the general case and following Tjelmeland, we let x_c denote the facies configuration on clique c , i.e., x_c gives the set of facies occupying the pixels of clique c . We define $V_c(x_c) = 0$ unless c is one of the seven pixel cliques shown in Fig. 5.4.2. On each of these seven pixel cliques, $V_c(x_c)$ is defined given by

$$V_c(x_c) = \begin{cases} \theta_l & \text{if } x_c \in D_l \quad l = 1, 2, \dots, N_c \\ 0 & \text{otherwise} \end{cases}, \quad (5.4.1)$$

where D_ℓ , $\ell = 1, 2, \dots, N_c$, are the predefined configuration sets (see Fig. 5.4.3 and Fig. 5.4.4 for the two facies case), and N_c denotes the number of configurations and the θ_ℓ 's represent values which define the potential functions. In Eq. 5.4.1, x_c represents the set of facies that occupy the pixels of clique c .

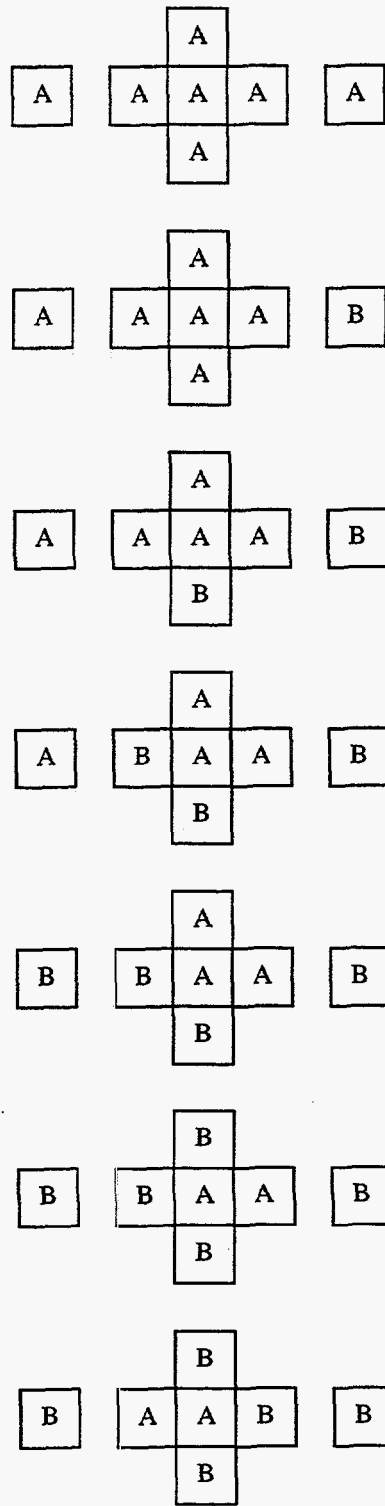
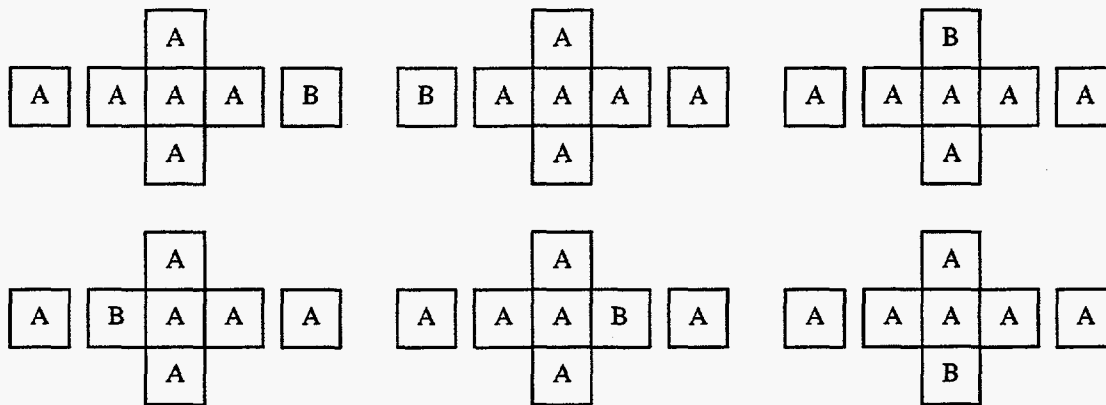


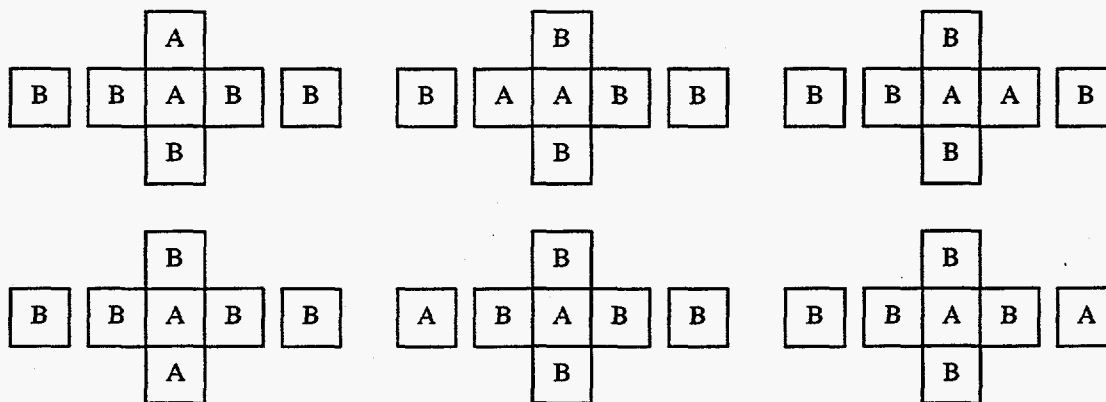
Fig. 5.4.3 - The 7 configurations specified with facies A in the center pixel.

The values specified for the θ_i 's control both the structure of any realization obtained as well as the expected value of the relative number of pixels occupied by each facies. However, how to choose the θ_i 's to control the structure of any realization or the expected number of pixels occupied by each facies is not an easy problem. As discussed later, it appears that the best approach is to try to specify the desirable number of each configuration that we wish to appear in any realization.



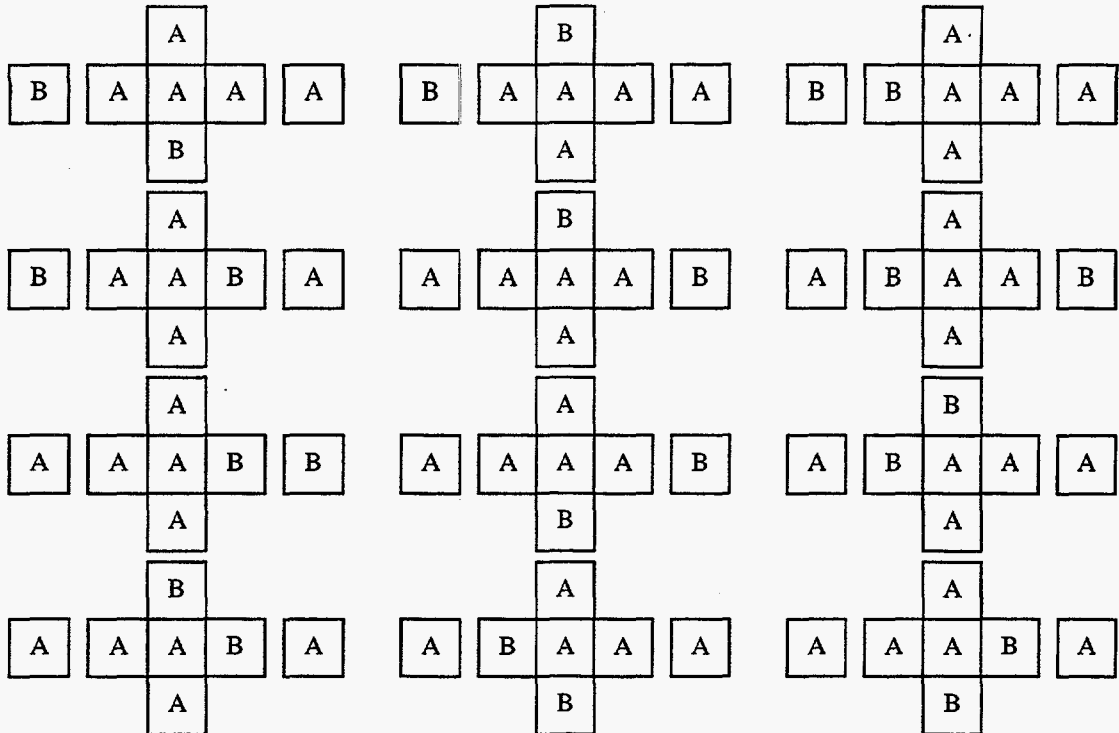
(a) Configuration 2 and its alternatives.

- Features: 1. There are 5 pixels with the same value as the central one.
 2. One pair of the opposite pixels is different and other two are similar.



(b) Configuration 6 and its alternatives.

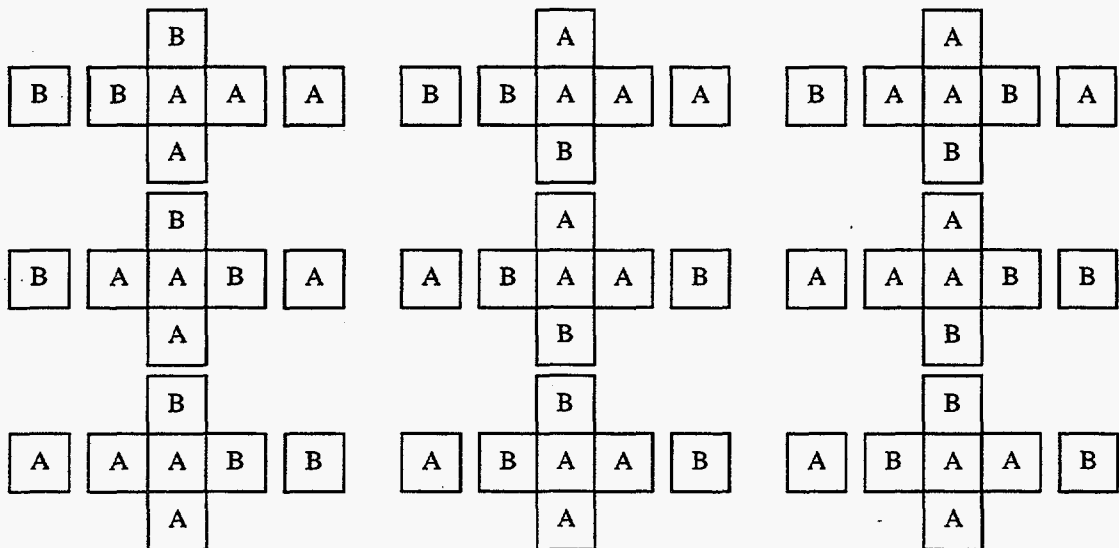
- Features: One of the pixels is similar to the central pixel and all the others are not.



(c) Configuration 3 and its alternatives.

Features: 1. Four pixels are similar to the central one.

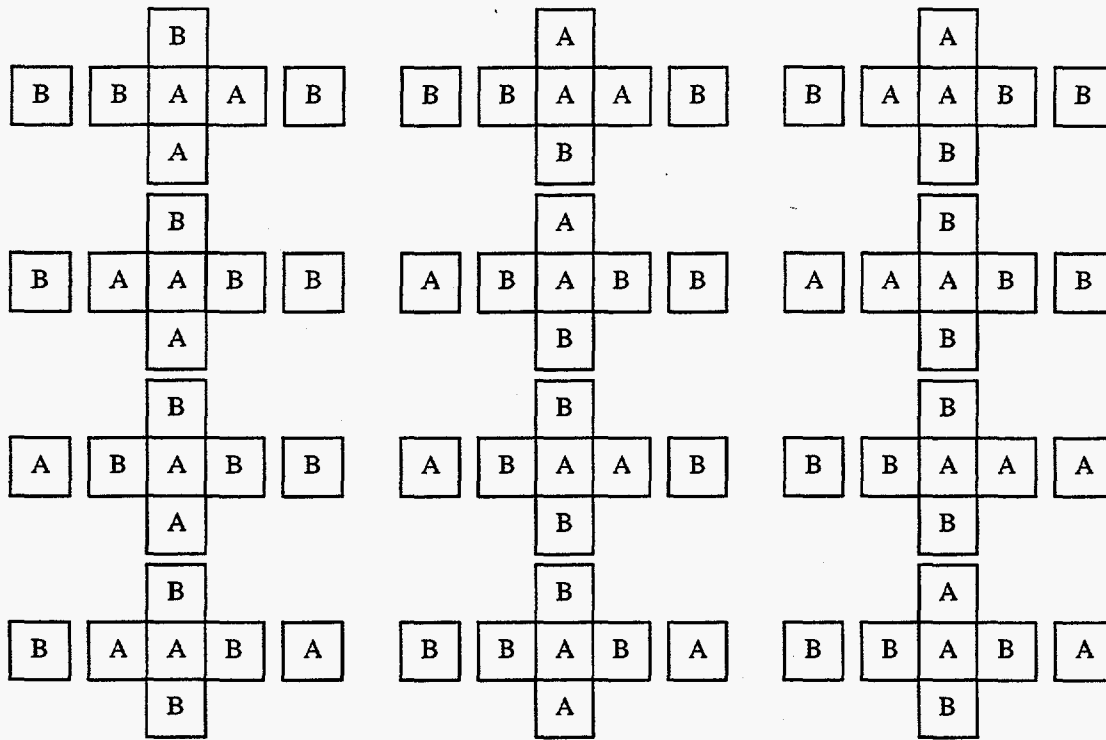
2. One pair of opposite pixels are the same as central pixel and other two pairs are not.



(d) Configuration 4 and its alternatives.

Features: 1. All the opposite pixels are different.

2. Three pixels are similar to the middle one.



(e) Configuration 4 and its alternatives.

Features: One pair of the opposite pixels is different from the central one.
 Other two pairs are both different but one pixel in each pair is similar to the middle pixel.

Fig. 5.4.4 – Clique configurations and alternatives.

Boundary treatment

A pixel adjacent to the boundary or one pixel away from the boundary has missing neighbors as discussed before. In stead of using a free boundary treatment, we work with an extended lattice, $\{-1:N_x+2, -1:N_y+2, -1:N_z+2\}$, rather than the real lattice of interest, $\{1:N_x, 1:N_y, 1:N_z\}$, where N_x , N_y and N_z are the numbers of pixels in x, y and z directions respectively. The guard region is $\{-1:0, -1:0, -1:0\}$ and $\{N_x+1:N_x+2, N_y+1:N_y+2, N_z+1:N_z+2\}$. The extended lattice is initialized by uniformly sampling from the K facies. But when doing a simulation, we only perturb the pixels in the real lattice of interest and all the facies values in the extra cells remain unchanged. This approach makes the development of the source code very easy. The disadvantage of this method is the extra storage required due to the increase in the lattice dimensions. More importantly, adding two rows of pixels is not sufficient to eliminate edge effects.

Two facies examples

Here, the procedure is applied to generate three-dimensional realizations of facies distributions in the case where only two facies (facies A and facies B), are present ($K =$

2). For this case, the fourteen configurations and all the equivalent alternatives are shown in Fig. 5.4.3 and Fig. 5.4.4. The values of the θ_ℓ 's are specified as

$$\begin{aligned}(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) &= (\theta_8, \theta_9, \theta_{10}, \theta_{11}, \theta_{12}, \theta_{13}, \theta_{14}) \\ &= (-2.0, -2.0, -1.2, -1.2, 1.0, 1.5, 5.0).\end{aligned}$$

Note that decreasing the values of some θ_ℓ with all other θ 's kept fixed will increase the probability of configuration ℓ ($\ell = 1, 2, \dots, N_c$). Thus, in this case, the most probable configurations are configurations 1 and 2, 8 and 9. Also note that since the clique potential with facies A in the center pixel are identical to those with facies B in the center, i.e., $\theta_i = \theta_{i+7}$, $i=1, 2, \dots, 7$, thus the expected value of the fraction of pixels occupied by each facies is 1/2. As before, in all simulation results, one iteration refers to proposing n perturbations where n is the number of pixels. For this example, we work with a $40 \times 40 \times 40$ lattice, i.e., $n = 64000$. Fig. 5.4.5 shows a realization of the facies distribution after 1,000 iterations. Qualitatively, similar realizations were obtained for all $L \geq 200$, where L denotes the number of iterations. This provides an indication that after 200 iterations, we begin sampling the probability function for the facies distribution correctly, i.e., it requires roughly 200 iterations to pass through the transient period of the Markov chain. Figs. 5.4.5 and 5.4.6 show realizations obtained after the 500 and 1000 iterations. Note that Figs. 5.4.5 and 5.4.6 display qualitatively similar structures but the specific pixels occupied by facies A are quite different in the two cases. In Figs. 5.4.5 and 5.4.6, resolutions in the coordinate directions refer to the size of the pixels, i.e., all pixels have dimensions $\Delta x, \Delta y, \Delta z$, where $\Delta x = 2\Delta y = 4\Delta z$. For the realization of Fig. 5.4.5, 51.02% of the pixels are occupied by facies A whereas in the realization of Fig. 5.4.6, 50.29% of the pixels are occupied by facies A.

The penalty term was not applied for the above example and the three facies example in the next section.

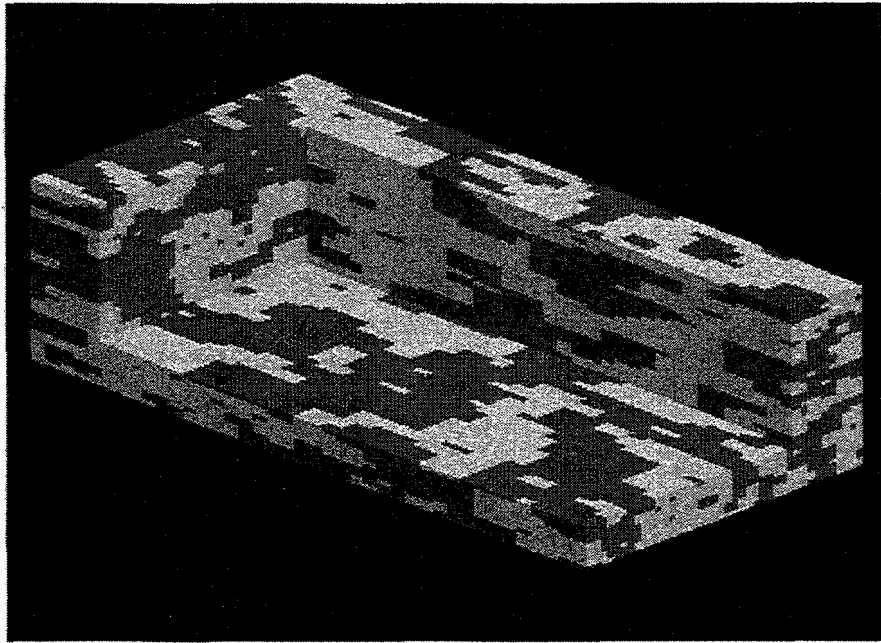


Fig. 5.4.5- A realization of 3D, two facies model, $40 \times 40 \times 40$ lattice, 500 iterations, sections cut at $x=y=10$, $z=5$, the resolutions in x , y and z directions are 1, 2, 4.

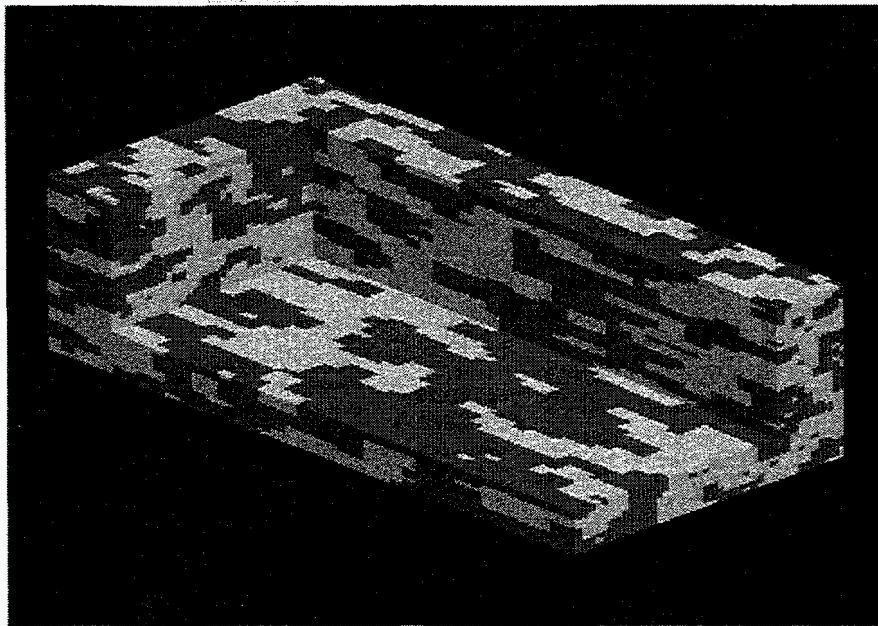


Fig. 5.4.6 - A realization of 3D, two facies model, $40 \times 40 \times 40$ lattice, 1000 iterations, sections cut at $x=y=10$, $z=5$, the resolutions in x , y and z directions are 1, 2, 4.

Three facies case

In the three facies case, we label the facies as A, B and C. In this case, we split configurations 3 and 4 in Fig. 5.4.3 into two configurations, one for cases where only two facies are present in the configuration and another for the cases where three facies are present. We denote the separated configurations by configuration 3 and 8 (these replace configuration 3 in Fig. 5.4.3), and configurations 4 and 9. Therefore, we have 27 basic configurations, nine of them have facies A in the center pixel, nine have facies B occupying the center pixel, finally, nine with facies C occupying the center pixel. In these configurations, there can be only two facies present in configurations 1-7, 10-16, 19-25. These two facies could be A and B, A and C, or B and C, depending on the facies in the center pixel. The other six configurations, 8 and 9, 17 and 18, and 26 and 27 have three facies present. Fig. 5.4.7 (which appear after Fig. 5.4.8) shows the nine configurations pertaining to the case where facies A occupies the central pixel. Only one possible case is shown in configuration 8 and 9, but there are several possible combinations for cases where three facies are present. For example, suppose facies A is in the center pixel of configuration 8, then both configurations shown in Fig. 5.4.8 are considered to be equivalent configurations and are give rise to the same value of θ_8 .

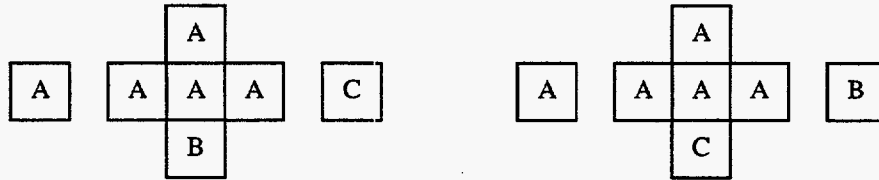


Fig. 5.4.8 - Configuration 8 with three facies present and facies A in the center pixel

We label the θ 's for these configurations as θ_j^A , θ_j^B and θ_j^C , $j = 1, 2, \dots, 9$, where the superscripts refer to the facies occupying the center pixel. As in Fig. 5.4.3, these configurations are assumed to be independent of rotation. Thus, there exist equivalent alternatives as in Fig. 5.4.4.

If x_c represents the facies distribution on a 7-pixel clique with facies D ($D = A, B$ or C) in the center pixel, and x_c is the configuration corresponding to θ_j^D , then the potential function is defined as

$$V_c(x_c) = \theta_j^D + \beta_j^l(D, D') + \beta_j^u(D, D''), \quad (5.4.2)$$

where D' denotes the facies located below the center pixel (in the z-direction) and D'' denotes the facies located above the center pixel (in the z-direction). The β_j^l 's and β_j^u 's are used to impose an ordering on the facies, e.g., facies B is above facies C and facies A is below facies C. If no such ordering is desired we need to set

$$\beta_j^l(D, D') = \beta_j^u(D, D'') = 0. \quad (5.4.3)$$

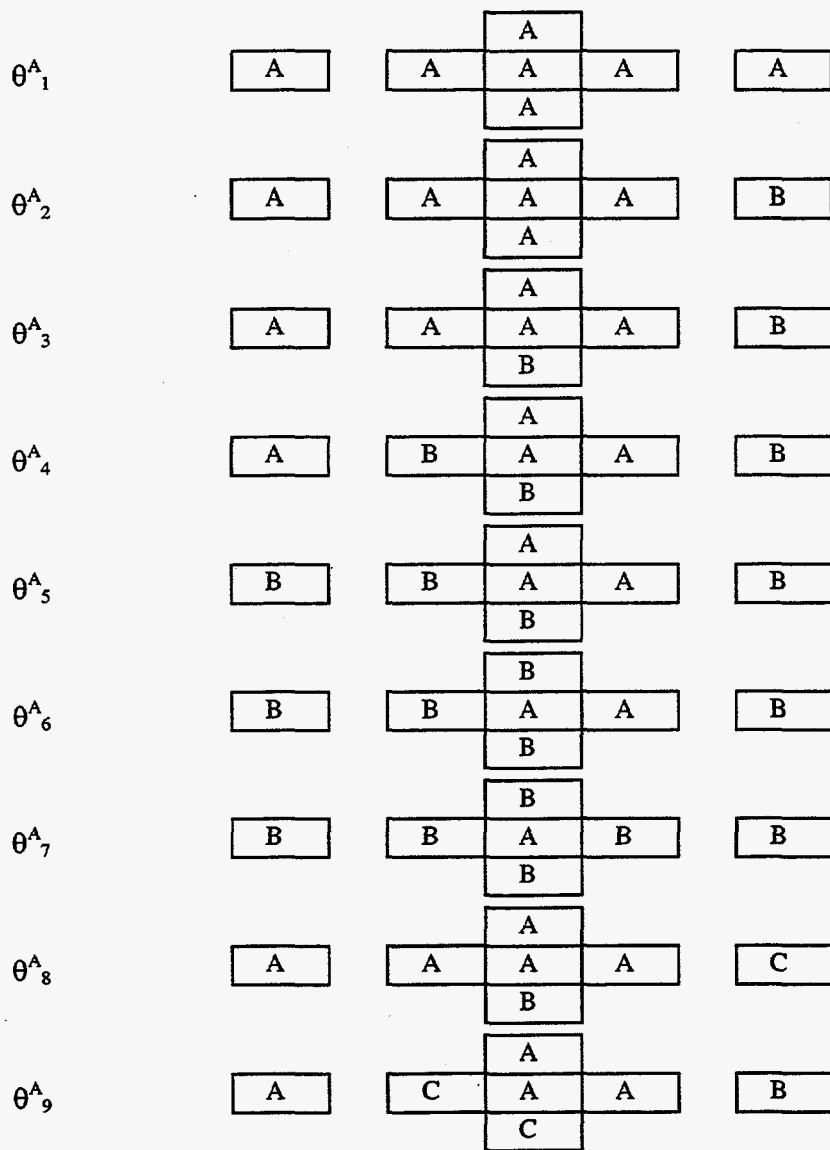


Fig. 5.4.7 - The specified 9 configurations with facies A in the center pixel

Note the β_j^l 's and β_j^u 's control ordering in the z-direction, but we could use similar terms to control the ordering in the x and y directions. Unlike Tjelmeland [55], we require that the β_j^l 's and β_j^u 's, depend on the overall configuration, not just on the facies in the center pixel and the ones above and below it.

Three facies examples

In this example, the θ 's and β 's are defined below.

If facies A is in the center pixel, then the values of the θ 's on the nine configurations are given by

$$\theta^A = (\theta_1^A, \theta_2^A, \dots, \theta_9^A) = (-2.0, -1.98, -1.6, -1.2, 1.0, 1.5, 5.0, -1.8, -1.6). \quad (5.4.4)$$

Similarly

$$\theta^B = (\theta_1^B, \theta_2^B, \dots, \theta_9^B) = \theta^A, \quad (5.4.5)$$

and also

$$\theta^C = (\theta_1^C, \theta_2^C, \dots, \theta_9^C) = (-1.9, -1.98, -1.8, -1.6, 1.0, 1.5, 5.0, -1.8, -1.6) \quad (5.4.6)$$

where the θ 's without subscripts are vectors.

The vectors $\beta_l(D, D')$ and $\beta_u(D, D'')$ are defined by

$$\begin{aligned} \beta_l(D, D') &= (\beta_1^l(D, D'), \beta_2^l(D, D'), \dots, \beta_9^l(D, D')) \\ \beta_u(D, D'') &= (\beta_1^u(D, D''), \beta_2^u(D, D''), \dots, \beta_9^u(D, D'')). \end{aligned} \quad (5.4.7)$$

In this example, we specify

$$\begin{aligned} \beta_j^l(A, A) &= \beta_j^l(A, B) = 0.0, \quad \beta_j^l(B, A) = \beta_j^l(B, B) = 0.0, \quad \beta_j^l(C, C) = 0.0, \quad j = 1, 2, \dots, 9, \\ \beta_1^l(A, C) &= \beta_1^l(B, C) = \beta_1^l(C, A) = \beta_1^l(C, B) = 0.0, \\ \beta_j^l(A, C) &= 1.0, \quad \beta_j^l(B, C) = -0.1, \quad \beta_j^l(C, A) = -0.1, \quad \beta_j^l(C, B) = 1.0, \quad j = 2, 3, \dots, 9. \end{aligned}$$

Similarly

$$\begin{aligned} \beta_j^u(A, A) &= \beta_j^u(A, B) = 0., \quad \beta_j^u(B, A) = \beta_j^u(B, B) = 0., \quad \beta_j^u(C, C) = 0., \quad j = 1, 2, \dots, 9, \\ \beta_1^u(A, C) &= \beta_1^u(B, C) = \beta_1^u(C, A) = \beta_1^u(C, B) = 0.0, \\ \beta_j^u(A, C) &= -0.1, \quad \beta_j^u(B, C) = 1., \quad \beta_j^u(C, A) = 1., \quad \beta_j^u(C, B) = -0.1, \quad j = 2, 3, \dots, 9. \end{aligned}$$

These β values were chosen to make it highly probable that facies C will occur below facies B and above facies A.

In the example under consideration, we again use $40 \times 40 \times 40$ pixels, i.e., 64,000 pixels. The x and y dimensions of the pixels are four times the z dimension. This increases the continuity in the x and y directions. The number of each facies contained in a state in the Markov chain stabilizes at about 2000 iterations, an indication (but not a guarantee) that the Markov chain has converged to the stationary distribution. Table 5.4.1 shows the percentage of each facies contained in the initial distribution obtained from the prior model with $T_1 = T_2 = T_3 = 1/3$ and the realizations obtained after 100, 300, 500, 1000, 2000, 3000 and 4000 iterations. Note that from iteration 300 onward, realizations contain

about 41% of facies A, 41.5% of facies B and 17.5 % of facies C. To perform 4000 iterations requires about 7 cpu-hours on a Pentium-133.

Table 5.4.1. Proportions of facies (percentage)

	The number of interations							
	0	100	300	500	1000	2000	3000	4000
Facies A	33.25	40.59	41.62	42.49	42.61	40.73	39.28	40.27
Facies B	33.24	40.68	41.05	40.64	40.82	41.99	43.19	41.51
Facies C	33.31	18.73	17.33	16.87	16.57	17.28	17.53	18.23

Fig. 5.4.9 shows four x-z cross-sections of realizations obtained after 3,000 iterations. Fig. 5.4.10 shows cross-sections cut at the same locations as in Fig. 5.4.9 corresponding to the realization obtained after 4,000 iterations. Although the general structures in the scenes after 3,000 and 4,000 iterations are similar, the two realizations are quite different.

As we can see from the last two examples, different types of structures, particularly very complicated local structures, can be obtained by the model that Tjelmeland used. Since the clique configurations are considered rotation free, the number of parameters is even less than the number of parameters in 2D model with the same number of facies type. But there are a few problems in this model. First of all, the use of rotation-invariant clique configurations is not allowed in our applications due to the orientation requirement of facies distribution. For example, the following two 5-pixel clique configurations shown in Fig. 5.4.11 promote totally different orientations for facies A and facies B. configuration a encourages interface between A and B along northwest-southeast direction, while configuration b proposes interface between A and B along the southwest-northeast direction. However, in the model used by Tjelmeland, these two configurations are considered to be equivalent. Secondly, we suspect that imposing ordering via the introduction of extra parameters, i.e., β 's, causes the parameter estimation to be more difficult but we have not verified this. Moreover, in 3D, third-order neighborhood system and 7-pixel cliques, the free boundary treatment does not appear to be feasible. Therefore, we pursue a simpler 3D model here by using second-order neighborhood and only 2-pixel cliques.

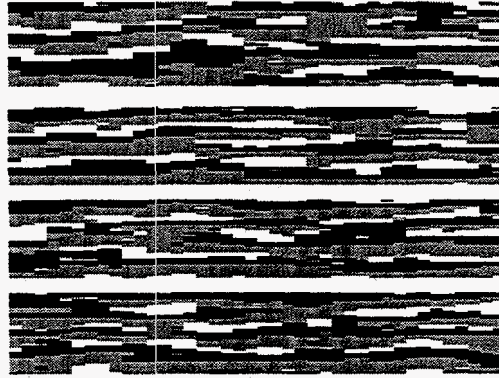


Fig. 5.4.9 – A realization of 3D three facies model, 3000 iterations, x-z cross-sections at 5, 10, 15, 20 (from top to bottom), (facies A: dark, facies B: grey, facies C: white).

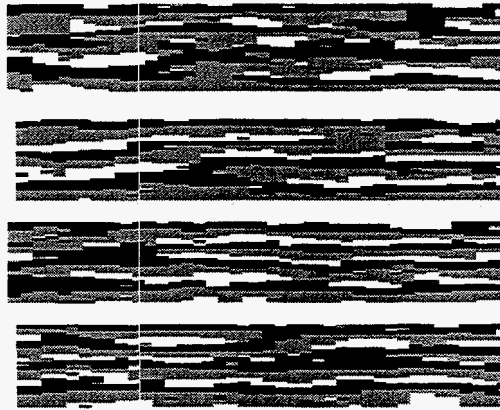


Fig. 5.4.10 – A realization of 3D three facies model, 4000 iterations, x-z cross-sections at 5, 10, 15, 20 (from top to bottom), parameters are the same as in Fig. 5.4.9, (facies A: dark, facies B: grey, facies C: white).

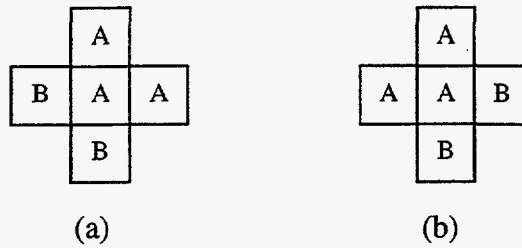


Fig. 5.4.11 – Examples of 5-pixel clique configurations

3D, second-order models

The 3D, second-order neighborhood system is shown in Fig. 5.4.12. There are nine types of 2-pixel cliques as shown in Fig. 5.4.13. In this example, we consider three facies labeled A, B and C. Just as in 2D, second-order model, we have 9 configurations for each type of 2-pixel clique, each with its own potential function. Thus, there are totally $9 \times 9 = 81$ model parameters.

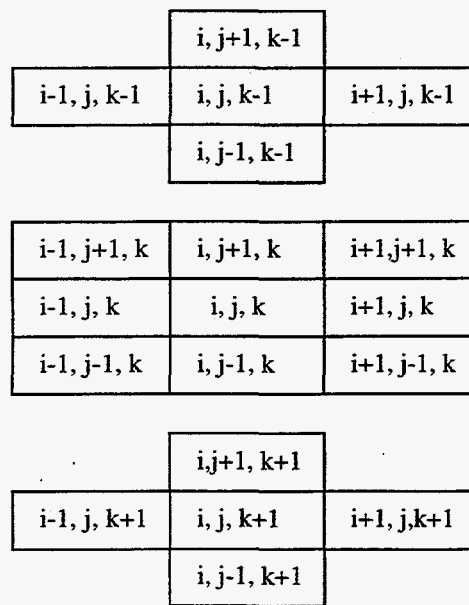


Fig. 5.4.12 - The cut layers of the three-dimensional 2nd-order neighborhood.

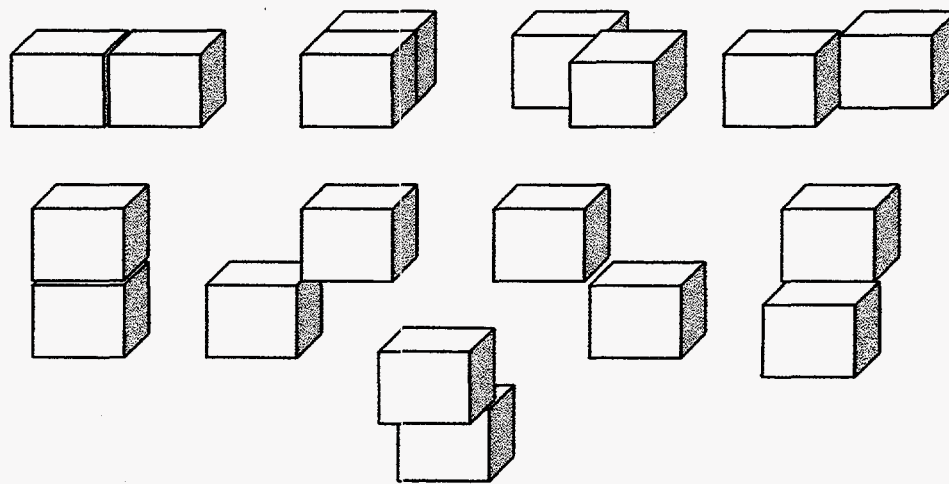


Fig. 5.4.13 – The 2-pixel cliques in the neighborhood system.

As an illustration, configurations for vertical 2-pixel cliques are shown in Fig. 5.4.14. We use purple to represent facies A, green for facies B and red for facies C.

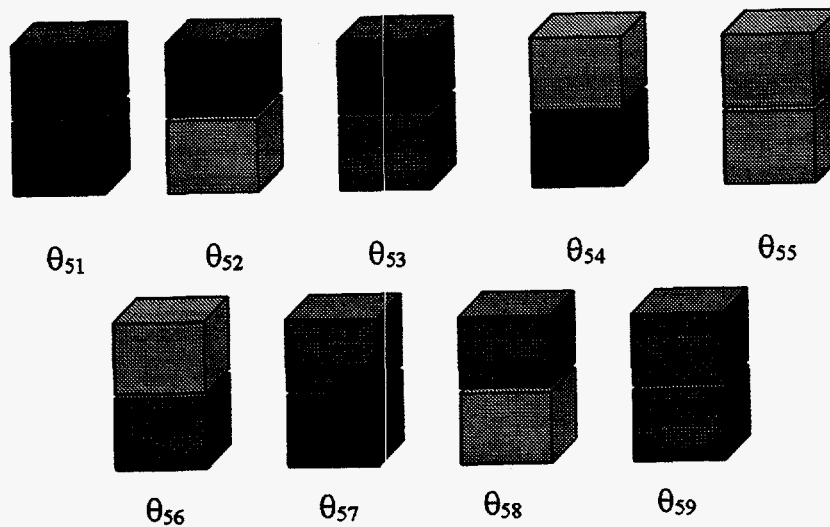


Fig. 5.4.14 – Vertical clique configurations, A-Purple, B-Green, C-Red.

In the examples, we work with $64 \times 64 \times 64$, i.e., 262,144 pixels. Following the conventions used in the previous section, we use the notations below for the parameters or the potentials of each type of 2-pixel clique.

$\theta_1=(\theta_{11}, \theta_{12}, \dots, \theta_{19})$, potential vector for 2-pixel cliques along x direction in x-y plane;
 $\theta_2=(\theta_{21}, \theta_{22}, \dots, \theta_{29})$, potential vector for 2-pixel cliques along y direction in x-y plane;
 $\theta_3=(\theta_{31}, \theta_{32}, \dots, \theta_{39})$, potential vector for cliques having 45° directionality in x-y plane;
 $\theta_4=(\theta_{41}, \theta_{42}, \dots, \theta_{49})$, potential vector for; cliques having 135° directionality in x-y plane;
 $\theta_5=(\theta_{51}, \theta_{52}, \dots, \theta_{59})$, potential vector for cliques in z-direction;
 $\theta_6=(\theta_{61}, \theta_{62}, \dots, \theta_{69})$, potential vector for cliques having 45° directionality in x-z plane;
 $\theta_7=(\theta_{71}, \theta_{72}, \dots, \theta_{79})$, potential vector for cliques having 135° directionality in x-z plane;
 $\theta_8=(\theta_{81}, \theta_{82}, \dots, \theta_{89})$, potential vector for cliques having 45° directionality in y-z plane;
 $\theta_9=(\theta_{91}, \theta_{92}, \dots, \theta_{99})$, potential vector for cliques having 135° directionality in y-z plane.

In the first application of this model, the preceding parameter vectors are defined as follows.

$$\theta_1=\theta_2=\theta_3=\theta_4=\theta_6=\theta_7=\theta_8=\theta_9= (-0.5, 0.5, 0.5, 0.5, -0.5, 0.5, 0.5, 0.5, -0.5)$$

$$\theta_5= (-0.5, -0.5, 0.5, 0.5, -0.5, -0.5, -0.5, 0.5, -0.5).$$

This choice promotes continuity of each facies in all directions of the 3D space, i.e., the potentials for all the cliques occupied by the same facies have similar negative values (-0.5) and most of the potentials controlling the transition between facies are all equal to 0.5 in order to discourage such transitions. However, we encourage transitions from A (purple) to B (green), from B (green) to C (red) and from C to A in the downward vertical direction (z direction). The results of Fig. 5.4.14 shows a realization of the model obtained at iteration 400 of the MCMC sampling procedure.

Fig. 5.4.14 (b) - A cut view of Fig. 5.4.14a, cut at $x=y=15, z=10$.

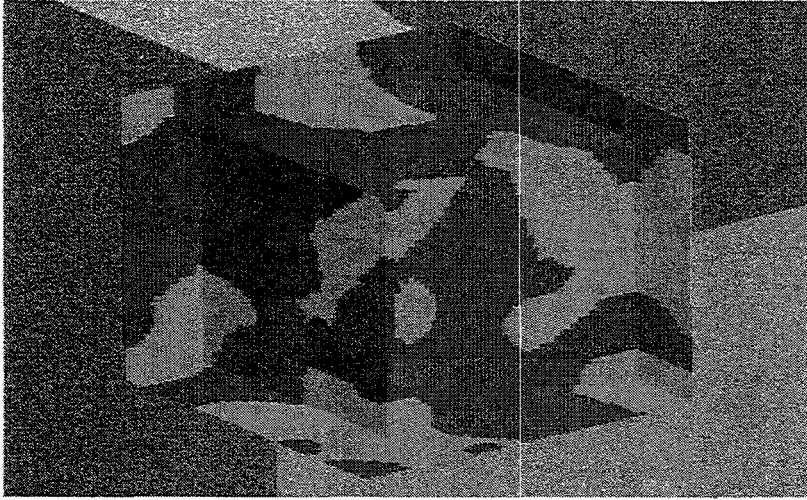


Fig. 5.4.14 (a) - A realization of the 3D, 2nd-order model, 400 iterations.



We can see from Fig. 5.4.14a that the overall vertical transition from facies A (purple) to facies B (green), from facies B (green) to facies C (red) and from facies C (red) to facies A (purple) is apparent, but there exist other types of transitions. The continuity of each facies in all directions is also very clear from the cut view of Fig. 5.4.14b although the scale of continuity varies somewhat from location to location. It should be pointed out that the strict penalty term was utilized in this case, i.e., in Eq. 5.2.9, all σ_k 's were set to 1.0 and all T_k 's were set equal to 1/3. Thus, in the realization of Fig. 5.4.14, each facies occupies almost exactly 1/3 of the pixels. Also note that visually, there are no boundary effects, i.e., no stray pixels appear along the boundaries.

In order to check whether we can impose other features, we conducted another experiment and tried to impose very strong ordering and orientation. We defined the potential vector for each type of cliques as follows.

$$\theta_1 = \theta_3 = \theta_4 = (-0.3, -0.5, 0.5, 0.5, -0.3, -0.5, -0.5, 0.5, -0.3)$$

$$\theta_5 = (-0.3, -0.1, 0.5, 0.5, -0.3, -0.1, -0.1, 0.5, -0.3)$$

$$\theta_6 = (-0.6, 0.5, 0.5, 0.5, -0.6, 0.5, 0.5, 0.5, -0.6)$$

$$\theta_2 = \theta_8 = \theta_9 = (-0.5, 0.5, 0.5, 0.5, -0.5, 0.5, 0.5, 0.5, -0.5)$$

$$\theta_7 = (-0.2, -0.5, 0.5, 0.5, -0.2, -0.5, -0.5, 0.5, -0.2).$$

With these values, we would expect to obtain continuity in the x-z plane with 45° directionality and y direction (potential=-0.6). We also expect to obtain more transitions from A to B, B to C, and C to A in four specific directions, both diagonal directions in the x-y plane, x-z plane with 135° directionality as well as the x direction of the x-y plane.

Fig. 5.4.15 shows a realization from this model. This realization was obtained by 500 iterations of MCMC sampling algorithm. Actually, a relatively long run was made (2000 iterations), but the basic structures of all the realizations obtained subsequent to the 400 iteration are similar. By inspection, we can see the realization displays the features we promoted by our choice of the statistical parameters. It is interesting to note that if we view this realization from another position as shown in Fig. 5.4.15b, it is pretty much like a layered reservoir model.

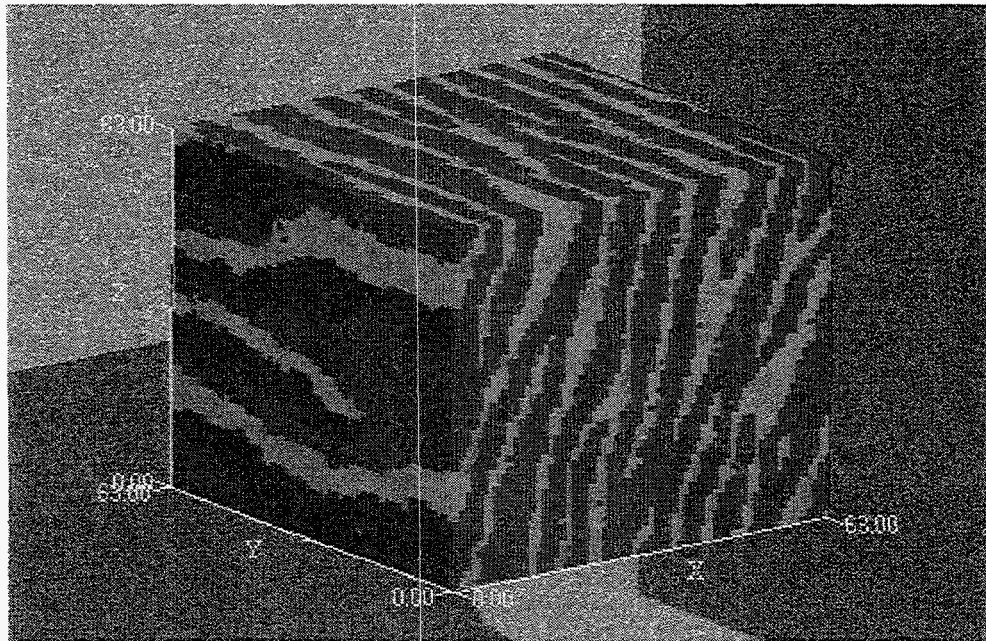


Fig. 5.4.15 (a) – A realization of the 3D model, 500 iterations.

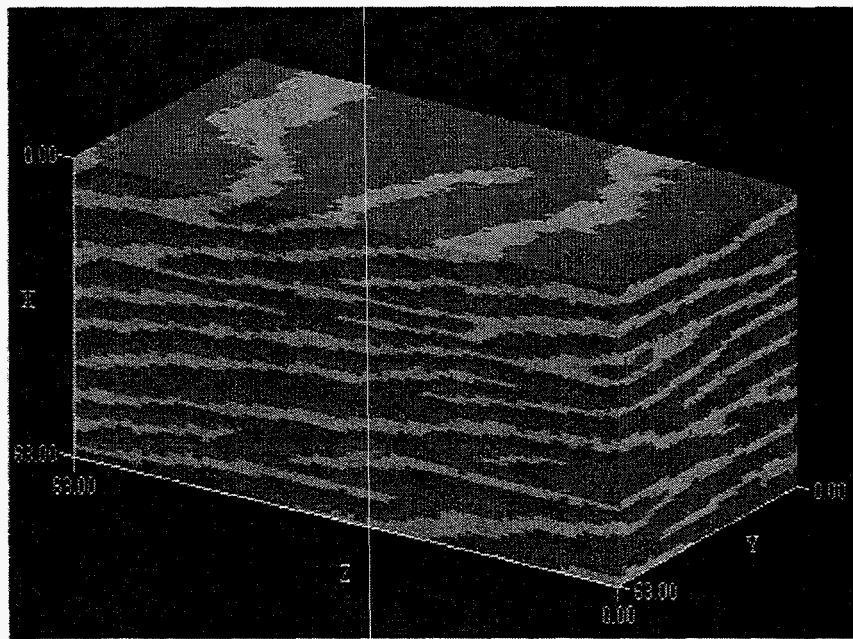


Fig. 5.4.15 (b) – View of Fig. 5.4.15a from another position, resolutions in x, y, z directions are 1, 1 and 2 respectively.

Model conditioning

In the preceding two examples, the strong penalty term was used to avoid phase transition problems although in fact, the penalty term for the second example has little influence on the realization obtained; the model is stable even if the penalty term is not used.

We now consider a model which does encounter a phase transition problem. We condition the model to the facies observed at 4 completely penetrating vertical wells. However, we actually do this by starting with no conditioning data and then adding conditioning data based on the facies we obtain at the pixels penetrated by wells after dozen iterations of the MCMC procedure. This is the same procedure used in some of the 2D model examples. By comparing the performance of the model before and after conditioning, we can understand how the conditioning affects the model. We still work with $64 \times 64 \times 64$ lattice. The areal locations of the chosen four wells are (10,10), (20,20), (32,32), (40,40). The MRF model is defined by the following parameter vectors:

$$\theta_1 = \theta_2 = \theta_6 = (-0.5, -0.5, -0.5, -0.5, -0.5, -0.5, -0.5, -0.5, -0.5)$$

$$\theta_5 = (-0.5, -0.5, 0.5, 0.5, -0.5, -0.5, -0.5, 0.5, -0.5)$$

$$\theta_8 = (-0.6, 0.5, 0.5, 0.5, -0.6, 0.5, 0.5, 0.5, -0.6)$$

$$\theta_3 = \theta_4 = \theta_9 = \theta_7 = (-0.5, -0.2, -0.2, -0.5, -0.5, -0.3, -0.3, -0.3, -0.5)$$

Fig. 5.4.16 represents the fraction of facies versus the number of iterations of MCMC algorithm with and without using well conditioning data. The solid curves pertain to the results obtained without conditioning data. Without conditioning data, it appears that after a large number of iterations of the MCMC method, all subsequent realizations generated will contain only facies B. Note that facies C disappear after 400 iterations. It can be seen that although the conditioning truly improves the model performance and stabilizes the fractions of facies A and B appearing in the realizations, facies C still disappears after 400 iterations. However, we believe this occurred because we selected the well conditioning data from the 100th iteration of the unconditioned run and by this iteration, only 10% of the pixels were occupied by facies C. Thus, the conditioning data may not have contained enough pixels occupied by facies C to stabilize the model. Nevertheless, this example demonstrates the effectiveness of conditioning on a 3D model. For example, the fractions of facies A and B at iteration 1000 are 0.273 and 0.727 respectively if no conditioning data are used, but with well conditioning data, the fractions of facies A and facies B are 0.521 and 0.479. Since the selection of model parameters imposes no preference on facies A and facies B, and the uniform prior probability distribution is used, the fractions of these two facies should be roughly the same in any legitimate realizations.

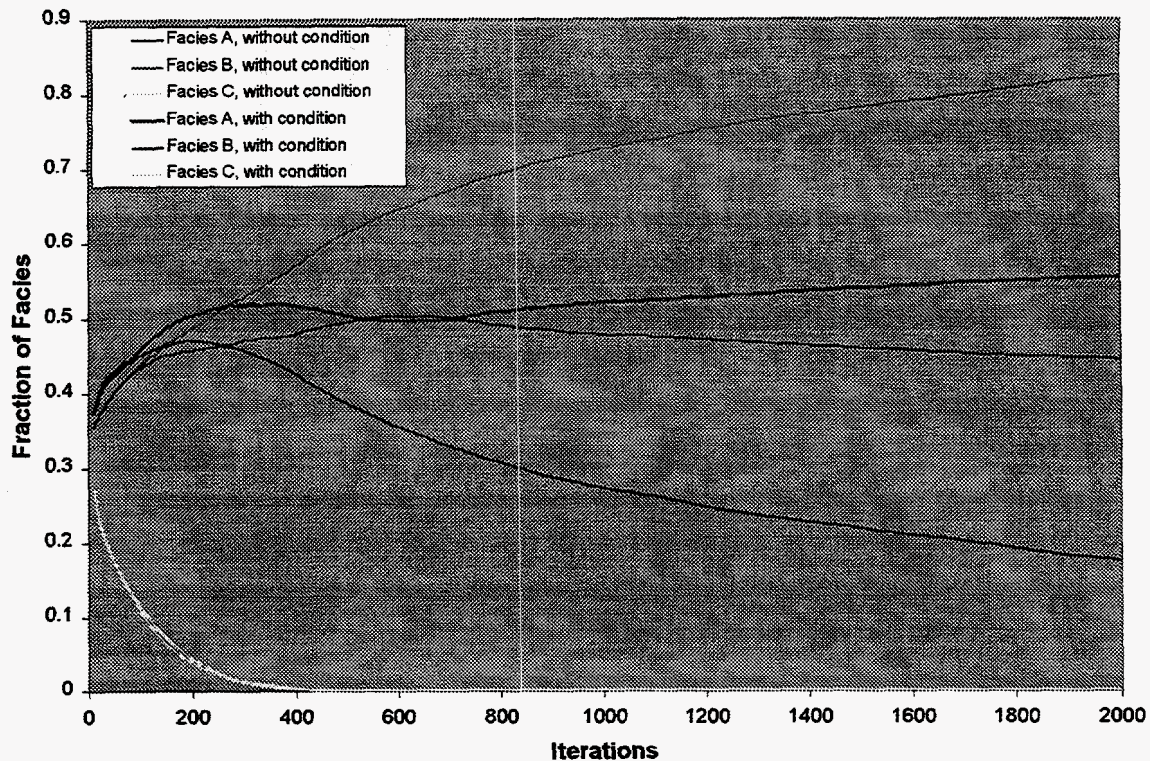


Fig. 5.4.16 – Fractions of facies versus the number of iterations, models with and without conditioning data.

ESTIMATION OF MODEL PARAMETERS

We can write the probability mass function of Eq. 5.2.7 as

$$P(x, \theta) = \frac{1}{Z(\theta)} h(x, \theta) = \frac{1}{Z(\theta)} \exp\{-\langle t(x), \theta \rangle\}, \quad (5.5.1)$$

where $P(x, \theta)$ is the probability mass function for the MRF model;

$h(x, \theta)$ is an exponential function referred to as the unnormalized probability function; $x = (x_1, x_2, \dots, x_n)$ denotes a realization of the random field X , with each element x_i representing the value of a random variable (facies value) corresponding to the i th pixel of the lattice S ;

θ is the vector of statistical parameters, i.e., $\theta = (\theta_1, \theta_2, \dots, \theta_{N_c})^T$, N_c here is the number of clique configurations specified;

$t(x)$ is called a canonical statistics vector, in our case, $t(x) = (n_1(x), n_2(x), \dots, n_{N_c}(x))^T$ and $n_j(x)$ is the number of cliques in x with configuration j ($j=1, 2, \dots, N_c$);

$\langle t(x), \theta \rangle$ denotes the inner product of the vectors t and θ ;

$Z(\theta)$ is the normalizing constant which is related to the parameter vector θ .

Given a realization (or an observation) of a MRF (e.g., a geological interpretation) and a basic MRF model, we would like to be able to estimate the values of the statistical parameters from the observation. Generally, there is only one realization available.

The standard way to obtain an estimate of model parameters, called statistical inference, is to apply Maximum Likelihood Estimation. The likelihood function in our case is simply the probability mass function corresponding to observation, which is denoted by y , defined as a function of the parameter vector θ , i.e.,

$$L(\theta) = \frac{1}{Z(\theta)} \exp\{-\langle t(y), \theta \rangle\} \quad (5.5.2)$$

$L(\theta)$ gives the likelihood of θ given the observed data y . Maximum likelihood estimation represents the process of determining the θ which maximizes $L(\theta)$. This problem is difficult because the normalizing constant, which is a function of theta, is unknown so the traditional statistical estimation methods are not applicable. Other methods for estimating MRF model parameters proposed in the literature, include the Stochastic Approximation (SA) procedure proposed by Younes [60], Monte Carlo Maximum Likelihood Estimation (MCMLE) by Geyer&Thompson [23,24,25], the Maximum Pseudo-Likelihood (MPL) by Beseg [1] and other *ad-hoc* approaches such as coding by Beseg [2], histogramming method by Derin and Elliott [14]. Because of the limitations of SA and MPL, MCMLE method has been extensively used and investigated in recent years. We consider the application of this approach to our problem. Because the basic computational tools for such problems involve optimization and solution of simultaneous nonlinear equations, we need to choose an appropriate optimization algorithm. Since maximum likelihood estimation is difficult unless a good initial estimate of the parameters is available, we have considered application of the coding and histogramming procedures to obtain an initial approximation of the parameter vector that maximizes the likelihood. However, for our problem neither of these methods is generally applicable or reliable. For other interesting papers about parameter estimation of MRF, we refer the reader to Refs. [7], [8], [20], [33], [41] and [42].

To consider a concrete example, we consider a two-dimensional problem where we define a third order neighborhood system as shown in Fig. 5.5.1a. Note the center pixel of this neighborhood is shaded and all pixels colored white represent the set of neighbors of the center pixel. Figure 5.5.1b shows all five pixel cliques that are contained within the neighborhood shown in Fig. 5.5.1a.

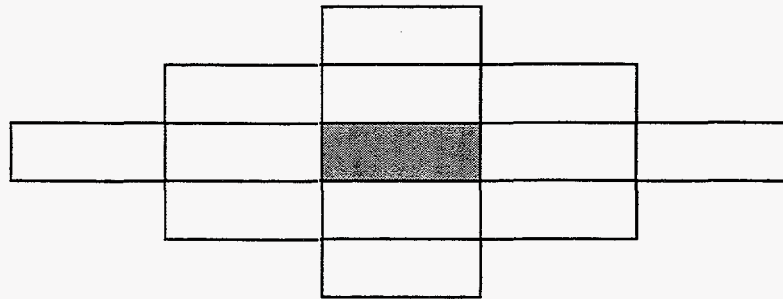


Fig. 5.5.1(a) – Third order neighborhood in 2D.

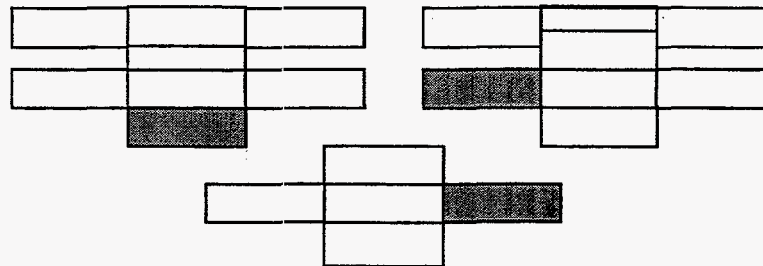


Fig. 5.5.1(b) – Five-pixel cliques for third order neighborhood.

In this example, we assume there are only two possible facies referred to as facies A and facies B. For any five-pixel clique, we consider 12 possible configurations for the arrangement of facies within the clique. Figure 5.5.2 shows the six possible configurations for the case where the center pixel of a five-pixel clique is occupied by facies A. If a pixel contains facies A, it is shaded or colored dark gray, whereas, if the pixel is occupied by facies B, it is shown as white.

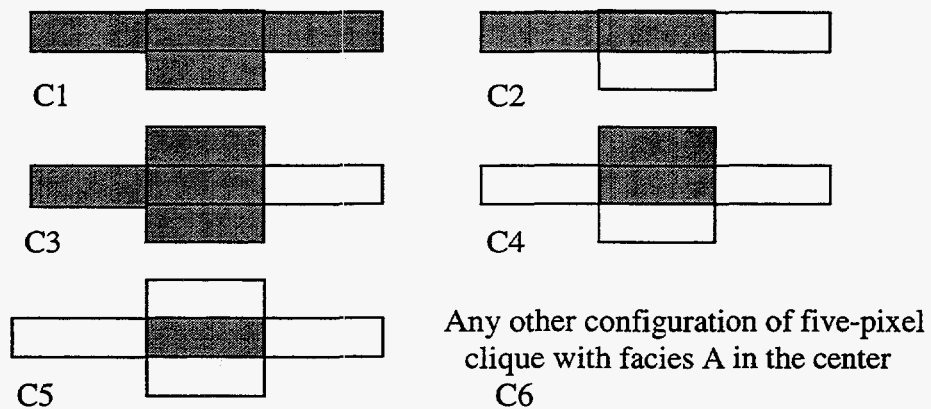


Fig. 5.5.2 – All configurations for a five-pixel clique with facies A in the center pixel of the clique.

One should note that there are actually four possible alternative configurations for the configurations labeled C2, C3, and C4. Fig. 5.5.3 shows the four possible configurations where the center pixel contains facies A and exactly one other pixel is occupied by facies A. These configurations are labeled A1, A2, A3 and A4 in Fig. 5.5.3. Note that if we define a clique potential such that configurations A1 and A3 are more probable than A2 and A4, then this will tend to promote continuity of facies A in the horizontal direction. In this example, we assume that configurations are rotationally invariant. This means that A1, A2, A3 and A4 are considered as equivalent configurations and the associated potential does not depend on which of these configurations actually exist. More generally, rotational invariance means that the four alternative configurations of C2 are considered equivalent, the four alternative configurations for C3 are considered equivalent and the four alternative configurations of C4 are considered equivalent. Similar to Fig. 5.5.2, there are 6 possible configurations for the case where the center pixel is occupied by facies B.

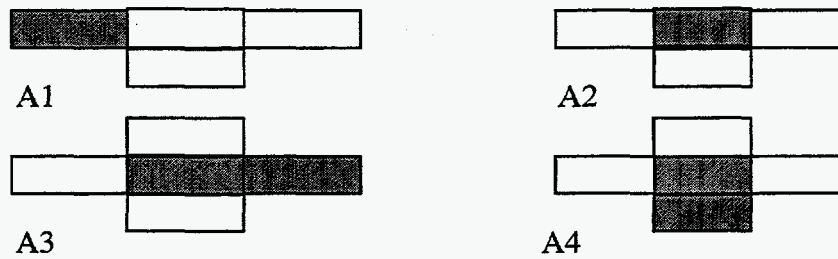


Fig. 5.5.3 – Alternative configurations for five pixel cliques, two pixels occupied by facies A with A in the center.

Now let D_ℓ , $\ell=1,2,\dots,N_c$, denote all possible configurations defined on five pixel cliques and let x be any realization of X , i.e., x defines the facies located in all pixels contained in S . Let x_c denote the facies distribution within any clique c , then the potential functions defined on any clique c are given by

$$V_c(x_c) = \begin{cases} \theta_l & \text{if } x_c \in D_l, l=1,2,\dots, N_c \\ 0 & \text{if } c \text{ is not a five pixel clique.} \end{cases} \quad (5.5.3)$$

The resulting probability function for the Markov random field can then be written as

$$P_\theta(x) = \frac{1}{Z} \exp \left[- \sum_{c \in C} V_c(x_c) \right] = \frac{1}{Z} \exp \left[- \sum_{l=1}^{N_c} n_l(x) \theta_l \right], \quad (5.5.4)$$

where n_l is the number of five-pixel cliques with configuration D_l and

$$\theta = (\theta_1, \theta_2, \dots, \theta_{N_c})^T. \quad (5.5.5)$$

Several comments are in order. In Eq. 5.5.4, the probability function is denoted by $P_\theta(x)$ to emphasize that the probability function depends on how the θ_i 's are defined in

Eq. 5.5.3. In essence, Eq. 5.5.4 gives a family of probability functions, one for each specification of the vector of statistical parameters, θ . It is appropriate to consider this family because our objective is to estimate the appropriate values of the components of θ directly from observations. Also note the normalizing constant, $Z=Z(\theta)$ depends on the definition of θ . Eq. 5.5.3 indicates that $V_c(x_c)$ is either equal to some θ_l or is zero. Moreover, V_c is zero unless c is a five pixel clique. Thus given x , if the set of all five pixel cliques includes exactly $n_l(x)$ cliques which contain configuration D_l , then the sum of all corresponding function V_c 's is $n_l(x)\theta_l$. Thus, the general expression (middle term of Eq. 5.5.4) can be represented by the final expression given in Eq. 5.5.4. If the vector of configuration numbers is defined by

$$n(x) = (n_1(x), n_2(x), \dots, n_{N_c}(x))^T, \quad (5.5.6)$$

where $\langle n(x), \theta \rangle$ denotes the inner product of $n(x)$ and θ , then Eq. 5.5.4 can also be written as

$$P_\theta(x) = \frac{1}{Z(\theta)} \exp[-\langle n(x), \theta \rangle]. \quad (5.5.7)$$

Given θ , Eq. 5.5.7, or equivalently Eq. 5.5.4, defines a probability function. However, if $x=y$ is an observation of the facies distribution (say from outcrop studies or geological interpretation), then Eq. 5.5.7 gives the likelihood function for θ , given the observation (data) y , and so we write Eq. 5.5.4 and 5.5.7 as

$$L(\theta) = \frac{1}{Z(\theta)} \exp\left[-\sum_{l=1}^{N_c} n_l(y)\theta_l\right] = \frac{1}{Z(\theta)} \exp[-\langle n(y), \theta \rangle]. \quad (5.5.8)$$

Given the data y , our objective is to find θ which maximizes the probability of obtaining the observation (sample) y . To do so, we maximize $L(\theta)$, or following Geyer and Thompson [23, 24], we minimize the negative log likelihood given by

$$\ell(\theta) = -\ln(L(\theta)) = \sum_{l=1}^{N_c} n_l(y)\theta_l + \ln(Z(\theta)). \quad (5.5.9)$$

Note there is no guarantee that $\ell(\theta)$ has a unique minimum. In fact, if θ is translated by a fixed constant vector, i.e., if θ is replaced by $\theta - \bar{\theta}$, then Eq. 5.5.8 becomes

$$L(\theta) = \frac{1}{b Z(\theta)} \exp\left[-\sum_{l=1}^{N_c} n_l(y)\theta_l\right], \quad (5.5.10)$$

where

$$b = \exp \left[- \sum_{l=1}^{N_c} n_l(y) \bar{\theta} \right], \quad (5.5.11)$$

and Eq. 5.5.9 becomes

$$\ell(\theta) = \sum_{l=1}^{N_c} n_l(y) \theta_l + \ln(Z(\theta)) + \ln(b). \quad (5.5.12)$$

If θ minimizes Eq. 5.5.9, then $\theta + \bar{\theta}$, minimizes Eq. 5.5.12. So it is clear that at best, θ can be estimated only up to an arbitrary constant translation. However, this fact does not cause difficulty. Since $P_\theta(x) = P_{\theta + \bar{\theta}}(x)$, realizations generated from either probability function will generate an equivalent sampling of X.

Our observation was denoted by y . More generally, if $p+1$ observations are available, they will be denoted by y_ℓ , $\ell=1,2,\dots,p$. Thus, it is convenient to denote a realization of X by u , i.e., u denotes any realization, whereas y_ℓ denotes a specific observation. Defining

$$h(u|\theta) = \exp \left[- \sum_{l=1}^{N_c} n_l(u) \theta_l \right] = \exp[-\langle n(u), \theta \rangle], \quad (5.5.13)$$

the probability function for a given θ can be written as

$$P_\theta(u) = \frac{1}{Z(\theta)} h(u|\theta). \quad (5.5.14)$$

Similarly, if $\theta = \psi$

$$P_\psi(u) = \frac{1}{Z(\psi)} h(u|\psi). \quad (5.5.15)$$

If Ω denotes the set of all realizations of X, then summing over all $u \in \Omega$ in Eq. 5.5.14 gives

$$\begin{aligned} 1 = \sum_{u \in \Omega} P_\theta(u) &= \frac{1}{Z(\theta)} \sum_{u \in \Omega} h(u|\theta) = \frac{1}{Z(\theta)} \sum_{u \in \Omega} \left[\frac{P_\psi(u)}{h(u|\psi)/Z(\psi)} \right] h(u|\theta) \\ &= \frac{Z(\psi)}{Z(\theta)} \sum_{u \in \Omega} \left[\frac{h(u|\theta)}{h(u|\psi)} \right] P_\psi(u). \end{aligned} \quad (5.5.16)$$

Note that the sum in the last term of Eq. 5.5.16 is the expected value of $h(u|\theta)/h(u|\psi)$ relative to the probability function $P_\psi(u)$, i.e.,

$$E_\psi \left[\frac{h(u|\theta)}{h(u|\psi)} \right] = \sum_{u \in \Omega} \left[\frac{h(u|\theta)}{h(u|\psi)} \right] P_\psi(u). \quad (5.5.17)$$

Using Eq. 5.5.17 in Eq. 5.5.16 and rearranging gives

$$\frac{Z(\theta)}{Z(\psi)} = E_{\psi} [h(u|\theta) / h(u|\psi)]. \quad (5.5.18)$$

The right side of Eq. 5.5.18 can be approximated by generating m samples, u_1, u_2, \dots, u_m , of X from the probability function $P_{\psi}(u)$; using this approximation, we have

$$\frac{Z(\theta)}{Z(\psi)} \approx \frac{1}{m} \sum_{j=1}^m \left[\frac{h(u_j|\theta)}{h(u_j|\psi)} \right]. \quad (5.5.19)$$

Using Eq. 5.5.19, the likelihood ratio $L(\theta)/L(\psi)$ can be approximated by

$$\frac{L(\psi)}{L(\theta)} = \frac{Z(\theta)h(u|\psi)}{Z(\psi)h(u|\theta)} \approx \left[\frac{1}{m} \sum_{j=1}^m \frac{h(u_j|\theta)}{h(u_j|\psi)} \right] \frac{h(u|\psi)}{h(u|\theta)}. \quad (5.5.20)$$

Using the definition of Eq. 5.5.13 to replace $h(u|\psi)$ and $h(u|\theta)$ in Eq. 5.5.20, and taking the natural logarithm of the resulting equation gives

$$f(\theta) = \ln \left\{ \frac{L(\psi)}{L(\theta)} \right\} = \ln \left[\frac{1}{m} \sum_{j=1}^m \frac{h(u_j|\theta)}{h(u_j|\psi)} \right] + \sum_{l=1}^{N_c} n_l(y)(\theta_l - \psi_l), \quad (5.5.21)$$

where the first equality in Eq. 5.5.21 serves to define $f(\theta)$. From Eq. 5.5.13, we have

$$\frac{h(u_j|\theta)}{h(u_j|\psi)} = \sum_{j=1}^m \exp \left[\langle n(u_j), \psi - \theta \rangle \right]. \quad (5.5.22)$$

It follows that Eq. 5.5.21 can be written as

$$f(\theta) = \ln \left[\frac{1}{m} \sum_{j=1}^m \exp \left(\sum_{l=1}^{N_c} n_l(u_j)(\psi_l - \theta_l) \right) \right] + \sum_{l=1}^{N_c} n_l(y)(\theta_l - \psi_l). \quad (5.5.23)$$

If ψ is fixed and $u_j, j=1,2,\dots,m$, are specific samples of X based on the probability function $P_{\psi}(u)$, then minimizing $f(\theta)$, when m is large, is equivalent to minimizing $-\ln[L(\theta)]$. Note that the l th component of the gradient of $f(\theta)$ is given by

$$\frac{\partial f}{\partial \theta_l} = n_l(y) - \sum_{j=1}^m n_l(u_j)w(j) \quad \text{for } l=1,2,\dots,N_c, \quad (5.5.24)$$

where

$$w(j) = \frac{\exp \left[\sum_{l=1}^{N_c} n_l(u_j)(\psi_l - \theta_l) \right]}{\sum_{j=1}^m \left\{ \exp \left[\sum_{l=1}^{N_c} n_l(u_j)(\psi_l - \theta_l) \right] \right\}}. \quad (5.5.25)$$

In our work, Eq. 5.5.23 is minimized by using the GBB algorithm presented by Raydan [43]. This algorithm can be thought of as a modified steepest descent algorithm which contains an automatic procedure for determining the size of the step to be taken in the direction of the negative gradient of $f(\theta)$. The algorithm was designed to be more robust for noisy data by allowing a slight increase in the objective function at any iteration.

As can be observed from Eqs. 5.5.23-25, if at any iteration of the GBB algorithm, $|\theta_i - \psi_i|$ becomes "large", we encounter overflow or underflow problem in evaluating $f(\theta)$ and its gradient. A procedure for avoiding this problem is to use a constrained minimization process. The resulting procedure for constructing the maximum likelihood estimate denoted by $\hat{\theta}$ from this point on is now presented in algorithm form.

Let $\psi = \theta^0$ denote an initial guess for $\hat{\theta}$;

1) Generate m samples of X , u_1, u_2, \dots, u_m , from the Markov chain Monte Carlo algorithm described before. These samples are generated using the probability function $P_{\theta^0}(u)$.

2) Minimize $f(\theta)$ in the region $N(\theta^0) = \{\theta \mid |\theta_i - \theta_i^0| \leq \delta\}$, using the GBB algorithm.

We let θ^c denote the θ vector which minimizes $f(\theta)$ in $N(\theta^0)$.

3) Check for convergence. For convergence to the maximum likelihood estimate, we require that the gradient of $f(\theta)$ at θ^c be close to zero, i.e., $\|\nabla f(\theta^c)\|_2 \leq \varepsilon$, where we typically use $\varepsilon < 10$. If the convergence condition is satisfied, we let $\hat{\theta} = \theta^c$ and the algorithm is complete. If not, let $\theta^0 = \theta^c$ and return to step 2).

As the computing cost of generating m samples of X in step (ii) is high, it is desirable to minimize the number of times we need to update the probability function P_{θ^0} and resample. Intuitively, if we can generate a good initial guess of $\hat{\theta}$, we might expect that only one, or at least only a very few, updates of θ^0 would be required. A procedure for generating an initial guess is described later.

Example 1

This is a synthetic two-dimensional, two-facies example where an observation was generated by sampling $P_{\theta}(x)$ using Markov chain Monte Carlo simulation. In this example, a two-dimensional region V is used and a 40×40 grid is used to partition S into 1600 pixels. Here θ is defined based on the neighborhood system and clique configurations defined previously, see Figs. 5.5.1 and 5.5.2. The values of θ (see Eq. 5.5.3) are specified by

$$\begin{aligned} \theta_1 = -0.7, \theta_2 = -0.5, \theta_3 = -0.2, \theta_4 = 1.0, \theta_5 = 4.0, \theta_6 = 5.0, \\ \theta_7 = -0.7, \theta_8 = -0.5, \theta_9 = -0.2, \theta_{10} = 1.0, \theta_{11} = 4.0, \theta_{12} = 5.0 \end{aligned} \quad (5.5.26)$$

Note θ_1 through θ_6 refer to five-pixel clique configurations where the center pixel is occupied by facies A and θ_7 through θ_{12} refer to the analogous five-pixel clique configurations where the center pixel is occupied by facies B.

Our objective is to use an observation of X to estimate θ^c . To illustrate our procedure, we sample $P_\theta(x)$ to construct a synthetic observation, y . From this point on, we assume that θ is unknown, and we wish to calculate the values of these statistical parameters by generating the maximum likelihood estimate, $\hat{\theta}$, based on the observation y and then set $\theta_l = \hat{\theta}_l, l = 1, 2, \dots, 12$ to obtain the probability function. The initial guess of $\hat{\theta}$, i.e., θ^0 , is shown in the third column of Table 5.5.1a (for this same example, we observed that if we did not constrain the GBB algorithm, the overall procedure did not converge). In step (ii) of the algorithm, we set $\delta = 0.05$. In sampling $P_{\theta^0}(x)$ in step (ii), we set $m=1000$. The last column of Table 5.5.1a gives the estimate \mathcal{E} of $\hat{\theta}$ obtained after one iteration of the algorithm. The table gives the number of GBB iterations required in step (ii) of the algorithm, the value of objective function ($f(\theta^c)$) and the two-norm of the gradient at θ^c . Table 5.5.1b presents a comparison of statistics obtained from the results of Table 5.5.1a. Column 2 of Table 5.5.1b gives the number of each of the 12 five-pixel configurations contained in the observation y . The third column gives the weighted average of the number of each configuration, i.e.,

$$\sum_{j=1}^m n_l(u_j) w(j) = \sum_{j=1}^m n_l(u_j) \frac{\exp \left[\sum_{l=1}^{N_c} n_l(u_j) (\psi_l - \theta_l^c) \right]}{\sum_{j=1}^m \left\{ \exp \left[\sum_{l=1}^{N_c} n_l(u_j) (\psi_l - \theta_l^c) \right] \right\}}, \quad (5.5.27)$$

where θ^c is the estimate of $\hat{\theta}$ obtained after the first iteration.

Table 5.5.2a and 5.5.2b show the results obtained after two iterations of the algorithm.

Table 5.5.1a – Results after 1 iteration of the algorithm

Parameter	True value	Initial guess	Estimate
θ_1	-0.7	-0.55	-0.597
θ_2	-0.5	-0.4	-0.383
θ_3	-0.2	-0.2	-0.172
θ_4	1.0	1.0	1.005
θ_5	4.0	4.0	3.999
θ_6	5.0	5.0	5.000
θ_7	-0.7	-0.55	-0.600
θ_8	-0.5	-0.4	-0.386
θ_9	-0.2	-0.2	-0.171
θ_{10}	1.0	1.0	1.005
θ_{11}	4.0	4.0	4.000
θ_{12}	5.0	5.0	5.000
# of GBB iterations	2		
Value of obj. function	-10.277		
Norm of gradient	73.48		

Table 5.5.1b – Comparison of statistics

Statistics	Observation	Weighted average	Difference
n_1	1551	1519	32
n_2	1526	1505	21
n_3	810	833	23
n_4	145	169	24
n_5	3	5	2
n_6	1	0	1
n_7	1494	1454	50
n_8	1512	1512	0
n_9	804	837	33
n_{10}	149	159	10
n_{11}	4	2	2
n_{12}	1	0	1

Table 5.5.2a – Results after 2 iterations of the algorithm

Parameter	True value	Initial guess	Estimate
θ_1	-0.7	-0.597	-0.606
θ_2	-0.5	-0.383	-0.394
θ_3	-0.2	-0.172	-0.161
θ_4	1.0	1.005	1.005
θ_5	4.0	3.999	3.999
θ_6	5.0	5.000	5.000
θ_7	-0.7	-0.600	-0.609
θ_8	-0.5	-0.386	-0.394
θ_9	-0.2	-0.171	-0.162
θ_{10}	1.0	1.005	1.020
θ_{11}	4.0	4.000	4.000
θ_{12}	5.0	5.000	5.000
# of GBB iterations	22		
Value of obj. function	-1.541		
Norm of gradient	3.728		

Table 5.5.2b – Comparison of statistics

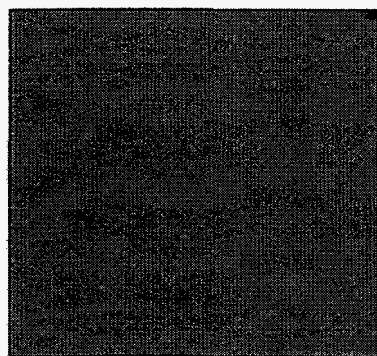
Statistics	Observation	Weighted average	Difference
n_1	1551	1549	2
n_2	1526	1526	0
n_3	810	810	0
n_4	145	146	1
n_5	3	1	2
n_6	1	1	0
n_7	1494	1492	2
n_8	1512	1512	0
n_9	804	803	1
n_{10}	149	151	2
n_{11}	4	3	1
n_{12}	1	0	1

Note that based on the convergence criteria $\|\nabla f(\theta_c)\|_2 \leq \varepsilon=10$, we have converged to $\hat{\theta}$. Note however, that $\hat{\theta}$ is not equal to the true values of θ . As discussed previously, we believe that $\hat{\theta}$ can only be estimated up to a constant vector of translation. If we subtract the constant $\bar{\theta} = 0.094$ from the estimate of $\hat{\theta}$ obtained in Table 5.5.2a, we obtain the following results (the value of $\bar{\theta}$ was chosen so that $\hat{\theta}_1 - \bar{\theta} = \theta_1$).

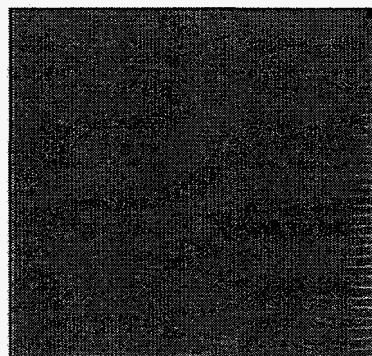
Table 5.5.3 – Translated maximum likelihood estimate

Parameter	True value	Translated MLE
θ_1	-0.7	-0.700
θ_2	-0.5	-0.488
θ_3	-0.2	-0.255
θ_4	1.0	1.099
θ_5	4.0	3.905
θ_6	5.0	4.906
θ_7	-0.7	-0.703
θ_8	-0.5	-0.488
θ_9	-0.2	-0.256
θ_{10}	1.0	0.926
θ_{11}	4.0	3.906
θ_{12}	5.0	4.906

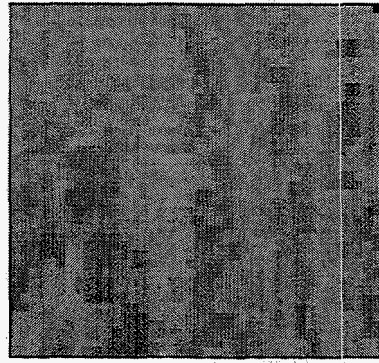
To illustrate that the values of $\hat{\theta}$ obtained in Table 5.5.2a give a reliable characterization of the true probability function, we generate a set of realizations by sampling $P_\theta(x)$ and a set of realizations by sampling $P_{\hat{\theta}}(x)$ using a Markov chain Monte Carlo sampling procedure. Note the basic structure characteristics of all realizations are the same.



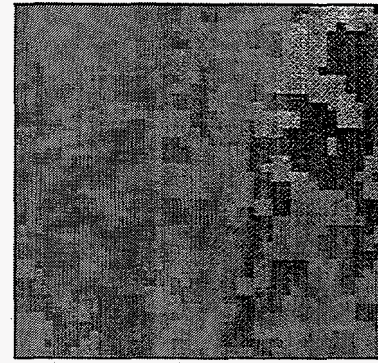
(a) Realization 1 (true θ)



(b) Realization 2 (true θ)



(c) Realization 1 (estimated θ)



(d) Realization 2 (estimated θ)

Fig. 5.5.4 – Realizations from the probability function based on true θ and realizations from the probability function based on the maximum likelihood estimate.

Figure 5.5.5 shows comparison of statistics n_1 for 3000 realizations obtained by sampling the true probability function with statistics n_1 for 3000 samples generated based on $P_{\hat{\theta}}(x)$ where $\hat{\theta}$ is the maximum likelihood estimate generated by our algorithm. Means as well as the standard deviations of n_1 from these realizations corresponding to each case are also listed on the top of the figures. By comparison, we can see that the estimated parameters give reliable characteristics of the true probability function.

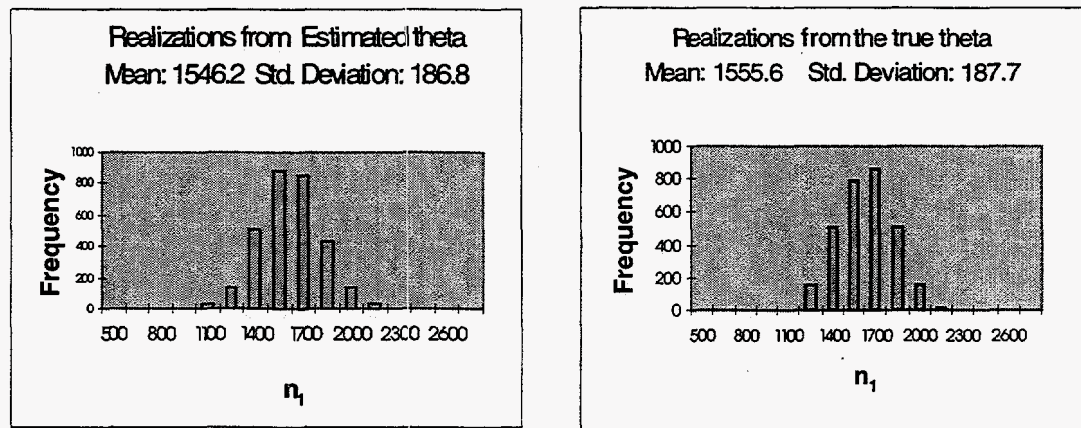


Fig. 5.5.5 – Comparison of statistic n_1 for 3000 samples from the probability functions based on the true parameters as well as estimated parameters.

Note, in generating the results of Figs. 5.5.4 and 5.5.5, we specified an initial guess for the maximum likelihood estimate which was close to the true value of θ . If a good initial guess is not available, the algorithm may either require an excessive **number** of iterations

or may fail to converge. Thus we implemented ideas of Derin and Elliott [14] to generate an initial guess for $\hat{\theta}$. This method allows us to generate an initial guess θ^0 for $\hat{\theta}$ directly from the observation y . Generation of the initial guess requires the solution of a linear least square problem. In our implementation, this least square problem was solved by singular value decomposition.

Table 5.5.4 shows the initial guess generated by our implementation of the Derin and Elliott method. Using these initial guesses in our algorithm, we obtained the maximum likelihood estimate of θ , shown in the second column of Table 5.5.5. The third column of Table 5.5.5 gives the number of each configuration contained in the observation x and the last column gives the weighted average of Eq. 5.5.27.

Table 5.5.4 – Initial Guess by Derin and Elliott method

$$\theta_1 = -0.519, \theta_2 = -0.103, \theta_3 = 0.263, \theta_4 = 0.59, \theta_5 = 0.0, \theta_6 = 0.0, \\ \theta_7 = -0.637, \theta_8 = -0.408, \theta_9 = -0.188, \theta_{10} = 1.0, \theta_{11} = 0.0, \theta_{12} = 0.0$$

Table 5.5.5 – Maximum likelihood estimate

Parameter	MLE by GBB	Observation	Weighted average
θ_1	-0.581	1551	1547
θ_2	-0.362	1526	1525
θ_3	-0.037	810	804
θ_4	0.973	145	152
θ_5	4.000	3	3
θ_6	5.000	1	1
θ_7	-0.569	1494	1490
θ_8	-0.373	1512	1514
θ_9	-0.110	804	801
θ_{10}	1.054	149	155
θ_{11}	4.000	4	3
θ_{12}	5.000	1	0

To check the validity of the maximum likelihood estimate $\hat{\theta}$, we generated 3000 realizations using the true probability function $P_{\theta}(x)$ and 3000 realizations from $P_{\hat{\theta}}(x)$ using a Markov chain Monte Carlo procedure. Fig. 5.5.6 shows a realization obtained from $P_{\theta}(x)$ and a realization obtained from $P_{\hat{\theta}}(x)$. Note that although the two realizations are distinctly different, the basic geological features are similar in both realizations. Fig. 5.5.7 presents two histograms of the statistic, n_3 , from the two sets of 3000 realizations, one set based on the true parameters and the other based on the MLE.

The means and the standard deviations of n_3 based on the respective 3000 samples are given on the figures. These results illustrate that appropriate samples of X can be generated using the probability function $P_{\hat{\theta}}(x)$ generated from the maximum likelihood estimate of θ constructed from our algorithm. For example, the mean of n_3 obtained from the probability function based on the estimates differs by only 10.9 from the mean value of n_3 obtained by sampling the true probability function.

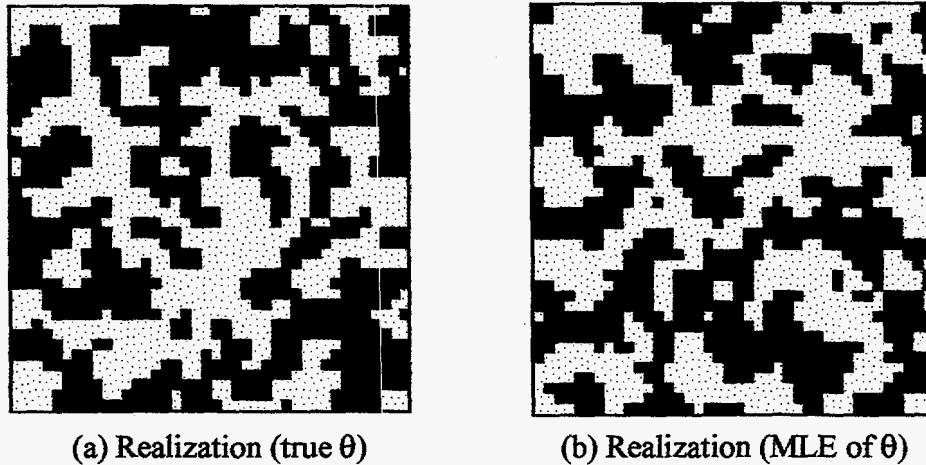


Fig. 5.5.6 – Realizations from the probability functions based on true θ and on the MLE of θ from initial guess obtained by Derin and Elliott method.

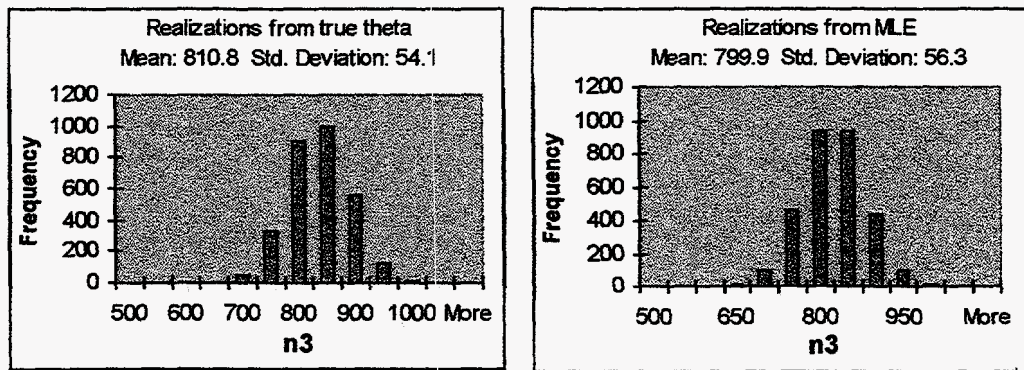


Fig. 5.5.7 – Comparison of statistics n_3 from the samples of the probability functions, one based on the true θ (left) and another based on the MLE (right).

Remarks

Although the results of the preceding example are quite impressive, it represents one of the few results where the procedure actually resulted in a reasonable

approximation of the maximum likelihood estimate. In the vast majority of cases, the overall algorithm failed to converge.

GENERATING RANDOMLY DISTRIBUTED CHANNELS IN FLUVIAL SYSTEMS

Introduction

The objective of this work is to generate realizations of facies distributions constrained to observations (well or outcrop) and geological interpretation. This particular section of the report focuses on Boolean methods. The work of this section uses well information together with geological interpretation to generate a wide variety of statistics (e.g. channel thickness, ratios of channel width to thickness etc.) and these are used to create probability distributions for the relevant quantities. These probability functions can then be sampled to generate realizations.

Throughout this section we focus on fluvial reservoirs. In these reservoirs the background is assumed to be nearly impermeable rock which contains permeable sandfilled channels. The models are important because sandfilled channels may contain oil and gas reserves. In the paper we model the distribution of channels within the background matrix for a three dimensional section of the reservoir. The model captures the natural clustering of channels and sheetsplay sands are easily added. We use Markov Chain Monte Carlo (MCMC) methods and both parametric and nonparametric density estimates in our statistical model building.

In nature, rivers are known to avulse (change channel course) in a quasisystematic manner (Ref. [36]). A regional avulsion event is typically followed by localized avulsion events. Geologic features, such as faulting, may control the preferred location of avulsion nodes. Continued deposition results in channel deposit clustering.

We use MCMC methods (see for example Refs. [2], [21], [51], and [26]) to generate the cluster centers for each set of channels. Noting that a random set of points tends to clump together, we use an MCMC method to generate random samples that tend to be well separated. The well-separated points represent the centers of clusters of channels. Once we have determined the channel cluster centers, individual channels can be generated around each cluster center according to a multivariate normal distribution. All of this can be done in accordance with measured statistics from fluvial deposits of interest.

However, in some cases the data measured from wells does not immediately give the statistics required. Take for example the problem of determining the distribution of maximum channel thickness. The information obtained from wells provides a random

sample of channel thickness and from this we need the distribution of maximum channel thickness. We show how this information can be obtained from well data.

Our goal is to distribute random channels in a rock volume of interest. We will first determine the channel centers in a 2 dimensional slice of the reservoir and then we will extend the channels into the third dimension. We want to create random collections of channels whose statistics match the data from field studies. The features that we want to capture are

- 1) the geometries and the statistics of those geometries of individual channels;
- 2) the clustering of channels; and
- 3) the fact that there are a random number of channels and channel clusters.

We begin with the problem of generating randomly distributed clusters.

Generating a random cluster of channels

By looking at a two dimensional cross-sections of the reservoir, we can accumulate statistics concerning the clustering of channels. An individual cluster may be modeled in many ways, but we will consider only the bivariate normal distribution with mean μ and covariance matrix Σ . By analyzing data we can form an estimate for the covariance of each cluster of channels. The standard formula and maximum likelihood estimator is (Ref. [18])

$$\Sigma_i = \frac{1}{N_i - 1} \sum_{k=1}^{N_i} (x_{i,k} - \bar{x}_i)(x_{i,k} - \bar{x}_i)^T \quad (5.6.1)$$

where $x_{i,k}$ denotes the k-th data point, \bar{x}_i is the sample mean of the data and N_i is the number of samples in the i-th cluster. If the covariance matrix for each cluster is the same, these covariance estimators can be pooled together to get a better estimate of this parameter. This is important since the estimates of Σ_i will be inaccurate when the sample sizes are small. The pooled covariance estimate using m individual covariance estimates is

$$\Sigma = \frac{1}{N} \sum_{i=1}^m N_i \Sigma_i \quad \text{where} \quad N = \sum_{i=1}^m N_i \quad (5.6.2)$$

The generation of random variables from the normal distribution with mean μ and covariance matrix Σ is straightforward. Let A be any matrix with the property that $\Sigma = AA^T$. If z is a standard normal deviate, i.e. mean zero and identity covariance matrix, then $x = Az + \mu$ is normally distributed with mean μ and covariance matrix Σ . The matrix A can be obtained by performing a Cholesky decomposition of the matrix Σ . The number of

channels in each cluster is considered to be a random variable with mean S and this distribution is sampled to determine the cluster sizes

Generating random cluster centers

We now describe our model for the clustering of channels. We recognize that there is a physical mechanism that causes this clustering. That is, small events can generate small changes in the location of the channel. New channels may erode older channels and in this case, the channels can be considered to overlap. Of course, the rock record will only preserve the latest of these channels. Our model handles this by assigning a depth order priority to the channels and using the channels with a higher priority to replace parts or all of some of the lower channels. Regional avulsion results in the formation of a new cluster.

We now describe a method for distributing the centers of these channel clusters. One possible option would be to distribute the cluster centers uniformly over a rectangular region of interest. But a uniformly distributed set of points in a rectangle will often have sets of points which are very close together and this would cause the clusters to overlap or for the clusters to cluster together. Figure 5.6.1 contains the plot of 100 randomly distributed points in the plain. The clustering or structures in this figure are not desirable in this application and so we will generate randomly distributed points using an MCMC method. We can use this to generate a sample of cluster centers which has a tendency to avoid one another.

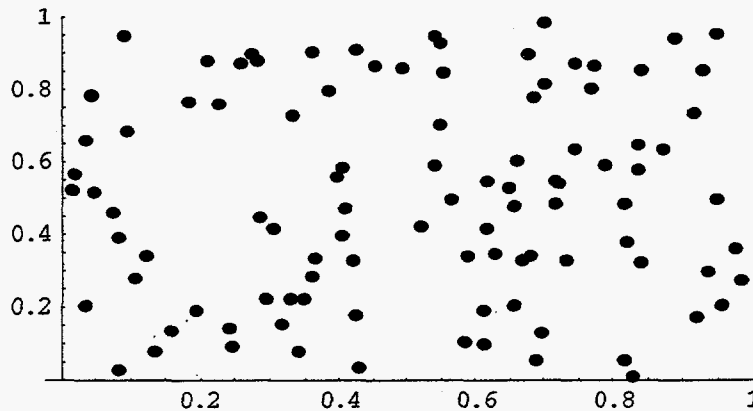


Fig. 5.6.1 -- 100 random points in a rectangular region.

To accomplish this we define a probability function

$$p(\mu_1, \mu_2, \dots, \mu_m) = C \text{Exp} \left(- \sum_{i < j} f(\|\mu_i - \mu_j\|) \right) \quad (5.6.3)$$

for some normalizing constant C , function f and norm $\| \cdot \|$. The function f should be nonincreasing on $(0, \infty)$ with a large value when its argument is small (i.e. when points are "too" close together) and should be essentially zero outside of some range. One such example would be

$$f(x) = \begin{cases} a & \text{if } x < b \\ \frac{c}{1 + \frac{4(x-b)}{d}} & \text{otherwise} \end{cases} \quad (5.6.4)$$

with nonnegative parameters (a, b, c, d) . These parameters all have a clear physical meaning. First the function is nonincreasing as long as $c \geq a$ and if we choose $c = a$, then the function f is also continuous on $[0, \infty)$. If a is much greater than c then the probability of points lying within a distance b of one another is small. When $x \gg d$, $f(x)$ will be close to zero which corresponds to an event with a relatively high probability. Figure 5.6.2 contains a graph of f using typical values of (a, b, c, d) . Since we are using a covariance matrix Σ , it is natural to define $\|x\|^2 = x^T \Sigma^{-1} x$. With this choice, the statistical model is independent of changes in scale.

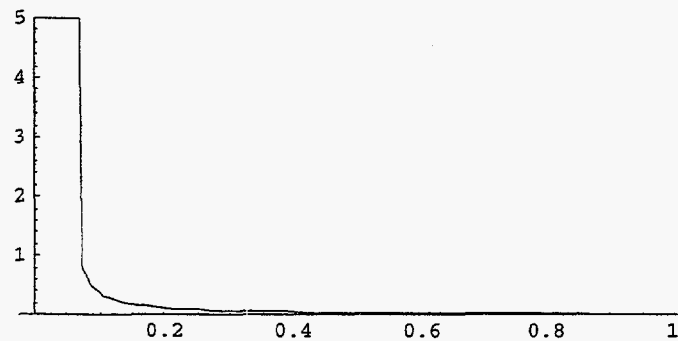


Fig. 5.6.2 -- Graph of f using $a=5$, $b=.07$, $c=1$ and $d=b$.

The main difficulty with this approach is that there is an edge effect. When a point is on the edge, it has fewer neighbors, and hence the probability of putting a point on the edge is larger than the probability of having a point fall in an equal size area away from the edge. But for the applications that we are working with, the area or volume of rocks that we are looking at is an arbitrary slice of some larger area or volume. So the edge has no physical meaning and should certainly not have any effect on the simulation.

We propose a method for fixing this problem. We first observe that points on the edge have about one half as many neighbors as points in the interior, while points in the corners have about one fourth as many neighbors. So we might consider defining a weight function of the form

$$h(x) = \begin{cases} 2 - x/b & \text{if } x \leq b; \\ 1 & \text{if } b < x < b-1; \\ 2 - (1-x)/b & \text{if } x \geq 1-b. \end{cases} \quad (5.6.5)$$

A more general model is a weight function of the form

$$h(x) = \begin{cases} \alpha(1-x/b) + x/b & \text{if } x \leq b; \\ 1 & \text{if } b < x < b-1; \\ \alpha(1-(1-x)/b) + (1-x)/b & \text{if } x \geq 1-b \end{cases} \quad (5.6.6)$$

that simplifies to the formula above when $\alpha=2$. This function gives a weight of α on the edges and α^2 in the corners. In the horizontal direction, points which are within $b_w = b \sigma_x$ (where σ_x^2 is the variance in the horizontal direction) are strongly penalized (at least when α is large). In the vertical direction, points which are within $b_h = b \sigma_y$ (where σ_y^2 is the variance in the vertical direction) are strongly penalized. So the edge has a strong effect when you are within b_w units from the left or right edges and when you are within b_h units from the top and bottom boundary. Since the function is defined on a unit square we define a weight function $g(x,y) = h(x/w, b_w)h(y/h, b_h)$. A graph of this function is shown in Figure 5.6.3.

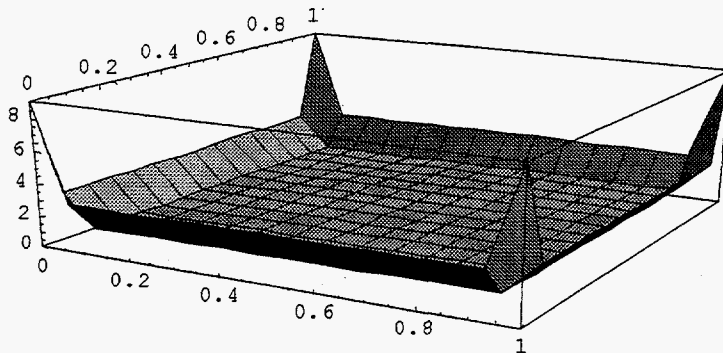


Fig. 5.6.3 – A graph of the correction function with $\alpha=3$ and $b=.07$.

We now modify the way we express the probability function from Eq. 5.6.3 in order to include the weight function. So we write

$$p(\mu_1, \mu_2, \dots, \mu_m) = C \text{Exp} \left(- \sum_i \sum_{i \neq j} f(\|\mu_i - \mu_j\|/2) \right). \quad (5.6.7)$$

For each point μ we should adjust the weight associated with the distance to each of the other points by multiplying by $g(\mu_i)$ which leads to

$$p(\mu_1, \mu_2, \dots, \mu_m) = C \text{Exp} \left(- \sum_i \sum_{i \neq j} f(\|\mu_i - \mu_j\|) g(\mu_i) / 2 \right). \quad (5.6.8)$$

But if we calculate the density this way, then we would compute each distance twice, so we rewrite the expression as

$$p(\mu_1, \mu_2, \dots, \mu_m) = C \text{Exp} \left(- \sum_{i < j} f(\|\mu_i - \mu_j\|) (g(\mu_i) + g(\mu_j)) / 2 \right). \quad (5.6.9)$$

We now show how we can use an MCMC method to sample from this distribution even though the normalizing constant C is unknown. The scheme is based on proposing new values from the sample and accepting them or rejecting them based on the test

$$U < \frac{p(\mu_1, \dots, \mu'_k, \dots, \mu_m)}{p(\mu_1, \dots, \mu_k, \dots, \mu_m)} \quad (5.6.10)$$

Specifically, at each step of the algorithm, we choose a random index k and propose to exchange μ_k with some random point μ'_k . After generating a random point U between 0 and 1, the new point μ'_k is accepted if Eq. 5.6.10 is satisfied and otherwise the new point is rejected and the old point is retained. This sampling scheme asymptotically samples the probability density function $p(\mu_1, \dots, \mu_m)$. Of course, most of the terms cancel in Eq. 5.6.10 leaving us with

$$\text{Exp} \left(- \sum_{i \neq k} f(\|\mu_i - \mu'_k\|) (g(\mu_i) + g(\mu'_k)) / 2 - \frac{f(\|\mu_i - \mu_k\|) (g(\mu_i) + g(\mu_k)) / 2}{f(\|\mu_i - \mu_k\|) (g(\mu_i) + g(\mu_k)) / 2} \right). \quad (5.6.11)$$

When points are on the edge, then there are only about one half as many nearby points, so it is natural to try a weight of $\alpha=2$. Numerical experiments with Mathematica seem to imply that the choice of the value of $\alpha=2$ is too small. We can compute the expected number of points which are within b_w or b_h of a horizontal or vertical edge respectively and compare that with the results of simulations. These limited simulations appear to show that a value closer to $\alpha=3$ in Eq. 5.6.6 gives better results in some cases. A simulation generating 100 random point in a unit rectangle is contained in Figure 5.6.4.

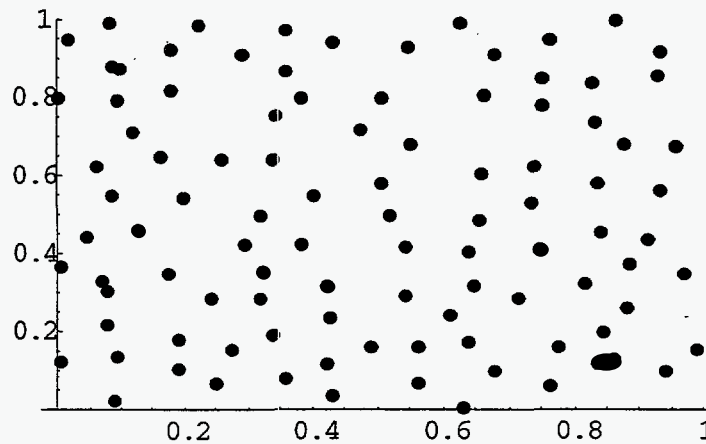


Fig. 5.6.4 - A picture of 100 well spaced points generated using the MCMC method.

The MCMC method is implemented in a standard way except for one slight change. Since the value of the exponent in Eq. 5.6.11 can be a large negative number, we choose to not evaluate the exponential function at this point. So in place of the acceptance rule in Eq. 5.6.10 (using U as a uniformly distributed random variate) we compare $\ln U$ to the exponent of Eq. 5.6.11.

The deconvolution data problem

As we work with these problems we need statistical descriptions for the model parameters. Unfortunately, in the case of channel thickness, we are not able to measure the maximum thickness from well data, but instead what we have are the results of measuring thickness of channels at random locations. Let's suppose that we have a single channel geometry for which the thickness is the only varying parameter. If $f(b)$ denotes the distribution of maximum thickness of channels and $g(t|b)$ denotes the distribution of random thickness given a choice of parameter b , then

$$g(t) = \int g(t|b)f(b) db, \quad (5.6.12)$$

is the distribution of thickness which would be observed by measuring a random thickness of a random channel. What we can measure is the resulting thickness and so we have data on the density function $g(t)$. Given a fixed geometry we can also determine the conditional density function $g(t|b)$. What we want to know, of course, is the distribution $f(b)$. This is then some sort of deconvolution problem.

Let's discuss the conditional density function $g(t|b)$ first. Suppose that we are given channels which look like the bottom half of the ellipse $x^2/a^2 + y^2/b^2 = 1$ so that the maximum depth is b . If we take a random vertical slice through this region, what is that distribution of values? First observe that if you take a slice through the region with a positive x value, then by symmetry, this gives you the same distribution as you would get from a random x value between $-a$ and a . We obtain the distribution by thinking of this process as a change of variables $y = -b\sqrt{1-x^2/a^2}$. The original density for X is

$$h(x) = \begin{cases} 1/a & \text{if } 0 \leq x \leq a \\ 0 & \text{otherwise} \end{cases} \quad (5.6.13)$$

and the Jacobian of this transformation is

$$\frac{dx}{dy} = \frac{d}{dy} \frac{a}{b} \sqrt{b^2 - y^2} = \frac{a}{b} \frac{-y}{\sqrt{b^2 - y^2}} \quad (5.6.14)$$

This gives us a density for $g(t|b)$ which is

$$g(t|b) = \frac{a}{b} \frac{t}{\sqrt{b^2 - t^2}} \chi_{[0,b]}(t) \frac{1}{a} = \frac{1}{b} \frac{t}{\sqrt{b^2 - t^2}} \chi_{[0,b]}(t), \quad (5.6.15)$$

which is fortunately independent of a . This brings us to the deconvolution type problem of determining $f(b)$ given $g(t)$ and $g(t|b)$. The procedure that we develop will be based on the type of information that we have on $g(t)$. Most likely, we have only a nonparametric density estimate of the function $g(t)$ and want to use this to construct an estimate of $f(b)$ of the same type. Consider, for example, the case that we have a histogram estimate for $g(t)$ with bin boundaries at the points (a_0, a_1, \dots, a_n) . We construct an estimate of $f(b)$ with the same structure. So we assume that $f(x) = c_i$ for $a_i \leq x < a_{i+1}$ and that f is zero outside of the interval $[a_0, a_n]$. We then have that

$$d_i \equiv \int_{a_i}^{a_{i+1}} g(t) dt = \sum_{j=0}^{n-1} c_j \int_{a_i}^{a_{i+1}} \int_{a_j}^{a_{j+1}} g(t|b) dt, \quad (5.6.16)$$

for $i=0, 1, \dots, n-1$. This provided us with a system of n equations in n unknowns. Of course we have, in addition to these equations, the constraints that the constants satisfy

$$\sum_{j=0}^{n-1} c_j (a_{j+1} - a_j) = 1 \quad \text{and} \quad c_j \geq 0 \quad \text{for } j = 0, 1, \dots, n-1. \quad (5.6.17)$$

The values of the integrals

$$\int_{a_i}^{a_{i+1}} \int_{a_j}^{a_{j+1}} g(t|b) dt \quad (5.6.18)$$

can be computed. First we note that $g(t|b) = 0$ for $t > b$. This implies that when $i > j$ that $a_j \leq b \leq a_{j+1} \leq a_i \leq t \leq a_{i+1}$ so that $g(t|b) = 0$ except perhaps at one point. So the value of Eq. 5.6.18 is zero in this case. To evaluate the integral in the other cases we will need to use

$$\int \frac{t}{x\sqrt{x^2 - t^2}} dx = \sec^{-1} \frac{x}{t} + C, \quad (5.6.19)$$

and

$$\int \sec^{-1} \frac{x}{t} dx = x \sec^{-1} \frac{b}{x} - \sqrt{b^2 - x^2} + C. \quad (5.6.20)$$

So in the case that $i=j$, then

$$\int_{a_i}^{a_{i+1}} \int_{a_j}^{a_{j+1}} g(t|b) dt = t \sec^{-1} \frac{b}{t} - \sqrt{b^2 - t^2} \Big|_{b=a_j}^{b=a_{j+1}} \Big|_{t=a_i}^{t=a_{i+1}}. \quad (5.6.21)$$

Since for $c > 0$,

$$\lim_{a_i \rightarrow 0^+} \sec^{-1} \frac{c}{a_i} = \pi/2 \quad \text{then} \quad \int_0^{a_1} \int_0^{a_1} g(t|b) dt = a_1. \quad (5.6.22)$$

In the case that $i < j$, then

$$\int_{a_i}^{a_{i+1}} \int_{a_j}^{a_{j+1}} g(t|b) dt = \sqrt{a_{i+1}^2 - a_i^2} - a_i \sec^{-1} \frac{a_{i+1}}{a_i} \quad (5.6.23)$$

where we use

$$\lim_{t \rightarrow 0^+} t \sec^{-1} \frac{b}{t} = 0. \quad (5.6.24)$$

to handle the case that $i=0$ and $a_0=0$.

The linear system in Eq. 5.6.16 can be written as $Ac=d$, where $c=(c_0, c_1, \dots, c_{n-1})^T$ and $d=(d_0, d_1, \dots, d_{n-1})^T$. The ij -th element of A is

$$\int_{a_i}^{a_{i+1}} \int_{a_j}^{a_{j+1}} g(t|b) dt, \quad (5.6.25)$$

and is zero when $i>j$. Hence A is upper triangular. Furthermore, if we assume that $a_0=0$, then since $g(t|b)$ is a probability density function, then we have that the value of the sum of the j th column of A is

$$\sum_{i=0}^{n-1} \int_{a_i}^{a_{i+1}} \int_{a_j}^{a_{j+1}} g(t|b) db dt = \int_{a_j}^{a_{j+1}} \int_{a_0}^{a_n} g(t|b) dt db = \int_{a_j}^{a_{j+1}} 1 db = a_{j+1} - a_j. \quad (5.6.26)$$

This implies that

$$\sum_{i=0}^{n-1} d_i = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} A_{ij} c_j = \sum_{j=0}^{n-1} \sum_{i=0}^{n-1} A_{ij} c_j = \sum_{j=0}^{n-1} (a_{j+1} - a_j) c_j. \quad (5.6.27)$$

And this implies that, if d is a vector of percentages which sums to 1, then the constants c will define a probability density function and so this extra constraint is automatically satisfied. Unfortunately, we must also satisfy $c_i \geq 0$ for each i and this is not automatically satisfied. Furthermore the process of deconvolution increases the level of noise in the data and smoothing may be required. To handle the first problem we might naturally consider the following optimization problem

$$\begin{aligned} & \text{Min} \|Ac - d\| \\ & \text{Subject to } c_i \geq 0, i=0, 1, \dots, n-1 \text{ and} \quad (5.6.28) \\ & \sum_{j=0}^{n-1} c_j (a_{j+1} - a_j) = 1 \end{aligned}$$

whose solution should provide an excellent solution to our problem.

Another procedure for the construction of a density function f which satisfies these constraints is the following.

1. Use the standard backsolve algorithm to solve the equation $Ac=d$ with the modification that c_i is set to zero if the calculated value from the backsolve step is less than zero.
2. Upon completion of step 1, renormalize c_j to satisfy

$$\sum_{j=0}^{n-1} c_j (a_{j+1} - a_j) = 1. \quad (5.6.29)$$

3. If smoothing is required apply a linear filter which has the property that it preserves the sum of the values and preserves nonnegativity.

In particular we define

$$c_i = \text{Max} \left[0, \frac{d_i - \sum_{j=i+1}^{n-1} a_{ij} c_j}{a_{ii}} \right], \quad (5.6.30)$$

for $i=n-1, n-2, \dots, 0$. Then for

$$\text{oldsum} = \sum_{j=0}^{n-1} c_j (a_{j+1} - a_j). \quad (5.6.31)$$

set $c_i = c_i \text{oldsum}$ for $i=0, 1, \dots, n-1$. Finally we define our filtered estimate by

$$\begin{aligned} cf_0 &= \frac{2c_0 + c_1}{3}, \\ cf_i &= \frac{c_{i-1} + c_i + c_{i+1}}{3}, \\ cf_{n-1} &= \frac{c_{n-2} + 2c_{n-1}}{3}. \end{aligned} \quad (5.6.32)$$

Nonparametric density estimation

For the distribution of cluster channel centers we were able to use a multivariate normal density function. The mean and covariance matrix for random variables, that are normally distributed, completely determines the function. The mean and covariance matrices are the only parameters in this model. Unfortunately, in many cases random variables come from density functions whose form is unknown and so we use nonparametric density estimation techniques.

The histogram (normalized to integrate to 1) is the most familiar example of a nonparametric density estimate. Recently B-Spline density estimators (see Ref. 44) have been studied for use in computer graphics and other applications. It has been discovered that these nonparametric density estimates are efficient to use since they are defined in terms of splines and that B-Spline nonparametric density estimates converge rapidly to the true density function as the sample size increases.

Let $N^m(x)$ be the m -th order normalized uniform B-Spline associated with the evenly spaced knots $0, 1, \dots, m$, i.e.,

$$N^m(x) = \sum_{i=0}^m (-1)^i \binom{m}{i} \frac{(x-i)_+}{(m-1)!}, \quad (5.6.33)$$

where

$$(z)_+ = \begin{cases} z, & \text{if } z > 0; \\ 0, & \text{otherwise.} \end{cases} \quad (5.6.34)$$

Given evenly spaced data point $\{x_i\}$ with $h=x_{i+1}-x_i$ we define

$$B_i^m(x) = N^m\left(\frac{x-x_i}{h}\right). \quad (5.6.35)$$

Given data X_1, X_2, \dots, X_N the B-Spline nonparametric density estimate on the interval $[a, b]$ is

$$\hat{f}_N(x) = \sum_i \frac{\alpha_i}{b_i} B_i^m(x), \quad (5.6.36)$$

where

$$b_i = \int_a^b B_i^m(x) dx \quad \text{and} \quad \alpha_i = \sum_{k=1}^N B_i^m(X_k), \quad (5.6.37)$$

for each integer i . This estimator can be used to estimate density functions over bounded, semi-infinite and unbounded intervals.

Estimation of parameters for the Oligocene Frio Sandstone of South Texas

Working in close collaboration with other members of the research team we have been able to gather data on the Oligocene Frio Sandstone of South Texas (Refs. 34, 59). It has been determined by observation that this geological interval contains two subintervals which have different characteristics. In the top portion of this interval we find channels which tend to cluster together, while in the lower portion, channels are randomly scattered throughout the rock volume with no apparent clustering. In the top lay we have statistics on the location of 59 channel centers, grouped into 10 clusters. Using this data we have estimated the pooled covariance matrix for this data to be

$$\begin{pmatrix} 14624790 & -10175 \\ -10175 & 123.68 \end{pmatrix}. \quad (5.6.38)$$

The correlation for this data is quite small and since data is measured from a marker bed it should probably be almost zero. So we will use a covariance matrix for the simulation of the form

$$\begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix}, \quad (5.6.39)$$

where $\sigma_x=3824.24$ and $\sigma_y=11.12$. The number of clusters in three different cross sections is 2, 2 and 4 respectively. We currently use a random integer between 2 and 6 for our simulations. The number of channels in these 8 clusters ranges from 2 to 14 and we currently use a random integer between 2 and 14 for our simulations.

We have data on channel thickness from well information (see Ref. 34). In Figure 5.6.5 we have a histogram of the original thickness data and the deconvolved thickness data. The deconvolved thickness data is represented by the right most histogram.

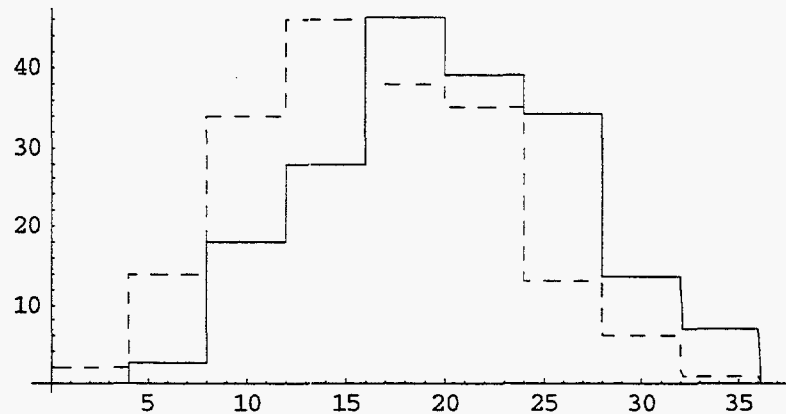


Fig. 5.6.5 – Distribution of channel thickness

We have made the observation that the width of channels varies with thickness and so we have determined the distribution of the width to thickness ratio. After discussion with geologists it was determined that a minimum ratio of 50 to 1 was reasonable and so the nonparametric density estimate of the ratio of width to thickness was determined on the interval $[50, \infty)$. The values for the weights are (.1252, .6674, .1681, .03926) using linear splines with 4 basis function on the interval $[50, 250]$. A graph of the nonparametric density estimate is given in Figure 5.6.6.

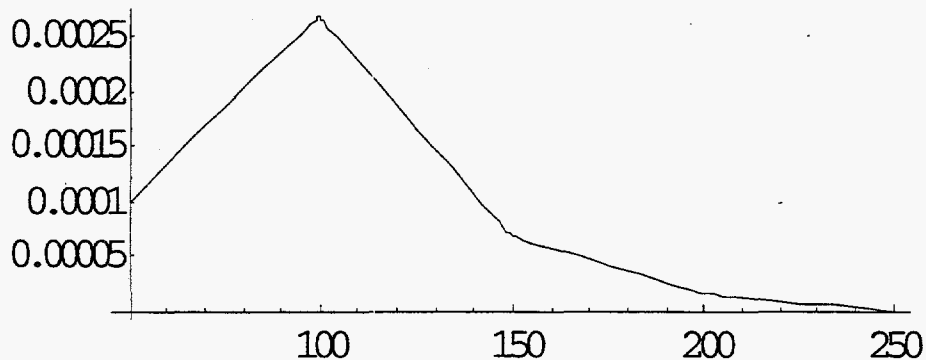


Fig. 5.6.6 – Distribution of width to thickness ratio.

The final distribution that we determined is the angle which the channels make with the primary direction of the river flow. We assume that the rock volume for the simulation is aligned in the primary direction of drainage for the river system. Under the assumption that the sign of the angle is random, we have determined a nonparametric density estimate of the magnitude of the angle on the interval $[0,90]$. The estimate is contained in Figure 5.6.7. The estimated alpha values are (.3088, .5053, .1544, .0316) using linear B-Splines with 4 basis function on the interval $[0,90]$.

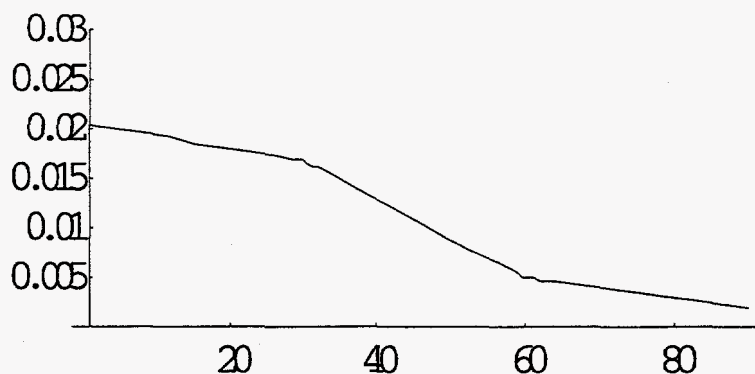


Fig. 5.6.7 – Distribution of angle magnitudes.

We have now described all of the distributions that we will use in the simulations. We will generate random channels over a three dimensional volume. The first simulation is of the upper top section of the interval and has dimensions 60,000 ft by 80 ft by 16,000 ft.

The simulation is performed by sampling the distribution of channel cluster centers in a vertical slice of the reservoir using the MCMC method. Next the bivariate normal density function is sampled to provide the centers of individual channels in this vertical slice. For each channel we sample the thickness distribution and the ratio of width to thickness distribution to determine the maximum thickness and width of the channel at the top of the channel. The angle distribution is then sampled and given a random sign to determine the angle that the three-dimensional channel makes with the vertical slice. Again, it is assumed the general direction of the drainage is perpendicular to the vertical slice. We then create channels of length 1 mile using the values generated above. Additional sections of the channel can easily be generated. In the simulation below we have assumed that each channel section for an individual channel has the same width and thickness, but this could easily be modified so that width and thickness of the channel change along its length. To determine the angle of each of the additional sections of the channel we sample the angle distribution and give the angles of connecting section opposite sign.

This sampling procedure provides one realization for the top interval where clustering is found. However, on the geologic interval immediately below the first layer, no clustering is found. The same software used to generate clusters of channels can also be used to generate a sample of channels that are randomly located within a slice of the rock volume and then extended to create three-dimensional channels. The final simulation composed of the two geologic intervals combined is shown in Figs. 5.6.8a, 5.6.8b and

5.6.8c at the end of this section. These images have been created using Mathematica and a description of the software is located in Appendix 5B. We note that sheetsplay sands could easily be added to the model but that statistics that determine the frequency and dimensions of the sheetsplay sands have not been compiled.

Closing Remarks

This section of the report describes a preliminary attempt to model a fluvial system using a combination of MCMC methods, and parametric and nonparametric density estimation. We have used MCMC methods to sample the distribution of channel centers. The bivariate normal distribution has been sampled to determine the location of individual channels within each cluster. Deconvolution of well data to obtain the distribution of maximum thickness values has also been described and implemented. Finally nonparametric density estimates of channel thickness, the width to thickness ratio and the channel angles have been obtained and sampled.

It should be noted that the statistical model is nonstationary due to several distinct factors. It is nonstationary due to the fact that we have identified two distinct layers in the Frio interval which have distinct statistical characteristics. In the top layer channels tend to cluster while in the lower layer they do not and are instead simply randomly distributed in the volume. The use of Boolean models in general provides a nonstationary model. Even if the distribution of object is constant over a region or volume the physical properties of the rock within an object are distinct from the rock outside of the object.

Future work will more fully integrate measured data from wells, seismic data and geological interpretation to make a model which more accurately reflects reality.

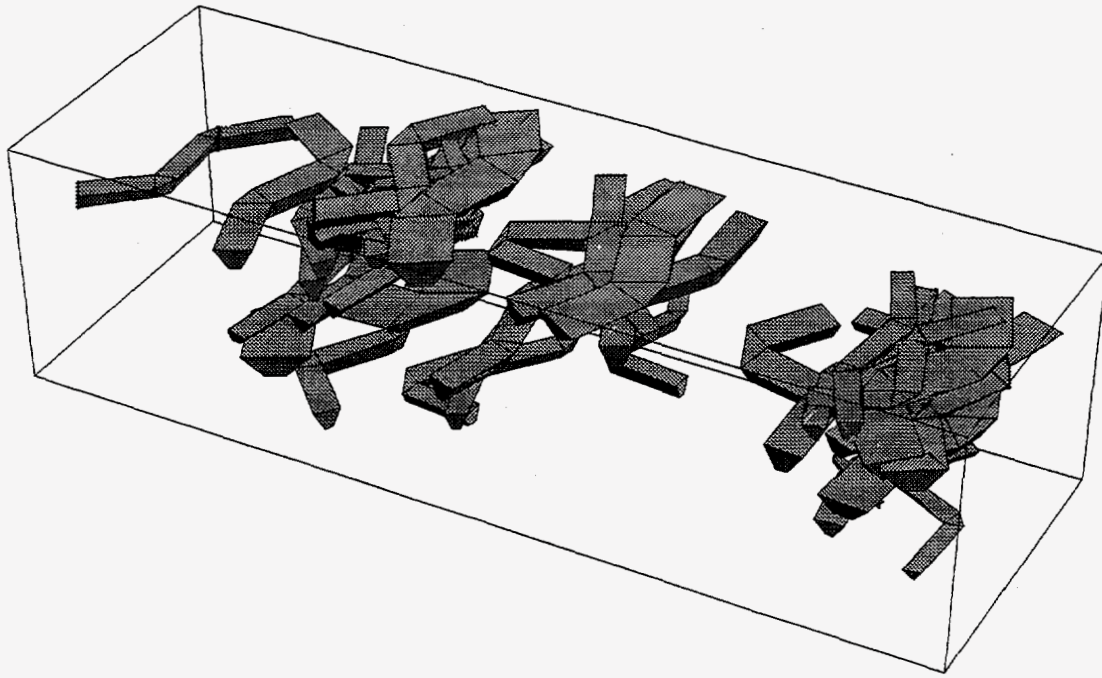


Fig. 5.6.8a – An oblique view of the simulated reservoir.

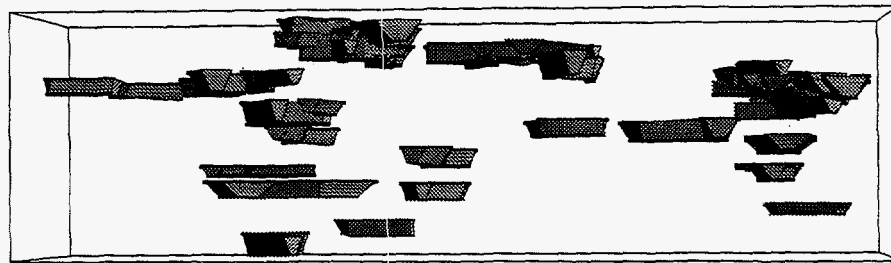


Fig. 5.6.8b – A side view of the simulated reservoir.

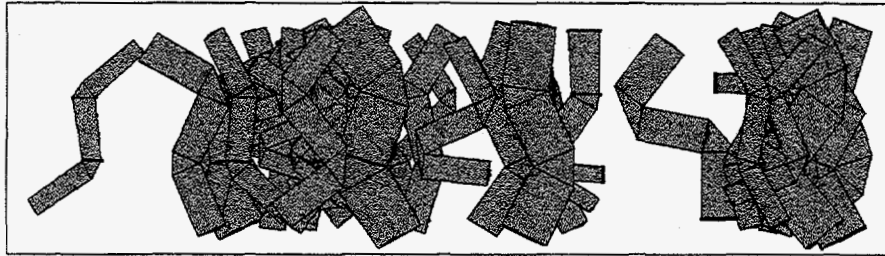
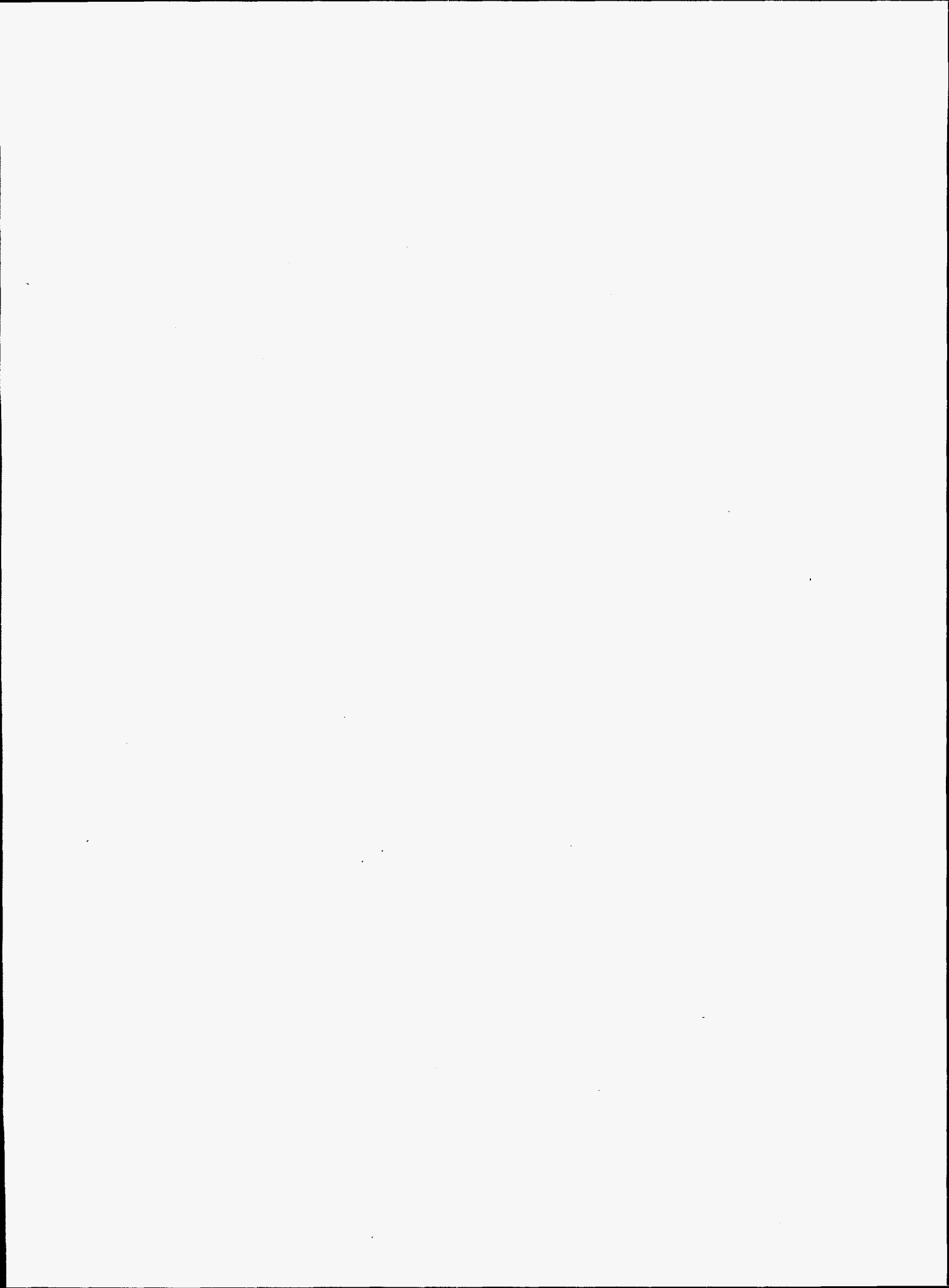


Fig. 5.6.8c – A top view of the simulated reservoir.



Bibliography

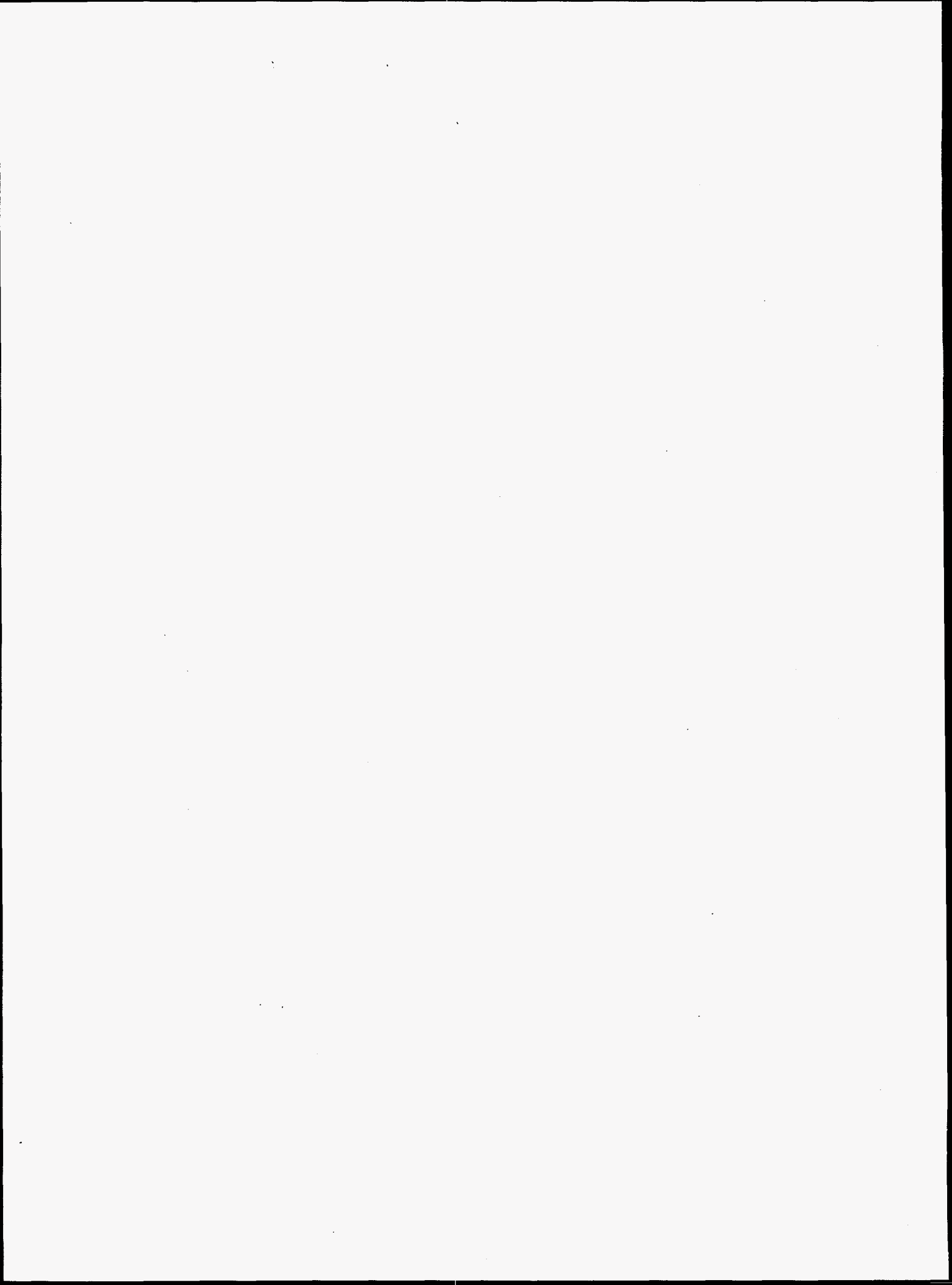
1. Beseg, J. (1975): "Statistical Analysis of Non-Lattice Data," *The Statistician*, Vol. 24, p179-195.
2. Beseg, J. (1974): "Spatial Interaction and the Statistical Analysis of Lattice System," *J. Royal Statist. Soc., Series B*, Vol. 36, p192-236.
3. Beseg, J. and Green, P. J.(1993): "Spatial Statistics and Bayesian Computation," *J. Royal Statist. Soc., Series B*, Vol. 55, No. 1, p25-37.
4. Beseg, J. (1986): "On the Statistical Analysis of Dirty Pictures," *Journal of Royal Statistical Society, Series B*, Vol. 48, No. 3, p259-302.
5. Bonet-Cunha, L., Oliver, D. S., Redner, R. A. and Reynolds, A. C. (1996): "A Hybrid Markov Chain Monte Carlo Method for Generating Permeability Fields Conditioned to Multiwell Pressure Data and Prior Information," SPE 36566, presented at the 1996 SPE Annual Technical Conference and Exhibition, Denver, Oct. 6-9.
6. Bonet-Cunha, L., Oliver, D. S., and Reynolds, A. C. (1995): "Simulated Annealing and Markov chain Monte Carlo Methods for Generating Permeability Fields Conditioned to The Prior Mean, the Varigram and Multiwell Pressure Data," TUPREP Research Report 12 (May, 1995)134-216
7. Chen, C. C. (1988): "Markov Random Field Models for Image Analysis," PhD thesis, East Lansing, Michigan State University.
8. Comets, F. and Gidas B. (1992): "Parameter Estimation for Gibbs Distribution from Partially Observed Data," *The Annals of Applied Probability*, Vol. 2, No. 1 p142-170.
9. Cressie, Noel A. C. (1993): "Statistics for Spatial Data," Wiley-Interscience Publication.
10. Cressie, N. and Lele S. (1992): "New Models for Markov Random Fields," *Journal of Applied Probability*, Vol. 29, p877-884.
11. Cross, G. R. and Jain A. K. (1983): "Markov Random Field Texture Models," *IEEE Transactions on PAMI*, Vol. PAMI-5, No. 1, Jan.
12. Damsleth, E., Tjolsen, C. B., Omre, H. and Holden, H. H. (1992): "A Two-Stage Stochastic Model Applied to a North Sea Reservoir," *JPT*, p402-408.
13. Damsleth, E. and Holden, L. (1994): "Mixed Reservoir Characterization Methods," SPE 27969 presented at the SPE/University of Tulsa Symposium, Tulsa, Aug. 29-31.

14. Derin, H., Elliott, H. (1987): "Modeling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields," IEEE Trans., Vol. PAMI-9, No. 1, Jan.
15. Derin, H. and Kelly, P. A. (1989): "Discrete-Index Markov-Type Random Processes," Proceedings of the IEEE, Vol. 77, No. 10, Oct.
16. Dobrushin R. L.(1965): "Existence of A Phase Transition in Two-Dimensional and Three-Dimensional Ising Models," Theory of Probability and Its Applications, Vol. X, No. 2, p193-213.
17. Dubes, R. C. and Jain, A. K. (1989): "Random Field Models in Image Analysis," Journal of Applied Statistics, Vol. 16, No. 2.
18. Duda, R. and Hart, P. (1973): "Pattern Classification and Scene Analysis," Wiley Interscience, John Wiley and Sons, New York.
19. Egeland, T., Geogesen, F., Knarud, R. and Omre, H. (1993): "Multifacies Modeling of Fluvial Reservoirs," SPE 26502 presented at the 1993 SPE Annual Technical Conference and Exhibition, Houston, Oct. 3-6.
20. Gelfand A. E. and Carlin B. P. (1993): "Maximum-Likelihood Estimation for Constrained or Missing Data Models," The Canadian Journal of Statistics, Vol. 21, No. 3, p303-311.
21. Geman, S. and Geman, D. (1984): "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images," IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-6, No. 6, p721-741.
22. Geogesen, F. and Omre, H. (1992): "Combining Fibre Pocesesses and Gaussian Random Functions for Modeling Fluvial Reservoirs." Geostatistical Troia 92 edited by A. Soares, p 425-439.
23. Geyer, C. J. and Thompson E. A. (1992): "Constrained Monte Carlo Maximum Likelihood for Dependent Data," Journal of Royal Statistical Society, Series B, Vol. 54, No. 3, p657-699.
24. Geyer, C. J. (1993): "Estimating Normalizing Constants and Reweighting in Markov Chain Monte Carlo," Technical Report No. 568, School of Statistics, University of Minnesota.
25. Geyer, C. J. (1991): "Reweighting Monte Carlo Mixtures," Technical Report No. 568, School of Statistics, University of Minnesota.

26. Gilks, W. R., Richardson, S. and Spiegelhalter, D. H. (1997): "Markov Chain Monte Carlo in practice," Chapman & Hall, New York.
27. Griffith D. A. and Amrhein C. G. (1983): "An Evaluation of Correction Techniques for Boundary Effects in Spatial Statistical Analysis: Traditional Methods," *Geographical Analysis*, Vol. 15, No. 4, Oct.
28. Haldorsen, H. H. and Damsleth, E. (1990): "Stochastic Modeling," *JPT* (April), p404-412.
29. Haldorsen, H. H. and Lake, L. W. (1984): "A New Approach to Shale Management in Field Scale Models," *Soc. Pet. Eng. J.* p 447-457.
30. Hastings, W. K. (1970): "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrika*, Vol. 57, No. 1. P97-109.
31. Hoiberg, J., Omre, H. and Tjelmeland, H. (1990): "Large-Scale Barriers in Extensively Drilled Reservoirs," *Proceeding of 2nd European Conference on the Mathematics of Oil Recovery*, D. Guerillot and O. Guillon(eds.) Editions Technip, Paris, p31-41.
32. Hoiberg, J., Omre, H. and Tjelmeland, H. (1992): "A Stochastic Model for Shale Distribution in Petroleum Reservoir," *2nd CODATA Conference on Geomathematics and Geostatistics*, P. A. Dund and J. J. Royer, Navy, p135-153.
33. Jensen, S. T. and Johansen, S. (1997): "Globally Convergent Algorithms for Maximizing a Likelihood Function," *Biometrika*, Vol. 78, No. 4, p867-877.
34. Kerr, D, Epili, D. Kelkar, M., Redner, R. and Reynolds, A. C. (1996): "Reservoir Architecture Modeling: Nonstationary Models for Quantitative Geological Characterization," *Second Quarterly Report, BDM-OK/DOE, Contract #0047.*
35. Kinderman, R. and Snell, J. S. (1980): "Markov Random Fields and their Applications," *Amer. Math. Soc.*
36. Mackey, S. D. and Bridge, J. S. (1995): "Three Dimensional Model of Alluvial Stratigraphy," *Journal of Sedimentary Research*, B65, p7-31.
37. Metropolis, N., Rosenbluth, A. and Teller, E. (1953): "Equation of State calculations by Fast Computing Machines," *Journal of Chemical Physics*, Vol. 21, p1087-1092.

38. Munthe, K. L., Holden, L. and Townsend, C. (1994) "Modeling Sunseismic Faults by a Marked Point Process," presented at Fourth European Conference on the Mathematics of Oil Recovery, Roros, Norway.
39. Omre, H. (1991): "Stochastic Models for Reservoir Characterization," in Recent Advances in Improved Oil Recovery Methods for North Sea Sandstone Reservoirs, Kleppe, J. and Skjaeveland, S. M. (eds.), Norwegian Petroleum Directorate, Stavanger, Norway.
40. Picard W. R., Elfadel, I. M. and Pentland A. P. (1991): "Markov/Gibbs Texture Modeling: Aura Matrices and Temperature Effects," Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, p371-377, June.
41. Potaminanos, G. G. and Goutsias, J. K. (1995): " Partition Function Estimation of Gibbs Random Field Images Using Monte Carlo Simulations," IEEE Transactions on Information Theory, Vol. 39, No. 4, July.
42. Qian, W. and Titterington, D. M. (1989): "On the Use of Gibbs Markov chain Models in the Analysis of Images Based on Second-order Pairwise Interactive Distributions," Journal of Applied Statistics, Vol. 16, No. 2.
43. Raydan, M. (1997): "The Barzilai and Borwein Gradient Method for Large Scale Unconstrained Minimization Problem," SIAM J. on Optimization, Vol. 7, No. 1, p26-33, Feb.
44. Redner, R. A.: "Convergence Rates for Uniform B-spline Density Estimators—Part I: One-Dimension," Accepted for publication in SIAM Journal on Scientific Computing.
45. Ripley, D. D. (1987): "Stochastic Simulation," John Wiley and Sons Inc., 1987.
46. Rudkiewicz, J. L., Guerillot and Galli A. (1990): "An Integrated Software for Stochastic Modeling of Reservoir Lithology and Property with an Example from the Yorkshire Middle Jurassic," North Sea Oil and Gas Reservoirs -II, edited by A. T. Buller et al, Graham&Trotman, p399-406.
47. Stoyan, D. Kendall, W. S. and Mecke, J. (1987): "Stochastic Geometry and its Applications," John Wiley and Sons Inc.
48. Strauss, D. (1986): "On a General Class of Models for Interaction," SIAM Review, Vol. 28, No. 4, Dec.

49. Syversveen, A. R. and Omre, H. (1994): "Marked Point Models with Complex Conditioning Used for Modeling of Shales," Unpublished Manuscript.
50. Syversveen, A. R. and Omre, H. (1994): "Marked Point Models for Facies Units Conditioned on Well Data," Unpublished Manuscript.
51. Taylor, H. M. and Karlin, S. (1994): "An Introduction to Stochastic Modeling," Academic Press, San Diego.
52. Tjelmeland, H. (1996): "Modeling of the Spatial Facies Distribution by Markov Random Fields," presented at the Fifth International Geostatistics Congress, Wollongong, Australia, Sept. 22-27.
53. Tjelmeland, H. (1996): "Stochastic Models in Reservoir Characterization and Markov Random Fields for Compact Objects," Dr. ing. thesis, Norwegian University of Science and Technology, Trondheim, Norway.
54. Tjelmeland, H. and Holden, L. (1992): "Semi-Markov Random Fields," Geostatistics Troia, A. Soares (ed.) Kluwer Academic Publishers, Dordrecht, Vol. 1, p479-492.
55. Tjelmeland, H. (1996): "Modeling of the Spatial Facies Distribution by Markov Random Fields," Unpublished Manuscripts.
56. Tyler, K. J., Svanes, T. and Henriques, A. (1994): "Heterogeneity Modeling Used for Production Simulation of a Fluvial Reservoir," SPE Formation Evaluation, p85-92.
57. Weber, K. J. (1982): "Influence of Common Sedimentary Structures on Fluid Flow in Reservoir Models," JPT, p665-672.
58. Weber, K. J. and vanGeuns, L. C. (1990): "Framework for Constructing Classic Reservoir Simulation Models," JPT (Oct.), p1248-1257.
59. Yang, K. (1998): "3D Fluvial Facies Architecture Simulation of Middle Frio Formation, Stratton Field, South Texas," Master thesis at the University of Tulsa.
60. Younes, L. (1988): "Estimation and Annealing for Gibbsian Fields," Ann. Inst. Henri Poincare Probab. Statist, Vol 24, p269-294.



APPENDIX A

DESCRIPTION AND USE OF THE FORTRAN CODE FOR A 3D, SECOND-ORDER MARKOV RANDOM FIELD MODEL

In this appendix, we explain the implementation of a 3D, second-order MRF model, using FORTRAN. This model is used to generate facies distributions in three dimensions. The corresponding FORTRAN code can be compiled and built on any machine with a FORTRAN 77 compiler. But in order to visualize the realizations of the model by SPYGLASS SLICER, the output of the program, i.e., the realization of the MRF model, is written in SPYGLASS SLICER format.

The driver for the program is MRF_2D2N2C with the corresponding executable file. Both source code with a sample data file and include file are given on disk. An executable file with sample input file is given on the same disk.

In this model, the second-order neighborhood system and 2-pixel cliques are used. As described in section 5.4, there are nine types of cliques. We use 9 two-dimensional arrays, **beta1** through **beta9** to represent the potential values (parameters of the model) of the corresponding clique configurations in the corresponding directions. The dimensions of the arrays are all $K \times K$, K is the number of facies considered. With our notation, the physical meaning of the parameters is very clear. For example, if three facies labeled 1, 2 and 3 are considered, **beta1**(1,1) simply represents the potential value for a 2-pixel clique along the x-direction in the x-y plane, with both pixels in the clique occupied by facies 1; **beta5**(2,3) is the potential value for the vertical (z-direction) two-pixel clique with facies 2 in the "top" pixel and facies 3 in the "bottom" pixel. Therefore, **beta1**(1,1) controls the continuity of facies 1 in the x direction of the x-y plane and **beta5**(2,3) controls the transition between facies 2 and facies 3 in the vertical direction.

In this model, the free boundary treatment is applied. This means that the values of all the missing pixels of the boundary pixel are simply set to be zero and only the remaining clique potentials are considered. The total potential in a neighborhood of a pixel is calculated in subroutine POTENTIAL which calls the subroutine C_POTENTIAL to calculate the potential values of the two cliques in each direction.

All the input data are read by subroutine READ_DATA and transmitted by several COMMON BLOCKS to other routines.

Subroutine SET_FACIES is used to initialize the whole lattice by assigning a facies value randomly from K facies. The random number generator used in this routine is function RAN2 from Numerical Recipes in FORTRAN, the art of Scientific Computing by W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, Second edition, Cambridge University Press, 1992. However, any reliable random number generator can be substituted for this small subroutine.

The routine SIMULATION carries out the MCMC sampling procedure based on the initial distribution created by SET_FACIES and finally the OUTPUT subroutine writes a realization into a datafile which is named "imageXX". XX here is a two-digit number

from 01 to 10. This is designed to output multiple realizations from a long run of MCMC procedure. The maximum number of realizations is limited to 10 in the program and the iteration interval between realization is controlled by a parameter called "LAG". If more than 10 realizations are expected from the sampling procedure, the user can just modify the dimension of the array "FILENAMES" and the corresponding data statement.

In the code, the origin of the lattice is at the lower-left corner of the three-dimensional volume. The first index of the lattice (PIXEL) is for the x direction and from 1 to N_x , the second index for the y direction and varies from 1 to N_y , and the third index pertains to the z direction and varies from 1 to N_z .

The dimension of the 3D lattice as well as the number of facies are defined in "parameter.inc" which is included in the routines. The user can easily change the dimensions in this file and then rebuild the code conveniently.

Input Data of the Program

All the required information is set in a parameter file which is named by user with less than 40 characters. Each parameter in the file can be put in free format (separated by commas). A sample of the parameters is listed below with explanations.

"facies_3D.par"

```

Parameter for the 3D, MRF model /* title of the parameter file */

64,64,64      /* the number of pixels in x, y and z directions: nx, ny, nz */

500,200,100   /* the number of iterations, iterations for convergence
               and lag for multiple realizations */

-0.3,-0.5,0.5 /* two-pixel clique potentials for direction 1 */
0.5,-0.3,-0.5
-0.5,0.5,-0.3

-0.5,0.5,0.5 /* two-pixel clique potentials for direction 2 */
0.5,-0.5,0.5
0.5,0.5,-0.5

-0.3,-0.5,0.5 /* two-pixel clique potentials for direction 3 */
0.5,-0.3,-0.5
-0.5,0.5,-0.3

-0.3,-0.5,0.5 /* two-pixel clique potentials for direction 4 */
0.5,-0.3,-0.5
-0.5,0.5,-0.3

-0.3,-0.1,0.5 /* two-pixel clique potentials for direction 5 */
0.5,-0.3,-0.1
-0.1,0.5,-0.3

-0.6,0.5,0.5 /* two-pixel clique potentials for direction 6 */

```



```

0.5,-0.6,0.5
0.5,0.5,-0.6

-0.2,-0.5,0.5      /* two-pixel clique potentials for direction 7 */
0.5,-0.2,-0.5
-0.5,0.5,-0.2

-0.5,0.5,0.5      /* two-pixel clique potentials for direction 8 */
0.5,-0.5,0.5
0.5,0.5,-0.5

-0.5,0.5,0.5      /* two-pixel clique potentials for direction 9 */
0.5,-0.5,0.5
0.5,0.5,-0.5

1.0,2.0,3.0        /* facies values */
0.0,0.333334,0.666667,1.0 /* prior probability distribution of the facies*/

1                  /* apply penalty term(1) or not (0) */
3*1.0             /* relaxation factors for the facies */

1.0               /* temperature */

-45366           /* random seed (No longer than 6_digits) for random number generator */

```

A sample of file "parameter.inc" is listed below.

```

C definition of the lattice and the number of facies
  Parameter (nf=3,maxx=64,maxy=64,maxz=64)
C this is included in the program

```

In the above input, the data corresponding to "prior probability distribution of the facies" needs explanation. Even though there are only three facies considered, this data line contains four entries: 0.0,0.33334,0.66667,1.0. This indicates that according to the prior probability distribution, the probability of drawing facies 1 is $0.33334-0.0=0.33334$, the probability of drawing facies 2 is $0.66667-0.33334=0.33333$ and the probability of drawing facies 3 is $1.0-0.66667=0.33333$. Also the parameter labeled "temperature" is set to 1.0. If any value other than $T=1.0$ is used, all input potential values (the β 's) will be divided by the value of T.

Output of the program

The output of the program is a realization or multiple realizations. If $\text{iteration} = \text{nconverge}$, then only one realization named image01 is generated, while if $\text{iteration} > \text{nconverge}$, then multiple realizations with lag "LAG" are generated. These realizations can be visualized by SPYGLASS SLICER. Each realization is output to a separate file. These files are labeled as image01 to image10 if we generate 10 realizations. Each image file is stored as one-dimensional array for input to SPYGLASS SLICER. Each one-dimensional array is indexed first in the x-direction, then in the y-

direction and finally in the z-direction. Specifically, the facies occupying each pixel in the image files is written out as in the following pseudo code:

```
      Do 5 k=1,Nz
      Do 5 j=1, Ny
      Do 5 i=1, Nx
        Write(100,*) pixel(i,j,k)
5     Continue.
6
```

In the above example, the program will output 3 realizations with lag=100 and in SLICER. With the potential values given in the above data file, the realizations of the model will look like Fig. 5.4.15 in section 5.4. Subroutine READ_DATA will display all the input parameters on the screen for the user to check and also subroutine SET_FACIES will write the initial fractions of each facies on the screen.

Appendix B

The symbolic computation package Mathematica was used to simulate fluvial systems and in particular to generate channels in a fluvial reservoir. In support of this project a variety of small Mathematica Notebooks were developed. A larger more substantial Notebook was constructed for the final Frio reservoir simulation.

The following Notebooks are included with this report.

Density Estimation is a Notebook for the estimation of B-Spline nonparametric density estimates if there is no assumption of boundary.

Boundary Estimation is a Notebook for the estimation of B-Spline nonparametric density estimates if the data is assume to lie on an interval of the form $[a, \infty)$.

Two-Boundary Estimation is a Notebook for the estimation of B-Spline nonparametric density estimates if the data is assumed to be on an interval of the form $[a, b]$.

Random Generation is a Notebook that generates multivariate normal random variables and data from B-Spline density functions for the bounded, semi-infinite and unbounded cases.

Deconvolve is a Notebook which deconvolves the thickness data measured from wells and computes the maximum thickness histogram.

ClusterSim is a Notebook which generates clusters of channels in a fluvial reservoir.

It is easy to use each of these Mathematica Notebooks. Of course, to use a Mathematica Notebook, you must have access to the Mathematica program. Given that this is true, then you may begin to use any Mathematica Notebook by double clicking on the Notebook icon. This will open the Mathematica application and give you access to the Notebook. Most of these Notebooks have two sections. The first section contains fields for the user to enter their data. We will describe the input to each of the particular Notebooks later in this Appendix. Given that the input data has been entered, you can execute the entire Notebook by using the **kernel** pulldown menu, and selecting

Evaluation and then selecting **Evaluate Notebook**. The entire Notebook will now be evaluated and you may scroll through the Notebook to get the results. However, if just want to explore the Notebook, each Notebook has sample data included and is ready to execute. You can then enter you own data and rerun the Notebook when you are ready.

We will now describe the function, input and output of each of the Mathematica Notebooks that have been included in this report. Throughout this documentation, file names and variable names from Mathematica Notebooks will be in bold face type. We will begin our description of the individual Notebooks by discussing the three nonparametric density estimation Notebooks.

Density Estimation, Boundary Estimation and Two-Boundary Estimation Notebooks

These three Notebooks are used in almost exactly the same way. In each case, the nonparametric density estimate is determined by a vector of data and four parameters. The specific input is

- data** A list of data points separated by commas and enclosed in curly braces.
- m** The order of the spline. A value of $m=3$ for example would give you quadratic splines.
- amin** and **bmax** The density estimate will have support on the interval **[amin,bmax]**. These values should be chosen so that all of the data lies in this interval.
- N1** The number of subintervals into which the interval **[amin,bmax]** is subdivided into to create the B-Spline density estimate.

For example, the following input will generate a piecewise linear nonparametric density estimate for the values in the list **data** on the interval **[0,10]** and this interval will be divided into intervals of length 2.5.

```
m=2;  
data={2,3,4,4,5,6}  
amin=0  
bmax=10  
N1=4
```

The output from the Notebook includes:

1. The mean and the standard deviation of the data.
2. The mean and standard deviation for the estimated B-Spline density function.
3. A graph of the unweighted basis functions $B_i(x)$ for each i .
4. A graph of the weighted basis functions $\alpha_i B_i(x)/b_i$ for each i .
5. A graph of the final B-Spline density function.
6. Values of α_i for each index i and the number of basis functions.

If you use the Notebook **Density Estimation**, your estimate will tail off to zero outside of the interval [**amin**,**bmax**]. If you use the Notebook **Boundary Estimation** your estimate will be truncated at the point **amin** and tail off to zero to the right of **bmax**. If you use the Notebook **Two-Boundary Estimation** the estimator will be truncated at **amin** and **bmax**.

Random Generation

This Mathematica Notebook contains routines for the generation of multivariate normal random variables and data from B-Spline density functions for the bounded, semi-infinite and unbounded cases. There is no input required to use these routines. Simply execute the Notebook and you will be able to use the newly defined functions to generate random variables.

In the first half of the Notebook we have the routines associated with the generation of multivariate normal random variables. To generate an n -dimensional multivariate normally distributed random variable with mean **mu** and covariance matrix **sigma** enter **mu** as a list of n values and **sigma** as a list of lists of the rows of your covariance matrix. The following statement will then return a n -dimensional multivariate normally distributed random variable with mean **mu** and covariance matrix **sigma**.

Normalvar[mu**,**sigma**]**

This routine will perform the necessary decomposition of the matrix **sigma** needed to for the generation of multivariate data. However, this decomposition take a small amount of computing resources and so this function is inefficient if you need to generate many data point. Two additional functions have been added to solve this problem. The first function generates the Cholesky factor needed in the generation of multivariate normal data. The second function uses this factor to efficiently generate data. To determine the Cholesky factor of **sigma** type (you will only need to do this once)

```
rootsigma=root[sigma]
```

Then use

```
Normalwithroot[mu,rootsigma]
```

each time that you wish to generate a random variable with mean **mu** and covariance matrix **sigma**.

For example you could generate a table of 50 independent multivariate normally distributed data points with mean (1,2,3) and covariance matrix with 1, 10 and 100 down the diagonal by executing

```
mu={1,2,3}
```

```
sigma=DiagonalMatrix[{1,10,100}]
```

```
data=Table[Normalvar[mu,sigma] ,{50}]
```

To generate the same data more efficiently use

```
rootsigma=root[sigma]
```

```
data=Table[Normalwithroot[mu,rootsigma] ,{50}]
```

You will find examples within the Notebook and Mathematica expressions for testing that everything is working correctly.

The second half of this Notebook contains the functions that can be used to generate data from a B-Spline nonparametric density function. If you execute the entire Notebook you will be able to use all of the functions defined by the Notebook.

The SplineDistribution function allows to you define a B-Spline distribution by name. You may give the distribution any name that you like and then you generate data from this distribution by using that name. So to define a B-Spline distribution enter

SplineDistribution[alpha,x1,h,m,"name"]

where

alpha is a list of proportions (a list of number that is nonnegative and sums to 1,

x1 is the minimum value of the support of the density function,

h is the distance between knots (must be positive),

m is the order of the spline (integer greater than 0) and

name is the name of the distribution.

For example if you enter

```
data={2/10,3/10,3/10,2/10};
```

```
SplineDistribution[data,0,30,2,"testname"]
```

you will have defined a B-Spline density function which is linear (order $m=2$), which starts at zero, has a bin width of $h=30$ and is named **testname**. This named distribution function can be interpreted as B-Spline density functions on an unbounded domain, a semi-infinite domain (left truncated estimator) and a bounded domain (truncated estimator). To generate a random value from each of these three types of distributions by entering

```
makeRpoint["testname"]
```

```
makeRpointLT["testname"]
```

```
makeRpointT["testname"]
```

The letters LT and T in these names stand for left truncated and truncated respectively.

Examples are included in the Notebook.

Deconvolve

This Notebook can be used to deconvolve thickness data measured from well and to compute the maximum thickness histogram. The input to the Mathematica Notebook is a description of the histogram. This data consists of the following values.

Sand is a list of the sand thickness values from the histogram.

Sandmin is the leftmost point on the histogram.

SandDelta is the width of the sand thickness histogram bins.

The following data is the input data for the Frio Sands reservoir from this report

Sand={2,14,34,46,38,35,13,6,1};

Sandmin=0;

SandDelta=4;

The output from the Notebook is a list of values for the new histogram and a plot of the new and old histogram.

ClusterSim

This *Mathematica* Notebook titled ClusterSim generates well separated clusters of channels in a reservoir. Cluster centers are generated using an MCMC algorithm to insure that the cluster centers are well separated. Each cluster of channels is assumed to follow a multivariate normal distribution. The input into the first portion of the Notebook is

thewidth the width of the reservoir

theheight the height of the reservoir

chlength the length of each section of the channel

maxnpoints the maximum number of clusters of channels

clustersize the average size of clusters of channels

sigma1 the standard deviation of an individual cluster in the horizontal direction.

sigma2 the standard deviation of an individual cluster in the vertical direction.

- a** controls the penalty for having clusters overlap, increasing a make it less probable that two cluster center will be close
- b** controls the separation of cluster centers, increasing b generates points which are further apart
- numiter** the number of iterations of the MCMC algorithm. One iteration is one pass through the data.

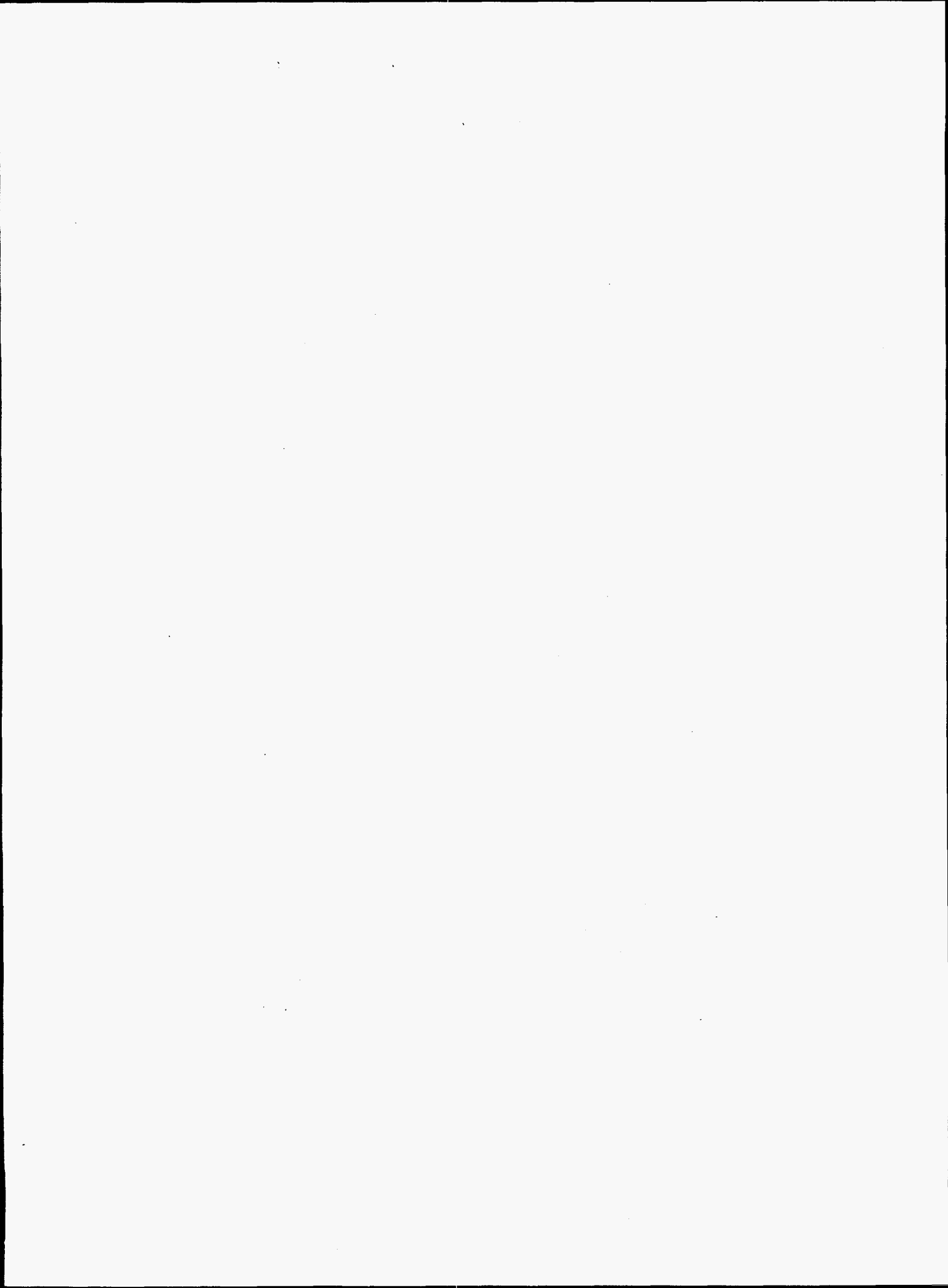
The second section of the input portion of the Notebook contains information, which specifies the thickness, width to thickness and angle distributions. For each distribution you must enter in the proportions which defines the B-Spline density function and initiate the SplineDistribution function, which allows you to use a named B-Spline throughout the Notebook. For example, to define the angle distribution with proportions (.3, .5, .15, .05) starting with a minimum angle of zero with 30 degree intervals and named **angle** you need to enter the following statements.

```
angle={.3,.5,.15,.05}
```

```
SplineDistribution[angle,0,30,2,"angle"]
```

The thickness distribution and the width to thickness distributions are defined in exactly the same way. All of these are already in the Notebook and you can easily change the values if you have different distribution that you want to use. After you have entered your data, simply execute the entire Notebooks and scroll down to see your results at the end of the Notebook.

The outputs of the ClusterSim program are images of the clusters of channels from an oblique view, a side view and a top view. Mathematica allows you to interactively rescale the images and to change the view.



PART VI

ESTIMATION OF GEOLOGICAL ARCHITECTURE USING BAYESIAN/MAXIMUM ENTROPY APPROACH

By: Ruijian Li, Mohan Kelkar, and Erdal Ozkan

Summary

This project investigates the use of the Bayesian Maximum Entropy (BME) method to generate geological facies as part of the reservoir description. The advantages of the proposed method over the conventional approaches are first, the method can account for non-stationarity in geological properties and second, it is general enough so that various constraints can be accommodated as part of the description.

During the first year of the project, we developed a generalized procedure, which accounts for the presence of seismic data as part of the geological description. These functions include variogram, covariances, and cross-covariances among the neighboring values. The method can also account for local probability distribution of facies, which can be obtained from the seismic information. The results indicate that the use of seismic data can improve the facies estimation compared to using simple kriging.

In the second year, we examined the feasibility of extending the method for multipoint connectivity function. Our studies in the second year comprised three main investigations. First we modeled higher order moments. We showed that the higher order moments bear similar characteristics to that of the second order moments. Second, based on a nonlinear regression scheme, we developed an independent means of incorporating multi-point connection into reservoir architecture modeling. This proves to be a robust procedure and improves our ability to estimate the reservoir architecture. It also provides the basis for checking the results obtained by the BME method. In the final phase of the study, we applied the BME method to the four-point connectivity case and showed the improvements of reservoir architecture modeling and thus the potential of the BME method.

Introduction

Geological description using quantitative methods is becoming increasingly popular. The two approaches commonly used in describing the geological facies are the grid based and object based methods.

The grid-based methods generate geological facies on a grid block level so that the information can be directly used as input in the reservoir simulation. The two grid based methods commonly used are sequential indicator simulation and sequential

Gaussian simulation. Both methods essentially follow the same approach in generating the geological facies at the grid block location. In the first step, the original data (categorical facies data) are transformed into quantitative form. In the second step, various continuity measures are estimated in the transformed domain. Typically these measures are variogram and co-variances. In the third step, unsampled locations are visited in a random sequence, and the value of the transformed variable is estimated using the sample values in a search neighborhood as well as prior estimated values in the same neighborhood. Once all the unsampled points are visited, the estimated values are back transformed into categorical facies data. The main difference between the two methods is the type of transform used in transforming the original categorical data into the transformed data. Indicator simulation transforms the data into indicator variables (discrete variables), whereas Gaussian simulation transforms the data into Gaussian variables (continuous variables).

Both these methods have been extensively applied in the literature with varying degree of success. The main drawbacks of these methods are: they do not account for non-stationarity as a function of spatial locations, they do not account for relationships which extend beyond two point connectivity functions, and they do not account for geological rules such as physical relationships between facies (e.g., splay is attached to the channel) explicitly. We need to develop methods that can overcome some of these drawbacks.

Approach

The overall approach of this project is to transform the facies data into indicator domain, develop various constraints, and develop a procedure that will allow the estimation of facies at the unsampled location. We briefly describe the steps below. Detailed derivations regarding the method are included in the Appendix.

Indicator Formalism

We propose to use indicator function to define the facies. The indicator function can be defined as:

$$I(\mathbf{u}, K_t) = \begin{cases} 1 & \text{if } K(\mathbf{u}) = K_t \\ 0 & \text{if } K(\mathbf{u}) \neq K_t \end{cases} \quad (6.1)$$

Where K_t is a threshold facies, $K(\mathbf{u})$ is the facies at a location \mathbf{u} . This definition is flexible enough to accommodate both the hard (knowledge with 100 % certainty) and the soft (knowledge with less than 100% confidence) information. For example, by assigning an indicator value between 0 and 1, we can represent the probability that a certain facies is present at a particular location. Further, by using indicator variables to describe the geological facies, we are able to capture different spatial characteristics for different facies.

Similar to any other variable, in the case that the information about spatial variability is available, we can estimate the connectivity between sample points. For example, variogram can be defined as,

$$\hat{\gamma}_I(h, K_t) = \frac{1}{2n(h)} \sum_{j=1}^{n(h)} [i(u_j, K_t) - i(u_{j+h}, K_t)]^2 \quad (6.2)$$

Cross variogram can be defined as,

$$\hat{\gamma}_I(h; K_t, K_t') = \frac{1}{2n(h)} \sum_{j=1}^{n(h)} [i(u_j, K_t) - i(u_{j+h}, K_t)][i(u_j, K_t') - i(u_{j+h}, K_t')] \quad (6.3)$$

Where, $\hat{\gamma}_I$ is the variogram, h is the lag distance, and $n(h)$ is the number of pairs at lag distance h . In addition, using the indicator formulation, we can also describe multi-point connectivity. For example,

$$\hat{H}_I(h_1, \dots, h_M; K_{t1}, \dots, K_{tM}) = \frac{1}{n} \sum_{l=1}^n \left[\prod_{j=1}^M i(u_l + h_j, K_{tj}) \right] \quad (6.4)$$

Where, H_I represents multi-point histogram connecting M neighboring values. This type of connectivity can capture geological objects, which have unique shapes.

Bayesian/Maximum Entropy (BME) Formalism

The BME approach is based on three basic principles. These principles can be stated as:

- The information is more valuable when the uncertainty prior to obtaining that information is bigger. For example, if the probability that event A will occur is 90%, the fact that event A has occurred is not as valuable. Compare that with the probability that event A will occur is only 5%. If the event A occurs, then that information is much more valuable.
- The posterior probability of an event is obtained by using Bayes' rule. Bayes' rule relates the prior probability to the posterior probability of an event.
- The best estimate of the posterior probability is obtained by maximizing the information.

In applying BME approach, we will assume that we are interested in estimating $I(u_0, K_t)$ at the unsampled location u_0 given sampled values $i(u_j, K_t)$, $j = 1, 2, \dots, n$.

Let $\hat{I}(u_0, K_t)$ be the estimator of $I(u_0, K_t)$ at an unsampled location u_0 where no

observation is available. If $f_i(u_0, u_1, u_2, \dots, u_n)$ is the joint probability density function of the associated random variables $I(u_j, K_t)$, $j = 1, 2, \dots, n$, prior to observing the data $I(u_j, K_t) = i(u_j, K_t)$, the fundamental probability density constraint needs to be satisfied.

$$\int \int \dots \int_{n+1 \text{ times}} f_i(u_0, u_1, u_2, \dots, u_n) di_0 di_1 \dots di_n = 1 \quad (6.5)$$

The information contained in the random variables is assumed to be measured by.

$$\inf[I(u_j, K_t), j = 0, 1, 2, \dots, n] = -\ln[f_i(u_0, u_1, u_2, \dots, u_n)] \quad (6.6)$$

This definition of the information is consistent with the first principle, which states that the information is more valuable when the probability of that event occurring is small. As the probability increases, the negative log normal value becomes smaller. The expected information can be written as,

$$E\{\inf[I(u_j, K_t), j = 0, 1, 2, \dots, n]\} = E\{-\ln[f_i(u_0, u_1, u_2, \dots, u_n)]\} = \varepsilon(f_x) \quad (6.7)$$

where $\varepsilon(f_x)$ is Shannon entropy function .

The prior physical constraints can be expressed mathematically as

$$E(g_q) = \int \int \dots \int_{n+1 \text{ times}} g_q(u_0, u_1, u_2, \dots, u_n) f_i(u_0, u_1, u_2, \dots, u_n) di_0 di_1 \dots di_n \quad (6.8)$$

where $g_q(u_0, u_1, u_2, \dots, u_n)$, $q = 1, 2, \dots, Q$, are suitable functions of $I(u, K_t)$.

For convenience, we define $g_0(u_0, u_1, u_2, \dots, u_n) = 1$ so that $E[g_0] = 1$ defines the normalization constraint defined in Eq. 6.5.

The function g can take various forms. For example, in the presence of observed values, we can write,

$$g_q(u_j) = i(u_j, K_t) \quad (6.9)$$

where $j = 0, 1, 2, \dots, n$; $q = 1, 2, \dots, m+1$.

another example would be a bivariate relationship,

$$g_q(\mathbf{u}_j, \mathbf{u}_k) = [i(\mathbf{u}_j, K_i) - E\{I(\mathbf{u}_j, K_i)\}][i(\mathbf{u}_k, K_i) - E\{I(\mathbf{u}_k, K_i)\}] \quad (6.10)$$

where $i, j=0, 1, 2, \dots, n$; $q=n+2, n+3, \dots, (n+1)(n+4)/2$.

The prior probability $f_i(\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n)$ and the posterior probability $f_i^*(\mathbf{u}_0 | \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n)$ are related by Bayes' Law,

$$f_i^*(\mathbf{u}_0 | \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n) = \frac{f_i(\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n)}{f_i(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n)} \quad (6.11)$$

We can define BME (Bayesian Maximum Entropy) function as,

$$\begin{aligned} B_i(\mathbf{u}_0) &= \ln[f_i^*(\mathbf{u}_0 | \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n)] \\ &= \ln[f_i(\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n)] - \ln[f_i(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n)] \end{aligned} \quad (6.12)$$

This function has to be maximized to get the desired benefit. Specifically, in BME approach, the estimated value $K(\mathbf{u}_0)$ is the solution of the equation

$$\left. \frac{d \ln\{f_k(K(\mathbf{u}_0), K(\mathbf{u}_1), \dots, K(\mathbf{u}_n))\}}{dK(\mathbf{u}_0)} \right|_{K(\mathbf{u}_0)=\hat{K}(\mathbf{u}_0)} = 0 \quad (6.13)$$

where the joint probability density function is defined as:

$$f_k(K(\mathbf{u}_0), K(\mathbf{u}_1), \dots, K(\mathbf{u}_n)) = \frac{1}{A} \exp\left[\sum_{q=0}^Q \mu_q g_q(K(\mathbf{u}_0), K(\mathbf{u}_1), \dots, K(\mathbf{u}_n))\right] \quad (6.14)$$

and the LaGrange multipliers μ_q , $q = 0, 1, \dots, Q$ can be obtained by solving

$$\begin{aligned} E[g_q(K(\mathbf{u}_0), K(\mathbf{u}_1), \dots, K(\mathbf{u}_n))] &= \\ \frac{1}{A} \int \dots \int_{n+1 \text{ times}} g_q(K(\mathbf{u}_0), K(\mathbf{u}_1), \dots, K(\mathbf{u}_n)) &\times \exp\left[\sum_{q=0}^Q \mu_q g_q(K(\mathbf{u}_0), K(\mathbf{u}_1), \dots, K(\mathbf{u}_n))\right] dK(\mathbf{u}_0) dK(\mathbf{u}_1) \dots dK(\mathbf{u}_n) \\ q = 0, 1, \dots, Q \quad (Q = (n+1)(n+4)/2) & \end{aligned} \quad (6.15)$$

where g_q are the constraints imposed on the estimation equation. Depending on the types of constraints imposed, Eq.15 can be solved analytically. For two-point connectivity case, for example, knowing co-variances, analytical solution of Eq. 6.15 is possible. For higher order connectivities, on the other hand, solution procedure must be numerical. Appendices 6.A and 6.B provide the details of the analytical and numerical solution procedures for two-point and multi-point connectivity cases, respectively.

An Application Example for Two-Point Connectivity Information

To illustrate the application of the BME method, here we consider an example of generating geological facies in two dimensions (details of the computational procedure can be found in Appendix 6.A). Suppose we know the means and covariance at points 1 and 2, and we want to estimate the facies value at the point 0 as shown in Fig. 6.1.

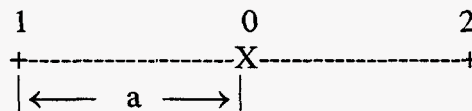


Fig. 6.1 Location of Sampled and Unsampled Points for the Example

In this example, we assume that we have seismic data, which provides us with local probability density function. That is, given a seismic attribute value, we assume that the probability of individual facies is known. This is a prior probability information, which is location dependent or non-stationary. We estimate the posterior probability density function using the spatial relationships as an additional information. Following the procedure derived in Appendix 6.A, the computational procedure is as follows:

Let $E[K]=0$, assume that the covariance is only the function of distance. From Eq. 6.A.37 of Appendix 6.A, we construct the following matrix:

$$B^{-1} = \begin{vmatrix} c(0) & c(a) & c(a) \\ c(a) & c(0) & c(2a) \\ c(a) & c(2a) & c(0) \end{vmatrix} \quad (6.16)$$

Inverting the matrix in Eq. 6.16, we obtain

$$B = \frac{1}{[c(0) - c(2a)][2c(a)^2 - c(0)^2 - c(0)c(2a)]} \times \begin{vmatrix} c(2a)^2 - c(0)^2 & c(0)c(a) - c(2a)c(a) & c(0)c(a) - c(2a)c(a) \\ c(0)c(a) - c(2a)c(a) & c(a)^2 - c(0)^2 & c(0)c(2a) - c(a)^2 \\ c(0)c(a) - c(2a)c(a) & c(0)c(2a) - c(a)^2 & c(a)^2 - c(0)^2 \end{vmatrix} \quad (6.17)$$

The entries of the matrix in Eq. 6.17 correspond to the negative of the LaGrange multipliers (see Eq 6.A.11). Thus, using Eq. 6.A.41, we have

$$\begin{aligned} \hat{K}(\mathbf{u}_0) &= m(\mathbf{u}_0) - \frac{1}{\mu_{00}} \sum_{i=1}^2 \mu_{i0} (K(\mathbf{u}_i) - m(\mathbf{u}_i)) \\ &= -\frac{1}{c(2a)^2 - c(0)^2} [(c(0)c(a) - c(2a)c(a))K(\mathbf{u}_1) + (c(0)c(a) - c(2a)c(a))K(\mathbf{u}_2)] \\ &= \frac{c(a)}{c(2a) + c(0)} (K(\mathbf{u}_1) + K(\mathbf{u}_2)) \end{aligned} \quad (6.18)$$

Using the procedure as described in Appendix 6.A, we compared the results of facies description under different conditions. In this case, we assumed that we had seismic data available at each location that provided us with prior probability of facies. We also assumed that we had actual facies information at limited number of wells. Fig. 6.2 shows the geological facies description using simple kriging method. In this figure, we assumed that only well data are available. Fig. 6.3 shows the geological description using only seismic attribute information. Fig. 6.4 shows the geological facies description using the well data as well as the seismic information using the procedure described in Appendix 6.A. As can be seen, there is a significant improvement in the geological description by incorporating the seismic data. Although not obvious, incorporation of both seismic and well data matches with the reference data set much better as compared to using only seismic or well data.

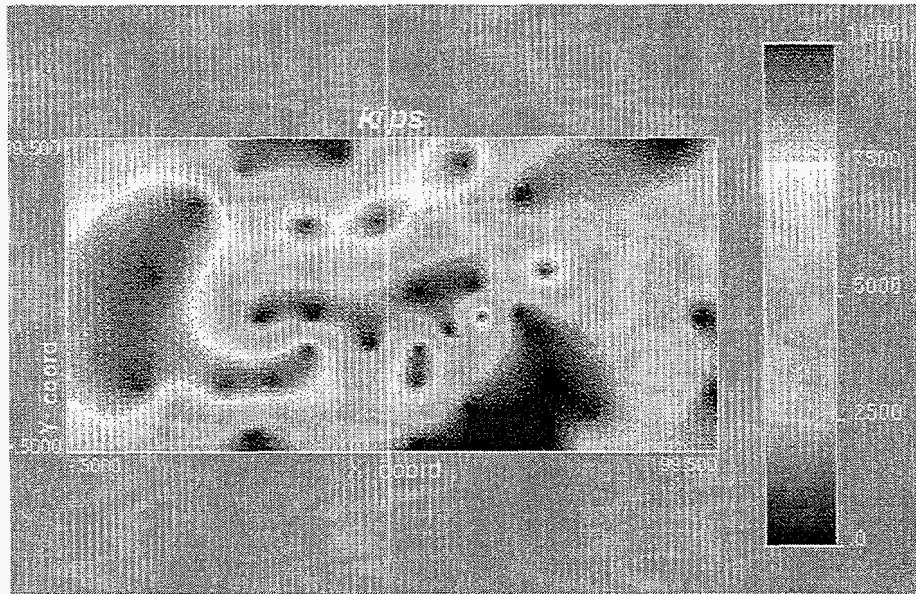


Fig. 6.2 Facies Distribution Estimated by Well Data

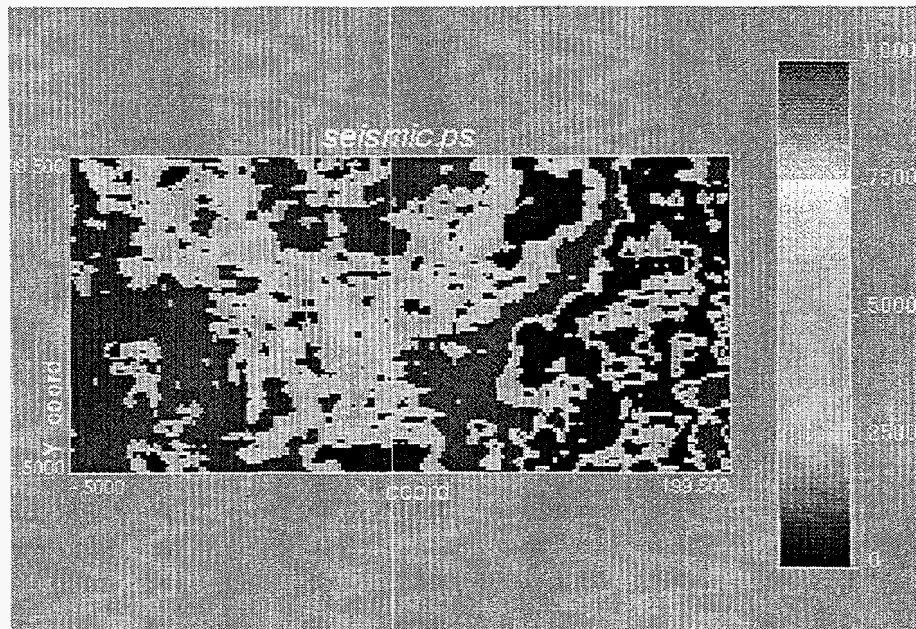


Fig. 6.3 Facies Distribution Estimated by Seismic Data



Fig. 6.4 Facies Distribution Estimated by Using Both Well and Seismic Data

High Order Moments

As we noted before, the advantage of the BME procedure is to consider multi-point connectivity information. The use of multi-point connectivity information in the area of geostatistics is new and practically no information about the modeling of high order moments is available. The correct usage of multi-point connectivity functions is restricted to computation of probabilities of various configurations based on training or a reference image. Since the number of combinations increase very rapidly as the number of points increase, the method is typically restricted to 3 or 4 nearest neighbors. This type of connectivity, however, is difficult to implement in practice because training image is rarely available. Instead, we wanted to develop a procedure that can capture multi-point connectivity based on well data or seismic data. Therefore, we devoted some of this study to modeling high order moments.

The basic advantages of multi-point connectivity functions are the following:

- 1) They provide flexibility in defining dis-jointed connectivity
- 2) They can be used with limited data using lag tolerance
- 3) Direction dependence can be incorporated
- 4) Connectivity between different facies can be integrated

The multi-point connectivity functions can be defined similar to the conventional two-point connectivity functions. The multi-point histogram representing the multi-point probability function, for example, is defined as follows:

$$H_I(\bar{h}_1, \dots, \bar{h}_n; K_{t_1}, \dots, K_{t_n}) = \frac{1}{m} \sum_{j=1}^m i(\bar{h}_{1_j}, K_{t_1}) \dots i(\bar{h}_{n_j}, K_{t_n}) \quad (6.19)$$

Similarly, the multivariate is defined by

$$C_I(\bar{h}_1, \dots, \bar{h}_n; K_t) = \frac{1}{m} \sum_{j=1}^m [i(\bar{h}_{1_j}, K_t) - m(K_t)] \dots [i(\bar{h}_{n_j}, K_t) - m(K_t)] \quad (6.20)$$

For example, the tri-variance can be defined as follows

$$C_I(\bar{h}_1, \bar{h}_2, \bar{h}_3; K_t) = \frac{1}{m} \sum_{j=1}^m [i(\bar{h}_{1_j}, K_t) - m(K_t)] [i(\bar{h}_{2_j}, K_t) - m(K_t)] [i(\bar{h}_{3_j}, K_t) - m(K_t)] \quad (6.21)$$

We have used these definitions of multi-point connectivity functions for the training image shown in Fig. 6.5. This image represents probability of Facies 1 based on seismic data. 1 indicates 100% probability that Facies 1 is present. 0 indicates 100% probability that Facies 1 is absent. A value in between indicates probability of that facies being present. Figs. 6.6 and 6.7 show the tri-variance and quadri-variance functions for this image, respectively. As can be seen from these figures, multi-point connectivity functions display similar characteristics as two-point connectivity functions at a higher dimension.

The functions are well behaved and are consistent with an intuitive understanding of connectivities. For example, at a given lag distance between two points, as the distance of the third point increases, the multi-variance is reduced indicating diminishing relationship. Beyond the range of correlation, the multi-variance is zero indicating uncorrelated nature. This range is the same as observed for two-point connectivity. This is consistent with the idea that if two points are uncorrelated beyond certain lag distance, three points will also be uncorrelated beyond that distance. Also observed is that the initial value of multi-variance is smaller as the distance between the first two points decreases. This is expected since with increasing distance, the relationship between these points should weaken indicating a smaller value of multi-variance.

In practice, the data usually come from the well information. In this case, it is very convenient to have appropriate models to represent the multi-point connectivity functions such as the ones shown in Figs. 6.6 and 6.7. We have tried the spherical, exponential, and Gaussian models for the quadri-variance functions shown in Fig. 6.7 and have presented the results in Figs. 6.8-6.10, respectively. The spherical and exponential models match the actual functions fairly well. To be used in practical applications, however, both models will require more improvements.

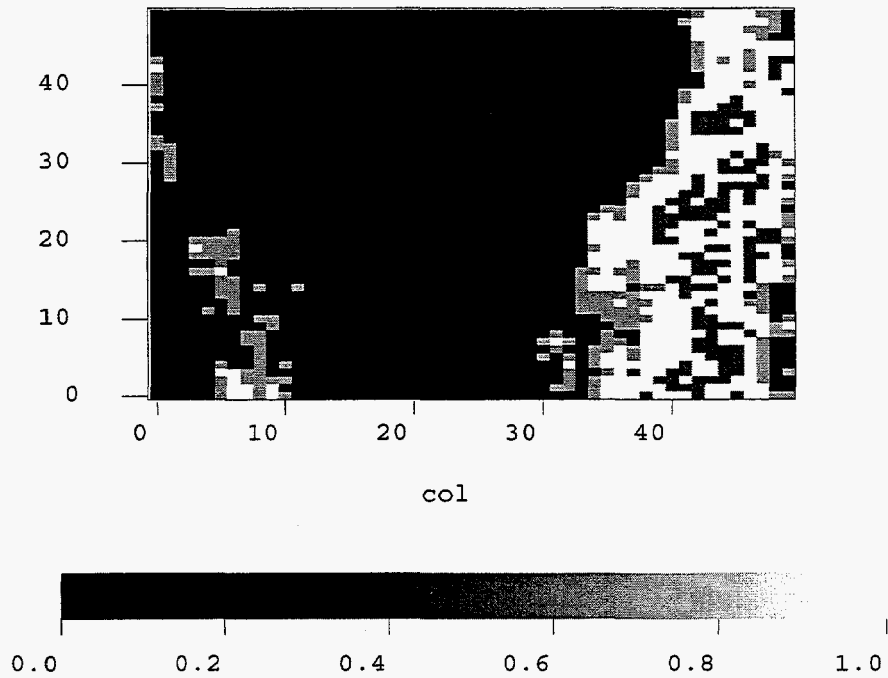


Fig. 6.5 The Training Image

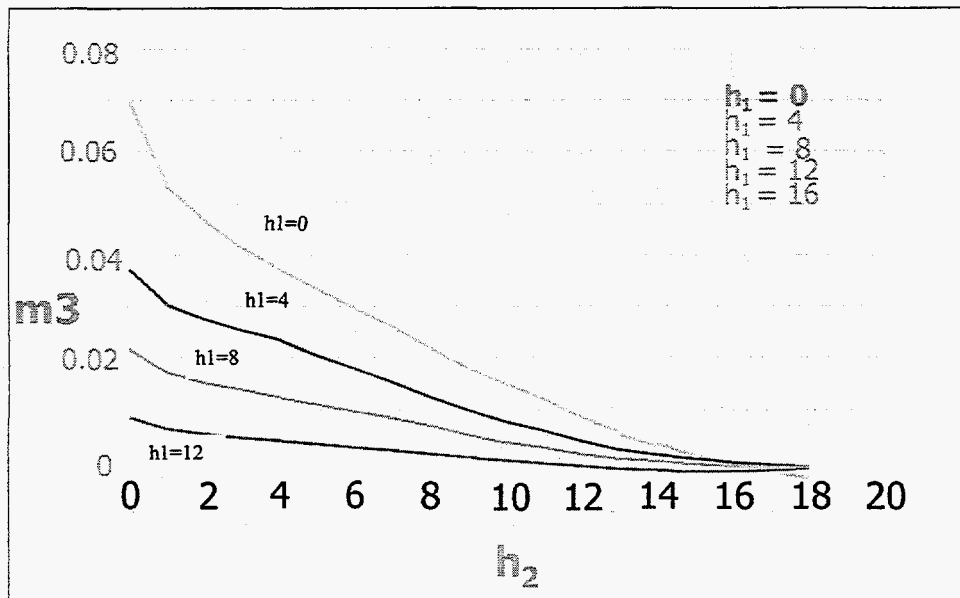


Fig. 6.6 Example of Tri-Variance

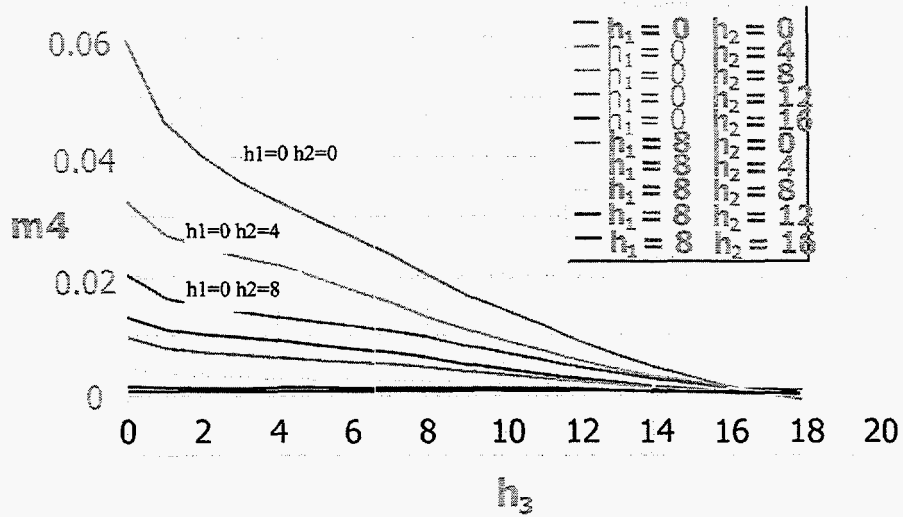


Fig. 6.7 Example of Quadri-Variance

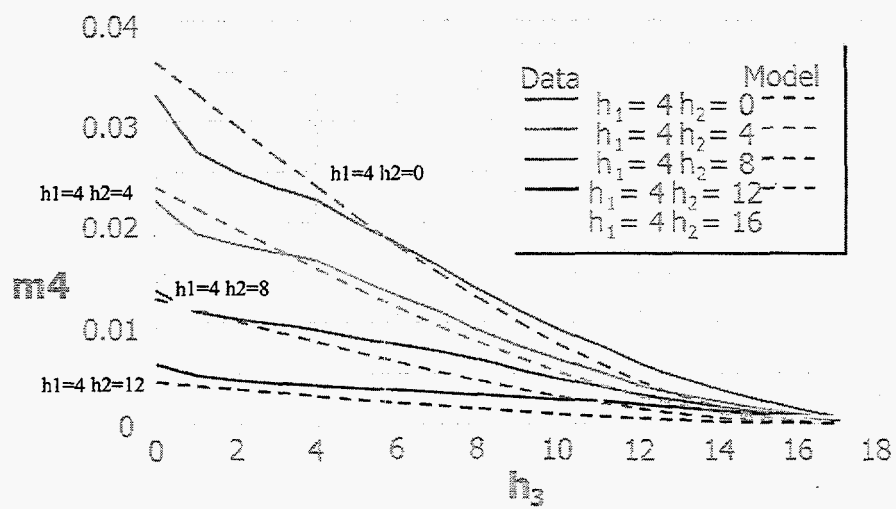


Fig. 6.8 Spherical Model for Quadri-Variance Function

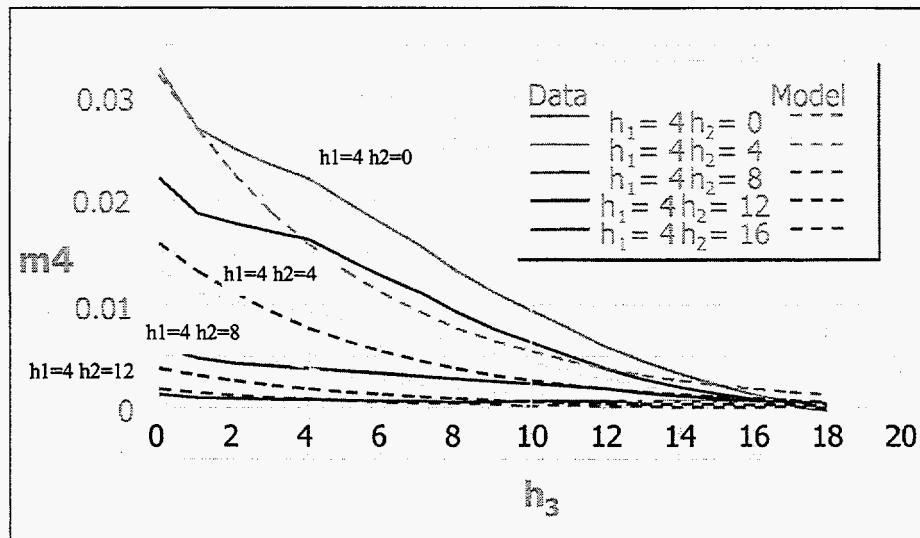


Fig. 6.9 Exponential Model for Quadri-Variance Function

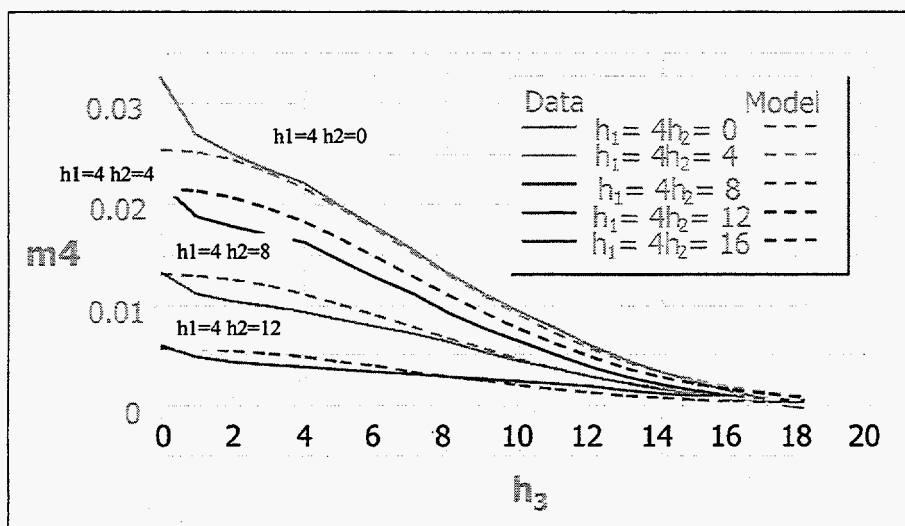


Fig. 6.10 Gaussian Model for Quadri-Variance Function

A Nonlinear Estimation Procedure

In this project, we have concentrated on defining the complex constraints that would provide a better description of the reservoir geology. We have investigated the high order moment methods and developed a semi-analytical method to incorporate multi-point connectivity information as opposed to the conventional two-point connectivity information. Here, we present a method that uses the fourth moments in describing the reservoir architecture. Although this method is useful in itself, it will also

provide the check for the numerical BME method we developed before and will help better understand the type, quality, and the quantity of the data required to describe the complex constraints based on multi-point connectivity information.

The Fourth Moment

The conventional methods use two-point connectivity information to describe the reservoir architecture. Our objective here is to investigate the possibility of incorporating multi-point connectivity information in reservoir architecture modeling. High order moments have been used in the literature to incorporate multi-point connectivity information (see, for example, Stuart and Ord, 1987). In the general framework of moments, two-point connectivity information corresponds to the second moment. Below we use the fourth moment.

The property of the fourth moment can be described by the following expression:

$$M_4(X - Y) = M_4(X) + M_4(Y) - 4M_{X^3Y} + 6M_{X^2Y^2} - 4M_{XY^3} \quad (6.22)$$

where

$$\begin{aligned} \mu_X &= E[X] \\ \mu_Y &= E[Y] \\ M_4(X) &= E[(X - \mu_X)^4] \\ M_{X^3Y} &= E[(X - \mu_X)^3(Y - \mu_Y)] \\ M_{X^2Y^2} &= E[(X - \mu_X)^2(Y - \mu_Y)^2] \\ M_{XY^3} &= E[(X - \mu_X)(Y - \mu_Y)^3] \\ M_4(Y) &= E[(Y - \mu_Y)^4] \end{aligned}$$

We will use the preceding property of the fourth moments in the estimation of the unknown value of X_0 from the known values of X_1, X_2, \dots, X_n .

Let

$$\hat{X}_0 = \mu_0 + \sum_{i=1}^n \lambda_i (X_i - \mu_i) \quad (6.23)$$

This equation is similar to conventional kriging equation. In this equation, we assume that the estimated value at the unsampled location is linearly related to nearby samples.

The fourth moments of the difference between the true value, X_0 , and the estimated value,

\hat{X}_0 is:

$$\begin{aligned}
\varepsilon &= M_4(\hat{X}_0 - X_0) = M_4\left(\sum_{i=1}^n \lambda_i(X_i - \mu_i) - (X_0 - \mu_0)\right) \\
&= M_4\left(\sum_{i=1}^n \lambda_i(X_i - \mu_i)\right) - 4E\left[\left(\sum_{i=1}^n \lambda_i(X_i - \mu_i)\right)^3(X_0 - \mu_0)\right] \\
&\quad + 6E\left[\left(\sum_{i=1}^n \lambda_i(X_i - \mu_i)\right)^2(X_0 - \mu_0)^2\right] - 4E\left[\left(\sum_{i=1}^n \lambda_i(X_i - \mu_i)\right)(X_0 - \mu_0)^3\right] \\
&\quad + M_4(X_0 - \mu_0) \\
&= \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n \lambda_i \lambda_j \lambda_k \lambda_l M_{ijkl} - 4 \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \lambda_i \lambda_j \lambda_k M_{ijk0} + 6 \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j M_{ij00} \\
&\quad - 4 \sum_{i=1}^n \lambda_i M_{i000} + M_4(X_0 - \mu_0)
\end{aligned} \tag{6.24}$$

where

$$M_{ijkl} = E[(X_i - \mu_i)(X_j - \mu_j)(X_k - \mu_k)(X_l - \mu_l)].$$

Similar to kriging, minimizing fourth moment (Eq. 6.24), we have

$$\frac{\partial \varepsilon}{\partial \lambda_i} = 4 \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n \lambda_j \lambda_k \lambda_l M_{ijkl} - 12 \sum_{j=1}^n \sum_{k=1}^n \lambda_j \lambda_k M_{ijk0} + 12 \sum_{j=1}^n \lambda_j M_{ij00} - 4 M_{i000} = 0 \tag{6.25}$$

or

$$\sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n \lambda_j \lambda_k \lambda_l M_{ijkl} - 3 \sum_{j=1}^n \sum_{k=1}^n \lambda_j \lambda_k M_{ijk0} + 3 \sum_{j=1}^n \lambda_j M_{ij00} - M_{i000} = 0; \text{ for } i=1,2,\dots,n \tag{6.26}$$

The solution of Eq. 6.26 provides the coefficients λ_i for $i=1,2,\dots,n$. Then, the estimated value of \hat{X}_0 can be obtained from Eq. 6.23. Note that Eq. 6.26 is a nonlinear equation and an iterative procedure is required to solve it. We have used Newton's method to solve Eq. 6.26. The details of the iterative solution by Newton's method are presented in Appendix 6.C. It must be emphasized that because we use multi-point connectivity information (higher order moments), the results estimated by this procedure should be better than those obtained by the procedures that use two-point connectivity information. A manual for the computer program, describing the procedure, is included in Appendix 6.D.

Application Examples for Four-Point Connectivity Information

Here we consider examples of generating geological facies in two dimensions by using the nonlinear regression method described above. For the first example, we use the training image shown in Fig. 6.5 and the four point connectivity functions shown in Fig. 6.7.

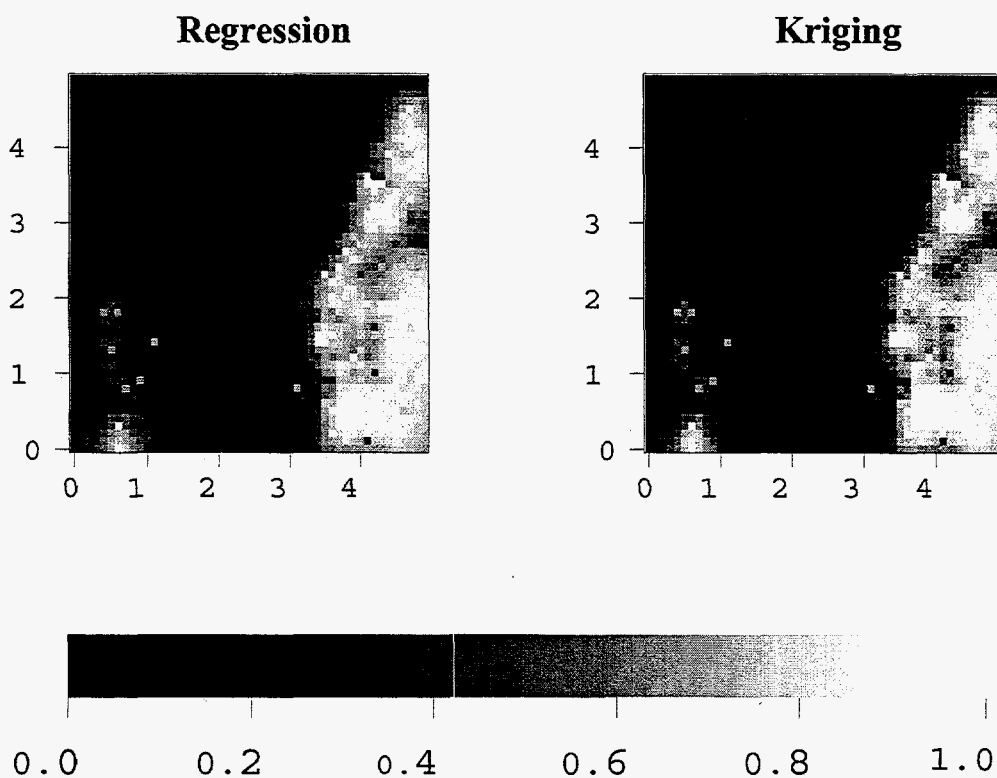


Fig. 6.11 – Comparison of Nonlinear Regression and Kriging Results; First Example, 200 points

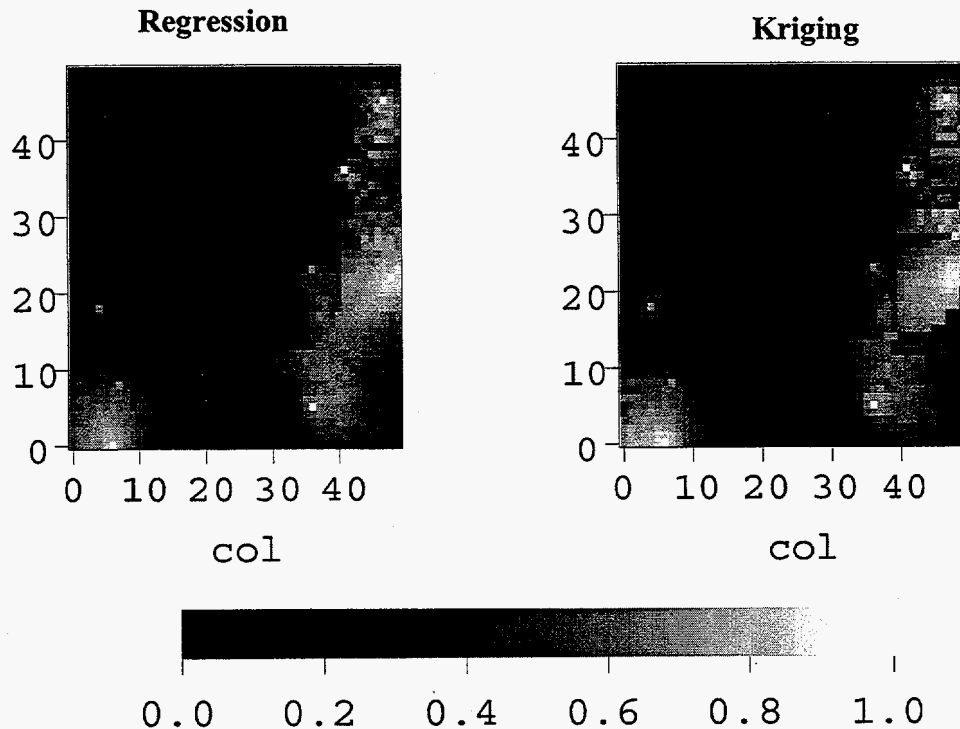


Fig. 6.12 – Comparison of Nonlinear Regression and Kriging Results; First Example, 50 points

To generate the images shown in Fig. 6.11, we used 200 sample points. Comparison of the image obtained by using the nonlinear regression method (Fig. 6.11) with the training image (Fig. 6.5) shows the ability of the proposed method to determine the reservoir architecture. For comparison, Fig. 6.11 also shows the image obtained by using the kriging procedure. The differences between the two images shown in Fig. 6.11 are indistinguishable for all practical purposes. We have, however, found that as the number of sample points decreases, the differences between the nonlinear regression and kriging results become noticeable. As an example, in Fig. 6.12, we compare the images obtained by nonlinear regression and kriging techniques by using 50 sample points. Here, the nonlinear regression method creates a better image and proves the advantage using higher order estimation methods. Especially notice the differences at central-west and northeast parts of the reservoir. It should be noted at this point that the nonlinear regression procedure suggested here is fast and accurate. Therefore, it can always be preferred to the kriging method.

For the second example, we use the training image shown in Fig. 6.13. Fig. 6.14 shows the estimates by using the kriging and nonlinear regression methods. It is clear that neither method provides a completely satisfactory image. In Fig. 6.15, we show the difference between the training image and the estimates by the two procedures. Based on the training image, on these figures, we have marked the regions with 100% probability of the facies. These figures

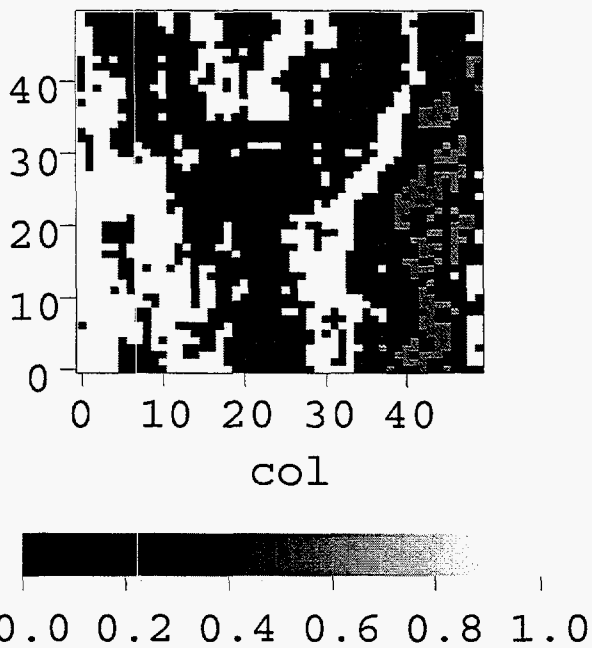


Fig. 6.13 – Training Image for the Second Example

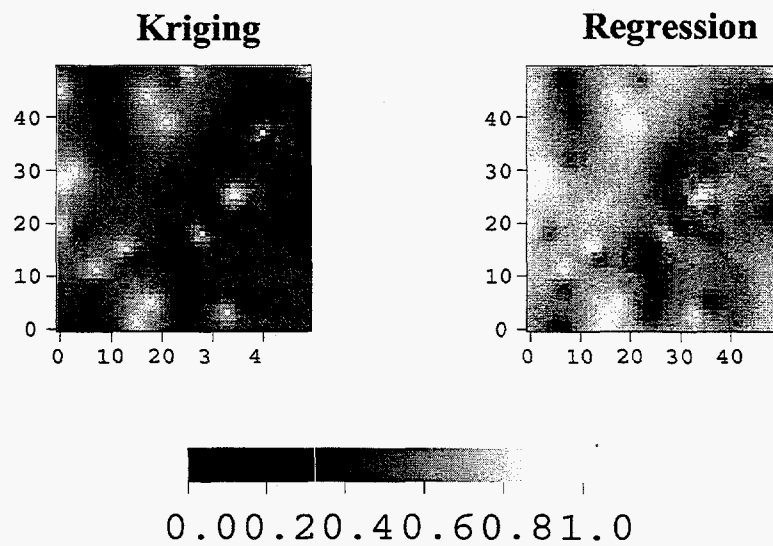


Fig. 6.14 – Comparison of Nonlinear Regression and Kriging Results; Second Example, 50 points

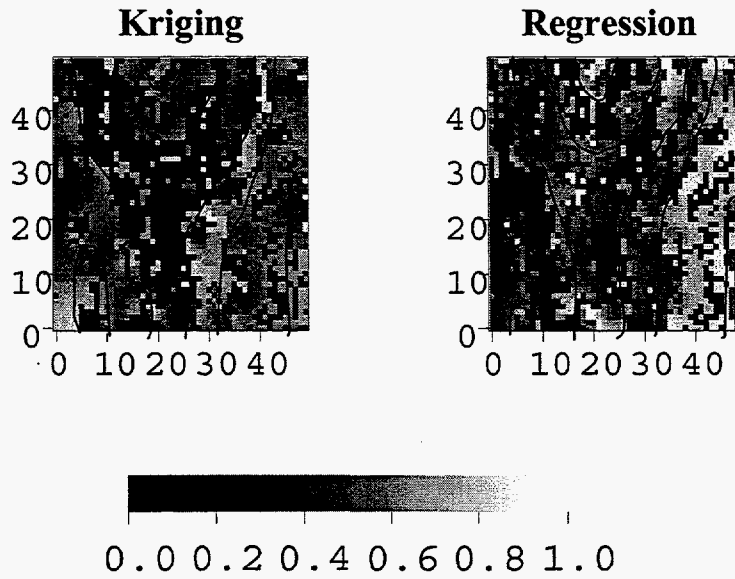


Fig. 6.15 – Differences Between the Training Image and the Estimates by the Two Methods; Second Example, 50 points

Conclusions

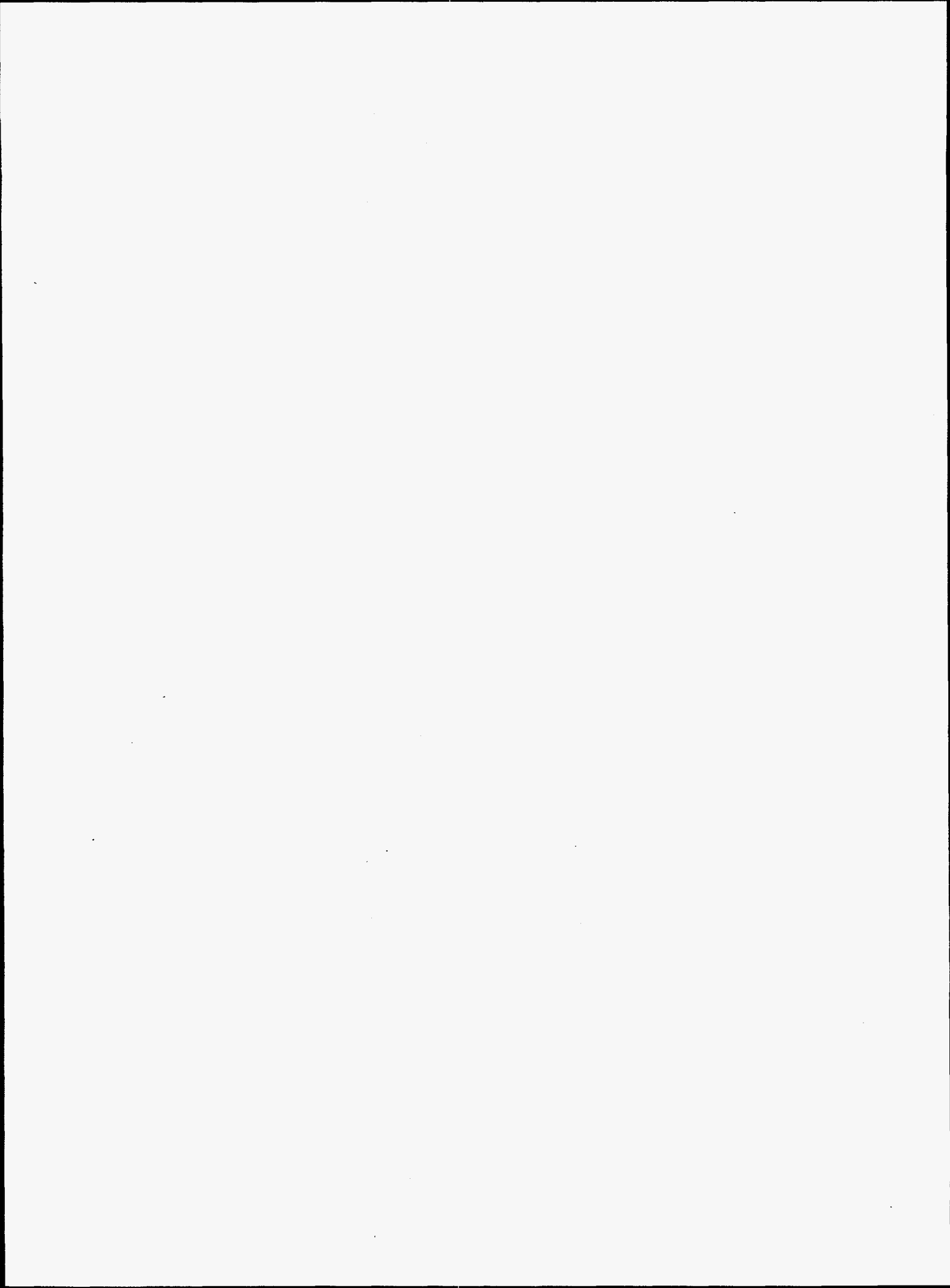
Based on the work presented in this report, following conclusions can be drawn:

- To properly capture geological features, it is critical that multi-point connectivity functions be properly represented.
- BME method has a desired flexibility to incorporate the multi-point connectivity functions. However, because the procedure is numerical, it is difficult to use in practice.
- A new regression procedure, which incorporates multi-point connectivity functions, is introduced. The method is robust and flexible enough to apply in practice. In general, it provides as good or better results than conventional kriging procedure. However, the method needs to be tested further under different conditions to validate it further.

References

Christakos, G.: "A Bayesian/Maximum Entropy View to the Spatial Estimation Problem," *Math Geology*, Vol. 22, No. 7 (1990) pp. 763-777.

Stuart, A. and Ord, J.K.: *Kendall's Advanced Theory of Statistics, Volume 1, Distribution Theory*, Fifth Edition, Oxford University Press, New York, 1987.



APPENDIX A

BME Approach

In BME approach, the estimated value $K(\mathbf{u}_0)$ is the solution of the equation

$$\frac{d \ln \{f_k(K(\mathbf{u}_0), K(\mathbf{u}_1), \dots, K(\mathbf{u}_n))\}}{dK(\mathbf{u}_0)} \Big|_{K(\mathbf{u}_0)=\hat{K}(\mathbf{u}_0)} = 0 \quad (6.A.1)$$

where the joint probability density function is defined as:

$$f_k(K(\mathbf{u}_0), K(\mathbf{u}_1), \dots, K(\mathbf{u}_n)) = \frac{1}{A} \exp \left[\sum_{q=0}^Q \mu_q g_q(K(\mathbf{u}_0), K(\mathbf{u}_1), \dots, K(\mathbf{u}_n)) \right] \quad (6.A.2)$$

and the LaGrange multipliers $\mu_q \quad q = 0, 1, \dots, Q$ can be obtained by solving

$$E[g_q(K(\mathbf{u}_0), K(\mathbf{u}_1), \dots, K(\mathbf{u}_n))] = \int \dots \int_{n+1 \text{ times}} g_q(K(\mathbf{u}_0), K(\mathbf{u}_1), \dots, K(\mathbf{u}_n)) \times f_k(K(\mathbf{u}_0), K(\mathbf{u}_1), \dots, K(\mathbf{u}_n)) dK(\mathbf{u}_0) dK(\mathbf{u}_1) \dots dK(\mathbf{u}_n) \\ q = 0, 1, \dots, Q \quad (Q = (n+1)(n+4)/2)$$

or

$$E[g_q(K(\mathbf{u}_0), K(\mathbf{u}_1), \dots, K(\mathbf{u}_n))] = \frac{1}{A} \int \dots \int_{n+1 \text{ times}} g_q(K(\mathbf{u}_0), K(\mathbf{u}_1), \dots, K(\mathbf{u}_n)) \times \exp \left[\sum_{q=0}^Q \mu_q g_q(K(\mathbf{u}_0), K(\mathbf{u}_1), \dots, K(\mathbf{u}_n)) \right] dK(\mathbf{u}_0) dK(\mathbf{u}_1) \dots dK(\mathbf{u}_n) \\ q = 0, 1, \dots, Q \quad (Q = (n+1)(n+4)/2) \quad (6.A.3)$$

where g_q are the constraints imposed on the estimation equation. Depending on the types of constraints imposed, Eq. 6.A.3 can be solved analytically. If we know the means $E[K(\mathbf{u}_i)] = m(\mathbf{u}_i)$ and the covariance $c_k(\mathbf{u}_i, \mathbf{u}_j)$, we can use following constraint conditions to estimate the indicator K value of unknown point \mathbf{u}_0 . The constraints are:

$$\begin{cases} g_0 = 1 \\ g_{q+1} = K(\mathbf{u}_q) \quad q = 0, 1, 2, \dots, n \\ g_{q+1} = (K(\mathbf{u}_i) - m(\mathbf{u}_i))(K(\mathbf{u}_j) - m(\mathbf{u}_j)) \quad q = n+1, n+2, \dots, (n+1)(n+4)/2 - 1 \\ \quad \quad \quad i, j = 0, 1, 2, \dots, n \quad q = \frac{(2n+2-i)(i+1)}{2} + j \end{cases} \quad (6.A.4)$$

Eqs. 6.A.7 - 6.A.9 can be solved analytically to get the LaGrange multipliers μ_q $q = 0, 1, \dots, (n+1)(n+4)/2$.

We can write the probability density function $f_k(K(\mathbf{u}_0), K(\mathbf{u}_1), \dots, K(\mathbf{u}_n))$ in a matrix form.

$$\begin{aligned} f_k(K(\mathbf{u}_0), K(\mathbf{u}_1), \dots, K(\mathbf{u}_n)) &= \frac{1}{A} \exp[\mu_0 + \sum_{i=0}^n \mu_{n+1} K(\mathbf{u}_i) + \frac{1}{2} \sum_{i=0}^n \sum_{j=0}^n \mu_{ij} (K(\mathbf{u}_i) - m(\mathbf{u}_i))(K(\mathbf{u}_j) - m(\mathbf{u}_j))] \\ &= \frac{1}{A} \exp[-\frac{1}{2} (\mathbf{K} - \mathbf{m})^T B (\mathbf{K} - \mathbf{m}) + \mathbf{K}^T \boldsymbol{\mu} + \mu_0] \end{aligned} \quad (6.A.10)$$

where

$$\mathbf{K} = \begin{pmatrix} K(\mathbf{u}_0) \\ K(\mathbf{u}_1) \\ \cdot \\ \cdot \\ K(\mathbf{u}_n) \end{pmatrix} \quad \mathbf{m} = \begin{pmatrix} m(\mathbf{u}_0) \\ m(\mathbf{u}_1) \\ \cdot \\ \cdot \\ m(\mathbf{u}_n) \end{pmatrix} \quad \boldsymbol{\mu} = \begin{pmatrix} -\mu_1 \\ \cdot \\ \cdot \\ -\mu_n \\ -\mu_{n+1} \end{pmatrix}$$

$$B = \begin{bmatrix} -\mu_{00} & -\mu_{01} & \cdot & \cdot & -\mu_{0n} \\ -\mu_{10} & -\mu_{11} & \cdot & \cdot & -\mu_{1n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ -\mu_{n1} & -\mu_{n2} & \cdot & \cdot & -\mu_{nn} \end{bmatrix} \quad (6.A.11)$$

$$B^T = B \quad \text{or} \quad \mu_{ij} = \mu_{ji}$$

$$i = 0, 1, \dots, n. \quad j = i, i+1, \dots, n$$

$$\mu_{ij} = \mu_q \quad q = \frac{(2n+1-i)}{2} + j$$

Eqs. 6.A.7 - 6.A.9 can be rewritten in matrix form as:

$$I_0 = \frac{1}{A} \int \int \dots \int_{m+1 \text{ times}} \exp[-\frac{1}{2} (\mathbf{K} - \mathbf{m})^T B (\mathbf{K} - \mathbf{m}) + \mathbf{K}^T \boldsymbol{\mu} + \mu_0] d\mathbf{K} = 1 \quad (6.A.12)$$

$$I_1 = \frac{1}{A} \int \int \dots \int_{m+1 \text{ times}} K(\mathbf{u}_q) \exp[-\frac{1}{2} (\mathbf{K} - \mathbf{m})^T B (\mathbf{K} - \mathbf{m}) + \mathbf{K}^T \boldsymbol{\mu} + \mu_0] d\mathbf{K} = m(\mathbf{u}_q)$$

$$q = 0, 1, 2, \dots, n \quad (6.A.13)$$

$$\begin{aligned}
I_2 &= \frac{1}{A} \int \int \dots \int_{m+1 \text{ times}} (K(\mathbf{u}_l) - m(\mathbf{u}_l))(K(\mathbf{u}_k) - m(\mathbf{u}_k)) \\
&\exp\left[-\left(\frac{1}{2}(\mathbf{K} - \mathbf{m})^T B(\mathbf{K} - \mathbf{m}) + \mathbf{K}^T \boldsymbol{\mu} + \mu_0\right)\right] d\mathbf{K} = c(\mathbf{u}_l, \mathbf{u}_k) \quad (6.A.14) \\
&l = 0, 1, 2, \dots, n \\
&k = l, l+1, \dots, n
\end{aligned}$$

For the multiple integral equations such as Eqs. 6.A.12 – 6.A.14, we have the following general analytical solutions

Let a_0 and b_0 be scalar constants; Let \mathbf{a} and \mathbf{b} be a vector of constants; let D be an $n \times n$ symmetric matrix of constants; let B be a positive definite matrix of constants. We get :

$$\begin{aligned}
&\int \int \dots \int (X^T A X + X^T \mathbf{a} + a_0) e^{-(X^T B X + X^T \mathbf{b} + b_0)} dx_1 dx_2 \dots dx_n \\
&= \frac{1}{2} \pi^{n/2} |B|^{-1/2} e^{\frac{1}{4} \mathbf{b}^T B^{-1} \mathbf{b} - b_0} [\text{tr}(A B^{-1}) - \mathbf{b}^T B^{-1} \mathbf{a} + \frac{1}{2} \mathbf{b}^T B^{-1} A B^{-1} \mathbf{b} + 2a_0] \quad (6.A.15)
\end{aligned}$$

Let $\mathbf{X} = \mathbf{K} - \mathbf{m}$. From Eq. 6.A.12, we have

$$\begin{aligned}
A &= \int \int \dots \int_{n+1 \text{ times}} \exp\left[-\left(\frac{1}{2}(\mathbf{K} - \mathbf{m})^T B(\mathbf{K} - \mathbf{m}) + \mathbf{K}^T \boldsymbol{\mu} + \mu_0\right)\right] d\mathbf{K} \\
&= e^{-\mathbf{m}^T \boldsymbol{\mu}} \int \int \dots \int_{n+1 \text{ times}} \exp\left[-\left(\frac{1}{2} \mathbf{X}^T B \mathbf{X} + \mathbf{X}^T \boldsymbol{\mu} + \mu_0\right)\right] d\mathbf{X} \quad (6.A.16) \\
&= \sqrt{\pi^{n+1}} \left| \frac{1}{2} B \right|^{-1/2} \exp\left(\frac{1}{4} \boldsymbol{\mu}^T \left(\frac{1}{2} B\right)^{-1/2} \boldsymbol{\mu} - \mathbf{m}^T \boldsymbol{\mu} - \mu_0\right)
\end{aligned}$$

From Eq. 6.A.13, we have

$$I_1 = \frac{1}{A} \tilde{I}_1 = m(\mathbf{u}_q) \quad (6.A.17)$$

where

$$\begin{aligned}
\tilde{I}_1 &= \int \int \dots \int_{m+1 \text{ times}} K(\mathbf{u}_q) \exp\left[-\left(\frac{1}{2}(\mathbf{K} - \mathbf{m})^T B(\mathbf{K} - \mathbf{m}) + \mathbf{K}^T \boldsymbol{\mu} + \mu_0\right)\right] d\mathbf{K} \quad (6.A.18) \\
&q = 0, 1, 2, \dots, n
\end{aligned}$$

Let $\mathbf{X} = \mathbf{K} - \mathbf{m}$. From Eq. 6.A.18, we have

$$\begin{aligned}\tilde{I}_1 &= e^{-\mu^T \mathbf{m}} \int \int \dots \int_{m+1 \text{ times}} (X_q + m(\mathbf{u}_q)) \exp[-(\frac{1}{2} \mathbf{X}^T B \mathbf{X} + \mathbf{X}^T \mu + \mu_0)] d\mathbf{X} \\ &= \tilde{I}_{11} + \tilde{I}_{12}\end{aligned}\quad (6.A.19)$$

where

$$\tilde{I}_{11} = e^{-\mu^T \mathbf{m}} \int \int \dots \int_{m+1 \text{ times}} \mathbf{X}_1 \exp[-(\frac{1}{2} \mathbf{X}^T B \mathbf{X} + \mathbf{X}^T \mu + \mu_0)] d\mathbf{X} \quad (6.A.20)$$

$$\tilde{I}_{12} = e^{-\mu^T \mathbf{m}} \int \int \dots \int_{m+1 \text{ times}} m(\mathbf{u}_q) \exp[-(\frac{1}{2} \mathbf{X}^T B \mathbf{X} + \mathbf{X}^T \mu + \mu_0)] d\mathbf{X} \quad (6.A.21)$$

Note from Eq. 6.A.16

$$\tilde{I}_{12} = m(\mathbf{u}_q) \times A = m(\mathbf{u}_q) \times \sqrt{\pi^{n+1}} \left| \frac{1}{2} B \right|^{-1/2} \exp[\frac{1}{4} \mu^T (\frac{1}{2} B)^{-1} \mu - \mathbf{m}^T \mu - \mu_0] \quad (6.A.22)$$

We also obtain

$$\begin{aligned}\tilde{I}_{11} &= \frac{1}{2} \sqrt{\pi^{n+1}} \left| \frac{1}{2} B \right|^{-1/2} \exp[\frac{1}{4} \mu^T (\frac{1}{2} B)^{-1} \mu - \mathbf{m}^T \mu - \mu_0] [-\mu^T (\frac{1}{2} B)^{-1} \mathbf{a}] \\ &= A \frac{1}{2} [-\mu^T (\frac{1}{2} B)^{-1} \mathbf{a}]\end{aligned}\quad (6.A.23)$$

where

$$\mathbf{a} = \begin{pmatrix} 0 \\ \cdot \\ \cdot \\ 1 \\ \cdot \\ \cdot \\ 0 \end{pmatrix} \quad \text{i.e.} \quad a_i = 1, \text{ when } i = q \quad a_i = 0, \text{ when } i \neq q \quad (6.A.24)$$

Thus we can write

$$\tilde{I}_1 = \tilde{I}_{11} + \tilde{I}_{12} = \frac{1}{2} [-\mu^T (\frac{1}{2} B)^{-1} \mathbf{a}] A + m(\mathbf{u}_q) A = [\frac{1}{2} [-\mu^T (\frac{1}{2} B)^{-1} \mathbf{a}] + m(\mathbf{u}_q)] A = m(\mathbf{u}_q) A \quad (6.A.25)$$

or

$$\frac{1}{2}[-\mu^T \left(\frac{1}{2}B\right)^{-1} \mathbf{a}] = 0 \quad (6.A.26)$$

From Eq. 6.A.26, we obtain:

$$\mu_q = 0 \quad \text{For } q = 1, 2, \dots, n+1 \quad (6.A.27)$$

For I_2 , we have

$$I_2 = \frac{1}{A} \tilde{I}_2 = c(\mathbf{u}_1, \mathbf{u}_k) \quad (6.A.28)$$

where

$$\tilde{I}_2 = \int \int \dots \int_{m+1 \text{ times}} (K(\mathbf{u}_1) - m(\mathbf{u}_1))(K(\mathbf{u}_k) - m(\mathbf{u}_k)) \exp\left[-\left(\frac{1}{2}(\mathbf{K} - \mathbf{m})^T B(\mathbf{K} - \mathbf{m}) + \mathbf{K}^T \mu + \mu_0\right)\right] d\mathbf{K} \quad (6.A.29)$$

Let $\mathbf{X} = \mathbf{K} - \mathbf{m}$, we obtain:

$$\begin{aligned} \tilde{I}_2 &= e^{-\mu^T m} \int \int \dots \int_{m+1 \text{ times}} X_l X_k \exp\left[-\left(\frac{1}{2} \mathbf{X}^T B \mathbf{X} + \mathbf{X}^T \mu + \mu_0\right)\right] d\mathbf{K} \\ &= e^{-\mu^T m} \int \int \dots \int_{m+1 \text{ times}} \mathbf{X}^T D \mathbf{X} \exp\left[-\left(\frac{1}{2} \mathbf{X}^T B \mathbf{X} + \mathbf{X}^T \mu + \mu_0\right)\right] d\mathbf{K} \end{aligned} \quad (6.A.30)$$

where

$$D = [d_{ij} = 1 \text{ for } i=l \text{ and } j=k, d_{ij} = 0 \text{ otherwise}] \quad (6.A.31)$$

Thus we have

$$\begin{aligned} \tilde{I}_2 &= \sqrt{\pi^{n+1}} \left| \frac{1}{2} B \right|^{-1/2} \exp\left[-\frac{1}{4} \mu^T \left(\frac{1}{2} B\right)^{-1} \mu - \mathbf{m}^T \mu - \mu_0\right] [-\mu^T \left(\frac{1}{2} B\right)^{-1} \mathbf{a}] [tr(DB^{-1})] \\ &= A [tr(DB^{-1})] = A \times c(\mathbf{u}_1, \mathbf{u}_k) \end{aligned} \quad (6.A.32)$$

So we get:

$$tr(DB^{-1}) = c(\mathbf{u}_1, \mathbf{u}_k) \quad (6.A.33)$$

Assume

$$B^{-1} = (b_{ij}) \quad (6.A.34)$$

From Eq. 6.A.33, we have

$$\text{tr}(DB^{-1})=b_{kl} \quad (6.A.35)$$

or

$$b_{kl}=c(\mathbf{u}_l, \mathbf{u}_k) \quad (6.A.36)$$

We obtain:

$$B^{-1} = \begin{vmatrix} c(\mathbf{u}_0, \mathbf{u}_0) & c(\mathbf{u}_0, \mathbf{u}_1) & \dots & c(\mathbf{u}_0, \mathbf{u}_n) \\ c(\mathbf{u}_1, \mathbf{u}_0) & c(\mathbf{u}_1, \mathbf{u}_1) & \dots & c(\mathbf{u}_1, \mathbf{u}_n) \\ \dots & \dots & \dots & \dots \\ c(\mathbf{u}_n, \mathbf{u}_0) & c(\mathbf{u}_n, \mathbf{u}_1) & \dots & c(\mathbf{u}_n, \mathbf{u}_n) \end{vmatrix} \quad (6.A.37)$$

By inverting the matrix in the Eq. 6.A.37, we can get the value of LaGrange multipliers μ_{ij} .

From Eq. 6.A.6, we get the density function

$$f_k(K(\mathbf{u}_0), K(\mathbf{u}_1), \dots, K(\mathbf{u}_n)) = C \exp\left[+\frac{1}{2} \sum_{i=0}^n \sum_{j=0}^n \mu_{ij} (K(\mathbf{u}_i) - m(\mathbf{u}_i))(K(\mathbf{u}_j) - m(\mathbf{u}_j))\right] \quad (6.A.38)$$

where C is a constant.

BME method requires

$$\frac{d \ln\{f_k(K(\mathbf{u}_0), K(\mathbf{u}_1), \dots, K(\mathbf{u}_n))\}}{dK(\mathbf{u}_0)} \Big|_{K(\mathbf{u}_0)=\hat{K}(\mathbf{u}_0)} = 0 \quad (6.A.39)$$

By combining Equations.A.37 and A.38, we get

$$\frac{d}{dK(\mathbf{u}_0)} \left[\sum_{i=0}^n \sum_{j=0}^n \mu_{ij} (K(\mathbf{u}_i) - m(\mathbf{u}_i))(K(\mathbf{u}_j) - m(\mathbf{u}_j)) \right] \Big|_{K(\mathbf{u}_0)=\hat{K}(\mathbf{u}_0)} = 0 \quad (6.A.40)$$

or

$$\mu_{00}(\hat{K}(\mathbf{u}_0) - m(\mathbf{u}_0)) + \sum_{i=1}^n \mu_{i0}(K(\mathbf{u}_i) - m(\mathbf{u}_i)) = 0$$

The estimated value of unknown point \mathbf{u}_0 is:

$$\mathbf{K}_l = \begin{pmatrix} K_l(\mathbf{u}_0) \\ K_l(\mathbf{u}_1) \\ \cdot \\ \cdot \\ K_l(\mathbf{u}_n) \end{pmatrix} \quad \mathbf{m}_l = \begin{pmatrix} m_l(\mathbf{u}_0) \\ m_l(\mathbf{u}_1) \\ \cdot \\ \cdot \\ m_l(\mathbf{u}_n) \end{pmatrix} \quad \mu_l = \begin{pmatrix} -\mu_{l,1} \\ \cdot \\ \cdot \\ -\mu_{l,n} \\ -\mu_{l,n+1} \end{pmatrix} \quad l=1,2,\dots,d$$

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}_1 \\ \mathbf{K}_2 \\ \cdot \\ \cdot \\ \mathbf{K}_d \end{pmatrix} \quad \mathbf{m} = \begin{pmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \cdot \\ \cdot \\ \mathbf{m}_d \end{pmatrix} \quad \mu = \begin{pmatrix} -\mu_1 \\ -\mu_2 \\ \cdot \\ \cdot \\ -\mu_d \end{pmatrix}$$

$$B_{lk} = \begin{bmatrix} -\mu_{lk,00} & -\mu_{lk,01} & \cdot & \cdot & -\mu_{lk,0n} \\ -\mu_{lk,10} & -\mu_{lk,11} & \cdot & \cdot & -\mu_{lk,1n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ -\mu_{lk,n1} & -\mu_{lk,n2} & \cdot & \cdot & -\mu_{lk,nn} \end{bmatrix} \quad \text{for } l, k = 1, 2, \dots, d, B_{lk} = B_{kl}, \text{ or } \mu_{lk,ij} = \mu_{kl,ij}$$

$$B = \begin{bmatrix} B_{11} & B_{12} & \cdot & \cdot & B_{1d} \\ B_{21} & B_{22} & \cdot & \cdot & B_{2d} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ B_{d1} & B_{d2} & \cdot & \cdot & B_{dd} \end{bmatrix} \quad B^T = B \quad \text{or} \quad B_{lk} = B_{kl} \quad (6.A.45)$$

Using similar steps as a single facies case, we obtain

$$B^{-1} = \begin{vmatrix} B_{11}^{-1} & B_{12}^{-1} & \cdot & \cdot & B_{1d}^{-1} \\ B_{21}^{-1} & B_{22}^{-1} & \cdot & \cdot & B_{2d}^{-1} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ B_{d1}^{-1} & B_{d2}^{-1} & \cdot & \cdot & B_{dd}^{-1} \end{vmatrix} \quad (6.A.46)$$

where

$$B_{lk}^{-1} = \begin{vmatrix} c_{lk}(\mathbf{u}_0, \mathbf{u}_0) & c_{lk}(\mathbf{u}_0, \mathbf{u}_1) & \cdot & \cdot & c_{lk}(\mathbf{u}_0, \mathbf{u}_n) \\ c_{lk}(\mathbf{u}_1, \mathbf{u}_0) & c_{lk}(\mathbf{u}_1, \mathbf{u}_1) & \cdot & \cdot & c_{lk}(\mathbf{u}_1, \mathbf{u}_n) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ c_{lk}(\mathbf{u}_n, \mathbf{u}_0) & c_{lk}(\mathbf{u}_n, \mathbf{u}_1) & \cdot & \cdot & c_{lk}(\mathbf{u}_n, \mathbf{u}_n) \end{vmatrix} \quad l, k = 1, 2, \dots, d \quad (6.A.47)$$

Eq. 6.A.46 can be rewritten as:

$$B^{-1} = \begin{vmatrix} c_{11}(\mathbf{u}_0, \mathbf{u}_0) & \cdot & c_{11}(\mathbf{u}_0, \mathbf{u}_n) & c_{12}(\mathbf{u}_0, \mathbf{u}_0) & \cdot & \cdot & c_{12}(\mathbf{u}_0, \mathbf{u}_n) & \cdot & \cdot & c_{1d}(\mathbf{u}_0, \mathbf{u}_0) & \cdot & c_{1d}(\mathbf{u}_0, \mathbf{u}_n) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ c_{11}(\mathbf{u}_n, \mathbf{u}_0) & \cdot & c_{11}(\mathbf{u}_n, \mathbf{u}_n) & c_{12}(\mathbf{u}_n, \mathbf{u}_0) & \cdot & \cdot & c_{12}(\mathbf{u}_n, \mathbf{u}_n) & \cdot & \cdot & c_{1d}(\mathbf{u}_n, \mathbf{u}_0) & \cdot & c_{1d}(\mathbf{u}_n, \mathbf{u}_n) \\ c_{21}(\mathbf{u}_0, \mathbf{u}_0) & \cdot & c_{21}(\mathbf{u}_0, \mathbf{u}_n) & c_{22}(\mathbf{u}_0, \mathbf{u}_0) & \cdot & \cdot & c_{22}(\mathbf{u}_0, \mathbf{u}_n) & \cdot & \cdot & c_{2d1}(\mathbf{u}_0, \mathbf{u}_0) & \cdot & c_{2d}(\mathbf{u}_0, \mathbf{u}_n) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ c_{21}(\mathbf{u}_n, \mathbf{u}_0) & \cdot & c_{21}(\mathbf{u}_n, \mathbf{u}_n) & c_{22}(\mathbf{u}_n, \mathbf{u}_0) & \cdot & \cdot & c_{22}(\mathbf{u}_n, \mathbf{u}_n) & \cdot & \cdot & c_{2d}(\mathbf{u}_n, \mathbf{u}_0) & \cdot & c_{2d}(\mathbf{u}_n, \mathbf{u}_n) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ c_{d1}(\mathbf{u}_0, \mathbf{u}_0) & \cdot & c_{d1}(\mathbf{u}_0, \mathbf{u}_n) & c_{d2}(\mathbf{u}_0, \mathbf{u}_0) & \cdot & \cdot & c_{d2}(\mathbf{u}_0, \mathbf{u}_n) & \cdot & \cdot & c_{dd}(\mathbf{u}_0, \mathbf{u}_0) & \cdot & c_{dd}(\mathbf{u}_0, \mathbf{u}_n) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ c_{d1}(\mathbf{u}_n, \mathbf{u}_0) & \cdot & c_{d1}(\mathbf{u}_n, \mathbf{u}_n) & c_{d2}(\mathbf{u}_n, \mathbf{u}_0) & \cdot & \cdot & c_{d2}(\mathbf{u}_n, \mathbf{u}_n) & \cdot & \cdot & c_{dd}(\mathbf{u}_n, \mathbf{u}_0) & \cdot & c_{dd}(\mathbf{u}_n, \mathbf{u}_n) \end{vmatrix} \quad (6.A.48)$$

By inverting the Eq. 6.A.48, we can obtain the LaGrange multipliers $\mu_{lk,ij}$ $l, k=1, 2, \dots, d$ and $i, j=0, 1, \dots, n$

The probability density function for multi-facies case is given by:

$$f_k(\mathbf{K}_1, \mathbf{K}_2, \dots, \mathbf{K}_d) = \frac{1}{A} \exp[\mu_0 + \frac{1}{2} \sum_{l=1}^d \sum_{k=1}^d \sum_{i=0}^n \sum_{j=0}^n \mu_{lk,ij} (K_l(\mathbf{u}_i) - m_l(\mathbf{u}_i))(K_k(\mathbf{u}_j) - m_k(\mathbf{u}_j))] \quad (6.A.49)$$

To estimate the l facies $K_l(\mathbf{u}_0)$, $l=1, 2, \dots, d$. we use the equation

$$\frac{d \ln \{f_k(\mathbf{K}_1, \mathbf{K}_2, \dots, \mathbf{K}_d)\}}{dK_l(\mathbf{u}_0)} \Big|_{K_l(\mathbf{u}_0) = \hat{K}_l(\mathbf{u}_0)} = 0 \quad (6.A.50)$$

By combining Eqs. 6.A.49 and 6.A.50, we get

$$\frac{d}{dK_l(\mathbf{u}_0)} \left[\sum_{l=1}^d \sum_{k=1}^d \sum_{i=0}^n \sum_{j=0}^n \mu_{lk,ij} (K_l(\mathbf{u}_i) - m_l(\mathbf{u}_i))(K_k(\mathbf{u}_j) - m_k(\mathbf{u}_j)) \right] \Big|_{K_l(\mathbf{u}_0) = \hat{K}_l(\mathbf{u}_0)} = 0$$

or

$$\mu_{ll,00}(\hat{K}_l(\mathbf{u}_0) - m_l(\mathbf{u}_i)) + \sum_{k=1}^d \sum_{i=1}^n \mu_{lk,i0} (K_k(\mathbf{u}_i) - m_k(\mathbf{u}_i)) = 0 \quad (6.A.51)$$

Thus, the estimated value of facies l at the unknown point \mathbf{u}_0 is:

$$\hat{K}_l(\mathbf{u}_0) = m_l(\mathbf{u}_0) - \frac{1}{\mu_{ll,00}} \sum_{k=1}^d \sum_{i=1}^n \mu_{lk,i0} (K_k(\mathbf{u}_i) - m_k(\mathbf{u}_i)) \quad (6.A.52)$$

$l=1,2,\dots,d$

APPENDIX B

BME Approach for Multi-Point Connectivity

For multi-point connectivity case, the analytical solution of Eqs. 6.A.12 – 6.A.14 is not possible. Therefore, a numerical algorithm is needed to compute the Lagrange multipliers required by the application of Bayesian/Maximum Entropy (BME) procedure. The algorithm we have developed includes two crucial elements: i) a convergent and stable iterative solution procedure for the Lagrange multipliers and ii) an efficient and accurate numerical integration procedure to compute multiple integrals. We have tested the Newton-Raphson iteration procedure and determined that this procedure is stable and convergent if a sufficient degree of accuracy can be attained in the numerical evaluation of multiple integrals. We have also compared various numerical integration procedures and convinced ourselves that integration by using Chebyshev polynomials would yield accurate results for our applications. Below, we present the specific details of the iteration procedure and document some results to demonstrate the validity of our numerical approach.

Numerical Solution Procedure for BME Method

Our computational scheme for the LaGrange multipliers involves an iterative solution that is accomplished by the Newton-Raphson procedure. The application of the Newton-Raphson iterative procedure is summarized below:

The probability density function is given by,

$$f = \frac{1}{A} \exp \left\{ \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \mu_{ij} (x_i - m_i)(x_j - m_j) \right\} \quad (6.B.1)$$

where μ_{ij} are the LaGrange multipliers and A is given by (the normalization constraint),

$$A = \iint \cdots \int_n \exp \left\{ \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \mu_{ij} (x_i - m_i)(x_j - m_j) \right\} dx_1 dx_2 \cdots dx_n \quad (6.B.2)$$

To obtain the LaGrange multipliers, μ_{ij} , for a given set of covariances, $C_{\ell k} = C(S_\ell, S_k)$, we need to solve the following set of equations,

$$\frac{1}{A} \iint \cdots \int_n (x_\ell - m_\ell)(x_k - m_k) \exp \left\{ \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \mu_{ij} (x_i - m_i)(x_j - m_j) \right\} dx_1 dx_2 \cdots dx_n = C(S_\ell, S_k) \quad (6.B.3)$$

for $\ell, k = 1, 2, \dots, n$.

Combining Eqs. 6.B.2 and 6.B.4 and rearranging, we can write,

$$g_{\ell k}(\mu) = \iint \cdots \int_n (y_\ell y_k - C_{\ell k}) \exp \left\{ \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \mu_{ij} y_i y_j \right\} dy_1 dy_2 \cdots dy_n = 0 \quad (6.B.4)$$

for $\ell, k = 1, 2, \dots, n$.

The set of equations given in Eq. 6.B.4 can be solved for $\mu_{ij}, i, j = 1, 2, \dots, n$ by the Newton-Raphson iteration procedure. For convenience, let us define

$$G_i(\mu) = [g_{i1}(\mu), g_{i2}(\mu), \dots, g_{in}(\mu)]^T \quad i = 1, \dots, n \quad (6.B.5)$$

$$\mu_i = [\mu_{i1}, \mu_{i2}, \dots, \mu_{in}] \quad i = 1, \dots, n \quad (6.B.6)$$

$$\mu = [\mu_1, \mu_2, \dots, \mu_n]^T \quad (6.B.7)$$

Then, we can write Eq. 6.B.4 as,

$$G(\mu) = [G_1(\mu), G_2(\mu), \dots, G_n(\mu)] = 0 \quad (6.B.8)$$

Let,

$$\mu^{k+1} = \mu^k + \delta^k \quad (6.B.9)$$

where δ^k is the solution of,

$$\phi(\mu^k) \delta^k = -G(\mu^k) \quad (6.B.10)$$

with the matrix $\phi(\mu)$ defined as,

$$\phi(\mu) = \left[\frac{\partial G_i(\mu)}{\partial \mu_j} \right] \quad i, j = 1, \dots, n \quad (6.B.11)$$

For a given set of covariances, $C_{\ell k} = C(S_\ell, S_k)$, starting with an initial guess $\mu^0 = [\mu_1^0, \mu_2^0, \dots, \mu_n^0]$, we solve Eq. 6.B.10 for δ^k . We update μ^k by using Eq. 6.B.9 until $\max\{\delta\} \leq TOL$, where TOL is a sufficiently small number.

We have coded the iterative computation algorithm and tested for convergence and stability. Our findings can be summarized as follows:

- i) The Newton-Raphson iteration procedure can be used to compute the Lagrange multipliers required by the BME approach.
- ii) The algorithm is always convergent for small initial guesses (in the order of 10^{-1} or less).
- iii) The key to the success of the iteration procedure is the accuracy of the numerical evaluation of multiple integrals.

As explained above, the integrals we need to evaluate have the following form:

$$I = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} f(x_1, x_2, \dots, x_n) dx_1, dx_2 \dots dx_n \quad (6.B.12)$$

We have found that the following relation leads to the transformation of the improper integral in Eq. 6.B.12 to a computationally efficient form:

$$\begin{aligned} I_i &= \int_{-\infty}^{+\infty} f(x_1, x_2, \dots, x_i, \dots, x_n) dx_i \\ &= \int_0^1 [f(x_1, x_2, \dots, \ln t_i, \dots, x_n) + f(x_1, x_2, \dots, -\ln t_i, \dots, x_n)] \frac{dt_i}{t_i} \end{aligned} \quad (6.B.13)$$

Note that the integral in Eq. 6.B.13 has an integrable singularity at the lower limit and can be numerically evaluated by using procedures that avoid the evaluation of the integrand at the end points. We have tried Simpson's rule with open formulas, Romberg integration, and integration by using Chebyshev polynomials. Our results indicate that Chebyshev polynomials are the most efficient numerical integration procedure for our purposes.

To demonstrate the success of our numerical solution algorithm, below we present some results. In all the examples, we use bivariate distributions to be able to compare our results with those obtained by the analytical procedure.

Examples

We now present some numerical results to demonstrate the accuracy of our procedure. In each example, for a given set of covariances, $C_{\ell k} = C(S_{\ell}, S_k)$, we compute the LaGrange multipliers, μ_{ij} . The initial guesses for the Lagrange multipliers to start the iteration process and the correct values of LaGrange multipliers computed by using the analytical (kriging) technique are noted in the examples for comparison purposes. CPU times for the computations are also provided. We use $TOL = 10^{-4}$ to terminate the iterations in all examples.

Example 1: 2-Data Points

$$\text{Covariences, } c(S_{\ell}, S_k) = \begin{pmatrix} 1.0 & 0.5 \\ 0.5 & 1.0 \end{pmatrix}$$

$$\text{Initial guess for Lagrange multipliers, } \mu_{ij}^0 = \begin{pmatrix} -0.05 & 0.03 \\ 0.03 & -0.05 \end{pmatrix}$$

$$\text{Estimated values of Lagrange multipliers, } \mu_{ij}^{k+1} = \begin{pmatrix} -0.666648 & 0.333333 \\ 0.333333 & -0.666648 \end{pmatrix}$$

$$\text{Correct values of Lagrange multipliers, } \mu_{ij} = \begin{pmatrix} -0.666667 & 0.333333 \\ 0.333333 & -0.666667 \end{pmatrix}$$

Total CPU time: 0 hours, 0 minutes, 6 seconds.

Example 2A: 3-Data Points

$$\text{Covariences, } c(S_{\ell}, S_k) = \begin{pmatrix} 1.0 & 0.8 & 0.8 \\ 0.8 & 1.0 & 0.5 \\ 0.5 & 0.5 & 1.0 \end{pmatrix}$$

$$\text{Initial guess for Lagrange multipliers, } \mu_{ij}^0 = \begin{pmatrix} -0.5 & 0.2 & 0.2 \\ 0.2 & -0.5 & -0.1 \\ 0.2 & -0.1 & -0.5 \end{pmatrix}$$

$$\text{Estimated values of Lagrange multipliers, } \mu_{ij}^{k+1} = \begin{pmatrix} -3.40905 & 1.81817 & 1.81817 \\ 1.81817 & -1.63634 & -0.63634 \\ 1.81817 & -1.63634 & -1.63634 \end{pmatrix}$$

$$\text{Correct values of Lagrange multipliers, } \mu_{ij} = \begin{pmatrix} -3.409091 & 1.818182 & 1.818182 \\ 1.818182 & -1.636364 & -0.636364 \\ 1.818182 & -0.636364 & -1.636364 \end{pmatrix}$$

Total CPU time: 0 hours, 15 minutes, 2 seconds.

Example 2B: 3-Data Points

$$\text{Covariences, } C(S_\ell, S_k) = \begin{pmatrix} 1.0 & 0.8 & 0.8 \\ 0.8 & 1.0 & 0.5 \\ 0.5 & 0.5 & 1.0 \end{pmatrix}$$

$$\text{Initial guess for Lagrange multipliers, } \mu_{ij}^0 = \begin{pmatrix} -3.2 & 1.6 & 1.6 \\ 1.6 & -1.5 & -0.5 \\ 1.6 & -0.5 & -1.5 \end{pmatrix}$$

$$\text{Estimated values of Lagrange multipliers, } \mu_{ij}^{k+1} = \begin{pmatrix} -3.40905 & 1.81817 & 1.81817 \\ 1.81817 & -1.63634 & -0.63634 \\ 1.81817 & -1.63634 & -1.63634 \end{pmatrix}$$

$$\text{Correct values of Lagrange multipliers, } \mu_{ij} = \begin{pmatrix} -3.409091 & 1.818182 & 1.818182 \\ 1.818182 & -1.636364 & -0.636364 \\ 1.818182 & -0.636364 & -1.636364 \end{pmatrix}$$

Total CPU time: 0 hours, 8 minutes, 2 seconds.

Example 3: 4-Data Points

$$\text{Covariences, } C(S_\ell, S_k) = \begin{pmatrix} 1.0 & 0.9 & 0.7 & 0.8 \\ 0.9 & 1.0 & 0.5 & 0.6 \\ 0.7 & 0.5 & 1.0 & 0.8 \\ 0.8 & 0.6 & 0.8 & 1.0 \end{pmatrix}$$

$$\text{Initial guess for Lagrange multipliers, } \mu_{ij}^0 = \begin{pmatrix} -6.3 & 4.1 & 0.8 & 1.9 \\ 4.1 & -3.4 & -0.5 & -0.8 \\ 0.8 & -0.5 & -1.5 & 0.8 \\ 1.9 & -0.8 & 0.8 & -2.2 \end{pmatrix}$$

$$\text{Estimated values of Lagrange multipliers, } \mu_{ij}^{k+1} = \begin{pmatrix} -6.38842 & 4.16611 & 0.833329 & 1.94442 \\ 4.16611 & -3.49952 & -0.499997 & -0.833322 \\ 0.833329 & -0.499997 & -1.49997 & 0.83332 \\ 1.94442 & -0.833322 & 0.83332 & -2.22218 \end{pmatrix}$$

$$\text{Correct values of Lagrange multipliers, } \mu_{ij} = \begin{pmatrix} -6.388886 & 4.166664 & 0.833333 & 1.944444 \\ 4.166664 & -3.499998 & -0.500000 & -0.833333 \\ 0.833333 & -0.500000 & -1.500000 & 0.833334 \\ 1.944444 & -0.833333 & 0.833333 & -2.222222 \end{pmatrix}$$

Total CPU time: 4 hours, 57 minutes, 43 seconds.

The examples presented above clearly show the stability and accuracy of the computational algorithm we have developed. The computational time increases as the number of data points increases. Examples 2A and 2B indicate, as expected, that the closer the initial guesses are to the correct values the shorter the computational time. It is clear that the computational time is largely owing to the numerical evaluation of multiple integrals.

APPENDIX C

Iterative Solution for Nonlinear Regression Procedure

Here we briefly describe the iterative solution of Eq. 6.26 by using Newton's method. Let us, for convenience, define the following:

$$f = [f_1(\lambda), f_2(\lambda), \dots, f_n(\lambda)] \quad (6.C.1)$$

where

$$f_i(\lambda) = \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n \lambda_j \lambda_k \lambda_l M_{ijkl} - 3 \sum_{j=1}^n \sum_{k=1}^n \lambda_j \lambda_k M_{ijk0} + 3 \sum_{j=1}^n \lambda_j M_{ij00} - M_{i000} = 0; \quad (6.C.2)$$

for $i=1, 2, \dots, n$

and

$$\lambda = [\lambda_1, \lambda_2, \dots, \lambda_n] \quad (6.C.3)$$

Let,

$$\lambda^{k+1} = \lambda^k + \delta^k \quad (6.C.4)$$

where δ^k is the solution of,

$$\phi(\lambda^k) \delta^k = -f(\lambda^k) \quad (6.C.5)$$

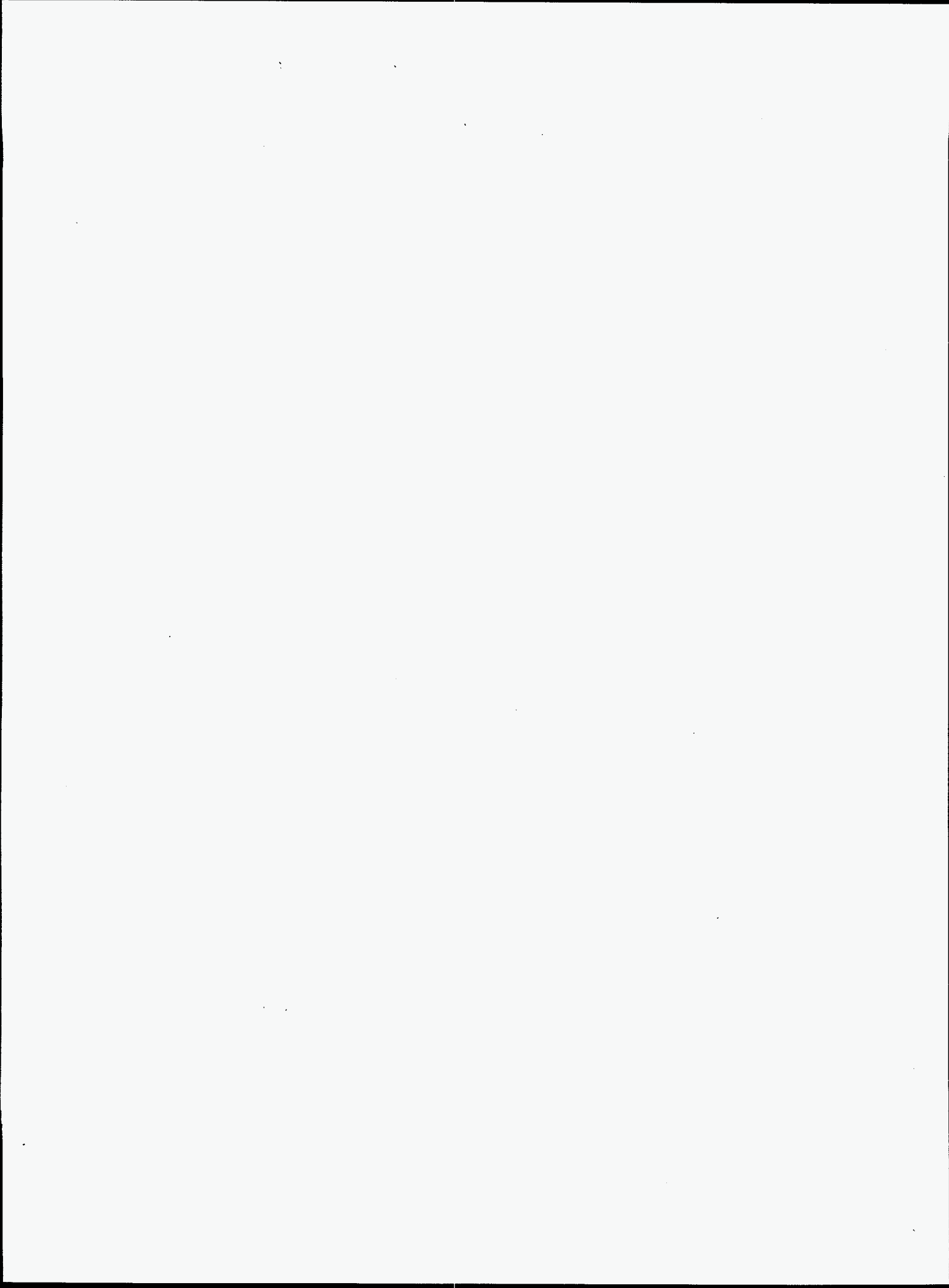
with the matrix $\phi(\lambda)$ defined as,

$$\phi(\lambda) = \left[\frac{\partial f_i(\lambda)}{\partial \lambda_m} \right] \quad i, m = 1, \dots, n \quad (6.C.6)$$

The entries of the matrix $\phi(\lambda)$ are given by

$$\frac{\partial f_i}{\partial \lambda_m} = 3M_{im00} - 6 \sum_{j=1}^n \lambda_j M_{ijm0} + 3 \sum_{j=1}^n \sum_{k=1}^n \lambda_j \lambda_k M_{ijkm} \quad (6.C.7)$$

For a given set of M_{ijkl} values, starting with an initial guess $\lambda^0 = [\lambda_1^0, \lambda_2^0, \dots, \lambda_n^0]$, we solve Eq. 6.C.6 for δ^k . We update λ^k by using Eq. 6.C.4 until $\max\{\delta\} \leq TOL$, where TOL is a sufficiently small number. We have observed that the initial guess of λ^0 's can be obtained either based on kriging estimates or starting with very small values. The convergence is always fast.



Appendix .D
User Manual for forth moments estimate

1. CalForthMoment

Calculate the forth moments by input indicator values.

Input file format,

x	y	Indicator
---	---	-----------

For example,

1	1	0
1	2	1
5	10	0.7
:	:	:
:	:	:

Output file format,

h1	h2	h3	Forth Moments
----	----	----	---------------

where h1, h2 and h3 are the lag distance.

For example,

0	1	12	0.00753541
5	1	13	0.00565549
0	7	14	0.00403898

2. ForthEstimate

This program solve the non linear equations and get the forth moments estimation by using Newton method. We use the solution of Kriging estimate as the initial guess of the non-linear equations. The input data are forth moments, data point to be estimated, sample points and covariance. The output is estimated values of the forth moments.

Input data file format,

Forth Moments

h1	h2	h3	moments
----	----	----	---------

data points

x	y
---	---

sample points

x	y	indicator
---	---	-----------

covariance

h	covariance
---	------------

Output file format,
x y indicator

3. **KrigingEstimate**

Estimate indicators by using simply Kriging. Input covariance and data points to be estimated. Output estimated values.

Input data format,
covariance
h covariance

data points
x y

Output file format,
x y indicator

4. **Other Programs**

Covariance: calculate covariance.

Lu: solve linear equations.