THE UNIVERSITY OF TULSA

THE GRADUATE SCHOOL

SAMPLING THE A POSTERIORI PROBABILITY DENSITY FUNCTION

FOR PERMEABILITY FIELDS CONDITIONED TO THE VARIOGRAM

AND WELL- TEST PRESSURE DATA

by

Luciane Bonet de Souza Cunha

A DISSERTATION

APPROVED FOR THE DISCIPLINE OF

PETROLEUM ENGINEERING

By Dissertation Committee

Albert C. Reynolds, Co-Chairperson

Dean S. Oliver, Co-Chairperson

Ruby Thompson

Richard A. Redner

ABSTRACT

Bonet-Cunha, L. S. (Doctor of Philosophy in Petroleum Engineering)

Sampling the a Posteriori Probability Density Function for Permeability Fields Conditioned to the Variogram and Well-Test Pressure Data (291 pp. - Chapter VI)

Co-Directed by Dr. Albert C. Reynolds, Jr. and Dr. Dean S. Oliver

(327 words)

To investigate the uncertainty in reservoir performance, it is necessary to formulate the classical inverse problem of generating a random permeability field that is consistent with observed pressure data, a known mean and a variogram model in a probabilistic way. This requires the definition of an a posteriori probability distribution that combines prior information (mean and variogram) with new information represented here by pressure data.

If one can generate a set of realizations of the permeability field which represents a correct sampling of the a posteriori probability density function, then by making a flow simulation with each realization and analyzing the set of outcomes, one can evaluate the uncertainty in predicted reservoir performance.

We present efficient Markov chain Monte Carlo procedures to sample the correct probability distribution, i.e., to generate realizations of the permeability field that correctly sample the true a posteriori probability distribution. The efficiency of our methods results from our procedures for proposing perturbations in the permeability field and from constructing a starting model that at least approximates a realization of the correct probability distribution. The proposed perturbations are such that the acceptance rate is extremely high.

Realizations of the permeability field constructed from Markov chain Monte Carlo methods are not equally probable, but we adopt the viewpoint that equally probable realizations are not desirable if one wishes to evaluate the uncertainty in reservoir performance by simulating from the set of reservoir realizations. Instead, one should construct realizations from the correct probability distribution (or correct probability density function) in order to quantify the uncertainty in reservoir performance.

In addition, two implementations of simulated annealing are presented that can be used to sample from the set of most probable models. The first method is derived from the classical theory of simulated annealing with a properly defined objective function. The second simulated annealing method is based on our new ideas for proposing perturbations of the log-permeability field at each iteration of a Markov chain Monte Carlo method.

ACKNOWLEDGMENTS

I wish to express my gratitude and appreciation to Dr. Albert C. Reynolds, Jr., and Dr. Dean S. Oliver for their invaluable assistance and guidance as co-directors of this research project. I also express my gratitude to Dr. Leslie G. Thompson of the University of Tulsa, and Dr. Richard A. Redner of the University of Tulsa, for their comments and suggestions and for serving on my dissertation committee.

I extend my thanks and appreciation to all the other faculty members and to my graduate students colleagues, who contributed to my education as a TU graduate student.

I want to acknowledge Chevron Petroleum Technology Company, TUPREP - The University of Tulsa Petroleum Reservoir Exploitation Projects, The U.S. Department of Energy (BDM - Oklahoma, Inc. subcontract No. 93-0001) and CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico, for supporting my graduate studies at the University of Tulsa. I also wish to acknowledge PETROBRÁS, Petróleo Brasileiro S.A., for providing me with the opportunity to attend the University of Tulsa.

My deepest thanks go to my parents Inês and Euclides for their support in my education and the love they gave me.

I am specially grateful to my husband José Carlos whose encouragement and companionship will always be appreciated.

This work is dedicated to my son Gabriel and my daughter Laura for all the joy they bring me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 The Inverse Problem

Reservoir description is an integral part of reservoir management and performance prediction. With a growing emphasis on the enhancement of hydrocarbon reserves by developing mature fields, the need for more accurate reservoir descriptions is crucial. A better description of the distribution of reservoir rock properties will improve our ability to evaluate hydrocarbon reserves, select infill well locations, and design and predict displacement processes.

In constructing a reservoir description, certain data must be honored. Data obtained at sampled locations can be divided into two categories: static and dynamic. Static data, such as core, log, seismic and geologic interpretation do not evolve significantly over time. On the other hand, dynamic data obtained at the well, such as production and pressure history, are functions of time.

It is now recognized that proper integration of static data with dynamic data is critical for reservoir description. For example, it is known that ignoring information obtained from static data when history matching production data yields nonunique

solutions, i.e., different estimates of the set of reservoir properties may all yield an acceptable match of the production history.

When considering the problem of incorporating both static and dynamic data into reservoir description, it is necessary to solve a classical inverse problem in which limited information is available to the geologist or reservoir engineer. Because of that, in practice, it is impossible to accurately specify the distribution of reservoir properties. As a result of this uncertainty, we can only expect to generate probabilistic answers to questions of reservoir performance. We want to be able to quantify the probability of certain reservoir performance levels. Because the flow and transport in petroleum reservoirs are not linear functions of reservoir parameters, it is not possible to calculate directly the probability distribution for future reservoir performance. Instead, we are forced to estimate the probability distribution from the outcomes of flow predictions for a large number of possible realizations of the reservoir. For this method to work, it is essential that the permeability and porosity realizations used in the flow simulation adequately reflect the uncertainty in reservoir properties. It is necessary that the permeability and porosity realizations used in the flow simulation are truly samples for a probability density function representative of the possible arrangement of reservoir properties.

As a first step, it is necessary to formulate this inverse problem in a probabilistic way. This can be accomplished by using inverse problem theory[1]. In what follows, the forward problem for generating pressure responses for a given permeability distribution is solved using a reservoir simulator. Thus, the permeability field is represented by the values of permeability in the gridblocks. We represent the natural logarithm of these permeabilities as random variables, assume the associated random function is stationary

and assume these random variables satisfy a multinormal distribution. Any given set of log-permeability values representing our physical system, is called a model. Every model m can be considered as a point in the model space M. We will define different probability densities over M. For instance, a probability density $\rho_M(m)$ will represent our priori information on models, and another probability density, $\pi(m)$ will represent our a posteriori information, deduced from $\rho_M(m)$ and from the degree of fit between data predicted from models and observed data. As shown if Ref. 1, $\pi(m) = a\rho_M(m)L(m)$, where $L(m)$ denotes the likelihood function, which is a measure of the degree of fit between data predicted from the model m and the observed data (a is an appropriate normalization constant.) Typically, the likelihood function is related to a misfit function $S(m)$, through an expression like $L(m) = k\exp(-S(m))$.

In terms of the a posteriori probability density on the model space, it is in general possible that there exists a multitude of local maxima as well as a set of global maxima which represents the set of most probable models. In our work, any estimate constructed for one of these most probable models will be referred to as the maximum a posteriori estimate or the most probable model.

When the goal is to obtain the maximum a posteriori estimate, it is possible to use methods that rely on using local information on the gradient of the misfit function to improve upon some starting model in an iterative fashion. In such cases, a local search for the maximum a posteriori estimate can be performed using a gradient method. Although gradient methods[2-4] have been recently successfully applied in the context of inverse problem theory to generate rock property fields conditioned to prior means, the variograms and multiwell pressure data, we believe that gradient methods may fail for

some problems, i.e., fail to converge to a physically feasible solution or converge to a local minimum which does not represent a maximum a posteriori estimate. In Refs. 2-4, such convergence problems were not encountered, which suggests that convergence problems may not be as bad as previously anticipated. However, multiphase flow problems have not yet been considered. Anticipated convergence problems in the application of gradient methods provided the original motivation for investigating the method of simulated annealing to determine the most probable model.

When analyzing the inverse problem, obtaining a maximum a posteriori estimate is usually not sufficient because we normally also wish to have some information on the uncertainty of the estimate. Because the flow in petroleum reservoirs is not a linear function of reservoir parameters, it is not possible to calculate directly the probability distribution from the outcomes of the flow simulation. But it is possible to pseudorandomly generate a large collection of models according to the a posteriori probability distribution and analyze and display models in such a way that information on the relative likelihoods of model properties is conveyed to the spectator. This can be accomplished by means of efficient simulated annealing and Monte Carlo methods.

1.2 Probabilistic Formulation of the Inverse Problem

In our application, the model m represents the set of gridblocks values of log-permeability ($\ln(k)$). Based on our assumption of a multinormal distribution, the prior distribution has a probability density function satisfying the following proportionality relation:

$$\rho_M(m) \propto \exp\left(-\frac{1}{2}(m-m_0)^T C_M^{-1}(m-m_0)\right). \quad (1.1)$$

Throughout, m_0 is the prior mean (expected value) of m and C_M is the covariance matrix obtained from the variogram model.

In the problem considered here, d_{obs} is the vector of measured pressure data, and includes all measured wellbore pressure data used as conditioning data. The relationship between the vector d of calculated data and the vector m of model reservoir parameters is written as

$$d=g(m). \quad (1.2)$$

The functional relationship of Eq. 1.2 represents the generation of d from a reservoir simulator for a given model m. If m is the true model (actual values of gridblock log-permeabilities), then Eq. 1.2 would give the "true" pressure data d, which will generally differ from d_{obs} due to pressure measurement errors. Here, pressure measurement errors are assumed to be independent identically distributed Gaussian random variables with zero mean and variance σ_d^2. Thus, the covariance matrix for these errors is a diagonal matrix C_D with all diagonal entries equal to σ_d^2.

For a given set of data, the likelihood function for the model is given by the following equation:

$$L(m) \propto \exp\left(-\frac{1}{2}(g(m)-d_{obs})^T C_D^{-1}(g(m)-d_{obs})\right). \quad (1.3)$$

As shown in Refs. 1-4, it follows from Bayes's theorem that the a posteriori probability density function for our model, denoted $\pi(m)$, satisfies the following relation:

$$\pi(m) = aL(m)\rho_M(m), \tag{1.4}$$

or using Eqs. 1.1 and 1.3,

$$\pi(m) = a\exp\left[-\frac{1}{2}((m-m_0)^T C_M^{-1}(m-m_0) + \right.$$

$$\left.(g(m)-d_{obs})^T C_D^{-1}(g(m)-d_{obs}))\right], \tag{1.5}$$

where a is a constant. The most probable model, referred to as the maximum a posteriori estimate, is the one that maximizes $\pi(m)$, or equivalently, minimizes

$$S(m) = \frac{1}{2}[(m-m_0)^T C_M^{-1}(m-m_0) +$$

$$(g(m)-d_{obs})^T C_D^{-1}(g(m)-d_{obs})]. \tag{1.6}$$

Among all models that are conditioned to or "honor" the prior mean and covariance (variogram), the maximum a posteriori estimate is the most probable model. Put another way, the most probable model is the one that minimizes the objective function $O = S(m)$. Here, our language suggests that there is a unique global minimum of our objective function, but, as noted previously, it is possible that there exist a set of global minima (set of most probable models or set of maximum a posteriori estimates.)

In Refs. 2-4, the maximum a posteriori estimate is estimated using a gradient method. In this work, we show that the maximum a posteriori estimate can also be approximated by simulated annealing.

It is important to realize that although we have made the assumption that the errors in the data are normally distributed, $\pi(m)$ is generally not a multivariate normal distribution unless $g(m) = Gm$ where G is a linear operator. As shown in Ref. 1 if $g(m) =$

Gm, then we can write directly a formula for the a posteriori probability distribution for our model as

$$\hat{\pi}(m) \propto \exp\left[-\frac{1}{2}((m-m_\infty)^T C_{MP}^{-1}(m-m_\infty))\right]. \quad (1.7)$$

In this case the a posteriori probability distribution for the random variable m is multivariate normal with mean, m_∞, and covariance, C_{MP}. Because this is a very simple distribution, it is possible to use any number of standard geostatistical methods of simulation, including an LU decomposition method to sample this distribution. For this case we can sample $\hat{\pi}(m)$ by generating models m_j , j = 1,...,n (n = number of desired samples) such that

$$m_j = m_\infty + LZ_j, \quad (1.8)$$

where $C_{MP} = LL^T$ and Z_j is a vector of standard normal deviates.

However, in this work we assume that the relationship between the model parameters and the data is not linear and then the a posteriori distribution to be sampled is the one given in Eq. 1.5. To do this it is necessary to have a sampling procedure that does not assume that the a posteriori probability density function is a multivariate Gaussian distribution or has any other special form. A simple, but general, method that satisfies this characteristic, and can be used to sample from nonlinear probability distribution functions, is a exhaustive search. A systematic exploration of the discretized model space is performed, and all models within the considered model subspace are visited. Although this method may be possible for problems with low dimensionality (i.e., with few parameters),

the task is not computationally feasible for problems which have a large number of model parameters.

For our problem, evaluation of Eq. 1.5 requires the calculation of a misfit function which requires a flow simulation for each choice of the random variables. Since, in the simplest case, there will be thousands to tens thousands of gridcells, each with a value of permeability assigned to it, and since a complete evaluation of the probability distribution would require a flow simulation for every possible combination of permeabilities, it is clearly impossible to exhaustively sample the parameter space.

When analyzing the nonlinear inverse problem, associated with Eq. 1.5, and the number of model parameters is large, it is therefore necessary to severely restrict the number of misfit calculations and consequently flow simulations. As will be shown in this work, one way to do this is to use a Markov chain Monte Carlo search.

1.3 Monte Carlo Markov Chains and Simulated Annealing Overview

The idea behind Markov chain Monte Carlo methods is old, and its actual application to the solution of scientific problems is closely connected to the advent of modern electronic computers. The ideas that form the basis of these methods were first published by Metropolis *et al.*[5](1953) in a algorithm, now known as the Metropolis algorithm, that was able to (asymptotically) sample a space according to a Gibbs-Boltzmann distribution. This algorithm was a biased random walk whose individual steps (iterations) were based on very simple probabilistic rules.

The main features of these methods for sampling from a distribution with density $\pi(m)$ are:

- A sequence of samples is obtained by simulating a Markov chain; i.e., the probability of generating some particular new state depends only on the preceding state in the sequence. This transition probability is careful chosen in such a way that, after a finite number of transitions, the chain samples converge to the desired $\pi(m)$ distribution. In practice, it often takes a large number of transitions to reach the stationary distribution and this is called the *transient* period of the chain;
- The computations depend on $\pi(m)$ only through ratios of the form $\pi(m_j)q_{ji}/\ \pi(m_i)\ q_{ij}$, where m_j and m_i are sample points and q_{ji} is the probability of proposing a transition from state m_j to m_i. Thus, the normalizing constant for $\pi(m)$ does not need to be known and the methods are very easily implemented on a computer;
- Since a sequence of samples is obtained by simulating a Markov chain, the resulting samples may be correlated and calculation of statistics based on the sampling requires some care.

Around the same time that the Metropolis algorithm was introduced, another thread of development began. Alder and Wainwrigth[6](1959) introduced the "molecular dynamics" method as a means of simulating physical systems in which new states are found by simulating the dynamical evolution of the system. Recently, this idea and the Metropolis algorithm have been united in the hybrid Monte Carlo method of Duane, Kennedy, Pendleton, and Roweth[7](1987) and some of its variants like the method proposed by Horowitz[8] (1991).

Basically, the hybrid Monte Carlo differs from the usual Monte Carlo applications by two factors. First, the a posteriori distribution to be sampled is expanded by the introduction of a set of 'fictitious' variables, p, in one-to-one correspondence with the model parameter variables. The sampling process gives pairs of sampled model and fictitious variables. However, the distribution of the model parameters is the marginal distribution of this expanded distribution. This means that, after the end of the sampling process we need only consider the samples obtained for the model parameters, while ignoring the values obtained for the fictitious variables. Second, the transitions from one state to another in the Markov chain is performed by combining an equation of motion method based on Hamiltonian dynamics with the Metropolis algorithm. As will be demonstrated later, these two main features of the hybrid Monte Carlo procedure make it efficient by eliminating the random walk behavior that is present in conventional Monte Carlo applications.

A wide variety of problems have been, or are being, approached using Markov chain Monte Carlo methods for sampling. Useful references on the application of Markov chain Monte Carlo methods to the earth sciences include publications by Tjelmeland *et al.*[9] (1994) and Hegstad *et al.*[10](1993) One of the first applications in this area was done by Smith and Freeze[11](1979) in groundwater flow. They conducted a stochastic analysis of two-dimensional steady-state groundwater flow in a bounded domain by means of Monte Carlo techniques. In their study, the flow domain was divided into a set of square blocks. A nearest-neighbor stochastic process model was used to generate a multilateral spatial dependence between hydraulic conductivity values in the block system. Both statistically isotropic and anisotropic autocorrelation functions were considered. This model led to a

realistic representation of the spatial variations in hydraulic conductivity in a discrete block medium. Results of the simulations provided estimates of the output distribution in hydraulic head. They concluded that uncertainties in the predicted hydraulic head values were strongly influenced by the presence of spatial trends in the mean hydraulic conductivity. They also concluded that in evaluating the concept of an effective conductivity for a heterogeneous medium, both the nature of the spatial heterogeneities in hydraulic conductivity and the flow system operating within the flow domain must be considered.

One recent example of using Bayes theorem and a Markov chain Monte Carlo algorithm for generating a posteriori probabilities for an inverse problem is given by Neal[12](1995). He used an implementation of the hybrid Monte Carlo method when developing a Bayesian approach to learning for neural networks. At first, he addressed the problem of defining classes of prior distributions for network parameters since his Bayesian approach requires an explicit formula for the prior distribution. As he pointed out, predictions of future observations when using a Bayesian approach are made by integrating the model's prediction with respect to the a posteriori parameter distribution obtained by updating this prior to account for the data. For neural networks, the task of integration over the a posteriori distribution is computationally demanding. He solved this problem using Markov chain Monte Carlo methods. More specifically, he demonstrated that the use of the hybrid Monte Carlo algorithm, which is based on dynamical simulation, is superior to methods based on simple random walks.

A technique based also on the Metropolis algorithm, proposed independently by Kirkpatrick *et al.*[13] (1983) and Cerny[14] (1985), for use in statistical mechanics, known as

simulated annealing has been widely applied to optimization problems. The main goal of simulated annealing is to find near-optimal solutions to complex optimization problems. To clarify the basis of simulated annealing let us assume that the probability density function for an hypothetical problem is given by

$$\rho(m) = a\exp[-O(m)]. \qquad (1.9)$$

In the simulated annealing technique, the state m of the system being optimized is identified with the state of a statistical mechanical system; the objective function $O(m)$ being minimized is identified with the physical energy (E), and the optimization process is controlled by a parameter T which can be identified with the physical temperature. The system to be optimized is allowed to 'equilibrate' by applying a set of moves, i.e. a set of system pertubations, and accepting or rejecting the moves according to the Metropolis algorithm; if the move cause the energy to decrease ($\Delta E \leq 0$), the perturbation is accepted; if ΔE is positive, then the perturbation is accepted with probability $P(\Delta E) = e^{-\Delta E/T}$. This conditional acceptance is easily implemented by choosing a random number α uniformly distributed between 0 and 1. If $\alpha \leq P(\Delta E)$, then the perturbation is accepted; otherwise the existing value for the parameter is retained.

The mathematical formalism of the simulated annealing algorithm is based on the theory of finite Markov chains. By exploiting the connection between simulated annealing and Markov chains, it is possible to prove that the system will converge to "thermal equilibrium" if the Metropolis rule is used and the following two assumptions hold: (1.) Any state of the system to be optimized can be reached from any other state of the system,

using the prescribed move criteria, (2.) There is a non-zero probability of staying in the current state in a given move; see Refs. 15 and 16.

In thermal equilibrium at temperature T, the states m of a hypothetical, large statistical ensemble of systems, identical to the considered system, are distributed according to the following distribution $\tau(m) = b(T)e^{-E(m)/T}$ which can be related to the probability density function for our hypothetical problem (Eq. 1.9) when $T = 1$. This suggests that it is possible to use the simulated annealing technique to sample a certain probability distribution. However, although is possible to use simulated annealing in probabilistic inference the same way a Markov chain Monte Carlo is used, most of the literature on annealing assumes that the goal is to find a minimum energy state by cooling to a temperature of zero.

Using Markov chain theory it is also possible to prove that by reducing the value of the temperature parameter T according to specific rules, the system will converge to one of the most probable models of the probability distribution given by Eq. 1.9. In this respect, the use of a global search algorithm like simulated annealing can be advantageous for nonlinear problems in which several local and/or global minima for the misfit function exist. In this case, methods that rely on local information from the objective function derivative can lead to a local minimum since the local geometry of the misfit function surface in the model space may not directly contain information about the direction to follow in a search for the global minimum.

A global search method is not confined to uphill (or downhill) moves in the model space and is therefore less influenced by the presence of local optima. In simulated annealing, the high initial temperature reduces the effect of energy barriers, allowing free

movement through state space. The hope is that as the temperature is then reduced, the simulation will end up on the right side of the energy barriers, that is, on the side where it is more likely to be under the canonical distribution at the final temperature. This is illustrated by Fig. 1.1. The plot shows the behavior of the energy of a uni-dimensional state at six different values of the temperature parameter. At high temperatures the energy barrier can easily be transversed but, as the temperature is reduced, the simulation at the final temperature will possibly end up in the region where there is the lowest energy value or equivalently the highest probability. Fig. 1.2 shows the effect of the temperature parameter in the corresponding probability function. (In Fig. 1.2 the probability density functions are normalized to have a maximum value of unity.)

The hope that simulated annealing will end up generating states near the one of the most probable models is based on the idea that the probability distribution at higher temperatures is a good guide to the distribution at lower temperatures. This may often be the case, but will not always be. Fig. 1.3 shows a problem presented by Neal[17](1993) where this assumption fails. The plot shows the behavior of the energy of a uni-dimensional state. The distribution at high temperatures is dominated here by the size of the two low-energy regions, not by the depths of their lowest points. Since the global minimum happens to be in the smaller region, annealing will in this case direct the simulation away from the region where the equilibrium distribution will be concentrated at very low temperatures.

Such examples are cause for caution in evaluating the prospects of success using annealing, but they should not be taken as discouraging its general use, as no realistic procedure can guarantee good results in all cases.

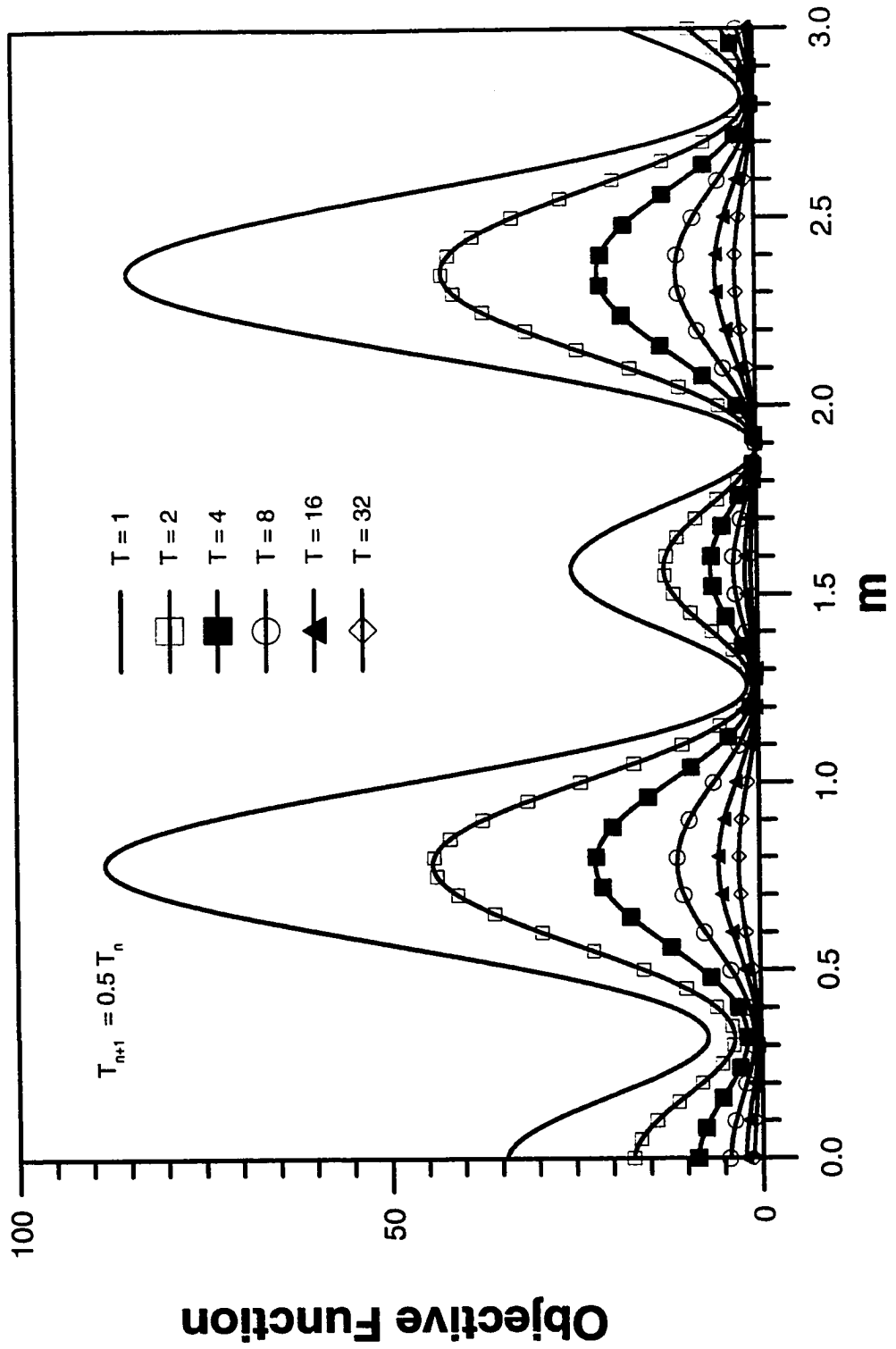

Fig. 1.1 - Effect of the Temperature Parameter in the Simulated Annealing Objective Function.

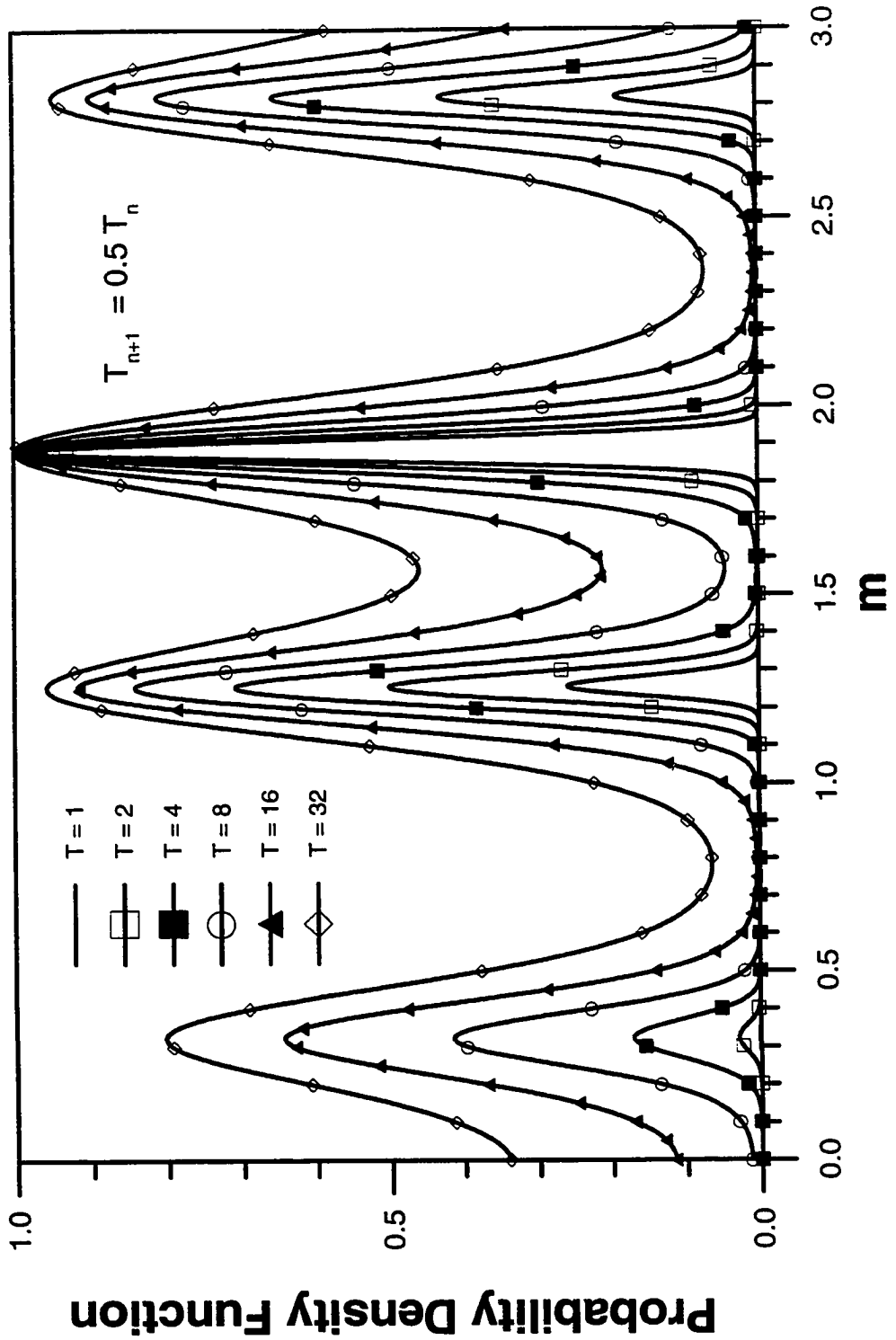

Fig. 1.2 - Effect of the Temperature Parameter in the Probability Density Function when Using Simulated Annealing.

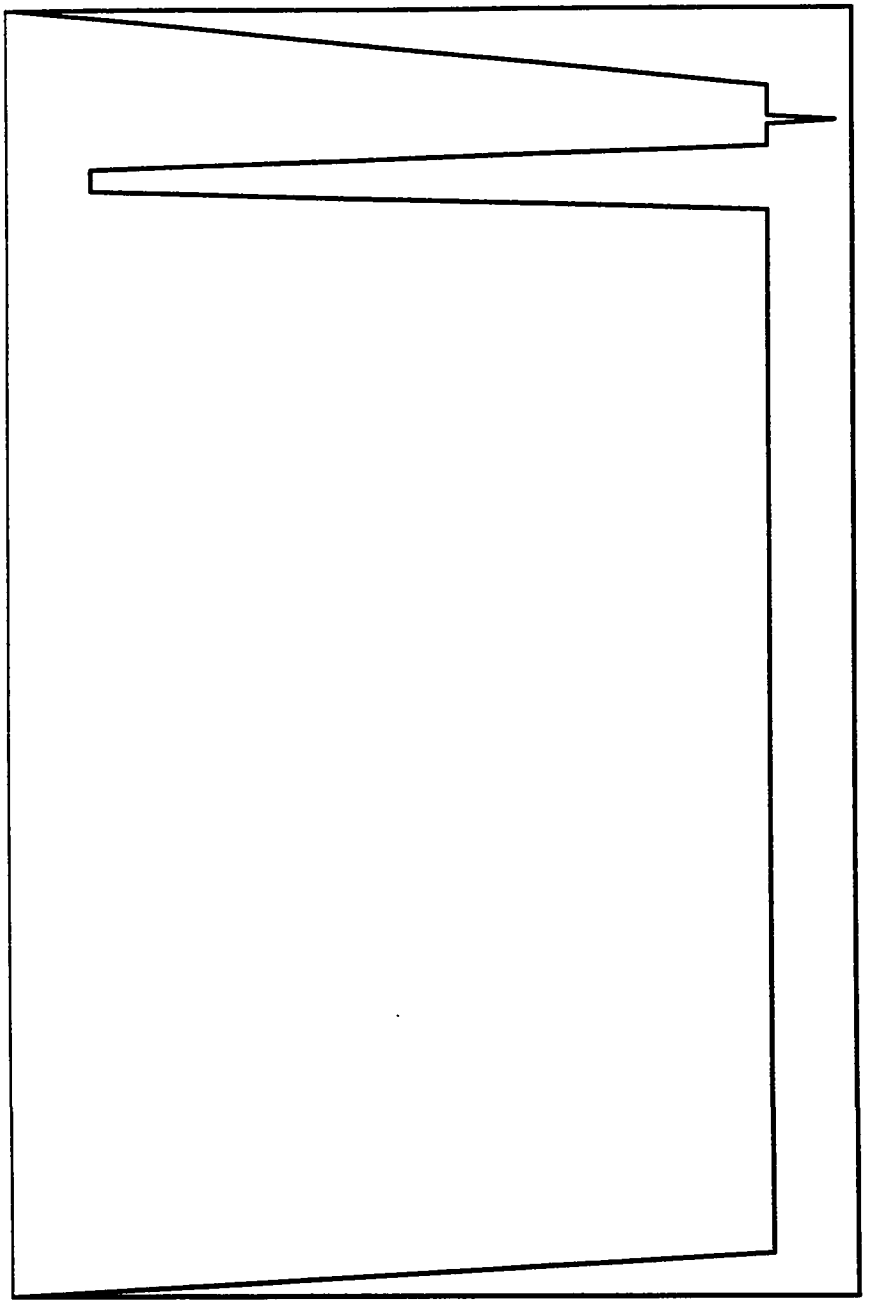

Fig. 1.3 - Potential Problems when Using Simulated Annealing Algorithm.

One of the first examples of solving an inverse problem by means of the simulated annealing method is given in the area of resolution of images by Geman and Geman[18] (1984). They discussed an application of simulated annealing to Bayesian image restoration. For their particular inverse problem, a two-dimensional deconvolution problem, they derived an expression for the a posteriori distribution from (1) the prior distribution, (2) a model of the convolutional two-dimensional image blurring mechanism, and (3) the parameters of the Gaussian noise model. By identifying this a posteriori distribution with a Gibbs-Boltzmann distribution, they performed a maximum a posteriori estimation in the model space, using a simulated annealing algorithm. In their paper, they mention the possibility of using the simulated annealing algorithm, not only for maximum a posteriori estimation, but also to sample the model space according to the a posteriori distribution. However, they did not pursue this possibility further, nor did they describe how to extend this idea to inverse problems in general.

Marroquin *et al.*[19] (1987) adopted an approach similar to that of Geman and Geman. However, they used the Metropolis algorithm to generate the a posteriori distribution, from which they computed model estimates. One of the problems noted by them was that their Bayesian approach requires an explicit formula for the prior distribution.

Geophysical examples of solution of inverse problems by means of the simulated annealing method are given by Koren *et al.*[20] (1991), Mosegaard and Vestergaard[21] (1991) and Vestergaard and Mosegaard[22] (1991), who all used simulated annealing within the difficult context of seismic waveform fitting. Koren *et al.* described the potential of simulated annealing methods, not only for solving a model optimization problem, but also

for performing an analysis of resolution in the inverse problem. By running a modified simulated annealing method, samples from the model space were drawn in such a way that their frequencies of occurrence approximate their a posteriori likelihoods. Using this method, maximum likelihood estimation and uncertainty analysis of seismic background velocity models are performed on multioffset seismic data. Another example is given by Rothman[23] (1985), who solved a strongly nonlinear optimization problem arising in seismic reflection surveys by means of simulated annealing.

Several stochastic and combinatorial optimization techniques exist for generating reservoir descriptions, but simulated annealing has rapidly become the dominant algorithm in use; see Refs. 24-28 and their reference lists. Although the popularity of the algorithm stems in part from its theoretical underpinnings[5,15,18] (convergence to a global minimum can be proved under stringent assumptions), its extensive use also arises from the fact that additional data constraints can be incorporated by a simple modification of the objective function and its apparent success in generating rock-property fields that yield good approximations to the global minimum of the objective function. Unfortunately, if well-test pressure data or production data are incorporated directly into the objective function, then a forward problem must be solved at each iteration of simulated annealing in order to update the objective function. In general, each solution of the forward problem requires one run of a reservoir simulator, which is extremely expensive when one must perform thousands of iterations in the simulated annealing algorithm.

To avoid this problem, when conditioning to the variogram and production data, Hird[27](1992) developed a conditional simulation method based on simulated annealing that instead of a fluid flow simulator, uses functions defining the connectivity of a reservoir

which are based on the spatial arrangement of permeability. These static connectivity functions are correlated to production performance parameters, i.e., water oil ratios and water breakthrough times. The connectivity functions and the spatial statistics are then used as constraints to develop alternate reservoir descriptions. Thus, dynamic production information about the reservoir is translated to quantifiable static information. The assumption is that this procedure yields descriptions of reservoir heterogeneities sufficient for future performance prediction.

To also avoid the problem of flow simulation runs, when conditioning to the variogram and well-test pressure data, various authors (see for example, Refs. 25 and 26) have incorporated pressure data by estimating an "average effective" permeability from pressure data and then incorporating the difference between this fixed average value computed from a simulated annealing image into the objective function. Thus, this method yields permeability descriptions conditioned not to the well-test pressure data, but to an interpretation of the pressure data. Although this approach avoids the solution of the forward problem and is computationally convenient, the permeability field obtained by simulated annealing may give a pressure response which is in poor agreement with the actual pressure data[26] and the two corresponding sets of pressure derivative data may be completely different.

It is important to note that current applications of simulated annealing in reservoir description generate a realization of the permeability field that is consistent with observed pressure data and a known variogram model. Unfortunately, the meaning of this resulting realization is not clear. In this work, our main goal is to generate a large number of realizations and ensure that the distribution of realizations represent a correct sampling of

the a posteriori distribution for our model. This is necessary if one wants to investigate the uncertainty of reservoir behavior.

1.4 Outline of the Remainder of the Thesis

The remainder of this thesis will deal with issues concerning sampling of permeability fields conditioned to the prior mean, the variogram and multiwell pressure data and is organized as follows.

In Chapter II, we address the problem of producing permeability fields conditioned to multiwell pressure information, the prior mean and a model variogram that are truly samples of the a posteriori distribution for our model by using Markov chain Monte Carlo methods. We focus on improving computational efficiency relative to two aspects: shortening the length of the transient period and increasing the number of accepted transitions. As will be shown, the first objective can be obtained by using a method to select the first state in the chain such that the probability for selecting any specific state is approximately the same as the probability assigned to it by the a posteriori probability distribution.

To design a random walk that will sample the a posteriori probability density function is not difficult. However, in cases where the a posteriori probability density function has narrow maxima, these maxima will be very sparsely sampled. In such cases, sampling of the model space can be improved by importance sampling[29] that is, by sampling the model space with a probability density as close to the desired a posteriori probability distribution as possible. One way to attempt to do this is to use a method

which samples the model space according to the prior distribution. This approach is superior to a uniform sampling by a crude Monte Carlo. However, due to the fact that the two distributions, prior and a posteriori, may be completely different, it would be preferable to draw sample models from the model space according to a probability distribution which is closer to the a posteriori distribution.

A way to accomplish this is discussed in Chapter II, where we use specially tailored perturbations to the states in such a way that the prior information (mean and model variogram) is automatically satisfied and the pressure data are approximately matched at every step with the objective being to increase the number of accepted transitions. These perturbations make use of sensitivity coefficients (derivatives of pressure with respect to a permeability gridblock value) calculated from the reservoir simulation.

Chapter III explores the connection between simulated annealing and Markov chain Monte Carlo methods and provides procedures for generating realizations of the permeability field. We consider various perturbation mechanisms for generating states in a simulated annealing process and show that, with a proper choice of the objective function, we can find a temperature level T such that images generated with simulated annealing at this temperature level correspond to states in a Markov chain that converges to yield a set of realizations which represents a correct sampling of the a posteriori probability function for our model.

Based on the theory described in Chapter III, it is possible to delineate two techniques to sample from the set of most probable models. The first technique relies on a proper definition of the objective function and uses the wellknown two-point swap as a perturbation mechanism. The second is based on new ideas for proposing perturbations of

the log-permeability field at each iteration of a Markov chain Monte Carlo method. In this respect, we show that if simulated annealing is applied properly, one can obtain a maximum a posteriori estimate that is in close agreement with the one obtained from a gradient method[2-4]. It is also possible to envision techniques that, using simulated annealing, are able to overcome the transient period in a Markov chain and then can be used in applications where the techniques described in other chapters do not work properly. Here, the main advantage of simulated annealing would be to avoid convergence to a local minimum.

In Chapter III, a discussion concerning the current practice of including the variogram in the objective function in simulated annealing applications is also presented. We show that variograms computed from realizations of permeability fields that are samples from the correct probability distribution function do not necessarily give a good match to the model variogram. This fact leads then to our argument that the variogram should not be included in the objective function.

In Chapter IV, we take a different approach for sampling the a posteriori probability distribution of the permeability field. This is accomplished by using a hybrid Monte Carlo method which is conceptually designed to improve the sampling process by avoiding the random walk behavior that is present in simpler Markov chain Monte Carlo method implementations.

Chapter V, considers the implementation of the techniques summarized in the previous chapters and also presents a comparison between the methods described in this work.

In the concluding chapter the contributions of the dissertation are summarized and areas for future research are outlined.

CHAPTER II

MARKOV CHAIN MONTE CARLO METHODS FOR SAMPLING

In this chapter, a description of the basic concept of Markov chains is presented and a Markov chain Monte Carlo implementation to generate a set of realizations of the permeability field, which represents a true sample from the a posteriori distribution described in Chapter I, is presented.

Generating realizations of the permeability field that are consistent with observed pressure data and a known variogram model in order to investigate the uncertainty in reservoir behavior is a difficult problem. At present, it appears that only the Markov chain Monte Carlo methods offer any hope of producing results whose accuracy is reasonably assured in a feasible amount of time, without the need for any questionable assumptions.

These methods have been used for many years to solve problems in statistical physics, and have recently been widely applied to Bayesian models in statistics. Markov chain Monte Carlo methods make no assumptions concerning the form of the distribution, such as whether it can be approximated by a Gaussian. In theory at least, Markov chain Monte Carlo methods account properly for multiple modes. The main disadvantage of Markov chain methods commonly used for statistical applications is that they may in some circumstances require a very long time to converge to the desired distribution.

In this chapter, the focus of our efforts will be on improving the efficiency of the overall method by reducing the number of perturbations to overcome the transient period of the chain and increasing the acceptance rate of proposed states.

2.1 Theory of Markov Chains

A Markov chain is a sequence of random variables $X^{(0)}, X^{(1)}, \ldots$, where the probability distribution for $X^{(n)}$ is determined entirely by the probability distribution of $X^{(n-1)}$. The set of all possible values for the X^n's is called the *state space*. If the state space is finite, then the Markov chain is said to be finite. A Markov chain can also be thought of a sequence of trials where the probability of the outcome of trial n depends only on the outcome of trial n-1. The *transition probability* $p_{ij}^{(n,n-1)}$ gives the probability that the process goes from state i (or outcome m^i) to state j (or outcome m^j) at trial n. In this work, we often simply refer to m^i as state i. If the transition probabilities do not depend on n, the Markov chain is said to be *homogeneous*. For a homogeneous Markov chain, the transition probabilities at every step or trial, can be denoted by p_{ij}. Since the p_{ij}'s are probabilities, we must have

$$\sum_j p_{ij} = 1, \tag{2. 1}$$

for all i where the sum is over all states. A Markov chain is said to *irreducible* if for every (i,j), there exists n such that $p^n_{ij}>0$,where here $p^n_{ij} = \text{Prob}\{X_{m+n} = j \mid X_m = i\}$ is the probability of going from state i to j in n trials.

For all n, we let $p_n(m^j)$ denote the probability of obtaining state j as the outcome of the nth trial so

$$p_n(m^j) = \sum_i p_{ij}\, p_{n-1}(m^i)\,. \qquad (2.2)$$

A *stationary* or *invariant* probability distribution $\pi(m^i)$ over the states of the Markov chain with transition probabilities p_{ij} is one that satisfies

$$\pi(m^j) = \sum_i \pi(m^i)\, p_{ij}\,. \qquad (2.3)$$

A stationary distribution persists for ever[17] once it is reached. This means that once we reach a point, where the states generated in the Markov chain represent samples of π, all successive states generated in our Markov chain will also represent a sample from π. In our applications, we wish to ensure that there is only one stationary distribution and that this stationary distribution represents the correct probability distribution for our model (the log-permeability field on the gridblocks.)

Invariance with respect to π is implied by the stronger condition of a detailed balance which means that for all m^i and m^j:

$$\pi(m^i)\, p_{ij} = \pi(m^j)\, p_{ji}\,. \qquad (2.4)$$

A chain satisfying detailed balance is said to be *reversible*.

Summing Eq. 2.4 over i and using Eq. 2.1 gives

$$\sum_i \pi(m^i) p_{ij} = \sum_i \pi(m^j) p_{ji} = \pi(m^j) \sum_i p_{ji} = \pi(m^j)\,. \qquad (2.5)$$

Note Eq. 2.5 indicates that $\pi(m)$ is a stationary distribution for the Markov chain. In our applications, we will use homogeneous Markov chains and require that the chain be reversible.

The following theorem can be found in Refs. 15 and 17.

Theorem: Let $X^{(n)}$,n=0,1,... be a finite irreducible homogeneous Markov chain with transition probabilities p_{ij} such

$$p_{ii} > 0, \tag{2. 6}$$

for at least one i. If $\pi(m)$ satisfies the detailed balance of Eq. 2.4, then $\pi(m)$ is the unique stationary distribution. Moreover, the Markov chain is *ergodic*, i.e.,

$$\lim_{n\to\infty} p_n(m) = \pi(m), \tag{2. 7}$$

for every probability distribution $p_0(m)$ on $X^{(0)}$.

Eq. 2.7 indicates that in the limit, the states obtained in the Markov chain will represent samples from the probability distribution $\pi(m)$. A Markov chain that is ergodic has a unique invariant distribution, its equilibrium distribution, to which it converges from any initial state. In our application, the state space for m will be the set of all gridblock log-permeability fields. Since in the limit, we will be sampling from the stationary distribution $\pi(m)$, we want to construct transition probabilities so that the detailed balance of Eq. 2.4 is satisfied when $\pi(m)$ is the correct probability distribution for m.

The preceding theorem is for finite Markov chains. Neal[17] indicates, however that the preceding theorem holds for a countably infinite state space, and with appropriate modification for continuous state spaces.

The one we consider is infinite (in fact continuous) in theory, but when implemented on a computer, it is finite since only a finite number of log-permeability fields defined on the gridblocks can be represented on a computer with finite word length. For such problems, Hastings[30](1970) shows that we can use probability density functions in

place of probabilities. Thus, in our application $\pi(m^i)$ will be given by the correct probability density function for the model m which is the field of log-permeabilities defined on the gridblocks.

In constructing the Markov chain, at step n, we will determine a specific m^i, i.e., a state (outcome or realization) of $X^{(n)}$. It is important to note that the basic theorem indicates that as n becomes sufficiently large, the actual states, will approximate realizations or samples of the correct stationary probability distribution $\pi(m)$. Approximate convergence will be obtained when $p_n(m) \approx \pi(m)$. Once this result is obtained, from that point on, all states generated in the Markov chain will represent realizations of the correct probability distribution. In practice, it often takes a large number of transitions to reach the stationary distribution. The early states represent the so called *transient period* and are discarded, since they may not be representative of the equilibrium distribution.

To use the Markov chain Monte Carlo method to generate realizations that are proper samples of some distribution $\pi(m)$, we need to construct a Markov chain which is ergodic, which has $\pi(m)$ as its equilibrium distribution, which converges to this distribution as rapidly as possible, and in which the states visited, once the equilibrium distribution is reached, are not highly dependent.

To construct such a chain for a complex problem, we expect to converge to the stationary distribution in fewer iterations if we begin the Markov chain by sampling from a distribution $p_0(m)$ which is close to $\pi(m)$. Similarly, it is clear that convergence can also be accelerated if our mechanism for proposing new states is such that the probability density function for the set of proposed states is approximately equal to the a posteriori probability density function.

We will present procedures which use both approaches to accelerate convergence of the Markov chains constructed by Monte Carlo techniques. Our methods allow us to construct Markov chains where, after a short transition period, almost all states generated in the chain represent samples from the correct stationary probability distribution.

2.2 Sampling the Log-Permeability Field

In the context of our problem, the state space of the Markov chain is the set of distributions of log-permeabilities on the gridblocks. The model m is the vector of gridblock values of log-permeability. A sequence of states or possible realizations will be a Markov chain provided that the probability of generating a particular new state depends only on the preceding state in the sequence. We denote a particular stochastic realization of m as m^i. Each possible state m^i will have a probability $\pi_i=\pi(m^i)$ associated with it, where π_i is the probability of being in state m^i based on the true probability distribution of gridblock log-permeability values. As mentioned previously, we can use probability density functions for π_i instead of actual probabilities and will do so throughout. However, for simplicity, we will speak of probability density functions as probabilities.

Our objective is to generate a sequence of states (sequence of m^i's) which sample the correct probability density function. Unfortunately, in most real applications, we cannot easily characterize the probability of the ith realization so we must instead use a method that relies only on the relative probabilities to generate realizations. This can be done using Markov chains with a proper choice of the conditional probabilities p_{ij}, where p_{ij} is the probability of generating m^j as the next state in the Markov chain given that the

previous state was m'. The goal is to construct the Markov chain so that after a transition period, states generated in the chain will represent samples (realizations) of the correct probability distribution. As discussed in the preceding section, this requires that the p_{ij} satisfy Eqs. 2.4 and 2.6 (for at least one i) and $p_{ij}>0$.

Metropolis *et al.*[5] were the first to observe that the problem of calculating a permissible transition matrix (matrix with (i,j) entry equal to p_{ij}) could be simplified by requiring that the p_{ij}'s satisfy the reversibility condition or detailed balance of Eq. 2.4. They also proposed partitioning p_{ij} as

$$p_{ij} = \alpha_{ij} q_{ij}, \tag{2.8}$$

where q_{ij} denotes the probability of proposing a transition from state m^i to state m^j and α_{ij} is the probability of accepting the proposed transition m^j as the next state in our Markov chain. In our work, we will propose transitions such that the probability of accepting them is high. We will also refer to m^j simply as state i.

The α_{ij}'s are not uniquely determined by Eqs. 2.4 and 2.8, but one fairly standard choice[30] is

$$\alpha_{ij} = s_{ij}\left[1+\frac{\pi_i q_{ij}}{\pi_j q_{ji}}\right]^{-1}. \tag{2.9}$$

If the detailed balance of Eq. 2.4 holds, it can be shown that the matrix of s_{ij} must be symmetric, i.e., we must require that

$$s_{ij} = s_{ji}, \tag{2.10}$$

for all (i,j). We must also require that

$$0 < s_{ij} \leq 1 + \frac{\pi_i q_{ij}}{\pi_j q_{ji}}, \tag{2. 11}$$

in order to ensure that the probability of accepting the proposed new state m^j is between zero and one, (see Eq. 2.9.) Since $[s_{ij}]$ is symmetric, Eq. 2.11 must also hold when i and j are interchanged, and thus it follows that we must require

$$0 < s_{ij} \leq 1 + \frac{\pi_j q_{ji}}{\pi_i q_{ij}}. \tag{2. 12}$$

Hastings' choice of s_{ij}'s that satisfy Eqs. 2.10, 2.11 and 2.12 is

$$s_{ij} = \min\left\{1 + \frac{\pi_i q_{ij}}{\pi_j q_{ji}}, 1 + \frac{\pi_j q_{ji}}{\pi_i q_{ij}}\right\}. \tag{2. 13}$$

Using Eq. 2.13 in Eq. 2.9 and simplifying, we find that the probability of accepting the proposed transition from state i to state j is given by

$$\alpha_{ij} = \min\left\{1, \frac{\pi_j q_{ji}}{\pi_i q_{ij}}\right\}. \tag{2. 14}$$

Note that if the mechanism for proposing transitions is symmetric, i.e., $q_{ij}=q_{ji}$, then the probability of accepting the proposed state j depends only on the ratio of the probability of state j to the probability of state i. The two-point swapping procedure often used in simulated annealing is an example of a symmetric procedure for proposing transitions.

In our applications, we consider only procedures where Eq. 2.14 applies since this choice is relatively efficient for sampling the desired probability distribution. Our emphasis will be focused on increasing the acceptance probability for proposed transitions q_{ij}. Since

π_i is a fixed, but unknown, function of the permeability distribution, we can only improve the efficiency by improving the process of proposing transitions. The resulting samples of the log-permeability field are conditioned to the prior mean, the prior covariance (variogram) and the multiwell pressure data.

A state i is a vector of M log-permeability values given by

$$m^i = (m_1, m_2, \ldots, m_M)^T, \tag{2. 15}$$

where m_l is the value of $\ln(k)$ in the lth gridblock.

2.2.1 Local Perturbations Based on the Variogram

We let

$$C_M = LL^T, \tag{2. 16}$$

be the Cholesky decomposition of the covariance matrix C_M derived from the variogram. As noted previously, we assume the model is multivariate normal. Thus, states which satisfy the prior mean and prior covariance (variogram) can be generated from

$$m^i = m_0 + LZ^i, \tag{2. 17}$$

where m_0 is the prior mean. In Eq. 2.17, Z^i is an M-dimensional column vector where the components of Z^i are independent standard random normal deviates. The stochastic variable defined by Eq. 2.17 honors the prior mean (m_0) and variogram in the sense that the expected value of m^i is m_0 and the covariance of m^i is C_M. This does not mean that the

variogram computed for any particular m^i generated from Eq. 2.17 will exactly match the model variogram.

Because L is invertable, there is a one-to-one correspondence between m^i and Z^i. Thus, we can refer to either m^i or Z^i as the ith state.

In this subsection, we propose a transition from state i to state j by perturbing only one element of Z^i. Letting

$$Z^i = (Z_1, Z_2, \ldots, Z_k, \ldots, Z_M)^T, \tag{2. 18}$$

we propose a new state by randomly selecting the kth component from the uniform distribution on the set $\{1,2,\ldots,M\}$ and then generating a $Z_k^{'}$ from the standard univariate normal distribution. We denote the proposed new state by

$$Z^j = (Z_1, Z_2, \ldots Z_k^{'}, \ldots, Z_M)^T. \tag{2. 19}$$

Since the Z_l's are independent and only one component is being changed, the probability of proposing a transition from state i to state j is given by

$$q_{ij} = \frac{1}{M\sqrt{2\pi}} \exp\left(-\frac{1}{2}(Z_k^{'})^2\right). \tag{2. 20}$$

Similarly, the probability of proposing a transition from state j to state i is given by

$$q_{ji} = \frac{1}{M\sqrt{2\pi}} \exp\left(-\frac{1}{2}(Z_k)^2\right). \tag{2. 21}$$

The probabilities of state i and state j are specified by the probability density function of Eq. 1.5, i.e., these probabilities are given by

$$\pi(m^i) = a\exp\left[-\frac{1}{2}((m^i - m_0)^T C_M^{-1}(m^i - m_0) + \Delta d^{iT} C_D^{-1}\Delta d^i)\right], \quad (2.22)$$

and

$$\pi(m^j) = a\exp\left[-\frac{1}{2}((m^j - m_0)^T C_M^{-1}(m^j - m_0) + \Delta d^{jT} C_D^{-1}\Delta d^j)\right], \quad (2.23)$$

where

$$\Delta d^i = g(m^i) - d_{obs}, \quad (2.24)$$

for all i. Note Δd^i measures how close pressure data predicted using the log-permeability distribution m^i is to the observed data, and thus $\Delta d^{iT}\Delta d^i$ is sometimes referred to as the pressure mismatch or the mismatch of pressure data. In this and in the next subsection, the pressure mismatch actually refers to $\sqrt{\Delta d^{iT} C_D^{-1}\Delta d^i}$.

Since m^i and m^j satisfy Eq. 2.17 and Eq. 2.16 holds, it is easy to show that

$$\pi_i = \pi(m^i) = a\exp\left[-\frac{1}{2}(Z^{iT}Z^i + \Delta d^{iT} C_D^{-1}\Delta d^i)\right], \quad (2.25)$$

and

$$\pi_j = \pi(m^j) = a\exp\left[-\frac{1}{2}(Z^{jT}Z^j + \Delta d^{jT} C_D^{-1}\Delta d^j)\right]. \quad (2.26)$$

Using Eqs. 2.20, 2.21, 2.25 and 2.26 and performing some simple algebra, it is easy to show that

$$\frac{\pi_j q_{ji}}{\pi_i q_{ij}} = \frac{\exp\left(-\frac{1}{2}\Delta d^{jT} C_D^{-1}\Delta d^j\right)}{\exp\left(-\frac{1}{2}\Delta d^{iT} C_D^{-1}\Delta d^i\right)} = \exp\left(-\frac{1}{2}(\Delta d^{jT} C_D^{-1}\Delta d^j - \Delta d^{iT} C_D^{-1}\Delta d^i)\right). \quad (2.27)$$

By using Eq. 2.27 in Eq. 2.14, we define the probability of accepting the proposed transition from state i to state j when local perturbations are based on the variogram. Eq. 2.27 indicates that when local perturbations are proposed in this way, the acceptance criterion depends only on the relative mismatch of the pressure data. If the proposed log-permeability field generates pressure data that fit the observed data more closely than the previous log-permeability field, then we accept the perturbation and update the field. If the fit is worse, we accept the transition with probability given by Eq. 2.27. This is easily implemented by choosing a random number β uniformly distributed between 0 and 1. If $\beta \leq \pi_j q_{ji}/\pi_i q_{ij}$, then the perturbation is accepted; otherwise the existing value for the parameter is retained, and is included in any time average that are being computed.

It is unnecessary to calculate the misfit to the variogram because all proposed log-permeability fields honor the variogram (prior covariance). We expect that convergence of the Markov chain to be accelerated compared to two-point swaps because all proposed transitions are consistent with our prior information or the probability distribution for the variogram.

Note that the α_{ij}'s and q_{ij}'s (see Eqs. 2.27, 2.14 and 2.20) are such that the corresponding p_{ij} of Eq. 2.8 satisfy $p_{ij}>0$ for all $i \neq j$ and $p_{ii}>0$. Thus, since our selection of the α_{ij} is always such that the detailed balance is satisfied, the conditions of our main theorem are satisfied. This means in the limit, the sequence of m^i's will represent samples from the correct a posteriori distribution for the model, $\pi(m)$.

Since we propose a new state in the Markov chain by randomly selecting and perturbing only one component of the vector of standard normal deviates, the proposed state m^j will not differ significantly from the previous accepted image m^i in the Markov

chain. Thus, the data mismatch for m^i and m^j may be quite similar and Eqs. 2.27 and 2.14 then indicate the probability of accepting the proposed image will be close to one. Thus, the local perturbation procedure will result in a high acceptance rate.

Recall that the ultimate goal is usually not to simply generate possible reservoir models but draw inferences based on predictions from realizations. These predictions are often computationally expensive so it is usually better to have 10-100 independent realizations from the a posteriori distribution than to have 10,000 dependent realizations. Unfortunately, we do not just want to generate a high number of samples of the correct probability density function. We want the samples to be independent. Since two successive state Z^i and Z^j will have all components identical but one, the two states will not be independent. Thus, we wish to sort through the states generated in the Monte Carlo Markov chain and choose a subset of independent images.

The effect of dependencies on the accuracy of a Monte Carlo estimate can be quantified in terms of the experimental variogram obtained from the values of $S(m_t)$, i.e., the absolute value of the argument of the exponential in Eq. 1.5 for the tth state at the subset of states for which the transitions of the chain were accepted once equilibrium is reached[12]. The experimental variogram is calculated from

$$\gamma(s) = \frac{1}{2N_p} \sum_{t=0}^{N_p-1} (S(m_t) - S(m_{t+s}))^2 , \qquad (2.28)$$

where m_t, $0 \le t \le N_{p-1}$ is the set of accepted images.

Because the dependence decreases with 'lag', an approximately independent set of realizations can be obtained by retaining every rth state from the chain, where r is the value of the lag for which the autocorrelation is approximately zero. Once this lag r has

been determined we can thin the set of accepted states $\{m_t\}_{t=1}^{N_p}$ m_{i+jr}, $j = 1,2,...l$ where l is the maximum integer such that $i + jr \leq N_p$ and m_i is the first state accepted which represents a sample from the stationary distribution. The thinned set of realizations should represent independent samples of the a posteriori probability density function. However, because this procedures assumes that if $S(m_t)$ and $S(m_{t+j})$ are uncorrelated for $j \geq r$, then m_t and m_{t+j} are also uncorrelated when $j \geq r$, thus our procedure is only approximate.

2.2.2 Global Perturbations Based on the Variogram

To overcome the problem of generating independent states, we can simply perturb every component of Z when proposing transitions. In this case, we propose a transition from state Z^i to state Z^j by generating a new random normal deviate for each component of the vector. In this case, Z^j is independent of Z^i and the probability of proposing the transition from state i to state j is given by

$$q_{ij} = (2\pi)^{-M/2} \exp\left(-\frac{1}{2} Z^{jT} Z^j\right), \tag{2. 29}$$

and the probability of proposing the reverse transition is

$$q_{ji} = (2\pi)^{-M/2} \exp\left(-\frac{1}{2} Z^{iT} Z^i\right). \tag{2. 30}$$

The relation between a Z^i and the state m^i is still given by Eq. 2.17 and Z^j and m^j satisfy the same equation with i replaced by j. Moreover, it is easy to show that Eq. 2.27 still holds so the probability of accepting the proposed state m^j is still determined by the

pressure mismatch. The detailed balance of Eq. 2.4 applies and the transition probabilities (p_{ij}'s of Eq. 2.8) still satisfy the conditions that guarantee convergence of the Markov chain to the correct stationary distribution.

The advantage of local perturbations is that a large number of proposed transitions will be accepted, but the disadvantage is that many of them will not be independent. The advantage of global perturbations is that every accepted transition results in an independent state; the disadvantage is that significantly fewer proposed transitions will be accepted.

2.2.3 Perturbations Based on Sensitivity Functions

Transitions proposed from the prior covariance matrix honor the variogram automatically, but are not directly related to the pressure mismatch part of the a posteriori probability density function. Thus, it seems clear that we can further increase the probability of acceptance by proposing transitions that at least approximately represent samples from the a posteriori probability density function. Sensitivity coefficients provide a means to propose such transitions.

To do this, we choose a reference state m_r and expand $g(m)$ in a Talyor series about m_r to obtain

$$g(m) = g(m_r) + G_r(m - m_r) + \varepsilon(m). \qquad (2.31)$$

Here the elements of the matrix G_r are simply the sensitivity coefficients evaluated at m_r; see Refs. 3 and 31. From Eq. 2.31, it follows that

$$\varepsilon(m) = g(m) - g(m_r) - G_r(m - m_r). \tag{2. 32}$$

We define corresponding reference data d_r by

$$d_r = d_{obs} - g(m_r) + G_r m_r. \tag{2. 33}$$

From Eqs. 2.32 and 2.33, it follows that

$$g(m) - d_{obs} = G_r m + \varepsilon(m) - d_r. \tag{2. 34}$$

The a posteriori probability density function of Eq. 1.5 can now be written as

$$\pi(m) = a\exp\left[-\frac{1}{2}((m-m_0)^T C_M^{-1}(m-m_0) + (d_r - G_r m)^T C_D^{-1}(d_r - G_r m))\right]\times$$

$$\exp\left[(d_r - G_r m)^T C_D^{-1}\varepsilon(m) - \frac{1}{2}\varepsilon(m)^T C_D^{-1}\varepsilon(m)\right]. \tag{2. 35}$$

We now define a new reference "mean" and reference covariance matrix, respectively, by

$$\mu_r = m_0 + C_M G_r^T (G_r C_M G_r^T + C_D)^{-1}(d_r - G_r m_0), \tag{2. 36}$$

and

$$C_{Mr} = (G_r^T C_D^{-1} G_r + C_M^{-1})^{-1}. \tag{2. 37}$$

Using basic results from inverse problem theory[1], Eq. 2.35 can be rewritten as

$$\pi(m) = a_r \exp\left[-\frac{1}{2}(m-\mu_r)^T C_{Mr}^{-1}(m-\mu_r)\right]\times$$

$$\exp\left[(d_r - G_r m)^T C_D^{-1}\varepsilon(m) - \frac{1}{2}\varepsilon(m)^T C_D^{-1}\varepsilon(m)\right], \tag{2. 38}$$

where a_r is a constant.

Denote the Cholesky decomposition of C_{Mr} by

$$C_{Mr} = L_r L_r^T. \tag{2.39}$$

Now we propose transitions by sampling from a multivariate normal distribution based on C_{Mr} and the reference mean μ_r, i.e., proposed transitions are generated from

$$m^j = \mu_r + L_r Z^j, \tag{2.40}$$

where Z^j still represents a vector of independent standard random normal deviates. If transitions are proposed by resampling all components of the vector of normal deviates, then the probability of proposing a transition from state i to state j is given by

$$q_{ij} = (2\pi)^{-M/2} \exp\left(-\frac{1}{2} Z^{jT} Z^j\right). \tag{2.41}$$

Using Eqs. 2.39 and 2.40, Eq. 2.41 can be rewritten as

$$q_{ij} = (2\pi)^{-M/2} \exp\left(-\frac{1}{2}[L_r^{-1}(m^j - \mu_r)]^T [L_r^{-1}(m^j - \mu_r)]\right) =$$

$$(2\pi)^{-M/2} \exp\left(-\frac{1}{2}(m^j - \mu_r)^T (L_r L_r^T)^{-1} (m^j - \mu_r)\right) =$$

$$(2\pi)^{-M/2} \exp\left(-\frac{1}{2}(m^j - \mu_r)^T C_{Mr}^{-1} (m^j - \mu_r)\right). \tag{2.42}$$

Thus, when a new state is proposed from Eq. 2.40 by globally perturbing the vector of random normal deviates, the proposed state honors the covariance C_{Mr} and reference mean μ_r. Similar to Eq. 2.42, we find that the probability of proposing a transition from state j to state i is given by

$$q_{ji} = (2\pi)^{-M/2} \exp\left(-\frac{1}{2} Z^{iT} Z^i\right) = (2\pi)^{-M/2} \exp\left(-\frac{1}{2}(m^i - \mu_r)^T C_{Mr}^{-1} (m^i - \mu_r)\right). \tag{2.43}$$

From Eq. 2.38, it follows that

$$\pi(m^i) = a_r \exp\left[-\frac{1}{2}(m^i - \mu_r)^T C_{Mr}^{-1}(m^i - \mu_r)\right] \times$$

$$\exp\left[(d_r - G_r m^i)^T C_D^{-1}\varepsilon(m^i) - \frac{1}{2}\varepsilon(m^i)^T C_D^{-1}\varepsilon(m^i)\right], \qquad (2.44)$$

and

$$\pi(m^j) = a_r \exp\left[-\frac{1}{2}(m^j - \mu_r)^T C_{Mr}^{-1}(m^j - \mu_r)\right] \times$$

$$\exp\left[(d_r - G_r m^j)^T C_D^{-1}\varepsilon(m^j) - \frac{1}{2}\varepsilon(m^j)^T C_D^{-1}\varepsilon(m^j)\right]. \qquad (2.45)$$

From Eqs. 2.41 through 2.45, it follows that

$$\frac{\pi_j q_{ji}}{\pi_i q_{ij}} = \frac{\exp\left[(d_r - G_r m^j)^T C_D^{-1}\varepsilon(m^j) - \frac{1}{2}\varepsilon(m^j)^T C_D^{-1}\varepsilon(m^j)\right]}{\exp\left[(d_r - G_r m^i)^T C_D^{-1}\varepsilon(m^i) - \frac{1}{2}\varepsilon(m^i)^T C_D^{-1}\varepsilon(m^i)\right]}. \qquad (2.46)$$

Using Eqs. 2.32, 2.33 and 2.34, Eq. 2.46 can be rewritten as

$$\frac{\pi_j q_{ji}}{\pi_i q_{ij}} = \frac{\exp\left[(d_{obs} - g(m^j))^T C_D^{-1}\varepsilon(m^j) + \frac{1}{2}\varepsilon(m^j)^T C_D^{-1}\varepsilon(m^j)\right]}{\exp\left[(d_{obs} - g(m^i))^T C_D^{-1}\varepsilon(m^i) + \frac{1}{2}\varepsilon(m^i)^T C_D^{-1}\varepsilon(m^i)\right]}. \qquad (2.47)$$

Using Eq. 2.46 or 2.47 in Eq. 2.14 defines Hastings' acceptance probabilities for global perturbations based on sensitivity matrices.

One can also do local perturbations based on the reference model and its associated reference mean and covariance matrix. Given state m^i in the Markov chain Monte Carlo, as before a local perturbation corresponds to randomly selecting one component of Z^i and

replacing it by a random standard normal deviate to obtain the proposed state Z^j. This method is referred to as local perturbations based on the sensitivity coefficients. Eqs. 2.46 and 2.47 still apply.

If $g(m)$ is approximately linear around a reference model m_r , then $\varepsilon(m)\approx 0$ in Eq. 2.34. If all proposed perturbations are such that the proposed log-permeability field m^j yields pressures close to the observed pressure data, then $d_{obs}-g(m^j)\approx 0$. When these two conditions are satisfied, Eq. 2.47 indicates that

$$\frac{\pi_j q_{ji}}{\pi_i q_{ij}} \approx 1, \tag{2.48}$$

and Hastings' probability of acceptance also satisfies

$$\alpha_{ij} \approx 1, \tag{2.49}$$

for all (i,j). This suggests we use the maximum a posteriori estimate, m_∞, generated from the gradient method[2-4] as m_r and use the associated a posteriori covariance matrix C_{MP} as C_{Mr}. Again we can do either local or global perturbations.

When considering perturbation techniques based on sensitivities we implemented two methods. The first method refers to proposing global perturbations with $m_r=m_\infty$ and $C_{Mr}=C_{MP}$. Perturbations are proposed by selecting a new vector of independent standard normal deviates and proposing a new state from Eq. 2.40. The second method refers to local perturbations with $m_r=m_\infty$ and $C_{Mr}=C_{MP}$. Given state m^i corresponding to Z^i, we propose a new state as follows: we randomly select one component of Z^i and replace it with a new standard random normal deviate; this gives Z^j, and the new proposed state is then generated from Eq. 2.40.

Again global perturbations guarantee that the states of the computed Markov chain are independent, but results in far fewer transitions accepted. For both types of perturbations, the conditions for convergence of the Markov chain Monte Carlo are theoretically satisfied.

A third method that could be considered, but was not investigated in this work, corresponds to using global or local perturbations based on a reference model and corresponding sensitivity coefficients generated from the simulator. In this case, the reference model always corresponds to some previous state m^i in the Markov chain. Whenever, a new image m^j with a smaller pressure mismatch than is obtained with m^i is accepted, the reference model can be updated to m^j. If m_r, is continuously updated, the resulting Markov chain is not homogeneous. Thus, m_r should only be updated until achieving the following pressure misfit:

$$\sqrt{\frac{(g(m_r)-d_{obs})^T(g(m_r)-d_{obs})}{N_d}} \leq \sigma_d, \qquad (2.50)$$

where N_d is the number of pressure data and σ^2_d is the variance of each pressure measurement error. Once a reference model satisfies Eq. 2.50, it remains fixed, and thus from this point on, a homogeneous Markov chain is generated. There are several other procedures that could be used, e.g., we could actually use a cycle of reference models continually repeated.

There is an important feature of starting a Markov chain from a state given by Eq. 2.40 where m_r and C_{Mr} are m_∞ and C_{MP} obtained from a gradient method, or where m_r is given by simulated annealing (see Chapter III) and satisfies Eq. 2.50. As stated before acceptance of a perturbation based on Hasting's acceptance criterion, Eq. 2.14, does not

necessarily mean that the resulting state is drawn from the correct probability distribution. There is usually a transient period at the beginning of a Markov chain that must be discarded on the grounds that equilibrium has not been reached. As illustrated by examples presented in the next subsection, for the Markov chain Monte Carlo methods that use C_{MP} to generate perturbations, the transient period is so short that it is difficult to see any evidence of it.

2.3 Computational Examples

2.3.1 Example 1 - Variance = 0.25 Case

Here, we apply the techniques described in the previous subsections for sampling from the correct a posteriori probability density function. All computational times given in all chapters are for implementations written in FORTRAN and run on a Pentium-90. All codes used were written with reasonable attention to efficiency, but were not carefully tuned to achieve maximum computational efficiency.

The reservoir is assumed to be a rectangular parallelepiped of uniform thickness h. We assume that the reservoir is produced by a fully-penetrating well located at the center. There exist four wells which are shut-in at all times and constitute observation or interference wells. We ignore gravity and capillary effects and consider only two-dimensional (x-y) single-phase flow. The active well is produced at a constant rate. We assume pressure data is measured not only at the active well but also at the observation wells. Pressure responses are obtained by a finite-difference simulator, i.e., a simulator is

used to generate synthetic multiwell pressure data and is also used at each iteration of the Markov chain Monte Carlo methods to calculate the pressure data resulting from a proposed perturbation of the permeability field.

The permeability field is assumed to be heterogeneous, but isotropic. It is assumed that gridblock log-permeabilities can be modeled as random variables with log-normal permeability distribution with known mean and variance and that the associated random function is stationary. It is also assumed that the spatial variability in permeability can be characterized by a known isotropic variogram (covariance function).

As is well known, unconditional simulation can be used to generate realizations honoring the variogram, or more precisely the covariance function, from a *LU* decomposition of the covariance matrix[32,33]. For the examples presented, the "true" permeability field is given as an input and the simulator is run to generate synthetic well test pressure data. This provides our observed or measured pressure data.

In the synthetic example considered here, a 1,500 ft x 1,500 ft square reservoir with no flow boundaries was used. Reservoir performance was simulated using a uniform spatial grid with 100 ft by 100 ft gridblocks. The total number of gridblocks is $M = 225$. It was assumed that porosity is constant with $\phi = 0.2$. We assume a slightly compressible fluid at constant compressibility and viscosity with $\mu = 0.5$ cp and $c_t = 10^{-5}$ psi^{-1}. All well bore radii are given by $r_w = 0.3$ ft. In all cases, reservoir thickness is $h = 100$ ft and initial pressure is $p_i = 6{,}000$ psi. The permeability distribution is assumed to be log-normal with log variance of 0.25 and log mean equal to 3.4. We assume the permeability field is isotropic and spatial continuity can be described by a spherical variogram model with range equal to 600 ft.

The well locations for the five-well system considered is shown in Fig. 2.4.1 and rates shown in Table 2.4.1. The location column in Table 2.4.1 gives the indices of the gridblock location with the first index representing the x-direction index and the second the y-direction index. The x-direction index increases from left to right and the y-direction index increases from bottom to top. (This same ordering of indices is used in the presentation of all realizations of the ln(k) field shown in the figures.) We condition to the drawdown pressures at the wells using pressure data from 0.05 to 5.7 days. The variance of all pressure measurement errors was assumed to be given by $\sigma^2_d = 0.15$ psi^2. At each well, we use 10 pressure data.

TABLE 2.4.1

Locations and Flow Rates for Five-Well System

Well Number	Location	Rate (RB/D)
1	(4,4)	0
2	(12,4)	0
3	(8,8)	750
4	(7,12)	0
5	(12,12)	0

The "true" permeability distribution represents an unconditional simulation generated from a Cholesky decomposition of the prior covariance matrix. This truth case

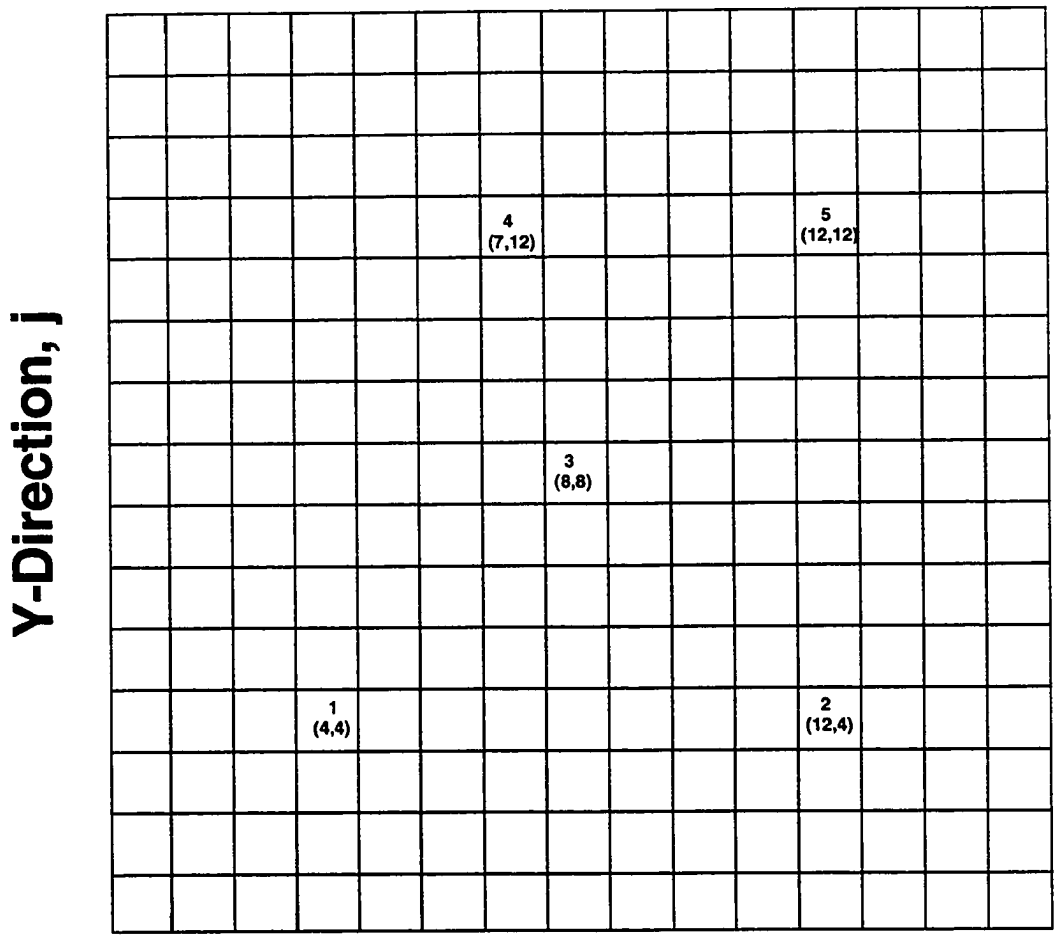

Fig. 2.4.1 - Well Numbers and Well Locations.

is shown in Fig. 2.4.2. The values of $\ln(k)$ shown range from 2.0 to 5.0, i.e., in terms of k, the scale ranges from 7 to 148 md.

Figure 2.4.3 shows a comparison between the local variogram obtained with the "true" log-permeability distribution of in Fig. 2.4.2 and the theoretical variogram model. Note the sill of the local variogram is less than the sill of the model variogram.

In this example, we compare three different Markov chain Monte Carlo methods used to sample the a posteriori probability distribution. In the first method considered, the proposed perturbations are local and are based on the Cholesky decomposition of the prior covariance matrix, see Eq. 2.17, so they honor only the variogram. In the second and third methods, the sensitivity coefficients corresponding to the maximum a posteriori estimate are used to modify the direction and the magnitude of the proposed perturbations; see Eq. 2.40. The perturbations in this second method are global; in the third method they are local, meaning that the permeability in every gridblock is modified in every transition.

One way to compare the three methods is to look at the rate of generation of acceptable independent realizations for each. Fig. 2.4.4 shows the values of the "objective function", (i.e., the argument of the exponential in Eq. 1.5) versus the perturbation number. Figure 2.4.5 shows a similar plot, but in this figure, the pressure misfit versus the perturbation number is plotted. In all the examples, the pressure misfit actually refers to $\sqrt{\Delta d^{iT} \Delta d^i / N_d}$ where N_d refers to the total number of pressure data used as conditioning data. The perturbation number in Figs. 2.4.4 and 2.4.5 refers to the number of transitions proposed, but only the values of accepted transitions are plotted. The state remains unchanged between accepted transitions, but those states have not been displayed.

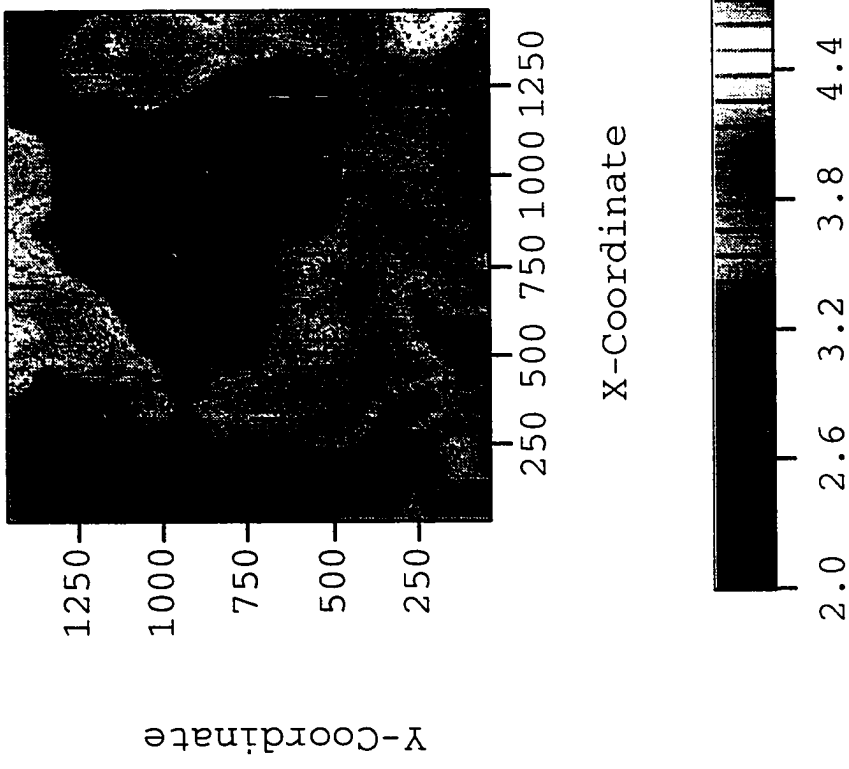

Fig. 2.4.2 - True Log-Permeability Distribution (Variance=0.25).

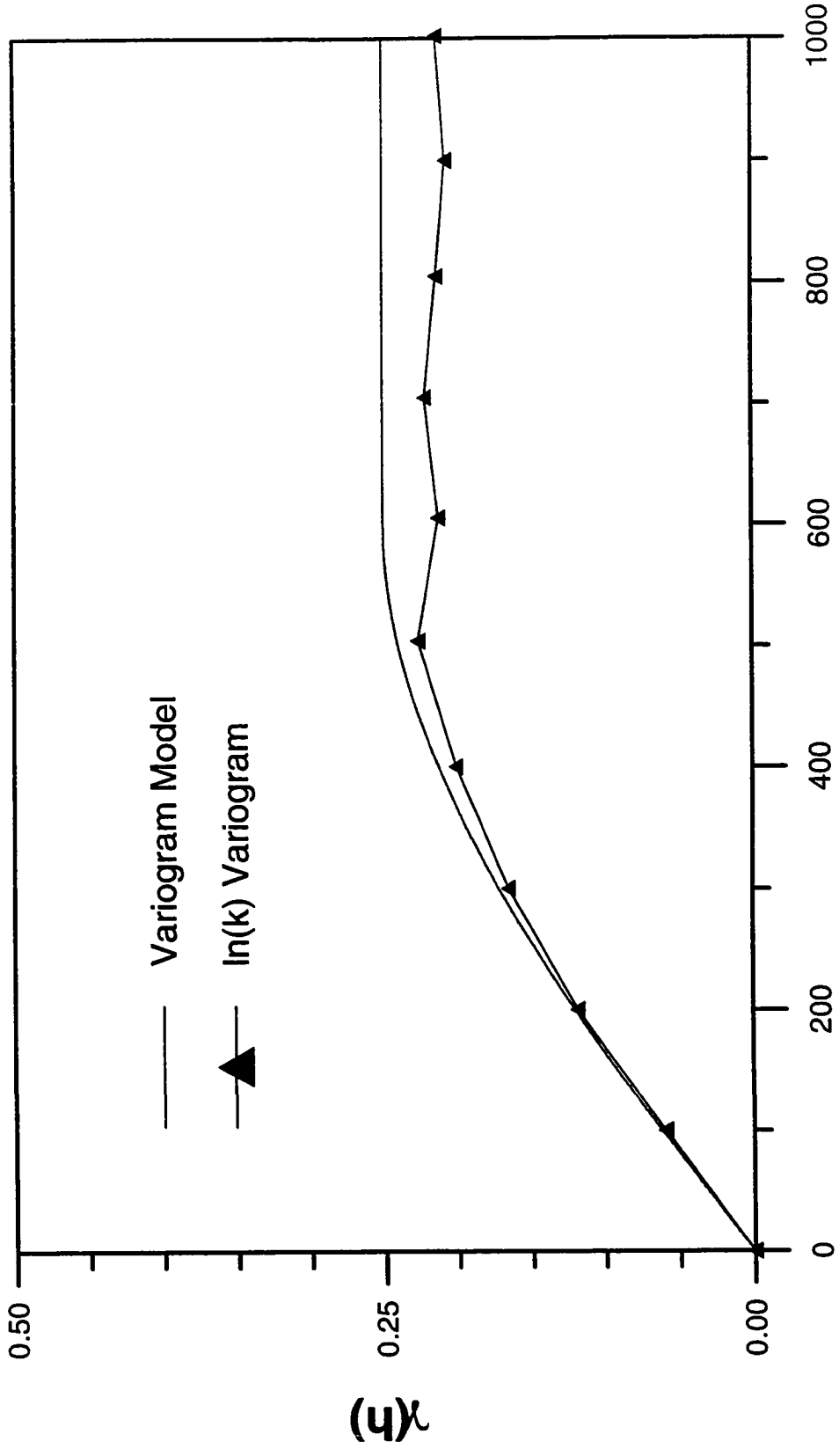

Fig. 2.4.3 - Local Variogram from Truth Case
(Sill=0.25 and Range=600ft).

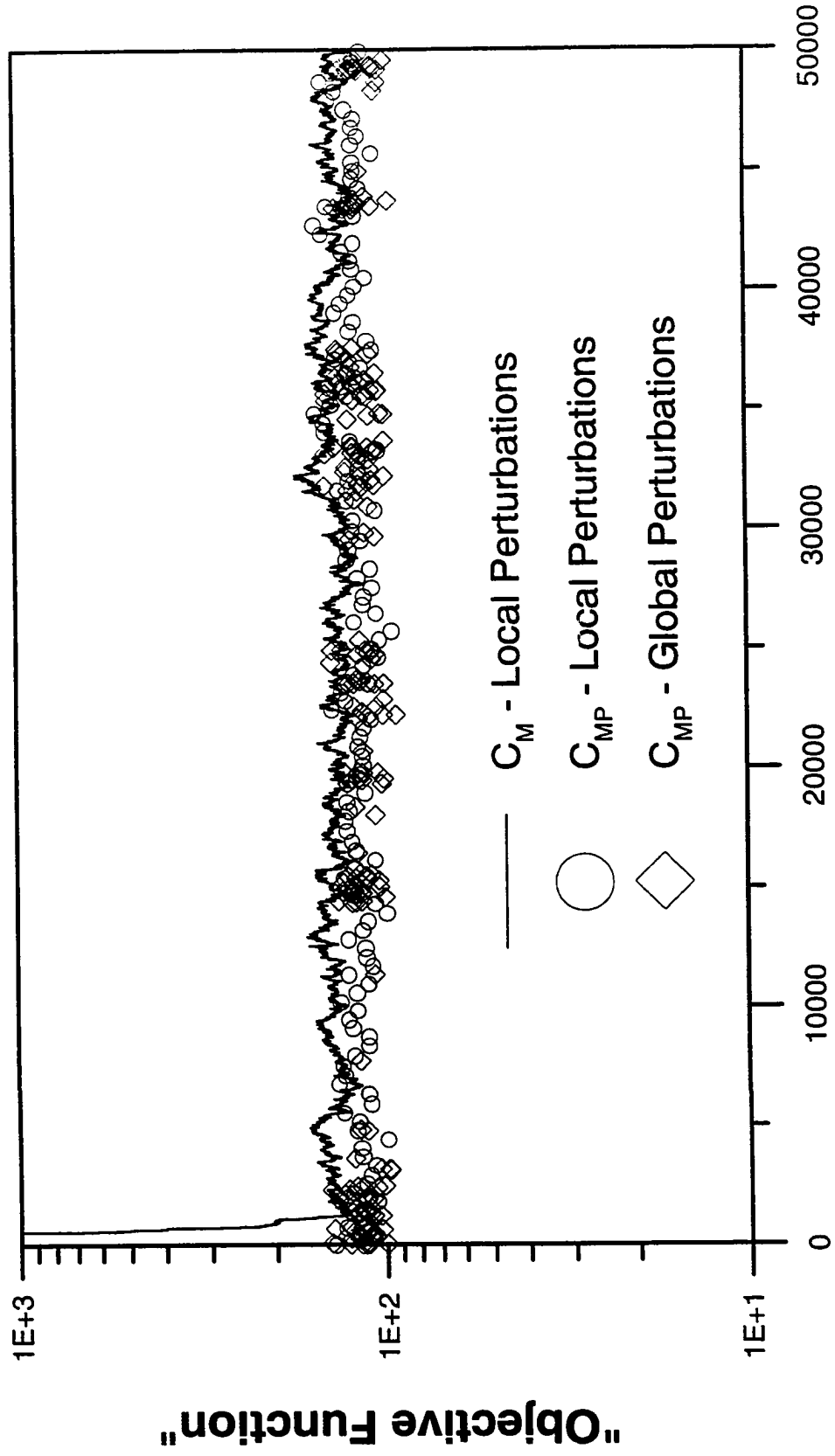

Fig. 2.4.4 - Values of the Objective Function for the Three Markov Chain Monte Carlo Methods (Variance = 0.25).

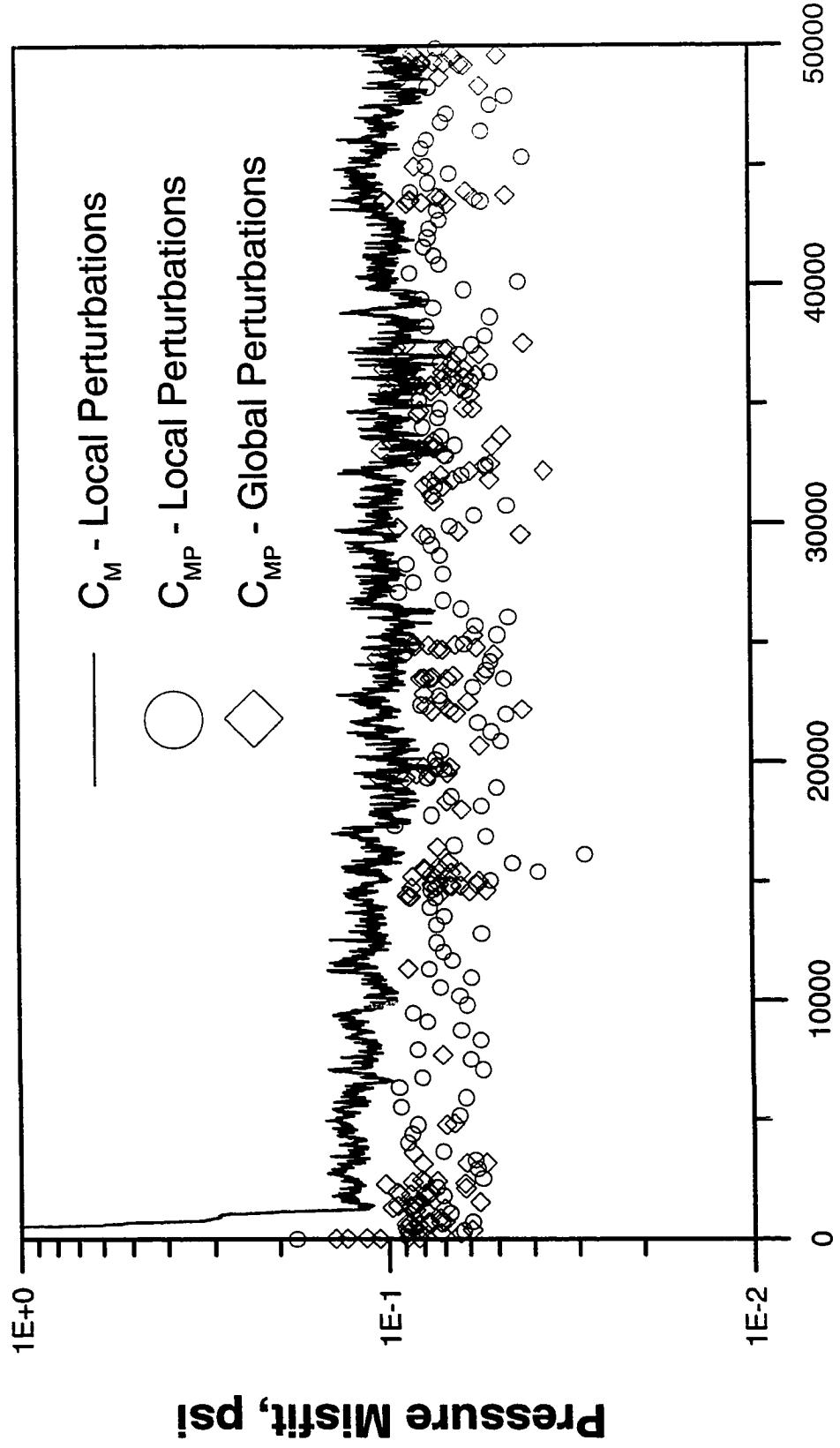

Fig. 2.4.5 - Values of the Pressure Misfit for the Three Markov Chain Monte Carlo Methods (Variance = 0.25).

The results of Figs. 2.4.4 and 2.4.5 allow us to discuss procedures for determining when the transient period of the Markov chain has ended. Recall that based on our previous discussions, the states generated early in the Markov chain Monte Carlo may not correspond to samples from the stationary distribution. As we continue to generate states, these states will eventually represent realizations of the stationary distribution, i.e., the a posteriori probability density function. As convergence to the stationary distribution is only guaranteed in the limit, i.e., as the number of states generated goes to infinity, it is important to have a practical method to decide when states correspond to realizations of the correct probability distribution. When we reach the stationary distribution, the values of the objective function at the states will appear to oscillate around a constant value so the overall plot will appear relatively constant, whereas, during the transient period, the plot of the objective function, though oscillatory, will show a decreasing trend.

For the Markov chain Monte Carlo methods that use the a posteriori covariance obtained from the gradient method to generate perturbations, the transient period is so short that it is difficult to see any evidence of it in Fig. 2.4.4. It might appear, from Fig. 2.4.4 that the transient period for the first Markov chain Monte Carlo method, which used local perturbations based only on the variogram, is also fairly short; however, as discussed below, this does not appear to be the case. Thus, we have not reached the stationary distribution. One way to check whether we have reached a stationary distribution is to consider the pressure misfit of Fig. 2.4.5. If all data are consistent, we should be able to obtain many states which give pressure mismatches less than the standard deviation of the pressure measurement errors, i.e., pressure mismatches less than 0.15 psi. However, the vast majority of the states generated using local perturbations based on the variogram do

not satisfy this pressure tolerance. This suggests that we did not reach the stationary distribution.

When the stationary distribution of the Markov chain is reached, the values of the objective function cluster around a value greater than or equal to $M/2$ where M is the number of model parameters. Because the pressure data used in these examples to calculate the maximum a posteriori model were exact, and the prior model was close to the a posteriori model, the average value of the objective function in the stationary phase should be approximately equal to $M/2 = 112.5$. The observed mean of the objective function, computed over the set of accepted realizations for the method that uses local perturbations based on the a posteriori covariance is 119 with a corresponding estimate of the standard deviation given by 10. The method that uses global perturbations based on the a posteriori covariance resulted in a mean value of 111 and a standard deviation of 10.

Figures 2.4.7 and 2.4.8 show log-permeability values at four gridblock positions (see locations depicted in Fig. 2.4.6) versus the perturbation number for the Markov chain Monte Carlo methods that used global and local perturbations based on the a posteriori covariance matrix. As before, the perturbation number in these figures refers to the number of transitions proposed, but only the values of accepted transitions are plotted. Regarding the areal location of the log-permeability gridblocks, we have a first location at one of the observation well gridblocks (X1), a second at the producing well gridblock (X2), a third location between the producing well gridblock and one observation well (X3) and a fourth almost at the reservoir boundary (X4). The a posteriori variance for the first, second third and fourth gridblocks were 0.17, 6×10^{-6}, 0.07 and 0.24, respectively. The

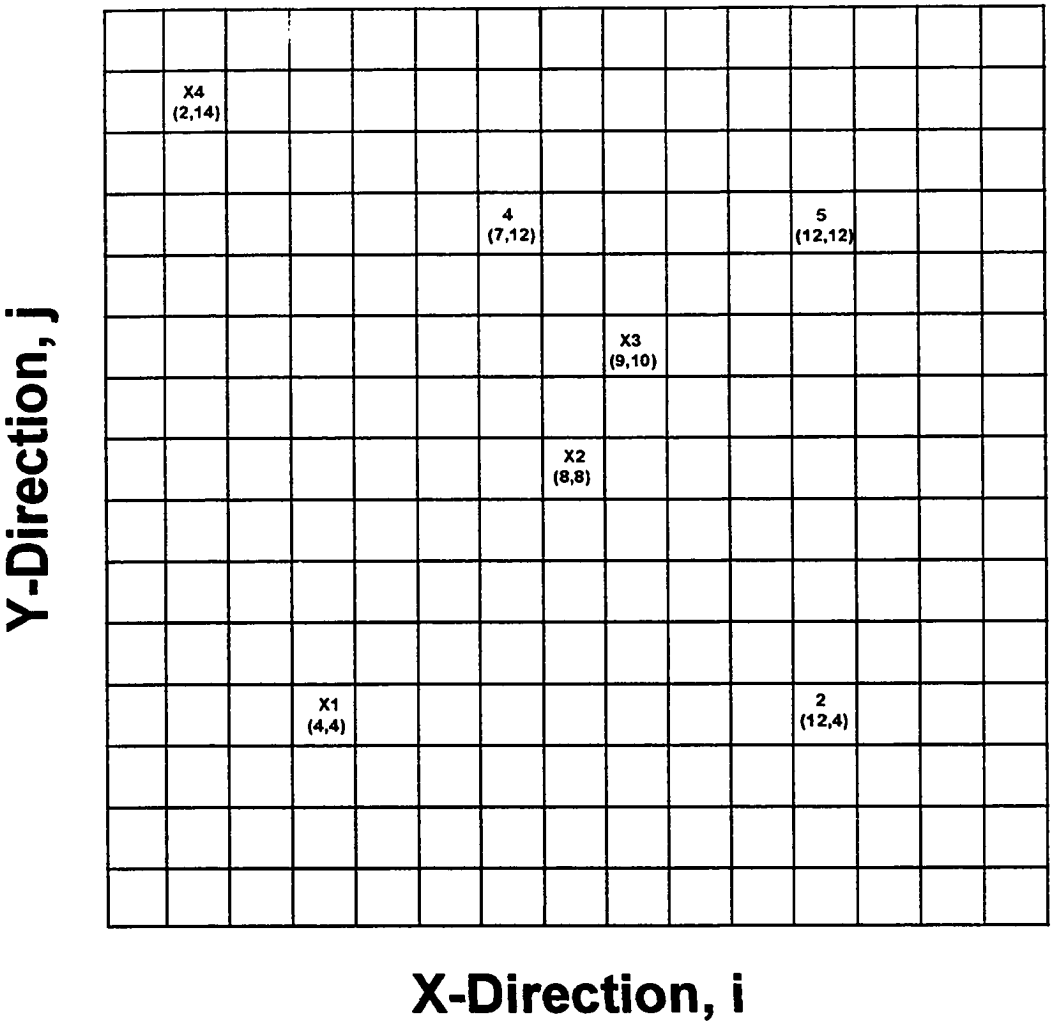

Fig. 2.4.6 - Grid Locations for Log-Permeability Plot.

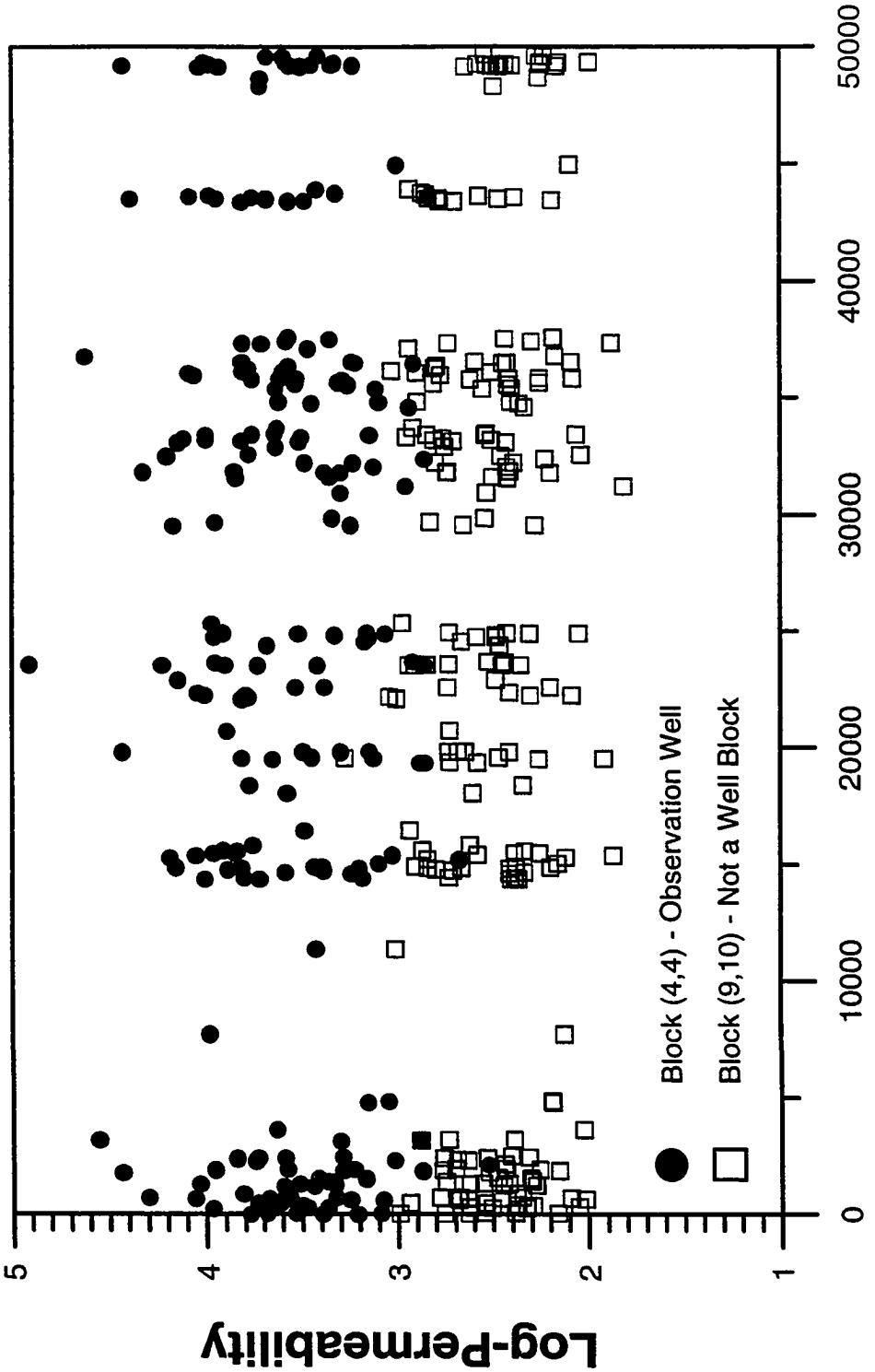

Fig. 2.4.7a - Log-Permeability Values During Sampling with Global Perturbations Based on the a Posteriori Covariance Matrix (Variance = 0.25).

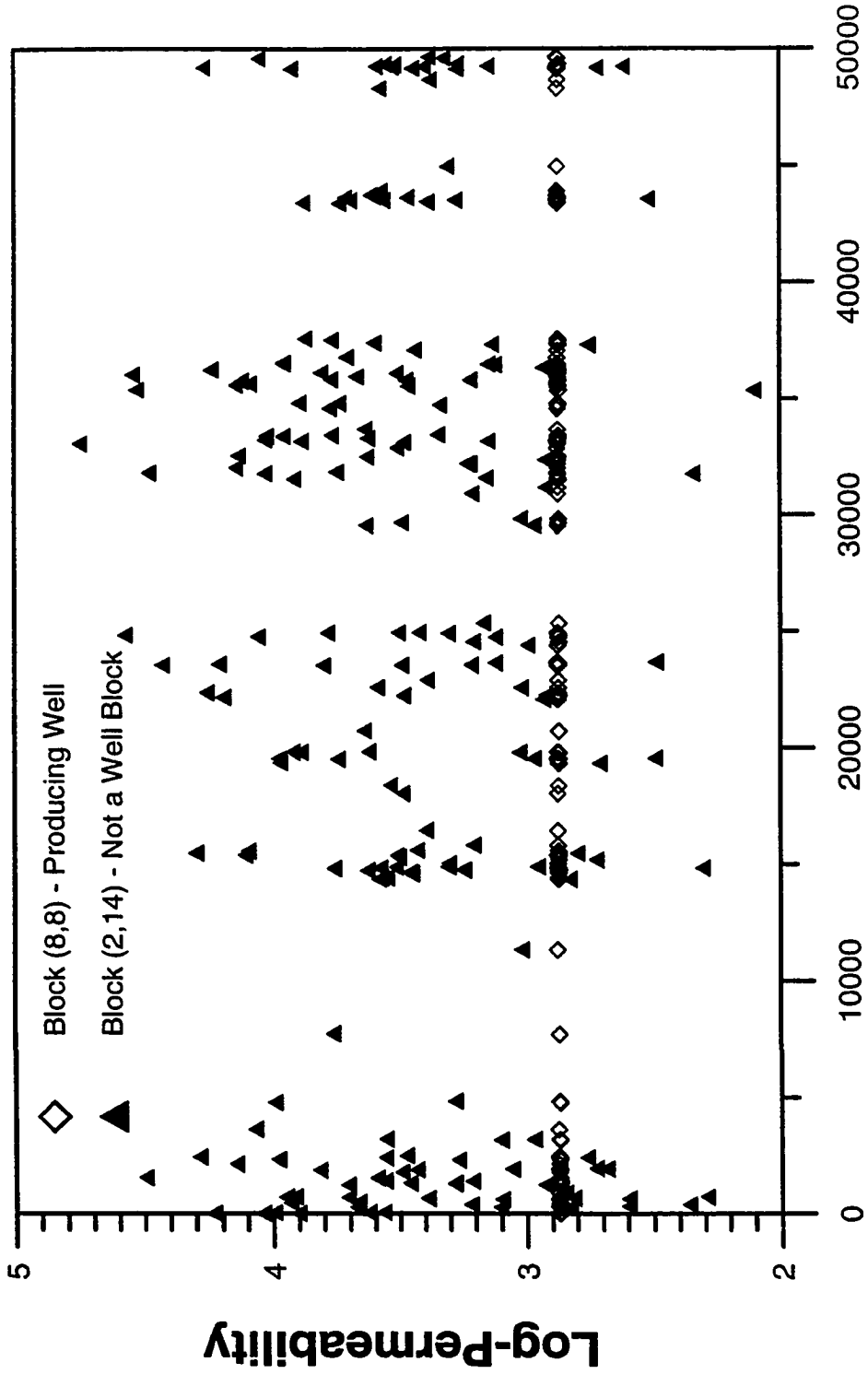

Fig. 2.4.7b - Log-Permeability Values During Sampling with Global Perturbations Based on the a Posteriori Covariance Matrix (Variance = 0.25).

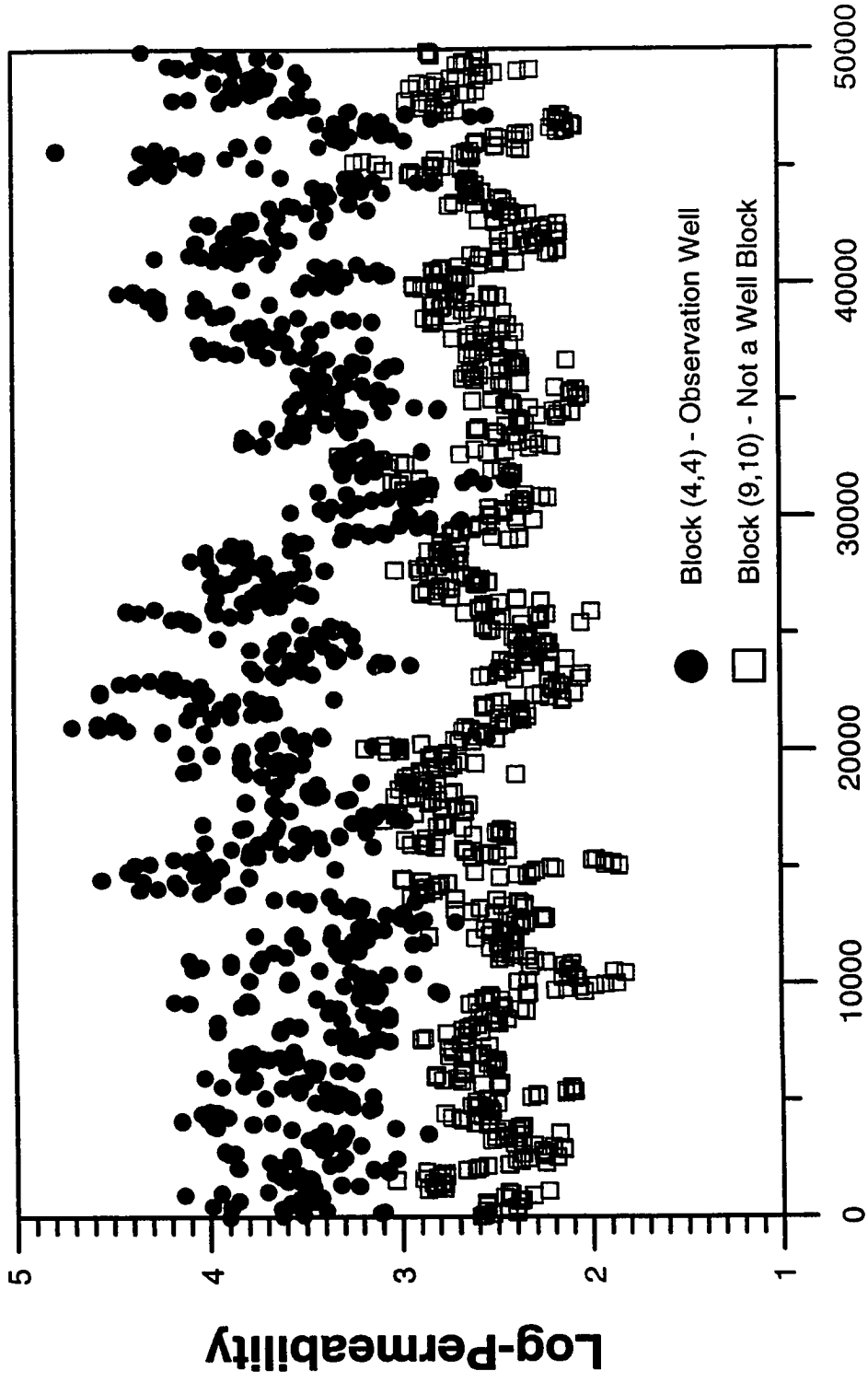

Fig. 2.4.8a - Log-Permeability Values During Sampling with Local Perturbations Based on the a Posteriori Covariance Matrix (Variance = 0.25).

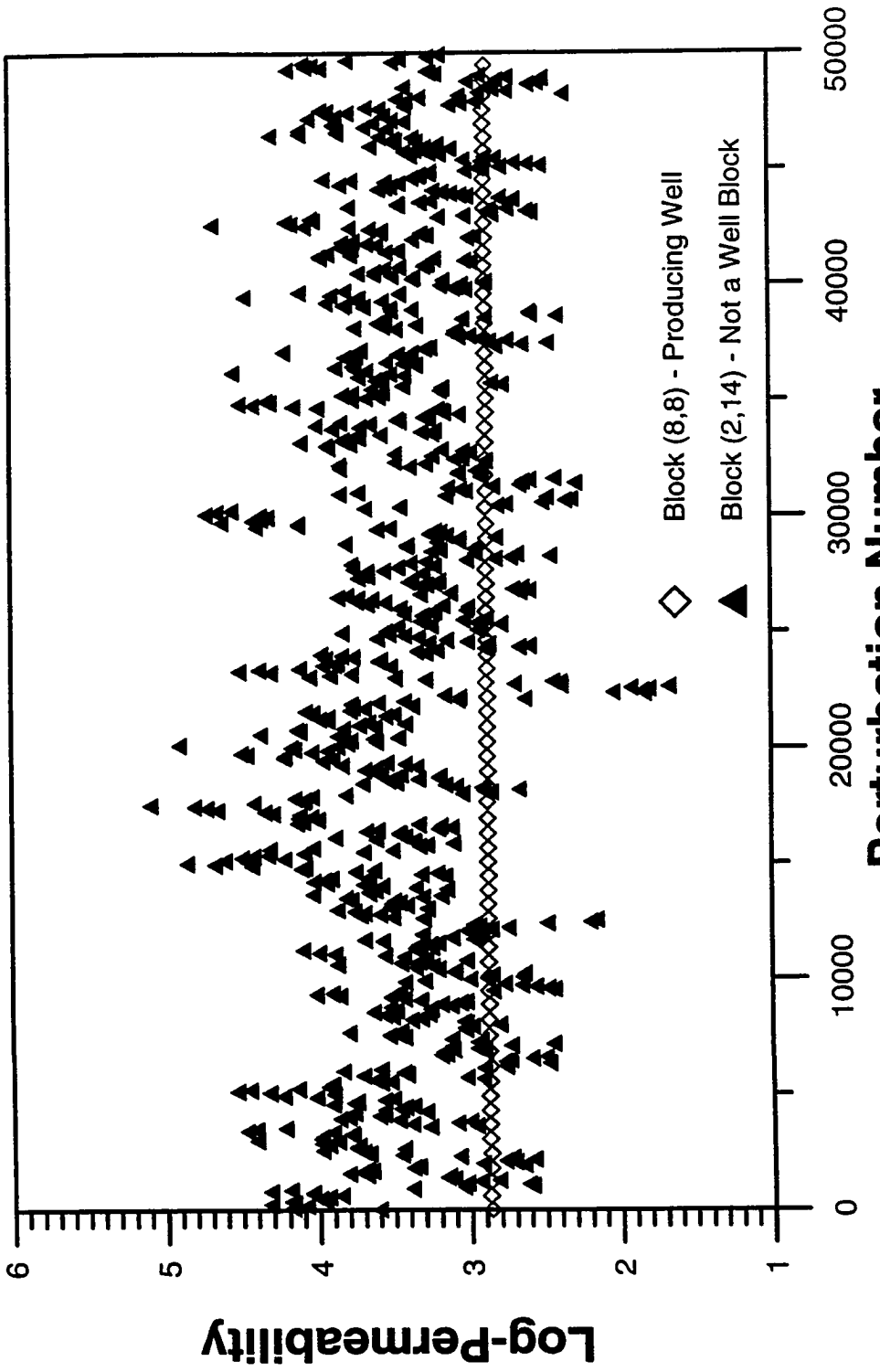

Fig. 2.4.8b - Log-Permeability Values During Sampling with Local Perturbations Based on the a Posteriori Covariance Matrix (Variance = 0.25).

maximum a posteriori estimates of log-permeability for the same gridblocks were 3.68, 2.87, 2.54 and 3.43, respectively.

When looking to the behavior of the sampling of permeability at each block we notice that the same features are common when using global and local perturbations. It is possible to visualize the dominant influence that the a posteriori variances have on the variability in realizations generated during the Markov chain Monte Carlo procedure. The sampled value for the producing well gridblock location stays almost constant and equal to the value encountered for the maximum a posteriori estimate for that gridblock through the whole simulation. This reflects the fact that at the producing well location the uncertainty is considerably reduced when incorporating pressure data together with the prior model information. This can be seen by the very low value encountered for the a posteriori variance.

At the gridblock between the producing and interference well, the pressure resolves quite well the uncertainty (note the low value for the a posteriori variance.) However, a larger variation in the log-permeability values for this gridblock during the sampling procedure when compared to the obtained to the producing well gridblock is expected to appear. This can be observed in Figs. 2.4.7 and 2.4.8.

On the other hand, at the observation well gridblock (X1), since the pressure data is not sufficient to reduce the uncertainty in the permeability estimate (see the higher value for the a posteriori variance when compared with the producing well gridblock; almost the original prior variance of 0.25) as well as in the producing well gridblock, we can see that a large variation in the log-permeability estimates during the sampling process occurs. This is also the case for the gridblock located close to the corner of the reservoir.

As depicted in Fig. 2.4.8 the permeability values sampled with the local perturbation method are highly correlated.

In relation to the number of images accepted, the method that used local perturbations based on the variogram did not generate any legitimate images of the permeability field during the allowed sampling period. The Markov chain Monte Carlo method that used global perturbations based on the a posteriori covariance matrix generated the total number of 200 distinct, independent realizations. Using local perturbations based on the a posteriori covariance matrix we obtained approximately 37500 distinct, but highly correlated images.

Recall that the ultimate goal is usually not to simply generate possible reservoir models, but to draw inferences based on predictions from the realizations. These predictions are often computationally expensive so its better to have fewer number of independent images than a multitude of dependent realizations. As described earlier in this chapter, we can use the variogram of S (the objective function variogram) to thin dependent images produced in a Markov chain sampling process. Because the dependence decreases with "lag", it is possible to calculate the lag for which the autocorrelation is approximately zero and based on this value thin the set of dependent images to obtain an independent group of realizations.

In Figs. 2.4.9 and 2.4.10 we have plotted the "experimental variogram" of the chain of values of the objective function for the Markov chain Monte Carlo methods that used global and local perturbations based on the a posteriori covariance matrix, respectively. It is clear from Figs. 2.4.9 and 2.4.10 that successive realizations generated using global perturbations are uncorrelated while realizations generated from local perturbations are

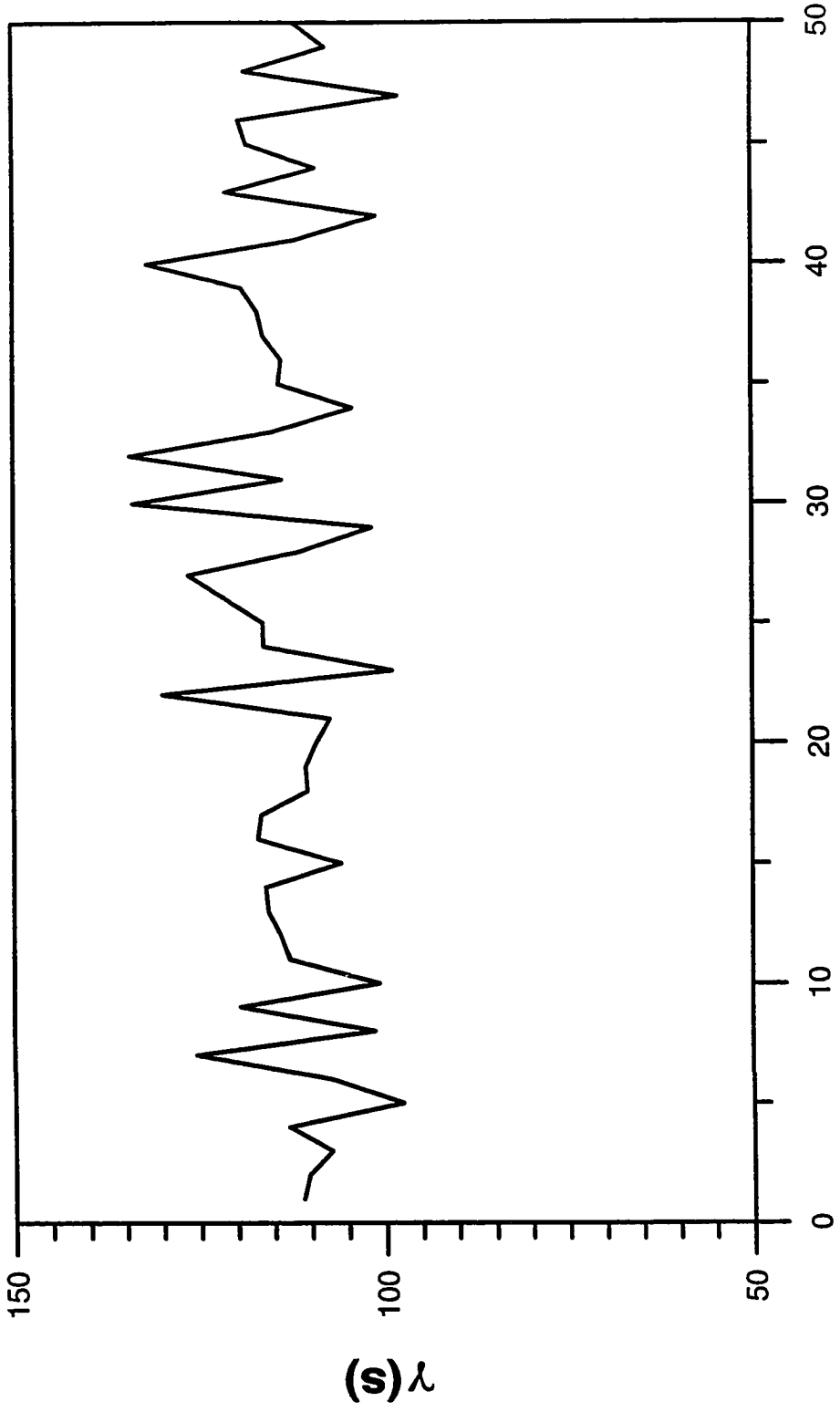

Fig. 2.4.9 - Experimental Variogram for the Objective Function of the Chain of Realizations Generated Using Global Perturbations Based on the a Posteriori Covariance Matrix (Variance = 0.25).

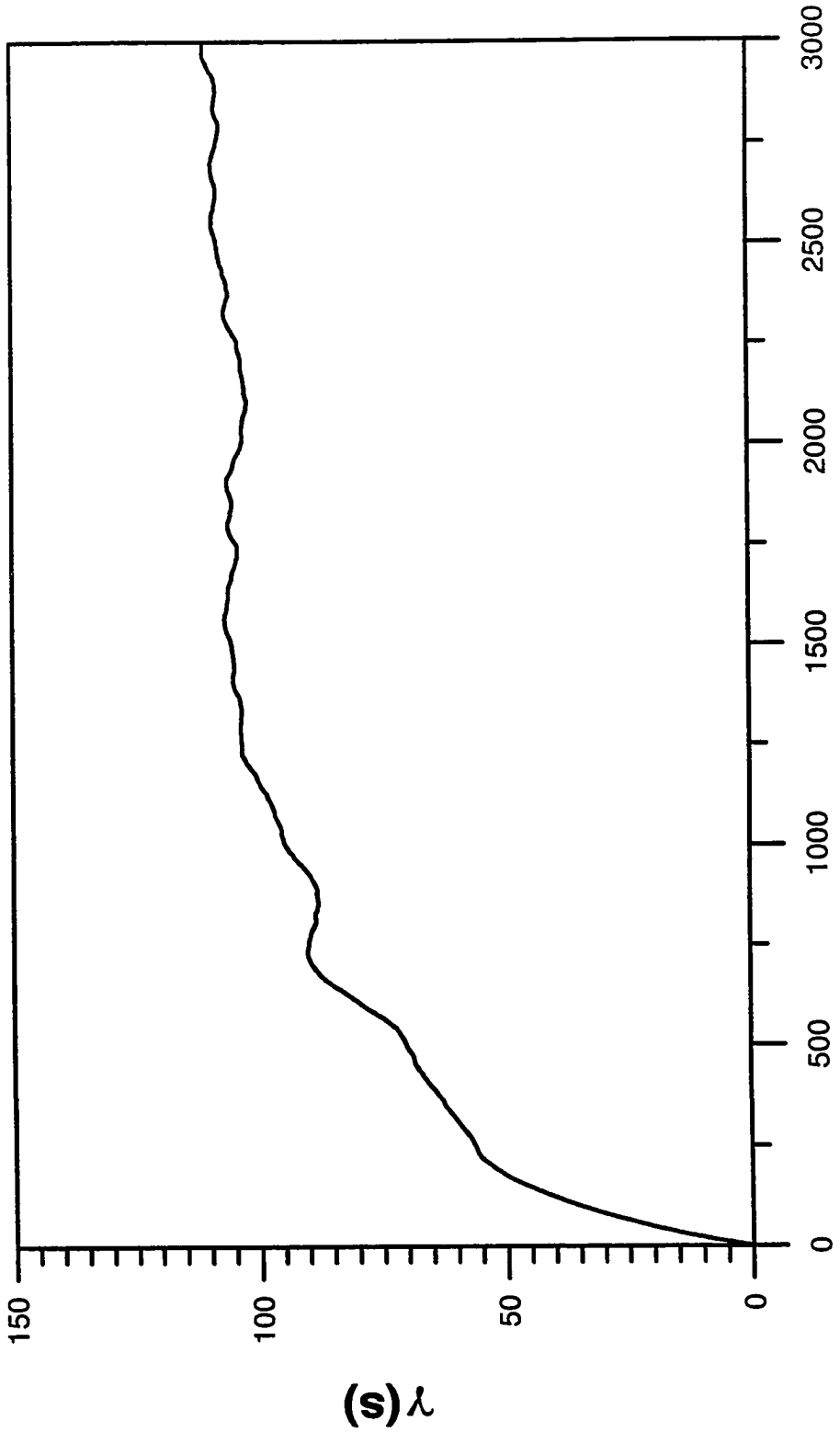

Fig. 2.4.10 - Experimental Variogram for the Objective Function of the Chain of Realizations Generated Using Local Perturbations Based on the a Posteriori Covariance Matrix (Variance = 0.25).

slightly correlated to lags of approximately 1200 for the third method. If, we retain only every 1200th realizations we would be left with 31 independent realizations from local perturbations compared to 200 from global perturbations for the method based on the a posteriori covariance matrix.

Regarding computational time, each of the three methods required approximately 13 hours to attempt 50000 perturbations.

Figures 2.4.11a and 2.4.11b show two realizations obtained with the Markov chain Monte Carlo method based on global perturbations generated from the a posteriori covariance matrix. Note that both realizations capture reasonably well most of the major trends in the true permeability field. For example, in both figures the low permeability region in the center of the truth case is captured. Figure 2.4.12 shows a comparison between the local variograms computed from the realizations of Figs. 2.4.11a and 2.4.11b and the model variogram. Note that local variograms generated from legitimate realizations do not exactly match the model variogram, which is expected, and it is going to be detailed in a later discussion.

2.3.2 Example 2 - Variance = 1.0 Case

In this example, we increase the prior variance of the log-permeability field to 1.0. All other information and data are the same as for Example 1.

Again the true log-permeability distribution was obtained from an unconditional simulation using the prior covariance matrix and is shown in Fig. 2.4.13. The values of $\ln(k)$ shown range from 0.1 to 7.0, i.e., in terms of permeability, the scale ranges from 1.1

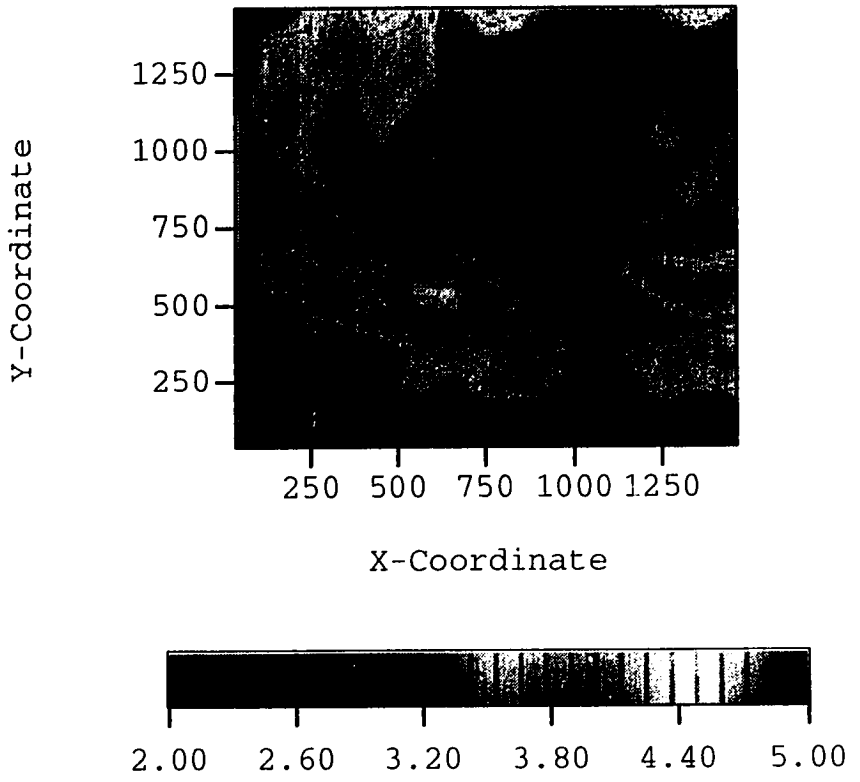

Fig. 2.4.11a - Realization 1 of Log-Permeability Field, Global Perturbations from a Posteriori Covariance Matrix (Variance = 0.25).

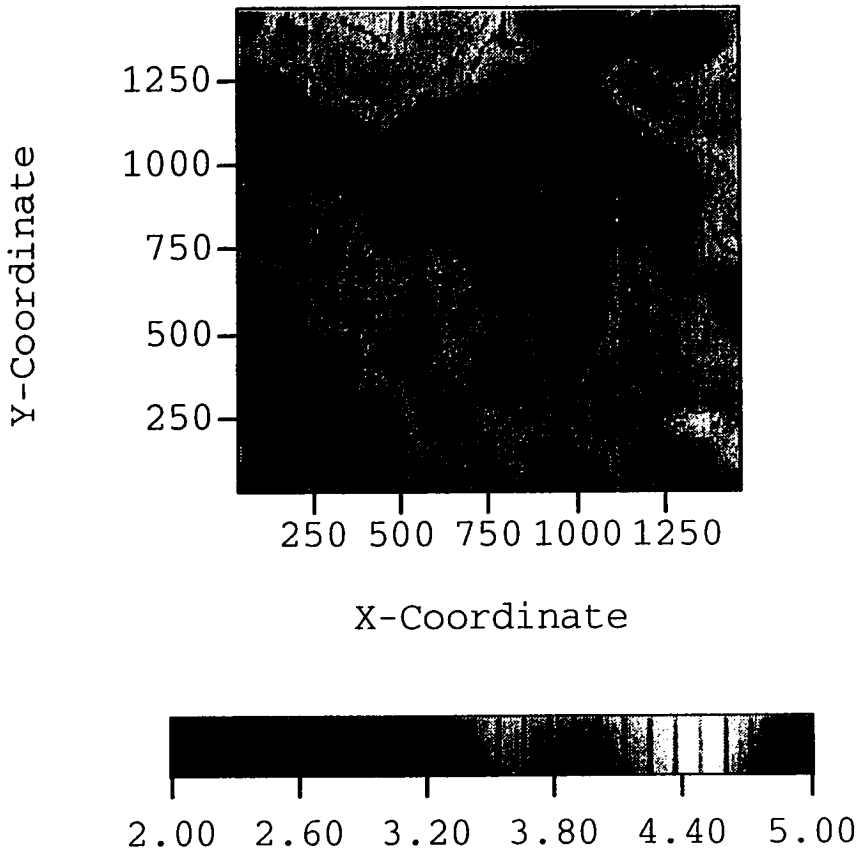

Fig. 2.4.11b - Realization 2 of Log-Permeability Field, Global Perturbations from a Posteriori Covariance Matrix (Variance = 0.25).

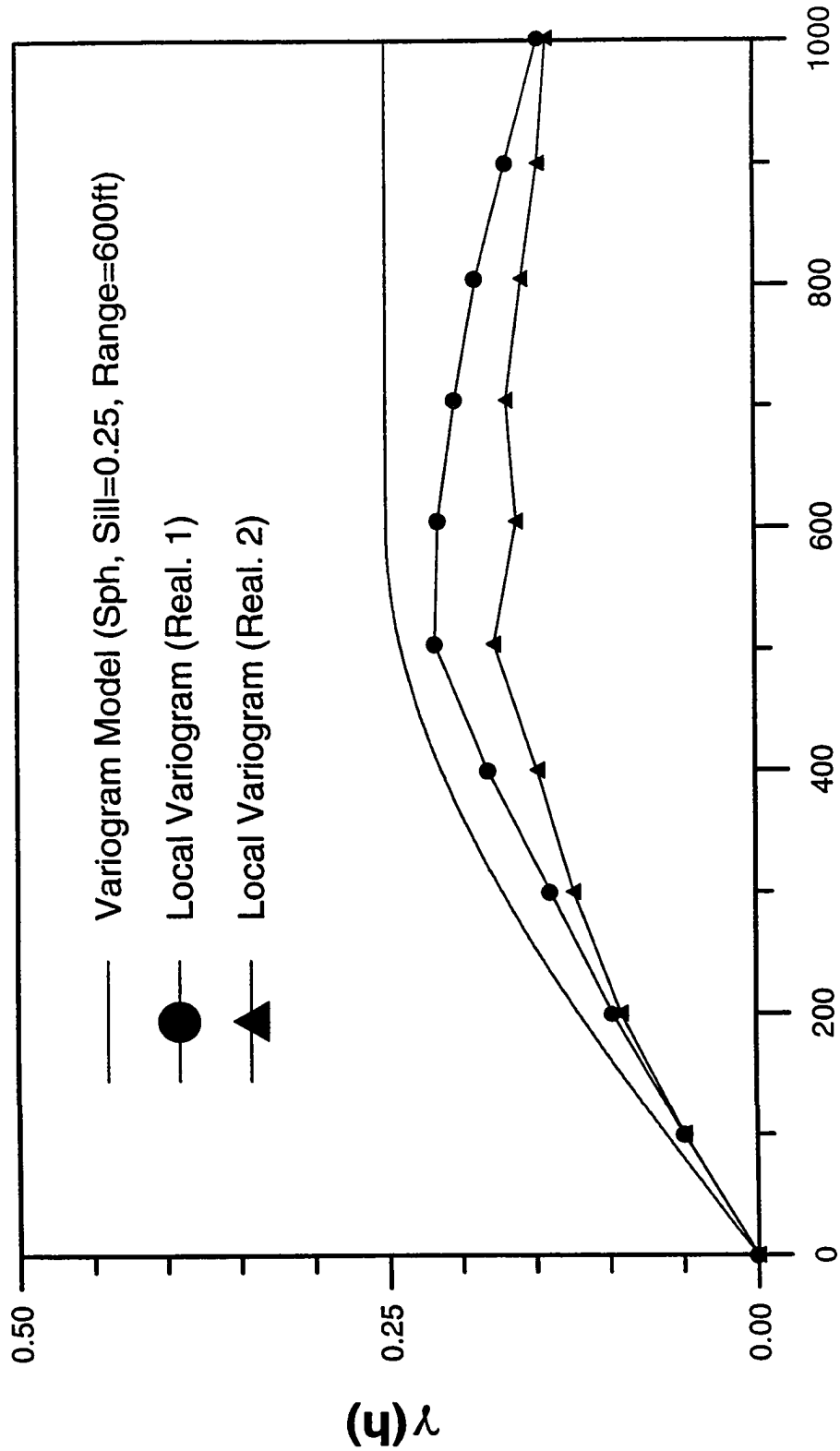

Fig. 2.4.12 - Variograms, Model Versus Calculated from Two Realizations Obtained Using Global Perturbations Based on the a Posteriori Covariance Matrix (Variance = 0.25).

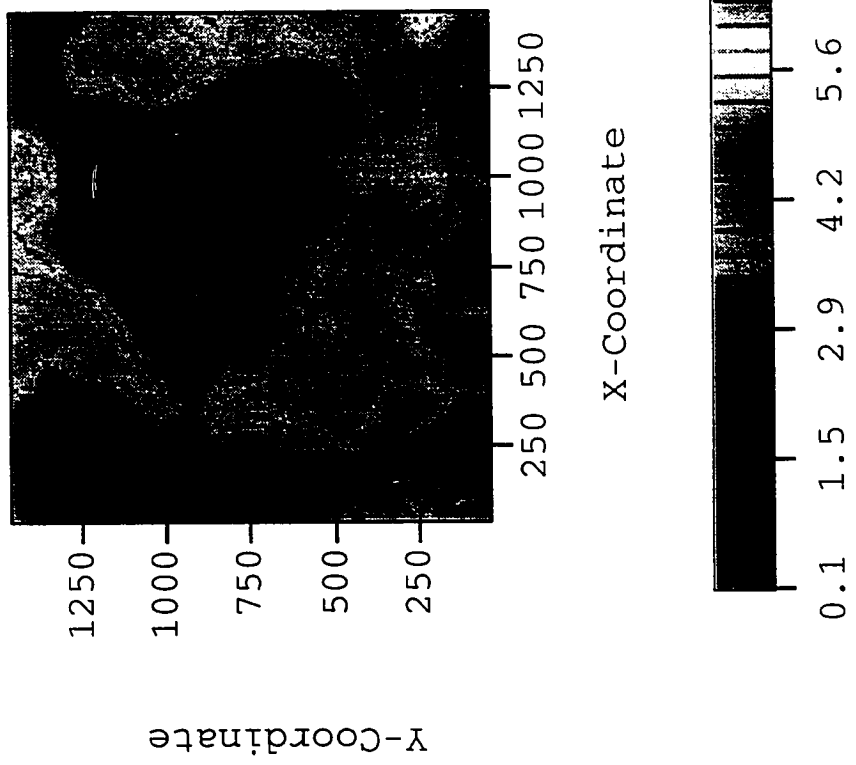

Fig. 2.4.13 - True Log-Permeability Distribution (Variance=1.0).

to 1,097 md. Fig. 2.4.14 shows a comparison between the local variogram obtained with the true log-permeability field and the model variogram.

Fig. 2.4.15 presents a plot of the value of "objective function" of accepted states versus the perturbation number for the same three Markov chain Monte Carlo methods that were compared in the previous example. A corresponding plot of the pressure misfit is shown in Fig. 2.4.16. As previously described, the sequence of states constructed using local perturbations based on the variogram do not reach the stationary distribution. Also, between accepted transitions the state remains unchanged, but those states have not been displayed. We see more clearly, in this example, that the chain of states generated from local perturbations of the variogram does not reach equilibrium in the first 50000 perturbations. Both methods based on using perturbations generated from the a posteriori matrix obtained from the gradient method have transient periods that are too short to be visible in Fig. 2.4.15.

Figures 2.4.17 and 2.4.18 shows log-permeability values at four gridblock positions (see locations depicted in Fig. 2.4.6) versus the perturbation number for the Markov chain Monte Carlo methods that used global and local perturbations based on the a posteriori covariance matrix. As before, the perturbation number in these figures refers to the number of transitions proposed, but only the values of accepted transitions are plotted. Regarding the areal location of the log-permeability gridblocks, we have a first location at one of the observation well gridblocks (X1), a second at the producing well gridblock (X2), a third location between the producing well gridblock and one observation well (X3) and a fourth almost at the reservoir boundary (X4). The a posteriori variance for the first, second, third and fourth gridblocks were 0.59, 2.4×10^{-6}, 0.24 and 0.93, respectively. The

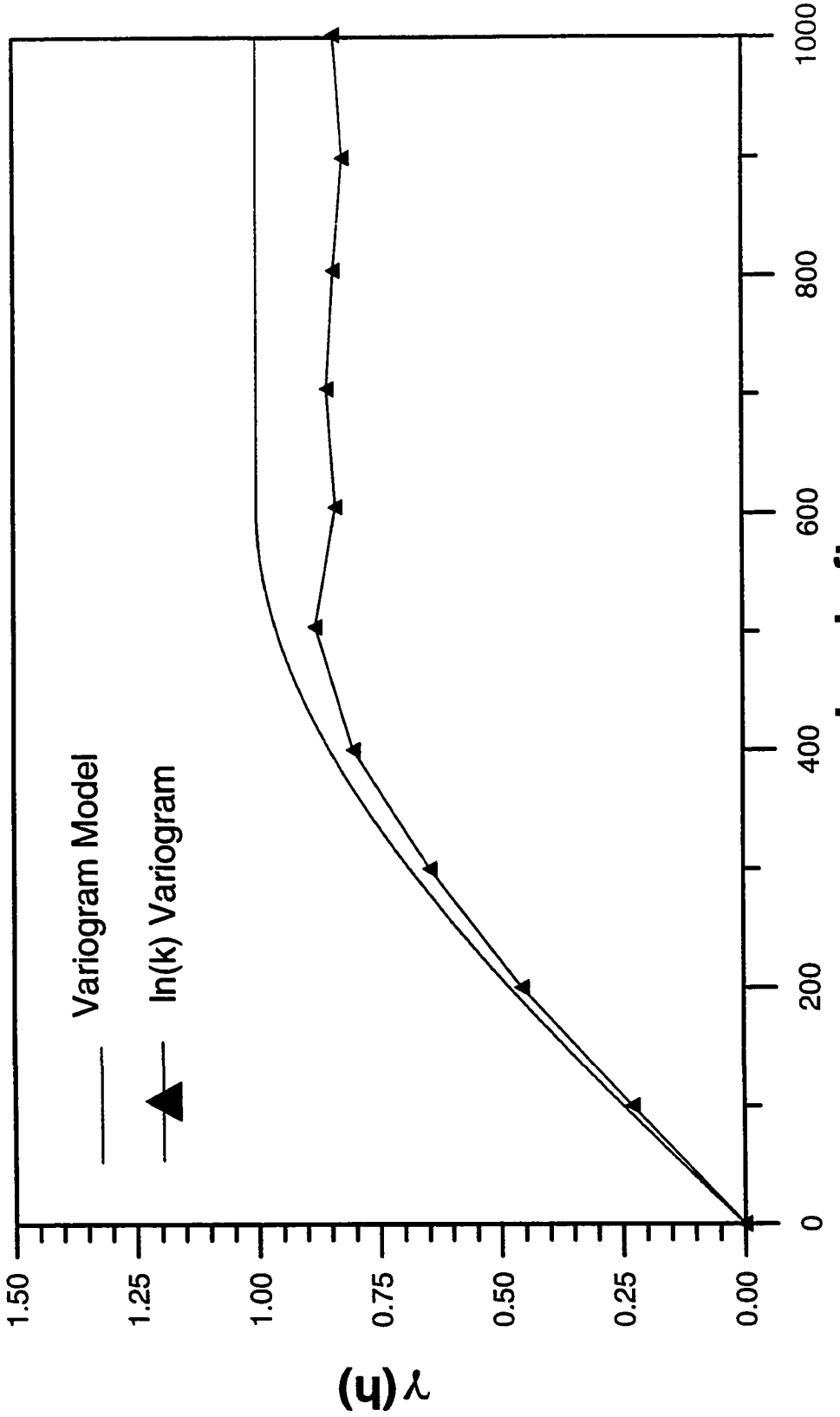

Fig. 2.4.14 - Local Variogram from Truth Case
(Sill=1.0 and Range=600ft).

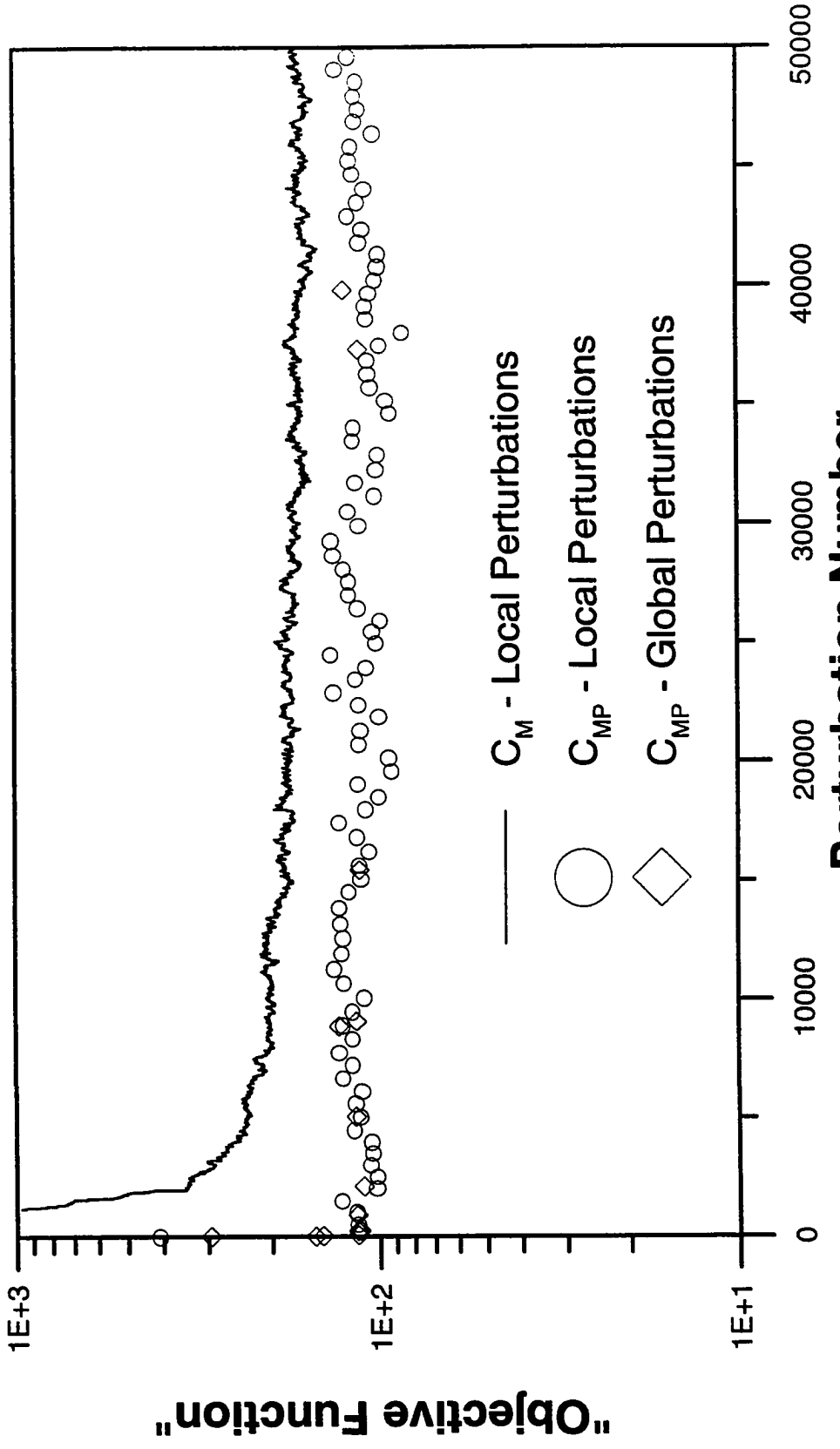

Fig. 2.4.15 - Values of the Objective Function for the Three Markov Chain Monte Carlo Methods (Variance = 1.0).

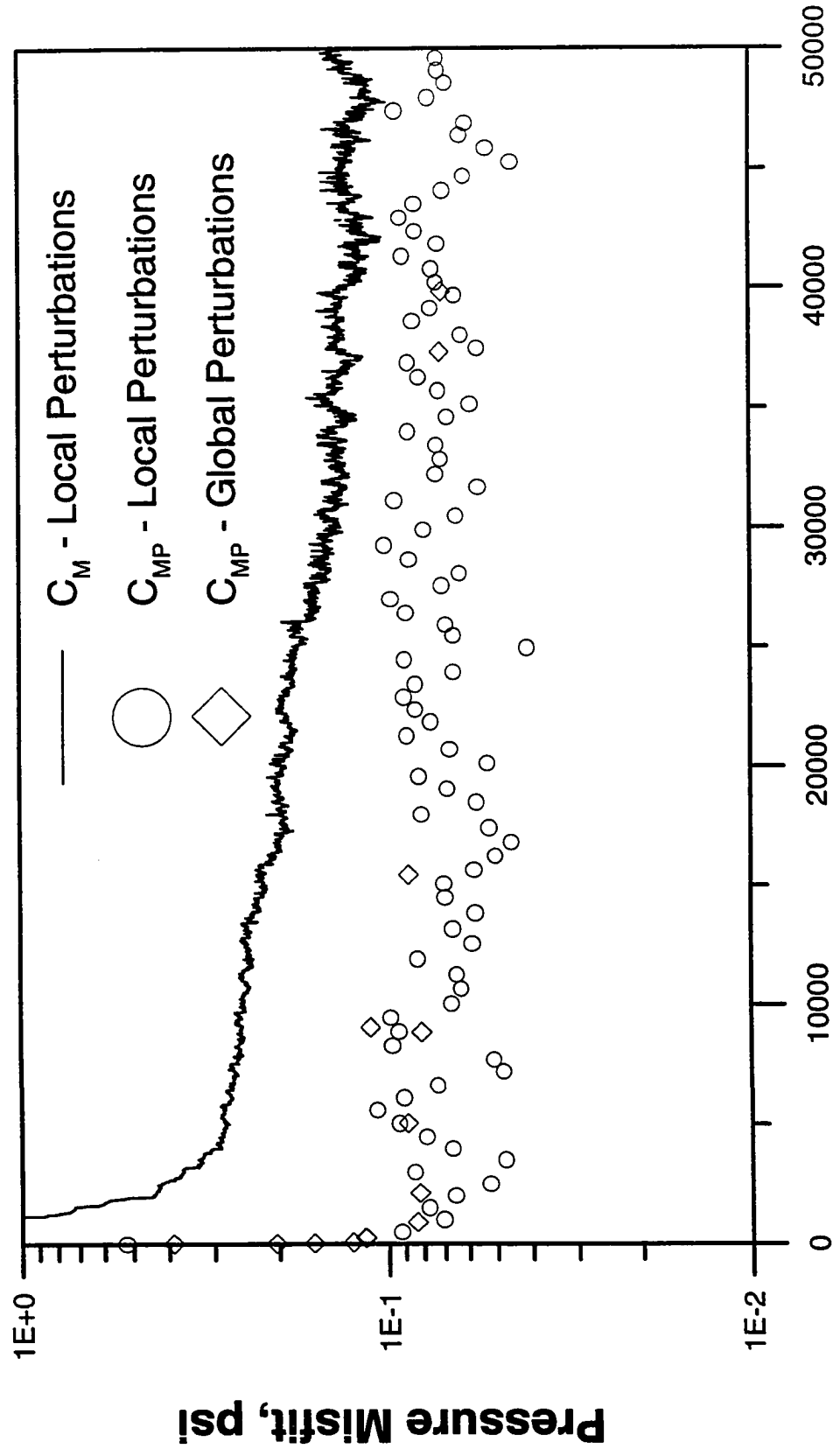

Fig. 2.4.16 - Values of the Pressure Misfit for the Three Markov Chain Monte Carlo Methods (Variance = 1.0).

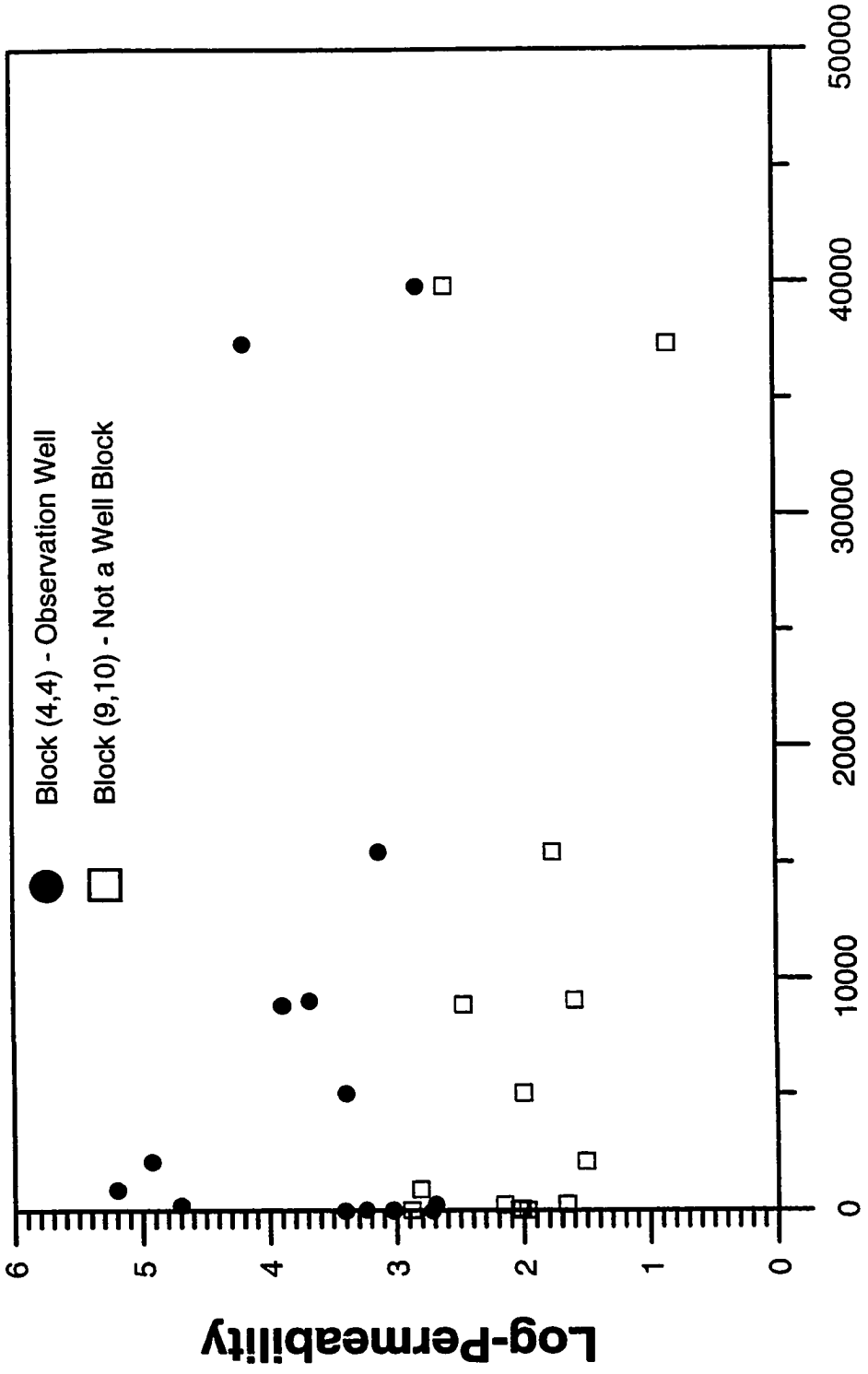

Fig. 2.4.17a - Log-Permeability Values During Sampling with Global Perturbations Based on the a Posteriori Covariance Matrix (Variance = 1.0).

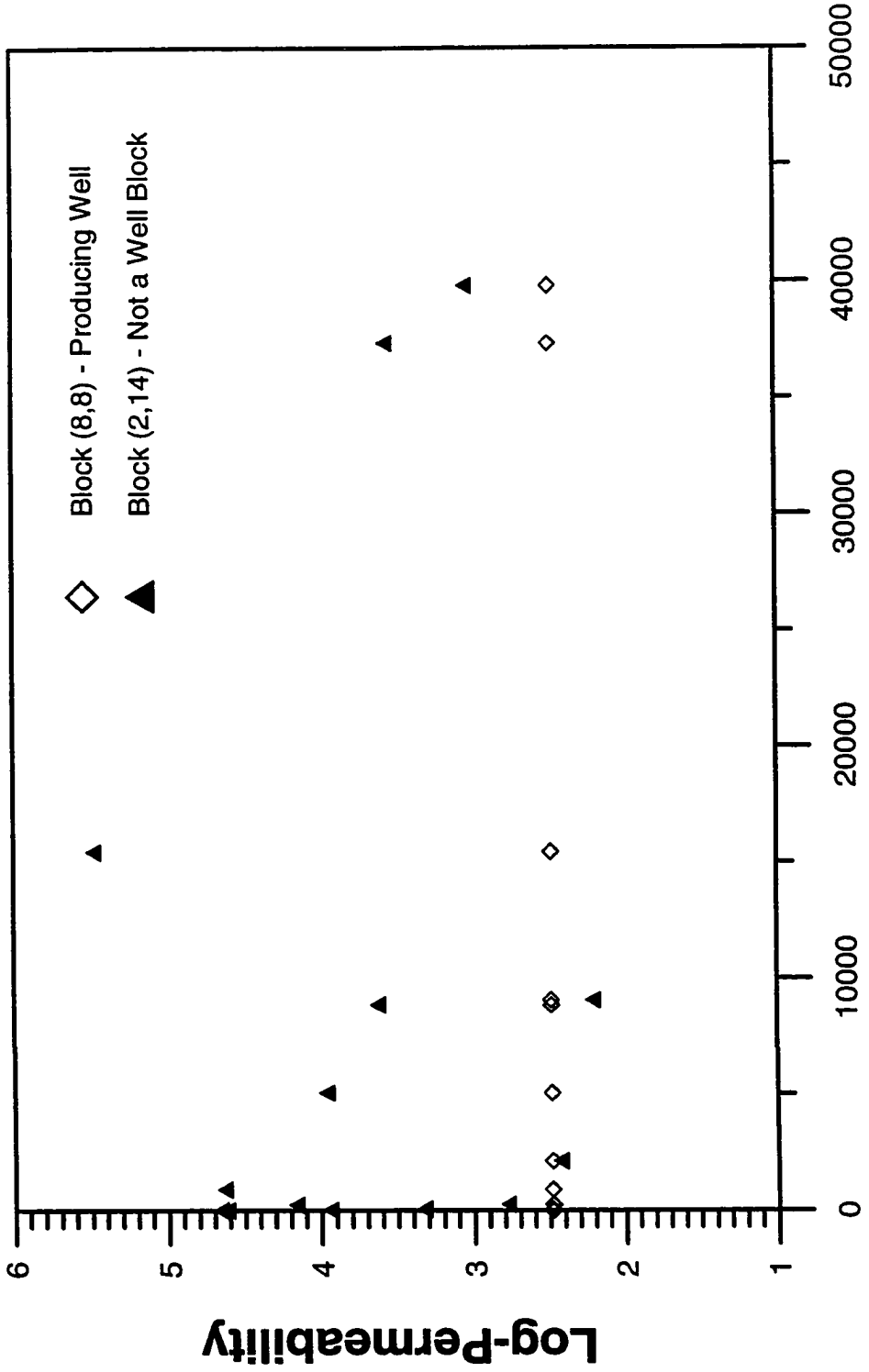

Fig. 2.4.17b - Log-Permeability Values During Sampling with Global Perturbations Based on the a Posteriori Covariance Matrix (Variance = 1.0).

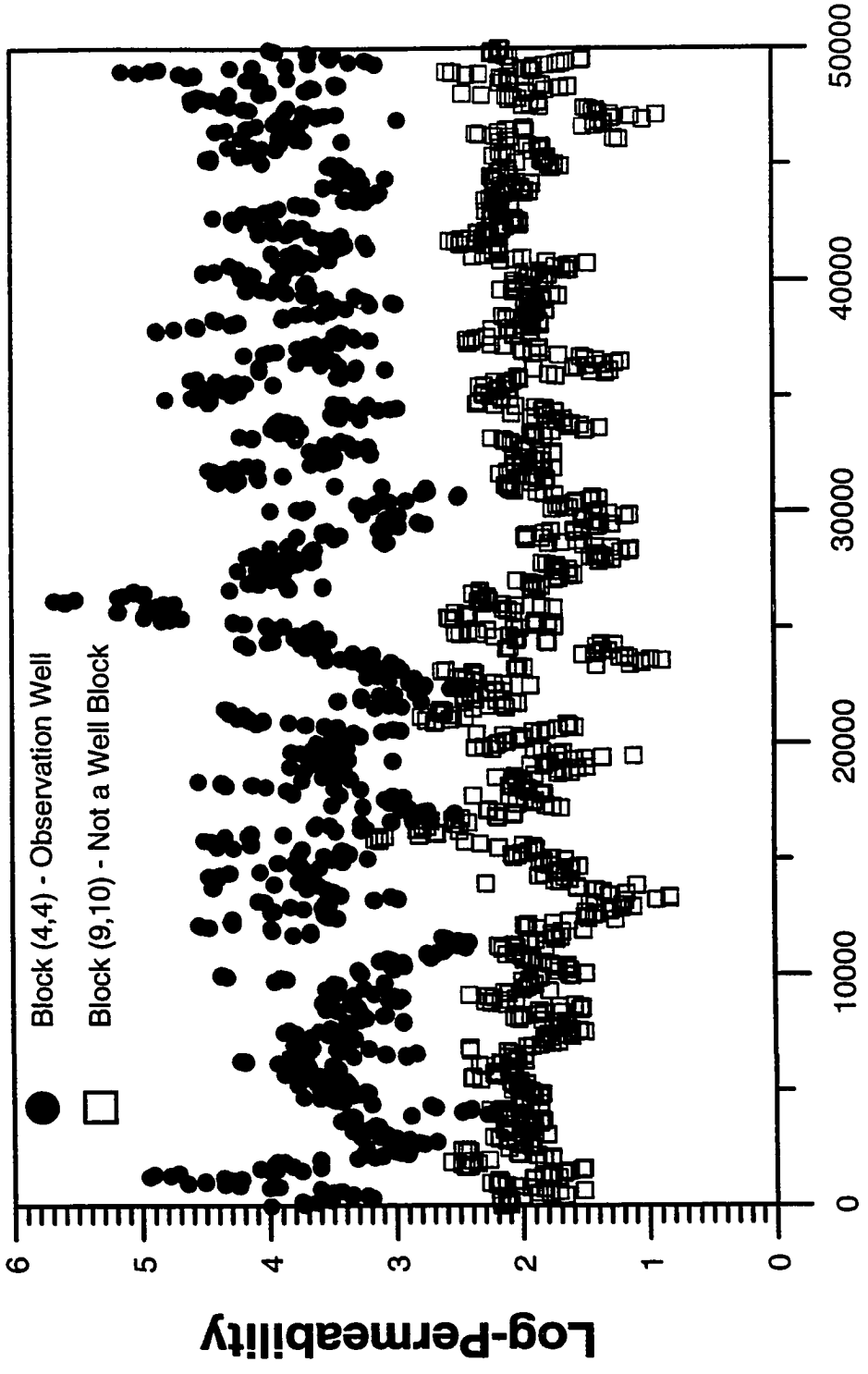

Fig. 2.4.18a - Log-Permeability Values During Sampling with Local Perturbations Based on the a Posteriori Covariance Matrix (Variance = 1.0).

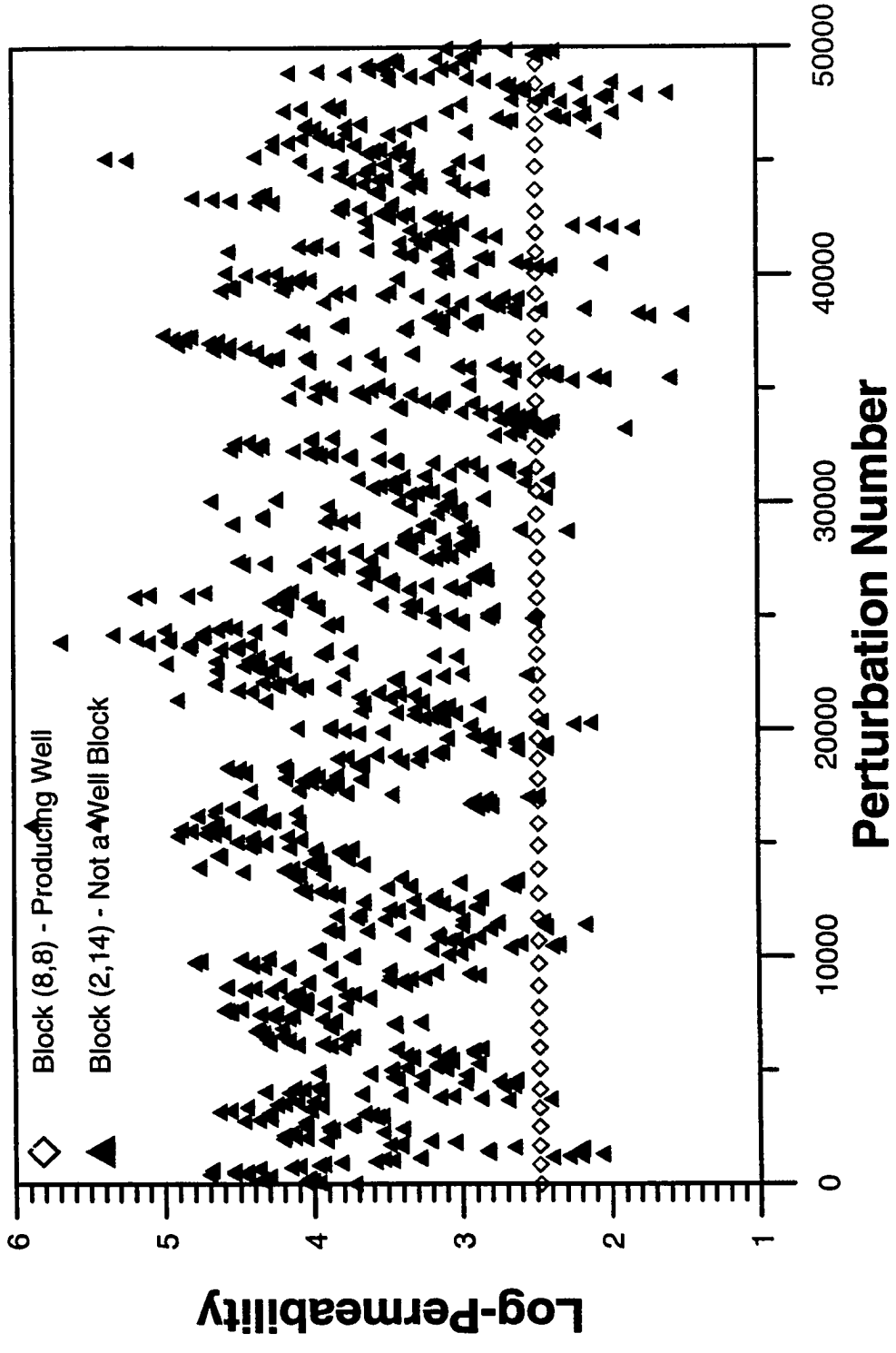

Fig. 2.4.18b - Log-Permeability Values During Sampling with Local Perturbations Based on the a Posteriori Covariance Matrix (Variance = 1.0).

maximum a posteriori estimates for the log-permeability values at the same gridblocks were 3.69, 2.48, 2.00 and 3.51, respectively.

If we consider the behavior of the permeability samples at each block we notice that the same features are common when using global and local perturbations. The sampled value for the producing well gridblock location stays almost constant and equal to the value encountered for the maximum a posteriori estimate for that gridblock through the whole simulation. This reflects the fact that at the producing well location the uncertainty is considerably reduced when incorporating pressure data together with the prior model information. This can be seen by the very low value encountered for the a posteriori variance.

At the gridblock between the producing and interference well the pressure resolves the uncertainty quite well (note the low value for the a posteriori variance). However, a larger variation in the log-permeability values for this gridblock during the sampling procedure when compared to the obtained to the producing well gridblock is expected to appear. This can be observed in Figs. 2.4.17 and 2.4.18.

On the other hand, at the analyzed observation well gridblock, since the pressure data is not sufficient to reduce the uncertainty in the permeability estimate (see the higher value for the a posteriori variance when compared with the producing well gridblock; almost the original prior variance of 1.0) as well as in the producing well gridblock, we can see that a large variation in the log-permeability estimates during the sampling process occurs. This is also the case for the gridblock located close to the corner of the reservoir.

The values of the log-permeability are highly correlated when using local perturbations to generate new states as can be seen by observing Fig. 2.4.18.

In relation to the number of images accepted, the method that used local perturbations based on the variogram did not generate any legitimate images of the permeability field during the sampling period. The Markov chain Monte Carlo method that used global perturbations based on the a posteriori covariance matrix generated 14 different, independent realizations but 6 had to be discarded because the chain had not reached equilibrium. Because of the greater non-linearity of the model, only 25300 different, but highly correlated images were obtained when the perturbations were local. As expected, using local perturbations increases the number of states accepted but successive states, or realizations, differ only slightly. As mentioned before, in the case of local perturbations it is necessary to thin the set of accepted images. To do this, we calculated the objective function experimental variogram for the chain of values of the objective function corresponding to the realization of the log-permeability field generated.

Fig. 2.4.19 shows the experimental variogram obtained. As we can conclude from Fig. 2.4.19, realizations generated from local perturbations are slightly correlated to lags of approximately 1800 for the third method. If, we retain only every 1800th realizations, we retain 14 independent realizations from local perturbations compared to 8 from global perturbations for the method based on the a posteriori covariance matrix.

Regarding computational time, each of the three methods required approximately 13 hours to attempt 50000 perturbations.

Figs. 2.4.20a and 2.4.20b show legitimate realizations of the log-permeability field obtained from the Markov chain when global perturbations based on the a posteriori matrix were used. Figs. 2.4.21a and 2.4.21b show two realizations obtained when using the Markov chain Monte Carlo method when local perturbations based on the a posteriori

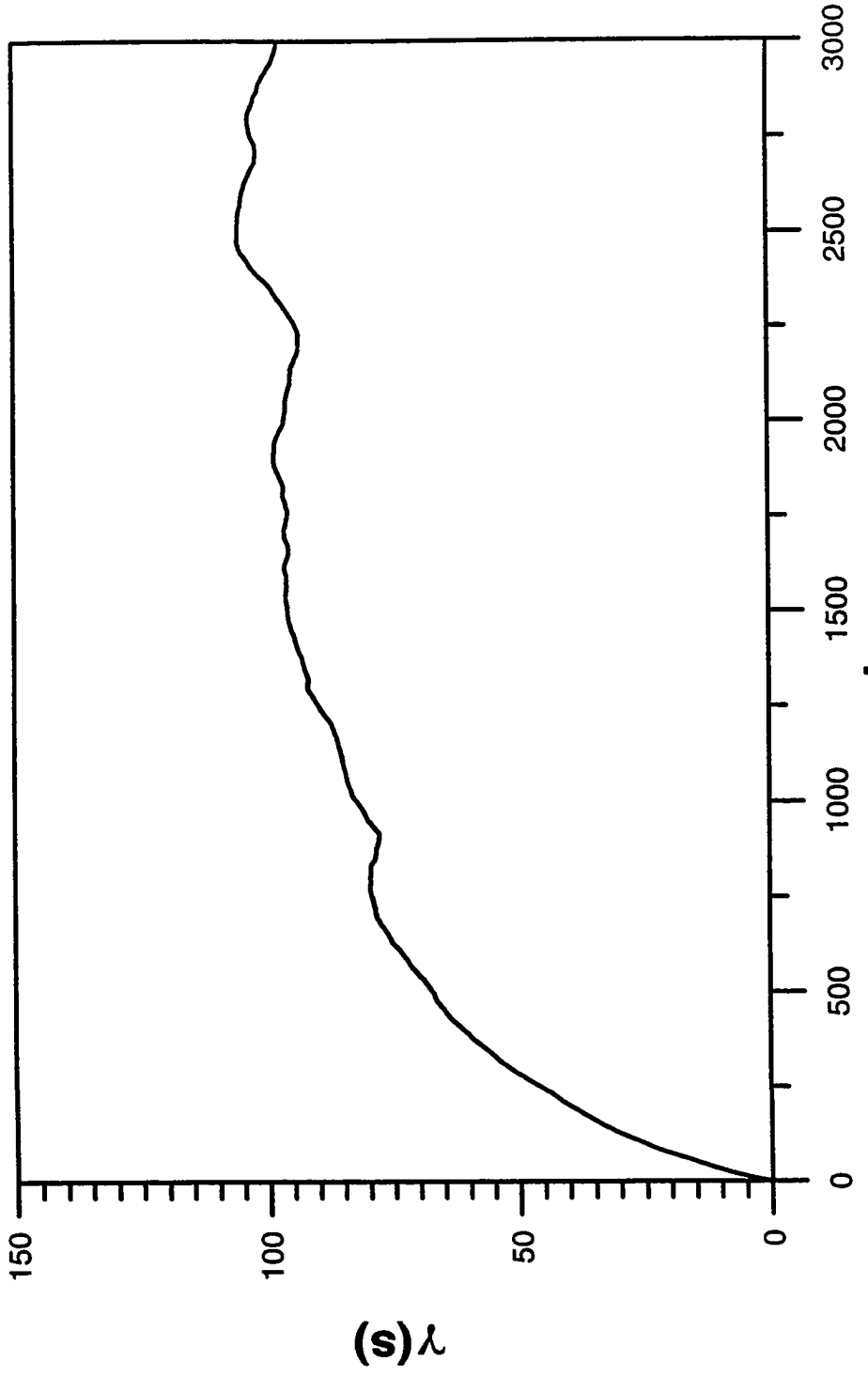

Fig. 2.4.19 - Experimental Variogram for the Objective Function of the Chain of Realizations Generated Using Local Perturbations Based on the a Posteriori Covariance Matrix (Variance = 1.0).

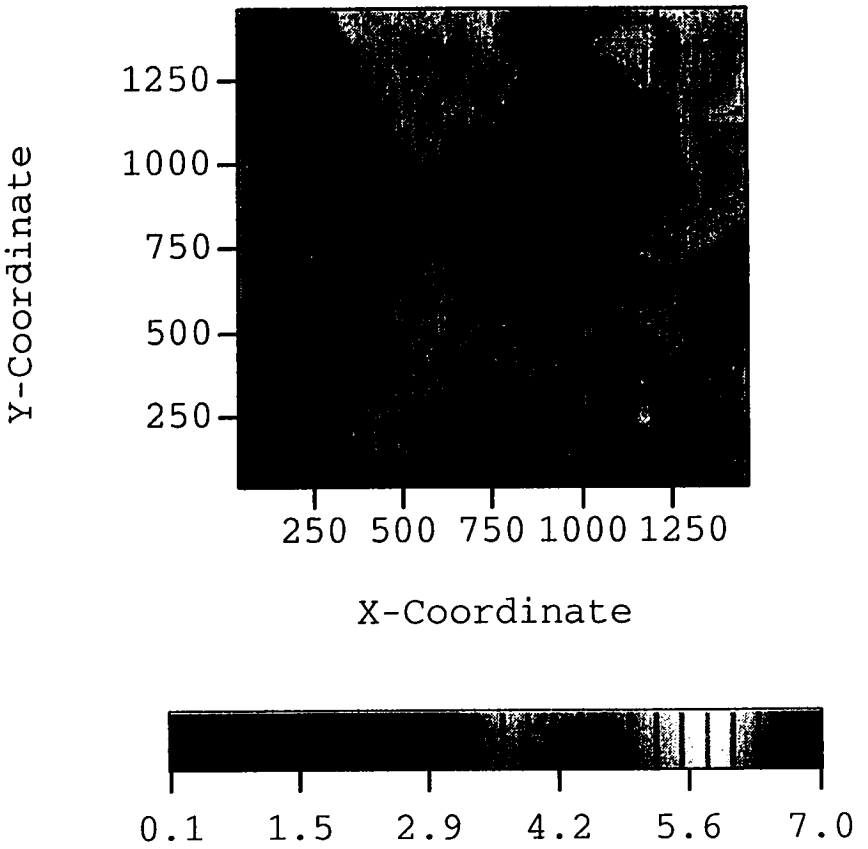

Fig. 2.4.20a - Realization 1 of Log-Permeability Field, Global Perturbations from a Posteriori Covariance Matrix (Variance = 1.0).

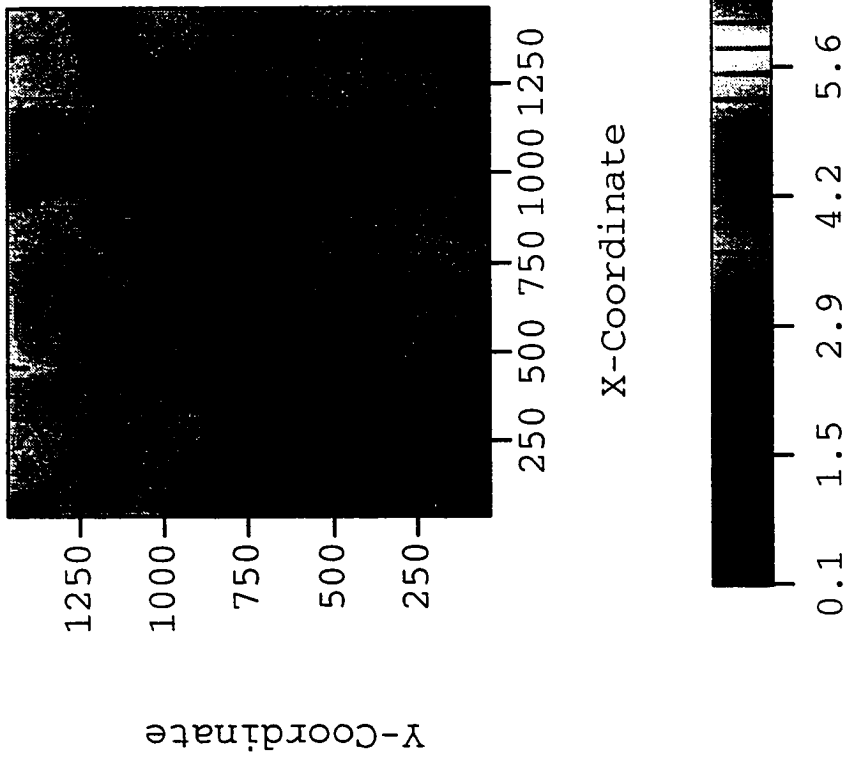

Fig. 2.4.20b - Realization 2 of Log-Permeability Field, Global Perturbations from a Posteriori Covariance Matrix (Variance = 1.0).

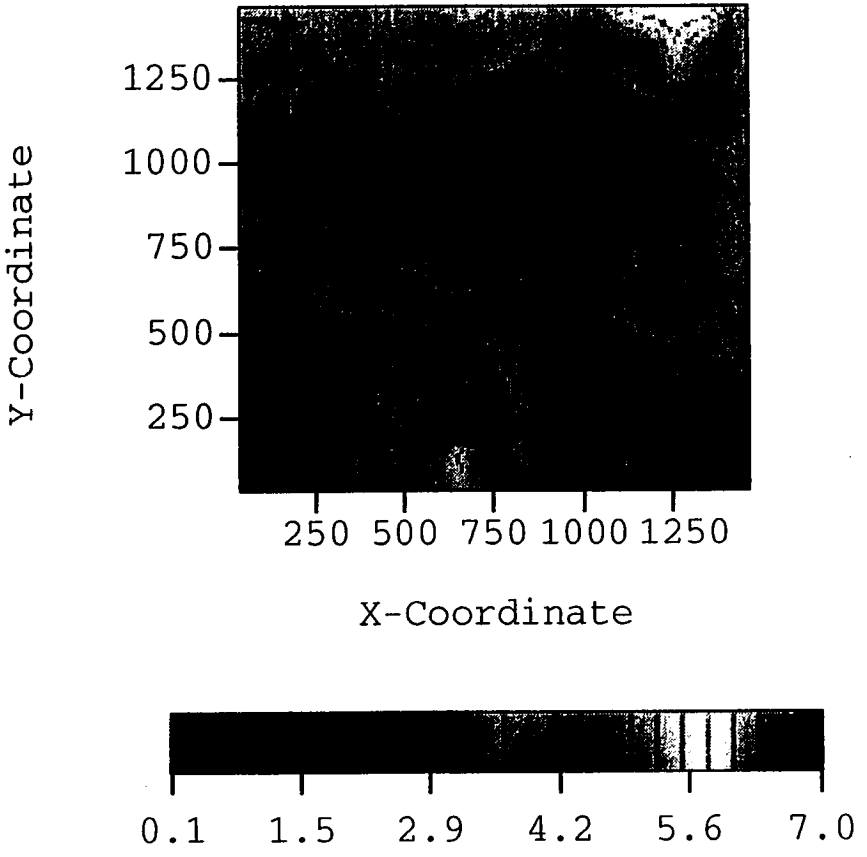

Fig. 2.4.21a - Realization 1 of Log-Permeability Field, Local Perturbations from a Posteriori Covariance Matrix (Variance = 1.0).

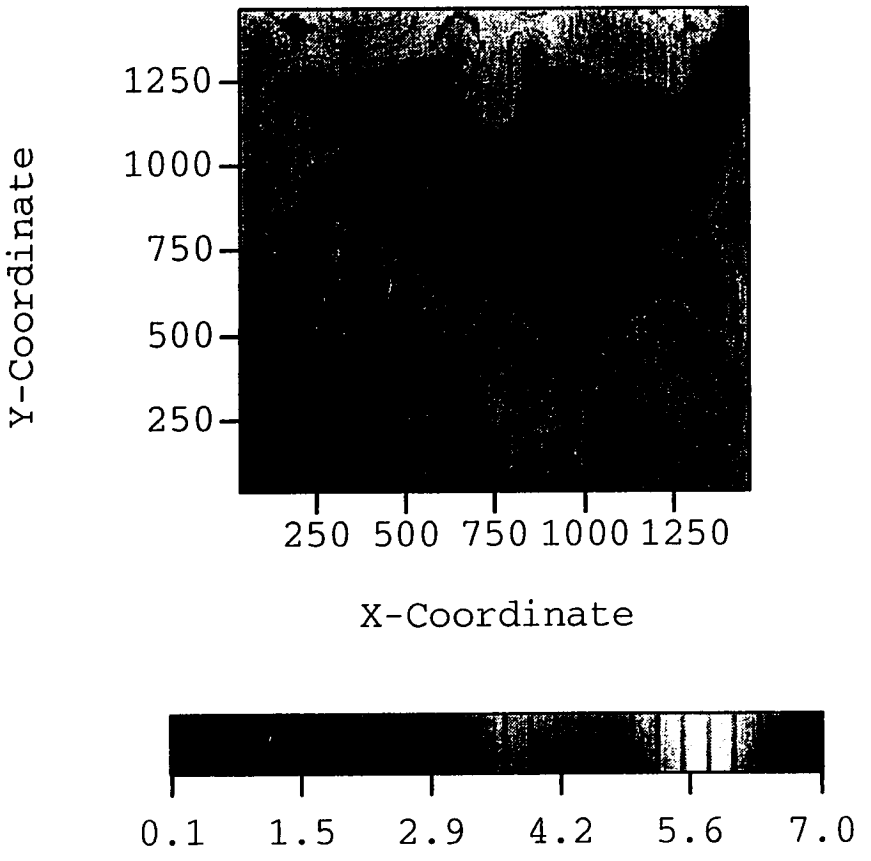

Fig. 2.4.21b - Realization 2 of Log-Permeability Field, Local Perturbations from a Posteriori Covariance Matrix (Variance = 1.0).

covariance matrix were used. Note that all four realizations capture reasonably well most of the major trends in the true permeability field. In all figures the low permeability region in the center of the truth case is captured. Fig. 2.4.22 compares the local variogram obtained from the realizations of Figs. 2.4.20a and 2.4.20b.

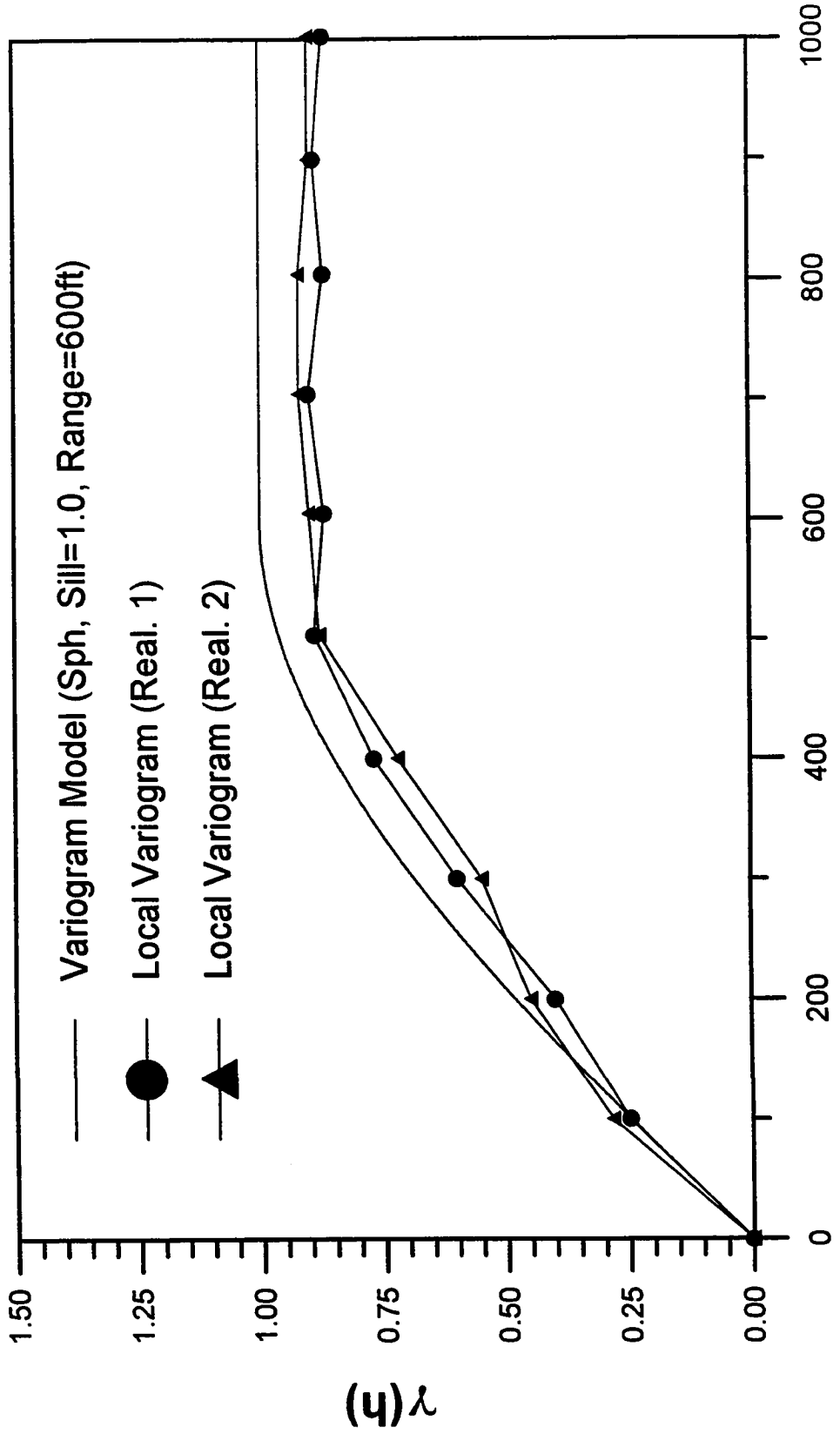

Fig. 2.4.22 - Variograms, Model and Calculated from Two Realizations Obtained Using Global Perturbations Based on the a Posteriori Covariance Matrix (Variance = 1.0).

CHAPTER III

RELATION BETWEEN SIMULATED ANNEALING AND A MARKOV CHAIN MONTE CARLO METHOD

In this major section, we first give a brief overview of simulated annealing as conventionally discussed. Then we discuss the classical results of Geman and Geman[18] and consider the notion of equiprobable realizations. Then we delineate the relationship between simulated annealing and Monte Carlo Markov chains. Specifically, we consider various perturbation mechanisms for simulated annealing and show that with a proper choice of the objective function, we can find a temperature level T such that the images generated with simulated annealing at temperature level T correspond to states in a Markov chain Monte Carlo that converge to realizations of the a posteriori probability density function for our model.

3.1 Theory of Simulated Annealing

Kirkpatrick *et al.*[13] and Cerny[14] independently introduced simulated annealing in an attempt to solve apparently intractable optimization problems. Simulated annealing is a special case of a wider class of global combinatorial optimization methods. We can say that simulated annealing is a Monte Carlo optimization procedure based on chemical

annealing. Annealing is the way in which crystals are grown - a melt is cooled very slowly until a crystal is formed. The rate of cooling is important, because a noncrystalline, metastable glass can form if cooling is too rapid. Kirkpatrick *et al.* and Cerny viewed the growth of a crystal as analogous to finding the global minimum in a optimization problem, and the development of a glass as the analog of incorrectly selecting a local minimum. Their primary application was directed at the combinatorial optimization problems that arise in the physical design of computers. At present, simulated annealing is being applied in several areas. A very comprehensive bibliography of simulated annealing use, classified by application area, has been produced by Collins *et al.*[34] (1987)

Simulated annealing is founded in concepts from statistical mechanics and the annealing process. Therefore, even today, simulated annealing is often presented in the original context which requires the definition of three components of a system which are analogous to the energy, the temperature and the interactions of a molecular system.

These components are:

(i) the objective function, which is analogous to the energy of a system and represents the function to be minimized (or maximized);

(ii) a control parameter, which is analogous to the temperature of the system and represents the temperature of a system which is an independent parameter and not necessarily related to any other parameter of the problem;

(iii) the interchange or perturbation mechanism, which corresponds to molecular interactions, and consists of a finite set of perturbations of the independent variables which produce a change in the objective function.

When used in an optimization context the goal of the simulated annealing method is to determine the configuration of the independent variables that yields a global minimum of the objective function. The procedure to reduce the temperature and specify the number of iterations or changes required at each value of the temperature is known as the annealing schedule.

The method proposed by Kirkpatrick *et al.* as well as Cerny is a variant of a Monte Carlo integration procedure of Metropolis *et al.* Metropolis *et al.* addressed the problem of random sampling from a Gibbs distribution at constant temperature, thereby simulating the average behavior of a physical system at thermal equilibrium. The Metropolis *et al.* algorithm proceeds in the following way. For each model parameter, a random perturbation is made, and the change in energy ΔE is computed. If $\Delta E \leq 0$ (i.e., if energy decreases), the perturbation is accepted. If ΔE is positive, then the perturbation is accepted with probability $P(\Delta E) = e^{-\Delta E/T}$. This conditional acceptance is easily implemented by choosing a random number α uniformly distributed between 0 and 1. If $\alpha \leq P(\Delta E)$, then the perturbation is accepted; otherwise the existing value for the parameter is retained.

Random perturbations according to these rules eventually causes the system to reach equilibrium, in which configurations of the model are realized with a Gibbs probability distribution. Because each step of the algorithm is dependent only on the present and not the past, the algorithm can be formally studied using Markov chain theory.

Kirkpatrick *et al.*'s and Cerny's optimization technique slowly lowers the temperature T during the execution of the Metropolis *et al.* algorithm. If the system is cooled sufficiently slowly and equilibrium is maintained, the model parameters eventually

converge to a state of minimum energy (maximum probability.) During this process an inhomogeneous chain is formed. There has been a substantial amount of research into the statistical behavior of the simulated annealing algorithm. The books by Aarts and Korst[15] and van Laarhoven and Aarts[16] and the technical report of Neal[17] discuss many of the theoretical convergence results and give a lengthy set of references. In general, theoretical results exploit the relation between simulated annealing and Markov chains to obtain convergence proofs. The essential characteristic of this optimization procedure is its ability to escape from local minima.

Most published attempts to condition permeability fields to production data rely on either simulated annealing or the genetic algorithm to reduce the misfit error between production data calculated from a theoretical model of the reservoir and the actual production data. The most commonly used measure of misfit is the sum of the squared differences. To fit this into the optimization framework an "objective function" is formed from the weighted sum of the misfit terms. Typically, when the objective is to condition to the variogram and pressure data, an objective function similar to the following one is used:

$$O_{SA} = w_1 O_1 + w_2 O_2 , \tag{3. 1}$$

where

$$O_1 = \sum_{l=1}^{n_l} (\gamma_S(h_l, m) - \gamma_{model}(h_l))^2 , \tag{3. 2}$$

and

$$O_2 = \sum_{n=1}^{N_d} (d_n - d_{obs,n})^2 = (g(m) - d_{obs})^T (g(m) - d_{obs}) . \tag{3. 3}$$

Here, γ_{model} denotes the model variogram, and $\gamma_S(h_l,m)$ denotes the variogram calculated from a given estimate of the model m (log-permeability field) at a lag distance h_l. The sum in Eq. 3.2 is over n_l lag distances. Also in our problem, N_d denotes the total number of pressure data, $d_{obs,j}$ is the jth pressure measurement and d_j is the corresponding component of calculated data for a given model m, see Eq. 1.2. The last equality of Eq. 3.3 follows from Eq. 1.2, where throughout the superscript T on a matrix (or vector) denotes the transpose of the matrix. Usually, the relative weights of the objective function, w_1 and w_2 are obtained by running an initialization stage of I independent perturbations and then setting

$$w_1 = \frac{1}{I}\sum_{i=1}^{I}(O_1 - O_1^i), \tag{3.4}$$

and

$$w_2 = \frac{1}{I}\sum_{i=1}^{I}(O_2 - O_2^i). \tag{3.5}$$

See, for example, Deutsch and Cockerham[35] (1994).

In Eqs. 3.4 and 3.5, O_j, j=1,2, represent fixed values of the objective functions obtained by evaluating these objective functions at an initial configuration of the model parameters and O^i_j, j=1,2, represent the values of the objective functions evaluated at m^i, where m^i is the image of the model obtained by the ith perturbation. In determining the relative weights, w_1 and w_2, the model is not updated from the initial configuration.

Typically, the annealing algorithm then uses the following steps.

(i) Generate an initial, uncorrelated distribution of the simulation variable (permeability in our case) by visiting each point in the simulation grid. This is usually done by sampling of the cumulative distribution function or histogram.

(ii) The initial distribution is then rearranged until the objective function is minimized.

A typical procedure is one in which random pairs of gridblock values are swapped (interchange mechanism) and the objective function is evaluated. The Metropolis algorithm is then applied; i.e., if the value of the objective function is reduced, the swap is accepted and the distribution is updated. If it is not reduced, the swap is accepted with probability

$$p(\Delta O^k, T_l) = \exp(-\frac{\Delta O^k}{T_l}), \tag{3.6}$$

where p is the probability function, ΔO^k is the change in the objective function resulting from the proposed swap and T_l is the value of the control parameter at the lth temperature level.

(iii) Step ii is applied at decreasing set of values of T_l. Accepted, although heuristic, procedures for determining the initial value of the control parameter (temperature) and reducing the temperature are given in Appendix A. The number of iterations that should be done at each temperature level is also heuristic, and one criterion that is used here is also given in Appendix A. The algorithm terminates when the change in the objective function is less than a specified tolerance or the acceptance ratio is less than a prescribed value. The acceptance ratio is the number of accepted swaps (or, more generally accepted perturbations) over the number of proposed swaps (proposed perturbations) at the lth step, i.e., the step corresponding to T_l.

For our purposes, it is better to write Eq. 3.6 as

$$p(m_j, T_l) = \exp\left(-\frac{(O^j - O^i)}{T_l}\right). \tag{3.7}$$

Here, $O^i=O(m^i)$ is the objective function evaluated at the current image or configuration of the model m^i ,and $O^j=O(m^j)$ is the objective function evaluated at the proposed new image of the model, m^j, proposed by the perturbation mechanism. If the $O^j<O^i$, then m^j is accepted with probability 1 and if $O^j > O^i$, then m^j is accepted with probability $p(m_j,T_l)$. When simulated annealing is viewed as a Markov chain at each temperature level, it will be apparent that at temperature T_l, we are sampling from a probability distribution for the model of the form

$$\pi(m, T_l) = a(T_l)\exp\left(-\frac{O(m)}{T_l}\right), \tag{3.8}$$

where $a(T_l)$ is a normalizing constant to ensure that probabilities sum to unity. The normalizing constant actually plays no role in the acceptance criterion since Eq. 3.7 is equivalent to

$$p(m_j, T_l) = \frac{\pi(m_j)}{\pi(m_i)} = \exp\left(-\frac{(O^j - O^i)}{T_l}\right). \tag{3.9}$$

Thus, the acceptance criterion says that the proposed perturbation m^j will always be accepted if the probability of m^j is greater than the probability of m^i and otherwise will be accepted with a probability equal to the ratio of their probabilities. Often in this work, we will use the notation π_i to denote $\pi(m^i)$.

Although our implementation of simulated annealing was not directly motivated by the work of Geman and Geman[18], a brief discussion of their work is important for historical purposes as they give detailed proofs of important theoretical results.

In addition to the probability distribution of Eq. 3.8, we also focus on the associated probability distribution obtained at T_l=1, i.e.,

$$\pi(m) = a(1)\exp(-O(m)), \tag{3. 10}$$

where $a(1)$ is still a normalizing constant.

Geman and Geman assume that if n_t, t=1,2,...,∞ denotes the order in which perturbations of components of the model are considered, then there is a positive integer jn such that for every t, perturbation of all components of the model are considered at some point in the finite "sequence" $\{n_{t+1},n_{t+2},...n_{t+jn}\}$. They also use the Gibbs sampler (which will not be considered here) to generate values of each component of the model. Under the preceding conditions, Geman and Geman show that when the Gibbs sampler is used and temperature decreases sufficiently slowly to zero, then for any starting configuration m^0,

$$\lim_{t\to\infty} \text{Prob}(X(t)=m \mid X(0)=m^0) = \pi_0(m). \tag{3. 11}$$

Here, $X(t)$ is a random variable which represents the state of the model (the distribution of log-permeabilities for the problem of interest to us) after t replacement opportunities, and π_0 is the uniform probability distribution on the set of most probable models for the probability distribution given by Eq. 3.10, i.e., the set of models which minimize the objective function. Prob($X(t)=m \mid X(0)=m^0$) represents the probability that $X(t)=m$ given that $X(0)$ equals m^0. The preceding theoretical result provides the fundamental basis for

the concept of equally probable realizations. In the limit as $T \to 0$ and $t \to \infty$, the permeability distributions generated by simulated annealing will converge to a realization that is contained in the set of most probable models based on the probability distribution of Eq. 3.10, or more precisely, we will obtain a sample from the uniform distribution on the set of most probable models (models which give global minima of the objective function $O(m)$). Since each of this most probable models are equally likely, if we are able to generate several of them then we have generated a set of equally probable realizations. If the objective function has only one global minimum, there is of course only one most probable model.

This result means that in the limit, a proper application of simulated annealing will generate a maximum a posteriori estimate. It is shown however, that the standard procedure of incorporating the variogram into the objective function for simulated annealing cannot be expected to give equally probable realizations.

Geman and Geman also show that if T_l is fixed equal to one and all components of the model are visited for updating infinitely often, then

$$\lim_{t \to \infty} \text{Prob}(X(t)=m \mid X(0)=m^0) = \pi(m), \qquad (3.12)$$

where $\pi(m)$ is the probability distribution of Eq. 3.10. Note this means that if we continue to sample at $T_l=1$, in the limit we will generate samples from the probability distribution $\pi(m)$. These samples will not be equally probable.

As a final remark, the results of Geman and Geman also assume that the state space is finite, i.e., that the number of possible models is finite. This assumption does not prohibit using the methodology for an infinite state space. In fact, since the procedure is

always applied on a computer, the number of permeability values that can be generated is extremely large, but is finite.

3.2 General Relationship Between Markov Chain Monte Carlo and Simulated Annealing

In Appendix A, we discuss procedures for choosing the initial temperature in simulated annealing, the schedule for reducing the temperature and the procedure for determining the number of perturbations done at each temperature level. Recall that in simulated annealing, a proposed change from the log-permeability field m^i to a proposed new configuration m^j will be accepted with probability 1 if

$$O^j = O(m^j) \leq O^i = O(m^i), \tag{3. 13}$$

and otherwise will be accepted with probability

$$p(m^j, T_l) = \exp\left(-\frac{1}{T_l}(O^j - O^i)\right), \tag{3. 14}$$

where $O(m)$ denotes the objective function evaluated at m. The preceding criteria defines the probability of accepting the transition from proposed m^i to m^j. The probability of accepting the proposed change from image m^i to m^j, is denoted by

$$p(i, j; T_l) = p(m^j, T_l). \tag{3. 15}$$

Defining $r_{ij}(T_l)$ by

$$r_{ij}(T_l) = \frac{\exp\left(-\frac{O^j}{T_l}\right)}{\exp\left(-\frac{O^i}{T_l}\right)} = \exp\left(-\frac{1}{T_l}(O^j - O^i)\right), \tag{3. 16}$$

the criteria for accepting the proposed change from m^i to m^j, we see that the probability of acceptance in simulated annealing is given by

$$p(i, j, T_l) = \min\{1, r_{ij}(T_l)\}. \tag{3. 17}$$

Recall that the probability of accepting a transition from state i to state j in our Markov chain Monte Carlo procedure is given by

$$\alpha_{ij} = \min\{1, \frac{\pi_j q_{ji}}{\pi_i q_{ij}}\}, \tag{3. 18}$$

where q_{ij} is the probability of proposing a transition and π_i and π_j, represent the a posteriori probability density function evaluated at m^i and m^j, respectively. From Eqs. 3.16 and 3.17, it follows that the images generated with simulated annealing at T_l will represent states of a Markov chain if

$$r_{ij}(T_l) = \frac{\pi_j q_{ji}}{\pi_i q_{ij}}. \tag{3. 19}$$

We will find that Eq. 3.19 holds at a particular temperature level if the objective function and perturbation mechanism are defined properly.

3.3 Using Simulated Annealing to Determine the Initial State in a Markov Chain

3.3.1 Perturbations Based on the Variogram

In this case, we begin simulated annealing from an initial configuration generated by Eq. 2.17. At all temperature levels, we propose new log-permeability fields with the same equation; i.e., perturbations are generated by unconditional simulation based on the variogram and prior mean. In generating a Markov chain using this same perturbations mechanism to propose transitions from state i to state j, we found that

$$\frac{\pi_j q_{ji}}{\pi_i q_{ij}} = \exp\left(-\frac{1}{2}(\Delta d^{jT} C_D^{-1} \Delta d^j - \Delta d^{iT} C_D^{-1} \Delta d^i)\right), \tag{3.20}$$

where

$$\Delta d^i = g(m^i) - d_{obs}, \tag{3.21}$$

is the pressure difference for state i. Since C_D is a diagonal matrix with all diagonal entries equal to σ^2_d Eq. 3.20 can be rewritten as

$$\frac{\pi_j q_{ji}}{\pi_i q_{ij}} = \exp\left(-\frac{1}{2\sigma^2_d}(\Delta d^{jT} \Delta d^j - \Delta d^{iT} \Delta d^i)\right). \tag{3.22}$$

We now define an objective function by

$$O = \frac{1}{2\sigma^2_d}(g(m) - d_{obs})^T (g(m) - d_{obs}) = \frac{1}{2\sigma^2_d}\sum_{n=1}^{N_d}(g_n(m) - d_{obs,n})^2, \tag{3.23}$$

where $g_n(m)$ denotes the nth component of the calculated pressure data vector $g(m)$, and $d_{obs,n}$ denotes the nth component of the observed pressure data vector d_{obs}. It follows that

$$O(m^j) - O(m^i) = \frac{1}{2\sigma^2_d}(\Delta d^{jT}\Delta d^j - \Delta d^{iT}\Delta d^i), \tag{3.24}$$

and

$$r_{ij}(T_l) = \exp\left(-\frac{1}{2T_l\sigma^2_d}(\Delta d^{jT}\Delta d^j - \Delta d^{iT}\Delta d^i)\right). \tag{3.25}$$

Comparing Eq. 3.25 and 3.22, it follows that Eq. 3.19 holds when T_l=1. Thus, if we decrease the temperature through an annealing schedule until we reach T_l=1 and do no further temperature decreases, the images generated at T_l=1 will represent states of a Markov chain which has Eq. 1.5 as its stationary distribution. This result holds regardless of whether we do local or global perturbations based on the variogram (prior covariance matrix).

The preceding arguments also show that at temperatures T_l>1, we are actually constructing a Markov chain which has

$$\hat{\pi}(m,T_l) = \hat{a}_l \exp\left[-\frac{1}{2}(m-m_0)^T C_M^{-1}(m-m_0)\right.$$

$$\left.-\frac{1}{2T_l}(g(m)-d_{obs})^T C_D^{-1}(g(m)-d_{obs})\right] \tag{3.26}$$

as its stationary distribution, whereas in the next section we construct a procedure so that at any T_l, the images generated with simulated annealing correspond to states of a Markov chain which has

$$\pi(m,T_l) = a_l \exp\left[-\frac{1}{2T_l}(m-m_0)^T C_M^{-1}(m-m_0)\right.$$

$$\left.-\frac{1}{2T_l}(g(m)-d_{obs})^T C_D^{-1}(g(m)-d_{obs})\right] \quad (3.27)$$

as its stationary distribution. Note at $T_l = 1$, both Eqs. 3.26 and 3.27 reduce to the correct a posteriori probability density, $\pi(m)$, that we wish to sample. Thus, if our objective is to sample $\pi(m)$ whether Eq. 3.26 or 3.27 is preferable depends on which probability density best serves to achieve the following two objectives: (i) minimize the chances of getting trapped at a local minimum and (ii) obtain an initial image at $T_l = 1$ so that the Markov chain constructed at $T_l = 1$ has a very short or no transient period. Intuition might suggest that Eq. 3.26 is preferable for minimizing the possibility of getting trapped in a local minimum since division of the pressure mismatch term by a $T_l > 1$ gives it less weighting in the probability density function. That is, it may reduce the chances of getting trapped in a local minimum which gives a good pressure mismatch.

On the other hand, if we wish to use simulated annealing to generate a maximum a posteriori estimate by driving "temperature" to zero, it seems clear that at low temperature ($T_l < 1$), we wish to use simulated annealing to construct states of a Markov chain with $\pi(m,T_l)$ as its stationary distribution. As $T_l \rightarrow 0$, $\pi(m,T_l) \rightarrow 0$ for all models except those constrained in the set of all maximum a posteriori estimates. If as $T_l \rightarrow 0$, $\hat{\pi}(m,T_l) \rightarrow 0$ we construct a Markov chain with stationary distribution, then in the limit we approach one of the most probable models based on a probability distribution satisfying

$$\tilde{\pi}(m) \propto \exp\left[-\frac{1}{2}(g(m)-d_{obs})^T C_D^{-1}(g(m)-d_{obs})\right] \quad (3.28)$$

instead of one of the most probable models based on the correct a posteriori probability density function of Eq. 1.5.

Sampling from a distribution with this approach can be seen as a two-phase process. In the *initial phase*, we start from some initial state, and use simulated annealing until reaching a temperature equal to one. At this point we start the *sampling phase*, in which we continue from the state reached at the end of the initial phase, using a Markov chain as described in previous chapter, proceeding for long enough that a close approximation to the equilibrium distribution has been reached, and enough subsequent data has been collected to produce Monte Carlo estimates of adequate accuracy.

At first glance, it appears that this application has no advantages over simply constructing a Markov chain. However, a Markov chain typically has a transitionary (transient) period during which the states generated do not correspond to the a posteriori probability density function which is the stationary distribution of the Markov chain. Our view is that there may be cases where simulated annealing will help us generate an initial state which approximately represents a sample from the stationary distribution and thus decrease the length of the transient period. Such a procedure would be useful for cases where the gradient method gets trapped in an unacceptable local minimum or does not converge. Since simulated annealing algorithms are designed to sample from the set of global minima, we should be able to apply the method to obtain a good starting point, m_r for the Markov chain where m_r yields calculated pressure data which are in good agreement with observed pressure data.

3.3.2 Perturbations Based on Sensitivities

Here, we consider a Markov chain Monte Carlo method in which we propose a transition from image m^i to image m^j by sampling from a normal distribution of zero mean and unit variance to obtain a vector of independent normal deviates Z^j and setting

$$m^j = \mu_r + L_r Z^j, \tag{3.29}$$

where $L_r L^T_r$ represents the Cholesky decomposition of C_{Mr}.

Setting $m_r = m^i$ we define one reference mean μ_r and corresponding reference covariance C_{Mr}. Instead of using Eq. 2.36 and 2.37 to compute μ_r and C_{Mr}, we use

$$\mu_r = m_0 + C_M G_r^T (G_r C_M G_r^T + T_l C_D)^{-1} (d_r - G_r m_0), \tag{3.30}$$

and

$$C_{Mr} = (\frac{1}{T_l} G_r C_D^{-1} G_r + C_M^{-1})^{-1}, \tag{3.31}$$

where T_l denotes the temperature. Using T_l in Eq. 3.31 (compare with Eq. 2.37) broadens the distribution associated with C_{Mr} and allows us to propose new simulated annealing images from a larger range around μ_r.

The probability of proposing this transition is

$$q_{ij} = a \exp\left(-\frac{1}{2} (m^j - \mu_r)^T C_{Mr}^{-1} (m^j - \mu_r) \right), \tag{3.32}$$

where a is a constant. The probability of proposing the reverse transition is

$$q_{ji} = a\exp\left(-\frac{1}{2}(m^i - \mu_r)^T C_{Mr}^{-1}(m^i - \mu_r)\right). \tag{3.33}$$

Similar to the derivation of Eq. 2.47, it can be shown that

$$\frac{\pi_j(T_l)q_{ji}}{\pi_i(T_l)q_{ij}} = \exp\left(\frac{1}{2T_l\sigma_d^2}(2\Delta d^{jT}\varepsilon(m^j) - 2\Delta d^{iT}\varepsilon(m^i) + \varepsilon(m^j)^T\varepsilon(m^j) - \varepsilon(m^i)^T\varepsilon(m^i))\right), \tag{3.34}$$

Thus, if we define a simulated annealing objective function by

$$O(m^i) = -\frac{1}{2\sigma_d^2}(2\Delta d^{iT}\varepsilon(m^i) + \varepsilon(m^i)^T\varepsilon(m^i)), \tag{3.35}$$

it follows that

$$O(m^j) - O(m^i) = -\frac{1}{2\sigma_d^2}[2\Delta d^{jT}\varepsilon(m^j) + \varepsilon(m^j)^T\varepsilon(m^j) - (2\Delta d^{iT}\varepsilon(m^i) + \varepsilon(m^i)^T\varepsilon(m^i))], \tag{3.36}$$

and

$$r_i(T_l) = \exp\left(-\frac{1}{2T_l\sigma_d^2}[2\Delta d^{jT}\varepsilon(m^j) + \varepsilon(m^j)^T\varepsilon(m^j) - (2\Delta d^{iT}\varepsilon(m^i) + \varepsilon(m^i)^T\varepsilon(m^i))]\right) \tag{3.37}$$

Comparing Eq. 3.37 and 3.34, it follows that Eq. 3.19 holds when $T_l = 1$. Thus, if we decrease the temperature through an annealing schedule until we reach $T_l = 1$ and do no further temperature decreases, the images generated at this temperature level will correspond to a state in a Markov chain which has the probability density function of Eq. 1.5 as its stationary distribution.

The preceding arguments also show that at temperatures $T_l > 1$, we are actually constructing a Markov chain which has Eq. 3.26 as its stationary distribution. Based on arguments presented in subsection 3.3.1, this approach may be more advantageous for generating an initial state for the Markov chain while the procedure that will be described in the next section is more advantageous when we are using simulated annealing to construct a maximum a posteriori estimate.

3.4 Using Simulated Annealing to Sample from the Set of Most Probable Models

While, there are a wide variety of procedures to obtain a sample from the set of most probable models that can be discussed based on the theoretical development described in sections 3.2 and 3.3, we consider only two. As discussed previously, we wish to generate a realization of the log-permeability field which corresponds to a global minimum of the objective function defined in Eq. 1.6. We consider two simulated annealing algorithms to obtain such a realization.

3.4.1 Perturbations Based on the Variogram

In this case, we define a probability density function by

$$\pi(m,T_l) = a\exp\left(-\frac{1}{2T_l}(m-m_0)^T C_M^{-1}(m-m_0)\right.$$

$$\left.-\frac{1}{2T_l}(g(m)-d_{obs})^T C_D^{-1}(g(m)-d_{obs})\right), \qquad (3.38)$$

so for any log-permeability field on the gridblocks, i.e., for any m^i,

$$\pi_i(T_l) = \pi(m^i, T_l) = c\exp\left(-\frac{1}{2T_l}(m^i - m_0)^T C_M^{-1}(m^i - m_0) - \frac{1}{2T_l\sigma^2_d}(\Delta d^{iT}\Delta d^i)\right). \quad (3.39)$$

Note the argument of the exponent in the probability density function of Eq. 3.38 is equal to the argument in the a posteriori probability density function divided by T_l. At T_l=1 Eq. 3.38 gives the true a posteriori probability density function.

Now suppose in the Markov chain Monte Carlo method, we propose a transition from state i to state j by sampling from a normal distribution of zero mean and unit variance to obtain a vector of independent normal deviates Z^j and set

$$m^j = m_0 + \sqrt{T_l}LZ^j, \quad (3.40)$$

where LL^T still represents the Cholesky decomposition of C_M. The expected value of m^j is still m_0, but the covariance matrix for m^j is now given by

$$C_M(T_l) = T_l C_M. \quad (3.41)$$

The probability of proposing this transition is given by

$$q_{ij} = a\exp\left(-\frac{1}{2T_l}(m^j - m_0)^T C_M^{-1}(m^j - m_0)\right), \quad (3.42)$$

where a is a constant. The probability of proposing the reverse transition is

$$q_{ji} = a\exp\left(-\frac{1}{2T_l}(m^i - m_0)^T C_M^{-1}(m^i - m_0)\right). \quad (3.43)$$

The preceding procedure for proposing states is equivalent to generating each component of Z^j from a Gaussian distribution with mean zero and variance T_l and deleting $\sqrt{T_l}$ in Eq. 3.40. With the objective function defined in Eq. 3.23, it now follows easily that

$$\frac{\pi_j(T_l)q_{ji}}{\pi_i(T_l)q_{ij}} = \exp\left(-\frac{1}{2\sigma^2{}_d T_l}(\Delta d^{jT}\Delta d^j - \Delta d^{iT}\Delta d^i)\right) =$$

$$= \exp\left(-\frac{1}{T_l}(O^j - O^i)\right) = r_{ij}(T_l). \quad (3.44)$$

Thus, at each temperature level T_l, the states generated from simulated annealing with the objective function of Eq. 3.23 will correspond to states of a Markov chain that in the limit represent realizations of the probability density function of Eq. 3.38. As discussed previously, this approach may be more advantageous when $T_l < 1$ and we are using simulated annealing to determine the maximum a posteriori estimate.

3.4.2 Perturbations Based on Sensitivities

Here, at each temperature level, T_l, we define one reference mean μ_r and corresponding reference covariance C_{Mr} as in Eqs. 2.36 and 2.37. There are various ways that these could be chosen. One way is to base them on the first simulated annealing image m_r accepted at T_l. Another way is to base them on the first image m_r that decreases the pressure misfit below the lowest pressure misfit obtained at the previous temperature level. Once the reference model m_r, μ_r and C_{Mr} are chosen, we keep them fixed for all iterations at temperature level T_l.

We now consider a Markov chain Monte Carlo method in which we propose a transition from image m^i to image m^j by sampling from a normal distribution of zero mean and unit variance to obtain a vector of independent normal deviates Z^j and setting

$$m^j = \mu_r + \sqrt{T_l} L_r Z^j, \tag{3.45}$$

where $L_r L^T_r$ represents the Cholesky decomposition of C_{Mr}. The probability of proposing this transition is

$$q_{ij} = a \exp\left(-\frac{1}{2T_l} (m^j - \mu_r)^T C_{Mr}^{-1} (m^j - \mu_r) \right), \tag{3.46}$$

where a is a constant. The probability of proposing the reverse transition is

$$q_{ji} = a \exp\left(-\frac{1}{2T_l} (m^i - \mu_r)^T C_{Mr}^{-1} (m^i - \mu_r) \right). \tag{3.47}$$

Similar to the derivation of Eq. 2.47, it can be shown that

$$\frac{\pi_j(T_l) q_{ji}}{\pi_i(T_l) q_{ij}} = \exp\left(\frac{1}{2T_l \sigma_d^2} (2\Delta d^{jT} \varepsilon(m^j) - 2\Delta d^{iT} \varepsilon(m^i) + \varepsilon(m^j)^T \varepsilon(m^j) - \varepsilon(m^i)^T \varepsilon(m^i)) \right), \tag{3.48}$$

where $\pi_i(T_l)$ and $\pi_j(T_l)$ are obtained from Eq. 3.38. Thus, if we define a simulated annealing objective function by

$$O(m^i) = -\frac{1}{2\sigma_d^2} (2\Delta d^{iT} \varepsilon(m^i) + \varepsilon(m^i)^T \varepsilon(m^i)), \tag{3.49}$$

then the images generated by simulated annealing at the temperature level T_l will correspond to a states in a Markov chain which has the probability density function of Eq.

3.38 as its stationary distribution. Again this approach may be more advantageous for $T_l <$ 1 when we are using simulated annealing to construct a maximum a posteriori estimate.

3.4.3 The Two-Point Swap Perturbation

Here, we consider a classical simulated annealing approach. In this procedure, the objective function is still given by Eq. 1.6. An uncorrelated field of gridblock values of $\ln(k)$ is generated from the log-normal univariate distribution for permeability (histogram) with given mean and variance. These values represent the only set of gridblock permeability values that can be generated in the procedure. Perturbations to the log-permeability field are proposed by randomly selecting two gridblocks and interchanging their permeability values. The simulated annealing algorithm terminates when the desired pressure mismatch is achieved or the acceptance ratio becomes too low. The problem with this approach is that the state space is limited to all possible configurations of the set of M permeability values drawn from the histogram so it may be difficult to converge to a image that gives a very low pressure misfit. The final image obtained, however, will be an approximation to the most probable model (maximum a posteriori) contained in this reduced state space. One way to put this in the context of Markov chain is to use the following objective function in simulated annealing:

$$O(m) = \frac{1}{2}[(m-m_0)^T C_M^{-1}(m-m_0) + (g(m)-d_{obs})^T C_D^{-1}(g(m)-d_{obs})]. \qquad (3.50)$$

From Eqs. 3.38 and 3.50, it follows that

$$\frac{\pi_j(T_l)q_{ji}}{\pi_i(T_l)q_{ij}} = \frac{q_{ji}}{q_{ij}}\exp\left(-\frac{(O^i - O^j)}{T_l}\right) = \frac{q_{ji}}{q_{ij}}r_{ij}(T_l). \quad (3.51)$$

Thus, Eq. 3.19 is satisfied if and only if

$$q_{ij}=q_{ji}. \quad (3.52)$$

If the two-point swap is used to propose a transition from state i to state j in the Markov chain Monte Carlo method, then Eqs. 3.51 and 3.19 hold and the images generated by simulated annealing at temperature level T_l correspond to states in a Markov chain with the probability density function of Eq. 3.38 as its stationary distribution.

It is important to note that using simulated annealing with two point swapping and a final temperature of $T_l = 1$ does not appear to be a reliable way to sample the a posteriori probability density function. Because all states generated at $T_l = 1$ simply represent rearrangements of the M fixed permeability values originally drawn, these states cannot represent a correct sampling of the a posteriori probability distribution. Intuitively, it appears that one could generate N realizations which represent a proper sampling by starting with N different initial configurations and applying simulated annealing with two-point swaps N times, one time for each initial set of log-permeability values. For each simulated annealing run, one could select the final image accepted at $T_l = 1$. Even assuming the validity of such a procedure, the computational expense would seem to rule out this approach for the problems of interest in this work. Even though the two-point swap does have the possible advantage that resulting realizations honor the histogram.

3.4.4 Including the Variogram in the Simulated Annealing Objective Function

As mentioned previously, many practitioners use simulated annealing to minimize objective functions that contain measures of the misfit of the variogram. For the problem considered here, such an approach would utilize the two component objective function defined by Eqs. 3.1-3.3, with the weights determined by Eqs. 3.4 and 3.5. The initial distribution of gridblock values of $\ln(k)$ is randomly selected from the log-normal distribution for permeability and new simulated annealing images are proposed by randomly selecting two gridblocks and swapping their log-permeability values. This procedure represents a common simulated annealing procedure currently practiced, see for example, refs. 24-28.

In this section, we show that the variogram computed from a realization of the correct probability distribution function does not necessarily give a good match to the model variogram. We also argue that the measure of variogram misfit should not be included in the objective function in the traditional way.

As in Journel and Huijbregts[36](1978), we distinguish between three different variograms: the theoretical variogram, the experimental variogram, and the local variogram. The experimental variogram is the variogram that is calculated from a limited sampling of a random function within a limited domain. The theoretical variogram is the variogram that we would calculate by averaging the variograms calculated over the entire ensemble of possible reservoirs or from exhaustive sampling of a reservoir whose dimensions are many times larger than the range of the correlation.

In practice, we can only estimate the theoretical variogram, based on a priori knowledge of what is reasonable, on closely spaced measurements from an outcrop, on limited samples obtained throughout the field, or on some combination of all of these. The important point is that the variogram model used for kriging and simulation is an estimate of the theoretical variogram for the ensemble of possible reservoirs or, equivalently, for a reservoir of infinite extent.

The local variogram, $\gamma_v(h)$, is the variogram that one would calculate from complete knowledge of a single realization of the random field within a domain v of limited extent. The expected value of the local variogram (for lags that can be calculated) is the depicted theoretical variogram, but the local variogram can fluctuate substantially around the expected value when the dimensions of the domain v are only a few times larger than the practical range of the corresponding theoretical variogram denoted as $\gamma(h)$. Unfortunately, the expected fluctuations in the calculated covariance are difficult to compute because they involve the calculation of moments of order 4 of the random variable. It appears that, for random variables whose spatial correlation is governed by a linear variogram, the relative fluctuation variance, defined by $E([\gamma(h)-\gamma_v(h)]^2/[\gamma(h)]^2$ is $o(h/L)$, i.e., approaches zero as $h/L \rightarrow 0$ for lags h that are small compared to dimensions L of the local domain. We would expect this same behavior to hold for other variogram models, including exponential and spherical.

Instead of calculating the fluctuation variance for large lags, we can determine the variability of the dispersion variance[36] of the random function for the domain v. Although not shown here, the dispersion variance computed based on an exponential variogram indicates that the variability in the sills of the local variogram will increase as the ratio of

the practical range of the variogram to the characteristic length of the domain increases. This ratio is denoted here by R. We expect that similar results will apply to other variogram models.

To illustrate the preceding discussion, we consider a spherical isotropic variogram for log-permeability with sill equal to 0.25 and range equal to 600 ft. For a two-dimensional reservoir of dimensions 1,500 ft by 1,500 ft partitioned into 225 uniform 100 ft by 100 ft gridblocks, we computed the associated covariance matrix and then generated 1000 unconditional realizations and then computed the local variogram corresponding to each realization. Fig. 3.1a illustrates the results. The top and bottom curves represent the two extreme local variograms obtained; i.e., all other local variograms obtained fell somewhere between these two. The curve between these two local variograms is the theoretical variogram. Note there is considerable scatter in the local variograms and except at small lag distances, a local variogram may differ significantly from the theoretical variogram when the ratio of the range of the variogram to a characteristic dimension of the domain is large. In this example, this ratio is given by $R = 600/1500 = 0.4$.

Next we repeated the experiment with range of the variogram decreased to 300 ft so $R = 0.2$. Fig. 3.1b illustrates the variability in local variograms computed from a thousand realizations. All other local variograms computed lie between the two local variograms shown in Fig. 3.1b. Note in this case, the variability in the local variograms is much smaller than for the previous example and local variograms are reasonably close to the theoretical variogram.

Under our assumptions, the probability density function for the prior model is given by Eq. 1.1 which is repeated here as

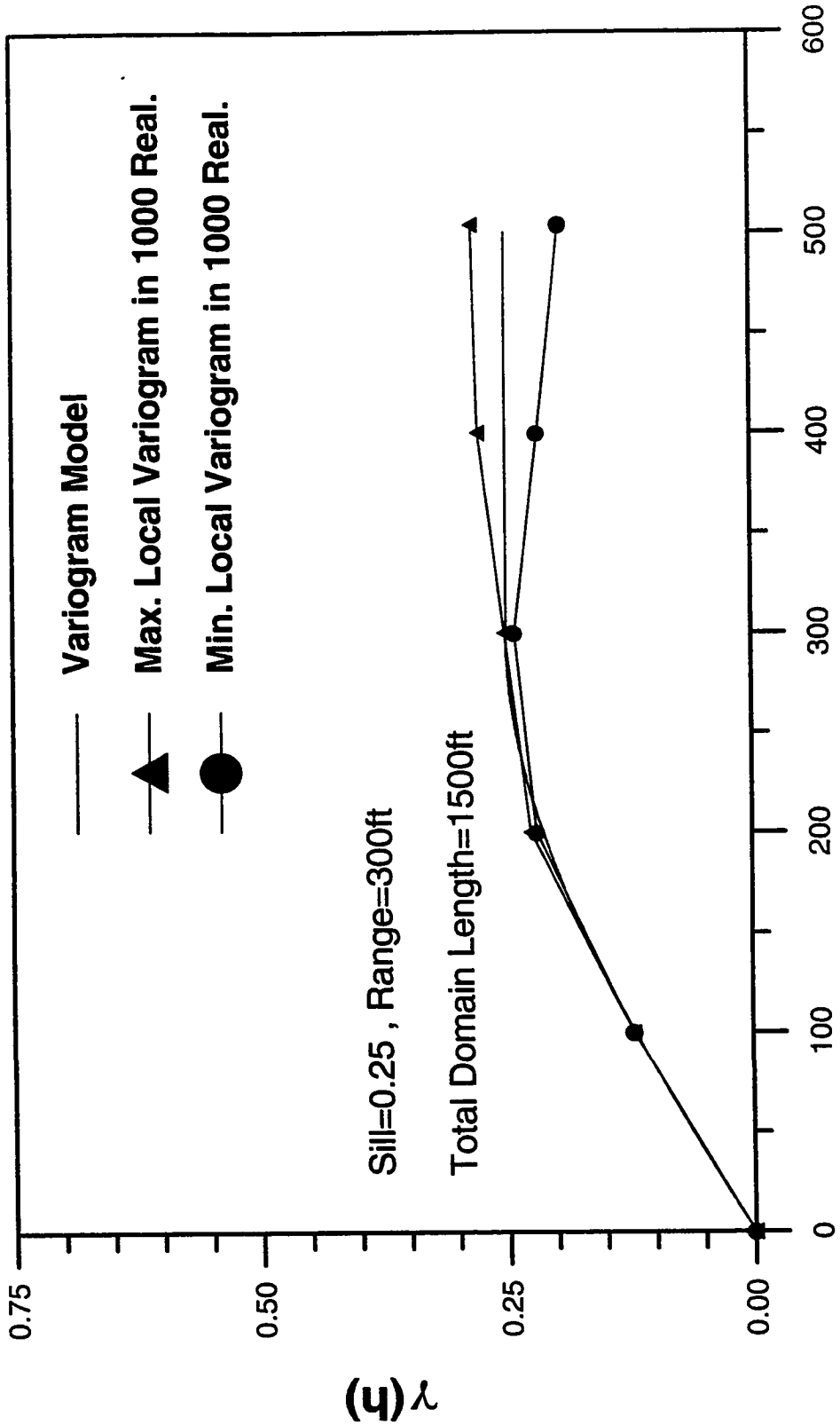

Fig. 3.1a - Local Variogram Fluctuations Compared to Theoretical Variogram (R = 0.4).

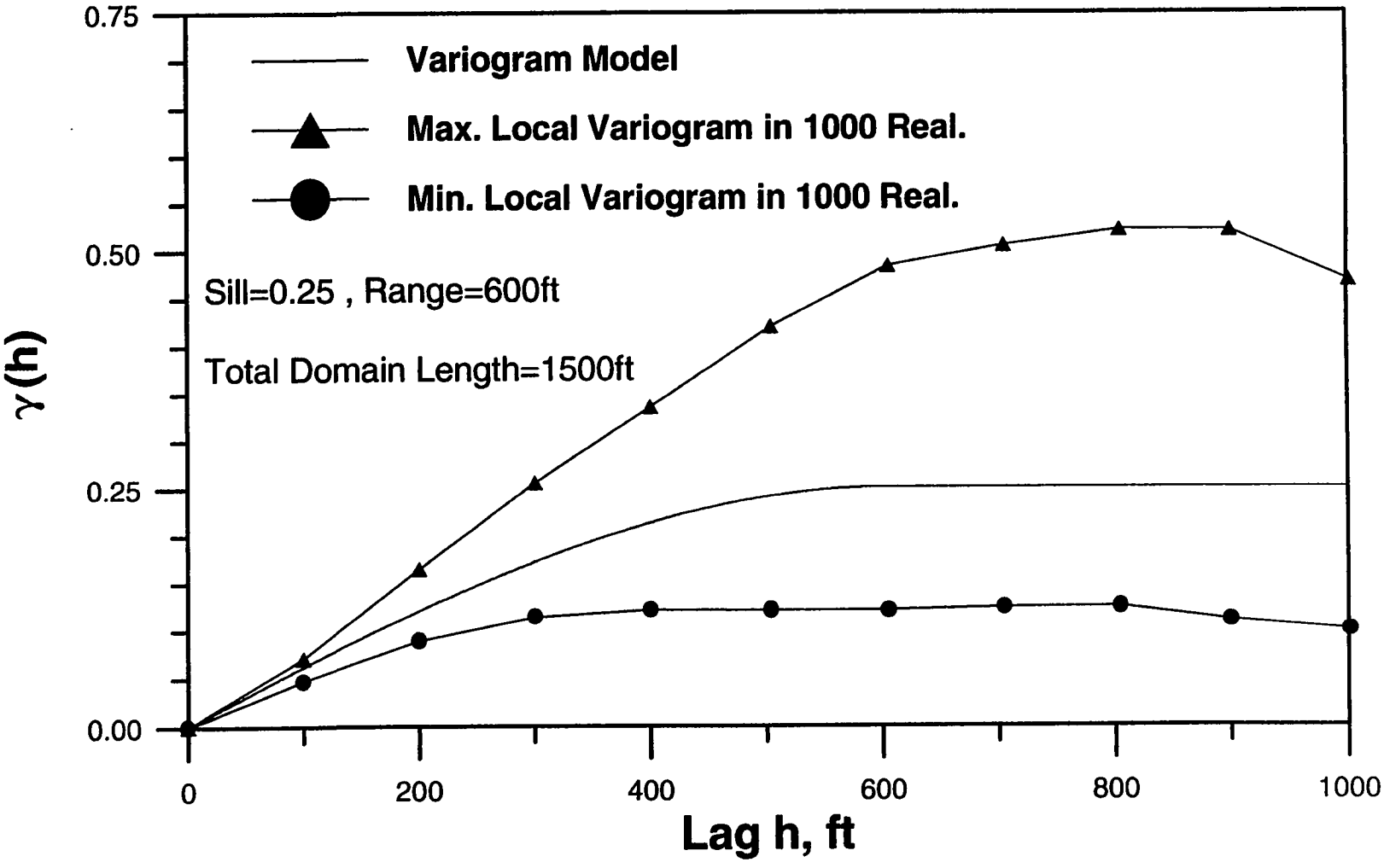

Fig. 3.1b - Local Variogram Fluctuations Compared to Theoretical Variogram (R = 0.2).

$$\rho_M(m) = b\exp\left(-\frac{1}{2}(m-m_0)^T C_M^{-1}(m-m_0)\right), \tag{3.53}$$

where b is a constant. The most probable model, the maximum a posteriori estimate, is the one that minimizes the objective function

$$\hat{O} = \frac{1}{2}[(m-m_0)^T C_M^{-1}(m-m_0)]. \tag{3.54}$$

Clearly, the most probable model is the prior mean m_0 and this is what one would obtain from a proper application of simulated annealing using the objective function $\hat{O}$ if we drive the temperature to zero. Clearly, this is not what we want to determine since we already know the prior mean. If we apply simulated annealing using as the objective function O_1 of Eq. 3.1, we will obtain a log-permeability field which matches the model variogram closely, but it will be neither a realization based on the correct probability distribution nor the most probable model since the correct probability density function is given by Eq. 3.53 which is equivalent to

$$\rho_M(m) = b\exp(-\hat{O}). \tag{3.55}$$

When we are using O_1 as the objective function in simulated annealing, we are assuming that the correct probability density function has the form

$$\hat{\rho}(m) = a\exp(-O_1). \tag{3.56}$$

Most generally, if we use the objective function O_{SA} of Eq. 3.1 in simulated annealing, we will obtain the most probable model only if the true a posteriori probability distribution is of the form

$$f_M(m) = \hat{a}\exp[-\hat{b}O_{SA}]. \tag{3.57}$$

where $\hat{a}$ and $\hat{b}$ are constants. However, assuming we wish to condition a Gaussian random field with covariance C_M and mean m_0 to pressure data, the correct a posteriori probability distribution is given by Eq. 1.5, and Eq. 3.57 is incorrect.

Based on the preceding arguments and the results of Figs. 3.1a and 3.1b, incorporating the variograms into the objective function is not recommended.

3.5 Computational Examples

3.5.1 Example 1 - Sampling from the Set of Most Probable Models with Simulated Annealing

Here, we apply the methods discussed previously for sampling from the set of most probable models. All data and information are the same as in the Example 1 described in Chapter II. The only difference is that the variance of all pressure measurement errors was assumed to be given by $\sigma^2_d = 0.25\ \text{psi}^2$.

Fig. 3.2 shows the maximum a posteriori estimate obtained from the gradient method and Fig. 3.3 compares the local variogram obtained from Fig. 3.2 with the model variogram. Note that the sill of the variogram for the maximum a posteriori estimate is much lower than the one for the model variogram. This is, as expected, a reflection of the fact that the maximum a posteriori estimate of the permeability field is much smoother than the true permeability field.

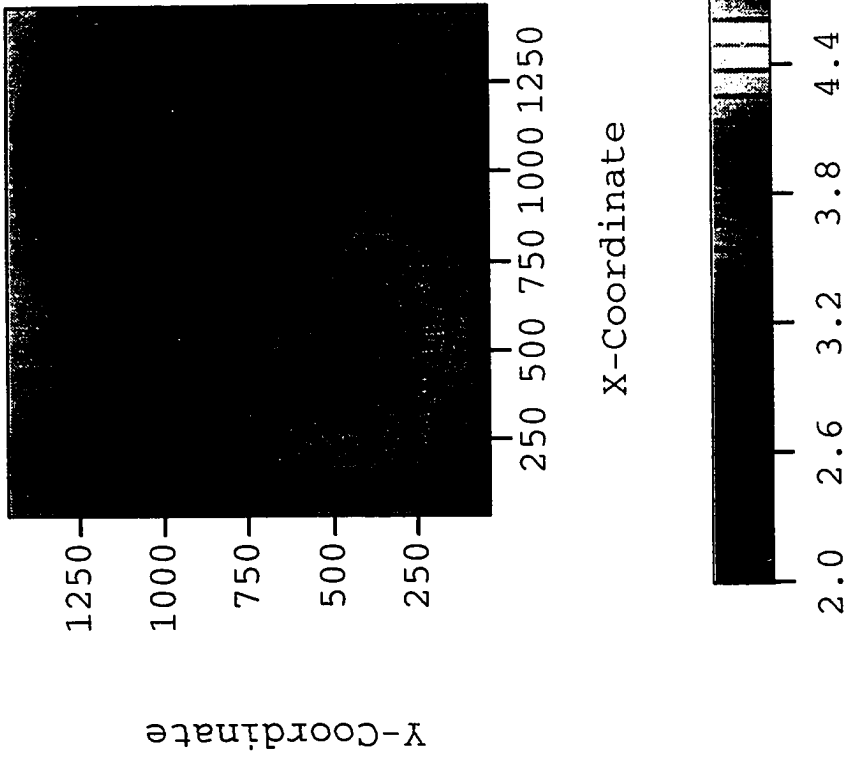

Fig. 3.2 - Maximum a Posteriori Estimate from Gradient Method.

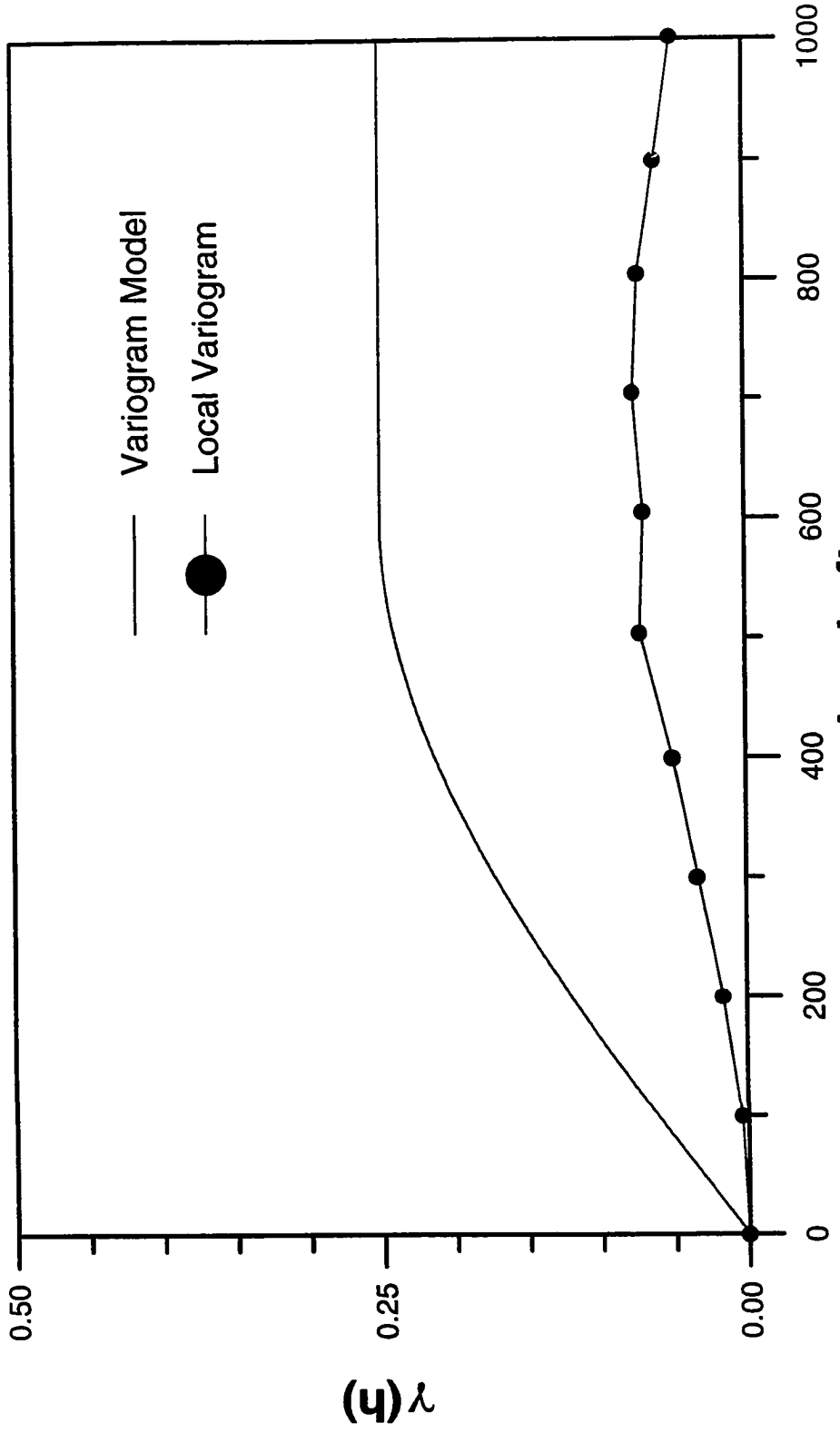

Fig. 3.3 - Variograms, Model versus Calculated from Maximum a Posteriori Estimate from Gradient Method.

Figure 3.4 shows our estimate of the most probable model obtained with the procedure of section 3.4.2, i.e. starting with perturbations based on the variogram, once a pressure misfit less than or equal to one psi ($\sqrt{(\Delta d^{iT}\Delta d^{i})/N_d} \leq 1\,\text{psi}$) was obtained, perturbations based on sensitivity coefficients (Eq. 3.45) were used with the objective function of Eq. 3.49. Note the sample from the set of most probable models shown in Fig. 3.4 is very similar to the maximum a posteriori estimate obtained with the gradient method. Generation of the realization of Fig. 3.4 required 9.0 hours of computer time, whereas, it required one minute to compute the maximum a posteriori estimate of Fig. 3.2 using the gradient method. Fig. 3.5 shows the local variogram computed from the log-permeability field of Fig. 3.4 compared to the model variogram.

Fig. 3.6 presents the estimate of a realization corresponding to the global minimum of the objective function (i.e., the argument of the exponential in Eq. 1.5) obtained by starting with an uncorrelated field of log-permeability defined on the gridblocks and performing two-point swaps. This image only roughly approximates the ones obtained in Figs. 3.2 and 3.4; 27 hours of computer time were used to achieve this estimate. Note in this method, we only rearrange the original set of M values of $\ln(k)$ drawn from the log-normal distribution form permeability. Using this set the lowest pressure mismatch we were able to obtain was 0.8 psi, whereas, using the preceding method, we were able to obtain a pressure mismatch less than 0.5 psi. Evaluated at the log-permeability field of Fig. 3.6, the value of the "objective function" of Eq. 3.50 was 418. Since the value of this objective function evaluated at the permeability field of Fig. 3.4 is 32.0 and the value of this objective function at the maximum a posteriori estimate is 5.2, clearly, we have not found an image which gives a global minimum of the objective function. Note, however,

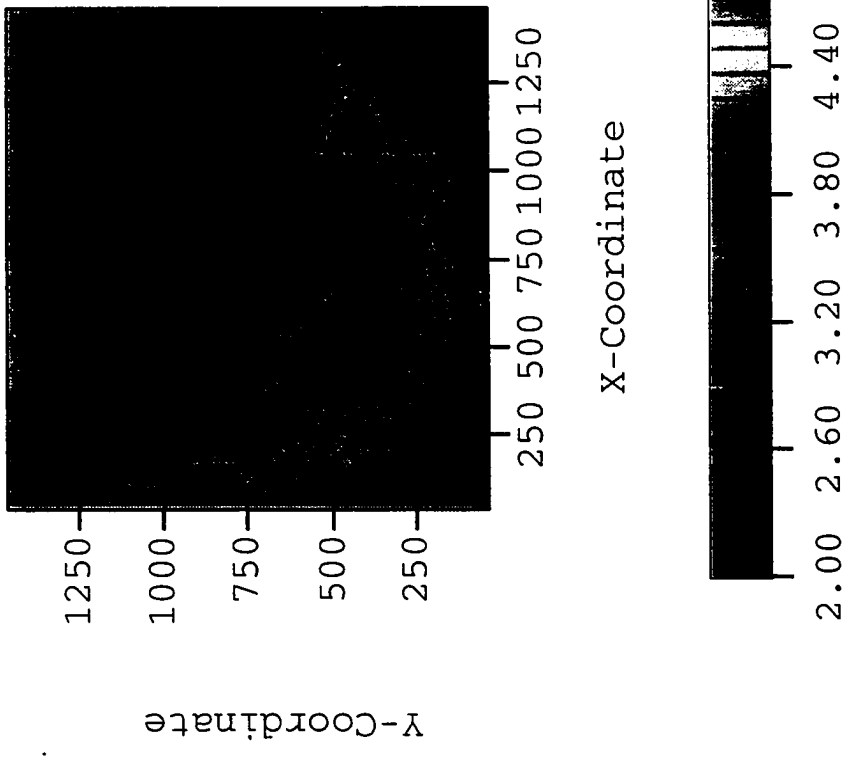

Fig. 3.4 - Maximum a Posteriori Estimate - Local Perturbations Based on Sensitivity Coefficients.

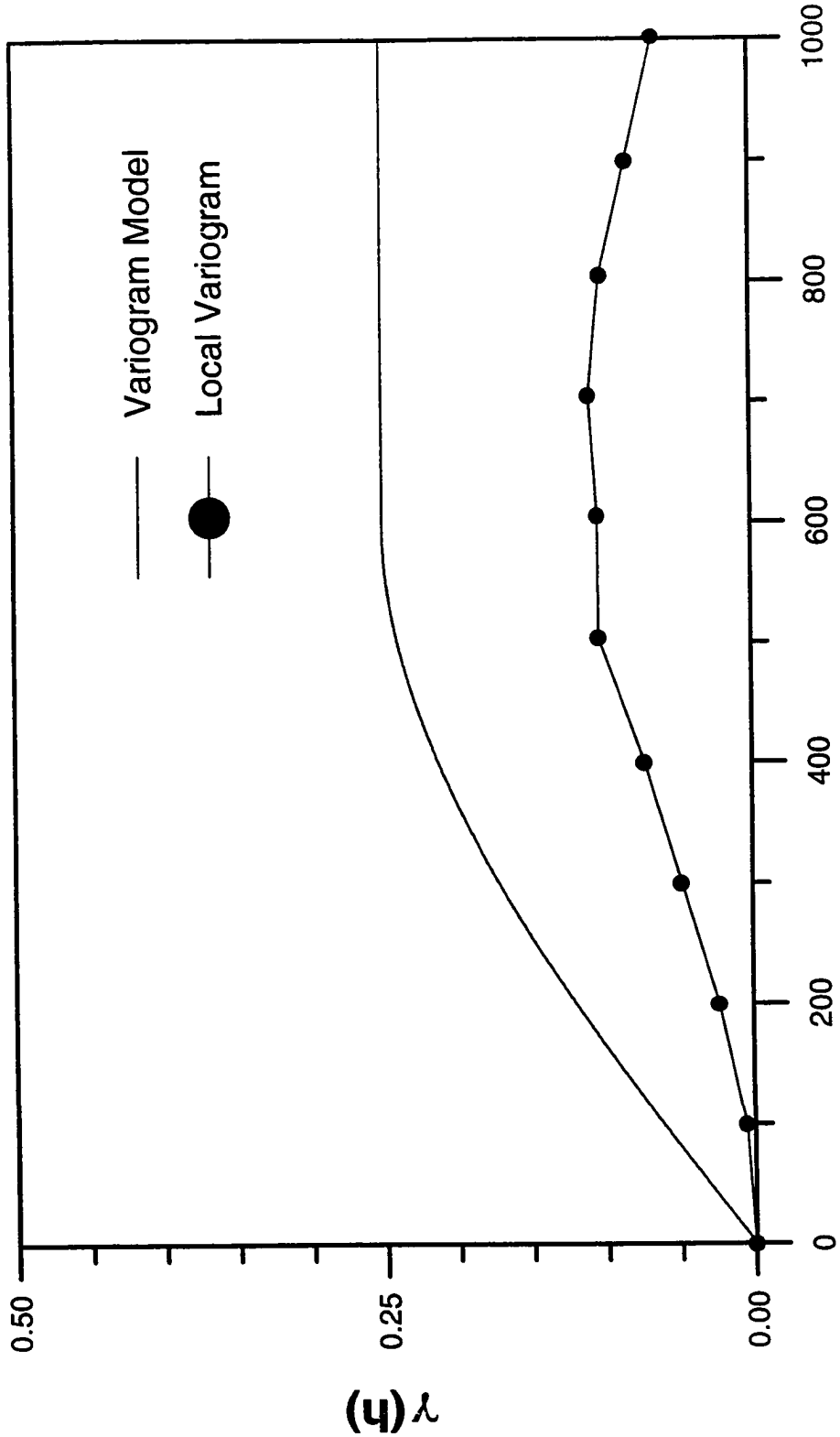

Fig. 3.5 - Variograms, Model versus Calculated from Maximum a Posteriori Estimate - Local Perturbations Based on Sensitivity Coefficients.

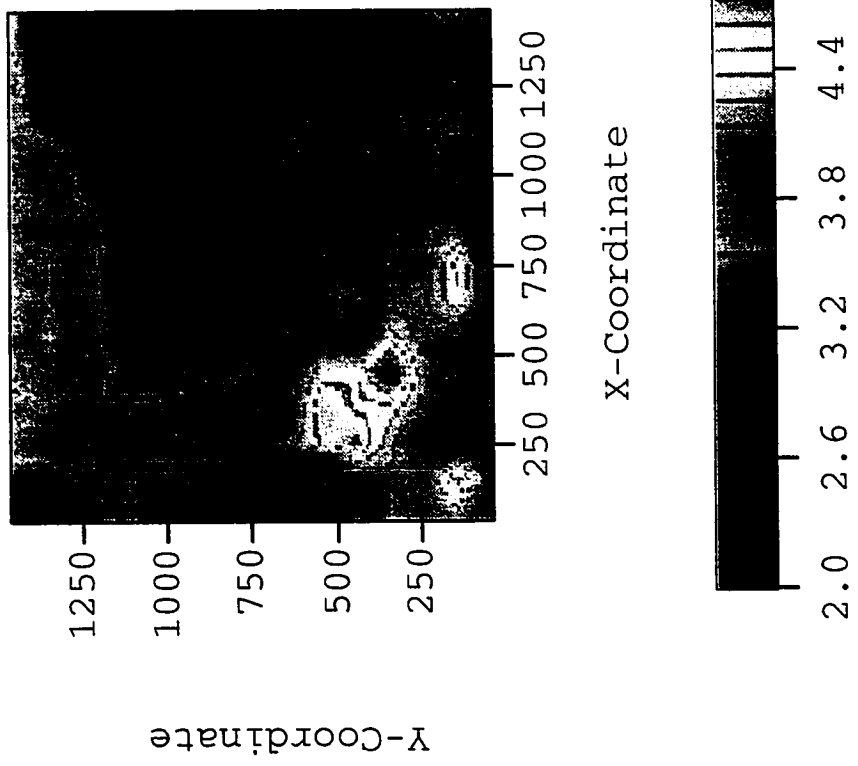

Fig. 3.6 - Maximum a Posteriori Estimate - Simulated Annealing with Two-Point Swap.

the results of Fig. 3.6 were obtained by starting with an uncorrelated log-permeability distribution at the gridblocks and performing two-point swaps. For such a procedure, we cannot expect to obtain a value of the objective function close to the true global minimum. Fig. 3.7 compares the variogram computed from the results of Fig. 3.6 with the model variogram.

Fig. 3.8 shows the log-permeability field obtained using simulated annealing with the objective function of Eq. 3.1 and using two-point swaps beginning with a set of gridblock values drawn from the log-normal distribution of permeability. Note this result differs significantly from the maximum a posteriori estimate obtained from the gradient method. This third method represents the standard simulated annealing approach. Since it is based on an incorrect objective function, it is not expected to yield a reliable maximum a posteriori estimate. The value of the objective function of Eq. 3.50 evaluated at this image gives 800. Again, we have not found a global minimum of the correct objective function. The lowest pressure mismatch obtained was 0.9 psi. The local variogram computed from the distribution of Fig. 3.8 is compared with the model variogram in Fig. 3.9.

3.5.2 Example 2 -Using Simulated Annealing to Obtain the Initial State of a Markov Chain

In this example, we consider the use of simulated annealing to obtain an initial state for the Markov chain. In this procedure, we begin simulated annealing at a high temperature and gradually reduce the temperature using the procedures described in Appendix A. At high temperatures, we use perturbations based on the variogram, and use

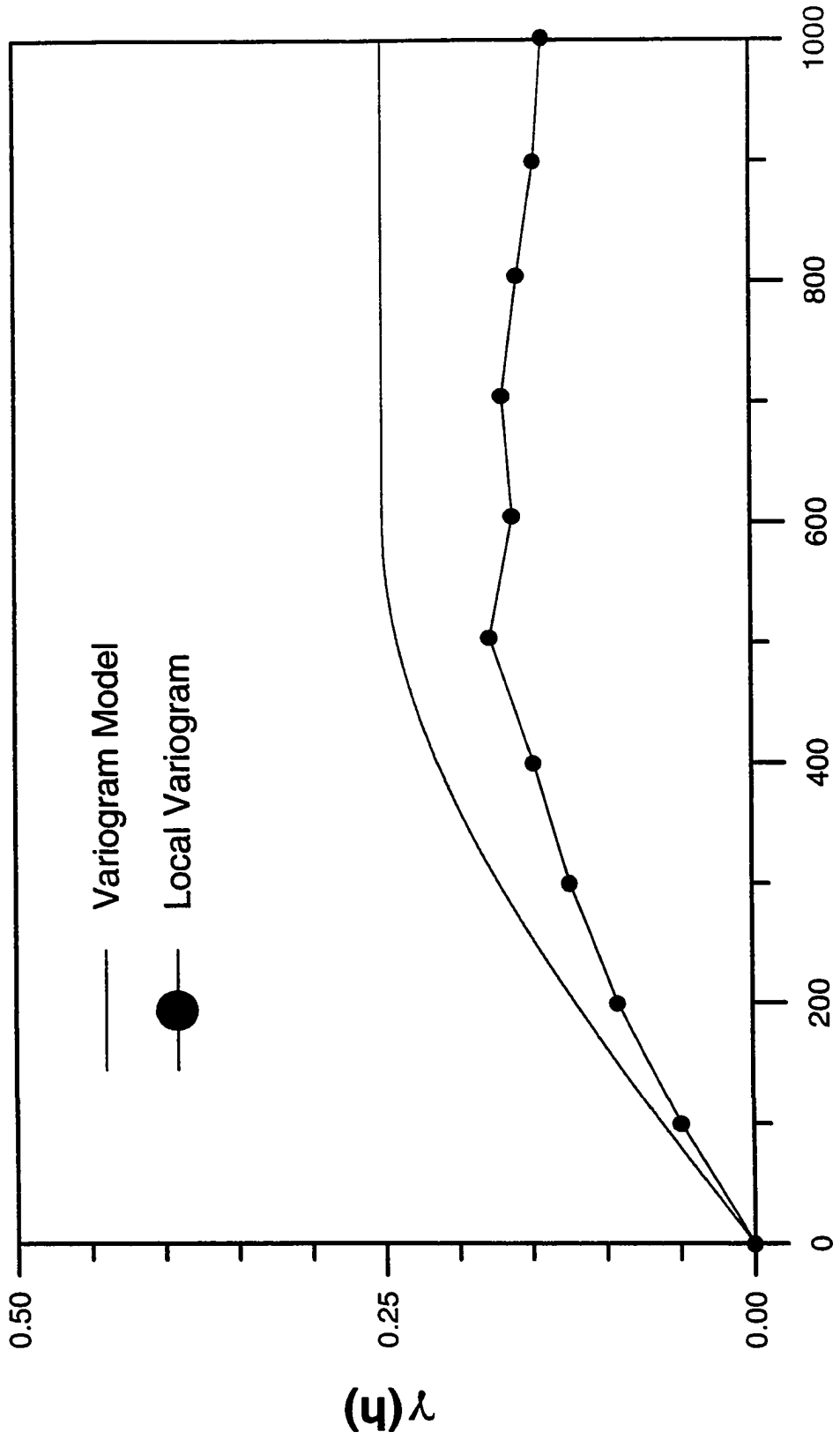

Fig. 3.7 - Variograms, Model Versus Calculated from Maximum a Posteriori Estimate - Two-Point Swap.

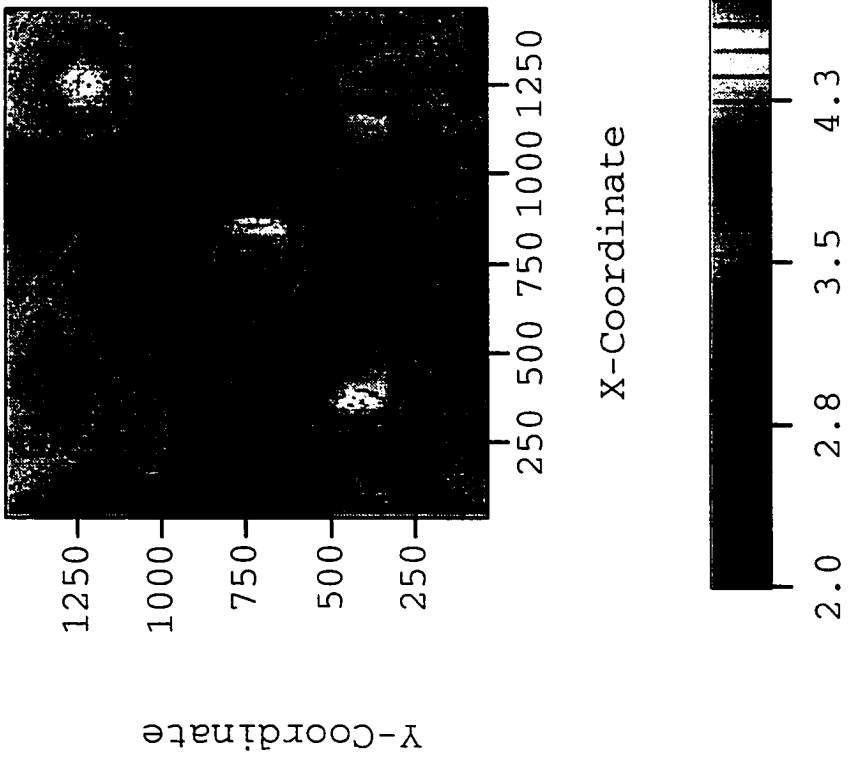

Fig. 3.8 - Maximum a Posteriori Estimate - Standard Simulated Annealing.

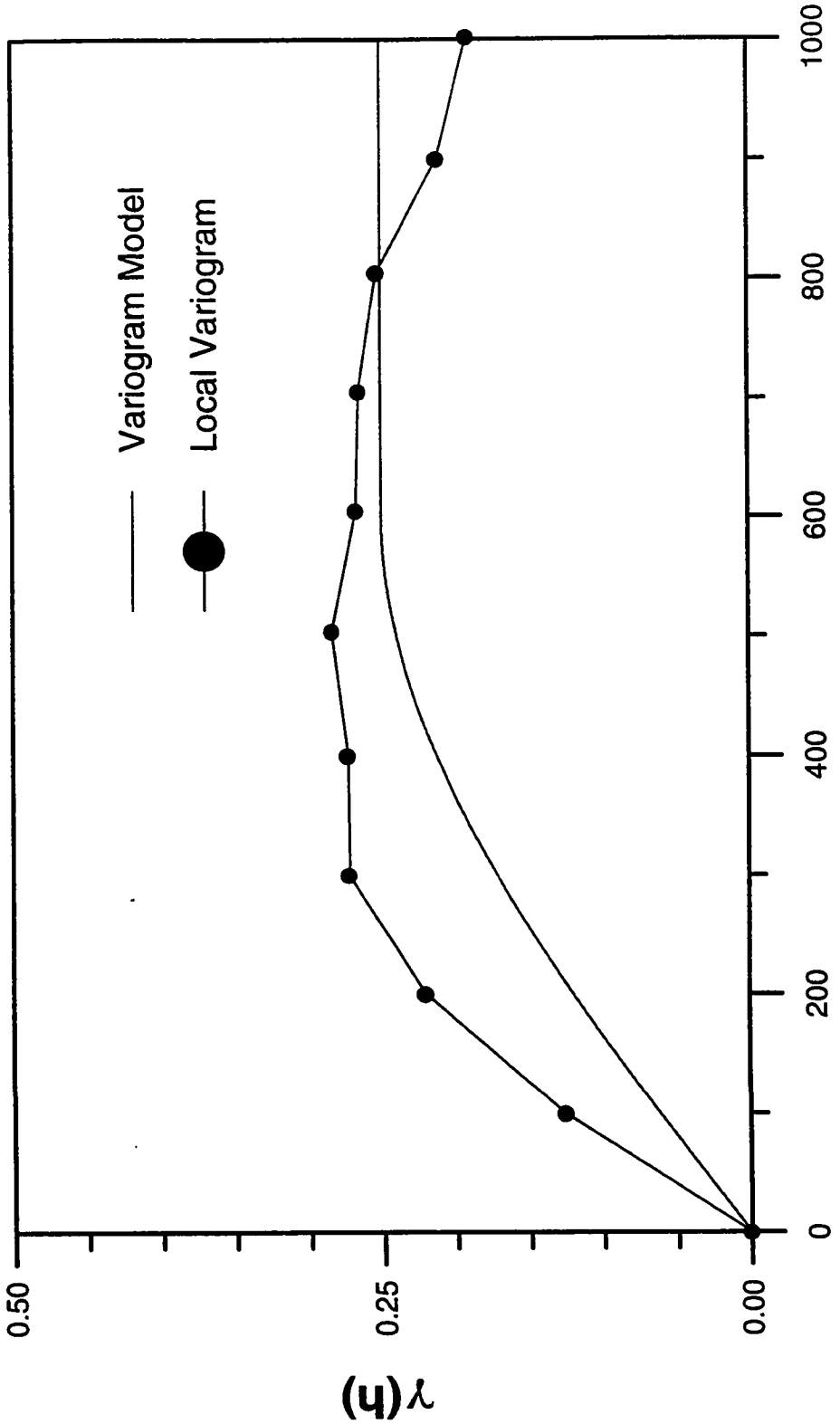

Fig. 3.9 - Variograms, Model versus Calculated from Maximum a Posteriori Estimate - Standard Simulated Annealing.

the objective function of Eq. 3.23 which is related to the pressure misfit. This is performed until we obtain a pressure misfit of 1 psi or less, i.e., until we obtain an image m^i in simulated annealing such that $\sqrt{(\Delta d^{iT}\Delta d^i)/N_d} \leq 1$ psi.

From this point on, we change the perturbation mechanism to use perturbations based on the sensitivity coefficients. Setting $m_r = m^i$ we compute μ_r and C_{Mr}, using Eqs. 3.30 and 3.31. At each temperature level, we propose a local perturbation by resampling one component of the vector of normal deviates to obtain Z^j, and then propose a new image as in Eq. 3.29. At this point we use the objective function of Eq. 3.35. The reference values μ_r, and the associated covariance matrix (C_{Mr}) are updated every time we obtain a pressure misfit less than the previous one.

Once we have reduced the temperature to T_i=1, we update the reference values using the first image accepted which gives a pressure mismatch lower than the smallest value obtained previously. From this point on μ_r and C_{Mr} are held fixed, i.e., μ_r is the last reference mean calculated in the simulated annealing process. Thus, at this point, we begin to construct a Markov chain using Hastings' acceptance criterion (Eq. 2.14) determined by Eq. 2.50. Local perturbations are used throughout the process so again, the set of states in the Markov chain must be thinned.

In this example section, we compare four methods for sampling the correct a posteriori probability density function expressed by Eq. 1.5. The first, second and third methods are the Markov chain Monte Carlo methods discussed previously in Chapter II. In the first method considered, the sampling is performed by using local perturbations that are based on the Cholesky decomposition of the prior covariance matrix. Both, the second

and third methods use perturbations which are based on the maximum a posterior estimate and on the Cholesky decomposition of the a posteriori matrix (C_{MP}). The second method uses global while the third uses local perturbations. The fourth method is the one described in the beginning of this subsection and uses simulated annealing to overcome the transient period that is present in a Markov chain run. In all that follows in this subsection, the pressure misfit is given by $\sqrt{(\Delta d^{iT} \Delta d^i) / N_d}$.

3.5.2.1 Example 2a - Variance = 0.25

Here, all information and data are the same as for Example 1 described in Chapter II. 50000 iterations were allowed for the sampling phase in each of the four methods analyzed.

One way to compare the four methods is to look at the generation of acceptable independent realizations for each. Fig. 3.10 presents a plot of the value of the "objective function" of accepted states versus the perturbation number for all the four methods. A corresponding plot of the pressure misfit is shown in Fig. 3.11. The perturbation number in all the figures in this example refers to the number of transitions proposed, but only the values of accepted transitions are plotted. Rejected transitions result in acceptable images (the last accepted image), but those are not plotted. Although a total number of 50000 iterations were allowed for the sampling phase for each method. For the second and third method that uses perturbations based on the a posteriori covariance matrix as well for the fourth method that uses simulated annealing to initialize the Markov chain, only the result

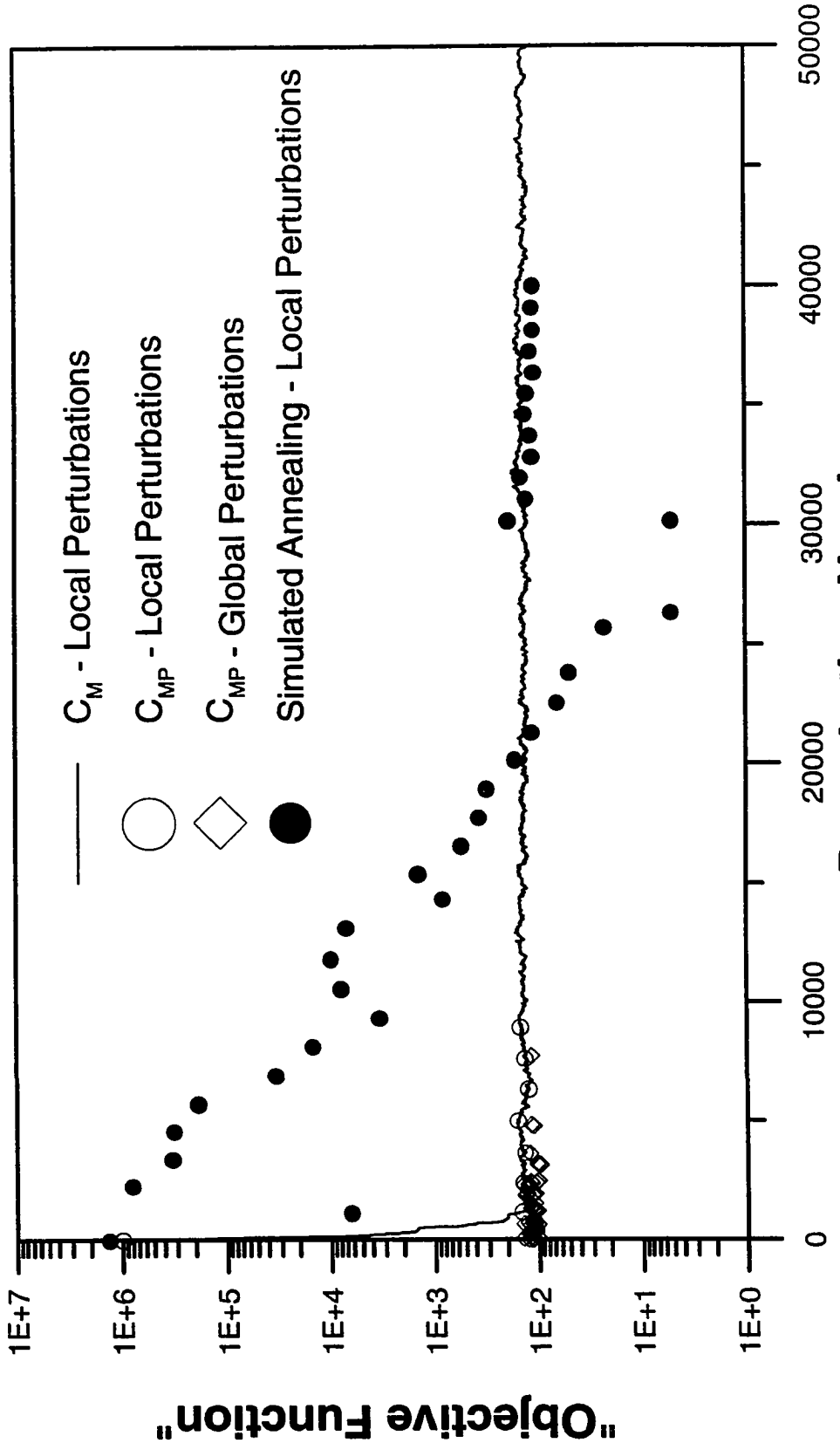

Fig. 3.10 - Values of the Objective Function for the Four Markov Chain Monte Carlo Methods (Variance = 0.25).

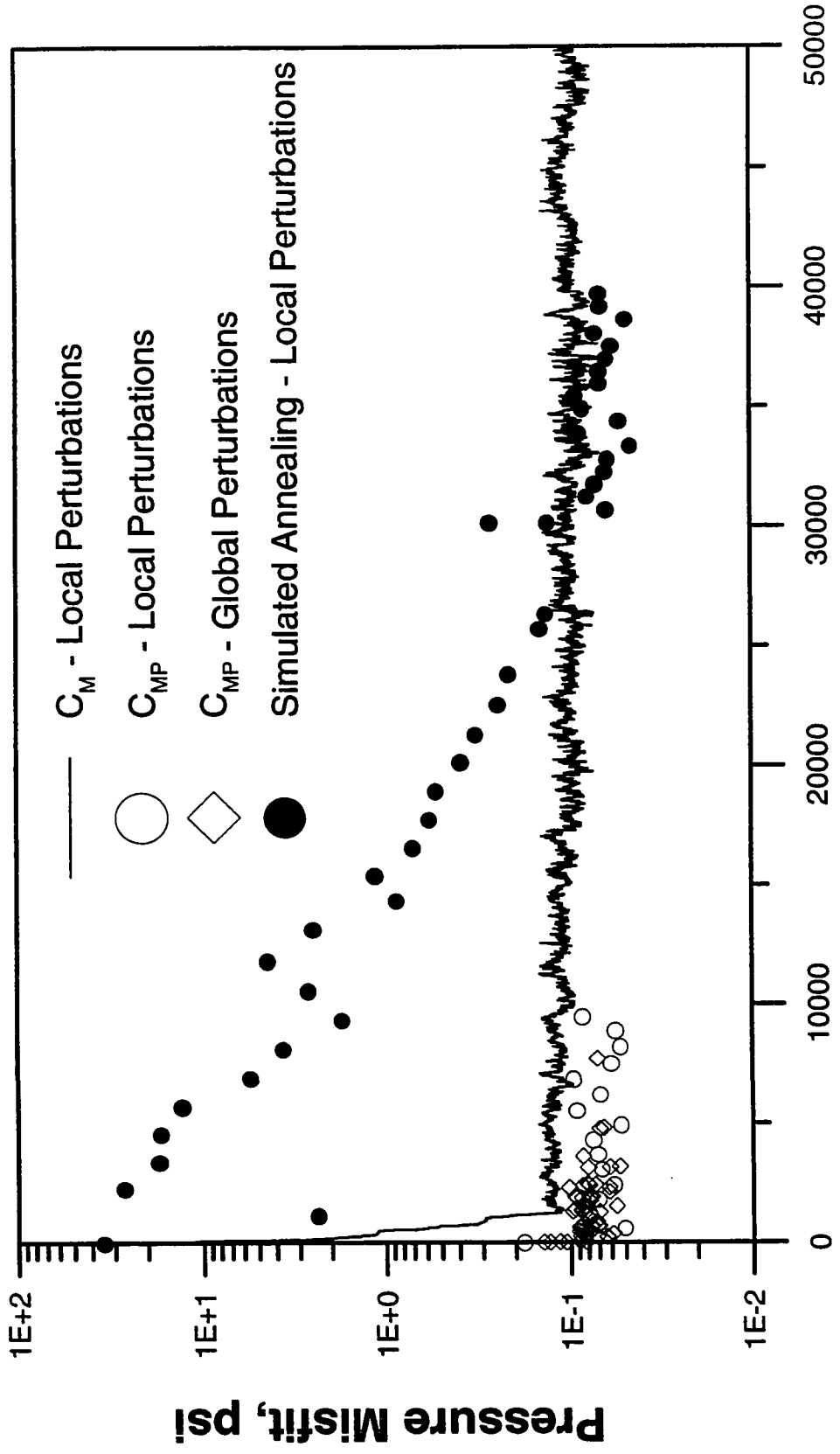

Fig. 3.11 - Values of the Pressure Misfit for the Four Markov Chain Monte Carlo Methods (Variance = 0.25).

of the 10000 first iterations are plotted in Figs. 3.10 and 3.11. For the fourth method the values plotted correspond to values obtained during the initial and sampling phases.

As mentioned before, the results of Figs. 3.10 and 3.11 allow us to discuss procedures for determining when the transient period of the Markov chain has ended. Based on previous arguments we see that for the Markov chain Monte Carlo methods that use the a posteriori covariance obtained from the gradient method to generate perturbations, the transient period is so short that it is difficult to see any evidence of it in Fig.3.10.The method developed here that uses simulated annealing reaches a stationary distribution after about 31000 perturbations. The first Markov chain Monte Carlo method, which used local perturbations based only on the variogram, has not reached the stationary distribution after 50000 perturbations.

Figure 3.12 shows log-permeability values at four gridblock positions (see locations depicted in Fig. 2.4.6) versus the perturbation number when using local perturbations based on the a posteriori covariance matrix with simulated annealing to initialize the Markov chain Monte Carlo method. The values of the log-permeability values are highly correlated, as expected, when using local perturbations to generate new states as can be seen by observing Fig. 3.12.

In relation to the number of images accepted, the method that used local perturbations based on the variogram did not generate any legitimate images of the permeability field during the 50000 iterations run. The Markov chain Monte Carlo method that used global perturbations based on the a posteriori covariance matrix generated a total of 200 distinct, independent realizations. Using local perturbations based on the a posteriori covariance matrix we obtained approximately 37500 different, but

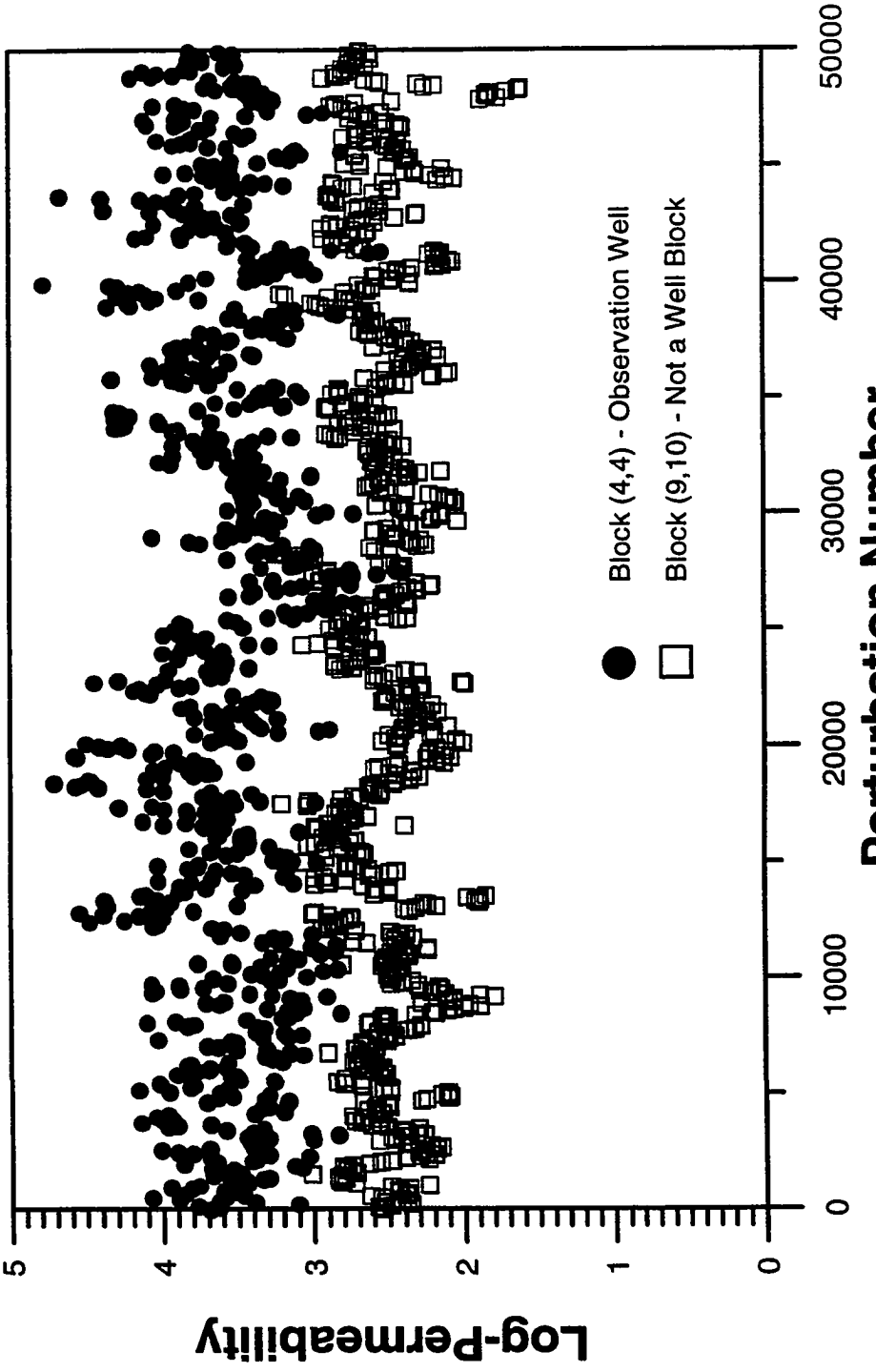

Fig. 3.12a - Log-Permeability Values During Sampling with Local Perturbations Based on Sensitivities Coefficients with Simulated Annealing to Initialize the Markov Chain (Variance = 0.25).

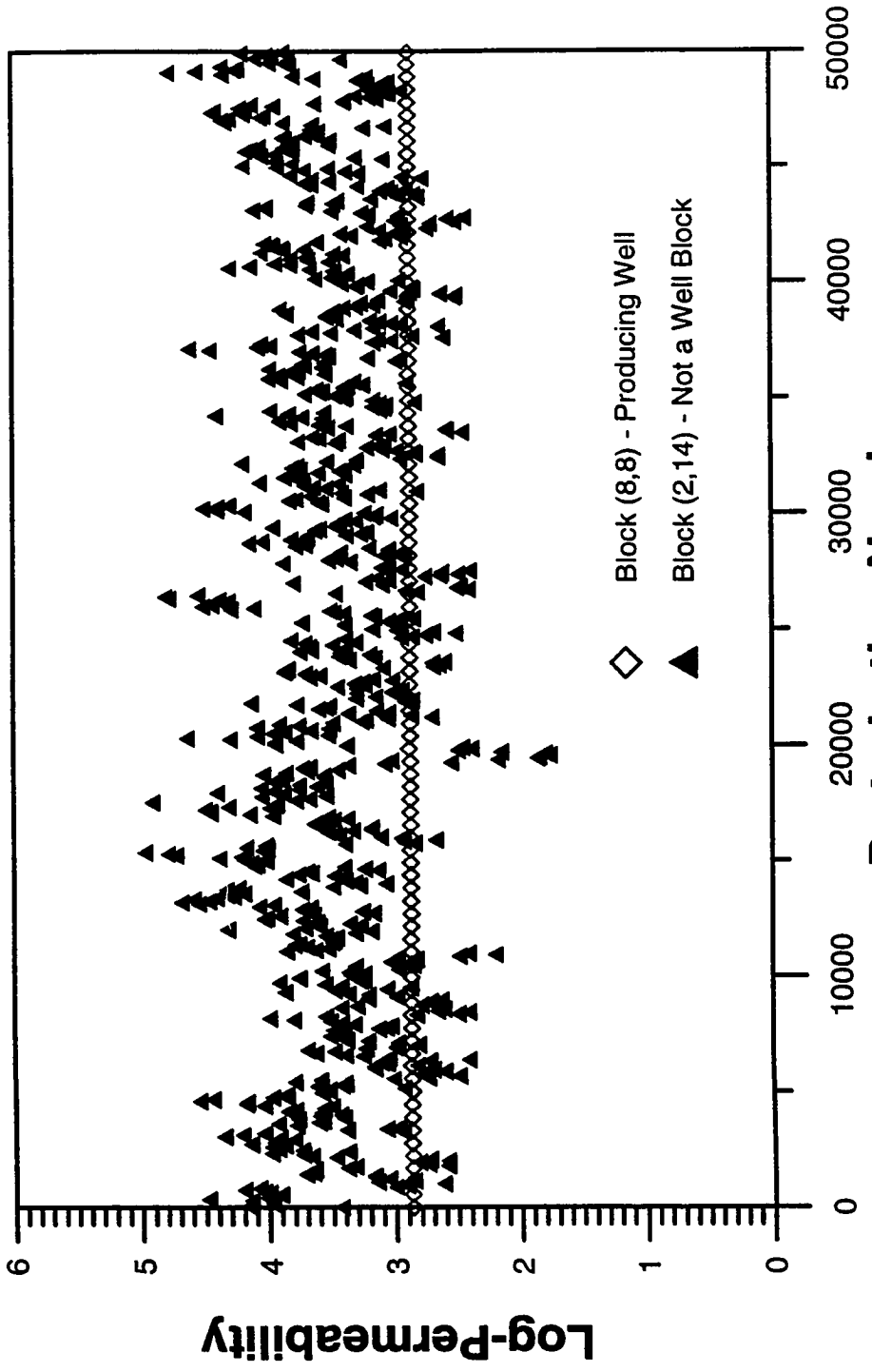

Fig. 3.12b - Log-Permeability Values During Sampling with Local Perturbations Based on Sensitivities Coefficients with Simulated Annealing to Initialize the Markov Chain (Variance = 0.25).

highly correlated images. With the method that uses simulated annealing 37000 distinct, but highly correlated images were obtained. As mentioned before, in the case of local perturbations it is necessary to thin the set of accepted images.

The thinning of images is performed by means of the experimental variogram computed from the chain of values of the objective function. Fig. 3.13 shows the experimental variogram obtained for the method that uses simulated annealing. As we can conclude from Fig. 3.13, the realizations generated are slightly correlated to lags of approximately 800. If, we retain only every 800th realization we would be left with 46 independent realizations. Recall from Chapter II, that for this same example, the Markov chain Monte Carlo method that uses local perturbations based on the a posteriori covariance matrix generated, after the thinning of images, yielded 31 independent images.

Regarding computational time, each of the four methods required approximately 13 hours to attempt 50000 perturbations during the sampling phase. However, the fourth method had an overhead of 9 hours to perform the 31000 iterations necessary to calculate an initial state for the start of Markov chain sampling period.

Figs. 3.14a and 3.14b show legitimate realizations of the log-permeability field obtained when, after using simulated annealing to overcome the transient period, a Markov chain with local perturbations based on the a posteriori covariance matrix sampling procedure was performed. Recall the true distribution is given in Fig. 2.4.2. Note that, as in the previous methods, both realizations capture the major trends present in the truth case.

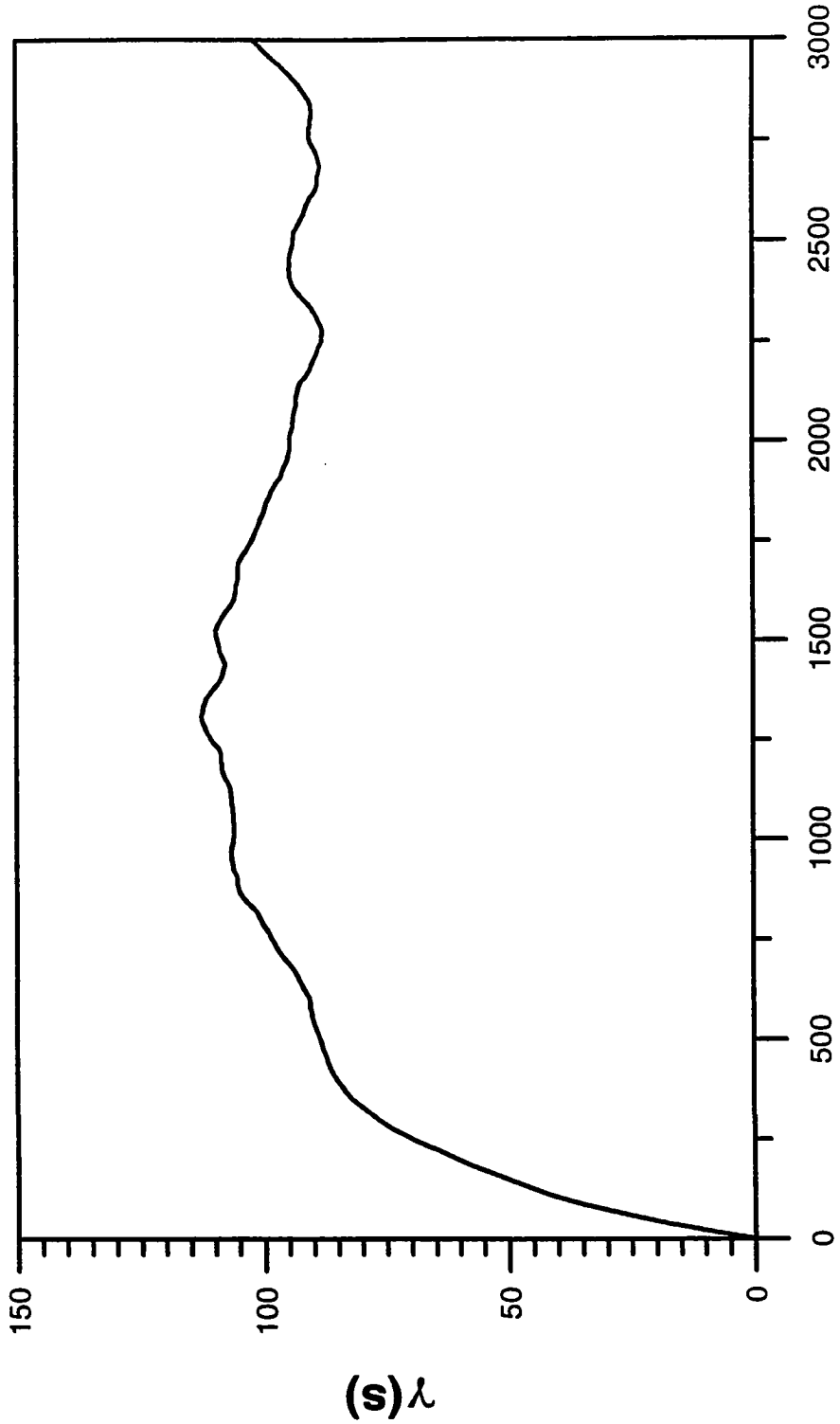

Fig. 3.13- Experimental Variogram for the Objective Function of the Chain of Realizations Generated Using Local Perturbations Based on the a Posteriori Covariance Matrix with Simulated Annealing to Initialize the Markov Chain (Variance = 0.25).

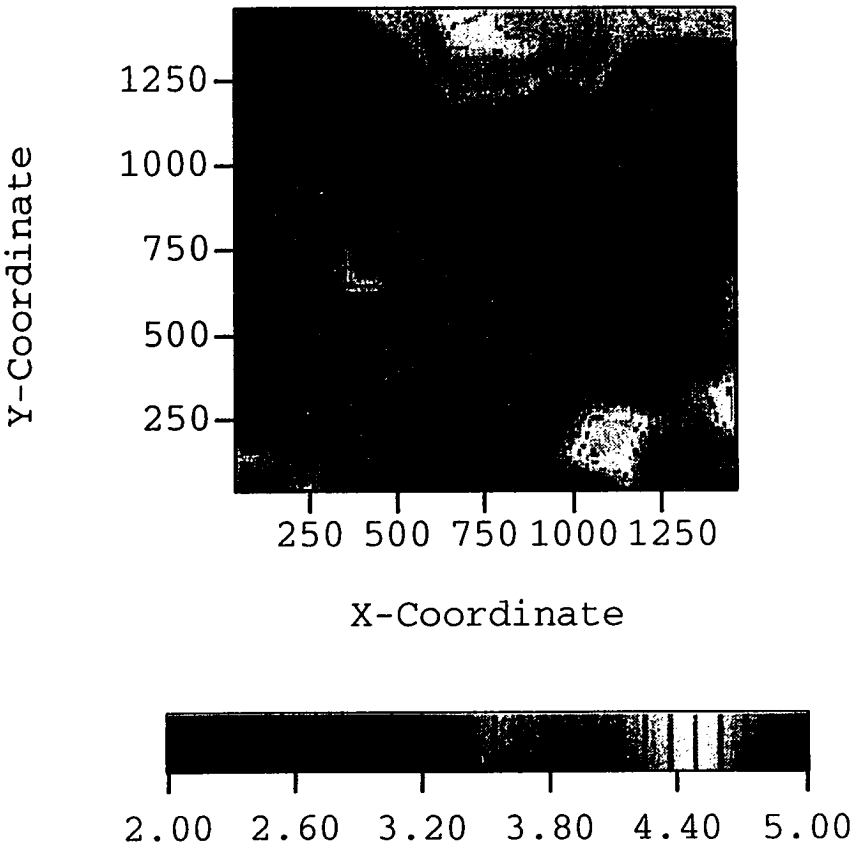

Fig. 3.14a - Realization 1 of Log-Permeability Field, Perturbations Based on Sensitivity Coefficients with Simulated Annealing to Initialize the Markov Chain Monte Carlo Method (Variance = 0.25).

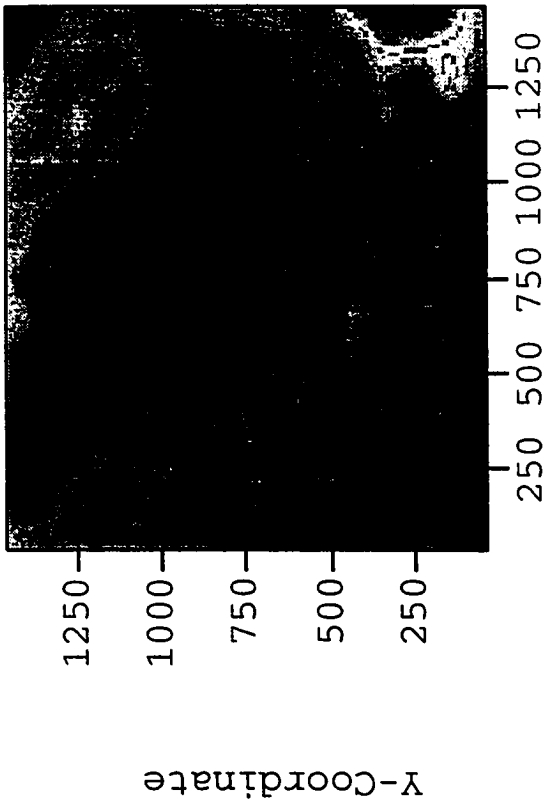

Fig. 3.14b - Realization 2 of Log-Permeability Field, Perturbations Based on Sensitivity Coefficients with Simulated Annealing to Initialize the Markov Chain Monte Carlo Method (Variance = 0.25).

3.5.2.2 Example 2b - Variance = 1.0

In this example, all information and data is the same as for Example 2 described in Chapter II. A total number of 50000 iterations for the sampling phase was allowed to each of the four methods analyzed.

Fig. 3.15 presents a plot of the value of "objective function" of accepted states versus the perturbation number for all the four methods. For the fourth method the values plotted corresponds to values obtained during the initial and sampling phases. A corresponding plot of the pressure misfit is shown in Fig. 3.16. Although a total number of 50000 iterations were allowed for the sampling phase for each method, for the second, third and fourth methods only the result of the 10000 first iterations are plotted in Figs. 3.15 and 3.16. As previously, the sequence of states constructed using local perturbations based on the variogram do not reach the stationary distribution. Between accepted transitions the state remains unchanged, but those states have not been displayed. Both methods based on using perturbations generated from the a posteriori matrix obtained from the gradient method have transient periods that are too short to be visible in Fig. 3.15. The method developed here that uses simulated annealing reaches a stationary distribution after about 30000 perturbations.

Figure 3.17 shows log-permeability values at four gridblock positions (see locations in Fig. 2.4.6) versus the perturbation number for when using local perturbations based on

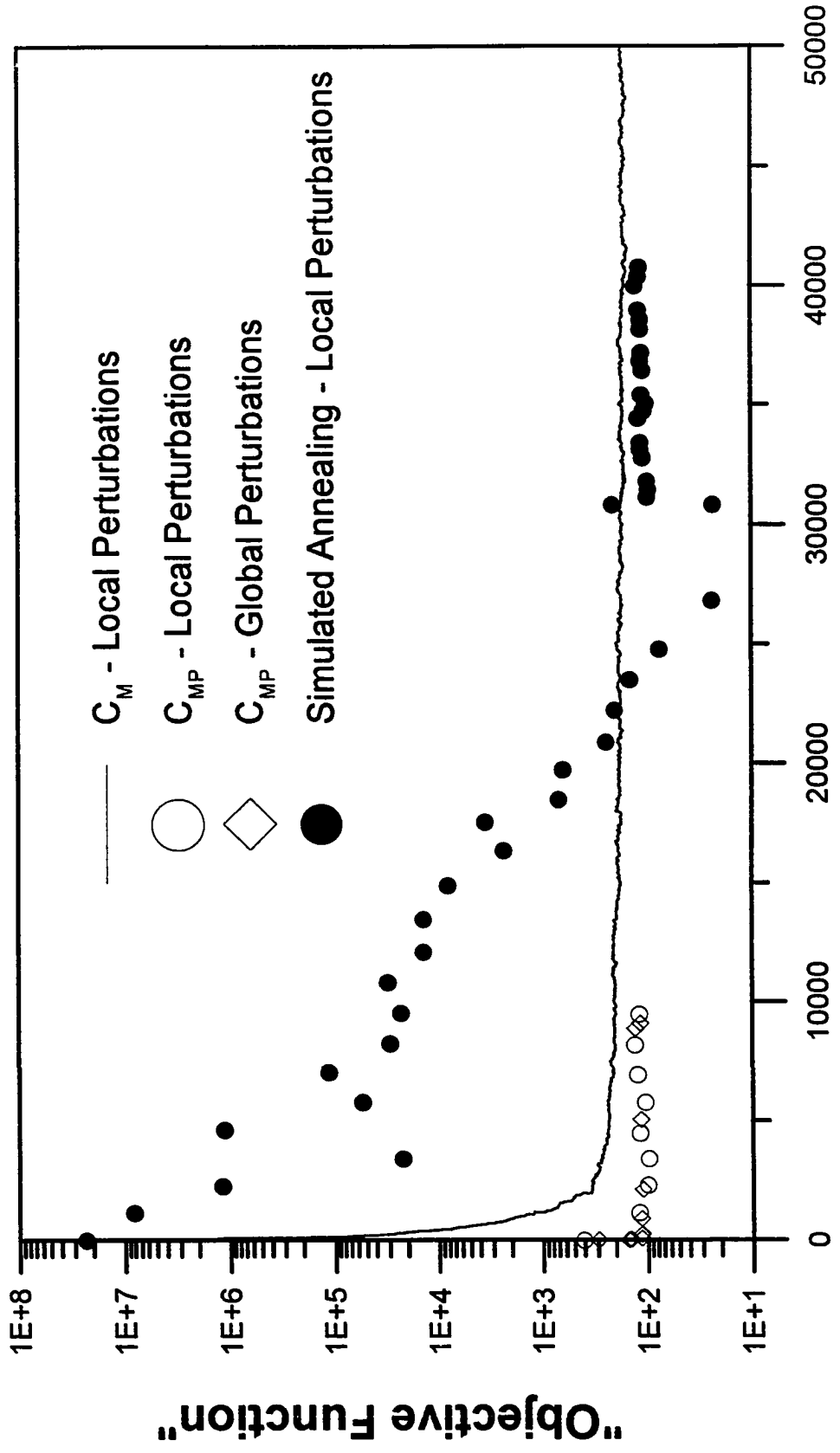

Fig. 3.15 - Values of the Objective Function for the Four Markov Chain Monte Carlo Methods (Variance = 1.0).

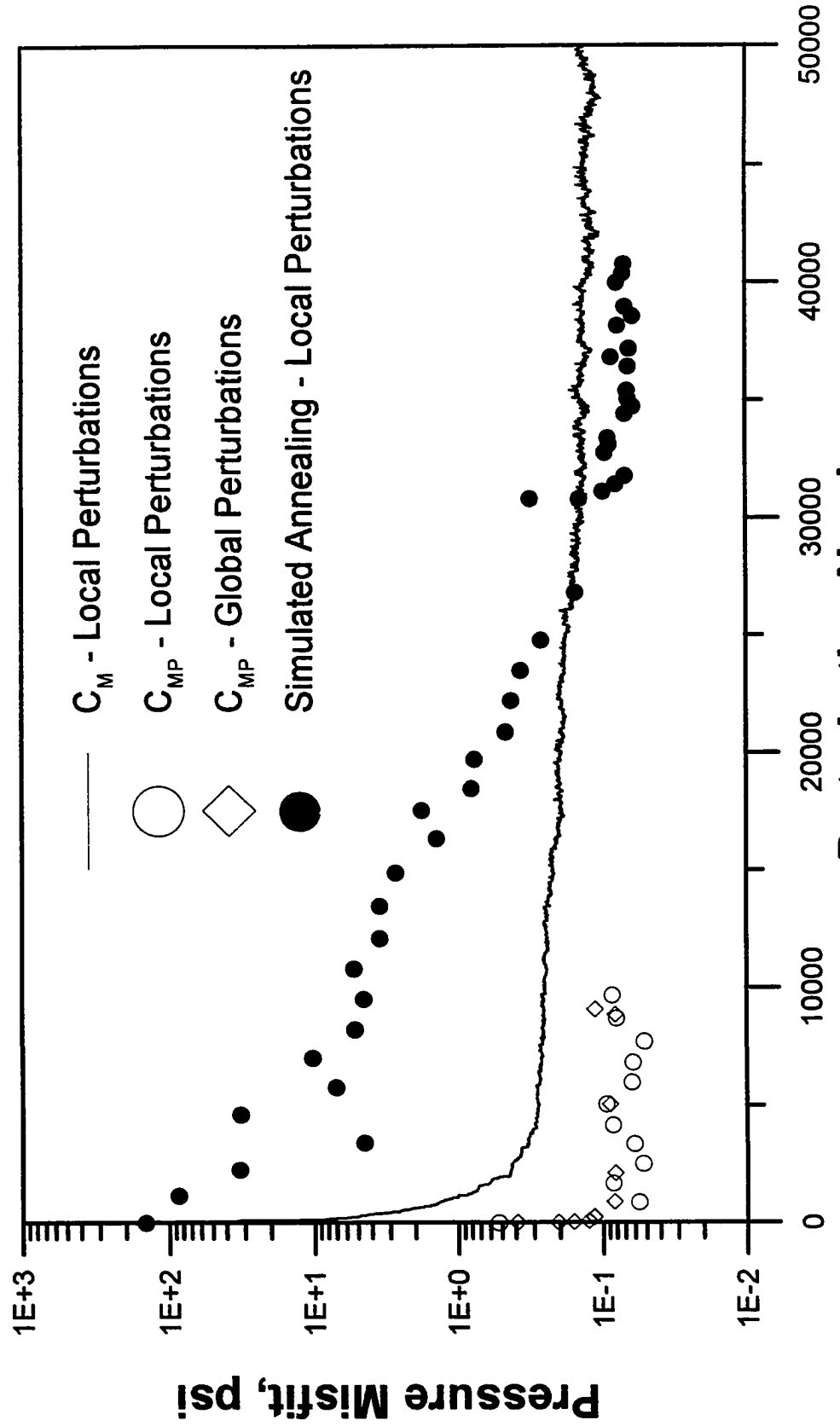

Fig. 3.16 - Values of the Pressure Misfit for the Four Markov Chain Monte Carlo Methods (Variance = 1.0).

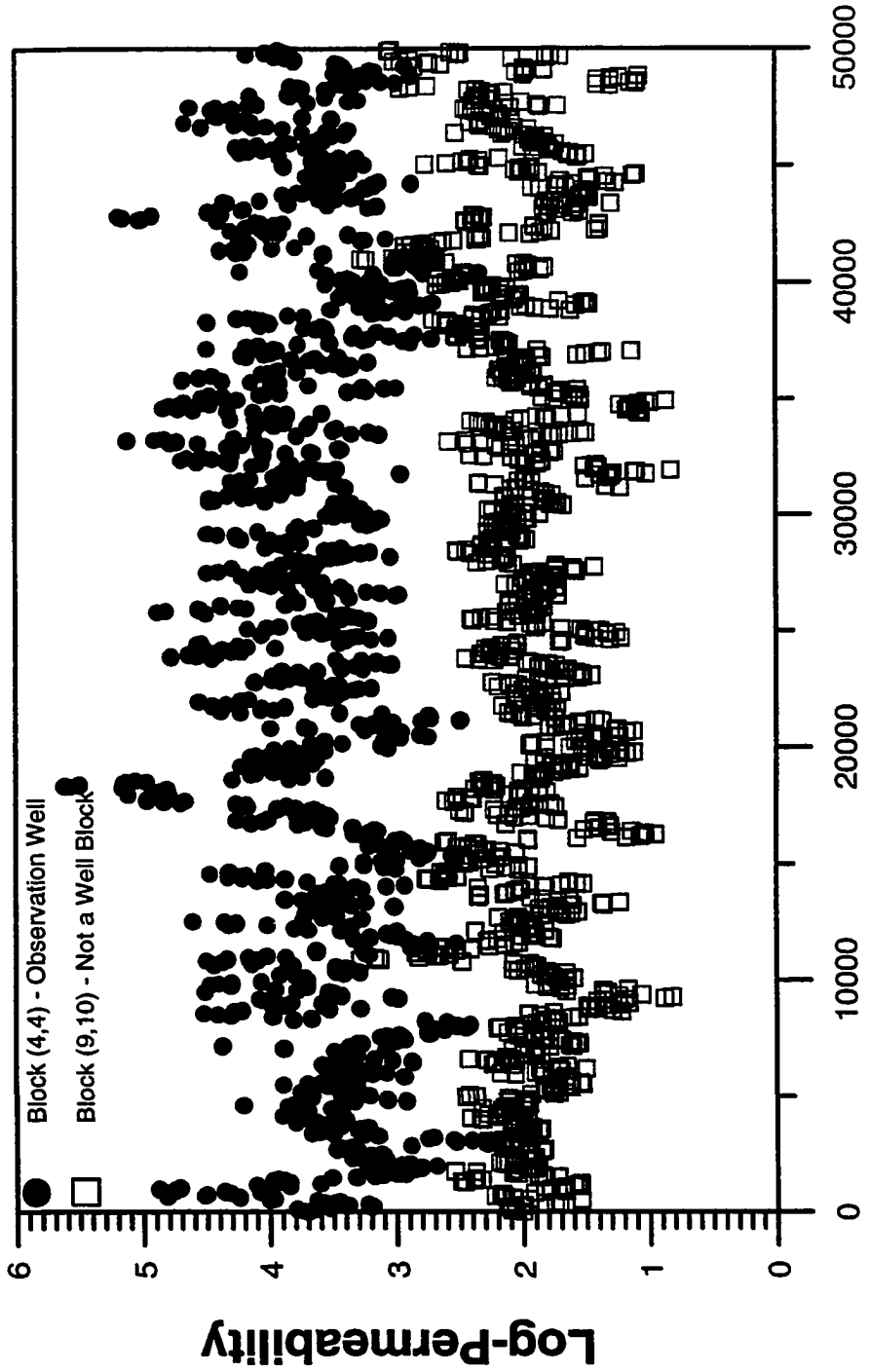

Fig. 3.17a - Log-Permeability Values During Sampling with Local Perturbations Based on Sensitivities Coefficients with Simulated Annealing to Initialize the Markov Chain (Variance = 1.0).

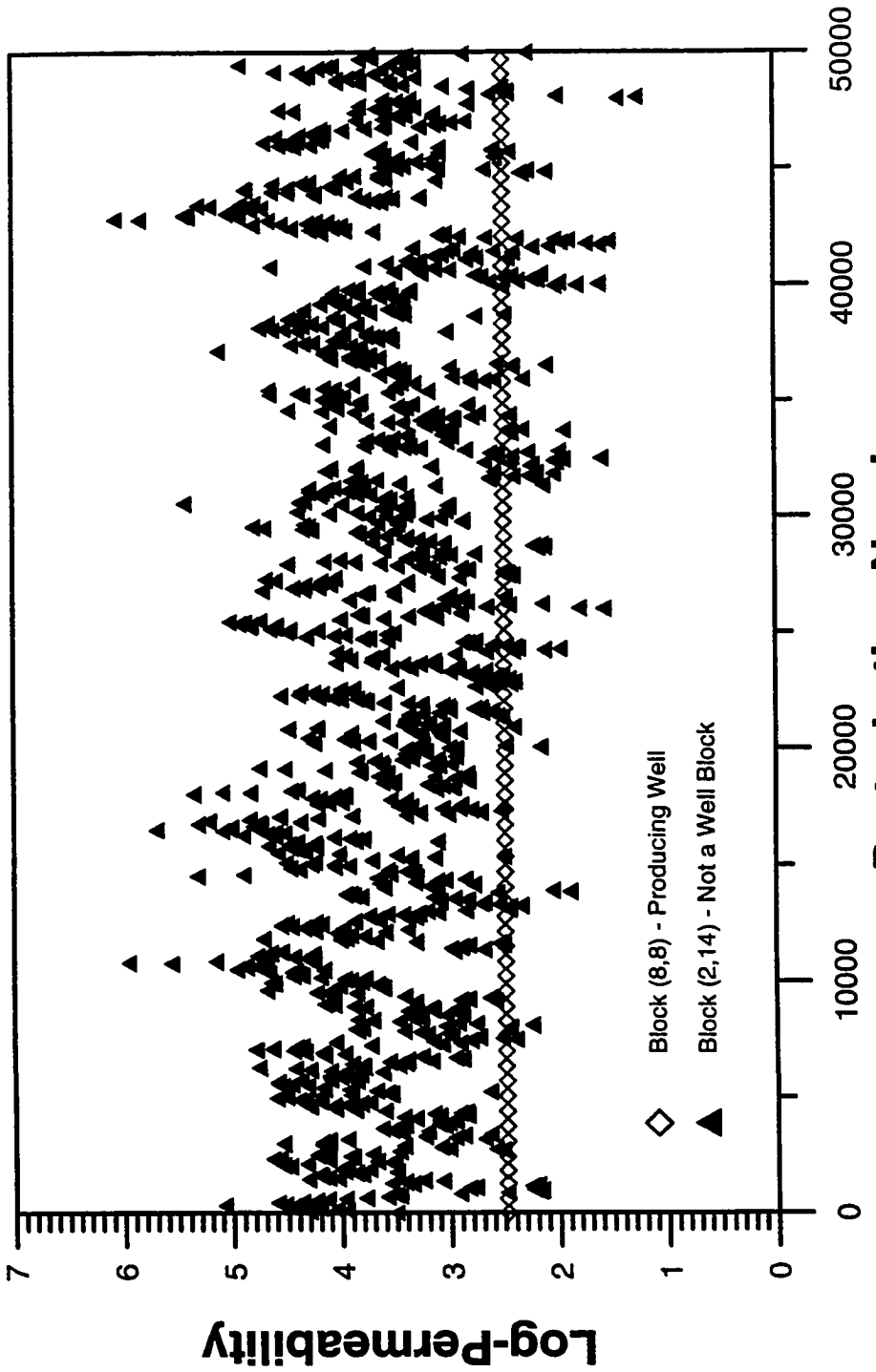

Fig. 3.17b - Log-Permeability Values During Sampling with Local Perturbations Based on Sensitivities Coefficients with Simulated Annealing to Initialize the Markov Chain (Variance = 1.0).

the a posteriori covariance matrix with simulated annealing to initialize the Markov chain Monte Carlo method.

With respect to the number of images obtained, recall from Chapter II, that the Markov chain Monte Carlo method that used global perturbations based on the a posteriori covariance matrix generated the total number of 8 different, independent realizations. It is important to notice that rejected transitions result in acceptable images (the image from the last transition accepted), but for global perturbations the number of different images may be small. Using local perturbations based on the a posteriori covariance matrix we obtained approximately 25300 distinct, but highly correlated images. With the newly developed method that uses simulated annealing 19000 different, but highly correlated images were obtained. As mentioned before, in the case of local perturbations it is necessary to thin the set of accepted images. This is performed by means of calculating the experimental variogram of the objective function for the accepted states.

Fig. 3.18 shows the experimental variogram obtained. As we can conclude from Fig. 3.18, realizations generated from local perturbations are slightly correlated to lags of approximately 1000. If, we retain only every 1000th realizations we would be left with 19 independent realizations. In Chapter II, for this same example, the Markov chain Monte Carlo method that uses local perturbations based on the a posteriori covariance matrix generated, after the thinning of images, 14 independent images.

Regarding computational time, each of the four methods required approximately 13 hours to attempt 50000 perturbations during the sampling phase. However, the fourth method had an overhead of 8 hours to perform the 30000 iterations necessary to calculate an initial state for the start of Markov chain sampling period.

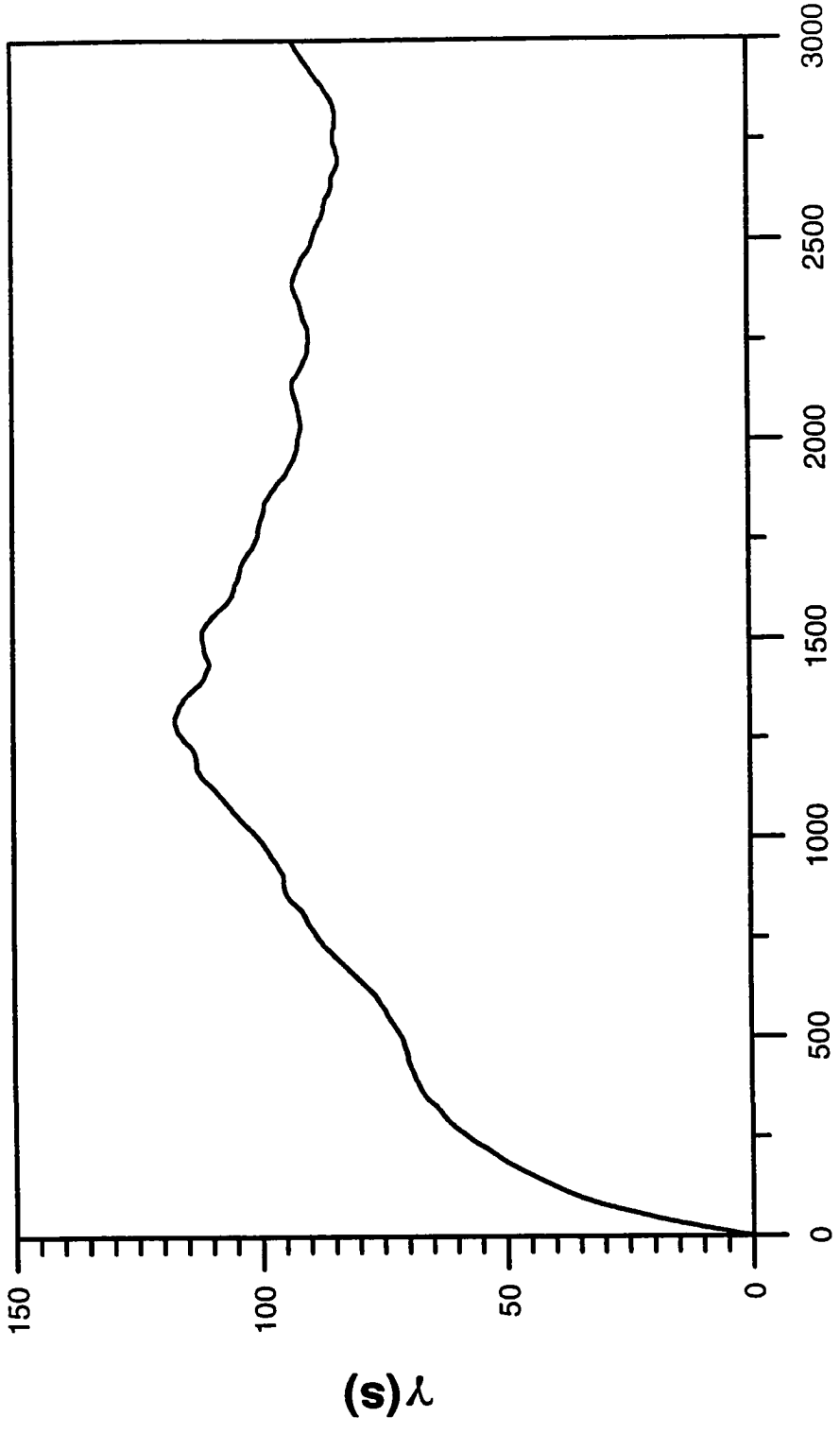

Fig. 3.18 - Experimental Variogram for the Objective Function of the Chain of Realizations Generated Using Local Perturbations Based on the a Posteriori Covariance Matrix with Simulated Annealing to Initialize the Markov Chain (Variance = 1.0).

CHAPTER IV

THE HYBRID MONTE CARLO METHOD

As described in Chapter II the most useful technique for generating a sequence of configurations with the desired distribution is to construct a Markov chain. A Markov chain process is a stochastic procedure which, given state m^i generates a new configuration m^j with transition probability p_{ij}. Any Markov chain will converge to a unique fixed point probability distribution π, i.e., is ergodic, provided it is irreducible (for all (i,j) there exists n such that $p^n_{ij} > 0$), $p_{i,i} > 0$ for some i and it satisfies detailed balance

$$\pi(m^i)\,p_{ij} = \pi(m^j)\,p_{ji}\,. \tag{4. 1}$$

As shown previously, it is convenient to construct a Markov chain process by two steps. First we propose a new configuration m^j with some probability q_{ij} by some yet unspecified procedure, and then we accept m^j with some probability α_{ij} or reject it and keep the old configuration m^i instead. One choice of α_{ij} which enables the detailed balance to be satisfied for any q_{ij} is a simple generalization[30] of the Metropolis algorithm[5]

$$\alpha_{ij} = \min\left\{1, \frac{\pi_j q_{ji}}{\pi_i q_{ij}}\right\}. \tag{4. 2}$$

As in Chapter II, the focus of our efforts here will be on improving the efficiency of the overall method by increasing the acceptance probability for transitions proposed with

probability q_{ij}. Since π_i is a fixed function of the permeability distribution, we can only improve the efficiency by improving the process of proposing transitions. We require a method for choosing candidate configurations that can be computed efficiently and that minimizes the correlation between accepted configurations.

In Chapter II, we enhanced the process of proposing transitions by proposing perturbations of a probability density function that approximate the true probability density function. Despite the improvements achieved, we believe, based on computational examples considered, that this algorithm decreases in performance and will be very slow when a high variability in the permeability field is present and the problem is highly nonlinear. For such problems, in order to achieve a reasonable acceptance probability, the size of the changes must be very small or local, and consequently a large number of iterations are required to reach and explore the stationary distribution because the images will be highly correlated. Furthermore, recalculating the output of pressure data after changing a single component is as expensive as calculating the output after changing all components. Generating a candidate state by randomly perturbing all components at once eliminates the correlation between accepted states, but is unlikely to be accepted.

These disadvantages of the "standard" Markov chain Monte Carlo method motivated the investigation of the hybrid Monte Carlo method by Duane, Kennedy, Pendleton, and Roweth[7]. The hybrid Monte Carlo method is an elaboration of the stochastic dynamics method[37], which merges the Metropolis algorithm with perturbations techniques based on dynamical simulation. These perturbations use gradient information provided by sensivity coefficients from a flow simulator to find candidate directions in which changes have a high probability of being accepted. The output of the algorithm is a

sample of points drawn from some specified distribution. For our problem, we wish to sample from the a posteriori probability density function, which incorporates multiwell pressure data, the prior mean and the model variogram. This class of methods is believed to perform better than simple Markov chain Monte Carlo methods, due to its avoidance of random walk behavior. The hybrid Monte Carlo algorithm itself, and methods related to it, have been reviewed by Toussaint[38] (1989), Kennedy[39](1990) and Neal[17] (1993).

4.1 The Stochastic Dynamics Method

The hybrid Monte Carlo method can be viewed as an extension of the stochastic dynamics method. This class of algorithms is founded in concepts from statistical physics and is usually expressed in terms of sampling from the canonical distribution for the state of a physical system, which is defined in terms of an energy function. Therefore, it is convenient to retain the physical terminology, even in non-physical contexts, by formulating the problem in terms of an energy function for a fictitious physical system.

Accordingly, suppose we wish to generate an ensemble of configurations of a set of "position" variables, q , which has n real-valued components, q_i distributed according to

$$P(q) = \frac{1}{w_E}\exp(-E(q)) \tag{4. 3}$$

where $E(q)$ is the "potential energy" function.

In a real physical system, q would consist of the coordinates of all particles; in our application, q will be the set of log-permeability values.

Note that formerly, a Markov chain Monte Carlo method was presented and used to sample from the probability distribution $P(q)$, where in the problem of interest to us, the log-permeability field plays the role of q, and the potential energy function is given by Eq. 1.6. Here, aiming to improve the sampling procedure, especially for higher variance cases, we use the hybrid Monte Carlo algorithm. This algorithm incorporates dynamical methods within the general framework of the Metropolis algorithm.

We wish to introduce a trick which allows large steps through configuration space, in order to produce independent configurations, while maintaining a large acceptance rate. We enlarge the space of field configurations by adding a "fictitious momentum" vector variable, p, which has n real-valued components, p_i, in one-to-one correspondence with the components of q. For a molecular dynamics problem, the p_i is the components of the momentum for the various particles. For other problems, the p_i's are introduced exclusively to allow the problem to be given a dynamical formulation.

The canonical distribution over the "phase space" of q and p together is defined to be

$$P(q,p) = \frac{1}{w_H}\exp(-H(q,p)) \tag{4.4}$$

where $H(q,p) = E(q) + K(p)$ is the "Hamiltonian" function, which gives the total energy. $K(p)$ is the "kinetic energy" due to the momentum, for which the usual choice is

$$K(p) = \sum_{i=1}^{n} \frac{p_i^2}{2m_i}. \tag{4.5}$$

The m_i are the "masses" associated with each component. As discussed later, adjustment of these mass values can improve efficiency, but for the moment they can all be taken to be equal to one.

The canonical distribution for the set of p_i variables is

$$P(p) = \frac{1}{w_K}\exp(-K(p)) = (2\pi)^{-n/2}\exp(-\frac{1}{2}\sum_i p_i^2); \qquad (4.6)$$

i.e., the p_i's are independent, and have Gaussian distribution with zero mean and unit variance.

In the distribution of Eq. 4.4, q and p are independent, and the marginal distribution for q is the same as that of Eq. 4.3, from which we wish to sample. This can be concluded by writing the canonical distribution over phase space defined by the total energy function as

$$P(q,p) = \frac{1}{w_H}\exp(-H(q,p))$$

$$= \left[\frac{1}{w_E}\exp(-E(q))\right]\left[\frac{1}{w_K}\exp(-K(p))\right]$$

$$= P(q)P(p). \qquad (4.7)$$

Therefore we can just as well define a Markov chain that converges to the probability distribution for the extended system for q and p, and then simply ignore the p values when estimating expectations of functions of q.

In the stochastic dynamics method, the Markov chain that gives samples from the probability distribution for q and p given by Eq. 4.4 is generated using two types of

transitions. The "dynamic" transitions which are deterministic, and are used to sample uniformly from values of q and p with a fixed total energy, $H(q,p)$. The "stochastic" transitions allow the algorithm to sample states with different values of H.

4.1.1 Hamiltonian Dynamics

The dynamic transitions sample at a fixed total energy by simulating the Hamiltonian dynamics of the system, in which the state evolves in fictitious time, τ, according to the following equations:

$$\frac{dq_i}{d\tau} = \frac{\partial H}{\partial p_i} = \frac{p_i}{m_i}, \tag{4.8}$$

$$\frac{dp_i}{d\tau} = -\frac{\partial H}{\partial q_i} = -\frac{\partial E}{\partial q_i}. \tag{4.9}$$

As pointed out by Refs. 12, 38 and 39, the motion defined by these equations has three important properties. First, the motion leaves the total energy H constant. Second, the motion is reversible - if following the dynamics for some period of time maps (p, q) to (p^*,q^*), then it also maps ($-p^*$,q^*) to ($-p$, q). Finally, the motion satisfies Liouville's theorem, which says that the volume element in phase space is conserved. This means that points, initially in some region of volume V moving according to the dynamical equations will end up after some given time interval at a region that also has volume V. This is important because for continuous variables we are really talking about probability densities and the "probability of a state configuration" really means the probability density times the volume element in phase space.

As shown in Refs. 38 and 39, due to the preservation of phase volume and the time reversibility properties, transitions based on Hamiltonian dynamics will generate a Markov chain such that the detailed balance of Eq. 4.1 is satisfied. However, due to the conservation of total energy property, such transitions are not sufficient to produce an ergodic Markov chain since regions with different values of H are never visited. As it will be described in subsection 4.1.3 the generation of an ergodic Markov chain is accomplished by alternating dynamic with stochastic transitions.

4.1.2 The Leapfrog Discretization

In reality, Hamiltonian dynamics cannot be simulated exactly, but can only be approximated by some discretization using finite time steps. The method that is commonly used is the leapfrog method. In this scheme a single iteration calculates approximations to the position and momentum, q and p, at time $\tau+\varepsilon$ from q and p at time τ as follows:

$$p_i\left(\tau+\frac{\varepsilon}{2}\right)=p_i(\tau)-\frac{\varepsilon}{2}\frac{\partial E}{\partial q_i}(q_i(\tau)), \tag{4. 10}$$

$$q_i(\tau+\varepsilon)=q_i(\tau)+\varepsilon\frac{p_i\left(\tau+\frac{\varepsilon}{2}\right)}{m_i}, \tag{4. 11}$$

$$p_i(\tau+\varepsilon)=p_i\left(\tau+\frac{\varepsilon}{2}\right)-\frac{\varepsilon}{2}\frac{\partial E}{\partial q_i}(q_i(\tau+\varepsilon)). \tag{4. 12}$$

This iteration consists of a half-step for the p_i , a full step for the q_i, and another half-step for the p_i. To follow the dynamics for some period of time, $\Delta\tau$, a value of ε that is thought

to be small enough to give an acceptable error is chosen, and equations 4.10-4.12 are applied for $L=\Delta\tau/\varepsilon$ iterations in order to reach the target time.

The preservation of phase space and the reversibility properties presented by the Hamiltonian dynamics remain true when using the leapfrog discretization[17,38,39]. However, the value of H no longer stays exactly constant. Because of this, Monte Carlo estimates found using the stochastic dynamics method will suffer from some systematic error, which will go to zero only as the stepsize, ε, is reduced to zero (with the number of steps needed to compute each trajectory then going to infinity.)

4.1.3 Stochastic Transitions

In the stochastic dynamics method, an ergodic Markov chain is obtained by alternately performing deterministic dynamical transitions and stochastic updates of the momentum. Since q and p are independent, p may be updated without reference to q by drawing a new value with probability density proportional to $\exp(-K(p))$. For the kinetic energy function of Eq. 4.5, this is easily done, since the p_i have independent Gaussian distributions. The p_i are sampled from a Gaussian distribution with zero mean and unit variance. These updates of p can change H, allowing the entire phase space to be explored.

In physical terms, a stochastic transition can be viewed as representing the effect on the system of contact with a heat reservoir. Such contact can be of varying strength, suggesting that we might also try controlling how large an effect the stochastic transitions have relative to that of the dynamical transitions. For stochastic transitions that completely

replace the p_i with newly chosen values, such control can be exercised by adjusting the duration of the dynamical trajectories simulated between stochastic transitions. In this case, the length in fictitious time of the trajectories is an adjustable parameter of the stochastic dynamics method and, as it is going to be shown in a later section, it is best to use trajectories that result in large changes to q. This avoids the random walk effects that would result from randomizing the momentum after each short trajectory.

An alternative, as described by Horowitz, is to simulate only short trajectories (perhaps just a single leapfrog iteration), but to use stochastic transitions of the following form

$$p_i' = \alpha p_i + (1-\alpha^2)^{1/2} n_i \qquad (4.\ 13)$$

where n_i is drawn from a Gaussian distribution with zero mean and unit variance, p_i are the momentum variables obtained at the end of a previous dynamical transition, p_i' are the transformed momentum variables needed to start the current dynamical transition and $0 \leq \alpha \leq 1$. Here, α is a parameter that controls the effect the of stochastic transitions.

Setting α to zero in Eq. 4.13 gives the previous method of just replacing the momenta. If this is done after every leapfrog iteration, there is a large random walk aspect to the motion, which is generally undesirable. When α is set to a value only slightly less than one, the momenta are only slightly altered in each stochastic transition, reducing the random walk effect. In this method, the number of leapfrog iterations is kept equal to one but it is necessary to have an estimate for the value of parameter α. A criterion to choose the value of this parameter when using a hybrid Monte Carlo method with the variant described by Horowitz to solve our problem are given in a later section.

4.2 Hybrid Monte Carlo

In the stochastic dynamics method, if the dynamics were simulated exactly, the total energy would be conserved and the change in H would always be zero. When the dynamics is simulated with a non-zero stepsize, H may change, causing some systematic errors in Monte Carlo estimates found by this method. The systematic errors coming from the simulation of the dynamics with a non-zero stepsize can be eliminated using a method developed by Duane *et al.*[7]

Like the stochastic dynamics method, the hybrid Monte Carlo algorithm samples points in phase space by means of a Markov chain in which stochastic and dynamical transitions alternate. In the stochastic transitions, the momentum is replaced by resampling the "momentum" variables p_i from a Gaussian distribution with mean zero and variance equal to one. The dynamical transitions in the hybrid Monte Carlo method are also similar to those in the stochastic dynamics method, but with a change - the point reached by following the dynamics is only a candidate for the new state, to be accepted or rejected based on the change in total energy, as in the Metropolis algorithm. If the dynamics were simulated exactly, the change in H would always be zero, and the new point would always be accepted. When the dynamics is simulated using some approximate discretization, H may change, and moves will occasionally be rejected. These rejections exactly eliminate the bias introduced by the inexact simulation[17].

In detail, given values for the magnitude of the leapfrog stepsize, ε, and the number of leapfrog iterations, L, a dynamical transition consists of the following steps:

Starting from the current state, $(q,p)=(q(t_0),p(t_0))$, perform L leapfrog iterations with a stepsize of ε, resulting in the state $(q(\varepsilon L),p(\varepsilon L))=(q^*,p^*)$.

Regard (q^*,p^*) as a candidate for the next state, as in the Metropolis algorithm, accepting it with probability

$$\min(1,\exp\{-[H(q^*,p^*)-H(q(t_0),p(t_0))]\}),$$

and otherwise letting the new state be the same as the old state.

As pointed out by Duane *et al.*, it is not necessary for the Hamiltonian used in simulating the dynamics to be the same as that which defines the canonical distribution from which we wish to sample. The algorithm remains valid (i.e. it produces an ergodic Markov chain) as long as the correct Hamiltonian is used when deciding whether to accept a candidate state. Of course, generating candidate states by following trajectories based on a drastically wrong Hamiltonian will lead to a very low acceptance rate.

The name Langevin Monte Carlo is given to hybrid Monte Carlo with $L=1$, that is, in which candidates states are generated using only a single leapfrog iteration. Only when L is reasonably large, however, does one obtain the principal benefit of hybrid Monte Carlo - the avoidance of random walks. One might think that a large error in H would develop over a long trajectory, leading to a very low acceptance rate. As described later, for sufficiently small stepsizes, this usually does not occur. Instead, the value of H oscillates along the trajectory, and the acceptance rate is almost independent of trajectory length.

4.3 Analysis of the Hybrid Monte Carlo Algorithm

In this section we concentrate in investigating the behavior of the hybrid Monte Carlo algorithm when sampling from a Gaussian distribution. Typically, the hybrid Monte Carlo algorithm is applied to more complex problems, but performance on Gaussian distributions may be indicative of general performance and give a better understanding of some of the specific features of the method.

4.3.1 The Hybrid Monte Carlo Algorithm to Sample a Univariate Gaussian Distribution

A univariate Gaussian distribution with mean zero and variance σ^2 can be represented as the marginal distribution for a "position" variable q with respect to the canonical distribution given by the following Hamiltonian:

$$H(q,p) = \frac{1}{2\sigma^2}q^2 + \frac{1}{2}p^2 . \qquad (4.\ 14)$$

A single iteration of the leapfrog method applied with this Hamiltonian produces a linear mapping from $(q(\tau),p(\tau))$ to $(q(\tau+\varepsilon),p(\tau+\varepsilon))$ that can be written as follows:

$$p\left(\tau+\frac{\varepsilon}{2}\right) = p(\tau) - \frac{\varepsilon}{2}\frac{q(\tau)}{\sigma^2}, \qquad (4.\ 15)$$

$$q(\tau+\varepsilon) = q(\tau) + \varepsilon p\left(\tau+\frac{\varepsilon}{2}\right), \qquad (4.\ 16)$$

$$p(\tau+\varepsilon)=p\left(\tau+\frac{\varepsilon}{2}\right)-\frac{\varepsilon}{2}\frac{q(\tau+\varepsilon)}{\sigma^2}, \qquad (4.17)$$

or in matrix form as

$$\begin{bmatrix} q(\tau+\varepsilon) \\ p(\tau+\varepsilon) \end{bmatrix} = \begin{bmatrix} 1-\varepsilon^2/2\sigma^2 & \varepsilon \\ -(\varepsilon/\sigma^2)(1-\varepsilon^2/4\sigma^2) & 1-\varepsilon^2/2\sigma^2 \end{bmatrix} \cdot \begin{bmatrix} q(\tau) \\ p(\tau) \end{bmatrix}. \qquad (4.18)$$

As described in Ref. 12, $H(q,p)$ diverges, that is to say that the numerical solution of the difference equation is unstable, if this leapfrog iteration is repeatedly applied with $\varepsilon \geq 2\sigma$, but H remains bounded when it is applied with $\varepsilon < 2\sigma$.

Several conclusions may be drawn from this analysis. First, when the hybrid Monte Carlo method is used with long trajectories, consisting of many leapfrog iterations, it is essential that the stepsize be small enough to ensure stability, as otherwise the value of H will rapidly diverge, resulting in a increase of the rejection rate. Second, as long as the stepsize is in the stable region, the error in H, at least for this example, does not grow larger and larger as the number of leapfrog iterations is increased.

Figure 4.1 shows the error in energy for trajectories computed with different stepsizes, ε which were defined as $\varepsilon = \eta.\sigma$ where, η is referred to as the stepsize adjustment factor and for this univariate example, $\sigma = 1$. Each point plotted in the figure shows the change in total energy (H) for a trajectory of fifty leapfrog iterations in which the stepsizes were adjusted by the factor in the horizontal axis. Changes in the total energy greater than ten were plotted at ten. Note that, as long as $\varepsilon < 2\sigma$, the solution is stable. This result was the same for several trajectory lengths tested reflecting the insensibility of the stability criterion to the number of the leapfrog iterations. As we can observe, for

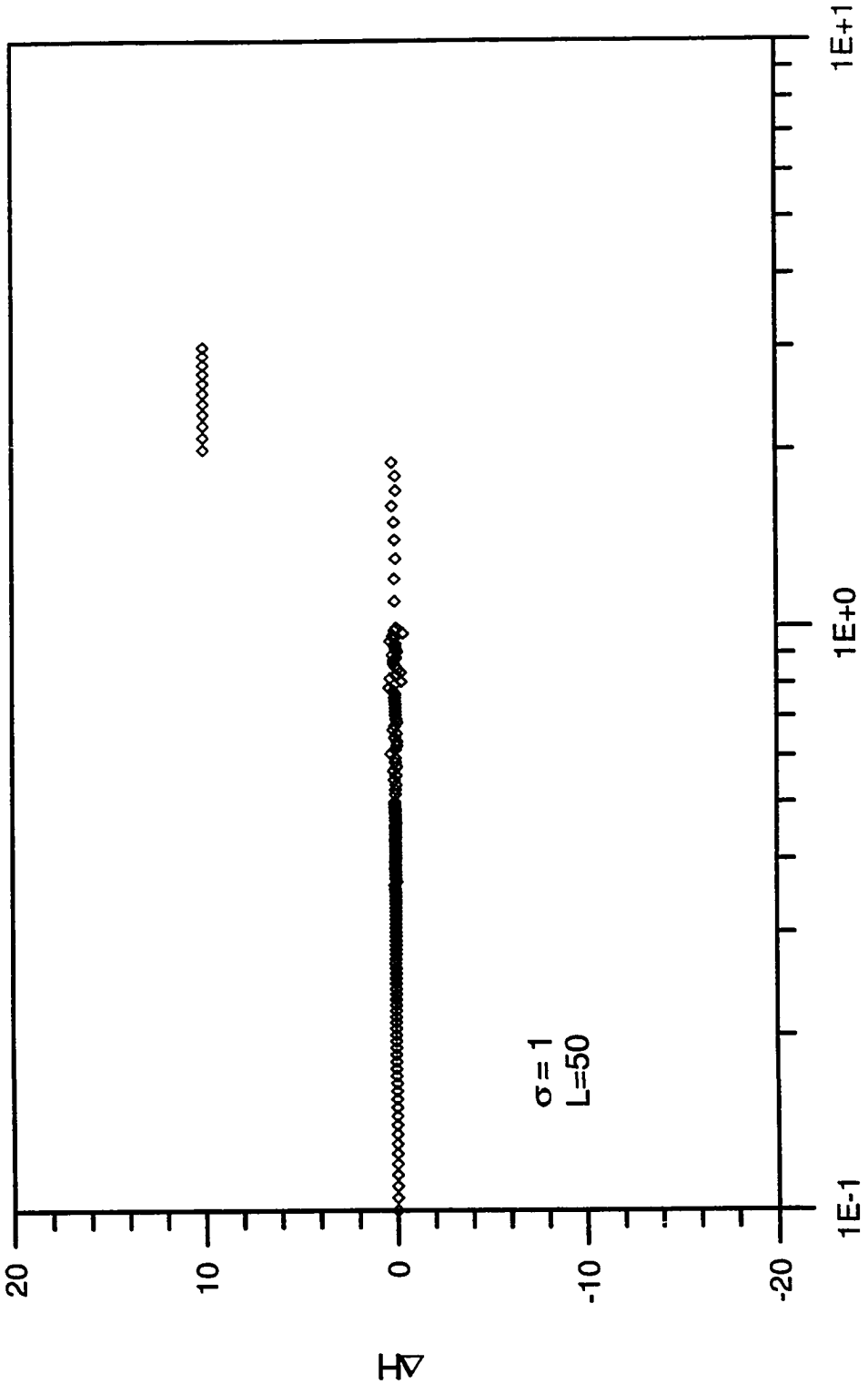

Fig. 4.1 - Errors in a Hamiltonian Function when Sampling a Univariate Gaussian Distribution.

stepsizes past the limit of stability, the leapfrog discretization becomes unstable, which will result in an acceptance rate close to zero. The optimal strategy is usually to select a stepsize just a bit below the point of instability.

4.3.2 The Hybrid Monte Carlo Algorithm to Sample a Multivariate Gaussian Distribution

The system studied before, with the Hamiltonian of Eq. 4.14, can be generalized to a system with several position variables, with the following Hamiltonian:

$$H(q,p) = \sum_i \frac{q_i^2}{2\sigma_i^2} + \sum_i \frac{p_i^2}{2m_i}. \qquad (4.19)$$

In the canonical distribution, this defines the marginal distribution for q as a multivariate Gaussian in which the components are independent, with component q_i having mean zero and standard deviation σ_i. When the leapfrog method is used with this Hamiltonian, there is no interaction between the different pairs of position and momentum variables. Each such pair, (q_i,p_i), evolves independently of the other pairs, since the value of $\partial E/\partial q_i$ does not depend on q_j for $j \neq i$. Once the end of the trajectory is reached, however, a decision to accept or reject the final state is made based on the total error in H, which is the sum of the errors due to the inaccuracies in the simulation with respect to each pair of variables. A large error resulting from inaccurate simulation with respect to any of these coordinate pairs will likely lead to the trajectory being rejected. If the proposed state is rejected then the new state is the old state.

If we set $m_i = 1$ for all i and use the same stepsize for all components in this system, then to keep the rejection rate low, we will have to use a stepsize less than $2\sigma_{min}$, where

σ_{min} is the smallest of the σ_i, as otherwise the error in H due to one or more components will diverge as the trajectory length increases. If the other σ_i are much larger than σ_{min}, then with this small stepsize a large number of leapfrog steps will be required before these other components change significantly.

This problem can be better illustrated if we construct the leapfrog iteration, with m_i = 1, for the Hamiltonian of Eq. 4.19. In this case we can write

$$p_i(\tau+\frac{\varepsilon}{2}) = p_i(\tau) - \frac{\varepsilon}{2}\frac{q_i(\tau)}{\sigma_i^2}, \tag{4.20}$$

$$q_i(\tau+\varepsilon) = \varepsilon p_i(\tau) + \left(1-\frac{\varepsilon^2}{2\sigma_i^2}\right)q_i(\tau), \tag{4.21}$$

$$p_i(\tau+\varepsilon) = \left(1-\frac{\varepsilon^2}{2\sigma_i^2}\right)p_i(\tau) - \frac{\varepsilon}{\sigma_i^2}\left(1-\frac{\varepsilon^2}{4\sigma_i^2}\right)q_i(\tau). \tag{4.22}$$

Let us analyze the effect in sampling components q_i. To do this, suppose we have a distribution with two components only, i.e. i = 1,2 and $p_i(\tau)$ = 0 at the beginning of the leapfrog step and that $\sigma_1 << \sigma_2$. Using this in Eq. 4.21 we can write

$$q_1(\tau+\varepsilon) = \left(1-\frac{\varepsilon^2}{2\sigma_1^2}\right)q_1(\tau), \tag{4.23}$$

$$q_2(\tau+\varepsilon) = \left(1-\frac{\varepsilon^2}{2\sigma_2^2}\right)q_2(\tau). \tag{4.24}$$

If we use the same stepsize for all components in this system, then to keep the rejection rate low, we will have to use a stepsize less than $2\sigma_1$, as otherwise the error in H

due to one or more components will diverge as the trajectory length increases. If we use $\varepsilon = \sigma_1/2$ for example, Eqs. 4.23 and 4.24 will give that

$$q_1(\tau+\varepsilon) = \frac{7}{8} q_1(\tau), \tag{4. 25}$$

$$q_2(\tau+\varepsilon) = \left(1 - \frac{\sigma_1^{\,2}}{2\sigma_2^2}\right) q_2(\tau). \tag{4. 26}$$

Since $\sigma_1 << \sigma_2$ Eqs. 4.25 and 4.26 show that in a leapfrog step, component q_2 will be essentially unchanged, i.e., leapfrog iterations will not sample the direction of the higher variance component well.

As mentioned before the adjustment of "masses" values, m_i, associated with each component can avoid this problem and improve the method efficiency. A basis for choosing optimal values for m_i can be found by examining the behavior of a leapfrog method applied to a system with the Hamiltonian described by Eq. 4.19. When incorporating the masses, m_i the leapfrog method applied to this system gives

$$q_i(\tau+\varepsilon) = \frac{\varepsilon}{m_i} p_i(\tau) + \left(1 - \frac{\varepsilon^2}{2m_i\sigma_i^2}\right) q_i(\tau), \tag{4. 27}$$

$$p_i(\tau+\varepsilon) = \left(1 - \frac{\varepsilon^2}{2m_i\sigma_i^2}\right) p_i(\tau) - \frac{\varepsilon}{\sigma_i^2}\left(1 - \frac{\varepsilon^2}{4m_i\sigma_i^2}\right) q_i(\tau). \tag{4. 28}$$

Let us analyze the effect in sampling of components q_i. To do this we will consider a distribution with only two components, i.e. $i = 1,2$ and that $\sigma_{min} = \sigma_1 < \sigma_2$ and $p_i(\tau) = 0$ at the beginning of one leapfrog step. For stability purposes we know that $\varepsilon < 2\sigma_1$. Defining $m_i = \sigma^2_{min}/\sigma^2_i = \sigma^2_1/\sigma^2_i$ and assuming that $\varepsilon = \sigma_1$ we have for each component that

$q_1(\tau+\varepsilon) = 0.5q_1(\tau)$ and $q_2(\tau+\varepsilon) = 0.5q_2(\tau)$. This final result shows that when considering the mass terms, all the directions are investigated adequately.

This means that if the "mass" terms are considered, and the leapfrog scheme as described in Eqs. 4.20-4.22, is applied to the Hamiltonian of Eq. 4.19 the sampling process will be optimized if $m_i = \sigma^2_{min}/\sigma^2_i$ and $\varepsilon < 2\sigma_{min}$. Note that in this case the stochastic transitions will have to be made by sampling the momentum variables p_i from a Gaussian distribution with mean zero and variance m_i.

One way to make the computation more efficient is to recast the choice of masses, m_i, and stepsize, ε, as a choice of individual stepsizes, ε_i, that are applied when updating each component of the position and momentum. To better explain this, it is convenient to rewrite the leapfrog method of equations as follows:

$$\frac{p_i\left(\tau+\frac{\varepsilon}{2}\right)}{\sqrt{m_i}} = \frac{p_i(\tau)}{\sqrt{m_i}} - \frac{\varepsilon}{2\sqrt{m_i}}\frac{\partial E}{\partial q_i}(q_i(\tau)), \tag{4.29}$$

$$q_i(\tau+\varepsilon) = q_i(\tau) + \frac{\varepsilon}{\sqrt{m_i}}\frac{p_i\left(\tau+\frac{\varepsilon}{2}\right)}{\sqrt{m_i}}, \tag{4.30}$$

$$\frac{p_i(\tau+\varepsilon)}{\sqrt{m_i}} = \frac{p_i\left(\tau+\frac{\varepsilon}{2}\right)}{\sqrt{m_i}} - \frac{\varepsilon}{2\sqrt{m_i}}\frac{\partial E}{\partial q_i}(q_i(\tau+\varepsilon)). \tag{4.31}$$

Rather than applying the leapfrog equations to update p_i and q_i, we can therefore store the values $p_i/\sqrt{m_i}$ instead of the p_i, and update these values (along with the q_i) using

leapfrog steps in which different components have different stepsizes, given by $\varepsilon_i = \varepsilon / \sqrt{m_i}$. In this case we can define a variable

$$\hat{p}_i = \frac{p_i}{\sqrt{m_i}}, \tag{4. 32}$$

and the leapfrog equations can be rewritten as follows:

$$\hat{p}_i\left(\tau + \frac{\varepsilon}{2}\right) = \hat{p}_i(\tau) - \frac{\varepsilon_i}{2}\frac{\partial E}{\partial q_i}(q_i(\tau)), \tag{4. 33}$$

$$q_i(\tau + \varepsilon) = q_i(\tau) + \varepsilon_i \hat{p}_i\left(\tau + \frac{\varepsilon}{2}\right), \tag{4. 34}$$

$$\hat{p}_i(\tau + \varepsilon) = \hat{p}_i\left(\tau + \frac{\varepsilon}{2}\right) - \frac{\varepsilon_i}{2}\frac{\partial E}{\partial q_i}(q_i(\tau + \varepsilon)). \tag{4. 35}$$

As seen before, for stability purposes we require $\varepsilon < 2\sigma_{min}$ and an appropriate choice for the i^{th} mass is $m_i = \sigma^2_{min} / \sigma^2_i$. Consequently $\varepsilon_i < 2\sigma_i$ or for the ith component, we can set the stepsize, ε_i, to a value a bit less than $2\sigma_i$, with the result that even short trajectories transverse the full range of all components.

Previously the stochastic transitions had to be made by sampling the momentum variables p_i from a Gaussian distribution with mean zero and variance m_i. With this re-expression of the leapfrog method the canonical distribution of $p_i / \sqrt{m_i}$ is independent of m_i and the stochastics transitions are performed by sampling $\hat{p}_i$ from a Gaussian distribution with zero mean and variance equal to one. This reduces slightly the amount of computation required since its not necessary to convert a standard normal variable to a Gaussian distribution with mean zero and variance m_i.

Fig. 4.2 illustrates the operation of the leapfrog method on a bivariate Gaussian distribution and the effect of the use of the mass term. In this example, $\sigma_1 = \sigma_{min} = 0.1$ and $\sigma_2 = 1.5$, the stepsize adjustment factor used was $\eta = 0.1$ and a single trajectory consisting of ten leapfrog iterations is shown. In Fig. 4.2a, the mass term m_i is considered equal to one; i.e. we use the same stepsize for all components in the system, and to ensure stability the stepsizes used were $\varepsilon_1 = \varepsilon_2 = \eta\sigma_{min} = \eta\sigma_1$. Previously (see Eqs. 4.20-4.26), we showed that if any of the σ_i are much larger than σ_{min}, the need to limit the stepsize to $\eta\sigma_{min}$ in all the directions slows the rate at which the algorithm can explore the distribution in directions corresponding to components with a higher standard deviation. This fact is reflected in Fig. 4.2a, in which the direction of the second component (the one associated with the higher standard deviation) does not change significatively during the ten leapfrog. In Fig. 4.2b, the mass term component is taken into account by using different stepsizes for each direction. In this case the stepsizes used were $\varepsilon_1 = \eta\sigma_1$ for the first component and $\varepsilon_2 = \eta\sigma_2$ for the second component. As described previously (see Eqs. 4.27 and 4.28) and as shown in Fig. 4.2b, the use of a different stepsize for each component (equivalently a different mass) improves the sampling efficiency. Here, for the same number of leapfrog steps given in the example described in Fig. 4.2a, the component with the higher standard deviation (the second component) changes significantly during the dynamical simulation steps.

Figure 4.3 shows the error in energy for trajectories computed with different stepsizes, ε_i, where $\varepsilon_i = \eta\sigma_i$. Each point plotted in the figure shows the change in total energy (H) for a trajectory of fifty leapfrog iterations for the same bivariate example ($\sigma_1 = 0.1$ and $\sigma_2 = 1.5$) for a given value of the stepsize adjustment factor, η. Changes in the

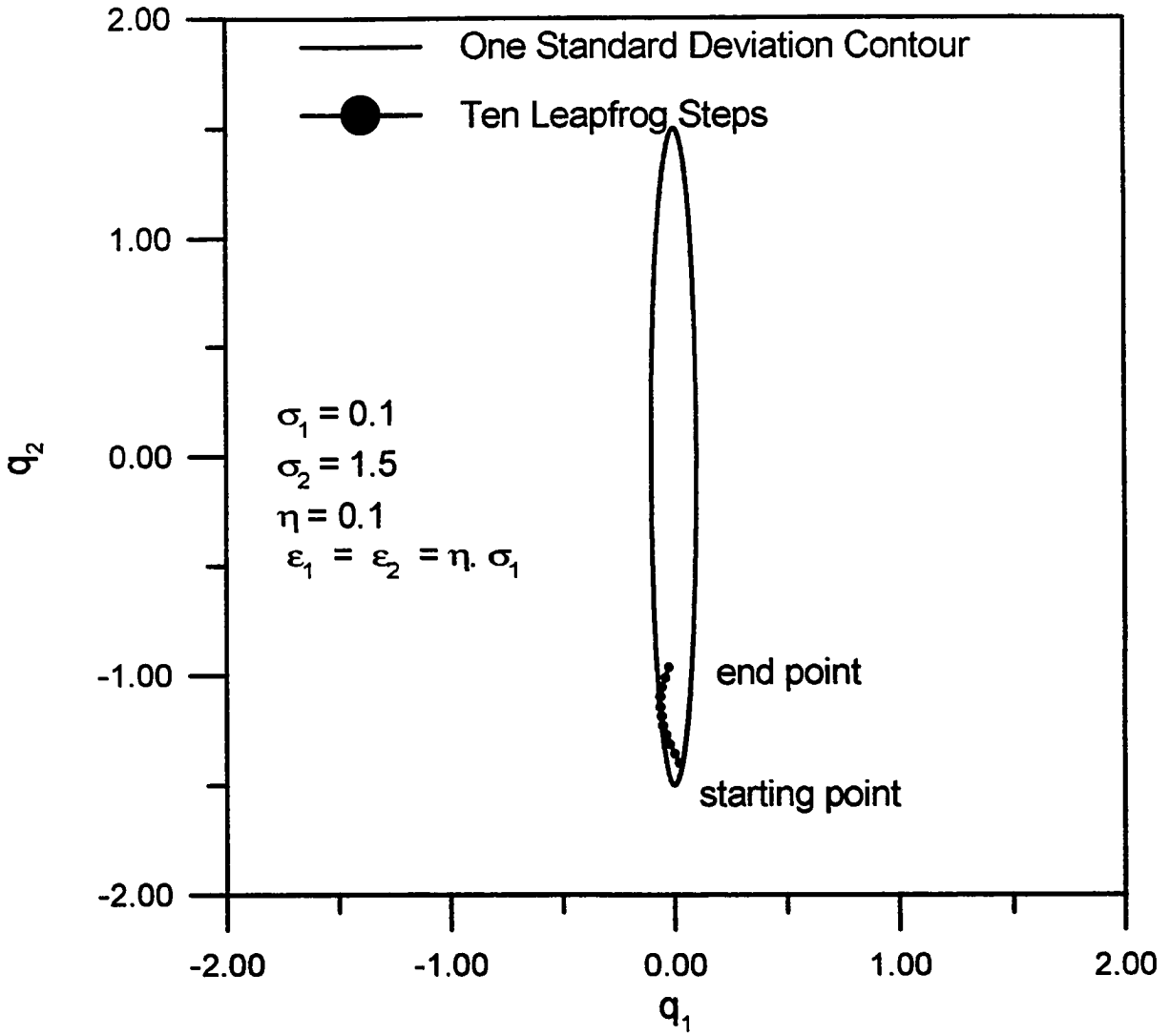

Fig. 4.2a - Behavior During One Dynamical Transition when Sampling a Bivariate Gaussian Distribution : no Mass Term.

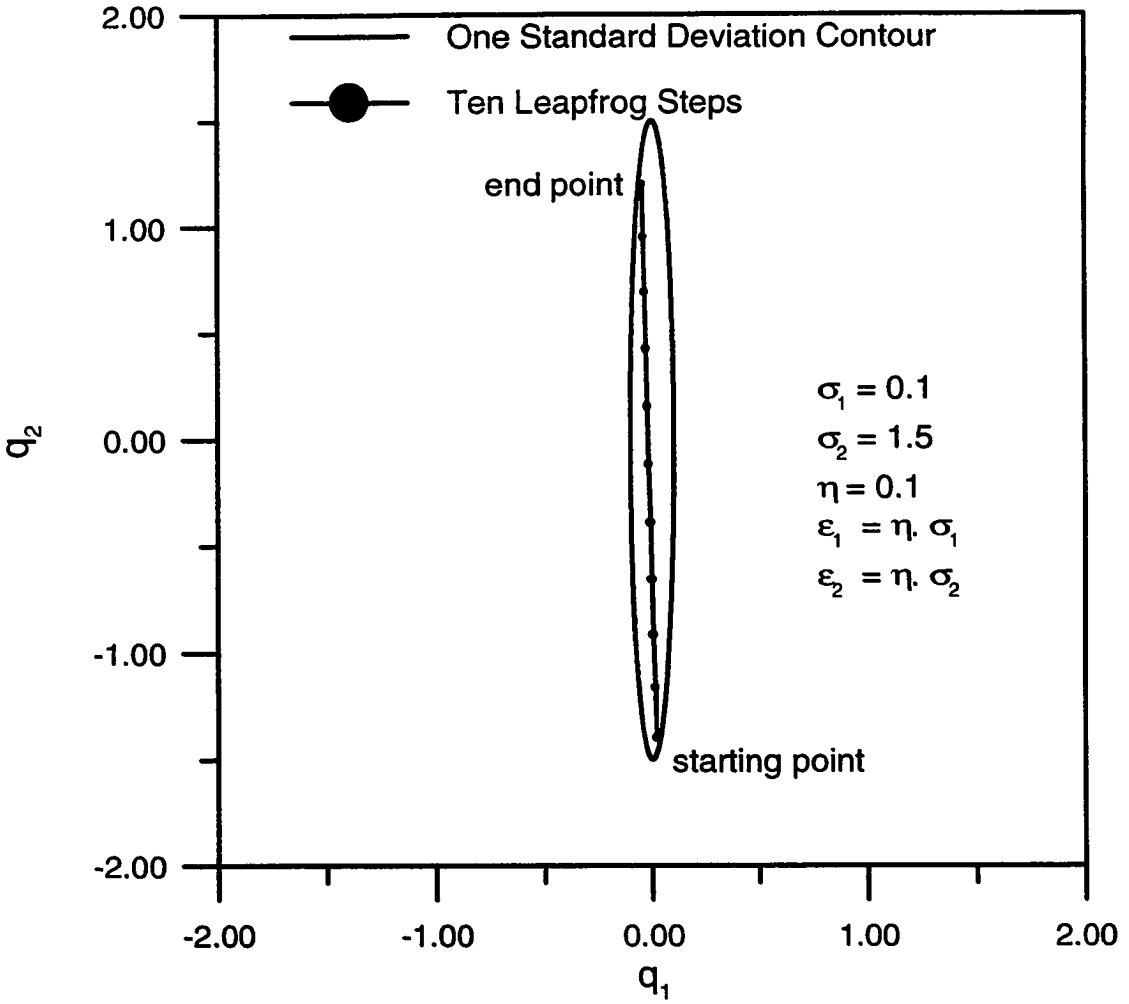

Fig. 4.2b - Behavior During One Dynamical Transition when Sampling a Bivariate Gaussian Distribution : with Mass Term.

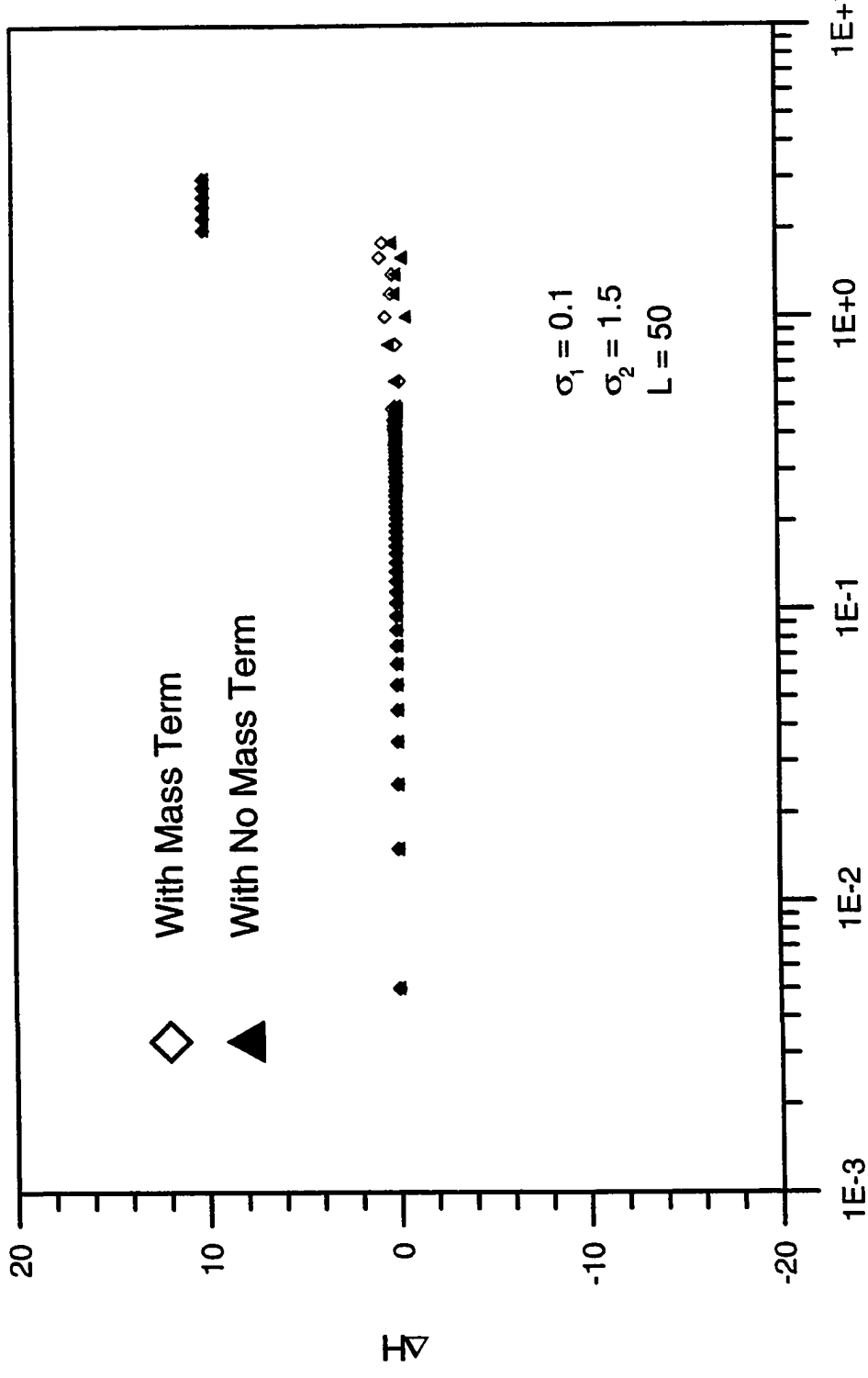

Fig. 4.3 - Errors in a Trajectory when Sampling a Bivariate Gaussian Distribution : Effect of Mass Term.

total energy greater than ten were plotted at ten. As we can observe, for stepsizes past the limit of stability, the leapfrog discretization becomes unstable, which will result in an acceptance rate very close to zero. The optimal strategy is usually to select a stepsize just a bit below this point of instability.

This simple system serves as an approximate model for the behavior of the leapfrog method when applied to a more complex system whose potential energy can locally be approximated by a quadratic function of q. By a suitable translation and rotation of coordinates such a quadratic energy function can be put in the form

$$E(q) = \sum_i q_i^2 / 2\sigma_i^2 . \tag{4.36}$$

If the energy can be written as in Eq. 4.36 then, as described by the previous bivariate example a good choice for the stepsize is to use different stepsizes, ε_i, defined by $\varepsilon_i = \eta\sigma_i$. As illustrated in Fig. 4.3 this criterion will give the same stability result encountered when sampling a univariate Gaussian distribution and will be optimum when using $\eta \approx 2$, i.e. each component will be sampled efficiently and a high acceptance rate will be obtained during the sampling process.

In practice, as shown later, it is not possible to take full advantage of the straightforward stepsize calculation criteria for a multivariate Gaussian, both because the potential energy is at best only approximately quadratic, and because we do not know how to translate and rotate the coordinate system so as to remove the interactions between components of q. When the different components interact, the stepsizes ε_i, defined by $\varepsilon_i = \eta\sigma_i$ may be too large, and it is necessary to use a small value for η in order to achieve a high acceptance rate.

The hybrid Monte Carlo method requires, besides the estimate of an optimal ε_i, the determination of one tunable parameter - the length of the trajectory defined by the number of steps L during dynamical transitions (or the frequency in simulation time with which we resample the momenta variables.) Fig. 4.4 schematically illustrates the considerations in determining this parameter. In this bivariate example, σ_1=0.10, σ_2=1.5 and $\varepsilon_i = 0.1\sigma_i$. In Fig. 4.4a, we only perform one leapfrog step (L = 1) before resampling $\hat{p}_1$ and $\hat{p}_2$. Thus, the coordinates q_i do not change significantly during a single dynamical simulation (Eqs. 4.33-4.35). When we resample the momenta again, the q's will move in a different direction. The result is basically a random walk, and the root-mean-square distance moved in the coordinate space in L steps is proportional to $\sqrt{L}$ (see Refs. 12,17 and 38.) This fact in illustrated in Fig. 4.4a which shows the progress of ten Langevin Monte Carlo iterations. Between each leapfrog step, new values for the momentum variables are chosen at random. Consequently, the less confined direction is in this case explored via a random walk.

If we perform longer trajectories without refreshing the momenta, the q's continue moving in the same direction, and at least for short times the distance moved in coordinate space is proportional to L. This is illustrated in Fig. 4.4b which shows the results of ten hybrid Monte Carlo trajectories each consisting of ten leapfrog iterations, with the momentum being randomized only at the start of each trajectory. Note that the path taken does not resemble a random walk. Instead, each trajectory transversed a large distance in phase space.

However, if we use very long trajectories, we are likely to wastefully transverse the whole distribution several times, ending finally at a point similar to one that might have

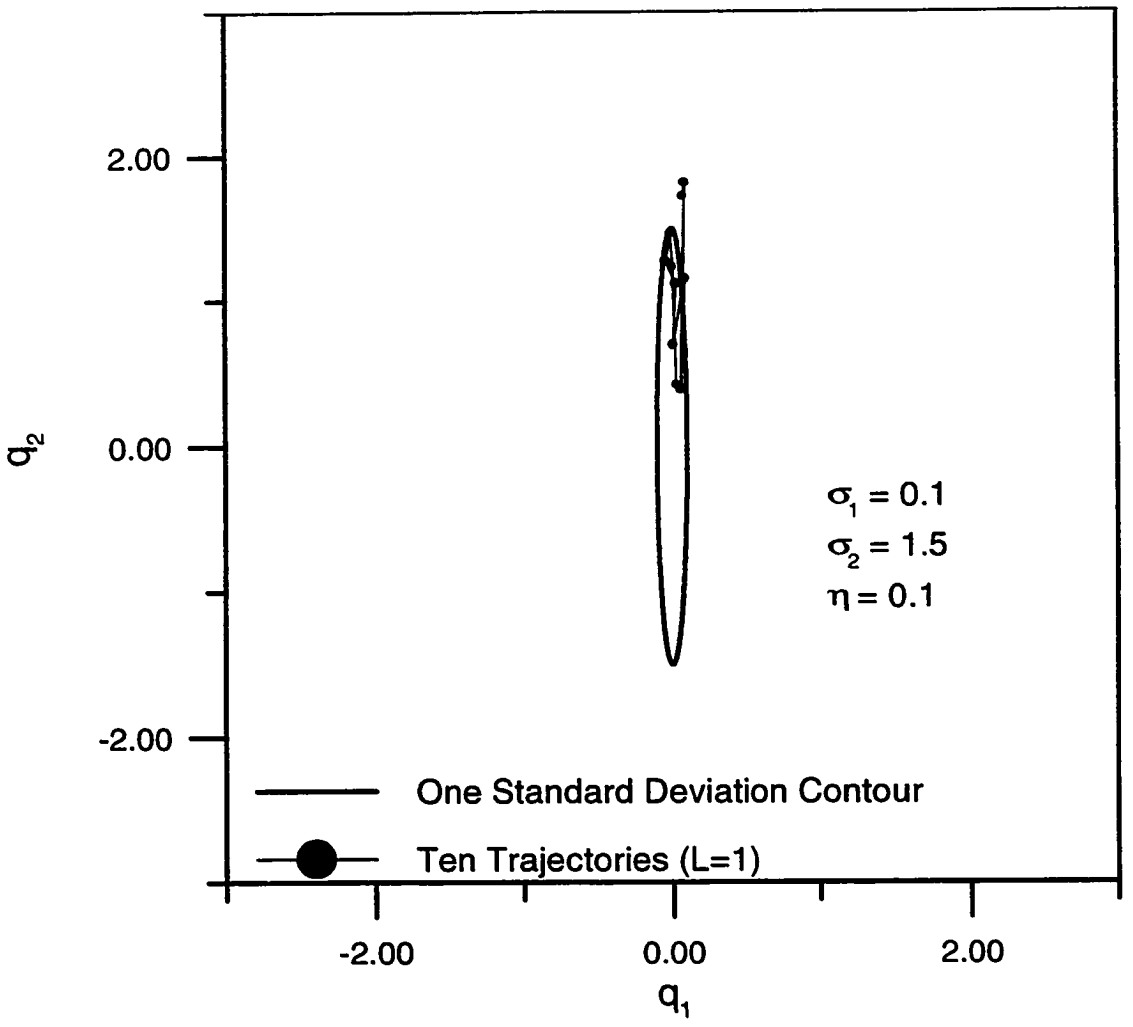

Fig. 4.4a - Sampling a Bivariate Gaussian Distribution with Langevin Method : Effect of Trajectory Size.

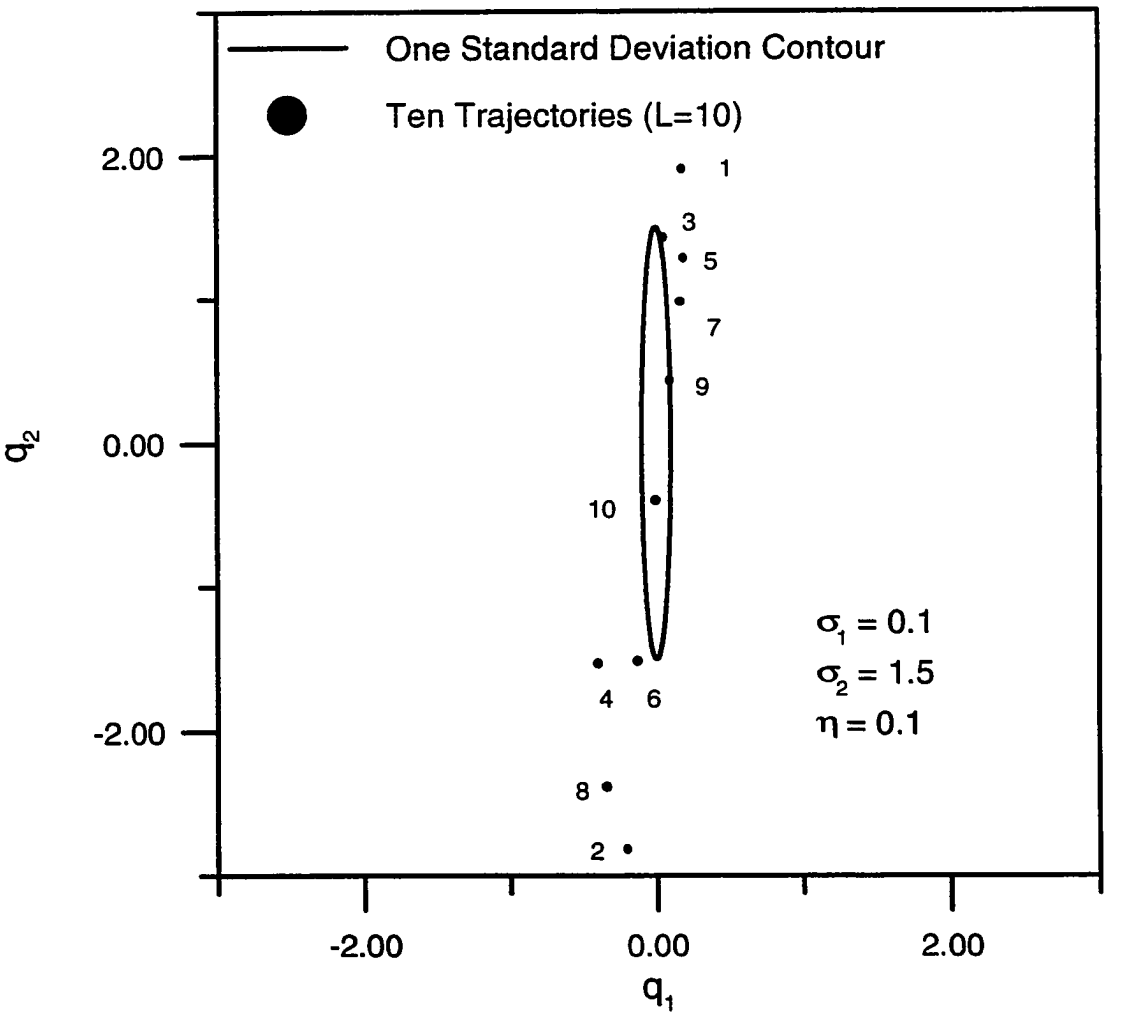

Fig. 4.4b - Sampling a Bivariate Gaussian Distribution with Hybrid Monte Carlo Method (L = 10) : Effect of Trajectory Size.

been reached by a shorter trajectory. In this case, simulating a long trajectory would simply be a waste of time because the final state may be close to the initial state. Mackenzie[40] (1989) has pointed out that the hybrid Monte Carlo method can fail to be ergodic for problems of this sort. This will occur if the stepsize and the number of leapfrog iterations used are such that the trajectory returns exactly to its starting point. This potential problem can easily be avoided by choosing the stepsize or the number of leapfrog iterations at random from within some range. We will see below that this fact was taken into account when using the variant developed by Horowitz. As mentioned before, since the dynamical steps conserve the total energy, we obviously cannot cover the whole phase space without refreshing the momenta. The optimal strategy is to choose L large enough so that successive samples are not correlated, i.e., so that we transverse the whole probability distribution instead of generating, all samples from a small region. Here, we used an heuristic approach to estimate a minimum value for L which is based on the bivariate case described in Eq. 4.19 and shown in Fig. 4.2b. As illustrated in Fig. 4.2b we see that, for the given stepsize value, a minimum value for L equal to ten is sufficient to transverse the whole distribution. In fact, we can say that we would like $L\varepsilon_i \approx \sigma_i$. If we adopt the choice of stepsize that $\varepsilon_i = \eta\sigma_i$ we have that $L\eta \approx 1$. We used this criterion to estimate a minimum value for L, and as described later, although heuristic, it provided good results during the sampling process, i.e. generated uncorrelated samples.

4.4 A Variant of the Hybrid Monte Carlo Method

Horowitz[8] introduced a modified form of the hybrid Monte Carlo algorithm in which the momenta are not completely replaced in the stochastic transitions, but are instead perturbed only slightly using the update of Eq. 4.13. This is coupled with the use of very short trajectories, consisting of only a single leapfrog step ($L = 1$).

In detail, given values for the magnitude of the leapfrog stepsize, ε and a value for α, a transition with the Horowitz variant consists of the following steps:

Starting from the current state, $(q,p)=(q(t_0),p(t_0))$, given α, a normal deviate n with mean zero and variance one and $p(t_0)$, a stochastic transition is performed using Eq. 4.13, and a new set of momentum variables p' is calculated. A dynamical step with one leapfrog iteration starting with $(q(t_0),p')$, with a stepsize of ε is performed, resulting in the state $(q(\varepsilon),p(\varepsilon))=(q^*,p^*)$. Regard (q^*,p^*) as a candidate for the next state, as in the Metropolis algorithm, accepting it with probability

$$\min(1,\exp\{-[H(q^*,p^*)-H(q(t_0),p')]\}),$$

and otherwise letting the new state have the same position variable as the old state with a different procedure for the momentum variables. The momentum variables for the new state in case of a rejection must be equal to $-p'$. This operation does not affect the Hamiltonian value obtained at the end of the dynamical transition but, as described by Horowitz[8], is necessary to make the process of updating the momentum variables by means of Eq. 4.13 satisfy the detailed balance.

Again, the analysis of the operation of ten hybrid Monte Carlo updates applied to the Hamiltonian of Eq. 4.14, with $\sigma = 1$, $\varepsilon = 1.0$ and $\alpha = 0.9$ shown in Fig. 4.5 can give insight into the main features of this variant. In this example, all the proposed (p,q) states were accepted. The starting point is shown at the point marked in the upper right quadrant, the values of variables $p(t_0)$, p', p^*, $q(t_0)$ and q^* at each trajectory are plotted and the circle where the initial trajectory lies is depicted.

Since after each update of the momentum variables by means of Eq. 4.13, these variables do not change very much, the total energy at the beginning of a hybrid Monte Carlo update changes only slightly from the value at the last state. Thus, the combination of stochastic and dynamical steps with this modified method is nearly equivalent to performing a single long trajectory of the standard algorithm (see Fig. 4.5). Since rejections will cause the subsequent transition to start with a negated value for the momentum variables, approximately preserves total energy, it is necessary to keep the rejection rate low. Later, we will take advantage of this nice feature of the Horowitz variant when sampling from the a posteriori probability distribution for our problem.

This simple example also helps to establishes a criteria to estimate of the parameter α. If the knowledge of the number of steps, L, necessary to generate an independent image at the end of a dynamical step for a given stepsize is available, it is possible to obtain a rough estimate of the parameter α. Recall that in the standard hybrid Monte Carlo algorithm, after L dynamical steps a stochastic transition is performed in which all the momentum variables are resampled. This operation provides a change in the total energy H and allows the algorithm to sample at this new level. Here, we want to calculate the parameter α that, given a value of L, makes the Horowitz method perform like the

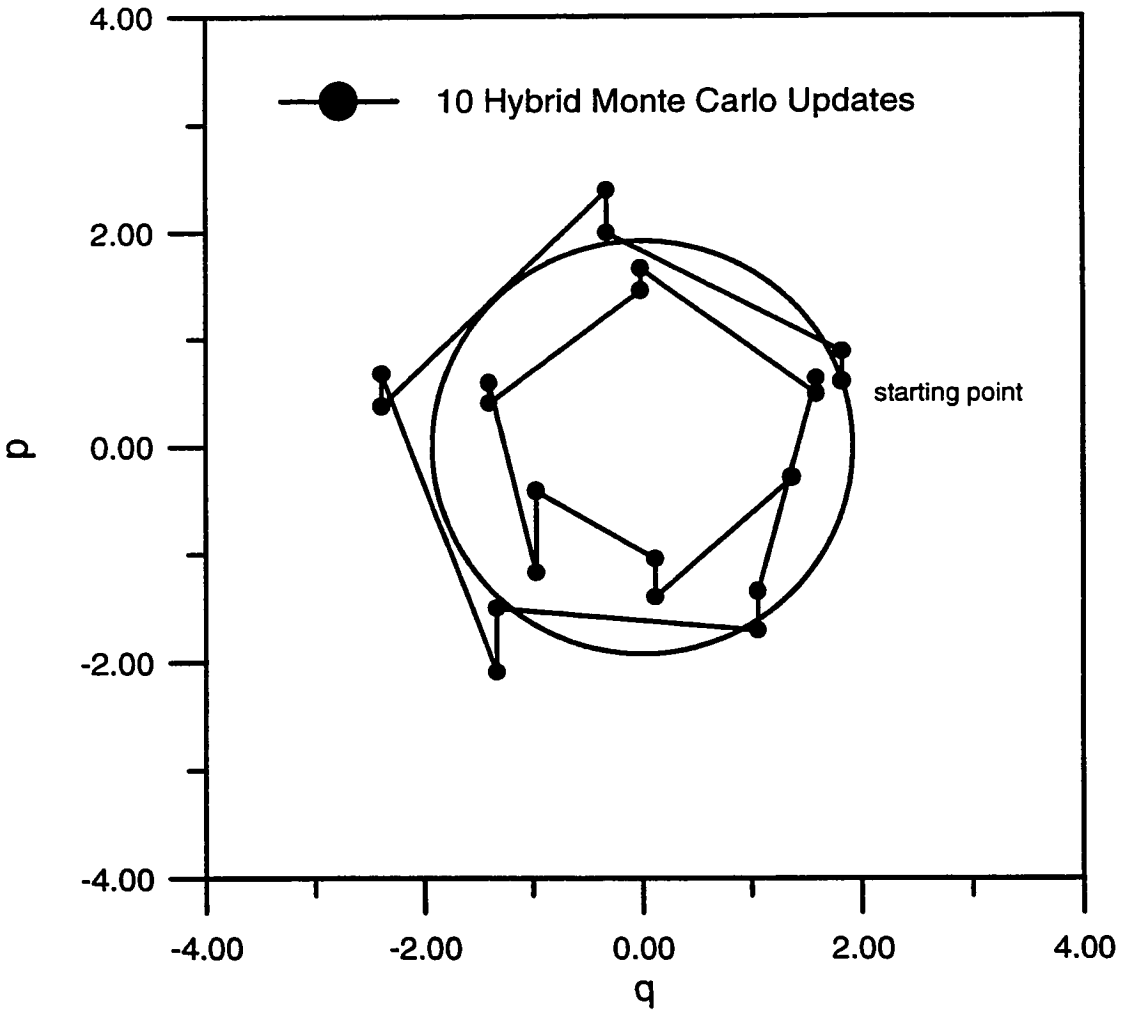

Fig. 4.5 - Horowitz Variant Applied to Sample a Univariate Gaussian Distribution.

standard algorithm. This can be accomplished assuming that after L hypothetical operations described by Eq. 4.13, we wish to change completely the momenta in order to reach a new total energy level, simulating then the standard hybrid Monte Carlo algorithm. For example, if the value desirable for L is 1000, we can conclude that using α equal to 0.99, we expect to completely change the original momenta vector in 1000 operations of Eq. 4.13.

4.5 Hybrid Monte Carlo Implementation

To apply the hybrid Monte Carlo method, we must formulate the desired distribution in terms of a potential energy function. Since we wish to sample from the a posteriori distribution for log-permeability fields, the energy will be a function of these parameters, which now play the role of the "position" variables, q, of an imaginary physical system. For the generic case described by Eq. 1.5, the potential energy is derived from the log of the expression for the a posteriori distribution as follows:

$$E(q) = -\ln[\pi(q)]. \tag{4.37}$$

The canonical distribution for this energy function, which is proportional to $\exp(-E(q))$, will then produce the a posteriori probability density for q.

For our problem the potential energy function is

$$E(m) = \frac{1}{2}((m-m_0)^T C_M^{-1}(m-m_0) + (g(m)-d_{obs})^T C_D^{-1}(g(m)-d_{obs})). \tag{4.38}$$

This energy function is similar to the objective function that is minimized with the use of a gradient method in Refs. 2-4. Recall, however, that the objective in the Monte Carlo

implemcntation here is not to find the minimum of the energy, but rather to sample from the corresponding canonical distribution.

To sample from this canonical distribution using the hybrid Monte Carlo method, we introduce transformed momentum variables, $\hat{p}_i = p_i / \sqrt{m_i}$, in one-to-one correspondence with the position variables, q_i. With each momentum variable, a positive "mass", m_i is associated. We work with these transformed variables $\hat{p}_i$, which define the kinetic energy, $K(\hat{p})$, associated with the momentum, with the result that the canonical distributions for the $\hat{p}_i$'s are Gaussian random variables with means of zero and variances equal to one and are independent and identically distributed. As described in Section 4.2, a single hybrid Monte Carlo update starts by generating new values for all the momentum variables from their canonical distribution. A candidate state is then found by following a trajectory computed using the leapfrog discretization of Hamiltonian dynamics, applied for L iterations, using some stepsize, ε_i. When using the variant of Horowitz, the momentum variables are given by Eq. 4.13 and L=1 is used in the leapfrog discretizations of Hamiltonian dynamics. Finally this candidate is accepted or rejected based on the change in total energy, $H(m,\hat{p}) = E(m) + K(\hat{p})$. Calculation of the derivatives or gradient of E with respect to the q_i is required in order to perform the leapfrog iterations; these derivatives are given by

$$\nabla_m E(m) = G^T C_D^{-1}(g(m) - d_{obs}) + C_M^{-1}(m - m_0). \qquad (4.39)$$

In Appendix B, we show that it is possible to reparameterize our model in terms of the spectral decomposition (eigenvalue/eigenvector) of the a posteriori covariance matrix (C_{MP}) evaluated at the maximum a posteriori estimate obtained by the Gauss-Newton

method. i.e., $C_{MP} = U\Lambda U^T$, where Λ is the diagonal matrix with the eigenvalues from the spectral decomposition of C_{MP} and U is an orthonormal matrix with columns that contain the eigenvectors obtained from the spectral decomposition of C_{MP}. In this case, our reparameterized model can be expressed by

$$m_p = U^T m \Leftrightarrow m = U m_p, \tag{4. 40}$$

$$m_{p,\infty} = U^T m_\infty \Leftrightarrow m_\infty = U m_{p,\infty}. \tag{4. 41}$$

As shown in Appendix B, we can write the gradient of the potential energy and the resulting expression in terms of the reparameterized model, m_p, as follows:

$$\begin{aligned} \nabla_{m_p} E(m_p) = {} & \Lambda^{-1}(m_p - m_{p,\infty}) \\ & + U^T G_\infty^T C_D^{-1} \varepsilon(m) \\ & + U^T [G^T - G_\infty^T] C_D^{-1} (g(m) - d_{obs}), \end{aligned} \tag{4. 42}$$

where

$$\varepsilon(m)^T = g(m)^T - g(m_\infty)^T - (m - m_\infty)^T G_\infty^T, \tag{4. 43}$$

and m_∞ is the maximum a posteriori estimate obtained with the Gauss-Newton method and Λ and U are matrices resulting from the spectral decomposition procedure.

This approach provides the basis for determining the stepsize ε_i to be applied during the dynamical transitions of the hybrid Monte Carlo method implemented in this work. This was accomplished by assuming that, as an approximation, our goal was to sample a linear problem described by the probability distribution

$$\hat{\pi}(m) = \hat{a}\exp[-\frac{1}{2}(m-m_\infty)^T C_{MP}^{-1}(m-m_\infty)]. \qquad (4.44)$$

In terms of the reparameterized model, the gradient of the potential energy term associated with Eq. 4.44 is given by

$$\nabla_{m_p} E(m_p) = \Lambda^{-1}(m_p - m_{p,\infty}). \qquad (4.45)$$

This means that for this linear case, the spectral (or eigenvalue-eigenvector) decomposition of C_{MP} will translate and rotate the coordinate system removing the interactions between the components. It is straightforward to see, through an analogy with our previous multivariate Gaussian example, that, in this case, the optimum stepsize for each component will be $\varepsilon_i \approx \eta\sqrt{\lambda_i}$ (the λ_i's are the eigenvalues resulting from the spectral decomposition of the a posteriori matrix) where for stability we require $\eta < 2$.

Figure 4.6 shows the error in energy for trajectories computed with different stepsizes, ε_i which were defined as $\varepsilon_i = \eta\sqrt{\lambda_i}$. Each point plotted in the figure shows the change in total energy (H) for a trajectory of one hundred leapfrog iterations for an example with 225 variables in which the stepsizes were adjusted by the factor η. Changes in the total energy greater than ten were plotted at ten. As we can observe, when $\eta \approx 1$ we have some inaccuracy exhibited, but only for $\eta \geq 2$, does the leapfrog discretization become unstable. Instability will result in an acceptance rate close to zero. The optimal strategy is usually to select a $\eta \approx 1$.

However, when considering our non-linear problem, reparameterization will not be able to remove all the interaction between the differents components, so in this implementation, a heuristic approach was used in which the stepsizes are set as follows:

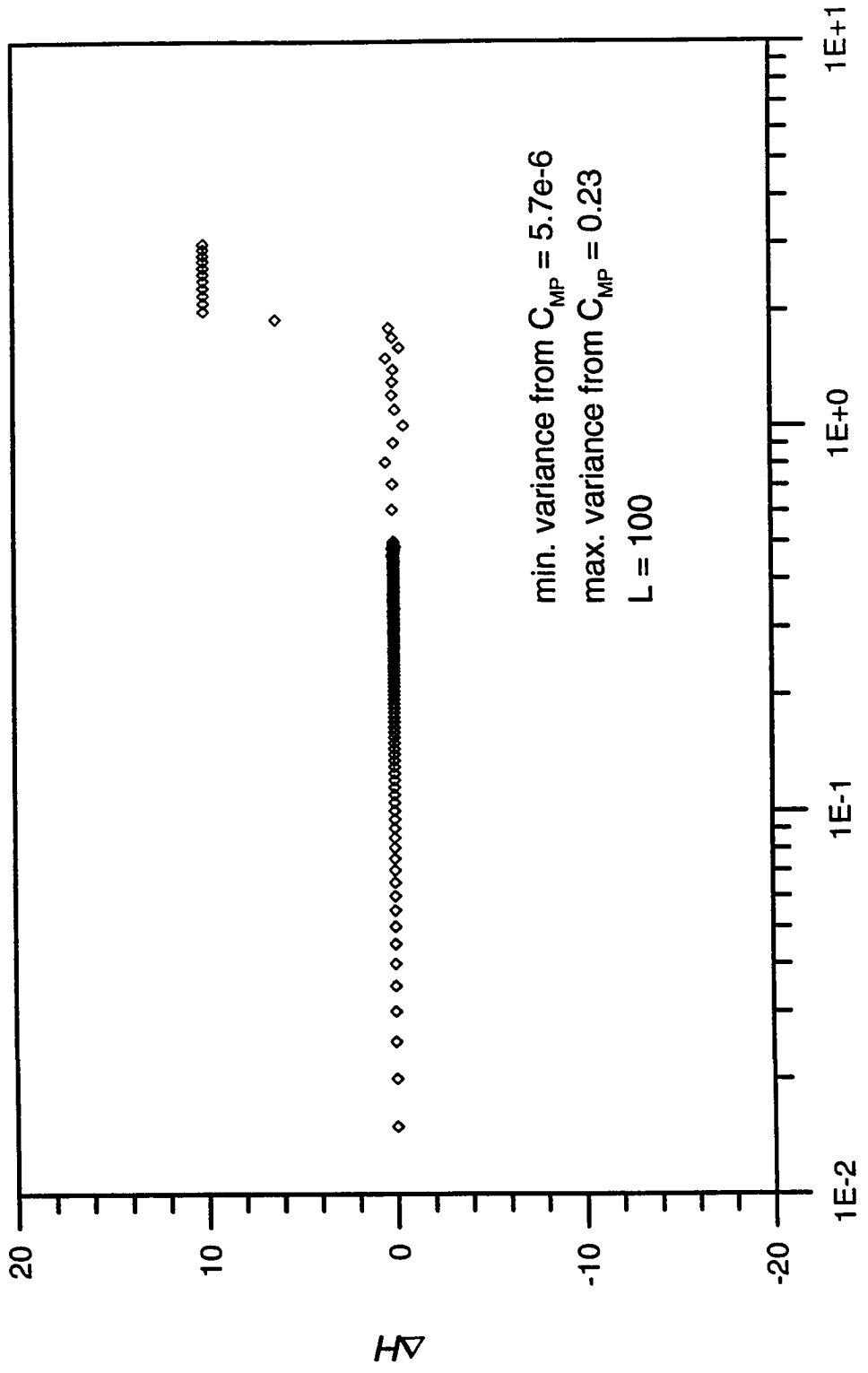

Fig. 4.6 - Errors in a Trajectory when Sampling a Correlated Multivariate Gaussian Distribution.

$$\varepsilon_i \approx \eta\sqrt{\lambda_i}\,, \tag{4.46}$$

where η is left as a tunable parameter chosen experimentally in order to obtain a good acceptance rate. In this approach, a spectral decomposition of the a posteriori matrix, obtained by a Gauss-Newton method, must first be obtained and the stepsize adjustment factor, η is then estimated.

The computation of the gradient at each leapfrog step requires a high percentage of the total time of the run when sampling with a hybrid Monte Carlo algorithm due to the fact that it requires the computation of pressure and the sensitivity coefficients from a flow simulator.

When improving the method efficiency, one important aspect to be considered is the decrease in the computational time spent when estimating the gradient. As described in Refs. 7, 38 and 39, the Hamiltonian used in the Metropolis step and the one used in the leapfrog equations can be different. This suggests, that at each leapfrog step we use an "approximate" Hamiltonian which considers only the first term of the Eq. 4.42 for the gradient. This interpretation is convenient because the first term of the expression listed in Eq. 4.42 requires almost no computational effort when compared with the second and third terms. This is so because at each leapfrog step the calculation of the second term on the right side of Eq. 4.42 requires using the simulator to compute the wellbore pressure ($g(m)$) for each model m and the evaluation of the third term requires the calculation of sensitivity coefficients as well as the wellbore pressure.

In Appendix C we show that when using the approximate Hamiltonian, the Hamiltonian equations can be solved analytically and the general solutions for the variables $\hat{p}$ and m_p after a trajectory of $\Delta\tau = \tau - \tau_0$ are

$$\hat{p} = \hat{p}_0 \cos\left[\frac{(\tau - \tau_0)}{\sqrt{\lambda}}\right] + \frac{1}{\sqrt{\lambda}}(m_{p,\infty} - m_{p,0})\sin\left[\frac{(\tau - \tau_0)}{\sqrt{\lambda}}\right], \tag{4. 47}$$

and

$$m_p = (m_{p,0} - m_{p,\infty})\cos\left[\frac{(\tau - \tau_0)}{\sqrt{\lambda}}\right] + \sqrt{\lambda}\,\hat{p}_0 \sin\left[\frac{(\tau - \tau_0)}{\sqrt{\lambda}}\right] + m_{p,\infty}. \tag{4. 48}$$

The value used for $\Delta\tau$ may be chosen at random from some fixed distribution. This may be useful when the best values are not known, or vary from place to place. Some random variation may also be needed to avoid periodicities that could interfere with ergodicity[40].

As usual, sampling from a distribution using a Markov chain Monte Carlo method is a two-phase process. As mentioned before, in the *initial or transient phase*, we start from some initial state, and simulate a Markov chain for as long as is needed for it to reach a rough approximation to the desired distribution. In the *sampling phase*, we continue from the state reached at the end of the initial phase, proceeding for a certain fixed number of iterations. If we start from an initial state that is an approximation of a sample of the probability distribution, we expect to reduce the length of the transient phase, improving the efficiency of the method.

It is important to notice that in this implementation, the log-permeability field used as a starting point was obtained from an unconditional simulation that used the maximum a posteriori estimate (m_∞) and the posteriori matrix (C_{MP}) obtained from a Gauss-Newton procedure through the expression

$$m_{initial} = m_{\infty} + L_r Z_{initial}, \tag{4.49}$$

where $Z_{initial}$ is a vector of uniform random deviates and L_r results from the Cholesky decomposition of the a posteriori covariance matrix, i.e. $C_{MP} = L_r L_r^T$. As demonstrated with the procedure given in Chapter II, initiating the chain using Eq. 4.49 considerably reduces the length of the transient period since the distribution of states from Eq. 4.49 constitute a reasonable approximation for the probability distribution for the log-permeability fields.

4.6 Computational Examples

4.6.1 A Demonstration Using the Standard Hybrid Monte Carlo Method

To illustrate the use of the implementation based on hybrid Monte Carlo, and provide an idea of its performance, we show here how it can be applied to sample a probability distribution for our problem. To do this, we use the examples in which all data and information are the same as for Example 1 (variance = 0.25) and Example 2 (variance = 1) given in Chapter II.

In this implementation, the reparameterized model is used solely to estimate the stepsizes ε_i, by means of Eq. 4.46 and the eigenvalues calculated from the spectral decomposition of the a posteriori covariance matrix C_{MP}. The dynamical steps are performed using the gradient expression given by Eq. 4.38 and the Metropolis criteria applied at the end of a dynamical transition is calculated using the Hamiltonian function, which is expressed as follows:

$$H(m,\hat{p}) = \frac{1}{2}((m-m_0)^T C_M^{-1}(m-m_0) + (g(m)-d_{obs})^T C_D^{-1}(g(m)-d_{obs})) + \sum_{i=1}^{n} \frac{\hat{p}_i^2}{2}. \quad (4.50)$$

The gradient calculation uses sensitivity coefficients given by a flow simulator.

4.6.1.1 Example 1a - Variance = 0.25

Here, we use an example in which all data and information are the same as for Example 1 listed in Chapter II. The first step of the implementation consists in finding good values for the stepsize adjustment factor, η, and the trajectory length, L, for use in the sampling phase.

Fig. 4.7 shows data on how the error in total energy varies with η. Trajectories of length L = 10 were used. As can be seen, for η greater than about $2x10^{-3}$, the leapfrog method becomes unstable, and very large errors result, which would lead to a very high rejection rate if such a value of η were used. Thus, the value for $\eta = 2x10^{-3}$ was used for the sampling phase. This result was the same for several trajectory lengths tested. In fact, the stability criterion seemed to be insensitive to the parameter L. If we compare this value with the magnitude of the permissible η values for the multivariate case discussed in previous sections, we can see that the value of η determined here is much smaller. This is a result of the linear approximation used to estimate the mass term in terms of the eigenvalues of C_{MP}; see Eq. 4.46. We estimate the values of the stepsizes, ε_i's, based on the linear case and apply the result to our problem.

In order to minimize the extent to which the Markov chain undertakes a random walk, L should be chosen so that components of the state at the end-point of a trajectory

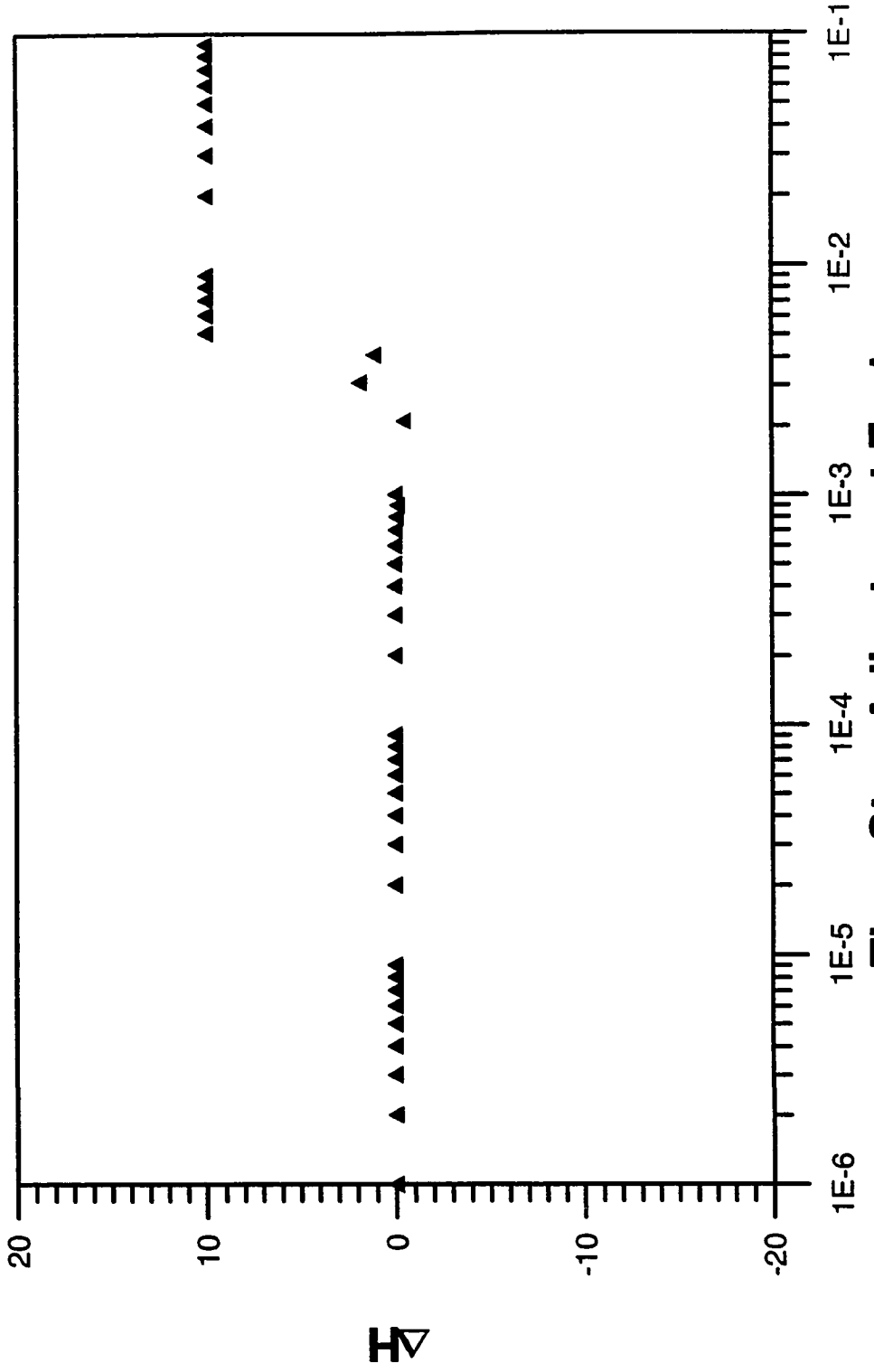

Fig. 4.7 - Error in a Trajectory when Sampling the a Posteriori Distribution Function (Variance = 0.25).

are uncorrelated with the corresponding values at the starting point. But since most of the computation time required in this implementation is expended in performing the leapfrog iterations, and since to evaluate the derivatives of E needed in each such iteration, one must use the flow simulator, it is desirable to have L as small as possible. Based on univariate Gaussian examples, we could conclude that a good estimate for a minimum L will be to choose a value such that $\eta L = 1$. In our example, since $\eta = 2 \times 10^{-3}$ which suggests we use L equal to 500 as a minimum value.

Fig. 4.8 presents a plot of the value of "objective function" of accepted states versus the perturbation number. A corresponding plot of the pressure misfit is shown in Fig. 4.9. Again, the perturbation number in Figs. 4.8 and 4.9 refers to the number of transitions proposed, but only the values of accepted transitions are plotted. The state remains unchanged between accepted transitions, but those have not been plotted. The results of Figs. 4.8 and 4.9 allow us to determine if the transient period of the Markov chain has ended. When this occurs, the values of the objective function at states will appear to oscillate around a constant value so the overall plot will appear relatively constant, whereas, during the transient period, the plot of the objective function, though oscillatory, will show a decreasing trend. In the pressure misfit plot, when the transient period has ended, we should be able to obtain states which give pressure misfits less than the standard deviation of the pressure measurement errors, i.e., pressure misfits less than 0.15 psi. From Figs. 4.8 and 4.9 and the previous criteria, we can see that the transient period for the hybrid Monte Carlo method, is fairly short.

Figs. 4.10a and 4.10b describe the variation in the log of permeability in four gridblock locations (see gridblock locations circled in Fig. 2.4.6) versus iteration number.

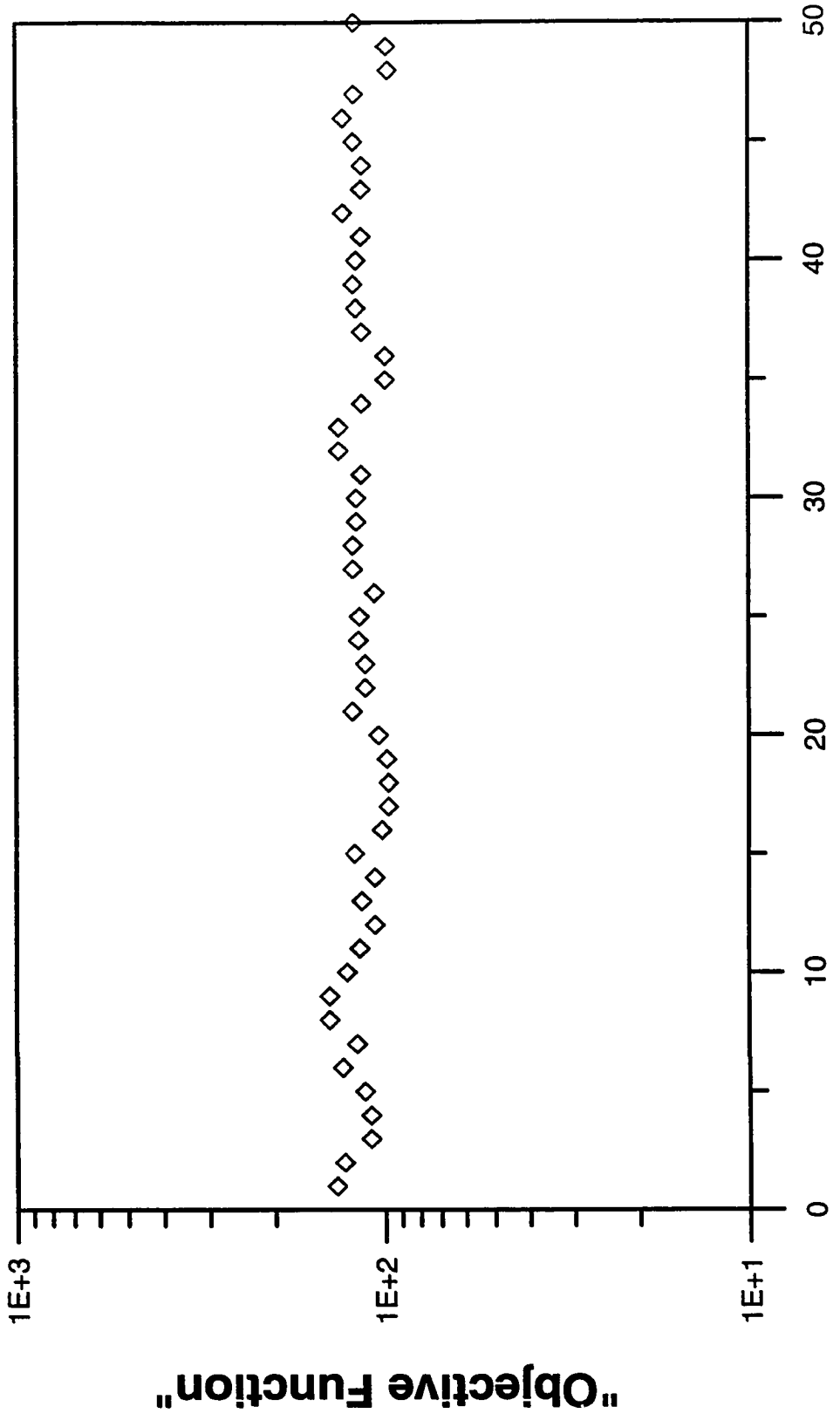

Fig. 4.8 - Values of the Objective Function for the Hybrid Monte Carlo Method (Variance = 0.25).

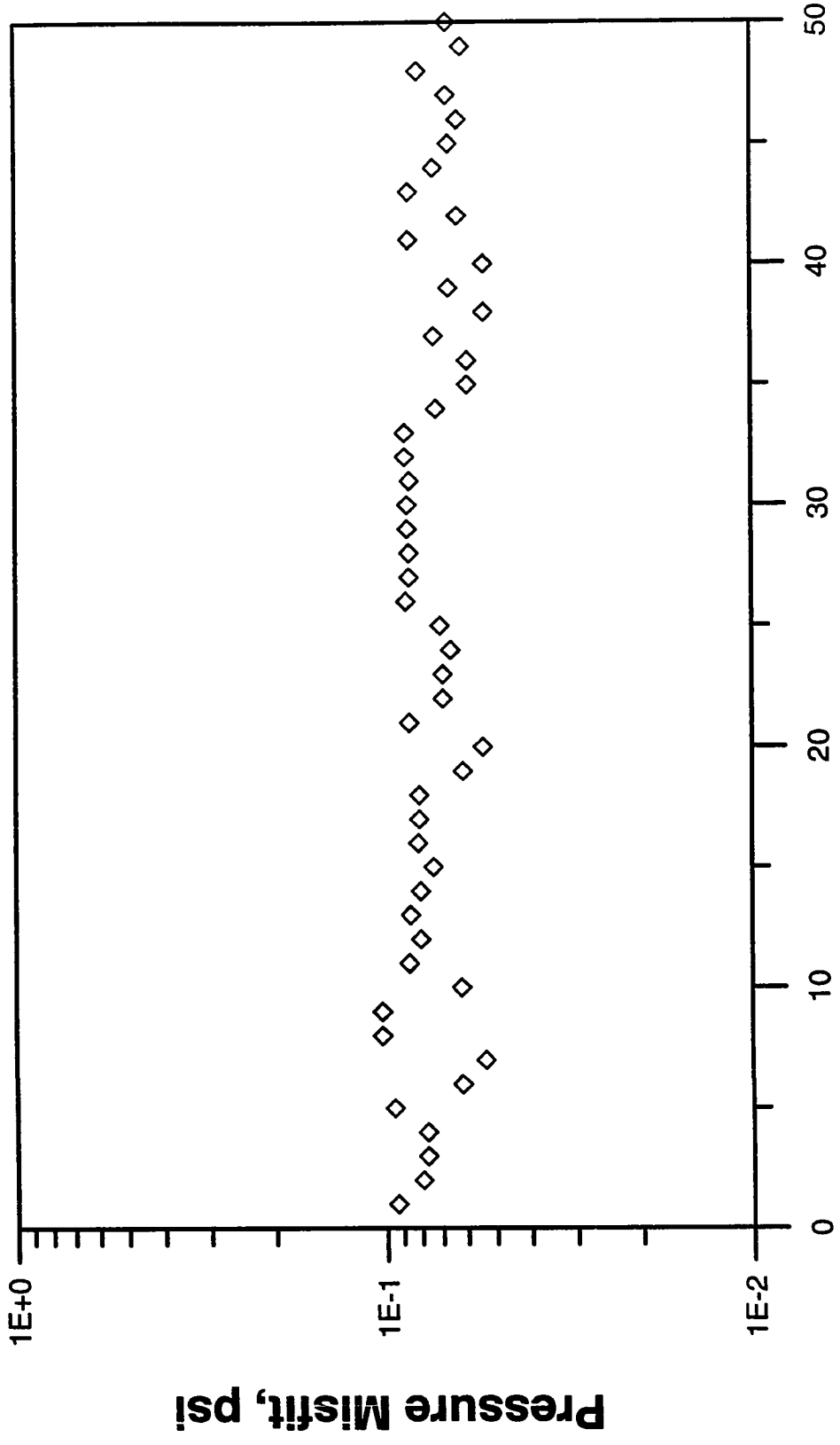

Fig. 4.9 - Values of the Pressure Misfit for the Hybrid Monte Carlo Method (Variance = 0.25).

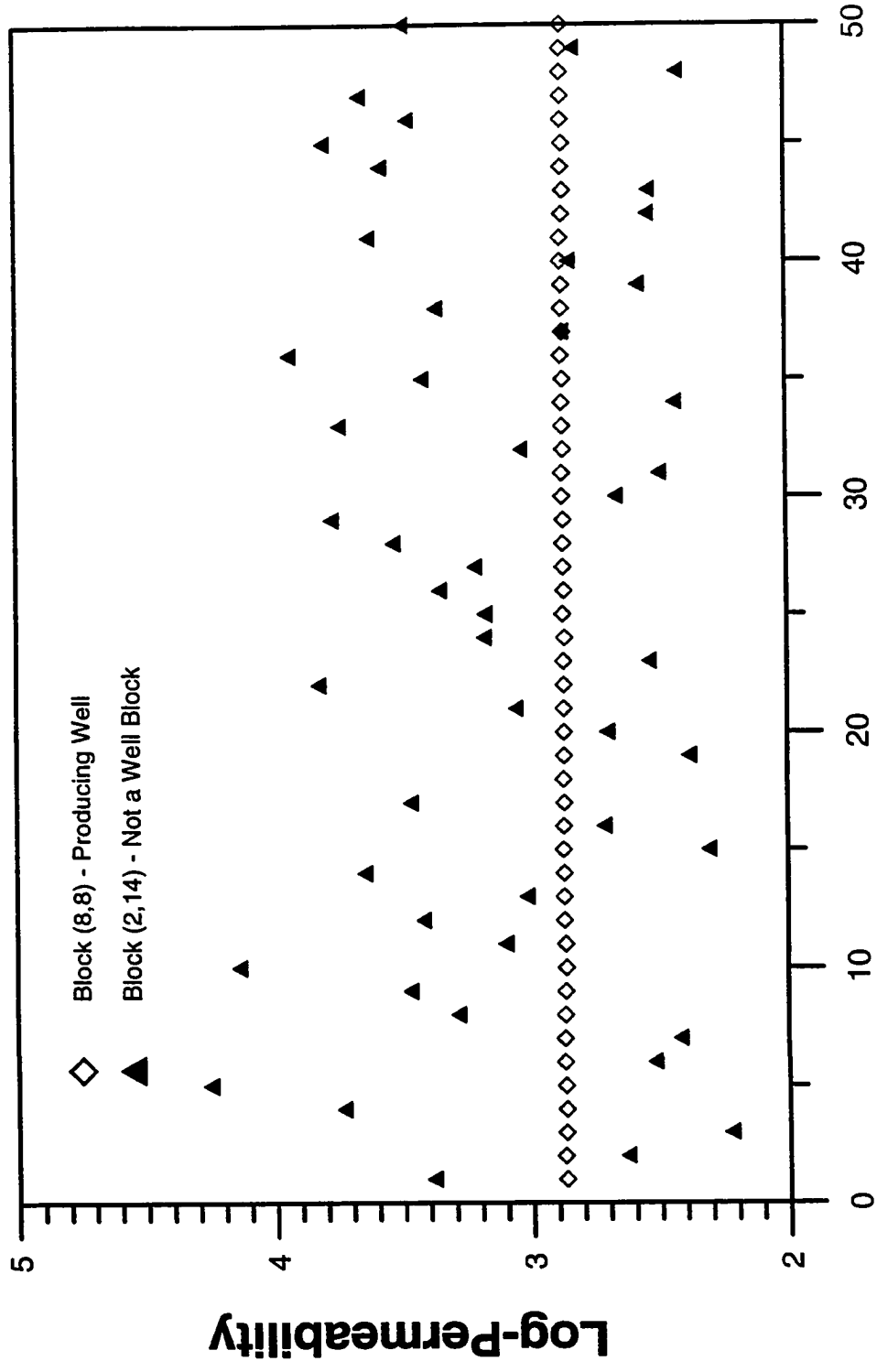

Fig. 4.10a - Log-Permeability Values During Sampling with Hybrid Monte Carlo Method (Variance = 0.25).

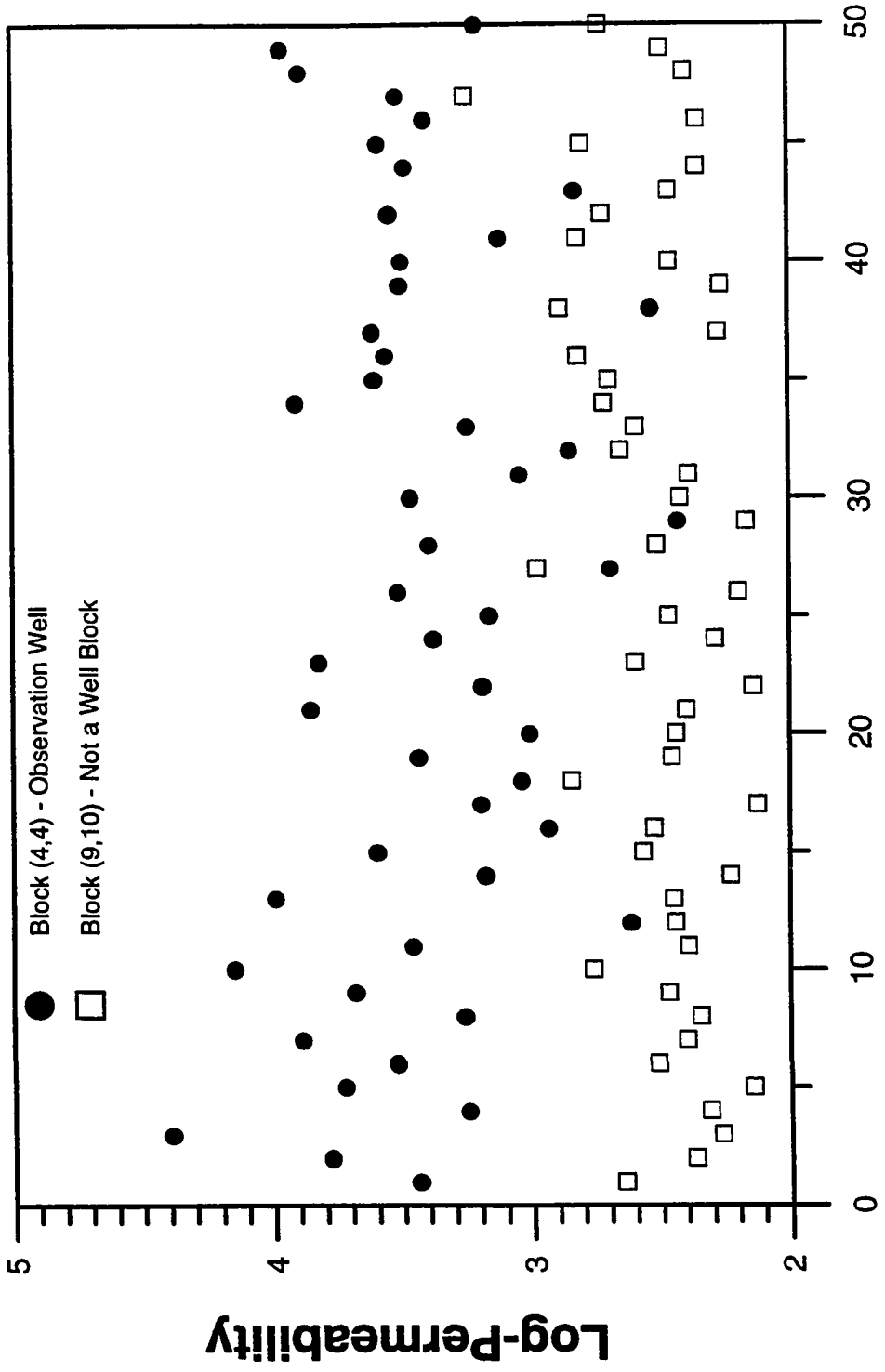

Fig. 4.10b - Log-Permeability Values During Sampling with Hybrid Monte Carlo Method (Variance = 0.25).

The a posteriori variance for $\ln(k)$ in gridblocks (4,4), (8,8), (9,10) and (2,14), respectively were 0.17, 6×10^{-6}, 0.07 and 0.24. The maximum a posteriori estimates of log-permeability for the same gridblocks were 3.68, 2.87, 2.54 and 3.43, respectively.

Considering the behavior of the sampling of log-permeability at each block, we can see the dominant influence that the a posteriori variances, through the stepsize calculation, have in the realizations generated during the hybrid Monte Carlo procedure. The sampled value for the producing well gridblock location stays almost constant and equal to the maximum a posteriori estimate for that gridblock through the whole simulation. This reflects the fact that at the producing well location the uncertainty is considerably reduced with incorporated pressure data together with the prior model information. This can be seen by the very low value encountered for the a posteriori variance.

At gridblock (9,10) between the producing and a interference well, the pressure is resolved quite well as indicated by the low value (0.07) of the a posteriori variance. Since the variance of 0.07 corresponds to a standard deviation of 0.26, a larger variation in the log-permeability values for this gridblock occurs during the sampling procedure as compared to that obtained for the producing well gridblock. This can be observed in Figs. 4.10a and 4.10b.

At the observation well gridblock, (4,4) the pressure data are not sufficient to reduce the uncertainty in the permeability estimate (a posteriori variance of 0.17) we can see that a large variation in the log-permeability occurs during the sampling process. This is also the case for gridblock (2,14) which is located close to the corner of the reservoir.

In relation to the number of images accepted, a total number of 50 different realizations were generated in 50 perturbations resulting in an acceptance rate of 100%.

Regarding computational time, this method required approximately 37 hours to attempt 50 perturbations.

4.6.1.2 Example 1b - Variance = 1

As described before, the first step of the implementation consists in finding good values for the stepsize adjustment factor, η, and the trajectory length, L, for use in the sampling phase. Fig. 4.11 shows data on how the error in total energy varies with η. These data were obtained by continuing the simulation from the initial state, using various values of η. Trajectories of length $L = 10$ and $L = 100$ were used. As can be seen, for η greater than about $1x10^{-3}$, the leapfrog method becomes unstable, and very large errors result, which would lead to a very high rejection rate if such a value of η were used. This result was the same for both values of L tested and although not shown, similar results were obtained for values of $L \geq 1000$, i.e. the optimum time step adjustment factor was insensitive to the parameter L. Thus, the value of $\eta = 1x10^{-3}$ was used for the sampling phase.

Figs. 4.12 presents a plot of the value of "objective function" of accepted states versus the perturbation number. A corresponding plot of the pressure misfit is shown in Fig. 4.13. The perturbation number in Figs. 4.12 and 4.13 refers to the number of transitions proposed, but only the values of accepted transitions are plotted. When sampling with the hybrid Monte Carlo method the rejected transitions result in acceptable images, i.e., if a proposed state is rejected the new state is identical to the old state, but those states have not been displayed in Figs. 4.12 and 4.13.

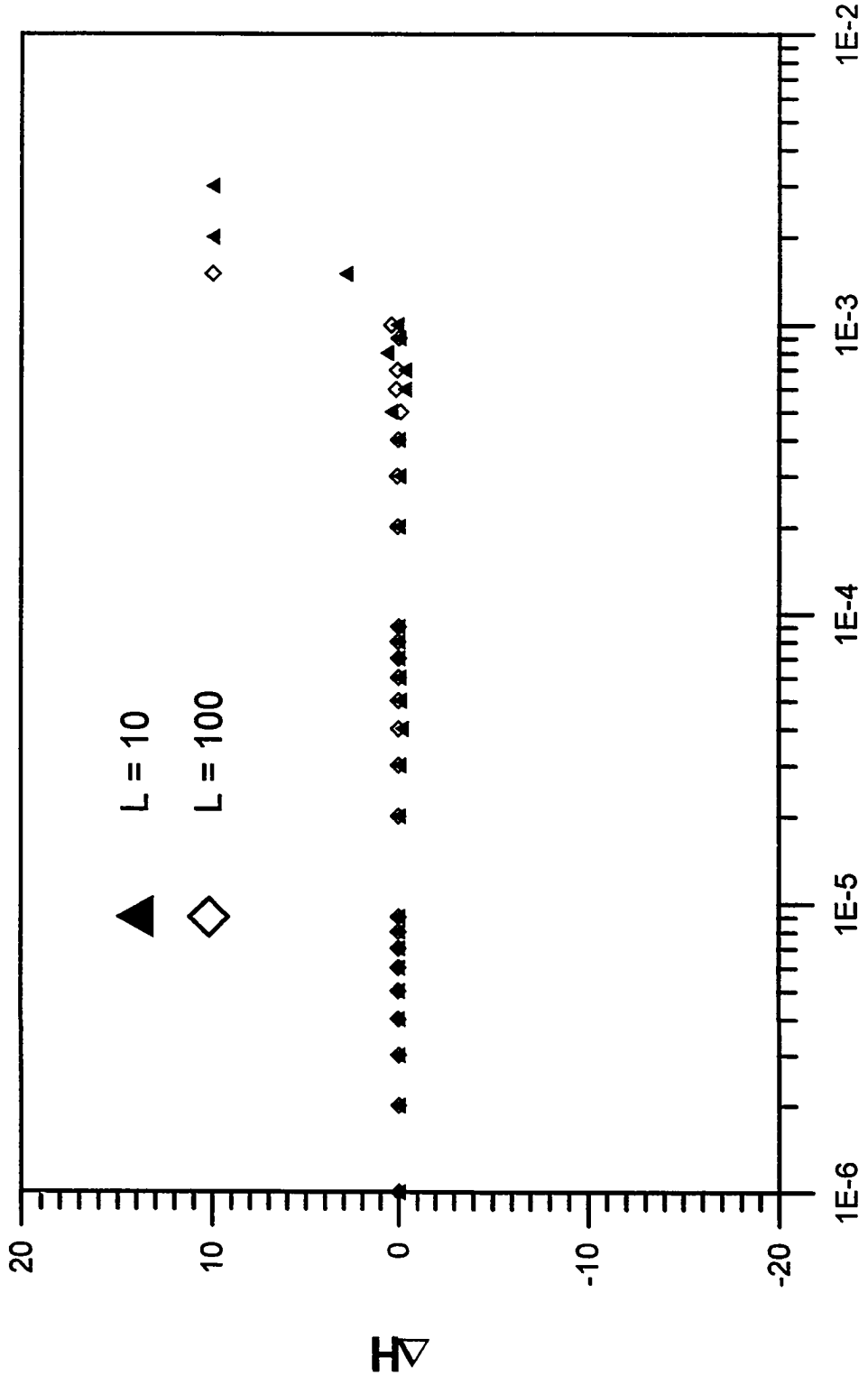

Fig. 4.11 - Error in a Trajectory when Sampling the a Posteriori Distribution Function (Variance = 1.0).

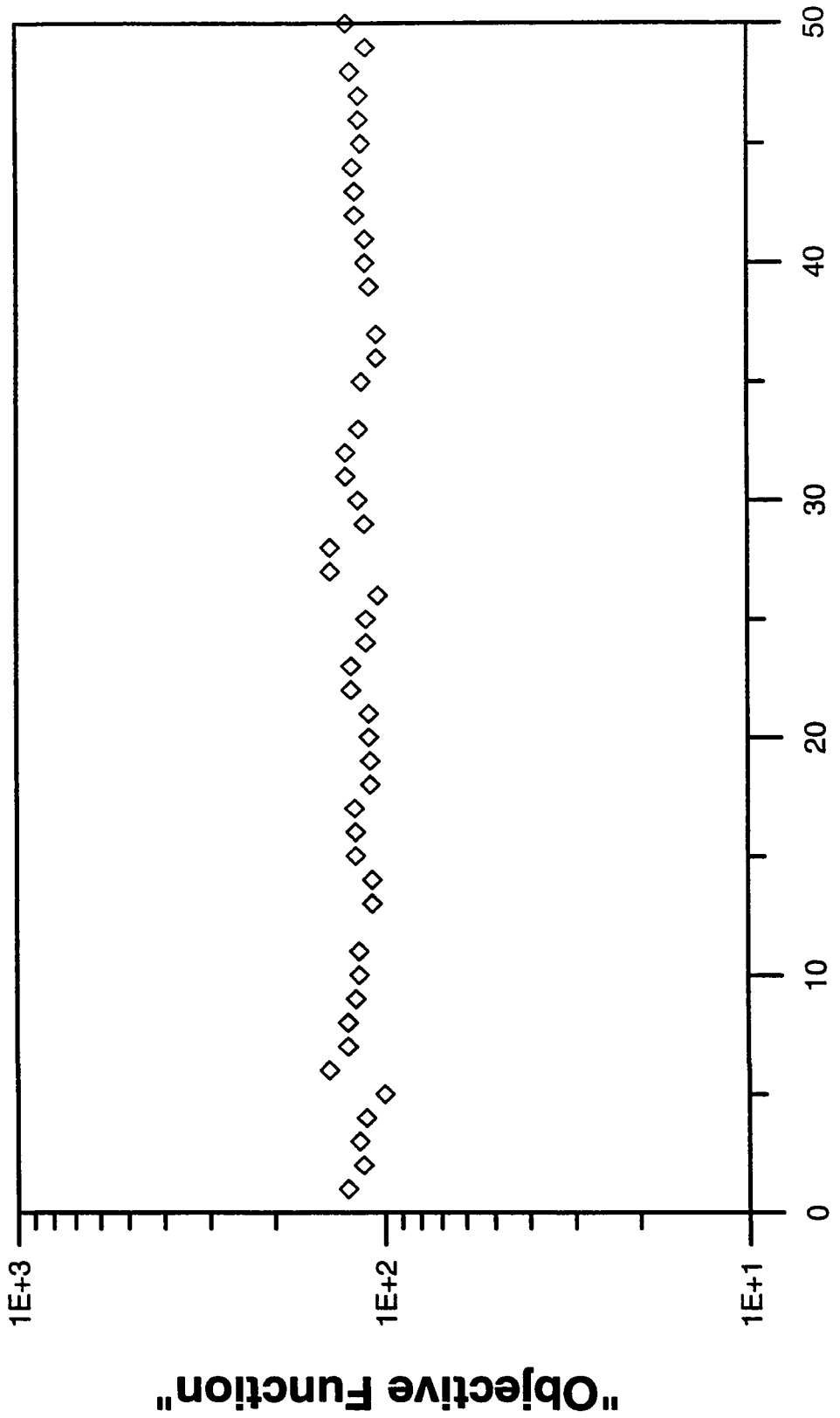

Fig. 4.12 - Values of the Objective Function for the Hybrid Monte Carlo Method (Variance = 1.0).

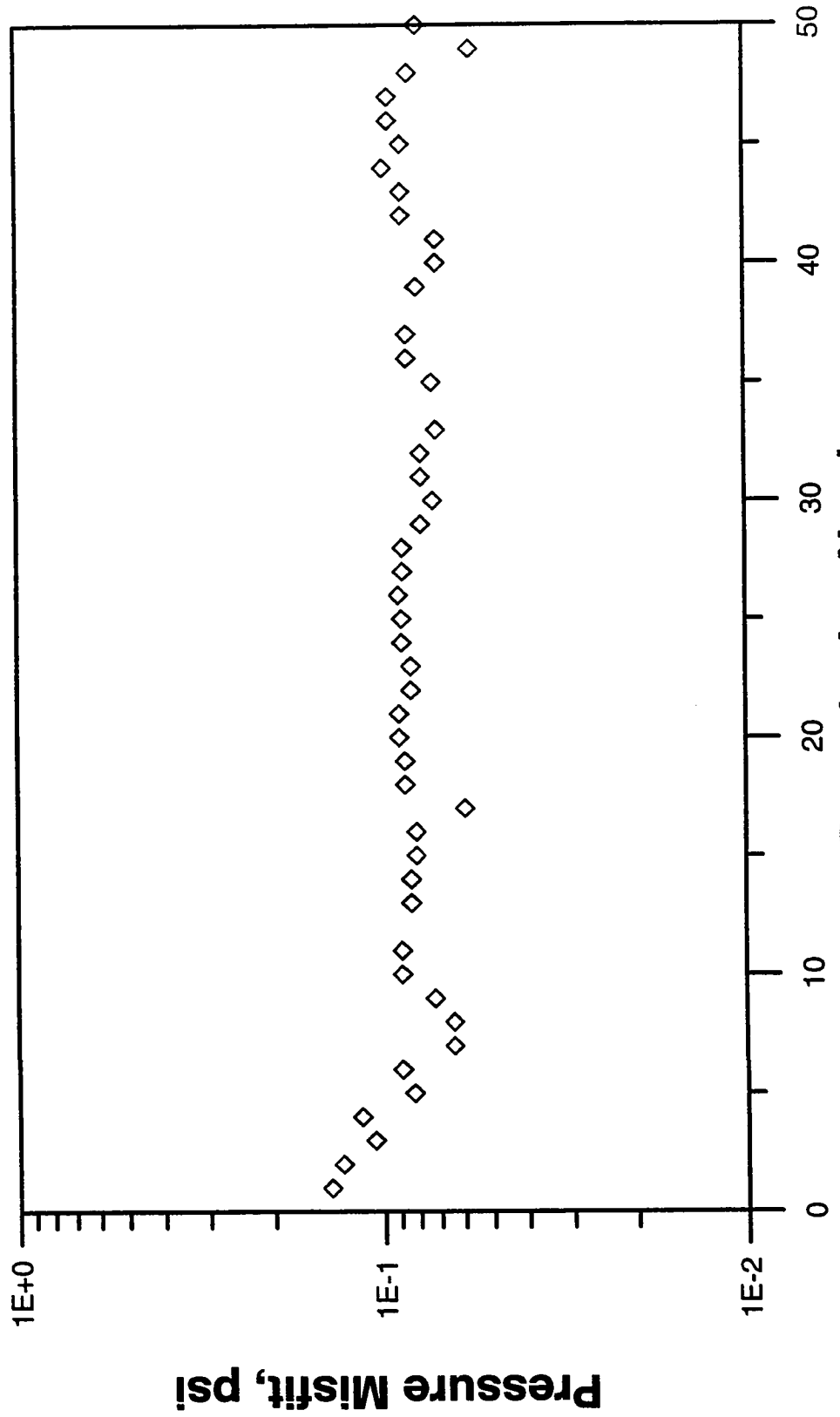

Fig. 4.13 - Values of the Pressure Misfit for the Hybrid Monte Carlo Method (Variance = 1.0).

The transient period is short, as expected due to our procedure of generating the initial log of permeability field from the decomposition of the a posteriori matrix obtained from a gradient method. The behavior of the Hamiltonian function is described in Fig. 4.14. The Hamiltonian function behaves similarly to the objective function and when the stationary distribution is reached, the values oscillate around some level. Note that, this level has to be higher than the one where the objective function oscillates. This is due to the fact that the Hamiltonian function reflects the value of the objective function ("potential" energy) plus a positive quantity which is a function of the momentum variables ("kinetic" energy).

Figs. 4.15a and 4.15b describe the variation in the log of permeability in four gridblock locations (see gridblock locations circled in Fig. 2.4.6) versus iteration number. The a posteriori variance for $\ln(k)$ in gridblocks (4,4), (8,8), (9,10) and (2,14), respectively were 0.59, 2.4×10^{-6}, 0.24 and 0.93, respectively. The maximum a posteriori estimates of log-permeability for the same gridblocks were 3.69, 2.48, 2.00 and 3.51, respectively.

In this implementation, the a posteriori matrix was used to give and initial starting point for the log-permeability field and also to give an estimate for the stepsize of each component. Considering the behavior of the sampling of permeability at each block, we can see the dominant influence that the a posteriori variances through the stepsize calculation have in the realizations generated during the hybrid Monte Carlo procedure. The sampled value for the producing well gridblock location stays almost constant and equal to the maximum a posteriori estimate for that gridblock through the whole simulation. This reflects the fact that at the producing well location the uncertainty is considerably reduced when incorporating pressure data together with the prior model

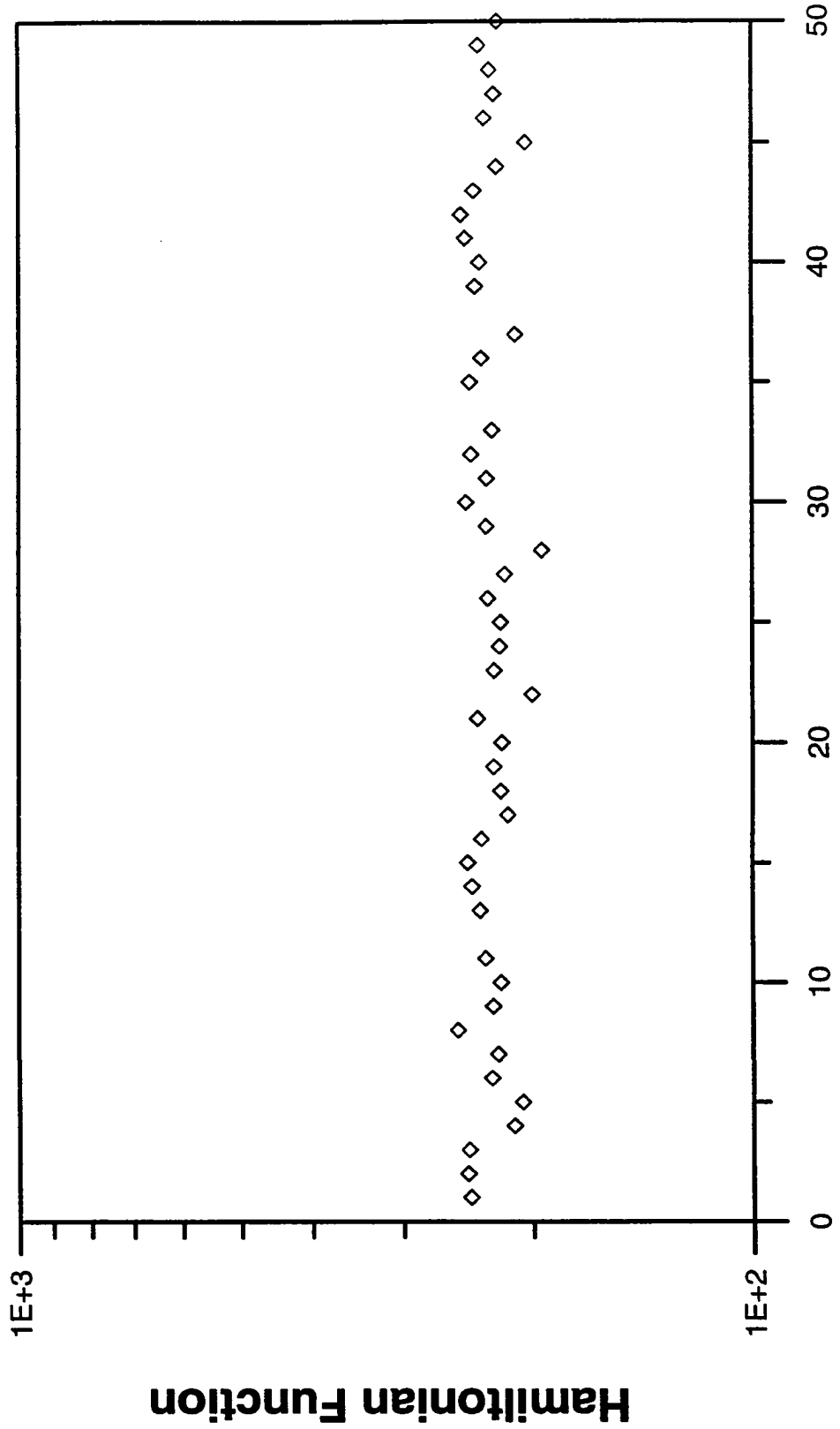

Fig. 4.14 - Values of the Hamiltonian Function for the Hybrid Monte Carlo Method (Variance = 1.0).

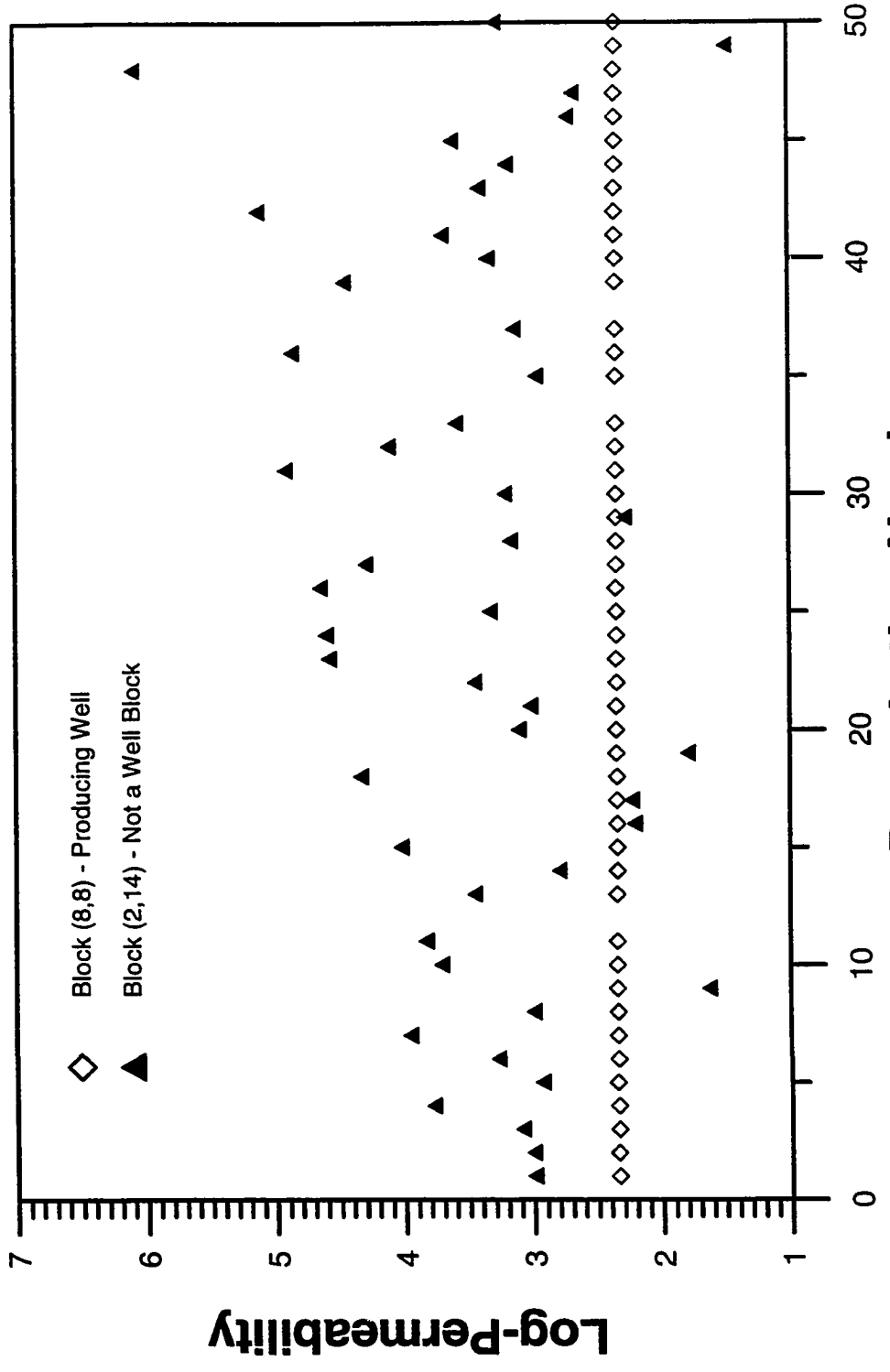

Fig. 4.15a - Log-Permeability Values From Sampling With Hybrid Monte Carlo Method (Variance = 1.0).

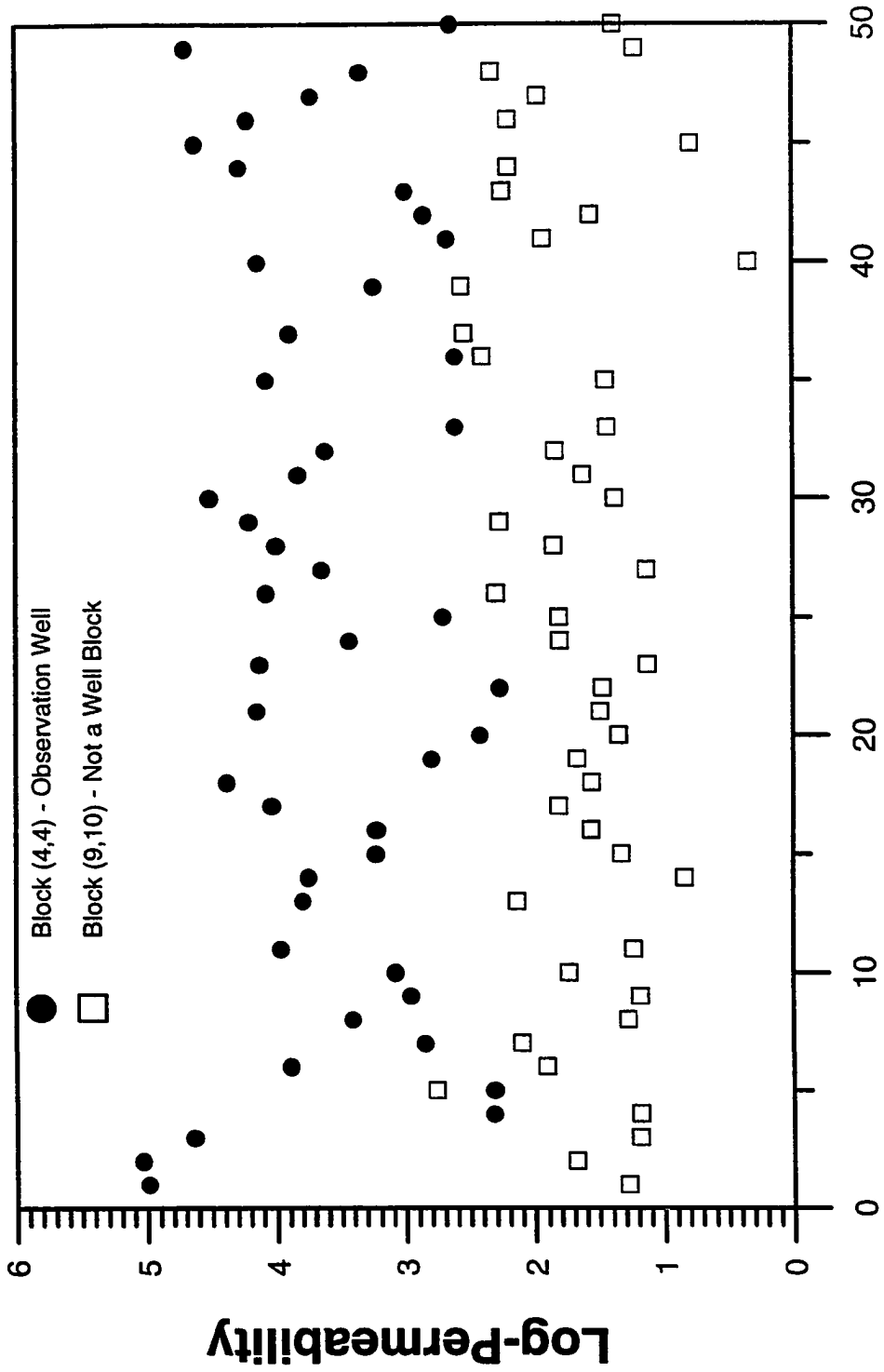

Fig. 4.15b - Log-Permeability Values During Sampling with Hybrid Monte Carlo Method (Variance = 1.0).

information. This can be seen by the very low value encountered for the a posteriori variance.

At the gridblock between the producing and interference well the pressure resolves quite well the uncertainty. However, a larger variation in the log-permeability values for this gridblock occurs during the sampling procedure since the corresponding standard deviation is 0.49 as compared to the standard deviation of 1.55×10^{-3} associated with log-permeability at producing well gridblock. This can be observed in Figs. 4.15a and 4.15b.

On the other hand, at the observation well gridblock (4,4), since the pressure data are not sufficient to significantly reduce the uncertainty in the permeability estimate, we can see that a large variation in the log-permeability values occurs during the sampling process. This is also the case for the gridblock located close to the corner of the reservoir, where the value of the a posteriori variance is 0.93.

In relation to the number of images accepted, a total number of 47 distinct realizations were generated in 50 perturbations resulting in an acceptance rate of 94%. As mentioned before, using this perturbation approach, we expected to increase the number of states accepted without increasing the correlation between accepted distinct states, or realizations. To check this hypothesis, we calculated the objective function experimental variogram. Fig. 4.16 shows the experimental variogram obtained. As we can conclude from Fig. 4.16, the realizations generated are not correlated. Regarding computational time, this method required approximately 75 hours to attempt 50 perturbations.

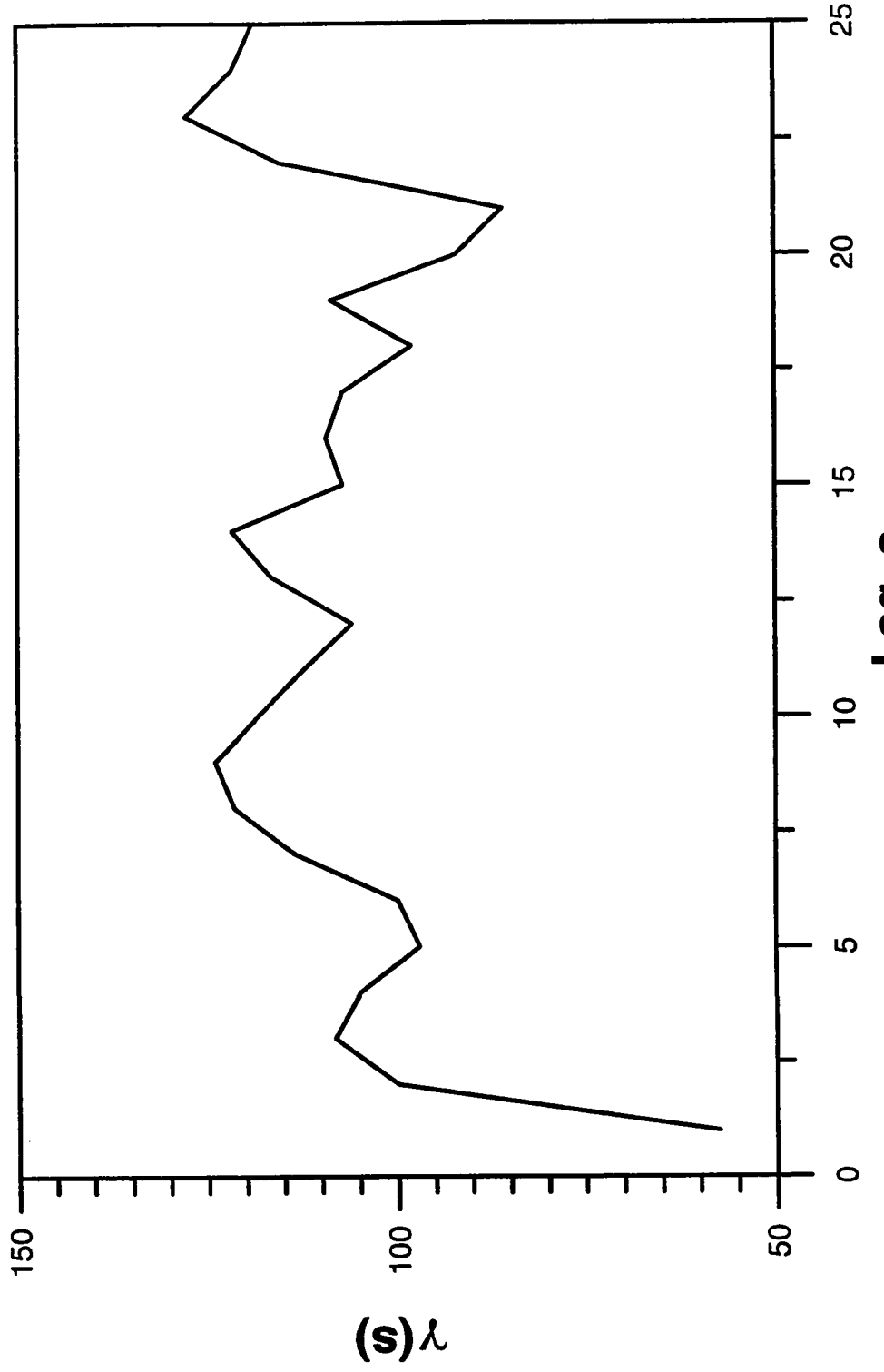

Fig. 4.16 - Experimental Variogram for the Objective Function of the Chain of Realizations Obtained Using Hybrid Monte Carlo Method (Variance = 1.0).

4.6.2 A Demonstration Using the Horowitz Variant

Here, our goal is to decrease the computational time spent when estimating the gradient by using an "approximate" Hamiltonian during the dynamical transitions. In this case, we consider only the first term of the right hand side of Eq. 4.42 for the gradient. As described in a previous section, a numerical method to solve the Hamiltonian equations is not necessary and an analytical solution is used given the values of variables m_p and p for a given total "time" (see Eq. 4.47 and 4.48). However, to maintain the validity of the algorithm, the Metropolis criterion applied at the end of a dynamical transition has to be calculated using the correct Hamiltonian expressed by Eq. 4.50. The correct Hamiltonian is a function of the model m while the leapfrog iterations during the dynamical steps are performed using the analytical solution expressed by Eqs. 4.47 and 4.48 which is a function of the reparameterized model m_p. Thus, it is necessary to convert between the model m and the reparameterized model m_p after each dynamical step in order to apply the Metropolis criteria. This is done by means of the following equations

$$m_p = U^T m \Leftrightarrow m = U m_p, \tag{4.51}$$

$$m_{p,\infty} = U^T m_\infty \Leftrightarrow m_\infty = U m_{p,\infty}, \tag{4.52}$$

where U is the matrix containing the eigenvectors obtained from the spectral decomposition of the a posteriori covariance matrix C_{MP}.

When using this approach, a limitation for the total time of each trajectory, due to the fact the a "wrong" Hamiltonian was considered, was encountered. To be able to

explore the potential of the method of avoiding random walks, i.e. be able of increase the trajectory length without increasing the rejection rate, we combined the use of an "approximate" Hamiltonian with the variant suggested by Horowitz. This idea is illustrated here by an example in which all data and information are the same as for the example listed in the previous subsection.

The first step of the implementation consists in finding a good estimate for the size of the trajectory, i.e. for the total "time" $\Delta\tau_i = L\eta\sqrt{\lambda_i}$ to be used in the analytical solution. Note that here, L is fixed and equal to one. The values of λ_i are obtained from the spectral decomposition of the a posteriori covariance matrix C_{MP}. Fig. 4.17 helps to obtain this estimate. It shows data on how the error in total energy varies with η. These data were obtained by performing a simulation in which only the first term of the right hand side of Eq. 4.42 is considered during the dynamical transitions, but the criterion to accept or reject is based on the Hamiltonian described in Eq. 4.50, using various values of η. Trajectories of length $L = 1$ were used. As can be seen, for η greater than about 3×10^{-3}, the leapfrog method becomes unstable, and very large errors result, which would lead to a very high rejection rate if such a value of η were used. The value of $\eta = 2\times10^{-3}$ was used for the sampling phase and the total "time", $\Delta\tau_i$, needed in the analytical solution is calculated by $\Delta\tau_i = \eta\sqrt{\lambda_i}$.

As suggested by Mackenzie[40], some random variation was considered for the value used for $\Delta\tau$. This was performed by randomly generating values for η from an interval of arbitrarily chosen values around $\eta = 2\times10^{-3}$, i.e., it could vary uniformly from 1.0×10^{-3} to

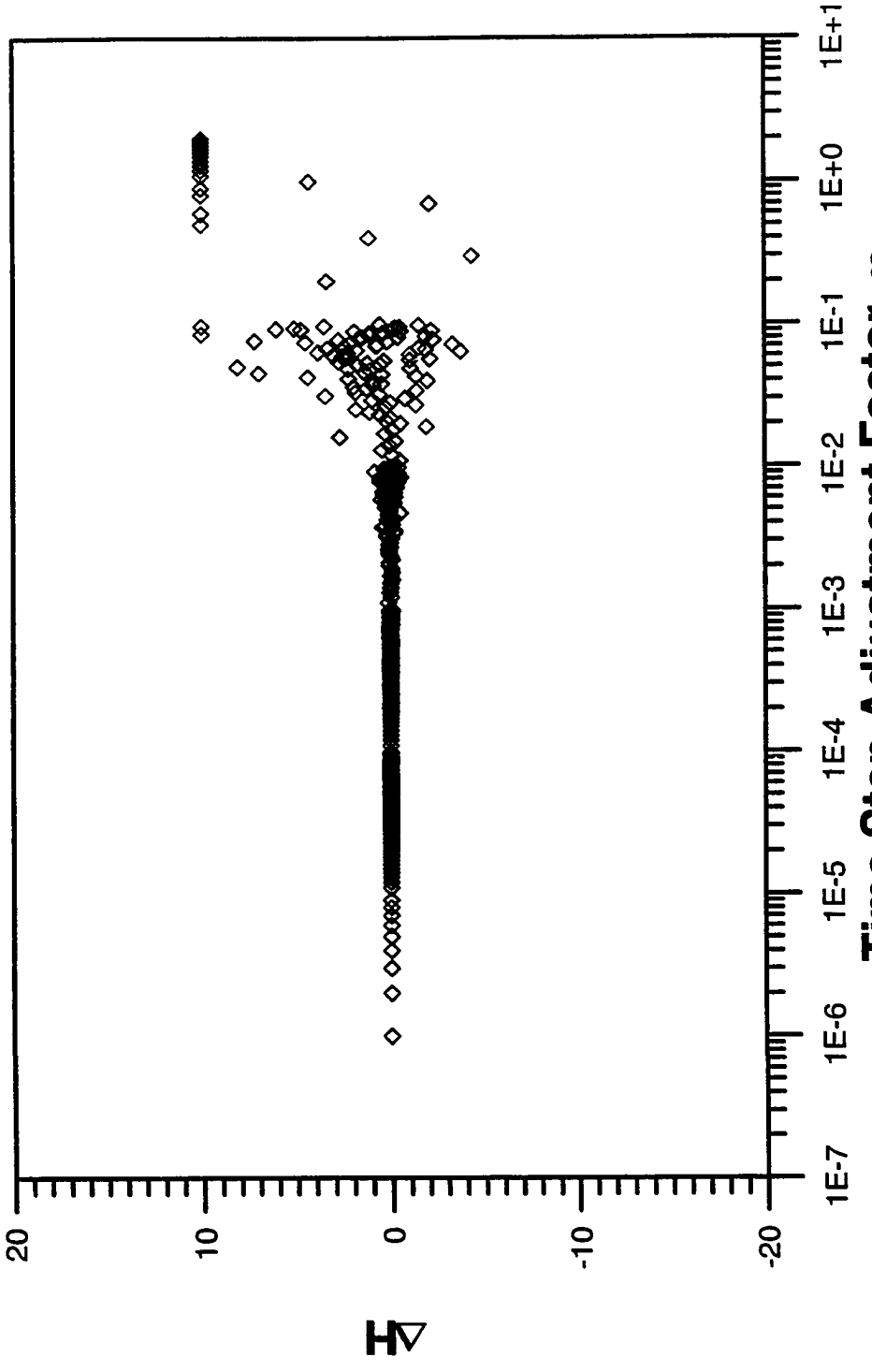

Fig. 4.17 - Error in Trajectory versus Time Step Adjustment Factor when Sampling with Horowitz Method (Variance = 1.0).

3.0×10^{-3}. This may be useful since the analytical solution is expressed in sines and cosines terms which can result in some periodicities that could interfere with ergodicity.

In this implementation, another tunable parameter is the value of the parameter α that is involved in the updates of the momentum variables. Again, based on the information in Fig. 4.17 and on our previous suggested criteria that $L\eta \approx 1$, we assumed an hypothetical value for L equal to 1000. A value for L = 1000 suggests, based on criterion previously described in section 4.4, that we use $\alpha = 0.99$. With this value for α we expect to change our momentum variables vector after an average of 1000 hybrid Monte Carlo updates.

Figs. 4.18 presents a plot of the value of "objective function" of accepted states versus the perturbation number. A corresponding plot of the pressure misfit is shown in Fig. 4.19. Figs. 4.18 and 4.19 allow us to identify the end of the transient period. In this example, after approximately 2000 perturbations, the value of the objective function oscillates around a constant value while in the pressure misfit plot, we see samples that give pressure misfit lower than 0.15 psi (standard deviation for the pressure measurement errors.) As described previously these facts suggest that the transient period has ended and the stationary distribution is being sampled.

Fig. 4.20 describes the variation in the log of permeability in three gridblock locations (see gridblock locations (4,4), (8,8) and (2,14) circled in Fig. 2.4.6) versus iteration number. The a posteriori variance for the (4,4), (8,8) and (2,14) gridblocks were 0.59, 2.4×10^{-6} and 0.93, respectively. The maximum a posteriori estimates of log-permeability for the same gridblocks were 3.69, 2.48 and 3.51, respectively.

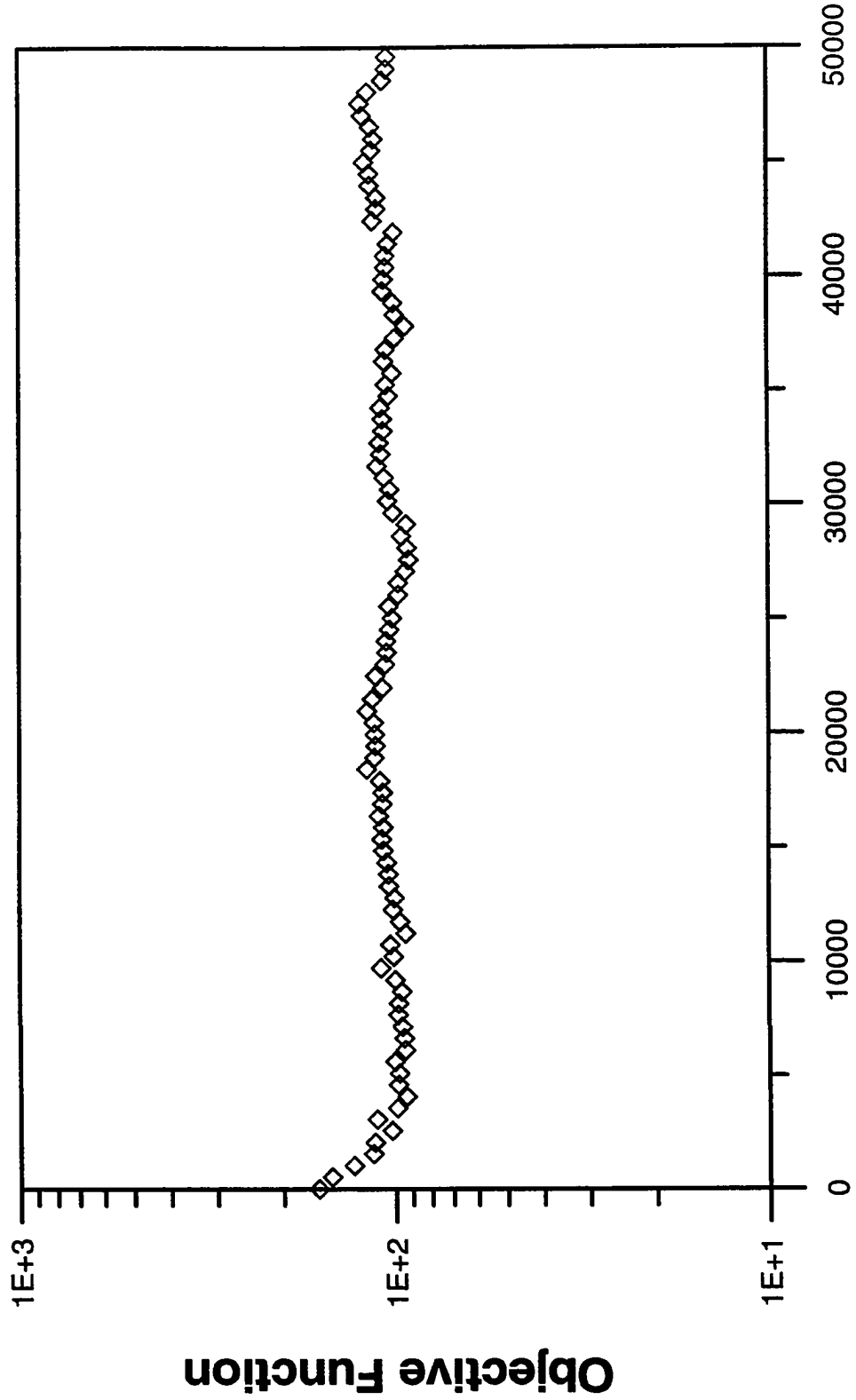

Fig. 4.18 - Values of the Objective Function for the Horowitz Method (Variance = 1).

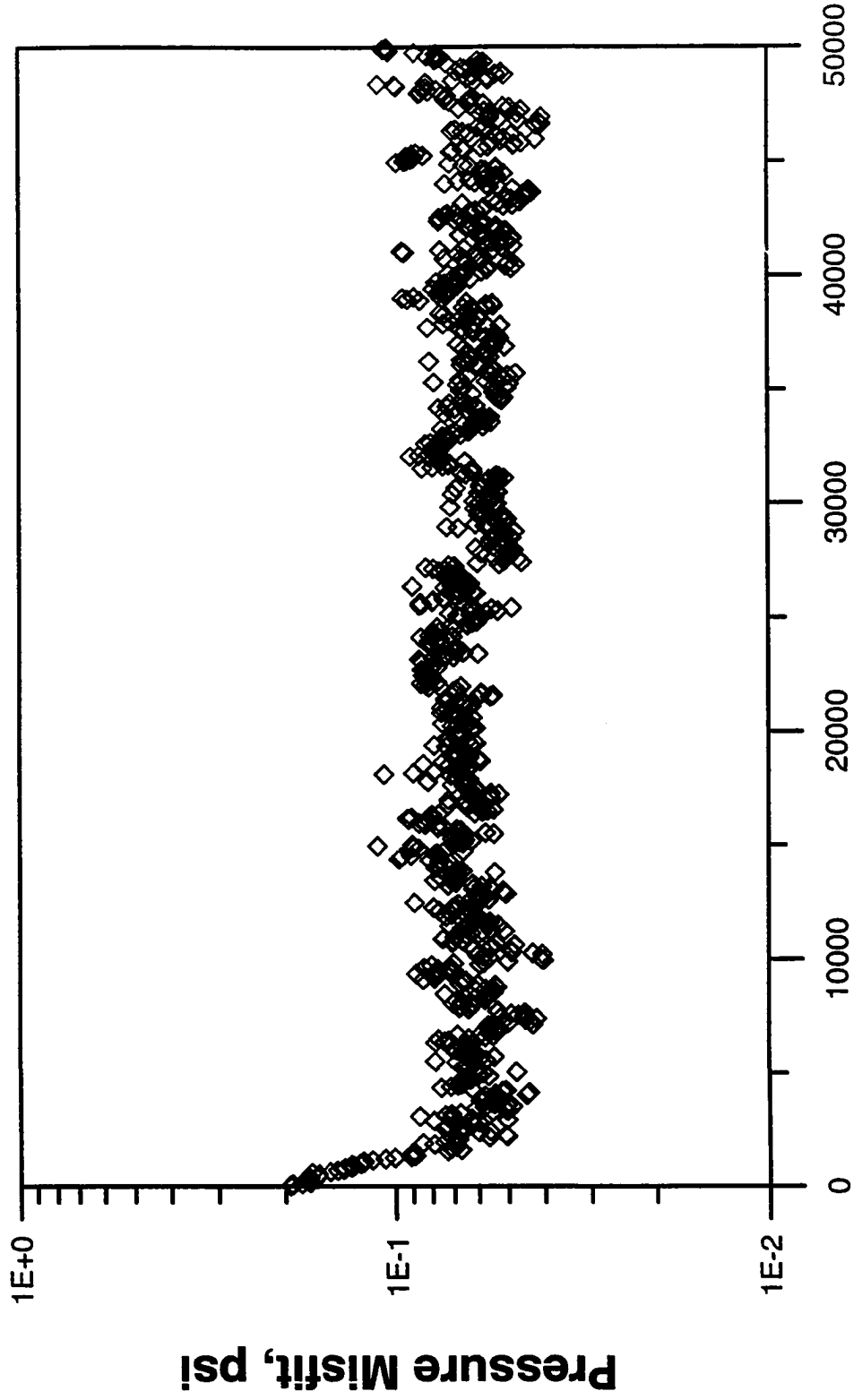

Fig. 4.19 - Behavior of Pressure Misfit during Sampling with Horowitz Method (Variance = 1).

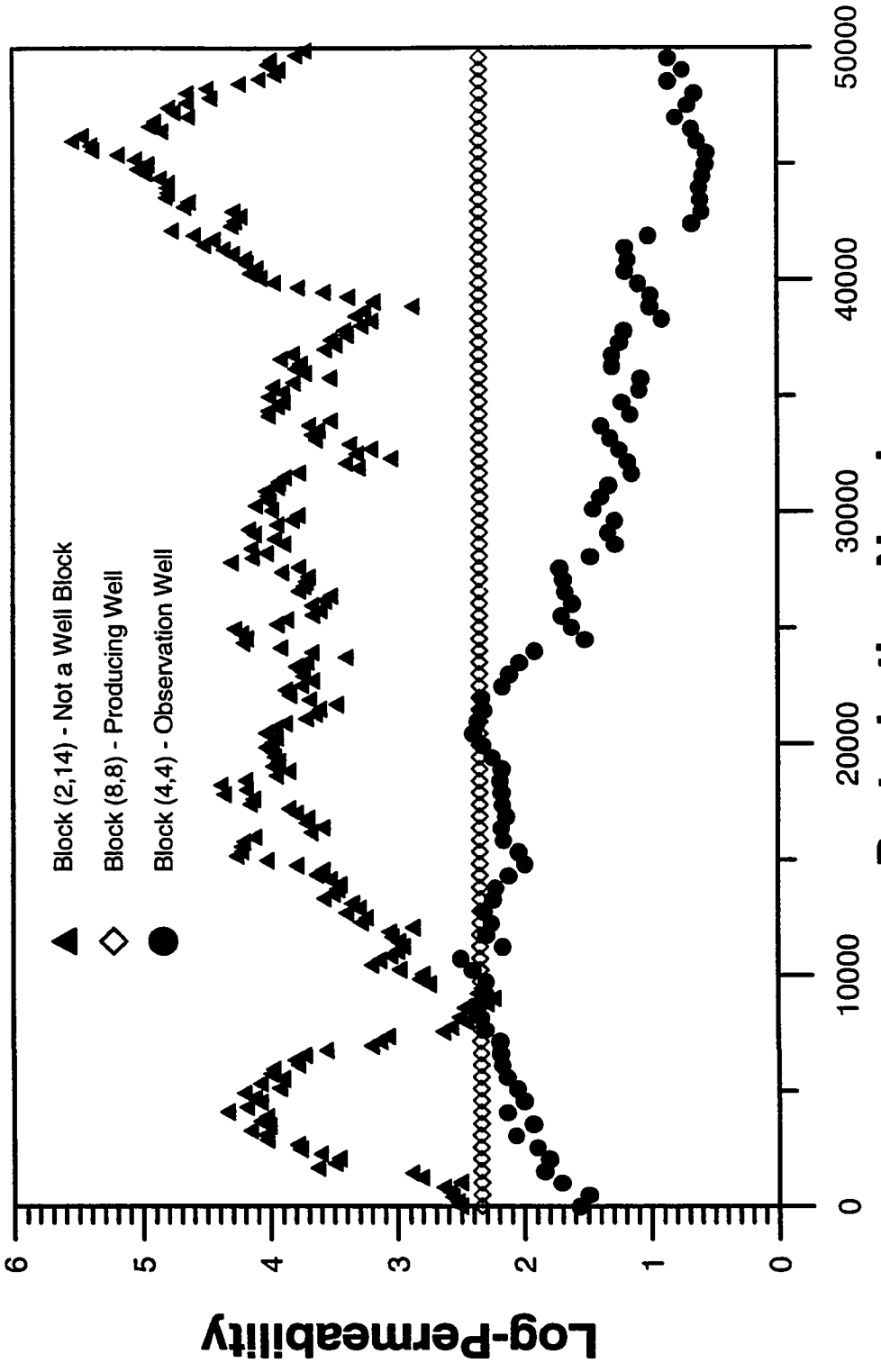

Fig. 4.20 - Log-Permeability Values During Sampling with Horowitz Method (Variance = 1.0).

Considering the behavior of the sampling of permeability at each block, we can see the dominant influence that the a posteriori variances, through the stepsize calculation, have in the realizations generated during the hybrid Monte Carlo procedure. The sampled value for the producing well gridblock location stays almost constant and equal to the value encountered for the maximum a posteriori estimate for that gridblock through the whole simulation. This reflects the fact that at the producing well location the uncertainty is considerably reduced when incorporating pressure data together with the prior model information. This can be seen by the very low value encountered for the a posteriori variance.

At the observation well gridblock analyzed, since the pressure data are not sufficient to reduce the uncertainty in the permeability estimate as well as in the producing well gridblock, we can see that a large variation in the log-permeability estimates occurs during the sampling process. This is also the case for the gridblock located close to the corner of the reservoir.

In relation to the number of images accepted, a total number of 49000 distinct realizations in a total of 50000 perturbations resulting in an acceptance rate of 98%. To check the correlation between the realizations, we calculated the objective function experimental variogram. Fig. 4.21 shows the experimental variogram obtained. As we can conclude from Fig. 4.21, the realizations generated are slightly correlated to lags of approximately 3000. If, we retain only every 3000th realization we would be left with only 16 independent realizations. The correlation between samples are directly related to the fact that the Horowitz method uses $L = 1$. The use of only one leapfrog step is not sufficient to produce an uncorrelated position variable at the end of the dynamical

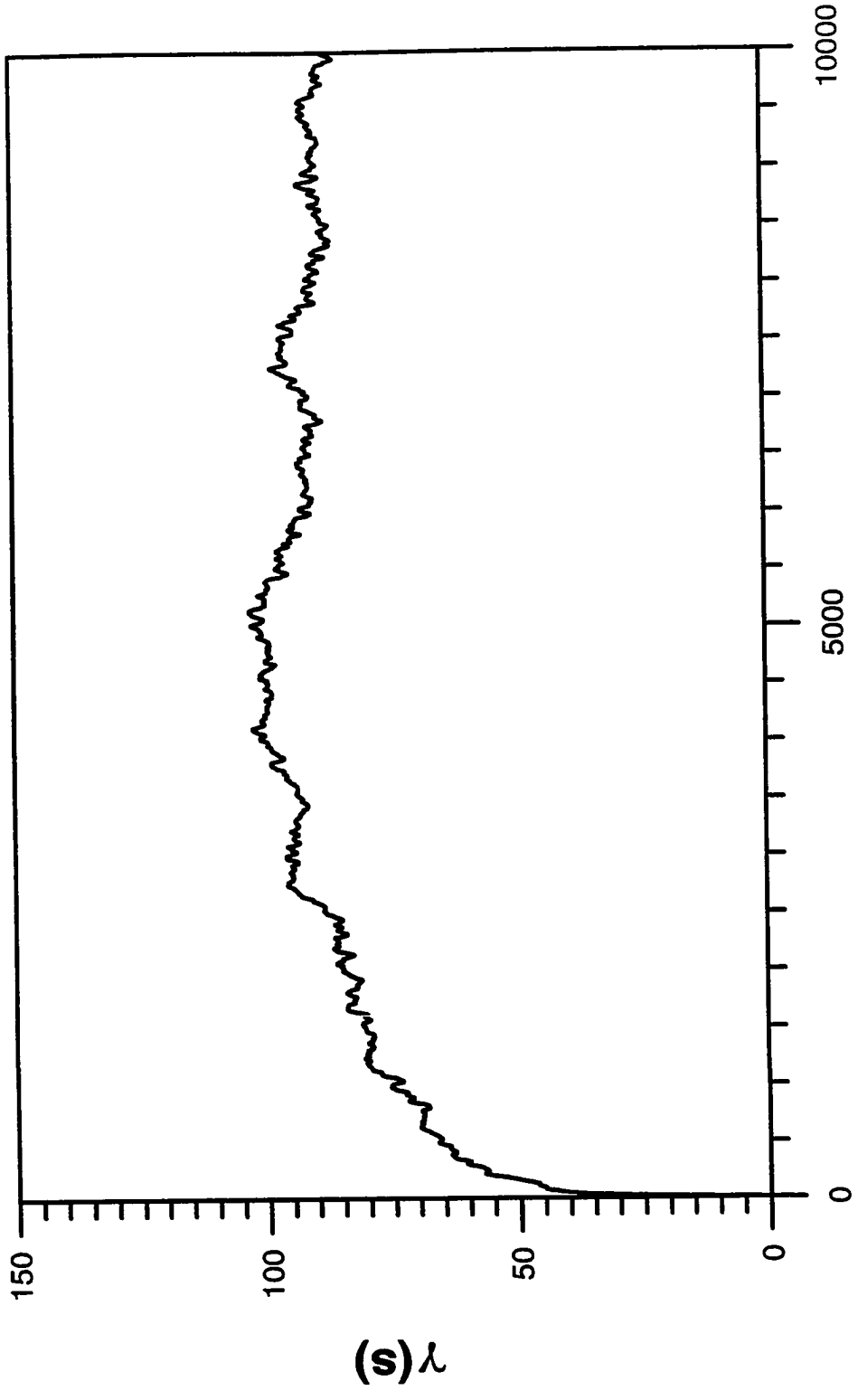

Fig. 4.21 - Experimental Variogram for the Objective Function of the Chain of Realizations Obtained with Horowitz Method (Variance = 1.0).

transition. To perform 50000 perturbations, this method required 14 hours of computational time.

In the Horowitz variant, since we were able to obtain an immense reduction in time spent per leapfrog iteration, the resulting ratios of total computational time spent per number of independent images were lower for this application when compared with the standard hybrid Monte Carlo method. However, if we wish to obtain a set of uncorrelated realizations, we must thin the total set of states obtained with the Horowitz method.

CHAPTER V

COMPARISON OF METHODS FOR SAMPLING THE A POSTERIORI PROBABILITY DISTRIBUTION

Here, we summarize and compare the performance of the previously described methods for sampling the a posteriori probability distribution function. This will be done by analyzing the behavior of each method when applied to the two synthetic examples considered previously in Chapter II and to one simpler example for a univariate probability density function.

5.1 Example 1 - Prior Mean, Covariance and Pressure Data Case

The first examples on which we base our comparison use the same data and information as for Examples 1 (prior variance = 0.25) and 2 (prior variance = 1.0) listed in Chapter II. The results obtained by applying the conventional Markov chain methods to these examples are given in detail in Chapter II. In Chapter III, we described the performance when simulated annealing was used to initialize the Markov chain Monte Carlo method. The results obtained by applying the hybrid Monte Carlo method to these examples are given in detail in Chapter IV.

Table 5.1 summarizes the results obtained when applying conventional Markov chain methods, simulated annealing to initialize the Markov chain Monte Carlo method and the hybrid Monte Carlo method to Examples 1 (prior variance = 0.25) and 2 (prior variance = 1.0) listed in Chapter II. The acceptance ratio refers to the number of transitions accepted divided by the total number of proposed transitions.

In Chapter II we outlined several different perturbation methods for use in Markov chain Monte Carlo simulations and gave corresponding acceptance criteria that will result in a correct sampling of the a posteriori probability distribution for the log-permeability field, given a prior mean and variogram model and pressure data. By carefully choosing the shape and amplitude of the proposed perturbations, very high acceptance ratios can be obtained when sampling with the Markov chain Monte Carlo method. We showed that by incorporating the sensitivity information from a flow simulator into the sampling distribution for proposed transitions, the probability of acceptance of the transitions can be increased significantly compared to methods that do not use this information.

In the examples analyzed here, comparing the Markov chain Monte Carlo (McMC) methods developed in Chapter II we found that, for the moderately nonlinear example, local perturbations based on the a posteriori covariance matrix presented the highest acceptance ratio when compared with global perturbations. However, images "close together" in the Markov chain are highly correlated; the fourth column of Table 5.1. For 50000 iterations, after thinning the images, the method based on local perturbations generated 31 independent images versus 200 for the global method. On the other hand, when the problem was more nonlinear, the local method produced about 14 independent realizations while the method based on global method produced about 8. Since each of the

Table 5.1 - Summary of Results of Methods for Sampling

Method	**Prior Variance**	**Acceptance Ratio (%)**	**cpu time/ ind. image**	**ind.image/ accept. trans. (%)**
McMC w/local perturbations	0.25	75	25 min.	0.08
McMC w/local perturbations	1.0	51	56 min.	0.05
McMC w/global perturbations	0.25	0.4	4 min.	100
McMC w/global perturbations	1.0	0.01	2 hours	100
Simulated Annealing to Initialize McMC	0.25	45	29min.	0.1
Simulated Annealing to Initialize McMC	1.0	24	1.1 hours	0.07
Conventional Hybrid McMC	0.25	100	45 min.	100
Conventional Hybrid McMC	1.0	94	1.6 hours	100
Horowitz Hybrid McMC	1.0	98	50 min.	0.03

methods took approximately the same amount of time to try 50000 transitions, it appears that global perturbations are more efficient for linear and slightly nonlinear problems, while local perturbations are more efficient for the highly nonlinear problems. However, if we wish to obtain a set of uncorrelated realizations, we must thin the total set of states obtained with the local method.

In Chapter III, we used simulated annealing to initialize the Markov chain Monte Carlo method. As summarized in Table 5.1 the statistics obtained from the runs performed with this approach resemble the ones with McMC with local perturbations. The main difference to be considered is the resulting overhead in computational time for the calculation of the starting point of the Markov chain by simulated annealing. This is reflected by a decrease in the value of the acceptance ratio. In the procedures described in Chapter II, this starting point is considered to be previously known and is chosen to be the maximum a posteriori estimate obtained from a gradient method.

Although a high acceptance ratio is desirable, it is generally more important for the perturbations schemes to be efficient in producing independent realizations from a Markov chain Monte Carlo procedure for a given available computational time. For the Markov chain procedures developed in Chapter II this is accomplished when using global perturbations. However, as shown in Table 5.1, when the problem becomes highly nonlinear, the performance of the global method is degraded (note the decrease in the acceptance ratio and consequent increase in the computational time needed to generate one independent image.) This fact provided the main motivation to investigate the hybrid Monte Carlo method. For problems considered in Ref. 12, this method was found to perform better than conventional Markov chain Monte Carlo methods. We show later in

this chapter an example where the hybrid Monte Carlo method improve the sampling process.

As described in Chapter IV and in Table 5.1, by applying the conventional hybrid Monte Carlo method to sample the a posteriori distribution for our problem we were able to keep a high acceptance ratio without increasing the correlation between the images. As described in Table 5.1, for the moderately nonlinear case the time spent to generate an independent image for the conventional hybrid Monte Carlo is larger than with the McMC procedure with global perturbations based on sensitivity coefficients. However, for the strongly nonlinear problem, this behavior changes and the time spent per independent image needed by the conventional hybrid Monte Carlo method is less than that needed by the McMC with global perturbations method.

One other aspect that should be also taken into account when comparing these two methods is the behavior during the sampling process. For the strongly nonlinear case, when sampling with McMC with global perturbations, we found that with this type of perturbation, we sometimes reached a state m^i in the Markov chain such that a high number of perturbations proposed from m^i are rejected, before we can generate a distinct image beyond m^i. For the examples performed in this work, the conventional hybrid Monte Carlo procedure didn't exhibit this behavior. Our explanation is that the McMC algorithm can get "trapped" for a long time in a local minimum with a very low probability of escaping from the minimum at any particular transition. We can envision a combination of two factors that can cause this low probability of escape. One factor occurs if m^i is such that the denominator π_i on the acceptance criteria, $\alpha_{ij} = \pi_j q_{ji} / \pi_i q_{ij}$, is extremely large so that the probability of accepting almost any new state generated by a global perturbation is

extremely small. The other factor, is that q_{ji} in the numerator may be very small if the global perturbations are based on the a posteriori covariance matrix and the covariance provides a poor representation of the probability density.

As mentioned before, the need to run a flow simulation at each leapfrog step in the conventional hybrid Monte Carlo method makes the procedure very computationally intensive. This was the main motivation to investigate the use of an approximate Hamiltonian for the dynamical steps coupled with the Horowitz hybrid Monte Carlo procedure. This procedure greatly reduced the computational time spent during the dynamical transitions. The resulting samples however, were highly correlated and the images were thinned based on the experimental variogram of the objective function values. As depicted in Table 5.1, since the time per leapfrog step was considerably reduced, the amount of time to generate an independent image was smaller than those obtained with the conventional hybrid Monte Carlo. However, if we wish to obtain a set of uncorrelated samples, we must thin the total set of states obtained with the Horowitz method.

When considering the behavior of permeability samples at each block, we notice that similar features are present for all the methods of sampling. Figures 5.1a and 5.1b show log-permeability values at four gridblock positions (see locations depicted in Fig. 2.4.6) versus the perturbation number. These figures, repeated here, were obtained when using the Markov chain Monte Carlo method that used global perturbations based on the a posteriori covariance matrix for the case with variance of 0.25, but the following conclusions apply to all the methods. As before, the perturbation number in these figures refers to the number of transitions proposed, but only the values of accepted transitions are plotted. Regarding the areal location of the log-permeability gridblocks, we have a first

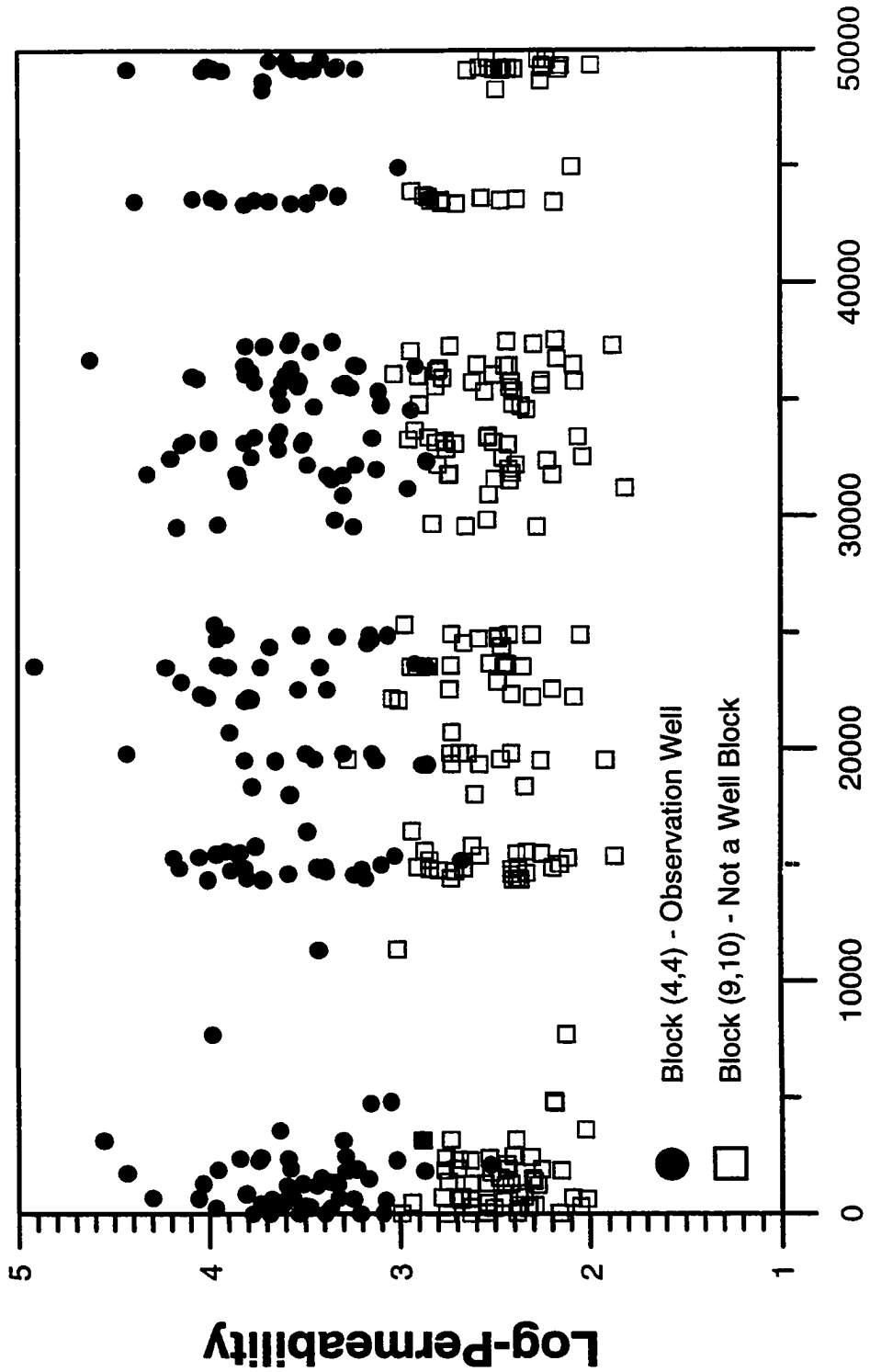

Fig. 5.1a - Log-Permeability Values During Sampling with Global Perturbations Based on the a Posteriori Covariance Matrix (Variance = 0.25).

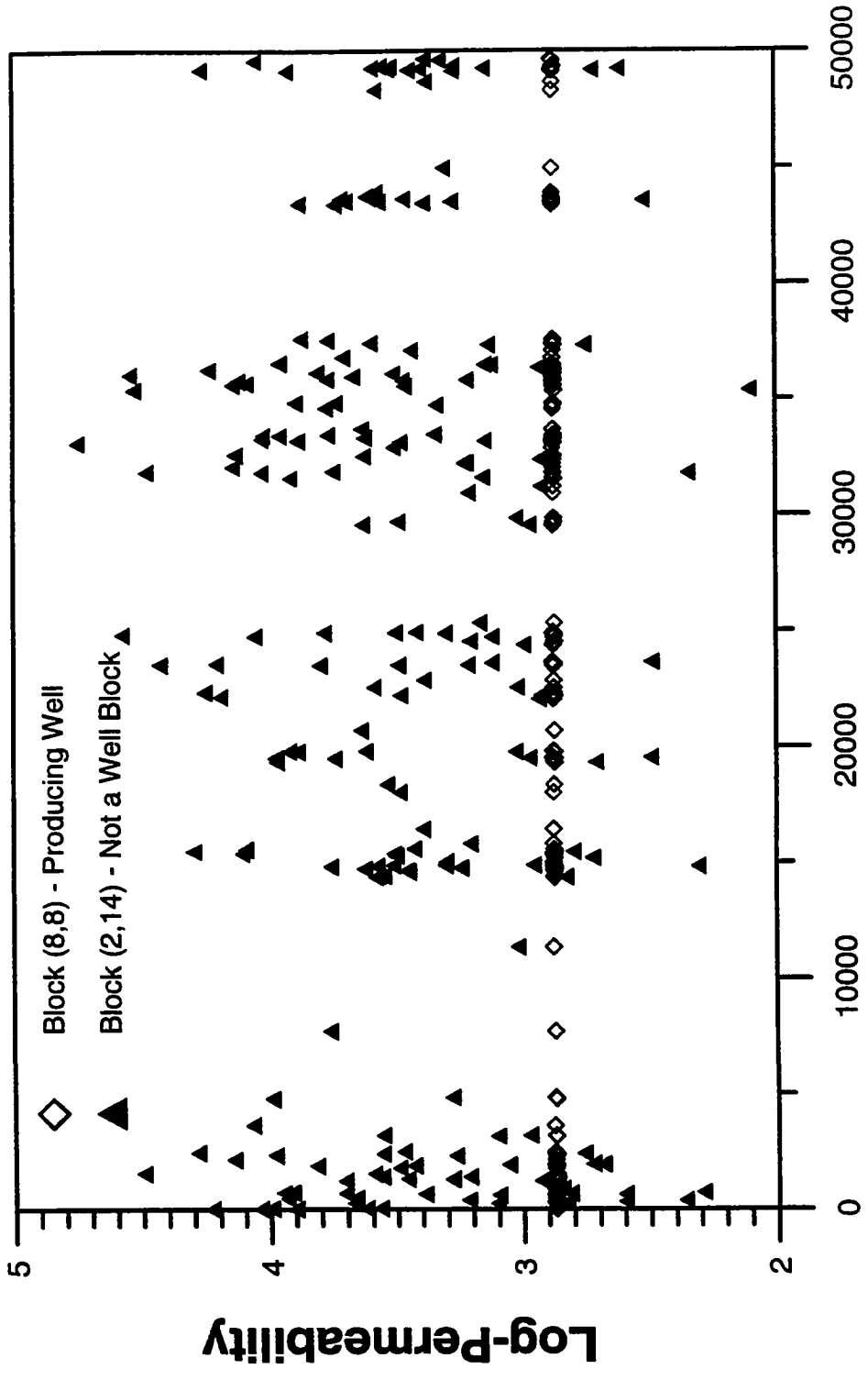

Fig. 5.1b - Log-Permeability Values During Sampling with Global Perturbations Based on the a Posteriori Covariance Matrix (Variance = 0.25).

location at one of the observation well gridblocks (X1), a second at the producing well gridblock (X2), a third location between the producing well gridblock and one observation well (X3) and a fourth almost at the reservoir boundary (X4). The sampled permeability value for the producing well gridblock location stays almost constant and equal to the value encountered for the maximum a posteriori estimate for that gridblock through the whole simulation. This reflects the fact that at the producing well location the uncertainty is considerably reduced when incorporating pressure data together with the prior model information.

On the other hand, at the observation well gridblock (X1), since the pressure data are not sufficient to reduce the uncertainty in the permeability estimate as well as in the producing well gridblock, we can see that a large variation in the log-permeability estimates during the sampling process occurs. This is also the case for the gridblock located close to the corner of the reservoir.

5.2 Example 2 - Synthetic Univariate Example

For this example, we assume that our goal is to sample the a posteriori probability density function given by

$$\pi(m) = a\exp\left[-\frac{(m-m_0)^2}{2\sigma_m^2} - \frac{(\cos(3m)-d)^2}{2\sigma_d^2}\right] \qquad (5.1)$$

where, we assume that $m_0 = 1.9$, $d = 0.8$, $\sigma_m^2 = 0.1$, $\sigma_d^2 = 0.01$, m is constrained to the interval [0,3] and $a \approx 4.394$ in order that the probability density integrate to one. For consistency with our previous discussions, we identify the Gaussian part of the probability,

$$\rho(m) = b\exp\left[-\frac{(m-m_0)^2}{2\sigma_m^2}\right] \qquad (5.2)$$

with a prior probability.

A plot of this function is shown in Fig. 5.2. The "a posteriori" probability density function given by Eq. 5.1 is shown in Fig. 5.3. Important piece of information about this probability function which will be used later is related to the value of probability associated with each of the two peaks. Integrating Eq. 5.1 in the interval associated with the primary peak, i.e. m = [1.70,2.05], gives a probability of 67% while integrating over the interval, (2.05,2.40], associated with the secondary peak, gives a probability of 33%.

This synthetic example can be viewed as an analog of our real problem of sampling the a posteriori probability distribution for permeability fields constrained to variogram and well-test data. The first term is a Gaussian distribution (with mean m_0 and variance σ_m^2) which represents the prior information, while the second term represents the pressure contribution to the probability density function. The consideration of the first term only is not troublesome since the resulting problem is linear. It is the second term that makes the problem non-linear and increases the difficulty of sampling in the model space.

For this problem, the objective function we want to minimize is

$$S(m) = \frac{(m-m_0)^2}{2\sigma_m^2} + \frac{(\cos(3m)-d)^2}{2\sigma_d^2}. \qquad (5.3)$$

Fig. 5.4 shows the plot of the objective function. With an initial guess of m = 2.0, application of Newton's method to the objective function of Eq. 5.3 converges to the minimum at m_∞ = 1.88. The value of m for which Eq. 5.3 is minimized is the maximum a posteriori model.

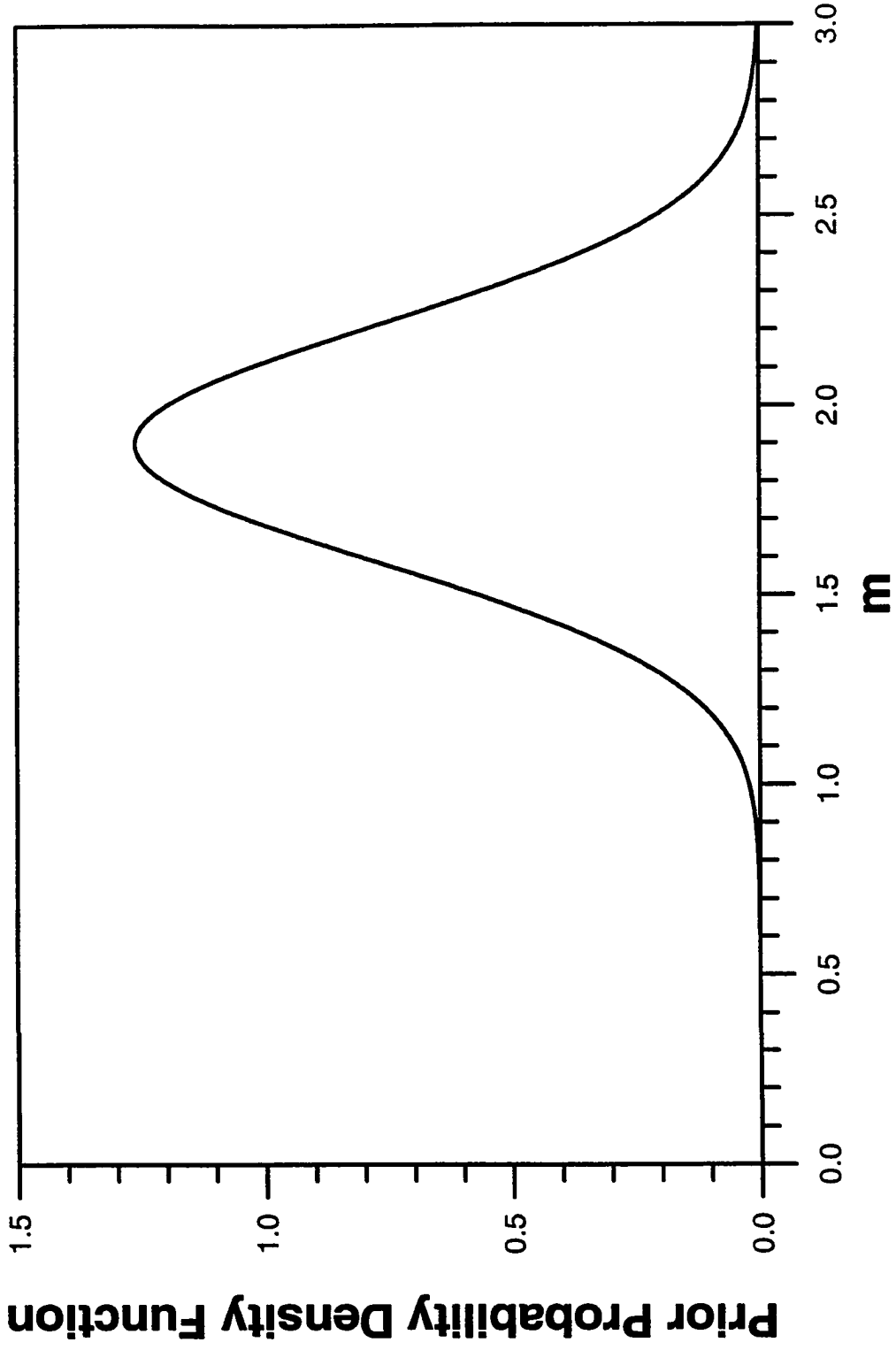

Fig 5.2 - Prior Probability Density Function (Example 2).

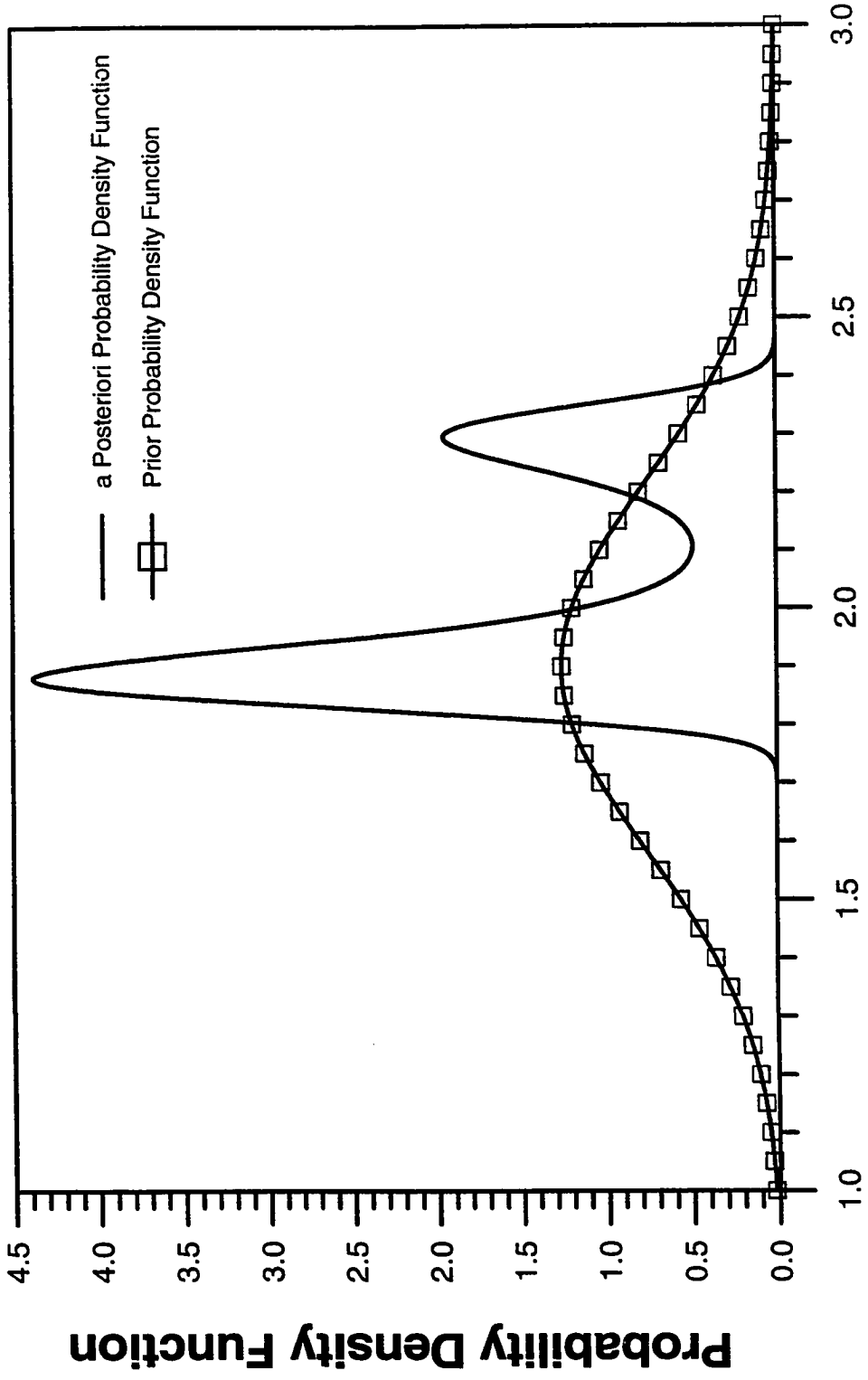

Fig. 5.3 - The a Posteriori and the Prior Probability Density Functions (Example 2).

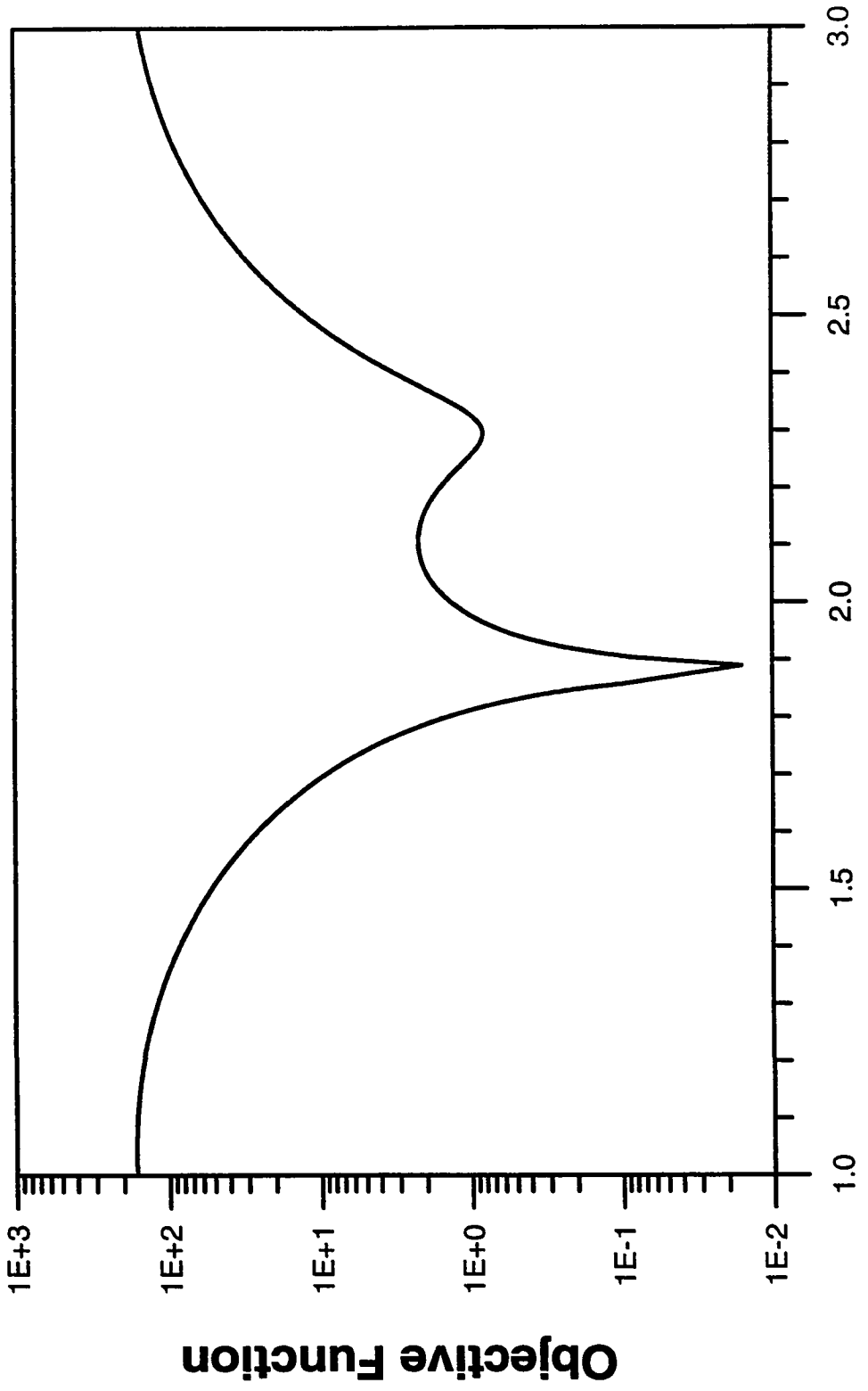

Fig. 5.4 - Objective Function Plot (Example 2).

It is possible to calculate a posteriori variance, in a direct analogy with the a posteriori covariance matrix. This can be done by using the following expression

$$\sigma^2_{C_{MP}} = (G_\infty(\sigma^2_d)^{-1}G_\infty + (\sigma^2_m)^{-1})^{-1}. \tag{5.4}$$

where G_∞ is $\left.\frac{d\cos(3m)}{dm}\right|_{m_\infty}$. In this example, G_∞= 1.79 and $\sigma^2_{C_{MP}} = 3.0\text{x}10^{-3}$.

Based on this a posteriori variance, we can write a linearized approximation to the probability density of Eq. 5.1:

$$\pi'(m) = c\exp\left[-\frac{1}{2}(m-m_\infty)^T\sigma^{-2}_{C_{MP}}(m-m_\infty)\right]. \tag{5.5}$$

Figure 5.5 compares the probability density function of Eq. 5.5 with the true a posteriori probability density function, Eq. 5.1.

Since one of the methods of interest is the hybrid Monte Carlo method, understanding the behavior of the Hamiltonian function is important. The Hamiltonian function for our problem can be expressed by the following equation

$$H(m,p) = E(m) + \frac{p^2}{2}. \tag{5.6}$$

Since

$$E(m) = \left[-\frac{(m-m_0)^2}{2\sigma^2_m} - \frac{(\cos(3m)-d)^2}{2\sigma^2_d}\right], \tag{5.7}$$

the Hamiltonian function becomes

$$H(m,p) = \left[-\frac{(m-m_0)^2}{2\sigma^2_m} - \frac{(\cos(3m)-d)^2}{2\sigma^2_d}\right] + \frac{p^2}{2}. \tag{5.8}$$

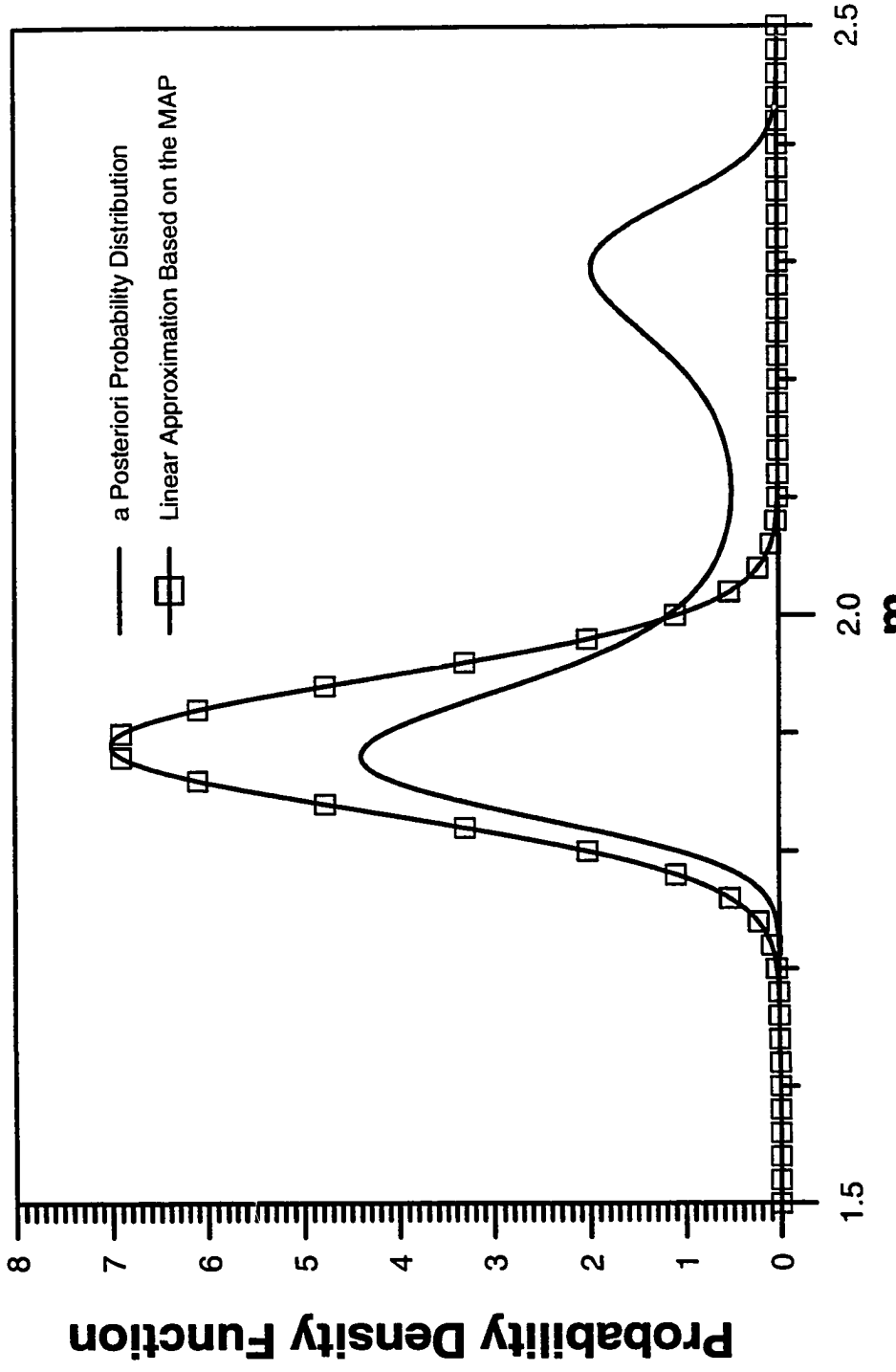

Fig. 5.5 - The a Posteriori Probability Distribution and the Linear Approximation to the a Posteriori Probability Distribution Based on the Sensitivity at the Maximum a Posteriori Estimate (Example 2).

Later, we show how the behavior of the Hamiltonian influences the performance of the hybrid Monte Carlo method.

5.2.1 Perturbations Based on the Maximum a Posteriori Estimate

Here, we apply the method described in Chapter II, where perturbations are based on the maximum a posteriori estimate (MAP) and a posteriori covariance matrix obtained from a gradient method. As noted previously, the maximum a posteriori estimate and a posteriori variance, respectively, are given by $m_\infty = 1.88$ and $\sigma^2_{CMP} = 3.0\text{x}10^{-3}$.

In this method a transition from state i to state j in the Markov chain Monte Carlo method is proposed by generating a random normal deviate Z^j and setting

$$m^j = m_\infty + \sigma_{C_{MP}} Z^j . \tag{5. 9}$$

The proposed transitions from the current state (m^i) to another state (m^j) denoted q_{ij} were based only on the probability of state j and can be expressed by

$$q_{ij} = c\exp\left(-\frac{1}{2}(m^j - m_\infty)^T \sigma^{-2}_{C_{MP}} (m^j - m_\infty)\right), \tag{5. 10}$$

Similarly, a transition from state j to state i, q_{ji} can be expressed by

$$q_{ji} = c\exp\left(-\frac{1}{2}(m^i - m_\infty)^T \sigma^{-2}_{C_{MP}} (m^i - m_\infty)\right). \tag{5. 11}$$

Hastings' acceptance criterion is now determined by

$$\frac{\pi_j q_{ji}}{\pi_i q_{ij}}, \tag{5. 12}$$

where π_i and π_j are calculated using Eq. 5.1 applied to states i and j.

Fig. 5.6 shows the resulting sampled values for the model variable m when using this method to sample the probability distribution described by Eq. 5.1. Note that the range of values is controlled by the probability distribution described by Eq. 5.5 and all samples are clustered about the primary peak at m_∞ 1.88. As shown in Fig. 5.6, no evidence of the secondary peak associated with the a posteriori probability distribution is seen during the sampling process.

This result can be easily understood when we consider the probability of proposing a transition to the second peak ($m \approx 2.3$) from the region of the maximum a posteriori estimate. Using Eq. 5.10 with $m = 2.3$ we have that

$$q(m = 2.3) = c\exp\left(-\frac{(2.3 - m_\infty)^2}{2\sigma^2_{C_{MP}}}\right),$$

and

$$q(m = 2.3) \approx 1.2\text{x}10^{-12}.$$

Clearly, it will be highly unlikely that states around $m = 2.3$ will be proposed in a chain of reasonable length. This occurs because the distribution used to propose states is much narrower than the true distribution (see Fig. 5.5). Moreover, although there is a very low probability for proposing a transition to the region of the secondary maximum, once it is proposed it will be accepted and the chain remains at this same state for a very large number of perturbations. In this case, although many transitions to states in the region of the maximum a posteriori estimate will be proposed, the acceptance criterion gives a very low probability of accepting those transitions. For this example, the Markov chain Monte

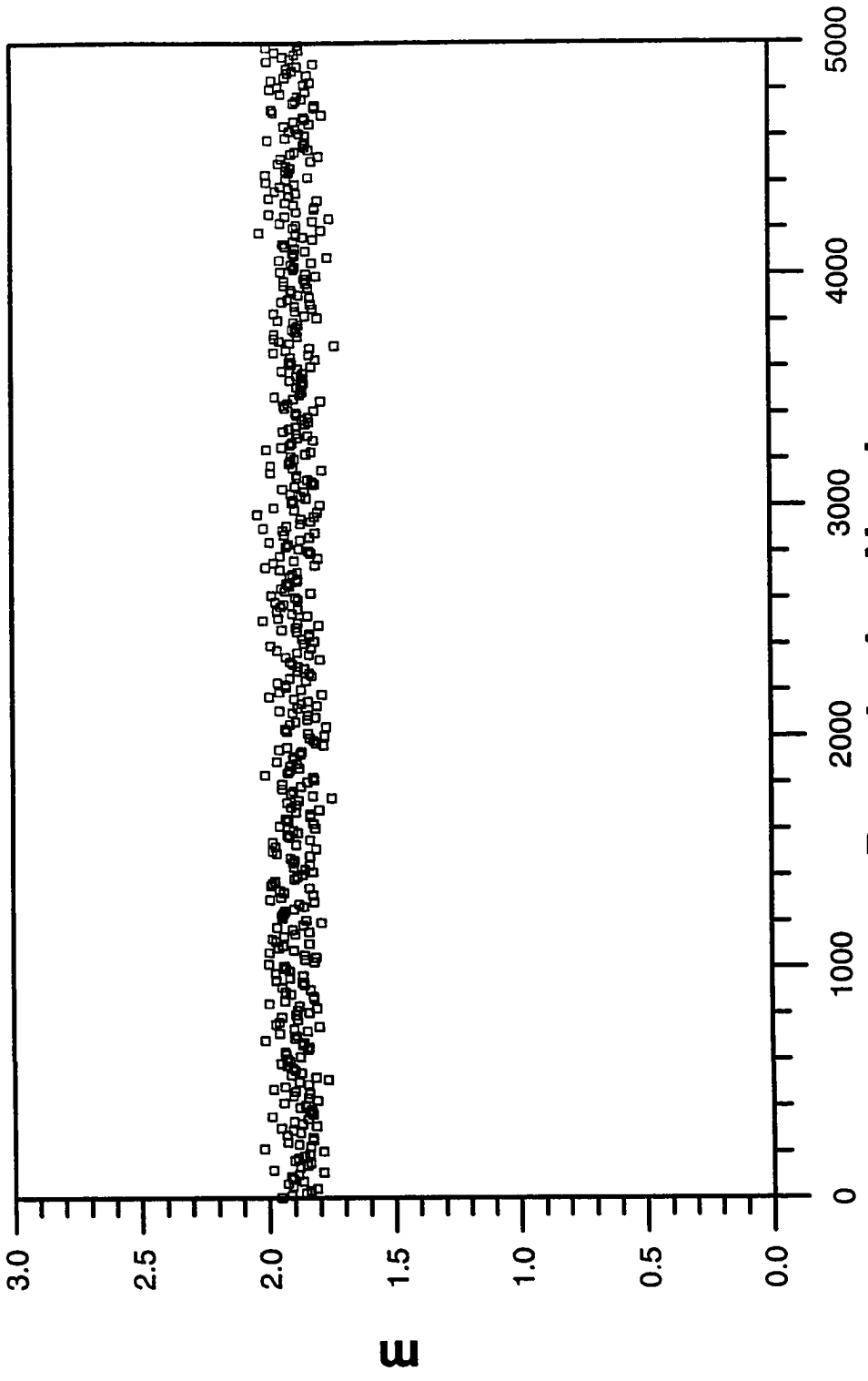

Fig. 5.6 - Sampling with Conventional Markov Chain Monte Carlo Algorithm (Example 2).

Carlo method is not able to "see" the secondary peak even though if we correctly sample the a posteriori probability density function about 33% of the samples should be in the interval (2.05,2.40] about the second peak. Thus, even though the behavior of the realizations shown in Fig. 5.6 suggests we have reached the stationary distribution of the Markov chain, we have not and have not obtained a correct sample.

5.2.2 The Hybrid Monte Carlo Implementation

We claimed earlier that the hybrid Monte Carlo method is superior to simple forms of the Markov chain Monte Carlo algorithm, because the hybrid method avoids random walk behavior. In this subsection, we will substantiate this claim with regard to the synthetic example previously described.

The heuristic procedure for determining stepsizes is based in the one described in Chapter IV. The stepsize was calculated by the following expression

$$\varepsilon = \eta \sigma_{C_{MP}}. \tag{5. 13}$$

As described in Chapter IV, it is necessary to find a good value for the stepsize adjustment factor, η, for use in the sampling phase. Fig. 5.7 shows data on how the error in total energy varies with η. These data were obtained by continuing the simulation from the initial state, using various values of η. Trajectories of length L = 10 were used here, but the results are similar for all longer trajectories. As can be seen, for η greater than about 1.0, the leapfrog method becomes unstable, and very large errors result, which

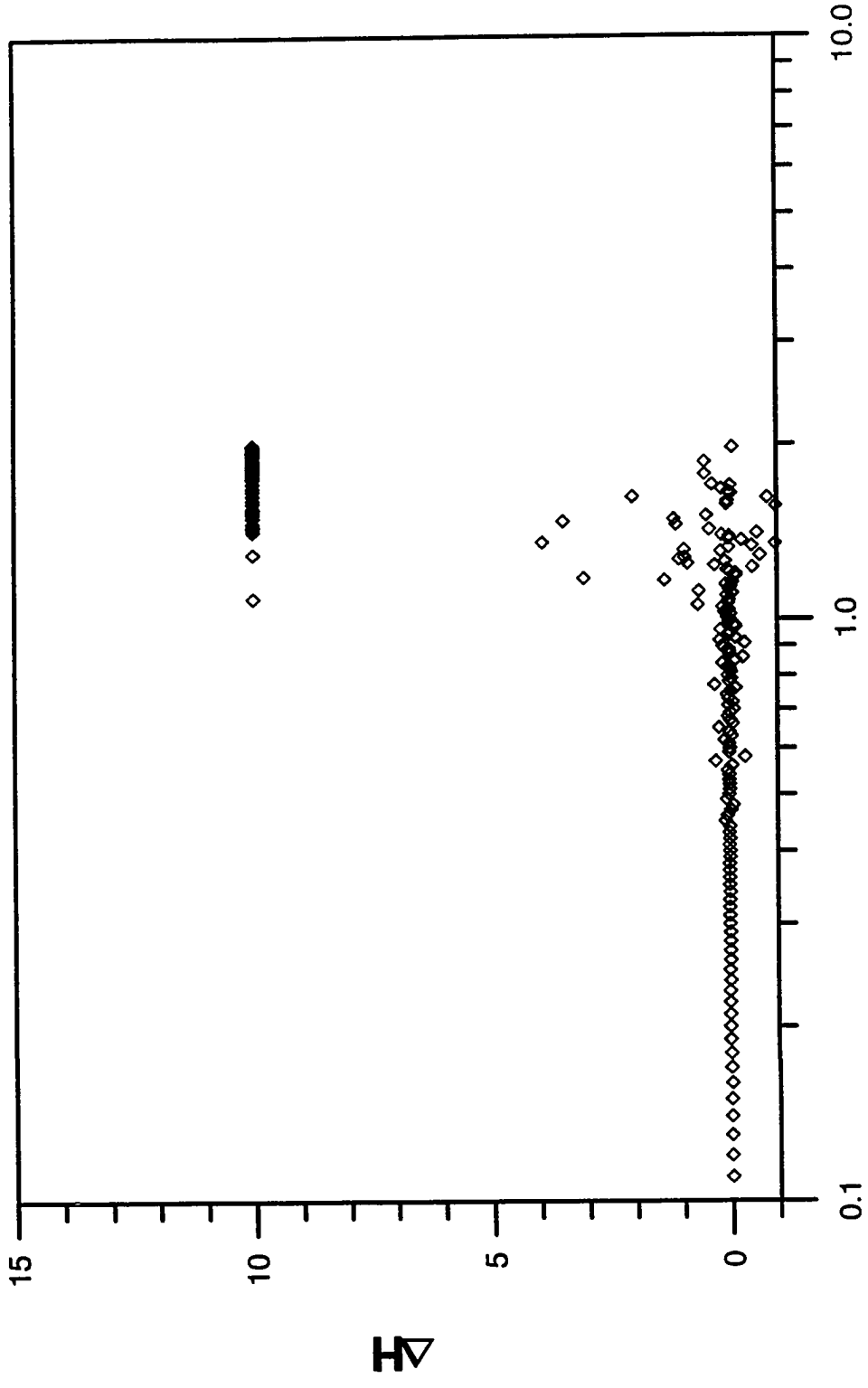

Fig. 5.7 - Error in Trajectory (L = 10, Example 2).

would lead to a very high rejection rate if such a value of η were used. The value of η = 1.0 was used for the sampling phase.

Fig. 5.8a shows the values for the model variable m obtained during the sampling procedure with the hybrid Monte Carlo method. Note that the two corresponding peaks of the a posteriori distribution (see Fig. 5.3) are sampled. By analysis of the samples, we found that 67% of the samples were in the interval [1.70,2.05] and 33% where in the interval (2.05,2.40]. This result corresponds exactly to the probability of a realization being in each of the respective interval. Recall these probability values were computed earlier by integrating the a posteriori probability density function over each interval. This fact can also be seen by examining the results of Fig. 5.8b where the values of the model variable m (position variable in the hybrid Monte Carlo method) sampled are plotted against the momentum variables p sampled.

The good performance of the hybrid Monte Carlo method is this example can be understood by analyzing the curves along which the Hamiltonian function is constant. In Fig. 5.9 each curve plotted describes the values of the momentum variable, p and of the position variable m along trajectories of 1000 leapfrog steps. Note it is possible to identify continuous paths from one region of probability to another. For example, if we have a sample (m_i,p_i) near $(m,p) = (m_\infty,0) = (1.88,0)$, then if we do a stochastic step (resample the probability function for p) and obtain say $p = 2.0$, then (m_i,p) may be a curve of constant H, $(H = c)$ which encompasses both peaks form. Since each time step of the dynamical step, generate a (m,p) value on this $H = c$ curve, at the completion of the dynamical step we may propose a new state (m_j,p) where m_j is near the second peak. Since generate a p =

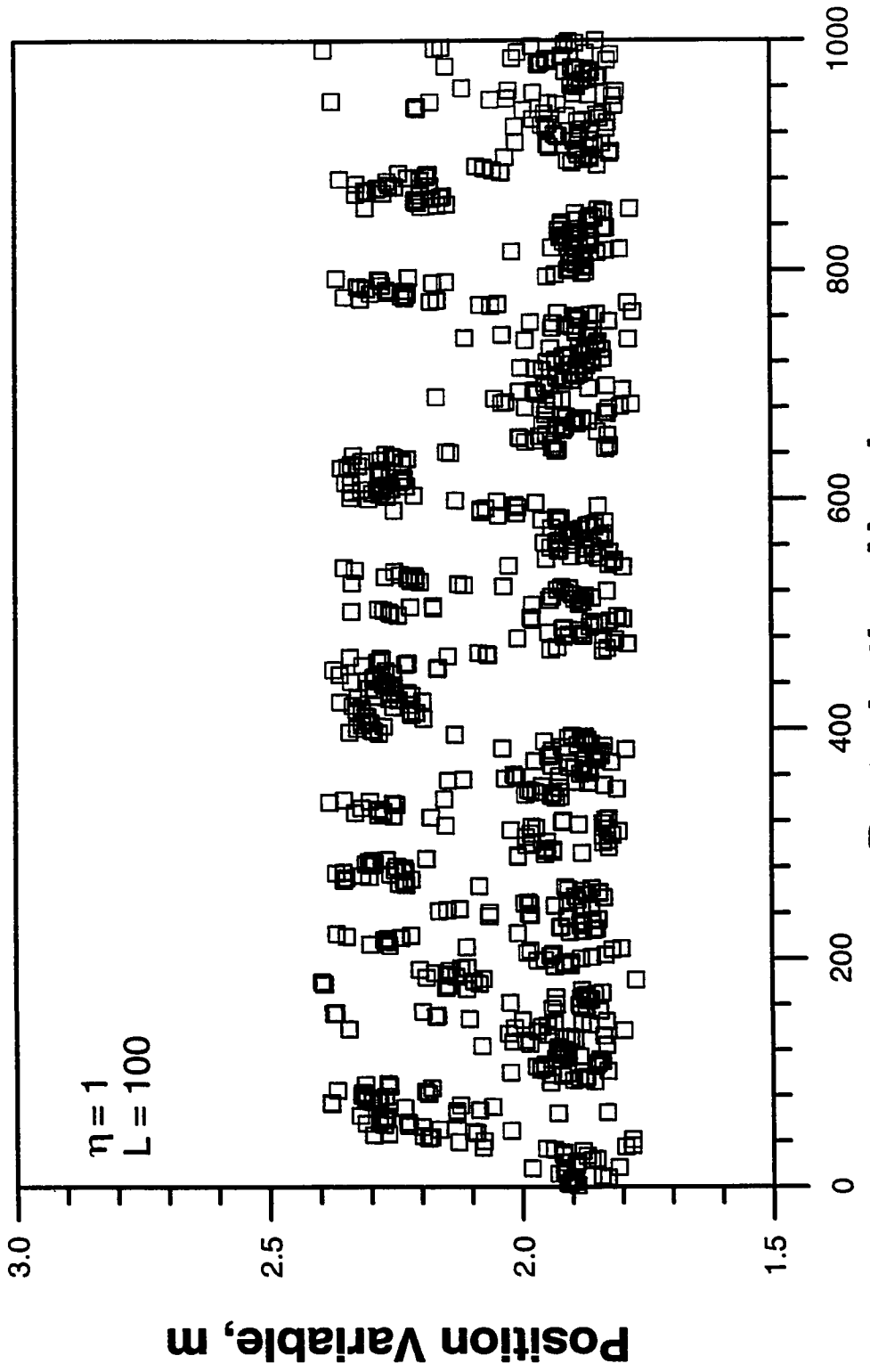

Fig. 5.8a - Behavior During Sampling with Hybrid Monte Carlo Method (Example 2).

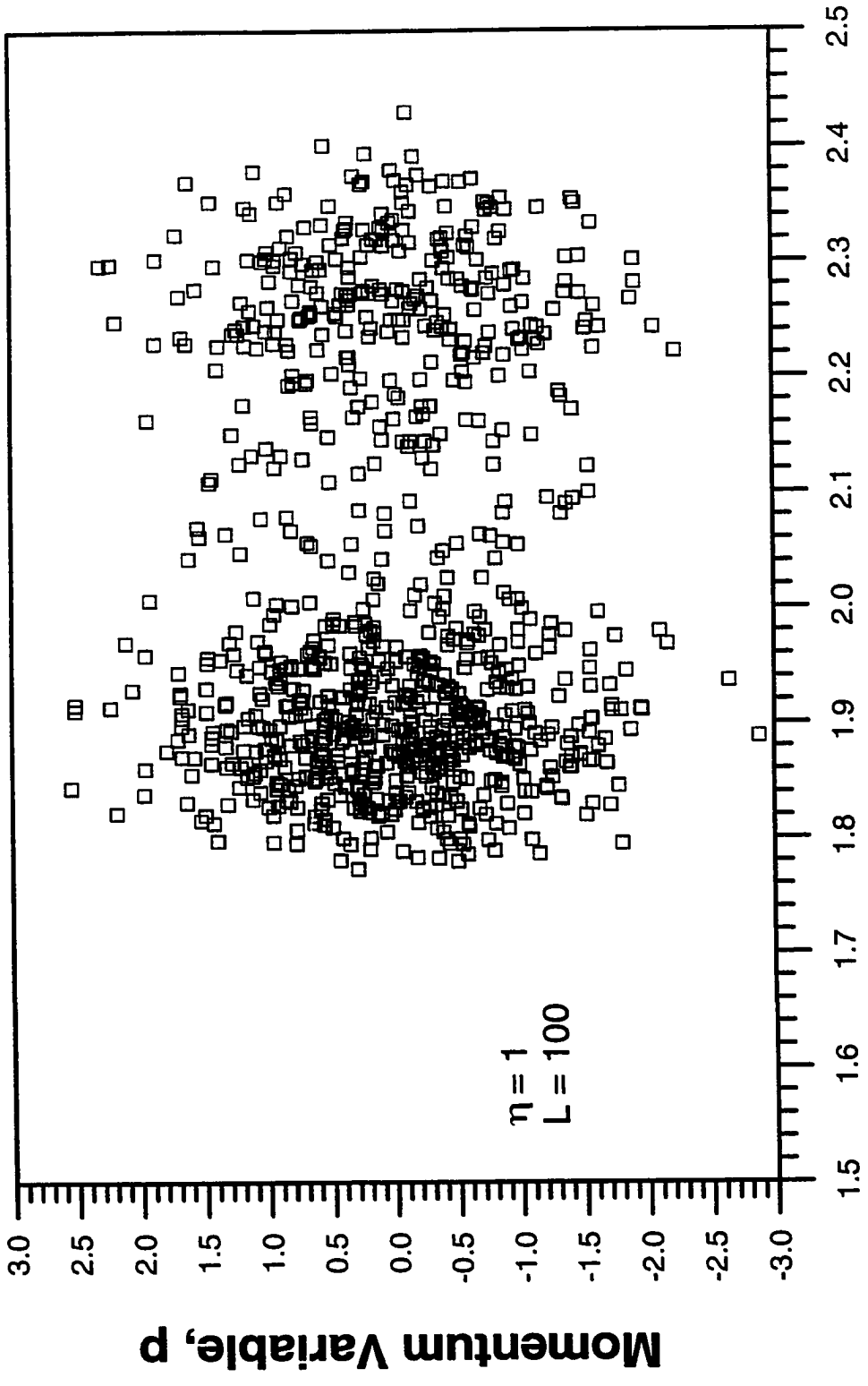

Fig. 5.8b - Behavior During Sampling with Hybrid Monte Carlo Method (Example 2).

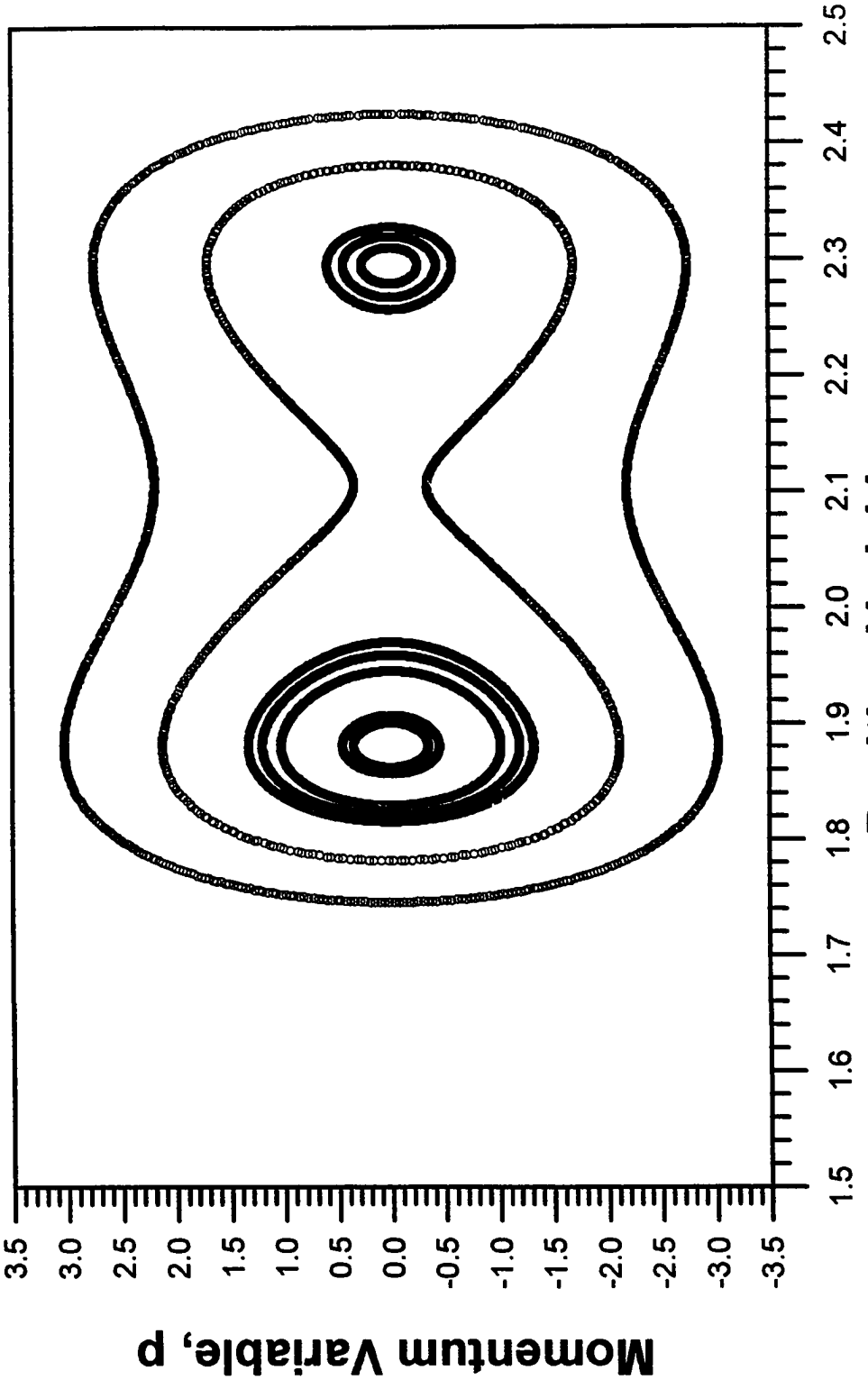

Fig. 5.9 - Equal Hamiltonian Function Contours (Example 2).

2.0 is not highly unlikely, it is clear that the hybrid method allows us to generate samples near both peaks.

When comparing both methods we see that, for this example, the conventional hybrid Monte Carlo method performed better than the Markov chain Monte Carlo procedure with perturbations based on the a posteriori variance. In this case, the assumption that the linear approximation based on the sensitivity at the MAP estimate is a good approximation for the a posteriori probability distribution fails resulting in a poor sampling of the a posteriori probability density function. However, as shown below, it is possible to modify the example slightly so that is difficult to sample the a posteriori probability density function with either method.

5.3 Example 3 - Synthetic Univariate Example

Similarly to Example 2, we assume that our goal is to sample the "a posteriori" probability density function given by

$$\pi(m) = a\exp\left[-\frac{(m-m_0)^2}{2\sigma_m^2} - \frac{(\cos(3m)-d)^2}{2\sigma_d^2}\right], \tag{5. 14}$$

where we assume that $m_0 = 1.9$, $d = 0.3$, $\sigma_m^2 = 0.1$, $\sigma_d^2 = 0.01$ and m is constrained to the interval [0,3]. Again, we identify the Gaussian part of the probability,

$$\rho(m) = b\exp\left[-\frac{(m-m_0)^2}{2\sigma_m^2}\right] \tag{5. 15}$$

with a prior probability.

Figure 5.10 shows the prior probability distribution of Eq. 5.15 and the a posteriori probability density function of Eq. 5.14. An important information about this probability function which will be used later is related to the value of probability associated with each of the two peaks. Integrating Eq. 5.14 in the interval associated with the primary peak, i.e. m = [1.50,1.80], gives a probability of 83% while integrating over the interval, [2.40,2.60], associated with the secondary peak, gives a probability of 17%.

For this problem, the objective function we want to minimize is

$$S(m) = \frac{(m - m_0)^2}{2\sigma_m^2} + \frac{(\cos(3m) - d)^2}{2\sigma_d^2}. \qquad (5.16)$$

With an initial guess of m = 1.5, application of Newton's method converges to the minimum at m_∞ = 1.67. The value of m for which Eq. 5.16 is minimized is the maximum a posteriori model.

It is possible to calculate a posteriori variance, in direct analogy with the a posteriori covariance matrix. This can be done by using the following expression

$$\sigma_{C_{MP}}^2 = (G_\infty(\sigma_d^2)^{-1}G_\infty + (\sigma_m^2)^{-1})^{-1}, \qquad (5.17)$$

where G_∞ is $\left.\frac{d\cos(3m)}{dm}\right|_{m_\infty}$. In this example, G_∞= 2.87 and $\sigma_{C_{MP}}^2$ = 1.2x10^{-3}.

Based on this a posteriori variance, we can write a linearized approximation to the probability density of Eq. 5.14:

$$\pi'(m) = c\exp\left[-\frac{1}{2}(m - m_\infty)^T \sigma_{C_{MP}}^{-2}(m - m_\infty)\right]. \qquad (5.18)$$

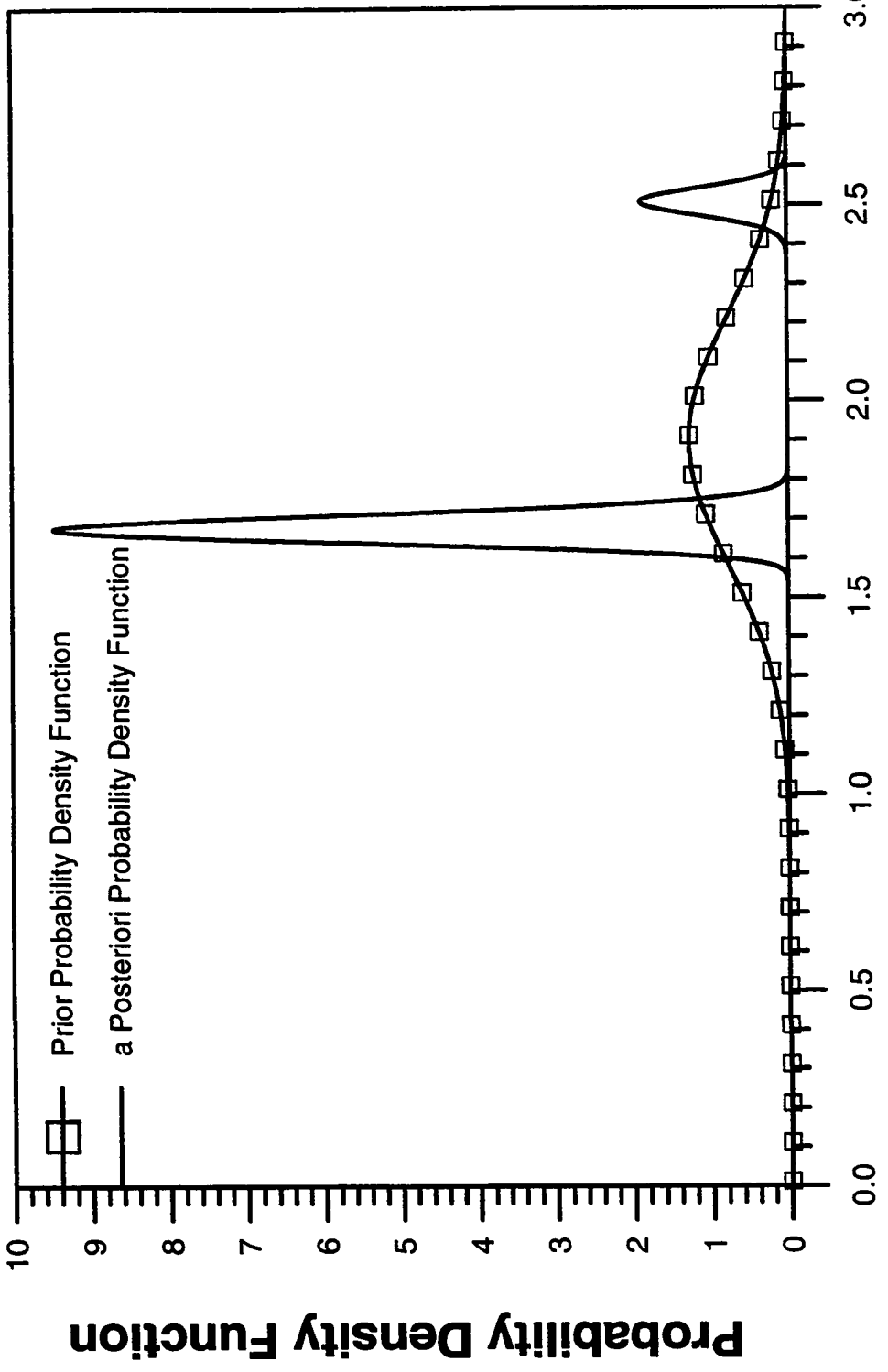

Fig. 5.10 - The a Posteriori and the Prior Probability Density Functions (Example 3).

Figure 5.11 compares the probability density function of Eq. 5.18 with the true a posteriori probability density function, Eq. 5.14.

Since one of the methods of interest is the hybrid Monte Carlo method, understanding the behavior of the Hamiltonian function is important. The Hamiltonian function for our problem can be expressed by the following equation

$$H(m,p)=\left[-\frac{(m-m_0)^2}{2\sigma_m^2}-\frac{(\cos(3m)-d)^2}{2\sigma_d^2}\right]+\frac{p^2}{2}. \qquad (5.19)$$

Later, we show how the behavior of the Hamiltonian influences the performance of the hybrid Monte Carlo method.

5.3.1 Perturbations Based on the Maximum a Posteriori Estimate

Here, we apply the method described in Chapter II, where perturbations are based on the maximum a posteriori estimate (MAP) and a posteriori covariance matrix obtained from a gradient method. As noted previously, the maximum a posteriori estimate and a posteriori variance, respectively, are given by $m_\infty = 1.67$ and $\sigma^2_{CMP} = 1.2\text{x}10^{-3}$.

In this method a transition from state i to state j in the Markov chain Monte Carlo method is proposed by generating a random normal deviate Z^j and setting

$$m^j = m_\infty + \sigma_{C_{MP}} Z^j . \qquad (5.20)$$

The proposed transitions from the current state (m^i) to another state (m^j) denoted q_{ij} were based only on the probability of state j and can be expressed by

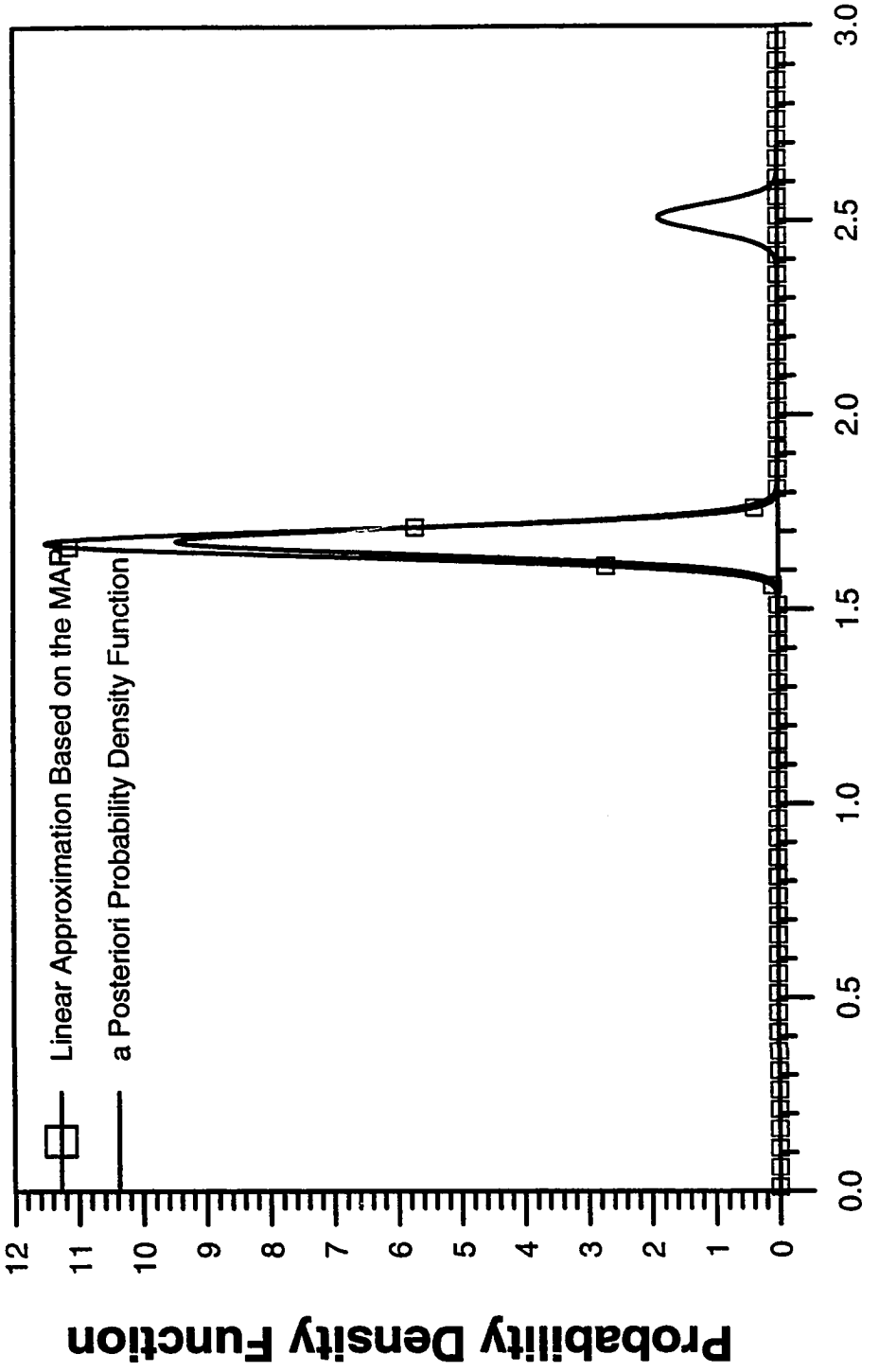

Fig. 5.11 - The a Posteriori Probability Distribution and the Linear Approximation to the a Posteriori Probability Distribution Based on the Sensitivity at the Maximum a Posteriori Estimate (Example 3).

$$q_{ij} = c\exp\left(-\frac{1}{2}(m^j - m_\infty)^T \sigma_{C_{MP}}^{-2}(m^j - m_\infty)\right). \qquad (5.21)$$

Similarly, a transition from state j to state i, q_{ji} can be expressed by

$$q_{ji} = c\exp\left(-\frac{1}{2}(m^i - m_\infty)^T \sigma_{C_{MP}}^{-2}(m^i - m_\infty)\right). \qquad (5.22)$$

Hastings' acceptance criterion is now determined by

$$\frac{\pi_j q_{ji}}{\pi_i q_{ij}}, \qquad (5.23)$$

where π_i and π_j are calculated using Eq. 5.14 applied to states i and j.

Fig. 5.12 shows the resulting sampled values for the model variable m when using this method to sample the probability distribution described by Eq. 5.14. Note that the range of values is controlled by the probability distribution described by Eq. 5.18 and all samples are clustered about the primary peak at $m_\infty = 1.67$. As shown in Fig. 5.12, no evidence of the secondary peak associated with the a posteriori probability distribution is seen during the sampling process.

Similarly to Example 2 we can calculate the probability of proposing a transition to the second peak ($m \approx 2.5$) from the region of the maximum a posteriori estimate. Using Eq. 5.21 with $m = 2.5$ we have that

$$q(m = 2.5) = c\exp\left(-\frac{(2.5 - m_\infty)^2}{2\sigma_{C_{MP}}^2}\right) = 3.22\text{x}10^{-124}.$$

Clearly, states around $m = 2.5$ will almost never be proposed with this perturbation mechanism. This occurs because the distribution used to propose states is much narrower

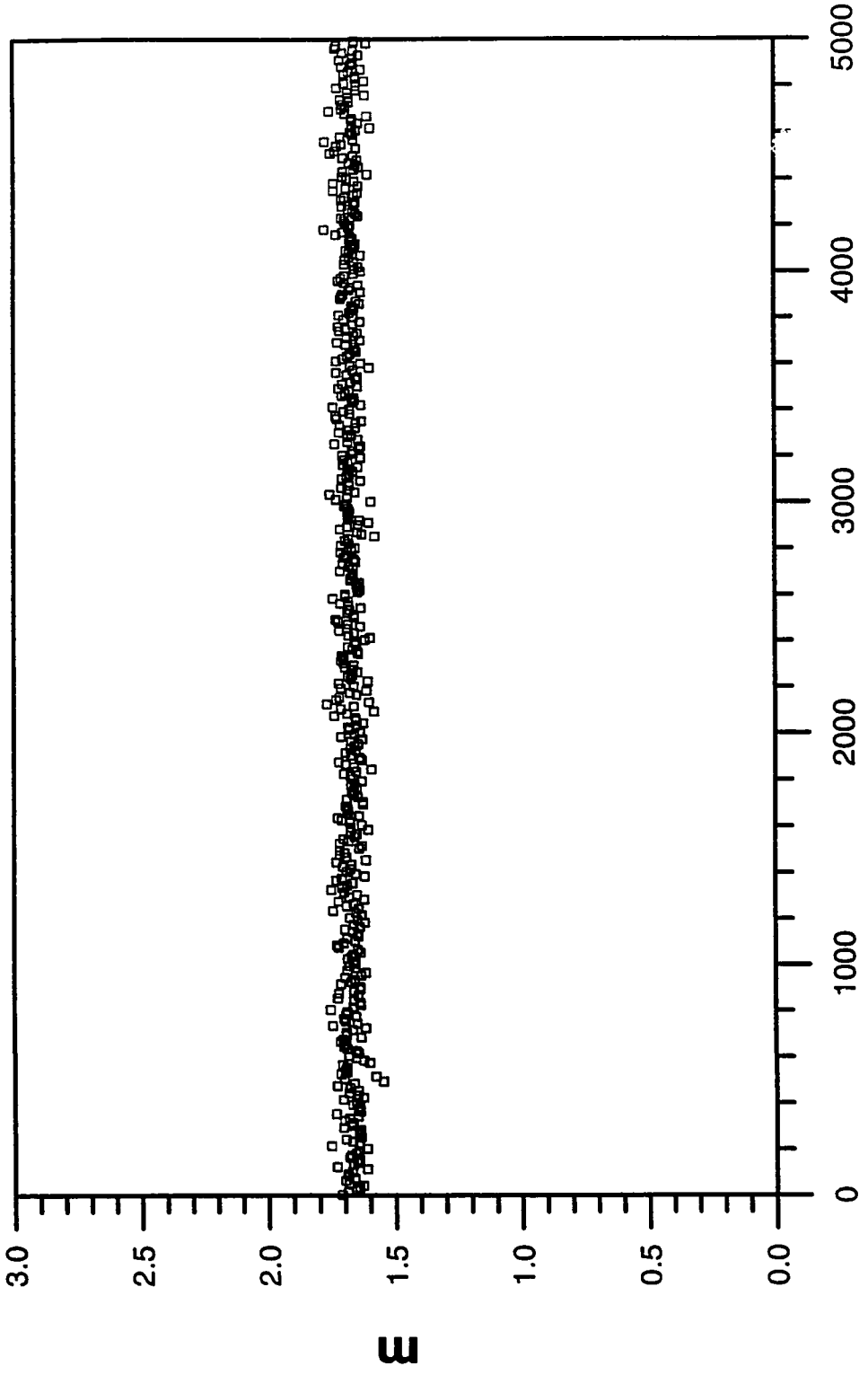

Fig. 5.12 - Sampling with Markov Chain Monte Carlo Algorithm (Example 3).

than the true distribution (see Fig. 5.11). For this example, the Markov chain Monte Carlo method is not able to "see" the secondary peak, even though if we correctly sample the a posteriori probability density function, about 17% of the samples should be in the interval [2.40,2.60] about the second peak.

5.3.2 The Hybrid Monte Carlo Implementation

Here, we apply the hybrid Monte Carlo method described in Chapter IV. The heuristic procedure for determining stepsizes is based in the one described in Chapter IV. The stepsize was calculated by the following expression

$$\varepsilon = \eta \sigma_{C_{MP}}. \tag{5.24}$$

Following the procedure described in Chapter IV a value of η equal to 1.0 showed to be optimum. For values higher than that, the leapfrog method becomes unstable, and very large errors result, which would lead to a very high rejection rate if such a value of η were used. The value of $\eta = 1.0$ was used for the sampling phase.

Fig. 5.13a shows the values for the model variable *m* obtained during the sampling procedure with the hybrid Monte Carlo method. Note that no evidence of the secondary peak associated with the a posteriori probability distribution is seen during the sampling process. Fig. 5.13b shows the values of the model variable *m* obtained as samples plotted against the associated values of momentum variables *p*.

In this example, the performance of the hybrid Monte Carlo method is degraded and can be understood by analyzing the curves along which the Hamiltonian function is

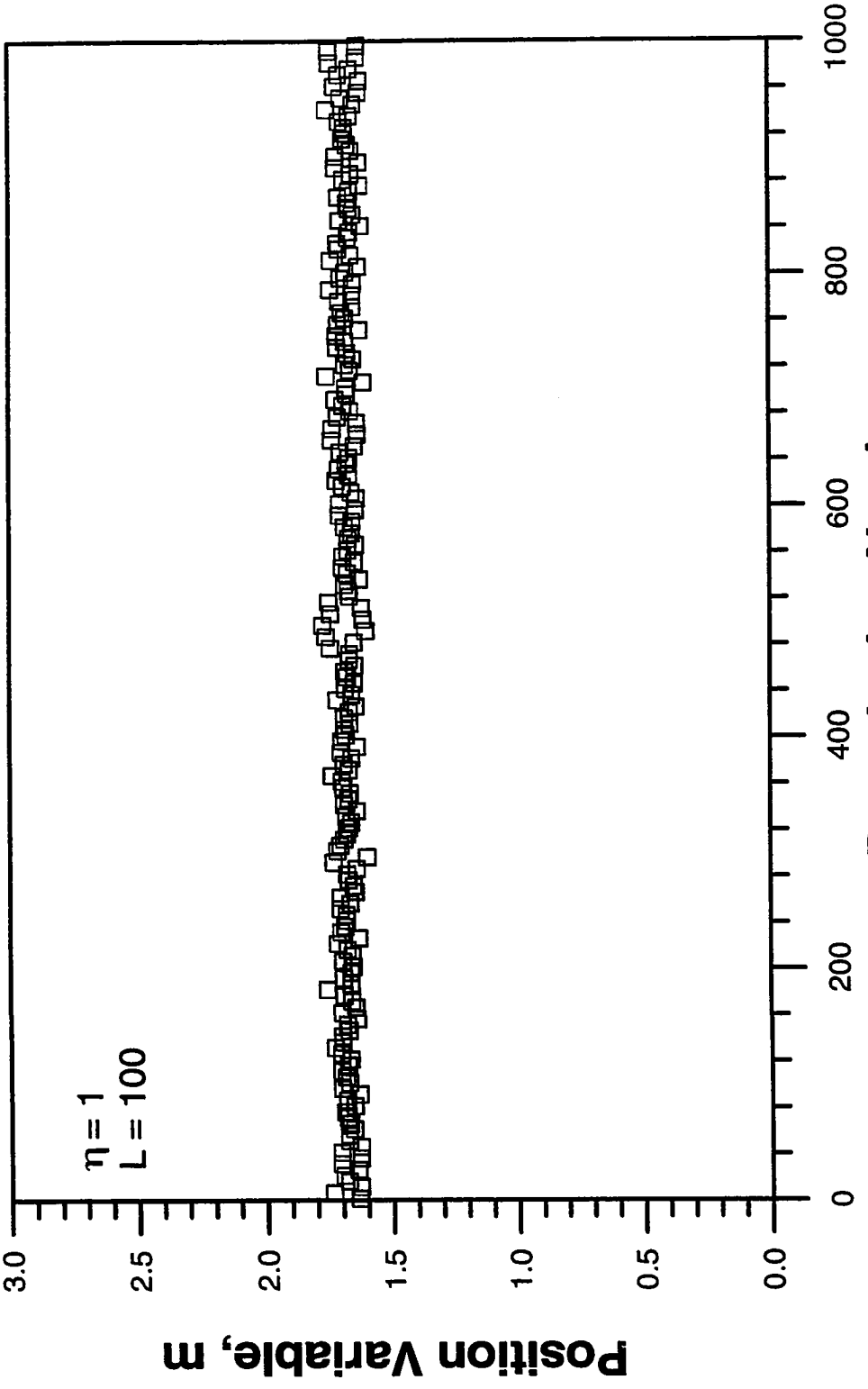

Fig. 5.13a - Behavior During Sampling with Hybrid Monte Carlo Method (Example 3).

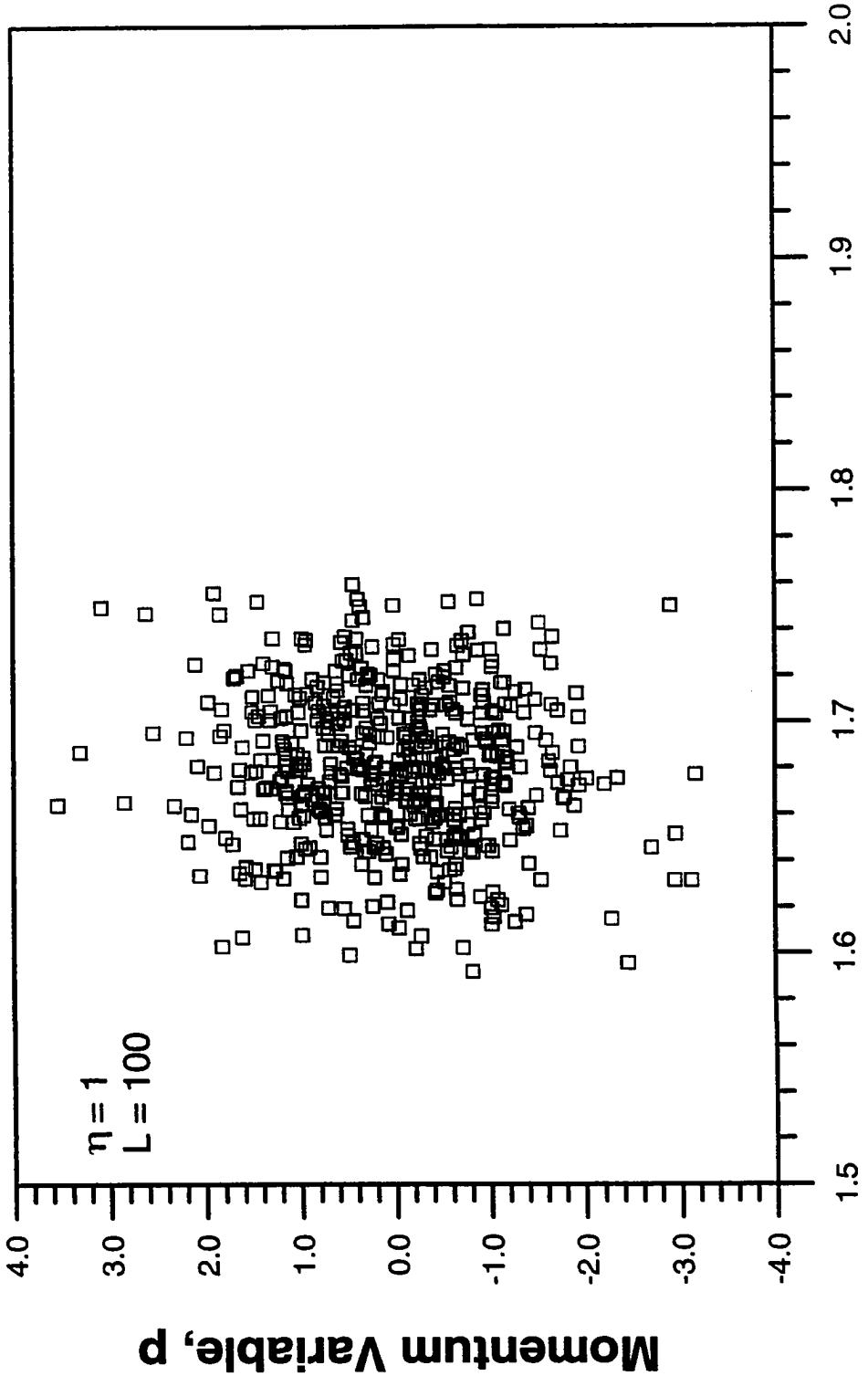

Fig. 5.13b - Behavior During Sampling with Hybrid Monte Carlo Method (Example 3).

constant. In Fig. 5.14 each curve plotted describes the values of the momentum variable, p and of the position variable m along trajectories of 1000 leapfrog steps. Note that, similarly to Example 2, it is possible to identify continuous paths from one region of probability to another. For example, if we have a sample (m_i,p_i) near $(m,p) = (m_\infty,0) = (1.50,0)$, then if we do a stochastic step (resample the probability function for p) and obtain say $p = 5.0$, then (m_i,p) may be on a curve of constant H, $(H = c)$ which encompasses both peaks. Since at each time step of the dynamical step, we generate a (m,p) value on this $H = c$ curve, at the completion of the dynamical step we may propose a new state (m_j,p) where m_j is near the second peak. Recall that variables p are sampled from a normal distribution with mean zero and standard deviation one. Thus, generating a $p = 5.0$ is highly unlikely and it will be difficult for the hybrid Monte Carlo to generate a sample around the secondary peak.

When comparing both methods, we see that, for this example both the conventional hybrid Monte Carlo method and the Markov chain Monte Carlo procedure (with perturbations based on the a posteriori variance) failed to yield a correct sampling. Thus, even though the behavior of the realizations shown in Fig. 5.12 and Fig. 5.13a suggests we have reached the stationary distribution of the Markov chain, we have not and consequently have not obtained a correct sample. However, we can also see that it is more likely that the hybrid method will obtain samples from the relevant peaks. As suggested by Fig. 5.14, to hope to sample from the second peak we need to propose a realization of p that is about five standard deviations away from its mean of zero. On the other hand, to propose a state new $m = 2.5$ using the standard Markov chain Monte Carlo

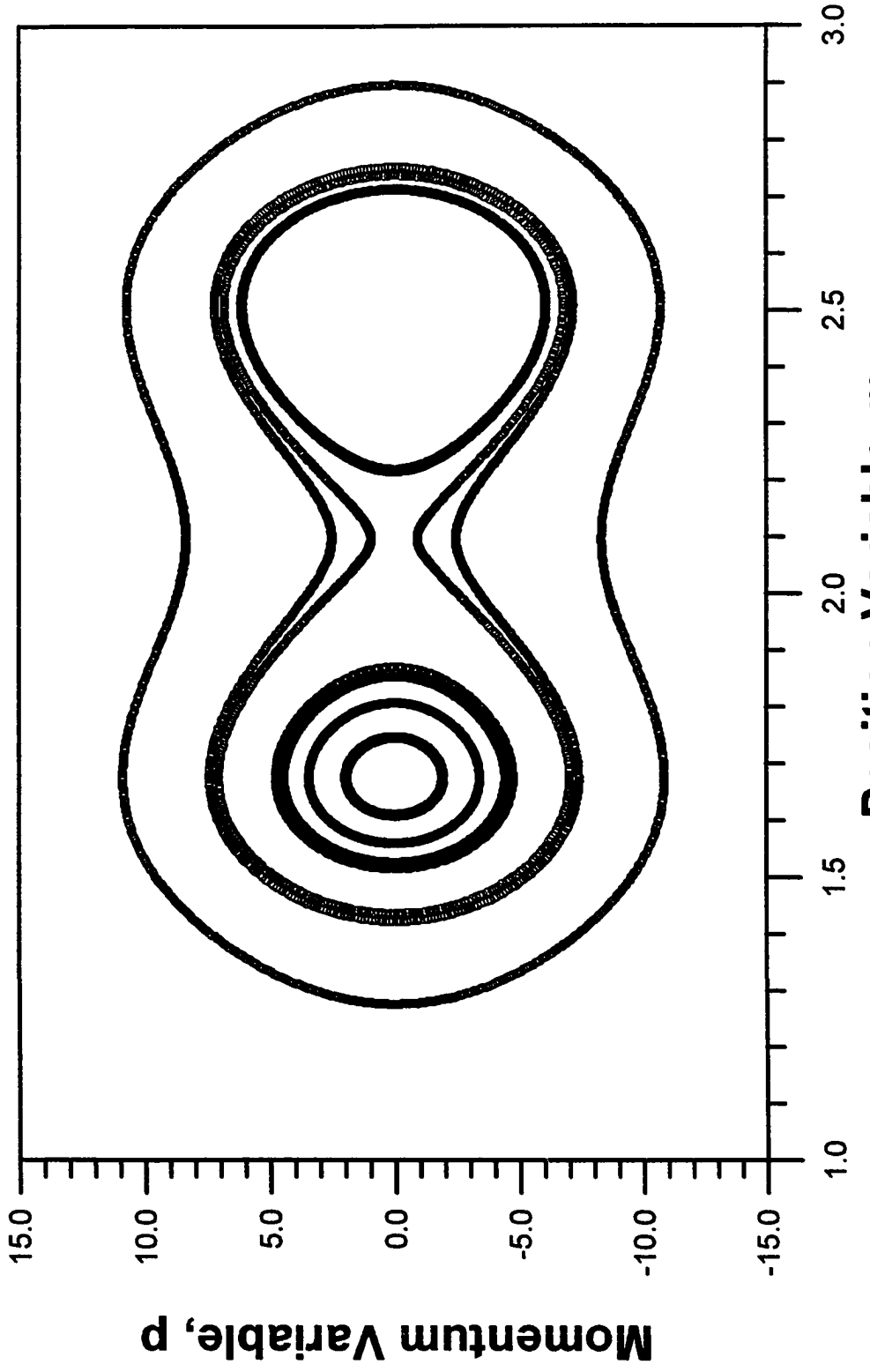

Fig. 5.14 - Equal Hamiltonian Function Contours (Example 3).

method (based on the a posteriori variance) requires that we propose a state 24 standard deviations away from its mean ($m_\infty = 1.67$).

CHAPTER VI

CONCLUSIONS

This chapter summarizes the work we have presented in the previous chapters, and enumerates the contributions and conclusions of this study.

The objective of this work has been to study methods to sample the a posteriori probability density function for permeability fields conditioned to a prior mean, the variogram and well-test pressure data.

In Chapter II, we have presented Markov chain Monte Carlo methods that can be used to generate realizations of the log-permeability field, which are conditioned to the prior mean, prior covariance (variogram) and multiwell pressure data. These realizations appear to represent samples from the correct a posteriori probability distribution. However, based on the simple example presented, we can not be certain that we have not simply generated samples around a peak in the probability density function close to m_{∞}. It is possible that there is another peak in the probability density function that we have not sampled thoroughly. In the event that the Markov chain Monte Carlo is actually failing to sample high probability regions far from the peak around m_{∞}, then correction of the method can only come from proposing new states from some distribution other than the one associated with C_{MP}^{-1} and centered at m_{∞}. It is possible this could be done by using

transition probabilities which are a sum or product of base transitions (see Neal[17]) where the base transitions could for example include states proposed from the prior covariance matrix and/or constitute a multiple of C_{MP}^{-1}. The problem is that we want to choose overall transition probabilities which give high rates of acceptance and at the same time generate samples from all regions of significant probability and it is not clear how to do this. The most efficient Markov chain Monte Carlo methods use either local or global perturbations generated from the a posteriori covariance matrix obtained from a gradient method. We have also presented a procedure where simulated annealing is used to generate the initial state in the Markov Chain and thereafter perturbations are generated from a covariance matrix associated with the sensitivity coefficients.

In Chapter III, we focused on examining the analogy between simulated annealing and Markov chain Monte Carlo methods. We have shown that a proper application of simulated annealing can be used to obtain a sample of the log-permeability field from the set of all most probable models. This method again is based on perturbations using a covariance matrix that incorporates sensitivity coefficients. The realization is one which approximately gives the global minimum of the correct objective function. This realization is in good agreement with the maximum a posteriori estimate obtained from the gradient method. We have also presented a much slower procedure for approximating a global minimum of the correct objective function by starting with a uncorrelated log-permeability field defined on the gridblocks and performing two point swaps. This method is considerably less efficient and does not yield a log-permeability field which represents a sample from the set of most probable models. The standard procedure which combines a heuristic weighting of the variogram and the pressure mismatch into one objective function

requires an excessively large amount of computer time and yields unreliable results. Our results and theoretical arguments indicate that the variogram should not be included as part of the simulated annealing objective function.

In Chapter IV, we investigated the use of the hybrid Monte Carlo method for sampling. For the problems considered, we showed that the hybrid Monte Carlo approach improves the sampling process for the highly nonlinear problems; i.e., a higher number of transitions is accepted without increasing the correlation between samples and the computer time required per independent image is less than in the standard Markov chain Monte Carlo method.

Although the hybrid Monte Carlo implementation we have described in this work enhances the sampling of the a posteriori probability distribution for permeability fields, there is no reason to think it is optimal. The time required for the sampling in Chapter IV shows that improvement in this respect is quite important. Many implementation schemes differing in detail could be investigated. For example, the leapfrog stepsizes could be chosen differently, a different parametrization for the model variables could be used, and the heuristic criteria used to choose a good trajectory length could be improved. One variant of the basic hybrid Monte Carlo method, the variant described by Horowitz, was investigated, but we found that although less computational time was required to perform the leapfrog iterations, the samples obtained were strongly correlated. Other variants remain to be tried, including those based on discretization of the dynamics accurate to higher order than the leapfrog method.

It is important to notice that the computational effort required by the hybrid Monte Carlo method is strongly related to the choice of stepsize (or mass term). If the values

calculated are optimum for each component direction, a high stepsize adjustment factor is permitted and consequently not a high number of leapfrog steps are necessary to sample each component. Since the major computational effort is expended in the computation of the gradient in the leapfrog steps, the computational time is reduced.

In this work, we estimate the stepsize values based on an analogy with a linear problem. The values are calculated at the beginning of the sampling phase and then kept constant. Keeping these values as constants through the whole sampling phase may not yield optimal performance since for the nonlinear case, the optimum values can change at each iteration. The fact that we required a stepsize adjustment factor of $\eta = 10^{-3}$ is a direct indication that we have not arrived at an optimal choice of the mass terms. Development of a criterion that considers this fact and is not computationally intensive is not an easy task and was not pursued here.

Another factor that impacts the computational time is the accuracy of the sensitivities calculated as a by-product of the flow simulation. These sensitivity coefficients are used to obtain the derivative of the energy with respect to the position variables during the leapfrog iterations. Incorrect sensitivities will cause an accumulation of errors in the calculation of the trajectories, and consequently a high rejection rate for a given value of L and η.

In Chapter V, we compared the performance of the methods for sampling described in this work. By means of a simple example which was chosen to have some features that are similar to the problem of sampling a posteriori probability distribution for a Gaussian random field conditioned to pressure data, we showed that it is possible to anticipate the behavior of problems in which the hybrid Monte Carlo can perform better than our

procedures based on Markov chain Monte Carlo methods. Although conventional Markov chain Monte Carlo methods are adequate for many problems, we believe that hybrid Monte Carlo can solve many problems faster than the methods presently used, and will permit the use of complex models for which the computations have until now been infeasible, for example, problems where we try to incorporate saturation or water cut data.

The time required for the sampling (see Chapters IV and V) shows that improvement in this area is quite important. In this respect, we identified three problems that prevents the hybrid Monte Carlo from being practical. Ultimately, it seems that the use of the hybrid Monte Carlo method to sample the a posteriori probability distribution for our problem depends strongly in three factors: (1) an efficient procedure to estimate the stepsize (or mass term) for each component to be sampled; (2) a higher order scheme than the leapfrog method to be applied to the dynamical transitions; (3) an efficient and more accurate procedure to estimate sensitivity coefficients from a flow simulation. We believe that improvements in these factors will certainly reflect in a overall increase in the method's efficiency .

NOMENCLATURE

a, b, c	constants.
$a_r, \hat{a}, \hat{b}$	constants.
c_t	total system compressibility, psi^{-1}.
C_D	covariance matrix for pressure measurement errors.
C_M	prior covariance matrix.
C_{MP}	a posteriori covariance matrix after incorporating all data.
C_{Mr}	reference covariance matrix.
d	vector of pressures calculated from simulator, psi.
d_{obs}	vector of measured wellbore pressure data, psi.
d_r	reference data vector, psi.
E	potential energy function.
f_M	a posteriori probability density function when considering the variogram in the simulated annealing objective function.
g	nonlinear transformation.
G	linear operator.
G_r	sensitivity coefficient matrix at the reference model, m_r.
h	formation thickness, ft; or variogram lag.

h_l variogram lag.

H Hamiltonian function.

k permeability, md; or constant.

K kinetic energy function.

L likelihood function; or lower matrix resulting from the Cholesky decomposition of a covariance matrix; or number of leapfrog steps; or local domain length.

L_r lower matrix resulting from the Cholesky decomposition of the reference covariance matrix.

m model variables; or mass term.

m_p reparameterized model variables.

$m_{p,\infty}$ reparameterized maximum a posteriori estimate.

m_r reference model.

m_∞ maximum a posteriori estimate after conditioning to all data.

m_0 prior mean.

m^0 set of most probable model.

M number of simulator gridblocks.

M model space.

n number of samples; or trial number; or sample from a standard Gaussian distribution.

n_l number of lags.

N_d number of conditioning pressure data.

N_p number of pairs in a lag.

O	objective function.
O_1, O_2	objective function components.
p	momentum variables.
$\hat{p}$	transformed momentum variable.
p_i	initial reservoir pressure, psi.
$p^n{}_{ij}$	transition probability from state i to state j at trial n.
P	canonical distribution for the momentum variables; or canonical distribution for the momentum and position variables.
q_{ij}	probability of proposing a transition from state i to state j.
r	variogram lag in which correlation between variables is negligible.
r_w	wellbore radius, ft.
r_{ij}	probability of accepting a proposed change from state i to j.
R	ratio between variogram range and length of local domain.
s	variogram lag.
s_{ij}	symmetric matrix used in Hastings' criteria.
S	misfit (objective) function.
t	time, days.
T	temperature parameter.
U	matrix containing eigenvectors of the spectral decomposition of the a posteriori matrix, C_{MP}.
X	random variable.
w_1, w_2	objective function components weights.

w_E, w_H, w_K constants.

Z vector of standard normal deviates.

Greek Symbols

$\nabla_m E$ gradient of the energy function to respect to model parameters.

α random uniform number from [0,1]; or tunable parameter for stochastic transitions in Horowitz method.

α_{ij} probability of accepting state j when the previous state is i.

β random uniform number from [0,1].

Δd difference between calculated and observed pressure.

ΔE energy change.

ΔO objective function change.

$\Delta\tau$ time interval.

ε stepsize; or error in a Taylor series expansion.

ϕ porosity, fraction.

γ variogram.

γ_{model} model variogram.

γ_s calculated variogram.

γ_v local variogram.

η stepsize adjustment factor.

λ eigenvalues from the spectral decomposition of C_{MP}.

μ	viscosity, cp.
μ_r	reference mean.
π	a posteriori probability distribution function.
$\hat{\pi}$	a posteriori probability distribution function based on C_{MP}.
$\tilde{\pi}$	approximate a posteriori probability distribution function.
σ^2	variance.
σ^2_{CMP}	a posteriori variance.
σ^2_d	variance at pressure measurement errors; or variance from observed data.
σ^2_m	prior variance.
ρ	probability distribution function.
ρ_M	prior probability distribution function.
τ	probability distribution function; or fictitious time variable.

Subscripts

i	ith component to be sampled.
i, j	states in a Markov chain.
k	kth component of a vector of standard normal deviates.
l	lth component of a vector of standard normal deviates; or lth temperature level.
t	index of an accepted realization.
sa	simulated annealing.

Superscripts

i, j	states in a Markov chain.
T	transpose of a matrix.

REFERENCES

1. Tarantola, A.: *Inverse Problem Theory, Methods for Data Fitting and Model Parameter Estimation*, Elsevier Science Publishers, Amsterdam (1987).
2. Oliver, D.S.: "Incorporation of Transient Pressure Data into Reservoir Characterization," *In Situ* (1994) **18**, 243-275.
3. Chu, L., Reynolds, A.C. and Oliver, D.S.: "Computation of Sensitivity Coefficients for Conditioning the Permeability Field to Well-Test Pressure Data," *In Situ* (1995) **19** (1995) 179-223.
4. Chu, L. and Reynolds, A.C.: "Application of Inverse Problem Theory to Generate Realizations of Rock Property Fields Conditioned to Variograms and Multiwell Pressure Data," TUPREP Research Report 12 (May 1995).
5. Metropolis, N., Rosenbluth, A., Teller, A. and Teller, E.: "Equation of State Calculations by Fast Computing Machines," *Journal of Chemical Physics* (1953) **21**, 1087-1092.
6. Alder, B.J. and Wainwright, T.E.: "Studies in Molecular Dynamics. I. General Method," *Journal of Chemical Physics* (1959) **31**, 459-466.
7. Duane, S., Kennedy, A.D., Pendleton, B.J. and Roweth, D.: "Hybrid Monte Carlo," *Physics Letters B* (1987) **195**, 216-222.

8. Horowitz, A.M.: "A Generalized Guided Monte Carlo Algorithm," *Physics Letters B* (1991) **268**, 247-252.

9. Tjelmeland, H., Omre, H. and Hegstad, B. K.: Sampling from Bayesian Models in Reservoir Characterization, *Technical Report Statistics* No. 2/1994, University of Trondheim (1994).

10. Hegstad, B. K., Omre, H., Tjelmeland, H. and Tyler, K.: Stochastic Simulation and Conditioning by Annealing in Reservoir Description, In M. Armstrong and P. A. Dowd, editors, *Geostatistical Simulation* (1993) 43-55. Kluwer Acad.

11. Smith, L. and Freeze, R.A.: "Stochastic Analysis of Steady State Groundwater Flow in a Bounded Domain 2. Two-Dimensional Simulations," *Water Resources Research* (1979) **15**, No. 6, 1543-1559.

12. Neal, R.M.: *Bayesian Learning for Neural Networks*, Ph.D. dissertation, University of Toronto (1995).

13. Kirkpatrick, S., Gelatt Jr., C.D. and Vecchi, M.P.: "Optimization by Simulated Annealing," *Science* (1983) 671-680.

14. Cerny, V.: "Thermodynamical Approach to the Traveling Salesman Problem.: An Efficient Simulation Algorithm," *Journal of Optimization Theory and Applications* (1985) **45**, 41-51.

15. Aarts, E. and Korst, J.: *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*, John Wiley and Sons, Chichester (1989).

16. Laarhoven, van, P.J.M. and Aarts, E.H.L.: *Simulated Annealing: Theory and Applications*, D. Reidel Publishing Company, Dordrecht (1987).

17. Neal, R.M.: *Probabilistic Inference Using Markov Chain Monte Carlo Methods*, Tech. Rep. CRG-TR-93-1, Dept. of Computer Science, University of Toronto (1993).

18. Geman, S. and Geman, D.: "Stochastic Relaxation, Gibbs Distributions, and Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-6 (1984) No. 6, 721-741.

19. Marroquin, J., Mitter, S. and Poggio, T.: "Probabilistic Solution of Ill-posed Problems in Computational Vision," *J. Am. Stat. Assoc.* (1987) **82**, 76-89.

20. Koren, Z., Mosegaard, K., Landa, E., Thore, P. and Tarantola, A.: "Monte Carlo Estimation and Resolution Analysis of Seismic Background Velocities," *J. Geophys. Res.* (1991) **96**, 20289-20299.

21. Mosegaard, K., and Vestegaard, P.D.: "A Simulated Annealing Approach to Seismic Model Optimization with Sparse Prior Information," *Geophys. Prospect.* (1991) **39**, 599-611.

22. Vestegaard, P.D. and Mosegaard, K.: "Inversion of Post-Stack Seismic Data Using Simulated Annealing," *Geophys. Prospect.* (1991) **39**, 613-624.

23. Rothman, D.H.: "Nonlinear Inversion, Statistical Mechanics, and Residual Statics Estimation," *Geophysics* (1985) **50**, 2797-2807.

24. Farmer, C.: "Numerical Rocks," *The Mathematical Generation of Reservoir Geology*, edited by J. Fayers and P.King, Oxford University Press, New York (1991) 235-252.

25. Deutsch, C.V.: *Annealing Techniques Applied to Reservoir Modeling and the Integration of Geological and Engineering (Well Test) Data*, Ph.D. dissertation, Stanford University, Stanford CA (1992).

26. Sagar, R.K., Kelkar, B.G. and Thompson, L.G.: "Reservoir Description by Integrating Well Test Data and Spatial Statistics," paper SPE 26462 presented at the 1993 SPE Annual Technical Conference and Exhibition, Houston, Oct. 3-6.

27. Hird, K.B. and Kelkar, M.: "Conditional Simulation for Reservoir Description Using Spatial and Well Performance Constraints," paper SPE 24750 presented at the 1992 SPE Annual Technical Conference and Exhibition, Washington, D.C., Oct. 4-7.

28. Panda, M. and Lake, L.W.: "Parallel Simulated Annealing for Stochastic Reservoir Modeling," paper SPE 26418 presented at the 1993 SPE Annual Technical Conference and Exhibition, Houston, TX, Oct. 3-6.

29. Smith, A.F.M. and Roberts, G.O.:"Bayesian Computation via the Gibbs Sampler and Related Markov Chain Monte Carlo Methods," *J. R.Statist. Soc.* B (1993) **55**, No. 1, 3-23.

30. Hastings, W.K.: "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrika* (1970) **57**, 97-109.

31. Chu, L. and Reynolds, A.C.: "A General Efficient Method For Generating Sensitivity Coefficients," TUPREP Research Report 12 (May 1995).

32. Alabert, F.: "The Practice of Fast Conditional Simulations Through the LU Decomposition of the Covariance Matrix," *Math. Geol.* (1987) **19**, 369-386.

33. Davis, M.W.: "Production of Conditional Simulations via the LU Decomposition of the Covariance Matrix," *Math. Geol.* (1987) **19**, 91-98.

34. Collins, N.E., Eglese, E.W. and Golden, B.L.: *Simulated Annealing - an Annotated Bibliography*, Cambridge University, preprint (1987).

35. Deutsch, C.V. and Cockerham, P.W.: "Practical Considerations in the Application of Simulated Annealing to Stochastic Simulation," *Mathematical Geology*, (1994) **26**, No. 1, 67-82.

36. Journel, A. and Huijbregts, C.J.: *Mining Geostatistics*, Academic Press, New York (1978).

37. Andersen, H.C.: "Molecular Dynamics Simulations at Constant Pressure and/or Temperature," *Journal of Chemical Physics* (1980) **72**, No. 4, 2384-2393.

38. Toussaint, D.: "Introduction to algorithms for Monte Carlo simulations and their application to QCD," *Computer Physics Communications* (1989) **56**, 69-92.

39. Kennedy, A.D.: "The Theory of Hybrid Stochastic Algorithms" in P.H. Dangaard *et al.* (editors) *Probabilistic Methods in Quantum Field Theory and Quantum Gravity*, Plenum Press, New York (1990).

40. Mackenzie, P.B.: "An Improved hybrid Monte Carlo method," *Physics Letters B*, (1989) **226**, 369-371.

41. Salamon, P. and Frost, R.: *SIAM Tutorial on Ensemble Based Simulated Annealing*, Course Notes (July 24, 1994) San Diego, CA.

42. Perez, G.: *Stochastic Conditional Simulation for Description of Reservoir Properties*, Ph.D. dissertation, University of Tulsa, Tulsa, OK (1992).

APPENDIX A

SIMULATED ANNEALING PROCEDURE

During the simulation algorithm two loops have to be considered. An external loop decreases the value of the control parameter (temperature) from an initial value. The second loop controls the maximum number of iterations (proposed perturbations) allowed at the step corresponding to a particular value of temperature. The algorithm terminates when either the objective function change is less then a specified tolerance or the acceptance ratio (number of accepted perturbations for the *k*th step (*k*th temperature level) divided by the total number of proposed perturbations for the *k*th step) is less than a specified tolerance.

A.1 Value of the Initial Control Parameter

The effect of the control parameter is evident from the Metropolis condition: if the value of the control parameter is high, then more of the perturbations proposed will be accepted. The initial control parameter or temperature is denoted by T_0 and is determined for the first step, numerically in the initial phase of the algorithm, the "warm up" phase[41]. An appropriate value for T_0 was derived by Aarts and Korst[15]. To determine T_0, we generate an initial distribution of the log-permeability field, m^0 and evaluate the objective

function to obtain $O^0=O(m^0)$. Then we propose $K=M_iN_s$ perturbations where N_s is the number of permeability values to be estimated. Here, we used $M_i=0.25$. For each proposed perturbation, we have a proposed log-permeability field m^j with resulting change in the objective function given by $\Delta O^j=O^j-O^0$. We let m_2 denote the number of proposed perturbations that give a positive change in the objective function, i.e., increase the objective function, and let ΔO^k_+ for $k=1,2,...m_2$ denote the values of these positive changes. We let m_1 denote the number of perturbations that give a decrease in the objective function and let X_0 denote the desired number of accepted perturbations at the initial temperature level. We let $\Delta O_{+,ave}$ denote the average positive change in the objective function, i.e.,

$$\Delta O_{+,ave} = \frac{1}{m_2}\sum_{k=1}^{m2}\Delta O_+ , \qquad \text{(A. 1)}$$

As suggested by Aarts and Korst[15], the initial value of the control parameter is then given by

$$T_0 = \frac{\Delta O_+}{\ln\left(m_2\frac{m_2}{X_0} - m_1(1-X_0)\right)}, \qquad \text{(A. 2)}$$

In this initial stage, we do not update the permeability field.

A.2 Annealing Schedule

The reduction in the temperature value is typically done by the following rule

$$T_r = \alpha T_{r-1} \tag{A. 3}$$

where α is the so-called convergence factor and must satisfy $0<\alpha<1$. In this work, we use $\alpha=0.5$.

A.3 Maximum Number of Iterations per Step

The method used to calculate the maximum number of total iterations allowed per step, i.e., the maximum number of proposed perturbations allowed at each temperature level was developed by Perez[42]. The method calculates the maximum number of iterations per step by estimating the acceptance ratio for the subsequent step. The maximum number of iterations allowed at the step corresponding to T_{r+1} is given by

$$M_t^{r+1} = M_a \frac{X_{r-1}}{(X_r)^2}. \tag{A. 4}$$

where M_a is the maximum number of iterations allowed at the initial temperature level. X_r and X_{r-1}, respectively, represent the acceptance ratios obtained at temperature levels T_r and T_{r-1}. In this work, we used $M_a=5N_s$.

A.4 Simulation Process

1. Generate the initial permeability distribution. In this work, we normally used an initial log-permeability field generated from an unconditional simulation based on the Cholesky decomposition of the prior covariance matrix. However, as discussed in the text, in some cases, we also initialized the log-permeability distribution on the gridblocks by sampling from the cumulative distribution for permeability.

2. Calculate the pressure response for the initial permeability distribution;

3. Calculate the value of the initial control parameter, T_0;

4. For the first step, set the maximum number of iterations per step equal to M_a, i.e., $M_t^1 = M_a$;

5. Propose a perturbation to a new log-permeability field;

6. Calculate the value of objective function at the proposed log-permeability field;

7. Apply the Metropolis condition to determine whether to accept the proposed log-permeability field;

8. If the perturbation is not accepted return to step 5;

9. If the perturbation is accepted, then update the log-permeability field to the values determined by the perturbation;

10. Check for convergence; if the tolerance condition of the objective function is satisfied the simulation ends; otherwise go to step 11;

11. Update the objective function for the next iteration;

12. Compare the number of perturbations proposed (k) at the current temperature to the maximum number allowed; if $k \geq M_t^r$ then go to step 5; otherwise go to step 13;

13. Check the acceptance ratio tolerance for the current step. The acceptance ratio, X_r is the fraction of the total iterations (attempted perturbations) in a step which have been accepted by the Metropolis condition; if $X_r<0.1$, the simulation is terminated; otherwise go to step 14. If the simulation terminates because the acceptance ratio becomes too low, it does not mean we have converged to a global minimum of the objective function. It simply means, we are having difficulty finding improved estimates of the permeability field. The acceptance ratio tolerance of 0.1 is a heuristic choice.

14. Calculate the control parameter for the next step from Eq. A.3;

15. Calculate the maximum number of iterations for the next step from Eq. A.4 and go to step 5.

APPENDIX B

DECOMPOSITION OF POTENTIAL ENERGY INTO LINEAR AND NONLINEAR COMPONENTS AND REPARAMETERIZATION IN TERMS OF EIGENVECTORS OF THE A POSTERIORI COVARIANCE MATRIX

The Hamiltonian function associated with our problem is expressed by the sum of two energy terms,

$$H(m) = E(m) + K(p) \tag{B.1}$$

where

$$E(m) = S(m) = \frac{1}{2}[(m - m_0)C_M^{-1}(m - m_0) + (g(m) - d_{obs})^T C_D^{-1}(g(m) - d_{obs})] \tag{B.2}$$

is the energy related to model mismatch ("potential" energy), and

$$K(p) = \frac{1}{2}\sum_{i=1}^{N} p_i^2 / \hat{m}_i = \frac{1}{2} p^T \hat{\mathrm{M}}^{-1} p \tag{B.3}$$

is a fictitious "kinetic" energy.

$$\hat{\mathrm{M}}^{-1} = \begin{bmatrix} 1/\hat{m}_1 & & \\ & \ddots & \\ & & 1/\hat{m}_N \end{bmatrix} \tag{B.4}$$

is the matrix of "masses".

In this appendix first we decompose the potential energy function expressed by Eq. B.2 into linear and nonlinear components. Second, we rewrite the potential energy function in terms of a reparameterized model m_p obtained from our model m and the eigenvectors that results from the spectral decomposition of the a posteriori covariance matrix C_{MP}. As shown later this potential energy and consequently Hamiltonian re-expression is particularly useful to analyze the computational effort involved when performing the dynamical steps during the hybrid Monte Carlo update.

Define the following gradient operators

$$\nabla_m = \begin{bmatrix} \frac{\partial}{\partial m_1} \\ \frac{\partial}{\partial m_2} \\ \vdots \\ \frac{\partial}{\partial m_N} \end{bmatrix}, \tag{B.5a}$$

$$\nabla_{mp} = \begin{bmatrix} \frac{\partial}{\partial m_{p1}} \\ \frac{\partial}{\partial m_{p2}} \\ \vdots \\ \frac{\partial}{\partial m_{pN}} \end{bmatrix}. \tag{B.5b}$$

Suppose $m = m(m_p)$ and $f(m)$ is a scalar function.

We can write

$$\nabla_m f(m(m_p)) = \nabla_m f(m) = \begin{bmatrix} \dfrac{\partial f}{\partial m_1} \\ \dfrac{\partial f}{\partial m_2} \\ \vdots \\ \dfrac{\partial f}{\partial m_N} \end{bmatrix} = \begin{bmatrix} \displaystyle\sum_{j=1}^{N} \dfrac{\partial f}{\partial m_{pj}} \dfrac{\partial m_{pj}}{\partial m_1} \\ \displaystyle\sum_{j=1}^{N} \dfrac{\partial f}{\partial m_{pj}} \dfrac{\partial m_{pj}}{\partial m_2} \\ \vdots \\ \displaystyle\sum_{j=1}^{N} \dfrac{\partial f}{\partial m_{pj}} \dfrac{\partial m_{pj}}{\partial m_N} \end{bmatrix} =$$

$$= \begin{bmatrix} \dfrac{\partial m_{p1}}{\partial m_1} & \dfrac{\partial m_{p2}}{\partial m_1} & \cdots & \dfrac{\partial m_{pN}}{\partial m_1} \\ \dfrac{\partial m_{p1}}{\partial m_2} & \dfrac{\partial m_{p2}}{\partial m_2} & \cdots & \dfrac{\partial m_{pN}}{\partial m_2} \\ \vdots & & \vdots & \vdots \\ \dfrac{\partial m_{p1}}{\partial m_N} & \dfrac{\partial m_{p2}}{\partial m_N} & \cdots & \dfrac{\partial m_{pN}}{\partial m_N} \end{bmatrix} \begin{bmatrix} \dfrac{\partial f}{\partial m_{p1}} \\ \dfrac{\partial f}{\partial m_{p2}} \\ \vdots \\ \dfrac{\partial f}{\partial m_{pN}} \end{bmatrix} \tag{B.6}$$

Note that

$$\nabla_m m_p^T = \begin{bmatrix} \dfrac{\partial}{\partial m_1} \\ \vdots \\ \dfrac{\partial}{\partial m_N} \end{bmatrix} \begin{bmatrix} m_{p1} & m_{p2} & \cdots & m_{pN} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial m_{p1}}{\partial m_1} & \dfrac{\partial m_{p2}}{\partial m_1} & \cdots & \dfrac{\partial m_{pN}}{\partial m_1} \\ \dfrac{\partial m_{p1}}{\partial m_2} & \dfrac{\partial m_{p2}}{\partial m_2} & \cdots & \dfrac{\partial m_{pN}}{\partial m_2} \\ \vdots & & \vdots & \vdots \\ \dfrac{\partial m_{p1}}{\partial m_N} & \dfrac{\partial m_{p2}}{\partial m_N} & \cdots & \dfrac{\partial m_{pN}}{\partial m_N} \end{bmatrix} \tag{B.7}$$

Using Eq. B.7, it follows that Eq. B.6 can be written as

$$\nabla_m f(m) = \nabla_m f(m(m_p)) = (\nabla_m m_p^T) \nabla_{m_p} f(m(m_p)) \tag{B.8}$$

Similarly

$$\nabla_{m_p} f(m_p) = \nabla_{m_p} f(m_p(m)) = (\nabla_{m_p} m^T) \nabla_m f(m) \tag{B.9}$$

Eqs. B.8 and B.9 are simply chain rules.

Choose $m_r = m_\infty$ so $G_r = G_\infty$ where m_∞ refers to the maximum a posteriori estimate obtained by the Gauss-Newton method. Thus, our expansions are

$$g(m) = g(m_\infty) + G_\infty(m - m_\infty) + \varepsilon(m) \tag{B.10}$$

or

$$\varepsilon(m) = g(m) - g(m_\infty) + G_\infty(m - m_\infty)\,. \tag{B.11}$$

Define

$$d_\infty = d_{obs} - g(m_\infty) + G_\infty m_\infty\,. \tag{B.12}$$

Eq. B.10 represents nothing more than a Taylor series expansion, i.e., the jth component of $g(m)$ is given by

$$g_j(m_{\infty,j}) + \sum_{l=1}^{N} \frac{\partial g_j(m_\infty)}{\partial m_l}(m_l - m_{\infty,l}) + \varepsilon_j(m), \qquad 1 \le j \le N_d$$

where N_d is the number of data and N is the number of model parameters. So Eq. B.10 is

$$\begin{bmatrix} g(m)_1 \\ \vdots \\ g(m)_{N_d} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial g_1}{\partial m_1} & \dfrac{\partial g_1}{\partial m_2} & \cdots & \dfrac{\partial g_1}{\partial m_N} \\ \vdots & & & \\ \dfrac{\partial g_{N_d}}{\partial m_1} & & \cdots & \dfrac{\partial g_{N_d}}{\partial m_N} \end{bmatrix}_{m_\infty} \begin{bmatrix} m_1 - m_{\infty,1} \\ \vdots \\ m_N - m_{\infty,N} \end{bmatrix} = G_\infty(m - m_\infty) + \varepsilon(m) \tag{B.13}$$

where

$$G_\infty = \begin{bmatrix} \frac{\partial g_1}{\partial m_1} & \cdots & \frac{\partial g_1}{\partial m_N} \\ \vdots & & \\ \frac{\partial g_{N_d}}{\partial m_1} & \cdots & \frac{\partial g_{N_d}}{\partial m_N} \end{bmatrix} = [\nabla_m (g^T)]^T_{m_\infty} \tag{B.14}$$

is the N_dxN sensitivity coefficient matrix.

Note

$$\nabla_m \varepsilon(m)^T = \begin{bmatrix} \frac{\partial}{\partial m_1} \\ \vdots \\ \frac{\partial}{\partial m_N} \end{bmatrix} [\varepsilon_1 \quad \cdots \quad \varepsilon_{N_d}] = \begin{bmatrix} \frac{\partial \varepsilon_1}{\partial m_1} & \cdots & \frac{\partial \varepsilon_{N_d}}{\partial m_N} \\ \vdots & & \\ \frac{\partial \varepsilon_1}{\partial m_N} & \cdots & \frac{\partial \varepsilon_{N_d}}{\partial m_N} \end{bmatrix} \tag{B.15}$$

is an NxN_d matrix.

With $m_r = m_\infty$, Eqs. 2.36. and 2.37 from Chapter II are

$$\mu = \mu_r = m_0 + C_M G_\infty^T [G_\infty C_M G_\infty^T + C_D]^{-1} [d_\infty - G_\infty m_0]$$

$$= m_0 + C_M G_\infty^T [G_\infty C_M G_\infty^T + C_D]^{-1} [d_{obs} - g(m_\infty) + G_\infty m_\infty - G_\infty m_0] \tag{B.16}$$

$$C_{MP}^{-1} = G_\infty^T C_D^{-1} G_\infty + C_M^{-1} \tag{B.17}$$

For the form of the Gauss-Newton method we use (see Tarantola, pg. 244 with μ_r=1)

$$m_{n+1} = m_0 - C_M G_n^T [G_n C_M G_n^T + C_D]^{-1} [g(m_n) - d_{obs} - G_n m_\infty + G_n m_0]. \quad \text{(B.18)}$$

As $n \to \infty$, $m_{n+1} \to m_\infty$, $m_n \to m_\infty$ so Eq. (B.18) becomes

$$m_\infty = m_0 - C_M G_\infty^T [C_D + G_\infty C_M G_\infty^T]^{-1} [g(m_\infty) - d_{obs} - G_\infty m_\infty + G_\infty m_0] \quad \text{(B.19)}$$

Comparing Eqs. B.16 and B.19 we see that $m_r = m_\infty$ implies that

$$\mu_r = m_\infty = m_0 + C_M G_\infty^T [G_\infty C_M G_\infty^T + C_D]^{-1} [d_\infty - G_\infty m_0] \quad \text{(B.20)}$$

Using Eqs. B.10 and B.12 in Eq. B.2, the potential energy can be written as

$$E(m) = S(m) = \frac{1}{2}(m - m_0)^T C_M^{-1} (m - m_0)$$

$$+\frac{1}{2}[g(m_\infty) + G_\infty (m - m_\infty) + \varepsilon(m) - d_{obs}]^T C_D^{-1} [g(m_\infty) + G_\infty (m - m_\infty) + \varepsilon(m) - d_{obs}]$$

$$= \frac{1}{2}(m - m_0)^T C_M^{-1} (m - m_0)$$

$$+\frac{1}{2}[G_\infty m - d_\infty + \varepsilon(m)]^T C_D^{-1} [G_\infty m - d_\infty + \varepsilon(m)]$$

$$= \frac{1}{2}(m - m_0)^T C_M^{-1} (m - m_0)$$

$$+\frac{1}{2}[G_\infty m - d_\infty]^T C_D^{-1} [G_\infty m - d_\infty + \varepsilon(m)]$$

$$+\frac{1}{2}\varepsilon(m)^T C_D^{-1}[G_\infty m - d_\infty + \varepsilon(m)]$$

$$= \frac{1}{2}(m-m_0)^T C_M^{-1}(m-m_0) + \frac{1}{2}(G_\infty m - d_\infty)^T C_D^{-1}(G_\infty m - d_\infty)$$

$$+\frac{1}{2}(G_\infty m - d_\infty)^T C_D^{-1}\varepsilon(m) + \frac{1}{2}\varepsilon(m)^T C_D^{-1}(G_\infty m - d_\infty)$$

$$+\frac{1}{2}\varepsilon(m)^T C_D^{-1}\varepsilon(m)$$

or

$$E(m) = S(m) = \frac{1}{2}(m-m_0)^T C_M^{-1}(m-m_0) +$$

$$+\frac{1}{2}(G_\infty m - d_\infty)^T C_D^{-1}(G_\infty m - d_\infty)$$

$$+(G_\infty m - d_\infty)^T C_D^{-1}\varepsilon(m) + \frac{1}{2}\varepsilon(m)^T C_D^{-1}\varepsilon(m) \qquad \text{(B.21)}$$

Matrix Identity (see Tarantola[1], 1987, page 158)

Identity 1

$$[G^T C_D^{-1} G + C_M^{-1}]^{-1} C_M^{-1} = I - C_M G^T [C_D + G C_M G^T]^{-1} G \qquad \text{(B.22a)}$$

Identity 2

$$[C_M^{-1} + G^T C_D^{-1} G]^{-1} G^T C_D^{-1} = C_M G^T [C_D + G C_M G^T]^{-1} \qquad \text{(B.22b)}$$

Using matrix algebra, Eq. B.19 can be written as

$$m_\infty = m_0 + C_M G_\infty^T [G_\infty C_M G_\infty^T + C_D]^{-1} [d_\infty - G_\infty m_0]$$

$$= [I - C_M G_\infty^T [G_\infty C_M G_\infty^T + C_D]^{-1} G_\infty] m_0$$

$$+ C_M G_\infty^T [G_\infty C_M G_\infty^T + C_D]^{-1} d_\infty$$

Using Eq. B.22a and then B.22b it follows that

$$m_\infty = [G_\infty^T C_D^{-1} G_\infty + C_M^{-1}]^{-1} C_M^{-1} m_0$$

$$+ C_M G_\infty^T [G_\infty C_M G_\infty^T + C_D]^{-1} d_\infty$$

$$= [G_\infty^T C_D^{-1} G_\infty + C_M^{-1}]^{-1} C_M^{-1} m_0$$

$$+ [C_M^{-1} + G_\infty^T C_D^{-1} G_\infty]^{-1} G_\infty^T C_D^{-1} d_\infty \qquad \text{(B.23)}$$

or

$$m_\infty = (G_\infty^T C_D^{-1} G_\infty + C_M^{-1})^{-1} (C_M^{-1} m_0 + G_\infty^T C_D^{-1} d_\infty) \qquad \text{(B.24)}$$

Note this last form of m_∞ is equivalent to Eq. 1.89, pg. 69, of Tarantola[1].

Thus, using the Equation solution Eq. 1.90 (pg. 69 of Tarantola) we have

$$S(m) = \frac{1}{2}(m - m_0)^T C_M^{-1}(m - m_0) + \frac{1}{2}(G_\infty m - d_\infty) C_D^{-1}(G_\infty m - d_\infty)$$

$$= \frac{1}{2}[(m - m_\infty)^T C_{MP}^{-1}(m - m_\infty)$$

$$-m_\infty^T C_M^{-1} m_\infty + d_\infty^T C_D^{-1} d_\infty + m_0^T C_M^{-1} m_0]. \qquad \text{(B.25)}$$

Using Eq. B.25 in Eq. B.21 gives

$$E(m) = S(m) = \frac{1}{2}(m - m_\infty)^T C_{MP}^{-1}(m - m_\infty)$$

$$-\frac{1}{2} m_\infty^T C_M^{-1} m_\infty + \frac{1}{2} d_\infty^T C_D^{-1} d_\infty + \frac{1}{2} m_0^T C_M^{-1} m_0$$

$$+(G_\infty m - d_\infty)^T C_D^{-1} \varepsilon(m)$$

$$+\frac{1}{2}\varepsilon(m)^T C_D^{-1} \varepsilon(m). \qquad \text{(B.26)}$$

We define the constant c by

$$c = -\frac{1}{2} m_\infty^T C_M^{-1} m_\infty + \frac{1}{2} d_\infty^T C_D^{-1} d_\infty + \frac{1}{2} m_0^T C_M^{-1} m_0.$$

So

$$E(m) = \frac{1}{2}(m - m_\infty)^T C_{MP}^{-1}(m - m_\infty)$$

$$+(G_\infty m - d_\infty)^T C_D^{-1}\varepsilon(m)$$

$$+\frac{1}{2}\varepsilon(m)^T C_D^{-1}\varepsilon(m)$$

$$+c. \qquad \text{(B.27)}$$

Now we can reparametrize in terms of eigenvectors from the spectral decomposition of C_{MP}, i.e.

$$C_{MP} = U\Lambda U^T, \qquad \text{(B.28)}$$

so, assuming that C_{MP} is positive definite,

$$C_{MP}^{-1} = U\Lambda^{-1}U^T \qquad \text{(B.29)}$$

where, U is the matrix of orthonormal eigenvectors so, $UU^T=U^TU=I$.

The reparametrized model, m_p, is defined by

$$m_p = U^T m \qquad \text{(B.30a)}$$

which is equivalent to

$$m = Um_p \qquad \text{(B.30b)}$$

Thus,

$$m_{p,\infty} = U^T m_\infty \qquad \text{(B.30c)}$$

or equivalently,

$$m_\infty = U m_{p,\infty} \,, \tag{B.30d}$$

where $m_{p,\infty}$ is the reparameterized maximum a posteriori estimate.

In terms of the reparameterized model, Eq. B.27 becomes

$$\begin{aligned} E(m) = E(m_p) = &\frac{1}{2}(m_p - m_{p,\infty})^T U^T (U\Lambda^{-1}U^T) U (m_p - m_{p,\infty}) \\ &+(G_\infty U m_p - d_\infty)^T C_D^{-1} \varepsilon(m) \\ &+\frac{1}{2}\varepsilon(m)^T C_D^{-1} \varepsilon(m) \\ &+c \\ \\ &= \frac{1}{2}(m_p - m_{p,\infty})^T \Lambda^{-1} (m_p - m_{p,\infty}) \\ &+(G_\infty U m_p - d_\infty)^T C_D^{-1} \varepsilon(m) + \frac{1}{2}\varepsilon(m)^T C_D^{-1} \varepsilon(m) \\ &+c. \end{aligned} \tag{B.31}$$

In order to apply the Hamiltonian dynamics in terms of m_p model, we need to calculate the gradient of the potential energy with respect to the parameters m_p.

$$\nabla_{m_p} E(m_p) = \Lambda^{-1}(m_p - m_{p,\infty})$$

$$+(\nabla_{m_p}[G_\infty U m_p - d_\infty]^T)C_D^{-1}\varepsilon(m)$$

$$+\nabla_{m_p}[(C_D^{-1}\varepsilon(m))^T][G_\infty U m_p - d_\infty]$$

$$+[\nabla_{m_p}\varepsilon(m)^T]C_D^{-1}\varepsilon(m)$$

$$= \Lambda^{-1}(m_p - m_{p,\infty})$$

$$+\nabla_{m_p}[m_p^T U^T G_\infty^T]C_D^{-1}\varepsilon(m)$$

$$+(\nabla_{m_p}\varepsilon(m)^T C_D^{-1})[G_\infty U m_p - d_\infty] + (\nabla_{m_p}\varepsilon(m)^T)C_D^{-1}\varepsilon(m)$$

$$= \Lambda^{-1}(m_p - m_{p,\infty}) + U^T G_\infty^T C_D^{-1}\varepsilon(m)$$

$$+(\nabla_{m_p}\varepsilon(m)^T C_D^{-1})[G_\infty U m_p - d_\infty] + (\nabla_{m_p}\varepsilon(m)^T)C_D^{-1}\varepsilon(m)\,. \qquad \text{(B.32)}$$

or using the chain rule Eq. B.9 (also see Eq. B.15)

$$\nabla_{m_p}E(m_p) = \Lambda^{-1}(m_p - m_{p,\infty}) + U^T G_\infty^T C_D^{-1}\varepsilon(m)$$

$$+(\nabla_{m_p}m^T)(\nabla_m\varepsilon(m)^T)C_D^{-1}[G_\infty U m_p - d_\infty] + (\nabla_{m_p}m^T)(\nabla_m\varepsilon(m)^T)C_D^{-1}\varepsilon(m)\,. \quad \text{(B.33)}$$

By taking the transpose of Eq. B.30a, and applying the gradient, we obtain

$$\nabla_{m_p} m^T = U^T \tag{B.34}$$

Using Eq. B.34 in B.33

$$\nabla_{m_p} E(m_p) = \Lambda^{-1}(m_p - m_{p,\infty}) + U^T G_\infty^T C_D^{-1} \varepsilon(m)$$

$$+U^T(\nabla_m \varepsilon(m)^T) C_D^{-1}(G_\infty U m_p - d_\infty)$$

$$+(U^T)(\nabla_m \varepsilon(m)^T) C_D^{-1} \varepsilon(m). \tag{B.35}$$

From Eq. B.11

$$\varepsilon(m)^T = g(m)^T - g(m_\infty)^T - (m - m_\infty)^T G_\infty^T \tag{B.36}$$

and

$$\nabla_m \varepsilon(m)^T = \nabla_m g(m)^T - G_\infty^T = G^T - G_\infty^T. \tag{B.37}$$

where, G is the sensitivity coefficient matrix evaluated at m.

Using Eq. B.37 in Eq. B.35, it follows that

$$\nabla_{m_p} E(m_p) = \Lambda^{-1}(m_p - m_{p,\infty}) + U^T G_\infty^T C_D^{-1} \varepsilon(m)$$

$$+U^T[G^T - G_\infty^T] C_D^{-1}(G_\infty U m_p - d_\infty)$$

$$+U^T[G^T - G_\infty^T] C_D^{-1} \varepsilon(m)$$

$$= \Lambda^{-1}(m_p - m_{p,\infty}) + U^T G_\infty^T C_D^{-1} \varepsilon(m)$$

$$+U^T [G^T - G_\infty^T] C_D^{-1} (G_\infty U m_p - d_\infty + \varepsilon(m)) . \qquad \text{(B.38)}$$

From Eq. B.30c, B.11 and B.12, we have

$$G_\infty U m_p - d_\infty + \varepsilon(m) = G_\infty m - d_\infty + g(m) - g(m_\infty) - G_\infty m + G_\infty m_\infty$$

$$= G_\infty m - (d_{obs} - g(m_\infty) + G_\infty m_\infty))$$

$$+g(m) - g(m_\infty) - G_\infty m + G_\infty m_\infty$$

$$= g(m) - d_{obs} \qquad \text{(B.39)}$$

Thus Eq. B.38 is

$$\nabla_{m_p} E(m_p) = \Lambda^{-1}(m_p - m_{p,\infty})$$

$$+U^T G_\infty^T C_D^{-1} \varepsilon(m)$$

$$+U^T [G^T - G_\infty^T] C_D^{-1} (g(m) - d_{obs}) \qquad \text{(B.40)}$$

which means that, when $g(m)$ is nearly linear, so $G^T \approx G^T_\infty$ and $\varepsilon(m) \approx 0$, the gradient of the potential energy can be approximated by the product of the inverse of the eigenvalues of the a posteriori covariance and the parameters themselves.

APPENDIX C

APPROXIMATE HAMILTONIAN AND ANALYTICAL SOLUTION

As described by Duane *et al.*[7] the Hamiltonian used during the dynamical transitions in a hybrid Monte Carlo update does not need to be the same as the one used in the Metropolis criterion. The algorithm remains valid provided the decision whether to accept or reject a transition is based on the true Hamiltonian. Recall that the dynamical transitions require the majority of computational effort during a hybrid Monte Carlo procedure. The previous arguments suggest that, during the dynamical transitions, we use an approximate Hamiltonian that can be computed with far less computational effort than is required to calculate the true Hamiltonian.

In this appendix, we use the total energy expressed as a function of the reparameterized model (Eq. B.31) to determine an "approximate" Hamiltonian which requires much less computational effort to evaluate than does the full Hamiltonian. When sampling with the hybrid Monte Carlo method, the majority of the total computational time used is expended in performing the dynamical step, i.e., in solving the dynamical equations (see Eqs. 4.33-4.35). The solution of these equations is computational expensive because the evaluation of the gradient term in Eqs. 4.33 and 4.35 requires that we generate sensitivity coefficients at each leapfrog step. Thus, if the Hamiltonian is replaced by an approximate Hamiltonian and the gradient of the approximate Hamiltonian

can be easily evaluated then we can obtain a significant increase in computational efficiency.

In Appendix B we showed that the full Hamiltonian for our reparemeterized model can be written as

$$H(m_p, p) = \frac{1}{2}(m_p - m_{p,\infty})^T \Lambda^{-1}(m_p - m_{p,\infty})$$

$$+(G_\infty U m_p - d_\infty)^T C_D^{-1}\varepsilon(m)$$

$$+c + \frac{1}{2}p^T p \qquad \text{(C.1)}$$

where

$$c = -\frac{1}{2}m_\infty^T C_M^{-1} m_\infty + \frac{1}{2}d_\infty^T C_D^{-1} d_\infty + \frac{1}{2}m_0^T C_M^{-1} m_0 .$$

Here, we consider an "approximate" Hamiltonian , $\hat{H}$ which has the form

$$\hat{H} = \frac{1}{2}(m_p - m_{p,\infty})^T \Lambda^{-1}(m_p - m_{p,\infty}) + \frac{1}{2}p^T p . \qquad \text{(C.2)}$$

The dynamical equations are,

$$\frac{dm_{p_i}}{d\tau} = \frac{\partial \hat{H}}{\partial p_i} \qquad \text{(C.3)}$$

$$\frac{dp_i}{d\tau} = -\frac{\partial \hat{H}}{\partial m_{p_i}} \qquad \text{(C.4)}$$

After rotation with the approximate Hamiltonian, components of the dynamical equations separate as

$$\frac{dm_{p_i}}{d\tau} = p_i \tag{C.5}$$

$$\frac{dp_i}{d\tau} = -\frac{1}{\lambda_i} m_{p_i} + \frac{1}{\lambda_i} m_{p_{\infty,i}} \tag{C.6}$$

Since components separate, we suppress the subscript i and write the Hamiltonian system as

$$\frac{dm_p}{d\tau} = p, \tag{C.7}$$

and

$$\frac{dp}{d\tau} = -\frac{1}{\lambda} m_p + \frac{1}{\lambda} m_{p,\infty}. \tag{C.8}$$

We wish to move forward in time an amount $\delta\tau$ from some time τ_0 where $m(\tau_0)=m_0$ and $p(\tau_0)=p_0$ are known. Although we could write the system of equations given by Eqs. C.5 and C.6 in matrix form as $\frac{dx}{d\tau} = Ax$ and seek solution of the form $\vec{x}_j = \vec{\alpha}_j e^{u_j \tau}$ where the u_j's are the eigenvalues of the coefficient matrix, we take a different approach.

Differentiating C.7 and C.8

$$\frac{d^2 m_p}{d\tau^2} = \frac{dp}{d\tau} = -\frac{1}{\lambda} m_p + \frac{1}{\lambda} m_{p,\infty}, \tag{C.9}$$

and

$$\frac{d^2 p}{d\tau^2} = -\frac{1}{\lambda}\frac{dm_p}{d\tau} = -\frac{1}{\lambda} p, \tag{C.10}$$

or,

$$\frac{d^2 m_p}{d\tau^2} + \frac{1}{\lambda} m_p = \frac{1}{\lambda} m_{p,\infty}, \tag{C.11}$$

and

$$\frac{d^2 p}{d\tau^2} + \frac{1}{\lambda} p = 0. \tag{C.12}$$

The general solutions of C.11 and C.12, respectively, are

$$m_p = a\cos\left[\frac{(\tau - \tau_0)}{\sqrt{\lambda}}\right] + b\sin\left[\frac{(\tau - \tau_0)}{\sqrt{\lambda}}\right] + m_{p,\infty}, \tag{C.13}$$

and

$$p = c\cos\left[\frac{(\tau - \tau_0)}{\sqrt{\lambda}}\right] + d\sin\left[\frac{(\tau - \tau_0)}{\sqrt{\lambda}}\right]. \tag{C.14}$$

We apply initial conditions at $\tau = \tau_0$ of the form

$$m_p(\tau_0) = m_{p,0} \tag{C.15}$$

$$p(\tau_0) = p_0 \tag{C.16}$$

Applying the initial condition of Eq. C.15 we have

$$m_p(\tau_0) = m_{p,0} = a + m_{p,\infty},$$

so

$$a = m_{p,0} - m_{p,\infty}. \tag{C.17}$$

Applying the initial condition of Eq. C.16 and noting the relation of Eq. C.7, we have

$$\frac{dm_p}{d\tau}(\tau_0) = p(\tau_0) = p_0 = \frac{b}{\sqrt{\lambda}}\cos 0 = \frac{b}{\sqrt{\lambda}}$$

Thus,

$$b = \sqrt{\lambda}\, p_0 \tag{C.18}$$

From Eqs. C.13, C.17 and C.18, it follows that

$$m_p = (m_{p,0} - m_{p,\infty})\cos\left[\frac{(\tau - \tau_0)}{\sqrt{\lambda}}\right] + \sqrt{\lambda}\, p_0 \sin\left[\frac{(\tau - \tau_0)}{\sqrt{\lambda}}\right] + m_{p,\infty}. \tag{C.19}$$

From Eq. C.14 and the initial condition of Eq. C.16 we have

$$c = p_0 = p(\tau_0) \tag{C.20}$$

Differentiating Eq. C.14 gives

$$\frac{dp}{d\tau} = \frac{-c}{\sqrt{\lambda}} \sin\left[\frac{(\tau - \tau_0)}{\sqrt{\lambda}}\right] + \frac{d}{\sqrt{\lambda}} \cos\left[\frac{(\tau - \tau_0)}{\sqrt{\lambda}}\right]. \qquad \text{(C.21)}$$

Evaluating Eq. C.21 at $\tau = \tau_0$ and using Eq. C.8 and initial condition C.15, we see that

$$\frac{dp}{d\tau}(\tau_0) = \frac{d}{\sqrt{\lambda}} = -\frac{1}{\lambda} m_{p,0} + \frac{1}{\lambda} m_{p,\infty},$$

thus

$$d = -\frac{1}{\sqrt{\lambda}} m_{p,0} + \frac{1}{\sqrt{\lambda}} m_{p,\infty} \qquad \text{(C.22)}$$

Using Eqs. C.20 and C.22 in Eq. C.14 gives

$$p = p_0 \cos\left[\frac{(\tau - \tau_0)}{\sqrt{\lambda}}\right] + \frac{1}{\sqrt{\lambda}} [m_{p,\infty} - m_{p,0}] \sin\left[\frac{(\tau - \tau_0)}{\sqrt{\lambda}}\right] \qquad \text{(C.23)}$$