

USING PARALLEL MARKOV CHAIN MONTE CARLO TO QUANTIFY UNCERTAINTIES IN GEOTHERMAL RESERVOIR CALIBRATION

T. Cui,^{1,*} C. Fox,² M. J. O'Sullivan,³ & G. K. Nicholls⁴

¹School of Mathematical Sciences, Monash University, VIC 3800, Australia

²Department of Physics, University of Otago, Dunedin 9016, New Zealand.

³Department of Engineering Sciences, The University of Auckland, Auckland 1010, New Zealand.

⁴Department of Statistics, University of Oxford, Oxford, OX1 3LG, UK.

*Address all correspondence to: T. Cui, School of Mathematical Sciences, Monash University, VIC 3800, Australia, E-mail: tiangang.cui@monash.edu

Original Manuscript Submitted: mm/dd/yyyy; Final Draft Received: mm/dd/yyyy

We introduce a parallel rejection scheme to give a simple but reliable way to parallelise the Metropolis-Hastings algorithm. This method can be particularly useful when the target density is computationally expensive to evaluate and the acceptance rate of the Metropolis-Hastings is low. We apply the resulting method to quantify uncertainties of inverse problems, in which we aim to calibrate a challenging nonlinear geothermal reservoir model using real measurements from well tests. We demonstrate the parallelised method on various well-test scenarios. In some scenarios, the sample-based statistics obtained by our scheme shows clear advantages in providing robust model calibration and prediction compared with those obtained by nonlinear optimisation methods.

KEY WORDS: parallel MCMC, statistical inverse problems, geothermal modelling, uncertainty qualification

1. INTRODUCTION

Methods based on Bayesian inference are now well established as a route to solving and quantifying uncertainties in inverse problems. Such methods have been applied to subsurface [1–3], electrical impedance tomography [4–6], meteorology [7], remote sensing [8], seismic inversion [9], and many other areas. In the Bayesian formulation, the solution to the inverse problem is found as expectation of some functions of interests summarised over the feasible range of parameters consistent with the measured data, which is governed by the posterior distribution.

Since inverse problems are commonly defined on a high-dimensional parameter space and the parameters have nonlinear interactions, the required expectation over the posterior distribution are analytically intractable. One widely used way to explore the posterior distribution, and hence to compute posterior expectations, is the Markov chain Monte Carlo (MCMC). However, applying MCMC to statistical inverse problems faces a major difficulty that the evaluation of the posterior density involves a computationally intensive computer model simulation. Moreover, the posterior can be multimodal in some scenarios. This can lead to low acceptance rates in MCMC, and thus makes the resulting estimator statistically inefficient for limited amount of computing resources. To improve the performance of MCMC sampling, we introduce a *parallel rejection* scheme that parallelises the Metropolis-Hastings [10,11] (MH) algorithm. This is achieved by parallelising evaluations of the target density for proposed states that can be potentially rejected. In comparison, parallelisation schemes such as parallel tempering [12] and parallel tempering using multi-level models [1] require modifications to the posterior distribution, methods such as [13–15] simultaneously simulate multiple Markov chains, the work of [16] constructs a general transition kernel that can be evaluated in parallel. Our method focuses on the parallelisation of a single serial chain and can be easily implemented on mainframe computers

or multi-core workstations.

In this paper, we apply the parallel rejection scheme to solve an inverse problem, in which a geothermal reservoir model is calibrated using real measurements taken in well tests. We are interested in estimating and predicting multiphase and non-isothermal subsurface flows that are governed by multiphase Darcy's law, and mass and energy conservation. In this process, the crucial step is to solve the inverse problem—inferring the driven model parameters, including permeability, porosity, relative permeability, capillary pressure, and boundary conditions, to fit the measured well data such as near-surface pressure and flow enthalpy.

Traditionally, model calibration is treated as an optimisation problem, which consists of making a point estimate of the parameter by best-fitting to measured data. This way, the subsequent model prediction is conditioned on that point estimate. In the geothermal setting, the computer program ITOUGH2 [17] or PEST [18] implemented this process by employing derivative-based optimisation methods such as the Levenberg-Marquart [19] and its variants to minimize the sum of squares of differences between measured data and observable model outputs. However, geothermal reservoirs often face strong non-linear interactions between model parameters, unstable phase transitions in the governing equations, constraints in thermodynamics, and possibly changing underground structures. This way, the data-misfit function can be characterised by local minima, extensive flat planes, and discontinuities in the multi-dimensional parameter space. These features present significant difficulties for finding the best model fit using derivative-based optimisation methods. Moreover, the point estimate provided by optimisation may not be able to summarise the above-mentioned complex structures of the parameter space. As a result, model predictions issued from the point estimate may not be robust in characterising the range of potential behaviours of the reservoir.

Using the Bayesian framework to formulate the inverse problem enables us to cast all the unknowns as random variables that are consistently modelled by the posterior distribution. This way, we can issue robust parameter estimations and model predictions as marginal distributions of the posterior predictive, quantiles of some statistics, or expectations of quantities of interests. Complement to the Bayesian inference, the use of stochastic algorithms, such as MCMC, can be robust to multi-modality and discontinuity in the parameter space. The main reason is that the convergence of the MCMC and the resulting estimators rely on the ergodicity of the Markov chain, the law of large numbers, and perhaps the central limited theorem (under certain technical conditions). As a result, an MCMC algorithm may be slow to converge, but its convergence can be guaranteed under mild conditions. This way, we can always put in more computing resource to generate more posterior samples to improve the accuracy of resulting estimators. This can overcome many of the difficulties associated with deterministic gradient-based optimisation, such as multi-modality and discontinuity. In many scenarios, the parameter space searching capabilities of many optimisation methods can be turned into MCMC samplers to further improve the performance of MCMC sampling. For example, gradient-based methods [20], Hessian-based methods [9], and differential evolution algorithms [21].

The well-test model used in geothermal reservoir modelling shares many of the difficulties of large-scale reservoir models. In particular, the associated posterior distributions are often multi-modal and have discontinuities, as the result of thermodynamic constraints and model nonlinearity. This does not only cause difficulties to gradient-based optimisations, but also make the MCMC suffer from low acceptance rate. Thus, it becomes a perfect candidate to validate the Bayesian framework for inverse problems and our parallel rejection scheme. In this paper, we apply the parallel rejection to analyse a set of well-test data obtained in the Fushime field, Japan, which is considered as one of the standard test case in the literature [22]. We aim to infer permeability, porosity, relative permeability, capillary pressure, and boundary conditions using the pressure and enthalpy data measured during a discharge test. We also compare our results to a previous estimate given by ITOUGH2 using the same data set [22].

This paper is structured as follows. In Section 2, we present the parallel rejection scheme. In Section 3, we discuss the problem of geothermal model calibration, and its Bayesian treatment including the proposal kernels used in the MH algorithm. Output statistics obtained by sample-based inference are discussed in Section 4, along with a comparison to the calibration given by gradient-based optimisation obtained by [22]. We conclude with a discussion in Section 5.

2. PARALLEL REJECTION SCHEME

Here we introduce the parallel rejection scheme to decrease the average CPU time per MCMC step by parallelising the serial MH algorithm.

The goal is to sample a target probability distribution $\pi_{\text{tar}}(\theta)$, e.g., the posterior, using the Metropolis-Hastings algorithm. The standard MH is an iterative procedure that simulates a sequential Markov chain $\{\Theta_i\}_{i=0}^{\infty}$ with $\pi_{\text{tar}}(\theta)$ as the invariant distribution. To construct the Markov chain, MH needs a (potentially) location-dependent proposal distribution

$$q(\theta'|\Theta_i = \theta_i),$$

where $\Theta_i = \theta_i$ is the current state of the Markov chain and θ' is the proposed state. Then the next state of the Markov chain is set to $\Theta_{i+1} = \theta'$ with the *acceptance probability*

$$\alpha(\theta_i, \theta') = \min \left\{ 1, \frac{\pi_{\text{tar}}(\theta')q(\theta_i|\theta')}{\pi_{\text{tar}}(\theta_i)q(\theta'|\theta_i)} \right\}, \quad (1)$$

or $\Theta_{i+1} = \theta_i$ with probability $1 - \alpha(\theta_i, \theta')$. Under mild technical conditions, the resulting Markov chain $\{\Theta_i\}_{i=0}^{\infty}$ asymptotically converges to the target distribution as the number of MCMC iterations increases. This way, the Markov chain $\{\Theta_i\}_{i=0}^{\infty}$ provides a set of samples following the target distribution.

The construction of Markov chain allows for exploiting the parameter space locally by utilising the proposal distribution based on the previous state that may be already in the high-probability region of the target distribution. This makes MCMC methods robust for sampling high-dimensional and analytically intractable probability distributions that can be hard to explore by other sampling methods such as importance sampling and rejection sampling [23]. We refer to [24,25] and references therein for the analysis of the asymptotic behaviour of MCMC in high-dimensions. However, the Markov structure of the sampling process renders that the MCMC simulation has to be constructed in a sequential manner. This way, it can be challenge to utilise modern parallel computing architectures to accelerate the sampling.

We present a *parallel rejection* scheme to parallelise the acceptance/rejection evaluations of MH. Suppose the Markov chain has current state $\Theta_i = \theta_i$. Suppose that n processors are available and are independent for computing purposes. On each of the processors indexed by $r = 1, 2, \dots, n$, an independent instance of the MH algorithm is simulated with the initial state θ_i . This gives n independent Markov chains $\{Y_{r,k}\}_{k=0}^{\infty}$ with $Y_{r,0} = \theta_i$. The resulting states are enumerated by the global index

$$t(r, k) = r + n(k - 1),$$

where $r = 1, 2, \dots, n$ is the index of the processor and $k = 1, 2, \dots$ is the iteration index of the Markov chain. This leads to the *total ordering* $t = 1, 2, \dots$. Note that t is an invertible function so the states on the parallel computer may be referred to by the total ordering t , i.e. as $\{Y_t\}_{t=0}^{\infty}$. The n parallel chains are simulated until the first non-trivial acceptance (in the ordering t) occurring at t_{\min} , i.e. the minimum t for which $Y_t \neq \theta_i$. Then set $\Theta_j = \theta_i$ for $j = i + 1, i + 2, \dots, i + t_{\min} - 1$ and $\Theta_{i+t_{\min}} = Y_{t_{\min}}$, and re-initialize the n parallel chains.

This algorithm is summarized in Algorithm 1. In practice, all our proposals are non-trivial, so each of the parallel chains can be halted after it has an acceptance, or when it performs an iteration with index t greater than the index of an accepted state in another chain. We note that a similar parallelisation has been previously introduced by (author?) [26] in constructing a synchronous parallel version of simulated annealing for optimisation.

To estimate the performance of this algorithm, we assume that the computing time of MCMC is dominated by the evaluation of the target density, which involves an expensive numerical model simulation. We also assume each acceptance/rejection occurs independently with probability $\bar{\alpha}$ of acceptance. This way, the expected number of MCMC steps for a standard MCMC to reach the first acceptance is about $\bar{\alpha}^{-1}$. In comparison, the probability of having at least one acceptance in the parallel rejection with n processors is about $1 - (1 - \bar{\alpha})^n$. Thus, the expected number of steps for the parallel rejection to reach the first acceptance is about $(1 - (1 - \bar{\alpha})^n)^{-1}$. By considering the computing time per MCMC step is constant, the speed-up factor is

$$\frac{t_s}{t_p} = \frac{1 - (1 - \bar{\alpha})^n}{\bar{\alpha}} \quad (2)$$

Algorithm 1: One iteration of parallel rejection

Data: Current state of the Markov chain $\Theta_i = \Theta_i$

```

1 for Simulate  $n$  Markov chains  $\{Y_{r,k}\}_{k=0}^{\infty}$  in parallel do
2   Initialise the  $r$ -th chain  $Y_{r,0} = \Theta_i$ 
3   Set  $t_r = \infty$ 
4   for  $k = 1, \dots, \infty$  do
5     Generate proposal candidate  $y' \sim q(\cdot | Y_{r,k-1})$ 
6     Compute the acceptance probability
           
$$\alpha(Y_{r,k-1}, \theta') = \min \left\{ 1, \frac{\pi_{\text{tar}}(\theta') q(Y_{r,k-1} | \theta')}{\pi_{\text{tar}}(Y_{r,k-1}) q(\theta' | Y_{r,k-1})} \right\}$$

7     Simulate a uniform random number  $U \sim \text{uniform}(0, 1)$ 
8     if  $U < \alpha(Y_{r,k-1}, y')$  then
9       Accept  $y'$  by  $Y_{r,k} = y'$ 
10      Set  $t_r = r + n(k - 1)$ 
11      Stop the  $r$ -th Markov chain simulation
12    else
13      Reject  $y'$  by  $Y_{r,k} = Y_{r,k-1}$ 
14    end
15    Compute  $t_{\min} = \min(t_1, t_2, \dots, t_n)$ 
16    if  $t_{\min} \leq r + n(k - 1)$  then
17      Stop the  $r$ -th Markov chain simulation
18    end
19  end
20 end
21 Set  $X_j = \Theta_i$  for  $j = i + 1, i + 2, \dots, i + t_{\min} - 1$  and  $X_{i+t_{\min}} = Y_{t_{\min}}$ 

```

where t_s and t_p are the time per update for serial and parallel Markov chain, respectively.

In our applications and potentially some other engineering applications, setting up each geothermal reservoir model simulation requires disk accesses (for accessing data files and simulator specifications) that cannot be parallelised. This *transaction overhead* makes the computing time of setting up parallel simulations linearly increases with the number of processors. Thus, we have to consider an alternative speed-up estimate that contains the transaction overhead. Let t_T denotes the computing time spent on evaluating the model (and hence the target density) and ρ be the ratio between the transaction time per process and the time per model evaluation. This way, each step of the standard MCMC costs $(1 + \rho)t_T$ computing time, whereas each step of the parallel rejection costs $(1 + n\rho)t_T$ computing time by considering the linear increase in transaction time. This leads to a reduction in the speed-up factor, which has the form

$$\frac{t_s}{t_p} = \frac{(1 - (1 - \alpha)^{n-1})}{\alpha} \frac{1 + \rho}{1 + n\rho}. \quad (3)$$

The parallel rejection is more effective in scenarios where the acceptance rate of MCMC is low and the transaction overhead is small. Figure 1(a) shows the theoretical speed-up factor for relative transaction time $\rho = 0.2$ (this is approximately the value we measured on the cluster we used), using $n = 1, 2, \dots, 19$ processors, with average acceptance rates $\bar{\alpha} = \{0.01, 0.03, 0.05, 0.1, 0.3, 0.5\}$. Note that the speed-up factor actually decreases when the acceptance rate is high and the number of processors is large.

To study the geothermal model calibration problem (see Section 3), we implemented the parallel rejection scheme on a cluster, in which each node consisted of several CPUs (2 or 4). The scalable parallel random number generators library [27] was used for random number generation. All the nodes on our cluster shared the same hard disk which

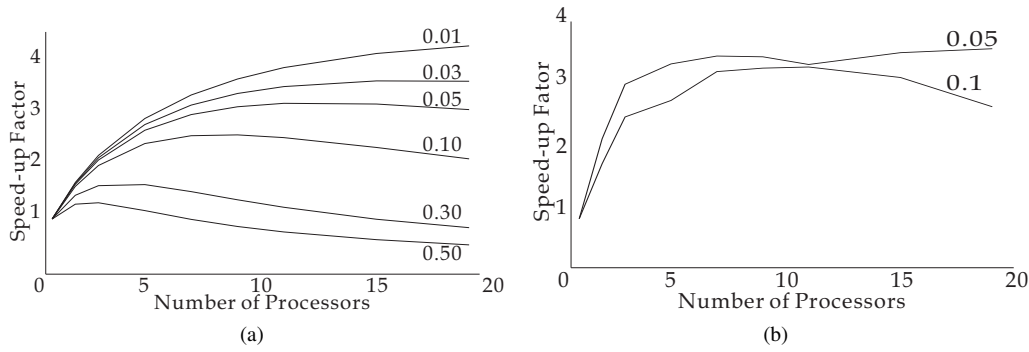


FIG. 1: Speed-up factor for the parallel rejection scheme. (a) Theoretical speed-up factor for average acceptance rates $\alpha = 0.01, 0.03, 0.05, 0.1, 0.3, 0.5$ with $\rho = 0.2$. (b) Measured speed-up factor found on the computer used. The average acceptance rates are $\alpha = 0.05$ and 0.10 .

had timing implications since our requires disk input/output at each time step to record thermodynamics information. Hence we could not utilise all the computing resource as a large number of processes running at the same time compete with each other for disk resources. We measured performance by using up to 19 CPUs with proposal distributions set to give an acceptance rate of 5% and 10% ($\pm 1\%$). Figure 1(b) shows that the measured speed-ups are consistent with our theoretical prediction. In this example, using five processors can give a roughly three times speed up.

The limitation on performance at greater number of processors is mainly due to the large transaction time compared to the time required to simulate the reservoir model. When the transaction fraction is small, the potential speed-up factor can be greater. Figure 2 gives a theoretical prediction of speed up factor for the case $\rho = 0.01$. We can observe that, with low target acceptance rate (0.03), the parallel rejection can speed up the MH algorithm by a factor of 15 when 30 processes are used. Such a low acceptance rate is not uncommon in applications of MCMC to large-

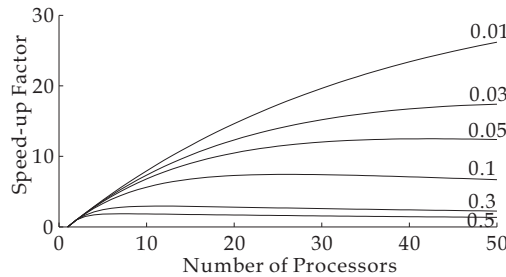


FIG. 2: Speed-up factor for the parallel rejection scheme with $\rho = 0.01$. The average acceptance rates shown are $\alpha = 0.01, 0.03, 0.05, 0.1, 0.3, 0.5$.

scale reservoir model, for example, [28]. So the parallel rejection makes a useful contribution in which computation of the numerical model needs no modification, as long as the relative transaction time can be kept small.

3. BAYESIAN INVERSION FOR GEOTHERMAL MODELS

3.1 Data Simulation

Consider a geothermal reservoir in a subdomain Ω with bounding surface $\partial\Omega$. Here we consider a two phase flow (water and vapour) modelled by general mass balance and energy balance equations in the form of

$$\frac{d}{dt} \int_{\Omega} M dV = \int_{\partial\Omega} Q \cdot \hat{n} d\Gamma + \int_{\Omega} q dV. \quad (4)$$

The accumulation term M represents mass (M_m) or energy (M_e) per unit volume, defined by

$$\begin{aligned} M_m &= \phi (\rho_l S_l + \rho_v S_v), \\ M_e &= (1 - \phi) \rho_r c_r T + \phi (\rho_l u_l S_l + \rho_v u_v S_v). \end{aligned} \quad (5)$$

Here ϕ is porosity, ρ is density, S is saturation, u is specific internal energy, c is specific heat and T is the temperature. The subscripts l, v and r indicate the quantities pertaining to liquid, vapour and rock, respectively. Note that the liquid saturation and vapour saturation sum to one, i.e., $S_l + S_v = 1$.

The mass flux term is given as a sum over liquid and vapour phases:

$$Q_m = \sum_{\beta=l,v} \frac{k k_{r\beta}}{\nu_\beta} (\nabla p - \rho_\beta \vec{g}). \quad (6)$$

Here k is a diagonal second order permeability tensor in 3-dimensions, \vec{g} represents the acceleration due to gravity, ∇p is the pressure gradient acting on the fluid flow, and ν_β is the kinematic viscosities. Relative permeability $k_{r\beta}$ is introduced to account for the interference between liquid and vapour phases as they move through the rock matrix in the geothermal reservoir. There are several empirically derived curves available to model $k_{r\beta}$ as functions of vapour saturation S_v . We use the van Genuchten-Mualem model [29], defined by:

$$k_{rl} = \begin{cases} \sqrt{S^*} \left\{ 1 - \left(1 - [S^*]^{1/m} \right)^m \right\}^2 & \text{if } S_l < S_{ls} \\ 1 & \text{if } S_l \geq S_{ls}, \end{cases} \quad (7)$$

where

$$\begin{aligned} S^* &= (S_l - S_{rl}) / (S_{ls} - S_{rl}); \\ k_{rv} &= \begin{cases} 1 - k_{rl} & \text{if } S_{rv} = 0 \\ (1 - \hat{S})^2 (1 - \hat{S}^2) & \text{if } S_{rv} > 0, \end{cases} ; \\ \hat{S} &= (S_l - S_{rl}) / (1 - S_{rl} - S_{rv}); \end{aligned}$$

subject to the restrictions

$$\begin{aligned} 0 &\leq k_{rl}, k_{rv} \leq 1, \\ S_{rl} + S_{rv} &< 1, \\ S_{rl} &< S_{ls}. \end{aligned} \quad (8)$$

Except for the hyperparameters m , S_{rl} , S_{ls} and S_{rv} , all the remaining parameters in the van Genuchten-Mualem model are components of the state of the geothermal system. In addition, we assume the hyperparameter S_{rv} takes value zero.

Energy is carried by the movement of vapour and liquid and by thermal conduction. This way, the energy flux is given by

$$Q_e = \sum_{\beta=l,v} \frac{k k_{r\beta}}{\nu_\beta} (\nabla p - \rho_\beta \vec{g}) h_\beta - K \nabla T. \quad (9)$$

Here h denotes specific enthalpy and K is the thermal conductivity in a saturated medium.

The accumulation term q in (4) represents the mass and heat sources or sinks in region Ω . The thermophysical properties of liquid and vapour (such as density, viscosity, specific internal energy, specific heat and specific enthalpy) are calculated using steam table equations given by the International Formulation Committee [30].

Spatial discretization of (4) is based on an integrated finite difference or finite volume method, which is implemented in the existing Fortran code TOUGH2 [31]. To guarantee the numerical stability of simulation, TOUGH2 uses a fully implicit method for numerical integration in time, and upstream weighting for calculating the velocity of fluid

movement between adjacent blocks (in (6) and (9)). For each time step, the Newton-Raphson method is used to solve the resulting system of non-linear difference equations. TOUGH2 uses a preconditioned iterative sparse matrix solver for solving the linear equations at each Newton-Raphson iteration.

As discussed in the introduction, we aim to sample the posterior distribution over field parameters conditioned on pressure data and flowing enthalpy data measured during a well discharge test. We model the geothermal field as a single layer radially symmetric model with homogeneous rock properties. The model is located at a depth of about 1600 meters which corresponds to the feed zone of the well. To balance the numerical accuracy and computational cost, we designed a specific grid for our well-testing purpose. Figure 3 shows a wedge of the computational grid of the model. The well is located in the central block shown at the vertex of the wedge, 10 small blocks are used adjacent to the well, followed by 65 blocks with a thickness expansion factor of 1.2.

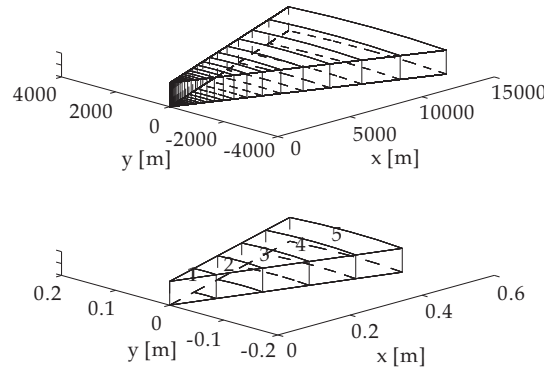


FIG. 3: Finite volume grid for the well discharge test model. The top figure gives the overview of (a wedge from) the single-layered one-dimensional radially-symmetric model. The bottom figure shows the first five blocks; the well located in the first block.

The numerical simulation generates ‘down hole’ values whereas observed data are measured at the well head. While the energy loss and enthalpy change along the well bore are typically not significant compared to the measurement noise, the pressure drop along the well bore can be significant. Accordingly, we introduce a pressure shift parameter variable p_s to model this difference. Since the model is assumed to be homogeneous, there is only one porosity value and one permeability value in the radial direction that need to be estimated. The initial vapour saturation (S_{v0}) and initial pressure (p_0) are used to represent the initial thermodynamics state of the two phase system, but are not known in advance. Together with the parameters from the relative permeability model, these make up the eight unknown parameters for the data simulation:

$$\theta = (\phi, \log_{10}(k), p_0, S_{v0}, p_s, m, S_{rl}, S_{ls}). \quad (10)$$

Note that the permeability k is represented on a base 10 logarithmic scale.

3.2 Observed Data and Likelihood Function

We analyse a measured data set in which production rate, flowing enthalpy, and wellhead pressure data were observed during a period of 140 days. The data is shown in Figure 4(a). The well was shut down three times (see Figure 4(a)), for 2 days around day 50, one day at day 55, and for 5 days around day 100. After day 105, the production rate was increased from about (minus) 4 kg/s to about (minus) 10 kg/s.

Due to the nonlinearity and physical constraints, our model (or well-test models in general) has limited capability in analysing the full data set. Thus, we select several subsets of data on to push the limit of well-test models and our proposed algorithm. In our model, the simulation could not continue with a low permeability value ($k < 1$ millidarcy) after day 105 because the model pressure dropped below atmospheric pressure which is not physical. As

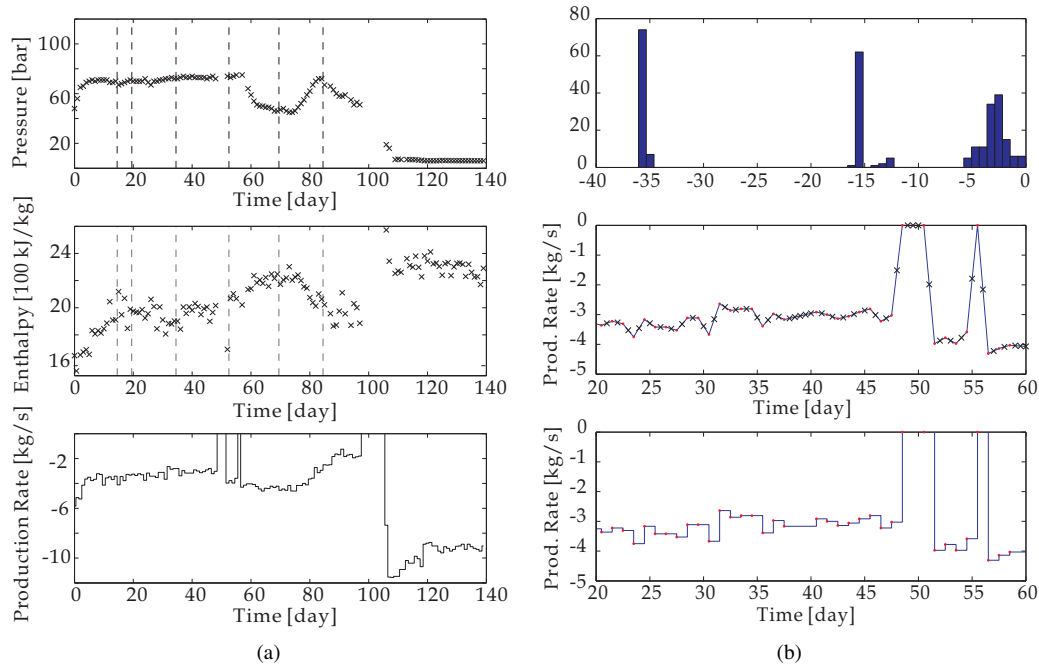


FIG. 4: (a) Data used for the well discharge test analysis. The top, middle and bottom plot corresponding to the pressure, the flowing enthalpy and the production rate, respectively. (b) Production rate from day 20 to day 60. Top: histogram of second differences. Middle: traditional linear interpolation between data points; interpolated data points (see text) are shown by crosses. Bottom: interpolation of measured production rate used in this study.

shown later, to match the data before day 105, the feasible range of permeability is usually below 1 millidarcy (i.e. $\log_{10}(k) < -15$). Figure 4(a) shows that the production rate drops between day 85 to day 95, while the pressure drops in this period as well. This effect may be caused by interaction with other wells or changes in the subsurface environment, which is beyond the capability of well-test model. Further, TOUGH2 is not able to reproduce the steep rise in the enthalpy data and the increase in the pressure data from day 0 to day 10, which can be caused by storage effect of well-bores. Thus, our analysis is based on the data collected from day 14 to day 84.

The production rate q_m are also recorded during the well discharge test at 24 hour intervals. The rate change within this interval is not specified. We found that the production rate at some points had been linearly interpolated from the adjacent measured data points. The top plot of Figure 4(b) is the histogram of the second difference of production rate data with respect to time, $\Delta^2 q_m$ shown on a base 10 logarithmic scale. Data points with second difference less than 10^{-10} are certainly the product of linear interpolation of neighbouring measured points, hence contribute no extra information. The middle plot of Figure 4(b) shows the original production rate (from day 20 to day 60), the crosses are the data points with a small second difference ($< 10^{-10}$). These points are discarded in our analysis, so do not contribute to our analysis.

The middle plot in Figure 4(b) also shows the traditional *linear* interpolation of production rate between measured data points. In a previous study, [22] analyzed the same data set using a linearly interpolated production rate; the results are shown later in Figure 8. From those plots we observe a sudden pressure and enthalpy change around day 55 resulting from the shutting down of the well around day 55 (see Figure 4(b)). In our analysis we found that when the permeability is low, the use of linear interpolation of production rate may shift down the enthalpy and bring the pressure up after day 55. This effect can be overcome by an alternative interpolation which treats the data points (bottom plot of Figure 4(b)) as the value of a step function. That interpolation is consistent with the interpretation that production rate is *set* for each 24 hours rather than being a measurement of a smoothly varying quantity.

We denote the observed data as $\tilde{\mathbf{y}} = (\tilde{\mathbf{h}}, \tilde{\mathbf{p}})$, where $\tilde{\mathbf{h}}$ and $\tilde{\mathbf{p}}$ are observed flowing enthalpy and well head

pressure, respectively. Conditional on the unknown parameter θ , we specify a Gaussian measurement error of the form:

$$L(\tilde{\mathbf{y}}|\theta) \propto \exp \left\{ -\frac{1}{2} [(\tilde{\mathbf{h}} - \mathbf{h}(\theta))^T \Sigma_h^{-1} (\tilde{\mathbf{h}} - \mathbf{h}(\theta))] \right\} \\ \times \exp \left\{ -\frac{1}{2} [(\tilde{\mathbf{p}} - \mathbf{p}(\theta))^T \Sigma_p^{-1} (\tilde{\mathbf{p}} - \mathbf{p}(\theta))] \right\}, \quad (11)$$

where $\mathbf{h}(\theta)$ and $\mathbf{p}(\theta)$ are the observable model outputs for parameter θ . The covariances of flowing enthalpy Σ_h and covariances of well head pressure Σ_p are fixed constant. Because the data observations are fairly evenly spaced, we assume that $\Sigma_h = \sigma_h^2 I_n$ and $\Sigma_p = \sigma_p^2 I_n$. The actual choice of standard deviations σ_h and σ_p are estimated from the residual of a smoothed spline interpolation of the time series data $\tilde{\mathbf{h}}$ and $\tilde{\mathbf{p}}$.

We divide the observed data from day 14 to day 84 into 5 sets (see Figure 4(a)). We perform two analyses using different partitions of observations for training with the remaining observations used for validation. Also, an analysis with all the observations used for training is conducted to compare with the results of the previous study by [22]. Table 1 shows the three scenarios we analyzed and the observations used for each scenario.

TABLE 1: The training data subsets for each of the three scenarios used to analyze the well discharge test. Columns labelled σ_h and σ_p give the estimated standard deviations of flowing enthalpy and well head pressure, respectively.

Scenario	Training Data	σ_h [kJ/kg]	σ_p [bar]
S1	{14-19, 35-52}	50	3
S2	{14-19, 53-69}	50	3
S3	{14-84}	100	6

3.3 Prior Modelling

We used expert knowledge and physical constraints to define bounds on the allowable range of each component of the parameter θ , and within that defined weak subjective prior densities. For example, the value of porosity is necessarily between 0 and 1, and commonly less than 0.5. To formulate such weak prior information into a functional form, we use a modified exponential distribution of the form

$$\pi(x) \propto \frac{x}{s} \left(1 - \frac{x}{s}\right) \exp\left(\iota \frac{x}{s}\right). \quad (12)$$

Here s scales the parameters into the range (0, 1) and ι is a non-zero constant which control the skewness of the distribution, setting the region of the mode. The density for various choices of ι are shown in Figure 5.

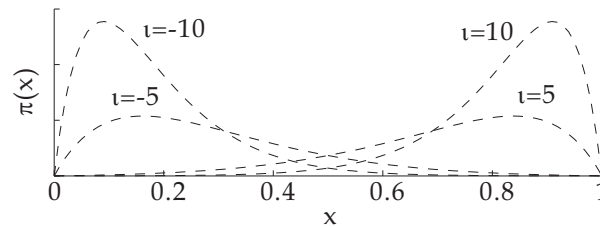


FIG. 5: Plot of $\pi(x)$ for different choice of ι , where $\iota = -10, -5, 5, 10$.

Table 2 summaries the physical constraints and bounds that we used for each component of the parameter θ , and the different choices of ι for each component, except the permeability k . Note that the constraints (8) in the van Genuchten-Mualem model is also included in the prior distribution. We make the prior density over permeability invariant with respect to changes of unit by specifying a uniform density over a logarithmic scale [32]. It is not necessary to truncate the range of allowable permeability values to make the prior proper as the range is effectively constrained by the likelihood function.

TABLE 2: The bounds for different components of parameter θ and the different choice of skewness parameter ι .

	ϕ [-]	m [-]	S_{rl} [-]	S_{ls} [-]	S_{v0} [-]	p_0 [bar]	p_s [bar]
U.B.	1	1	1	1	1	250	150
L.B.	0	0	0	0	0	50	50
ι	-5	5	2	10	-10	-2	-2

3.4 Posterior Exploration

We explore the posterior distribution

$$\pi(\theta|\tilde{\mathbf{y}}) \propto L(\tilde{\mathbf{y}}|\theta)\pi_{\text{pr}}(\theta)$$

by MCMC sampling using MH dynamics, paralised by the parallel rejection scheme in Algorithm 1. Because of physical bounds and multi-modal behaviour of the posterior, automatically tuned MCMC proposals such as the Adaptive Metropolis [33] does not perform well in our examples. We designed a mixture proposal distribution to propose a new candidate parameter θ' . Suppose we have L different proposal distributions $q^{(i)}(\theta'|\theta)$, $i = 1, \dots, L$ and let ζ_i , $i = 1, \dots, L$, be the probability of choosing a proposal distributions $q^{(i)}$. The overall proposal distribution is given by

$$q(\theta', \theta) = \sum_{i=1}^L \zeta_i q^{(i)}(\theta'|\theta). \quad (13)$$

To ensure good mixing and ergodicity of the Markov chain, we specifically design the following three types of proposals. Each proposal we select a subset θ_I of components of the parameter θ to update, where I represents a set of indices. The proposals are:

- **Translate:** Translate θ_I by $\lambda \sim U(-\epsilon, \epsilon)$: $\theta'_I = \theta_I + \lambda$.
- **Scale:** Scale θ_I by $\lambda \sim U(1/\epsilon, \epsilon)$: $\theta'_I = \lambda\theta_I$.
- **Shoot:** Draw a two dimensional random vector

$$\lambda \sim U(-\epsilon_1, \epsilon_1) \times U(-\epsilon_2, \epsilon_2),$$

then update the two dimensional vector θ_I by

$$\theta'_I = \theta_I + \begin{pmatrix} \cos(\gamma) & \sin(\gamma) \\ -\sin(\gamma) & \cos(\gamma) \end{pmatrix} \lambda,$$

where γ is a given angle.

The translation proposal operates on one randomly selected component from $\{\log_{10}(k), S_{rl}, \phi\}$, which is selected with probability $\{0.4, 0.4, 0.2\}$. The scaling proposal is particular useful for handling parameters that are more likely to have values towards to physical bounds. To operate with the scaling proposals, we randomly selects one parameter from $\{\phi, m\}$ with probability $\{0.3, 0.7\}$. The shooting proposal is designed to capture correlations between parameters. Here we use it to update a two dimensional subset of parameters selected uniformly at random from $\{\{p_0, p_s\}, \{S_{ls}, S_{v0}\}\}$, and a shooting angle of $\gamma = \pi/4$ is used for both subsets. The Jacobian determinant in the MH algorithm for translate and shoot proposals is 1, and is λ^{-2} for the scale proposal. We select one of the proposals with probability $\{0.45, 0.2, 0.35\}$. These three kernels are sufficient to ensure the resulting Markov chain is aperiodic and irreducible, and hence standard results guarantee that the resulting reversible chain has the desired ergodic properties.

4. COMPUTED RESULTS

We ran the MCMC simulation for about 60,000 iterations for each of the scenarios given in section 3, using the MacLaurin Cluster at the University of Auckland. The computing time for scenario S1, S2 and S3 were roughly 20 hours, 60 hours and 90 hours, respectively. Figure 6, 7 and 8 shows summary statistics of predicted well head pressure and flowing enthalpy for each scenario. Here we provide the mean, 5% quantiles and 95% quantiles of the predictions at each time. The posterior means are shown as a solid line, while the 90% credible interval bounded by 5% and 95% quantiles is shown by the shaded region.

The squares in Figure 6, 7 and 8 indicate data used for training, while triangles are data used for validation. Scenario S1 produces a larger error band in both pressure and enthalpy predictions after day 52, and both predictions are flatter than the observed data. The maximum mismatch in pressure and enthalpy are about 20 bars (too high) and 200 kJ/kg (too low), respectively. Scenario S2 used the data between day 53 and day 69 for training which has larger variation; Correspondingly S2 produces a much tighter predictive interval that covers the observed data better than S1. Since all the observed data points are used for training in scenario S3, that scenario effectively implements model-based smoothing and provides an even tighter prediction interval than S2 with the model output uncertainty intervals covering the observed data reasonably well.

Figure 6, 7 and 8 show the marginal distributions for each component of θ , and show that there exists strong negative correlation between initial vapour saturation (S_{v0}) and S_{ls} . In the scenario S1, there is some negative correlation between $\log_{10}(k)$ and p_0 and strong negative correlation between p_0 and p_s . In the scenario S2, both pairs $\log_{10}(k), p_0$ and p_0, p_s are strongly correlated, whereas $\log_{10}(k), p_0$ and p_s are negatively correlated in scenario S3.

The summary statistics for these three scenarios show that the mean prediction and uncertainty intervals are different when different training data sets are used. Together with the plots of the marginal distributions for each component of θ (Figure 6, 7 and 8), we can conclude that the posterior distribution conditioned on the different training data sets have different shapes, and modes in different regions of the parameter space, even though the data sets are measured from the same physical process. That clearly indicates that there is appreciable modelling error, i.e., that there is a discrepancy between the computer model used for simulation and the physical process being measured.

We also compare our analysis with the previous study by [22] who calibrated the same TOUGH2 model based on the the same observed data from day 10 to day 85 (our scenario S3) using gradient-based optimisation implemented in ITOUGH2. The least-squares estimate obtained by [22] is shown by the red lines in Figure 8. We note here the optima reported shown in Figure 8 is obtained by initialising gradient-based optimisation at multiple initial guesses and choosing the one with minimal data misfit. We can observe that the least-squares estimates of permeability $\log_{10}(k)$, the van Genuchten parameter m , and the constant pressure shift p_s all lie out of the significant support of their marginal distributions. In this example, the point estimate obtained by optimisation does not lie in the high-probability region of the posterior, and hence may not be able to provide robust model predictions.

We note that the gradient-based optimisation using here computes the derivatives using the finite difference method, as the forward model does not have adjoint capability. In this example, computing the least square estimate can take more computing time than using sample-based inference.

5. DISCUSSION

In this paper we used a practical sample-based method to calibrate a well test model of a geothermal reservoir. The parallel rejection version of the Metropolis-Hastings algorithm was introduced to decrease the CPU time per acceptance in MCMC sampling.

To make model predictions and to be able to quantify the prediction uncertainty, the posterior distribution is used to characterise the parameters uncertainty conditioned on observed data. Our sample-based predictive density shows a more robust and accurate result than the traditional least-squares estimate, and the MCMC sampling is not limited by the difficulties associated with gradient based optimisation such as local minima. Using this example, we have also demonstrated that using the parallel rejection can speed up MCMC sampling for problems where MCMC suffers from low acceptance rate. However, the efficiency of the parallelisation is limited by the simulator used in this investigation, as the non-parallelisable transaction overhead is high. As shown in the performance analysis, we do expect

the efficiency of the parallel rejection can be further improved for applications with less non-parallelisable transaction overhead. The parallelisation introduced here can also be integrate with surrogate modelling techniques such as the collocation methods [34–36] and projection based model reduction [37,38] to further enhance the performance of sample-based solution to Bayesian inverse problems.

By comparing the three scenarios that we analysed, we find that the inversion result is significantly improved when data collected between day 53 and day 69 are used. This set of data corresponds to an increased production rate during the same period, the production rate (Figure 4(b)) is increased from about (minus) 3 kg/s to more than (minus) 4 kg/s after day 52. If all the training data is from one regime (in scenario S1, all the data are measured with low production rate), the model could not reproduce the system behaviour across different regimes, i.e., give predictions after day 53. When data sets from both regimes are used (as in scenario S2) the solution behaves more like an average through the different regimes, hence it could follow the trend of observed data well. This suggests that including training data from different regimes is more important than the amount of data used in training. This behaviour also indicates that our computer model does not completely mimic the physical system. This will lead to future opportunities of investigating the role of model error in Bayesian inversion.

REFERENCES

1. Higdon, D., Lee, H., and B., Z., A bayesian approach to characterizing uncertainty in inverse problems using coarse and fine-scale information, *IEEE Transactions on Signal Processing*, 50(2):389–399, 2002.
2. Andersen, K.E., Brooks, S.P., and Hansen, M.B., Bayesian inversion of geoelectrical resistivity data, *Journal of the Royal Statistical Society: Series B*, 65:619–644, 2004.
3. Cui, T., Fox, C., and O'Sullivan, M.J., Bayesian calibration of a large-scale geothermal reservoir model by a new adaptive delayed acceptance Metropolis Hastings algorithm, *Water Resource Research*, 47:W10521, 2011.
4. Nicholls, G.K. and Fox, C., Prior modelling and posterior sampling in impedance imaging, In *Bayesian Inference for Inverse Problems*, Vol. 3459, pp. 116–127. SPIE, 1998.
5. Vauhkonen, P., Vauhkonen, M., Savolainen, T., and Kaipio, J., Three-dimensional electrical impedance tomography based on the complete electrode model, *IEEE Transactions on Biomedical Engineering*, 46:1150–1160, 1999.
6. Watzenig, D. and Fox, C., A review of statistical modelling and inference for electrical capacitance tomography, *Measurement Science and Technology*, 20(5), 2009.
7. McKeague, I.W., Nicholls, K., G., Speer, K., and Herbei, R., Statistical inversion of South Atlantic circulation in an abyssal neutral density layer, *Journal of Marine Research*, 63(4):in press, 2005.
8. Haario, H., Laine, M., Lehtinen, M., Saksman, E., and Tamminen, J., Markov chain Monte Carlo methods for high dimensional inversion in remote sensing, *Journal of the Royal Statistical Society: Series B*, 66:591–608, 2004.
9. Martin, J., Wilcox, L.C., Burstedde, C., and Ghattas, O., A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion, *SIAM Journal on Scientific Computing*, 34(3):A1460–A1487, 2012.
10. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E., Equation of state calculations by fast computing machines, *Journal of chemical physics*, 21:1087–1092, 1953.
11. Hastings, W., Monte Carlo sampling using Markov chains and their applications, *Biometrika*, 57:97–109, 1970.
12. Geyer, C.J., Markov chain Monte Carlo maximum likelihood, In: Keramigas, E. (Ed.), *Computing Science and Statistics: 23rd symposium on the interface*, Vol. 5, pp. 156–163, Interface Foundation, Fairfax, 1991.
13. Chen, Y., Keyes, D., Law, K.J.H., and Ltaief, H., Accelerated dimension-independent adaptive metropolis, *SIAM Journal on Scientific Computing*, 38(5):S539–S565, 2016.
14. Craiu, R.V., Rosenthal, J., and Yang, C., Learn from thy neighbor: Parallel-chain and regional adaptive mcmc, *Journal of the American Statistical Association*, 104(488):1454–1466, 2009.
15. Solonen, A., Ollinaho, P., Laine, M., Haario, H., Tamminen, J., and Järvinen, H., Efficient mcmc for climate model parameter estimation: Parallel adaptive chains and early rejection, *Bayesian Analysis*, 7 (3):715–736, 2012.
16. Calderhead, B., A general construction for parallelizing metropolis- hasting algorithms, *Proceedings of the National Academy of Sciences*, 111(49):17408–17413, 2014.

17. Finsterle, S. *ITOUGH2 User's Guide*. Lawrence Berkeley National Laboratory, Berkeley, California, 1993.
18. Doherty, J., Groundwater model calibration using pilot points and regularisation, *Ground Water*, 41:170–177, 2003.
19. Marquardt, D.W., An algorithm for least square estimation of nonlinear parameters, *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.
20. Roberts, G.O. and Stramer, S., Langevin diffusions and Metropolis-Hastings algorithms, *Methodology and Computing in Applied Probability*, 4(4):337–357, 2002.
21. Braak, C.J.F.T., A markov chain monte carlo version of the genetic algorithm differential evolution: easy bayesian computing for real parameter spaces, *Statistics and Computing*, 16:239–249, 2006.
22. Finsterle, S., Pruess, K., Bullivant, D., and O'Sullivan, M.J., Application of inverse modeling to geothermal reservoir simulation, In *Twenty-Second Workshop on Geothermal Reservoir Engineering*, pp. 309–316, Stanford University, Stanford, California, 1997.
23. Liu, J.S., *Monte Carlo Strategies in Scientific Computing*, Springer-Verlag, 2001.
24. Roberts, G.O. and Rosenthal, J.S., Optimal scaling for various metropolis-hastings algorithms, *Statistical Science*, 16:351–367, 2001.
25. Roberts, G.O., Gelman, A., and Gilks, W.R., Weak convergence and optimal scaling of random walk metropolis algorithms, *Annals of Applied Probability*, 7:110–120, 1997.
26. Sohn, A., Parallel n-ary speculative computation of simulated annealing, *IEEE Transactions on Parallel Distributed System*, 6(10):997–1005, 1995.
27. Mascagni, M. and Srinivasan, A., Algorithm 806: Sprng: A scalable library for pseudorandom number generation, *ACM Transactions on Mathematical Software*, 26:436–461, 2000.
28. Efendiev, Y., Thomas, H., and Luo, W., Preconditioning markov chain monte carlo simulations using coarse-scale models, *SIAM Journal on Scientific Computing*, 28:776–803, 2006.
29. van Genuchten, M.T., A closed-form equation for predicting the hydraulic conductivity of unsaturated soils, *Soil Science Society of America Journal*, 44:892–898, 1980.
30. International Formulation Committee, *A Formulation of the Thermodynamic Properties of Ordinary Water Substance*, IFC Secretariat, 1967.
31. Pruess, K. *TOUGH2 - A General-Purpose Numerical Simulator for Multiphase Fluid and Heat Flow*. Lawrence Berkeley National Laboratory, Berkeley, California, 1991.
32. Jaynes, E.T., Prior probability, *IEEE Transactions on System Science and Cybernetics*, sec-4 (3):227–241, 1968.
33. Haario, H., Saksman, E., and Tamminen, J., An adaptive metropolis algorithm, *Bernoulli*, 7:223–242, 2001.
34. Narayan, A., Jakeman, J., and Zhou, T., A christoffel function weighted least squares algorithm for collocation approximations, *Math. Comput.*, 86:1913–1947, 2017.
35. Yan, L. and Zhou, T., An adaptive multi-fidelity pc-based ensemble kalman inversion for inverse problems, *International Journal for Uncertainty Quantification*, to appear, 2019.
36. Yan, L. and Zhou, T., Adaptive multi-fidelity polynomial chaos approach to bayesian inference in inverse problems, *J. Comput. Phys.*, 2019.
37. Cui, T., Marzouk, Y.M., and Willcox, K.E., Data-driven model reduction for the bayesian solution of inverse problems, *International Journal for Numerical Methods in Engineering*, 102:966–990, 2015.
38. Cui, T., Marzouk, Y.M., and Willcox, K.E., Scalable posterior approximations for large-scale bayesian inverse problems via likelihood-informed parameter and state reduction, *Journal of Computational Physics*, 315:363–387, 2016.

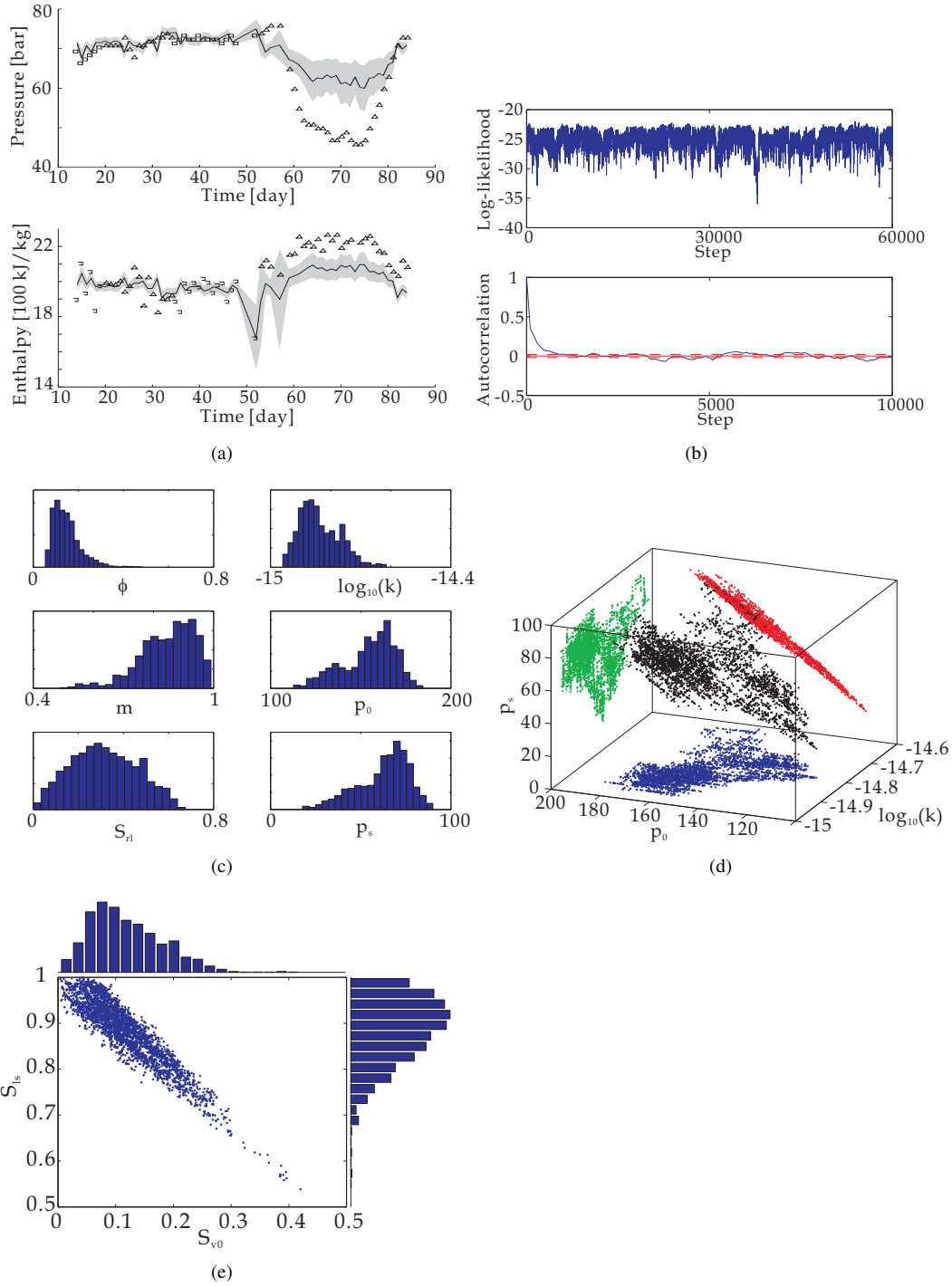


FIG. 6: Results and output analysis for scenario S1. (a) The prediction result: pressure (top) and flowing enthalpy (bottom). The black lines are mean prediction and the gray bands are credible intervals. The squares and triangles are training data points and validating data points, respectively. (b) Trace of log-likelihood (top) and autocorrelation of log-likelihood (bottom). (c) Histogram of the marginal distribution for ϕ , m , S_{rl} , $\log_{10}(k)$, p_0 and p_s . (d) Scatter plot of the joint marginal distribution over $\log_{10}(k)$, p_0 and p_s . (e) Histogram for the marginal distribution of S_{v0} and S_{ls} and scatter plot of their joint marginal distribution.

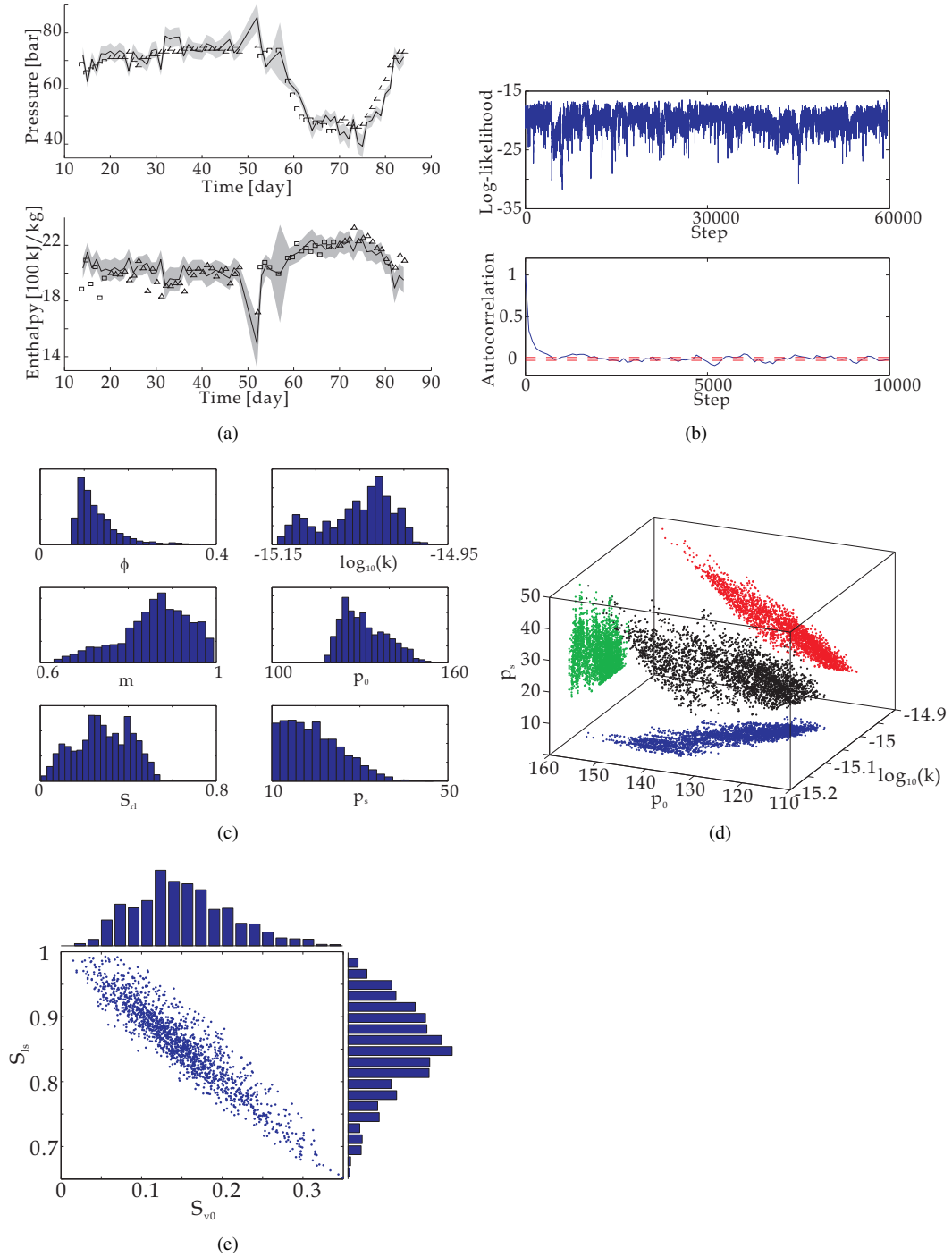


FIG. 7: Results and output analysis for scenario S2, with same plots as Figure 6.

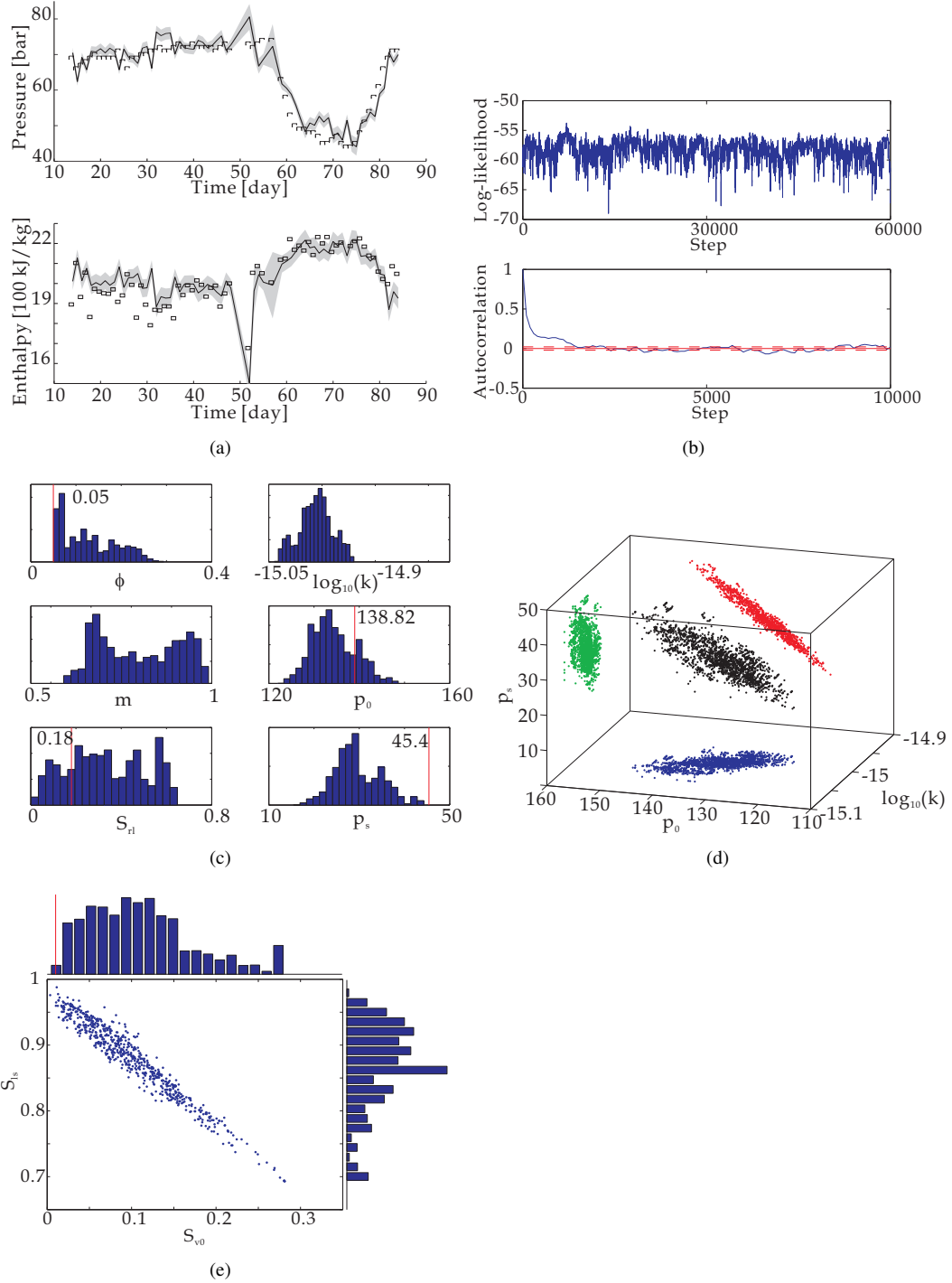


FIG. 8: Results and output analysis for scenario S3, with same plots as Figure 6. The red lines in the histograms in (c) and (d) mark the least-squares estimates reported by [22]. The least-squares estimates they reported for permeability $\log_{10}(k)$ was -14.48, and for m was 0.408; these estimates lie off the ordinate range of the histograms. They did not report the least-squares estimate for S_{ls} .