Efficient Bayesian inference of subsurface flow models using nested sampling and sparse polynomial chaos surrogates

Ahmed H. Elsheikh, Ibrahim Hoteit, Mary F. Wheeler

Center for Subsurface Modeling (CSM), Institute for Computational Engineering and Sciences (ICES), University of Texas at Austin, TX, USA
Institute of Petroleum Engineering (IPE), Heriot-Watt University, Edinburgh Campus, Edinburgh, EH14 1AS, Scotland, UK
Department of Earth Sciences and Engineering, King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia

Abstract

An efficient Bayesian calibration method based on the nested sampling (NS) algorithm and non-intrusive polynomial chaos method is presented. Nested sampling is a Bayesian sampling algorithm that builds a discrete representation of the posterior distributions by iteratively refocusing a set of samples to high likelihood regions. NS allows representing the posterior probability density function (PDF) with a smaller number of samples and reduces the curse of dimensionality effects. The main difficulty of the NS algorithm is in the constrained sampling step which is commonly performed using a random walk Markov Chain Monte-Carlo (MCMC) algorithm. In this work, we perform a two-stage sampling using a polynomial chaos response surface to filter out rejected samples in the Markov Chain Monte-Carlo method. The combined use of nested sampling and the two-stage MCMC based on approximate response surfaces provides significant computational gains in terms of the number of simulation runs. The proposed algorithm is applied for calibration and model selection of subsurface flow models.

Crown Copyright © 2013 Published by Elsevier B.V. All rights reserved.

1. Introduction

Subsurface flow models rely on many parameters that cannot be directly measured. Instead, a sparse set of measurements may be available at the wells locations. The complete distributions of these unknown fields are commonly inferred by a model calibration process that takes into account historical records of the model input–output. During model calibration, the parameters are adjusted to minimize the difference between the model output and the measured time dependent data. Automated calibration is essential for accurate prediction of groundwater flow models in order to understand the fate of subsurface contaminants [1,2]. The multiphase flow of hydrocarbons in an oil reservoir is another example where model calibration is needed for accurate predictions [3,4]. However, the amount of available data to constrain the models is usually limited in both quantity and quality. This results in an ill-posed inverse problem that might admit many different solutions. Different parameter estimation techniques can be applied to tackle this problem. These techniques can be classified into Bayesian methods based on Markov Chain Monte Carlo (MCMC) [5–7], gradient based optimization methods [1,2,8], stochastic search algorithms [9–13] and ensemble Kalman filter methods [14,3,4,15,16].
The Bayesian framework is the most general framework for uncertainty quantification. Several studies have applied Bayesian techniques for calibration of subsurface flow models. Markov Chain Monte Carlo (MCMC) was used by Oliver et al. [5] to sample the posterior probability density function of the log-permeability maps for single phase flow problems. Efendiev et al. [6] presented an efficient two-stage Markov Chain Monte Carlo method for dynamic data integration following the original work of [17]. Along that line, a two-stage MCMC method was proposed for conditioning the permeability map to dynamic production data of oil reservoirs [18]. For each proposed permeability map, an approximate model using a coarse grid was used to estimate if there is an improvement to the data match over the current state. If an improvement is predicted, the fine-scale model is run and a modified Metropolis–Hastings (MH) acceptance criterion is tested. The two-stage MCMC algorithm has shown higher acceptance rates than standard MCMC methods [18].

Another class of efficient methods for uncertainty quantification (UQ) is based on generalized polynomial chaos (gPC) [19]. In gPC methods, an orthogonal polynomial approximation is constructed to represent the solution of the stochastic system over the entire random space. In the cases of smooth solution, polynomial chaos expansion methods exhibit fast convergence rates [20,21]. Generalized polynomial chaos methods can be divided into intrusive Galerkin methods where extensive modifications of the deterministic code are required and non-intrusive approaches where the existing deterministic codes are used as a blackbox. The construction of polynomial approximation of the stochastic system response within the response surface methods can be formulated as an interpolation problem where the polynomial approximation matches the solution at the collocation nodes or as a regression problem where an average error norm is minimized [21]. Marzouk and Xiu [22] introduced a stochastic collocation approach to Bayesian inference where a polynomial approximation of the forward solution is constructed over the prior distribution support. This approximation was used as a surrogate posterior probability density function (PDF) that can be efficiently evaluated within a MCMC algorithm. Ma and Zabaras [23] presented an efficient Bayesian inference approach to inverse problems utilizing an adaptive sparse grid collocation method and MCMC method. Zeng et al. [24] applied a sparse grid based Bayesian method for contaminant source identification with promising results. More recently, [25] investigated the use of two-stage Markov Chain Monte Carlo simulation based on approximate gPC response surface for high-dimensional groundwater models. However, the presented test cases in [25] were limited to steady state models.

In practice many target functions are sparse and contain only a small number of significant terms in their frequency domain. Efficient methods for reconstructing sparse functions (signals) were developed in the field of compressed sensing (CS) [26,27]. Standard reconstruction methods rely on defining a set of basis functions and then one tries to find the optimal set of weights to reconstruct the measurements. This reconstruction problem is an ill-posed problem and regularization techniques (e.g., Tikhonov regularization) that constrain the $\ell_2$-norm of the solution are commonly applied. The quality of the solution depends on the class of basis functions that are used to parameterize the search space. In sparse calibration methods, a large collection of basis functions are included in a dictionary and the solution process consists of two steps: Picking from the dictionary the best set of basis functions for accurate reconstruction of the unknown field, and then finding the associated weights of the selected basises. Doostan and Owhadi [28] combined the idea of sparse calibration and gPC to approximate the solution of PDEs with stochastic coefficients. This method was shown to converge in probability (with probabilistic error bounds) as a consequence of sparsity and a concentration of measure phenomenon on the empirical correlation between samples [28]. Blatman and Sudret [29] developed an adaptive non-intrusive method that builds a sparse PC expansion using the least angle regression algorithm for automatically detecting the significant coefficients of the PC expansion. Along the same line, [30] studied a high-dimensional stochastic collocation method where the polynomial coefficients are obtained by solving an $\ell_1$ constrained minimization problem.

In this paper, the unknown model parameters (i.e., permeability field) are calibrated in two stages, static data integration and dynamic data integration. In the static data integration step, the permeability values at the wells are interpolated stochastically using Gaussian process regression (GPR) assuming a Matérn covariance function. Following that, the mean field and covariance matrix, obtained by the GPR, are used to define a reduced search space using Karhunen–Loève (KL) expansion [31–33]. The use of GPR enables efficient estimation of the optimal correlation lengths. This is to be contrasted with methods that utilize a pre-selected correlation lengths [34,6,35]. For the dynamic data integration step, the nested sampling algorithm [36–39] is utilized for Bayesian inference on the reduced search space. A crucial step within the nested sampling algorithm relies on MCMC algorithm to iteratively refocus the samples to high-likelihood regions. For the MCMC algorithm, a two-stage MCMC algorithm, following [17,6], is utilized where a surrogate based on gPC is used to filter out rejected samples and to reduce the total number of forward runs. The construction of the gPC surrogate model is formulated as an $\ell_1$ minimization problem. The combined use of nested sampling, stochastic response surface and sparsity promoting $\ell_1$ regularization resulted in an efficient sampling algorithm for inverse UQ. One notable feature of the proposed algorithm is the adaptive enrichment of the response surface based on the samples obtained by the nested sampling algorithm, which resulted in an increase in the response surface accuracy in the high-likelihood regions. This is to be contrasted with the work of [22,23], where a global response surface is constructed to a certain accuracy and then a Bayesian inference algorithm is run on the response surface.

The organization of this paper is as follows. Section 2 introduces Bayesian inference using the nested sampling algorithm. Section 3 presents the details of non-intrusive stochastic response surface method with sparsity promoting regularization. Section 4 describes the algorithmic details for the proposed nested sampling algorithm with two-stage local MCMC sampling. Section 5 provides an abstract formulation of the multiphase subsurface flow problem and a simple description of
Gaussian process regression [40], as it is used to integrate prior knowledge about the unknown fields. Following that, a brief discussion of a search space dimension reduction method based on Karhunen–Loève expansion is presented. In Section 6, the proposed algorithm is applied for calibration and model selection of several subsurface flow models. Finally, the conclusions of our work are drawn in Section 7.

2. Bayesian nested sampling

Nested sampling [36,37] is an efficient sampling algorithm for general Bayesian inference. In nested sampling, a number of samples (called the active set) are drawn from the prior PDF. The likelihood of each of these samples is evaluated based on the deviations from the observed data. Following that, a constrained sampling step is performed to add more samples to high-likelihood regions. This constrained sampling is repeated several times. The final output of the nested sampling algorithm is a set of samples with associated weights that can be resampled to obtain the posterior distribution. The re-focusing of samples towards high-likelihood regions (i.e., close to observations) by the constrained sampling step reduces the effects of the curse of dimensionality. The next subsection provides the details of the nested sampling algorithm.

### 2.1. Nested sampling algorithm

NS provides a discrete representation of the posterior distribution in the form of a set of weighted samples. The following is a description of the NS algorithm with a focus on efficient evaluation of the Bayesian evidence. Subsequently, we will present a simple method for obtaining a discrete representation of the posterior distribution based on the NS algorithm. Starting from Bayes’ theorem

\[
p(x|D) = \frac{p(D|x)p(x)}{p(D)} = \frac{\text{likelihood} \times \text{prior}}{\text{normalizing constant}},
\]

where \(x\) is the vector of unknown parameters and \(D\) is the observable data. The normalizing constant (marginal likelihood or Bayesian evidence) is defined as

\[
Z = p(D) = \int p(D|x)p(x)dx.
\]

The main idea of nested sampling is to convert the high-dimensional integral needed for evaluating the evidence into a one dimensional integral that is easier to evaluate numerically. A change of variable is performed by defining a cumulative prior mass \(X\) as

\[
X(\lambda) = \int_{L(x) > \lambda} p(x)dx,
\]

where \(L(x) = p(D|x)\) is the likelihood and \(\lambda\) is a real number from the interval \([0, 1]\). This prior mass will have a value 1 if \(\lambda\) is set to 0 and will decrease to 0 as \(\lambda\) approaches 1. The integration of the Bayesian evidence \(Z\) given by Eq. (2) can then be transformed into

\[
Z = p(D) = \int_{0}^{1} L(X)dX,
\]

with \(X\) as the fraction of the total prior mass such that \(dX = p(x)dx\) and \(L(X)\) is assigned the likelihood value \(L(x)\). Thus, the problem of calculating the normalizing constant becomes a one dimensional integral where the integrand is positive and decreasing. If a set of samples \(x^1, x^2, \ldots, x^m\) is generated such that the likelihood of these samples is an ordered sequence of increasing values \(L^1 < L^2 < \cdots < L^{m-1} < L^m\), then the corresponding prior masses \(X^1, X^2, \ldots, X^m\) will be a sequence of decreasing values,

\[
0 < X^m < \cdots < X^2 < X^1 < 1.
\]

The one dimensional integrand \(Z\) can be estimated using the trapezoidal rule as

\[
Z = \sum_{j=1}^{m} Z^j, \quad Z^j = \frac{1}{2} (X^{j-1} - X^j).
\]

This equation can be easily evaluated if the difference of the prior masses \((X^{j+1} - X^j)\) can be estimated. To obtain \(X^j\), \(N\) samples are drawn from the prior such that all the samples satisfy the constraint \(L(x^i) > L^j\) for all \(i \in \{1, 2, \ldots, N\}\. If the sample with the lowest likelihood is selected and \(L^m\) is set equal to \(L(x_{\text{worst}})\), the prior mass shrinkage factor \(t_j = X^j/X^{j-1}\), can be estimated statistically from the distribution \(p(t_j) = N t_j^{N-1}\), where \(N\) is the number of samples and \(t_j \in (0, 1)\. This is the probability distribution for the largest of \(N\) samples drawn uniformly from the interval \([0, 1]\) (largest in terms of \(X\) value and smallest in terms of the likelihood value). This fact is based on order statistics [41] where the \(N\)-th ordered point drawn from a
distribution has its cumulative quantity distributed according to the distribution Beta \( (N, 1) \) [37,42]. This distribution has the following properties: \( E(\log t) = -1/N \) and \( \sigma(\log t) = 1/N \) for the mean and standard deviation, respectively. Since each \( t_j \) is independent, after \( j \) iterations the prior volume will shrink down such that \( \log X^j \approx -(j + \sqrt{j})/N \). As an approximation, \( X^j \) can be taken as a deterministic value of \( \exp(-j/N) \) [37]. Based on this description, an iterative algorithm starting from the iteration index \( j = 1 \) and \( X^0 = 1 \) can be formulated as detailed in Algorithm 1.

### Algorithm 1: Nested Sampling Algorithm

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Initialize Nested Sampling:</strong></td>
</tr>
<tr>
<td></td>
<td>Set ( S_{\text{prior}} = {x_1, x_2, \ldots, x_N} ),</td>
</tr>
<tr>
<td></td>
<td>Set ( Z^0 = 0, X^0 = 1, S_{\text{posterior}} = {} ),</td>
</tr>
<tr>
<td>2</td>
<td><strong>Nested Sampling Iteration:</strong></td>
</tr>
<tr>
<td></td>
<td>for ( j = 1, \ldots, M ) do</td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td><strong>Resampling Step:</strong></td>
</tr>
<tr>
<td></td>
<td>Set ( Z = Z^M + (X^M/N)(L(x^1) + \cdots + L(x^N)) )</td>
</tr>
<tr>
<td></td>
<td>Set ( S_{\text{posterior}} \leftarrow S_{\text{posterior}} \cup {x^i, (X^M/N)L(x^i)} ) ( \forall x^i \in S_{\text{active}} )</td>
</tr>
<tr>
<td>12</td>
<td>Resample ( N ) samples from ( S_{\text{posterior}} ) to obtain equi-probable samples from the posterior probability distribution.</td>
</tr>
</tbody>
</table>

In the detailed pseudocode, we did not draw \( N \) new samples at each iteration. Instead, the samples were reused between the iterations by replacing the sample having the lowest likelihood (i.e., highest \( X \)) with a new sample while keeping the other \( N - 1 \) samples in the active set. Effectively, the NS algorithm climbs up the likelihood function, by shrinking the parameter space by a factor of \( t_j \) every time a sample is replaced. Ultimately, it samples only the part of the likelihood close to the maximum.

The main difficulty of NS algorithm is in the constrained sampling step where a new sample \( x_{\text{new}} \) is proposed satisfying the restriction \( L'(x_{\text{new}}) > L' \). This step becomes harder after applying several nested sampling iterations as \( L' \) reaches a high value and the parameter space satisfying the lower bound constraint shrinks to a smaller region. One solution for this problem is to use Markov Chain Monte-Carlo (MCMC) algorithm to explore the search space and find new samples conforming to the required constraint. Following [37], a more efficient solution is to start a short MCMC chain by randomly selecting one sample from the active set’s \( N \) particles, excluding the sample with the lowest likelihood. The starting state of this MCMC chain will satisfy the desired condition of the prior and the likelihood constraint. However, the new sample has to be independent from previous samples. Thus, the objective is to move away from the starting state without violating the likelihood constraint but far enough to lose memory of the starting state. Assuming the MCMC chain starts from the state \( x^i \) where \( x^i \in S_{\text{active}}, x^i \neq x_{\text{worst}} \), we utilize a simple random walk proposal distribution \( q(\xi | x^i) \) for the MCMC chain of the form

\[
\xi = x^i + \delta w, \quad w \sim N(0, C),
\]
where \( \delta \) is a scaling factor controlling the update step-size (0.1 in our numerical testing), \( \mathbf{C} \) is the sample covariance matrix for all the samples in the active set and \( \mathcal{N} \) denotes the normal distribution. The details of the constrained local MCMC sampling is presented in Algorithm 2.

### Algorithm 2: Constrained local sampling via Metropolis–Hastings algorithm

**Input:** Likelihood hard constraint \( \mathcal{L}(x_{\text{corr}}) \), Starting sample of the MCMC chain \( x^i \) and number of local MCMC iterations \( l \)

**Output:** \( x_{\text{corr}} \)

1. Set \( \theta^0 = x^i \)
2. for \( r = 1, \ldots, l \) do
   3. \( \theta^r \leftarrow \theta^{r-1} \)
   4. Propose \( \xi \sim q(\cdot \mid \theta^r) \) using Eq. (7)
   5. Evaluate \( \mathcal{L}(\xi) \)
   6. if \( \mathcal{L}(\xi) > \mathcal{L}(x_{\text{corr}}) \) then
   7. Compute \( \alpha = \min\left\{ 1, \frac{\mathcal{L}(\xi)}{\mathcal{L}(\theta^r)} \right\} \)
   8. Generate \( u \sim \mathcal{U}(0, 1) \) (i.e., uniform distribution)
   9. if \( u \leq \alpha \) then
   10. \( \theta^r \leftarrow \xi \)
11. end
12. end
13. end
14. Set \( x_{\text{corr}} \leftarrow \theta^l \).

So far, our description of the nested sampling was focused on evaluating the evidence \( Z \). However, the set of samples discarded during the nested sampling iteration as well as the samples included in the active set after terminating the algorithm can be used for estimating the posterior distribution. These samples contribute to the one-dimensional integral in Eq. (6). In other words, the existing sequence of discarded samples \( x^1, x^2, \ldots \) already provides a discrete representation of the posterior PDF with an importance weight given by \( \mathcal{L}(X^1 - X^i) / 2 \). To obtain equally-weighted posterior samples, we need to resample (with replacement) a subset of these samples with normalized weights obtained from the area under the one-dimensional integral [37].

For a complete theoretical analysis of the NS algorithm, interested readers are referred to the original papers by Skilling [36,37] and further theoretical investigations by Evans [43] and Chopin and Robert [42]. A detailed comparison of the NS to other Bayesian techniques has been performed by Murray [44]. In the next section, we present an effective algorithm for accelerating the constrained sampling algorithm. The proposed algorithm is a two-stage MCMC algorithm [17,6] where a proxy model is built using polynomial chaos to filter out rejected samples.

### 3. Sparse polynomial chaos surrogates

The main objective of this work is to improve the performance of the NS algorithm by re-using the samples at intermediate iterations of the NS algorithm to fit a surrogate model that can accelerate the constrained local sampling step detailed in Algorithm 2. In this section, we present a brief introduction to generalized polynomial chaos (gPC). Following that we present the stochastic response surface, a non-intrusive technique that treats the numerical simulator as a blackbox where the response of the system is represented using generalized polynomial chaos (gPC). In the last subsection, we present the Least Angle Regression (LARS) method for estimating the parameters of the polynomial chaos surrogate using sparse promoting regularization.

#### 3.1. Generalized polynomial chaos

Generalized polynomial chaos (gPC) expansion seeks to approximate a random process by orthogonal polynomials of random variables. The orthonormal polynomial bases are commonly selected from the Wiener–Askey polynomial chaos [19] as it contains various classes of orthogonal polynomials that can be used to represent different random variables. The chaos expansion for a response \( f \) takes the form
where $\xi_i$ represents each random variable, $N$ is the total number of random variables, $c$ denotes the different polynomial constants and $\Gamma$ is a multivariate polynomial (i.e., Hermite or Legendre) of order $n$.

The multivariate orthogonal polynomial $\Gamma$ can be represented by a product of one-dimensional polynomials involving a multi-index [20,21]. Let $\lambda$ denote the multi-index, $\lambda = \{\lambda_1, \ldots, \lambda_n\}$ and let $\lambda(p)$ denote the following set of multi-indices [21]

$$\lambda(p) = \left\{ \gamma : \sum_{i=1}^{N} \gamma_i = p \right\}.$$  

The $p$-th order polynomial chaos is then defined as

$$\Gamma_p = \left\{ \bigcup_{\gamma \in \lambda(p)} \prod_{i=1}^{N} \phi_{\gamma_i} (\xi_i) \right\},$$  

where $\phi_i(\xi)$ is a one–dimensional Hermite polynomial basis function and $i$ is the order of this basis function. Traditionally, the polynomial chaos expansion includes a complete basis of polynomials up to a fixed total–order specification and the expansion in Eq. (8) after truncation can be written in a compact form as [21]

$$f(\xi_1, \xi_2, \ldots, \xi_N) = \sum_{i=0}^{M} c_i \psi_i(\xi_1, \xi_2, \ldots, \xi_N),$$  

where $\psi_i$ represents the different $\Gamma$ functions. The total number of terms $N_t = M + 1$ corresponding to a total order truncation for an expansion of order $P$ involving $N$ random variables is given by Le Maitre and Knio [21] and Hosder [45]

$$N_t = M + 1 = \binom{P + N}{N} = \frac{(P+N)!}{P!N!}.$$  

### 3.2. Stochastic response surface

If the response of the stochastic system is evaluated for a set of points in the stochastic space, a linear regression approach (also known as point collocation or stochastic response surfaces [45]) can be used to evaluate the expansion coefficients in Eq.(11) using a single linear least squares solution of the form

$$f = \Psi c,$$

where $c \in \mathbb{R}^{N_t}$ is the vector of the gPC coefficients to be determined, $f \in \mathbb{R}^K$ is the vector of samples (system response), $\Psi \in \mathbb{R}^{K \times N_t}$ is the measurement matrix where each column contains samples of the $j$th element of the gPC basis and $K$ is the number of samples (system response). Eq. (13), can be solved using least squares if the number of samples $K$ is larger than the number of unknown constants $N_t$. However, for systems with high number of stochastic dimensions $N$, the number of polynomial constants becomes very large. Similar to the work of [28], we are interested in the case where the number of samples $K$ is much smaller than the unknown gPC coefficients $N_t$, i.e., $K \ll N_t$. In those cases, the linear system in Eq. (13) is ill-posed and admits infinitely many solutions. In the current manuscript, we utilize sparsity promoting regularization where only a small fraction of the coefficients $c_i$ are assumed significant.

In practice many target functions appear to be rough in time or space, however these functions can be represented using a small number of notable terms in the frequency domain. This observation is extensively utilized in the field of compressed sensing [26,27,46]. The sparse recovery (reconstruction) problem can be formulated as an optimization problem with sparsity promoting term

$$(P_0) : \min_c \| c \|_0 \quad \text{subject to} \quad f = \Psi c,$$

where $\| c \|_0$ is a semi-norm counting the number of nonzero terms of $c$. A relaxed version of this optimization problem is

$$(P_{0,\delta}) : \min_c \| c \|_0 \quad \text{subject to} \quad \| f - \Psi c \|_2 \leq \delta,$$

where $\delta$ is an error-tolerance. Different algorithms have been developed to solve $(P_0)$ or $(P_{0,\delta})$ in the active field of compressed sensing [26,27,46]. Notably, the orthogonal matching pursuit (OMP) [47,48] was used by Doostan and Owhadi [28] for reconstruction of sparse polynomial chaos and by Elsheikh et al. [10] for sparse calibration of subsurface flow models.

A different formulation for the sparsity promoting regularization was developed in the statistics literature. The problem is formulated as

$$(L_1,\beta) : \min_{c \in \mathbb{R}^{N_t}} \| f - \Psi c \|_2^2 + \beta \| c \|_1,$$

where $\beta$ is an error-tolerance. Different algorithms have been developed to solve $(L_1,\beta)$ in the active field of compressed sensing [26,27,46]. Notably, the orthogonal matching pursuit (OMP) [47,48] was used by Doostan and Owhadi [28] for reconstruction of sparse polynomial chaos and by Elsheikh et al. [10] for sparse calibration of subsurface flow models.
where \( |c|_1 \) is the \( \ell_1 \) norm and equals \( \sum_{j=1}^{M} |c_j| \) and \( \beta \) is a regularization parameter. The \( \ell_1 \) penalized linear regression defined by Eq. (16), also known as the least absolute shrinkage and selection operator (LASSO) problem [49,50], was shown to be equivalent to sparsity promoting regularization [46]. In the current manuscript, we utilize the Least Angle Regression (LARS) algorithm [51] to solve the \( \ell_1 \) penalized linear regression problem. The LARS algorithm is similar to forward stepwise regression, but instead of including variables at each step, the estimated parameters are increased in a direction equiangular to all variables included in the solution support. In the following subsection we present a brief description of the LARS algorithm.

3.3. Least Angle Regression

Least Angle Regression (LARS) [51] is related to classic model-selection method known as forward stepwise regression algorithm (FSRA). In FSRA, the algorithm starts with all coefficients \( c_j \) set to zero. The column \( \Psi_j \) from the measurement matrix \( \Psi \) with the largest absolute correlation with the response \( f \) is selected for simple linear regression. Following that, the residual vector orthogonal to \( \Psi, c \), is considered to be the response, and the iterations continue. The LARS algorithm initially proceeds in the same way as FSRA. However, LARS is less greedy than FSRA as the largest possible step is taken in the direction of the most correlated column \( \Psi_i \) with the residual vector until some other column \( \Psi_j \) has as much correlation with the current residual. At this point instead of continuing along the second column \( \Psi_i \), the LARS algorithm proceeds in a direction equiangular between the two columns \( \Psi_i, \Psi_j \) until a third column \( \Psi_k \) earns its way into the “most correlated” set. LARS then proceeds equiangularly between \( \Psi_i, \Psi_j, \Psi_k \), that is, along the “least angle direction”, until a fourth variable enters, and so on. Algorithm 3 presents an informal description of the LARS algorithm adopted from [52]. The details of the mathematical implementation of the algorithm can be found in [51,52].

Algorithm 3: Least Angle Regression (LARS) algorithm

1. **Input:** Measurement matrix \( \Psi; \) response vector \( f \);
2. **Initialize:** Initial solution \( c = 0 \), Initial residual \( r = f \);
3. – Find the column \( \Psi_i \) most correlated with \( r \).
4. – Move \( c_i \) from 0 towards its least-squares coefficient \( (\Psi_i, r) \), until some other column \( \Psi_j \) has as much correlation with the current residual as does the column \( \Psi_i \).
5. – Move \( (c_i, c_j) \) in the direction defined by their joint least squares coefficient of the current residual on \( (\Psi_i, \Psi_j) \), until some other column \( \Psi_k \) has as much correlation with the current residual.
6. – Continue until all \( m \) columns of \( \Psi \) have entered the solution.
7. – After \( N_i \) steps, the full least-squares solution is obtained.

The LARS produces a full piecewise linear solution path (solution with incremental sparsity levels), which is useful when combined with an information based criterion to select the optimal level of sparsity and avoid overfitting. This is a major advantage over other sparse recovery algorithms that rely on cross-validation to tune the \( \ell_1 \) regularization constants requiring multiple runs [53]. In the current study, we utilized the LARS algorithm implemented in [54], and selected the solution corresponding to the minimum Akaike information criterion (AIC) condition to find the right balance between data mismatch and solution sparsity. Akaike [55] introduced the AIC as a method for discriminating between statistical models. AIC was proposed as an asymptotically unbiased estimator of the expected Kullback–Leibler information. In the current implementation, the number of degrees of freedom corresponds to the number of non-zero components of the solution vector \( c \). The AIC corresponding to a solution with \( k \) non-zero terms in the solution is defined by:

\[
\text{AIC}(k) = r^\top r + 2\sigma^2_k k.
\]

where \( \sigma^2_k = (f - \Psi c)\top (f - \Psi c)/K \) is the number of data points (samples) and \( \tilde{c} \) is a regularized solution using Moore-Penrose pseudoinverse evaluated as \( \tilde{c} = \Psi^\top f \). The residual \( r \) is calculated at each solution along the path as \( r = f - \Psi c \).

The presented stochastic response surface method and the sparsity promoting regularization using the LARS algorithm will be combined to build a response surface representing the input/output relation of the forward numerical simulator. This response surface will be utilized in a two-stage MCMC for the local constrained sampling at each iteration of the nested sampling algorithm.

4. Surrogate based two-stage MCMC Sampling

The proposed algorithm is based on the NS algorithm in which the constrained sampling step (step 12 in Algorithm 1) will be performed using a two-stage MCMC algorithm instead of the full MCMC constrained sampling detailed in Algorithm 2. A polynomial chaos surrogate is built using a database of all the samples corresponding to completed forward runs including the active set samples, discarded samples during the NS iterations and rejected/accepted proposed samples during
the local MCMC runs. The constructed surrogate model will act as a filter in a two-stage MCMC sampling based on the work of [6,56].

In standard MCMC algorithms, the major computational cost is attributed to evaluating the likelihood of a proposed sample $\xi$, which involves solving the coupled nonlinear system of PDEs (forward run). However, many of the proposed samples are rejected and the corresponding computational effort is wasted. Here, we propose to adapt the proposal distribution $q(\xi|\theta)$ using a surrogate model, where $\theta$ is the current sample. If the proposal is accepted based on the surrogate response, then a full forward simulation run is conducted and the proposal $\xi$ will be tested again using a modified acceptance/rejection test condition. Otherwise, the proposal will be rejected based on the approximated response estimated from the surrogate model. We utilized the stochastic response surface method with a sparsity promoting regularization as detailed in the previous section for constructing the surrogate model. In some sense, the acceptance/rejection test based on the surrogate model response is a filter that modifies the proposal distribution.

Algorithm 4 shows the details of the two-stage MCMC algorithm where a surrogate model is used to evaluate the approximate likelihood $\tilde{L}$. The acceptance of the proposed sample $\xi$ is performed in two stages. The first level of acceptance relies on the approximate likelihood using the surrogate model. The second level of acceptance relies on running the full simulator and then applying a modified acceptance rejection rule [56]

$$\alpha = \min \left\{ 1, \frac{L(\xi) \tilde{L}(\theta)}{\tilde{L}(\theta) \tilde{L}(\xi)} \right\}.$$  

(18)

It is possible that the test based on the approximate likelihood using the surrogate model may reject an individual sample which would otherwise have a (small) probability to be accepted if a full MCMC algorithm is run. However, it was shown in [56] that this type of preconditioned MCMC algorithm converges under some mild assumptions to the true posterior distribution. The approximate likelihood relies on the measurement noise to scale the misfit error function and similar to [6], we utilize the same measurement error for the approximate likelihood and for the full forward model.

**Algorithm 4: Constrained sampling via a two level MCMC algorithm**

**Input:** Likelihood hard constraint $L(\mathbf{x}_{\text{true}})$, Starting sample of the MCMC chain $x'$

and number of local MCMC iterations $l$, surrogate model to evaluate an

approximate likelihood $\tilde{L}$.

**Output:** $x_{\text{true}}$

1. Set $\theta^0 = x'$

2. for $r = 1, \ldots, l$

3. $\theta^r \leftarrow \theta^{r-1}$

4. Propose $\xi \leftarrow q(.|\theta^r)$ using Eq. (7)

5. Evaluate $\tilde{L}(\xi)$

6. if $\tilde{L}(\xi) > L(\mathbf{x}_{\text{true}})$ then

7. Compute $\bar{\alpha} = \min \left\{ 1, \frac{\tilde{L}(\xi)}{\tilde{L}(\theta^r)} \right\}$

8. Generate $u \sim \mathcal{U}(0,1)$ (i.e., uniform distribution)

9. if $u \leq \bar{\alpha}$ then

10. Evaluate $L(\xi)$

11. if $L(\xi) > L(\mathbf{x}_{\text{true}})$ then

12. Compute $\alpha = \min \left\{ 1, \frac{L(\xi)\tilde{L}(\theta^r)}{\tilde{L}(\theta^r)\tilde{L}(\xi)} \right\}$

13. if $u \leq \alpha$ then

14. $\theta^r \leftarrow \xi$

15. end

16. end

17. end

18. end

19. end

20. Set $x_{\text{true}} \leftarrow \theta^r$. 
5. Problem formulation and parametrization

A two-phase immiscible flow in a heterogeneous porous subsurface region is considered. For clarity of exposition, gravity and capillary effects are neglected. However, the proposed model calibration algorithm is independent of the selected physical mechanisms. The two phases will be referred to as water with the subscript \( w \) for the aqueous phase and oil with the subscript \( o \) for the non-aqueous phase. This subsurface flow problem is described by the mass conservation equation and Darcy’s law

\[
\nabla \cdot \mathbf{v}_t = q, \quad \mathbf{v}_t = -\kappa \nabla p \quad \text{over} \quad \Omega,
\]

where \( \mathbf{v}_t \) is the total Darcy velocity of the engaging fluids, \( q = Q_o/\rho_o + Q_w/\rho_w \) is the normalized source or sink term, \( \kappa \) is the absolute permeability tensor, \( S_w \) is the water saturation, \( \lambda_t(S_w) = \lambda_w(S_w) + \lambda_o(S_w) \) is the total mobility, \( p = p_o = p_w \) is the pressure and \( \rho_w, \rho_o \) are the water and oil fluid densities, respectively. These equations can be combined to produce the pressure equation

\[
-\nabla \cdot (\kappa \lambda_t(S_w) \nabla p) = q.
\]

The pore space is assumed to be filled with fluids and thus the sum of the fluid saturations is one (i.e., \( S_n + S_w = 1 \)). Then, only the water saturation equation is solved

\[
\phi \frac{\partial S_w}{\partial t} + \nabla \cdot (f(S_w) \mathbf{v}_t) = \frac{Q_w}{P_w},
\]

where \( \phi \) is the porosity, \( f(S_w) = \lambda_w/\lambda_t \) is the fractional flow function. The relative mobilities are modeled using polynomial equations of the form

\[
\lambda_w(S_w) = \frac{(S_{nw})^2}{\mu_w}, \quad \lambda_o(S_w) = \frac{(1 - S_{nw})^2}{\mu_o}, \quad S_{nw} = \frac{S_w - S_{nc}}{1 - S_w - S_{nc}},
\]

where \( S_{nc} \) is the connate or irreducible water saturation, \( S_{or} \) is the irreducible oil saturation and \( \mu_w, \mu_o \) are the water and oil fluid viscosities, respectively. The pressure Eq. (20) is discretized using the standard two-point flux approximation (TPFA) method and the saturation Eq. (21) is discretized using a finite-volume scheme in space and backward Euler method in time and solved by a standard Newton iteration [57]. For simplicity, we limit the parameter estimation to the subsurface permeability field

The objective of the subsurface flow model calibration is to estimate the permeability field which is assumed to be log-normal random variable as it is usually heterogeneous and shows a high range of variability. In the following subsections, a brief review of Gaussian process regression (GPR) is presented as it is used for conditioning the stochastic permeability field to the values at the wells locations (static data integration). Following that, the details of search space dimension reduction using Karhunen–Loève expansion is presented. Both methods are used for parameterizing the inverse problem.

5.1. Gaussian process regression

We utilize Gaussian process regression to perform stochastic interpolation of static data collected at the wells. A detailed description of this technique can be found in [40,7,15]. Here, we highlight some of the important details specific to our numerical test cases. In the current study, the Matérn covariance function [40] is used

\[
C_{\text{Matérn}}(r) = \frac{2^{1-v}}{\Gamma(v)} \left( \sqrt{2\Gamma(v)} \right)^v J_v \left( \sqrt{2\Gamma(v)} \right),
\]

where \( \Gamma \) denotes the gamma function, \( J_v \) is the modified Bessel function of the second kind of order \( v > 0, r = ||x - x'|| \) is the distance between two points and \( l \) is the correlation length. The set of parameters \( v, l \) and the global noise variance \( \sigma_n \) are known as the hyperparameters of the Gaussian process \( \mathcal{P} = \langle v, l, \sigma_n \rangle \). Finding the optimal set of hyperparameters is a model selection problem. However, for the case of GPR with Gaussian measurement noise the integrals over the parameter space are analytically tractable. Thus, the optimal hyperparameters can be evaluated efficiently by maximizing the logarithm of the marginal likelihood (see algorithm 5.1 in [40]). The estimated optimal set of hyperparameters is called the Maximum Likelihood type II (ML-II) estimates. These parameters are found by solving a non-convex optimization problem using standard methods for unconstrained optimization (e.g., conjugate gradient with random restart [58]).

5.2. Parameterization using Karhunen–Loève expansion

The results of GPR is a real-valued random field \( \kappa \) with mean \( \mu(x) \) and a covariance function \( C(x_1, x_2) \). The Karhunen–Loève (KL) expansion [31–33], is a method for expanding Gaussian fields as a linear combination of a set of orthonormal deterministic basis functions with the linear combination weights defined as random variables. The logarithm of the permeability field can be parameterized using a limited number \( n \) of eigenvectors as
\[ \mathcal{K}(x, \xi) = \mu(x) + \sum_{k=1}^{n} \sqrt{\lambda_k} \xi_k \psi_k(x), \]

(24)

where \( \xi_k \) is a set of random variables, \( \lambda_k \) is a set of real constants and \( \psi_k(x) \) are an orthonormal set of deterministic functions. The functions \( \lambda_k, \psi_k \) are obtained from the spectral decomposition of the covariance function \( \mathcal{C} \)

\[ \mathcal{C}(x_1, x_2) = \sum_{k=1}^{\infty} \lambda_k \psi_k(x_1) \psi_k(x_2), \]

(25)

where \( \lambda_k > 0 \) are the eigenvalues, \( \psi_k \) are the corresponding eigenvectors. Different realizations can be generated for different values of \( \xi_k \). In model calibration, the values of \( \xi_k \) are estimated such that the measured data match the simulation results [7,15].

6. Numerical evaluation

The proposed algorithm is evaluated on two test problems modeling water flooding in a subsurface reservoir under different injection/production patterns. For both test cases, the discretized model uses a 2D regular grid of 41 \times 41 blocks in the \( x \) and \( y \) directions, respectively. The size of each grid block is 10 meters in each direction and a unit thickness in the \( z \) direction. The porosity is assumed to be constant in all grid blocks and equals 0.2. The water viscosity \( \mu_w \) is set to 0.3 cp and the oil viscosity \( \mu_o \) is set to 0.9 cp. The irreducible water saturation and irreducible oil saturation are set as \( S_{or} = S_{wc} = 0.2 \). In the figures, injection wells are shown as black dots and the production wells as white dots.

Fig. 1(a) shows the logarithm of the reference permeability field (permeability units is Darcy = 9.869233 \times 10^{-13} m^2) and Fig. 1(b) shows the location of the sampled 9 points for static data integration using Gaussian process regression. GPR utilizes a Matérn covariance function with an order parameter \( \nu = 3 \). The correlation lengths were optimized using the ML-II estimator and Fig. 2 shows the mean log-permeability field obtained from the GPR along with the \( \pm 2 \) standard deviations.

Fig. 1. Permeability fields for the test problems (blue dots for static data points) (a) reference log-permeability field for test problem 1 and 2 (in Darcy) and (b) mean log-permeability field obtained by Gaussian process regression. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 2. GPR results showing the reference log-permeability field in red, mean regression field in blue and the two standard deviation surfaces in gray. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
bounds where the reference field is plotted in red and the interpolated mean value is shown in blue color. Using the covariance matrix obtained by GPR, KL expansion of the field is applied. For each test case, different NS runs are performed while retaining different number of KL terms. This can be considered as using different prior models with different levels of complexity corresponding to the number of unknown KL weights. The Bayesian evidence is then estimated for each model using the NS algorithm and the results are shown for the prior model corresponding to the maximum Bayesian evidence as the Bayesian evidence quantitatively embodies Occam’s razor [59] where complex models are automatically penalized under Bayes’ rule as explained by MacKay [60].

For the parameter estimation step, the water cut curves at the production wells are used to define the misfit function and guide the inverse problem solution. Each water cut curve was sampled at 50 points and these samples were used for calculating the likelihood function. For many of the presented results, an average of 10 runs is presented to account for the stochastic nature of the sampling methods.

6.1. Water flooding test case 1

The reference field for this problem is shown in Fig. 2 and the location of the injection/production wells are shown in Fig. 3(a) as black and white dots, respectively. Fig. 4 shows the prior and posterior water cut curves obtained by the NS algorithm for a model with 2 KL basis functions as it is the model parameterization corresponding to the maximum Bayesian evidence as we will show later in Fig. 7. The results are shown in terms of dimensionless time defined by the pore volume injected (PVI). The initial active set contained 50 active samples and the NS algorithm was terminated after 100 nested sampling iterations. At each NS iteration, a short MCMC chain was run with a length of 10 steps. The scaling parameter for the proposal distribution is set to 0.1 and the measurement error covariance used in defining the likelihood is set to 0.01. The results are shown for the nested sampling algorithm with local full MCMC sampling and for the local two-stage MCMC (denoted as TS-MCMC). The surrogate model is built using Hermite polynomials with a maximum order of 4 and the polynomial chaos parameters were estimated using the LARS algorithm. It is clear that both methods managed to successfully fit the production data.

Fig. 5 shows the mean posterior log-permeability fields (in Darcy) obtained by running the nested sampling algorithm where the results corresponding to local full MCMC chains are shown in Fig. 5(a) and the results corresponding to the local two-stage MCMC sampling are shown in Fig. 5(b). These two fields are almost identical in the eyeball norm. The corresponding standard deviation of the posterior samples is shown in Fig. 6. We observe slight differences between the local full MCMC run and the local two-stage MCMC run. This is justified as the two algorithms are sampling based algorithms and the total number of forward runs is in the order of 1000 forward runs only (as will be shown later). Given the relatively small number of performed forward runs, each nested sampling run might result in a slightly different solution, however the main features of the solution are expected to be present in all the different runs.

The NS algorithm is run for a set of different models with varying number of KL terms ranging from 2 KL terms up to 6 KL terms. The Bayesian evidence (marginal likelihood) for the different models are shown in Fig. 7(a) and (b) for the local full MCMC sampling and the local two-stage MCMC sampling, respectively. For the evidence, the model with 2 KL terms show the highest evidence in both cases. This result is consistent with Occam’s razor as the models with the fewest parameters should be selected if a good match to the data can be attained. It is also observed that the estimated values of the Bayesian evidence is almost the same for the two local constrained sampling methods. However, for the model with 4 KL terms, the NS equipped with the two-stage MCMC produces a lower estimate for the Bayesian evidence. Nevertheless, if one is interested in selecting the best model, the accelerated two-stage MCMC sampling within the NS algorithm would result in selecting the model with 2 KL terms similar to the nested sampling algorithm with local full MCMC sampling.

In terms of computational cost, Fig. 8 shows a comparison of the computational cost of the full nested sampling algorithm versus the surrogate accelerated nested sampling algorithm for four different models corresponding to 2, 4, 5 and 6 KL terms.

![Fig. 3. Injection and production patterns (black dots for injection wells and white dots for production wells). (a) Test case 1 (b) Test case 2.](image-url)
Significant computational saving, in terms of the total number of forward runs, is observed for all the NS runs with the local two-stage MCMC constrained sampling ($\approx 10$ to $18\%$). This should be assessed in relation with the acceptance rates shown in Fig. 9 where the goal of the two-stage MCMC local sampling is to avoid running the simulator on rejected samples. If the acceptance rates are as high as $80\%$ then the range of possible improvement is limited. It is worth mentioning that the observed high acceptance rates of the local MCMC sampling are attributed to the adaptive proposal defined in Eq. (7), which depends on the covariance of the samples in the active set. For models with higher dimensions, the accuracy of the surrogate might be questionable and false accepted samples results in a slight increase in the computational cost. However, the number of forward runs is still lower than the NS with local full MCMC constrained sampling.

In order to provide more insight about the accuracy of the surrogate model, Fig. 9 shows the accumulated acceptance rate versus the number of NS iterations for the different models with 2, 4, 5 and 6 KL terms. Intuitively, if the surrogate model is perfect, the local two-stage MCMC should produce an acceptance ratio of 1. This is observed in Fig. 9(a) as the problem is low
Fig. 5. Inferred mean log-permeability fields at the end of the nested sampling algorithm for test case 1 for the model with 2 KL terms. (a) Full-MCMC and (b) TS-MCMC ($\ell_2$ norm of the difference = 0.6683).

Fig. 6. Standard deviations of the posterior permeability field using the nested sampling algorithm on test case 1 for the model with 2 KL terms. (a) Full-MCMC and (b) TS-MCMC ($\ell_2$ norm of the difference = 0.0529).

Fig. 7. Logarithm of the Bayesian evidence versus number of KL terms for test case 1. Results are shown after 100 NS iterations and an active set of 50. For the TS-MCMC, the Hermite polynomials order is set to 4. (a) Full-MCMC and (b) TS-MCMC.
dimensional and the surrogate model becomes accurate after few nested sampling iterations leading to an acceptance ratio slightly above 0.95. However, for high dimensional cases, the surrogate model is less accurate in the early stages of the nested sampling iterations. This results in false accepted samples in the first stage of the local two-stage MCMC algorithm. Later in the NS algorithm, we observe an increase in the acceptance ratio corresponding to a higher accuracy level of surrogate model as more samples are used in the fitting process. We conclude that the two-stage local MCMC sampling outperforms the standard (full) MCMC local sampling after a number of nested sampling iterations that depends on the dimension of the problem.

6.2. Water flooding test case 2

Similar to the previous test case, the reference field for this problem is shown in Fig. 1(a), however the location of the injection/production wells are shown in Fig. 3(b). For this test case, the NS algorithm utilizes an initial active set of 100 samples and the NS algorithm was terminated after 200 iterations. At each NS iteration a short MCMC chain is run with a length

![Fig. 8. Number of forward runs versus number of nested sampling iterations for test case 1. (a) 2 KL terms, (b) 4 KL terms, (c) 5 KL terms and (d) 6 KL terms.](image-url)
of 10 steps and the scaling parameter for the proposal distribution was set to 0.1 and the measurement error covariance was set to 0.01. For the two-stage local MCMC sampling, the surrogate model is built using Hermite polynomials with a maximum order of 4 and the polynomial chaos parameters are estimated using the LARS algorithm.

Fig. 10 shows the prior and posterior water cut curves obtained by the NS algorithm for a model with 7 KL basis functions as it is the model corresponding to the maximum Bayesian evidence as we will show later in Fig. 13. The posterior distribution of the water cut curves obtained by running the NS algorithm with a local full MCMC chains are shown in Fig. 10(b) and the results of the NS algorithm with the local two-stage MCMC sampling are shown in Fig. 10(d). A good fit of the calibration data is observed for both methods.

The mean posterior log-permeability fields (in Darcy) obtained using the NS algorithm with the two different local sampling strategies are shown in Fig. 11(a) and (b) for the local full MCMC and local two-stage MCMC, respectively. We observe that the two mean posterior fields share the same major features. However, these two fields are not identical as in the previous test case. The corresponding standard deviation of the posterior samples are shown in Fig. 12. Again, we observe some
differences between the full nested sampling algorithm and the surrogate accelerated run. We attribute this to the sampling nature of these algorithms and the limited number of forward runs (2100 for the case of NS with local full MCMC sampling).

The nested sampling algorithm is run for a set of different models with varying number of KL terms ranging from 3 KL terms up to 12 KL terms. The Bayesian evidence (marginal likelihood) for the different models are shown in Fig. 13. The results are shown for the full NS algorithm and the surrogate accelerated NS algorithm. The difference between the estimated logarithm of the Bayesian evidence from NS with full local MCMC run and NS run with the two-stage local MCMCM run is $f_0 = 1637$, $C_0 = 5001$, $C_0 = 9229$, $C_0 = 8583$, $C_0 = 5306$ for the models with $3, 5, 7, 9, 12$ KL terms, respectively. These results show that the surrogate accelerated NS algorithm slightly underestimated the Bayesian evidence (in comparison to the full local MCMC) when the dimension of the search space is increased. However, the relative values of the estimated Bayesian evidence for the different models is the same. This consistency in the estimated relative Bayesian evidence values will result in a consistent decision making as the model with 7 KL terms will be selected based on either of NS runs.

**Fig. 10.** Prior versus Posterior water cut curves at the production wells for test case 2 for the model with 7 KL terms (Reference water-cut curve in black and a different color is used for each production well water-cut curve).
**Fig. 11.** Inferred mean log-permeability fields at the end of the nested sampling algorithm for test case 2 and the model with 7 KL terms. (a) Full-MCMC and (b) TS-MCMC ($\ell_2$ norm of the difference = 3.7909).

**Fig. 12.** Standard deviations of the posterior permeability field using the nested sampling algorithm test case 2 and the model with 7 KL terms. (a) Full-MCMC and (b) TS-MCMC ($\ell_2$ norm of the difference = 0.1308).

**Fig. 13.** Logarithm of the Bayesian evidence versus number of KL terms for test case 2. Results are shown after 200 NS iterations and an active set of 100. For the TS-MCMC, the Hermite polynomials order is set to 4. (a) Full-MCMC and (b) TS-MCMC.
Fig. 14 shows the accumulated acceptance rate versus the number of NS iterations for the different models with 5, 7, 9 and 10 KL terms. For models with few KL terms, the local TS–MCMC constrained sampling clearly outperformed the full MCMC local sampling. However, as the dimension of the problem is increased, the computational advantages are realized at later stages of the NS iteration. For the model with 12 KL terms, the acceptance ratio shown in Fig. 14(d) indicate that the surrogate based NS has a lower acceptance ratio than the full local MCMC after 200 nested sampling iterations. This is attributed to the difficulties in constructing a surrogate model that is accurate for a 12 dimensional search space using only 2100 samples. Fig. 15 shows the error decay in the $\ell_2$ norm versus the total number of forward runs for different models obtained by different number of retained KL terms. The plotted error corresponds to the sample from the active set with the maximum mismatch error at each NS iteration. It is clear, that the NS algorithm with the local two-stage MCMC sampling outperforms the local full MCMC sampling in the models with low to average search space size. For problems with high-dimensional stochastic space, the local two-stage MCMC initially has little advantage over the local full MCMC sampling. However, as the number of nested sampling iterations is increased the local two-stage MCMC sampling clearly outperformed the local full MCMC in terms of error reduction.
During the evaluation process, we observed that the quality of the surrogate model controls the success of the proposed algorithm. In the initial few iterations of the nested sampling algorithm, only few samples are used to construct the response surface and that might affect the quality of the fitted response surface especially for high dimensional problems. During these early stages of the nested sampling algorithm, local sampling using the standard MCMC algorithm shows better acceptance rates than the two-stage local MCMC sampling. Later in the process, the number of samples used to construct the response surface is increased and the advantages of the two-stage local MCMC sampling becomes clear. However, for high dimensional search spaces ($d > 10$), the computational budget might limit the application of the proposed algorithm.

In this subsection, we present some computational results justifying the use of sparsity promoting regularization to build the gPC response surface for the current application. Following that, we evaluate two important features of the proposed algorithm, the effect of the active set size and the effect of the interpolation order of the surrogate model in the two-stage MCMC algorithm.

**Fig. 15.** Maximum $\ell_2$ error in the active set versus number of forward runs for test case 2. (a) 5 KL terms, (b) 7 KL terms, (c) 9 KL terms and (d) 12 KL terms.
The construction of the stochastic response surface corresponds to solving the linear system in Eq. (13), which might be ill-posed. In the current manuscript we adopted a sparsity promoting regularization to solve this system. Of course, standard regularization techniques (e.g., Tikhonov method) that constrain the $\| \cdot \|_2^2$ norm of the solution could be used. Here, we try to evaluate computationally the accuracy of response surface built based on sparsity promoting regularization using the LARS algorithm versus Tikhonov regularization with automatic selection of the regularization parameter using the L-curve method [61,62]. The quality of the response surface is assessed indirectly by the acceptance ratio of the two-stage local MCMC sampling embedded within the NS algorithm. An accurate response surface will filter out the rejected samples in the local MCMC step and will result in higher acceptance ratios.

Fig. 16. Effect of response surface construction method on the acceptance ratio for test case 1 with different PCE orders. (a) 4 KL terms, (b) 6 KL terms.

![Fig. 16.](image)

Fig. 17. Effect of the active set size on the acceptance ratio and error reduction for test case 2, with 7 KL terms and PCE order of 4. (a) Acceptance ratio versus number of forward runs, and (b) Maximum $\ell_2$ error in the active set versus number of forward runs.

The construction of the stochastic response surface corresponds to solving the linear system in Eq. (13), which might be ill-posed. In the current manuscript we adopted a sparsity promoting regularization to solve this system. Of course, standard regularization techniques (e.g., Tikhonov method) that constrain the $\| \cdot \|_2^2$ norm of the solution could be used. Here, we try to evaluate computationally the accuracy of response surface built based on sparsity promoting regularization using the LARS algorithm versus Tikhonov regularization with automatic selection of the regularization parameter using the L-curve method [61,62]. The quality of the response surface is assessed indirectly by the acceptance ratio of the two-stage local MCMC sampling embedded within the NS algorithm. An accurate response surface will filter out the rejected samples in the local MCMC step and will result in higher acceptance ratios. Fig. 16 shows the accumulated acceptance rate versus the number of NS iterations for the prior model with 4 and 6 retained KL terms and for response surfaces with different polynomial chaos expansion (PCE) orders. The response surface coefficients were obtained either by the LARS algorithm as outlined in this
manuscript (denoted as sparse regularization) or by Tikhonov regularization. The computational results show the clear advantage of using sparsity promoting regularization as it results in higher acceptance rates. These results are consistent regardless of the utilized PCE order and implies the sparsity of the response surface for the subsurface flow problem model considered in the current study.

Fig. 17(a) shows the effect of the active set size on the acceptance ratio for the second test problem with 7 KL terms and utilizing PCE with order 4. During the early stages of the algorithm, full sampling outperformed the two-stage local sampling in terms of the acceptance ratio. With the increase of the number of nested sampling iterations, the two-stage local sampling clearly outperformed the local MCMC sampling algorithm. These results are consistent with perviously presented results. With regards to the effect of the active set size on the acceptance ratio for the full MCMC local sampling, we observe that the larger active set size outperformed the smaller active set sized runs during the first few iterations. We attribute that to the increased accuracy of the sample covariance matrix in the proposal distribution defined in Eq. (7). Larger active set sizes reduces the sampling error effects and results in proposing samples within the constrained region. However, later in the algorithm the run with the smaller active set outperformed the runs with the larger active set size. This is attributed to the tendency of the NS algorithm to perform as an optimization like algorithm when a small active set size is utilized. In contrast, the runs with larger active set sizes produces good search space exploration but with lower acceptance rates. For the two-stage MCMC algorithm, initially the larger active set size outperformed the run with smaller active set size. This is attributed to the increased support (number of samples) utilized in building the surrogate model. Later in the algorithm, the run with the smaller active set outperformed the runs with the larger active set size as the NS algorithm behaves as an optimization like algorithm. This explanation is confirmed in Fig. 17(b) where the maximum misfit error in the active set is plotted against the total number of forward runs. The NS runs with smaller active sets shows better error reduction for the same number of forward runs. This is easily explained in the limit by setting the active set size to one. The NS algorithm is then a pure randomized optimization algorithm where at each NS iteration the current state is replaced with another state up-hill the likelihood surface.

We finally investigate the effect of the approximation order of the response surface on the acceptance rates. Fig. 18 shows the acceptance ratio versus the total number of forward runs for test case 2 with different PCE orders with an active set of 100 samples. (a) 5 KL terms, and (b) 7 KL terms.

7. Conclusions

Nonlinear model calibration and uncertainty quantification is a challenging problem in terms of the required computational recourses and the selection of the right model complexity. An over-parameterized model is prone to overfitting during
the calibration process. In the current manuscript, we presented a highly efficient algorithm that combines the nested sampling algorithm and a two-stage MCMC local sampling. Nested sampling can be considered as a global optimization algorithm as it utilizes an active set of samples that covers the prior distribution. This active set incrementally climbs the likelihood function to the higher likelihood regions without selecting the best member and thus avoids being trapped in a local minima. The climbing process is done iteratively by replacing the sample with lowest data fit with a better sample, one at a time. These discarded samples are used to estimate the marginal likelihood (Bayesian evidence) statistically as well as estimating the posterior distribution of the inferred parameters. Nested sampling relies on converting a high dimensional integral into a one-dimensional integral in the likelihood space for estimating the marginal likelihood. This one dimensional integral is evaluated using the trapezoidal integration method. Once the nested sampling algorithm is terminated, the active set is concentrated in high likelihood regions and the best sample in the active set provides the maximum a posteriori (MAP) estimate of the unknown parameters. For uncertainty quantification, the posterior distribution can be estimated from the set of discarded samples during the nested sampling iterations and the samples in the final active set.

The main difficulty with the nested sampling algorithm is in performing a constrained sampling step from the prior at each iteration. We proposed a two-stage local MCMC to accelerate this step. In the two-stage MCMC algorithm a surrogate model is built using stochastic response surface method and sparsity promoting regularization. This surrogate model is utilized to filter out rejected samples and to increase the acceptance ratio of the MCMC algorithm. All the forward runs performed by the nested sampling algorithm are utilized to build the surrogate and thus the accuracy of the surrogate increases with the number of nested sampling iterations. The proposed algorithm was applied to the inference of subsurface properties as well as selecting the most plausible parameterization of the model from a set of models with varying number of unknown parameters. We obtained various numerical results showing the interplay between the accuracy of the proxy model, model complexity in terms of retained KL-terms and the active set size used in the nested sampling algorithm. We observed that models with more parameters might need more iterations to benefit from the proposed algorithm. However, for many problems the Bayesian evidence shows that it is better to calibrate simpler models in order to avoid over–fitting. We also observed for moderate size problems that full sampling using MCMC outperformed the two-stage MCMC sampling in the initial iterations of the nested sampling algorithm. This suggests investigating an adaptive algorithm that starts with sampling scheme based on full local MCMC sampling and then switches to the two-stage MCMC algorithm at later iterations of the nested sampling algorithm as the surrogate model becomes more accurate.

References
