This is the submitted version of a paper published in *Computers & Geosciences*.

Citation for the original published paper (version of record):


Access to the published version may require subscription.

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-298748
Gaussian Process Emulators for Quantifying Uncertainty in CO₂ Spreading Predictions in Heterogeneous Media

Liang Tian, Richard Wilkinson, Zhibing Yang, Henry Power, Fritjof Fagerlund, Auli Niemi

Abstract

We explore the use of Gaussian process emulators (GPE) in the numerical simulation of CO₂ injection into a deep heterogeneous aquifer. The model domain is a two-dimensional, log-normally distributed stochastic permeability field. We first estimate the cumulative distribution functions (CDFs) of the CO₂ breakthrough time and the total mass using a computationally expensive Monte Carlo (MC) simulation. We then show that we can accurately reproduce these CDF estimates with a GPE, but using only a fraction of the computational cost compared to Monte Carlo. In order to build a GPE that can predict the simulator output from a permeability field consisting of 1000s of values, we use a truncated Karhunen-Loève expansion of the permeability field, and then use a Bayesian functional regression approach. We explore the use of GPEs for CO₂ storage problems, including the optimization of the experiment design. Finally, we provide perspectives for future applications.

Keywords: CO₂, heterogeneity, Bayesian, Permeability, KL expansion, Monte Carlo, Cumulative distribution function, Uncertainty analysis

This is a preprint version. The accepted version is found at DOI:10.1016/j.cageo.2017.04.006

Email addresses: liang.tian@geo.uu.se (Liang Tian), r.d.wilkinson@sheffield.ac.uk (Richard Wilkinson), zhibing.yang@hyd.uu.se (Zhibing Yang), henry.power@nottingham.ac.uk (Henry Power), fritjof.fagerlund@geo.uu.se (Fritjof Fagerlund), auli.niemi@geo.uu.se (Auli Niemi)

Preprint submitted to Computers & Geosciences May 2, 2017
Planning and operation of a carbon dioxide capture and storage (CCS) project requires reliable model predictions concerning the fate of the stored CO$_2$. Carefully conducted numerical simulations are critical for the understanding of the associated coupled physical and chemical processes (Pruess and García, 2002; Juanes et al., 2006; Doughty, 2007). An important additional complication arises from the geological heterogeneity of the target formation, such as stratigraphic architecture and facies distribution, which is difficult to estimate from the limited amount of observations available (i.e. from the sparse networks of primarily vertical investigation wells) in a deterministic manner (Ambrose et al., 2007; Tsang et al., 2008). Therefore, robust and computationally effective methods for dealing with the uncertainty arising from the geological heterogeneity are in great need. In general, two components contribute to the modeling uncertainty for CO$_2$ geological storage: (1) input uncertainty, including the aforementioned parameter uncertainties (unknown geology), and (2) model uncertainty, or “structural uncertainty” according to the conventional hydrological modeling terminology (Renard et al., 2010), as modeling approaches are developed under different conceptual and methodological frameworks, involving various approximations and simplifications. An example on the latter is the work reported by Nordbotten et al. (2012), where a benchmark simulation case was run with various numerical codes and effort was made to evaluate the significance of deviated solutions from various modelling strategies and assumptions. In the present work, we focus on the input uncertainty.

Standard geostatistical techniques are used to resolve the input uncertainty when evaluating reservoir CO$_2$ storage performance. For example, the Umbrella Point power plant model (based on the Frio formation) was created using TProGs program by Doughty and Pruess (2004) where multiple two-dimensional stochastic representations of fluvial depositional settings were picked deliberately to reproduce realistic three-dimensional geologic structures. A sequential indicator simulation approach was used by Flett et al. (2007) to create realistic
shale facies distribution for 3-D notional marine sand system models with varying net-sand-to-gross-shale ratios. A sequential Bayesian simulation technology was used by Claprood et al. (2014) in constructing a porosity distribution for a 3-D model of Beauharnois Formation to understand its CO$_2$ storage potential. In terms of the characterization of the spatial permeability distribution, Han et al. (2010) created multiple two-dimensional permeability fields with inclusion of low permeability lenses using a sequential Gaussian simulation approach. Discussions on effects of the permeability heterogeneity include the contributions from Jahangiri and Zhang (2011) with a focus on the plume distribution, and from Lengler et al. (2010) with a focus on small-scale heterogeneity ($< 100m$). Using a macroscopic invasion percolation model, Yang et al. (2013) performed a detailed parametric sensitivity study on upscaled capillary pressure saturation relative permeability relationships for CO$_2$ migration in multimodal heterogeneous media. A more recent sensitivity study was reported by Tian et al. (2016) where the parameters controlling the spatial correlation structures of the permeability fields were systematically analysed so as to understand their effects on CO$_2$ storage performance.

A Monte Carlo simulation method is normally used when a deterministic description of the model input cannot be used (James, 1980). In this approach, multiple, mutually different but equally likely realizations of the parameter field are generated, the model problem simulated for all of them, and the output analyzed in terms of the statistics of the outputs. The method has been proved viable for the simulation of geological storage of CO$_2$ (Jahangiri and Zhang, 2011; Deng et al., 2012; Tian et al., 2016). However, an obvious limitation for the method is the high computational cost, which limits the number of possible runs for large-scale, long-term simulations of CO$_2$ migration in 3-D heterogeneous medium. This in turn violates the underlying criteria of the Monte Carlo approach, which require the model to be run at many input configurations in order to accurately infer the uncertainty in the model predictions. Therefore, new reduced-order models that can capture the essential behavior of the fully physically based models, yet avoiding the prohibitive computational cost of them are
of great interest. A general overview on surrogate modeling in water resources was given by Razavi et al. (2012). More recently, Liu et al. (2013) developed geostatistical reduced order models (GROMs) in the parameter domain to solve under-determined inverse problems addressing subsurface multiphase transport.

In this paper, we propose a Bayesian approach for uncertainty analysis (UA), that is, the forward propagation of uncertainty through a model, such as TOUGH2 / ECO2N (Pruess et al., 1999; Pruess and Spycher, 2007), for the numerical simulations of CO\textsubscript{2} injection into a deep heterogeneous aquifer. The functionality of the numerical model, here called the simulator, is regarded as a mathematical function \( f(\cdot) \) which will generate the same output (quantities of interests) repeatedly given the same input. As we are uncertain about the input \( X \) (i.e., the true permeability is unknown), such uncertainty is transferred into \( f(X) \). The objective of UA is therefore to estimate the distribution of \( f(X) \), given the distribution of the inputs.

The remaining text has been organized to first presenting the methodologies, followed by results, discussions along with an concluding remark.

2. Methodology

2.1. CO\textsubscript{2} injection simulations using TOUGH2 model

The CO\textsubscript{2} injection simulations are performed using the TOUGH2/ECO2N code (Pruess et al., 1999; Pruess and Spycher, 2007). An idealized two-dimensional reservoir layer with heterogeneous permeability has been the basis for the analyses where the CO\textsubscript{2} is injected from a vertical borehole and the migration is observed until the plume front reaches the monitoring well at the farther end of the domain (Figure 1). The breakthrough time (BT) and the total mass (TM) of the injected CO\textsubscript{2} are the quantities of our interest. The modelling is briefed in Supporting Information (SI); detailed numerical procedures can be found in Tian et al. (2016)
2.2. Modelling the heterogeneous permeability field

We use the notation $Z$ for the spatial distribution of the permeability and assume the permeability can be modelled by a log-Gaussian random field (log-GRF). The $x$ in the notation $Z(x)$ is the location coordinates vector, emphasizing that the $Z$ is usually location dependent. For the purposes of computation, we only need to consider a finite-dimensional representation of $Z$ on a grid, which in our case is a $100 \times 20$ grid of values. Our prior model for $Z$ is then that

$$\log Z \sim N(\mu, \Sigma)$$

(1)

where we specify $\Sigma$ through a covariance function that describes the permeability covariance between any two locations in the domain (see SI).

Let $Z_i$ stand for the $i$th realization of the heterogeneous permeability field.

Our aim here is to conduct an uncertainty analysis to find the distribution of $f(Z)$ given the distribution for $Z$, where $f(\cdot)$ represents the simulator output (e.g. either the total mass or the breakthrough time of the CO$_2$ from the simulator). Our objective is to estimate the cumulative distribution functions (CDFs)

$$F(y) = \mathbb{P}(f(Z) \leq y)$$

(2)

which can be done using Monte Carlo if sufficient computer power is available. If $Z_1, \ldots, Z_n$ is a large sample from the log-GRF field, then the empirical CDF (ECDF),

$$F(t) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}_{f(Z_i) \leq t},$$

(3)
is an unbiased estimator of the CDF. Here, $I_A$ is an indicator function taking value 1 if $A$ occurred and 0 otherwise.

2.3. Gaussian process emulation

An emulator (Kennedy et al., 2008) is a statistical model that closely mirrors the simulator. It is built using an ensemble of input-output pairs \( \{X_i, y_i\}_{i=1}^N \) and will be used to predict the simulator output for any $y_i = f(X_i)$. The most popular approach to building emulators are Gaussian processes (GP) (Oakley and O’Hagan, 2002; Rasmussen and Williams, 2006), which are equivalent to the kriging models used in geostatistics (Deutsch and Journel, 1998), although usually GPs are thought of within a Bayesian formalism. GPs are popular because they are flexible non-parametric families that can capture a range of behaviours, from smooth cubic splines interpolants, to infinitely variable processes such as Brownian motion. They can incorporate other types of functional knowledge, such as derivative and integral information. A property that is particularly useful for uncertainty quantification applications is that they also come with an estimate of their accuracy, as all predictions are made in the form of distributions, not just point estimates as is the case with some other emulator approaches.

By exploiting the spatial structure in $Z$, we can construct an emulator from $Z$ to $f(Z)$ that uses a relatively small number of simulator evaluations. Our prior model for $Z(x)$, is that it is a log-Gaussian random field, with continuous covariance function (SI). The Karhunen-Loève (K-L) theorem says that $Z(x)$ admits a representation of the form

$$Z(x) = \sum_{i=1}^{\infty} \xi_i \lambda_i \phi_i(x)$$

where the $\lambda_i$ and $\phi_i(x)$ are the ordered eigenvalues and eigenfunctions of the covariance function, and the $\xi_i$ are independent $N(0,1)$ random variables. Note that if interest lies solely in the value of $Z$ on a finite grid of $n$ values (as in our case), then this reduces to finite sum of $n$ terms, and $\lambda_i$ and $\phi_i$ are the eigenvalues and eigenvectors of the covariance matrix $\Sigma$ formed from the
covariance function. To reconstruct $Z(x)$, only the $\{\xi_i\}_{i=1}^n$ need to be saved, since $\lambda_i$ and $\phi_i$ are determined by the covariance function and thus remain the same throughout the uncertainty analysis. We can thus consider the simulator to be a function of $\xi = \{\xi_i\}_{i=1}^N$ instead of $Z$, i.e., $f(Z) \equiv f(\xi)$.

If we order the eigenvalues so that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$, then by truncating the expansion to the first $d$ terms

$$\tilde{Z}(x) = \sum_{i=1}^d \xi_i \lambda_i \phi_i(x)$$

we can achieve a form of data compression, and represent the permeability in a lower dimensional space. This truncation explains the most variance and achieves the minimum mean square error amongst all such approximations. We will exploit this truncation to build a reduced order emulator. By building an emulator from $\tilde{Z}$ rather than $Z$, which is equivalent to building an emulator with input $\xi_1, \ldots, \xi_d$, we have reduced the dimensionality of the problem from $n$ to $d$, where we are free to choose $d$. All we require is that $f(\xi)$ is similar to $f(\xi + \delta)$ when $\delta$ is small.

There is a trade-off to be made when choosing $d$. If we choose $d$ too small we loose too much information in the compression from $Z$ to $\tilde{Z}$, but may be able to construct an accurate emulator. But if we choose $d$ too large, the data loss in the compression will be small, but we will not be able to construct an accurate emulator. We will examine the optimal value of $d$ using predictive performance measures in the next section.

A training ensemble of simulator runs is needed to build the emulator. As we are using a K-L expansion for the modelling of $Z$, this will be a set $\{\xi_i, y_i\}_{i=1}^N$ where each $\xi_i \in \mathbb{R}^d$. Careful selection of the training input locations can greatly improve GP predictions, and space-filling designs such as Latin hypercubes (McKay et al., 1979) are close to optimal in most cases. However, these designs are usually considered on a bounded domain. We could simply truncate the input domain, as each $\xi \sim N(0,1)$ and so truncating to $[-4,4]$ would capture nearly all of the inputs space in practice. However, we would find that the majority of training points, especially when $d$ is large, lay near the edge of the
domain. Thus, we would have only a few points in the region where most of the \( \xi \) will fall, and where we want to predict most accurately. One solution is to build a space filling design on \([0,1]^m\) and then transforming it to \(\mathbb{R}^d\) by pushing it through the inverse CDF of a \(N(0, \sigma^2)\) random variable. If we take \(\sigma^2 = 1\), then points lie in regions of input space in proportion to the density specified by the K-L theorem, however, we find this does not put enough training points on the edge of the domain, and so we find ourselves extrapolating rather than interpolating too often when we do the uncertainty analysis. For this reason, we use a value of \(\sigma^2 > 1\) to spread the points relative to their distribution. We used Latin Hypercube designs with a criteria of maximizing minimum distance (in \(\xi\)) in order to ensure a space-filling design (Morris and Mitchell, 1995).

GPs require that we specify a prior mean, a covariance function, and estimate hyperparameters involved in these two terms (training). We use a constant mean, and choose between with squared exponential and Matérn covariance functions, estimating hyper-parameters using type-II maximum likelihood (Rasmussen and Williams, 2006). We use the GPstuff implementation of Gaussian processes (Vanhatalo et al., 2012), which are a set of MATLAB computer codes integrating Gaussian process models for Bayesian analysis. Notice that the GP covariance function (also called the kernel) should be distinguished from the one
2.4. Using GP for UA

Once we have a GP emulator of the simulator, we can use this to estimate the simulator CDF and to quantify the uncertainty in our estimate. To estimate the CDFs, we use the procedure suggested in Oakley and O’Hagan (2002). This involves drawing sample functions \( \{ f_j \}_{j=1}^{L} \) from the GP that are consistent with the training data by adding in new design points \( \{ \xi^*_i \}_{i=1}^{1000} \), and simulating a value for the response from the GP emulator. We then update the emulator to take into account the fake simulated data. The placement and number of additional design points was chosen so as to make the uncertainty in the simulated functions \( f_j \) essentially zero. We then estimate the CDF for each simulated function using Monte Carlo in the usual manner, giving us \( L \) realizations \( F^*_1, \ldots, F^*_L \). From this we use the median of the CDFs as a point estimate, and can calculate uncertainty about our estimates using the ensemble of CDFs.

2.5. Numerical procedure

The two-phase flow behaviour as well as the migration pattern of the CO\(_2\) injection into a deep saline aquifer has been discussed in our previous study (Tian et al., 2016). We here focus on evaluating the ECDFs for the breakthrough time (BT) and the total mass (TM) of CO\(_2\) accumulated in the formation using GP.

We start our numerical experiments by varying the correlation structure of the permeability of the underlying geology (see SI).

In order to evaluate the performance of the GP emulator, two datasets are generated for each scenario: one consists of \( 10^4 \) input-output pairs and is used to produce a MC estimate of the CDF; the other is for emulator training and consists of only a small number of simulator evaluations. The numerical procedure is illustrated in Figure 2. The design points used for emulator training consist of far less data points than the Monte Carlo set. For the three cases considered (Table 1), we generate 800, 400 and 400 design points, respectively.
3. Results

3.1. Estimating the CDF

Each quantity of interest (TM or BT) from each of the three cases is considered as standalone model process. As the training set being based on Latin Hypercube design, we use a fixed number of training points (Table 1) to construct each of the three GP structures. For each emulated ECDF curve, 1,000 random sample points are first generated using a pseudorandom number (vector) generator in Matlab assuming a dimension corresponding to $d_{\text{train}} = 30$ (Case 1) or $d_{\text{train}} = 20$ (Case 2 and 3). Then this set of random inputs, altogether with the corresponding training pairs, were used to feed the designated GP structure in order to produce / draw one sample from the posterior distribution. For each quantity of interest, 100 posterior samples ($L = 100$) were used to calculate the median ECDF.

Figure 3 shows the breakthrough time of Case 1. GP is the median curve calculated from the 100 posterior samples. The confidence intervals of the MC CDF are omitted for visual clarity. The dashed lines (posterior credible intervals) indicate that the MC CDF can be enveloped by using only 100 emulator runs. Excellent matches are observed: for all cases examined, the median GP curves can replicate the MC ones almost exactly. The mean CRPS (Continuous Rank Probability Score, see also SI) for the three correlation length cases are 0.00640, 0.00193 and 0.00153 respectively. A similar procedure was used for the total CO$_2$ mass at the breakthrough time. The TM ECDF curves from the MC are also well predicted by the median GP results. The TM result exhibits slightly less good match in comparison to the observation from the BT, especially for the lower and upper tail of the ECDF. However, the 5th to the 95th percentiles of the GP prediction agree closely with the MC results. The CRPSs for three tested cases are, respectively, 0.00490, 0.00766 and 0.00975.

Note that for TM smaller CRPSs are observed for Case 1 in comparison to the other cases (Table 1) due to a larger number of training points ($n_{\text{train,case 1}} = 800$) as well as a higher dimension of training inputs ($d_{\text{case 1}} = 30$). The differ-
Table 1: Case specifications and results for model selection

<table>
<thead>
<tr>
<th>Case No.</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation length</td>
<td>0.075</td>
<td>0.15</td>
<td>0.30</td>
</tr>
<tr>
<td>size of MC set</td>
<td>$N_{MC}$</td>
<td>10,000</td>
<td>10,000</td>
</tr>
<tr>
<td>size of Training set</td>
<td>$n_{train}$</td>
<td>800</td>
<td>400</td>
</tr>
<tr>
<td>dimension of the Training set</td>
<td>$d_{train}$</td>
<td>30</td>
<td>20</td>
</tr>
<tr>
<td>CRPS$_{BT,\text{Matérn}}$</td>
<td>0.00640</td>
<td>0.00193</td>
<td>0.00153</td>
</tr>
<tr>
<td>CRPS$<em>{BT,\text{SE}}$ ($d</em>{train} = 20$)</td>
<td>0.00108</td>
<td>0.00187</td>
<td>0.00135</td>
</tr>
<tr>
<td>CRPS$_{TM,\text{Matérn}}$</td>
<td>0.00490</td>
<td>0.00766</td>
<td>0.00975</td>
</tr>
<tr>
<td>CRPS$<em>{TM,\text{SE}}$ ($d</em>{train} = 20$)</td>
<td>0.02489</td>
<td>0.02508</td>
<td>0.02534</td>
</tr>
</tbody>
</table>

ence is also spotted in that the overall CRPS for BT are noticeably smaller in comparison to the TM ones (one order of magnitude). The difference in confidence interval for BT results is visually indistinguishable (Figure 3). For Case 2 and Case 3, the results are visually similar to Case 2 and are therefore not included for space considerations.

3.2. Cross validation

At the initial stage of the experimental design, two key factors are very difficult to determine beforehand, namely the size of training set ($n_{train}$) and its dimension ($d_{train}$), the number of K-L components retained for the prediction. The cross validation using leave-one-out cross validation (LOO-CV, see also SI) provides a tool to tackle this issue. For each GP, the LOO-CV has been done in two steps: Step 1, a training set with fixed size is selected so that the sensitivity of prediction performance, in term of Dawid score (Wilkinson et al., 2011) (see also SI), is plotted as a function of the number of K-L components; Step 2, the number of K-L components is first fixed and then the prediction performance is plotted as a function of the size of the training set.

The LOO-CV result using Dawid score ($DS$) is plotted as a function of the number of K-L components (Fig. 4). It is found that by using a fixed size of the training set for all cases, the $DS$ score becomes stabilized when using more than 15 K-L components ($d_{train} \geq 15$). When using exactly 15 K-L components for
each case to fit the GPs, the DS score appears to become stabilized when using a training set with more than 100 data points ($n_{train} \geq 100$, see Fig. 5).

4. Discussion

The investigated model domain is $10 \times 100$ meters 2D model domain and has 2,000 elements representing a spatially correlated heterogeneous permeability
field. The computational cost for uncertainty analysis using the classical MC method is high due to the large number of multiphase flow simulations required. The main part of computational cost for GP emulator comes from the simulator runs needed for the training ensemble. GP posterior sampling has virtually no computational cost. In this section we discuss the design and the construction of the GP emulator in particular: 4.1 model configuration, 4.2 cross-validation and optimization, and 4.3 perspectives for uncertainty analysis.

4.1 Model configuration

In order for the GP emulator to be used as a practical tool for prediction, we will need to make decision about the specification of the Gaussian structure. One very important aspect is the choice of the covariance function that defines the nearness or similarity in the input space (Rasmussen and Williams, 2006). In other words, how some inputs \( x \) (vector) which are close be likely to have similar target \( y = f(x) \) values. The covariance function can formally be any functional form that can generate a non-negative covariance matrix for any set of inputs. Some of the very often used functions are the squared exponential covariance function (SE) and the Matérn class of functions. We constructed alternative GPs using both for each of the cases examined in Section 3 (see Table 1). The ECDFs calculated using Matérn \( (\nu = \frac{3}{2}) \) covariance function exhibit smaller CRPS values in comparison to the ones calculated using SE. For the emulation of BT, the choice of SE over Matérn does not have any noticeable effect. However, for TM the use of Matérn exhibits much better performance. Notice that the choice of \( d \) (the dimension of training points, in our case equivalent to the number of K-L components), which is related to the size of training points \( (n_{\text{train}}) \), will also affect the performance of the GP. We want to convey the point that the choice of covariance function can affect the performance of GP, however an extended discussion is beyond the scope of the current work.
4.2. Cross-validation and optimization

The question of model configuration is connected to the evaluation of the emulator performance. To investigate this we use the method of cross-validation (CV). The idea is to split the training set into two disjoint sets, one of which is used for the training and the other is used for the validation of the emulator. Notice that such splits can be done repeatedly in multiple ways ($k$-fold CV), one extreme case is when $k = n$, also known as leave-one-out cross-validation (LOO-CV). The question of optimization shall then address the optimum level of input dimensionality ($d_{\text{train}}$), which is the number of K-L coefficients and the number of design training points ($n_{\text{train}}$) to be retained in the GP. The evaluation is done by looking at variance of the predicted value in LOO-CV as well as DS for the overall prediction error.

In our calculations, the size of the training ensemble is 800 for Case 1 but 400 for Case 2 and Case 3. The reason is that we need to generate a larger training ensemble (number of realizations) to maintain comparable space filling when changing the correlation structures. For predicting the BT ECDF (Fig. 4), using GPs with 15 K-L components would have provided reasonably good results whereas for predicting the TM ECDF, around 20 K-L components would be preferred. The indication is that the calculations of breakthrough time and total mass for the injection simulation of CO$_2$ are two very different processes.

It is difficult to provide exact value for adequate or appropriate number of training points needed for a GP. A requirement for experiment design is to be able to fill up the input space ($\xi$). Optimization often starts from changing the space filling design, which in our case means to draw new sample from $\xi_i \in \mathbb{R}^{d=2000}$. To understand its improvement in GP performance the simulator (TOUGH2/ECO2N) needs to be rerun so as to generate the corresponding new training ensembles. In other word, one may need to build new GPs based on additional simulator runs to understand the potential gain from optimization. This will be very time consuming. A different approach has been applied.

Considering Case 1, for example, where we have generated 800 training pairs ($n_{\text{train}} = 800$), we start building a first $GP_{0,j=20}$ by a random draw of $j = 20$
pairs out of the 800. A first DS score can then be calculated for $GP_{0,j=20}$ using LOO-CV. By randomly sampling one at a time from the remaining training pairs, we can iteratively create $GP_{i,j=20+i}$ and the resulting DS scores should reflect that we can improve the prediction performance by taking into account
more information from the prior. It should be noted that latin-hypercube sampling has been used to create the initial 800 points. The re-sample of the existing latin-hypercube set should be path-independent. In Figure 5, the decreasing trend of DS score reflects that more information from training pairs is
taken into account. It can been seen that 100 training pairs would be needed
for Case 1 when building GP for BT ECDF using only 15 K-L components.
Note that the pattern of TM LOO-CV result for Case 1 (Fig. 5, lower panel) is
different from the other cases. We further extended the LOO-CV test for Case 1
and the decreasing trend in the DS score was confirmed (Fig. 6). This indicates
that for heterogeneous domain with smaller correlation length, a larger training
set may be needed for constructing the GP so as to enable similar predicting
performance comparing to the cases with relatively large correlation length.

Figure 6: LOO-CV Scores vs. the size of the training set ($n_{\text{train}}$), Case 1

4.3. Using GP for uncertainty analysis

In Section 2.3, we gave a technical overview on using the Bayesian formalism
for predicting results from a complex computer code. It was then extended in
Section 2.4 for uncertainty analysis where the cumulative distribution function
is the speculated statistic of interest. The output from each GP constructed
in Section 3 is a collection of random variables indexed by $\xi$ (where $d$ is its
dimensionality). An assumption has been made that the spatial distribution of
the heterogeneous field can adequately be described by $\xi$. Using a geostatistics perspective, the conventional perception of correlation length ($\lambda$) and the descriptive covariance function (see SI) of the permeability field can both be interpreted as possible projections of $\xi$.

We therefore used standalone GP in predicting the ECDF for each uncertain output of interest. It is worth noting that even for a simple numerical experiment the different outputs, in our case the breakthrough time and the total mass, are fundamentally different processes. This has been observed from the CPRS for each predicted result (Table 1). The physical indication is that the main driver for breakthrough time is the medium permeability where in a thin heterogeneous aquifer it is analogue to conventional 1D single phase problem (relative simple processes). However for the total CO$_2$ mass at the breakthrough time, a simple analogy does not exist. More factors / parameters shall be considered such as pathway properties, gravity effects and multiphase interactions.

We have shown that the use of GP for UA, in our case exploring the ECDFs of BT and TM, results in considerably reduced computational cost compared to classical MC analyses. By improving experiment design as exemplified in cross validation, it is possible to further improve the model performance.

5. Concluding remarks

We have conducted Monte Carlo simulations for the uncertainty analysis of the breakthrough time (BT) and total mass accumulation (TM) for CO$_2$ injection simulation into a deep heterogeneous saline aquifer. Multiple GP structures were created and were successfully used to predict ECDFs for the selected cases. It has been shown that the use of a Gaussian process emulator can replicate the ECDF calculated using 10,000 Monte Carlo simulations (realizations) at only a fraction of the computational cost.

We looked at the issues encountered in the experiment design and explored the possibility to further optimize the GP. An optimum design may need to re-sample the input space, and therefore need additional simulator runs. To ad-
dress this, an alternative approach was taken by down-sampling the training set. The results from the cross validation exemplified using DS score indicate significant performance gain from potential optimization. This information provides a good starting point for further applications. The present work is a first attempt to adapt the tool in the uncertainty analysis of the numerical simulations of geological CO$_2$ storage using TOUGH2 / ECO2N.

We treat the two outputs namely the CO$_2$ breakthrough time and the total CO$_2$ mass as two physical processes and therefore built standalone GPs for each one. In fact, it is possible to construct a single GP with multiple outputs (Alvarez et al., 2011). This provides one future perspective for exploring the internal physical mechanism for a complex system such as the one of our interest. Another future aspect is in the application of the GP through exploring the control over the accuracy of the prediction with, for example using multi-level GP trainings (analogy to the multi-level Monte Carlo).

We explored also the indication from modelling of heterogeneous media and identified that the conventional perception on correlation length is, from a geostatistic perspective, a matter of parameter bounds and dimensions. Future studies should resolve the correlations between the parameters with physical meaning to those critical in describing the statistics. More studies are needed to evaluate how GP emulators can be effectively used for complex modelling analyses of heterogeneous systems.

Acknowledgement

The authors would like to acknowledge Prof. Andrew Cliffe at the School of Mathematical Sciences, University of Nottingham for his invaluable contributions to this work. Unfortunately Prof. Cliffe passed away during the preparation of this manuscript. He will be remembered for his great qualities as a researcher and friend.

The work has been supported by the European Community’s Seventh Framework Programme (FP7) MUSTANG (No. 227286), PANACEA (No. 282900)
and TRUST (No. 309067) projects. The computations were performed on re-
resources at Uppsala Multidisciplinary Center for Advanced Computational Sci-
ence (UPPMAX), Chalmers Centre for Computational Science and Engineering
(C3SE) both provided by the Swedish National Infrastructure for Computing
(SNIC).

References


Ambrose, W. A., Lakshminarasimhan, S., Holtz, M. H., Núñez-López, V., Hov-
orka, S. D., Duncan, I., 2007. Geologic factors controlling CO2 storage capac-
ity and permanence: case studies based on experience with heterogeneity in
oil and gas reservoirs applied to CO2 storage. Environmental Geology 54 (8),
1619–1633.

Adapted sequential Gaussian simulations with Bayesian approach to evaluate
the CO2 storage potential in low porosity environment. Greenhouse Gases:
Science and Technology 4 (6), 761–776.

of industrial-scale CO2 storage: Multi-scale heterogeneity and its impacts on
storage capacity, injectivity and leakage. International Journal of Greenhouse
Gas Control 10 (0), 397–418.

Deutsch, C. V., Journel, A. G., 1998. GSLib. Geostatistical software library and
user’s guide, 369.

of non-hysteretic and hysteretic characteristic curves. Energy Conversion and
Management 48 (6), 1768–1781.


Supporting Information for

Gaussian Process Emulators for Quantifying Uncertainty in CO\textsubscript{2} Spreading Predictions in Heterogeneous Media

Liang Tian\textsuperscript{a}, Richard Wilkinson\textsuperscript{b}, Zhibing Yang\textsuperscript{a}, Henry Power\textsuperscript{c}, Fritjof Fagerlund\textsuperscript{a}, Auli Niemi\textsuperscript{a}

\textsuperscript{a}Air, Water and Landscape Sciences, Department of Earth Sciences, Villav"agen 16, SE-752 36 Uppsala University, Sweden
\textsuperscript{b}School of Mathematics and Statistics, University of Sheffield, UK
\textsuperscript{c}Faculty of Engineering, University of Nottingham, University Park, Nottingham NG7 2RD, UK

In this document, we present extended descriptions for (1) CO\textsubscript{2} storage modelling using TOUGH2/ECO2N; (2) the modelling of the heterogeneous permeability fields; and (3) evaluating the Gaussian process emulator.

1. CO\textsubscript{2} injection simulations using TOUGH2 model

An idealized two-dimensional heterogeneous reservoir layer (Figure 1) has been the basis for the analyses on the breakthrough time (BT) and the total mass (TM) of the injected CO\textsubscript{2}.

The injection well consists of one column of homogeneous numerical elements and is treated as a specified pressure boundary where the pressure distribution is calculated beforehand by assuming that the injection elements are fully filled with pressurized supercritical CO\textsubscript{2}. The initial hydrostatic condition for the rest of the model is also calculated beforehand by assuming it is fully saturated with formation brine (salinity 0.7 mol/kg). The hydrostatic pressure condition is

\begin{itemize}
  \item Email addresses: liang.tian@geo.uu.se (Liang Tian),
  \quad r.d.wilkinson@sheffield.ac.uk (Richard Wilkinson),
  \quad zhbing.yang@hyd.uu.se (Zhibing Yang),
  \quad henry.power@nottingham.ac.uk (Henry Power),
  \quad fritjof.fagerlund@geo.uu.se (Fritjof Fagerlund),
  \quad auli.niemi@geo.uu.se (Auli Niemi)
\end{itemize}
maintained throughout the injection simulation. For CO$_2$ injection simulation, for each heterogeneous permeability field realization, the injection simulation is continued until the front of the plume reaches the right boundary of the domain. The breakthrough time as well as the total mass of CO$_2$ remaining in the model domain at that time (trapped instantaneously as gaseous or dissolved CO$_2$) are the quantities of interest.

Figure 1: Conceptual model of the simulation domain (Tian et al., 2016)

The two-phase flow characteristic functions are the van Genuchten model (Van Genuchten, 1980) for the capillary pressuresaturation function as well as the relative permeability of the liquid phase, and the Corey model (Corey, 1954) for the relative permeability of the gaseous phase. The Leverett scaling (Leverett, 1941) is applied, i.e., capillary entry pressure is scaled in relation to the permeability according to

\[
P_c = P_{c,ref} \sqrt{\frac{k_{ref}}{k}},
\]

where \(k\) and \(P_c\) are intrinsic permeability and capillary entry pressure at a given location, \(k_{ref}\) and \(P_{c,ref}\) are respectively the reference values for intrinsic permeability and capillary pressure. The relationship accounts for the fact that domains of lower permeability have higher capillary entry pressures. The parameters used for TOUGH2 simulations are summarized in Table 1 and represent characteristics that can be encountered in CO$_2$ storage reservoirs.
Table 1: Parameters for TOUGH2 numerical simulations (Tian et al., 2016)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Porosity [%]</td>
<td>0.15</td>
</tr>
<tr>
<td>Pore compressibility [Pa(^{-1})]</td>
<td>4.5 \times 10^{-10}</td>
</tr>
<tr>
<td>Reference permeability, (k_{ref}) [m(^2)]</td>
<td>1 \times 10^{-13}</td>
</tr>
<tr>
<td>Temperature [\degree C]</td>
<td>50</td>
</tr>
<tr>
<td>Salinity [mol/kg]</td>
<td>0.7</td>
</tr>
<tr>
<td>Initial hydrostatic pressure at the formation top, (P_{eq}(Z_0)) [MPa]</td>
<td>15.5 (i.e. (Z_0 = -1550)m)</td>
</tr>
<tr>
<td>Injection pressure at the topmost section of injection well [MPa]</td>
<td>18.6</td>
</tr>
<tr>
<td>Relative permeability</td>
<td></td>
</tr>
<tr>
<td>(S_{l,r}) [%]</td>
<td>0.200</td>
</tr>
<tr>
<td>(S_{l,r,\text{max}}) [%]</td>
<td>1.000</td>
</tr>
<tr>
<td>(S_{g,r}) [%]</td>
<td>0.050</td>
</tr>
<tr>
<td>(m) [%]</td>
<td>0.457</td>
</tr>
<tr>
<td>Capillary pressure</td>
<td></td>
</tr>
<tr>
<td>(S_{l,r}) [%]</td>
<td>0.199</td>
</tr>
<tr>
<td>(S_{l,r,\text{max}}) [%]</td>
<td>0.999</td>
</tr>
<tr>
<td>(m) [%]</td>
<td>0.457</td>
</tr>
<tr>
<td>(P_{e,ref}) [Pa]</td>
<td>1.963 \times 10^{4}</td>
</tr>
</tbody>
</table>

2. Modelling the heterogeneous permeability fields

In common with (Cliffe et al., 2011; Deutsch and Journel, 1998), we assume an isotropic exponential covariance function for a permeability field:

\[
c(x, x') = \sigma^2 \exp \left( -\sum_{i=1}^{2} \frac{|x_i - x'_i|}{\lambda_i} \right)
\] (2)

where \(\lambda = (\lambda_1, \lambda_2)\) is the characteristic length scale (correlation length) in each dimension, and \(\sigma^2\) represents the variance in term of the isotropic absolute permeability. The term \(|x_i - x'_i|\) is the distance between the two points on coordinate axis \(i\).

We considered three cases of different \(\lambda_1\) (the length-scale in the horizontal direction) through varying \(\lambda_1 = \lambda_x / H_x = 0.075, 0.15, 0.30\), where \(H_x\) is the domain length in the horizontal direction. All the cases are with fixed \(\lambda_2\) at
\[ \lambda_2 = \frac{\lambda_z}{H_z} = 0.30 , \text{ where } \lambda_2 \text{ is the vertical length-scale, and } H_z \text{ is the vertical domain length.} \]

3. Evaluating the Gaussian process emulator

Building an emulator is an exercise in regression. We wish to build an emulator, \( \hat{f}(\cdot) \), which approximates the simulator, \( f(\cdot) \). The difficulty of this task depends upon the smoothness of the simulator response, and the dimension of the problem. In our case, the input to the simulator is the 2-D permeability field, \( Z(x) \) where \( x \) indexes the location in the domain. Direct application of the usual Gaussian process approach will not work, as even if we only consider the permeability on the grid of 100 \( \times \) 20 values required by the numerical solver, this would still require us to construct an emulator with 2000 inputs, a task which would require at least as many simulator evaluations as estimating the CDF using classical Monte Carlo.

We use two approaches to evaluate the GP emulator. Firstly, we evaluate the GP prediction of the simulator using cross-validation (CV). The idea is to split the training set into two disjoint sets, one part of which is used for model fitting, and the other is held-back for validation of the prediction. An particular type of CV is leave-one-out cross validation (LOO-CV), in which we leave out each of the \( N - 1 \) training points in turn, build a GP model using the \( N - 1 \) remaining points, and predict the left out point and then compute the prediction error. To score the set of \( N \) predictions, we use a proper scoring rule (Gneiting and Raftery, 2007) and use the Dawid score, which depends only on the first two moments of the prediction, to evaluate the GP performance (Wilkinson et al., 2011). This is given by

\[
DS(\hat{f}) = \frac{1}{N} \sum_{j=1}^{N} \left( \frac{(y_j - m_j)^2}{s_j^2} + \log s_j^2 \right) \quad (3)
\]

where \( j \) indices across the training points, \( y_j \) is the simulator value, and \( m_j \) and \( s_j^2 \) are the mean and variance of the emulator prediction when we have trained the emulator leaving out training point \( \{\xi_i, y_i\} \). The use of a proper
score ensures we are assessing the emulator on the basis of its mean prediction, and the uncertainty in that prediction (the variance).

The second method of assessment is to evaluate the approach of using GP emulators to estimate CDFs using the following score, which is based upon the continuously ranked probability score (CRPS) (Gneiting and Raftery, 2007) and which compares the GP emulator estimate of the CDF with an accurate estimate obtained from a larger Monte Carlo sample:

\[
CRPS(\hat{f}) = \frac{1}{L} \sum_{i=1}^{L} \int_{x=-\infty}^{x=\infty} (F_j(x) - F(x))^2 \, dx.
\] (4)

Here, \(F_j\) are posterior samples of the CDF generated using the emulator, and \(F\) is the empirical CDF from a large Monte Carlo analysis, used as an estimate of the truth. If we find that \(CRPS(\hat{f}_1) \leq CRPS(\hat{f}_2)\), we can deduce that emulator \(\hat{f}_1\) is superior to \(\hat{f}_2\) for CDF estimation. This score, while assessing the accuracy in the quantity of interest, is only useful when we have an estimate of the true \(F\). Note that in practical problems, we would not usually be able to do this, as if we could afford to create a large Monte Carlo sample, then there would be no reason to use an emulator. However, it is useful in this case to assess the entire estimation procedure, as the LOO-CV approach only assesses the emulator’s ability to predict the simulator output in a specific way, and it is possible, for example, that this would fail to diagnose problems with tail predictions.

References


