A Bayesian Optimisation Workflow for Field Development Planning Under Geological Uncertainty

R. Bordas[1*], J.R. Heritage[1], M.A. Javed[1], G. Peacock[1], T. Taha[1], P. Ward[1], I. Vernon[2], R.P. Hammersley[1]

[1] Emerson Exploration & Production Software; [2] Department of Mathematical Sciences, Durham University

Summary

Field development planning using reservoir models is a key step in the field development process. Numerical optimisation of specific field development strategies is often used to aid planning. Bayesian Optimisation is a popular optimisation method that has previously been applied to this problem. However, reservoir models can have a high degree of geological uncertainty associated with them, even after history matching. It is important to be able to perform optimisation that accounts for this uncertainty. To date, limited attention has been given to Bayesian Optimisation of field development strategies under geological uncertainty.

Much of the recent work in this area has focused on Ensemble Optimisation methods. These naturally handle geological uncertainty using ensembles of geological realisations. This can result in a high computational cost, as large ensembles are required to capture the geological uncertainty. Bayesian Optimisation offers an alternative solution using probabilistic surrogate or proxy models that can capture the geological uncertainty. However, incorporating geological uncertainty into proxy models and using those models in a Bayesian Optimisation loop remains a challenging task. Further, the effect of the additional proxy model uncertainty on optimisation results has not been well studied.

We propose a Bayesian Optimisation workflow comprising a Stochastic Bayes Linear proxy model and a combination of experimental and sequential design techniques. The workflow is designed to include a combination of static and dynamic uncertainties, with a new geological realisation generated and used to simulate fluid flow during each run of the model. The workflow is demonstrated by optimising several field development strategies in a synthetic North Sea reservoir model. The ability of the workflow to locate optima and correctly account for the geological uncertainty is studied and the computational cost is quantified.

The performance and practical implications of the proposed approach are discussed. These are important in designing an accurate and computationally efficient optimisation workflow under geological uncertainty and, ultimately, are factors in developing decision support tools for field development.

Introduction

Numerical optimisation of specific field development strategies is often applied to reservoir simulation models and used to aid the field development planning process. The goal is to find values for a set of control parameters that maximise an objective function. Typically, the objective function quantifies the revenue from the field, balanced against the operating costs and both geological and economic constraints. It can vary from as simple as the total oil recovered through to a comprehensive estimate of the net present value (NPV) of the field. The control parameters are defined by the field development strategy. For example, an engineer may wish to find the best control strategy for the drilled wells in the field. In this case, controls could include targets for production and injection rates. Other scenarios may involve drilling new wells, in which case the controls would include the location, trajectory and drill time for the new wells.

Many numerical optimisation schemes have been applied to these problems in the absence of geological uncertainty. Well control problems have been approached using both gradient and adjoint based techniques (Jansen, 2011) and derivative free techniques (Ciaurri, et al., 2011). Discontinuities in the response due to geological features such as faults have led to derivative free approaches being taken for well placement optimisation. For example, (Emerick, et al., 2009) optimised the placement of wells using a genetic algorithm with non-linear constraints. Genetic algorithms have also been applied to the problem in conjunction with statistical proxies (Artus, et al., 2006). More recently, (Alrashdi & Sayyafzadeh, 2019) compared an $(\mu + \lambda)$ evolution strategy algorithm with a genetic algorithm, a particle swarm optimiser and an covariance matrix adaptation evolution strategy for both well placement & trajectory optimisation and well control optimisation.

The lack of direct subsurface observations means that reservoir models often have a high degree of geological uncertainty. From a probabilistic perspective, this means that the reservoir model will be parameterised over a set of inputs, representing geological uncertainties, that are characterised by a probability distribution. In the case of appraisal studies, where no production data is available for the field, there will be a high level of geological uncertainty and uncertainty parameters will be specified by expert defined analytic prior distributions. In the case of previously developed, or brownfield, reservoirs the geological uncertainty will have been reduced by history matching the model to production data and other measurements. In this case, the probability distribution of the uncertainty parameters, termed the posterior distribution, will be a numerical approximation and is usually specified by a set of samples derived from the history matching process. The addition of further data and calibration of the model can reduce geological uncertainty, but the uncertainty can never be eliminated. Thus, it is important to account for this uncertainty when performing optimisation of a reservoir development scenario – a process known as optimisation under uncertainty.

Differing numbers of samples may be available when the distribution of the uncertainty parameters is defined as the result of a history matching process. Ideally, there will be a large number of samples and the posterior probability distribution will be well characterised. This is typically the case for history matching approaches that use statistical emulators and Markov chain Monte Carlo (MCMC) or similar approaches to obtain samples of the posterior distribution (Fillacier, et al., 2014). However, some history matching approaches, notably those designed for high dimensional history matching problems such as the Ensemble Kalman filter (Evensen, 2003) or Ensemble Smoother (Emerick & Reynolds, 2013), limit the number of samples of the posterior distribution to a fixed size ensemble of geological realisations. As we will see, some optimisation under uncertainty approaches are specifically designed to work with a set of fixed realisations such as these. The optimisation under uncertainty method described here can work with such a fixed set of realisations. One of its main benefits, however, is the ability to refine its representation of the geological uncertainty by incorporating further samples, resulting in extra geological realisations. As such, it is most naturally applied in the appraisal case or in the brownfield case in conjunction with a similar, emulation based, history matching process.

Methods for performing optimisation under uncertainty have been an active research topic in recent years. In part, this has been driven by benchmark challenges such as the Brugge (Peters, et al., 2010) and OLYMPUS (Fonseca, et al., 2018) challenges. The Brugge challenge featured both history matching and follow on optimisation components. This provided more variety in the uncertainty data that was used with the optimisation scheme and consequently more variety in the optimisation under uncertainty approaches used. The OLYMPUS challenge focused solely on optimisation and provided a set of 50 geological realisations to describe the uncertain geology.

Ensemble optimisation methods, which are a combination of gradient based optimisation with geological uncertainties described by a fixed set of realisations, have been very successful for problems of this type. The first ensemble optimisation method was described by (Lorentzen, et al., 2006), who adapted the ensemble Kalman filter to optimise water flooding in a reservoir model. (Chen, et al., 2009) used ensemble optimisation in a closed loop scheme to optimise well controls. More recently, (Fonseca, et al., 2017) introduced a refined version of ensemble optimisation and benchmarked their scheme against several others. Ensemble optimisation schemes were well represented in the entries to the OLYMPUS optimisation challenge.

Bayesian optimisation (Shahriari, et al., 2016) and statistical emulators, or proxy models, have previously been applied to optimisation under uncertainty of reservoir models. (Schulte, et al., 2020) built separate emulators for a fixed set of geological realisations and used them to help optimise well placement in a geothermal reservoir. (Goodwin, 2018) applied emulators to the OLYMPUS challenge. Separate emulators were constructed for each quantity of interest rather than for each geological realisation. Geological realisations were included as a categorical variable in the emulator and a mapping between this variable and the objective was used to account for the geological uncertainty. This allowed the emulators to be updated using single simulation runs, each with different geological models and control parameter values.

In this paper, we develop a Bayesian Optimisation workflow for reservoir model optimisation under uncertainty based on a stochastic Bayes Linear emulator. Similar methods have been used with stochastic models in other domains (Andrianakis, et al., 2015), (Ankenman, et al., 2010). However, they have not been used for reservoir model optimisation under uncertainty. The workflow differs from existing emulator-based approaches to optimisation of reservoir models in that separate emulators are not constructed for a fixed ensemble of geological models. Nor is an auxiliary categorical variable used to account for different realisations. Instead, a single emulator is constructed for each quantity of interest and the emulator accounts for the geological uncertainty in a statistically consistent way. A relatively small number of repeated reservoir model runs, with differing geological parameter values, are executed for each tested control parameter set. The set of geological parameter values that can be used is not fixed *a priori*. Instead, new geological parameter values are sampled each time from the probability distribution that defines the geological uncertainty. The number of repetitions can be dynamically increased for control parameter sets that are of interest. This allows further understanding of the effect of the geological uncertainty and refinement of the emulator in this location. The emulator is used in an adapted Bayesian optimisation framework, where experimental design techniques are used to create an initial set of runs to build the emulator and sequential design techniques are used to optimise the control parameter values and refine the emulator in regions of interest. This allows for a flexible optimisation scheme that can use a low number of simulations runs while respecting geological uncertainty.

Theory and Methods

Overview

The workflow supports optimisation of control parameters in simulation models under both static and dynamic geological uncertainties. A common platform integrator allows multiple software components to be joined together to form a composite reservoir model. Most commonly, and in the third example described here, a geological modelling package is combined with reservoir fluid flow simulation package to create a model with both static and dynamic uncertainties (Aarnes, et al., 2015). The integrator allows other software packages to be incorporated to create the model to be optimised. For example, (Aslam & Bordas, 2020) integrated a discrete fracture modelling package with a fluid flow simulator to study history matching in shale reservoirs.

The mathematical framework uses a statistical emulator for the composite simulation model to accelerate optimisation. The emulator acts as a probabilistic surrogate or proxy for the model, which allows fast model approximation and a measure of the uncertainty associated with the approximation. The emulator is built using results from model runs with inputs carefully chosen using a combination of experimental and sequential design techniques. Repeated runs with common control parameters and varying geological parameters are used to assess the impact of the geological uncertainties and inform the emulator.

A Mathematical Framework for Reservoir Models with Geological Uncertainty

Consider a deterministic reservoir simulation model, $f(x,y)$, where x is vector of control input values, such as production rate targets, well pressures and well locations, and y a vector of geological input values, such as permeabilities and porosities. In models that use stochastic geological modelling algorithms, y may also contain random seed values. f itself is a vector of production values of interest, such as production rates and totals, with individual entries in the vector denoted f_j. As the model is deterministic, running it twice using the same input parameter settings for both x and y results in the same vector of outputs, $f(x,y)$.

However, in a reservoir model with geological uncertainty the vector y of geological input values is not fixed. Instead the geological uncertainty is defined as a probability distribution, $\pi(Y)$. This distribution may be defined by expert knowledge, as in the case of an appraisal study, or be approximated as the result of some history matching process. The geological input values to the reservoir model can be considered as a random variable with this distribution, $Y \sim \pi(Y)$. Individual realisations of this random variable, y, can then be used to run the reservoir simulation model in conjunction with a chosen set of control inputs.

Under geological uncertainty, the reservoir model is a function of the random variable Y. For fixed control inputs, x, its outputs are also a random variable with an unknown distribution, such that

$$F(x) = f(x, y = Y),$$

$$Y \sim \pi(Y).$$

Repeated evaluations of $F(x)$ with the same control input parameter settings give different results, as the value of the geological input parameters changes each time. In the case of a composite model containing a geological modelling package this will also result in a different geological realisations being created each time.

Objective Function

A flexible system of user defined quantities of interest, termed optimisation points, is used to construct an objective function. Optimisation points can be placed on any scalar output from the model. In the case of time series outputs, this means that the optimisation point is defined at a specific time. Multiple optimisation points can be defined at different times on each time series if required. In the framework above, each optimisation point corresponds to an entry F_j in the vector of outputs of interest, F. As F is a random variable, there are many choices over which specific feature of its distribution we may wish to optimize. In this initial work, we choose to optimise the mean, or expected value, of each output, denoted $\mathbb{E}\left(F_j(x)\right)$. The framework can be extended to optimise other choices through emulation of the variance surface (Andrianakis, et al., 2016).

Each optimisation point is assigned a weighting, w_j, by the user. Weightings allow the relative dimensionless contribution of each optimisation point to be set and may be negative to allow desired quantities to be minimised. The final composite objective function is given by the sum of the expected value of the model at the optimisation points multiplied by their weightings. Thus, the objective function m is defined as

$$m(x) = \sum_j w_j \mathbb{E}(F_j(x)).$$

Emulator Construction

For simplicity, assume $F(x)$ has just a single scalar output, and so an emulator is constructed for $m(x) = \mathbb{E}(F(x))$. The method shown can be extended to the vector output case by constructing a separate emulator for each output. The variance surface is not separately emulated, as it is not required when optimising on the expectation of F.

The task of constructing an emulator for $m(x)$ is made more difficult as we cannot directly evaluate $m(x)$ for a given x. Instead $m(x)$ is approximated using the sample means from a finite number of repeated runs with common control parameters. To achieve this, r repeated evaluations of $F(x)$ are performed with fixed control parameters, x. The number of repeats is not fixed and can be scaled dynamically as required. For r repeats, the sample mean is calculated as an estimate for the true mean,

$$\overline{F}^r = \frac{1}{r}\sum_{j=1}^{r} F(x).$$

The emulation approach is based on *Bayes linear* methodology (Goldstein & Wooff, 2007), which is equivalent to Gaussian Process regression with partially specified probability distributions (Goldstein, 2012). The emulator takes the form

$$m(x) = \sum_k \beta_k g_k(x) + u(x) + v^r(x),$$

where β_k are unknown constants, $g_k(x)$ are known deterministic functions of the inputs, $u(x)$ is a weakly stationary stochastic process with zero mean and a chosen correlation function that represents the spatial error in the emulator and $v^r(x)$ is a stochastic process that describes the difference between the sample mean estimate and the true expectation of $m(x)$.

The first terms in the emulator are chosen to be a second-order multivariable polynomial with additional third-order terms. These terms are used to capture the global trends in the response and any low order non-linearities. Not all terms are used by the emulator. The terms used are selected by a stepwise regression process. In stepwise regression, the terms (starting with the lowest order terms) are added one at a time and the polynomial fitted to existing run data. The Bayesian Information Criterion (BIC) (Schwarz, 1978) is calculated at each step. If the new term does not improve the BIC over the previous model it is discarded, otherwise, it is kept and stepwise regression proceeds to the next term. Further restrictions on the number of terms that can be selected are imposed using a heuristic based on the total number of available runs. The stepwise process and these extra restrictions act both to prevent overfitting and as a form of automatic relevance determination. Control parameters that are present in the stepwise regression terms are classified as *active* and those that are not classified as *inactive*, this affects how they are treated in the local variation error term. When all terms have been selected, the final polynomial is regressed to the data and the β coefficients are calculated together with their variance.

The $u(x)$ term expresses the local variation and is calculated after the linear regression terms. It is modeled as a weakly second-order stationary stochastic process with covariance structure of the form

$$\mathrm{Cov}\big(u(x), u(x')\big) = \sigma^2 \exp(-\theta_{act}|x_{act} - x'_{act}|^2 - \theta_{inact}|x_{inact} - x'_{inact}|^2),$$

Where x, x' are two vectors of input uncertainty parameters, x_{act} and x_{inact} are projections of these vectors on to the space of active and inactive parameters and σ^2 is the variance of the residuals after linear regression. The parameters θ_{act} and θ_{inact} are hyperparameters fitted to obtain the maximum likelihood that the emulator matches the run data. Thus, the emulator has separate characteristic length scales for the active and inactive control parameters.

The $v^r(x)$ term is a stochastic process, uncorrelated with β_k and $u(x)$, that describes the difference between the sample mean estimate and the true mean. It is this term that allows the emulator to account for geological uncertainty and its effect on the quantity of interest. The expectation of $v^r(x)$ is zero due to the sample mean being an unbiased estimator for $m(x)$ and, due to the independence of the realisations from the stochastic simulator, its variance and covariance are given by

$$\mathbb{E}[v^r(x)] = 0,$$
$$\mathrm{Var}[v^r(x)] = \frac{\sigma_s^2(x)}{r},$$
$$\mathrm{Cov}\big(v^r(x), v^r(x')\big) = \begin{cases} \frac{\sigma_s^2(x)}{r}, & x = x' \\ 0, & x \neq x' \end{cases}$$

where $\sigma_s^2(x) = \mathrm{Var}[F(\mathrm{x})]$ is the variance of the uncertain model output. As we do not know the true variance of the model output, we use the sample variance, $s_r^2(x)$, of r repeated runs as an unbiased estimator, such that

$$\mathrm{Var}[v^r(x)] \approx \frac{s_r^2(x)}{r}.$$

The $v^r(x)$ term allows an independent estimate of the error due to geological uncertainty in the emulator's estimate of the expected value at each design point. By increasing the number of runs, r, this error can be reduced as needed in areas of interest.

Run Design & Updating the Emulator

Once constructed, the emulator allows quick approximation of the expected value of reservoir outputs. An initial set of model runs, referred to as *Scoping Runs*, are required to act as a training data set to fit the emulator. The parameter values for Scoping Runs are carefully chosen using experimental design techniques. Subsequent batches of runs, referred to as *Refinement Runs,* are used to find optimal control parameter values and update the emulator. The parameter values for Refinement Runs are chosen using sequential design techniques.

The values of the control inputs used for the initial Scoping Runs are carefully chosen to explore the space of control parameters while respecting the probability distributions associated with the geological parameters. The control parameter input values are generated using a Latin Hypercube Sampling (LHS) design (Stein, 1987). Many LHS designs are generated and the best is selected via a maximin criteria. This gives good coverage of control parameter space and reduces spurious associations between the parameters. At each control parameter input location, $x^i, i = 1, \dots, n$, we perform r_i repetitions of the model, where each r_i may be different for different i, with differing geological parameter inputs. For each repetition run, numbered $j = 1, \dots, r_i$, independent samples y^j are drawn from the distribution $\pi(Y)$ and the reservoir model is run. This gives model outputs

$$F^j\big(x^i\big) = f\big(x^i, y^j\big), \ \ i = 1, \dots, n, \ \ j = 1, \dots, r_i.$$

The sample means and variances of the responses are calculated for each set of control inputs, x^i, as described in the preceding section, to obtain a vector of sample means, D, and sample variances. These are combined with variances and covariances due to the spatial terms in the emulator. We may then use the standard Bayes linear update formula to obtain the adjusted expectation, $\mathrm{E}_D(m(x))$ and variance $\mathrm{Var}_D(m(x))$ for the expected value of the reservoir model $m(x)$ at an input point x:

$$\mathrm{E}_D\big(m(x)\big) = \mathrm{E}\big(m(x)\big) + \mathrm{Cov}(m(x), D)\mathrm{Var}(D)^{-1}\big(D - E(D)\big),$$
$$\mathrm{Var}_D\big(m(x)\big) = \mathrm{Var}\big(m(x)\big) - \mathrm{Cov}(m(x), D)\mathrm{Var}(D)^{-1}\mathrm{Cov}\big(D, m(x)\big).$$

The adjusted values are analogous to the posterior predictive distribution when using full Gaussian regression. The adjusted expectation is the emulator's prediction of the expected value of the reservoir model given a set of control parameters. The adjusted variance gives the uncertainty of that prediction accounting for both the spatial uncertainty due to interpolating between existing runs and due to the geological uncertainty.

Optimisation Using the Emulator

The goal is to find the values of the control parameters that maximise the objective function described earlier, that is

$$x_{max} = \underset{x}{\mathrm{argmax}}\, m(x),$$

and hence find the corresponding objective maximum $m_{max} = m(x_{max})$. The emulator constructed in the previous sections is used to accelerate this process by finding the optimal value of an *acquisition function* given the current emulator. This allows us to guide the placement of subsequent Refinement Runs. Once Refinement Runs have been executed the emulator is updated to incorporate the new run data and can be used to generate further Refinement Runs if required.

Many choices of acquisition function are available, with each acquisition function offering different characteristics and trade-offs between *exploitation* and *exploration. Exploitation* refers to selecting parameters where the emulator predicts a high objective and *exploration* means selecting parameters where the prediction uncertainty is high. The *Expected Improvement (EI)* acquisition function offers a balanced trade-off between exploitation and exploration (Shahriari, et al., 2016). A version of EI adapted for stochastic functions is used here.

Assume repetitions of the model have already been evaluated at an initial set of control parameter points. The current best sample mean output is denoted, $\bar{F}^+ \ = \bar{F}^r(x^+)$, where x^+ is the current best set of input parameters. The linear improvement function, $I(x)$, is given by

$$I(x) = \max\{0, m(x) - \bar{F}^+\}.$$

The *EI* criterion requires that the new parameter location, x, is chosen to maximise the expectation of $I(x)$, that is

$$x = \underset{x}{\mathrm{argmax}}\, \mathbb{E}[I(x)].$$

Assuming the uncertainty in the emulator $m(x)$ is normally distributed gives

$$m(x) \sim N\big(\mu(x), \sigma^2(x)\big), \qquad \text{where } \mu(x) = \mathrm{E_D}\big(m(x)\big) \text{ and } \sigma^2(x) = \mathrm{Var_D}\big(m(x)\big).$$

Rewriting $I(x)$ in terms of the random variable $z = \frac{m(x) - \mu(x)}{\sigma(x)}$, where $z \sim N(0,1)$ has a standard normal distribution, gives

$$I(x) = \max\{0, \sigma(x)z + \mu(x) \ - \bar{F}^{+}\}.$$

With some manipulation, the expectation of $I(x)$ can be shown to be

$$\mathbb{E}[I(x)] = \sigma(x)\phi(z^{*}) + (\mu(x) \ - \bar{F}^{+})\Phi(z^{*}),$$

Where $z^{*} = -z^{+} = (\mu(x) - \bar{F}^{+})/\sigma(x)$ and ϕ and Φ denote the probability distribution function for the standard normal and the cumulative distribution function for the standard normal respectively. The covariance matrix adaptation evolution strategy (CMA-ES) algorithm (Hansen & Ostermeier, 1996) is used at each step to find the value of x that maximises EI. The resulting value of x is used as the control parameters for the next set of runs.

Immediate application of EI requires that the emulator is updated after every run. This is a major limitation, as it does not allow runs to be executed concurrently. To avoid this, an *emulator believer* strategy is used (Ginsbourger, et al., 2008). This allows multiple Refinement Run control parameter values to be generated and executed concurrently between updates of the emulator, with the number chosen to fully utilise the available computational resources.

Practical application of the workflow

The workflow offers an assisted optimisation approach guided by the user instead of full automation. Figure 1 shows the major steps of the workflow. First the user defines the objective by combining weighted quantities of interest as described above. The initial repetition number, r, representing the number of runs that are repeated with the same control inputs values, must be selected. The appropriate number of repetition runs depends on the ratio between the size of the geological uncertainty and the impact the controls can have on the objective. For problems with relatively small uncertainty it can be as low as 2 or 3, while for problems with much greater uncertainty a value of 20 or more may be appropriate. In practice we have found that 5-10 works well for many realistic problems. The number of initial scoping runs may then be selected. The appropriate number depends on the number of control inputs and how rapidly varying the response is to these inputs. In practice we have found that $n = 2 * \mathit{number\ of\ control\ inputs} + 1$ is sufficient. This gives $n * r$ initial scoping runs. Runs can be executed in parallel and submitted to high performance computing (HPC) resources as needed to allow a high degree of run throughput and maximum utilisation of available resources.

The emulator is constructed using the resulting run data. The maximum expected value for the objective, and the corresponding prediction uncertainty, is determined using the current emulator. This value is the best prediction for the global optimum that the emulator can make given the current data and allows optimisation progress to be monitored. One or more batches of refinement runs are then launched to refine the emulator using the EI criteria described above. If the error in the estimate of the expected value calculated from the repetitions is too high to allow optimisation to proceed the repetition number may be increased for each batch. The use of an emulator believer strategy allows for more than one set of candidate control parameter values to be generated for each batch without subsequently updating the emulator. However, as this does not allow the emulator to be updated before generating each new set of values, we recommend that the size of each batch should be as small as possible while still allowing full utilisation of the available computational resources. The emulator is updated for the new run data and the maximum expected value for the objective over the control parameter ranges is recalculated. The user can then decide when optimisation has proceeded sufficiently. We suggest that for most problems this is when the prediction uncertainty of the current expected value for the objective maximum drops below the geological uncertainty for the model.

While this outlines the basic workflow, it is not prescriptive. If needed *verification* runs may be launched to add extra repetitions to existing groups of runs. For some problems, it may be appropriate to run a small number of runs with a very high repetition count to fully explore certain areas of control space before continuing. The focus is on understanding the problem rather than fully automating the optimisation process.

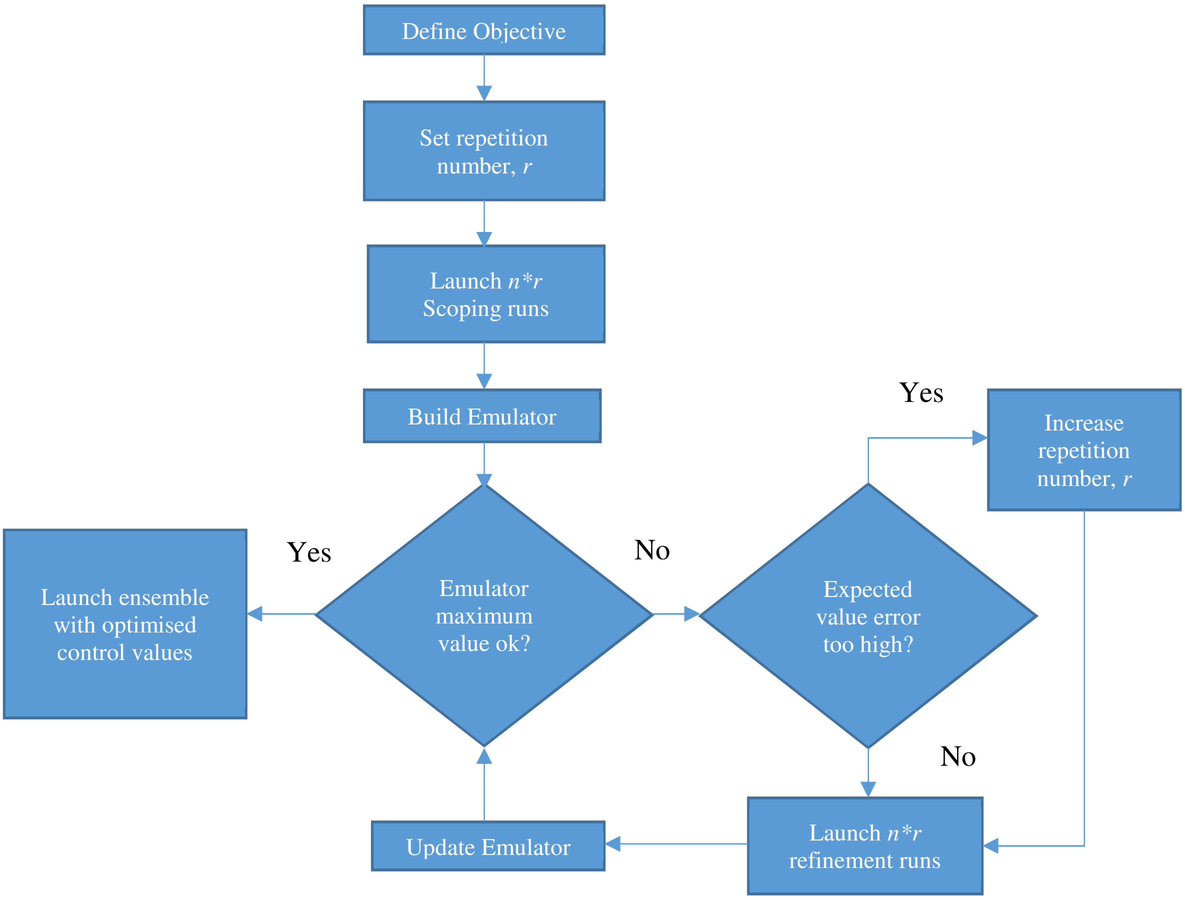
Define Objective
Set repetition number, r
Launch n*r Scoping runs
Build Emulator
Emulator maximum value ok?
Yes
Launch ensemble with optimised control values
No
Expected value error too high?
Yes
Increase repetition number, r
No
Launch n*r refinement runs
Update Emulator

Figure 1 Bayesian optimisation under uncertainty workflow. The repetition number, r, and the number of runs at each stage, n, and determining when optimisation has completed are decisions made by the user.

Examples

Several numerical examples are used to demonstrate the optimisation scheme. The first is an idealised 1D multimodal problem to help explain how the method works. The second is an adaptation of a relatively simple reservoir model that has previously been used to study history matching using emulators. The third is an adaptation of a more recent synthetic model of a real North Sea field with both static and dynamic uncertainties (Taha, et al., 2019).

One-dimensional optimisation problem with uncertainty

Many one-dimensional problems have previously been used to test optimisation routines. We adapt a multimodal example by adding multiplicative Gaussian noise to its output. A collection of similar examples, with varying forms of uncertainty, has been developed and is run regularly to benchmark the method described here. The function is given by

$$f(x) = (1 + y) * \, (x\sin(x) + x\cos(2x)), \;\; \text{where } y = Y \sim \mathrm{N}(0,1), \;\; x \in [0,10].$$

The parameter x represents the control to be optimised and Y is a random variable representing the geological uncertainty. The structure of the function results in different degrees of uncertainty for different values of x. Notably, the uncertainty is highest around local optima. We seek to find the value of x that maximises the mean of the f.

Figure 2 shows the steps the method takes to find the global optimum. The top plot in each panel shows the function being optimised and the emulator at the current iteration. The bottom plot in each panel shows the expected improvement given the current emulator. The number of repetitions is fixed at three. Three sets of control parameters (giving nine runs in total) are used initially to create the emulator. The iterations show how the emulator can account for the uncertainty in the estimates for the mean of the objective and how the expected improvement acquisition function balances exploration and exploitation Figure 3 shows the objective values of the runs chosen by the method during optimisation and the expected maximum value predicted by the emulator after each new batch. It allows optimisation progress to be tracked and gives an indication of when optimisation may be complete.

The last three iterations are of note as the method continually adds more runs around the current best optimum. This shows one of the major differences when using Bayesian optimisation on a function with or without uncertainty - without uncertainty the expected improvement would be zero at an already explored point and the method would explore other areas. In our example, the uncertainty in the mean estimate is still sufficiently high that the method calls for more runs in that area to try to reduce the uncertainty rather than explore more widely. If enough iterations are performed at that point the uncertainty in the mean estimate will reduce sufficiently that the method will continue to explore elsewhere. For higher dimensional problems this can take many iterations. If the user observes the method repeatedly launching runs close to a single set of control values, we recommend that a large set of repetitions are run at that point in order to refine the estimate of the expected value and reduce the uncertainty sufficiently for the method to continue to explore.

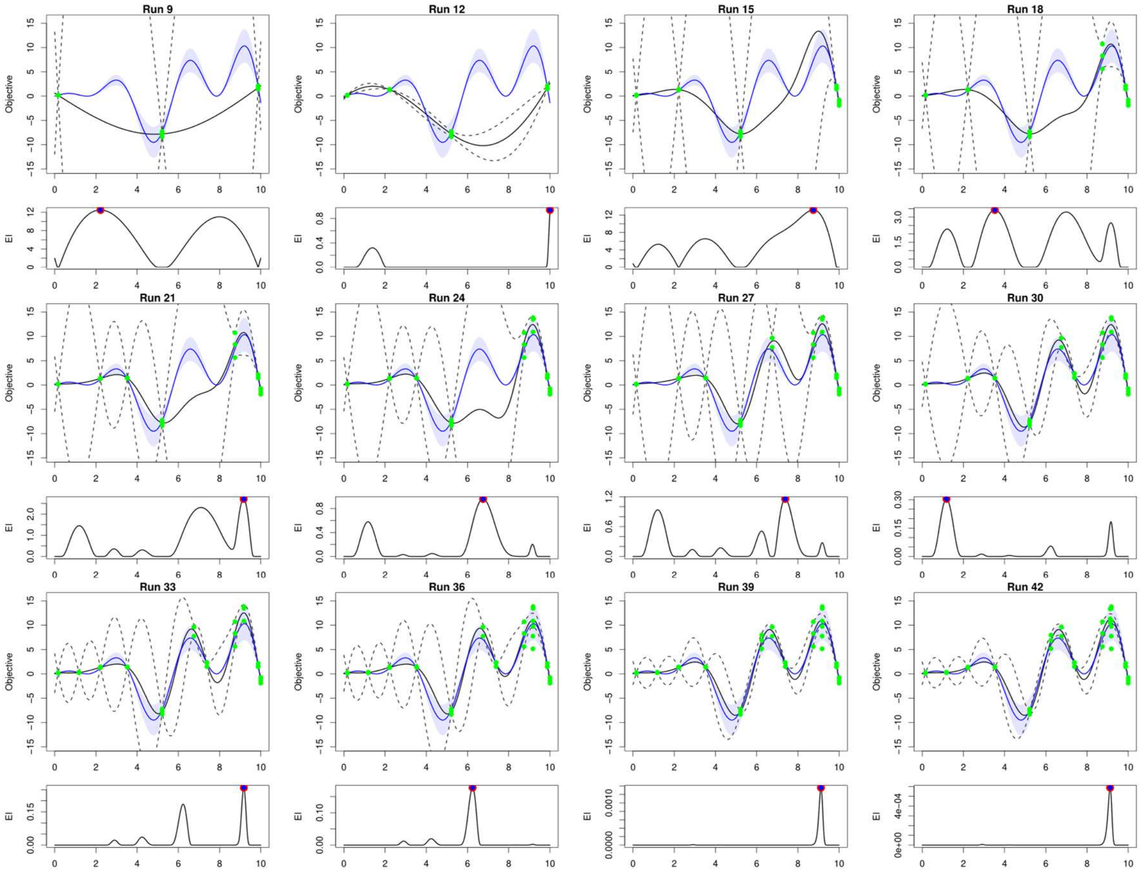

Figure 2 Optimisation of a 1D function with multiplicative Gaussian noise. Three repeated evaluations of the function are used at each of step with twelve iterations shown. Top plot in each panel shows the function and emulator at the current iteration. The thick blue line is the function mean and the shaded blue area the function mean $\pm\mathbf{2}$ *standard deviations. Green dots are current function evaluations. The thick black line is the emulator estimate of the mean and the dotted lines* $\pm\mathbf{2}$ *standard deviations showing the current emulator uncertainty of the mean estimate. The bottom plot shows the expected improvement given the current emulator. The black line is the expected improvement and the blue dot the maximum of the expected improvement. This maximum value is used to generate function evaluations in the subsequent panel.*

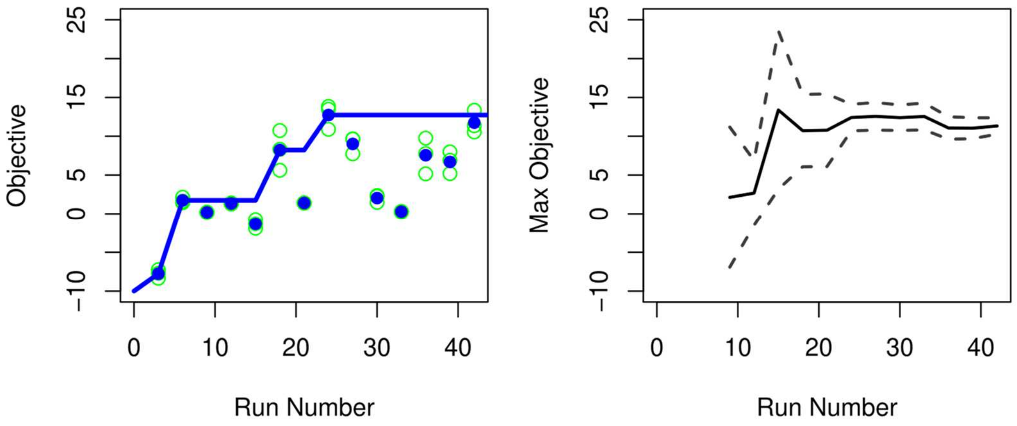

Figure 3 Left: Objective values for runs chosen by the method during optimisation of the one-dimensional problem. Green circles are individual runs, blue dots are sample means. Blue line is a cumulative maximum value for the sample means of the objective. Right: Evolution of the emulator's current expected maximum value for the 1D function with multiplicative Gaussian noise. The thick black line is the emulator expected maximum value of the mean of the function and the dotted lines $\pm\mathbf{2}$ *standard deviations showing the current emulator uncertainty of this estimate.*

Simple Reservoir Model Well Positioning

The second demonstration problem is a vertical well positioning problem is using a simple model with characteristics of reservoirs found on the UK continental shelf**.** The model was previously developed for use as a training tool for the use with emulation-based history matching. The grid spans 12000ft x 22000ft and a modest simulation grid with 12 x 22 x 10 grid cells is used. The initial state of the model is shown in Figure 4. The model features several faults and three layers with different rock properties. It is unknown if the faults are open or closed and this is represented by three fault transmissibility multipliers. The exact rock properties in each layer are also uncertain and subject to porosity and permeability multipliers in the horizontal directions. The geological uncertainties are shown in Table 1.

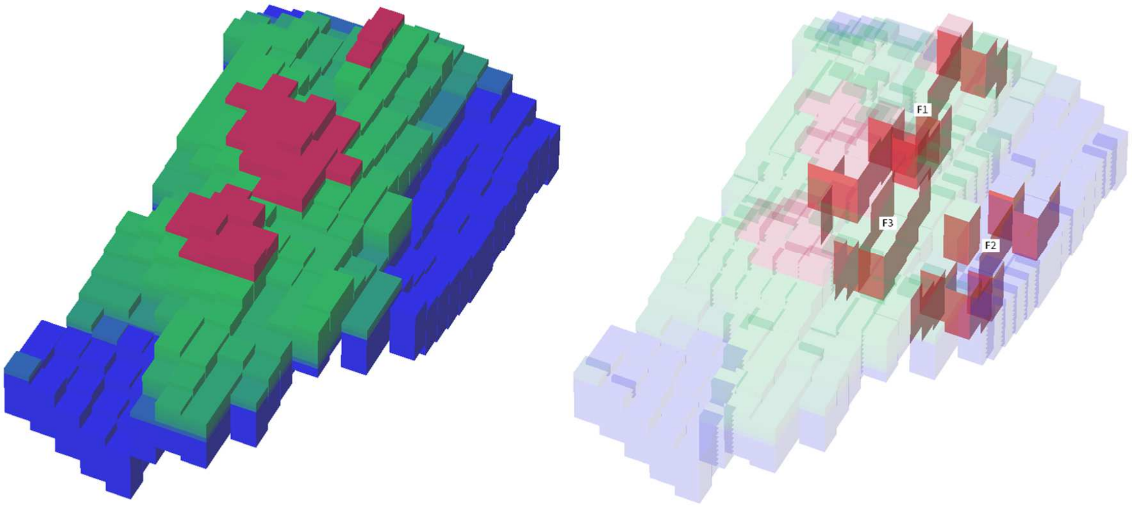

Figure 4 The simple reservoir model. Left: The Initial state of the simple reservoir model, Gas is shown in red, oil in green and water in blue. Right: The faults in the model in three groups.

Uncertainty Parameter	Most Likely Value	Lower Bound	Upper Bound	Distribution Type
Fault Transmissibility Multipliers				
FMUL_F1	0.5	0	1	Uniform
FMUL_F2	0.5	0	1	Uniform
FMUL_F3	0.5	0	1	Uniform
Permeability Multipliers				
Kx_mult1	4.5	3	6	Triangle
Kx_mult2	4.5	3	6	Triangle
Kx_mult3	4.5	3	6	Triangle
Porosity Multipliers				
Phi1	1.0	0.8	1.2	Gaussian truncated at 2 std. dev.
Phi2	1.0	0.8	1.2	Gaussian truncated at 2 std. dev.
Phi3	1.0	0.8	1.2	Gaussian truncated at 2 std. dev.

Table 1 Uncertainty distributions for well positioning example

An aquifer provides good pressure support from underneath the reservoir and the reservoir is expected to achieve high oil recovery without the need for drilling injector wells. Three vertical producer wells are added to the model and the method described here is used to select optimal values for the X and Y locations of each well. Wells are completed throughout the reservoir. Bottom hole pressures are held constant at each producer. The model is run for three years and the objective function is defined as

$$m(x) = \mathbb{E}(total\ oil\ production) - 0.2 * \mathbb{E}(total\ water\ production)$$

The optimisation workflow was applied to the problem. An increasing number of repetitions scheme was used, with 4 repetitions for runs 1-100, 6 repetitions for runs 101-232 and 12 repetitions for runs 232-304. This allowed the method to rule out large areas where it was inappropriate to position a well with a minimal number of runs and then refine the optimum with a larger number of runs. Finally, three large ensembles of 50 runs were executed at optimal positions predicted by the emulator. Figure 5 shows the objective values of the runs chosen during optimisation and the expected maximum value predicted by the emulator after each new batch.

Figure 6 shows the best well locations obtained by the method and a prediction ensemble showing the total oil production for this well configuration with a large sample of the geological uncertainty. The prediction plot shows how large the geological uncertainty is for this problem compared to the ability of the controls to increase the objective. A consequence of this is that just choosing a single well configuration, as shown in Figure 6, may not be appropriate and that many well configurations exist that may be considered optimal when allowing for the envelope of the geological uncertainty.

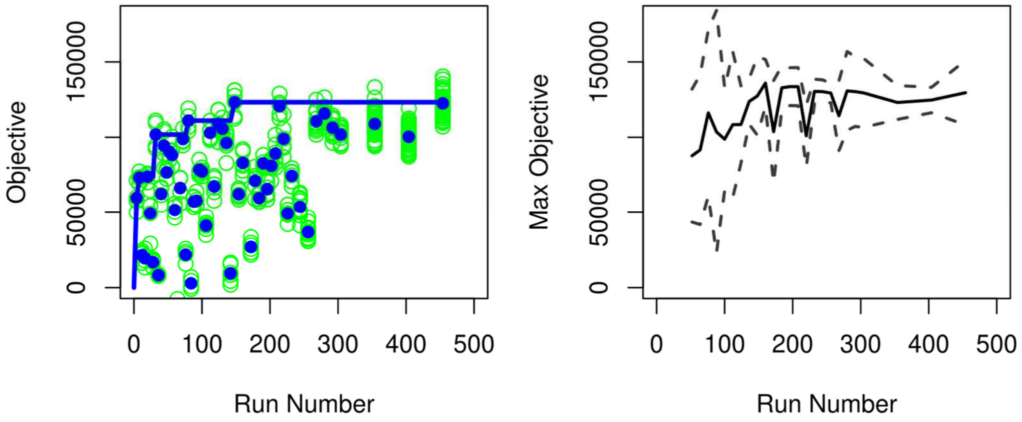

Figure 5 Left: Objective values for runs chosen by the method during optimisation of the simple reservoir model problem. Green circles are individual runs, blue dots are sample means. Blue line is a cumulative maximum value for the sample means of the objective. Right: Evolution of the emulator's current expected maximum value for the simple reservoir model. The thick black line is the emulator expected maximum value of the mean of the function and the dotted lines ±**2** *standard deviations showing the current emulator uncertainty of this estimate.*

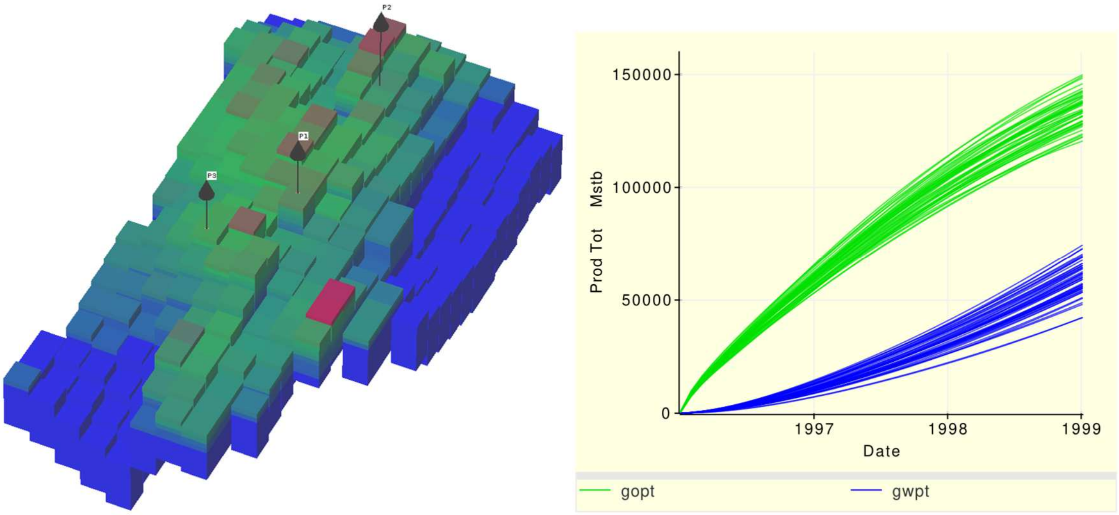

Figure 6 Left: Final optimised well positions and distribution of fluids after three years of production. Right: 50 member prediction ensemble using the best single well locations obtained during the optimisation process and sampling the full range of uncertainty. Total oil production in green, total water production in blue.

Synthetic North Sea Reservoir Model

The model has been described in detail in (Taha, et al., 2019), where it was used to demonstrate history matching with 4D seismic data. In brief, the model is a state-of-the-art synthetic reservoir model based on real North Sea fields. The model has uncertainties in the full geology to simulation workflow. These include in the PVT, structure (faults and horizons), facies, relative permeability and water saturation. Though, for this example, only structural and facies uncertainties are used. Notably, this results in a complete new geological model being created with each run. The initial state of the model is shown in Figure 7.

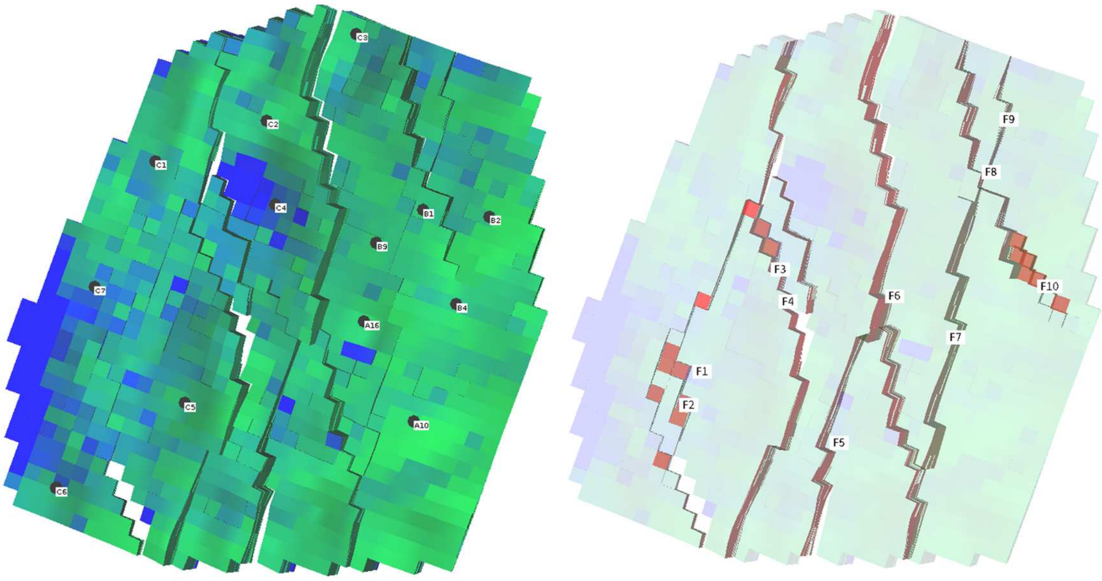

Figure 7 The full featured synthetic reservoir model. Left: The Initial state of the model with existing wells, Gas is shown in red, oil in green and water in blue. Right: The faults in the model

The method was used to optimise the location of a new vertical injector well. The vertical locations were defined using two controls, representing the horizontal and vertical directions. The parameter ranges for these controls defined a square region. To allow for the irregular geometry of the model and the constantly changing grid, a self-organising Kohonen map (Kohonen, 1982) was used to map between these values and the true location in the model. All geological uncertainty parameters had their ranges set equivalent to those found during history matching in (Taha, et al., 2019). The model was run for 7 years of production and the same objective function was used as for the simple reservoir model problem. The optimisation workflow was applied to the problem. 10 repetition runs were used for each set of candidate control parameter values. The relatively high number of repetitions was chosen due to the high level of geological uncertainty and relatively small impact of drilling a single injector in an already developed field.

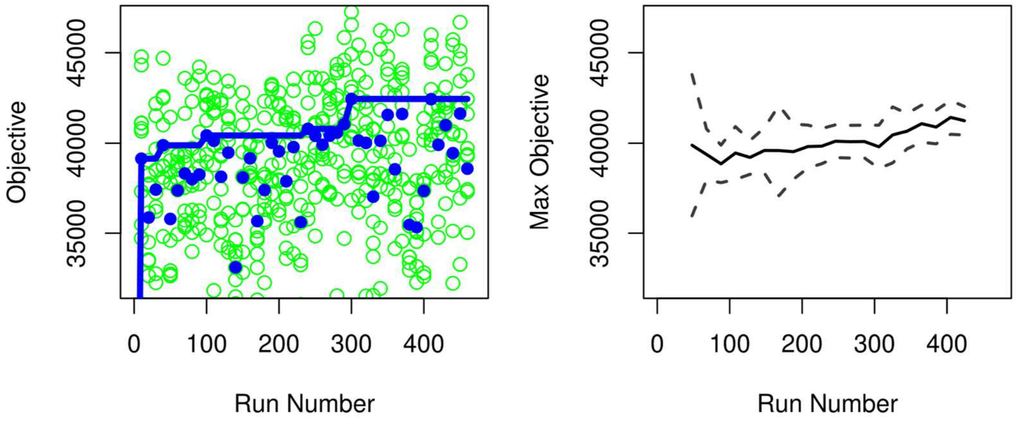

Figure 8 Left: Objective values for runs chosen by the method during optimisation of the synthetic North Sea reservoir. Green dots are individual runs, blue are sample means. Blue line is a cumulative maximum value for the sample means of the objective. Right: Evolution of the emulator's current expected maximum value. The thick black line is the emulator expected maximum value of the mean of the function and the dotted lines $\pm\mathbf{2}$ *standard deviations showing the current emulator uncertainty of this estimate.*

Figure 8 shows the objective values of the runs chosen by the method during optimisation and the expected maximum value predicted by the emulator after each new batch. As well as showing optimisation progress, the figure highlights the high degree of geological uncertainty relative to the impact of drilling a single new well. Figure 9 shows the final emulator’s prediction of the impact of drilling the new well at every point in the model and demonstrates the impact on the objective value of drilling a new well at the location determined by the method verses taking no further action using a large ensemble of runs. The emulator predictions plot highlights the difficulty of placing a new well in this model. The plot shows several local maxima, notably with one in the far top right corner that has a large basin of attraction. The method is able to quickly explore the maxima before further refining around the true maximum. Gradient based methods could struggle with local maxima such as these.

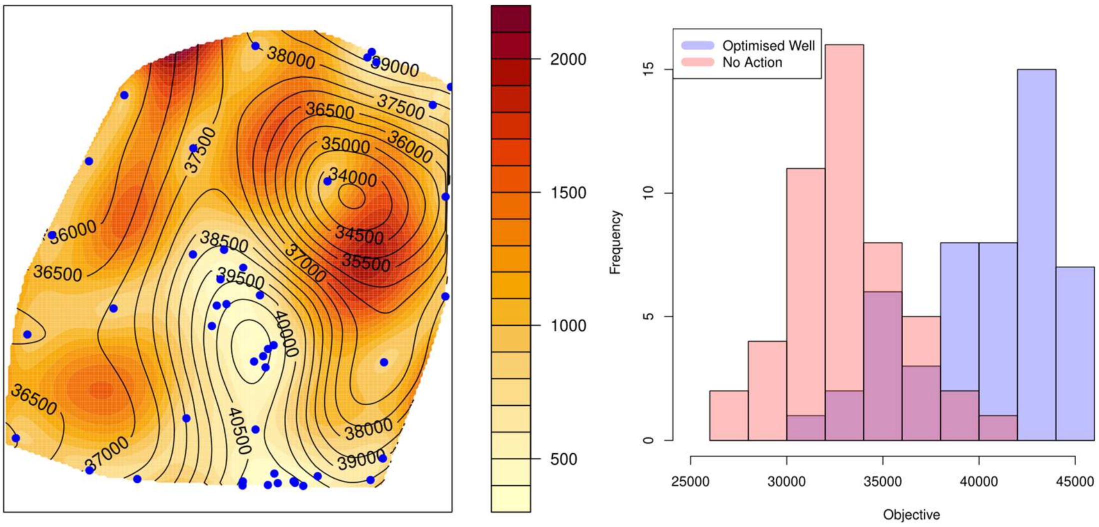

Figure 9 Left: The emulator's belief about the optimal well locations at the end of the optimisation process. Contour lines show the predicted objective value, the heat map shows the emulator uncertainty (2 standard deviations) in the predicted objective value and the blue dots are the control locations that were selected and tested by the method. Right: Comparison of the predicted objective value with no action taken with drilling of an injector well at the optimal location predicted by the method. Histograms are overlaid and translucent to show bins that would otherwise be hidden. They were calculated using a 50 member ensemble of runs to sample the geological uncertainty.

Conclusions

We have presented a Bayesian optimisation workflow applicable to reservoir simulation models with geological uncertainty. The workflow is underpinned by a Bayes Linear statistical emulator that contains extra terms allowing it to quantify the effect of geological uncertainty. The methodology has been demonstrated using both an idealised test problem and a pair of synthetic test cases and has been shown to be an effective optimiser requiring comparatively few runs, particularly when compared to ensemble optimisation techniques.

In comparison to other methods for optimisation under uncertainty, the workflow does not rely on a fixed set of pre-generated geological realisations. Instead, the method constantly generates new samples from a probability distribution representing the geological uncertainty. This gives flexibility to scale the number of repeated runs with common control inputs as needed. Early in the optimisation process, few repetitions may be used with the number increased as needed when the approximate location of the optimum values has been located. This leads to a scheme that can use comparatively few runs while also allowing for a full analysis of the impact of geological uncertainty.

In common with standard Bayesian optimisation techniques, the method is particularly suitable for use with computationally expensive reservoir models where the number of available runs will be limited. The method can quickly and efficiently explore multiple local optima before locating the true global optimum. However, again in common with other Bayesian optimisation techniques, it is not the most efficient method to refine parameter values that are close to the global optimum. We argue that in models with large geological uncertainty this process is not appropriate – once the potential improvement to the optimal value falls below the size of the impact of the geological uncertainty further refinement may not be meaningful. However, in practice, doing such local refinement may be required. In this case, we advocate applying a gradient based optimisation scheme using the output of the method described here as a starting point.

There are various technical improvements that can be made to this work. We have limited ourselves to drawing simple random samples from the probability distribution that defines the geological uncertainty. While straightforward, it would be possible to draw samples more efficiently using a stratified sampling scheme. This would result in smaller number of repetitions being required. We have not developed formal stopping criteria for our method, instead relying on diagnostic plots to allow the user to judge when optimisation is complete. Similarly, we have not developed automated tools to test when the number of repetition runs should be increased or decreased, again offering diagnostic tools to allow the user to make an informed decision. Our method goes some way towards solving the sequential design run placement problem using Expected Improvement and offers a concurrent sequential design technique to allow the exploitation of HPC resources using an Emulator believer strategy. However, many acquisition functions are available in the literature and it would be useful to investigate their performance on these problems. While the emulator believer strategy allows for concurrent runs it remains a one-step look ahead approach. Developing a full multi-step sequential design solution for the method is an important, if challenging, area of future improvement.

There are also extensions to the methodology. We have focused on optimising the expected value of the quantities of interest. Robust optimisation schemes require optimisation of a risk tolerant measure of the quantity of interest This may mean a percentile, such as P10, or simultaneously maximising the expected value while minimising the variance. As already discussed, the framework presented here can be extended to handle this requirement through emulating the variance surface. While we have a flexible method of defining the optimisation objective from multiple quantities of interest, we have not considered true multi-objective optimisation in which the trade-off between optimising different quantities of interest can be analysed by calculating a Pareto front. The use of a statistical emulator in the Bayesian optimisation approach should allow rapid calculation of a Pareto front. However, further work will be needed to investigate this.

Finally, we have explicitly framed our method as a solution to the optimisation under uncertainty problem. This was motivated by current research in this area. However, to allow for rational decision making, the objective function would properly be expressed as a Utility function that incorporates the decision maker's risk tolerance (Berger, 2010). Rather than determining a best utility value with single values for the control parameters, it is then desirable to offer a set or range of control parameter values that can be considered optimal given the utility and the geological uncertainty. The flexible method of defining objective functions described here could be extended to allow definition of utility functions. Further, the use of emulators allows the space of optimal utility values to be explored and visualised after optimisation is complete. This was briefly discussed in our second example and demonstrated in our third example. However, more work would be required to determine how to extend our optimisation under uncertainty tools to encompass full decision making under uncertainty.

Despite the many avenues of future work, the workflow described here has shown promising results on the test problems we have presented. Application to more complex cases and different types of optimisation problems is ongoing and will give more insights about the potential of the method.

Author Contributions

Designed method: RB, JRH, IV, RH. Developed software: RB, JRH, GP, PW, RH. Developed models: RB, MAJ, TT. Tested method & analysed results: RB, JRH, MAJ. Wrote manuscript: RB, MAJ. Critically reviewed manuscript: All.

Acknowledgements

The authors would like to thank Emerson Exploration and Production Software for permission to publish this work.

References

Aarnes, I., Midtveit, K. & Skorstad, A., 2015. Evergreen workflows that capture uncertainty – the benefits of an unlocked structure. *First Break,* 33(10), pp. 89 - 92.
Alrashdi, Z. & Sayyafzadeh, M., 2019. (μ+λ) Evolution strategy algorithm in well placement, trajectory, control and joint optimisation. *Journal of Petroleum Science and Engineering,* Volume 177, pp. 1042-1058.
Andrianakis, I. et al., 2016. History matching of a complex epidemiological model of human immunodeficiency virus transmission by using variance emulation. *Journal of the Royal Statistical Society: Series C (Applied Statistics),* Volume 66.
Andrianakis, I. et al., 2015. Bayesian History Matching of Complex Infectious Disease Models Using Emulation: A Tutorial and a Case Study on HIV in Uganda. *PLOS Computational Biology,* 1, Volume 11, pp. 1-18.
Ankenman, B., Nelson, B. L. & Staum, J., 2010. Stochastic Kriging for Simulation Metamodeling. *Operations Research,* Volume 58, pp. 371-382.
Artus, V., Durlofsky, L. J., Onwunalu, J. & Aziz, K., 2006. Optimization of nonconventional wells under uncertainty using statistical proxies. *Computational Geosciences,* Volume 10, p. 389–404.
Aslam, U. & Bordas, R., 2020. An Ensemble-Based History Matching Approach for Reliable Production Forecasting from Shale Reservoirs. *Accepted to Unconventional Resources Techology Conference.*
Berger, J. O., 2010. *Statistical decision theory and Bayesian analysis.* 2nd ed. New York: Springer.
Chen, Y., Oliver, D. S. & Zhang, D., 2009. Efficient Ensemble-Based Closed-Loop Production Optimization. *SPE Journal,* Volume 14, p. 634–645.
Ciaurri, D. E., Mukerji, T. & Durlofsky, L. J., 2011. Derivative-Free Optimization for Oil Field Operations. In: X. Yang & S. Koziel, eds. *Computational Optimization and Applications in Engineering and Industry.* Berlin: Springer, p. 19–55.
Emerick, A. et al., 2009. *Well Placement Optimization Using a Genetic Algorithm With Nonlinear Constraints.* SPE Reservoir Simulation Symposium, SPE-118808-MS.
Evensen, G., 2003. The Ensemble Kalman Filter: theoretical formulation and practical implementation. *Ocean Dynamics,* Volume 53, p. 343–367.
Fillacier, S. et al., 2014. *Calculating Prediction Uncertainty using Posterior Ensembles Generated from Proxy Models.* SPE Russian Oil and Gas Exploration & Production Technical Conference and Exhibition, SPE-171237-MS, p. 11.
Fonseca, R. M. et al., 2018. *Overview of the olympus field development optimization challenge.* 16th European Conference on the Mathematics of Oil Recovery, ECMOR 2018, EAGE.
Fonseca, R. R.-M., Chen, B., Jansen, J. D. & Reynolds, A., 2017. A Stochastic Simplex Approximate Gradient (StoSAG) for optimization under uncertainty. *International Journal for Numerical Methods in Engineering,* Volume 109, pp. 1756-1776.
Ginsbourger, D., Le Riche, R. & Carraro, L., 2008. A Multi-points Criterion for Deterministic Parallel Global Optimization based on Kriging. *HAL,* 3.
Goldstein, M., 2012. *Bayes Linear Analysis for Complex Physical Systems Modeled by Computer Simulators.* Berlin, Springer, p. 78–94.
Goldstein, M. & Wooff, D., 2007. *Bayes Linear Statistics: Theory and Methods.* Chichester: John Wiley & Sons Ltd.
Goodwin, N. H., 2018. *Efficient Large Scale Optimization Under Uncertainty With Multiple Proxy Models.* EAGE/TNO Workshop on OLYMPUS Field Development Optimization, European Association of Geoscientists & Engineers.
Hansen, N. & Ostermeier, A., 1996. *Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation.* Proceedings of IEEE International Conference on Evolutionary Computation, IEEE, p. 312–317.
Jansen, J. D., 2011. Adjoint-based optimization of multi-phase flow through porous media – A review. *Computers & Fluids,* Volume 46, pp. 40-51.
Kohonen, T., 1982. Self-organized formation of topologically correct feature maps. *Biological Cybernetics,* Volume 43, p. 59–69.

Lorentzen, R. J., Berg, A., Naevdal, G. & Vefring, E. H., 2006. *A New Approach For Dynamic Optimization Of Water Flooding Problems.* Intelligent Energy Conference and Exhibition, SPE-99690-MS.

Peters, L. et al., 2010. Results of the Brugge Benchmark Study for Flooding Optimization and History Matching. *SPE Reservoir Evaluation & Engineering,* Volume 13, p. 391–405.

Schulte, D. O. et al., 2020. Multi-objective optimization under uncertainty of geothermal reservoirs using experimental design-based proxy models. *Geothermics,* Volume 86, p. 101792.

Schwarz, G., 1978. Estimating the Dimension of a Model. *Ann. Statist.,* Volume 6, p. 461–464.

Shahriari, B. et al., 2016. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE,* 1, Volume 104, pp. 148-175.

Stein, M., 1987. Large Sample Properties of Simulations Using Latin Hypercube Sampling. *Technometrics,* 5, Volume 29, p. 143–151.

Taha, T. et al., 2019. *History Matching Using 4D Seismic in an Integrated Multi-Disciplinary Automated Workflow.* SPE Reservoir Characterisation and Simulation Conference and Exhibition, SPE-196680-MS.