Efficient Optimization with Gray-box PDE Simulations

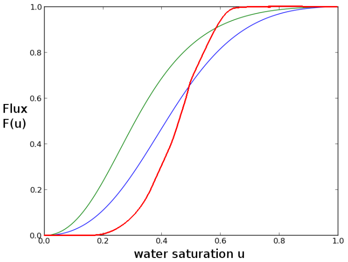
November 19, 2014

What is gray-box PDE simulation?

Consider an example PDE simulating 1-D water-oil two phase flow:

$$\frac{\partial u}{\partial t} + \frac{\partial F(u)}{\partial x} = 0 \,,$$

with known boundary and initial conditions.



$$\text{Gray-box} \begin{cases} \text{PDE (for example, F(u) can be unkown.)} \\ \text{Numerical implementation} \end{cases}$$

PDE simulations can be gray-box

In many cases the general form of the PDE is known, but the specific model choices and/or numerical scheme choices are not accessible (for example, for many commercial and legacy codes).

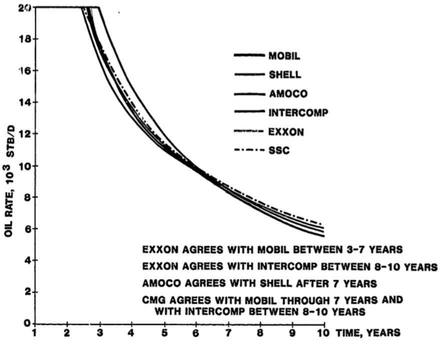


Fig. 3 – Case 1 – oil rate vs. time.

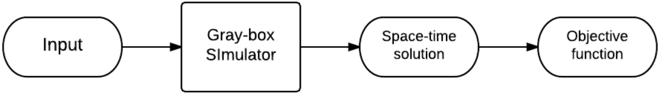
[Akand W. Islam et al., A review on SPE's comparative solution projects, 2013]

Research objective

Design an efficient method for the optimization of gray-box simulations.

Optimization based on gray-box simulation requires non-intrusive methods

- My research considers optimization based on gray-box simulations.
- A gray-box PDE simulation maps a set of input to a space-time solution. An objective function is then computed from the solution.



- Optimization based on gray-box simulators requires non-intrusive methods.

Conventional non-intrusive methods can be expensive

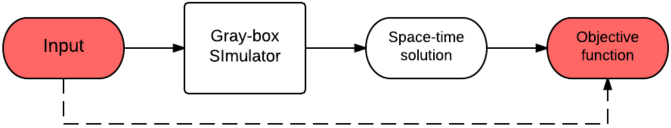
Gray-box simulations can not provide the objective function's gradient.

- In conventional non-intrusive methods, each simulation provides only one sample of the objective function value.
- When the number of inputs increases, more simulations are required (known as the curse of dimensionality).

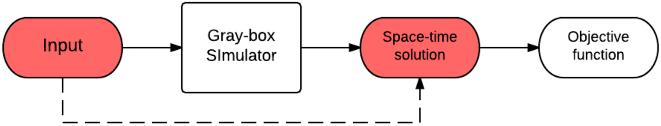
The challenges above motivates my research.

A different view of gray-box simulation

Conventional non-intrusive optimization methods view each simulation as a map from the input to the objective function, discarding the information contained in the space-time solution.

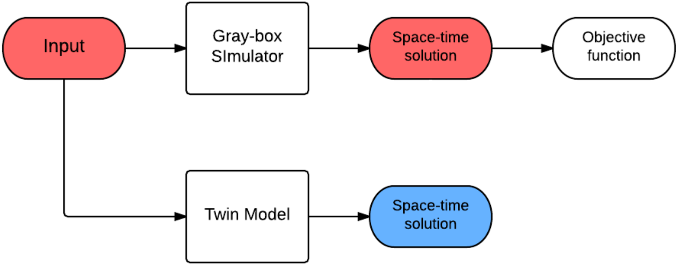


Instead, our method views the simulation as a map from the input to the space-time solution.



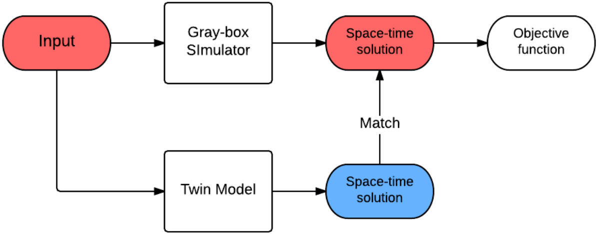
How to use the space-time solution?

We propose a PDE-based surrogate simulator.
The PDE governing surrogate simulator is parameterized. Given the same input, the surrogate and the gray-box simulator both generate a space-time solution.



We call this surrogate simulator a *twin model*.

Twin model calibrates itself to minimize the mismatch of solutions



Given the same inputs, twin model calibrate its PDE's parameterization to minimize the mismatch between the space-time solutions.

Example

Gray-box model

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} F(u) = J(t,x)$$

with known initial and boundary conditions.
$u = u(t,x)$ is water saturation. $J(t,x)$ is the control (water injection). J determines the injection rate. $F(u)$ is an unkown flux function.

Twin model

$$\frac{\partial \hat{u}}{\partial t} + \frac{\partial}{\partial x} \left(\sum_i c_i \phi_i(\hat{u}) \right) = J(t,x)$$

with the same initial and boundary conditions. $\phi_i(u)$'s are the basis functions for parameterization. c_i's are the parameterization coefficients.

Both models share the same objective function $L(u)$.
c_i's are calibrated to minimize $|u - \hat{u}|$ (with appropriate norm).

What is the advantage of twin model?

Easier target to infer

- Conventional non-intrusive methods infer the relations $J(t, x) \rightarrow L(u)$.
- Twin model infers the unkown functions inside the PDE, i.e. the function form of $F(u)$, or c_i's.

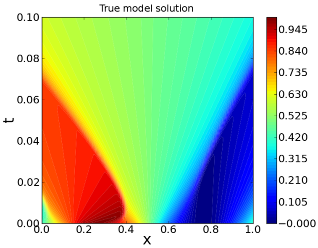
More training information

- Conventional non-intrusive methods treat each gray-box simulation as just one sample of L.
- In the twin model approach, the whole space-time solution $u(t, x)$ from the gray-box simulation are used for inference purpose.

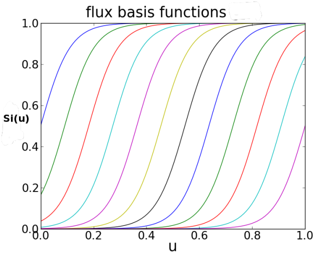
Numerical test: problem setup

$$\frac{\partial u}{\partial t} + \frac{\partial F(u)}{\partial x} = J\,, \qquad \frac{\partial \hat{u}}{\partial t} + \frac{\partial \hat{F}(\hat{u})}{\partial x} = J$$

The control J is a constant independent of t and x. $F(u)$ is an unknown 6-order Chebyshev polynomial. Neumann boundary condition is used. The two simulations use the same initial condition.



$u(t,x)$ at $J=0$.

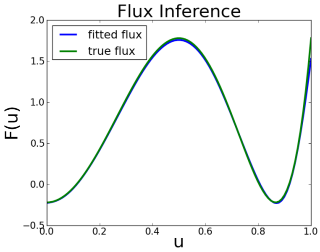


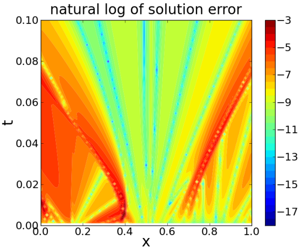
Basis $\phi(x)$.

Numerical test: infer the twin model

We only sample the gray-box simulation at $J = 0$, and use its $u(t, x)$ to infer $\hat{F}$ by

$$\min_{\hat{F}} \int_x \int_t |u - \hat{u}|^2 \, dtdt \qquad [1]$$

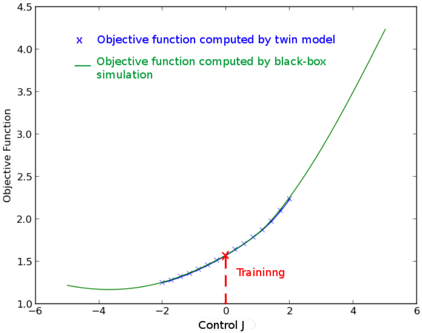




[1] Because the solutions u and $\hat{u}$ are discretized, in reality we minimize $\sum_{i,j} \left| u(t_i, x_j) - \hat{u}(t_i, x_j) \right|^2$. For simplicity we assume the twin model and the gray-box simulation share the same space-time grid.

Numerical test: using twin model for different control

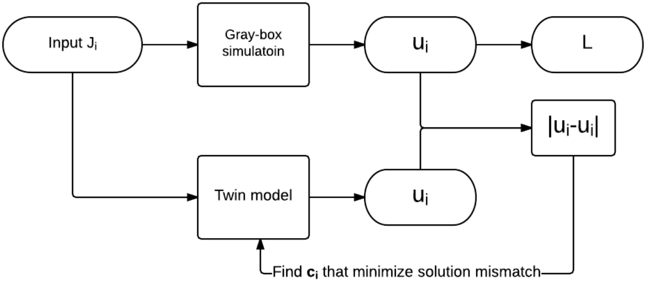
The twin model is only trained for $J = 0$. But if $F(\cdot)$ is inferred accurately, the twin model can approximate the gray-box simulation at different J. Also the twin model can provide an approximate gradient for the gray-box simulation.



A vanilla optimization using twin model

While not converged:

Step 1: evaluate the gray-box simulation at J_i and train the twin model



Step 2: Compute dL/dJ from the twin model (fixing c_i)

Step 3: Line search based on the current twin model's dL/dJ and obtain a new trial input J_{i+1}

The vanilla approach is not ideal

The drawbacks of the vanilla approach are:

- When the twin model is updated to its latest x, all previous twin model training and its gradients are discarded.
- No guarantee for convergence.

An ideal approach should be able to

- Consider all previously trained twin models and their gradients.
- Gradually improve the overal approximation quality when the number of twin model fittings increases.
- Judge the quality of the twin model. When the twin model is good, our optimization shall be close to a gradient driven optimization; otherwise, it shall be close to a gradient free optimization.

Gaussian processes modeling

1. We model the gray-box simulation's mapping from J to L as a realization of a Gaussian process $\mathcal{N}(\bar{L}, cov_1(\cdot,\cdot))$, where $\bar{L}$ is the existing samples' average.
2. The gradient of twin model is modeled as

$$g(J) = \nabla L(J) + \epsilon(J)\,,$$

where $\epsilon(J)$ is an unkown realization of a Gaussian process $\mathcal{N}(0, cov_2(\cdot,\cdot))$.

Assume

$$cov(\nabla L, \epsilon) = 0\,, \quad cov(L, \epsilon) = 0$$

For simplicity, we also assume the covariances have square-exponential kernels.

Joint distribution of gray-box simulation and twin model's gradient

$$
\begin{pmatrix} L \\ g \end{pmatrix} = \mathcal{N}\left(\begin{pmatrix} \bar{L} \\ 0 \end{pmatrix}, \Sigma = \begin{pmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} + E \end{pmatrix} \right),
$$

where

$$
\begin{aligned}
A_{11}: &\quad cov_1(L(J_1), L(J_2)) = \xi_1^2 \exp\left\{ -\frac{(J_1 - J_2)^2}{2\sigma_1^2} \right\} \\
A_{12}: &\quad cov(L(J_1), g(J_2)) = \frac{\xi_1^2}{\sigma_1^2}(J_1 - J_2) \exp\left\{ -\frac{(J_1 - J_2)^2}{2\sigma_1^2} \right\} \\
A_{22}: &\quad cov(\nabla L(J_1), \nabla L(J_2)) = \frac{\xi_1^2}{\sigma_1^2} \exp\left\{ -\frac{(J_1 - J_2)^2}{2\sigma_1^2} \right\} \left(\boldsymbol{I} - \frac{1}{\sigma_1^2}(J_1 - J_2)(J_1 - J_2)^T \right) \\
E: &\quad cov_2(\epsilon(J_1), \epsilon(J_2)) = \xi_2^2 \boldsymbol{I} \exp\left\{ -\frac{(J_1 - J_2)^2}{2\sigma_2^2} \right\}
\end{aligned}
$$

Posterior distribution of $L(J)$

Dataset

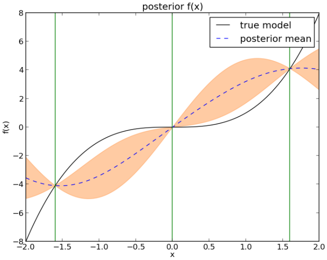
1. The gray-box simulations on sample points $D = \{J_1, \cdots, J_m\}$.
2. A twin model is trained on $\forall J_i \in D$. We also compute an approximate gradient $g(J_i)$ on $\forall J_i \in D$.

We can obtain

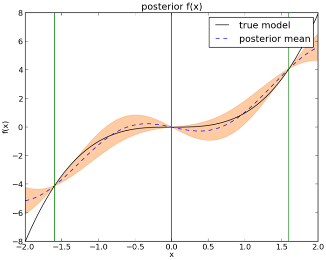
$$p[L(J')|L_D, g_D] \quad \text{for } \forall J'$$

Have an approximate gradient can improve the posterior estimate

Consider an example: $L(J) = J^3$.



Posterior with no gradient information.



Posterior with noisy gradient information.

Bayesian optimization

Bayesian optimization is a way to dictate the next trial point J_i from the posterior of $L(J)$.

Acquisition function

- Define an *acquisition function* $\alpha : p(J) \to \mathbb{R}$, which maps the posterior distribution to a scalar.
- One popular choice is the LCB (lower-confidence bound) acquisition function.
$$\alpha(J) = \mu(J) - \kappa\sigma(J)$$
where μ and σ are the posterior mean and standard deviation, κ is a user-defined constant. Also $\frac{\alpha(J)}{\partial J}$ can be computed.

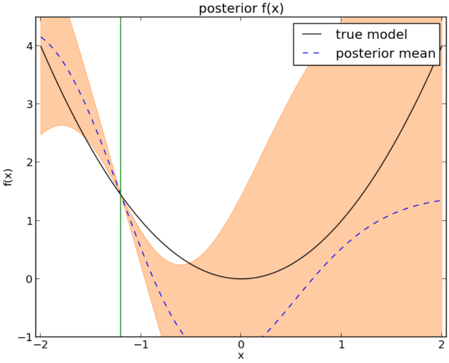
1. We use gradient-driven optimization method (e.g. SLSQP) to minimize $\alpha(J)$. The minimizer J^* dictates the next point to sample the gray-box simulation and to train the twin model.
2. Update D and the posterior distribution.

Bayesian optimization procedure

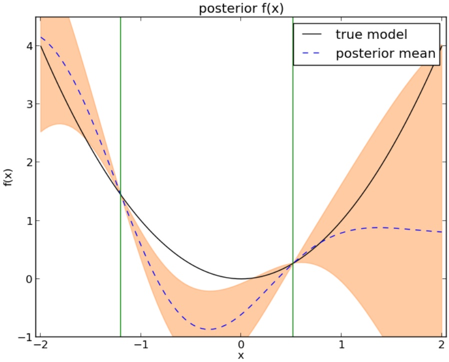
Algorithm 1: Optimization with Twin Model

Initial sample point J_0;
$J^* = J_0$;
$J_{best} = J^*$;
while *not converged* **do**
  Evaluate gray-box simulation at J^*, obtain $u(t, x; J^*)$ and $L(J^*)$;
  Train twin model from $u(t, x; J^*)$;
  Evaluate $\left.\frac{\partial \hat{L}}{\partial J}\right|_{J^*}$ of twin model;
  Update the posterior of $L(J)$ using $u(J^*)$ and $\left.\frac{\partial \hat{L}}{\partial J}\right|_{J^*}$;
  Construct the acquisition function $\alpha(J)$ and its gradient $\frac{d\alpha}{dJ}$ using the posterior;
  Minimize $\alpha(J)$ by a gradient-driven method. $J^* \leftarrow \operatorname{argmin}_J \alpha(J)$;
  $J_{best} \leftarrow \operatorname{argmin}_{J_{best}, J^*} L(J)$;

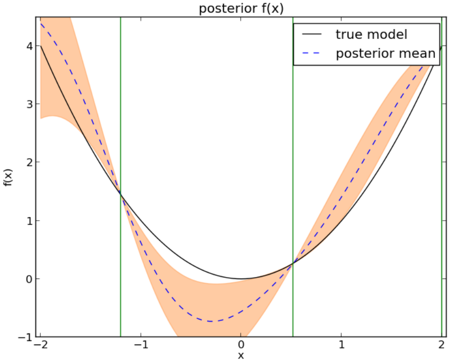
Example: Bayesian optimization of x^2



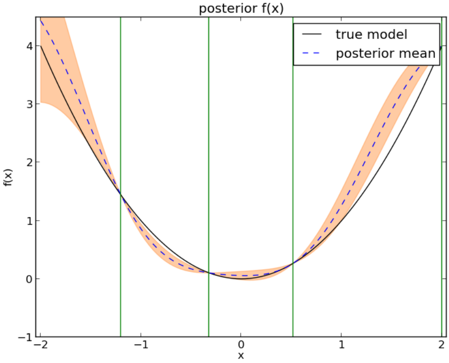
Example: Bayesian optimization of x^2



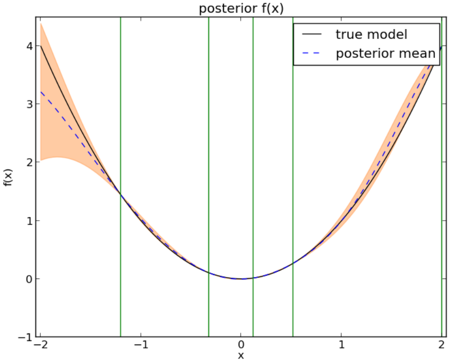
Example: Bayesian optimization of x^2



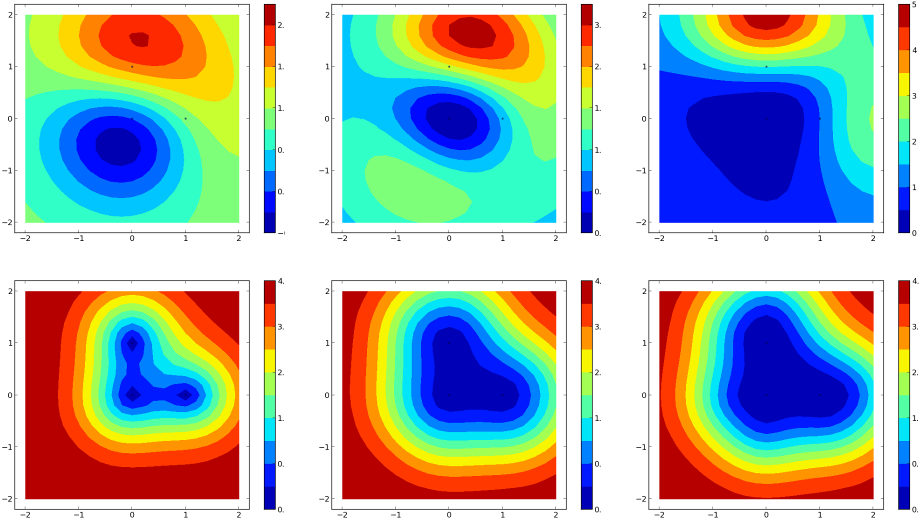
Example: Bayesian optimization of x^2



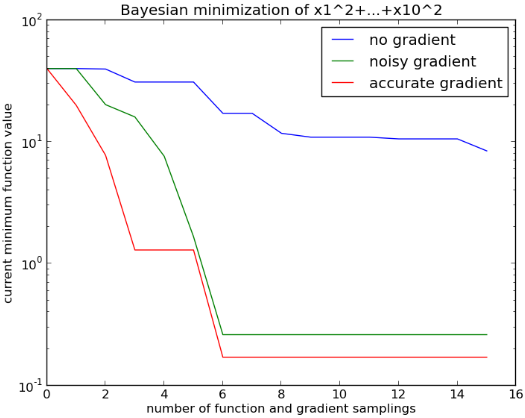
Example: Bayesian optimization of x^2



Example: $x^2 + 2y^2$, posterior mean and standard deviation



Example: minimize $x_1^2 + \cdots + x_{10}^2$



Summary

- Twin model has adjoint ability, and can provide approximate gradient.
- The posterior of the objective function is constructed using both the objective function value and the approximate gradient.
- When the input dimension is high, having the approximate gradient information can be important for obtaining better posterior.
- Bayesian optimization can be used for optimization with twin model.

Immediate next steps

- Estimate the covariance kernel parameters $\xi_1, \xi_2, \sigma_1, \sigma_2$ by maximum likelihood.
- Embed Bayesian optimization into a trust-region framework. Try to prove convergence.
- Showcase the Bayesian optimization method on a 1D twin model example.
- Thesis proposal defense.