An adjoint-based optimization method using the solution of gray-box conservation laws

Han Chen

Committee: Qiqi Wang, Karen Willcox, Youssef Marzouk
Thesis readers: Hector Klie, Paul Constantine

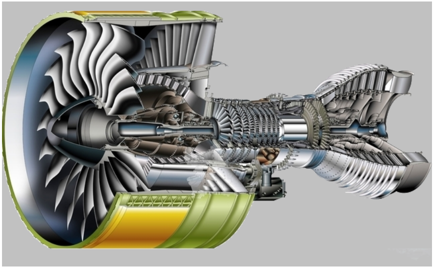
Mar 17, 2016

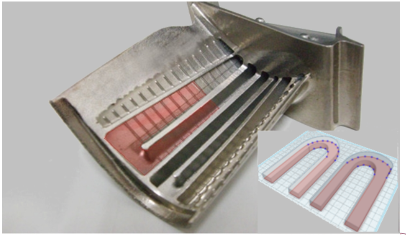
Outline

- Background and contributions.
- Estimate gradient by using the space(-time) solution.
- Optimization framework.
- Numerical examples.

Background

- Optimization constrained by conservation law simulation.
- Expensive simulation.
- High dimensional design.
- Code has no adjoint and is proprietary / legacy.





Internal cooling of turbine airfoil
Source: http://www.amaterastvo.biz/eng/technologies.html

Research scope

- Gray-box conservation law simulation:
  - adjoint not available nor implementable.
  - provide space(-time) solution.

	space(-time) solution	Adjoint
Black-box	✘	✘
Gray-box	✔	✘
Open-box		✔

- High-dimensional design space.

Different optimization methods apply

- Black-box: use derivative-free optimization, pattern search methods, evolution based methods, etc.
  - not require derivative evaluation.
  - not suitable for high dimensional optimization.
- Open-box: use gradient-based optimization, quasi-Newton methods, etc.
  - requires efficient gradient evaluation, generally using adjoint [Lions 71] .
  - suitable for high dimensional optimization.
- Gray-box: treated as black-box. Space(-time) solution wasted!

Problem statement

- Minimize

$$\xi(c) = \int_0^T \int_\Omega j(\boldsymbol{u}, c)\mathrm{d}\mathbf{x}\mathrm{d}t$$

using a gradient-based optimization method.
$\boldsymbol{u}$: the space-time solution of

$$\dot{\boldsymbol{u}} + \nabla \cdot \boldsymbol{F}(\boldsymbol{u}) = \boldsymbol{q}(\boldsymbol{u}, c)$$

whose $\boldsymbol{F}$ is unknown.

- Steady state solution $\boldsymbol{u}_\infty$

$$\nabla \cdot \boldsymbol{F}(\boldsymbol{u}_\infty) = \boldsymbol{q}(\boldsymbol{u}_\infty, c)$$

is a special case of the time-dependent solution.

Contributions

- Developed the twin model method to efficiently estimate the gradient of objective functions constrained by gray-box simulations.
- Practically and theoretically presented the utility of the estimated gradient in a Bayesian optimization framework.
- Applied the twin-model GPO to a Buckley-Leverett simulation and the Navier-Stokes simulaiton to reduce the number of gray-box simulations required to achieve desired optimization results.

Outline

- Background and contribution.
- Estimate gradient by using the space(-time) solution.
  - Leverage the space-time solution.
  - Parameterize the flux.
  - Algorithm for training twin model.
  - Numerical examples.
- Optimization framework.
- Numerical examples.

Estimate adjoint from space-time solution

- Primal:

$$\frac{\partial u}{\partial t} + \frac{dF}{du}\frac{\partial u}{\partial x} = q\,, \quad \text{with I.C. and B.C.}$$

- Adjoint:

$$\frac{\partial v}{\partial t} + \frac{dF}{du}\frac{\partial v}{\partial x} = p\,, \quad \text{with I.C. and B.C.}$$

Observations:

- From $u(t,x)$, we get $\frac{\partial u}{\partial t}$, $\frac{\partial u}{\partial x}$.
- From $\frac{\partial u}{\partial t}$, $\frac{\partial u}{\partial x}$, we estimate $\frac{dF}{du}$.
- From $\frac{dF}{du}$, we solve for v.
- Primal / adjoint depend on $\frac{dF}{du}$, not F.

Leverage the space(-time) solution

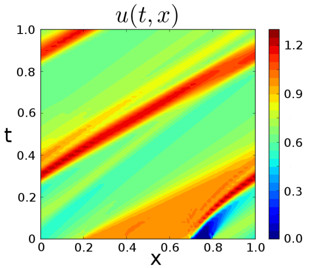
Estimate the gradient:

- Infer the conservation law from the space(-time) solution.
- Estimate the gradient by adjoint method.

Example: infer flux $F(u)$ from space-time solution.

$$\frac{\partial u}{\partial t} + \frac{\partial F(u)}{\partial x} = c(x)$$
$$u(t=0,x) = u_0(x)$$
$$u(t,x=0) = u(t,x=1)$$

Space-time solution

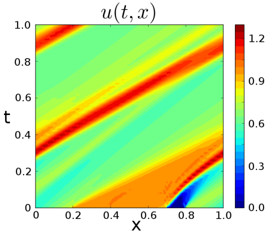


Infer the flux that reproduces the space(-time) solution. The inferred conservation law is called twin model.

Inference can boil down to minimization

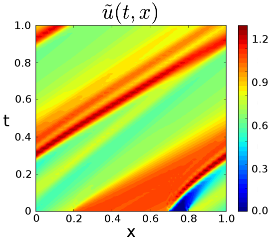
Gray-box model

$$\frac{\partial u}{\partial t} + \frac{\partial F(u)}{\partial x} = c(x)$$
$$u(t=0,x) = u_0(x)$$
$$u(t,x=0) = u(t,x=1)$$

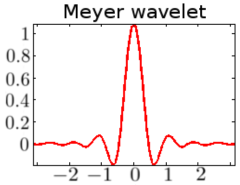


Twin model

$$\frac{\partial \tilde{u}}{\partial t} + \frac{\partial \tilde{F}(\tilde{u})}{\partial x} = c(x)$$
$$\tilde{u}(t=0,x) = u_0(x)$$
$$\tilde{u}(t,x=0) = \tilde{u}(t,x=1)$$

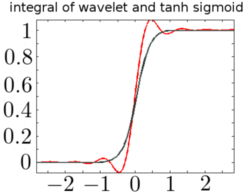


$$\min_{\tilde{F}} \left\{ \mathcal{M}(\tilde{F},u) \equiv \int_0^1 \int_0^1 (u-\tilde{u})^2_{L_2} \; dt\, dx \right\}$$



- Multi-resolution analysis (MRA) [Mallat 89]
  Wavelet basis
- Adjoint depends on $\nabla_u F$
- Use the integral of wavelet as basis for univariate flux.

$$\phi(u) = \begin{cases} 0 \,, u \to -\infty \\ 1 \,, u \to \infty \end{cases} \,, \quad \phi(u) = \frac{1}{2}(\tanh(u) + 1)$$



- Univariate flux: use sigmoid function as basis [Mhaskar 92] .

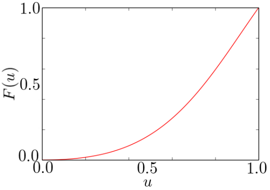
$$\phi_{j,\eta}(u) = \phi(\lambda_j u - \eta) \,, \quad \tilde{F}(u) = \sum_{j,\eta} \alpha_{j,\eta} \phi_{j,\eta}(u)$$

- Multivariate flux: tensor product of univariate sigmoid basis.

Estimate gradient for BL equation

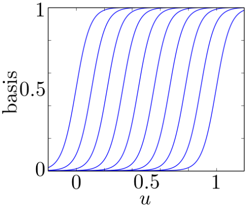
- Consider a Buckley-Leverett equation [Buckley 42]

$$F(u) = \frac{u^2}{1+2(1-u)^2}\ , \quad c: \ \text{constant-valued control}$$



- Infer the twin model:

$$\min_{\boldsymbol{\alpha}} \left\{ \underbrace{\int_0^1 \int_0^1 (u - \tilde{u})^2 \ dt\, dx}_{\mathcal{M}} + \lambda \left\|\boldsymbol{\alpha}\right\|_{L_1} \right\}, \ \lambda > 0$$

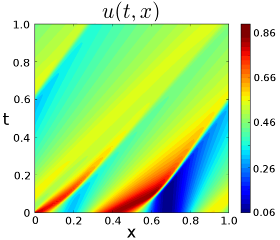


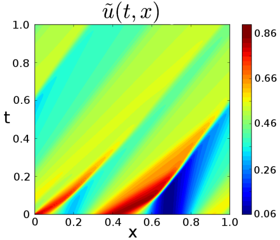
- Estimate $d\xi/dc$ [Kucuk 06]

$$\xi(c) \equiv \int_{x=0}^{1} \left(u(x, t=1; c) - 0.5\right)^2 \mathrm{d}x$$

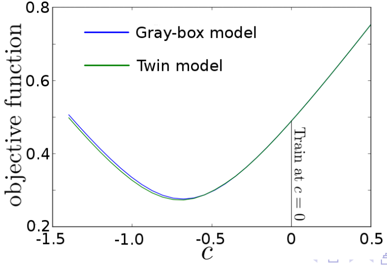
Estimate gradient for BL equation, cont.

- Train a twin model using $u(t, x)$ at $c = 0$.

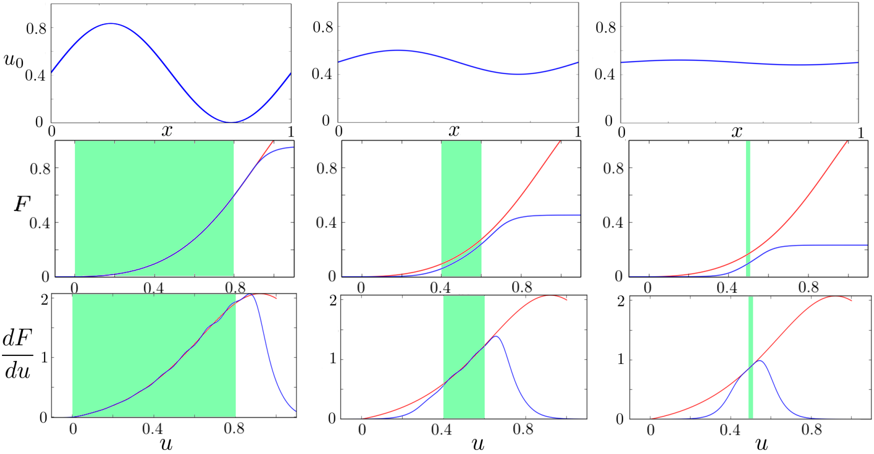




- Twin model can estimate $d\xi/dc$ at $c = 0$.



Only expect match where there is solution



- The range of the discrete gray-box solution $\mathcal{E}_u := [u_{\min}, u_{\max}]$ are shaded by green.
- The twin model flux does not affect $\mathcal{M}$ outside the range of the gray-box solution.

Reduce mismatch by adding more basis

Consider improvement of least solution mismatch using more basis functions:

- Basis $[\mathbf{\Phi}_0]$:

$$\mathcal{M}_0^* = \min_{\boldsymbol{\alpha}_0} \mathcal{M}\left(\tilde{F}(\boldsymbol{\alpha}_0), u\right)$$

- Append basis $[\mathbf{\Phi}_0; \mathbf{\Phi}_1]$:

$$\mathcal{M}_{01}^* = \min_{\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1} \mathcal{M}\left(\tilde{F}(\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1), u\right)$$

- Mismatch improvement due to appending $\mathbf{\Phi}_1$:

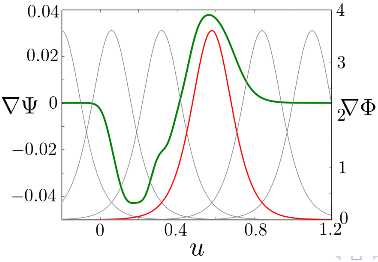
$$\Delta\mathcal{M}^* = \mathcal{M}_0^* - \mathcal{M}_{01}^* \geq 0$$

Rank the basis to be added

- $$\Delta\mathcal{M}^* \approx -\underbrace{\left(\int_\Omega \frac{\partial \mathcal{M}}{\partial \tilde{F}}\bigg|_{\tilde{F}(\boldsymbol{\alpha}_0^*)} \boldsymbol{\Phi}_1 \, \mathrm{d}u\right)}_{\text{weight vector}} \cdot \boldsymbol{\alpha}_1 = \underbrace{\left(\int_\Omega \nabla\Psi \cdot \nabla\boldsymbol{\Phi}_1 du\right)}_{\text{weight vector}} \cdot \boldsymbol{\alpha}_1 \ ,$$

  where $\nabla^2\Psi = \frac{\partial \mathcal{M}}{\partial \tilde{F}}\Big|_{\tilde{F}(\boldsymbol{\alpha}_0^*)}$

- The weight vector ranks the importance of basis. Select the basis with the largest absolute value of weight [Miller 90] .



Represent the sigmoid functions

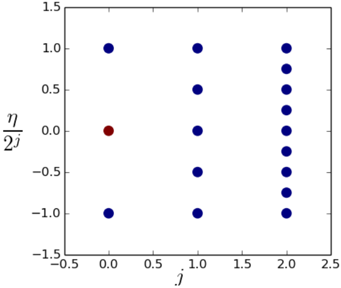
- Consider the sigmoid functions:

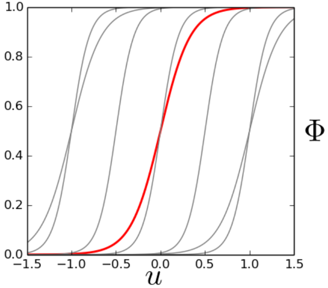
$$\phi_{j,\eta}(u) = \phi(2^j u - \eta)$$

  Denote $\phi_{j,\eta}(u)$ by the tuple $\{j, \eta\}$.

  $j \in \{j_0, j_0 + 1, j_0 + 2, \cdots\}$ represents resolution. $\frac{\eta}{2^j}$ represents the center of the sigmoid's derivative.

- $j = 0$, $\eta = 0$





Represent the sigmoid functions

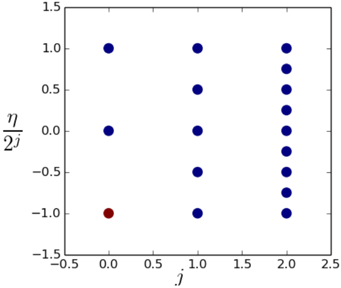
- Consider the sigmoid functions:

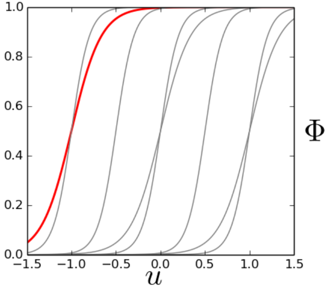
$$\phi_{j,\eta}(u) = \phi(2^j u - \eta)$$

  Denote $\phi_{j,\eta}(u)$ by the tuple $\{j, \eta\}$.

  $j \in \{j_0, j_0 + 1, j_0 + 2, \cdots\}$ represents resolution. $\frac{\eta}{2^j}$ represents the center of the sigmoid's derivative.

- $j = 0$, $\eta = -1$





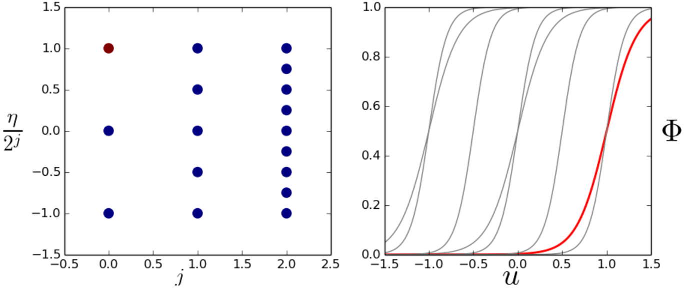
- Consider the sigmoid functions:

$$\phi_{j,\eta}(u) = \phi(2^j u - \eta)$$

Denote $\phi_{j,\eta}(u)$ by the tuple $\{j, \eta\}$.

$j \in \{j_0, j_0 + 1, j_0 + 2, \cdots\}$ represents resolution. $\frac{\eta}{2^j}$ represents the center of the sigmoid's derivative.

- $j = 0$, $\eta = 1$



Represent the sigmoid functions

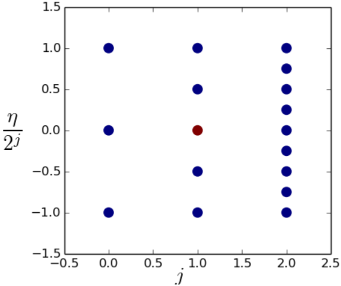
- Consider the sigmoid functions:

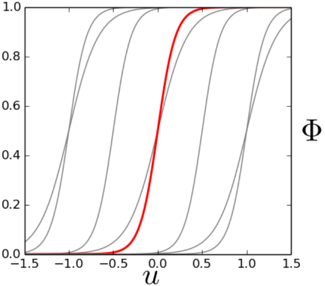
$$\phi_{j,\eta}(u) = \phi(2^j u - \eta)$$

  Denote $\phi_{j,\eta}(u)$ by the tuple $\{j,\eta\}$.

  $j \in \{j_0, j_0+1, j_0+2, \cdots\}$ represents resolution. $\frac{\eta}{2^j}$ represents the center of the sigmoid's derivative.

- $j = 1$, $\eta = 0$





Represent the sigmoid functions

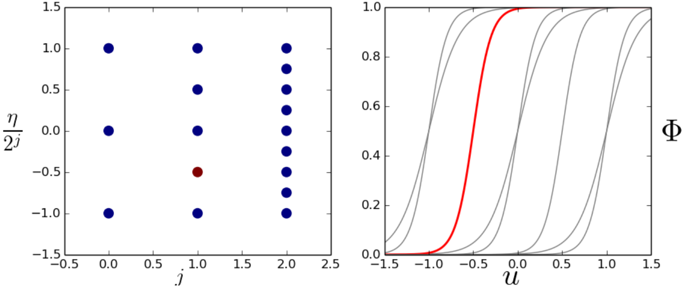
- Consider the sigmoid functions:

$$\phi_{j,\eta}(u) = \phi(2^j u - \eta)$$

Denote $\phi_{j,\eta}(u)$ by the tuple $\{j, \eta\}$.

$j \in \{j_0, j_0 + 1, j_0 + 2, \cdots\}$ represents resolution. $\frac{\eta}{2^j}$ represents the center of the sigmoid's derivative.

- $j = 1$, $\eta = -1$



Represent the sigmoid functions

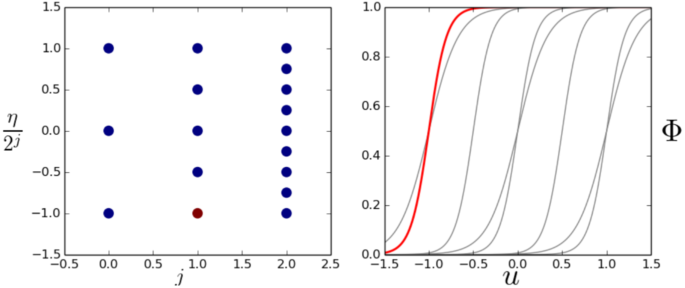
- Consider the sigmoid functions:

$$\phi_{j,\eta}(u) = \phi(2^j u - \eta)$$

Denote $\phi_{j,\eta}(u)$ by the tuple $\{j, \eta\}$.

$j \in \{j_0, j_0 + 1, j_0 + 2, \cdots\}$ represents resolution. $\frac{\eta}{2^j}$ represents the center of the sigmoid's derivative.

- $j = 1$, $\eta = -2$



Represent the sigmoid functions

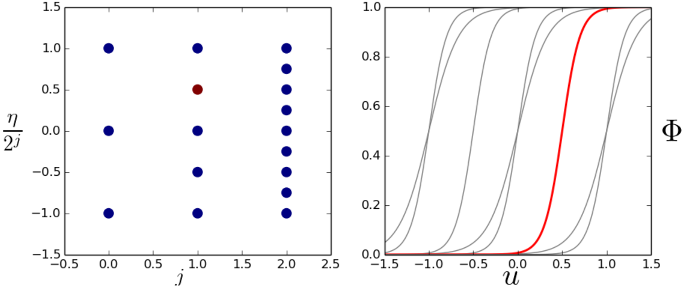
- Consider the sigmoid functions:

$$\phi_{j,\eta}(u) = \phi(2^j u - \eta)$$

Denote $\phi_{j,\eta}(u)$ by the tuple $\{j, \eta\}$.

$j \in \{j_0, j_0 + 1, j_0 + 2, \cdots\}$ represents resolution. $\frac{\eta}{2^j}$ represents the center of the sigmoid's derivative.

- $j = 1$, $\eta = 1$



Represent the sigmoid functions

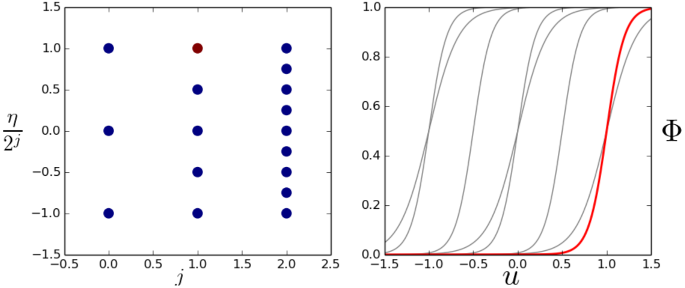
- Consider the sigmoid functions:

$$\phi_{j,\eta}(u) = \phi(2^j u - \eta)$$

Denote $\phi_{j,\eta}(u)$ by the tuple $\{j, \eta\}$.

$j \in \{j_0, j_0 + 1, j_0 + 2, \cdots\}$ represents resolution. $\frac{\eta}{2^j}$ represents the center of the sigmoid's derivative.

- $j = 1$, $\eta = 2$

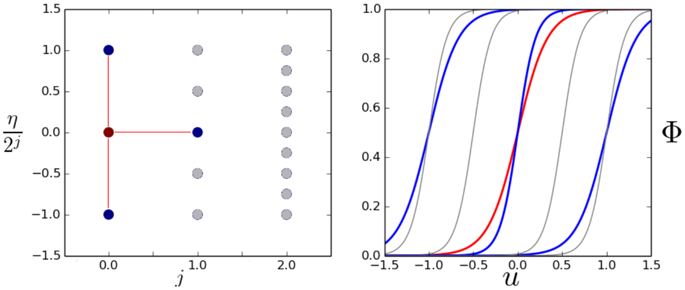


Neighborhood of a sigmoid function

- Define the "neighborhood" of a sigmoid function $\{j, \eta\}$ to be

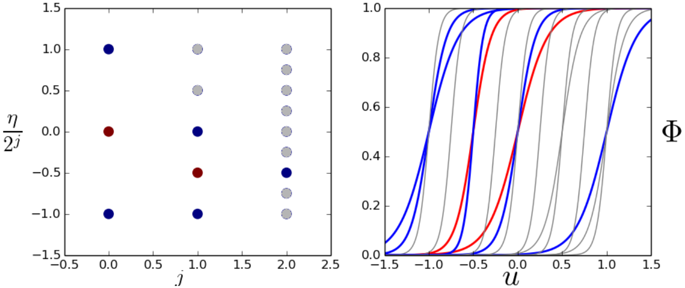
$$\mathcal{N}(\{j,\eta\}) := \{\{j+1, 2\eta\}, \{j, \eta \pm 1\}\}$$

- $\mathcal{N}(\{0, 0\})$



- Define

$$\mathcal{N}(\{j_1,\eta_1\},\cdots,\{j_n,\eta_n\}) := \mathcal{N}(\{j_1,\eta_1\})\bigcup\cdots\bigcup\mathcal{N}(\{j_n,\eta_n\})$$



Select the basis adaptively

- Finite dictionary.
  Can be solved by matching pursuit [Mallat 93]. (forward selection [Friedman 94] , backward prunning [Reed 93]).

  Requires the predetermined basis dictionary contains a subset sufficient for approximation.

- Infinite dictionary. [Jekabsons 10, Blatman 10] .

  Start from an empty or simple dictionary.

  - **Forward step:** (improve accuracy)
    Add a new basis to the dictionary.
  - **Backward step:** (avoid overfit)
    Try to remove a basis from the dictionary.

  Iterate forward / backward steps until the “approximation quality” no longer improves.

Train twin model

Algorithm 1 Train twin model

Input: Basis library $\Psi = \Psi_0$, coefficients $\alpha = \mathbf{0}$. Gray-box solution u

loop

Minimize solution mismatch $\alpha_\Psi \leftarrow \min_\alpha \mathcal{M}\left(\tilde{F}(\Psi, \alpha), u\right)$

if No more basis needed **then**

break

else

Add a basis.

Forward step

if No basis shall be deleted **then**

continue

else

Delete a basis.

Backward step

Output: Ψ, α

Criterion for basis addition / deletion

- Balance approximation accuracy and model complexity.
  k-fold cross validation [Geisser 93] .

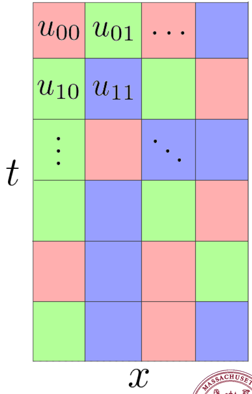
- Shuffle u dataset into k disjoint sets:
  u_1, u_2, $\cdots$, u_k.
  `ravel(`u`);  [`$u_1, \cdots, u_k$`]=shuffle(`u`,k)`

- Train k twin models.
  T_1 = `TrainTwinModel(`$u_2, \cdots, u_k$`)`
  ...
  T_k = `TrainTwinModel(`$u_1, \cdots, u_{k-1}$`)`

- Validate.
  $\mathcal{M}_1$ = `SolutionMismatch(`T_1`, ` u_1`)`
  ...
  $\mathcal{M}_k$ = `SolutionMismatch(`T_k`, ` u_k`)`

- Compute $\overline{\mathcal{M}}$=`mean(`$\mathcal{M}_1, \cdots, \mathcal{M}_k$`)`

- Add / delete basis if $\overline{\mathcal{M}}$ decreases.



Train twin model

Algorithm 1 Train twin model

Input: Basis library $\Psi = \Psi_0$, coefficients $\alpha = \mathbf{0}$. Gray-box solution u

loop

Minimize solution mismatch $\alpha_\Psi \leftarrow \min_\alpha \mathcal{M}\left(\tilde{F}(\Psi,\alpha), u\right)$

if No more basis needed **then**

  break

else

  Add a basis.

Forward step

if No basis shall be deleted **then**

  continue

else

  Delete a basis.

Backward step

Output: Ψ, α

Algorithm 1 Train twin model

Input: Basis library $\Psi = \Psi_0$, coefficients $\alpha_\Psi = \mathbf{0}$. $\overline{\mathcal{M}}_0 = \infty$. Gray-box solution u

loop

Minimize solution mismatch $\alpha_\Psi \leftarrow \min_\alpha \mathcal{M}\left(\tilde{F}(\Psi, \alpha), u\right)$

Compute $v = \frac{\partial \mathcal{M}}{\partial \tilde{F}}\Big|_{\tilde{F}(\Psi, \alpha_\Psi)}$

Find $\phi_{add} \in \mathcal{N}(\Psi)$ with maximum $\left|\int_\Omega v \phi_{add} \, \mathrm{d}u\right|$

$\Psi \leftarrow \Psi \bigcup \{\phi_{add}\}$, $\alpha \leftarrow \{\alpha, 0\}$

Train twin model via k-fold cross validation, compute $\overline{\mathcal{M}}$

if $\overline{\mathcal{M}} < \overline{\mathcal{M}}_0$ **then**

Accept addition, $\overline{\mathcal{M}}_0 \leftarrow \overline{\mathcal{M}}$, train twin model using all u, update α

else

Reject addition, $\Psi \leftarrow \Psi \backslash \{\phi_{add}\}$, $\alpha \leftarrow \alpha \backslash \{\alpha_{\phi_{add}}\}$, **break**

Compute $v = \frac{\partial \mathcal{M}}{\partial \tilde{F}}\Big|_{\tilde{F}(\Psi, \alpha_\Psi)}$

Find $\phi_{del} \in \Psi$ with least $\left|\int_\Omega v \phi_{del} \, \mathrm{d}u\right|$

if $\phi_{del} \neq \phi_{add}$ **then**

$\Psi \leftarrow \Psi \backslash \{\phi_{del}\}$, $\alpha \leftarrow \alpha \backslash \{\alpha_{\phi_{del}}\}$

Train twin model via k-fold cross validation, compute $\overline{\mathcal{M}}$

if $\overline{\mathcal{M}} > \overline{\mathcal{M}}_0$ **then**

Reject deletion, $\Psi \leftarrow \Psi \bigcup \{\phi_{del}\}$, $\alpha \leftarrow \alpha \bigcup \{\alpha_{\phi_{del}}\}$

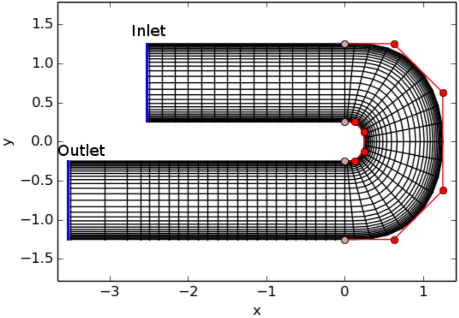
else

Accept deletion, train twin model using all u, update α

Output: Ψ, α

Estimate gradient for a Navier-Stokes flow

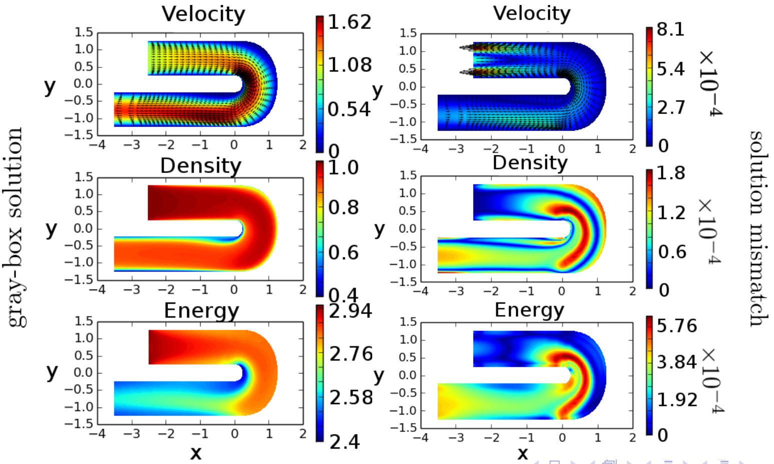
- Steady-state, 2-D, compressible, laminar Navier-Stokes flow with unknown state equation.
  ($\rho, U \to p$: ideal, van der Waals, Redlich-Kwong [Redlich 49] , etc)
- Geometry



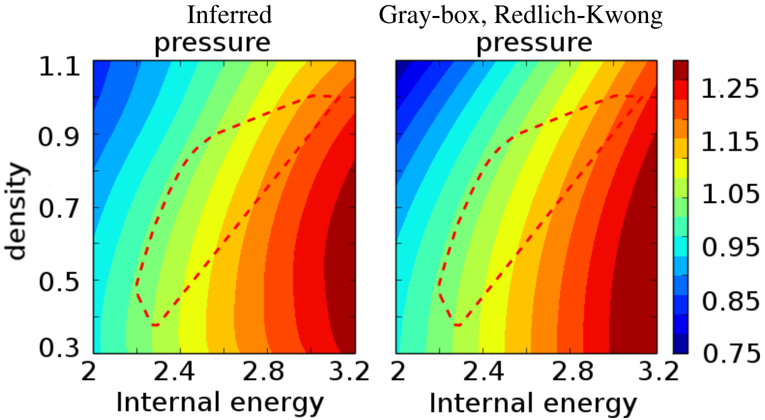
- Bending boundary generated by B-splines.
- Inlet: fixed ρ, p^t. Outlet: fixed p. No slip boundary.

Train the state equation

$$\mathcal{M} = w_\rho \int_\Omega |\tilde{\rho}_\infty - \rho_\infty|^2 \, d\mathbf{x} + w_u \int_\Omega |\tilde{u}_\infty - u_\infty|^2 \, d\mathbf{x}$$
$$+ w_v \int_\Omega |\tilde{v}_\infty - v_\infty|^2 \, d\mathbf{x} + w_E \int_\Omega \left|\tilde{E}_\infty - E_\infty\right|^2 \, d\mathbf{x}$$

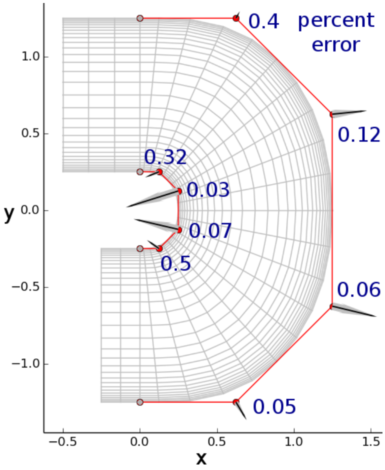


Twin model fits the state equation

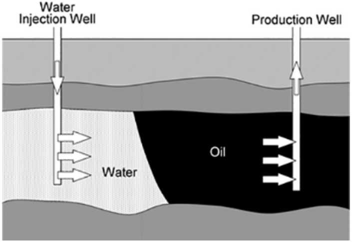


Estimate the gradient

- ξ: steady state mass flux rate. c: control points' coordinates.



- Waterflooding for secondary recovery, high water cut.
- Inject polymer to enhance water-phase viscosity and remove residual oil.



Governing equation:

$$\frac{\partial}{\partial t}(\rho_\alpha \phi S_\alpha) + \nabla \cdot (\rho_\alpha \vec{v}_\alpha) = 0\,, \quad \alpha \in \{w, o\}$$

$$\frac{\partial}{\partial t}(\rho_w \phi S_w c) + \nabla \cdot (c\rho \vec{v}_{wp}) = 0$$

Darcy's law:

$$\vec{v}_\alpha = -M_\alpha k_{r\alpha} \boldsymbol{K} \cdot (\nabla p - \rho_w g \nabla z), \quad \alpha \in \{w, o\}$$

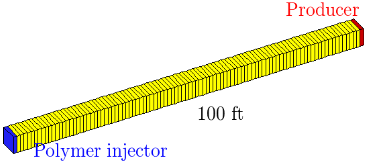
$$\vec{v}_{wp} = M_{wp} k_{rw} \boldsymbol{K} (\nabla p - \rho_w g \nabla z)$$

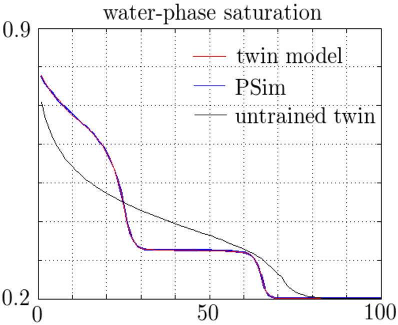
Mobility factors M_o, M_w, M_{wp} depends on S_w, p, c.

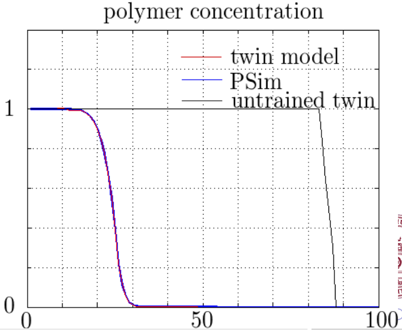
Train the mobility factors, 1D

Minimize solution mismatch:

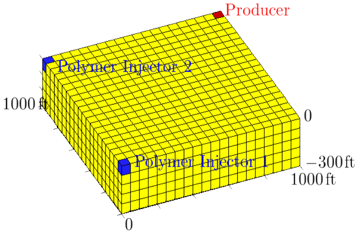
$$\mathcal{M} = w_{S_w} \int_T \int_\Omega |S_w - \tilde{S}_w|^2 d\boldsymbol{x} dt + w_c \int_T \int_\Omega |c - \tilde{c}|^2 d\boldsymbol{x} dt + w_p \int_T \int_\Omega |p - \tilde{p}|^2 d\boldsymbol{x} dt$$

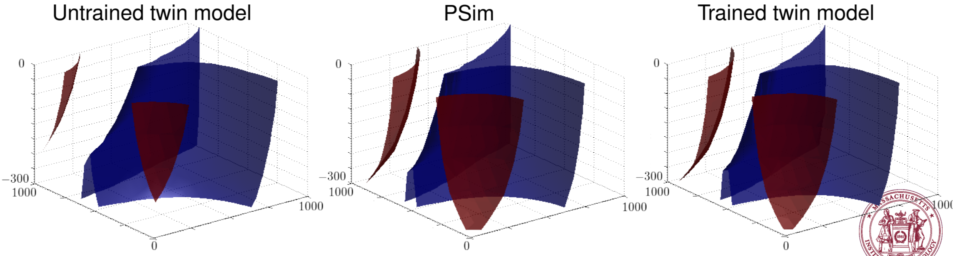






Train the mobility factors, 3D



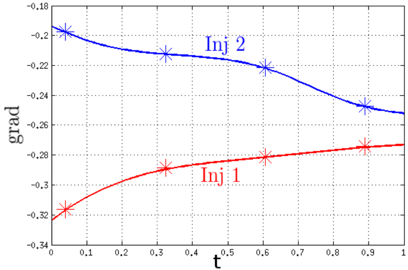


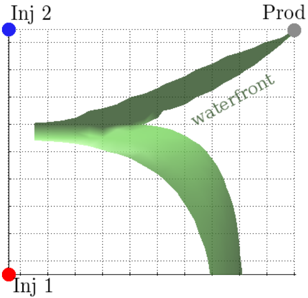
Estimate gradient for injection schedule

Objective: oil residual at $t = 50\,\mathrm{days}$,

$$\xi = \int_\Omega \rho_o \phi S_o \,\mathrm{d}\boldsymbol{x}$$

Controls: injection rate schedule at two injectors.





Outline

- Background and contribution.
- Estimate gradient by using the space(-time) solution.
- Optimization framework.
  - Model the estimated gradient.
  - Twin-GPO framework.
  - Convergence properties.
  - A numerical example.
- Numerical examples.

- The gradient estimated by twin model is not exactly the true gradient.
  - Gray-box solution under-resolved.
  - Solvers use different numerical implementations.

- To identify the sources of the errors and quantify the errors are difficult. Model the error of each component by "model discrepancy" [Kennedy 01, Higdon 04].

$$\xi_{\tilde{\nabla}}(c)_1 = \rho_1 \nabla\xi(c)_1 + \epsilon_1(c)$$
$$\cdots$$
$$\xi_{\tilde{\nabla}}(c)_d = \rho_d \nabla\xi(c)_d + \epsilon_d(c)$$

$$\xi_{\tilde{\nabla}} : \text{ estimated gradient }, \qquad \nabla\xi : \text{ true gradient }.$$

Assumptions for modeling the gradient

- Model $\xi, \epsilon_1, \cdots, \epsilon_d$ as stationary Gaussian processes with covariances $K, G_1, \cdots, G_d$ respectively [Kennedy 01, Higdon 04] .
- Assume the gradient error to be independent with the objective.

$$\operatorname{cov}\left[\xi(c_1), \epsilon_i(c_2)\right] = 0 \qquad \text{for all } c_1, c_2 \in \mathcal{C}\ .\ i = 1, \cdots, d\ .$$

- For simplicity, assume the components of the gradient error are pairwise independent.

$$\operatorname{cov}\left[\epsilon_i, \epsilon_j\right] = 0 \quad \text{for } i \neq j$$

- For simplicity, assume the covariance functions are isotropic. $K(c_1, c_2),\ G_1(c_1, c_2),\ \cdots\ G_d(c_1, c_2)$ only depend on $\left\| c_1 - c_2 \right\|$. (Use L_2 norm.)

Modeling the joint distribution

Predict ξ and its error bar at a new point c using co-Kriging.
$\underline{c}_n := (c_1, \cdots, c_N)$: sampled points.

$$\begin{pmatrix} \xi(c) \\ \xi(\underline{c}_n) \\ \xi_{\tilde{\nabla}}(\underline{c}_n) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu \\ \boldsymbol{\mu} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} K(c,c) & \mathbf{v} & \mathbf{w} \\ \mathbf{v}^T & \mathbf{D} & \mathbf{H} \\ \mathbf{w}^T & \mathbf{H}^T & \mathbf{E}+\mathbf{G} \end{pmatrix} \right),$$

$$\mathbf{v} = (K(c,c_1), \cdots, K(c,c_N)), \quad \mathbf{w} = (\nabla_{c_1} K(c,c_1), \cdots, \nabla_{c_N} K(c,c_N))$$

$$\mathbf{D} = \begin{pmatrix} K(c_1,c_1) & \cdots & K(c_1,c_N) \\ \vdots & \ddots & \vdots \\ K(c_N,c_1) & \cdots & K(c_N,c_N) \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} \nabla_{c_1'} K(c_1,c_1') & \cdots & \nabla_{c_N'} K(c_1,c_N') \\ \vdots & \ddots & \vdots \\ \nabla_{c_1'} K(c_N,c_1') & \cdots & \nabla_{c_N'} K(c_N,c_N') \end{pmatrix}$$

$$\mathbf{E} = \begin{pmatrix} \nabla_{c_1}\nabla_{c_1'} K(c_1,c_1') & \cdots & \nabla_{c_1}\nabla_{c_N'} K(c_1,c_N') \\ \vdots & \ddots & \vdots \\ \nabla_{c_1}\nabla_{c_N'} K(c_N,c_1') & \cdots & \nabla_{c_N}\nabla_{c_N'} K(c_N,c_N') \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} G(c_1,c_1') & \cdots & G(c_1,c_N') \\ \vdots & \ddots & \vdots \\ G(c_N,c_1') & \cdots & G(c_N,c_N') \end{pmatrix}$$

$$G(c_i,c_j) = \mathrm{diag}\big(G_1(c_i,c_j), \cdots, G_d(c_i,c_j)\big)$$

- The Matern 5/2 kernel [Matern 60, Snoek 12]

$$C_{\frac{5}{2}}(c_1, c_2) = \sigma^2 \left(1 + \frac{\sqrt{5}|c_1 - c_2|}{L} + \frac{5|c_1 - c_2|^2}{3L^2}\right) \exp\left(-\frac{\sqrt{5}|c_1 - c_2|}{L}\right)$$

  Hyper-parameters: σ^2's, L's, ρ's, and μ.

- Maximum likelihood estimate (MLE)[Jones 98]

$$\max_{\sigma^2, L, \rho, \mu} \left\{\log p\left(\xi(\underline{c}_n), \xi_{\tilde{\nabla}}(\underline{c}_n) \middle| \sigma^2, L, \rho, \mu\right)\right\}$$

  Solved by gradient based optimization.

- Full Bayesian approach [Higdon 04, Kennedy 01]

$$p(\sigma^2, L, \rho | \xi(\underline{c}_n), \xi_{\tilde{\nabla}}(\underline{c}_n))$$

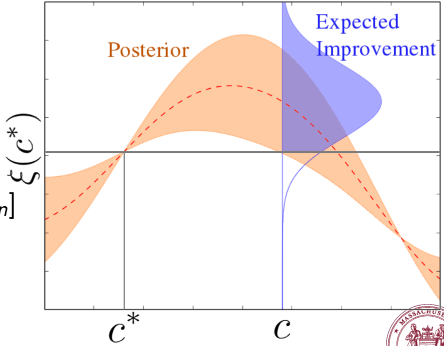
- Both approaches provide similar results [Bayarri 07] . We use the MLE approach.

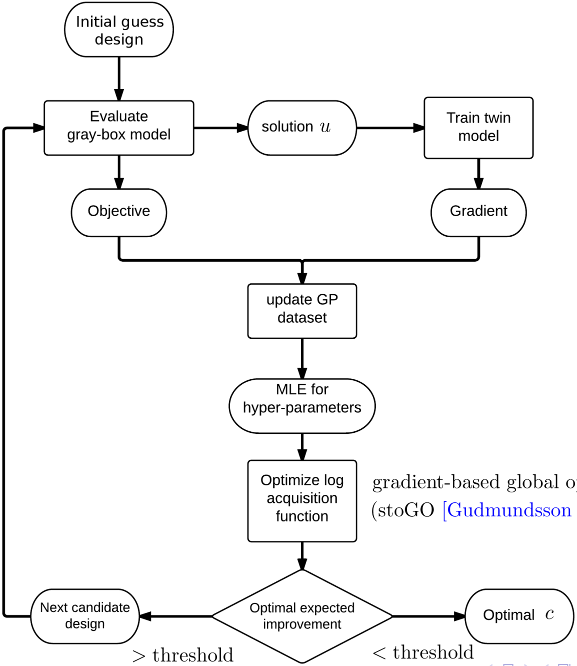
Gaussian process optimization (GPO)

- GPO introduces an acquisition function:
  $\rho(c)$: the expected utility of investing the next sample at c given the posterior.

- Expected improvement
  [Mokus 78, Snoek 12]

$$\rho_{\mathrm{EI}}(c) = \mathbb{E}\left[\max\left(\xi(c) - \xi(c^*), 0\right) \mid \mathcal{F}_n\right]$$





gradient-based global optimization (stoGO [Gudmundsson 98])

GPO will explore the entire design space as $n \to \infty$. [**Vazquez 10**]

Twin-model GPO will explore the entire design space as $n \to \infty$.

- Let $\xi \in \mathcal{K}(\mathcal{C})$. $\mathcal{K}$ is the reproducing kernel Hilbert space (RKHS) with the semi-positive definite kernel $K : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$. Define $\Phi(c) := K(c, 0)$ for all $c \in \mathcal{C}$. $\hat{\Phi}$ denotes the Fourier transform of Φ.

  Let $\epsilon_i \in \mathcal{H}_G^i$. $\mathcal{H}_G^i$ is the RKHS with the semi-positive definite kernel G_i.
- Assume there exist $C \geq 0$ and $k \in \mathbb{N}^+$, such that $(1 + |\eta|^2)^k \left|\hat{\Phi}(\eta)\right| \geq C$ for all $\eta \in \mathbb{R}^d$.
- For all $c_{init} \in \mathcal{C}$, all $\xi \in \mathcal{H}_K$ and $\epsilon_i \in \mathcal{H}_G^i$, $i = 1, \cdots, d$, the sequence c_n generated by twin-model GPO is dense in $\mathcal{C}$.

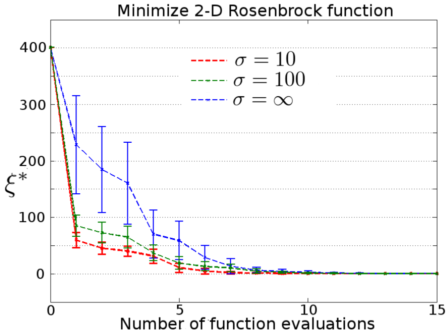
Gradient improves optimization performance

- Rosenbrock 2-D

$$\xi(c_1, c_2) = (1 - c_1)^2 + 100(c_2 - c_1^2)^2$$

  Global minimum $\xi(1,1) = 0$

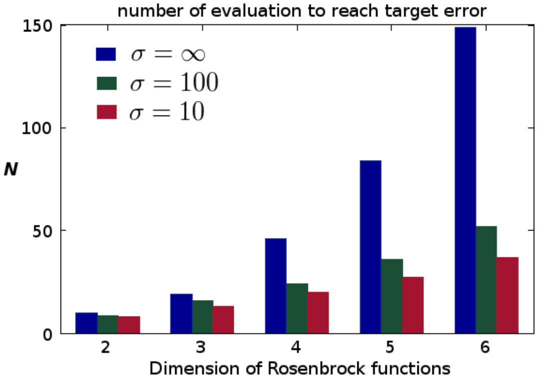
- Simulate ϵ_i, $i = 1, \cdots, d$ by i.i.d. stationary GP with correlation length 1 and variance σ^2.
  Accurate: $\sigma = 10$. Noisy: $\sigma = 100$. None: $\sigma = \infty$.



More benefit for higher dimension

- Generalized *n*-D Rosenbrock function

$$\xi(c) = \sum_{i=0}^{d-2} 100(c_{i+1} - c_i)^2 + 1(1 - c_i)^2$$



Outline

- Background and contribution.
- Estimate gradient by using the space(-time) solution.
- Optimization framework.
- Numerical examples

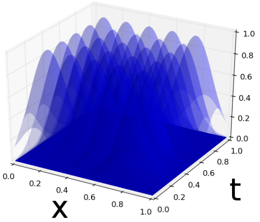
Optimize injection for BL equation

- $u(t,x)$ generated by:

$$\frac{\partial u}{\partial t} + \frac{\partial F(u)}{\partial x} = c(t,x) \quad x \in [0,1]\ t \in [0,1]$$
$$u(t=0,x) = u_0(x),\ u(t,x=0) = u(t,x=1)$$

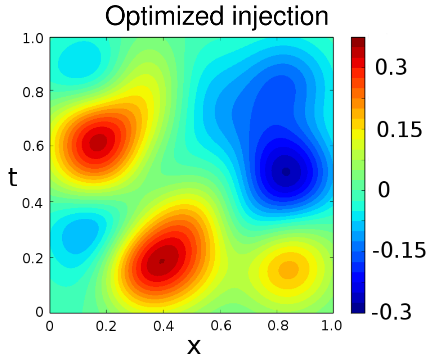
- Control:

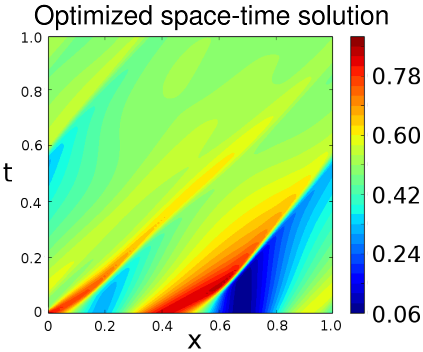
$$c(t,x) = \sum_{i=1}^{m} \sum_{j=n}^{s} c_{ij} \cdot B_{ij}(t,x)$$
$$B_{ij} = \exp\left(-\frac{(t-t_i)^2}{L_t^2}\right) \exp\left(-\frac{(x-x_j)^2}{L_x^2}\right)$$



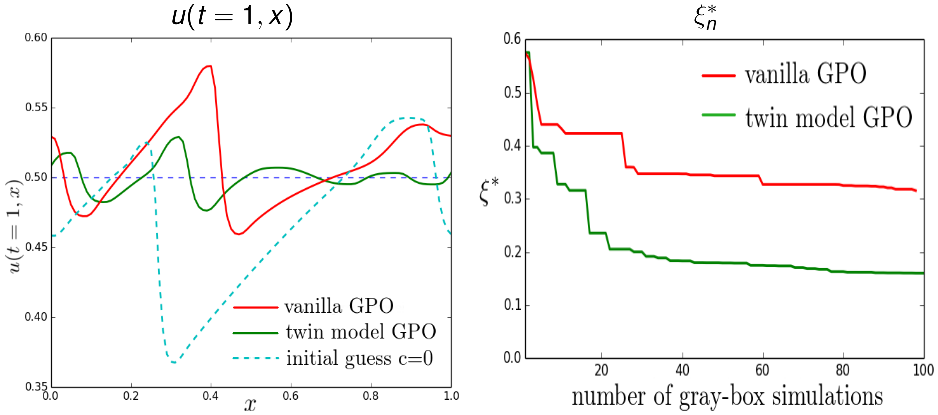
- Objective function:

$$\xi(c) = \int_{x=0}^{1} \left| u(t=1,x) - \frac{1}{2} \right|^2 + \lambda \sum_{ij} c_{ij}^2, \quad \lambda > 0$$


Optimized injection
1.0
0.8
0.6
t
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1.0
x
0.3
0.15
0
-0.15
-0.3

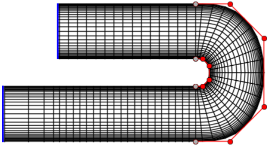

Optimized space-time solution
1.0
0.8
0.6
t
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1.0
x
0.78
0.60
0.42
0.24
0.06

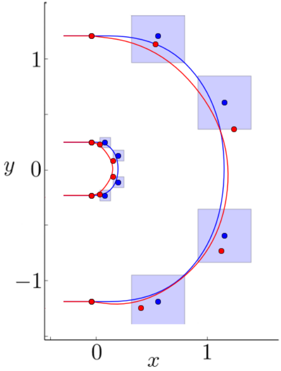
Compare optimization performance

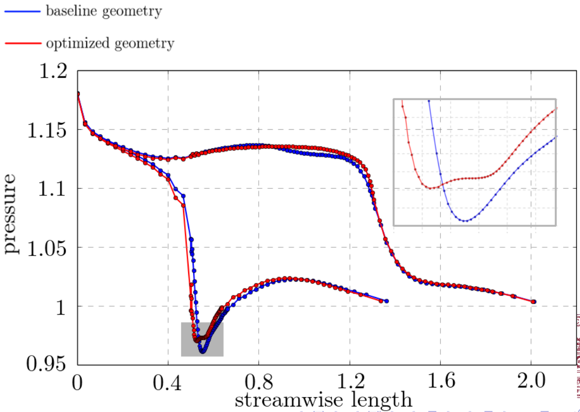


Optimize return bend

Objective: maximize mass flux.
Constraint by fixed area of the return bend.
Unknown state equation.

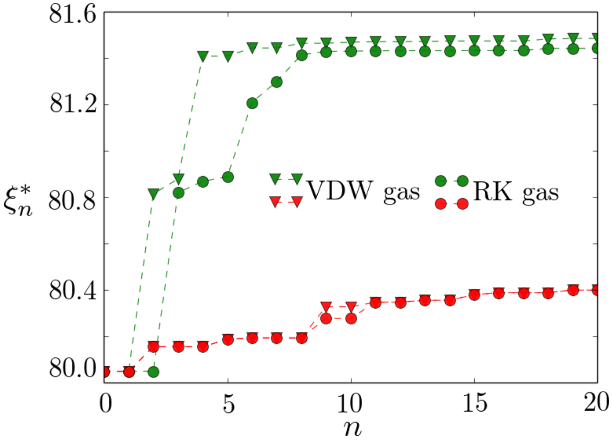






Twin model improves GPO performance

- Faster improvement for the current best objective function evaluation.
- Most improvement is achieved for small *n*.



Thank you!

Thesis progress

- Introduction
  - Optimization constrained by conservation law simulations.
  - Literature review.
    - Derivative-free optimization methods.
    - Gradient-based optimization methods.
    - Adjoint method.
  - Challenges for optimization constrained by gray-box simulations.
  - Conservation law with unknown flux function.
  - Thesis Objectives.
  - Contributions.

Thesis progress

- Estimate gradient by using the space-time solution.
  - Approach.
    - Infer conservation law from space(-time) solution.
    - Estimate the gradient for inferred conservation law.
  - Implementation.
    - Flux parameterization.
    - Adaptive refinement of basis functions.
    - Algorithm for training twin model.
    - Numerical examples.

Thesis progress

- Optimization framework.
  - Model the estimated gradient.
  - Twin-model GPO.
  - Discussion of onvergence properties.
    - Theoretical results on the design sequence.
    - Experimental results on the convergence rate.
  - Numerical examples.
- Conclusions.

Key dates

- April 29. Submit draft to Prof. Wang.
- May 9. Submit draft to committee and thesis readers.
- June 20-. Thesis defense.

Details of return bend testcase

N-S equation:

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho E\end{pmatrix}+\frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2+p-\sigma_{xx}\\ \rho uv-\sigma_{xy}\\ u(E\rho+p)-\sigma_{xx}u-\sigma_{xy}v\end{pmatrix}+\frac{\partial}{\partial y}\begin{pmatrix}\rho v\\ \rho uv-\sigma_{xy}\\ \rho v^2+p-\sigma_{yy}\\ v(E\rho+p)-\sigma_{xy}u-\sigma_{yy}v\end{pmatrix}=\mathbf{0}$$

where

$$\sigma_{xx}=\mu\left(2\frac{\partial u}{\partial x}-\frac{2}{3}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)\right)$$
$$\sigma_{yy}=\mu\left(2\frac{\partial v}{\partial y}-\frac{2}{3}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)\right)$$
$$\sigma_{xy}=\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)$$

$$p_{\mathrm{ideal}}=(\gamma-1)U$$
$$p_{\mathrm{vdw}}=\frac{(\gamma-1)U}{1-b_{vdw}\rho}-a_{vdw}\rho^2$$
$$p_{\mathrm{rk}}=\frac{(\gamma-1)U}{1-b_{rk}\rho}-\frac{a_{rk}\rho^{5/2}}{((\gamma-1)U)^{1/2}(1+b_{rk}\rho)}$$

Inlet: $\rho,\ p_t=p\left(1+\frac{1}{5}M^2\right)^{3.5}$ fixed.

Outlet: p fixed.

The sigmoid functions can form the bases for continuous $f : \mathbb{R}^d \to \mathbb{R}$.

- [Mhaskar 92] has shown that: for any integer $s \geq 1$, any compact set $\Omega \subset \mathbb{R}^d$, any continuous function $f : \Omega \to \mathbb{R}$, and any $\epsilon > 0$, there exist an integer N, numbers $\alpha_k, t_k \in \mathbb{R}$ and $\lambda_k \in \mathbb{R}^d$, such that

$$\sup_{x \in \Omega} \left| f(x) - \sum_{k=1}^{N} \alpha_k \phi(\lambda_k \cdot (x - t_k)) \right| < \epsilon$$

- If $v_1, \cdots, v_n$ is a basis for V, $w_1, \cdots, w_m$ is a basis for W, then $\{v_i \otimes w_j\}_{1 \leq i \leq n, 1 \leq j \leq m}$ is a basis for $V \otimes W$.
  The theorem also holds for infinite dimensional V, W.

Accuracy of gradient estimation

- What flow quantity to use to compute $\mathcal{M}$? (Goal-oriented approach?)
- How to quantify $\mathcal{M}$ and estimation error in $\frac{d\xi}{dc}$?
  - Consider a special case

$$\frac{\partial u}{\partial t} + \nabla \cdot F(u) = c\,, \quad \frac{\partial \tilde{u}}{\partial t} + \nabla \cdot \tilde{F}(\tilde{u}) = c \,,$$
$$\tilde{F} = F + \delta F$$

with $u\big|_{t=0} = u_0$ and $\frac{\partial u}{\partial \mathbf{n}}\big|_{\partial\Omega} = 0$.

$$\text{Objective:} \quad \xi = \int_t \int_\Omega g(u) dx dt$$

  - After linearization, the solution error δu and the adjoint error δv satisfy

$$\frac{\partial \delta u}{\partial t} + \nabla \cdot \left(\frac{dF}{du} \delta u \right) = -\nabla \cdot \delta F$$

$$\frac{\partial \delta v}{\partial t} + \frac{dF}{du} \cdot \nabla \delta v = -\frac{d^2 g}{du^2} \delta u - \frac{d\delta F}{du} \cdot \nabla v - \frac{d^2 F}{du^2} \cdot \nabla v \delta u$$

Non-stationary GP modeling

- Non-stationary covariance functions allow GP to adapt to functions whose smoothness varies with the inputs.
-
$$K^{NS}(c_i, c_j) = \int k_{c_i}(c) k_{c_j}(c) dc$$

  For Gaussian kernels [Higdon 99] ,

$$K^{NS}(c_i, c_j) = \sigma^2 |\Sigma_i|^{1/4} |\Sigma_j|^{1/4} \left| \left(\Sigma_i + \Sigma_j \right) / 2 \right|^{-1/2} \exp(-Q_{ij})$$

$$Q_{ij} = (c_i - c_j)^T \left((\Sigma_i + \Sigma_j)/2 \right)^{-1} (c_i - c_j)$$

- The extension of GP to non-stationary introduces additional parameterization that models the variation of the kernel.

Reduce computational cost of GPO

- The search of the next candidate design requires optimizing an acquisition function.
- Requires repetitive evaluation of

$$(\mathbf{v}, \mathbf{w}) \underbrace{\begin{pmatrix} \mathbf{D} & \mathbf{H} \\ \mathbf{H}^T & \mathbf{E} + \mathbf{G} \end{pmatrix}^{-1}}_{\Sigma} (\mathbf{v}, \mathbf{w})^T$$

  for different c. $\Sigma : n(d+1) \times n(d+1)$.
- Cholesky decomposition $\Sigma = \mathbf{L}\mathbf{L}^T$, $\mathcal{O}(n^3(d+1)^3)$ FLOPs.
- The inclusion of new data at $c = c'$ updates $\mathbf{L}$ by

$$\mathbf{L}_{\text{updated}} \leftarrow \begin{pmatrix} \mathbf{L} & \lambda^T \\ \lambda & K(c', c') - \lambda^T \lambda \end{pmatrix}, \quad \lambda = (\mathbf{v}, \mathbf{w})\mathbf{L}^{-T}$$

  Updating the Cholesky decomposition requires $\mathcal{O}(n^2(d+1)^2)$ FLOPs.
- Sparse GP is also applicable (e.g. greedy subset selection [Smola 01], latent variable method [Lawrence 04]), at the cost of reducing accuracy.

Reduce computational cost of twin model

- Instead of minimizing the solution mismatch $\mathcal{M}$, we can minimize

$$\mathcal{R} := \int_t \int_x \left\| \tilde{R}(u) \right\| dxdt \,, \qquad \tilde{R}(u) = \dot{u} + \nabla \cdot \tilde{F}(u) - q(u, c) \,,$$

  thus avoid the integration of twin model PDE.

- Small residual does not guarantee small solution mismatch.
  - Consider the discretized the gray-box and twin models:

$$\begin{aligned} \text{twin model: } \tilde{u}_{t+1} - \mathcal{G}\tilde{u}_t &= 0 \\ \text{gray-box model: } u_{t+1} - \mathcal{H}u_t &= 0 \end{aligned} \,, \quad t = 0, \cdots, T-1$$

$$\tilde{R}(u) \approx \left\| u_1 - \mathcal{G}u_0 \right\|^2 + \cdots \left\| u_T - \mathcal{G}u_{T-1} \right\|^2$$

If $\|\mathcal{G}u - \mathcal{G}u'\| \leq \alpha \|u - u'\|$ for all u, u' with $\alpha < 1$, then $\mathcal{M} \leq \frac{1}{1-\alpha}\mathcal{R}$.

- We first minimize the residual, then minimize the solution mismatch. The procedure reduces the cost of training, and yields good twin model.

Optimization with constraints

- Constraints independent of u,

$$\text{e.g.}\quad \boldsymbol{c}_{\text{lower}} \leq \boldsymbol{c} \leq \boldsymbol{c}_{\text{upper}}$$

  Enforced in optimizing $\rho(c \big| \mathcal{F}_n)$.

- Constraints depends on u
  - Modify the objective function
    e.g. penalty methods[Homaifar 94] , augmented lagrangian methods [Conn 91], barrier function methods [Conn 97].
  - Modify the acquisition
    e.g. expected improvement with constraints (EIC [Gardner 14])

$$\mathbb{E}\left[\max(\xi(c) - \xi(c_n^*), 0) \big| \mathcal{F}_n\right] \mathbb{P}\left[g(c) \leq 0\right]$$

    integrated expected conditional improvement (IECI [Gramacy 11]).

$$\int_{\mathcal{C}} \left[\rho\left(c'\right) - \rho\left(c' \big| c\right)\right] \mathbb{P}[g(c) \leq 0]\, \mathrm{d}c'$$

Analyze computational cost: twin model

n: num of gray-box simulations.
n_b: average num of basis addition / deletion.
$n_{\mathcal{M}}$: average num of twin model simulations to minimize $\mathcal{M}$.
d: $\dim(c)$.
C_T: cost of running the twin model for once.
C_G: cost of running the gray-box model for once.

- Cost for training twin model at each design c
  - Twin model may offer significant benefit if

$$\frac{C_T}{C_G} < \frac{d}{n_b n_{\mathcal{M}}}$$

  - Reuse trained twin model to reduce n_b, $n_{\mathcal{M}}$.
  - Reduce C_T by minimizing the twin model residual's first.

n: num of gray-box simulations.
d: $\dim(c)$.
n_ρ: average num of ρ evalutions for each max ρ.
n_{MLE}: average num of likelihood evalutions for each max MLE.

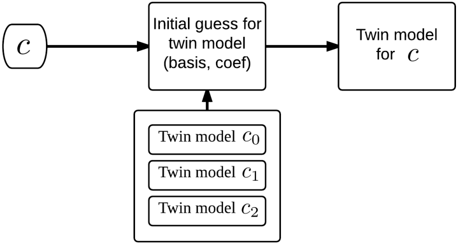
- Cost for MLE and optimizing $\rho(c|\mathcal{F}_n)$ in GPO
  - Cost of likelihood evalution is $\mathcal{O}(n^3 d^3)$.
  - Cost of ρ evaluation is $\mathcal{O}(n^2 d^2)$.
  - Total cost:

$$n\left(\underbrace{n_{MLE}\, n^3 d^3}_{\text{MLE cost}} + \underbrace{n_\rho n^2 d^2}_{\max\ \rho \text{ cost}}\right)$$

  - MLE cost can be controlled by
    - Use full Bayesian approach [Kennedy 01, Snoek 12] (MCMC).
    - Only update hyper-parameters when determined necessary. Future work.
  - Assume GPO cost negligible vs gray-box simulation cost.

Reuse gray-box solutions and trained twin models backup

- Use previously trained twin model as an initial guess.



  - Search for a design in the trained database which is “closest” to c. Prune the model using the backward steps.
  - More efficient reuse of twin models remains a future work.
- Use multiple solutons to train a single twin model.
  - Incorporate multiple solutions to calibrate a model can improve the predictive performance of the calibrated model. [Arendt 12]
  - In the adjoint analysis, however, $\frac{\partial \xi}{\partial c}$ at a design c only involves $u(c)$.

Prove the search sequence is dense

backup

Lemma (Chapter 1, Theorem 4.1, [Berlinet 11])
Let K_1, K_2 be the reproducing kernels of functions on $\mathcal{C}$ with norms $\|\cdot\|_{\mathcal{H}_1}$ and $\|\cdot\|_{\mathcal{H}_2}$ respectively. Then $K = K_1 + K_2$ is the reproducing kernel of the space

$$\mathcal{H} = \mathcal{H}_1 \oplus \mathcal{H}_2 = \{f = f_1 + f_2,\ f_1 \in \mathcal{H}_1,\ f_2 \in \mathcal{H}_2\}$$

with norm $\|\cdot\|_{\mathcal{H}}$ defined by

$$\forall f \in \mathcal{H} \quad \|f\|^2_{\mathcal{H}} = \min_{f=f_1+f_2,\ f_1 \in \mathcal{H}_1, f_2 \in \mathcal{H}_2} \left(\|f_1\|_{\mathcal{H}_1^2} + \|f_2\|^2_{\mathcal{H}_2} \right)$$

Theorem 1 (Cauchy inequality)

Let $\xi \in \mathcal{K}(\mathcal{C})$. $\mathcal{K}$ is the reproducing kernel Hilbert space (RKHS) with the semi-positive definite kernel $K : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$.

Let $\epsilon_i \in \mathcal{H}^i_G$. $\mathcal{H}^i_G$ is the RKHS with the semi-positive definite kernel G_i.

$$\left| \xi(c, \omega_\xi) - \hat{\xi}(c; \underline{c}_n) \right|^2 \leq C\sigma^2(c; \underline{c}_n)$$

$$C = \left(1 + \frac{4d}{3}\right) \left\| \xi(c; \omega_\xi) \right\|_{\mathcal{H}_K} + \frac{4d}{3} \left\| \nabla_c \xi(c; \omega_\xi) \right\|_{\mathcal{H}_{K_\nabla}} + \frac{4}{3} \sum_{i=1}^{d} \left\| \epsilon_i(c; \omega^i_\epsilon) \right\|_{\mathcal{H}^i_G}$$

Prove the search sequence is dense, cont.

Theorem 2 Let $(\underline{c}_n)_{n\geq 1}$ and $(\underline{a}_n)_{n\geq 1}$ be two sequences in $\mathcal{C}$. Assume that the sequence (a_n) is convergent, and denote by a^* its limit. Then each of the following conditions implies the next one:

1. a^* is an adherent point of $\underline{c}_n$ (there exists a subsequence in $\underline{c}_n$ that converges to a^*) ,
2. $\sigma^2(a_n; \underline{c}_n) \to 0$ when $n \to \infty$,
3. $\hat{\xi}(a_n; \underline{c}_n) \to \xi(a^*, \omega)$ when $n \to \infty$, for all $\xi \in \mathcal{H}_K$, $\epsilon \in \mathcal{H}_G$.

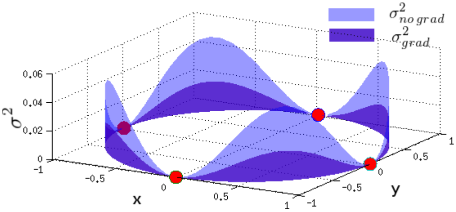
The proof of theorem 2 is similar to the proof of proposition 8 in **[Vazquez 10]**.

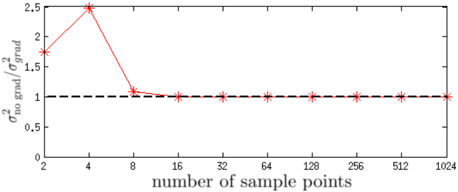
Theorem 3 Under the assumption
There exist $C \geq 0$ and $k \in \mathbb{N}^+$, such that $(1 + |\eta|^2)^k \left|\hat{\Phi}(\eta)\right| \geq C$ for all $\eta \in \mathbb{R}^d$.
We have (E. Vazquez, Theorem 5 **[Vazquez 10]**)

If the 3 conditions in Theorem 2 are equivalent, then for all $c_{init} \in \mathcal{C}$ and all $\omega \in \mathcal{H}$, the sequence $\underline{c}_n$ generated by the GP-EI algorithm is dense in $\mathcal{C}$.

Conjecture: convergence rate

- Only the objective value: GPO converges at rate $n^{-\frac{\nu}{d}}$ for ξ in the RKHS associated with Matern ν kernel. [Bull 11].
- Conjecture: Twin-model GPO converges at the same rate as vanilla GPO
  - 1-D Periodic GP $\xi(c)$, $c \in [-\pi, \pi]$
  - Sample uniformly: (1) ξ (2) ξ and noisy $\frac{d\xi}{dc}$.
  - Posteriors becomes indistinguishable as $n \to \infty$.





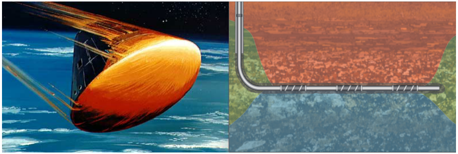
- Twin model boosts optimization in an initial phase of GPO. The boost diminishes as $n \to \infty$.

What if twin model fails?

- Twin model may yield low-quality gradient estimation
  - Gray-box solution is poorly resolved.
  - Wrong conditions used in twin model.
    e.g. wrong I.C. B.C., source term.
  - ...
- We do NOT expect good gradient estimate when assumptions are violated.
- Twin model searches for a model within its model scope to best match the solution.
- The error can be catched by the model discrepancy.
- Worst case scenario: degenerates to derivative-free GPO.

Unknown flux / source / B.C.

- Unknown flux is a common problem:
  - Re-entry vehicle control.
  - Oil reservoir wells / B.C.



- My work is to demonstrate the value of PDE solution in inferring the adjoint.
- Future work to explore the applicability of twin model to unknown source / B.C.

Parallel twin-model GPO

- Parallel Bayesian optimization [Snoek 12] . N evaluations completed, $\{c_i, \xi_i, \xi_{\tilde{\nabla}}\}_{i=1}^{N}$.
  Running gray-box and twin model on J processes, pending data $\{c_j, \xi_j, \xi_{\tilde{\nabla}}\}_{j=1}^{J}$.
  Expected acquisition:

$$\rho(c; \{c_i, \xi_i, \xi_{\tilde{\nabla}}\}_{i=1}^{N}, \{c_j\}_{j=1}^{J}) = \int \rho(c; \{c_i, \xi_i, \xi_{\tilde{\nabla}}\}_{i=1}^{N}, \{c_j, \xi_j, \xi_{\tilde{\nabla}}\}_{j=1}^{J}) d\boldsymbol{\xi} d\boldsymbol{\xi}_{\tilde{\nabla}}$$

- Parallel twin model:
  Twin model is a conservation law simulator that may be parallelled.

References I

RC. Eberhart et al.
Particle swarm optimization.
Encyclopedia of machine learning, 2010.

L. Davis
Handbook of genetic algorithms.
New York: Van Nostrand Reinhold, 1991.

K. Tamara et al.
Optimization by direct search: new perspective on some classical and modern methods.
SIAM review, 45.3:385-482, 2003.

D. Dennis et al.
Quasi-Newton methods, motivation and theory.
SIAM review, 19.1:46-89, 1977.

References II

R. Adler
Comparison of basis selection methods.
Signals, systems and computers, Thirtieth Asilomar Conference on. Vol. 1. IEEE, 1996.

S. Billing et al.
Feature subset selection and ranking for data dimensionality reduction.
Pattern analysis and machine intelligence, IEEE transactions on 29.1 (2007): 162-166.

G. Schwarz
Estimating the dimension of a model.
The annals of statistics 6.2 (1978): 461-464.

References III

M. Stone
An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion.
Journal of the royal statistical society. series B (methodological) (1977): 44-47.

R. Tibshirani
Regression shrinkage and selection via the lasso.
Journal of the royal statistical society. series B (methodological) (1996): 267-288.

I. Kucuk
An efficient computational method for the optimal control problem for the Burgers equation.
Mathematical and computer modelling 44.11 (2006): 973-982.

References IV

A. Booker
A rigorous framework for optimization of expensive functions by surrogates.
Structural optimization 17.1 (1999): 1-13.

S. Wild et al.
Global convergence of radial basis function trust-region algorithms for derivative-free optimization.
SIAM Review 55.2 (2013): 349-371.

A. March
Multifidelity methods for multidisciplinary system design
Dissertation, Massachusetts Institute of Technology (2012)

T. Robinson
Multifidelity optimization for variablecomplexity design.
Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Portsmouth, VA. 2006.

M. Kennedy et al.
Bayesian calibration of computer models
Journal of the royal statistical society, series B (statistical methodology) 63.3 (2001): 425-464.

D.Higdon et al.
Combining field data and computer simulations for calibration and prediction
SIAM Journal of Scientific computations (USA), 26.2, (2004): 448-466.

References VI

A. March et al.
Gradient-based multifidelity optimisation for aircraft design using Bayesian model calibration
Aeronautical Journal, 115.1174 (2011): 729.

H. Chung et al.
Using gradients to construct cokriging approximation models for high-dimensional design optimization problems.
AIAA paper 317 (2002): 14-17.

J. Snoek et al.
Practical Bayesian optimization of machine learning algorithms.
Advances in neural information processing systems, 2012.

J. Mockus et al.
The application of Bayesian methods for seeking the extreme
Towards Global Optimization, 2 (1978): 117-229

References VII

N. Srinivas et al.
Gaussian process optimization in the bandit setting: No regret and experiment design
arXiv preprint arXiv:0912.3995 (2009).

F. Coletti et al.
Optimization of a U-Bend for Minimal Pressure Loss in Internal Cooling Channels-Part II: Experimental Validation.
Journal of Turbomachinery 135.5 (2013): 051016.

H. Chen et al.
Adjoint-based gradient estimation for gray-box solutions of unknown conservation laws
Submitted to Journal of Computational Physics.

References VIII

D. Wilcox
Turbulence modeling for CFD.
Vol. 2. La Canada, CA: DCW industries, (1998)

J. Gardner
Bayesian optimization with inequality constraints
Proceedings of The 31st International Conference on Machine Learning. 2014.

H. Chen
Adjoint-based gradient estimation from gray-box solutions of unknown conservation laws
arXiv preprint arXiv:1511.04576 (2015).

S.E. Buckley
Mechanism of fluid displacement in sands.
Transactions of the AIME 146 (1942): 107-116

References IX

S.G. Mallat
A theory for multiresolution signal decomposition: the wavelet representation.
Pattern Analysis and Machine Intelligence, IEEE Transactions on 11.7 (1989): 674-693

A.J. Miller
Subset selection in regression
London: Chapmen and Hall press, 1990

H.N. Mhaskar et al.
Approximation by superposition of sigmoidal and radial basis functions.
Advances in Applied mathematics 13.3 (1992): 350-373.

M.J. Bayarri et al.
A framework for validation of computer models
Technometrics 49.2 (2007): 138-154.

D.R. Jones et al.
Efficient global optimization of expensive black-box functions
Journal of Global Optimization 13.4 (1998): 455-492.

D. Higdon et al.
Non-stationary spatial modeling
Bayesian Statistics 6 (1999): 761-768. Oxford University Press.

A.J. Smola et al.
Sparse greedy Gaussian process regression
Advances in Neural Information Processing Systems 13. 2001.

References XI

N. Lawrence et al.
Gaussian process latent variable models for visualization of high dimensional data
Advances in Neural Information Processing Systems 16.3 (2004): 329-336

J.H. Friedman et al.
An overview of predictive learning and function approximation
From Statistics to Neural Networks, Springer, NY, 1994

R. Reed
Pruning algorithm - a survey
IEEE Transactions on Neural Networks, 4.5 (1993): 740-747

References XII

G. Jekabsons
Adaptive basis function construction: an approach for adaptive building of sparse polynomial regression models.
INTECH Open Access Publiser, 2010.

S.G. Mallat et al.
Matching pursuits with time-frequency dictionaries
IEEE Transactions on Signal Processing, 41.12 (1993): 3397-3415

G. Blatman et al.
An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis.
Probabilistic Engineering Mechanics, 25.2 (2010): 183-197

B. Matern
Spatial Variation
Springer, New York, 1960

References XIII

S. Gudmundsson
Parallel global optimization
Ms. Thesis, IMM, Technical University of Denmark (1998)

E. Vazquez et al.
Convergence properties of the expected improvement algorithm with fixed mean and covariance functions
Journal of Statistical Planning and inference 140.11 (2010): 3088-3095.

A. Berlinet
Reproducing kernel Hilbert spaces in probability and statistics.
Springer Science and Business Media, 2011.

References XIV

P.D. Arendt et al.
Improving identifiability in model calibration using multiple responses.
Journal of Mechanical Design 134.10 (2012): 100909

A.R. Conn et al.
A globally convergent Lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds
Mathematics of Computation of the American Mathematical Society 66.217 (1997): 261-288

A. Homaifar et al.
Constrained optimization via genetic algorithms.
Simulation 62.4 (1994): 242-253.

References XV

A.R. Conn et al.
A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds.
SIAM Journal on Numerical Analysis 28.2 (1991): 545-572.

R.B. Gramacy et al.
Optimization under unknown constraints.
Bayesian Statistics, 9. (2011)

A.D. Bull
Convergence rates of efficient global optimization algorithms.
Journal of Machine Learning Research, 12 (2011) 2879-2904

O. Redlich
On the thermodynamics of solutions
Chemical Reviews, 44.1 (1949) 233-244.

References XVI

J.P. Lions
Optimal control of systems governed by partial differential equations
Springer-Verlag, New York.

S. Geisser
Predictive inference
New York, NY: Champman and Hall, 1993.