

Derivative-free optimization under uncertainty using multifidelity sampling

Friedrich Menhorn¹, Florian Augustin^{2,3}, Youssef M. Marzouk², Gianluca Geraci⁴, Michael S. Eldred⁴, Hans-Joachim Bungartz¹

menhorn@in.tum.de

¹Technical University of Munich, GER

²Massachusetts Institute of Technology, USA

³Mathworks, USA

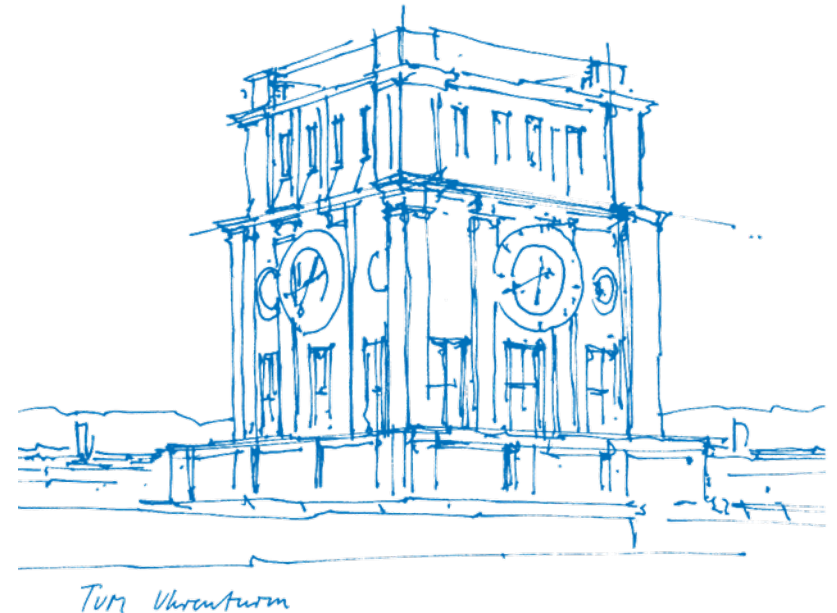
⁴Sandia National Laboratories, USA

ICIAM

MS GH-0-2 7:

MF methods for UQ and OPT in complex systems

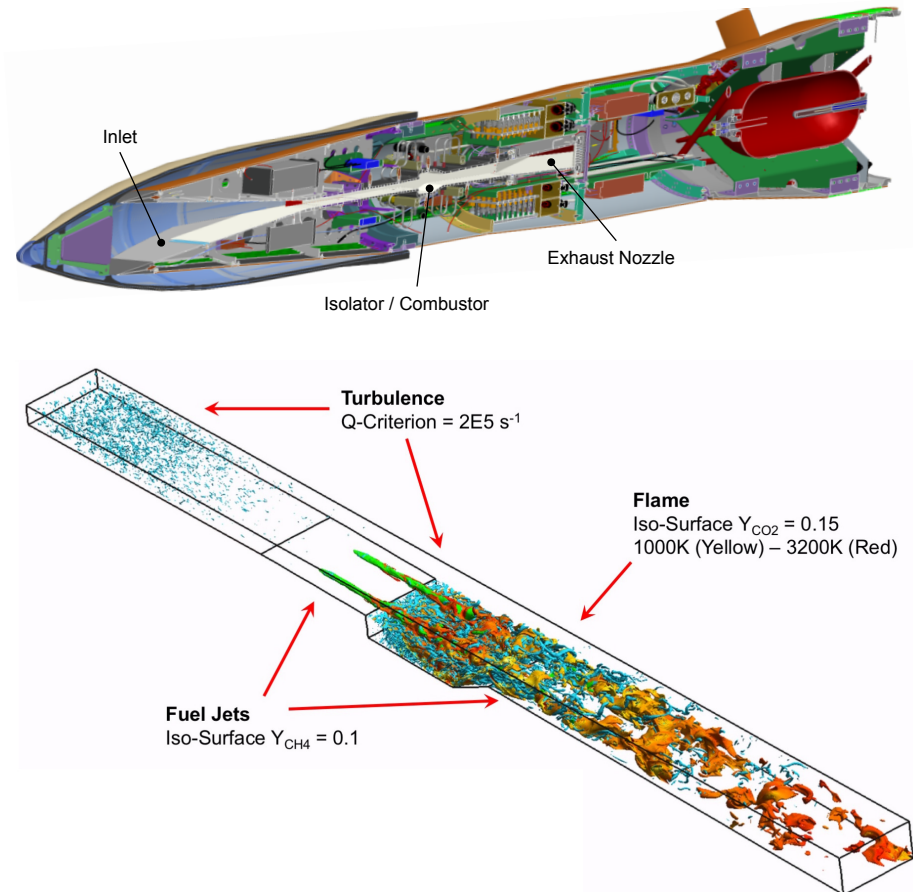
Valencia, July 17th, 2019



Motivation: Design optimization of a SCRAMJET

Provided by Sandia National Laboratories

- No rotating elements for compression
 - Air compressed dynamically
 - **Supersonic** mixture and combustion
 - (Some) challenges:
 - Low throughput time
- vs.
- mixture and self-ignition
 - Compressibility effects
 - Stable combustion for constant thrust
- [Javier Urzay, 2018]:
The challenge of enterprising supersonic combustion in scramjet is [...] as difficult as lighting a match in a hurricane.



SNOWPAC



SNOWPAC ¹

Robust optimization problem statement

- Find **robust** solution with respect to uncertainty
- Using measures of robustness \mathcal{R} , e.g. \mathbb{E} , \mathbb{V} , CVaR.
- E.g., weigh expected gain vs. confidence: $\max \mathbb{E} - \lambda \mathbb{V}^{\frac{1}{2}}$

$$\mathcal{R}_{\omega}^* = \mathcal{R}(\mathbf{x}^*, \omega) = \min_{\mathcal{R}^c(\mathbf{x}, \omega) \leq 0} \mathcal{R}^f(\mathbf{x}, \omega)$$

¹F. Augustin, Y. Marzouk, A trust-region method for derivative-free nonlinear constrained stochastic optimization. 2017
Friedrich Menhorn (TUM), et al. | menhorn@in.tum.de | Derivative-free OUU using MF sampling

SNOWPAC ¹

Robust optimization problem statement

$$\mathcal{R}_\omega^* = \mathcal{R}(\mathbf{x}^*, \omega) = \min_{\mathcal{R}^c(\mathbf{x}, \omega) \leq 0} \mathcal{R}^f(\mathbf{x}, \omega)$$

¹F. Augustin, Y. Marzouk, A trust-region method for derivative-free nonlinear constrained stochastic optimization. 2017
Friedrich Menhorn (TUM), et al. | menhorn@in.tum.de | Derivative-free OUU using MF sampling

SNOWPAC ¹

Robust optimization problem statement

$$\mathcal{R}_\omega^* = \mathcal{R}(\mathbf{x}^*, \omega) = \min_{\mathcal{R}^c(\mathbf{x}, \omega) \leq 0} \mathcal{R}^f(\mathbf{x}, \omega)$$

Features of SNOWPAC:

¹F. Augustin, Y. Marzouk, A trust-region method for derivative-free nonlinear constrained stochastic optimization. 2017
Friedrich Menhorn (TUM), et al. | menhorn@in.tum.de | Derivative-free OUU using MF sampling

SNOWPAC ¹

Robust optimization problem statement

$$\mathcal{R}_\omega^* = \mathcal{R}(\mathbf{x}^*, \omega) = \min_{\mathcal{R}^c(\mathbf{x}, \omega) \leq 0} \mathcal{R}^f(\mathbf{x}, \omega)$$

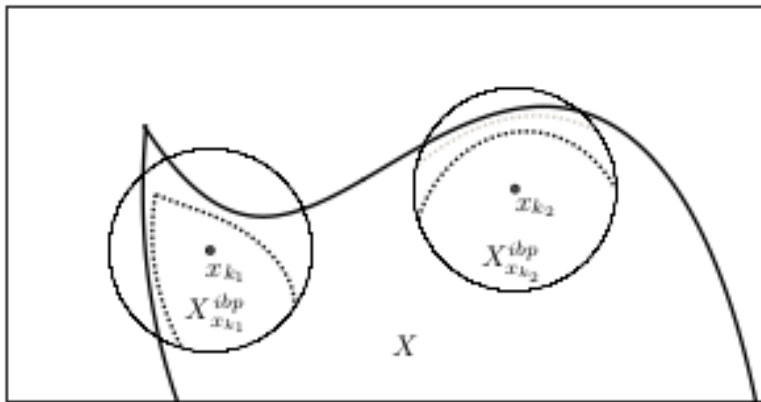
Features of SNOWPAC:

0. **Extension of NOWPAC:** Derivative-free nonlinear constraint optimization method using trust-regions (deterministic)

¹F. Augustin, Y. Marzouk, A trust-region method for derivative-free nonlinear constrained stochastic optimization. 2017
Friedrich Menhorn (TUM), et al. | menhorn@in.tum.de | Derivative-free OUU using MF sampling

Derivative-free optimization using NOWPAC ²

- **Non-intrusive optimization** framework
- **Trust region approach** for nonlinearly-constrained DFO
- Build **fully linear surrogate models** of objective and constraints
- Find improved designs by **minimizing surrogate models**



Inner Boundary Convexification

- New way of **handling constraints using an inner boundary path**
 - The inner boundary path is an additive convex function to the constraints
- Global **convergence to a first-order locally optimal design**

²F. Augustin, Y. Marzouk, NOWPAC: A path-augmented constraint handling approach for nonlinear derivative-free optimization.

SNOWPAC ¹

Robust optimization problem statement

$$\mathcal{R}_\omega^* = \mathcal{R}^f(\mathbf{x}^*, \omega) = \min_{\mathcal{R}^c(\mathbf{x}, \omega) \leq 0} \mathcal{R}^f(\mathbf{x}, \omega)$$

Features of SNOWPAC:

0. **Extension of NOWPAC:** Derivative-free nonlinear constraint optimization method using trust-regions (deterministic)
1. **Estimate robustness measures:** Use sampling, e.g. $\mathcal{R}_\omega^f = \mathbb{E}[f_\omega(\mathbf{x})] \approx R^f = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}, \theta_i) + \varepsilon_N$

¹F. Augustin, Y. Marzouk, A trust-region method for derivative-free nonlinear constrained stochastic optimization. 2017 Friedrich Menhorn (TUM), et al. | menhorn@in.tum.de | Derivative-free OUU using MF sampling

SNOWPAC ¹

Robust optimization problem statement

$$\mathcal{R}_\omega^* = \mathcal{R}^f(\mathbf{x}^*, \omega) = \min_{\mathcal{R}^c(\mathbf{x}, \omega) \leq 0} \mathcal{R}^f(\mathbf{x}, \omega)$$

Features of SNOWPAC:

0. **Extension of NOWPAC:** Derivative-free nonlinear constraint optimization method using trust-regions (deterministic)
1. **Estimate robustness measures:** Use sampling, e.g. $\mathcal{R}_\omega^f = \mathbb{E}[f_\omega(\mathbf{x})] \approx R^f = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}, \theta_i) + \varepsilon_N$
NEW: Leverage multilevel estimators.

¹F. Augustin, Y. Marzouk, A trust-region method for derivative-free nonlinear constrained stochastic optimization. 2017 Friedrich Menhorn (TUM), et al. | menhorn@in.tum.de | Derivative-free OUU using MF sampling

Multilevel error estimator for SNOWPAC

Generic MLMC estimators:

- Mean:

$$\mathbb{E}[Q] \approx \hat{Q}_L = \sum_{\ell=0}^L \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} (Q_\ell^{(i)} - Q_{\ell-1}^{(i)})$$

- Variance:

$$\begin{aligned} \mathbb{V}[Q] &= \sum_{\ell=0}^L \mathbb{E} \left[(Q_\ell - \mathbb{E}[Q_\ell])^2 - (Q_{\ell-1} - \mathbb{E}[Q_{\ell-1}])^2 \right] = \sum_{\ell=0}^L \mathbb{E}[P_\ell^2] - \mathbb{E}[P_{\ell-1}^2] \\ &\approx s_{ML}^2 = \sum_{\ell=0}^L (\widehat{P}_\ell^2 - \widehat{P}_{\ell-1}^2), \quad \text{where} \quad \widehat{P}_\ell^2 = \frac{1}{N_\ell - 1} \sum_{i=1}^{N_\ell} (Q_\ell^{(i)} - \hat{Q}_\ell)^2. \end{aligned}$$

- Standard deviation:

$$\sqrt{\mathbb{V}[Q]} \approx \sqrt{s_{ML}^2}$$

Multilevel error estimator for SNOWPAC

Generic MLMC estimators:

- Mean:

$$\mathbb{E}[Q] \approx \hat{Q}_L = \sum_{\ell=0}^L \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} (Q_\ell^{(i)} - Q_{\ell-1}^{(i)})$$

- Variance:

$$\begin{aligned} \mathbb{V}[Q] &= \sum_{\ell=0}^L \mathbb{E} \left[(Q_\ell - \mathbb{E}[Q_\ell])^2 - (Q_{\ell-1} - \mathbb{E}[Q_{\ell-1}])^2 \right] = \sum_{\ell=0}^L \mathbb{E}[P_\ell^2] - \mathbb{E}[P_{\ell-1}^2] \\ &\approx s_{ML}^2 = \sum_{\ell=0}^L (\widehat{P}_\ell^2 - \widehat{P}_{\ell-1}^2), \quad \text{where} \quad \widehat{P}_\ell^2 = \frac{1}{N_\ell - 1} \sum_{i=1}^{N_\ell} (Q_\ell^{(i)} - \hat{Q}_\ell)^2. \end{aligned}$$

- Standard deviation:

$$\sqrt{\mathbb{V}[Q]} \approx \sqrt{s_{ML}^2}$$

⇒ Multilevel estimators for \mathbb{E} , \mathbb{V} and $\sqrt{\mathbb{V}}$.

Multilevel error estimator for SNOWPAC

Generic MLMC **error** estimators:

- Mean \hat{Q}_L :

$$\mathbb{V}[\hat{Q}_L] = \sum_{\ell=0}^L \frac{1}{N_\ell} \mathbb{V}[\hat{Y}_L]$$

- Variance s_{ML}^2 :

$$\mathbb{V}[s_{ML}^2] = \sum_{\ell=0}^L \mathbb{V}[\widehat{P}_\ell^2] + \mathbb{V}[\widehat{P}_{\ell-1}^2] - 2\text{Cov}(\widehat{P}_\ell^2, \widehat{P}_{\ell-1}^2)$$

where $\mathbb{V}[\widehat{P}_\ell^2] = \frac{1}{N_\ell} (\mu_{4,\ell} - \mathbb{V}[Q_\ell]^2) + \frac{2}{N_\ell(N_\ell - 1)} \mathbb{V}[Q_\ell]^2$

- Standard deviation $\sqrt{s_{ML}^2}$:

$$SE(s_{ML}^2) \approx \frac{1}{2\sqrt{s_{ML}^2}} \sqrt{\mathbb{V}[s_{ML}^2]}$$

Multilevel error estimator for SNOWPAC

Generic MLMC **error** estimators:

- Mean \hat{Q}_L :

$$\mathbb{V}[\hat{Q}_L] = \sum_{\ell=0}^L \frac{1}{N_\ell} \mathbb{V}[\hat{Y}_L] = \left(\frac{\epsilon_N}{2}\right)^2$$

- Variance s_{ML}^2 :

$$\mathbb{V}[s_{ML}^2] = \sum_{\ell=0}^L \mathbb{V}[\widehat{P}_\ell^2] + \mathbb{V}[\widehat{P}_{\ell-1}^2] - 2\text{Cov}(\widehat{P}_\ell^2, \widehat{P}_{\ell-1}^2) = \left(\frac{\epsilon_N}{2}\right)^2$$

$$\text{where } \mathbb{V}[\widehat{P}_\ell^2] = \frac{1}{N_\ell} (\mu_{4,\ell} - \mathbb{V}[Q_\ell]^2) + \frac{2}{N_\ell(N_\ell - 1)} \mathbb{V}[Q_\ell]^2$$

- Standard deviation $\sqrt{s_{ML}^2}$:

$$SE(s_{ML}^2) \approx \frac{1}{2\sqrt{s_{ML}^2}} \sqrt{\mathbb{V}[s_{ML}^2]} = \frac{\epsilon_N}{2}$$

\Rightarrow Multilevel error estimators for \mathbb{E} , \mathbb{V} and $\sqrt{\mathbb{V}}$.

SNOWPAC ¹

Robust optimization problem statement

$$\mathcal{R}_\omega^* = \mathcal{R}^f(\mathbf{x}^*, \omega) = \min_{\mathcal{R}^c(\mathbf{x}, \omega) \leq 0} \mathcal{R}^f(\mathbf{x}, \omega)$$

Features of SNOWPAC:

0. **Extension of NOWPAC:** Derivative-free nonlinear constraint optimization method using trust-regions (deterministic)
1. **Estimate robustness measures:** Use sampling, e.g. $\mathcal{R}_\omega^f = \mathbb{E}[f_\omega(\mathbf{x})] \approx R^f = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}, \theta_i) + \varepsilon_N$
2. **Implement new trust region management:** Account for noise ε_N in objective/constraint evaluations $\Rightarrow \Delta_{k+1} \geq \sqrt{\lambda_t \varepsilon_N}$

¹F. Augustin, Y. Marzouk, A trust-region method for derivative-free nonlinear constrained stochastic optimization. 2017 Friedrich Menhorn (TUM), et al. | menhorn@in.tum.de | Derivative-free OUU using MF sampling

SNOWPAC ¹

Robust optimization problem statement

$$\mathcal{R}_\omega^* = \mathcal{R}^f(\mathbf{x}^*, \omega) = \min_{\mathcal{R}^c(\mathbf{x}, \omega) \leq 0} \mathcal{R}^f(\mathbf{x}, \omega)$$

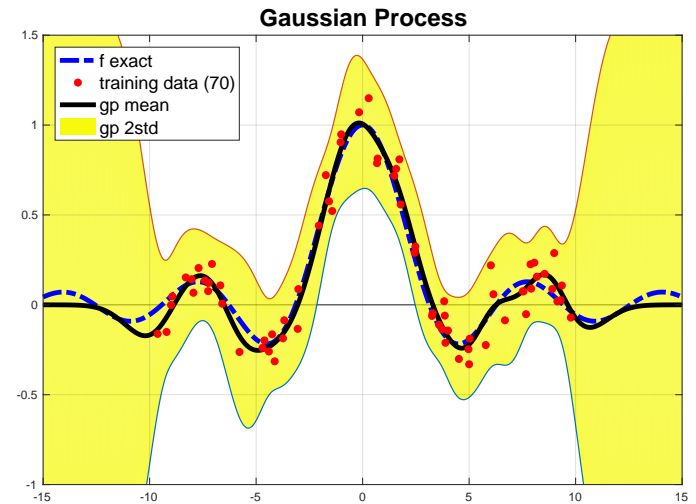
Features of SNOWPAC:

0. **Extension of NOWPAC:** Derivative-free nonlinear constraint optimization method using trust-regions (deterministic)
1. **Estimate robustness measures:** Use sampling, e.g. $\mathcal{R}_\omega^f = \mathbb{E}[f_\omega(\mathbf{x})] \approx R^f = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}, \theta_i) + \varepsilon_N$
2. **Implement new trust region management:** Account for noise ε_N in objective/constraint evaluations $\Rightarrow \Delta_{k+1} \geq \sqrt{\lambda_t \varepsilon_N}$
3. **Introduce Gaussian process surrogates:** Mitigate effect of noise ε_N

¹F. Augustin, Y. Marzouk, A trust-region method for derivative-free nonlinear constrained stochastic optimization. 2017
Friedrich Menhorn (TUM), et al. | menhorn@in.tum.de | Derivative-free OUU using MF sampling

SNOWPAC – Gaussian process surrogate

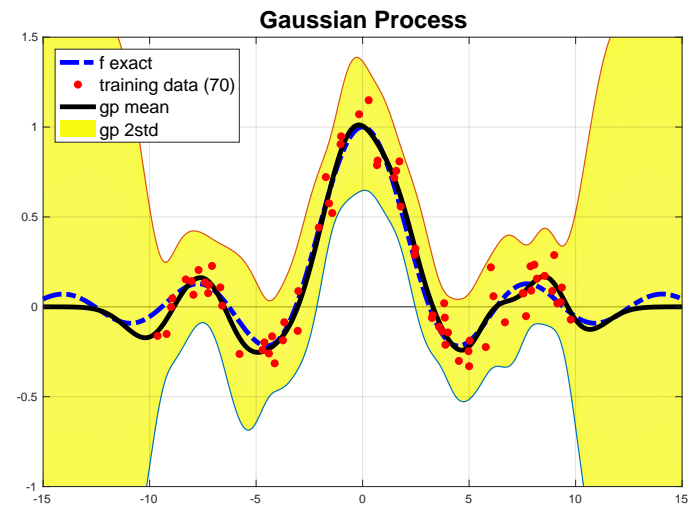
- **Use black box evaluations** to build global Gaussian process surrogates
- **Probabilistic estimator:**
 - GP mean: $\mu_{\text{GP}}(\mathbf{x}) = \mathbf{k}_{\text{xx}}[\mathbf{K}_{\text{xx}} + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{R}$
 - GP variance: $\sigma_{\text{GP}}^2(\mathbf{x}) = k_{\text{xx}} - \mathbf{k}_{\text{xx}}[\mathbf{K}_{\text{xx}} + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{k}_{\text{xx}}$



SNOWPAC – Gaussian process surrogate

- **Use black box evaluations** to build global Gaussian process surrogates
- **Probabilistic estimator:**
 - GP mean: $\mu_{\text{GP}}(\mathbf{x}) = \mathbf{k}_{\text{xx}}[\mathbf{K}_{\text{XX}} + \sigma_n^2\mathbf{I}]^{-1}\mathbf{R}$
 - GP variance: $\sigma_{\text{GP}}^2(\mathbf{x}) = k_{\text{xx}} - \mathbf{k}_{\text{xx}}[\mathbf{K}_{\text{XX}} + \sigma_n^2\mathbf{I}]^{-1}\mathbf{k}_{\text{xx}}$
- **Replace noisy black box evaluations by GP mean:**

$$\tilde{R} = \alpha \cdot \mu_{\text{GP}} + (1 - \alpha) \cdot R_\omega$$



SNOWPAC – Gaussian process surrogate

- **Use black box evaluations** to build global Gaussian process surrogates

- **Probabilistic estimator:**

– GP mean: $\mu_{\text{GP}}(\mathbf{x}) = \mathbf{k}_{\text{xx}}[\mathbf{K}_{\text{XX}} + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{R}$

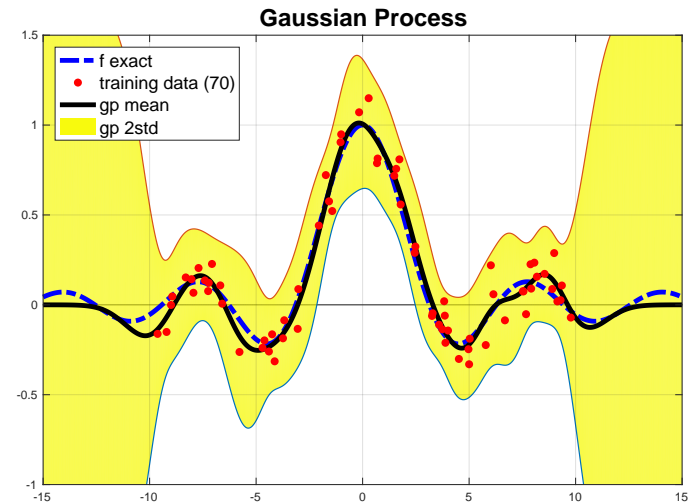
– GP variance: $\sigma_{\text{GP}}^2(\mathbf{x}) = k_{\text{xx}} - \mathbf{k}_{\text{xx}}[\mathbf{K}_{\text{XX}} + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{k}_{\text{xx}}$

- **Replace noisy black box evaluations by GP mean:**

$$\tilde{R} = \alpha \cdot \mu_{\text{GP}} + (1 - \alpha) \cdot R_\omega$$

- **Replace noise estimate by:**

b) Heuristic: $\tilde{\varepsilon} = \alpha \cdot 2\sigma_{\text{GP}}(\mathbf{x}) + (1 - \alpha) \cdot \varepsilon_N$, where $\alpha = e^{-\sqrt{\sigma_{\text{GP}}^2(\mathbf{x})}}$



SNOWPAC – Gaussian process surrogate

- **Use black box evaluations** to build global Gaussian process surrogates

- **Probabilistic estimator:**

– GP mean: $\mu_{\text{GP}}(\mathbf{x}) = \mathbf{k}_{\text{xx}}[\mathbf{K}_{\text{XX}} + \sigma_n^2\mathbf{I}]^{-1}\mathbf{R}$

– GP variance: $\sigma_{\text{GP}}^2(\mathbf{x}) = k_{\text{xx}} - \mathbf{k}_{\text{xx}}[\mathbf{K}_{\text{XX}} + \sigma_n^2\mathbf{I}]^{-1}\mathbf{k}_{\text{xx}}$

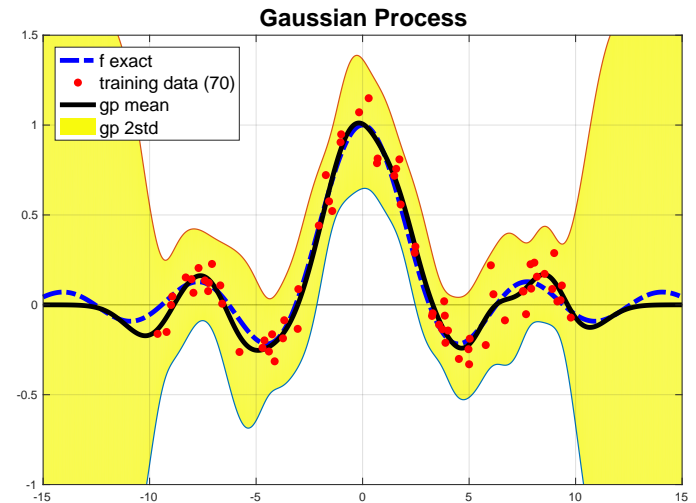
- **Replace noisy black box evaluations by GP mean:**

$$\tilde{R} = \alpha \cdot \mu_{\text{GP}} + (1 - \alpha) \cdot R_\omega$$

- **Replace noise estimate by:**

NEW a) Analytic: $\tilde{\varepsilon} = 2 \cdot \min_{\alpha} \text{RMSE}(\tilde{R})$, where $\alpha = \arg \min_{\alpha} \text{RMSE}(\tilde{R})$ **NEW**

b) Heuristic: $\tilde{\varepsilon} = \alpha \cdot 2\sigma_{\text{GP}}(\mathbf{x}) + (1 - \alpha) \cdot \varepsilon_N$, where $\alpha = e^{-\sqrt{\sigma_{\text{GP}}^2(\mathbf{x})}}$



SNOWPAC – Gaussian Process Noise Correction

MSE:

$$\begin{aligned} \text{MSE}_\alpha &= \text{BIAS}(\tilde{R})^2 + \mathbb{V}[\tilde{R}] \\ &= [\alpha(\mu_{GP}[\mathcal{R}] - \mathcal{R}_\omega)]^2 + \alpha^2 \mathbb{V}[\mu_{GP}] + (1 - \alpha)^2 \mathbb{V}[R] + \alpha(1 - \alpha) 2 \text{cov}[\mu_{GP}, R] \end{aligned}$$

SNOWPAC – Gaussian Process Noise Correction

MSE:

$$\begin{aligned} \text{MSE}_\alpha &= \text{BIAS}(\tilde{R})^2 + \mathbb{V}[\tilde{R}] \\ &= [\alpha(\mu_{GP}[\mathcal{R}] - \mathcal{R}_\omega)]^2 + \alpha^2 \mathbb{V}[\mu_{GP}] + (1 - \alpha)^2 \mathbb{V}[R] + \alpha(1 - \alpha) 2\text{cov}[\mu_{GP}, R] \end{aligned}$$

Optimal α :

$$\alpha^* = \frac{\mathbb{V}[R] - \text{cov}[\mu_{GP}, R]}{(\mu_{GP}[\mathcal{R}] - \mathcal{R}_\omega)^2 + \mathbb{V}[\mu_{GP}] + \mathbb{V}[R] - 2\text{cov}[\mu_{GP}, R]}$$

SNOWPAC – Gaussian Process Noise Correction

MSE:

$$\begin{aligned} \text{MSE}_{\alpha} &= \text{BIAS}(\tilde{R})^2 + \mathbb{V}[\tilde{R}] \\ &= [\alpha(\mu_{GP}[\mathcal{R}] - \mathcal{R}_{\omega})]^2 + \alpha^2 \mathbb{V}[\mu_{GP}] + (1 - \alpha)^2 \mathbb{V}[R] + \alpha(1 - \alpha) 2 \text{cov}[\mu_{GP}, R] \end{aligned}$$

Optimal α :

$$\alpha^* = \frac{\mathbb{V}[R] - \text{cov}[\mu_{GP}, R]}{(\mu_{GP}[\mathcal{R}] - \mathcal{R}_{\omega})^2 + \mathbb{V}[\mu_{GP}] + \mathbb{V}[R] - 2 \text{cov}[\mu_{GP}, R]}$$

Optimal estimator:

$$\begin{aligned} \tilde{R} &= \alpha^* \cdot \mu_{GP} + (1 - \alpha^*) \cdot R_{\omega} \\ \tilde{\epsilon} &= 2 \cdot \sqrt{\text{MSE}_{\alpha^*}(\tilde{R})} \end{aligned}$$

SNOWPAC – Gaussian Process Noise Correction

MSE:

$$\begin{aligned}\text{MSE}_{\alpha} &= \text{BIAS}(\tilde{R})^2 + \mathbb{V}[\tilde{R}] \\ &= [\alpha(\mu_{GP}[\mathcal{R}] - \mathcal{R}_{\omega})]^2 + \alpha^2 \mathbb{V}[\mu_{GP}] + (1 - \alpha)^2 \mathbb{V}[R] + \alpha(1 - \alpha) 2\text{cov}[\mu_{GP}, R]\end{aligned}$$

Optimal α :

$$\alpha^* = \frac{\mathbb{V}[R] - \text{cov}[\mu_{GP}, R]}{(\mu_{GP}[\mathcal{R}] - \mathcal{R}_{\omega})^2 + \mathbb{V}[\mu_{GP}] + \mathbb{V}[R] - 2\text{cov}[\mu_{GP}, R]}$$

Optimal estimator:

$$\begin{aligned}\tilde{R} &= \alpha^* \cdot \mu_{GP} + (1 - \alpha^*) \cdot R_{\omega} \\ \tilde{\varepsilon} &= 2 \cdot \sqrt{\text{MSE}_{\alpha^*}(\tilde{R})}\end{aligned}$$

Approximations:

- $\mathbb{V}[R] = \left(\frac{\varepsilon_R}{2}\right)^2$

SNOWPAC – Gaussian Process Noise Correction

MSE:

$$\begin{aligned}\text{MSE}_\alpha &= \text{BIAS}(\tilde{R})^2 + \mathbb{V}[\tilde{R}] \\ &= [\alpha(\mu_{GP}[\mathcal{R}] - \mathcal{R}_\omega)]^2 + \alpha^2 \mathbb{V}[\mu_{GP}] + (1 - \alpha)^2 \mathbb{V}[R] + \alpha(1 - \alpha) 2\text{cov}[\mu_{GP}, R]\end{aligned}$$

Optimal α :

$$\alpha^* = \frac{\mathbb{V}[R] - \text{cov}[\mu_{GP}, R]}{(\mu_{GP}[\mathcal{R}] - \mathcal{R}_\omega)^2 + \mathbb{V}[\mu_{GP}] + \mathbb{V}[R] - 2\text{cov}[\mu_{GP}, R]}$$

Optimal estimator:

$$\begin{aligned}\tilde{R} &= \alpha^* \cdot \mu_{GP} + (1 - \alpha^*) \cdot R_\omega \\ \tilde{\varepsilon} &= 2 \cdot \sqrt{\text{MSE}_{\alpha^*}(\tilde{R})}\end{aligned}$$

Approximations:

- $\mathbb{V}[R] = \left(\frac{\varepsilon_R}{2}\right)^2$
- $\mathbb{V}[\mu_{GP}] = \mathbf{k}_{\mathbf{xx}}[\mathbf{K}_{\mathbf{xx}} + \sigma_n^2 \mathbf{I}]^{-1} \left(\frac{\varepsilon_R}{2}\right)^2$

SNOWPAC – Gaussian Process Noise Correction

MSE:

$$\begin{aligned}\text{MSE}_\alpha &= \text{BIAS}(\tilde{R})^2 + \mathbb{V}[\tilde{R}] \\ &= [\alpha(\mu_{GP}[\mathcal{R}] - \mathcal{R}_\omega)]^2 + \alpha^2 \mathbb{V}[\mu_{GP}] + (1 - \alpha)^2 \mathbb{V}[R] + \alpha(1 - \alpha) 2 \text{cov}[\mu_{GP}, R]\end{aligned}$$

Optimal α :

$$\alpha^* = \frac{\mathbb{V}[R] - \text{cov}[\mu_{GP}, R]}{(\mu_{GP}[\mathcal{R}] - \mathcal{R}_\omega)^2 + \mathbb{V}[\mu_{GP}] + \mathbb{V}[R] - 2 \text{cov}[\mu_{GP}, R]}$$

Optimal estimator:

$$\begin{aligned}\tilde{R} &= \alpha^* \cdot \mu_{GP} + (1 - \alpha^*) \cdot R_\omega \\ \tilde{\epsilon} &= 2 \cdot \sqrt{\text{MSE}_{\alpha^*}(\tilde{R})}\end{aligned}$$

Approximations:

- $\mathbb{V}[R] = \left(\frac{\epsilon_R}{2}\right)^2$
- $\text{cov}[\mu_{GP}, R] = \left(\frac{\epsilon_R}{2}\right)^2 \sum_{i=1}^N \mathbf{k}_{\mathbf{x}^* \mathbf{x}_i} (\mathbf{k}_{\mathbf{x}_i \mathbf{x}_N} + \delta_{iN} \sigma_i^2)^{-1}$
- $\mathbb{V}[\mu_{GP}] = \mathbf{k}_{\mathbf{x}\mathbf{x}} [\mathbf{K}_{\mathbf{X}\mathbf{X}} + \sigma_n^2 \mathbf{I}]^{-1} \left(\frac{\epsilon_R}{2}\right)^2$

SNOWPAC – Gaussian Process Noise Correction

MSE:

$$\begin{aligned} \text{MSE}_\alpha &= \text{BIAS}(\tilde{R})^2 + \mathbb{V}[\tilde{R}] \\ &= [\alpha(\mu_{GP}[\mathcal{R}] - \mathcal{R}_\omega)]^2 + \alpha^2 \mathbb{V}[\mu_{GP}] + (1 - \alpha)^2 \mathbb{V}[R] + \alpha(1 - \alpha) 2 \text{cov}[\mu_{GP}, R] \end{aligned}$$

Optimal α :

$$\alpha^* = \frac{\mathbb{V}[R] - \text{cov}[\mu_{GP}, R]}{(\mu_{GP}[\mathcal{R}] - \mathcal{R}_\omega)^2 + \mathbb{V}[\mu_{GP}] + \mathbb{V}[R] - 2 \text{cov}[\mu_{GP}, R]}$$

Optimal estimator:

$$\begin{aligned} \tilde{R} &= \alpha^* \cdot \mu_{GP} + (1 - \alpha^*) \cdot R_\omega \\ \tilde{\epsilon} &= 2 \cdot \sqrt{\text{MSE}_{\alpha^*}(\tilde{R})} \end{aligned}$$

Approximations:

- $\mathbb{V}[R] = \left(\frac{\epsilon_R}{2}\right)^2$
- $\text{cov}[\mu_{GP}, R] = \left(\frac{\epsilon_R}{2}\right)^2 \sum_{i=1}^N \mathbf{k}_{\mathbf{x}^* \mathbf{x}_i} (\mathbf{k}_{\mathbf{x}_i \mathbf{x}_N} + \delta_{iN} \sigma_i^2)^{-1}$
- $\mathbb{V}[\mu_{GP}] = \mathbf{k}_{\mathbf{x}\mathbf{x}} [\mathbf{K}_{\mathbf{x}\mathbf{x}} + \sigma_n^2 \mathbf{I}]^{-1} \left(\frac{\epsilon_R}{2}\right)^2$
- $\mu_{GP}[\mathcal{R}] - \mathcal{R}_\omega = \mathbb{E}[\mu_{GP}] - \mathcal{R}_\omega \approx \mathbb{E}[\mu_{GP}[\hat{R}]] - \mu_{GP}[R]$

SNOWPAC ¹

Robust optimization problem statement

$$\mathcal{R}_\omega^* = \mathcal{R}^f(\mathbf{x}^*, \omega) = \min_{\mathcal{R}^c(\mathbf{x}, \omega) \leq 0} \mathcal{R}^f(\mathbf{x}, \omega)$$

Features of SNOWPAC:

- 0. Extension of NOWPAC:** Derivative-free nonlinear constraint optimization method using trust-regions (deterministic)
- 1. Estimate robustness measures:** Use sampling, e.g. $\mathcal{R}_\omega^f = \mathbb{E}[f_\omega(\mathbf{x})] \approx R^f = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}, \theta_i) + \varepsilon_N$
- 2. Implement new trust region management:** Account for noise ε_N in objective/constraint evaluations $\Rightarrow \Delta_{k+1} \geq \sqrt{\lambda_t \varepsilon_N}$
- 3. Introduce Gaussian process surrogates:** Mitigate effect of noise ε_N
- 4. Only feasible trial points, i.e. $\mathcal{R}_\omega^c(\mathbf{x}_{k+1}) \leq 0$, should be accepted**

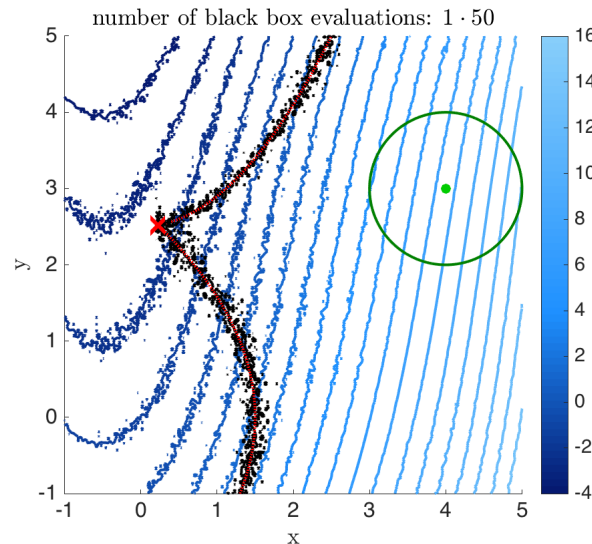
\Rightarrow Feasibility restoration mode:

$$\min_{\substack{m_k^c(\mathbf{x}) \leq \tau \\ \|\mathbf{x} - \mathbf{x}_k\| \leq \Delta_k}} \sum_{i \in \mathcal{I}} (m_k^{c_i}(\mathbf{x})^2 + \lambda_g m_k^{c_i}(\mathbf{x}))$$

¹F. Augustin, Y. Marzouk, A trust-region method for derivative-free nonlinear constrained stochastic optimization. 2017
Friedrich Menhorn (TUM), et al. | menhorn@in.tum.de | Derivative-free OUU using MF sampling

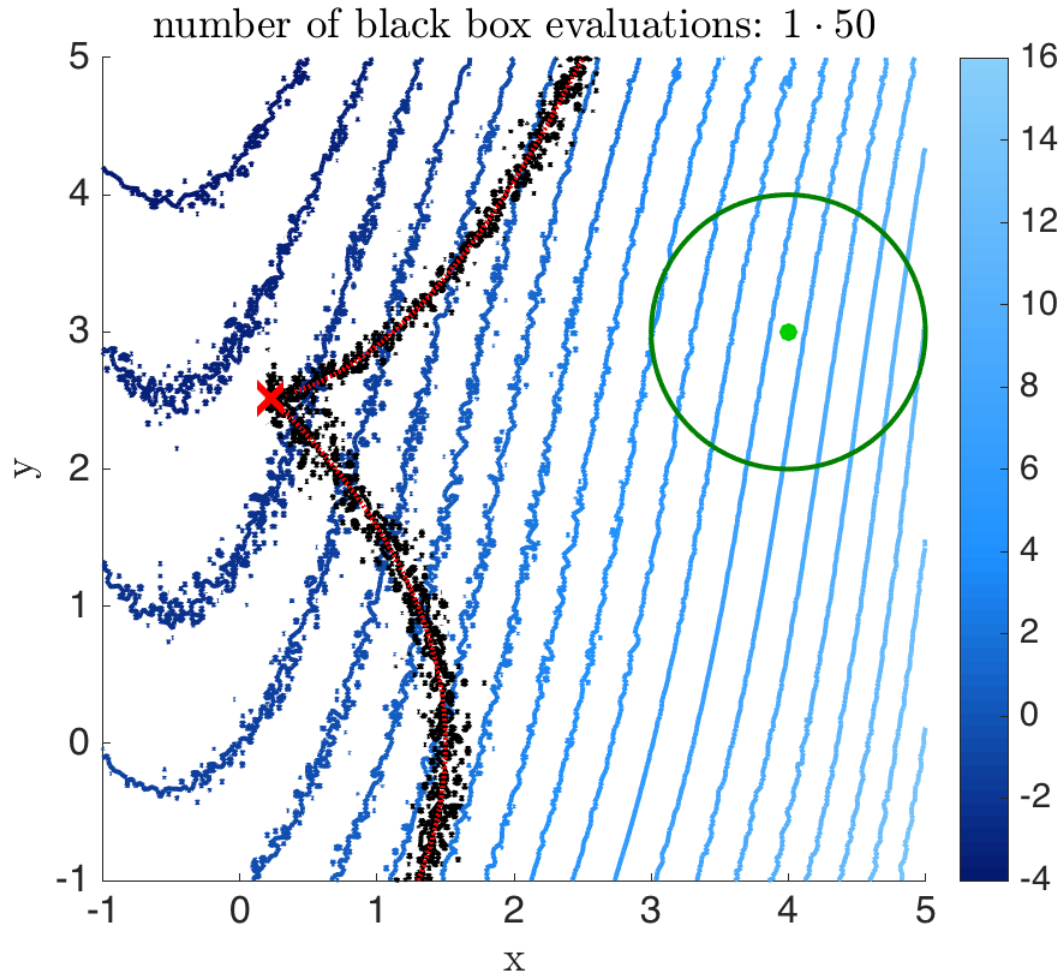
SNOWPAC – Example

$$\begin{aligned}
 & \min \mathbb{E}[\sin(x - 1 + \theta_1) + \sin(\frac{1}{2}y - 1 + \theta_1)^2] + \frac{1}{2}(x + \frac{1}{2})^2 - y \\
 & \text{s.t. } \mathbb{E}[-4x^2(1 + \theta_2) - 10\theta_3] \leq 25 - 10y, \theta_i \sim \mathcal{U}(\theta_i | -1, 1), i = 1, \dots, 4 \\
 & \mathbb{E}[-2y^2(1 + \theta_4) - 10(\theta_4 + \theta_2)] \leq 20x - 15, \mathbf{x}^{(0)} = (x^{(0)}, y^{(0)}) = (4, 3).
 \end{aligned} \tag{1}$$

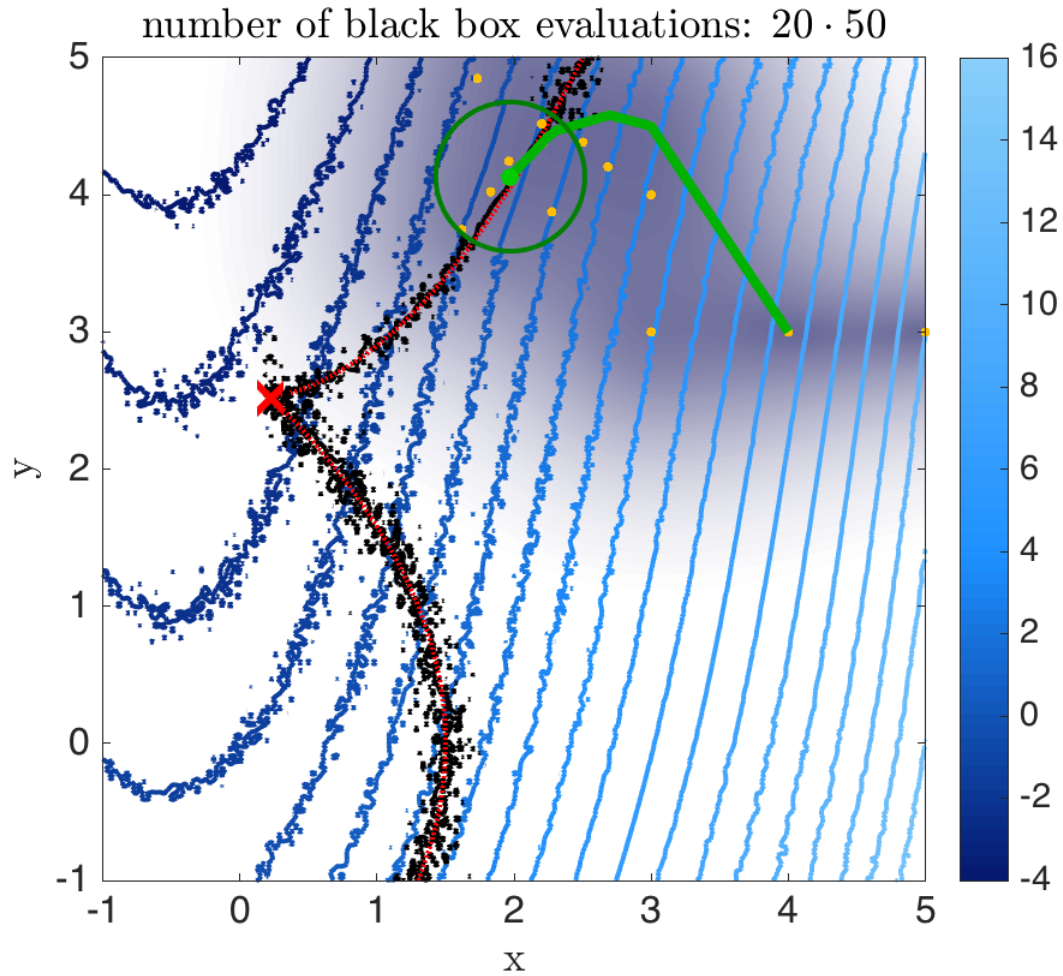


- Locally smoothed black box functions within the trust region
- Optimal design (red cross), exact constraints (red dotted lines)
- Objective (blue lines), constraints (black lines)
- Current design and trust region (green dot and circle)
- GP points (yellow dots), scaling factor γ (gray shade)

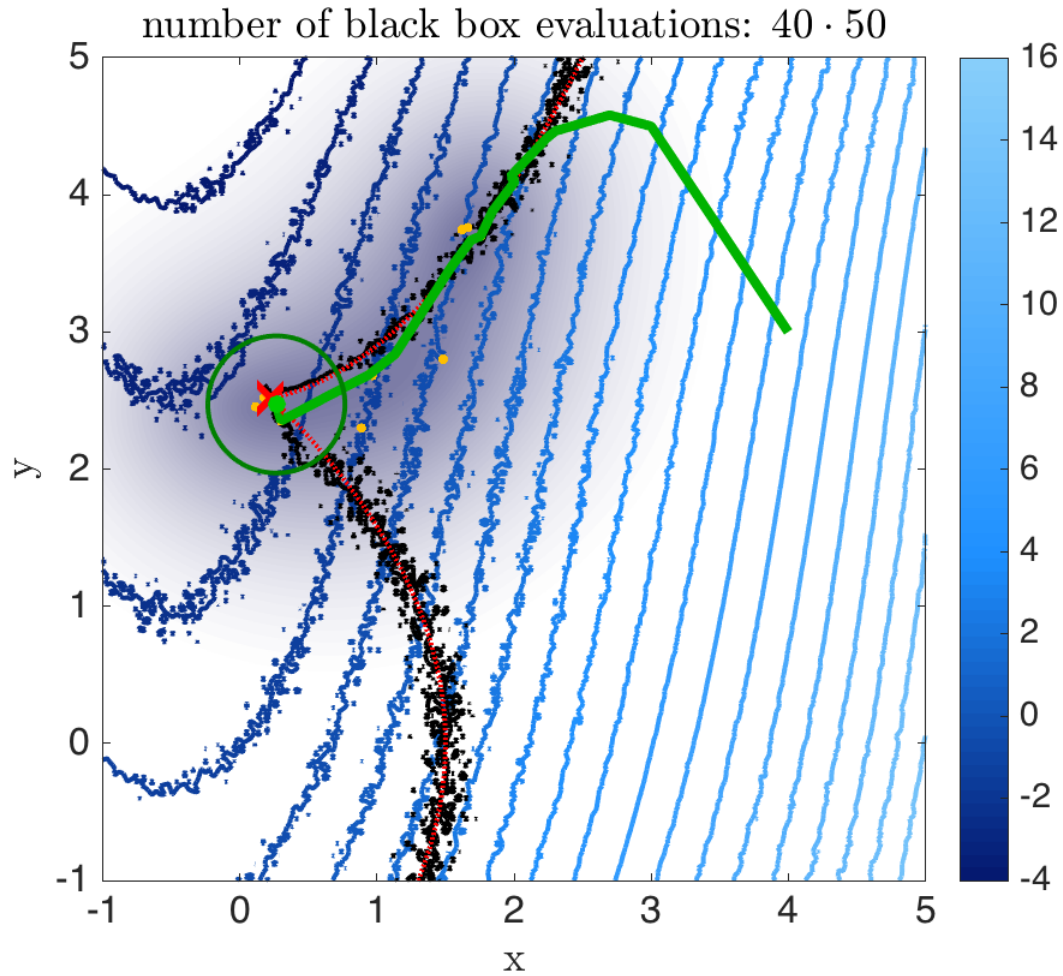
SNOWPAC – Example



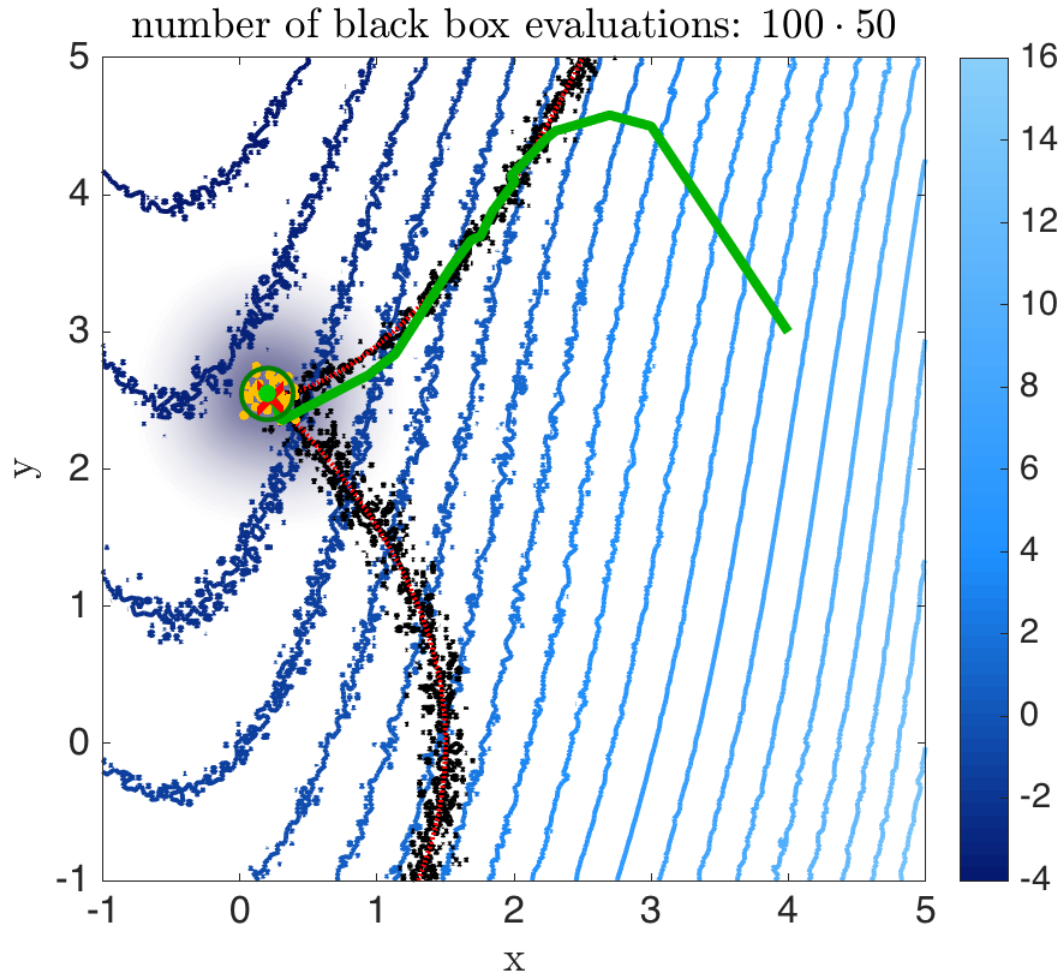
SNOWPAC – Example



SNOWPAC – Example



SNOWPAC – Example



Benchmark results



SNOWPAC – Benchmark setup

- Benchmark comparison of performance of SNOWPAC to COBYLA, NOMAD, SPSA and KWSA
- Use 8 CUTEst benchmark problems with added noise

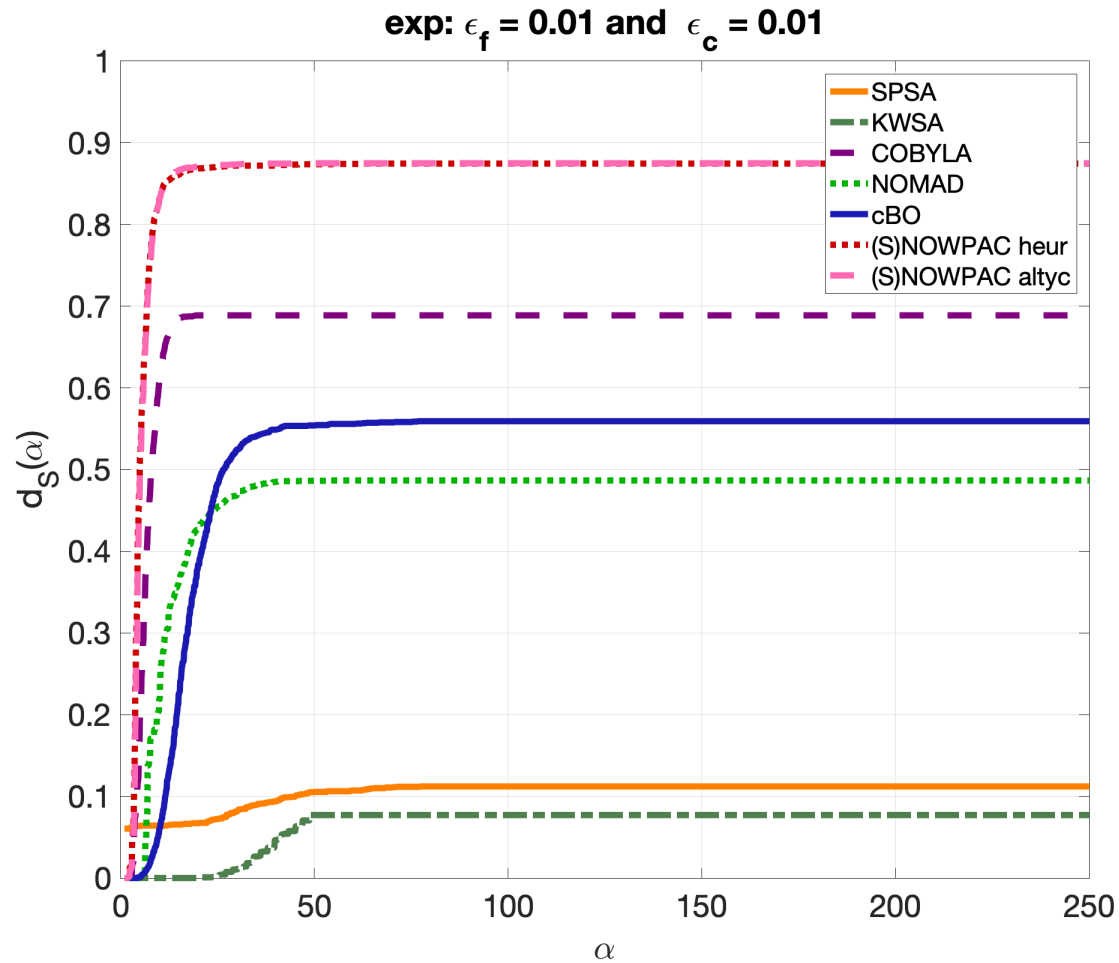
$$\begin{array}{l} \min R_N[f(\mathbf{x}) + \omega_1] \\ \text{s.t. } R_N[c_i(\mathbf{x}) + \omega_{2,i}] \leq 0, \end{array}$$

and approximate robustness measures with $N \in \{200, 1000, 2000\}$ samples of

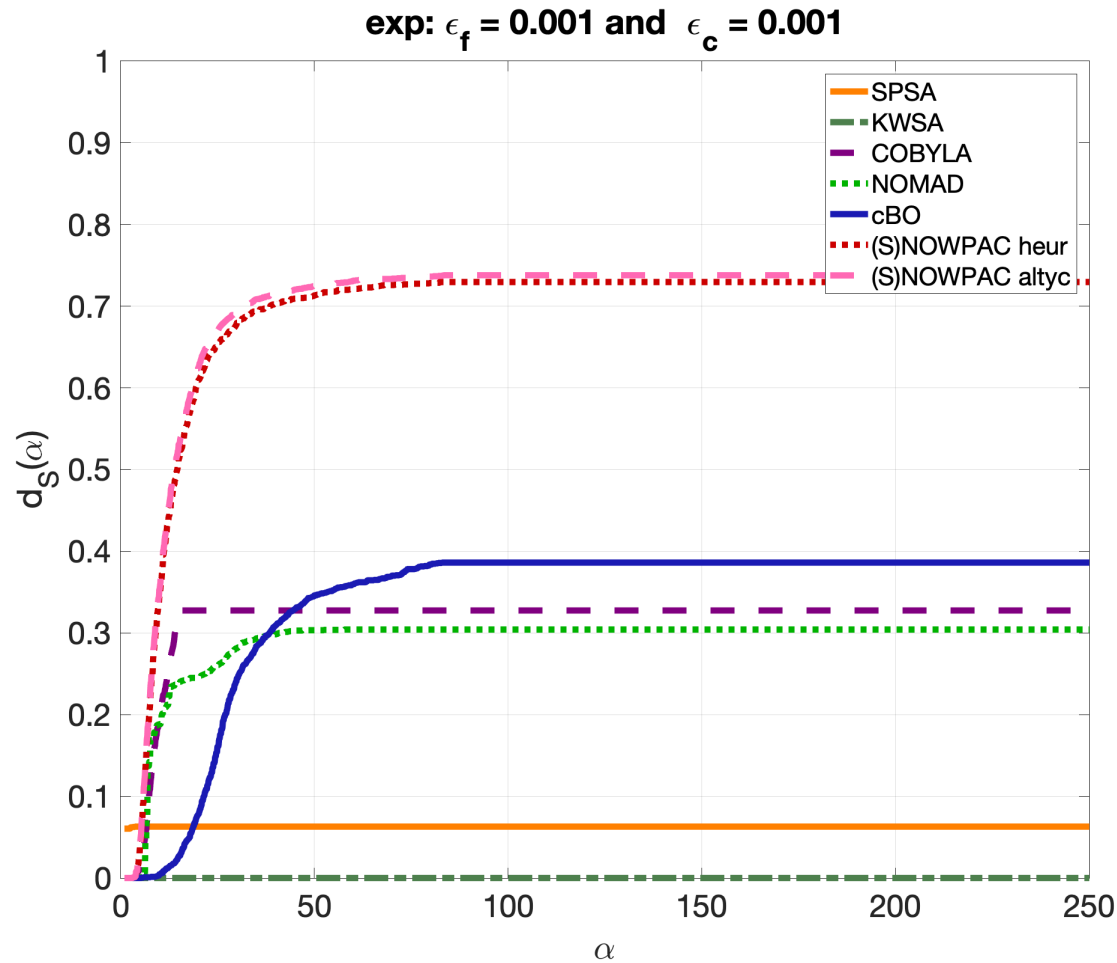
$$\omega_1, \omega_{2,i} \sim \mathcal{U}[-1, 1]$$

- Limit max number of black box evaluations to 250
- Comparison of results from 100 repeated optimization runs
- Use data profile [Moré/Wild2009] to compare performance $d_S(\alpha) = \frac{1}{2400} \left| \left\{ p \in \mathcal{P} : \frac{t_{p,S}}{n_p+1} \leq \alpha \right\} \right|$
 - Based on $|\mathcal{P}| = 8 \cdot 100 \cdot 3 = 2400$ optimization runs

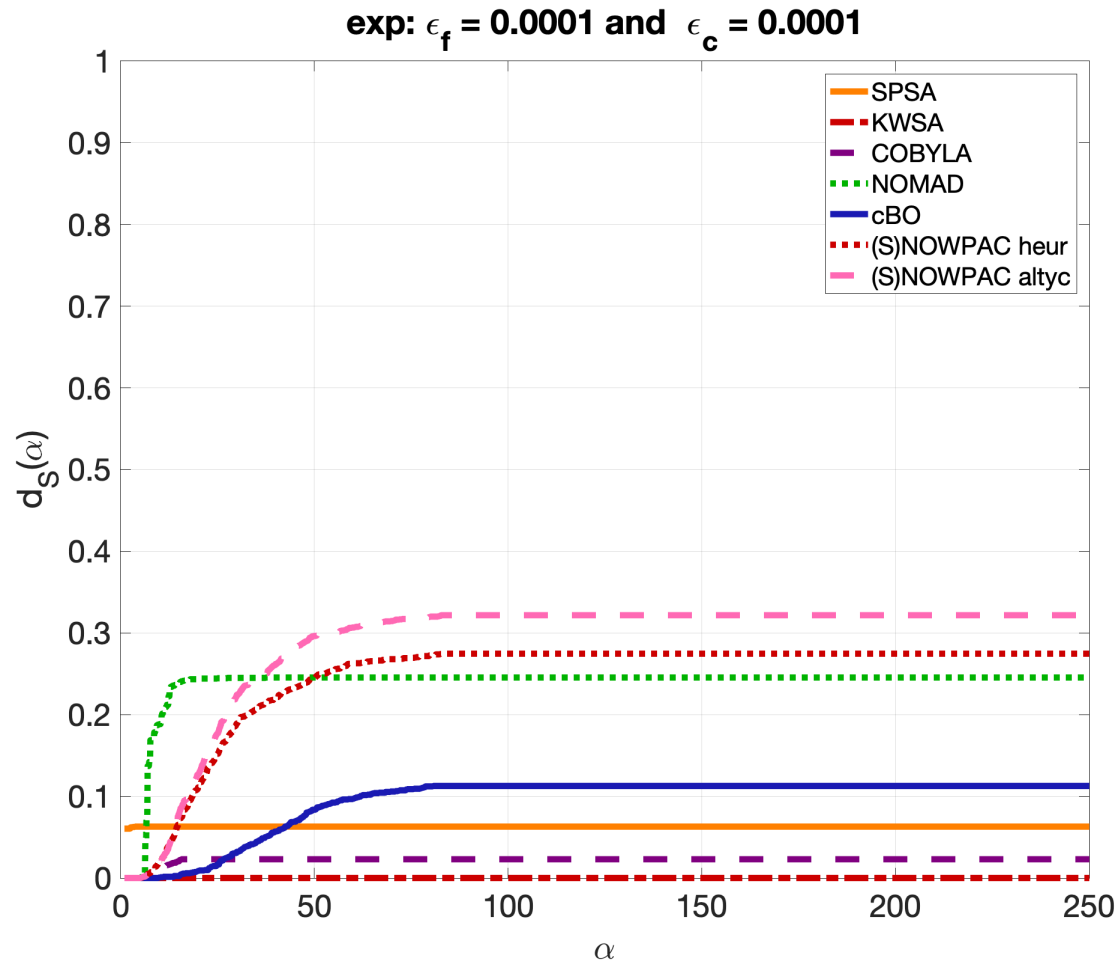
SNOWPAC – Benchmark results



SNOWPAC – Benchmark results



SNOWPAC – Benchmark results



MLMC Results



Problem statement

Objective:

$$f(x) = \begin{cases} (x-2)^2 & \text{if } x \leq 3 \\ 2\log(x-2) + 1 & \text{if } x > 3 \\ x \in [0, 6] \end{cases}$$

Constraint:

$$g_{det}(x) = \frac{2 \cdot \log(1.5)}{2.5} x + 1 - \frac{2 \cdot \log(1.5)}{2.5}$$
$$g_H(x, \xi) = g_{det}(x) + \xi^3$$
$$g_L(x, \xi) = g_{det}(x) + A\xi^3, \xi \sim \mathcal{U}(-0.5, 0.5)$$

OUU:

$$\min_x f(x)$$

Mean:

$$\text{s.t. } f(x) \geq \mathbb{E}[g_H(x, \xi)]$$

Push back:

$$\text{s.t. } f(x) \geq \mathbb{E}[g_H(x, \xi)] + 3\sigma[g_H(x, \xi)]$$

Problem statement

Objective:

$$f(x) = \begin{cases} (x-2)^2 & \text{if } x \leq 3 \\ 2\log(x-2) + 1 & \text{if } x > 3 \\ x \in [0, 6] \end{cases}$$

Constraint:

$$g_{det}(x) = \frac{2 \cdot \log(1.5)}{2.5}x + 1. - \frac{2 \cdot \log(1.5)}{2.5}$$

$$g_H(x, \xi) = g_{det}(x) + \xi^3$$

$$g_L(x, \xi) = g_{det}(x) + A\xi^3, \xi \sim \mathcal{U}(-0.5, 0.5)$$

OUU:

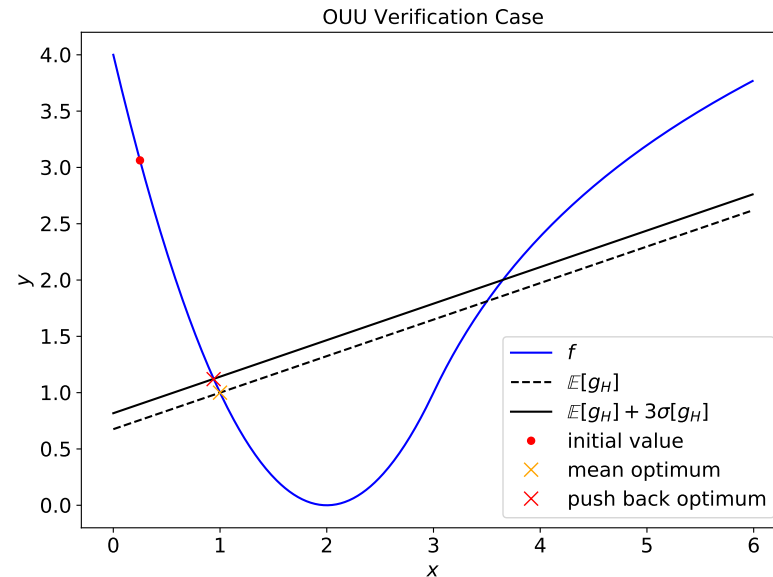
$$\min_x f(x)$$

Mean:

$$\text{s.t. } f(x) \geq \mathbb{E}[g_H(x, \xi)]$$

Push back:

$$\text{s.t. } f(x) \geq \mathbb{E}[g_H(x, \xi)] + 3\sigma[g_H(x, \xi)]$$



Problem statement

Objective:

$$f(x) = \begin{cases} (x-2)^2 & \text{if } x \leq 3 \\ 2\log(x-2) + 1 & \text{if } x > 3 \\ x \in [0, 6] \end{cases}$$

Constraint:

$$g_{det}(x) = \frac{2 \cdot \log(1.5)}{2.5}x + 1 - \frac{2 \cdot \log(1.5)}{2.5}$$

$$g_H(x, \xi) = g_{det}(x) + \xi^3$$

$$g_L(x, \xi) = g_{det}(x) + A\xi^3, \xi \sim \mathcal{U}(-0.5, 0.5)$$

OUU:

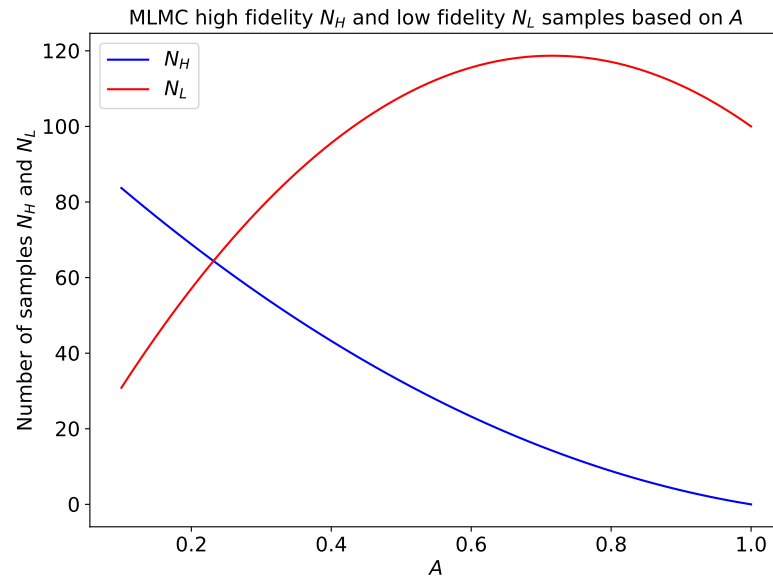
$$\min_x f(x)$$

Mean:

$$\text{s.t. } f(x) \geq \mathbb{E}[g_H(x, \xi)]$$

Push back:

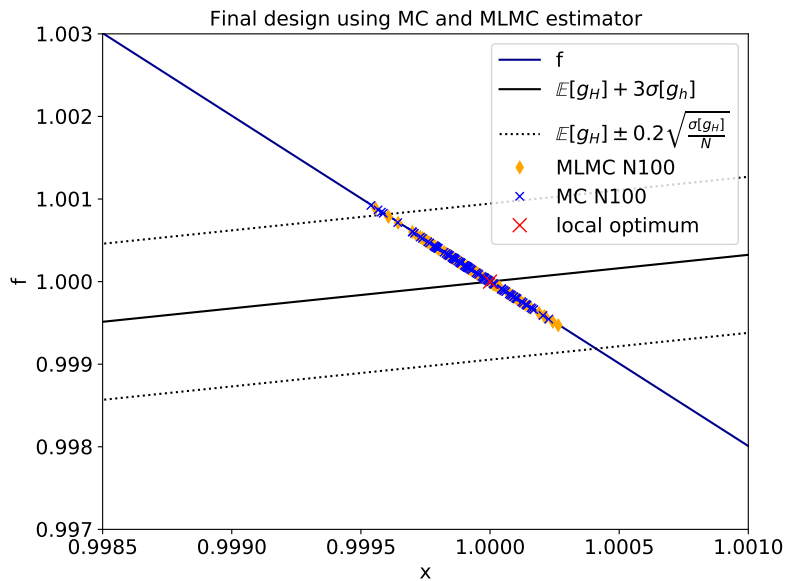
$$\text{s.t. } f(x) \geq \mathbb{E}[g_H(x, \xi)] + 3\sigma[g_H(x, \xi)]$$



Results

Mean:

$$\text{s.t. } f(x) \geq \mathbb{E}[g_H(x, \xi)]$$



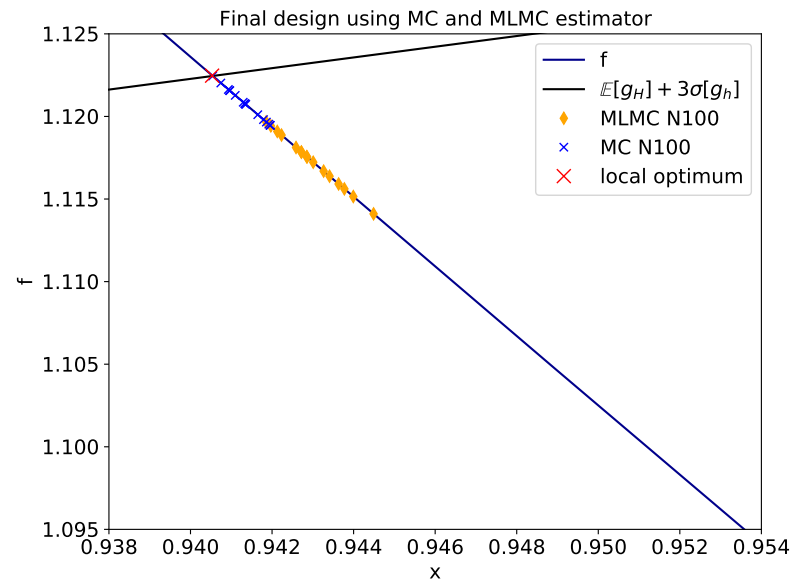
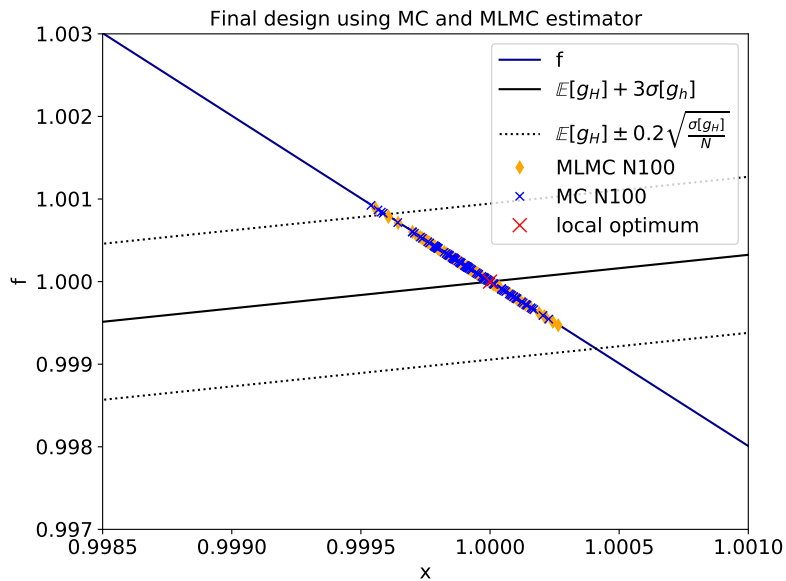
Results

Mean:

$$\text{s.t. } f(x) \geq \mathbb{E}[g_H(x, \xi)]$$

Push back:

$$\text{s.t. } f(x) \geq \mathbb{E}[g_H(x, \xi)] + 3\sigma[g_H(x, \xi)]$$



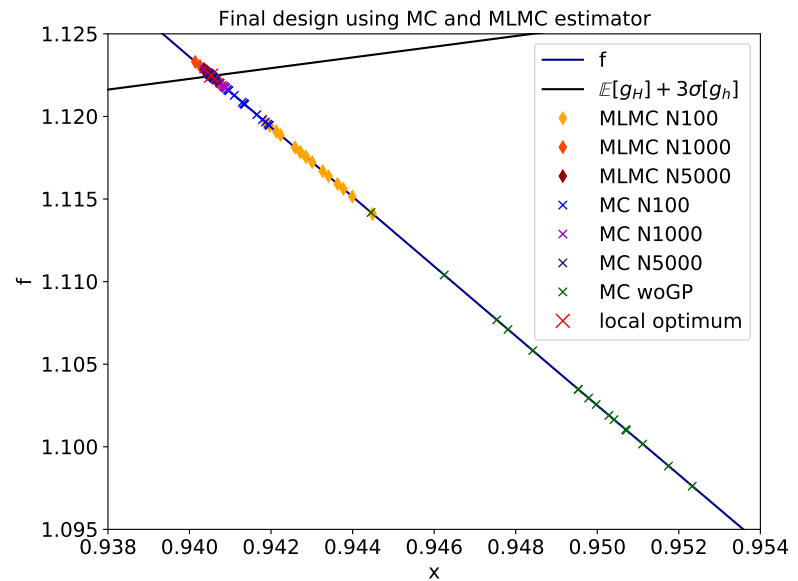
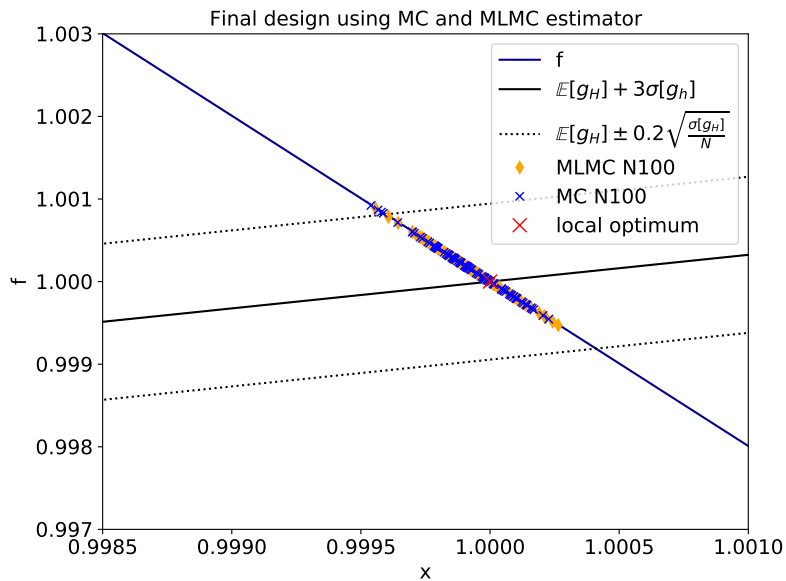
Results

Mean:

$$\text{s.t. } f(x) \geq \mathbb{E}[g_H(x, \xi)]$$

Push back:

$$\text{s.t. } f(x) \geq \mathbb{E}[g_H(x, \xi)] + 3\sigma[g_H(x, \xi)]$$



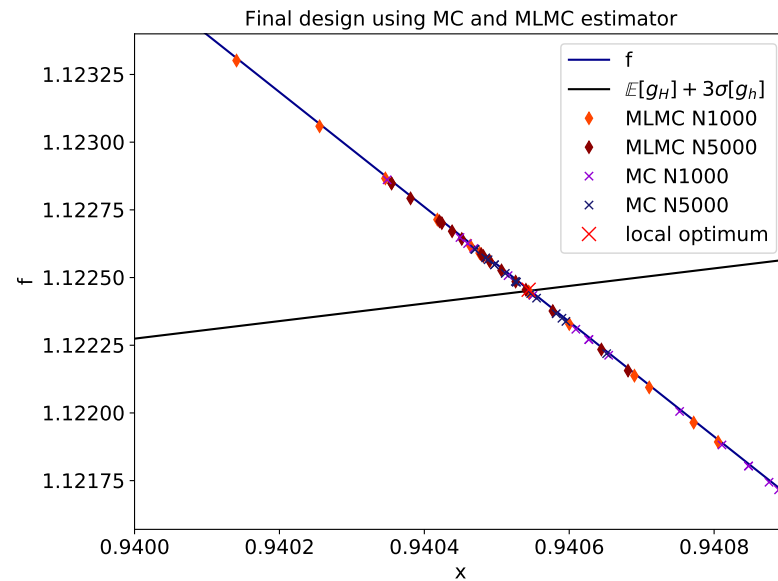
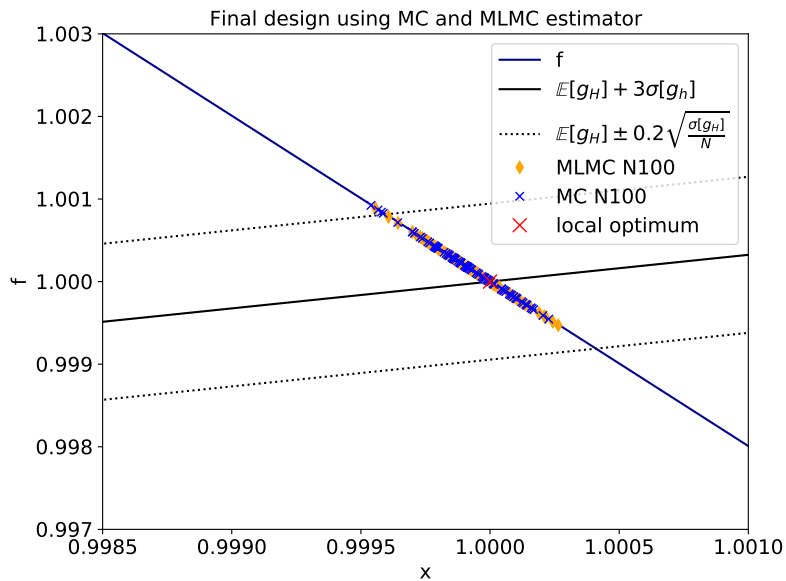
Results

Mean:

$$\text{s.t. } f(x) \geq \mathbb{E}[g_H(x, \xi)]$$

Push back:

$$\text{s.t. } f(x) \geq \mathbb{E}[g_H(x, \xi)] + 3\sigma[g_H(x, \xi)]$$



Summary:

- **NOWPAC** – Derivative-free trust region methods for constrained nonlinear optimization
- **SNOWPAC** – Stochastic derivative-free optimization using Gaussian process surrogates
⇒ **New** analytic approach for noise reduction
- **DAKOTA** – Design Analysis Kit for Optimization and Terascale Applications
⇒ **New** standard error estimates for MLMC used in **SNOWPAC**.

Future work and open questions:

- Alternatives for surrogate model (e.g. RBF surrogates)
- Integrate new developments for Gaussian process surrogates (e.g. non-stationary kernels)
- Investigate MLMC and MC behavior for benchmark problem

Links:

- SNOWPAC: bitbucket.org/fmaugust/nowpac
- Dakota: dakota.sandia.gov

References:

- F. Augustin, Y. Marzouk, A trust-region method for derivative-free nonlinear constrained stochastic optimization. 2017
- GG, FM, X. Huan, C. Safta, YM, H. Najm, ME, Progress in scramjet design optimization under uncertainty using simulations of the HIFiRE configuration. AIAA 2019