Scientific Computing
Department of Informatics
Technical University of Munich

Derivative-Free Multifidelity Design Optimization under Uncertainty of a Scramjet(-inspired Problem)

Friedrich Menhorn[1], Florian Augustin[2,3], Youssef M. Marzouk[2], Gianluca Geraci[4], Michael S. Eldred[4], Hans-Joachim Bungartz[1]

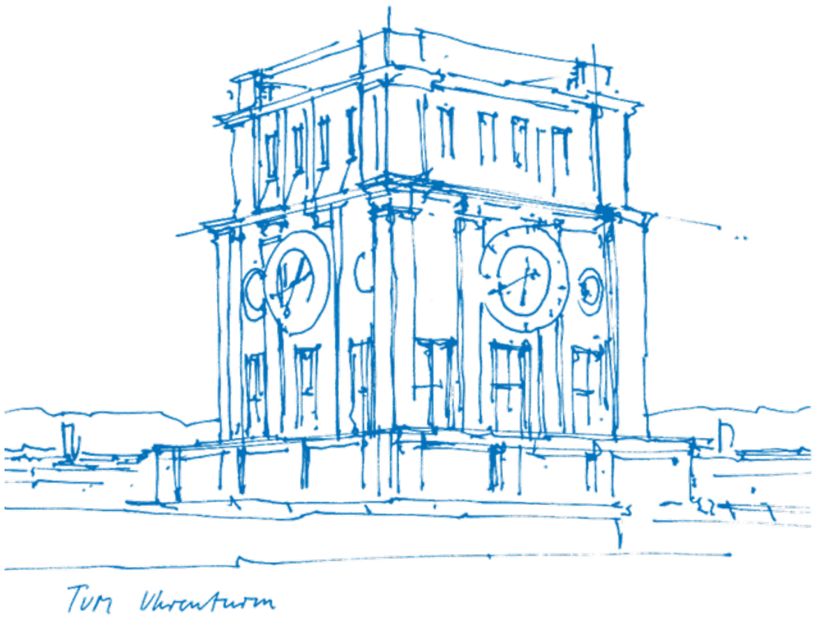
menhorn@in.tum.de

[1]Technical University of Munich, GER

[2]Massachusetts Institute of Technology, USA

[3]Mathworks, USA

[4]Sandia National Laboratories, USA



SIAM CSE

MS312: OUU Using MF and DF Approaches

Spokane, February 28th, 2019

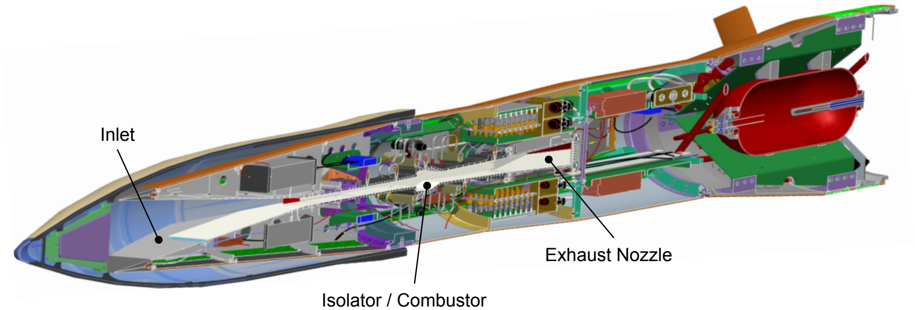
Scientific Computing
Department of Informatics
Technical University of Munich

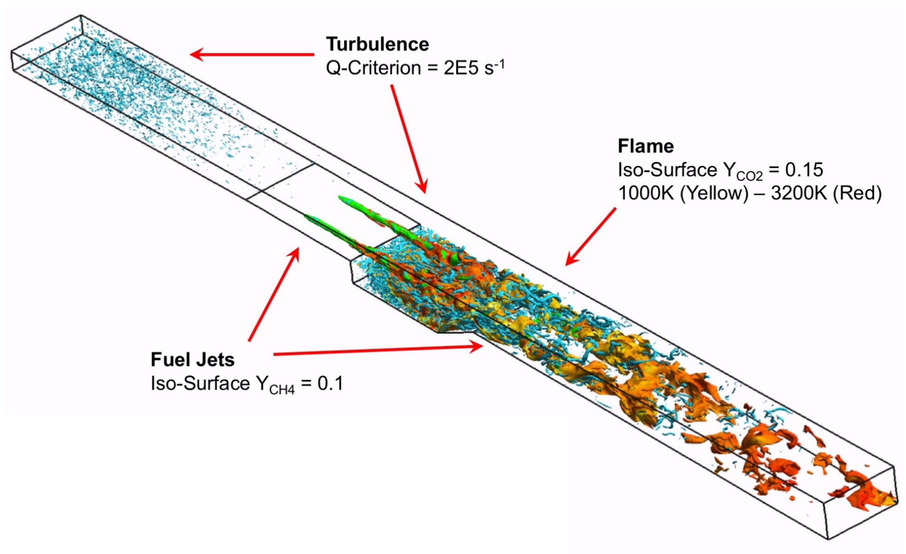
Motivation: Design optimization of a SCRAMJET

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 $\mathscr{R}$, e.g. $\mathbb{E}, \mathbb{V}$, CVaR.
- E.g., weigh expected gain vs. confidence: $\max \mathbb{E} - \lambda \mathbb{V}^{\frac{1}{2}}$

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

[1] F. Augustin, Y. Marzouk, A trust-region method for derivative-free nonlinear constrained stochastic optimization. 2017

SNOWPAC [1]

Robust optimization problem statement

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

[1] F. Augustin, Y. Marzouk, A trust-region method for derivative-free nonlinear constrained stochastic optimization. 2017

SNOWPAC [1]

Robust optimization problem statement

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

Features of SNOWPAC:

[1] F. Augustin, Y. Marzouk, A trust-region method for derivative-free nonlinear constrained stochastic optimization. 2017

SNOWPAC [1]

Robust optimization problem statement

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

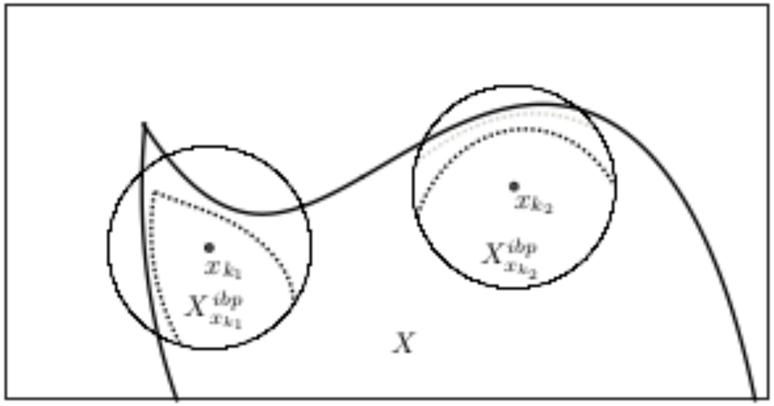
Features of SNOWPAC:

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

[1] F. Augustin, Y. Marzouk, A trust-region method for derivative-free nonlinear constrained stochastic optimization. 2017

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**

[2] F. Augustin, Y. Marzouk, NOWPAC: A path-augmented constraint handling approach for nonlinear derivative-free optimization. 2014

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. $\mathcal{R}^f_\omega = \mathbb{E}[f_\omega(\mathbf{x})] \approx R^f = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}, \theta_i) + \varepsilon_N$

[1] F. Augustin, Y. Marzouk, A trust-region method for derivative-free nonlinear constrained stochastic optimization. 2017

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. $\mathcal{R}^f_\omega = \mathbb{E}[f_\omega(\mathbf{x})] \approx R^f = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}, \theta_i) + \varepsilon_N$

   NEW: Leverage multilevel estimators.

[1] F. Augustin, Y. Marzouk, A trust-region method for derivative-free nonlinear constrained stochastic optimization. 2017

Multilevel error estimator for SNOWPAC

Generic MLMC estimators:

- Mean:

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

- Variance:

$$\mathbb{V}[Q] = \sum_{\ell=0}^{L} \mathbb{E}\left[(Q_\ell - \mathbb{E}[Q_\ell])^2 - (Q_{\ell-1} - \mathbb{E}[Q_{\ell-1}])^2 \right] = \sum_{\ell=0}^{L} \mathbb{E}[P_\ell^2] - \mathbb{E}[P_{\ell-1}^2]$$

$$\approx \quad s_{ML}^2 \quad = \sum_{\ell=0}^{L} (\widehat{P_\ell^2} - \widehat{P_{\ell-1}^2}), \quad \text{where} \quad \hat{P}_\ell^2 = \frac{1}{N_\ell - 1} \sum_{i=1}^{N_\ell} \left(Q_\ell^{(i)} - \hat{Q}_\ell \right)^2.$$

- Standard deviation:

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

Multilevel error estimator for SNOWPAC

Generic MLMC estimators:

- Mean:

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

- Variance:

$$\mathbb{V}[Q] = \sum_{\ell=0}^{L} \mathbb{E}\left[(Q_\ell - \mathbb{E}[Q_\ell])^2 - (Q_{\ell-1} - \mathbb{E}[Q_{\ell-1}])^2 \right] = \sum_{\ell=0}^{L} \mathbb{E}[P_\ell^2] - \mathbb{E}[P_{\ell-1}^2]$$

$$\approx \quad s_{ML}^2 \quad = \sum_{\ell=0}^{L} (\widehat{P_\ell^2} - \widehat{P_{\ell-1}^2}), \quad \text{where} \quad \hat{P}_\ell^2 = \frac{1}{N_\ell - 1} \sum_{i=1}^{N_\ell} \left(Q_\ell^{(i)} - \hat{Q}_\ell \right)^2.$$

- Standard deviation:

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

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

Scientific Computing
Department of Informatics
Technical University of Munich

Multilevel error estimator for SNOWPAC

Generic MLMC error estimators:

- Mean $\hat{Q}_L$:

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

- Variance s^2_{ML}:

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

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

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

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

Multilevel error estimator for SNOWPAC

Generic MLMC error estimators:

- Mean $\hat{Q}_L$:

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

- Variance s^2_{ML}:

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

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

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

$$SE(s^2_{ML}) \approx \frac{1}{2\sqrt{s^2_{ML}}} \sqrt{\mathbb{V}[s^2_{ML}]} = \frac{\varepsilon_N}{2}$$

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

Scientific Computing
Department of Informatics
Technical University of Munich

SNOWPAC [1]

Robust optimization problem statement

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

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. $\mathscr{R}^f_\omega = \mathbb{E}[f_\omega(\mathbf{x})] \approx R^f = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}, \theta_i) + \varepsilon_N$
2. **Implement new trust region management**: Account for noise ε_N in objective/constraint evaluations $\Rightarrow \Delta_{k+1} \geq \sqrt{\lambda_t \varepsilon_N}$

[1] F. Augustin, Y. Marzouk, A trust-region method for derivative-free nonlinear constrained stochastic optimization. 2017

SNOWPAC [1]

Robust optimization problem statement

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

Features of SNOWPAC:

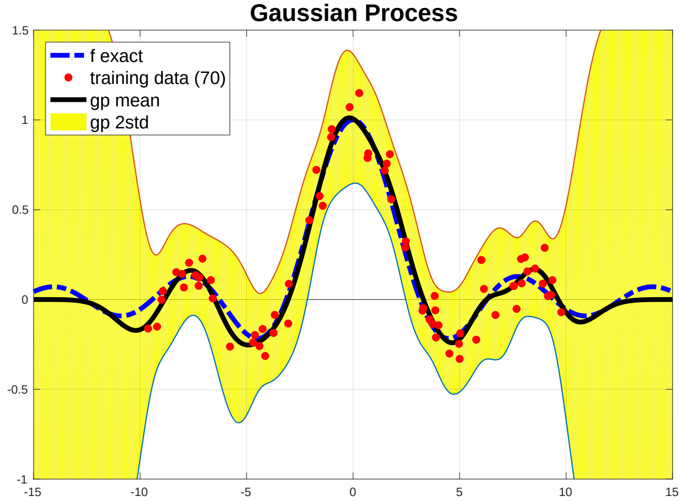
0. **Extension of NOWPAC**: Derivative-free nonlinear constraint optimization method using trust-regions (deterministic)
1. **Estimate robustness measures**: Use sampling, e.g. $\mathscr{R}^f_\omega = \mathbb{E}[f_\omega(\mathbf{x})] \approx R^f = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}, \theta_i) + \varepsilon_N$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

[1] F. Augustin, Y. Marzouk, A trust-region method for derivative-free nonlinear constrained stochastic optimization. 2017

SNOWPAC – Gaussian process surrogate

Build Gaussian process surrogate

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



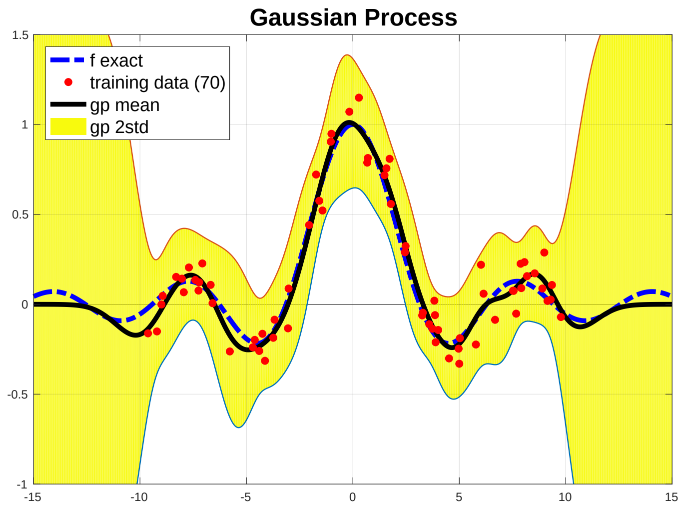
Scientific Computing
Department of Informatics
Technical University of Munich

SNOWPAC – Gaussian process surrogate

Build Gaussian process surrogate

- **Use black box evaluations** to build global Gaussian process surrogates
- **Replace noisy black box evaluations by GP mean**:

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



Scientific Computing
Department of Informatics
Technical University of Munich

SNOWPAC – Gaussian process surrogate

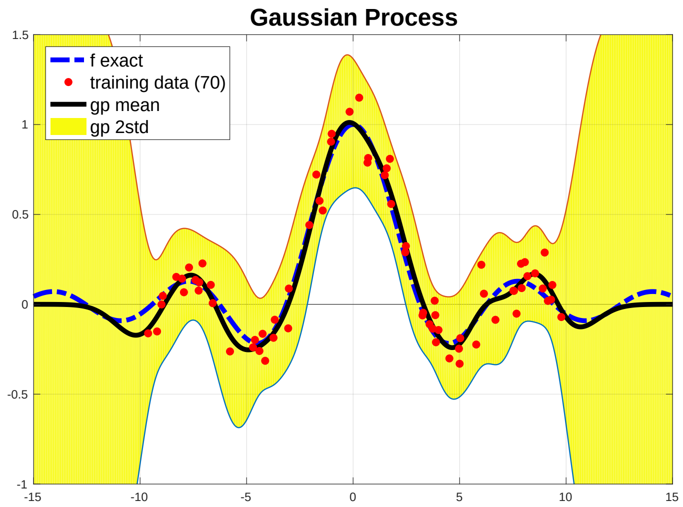
Build Gaussian process surrogate

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

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

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

- **Replace noise estimate by**:



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

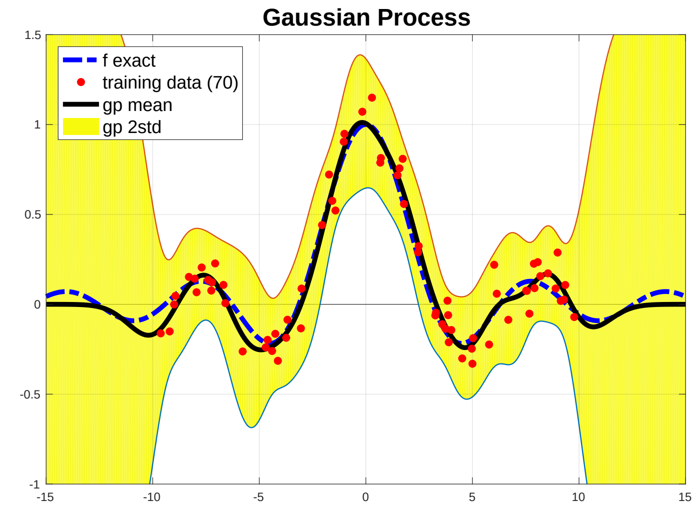
with

– GP mean: $\mu_{GP}(\mathbf{x}) = \mathbf{k_{xX}}[\mathbf{K_{XX}} + \sigma^2_\mathbf{n}\mathbf{I}]^{-1}\mathbf{R}$

– GP variance: $\sigma^2_{GP}(\mathbf{x}) = k_{\mathbf{xx}} - \mathbf{k_{xX}}[\mathbf{K_{XX}} + \sigma^2_\mathbf{n}\mathbf{I}]^{-1}\mathbf{k_{Xx}}$

SNOWPAC – Gaussian process surrogate

Build Gaussian process surrogate



- **Use black box evaluations** to build global Gaussian process surrogates
- **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 $\alpha = e^{-\sqrt{\sigma^2_{\text{GP}}(\mathbf{x})}}$

with

– GP mean: $\mu_{\text{GP}}(\mathbf{x}) = \mathbf{k_{xX}}[\mathbf{K_{XX}} + \sigma_\mathbf{n}^\mathbf{2}\mathbf{I}]^{-1}\mathbf{R}$

– GP variance: $\sigma^2_{\text{GP}}(\mathbf{x}) = k_{\mathbf{xx}} - \mathbf{k_{xX}}[\mathbf{K_{XX}} + \sigma_\mathbf{n}^\mathbf{2}\mathbf{I}]^{-1}\mathbf{k_{Xx}}$

SNOWPAC – Gaussian Process Noise Correction

MSE:

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

SNOWPAC – Gaussian Process Noise Correction

MSE:

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

Optimal α:

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

SNOWPAC – Gaussian Process Noise Correction

MSE:

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

Optimal α:

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

Optimal estimator:

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

SNOWPAC – Gaussian Process Noise Correction

MSE:

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

Optimal α:

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

Optimal estimator:

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

Approximations:

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

SNOWPAC – Gaussian Process Noise Correction

MSE:

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

Optimal α:

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

Optimal estimator:

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

Approximations:

- $\mathbb{V}[R] = (\frac{\varepsilon_R}{2})^2$
- $\mathbb{V}[\mu_{GP}] = \mathbf{k_{xX}}[\mathbf{K_{XX}} + \sigma_\mathbf{n}^\mathbf{2}\mathbf{I}]^{-1}(\frac{\varepsilon_\mathbf{R}}{2})^2$

SNOWPAC – Gaussian Process Noise Correction

MSE:

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

Optimal α:

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

Optimal estimator:

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

Approximations:

- $\mathbb{V}[R] = (\frac{\varepsilon_R}{2})^2$
- $\mathbb{V}[\mu_{GP}] = \mathbf{k_{xX}}[\mathbf{K_{XX}} + \sigma_\mathbf{n}^2\mathbf{I}]^{-1}(\frac{\varepsilon_\mathbf{R}}{2})^2$
- $\text{cov}[\mu_{GP}, R] = (\frac{\varepsilon_R}{2})^2 \sum_{i=1}^N \mathbf{k}_{\mathbf{x}^*\mathbf{x}_i}(\mathbf{k}_{\mathbf{x}_i\mathbf{x}_N} + \delta_{iN}\sigma_i^2)^{-\tilde{1}}$

SNOWPAC – Gaussian Process Noise Correction

MSE:

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

Optimal α:

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

Optimal estimator:

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

Approximations:

- $\mathbb{V}[R] = (\frac{\varepsilon_R}{2})^2$
- $\mathbb{V}[\mu_{GP}] = \mathbf{k_{xX}}[\mathbf{K_{XX}} + \sigma_\mathbf{n}^\mathbf{2}\mathbf{I}]^{-1}(\frac{\varepsilon_\mathbf{R}}{2})^2$
- $\mathrm{cov}[\mu_{GP}, R] = (\frac{\varepsilon_R}{2})^2 \sum_{i=1}^N \mathbf{k}_{\mathbf{x}^*\mathbf{x}_i}(\mathbf{k}_{\mathbf{x}_i\mathbf{x}_N} + \delta_{iN}\sigma_i^2)^{-\tilde{1}}$
- $\mu_{GP}[\mathscr{R}] - \mathscr{R}_\omega = \mathbb{E}[\mu_{GP}] - \mathscr{R}_\omega \approx \mathbb{E}[\mu_{GP}[\hat{R}]] - \mu_{GP}[R]$

SNOWPAC [1]

Robust optimization problem statement

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

Features of SNOWPAC:

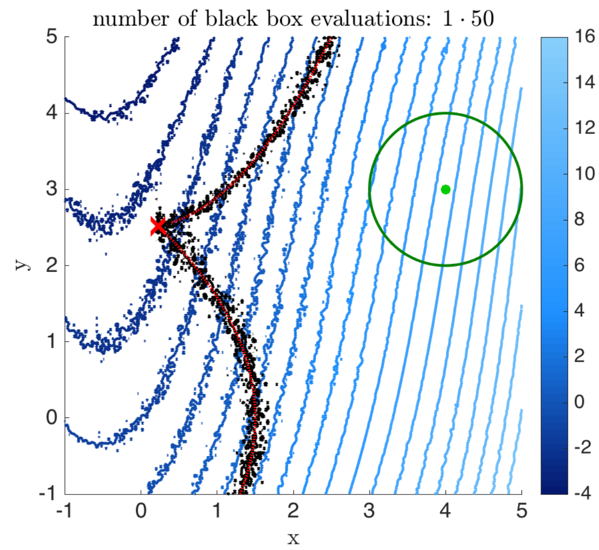
0. **Extension of NOWPAC**: Derivative-free nonlinear constraint optimization method using trust-regions (deterministic)
1. **Estimate robustness measures**: Use sampling, e.g. $\mathscr{R}^f_\omega = \mathbb{E}[f_\omega(\mathbf{x})] \approx R^f = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}, \theta_i) + \varepsilon_N$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

$\Rightarrow$ Feasibility restoration mode: $\min_{\substack{m_k^c(\mathbf{x})\leq\tau \\ \|\mathbf{x}-\mathbf{x}_k\|\leq\Delta_k}} \sum_{i\in\mathscr{I}} \left(m_k^{c_i}(\mathbf{x})^2 + \lambda_g m_k^{c_i}(\mathbf{x})\right)$

[1] F. Augustin, Y. Marzouk, A trust-region method for derivative-free nonlinear constrained stochastic optimization. 2017

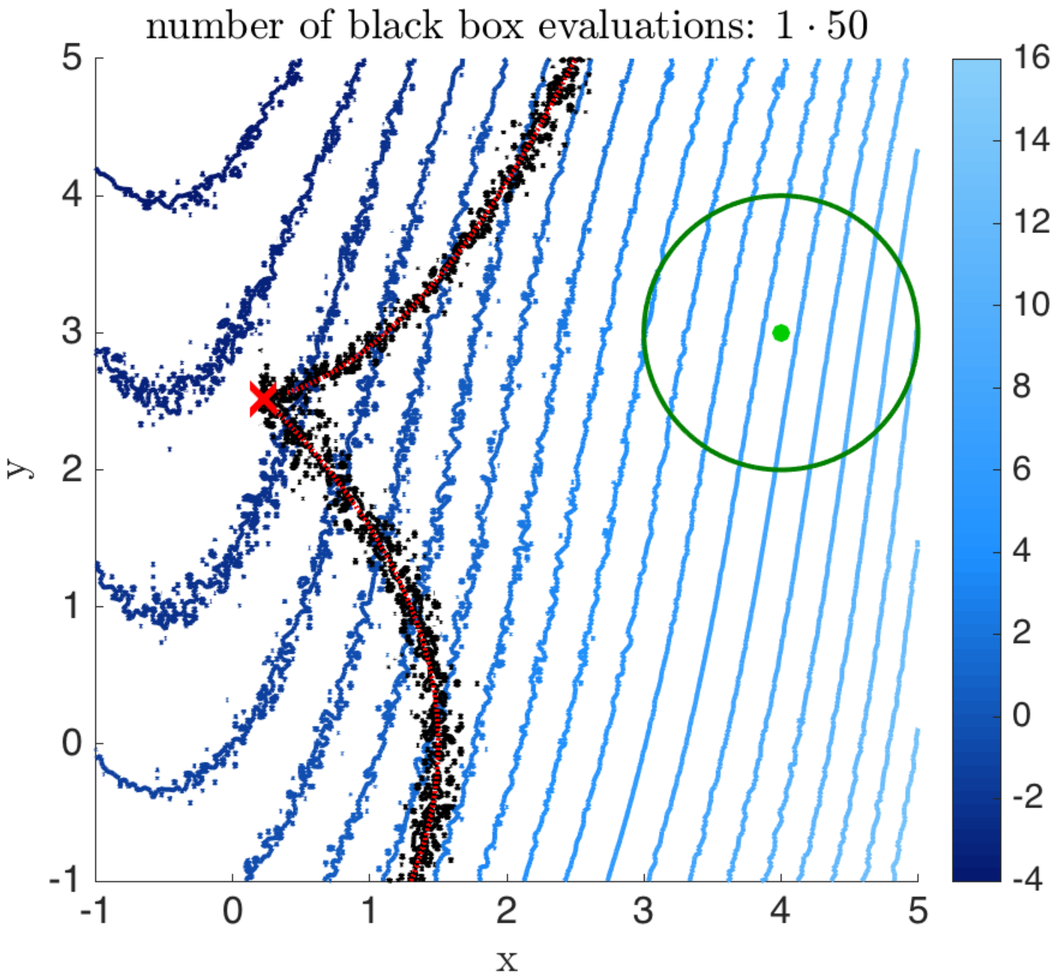
SNOWPAC – Example

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

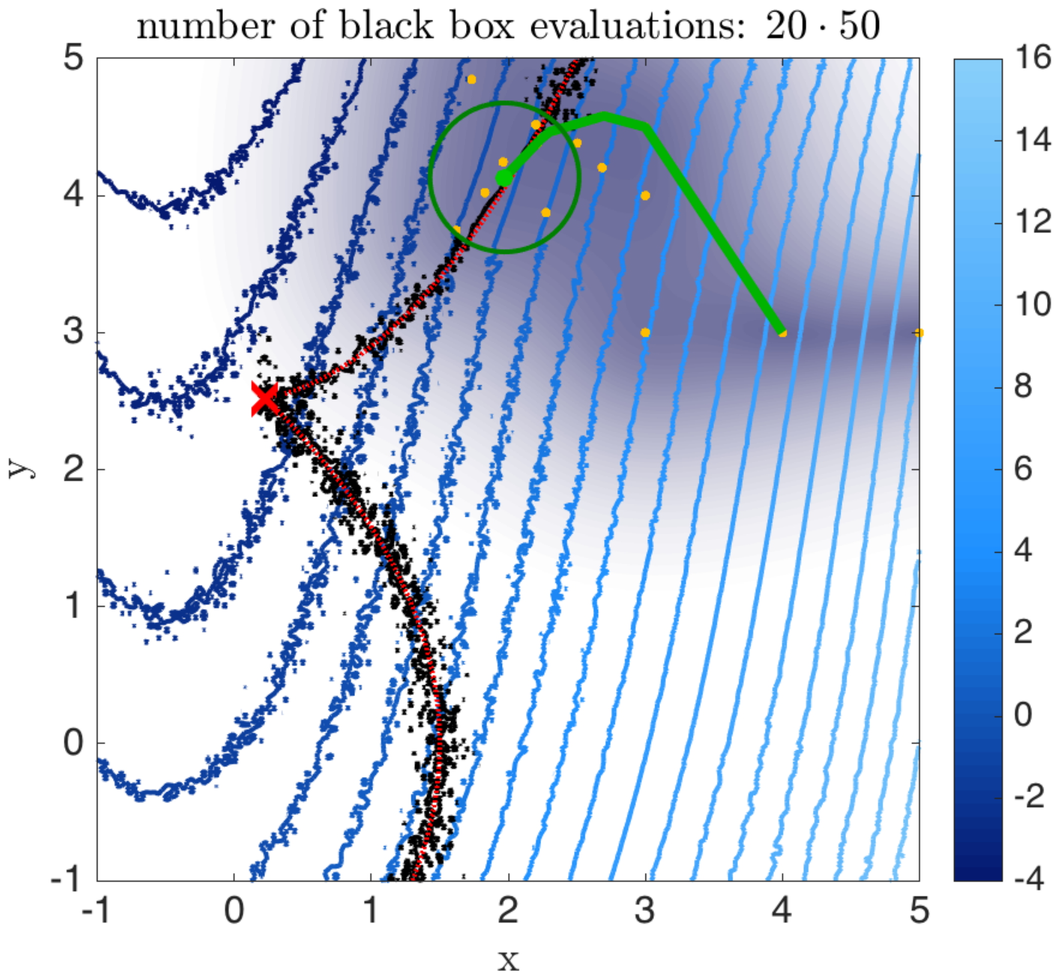


- Locally smoothed black box functions within the trust region
- Optimal design (red cross), exact constraints (red dotted lines)
- Objective (blue lines), constraints (black lines)
- Current design and trust region (green dot and circle)
- GP points (yellow dots), scaling factor γ (gray shade)

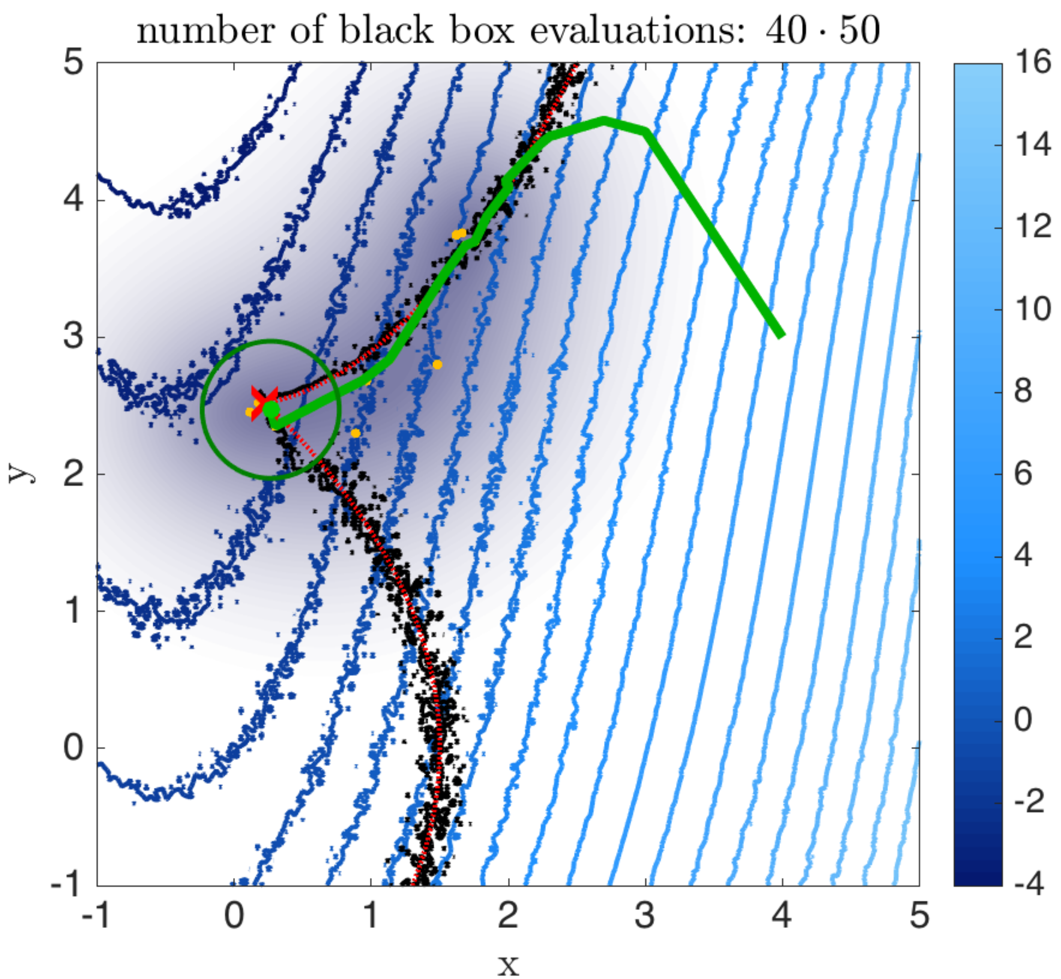
SNOWPAC – Example



SNOWPAC – Example

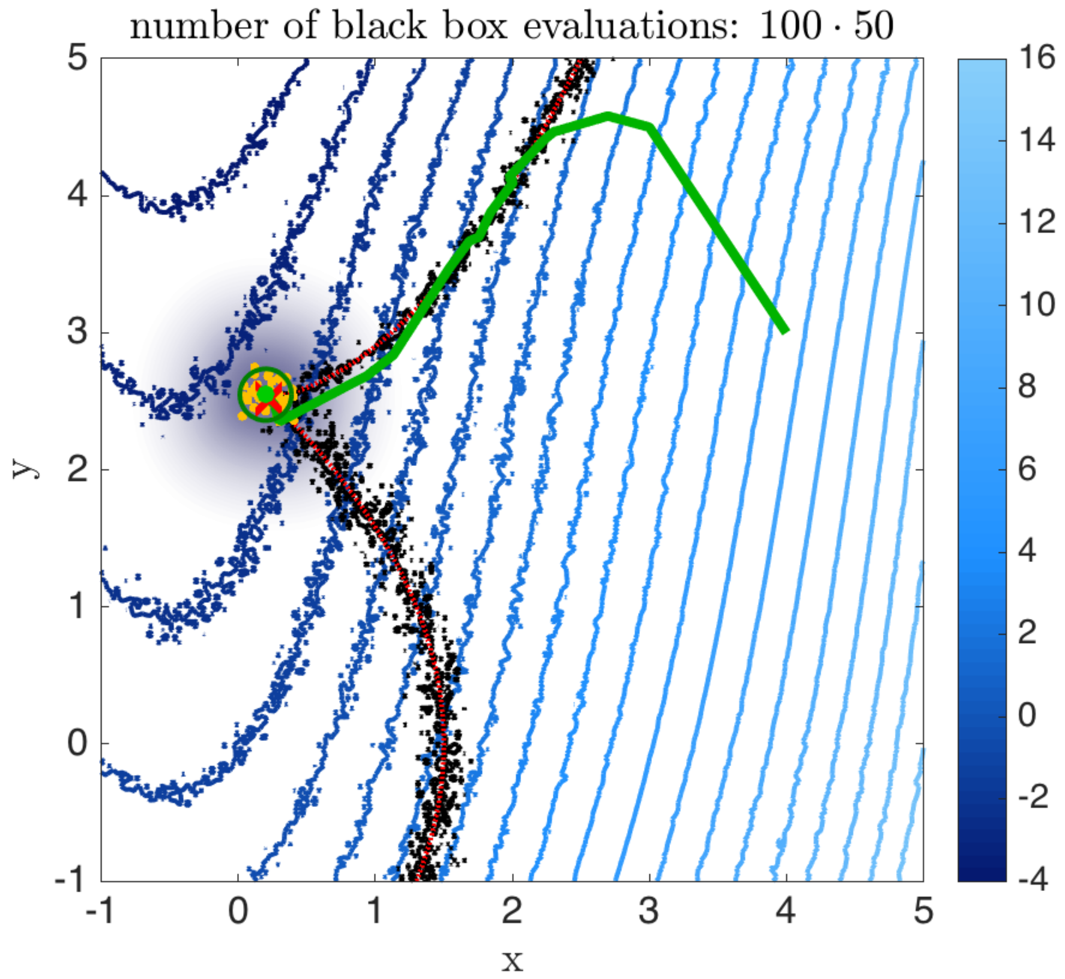


SNOWPAC – Example



Scientific Computing
Department of Informatics
Technical University of Munich

SNOWPAC – Example



SNOWPAC – Benchmark setup

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

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

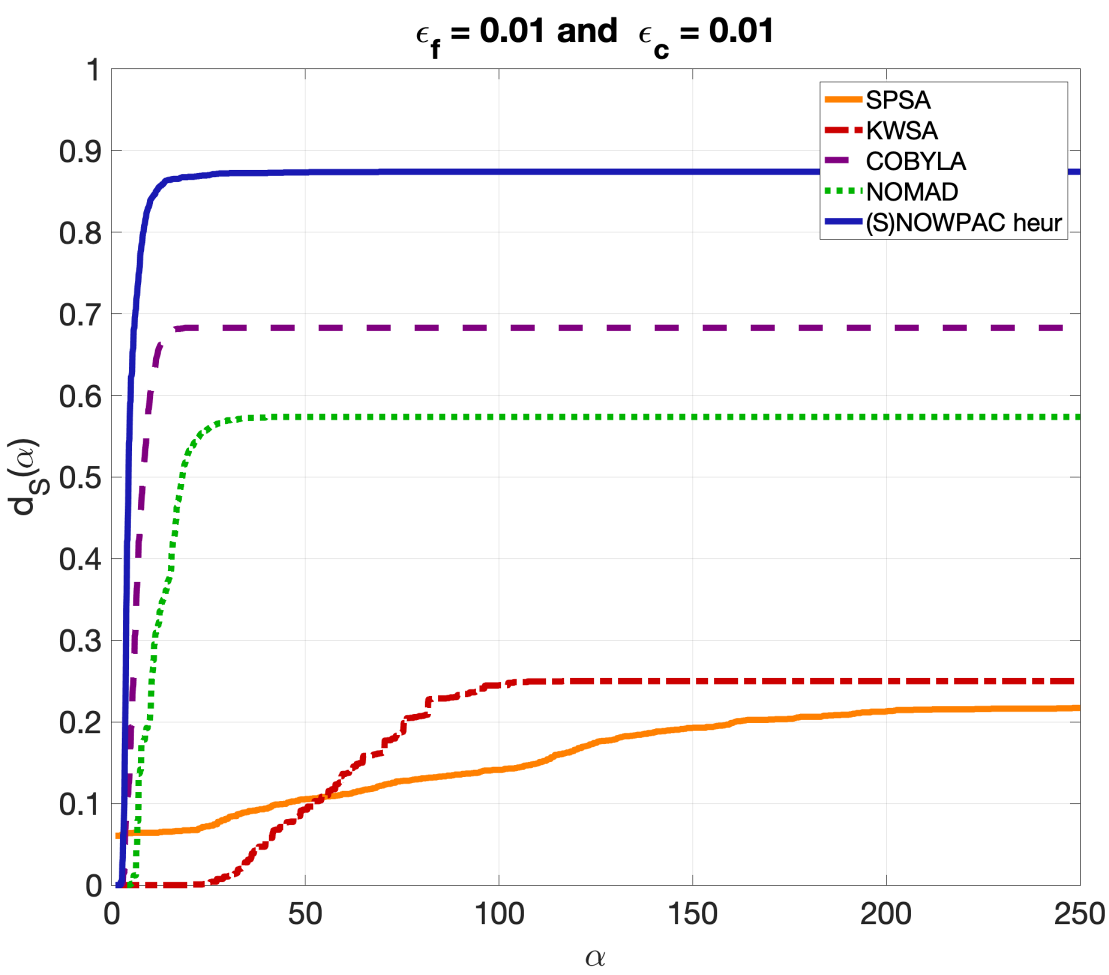
  and approximate robustness measures with $N \in \{200, 1000, 2000\}$ samples of

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

- Limit max number of black box evaluations to $1000N$
- Comparison of results from 100 repeated optimization runs
- Use data profile [Moré/Wild2009] to compare performance $d_S(\alpha) = \frac{1}{2400}\left|\left\{p \in \mathscr{P} \;:\; \frac{t_{p,S}}{n_p+1} \leq \alpha\right\}\right|$
  - Based on $|\mathscr{P}| = 8 \cdot 100 \cdot 3 = 2400$ optimization runs

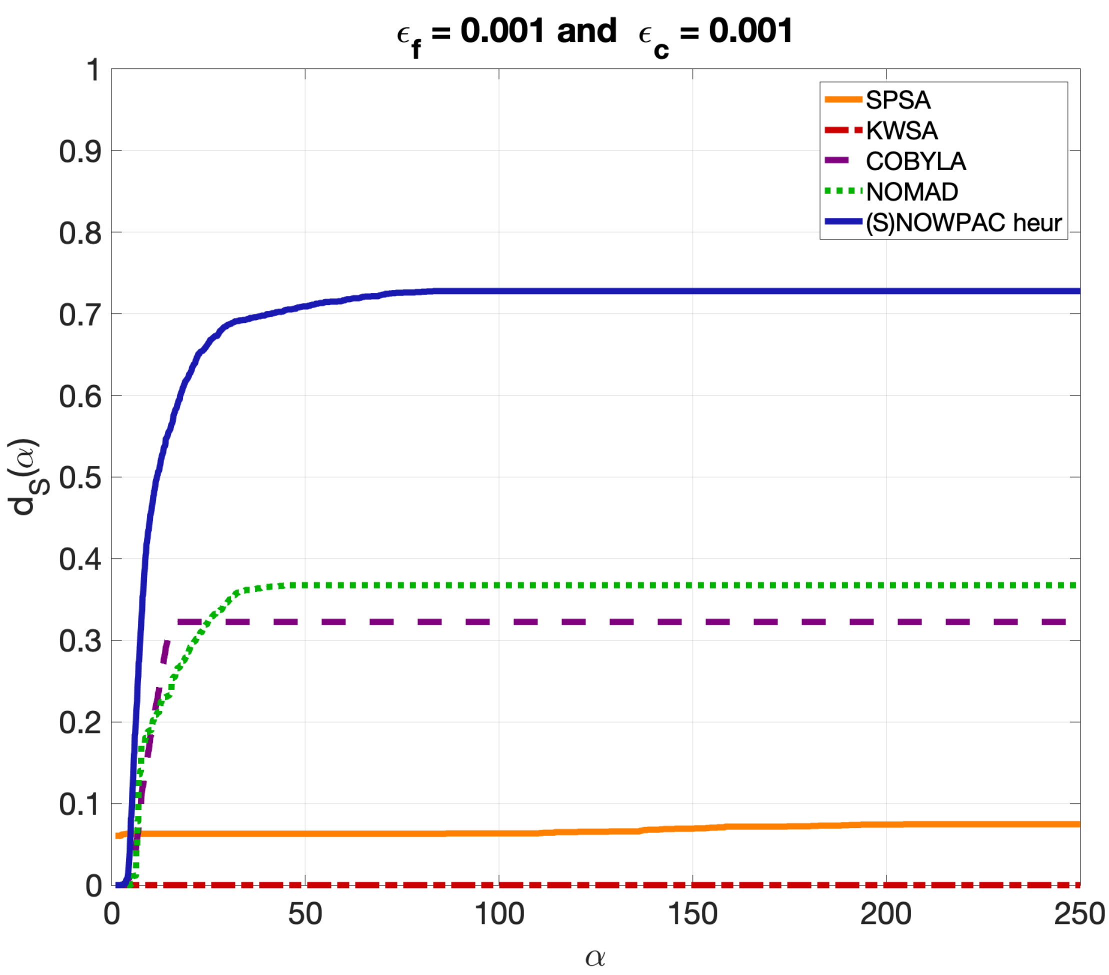
Scientific Computing
Department of Informatics
Technical University of Munich

SNOWPAC – Benchmark results

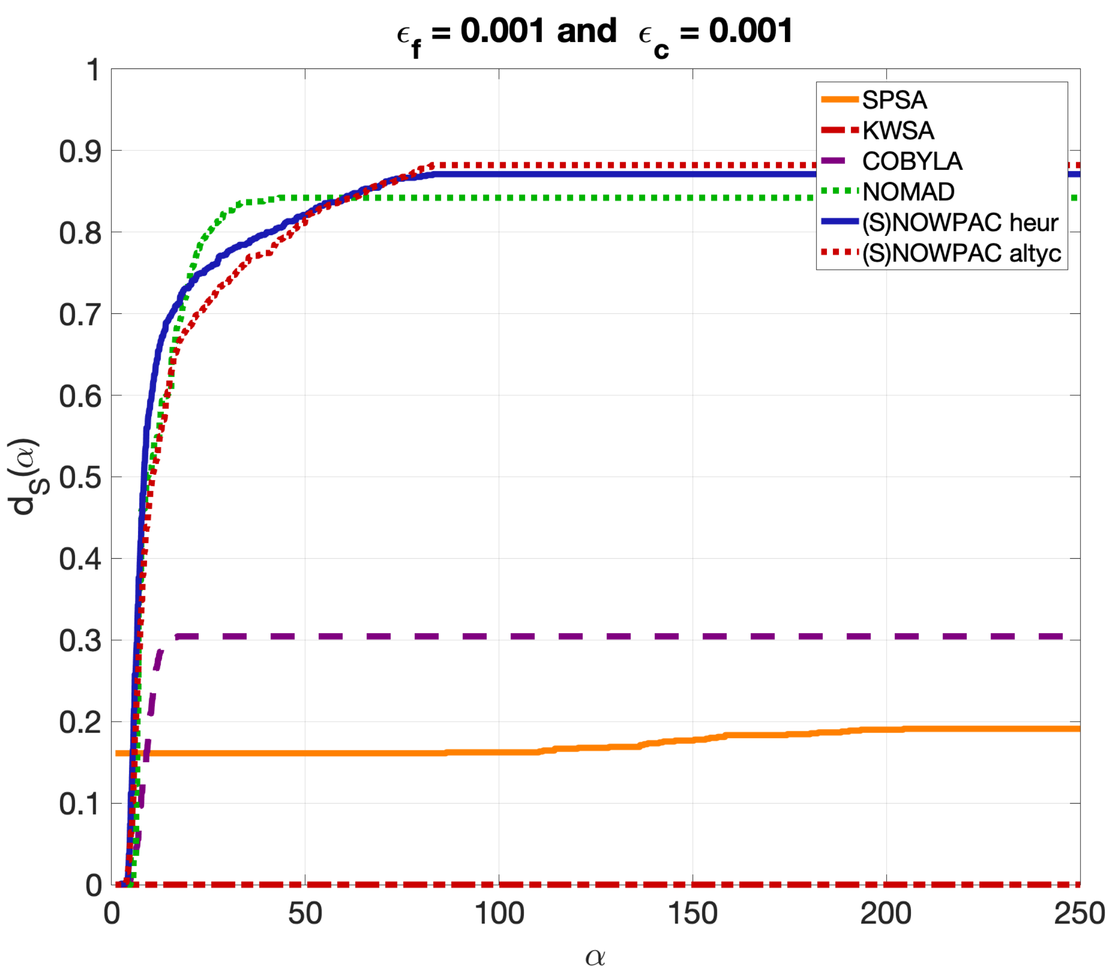


Scientific Computing
Department of Informatics
Technical University of Munich

SNOWPAC – Benchmark results

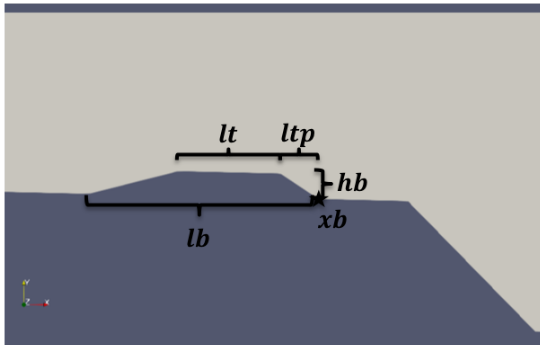


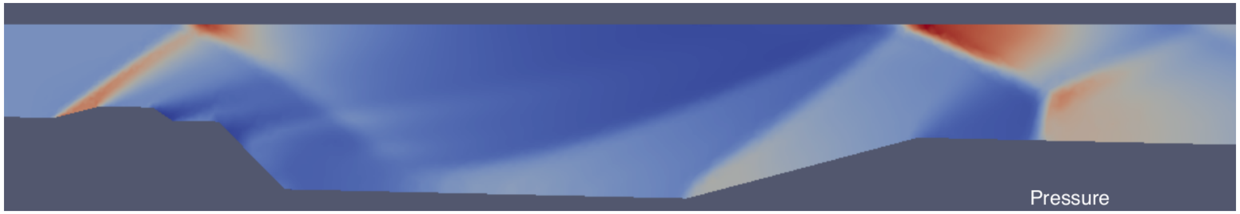
SNOWPAC – Benchmark results



APPLICATION RESULTS

SU2 Optimization – Problem formulations





Meshes: Fine: ∼18000, Medium: ∼5000, Coarse: ∼2500 elements

Deterministic (NOWPAC)

$$\min_{\mathbf{x}} p_{\text{loss}}(\mathbf{x})$$
$$\text{s.t. } 0.51 \leq T_{\text{cav}}(\mathbf{x})$$

Robust OUU MC and MLMC (SNOWPAC)

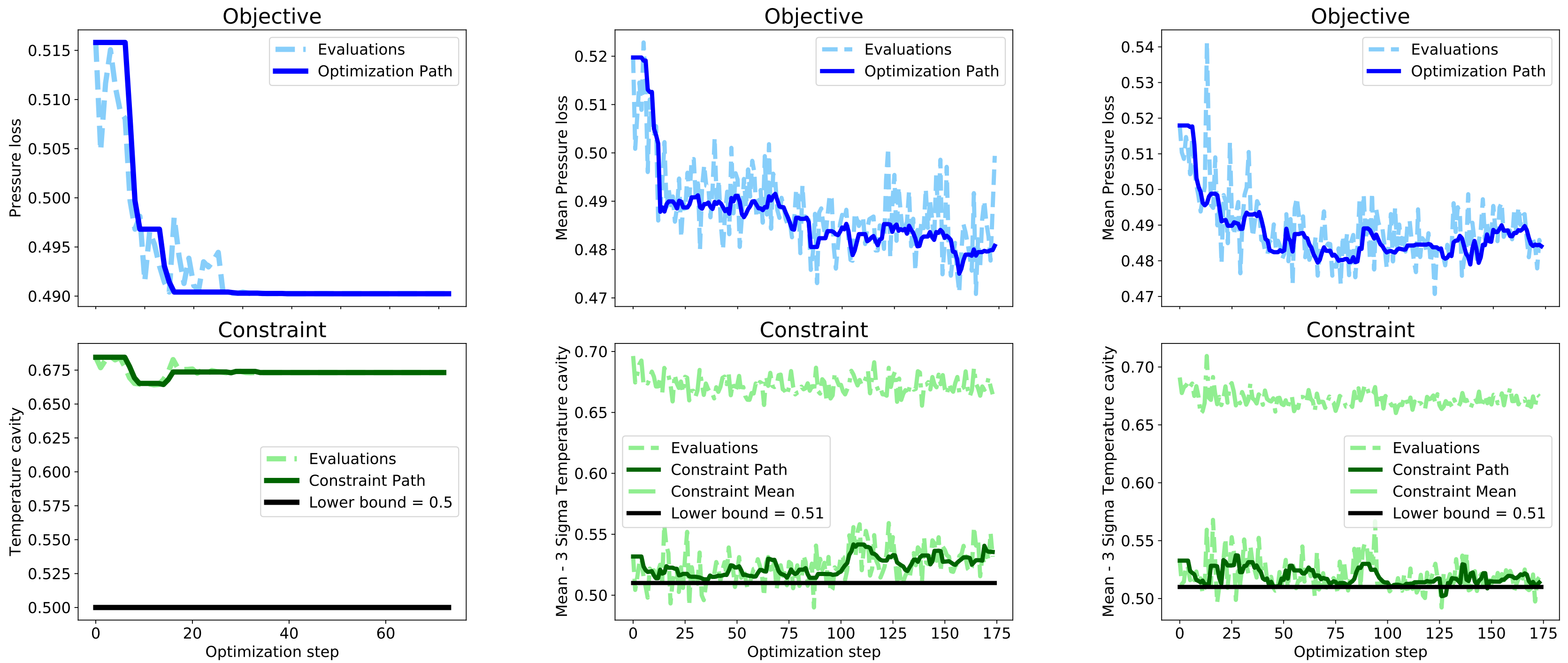
$$\min_{\mathbf{x}} \mathbb{E}[p_{\text{loss}}(\mathbf{x}, \omega)]$$
$$\text{s.t. } 0.51 \leq \mathbb{E}[T_{\text{cav}}(\mathbf{x}, \omega)] - 3\sigma[T_{\text{cav}}(\mathbf{x}, \omega)]$$

Constraints on design and distributions on uncertain parameters:

- $0.5 \leq hb \leq 2.5$
- $7.5 \leq lt \leq 11.5$
- $2.5 \leq ltp \leq 4.5$
- $17.5 \leq lb \leq 20.5$
- $79.5 \leq xb \leq 85.5$

- $p_{0,in} \sim \mathscr{U}(1.332e6, 1.628e6)$
- $T_{0,in} \sim \mathscr{U}(1.395e3, 1.705e3)$
- $M_{in} \sim \mathscr{U}(2.259e0, 2.761e0)$

SU2 Optimization – Results

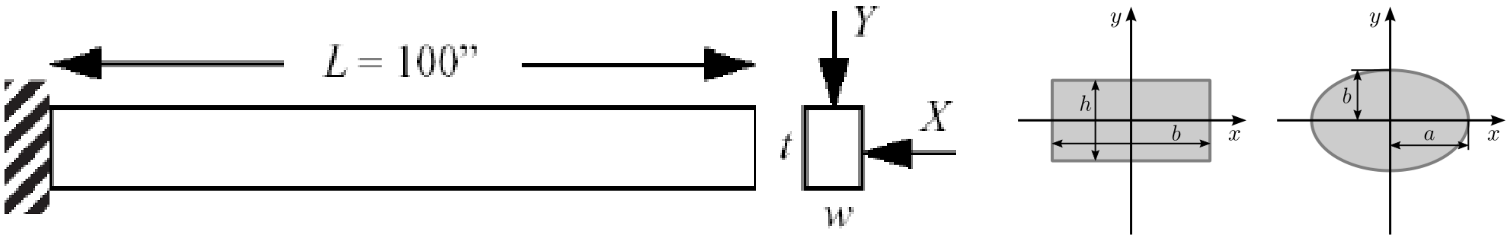


Deterministic | OUU MC | OUU MLMC

Problem	$\mathbf{x}^*$	$\mathbf{p}^*_{\text{loss}}$	$\mathbf{c}^*$	**Cost:** Eval (C, M, F)	~**Cost:** Time (5s, 10s, 5m)
DET	$[1.46, 11.25, 2.51, 17.62, 79.50]$	0.4902	0.6732	(0, 0, 73)	6 h
OUU MC	$[2.06, 8.63, 3.84, 20.5, 79.5]$	0.4807	0.5355	(0, 0, 9396)	800 h
OUU MLMC	$[2.37, 10.63, 3.52, 20.5, 79.75]$	0.4840	0.5138	(34626, 4350, 870)	150 h

Scientific Computing
Department of Informatics
Technical University of Munich

Cantilever Optimization – Problem formulations



Rectangle

$$\min \quad wt$$

$$\text{s.t. } D_R = \frac{\frac{4L^3}{Ewt}\sqrt{(\frac{D_y}{t^2})^2 + (\frac{D_x}{w^2})^2}}{D_0} - 1 \leq 0$$

$$\sigma_R = \frac{\frac{6L}{wt^2}D_y + \frac{6L}{w^2 t}D_x}{R} - 1 \leq 0$$

Ellipse

$$\min \quad wt$$

$$\text{s.t. } D_E = \frac{\sqrt{(\frac{D_x L}{3E\frac{\pi ab^3}{4}})^2 + (\frac{D_y L}{3E\frac{\pi a^3 b}{4}})^3}}{D} - 1 \leq 0$$

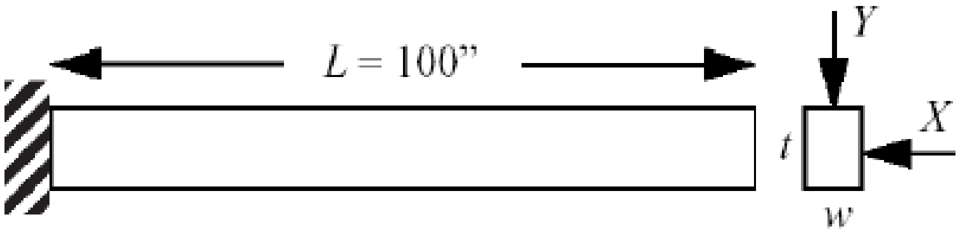
$$\sigma_E = \frac{\frac{4L}{\pi ab}\sqrt{\frac{D_y^2}{a^2} + \frac{D_x^2}{b^2}}}{R} - 1 \leq 0, a = \frac{2t}{\pi}, b = \frac{w}{2}$$

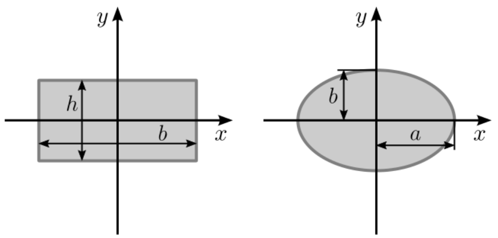
For Multilevel:

- Level 1
- Cost := 1
- $N_1 = 25$

- Level 0
- Cost := 0.01
- $N_0 = 550 + 25$

Cantilever Optimization – Problem formulations





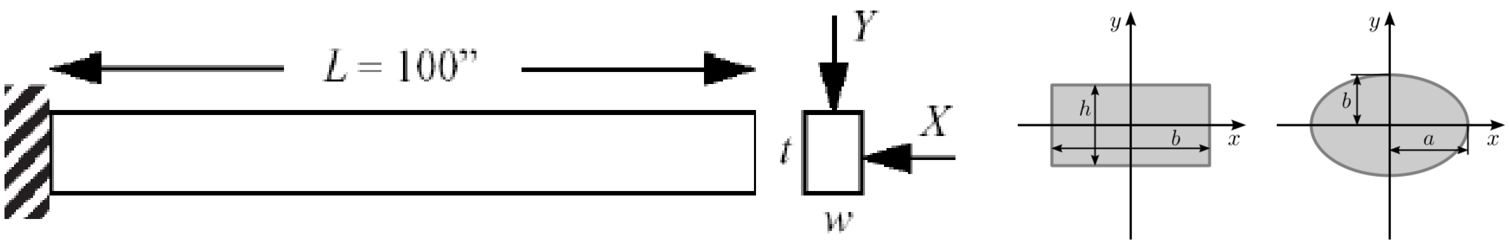
Deterministic (NOWPAC)

$$\begin{aligned} &\min wt \\ \text{s.t. } &D_R \leq 0 \\ &\sigma_R \leq 0 \end{aligned}$$

Robust OUU MC and MLMC (SNOWPAC)

$$\begin{aligned} &\min wt \\ \text{s.t. } &\mathbb{E}[D(w,t)] + 3\mathbb{V}^{\frac{1}{2}}[D(w,t)] \leq 0 \\ &\mathbb{E}[\sigma(w,t)] + 3\mathbb{V}^{\frac{1}{2}}[\sigma(w,t)] \leq 0 \end{aligned}$$

Cantilever Optimization – Problem formulations



Deterministic (NOWPAC)

$$\begin{aligned} &\min wt \\ \text{s.t. } &D_R \leq 0 \\ &\sigma_R \leq 0 \end{aligned}$$

Robust OUU MC and MLMC (SNOWPAC)

$$\begin{aligned} &\min wt \\ \text{s.t. } &\mathbb{E}[D(w,t)] + 3\mathbb{V}^{\frac{1}{2}}[D(w,t)] \leq 0 \\ &\mathbb{E}[\sigma(w,t)] + 3\mathbb{V}^{\frac{1}{2}}[\sigma(w,t)] \leq 0 \end{aligned}$$

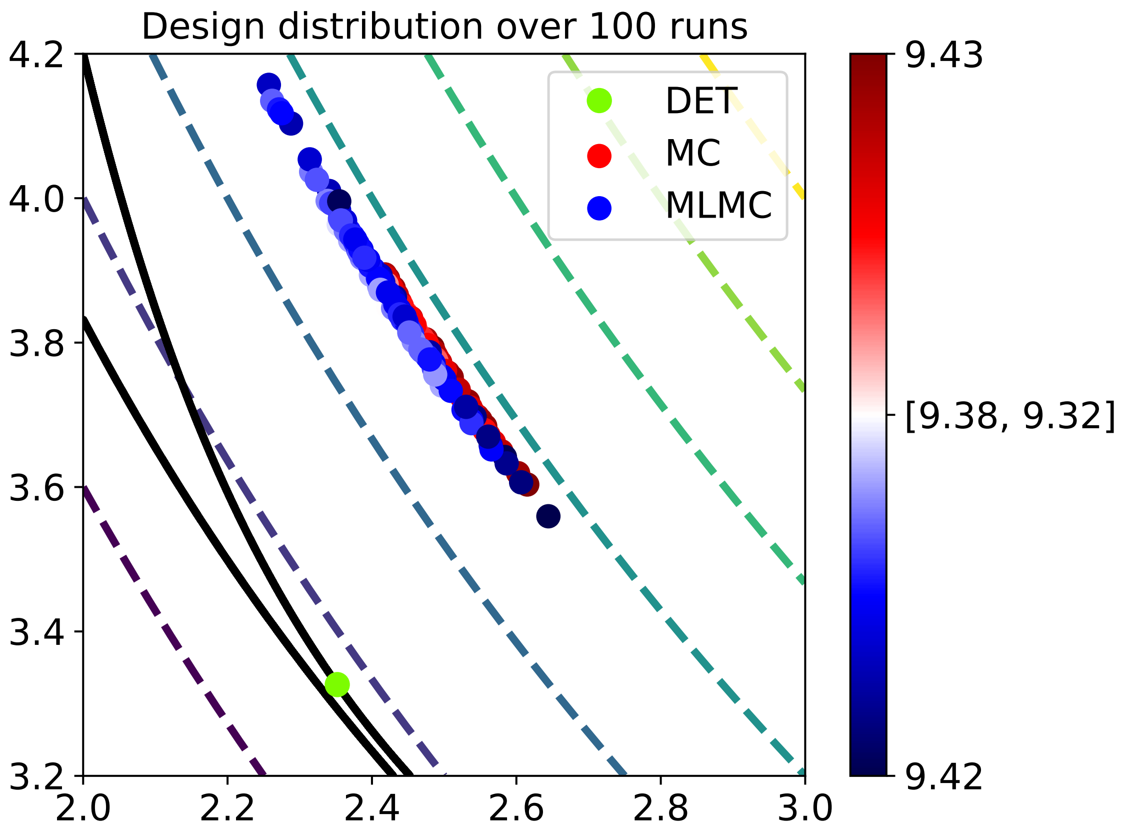
Constraints on design and distributions on uncertain parameters:

- $1.0 \leq w \leq 10$
- $1.0 \leq t \leq 10$
- $L = 100[\text{in}]$
- $D = 2.2535[\text{in}]$

- $R \sim \mathcal{N}(40000, 2000^2)$
- $E \sim \mathcal{N}(2.9e7, 1.45e6^2)$
- $D_x \sim \mathcal{N}(500, 100^2)$
- $D_y \sim \mathcal{N}(1000, 100^2)$

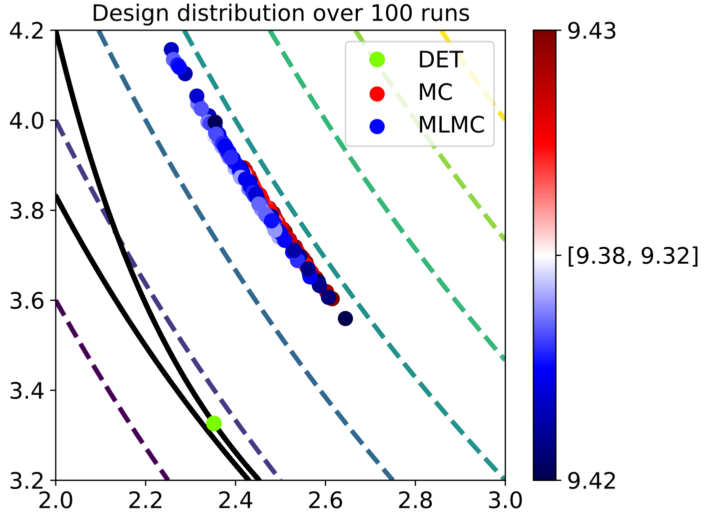
Scientific Computing
Department of Informatics
Technical University of Munich

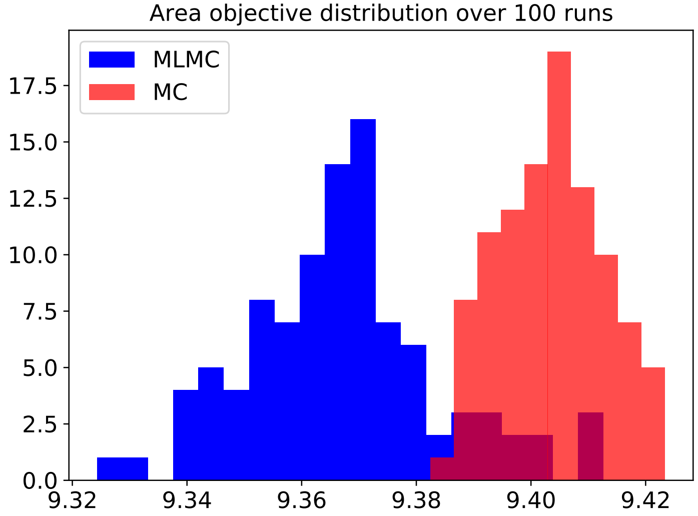
Cantilever Optimization – Results



Scientific Computing
Department of Informatics
Technical University of Munich

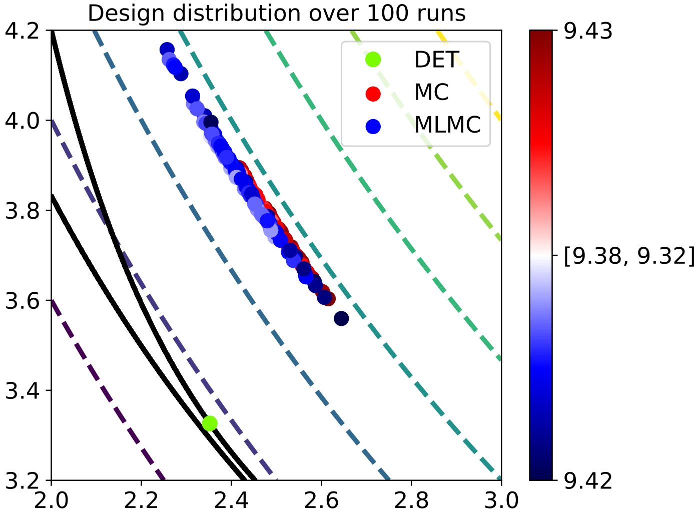
Cantilever Optimization – Results

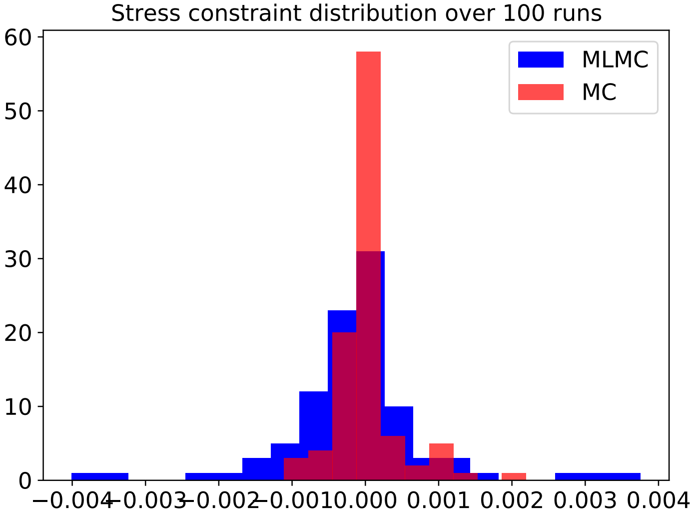




Scientific Computing
Department of Informatics
Technical University of Munich

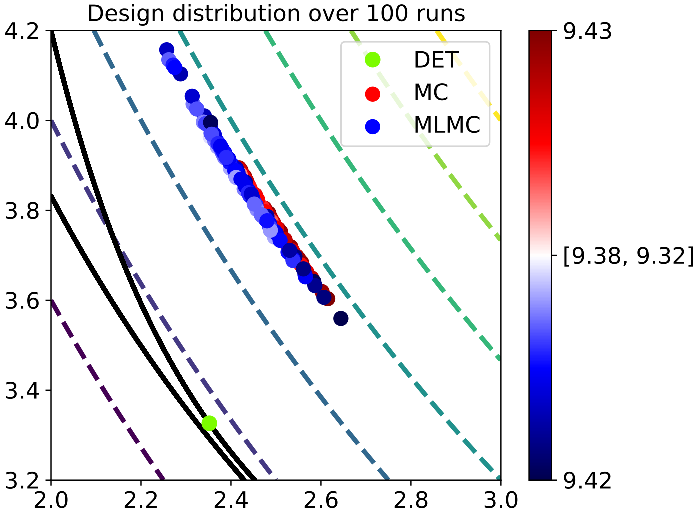
Cantilever Optimization – Results

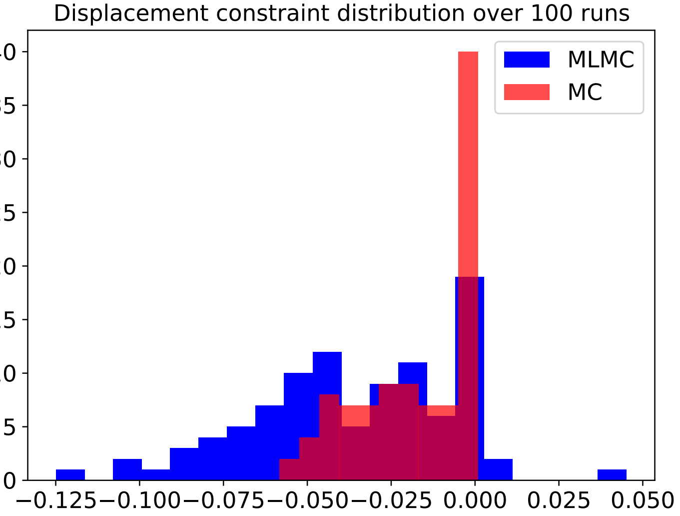




Scientific Computing
Department of Informatics
Technical University of Munich

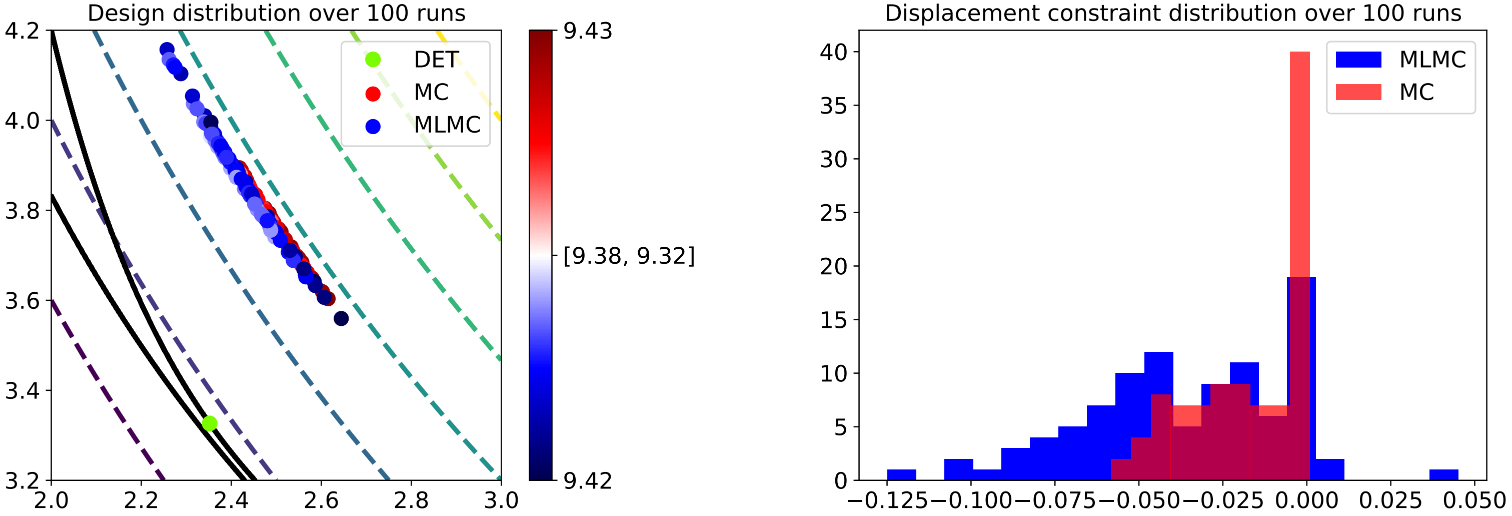
Cantilever Optimization – Results





Scientific Computing
Department of Informatics
Technical University of Munich

Cantilever Optimization – Results



Problem	$[\mathbf{w}^*, \mathbf{t}*]$	$\mathbf{w}^* \cdot \mathbf{t}^*$	$\mathbf{c}_1^*$	$\mathbf{c}_2^*$	**Cost:** Eval (R, E)	**Costratio**
DET	$[2.35, 3.32]$	7.82	$-1.60e-2$	$-7.52e-7$	(56, 0)	2.24e-3
OUU MC	$[2.48, 3.80]$	9.40	$7.60e-6$	$-2.19e-2$	(25000, 0)	1
OUU MLMC	$[2.43, 3.85]$	9.36	$-9.02e-4$	$-8.47e-2$	(6250, 137500)	0.3075

Summary:

- **NOWPAC** – Derivative-free trust region methods for constrained nonlinear optimization
- **SNOWPAC** – Stochastic derivative-free optimization using Gaussian process surrogates

$\Rightarrow$ New analytic approach for noise reduction
- **DAKOTA** – Design Analysis Kit for Optimization and Terascale Applications

$\Rightarrow$ 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