Near-optimal smoothing in derivative-free stochastic optimization

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

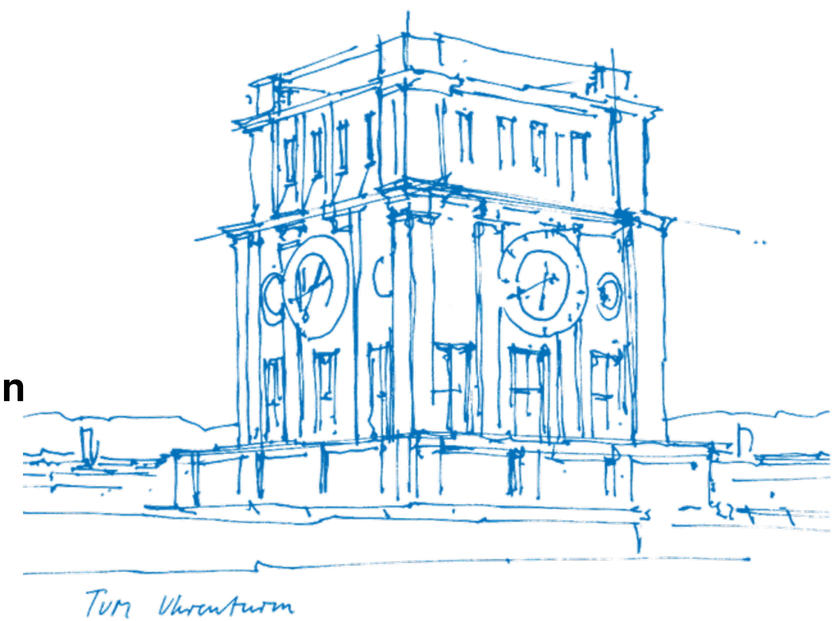
menhorn@in.tum.de

[1]Technical University of Munich, GER

[2]Massachusetts Institute of Technology, USA

[3]Mathworks, USA

[4]Sandia National Laboratories, USA



UQOP: Uncertainty Quantification and OPtimization

Optimization under Uncertainty I

Paris, March 18th, 2019

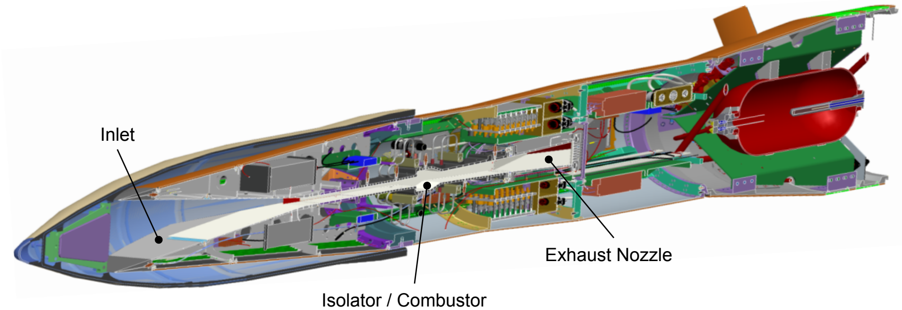
Scientific Computing
Department of Informatics
Technical University of Munich

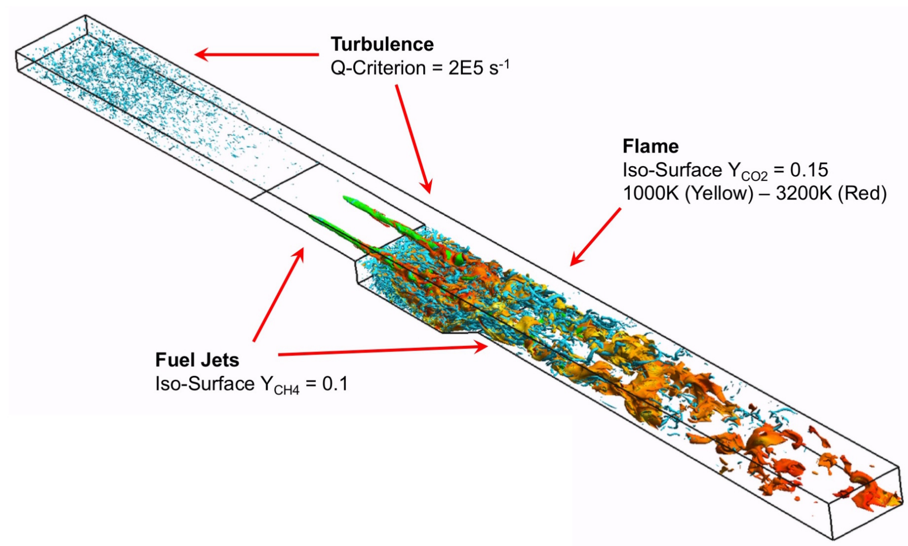
Motivation: Design optimization of a SCRAMJET

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

Scientific Computing
Department of Informatics
Technical University of Munich

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

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

Features of SNOWPAC:

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

SNOWPAC [1]

Robust optimization problem statement

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

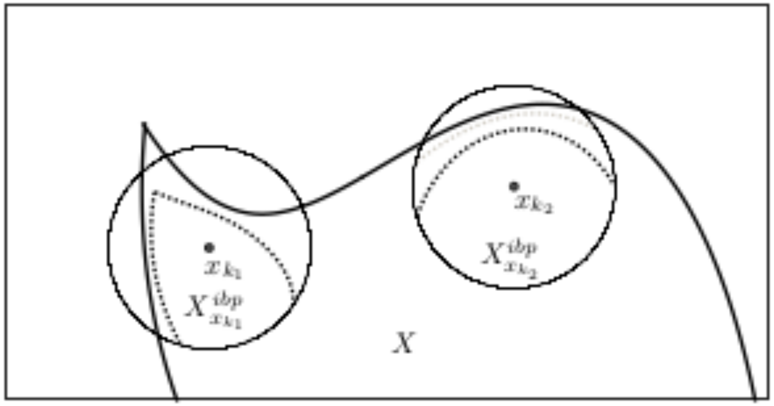
Features of SNOWPAC:

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

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

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

   NEW: Leverage multilevel estimators.

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

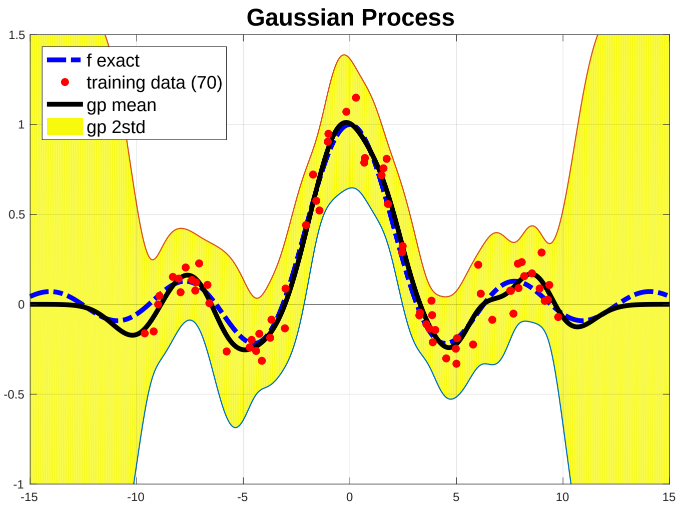
0. **Extension of NOWPAC**: Derivative-free nonlinear constraint optimization method using trust-regions (deterministic)
1. **Estimate robustness measures**: Use sampling, e.g. $\mathscr{R}^f_\omega = \mathbb{E}[f_\omega(\mathbf{x})] \approx R^f = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}, \theta_i) + \varepsilon_N$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



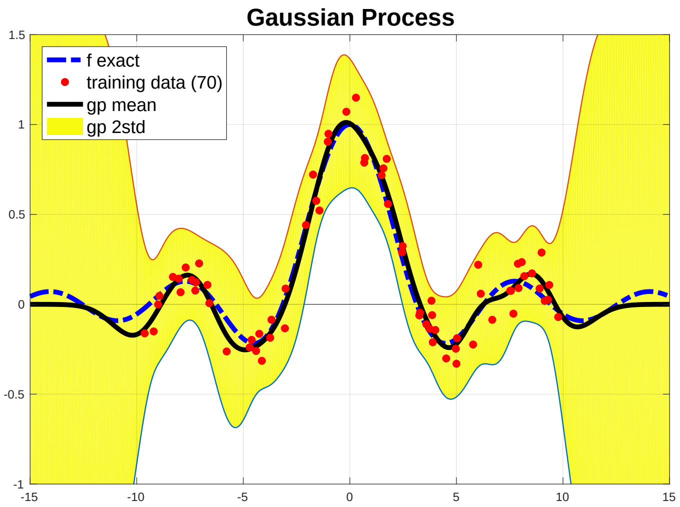
Scientific Computing
Department of Informatics
Technical University of Munich

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**:

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



Scientific Computing
Department of Informatics
Technical University of Munich

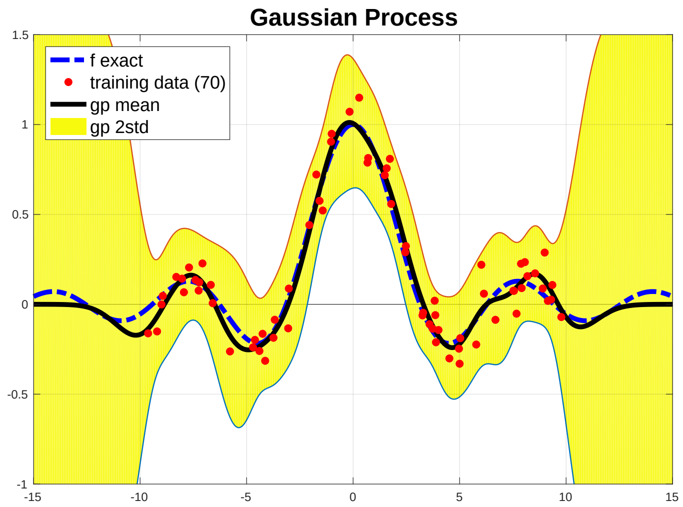
SNOWPAC – Gaussian process surrogate

Build Gaussian process surrogate

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

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

- **Replace noise estimate by**:



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

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

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 $\Rightarrow \Delta_{k+1} \geq \sqrt{\lambda_t \varepsilon_N}$
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**

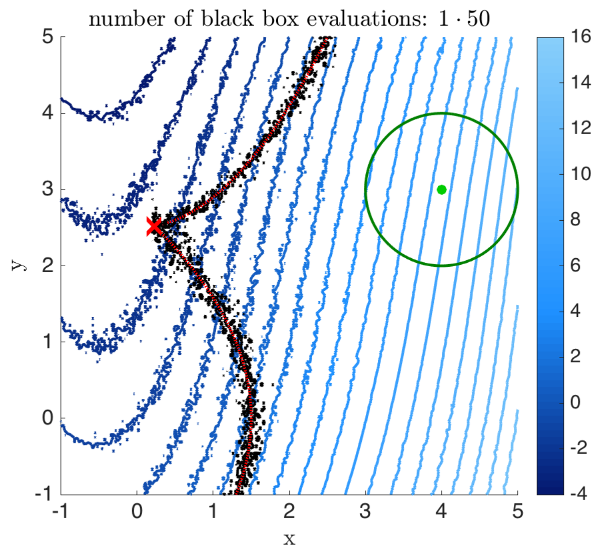
$\Rightarrow$ Feasibility restoration mode: $$\min_{\substack{m_k^c(\mathbf{x})\leq\tau \\ \|\mathbf{x}-\mathbf{x}_k\|\leq\Delta_k}} \sum_{i\in\mathscr{I}} \left(m_k^{c_i}(\mathbf{x})^2 + \lambda_g m_k^{c_i}(\mathbf{x})\right)$$

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

Scientific Computing
Department of Informatics
Technical University of Munich

SNOWPAC – Example

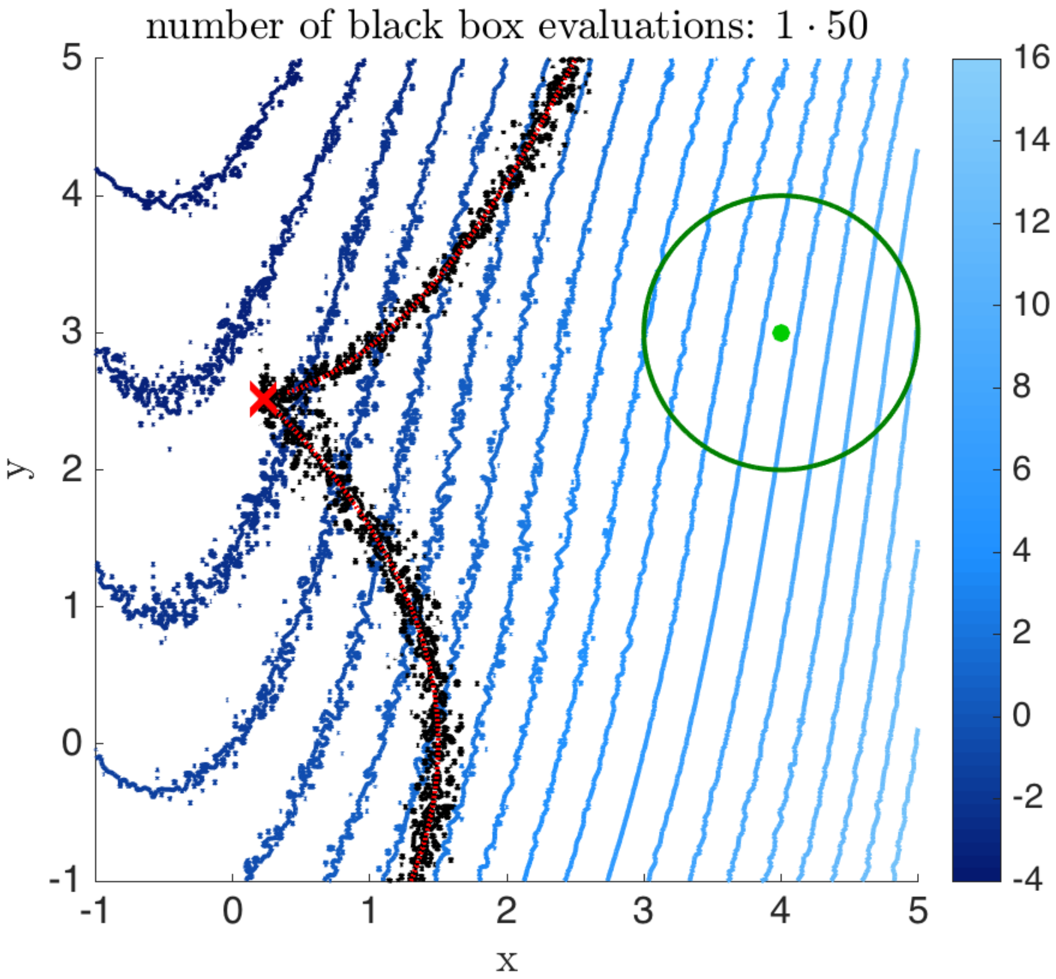
$$\min \mathbb{E}[\sin(x-1+\theta_1)+\sin(\frac{1}{2}y-1+\theta_1)^2]+\frac{1}{2}(x+\frac{1}{2})^2-y$$
$$\text{s.t.} \quad \mathbb{E}[-4x^2(1+\theta_2)-10\theta_3] \quad \leq 25-10y, \theta_i \sim \mathscr{U}(\theta_i|-1,1), i=1,\ldots 4 \tag{1}$$
$$\mathbb{E}[-2y^2(1+\theta_4)-10(\theta_4+\theta_2)]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



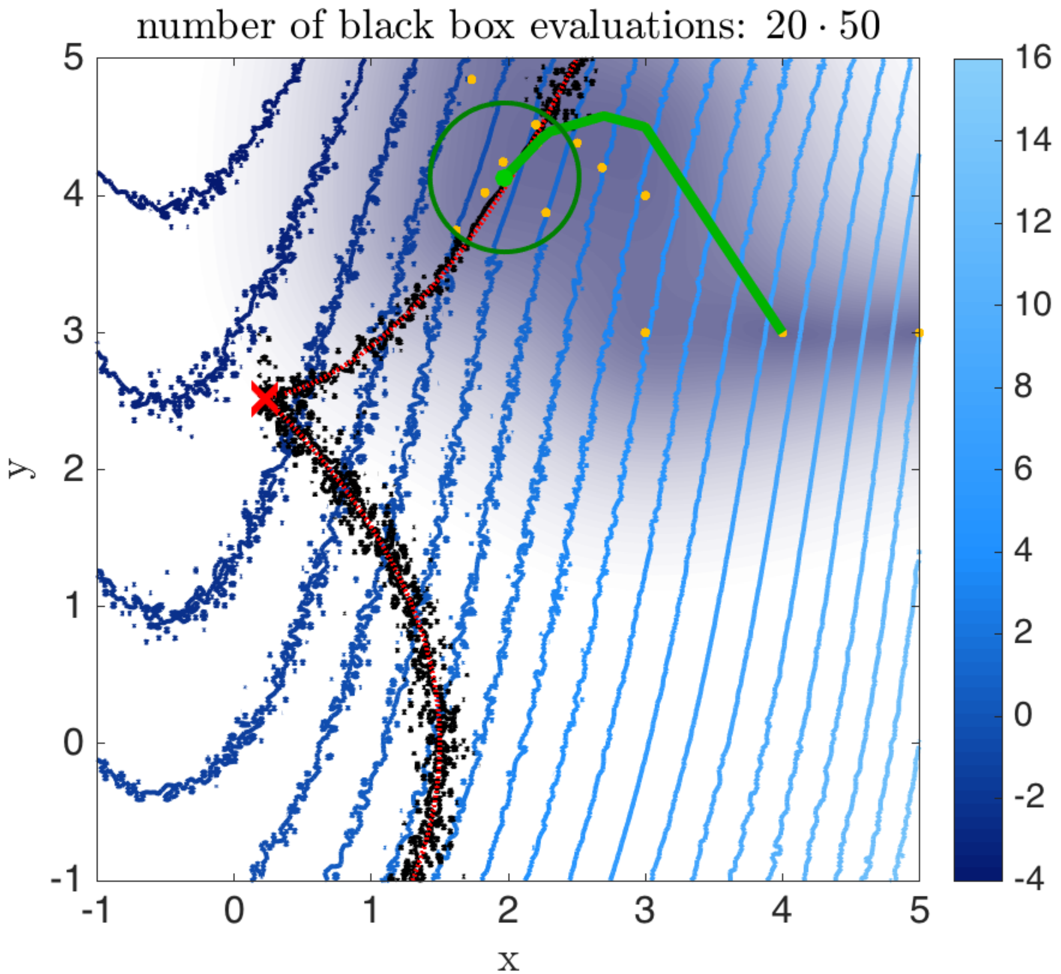
- Locally smoothed black box functions within the trust region
- Optimal design (red cross), exact constraints (red dotted lines)
- Objective (blue lines), constraints (black lines)
- Current design and trust region (green dot and circle)
- GP points (yellow dots), scaling factor γ (gray shade)

Scientific Computing
Department of Informatics
Technical University of Munich

SNOWPAC – Example

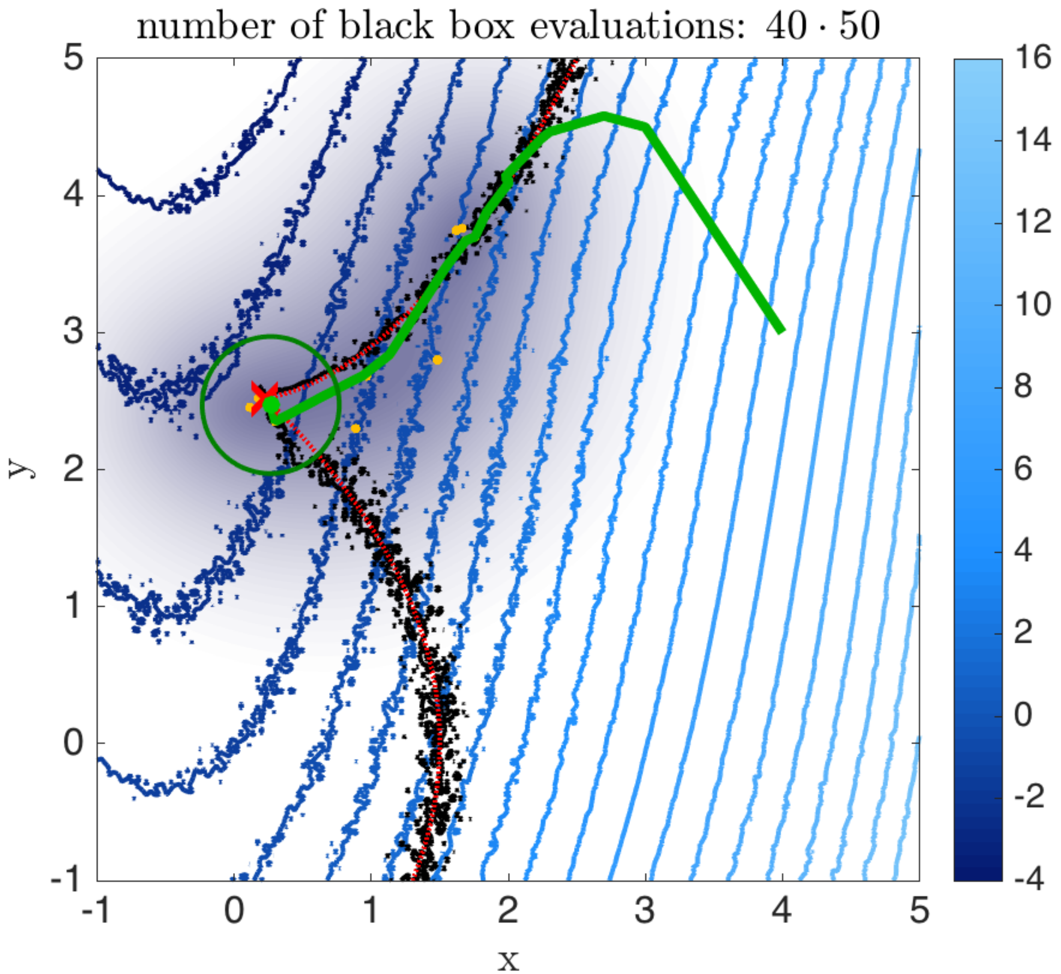


SNOWPAC – Example

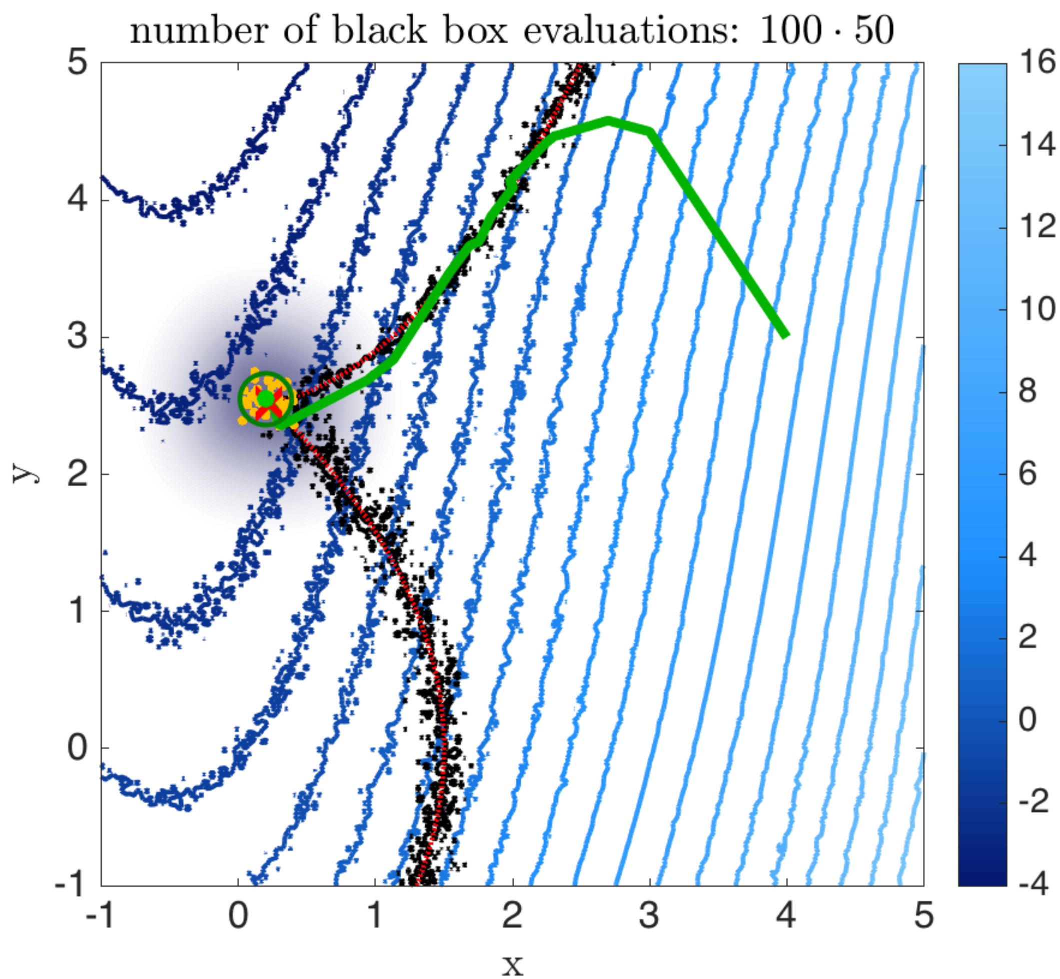


Scientific Computing
Department of Informatics
Technical University of Munich

SNOWPAC – Example



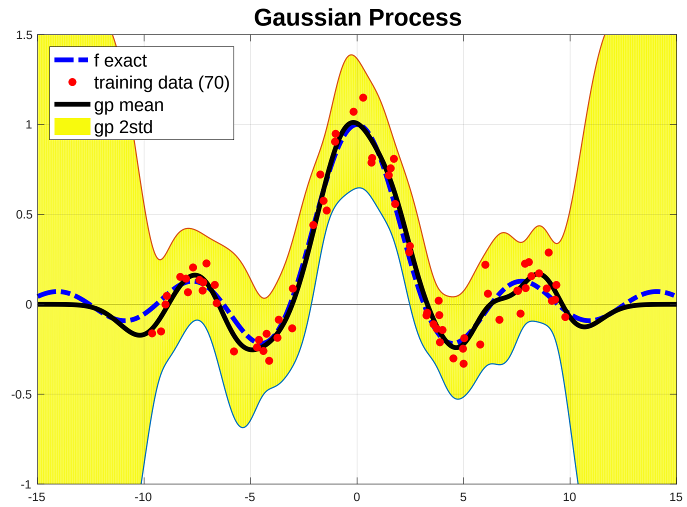
SNOWPAC – Example



Near-optimal smoothing in SNOWPAC

SNOWPAC – Near-optimal smoothing

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**:

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

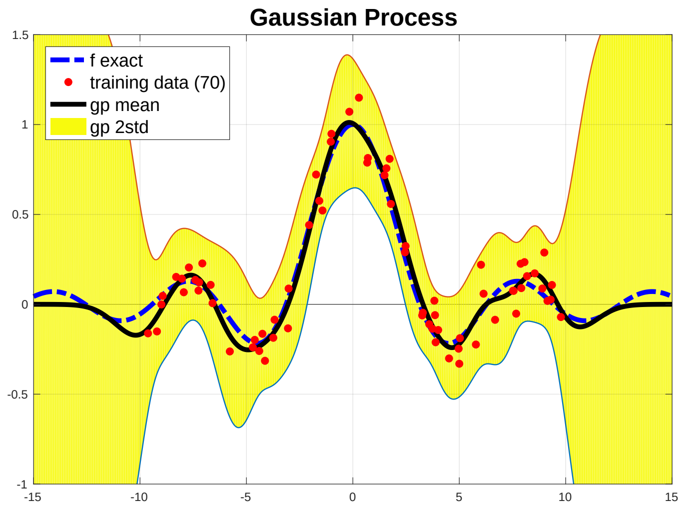
with

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

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

SNOWPAC – Near-optimal smoothing

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: $\mu_{\text{GP}}(\mathbf{x}) = \mathbf{k}_{\mathbf{x}X}[\mathbf{K}_{XX} + \sigma_\mathbf{n}^2 \mathbf{I}]^{-1}\mathbf{R}$

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

SNOWPAC – Near-optimal smoothing

MSE:

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

SNOWPAC – Near-optimal smoothing

MSE:

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

Optimal α:

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

Scientific Computing
Department of Informatics
Technical University of Munich

SNOWPAC – Near-optimal smoothing

MSE:

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

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:

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

SNOWPAC – Near-optimal smoothing

MSE:

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

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:

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

SNOWPAC – Near-optimal smoothing

MSE:

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

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 – Near-optimal smoothing

MSE:

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

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:

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

SNOWPAC – Near-optimal smoothing

MSE:

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

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:

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

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

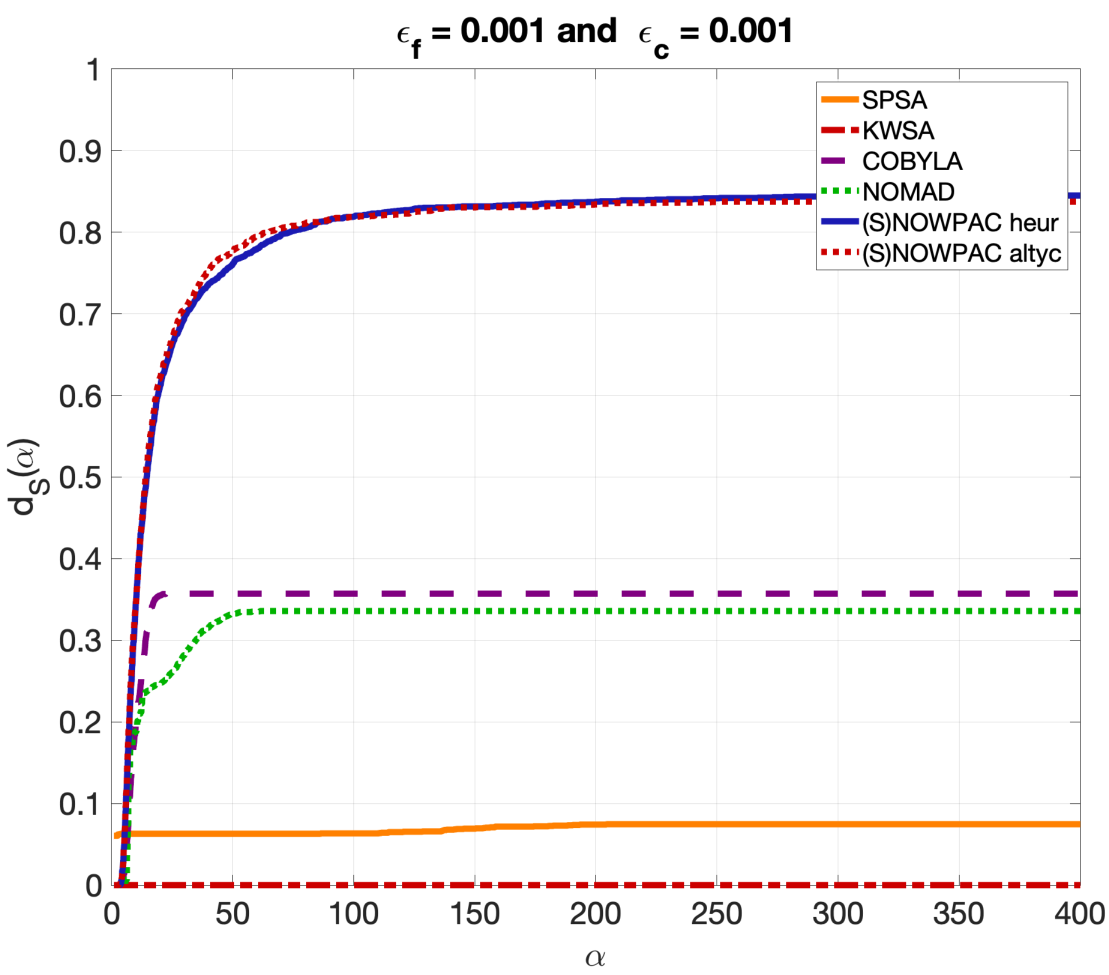
$$\min R_N[f(\mathbf{x}) + \omega_1]$$
$$\text{s.t. } R_N[c_i(\mathbf{x}) + \omega_{2,i}] \leq 0,$$

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

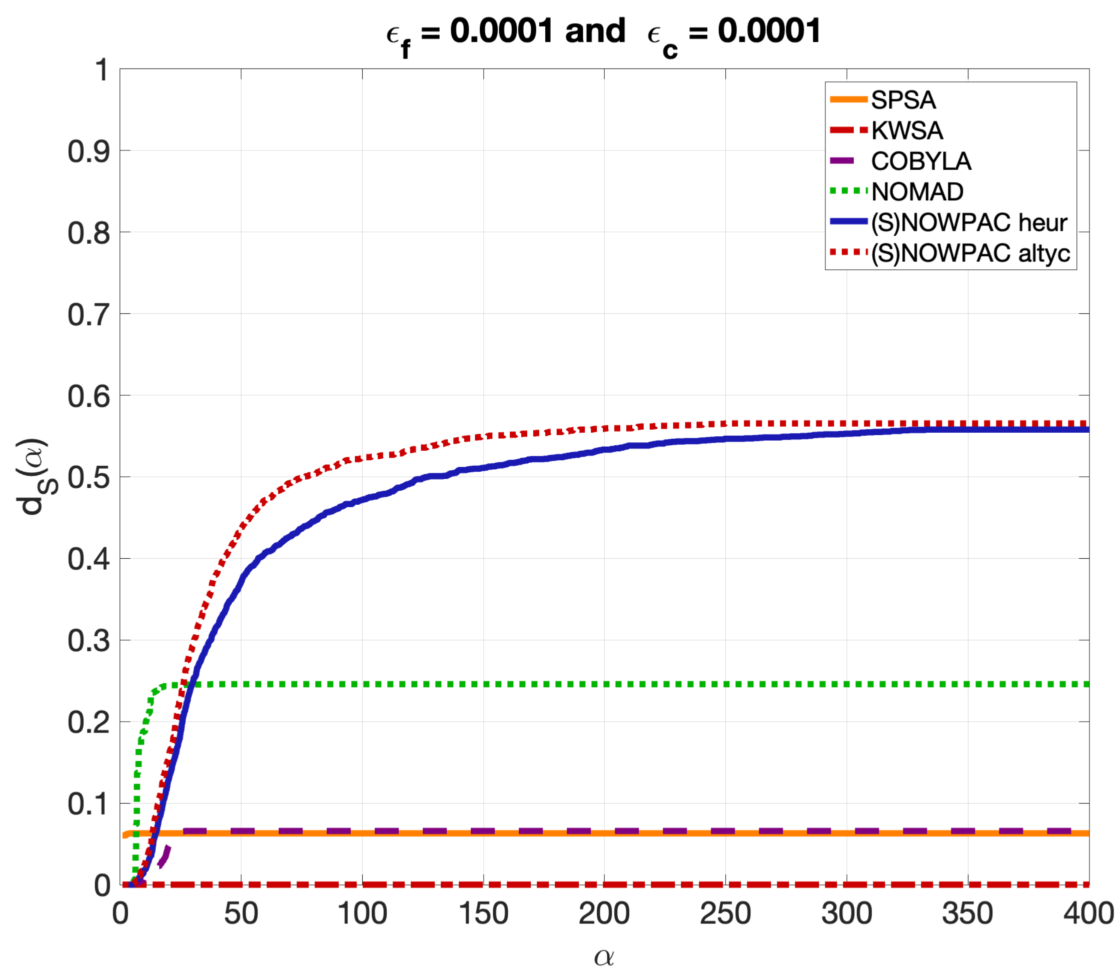
$$\omega_1, \omega_{2,i} \sim \mathscr{U}[-1, 1]$$

- 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

SNOWPAC – Benchmark results



SNOWPAC – Benchmark results



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