Scientific Computing
Department of Informatics
Technical University of Munich

SAND2020-13851C

Multifidelity strategies for optimization under uncertainty of wind power plants

Friedrich Menhorn[1], Gianluca Geraci[2], Daniel T. Seidl [2], Michael S. Eldred[2], Ryan N. King [3], Hans-Joachim Bungartz[1], Youssef M. Marzouk[4]

menhorn@in.tum.de

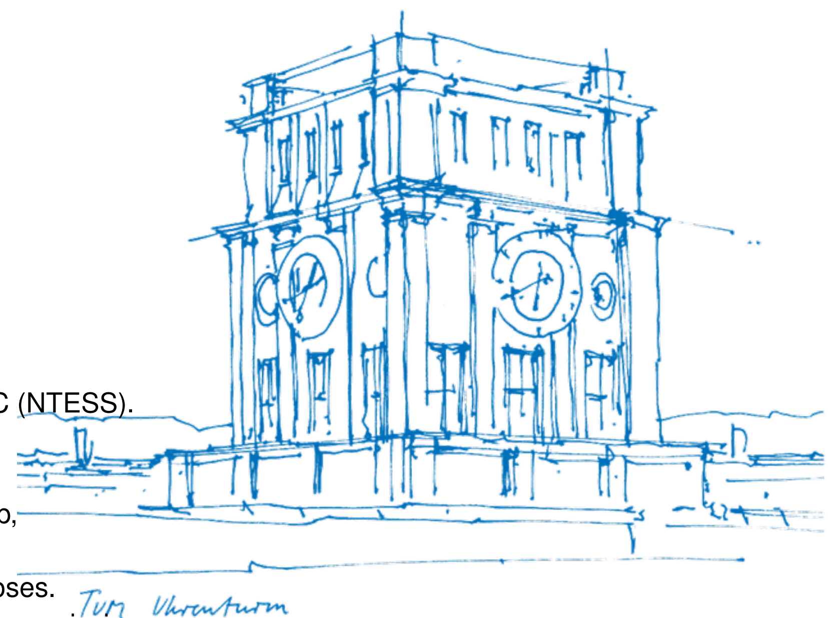
[1]Technical University of Munich, GER

[2]Sandia National Laboratories, USA

[3]National Renewable Laboratory, USA

[4]Massachusetts Institute of Technology, USA

2021 AIAA SciTech Forum
NDA-14/WE-09
Uncertainty Analysis Advancements
for Wind Energy Applications
Virtual, Jan 19th, 2021



Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Scientific Computing
Department of Informatics
Technical University of Munich

Motivation: Design optimization of a wind power plant

Provided by NREL

Vattenfall's Horns Rev wind farm off Denmark*

- Wake steering scenario
- Uncertain inputs
- Maximize total power production
- Black box code

*Figure from https://www.rechargenews.com/wind/will-wind-wake-slow-industrys-ambitions-offshore-/2-1-699430

SNOWPAC*

OUU problem statement

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

*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*

OUU problem statement

$$\boxed{\mathscr{R}_\theta^* = \mathscr{R}^f(\mathbf{x}^*, \theta) = \min_{\mathscr{R}^c(\mathbf{x},\theta)\le 0} \mathscr{R}^f(\mathbf{x}, \theta)}$$

Features of SNOWPAC:

*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*

OUU problem statement

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

Features of SNOWPAC:

0. **Extension of NOWPAC**: Deterministic derivative-free nonlinear constraint optimization method using trust-regions

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

SNOWPAC*

OUU problem statement

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

Features of SNOWPAC:

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

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

SNOWPAC*

OUU problem statement

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

Features of SNOWPAC:

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

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

SNOWPAC*

OUU problem statement

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

Features of SNOWPAC:

0. **Extension of NOWPAC**: Deterministic derivative-free nonlinear constraint optimization method using trust-regions
1. **Estimate robustness measures**: Use sampling, e.g. $\mathscr{R}^f_\theta = \mathbb{E}[f_\theta(\mathbf{x})] \approx R^f = \frac{1}{N}\sum_{i=1}^N f(\mathbf{x}, \theta_i) + \varepsilon_N$
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 $\tilde{\varepsilon} = \alpha \cdot 2\sigma_{\text{GP}}(\mathbf{x}) + (1-\alpha) \cdot \varepsilon_N$

*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*

OUU problem statement

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

Features of SNOWPAC:

0. **Extension of NOWPAC**: Deterministic derivative-free nonlinear constraint optimization method using trust-regions
1. **Estimate robustness measures**: Use sampling, e.g. $\mathscr{R}^f_\theta = \mathbb{E}[f_\theta(\mathbf{x})] \approx R^f = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}, \theta_i) + \varepsilon_N$
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 $\tilde{\varepsilon} = \alpha \cdot 2\sigma_{\text{GP}}(\mathbf{x}) + (1-\alpha)\cdot \varepsilon_N$
4. **Only feasible trial points**, i.e. $\mathscr{R}^c_\omega(\mathbf{x}_{k+1}) \leq 0$, **should be accepted**

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

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

SNOWPAC*

OUU problem statement

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

Features of SNOWPAC:

0. **Extension of NOWPAC**: Deterministic derivative-free nonlinear constraint optimization method using trust-regions
1. **Estimate robustness measures**: Use sampling, e.g. $\mathscr{R}^f_\theta = \mathbb{E}[f_\theta(\mathbf{x})] \approx R^f = \frac{1}{N}\sum_{i=1}^N f(\mathbf{x}, \theta_i) + \varepsilon_N$
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 $\tilde{\varepsilon} = \alpha \cdot 2\sigma_{GP}(\mathbf{x}) + (1-\alpha) \cdot \varepsilon_N$
4. **Only feasible trial points**, i.e. $\mathscr{R}^c_\omega(\mathbf{x}_{k+1}) \leq 0$, **should be accepted**

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

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

Multilevel Monte Carlo estimator: Mean

Mean in OUU:

$$\max_x \mathscr{R}_{\text{Mean}} := \max_x \mathbb{E}[Q(x,\theta)]$$

†Giles, M.B., "Multilevel Monte Carlo methods," Acta Numerica, Vol.24, 2015, p.259–328

Multilevel Monte Carlo estimator: Mean

Mean in OUU:

$$\max_x \mathscr{R}_{\text{Mean}} := \max_x \mathbb{E}[Q(x,\theta)]$$

- Estimator[†]:

$$\mathbb{E}[Q_L] = \mu_0^{\text{ML}}[Q_L] \approx \widehat{\mu_0^{\text{ML}}}[Q_L] = \sum_{\ell=0}^{L} \widehat{\mu_0}[Q^{(\ell)} - Q^{(\ell-1)}] = \sum_{\ell=0}^{L} \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} (Q_i^{(\ell)} - Q_{i,\ell}^{(\ell-1)}), \quad Q_{i,0}^{(-1)} := 0$$

[†]Giles, M.B., "Multilevel Monte Carlo methods," Acta Numerica, Vol.24, 2015, p.259–328

Multilevel Monte Carlo estimator: Mean

Mean in OUU:

$$\max_x \mathscr{R}_{\text{Mean}} := \max_x \mathbb{E}[Q(x,\theta)]$$

- Estimator†:

$$\mathbb{E}[Q_L] = \mu_0^{\text{ML}}[Q_L] \approx \widehat{\mu_0^{\text{ML}}}[Q_L] = \sum_{\ell=0}^{L} \widehat{\mu_0}[Q^{(\ell)} - Q^{(\ell-1)}] = \sum_{\ell=0}^{L} \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} (Q_i^{(\ell)} - Q_{i,\ell}^{(\ell-1)}), \quad Q_{i,0}^{(-1)} := 0$$

- Sample allocation:

$$\min_{N_\ell^{\mathbb{E}}} \sum_{\ell=0}^{L} C_\ell N_\ell^{\mathbb{E}},$$

$$\text{s.t. } \mathbb{V}[\widehat{\mu_0^{\text{ML}}}] = \varepsilon^2, \text{ where } \mathbb{V}[\widehat{\mu_0^{\text{ML}}}] = \sum_{\ell=0}^{L} \mathbb{V}[\widehat{\mu_0^{(\ell)} - \mu_{0,\ell}^{(\ell-1)}}]$$

†Giles, M.B., "Multilevel Monte Carlo methods," Acta Numerica, Vol.24, 2015, p.259–328

Multilevel Monte Carlo estimator: Mean

Mean in OUU:

$$\max_x \mathscr{R}_{\text{Mean}} := \max_x \mathbb{E}[Q(x,\theta)]$$

- Estimator†:

$$\mathbb{E}[Q_L] = \mu_0^{\text{ML}}[Q_L] \approx \widehat{\mu_0^{\text{ML}}}[Q_L] = \sum_{\ell=0}^{L} \widehat{\mu_0}[Q^{(\ell)} - Q^{(\ell-1)}] = \sum_{\ell=0}^{L} \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} (Q_i^{(\ell)} - Q_{i,\ell}^{(\ell-1)}), \quad Q_{i,0}^{(-1)} := 0$$

- Sample allocation:

$$\min_{N_\ell^{\mathbb{E}}} \sum_{\ell=0}^{L} C_\ell N_\ell^{\mathbb{E}},$$

$$\text{s.t. } \mathbb{V}[\widehat{\mu_0^{\text{ML}}}] = \varepsilon^2, \text{ where } \mathbb{V}[\widehat{\mu_0^{\text{ML}}}] = \sum_{\ell=0}^{L} \mathbb{V}[\widehat{\mu_0^{(\ell)} - \mu_{0,\ell}^{(\ell-1)}}]$$

- Solution:

$$N_\ell^{\mathbb{E}} = \left\lceil \lambda \sqrt{\frac{\mathbb{V}[Q_\ell - Q_{\ell-1}]}{C_\ell}} \right\rceil, \text{ where } \lambda = \varepsilon^{-2} \sum_{\ell=0}^{L} \sqrt{\mathbb{V}[Q_\ell - Q_{\ell-1}] C_\ell}$$

†Giles, M.B., "Multilevel Monte Carlo methods," Acta Numerica, Vol.24, 2015, p.259–328

Multilevel Monte Carlo estimator: Standard deviation

Mean in OUU:

$$\max_x \mathscr{R}_{\text{Mean}} := \max_x \mathbb{E}[Q(x,\theta)]$$

Standard deviation in OUU:

$$\max_x \mathscr{R}_{\text{Pback}} := \max_x \mathbb{E}[Q(x,\theta)] - \alpha\sigma[Q(x,\theta)]$$

Multilevel Monte Carlo estimator: Standard deviation

- Estimator:

$$\sigma[Q_L] = \sqrt{\mathbb{V}[Q_L]} \approx \sqrt{\widehat{\mu_2^{\mathrm{ML}}}} := \widehat{\sigma_{\text{biased}}^{\mathrm{ML}}}$$

where

$$\mathbb{V}[Q_L] \approx \widehat{\mu_2^{\mathrm{ML}}}[Q_L] = \sum_{\ell=0}^{L} \widehat{\mu_2}[Q^{(\ell)}] - \widehat{\mu_2}[Q^{(\ell-1)}]$$

$$= \sum_{\ell=0}^{L} \frac{1}{N_\ell - 1} \left(\sum_{i=1}^{N_\ell} (Q_i^{(\ell)} - \widehat{\mu_0^{(\ell)}})^2 - (Q_{i,\ell}^{(\ell-1)} - \widehat{\mu_{0,\ell}^{(\ell-1)}})^2 \right), \quad Q_{i,0}^{(-1)} := 0$$

‡For $\mathbb{V}[\widehat{\mu_2^{\mathrm{ML}}}]$ see FM, GG, DTS, MSE, RNK, HJB, YMM: Higher moment ML estimators for OUU applied to wind plant design, AIAA Scitech 2020 Forum

Multilevel Monte Carlo estimator: Standard deviation

- Estimator:

$$\sigma[Q_L] = \sqrt{\mathbb{V}[Q_L]} \approx \sqrt{\widehat{\mu_2^{\text{ML}}}} := \widehat{\sigma_{\text{biased}}^{\text{ML}}}$$

where

$$\mathbb{V}[Q_L] \approx \widehat{\mu_2^{\text{ML}}}[Q_L] = \sum_{\ell=0}^{L} \widehat{\mu_2}[Q^{(\ell)}] - \widehat{\mu_2}[Q^{(\ell-1)}]$$

$$= \sum_{\ell=0}^{L} \frac{1}{N_\ell - 1} \left(\sum_{i=1}^{N_\ell} (Q_i^{(\ell)} - \widehat{\mu_0^{(\ell)}})^2 - (Q_{i,\ell}^{(\ell-1)} - \widehat{\mu_{0,\ell}^{(\ell-1)}})^2 \right), \quad Q_{i,0}^{(-1)} := 0$$

- Sample allocation‡:

$$\min_{N_\ell^\sigma} \sum_{\ell=0}^{L} C_\ell N_\ell^\sigma,$$

$$\text{s.t. } \mathbb{V}[\widehat{\sigma_{\text{biased}}^{\text{ML}}}] = \varepsilon^2, \text{ where } \mathbb{V}[\widehat{\sigma_{\text{biased}}^{\text{ML}}}] \approx \frac{1}{4} \frac{\mathbb{V}[\widehat{\mu_2^{\text{ML}}}]}{\widehat{\mu_2^{\text{ML}}}}$$

‡For $\mathbb{V}[\widehat{\mu_2^{\text{ML}}}]$ see FM, GG, DTS, MSE, RNK, HJB, YMM: Higher moment ML estimators for OUU applied to wind plant design, AIAA Scitech 2020 Forum

Multilevel Monte Carlo estimator: Standard deviation

- Estimator:

$$\sigma[Q_L] = \sqrt{\mathbb{V}[Q_L]} \approx \sqrt{\widehat{\mu_2^{\text{ML}}}} := \widehat{\sigma_{\text{biased}}^{\text{ML}}}$$

where

$$\begin{aligned}\mathbb{V}[Q_L] \approx \widehat{\mu_2^{\text{ML}}}[Q_L] &= \sum_{\ell=0}^{L} \widehat{\mu_2}[Q^{(\ell)}] - \widehat{\mu_2}[Q^{(\ell-1)}] \\ &= \sum_{\ell=0}^{L} \frac{1}{N_\ell - 1} \left(\sum_{i=1}^{N_\ell} (Q_i^{(\ell)} - \widehat{\mu_0^{(\ell)}})^2 - (Q_{i,\ell}^{(\ell-1)} - \widehat{\mu_{0,\ell}^{(\ell-1)}})^2 \right), \quad Q_{i,0}^{(-1)} := 0\end{aligned}$$

- Sample allocation[‡]:

$$\min_{N_\ell^\sigma} \sum_{\ell=0}^{L} C_\ell N_\ell^\sigma,$$

$$\text{s.t. } \mathbb{V}[\widehat{\sigma_{\text{biased}}^{\text{ML}}}] = \varepsilon^2, \text{ where } \mathbb{V}[\widehat{\sigma_{\text{biased}}^{\text{ML}}}] \approx \frac{1}{4} \frac{\mathbb{V}[\widehat{\mu_2^{\text{ML}}}]}{\widehat{\mu_2^{\text{ML}}}}$$

- Solution: $\Rightarrow$ Numerical Optimization

[‡]For $\mathbb{V}[\widehat{\mu_2^{\text{ML}}}]$ see FM, GG, DTS, MSE, RNK, HJB, YMM: Higher moment ML estimators for OUU applied to wind plant design, AIAA Scitech 2020 Forum

Scientific Computing
Department of Informatics
Technical University of Munich

Multilevel Monte Carlo estimator: Scalarization

Mean in OUU:

$$\max_x \mathscr{R}_{\mathrm{Mean}} := \max_x \mathbb{E}[Q(x,\theta)]$$

Scalarization in OUU:

$$\max_x \mathscr{R}_{\mathrm{Pback}} := \max_x \mathbb{E}[Q(x,\theta)] - \alpha\sigma[Q(x,\theta)]$$

Multilevel Monte Carlo estimator: Scalarization

- Estimator:

$$\mathbb{S}[Q_L] := \mathbb{E}[Q_L] + \alpha\sigma[Q_L] \approx \widehat{\mu_0^{\text{ML}}} + \alpha\widehat{\sigma_{\text{biased}}^{\text{ML}}} := \widehat{\zeta^{\text{ML}}}$$

Scientific Computing
Department of Informatics
Technical University of Munich

Multilevel Monte Carlo estimator: Scalarization

- Estimator:

$$\mathbb{S}[Q_L] := \mathbb{E}[Q_L] + \alpha\sigma[Q_L] \approx \widehat{\mu_0^{\text{ML}}} + \alpha\widehat{\sigma_{\text{biased}}^{\text{ML}}} := \widehat{\zeta^{\text{ML}}}$$

- Sample allocation:

$$\min_{N_\ell^{\mathbb{S}}} \sum_{\ell=0}^{L} C_\ell N_\ell^{\mathbb{S}},$$

$$\text{s.t. } \mathbb{V}[\widehat{\zeta^{\text{ML}}}] = \varepsilon^2, \text{ where } \mathbb{V}[\widehat{\zeta^{\text{ML}}}] \approx \mathbb{V}[\widehat{\mu_0^{\text{ML}}} + \alpha\widehat{\sigma_{\text{biased}}^{\text{ML}}}]$$

Multilevel Monte Carlo estimator: Scalarization

- Estimator:

$$\mathbb{S}[Q_L] := \mathbb{E}[Q_L] + \alpha\sigma[Q_L] \approx \widehat{\mu_0^{\text{ML}}} + \alpha\widehat{\sigma_{\text{biased}}^{\text{ML}}} := \widehat{\zeta^{\text{ML}}}$$

- Sample allocation:

$$\min_{N_\ell^{\mathbb{S}}} \sum_{\ell=0}^{L} C_\ell N_\ell^{\mathbb{S}},$$
$$\text{s.t. } \mathbb{V}[\widehat{\zeta^{\text{ML}}}] = \varepsilon^2, \text{ where } \mathbb{V}[\widehat{\zeta^{\text{ML}}}] \approx \mathbb{V}[\widehat{\mu_0^{\text{ML}}} + \alpha\widehat{\sigma_{\text{biased}}^{\text{ML}}}]$$

- Variance of scalarization:

$$\begin{aligned}
\mathbb{V}[\widehat{\zeta^{\text{ML}}}] &= \mathbb{V}[\widehat{\mu_0^{\text{ML}}} + \alpha\widehat{\sigma_{\text{biased}}^{\text{ML}}}] \\
&= \mathbb{V}[\widehat{\mu_0^{\text{ML}}}] + \alpha^2\mathbb{V}[\widehat{\sigma_{\text{biased}}^{\text{ML}}}] + 2\alpha\text{Cov}[\widehat{\mu_0^{\text{ML}}}, \widehat{\sigma_{\text{biased}}^{\text{ML}}}] \\
&\leq \mathbb{V}[\widehat{\mu_0^{\text{ML}}}] + \alpha^2\mathbb{V}[\widehat{\sigma_{\text{biased}}^{\text{ML}}}] + 2|\alpha|\sqrt{\mathbb{V}[\widehat{\mu_0^{\text{ML}}}] \cdot \mathbb{V}[\widehat{\sigma_{\text{biased}}^{\text{ML}}}]}
\end{aligned}$$

- Solution: $\Rightarrow$ Numerical Optimization

Multilevel MC coupling with SNOWPAC

Generic MLMC error estimators:

- Reminder: (1) $R^f = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x},\theta_i) + \varepsilon_N$ | (2) $\Delta_{k+1} \geq \sqrt{\lambda_t \varepsilon_N}$ | (3) $\tilde{\varepsilon} = \alpha \cdot 2\sigma_{\text{GP}}(\mathbf{x}) + (1-\alpha)\cdot\varepsilon_N$
- Mean $\widehat{\mu_0^{\text{ML}}}$:

$$\mathbb{V}[\widehat{\mu_0^{\text{ML}}}] = (\frac{\varepsilon_N}{2})^2$$

- Standard deviation $\sqrt{\widehat{\mu_2^{\text{ML}}}}^{\S}$:

$$SE(\sigma[Q_L]) \approx \frac{1}{2\sqrt{\widehat{\mu_2^{\text{ML}}}}}\sqrt{\mathbb{V}[\widehat{\mu_2^{\text{ML}}}]} = \frac{\varepsilon_N}{2}$$

- Scalarization $\sqrt{\widehat{\zeta^{\text{ML}}}}$:

$$SE(\mathbb{S}[Q_L]) \leq \sqrt{\mathbb{V}[\widehat{\mu_0^{\text{ML}}}] + \alpha^2\mathbb{V}[\widehat{\sigma_{\text{biased}}^{\text{ML}}}] + 2|\alpha|\sqrt{\mathbb{V}[\widehat{\mu_0^{\text{ML}}}]\cdot\mathbb{V}[\widehat{\sigma_{\text{biased}}^{\text{ML}}}]}} = \frac{\varepsilon_N}{2}$$

§Delta method: $SE[f(\widehat{\theta})] \approx |f'(\widehat{\theta})| SE[\widehat{\theta}], \widehat{\theta} = \widehat{\mu_2}, f(\widehat{\theta}) = \widehat{\theta}^{\frac{1}{2}}$

Scientific Computing
Department of Informatics
Technical University of Munich

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:

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

OUU:

$$\min_x f(x)$$

Mean:

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

Push back:

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:

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

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

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

OUU:

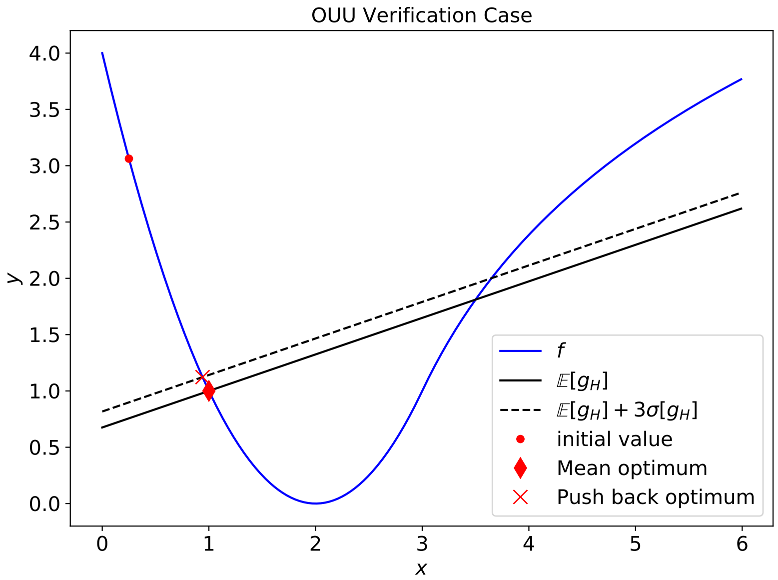
$$\min_x f(x)$$

Mean:

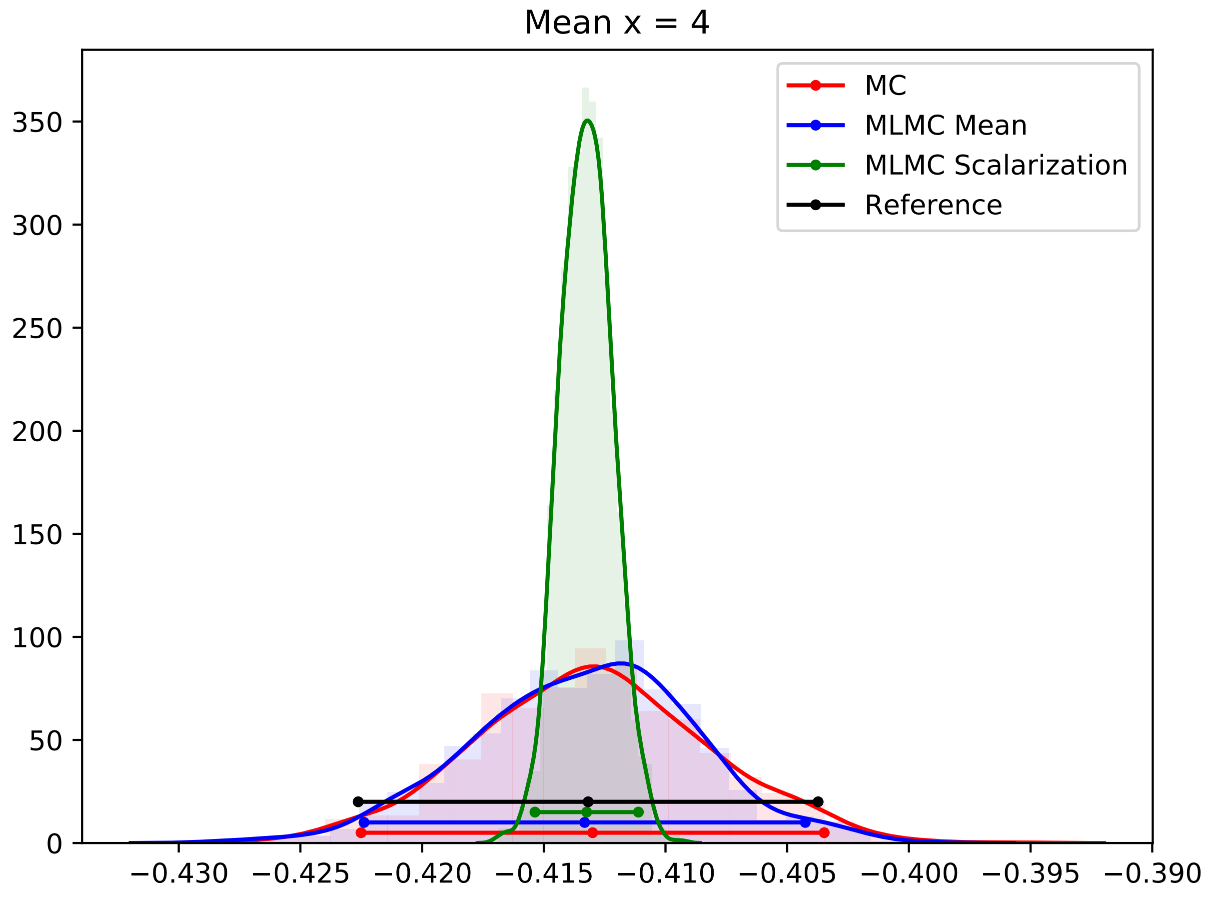
$$\text{s.t. } f(x) \geq \mathbb{E}[g_H(x,\xi)]$$

Push back:

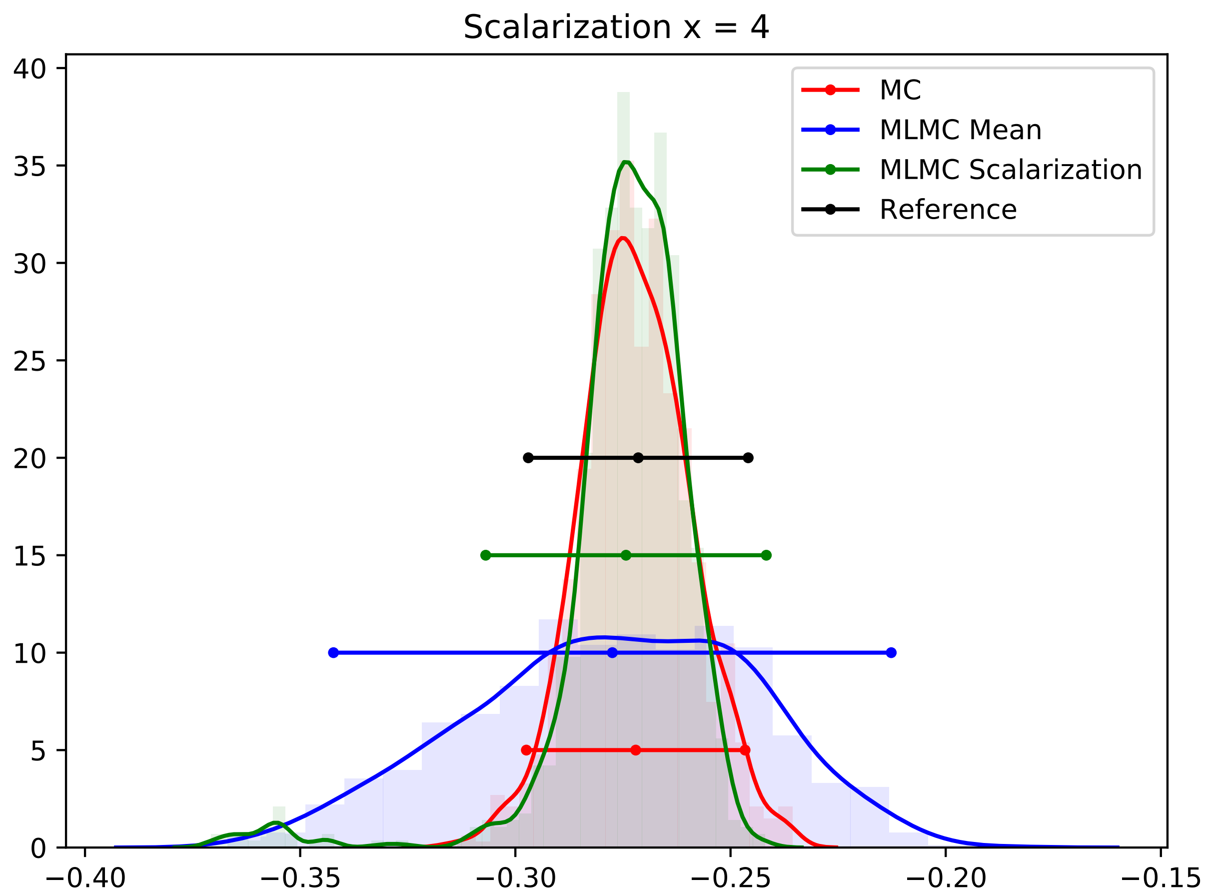
$$\text{s.t. } f(x) \geq \mathbb{E}[g_H(x,\xi)] + 3\sigma[g_H(x,\xi)]$$



Sampling Results



Sampling Results

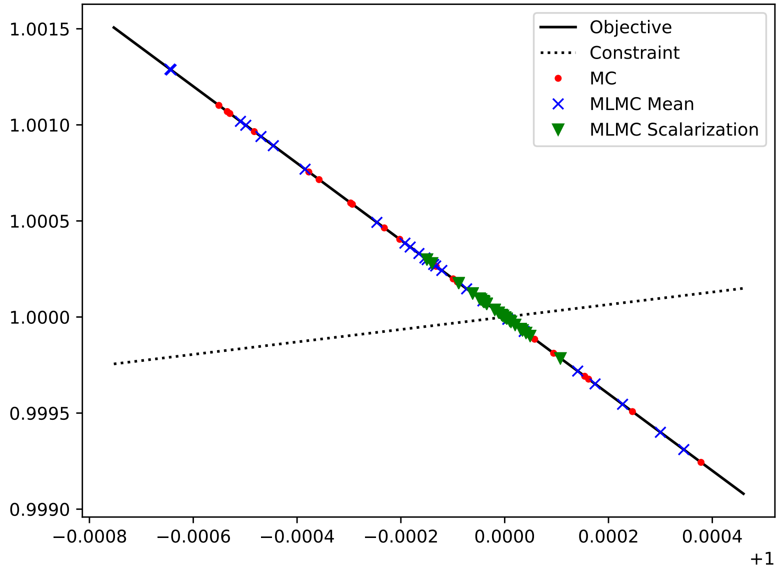


Scientific Computing
Department of Informatics
Technical University of Munich

OUU Results

Mean:

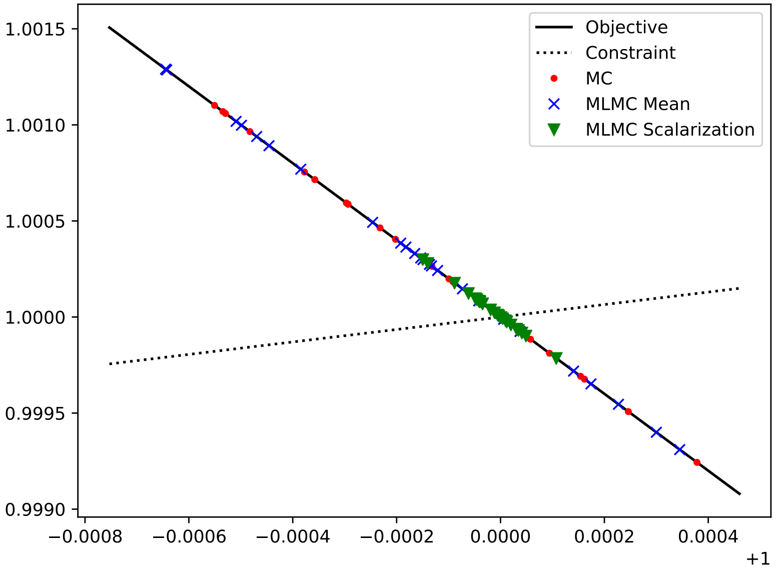
s.t. $f(x) \geq \mathbb{E}[g_H(x,\xi)]$



OUU Results

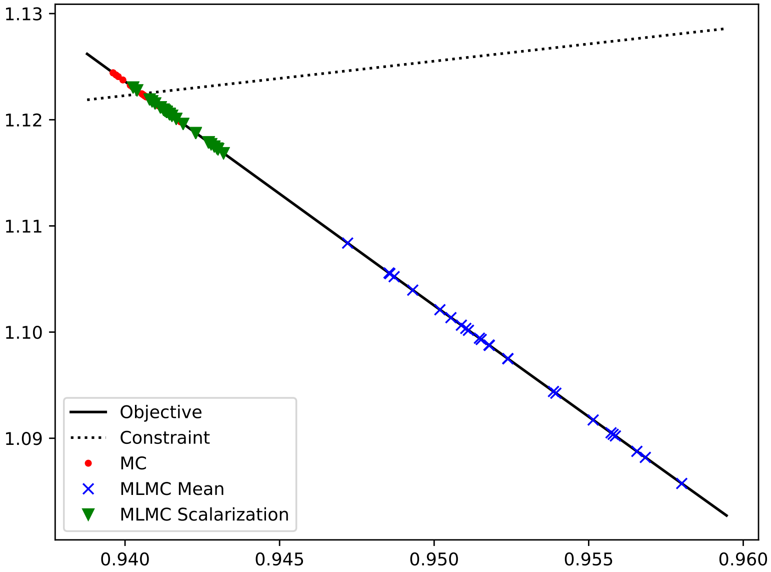
Mean:

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



Push back:

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



Wind Plant Results

Wind plant application

Setup:

- three NREL 5 MW turbines (RD 130 m, HH 110 m) standing 650 m and 1300 m apart
- RANS model with actuator disk turbine representation using WindSE (https://github.com/NREL/WindSE)

Scientific Computing
Department of Informatics
Technical University of Munich

Wind plant application

Setup:

- three NREL 5 MW turbines (RD 130 m, HH 110 m) standing 650 m and 1300 m apart
- RANS model with actuator disk turbine representation using WindSE (https://github.com/NREL/WindSE)

Parameters: (Turbine i = 1, 2, 3)

- Yaw angle design: $\gamma_i \in [-45^\circ, 45^\circ]$
- Yaw angle noise: $\theta_{\gamma_i} \sim \mathscr{L}(0^\circ, 5^\circ)$
- Inflow wind speed: $\theta_u \sim \mathscr{N}(7.5\frac{m}{s}, 1\frac{m}{s})$

Wind plant application

Setup:

- three NREL 5 MW turbines (RD 130 m, HH 110 m) standing 650 m and 1300 m apart
- RANS model with actuator disk turbine representation using WindSE (https://github.com/NREL/WindSE)

OUU Mean:

$$\max_{\gamma_1,\gamma_2,\gamma_3} \mathscr{R}_{\text{Mean}} := \max_{\gamma_1,\gamma_2,\gamma_3} \mathbb{E}[f_{\text{power}}(\gamma_1,\gamma_2,\gamma_3,\theta_u,\theta_{\gamma_1},\theta_{\gamma_2},\theta_{\gamma_3})]$$

OUU Push back:

$$\max_{\gamma_1,\gamma_2,\gamma_3} \mathscr{R}_{\text{Pback}} := \max_{\gamma_1,\gamma_2,\gamma_3} \mathbb{E}[f_{\text{power}}(\cdot)] - 3\sigma[f_{\text{power}}(\cdot)]$$

Parameters: (Turbine i = 1, 2, 3)

- Yaw angle design: $\gamma_i \in [-45^\circ, 45^\circ]$
- Yaw angle noise: $\theta_{\gamma_i} \sim \mathscr{L}(0^\circ, 5^\circ)$
- Inflow wind speed: $\theta_u \sim \mathscr{N}(7.5\frac{m}{s}, 1\frac{m}{s})$

Wind plant application

Setup:

- three NREL 5 MW turbines (RD 130 m, HH 110 m) standing 650 m and 1300 m apart
- RANS model with actuator disk turbine representation using WindSE (https://github.com/NREL/WindSE)

OUU Mean:

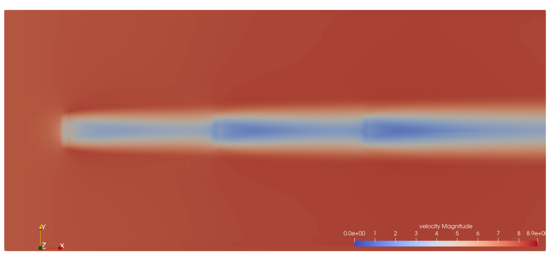
$$\max_{\gamma_1,\gamma_2,\gamma_3} \mathscr{R}_{\text{Mean}} := \max_{\gamma_1,\gamma_2,\gamma_3} \mathbb{E}[f_{\text{power}}(\gamma_1,\gamma_2,\gamma_3,\theta_u,\theta_{\gamma_1},\theta_{\gamma_2},\theta_{\gamma_3})]$$

OUU Push back:

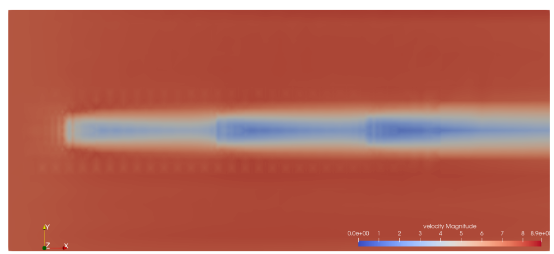
$$\max_{\gamma_1,\gamma_2,\gamma_3} \mathscr{R}_{\text{Pback}} := \max_{\gamma_1,\gamma_2,\gamma_3} \mathbb{E}[f_{\text{power}}(\cdot)] - 3\sigma[f_{\text{power}}(\cdot)]$$

Parameters: (Turbine i = 1, 2, 3)

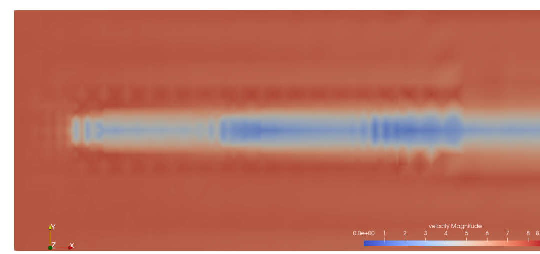
- Yaw angle design: $\gamma_i \in [-45^\circ, 45^\circ]$
- Yaw angle noise: $\theta_{\gamma_i} \sim \mathscr{L}(0^\circ, 5^\circ)$
- Inflow wind speed: $\theta_u \sim \mathscr{N}(7.5\frac{m}{s}, 1\frac{m}{s})$



(a) FINE (DoF 494760)



(b) MEDIUM (DoF 61476)



(c) COARSE (DoF 20760)

Scientific Computing
Department of Informatics
Technical University of Munich

Wind plant application

Setup:

- three NREL 5 MW turbines (RD 130 m, HH 110 m) standing 650 m and 1300 m apart
- RANS model with actuator disk turbine representation using WindSE (https://github.com/NREL/WindSE)

OUU Mean:

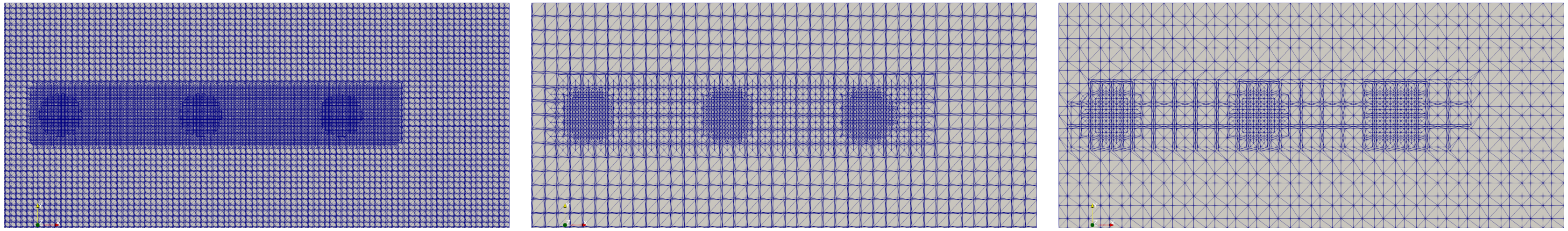
$$\max_{\gamma_1,\gamma_2,\gamma_3} \mathscr{R}_{\text{Mean}} := \max_{\gamma_1,\gamma_2,\gamma_3} \mathbb{E}[f_{\text{power}}\left(\gamma_1,\gamma_2,\gamma_3,\theta_u,\theta_{\gamma_1},\theta_{\gamma_2},\theta_{\gamma_3}\right)]$$

OUU Push back:

$$\max_{\gamma_1,\gamma_2,\gamma_3} \mathscr{R}_{\text{Pback}} := \max_{\gamma_1,\gamma_2,\gamma_3} \mathbb{E}[f_{\text{power}}(\cdot)] - 3\sigma[f_{\text{power}}(\cdot)]$$

Parameters: (Turbine i = 1, 2, 3)

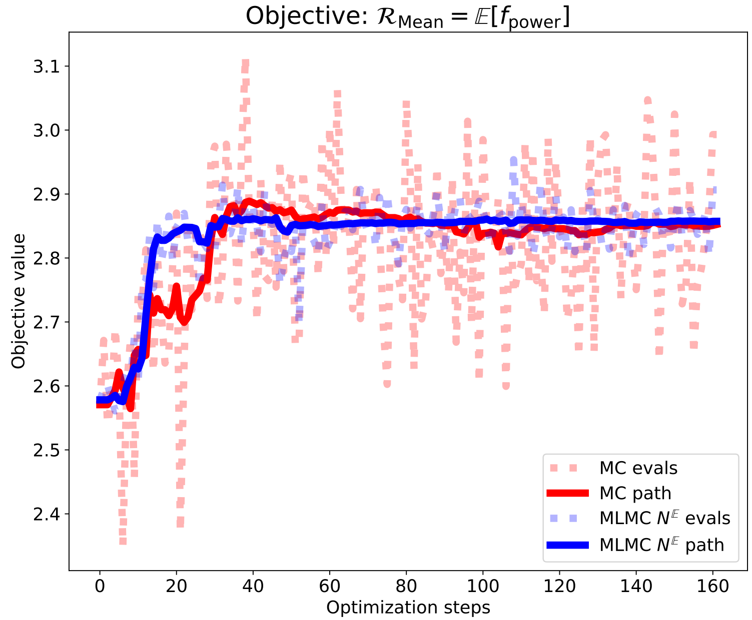
- Yaw angle design: $\gamma_i \in [-45^\circ, 45^\circ]$
- Yaw angle noise: $\theta_{\gamma_i} \sim \mathscr{L}(0^\circ, 5^\circ)$
- Inflow wind speed: $\theta_u \sim \mathscr{N}(7.5\frac{m}{s}, 1\frac{m}{s})$

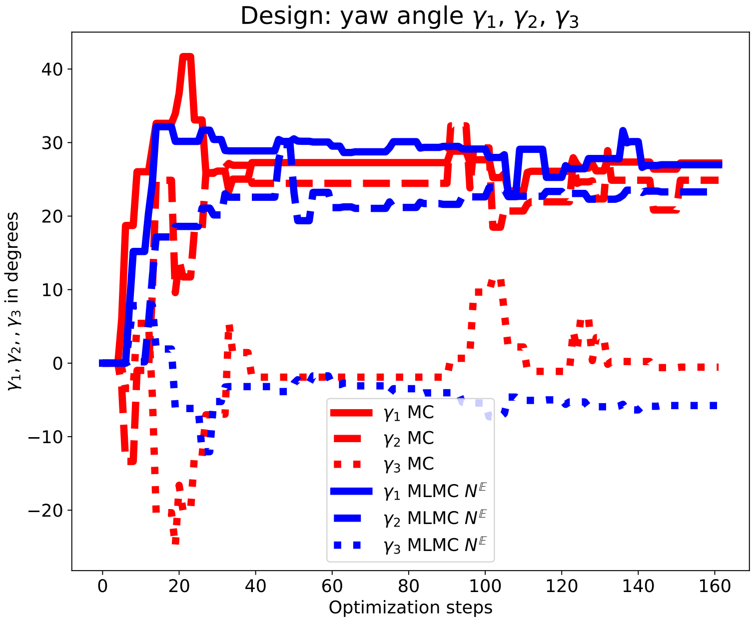


(a) FINE (DoF 494760) (b) MEDIUM (DoF 61476) (c) COARSE (DoF 20760)

Results

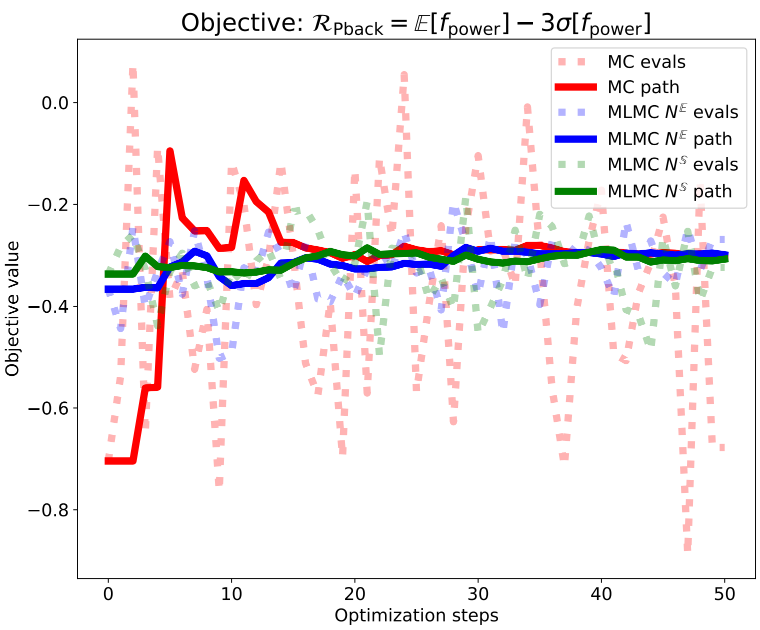
Mean:

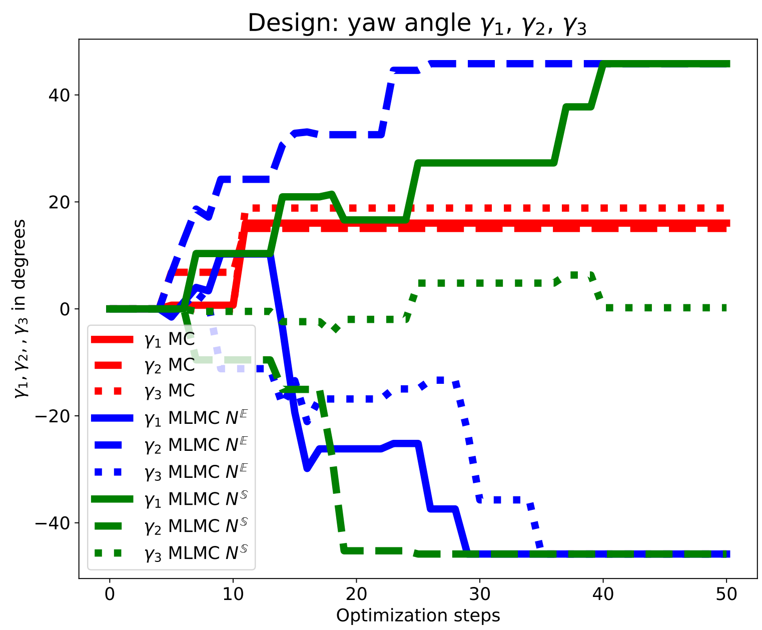




Results

Push back:





Summary:

- **NOWPAC** – Derivative-free trust region methods for constrained nonlinear optimization
- **SNOWPAC** – Stochastic derivative-free optimization using Gaussian process surrogates
- **WindSE** – Python Package for Wind Farm Simulations
- **DAKOTA** – Design Analysis Kit for Optimization and Terascale Applications

$\Rightarrow$ New MLMC estimators for Standard Deviation and Scalarization coupled with **SNOWPAC**.

Future work and open questions:

- Target different wind applications/setups
- From MLMC to MFMC

Links:

- SNOWPAC: github.com/snowpac/snowpac
- Dakota: dakota.sandia.gov
- WindSE: github.com/NREL/WindSE

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

References:

- FM, GG, DTS, MSE, RNK, HJB, YMM, Higher moment multilevel estimators for optimization under uncertainty applied to wind plant design. AIAA 2020
- F. Menhorn, F. Augustin, HJB, YMM, A trust-region method for derivative-free nonlinear constrained stochastic optimization. to be submitted