simula

Risk-Averse PDE-Constrained Optimization: Theory, Algorithms, and Statistical Foundations

Thomas M. Surowiec

Department of Numerical Analysis and Scientific Computing
SURE-AI: Centre for Sustainable, Risk-averse and Ethical AI
Simula Research Laboratory
Oslo, Norway

PGMO Days 2025, November 17, 2025

Figure: Drew P. Kouri, Sandia National Laboratories

D. P. Kouri and T. M. Surowiec.
Risk-averse PDE-constrained optimization using the conditional value-at-risk.
SIAM Journal on Optimization 26, 1 (2016), 365–396.

D. P. Kouri and T. M. Surowiec.
Existence and optimality conditions for risk-averse PDE-constrained optimization
SIAM/ASA J. Uncertainty Quantification 6, 2 (2018), 787–815.

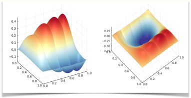
D. P. Kouri and T. M. Surowiec.
Risk-averse optimal control of semilinear elliptic PDEs
ESAIM: Control, Optimisation and Calculus of Variations 26, 53 (2020).

D. P. Kouri and T. M. Surowiec.
Epi-Regularization of Risk Measures
Mathematics of Operations Research 45, 2 (2020), 774–795

D. P. Kouri and T. M. Surowiec.
A primal-dual algorithm for risk minimization
Mathematical Programming 193, 337–363 (2022).

D.P. Kouri, M. Staudigl, and T.M. Surowiec
A Relaxation-based Probabilistic Approach for PDE-constrained Optimization under Uncertainty with Pointwise State Constraints
Comput Optim Appl 85, 441–478 (2023).

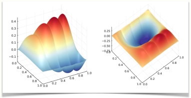
From data to decision-making



Data

$\mathcal{D} = \{((A_i, f_i), u_i)\}$

From data to decision-making

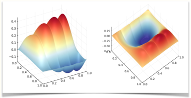


Data
$\mathcal{D} = \{((A_i, f_i), u_i)\}$

$\longrightarrow$

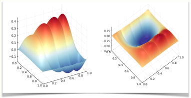
Statistical inversion
to infer measure $\mu \in \mathcal{P}(\Omega)$

From data to decision-making



Data
$\mathcal{D} = \{((A_i, f_i), u_i)\}$
→
Statistical inversion
to infer measure $\mu \in \mathcal{P}(\Omega)$
→
Random PDE
$\nabla \cdot (A(\omega)\, \nabla u) = f(\omega)$

From data to decision-making



Data
$\mathcal{D} = \{((A_i, f_i), u_i)\}$

→

Statistical inversion
to infer measure $\mu \in \mathcal{P}(\Omega)$

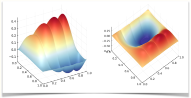
→

Random PDE
$\nabla \cdot (A(\omega)\, \nabla u) = f(\omega)$

→

Risk-Averse Optimization
$\min_{z} \; \mathcal{R}\big(J(u(\cdot), \cdot; z)\big)$

From data to decision-making



Data
$\mathcal{D} = \{((A_i, f_i), u_i)\}$

$\longrightarrow$

Statistical inversion
to infer measure $\mu \in \mathcal{P}(\Omega)$

$\longrightarrow$

Random PDE
$\nabla \cdot (A(\omega)\, \nabla u) = f(\omega)$

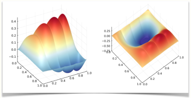
$\longrightarrow$

Risk-Averse Optimization
$$\min_z \; \mathcal{R}\big(J(u(\cdot), \cdot; z)\big)$$

$\longleftarrow$

Risk models
$\mathcal{R} = \mathbb{E}_\mu, \mathrm{CVaR}_\alpha \ldots$

From data to decision-making



Data
$\mathcal{D} = \{((A_i, f_i), u_i)\}$

Statistical inversion
to infer measure $\mu \in \mathcal{P}(\Omega)$

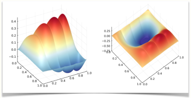
Random PDE
$\nabla \cdot (A(\omega)\, \nabla u) = f(\omega)$

Stoch. Optim.
$z^{k+1} := z^k - \gamma^k \partial \mathcal{J}(z^k, \xi^k)$

Risk-Averse Optimization
$\min_z \ \mathcal{R}\big(J(u(\cdot), \cdot; z)\big)$

Risk models
$\mathcal{R} = \mathbb{E}_\mu, \mathrm{CVaR}_\alpha \ldots$

From data to decision-making



Data
$\mathcal{D} = \{((A_i, f_i), u_i)\}$

Statistical inversion
to infer measure $\mu \in \mathcal{P}(\Omega)$

Random PDE
$\nabla \cdot (A(\omega)\,\nabla u) = f(\omega)$

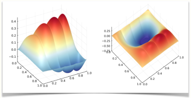
Stoch. Optim.
$z^{k+1} := z^k - \gamma^k \partial \mathcal{J}(z^k, \xi^k)$

HPC
$z^\star = \mathcal{O}_{\|}(J)$

Risk-Averse Optimization
$\min_z \; \mathcal{R}\big(J(u(\cdot), \cdot; z)\big)$

Risk models
$\mathcal{R} = \mathbb{E}_\mu, \mathrm{CVaR}_\alpha \ldots$

From data to decision-making



Data
$\mathcal{D} = \{((A_i, f_i), u_i)\}$

Statistical inversion
to infer measure $\mu \in \mathcal{P}(\Omega)$

Random PDE
$\nabla \cdot (A(\omega)\, \nabla u) = f(\omega)$

Stoch. Optim.
$z^{k+1} := z^k - \gamma^k \partial \mathcal{J}(z^k, \xi^k)$

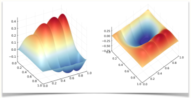
HPC
$z^\star = \mathcal{O}_{\|}(J)$

Asymptotic Statistics
$\frac{1}{N} \sum X_i \to \mathbb{E}[X]$

Risk-Averse Optimization
$\min_z \; \mathcal{R}\big(J(u(\cdot), \cdot; z)\big)$

Risk models
$\mathcal{R} = \mathbb{E}_\mu, \mathrm{CVaR}_\alpha \ldots$

From data to decision-making



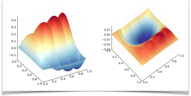
Data
$\mathcal{D} = \{((A_i, f_i), u_i)\}$

Statistical inversion
to infer measure $\mu \in \mathcal{P}(\Omega)$

Random PDE
$\nabla \cdot (A(\omega)\, \nabla u) = f(\omega)$

Risk-Averse Optimization
$$\min_z \; \mathcal{R}\big(J(u(\cdot), \cdot; z)\big)$$

Dealing with random objectives



Using a **random PDE**, means our objective becomes **implicitly random**:

$$\min_{z \in Z_{\mathrm{ad}}} \mathcal{J}(S(z))(\omega) + \rho(z)$$

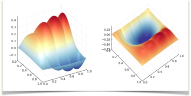
$z \in Z_{\mathrm{ad}}$ decision variables, designs, controls, etc. (**deterministic**)

$z \mapsto S(z)$ solution of the random PDE. (**stochastic**)

$\mathcal{J}$ objective. (either **deterministic** or **stochastic**)

ρ cost or regularization term.

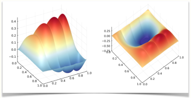
Shaping the Distribution



Consider the typical objective function:

$$X_z(\omega) := \mathcal{J}(S(z))(\omega) + \rho(z).$$

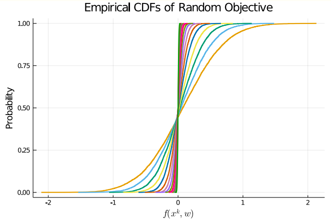
Shaping the Distribution



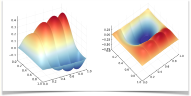
Consider the typical objective function:

$$X_z(\omega) := \mathcal{J}(S(z))(\omega) + \rho(z).$$

Every z yields a difference distribution for X_z (see fig.)



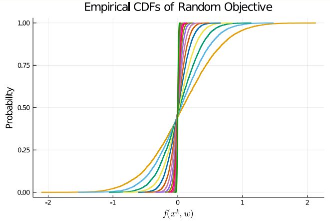
Shaping the Distribution



Consider the typical objective function:

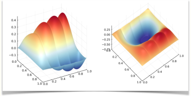
$$X_z(\omega) := \mathcal{J}(S(z))(\omega) + \rho(z).$$

Every z yields a difference distribution for X_z (see fig.)



We need to choose a **numerical surrogate for risk** $\mathcal{R}$.

From data to decision-making



Data
$\mathcal{D} = \{((A_i, f_i), u_i)\}$

→

Statistical inversion
to infer measure $\mu \in \mathcal{P}(\Omega)$

→

Random PDE
$\nabla \cdot (A(\omega)\,\nabla u) = f(\omega)$

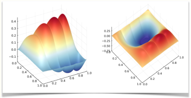
→

Risk-Averse Optimization
$\min_z \; \mathcal{R}\big(J(u(\cdot), \cdot; z)\big)$

←

Risk models
$\mathcal{R} = \mathbb{E}_\mu, \mathrm{CVaR}_\alpha \ldots$

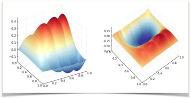
Risk Measures[1]



Risk Neutral $\mathcal{R} = \mathbb{E}$: Optimize to achieve best performance **on average**, ignores outliers, tail events.

[1] See e.g. A. Shapiro, D. Dentcheva, A. Ruszczynski *Lectures on Stochastic Programming: Modeling and Theory.* SIAM, Philadelphia, 2009.

Risk Measures[1]

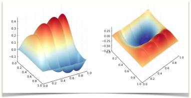


Risk Neutral $\mathcal{R} = \mathbb{E}$: Optimize to achieve best performance **on average**, ignores outliers, tail events.

Mean-Variance $\mathcal{R} = \nu\mathbb{E} + (1-\nu)\mathbb{V}$: Accounts for **risk** via variance $\mathbb{V}$, but $\mathbb{V}$ not monotone.

[1] See e.g. A. Shapiro, D. Dentcheva, A. Ruszczynski *Lectures on Stochastic Programming: Modeling and Theory.* SIAM, Philadelphia, 2009.

Risk Measures[1]



Risk Neutral $\mathcal{R} = \mathbb{E}$: Optimize to achieve best performance **on average**, ignores outliers, tail events.

Mean-Variance $\mathcal{R} = \nu\mathbb{E} + (1-\nu)\mathbb{V}$: Accounts for **risk** via variance $\mathbb{V}$, but $\mathbb{V}$ not monotone.

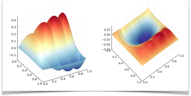
Value-at-Risk $\mathcal{R}[X] = \inf\{\tau : \mathbb{P}(X \leq \tau) \geq \beta\}$:

Find smallest τ such that with **probability** β, $\mathcal{J}(S(z))$ does not exceed the value τ.

Conceptually very useful, but not subadditive, mathematically difficult to handle numerically.

[1] See e.g. A. Shapiro, D. Dentcheva, A. Ruszczynski *Lectures on Stochastic Programming: Modeling and Theory.* SIAM, Philadelphia, 2009.

Risk Measures[1]



Risk Neutral $\mathcal{R} = \mathbb{E}$: Optimize to achieve best performance **on average**, ignores outliers, tail events.

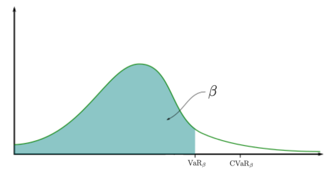
Mean-Variance $\mathcal{R} = \nu\mathbb{E} + (1-\nu)\mathbb{V}$: Accounts for **risk** via variance $\mathbb{V}$, but $\mathbb{V}$ not monotone.

Value-at-Risk $\mathcal{R}[X] = \inf\{\tau : \mathbb{P}(X \leq \tau) \geq \beta\}$:

Find smallest τ such that with **probability** β, $\mathcal{J}(S(z))$ does not exceed the value τ.

Conceptually very useful, but not subadditive, mathematically difficult to handle numerically.

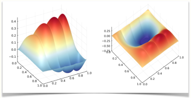
Conditional Value-at-Risk $\mathcal{R}[X] := \frac{1}{1-\beta}\int_\beta^1 \mathrm{VaR}_\alpha[X]\mathrm{d}\alpha \quad \beta \in (0,1)$:



Many names: Excess Loss, Mean Shortfall, Average VaR, Tail VaR

[1] See e.g. A. Shapiro, D. Dentcheva, A. Ruszczynski *Lectures on Stochastic Programming: Modeling and Theory.* SIAM, Philadelphia, 2009.

Risk Measures[1]



Risk Neutral $\mathcal{R} = \mathbb{E}$: Optimize to achieve best performance **on average**, ignores outliers, tail events.

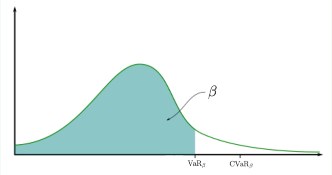
Mean-Variance $\mathcal{R} = \nu\mathbb{E} + (1-\nu)\mathbb{V}$: Accounts for **risk** via variance $\mathbb{V}$, but $\mathbb{V}$ not monotone.

Value-at-Risk $\mathcal{R}[X] = \inf\{\tau : \mathbb{P}(X \leq \tau) \geq \beta\}$:

Find smallest τ such that with **probability** β, $\mathcal{J}(S(z))$ does not exceed the value τ.

Conceptually very useful, but not subadditive, mathematically difficult to handle numerically.

Conditional Value-at-Risk $\mathcal{R}[X] := \frac{1}{1-\beta}\int_\beta^1 \mathrm{VaR}_\alpha[X]\mathrm{d}\alpha \quad \beta \in (0,1)$:

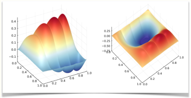


Many names: Excess Loss, Mean Shortfall, Average VaR, Tail VaR

Positively homogeneous, subadditive, monotone, translation equivariant: $\mathrm{CVaR}_\beta[X+c] = \mathrm{CVaR}_\beta[X] + c$ for any $c \in \mathbb{R}$.

[1] See e.g. A. Shapiro, D. Dentcheva, A. Ruszczynski *Lectures on Stochastic Programming: Modeling and Theory.* SIAM, Philadelphia, 2009.

Risk Measures[1]



Risk Neutral $\mathcal{R} = \mathbb{E}$: Optimize to achieve best performance **on average**, ignores outliers, tail events.

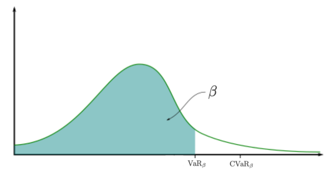
Mean-Variance $\mathcal{R} = \nu\mathbb{E} + (1-\nu)\mathbb{V}$: Accounts for **risk** via variance $\mathbb{V}$, but $\mathbb{V}$ not monotone.

Value-at-Risk $\mathcal{R}[X] = \inf\{\tau : \mathbb{P}(X \leq \tau) \geq \beta\}$:

Find smallest τ such that with **probability** β, $\mathcal{J}(S(z))$ does not exceed the value τ.

Conceptually very useful, but not subadditive, mathematically difficult to handle numerically.

Conditional Value-at-Risk $\mathcal{R}[X] := \frac{1}{1-\beta}\int_\beta^1 \mathrm{VaR}_\alpha[X]\mathrm{d}\alpha \quad \beta \in (0,1)$:



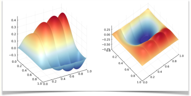
Many names: Excess Loss, Mean Shortfall, Average VaR, Tail VaR

Positively homogeneous, subadditive, monotone, translation equivariant: $\mathrm{CVaR}_\beta[X+c] = \mathrm{CVaR}_\beta[X] + c$ for any $c \in \mathbb{R}$.

Belongs to the important class of **coherent risk measures** (Arztner, Delbaen, Eber, Heath 1999)

[1] See e.g. A. Shapiro, D. Dentcheva, A. Ruszczynski *Lectures on Stochastic Programming: Modeling and Theory.* SIAM, Philadelphia, 2009.

Risk Measures[1]



Risk Neutral $\mathcal{R} = \mathbb{E}$: Optimize to achieve best performance **on average**, ignores outliers, tail events.

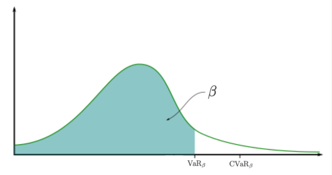
Mean-Variance $\mathcal{R} = \nu\mathbb{E} + (1-\nu)\mathbb{V}$: Accounts for **risk** via variance $\mathbb{V}$, but $\mathbb{V}$ not monotone.

Value-at-Risk $\mathcal{R}[X] = \inf\{\tau : \mathbb{P}(X \leq \tau) \geq \beta\}$:

Find smallest τ such that with **probability** β, $\mathcal{J}(S(z))$ does not exceed the value τ.

Conceptually very useful, but not subadditive, mathematically difficult to handle numerically.

Conditional Value-at-Risk $\mathcal{R}[X] := \frac{1}{1-\beta}\int_\beta^1 \mathrm{VaR}_\alpha[X]\mathrm{d}\alpha \quad \beta \in (0,1)$:



Many names: Excess Loss, Mean Shortfall, Average VaR, Tail VaR

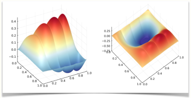
Positively homogeneous, subadditive, monotone, translation equivariant: $\mathrm{CVaR}_\beta[X+c] = \mathrm{CVaR}_\beta[X] + c$ for any $c \in \mathbb{R}$.

Belongs to the important class of **coherent risk measures** (Arztner, Delbaen, Eber, Heath 1999)

$$\mathrm{CVaR}_\beta[X] = \inf_{t\in\mathbb{R}}\left\{t + \frac{1}{1-\beta}\mathbb{E}[(X-t)_+]\right\} \quad \text{(Rockafellar, Uryasev 2000)}$$

[1] See e.g. A. Shapiro, D. Dentcheva, A. Ruszczynski *Lectures on Stochastic Programming: Modeling and Theory.* SIAM, Philadelphia, 2009.

A Contaminant Mitigation Problem[2]

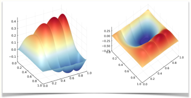


$$
\begin{aligned}
-\nabla \cdot (\epsilon \nabla u) + \mathbb{V} \cdot \nabla u &= f && \text{in } D \\
u &= 0 && \text{on } \Gamma_d = \{0\} \times (0,1) \\
\epsilon \nabla u \cdot n &= 0 && \text{on } \partial D \setminus \Gamma_d
\end{aligned}
$$

$D = (0,1)^2$ physical domain, u is the advected pollutant.

[2] D.P. Kouri, T. M. Surowiec, (2018). SIAM/ASA J. Uncertain. Quantif., 6(2), 787-815.

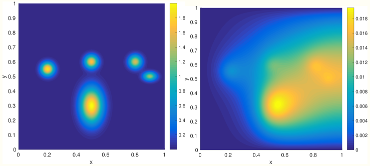
A Contaminant Mitigation Problem[2]



$$
\begin{aligned}
-\nabla \cdot (\epsilon(\omega)\nabla u) + \mathbb{V}(\omega) \cdot \nabla u &= f(\omega) && \text{in } D, \text{ a.s.} \\
u &= 0 && \text{on } \Gamma_d = \{0\} \times (0,1), \text{ a.s.} \\
\epsilon(\omega)\nabla u \cdot n &= 0 && \text{on } \partial D \setminus \Gamma_d, \text{ a.s.}
\end{aligned}
$$

$D = (0,1)^2$ physical domain, u is the advected pollutant.

Random inputs: $\epsilon, \mathbb{V}, f$ permeability, wind, sources of contaminant defined over probability space $(\Omega, \mathcal{F}, \mathbb{P})$.

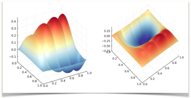


(top left) mean of f

(top right) u with mean values for ϵ, V, f, $z = 0$

[2] D.P. Kouri, T. M. Surowiec, (2018). SIAM/ASA J. Uncertain. Quantif., 6(2), 787-815.

A Contaminant Mitigation Problem[2]

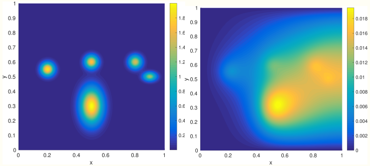


$$
\begin{aligned}
-\nabla \cdot (\epsilon(\omega)\nabla u) + \mathbb{V}(\omega)\cdot\nabla u &= f(\omega) - Bz && \text{in } D, \text{ a.s.}\\
u &= 0 && \text{on } \Gamma_d = \{0\}\times(0,1), \text{ a.s.}\\
\epsilon(\omega)\nabla u \cdot n &= 0 && \text{on } \partial D \setminus \Gamma_d, \text{ a.s.}
\end{aligned}
$$

$D = (0,1)^2$ physical domain, u is the advected pollutant.

Random inputs: $\epsilon, \mathbb{V}, f$ permeability, wind, sources of contaminant defined over probability space $(\Omega, \mathcal{F}, \mathbb{P})$.

$\mathcal{Z}$ is the control space, e.g., $L^2(D)$ or $\mathbb{R}^n$; $\mathcal{Z}_{\text{ad}} = \{z \in \mathcal{Z} \,|\, 0 \le z \le 1\}$.

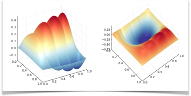


(top left) mean of f

(top right) u with mean values for ϵ, V, f, $z = 0$

[2] D.P. Kouri, T. M. Surowiec, (2018). SIAM/ASA J. Uncertain. Quantif., 6(2), 787-815.

A Contaminant Mitigation Problem[2]



Find optimal placement of mitigating factors z by solving:

$$\min_{z \in \mathcal{Z}_{\text{ad}}} \left\{ \mathcal{R}\left[\frac{1}{2}\int_D S(z)^2 \,\mathrm{d}x\right] + \|z\|_1 \right\}$$

where $S(z) = u : \Omega \to H^1(D)$ solves the weak form of

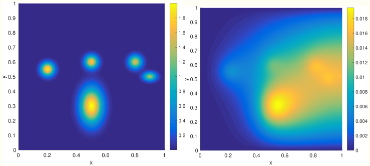
$$\begin{aligned} -\nabla \cdot (\epsilon(\omega)\nabla u) + \mathbb{V}(\omega) \cdot \nabla u &= f(\omega) - Bz && \text{in } D, \text{ a.s.} \\ u &= 0 && \text{on } \Gamma_d = \{0\} \times (0,1), \text{ a.s.} \\ \epsilon(\omega)\nabla u \cdot n &= 0 && \text{on } \partial D \setminus \Gamma_d, \text{ a.s.} \end{aligned}$$

$D = (0,1)^2$ physical domain, u is the advected pollutant.

Random inputs: $\epsilon, \mathbb{V}, f$ permeability, wind, sources of contaminant defined over probability space $(\Omega, \mathcal{F}, \mathbb{P})$.

$\mathcal{Z}$ is the control space, e.g., $L^2(D)$ or $\mathbb{R}^n$; $\mathcal{Z}_{\text{ad}} = \{z \in \mathcal{Z} \,|\, 0 \le z \le 1\}$.

$\mathcal{R} : \mathcal{X} \to \overline{\mathbb{R}}$ is a numerical surrogate for "risk", i.e., a risk measure.

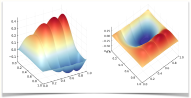


(top left) mean of f

(top right) u with mean values for ϵ, V, f, $z = 0$

[2] D.P. Kouri, T. M. Surowiec, (2018). SIAM/ASA J. Uncertain. Quantif., 6(2), 787-815.

A Contaminant Mitigation Problem[2]



Find optimal placement of mitigating factors z by solving:

$$\min_{z \in \mathcal{Z}_{\text{ad}}} \left\{ \mathbb{E}\left[\frac{1}{2} \int_D S(z)^2 \, \mathrm{d}x \right] + \|z\|_1 \right\}$$

where $S(z) = u : \Omega \to H^1(D)$ solves the weak form of

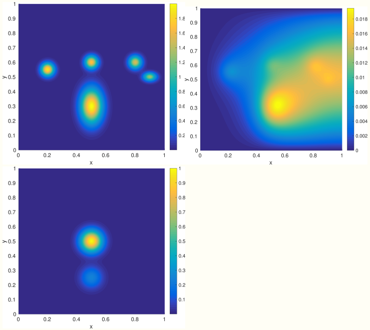
$$\begin{aligned} -\nabla \cdot (\epsilon(\omega)\nabla u) + \mathbb{V}(\omega) \cdot \nabla u &= f(\omega) - Bz && \text{in } D, \text{ a.s.} \\ u &= 0 && \text{on } \Gamma_d = \{0\} \times (0,1), \text{ a.s.} \\ \epsilon(\omega)\nabla u \cdot n &= 0 && \text{on } \partial D \setminus \Gamma_d, \text{ a.s.} \end{aligned}$$

$D = (0,1)^2$ physical domain, u is the advected pollutant.

Random inputs: $\epsilon, \mathbb{V}, f$ permeability, wind, sources of contaminant defined over probability space $(\Omega, \mathcal{F}, \mathbb{P})$.

$\mathcal{Z}$ is the control space, e.g., $L^2(D)$ or $\mathbb{R}^n$; $\mathcal{Z}_{\text{ad}} = \{z \in \mathcal{Z} \,|\, 0 \le z \le 1\}$.

$\mathcal{R} : \mathcal{X} \to \overline{\mathbb{R}}$ is a numerical surrogate for "risk", i.e., a risk measure.



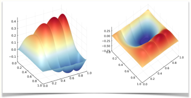
(top left) mean of f

(top right) u with mean values for ϵ, V, f, $z = 0$

(bottom left) optimal solution with $\mathcal{R} = \mathbb{E}$

[2] D.P. Kouri, T. M. Surowiec, (2018). SIAM/ASA J. Uncertain. Quantif., 6(2), 787-815.

A Contaminant Mitigation Problem[2]



Find optimal placement of mitigating factors z by solving:

$$\min_{z \in \mathcal{Z}_{\text{ad}}} \left\{ \mathrm{CVaR}_\beta \left[\frac{1}{2} \int_D S(z)^2 \, \mathrm{d}x \right] + \|z\|_1 \right\}$$

where $S(z) = u : \Omega \to H^1(D)$ solves the weak form of

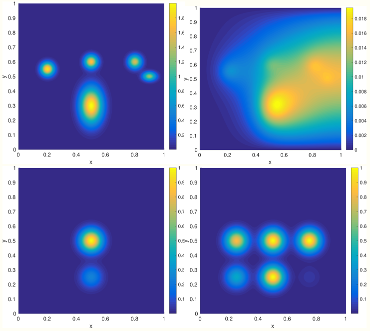
$$\begin{aligned} -\nabla \cdot (\epsilon(\omega) \nabla u) + \mathbb{V}(\omega) \cdot \nabla u &= f(\omega) - Bz && \text{in } D, \text{ a.s.} \\ u &= 0 && \text{on } \Gamma_d = \{0\} \times (0,1), \text{ a.s.} \\ \epsilon(\omega) \nabla u \cdot n &= 0 && \text{on } \partial D \setminus \Gamma_d, \text{ a.s.} \end{aligned}$$

$D = (0,1)^2$ physical domain, u is the advected pollutant.

Random inputs: $\epsilon, \mathbb{V}, f$ permeability, wind, sources of contaminant defined over probability space $(\Omega, \mathcal{F}, \mathbb{P})$.

$\mathcal{Z}$ is the control space, e.g., $L^2(D)$ or $\mathbb{R}^n$; $\mathcal{Z}_{\text{ad}} = \{z \in \mathcal{Z} \,|\, 0 \le z \le 1\}$.

$\mathcal{R} : \mathcal{X} \to \overline{\mathbb{R}}$ is a numerical surrogate for "risk", i.e., a risk measure.



(top left) mean of f

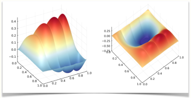
(top right) u with mean values for ϵ, V, f, $z = 0$

(bottom left) optimal solution with $\mathcal{R} = \mathbb{E}$

(bottom right) with $\mathcal{R} = \mathrm{CVaR}\beta$

[2] D.P. Kouri, T. M. Surowiec, (2018). SIAM/ASA J. Uncertain. Quantif., 6(2), 787-815.

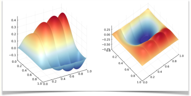
Thermal Compliance Topology Optim.[3]



$$
\begin{aligned}
-\nabla \cdot (\mathbf{E} : \epsilon u) &= F && \text{in } D \\
\epsilon u &= \tfrac{1}{2}(\nabla u + \nabla u^\top) && \text{in } D \\
u &= g && \text{on } \partial D
\end{aligned}
$$

[3] R. Bollapragada, C. Karamanli, B. Keith, B. Lazarov, S. Petrides, & J. Wang (2023). Comput. Math. Appl., 149, 239–258.

Thermal Compliance Topology Optim.[3]



$$-\nabla \cdot (\mathbf{E}(\omega) : \epsilon u) = F(\omega) \qquad \text{in } D, \text{a.s.}$$

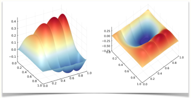
$$\epsilon u = \frac{1}{2}(\nabla u + \nabla u^\top) \qquad \text{in } D, \text{a.s.}$$

$$u = g(\omega) \qquad \text{on } \partial D, \text{a.s.}$$

Random inputs: Linear elastic isotropic material with uncertain Lamé coefficients $\mathbf{E}$ traction forces g bulk forces F.

[3] R. Bollapragada, C. Karamanli, B. Keith, B. Lazarov, S. Petrides, & J. Wang (2023). Comput. Math. Appl., 149, 239–258.

Thermal Compliance Topology Optim.[3]



$$-\nabla \cdot (\mathbf{E}(\omega)(z)) : \epsilon u) = F(\omega) \qquad \text{in } D, \text{a.s.}$$

$$\epsilon u = \frac{1}{2}(\nabla u + \nabla u^\top) \qquad \text{in } D, \text{a.s.}$$

$$u = g(\omega) \qquad \text{on } \partial D, \text{a.s.}$$

Random inputs: Linear elastic isotropic material with uncertain Lamé coefficients $\mathbf{E}$ traction forces g bulk forces F.

The material density $z \in \mathcal{Z}_{\text{ad}}$ fulfills

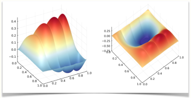
$z : D \to \mathbb{R}$.

$z(x) \in [0, 1]$ a.e. on D ($z = 0$ "no material", $z = 1$ "material").

$\int_D z \, \mathrm{d}x \leq V_0|D|$ (volume fraction).

[3] R. Bollapragada, C. Karamanli, B. Keith, B. Lazarov, S. Petrides, & J. Wang (2023). Comput. Math. Appl., 149, 239–258.

Thermal Compliance Topology Optim.[3]



Find optimal material distribution $z^\star$ that minimizes compliance:

$$\min_{z \in \mathcal{Z}_{\text{ad}}} \mathcal{R}\left[\int_D F(\cdot) \cdot S(z)\,\mathrm{d}x\right] + \wp(z)$$

where $S(z) = u$ solves

$$\begin{aligned} -\nabla \cdot (\mathbf{E}(\omega)(z)) : \epsilon u) &= F(\omega) && \text{in } D, \text{a.s.} \\ \epsilon u &= \tfrac{1}{2}(\nabla u + \nabla u^\top) && \text{in } D, \text{a.s.} \\ u &= g(\omega) && \text{on } \partial D, \text{a.s.} \end{aligned}$$

Random inputs: Linear elastic isotropic material with uncertain Lamé coefficients $\mathbf{E}$ traction forces g bulk forces F.

The material density $z \in \mathcal{Z}_{\text{ad}}$ fulfills

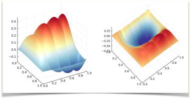
$z : D \to \mathbb{R}$.

$z(x) \in [0, 1]$ a.e. on D ($z = 0$ "no material", $z = 1$ "material").

$\int_D z\,\mathrm{d}x \leq V_0|D|$ (volume fraction).

[3] R. Bollapragada, C. Karamanli, B. Keith, B. Lazarov, S. Petrides, & J. Wang (2023). Comput. Math. Appl., 149, 239–258.

Thermal Compliance Topology Optim.[3]



Find optimal material distribution $z^\star$ that minimizes compliance:

$$\min_{z\in\mathcal{Z}_{\text{ad}}} \mathbb{E}\left[\int_D F(\cdot)\cdot S(z)\,\mathrm{d}x\right] + \wp(z)$$

where $S(z) = u$ solves

$$\begin{aligned} -\nabla\cdot(\mathbf{E}(\omega)(z)) : \epsilon u) &= F(\omega) && \text{in } D, \text{a.s.} \\ \epsilon u &= \tfrac{1}{2}(\nabla u + \nabla u^\top) && \text{in } D, \text{a.s.} \\ u &= g(\omega) && \text{on } \partial D, \text{a.s.} \end{aligned}$$

Random inputs: Linear elastic isotropic material with uncertain Lamé coefficients $\mathbf{E}$ traction forces g bulk forces F.

The material density $z \in \mathcal{Z}_{\text{ad}}$ fulfills

$z : D \to \mathbb{R}$.

$z(x) \in [0,1]$ a.e. on D ($z = 0$ "no material", $z = 1$ "material").

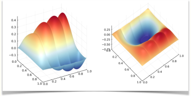
$\int_D z\,\mathrm{d}x \leq V_0|D|$ (volume fraction).

(top) Optimal density field ignoring random inputs

(bottom) optimal density field using $\mathcal{R} = \mathbb{E}$.

[3] R. Bollapragada, C. Karamanli, B. Keith, B. Lazarov, S. Petrides, & J. Wang (2023). Comput. Math. Appl., 149, 239–258.

From data to decision-making



Data
$\mathcal{D} = \{((A_i, f_i), u_i)\}$

Statistical inversion
to infer measure $\mu \in \mathcal{P}(\Omega)$

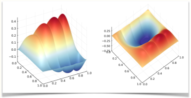
Random PDE
$\nabla \cdot (A(\omega)\, \nabla u) = f(\omega)$

Stoch. Optim.
$z^{k+1} := z^k - \gamma^k \partial \mathcal{J}(z^k, \xi^k)$

Risk-Averse Optimization
$\min_z \; \mathcal{R}\big(J(u(\cdot), \cdot; z)\big)$

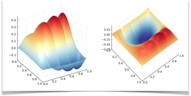
Risk models
$\mathcal{R} = \mathbb{E}_\mu, \mathrm{CVaR}_\alpha \ldots$

A Class of Risk Measures



Using $\Phi(X) = \mathbb{E}[\max\{0, X\}] = \mathbb{E}[(X)_+]$, we can define several useful risk measures:

A Class of Risk Measures

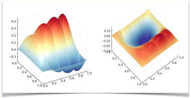


Using $\Phi(X) = \mathbb{E}[\max\{0, X\}] = \mathbb{E}[(X)_+]$, we can define several useful risk measures:

Convex combination of the expected value and CVaR

$$\mathcal{R}(X) = (1-t)\mathbb{E}[X] + t \inf_{a \in \mathbb{R}} \left\{ a + \tfrac{1}{1-\beta}\Phi(X-a) \right\}, \;\; \beta \in (0,1), \;\; t \in (0,1],$$

A Class of Risk Measures



Using $\Phi(X) = \mathbb{E}[\max\{0, X\}] = \mathbb{E}[(X)_+]$, we can define several useful risk measures:

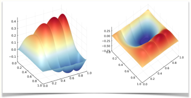
Convex combination of the expected value and CVaR

$$\mathcal{R}(X) = (1-t)\mathbb{E}[X] + t \inf_{a \in \mathbb{R}} \left\{ a + \tfrac{1}{1-\beta} \Phi(X - a) \right\}, \;\; \beta \in (0,1), \;\; t \in (0,1],$$

Mean-plus-semideviation-from-target of order 1

$$\mathcal{R}(X) = \mathbb{E}[X] + c\Phi(X - t), \;\; c > 0, \;\; t \in \mathbb{R},$$

A Class of Risk Measures



Using $\Phi(X) = \mathbb{E}[\max\{0, X\}] = \mathbb{E}[(X)_+]$, we can define several useful risk measures:

Convex combination of the expected value and CVaR

$$\mathcal{R}(X) = (1-t)\mathbb{E}[X] + t \inf_{a \in \mathbb{R}} \left\{ a + \tfrac{1}{1-\beta} \Phi(X - a) \right\}, \;\; \beta \in (0,1), \;\; t \in (0,1],$$

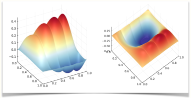
Mean-plus-semideviation-from-target of order 1

$$\mathcal{R}(X) = \mathbb{E}[X] + c\Phi(X - t), \;\; c > 0, \;\; t \in \mathbb{R},$$

Mean-plus-semideviation of order 1

$$\mathcal{R}(X) = \mathbb{E}[X] + c\Phi(X - \mathbb{E}[X]), \quad c > 0,$$

Abstract Formulation



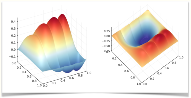
Many problems take the abstract form (with $\mathcal{X} = \mathcal{Z}$ or $\mathcal{Z} \times \mathbb{R}$, $\mathcal{X}_{\text{ad}} \subset \mathcal{X}$):

$$\min_{x \in \mathcal{X}_{\text{ad}}} \{g(x) + \Phi(G(x))\} \tag{P}$$

$G : \mathcal{X} \to \mathcal{Y}$ is a random operator (e.g. $J(S(z)(\omega))$), g is a differentiable function (e.g. $\mathbb{E}[J(S(z))]$), $(\Omega, \mathcal{F}, \mathbb{P})$ complete probability space, $\mathcal{Y} := L^2(\Omega, \mathcal{F}, \mathbb{P})$.

$\Phi : \mathcal{Y} \to \mathbb{R}$ is a functional that maps random variables into $\mathbb{R}$ (e.g. part of a risk measure)

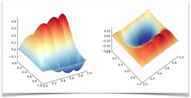
Assumptions and Observations on Φ



Assume $\Phi : \mathcal{Y} \to \mathbb{R}$ convex, positively homogeneous, monotonic wrt partial order on $\mathcal{Y}$

Example: $\Phi(Y) := \mathbb{E}[(Y)_+]$.

Assumptions and Observations on Φ

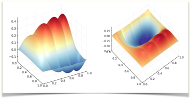


Assume $\Phi : \mathcal{Y} \to \mathbb{R}$ convex, positively homogeneous, monotonic wrt partial order on $\mathcal{Y}$

Example: $\Phi(Y) := \mathbb{E}[(Y)_+]$.

Φ finite and convex on all of $\mathcal{Y}$ $\Rightarrow$ Φ is continuous and subdifferentiable.

Assumptions and Observations on Φ



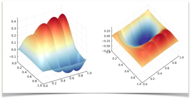
Assume $\Phi : \mathcal{Y} \to \mathbb{R}$ convex, positively homogeneous, monotonic wrt partial order on $\mathcal{Y}$

Example: $\Phi(Y) := \mathbb{E}[(Y)_+]$.

Φ finite and convex on all of $\mathcal{Y} \Rightarrow \Phi$ is continuous and subdifferentiable.

$\Phi(Y) = \Phi^{**}(Y)$ and $\Phi(Y) = \sup_{\lambda \in \partial\Phi(0)} \mathbb{E}[\lambda Y] \quad \forall Y \in \mathcal{Y}$.

Assumptions and Observations on Φ



Assume $\Phi : \mathcal{Y} \to \mathbb{R}$ convex, positively homogeneous, monotonic wrt partial order on $\mathcal{Y}$

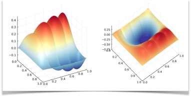
Example: $\Phi(Y) := \mathbb{E}[(Y)_+]$.

Φ finite and convex on all of $\mathcal{Y} \Rightarrow \Phi$ is continuous and subdifferentiable.

$\Phi(Y) = \Phi^{**}(Y)$ and $\Phi(Y) = \sup_{\lambda \in \partial\Phi(0)} \mathbb{E}[\lambda Y] \quad \forall Y \in \mathcal{Y}$.

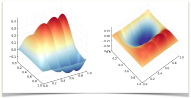
$\partial\Phi(0) := \mathfrak{A} \subseteq \{\lambda \in \mathcal{Y} \,|\, \lambda \geq 0 \text{ a.s.}\}$ is a nonempty, closed, bounded, convex "risk envelope"

Towards a Primal-Dual Algorithm



$$\min_{x\in\mathcal{X}_{\mathsf{ad}}} g(x) + \Phi(G(x)) = \min_{x\in\mathcal{X}_{\mathsf{ad}}} \sup_{\lambda\in\mathfrak{A}} g(x) + \mathbb{E}[\lambda G(x)] \quad \text{set} \quad \ell(x,\lambda) := g(x) + \mathbb{E}[\lambda G(x)].$$

Towards a Primal-Dual Algorithm

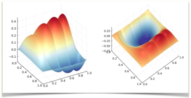


$$\min_{x \in \mathcal{X}_{\text{ad}}} g(x) + \Phi(G(x)) = \min_{x \in \mathcal{X}_{\text{ad}}} \sup_{\lambda \in \mathfrak{A}} g(x) + \mathbb{E}[\lambda G(x)] \quad \text{set} \quad \ell(x, \lambda) := g(x) + \mathbb{E}[\lambda G(x)].$$

The generalized augmented Lagrangian is then

$$L(x, \lambda, r) = \max_{\theta \in \mathfrak{A}} \left\{ \ell(x, \theta) - \frac{1}{2r} \mathbb{E}[(\lambda - \theta)^2] \right\}$$

Towards a Primal-Dual Algorithm



$$\min_{x\in\mathcal{X}_{\text{ad}}} g(x) + \Phi(G(x)) = \min_{x\in\mathcal{X}_{\text{ad}}} \sup_{\lambda\in\mathfrak{A}} g(x) + \mathbb{E}[\lambda G(x)] \quad \text{set} \quad \ell(x,\lambda) := g(x) + \mathbb{E}[\lambda G(x)].$$

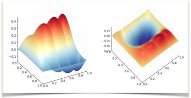
The generalized augmented Lagrangian is then

$$L(x,\lambda,r) = \max_{\theta\in\mathfrak{A}} \left\{ \ell(x,\theta) - \frac{1}{2r}\mathbb{E}[(\lambda-\theta)^2] \right\} = g(x) + \Phi_{r,\lambda}(G(x)).$$

where $\Phi_{r,\lambda}(Y) = \inf_Z\{\Phi(Z) + \Psi_{r,\lambda}(Z-Y)\}$ with $\Psi_{r,\lambda}(Y) = \mathbb{E}[\lambda Y] + \frac{r}{2}\mathbb{E}[Y^2]$.

[a] See D. P. Kouri and T. M. Surowiec, *Epi-Regularization of Risk Measures* Mathematics of Operations Research 45, 2 (2020), 774–795 for more on this technique.

Towards a Primal-Dual Algorithm



$$\min_{x\in\mathcal{X}_{\text{ad}}} g(x)+\Phi(G(x)) = \min_{x\in\mathcal{X}_{\text{ad}}}\sup_{\lambda\in\mathfrak{A}} g(x)+\mathbb{E}[\lambda G(x)] \quad \text{set} \quad \ell(x,\lambda) := g(x)+\mathbb{E}[\lambda G(x)].$$

The generalized augmented Lagrangian is then

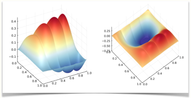
$$L(x,\lambda,r) = \max_{\theta\in\mathfrak{A}}\left\{\ell(x,\theta)-\frac{1}{2r}\mathbb{E}[(\lambda-\theta)^2]\right\} = g(x)+\Phi_{r,\lambda}(G(x)).$$

where $\Phi_{r,\lambda}(Y) = \inf_Z\{\Phi(Z)+\Psi_{r,\lambda}(Z-Y)\}$ with $\Psi_{r,\lambda}(Y) = \mathbb{E}[\lambda Y]+\frac{r}{2}\mathbb{E}[Y^2]$.

Since $\Phi_{r,\lambda}$ is an epi-regularized[a] version of Φ, it is convex, monotonic, $\mathcal{C}^1$ and Γ-converges to $\Phi(\cdot)$.

[a] See D. P. Kouri and T. M. Surowiec, *Epi-Regularization of Risk Measures* Mathematics of Operations Research 45, 2 (2020), 774–795 for more on this technique.

Towards a Primal-Dual Algorithm



$$\min_{x \in \mathcal{X}_{\text{ad}}} g(x) + \Phi(G(x)) = \min_{x \in \mathcal{X}_{\text{ad}}} \sup_{\lambda \in \mathfrak{A}} g(x) + \mathbb{E}[\lambda G(x)] \quad \text{set} \quad \ell(x, \lambda) := g(x) + \mathbb{E}[\lambda G(x)].$$

The generalized augmented Lagrangian is then

$$L(x, \lambda, r) = \max_{\theta \in \mathfrak{A}} \left\{ \ell(x, \theta) - \frac{1}{2r} \mathbb{E}[(\lambda - \theta)^2] \right\} = g(x) + \Phi_{r,\lambda}(G(x)).$$

where $\Phi_{r,\lambda}(Y) = \inf_Z \{\Phi(Z) + \Psi_{r,\lambda}(Z - Y)\}$ with $\Psi_{r,\lambda}(Y) = \mathbb{E}[\lambda Y] + \frac{r}{2}\mathbb{E}[Y^2]$.

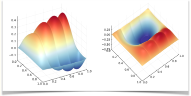
Since $\Phi_{r,\lambda}$ is an epi-regularized[a] version of Φ, it is convex, monotonic, $\mathcal{C}^1$ and Γ-converges to $\Phi(\cdot)$.

Letting $\mathfrak{P}$ be the L^2-projection onto $\mathfrak{A}$, the maximizer above is given by

$$\Lambda(x, \lambda, r) := \mathfrak{P}_{\mathfrak{A}}(rG(x) + \lambda).$$

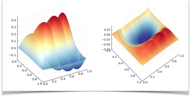
[a] See D. P. Kouri and T. M. Surowiec, *Epi-Regularization of Risk Measures* Mathematics of Operations Research 45, 2 (2020), 774–795 for more on this technique.

Towards a Primal-Dual Algorithm



This viewpoint goes back to the method of multipliers/augmented Lagrangian e.g. Hestenes 1969, Powell 1969, Rockafellar 1976.

Towards a Primal-Dual Algorithm

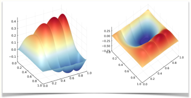


This viewpoint goes back to the method of multipliers/augmented Lagrangian e.g. Hestenes 1969, Powell 1969, Rockafellar 1976.

Solve a sequence of subproblems in x for minimizing $L(x, \lambda, r)$.

Use $\Lambda(x, \lambda, r)$ to update λ.

The PD-Risk Algorithm

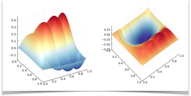


Algorithm The Primal-Dual Risk Minimization Algorithm [4]

Given $x_0 \in \mathcal{X}_{\text{ad}}$, $r_0 \in (0,\infty)$, $\lambda_0 \in \mathfrak{A}$, and stationarity error: $\mathfrak{S}(x,\lambda,r) := \|x - \mathfrak{P}_{\mathcal{X}_{\text{ad}}}(x - \nabla_x L(x,\lambda,r))\|_{\mathcal{X}}$
Parameters $\rho_x \in (0,1)$, $\rho_\lambda \in (0,1)$, $\rho_r \in (1,\infty)$,
Tolerances $0 < \tau_x < \tau_{x,0}$, and $0 < \tau_\lambda < \tau_{\lambda,0}$.
for $k = 0,1,2,\ldots$ **do**

[4] See D.P. Kouri, T.M. Surowiec *A primal-dual algorithm for risk minimization*. Math. Programm. 193, 337–363 (2022). for a full convergence theory in the continuous setting.

The PD-Risk Algorithm



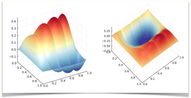
Algorithm The Primal-Dual Risk Minimization Algorithm [4]

Given $x_0 \in \mathcal{X}_{\text{ad}}$, $r_0 \in (0, \infty)$, $\lambda_0 \in \mathfrak{A}$, and stationarity error: $\mathfrak{S}(x, \lambda, r) := \|x - \mathfrak{P}_{\mathcal{X}_{\text{ad}}}(x - \nabla_x L(x, \lambda, r))\|_{\mathcal{X}}$
Parameters $\rho_x \in (0, 1)$, $\rho_\lambda \in (0, 1)$, $\rho_r \in (1, \infty)$,
Tolerances $0 < \tau_x < \tau_{x,0}$, and $0 < \tau_\lambda < \tau_{\lambda,0}$.
for $k = 0, 1, 2, \ldots$ **do**

1. Find $x_{k+1} \in \mathcal{X}_{\text{ad}}$ s.t. $\mathfrak{S}(x_{k+1}, \lambda_k, r_k) \le \tau_{x,k}$ ▷ Approximate primal solution

[4] See D.P. Kouri, T.M. Surowiec *A primal-dual algorithm for risk minimization*. Math. Programm. 193, 337–363 (2022). for a full convergence theory in the continuous setting.

The PD-Risk Algorithm



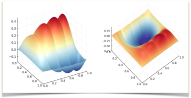
Algorithm The Primal-Dual Risk Minimization Algorithm [4]

Given $x_0 \in \mathcal{X}_{\text{ad}}$, $r_0 \in (0,\infty)$, $\lambda_0 \in \mathfrak{A}$, and stationarity error: $\mathfrak{S}(x,\lambda,r) := \|x - \mathfrak{P}_{\mathcal{X}_{\text{ad}}}(x - \nabla_x L(x,\lambda,r))\|_{\mathcal{X}}$
Parameters $\rho_x \in (0,1)$, $\rho_\lambda \in (0,1)$, $\rho_r \in (1,\infty)$,
Tolerances $0 < \tau_x < \tau_{x,0}$, and $0 < \tau_\lambda < \tau_{\lambda,0}$.
for $k = 0,1,2,\ldots$ **do**

1. Find $x_{k+1} \in \mathcal{X}_{\text{ad}}$ s.t. $\mathfrak{S}(x_{k+1},\lambda_k,r_k) \le \tau_{x,k}$ ▷ Approximate primal solution
2. Set $\lambda_{k+1} = \Lambda(x_{k+1},\lambda_k,r_k)$ ▷ Dual update

[4] See D.P. Kouri, T.M. Surowiec *A primal-dual algorithm for risk minimization*. Math. Programm. 193, 337–363 (2022). for a full convergence theory in the continuous setting.

The PD-Risk Algorithm



Algorithm The Primal-Dual Risk Minimization Algorithm [4]

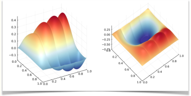
Given $x_0 \in \mathcal{X}_{\text{ad}}$, $r_0 \in (0, \infty)$, $\lambda_0 \in \mathfrak{A}$, and stationarity error: $\mathfrak{S}(x, \lambda, r) := \|x - \mathfrak{P}_{\mathcal{X}_{\text{ad}}}(x - \nabla_x L(x, \lambda, r))\|_{\mathcal{X}}$
Parameters $\rho_x \in (0, 1)$, $\rho_\lambda \in (0, 1)$, $\rho_r \in (1, \infty)$,
Tolerances $0 < \tau_x < \tau_{x,0}$, and $0 < \tau_\lambda < \tau_{\lambda,0}$.
for $k = 0, 1, 2, \dots$ **do**

1. Find $x_{k+1} \in \mathcal{X}_{\text{ad}}$ s.t. $\mathfrak{S}(x_{k+1}, \lambda_k, r_k) \leq \tau_{x,k}$ ▷ Approximate primal solution
2. Set $\lambda_{k+1} = \Lambda(x_{k+1}, \lambda_k, r_k)$ ▷ Dual update
3. **if** $\mathfrak{S}(x_{k+1}, \lambda_k, r_k) \leq \tau_x$ *and* $\|\lambda_k - \lambda_{k+1}\|_{\mathcal{Y}} \leq \tau_\lambda$ **then**
   return x_{k+1}

▷ Check for primal and dual convergence

[4] See D.P. Kouri, T.M. Surowiec *A primal-dual algorithm for risk minimization*. Math. Programm. 193, 337–363 (2022). for a full convergence theory in the continuous setting.

The PD-Risk Algorithm



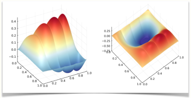
Algorithm The Primal-Dual Risk Minimization Algorithm [4]

Given $x_0 \in \mathcal{X}_{\text{ad}}$, $r_0 \in (0,\infty)$, $\lambda_0 \in \mathfrak{A}$, and stationarity error: $\mathfrak{S}(x,\lambda,r) := \|x - \mathfrak{P}_{\mathcal{X}_{\text{ad}}}(x - \nabla_x L(x,\lambda,r))\|_{\mathcal{X}}$
Parameters $\rho_x \in (0,1)$, $\rho_\lambda \in (0,1)$, $\rho_r \in (1,\infty)$,
Tolerances $0 < \tau_x < \tau_{x,0}$, and $0 < \tau_\lambda < \tau_{\lambda,0}$.
for $k = 0,1,2,\ldots$ **do**

1. Find $x_{k+1} \in \mathcal{X}_{\text{ad}}$ s.t. $\mathfrak{S}(x_{k+1},\lambda_k,r_k) \le \tau_{x,k}$ ▷ Approximate primal solution
2. Set $\lambda_{k+1} = \Lambda(x_{k+1},\lambda_k,r_k)$ ▷ Dual update
3. **if** $\mathfrak{S}(x_{k+1},\lambda_k,r_k) \le \tau_x$ *and* $\|\lambda_k - \lambda_{k+1}\|_{\mathcal{Y}} \le \tau_\lambda$ **then**
   return x_{k+1}
   ▷ Check for primal and dual convergence
4. **if** $\|\lambda_k - \lambda_{k+1}\|_{\mathcal{Y}} > \tau_{\lambda,k}$ **then**
   $r_{k+1} = \rho_r r_k$
   ▷ Increase penalty r_k if dual variable changes significantly

[4] See D.P. Kouri, T.M. Surowiec *A primal-dual algorithm for risk minimization*. Math. Programm. 193, 337–363 (2022). for a full convergence theory in the continuous setting.

The PD-Risk Algorithm



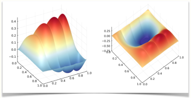
Algorithm The Primal-Dual Risk Minimization Algorithm [4]

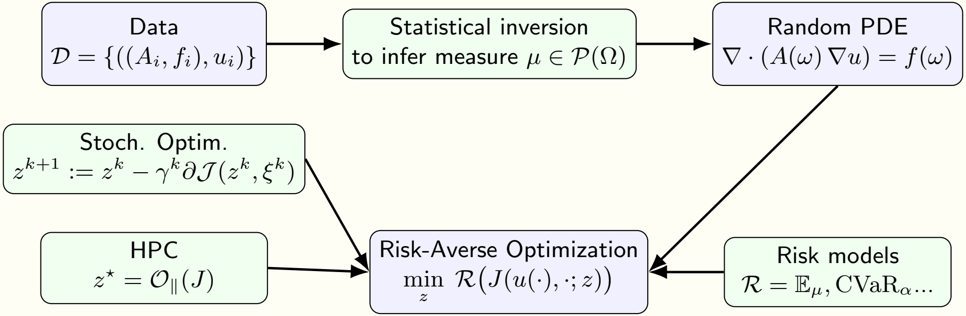
Given $x_0 \in \mathcal{X}_{\text{ad}}$, $r_0 \in (0,\infty)$, $\lambda_0 \in \mathfrak{A}$, and stationarity error: $\mathfrak{S}(x,\lambda,r) := \|x - \mathfrak{P}_{\mathcal{X}_{\text{ad}}}(x - \nabla_x L(x,\lambda,r))\|_{\mathcal{X}}$
Parameters $\rho_x \in (0,1)$, $\rho_\lambda \in (0,1)$, $\rho_r \in (1,\infty)$,
Tolerances $0 < \tau_x < \tau_{x,0}$, and $0 < \tau_\lambda < \tau_{\lambda,0}$.
for $k = 0,1,2,\dots$ **do**

1. Find $x_{k+1} \in \mathcal{X}_{\text{ad}}$ s.t. $\mathfrak{S}(x_{k+1},\lambda_k,r_k) \le \tau_{x,k}$ ▷ Approximate primal solution
2. Set $\lambda_{k+1} = \Lambda(x_{k+1},\lambda_k,r_k)$ ▷ Dual update
3. **if** $\mathfrak{S}(x_{k+1},\lambda_k,r_k) \le \tau_x$ *and* $\|\lambda_k - \lambda_{k+1}\|_{\mathcal{Y}} \le \tau_\lambda$ **then**
   return x_{k+1}
   ▷ Check for primal and dual convergence
4. **if** $\|\lambda_k - \lambda_{k+1}\|_{\mathcal{Y}} > \tau_{\lambda,k}$ **then**
   $r_{k+1} = \rho_r r_k$
   ▷ Increase penalty r_k if dual variable changes significantly
5. Set $\tau_{x,k+1} = \rho_x \tau_{x,k}$ and $\tau_{\lambda,k+1} = \rho_\lambda \tau_{\lambda,k}$. ▷ Decrease tolerances for increased accuracy (continuation)

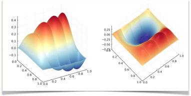
[4] See D.P. Kouri, T.M. Surowiec *A primal-dual algorithm for risk minimization*. Math. Programm. 193, 337–363 (2022). for a full convergence theory in the continuous setting.

From data to decision-making



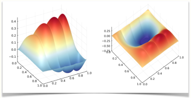


∞-Dim. Stochastic Optimization?



$$\min_{z \in \mathcal{Z}_{\mathsf{ad}}} \mathbb{E}[J(S(z), z)]$$

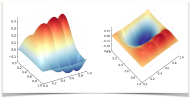
∞-Dim. Stochastic Optimization?



$$\min_{z \in \mathcal{Z}_{\text{ad}}} \mathbb{E}[J(S(z), z)]$$

PD-Risk (and related methods) require:

∞-Dim. Stochastic Optimization?

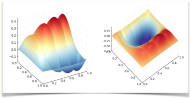


$$\min_{z \in \mathcal{Z}_{\text{ad}}} \mathbb{E}[J(S(z), z)]$$

PD-Risk (and related methods) require:

Gradients $\nabla_z \mathbb{E}[J(S(z), z)]$

∞-Dim. Stochastic Optimization?



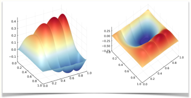
$$\min_{z \in \mathcal{Z}_{\text{ad}}} \mathbb{E}[J(S(z), z)]$$

PD-Risk (and related methods) require:

Gradients $\nabla_z \mathbb{E}[J(S(z), z)]$

Hessian-vector products $\nabla_z^2 \mathbb{E}[J(S(z), z)]\delta z$

∞-Dim. Stochastic Optimization?



$$\min_{z \in \mathcal{Z}_{\text{ad}}} \mathbb{E}[J(S(z), z)]$$

PDE-Optimization:

Replace Z by a finite-dim. space Z_h.

$S(z)$ requires the solution of a PDE.

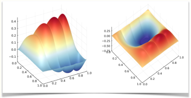
PDEs can be very expensive to solve.

PD-Risk (and related methods) require:

Gradients $\nabla_z \mathbb{E}[J(S(z), z)]$

Hessian-vector products $\nabla_z^2 \mathbb{E}[J(S(z), z)]\delta z$

∞-Dim. Stochastic Optimization?



$$\min_{z \in \mathcal{Z}_{\text{ad}}} \mathbb{E}[J(S(z), z)]$$

PD-Risk (and related methods) require:

Gradients $\nabla_z \mathbb{E}[J(S(z), z)]$

Hessian-vector products $\nabla_z^2 \mathbb{E}[J(S(z), z)]\delta z$

PDE-Optimization:

Replace Z by a finite-dim. space Z_h.

$S(z)$ requires the solution of a PDE.

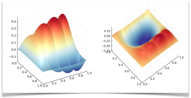
PDEs can be very expensive to solve.

Stochastic Optimization:

Cannot evaluate $S(z)$ or $S(z_h)$.

Replace $\mathbb{E}[\mathcal{J}(z)]$ by $\frac{1}{N}\sum_{i=1}^{N} \mathcal{J}(z, \xi^i)$.

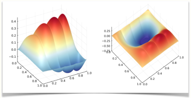
Essential computations



How do we efficiently **compute gradients**?

How do we efficiently **compute Hessian-vector products**?

Essential computations

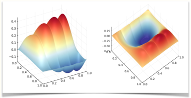


How do we efficiently **compute gradients**?

How do we efficiently **compute Hessian-vector products**?

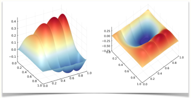
The true **Hessian** is never available, **Hessian-vector products** suffice for second-order methods.

Computing the Gradient



Given $z \in Z$, calculate **state** $u = S(z)$ by solving $e(u, z) = 0$.

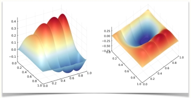
Computing the Gradient



Given $z \in Z$, calculate **state** $u = S(z)$ by solving $e(u, z) = 0$.

Given u, solve **adjoint equation** for $\lambda = P(z)$: $\partial_u e(u, z)^* \lambda = -\partial_z J(u, z)$.

Computing the Gradient



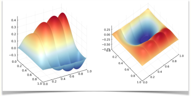
Given $z \in Z$, calculate **state** $u = S(z)$ by solving $e(u, z) = 0$.

Given u, solve **adjoint equation** for $\lambda = P(z)$: $\partial_u e(u, z)^* \lambda = -\partial_z J(u, z)$.

Given z, u, λ calculate the full gradient

$$\nabla \mathcal{J}(z) = \partial_z e(u, z)^* \lambda + \partial_u J(u, z).$$

Computing the Gradient



Given $z \in Z$, calculate **state** $u = S(z)$ by solving $e(u, z) = 0$.

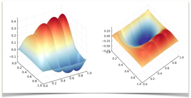
Given u, solve **adjoint equation** for $\lambda = P(z)$: $\partial_u e(u, z)^* \lambda = -\partial_z J(u, z)$.

Given z, u, λ calculate the full gradient

$$\nabla \mathcal{J}(z) = \partial_z e(u, z)^* \lambda + \partial_u J(u, z).$$

This might require an additional solve for the true discrete gradient.

Computing the Gradient



Given $z \in Z$, calculate **state** $u = S(z)$ by solving $e(u, z) = 0$.

Given u, solve **adjoint equation** for $\lambda = P(z)$: $\partial_u e(u, z)^* \lambda = -\partial_z J(u, z)$.

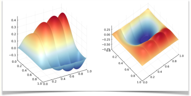
Given z, u, λ calculate the full gradient

$$\nabla \mathcal{J}(z) = \partial_z e(u, z)^* \lambda + \partial_u J(u, z).$$

This might require an additional solve for the true discrete gradient.

In the linear case, we only require the solution of **sparse structured linear** systems for $\nabla \mathcal{J}$.

Computing the Gradient



Given $z \in Z$, calculate **state** $u = S(z)$ by solving $e(u, z) = 0$.

Given u, solve **adjoint equation** for $\lambda = P(z)$: $\partial_u e(u, z)^* \lambda = -\partial_z J(u, z)$.

Given z, u, λ calculate the full gradient

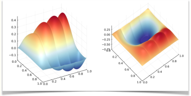
$$\nabla \mathcal{J}(z) = \partial_z e(u, z)^* \lambda + \partial_u J(u, z).$$

This might require an additional solve for the true discrete gradient.

In the linear case, we only require the solution of **sparse structured linear** systems for $\nabla \mathcal{J}$.

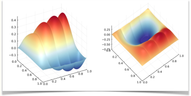
Without **adjoints**, $\nabla \mathcal{J}$ would contain large dense matrices due to the solution operators S, P.

Parallelization in Smooth Linear Case



Computing the gradient when using Monte Carlo is largely parallelizable.

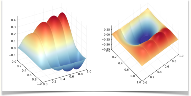
Parallelization in Smooth Linear Case



Computing the gradient when using Monte Carlo is largely parallelizable.
N **states solves** (in parallel)

$$\boldsymbol{A}(\xi^i)u = \boldsymbol{B}(\xi^i)z + f(\xi^i) \text{ in } H^{-1}(D) \quad i = 1, \ldots, N,$$

Parallelization in Smooth Linear Case



Computing the gradient when using Monte Carlo is largely parallelizable.

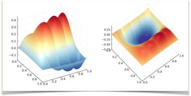
N **states solves** (in parallel)

$$\boldsymbol{A}(\xi^i)u = \boldsymbol{B}(\xi^i)z + f(\xi^i) \text{ in } H^{-1}(D) \quad i = 1, \dots, N,$$

N **adjoint solves** (in parallel, serial with i^{th} state u^i)

$$\boldsymbol{A}(\xi^i)^* \lambda = -J_u'(u(\xi^i), z) \text{ in } H^{-1}(D) \quad i = 1, \dots, N,$$

Parallelization in Smooth Linear Case



Computing the gradient when using Monte Carlo is largely parallelizable.

N **states solves** (in parallel)

$$\boldsymbol{A}(\xi^i)u = \boldsymbol{B}(\xi^i)z + f(\xi^i) \text{ in } H^{-1}(D) \quad i = 1, \dots, N,$$

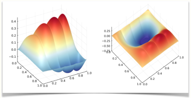
N **adjoint solves** (in parallel, serial with i^{th} state u^i)

$$\boldsymbol{A}(\xi^i)^*\lambda = -J'_u(u(\xi^i), z) \text{ in } H^{-1}(D) \quad i = 1, \dots, N,$$

N **“matrix-vector” products**

$$\boldsymbol{B}^*(\xi^i)\lambda(\xi^i) \quad i = 1, \dots, N.$$

Parallelization in Smooth Linear Case



Computing the gradient when using Monte Carlo is largely parallelizable.

N **states solves** (in parallel)

$$\boldsymbol{A}(\xi^i)u = \boldsymbol{B}(\xi^i)z + f(\xi^i) \text{ in } H^{-1}(D) \quad i = 1, \ldots, N,$$

N **adjoint solves** (in parallel, serial with i^{th} state u^i)

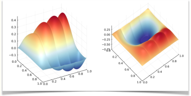
$$\boldsymbol{A}(\xi^i)^*\lambda = -J'_u(u(\xi^i), z) \text{ in } H^{-1}(D) \quad i = 1, \ldots, N,$$

N **"matrix-vector" products**

$$\boldsymbol{B}^*(\xi^i)\lambda(\xi^i) \quad i = 1, \ldots, N.$$

$\approx N$ **"axpy's"** for the weighted sum : $\nabla \mathcal{J}(z) = -\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{B}^*(\xi^i)\lambda(\xi^i)$

Parallelization in Smooth Linear Case



Computing the gradient when using Monte Carlo is largely parallelizable.

N **states solves** (in parallel)

$$\boldsymbol{A}(\xi^i)u = \boldsymbol{B}(\xi^i)z + f(\xi^i) \text{ in } H^{-1}(D) \quad i = 1, \dots, N,$$

N **adjoint solves** (in parallel, serial with i^{th} state u^i)

$$\boldsymbol{A}(\xi^i)^*\lambda = -J'_u(u(\xi^i), z) \text{ in } H^{-1}(D) \quad i = 1, \dots, N,$$

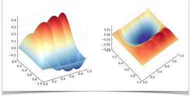
N **"matrix-vector" products**

$$\boldsymbol{B}^*(\xi^i)\lambda(\xi^i) \quad i = 1, \dots, N.$$

$\approx N$ **"axpy's"** for the weighted sum : $\nabla \mathcal{J}(z) = -\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{B}^*(\xi^i)\lambda(\xi^i)$

Don't forget the Riesz maps/proper inner products!

Essential computations

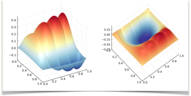


A similar computation can be made for Hessian vector products that requires:

$$N \times \ \textbf{State} \ + N \times \ \textbf{Adjoint} \ + N \times \ \textbf{State Sensitivity} \ + N \times \ \textbf{Adjoint Sensitivity}$$

solves, where each class of N solves can be made in parallel.

CVaR Minimization



We consider the following stochastic optimization problem:

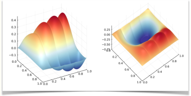
$$\min \left\{ t + \frac{1}{2(1-\beta)} \mathbb{E}\left[\left(\|S(z) - u_d\|_{L^2}^2 - t \right)_+ \right] + \frac{\alpha}{2}\|z\|_Z^2 \text{ over } (z,t) \in \mathcal{Z}_{\text{ad}} \times \mathbb{R} \right\}, \tag{1}$$

where $\mathcal{Z}_{\text{ad}} \subset Z$ is a nonempty, closed, and convex set and $S(z) = u$ is the unique solution to

$$\text{Find } u \in \mathcal{U} : \mathbb{E}\left[\int_D A\nabla u \cdot \nabla v \mathrm{d}x \right] = \mathbb{E}[\langle Bz + f, v\rangle_{U^*,U}], \quad \forall v \in \mathcal{U}.$$

where $U := H_0^1(\Omega)$, $\mathcal{U} := L^2(\Omega, \mathcal{F}, \mathbb{P}; U)$.

CVaR Minimization



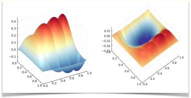
We consider the following stochastic optimization problem:

$$\min \left\{ t + \frac{1}{2(1-\beta)} \mathbb{E}\left[\|S(z) - u_d\|_{L^2}^2 - t)_+ \right] + \frac{\alpha}{2}\|z\|_Z^2 \text{ over } (z,t) \in \mathcal{Z}_{\text{ad}} \times \mathbb{R} \right\}, \tag{1}$$

where $\mathcal{Z}_{\text{ad}} \subset Z$ is a nonempty, closed, and convex set and $S(z) = u$ is the unique solution to

$$\boldsymbol{A}(\omega)u = \boldsymbol{B}(\omega)z + f(\omega) \text{ in } H^{-1}(D), \quad \mathbb{P}\text{-a.s. in } \Omega.$$

Is linear algebra **really** an issue?



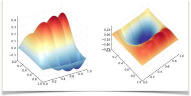
The Robust Stochastic Mirror Descent method yielded the following:

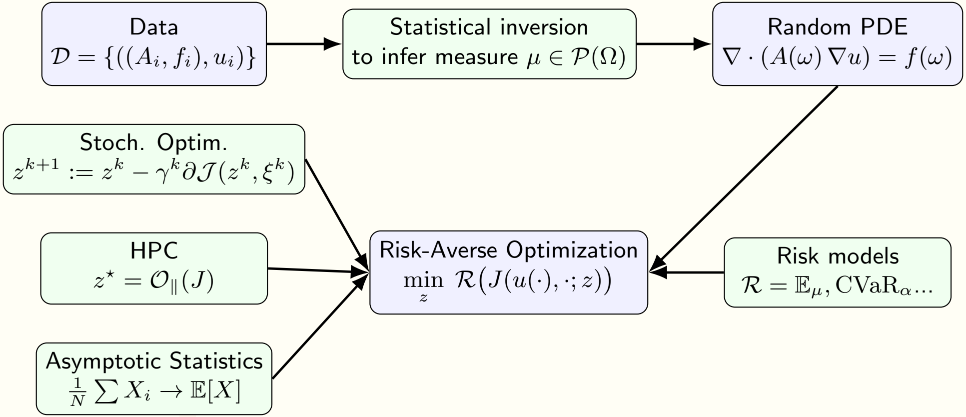
iter	time(s)	fval	abs-err f	rel-err f	abs-err x_k	rel-err x_k
100	1.4	3.1274e-01	1.3403e-01	7.4999e-01	2.7098e+02	8.3928e-01
1000	14.7	2.5017e-01	7.1464e-02	3.9989e-01	2.0949e+02	6.4885e-01
10000	152.0	2.0502e-01	2.6312e-02	1.4723e-01	1.4802e+02	4.5846e-01
100000	2054.2	1.8906e-01	1.0353e-02	5.7933e-02	1.0943e+02	3.3892e-01
1000000	104636.2	1.8411e-01	5.3994e-03	3.0213e-02	8.8822e+01	2.7510e-01

In contrast, using the PD-Risk with Monte Carlo we obtain:

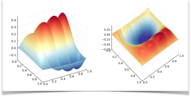
N	time(s)	fval	nstate	nadjoint	nstatesens	nadjointsens	totalsolves
100	680.0	1.7924e-01	27500	7112	36554	36554	107720
1000	2889.3	1.7871e-01	83000	30035	197168	197168	507371
10000	23540.5	1.7871e-01	700000	268612	1594077	1594077	4156766

From data to decision-making





A More *Statistical* Numerical Analysis

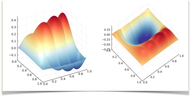


Are solutions and optimal values stable with respect to shifts in distribution?

Can this stability be quantified?

What happens asymptotically in the “big data” limit?

A More *Statistical* Numerical Analysis



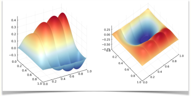
Are solutions and optimal values stable with respect to shifts in distribution?

Can this stability be quantified?

What happens asymptotically in the "big data" limit?

Fundamentally different questions to traditional numerical analysis in PDEs and optimal control:

A More *Statistical* Numerical Analysis



Are solutions and optimal values stable with respect to shifts in distribution?

Can this stability be quantified?

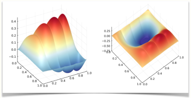
What happens asymptotically in the "big data" limit?

Fundamentally different questions to traditional numerical analysis in PDEs and optimal control:

FEM: finite dimensional spaces replace ∞-dimensional ones, but we always use the Lebesgue measure.

Mesh refinement increases the dimension of these spaces, but refinements are not random.

Defining Stability[5]

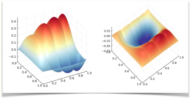


Optimal Value: $$\nu(P) = \inf_{z \in \mathcal{Z}_{\mathrm{ad}}} \int_\Omega f(z,\omega)\, \mathrm{d}P(\omega).$$

Optimal Solutions: $$z(P) \in \operatorname*{argmin}_{z \in \mathcal{Z}_{\mathrm{ad}}} \int_\Omega f(z,\omega)\, \mathrm{d}P(\omega)$$

[5] See J. Milz, T.M. Surowiec *Asymptotic consistency for nonconvex risk-averse stochastic optimization with infinite-dimensional decision spaces* Mathematics of Operations Research 49 (3), 1403-1418 (2024) for results on the risk averse case.

Defining Stability[5]



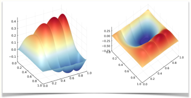
$$\text{Optimal Value:} \qquad \nu(P) = \inf_{z \in \mathcal{Z}_{\text{ad}}} \int_\Omega f(z,\omega)\, \mathrm{d}P(\omega).$$

$$\text{Optimal Solutions:} \quad z(P) \in \operatorname*{argmin}_{z \in \mathcal{Z}_{\text{ad}}} \int_\Omega f(z,\omega)\, \mathrm{d}P(\omega)$$

How does ν or z change, when we replace P by Q?

[5] See J. Milz, T.M. Surowiec *Asymptotic consistency for nonconvex risk-averse stochastic optimization with infinite-dimensional decision spaces* Mathematics of Operations Research 49 (3), 1403-1418 (2024) for results on the risk averse case.

Defining Stability[5]



$$\text{Optimal Value:} \qquad \nu(P) = \inf_{z \in \mathcal{Z}_{\mathrm{ad}}} \int_\Omega f(z,\omega)\, \mathrm{d}P(\omega).$$

$$\text{Optimal Solutions:} \quad z(P) \in \operatorname*{argmin}_{z \in \mathcal{Z}_{\mathrm{ad}}} \int_\Omega f(z,\omega)\, \mathrm{d}P(\omega)$$

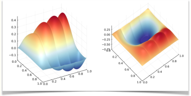
How does ν or z change, when we replace P by Q?

If $Q := P_N$ such that $P_N \Rightarrow P$, does it hold that

$$\nu(P_N) \to \nu(P) \text{ and } \|z(P_N) - z(P)\|_{\mathcal{Z}} \to 0?$$

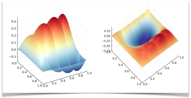
[5] See J. Milz, T.M. Surowiec *Asymptotic consistency for nonconvex risk-averse stochastic optimization with infinite-dimensional decision spaces* Mathematics of Operations Research 49 (3), 1403-1418 (2024) for results on the risk averse case.

A (Very) Brief History



Before 1980s: Foundational work on consistency in Maximum Likelihood Estimation (e.g., Wald 1949, Huber 1967).

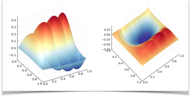
A (Very) Brief History



Before 1980s: Foundational work on consistency in Maximum Likelihood Estimation (e.g., Wald 1949, Huber 1967).

1980s - 1990s: Extensive stability analysis for Stochastic Programming (finite-dimensions) (e.g., Wets, Dupačová 1988, Shapiro 1991 King, Rockafellar 1993).

A (Very) Brief History

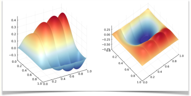


Before 1980s: Foundational work on consistency in Maximum Likelihood Estimation (e.g., Wald 1949, Huber 1967).

1980s - 1990s: Extensive stability analysis for Stochastic Programming (finite-dimensions) (e.g., Wets, Dupačová 1988, Shapiro 1991 King, Rockafellar 1993).

1990s - Present: Increasingly more complex parameter spaces and objective functions, e.g., nonsmooth objectives, constraints, integer variables
(e.g. Shapiro, Homem-de-Mello, Pflug, Römisch...)

A (Very) Brief History



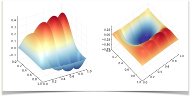
Before 1980s: Foundational work on consistency in Maximum Likelihood Estimation (e.g., Wald 1949, Huber 1967).

1980s - 1990s: Extensive stability analysis for Stochastic Programming (finite-dimensions) (e.g., Wets, Dupačová 1988, Shapiro 1991 King, Rockafellar 1993).

1990s - Present: Increasingly more complex parameter spaces and objective functions, e.g., nonsmooth objectives, constraints, integer variables
(e.g. Shapiro, Homem-de-Mello, Pflug, Römisch...)

Our Core Framework: Based on the metric approach to studying optimal solutions under varying probability laws, particularly following Rachev & Römisch (2002).

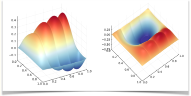
Personal Contributions



- M. Hoffhues, W. Römisch, T.M. Surowiec
  On quantitative stability in infinite-dimensional optimization under uncertainty
  Optim Lett 15, 2733–2756 (2021). https://doi.org/10.1007/s11590-021-01707-2
- W. Römisch, T.M. Surowiec
  Asymptotic properties of Monte Carlo methods in elliptic PDE-constrained optimization under uncertainty
  Numer. Math. 156, 1887–1914 (2024). https://doi.org/10.1007/s00211-024-01436-5

Figure: Werner Römisch, 28.12.1947 - 12.7.2024

Abstract Setting



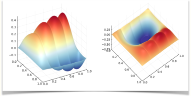
Decision variables are taken from $\mathcal{Z}$, an infinite dimensional Hilbert space.

We assume the randomness arises from a random element $\xi(\omega)$ and work with the law of ξ.

Ξ is a metric space, $P \in \mathcal{P}(\Xi)$ the set of all Borel probability measures.

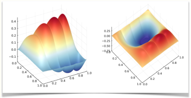
The probability measures themselves are the exogenous parameters.

A Class of Random Operator Equations



We derive a class of integrands $f : \mathcal{Z} \times \Xi \to \mathbb{R}$ to motivate the general framework.

A Class of Random Operator Equations

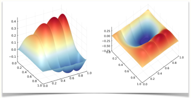


We derive a class of integrands $f : \mathcal{Z} \times \Xi \to \mathbb{R}$ to motivate the general framework.

Our random operator equation will be written in the form

$$\boldsymbol{A}(\xi)u = z + g(\xi) \quad (P\text{-a.e. } \xi \in \Xi).$$

A Class of Random Operator Equations



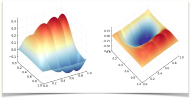
We derive a class of integrands $f : \mathcal{Z} \times \Xi \to \mathbb{R}$ to motivate the general framework.

Our random operator equation will be written in the form

$$\boldsymbol{A}(\xi)u = z + g(\xi) \quad (P\text{-a.e. } \xi \in \Xi).$$

The real Hilbert spaces $V \subset H \subset V^*$ constitute a Gelfand triple.

A Class of Random Operator Equations



We derive a class of integrands $f : \mathcal{Z} \times \Xi \to \mathbb{R}$ to motivate the general framework.

Our random operator equation will be written in the form

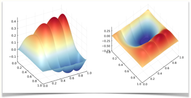
$$\boldsymbol{A}(\xi)u = z + g(\xi) \quad (P\text{-a.e. } \xi \in \Xi).$$

The real Hilbert spaces $V \subset H \subset V^*$ constitute a Gelfand triple.

There exist $0 < \gamma < L < \infty$, for each $\xi \in \Xi$, such that $\boldsymbol{A}(\xi) \in \mathcal{L}(V, V^*)$ and

$$\underbrace{\gamma \|v\|_V^2 \leq \langle A(\xi)v, v\rangle \quad \forall v \in V}_{\text{uniform coercivity}} \quad \text{and} \quad \underbrace{\langle A(\xi)v, w\rangle \leq L\|v\|\|w\| \quad \forall v, w \in V}_{\text{uniform boundedness}}$$

A Class of Random Operator Equations



We derive a class of integrands $f : \mathcal{Z} \times \Xi \to \mathbb{R}$ to motivate the general framework.

Our random operator equation will be written in the form

$$\boldsymbol{A}(\xi)u = z + g(\xi) \quad (P\text{-a.e. } \xi \in \Xi).$$

The real Hilbert spaces $V \subset H \subset V^*$ constitute a Gelfand triple.

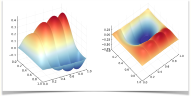
There exist $0 < \gamma < L < \infty$, for each $\xi \in \Xi$, such that $\boldsymbol{A}(\xi) \in \mathcal{L}(V, V^*)$ and

$$\underbrace{\gamma\|v\|_V^2 \leq \langle A(\xi)v, v\rangle \quad \forall v \in V}_{\text{uniform coercivity}} \quad \text{and} \quad \underbrace{\langle A(\xi)v, w\rangle \leq L\|v\|\|w\| \quad \forall v, w \in V}_{\text{uniform boundedness}}$$

$A : \Xi \to \mathcal{L}(V, V^*)$ is measurable and essentially bounded.

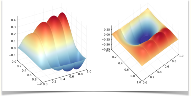
$g : \Xi \to V^*$ is measurable and essentially bounded (or more regular).

Linear Quadratic Risk-Neutral Problems



We derive a class of integrands $f : \mathcal{Z} \times \Xi \to \mathbb{R}$ to motivate the general framework.

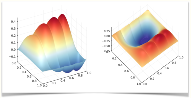
Linear Quadratic Risk-Neutral Problems



We derive a class of integrands $f : \mathcal{Z} \times \Xi \to \mathbb{R}$ to motivate the general framework.

$\boldsymbol{A}(\xi)^{-1} : V^\star \to V$ exists and is positive definite (with $\frac{1}{L}$) and bounded (with $\frac{1}{\gamma}$).

Linear Quadratic Risk-Neutral Problems



We derive a class of integrands $f : \mathcal{Z} \times \Xi \to \mathbb{R}$ to motivate the general framework.

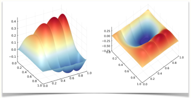
$\boldsymbol{A}(\xi)^{-1} : V^\star \to V$ exists and is positive definite (with $\frac{1}{L}$) and bounded (with $\frac{1}{\gamma}$).

We consider the integrand inspired by PDE-constrained optimization:

$$f(z,\xi) \quad = \quad \frac{1}{2}\left\|\boldsymbol{A}(\xi)^{-1}(z - g(\xi)) - u_d\right\|_H^2 + \frac{\alpha}{2}\|z\|_{\mathcal{Z}}^2 \quad (\alpha > 0, u_d \in H)$$

for $z \in \mathcal{Z}_{\mathrm{ad}}$ (closed, bounded, convex set) and $\xi \in \Xi$.

Linear Quadratic Risk-Neutral Problems



We derive a class of integrands $f : \mathcal{Z} \times \Xi \to \mathbb{R}$ to motivate the general framework.

$\boldsymbol{A}(\xi)^{-1} : V^\star \to V$ exists and is positive definite (with $\frac{1}{L}$) and bounded (with $\frac{1}{\gamma}$).

We consider the integrand inspired by PDE-constrained optimization:

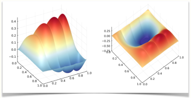
$$f(z,\xi) \quad = \quad \frac{1}{2}\left\|\boldsymbol{A}(\xi)^{-1}(z - g(\xi)) - u_d\right\|_H^2 + \frac{\alpha}{2}\|z\|_{\mathcal{Z}}^2 \quad (\alpha > 0, u_d \in H)$$

for $z \in \mathcal{Z}_{\mathrm{ad}}$ (closed, bounded, convex set) and $\xi \in \Xi$.

This leads to the risk-neutral optimization problem:

$$\min \left\{ \mathbb{E}_P[f(z)] = \int_\Xi f(z,\xi)\mathrm{d}P(\xi) : z \in \mathcal{Z}_{\mathrm{ad}} \right\}.$$

A Growth Condition

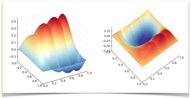


Lemma (Hoffhues, Römisch, Surowiec (2021))

For each $P \in \mathcal{P}(\Xi)$ *and any* $z \in \mathcal{Z}_{\mathrm{ad}}$ *we have*

$$\|z - z(P)\|_{\mathcal{Z}}^2 \leq \frac{8}{\alpha}(\mathbb{E}_P[f(z)] - v(P))$$

A Growth Condition



Lemma (Hoffhues, Römisch, Surowiec (2021))

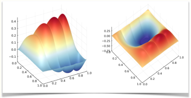
For each $P \in \mathcal{P}(\Xi)$ *and any* $z \in \mathcal{Z}_{\mathrm{ad}}$ *we have*

$$\|z - z(P)\|_{\mathcal{Z}}^2 \leq \frac{8}{\alpha}(\mathbb{E}_P[f(z)] - v(P))$$

For any two $P, Q \in \mathcal{P}(\Xi)$ we immediately deduce

$$\begin{aligned}\|z(Q) - z(P)\|_{\mathcal{Z}}^2 \leq \frac{4}{\alpha}\Big[&\mathbb{E}_P[f(z(Q))] - \mathbb{E}_Q[f(z(Q))] \\ &+ \mathbb{E}_Q[f(z(P))] - \mathbb{E}_P[f(z(P))]\Big]\end{aligned}$$

A Growth Condition



Lemma (Hoffhues, Römisch, Surowiec (2021))

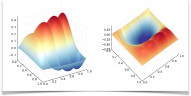
For each $P \in \mathcal{P}(\Xi)$ *and any* $z \in \mathcal{Z}_{\mathrm{ad}}$ *we have*

$$\|z - z(P)\|_{\mathcal{Z}}^2 \leq \frac{8}{\alpha}(\mathbb{E}_P[f(z)] - v(P))$$

For any two $P, Q \in \mathcal{P}(\Xi)$ we immediately deduce

$$\|z(Q) - z(P)\|_{\mathcal{Z}}^2 \leq \frac{4}{\alpha}\Big[\mathbb{E}_P[f(z(Q))] - \mathbb{E}_Q[f(z(Q))] + \mathbb{E}_Q[f(z(P))] - \mathbb{E}_P[f(z(P))]\Big] \Leftarrow \text{Difference of expectations}$$

Zolotarev's ζ-distances

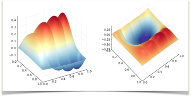


Zolotarev ζ-distance on $\mathcal{P}(\Xi)$ (Zolotarev 83):

$$d_{\mathfrak{F}}(\mathbb{P},\mathbb{Q}) = \sup_{f\in\mathfrak{F}} |\mathbb{E}_P[f] - \mathbb{E}_Q[f]| \Leftarrow \text{A difference of expectations!}$$

where $\mathfrak{F}$ is a family of real-valued Borel measurable functions on Ξ and $\mathbb{P},\mathbb{Q} \in \mathcal{P}(\Xi)$.

Probability Metrics



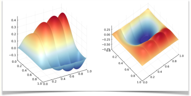
Compared with classical probability metrics we consider a much smaller family $\mathfrak{F}$:

$$\mathfrak{F}_{\mathrm{mi}} = \{f(z,\cdot) : z \in \mathcal{Z}_{\mathrm{ad}}\} \quad \text{(minimal information (m.i.) family).}$$

This leads to a pseudometric[a]: the minimal information (m.i.) distance $d_{\mathfrak{F}_{\mathrm{mi}}}$.

[a] Distance between distinct points can be zero.

Probability Metrics



Compared with classical probability metrics we consider a much smaller family $\mathfrak{F}$:

$$\mathfrak{F}_{\mathrm{mi}} = \{f(z,\cdot) : z \in \mathcal{Z}_{\mathrm{ad}}\} \quad \text{(minimal information (m.i.) family).}$$

This leads to a pseudometric[a]: the minimal information (m.i.) distance $d_{\mathfrak{F}_{\mathrm{mi}}}$.

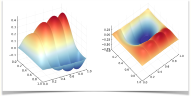
[a] Distance between distinct points can be zero.

Rachev & Römisch: Use the right metric for your problem class!

Wasserstein W_1 is a ζ-distance, but $\mathfrak{F}$[a] is too rich and would give worse convergence rates.

[a] set of all 1-Lipschitz functions on Ξ

Lipschitz-type Estimates for $d_{\mathfrak{F}_{\mathrm{mi}}}$



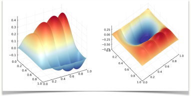
Theorem (Hoffhues, Römisch, Surowiec (2021))

Under the standing assumptions we obtain the estimates

$$
\begin{aligned}
|v(Q) - v(P)| &\leq d_{\mathfrak{F}_{mi}}(P, Q) \\
\|z(Q) - z(P)\|_{\mathcal{Z}} &\leq 2\sqrt{\frac{2}{\alpha} d_{\mathfrak{F}_{mi}}(P, Q)^{\frac{1}{2}}}
\end{aligned}
$$

for P and $Q \in \mathcal{P}(\Xi)$.

A Refined Estimate



Theorem (Römisch, Surowiec (2024))

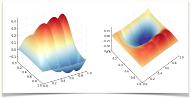
Under the standing assumptions the Lipschitz-type estimate

$$\|z(Q) - z(P)\|_H \leq \frac{8}{\alpha} d_{\mathfrak{F}_{di}}(P, Q)$$

holds for all $P, Q \in \mathcal{P}(\Xi)$, *where* $\mathfrak{F}_{di}$ *denotes the following function class on* Ξ

$$\mathfrak{F}_{di} = \{\langle \partial_z f(z, \cdot), h \rangle_H : z \in Z_{\mathrm{ad}}, \|h\|_H \leq 1\}.$$

A Refined Estimate



Theorem (Römisch, Surowiec (2024))

Under the standing assumptions the Lipschitz-type estimate

$$\|z(Q) - z(P)\|_H \leq \frac{8}{\alpha} d_{\mathfrak{F}_{di}}(P, Q)$$

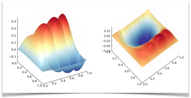
holds for all $P, Q \in \mathcal{P}(\Xi)$*, where* $\mathfrak{F}_{di}$ *denotes the following function class on* Ξ

$$\mathfrak{F}_{di} = \{\langle \partial_z f(z, \cdot), h\rangle_H : z \in Z_{\mathrm{ad}}, \|h\|_H \leq 1\}.$$

Quantifies changes under shifts in the underlying distribution.

Nonasymptotic, what about when $Q = P_N$ with $N \to +\infty$?

A Key Property for Asymptotics

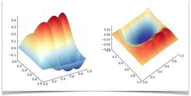


Need $\mathfrak{F}$ to be a $\mathbb{P}$-uniformity class[a], i.e. $(\mathbb{P}_N)$, $\mathbb{P} \in \mathcal{P}(\Xi)$, weak conv. of $(\mathbb{P}_N)$ to $\mathbb{P}$ implies

$$\lim_{N \to \infty} d_{\mathfrak{F}}(\mathbb{P}_N, \mathbb{P}) = 0.$$

[a] Can be guaranteed by, e.g., Topsøe (1967) under uniform boundedness and equicontinuity conditions

A Key Property for Asymptotics



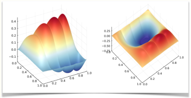
Need $\mathfrak{F}$ to be a $\mathbb{P}$-uniformity class[a], i.e. $(\mathbb{P}_N)$, $\mathbb{P} \in \mathcal{P}(\Xi)$, weak conv. of $(\mathbb{P}_N)$ to $\mathbb{P}$ implies

$$\lim_{N \to \infty} d_{\mathfrak{F}}(\mathbb{P}_N, \mathbb{P}) = 0.$$

[a] Can be guaranteed by, e.g., Topsøe (1967) under uniform boundedness and equicontinuity conditions

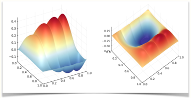
The standing assumptions imply both $\mathfrak{F}_{mi}$ and $\mathfrak{F}_{di}$ are $\mathbb{P}$-uniformity classes.

Monte Carlo Approximation



For an iid random sample $\{\xi^i\}_{i=1}^n$ with law P, the empirical measure $P_n(\cdot)$ is a perturbation of P.

Monte Carlo Approximation



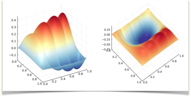
For an iid random sample $\{\xi^i\}_{i=1}^n$ with law P, the empirical measure $P_n(\cdot)$ is a perturbation of P.

By the LLN $(P_n(\cdot)) \Rightarrow P$ $\mathbb{P}$-almost surely.

Since $\mathfrak{F}_{mi}$, $\mathfrak{F}_{di}$ are P-uniformity classes, it holds that

$$v(P_n(\cdot)) \to v(P) \text{ and } z(P_n(\cdot)) \to z(P) \qquad \mathbb{P}\text{-a.s.}$$

Monte Carlo Approximation



For an iid random sample $\{\xi^i\}_{i=1}^n$ with law P, the empirical measure $P_n(\cdot)$ is a perturbation of P.

By the LLN $(P_n(\cdot)) \Rightarrow P$ $\mathbb{P}$-almost surely.

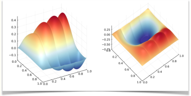
Since $\mathfrak{F}_{mi}$, $\mathfrak{F}_{di}$ are P-uniformity classes, it holds that

$$v(P_n(\cdot)) \to v(P) \text{ and } z(P_n(\cdot)) \to z(P) \qquad \mathbb{P}\text{-a.s.}$$

P-uniformity (consistency) and stability implies convergence of the estimators![a]

[a] Cf. Lax-Richtmyer (1956) Equivalence Theorem: Stability + Consistency $\Rightarrow$ Convergence for numerical PDEs!

Rate of Convergence and CLT



Theorem (Römisch, Surowiec (2024))

$\Xi \subset \mathbb{R}^d$ *is bounded, convex set and* $\Xi \subseteq \mathrm{cl\,int}\,\Xi$

$A(\cdot)$, $g(x,\cdot)$ *have continuous partial derivatives up to order* k[a], *with bounded, measurable derivatives,*

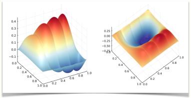
$k \in \mathbb{N}$ *satisfies* $d < 2k$.

Then:

1. $\mathbb{E}_{\mathbb{P}}[|v(P_n(\cdot)) - v(P)|] = O(n^{-\frac{1}{2}})$,
2. $\mathbb{E}_{\mathbb{P}}[\|z(P_n(\cdot)) - z(P)\|_H] = O(n^{-\frac{1}{2}})$,
3. $(\sqrt{n}(v(P_n(\cdot)) - v(P))) \rightsquigarrow \mathcal{N}(0, P(f(z(P)))^2)$.

[a] A is generated by a second-order elliptic operator with coefficient functions $a_{ij}(x,\xi)$ exibiting the requisite smoothness.

A Random Elliptic Operator



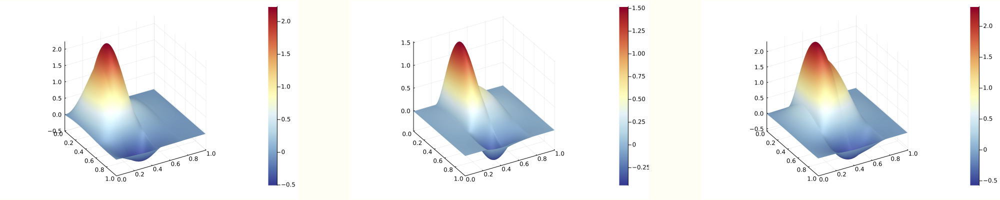
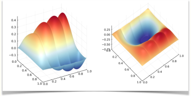


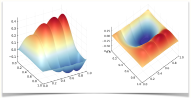
Figure: Uncontrolled, random states: Three realizations of $u(\xi)$ computed by setting $z \equiv 0$.

Convergence Rate Validation



Reference Solution: Compute $v(P_n)$ and $z(P_n)$ with sample size $n = 500$ and FEM for function spaces.

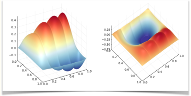
Convergence Rate Validation



Reference Solution: Compute $v(P_n)$ and $z(P_n)$ with sample size $n = 500$ and FEM for function spaces.

Test Runs: Select a range of smaller sample sizes $m = 1, \ldots, 100$.

Convergence Rate Validation

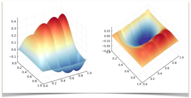


Reference Solution: Compute $v(P_n)$ and $z(P_n)$ with sample size $n = 500$ and FEM for function spaces.

Test Runs: Select a range of smaller sample sizes $m = 1, \ldots, 100$.

Validation: For each m, run the experiment $M = 100$ times to generate a sample of solutions and values.

Convergence Rate Validation



Reference Solution: Compute $v(P_n)$ and $z(P_n)$ with sample size $n = 500$ and FEM for function spaces.

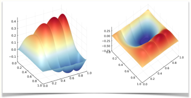
Test Runs: Select a range of smaller sample sizes $m = 1, \ldots, 100$.

Validation: For each m, run the experiment $M = 100$ times to generate a sample of solutions and values.

Metrics: Compute the error against the reference solution P_n:

$$\|z(P_{m,j}) - z(P_n)\|_{L^2} \text{ and } |v(P_{m,j}) - v(P_n)|.$$

Convergence Rate Validation



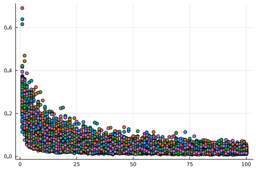
Reference Solution: Compute $v(P_n)$ and $z(P_n)$ with sample size $n = 500$ and FEM for function spaces.

Test Runs: Select a range of smaller sample sizes $m = 1, \ldots, 100$.

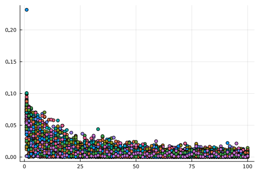
Validation: For each m, run the experiment $M = 100$ times to generate a sample of solutions and values.

Metrics: Compute the error against the reference solution P_n:

$$\|z(P_{m,j}) - z(P_n)\|_{L^2} \text{ and } |v(P_{m,j}) - v(P_n)|.$$

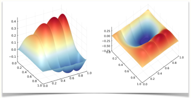


Solution Error (Control)



Value Error

Convergence Rate Validation



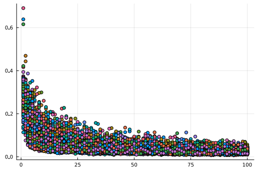
Reference Solution: Compute $v(P_n)$ and $z(P_n)$ with sample size $n = 500$ and FEM for function spaces.

Test Runs: Select a range of smaller sample sizes $m = 1, \ldots, 100$.

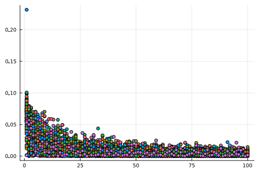
Validation: For each m, run the experiment $M = 100$ times to generate a sample of solutions and values.

Metrics: Compute the error against the reference solution P_n:

$$\|z(P_{m,j}) - z(P_n)\|_{L^2} \text{ and } |v(P_{m,j}) - v(P_n)|.$$



Solution Error (Control)

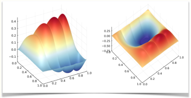


Value Error

Empirical Convergence Rates:

- **Optimal Solution (z)**: $O(m^{-0.536})$
- **Optimal Value (v)**: $O(m^{-0.660})$

Using the CLT

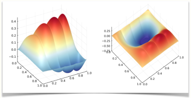


We know the estimated optimal value $v(P_n)$ converges asymptotically to the true value $v(P)$:

$$\sqrt{n}(v(P_n) - v(P)) \xrightarrow{d} \mathcal{N}(0, P(f(z(P)))^2)$$

where $P(f(z(P)))^2$ is the asymptotic variance.

Using the CLT



We know the estimated optimal value $v(P_n)$ converges asymptotically to the true value $v(P)$:

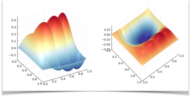
$$\sqrt{n}(v(P_n) - v(P)) \xrightarrow{d} \mathcal{N}(0, P(f(z(P)))^2)$$

where $P(f(z(P)))^2$ is the asymptotic variance.

The practical limiations are

- The true optimal value, $v(P)$, is **unknown**.
- The asymptotic variance, $P(f(z(P)))^2$, is **unknown**.
- We cannot directly use the CLT to construct confidence intervals.

Using the CLT



We know the estimated optimal value $v(P_n)$ converges asymptotically to the true value $v(P)$:

$$\sqrt{n}(v(P_n) - v(P)) \xrightarrow{d} \mathcal{N}(0, P(f(z(P)))^2)$$

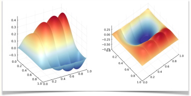
where $P(f(z(P)))^2$ is the asymptotic variance.

The practical limiations are

- The true optimal value, $v(P)$, is **unknown**.
- The asymptotic variance, $P(f(z(P)))^2$, is **unknown**.
- We cannot directly use the CLT to construct confidence intervals.

We remedy this with subsampling, which provides an asymptotically equivalent distribution that **mimics the true limiting distribution** $\mathcal{N}(0, P(f(z(P)))^2)$.

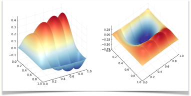
The Subsampling Mechanics



The Idea: Right Distribution, Wrong Size

- **Traditional Bootstrap** $\rightarrow$ Uses size n (right size), often inconsistent.
- **Subsampling** $\rightarrow$ Uses size $b \ll n$ (wrong size), achieves the **right limiting distribution**.

The Subsampling Mechanics

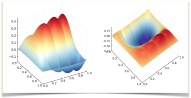


The Idea: Right Distribution, Wrong Size

- **Traditional Bootstrap** $\rightarrow$ Uses size n (right size), often inconsistent.
- **Subsampling** $\rightarrow$ Uses size $b \ll n$ (wrong size), achieves the **right limiting distribution**.

- **Original Sample Size**: n (large).
- **Subsample Size**: Choose b, such that $b \rightarrow \infty$ but $\frac{b}{n} \rightarrow 0$. (sample without replacement)
- **Iteration Count**: m replicates, $m \rightarrow \infty$.
- For $j = 1, \ldots, m$, draw subsample, compute $v(P_n^*(N_j^{n,b}))$, optimal value for subsample.

The Subsampling Mechanics



The Idea: Right Distribution, Wrong Size

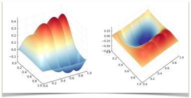
- **Traditional Bootstrap** $\to$ Uses size n (right size), often inconsistent.
- **Subsampling** $\to$ Uses size $b \ll n$ (wrong size), achieves the **right limiting distribution**.

- **Original Sample Size**: n (large).
- **Subsample Size**: Choose b, such that $b \to \infty$ but $\frac{b}{n} \to 0$. (sample without replacement)
- **Iteration Count**: m replicates, $m \to \infty$.
- For $j = 1, \ldots, m$, draw subsample, compute $v(P_n^*(N_j^{n,b}))$, optimal value for subsample.

The distribution of the subsampled statistic converges to the desired asymptotic distribution:

$$\sqrt{b}(v(P_n^*(N_j^{n,b})) - v(P_n)) \xrightarrow{d} \mathcal{N}(0, P(f(z(P)))^2)$$

Practical Application

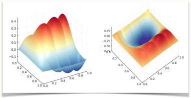


Using $L_{n,b}(\cdot)$, the empirical CDF of the m subsample statistics, $(1-\alpha)$ confidence interval for $v(P)$:

$$\mathsf{CI}_{1-\alpha}(v(P)) = \left[v(P_n) - n^{-1/2}L_{n,b}^{-1}(1-\alpha/2), \quad v(P_n) - n^{-1/2}L_{n,b}^{-1}(\alpha/2)\right]$$

$v(P_n)$ provides the center; $n^{-1/2}$ scales the empirical quantiles $L_{n,b}^{-1}(\cdot)$.

Practical Application



Using $L_{n,b}(\cdot)$, the empirical CDF of the m subsample statistics, $(1-\alpha)$ confidence interval for $v(P)$:

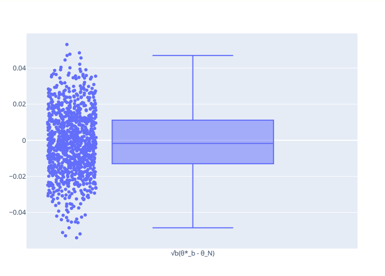
$$\mathrm{CI}_{1-\alpha}(v(P)) = \left[v(P_n) - n^{-1/2}L_{n,b}^{-1}(1-\alpha/2), \quad v(P_n) - n^{-1/2}L_{n,b}^{-1}(\alpha/2)\right]$$

$v(P_n)$ provides the center; $n^{-1/2}$ scales the empirical quantiles $L_{n,b}^{-1}(\cdot)$.

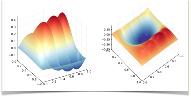
Parameters Used: $n = 2000$, $b = 1000$, $m = 1000$
Confidence Level: 95% $(\alpha = 0.05)$ yields

$$\mathbf{CI}_{95\%} = [0.089148, \quad 0.090720]$$

Validation Check: The 95% CI was found to capture the true optimal value in 84 out of 100 runs.



Conclusion

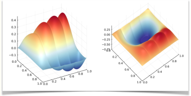


Key Scientific Achievements

Modeling $\longrightarrow$ Algorithms $\longrightarrow$ Implementation $\longrightarrow$ Stability

Open Challenges & Future Directions

Conclusion



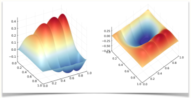
Key Scientific Achievements

Modeling $\rightarrow$ Algorithms $\rightarrow$ Implementation $\rightarrow$ Stability

PD-Risk: A robust primal-dual framework for **non-smooth risk-averse optimization** in ∞-dimensions.

Open Challenges & Future Directions

Conclusion



Key Scientific Achievements

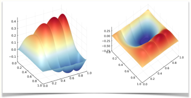
Modeling $\rightarrow$ Algorithms $\rightarrow$ Implementation $\rightarrow$ Stability

PD-Risk: A robust primal-dual framework for **non-smooth risk-averse optimization** in ∞-dimensions.

Scalability via **adjoint states** and **parallelizable** gradient and Hess-vec computations.

Open Challenges & Future Directions

Conclusion



Key Scientific Achievements

Modeling $\rightarrow$ Algorithms $\rightarrow$ Implementation $\rightarrow$ Stability

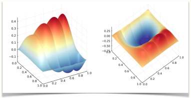
PD-Risk: A robust primal-dual framework for **non-smooth risk-averse optimization** in ∞-dimensions.

Scalability via **adjoint states** and **parallelizable** gradient and Hess-vec computations.

Rigorous **quantitative stability bounds** using probability metrics like $\|z(Q) - z(P)\| \leq C d_{\mathcal{F}}(P, Q)$.

Open Challenges & Future Directions

Conclusion



Key Scientific Achievements

Modeling $\rightarrow$ Algorithms $\rightarrow$ Implementation $\rightarrow$ Stability

PD-Risk: A robust primal-dual framework for **non-smooth risk-averse optimization** in ∞-dimensions.

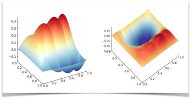
Scalability via **adjoint states** and **parallelizable** gradient and Hess-vec computations.

Rigorous **quantitative stability bounds** using probability metrics like $\|z(Q) - z(P)\| \leq C d_{\mathcal{F}}(P, Q)$.

Asymptotic Consistency and $O(n^{-1/2})$ rates in ∞-dimensional spaces, validated by empirical experiments.

Open Challenges & Future Directions

Conclusion



Key Scientific Achievements

Modeling $\rightarrow$ Algorithms $\rightarrow$ Implementation $\rightarrow$ Stability

PD-Risk: A robust primal-dual framework for **non-smooth risk-averse optimization** in ∞-dimensions.

Scalability via **adjoint states** and **parallelizable** gradient and Hess-vec computations.

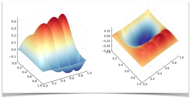
Rigorous **quantitative stability bounds** using probability metrics like $\|z(Q) - z(P)\| \leq C d_{\mathcal{F}}(P, Q)$.

Asymptotic Consistency and $O(n^{-1/2})$ rates in ∞-dimensional spaces, validated by empirical experiments.

Open Challenges & Future Directions

Inference for Non-Convexity: Extend stability, consistency, and CLT to non-convex objectives and other risk measures.

Conclusion



Key Scientific Achievements

Modeling → Algorithms → Implementation → Stability

PD-Risk: A robust primal-dual framework for **non-smooth risk-averse optimization** in ∞-dimensions.

Scalability via **adjoint states** and **parallelizable** gradient and Hess-vec computations.

Rigorous **quantitative stability bounds** using probability metrics like $\|z(Q) - z(P)\| \leq C d_{\mathcal{F}}(P, Q)$.

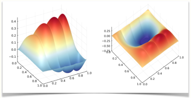
Asymptotic Consistency and $O(n^{-1/2})$ rates in ∞-dimensional spaces, validated by empirical experiments.

Open Challenges & Future Directions

Inference for Non-Convexity: Extend stability, consistency, and CLT to non-convex objectives and other risk measures.

General Constraints: Extend PD-Risk variants to treat additional bound constraints.

Conclusion



Key Scientific Achievements

Modeling $\to$ Algorithms $\to$ Implementation $\to$ Stability

PD-Risk: A robust primal-dual framework for **non-smooth risk-averse optimization** in ∞-dimensions.

Scalability via **adjoint states** and **parallelizable** gradient and Hess-vec computations.

Rigorous **quantitative stability bounds** using probability metrics like $\|z(Q) - z(P)\| \leq C d_{\mathcal{F}}(P, Q)$.

Asymptotic Consistency and $O(n^{-1/2})$ rates in ∞-dimensional spaces, validated by empirical experiments.

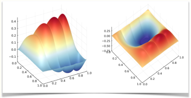
Open Challenges & Future Directions

Inference for Non-Convexity: Extend stability, consistency, and CLT to non-convex objectives and other risk measures.

General Constraints: Extend PD-Risk variants to treat additional bound constraints.

Time-Dependent Systems: Generalize theory and algorithms to time-dependent (stochastic control) problems(?)

Conclusion



Key Scientific Achievements

Modeling $\rightarrow$ Algorithms $\rightarrow$ Implementation $\rightarrow$ Stability

PD-Risk: A robust primal-dual framework for **non-smooth risk-averse optimization** in ∞-dimensions.

Scalability via **adjoint states** and **parallelizable** gradient and Hess-vec computations.

Rigorous **quantitative stability bounds** using probability metrics like $\|z(Q) - z(P)\| \leq C d_{\mathcal{F}}(P, Q)$.

Asymptotic Consistency and $O(n^{-1/2})$ rates in ∞-dimensional spaces, validated by empirical experiments.

Open Challenges & Future Directions

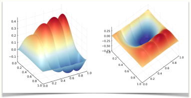
Inference for Non-Convexity: Extend stability, consistency, and CLT to non-convex objectives and other risk measures.

General Constraints: Extend PD-Risk variants to treat additional bound constraints.

Time-Dependent Systems: Generalize theory and algorithms to time-dependent (stochastic control) problems(?)

Adaptive Sampling or Sketching: Go beyond current work on risk-neutral case to treat non-smooth objectives, state constraints, etc.

Conclusion



Key Scientific Achievements

Modeling → Algorithms → Implementation → Stability

PD-Risk: A robust primal-dual framework for **non-smooth risk-averse optimization** in ∞-dimensions.

Scalability via **adjoint states** and **parallelizable** gradient and Hess-vec computations.

Rigorous **quantitative stability bounds** using probability metrics like $\|z(Q) - z(P)\| \leq C d_{\mathcal{F}}(P, Q)$.

Asymptotic Consistency and $O(n^{-1/2})$ rates in ∞-dimensional spaces, validated by empirical experiments.

Open Challenges & Future Directions

Inference for Non-Convexity: Extend stability, consistency, and CLT to non-convex objectives and other risk measures.

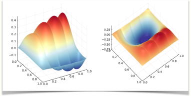
General Constraints: Extend PD-Risk variants to treat additional bound constraints.

Time-Dependent Systems: Generalize theory and algorithms to time-dependent (stochastic control) problems(?)

Adaptive Sampling or Sketching: Go beyond current work on risk-neutral case to treat non-smooth objectives, state constraints, etc.

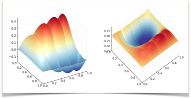
Thank You!

Details of the Implementation



Implemented in Julia using the FE package Gridap on Simula's eX3 computer.

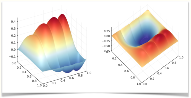
Details of the Implementation



Implemented in Julia using the FE package Gridap on Simula's eX3 computer.

State, adjoint, and sensitivity equations for separate samples ξ_i do not communicate.

Details of the Implementation

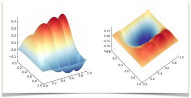


Implemented in Julia using the FE package Gridap on Simula's eX3 computer.

State, adjoint, and sensitivity equations for separate samples ξ_i do not communicate.

Semismooth Newton for optimization, direct solves for PDEs, CG algorithm for Newton steps.

Details of the Implementation



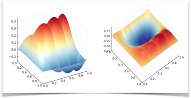
Implemented in Julia using the FE package Gridap on Simula's eX3 computer.

State, adjoint, and sensitivity equations for separate samples ξ_i do not communicate.

Semismooth Newton for optimization, direct solves for PDEs, CG algorithm for Newton steps.

No Hessian actually formed, only Hessian-vector products

Details of the Implementation



Implemented in Julia using the FE package Gridap on Simula's eX3 computer.

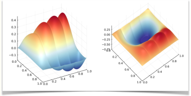
State, adjoint, and sensitivity equations for separate samples ξ_i do not communicate.

Semismooth Newton for optimization, direct solves for PDEs, CG algorithm for Newton steps.

No Hessian actually formed, only Hessian-vector products

Iterations reuse the n random stiffness matrices (state, adjoint, hess-vec); prefactor, cache.

Details of the Implementation



Implemented in Julia using the FE package Gridap on Simula's eX3 computer.

State, adjoint, and sensitivity equations for separate samples ξ_i do not communicate.

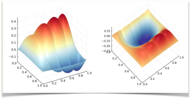
Semismooth Newton for optimization, direct solves for PDEs, CG algorithm for Newton steps.

No Hessian actually formed, only Hessian-vector products

Iterations reuse the n random stiffness matrices (state, adjoint, hess-vec); prefactor, cache.

Stopping criterion: absolute, relative tolerance of 10^{-8} for the discrete L^2-norm of residual.

Details of the Implementation



Implemented in Julia using the FE package Gridap on Simula's eX3 computer.

State, adjoint, and sensitivity equations for separate samples ξ_i do not communicate.

Semismooth Newton for optimization, direct solves for PDEs, CG algorithm for Newton steps.

No Hessian actually formed, only Hessian-vector products

Iterations reuse the n random stiffness matrices (state, adjoint, hess-vec); prefactor, cache.

Stopping criterion: absolute, relative tolerance of 10^{-8} for the discrete L^2-norm of residual.

Over $+$100K runs with random data, SSN converges in less than 10 iterations on average.