An Introduction to Risk-Averse PDE-Constrained Optimization: Theory, Numerical Solution, and Open Problems

Thomas M. Surowiec

Philipps-Universität Marburg
FB12 Mathematik und Informatik

CMAI Short Course 2021
Part II: Algorithms and Numerical Solution
June 18, 2021

DFG Deutsche Forschungsgemeinschaft

Philipps Universität Marburg

Overview of Part II

1. Aspects of Stochastic Optimization
   - Stochastic Optimization in ∞-Dimensions?
   - SGD Methods: Pros, Cons, and Behavior in Practice
   - EA Methods: Pros and Cons

2. Primal-Dual Risk Minimization
   - Motivation
   - The Algorithm
   - Convergence Theory

3. Implementation and Numerical Solution
   - Implementation Aspects
   - Numerical Examples
   - Summary and Outlook

Section 1:
Infinite-Dimensional Stochastic Optimization

Stochastic Optimization in ∞-Dimensions?

- We are interested in composite optimization problems of the type:

$$\min_{x \in X_{\mathrm{ad}}} f(x) + \Phi(F(x)) \text{ over } x \in X.$$

- X, Y are real **Hilbert spaces**.
- $f : X \to \mathbb{R}$ is typically continuous, convex, **differentiable**.
- $\Phi : Y \to \mathbb{R}$ is **nonsmooth**, convex, positive homogeneous, monotone.
- $F : X \to Y$ is continuous, typically differentiable, **expensive to evaluate**.

Stochastic Optimization in ∞-Dimensions?

- We are interested in composite optimization problems of the type:

$$\min_{x \in X_{\mathrm{ad}}} f(x) + \Phi(F(x)) \text{ over } x \in X.$$

- X, Y are real **Hilbert spaces**.
- $f : X \to \mathbb{R}$ is typically continuous, convex, **differentiable**.
- $\Phi : Y \to \mathbb{R}$ is **nonsmooth**, convex, positive homogeneous, monotone.
- $F : X \to Y$ is continuous, typically differentiable, **expensive to evaluate**.

- **"PDE"-part**: replace X by $\{X_h\}$, X_h finite dim. subspaces with elements $u_h(x) = \sum_{i=1}^{N} u_i^h \varphi_i^h(x)$.

Stochastic Optimization in ∞-Dimensions?

- We are interested in composite optimization problems of the type:

$$\min_{x \in X_{\mathrm{ad}}} f(x) + \Phi(F(x)) \text{ over } x \in X.$$

- X, Y are real **Hilbert spaces**.
- $f : X \to \mathbb{R}$ is typically continuous, convex, **differentiable**.
- $\Phi : Y \to \mathbb{R}$ is **nonsmooth**, convex, positive homogeneous, monotone.
- $F : X \to Y$ is continuous, typically differentiable, **expensive to evaluate**.

- **"PDE"-part**: replace X by $\{X_h\}$, X_h finite dim. subspaces with elements $u_h(x) = \sum_{i=1}^{N} u_i^h \varphi_i^h(x)$.
- **"PDE"-part**: $F(x)$ requires the solution of a PDE with input x.

Stochastic Optimization in ∞-Dimensions?

- We are interested in composite optimization problems of the type:

$$\min_{x \in X_{\mathrm{ad}}} f(x) + \Phi(F(x)) \text{ over } x \in X.$$

- X, Y are real **Hilbert spaces**.
- $f : X \to \mathbb{R}$ is typically continuous, convex, **differentiable**.
- $\Phi : Y \to \mathbb{R}$ is **nonsmooth**, convex, positive homogeneous, monotone.
- $F : X \to Y$ is continuous, typically differentiable, **expensive to evaluate**.

- **"PDE"-part**: replace X by $\{X_h\}$, X_h finite dim. subspaces with elements $u_h(x) = \sum_{i=1}^{N} u_i^h \varphi_i^h(x)$.
- **"PDE"-part**: $F(x)$ requires the solution of a PDE with input x.
- **"Random"-part**: cannot evaluate $F(x)$ or even $F(x_h)$ $x_h \in X_h$.

Stochastic Optimization in ∞-Dimensions?

- We are interested in composite optimization problems of the type:

$$\min_{x \in X_{\mathrm{ad}}} f(x) + \Phi(F(x)) \text{ over } x \in X.$$

- X, Y are real **Hilbert spaces**.
- $f : X \to \mathbb{R}$ is typically continuous, convex, **differentiable**.
- $\Phi : Y \to \mathbb{R}$ is **nonsmooth**, convex, positive homogeneous, monotone.
- $F : X \to Y$ is continuous, typically differentiable, **expensive to evaluate**.

- **"PDE"-part**: replace X by $\{X_h\}$, X_h finite dim. subspaces with elements $u_h(x) = \sum_{i=1}^N u_i^h \varphi_i^h(x)$.
- **"PDE"-part**: $F(x)$ requires the solution of a PDE with input x.
- **"Random"-part**: cannot evaluate $F(x)$ or even $F(x_h)$ $x_h \in X_h$.
- Derivatives are **dual** objects, gradients are **primal**.

Stochastic Optimization in ∞-Dimensions?

- We are interested in composite optimization problems of the type:

$$\min_{x \in X_{\mathrm{ad}}} f(x) + \Phi(F(x)) \text{ over } x \in X.$$

- X, Y are real **Hilbert spaces**.
- $f : X \to \mathbb{R}$ is typically continuous, convex, **differentiable**.
- $\Phi : Y \to \mathbb{R}$ is **nonsmooth**, convex, positive homogeneous, monotone.
- $F : X \to Y$ is continuous, typically differentiable, **expensive to evaluate**.

- **“PDE”-part**: replace X by $\{X_h\}$, X_h finite dim. subspaces with elements $u_h(x) = \sum_{i=1}^{N} u_i^h \varphi_i^h(x)$.
- **“PDE”-part**: $F(x)$ requires the solution of a PDE with input x.
- **“Random”-part**: cannot evaluate $F(x)$ or even $F(x_h)$ $x_h \in X_h$.
- Derivatives are **dual** objects, gradients are **primal**.
- If we want to do $x_k - \gamma_k \nabla f(x_k)$ in an algorithm, we need to first compute $\nabla f(x) = \mathfrak{R}_{\mathrm{Riesz}} f'(x)$.

Riesz Mappings

Example

- Let $D \subset \mathbb{R}^n$ be open and bounded. Let $A \subset D$ be Lebesgue measurable with $|A| > 0$.

Riesz Mappings

Example

- Let $D \subset \mathbb{R}^n$ be open and bounded. Let $A \subset D$ be Lebesgue measurable with $|A| > 0$.
- For $x \in D$, set $\varphi(x) = 1$ if $x \in A$ and 0 otherwise.

Riesz Mappings

Example

- Let $D \subset \mathbb{R}^n$ be open and bounded. Let $A \subset D$ be Lebesgue measurable with $|A| > 0$.
- For $x \in D$, set $\varphi(x) = 1$ if $x \in A$ and 0 otherwise.
- Since $H_0^1(D) \subset L^2(D)$ and φ is bounded and measurable, we see that

$$H_0^1(D) \ni u \mapsto \int_D \varphi(x)u(x)\,\mathrm{d}x$$

defines a bounded linear functional on $H_0^1(D)$.

Riesz Mappings

Example

- Let $D \subset \mathbb{R}^n$ be open and bounded. Let $A \subset D$ be Lebesgue measurable with $|A| > 0$.
- For $x \in D$, set $\varphi(x) = 1$ if $x \in A$ and 0 otherwise.
- Since $H_0^1(D) \subset L^2(D)$ and φ is bounded and measurable, we see that

  $$H_0^1(D) \ni u \mapsto \int_D \varphi(x) u(x) \,\mathrm{d}x$$

  defines a bounded linear functional on $H_0^1(D)$.
- We can therefore identity φ with an element of $H^{-1}(D) = (H_0^1(D))^*$.

Riesz Mappings

Example

- Let $D \subset \mathbb{R}^n$ be open and bounded. Let $A \subset D$ be Lebesgue measurable with $|A| > 0$.
- For $x \in D$, set $\varphi(x) = 1$ if $x \in A$ and 0 otherwise.
- Since $H_0^1(D) \subset L^2(D)$ and φ is bounded and measurable, we see that

  $$H_0^1(D) \ni u \mapsto \int_D \varphi(x)u(x)\,\mathrm{d}x$$

  defines a bounded linear functional on $H_0^1(D)$.
- We can therefore identity φ with an element of $H^{-1}(D) = (H_0^1(D))^*$.
- It's tempting to make the fallacious argument:

  > *$H_0^1(D)$ is a Hilbert space, I can just identify it with its dual and treat elements in $H^{-1}(D)$ like they are in $H_0^1(D)$.*

Riesz Mappings

Example

- Let $D \subset \mathbb{R}^n$ be open and bounded. Let $A \subset D$ be Lebesgue measurable with $|A| > 0$.
- For $x \in D$, set $\varphi(x) = 1$ if $x \in A$ and 0 otherwise.
- Since $H_0^1(D) \subset L^2(D)$ and φ is bounded and measurable, we see that

$$H_0^1(D) \ni u \mapsto \int_D \varphi(x)u(x)\,\mathrm{d}x$$

  defines a bounded linear functional on $H_0^1(D)$.
- We can therefore identity φ with an element of $H^{-1}(D) = (H_0^1(D))^*$.
- It's tempting to make the fallacious argument:

> *$H_0^1(D)$ is a Hilbert space, I can just identify it with its dual and treat elements in $H^{-1}(D)$ like they are in $H_0^1(D)$.*

- But φ will have **jumps** in general, so $\varphi \notin H_0^1(D)$!

Riesz Mappings

Example

- The **Riesz Representation Theorem** states: Let H be a real Hilbert space and $\varphi \in H^*$ (the topological dual of H). Then there exists a unique element $u_\varphi \in H$ such that

$$\langle \varphi, v \rangle_{H^*,H} = (u_\varphi, v)_H \quad \forall v \in H \text{ and } \|\varphi\|_{H^*} = \|u_\varphi\|_H$$

Riesz Mappings

Example

- The **Riesz Representation Theorem** states: Let H be a real Hilbert space and $\varphi \in H^*$ (the topological dual of H). Then there exists a unique element $u_\varphi \in H$ such that

$$\langle \varphi, v \rangle_{H^*,H} = (u_\varphi, v)_H \quad \forall v \in H \text{ and } \|\varphi\|_{H^*} = \|u_\varphi\|_H$$

- ...so if φ on p. 5 is in $H^{-1}(D)$, then there exists a unique $u_\varphi \in H_0^1(D)$ such that

$$(u_\varphi, v)_{H_0^1(D)} = \langle \varphi, v \rangle_{H^{-1}(D),H_0^1(D)} \quad \forall v \in H_0^1(D).$$

You promised to talk about algorithms and numerical methods:
What does this mean for my computations?

Riesz Mappings

Example

- The **Riesz Representation Theorem** states: Let H be a real Hilbert space and $\varphi \in H^*$ (the topological dual of H). Then there exists a unique element $u_\varphi \in H$ such that

$$\langle \varphi, v \rangle_{H^*,H} = (u_\varphi, v)_H \quad \forall v \in H \text{ and } \|\varphi\|_{H^*} = \|u_\varphi\|_H$$

- ...so if φ on p. 5 is in $H^{-1}(D)$, then there exists a unique $u_\varphi \in H^1_0(D)$ such that

$$(u_\varphi, v)_{H^1_0(D)} = \langle \varphi, v \rangle_{H^{-1}(D),H^1_0(D)} \quad \forall v \in H^1_0(D).$$

You promised to talk about algorithms and numerical methods:
What does this mean for my computations?

- Since $(u_\varphi, v)_{H^1_0(D)} = (\nabla u_\varphi, \nabla v)_{L^2(D)}$ $\forall v \in H^1_0(D)$, we have

$$u_\varphi = (-\Delta)^{-1}\varphi,$$

i.e., $\mathfrak{R}_{\mathrm{Riesz}}$ requires the solution of Poisson problem in $H^1_0(D)$.

Using the Correct Discrete Gradients

Example

- $f : L^2(D) \to \mathbb{R}$ is defined via continuous bilinear form.
- $X_h \subset L^2(D)$ is a finite dimensional subspace defined by the nodal basis arising from a finite element discretization.
- $u_h \in X_h$ is associated with coefficient vector $\mathbf{u}_h \in \mathbb{R}^n$

Using the Correct Discrete Gradients

Example

- $f : L^2(D) \to \mathbb{R}$ is defined via continuous bilinear form.
- $X_h \subset L^2(D)$ is a finite dimensional subspace defined by the nodal basis arising from a finite element discretization.
- $u_h \in X_h$ is associated with coefficient vector $\mathbf{u}_h \in \mathbb{R}^n$
- $f_h : \mathbb{R}^n \to \mathbb{R}$ is the FE discretization of f, e.g.

$$f_h(\mathbf{u}_h) = \frac{1}{2}\mathbf{u}_h^T L_h \mathbf{u}_h$$

- The correct gradient for a numerical approach would then be

$$\mathfrak{R}_{L^2}(f_h'(\mathbf{u}_h)) = M_h^{-1} L_h \mathbf{u}_h.$$

  M_h is the mass matrix associated with the discrete L^2-inner product.
- If $f(u) = (u, u)_{L^2(D)}$, then $\mathfrak{R}_{L^2}(f_h'(\mathbf{u}_h))$ is just the **vector of nodal values $\boldsymbol{u}_h$**, not $M_h \boldsymbol{u}_h$!

Using the Correct Discrete Gradients

Example

- Suppose $f : L^2(D) \to \mathbb{R}$ is defined

$$f(u) := \frac{1}{2} \int_\Omega (L(\xi(\omega))u, u)_{L^2(D)} \,\mathrm{d}\mathbb{P}(\omega),$$

where $\xi : (\Omega, \mathcal{F}, \mathbb{P}) \to \mathbb{R}^d$ is a random vector.

Using the Correct Discrete Gradients

Example

- Suppose $f : L^2(D) \to \mathbb{R}$ is defined

$$f(u) := \frac{1}{2} \int_\Omega (L(\xi(\omega))u, u)_{L^2(D)} \,\mathrm{d}\mathbb{P}(\omega),$$

where $\xi : (\Omega, \mathcal{F}, \mathbb{P}) \to \mathbb{R}^d$ is a random vector.

- Given an iid sample $\xi^1, \ldots, \xi^N$ we would use

$$f_N(u) := \frac{1}{2N} \sum_{i=1}^{N} (L(\xi^i)u, u)_{L^2(D)}$$

in **empirical approximations** of $\mathbb{P}$.

Using the Correct Discrete Gradients

Example

- Suppose $f : L^2(D) \to \mathbb{R}$ is defined

$$f(u) := \frac{1}{2}\int_\Omega (L(\xi(\omega))u, u)_{L^2(D)}\,\mathrm{d}\mathbb{P}(\omega),$$

where $\xi : (\Omega, \mathcal{F}, \mathbb{P}) \to \mathbb{R}^d$ is a random vector.

- Given an iid sample $\xi^1, \dots, \xi^N$ we would use

$$f_N(u) := \frac{1}{2N}\sum_{i=1}^{N} (L(\xi^i)u, u)_{L^2(D)}$$

in **empirical approximations** of $\mathbb{P}$.

- The correct fully discrete gradient would then be

$$\mathfrak{R}_{L^2}(f'_{N,h}(\mathbf{u}_h)) = \frac{1}{N}\sum_{i=1}^{N} M_h^{-1} L_h^i \mathbf{u}_h,$$

where $\mathbf{u}_h^T L_h^i \mathbf{u}_h = (L(\xi^i)u_h, u_h)_{L^2(D)}$.

Using the Correct Discrete Gradients

- A fully discrete problem would take the form:

$$\min_{x \in X^h_{\mathrm{ad}}} f_h(x_h) + \Phi_{h,N}(F_{h,N}(x_h)) \text{ over } x_h \in X_h. \tag{1}$$

- h denotes dependence on mesh or scale, N is the sample size.
- However, solvers based solely on n-dim algorithms applied to (1) are usually **mesh dependent**.
- This can be a result either of using
  - $\mathfrak{R}_{\mathrm{Riesz}} = \mathrm{Id}_{\mathbb{R}^n}$ or
  - Not actually having the required differentiability properties in the fully continuous setting.
- The **former** means we have poor scaling and require more iterations than necessary.
- The latter means the **discrete derivatives** (gradients, Hessians) do not correspond to an infinite dimensional counterpart.

Using the Correct Discrete Gradients

Example (Scaling)

- Clearly $u \equiv 0 \in L^2(D)$ minimizes $f(u) = \frac{1}{2}\|u\|^2_{L^2(D)}$.
- Likewise $\boldsymbol{u}_h = 0 \in \mathbb{R}^n$ minimizes $f_h(\boldsymbol{u}_h) = \frac{1}{2}\boldsymbol{u}_h^T M_h \boldsymbol{u}_h$.
- Using the correct discrete gradients to minimize f_h with a simple steepest descent approach yields

$$\boldsymbol{u}_{h,k+1} = (1 - \alpha_k)\boldsymbol{u}_{u,k}.$$

- This involves a strongly convex quadratic objective
- We can determine α_k by an exact line search.
- This (regardless of u_0) yields $\alpha_1 = 1$! The algorithm will stop in one iteration.
- If the **incorrect discrete gradient** is used, then this will not be the case. Try it at home.

Basic SGD-Type Algorithm

Preliminary Remarks

- **SGD**: **S**tochastic **G**radient **D**escent, also **SA**: **S**tochastic **A**pproximation.

Basic SGD-Type Algorithm

Preliminary Remarks

- **SGD**: **S**tochastic **G**radient **D**escent, also **SA**: **S**tochastic **A**pproximation.
- The problems of interest typically do not arise in a **machine learning** context.
- We do not train **prediction functions**, rather we want **robust solutions** to **engineering problems**.
- In **machine learning**, one commonly held belief is

  By "undercomputing" we avoid "overfitting".
  -T. Dietterich in "*Overfitting and Undercomputing in Machine Learning*" (1995).
- Thus, slower **first-order methods** that are **stopped prematurely** make sense in that framework.
- If your model is tenuous at best and your data is noisy, then it's probably not a good idea to solve to high accuracy and expect good generalization out-of-sample.

Basic SGD-Type Algorithm

- We consider the stochastic optimization problem

$$\min_{x \in X} \left\{ f(x) := \mathbb{E}_{\mathbb{P}}[F(x, \cdot)] \right\}.$$

- X is nonempty, closed, and convex.

Basic SGD-Type Algorithm

- We consider the stochastic optimization problem

$$\min_{x \in X} \left\{ f(x) := \mathbb{E}_{\mathbb{P}}[F(x, \cdot)] \right\}.$$

- X is nonempty, closed, and convex.
- $F : X \times \Omega \to \mathbb{R}$ is convex and continuous in x for each ω and measurable in ω for each x.

Basic SGD-Type Algorithm

- We consider the stochastic optimization problem

$$\min_{x\in X} \{f(x) := \mathbb{E}_{\mathbb{P}}[F(x,\cdot)]\}.$$

- X is nonempty, closed, and convex.
- $F : X \times \Omega \to \mathbb{R}$ is convex and continuous in x for each ω and measurable in ω for each x.
- $f : X \to \mathbb{R}$ is finite and continuous.

Basic SGD-Type Algorithm

- We consider the stochastic optimization problem

$$\min_{x \in X} \left\{ f(x) := \mathbb{E}_{\mathbb{P}}[F(x, \cdot)] \right\}.$$

- X is nonempty, closed, and convex.
- $F : X \times \Omega \to \mathbb{R}$ is convex and continuous in x for each ω and measurable in ω for each x.
- $f : X \to \mathbb{R}$ is finite and continuous.
- Finite-dimensional noise: $F(x, \omega) = \widehat{F}(x, \xi(\omega))$ where $\xi : \Omega \to \mathbb{R}^d$ is a random vector.

Basic SGD-Type Algorithm

- We consider the stochastic optimization problem

$$\min_{x\in X}\left\{f(x) := \mathbb{E}_{\mathbb{P}}[F(x,\cdot)]\right\}.$$

- X is nonempty, closed, and convex.
- $F : X \times \Omega \to \mathbb{R}$ is convex and continuous in x for each ω and measurable in ω for each x.
- $f : X \to \mathbb{R}$ is finite and continuous.
- Finite-dimensional noise: $F(x,\omega) = \widehat{F}(x,\xi(\omega))$ where $\xi : \Omega \to \mathbb{R}^d$ is a random vector.
- We can generate an iid sample $\xi_1, \xi_2, \ldots$ in $\Xi \subset \mathbb{R}^d$. We now use the probability law $\mathbb{P}_{\boldsymbol{\xi}} := \mathbb{P} \circ \boldsymbol{\xi}^{-1}$.

Basic SGD-Type Algorithm

- We consider the stochastic optimization problem

$$\min_{x \in X} \{f(x) := \mathbb{E}_{\mathbb{P}}[F(x, \cdot)]\}.$$

- X is nonempty, closed, and convex.
- $F : X \times \Omega \to \mathbb{R}$ is convex and continuous in x for each ω and measurable in ω for each x.
- $f : X \to \mathbb{R}$ is finite and continuous.
- Finite-dimensional noise: $F(x, \omega) = \widehat{F}(x, \xi(\omega))$ where $\xi : \Omega \to \mathbb{R}^d$ is a random vector.
- We can generate an iid sample $\xi_1, \xi_2, \ldots$ in $\Xi \subset \mathbb{R}^d$. We now use the probability law $\mathbb{P}_{\boldsymbol{\xi}} := \mathbb{P} \circ \boldsymbol{\xi}^{-1}$.
- We can generate a stochastic subgradient $G(x, \xi) \in \partial_x F(x, \xi)$ for $\mathbb{P}_{\boldsymbol{\xi}}$-a.e. $\xi \in \Xi$.

Basic SGD-Type Algorithm

- We consider the stochastic optimization problem

$$\min_{x \in X} \left\{ f(x) := \mathbb{E}_{\mathbb{P}}[F(x,\cdot)] \right\}.$$

- X is nonempty, closed, and convex.
- $F : X \times \Omega \to \mathbb{R}$ is convex and continuous in x for each ω and measurable in ω for each x.
- $f : X \to \mathbb{R}$ is finite and continuous.
- Finite-dimensional noise: $F(x,\omega) = \widehat{F}(x,\xi(\omega))$ where $\xi : \Omega \to \mathbb{R}^d$ is a random vector.
- We can generate an iid sample $\xi_1, \xi_2, \ldots$ in $\Xi \subset \mathbb{R}^d$. We now use the probability law $\mathbb{P}_{\boldsymbol{\xi}} := \mathbb{P} \circ \boldsymbol{\xi}^{-1}$.
- We can generate a stochastic subgradient $G(x,\xi) \in \partial_x F(x,\xi)$ for $\mathbb{P}_{\boldsymbol{\xi}}$-a.e. $\xi \in \Xi$.
- $\mathbb{E}[G(x,\xi_t)] \in \partial f(x)$ for $t = 1, 2, \ldots$.

Basic SGD-Type Algorithm

- We consider the stochastic optimization problem

$$\min_{x \in X} \{ f(x) := \mathbb{E}_{\mathbb{P}}[F(x, \cdot)] \} .$$

- X is nonempty, closed, and convex.
- $F : X \times \Omega \to \mathbb{R}$ is convex and continuous in x for each ω and measurable in ω for each x.
- $f : X \to \mathbb{R}$ is finite and continuous.
- Finite-dimensional noise: $F(x, \omega) = \widehat{F}(x, \xi(\omega))$ where $\xi : \Omega \to \mathbb{R}^d$ is a random vector.
- We can generate an iid sample $\xi_1, \xi_2, \dots$ in $\Xi \subset \mathbb{R}^d$. We now use the probability law $\mathbb{P}_{\boldsymbol{\xi}} := \mathbb{P} \circ \boldsymbol{\xi}^{-1}$.
- We can generate a stochastic subgradient $G(x, \xi) \in \partial_x F(x, \xi)$ for $\mathbb{P}_{\boldsymbol{\xi}}$-a.e. $\xi \in \Xi$.
- $\mathbb{E}[G(x, \xi_t)] \in \partial f(x)$ for $t = 1, 2, \dots$.
- There is some $\sigma \in \mathbb{R}$ for all $x \in X$ such that $\mathbb{E}_{\mathbb{P}_{\boldsymbol{\xi}}}[\| G(x, \xi) - f'(x) \|_{X^*}^2] \leq \sigma$ and $f'(x) \in \partial f(x)$.

Basic SGD-Type Algorithm

- We consider the stochastic optimization problem

$$\min_{x\in X} \{ f(x) := \mathbb{E}_{\mathbb{P}}[F(x,\cdot)] \} .$$

- X is nonempty, closed, and convex.
- $F : X \times \Omega \to \mathbb{R}$ is convex and continuous in x for each ω and measurable in ω for each x.
- $f : X \to \mathbb{R}$ is finite and continuous.
- Finite-dimensional noise: $F(x,\omega) = \widehat{F}(x,\xi(\omega))$ where $\xi : \Omega \to \mathbb{R}^d$ is a random vector.
- We can generate an iid sample $\xi_1, \xi_2, \ldots$ in $\Xi \subset \mathbb{R}^d$. We now use the probability law $\mathbb{P}_{\boldsymbol{\xi}} := \mathbb{P} \circ \boldsymbol{\xi}^{-1}$.
- We can generate a stochastic subgradient $G(x,\xi) \in \partial_x F(x,\xi)$ for $\mathbb{P}_{\boldsymbol{\xi}}$-a.e. $\xi \in \Xi$.
- $\mathbb{E}[G(x,\xi_t)] \in \partial f(x)$ for $t = 1, 2, \ldots$.
- There is some $\sigma \in \mathbb{R}$ for all $x \in X$ such that $\mathbb{E}_{\mathbb{P}_{\boldsymbol{\xi}}}[\|G(x,\xi) - f'(x)\|^2_{X^*}] \leq \sigma$ and $f'(x) \in \partial f(x)$.

Basic SGD Iteration

Given $x_0 \in X$ and $\{\gamma_k\}$ with $\gamma_k > 0$, define $\{x_k\}$ such that $x_{k+1} = \mathrm{Proj}_X(x_k - \gamma_k G(x_k, \xi_k))$

Basic SGD Algorithm for the Canonical Example

- Recall our canonical example from Part I (p. 52).

$$\min\left\{f(z) := \mathrm{CVaR}_\beta\left[\frac{1}{2}\int_D |S(z)-u_d|^2 \mathrm{d}x\right] + \frac{\alpha}{2}\|z\|_Z^2 \text{ over } z \in \mathcal{Z}_{\text{ad}}\right\}, \tag{2}$$

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

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

- We also recall the variational formulation of CVaR_β:

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

Basic SGD Algorithm for the Canonical Example

- This leads to the "expanded" reduced problem:

$$\min\left\{\mathbb{E}_{\mathbb{P}_\xi}\left[t+\frac{1}{2-2\beta}(\|S(z)-u_d\|^2_{L^2(D)}-2t)_+ +\frac{\alpha}{2}\|z\|^2_Z\right] \text{ over } (z,t)\in\mathcal{Z}_{\text{ad}}\times\mathbb{R}\right\}. \qquad (3)$$

  We then set

- $x=(z,t)$, $X=\mathcal{Z}_{\text{ad}}\times\mathbb{R}$
- $F(x,\xi)=t+\frac{1}{2-2\beta}(\|S(z,\xi)-u_d\|^2_{L^2(D)}-2t)_+ +\frac{\alpha}{2}\|z\|^2_Z$
- $f(x)=\mathbb{E}_{\mathbb{P}}[F(x,\cdot)]$.
- f is not strictly convex.

Pros

- Due to the standing assumptions, we can show that $G(x,\xi)$ can be split into a 2-tuple $G(x,\xi) = ([G(x,\xi)]_z, [G(x,\xi)]_t)$ with

$$
\begin{aligned}
[G(x,\xi)]_z &= \alpha z + (1-\beta)^{-1}(\chi_{\{F(z,\xi)>t\}} + q\chi_{\{F(z,\xi)=t\}})\boldsymbol{B}(\xi)^*\Lambda(z,\xi) \\
[G(x,\xi)]_t &= 1 - (1-\beta)^{-1}(\chi_{\{F(z,\xi)>t\}} + q\chi_{\{F(z,\xi)=t\}})
\end{aligned}
$$

Pros

- Due to the standing assumptions, we can show that $G(x,\xi)$ can be split into a 2-tuple $G(x,\xi) = ([G(x,\xi)]_z, [G(x,\xi)]_t)$ with

$$[G(x,\xi)]_z = \alpha z + (1-\beta)^{-1}(\chi_{\{F(z,\xi)>t\}} + q\chi_{\{F(z,\xi)=t\}})\boldsymbol{B}(\xi)^*\Lambda(z,\xi)$$
$$[G(x,\xi)]_t = 1 - (1-\beta)^{-1}(\chi_{\{F(z,\xi)>t\}} + q\chi_{\{F(z,\xi)=t\}})$$

where $q \in [0,1]$, e.g. $q = 0$, $\lambda = \Lambda(z,\xi)$, and $u(\xi) = S(z,\xi)$ such that

$$\boldsymbol{A}(\xi)^*\lambda = u(\xi) - u_d \text{ in } H^{-1}(D), \qquad \boldsymbol{A}(\xi)u = \boldsymbol{B}(\xi)z + f(\xi) \text{ in } H^{-1}(D).$$

Pros

- Due to the standing assumptions, we can show that $G(x,\xi)$ can be split into a 2-tuple $G(x,\xi) = ([G(x,\xi)]_z, [G(x,\xi)]_t)$ with

$$[G(x,\xi)]_z = \alpha z + (1-\beta)^{-1}(\chi_{\{F(z,\xi)>t\}} + q\chi_{\{F(z,\xi)=t\}})\boldsymbol{B}(\xi)^* \Lambda(z,\xi)$$
$$[G(x,\xi)]_t = 1 - (1-\beta)^{-1}(\chi_{\{F(z,\xi)>t\}} + q\chi_{\{F(z,\xi)=t\}})$$

where $q \in [0,1]$, e.g. $q = 0$, $\lambda = \Lambda(z,\xi)$, and $u(\xi) = S(z,\xi)$ such that

$$\boldsymbol{A}(\xi)^*\lambda = u(\xi) - u_d \text{ in } H^{-1}(D), \qquad \boldsymbol{A}(\xi)u = \boldsymbol{B}(\xi)z + f(\xi) \text{ in } H^{-1}(D).$$

- **Pro**: 1 **Forward Solve**, ≤ 1 **Adjoint Solve** for each SGD-iteration:
  - Given (z_k, t_k), draw a sample $\xi_k \in \Xi$.

Pros

- Due to the standing assumptions, we can show that $G(x,\xi)$ can be split into a 2-tuple $G(x,\xi) = ([G(x,\xi)]_z, [G(x,\xi)]_t)$ with

$$\begin{aligned} [G(x,\xi)]_z &= \alpha z + (1-\beta)^{-1}(\chi_{\{F(z,\xi)>t\}} + q\chi_{\{F(z,\xi)=t\}})\boldsymbol{B}(\xi)^*\Lambda(z,\xi) \\ [G(x,\xi)]_t &= 1 - (1-\beta)^{-1}(\chi_{\{F(z,\xi)>t\}} + q\chi_{\{F(z,\xi)=t\}}) \end{aligned}$$

  where $q \in [0,1]$, e.g. $q = 0$, $\lambda = \Lambda(z,\xi)$, and $u(\xi) = S(z,\xi)$ such that

$$\boldsymbol{A}(\xi)^*\lambda = u(\xi) - u_d \text{ in } H^{-1}(D), \qquad \boldsymbol{A}(\xi)u = \boldsymbol{B}(\xi)z + f(\xi) \text{ in } H^{-1}(D).$$

- **Pro**: 1 **Forward Solve**, $\leq$ 1 **Adjoint Solve** for each SGD-iteration:
  - Given (z_k, t_k), draw a sample $\xi_k \in \Xi$.
  - Solve for $u(\xi_k) = S(z_k, \xi_k)$ (in discrete setting: sparse structured linear system)

Pros

- Due to the standing assumptions, we can show that $G(x,\xi)$ can be split into a 2-tuple $G(x,\xi) = ([G(x,\xi)]_z, [G(x,\xi)]_t)$ with

$$[G(x,\xi)]_z = \alpha z + (1-\beta)^{-1}(\chi_{\{F(z,\xi)>t\}} + q\chi_{\{F(z,\xi)=t\}})\boldsymbol{B}(\xi)^* \Lambda(z,\xi)$$
$$[G(x,\xi)]_t = 1 - (1-\beta)^{-1}(\chi_{\{F(z,\xi)>t\}} + q\chi_{\{F(z,\xi)=t\}})$$

where $q \in [0,1]$, e.g. $q = 0$, $\lambda = \Lambda(z,\xi)$, and $u(\xi) = S(z,\xi)$ such that

$$\boldsymbol{A}(\xi)^*\lambda = u(\xi) - u_d \text{ in } H^{-1}(D), \qquad \boldsymbol{A}(\xi)u = \boldsymbol{B}(\xi)z + f(\xi) \text{ in } H^{-1}(D).$$

- **Pro**: 1 **Forward Solve**, $\leq$ 1 **Adjoint Solve** for each SGD-iteration:
  - Given (z_k, t_k), draw a sample $\xi_k \in \Xi$.
  - Solve for $u(\xi_k) = S(z_k, \xi_k)$ (in discrete setting: sparse structured linear system)
  - If $F(z_k, \xi_k) > t_k$, solve for $\Lambda(z_k, \xi_k)$ (in discrete setting: sparse structured linear system)

Pros

- Due to the standing assumptions, we can show that $G(x,\xi)$ can be split into a 2-tuple $G(x,\xi) = ([G(x,\xi)]_z, [G(x,\xi)]_t)$ with

$$[G(x,\xi)]_z = \alpha z + (1-\beta)^{-1}(\chi_{\{F(z,\xi)>t\}} + q\chi_{\{F(z,\xi)=t\}})\boldsymbol{B}(\xi)^*\Lambda(z,\xi)$$
$$[G(x,\xi)]_t = 1 - (1-\beta)^{-1}(\chi_{\{F(z,\xi)>t\}} + q\chi_{\{F(z,\xi)=t\}})$$

where $q \in [0,1]$, e.g. $q = 0$, $\lambda = \Lambda(z,\xi)$, and $u(\xi) = S(z,\xi)$ such that

$$\boldsymbol{A}(\xi)^*\lambda = u(\xi) - u_d \text{ in } H^{-1}(D), \qquad \boldsymbol{A}(\xi)u = \boldsymbol{B}(\xi)z + f(\xi) \text{ in } H^{-1}(D).$$

- **Pro**: 1 **Forward Solve**, $\leq$ 1 **Adjoint Solve** for each SGD-iteration:
  - Given (z_k, t_k), draw a sample $\xi_k \in \Xi$.
  - Solve for $u(\xi_k) = S(z_k, \xi_k)$ (in discrete setting: sparse structured linear system)
  - If $F(z_k, \xi_k) > t_k$, solve for $\Lambda(z_k, \xi_k)$ (in discrete setting: sparse structured linear system)
  - Else set $q_k = 0$
  - Construct $G(x_k, \xi_k)$ go to projection step.

Pros

- Let $Z = L^2(D)$, $a, b \in \mathbb{R}$ such that $a < b$ and

$$\mathcal{Z}_{\text{ad}} := \{z \in Z \,|\, a \leq z \leq b \text{ a.e. } D\}$$

Pros

- Let $Z = L^2(D)$, $a, b \in \mathbb{R}$ such that $a < b$ and
$$\mathcal{Z}_{\text{ad}} := \{z \in Z \,|\, a \leq z \leq b \text{ a.e. } D\}$$
- **Pro**: Projection step for pointwise bilateral constraints is cheap:
  - $t_{k+1} = t_k - \gamma_k [G(x_k, \xi_k)]_t$
  - $z_{k+1} = z_k - \gamma_k [G(x_k, \xi_k)]_z - (z_k - \gamma_k [G(x_k, \xi_k)]_z - b)_+ + (a - z_k + \gamma_k [G(x_k, \xi_k)]_z)_+$

Pros

- Let $Z = L^2(D)$, $a, b \in \mathbb{R}$ such that $a < b$ and

$$\mathcal{Z}_{\text{ad}} := \{z \in Z \,|\, a \leq z \leq b \text{ a.e. } D\}$$

- **Pro**: Projection step for pointwise bilateral constraints is cheap:
  - $t_{k+1} = t_k - \gamma_k[G(x_k, \xi_k)]_t$
  - $z_{k+1} = z_k - \gamma_k[G(x_k, \xi_k)]_z - (z_k - \gamma_k[G(x_k, \xi_k)]_z - b)_+ + (a - z_k + \gamma_k[G(x_k, \xi_k)]_z)_+$
- The projection is usually done at the nodal values in the discrete setting if, e.g., piecewise constant or piecewise lineare FE are used.

Pros

- Let $Z = L^2(D)$, $a, b \in \mathbb{R}$ such that $a < b$ and

$$\mathcal{Z}_{\text{ad}} := \{z \in Z \,|a \leq z \leq b \text{ a.e. } D\}$$

- **Pro**: Projection step for pointwise bilateral constraints is cheap:
  - $t_{k+1} = t_k - \gamma_k[G(x_k, \xi_k)]_t$
  - $z_{k+1} = z_k - \gamma_k[G(x_k, \xi_k)]_z - (z_k - \gamma_k[G(x_k, \xi_k)]_z - b)_+ + (a - z_k + \gamma_k[G(x_k, \xi_k)]_z)_+$
- The projection is usually done at the nodal values in the discrete setting if, e.g., piecewise constant or piecewise lineare FE are used.
- **Pro**: The **convergence theory** includes the **sampling** aspect.

Pros

- Let $Z = L^2(D)$, $a, b \in \mathbb{R}$ such that $a < b$ and

$$\mathcal{Z}_{\text{ad}} := \{z \in Z \,|a \leq z \leq b \text{ a.e. } D\}$$

- **Pro**: Projection step for pointwise bilateral constraints is cheap:
  - $t_{k+1} = t_k - \gamma_k[G(x_k, \xi_k)]_t$
  - $z_{k+1} = z_k - \gamma_k[G(x_k, \xi_k)]_z - (z_k - \gamma_k[G(x_k, \xi_k)]_z - b)_+ + (a - z_k + \gamma_k[G(x_k, \xi_k)]_z)_+$
- The projection is usually done at the nodal values in the discrete setting if, e.g., piecewise constant or piecewise lineare FE are used.
- **Pro**: The **convergence theory** includes the **sampling** aspect.
- We could have easily taken **batches** of $G(x_k, \xi_k^j)$ and used their weighted averages at each step.

Pros

- Let $Z = L^2(D)$, $a, b \in \mathbb{R}$ such that $a < b$ and

$$\mathcal{Z}_{\text{ad}} := \{z \in Z \,|\, a \le z \le b \text{ a.e. } D\}$$

- **Pro**: Projection step for pointwise bilateral constraints is cheap:
  - $t_{k+1} = t_k - \gamma_k [G(x_k, \xi_k)]_t$
  - $z_{k+1} = z_k - \gamma_k [G(x_k, \xi_k)]_z - (z_k - \gamma_k [G(x_k, \xi_k)]_z - b)_+ + (a - z_k + \gamma_k [G(x_k, \xi_k)]_z)_+$
- The projection is usually done at the nodal values in the discrete setting if, e.g., piecewise constant or piecewise lineare FE are used.
- **Pro**: The **convergence theory** includes the **sampling** aspect.
- We could have easily taken **batches** of $G(x_k, \xi_k^j)$ and used their weighted averages at each step.
- The basic theory indicates that we should choose diminishing sizes, e.g. $1/k$.
- Our practical experience yielded poor results. The fixed step $\gamma_k = \gamma = 10$ worked best.
- Robust SA was invented to address this issue.
- The theory tells us that we should take $\gamma_k = O(\frac{1}{\sqrt{k}})$, but...

Cons

Stepsizes

- ...some O's are bigger than others: It is usually assumed that the feasible set has a finite diameter D_X.
- In our case, there are no known bounds on t.

Cons

Stepsizes

- ...some O's are bigger than others: It is usually assumed that the feasible set has a finite diameter D_X.
- In our case, there are no known bounds on t.
- Back to CVaR_β:

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

- Assume for discussion $X \in L^1(\Omega, \mathcal{F}, \mathbb{P})$ has a continuous cdf.
- Then the minimizing $t^\star$ is the left-side β-quantile of F_X, i.e., $t^\star = F_X^{-1}(1-\beta)$.
- Since our objective integrand $J \circ S$ is nonnegative, we could put an artificial lower bound on all t.
- We do not know a priori what the maximum value of $(J \circ S)(z)$ for $z \in \mathcal{Z}_{\mathrm{ad}}$ is. We could estimate it, but that would costs thousands of additional PDE solves.

Cons

Stepsizes

- ...some O's are bigger than others: It is usually assumed that the feasible set has a finite diameter D_X.
- In our case, there are no known bounds on t.
- Back to CVaR_β:
$$\mathrm{CVaR}_\beta[X] = \inf_{t\in\mathbb{R}} \left\{ t + \frac{1}{1-\beta}\mathbb{E}[(X-t)_+] \right\}$$
- Assume for discussion $X \in L^1(\Omega, \mathcal{F}, \mathbb{P})$ has a continuous cdf.
- Then the minimizing $t^\star$ is the left-side β-quantile of F_X, i.e., $t^\star = F_X^{-1}(1-\beta)$.
- Since our objective integrand $J \circ S$ is nonnegative, we could put an artificial lower bound on all t.
- We do not know a priori what the maximum value of $(J \circ S)(z)$ for $z \in \mathcal{Z}_{\mathrm{ad}}$ is. We could estimate it, but that would costs thousands of additional PDE solves.
- O also requires knowledge of σ (p. 12) as well as a constant M such that
$$\|\mathbb{E}_{\mathbb{P}_\xi}[G(x,\xi)]\| \le M \quad \forall x \in \mathcal{Z}_{\mathrm{ad}} \times \mathbb{R}.$$

Cons

Stopping and Behavior of the Iterates

- When should we stop?
- The theory provides an idea about the mean behavior of the **objective function values** in relation to the optimal value for the **aggregated mean** of iterates $\bar{x}_k = \frac{1}{k}\sum_{i=1}^{k} x_i$.
- These would say: Stop after k iterations, expect $\mathbb{E}_{\mathbb{P}_\xi}[f(\bar{x}_k)]$ to be within $O(\frac{1}{\sqrt{k}})$ of $f(x^\star)$.
- There are upper bounds for computable accuracy certificates. These typically contain σ, D_X, M.
- **But** we are interested in the **iterates** x_k, not necessarily $f(x_k)$.
- We do not have $f(x_k)$, nor $\widehat{f}_N(x_k) = \frac{1}{N}\sum_{i=1}^{N} F(x_k, \xi^i)$ with N large to check the behavior of $f(x_k)$
- We do not have $\nabla f(x_k)$ nor $\nabla \widehat{f}_N(x_k) = \frac{1}{N}\sum_{i=1}^{N} \nabla F(x_k, \xi^i)$ to check first order system.
- These would require **many** more **forward** and **adjoint solves**.

How slow is it really? An example.

- In this basic SGD approach, no **second-order** information is included.
- In a deterministic setting, we would **expect** potentially **slow convergence**.
- It is possible to add intermediate steps to get a better algorithm, e.g. stochastic accelerated gradient descent method/stochastic accelerated approximation.
- But how slow could it really be?

How slow is it really? An example.

- In this basic SGD approach, no **second-order** information is included.
- In a deterministic setting, we would **expect** potentially **slow convergence**.
- It is possible to add intermediate steps to get a better algorithm, e.g. stochastic accelerated gradient descent method/stochastic accelerated approximation.
- But how slow could it really be?

Example

- Assume $D \subset \mathbb{R}^2$ is the unit square. PDE from contaminant mitigation problem.
- Discretize the control and state (Q1-FEM) spaces on a uniform 32 x 32 grid. (coarse!)
- Solve the problem using MC (10000 samples) and the PD-Risk algorithm (Section 2) up to a tolerance of 1e-7 for the discrete-L^2-norm of the first-order optimality system.
- Use this as the “true” solution.
- More samples and finer grids possible, this is just for discussion.

How slow is it really? An example.

- $\gamma_k = c/\sqrt{k}$ $c = 1e4$. Different c values yield similar behavior.
- **iter**: number of SGD-iterations.
- **fval**: function value.
- **abs-err** f: absolute error of objective function values.
- **rel-err** f: relative error of objective function values.
- **abs-err** x_k: absolute error in discrete L^2-norm of iterate from "true" solution
- **rel-err** x_k: relative error in discrete L^2-norm of iterate from "true" solution.

How slow is it really? An example.

- $\gamma_k = c/\sqrt{k}$ $c = 1e4$. Different c values yield similar behavior.
- **iter**: number of SGD-iterations.
- **fval**: function value.
- **abs-err** f: absolute error of objective function values.
- **rel-err** f: relative error of objective function values.
- **abs-err** x_k: absolute error in discrete L^2-norm of iterate from "true" solution
- **rel-err** x_k: relative error in discrete L^2-norm of iterate from "true" solution.

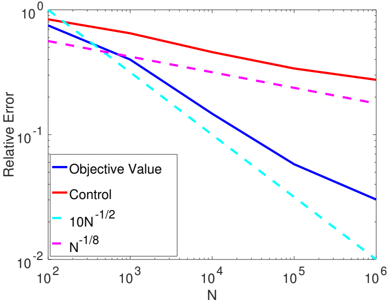
Example

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

Objective errors based on 100,000 samples that are different from those used for SAA and SA.

How slow is it really? An example.

After a million iterations and days of computing the **relative error** in the iterates is greater than 0.1.



Thus, hundreds of millions of PDE-solves are needed to get an **accurate solution**.

Empirical Approximation

- In our canonical example, we can expand the decision space by a real-valued decision variable t and consider an objective of the form

$$f(x) = \mathbb{E}_{\mathbb{P}_{\xi}}[F(x,\cdot)] = \int_{\Xi} F(x,\xi)\,\mathrm{d}\mathbb{P}(\xi)$$

  where F is nonsmooth and convex. Thus, the objective involves a high-dimensional integral.
- We use quadrature everywhere in PDE-constrained optimization, why not here?
- What are our options if $\dim \Xi >> 1$?

Empirical Approximation

- In our canonical example, we can expand the decision space by a real-valued decision variable t and consider an objective of the form

$$f(x) = \mathbb{E}_{\mathbb{P}_\xi}[F(x,\cdot)] = \int_{\Xi} F(x,\xi)\,\mathrm{d}\mathbb{P}(\xi)$$

  where F is nonsmooth and convex. Thus, the objective involves a high-dimensional integral.
- We use quadrature everywhere in PDE-constrained optimization, why not here?
- What are our options if $\dim \Xi >> 1$?

- EA = **E**mpirical **A**pproximations. Sometimes called "**S**ample **A**verage **A**pproximation" (SAA).
- Includes
  - **M**onte **C**arlo (MC),
  - **Q**uasi-**M**onte **C**arlo (QMC), Randomized QMC,
  - Deterministic quadrature approaches, e.g., **sparse grids**.
- Sample-**before**-you-go: Replace $f(x)$ by $\widehat{f}_N(x) = \int_{\Xi} F(x,\xi)\,\mathrm{d}\mathbb{P}_N(\xi) = \sum_{i=1}^{N} \pi_i F(x,\xi^i)$

Pros

- Recall again (3), using an EA for the expectation leads to

$$\min\left\{\widehat{f}_N(x)=\frac{1}{N}\sum_{i=1}^{N}\left[t+\frac{1}{2-2\beta}(\|S(z,\xi^i)-u_d\|_{L^2(D)}^2-2t)_+ +\frac{\alpha}{2}\|z\|_Z^2\right]\text{ over }(z,t)\in\mathcal{Z}_{\text{ad}}\times\mathbb{R}\right\}.$$

- There have been enormous advances in nonlinear programming and numerical PDE-constrained optimization over the past several decades.
- **Pro**: we may use a number of powerful optimization algorithms with convergence theory in a fully continuous setting.
- **Pro**: For a fixed EA of the objective, the stopping criterion can be based on the residual of the first-order system.
- **Pro**: Calculation of the ω-dependent states, adjoint states, and Hessian vector products are parallelizable.

Cons

- **Con**: In contrast to SGD-type methods, the convergence theory either works in the fully continuous (pre-EA) or the sample-based deterministic (post-EA) regime. OPEN PROBLEM
- **Con**: PDE-constrained optimization problems are large scale, the EA PDE-constrained problems can be significantly larger depending on N even for low stochastic dimension.
- So if we're using second-order information and globalization strategies:

  How slow could such an approach really be?

Does linear algebra *necessarily* slow things down?

Example

Recall the Example from p. 20.

Does linear algebra *necessarily* slow things down?

Example

Recall the Example from p. 20. 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 method with SAA (Section 2 below)

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

Open Problem(s)

1. Asymptotic consistency and rates of convergence as $\mathbb{P}_N \to \mathbb{P}$.
2. Link convergence of EA to convergence of the deterministic algorithms.

- ad 1. Some work for the strongly convex linear quadratic risk-neutral case has been done, but for nonsmooth, nonconvex problems such questions of stability are largely open.
- ad 2. The PD-Risk algorithm (Section 2) converges in the fully continuous setting, how can 1. or another method be used to argue convergence when using EA methods?

Section 2:
The Primal-Dual Risk Minimization Algorithm

Rethinking the Canonical Example

- We consider as our risk measure mean + CVaR:

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

Rethinking the Canonical Example

- We consider as our risk measure mean + CVaR:

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

- We can then transform the minimization problem

$$\min_{z\in\mathcal{Z}_{\text{ad}}} \{\mathcal{R}[\mathcal{J}(S(z))] + \wp(z)\}$$

Rethinking the Canonical Example

- We consider as our risk measure mean + CVaR:

$$\mathcal{R}(X) = (1-\nu)\mathbb{E}[X] + \nu \inf_{a\in\mathbb{R}} \left\{ a + \tfrac{1}{1-\beta}\mathbb{E}[(X-a)_+] \right\}, \quad \beta \in (0,1), \quad \nu \in (0,1].$$

- We can then transform the minimization problem

$$\min_{z\in\mathcal{Z}_{\text{ad}}} \{\mathcal{R}[\mathcal{J}(S(z))] + \wp(z)\}$$

into

$$\min_{(z,t)\in\mathcal{Z}_{\text{ad}}\times\mathbb{R}} \underbrace{(1-\nu)E_{\mathbb{P}}[\mathcal{J}(S(z))] + \wp(z) + \nu t}_{=:g(x)} + \overbrace{\frac{\nu}{1-\beta}\mathbb{E}_{\mathbb{P}}\left[(\mathcal{J}(S(z))-t)_+\right]}^{=:\Phi[G(x)]}$$

where $x = (z,t)$, $\mathcal{X}_{\text{ad}} = \mathcal{Z}_{\text{ad}} \times \mathbb{R}$, $G(x) := \mathcal{J}(S(z)) - t$ and

$$\Phi[X] := \frac{\nu}{1-\beta}\mathbb{E}_{\mathbb{P}}\left[(X)_+\right]$$

Examples

You could make a similar transformation with a number of other important examples:

Using $\Phi(X) = \mathbb{E}[(X)_+]$, we can represent three other popular risk measures:

- Mean-plus-semideviation of order 1

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

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

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

- The buffered probability of exceedence

$$\mathcal{R}(X) = \inf_{a \geq 0} \mathbb{E}[(a(X - \tau) + 1)_+], \;\; \tau \in \mathbb{R}.$$

General coherent measures of risk also treatable.

Properties of Φ

- Φ has a number of very favorable properties:
  - **positively homogeneous**
  - **subadditive**
  - **convex**
  - **montone wrt the usual ordering on** $L^1(\Omega, \mathcal{F}, \mathbb{P})$
  - **continuous on** $L^1(\Omega, \mathcal{F}, \mathbb{P})$
- If Φ^* is the Fenchel conjugate of Φ and $\mathfrak{A} = \mathrm{dom}(\Phi^*)$ its essential domain, then

$$\Phi[X] = \sup_{\vartheta \in \mathfrak{A}} \mathbb{E}[\vartheta X]$$

  where

$$\mathfrak{A} = \{\vartheta \in L^\infty(\Omega, \mathcal{F}, \mathbb{P}) \mid 0 \leq \vartheta \leq \nu/(1-\beta) \text{ a.s. } \}$$

- We will work mainly work with $L^2(\Omega, \mathcal{F}, \mathbb{P})$ instead of $L^1(\Omega, \mathcal{F}, \mathbb{P})$ for obvious reasons below.

Rethinking the Canonical Example

- Our canonical example takes on the form

$$\min_{x \in \mathcal{X}_{\mathrm{ad}}} \sup_{\vartheta \in \mathfrak{A}} g(x) + \mathbb{E}[\vartheta G(x)]$$

Rethinking the Canonical Example

- Our canonical example takes on the form

$$\min_{x\in\mathcal{X}_{\mathrm{ad}}} \sup_{\vartheta\in\mathfrak{A}} g(x)+\mathbb{E}[\vartheta G(x)]$$

- The objective $\ell : X \times L^2(\Omega,\mathcal{F},\mathbb{P}) \to \mathbb{R}$

$$\ell(x,\vartheta) := g(x)+\mathbb{E}[\vartheta G(x)]$$

  looks a lot like a Lagrangian.

- This motivates an algorithmic approach based on the **method of multipliers**.
- We need the generalized augmented Lagrangian (on the **dual** variables).

Rethinking the Canonical Example

- Our canonical example takes on the form

$$\min_{x \in \mathcal{X}_{\mathrm{ad}}} \sup_{\vartheta \in \mathfrak{A}} g(x) + \mathbb{E}[\vartheta G(x)]$$

- The objective $\ell : X \times L^2(\Omega, \mathcal{F}, \mathbb{P}) \to \mathbb{R}$

$$\ell(x, \vartheta) := g(x) + \mathbb{E}[\vartheta G(x)]$$

  looks a lot like a Lagrangian.

- This motivates an algorithmic approach based on the **method of multipliers**.
- We need the generalized augmented Lagrangian (on the **dual** variables).
- This means we will solve subproblems in x using the objective

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

Rethinking the Canonical Example

- Our canonical example takes on the form

$$\min_{x \in \mathcal{X}_{\text{ad}}} \sup_{\vartheta \in \mathfrak{A}} g(x) + \mathbb{E}[\vartheta G(x)]$$

- The objective $\ell : X \times L^2(\Omega, \mathcal{F}, \mathbb{P}) \to \mathbb{R}$

$$\ell(x, \vartheta) := g(x) + \mathbb{E}[\vartheta G(x)]$$

  looks a lot like a Lagrangian.
- This motivates an algorithmic approach based on the **method of multipliers**.
- We need the generalized augmented Lagrangian (on the **dual** variables).
- This means we will solve subproblems in x using the objective

$$\begin{aligned} L(x, \lambda, r) &= \max_{\theta \in \mathfrak{A}} \left\{ \ell(x, \theta) - \frac{1}{2r} \mathbb{E}[(\lambda - \theta)^2] \right\} \\ &= g(x) + \mathbb{E}[\lambda G(x)] + \frac{r}{2} \mathbb{E}[G(x)^2] - \frac{1}{2r} \mathbb{E}[\{(\mathrm{Id} - \mathfrak{P}_{\mathfrak{A}})(rG(x) + \lambda)\}^2]. \end{aligned}$$

Properties of L

- The proposed algorithm should make use of the functional L given by

$$L(x,\lambda,r) = g(x) + \mathbb{E}[\lambda G(x)] + \frac{r}{2}\mathbb{E}[G(x)^2] - \frac{1}{2r}\mathbb{E}[\{(\mathrm{Id} - \mathfrak{P}_{\mathfrak{A}})(rG(x)+\lambda)\}^2].$$

where Id **identity** on $L^2(\Omega,\mathcal{F},\mathbb{P})$, $\mathfrak{P}$ **projection** onto $\mathfrak{A}$, and the maximizer is given by

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

- Note that the current structure allows a simple pointwise projection:

$$\mathfrak{P}(X) = X - (-X)_+ - (X - \frac{\nu}{1-\beta})_+$$

- In an EA setting, this amounts to one projection for each $i = 1, \ldots, N$.

Properties of L

- The proposed algorithm should make use of the functional L given by

$$L(x,\lambda,r) = g(x) + \mathbb{E}[\lambda G(x)] + \frac{r}{2}\mathbb{E}[G(x)^2] - \frac{1}{2r}\mathbb{E}[\{(\mathrm{Id} - \mathfrak{P}_{\mathfrak{A}})(rG(x)+\lambda)\}^2].$$

  where Id **identity** on $L^2(\Omega,\mathcal{F},\mathbb{P})$, $\mathfrak{P}$ **projection** onto $\mathfrak{A}$, and the maximizer is given by

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

- Note that the current structure allows a simple pointwise projection:

$$\mathfrak{P}(X) = X - (-X)_+ - (X - \frac{\nu}{1-\beta})_+$$

- In an EA setting, this amounts to one projection for each $i = 1,\dots,N$.
- L can be written as the functional

$$L(x,\lambda,r) = g(x) + \Phi_{r,\lambda}(G(x)),$$

  where $\Phi_{r,\lambda}$ is the **infimal convolution** of Φ with $\Psi_{r,\lambda}(Y) = \mathbb{E}[\lambda Y] + \frac{r}{2}\mathbb{E}[Y^2]$.

Properties of L

- In fact, $\Phi_{r,\lambda}[X] = \mathbb{E}[\phi(X,\lambda,r)]$ where the scalar function $\phi : \mathbb{R} \times \mathbb{R} \times (0,\infty) \to \mathbb{R}$ is given by

$$\phi(x,t,r) := \frac{1}{2r}\{(rx+t)_+^2 - (rx+(t-1))_+^2 - t^2\}$$

- Thus, despite the initially complicated formula for L, there is a simple equivalent formula.

Properties of L

- In fact, $\Phi_{r,\lambda}[X] = \mathbb{E}[\phi(X, \lambda, r)]$ where the scalar function $\phi : \mathbb{R} \times \mathbb{R} \times (0, \infty) \to \mathbb{R}$ is given by

$$\phi(x, t, r) := \frac{1}{2r}\{(rx + t)_+^2 - (rx + (t-1))_+^2 - t^2\}$$

- Thus, despite the initially complicated formula for L, there is a simple equivalent formula.
- We also have the bounds[a]

$$\Phi(X) - \frac{K^2}{r} \le \Phi_{r,\lambda}(X) \le \Phi(X) \quad \forall\, \theta \in \mathfrak{A},\ r > 0$$

  where K is the Lipschitz modulus of Φ at 0.
- This provides simple upper and lower bounds on L!
- A generalized (Newton/slant) derivative of $\phi'_x(\cdot, t, r)$ is given by

$$\mathcal{G}_{t,r}(x) = r[\mathrm{sign}(rx + 1) - \mathrm{sign}(rx + t - 1)]$$

  These are needed for the quadratic model in the trust-region subproblem, see p. 51, 52.

[a] These are a direct consequence of epi-regularized risk measures.

Summary

- We seek to **solve** the canonical example by directly **exploiting the properties** of the risk measure.

Summary

- We seek to **solve** the canonical example by directly **exploiting the properties** of the risk measure.
- The **step calculation** the primal variables $x \in X$ should be determined by solving (inexactly)

$$\min_x L(x, \lambda_k, r_k) = g(x) + \Phi_{r_k,\lambda_k}(G(x))$$

  for fixed **dual variable** λ_k and **penalty parameter** r_k.

Summary

- We seek to **solve** the canonical example by directly **exploiting the properties** of the risk measure.
- The **step calculation** the primal variables $x \in X$ should be determined by solving (inexactly)

  $$\min_x L(x, \lambda_k, r_k) = g(x) + \Phi_{r_k,\lambda_k}(G(x))$$

  for fixed **dual variable** λ_k and **penalty parameter** r_k.
- $g(x)$ is at least C^1 and $\Phi_{r,\lambda}(G(x))$ is $C^{1,1}$ (provided $\mathcal{J}(S(z))$, $\wp$ are sufficiently smooth).

Summary

- We seek to **solve** the canonical example by directly **exploiting the properties** of the risk measure.
- The **step calculation** the primal variables $x \in X$ should be determined by solving (inexactly)

  $$\min_{x} L(x, \lambda_k, r_k) = g(x) + \Phi_{r_k,\lambda_k}(G(x))$$

  for fixed **dual variable** λ_k and **penalty parameter** r_k.
- $g(x)$ is at least C^1 and $\Phi_{r,\lambda}(G(x))$ is $C^{1,1}$ (provided $\mathcal{J}(S(z))$, $\wp$ are sufficiently smooth).
- Given x_k we update λ_k using the prox-operator

  $$\lambda_{k+1} = \mathfrak{P}(r_k G(x_k) + \lambda_k)$$

  and $r_{k+1} \geq r_k$ (if necessary).

The Algorithm

Algorithm 1: Primal-Dual Risk Minimization

```
Given $x_0 \in \mathcal{X}_{\text{ad}}$, $r_0 \in (0, \infty)$, $\lambda_0 \in \mathfrak{A}$, $\rho_x \in (0,1)$, $\rho_\lambda \in (0,1)$, $\rho_r \in (1, \infty)$, $0 < \tau_x < \tau_{x,0}$, 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. $\|x_{k+1} - \mathfrak{P}_{\mathcal{X}_{\text{ad}}}(x_{k+1} - \nabla_x L(x_{k+1}, \lambda_k, r_k))\|_{\mathcal{X}} \leq \tau_{x,k}$;
    2 Set $\lambda_{k+1} = \Lambda(x_{k+1}, \lambda_k, r_k)$;
    3 if (4) is satisfied then
        return $x_{k+1}$
    4 if $\|\lambda_k - \lambda_{k+1}\|_{\mathcal{Y}} > \tau_{\lambda,k}$ then
        $r_{k+1} = \rho_r r_k$
    5 Set $\tau_{x,k+1} = \rho_x \tau_{x,k}$ and $\tau_{\lambda,k+1} = \rho_\lambda \tau_{\lambda,k}$.
```

Practical Stopping Criterion

$$\|x_{k+1} - \mathfrak{P}_{\mathcal{X}_{\text{ad}}}(x_{k+1} - \nabla_x L(x_{k+1}, \lambda_k, r_k))\|_{\mathcal{X}} \leq \tau_x \tag{4a}$$

$$\|\lambda_k - \lambda_{k+1}\|_{\mathcal{Y}} \leq \tau_\lambda \tag{4b}$$

Remarks on convergence

- The **PD-Risk Algorithm** is more versatile than just the canonical example.
- There is **convergence theory** for the **primal variables** x, **dual variables** λ in both **convex** and **nonconvex** settings.
- In the convex setting, we can consider ϵ-minimizers.
- In the nonconvex setting, we need to consider ϵ-stationary points.
- We will only discuss the convergence theory for the **primal variables** in the context of the **canonical example** here (i.e. in the convex case).
- For readability, we work with the general structure

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

Convergence of the primal variables

Definition (ϵ-Minimizers)

An ϵ-minimizer, $\epsilon \geq 0$, is any $x \in \mathcal{X}_{\mathrm{ad}}$ such that

$$g(x) + \Phi(G(x)) - \epsilon \leq \min_{y \in \mathcal{X}_{\mathrm{ad}}} \{g(y) + \Phi(G(y))\}.$$

Convergence of the primal variables

Definition (ϵ-Minimizers)

An ϵ-minimizer, $\epsilon \geq 0$, is any $x \in \mathcal{X}_{\mathrm{ad}}$ such that

$$g(x) + \Phi(G(x)) - \epsilon \leq \min_{y \in \mathcal{X}_{\mathrm{ad}}} \{g(y) + \Phi(G(y))\}.$$

Similarly, an ϵ-minimizer, $\epsilon \geq 0$, of $L(\cdot, \lambda, r)$ over $\mathcal{X}_{\mathrm{ad}}$ is any $x \in \mathcal{X}_{\mathrm{ad}}$ such that

$$L(x, \lambda, r) - \epsilon \leq \inf_{y \in \mathcal{X}_{\mathrm{ad}}} L(y, \lambda, r).$$

- The general theory requires four main assumptions.
- These are stated on the next slide for your later convenience.
- They are all satisfied for the canonical example. HOMEWORK!

Convergence of the primal variables

Assumption

1. *The optimization space* $\mathcal{X}$ *is a reflexive Banach space and* $\neq \mathcal{X}_{\text{ad}} \subseteq \mathcal{X}$ *is convex and closed.*
2. *The deterministic objective function* $g : \mathcal{X} \to \mathbb{R}$ *is weakly lower semicontinuous and the uncertain objective function* $G : \mathcal{X} \to L^2(\Omega, \mathcal{F}, \mathbb{P})$ *is completely continuous, i.e.,*

$$x_k \rightharpoonup x \;\; \textit{in } \mathcal{X} \quad \Longrightarrow \quad G(x_k) \to G(x) \;\; \textit{in } L^2(\Omega, \mathcal{F}, \mathbb{P}).$$

3. *The functional* $\Phi : \mathcal{Y} \to \mathbb{R}$ *is convex, positively homogeneous and monotonic*
4. *There exists a constant* $\gamma \in \mathbb{R}$ *such that the lower* γ*-level*

$$\operatorname{lev}_\gamma(\mathrm{P}) := \{x \in \mathcal{X} \mid g(x) + \Phi(G(x)) \leq \gamma\} \cap \mathcal{X}_{\text{ad}}, \tag{5}$$

is nonempty and bounded.

Convergence of the primal variables

Theorem

Convergence of the primal variables

Theorem

Let $\{(x_k, \lambda_k, r_k)\} \subset \mathcal{X}_{\mathrm{ad}} \times \mathcal{Y} \times (0, \infty)$ *denote the sequence of iterates produced by Algorithm 1,*

Convergence of the primal variables

Theorem

Let $\{(x_k, \lambda_k, r_k)\} \subset \mathcal{X}_{\mathrm{ad}} \times \mathcal{Y} \times (0, \infty)$ *denote the sequence of iterates produced by Algorithm 1, where* $r_k \to r^\star > 0$ *(possibly* $+\infty$*) and* x_k *satisfies*

$$L(x_k, \lambda_{k-1}, r_{k-1}) - \epsilon_k \leq \inf_{x \in \mathcal{X}_{\mathrm{ad}}} L(x, \lambda_{k-1}, r_{k-1}) \tag{6}$$

for some sequence $\{\epsilon_k\} \subset [0, \infty)$ *with* $\epsilon_k \to \epsilon^\star$ *(possibly zero).*

Convergence of the primal variables

Theorem

Let $\{(x_k, \lambda_k, r_k)\} \subset \mathcal{X}_{\text{ad}} \times \mathcal{Y} \times (0, \infty)$ *denote the sequence of iterates produced by Algorithm 1, where* $r_k \to r^\star > 0$ *(possibly* $+\infty$*) and* x_k *satisfies*

$$L(x_k, \lambda_{k-1}, r_{k-1}) - \epsilon_k \leq \inf_{x \in \mathcal{X}_{\text{ad}}} L(x, \lambda_{k-1}, r_{k-1}) \tag{6}$$

for some sequence $\{\epsilon_k\} \subset [0, \infty)$ *with* $\epsilon_k \to \epsilon^\star$ *(possibly zero). Then, any weak accumulation point* $x^\star$ *of* $\{x_k\}$ *satisfies*

$$g(x^\star) + \Phi(G(x^\star)) - \left(\frac{K^2}{r^\star} + \epsilon^\star\right) \leq \min_{x \in \mathcal{X}_{\text{ad}}} \{g(x) + \Phi(G(x))\}$$

where $K > 0$ *is such that* $\|\theta\|_{\mathcal{Y}} \leq K$ *for all* $\theta \in \mathfrak{A}$*. That is,* $x^\star$ *is a* $(\frac{K^2}{r^\star} + \epsilon^\star)$*-optimal solution.*

Convergence of the primal variables

Theorem

Let $\{(x_k, \lambda_k, r_k)\} \subset \mathcal{X}_{\mathrm{ad}} \times \mathcal{Y} \times (0, \infty)$ *denote the sequence of iterates produced by Algorithm 1, where* $r_k \to r^\star > 0$ *(possibly* $+\infty$*) and* x_k *satisfies*

$$L(x_k, \lambda_{k-1}, r_{k-1}) - \epsilon_k \leq \inf_{x \in \mathcal{X}_{\mathrm{ad}}} L(x, \lambda_{k-1}, r_{k-1}) \tag{6}$$

for some sequence $\{\epsilon_k\} \subset [0, \infty)$ *with* $\epsilon_k \to \epsilon^\star$ *(possibly zero). Then, any weak accumulation point* $x^\star$ *of* $\{x_k\}$ *satisfies*

$$g(x^\star) + \Phi(G(x^\star)) - \left(\frac{K^2}{r^\star} + \epsilon^\star\right) \leq \min_{x \in \mathcal{X}_{\mathrm{ad}}} \{g(x) + \Phi(G(x))\}$$

where $K > 0$ *is such that* $\|\theta\|_{\mathcal{Y}} \leq K$ *for all* $\theta \in \mathfrak{A}$*. That is,* $x^\star$ *is a* $(\frac{K^2}{r^\star} + \epsilon^\star)$*-optimal solution.*

- As mentioned earlier, this is the relevant convergence statement for convex problems.
- It holds for more than just the canonical example!

Convergence of the primal variables

Proof.

- This type of convergence proof is **typical** for PDE-constrained optimization.
- We need to work with the **weak topology** due to a lack of norm compactness of closed balls in general Hilbert spaces. **Convergence** is done in the **continuous** setting.

Convergence of the primal variables

Proof.

- This type of convergence proof is **typical** for PDE-constrained optimization.
- We need to work with the **weak topology** due to a lack of norm compactness of closed balls in general Hilbert spaces. **Convergence** is done in the **continuous** setting.
- Suppose $x^\star$ is a weak accumulation point of $\{x_k\}$.

Convergence of the primal variables

Proof.

- This type of convergence proof is **typical** for PDE-constrained optimization.
- We need to work with the **weak topology** due to a lack of norm compactness of closed balls in general Hilbert spaces. **Convergence** is done in the **continuous** setting.
- Suppose $x^\star$ is a weak accumulation point of $\{x_k\}$.
- By definition, there exists $\{x_{k_j}\}$ such that $x_{k_j} \rightharpoonup x^\star$ in $\mathcal{X}_{\text{ad}}$.

Convergence of the primal variables

Proof.

- This type of convergence proof is **typical** for PDE-constrained optimization.
- We need to work with the **weak topology** due to a lack of norm compactness of closed balls in general Hilbert spaces. **Convergence** is done in the **continuous** setting.
- Suppose $x^\star$ is a weak accumulation point of $\{x_k\}$.
- By definition, there exists $\{x_{k_j}\}$ such that $x_{k_j} \rightharpoonup x^\star$ in $\mathcal{X}_{\mathrm{ad}}$.
- The bounds on $\Phi_{r,\lambda}$ (p. 33), continuity of Φ, weak lsc of g and complete continuity of G yield:

Convergence of the primal variables

Proof.

- This type of convergence proof is **typical** for PDE-constrained optimization.
- We need to work with the **weak topology** due to a lack of norm compactness of closed balls in general Hilbert spaces. **Convergence** is done in the **continuous** setting.
- Suppose $x^\star$ is a weak accumulation point of $\{x_k\}$.
- By definition, there exists $\{x_{k_j}\}$ such that $x_{k_j} \rightharpoonup x^\star$ in $\mathcal{X}_{\mathrm{ad}}$.
- The bounds on $\Phi_{r,\lambda}$ (p. 33), continuity of Φ, weak lsc of g and complete continuity of G yield:

$$\begin{aligned}
g(x) + \Phi(G(x)) &\geq \liminf_{k_j\to\infty} L(x, \lambda_{k_j-1}, r_{k_j-1}) && \textbf{(upper bd. on } \Phi\textbf{)}\\
&\geq \liminf_{k_j\to\infty} \{L(x_{k_j}, \lambda_{k_j-1}, r_{k_j-1}) - \epsilon_{k_j-1}\} && \textbf{(def. of } x_k\textbf{)}\\
&\geq \liminf_{k_j\to\infty} \{g(x_{k_j}) + \Phi(G(x_{k_j})) - \frac{K^2}{r_{k_j-1}} - \epsilon_{k_j-1}\} && \textbf{(lower bd. on } \Phi\textbf{)}\\
&\geq g(x^\star) + \Phi(G(x^\star)) - \left(\frac{K^2}{r^\star} + \epsilon^\star\right) \qquad \forall x \in \mathcal{X}_{\mathrm{ad}}. && \textbf{(}g\textbf{ wlsc, } G\textbf{ cc, } \Phi\textbf{ cont.)}
\end{aligned}$$

□

Convergence of the primal variables

Sufficient condition

The existence of weak accumulation points can be guaranteed by the condition: $\exists \gamma^\star > 0$ satisfying

$$\gamma^\star > \left(\min_{x \in \mathcal{X}_{\mathrm{ad}}} \{ g(x) + \Phi(G(x)) \} \right) + \frac{K^2}{r^\star} + \epsilon^\star$$

such that the $\gamma^\star$-lower level set of $\mathrm{lev}_{\gamma^\star}(P)$ is bounded.

Convergence of the dual variables

Theorem

In addition to the standing assumptions, define $v(\theta) := \inf_{x \in \mathcal{X}_{\mathrm{ad}}} \ell(x, \theta)$*. Assume there exists* $\lambda_0 \in \mathfrak{A}$ *such that* $v(\lambda_0) > -\infty$*.*

Convergence of the dual variables

Theorem

In addition to the standing assumptions, define $v(\theta) := \inf_{x\in\mathcal{X}_{\rm ad}} \ell(x,\theta)$. *Assume there exists* $\lambda_0 \in \mathfrak{A}$ *such that* $v(\lambda_0) > -\infty$. *Suppose Algorithm 1 is executed with* $\{r_k\} \subset [r,\infty)$ *for some* $r > 0$ *and* $x_{k+1} \in \mathcal{X}_{\rm ad}$ *satisfying*

$$L(x_{k+1},\lambda_k,r_k) - \frac{\epsilon_k^2}{2r_k} \le \inf_{x\in\mathcal{X}_{\rm ad}} L(x,\lambda_k,r_k) \quad \textit{with} \quad \sum_{k=0}^{\infty} \epsilon_k < \infty,\ \epsilon_k \ge 0. \tag{7}$$

Convergence of the dual variables

Theorem

In addition to the standing assumptions, define $v(\theta) := \inf_{x \in \mathcal{X}_{\text{ad}}} \ell(x,\theta)$*. Assume there exists* $\lambda_0 \in \mathfrak{A}$ *such that* $v(\lambda_0) > -\infty$*. Suppose Algorithm 1 is executed with* $\{r_k\} \subset [r, \infty)$ *for some* $r > 0$ *and* $x_{k+1} \in \mathcal{X}_{\text{ad}}$ *satisfying*

$$L(x_{k+1}, \lambda_k, r_k) - \frac{\epsilon_k^2}{2r_k} \leq \inf_{x \in \mathcal{X}_{\text{ad}}} L(x, \lambda_k, r_k) \quad \textit{with} \quad \sum_{k=0}^{\infty} \epsilon_k < \infty, \ \epsilon_k \geq 0. \tag{7}$$

Furthermore, assume that for each k*, there exists a saddle point:*

$$\inf_{x \in \mathcal{X}_{\text{ad}}} \sup_{\theta \in \mathfrak{A}} \left\{ \ell(x,\theta) - \frac{1}{2r_k} \mathbb{E}[(\theta - \eta)^2] \right\} = \sup_{\theta \in \mathfrak{A}} \inf_{x \in \mathcal{X}_{\text{ad}}} \left\{ \ell(x,\theta) - \frac{1}{2r_k} \mathbb{E}[(\theta - \eta)^2] \right\} \quad \forall \eta \in \mathcal{Y}.$$

Convergence of the dual variables

Theorem

In addition to the standing assumptions, define $v(\theta) := \inf_{x \in \mathcal{X}_{\text{ad}}} \ell(x, \theta)$*. Assume there exists* $\lambda_0 \in \mathfrak{A}$ *such that* $v(\lambda_0) > -\infty$*. Suppose Algorithm 1 is executed with* $\{r_k\} \subset [r, \infty)$ *for some* $r > 0$ *and* $x_{k+1} \in \mathcal{X}_{\text{ad}}$ *satisfying*

$$L(x_{k+1}, \lambda_k, r_k) - \frac{\epsilon_k^2}{2r_k} \leq \inf_{x \in \mathcal{X}_{\text{ad}}} L(x, \lambda_k, r_k) \quad \textit{with} \quad \sum_{k=0}^{\infty} \epsilon_k < \infty, \ \epsilon_k \geq 0. \tag{7}$$

Furthermore, assume that for each k*, there exists a saddle point:*

$$\inf_{x \in \mathcal{X}_{\text{ad}}} \sup_{\theta \in \mathfrak{A}} \left\{ \ell(x, \theta) - \frac{1}{2r_k} \mathbb{E}[(\theta - \eta)^2] \right\} = \sup_{\theta \in \mathfrak{A}} \inf_{x \in \mathcal{X}_{\text{ad}}} \left\{ \ell(x, \theta) - \frac{1}{2r_k} \mathbb{E}[(\theta - \eta)^2] \right\} \quad \forall \eta \in \mathcal{Y}.$$

Then $\{\lambda_k\}$ *converges weakly to* $\lambda^\star \in \mathfrak{A}$*, which is a solution to the dual problem:* $\max_{\theta \in \mathfrak{A}} \ v(\theta)$

- Uses the same assumptions as the primal theorem.
- **Much** more involved proof.

Convergence of the dual variables

Sufficient conditions for the dual variables

$\ell(x,\cdot) - \frac{1}{2r_k}\mathbb{E}[(\cdot - \eta)^2]$ is concave and weakly usc over the weakly compact convex set $\mathfrak{A}$. If $\ell(\cdot,\lambda)$ is quasiconvex for each $\lambda \in \mathfrak{A}$, then Sion's theorem ensures the existence of a saddle point.

Extension to Nonconvex Case

1. Also addressed in the paper.
2. Need to consider convergence of ϵ-stationary points.
3. Convergence proofs require slightly more structure on g and G.

Section 3:
Implementation and Numerical Solution

Solving the Subproblems

- We recommend using a **matrix-free trust-region approach** to solve the subproblems.
- “**Matrix-free**” does not mean there are no matrices involved, rather no matrix factorizations, i.e., direct solvers, or eigenvalue calculations are used.
- We briefly discuss the necessary computations using the CG method proposed by Steihaug.
- Assume for discussion that $\mathcal{Z}_{\text{ad}} = Z$.
- This will tie in parts of Part I and Part II.
- We remain in the fully continuous setting for readability, but indicate the necessary computations in an EA approach.

Solving the Subproblems: Quadratic Model, Gradients, & Hessians

- Take as the quadratic model for the canonical

$$\varphi(p) = (\nabla f(x), d)_X + \frac{1}{2}(H(x)d, d)_X$$

- Here, $\nabla f(x) = \mathfrak{R}_{\mathrm{Riesz}} f'(x)$, where

$$f'(x) = g'(x) + \Phi'_{r,\theta}(G(x)) \circ G'(x).$$

- The components of $f'(x)$ are as follows:

$$\begin{aligned}
\left[g'(x)\right]_t \delta t &= \nu \delta t \\
\left[g'(x)\right]_z \delta z &= (1-\nu)\mathbb{E}_{\mathbb{P}}[(S(z)-u_d, S'(z)\delta z)_Z] + \alpha(z, \delta z)_Z \\
\left[\Phi'_{r,\theta}(G(x)) \circ G'(x)\right]_t \delta t &= \frac{\nu}{r(1-\beta)}\mathbb{E}_{\mathbb{P}}[\phi'(G(x), \theta, r)]\delta t \\
\left[\Phi'_{r,\theta}(G(x)) \circ G'(x)\right]_z \delta z &= \frac{\nu}{r(1-\beta)}\mathbb{E}_{\mathbb{P}}[\phi'(G(x), \theta, r)(S(z)-u_d, S'(z)\delta z)_Z]
\end{aligned}$$

Solving the Subproblems: Quadratic Model, Gradients, & Hessians

- Combining these observations, we have

$$\left[f'(x)\right]_t \delta t = \mathbb{E}_{\mathbb{P}_\xi}[\nu + \frac{\nu}{r(1-\beta)}\phi'(G(x),\theta,r)]\delta t$$

$$[f'(x)]_z \delta z = \left(\alpha z + \mathbb{E}_{\mathbb{P}_\xi}\left[\left((1-\nu) + \frac{\nu}{r(1-\beta)}\phi'(G(x),\theta,r)\right)(\boldsymbol{B}^*\lambda)\right], \delta z\right)_Z$$

where $\lambda(\xi)$ solves

$$\boldsymbol{A}(\xi)^*\lambda = u(\xi) - u_d \text{ in } H^{-1}(D)$$

and $u(\xi) = S(z,\xi)$ solves

$$\boldsymbol{A}(\xi)u = \boldsymbol{B}(\xi)z + f(\xi) \text{ in } H^{-1}(D).$$

for $\mathbb{P}_\xi$-a.e. $\xi \in \Xi$.

- Compare this to the SGD discussions (with $\nu = 1$):
  - $\mathbb{E}_{\mathbb{P}_\xi}$ disappears
  - $\phi'(G(x),\theta,r)/r(1-\beta)$ is a regularization of the characteristic functions.

Solving the Subproblems: Quadratic Model, Gradients, & Hessians

- Keeping the spacial terms continuous, in an EA setting, e.g., Monte Carlo, we would need:
- N **states solves**
$$\boldsymbol{A}(\xi^i)u = \boldsymbol{B}(\xi^i)z + f(\xi^i) \text{ in } H^{-1}(D) \quad i = 1, \dots, N,$$
which can be done in parallel.
- $\leq N$ **adjoint solves**
$$\boldsymbol{A}(\xi^i)^*\lambda = u(\xi^i) - u_d \text{ in } H^{-1}(D) \quad i = 1, \dots, N,$$
which can also be done in parallel, but only when the i^{th} state solve is available.
- N **function evaluations**
$$\phi'(G(x, \xi^i), \theta, r) \quad i = 1, \dots, N$$
Recall: $G(x, \xi) = \mathcal{J}(S(z, \xi)) - t$.
- $\leq N$ **"matrix-vector" products**
$$\boldsymbol{B}^*(\xi^i)\lambda(\xi^i) \quad i = 1, \dots, N.$$
- $N + 1$ **"axpy's"** for the weighted sum.
- Don't forget the Riesz maps/proper inner products when considering the fully discrete problem!

Solving the Subproblems: Quadratic Model, Gradients, & Hessians

- For a matrix-free trust region approach to the subproblems, the previous slide tells us how to calculate the gradient used in the quadratic model and for the trial point (first step) of the algorithm.
- We only need to do this once.
- Otherwise, we need a way of computing "Hessian-vector" products. This mirrors the discussion on p. 33 in Part I. However, Φ is only $C^{1,1}$.
- Start with $g(x)$. Given a direction $\delta x = (\delta z, \delta t) \in X$, we have

  $$g''(x)\delta x = \alpha \delta z + (1-\nu)\mathbb{E}[\mathbf{B}^*\mu]$$

  where $\mu(\xi)$ solves

  $$\boldsymbol{A}(\xi)^*\mu = w(\xi) \text{ in } H^{-1}(D)$$

  and $w(\xi)$ solves

  $$\boldsymbol{A}(\xi)w = \boldsymbol{B}(\xi)w \text{ in } H^{-1}(D).$$

  for $\mathbb{P}_{\boldsymbol{\xi}}$-a.e. $\xi \in \Xi$.
- We can easily adapt this to an EA replacing ξ and "$\mathbb{P}_{\boldsymbol{\xi}}$-a.e. $\xi \in \Xi$" by ξ^i for all $i = 1, \dots, N$.

Solving the Subproblems: Quadratic Model, Gradients, & Hessians

- We can use generalized Hessians (Newton/Slant derivatives) for the second derivative of Φ without violating the assumptions used to prove convergence of Steihaug's CG method.

Solving the Subproblems: Quadratic Model, Gradients, & Hessians

Hessian Computation for "Φ-part"

- Define the Lagrangian $\mathcal{L}(u, x, \lambda)$:

$$\mathcal{L}(u, x, \lambda) = \frac{\nu}{1-\beta}\mathbb{E}_{\mathbb{P}_{\boldsymbol{\xi}}}[\phi(J(u, z) - t, \theta, r)] + \langle \boldsymbol{A}u - [\boldsymbol{B}z + f], \lambda\rangle_{V,V^*}$$

Solving the Subproblems: Quadratic Model, Gradients, & Hessians

Hessian Computation for "Φ-part"

- Define the Lagrangian $\mathcal{L}(u, x, \lambda)$:

$$\mathcal{L}(u, x, \lambda) = \frac{\nu}{1-\beta}\mathbb{E}_{\mathbb{P}_{\boldsymbol{\xi}}}[\phi(J(u, z) - t, \theta, r)] + \langle \boldsymbol{A}u - [\boldsymbol{B}z + f], \lambda\rangle_{V,V^*}$$

- Given $x \in X$, we have $u(\xi)$: $\boldsymbol{A}(\xi)u(\xi) = \boldsymbol{B}(\xi)z + f(\xi)$ w.p.1

Solving the Subproblems: Quadratic Model, Gradients, & Hessians

Hessian Computation for "Φ-part"

- Define the Lagrangian $\mathcal{L}(u, x, \lambda)$:

$$\mathcal{L}(u, x, \lambda) = \frac{\nu}{1-\beta}\mathbb{E}_{\mathbb{P}_{\boldsymbol{\xi}}}[\phi(J(u, z) - t, \theta, r)] + \langle \boldsymbol{A}u - [\boldsymbol{B}z + f], \lambda \rangle_{V,V^*}$$

- Given $x \in X$, we have $u(\xi)$: $\boldsymbol{A}(\xi)u(\xi) = \boldsymbol{B}(\xi)z + f(\xi)$ w.p.1
- Given $u(\cdot)$, we solve for $\lambda(\xi)$: $\boldsymbol{A}^*(\xi)\lambda(\xi) = u_d - u(\xi)$ w.p.1

Solving the Subproblems: Quadratic Model, Gradients, & Hessians

Hessian Computation for "Φ-part"

- Define the Lagrangian $\mathcal{L}(u, x, \lambda)$:

$$\mathcal{L}(u, x, \lambda) = \frac{\nu}{1-\beta} \mathbb{E}_{\mathbb{P}_{\boldsymbol{\xi}}}[\phi(J(u, z) - t, \theta, r)] + \langle \boldsymbol{A} u - [\boldsymbol{B} z + f], \lambda \rangle_{V, V^*}$$

- Given $x \in X$, we have $u(\xi)$: $\boldsymbol{A}(\xi) u(\xi) = \boldsymbol{B}(\xi) z + f(\xi)$ w.p.1
- Given $u(\cdot)$, we solve for $\lambda(\xi)$: $\boldsymbol{A}^*(\xi) \lambda(\xi) = u_d - u(\xi)$ w.p.1
- Given $u(\cdot), x$, direction $v \in X$ solve for $w(\xi)$: $\boldsymbol{A}(\xi) w = \boldsymbol{B}(\xi)[v]_z$.

Solving the Subproblems: Quadratic Model, Gradients, & Hessians

Hessian Computation for "Φ-part"

- Define the Lagrangian $\mathcal{L}(u, x, \lambda)$:

$$\mathcal{L}(u, x, \lambda) = \frac{\nu}{1-\beta}\mathbb{E}_{\mathbb{P}_{\boldsymbol{\xi}}}[\phi(J(u, z) - t, \theta, r)] + \langle \boldsymbol{A}u - [\boldsymbol{B}z + f], \lambda\rangle_{V,V^*}$$

- Given $x \in X$, we have $u(\xi)$: $\boldsymbol{A}(\xi)u(\xi) = \boldsymbol{B}(\xi)z + f(\xi)$ w.p.1
- Given $u(\cdot)$, we solve for $\lambda(\xi)$: $\boldsymbol{A}^*(\xi)\lambda(\xi) = u_d - u(\xi)$ w.p.1
- Given $u(\cdot), x$, direction $v \in X$ solve for $w(\xi)$: $\boldsymbol{A}(\xi)w = \boldsymbol{B}(\xi)[v]_z$.
- Given $u(\cdot), z, \lambda(\cdot), w(\cdot), v$ solve for $p(\xi)$:

$$\boldsymbol{A}(\xi)^* p(\xi) = \mathcal{L}''_{uu}(u, x, \lambda)w - \mathcal{L}''_{ux}(u, x, \lambda)v.$$

Solving the Subproblems: Quadratic Model, Gradients, & Hessians

Hessian Computation for "Φ-part"

- Define the Lagrangian $\mathcal{L}(u, x, \lambda)$:

$$\mathcal{L}(u, x, \lambda) = \frac{\nu}{1-\beta}\mathbb{E}_{\mathbb{P}_{\boldsymbol{\xi}}}[\phi(J(u, z) - t, \theta, r)] + \langle \boldsymbol{A}u - [\boldsymbol{B}z + f], \lambda\rangle_{V, V^*}$$

- Given $x \in X$, we have $u(\xi)$: $\boldsymbol{A}(\xi)u(\xi) = \boldsymbol{B}(\xi)z + f(\xi)$ w.p.1
- Given $u(\cdot)$, we solve for $\lambda(\xi)$: $\boldsymbol{A}^*(\xi)\lambda(\xi) = u_d - u(\xi)$ w.p.1
- Given $u(\cdot), x$, direction $v \in X$ solve for $w(\xi)$: $\boldsymbol{A}(\xi)w = \boldsymbol{B}(\xi)[v]_z$.
- Given $u(\cdot), z, \lambda(\cdot), w(\cdot), v$ solve for $p(\xi)$:

$$\boldsymbol{A}(\xi)^* p(\xi) = \mathcal{L}''_{uu}(u, x, \lambda)w - \mathcal{L}''_{ux}(u, x, \lambda)v.$$

- Given $u(\cdot), z, \lambda(\cdot), w(\cdot), v, p(\cdot)$ yields

$$H(x, \xi)v = \boldsymbol{B}(\xi)^* p - \nabla_{xu}\mathcal{L}(u, x, \lambda)w + \nabla_{xx}\mathcal{L}(u, x, \lambda)v.$$

- **This is just notation**. These red terms are not second derivatives as they do not exist a priori.

Solving the Subproblems: Quadratic Model, Gradients, & Hessians

Hessian Computation for "Φ-part"

- For the terms

$$\mathcal{L}''_{uu}(u,x,\lambda)w, \quad \mathcal{L}''_{ux}(u,x,\lambda)v, \quad \nabla_{xu}\mathcal{L}(u,x,\lambda)w, \quad \nabla_{xx}\mathcal{L}(u,x,\lambda)v.$$

  we recommend to formally apply the notation of Newton/slant derivatives from semismooth Newton methods to obtain expressions.

- These can be derived using the usual chain rule and the pointwise formula for the Newton derivatives of the scalar function ϕ. HOMEWORK!

Implementation Details

- We conclude with two illustrative examples.
- The parameters are

name	Pollutant	Burgers
CVAR	$\beta = 0.9$, $\nu = 0.75$	$\beta = 0.9$, $\nu = 0.75$

- The Burgers example is nonconvex as the associated differential equation is nonlinear.

- `iter`: Number of iterations for Algorithm 1.
- `nfval` is the total number of evaluations of $L(x, \lambda, r)$
- `ngrad` is the total number of evaluations of the gradient of $L(\cdot, \lambda, r)$
- `nhess` Hessian vector products
- `subiter` is the total number of subproblem iterations from step 1
- $\tau_x = 10^{-8}$, $\tau_\lambda = 10^{-6}$, $\rho_x = 0.1$, $\rho_\lambda = 0.1$ and $\rho_r = 10$

Example 1: A Contaminant Mitigation Problem

Find optimal placement of mitigating factors z by solving:

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

where $\kappa_s = 10^5$, $\kappa_c = 1$ and $S(z) = u : \Omega \to H^1(D)$ solves the weak form of

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

- $D = (0,1)^2$ is the physical domain, $(\Omega, \mathcal{F}, \mathbb{P})$ complete probability space
- $\mathcal{Z}$ is the control space, e.g., $L^2(D)$ or $\mathbb{R}^n$; $\mathcal{Z}_{\mathrm{ad}} = \{z \in \mathcal{Z}\,|\,0 \le z \le 1\}$.
- $\epsilon, \mathbb{V}, f$ are random variables or fields: permeability, wind, sources, etc.
- u is the advected pollutant.
- $\mathcal{R} : \mathcal{X} \to \overline{\mathbb{R}}$ is a convex combination of mean and CVaR

Example 1: A Contaminant Mitigation Problem

- Control $z \in \mathbb{R}^9$ for intensity of mitigating factors at nine fixed positions in space:

$$Bz = \sum_{k=1}^{9} z_k \exp\left(-\frac{(x-p_k)^\top (x-p_k)}{2\sigma^2}\right), \qquad \sigma = 0.05.$$

- **M**ean **V**alue: Replace uncertain parameters by mean values, solve deterministic problem.
- The least robust type of solution.

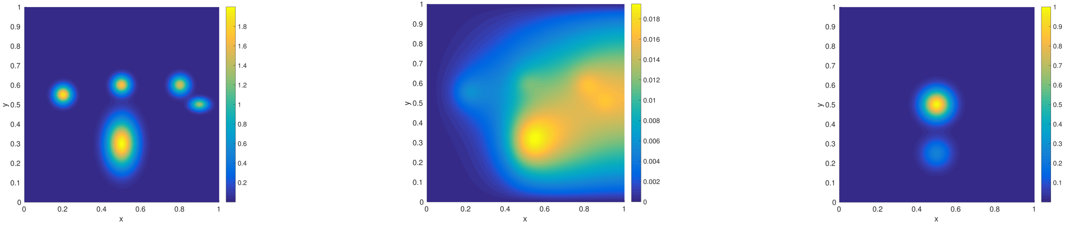


Figure: MV of sources f (l.), uncontrolled state in MV problem (m.), optimal control MVP (r.).

Results Example 1

- Q1-FEM on uniform mesh of 4096 quadrilaterals.
- Empirical approximation with 10,000 Monte Carlo samples.

	Algorithm 1				
name	iter	nfval	ngrad	subiter	nhess
CVAR	9	65	43	46	18

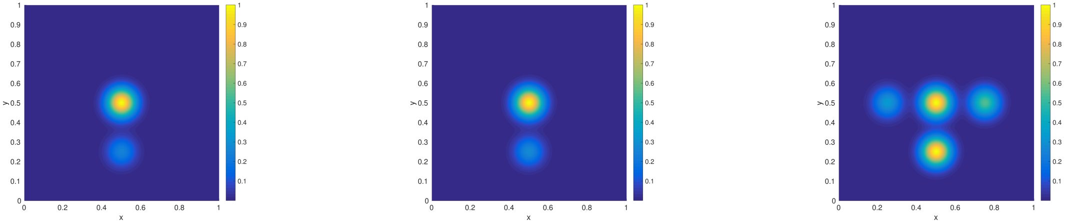


Figure: optimal solutions for MVP (l.), risk neutral (c.), Mean-CVaR (r.).

Example 2: Optimization of Steady Viscous Burger's Eq.

Find risk-averse optimal forcing term z by solving

$$\min_{z\in\mathcal{Z}} \left\{ \mathcal{R}\left(\frac{1}{2}\int_D (S(z)-1)^2 \,\mathrm{d}x\right) + \frac{\alpha}{2}\int_D z^2 \,\mathrm{d}x \right\}$$

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

$$\begin{aligned} -\nu(\omega)\partial_{xx} u + u\partial_x u &= f(\omega) + z && \text{in } D, \text{ a.s.} \\ [u(\omega)](0) = d_0(\omega), \quad [u(\omega)](1) &= d_1(\omega) && \text{a.s.} \end{aligned}$$

- $\alpha = 10^{-3}$, $D = (0,1)$, $\mathcal{Z} = L^2(0,1)$
- ν, f, d_0, d_1: Random viscosity, forcing, boundary values.

Results Example 2

- P1-FEM on (non-uniform) mesh of 256 intervals
- Sample average approximation with 10,000 Monte Carlo samples.
- Nonlinear PDE solve: Newton globalized with a backtracking line search.
- This problem is nonconvex.

	Algorithm 1				
name	iter	nfval	ngrad	subiter	nhess
CVAR	8	44	42	116	36

Summary Part II

- PDE-constrained optimization under uncertainty can be viewed as an extension of stochastic programming to infinite dimensions.
- We need to be especially careful when building algorithms (correct derivatives, inner products, matrix-free subproblems solvers,...)
- Stochastic approximation-based methods are initially attractive due to the low amount of PDE-solves per iteration.
- However, the sheer number of iterations needed to reach a solution of high accuracy is intractable.
- Coherent risk measures, e.g., CVaR, have a significant amount of exploitable structure.
- This leads to a provably convergent SAA-based primal-dual risk minimization algorithm.
- The performance of this algorithm is promising.

References: Papers on optimization of optoelectronic devices

These non-exhaustive lists are meant to give you a head start on the literature!

L. Adam, M. Hintermüller, D. Peschka, and T.M. Surowiec.
Optimization of a multiphysics problem in semiconductor laser design.
SIAM Journal on Applied Mathematics, 79(1):257–283, (2019).

L. Adam, M. Hintermüller, and T.M. Surowiec.
A semismooth Newton method with analytical path-following for the H^1-projection onto the Gibbs simplex.
IMA Journal of Numerical Analysis 39(3):1276–1295, (2019).

L. Adam, M. Hintermüller, and T.M. Surowiec.
A PDE-constrained optimization approach for topology optimization of strained photonic devices.
Optimization and Engineering 19(3), 521–557, (2018).

D. Peschka, N. Rotundo, and M. Thomas.
Doping optimization for optoelectronic devices.
Optical and Quantum Electronics, 50(3):125, (2018).

D. Peschka, M. Thomas, A. Glitzky, R. Nürnberg, K. Gärtner, M. Virgilio, S. Guha, Th. Schroeder, G. Capellini, and Th. Koprucki.
Modeling of edge-emitting lasers based on tensile strained germanium microstrips.
IEEE Photonics Journal, 7(3):1–15, (2015).

References: Books on PDE-constrained optimization

Lions, J.-L.

Optimal control of systems governed by partial differential equations.

Springer-Verlag, New York-Berlin, 1971.

Tröltzsch, F.

Optimal control of partial differential equations. Theory, methods and applications.

American Mathematical Society, Providence, RI, 2010.

Borzì, A. and Schulz, V.

Computational optimization of systems governed by partial differential equations.

Computational Science & Engineering, 8. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2012.

De los Reyes, J.C.

Numerical PDE-constrained optimization.

SpringerBriefs in Optimization. Springer, Cham, 2015.

Hinze, M., Pinnau, R., Ulbrich, M., and Ulbrich, S.

Optimization with PDE constraints.

Mathematical Modelling: Theory and Applications, 23. Springer, New York, 2009

Antil, A., Kouri, D.P., Lacasse, M.-D., and Ridzal, D. (eds.)

Frontiers in PDE-constrained optimization.

Papers based on the workshop held at the Institute for Mathematics and its Applications, Minneapolis, MN, June 610, 2016. The IMA Volumes in Mathematics and its Applications, 163. Springer, New York, 2018.

References: Books on stochastic optimization

Birge, J. R. and Louveaux, F.,

Introduction to stochastic programming.

Springer-Verlag, New York, 1997.

Shapiro, A., Dentcheva, D. and Ruszczyński, A..

Lectures on Stochastic Programming: Modeling and Theory.

SIAM, Philadelphia, 2009.

Ben-Tal, A., El Ghaoui, L., Nemirovski, A.

Robust optimization.

Princeton University Press, Princeton, NJ, 2009.

Prékopa, A.

Stochastic programming.

Mathematics and its Applications, 324. Kluwer Academic Publishers Group, Dordrecht, 1995.

Lan, G.

First-order and stochastic optimization methods for machine learning.

Springer, Cham, 2020.

Ruszczyński, A. and Shapiro, A. (eds.)

Handbooks in Operations Reseach & Management Science, Vol. 10 Elsevier Science B.V., 2003.

References: Books on optimization theory and PDEs

ATTOUCH, H., BUTTAZZO, G. AND MICHAILLE, G.
Variational analysis in Sobolev and BV spaces,
Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2006.

BONNANS, J. F. AND SHAPIRO, A.
Perturbation Analysis of Optimization Problems.
Springer Verlag, Berlin, Heidelberg, New York, 2000.

EVANS, L.C.
Partial differential equations.
American Mathematical Society, Providence, RI, 2010.

GILBARG, D. AND TRUDINGER, N. S.
Elliptic partial differential equations of second order.
Reprint of the 1998 edition. Classics in Mathematics. Springer-Verlag, Berlin, 2001.

GRISVARD, P.
Elliptic problems in nonsmooth domains.
Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2011.

LUENBERGER, D.G.
Optimization by vector space methods.
John Wiley & Sons, Inc., New York-London-Sydney, 1969.

References: Books on numerical methods for PDEs

Brenner, S.C. and Scott, L.R.
The mathematical theory of finite element methods.
Springer, New York, 2008.

Ern, A. and Guermond, J.-L.
Theory and practice of finite elements.
Springer-Verlag, New York, 2004.

Braess, D.
Finite elements. Theory, fast solvers, and applications in elasticity theory.
Cambridge University Press, Cambridge, 2007.

Bartels, S.
Numerical methods for nonlinear partial differential equations.
Springer, Cham, 2015.

References: Applications and numerics of random PDE

I. Babuška, R. Tempone, and G. E. Zouraris.
Galerkin finite element approximations of stochastic elliptic partial differential equations.
SIAM J. Numer. Anal., 42 (2004), pp. 800–825.

I. Babuška, R. Tempone, and G. E. Zouraris.
Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation.
Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 1251–1294.

I. Babuška, F. Nobile, and R. Tempone.
A stochastic collocation method for elliptic partial differential equations with random input data.
SIAM Rev., 52 (2010), pp. 317–355.

L. Bonfiglio, P. Perdikaris, S. Brizzolara, and G. Karniadakis.
A multi-fidelity framework for investigating the performance of super-cavitating hydrofoils under uncertain flow conditions.
in 19th AIAA Non-Deterministic Approaches Conference, American Institute of Aeronautics and Astronautics, 2017.

L. J. Durlofsky and Y. Chen.
Uncertainty Quantification for Subsurface Flow Problems Using Coarse-Scale Models,
Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 163–202.

G. E. Karniadakis, C.-H. Su, D. Xiu, D. Lucor, C. Schwab, and R. A. Todor.
Generalized polynomial chaos solution for differential equations with random inputs.
Tech. Report 2005-01, Seminar for Applied Mathematics, ETH Zurich, Zurich, Switzerland, 2005.

B. Khoromskij and C. Schwab.
Tensor-structured galerkin approximation of parametric and stochastic elliptic PDEs.
SIAM J. Sci. Comput., 33 (2011), pp. 364–385.

References: Applications and numerics of random PDE

F. Nobile, R. Tempone, and C. G. Webster
An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data.
SIAM J. Numer. Anal., 46 (2008), pp. 2411–2442.

F. Nobile, R. Tempone, and C. G. Webster
A sparse grid stochastic collocation method for partial differential equations with random input data.
SIAM J. Numer. Anal., 46 (2008), pp. 2309–2345.

G. Stadler, M. Gurnis, C. Burstedde, L. C. Wilcox, L. Alisic, and O. Ghattas.
The dynamics of plate tectonics and mantle flow: From local to global scales.
Science, 329 (2010), pp. 1033–1038.

C. Schwab and C. J. Gittelson.
Sparse tensor discretizations of high-dimensional parametric and stochastic PDEs.
Acta Numerica, 20 (2011), pp. 291–467.

D. M. Tartakovsky, A. Guadagnini, and M. Riva.
Stochastic averaging of nonlinear flows in heterogeneous porous media.
Journal of Fluid Mechanics, 492 (2003), pp. 47–62.

D. Xiu and J. S. Hesthaven.
High-order collocation methods for differential equations with random inputs.
SIAM J. Sci. Comput., 27 (2005), pp. 1118–1139.

D. Xiu and G. E. Karniadakis.
Modeling uncertainty in flow simulations via generalized polynomial chaos.
J. Comput. Phys., 187 (2003), pp. 137–167.

References: Risk Neutral PDE-Constrained Optimization $\mathcal{R} = \mathbb{E}$

A. A. Ali, E. Ullmann, and M. Hinze,
Multilevel Monte Carlo analysis for optimal control of elliptic PDEs with random coefficients.
SIAM/ASA J. Uncertain. Quantif. 5 (2017), no. 1, 466492.

A. Borzì, V. Schulz, C. Schillings, and G. von Winckel.
On the treatment of distributed uncertainties in PDE-constrained optimization.
GAMM Mitteilungen, 33 (2010), pp. 230–246.

A. Borzì and G. von Winckel.
A POD framework to determine robust controls in PDE optimization.
Comput. Vis. Sci., 14 (2011), pp. 91–103.

P. Chen and A. Quarteroni.
Weighted reduced basis method for stochastic optimal control problems with elliptic PDE constraint
SIAM/ASA Journal on Uncertainty Quantification, 2 (2014), pp. 364–396.

P. Chen, A. Quarteroni, and G. Rozza.
Stochastic optimal robin boundary control problems of advection-dominated elliptic equations.
SIAM Journal on Numerical Analysis, 51 (2013), pp. 2700–2722.

M. H. Farshbaf-Shaker, R. Henrion, and D. Hömberg.
Properties of chance constraints in infinite dimensions with an application to PDE constrained optimization.
Set-Valued Var. Anal., 26(4):821–841, 2018.

S. Garreis and M. Ulbrich.
Constrained optimization with low-rank tensors and applications to parametric problems with PDEs.
SIAM J. Sci. Comput. 39 (2017), no. 1, A25A54.

References: Risk Neutral PDE-Constrained Optimization $\mathcal{R} = \mathbb{E}$

C. Geiersbach, C. and G. Ch. Pflug.
Projected stochastic gradients for convex constrained problems in Hilbert spaces.
SIAM J. Optim. 29 (2019), no. 3, 20792099.

C. Geiersbach and W. Wollner.
A stochastic gradient method with mesh refinement for PDE-constrained optimization under uncertainty.
SIAM J. Sci. Comput. 42 (2020), no. 5, A2750A2772.

C. Geiersbach and W. Wollner.
Optimality conditions for convex stochastic optimization problems in Banach spaces with almost sure state constraint.
WIAS Preprint No. 2755, DOI: 10.20347/WIAS.PREPRINT.2755, 2020.

M. Hoffhues, W. Römisch, T. M. Surowiec
On Quantitative Stability in Infinite-Dimensional Optimization under Uncertainty
Optimization Letters. `https://doi.org/10.1007/s11590-021-01707-2`

D. P. Kouri.
A multilevel stochastic collocation algorithm for optimization of PDEs with uncertain coefficients.
SIAM/ASA Journal on Uncertainty Quantification, 2 (2014), pp. 55–81.

D. P. Kouri, M. Heinkenschloss, D. Ridzal, and B. G. van Bloemen Waanders.
A trust-region algorithm with adaptive stochastic collocation for PDE optimization under uncertainty.
SIAM Journal on Scientific Computing, 35 (2013), pp. A1847–A1879.

W. Römisch and T. .M. Surowiec.
Asymptotic Properties of Monte Carlo Methods in Elliptic PDE-Constrained Optimization under Uncertainty
Preprint arXiv:2106.06347 `https://arxiv.org/abs/2106.06347`

References: Risk Neutral PDE-Constrained Optimization $\mathcal{R} = \mathbb{E}$

D. P. Kouri, M. Heinkenschloss, D. Ridzal, and B. G. van Bloemen Waanders.
Inexact objective function evaluations in a trust-region algorithm for PDE-constrained optimization under uncertainty.
SIAM Journal on Scientific Computing, 36 (2014), pp. A3011–A3029.

J. Milz
Sample average approximations of strongly convex stochastic programs in Hilbert spaces.
Preprint arXiv:2104.05114v1.

N. Petra, H. Zhu, G. Stadler, T. J. R. Hughes, and O. Ghattas.
An inexact Gauss-Newton method for inversion of basal sliding and rheology parameters in a nonlinear Stokes ice sheet model.
Journal of Glaciology, 58 (2012).

V. Schulz and C. Schillings.
On the nature and treatment of uncertainties in aerodynamic design.
AIAA Journal, 47 (2009), pp. 646–654.

H. Tiesler, R. M. Kirby, D. Xiu, and T. Preusser.
Stochastic collocation for optimal control problems with stochastic PDE constraints.
SIAM Journal on Control and Optimization, 50 (2012), pp. 2659–2682.

References: Risk Averse PDE-Constrained Optimization

A. Alexanderian, N. Petra, G. Stadler, and O. Ghattas.
Mean-variance risk-averse optimal control of systems governed by PDEs with random parameter fields using quadratic approximations.
SIAM/ASA J. Uncertainty Quantification, 5(1), (2017) pp. 1166–1192.

H. Antil, D. .P. Kouri, and J. Pfefferer
Risk-Averse Control of Fractional Diffusion with Uncertain Exponent.
SIAM J. Control Optim., 59(2), (2021) 11611187.

S. Garreis, T. M. Surowiec, and M. Ulbrich.
An interior-point approach for solving risk-averse PDE-constrained optimization problems with Coherent Risk Measures
To appear in SIAM Journal on Optimization

M. Heinkenschloss, B. Kramer, T. Takhtaganov, and K. Willcox.
Conditional-value-at-risk estimation via reduced-order models.
SIAM/ASA J. Uncertain. Quantif., 6 (2018), pp. 1395–1423.

O. Lass and S. Ulbrich.
Model order reduction techniques with a posteriori error control for nonlinear robust optimization governed by partial differential equations.
SIAM J. Sci. Comput., 39 (2017), pp. S112–S139.

J. Milz and M. Ulbrich.
An Approximation Scheme for Distributionally Robust PDE-Constrained Optimization.
Preprint No. IGDK-2020-09, Technische Universitt Mnchen, 2020.

P. Kolvenbach, O. Lass, and S. Ulbrich.
An approach for robust PDE-constrained optimization with application to shape optimization of electrical engines and of dynamic elastic structures under uncertainty.
Optim. Eng., 19 (2018), pp. 697–731.

References: Risk Averse PDE-Constrained Optimization

D. P. Kouri

A measure approximation for distributionally robust PDE-constrained optimization problems.

SIAM Journal on Numerical Analysis, 55 (2017), pp. 3147–3172.

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.

A primal-dual algorithm for risk minimization

To appear in Mathematical Programming Ser. A

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.

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.

Existence and optimality conditions for risk-averse PDE-constrained optimization

SIAM/ASA J. Uncertainty Quantification 6, 2 (2018), 787–815.

Z. Zou, D. P. Kouri, and W. Aquino.

An adaptive local reduced basis method for solving PDEs with uncertain inputs and evaluating risk.

Comput. Methods Appl. Mech. Engrg., 345 (2019), pp. 302–322.