SAND2018-4059C

Scalable Algorithms and Software for PDE-Constrained Optimization Under Uncertainty

Denis Ridzal and Drew Kouri

Center for Computing Research

Sandia National Laboratories

SIAM Uncertainty Quantification, 04-17-2018

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

Research sponsored by the DOE-NNSA ASC Program and the DOD DARPA-EQUiPS Program. SAND2018-0000 C.

Motivation

Optimization formulations

From expectation to risk and back

Inexact trust-region algorithms

Software

Numerical results

Motivation

Optimization formulations

From expectation to risk and back

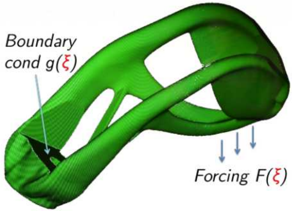
Inexact trust-region algorithms

Software

Numerical results

Topology Optimization for Structural Design

RISK-AVERSE DESIGN OF STRUCTURES



Setup: The forcing or load $F(\xi)$ on the right part of the bracket is uncertain. Additionally, there is an uncertain Dirichlet condition on the displacement at the bolt location, see $g(\xi)$.

Given volume fraction $V_0 \in (0,1)$, max compliance η, $\Omega \subset \mathbb{R}^3$,

$$\underset{0 \le z \le 1}{\text{Minimize}} \ \text{Prob}\left[\int_\Omega F(\xi,x)\cdot(U(z))(\xi,x)\,dx \ge \eta\right]$$

s.t. $\int_\Omega z(x)\,dx \le V_0|\Omega|$, where $U(z) = u : \Xi \to (H^1(\Omega))^3$ solves

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

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

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

- Uncertainty in external forces (loads) and boundary conditions.
- Reliability formulation of the compliance objective: Compute light-weight designs that reduce the probability of structural failure.

Optimal Control of Thermal Fluids

RISK-AVERSE CONTROL OF DYNAMICAL MULTIPHYSICS SYSTEMS

$$\underset{a(t) \le z(t) \le b(t)}{\text{Minimize}} \; \text{Risk}\left(\int_{\Omega_R} |\nabla \times (\mathbf{U}(z))(\xi, x, \tau^{\text{final}})|^2 dx \right)$$

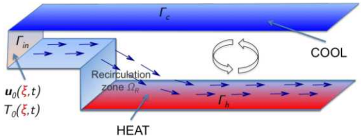
where $\mathbf{U}(z) = \mathbf{u} : \Xi \to (H^1(\Omega))^3 \times L^2([0, \tau^{\text{final}}])$ solves

$$\frac{\partial \mathbf{u}(\xi)}{\partial t} - \nu_1 \Delta \mathbf{u}(\xi) + (\mathbf{u}(\xi) \cdot \nabla)\mathbf{u}(\xi) + \nabla p(\xi) + \nu_2 T(\xi)\mathbf{g} = 0$$

$$\frac{\partial T(\xi)}{\partial t} - \nu_3 \Delta T(\xi) + \mathbf{u}(\xi) \cdot \nabla T(\xi) = 0$$

a.s., and $\nabla \cdot \mathbf{u}(\xi) = 0$, in $\Omega \times [0, \tau^{\text{final}}]$, with BCs $\mathbf{u} = \mathbf{u}_0(\xi, t)$ and $T = T_0(\xi, t)$ on $\Gamma_{\text{in}} \times [0, \tau^{\text{final}}]$, etc., and heat-flux control $z(t) = z$ satisfying

$$\frac{\partial T(\xi)}{\partial \mathbf{n}} = h(z - T(\xi)) \quad \text{on } \Gamma_h \text{ and } \Gamma_c.$$



Setup: The velocity field and temperature at the inlet Γ_{in} are time dependent and uncertain, see $\mathbf{u}_0(\xi, t)$ and $T_0(\xi, t)$. A time dependent temperature control $z(t)$ is applied on the top and bottom boundaries, Γ_c and Γ_h, to create thermal flow counteracting the vortex in the recirculation zone Ω_R.

- Uncertainty in the velocity field and temperature at the inlet.
- A thermal fluid system with time-dependent temperature control.

Challenges

- **Function-space** optimization of **PDEs with random inputs**.
- **Large-scale** numerical optimization.
- **Large-dimensional** spaces of **uncertain** parameters.
- **Risk functions**: mathematics, computational cost, usefulness.
- **Nonsmooth** objective functions and constraints.
- **Time consistency** for optimal control of dynamical systems.

Challenges

- **Function-space** optimization of **PDEs with random inputs**.
- **Large-scale** numerical optimization.
- **Large-dimensional** spaces of **uncertain** parameters.
- **Risk functions**: mathematics, computational cost, usefulness.
- **Nonsmooth** objective functions and constraints.
- **Time consistency** for optimal control of dynamical systems.

★ **Identify a computational core and discuss algorithms** . . .

Challenges

- **Function-space** optimization of **PDEs with random inputs**.
- **Large-scale** numerical optimization.
- **Large-dimensional** spaces of **uncertain** parameters.
- **Risk functions**: mathematics, computational cost, usefulness.
- **Nonsmooth** objective functions and constraints.
- **Time consistency** for optimal control of dynamical systems.

★ **Identify a computational core and discuss algorithms** ...
for nonlinear nonconvex constraints and objective functions.

Motivation

Optimization formulations

From expectation to risk and back

Inexact trust-region algorithms

Software

Numerical results

Full space and Reduced space

$$\min_{u \in U, z \in Z} \{\mathcal{R}(J_F(u, z; \xi)) + \wp(z)\}$$

subject to $c(u, z; \xi) = 0$ a.s.

$$\min_{z \in Z} \{\mathcal{R}(J_R(S(z; \xi), \xi)) + \wp(z)\}$$

a.s.:	almost surely
Probability Space:	$(\Omega, \mathcal{F}, \mathbb{P})$ where $\mathcal{F} \subseteq 2^\Omega$ and $\mathbb{P} : \mathcal{F} \to [0, 1]$
Uncertain Inputs:	$\xi : \Omega \to \Xi$ where $\Xi := \xi(\Omega) \subseteq \mathbb{R}^m$
Random Variable Space:	$\mathcal{X}$ is a space of random variables
Deterministic State Space:	U is a Hilbert space
Control Space:	Z is a Hilbert space (deterministic)
Numerical Surrogate:	$\mathcal{R} : \mathcal{X} \to \mathbb{R} \cup \{\infty\}$
Uncertain Objectives:	$J_R : U \times \Xi \to \mathcal{X}$; $J_F : U \times Z \times \Xi \to \mathcal{X}$
Random Field PDE Solution:	$S : Z \times \Xi \to U$ with $c(S(z), z; \xi) = 0$ a.s.
Control Penalty:	$\wp : Z \to \mathbb{R} \cup \{\infty\}$

Motivation

Optimization formulations

From expectation to risk and back

Inexact trust-region algorithms

Software

Numerical results

Modeling Risk

What is risk? . . .

In our optimization problems, $J_R(S(z;\xi),\xi)$ and $J_F(u,z;\xi)$ are **risks**!

We **cannot** directly minimize J_R and J_F. Range space: random variables!

We must **quantify**, i.e., **scalarize risk**.

- **Risk-Neutral Stochastic Programming:** Minimize *on average*

$$\mathcal{R}(X) = \mathbb{E}[X].$$

- **Risk-Averse Stochastic Programming:** Model *risk preference*

$$\mathcal{R}(X) = \mathbb{E}[X] + c\mathbb{E}[(X - \mathbb{E}[X])_+^p]^{1/p}.$$

- **Probabilistic Optimization:** Minimize the *probability of loss*

$$\mathcal{R}(X) = \mathbb{P}(X > \tau).$$

- **Stochastic Orders:** Model risk preference with a *benchmark Y*

$$\mathbb{P}(X \leq x) \leq \mathbb{P}(Y \leq x) \quad \forall x \in \mathbb{R}.$$

Measuring Risk

Optimized Certainty Equivalents (Ben Tal & Teboulle, 2007, Math. Fin.)

$$\mathcal{R}(X) = \inf_{t\in\mathbb{R}}\{t + \mathbb{E}[v(X-t)]\}$$

where $v : \mathbb{R} \to \mathbb{R}$ is a convex **regret** function that satisfies

$$v(0) = 0, \quad v(x) > x \quad \forall\, x \neq 0.$$

A functional $\mathcal{H} : \mathcal{X} \to \mathbb{R} \cup \{\infty\}$ is a **coherent** measure of risk if

- **Subadditivity:** $\mathcal{H}(X + X') \leq \mathcal{H}(X) + \mathcal{H}(X')$.
- **Monotonicity:** $X \geq X'$ a.e. $\implies \mathcal{H}(X) \geq \mathcal{H}(X')$.
- **Translation Equivariance:** $\mathcal{H}(X + t) = \mathcal{H}(X) + t, \quad \forall t \in \mathbb{R}$.
- **Positive Homogeneity:** $\mathcal{H}(tX) = t\mathcal{H}(X), \quad \forall t > 0$.

Properties: $\mathcal{R}$ is convex and translation equivariant.

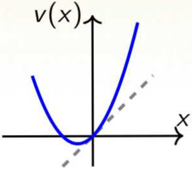
$\mathcal{R}$ is **positive homogeneous** $\iff$ v is piecewise linear with kink at 0

$\mathcal{R}$ is **monotonic** $\iff$ v is nondecreasing

Risk Measure Examples

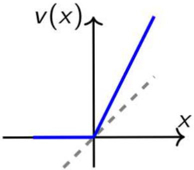
Mean-Plus-Variance: Not monotonic

$$\mathcal{R}(X) = \mathbb{E}[X] + c\mathbb{E}[|X - \mathbb{E}[X]|^2]$$
$$v(x) = x + cx^2$$



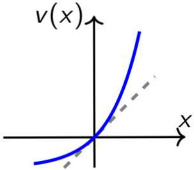
Conditional value-at-risk: Coherent, not diff'ble

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



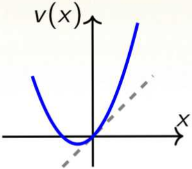
Entropic Risk: Not coherent, derivative not Lipschitz

$$\mathcal{R}(X) = c^{-1} \log \mathbb{E}[\exp(cX)]$$
$$v(x) = (\exp(cx) - 1)/c$$



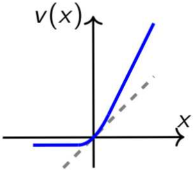
Risk Measure Examples

Mean-Plus-Variance: **Not monotonic**



$$\mathcal{R}(X) = \mathbb{E}[X] + c\mathbb{E}[|X - \mathbb{E}[X]|^2]$$
$$v(x) = x + cx^2$$

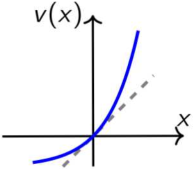
Smoothed CVaR: **Not positive homogeneous**



$$\mathcal{R}(X) = \inf_{t\in\mathbb{R}} \{t + \mathbb{E}[v(X - t)]\}$$

$$v(x) = \begin{cases} -\frac{\varepsilon}{2} & \text{if } x \le -\varepsilon \\ \frac{1}{2\varepsilon}x^2 + x & \text{if } x \in \left(-\varepsilon, \frac{\varepsilon\beta}{1-\beta}\right) \\ \frac{1}{1-\beta}\left(x - \frac{\varepsilon\beta^2}{2(1-\beta)}\right) & \text{if } x \ge \frac{\varepsilon\beta}{1-\beta} \end{cases}$$

Entropic Risk: **Not coherent, derivative not Lipschitz**



$$\mathcal{R}(X) = c^{-1} \log \mathbb{E}[\exp(cX)]$$
$$v(x) = (\exp(cx) - 1)/c$$

Minimizing Expectation

Minimizing Expectation

. . . **is at the core of minimizing many useful risk measures.**

Minimizing Expectation

... is at the core of minimizing many useful risk measures.

... and key for minimizing smooth approximations of nonsmooth risk measures.

Minimizing Expectation

... is at the core of minimizing many useful risk measures.

... and key for minimizing smooth approximations of nonsmooth risk measures.

... and so for scalable computations we consider:

$$\min_{u\in U, z\in Z} \{\mathbb{E}(J_F(u,z;\xi)) + \wp(z)\}$$
$$\text{subject to } \; c(u,z;\xi) = 0 \; \text{ a.s.}$$

$$\min_{z\in Z} \{\mathbb{E}(J_R(S(z;\xi),\xi)) + \wp(z)\}$$

Minimizing Expectation

... is at the core of minimizing many useful risk measures.

... and key for minimizing smooth approximations of nonsmooth risk measures.

... and so for scalable computations we consider:

$$\min_{u \in U, z \in Z} \{\mathbb{E}(J_F(u, z; \xi)) + \wp(z)\}$$
$$\text{subject to } c(u, z; \xi) = 0 \text{ a.s.}$$

$$\min_{z \in Z} \{\mathbb{E}(J_R(S(z; \xi), \xi)) + \wp(z)\}$$

... and generalize to solving smooth NLPs:

$$\min_{x \in X := \{U \times Z \times \Xi\}} J(x)$$
$$\text{subject to } c(x) = 0$$

CONSTRAINED MINIMIZATION

$$\min_{z \in Z} \mathcal{J}(z)$$

UNCONSTRAINED MINIMIZATION

Motivation

Optimization formulations

From expectation to risk and back

Inexact trust-region algorithms

Software

Numerical results

Trust Regions for Unconstrained NLPs

Given: z_0, $m_0(s) \approx \mathcal{J}(z_0 + s)$, $\mathcal{J}_0 \approx \mathcal{J}$, $\Delta_0 \geq 0$, and $\mathtt{gtol} > 0$.
While $\|\nabla m_k(s)\|_Z > \mathtt{gtol}$

1. **Model Update:** Choose a new $m_k(s) \approx \mathcal{J}(z_k + s)$. $\leftarrow$ **Adaptivity**
2. **Step Computation:** Approximate a solution, s_k, to the subproblem

$$\min_{s \in Z} \; m_k(s) \quad \text{subject to} \quad \|s\|_Z \leq \Delta_k.$$

3. **Objective Update:** Choose a new $\mathcal{J}_k(z) \approx \mathcal{J}(z)$. $\leftarrow$ **Adaptivity**
4. **Step Acceptance:** Compute

$$\rho_k = \frac{\mathcal{J}_k(z_k) - \mathcal{J}_k(z_k + s_k)}{m_k(0) - m_k(s_k)}.$$

   If $\rho_k \geq \eta \in (0,1)$, then $z_{k+1} = z_k + s_k$ else $z_{k+1} = z_k$.

5. **Trust Region Update:** Choose a new trust region radius, Δ_{k+1}.

EndWhile

Inexact Gradients and Objective Functions

Kouri, Heinkenschloss, Ridzal, van Bloemen Waanders, 2014, SISC

Inexact Gradients

There exists $c > 0$ independent of k such that

$$\|\nabla m_k(0) - \nabla \mathcal{J}(z_k)\|_Z \leq c \min\{\|\nabla m_k(0)\|_Z, \Delta_k\}$$

(Carter 1989, Heinkenschloss and Vicente 2001).

Inexact Objective Functions

There exists $K > 0$, $\omega \in (0,1)$, and $\theta_r(z,s) \to 0$ as $r \to 0$ such that

$$|(\mathcal{J}(z_k) - \mathcal{J}(z_k + s_k)) - (\mathcal{J}_k(z_k) - \mathcal{J}_k(z_k + s_k))| \leq K\theta_{r_k}(z_k, s_k)$$

$$\theta_{r_k}(z_k, s_k)^{\omega} \leq \eta \min\{(m_k(0) - m_k(s_k)), r_k\}.$$

Here, $\eta > 0$ is tied to algorithmic parameters and $\lim_{k\to\infty} r_k = 0$.
(Carter 1989, Ziems and Ulbrich 2013).

Inexact Gradients and Objective Functions

Kouri, Heinkenschloss, Ridzal, van Bloemen Waanders, 2014, SISC

Inexact Gradients

There exists $c > 0$ independent of k such that

$$\|\nabla m_k(0) - \nabla \mathcal{J}(z_k)\|_Z \leq c \min\{\|\nabla m_k(0)\|_Z, \Delta_k\}$$

(Carter 1989, Heinkenschloss and Vicente 2001).

Inexact Objective Functions

There exists $K > 0$, $\omega \in (0,1)$, and $\theta_r(z,s) \to 0$ as $r \to 0$ such that

$$|(\mathcal{J}(z_k) - \mathcal{J}(z_k + s_k)) - (\mathcal{J}_k(z_k) - \mathcal{J}_k(z_k + s_k))| \leq K\theta_{r_k}(z_k, s_k)$$

$$\theta_{r_k}(z_k, s_k)^\omega \leq \eta \min\{(m_k(0) - m_k(s_k)), r_k\}.$$

Here, $\eta > 0$ is tied to algorithmic parameters and $\lim_{k\to\infty} r_k = 0$.
(Carter 1989, Ziems and Ulbrich 2013).

- **Cannot** compute $\mathcal{J}(z_k)$ and $\nabla \mathcal{J}(z_k)$.

Inexact Gradients and Objective Functions

Kouri, Heinkenschloss, Ridzal, van Bloemen Waanders, 2014, SISC

Inexact Gradients

There exists $c > 0$ independent of k such that

$$\|\nabla m_k(0) - \nabla \mathcal{J}(z_k)\|_Z \leq c \min\{\|\nabla m_k(0)\|_Z, \Delta_k\}$$

(Carter 1989, Heinkenschloss and Vicente 2001).

Inexact Objective Functions

There exists $K > 0$, $\omega \in (0,1)$, and $\theta_r(z,s) \to 0$ as $r \to 0$ such that

$$|(\mathcal{J}(z_k) - \mathcal{J}(z_k + s_k)) - (\mathcal{J}_k(z_k) - \mathcal{J}_k(z_k + s_k))| \leq K\theta_{r_k}(z_k, s_k)$$

$$\theta_{r_k}(z_k, s_k)^\omega \leq \eta \min\{(m_k(0) - m_k(s_k)), r_k\}.$$

Here, $\eta > 0$ is tied to algorithmic parameters and $\lim_{k\to\infty} r_k = 0$.
(Carter 1989, Ziems and Ulbrich 2013).

- **Cannot** compute $\mathcal{J}(z_k)$ and $\nabla \mathcal{J}(z_k)$.
- Control errors using **dimension-adaptive sparse grids**, by estimating them on a forward neighborhood of a current index set.

Adaptive sparse grids for the gradient

- Admissible index set $\mathcal{I}_k^{\mathrm{g}} \subset \mathbb{N}^m$ and quadrature approximation of $\mathcal{J}(z) = \mathbb{E}[\, J_R(z;\xi)\,]$, i.e., $\mathcal{J}_{\mathcal{I}_k^{\mathrm{g}}}(z) = \sum_{\mathbf{i}\in\mathcal{I}_k^{\mathrm{g}}} (\delta_1^{i_1} \otimes \cdots \otimes \delta_m^{i_m})[\, J_R(z;\xi)\,]$.
- $\left\| \sum_{\mathbf{i}\notin\mathcal{I}_k^{\mathrm{g}}} (\delta_1^{i_1} \otimes \cdots \otimes \delta_M^{i_M})[\nabla J_R(z;\xi)] \right\|_Z \leq c \min\left\{ \|\nabla \mathcal{J}_{\mathcal{I}_k^{\mathrm{g}}}(z_k)\|_Z, \Delta_k \right\}$

Initialization: Set $\mathbf{i} = (1,\dots,1)$, $\mathcal{A} = \{\mathbf{i}\}$, $\mathcal{O} = \emptyset$, $g_{\mathbf{i}} = (\delta_1^{i_1} \otimes \cdots \otimes \delta_M^{i_M})[\nabla J_R(z_k;\xi)]$ and $\beta = \beta_{\mathbf{i}} = \|g_{\mathbf{i}}\|_{\mathcal{Z}}$, $g = g_{\mathbf{i}}$, and $\mathtt{TOL} = c \min\{\|g\|_Z, \Delta_k\}$.

While $\beta > \mathtt{TOL}$

1. Select $\mathbf{i} \in \mathcal{A}$ corresponding to the largest $\beta_{\mathbf{i}}$
2. Set $\mathcal{A} \leftarrow \mathcal{A} \setminus \{\mathbf{i}\}$ and $\mathcal{O} \leftarrow \mathcal{O} \cup \{\mathbf{i}\}$
3. Update the error indicator $\beta \leftarrow \beta - \beta_{\mathbf{i}}$
4. **For** $\ell = 1,\dots,m$
   1. Set $\mathbf{j} = \mathbf{i} + \mathbf{e}_\ell$
   2. **If** $\mathcal{O} \cup \{\mathbf{j}\}$ is admissible
      1. Set $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mathbf{j}\}$
      2. Set $g_{\mathbf{j}} = (\delta_1^{j_1} \otimes \cdots \otimes \delta_m^{j_m})[\nabla J_R(z_k;\xi)]$
      3. Set $\beta_{\mathbf{j}} = \|g_{\mathbf{j}}\|_{\mathcal{Z}}$
      4. Update the gradient approximation $g \leftarrow g + g_{\mathbf{j}}$
      5. Update the error indicator $\beta \leftarrow \beta + \beta_{\mathbf{j}}$
      6. Update the stopping tolerance $\mathtt{TOL} = c \min\{\|g\|_Z, \Delta_k\}$
   3. **EndIf**
5. **EndFor**

EndWhile

Set $\mathcal{I}_k^{\mathrm{g}} = \mathcal{A} \cup \mathcal{O}$ and $\nabla m_k(0) = g$.

Adaptive sparse grids for the objective

- $\left|\sum_{\mathbf{i}\notin\mathcal{I}_k^{\circ}}(\delta_1^{i_1}\otimes\cdots\otimes\delta_m^{i_m})[J_R(z_k+s_k;\xi)-J_R(z_k;\xi)]\right| = \theta_k$

Intialization: Set $\mathbf{i}=(1,\ldots,1)$, $\mathcal{A}=\{\mathbf{i}\}$, $\mathcal{O}=\emptyset$, $\mathtt{TOL}=(\eta\min\{\mathrm{pred}_k, r_k\})^{1/\omega}$, $\tilde{\theta}_k=\vartheta_{\mathbf{i}}=(\delta_1^{i_1}\otimes\cdots\otimes\delta_M^{i_M})[J_R(z_k+s_k;\xi)-J_R(z_k;\xi)]$ and $cred_k=\vartheta_{\mathbf{i}}$.

While $|\tilde{\theta}_k|>\mathtt{TOL}$

1. Select $\mathbf{i}\in\mathcal{A}$ corresponding to the largest $|\vartheta_{\mathbf{i}}|$
2. Set $\mathcal{A}\leftarrow\mathcal{A}\setminus\{\mathbf{i}\}$ and $\mathcal{O}\leftarrow\mathcal{O}\cup\{\mathbf{i}\}$
3. Update the error indicator $\tilde{\theta}_k\leftarrow\tilde{\theta}_k-\vartheta_{\mathbf{i}}$
4. **For** $\ell=1,\ldots,m$
    1. Set $\mathbf{j}=\mathbf{i}+\mathbf{e}_\ell$
    2. **If** $\mathcal{O}\cup\{\mathbf{j}\}$ is admissible
        1. Set $\mathcal{A}\leftarrow\mathcal{A}\cup\{\mathbf{j}\}$
        2. Set $\vartheta_{\mathbf{j}}=(\delta_1^{j_1}\otimes\cdots\otimes\delta_m^{j_m})[J_R(z_k+s_k;\xi)-J_R(z_k;\xi)]$
        3. Update the computed reduction $cred_k\leftarrow cred_k+\vartheta_{\mathbf{j}}$
        4. Update the error indicator $\tilde{\theta}_k\leftarrow\tilde{\theta}_k+\vartheta_{\mathbf{j}}$
    3. **EndIf**
5. **EndFor**

EndWhile

Return $\mathcal{I}_k^{\circ}=\mathcal{A}\cup\mathcal{O}$ and $cred_k$.

Trust Regions for Constrained NLPs

Solve equality-constrained optimization problem:

$$\min_{x \in X} \; J(x) \qquad \text{subject to} \qquad c(x) = 0$$

Define Lagrangian functional $L : X \times C \to \mathbb{R}$:

$$L(x, \lambda) = J(x) + \langle \lambda, c(x) \rangle_C$$

If *regular* point x_* is a local solution of the NLP, then there exists a $\lambda_* \in C$ satisfying the *first-order necessary optimality conditions*:

$$\begin{aligned} \nabla_x J(x_*) + c_x(x_*)^* \lambda_* &= 0 \\ c(x_*) &= 0 \end{aligned}$$

Solve a sequence of *nonconvex* quadratic trust-region subproblems:

$$\begin{aligned} \min_{s \in X} \quad & \frac{1}{2} \langle \nabla_{xx} L(x_k, \lambda_k) s, s \rangle_X + \langle \nabla_x L(x_k, \lambda_k), s \rangle_X + L(x_k, \lambda_k) \\ \text{s.t.} \quad & c_x(x_k) s + c(x_k) = 0 \,, \quad \|s\|_X \le \Delta_k \end{aligned}$$

Possible incompatibility of constraints: composite-step approach.

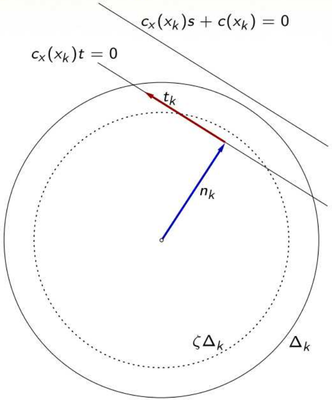
Composite-step Approach for the Solution of the Trust-region Subproblem

- **Trust-region step**:
  $s_k = n_k + t_k$

- **Quasi-normal step** n_k:
  reduces linear infeasibility

$$\min_{n\in X} \quad \|c_x(x_k)n + c(x_k)\|_C^2$$
$$\text{s.t.} \quad \|n\|_X \le \zeta\Delta_k$$

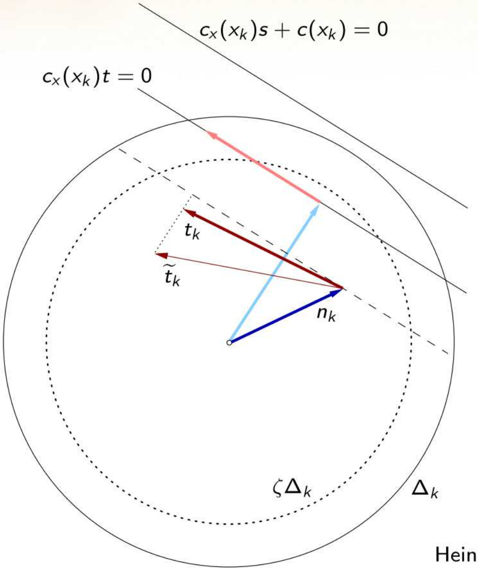
- **Tangential step** t_k:
  improves optimality while staying in the null space of the linearized constraints



$$\min_{t\in X} \quad \frac{1}{2}\langle \nabla_{xx}L(x_k,\lambda_k)(t+n_k), t+n_k\rangle_X + \langle \nabla_x L(x_k,\lambda_k), t+n_k\rangle_X + L(x_k,\lambda_k)$$
$$\text{s.t.} \quad c_x(x_k)t = 0, \quad \|t+n_k\|_X \le \Delta_k$$

Omojokun (1989), Byrd, Hribar, Nocedal (1997), Dennis, El-Alem, Maciel (1997)

Matrix-free Composite-step Algorithm

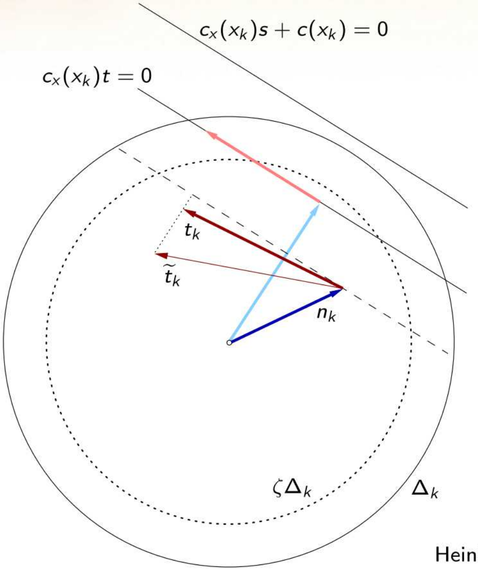


Composite step: $s_k = n_k + t_k$

1. Compute quasi-normal step n_k using **Powell's dogleg method**, satisfying inexactness conditions.
2. Solve tangential subproblem for $\widetilde{t}_k$ with **projected Steihaug-Toint CG**, satisfying inexactness conditions.
3. Restore linearized feasibility, yielding tangential step t_k.
4. Update Lagrange multipliers λ_{k+1}.
5. Evaluate progress.

Heinkenschloss, Ridzal, *SIAM J. Opt. (2014)*

Matrix-free Composite-step Algorithm



Composite step: $s_k = n_k + t_k$

1. Compute quasi-normal step n_k using **Powell's dogleg method**, satisfying inexactness conditions.
2. Solve tangential subproblem for $\widetilde{t}_k$ with **projected Steihaug-Toint CG**, satisfying inexactness conditions.
3. Restore linearized feasibility, yielding tangential step t_k.
4. Update Lagrange multipliers λ_{k+1}.
5. Evaluate progress.

Inexactness is interpreted in the context of linear systems, and their iterative solution.

Heinkenschloss, Ridzal, *SIAM J. Opt. (2014)*

Linear Systems

... are all augmented constraint systems

$$\begin{pmatrix} I & c_x(x_k)^* \\ c_x(x_k) & 0 \end{pmatrix} \begin{pmatrix} y^1 \\ y^2 \end{pmatrix} = \begin{pmatrix} b^1 \\ b^2 \end{pmatrix} + \begin{pmatrix} e^1 \\ e^2 \end{pmatrix}$$

- The size of $(e^1 \; e^2)$ is governed by various model reduction conditions, i.e., the progress of the optimization algorithm:

$$\|e^1\|_X + \|e^2\|_C \leq \text{func}\left(\|b^1\|_X, \|b^2\|_C, \|y^1\|_X, \Delta_k, \xi\right)$$

- **Preconditioning** for $X = U \times Z$ with $c_x(x_k) = C_x = [C_u \;\; C_z]$:

$$\begin{pmatrix} I_U & 0 & C_u^* \\ 0 & I_Z & C_z^* \\ C_u & C_z & 0 \end{pmatrix} \longrightarrow \mathcal{P} = \begin{pmatrix} I_U & 0 & 0 \\ 0 & I_Z & 0 \\ 0 & 0 & C_u^{-*} C_u^{-1} \end{pmatrix}$$

Semi-discretization for the Expected Value

- We approximate $\mathbb{E}[X] = \int_\Xi \rho(\xi) X(\xi) \mathrm{d}\xi$ with the quadrature $\mathbb{E}_Q$,

$$\mathbb{E}_Q[X] = \sum_{k=1}^{Q} w_k X(\xi_k), \text{ with points } \{\xi_1, ..., \xi_Q\}, \text{ weights } \{w_1, ..., w_Q\}.$$

- Our semi-discrete optimization problem is

$$\underset{u_1,\dots,u_Q \in U,\, z \in Z}{\text{minimize}} \sum_{k=1}^{Q} w_k J_F(u_k, z) \quad \text{subject to} \quad c(u_k, z; \xi_k) = 0,\ k = 1, ..., Q$$

- Family of possible Lagrangians $L_Q : U^Q \times Z \times U^Q \to \mathbb{R} \cup \{\infty\}$,

$$L(u_1, ..., u_Q, z, \lambda_1, ..., \lambda_Q) = \sum_{k=1}^{Q} w_k J_F(u_k, z) + \sum_{k=1}^{Q} v_k \langle \lambda_k, c(u_k, z; \xi_k) \rangle_C,$$

governed by the choice of weights $\{v_k\}_{k=1}^{Q}$. Two choices are:

— $v_k = 1$, $k = 1, ..., Q$, for a "finite-dimensional" view; and

— $v_k = w_k$, $k = 1, ..., Q$, for an "infinite-dimensional" view.

Augmented System for Minimizing Expectation

$$\begin{pmatrix}
I_U & 0 & \dots & 0 & 0 & v_1(C_u^1)^* & 0 & \dots & 0 \\
0 & I_U & \dots & 0 & 0 & 0 & v_2(C_u^2)^* & \dots & 0 \\
\vdots & \vdots & \ddots & 0 & \vdots & \vdots & \vdots & \ddots & 0 \\
0 & 0 & \dots & I_U & 0 & 0 & 0 & \dots & v_Q(C_u^Q)^* \\
0 & 0 & \dots & 0 & I_Z & v_1(C_z^1)^* & v_2(C_z^2)^* & \dots & v_Q(C_z^Q)^* \\
v_1 C_u^1 & 0 & \dots & 0 & v_1 C_z^1 & 0 & 0 & \dots & 0 \\
0 & v_2 C_u^2 & \dots & 0 & v_2 C_z^2 & 0 & 0 & \dots & 0 \\
\vdots & \vdots & \ddots & 0 & \vdots & \vdots & \vdots & \ddots & 0 \\
0 & 0 & \dots & v_Q C_u^Q & v_Q C_z^Q & 0 & 0 & \dots & 0
\end{pmatrix}$$

Augmented System for Minimizing Expectation

$$
\begin{pmatrix}
I_U & 0 & \dots & 0 & 0 & v_1(C_u^1)^* & 0 & \dots & 0 \\
0 & I_U & \dots & 0 & 0 & 0 & v_2(C_u^2)^* & \dots & 0 \\
\vdots & \vdots & \ddots & 0 & \vdots & \vdots & \vdots & \ddots & 0 \\
0 & 0 & \dots & I_U & 0 & 0 & 0 & \dots & v_Q(C_u^Q)^* \\
0 & 0 & \dots & 0 & I_Z & v_1(C_z^1)^* & v_2(C_z^2)^* & \dots & v_Q(C_z^Q)^* \\
v_1 C_u^1 & 0 & \dots & 0 & v_1 C_z^1 & 0 & 0 & \dots & 0 \\
0 & v_2 C_u^2 & \dots & 0 & v_2 C_z^2 & 0 & 0 & \dots & 0 \\
\vdots & \vdots & \ddots & 0 & \vdots & \vdots & \vdots & \ddots & 0 \\
0 & 0 & \dots & v_Q C_u^Q & v_Q C_z^Q & 0 & 0 & \dots & 0
\end{pmatrix}
$$

- **Applying the preconditioner $\mathcal{P}$ to this system is scalable, due to the parallelization in block-diagonal operators C_u and $(C_u)^*$.**
- **Inverses of each C_u^i and $(C_u^i)^*$ are applied *very* inexactly.**

Motivation

Optimization formulations

From expectation to risk and back

Inexact trust-region algorithms

Software

Numerical results

Rapid Optimization Library

- Trilinos package for **matrix-free nonlinear optimization**: unconstrained, equality-constrained, inequality-constrained, line searches, trust regions, ...

Rapid Optimization Library

- Trilinos package for **matrix-free nonlinear optimization**: unconstrained, equality-constrained, inequality-constrained, line searches, trust regions, ...
- Adjoint interface for **simulation-based optimization** – **SimOpt**.

Rapid Optimization Library

ROL

- Trilinos package for **matrix-free nonlinear optimization**: unconstrained, equality-constrained, inequality-constrained, line searches, trust regions, ...
- Adjoint interface for **simulation-based optimization** – **SimOpt**.
- Built from ground-up for **optimization under uncertainty**:
  - **Risk modeling** based on various concepts: risk neutrality, risk aversion, buffered probability, stochastic orders.
  - Utilities for **statistical estimation**; concept of **risk quadrangle**.
  - **Bring-your-own sampler**, in addition to the default Monte Carlo.
  - Interface to **Dakota** for sparse grids.

Rapid Optimization Library

ROL

- Trilinos package for **matrix-free nonlinear optimization**: unconstrained, equality-constrained, inequality-constrained, line searches, trust regions, ...
- Adjoint interface for **simulation-based optimization** – **SimOpt**.
- Built from ground-up for **optimization under uncertainty**:
  - **Risk modeling** based on various concepts: risk neutrality, risk aversion, buffered probability, stochastic orders.
  - Utilities for **statistical estimation**; concept of **risk quadrangle**.
  - **Bring-your-own sampler**, in addition to the default Monte Carlo.
  - Interface to **Dakota** for sparse grids.
- **If you have a (parametrized) deterministic optimization ready, turning it into stochastic optimization amounts to flipping a few switches in the ROL options file.**

Motivation

Optimization formulations

From expectation to risk and back

Inexact trust-region algorithms

Software

Numerical results

Optimizing CVD Reactors under Uncertainty

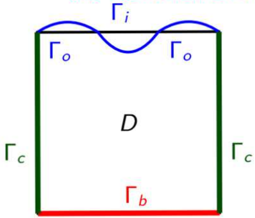
$$\min_{z\in Z} \frac{1}{2}\mathbb{E}\left[\int_D (\nabla \times u(z))\,\mathrm{d}x\right] + \frac{\gamma}{2}\int_{\Gamma_c} |z|^2\,\mathrm{d}x$$

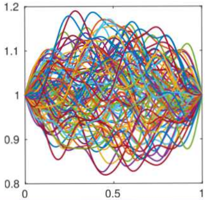
where $S(z) = (u(z), p(z), T(z))$ solves the Boussinesq flow equations,

$$\begin{aligned}
-\nu(\xi)\nabla^2 u + (u\cdot\nabla)u + \nabla p + \eta(\xi)Tg &= 0, && && \text{in } D, \text{ a.s.},\\
\nabla\cdot u &= 0, && && \text{in } D, \text{ a.s.},\\
-\kappa(\xi)\Delta T + u\cdot\nabla T &= 0, && && \text{in } D, \text{ a.s.},\\
u - u_i &= 0, && T = 0, && \text{on } \Gamma_i, \text{ a.s.},\\
u - u_o &= 0, && \kappa(\xi)\tfrac{\partial T}{\partial n} = 0, && \text{on } \Gamma_o, \text{ a.s.},\\
u &= 0, && T = T_b(\xi), && \text{on } \Gamma_b, \text{ a.s.},\\
u &= 0, && \kappa(\xi)\tfrac{\partial T}{\partial n} + h(\xi)(z - T) = 0, && \text{on } \Gamma_c, \text{ a.s.},
\end{aligned}$$

where $\Gamma_i = [1/3, 2/3]\times\{1\}$, $\Gamma_o = ([0,1/3]\cup[2/3,1])\times\{1\}$, $\Gamma_b = [0,1]\times\{0\}$ and $\Gamma_c = \{0,1\}\times[0,1]$. The inflow and outflow velocities, u_i and u_o are deterministic while the coefficients ν, η, κ, h and T_b are uncertain.

Comput. domain D.





Scenarios of T_b (Brownian bridge-like).

Optimizing CVD Reactors under Uncertainty

- **Uncertain variables:**

$$\nu = \frac{1}{\mathrm{Re}} = \frac{100}{1+0.01\xi_{N+1}}, \quad \eta = \frac{\mathrm{Gr}}{\mathrm{Re}^2} = 0.72\frac{1+0.01\xi_{N+1}}{1+0.01\xi_{N+2}},$$
$$\text{and} \quad \kappa = \frac{1}{\mathrm{Re}\,\mathrm{Pr}} = 10^5\frac{1+0.01\xi_{N+3}}{(1+0.01\xi_{N+1})^2}$$

  where Re is the Reynolds number, Gr is the Grashof number and Pr is the Prandtl number.

- The offset N is the total number of random variables associated with T_b and h. The uncertainty in T_b is modeled by the expansion

$$T_b(x,\xi) = 1 + 0.025\sum_{k=1}^{n_b}\xi_k\frac{\sqrt{2}\sin(\pi kx)}{\pi k}.$$

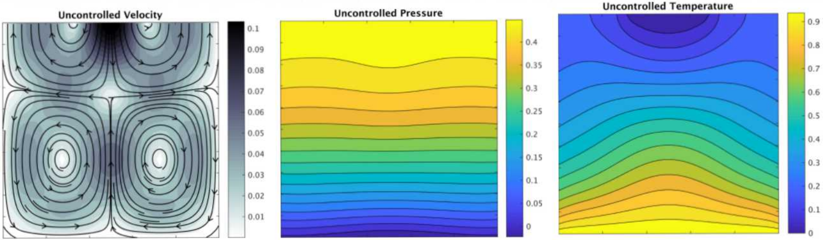
  The coefficient h has a similar expansion for $x=0$ and for $x=1$ with n_ℓ and n_r terms respectively. **All ξ_k are uniformly distributed on $[-1,1]$.**

- The curves on the top of the computational domain schematic are the inflow and outflow profiles of the velocity, given by

$$u(x) = \begin{cases} 2\left(\frac{1}{3}-x\right)x & \text{if } 0 \le x \le \frac{1}{3} \\ -4\left(x-\frac{1}{3}\right)\left(\frac{2}{3}-x\right) & \text{if } \frac{1}{3} < x < \frac{2}{3} \\ 2\left(x-\frac{2}{3}\right)(1-x) & \text{if } \frac{2}{3} \le x \le 1 \end{cases}.$$

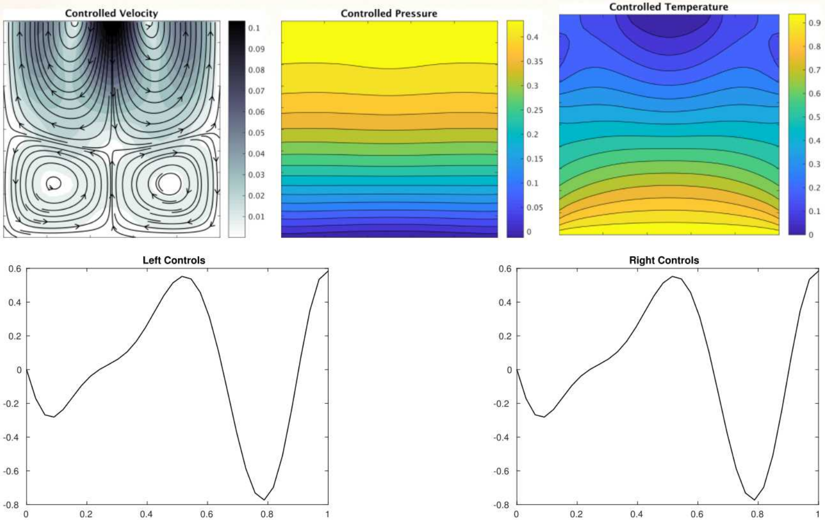
Uncontrolled State

Expected values of the uncontrolled velocity field (left), pressure (middle) and temperature (right):



- Finite elements on a uniform mesh of 33×33 quadrilaterals.
- For the velocity and pressure, Q2-Q1 Taylor-Hood finite element pair; for the temperature, Q2 finite element.
- Sparse grids built on one-dimensional Clenshaw-Curtis quadrature.
- Maximum sparse grid: level-3 isotropic Clenshaw-Curtis, $Q = 2245$ points.
- Implemented in ROL, PDE-OPT Application Development Kit.

Reduced Space: Sparse-grid Adaptivity



Optimal controls along the left vertical side wall (left image) and the right vertical side wall (right image) of the problem domain D.

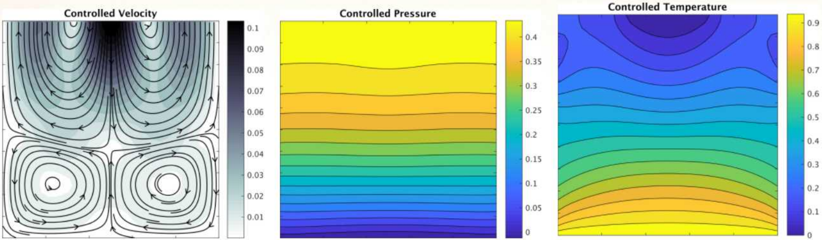
Reduced Space: Sparse-grid Adaptivity

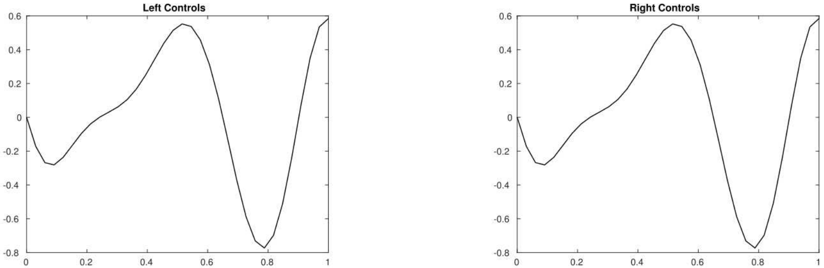
iter	$\mathcal{J}_k(z_k)$	$\lVert\nabla m_k(0)\rVert_Z$	$\lVert s_k\rVert_Z$	Δ_k	cg	accept	obj	grad
0	0.07457916	5.063×10^{-2}	—	10.000	—	—	1	3
1	0.07469930	5.063×10^{-2}	10.000	1.445	1	0	3	3
2	0.07469930	5.063×10^{-2}	1.445	0.361	1	0	3	3
3	0.05636707	4.875×10^{-2}	0.361	0.903	1	1	3	3
4	0.05636707	4.875×10^{-2}	0.903	0.226	1	0	3	3
5	0.04757099	2.059×10^{-2}	0.226	0.226	1	1	3	3
6	0.04680338	1.143×10^{-2}	0.226	0.226	2	1	103	117
7	0.04611002	3.468×10^{-3}	0.226	0.564	2	1	139	195
8	0.04511802	3.255×10^{-3}	0.564	1.411	2	1	117	233
9	0.04494516	1.085×10^{-3}	1.411	3.527	3	1	229	579
10	0.04499733	2.331×10^{-4}	2.838	8.818	6	1	579	949
11	0.04499338	6.211×10^{-5}	0.967	22.045	7	1	2245	1219
12	0.04499329	1.002×10^{-6}	0.127	55.113	8	1	2245	2245
13	0.04499327	7.034×10^{-9}	0.072	137.784	11	1	2245	2245

Iteration history for reduced-space adaptive sparse-grid approach.

- Sequential solution, direct linear solver, single computational core.

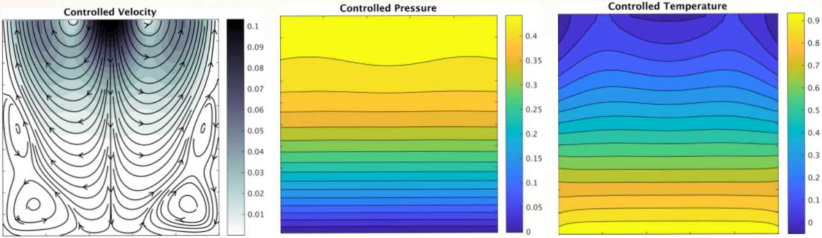
Reduced Space: Sparse-grid Adaptivity

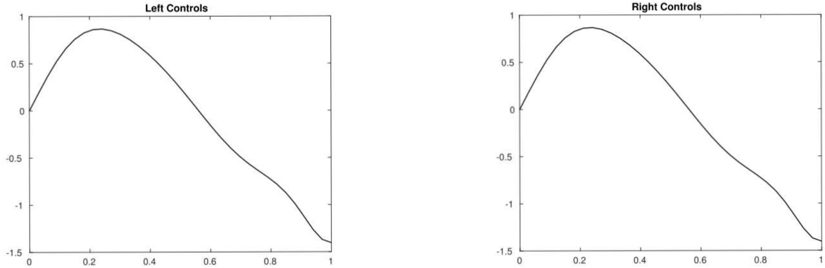




Optimal controls along the left vertical side wall (left image) and the right vertical side wall (right image) of the problem domain D.

Full Space: Inexact Linear Solvers





Optimal controls along the left vertical side wall (left image) and the right vertical side wall (right image) of the problem domain D.

Full Space: Inexact Linear Solvers

- F-GMRES used to solve augmented systems; `tol` $= 10^{-6}$.
- F-GMRES used to solve "inner" linearized forward and adjoint solves, C_u^{-1} and C_u^{-*} in the $\mathcal{P}$ preconditioner; `tol` $= 10^{-4}$.
- Non-overlapping additive Schwarz domain decomposition used to precondition inner solves: 4 subdomains, horizontal strips.
- Infinite-dimensional view of Lagrangian, $v_k = w_k$.
- Full fixed level-3 Clenshaw-Curtis sparse grid.

Full Space: Inexact Linear Solvers

- F-GMRES used to solve augmented systems; `tol` $= 10^{-6}$.
- F-GMRES used to solve "inner" linearized forward and adjoint solves, C_u^{-1} and C_u^{-*} in the $\mathcal{P}$ preconditioner; `tol` $= 10^{-4}$.
- Non-overlapping additive Schwarz domain decomposition used to precondition inner solves: 4 subdomains, horizontal strips.
- Infinite-dimensional view of Lagrangian, $v_k = w_k$.
- Full fixed level-3 Clenshaw-Curtis sparse grid.

- Serrano cluster at Sandia Labs; 80 dual-socket nodes, used 36 cores per core, i.e., 2880 cores total.
- Cores partitioned into 720 groups, for quadrature, with 4 cores each dedicated to iterative linear solver.
- **Hierarchical parallelism enabled through ROL.**

Full Space: Inexact Linear Solvers

iter	$J(x_k)$	$\|c(x_k)\|_C$	$\|\nabla L(x_k, \lambda_k)\|_X$	Δ_k	pcg	accept	ls calls	ls iters
0	0.07484675	7.820623×10^{-15}	8.377793×10^{-3}	1.00×10^{4}	—	—	—	—
1	0.05533699	1.661657×10^{-2}	3.641571×10^{-4}	1.00×10^{4}	11	1	16	597
2	0.03588474	3.052458×10^{-3}	9.338262×10^{-5}	1.00×10^{4}	13	1	33	1292
3	0.03515891	1.017679×10^{-4}	7.117806×10^{-5}	1.00×10^{4}	20	1	56	2303
4	0.03480817	1.444319×10^{-4}	2.439603×10^{-5}	1.00×10^{4}	15	1	75	3108
5	0.03480817	1.444319×10^{-4}	2.439321×10^{-5}	4.08×10^{0}	20	0	98	4157
6	0.03465050	2.237452×10^{-6}	4.364539×10^{-6}	3.03×10^{1}	2	1	104	4438
7	0.03464773	2.716452×10^{-7}	1.042585×10^{-7}	3.03×10^{1}	8	1	116	4989

Iteration history for full-space approach with iterative augmented system solves.

- About 43 F-GMRES iterations per solve; encouraging, considering that the size of the state space is more than 30 million, and that a fairly tight tolerance was used (10^{-6}).
- Many improvements are possible.

Conclusions

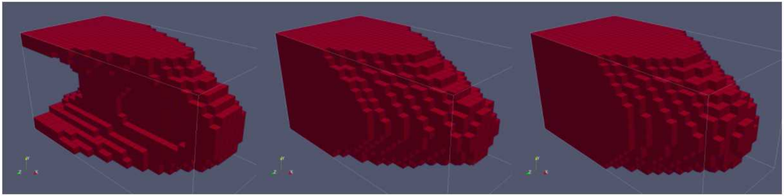
- **Expectation minimization** is an important **computational core** for optimization under uncertainty with a variety of risk models.
- Reduced-space trust-region methods enable adaptive sampling.
- Full-space trust-region methods enable large-scale iterative solvers.

Conclusions

- **Expectation minimization** is an important **computational core** for optimization under uncertainty with a variety of risk models.
- Reduced-space trust-region methods enable adaptive sampling.
- Full-space trust-region methods enable large-scale iterative solvers.

- Working on deploying these algorithms in structural optimization:

Mean Value **Risk Neutral** **Buffered Prob.**

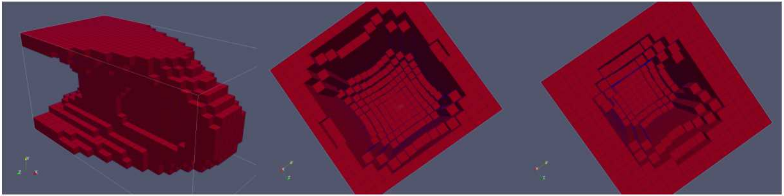


Conclusions

- **Expectation minimization** is an important **computational core** for optimization under uncertainty with a variety of risk models.
- Reduced-space trust-region methods enable adaptive sampling.
- Full-space trust-region methods enable large-scale iterative solvers.

- Working on deploying these algorithms in structural optimization:

Mean Value **Risk Neutral** **Buffered Prob.**



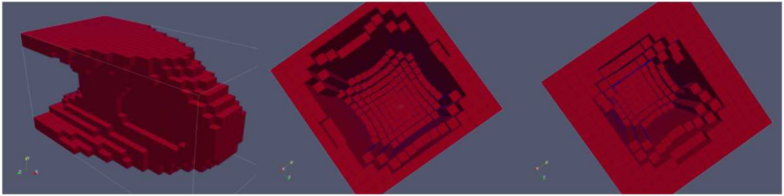
Topology changes from beam to thin shell to thick shell!

Conclusions

- **Expectation minimization** is an important **computational core** for optimization under uncertainty with a variety of risk models.
- Reduced-space trust-region methods enable adaptive sampling.
- Full-space trust-region methods enable large-scale iterative solvers.

- Working on deploying these algorithms in structural optimization:

Mean Value **Risk Neutral** **Buffered Prob.**



Topology changes from beam to thin shell to thick shell!

- For a deep dive, see Drew Kouri's Minitutorial MT8, **Thursday, April 19, 2:30pm-4:30pm, Grand Ballroom G**.