



Adaptive Randomized Sketching for Dynamic Nonsmooth Optimization

Robert Baraldi, Drew Kouri, Harbir Antil

SIAM Conference on Uncertainty Quantification

Trieste

2/28/24



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



Goal: Develop efficient algorithms to solve the regularized nonsmooth optimization problem,

$$\min_{\mathbf{z} \in \mathcal{Z}} F(\mathbf{z}) := j(\mathbf{z}) + \phi(\mathbf{z}). \quad (1)$$

- ▶ $\mathcal{Z}$ is a Hilbert space with $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$;
- ▶ $\phi : \mathcal{Z} \rightarrow [-\infty, \infty]$ is proper, closed, and convex, but may be nonsmooth;
- ▶ $j : \mathcal{Z} \rightarrow \mathbb{R}$ has Lipschitz continuous gradients on an open set containing $\text{dom } \phi$, but may be inexact;
- ▶ $F > -\infty$, bounded below on $\text{dom } \phi$.

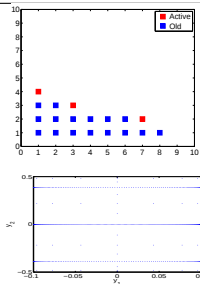
$j(\mathbf{z})$ may be nonconvex and often **impossible** to compute **exactly**.

- Stems from discretization, iterative procedures, adaptive model reduction, surrogate models, iterative linear and nonlinear solves, etc.

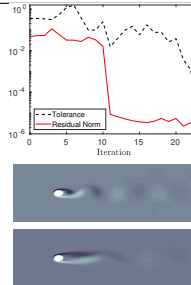
Adaptive Finite Elements



Adaptive Quadrature



Adaptive Compression





$\phi(\mathbf{z})$ are typically sparsity-inducing and temper model complexity, but **lack derivatives**.

- **Sparse regularization:** $\mathcal{Z} \hookrightarrow L^1(\Omega)$, $\beta \in L^\infty(\Omega)$, $\beta \geq 0$ a.e., and

$$\phi(\mathbf{z}) = \int_{\Omega} \beta(\omega) |\mathbf{z}(\omega)| d\omega.$$

Applications: Optimal control, data-science, learning, basis-pursuit.

- **Total Variation:** $\mathcal{U} \hookrightarrow BV(\Omega)$, $\beta \in L^\infty(\Omega)$, $\beta \geq 0$ a.e.,

$$\phi(\mathbf{u}) = \int_{\Omega} \beta(\omega) |\operatorname{div} \mathbf{u}(\omega)| d\omega.$$

Applications: Image processing, digital image correlation, topology optimization.

- **Convex Constraints:** $C \subset \mathcal{Z}$ nonempty, closed and convex,

$$\phi(\mathbf{z}) = \begin{cases} 0, & \text{if } \mathbf{z} \in C \\ +\infty, & \text{otherwise.} \end{cases}$$

Applications: Optimal control, inverse problems, optimal design.

- **Others:** Matrix completion (rank), phase retrieval.

Motivation: Nonconvexity, Nonsmoothness, Inexactness



- ▶ Arise naturally in physical problems, and are useful in enforcing certain solution properties.
 - ▶ **Problems:** local min, numerically complex, lacking derivatives, large-scale.
 - ▶ **Theory/software** exists for smoothed/convex counterparts (IPOPT, CVX, various Matlab/Julia/Python/... implementations).
 - ▶ **Memory** required to store state trajectory (and auxiliary info like Lagrange multipliers) is often prohibitively expensive: $\mathcal{O}(N(M + m))$ for $N \approx 10^5$, $M \approx 10^{10}$.
- ▶ **Key Algorithmic Requirements:**
 1. **Nonconvex** functions and **nonsmooth** regularizers.
 2. Handle **large-scale** problems with rapid convergence, mesh independence, and matrix free operations.
 3. **Leverage inexactness** by proving convergence for $j, \nabla j$ computed inexactly via discretization, reduced-order modeling, compression, etc.



- ▶ Dynamical Problem Reformulation $\rightarrow$ problem assumptions.
- ▶ Trust Region Overview.
- ▶ Inexactness Assumptions.
- ▶ Matrix Sketching.
- ▶ Convergence Results.
- ▶ Numerical Results:
 - ▶ Measure-Valued Parabolic Control.

$$\min_{u_n, z_n} \sum_{n=1}^N f_n(u_{n-1}, u_n, z_n) + \phi_n(z_n) \quad \text{s.t.} \quad c_n(u_{n-1}, u_n, z_n) = 0 \quad \forall n = 1, \dots, N. \quad (2)$$

- ▶ Replace u_n with the unique solution to $c(u_{n-1}, u_n, z_n) = 0$ for fixed u_{n-1}, z_n .
- ▶ Stack controls $\mathbf{z} = [z_1^T, \dots, z_N^T]^T \in \mathcal{Z} := \mathbb{R}^{Nm}$ and states $\mathbf{u} = [u_1^T, \dots, u_N^T]^T \in \mathcal{U} := \mathbb{R}^{NM}$

$$c(\mathbf{u}, \mathbf{z}) := \begin{bmatrix} c_1(u_0, u_1, z_1) \\ \vdots \\ c_N(u_{N-1}, u_N, z_N) \end{bmatrix}, \quad f(\mathbf{u}, \mathbf{z}) := \sum_{n=1}^N f_n(u_n, z_n), \quad \text{and} \quad \phi(\mathbf{z}) := \sum_{n=1}^N \phi_n(z_n).$$

$$\min_{\mathbf{u} \in \mathcal{U}, \mathbf{z} \in \mathcal{Z}} f(\mathbf{u}, \mathbf{z}) + \phi(\mathbf{z}) \quad \text{subject to} \quad c(\mathbf{u}, \mathbf{z}) = 0. \quad (3)$$



Assumption (1 - Function Characteristics)

- ▶ Functions f, c are twice continuously differentiable.
- ▶ There exists a unique state trajectory $\mathbf{z} \mapsto S(\mathbf{z}) : \mathcal{Z} \rightarrow \mathcal{U}$ satisfying $c(S(\mathbf{z}), \mathbf{z}) = 0$ for each $\mathbf{z} \in \mathcal{Z}$.
- ▶ The state Jacobian of c $d_{\mathbf{u}}c(\mathbf{u}, \mathbf{z})$, has a bounded inverse for all controls $\mathbf{z} \in \mathcal{Z}$.

Note: control Jacobian is $d_{\mathbf{z}}c(\mathbf{u}, \mathbf{z})$ and the partial derivatives are $d_{\mathbf{u}}f(\mathbf{u}, \mathbf{z})$, $d_{\mathbf{z}}f(\mathbf{u}, \mathbf{z})$.

Additionally, the Implicit Function Theorem [Hinze et al., 2009, Th. 1.41] ensures S_n and S are continuously differentiable for $S(\mathbf{z})$ stacked (recursively)

$$S(\mathbf{z}) := [S_1(u_0, z_1)^T \dots S_N(S_{N-1}(\dots, z_{N-1}), z_N)^T]^T.$$

$$\min_{\mathbf{z} \in \mathcal{Z}} \{F(\mathbf{z}) := j(\mathbf{z}) + \phi(\mathbf{z})\}, \quad (4)$$

where $j(\mathbf{z}) := f(S(\mathbf{z}), \mathbf{z})$ is the continuously differentiable reduced objective function with gradient

$$\nabla j(\mathbf{z}) = d_{\mathbf{z}}f(S(\mathbf{z}), \mathbf{z}) + (d_{\mathbf{z}}c(S(\mathbf{z}), \mathbf{z}))^{\top} \boldsymbol{\lambda}, \quad (5)$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{MN}$ solves the adjoint equation

$$d_{\mathbf{u}}c(S(\mathbf{z}), \mathbf{z})^{\top} \boldsymbol{\lambda} = -d_{\mathbf{u}}f(S(\mathbf{z}), \mathbf{z}). \quad (6)$$

Note: the adjoint equation (6) is solved backward in time, starting at $n = N$ and requires the entire state trajectory.

General Trust Region Logic & Methodology



Nonsmooth Trust Regions

- ▶ **Tools:** we want to solve (4) with a trust-region method.
- ▶ **Trust-region functionality:** at the k^{th} iteration, use surrogate (quadratic) model of smooth j to make progress:
 - ▶ (Approximate) Gradient ∇j , Hessian $B_k = B_k^T$;
 - ▶ Valid within a region determined by model performance and accuracy.
- ▶ Saves numerical cost for expensive forward solutions.
- ▶ **Problem:** nonsmooth trust-region methods are restrictive/impractical and computing gradient curvature information may be prohibitively expensive.
- ▶ **Solution:** rework standard trust-region literature for regularized functions with inexact function/gradient calculations (Baraldi and Kouri [2022], Aravkin et al. [2021]).

Algorithm 1: Nonsmooth Trust-Region Method (Baraldi and Kouri [2022])

Data: $\mathbf{z}_0 \in \text{dom } \phi$, $\Delta_0 > 0$, $0 < \eta_1 < \eta_2 < 1$, and $0 < \gamma_1 \leq \gamma_2 < 1 \leq \gamma_3$

for $k = 1, 2, \dots$ **do**

Model Selection: Choose a subproblem model j_k of j near $\mathbf{z}_k$ (j_k inexact!).

Step Computation: Compute $\mathbf{z}_{k+1} \in \mathcal{Z}$ that solves

$$\min_{\mathbf{z} \in \mathcal{Z}} m_k(\mathbf{z}) := j_k(\mathbf{z}) + \phi(\mathbf{z}) \quad \text{subject to} \quad \|\mathbf{z} - \mathbf{z}_k\| \leq \Delta_k.$$

Computed Reduction: Compute $\text{cred}_k \approx \text{ared}_k$ (inexact!).

Step Acceptance: Compute ratio of computed and predicted reduction

$$\rho_k := \frac{\text{cred}_k}{m_k(\mathbf{z}_k) - m_k(\mathbf{z}_{k+1})} < \eta_1 \quad \Rightarrow \quad \mathbf{z}_{k+1} \leftarrow \mathbf{z}_k.$$

Update Trust-Region Radius: $\Delta_{k+1} \in \begin{cases} [\gamma_1 \Delta_k, \gamma_2 \Delta_k], & \text{if } \rho_k < \eta_1 \\ [\gamma_2 \Delta_k, \Delta_k], & \text{if } \rho_k \in [\eta_1, \eta_2) \\ [\Delta_k, \infty), & \text{if } \rho_k \geq \eta_2. \end{cases}$

end



Trust-region subproblem: at each iteration, we solve

$$\min_{\mathbf{z} \in \mathcal{Z}} \{m_k(\mathbf{z}) := j_k(\mathbf{z}) + \phi(\mathbf{z})\} \quad \text{subject to } \|\mathbf{z} - \mathbf{z}_k\|_x \leq \Delta_k$$

where $\Delta_k > 0$ is the radius, $j_k : \mathcal{Z} \rightarrow \mathbb{R}$ is a model of j near $\mathbf{z}$.

Quadratic model: $j_k(x) = \langle g_k, \mathbf{z} - \mathbf{z}_k \rangle_{\mathcal{Z}} + \frac{1}{2} \langle B_k(\mathbf{z} - \mathbf{z}_k), \mathbf{z} - \mathbf{z}_k \rangle_{\mathcal{Z}}$ where $g_k \approx \nabla j(\mathbf{z}_k)$ and B_k contains some curvature information, $B_k \approx \nabla^2 j(\mathbf{z}_k)$ or some quasi-Newton approximation.

Theoretical Challenges:

1. Ensure convergence with j_k, g_k inexact;
2. Ensure subproblem step yields Fraction of Cauchy Decrease (FCD, shown in Baraldi and Kouri [2022]);
3. Handle nonsmooth subproblems efficiently (Proximal subproblem solvers, show in Baraldi and Kouri [2022]).



Recall: Infinite-dimensional optimization function and gradient evaluations are often **impossible** to compute without discretization or even store, leading to **inexactness**.

When evaluating the reduction of the objective function, we approximate

$$\text{cred}_k \approx \text{ared}_k := (j(\mathbf{z}_k) + \phi(\mathbf{z}_k)) - (j(\mathbf{z}_{k+1}) + \phi(\mathbf{z}_{k+1}))$$

and with $\text{pred}_k := m_k(\mathbf{z}_k) - m_k(\mathbf{z}_{k+1})$ and construct the bound

$$|\text{ared}_k - \text{cred}_k| \leq \kappa_{\text{obj}} [\eta \min\{\text{pred}_k, \theta_k\}]^\zeta, \quad (7)$$

$\eta < \min\{\eta_1, (1 - \eta_2)\}$, $\{\theta_k\}_{k=1}^{+\infty} \subset [0, +\infty)$, $\lim_{k \rightarrow +\infty} \theta_k = 0$.

Similarly for the gradient with $\kappa_{\text{grad}} > 0$ and $h_k := t_0^{-1} \|\text{Prox}_{t_0\phi}(\mathbf{z}_k - t_0 g_k) - \mathbf{z}_k\|_{\mathcal{Z}}$:

$$\|g_k - \nabla j(\mathbf{z}_k)\| \leq \kappa_{\text{grad}} \min\{\Delta_k, h_k\} \quad \forall k. \quad (8)$$



Idea: use randomized sketching to compress $M \times N$ state trajectory matrix

$$U = (u_1 | \dots | u_N) \in \mathbb{R}^{M \times N}$$

and use the sketched state U^r for gradient evaluations and Hessian applications.

Goal: Generate an accurate rank- r approximation of U using $\mathcal{O}(r(M + N))$ storage from [Tropp et al., 2019, Muthukumar et al., 2021].

Matrix Sketching: Given the four random linear dimension reduction maps and $r \geq k \geq s$, $k = 2r + 1$, and $s = 2k + 1$,

$$\Upsilon \in \mathbb{R}^{k \times M}, \quad \Omega \in \mathbb{R}^{k \times N}, \quad \Phi \in \mathbb{R}^{s \times M}, \quad \text{and} \quad \Psi \in \mathbb{R}^{s \times N}$$

$$X := \Upsilon U \in \mathbb{R}^{k \times N},$$

co-range sketch (row space),

$$Y := U \Omega^* \in \mathbb{R}^{M \times k},$$

range sketch (column space),

$$Z := \Phi U \Psi^* \in \mathbb{R}^{s \times s},$$

core sketch (singular values).



Algorithm 2: Online State Sketching

Require: $X = 0, Y = 0$

- 1: **for** $n = 1, \dots, N$ **do**
 - 2: Given u_{n-1} and z_n , solve $c_n(u_{n-1}, u_n, z_n) = 0$ for u_n
 - 3: $X \leftarrow X + \Upsilon u_n e_n^T$
 - 4: $Y \leftarrow Y + u_n (\Omega e_n)$
 - 5: $Z \leftarrow Z + (\Phi u_n)(\Psi e_n)^T$
 - 6: **end for**
-

where e_i is the i^{th} unit vector.

Storage Requirement: Sketching requires $k(M + N) + s^2$ FLOPs.



Algorithm 3: Recovery

- 1: $(Q, R_2) \leftarrow \text{qr}(Y, 0), \quad Q \in \mathbb{R}^{M \times k}, R_2 \in \mathbb{R}^{k \times k}.$
 - 2: $(P, R_1) \leftarrow \text{qr}(X^T, 0), \quad P \in \mathbb{R}^{N \times k}, R_1 \in \mathbb{R}^{k \times k}.$
 - 3: $C \leftarrow (\Phi Q)^\dagger Z((\Psi P)^\dagger)^T, \quad C \in \mathbb{R}^{k \times k}.$
 - 4: $W \leftarrow CP^T W \in \mathbb{R}^{k \times N}.$
 - 5: $u_n \approx QWe_n.$
-

- ▶ Approximately recover U via QR factorizations of X^T and Y and solve 2 small LS problems (Muthukumar et al. [2021]).
- ▶ **Storage Requirement:** Recovery requires $k(M + N) + k^2$ FLOPs.
- ▶ Further reductions can be gained by overwriting X and Y with Q and W : j^{th} column then becomes $U^r[:, j] = (QW)[:, j]$. This reduces total complexity to $\mathcal{O}(k(M + N))$.

**Lemma (Total Reconstruction Error - Muthukumar et al. [2021])**

Let U^r denote sketch of U associated with parameter r . Then

$$\mathbb{E}_{\Upsilon, \Omega, \Phi, \Psi} \|U - U^r\|_F \leq \sqrt{6} \left(\sum_{i \geq r+1} \sigma_i^2(U) \right)^{\frac{1}{2}}.$$

The rank k approximation is a constant factor farther than the best rank r approximation on average. Similar results exist for the probability of large deviation in Tropp et al. [2019].

Adaptive Compression: Increase sketch rank r until dynamical system residual, $\|c(U^r, z)\|$ satisfies required tolerance.

Algorithm 4: Inexact Gradient Computation with Adaptive Rank - (8)



Require: $\mathbf{z}_k \in \mathbb{R}^{mN}$, initial r , sketch for state $\mathbf{u}_k^r := \text{vec}(U^r)$, $\Delta_k > 0$, $\kappa_{\text{scale}} > 0$, and tolerance $\mu_{\text{grad}} > 1$.

- 1: Set $\tau_k^- \leftarrow \kappa_{\text{scale}} \Delta_k$
- 2: Compute $g_k \leftarrow g(\Lambda(\mathbf{u}_k^r, \mathbf{z}_k), \mathbf{u}_k^r, \mathbf{z}_k)$, $h_k \leftarrow t^{-1} \|\text{Prox}_{r\phi}(\mathbf{z}_k - tg_k) - \mathbf{z}_k\|$, and $\tau_k^+ \leftarrow \kappa_{\text{scale}} \min\{h_k, \Delta_k\}$
- 3: **while** $\tau_k^- > \mu_{\text{grad}} \tau_k^+$ **do**
- 4: **while** $r < \min\{M, N\}$ **do**
- 5: Compute norm of the constraint residual $\text{rnorm} \leftarrow \|c(\mathbf{u}_k^r, \mathbf{z}_k)\|$
- 6: **if** $\text{rnorm} < \tau_k^+$ **then**
- 7: Compute gradient $g_k^r \leftarrow g(\Lambda(\mathbf{u}_k^r, \mathbf{z}_k), \mathbf{u}_k^r, \mathbf{z}_k)$, **break**
- 8: **end if**
- 9: $r \leftarrow 2r$ and solve state equation at $\mathbf{z}_k$ and resketch $\mathbf{u}_k^r$
- 10: **end while**
- 11: Set $g_k \leftarrow g_k^r$ and compute $h_k \leftarrow t^{-1} \|\text{Prox}_{r\phi}(\mathbf{z}_k - tg_k) - \mathbf{z}_k\|$
- 12: Set $\tau_k^- \leftarrow \tau_k^+$ and $\tau_k^+ \leftarrow \kappa_{\text{scale}} \min\{h_k, \Delta_k\}$
- 13: **end while**
- 14: **return** Approximate gradient $g_k \approx \nabla j(\mathbf{z}_k)$ using $\mathcal{O}(r(M + N) + mN)$ storage for $r \leq \min\{M, N\}$. **Is this guaranteed to converge in finite time?**



Assumption (2 - Regularity Properties for (1))

The following conditions hold for the data in (1):

1. There exists $\mathcal{U}_0 \subset \mathcal{U}$ open and bounded such that $\{\mathbf{u} \in \mathcal{U} | \exists \mathbf{z} \in \mathcal{Z}_0, c(\mathbf{u}, \mathbf{z}) = 0\} \subseteq \mathcal{U}_0$.
2. There exists singular value thresholds $0 < \sigma_0 \leq \sigma_1 < +\infty$ such that for any $\mathbf{u} \in \mathcal{U}_0$ and $\mathbf{z} \in \mathcal{Z}_0$, $\sigma_0 \leq \sigma_{\min}(\mathbf{d}_{\mathbf{u}}c(\mathbf{u}, \mathbf{z})) \leq \sigma_{\max}(\mathbf{d}_{\mathbf{u}}c(\mathbf{u}, \mathbf{z})) \leq \sigma_1$.
3. The following functions are Lipschitz continuous on $\mathcal{U}_0 \times \mathcal{Z}_0$ with respect to their first arguments, and their respective Lipschitz moduli are independent of $\mathbf{z} \in \mathcal{Z}_0$:
 - 3.1 the state Jacobian of the constraint $\mathbf{d}_{\mathbf{u}}c(\mathbf{u}, \mathbf{z})$;
 - 3.2 the control Jacobian of the constraint $\mathbf{d}_{\mathbf{u}}c(\mathbf{u}, \mathbf{z})$;
 - 3.3 the state gradient of the smooth objective term $\mathbf{d}_{\mathbf{u}}f(\mathbf{u}, \mathbf{z})$;
 - 3.4 the control gradient of the smooth objective term $\mathbf{d}_{\mathbf{z}}f(\mathbf{u}, \mathbf{z})$.



Using Assumption 2, we can bound the state, adjoint and gradient errors as in [Muthukumar et al., 2021, Prop. 4.1].

We use this to show our gradient approximation algorithm satisfies (8).

Lemma (Adaptive Rank Gradient Approximation)

If Assumption 2 holds, then Algorithm 4 produces a gradient approximation $g_k = g(\Lambda(\mathbf{u}_k^r, \mathbf{z}_k), \mathbf{u}_k^r, \mathbf{z}_k)$, in finitely many iterations, that satisfies the gradient error bound (8) with $\kappa_{\text{grad}} = \kappa_4 \kappa_{\text{scale}} \mu_{\text{grad}}$.

Algorithm 5: Final Sketched Nonsmooth Trust-Region Algorithm

Data: $\mathbf{z}_0 \in \text{dom } \phi$, $\Delta_0 > 0$, $0 < \eta_1 < \eta_2 < 1$, and $0 < \gamma_1 \leq \gamma_2 < 1 \leq \gamma_3$

for $k = 1, 2, \dots$ **do**

Model Selection: Use Algorithm 4 with rank r_k to compute g_k and choose B_k

Step Computation: Compute $\mathbf{z}_{k+1} \in \mathcal{Z}$ (**efficient**) that solves

$$\min_{\mathbf{z} \in \mathcal{Z}} m_k(\mathbf{z}) := j_k(\mathbf{z}) + \phi(\mathbf{z}) \quad \text{subject to} \quad \|\mathbf{z} - \mathbf{z}_k\| \leq \Delta_k.$$

Computed Reduction: Compute $\text{cred}_k \approx \text{ared}_k$ (**inexact!**).

Step Acceptance: Compute ratio of computed and predicted reduction

$$\rho_k := \frac{\text{cred}_k}{m_k(\mathbf{z}_k) - m_k(\mathbf{z}_{k+1})} < \eta_1 \quad \Rightarrow \quad \mathbf{z}_{k+1} \leftarrow \mathbf{z}_k.$$

Update Trust-Region Radius: $\Delta_{k+1} \in \begin{cases} [\gamma_1 \Delta_k, \gamma_2 \Delta_k], & \text{if } \rho_k < \eta_1 \\ [\gamma_2 \Delta_k, \Delta_k], & \text{if } \rho_k \in [\eta_1, \eta_2] \\ [\Delta_k, \infty), & \text{if } \rho_k > \eta_2. \end{cases}$

end



Recall: $h_k := r_0^{-1} \|\text{Prox}_\phi(x_k - r_0 g_k) - x_k\|_X$

Theorem (Algorithm Convergence - Baraldi and Kouri [2022])

Let $\{x_k\}$ be the sequence of iterates generated

$$\liminf_{k \rightarrow \infty} h_k = 0 \quad \Rightarrow \quad \liminf_{k \rightarrow \infty} r_0^{-1} \|\text{Prox}_\phi(x_k - r_0 \nabla f(x_k)) - x_k\|_X = 0. \quad (9)$$

Note: Permits unbounded model curvature.

Given $\varepsilon > 0$ and bounded model curvature, then Trust-Region Algorithm satisfies $h_k \leq \min\{\varepsilon, 1\}$ in at most $\mathcal{O}(\varepsilon^{-2})$ iterations.

Note: This is a **worst-case bound**; we find much better performance in practice. Can get superlinear or quadratic local convergence (Baraldi and Kouri [2023]).



Goals: Demonstrate Algorithm 1 inexactness control for adaptive compression.

Let $\Omega = (0, 1)^2$, $T = 2$ and $\alpha = 0.1$, and solve

$$\min_{z \in M(\Omega)} \frac{1}{2} \int_0^T \|S(z) - w\|_{L^2(\Omega)}^2 dt \quad \text{subject to} \quad \|z\|_{M(\Omega)} \leq \alpha, \quad z \succeq 0,$$

for $w = |(\sin(2\pi x_1) \sin(2\pi x_2))|^{10}$ and $S(z) = u$ solves

$$\begin{aligned} \partial_t u - \Delta u &= 0 && \text{in } \Omega \times (0, T) \\ \nabla u \cdot n &= 0 && \text{on } \partial\Omega \times (0, T) \\ u(0) &= z && \text{in } \Omega. \end{aligned}$$

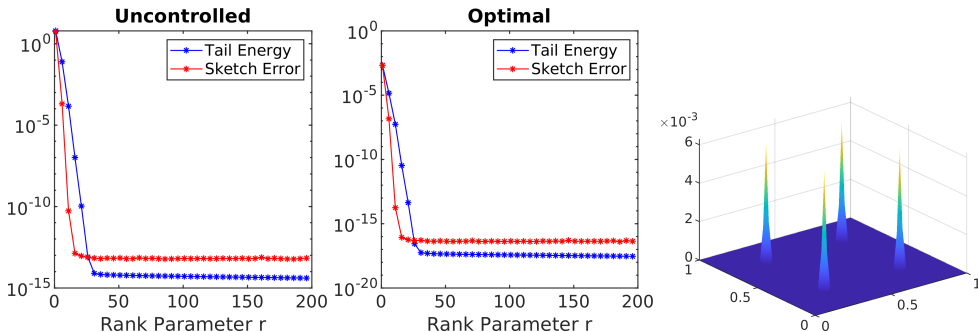
Discretization: P1 FEM + implicit Euler for states and variational discretization for controls.

Problem size: 4225 control degrees of freedom.

State Tail Energy Decay



Figure 1 depicts the tail energy and sketching error averaged over 20 realizations for uncontrolled (random initial point) and optimal $\mathbf{z}$.



The sketching error averaged over 20 realizations and the tail energy for the uncontrolled state (left) and the optimal state(right). recall that the rank of the sketch is $k = 2r + 1$.



Termination Criteria: $h_k \leq 10^{-4}h_1$ or $\|z_k^+ - z_k\| \leq 10^{-6}h_1$ or $k = 40$.
Rank Increase Function: $r \leftarrow 2r$.

rank	objective	niter	nobjs	ngrad	nhess	nobjn	nprox	ζ
*1	2.680962e-02	16	17	9	521	1036	1726	148.78
2	2.680946e-02	37	38	38	1948	4597	3873	89.12
3	2.680946e-02	31	32	32	1635	3759	3248	63.55
4	2.680946e-02	22	23	23	1163	2654	2308	49.35
5	2.680946e-02	22	23	23	1160	2640	2305	40.31
Adaptive	2.680946e-02	22	23	25	1162	2530	2309	25.95
Full	2.680946e-02	23	24	24	1212	2793	2409	---

* rank 1 experiment failed to converge due to step-size tolerance.

The final adapted rank was $r = 8$ leading to 26x compression.



- ▶ **Numerical solutions** of infinite-dimensional problems requires **expensive approximations**.
 - ▶ Objectives and gradients can only be compute **inexactly**.
- ▶ Nonsmooth trust-region is **provably convergent** even with **inexact computation** via matrix sketching/compression.
 - ▶ Next: incorporate other compression techniques, harder examples (fluid flow around a cylinder).

Acknowledgements



Questions?

Thank you-

- ▶ SIOPT Attendees + Session Organizers
- ▶ Collaborators: Drew Kouri, Harbir Antil



- A. Y. Aravkin, R. Baraldi, and D. Orban. A proximal quasi-Newton trust-region method for nonsmooth regularized optimization, 2021.
- Robert J. Baraldi and Drew P. Kouri. A proximal trust-region method for nonsmooth optimization with inexact function and gradient evaluations. *Mathematical Programming*, pages 1–40, 2022.
- Robert J. Baraldi and Drew P. Kouri. Local convergence analysis of an inexact trust-region method for nonsmooth optimization. *Operations Research Letters*, page submitted, 2023.
- Michael Hinze, Rene Pinnau, Michael Ulbrich, and Stefan Ulbrich. *Optimization with PDE Constraints*, volume 23. Springer, New York, NY, 2009.
- Ramchandran Muthukumar, Drew P Kouri, and Madeleine Udell. Randomized sketching algorithms for low-memory dynamic optimization. *SIAM Journal on Optimization*, 31(2):1242–1275, 2021.
- J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher. Streaming low-rank matrix approximation with an application to scientific simulations. *SIAM Journal on Scientific Computing*, 41:A2430 – A2463, 2019.



Definition (Moreau-Yosida Envelope)

For a proper, lower semicontinuous function $\phi : \mathcal{Z} \rightarrow \overline{\mathbb{R}}$ and a parameter $t > 0$, the *Moreau envelope* $e_{t\phi} : \mathcal{Z} \rightarrow \mathbb{R}$ and the *proximal operator* $\text{Prox}_{t\phi} : \mathcal{Z} \rightarrow \mathcal{Z}$ are defined by

$$e_{t\phi}(x) := \inf_{z \in \mathcal{Z}} \frac{1}{2t} \|z - x\|^2 + \phi(z), \quad (10a)$$

$$\text{Prox}_{t\phi}(x) := \operatorname{argmin}_{z \in \mathcal{Z}} \frac{1}{2t} \|z - x\|^2 + \phi(z). \quad (10b)$$

- ▶ **Interpretation:** extension of cost function to minimizing ϕ and near x .
- ▶ **Utility:** many proximal operators have **analytic** solutions;
 - ▶ **L^1 -Norm:** $\mathcal{Z} = L^2(\Omega)$, $\phi(\mathbf{z}) = \beta \|\mathbf{z}\|_{L^1(\Omega)} \Rightarrow \text{Prox}_{t\phi} = \text{sign}(\mathbf{z}) \odot (|\mathbf{x}| - t\beta)_+$.
 - ▶ **ReLU:** $\mathcal{Z} = \mathbb{R}$, $\phi(\mathbf{z}) = \max\{0, \mathbf{z}\} \Rightarrow \text{Prox}_{t\phi} = \max\{\mathbf{z} - t, \min\{0, \mathbf{z}\}\}$.

Nonsmooth Stationarity

Definition (Local Minimizer)

$\bar{\mathbf{z}} \in \mathcal{Z}$ is a **local minimizer** of $(j + \phi)$ if there exists a neighborhood U of $\bar{\mathbf{z}}$ on which $(j + \phi)(\bar{\mathbf{z}}) \leq (j + \phi)(\mathbf{z}) \quad \forall \mathbf{z} \in U$. Additionally, $\bar{\mathbf{z}}$ satisfies

$$-\nabla j(\bar{\mathbf{z}}) \in \partial\phi(\bar{\mathbf{z}}) \quad \Leftrightarrow \quad \bar{\mathbf{z}} = \text{Prox}_{t\phi}(\bar{\mathbf{z}} - t\nabla j(\bar{\mathbf{z}})) \quad \forall t > 0. \quad (11)$$

Lemma (Generalized Gradient & Stationary Point)

If $\mathbf{z} \in \mathcal{Z}$ is a stationary point of $(j + \phi)$, then $h(\bar{\mathbf{z}}) = 0$ where

$$h(\mathbf{z}) := t^{-1} \|\text{Prox}_{t\phi}(\mathbf{z} - t\nabla j(\mathbf{z})) - \mathbf{z}\|$$

for arbitrary fixed $t > 0$.



Assume $\tau_k^{\text{obj}} \leq \mu_{\text{obj}} \tau_{k+1}^{\text{obj}}$, then $\kappa_{\text{obj}} = (1 + \mu_{\text{obj}}) \bar{\kappa}_{\text{obj}} C_{\text{obj}}$ since

$$|\text{ared}_k - \text{cred}_k| \leq (1 + \mu_{\text{obj}}) \bar{\kappa}_{\text{obj}} C_{\text{obj}} [\eta \min\{\text{pred}_k, \theta_k\}]^\zeta.$$

Algorithm 6: Inexact Objective Function Computation

Require: Constants $\bar{\kappa}_{\text{obj}} > 0$ and $\mu_{\text{obj}} \geq 1$, the current objective function tolerance τ_k , the current iterate $x_k \in X$ and approximate objective value $v_k = \bar{f}(x_k, \tau_k)$, the new iterate $z_{k+1} \in X$, and the predicted reduction pred_k .

Set $\tau_{k+1} \leftarrow \bar{\kappa}_{\text{obj}} [\eta \min\{\text{pred}_k, \theta_k\}]^\zeta$

if $\mu_{\text{obj}} \tau_{k+1} < \tau_k$ **then**

 Compute $v_k \leftarrow \bar{f}(x_k, \tau_{k+1})$

end if

Compute $v_{k+1} \leftarrow \bar{f}(z_{k+1}, \tau_{k+1})$ and set $\text{cred}_k = (v_k + \phi(x_k)) - (v_{k+1} + \phi(z_{k+1}))$



Assume that we have an approximation $\bar{g} : X \times [0, +\infty) \rightarrow X$ that satisfies $C_{\text{grad}} \geq 0$ such that

$$\bar{g}(x, 0) = \nabla f(x) \quad \text{and} \quad \|\nabla f(x) - \bar{g}(x, \tau)\| \leq C_{\text{grad}} \tau \quad \forall \tau \in \mathbb{R}_+. \quad (12)$$

Algorithm 7: Inexact Gradient Computation

Require: Constant $\bar{\kappa}_{\text{grad}} > 0$, a tolerance $\mu_{\text{grad}} > 1$, current iterate $x_k \in X$, and Δ_k .

Set $\tau_k^- \leftarrow \bar{\kappa}_{\text{grad}} \Delta_k$

Compute $g_k \leftarrow \bar{g}(x_k, \tau_k^-)$ and $h_k \leftarrow r_0^{-1} \|\text{Prox}_{r_0 \phi}(x_k - r_0 g_k) - x_k\|$

Set $\tau_k^+ \leftarrow \bar{\kappa}_{\text{grad}} \min\{h_k, \Delta_k\}$

while $\mu_{\text{grad}} \tau_k^+ < \tau_k^-$ **do**

 Compute $g_k \leftarrow \bar{g}(x_k, \tau_k^+)$ and $h_k \leftarrow r_0^{-1} \|\text{Prox}_{r_0 \phi}(x_k - r_0 g_k) - x_k\|$

 Set $\tau_k^- \leftarrow \tau_k^+$ and $\tau_k^+ \leftarrow \bar{\kappa}_{\text{grad}} \min\{h_k, \Delta_k\}$

end while

If $h(x_k) > 0$, Algorithm 7 terminates finitely and the inexact gradient condition holds with $\kappa_{\text{grad}} = \mu_{\text{grad}} \bar{\kappa}_{\text{grad}} C_{\text{grad}}$.



Define the adjoint equation residual $G : \mathcal{U} \times \mathcal{U} \times \mathcal{Z} \rightarrow \mathcal{U}$ by

$$G(\lambda, \mathbf{u}, \mathbf{z}) := d_{\mathbf{u}}f(\mathbf{u}, \mathbf{z}) + (d_{\mathbf{u}}c(\mathbf{u}, \mathbf{z}))^* \lambda$$

with $\Lambda(\mathbf{u}, \mathbf{z}) \in \mathcal{U}$ the solution of $G(\Lambda(\mathbf{u}, \mathbf{z}), \mathbf{u}, \mathbf{z}) = 0$ for the fixed $\mathbf{u}, \mathbf{z}$. Next define the equation

$$g(\lambda, \mathbf{u}, \mathbf{z}) := d_{\mathbf{z}}f(\mathbf{u}, \mathbf{z}) + (d_{\mathbf{z}}c(\mathbf{u}, \mathbf{z}))^* \lambda.$$

When evaluated at $\mathbf{u} = S(\mathbf{z})$ and $\lambda = \Lambda(\mathbf{u}, \mathbf{z})$, $g(\lambda, \mathbf{u}, \mathbf{z})$ is the gradient of the reduced objective function j as in (5).

Evaluating $g(\lambda, \mathbf{u}, \mathbf{z})$ at the sketched state $\mathbf{u}^r = \text{vec}(U^r)$ instead of the full state trajectory $\mathbf{u} = S(\mathbf{z})$ reduces the memory burden¹ for gradient computation but $g^r(\mathbf{z}) = g(\Lambda(\mathbf{u}^r, \mathbf{z}), \mathbf{u}^r, \mathbf{z})$ an approximation of true gradient $g(\Lambda(S(\mathbf{z}), \mathbf{z}), S(\mathbf{z}), \mathbf{z})$.