A Matrix-free Algorithm for Reduced-space PDE-governed Optimization

Pengfei Meng

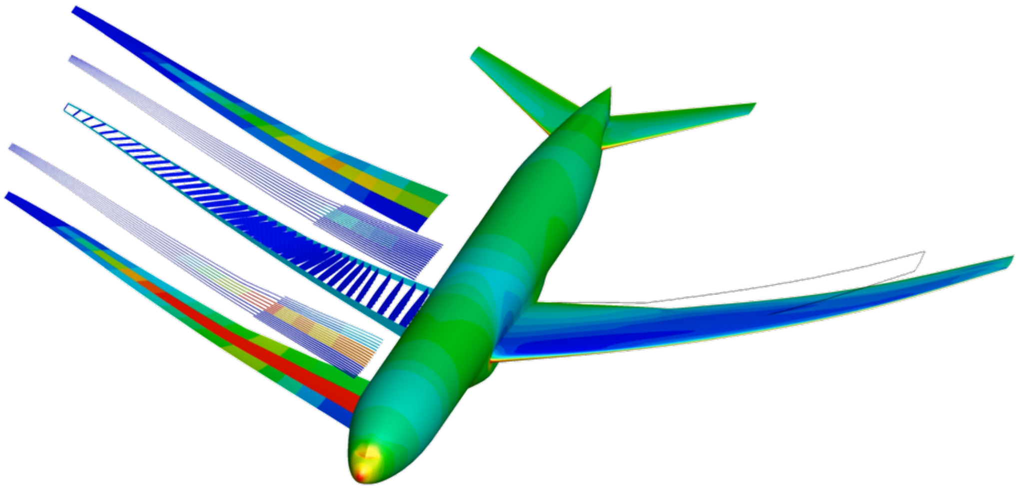
Outline

1. Introduction
2. Homotopy-Based Globalization
3. Iterative Solver & Preconditioner
4. Tests and Applications
5. Contributions and Recommendations

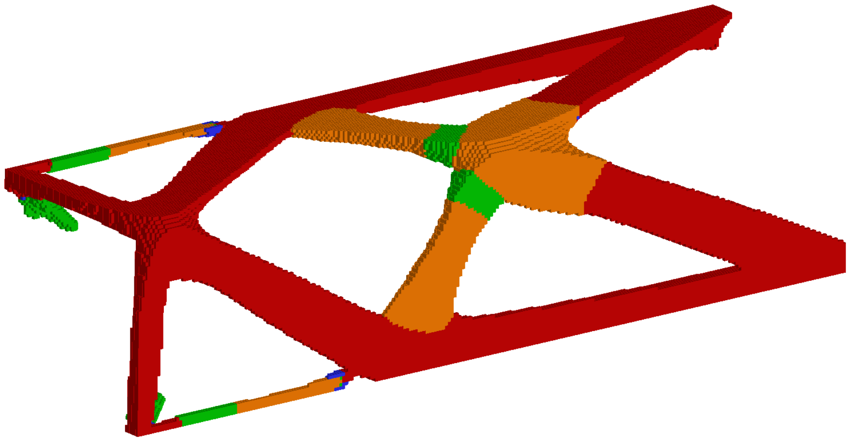
Improved engineering design is needed to address climate change

Aviation can reduce its emissions by improving aircraft through optimization

Optimization problems with state-based constraints are common, yet challenging to solve



Aero-Structural Optimization
Kenway and Martins [16]



Topology Optimization
Kennedy [17]

Conventional optimization algorithms are poorly suited for state-based constraints

Conventional optimization algorithms require the explicit total constraint Jacobian

- The gradient of each constraint requires the solution of a linear PDE (the adjoint)
- Cost is intractable when there are hundreds of constraints
- Storing the (dense) constraint Jacobian can also be expensive

We will consider the following, generic PDE-constrained optimization problem

$$
\begin{array}{rll}
\min\limits_{x,u} & f(x,u) & \\
\text{subject to} & h(x,u) & = 0 \\
 & g(x,u) & \geq 0 \\
\text{governed by} & \mathcal{R}(x,u) & = 0
\end{array}
$$

The Lagrangian:

$$
\mathcal{L}(x,u,\psi,s,\lambda_h,\lambda_g) = f(x,u) + \lambda_h^T h(x,u) + \lambda_g^T (g(x,u) - s) + \psi^T \mathcal{R}(x,u)
$$

Optimal designs satisfy the first-order optimality conditions (KKT condition)

$$\partial_x \mathcal{L} = \partial_x f + \lambda_h^T \partial_x h + \lambda_g^T \partial_x g + \psi^T \partial_x \mathcal{R} = 0,$$

Adjoint Equation: $$\partial_u \mathcal{L} = \partial_u f + \lambda_h^T \partial_u h + \lambda_g^T \partial_u g + \psi^T \partial_u \mathcal{R} = 0,$$

State Equation: $$\partial_\psi \mathcal{L} = \mathcal{R} = 0,$$

$$\partial_{\lambda_h} \mathcal{L} = h = 0,$$

$$\partial_{\lambda_g} \mathcal{L} = g - s = 0,$$

Complementarity: $$\begin{cases} -\mathsf{S}\Lambda_g e = 0, \\ s \geq 0, \quad \lambda_g \leq 0. \end{cases}$$

- A coupled, nonlinear system of equations;
- ill-conditioned;
- potentially large;

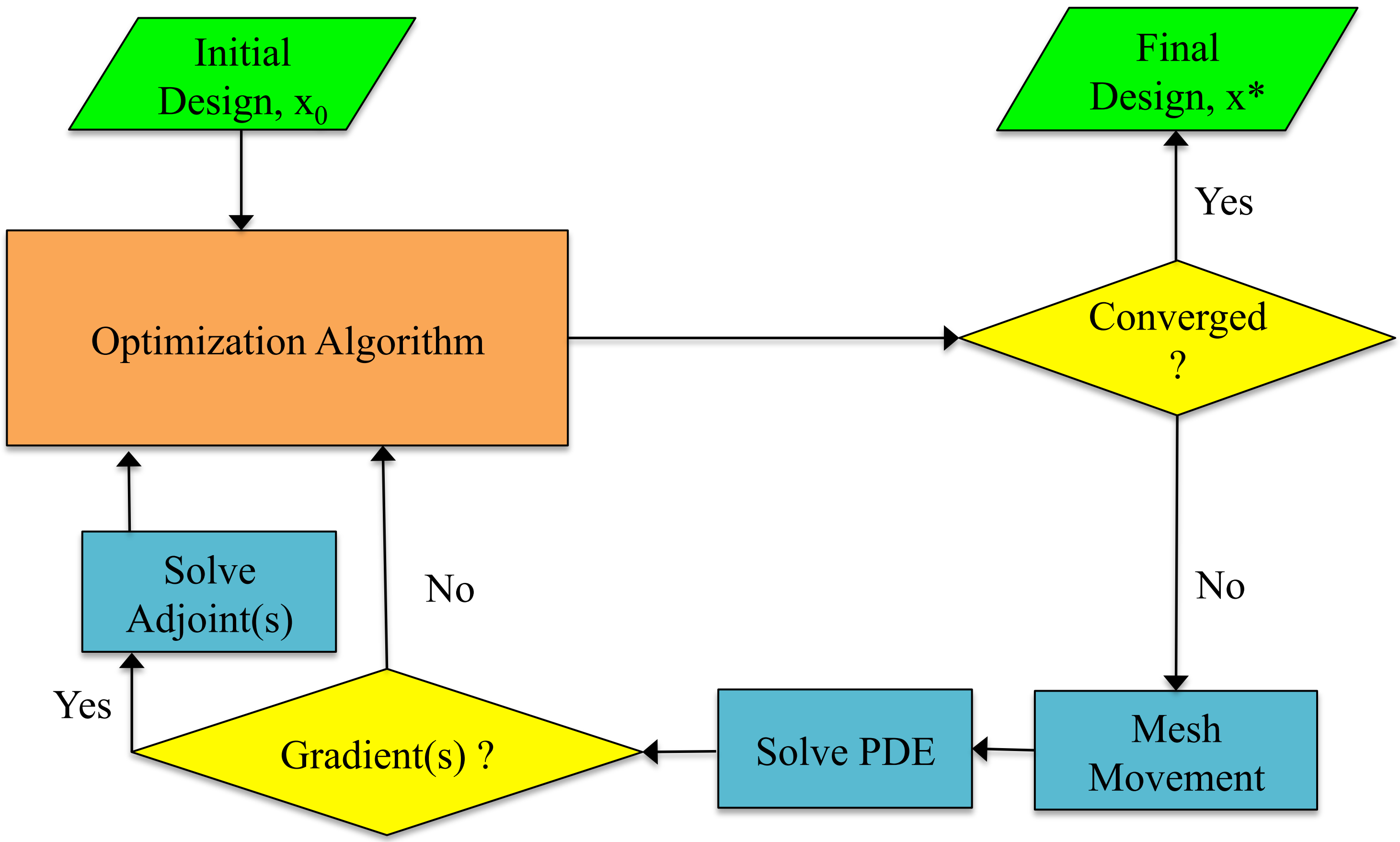
The full-space approach solves all the variables simultaneously, including the PDE state and adjoint

- The full-space KKT-system is large (at least 2 times the number of states), indefinite, and ill-conditioned
- Globalization algorithms for nonlinear PDEs are difficult to incorporate into full-space methods
- If the full-space optimization does not converge, the PDE is not even satisfied and cannot be used to inform decision making

The reduced-space approach treats the PDE state and adjoint as implicit functions

- Reduced-space methods can make use of existing simulation solvers and adjoint solvers
- The reduced-space optimization system is much smaller than the full-space system
- Reduced-space algorithms have been successfully applied to unconstrained and equality-constrained problems

An advantage of the reduced-space approach is its modular structure



The reduced-space KKT conditions are as follows:

$$F(q) \equiv F(x, s, \lambda_h, \lambda_g) \equiv \begin{bmatrix} \nabla_x f + \lambda_h^T \nabla_x h + \lambda_g^T \nabla_x g \\ -\mathsf{S}\Lambda_g e \\ h \\ g - s \end{bmatrix} = 0,$$

$$\text{subject to} \quad s_i \geq 0, \quad \text{and} \quad \lambda_{gi} \leq 0 \qquad \forall i = 1, 2, \ldots, m$$

Inexact-Newton are attractive for solving the reduced-space problem

- We solve the root-finding problem $F(q) = 0$ using a Newton-based approach

$$\left\|(\nabla_q F)\Delta q^{(k)} + F(q^{(k)})\right\| \leq \eta_k \left\|F(q^{(k)})\right\|$$

- The KKT-vector products are evaluated matrix-freely using second-order adjoints
- This way the explicit Hessian and the explicit total constraint Jacobian are not required
- The cost of the KKT-vector products is independent of the number of design variables and nonlinear constraints

The following challenges must be addressed in order to use inexact-Newton methods for general nonlinear constraints

- Globalization: must avoid stationary points that are not local minimizers, i.e. handle nonconvexity.
- Preconditioning: accelerate the convergence of Krylov iterative methods in a matrix-free way.

Outline

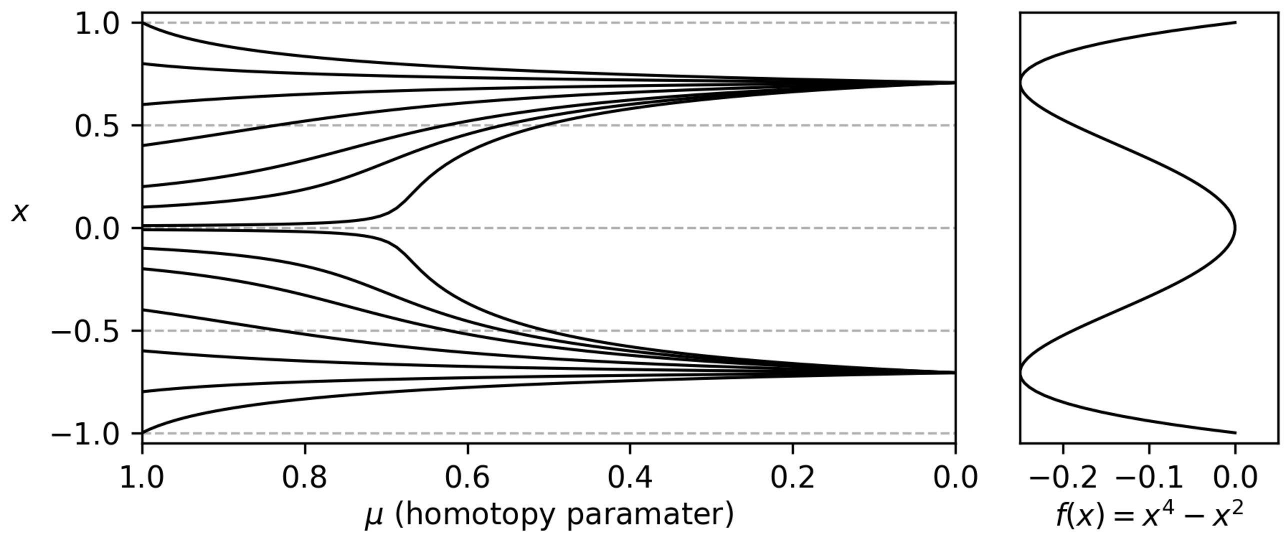
1. Introduction
2. Homotopy-Based Globalization
3. Iterative Solver & Preconditioner
4. Tests and Applications
5. Contributions and Recommendations

A homotopy map defines a solution path from an easy-to-obtain solution to the desired solution

The desired problem: $\min_x \quad f(x) = x^4 - x^2$

The easy problem: $\min_x \quad f(x) = \frac{1}{2}(x - x_0)^2$

$$H(x, \mu) = (1-\mu)\nabla f(x) + \mu(x - x_0) = 0, \quad x_0 \in [-1.0, 1.0]$$



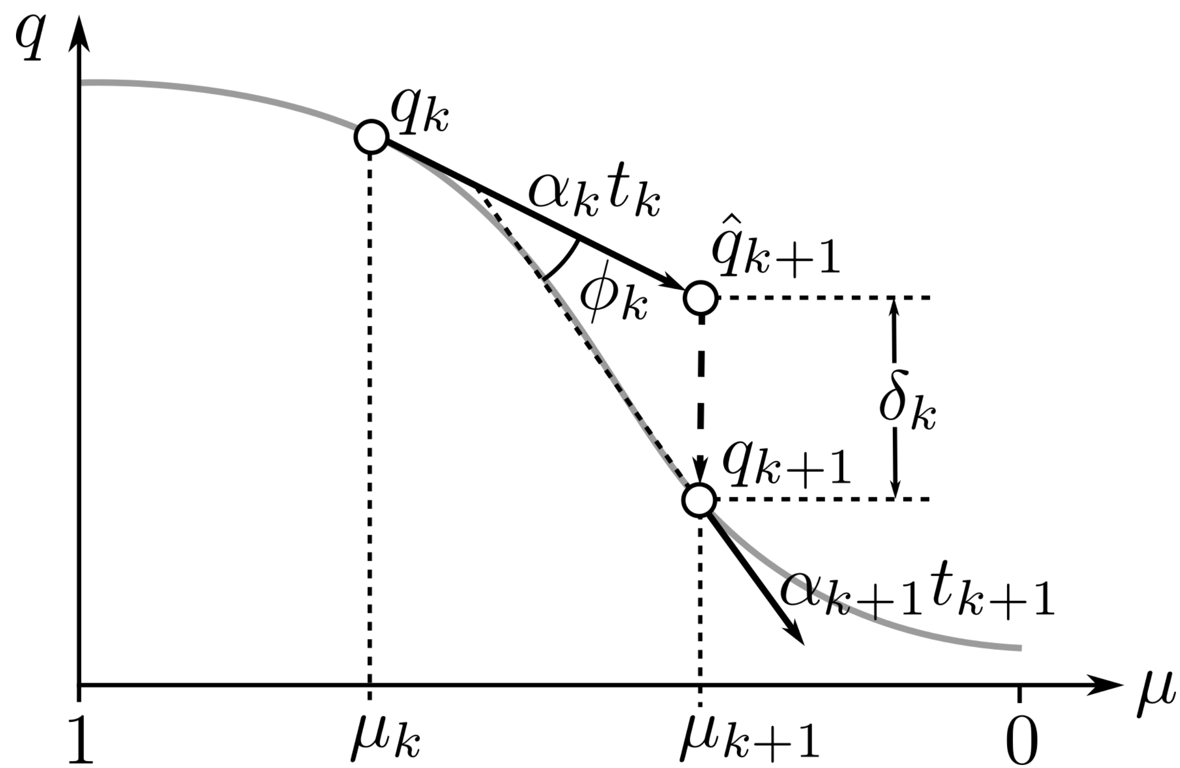
For constrained optimization we use the following convex-homotopy

$$H(q, q_0, \mu) = (1-\mu)F(q) + \mu G(q)$$
$$= (1-\mu)\begin{bmatrix} \nabla_x \mathcal{L} \\ -\mathsf{S}\Lambda_g e \\ h(x) \\ g(x) - s \end{bmatrix} + \mu \begin{bmatrix} x - x_0 \\ s - s_0 \\ -\lambda_h \\ -\lambda_g \end{bmatrix}$$

The Jacobian of the Homotopy function is

$$\nabla_q H = (1-\mu)\nabla_q F(q) + \mu \nabla_q G(q, q_0)$$
$$= (1-\mu)\begin{bmatrix} \nabla_{xx}\mathcal{L} & \mathbf{0} & \nabla_x h^T & \nabla_x g^T \\ \mathbf{0} & -\Lambda_g & \mathbf{0} & -\mathsf{S} \\ \nabla_x h & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \nabla_x g & -\mathsf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix} + \mu \begin{bmatrix} \mathsf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathsf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathsf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathsf{I} \end{bmatrix}$$

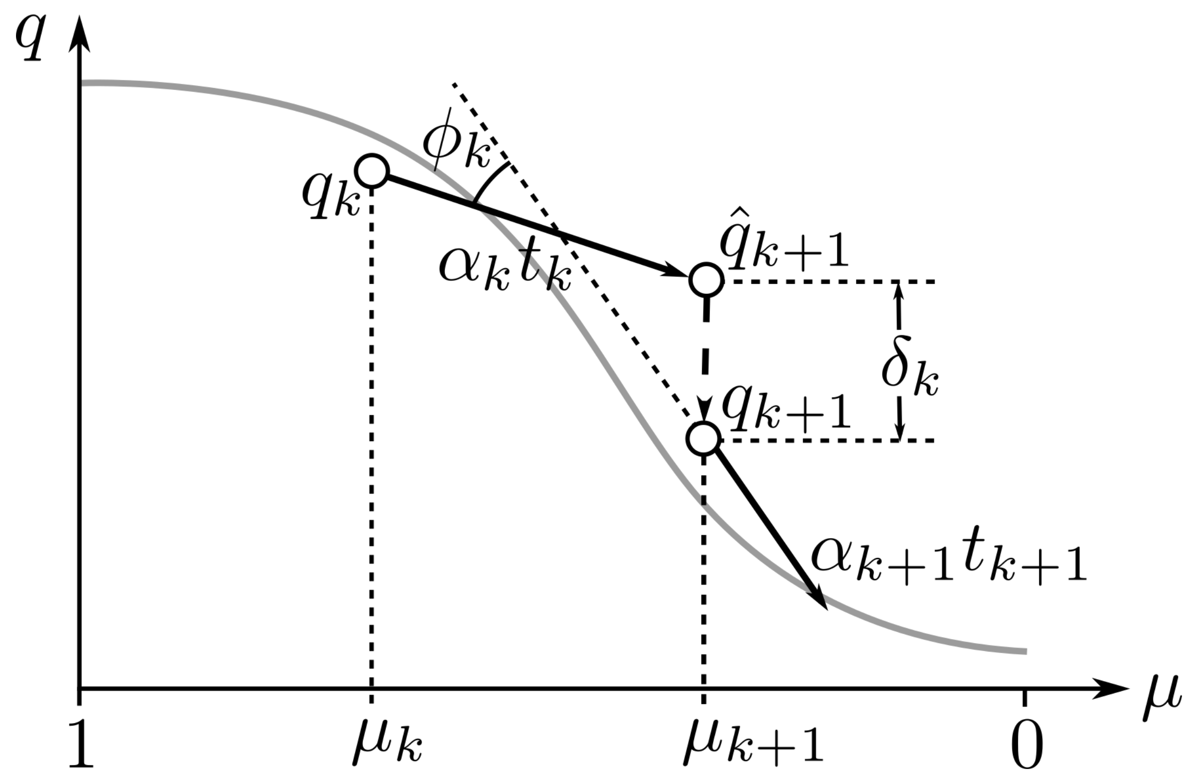
A predictor-corrector algorithm is used to trace the homotopy zero-curve



In predictor step: $(\nabla_q H)_k \, q_k' = -\nabla_\mu H_k = F(q_k) - G(q_k, q_0)$

In corrector step: $H(q_{k+1}, \mu_{k+1}) = 0$

In practice, the predictor and corrector steps use inexact solves



In predictor step: $\|(\nabla_q H)_k\, q_k' - F(q_k) + G(q_k, q_0)\| \le \tau \|F(q_k) - G(q_k, q_0)\|$

In corrector step: $\|H(q_{k+1}, \mu_{k+1})\| \le \epsilon_H \|H(\hat{q}_{k+1}, \mu_{k+1})\|$

The tangent linear problem in the predictor step involves several steps:

1. Inexact Krylov solve:

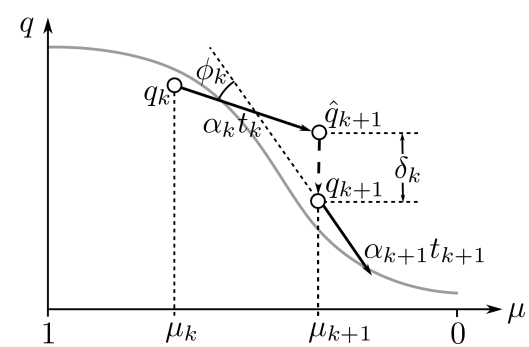
$$\|(\nabla_q H)_k \, q_k' - F(q_k) + G(q_k, q_0)\| \leq \tau \|F(q_k) - G(q_k, q_0)\|$$

2. The tangent direction t_k is obtained from

$$t_k \equiv \frac{1}{\sqrt{\|q_k'\|^2 + 1}} \begin{bmatrix} -q_k' \\ -1 \end{bmatrix}$$

3. The predictor point is calculated as follows:

$$\begin{bmatrix} \hat{q}_{k+1} \\ \mu_{k+1} \end{bmatrix} = \begin{bmatrix} q_k \\ \mu_k \end{bmatrix} + \alpha_k t_k$$

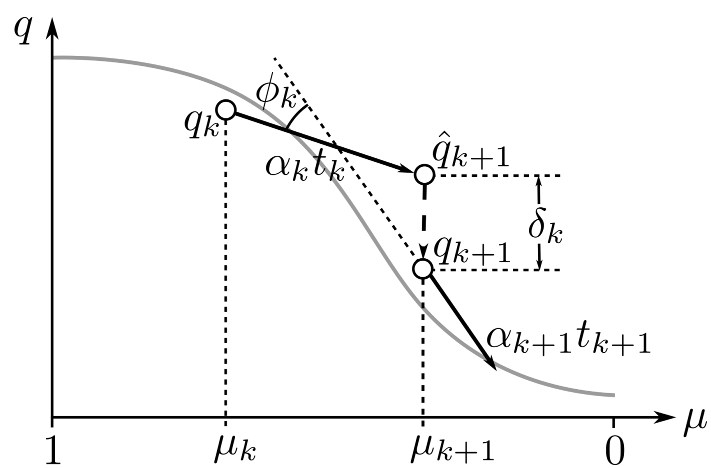


The step-size is controlled adaptively to help keep the predictor close to the zero curve

Initial parameters: $\alpha_0, \delta_{\text{targ}}, \phi_{\text{targ}}$

Step-size factor: $\zeta_k \equiv \max\left(\sqrt{\delta_k/\delta_{\text{targ}}}, \phi_k/\phi_{\text{targ}}\right)$

Step-size: $\alpha_k = \min\left(\sqrt{\|q_k'\|^2 + 1}\Delta\mu_{\max}, \alpha_{\max}, \alpha_{k-1}/\zeta_k\right)$

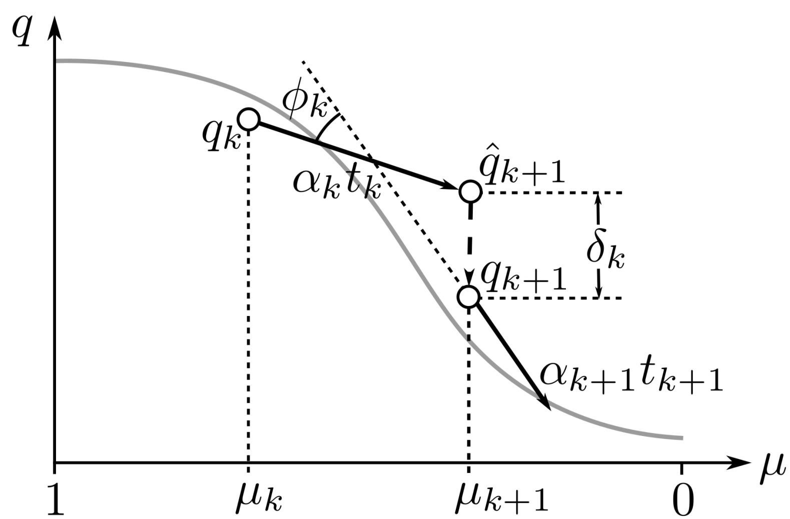


The slack variables and inequality multipliers must be safeguarded

Slack Variable $$\alpha_{\max} = \max\{\alpha \in (0,1] \mid s + \alpha s' \geq \tau_s\}$$

$$s \leftarrow \max(s, \tau_s)$$

Inequality Multipliers $$\lambda_g \leftarrow \min(\lambda_g, 0)$$



Globalization Experiment 1: Sphere Problem

Problem Formulation:

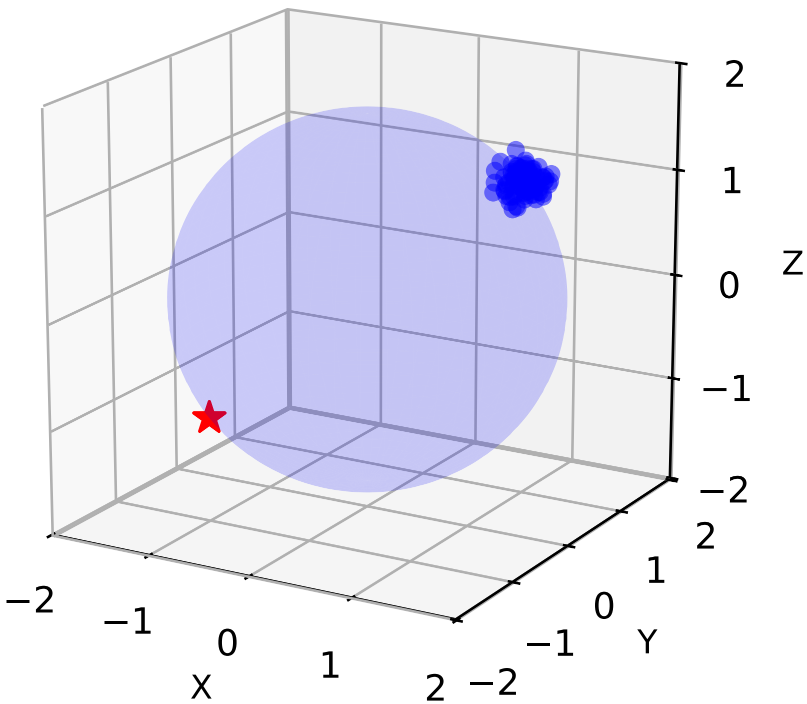
$$\min_{x,y,z} \quad x + y + z$$
$$\text{subject to} \quad x^2 + y^2 + z^2 \leq 3$$

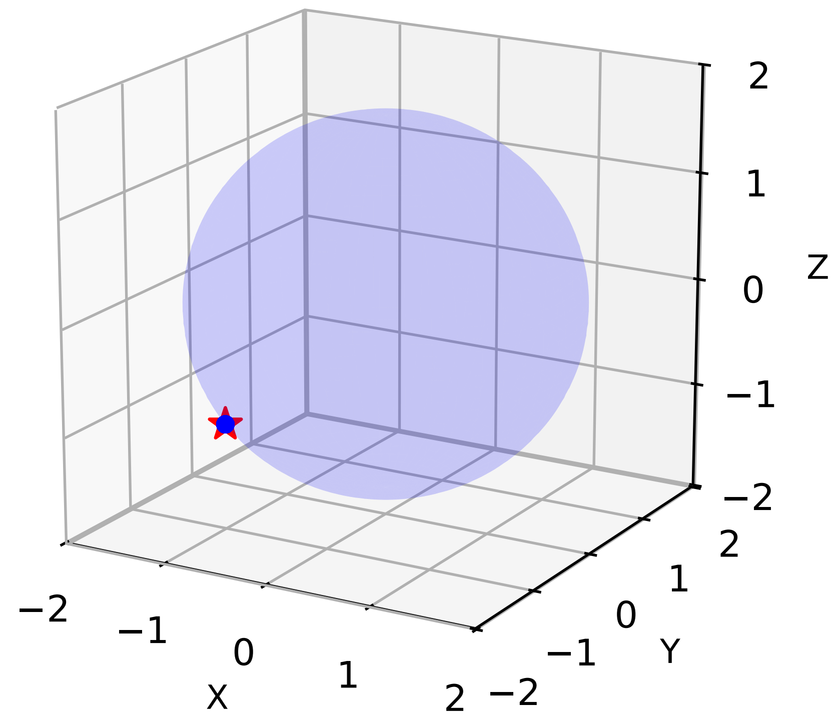
True Solution: $(-1, -1, -1)^T$

Local Maximizer: $(1, 1, 1)^T$

Globalization Experiment 1: Sphere Problem

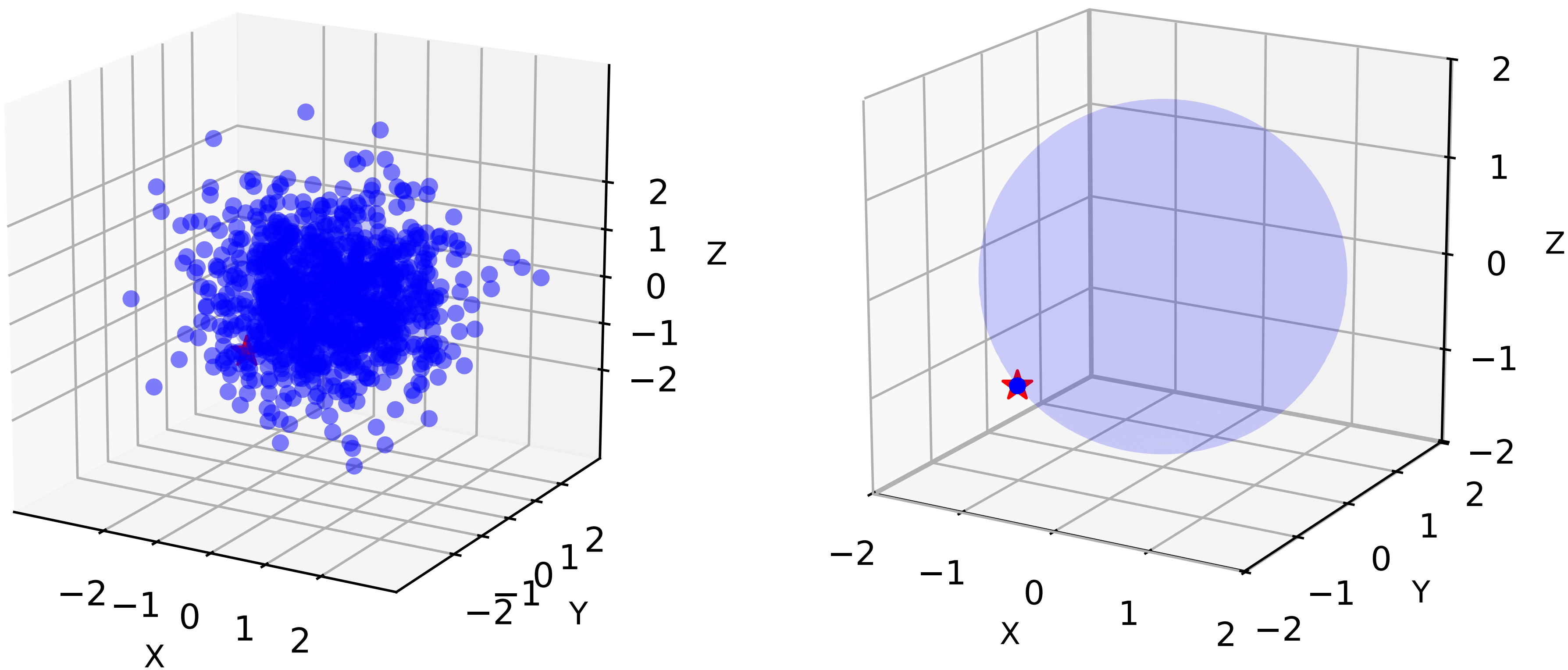
The initial guesses were generated by adding Gaussian perturbations, $\Delta x_i \sim \mathcal{N}(0, 0.1^2)$, to each coordinate of $(1, 1, 1)^T$





Globalization Experiment 1: Sphere Problem

As a second test, the initial guesses were generated by standard normal distribution around the origin



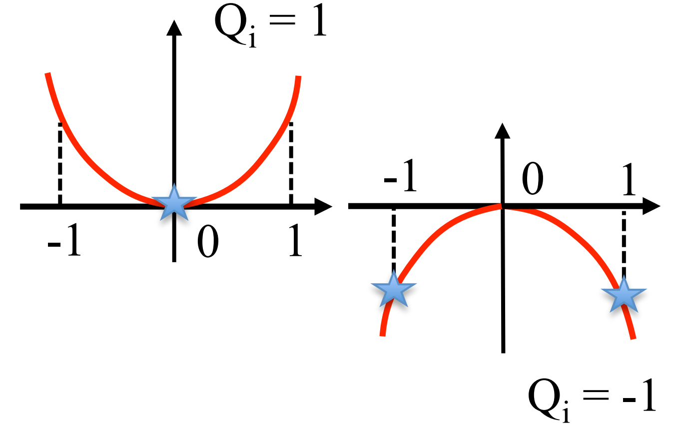
Globalization Experiment 2: Non-convex Problem

Problem Formulation:

$$\min_{x \in R^{100}} \quad \frac{1}{2} x^T \mathsf{Q} x$$
$$\text{subject to} \quad -1 \leq x_i \leq 1 \quad i = 1, 2, \ldots, 100$$

The pattern of the Hessian matrix Q and the initial point x_0 are both randomly generated

$$\mathrm{diag}(\mathsf{Q}) = [1, -1, ...1, -1, -1] \in \mathbb{R}^n$$
$$x_0 \in [-2, 2]$$



Globalization Experiment 2: Non-convex Problem

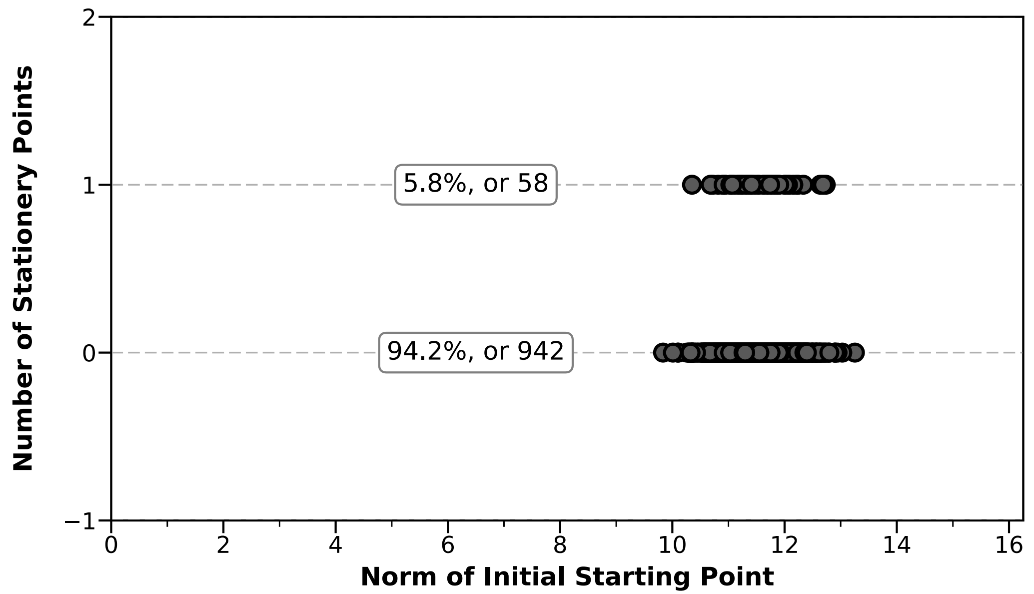
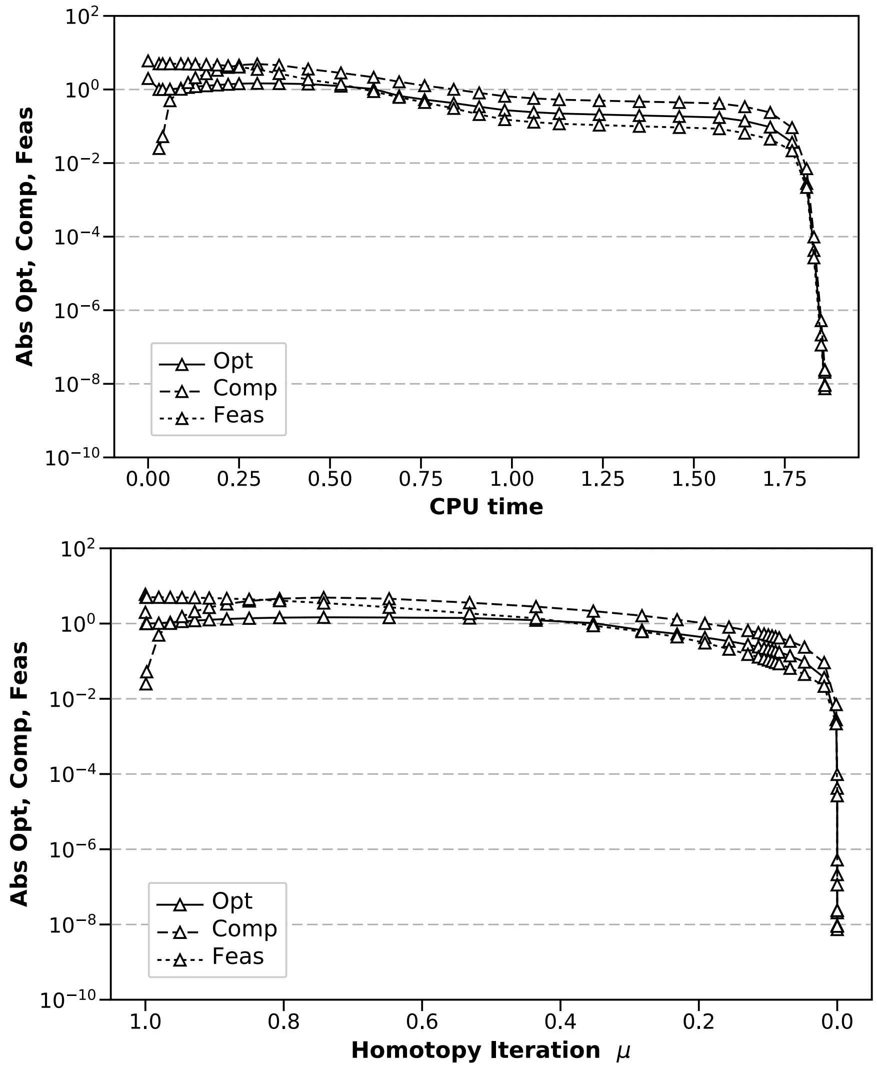


Table 2.1: Success Rate with Different Parameters

		ϵ_{krylov}					
		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
τ and ϵ_H	10^{-1}	51%	90.0%	94.2%	94.6%	93.9%	93.8%
	10^{-2}	47.2%	93.1%	94.4%	93.9%	94.2%	94.5%

Globalization Experiment 2: Non-convex Problem



The optimization criteria are described later

Outline

The most expensive part of the matrix-free algorithm is inexactly solving the linear systems

Predictor Step

$$\|(\nabla_q H)_k \, q'_k - F(q_k) + G(q_k, q_0)\| \leq \tau \|F(q_k) - G(q_k, q_0)\|$$

Newton Step in Corrector

$$\|(\nabla_q H)_{k+1} \, \Delta q_{k+1} + H_{k+1}\| \leq \epsilon_{\mathrm{krylov}} H_{k+1}$$

The linear systems that arise are solved inexactly using FGMRES

In FGMRES, the solution basis Z_i satisfies

$$(\nabla_q H)\mathsf{Z}_i = \mathsf{V}_{i+1}\bar{\mathsf{H}}_i$$

Solve a small least-squares problem to get the solution

$$y_i = \operatorname*{argmin}_{y \in \mathbb{R}^i} \|b - (\nabla_q H)\mathsf{Z}_i y\| = \operatorname*{argmin}_{y \in \mathbb{R}^i} \| \|b\| e_1 - \bar{\mathsf{H}}_i y\|$$

Preconditioner $\quad z_j = P_j(v_j), \quad \forall j = 1, 2, \ldots, i$

Final solution $\quad x_i = \mathsf{Z}_i y_i$

The reduced-space KKT system must be preconditioned

- The true KKT system in optimization is ill-conditioned, FGMRES will stall without an effective preconditioner.
- A preconditioner transforms the linear system to a better conditioned one with the same solution:

$$P_j(u) \approx (\nabla_q H)^{-1} u$$

- General preconditioners based on an explicit matrix are not applicable, e.g. ILU factorizations.

$$\begin{bmatrix} \mathsf{W}_\mu & 0 & \mathsf{A}_{h,\mu}^T & \mathsf{A}_{g,\mu}^T \\ 0 & -\mathsf{\Lambda}_\mu & 0 & -\mathsf{S}_\mu \\ \mathsf{A}_{h,\mu} & 0 & -\mu \mathsf{I} & 0 \\ \mathsf{A}_{g,\mu} & -(1-\mu)\mathsf{I} & 0 & -\mu \mathsf{I} \end{bmatrix} \begin{bmatrix} v_x \\ v_s \\ v_h \\ v_g \end{bmatrix} = \begin{bmatrix} u_x \\ u_s \\ u_h \\ u_g \end{bmatrix}$$

For inequality-only constrained problems, the matrix-free preconditioner involves three steps

Step 1: elimination of the slack and multiplier terms $\left[v_s^T, v_g^T\right]^T$ using the Schur complement

$$\left[\mathsf{W}_\mu + \mathsf{A}_{g,\mu}^T \mathsf{C}_\mu^{-1} \Lambda_\mu \mathsf{A}_{g,\mu}\right] v_x = u_x - \mathsf{A}_{g,\mu}^T \mathsf{C}_\mu^{-1} \left[(1-\mu) u_s - \Lambda_\mu u_g\right]$$

$$\begin{bmatrix} v_s \\ v_g \end{bmatrix} = \begin{bmatrix} \mathsf{C}_\mu^{-1} & 0 \\ 0 & \mathsf{C}_\mu^{-1} \end{bmatrix} \begin{bmatrix} -\mu \mathsf{I} & \mathsf{S}_\mu \\ (1-\mu)\mathsf{I} & -\Lambda_\mu \end{bmatrix} \begin{bmatrix} u_s \\ u_g - \mathsf{A}_{g,\mu} v_x \end{bmatrix}$$

$$\mathsf{C}_\mu \equiv \mu \Lambda_\mu - (1-\mu)\mathsf{S}_\mu = \mu(1-\mu)\Lambda_g - \mu^2 \mathsf{I} - (1-\mu)^2 \mathsf{S}$$

For inequality-only constrained problems, the matrix-free preconditioner involves three steps

Step 2: Use a Lanczos SVD approximation for the constraint Jacobian terms

$$\tilde{\mathsf{A}}_{g,\mu}^{T}\mathsf{C}_{\mu}^{-1}\Lambda_{\mu}\tilde{\mathsf{A}}_{g,\mu} = M_{n\times k}\Gamma_{k\times k}N_{k\times n}^{\star}$$

Step 3: Apply the Sherman-Morrison formula to obtain the desired design update

$$\begin{aligned}\left[\mathsf{W}_{\mu} + \mathsf{A}_{\mu}^{T}\mathsf{C}_{\mu}^{-1}\Lambda_{\mu}\mathsf{A}_{\mu}\right]^{-1} &\approx \left[\mathsf{B}_{\mu} + \mathsf{U}\Sigma\mathsf{V}^{T}\right]^{-1} \\ &= \mathsf{B}_{\mu}^{-1} - \mathsf{B}_{\mu}^{-1}\mathsf{U}\left(\mathsf{I}_{n_{\Sigma}} + \Sigma\mathsf{V}^{T}\mathsf{B}_{\mu}^{-1}\mathsf{U}\right)^{-1}\Sigma\mathsf{V}^{T}\mathsf{B}_{\mu}^{-1}\end{aligned}$$

When equality constraints are included, the Schur complement is augmented

$$\begin{bmatrix} \mathsf{W}_\mu + \mathsf{A}_{g,\mu}^T \mathsf{C}_\mu^{-1} \Lambda_\mu \mathsf{A}_{g,\mu} & \mathsf{A}_{h,\mu}^T \\ \mathsf{A}_{h,\mu} & -\mu \mathsf{I} \end{bmatrix} \begin{bmatrix} v_x \\ v_h \end{bmatrix} = \begin{bmatrix} u_x - \mathsf{A}_{g,\mu}^T \mathsf{C}_\mu^{-1} \left[(1-\mu)u_s - \Lambda_\mu u_g\right] \\ u_h \end{bmatrix}$$

To solve this, μ is kept away from zero in the preconditioner using the clipping rule $\bar{\mu} = \max(\mu, 10^{-4})$; consequently, the system can be solved approximated as follows.

$$\begin{bmatrix} v_x \\ v_h \end{bmatrix} = \begin{bmatrix} I & 0 \\ \frac{1}{\bar{\mu}} \mathsf{A}_{h,\mu} & I \end{bmatrix} \begin{bmatrix} \left(\mathsf{W}_\mu + \mathsf{A}_{g,\mu}^T \mathsf{C}_\mu^{-1} \Lambda_\mu \mathsf{A}_{g,\mu} + \frac{1}{\bar{\mu}} \mathsf{A}_{h,\mu}^T \mathsf{A}_{h,\mu}\right)^{-1} & 0 \\ 0 & -\frac{1}{\bar{\mu}} \mathsf{I} \end{bmatrix} \begin{bmatrix} I & \frac{1}{\bar{\mu}} \mathsf{A}_{h,\mu}^T \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{u}_x \\ u_h \end{bmatrix}$$

The Lanczos SVD approximation is applied to the sum of the equality and inequality Jacobian terms

The algorithms have been implemented in a Python library[44]

- Algorithms have been implemented in Kona, a matrix-free optimization package for PDE-constrained problems

- Kona uses the following convergence metrics:

$$\text{Optimality} = \max_j |(\nabla_x \mathcal{L})_j|,$$

$$\text{Complementarity} = \max_j |s_j \lambda_j|,$$

$$\text{Feasibility} = \max_j |c_j|, \quad \text{where} \quad c = \left[h(x)^T, (g(x) - s)^T\right]$$

- For benchmarking, we compare against SNOPT, which is a state-of-the-art active-set algorithm frequently used in the engineering optimization community

Preconditioner Experiment: Scalable Quadratic Problem

Problem Formulation:

$$\min_{x \in R^n} \quad \frac{1}{2} x^T \mathsf{Q} x + g^T x$$
$$\text{subject to} \quad \mathsf{A} x \geq b$$

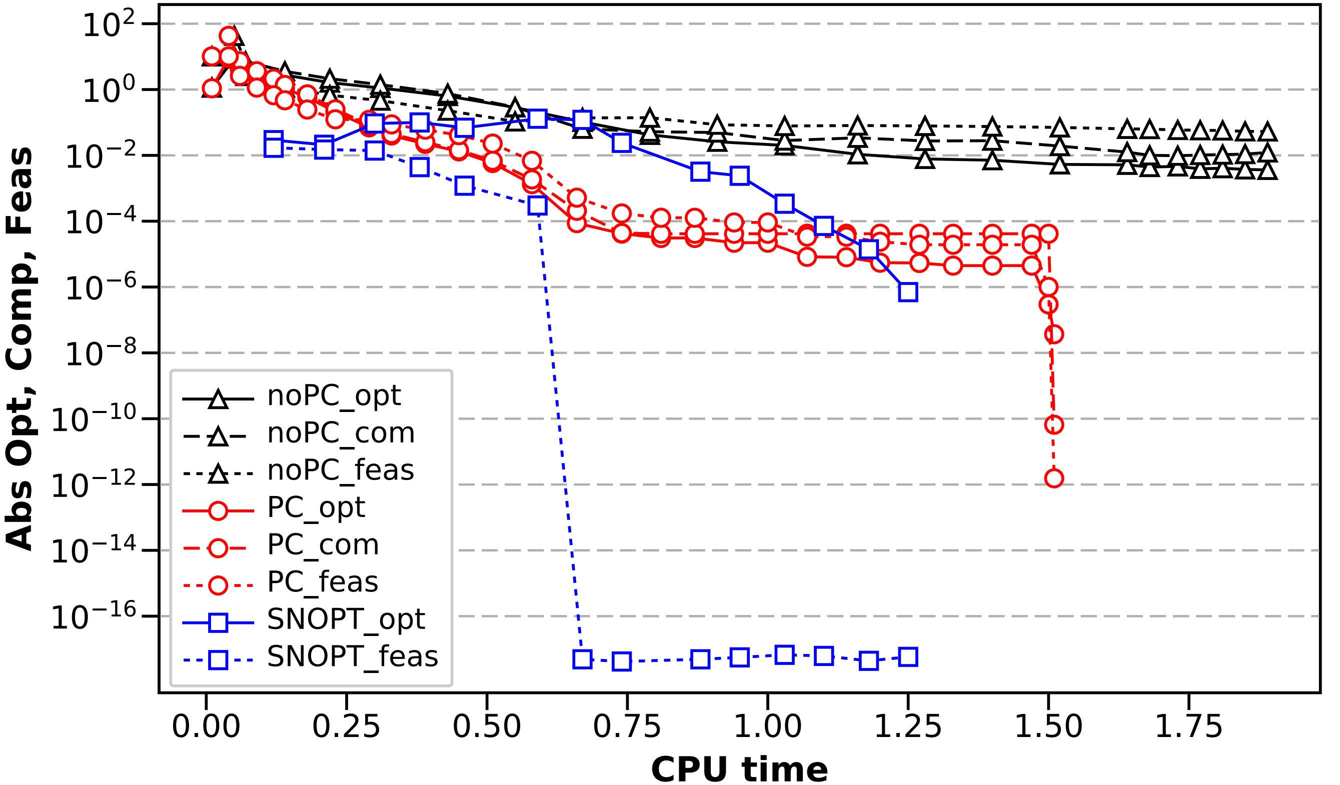
The singular values of Q and A are predefined:

$$\mathsf{Q}_{ii} = \begin{cases} \frac{1}{i}, & i = 1, 2, ..., \kappa, \\ \frac{1}{\kappa}, & i = \kappa + 1, ..., n, \end{cases}$$

$$\mathsf{A} = \mathsf{Q}_L \mathsf{D} \mathsf{Q}_R \quad \begin{cases} \mathsf{D}_{ii} = \begin{cases} \frac{1}{i^2}, & i = 1, 2, ..., \nu, \\ \frac{1}{\nu^2}, & i = \nu + 1, ..., n, \end{cases} \\ \mathsf{Q}_L \quad \mathsf{Q}_R \quad \text{Randomly generated orthonormal matrix} \end{cases}$$

This simple example illustrates the effectiveness of the preconditioner

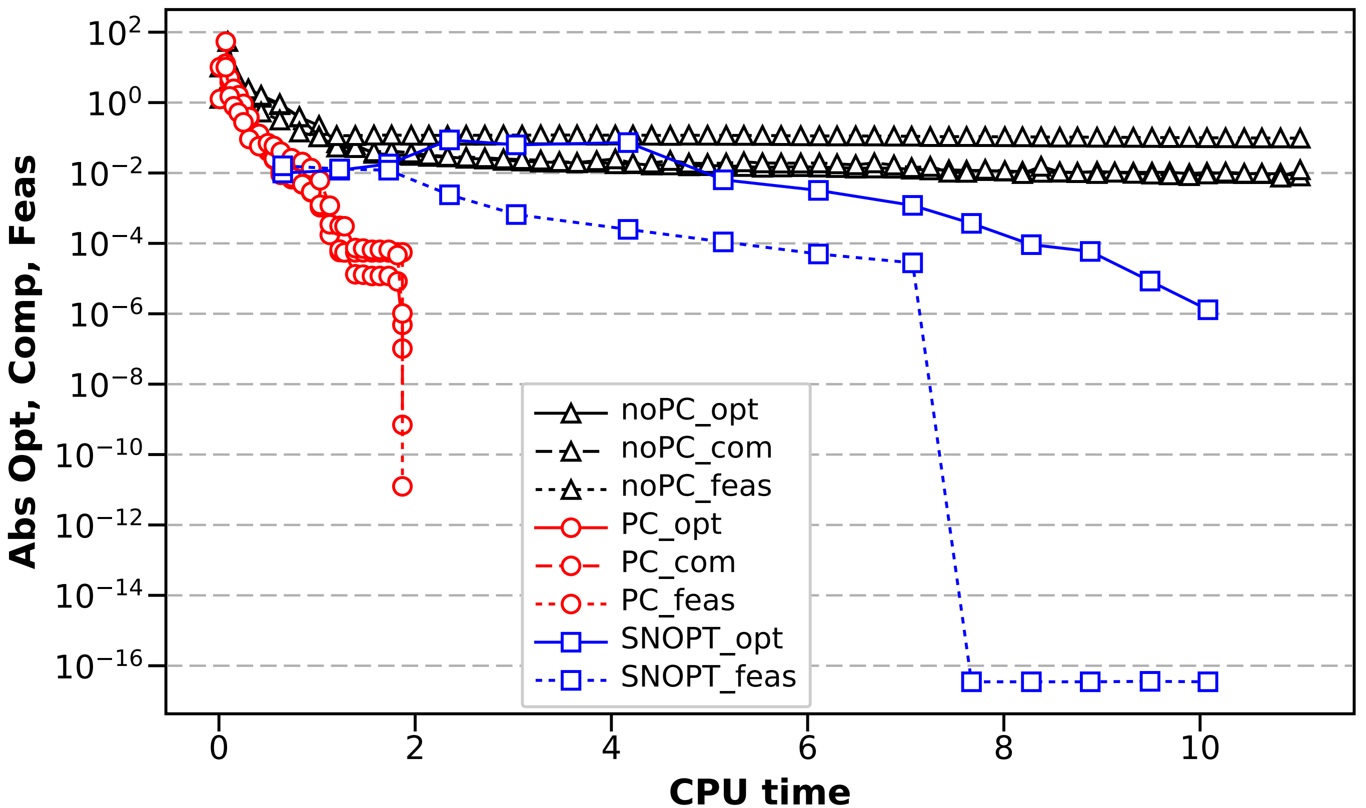
200 design variables



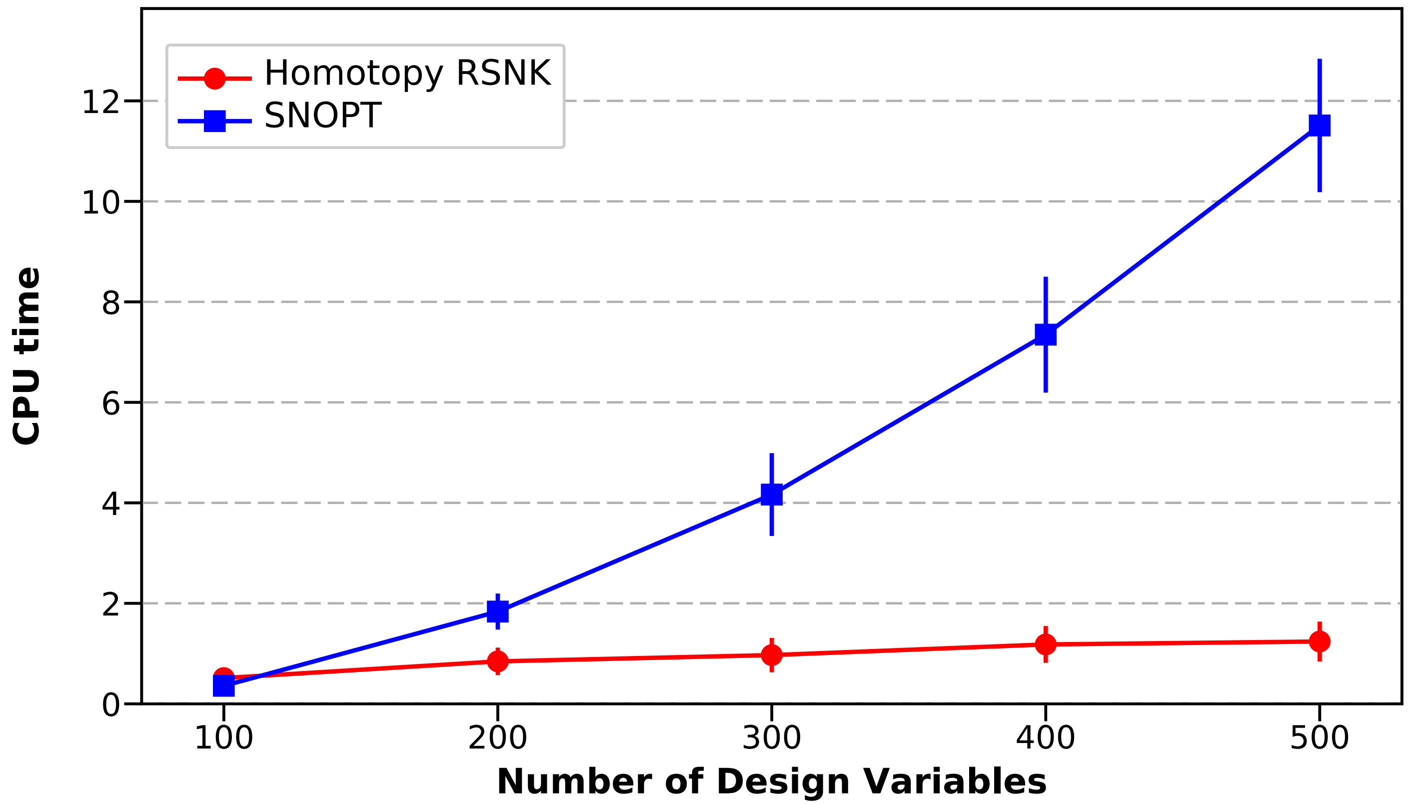
The optimization criteria are defined later

This simple example illustrates the effectiveness of the preconditioner

500 design variables



The proposed preconditioner helps the inexact-Newton algorithm scale well with problem size



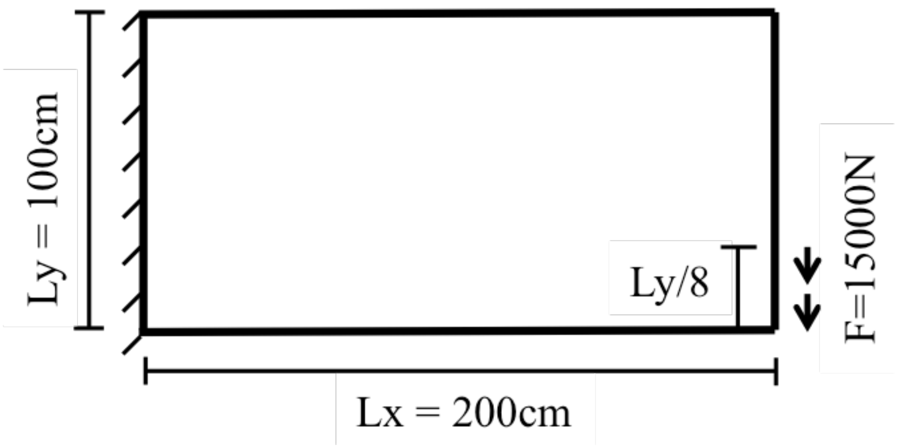
Outline

The algorithms were verified on a subset of the CUTEr test suite.

A summary of Table 4.1, which contains the test results of a subset of the CUTEr problems:

- Overall 63 problems are tested
- Kona accurate: 60
- SNOPT accurate: 39

PDE-governed problem 1: Stress-Constrained Mass Minimization

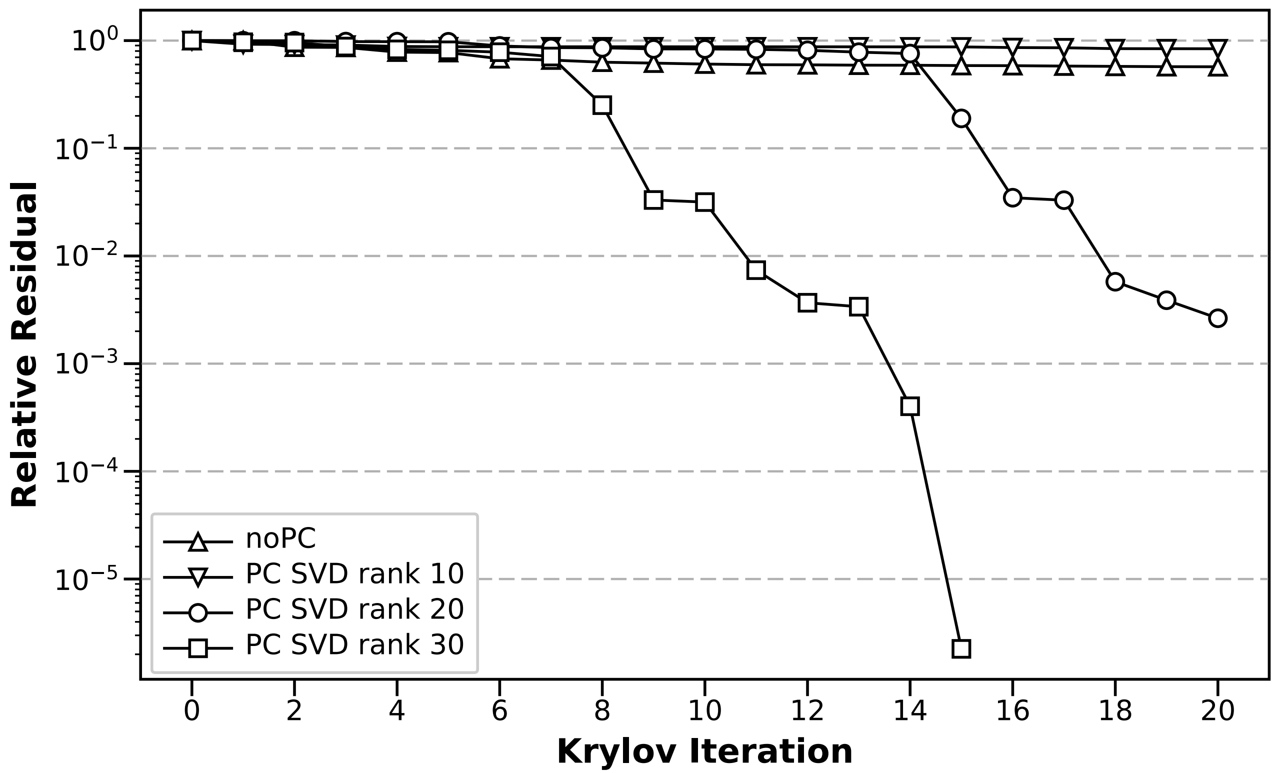


Case	nx	ny	Number of design
Small	16	8	128
Medium	32	16	512
Large	64	32	2048

$$\begin{aligned} \min_{x} \quad & \mathsf{mass}(x) \\ \text{subject to} \quad & \mathsf{stress}_i(x) \le \sigma_{\max}, \forall\, i = 1, 2, \ldots, n, \\ & x_l \le x_i \le x_u, \qquad \forall\, i = 1, 2, \ldots, n, \end{aligned}$$

The governing PDE models the response of the plate to the applied force using linear elasticity with 2D plane stress

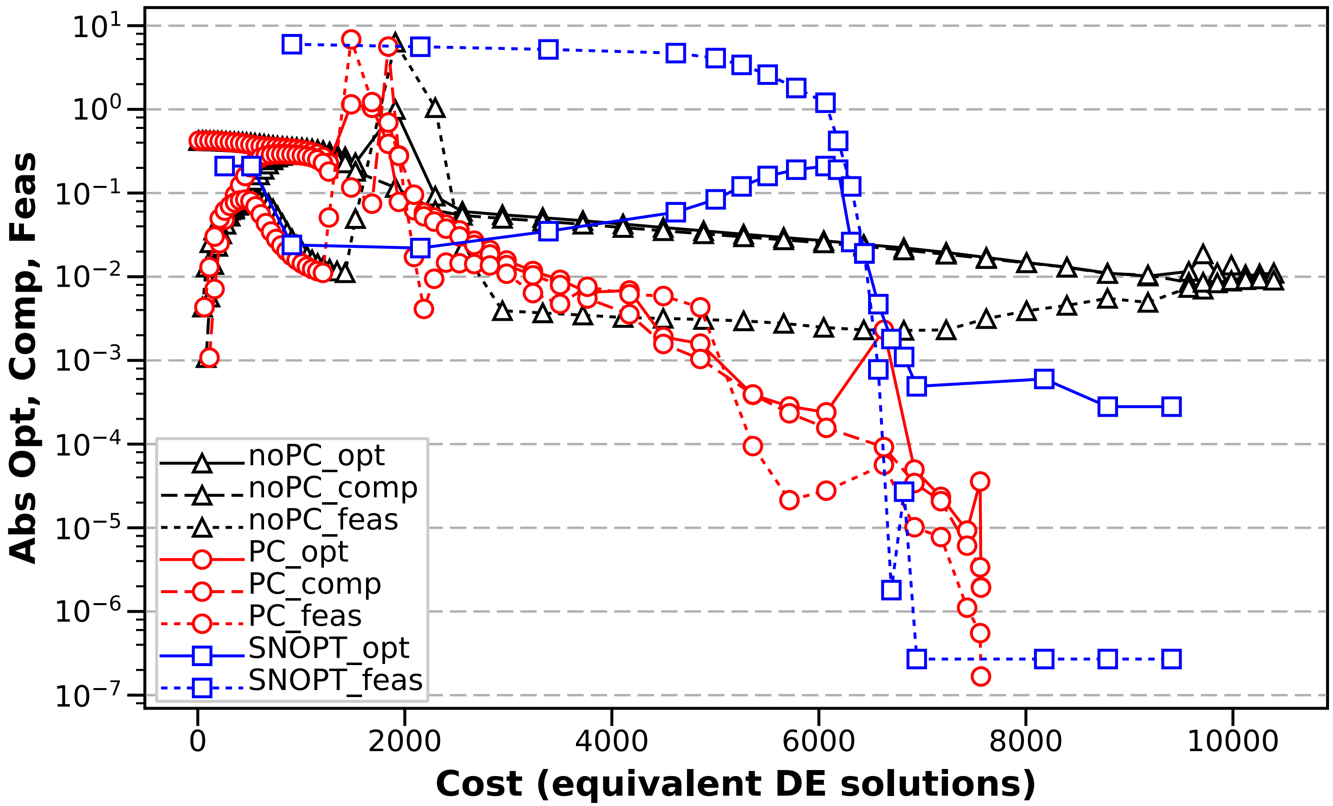
The preconditioner is required to ensure convergence of the Krylov iterative method



Krylov-solver convergence history corresponding to the first Newton for the first Newton step at the final corrector phase for the small case

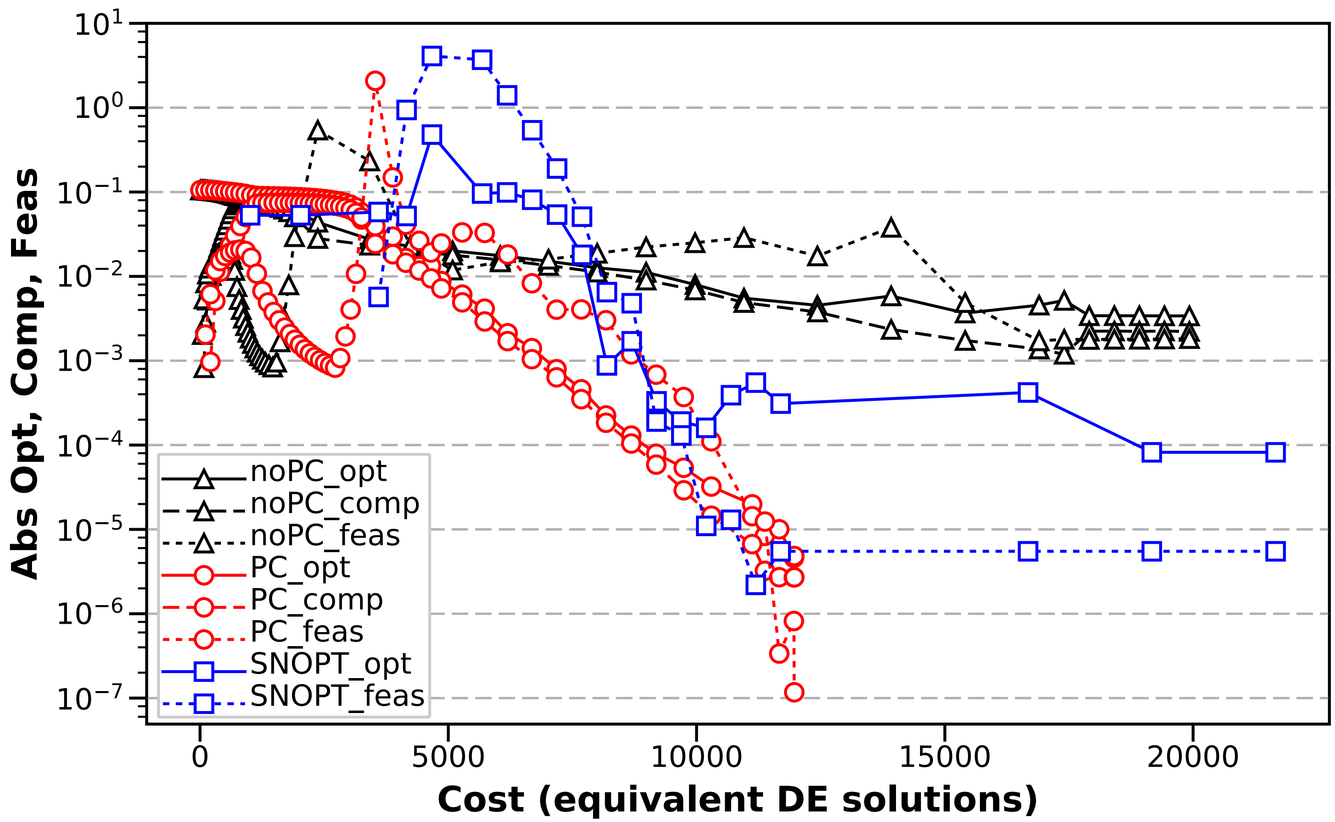
The proposed algorithm is able to converge the problem on all three grids

Small

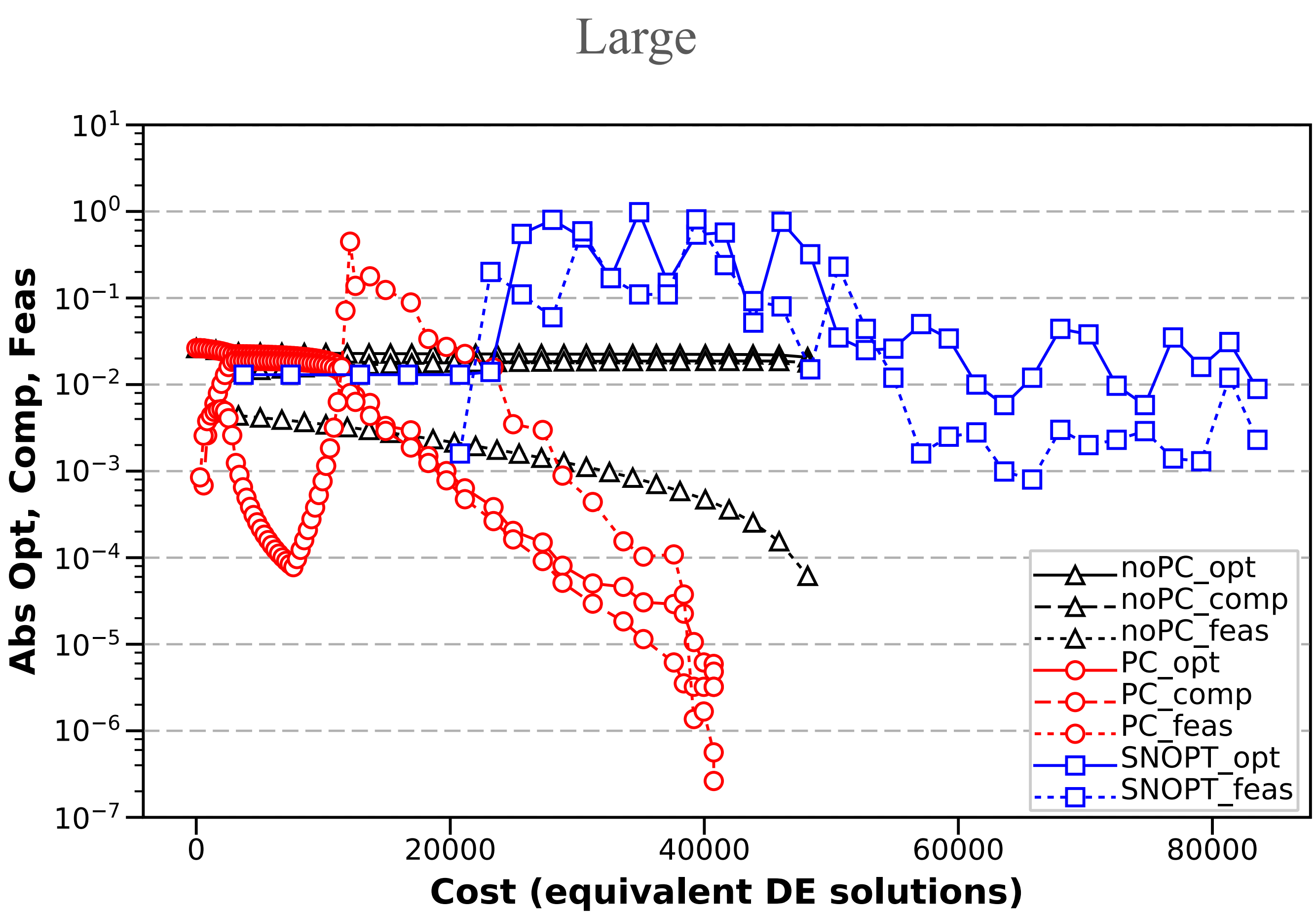


The proposed algorithm is able to converge the problem on all three grids

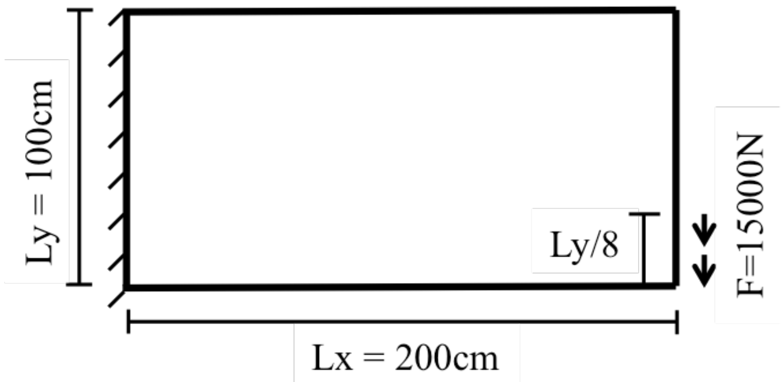
Medium

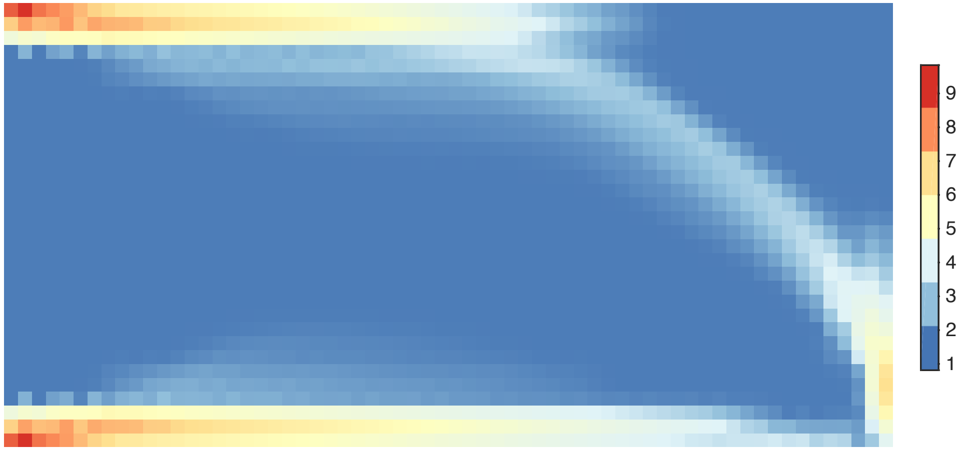


The proposed algorithm is able to converge the problem on all three grids

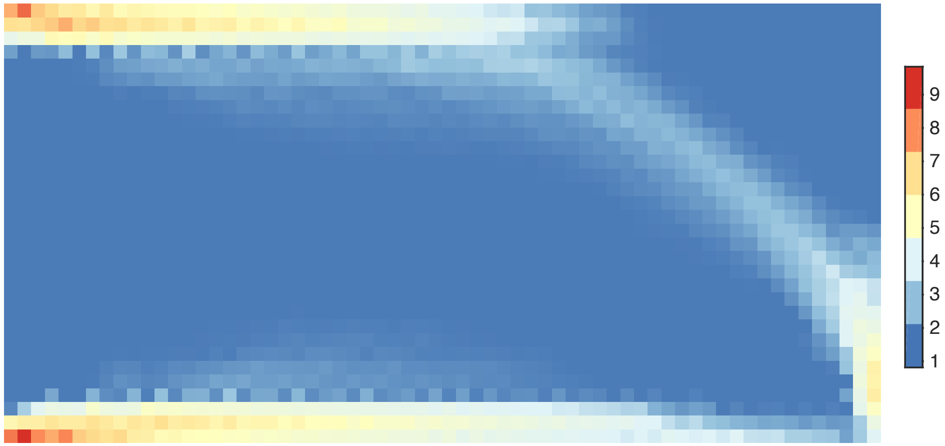


The thickness distribution on all three grids

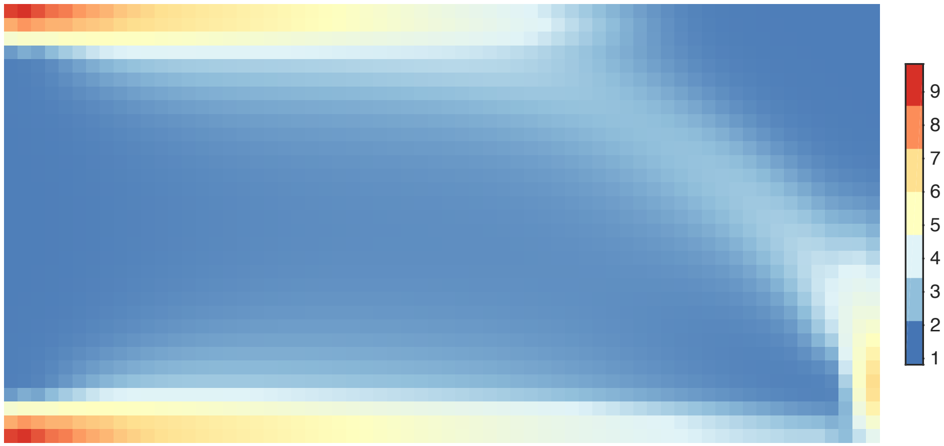




New method with preconditioner

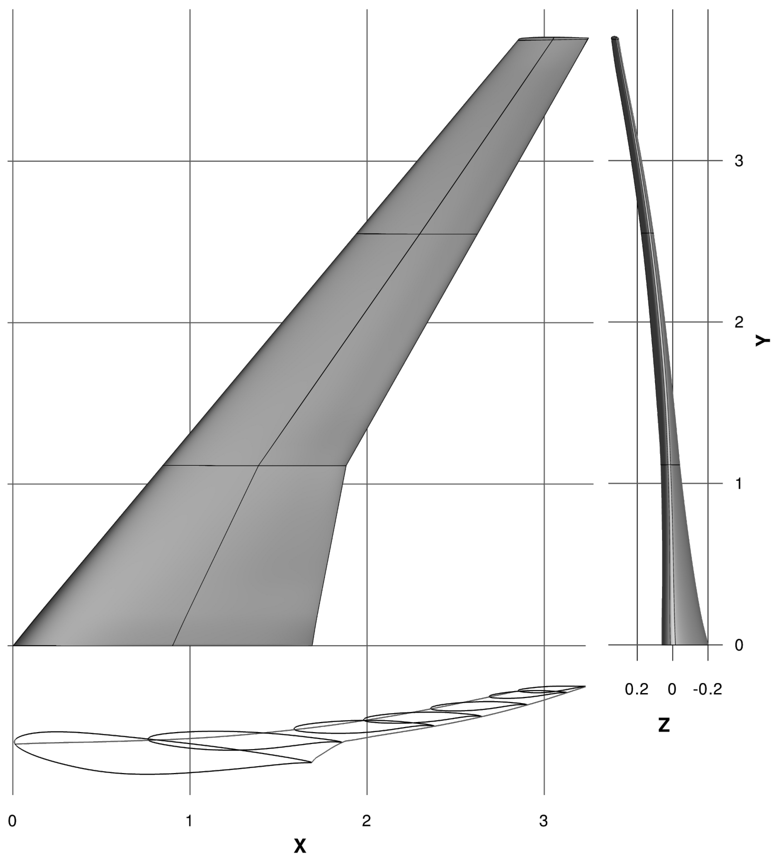


SNOPT



New method without preconditioner

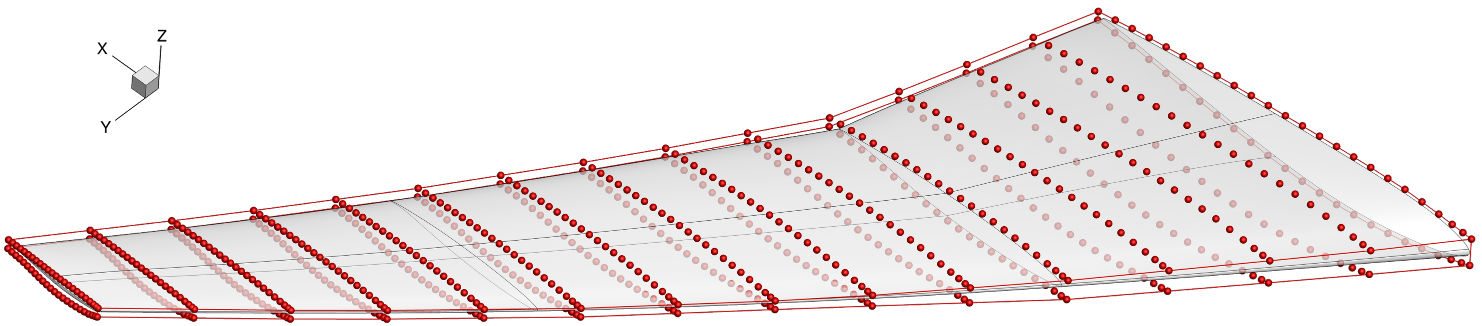
PDE-governed problem 2: Aerodynamic Shape Optimization Problem



$$
\begin{aligned}
\min_{x} \quad & C_D \\
\text{subject to} \quad & C_L \geq 0.5 \\
& C_{M_y} \geq -0.17 \\
& t \geq 0.25 t_{\text{base}} \\
& V \geq V_{\text{base}} \\
& \Delta z_{TE,\text{upper}} = -\Delta z_{TE,\text{lower}} \\
& \Delta z_{LE,\text{upper, root}} = -\Delta z_{LE,\text{lower, root}} \\
\text{governed by} \quad & \mathcal{R}_{\text{Euler}} = 0
\end{aligned}
$$

NASA Common Research Model (CRM) wing defined by AIAA/ADODG

Free-form deformation is used to parameterize the wing shape

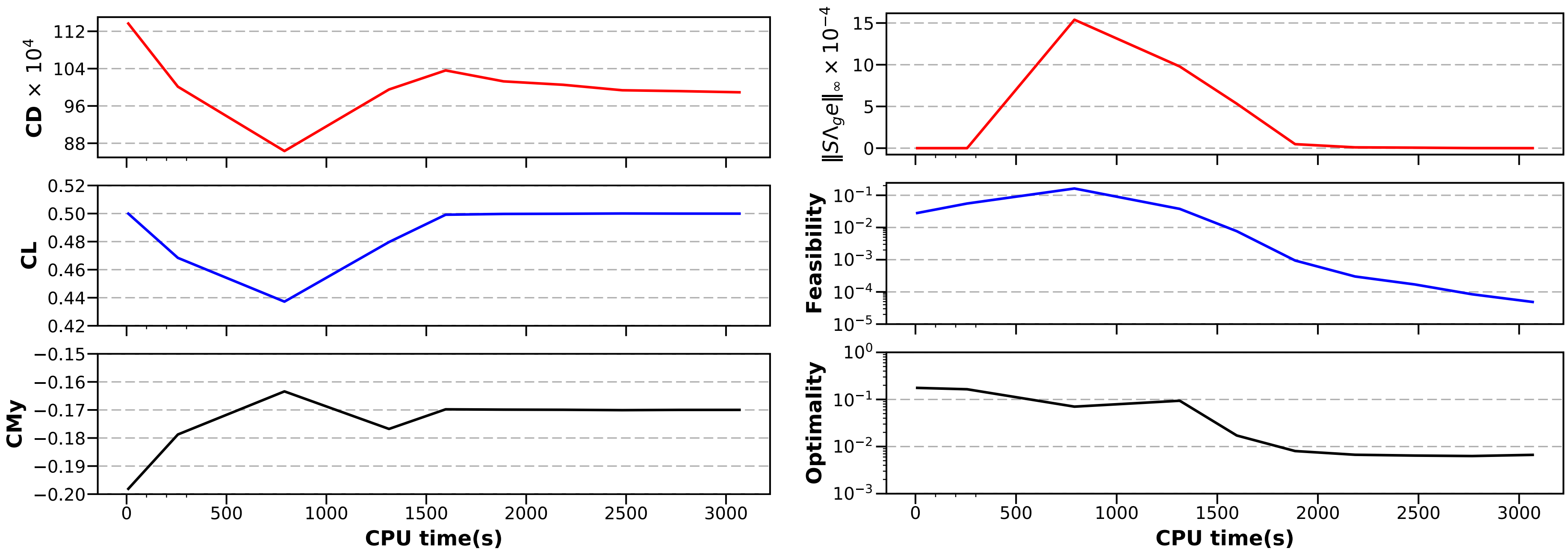


Three different sized FFD control volumes were considered

	192	**480**	**768**
Chordwise	12	20	24
Spanwise	8	12	16
Vertical	2	2	2

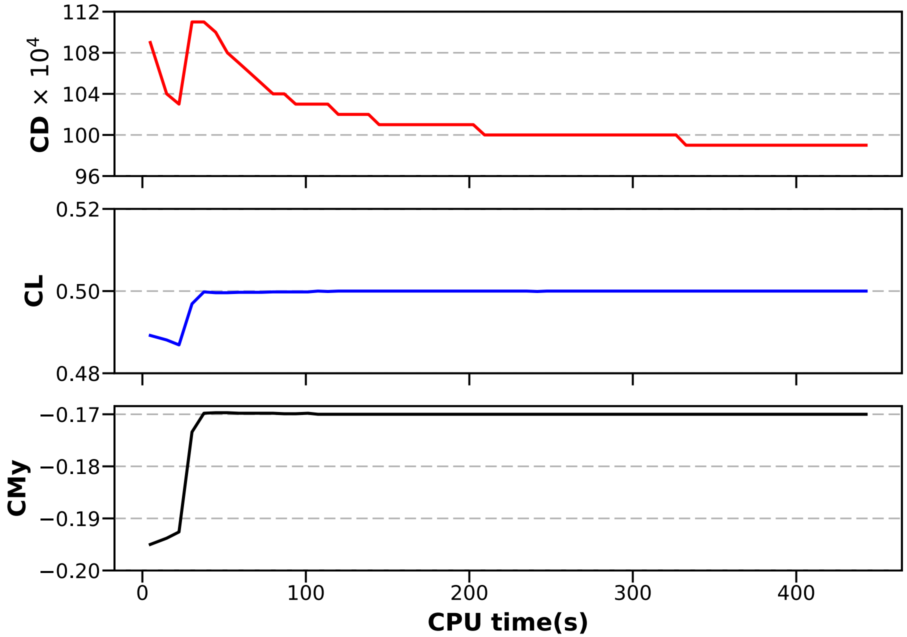
Aerodynamic coefficient history and optimization plots from Kona

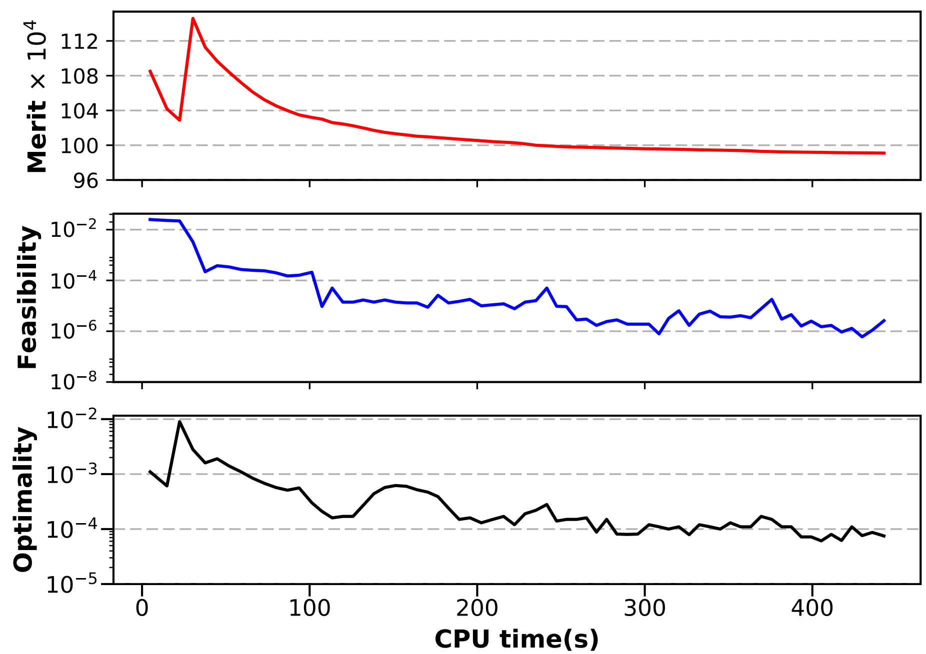
Number of Design: 768



Aerodynamic coefficient history and optimization plots from SNOPT

Number of Design: 768





Outline

Contributions

- Developed a reduced-space Newton-Krylov method for general constrained PDE-governed design problems
- Used a Homotopy-based method to increase the reduced-space inexact-Newton method's globalization capacity
- Created a novel and effective matrix-free preconditioner to accelerate the convergence rate of the iterative methods

Recommendations

- No "silver bullet" preconditioner: provide preconditioners for different types of problems in Kona
- Separate the nonlinear and linear constraints in Kona's API
- Improve the robustness of the method in the context of non-convex problems
- Improve the efficiency of the preconditioner for ASO problem and run the RANS-based optimization
- Investigate methods to automate the parameter values

Thank you!
Questions?

Backup Slide

$$\mathsf{W}_\mu + \mathsf{A}_\mu^T \Sigma_\mu \mathsf{A}_\mu = \mathsf{W}_\mu + \mathsf{A}_{g,\mu}^T \mathsf{C}_\mu^{-1} \Lambda_\mu \mathsf{A}_{g,\mu} + \frac{1}{\bar{\mu}} \mathsf{A}_{h,\mu}^T \mathsf{A}_{h,\mu}$$

$$\mathsf{A}_\mu = \begin{bmatrix} \mathsf{A}_{h,\mu} \\ \mathsf{A}_{g,\mu} \end{bmatrix} \qquad \text{and} \qquad \Sigma_\mu = \begin{bmatrix} \frac{1}{\bar{\mu}} \mathsf{I} & 0 \\ 0 & \mathsf{C}_\mu^{-1} \Lambda_\mu \end{bmatrix}$$

$$\left(\nabla_q H\right)_{k+1} \Delta q_{k+1} = -H_{k+1}$$

The optimization package Kona:

- is a matrix-free optimization package designed to solve reduced-space PDE-constrained problems
- separates the optimization algorithms from PDE-solver specific implementations
- has a solver interface that provides Jacobian-vector and vector-Jacobian products that are necessary to assemble total sensitivities