Multi-Objective Stochastic Optimization of Magnetic Fields

David S Bindel, Cornell University

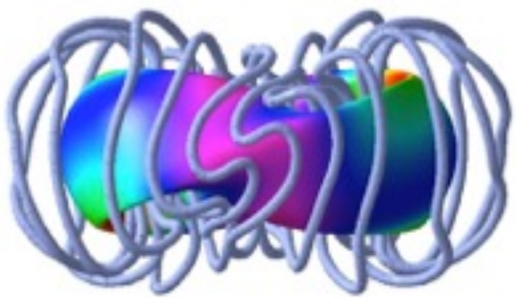
Current State of the Art (STELLOPT)

Optimizer

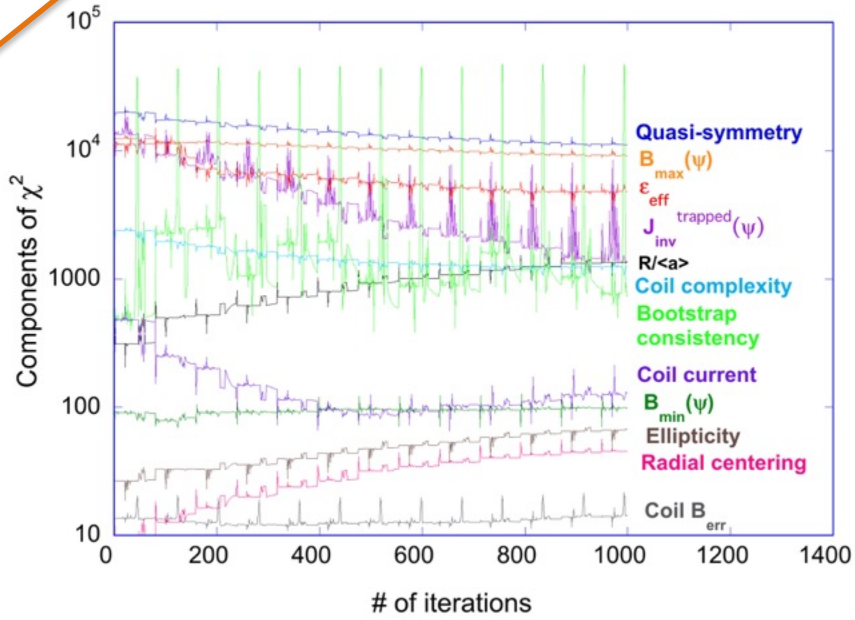
Calculate χ^2
(physics + engineering targets)

Adjust plasma boundary
(or coil shape)

Solve 3D
equilibrium



$$r(\phi,\theta) + iz(\phi,\theta) = \sum \alpha_{m,n} e^{i(m\phi - n\theta)}$$



STELLOPT Approach

Goal: Design MHD equilibrium (coil optimization often separate)

- Possible parameters for boundary: $C \subset \mathbb{R}^n$
- Physics / engineering properties: $F : C \subset \mathbb{R}^n \to \mathbb{R}^m$
- Target vector $F^* \in \mathbb{R}^m$

Minimize χ^2 objective over C:

$$\chi^2(x) = \sum_{k=1}^{m} \sigma_k^{-2} J_k(x), \qquad J_k(x) = (F_k(x) - F_k^*)^2$$

Solve via Levenberg-Marquardt, GA, differential evolution (avoids gradient information apart from finite differences)

Challenges

1. Costly and “black box” physics computations

   Each step: MHD equilibrium solve, transport calculation, coil design…

   Compute several times per step for finite-difference gradient estimates!

2. Managing tradeoffs

   How do we choose the weights in the χ^2 measure? By gut?

   Does varying the weights expose tradeoffs in a sensible way? No!

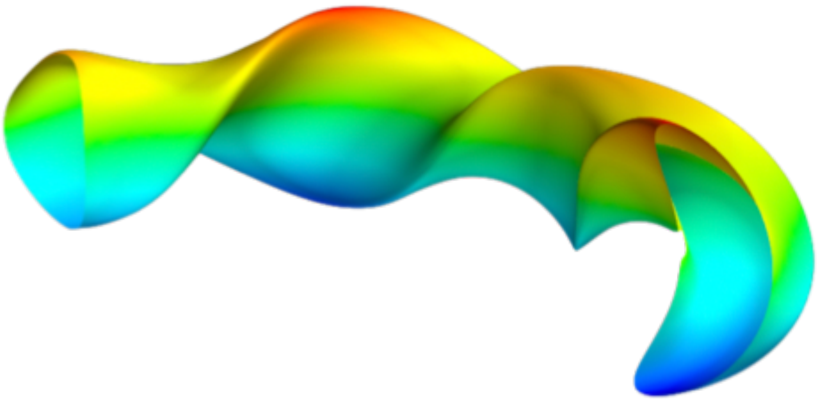
3. Dealing with uncertainties

   What you simulate $\neq$ what you build – will performance suffer?

4. Global search

   How do we avoid getting stuck in local minima without excessive cost?

Challenge 1: Costly Physics Constraints



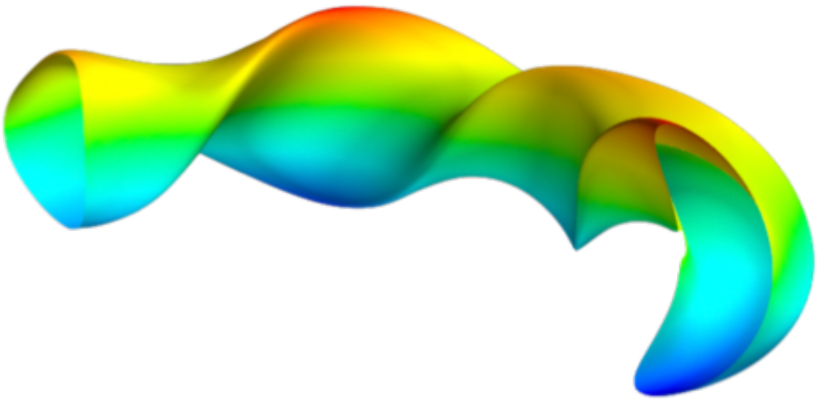
Beltrami field (Taylor state):

$$\nabla \times B = \lambda B \text{ on } \Omega$$
$$B \cdot n = 0 \text{ on } \partial\Omega$$
$$\nabla \cdot B = 0$$

+ flux conditions for well-posedness

- Requires costly physics solves (MHD equilibrium, transport, …)
- Derivatives require PDE sensitivity / adjoints (not black-box)

Physics-Constrained Optimization



Beltrami field (Taylor state):

$$\nabla \times B = \lambda B \text{ on } \Omega$$
$$B \cdot n = 0 \text{ on } \partial\Omega$$
$$\nabla \cdot B = 0$$

+ flux conditions for well-posedness

- Key: Exploit PDE properties
  - PDE-constrained: Solves are part of the optimization, not a black box
  - PDE structure influences optimization objective landscape
  - PDE operator properties: compactness, smoothing, near/far field interactions, etc
  - Provides opportunities for dimension reduction in optimization

Challenge 2: Multi-Objective Optimization

What makes an "optimal" stellarator?

- Approximates field symmetries (which measures?)
- Satisfies macroscopic and local stability
- Includes divertor fields for particle and heat exhaust
- Minimizes collisional and energetic particle transport
- Minimizes turbulent transport
- Satisfies basic engineering constraints (cost, size, etc)

Each objective involves different approximations, uncertainties, and computational costs.

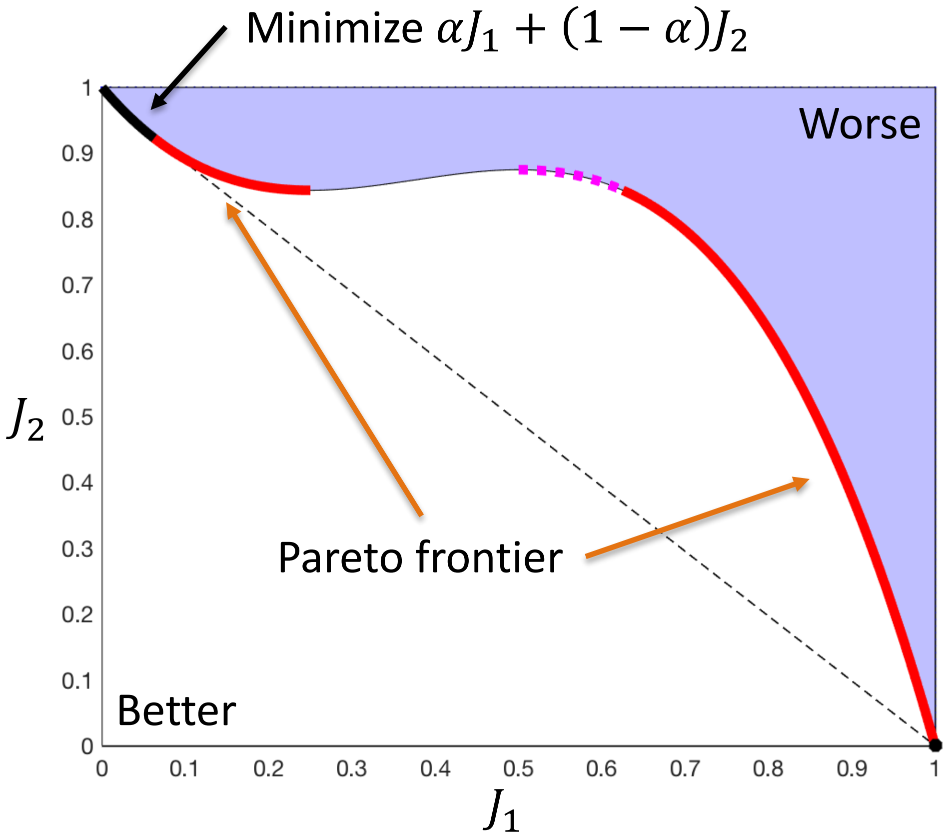
Exploring the Pareto Frontier

x *dominates* y if
$J(x) \neq J(y)$ and $\forall k, J_k(x) \leq J_k(y)$

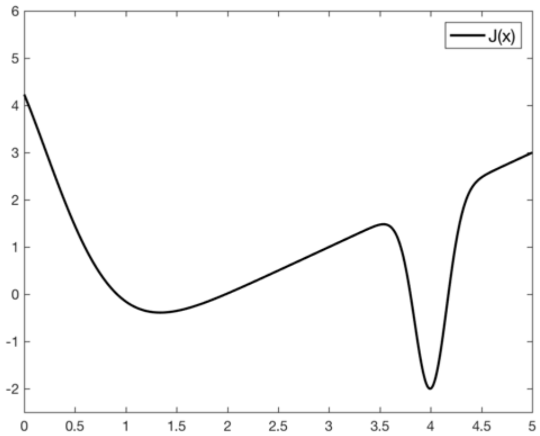
Pareto optimal (*non-dominated, non-inferior, efficient*): no y dominates x.

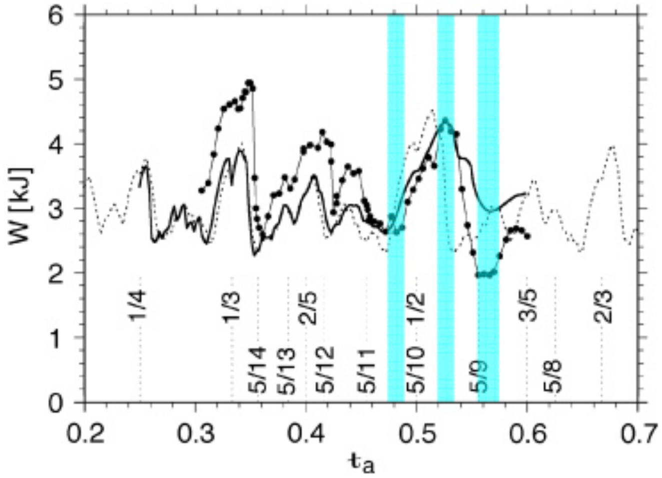
Pareto frontier generally an $(m-1)$-dimensional manifold with corners.

Minimizing $\sum_k \alpha_k J_k$ only explores convex hull of Pareto frontier!



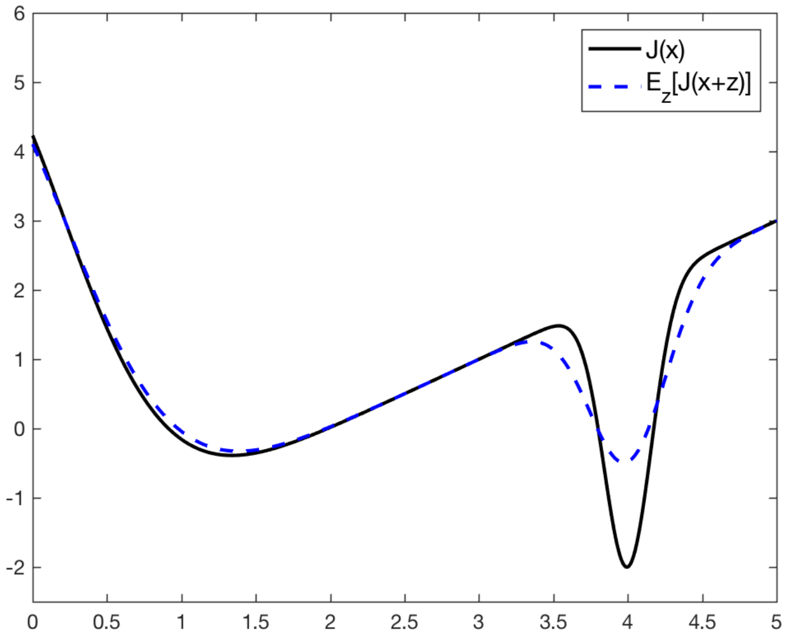
Challenge 3: Optimization Under Uncertainty



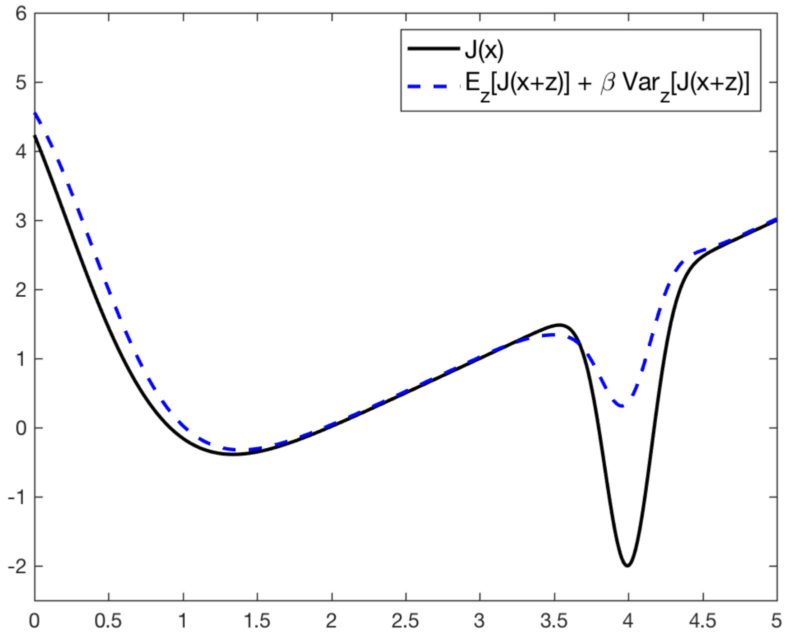


- Want performance not to depend on
  - Tiny changes to coil geometry (within engineering tolerance)
  - Changes to control parameters during operation
  - Uncertainty in approximations to physics or model parameters

Risk-Neutral and Risk-Averse Optimization



Risk-Neutral Objective

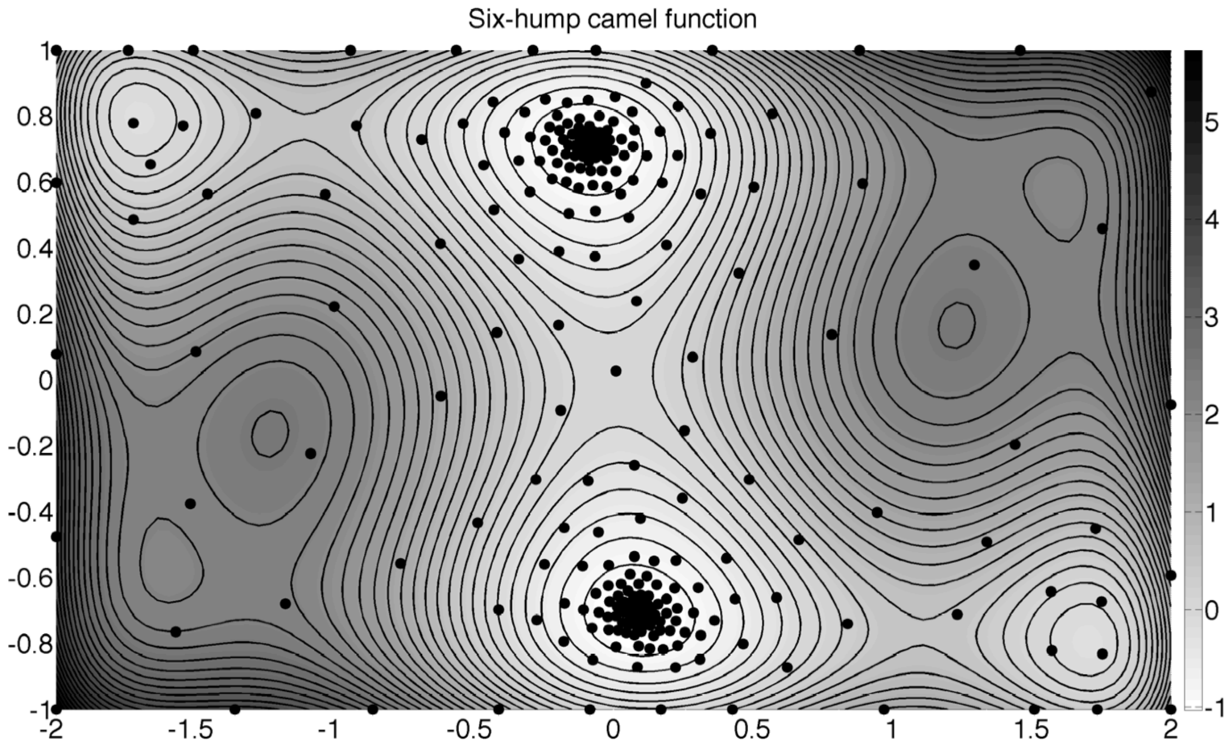


Risk-Averse Objective

Challenge 4: Global Optimization

- Global optimization is hard!
  - Especially in high-dimensional spaces
  - Effective solvers are tailored to structure (e.g. convexity)
  - More general methods are mainly heuristic
- Want algorithms that balance
  - **Exploration**: Evaluating novel designs with unknown properties
  - **Exploitation**: Refining known designs from previously explored regions
- Global model-based techniques help (with the right models!)

Exploration vs Exploitation



General Formulation

$$\min_{\text{coils}} \mathbb{E}_z[J_{\text{int}}(B,z)], \mathbb{E}_z\left[J_{\text{qs}}(B,z)\right], \mathbb{E}_z[J(B,q,z)], \dots, \mathcal{R}(B,q,z,\dots)$$

Subject to: manufacturing and physics constraints,

PDEs relating coils to field B,

particle or heat transport q, etc.

- Optimize integrability (J_{int}), quasi-symmetry (J_{qs}), etc
- Take into account uncertain parameters z
- Include a risk aversion objective $\mathcal{R}$
- Find Pareto points vs using weighted sums of objectives

General Formulation

$$\min_{\text{coils}} \mathbb{E}_z[J_{\text{int}}(B,z)], \mathbb{E}_z\left[J_{\text{qs}}(B,z)\right], \mathbb{E}_z[J(B,q,z)], \dots, \mathcal{R}(B,q,z,\dots)$$

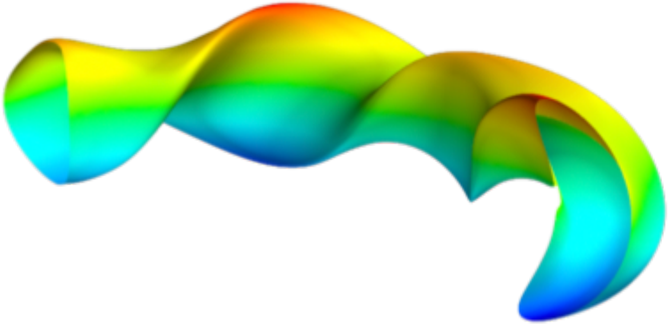
Subject to: manufacturing and physics constraints,

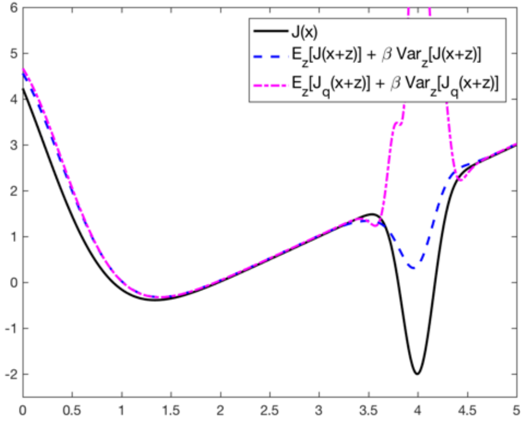
PDEs relating coils to field B,

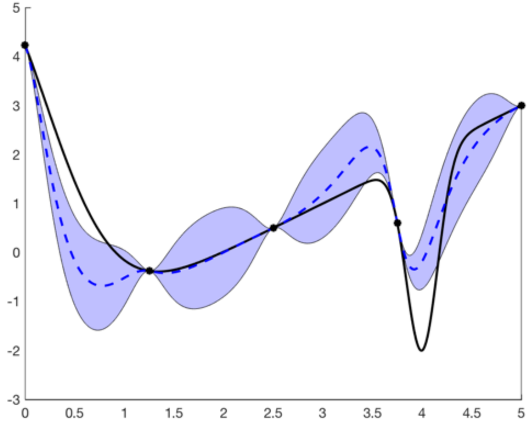
particle or heat transport q, etc.

- Costs beyond deterministic PDE solves:
  - Stochastic objectives require many deterministic solves each
  - Pareto frontier is an $(m-1)$-dimensional manifold with corners
  - Non-convex global optimization requires a lot of searching
- Common issue: the **curse of dimensionality**

Addressing the Challenges

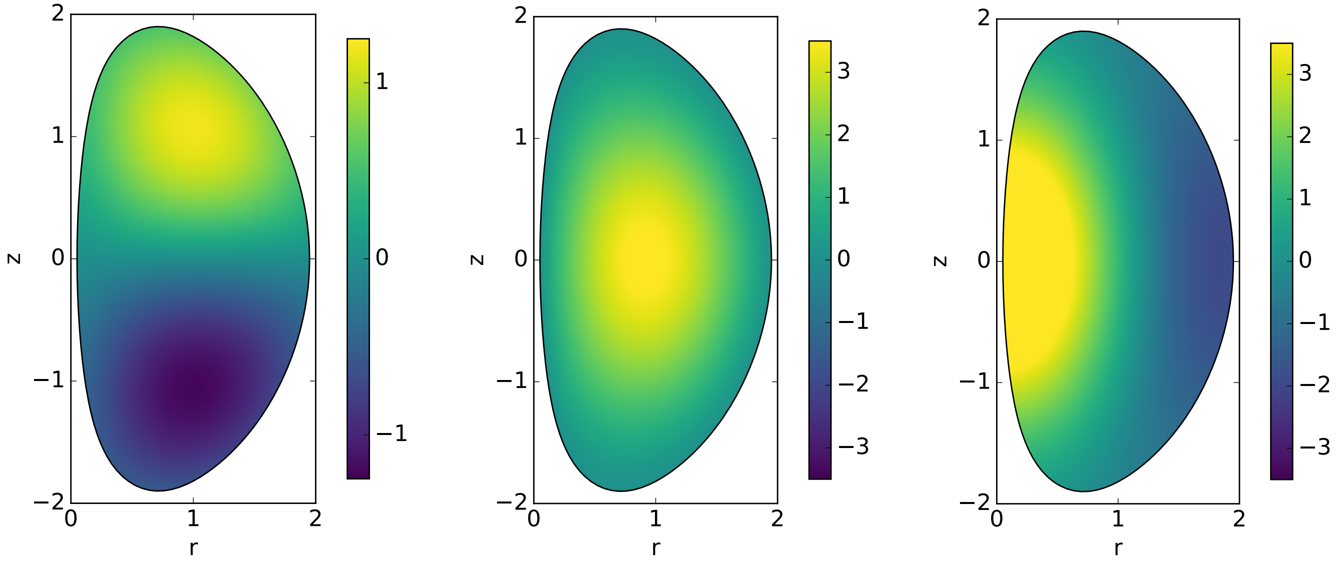






- Fast physics solver formulations
- Efficient optimization under uncertainty
- Surrogates and multi-fidelity methods

Fast Equilibrium Solvers



(a) The radial component, B_r. (b) The azimuthal component, B_ϕ. (c) The z component, B_z.

Figure 4: The Beltrami field $\boldsymbol{B}$ in the $\phi = 0$ plane.

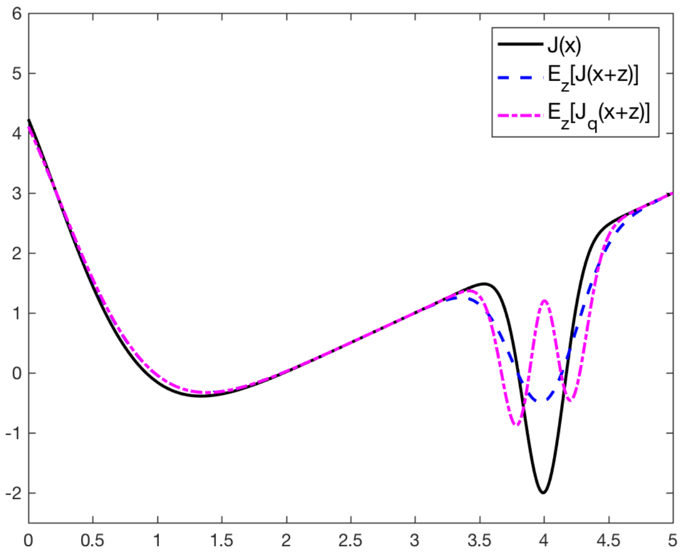
Integral equation solver for Taylor states in toroidal geometries [O'Neill, Cerfon, 2018]

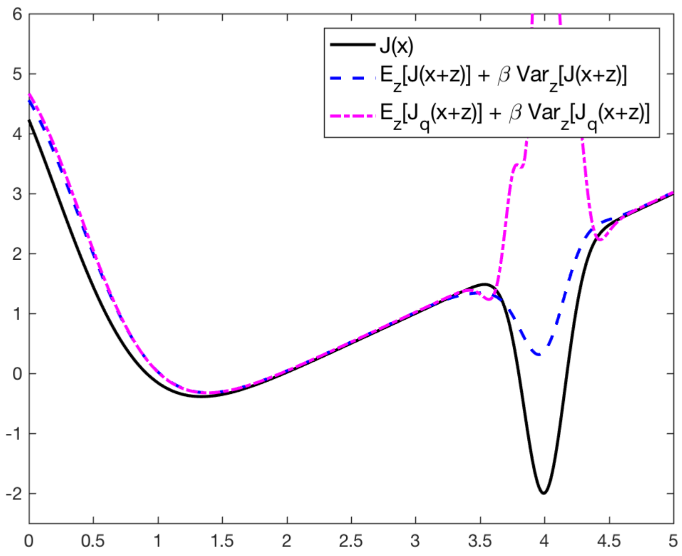
Laplace-Beltrami solver on genus 1 surfaces [Imbert-Gérard, Greengard, 2017]

Fast Equilibrium Solvers

- Integral equation formulation for coil fields + MHD equilibria
  - Respects underlying physical conditioning of problem
  - Generally only need boundary discretization (vs volume meshing)
  - Fast high-order algorithms exist for required integral operators
- Associated adjoint solvers to compute sensitivities
- Fast re-solves in optimizer under low-rank geometry updates

Efficient Optimization Under Uncertainty





$$J_q(x+Z) = J(x) + J'(x)Z + \frac{Z^T H_J(x) Z}{2}, \qquad Z \sim N(0, C)$$

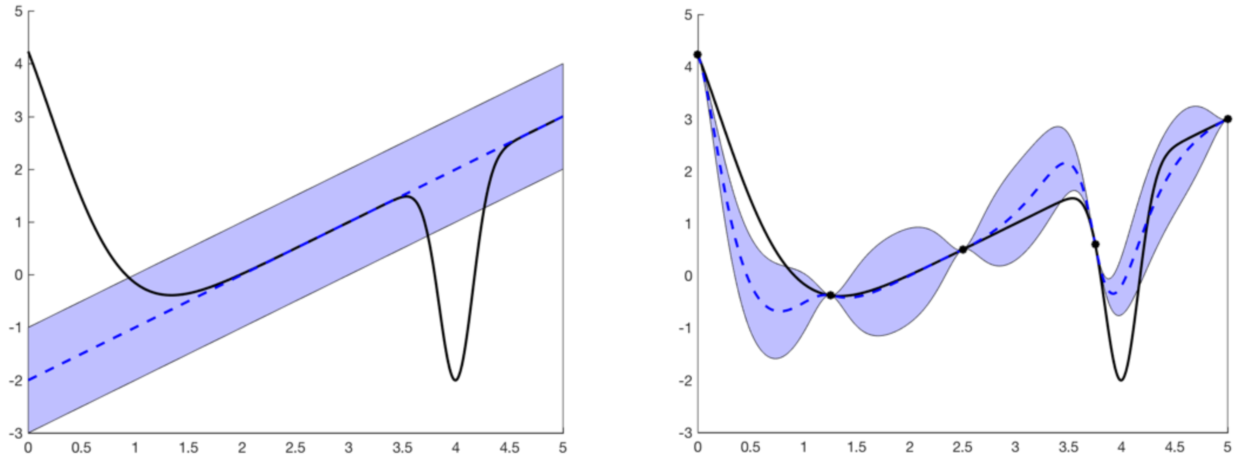
Use a quadratic approximation to compute stochastic, possibly risk-averse, objective. [c.f. Alexanderian, Petra, Ghattas, Stadler, 2017].

Efficient Optimization Under Uncertainty

- Consider objective $J(x, z)$ where x is control and z uncertain
- Model z as multivariate Gaussian
- Use local quadratic approximation in stochastic variables
  - Require $\partial J / \partial z$ and action of Hessian $\partial^2 J / \partial z^2$ on vectors
  - Assume Hessian is (approximately) low rank – dimension reduction
  - Scaling with low intrinsic dimension vs. number of parameters
- Beyond Gaussian: use approximation as a control variate

Lots of remaining challenges (high nonlinearity, turbulence, etc)

Surrogate Methods



- *Surrogates* (*response surfaces*) approximate costly functions
- May also estimate uncertainty (e.g. Gaussian process models)
- Different variants: fixed, parametric, non-parametric
- Incorporate function values, gradients, bounds, …

Surrogate Optimization

- Example: Single objective Bayesian optimization
  - Sample objective and fit a GP model
  - Use acquisition function to guide further sampling (EI, PI, UCB, KG); goal is to balance exploration vs exploitation
- Active work on recent variants for
  - Pareto (ParEGO [Knowles 2004], GPareto [Binois, Picheny, 2018])
  - Multi-fidelity optimization [e.g. March, Willcox, Wang, 2011]
  - Incorporating gradients [Wu, Poloczek, Wilson, Frazier, 2018]
  - Objectives with quadrature [Toscano-Palmerin, Frazier, 2018]
- Several options in PySOT toolkit [B, Eriksson, Shoemaker]

Surrogates with Side Information

- Problem: Need predictions from *limited data*
- Shape surrogate to have known structure (*inductive bias*)
  - Meaningful mean fields
  - Structured kernels (symmetry, regularity, dimension reduction, etc)
  - Tails that capture known singularities and other features
- Alternative: Jointly predict $J_{\mathrm{costly}}(x)$ and $J_{\mathrm{corr}}(x)$
  - Kernel captures correlation between functions as well as across space
  - Basic idea is old: e.g. *co-kriging* in geostatistics
  - Use in computational science and engineering is active research [Peherstorfer, Willcox, Gunzburger, others – also my sabbatical!]

The Bigger Picture

- Many of the challenges of stellarators are universal in computational science and engineering!
  - Physics is often governed by expensive-to-solve PDEs
  - Physics-agnostic optimization infeasibly hard, even with big computers
  - Need structure to reduce problem dimension / model complexity
- Stellarator problem involves many common components
  - Mechanisms described by PDEs for transport, diffusion, reaction
  - Methods we develop will impact other areas
- But success depends on using specific problem structure!

Summary

- Challenge: Multi-objective risk-averse stellarator optimization
- Approach: Fast equilibrium solvers, scalable optimization under uncertainty, multi-fidelity surrogate methods
- Specific goals:
  - Test problem formulation (vacuum and positive pressure)
  - Extension to multi-objective programming formulation
  - Scalable risk-averse stochastic programming methods
  - Optimization via physics-sensitive multi-fidelity surrogates
- Methods to be incorporated into new SIMSOPT code