Global, Robust, Multi-Objective Optimization of Stellarators

David Bindel
29 March 2019

Department of Computer Science
Cornell University
What Makes a Good Stellarator?

Half-Module
($\frac{1}{10}$ of W7-X)

Outside

Inside

Winding Surface

Limiting Surfaces

Clearance

Coil

Surface

Plasma Boundary

Field Error

Axis Position (Triangle)

Magnetic Well & Iota

Ripple

Coil Curvature

Magnetic Axis

36°

Objects used in Optimization

Quality Criteria

Field Error + Geometric Properties

Properties of the Vacuum Field

Optimization of Fourier Coefficients

Figure courtesy Jim Lobsien
How Do We Optimize? (STELLOPT Approach)

**Optimizer**

- Calculate $\chi^2$ (physics + engineering targets)
- Adjust plasma boundary (or coil shape)
- Solve 3D equilibrium

Mathematical equation:

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

Goal: Design MHD equilibrium (coil opt often separate)

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

Minimize $\chi^2$ objective over $C$:

$$\chi^S(x) = \sum_{k=1}^{m} \frac{J_k(x)}{\sigma_k^2}, \quad J_k(x) = (F_k(x) - F^*_k)^2$$

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, coil design, ...
   - Several times per step for finite-difference gradients

2. Managing tradeoffs
   - How do we choose the weights in the $\chi^2$ measure? By gut?
   - Varying the weights does not expose tradeoffs sensibly

3. Dealing with uncertainties
   - What you simulate $\neq$ what you build!

4. Global search
   - How to avoid getting stuck in local minima?
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

What goes into the optimization objective and constraints?

- Costly physics solves (MHD equilibrium, transport, ...)
  - PH: “The equations are all first order, and should not be taken too literally.”
  - One approach: work with simpler/cheaper proxies
  - Does this actually get us what we want?
- Derivatives require PDE sensitivity / adjoints
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
- PDE structure influences objective landscape
- PDE properties: compactness, smoothing, near/far fields
- Opportunities for dimension reduction in optimization
Fast Equilibrium Solvers

Figure 1: An example of a Taylor state computed in a toroidal-shell domain. Using our boundary integral method, we only need to discretize the domain boundary. This reduces the dimensionality of the unknowns needed, and leads to significant saving simp (up) computation work. Once the boundary integral equations is solved, the magnetic field $B$ can be evaluated at off-surface points very efficiently. On the right, we show the magnitude of $B$ in different cross-sections of the domain as well as Poincaré plots of the field in each cross-section, generated by tracing the field lines using a 10th-order spectral deferred correction (SDC) scheme. Additionally, we touch on subtle mathematical questions [7, 21, 25, 30]. Until recently, computational efforts to solve this problem could be divided into two categories [23]. The first category of numerical solvers relies on the assumption of the existence of nested magnetic flux surfaces throughout the computational domain [3, 26–28, 52]. These solvers have played an important role in the design of new non-axisymmetric magnetic confinement devices as well as the analysis of experimental results obtained from existing ones. However, they are fundamentally limited in terms of both robustness and accuracy for the computation of equilibria with both a smooth plasma pressure profile and smooth magnetic field line pitch. In this regime, this class of solvers (and the model upon which they are based) is unable to accurately approximate the singular structures that must naturally occur in such situations [25, 38, 39, 47]. On the other hand, an alternative, second class of solvers, does not constrain the space of solutions to equilibria with nested flux surfaces, and computes equilibria which may have magnetic islands and chaotic magnetic field lines [23, 46, 49]. These solvers also play an important role in the magnetic fusion program since they can be used to study, among other significant questions, the disappearance of magnetic islands, often called island healing, corresponding to an increase of the plasma pressure [24, 36] or to a change of the coil configuration [31]. They are also often able to compute the details of the magnetic field configuration in the vicinity of the plasma edge [12]. Despite these additional capabilities, equilibrium codes in the first category are often favored because existing solvers in the second category converge substantially slower [12] and are much more computationally intensive [32].

Recently, a third approach has been developed which combines aspects of both categories of solvers described above. In this approach, the entire computational domain is subdivided into separate regions, each with constant pressure, assumed to have undergone Taylor relaxation [51] to a minimum energy state subject to conserved fluxes and magnetic helicity. Each of these regions is assumed to be separated by ideal MHD barriers [33, 34]. This model has a significant limitation: for general pressure profiles, solutions of this model...
Adjoint-Based Vacuum-Field Optimization

Single-stage optimization of coil shapes and vacuum-field properties:

- Targets: rotational transform, ripple, coil length, magnetic axis length
- Constraints: Magnetic axis is generated by coils

With adjoint solves, not a problem to have many geometric parameters:

<table>
<thead>
<tr>
<th>$N_p$</th>
<th>102</th>
<th>192</th>
<th>282</th>
<th>372</th>
<th>462</th>
<th>552</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finite differences</td>
<td>84</td>
<td>222</td>
<td>411</td>
<td>664</td>
<td>1057</td>
<td>1473</td>
</tr>
<tr>
<td>Adjoint approach</td>
<td>4</td>
<td>11</td>
<td>26</td>
<td>48</td>
<td>83</td>
<td>116</td>
</tr>
</tbody>
</table>

Timings on a modern laptop.

[Giuliani, Cerfon, Landreman, Stadler]
Example: Optimization of Ripple in NCSX Coils

(a) Convergence curve.

(b) Coils before and after optimization.

[Giuliani, Cerfon, Landreman, Stadler]
Challenge 2: Multi-Objective Optimization

Half-Module (\(\frac{1}{10}\) of W7-X)

Outside

Winding Surface

Limiting Surfaces

Clearance

Coil

Inside

Plasma Boundary

Field Error

Axis Position (Triangle)

Magnetic Well & Iota

Ripple

Coil Curvature

Magnetic Axis

36°

Objects used in Optimization

Field Error + Geometric Properties

Properties of the Vacuum Field

Optimization of Fourier Coefficients

Figure courtesy Jim Lobsien
What makes an “optimal” stellarator?

- Approximates field symmetries (which measures?)
- Satisfies macroscopic and local stability
- 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.
A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems

I. Das and J.E. Dennis
Department of Computational and Applied Mathematics, Rice University of Houston, TX 77251-1892, USA

June 4, 2015

Some optimal solutions to a smooth multi-objective problem cannot be found by minimizing a total $\chi^2$
Exploring the Pareto Frontier

$x$ dominates $y$ if

\[ \forall k, J_k(x) \leq J_k(y) \]

and not all strict.

Best points called **Pareto optimal**
(non-dominated, non-inferior, efficient)

**Pareto frontier** is an \((m - 1)\)-manifold with corners, generally.

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

Other methods sample / approximate the full frontier.
Challenge 3: Optimization Under Uncertainty

Low construction tolerances:
- NCSX: 0.08%
- Wendelstein 7-X: 0.1% – 0.17%

Want: higher tolerances as coil optimization goal!

Also want tolerance to
- Changes to control parameters in operation
- Uncertainty in physics or model parameters
For the Pessimist: Robust Optimization

Robust optimization idea:
- Characterize uncertainty region
- Optimize for worst-case

Drawbacks:
- May be pessimistic
- Need an inner optimization
- Non-smooth outer objective
Optimization Under Uncertainty

Risk-Neutral Objective

- Requires distributional assumption for uncertainty
- Inner computation of moments (MC or quadrature)
- Outer objective becomes smoother

Risk-Averse Objective
Monte Carlo Approach

Robustness & average performance significantly improved

[Lobsien, Drevlak, Sunn Pedersen]
Example

Classic Coil Optimization
(Deviation 0 mm, Penalty = 4.19)

Stochastic Optimization
(Deviation 2.5 mm, Penalty = 2.24)
Efficient Optimization Under Uncertainty

Risk-Neutral Objective

\[ J_q(x + z) = J(x) + J'(x)z + \frac{1}{2}z^T H_j(x)z, \quad z \sim N(0, C) \]

Risk-Averse Objective

Use quadratic approximation to compute robust or uncertain objectives [c.f. Alexanderian, Petra, Ghattas, Stadler, 2017]
• 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)
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 often 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!)
Surrogates (aka response surfaces) approximate costly functions

- Different variants
  - Fixed physics-based approximations
  - Parametric: polynomial, ridge, NN
  - Non-parametric: kernel methods
- Incorporate function values, gradients, bounds, ...
- May also estimate uncertainty (e.g. Gaussian process models)

Bayesian optimization uses GP mean and variance to guide sampling.
Kernel-Based Surrogates

Prior

Function data
Kernel-Based Surrogates

Prior

Hermite data
Exploration vs Exploitation (Eriksson and B)
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)

• 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]
  • Simultaneous dimension reduction [Eriksson, Dong, Lee, B, 2018]
  • Objectives with quadrature [Toscano-Palmerin, Frazier, 2018]

• Several options in PySOT toolkit [Eriksson, Shoemaker, B]
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 $J_{\text{costly}}(x)$ and $J_{\text{corr}}(x)$
  - Kernel captures correlation between functions and across space
  - Basic idea is old: e.g. *co-kriging* in geostatistics
  - Use in computational science and engineering is active research
    [Peherstorfer, Willcox, Gunzburger, many others]
General Formulation

\[
\min_{\text{coils}} \mathbb{E}_z[j_{\text{int}}(B, z), \mathbb{E}_z[j_{qs}(B, z)], \mathbb{E}_z[j(B, q, z)], \ldots, \mathcal{R}(B, q, z, \ldots)}
\]

s.t. Manufacturing and physics constraints

PDEs relating coils to magnetic field \( B \)

Physics of particle or heat transport \( q \)

- Optimize integrability \( j_{\text{int}} \), quasi-symmetry \( j_{qs} \), etc
- Take into account uncertain parameters \( z \)
- Include risk aversion objective \( \mathcal{R} \)
- Find Pareto points vs using scalarized objective
General Formulation

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

s.t. Manufacturing and physics constraints

- PDEs relating coils to magnetic field \( B \)
- Physics of particle or heat transport \( q \)

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: **curse of dimensionality** — dimension reduction a common theme
I was tense, I was nervous, I guess it just wasn’t my night.
Art Fleming gave the answers; oh, but I couldn’t get the questions right.
— Weird Al Yankovic, “I Lost on Jeopardy”

Stellarator optimization is hard. Challenges include:

1. Costly and “black box” physics computations
2. Managing tradeoffs
3. Dealing with uncertainties
4. Global search

Many challenges/opportunities in the formulation – not unique to stellarators!