Optimizing Magnetic Confinement Devices for Fusion Plasmas

David Bindel

3 Apr 2024

Department of Computer Science
Cornell University

Plan for the hour

- Stellarators, fusion, and the Simons collaboration
- Stellarator optimization under uncertainty
- Multi-objective optimization

Who?

Simons Collaboration: "Hidden Symmetries and Fusion Energy"

`https://hiddensymmetries.princeton.edu/`

A collaboration of plasma physicists and mathematicians from:

Princeton, NYU, Maryland, IPP Greifswald, Warwick, CU Boulder, Cornell, UW Madison, EPFL, ANU, UT Austin, U Arizona.

(along with many unfunded collaborators)

- Phase 0: Aug 2017-Aug 2018
- Phase 1: Sep 2018-Aug 2022
- Phase 2: Sep 2022-Aug 2025

Some Phase 0 recollections

2017-08-22	Email from Antoine Cerfon, “would you be interested in participating in these initial conversations?”
2017-09-01	Initial conversation
2017-10-04	LOI submitted
2017-12-06	First two-day proposal meeting
2018-01-31	Second two-day proposal meeting
2018-02-15	Proposal submitted
2018-04-18	Panel pitch (Bhattacharjee, MacKay, Bindel)
2018-05-30	Award announced to collaboration (recommended change in title to add Fusion Energy).

“Fusion for a 5 Year Old”

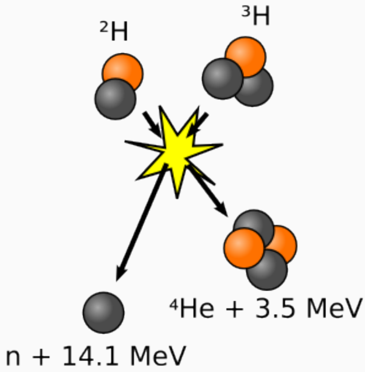
“Fusion for a 5 Year Old”

At the risk of sounding like a broken record, I will lobby for the addition of a paragraph in the introduction of the proposal that describes magnetically confined fusion as if it were being explained to a five year old.
– Mike O’Neill (2018-02-07)

“Adiabatic invariants of Hamiltonian mechanics” is well beyond the level of sophistication that should be included in the intro, in my opinion.
– Response to a proposed revision (2018-02-08)

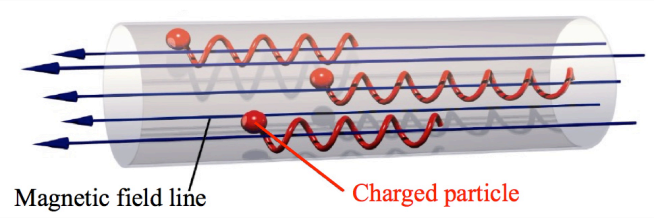
Ad: *Introduction to Stellarators* by Imbert-Gerard, Paul, Wright (`https://arxiv.org/abs/1908.05360`, coming to SIAM)

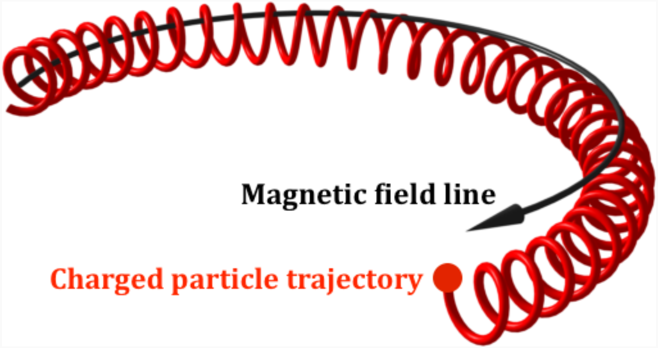
“Fusion for a 5 Year Old”



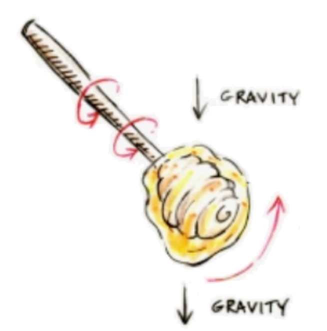
Lawson: Need combination of high density, temperature, energy confinement time

Magnetic confinement basics



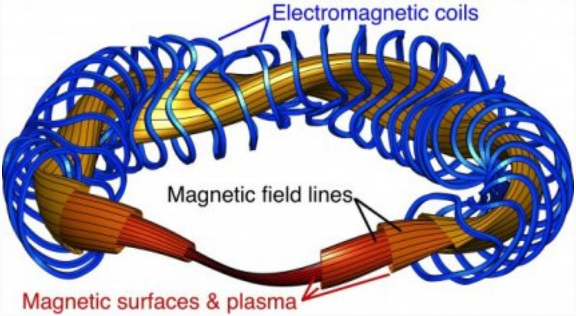


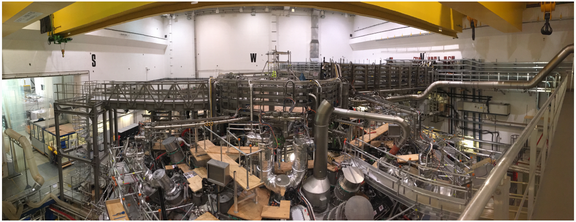
Magnetic confinement basics



- Ensure drift in and out averages to zero.
- Tokamaks: axisymmetric field (requires plasma current)
- Stellarators: use a “hidden symmetry”

Stellarator Concept and Practice





Stellarator Quality Measures

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.

How Do We Optimize? (STELLOPT Approach)

Goal: Design MHD equilibrium (coil opt often separate)

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

Minimize χ^2 objective over C:

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

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

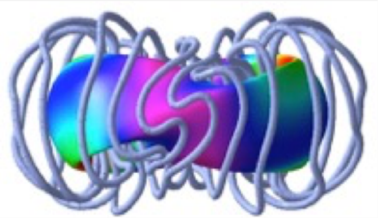
How Do We Optimize? (STELLOPT Approach)

Optimizer

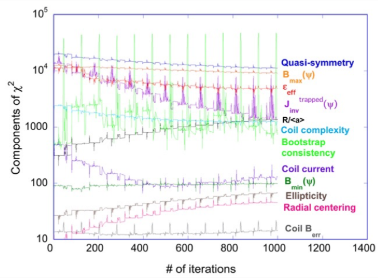
Calculate χ^2 (physics + engineering targets)

Adjust plasma boundary (or coil shape)

Solve 3D equilibrium



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



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 χ^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?

Progress on Costly Computations: Vacuum Field

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

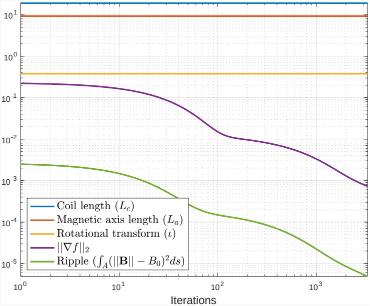
With adjoint solves, not a problem to have many geometric parameters:

N_p	102	192	282	372	462	552
Finite differences	84	222	411	664	1057	1473
Adjoint approach	4	11	26	48	83	116

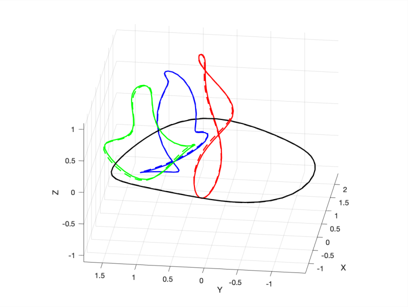
Timings on a modern laptop.

[Giuliani, Cerfon, Landreman, Stadler]

Example: Optimization of Ripple in NCSX Coils



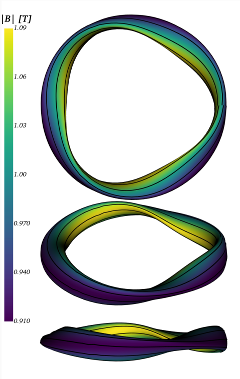
(a) Convergence curve.

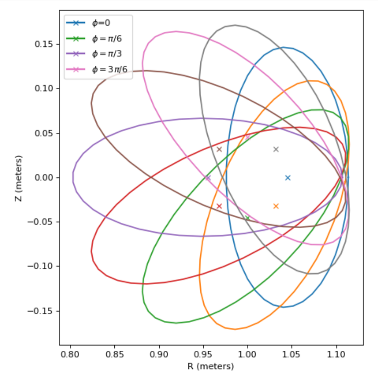


(b) Coils before and after optimization.

[Giuliani, Cerfon, Landreman, Stadler]

Progress on Costly Computations: Near Axis



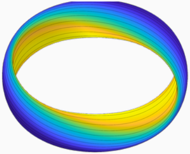


Long theoretical history: Mercier (1964), Lortz & Nührenberg (1976), Garren & Boozer (1991)

Recent: Landreman, Sengupta, Rodriguez, Plunck, Jorge, …

Fast, gives good initial guesses

Progress on Costly Computations: MHD Solvers + Optimizers



- New equilibrium solvers (DESC, SPEC, BIEST)
- Faster, better accuracy, provide derivatives
- Coupled to optimizer frameworks (SIMSOPT, DESC)

Progress of the Simons collaboration

- Collaboration has made a lot of progress (though work remains) on
    - Fast MHD equilibrium codes (Maryland, Princeton, NYU, Arizona, Flatiron)
    - With derivatives (NYU, Maryland, Princeton)
    - Optimized under uncertainty (Greifswald, Cornell)
    - Plasmas with high quasisymmetry (Maryland, Princeton)
- More limited progress on
    - Global search (though near axis helps)
    - Fast and accurate proxies for turbulent transport
    - Optimizing with instabilities (micro/macro)
    - Optimization of divertors

Cornell work

Rest of the session on two Cornell-centered projects:

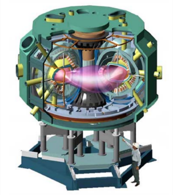
- Optimization under uncertainty
- Multi-objective optimization (if time)

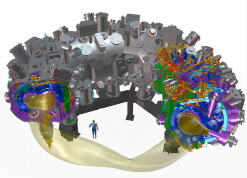
Optimization Under Uncertainty

Low construction tolerances:

- NCSX: 0.08%
- Wendelstein 7-X: 0.1% – 0.17%

Higher tolerances as coil opt goal!

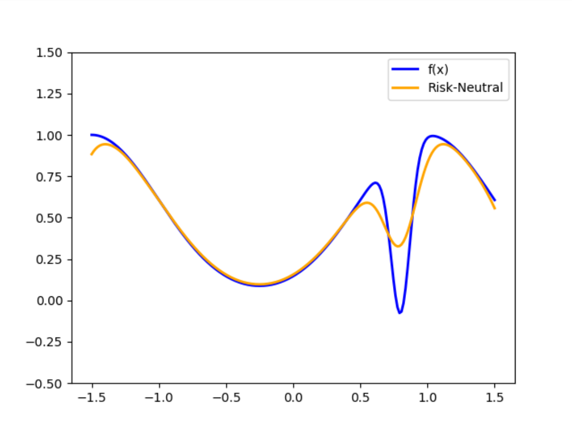




Also want tolerance to

- Changes to control parameters
- Uncertainty in physics or model

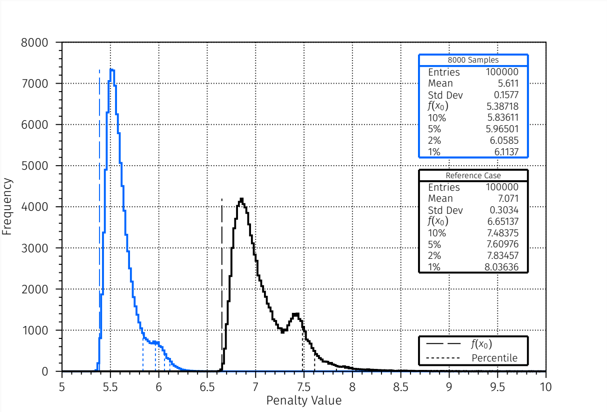
Risk-neutral OUU



Want efficient OUU in ~ 200 dimensions

$$\min_{x \in \Omega} \mathbb{E}_U[f(x - U)]$$

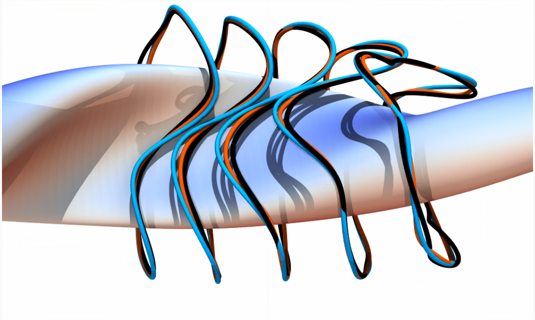
(Recent) Prior: Monte Carlo Approach



Robustness & mean perf greatly improved (w/ $\sim 10^8$ evals)

J.-F. Lobsien, M. Drevlak, T. Kruger, S. Lazerson, C. Zhu, T. S. Pedersen, *Improved performance of stellarator coil design optimization*, Journal of Plasma Physics, 2020.

Our Approach: fast TuRBO-ADAM



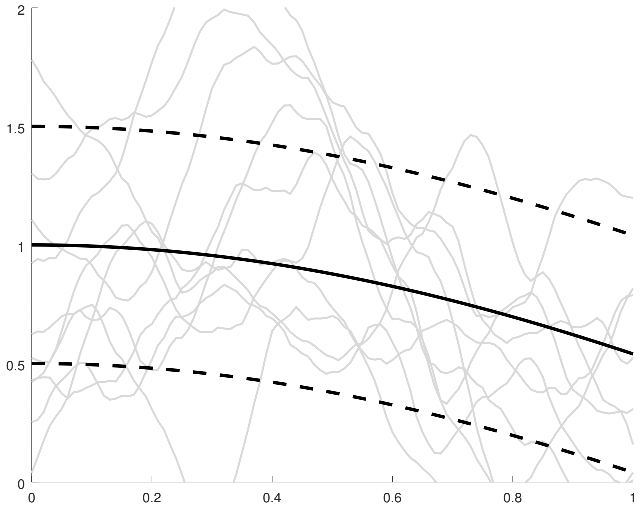
Black: ref; red: TuRBO-ADAM 10mm; blue: TuRBO-ADAM 20mm.

Evaluate objective with FOCUS from PPPL.

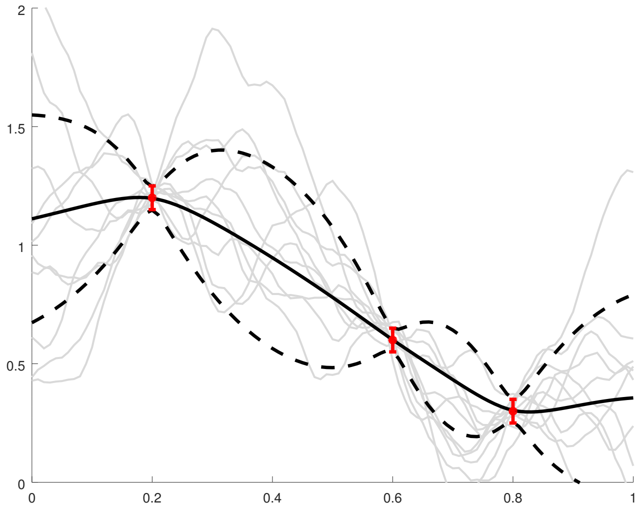
- Global search with modified TuRBO
- Local refinement with ADAM with control variate

Costs about 0.01% the evaluation budget.

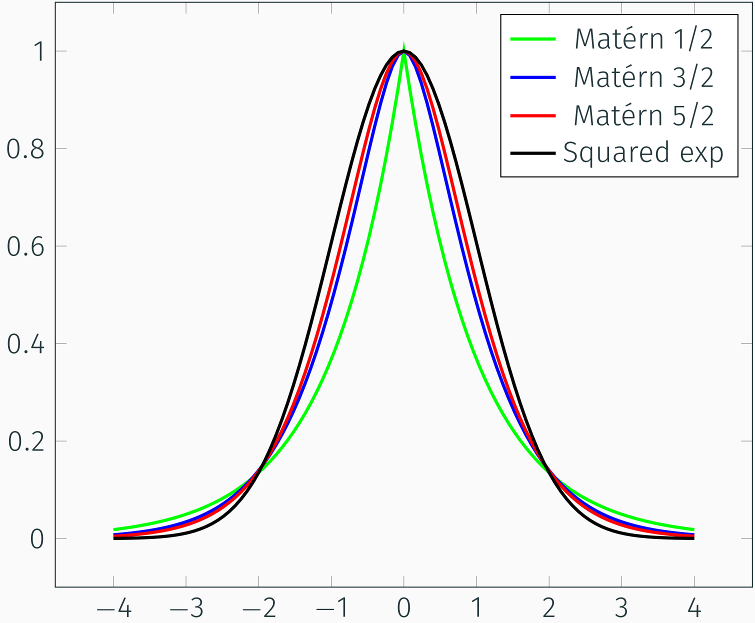
Gaussian Processes (GPs)



Being Bayesian



Matérn and SE kernels



Gaussian Processes (GPs)

Our favorite continuous distributions over

$\mathbb{R}$:	Normal(μ, σ^2),	$\mu, \sigma^2 \in \mathbb{R}$
$\mathbb{R}^n$:	Normal(μ, C),	$\mu \in \mathbb{R}^n, C \in \mathbb{R}^{n \times n}$
$\mathbb{R}^d \to \mathbb{R}$:	GP(μ, k),	$\mu : \mathbb{R}^d \to \mathbb{R}$, $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$

More technically, define GPs by looking at finite sets of points:

$$\forall X = (x_1, \ldots, x_n), x_i \in \mathbb{R}^d,$$

$$\text{have } f_X \sim N(\mu_X, K_{XX}), \text{ where}$$

$$f_X \in \mathbb{R}^n, \quad (f_X)_i \equiv f(x_i)$$

$$\mu_X \in \mathbb{R}^n, \quad (\mu_X)_i \equiv \mu(x_i)$$

$$K_{XX} \in \mathbb{R}^{n \times n}, \quad (K_{XX})_{ij} \equiv k(x_i, x_j)$$

When X is unambiguous, we will sometimes just write K.

Being Bayesian

Now consider prior of $f \sim \mathrm{GP}(\mu, k)$, noisy measurements

$$f_X \sim y + \epsilon, \quad \epsilon \sim N(0, W), \qquad \text{typically } W = \sigma^2 I$$

Posterior is $f \sim \mathrm{GP}(\mu', k')$ with

$$\begin{aligned} \mu'(x) &= \mu(x) + K_{xX} c & \tilde{K} &= K_{XX} + W \\ k'(x, x') &= K_{xx'} - K_{xX} \tilde{K}^{-1} K_{Xx'} & c &= \tilde{K}^{-1}(y - \mu_X) \end{aligned}$$

The expensive bit: solves with $\tilde{K}$.

Bayesian Optimization (BO)

Typical GP-based BO:

- Evaluate f on initial sample in Ω
- Condition a GP on sample data
- Until budget exhausted
    - Optimize *acquistion function* $\alpha(x)$ over Ω
      (e.g. $\alpha_{\mathrm{EI}}(x) = E\left[[f(x_{\mathrm{best}}) - f(x)]_+\right]$ where x_{best} is best so far)
    - Evaluate at selected point
    - Update the GP model (including hyper-parameters)
- Standard cost: $O(n^3)$ per step (with n data points)

Multi-Start

Suppose d large, but not too many minimizers:

- Choose M starting points scattered over Ω
- Run local minimizer (gradient descent, Newton, etc)
- Hope for at least one start per convergence basin

Q: How to allocate effort to different starts?

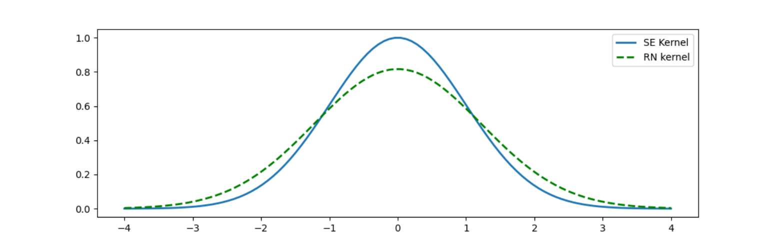
TuRBO: Trust-Region BO

For high-d: combine local BO with multi-start strategy

- Rough global sampling at M points
- Local GP models and trust-region around each point
- Thompson sampling to choose which local model (and trust region) to refine next

(Eriksson, Pearce, Gardner, Turner, Poloczek, 2019)

TuRBO + OUU



- TuRBO builds GP models for $f(x)$ (nominal objective)
- Simple transform from GP for $f(x)$ to GP for $E_U[f(x+U)]$ (Beland and Nair, 2017)

Problem: TuRBO explores a lot — want more refinement

Stochastic Gradient Descent (SGD)

Ordinary gradient descent is

$$x_{k+1} = x_k - \alpha_k \nabla \phi(x_k)$$

SGD is

$$x_{k+1} = x_k - \alpha_k g_k$$

where g_k is a random draw, $E[g_k] = \nabla \phi(x_k)$.

For $\phi(x) = E_U[f(x + U)]$, use $g_k = \nabla f(x_k + u_k)$.

Convergence is slow ($O(1/m)$), but steps can be cheap.
Speed depends a lot on variance of gradient estimator.

Adam + Control Variates

- Regular Adam: stochastic gradient algorithm with "adaptive momentum" for step size control. Use directions

$$g(x) = \nabla f(x + U)$$

  for a random draw U (can also do mini-batch).
- Variance reduction with control variates (Wang, Chen, Smola, Xing, 2013)

$$\begin{aligned} g(x) &= \nabla f(x + U) + \alpha(\hat{g}(x) - E[\hat{g}(x)]) \\ \hat{g}(x) &= \nabla f(x) + HU. \end{aligned}$$

- True Hessian not avail, so set H to be an approximate Hessian (BFGS approximation via gradients from Adam).

Additional Information

Multi-output GPs model $f : \Omega \subset \mathbb{R}^d \to \mathbb{R}^k$

- Model covariance over space and across outputs.
- Example: function values + derivatives

$$\mu^\nabla(\mathsf{x}) = \begin{bmatrix} \mu(x) \\ \nabla_x \mu(x) \end{bmatrix}, \quad k^\nabla(x, x') = \begin{bmatrix} k(x, x') & (\nabla_{x'} k(x, x'))^T \\ \nabla_x k(x, x') & \nabla^2 k(x, x') \end{bmatrix}$$

- Can also model multi-fidelity sims, etc

Pro: FOCUS provides gradients, easy to incorporate!
Con: Matrix dimensions scale like $n(d+1)$
(Partial) Fix: Variational inference (Bindel, Gardner, Huang, Padidar, Zhu, NeurIPS 2021)

Cornell work

Rest of the session on two Cornell-centered projects:

- Optimization under uncertainty
- Multi-objective optimization (if time)

(Glas, Padidar, Kellison, B, JPP 2022; B, Landreman, Padidar, JPP 2023)

Constrained and Multi-Objective

Naive: put everything we care about in a nonlinear LS problem

- $f_k(x)$ is deviation from kth target
- Add some weighting (chosen by the user)

But is this actually what we want?

- Choice of target values is unclear
- Choice of weights is unclear

And there are reasons for numerical nervousness:

- Maybe too few objectives (underdetermined LS problems)
- Maybe poorly conditioned (esp. with “large” weights)
- May not have small residual

Tackling Constraints

General problem

$$\text{minimize } \phi(x) \text{ s.t. } \begin{cases} c_j(x) = 0, & j \in \mathcal{E} \\ c_j(x) \leq 0, & j \in \mathcal{I} \end{cases}$$

Convert into unconstrained optimization / nonlinear equation solving problem with:

- Fewer degrees of freedom (constraint elimination)
- Same degrees of freedom (penalties and barriers)
- More degrees of freedom (Lagrange multipliers)

Constraint elimination usually only for linear constraints.

KKT Conditions

$$\text{minimize } \phi(x) \text{ s.t. } \begin{cases} c_j(x) = 0, & j \in \mathcal{E} \\ c_j(x) \leq 0, & j \in \mathcal{I} \end{cases}$$

Define the Lagrangian

$$L(x, \lambda, \mu) = \phi(x) + \sum_{i \in \mathcal{E}} \lambda_i c_i(x) + \sum_{i \in \mathcal{I}} \mu_i c_i(x).$$

KKT conditions are

$$\begin{aligned} \nabla_x L(x^*) &= 0 & \\ c_i(x^*) &= 0, \quad i \in \mathcal{E} & \text{equality constraints} \\ c_i(x^*) &\leq 0, \quad i \in \mathcal{I} & \text{inequality constraints} \\ \mu_i &\geq 0, \quad i \in \mathcal{I} & \text{non-negativity of multipliers} \\ c_i(x^*)\mu_i &= 0, \quad i \in \mathcal{I} & \text{complementary slackness} \end{aligned}$$

Penalties and Barriers

Want to minimize

$$\text{minimize } \phi(x) \text{ s.t. } \begin{cases} c_j(x) = 0, & j \in \mathcal{E} \\ c_j(x) \leq 0, & j \in \mathcal{I} \end{cases}$$

Instead minimize for small γ

$$\psi_\gamma(x) = \phi(x) + \frac{1}{2\gamma} \sum_{i \in \mathcal{E}} c_i(x)^2 - \gamma \sum_{i \in \mathcal{I}} \log(-c_i(x)).$$

Note that at minimizer x^*:

$$\nabla \psi_\gamma(x^*) = \nabla \phi(x^*) + \sum_{i \in \mathcal{E}} \tilde{\lambda}_i \nabla c_i(x^*) + \sum_{i \in \mathcal{I}} \tilde{\mu}_i \nabla c_i(x^*)$$

where Lagrange multiplier estimates come from the c_i:

$$\tilde{\lambda}_i = c_i(x^*)/\gamma, \quad \tilde{\mu}_i = \gamma / c_i(x^*)$$

Standard trick: Penalty to estimate multipliers.

Managing Tradeoffs

What about using nonlinear least squares for tradeoffs?

More generally, consider $f : \mathbb{R}^n \to \mathbb{R}^m$, maybe minimize

$$w^T f(x) = \sum_{k=1}^{m} w_k f_k(x).$$

Incompleteness of χ-square Combination

Structural Optimization 14, 63–69 © Springer-Verlag 1997

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 Matt Landreman

Some optimal solutions to a smooth multi-objective problem cannot be found by minimizing a total χ^2

Exploring the Pareto Frontier

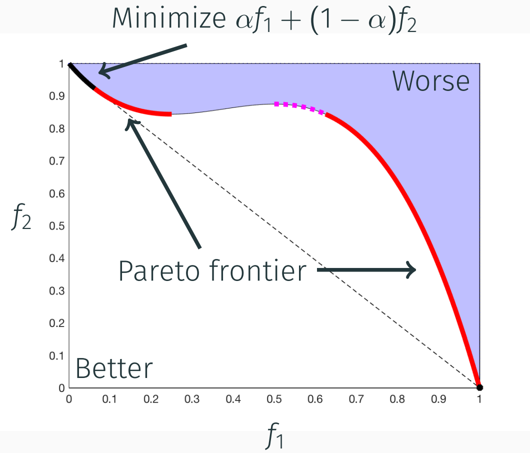
x **dominates** y if

$$\forall k, f_k(x) \leq f_k(y)$$

and not all strict.

Best points are:
Pareto optimal,
aka non-dominated,
aka non-inferior,
aka non-efficient.

Form **Pareto frontier**



Minimizing $\sum_k \alpha_k f_k$ only explores convex hull!
Other methods sample / approximate the full frontier.

First-order condition

Stationary condition:

$$\{J(x)u : u \geq 0\} \cap \mathbb{R}^n_+ = \{0\}.$$

Fritz John stationary condition: for some $\lambda \geq 0, \lambda \neq 0$

$$J(x)^T \lambda = 0.$$

Follows via Motzkin's theorem of the alternative: if A and C are given matrices, can either solve

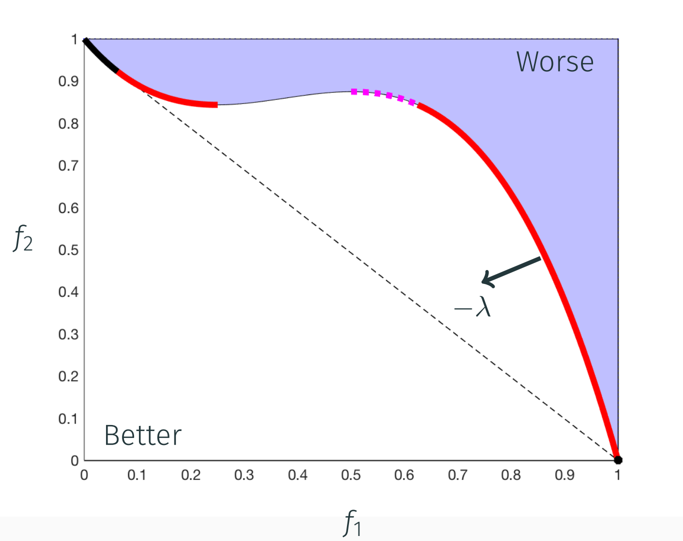
$$Ax < 0, \quad Cx \leq 0$$

or

$$A^T \lambda + C^T \mu = 0, \quad \lambda \geq 0, \lambda \neq 0, \mu \geq 0$$

But not both.

Fritz John multiplier geometry



(χ^2 case: $\lambda_k \propto \sigma_k^{-2}$)

Fritz John vs KKT

Fritz John condition (with constraints): Weak Pareto for

$$\text{minimize } f(x) \text{ s.t. } c(x) \leq 0$$

requires $\lambda \geq 0$ and $\mu \geq 0$ not both all zero such that

$$\begin{aligned} \lambda^T f'(x^*) + \mu^T c'(x^*) &= 0 \\ \mu_i c_i(x^*) &= 0 \end{aligned}$$

Very similar to KKT conditions for constrained opt:

$$\begin{aligned}
\nabla_x L(x^*) &= 0, & & L(x, \lambda, \mu) = \phi(x) + \lambda^T c_{\mathcal{E}}(x) + \mu^T c_{\mathcal{I}}(x) \\
c_i(x^*) &= 0, & i \in \mathcal{E} & \quad \text{equality constraints} \\
c_i(x^*) &\leq 0, & i \in \mathcal{I} & \quad \text{inequality constraints} \\
\mu_i &\geq 0, & i \in \mathcal{I} & \quad \text{non-negativity of multipliers} \\
c_i(x^*)\mu_i &= 0, & i \in \mathcal{I} & \quad \text{complementary slackness}
\end{aligned}$$

Constrained vs multi-objective

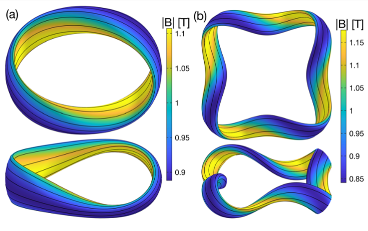
- First-order conditions are *almost* the same
- Can mix and match (constrained multi-objective)
- Multi-objective involves many solves to explore space
- Curse of dimensionality: exploration cost scales exponentially with m

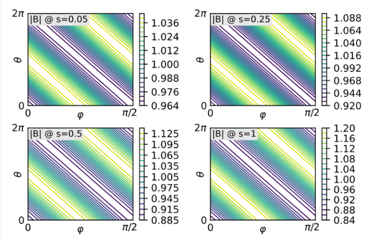
Scalarizing

Find Pareto points via a single-objective optimization problem:

- Linear: $\phi(x) = w^T f(x)$
  - Need to consider stationary points to get full frontier.
  - Uniform weight sampling $\neq$ uniform frontier sampling.
- Projection: $\phi(x) = \sum_i w_i (f_i(x) - f_i^*)^2$
  - Effectively what is done now.
  - Similar tradeoffs to linear scalarization.
- Chebyshev: $\phi(x) = \max_i w_i f_i(x)$
  - Nonsmooth where max is non-unique.
  - Uniform weight $\neq$ uniform frontier sampling.
- ϵ-constraint: $\phi(x) = f_i(x)$, $f_j(x) \leq \epsilon_j$ for $j \neq i$
  - Subproblem is constrained.
  - Can get uniform sampling in components other than i

Example: Quasi-symmetry

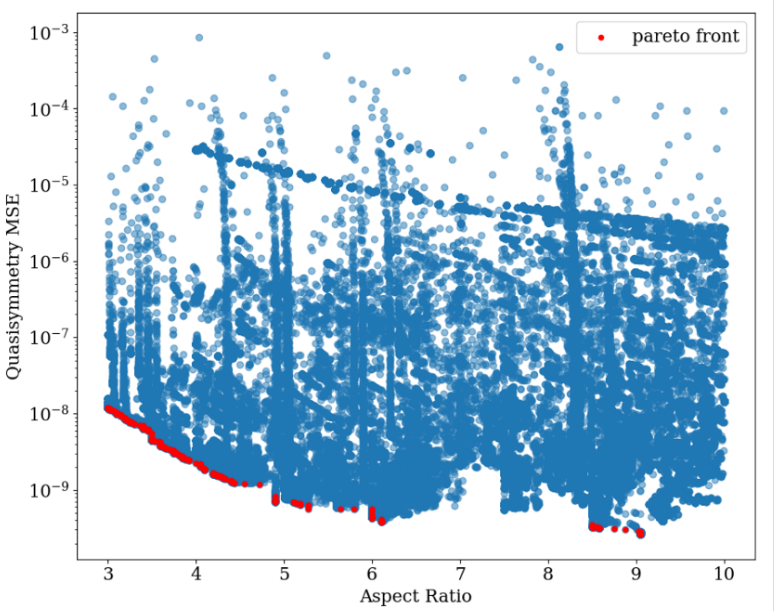




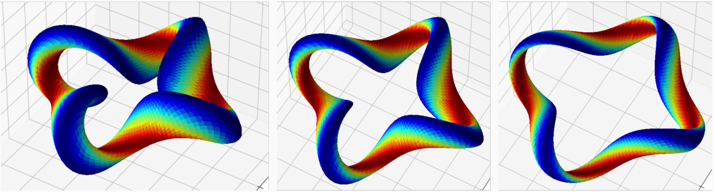
Landreman-Paul QA and QH configurations,
optimized with target aspect ratio 6 and 8.

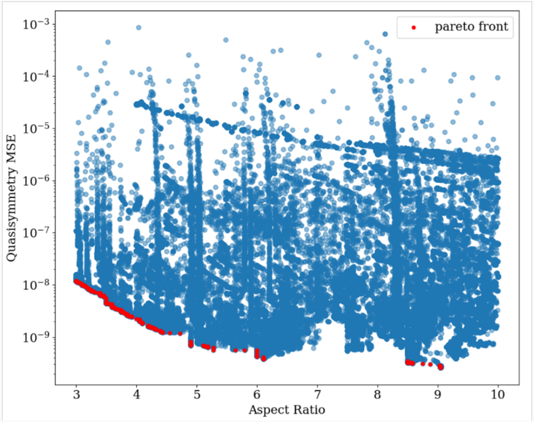
Q: tradeoff between quasisymmetry and aspect ratio?
(Padidar, Landreman, Bindel)

Pareto frontier (QH with 4 field periods)



Aspect ratio 3.3, 5, 8.67





Continuation

Algorithm in this case: continuation in A

- Start at one Pareto point $(A(x), Q(x))$
- Write stationarity conditions via

$$\begin{aligned} \nabla Q(x) + \lambda \nabla A(x) &= 0 \\ \lambda (A(x) - A^*) &= 0 \\ A(x) &\leq A^* \end{aligned}$$

- Differentiate vs A^* to get tangent direction

$$\begin{bmatrix} \nabla^2 Q(x) + \lambda \nabla^2 A(x) & \nabla A(x) \\ \nabla A(x)^T & 0 \end{bmatrix} \begin{bmatrix} x' \\ \lambda' \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- Predictor moves a little in tangent direction
- Correct prediction via local solver (e.g. Newton)
- Can re-use Hessians, etc for more efficiency

Which parameterization?

What if Pareto frontier goes vertical?

- Can switch to using Q as continuation parameter
- *Or* use a **pseudo-arclength** parameter
- Generalizations to more than two functions are available (e.g. normal boundary intersection)

Some concluding notes

> *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. Beyond formulating reasonable objectives, 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!