

# Stellarator Optimization with Constraints

Rory Conlin <sup>1</sup>†, Patrick Kim <sup>2</sup>, Daniel W. Dudt <sup>1</sup>, Dario Panici <sup>1</sup>, and Egemem Kolemen <sup>1</sup>‡

<sup>1</sup>Princeton University, Princeton, New Jersey 08544

(Received xx; revised xx; accepted xx)

In this work we consider the problem of optimizing a stellarator subject to hard constraints on the design variables and physics properties of the equilibrium. We survey current numerical methods for handling these constraints, and summarize a number of methods from the wider optimization community that have not been used extensively for stellarator optimization thus far. We demonstrate the utility of new methods of constrained optimization by optimizing a QA stellarator for favorable physics properties while preventing strong shaping of the plasma boundary which can be difficult to create with external current sources.

## 1. Introduction

Stellarator optimization is naturally full of constraints that must be satisfied by an optimized configuration. Many of these constraints are due to engineering requirements, such as reasonable aspect ratio, minimum coil-plasma distance, or maximum curvature of the coils or plasma boundary. Others come from physics requirements, such as having a prescribed rotational transform at the boundary for an island divertor (Feng *et al.* 2006), having a stable magnetic well (Landreman & Jorge 2020), or ensuring the net current is zero. There are also constraints to enforce self-consistency between different models of the plasma, such as requiring that the prescribed current in an ideal MHD equilibrium solver matches with the actual bootstrap current from a kinetic calculation (Landreman *et al.* 2022), or that the coils correctly cancel the normal field on the plasma boundary. Perhaps the most important constraint is that any optimized configuration must actually be in MHD equilibrium.

A generic optimization problem can be written as:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{g}_E(\mathbf{x}) = 0 \\ & \mathbf{l} \leq \mathbf{g}_I(\mathbf{x}) \leq \mathbf{u} \end{aligned} \tag{1.1}$$

where  $\mathbf{x} \in \mathcal{R}^n$  is a vector of decision variables,  $f : \mathcal{R}^n \rightarrow \mathcal{R}^1$  is the objective function,  $\mathbf{g}_E : \mathcal{R}^n \rightarrow \mathcal{R}^{m_1}$  are equality constraints,  $\mathbf{g}_I : \mathcal{R}^n \rightarrow \mathcal{R}^{m_2}$  are inequality constraints, and  $\mathbf{l}, \mathbf{u} \in \mathcal{R}^{m_2}$  are lower and upper bounds on the inequality constraints (which may be infinite for one sided constraints). By introducing slack variables  $\mathbf{s}$ , we can transform the inequality constraint into an equality constraint and simple bounds:

† Email address for correspondence: wconlin@princeton.edu

‡ Email address for correspondence: ekolemen@princeton.edu

$$\begin{aligned}
& \min_{\mathbf{x}, \mathbf{s}} f(\mathbf{x}) \\
& \text{s.t.} \quad \mathbf{g}_E(\mathbf{x}) = 0 \\
& \quad \quad \mathbf{g}_I(\mathbf{x}) - \mathbf{s} = 0 \\
& \quad \quad \mathbf{l} \leq \mathbf{s} \leq \mathbf{u}
\end{aligned} \tag{1.2}$$

Which if we absorb  $\mathbf{s}$  into  $\mathbf{x}$  and combine the constraints into a single function  $\mathbf{g}$ , we can write in standard form as:

$$\begin{aligned}
& \min_{\mathbf{x}} f(\mathbf{x}) \\
& \text{s.t.} \quad \mathbf{g}(\mathbf{x}) = 0 \\
& \quad \quad \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}
\end{aligned} \tag{1.3}$$

In [section 2](#) we will summarize methods for handling constraints that are already in use in stellarator optimization such as linear constraint projection and sum of squares. In [section 3](#) we introduce several methods from the wider optimization literature and discuss their properties and relevance for the stellarator optimization field. In [section 4](#) we detail the implementation of a new augmented Lagrangian optimization algorithm into the DESC code, and demonstrate it's use with an example problem in [section 5](#).

## 2. Existing methods

### 2.1. Linear Equality Constraints

Constraints where the decision variables  $\mathbf{x}$  appear only linearly can be handled efficiently by suitably redefining variables. A problem with linear constraints can be written as:

$$\begin{aligned}
& \min_{\mathbf{x} \in R^n} f(\mathbf{x}) \\
& \text{s.t.} \quad A\mathbf{x} - \mathbf{b} = 0
\end{aligned} \tag{2.1}$$

Where  $A \in R^{m \times n}$  and  $\mathbf{b} \in R^m$ ,  $m \leq n$  are the coefficients that define the linear constraint. We can then decompose  $\mathbf{x} = \mathbf{x}_p + Z\mathbf{y}$  where  $\mathbf{x}_p \in R^n$  is any particular solution to the under-determined system  $A\mathbf{x} = \mathbf{b}$  (typically taken to be the least-norm solution),  $Z \in R^{n \times (n-m)}$  is a (typically orthogonal) representation for the null space of  $A$ , such that  $AZ = 0$ , and  $\mathbf{y} \in R^{n-m}$  is an arbitrary vector. This allows us to write the problem as:

$$\min_{\mathbf{y} \in R^{n-m}} f(\mathbf{x}_p + Z\mathbf{y}) \tag{2.2}$$

which is now an unconstrained problem, where any value for  $\mathbf{y}$  is feasible, and the full solution  $\mathbf{x}^*$  can be recovered as  $\mathbf{x}_p + Z\mathbf{y}^*$ .

While simple to deal with, linear equality constraints are very limited in their applicability. Most constraints that are of interest in stellarator optimization are nonlinear, making linear equality constraints only useful for enforcing simple boundary conditions on the plasma boundary and profiles.

## 2.2. Sum of Squares

Perhaps the simplest way to (approximately) solve [Equation 1.3](#) is with a quadratic penalty method, where the objective and constraints are combined into a single function, which is then minimized without constraints:

$$\min_{\mathbf{x}} f(\mathbf{x}) + w_1 \|\mathbf{g}(\mathbf{x})\|^2 + w_2 \|I_{(l,u)}(\mathbf{x})\|^2 \quad (2.3)$$

Where  $w_1$  and  $w_2$  are weights, and  $I_{(l,u)}(\mathbf{x})$  is a "deadzone" function:

$$I_{(l,u)}(\mathbf{x}) = \begin{cases} 0 & \text{if } l \leq \mathbf{x} \leq u \\ \max(u - \mathbf{x}, \mathbf{x} - l) & \text{otherwise} \end{cases} \quad (2.4)$$

This formulation is used by a number of codes for optimizing stellarator equilibria, such as SIMSOPT, STELLOPT, ROSE, and DESC. While simple to implement, this formulation has a number of drawbacks:

- (i) The weights  $w$  must be determined by the user, often through extensive trial and error
- (ii) The constraints are only satisfied as  $w \rightarrow \infty$ , leading to a badly scaled problem.
- (iii) The max term in the deadzone function introduces a discontinuity in the second derivative of the objective, which can hinder progress of some optimization methods that assume the objective and constraints to be at least C2.

## 2.3. Projection Methods

The constraint of MHD equilibrium is most commonly dealt with by using a fixed boundary equilibrium solver at each step in the optimization. Given the plasma boundary and profiles at the latest optimization step, the equilibrium solver is run to determine the full solution throughout the plasma volume, and this is then used to evaluate the objective function. This is akin to a projection method, where after each step of the optimizer, the solution is projected back onto the feasible region  $\{\mathbf{x} \mid \mathbf{g}(\mathbf{x}) = 0\}$ . This also suffers from a number of drawbacks:

- (i) The equilibrium must be re-solved at each step of the optimizer, possibly multiple times if finite differences are used to evaluate derivatives. This can be made somewhat more efficient if a warm start is used based on the previous accepted step ([Conlin et al. 2023](#); [Dudt et al. 2023](#)), though this is still often the most expensive part of the optimization loop.
- (ii) Projection type methods are limited in applicability to constraints for which a projection operator can be defined. MHD equilibrium solvers can serve this function for equilibrium constraints, but a projection operator for a general nonlinear constraint is not known.
- (iii) Enforcing that the constraints are satisfied at each step of the optimization may be too strict, and cause the optimizer to stop at a poor local minimum.

This last point is illustrated in [Figure 1](#). Points where the constraint curve  $g(\mathbf{x}) = 0$  is tangent to the contours of  $f$  are local minima of the constrained problem, where following the constraint curve will lead to higher objective function values, but moving in a direction to decrease the objective function will violate the constraints. If we enforce that the constraint is satisfied at each step, we will follow the path in red and stop at the local minimum  $x_1$ . However, if we are allowed to temporarily violate the constraints, we can make rapid progress towards the better local minimum  $x_2$ .

Some may worry that not enforcing MHD equilibrium at each step may lead the optimizer in directions which are not dynamically accessible during an actual plasma

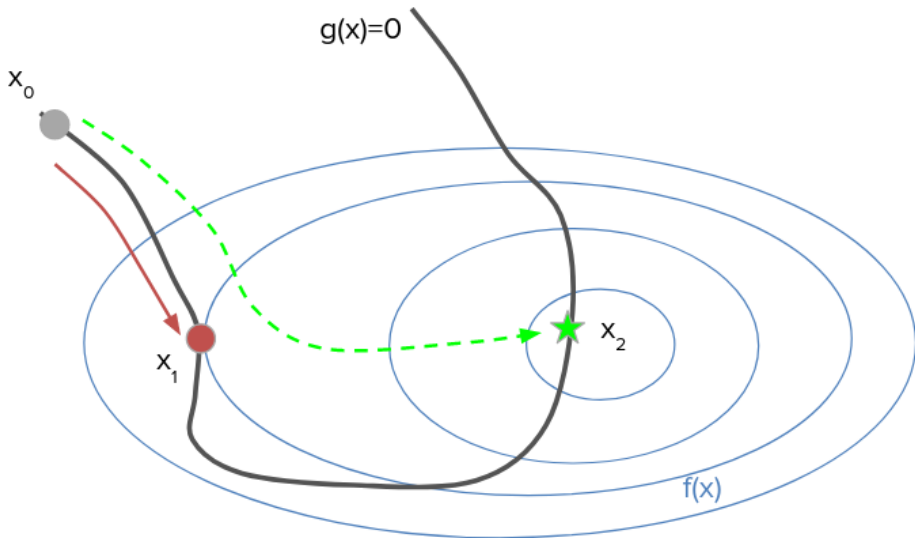


Figure 1: Sketch of optimization landscape (contours of  $f$  in blue) with constraints (curve of  $g(x) = 0$  in black). Starting from  $x_0$  and enforcing the constraints exactly at each step will follow the red path and end at  $x_1$ . If we allow ourselves to temporarily violate the constraints, we can follow the green path and arrive at the better solution  $x_2$ .

discharge, where the plasma is assumed to be in equilibrium, however it is important to note that the path taken through phase space during optimization in general has no connection to the path taken during an actual discharge, and just because the optimizer finds a path that violates equilibrium does not mean there isn't a nearby path that satisfies it. Dynamical accessibility is an important aspect of stellarator optimization, but the expense of time dependent simulations preclude their use inside an optimization loop, so are generally limited to post-optimization validation and analysis.

### 3. Methods for Constrained Optimization

We consider a general constrained optimization problem of the form:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{g}_E(\mathbf{x}) = 0 \\ & \mathbf{g}_I(\mathbf{x}) \geq 0 \end{aligned} \tag{3.1}$$

where as before  $\mathbf{g}_E$  and  $\mathbf{g}_I$  denote the equality and inequality constraints respectively. Most analysis and algorithms start from the Lagrangian for this problem which is

$$\mathcal{L}(\mathbf{x}, \mathbf{y}_E, \mathbf{y}_I) = f(\mathbf{x}) - \mathbf{y}_E^T \mathbf{g}_E(\mathbf{x}) - \mathbf{y}_I^T \mathbf{g}_I(\mathbf{x}) \tag{3.2}$$

where  $\mathbf{y}_E$  and  $\mathbf{y}_I$  are the *Lagrange multipliers* associated with the equality and inequality constraints respectively. The conditions for a given set  $(\mathbf{x}^*, \mathbf{y}_E^*, \mathbf{y}_I^*)$  to be a solution to [Equation 3.1](#) are the so called Karush-Kuhn-Tucker or KKT conditions ([Nocedal & Wright 2006](#)):

$$\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}^*, \mathbf{y}_E^*, \mathbf{y}_I^*) = \nabla f - \mathbf{y}_E^T \nabla \mathbf{g}_E - \mathbf{y}_I^T \nabla \mathbf{g}_I = 0 \quad (3.3a)$$

$$\mathbf{g}_E(\mathbf{x}^*) = 0 \quad (3.3b)$$

$$\mathbf{g}_I(\mathbf{x}^*) \geq 0 \quad (3.3c)$$

$$\mathbf{y}_I^* \geq 0 \quad (3.3d)$$

$$\mathbf{y}_E^* \cdot \mathbf{g}_E(\mathbf{x}^*) = 0 \quad (3.3e)$$

$$\mathbf{y}_I^* \cdot \mathbf{g}_I(\mathbf{x}^*) = 0 \quad (3.3f)$$

The first condition is reminiscent of the first order optimality condition for an unconstrained problem,  $\nabla f(\mathbf{x}^*) = 0$ , with the objective function  $f$  replaced by the Lagrangian  $\mathcal{L}$ , however it is important to note that we do not seek a minimum of the Lagrangian. It can be shown that solutions to [Equation 3.3](#) always lie at a saddle point where the Lagrangian is minimized with respect to the primal variables  $\mathbf{x}$  and maximized with respect to the dual variables  $\mathbf{y}$ .

It is useful to take a moment to consider the meaning and utility of the Lagrange multipliers  $\mathbf{y}$ . A first insight into the meaning of Lagrange multipliers can be found from the last of the KKT conditions in [Equation 3.3f](#):

$$\mathbf{y}_I^* \cdot \mathbf{g}_I(\mathbf{x}^*) = 0 \quad (3.4)$$

Meaning that either the constraint is zero, or it's associated multiplier is zero for each constraint. For a problem in the form of [Equation 3.1](#), an inequality constraint not equal to zero is said to be "non binding" or "not active", meaning that the constraint has no effect on the solution, and the minimum would not change if the constraint were removed. This is contrasted with a "binding" or "active" constraint that is preventing the solution from improving the objective  $f$ . This suggests that the Lagrange multipliers contain information about how much the constraints are affecting the solution. This can be made more explicit by considering the following problem:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0 \end{aligned} \quad (3.5)$$

The Lagrangian for this problem is

$$\mathcal{L}(x, y) = f(x) - yg(x) \quad (3.6)$$

We can now consider what happens if we perturb the constraint  $g = 0$  to  $g + \delta g = 0$ . From [Equation 3.3a](#) we see that at an optimum  $x^*$  the change in the Lagrangian must be zero to first order, which gives

$$\delta f = y\delta g \quad (3.7)$$

Here we see that the Lagrange multipliers tell us how much our objective can improve by relaxing our constraints, or the so called "marginal cost" of enforcing the constraint  $g = 0$ . Additionally, the sign of the multipliers tells us which direction we should relax the constraints to improve the objective.

We can use a similar approach to consider trade offs between conflicting constraints, where if we have constraints  $g_1$  and  $g_2$  that cannot both be satisfied, the ratio of their

multipliers  $y_1/y_2$  tells us how much we can expect to improve  $g_1$  by relaxing  $g_2$  and vice versa.

A number of methods have been developed for solving constrained optimization problems which we briefly summarize in the following subsections. More information can be found in standard texts on numerical optimization (Nocedal & Wright 2006; Conn *et al.* 2000)

### 3.1. Interior Point Methods

A popular class of methods for dealing with inequality constraints are the so-called "interior point" methods, or sometimes referred to as "log barrier" methods. These methods solve problems of the form:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{x}) = 0 \\ & 0 \leq \mathbf{x} \end{aligned} \tag{3.8}$$

(any problem of the form of Equation 1.3 can be transformed into this form by appropriate change of variables). The bound on the variables is replaced by a logarithmic penalty term:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) - \mu \sum_i \log x_i \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{x}) = 0 \end{aligned} \tag{3.9}$$

This problem is then solved repeatedly with an equality constrained optimization method such as sequential quadratic programming (see the following section) for a decreasing sequence of barrier parameters  $\mu$ . For finite values of  $\mu$ , the log term pushes the solution into the interior of the feasible set (hence the name), and the solution is allowed to approach the boundary as the barrier gets sharper as  $\mu$  approaches zero.

The interior point method is the basis of a number of popular commercial and open source optimization codes (Biegler & Zavala 2009; Wächter & Biegler 2006), but we have found that in practice they do not perform well on the types of problems commonly encountered in stellarator optimization. We hypothesize this is due to two primary reasons:

(i) Stellarator optimization problems tend to be badly scaled due to spectral discretizations leading to different variables having exponentially more effect on the objective and constraints. In many cases this is not a problem, but the log barrier term can also lead to very poorly conditioned derivatives as the barrier is approached. Combined, this can significantly slow convergence or cause early termination due to loss of precision.

(ii) Many interior point methods require any initial guess to be strictly feasible with respect to any inequality constraints. In many cases, an initial guess can be found from a near axis expansion or surrogate model, but it will generally not be a strictly feasible solution.

### 3.2. Sequential Quadratic Programming (SQP) Methods

Another popular class of methods are based on sequentially minimizing an approximate model of the objective and constraints. Typically, the objective is modelled as quadratic and the constraints are modelled as linear, leading to a standard quadratic program. If the current estimate of the solution is  $\mathbf{x}_k$ , we seek the next iterate  $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}$  by solving the following subproblem:

$$\begin{aligned}
\min_{\mathbf{p}} \quad & \mathbf{p}^T H \mathbf{p} + \mathbf{g}^T \mathbf{p} \\
\text{s.t.} \quad & A_E \mathbf{p} = -\mathbf{g}_E(\mathbf{x}_k) \\
& \mathbf{l} - \mathbf{g}_I(\mathbf{x}_k) \leq A_I \mathbf{p} \leq \mathbf{u} - \mathbf{g}_I(\mathbf{x}_k)
\end{aligned} \tag{3.10}$$

where  $\mathbf{g}$  and  $H$  are the gradient and Hessian of  $f$  and  $A_E$  and  $A_I$  are the Jacobian matrices of the equality and inequality constraints  $\mathbf{g}_E$  and  $\mathbf{g}_I$ .

One common issue with SQP methods is the possible inconsistency of the linearized constraints, such that the subproblem [Equation 3.10](#) has no solution. This problem can be made worse by the inclusion of a trust region constraint, as is often used to ensure global convergence. To overcome this, the full step  $\mathbf{p}$  is typically broken into two sub-steps, one of which attempts to achieve feasibility with respect to the linearized constraints as much as possible, while the other attempts to reduce the objective function. A merit function is then used to balance between improving the objective and decreasing the constraint violation ([Conn et al. 2000](#)).

An alternative formulation, called "active set" methods, makes use of the fact that if one knows ahead of time which inequality constraints are active at the solution, then we can turn those into equality constraints and ignore all other inequality constraints, leading to a purely equality constrained problem which is often easier to solve. Active set methods make a guess for which constraints are active at the solution and iteratively update this set based on the results of sequential equality constrained optimizations.

Similar to interior point methods, SQP methods form the core of a number of commercial and open source optimization codes ([Gill et al. 2005](#); [Johnson & Schueller 2021](#)). However, we similarly see that their performance is limited on the types of problems encountered in practice.

(i) If the Hessian matrix is not positive definite (which is common when far from the solution) subproblem [Equation 3.10](#) is in general NP hard to solve (in that there is no polynomial time algorithm), so one must often be satisfied with an approximate solution which can significantly slow global convergence

(ii) Active set methods can be very efficient for small problems, however the number of iterations required to determine the correct active set grows with the number of constraints, making active set methods inefficient for high dimensional problems.

### 3.3. Augmented Lagrangian Methods

For the quadratic penalty method considered in [Equation 2.3](#), one can show that the constraint violation is inversely proportional to the penalty parameter:

$$g_i(\mathbf{x}_k) \approx -y_i^*/\mu_k \tag{3.11}$$

where  $y_i^*$  is the true Lagrange multiplier (from the solution to [Equation 3.3](#)) associated with the constraint  $g_i$  and  $\mu_k$  is the penalty parameter. We see that if we want to satisfy the constraints exactly, we must have  $\mu \rightarrow \infty$ , which as discussed can be numerically intractable. However, if we consider the "Augmented" Lagrangian

$$\mathcal{L}_A(\mathbf{x}, \mathbf{y}, \mu) = f(\mathbf{x}) - \mathbf{y}^T \mathbf{g}(\mathbf{x}) + \frac{\mu}{2} \|\mathbf{g}(\mathbf{x})\|^2 \tag{3.12}$$

where  $\mathbf{y}$  are estimates of the true Lagrange multipliers, one can show that now ([Nocedal & Wright 2006](#))

$$g_i(\mathbf{x}_k) \approx -(y_i^* - y_i^k)/\mu_k \tag{3.13}$$

So that if our estimate of the true Lagrange multipliers is accurate, we can significantly reduce the constraint violation without requiring the penalty parameter to grow without bound. This also gives us a rule to update our estimate of the multipliers:

$$y_i^{k+1} = y_i^k - \mu_k g_i(\mathbf{x}_k) \quad (3.14)$$

Minimizing the augmented Lagrangian with fixed  $\mathbf{y}$  and  $\mu$  is an unconstrained or bound constrained problem, for which standard gradient or Newton based methods can be used. From this solution we obtain an approximation to the true solution of the constrained problem. We can then update the multipliers and penalty term and re-optimize, eventually converging to the true solution. This is the basic method used in several widely used codes (Conn *et al.* 2013; Fowkes *et al.* 2023) and can also be used as a generic technique to handle constraints for algorithms that otherwise would only be able to solve unconstrained problems (Johnson & Schueller 2021).

One can also construct an augmented Lagrangian for a least squares problem. Because  $L(\mathbf{x}, \mathbf{y}, \mu)$  is only ever minimized with respect to  $\mathbf{x}$ , one can add any quantity independent of  $\mathbf{x}$  without affecting the result. In particular, if we add  $\|\mathbf{y}\|^2/2\mu$ , we get

$$\mathcal{L}_A(\mathbf{x}, \mathbf{y}, \mu) = \frac{1}{2} \|\mathbf{f}(\mathbf{x})\|^2 - \mathbf{y}^T \mathbf{g}(\mathbf{x}) + \mu/2 \|\mathbf{g}(\mathbf{x})\|^2 + \|\mathbf{y}\|^2/2\mu \quad (3.15)$$

We can then complete the square to obtain

$$\mathcal{L}_A(\mathbf{x}, \mathbf{y}, \mu) = \frac{1}{2} \|\mathbf{f}(\mathbf{x})\|^2 + \frac{1}{2} \|\mathbf{y} - \sqrt{\mu} \mathbf{g}(\mathbf{x})\|^2 \quad (3.16)$$

This allows us to use standard unconstrained or bound constrained least squares routines to solve the augmented Lagrangian subproblem, which has a number of benefits:

(i) Super-linear convergence can be obtained using only first derivatives of  $\mathbf{f}$  and  $\mathbf{g}$  (as opposed to standard Newton methods which require second derivatives of the objective and constraints).

(ii) The quadratic subproblem that arises in least squares is always convex, for which the exact solution can be found efficiently.

## 4. Constrained Optimization in DESC

As previously mentioned there are a wide variety of commercial and open source codes for constrained optimization, many of which have been interfaced to the DESC stellarator optimization code (Dudt & Kolemen 2020; Panici *et al.* 2023; Conlin *et al.* 2023; Dudt *et al.* 2023, 2024). In practice, we have found that most of the existing methods perform somewhat poorly on the types of problems encountered in stellarator optimization. We hypothesize this is for a number of reasons:

(i) Many commercial solvers assume a particular structure for the problem, such as quadratic, conic, and semi-definite programming solvers. Others rely on convexity, partial separability, or other conditions that generally don't hold in stellarator optimization.

(ii) Many codes are designed and optimized for solving very high dimensional problems ( $O(10^6)$  variables or more) but with a high degree of sparsity, and make extensive use of sparse and iterative linear algebra methods. In stellarator optimization problems, we generally do not have significant sparsity due to the wide use of spectral methods, making these methods inefficient.

(iii) The types of problems in stellarator optimization are often poorly scaled, due to a wide range of scales in physical units, and due to spectral discretizations. This can lead



to failure of iterative linear algebra subproblems without a proper preconditioner (which may not be easy to determine) and can also slow progress of the overall optimization because the optimization landscape is strongly anisotropic (meaning the objective is much more sensitive to changes in certain directions in parameter space than others, such as a long narrow valley).

(iv) No existing methods that we are aware of are able to take advantage of GPU acceleration for the underlying linear algebra, limiting the overall efficiency on modern heterogeneous computing environments.

Because of this, we have implemented a new optimizer based on the least squares augmented Lagrangian formalism from [subsection 3.3](#), and a sketch of the algorithm is found in [Algorithm 1](#).

---

**Algorithm 1** Augmented Lagrangian Algorithm (adapted from [\(Nocedal & Wright 2006; Conn et al. 2000\)](#))

---

**Require:**  $\mu_0 > 0$ ,  $\omega_* > 0$ ,  $\eta_* > 0$

$\mu_k \leftarrow \mu_0$

$\omega_k \leftarrow 1/\mu_0$

$\eta_k \leftarrow 1/\mu_0^{0.1}$

$k \leftarrow 1$

**while**  $\|\nabla \mathcal{L}_A(\mathbf{x}_k, \mathbf{y}_k, \mu_k)\| > \omega_*$  or  $\|\mathbf{g}(\mathbf{x}_k, \mathbf{y}_k, \mu_k)\| > \eta_*$  **do**

$\mathbf{x}_{k+1} \leftarrow \arg \min_{\mathbf{x}} \mathcal{L}_A(\mathbf{x}_k, \mathbf{y}_k, \mu_k)$        $\triangleright$  using unconstrained or bound constrained method, solved to tolerance  $\omega_k$

**if**  $\|\mathbf{g}(\mathbf{x}_k, \mathbf{y}_k, \mu_k)\| < \eta_k$  **then**       $\triangleright$  Constraints are making good progress, update multipliers

$\mathbf{y}_{k+1} \leftarrow \mathbf{y}_k - \mu_k \mathbf{g}(\mathbf{x}_{k+1})$

$\mu_{k+1} \leftarrow \mu_k$

$\omega_{k+1} \leftarrow \omega_k / \mu_{k+1}$

$\eta_{k+1} \leftarrow \eta_k / \mu_{k+1}$

**else**       $\triangleright$  Need to improve constraint satisfaction, increase penalty term

$\mathbf{y}_{k+1} \leftarrow \mathbf{y}_k$

$\mu_{k+1} \leftarrow 10\mu_k$

$\omega_{k+1} \leftarrow 1/\mu_{k+1}$

$\eta_{k+1} \leftarrow 1/\mu_{k+1}$

**end if**

$k \leftarrow k + 1$

**end while**

---

The precise rules for updating the tolerances and penalty parameters can be varied, see [\(Conn et al. 2000\)](#) for a more complex version, though we have found that in practice the results and performance do not significantly depend on the choice of hyper-parameters. These rules are effectively automatically finding the correct weights to manage the trade-off between improving the objective value and constraint feasibility that would normally require extensive user trial and error.

## 5. Examples

Many optimized stellarators have very strongly shaped boundaries, and a "bean" shaped cross section is quite common, appearing in designs such as NCSX (shown in [Figure 2](#)), W7-X, ARIES-CS, and many recent designs such as the "precise" quasisymmetric equilibria of Landreman and Paul ([Landreman & Paul 2022](#)). These strongly

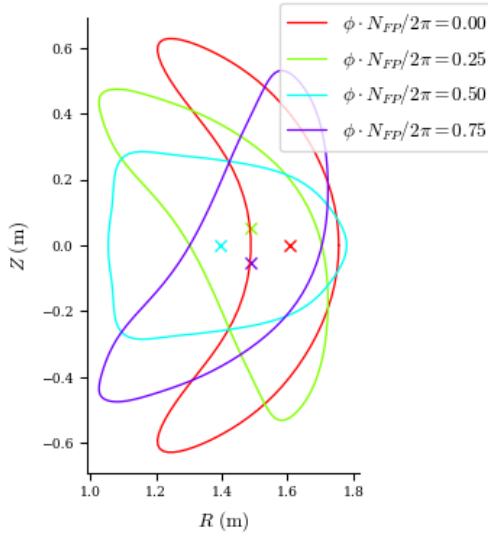


Figure 2: Plasma boundary of NCSX at several cross sections, with the "bean" section at  $\phi = 0$ .

indented shapes can be difficult to create with external coils or magnets, often requiring coils that are very close on the inboard concave side (Kappel *et al.* 2024; Landreman & Boozer 2016).

We can attempt to alleviate this problem with constrained optimization by properly constraining the curvature of the surface. We consider a surface parameterized by poloidal angle  $\theta$  and toroidal angle  $\zeta$ , with parameterization  $\mathbf{r}(\theta, \zeta)$ . The covariant basis vectors are given by  $\mathbf{r}_\alpha(\theta, \zeta) = \partial_\alpha \mathbf{r}(\theta, \zeta)$  for  $\alpha \in (\theta, \zeta)$ . From these we can construct the "first fundamental form" (Do Carmo 2016)

$$\mathcal{F} = \begin{pmatrix} E & F \\ F & G \end{pmatrix} \quad (5.1)$$

where the coefficients are dot products of the first derivative of the position vector:

$$E = \mathbf{r}_\theta \cdot \mathbf{r}_\theta \quad (5.2)$$

$$F = \mathbf{r}_\theta \cdot \mathbf{r}_\zeta \quad (5.3)$$

$$G = \mathbf{r}_\zeta \cdot \mathbf{r}_\zeta \quad (5.4)$$

$$(5.5)$$

Similarly we can define the "second fundamental form"

$$\mathcal{S} = \begin{pmatrix} L & M \\ M & N \end{pmatrix} \quad (5.6)$$

Where the coefficients are given by the projection of the second derivatives of the position vector onto the surface normal:

$$L = \mathbf{r}_{\theta\theta} \cdot \mathbf{n} \quad (5.7)$$

$$M = \mathbf{r}_{\theta\zeta} \cdot \mathbf{n} \quad (5.8)$$

$$N = \mathbf{r}_{\zeta\zeta} \cdot \mathbf{n} \quad (5.9)$$

$$(5.10)$$

We can then compute the principal curvatures  $\kappa_1$  and  $\kappa_2$  of the surface as the solutions to the generalized eigenvalue problem

$$\mathcal{S}\mathbf{u} = \kappa\mathcal{F}\mathbf{u} \quad (5.11)$$

where the eigenvectors  $\mathbf{u}$  are called the principal directions. The principal curvatures are the maximum and minimum of the inverse radii of curvature of all the curves that are tangent to the surface at a given point, with the sign convention that positive curvature means the surface curves towards the normal vector.

On the inboard side of a toroidal surface, we expect at least one of the principal curvatures to be positive because the surface curves towards the normal in the toroidal direction, while at least one should be negative on the outboard side. Indenting of the poloidal cross section characteristic of the bean shape would manifest as both principal curvatures being positive on the inboard side, and both negative on the outboard side.

To prevent strong indentation, we will use the mean curvature, defined as  $H = 1/2(\kappa_1 + \kappa_2)$  and enforce that the mean curvature is everywhere negative. On the outboard side, where both curvatures are already negative this should have minimal effect, while on the inboard it forces any indent to have a larger radius of curvature than the local major radius.

In addition to this constraint on the curvature of the plasma surface, we also include constraints to enforce MHD equilibrium, fix the major radius  $R_0$ , aspect ratio  $R_0/a$ , total enclosed volume  $V$  (which we fix equal to the volume of the initial guess, a circular torus) and include inequality constraints on the rotational transform profile  $\iota$ . We also fix the profiles of pressure  $p$  and current  $j$ , and the total enclosed flux  $\Psi$  to give an average field strength of 1T. In summary, our constraints are

$$\begin{aligned} R_0 &= 1 \text{ m} \\ R_0/a &= 6 \\ V &= V_{init} \sim 0.55 \text{ m}^3 \\ 0.43 &< \iota(\rho) < 0.5 \\ H(\theta, \zeta) &< 0 \text{ m}^{-1} \\ p(\rho) &= 0 \text{ Pa} \\ j(\rho) &= 0 \text{ A} \\ \Psi &= 0.087 \text{ Tm}^2 \\ \mathbf{J} \times \mathbf{B} - \nabla p &= 0 \end{aligned}$$

As our objective, we use the "two-term" metric  $f_C$  for quasisymmetry (section 2.3.2 of (Dudt *et al.* 2023)), targeting quasi-axisymmetry at 10 surfaces evenly spaced throughout the volume.

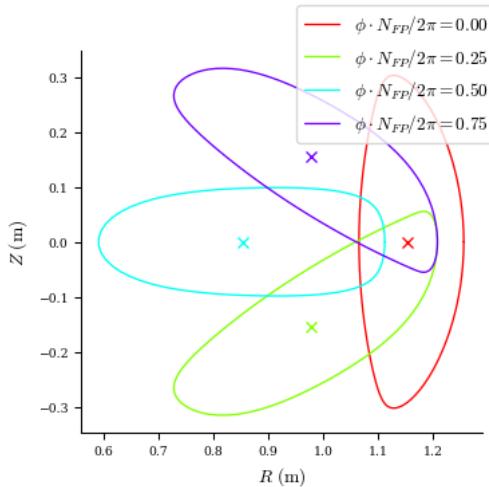


Figure 3: Plasma boundary of new optimized configuration with constraint on surface curvature.

$$f_C = (M\iota - N)(\mathbf{B} \times \nabla\psi) \cdot \nabla B - (MG + NI)\mathbf{B} \cdot \nabla B \quad (5.12)$$

Here,  $\mathbf{B}$  is the magnetic field with magnitude  $B$ ,  $M$  and  $N$  give the type of quasisymmetry (for QA,  $M = 1$ ,  $N = 0$ ), and  $G$  and  $I$  are the poloidal and toroidal covariant components of the field in Boozer coordinates.

The plasma boundary after optimization is shown in [Figure 3](#), and we can see that we have successfully avoided any concave regions that may be difficult to produce with external coils. In [Figure 6](#) we also see that the level of quasisymmetry in this new configuration compares quite well with similar optimized designs such as NCSX. However, upon further analysis we find that this new configuration has a negative magnetic well, indicating it is MHD unstable.

We can attempt to rectify this by adding a constraint that the magnetic well be positive and re-running the optimization. Doing this, we find that the optimizer does not converge, indicating that the constraints cannot all be satisfied at the same time. This is perhaps expected, as empirical evidence suggests that bean-like cross sections can be favorable for MHD stability ([Nührenberg & Zille 1986](#); [Nührenberg 2010](#)), though may not be strictly required from a near axis formalism ([Rodríguez 2023](#)).

However, in this case, we can use the estimated Lagrange multipliers to get an idea of which constraints are the most limiting. Because the curvature constraint is enforced point wise in real space, the Lagrange multipliers also have a local characteristic, and we can plot them over the surface to see where the optimizer wanted to indent the surface, as shown in [Figure 4](#). As expected, the curvature constraint is most prominent on the inboard side, and the negative sign of the multipliers indicates that increasing the curvature in those regions will be favorable for stability.

Using this information, we re-run the optimization with the magnetic well constraint, but reduce the upper bound on mean curvature from  $0 \text{ m}^{-1}$  to  $0.5 \text{ m}^{-1}$ . With this relaxed constraint we are able to achieve a stable magnetic well, while maintaining most of the benefits of the curvature constraint, as shown in [Figure 5](#). This is not without some cost, as we see in [Figure 6](#) that the level of quasisymmetry for the stable configuration

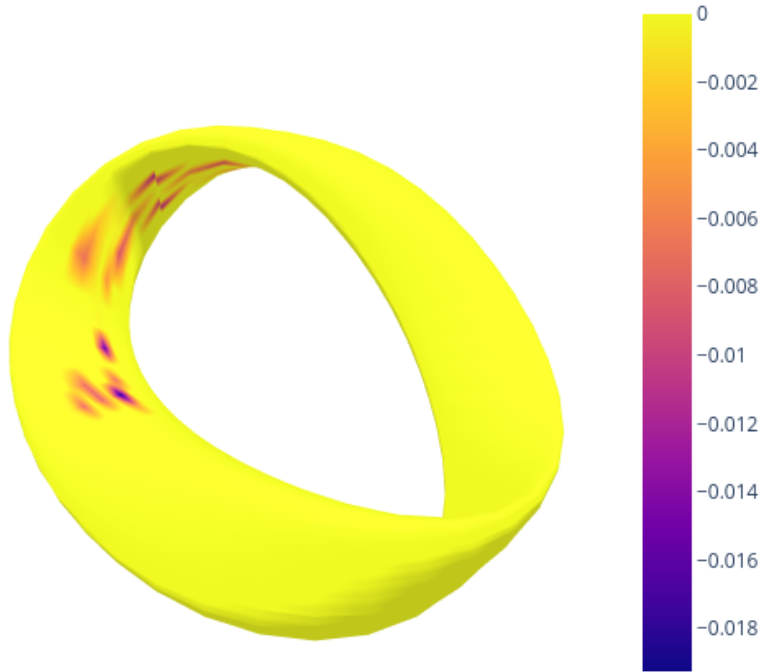


Figure 4: Lagrange multipliers for the curvature constraint plotted on the surface. The dark areas indicate places where the constraint is binding, and the optimizer wants to indent the boundary in those locations to improve the magnetic well.

is degraded somewhat compared to the unstable one, though still at reasonable levels compared to other optimized configurations such as NCSX.

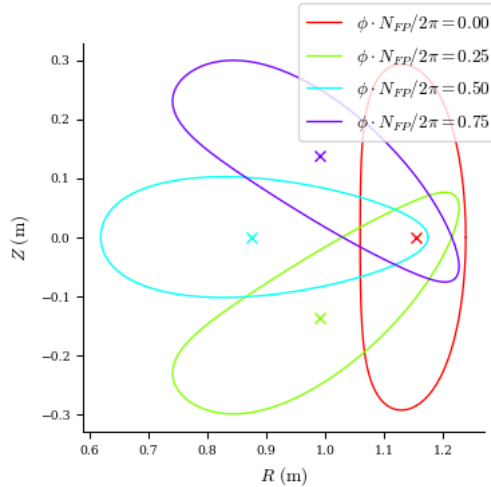


Figure 5: Plasma boundary of new optimized configuration with relaxed curvature constraint and stable magnetic well.

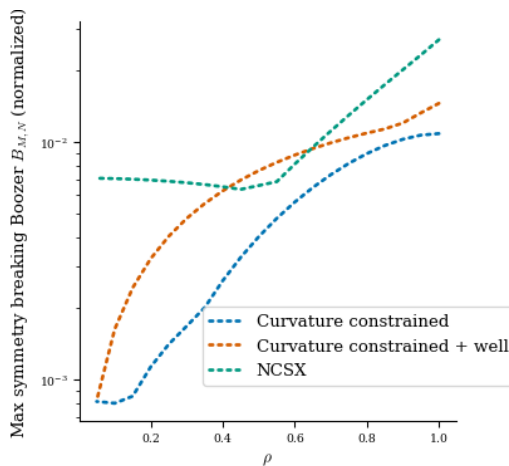


Figure 6: Quasisymmetry error measured by the maximum amplitude of the symmetry breaking Boozer harmonics for the newly optimized configuration with and without magnetic well constraint, compared to NCSX.

## 6. Conclusion

In this work we have described the many ways that constraints naturally arise in stellarator optimization problems and surveyed a number of methods to solve constrained optimization problems, including methods that have not been widely applied in the field thus far. We have further developed and implemented a new constrained optimization algorithm in the DESC code and demonstrated its usefulness in designing a newly optimized equilibrium that achieves good quasisymmetry and a stable magnetic well without strong shaping of the plasma boundary. We are hopeful that these new techniques will open new avenues of possibility for the design and optimization of future stellarators, and significantly reduce the amount of guesswork and "black magic" required when

designing objectives for optimization, as well as offering insights into trade offs between different objectives and constraints.

## Funding

This work was supported by the U.S. Department of Energy under contract numbers DE-AC02-09CH11466, DE- SC0022005 and Field Work Proposal No. 1019. The United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

## Declaration of interests

The authors report no conflict of interest.

## Data availability statement

The source code to generate the results and plots in this study are openly available in DESC at <https://github.com/PlasmaControl/DESC> or <http://doi.org/10.5281/zenodo.4876504>

## Author ORCID

R. Conlin, <https://orcid.org/0000-0001-8366-2111>; D. Dudt, <https://orcid.org/0000-0002-4557-3529>; D. Panici, <https://orcid.org/0000-0003-0736-4360>; E. Kolemen, <https://orcid.org/0000-0003-4212-3247>

## REFERENCES

- BIEGLER, LORENZ T & ZAVALA, VICTOR M 2009 Large-scale nonlinear programming using ipopt: An integrating framework for enterprise-wide dynamic optimization. *Computers & Chemical Engineering* **33** (3), 575–582.
- CONLIN, RORY, DUDT, DANIEL W, PANICI, DARIO & KOLEMEN, EGEMEN 2023 The desc stellarator code suite. part 2. perturbation and continuation methods. *Journal of Plasma Physics* **89** (3), 955890305.
- CONN, ANDREW R, GOULD, GIM & TOINT, PHILIPPE L 2013 *LANCELOT: a Fortran package for large-scale nonlinear optimization (Release A)*, , vol. 17. Springer Science & Business Media.
- CONN, ANDREW R, GOULD, NICHOLAS IM & PHILIPPE, L 2000 Toint. trust-region methods. mps/siam series on optimization. *Society for Industrial Mathematics* .
- DO CARMO, MANFREDO P 2016 *Differential geometry of curves and surfaces: revised and updated second edition*. Courier Dover Publications.
- DUDT, DANIEL, CONLIN, RORY, PANICI, DARIO, KOLEMEN, EGEMEN, UNALMIS, KAYA & KIM, PATRICK 2024 DESC.
- DUDT, DW & KOLEMEN, E 2020 Desc: A stellarator equilibrium solver. *Physics of Plasmas* **27** (10).
- DUDT, DANIEL W, CONLIN, RORY, PANICI, DARIO & KOLEMEN, EGEMEN 2023 The desc stellarator code suite part 3: Quasi-symmetry optimization. *Journal of Plasma Physics* **89** (2), 955890201.
- FENG, Y, SARDEI, F, GRIGULL, P, MCCORMICK, K, KISSLINGER, J & REITER, D 2006 Physics of island divertor plasmas in stellarators. *Contributions to plasma physics* **46** (7-9), 504–514.
- FOWKES, JAROSLAV M & OTHERS 2023 Galahad 4.0: an open source library of fortran packages with c and matlab interfaces for continuous optimization. *Journal of Open Source Software* **8** (87), 4882.

- GILL, PHILIP E, MURRAY, WALTER & SAUNDERS, MICHAEL A 2005 Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review* **47** (1), 99–131.
- JOHNSON, STEVEN G & SCHUELLER, JULIEN 2021 Nlopt: Nonlinear optimization library. *Astrophysics Source Code Library* pp. ascl–2111.
- KAPPEL, JOHN, LANDREMAN, MATT & MALHOTRA, DHAIRYA 2024 The magnetic gradient scale length explains why certain plasmas require close external magnetic coils. *Plasma Physics and Controlled Fusion* **66** (2), 025018.
- LANDREMAN, MATT & BOOZER, ALLEN H 2016 Efficient magnetic fields for supporting toroidal plasmas. *Physics of Plasmas* **23** (3).
- LANDREMAN, MATT, BULLER, STEFAN & DREVLAK, MICHAEL 2022 Optimization of quasi-symmetric stellarators with self-consistent bootstrap current and energetic particle confinement. *Physics of Plasmas* **29** (8).
- LANDREMAN, MATT & JORGE, ROGERIO 2020 Magnetic well and mercier stability of stellarators near the magnetic axis. *Journal of Plasma Physics* **86** (5), 905860510.
- LANDREMAN, MATT & PAUL, ELIZABETH 2022 Magnetic fields with precise quasisymmetry for plasma confinement. *Physical Review Letters* **128** (3), 035001.
- NOCEDAL, JORGE & WRIGHT, STEPHEN 2006 *Numerical optimization*. Springer Science & Business Media.
- NÜHRENBERG, JÜRGEN 2010 Development of quasi-isodynamic stellarators. *Plasma Physics and Controlled Fusion* **52** (12), 124003.
- NÜHRENBERG, J & ZILLE, R 1986 Stable stellarators with medium  $\beta$  and aspect ratio. *Physics Letters A* **114** (3), 129–132.
- PANICI, DARIO, CONLIN, RORY, DUDT, DANIEL W, UNALMIS, KAYA & KOLEMEN, EGEMEN 2023 The desc stellarator code suite. part 1. quick and accurate equilibria computations. *Journal of Plasma Physics* **89** (3), 955890303.
- RODRÍGUEZ, EDUARDO 2023 Magnetohydrodynamic stability and the effects of shaping: a near-axis view for tokamaks and quasisymmetric stellarators. *Journal of Plasma Physics* **89** (2), 905890211.
- WÄCHTER, ANDREAS & BIEGLER, LORENZ T 2006 On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming* **106**, 25–57.