Achieving energetic particle confinement in stellarators with precise quasisymmetry

M Landreman\textsuperscript{a}, S Buller\textsuperscript{a}, A Cerfon\textsuperscript{b}, M Drevlak\textsuperscript{c}, A Giuliani\textsuperscript{b}, B Medasani\textsuperscript{d}, E J Paul\textsuperscript{d}, G Stadler\textsuperscript{b}, F Wechsung\textsuperscript{b}, C Zhu\textsuperscript{e}

\textsuperscript{a} U of Maryland, \textsuperscript{b} New York U, \textsuperscript{c} Max Planck Institute for Plasma Physics, \textsuperscript{d} PPPL, \textsuperscript{e} U of Science & Technology of China

Landreman & Paul, PRL (2022), Wechsung et al, PNAS (2022)
Remarkable progress in stellarator confinement in the last year. All configurations scaled to same minor radius and $|B|$. See also Bader et al, Nuclear Fusion (2021).
These new configurations with good alpha confinement use the principle of \textit{quasisymmetry}.

\[ B = B(s, \theta - N \varphi) \]

\[ \Rightarrow \oint (v_d \cdot \nabla s) dt = 0 \]
Goal: $B = B(s, \theta - N \varphi)$

Since 2021

- ML & Paul, Phys Rev Lett (2022)
- Wechsung et al, PNAS (2022)
- Giuliani et al, 1-stage, arXiv (2022)
- Nies & Paul Adjoint method

Near-axis expansion
5% $\beta$, Self-consistent plasma current

Quasi-axisymmetry (QA)
$N = 0$

Quasi-helical symmetry (QH)
$N \neq 0$
• Optimizing stellarator geometry for precise quasisymmetry

• Constructing quasisymmetric geometries using near-axis expansion

• Self-consistent bootstrap current

• Future directions
• Optimizing stellarator geometry for precise quasisymmetry

• Constructing quasisymmetric geometries using near-axis expansion

• Self-consistent bootstrap current

• Future directions
Optimization problem

• 2 stage approach, as for W7-X: First optimize shape of boundary surface, then coils.

• Objective functions:

\[
f_{QS} = \int d^3x \left( \frac{1}{B^3} \left[ (N-1)B \times \nabla B \cdot \nabla \psi - (G + N I) B \cdot \nabla B \right] \right)^2
\]

\[
f_{QH} = (A - A^*)^2 + f_{QS}
\]

\[
f_{QA} = (A - A^*)^2 + \left( \iota^* - \int_0^1 \iota ds \right)^2 + f_{QS}
\]

Boundary aspect ratio

Goal: \( B = B(s, \theta - N \varphi) \).

For quasi-axisymmetry, \( N = 0 \).

For quasi-helical symmetry, \( N \) is the number of field periods, e.g. \( N = 4 \) here.
Optimization problem

- 2 stage approach, as for W7-X: First optimize shape of boundary surface, then coils.
- Objective functions:

\[ f_{QS} = \int d^3 x \left( \frac{1}{B^3} \left[ (N - \iota) \mathbf{B} \times \nabla \mathbf{B} \cdot \nabla \psi - (G + NI) \mathbf{B} \cdot \nabla \mathbf{B} \right] \right)^2 \]

\[ f_{QH} = (A - A^*)^2 + f_{QS} \]

\[ f_{QA} = (A - A^*)^2 + \left( \iota - \int_0^1 \iota ds \right)^2 + f_{QS} \]

- Parameter space: \( R_{m,n} \) & \( Z_{m,n} \) defining a toroidal boundary

\[ R(\theta,\phi) = \sum_{m,n} R_{m,n} \cos(m\theta - n\phi), \quad Z(\theta,\phi) = \sum_{m,n} Z_{m,n} \sin(m\theta - n\phi) \]

- Codes used: SIMSOPT with VMEC
- Cold start: circular cross-section torus
- Vacuum fields at first, allowing precise checks
- Algorithm: default for least-squares in scipy (trust region reflective)
- 6 steps: increasing # of modes varied & VMEC resolution
- Run many optimizations, pick the best

Boundary aspect ratio
Straight $|B|$ contours are possible for quasi-axisymmetry

aspect = 6

$|B|$ on flux surfaces of the quasi-axisymmetric field

Straight $|B|$ contours are possible for quasi-helical symmetry

aspect $= 8$

$|B|$ on flux surfaces of the quasi-helically symmetric field

Good symmetry also exists with magnetic well

\[
\frac{d^2 (\text{flux surface volume})}{d (\text{toroidal flux})^2} < 0 \quad \text{everywhere}
\]

QA

QH

16-coil solutions have been found for the quasi-axisymmetric configurations.

Wechsung et al., PNAS (2022).<br>

Without magnetic well: <R> / 10 between filament centers. Haven’t looked at the QHs yet.
Synergy-breaking modes can be made extremely small

New QA configuration

Fourier amplitudes $|B_{m,n}|$ [Tesla]

$m = 0, n = 0$ (Background)

$m = 0, n = 0$ (Quasiasymmetric)

Geomagnetic field

$n \neq 0$ (Symmetry-breaking)

$s = \text{Normalized toroidal flux}$

Symmetry-breaking $B_{m,n}$ [Tesla]
|B| in Boozer coordinates was verified by independent SPEC calculations

\[
\text{max } |B_{m,n}| \text{ [Tesla] with } n \neq 0
\]

(Ntor = Mpol, Lrad = Mpol + 4)

By Elizabeth Paul
Quasisymmetry works: alpha particle confinement is significantly improved

- All configs scaled to minor radius and $|B|$ of ARIES-CS.
- Fusion alpha birth distribution.
- Same $n(s)$ and $T(s)$ profiles for alpha birth & collisions in each config.
- ANTS code, with collisions.
- Particle considered lost when $s > 1$. 

Fraction of alpha particle energy lost before thermalization
Alpha confinement in quasi-helical stellarators can be better than in a tokamak due to thinner bananas.

\[ 0.00 \quad 0.05 \quad 0.10 \quad 0.15 \quad 0.20 \quad 0.25 \quad 0.30 \]

\[ \sqrt{s \cos(\theta - N\zeta)} \]

\[ \sqrt{\sin^2(\psi) - 1} \]

Width of banana orbit \( \Delta s \propto 1/|t-N| \).

\( N = 0 \) for QA, \( N = \# \) of field periods for QH.
The symmetry also yields extremely low collisional transport for a thermal plasma.

Dotted = with coils

Wechsung et al, PNAS (2022)
• Optimizing stellarator geometry for precise quasisymmetry

• Constructing quasisymmetric geometries using near-axis expansion

• Self-consistent bootstrap current

• Future directions
Expansion about the magnetic axis reduces 3D PDE → 1D ODEs


$r = a_{\text{eff}} \sqrt{s}$

$\frac{r}{\text{radius of curvature of axis}} \ll 1$
The expansion by Garren & Boozer (1991) has been converted into a practical algorithm for generating stellarator shapes

- **Inputs:**
  - Shape of the magnetic axis.
  - 3-5 other numbers (e.g. current on the axis).

- **Outputs:**
  - Shape of the surfaces around the axis.
  - Rotational transform on axis.
  - ...

- **Quasisymmetry guaranteed in a neighborhood of axis.**

- **Can pick any surface to pass to traditional 3D MHD fixed-boundary solve.**
Though quasisymmetry can be guaranteed in a neighborhood of the axis, optimization can greatly increase the volume of good symmetry.

Parameter space: axis shape, few other parameters.

Objective function to minimize:

\[
f = \frac{1}{L} \int d\ell \left| \nabla B \right|^2 + \frac{w_{yy}}{L} \int d\ell \left| \nabla \nabla B \right|^2 + w_L (L - L_*)^2 + w_t (t - t_*)^2
\]

\[
+ \frac{w_{B20}}{L} \int d\ell \left( B_{20} - \frac{1}{L} \int d\ell' B_{20} \right)^2 + w_{\text{well}} \left[ \max \left( 0, \frac{d^2 V}{d\psi^2} - W_* \right) \right]^2
\]

\(w_{yy}, w_L, w_t, w_{B20}, w_{\text{well}}\): Weights chosen by user.
The near-axis equations can be solved so quickly that tensor-product scans over many parameters are feasible.

3.1 × 10^5 optimized stellarators shown (O(r^2))
5.1 × 10^6 optimizations computed
6.0 × 10^{10} equilibria computed
The near-axis equations can be solved so quickly that tensor-product scans over many parameters are feasible.

- $3.1 \times 10^5$ optimized stellarators shown ($O(r^2)$)
- $5.1 \times 10^6$ optimizations computed
- $6.0 \times 10^{10}$ equilibria computed
The near-axis expansion can yield configurations very similar to finite-aspect-ratio optimization, but much faster.

<table>
<thead>
<tr>
<th></th>
<th>Time for 1 objective evaluation:</th>
<th>Total time for optimization (cold start):</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5e-4 CPU-sec</td>
<td>1 CPU-sec</td>
</tr>
</tbody>
</table>

*ML & Paul, PRL (2022)*
In some cases, the near-axis construction can directly generate configurations with excellent confinement.
• Optimizing stellarator geometry for precise quasisymmetry

• Constructing quasisymmetric geometries using near-axis expansion

• Self-consistent bootstrap current

• Future directions
How can bootstrap current be included self-consistently in stellarator optimization?

- Need self-consistency between MHD equilibrium and drift-kinetic equation.
- Previous method: fixed-point iteration, only after an optimization.
- Accurate drift-kinetic bootstrap calculations in stellarators are computationally expensive. Preferably not in the optimization loop.

VMEC: given $I_0(s)$, determine $B_0$.
SFINCS: given $B_0$, determine $I_1(s)$.
VMEC: given $I_1(s)$, determine $B_1$.
SFINCS: given $B_1$, determine $I_2(s)$.

\[
\langle \mathbf{J} \cdot \mathbf{B} \rangle \ [\text{MA T}] \text{ from VMEC}
\]

- \(\langle \mathbf{J} \cdot \mathbf{B} \rangle \) vs. normalized toroidal flux $s$

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>pink</td>
</tr>
<tr>
<td>1</td>
<td>blue</td>
</tr>
<tr>
<td>2</td>
<td>green</td>
</tr>
<tr>
<td>3</td>
<td>yellow</td>
</tr>
<tr>
<td>4</td>
<td>dotted</td>
</tr>
</tbody>
</table>
New idea: exploit quasisymmetry & use analytic expressions for tokamaks


\[ t \to t - N \]

Bootstrap current (& other quantities) in quasisymmetry are the same as in axisymmetry, up to some substitutions:

Should be accurate for the new precisely quasisymmetric configurations.

A new set of analytical formulae for the computation of the bootstrap current and the neoclassical conductivity in tokamaks

Cite as: Phys. Plasmas 28, 022502 (2021); doi: 10.1063/5.0012664
Submitted: 6 May 2020 · Accepted: 11 December 2020 ·
Published Online: 2 February 2021

A. Redl,1,2,a) C. Angioni,1 E. Belli,3 O. Sauter,4 ASDEX Upgrade Teamb) and EUROfusion MST1 Teamc)
Before doing new optimizations: Redl formula is accurate in previous QA & QH stellarators

\[ n_e = (1 - s^5) \times 10^{20} \text{ m}^{-3}, \quad T_e = T_i = (1 - s) \times 12 \text{ keV} \]

---

\[ \langle B \rangle \text{ [MegaAmpere Tesla / meter]} \]

Bootstrap current in quasi-axisymmetry

\[ \langle \mathbf{B} \rangle \text{ [MegaAmpere Tesla / meter]} \]

Bootstrap current in quasi-helical symmetry

\[ s = (1 - s^5) \times 10^{20} \text{ m}^{-3}, \quad T_e = T_i = (1 - s) \times 12 \text{ keV} \]

(Not self-consistent yet)
Optimization recipe

- Objective function: \[ f = f_{QS} + f_{bootstrap} + (A-6.5)^2 + (a-a_{ARIES-CS})^2 + \left( \langle B \rangle - \langle B \rangle_{ARIES-CS} \right)^2 \]

\[ f_{QS} = \int d^3x \left( \frac{1}{B^3} \left[ (N-I) \mathbf{B} \times \nabla \mathbf{B} \cdot \nabla \psi - (G+NI) \mathbf{B} \cdot \nabla \mathbf{B} \right] \right)^2 \]

\[ f_{bootstrap} = \frac{\int_0^1 ds \left[ \langle \mathbf{j} \cdot \mathbf{B} \rangle_{vmec} - \langle \mathbf{j} \cdot \mathbf{B} \rangle_{Redl} \right]^2}{\int_0^1 ds \left[ \langle \mathbf{j} \cdot \mathbf{B} \rangle_{vmec} + \langle \mathbf{j} \cdot \mathbf{B} \rangle_{Redl} \right]^2} \]

- Parameter space: \( \{R_{m,n}, Z_{m,n}, \text{toroidal flux, current spline values}\} \)
  or \( \{R_{m,n}, Z_{m,n}, \text{toroidal flux, iota spline values}\} \)

- Cold start

- Algorithm: default for least-squares in scipy (trust region reflective)

- Steps: increasing # of modes varied: m and \(|n/nfp|\) up to j in step j
Example of optimization with self-consistent bootstrap current

\[ n_{e0} = 2.5 \times 10^{20} / \text{meters}^3 \]

\[ T_{e0} = T_{i0} = 10 \text{ keV} \]

\[ \beta = 2.8\%, \quad I_p = 1.3 \text{ MA} \]
To reach reactor-relevant 5% beta in QH without crossing $\iota=1$, a constraint on $\iota$ can be included.

Crossing $\iota=1$, the worst resonance, is probably unacceptable.

$n_e = 3 \times 10^{20}/\text{meters}^3, T_e = T_i = 15 \text{ keV}$
To reach reactor-relevant 5% beta in QH without crossing $\iota=1$, a constraint on $\iota$ can be included. Crossing $\iota=1$, the worst resonance, is probably unacceptable.

Solution: Add barrier term in objective

$$f_+ = \int_0^1 ds \left[ \min\left(\iota(s) - 1.03, 0\right) \right]^2$$

Quasisymmetry & bootstrap consistency remain good:
If you want perfectly self-consistent current, you can do a few fixed-point iterations at the end.

\[ \langle \beta \rangle = 5\%, \quad \varepsilon_{\text{eff}}^{3/2} < 6 \times 10^{-5} \]

\( \alpha \)-particle losses < 0.3%

No significant degradation in quasisymmetry:

Optimization with Redl current

After SFINCS fixed-point iterations
The optimization with self-consistent bootstrap current also works for quasi-axisymmetry.

Symmetry is not as good as for vacuum, but sufficient for excellent confinement.

Possible islands where $\epsilon = 2/3$, $\epsilon = 4/7 = 0.57$?

$\langle \beta \rangle = 3\%$, $\epsilon_{\text{eff}}^{3/2} < 7 \times 10^{-6}$

$\alpha$-particle losses < 1%
• Optimizing stellarator geometry for precise quasisymmetry

• Constructing quasisymmetric geometries using near-axis expansion

• Self-consistent bootstrap current

• Future directions
Future directions

- For the high $\beta$ configurations, check surface quality, & eliminate any islands.
- Coils & MHD stability for the high $\beta$ configurations.
- Check robustness to uncertainty in the pressure profile.
- Similar recipes for quasi-poloidal symmetry or quasi-isodynamic?

Before optimization

After optimization

ML, Medasani & Zhu (2021), Baillod et al (2022)
It is now possible to design stellarators with alpha confinement close to or better than a tokamak.
Extra slides
Advantages of stellarators: steady-state, no disruptions, no Greenwald density limit, no power recirculated for current drive.

But, alpha particle losses & collisional transport would be too large unless you carefully choose the geometry.
**Simopt documentation**

*simopt* is a framework for optimizing *stellarators*. The high-level routines are in python, with calls to C++ or fortran where needed for performance. Several types of components are included:

- Interfaces to physics codes, e.g. for MHD equilibrium.
- Tools for defining objective functions and parameter spaces for optimization.
- Geometric objects that are important for stellarators – surfaces and curves – with several available parameterizations.
- Efficient implementations of the Biot-Savart law and other magnetic field representations, including derivatives.
- Tools for parallelized finite-difference gradient calculations.

- Handles both stage 1 (plasma shape) and stage 2 (coil shapes)
- 100% open source
- Both derivative-free and derivative-based problems
- Try out new objective functions or new surface/curve representations without touching any working code.

*ML, B Medasani, F Wechsung, A Giuliani, R Jorge, & C Zhu, J. Open Source Software 6, 3525 (2021).*
Why do the configurations with best quasisymmetry not have the best trajectory confinement?

Lost trajectories in the new QA look like this:

Width of banana orbit \[ \Delta s \approx \frac{mvR\sqrt{2r\eta}}{(t-N)\psi_{edge}Ze} \approx \frac{1}{t-N} \]

For fixed minor radius, \[ \frac{\Delta s_{QA}}{\Delta s_{QH}} \sim 4 \]
2 types of quasisymmetry

Quasi-axisymmetry (QA): $B = B(r, \theta)$

Quasi-helical symmetry (QH): $B = B(r, \theta - N\phi)$

General stellarator (not symmetric)

Contours of $B = |B|$: $B_{min}$ to $B_{max}$
Is there an optimization recipe that can give consistently straight $|B|$ contours?

We want $B = B(r, \theta - N \varphi)$

(a) Zarnstorff et al (2001)  
(b) Najambadi et al (2008)  
(c) Garabedian (2008)  
(d) Liu et al (2018)  
(e) Henneberg et al (2019)  
(f) Nuhrenberg & Zille (1988)  
(g) Anderson et al (1995)  
(h) Bader et al (2020)
The new configurations have small magnetic shear.

New QH

New QH+well

New QA

New QA+well

Rotational transform $\tau$

Normalized toroidal flux $s$

VMEC

SPEC

New QA

New QH
Self-consistent bootstrap current profiles have previously been computed by fixed-point iteration between VMEC and a bootstrap current code.

Available codes: DKES/NTSS, SFINCS, + others for tokamaks.

VMEC: given $I_0(s)$, determine $B_0$.
SFINCS: given $B_0$, determine $I_1(s)$.
VMEC: given $I_1(s)$, determine $B_1$.
SFINCS: given $B_1$, determine $I_2(s)$.

SFINCS: >20 node-seconds per surface for reactor n/T, cost much higher at low collisionality, uses PETSc, tricky to set resolution parameters.
New idea: exploit quasisymmetry & use analytic expressions for tokamaks

Redl (2021)

ASDEX Upgrade #33173, time = 4.75 sec

Geometry enters through

\[ f_i = 1 - f_c = 1 - \frac{3}{4} \left( \frac{\langle B^2 \rangle}{4} \right) \int_0^{1/B_{\text{max}}} \frac{\lambda d\lambda}{\sqrt{1 - \lambda B}}. \]

\[ \nu_\ast = 6.921 \cdot 10^{-18} \frac{q R n_e Z \ln \Lambda_e}{T_e} \varepsilon^{1/2}, \]

\[ \nu_\ast = 4.90 \cdot 10^{-18} \frac{q R n_i Z^4 \ln \Lambda_{ii}}{T_i} \varepsilon^{3/2}. \]
Decent 16-coil solutions have been found for the new QAs by Florian Wechsung @ NYU. <R>/10 between filament centers. Haven’t looked at the QHs yet.
The symmetry yields extremely good confinement of collisionless trajectories.

All configurations scaled to ARIES-CS minor radius (1.7 m) and $|B|$ (5.7 T).

5000 alpha particles initialized isotropically at $s=0.3$.

*SIMPLE code: Albert et al, JCP (2020).*
Simsopt documentation

Simsopt is a framework for optimizing stellarators. The high-level routines are in python, with calls to C++ or fortran where needed for performance. Several types of components are included:

- Interfaces to physics codes, e.g. for MHD equilibrium.
- Tools for defining objective functions and parameter spaces for optimization.
- Geometric objects that are important for stellarators – surfaces and curves – with several available parameterizations.
- Efficient implementations of the Biot-Savart law and other magnetic field representations, including derivatives.
- Tools for parallelized finite-difference gradient calculations.

The design of Simsopt is guided by several principles:

- Thorough unit testing, regression testing, and continuous integration.
- Extensibility. It should be possible to add new codes and terms to the objective function without editing modules that already work, i.e. the open-closed principle. This is because any edits to working code can potentially introduce bugs.
- Modularity: Physics modules that are not needed for your optimization problem do not need to be installed. For instance, to optimize SPEC equilibria, the VMEC module need not be installed.
- Flexibility: The components used to define an objective function can be re-used for applications other than standard optimization. For instance, a Simsopt objective function is a standard python function that can be plotted, passed to optimization packages outside of Simsopt, etc.

Simsopt is fully open-source, and anyone is welcome to use it, make suggestions, and contribute.
Previous quasisymmetric configurations \((s=0.5)\)
Previous quasisymmetric configurations (s=1)
$|B|$ along a field line for new QH

$s=0.01$

$s=0.02$

$s=0.04$

$s=0.25$

$s=0.50$

$s=1.00$
$|B|$ along a field line for new QA with magnetic well
SPEC confirms the new QA/QH configurations have good surfaces
Good flux surface exist with coils

New QA

New QA+well
• We’d like to minimize islands/chaos if they exist.

• But, many stellarator codes and objective functions assume nested surfaces, & build on the VMEC 3D MHD equilibrium code [1].

• Idea:
  – Compute two $B$ representations at each iteration: one assuming surfaces (VMEC) and one not (SPEC [2]).
  – Include both island width (from SPEC) and surface-based quantities (from VMEC) in the objective function.
  – Measure island width using Greene’s residue [3,4]

Example: Start with a configuration that has islands

$n_{\text{fpl}} = 2$, decent quasi-axisymetry (QA), aspect $= 6$, $\beta = 0$, island chain at $\iota = 2/5 = 0.4$
mpi = MpiPartition()
vmec = Vmec("input.nfp2_QA", mpi)
surf = vmec.boundary

spec = Spec("nfp2_QA.sp", mpi)
spec.boundary = surf

# Define parameter space:
surf.fix_all()
surf.fixed_range(mmin=0, mmax=3,
                   nmin=-3, nmax=3, fixed=False)
surf.fix("rc(0,0)") # Major radius

# Configure quasisymmetry objective:
qs = Quasisymmetry(Boozer(vmec),
                   0.5, # Radius s to target
                   1, 0) # (M, N) you want in |B|

# Specify resonant iota = p / q
p = -2; q = 5
residue1 = Residue(spec, p, q)
residue2 = Residue(spec, p, q, theta=np.pi)

# Define objective function
prob = LeastSquaresProblem([(
                            vmec.aspect, 6, 1),
                            (vmec.iota_axis, 0.39, 1),
                            (vmec.iota_edge, 0.42, 1),
                            (qs, 0, 2),
                            (residue1, 0, 2),
                            (residue2, 0, 2)])

least_squares_mpi_solve(prob, mpi, grad=True)

Simsopt driver script applied:

SPEC told to use the same boundary surface object as VMEC.

Objective function includes both quasisymmetry from VMEC and residues from SPEC.
The optimization eliminates the islands.
Quasisymmetry is simultaneously improved during the optimization.
Expansion about the magnetic axis reduces 3D PDE -> 1D ODEs

High aspect ratio tori

Low aspect ratio tori

\[
\frac{r}{\text{radius of curvature of axis}} \ll 1
\]

Aspect ratio = \(\frac{R}{r}\)