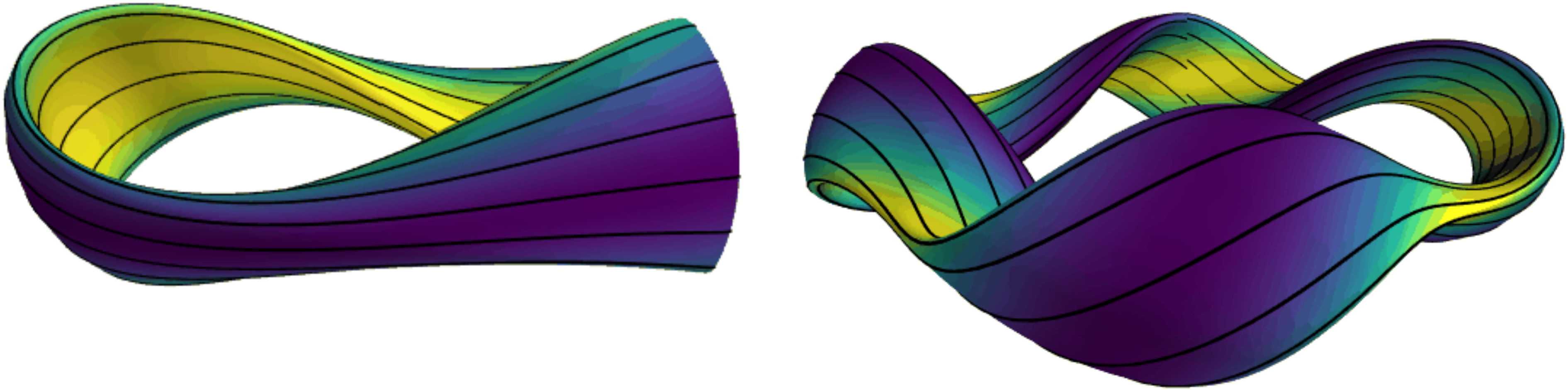
Innovations in stellarator optimization for quasisymmetry

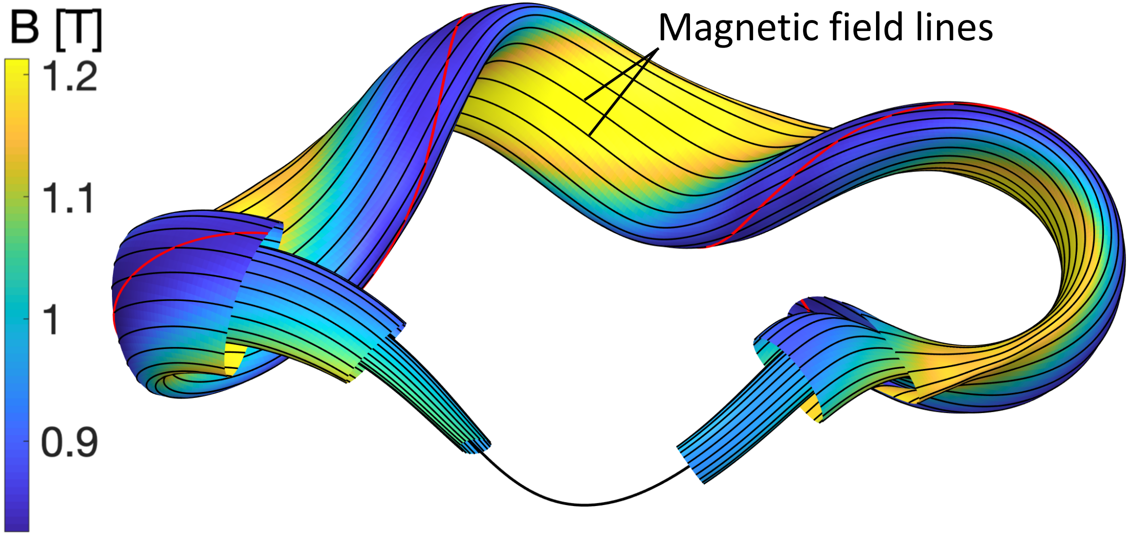


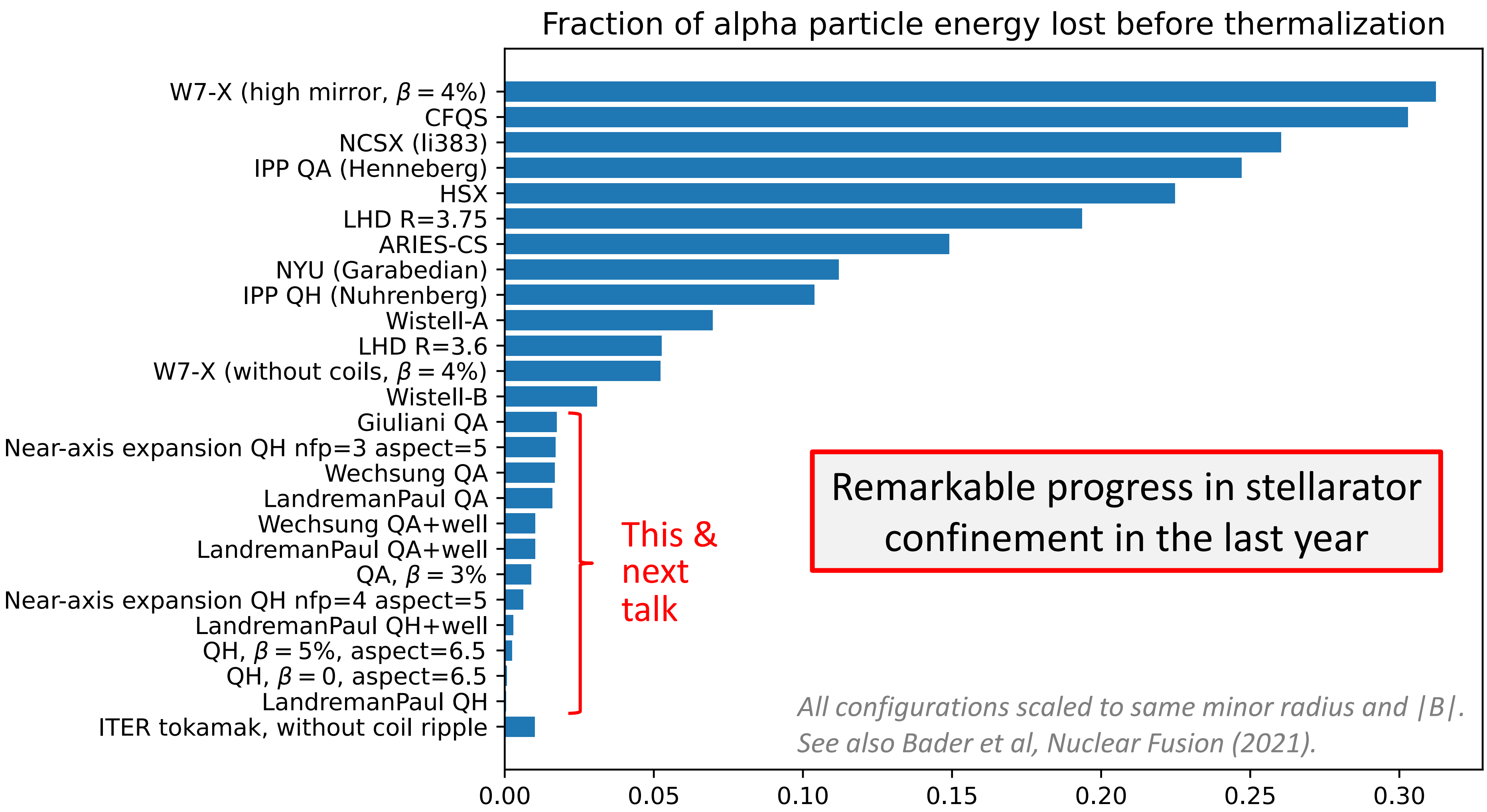
M. Landreman[a], A. Cerfon[b], M. Drevlak[c], A. Giuliani[b], R. Jorge[d], B. Medasani[e], E. J. Paul[e], F. Wechsung[b], C. Zhu[e]

[a] University of Maryland, [b] New York University, [c] Max Planck Institute for Plasma Physics, [d] Téchnico Lisboa, [e] Princeton Plasma Physics Laboratory

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.

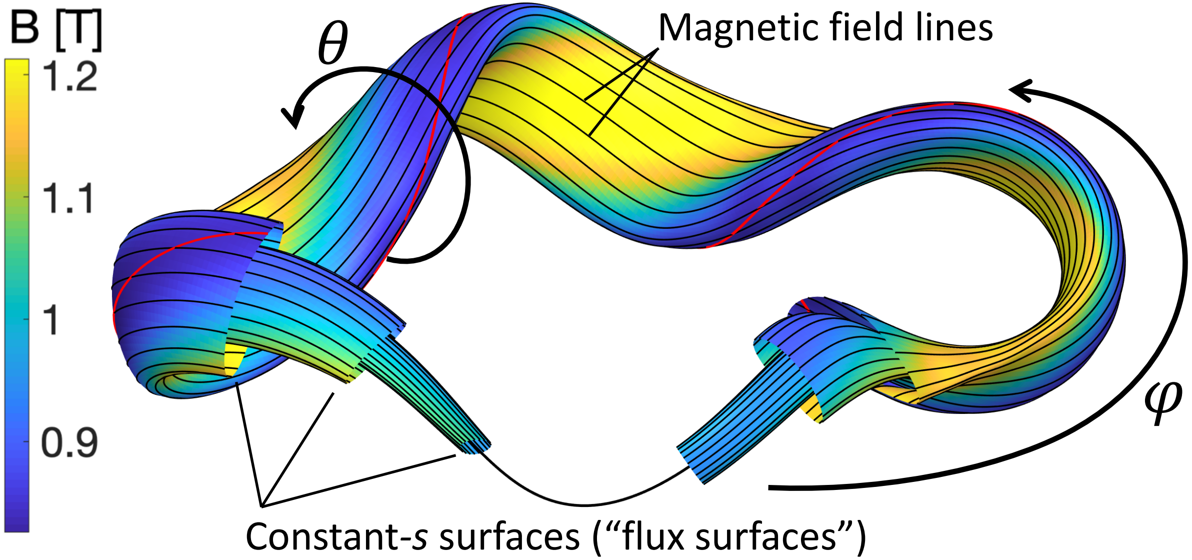



Fraction of alpha particle energy lost before thermalization
W7-X (high mirror, $\beta = 4\%$)
CFQS
NCSX (li383)
IPP QA (Henneberg)
HSX
LHD R=3.75
ARIES-CS
NYU (Garabedian)
IPP QH (Nuhrenberg)
Wistell-A
LHD R=3.6
W7-X (without coils, $\beta = 4\%$)
Wistell-B
Giuliani QA
Near-axis expansion QH nfp=3 aspect=5
Wechsung QA
LandremanPaul QA
Wechsung QA+well
LandremanPaul QA+well
QA, $\beta = 3\%$
Near-axis expansion QH nfp=4 aspect=5
LandremanPaul QH+well
QH, $\beta = 5\%$, aspect=6.5
QH, $\beta = 0$, aspect=6.5
LandremanPaul QH
ITER tokamak, without coil ripple
This & next talk
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).
0.00
0.05
0.10
0.15
0.20
0.25
0.30

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.

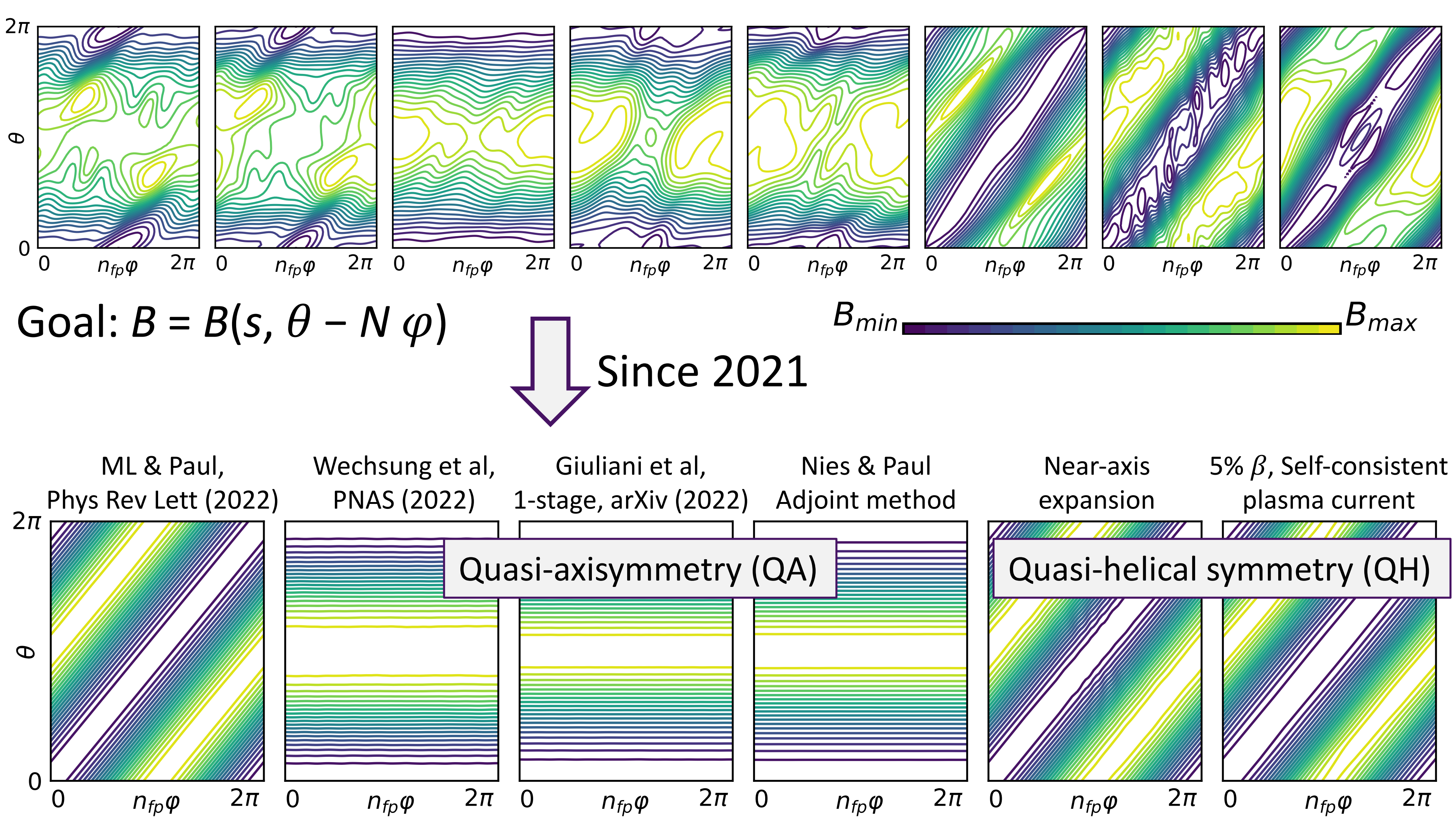
A solution: quasisymmetry



$B = B(s, \theta - N\,\varphi)$

Boozer angles

$$\Rightarrow \quad \oint \left(\mathbf{v}_d \cdot \nabla s \right) dt = 0$$



Goal: $B = B(s, \theta - N\varphi)$
Since 2021
B_{min}
B_{max}
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% β, Self-consistent plasma current
Quasi-axisymmetry (QA)
Quasi-helical symmetry (QH)
θ
$n_{fp}\varphi$
0
2π

- Optimizing stellarator geometry for precise quasisymmetry
- Constructing quasisymmetric geometries using near-axis expansion
- Self-consistent current in the plasma
- Future directions

- Optimizing stellarator geometry for precise quasisymmetry
- Constructing quasisymmetric geometries using near-axis expansion
- Self-consistent current in the plasma
- Future directions

Optimization problem

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

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

$$f_{QH} = \left(A - A_* \right)^2 + f_{QS} \qquad\qquad f_{QA} = \left(A - A_* \right)^2 + \left(\iota_* - \int_0^1 \iota\, ds \right)^2 + f_{QS}$$

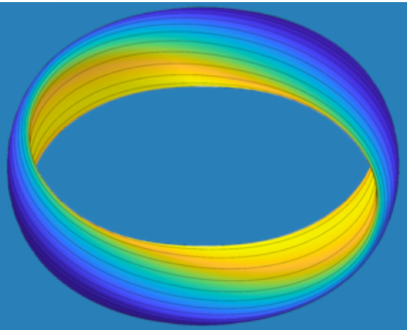
Boundary aspect ratio

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

$$R(\theta,\phi) = \sum_{m,n} R_{m,n} \cos(m\theta - n\phi), \qquad Z(\theta,\phi) = \sum_{m,n} Z_{m,n} \sin(m\theta - n\phi)$$

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

simsopt.readthedocs.io/en/latest/



latest

Search docs

CONTENTS

EXAMPLES

Simsopt documentation

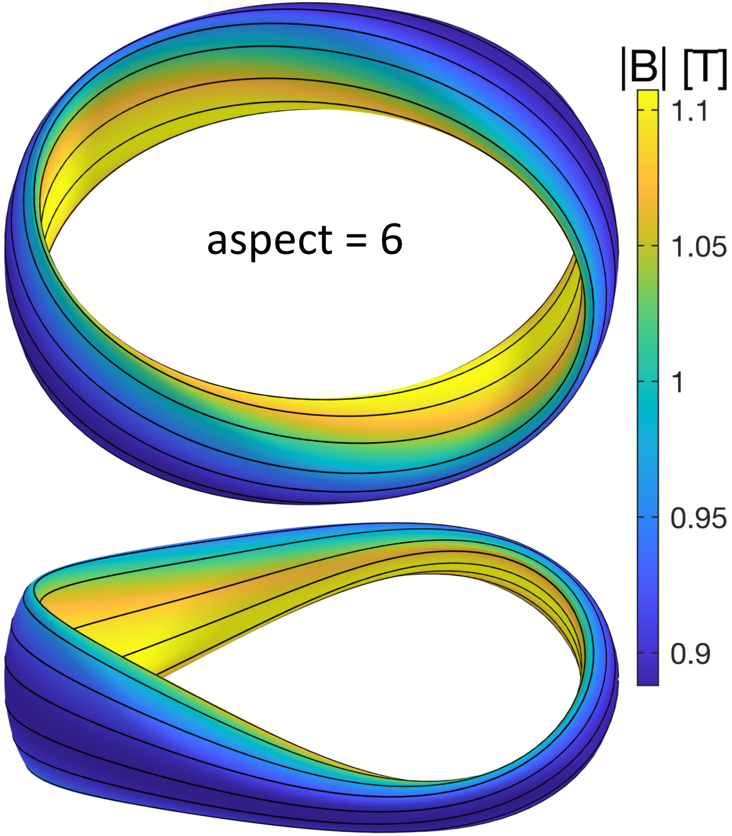
`simsopt` is a framework for optimizing stellarators. The high-level routines are in python, with calls to C++ or fortran where needed for performance. Several types of components are included:

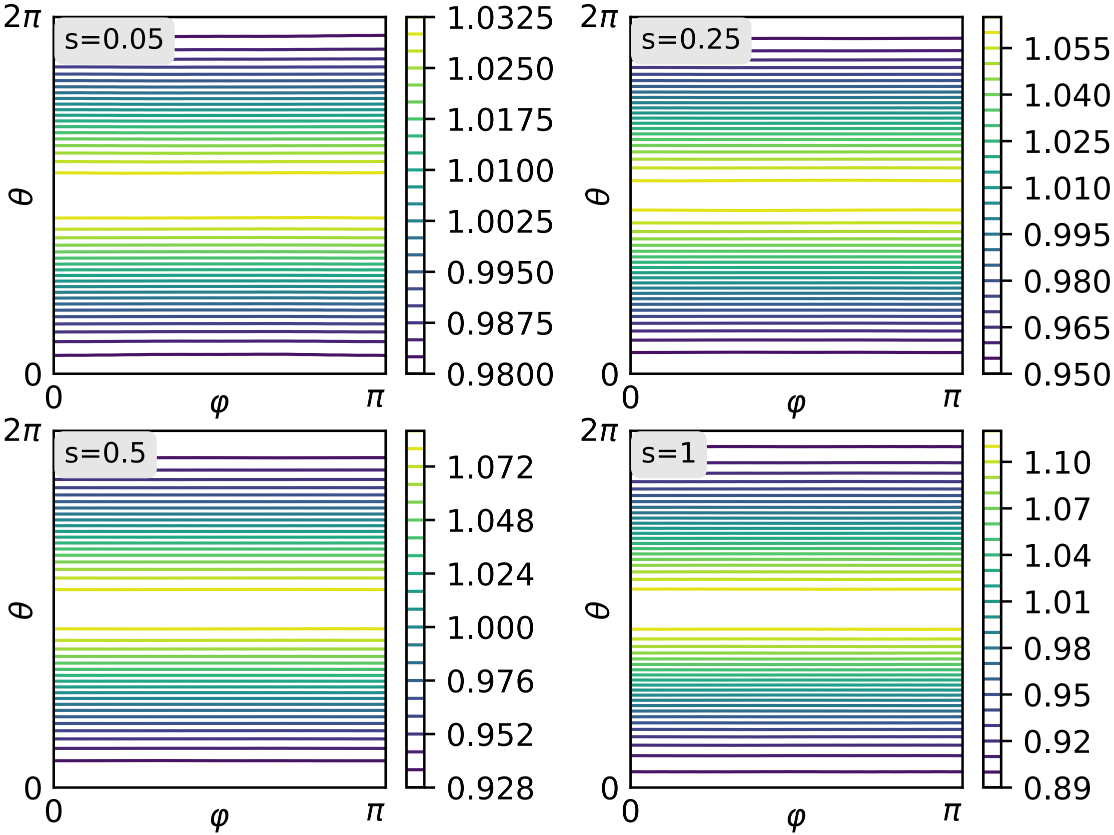
- Interfaces to physics codes, e.g. for MHD equilibrium.
- Tools for defining objective functions and parameter spaces for optimization.
- Geometric objects that are important for stellarators – surfaces and curves – with several available parameterizations.
- Efficient implementations of the Biot-Savart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

- 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).

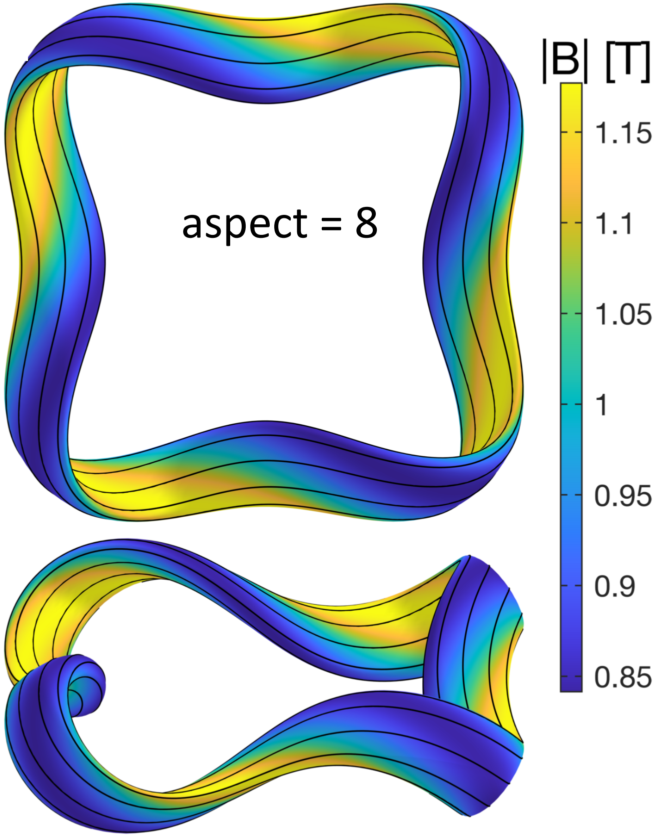
Straight |B| contours are possible for quasi-axisymmetry



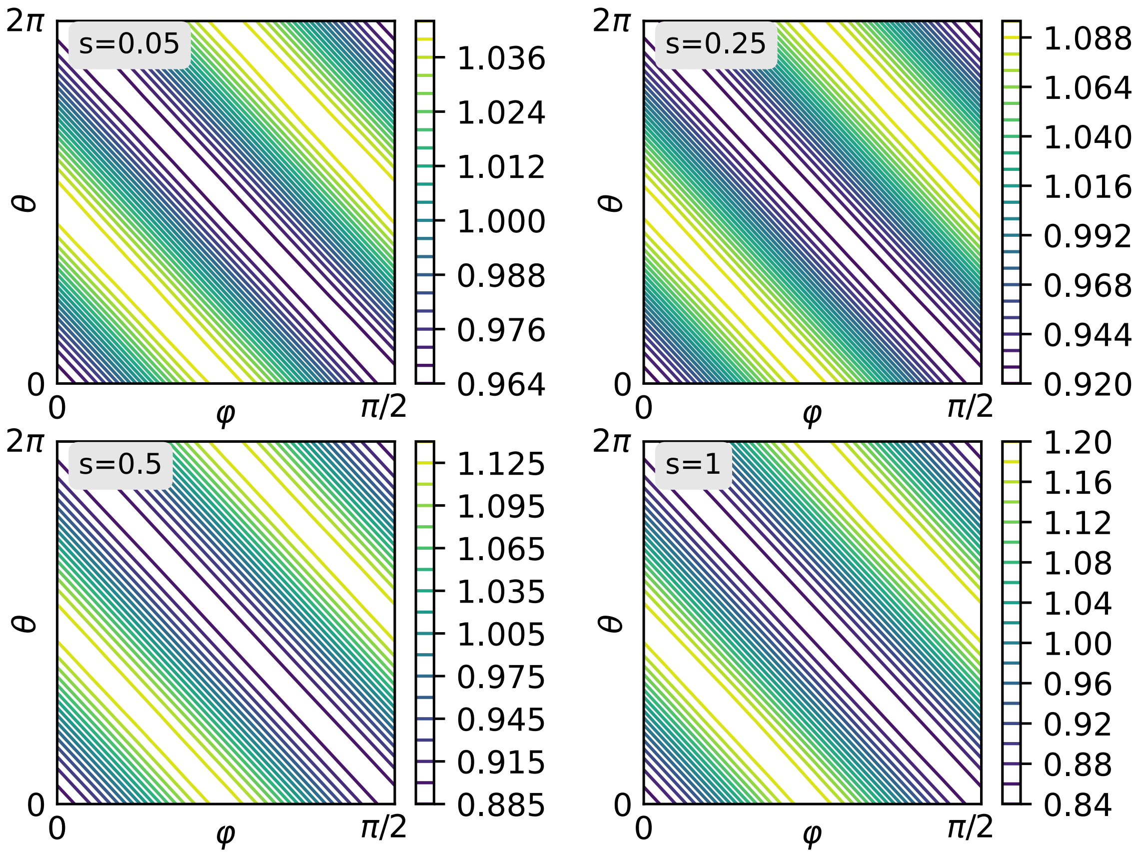


ML & Paul, PRL (2022). All input/output files and optimization scripts online at doi.org/10.5281/zenodo.5645412

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



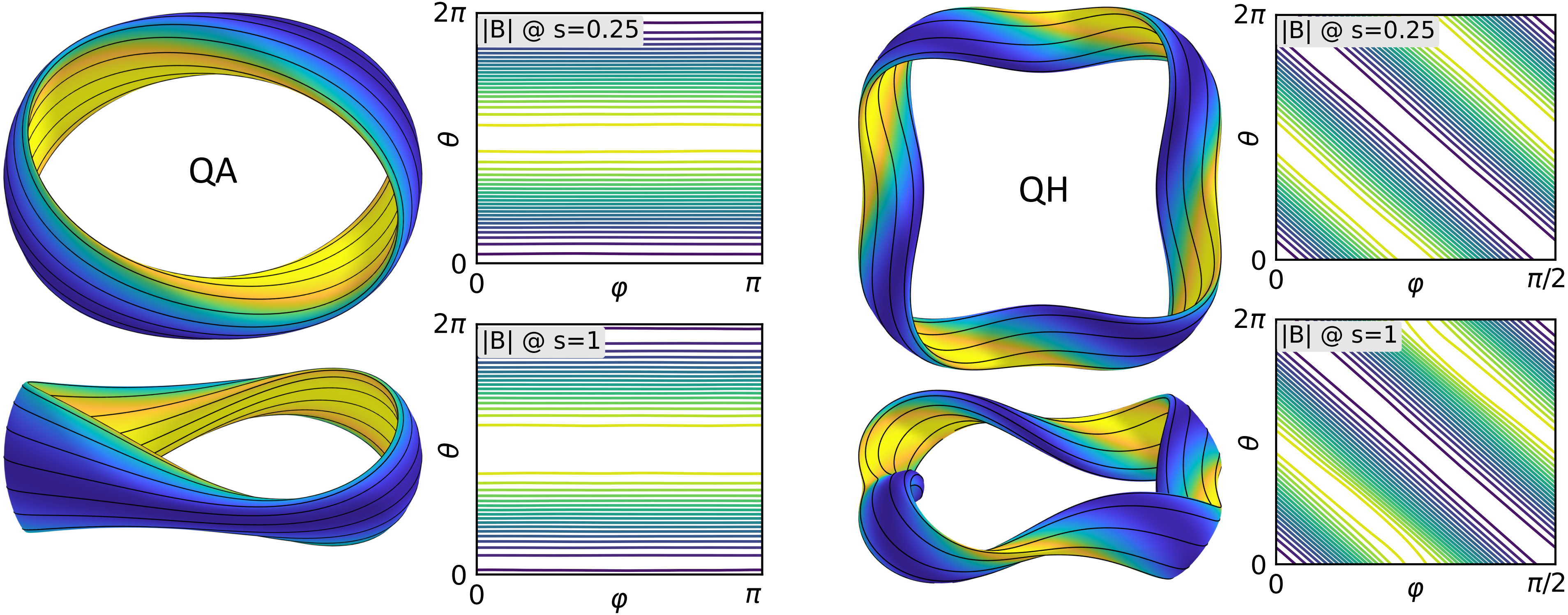
|B| on flux surfaces of the quasi-helically symmetric field



ML & Paul, PRL (2022). All input/output files and optimization scripts online at doi.org/10.5281/zenodo.5645412

Good symmetry also exists with magnetic well

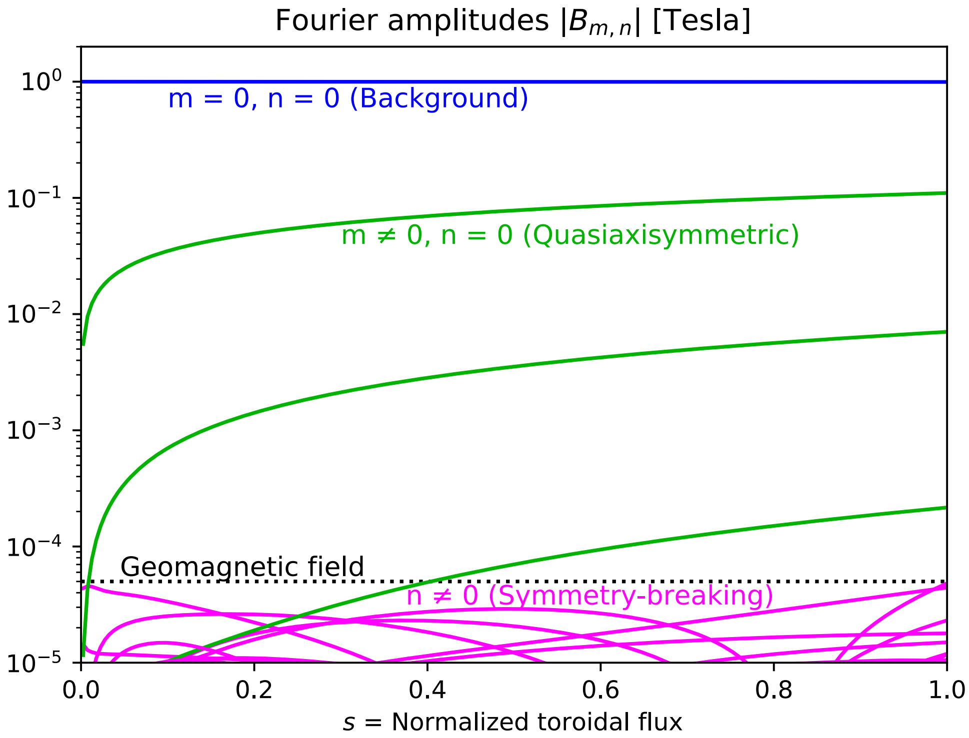
$$\frac{d^2\left(\text{flux surface volume}\right)}{d\left(\text{toroidal flux}\right)^2} < 0 \;\; \text{everywhere}$$

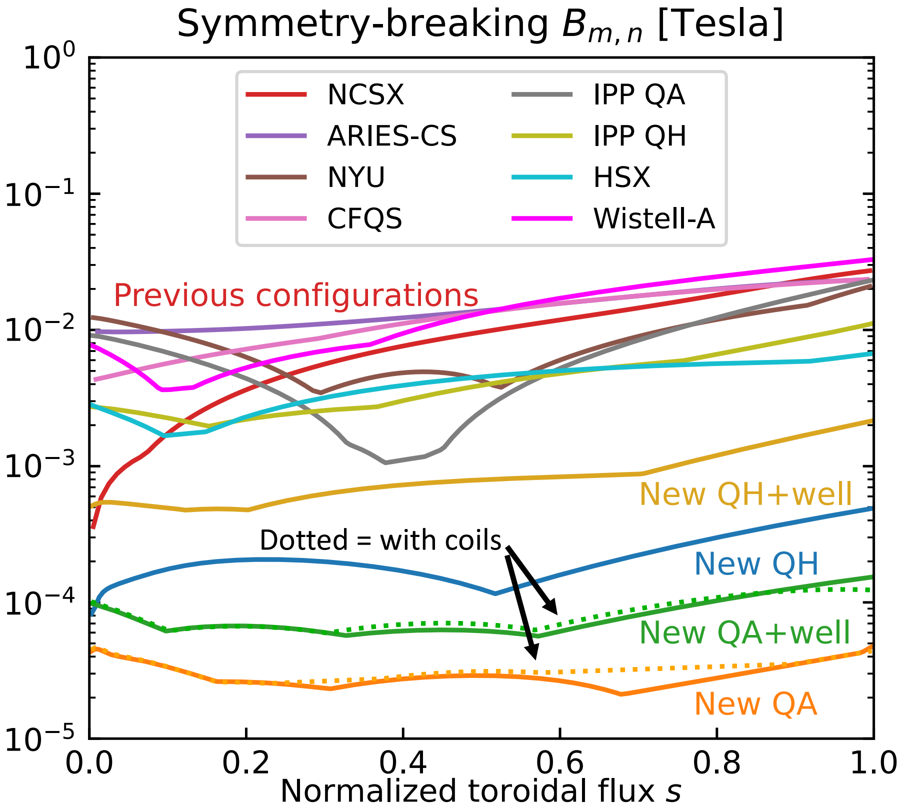


ML & Paul, PRL (2022). All input/output files and optimization scripts online at doi.org/10.5281/zenodo.5645412

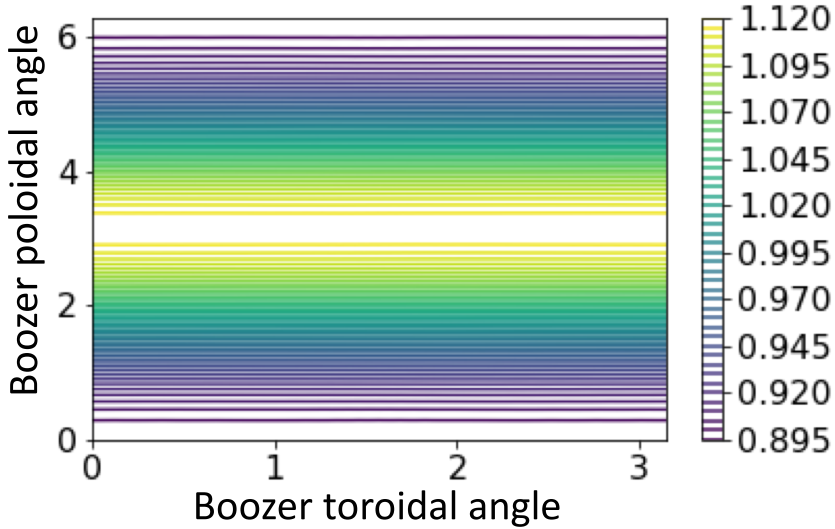
Symmetry-breaking modes can be made extremely small

New QA configuration

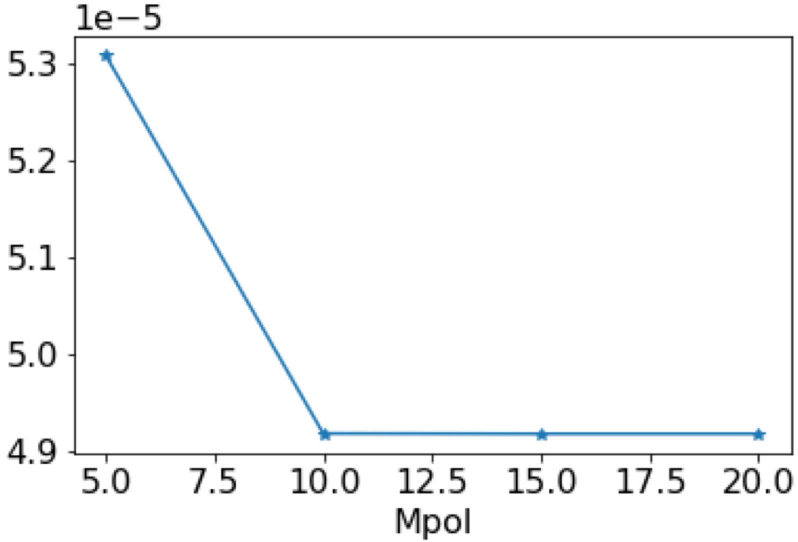




|B| in Boozer coordinates was verified by independent SPEC calculations



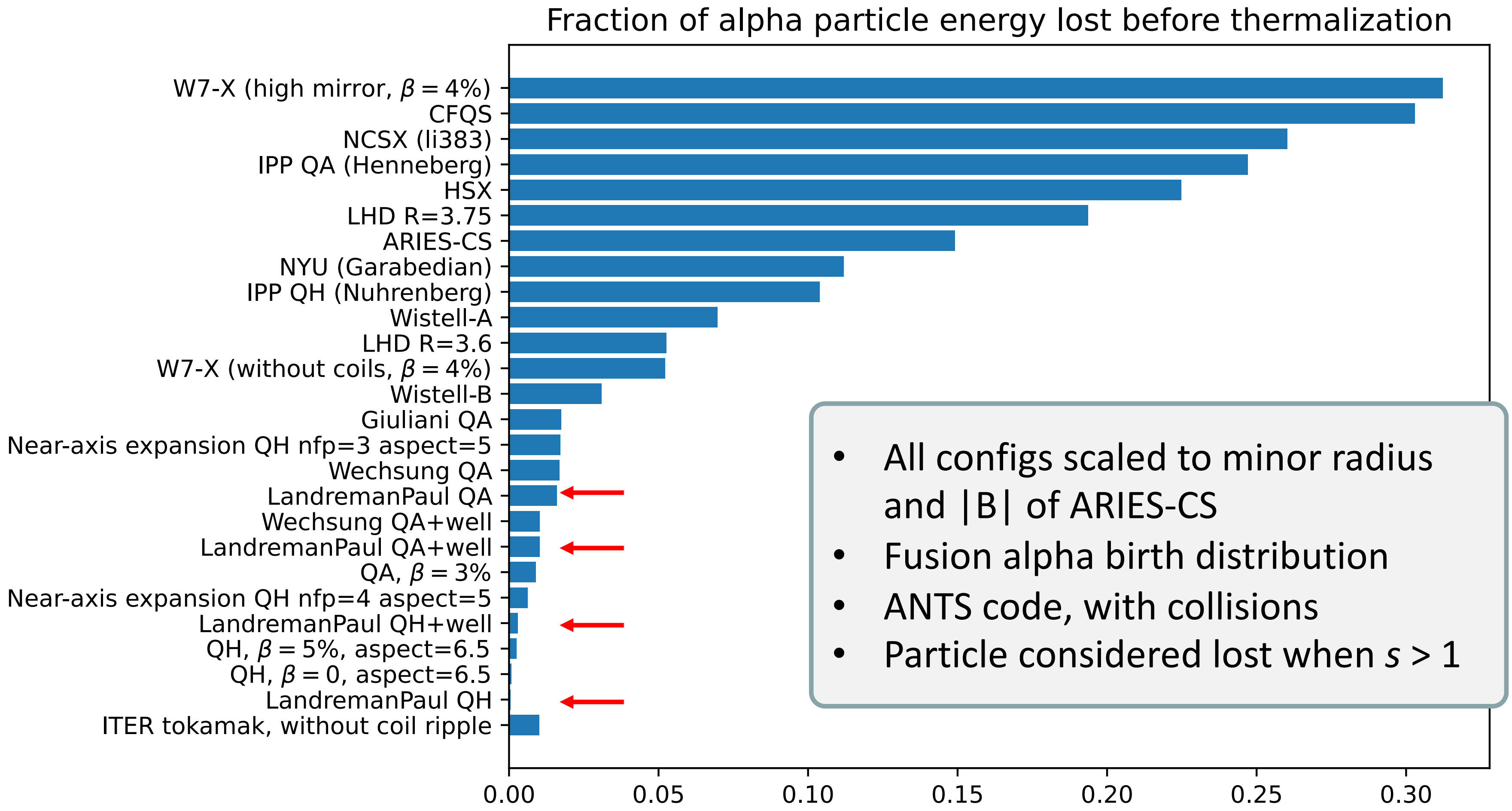
max $|B_{m,n}|$ [Tesla] with n ≠ 0



(Ntor = Mpol, Lrad = Mpol + 4)

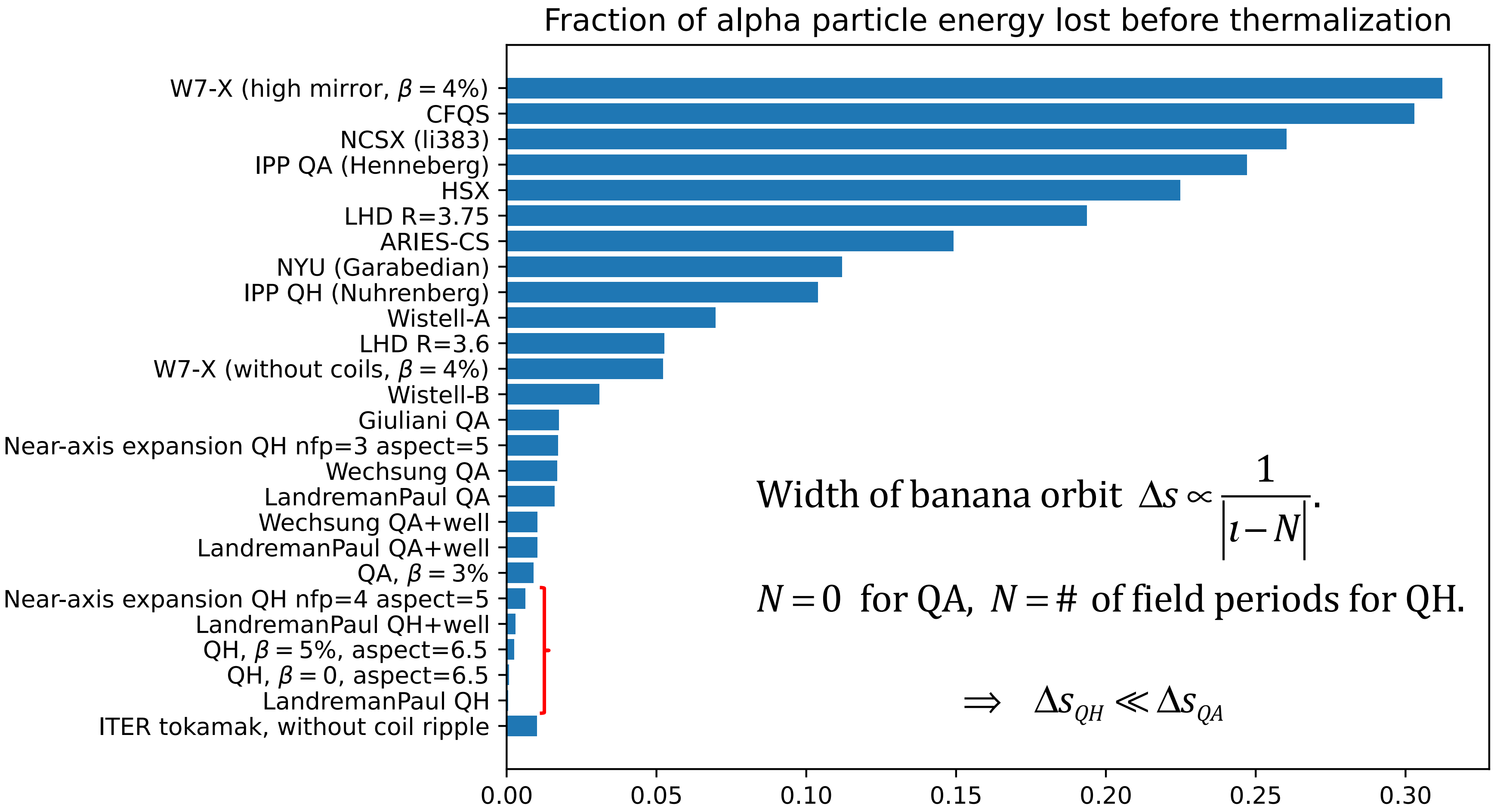
By Elizabeth Paul

Quasisymmetry works: alpha particle confinement is significantly improved



- All configs scaled to minor radius and |B| of ARIES-CS
- Fusion alpha birth distribution
- ANTS code, with collisions
- Particle considered lost when $s > 1$

Alpha confinement in quasi-helical stellarators can be better than in a tokamak

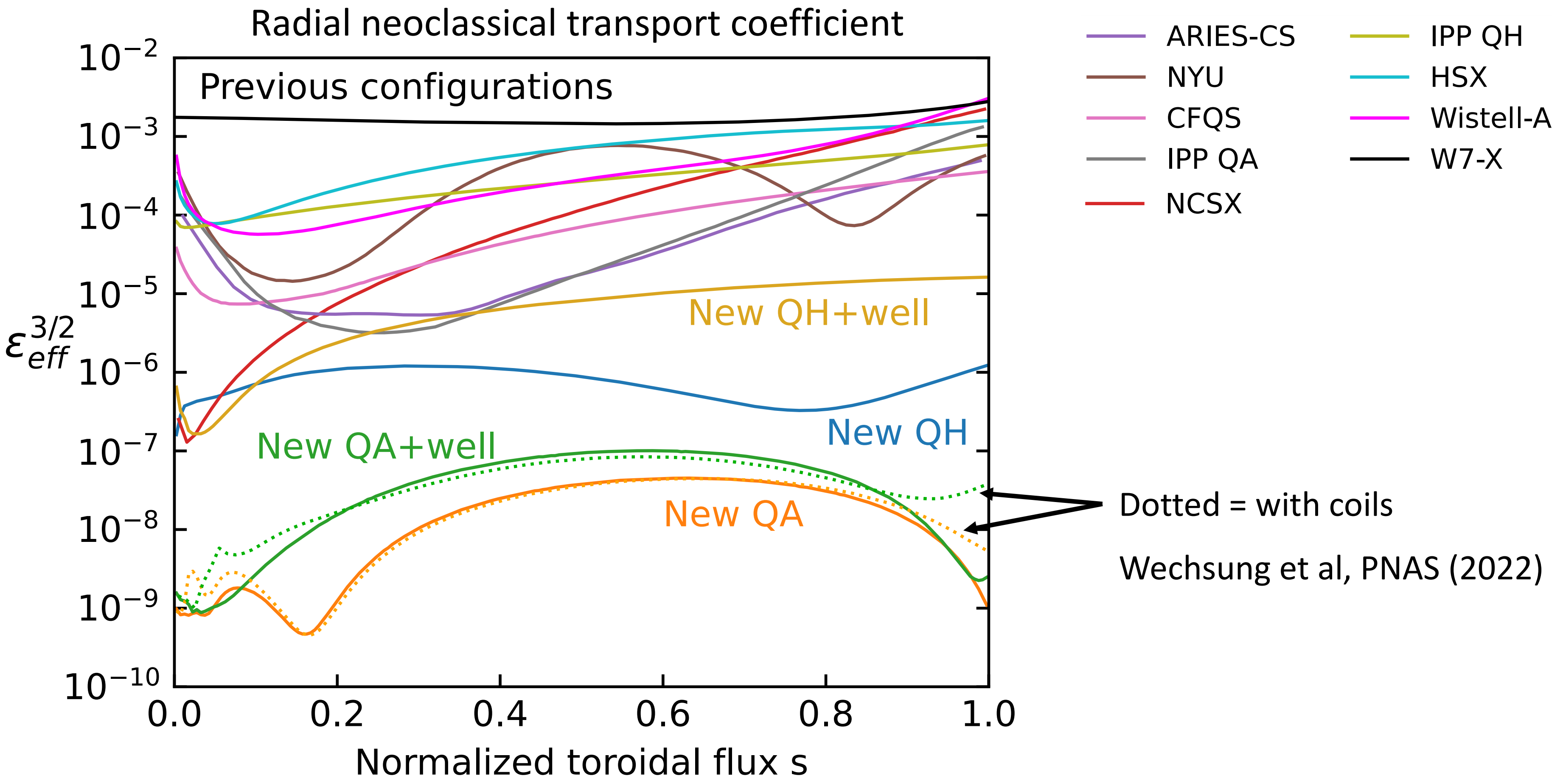


Width of banana orbit $\Delta s \propto \dfrac{1}{|\iota - N|}$.

$N = 0$ for QA, $N = \#$ of field periods for QH.

$\Rightarrow \quad \Delta s_{QH} \ll \Delta s_{QA}$

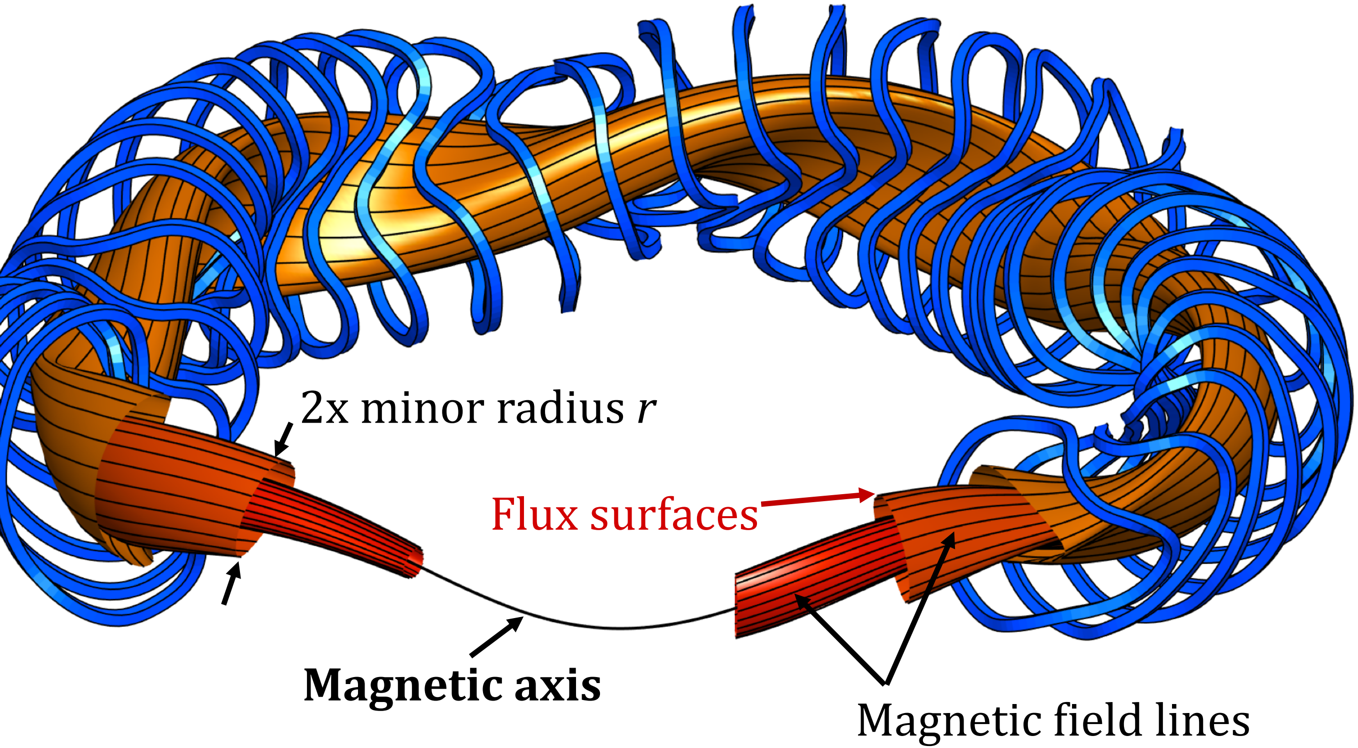
The symmetry also yields extremely low collisional transport for a thermal plasma



Wechsung et al, PNAS (2022)

- Optimizing stellarator geometry for precise quasisymmetry
- Constructing quasisymmetric geometries using near-axis expansion
- Self-consistent current in the plasma
- Future directions

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



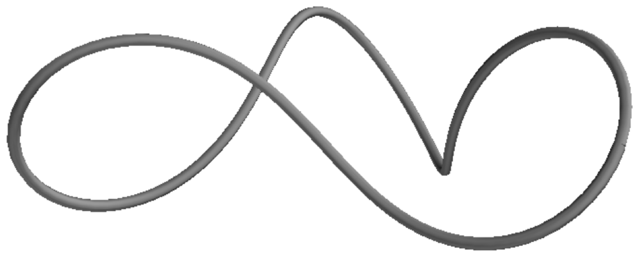
$$r = a_{eff}\sqrt{s}$$

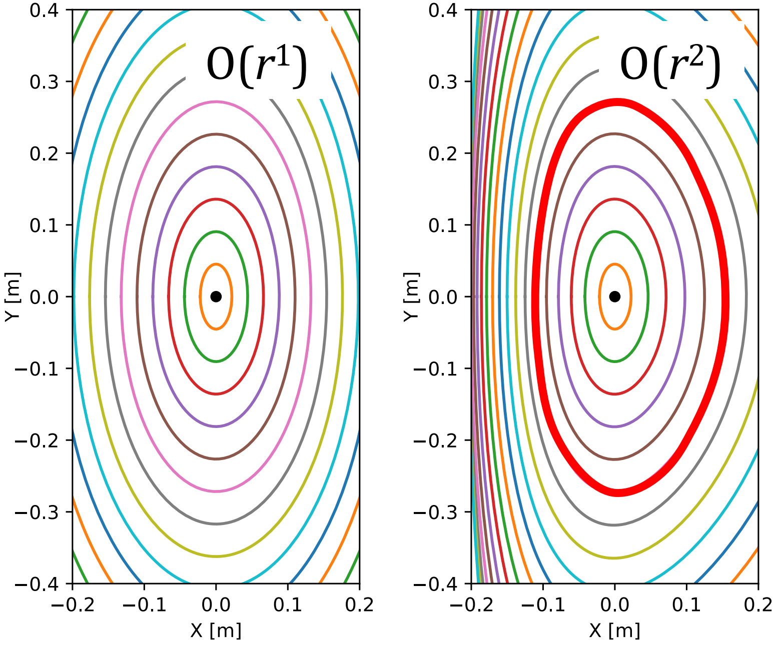
$$\frac{r}{\text{radius of curvature of axis}} \ll 1$$

Garren & Boozer (1991),
ML & Sengupta (2019)

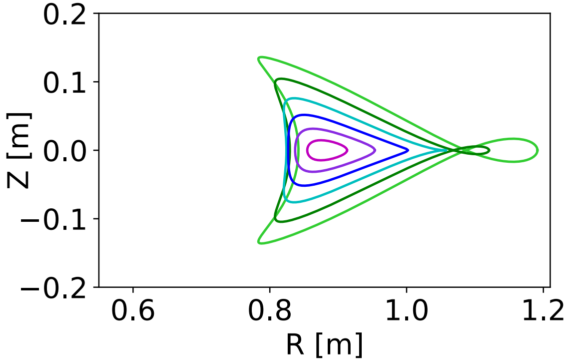
The expansion by Garren & Boozer (1991) has been converted into a practical algorithm for generating stellarator shapes

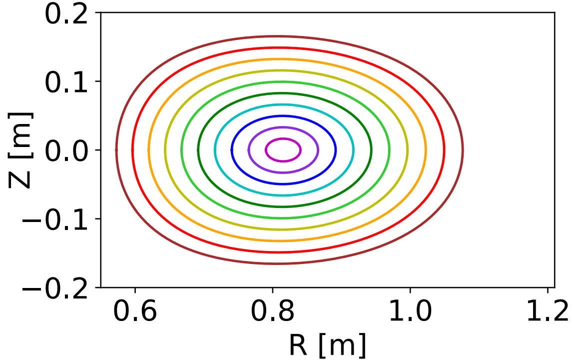
- Inputs:
  - Shape of the magnetic axis.
  - 3-5 other numbers (e.g. current on the axis).
- Outputs:
  - Shape of the surfaces around the axis.
  - Rotational transform on axis.
  - …
- Quasisymmetry guaranteed in a neighborhood of axis.
- Can pick any surface to pass to traditional 3D MHD fixed-boundary 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:

Axis length

Desired axis length

Desired rotational transform

$$f = \overbrace{\frac{1}{L}\int d\ell}^{\text{Average along magnetic axis}} \left|\nabla \mathbf{B}\right|^2 + \frac{w_{\nabla\nabla}}{L}\int d\ell \left|\nabla\nabla \mathbf{B}\right|^2 + w_L\left(L - L_*\right)^2 + w_\iota\left(\iota - \iota_*\right)^2$$

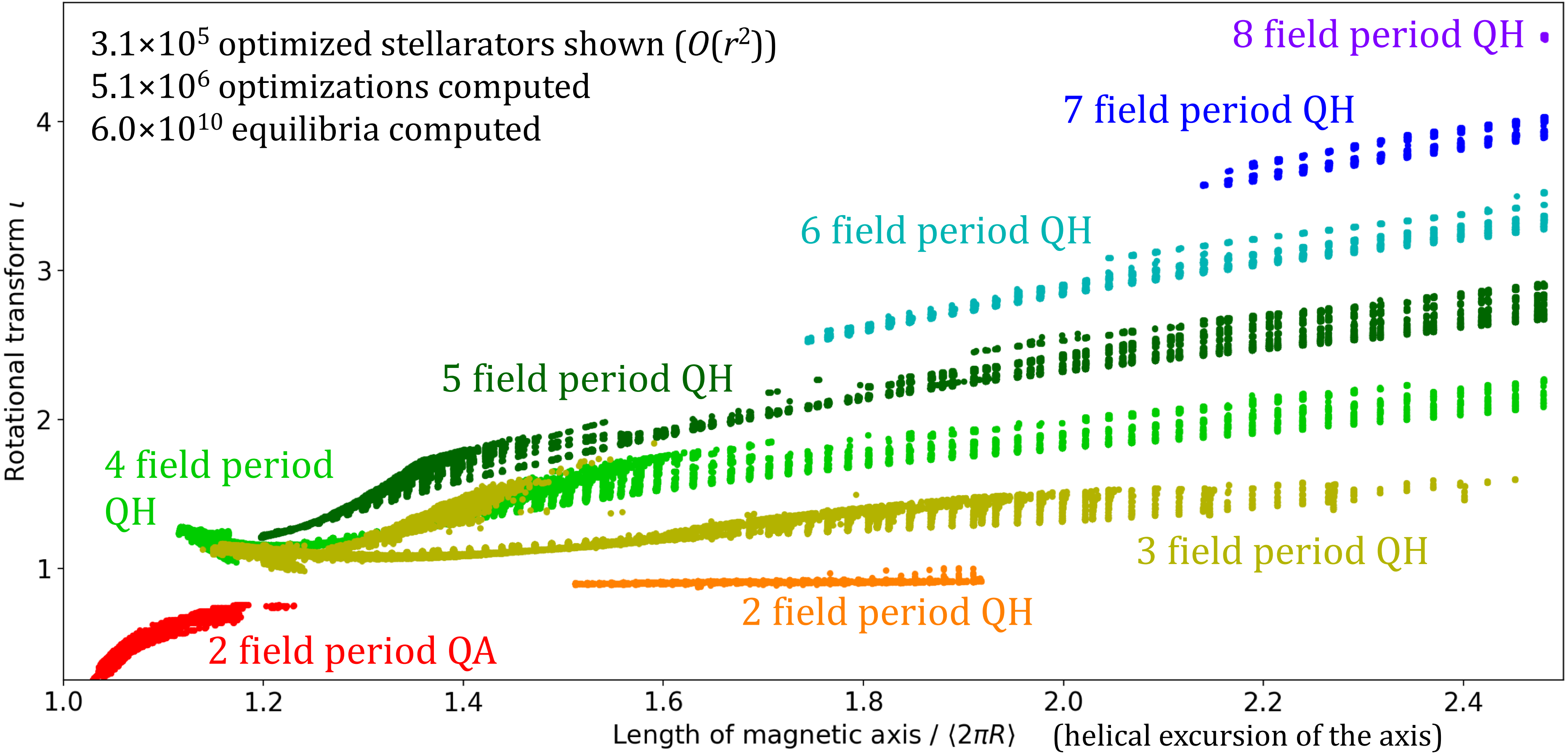
$$+ \frac{w_{B20}}{L}\int d\ell \underbrace{\left(B_{20} - \frac{1}{L}\int d\ell' B_{20}\right)^2}_{\text{Deviation from quasisymmetry at } O(r^2)} + w_{well}\left[\max\left(0,\ \frac{d^2V}{d\psi^2} - W_*\right)\right]^2$$

Magnetic well

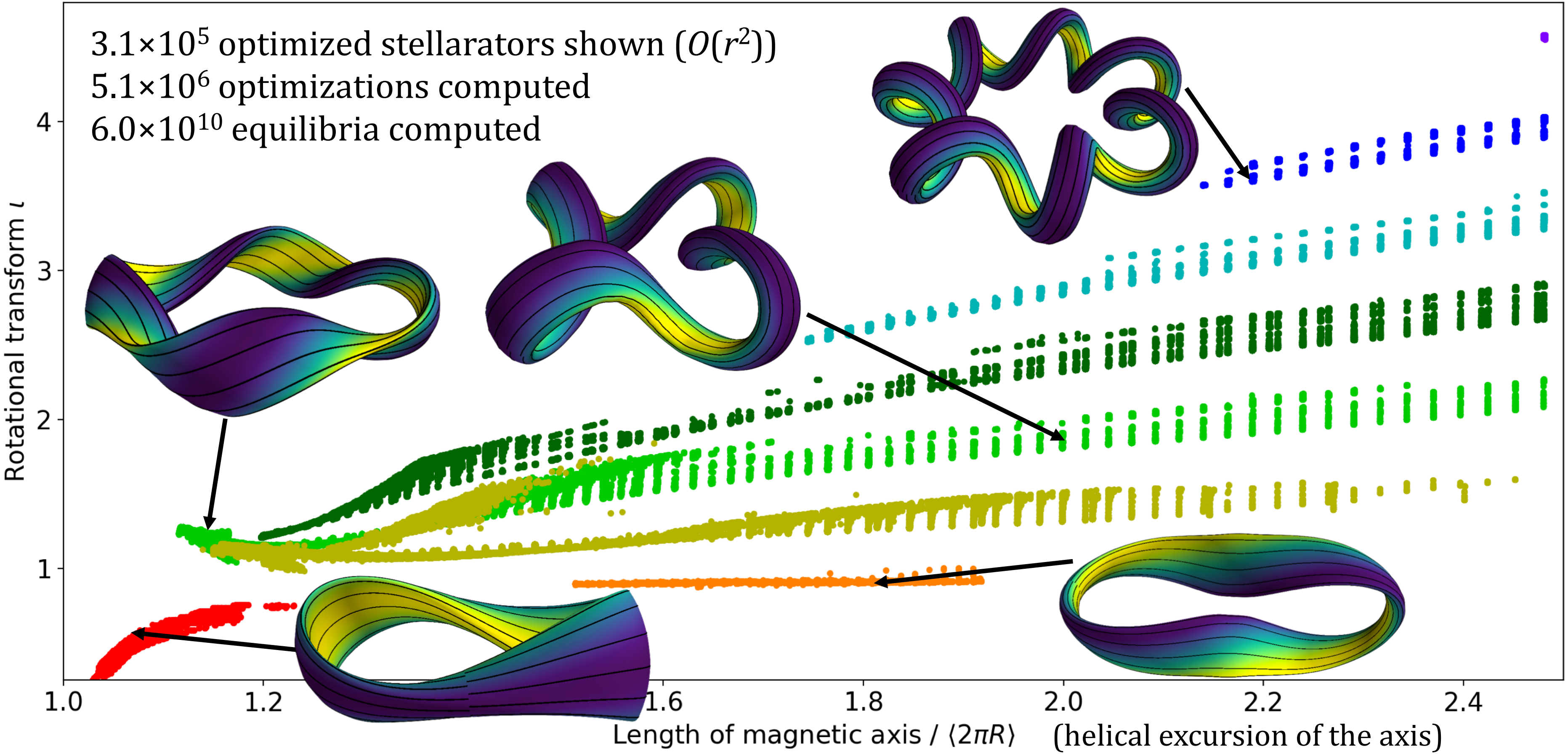
Desired well

$w_{\nabla\nabla},\ w_L,\ w_\iota,\ w_{B20},\ w_{well}$: Weights chosen by user

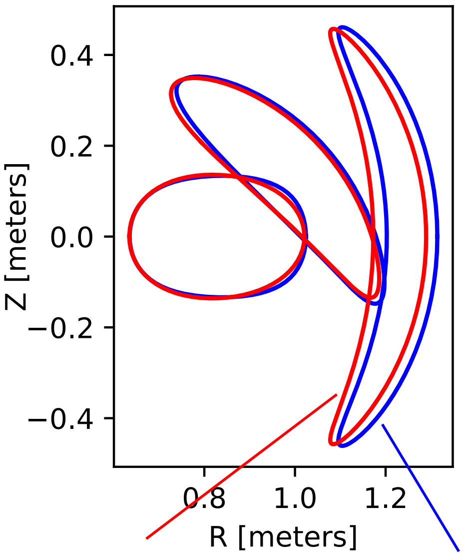
The near-axis equations can be solved so quickly that tensor-product scans over many parameters are feasible



The near-axis equations can be solved so quickly that tensor-product scans over many parameters are feasible



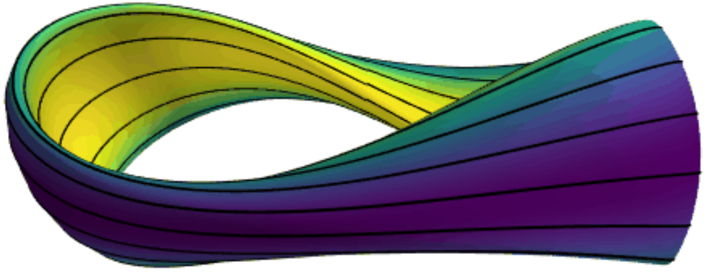
The near-axis expansion can yield configurations very similar to finite-aspect-ratio optimization, but much faster



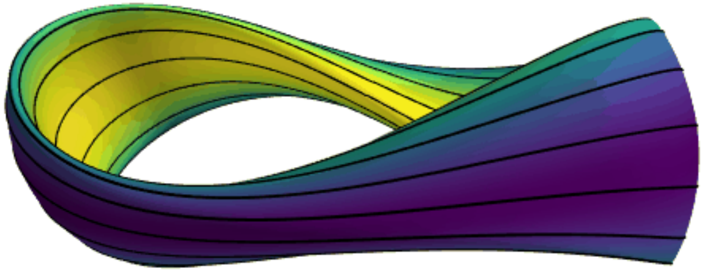
Optimized near-axis expansion

Finite aspect ratio optimization*

Finite aspect ratio optimization



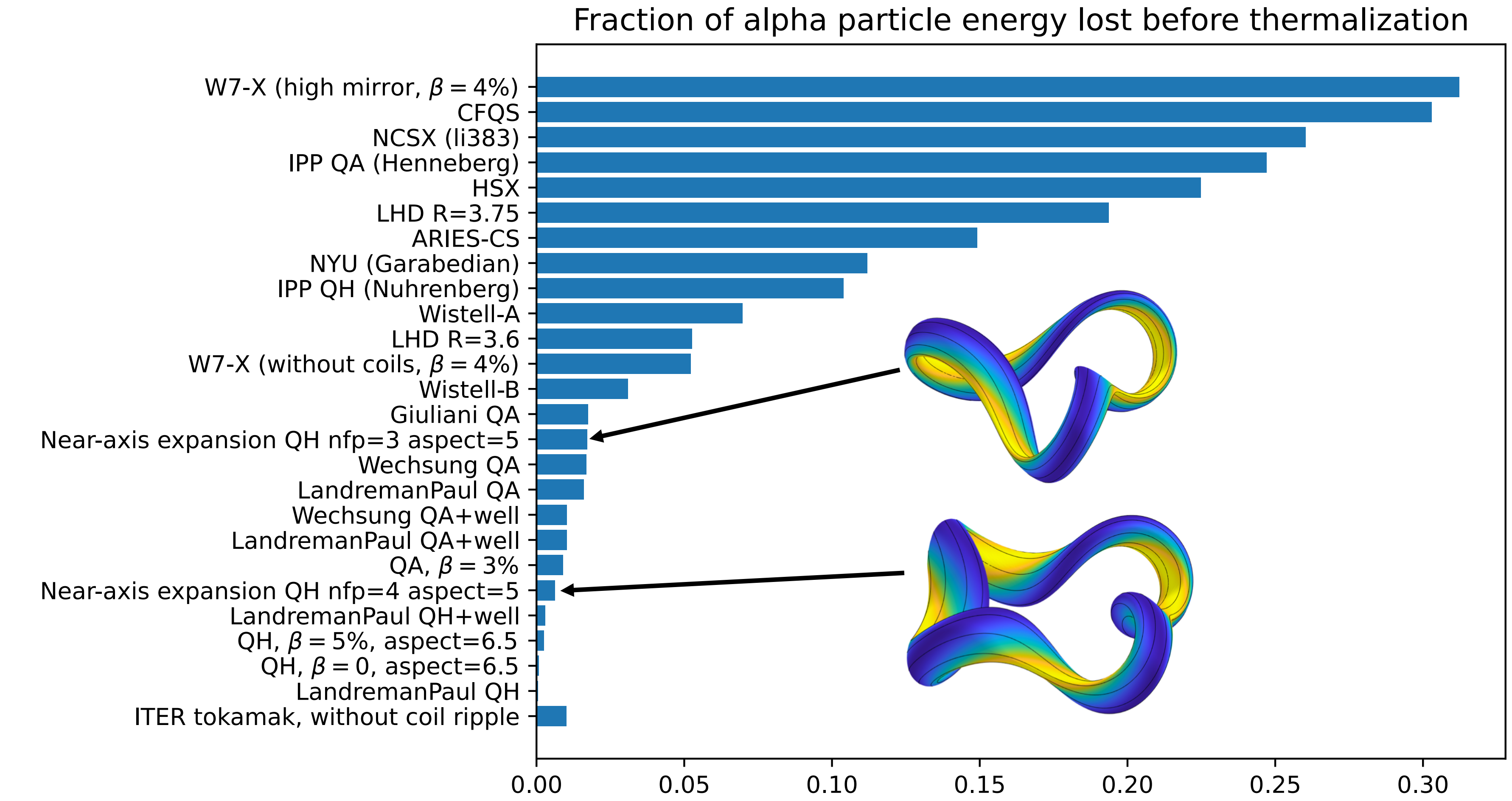
Optimized near-axis expansion



Time for 1 objective evaluation:	5e-4 CPU-sec
Total time for optimization (cold start):	1 CPU-sec

** ML & Paul, PRL (2022)*

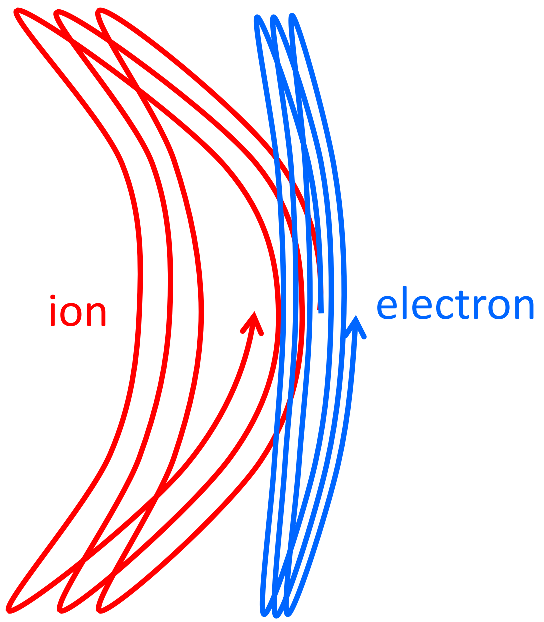
In some cases, the near-axis construction can directly generate configurations with excellent confinement



- Optimizing stellarator geometry for precise quasisymmetry
- Constructing quasisymmetric geometries using near-axis expansion
- Self-consistent current in the plasma
- Future directions

The bootstrap current arises in tokamaks & stellarators when the density & temperature become significant

- Ions and electrons have different trajectories. Different mean flows = electric current.
- Current depends on geometry, density, & temperature.
- For $\beta > 0$, we don't know **B** until we include this effect.
- Computing the bootstrap current requires kinetic theory: coupled 4D advection-dominated integro-differential equations.
- Need *self-consistency* between MHD equilibrium and kinetic equation.
- How can a self-consistent bootstrap current calculation be integrated with stellarator optimization?



New idea: exploit quasisymmetry & use analytic expressions for tokamaks

Pytte & Boozer (1981), Boozer (1983):

$$\iota \rightarrow \iota - N$$

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

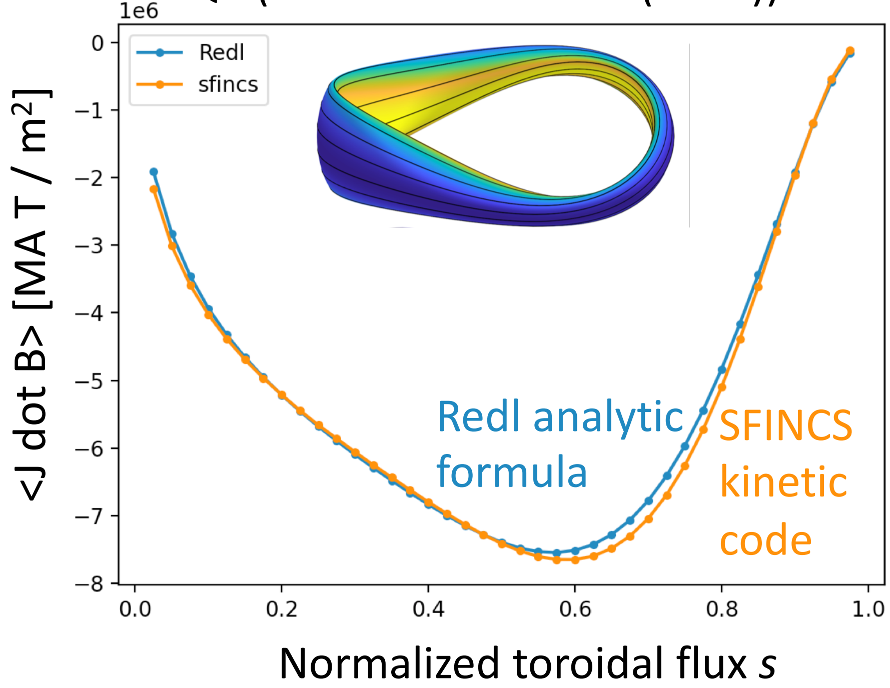
View Online Export Citation

A. Redl,[1,2,a)] C. Angioni,[1] E. Belli,[3] O. Sauter,[4] ASDEX Upgrade Team[b)] and EUROfusion MST1 Team[c)]

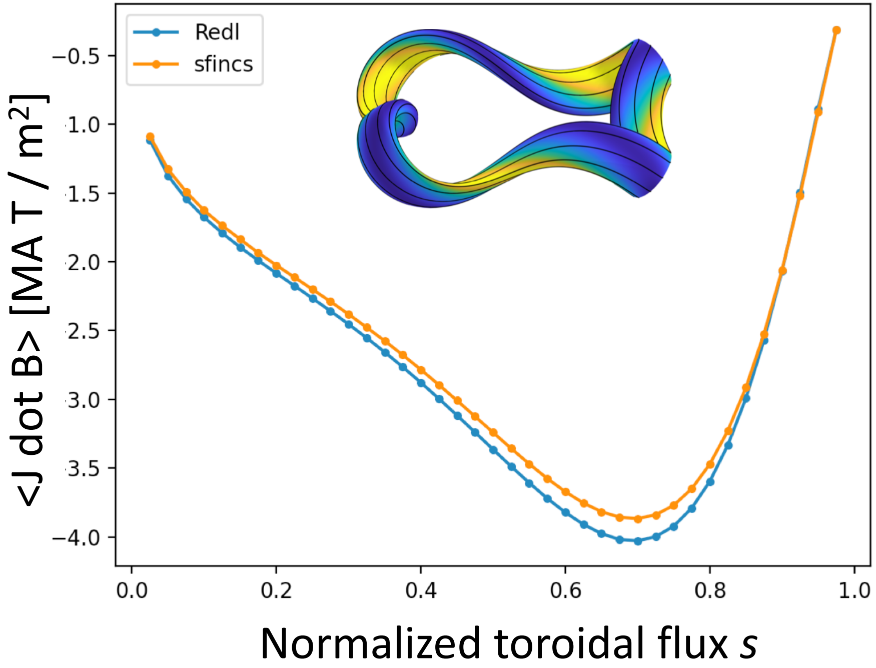
Before doing new optimizations: Redl formula is accurate in previous QA & QH stellarators

$n_e = (1 - s^5)\ 4\text{x}10^{20}\ \text{m}^{-3}, \quad T_e = T_i = (1 - s)\ 12\ \text{keV}$

QA (Landreman & Paul (2021))



QH (Landreman & Paul (2021))



(Not self-consistent yet)

Optimization recipe

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

Boundary aspect ratio Minor radius

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

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

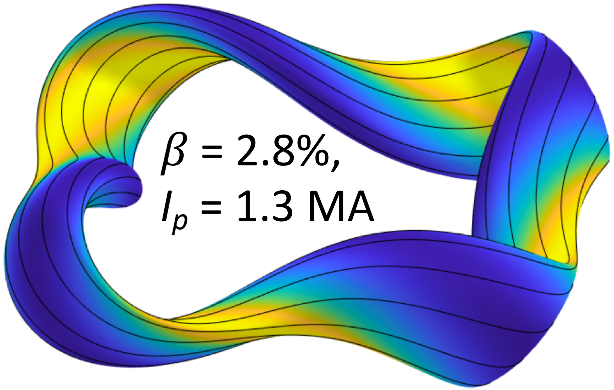
- Parameter space: {$R_{m,n}$, $Z_{m,n}$, toroidal flux, current spline values}

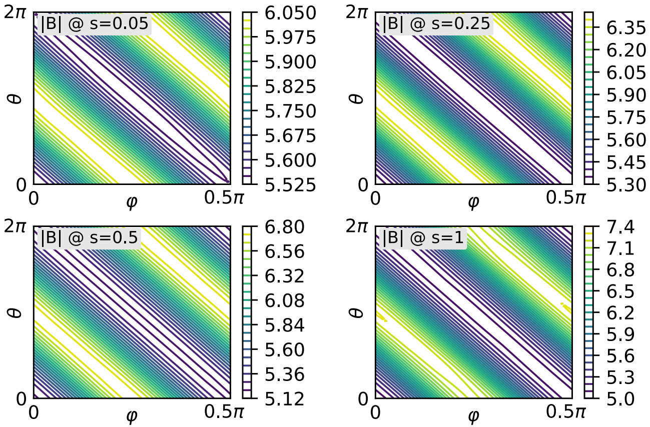
  or {$R_{m,n}$, $Z_{m,n}$, toroidal flux, iota spline values}
- Cold start
- Algorithm: default for least-squares in scipy (trust region reflective)
- Steps: increasing # of modes varied: m and |n/nfp| up to j in step j

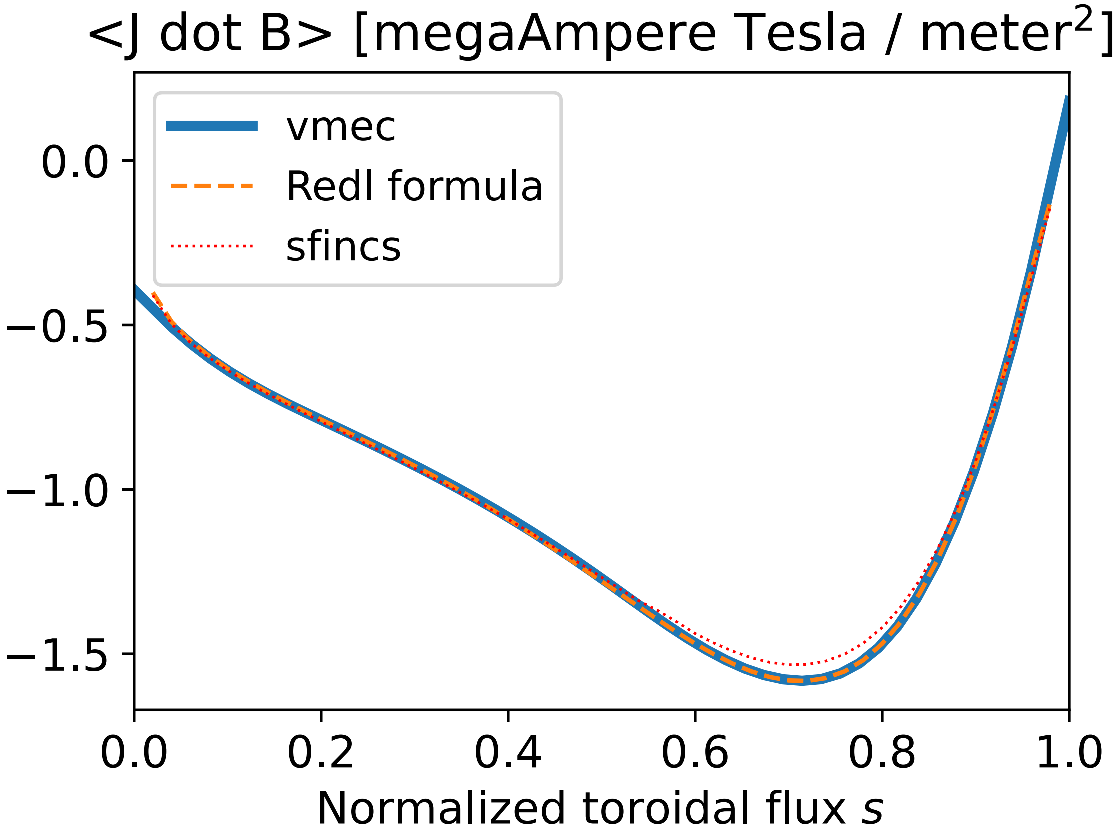
Example of optimization with self-consistent bootstrap current

n_{e0} = 2.5e20/meters^3

$T_{e0} = T_{i0}$ = 10 keV

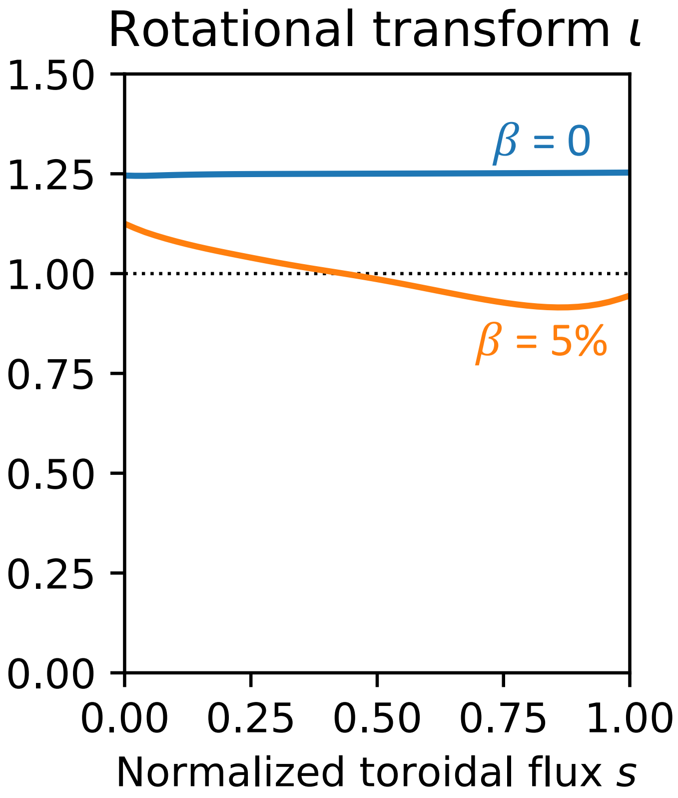






To reach reactor-relevant 5% beta in QH without crossing iota=1, a constraint on iota can be included

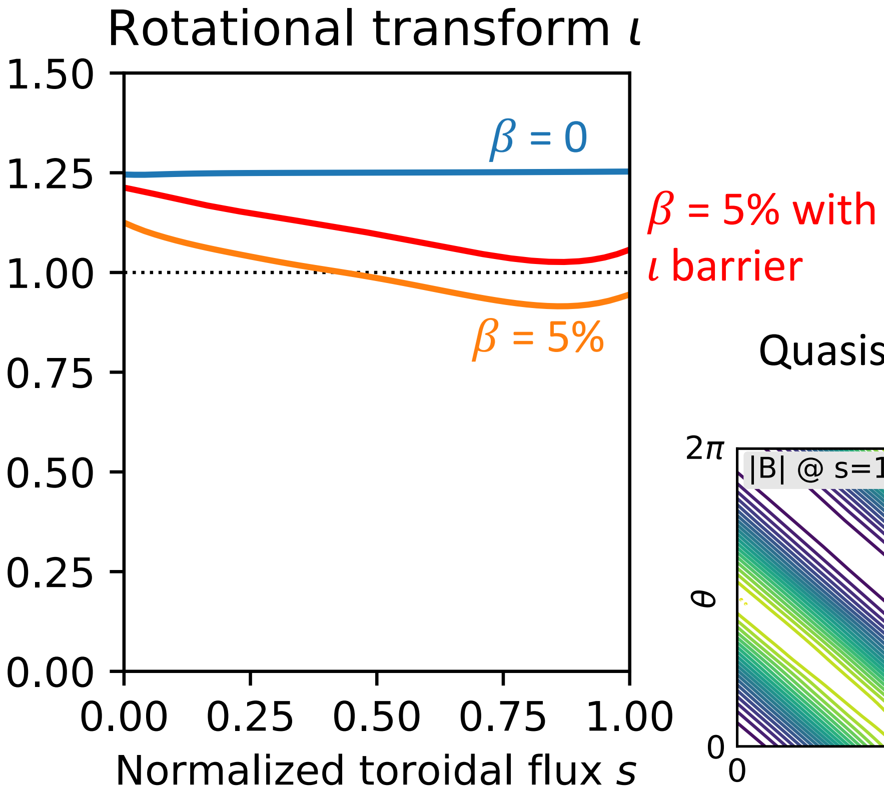
Crossing iota=1, the worst resonance, is probably unacceptable.



n_{e0} = 3e20/meters^3, $T_{e0} = T_{i0}$ = 15 keV

To reach reactor-relevant 5% beta in QH without crossing iota=1, a constraint on iota can be included

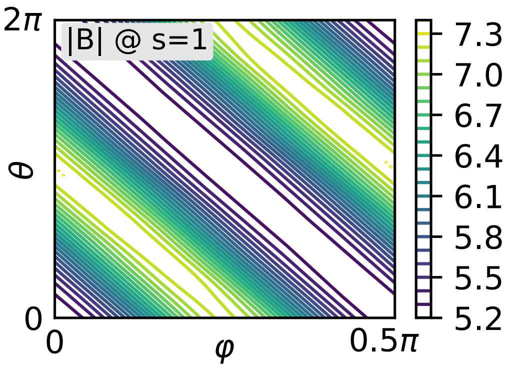
Crossing iota=1, the worst resonance, is probably unacceptable.

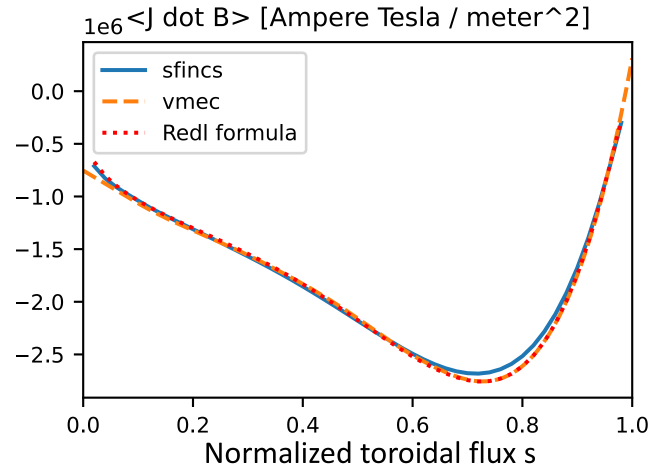


Solution: Add barrier term in objective

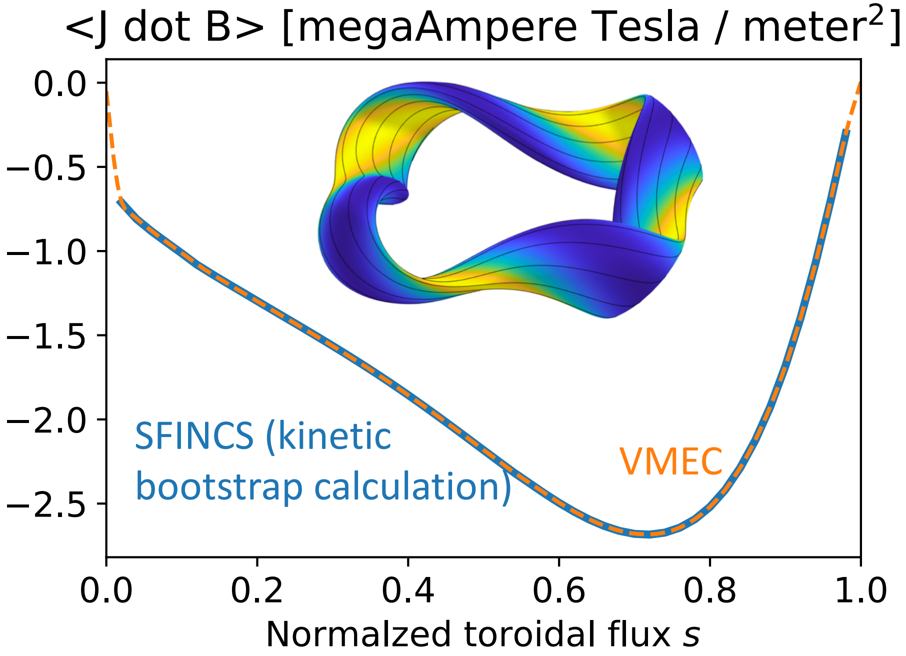
$$f \mathrel{+}= \int_0^1 ds \left[\min\left(\left| \iota(s) \right| - 1.03,\ 0 \right) \right]^2$$

Quasisymmetry & bootstrap consistency remain good:





If you want *perfectly* self-consistent current, you can do a few fixed-point iterations at the end

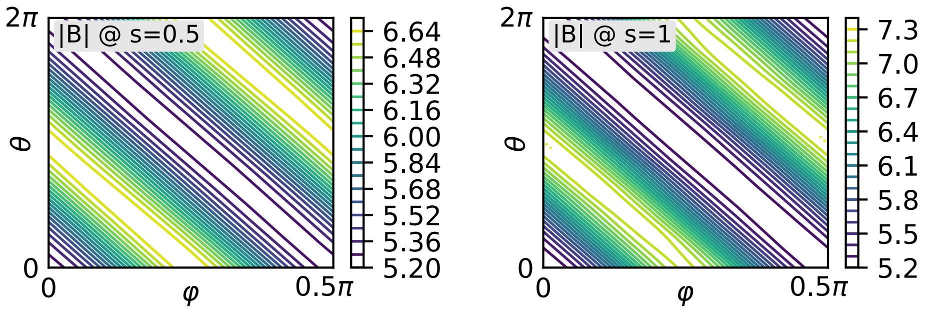


$$\langle \beta \rangle = 5\%, \qquad \varepsilon_{eff}^{3/2} < 6 \times 10^{-5}$$

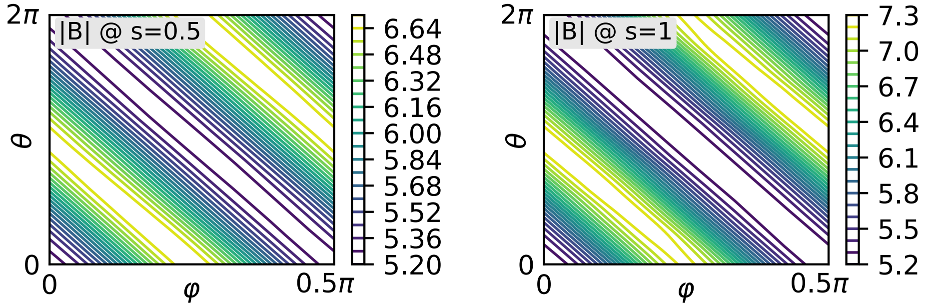
α-particle losses < 0.3%

No significant degradation in quasisymmetry:

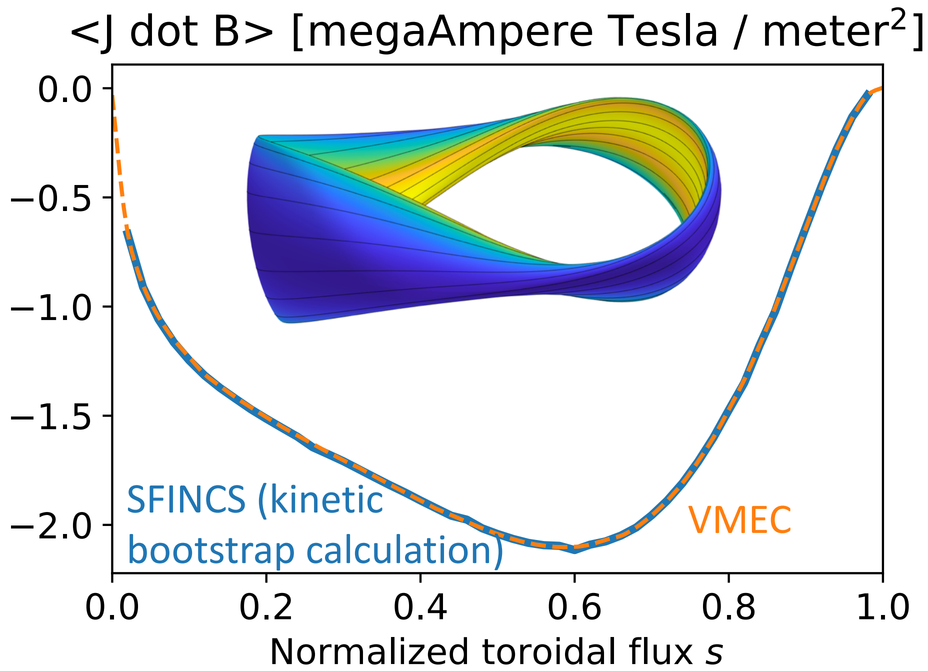
Optimization with Redl current



After SFINCS fixed-point iterations



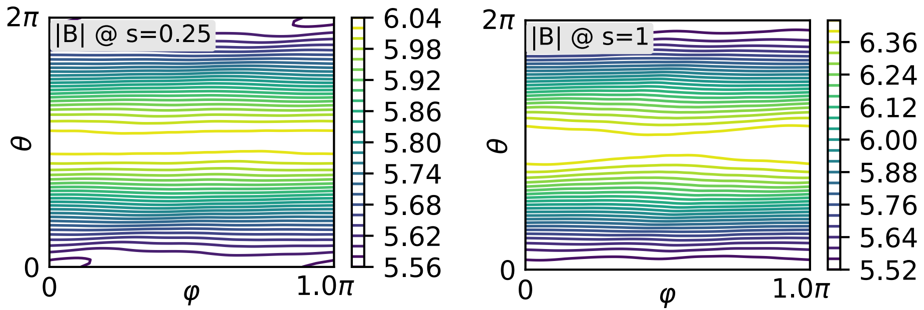
The optimization with self-consistent bootstrap current also works for quasi-axisymmetry

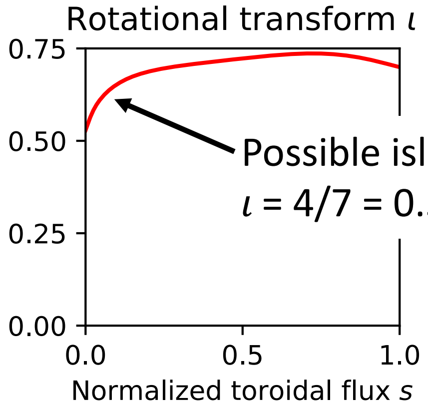


$$\langle \beta \rangle = 3\%, \qquad \varepsilon_{eff}^{3/2} < 7\times 10^{-6}$$

α-particle losses < 1%

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



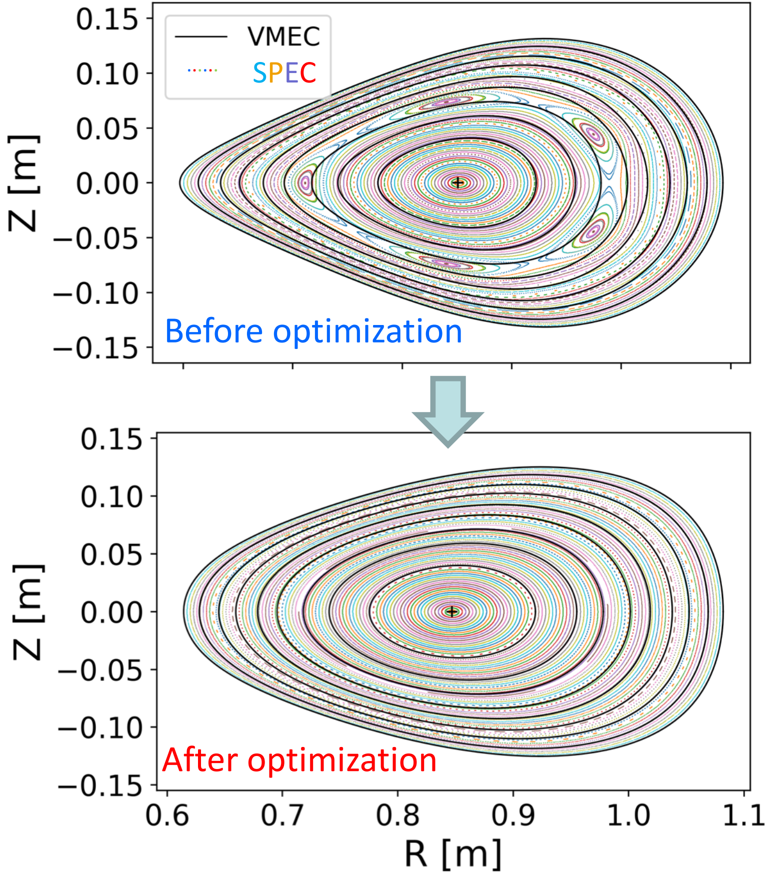


Possible islands where $\iota = 2/3$, $\iota = 4/7 = 0.57$?

- Optimizing stellarator geometry for precise quasisymmetry
- Constructing quasisymmetric geometries using near-axis expansion
- Self-consistent current in the plasma
- Future directions

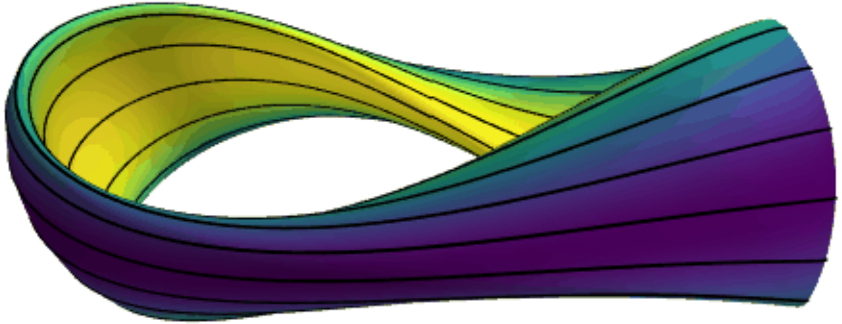
Future directions

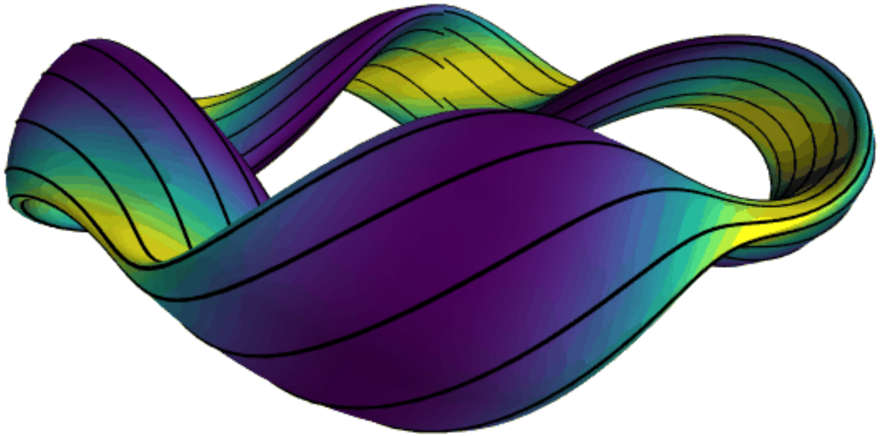
- Eliminate islands at high β (plasma pressure & current)
- Coils & MHD stability for the high β configurations
- Quasi-isodynamic (|B| contours close poloidally)
- Understand trade-offs using multi-objective optimization
- Robustness to uncertainty in the pressure profile
- Interact with systems codes & reactor studies
- Combined coil + confinement (1-stage) optimization
- Optimization of turbulence & divertor

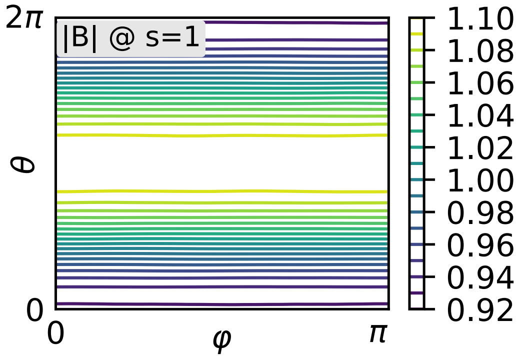


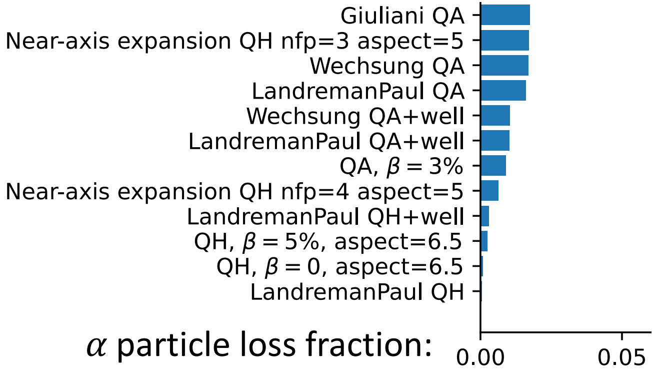
ML, Medasani & Zhu (2021)

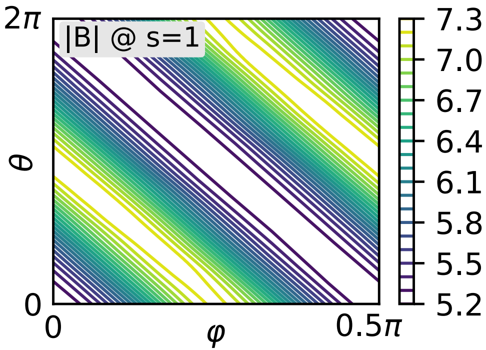
The Hidden Symmetries project has enabled significant progress in stellarator optimization & confinement





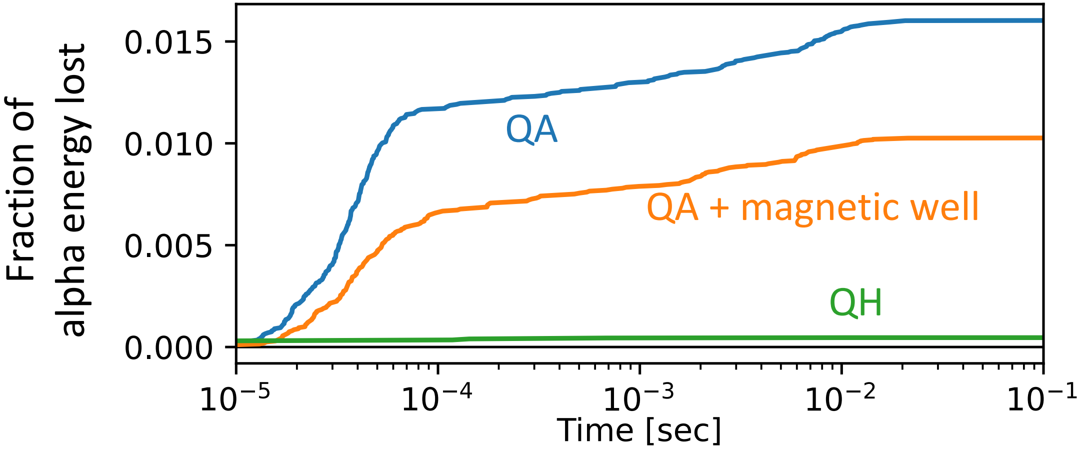




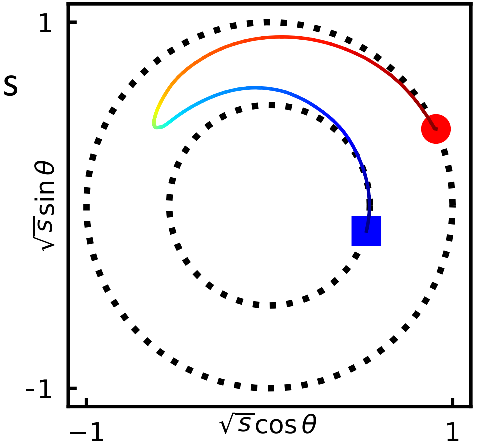


Extra slides

Why do the configurations with best quasisymmetry not have the best trajectory confinement?



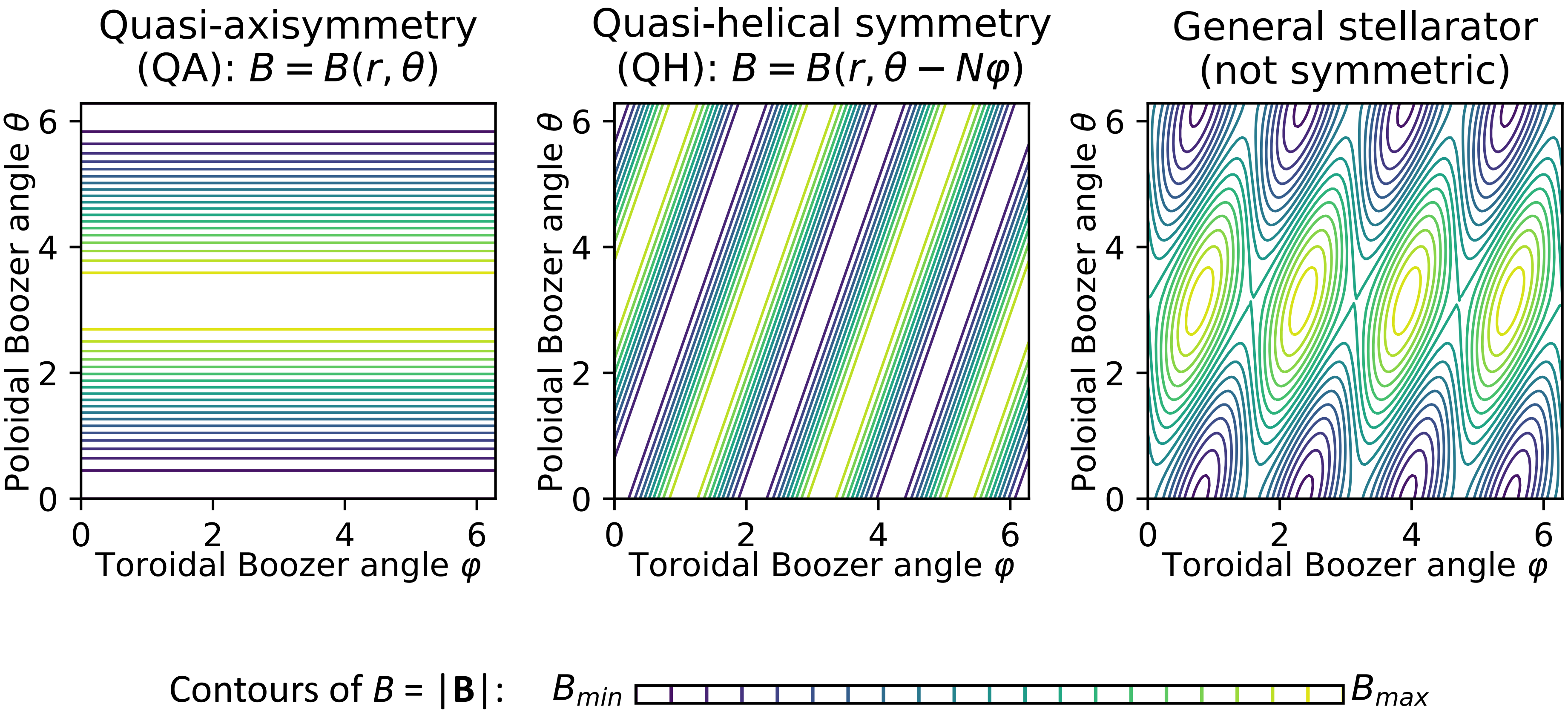
Lost trajectories in the new QA look like this:



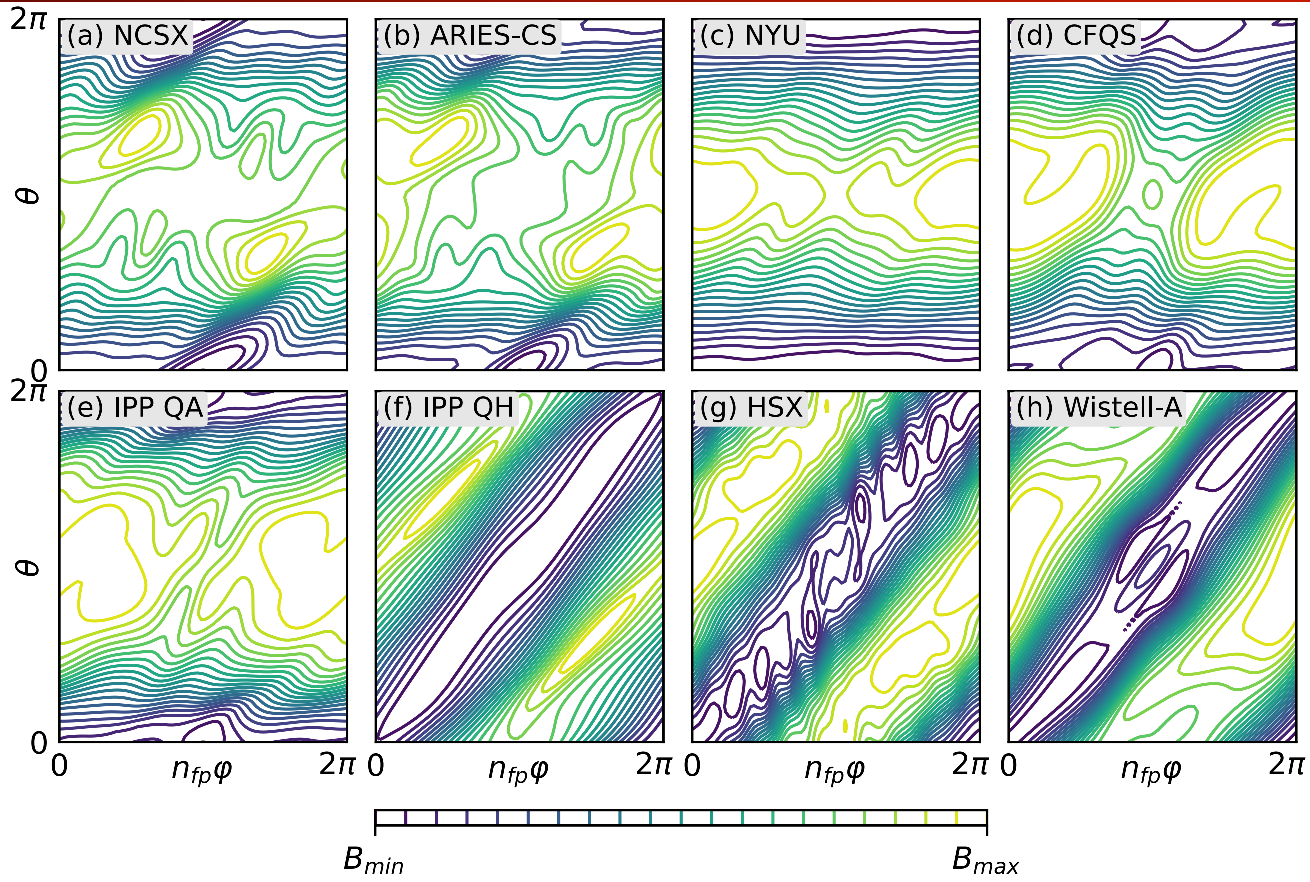
Width of banana orbit $\Delta s \approx \left| \frac{mvR\sqrt{2r\bar{\eta}}}{(\iota - N)\psi_{edge} Ze} \right| \propto \frac{1}{|\iota - N|}$

For fixed minor radius, $\frac{\Delta s_{QA}}{\Delta s_{QH}} \sim 4$

2 types of quasisymmetry



Previous quasisymmetric configurations

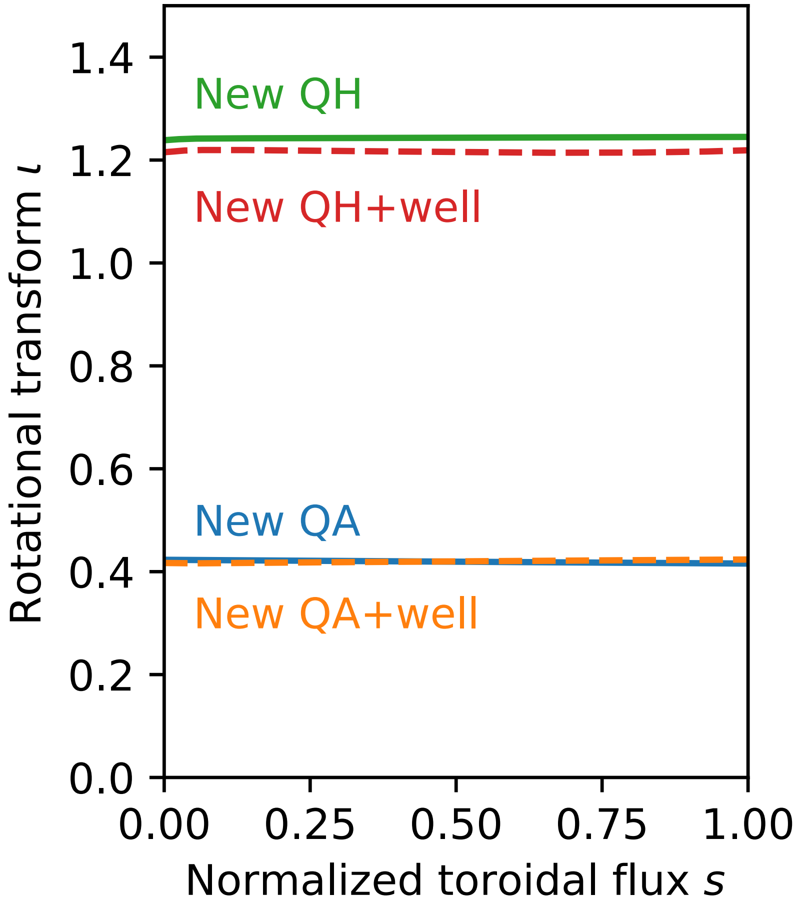


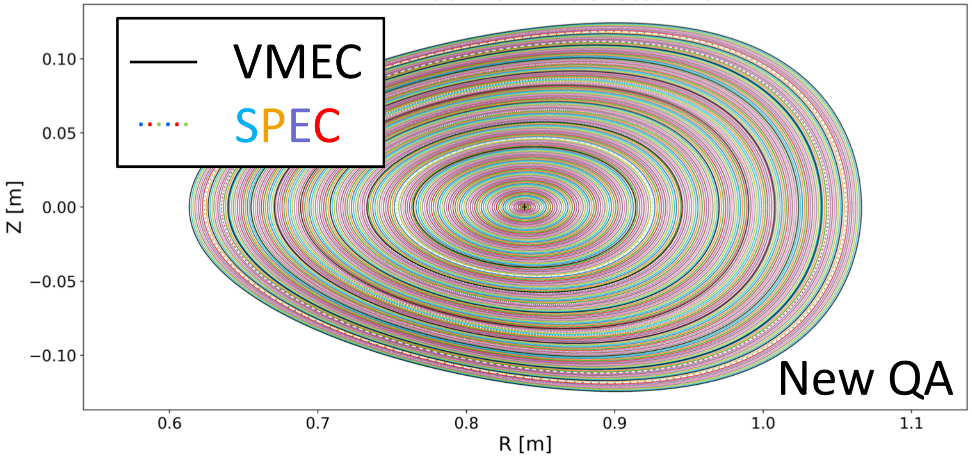
(a) Zarnstorff et al (2001)
(b) Najambadi et al (2008)
(c) Garabedian (2008)
(d) Liu et al (2018)
(e) Henneberg et al (2019)
(f) Nuhrenberg & Zille (1988)
(g) Anderson et al (1995)
(h) Bader et al (2020)

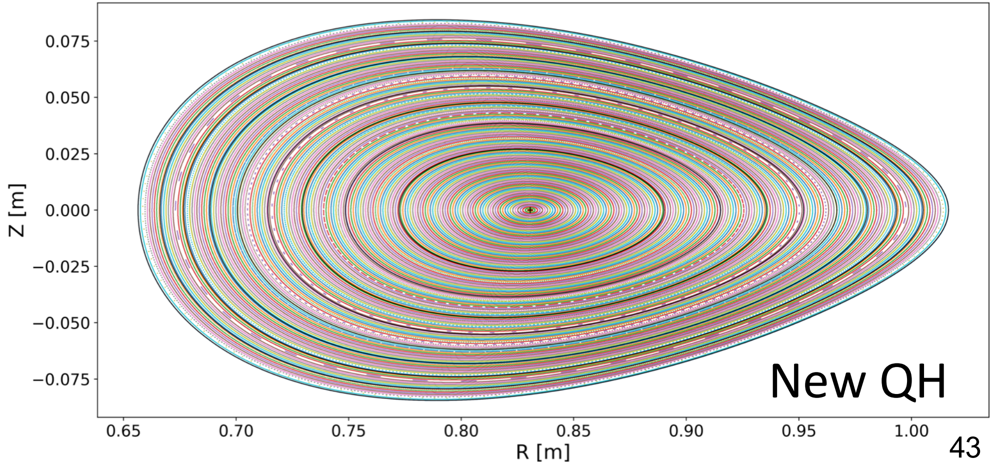
We want
$B = B(r, \theta - N\,\varphi)$

Is there an optimization recipe that can give consistently straight |B| contours?

The new configurations have small magnetic shear







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 $\mathbf{B}_0$.

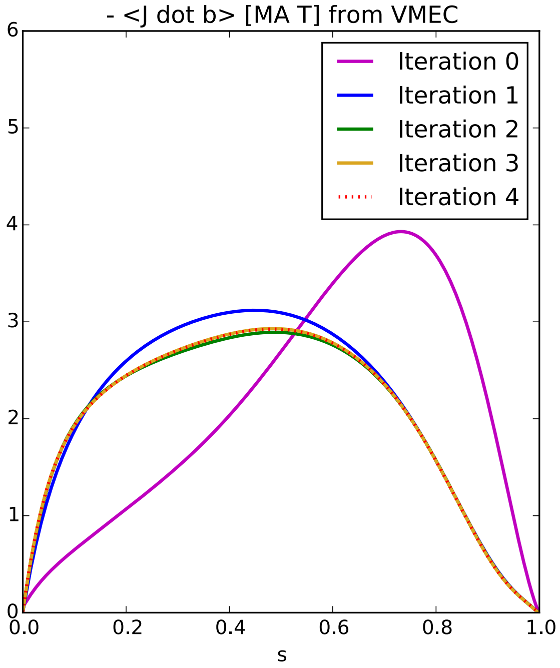
SFINCS: given $\mathbf{B}_0$, determine $I_1(s)$.

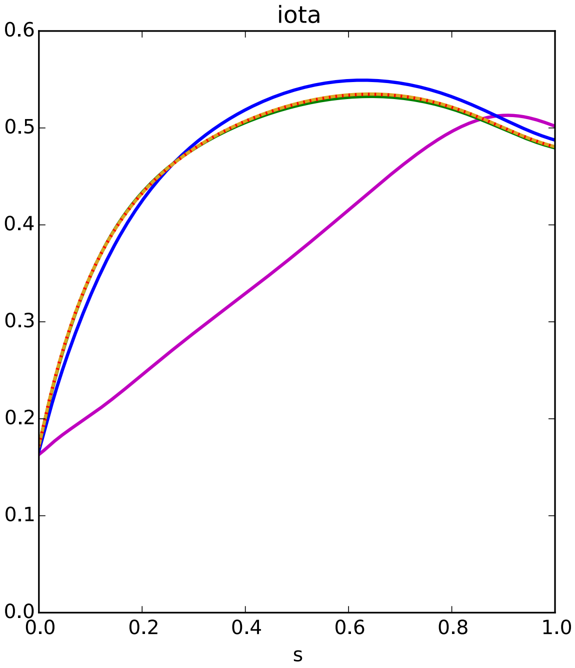
VMEC: given $I_1(s)$, determine $\mathbf{B}_1$.

SFINCS: given $\mathbf{B}_1$, determine $I_2(s)$.

…

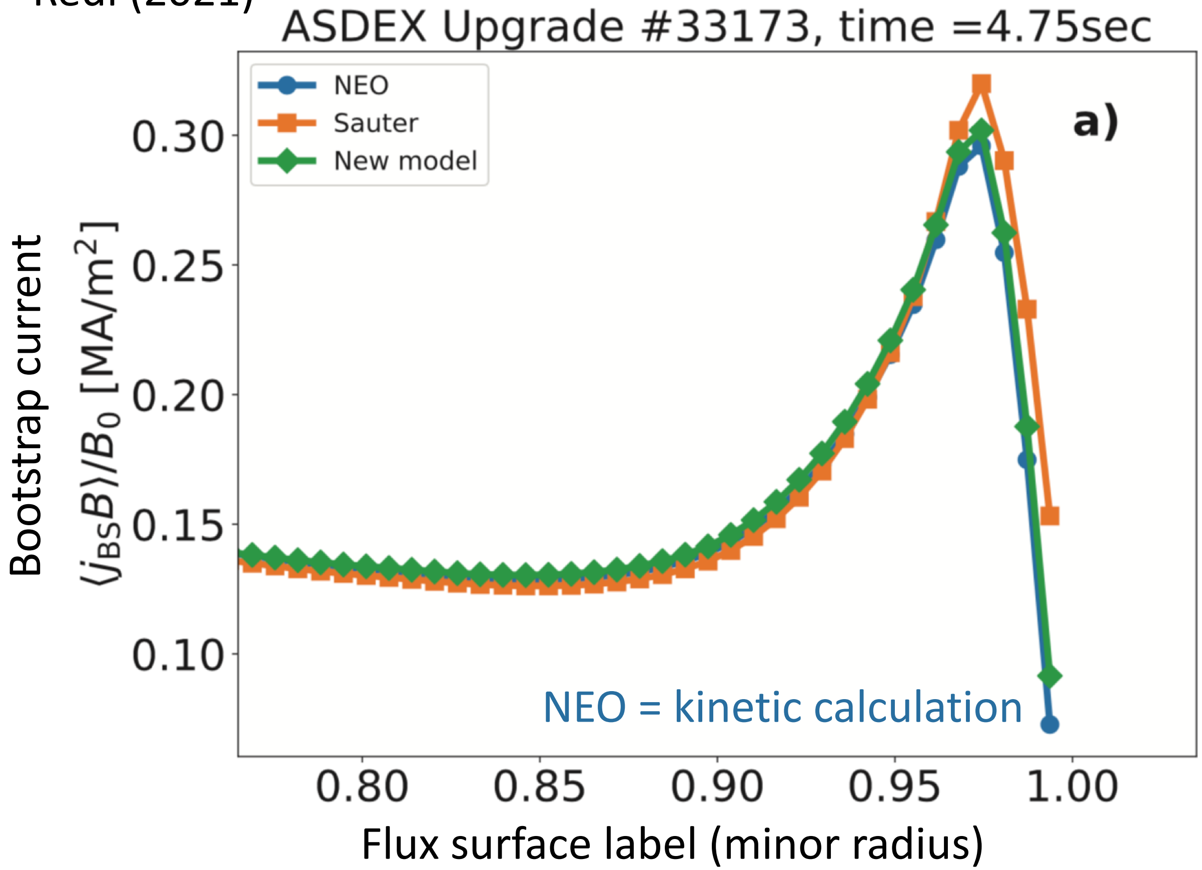
SFINCS: >20 node-seconds per surface for reactor n/T, cost much higher at low collisionality, uses PETSc, tricky to set resolution parameters





New idea: exploit quasisymmetry & use analytic expressions for tokamaks

Redl (2021)



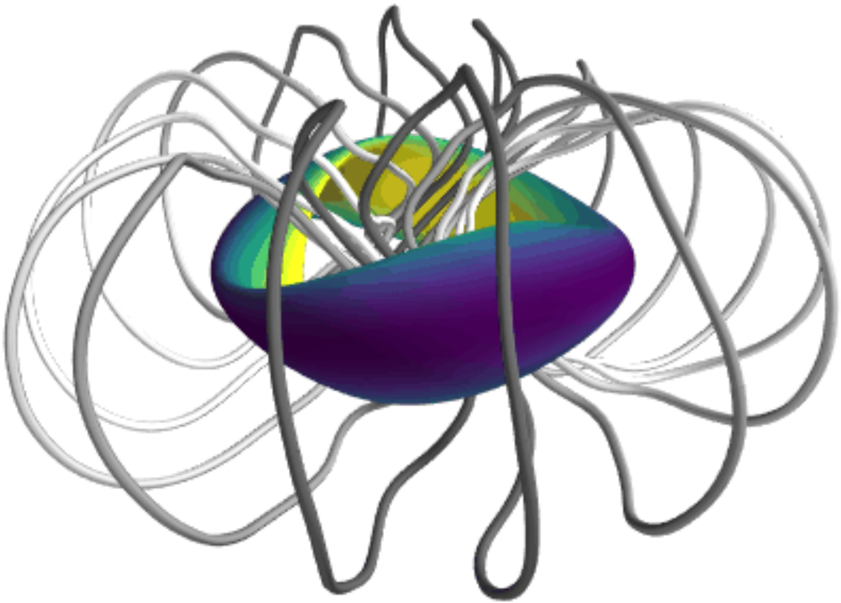
Geometry enters through

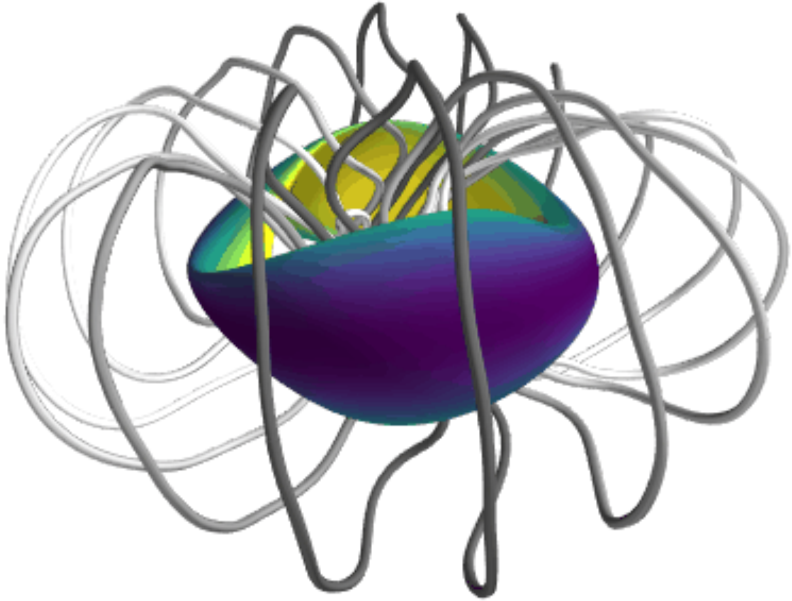
$$f_t = 1 - f_c = 1 - \frac{3}{4}\langle B^2 \rangle \int_0^{1/B_{\max}} \frac{\lambda d\lambda}{\langle \sqrt{1-\lambda B} \rangle}.$$

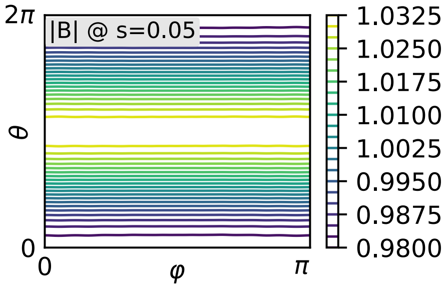
$$\nu_{e*} = 6.921 \cdot 10^{-18} \frac{qRn_e Z \ln \Lambda_e}{T_e^2 \epsilon^{3/2}},$$

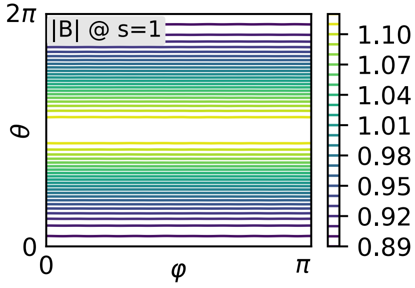
$$\nu_{i*} = 4.90 \cdot 10^{-18} \frac{qRn_i Z^4 \ln \Lambda_{ii}}{T_i^2 \epsilon^{3/2}},$$

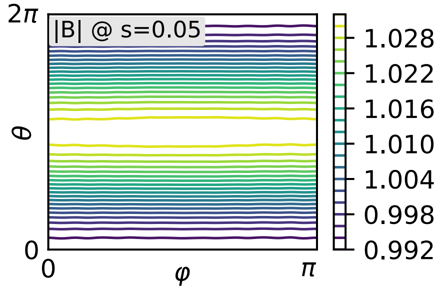
Decent 16-coil solutions have been found for the new QAs

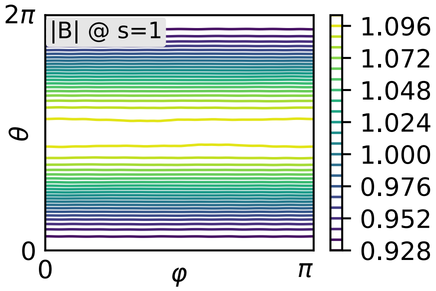






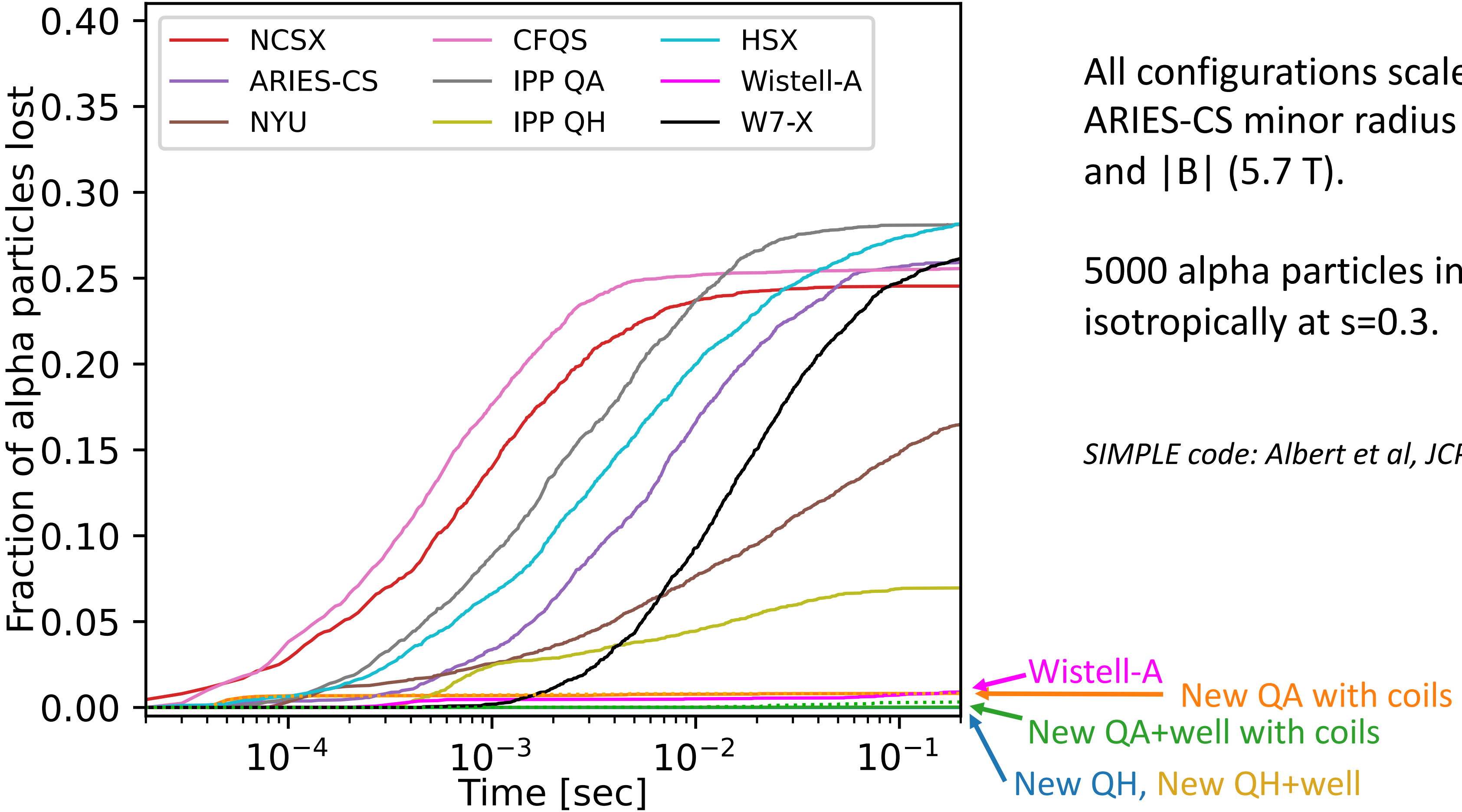






By Florian Wechsung @ NYU. <R>/10 between filament centers. Haven’t looked at the QHs yet

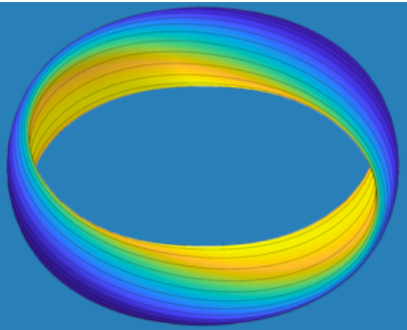
The symmetry yields extremely good confinement of collisionless trajectories



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).



latest

Search docs

CONTENTS

EXAMPLES

Simsopt documentation

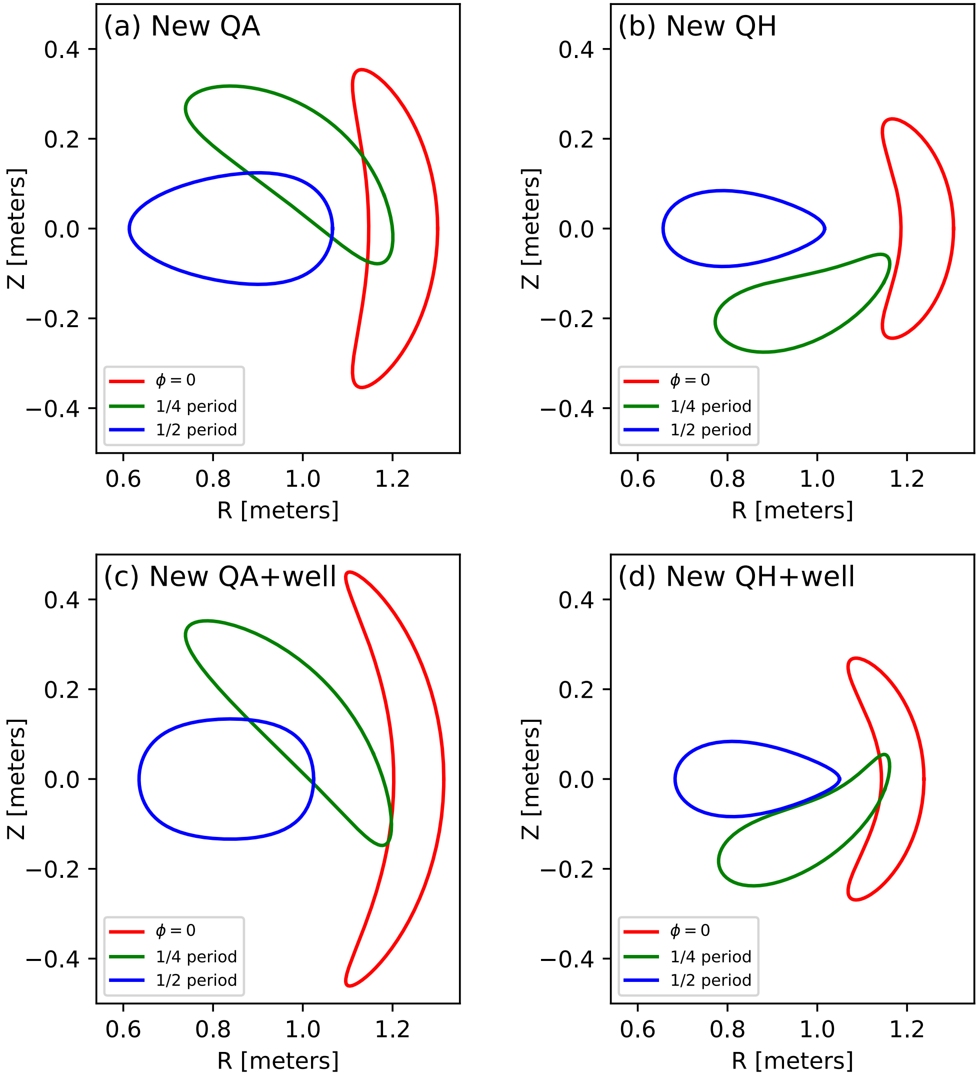
`simsopt` is a framework for optimizing stellarators. The high-level routines are in python, with calls to C++ or fortran where needed for performance. Several types of components are included:

- 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.

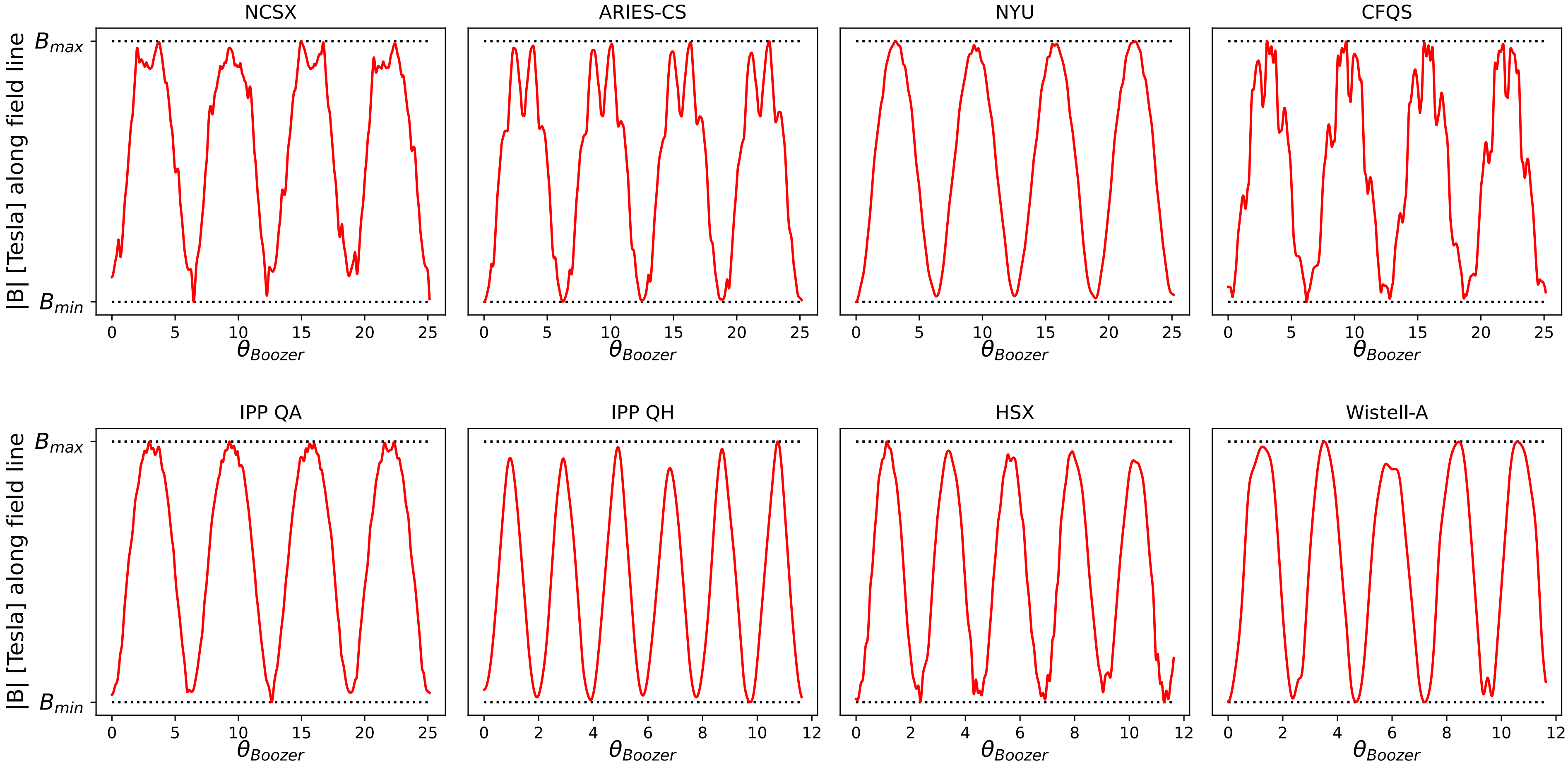
The design of `simsopt` is guided by several principles:

- 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.

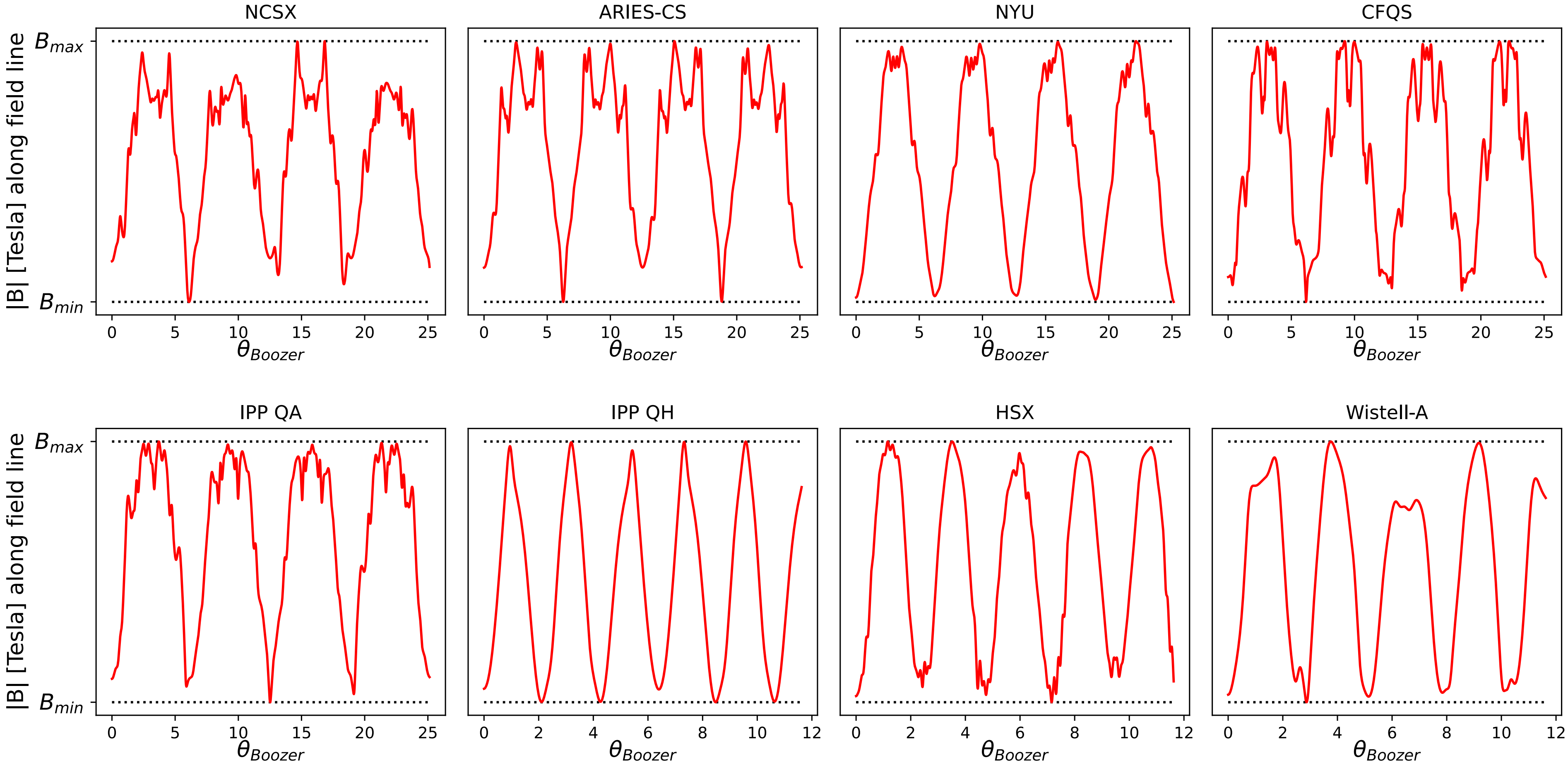
`simsopt` is fully open-source, and anyone is welcome to use it, make suggestions, and contribute.


(a) New QA
(b) New QH
(c) New QA+well
(d) New QH+well
$\phi = 0$
1/4 period
1/2 period
Z [meters]
R [meters]
0.4
0.2
0.0
−0.2
−0.4
0.6
0.8
1.0
1.2

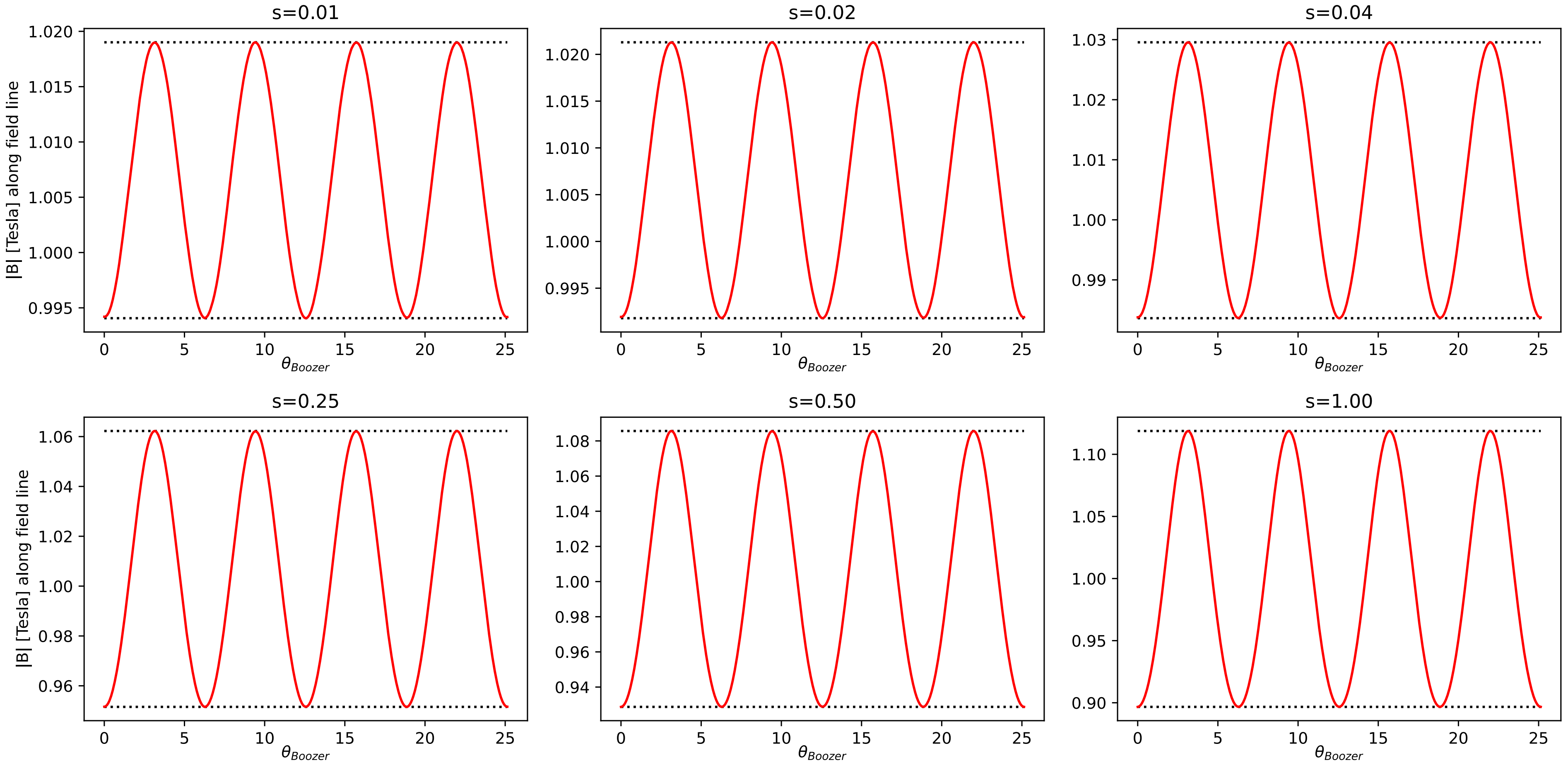
Previous quasisymmetric configurations (s=0.5)



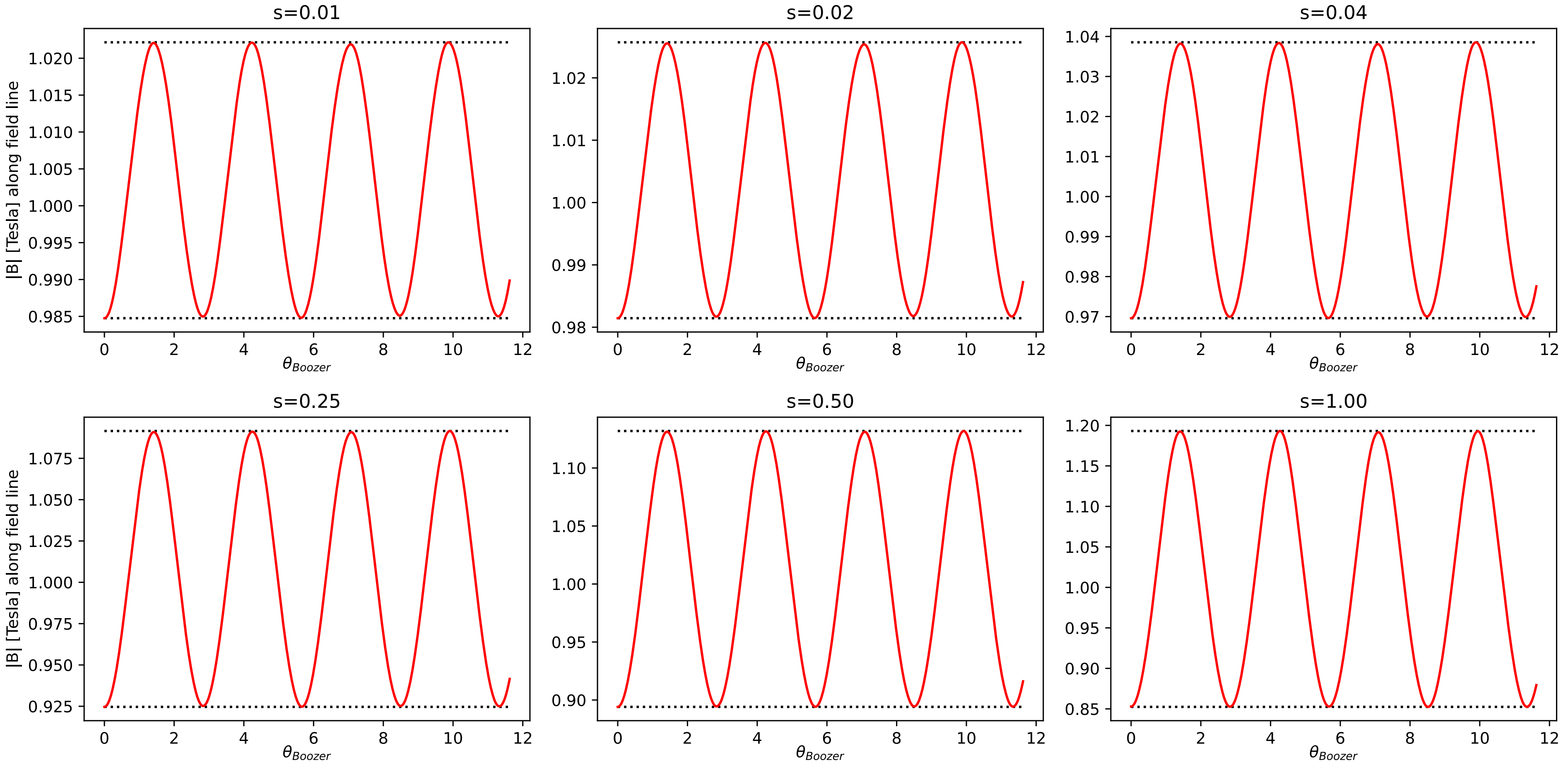
Previous quasisymmetric configurations (s=1)



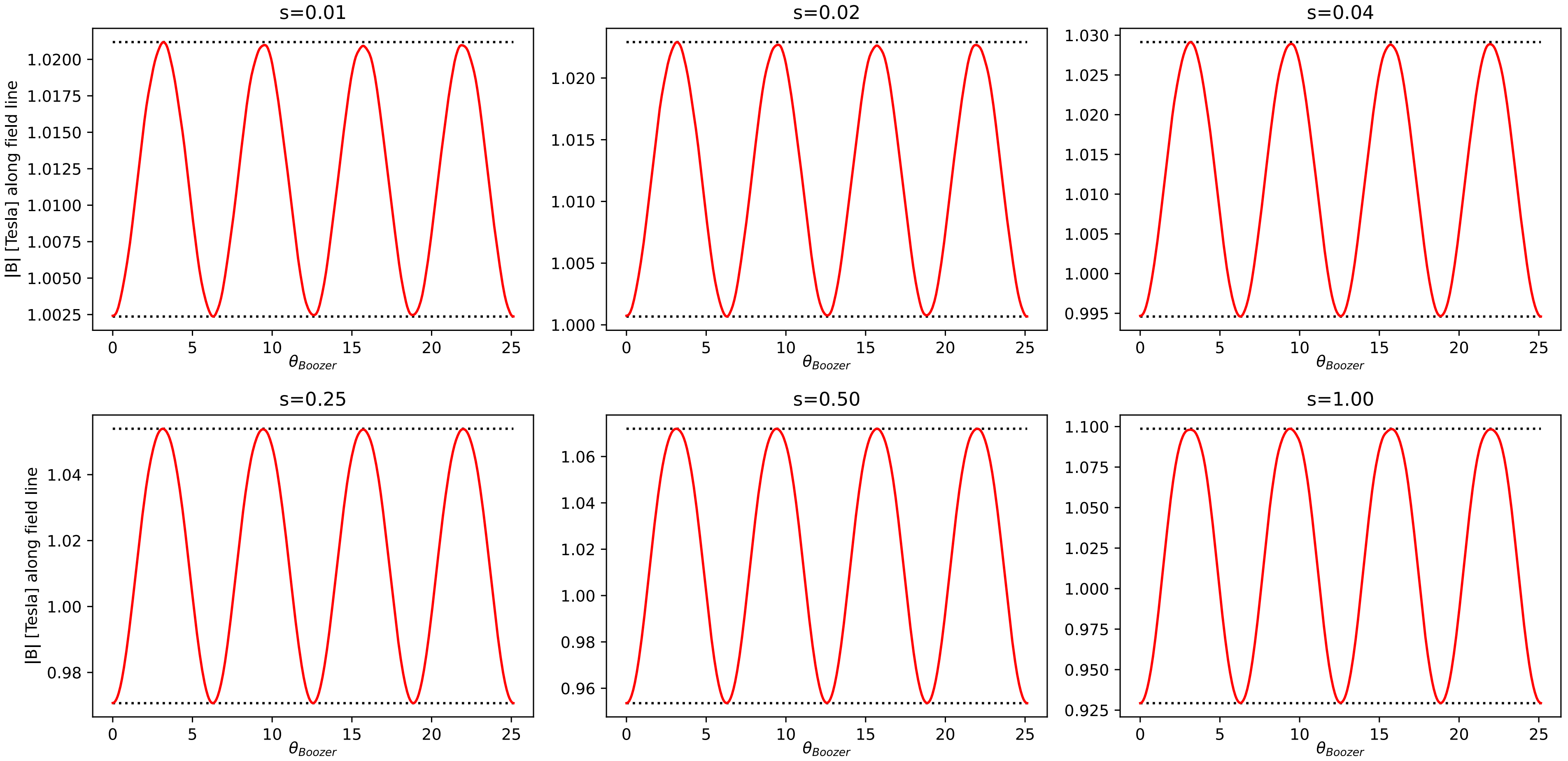
|B| along a field line for new QA



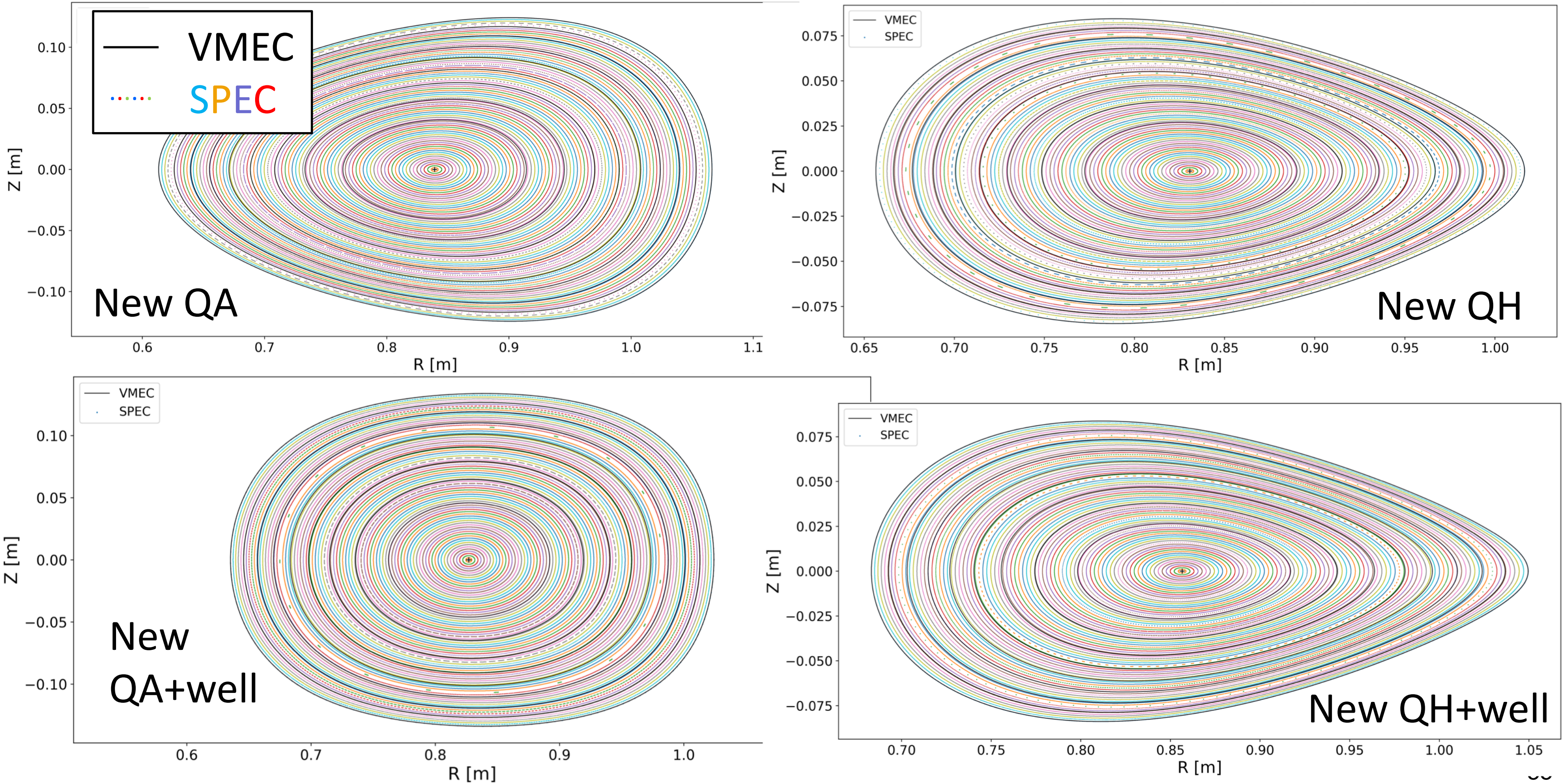
|B| along a field line for new QH



|B| along a field line for new QA with magnetic well

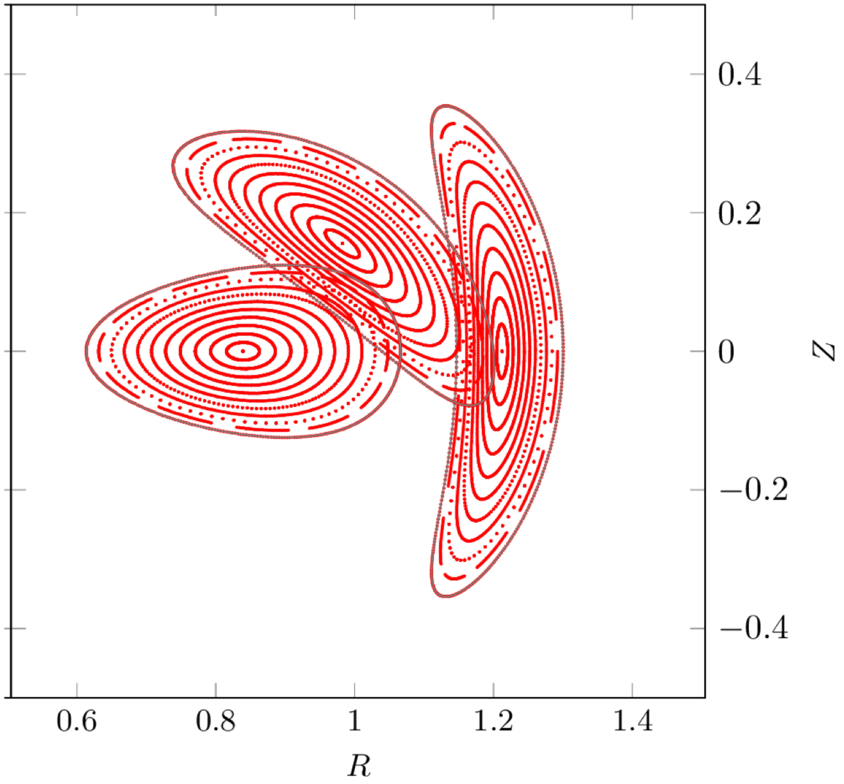


SPEC confirms the new QA/QH configurations have good surfaces

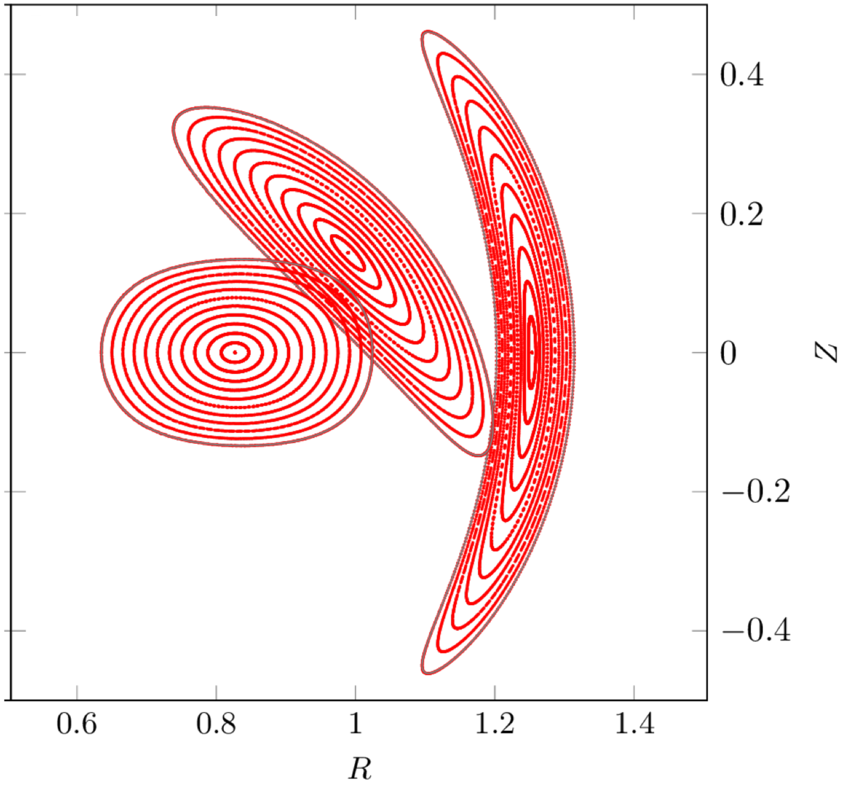


Good flux surface exist with coils

New QA



New QA+well



Overview

- 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]

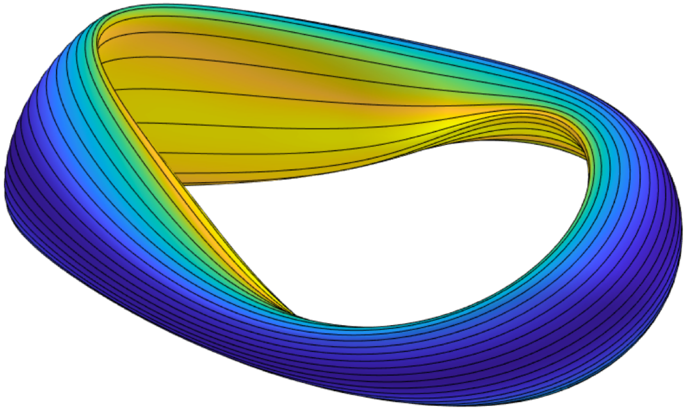
[1] Hirshman & Whitson, *Phys. Fluids* (1993)

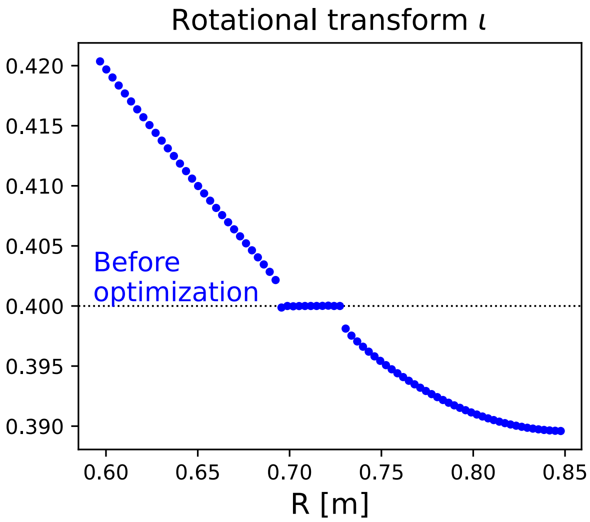
[2] Hudson, Dewar, et al, *Phys. Plasmas* (2012)

[3] Greene, *J. Math. Phys.* (1979)

[4] Hanson & Cary, *Phys. Fluids* (1984)

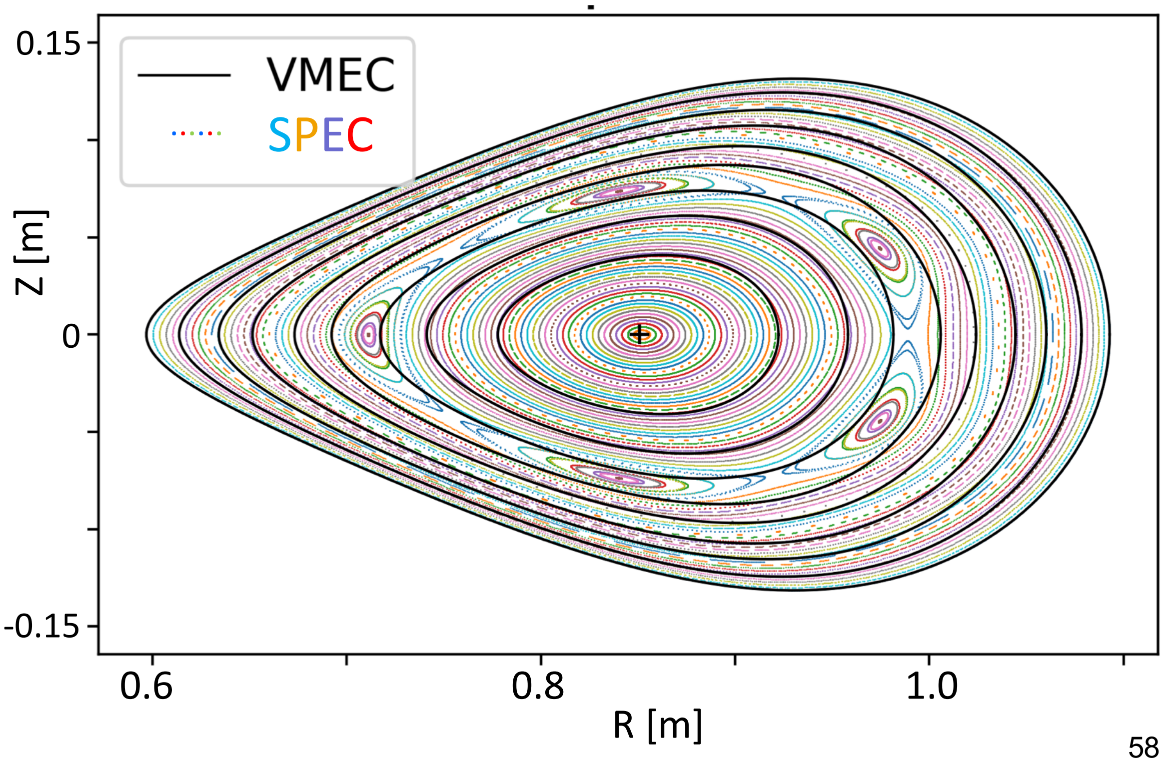
Example: Start with a configuration that has islands





nfp = 2, decent quasi-axisymetry (QA), aspect = 6,

β = 0, island chain at ι = 2/5 = 0.4



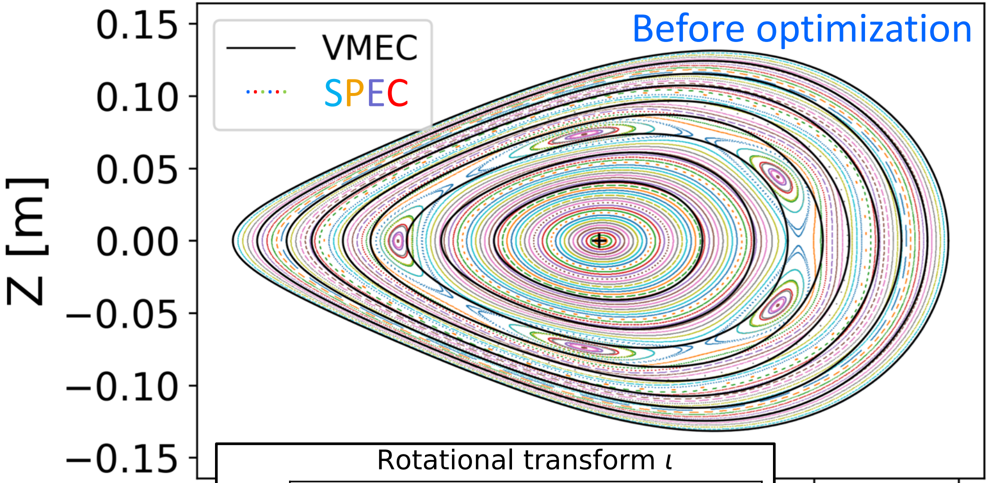
Simsopt driver script applied:

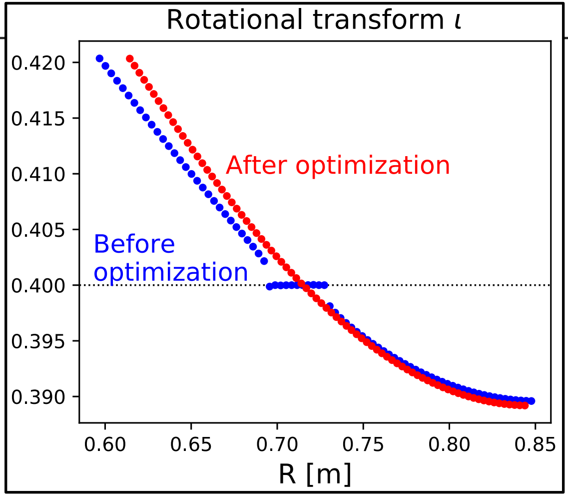
SPEC told to use the same boundary surface object as VMEC.

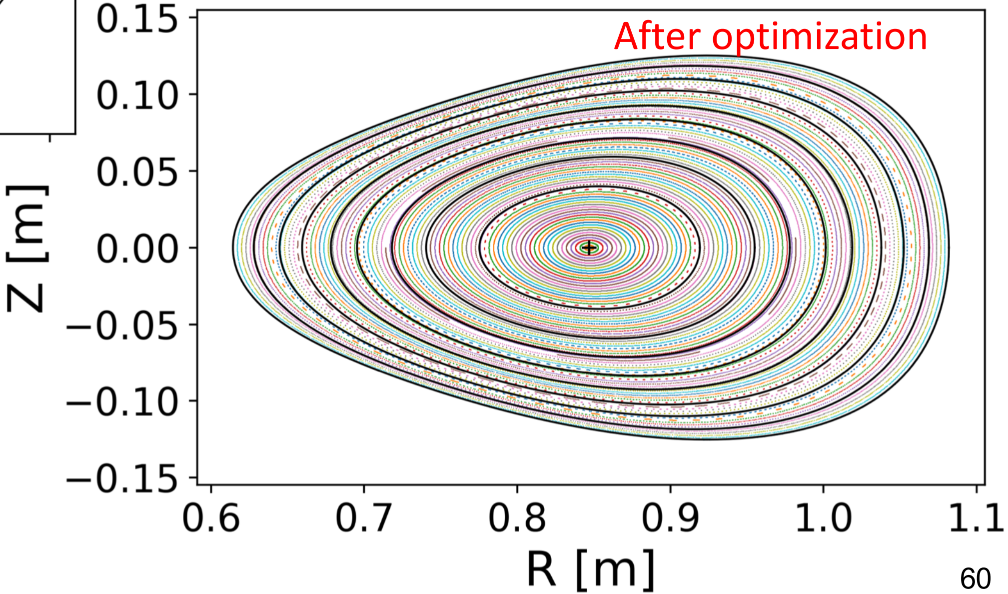
```
mpi = MpiPartition()
vmec = Vmec("input.nfp2_QA”, mpi)
surf = vmec.boundary

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)
```

Objective function includes both quasisymmetry from VMEC and residues from SPEC.

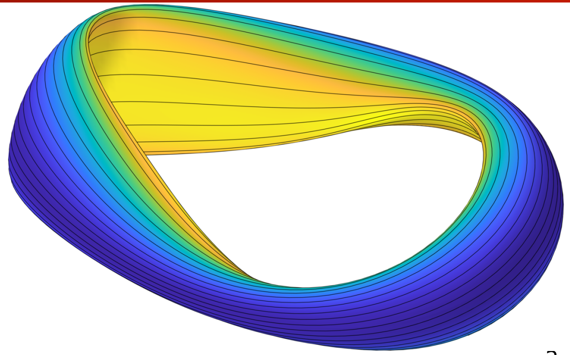
The optimization eliminates the islands

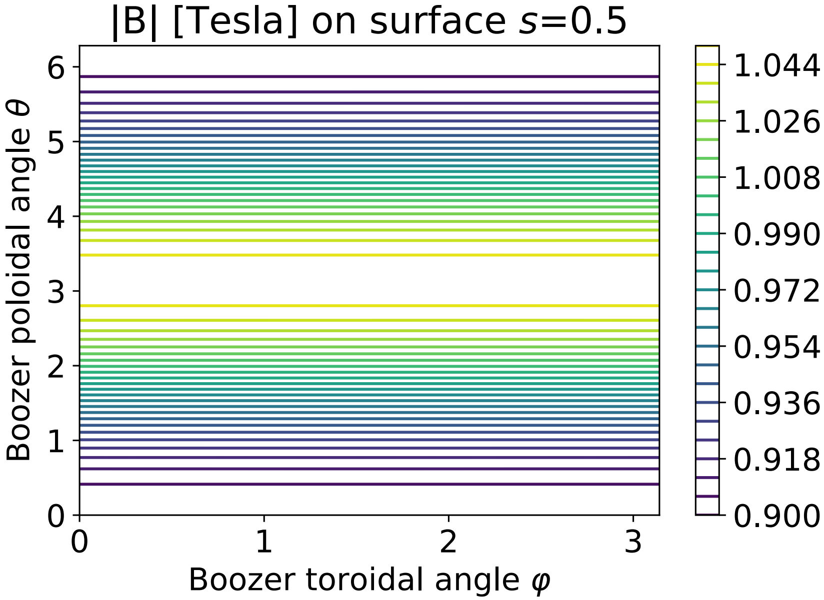


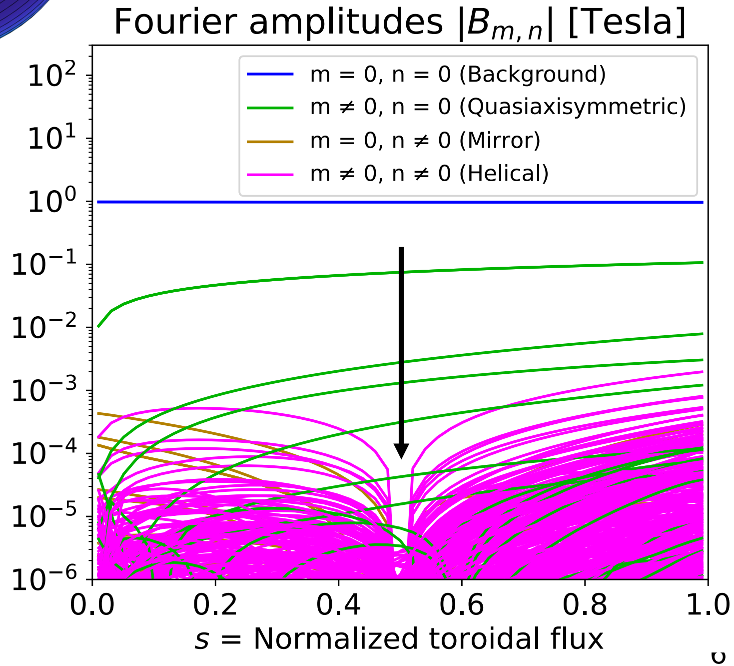




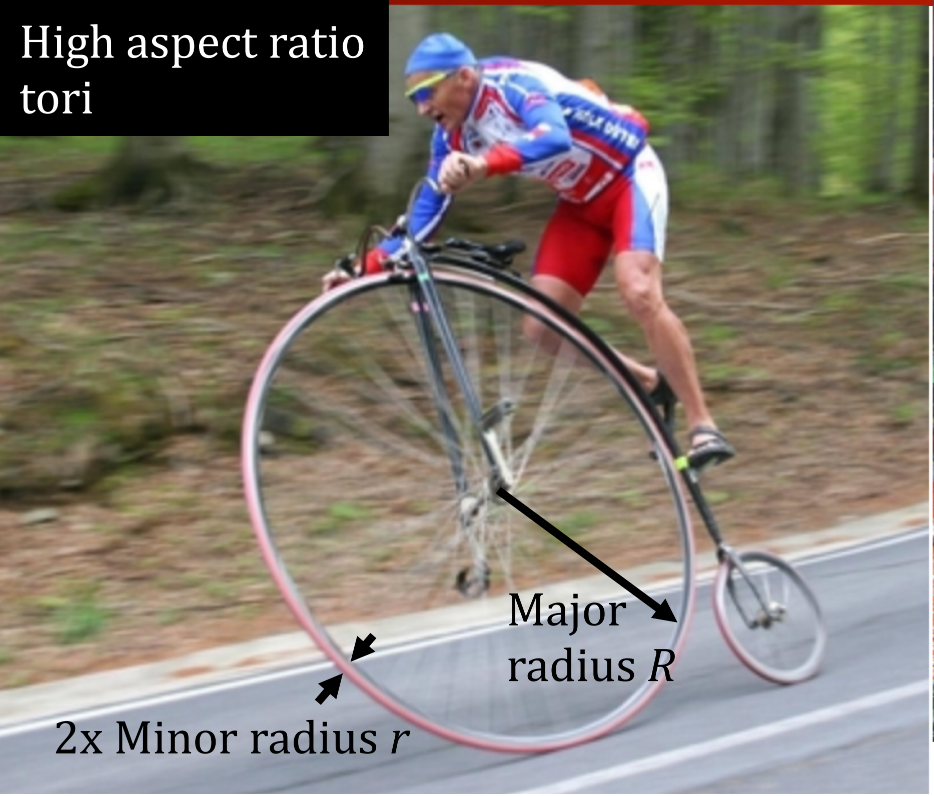
Quasisymmetry is simultaneously improved during the optimization







Expansion about the magnetic axis reduces 3D PDE -> 1D ODEs



$$\frac{r}{\text{radius of curvature of axis}} \ll 1 \qquad \text{Aspect ratio} = \frac{R}{r}$$