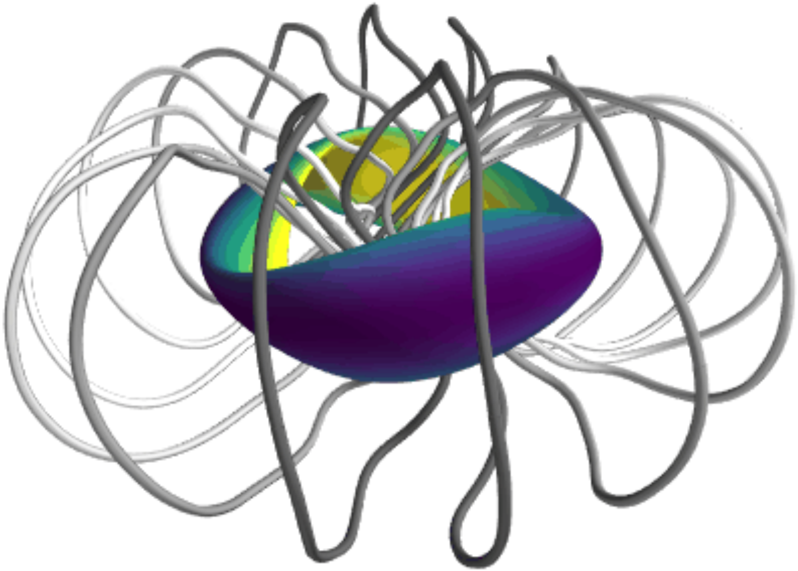
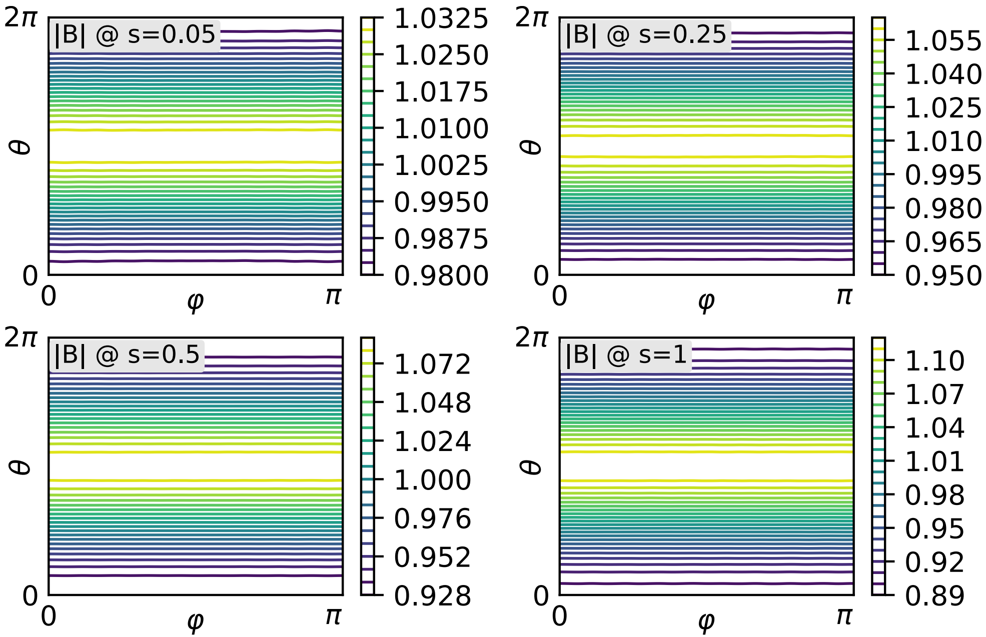
Innovations in stellarator optimization for quasisymmetry





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

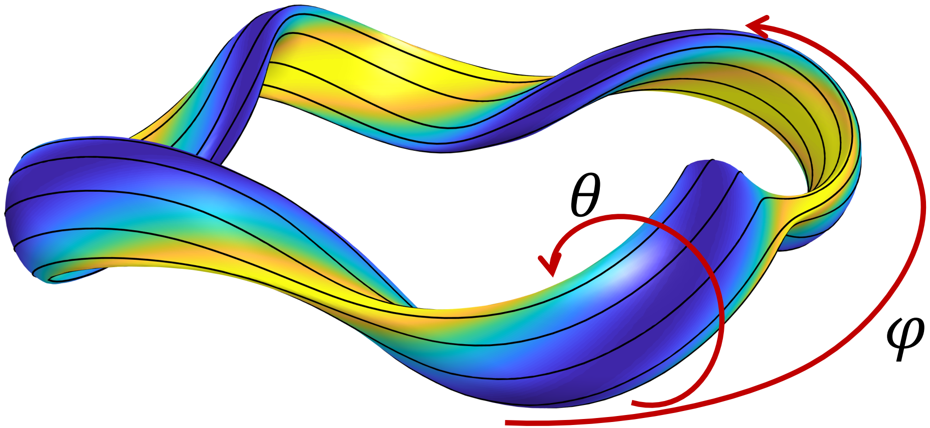
[a] University of Maryland, [b] New York University, [c] Max Planck Institute for Plasma Physics, [d] Princeton Plasma Physics Laboratory

Advantages of stellarators: steady-state, no disruptions, no Greenwald density limit, no power recirculated for current drive.

But,

- Good flux surfaces are not guaranteed.
- Alpha losses & neoclassical transport would be too large unless you carefully choose the geometry.

A solution: quasisymmetry



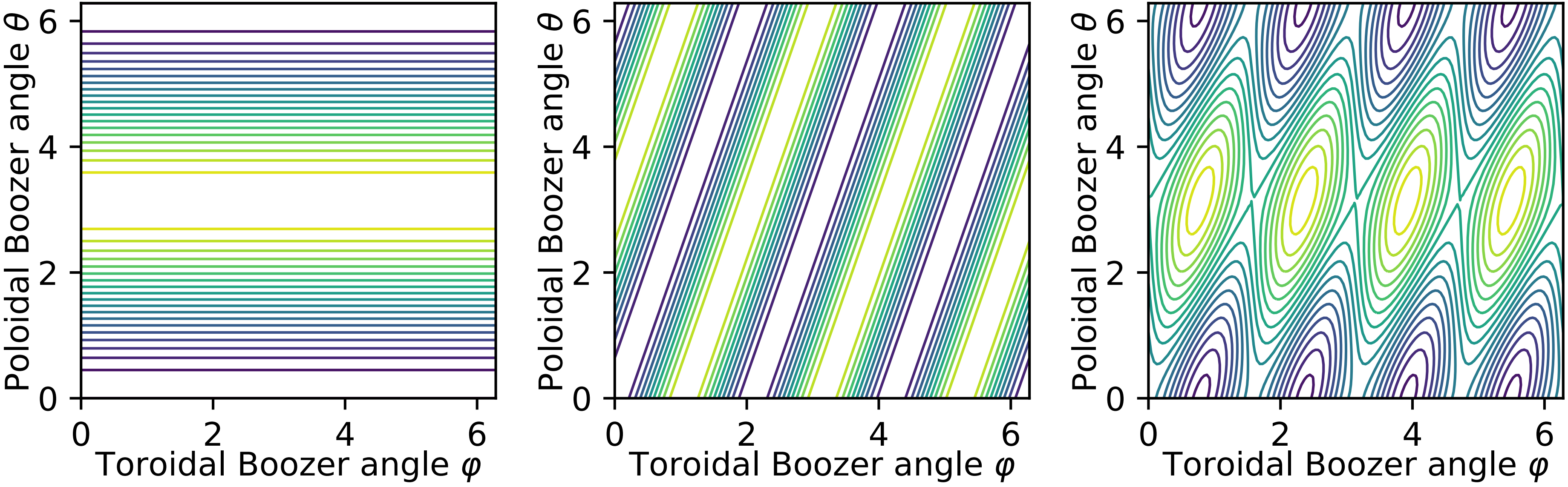
$$B = B(r, \theta - N\varphi)$$

Boozer angles

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

2 types of quasisymmetry

Quasi-axisymmetry (QA): $B = B(r, \theta)$	Quasi-helical symmetry (QH): $B = B(r, \theta - N\varphi)$	General stellarator (not symmetric)



Contours of $B = |\mathbf{B}|$: B_{min} — B_{max}

- Combined optimization for good flux surfaces & other features

  ML, Medasani, & Zhu, Phys Plasmas (2021)

- Precise quasisymmetry

  ML & Paul, arXiv:2108.03711 (2021)

Both use new stellarator optimization framework SIMSOPT:
https://github.com/hiddenSymmetries/simsopt
ML, Medasani, Wechsung, et al, J Open Source Software (2021)

- Combined optimization for good flux surfaces & other features

  ML, Medasani, & Zhu, Phys Plasmas (2021)

- Precise quasisymmetry

  ML & Paul, arXiv:2108.03711 (2021)

Both use new stellarator optimization framework SIMSOPT:
https://github.com/hiddenSymmetries/simsopt
ML, Medasani, Wechsung, et al, J Open Source Software (2021)

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]

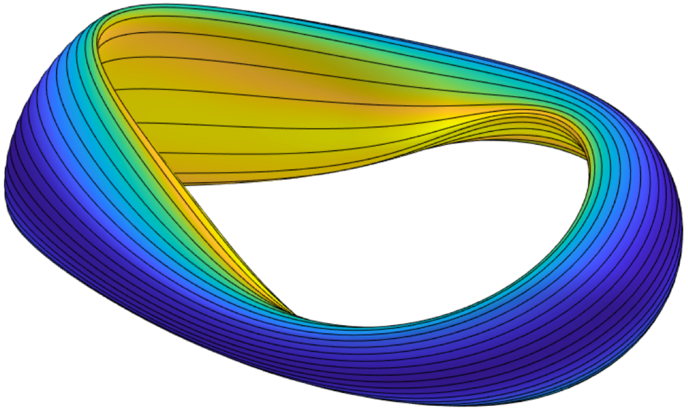
[1] Hirshman & Whitson, *Phys. Fluids* (1993)

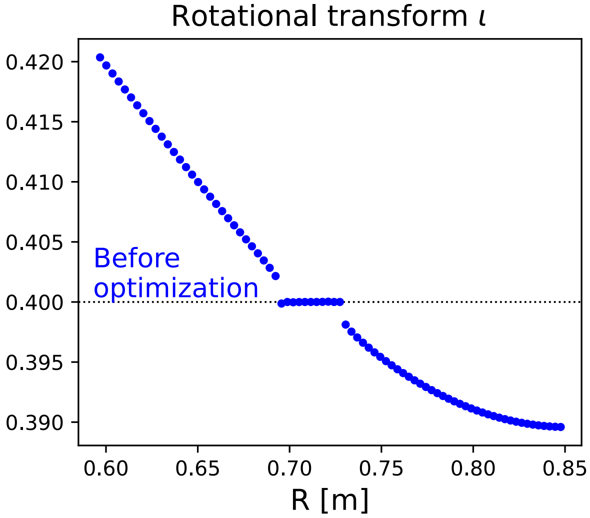
[2] Hudson, Dewar, et al, *Phys. Plasmas* (2012)

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

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

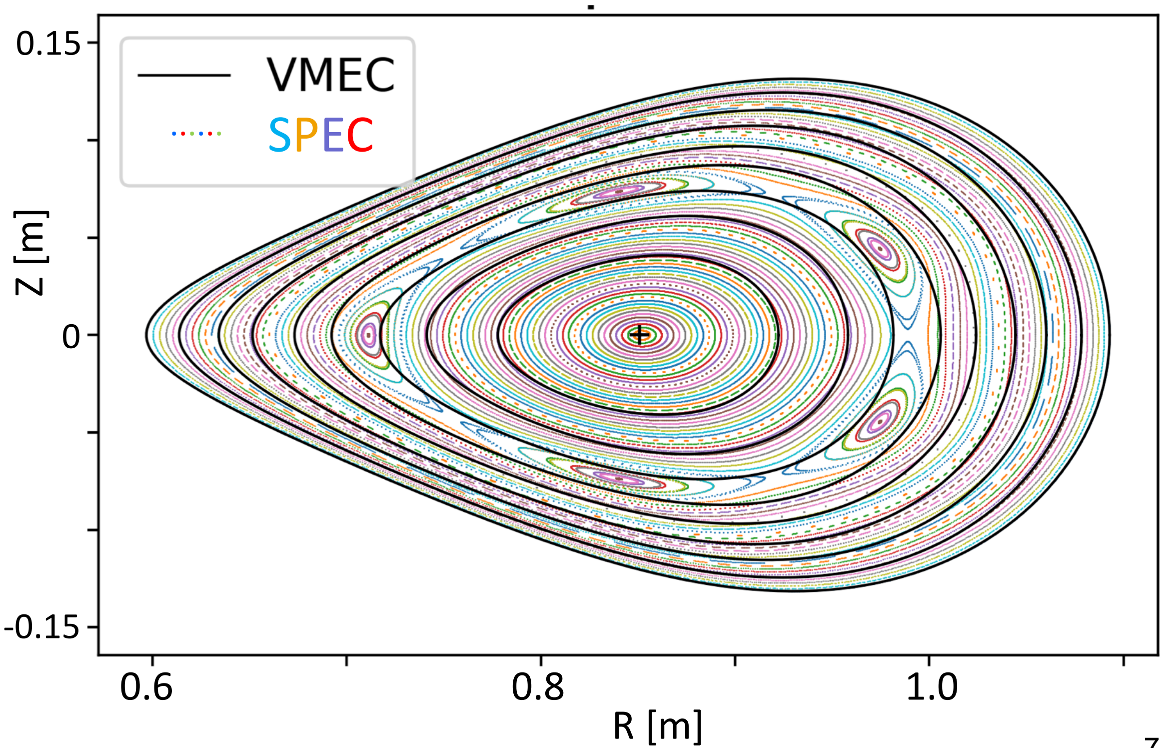
Example: Start with a configuration that has islands





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

β = 0, island chain at ι = 2/5 = 0.4



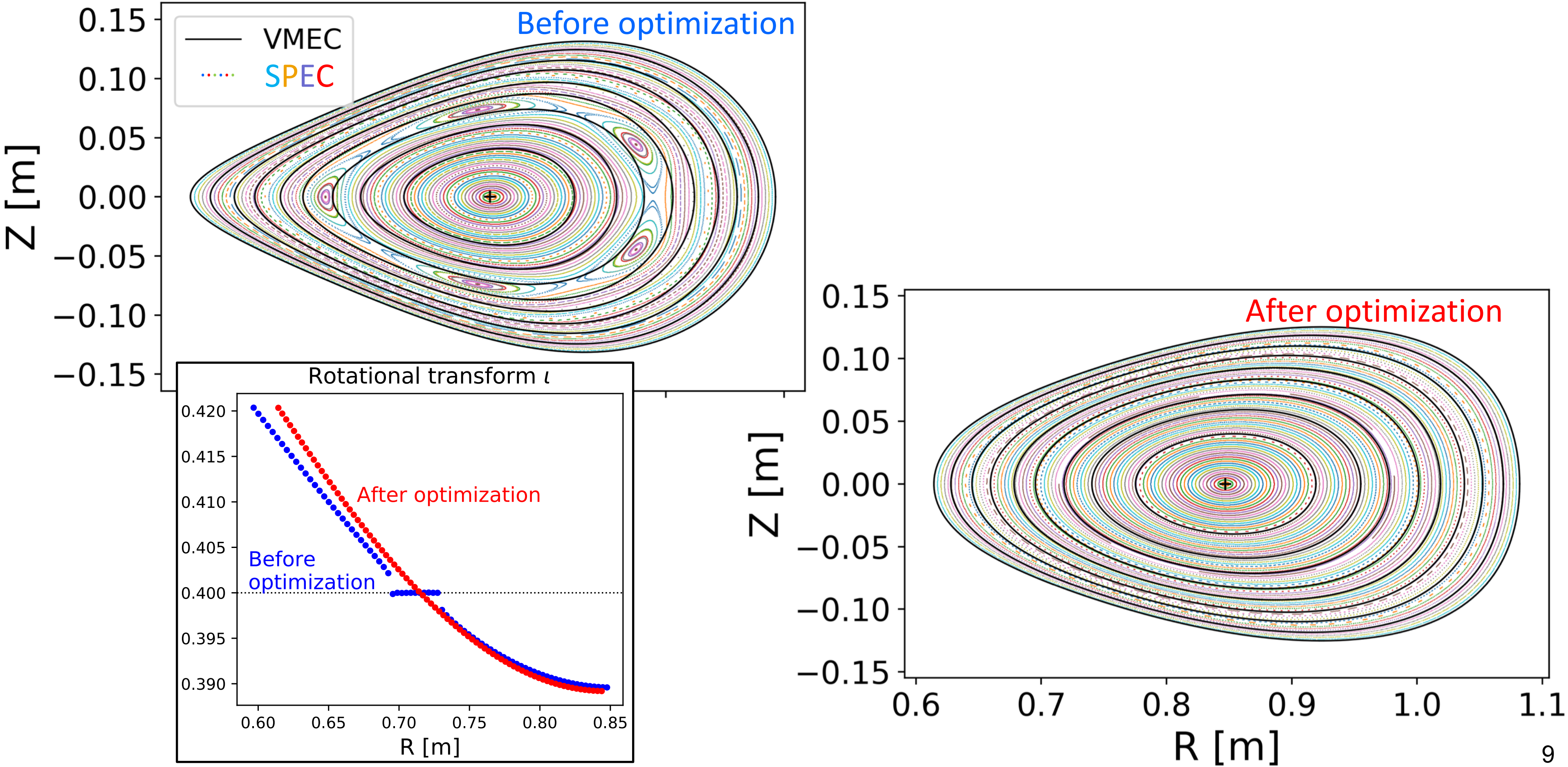
Simsopt driver script applied:

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

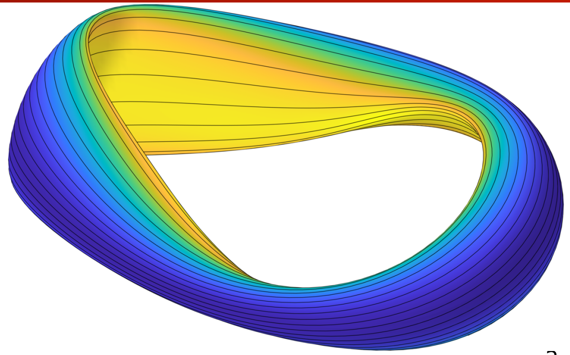
```
mpi = MpiPartition()
vmec = Vmec("input.nfp2_QA", mpi)
surf = vmec.boundary

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)
```

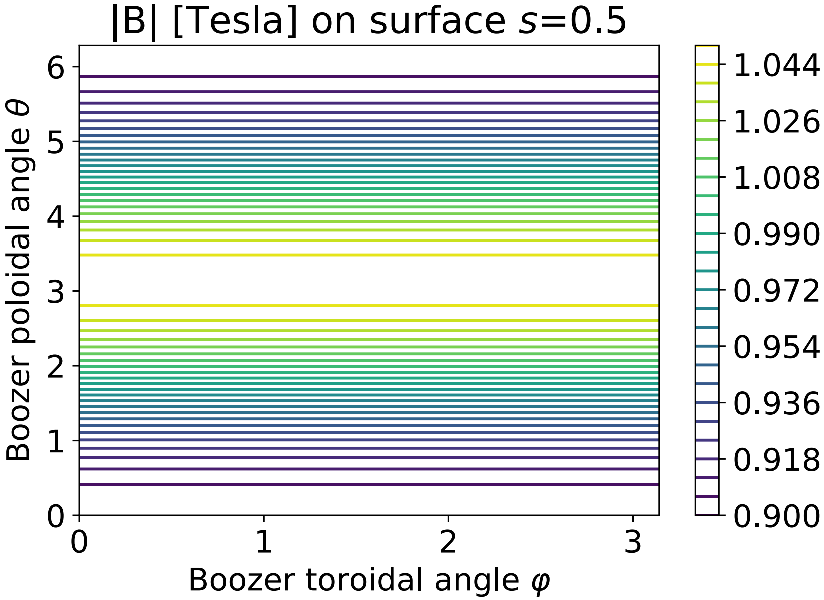
Objective function includes both quasisymmetry from VMEC and residues from SPEC.

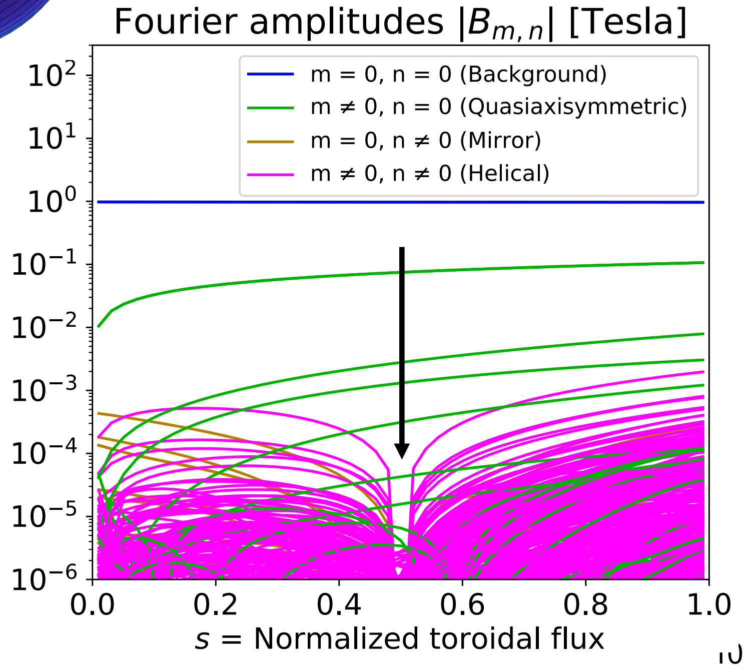
The optimization eliminates the islands



Quasisymmetry is simultaneously improved during the optimization







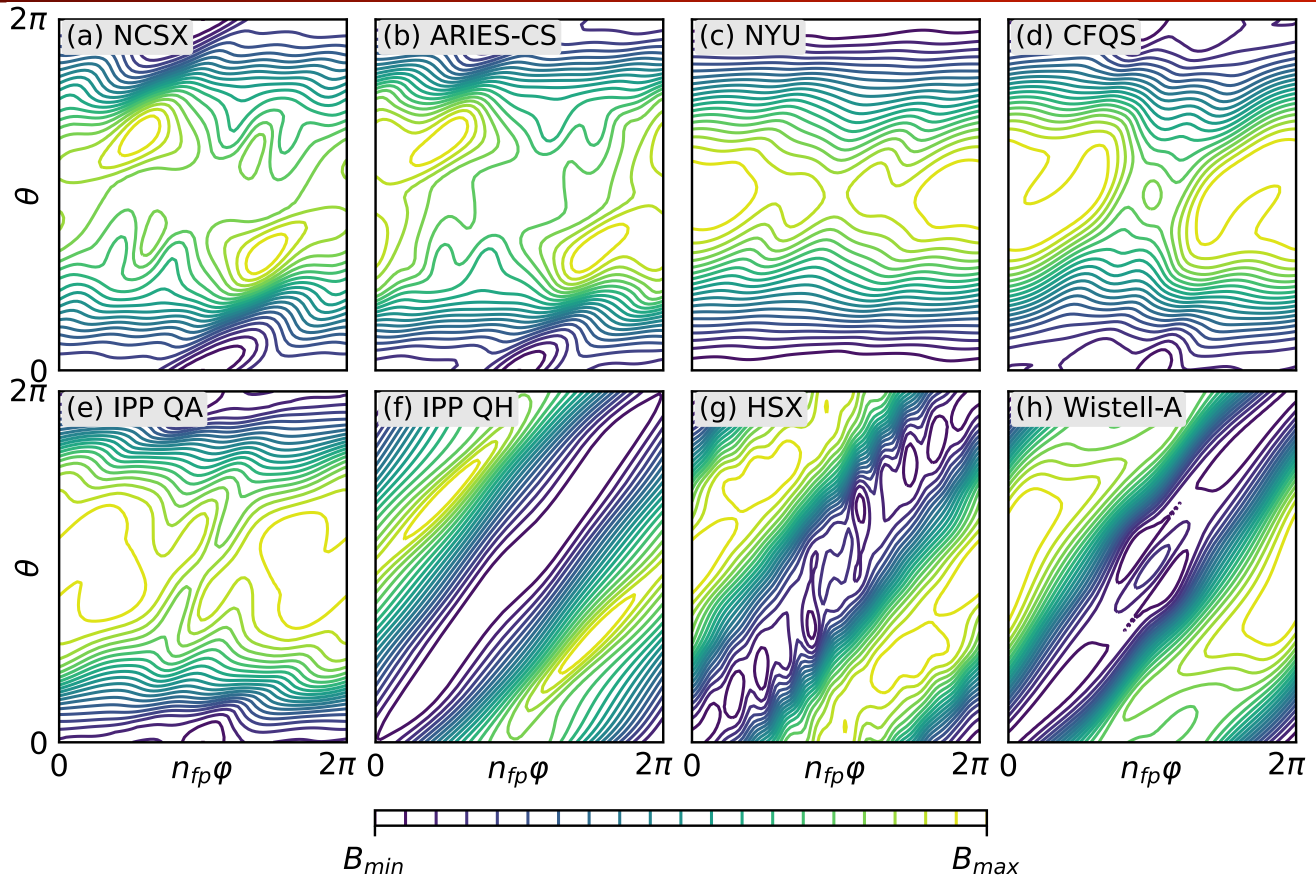
- Combined optimization for good flux surfaces & other features

  ML, Medasani, & Zhu, Phys Plasmas (2021)

- Precise quasisymmetry

  ML & Paul, arXiv:2108.03711 (2021)

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

Can we get |B| contours to look truly straight if we optimize for only quasisymmetry?

Optimization problem

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

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

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

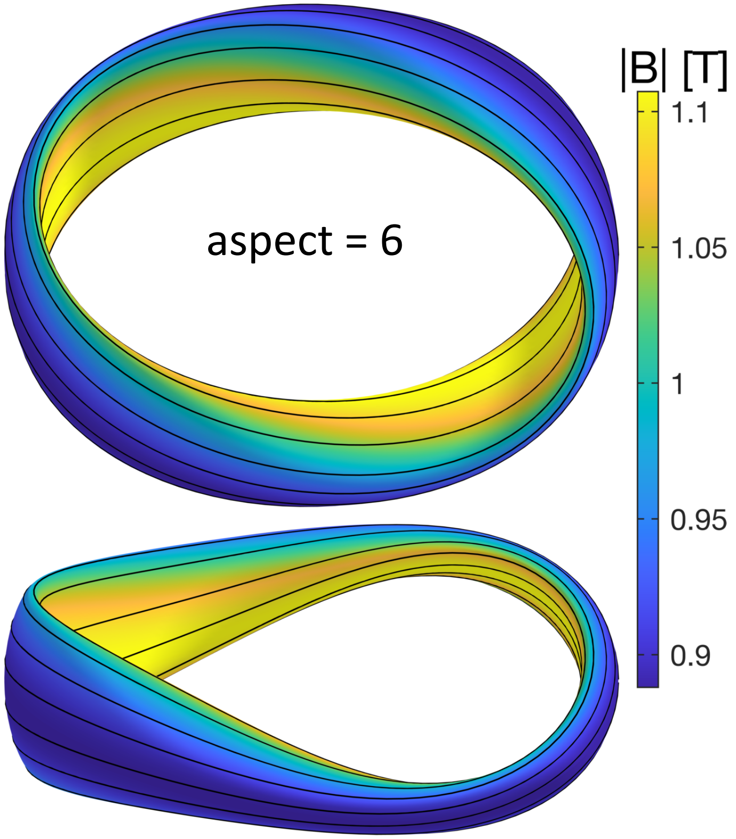
Boundary aspect ratio

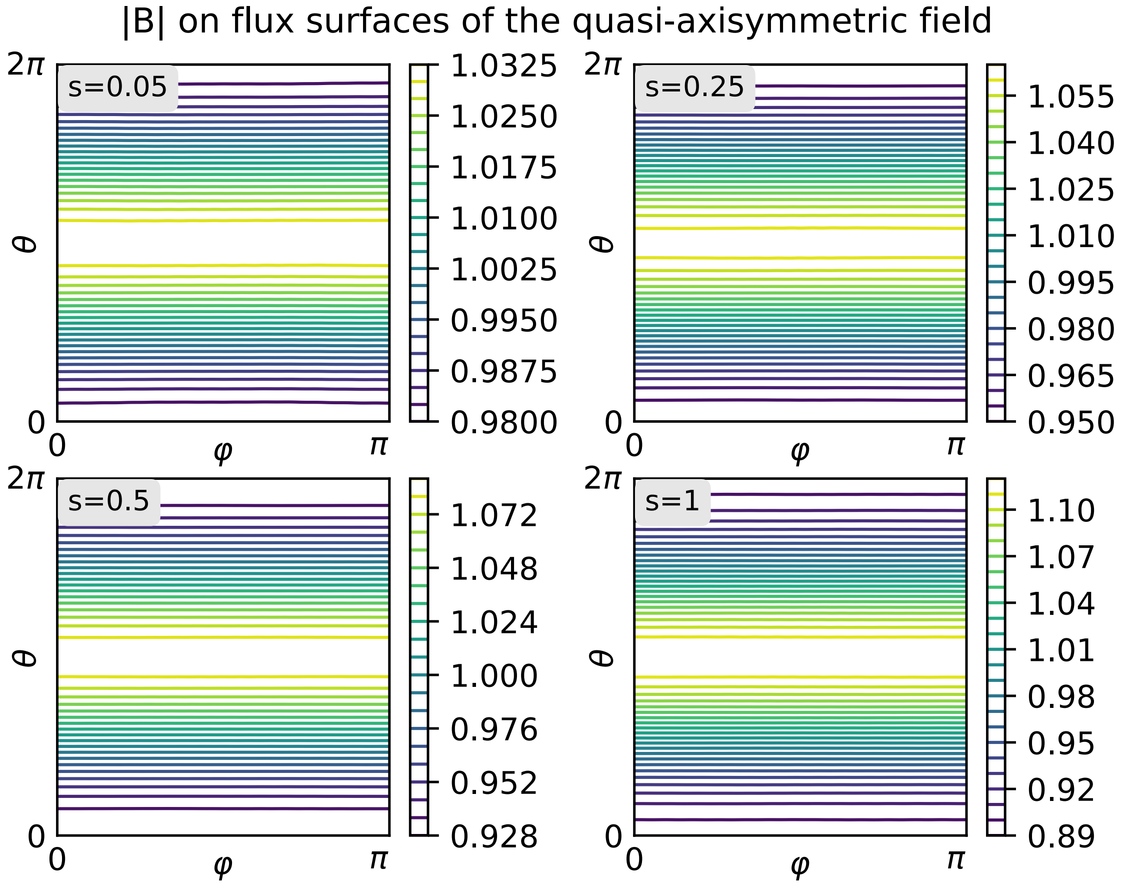
- The usual parameter space: $R_{m,n}$ & $Z_{m,n}$ defining a toroidal boundary

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

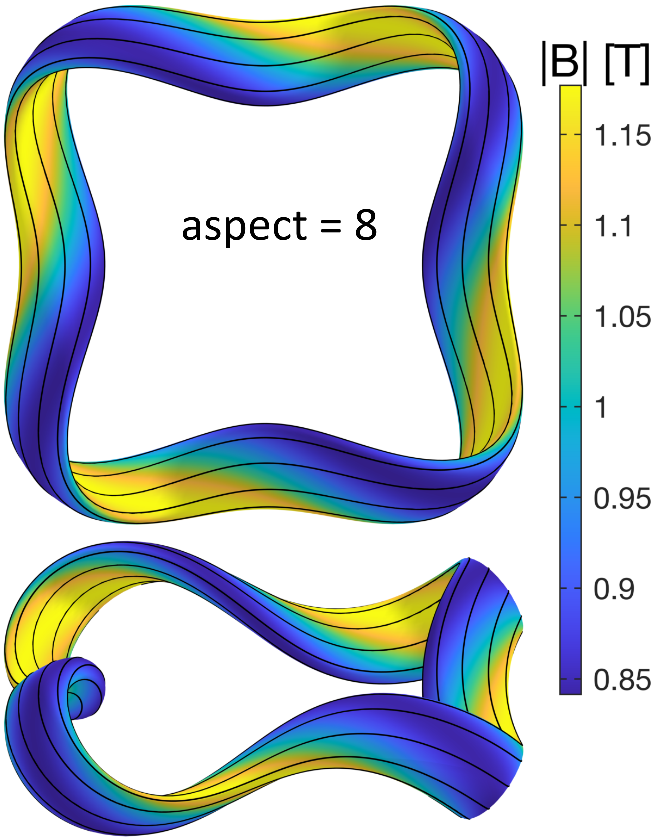
- SIMSOPT with VMEC
- Cold start
- Vacuum fields, allowing precise checks against SPEC & Biot-Savart
- Algorithm: default for least-squares in scipy (trust region reflective)
- 6 stages: increasing # of modes varied & VMEC resolution
- Run many optimizations, pick the best

Straight |B| contours are possible for QA

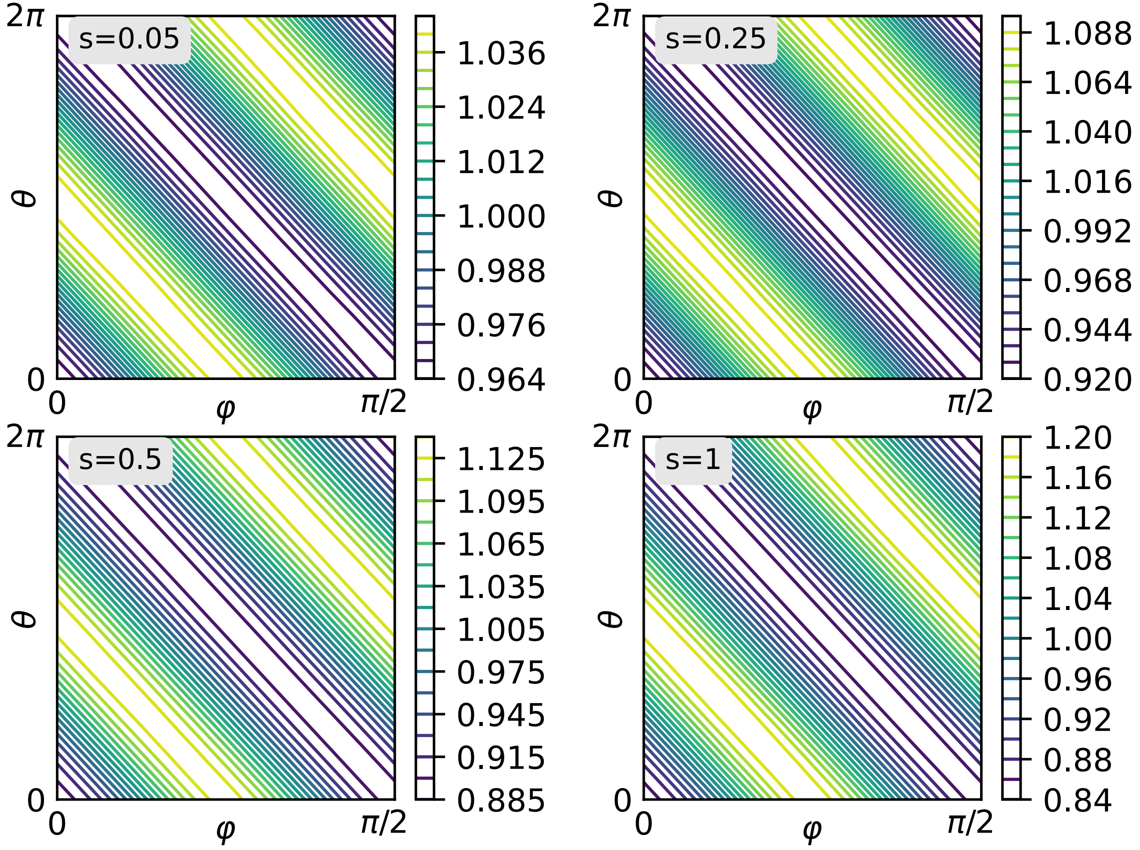




Straight |B| contours are possible for QH

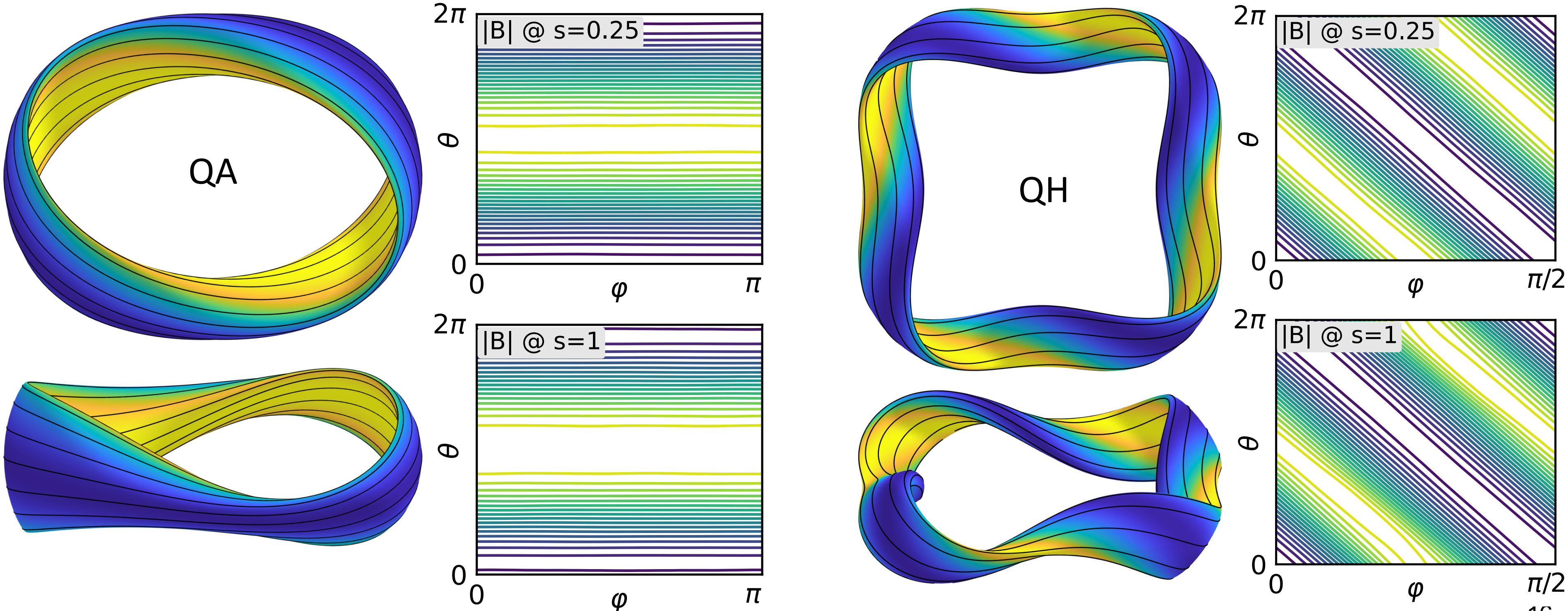


|B| on flux surfaces of the quasi-helically symmetric field

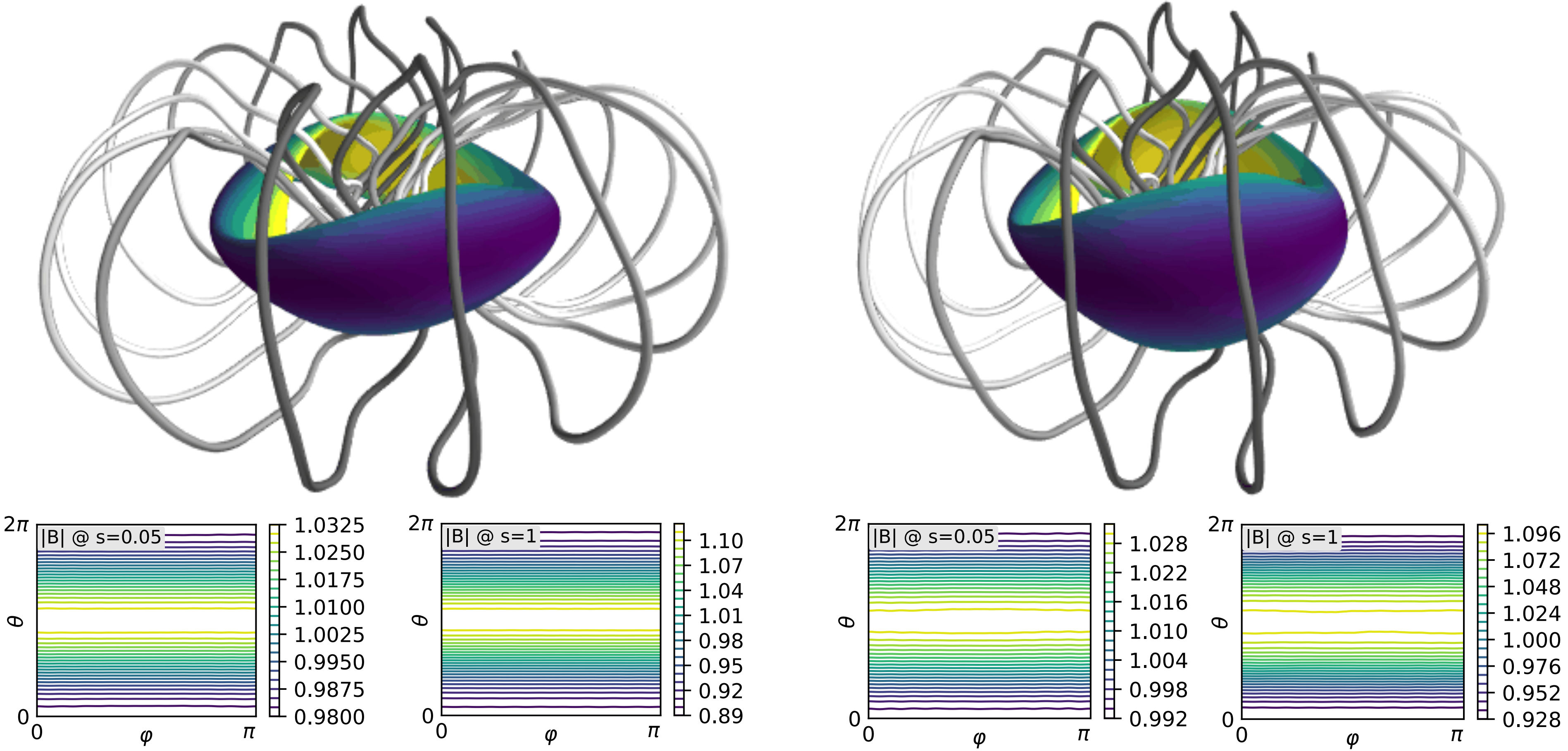


Good symmetry also exists with magnetic well

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



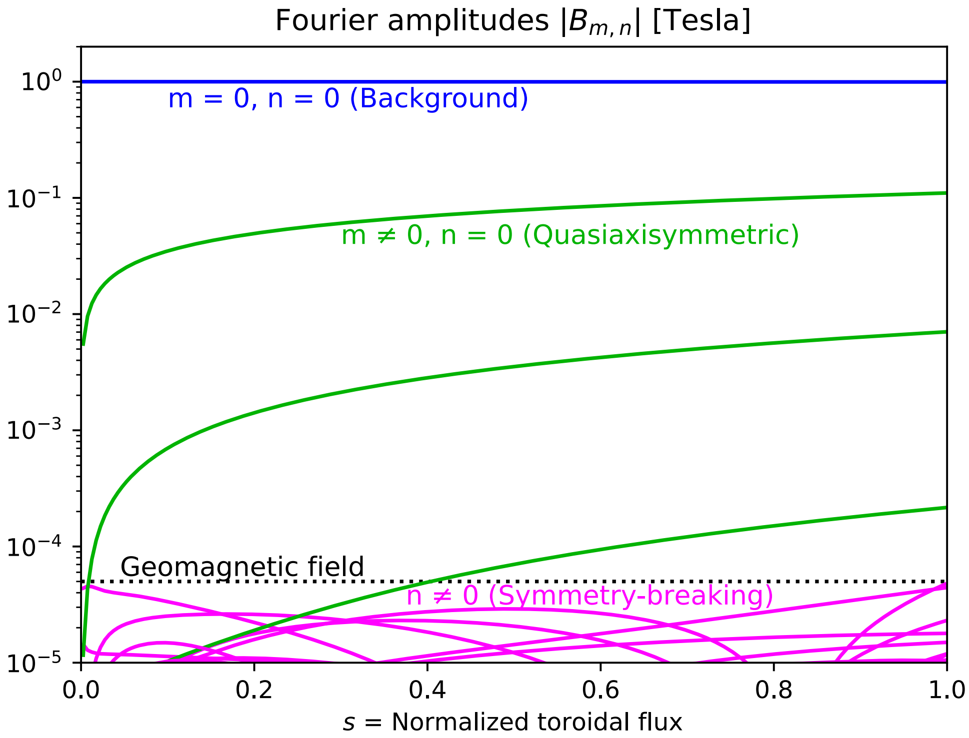
Decent 16-coil solutions have been found for the new QAs

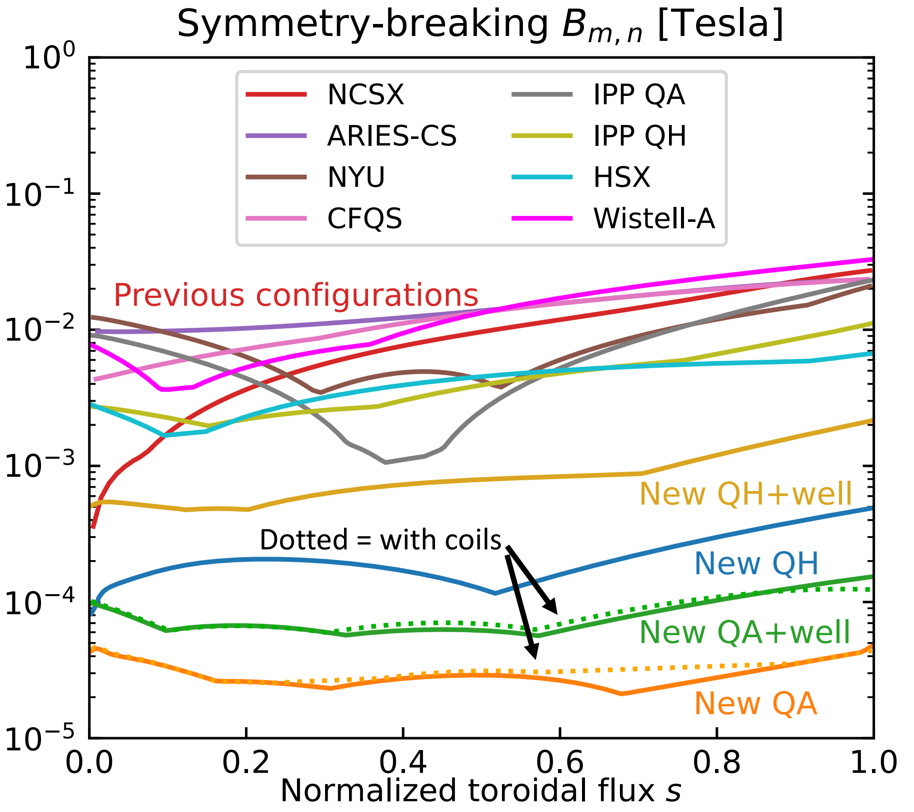


By Florian Wechsung @ NYU. <R>/10 between filament centers. Haven't looked at the QHs yet

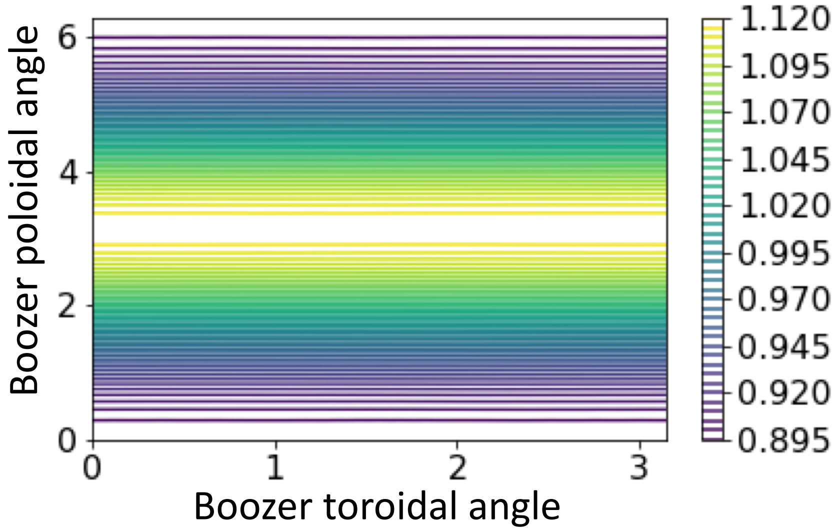
Symmetry-breaking modes can be made extremely small

New QA configuration

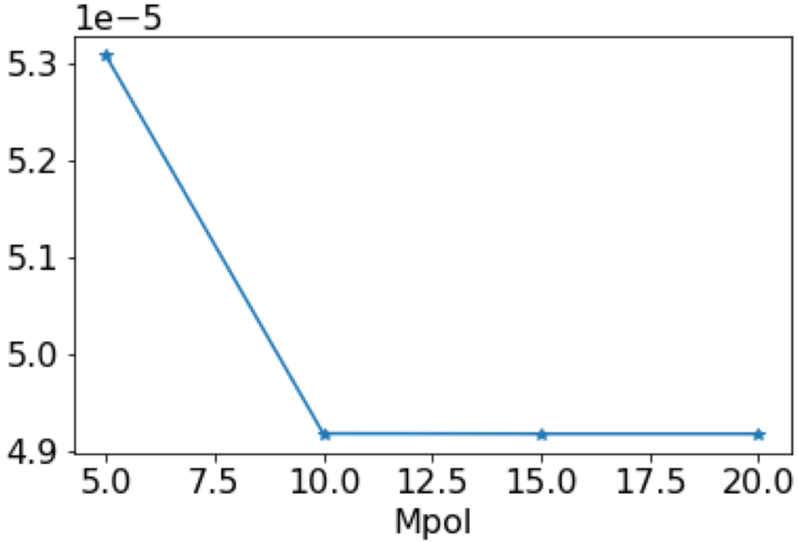




|B| in Boozer coordinates was verified by independent SPEC calculations



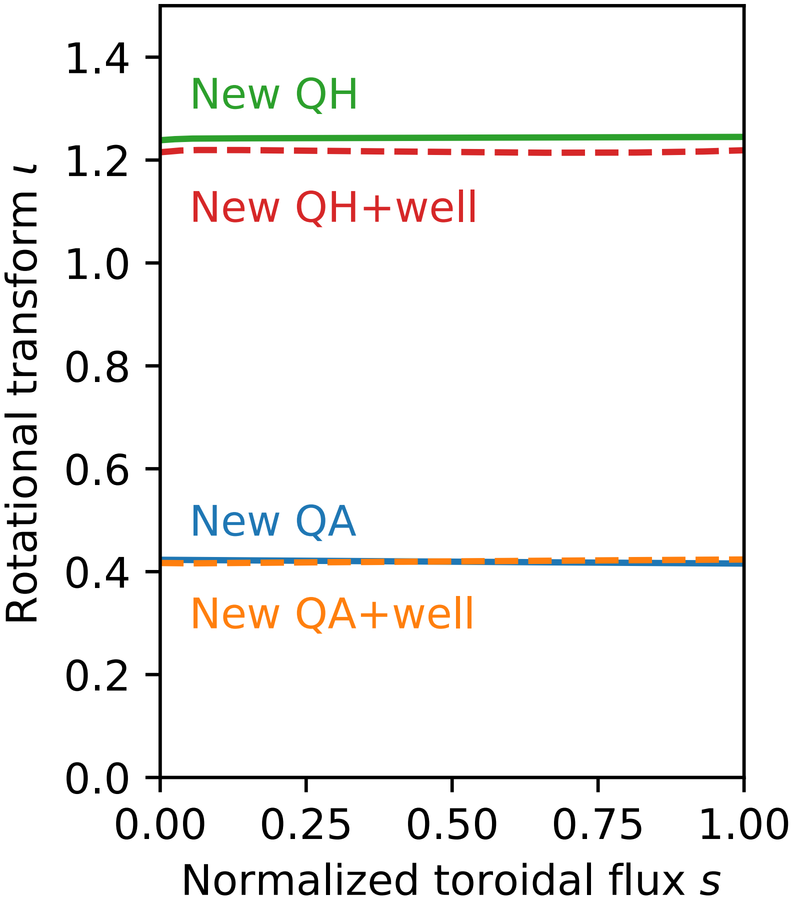
max $|B_{m,n}|$ [Tesla] with n ≠ 0

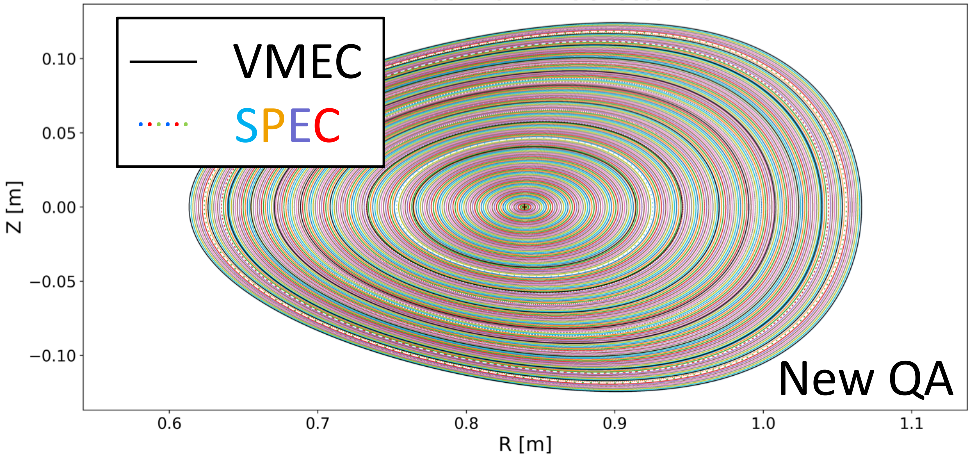


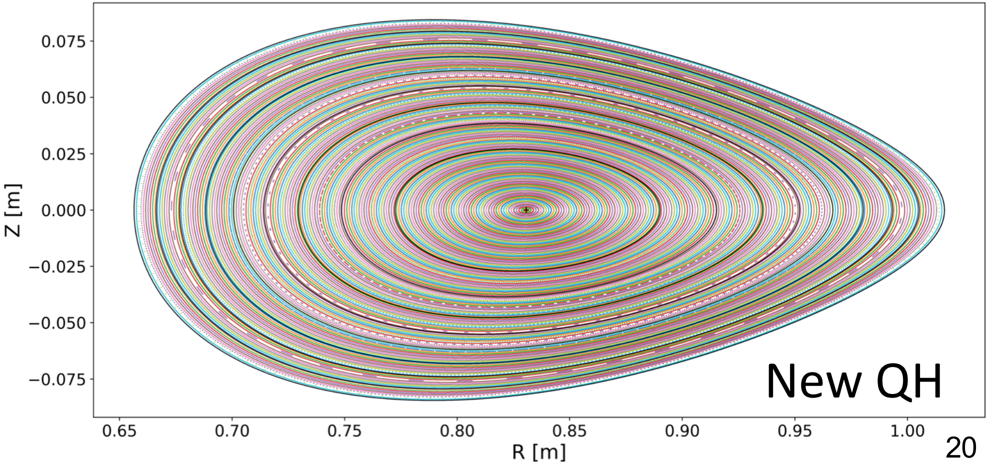
(Ntor = Mpol, Lrad = Mpol + 4)

By Elizabeth Paul

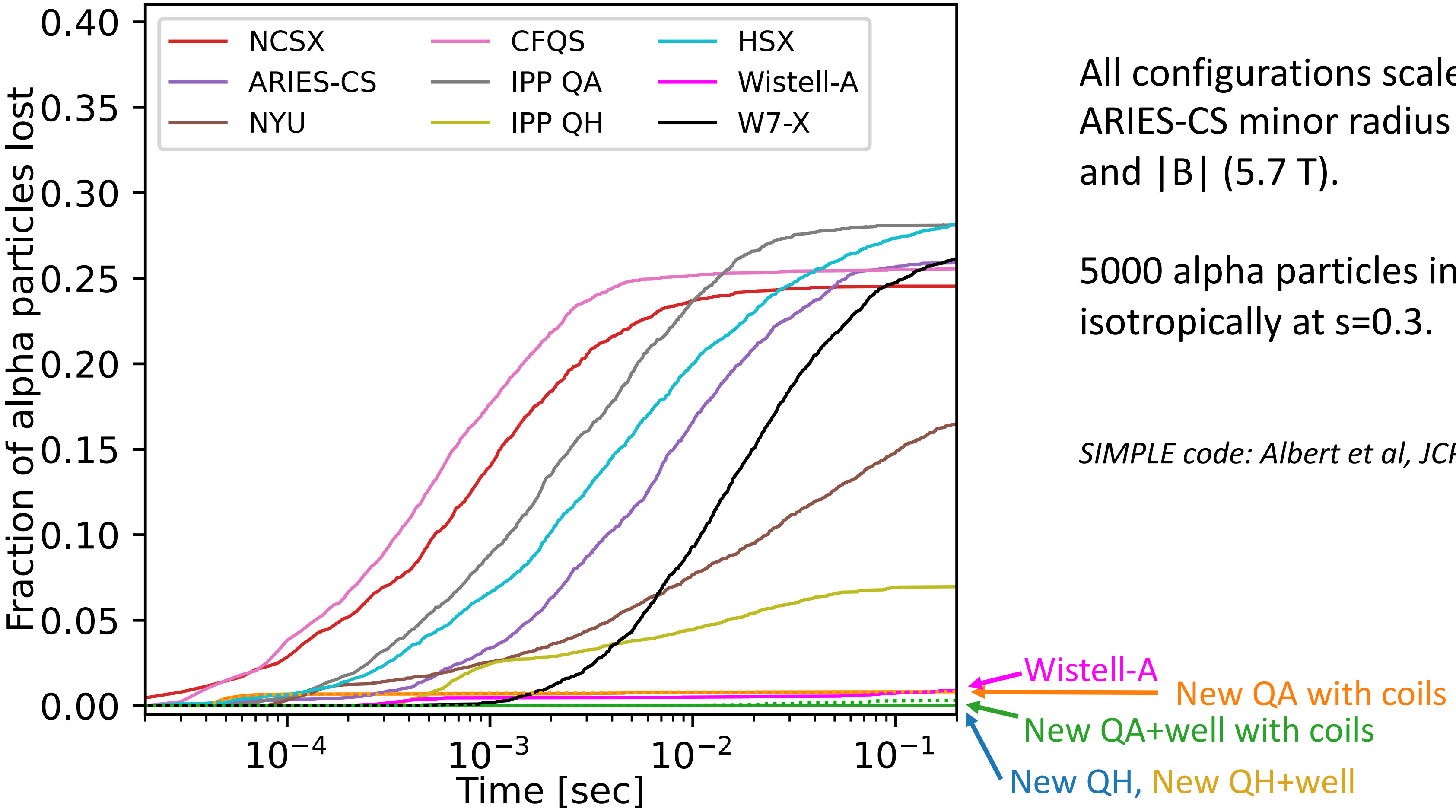
The new configurations have small magnetic shear







The symmetry yields extremely good confinement of collisionless trajectories

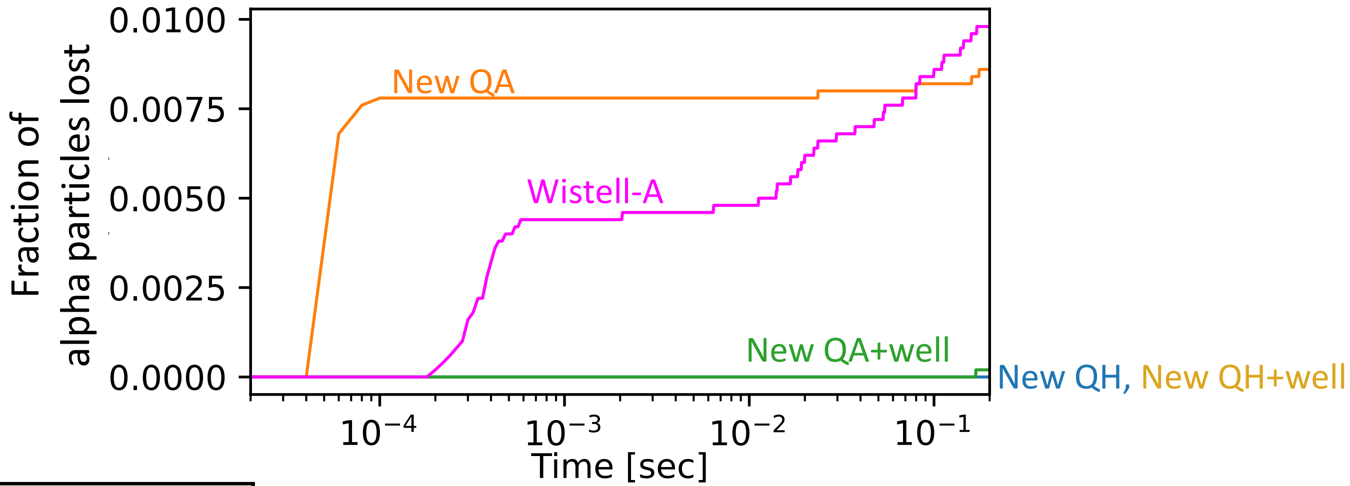


All configurations scaled to ARIES-CS minor radius (1.7 m) and |B| (5.7 T).

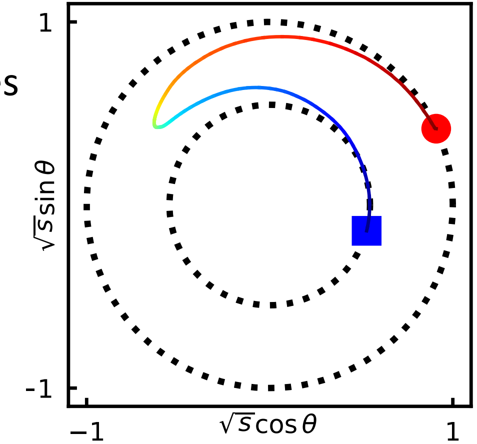
5000 alpha particles initialized isotropically at s=0.3.

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

Why does the configuration with best symmetry not have the best trajectory confinement?



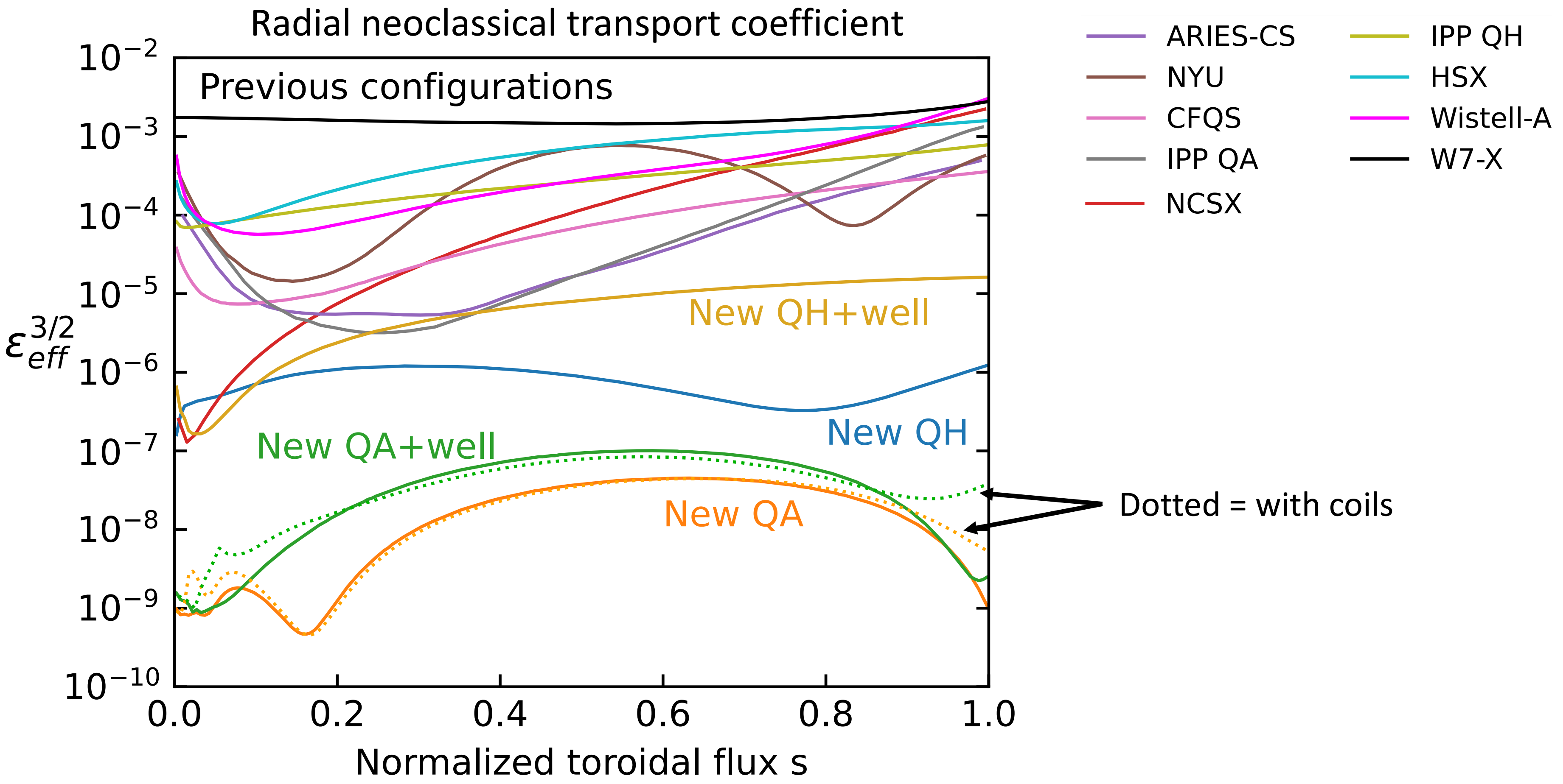
Lost trajectories in the new QA look like this:



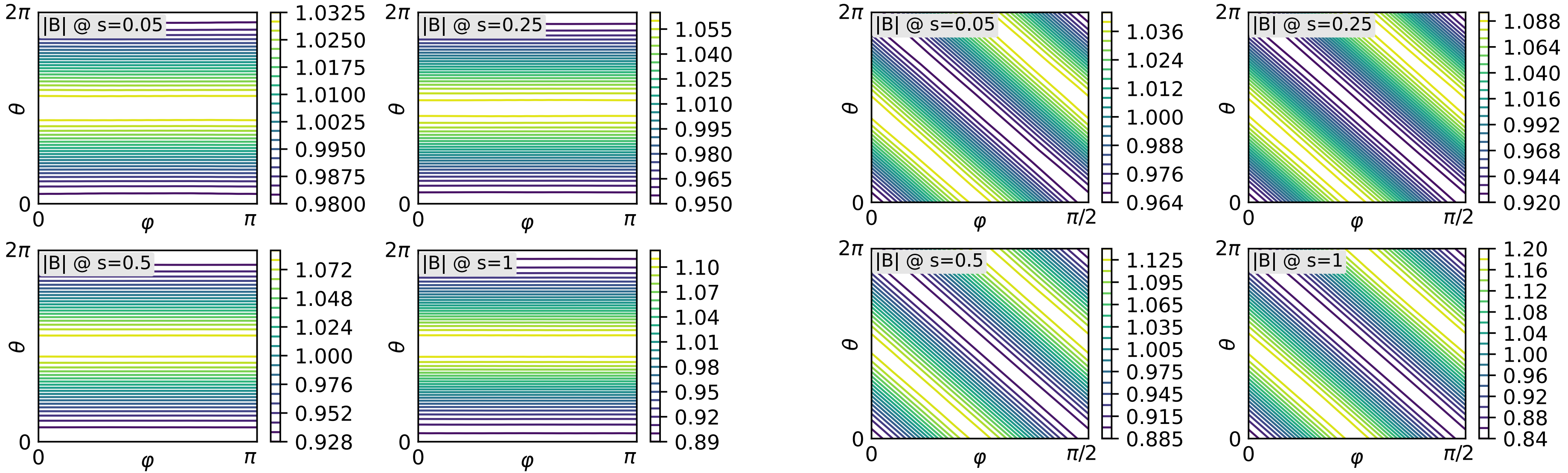
$$\text{Width of banana orbit } \Delta s \approx \left| \frac{mvR\sqrt{2r\bar{\eta}}}{(\iota - N)\psi_{edge} Ze} \right| \propto \frac{1}{|\iota - N|}$$

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

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



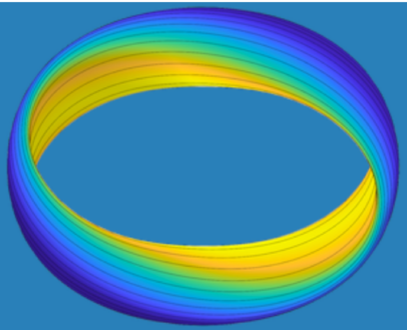
Conclusion: Quasisymmetry can be achieved throughout substantial volumes to high precision, giving excellent confinement



Many questions:

- Do similar solutions exist with substantial plasma pressure & bootstrap current?
- How close can you come to this confinement with additional objectives & constraints?
- What are the trade-offs for symmetry vs aspect ratio, iota, well depth?
- How precisely can you attain omnigenity with poloidally closed B contours?

Extra slides



latest

Search docs

CONTENTS

EXAMPLES

Read the Docs

Simsopt documentation

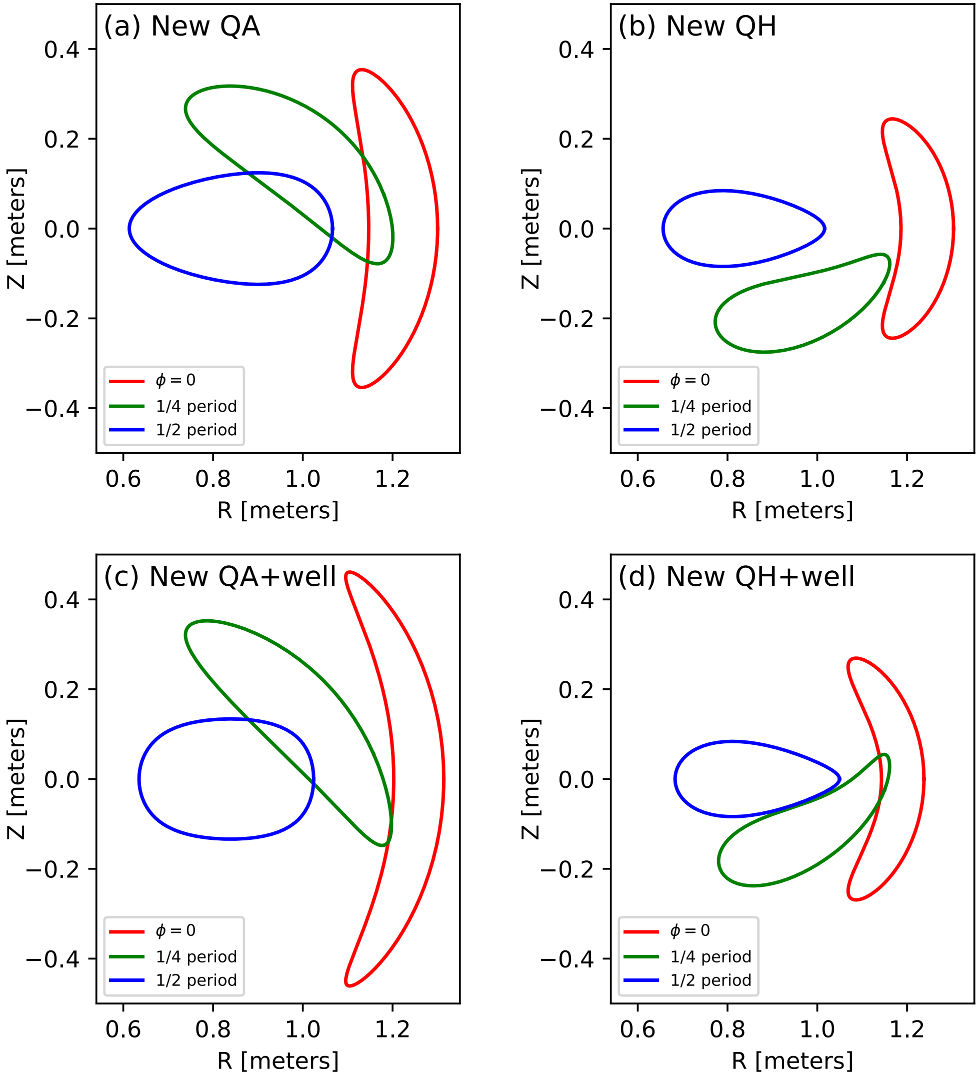
`simsopt` is a framework for optimizing stellarators. The high-level routines are in python, with calls to C++ or fortran where needed for performance. Several types of components are included:

- 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:

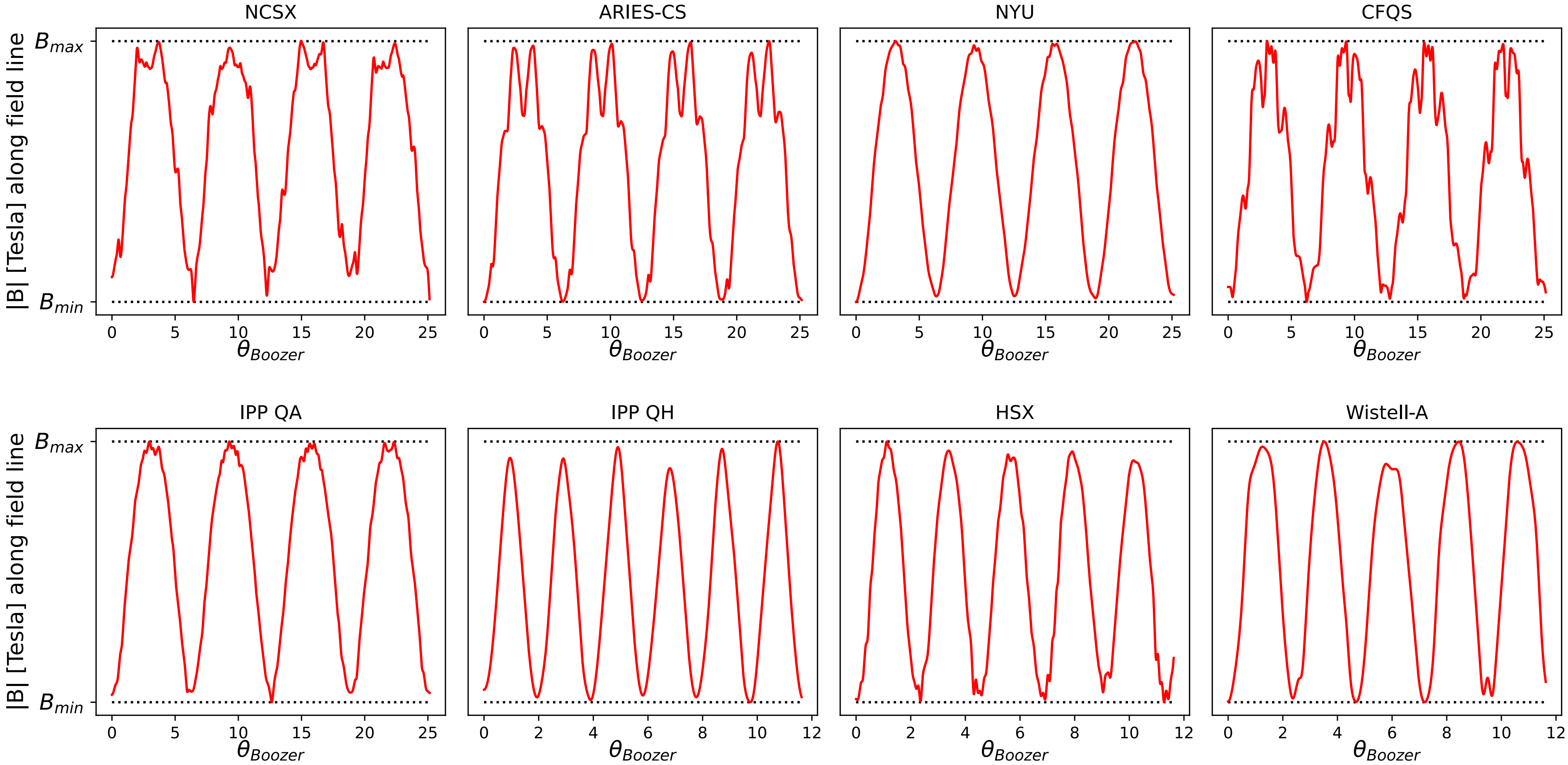
- Thorough unit testing, regression testing, and continuous integration.
- Extensibility: It should be possible to add new codes and terms to the objective function without editing modules that already work, i.e. the open-closed principle . This is because any edits to working code can potentially introduce 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.

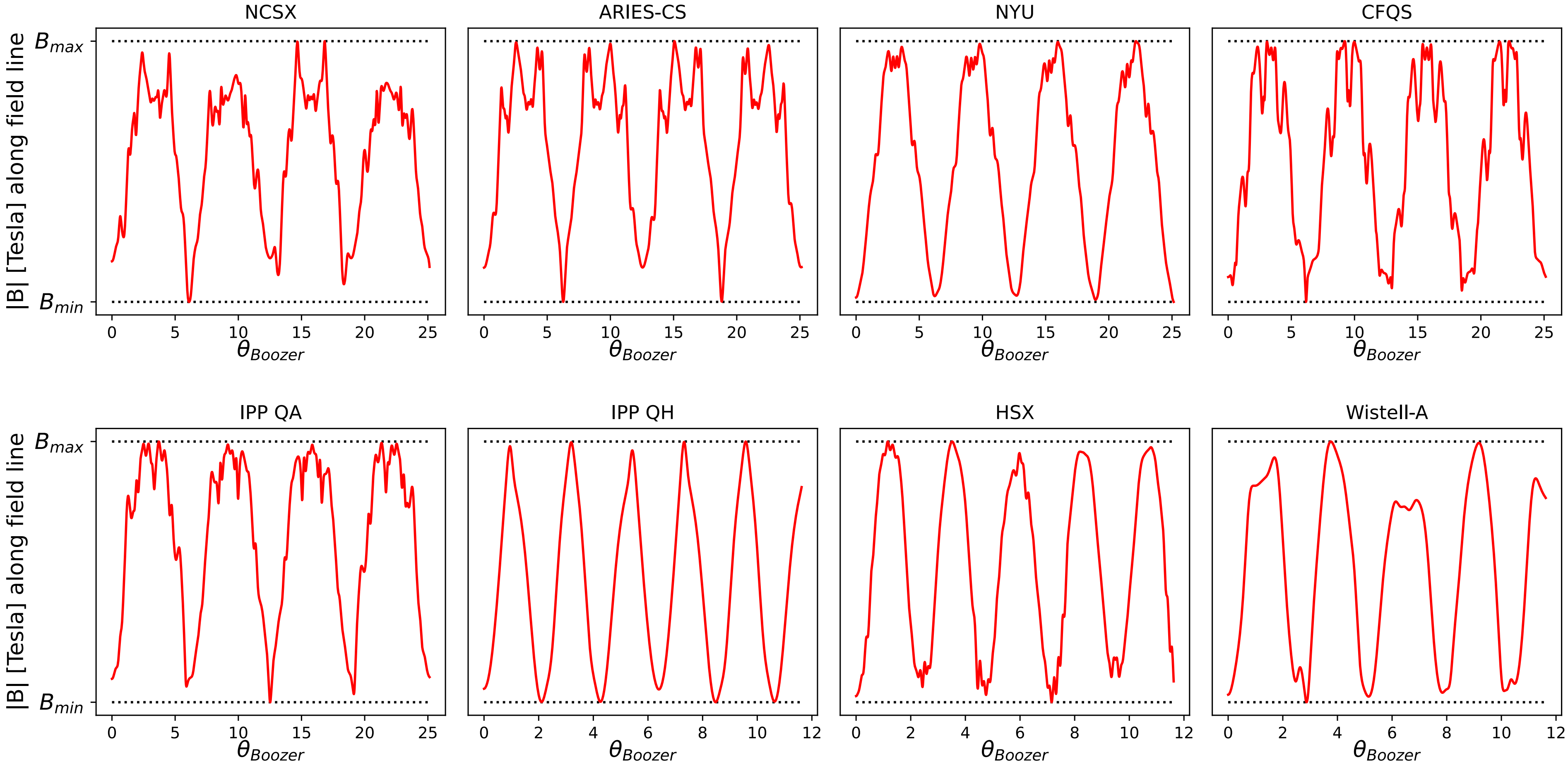


(a) New QA
(b) New QH
(c) New QA+well
(d) New QH+well
Z [meters]
R [meters]
0.4
0.2
0.0
−0.2
−0.4
0.6
0.8
1.0
1.2
$\phi = 0$
1/4 period
1/2 period

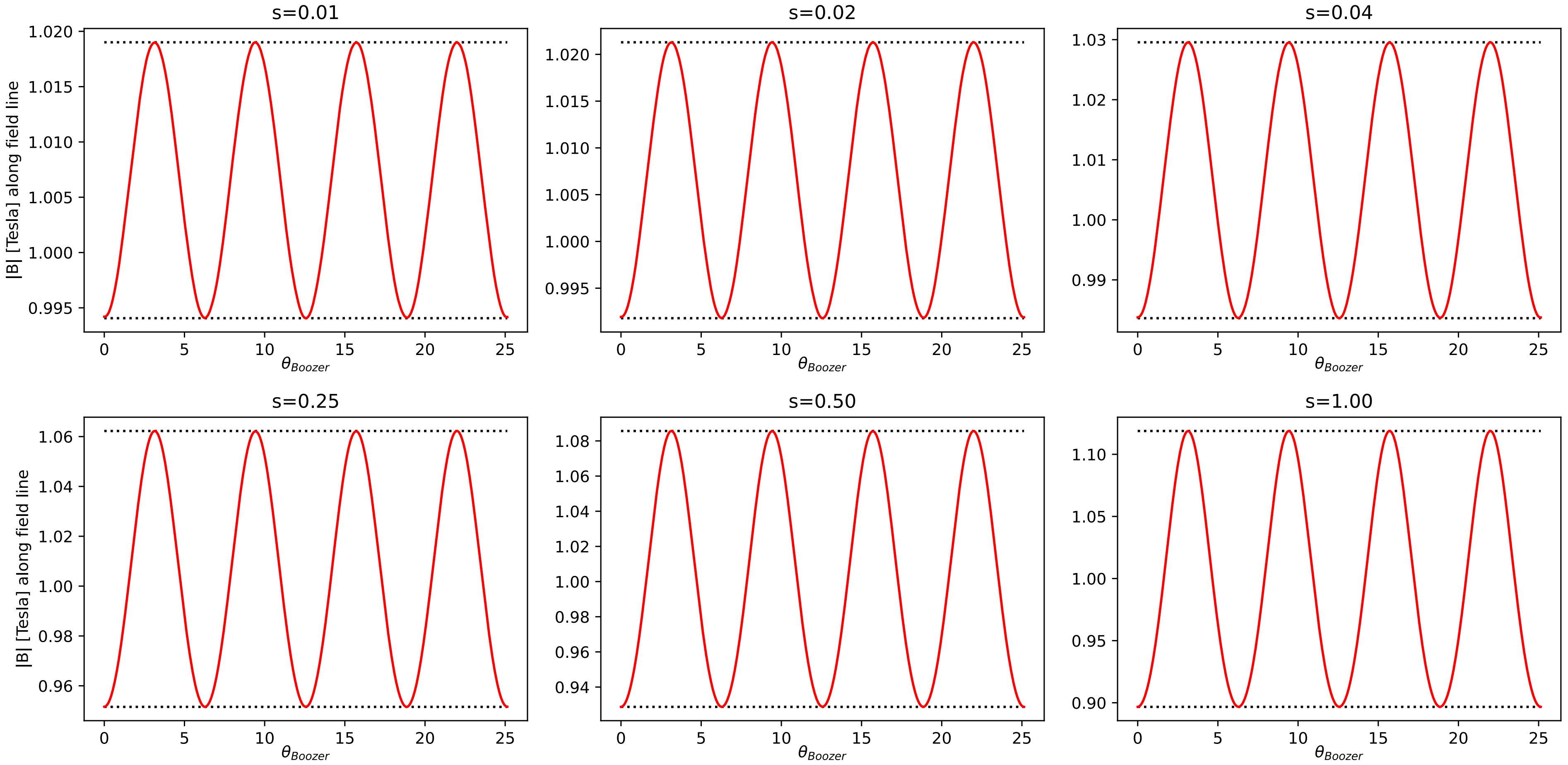
Previous quasisymmetric configurations (s=0.5)



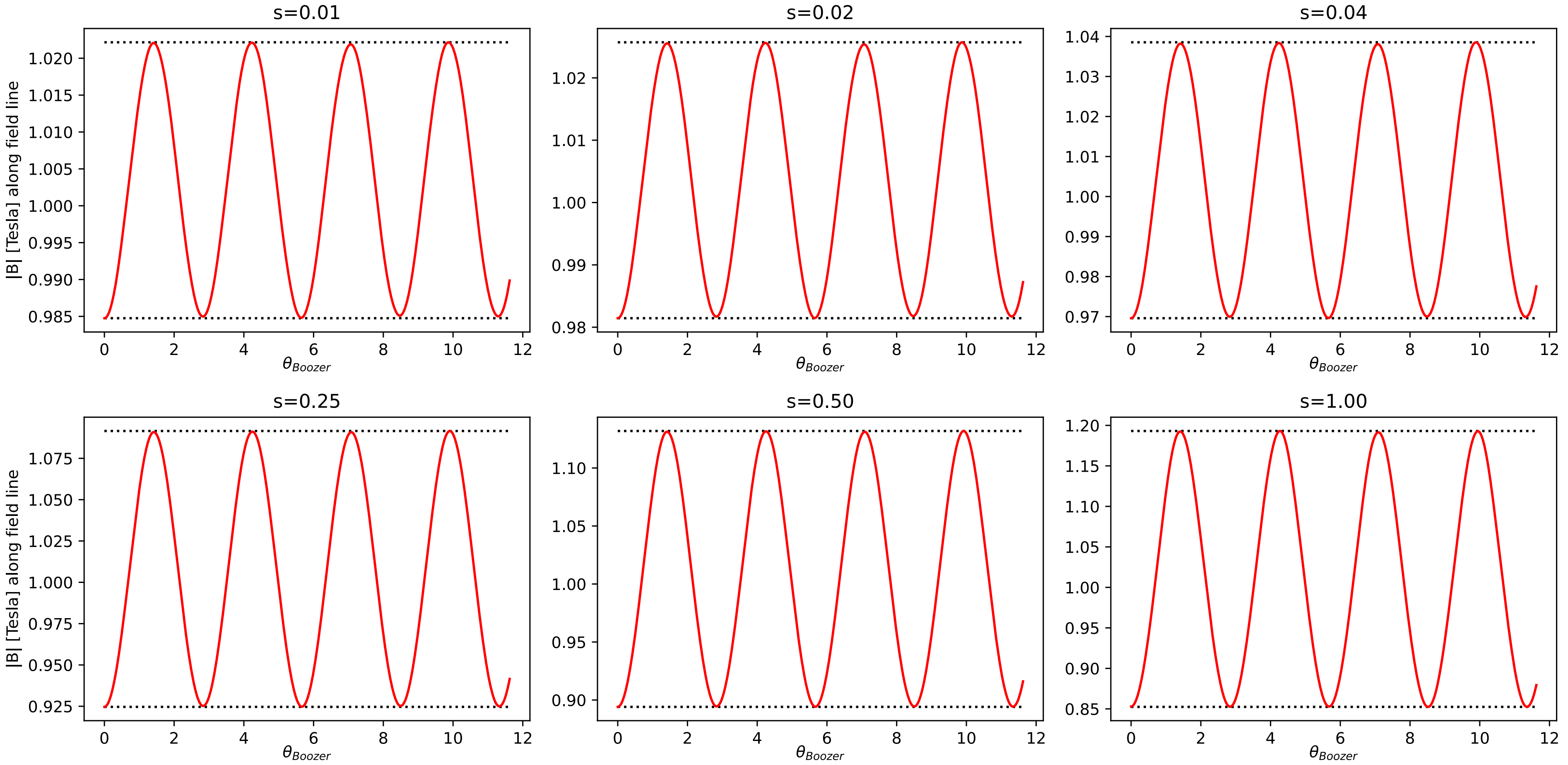
Previous quasisymmetric configurations (s=1)



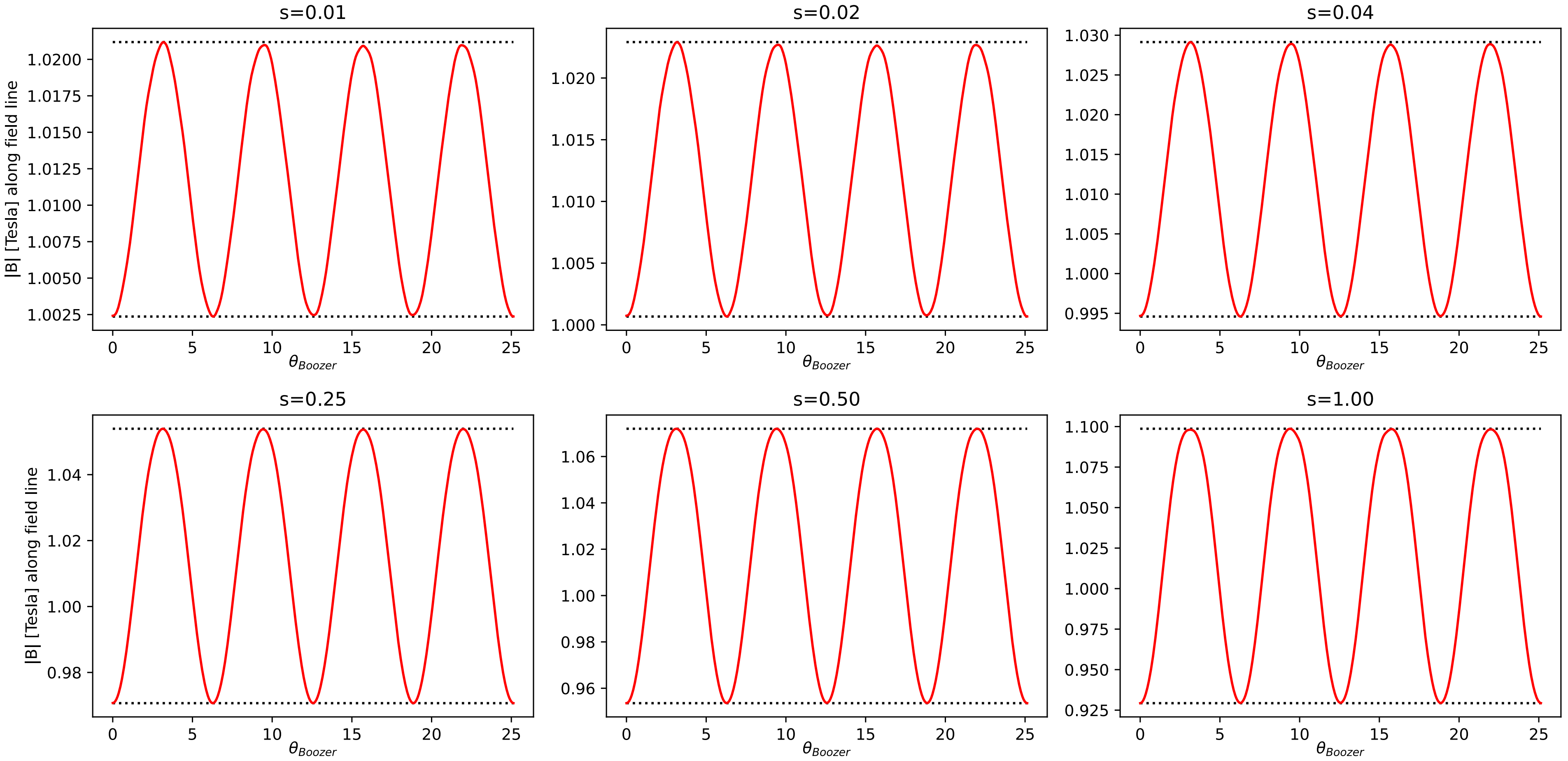
|B| along a field line for new QA



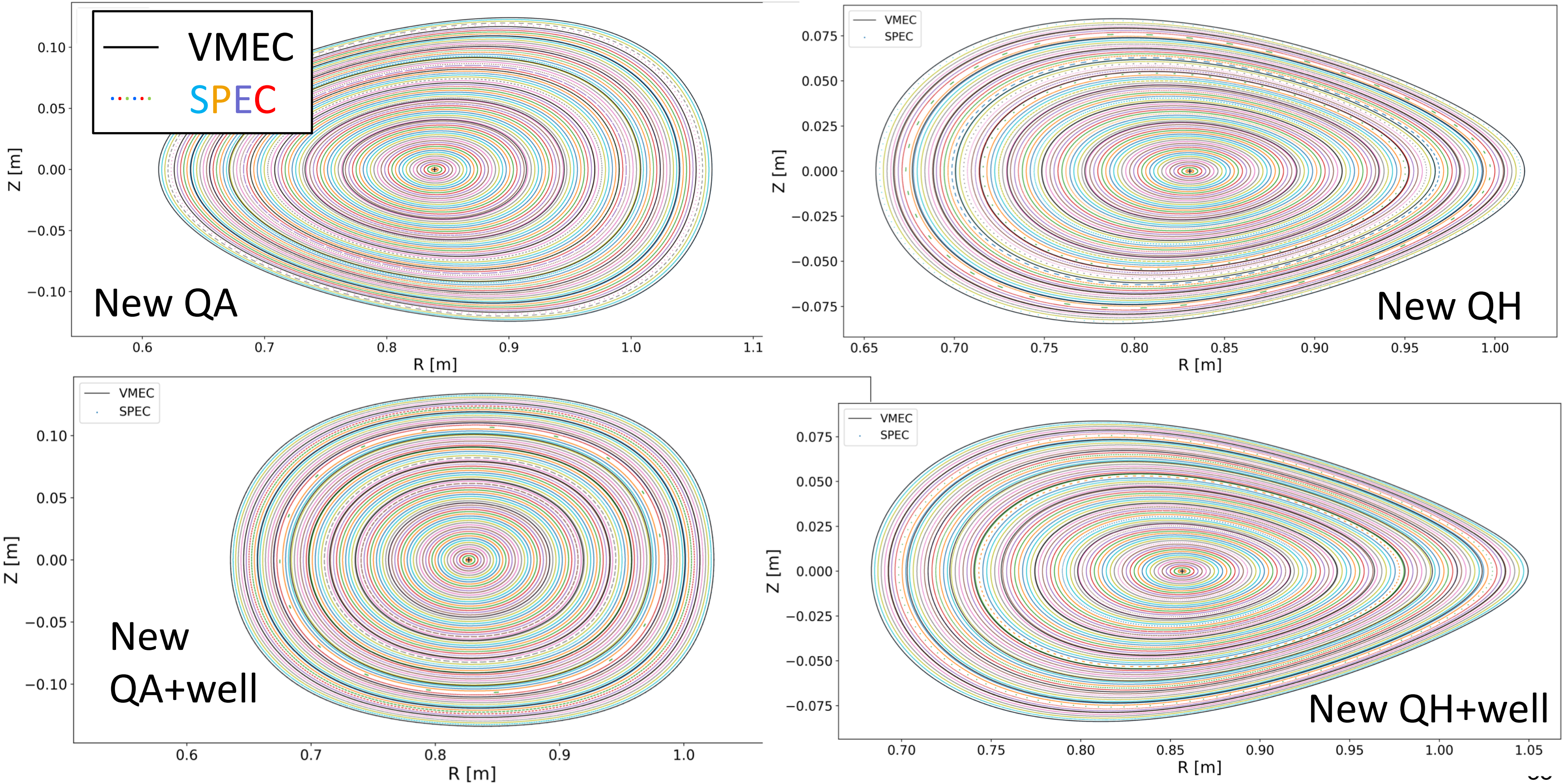
|B| along a field line for new QH



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

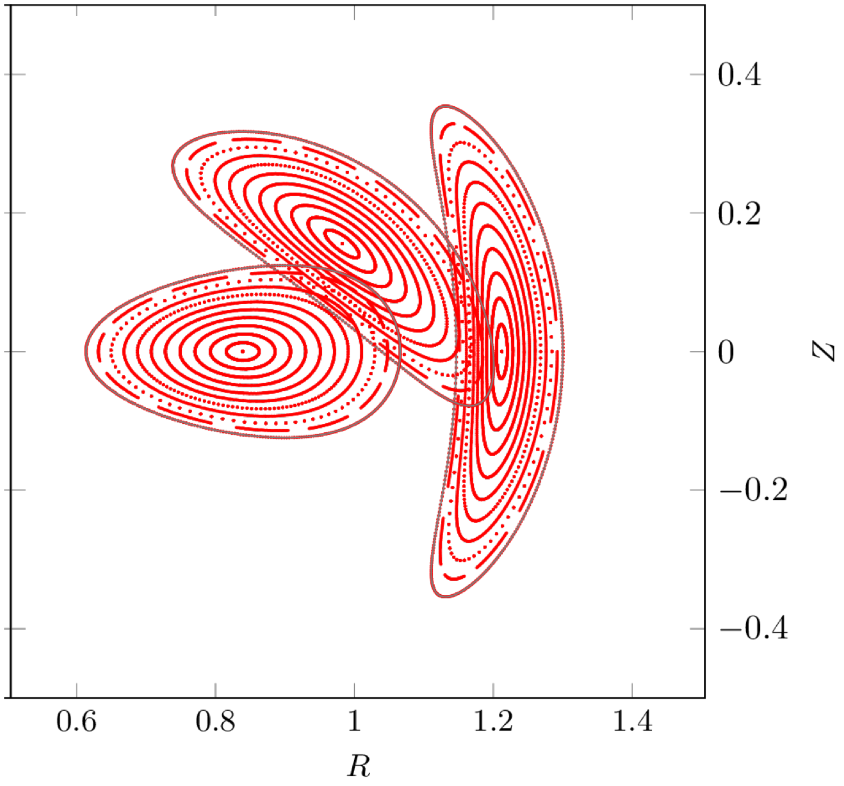


SPEC confirms the new QA/QH configurations have good surfaces



Good flux surface exist with coils

New QA



New QA+well