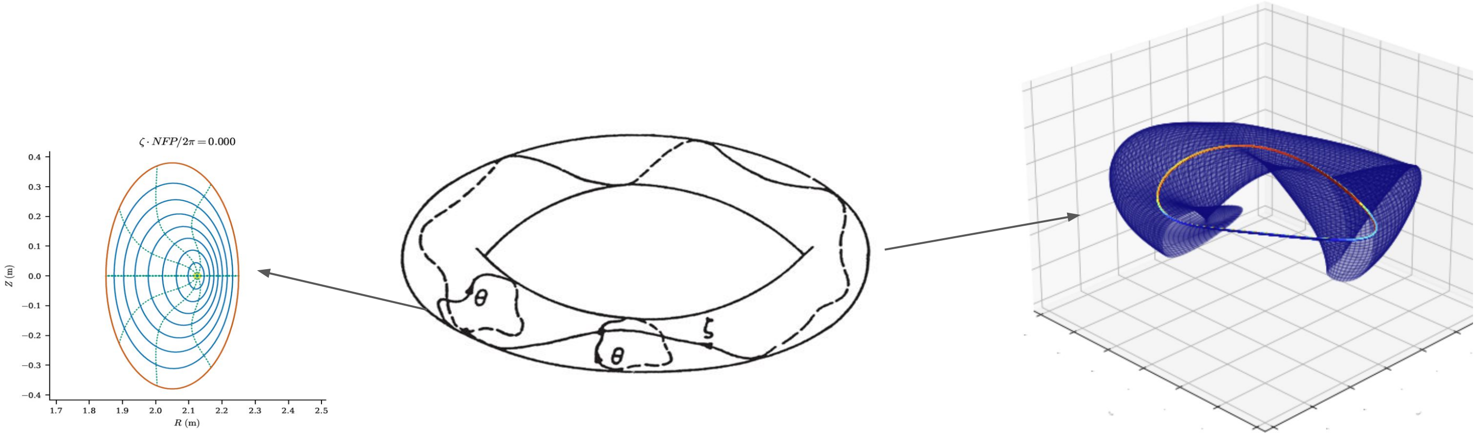
Flexible Equilibrium Constraints with the DESC code

Dario Panici, Daniel Dudt, Rory Conlin, Eduardo Rodriguez,
Egemen Kolemen



Ideal MHD Equilibrium

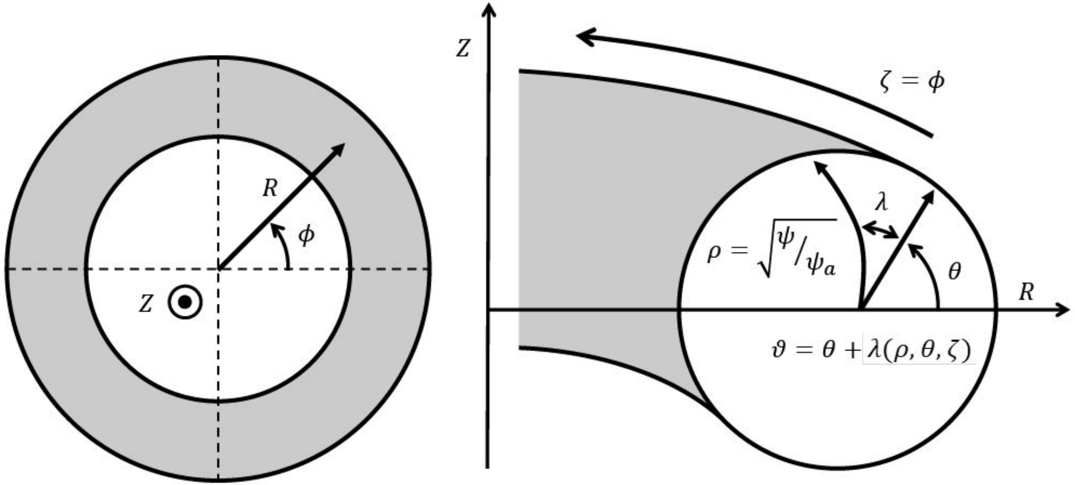
$$\boldsymbol{J} \times \boldsymbol{B} = \nabla p$$

$$\nabla \times \boldsymbol{B} = \mu_0 \boldsymbol{J}$$

$$\nabla \cdot \boldsymbol{B} = 0$$

Inverse Equilibrium Problem

- Cast problem as solving for the locations of the flux surfaces
- Yields 2nd order PDE in (ϱ,θ,$\boldsymbol{\zeta}$)



Multiple Ways to Solve Ideal MHD Equilibrium PDE

BVP in Radial Variable ϱ

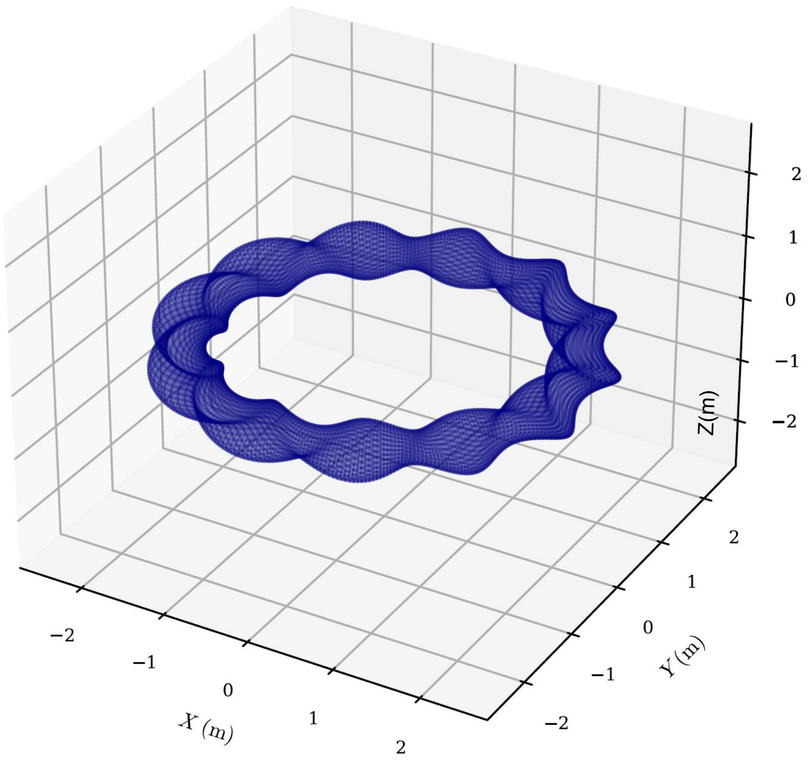
Boundary Input is the ϱ=1 flux surface

BVP in Toroidal Angle ζ

Boundary Input is the ζ=0 XS of the equilibrium

BVP in Poloidal Angle θ

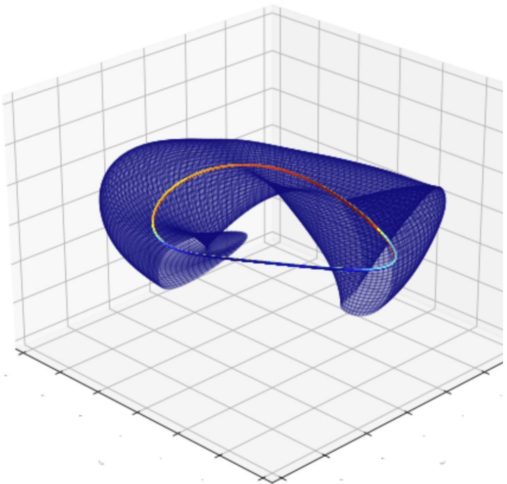
Boundary Input is the θ=0 ribbon of the equilibrium



Physical Insights Yield Constraints on XS or near Axis

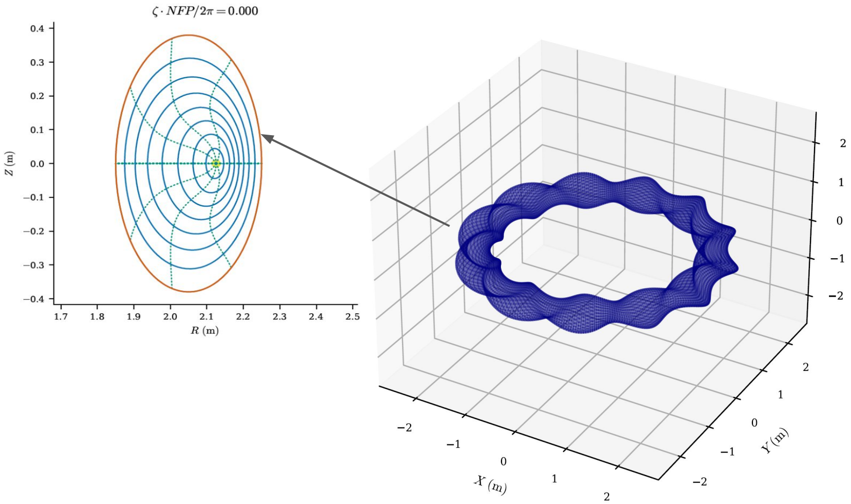
Axis + Near-Axis Behavior

Near-Axis Expansion (NAE) yields what **asymptotic behavior** of equilibrium should be near the axis, and what the **axis shape** should be

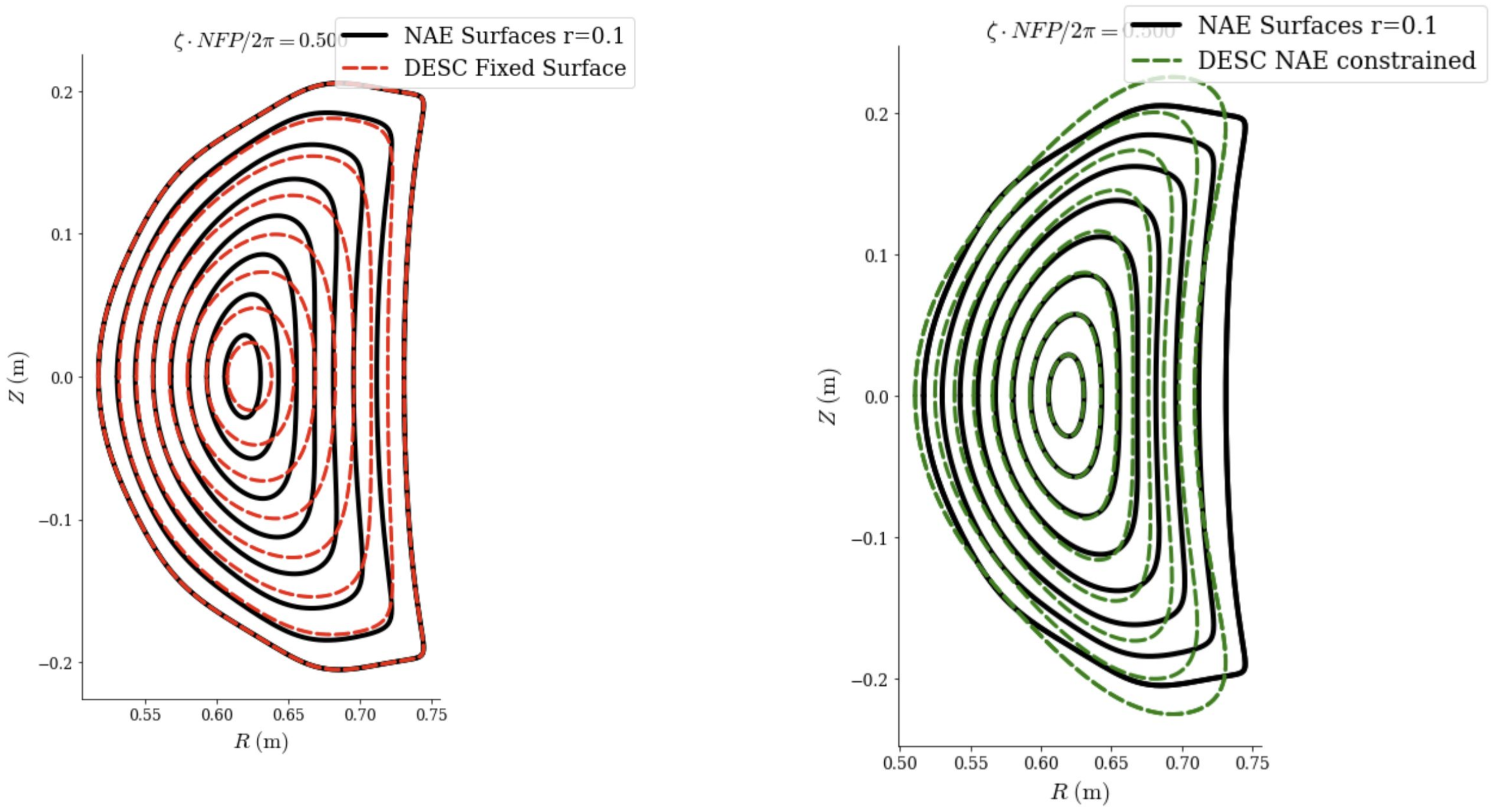


Poincare Section

Desire to avoid magnetic islands, and decoupling poloidal and toroidal resolution

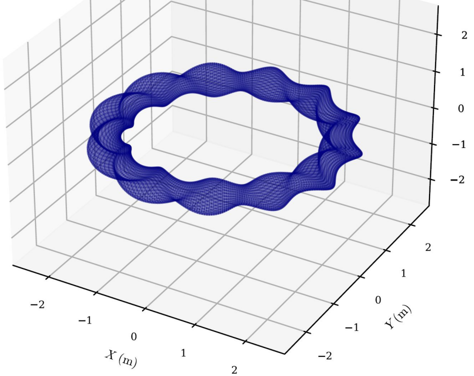


O(ϱ^1) NAE Constraint in DESC

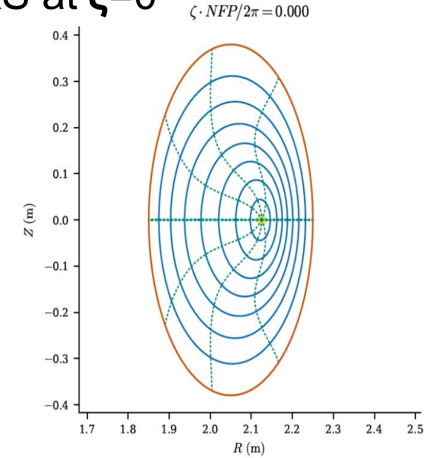


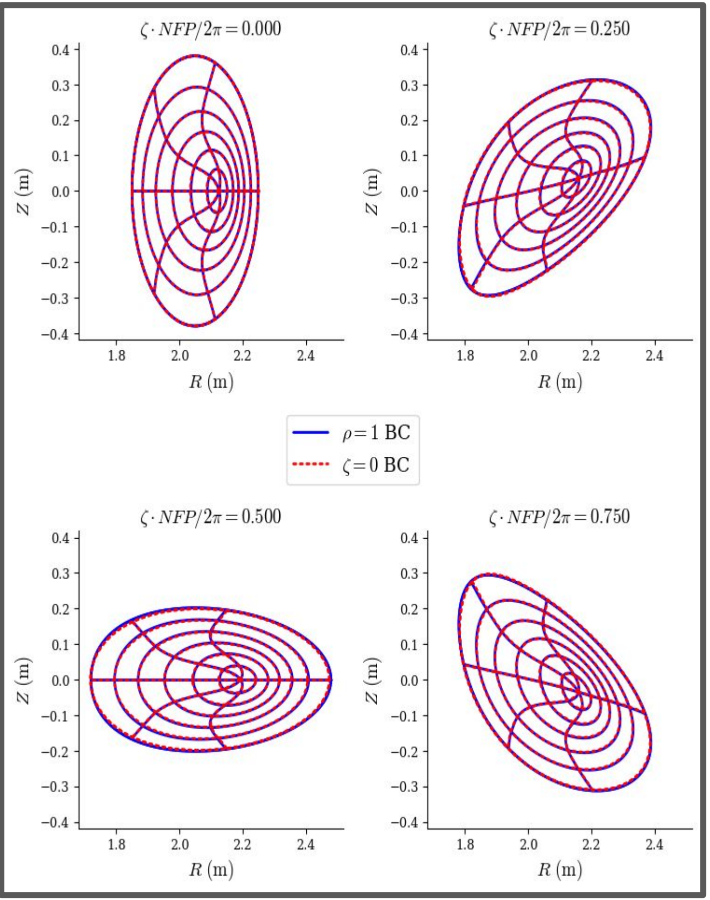
Poincare BC can Recover Stellarator LCFS Solution

INPUT: R,Z of LCFS at ρ=1



INPUT: R,Z, λ of Poincare XS at ζ=0





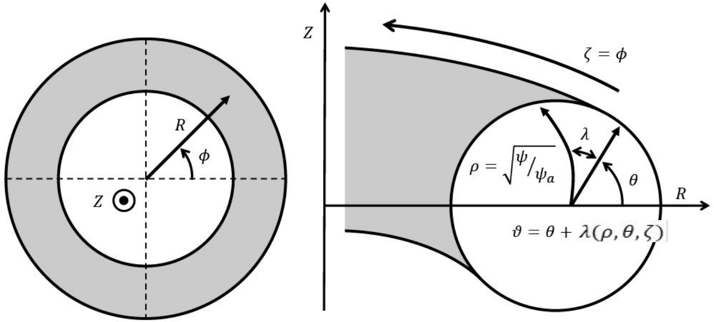
Stellarator Equilibrium - DESC

$$\boldsymbol{F} = \boldsymbol{J} \times \boldsymbol{B} - \nabla p = 0$$

(Dudt and Kolemen 2020)

- 3D Ideal MHD Equilibrium Code
- Assumes Nested Flux Surfaces
- Inverse Equilibrium Problem
- **Minimizes Force Error Directly**
- **Pseudospectral Code**

3D Spectral Representation of $\mathbf{x} = (R, \lambda, Z)$ using Fourier-Zernike Basis

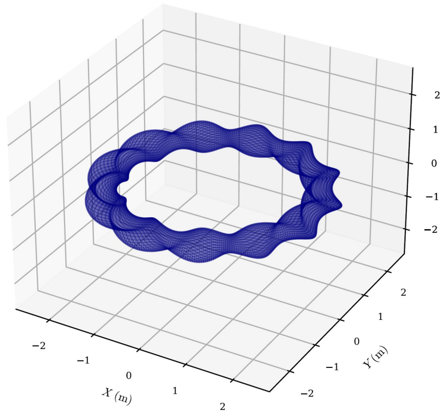


DESC Allows Flexible Constraints when Defining Equilibrium Problem

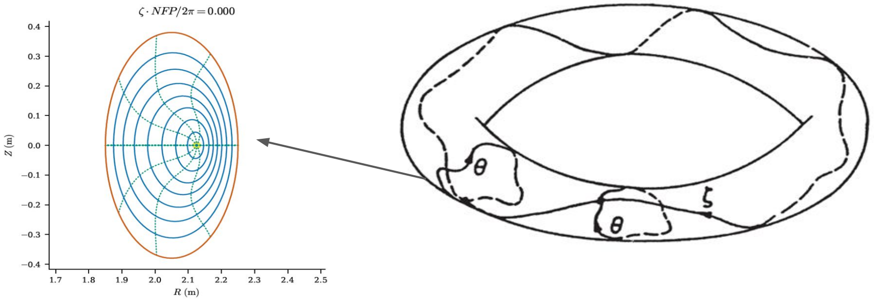
Fixed Toroidal Surface

$$R^b(\theta,\zeta) = \sum_{m=0}^{M} \sum_{n=-N}^{N} R^b_{m,n} cos(m\theta - n\zeta)$$

$$Z^b(\theta,\zeta) = \sum_{m=0}^{M} \sum_{n=-N}^{N} Z^b_{m,n} sin(m\theta - n\zeta)$$



Fixed Poincare Section



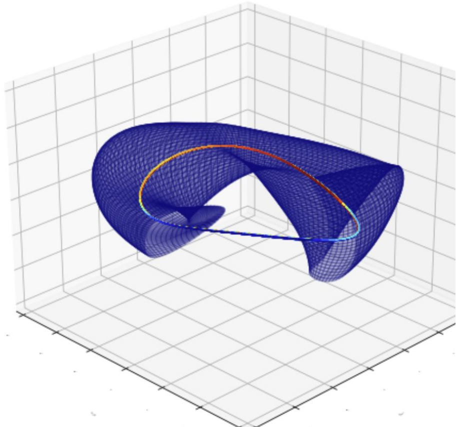
$$R^b(\rho,\theta) = \sum_{m=-M,l=0}^{M,L} R^b_{lm} \mathcal{Z}_l^m(\rho,\theta)$$

$$Z^b(\rho,\theta) = \sum_{m=-M,l=0}^{M,L} Z^b_{lm} \mathcal{Z}_l^m(\rho,\theta)$$

Fixed Axis + Near-Axis Behavior

$$R_n^{C/S} = \sum_{k=0}^{\infty} (-1)^k R_{2k,0,\pm|n|}$$

$$Z_n^{C/S} = \sum_{k=0}^{\infty} (-1)^k Z_{2k,0,\pm|n|}$$



DESC Allows Flexible Constraints when Defining Equilibrium Problem - Fixed ρ=1 Boundary

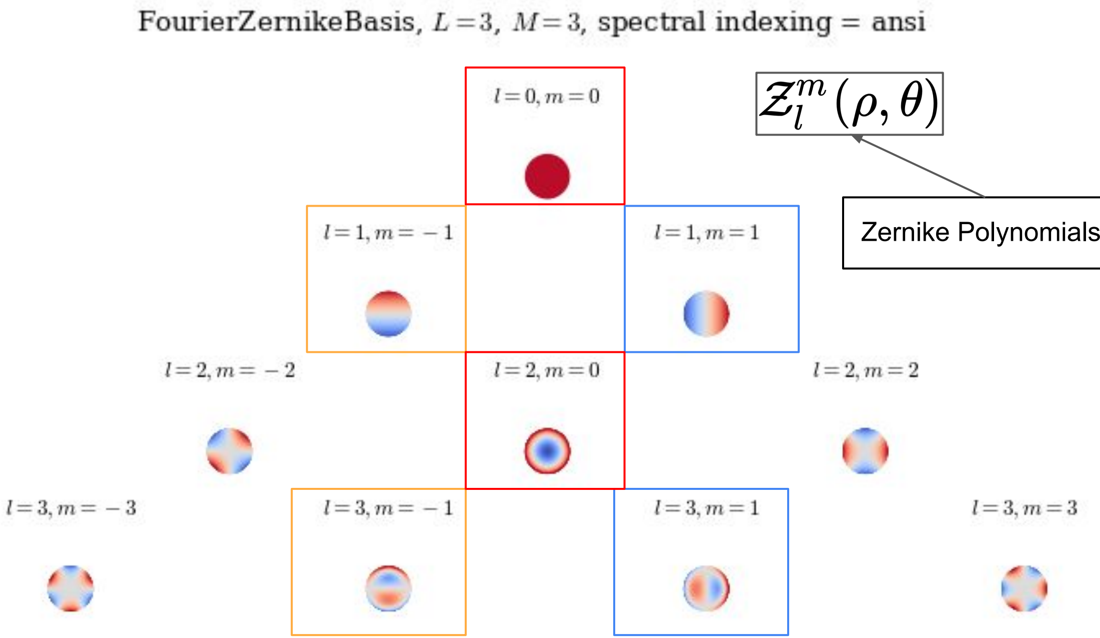
$$R^b(\theta,\zeta) = \sum_{m=0}^{M} \sum_{n=-N}^{N} R^b_{m,n} cos(m\theta - n\zeta)$$

$$Z^b(\theta,\zeta) = \sum_{m=0}^{M} \sum_{n=-N}^{N} Z^b_{m,n} sin(m\theta - n\zeta)$$

<u>Fixed-Boundary ρ=1 Constraint</u>

$$\sum_{l=0}^{L} R_{lmn} \mathcal{Z}_l^m(\rho=1,\theta) = R^b_{mn} \quad \forall m,n$$

$$\sum_{l=0}^{L} Z_{lmn} \mathcal{Z}_l^m(\rho=1,\theta) = Z^b_{mn} \quad \forall m,n$$



DESC Allows Flexible Constraints when Defining Equilibrium Problem - Fixed-Poincare ζ=0

$$R^b(\rho,\theta) = \sum_{m=-M,l=0}^{M,L} R^b_{lm} \mathcal{Z}^m_l(\rho,\theta)$$

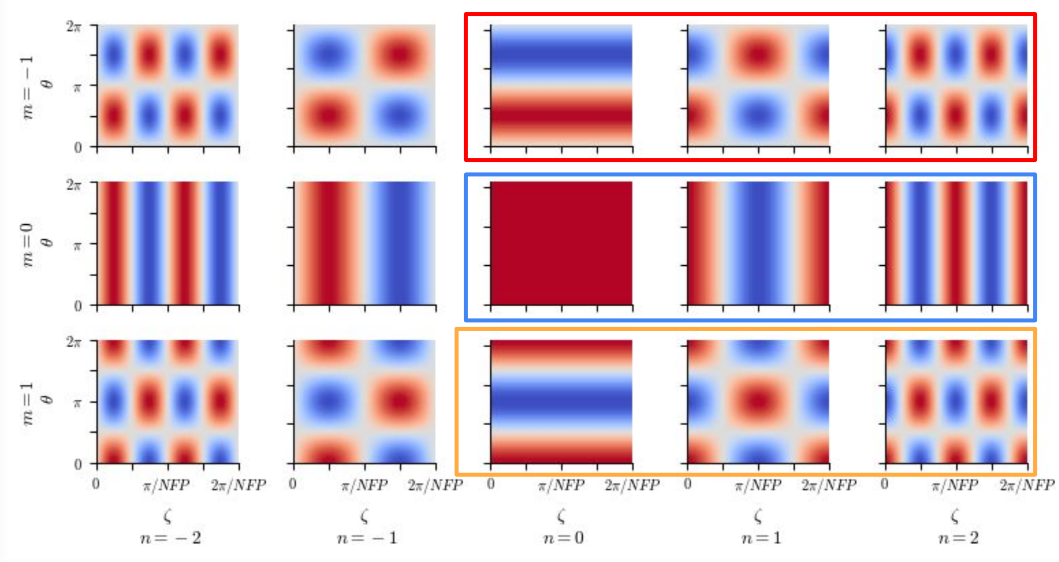
$$Z^b(\rho,\theta) = \sum_{m=-M,l=0}^{M,L} Z^b_{lm} \mathcal{Z}^m_l(\rho,\theta)$$

Zernike Polynomials

<u>Fixed-Poincare ζ=0 Constraint</u>

$$\sum_{n=0}^{N} R_{lmn} = R^b_{lm} \quad \forall l,m$$

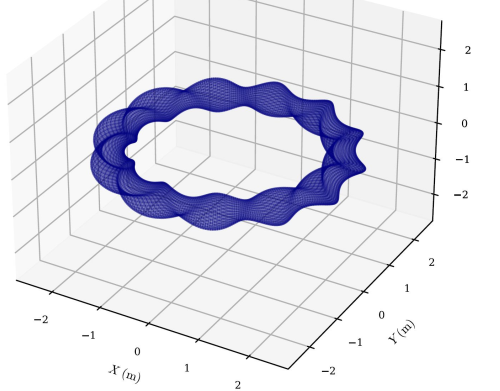
$$\sum_{n=0}^{N} Z_{lmn} = Z^b_{lm} \quad \forall l,m$$



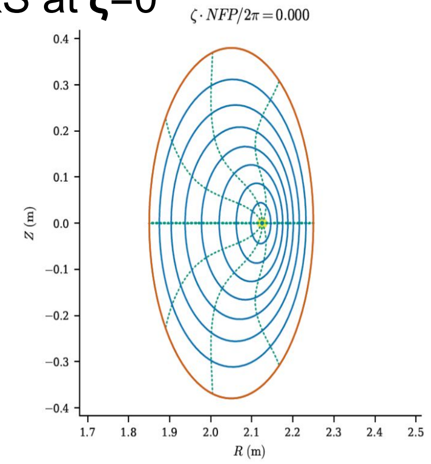
$$\mathcal{F}^n(\zeta) = \begin{cases} cos(|n|N_{FP}\zeta) \text{for } n \geq 0 \\ sin(|n|N_{FP}\zeta) \text{for } n < 0 \end{cases}$$

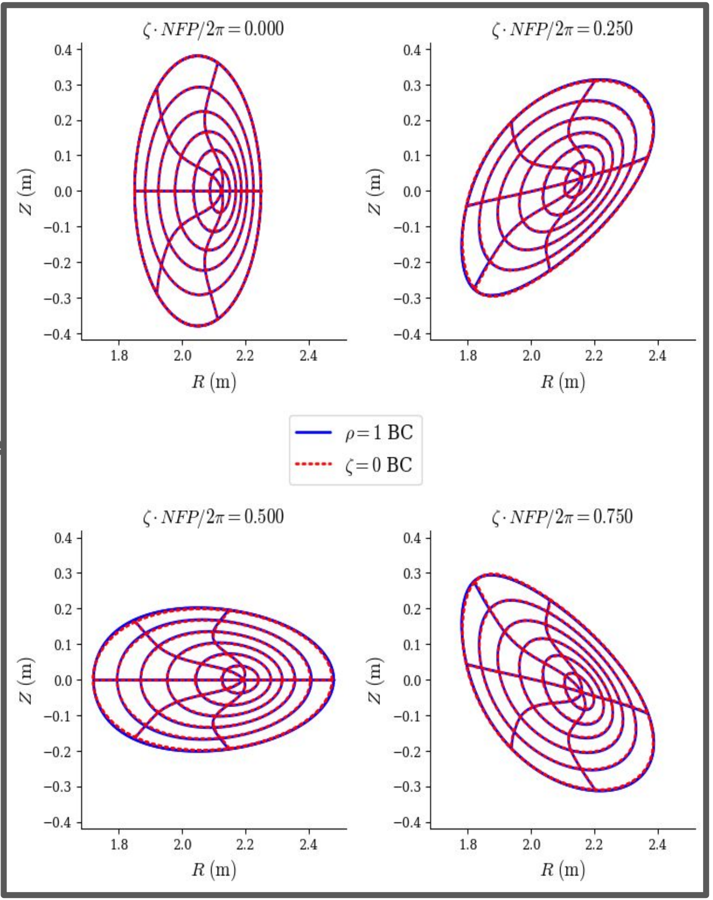
Poincare BC can Recover Stellarator LCFS Solution

INPUT: R,Z of LCFS at ϱ=1



INPUT: R,Z, λ of Poincare XS at ζ=0





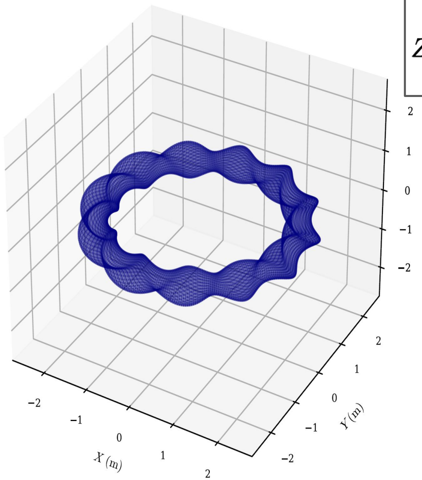
Poincare BC Requires a Fraction of Variables To Represent as compared to LCFS

Method	Number of Coefficients To Represent Boundary
R,Z, λ **Poincare** XS Zernike Polynomial Series	~ $\frac{5}{4}M^2$
R,Z **LCFS** Fourier Series	~ $4M^2$

Assuming Boundary Resolutions L=M=N and Stellarator Symmetry

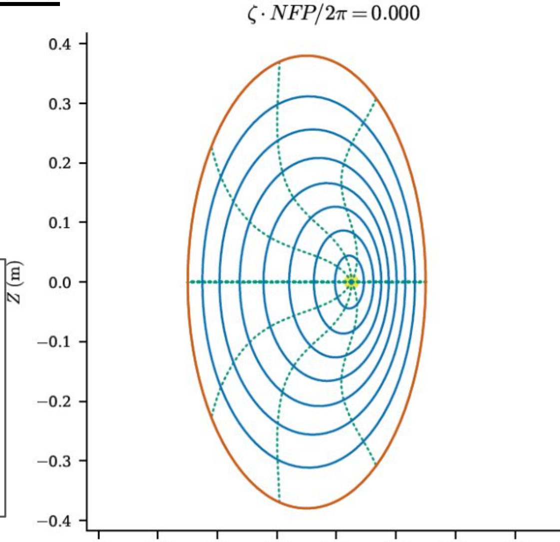
<u>LCFS</u>

$$R^b(\theta,\zeta) = \sum_{m=0}^{M}\sum_{n=-N}^{N} R^b_{m,n} cos(m\theta - n\zeta)$$
$$Z^b(\theta,\zeta) = \sum_{m=0}^{M}\sum_{n=-N}^{N} Z^b_{m,n} sin(m\theta - n\zeta)$$



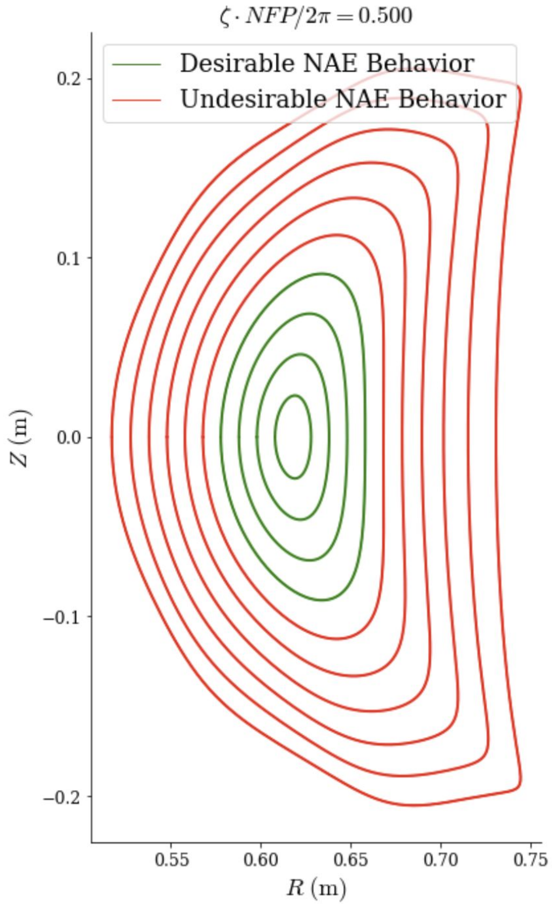
<u>Poincare</u>

$$R^b(\rho,\theta) = \sum_{m=-M,l=0}^{M,L} R^b_{lm} \mathcal{Z}_l^m(\rho,\theta)$$
$$Z^b(\rho,\theta) = \sum_{m=-M,l=0}^{M,L} Z^b_{lm} \mathcal{Z}_l^m(\rho,\theta)$$



Near-Axis Expansion (NAE) Constraints in DESC
(with E. Rodriguez)

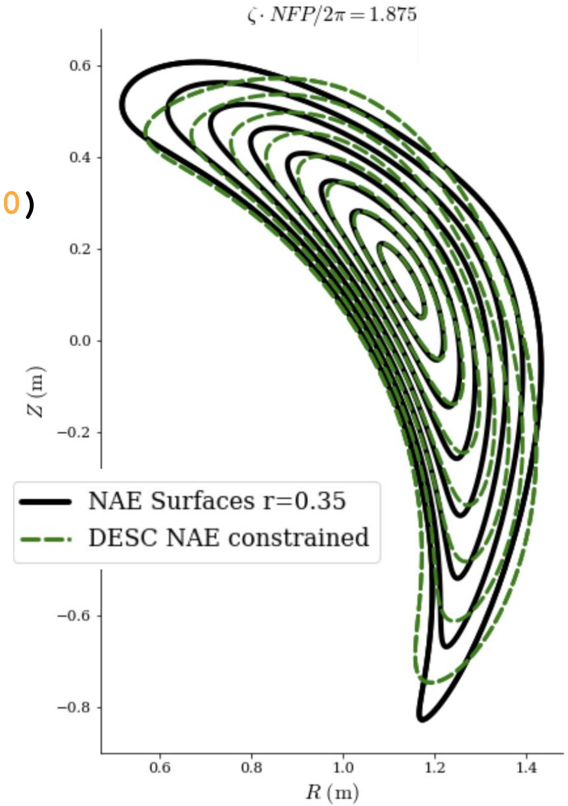
- Idea is to constrain the global equilibrium to have NAE behavior as $\varrho \rightarrow 0$
  - only use information from NAE **where it is most valid**
  - Avoid singular behavior present when evaluating at large **r**
- Map NAE coefficients to Fourier-Zernike modes of DESC to fix $O(\varrho^0)$ (axis) and $O(\varrho^1)$ behavior



pyQSC equilibrium evaluated at **r** =0.1

Aside: Interface Between DESC and pyQSC Makes Using NAE Information Easy

```
# get DESC equilibrium from qsc
qsc = Qsc.from_paper('precise QA',nphi=99)
desc_eq = Equilibrium.from_near_axis(qsc,r=0.35,L=9,M=9,N=10)

# gets constraints on axis and O(r) coefficients
to pass to eq.solve using utility function
cs = get_NAE_constraints(desc_eq,qsc,order=1)

# solve
desc_eq.solve(objective="force",constraints=cs);
```



O(ϱ^0) (axis) Constraint in DESC

NAE axis in pyQSC given as Fourier series in cylindrical toroidal angle ϕ:

$$R = R_0 + \sum_{n=1}^{N} (R_n^C \cos m\phi + R_n^S \sin m\phi) \qquad Z = \sum_{n=1}^{N} (Z_n^C \cos m\phi + Z_n^S \sin m\phi)$$

Constraint in DESC representation is simple: Evaluate DESC R(ϱ,θ,ϕ), Z(ϱ,θ, ϕ) at ϱ=0 and match terms:

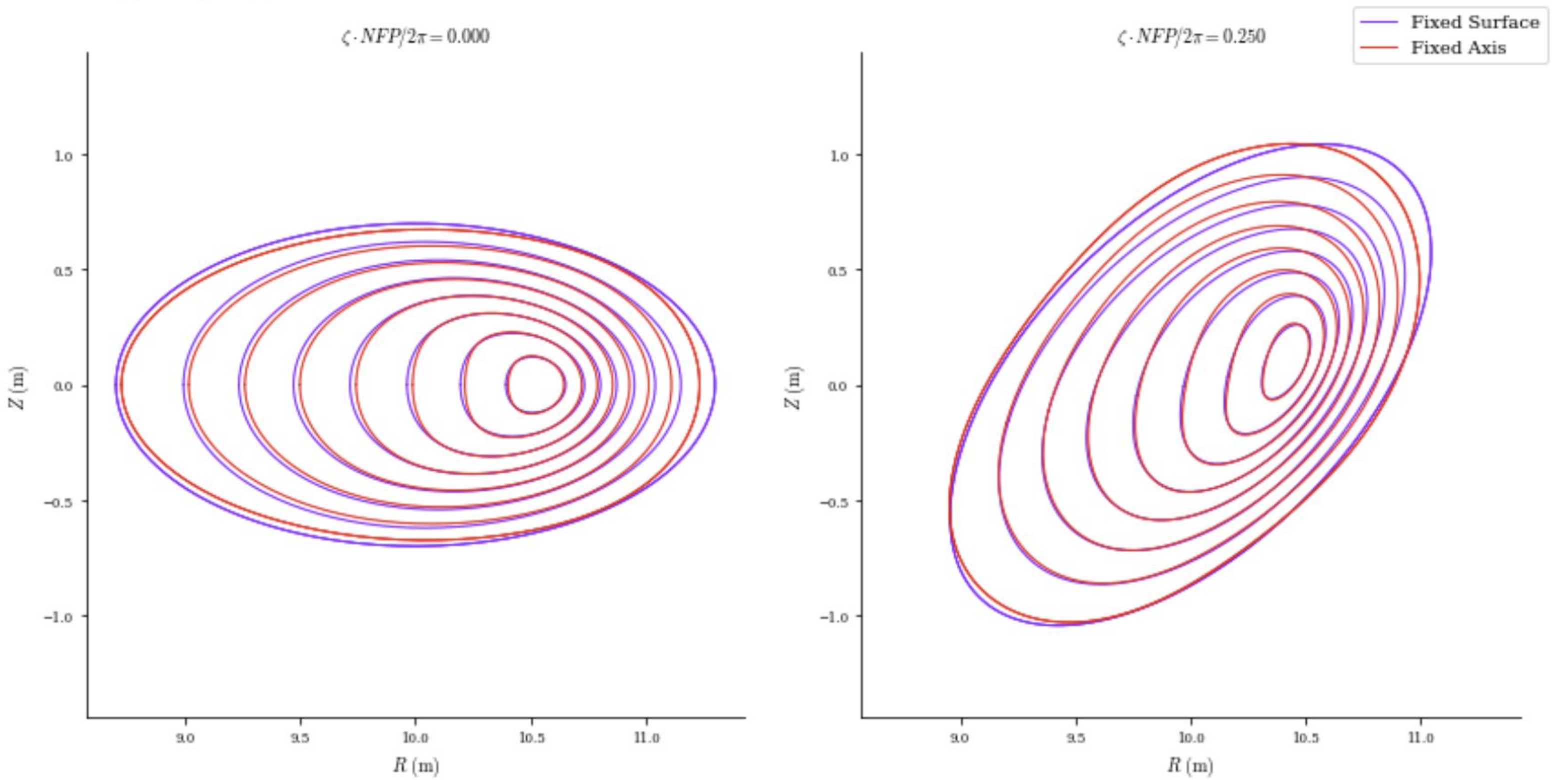
NAE Axis Coefficients

$$R_n^{C/S} = \sum_{k=0}^{\infty} (-1)^k R_{2k,0,\pm|n|}$$

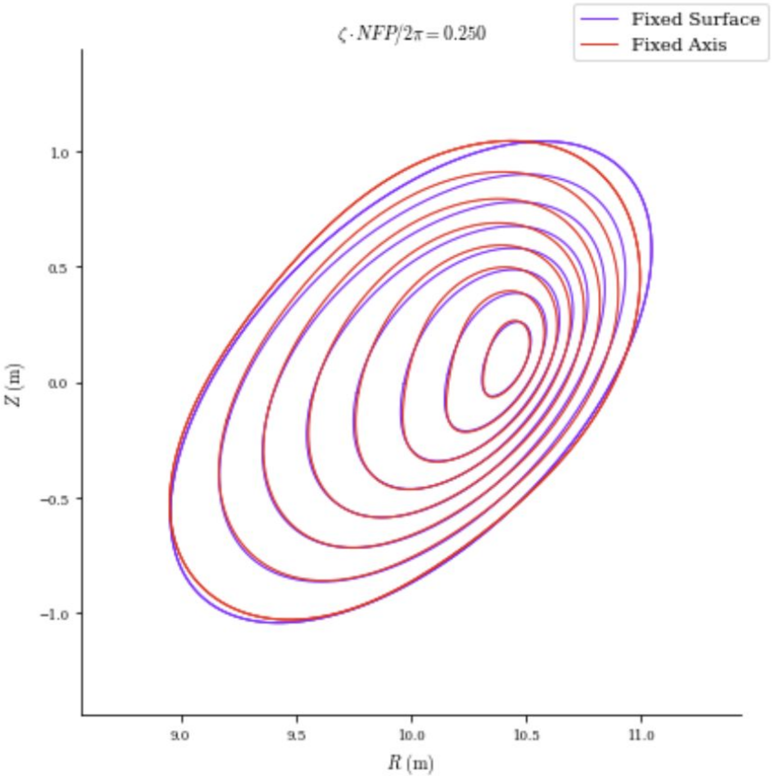
$$Z_n^{C/S} = \sum_{k=0}^{\infty} (-1)^k Z_{2k,0,\pm|n|}$$

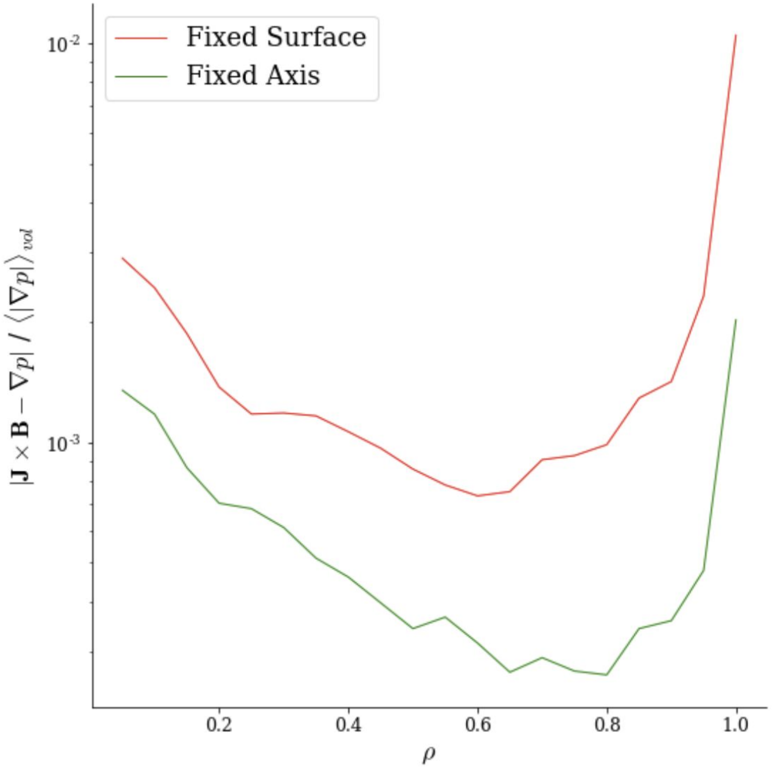
DESC Fourier-Zernike Coefficients

O(ϱ^0) (axis) Constraint in DESC - Example Solve

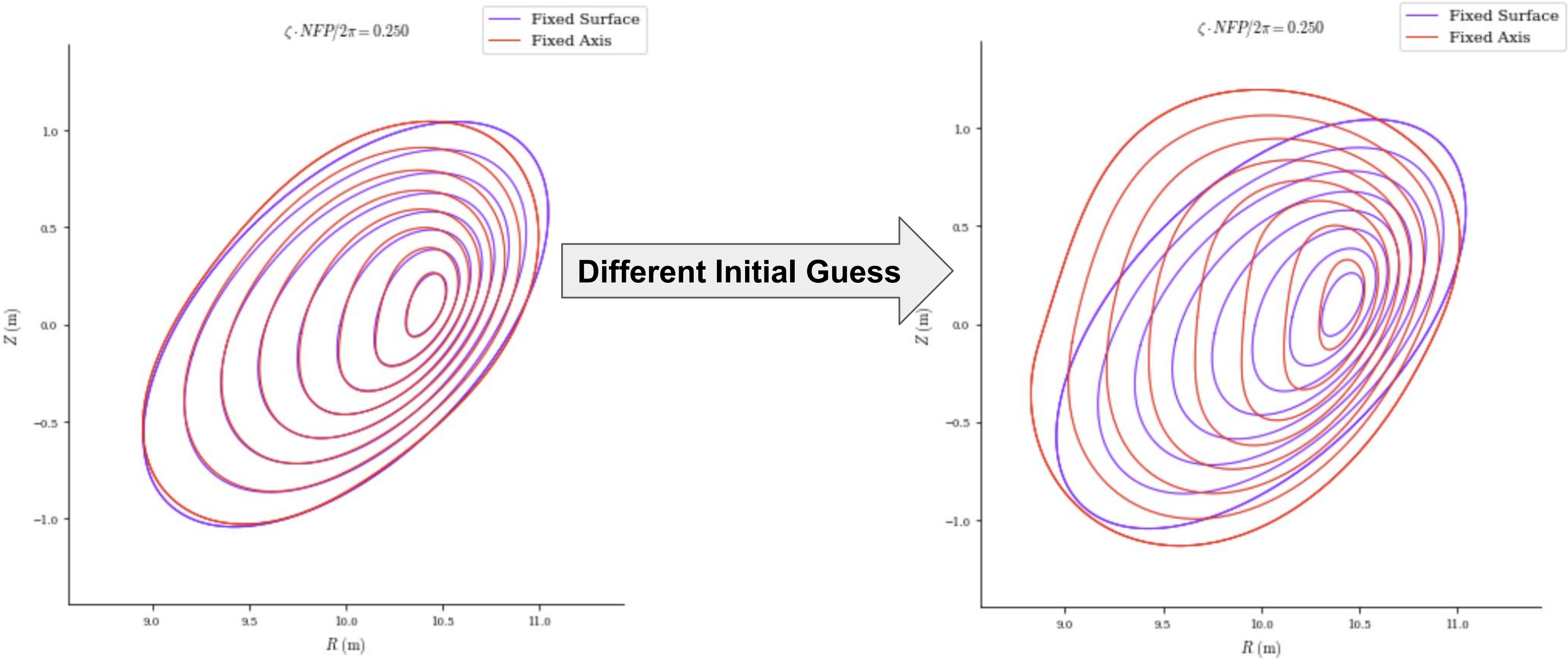


O(ϱ^0) (axis) Constraint in DESC - Finds Nearby Equilibrium with lower Force Error





O(ϱ^0) (axis) Constraint in DESC - Under-constrained Problem, Finds Closest Equilibrium



O(ϱ^1) NAE Constraint in DESC

After a short geometric derivation, one can derive (up to O(ϱ)) the R,Z position of a point on a flux surface from the NAE in terms of the cylindrical angle

$$\mathbf{r} \approx \mathbf{r}_0(\phi) + \rho R_1 \hat{\mathbf{R}} + \rho Z_1 \hat{\mathbf{Z}}$$

where

$$R_1 = \mathcal{R}_{1,1}(\phi)\cos\theta + \mathcal{R}_{1,-1}(\phi)\sin\theta \qquad Z_1 = Z_{1,1}(\phi)\cos\theta + Z_{1,-1}(\phi)\sin\theta$$

And the coefficients are functions of the NAE **X,Y** coefficients and the Frenet-Serret basis vectors
Then equating the O(ϱ) coefficients in the DESC Fourier-Zernike basis with the above expressions yields:

(Identical expressions for Z as well)

NAE Axis Coefficients

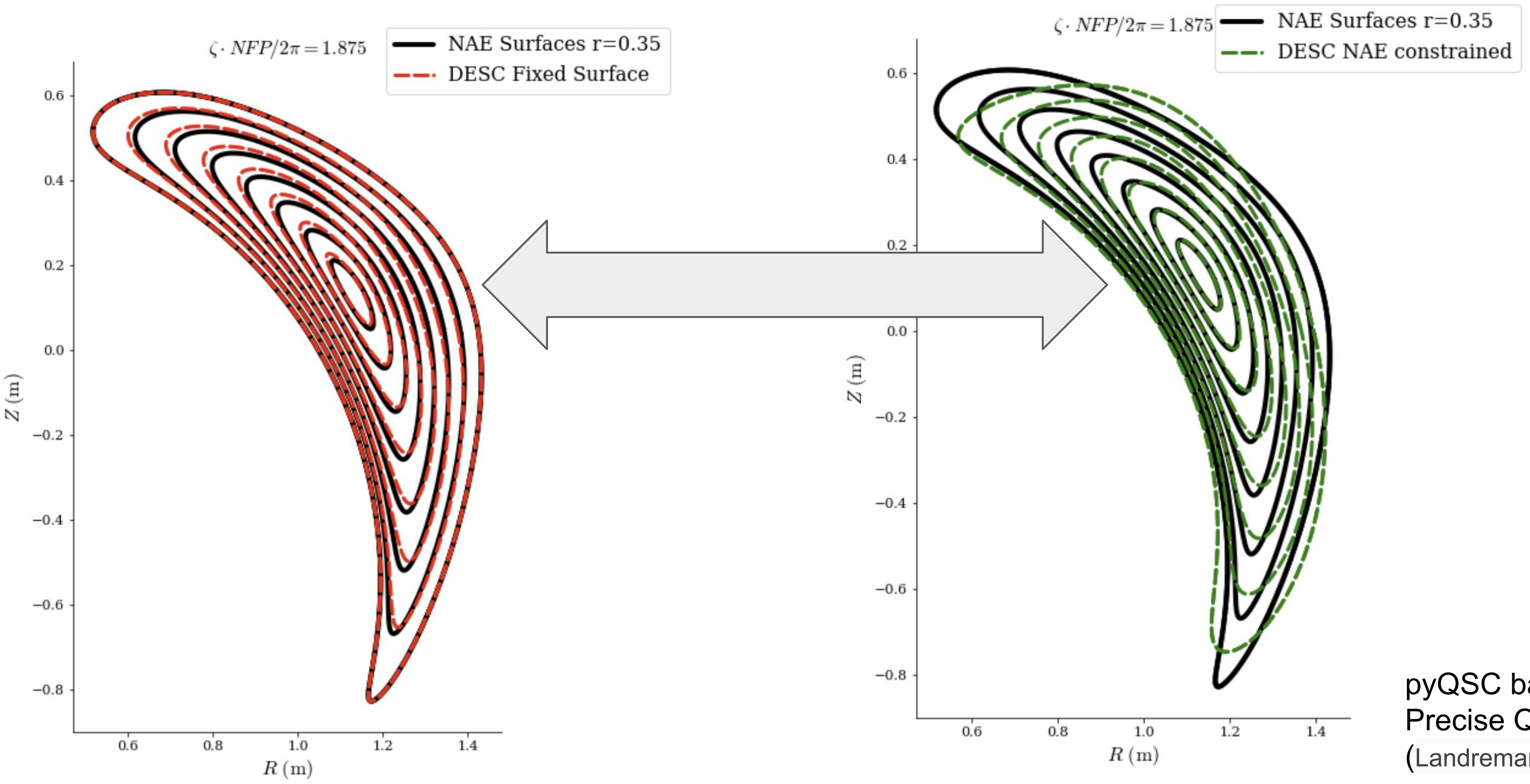
$$\mathcal{R}_{1,1,n} = -\sum_{k=1}^{M}(-1)^k k R_{2k-1,1,n},$$

$$\mathcal{R}_{1,-1,n} = -\sum_{k=1}^{M}(-1)^k k R_{2k-1,-1,n},$$

DESC Fourier-Zernike Coefficients

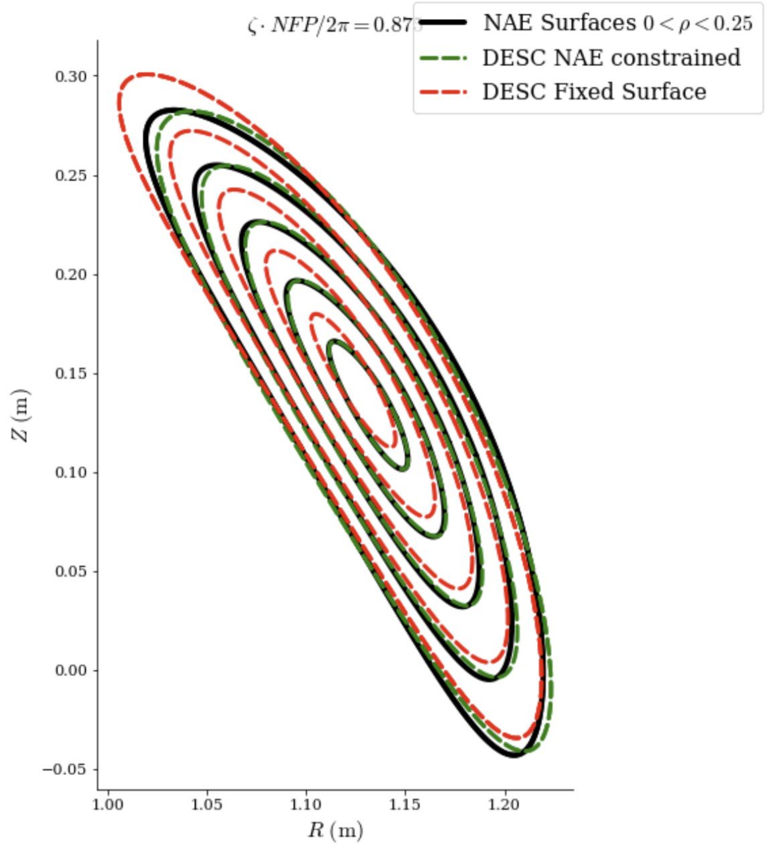
Assumes Boozer poloidal angle

$O(\varrho^1)$ Constraint in DESC - Solved Equilibrium Agrees with NAE surfaces NEAR-AXIS, unlike Surface Solve



pyQSC based on Precise QA from (Landreman and Paul 2022)

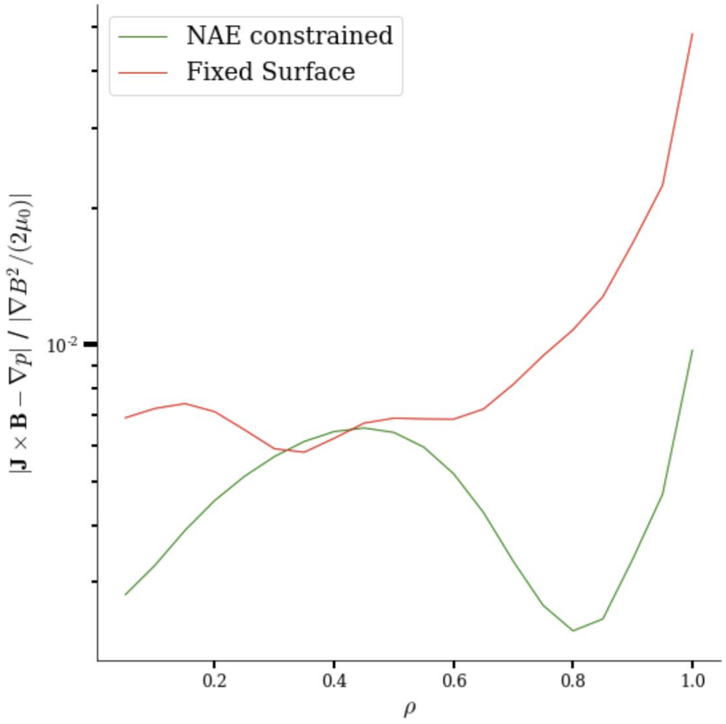
O(ϱ^1) Constraint in DESC - Solved Equilibrium Agrees with NAE surfaces NEAR-AXIS, unlike Surface Solve



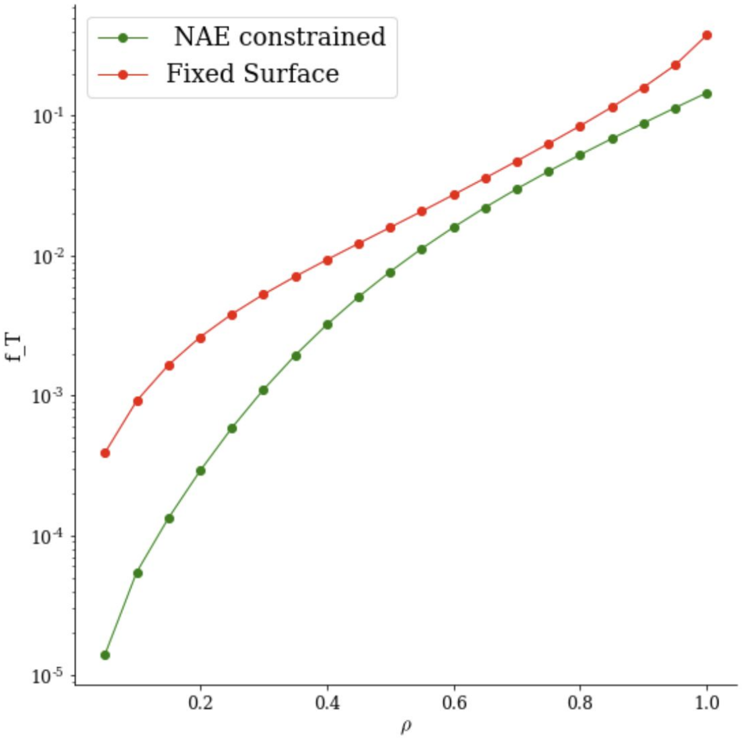
pyQSC based on Precise QA from (Landreman and Paul 2022)

$O(\varrho^1)$ Constraint in DESC - Lower Error Near-Axis and Better QS

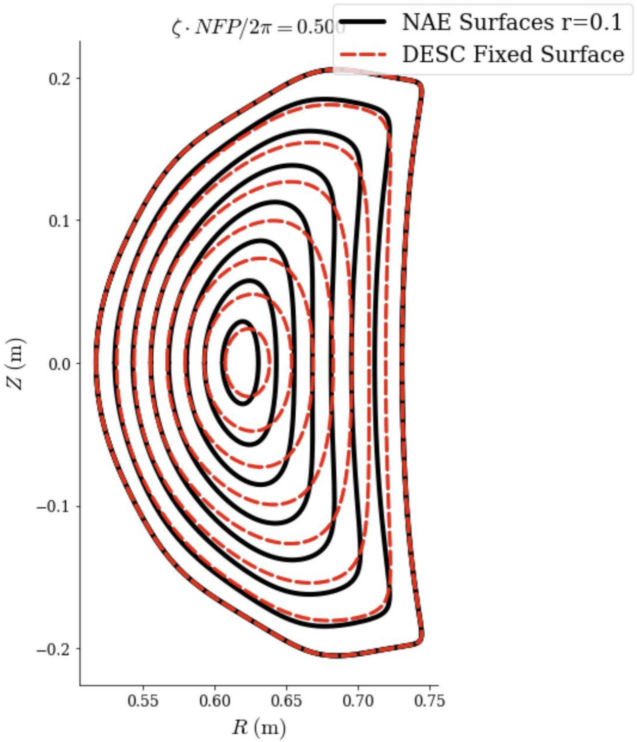
Force Error

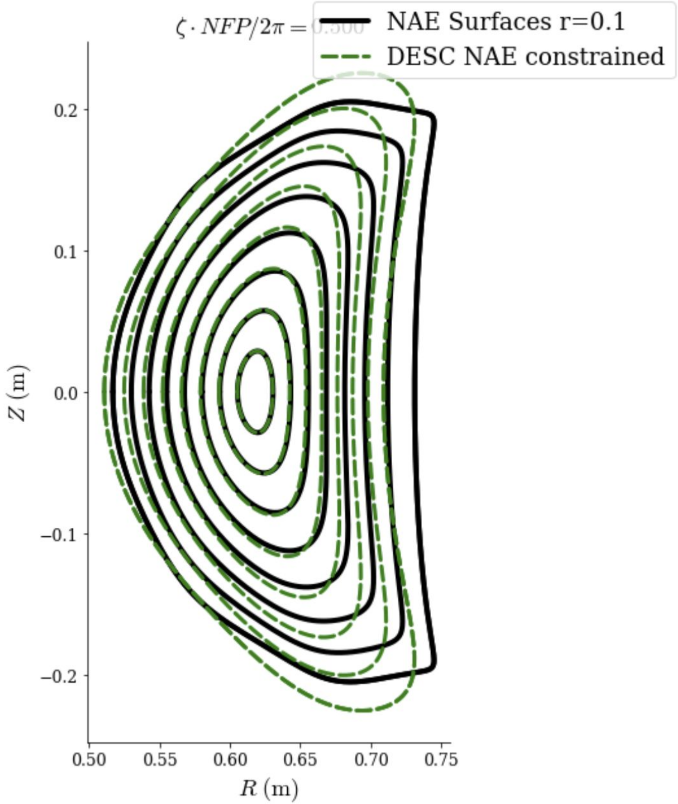


QS Triple Product

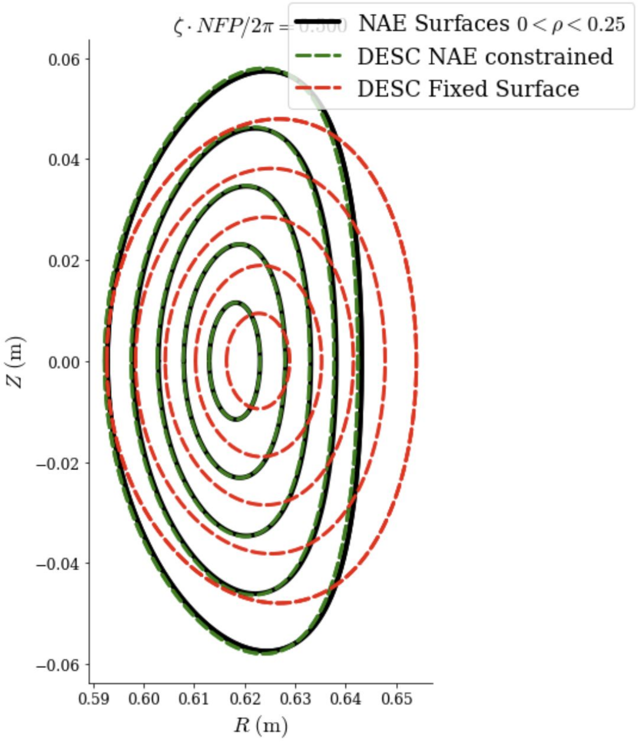


O(ϱ^1) Constraint in DESC - Example Solve where Fixed Surface Struggles (Example From E. Rodriguez)



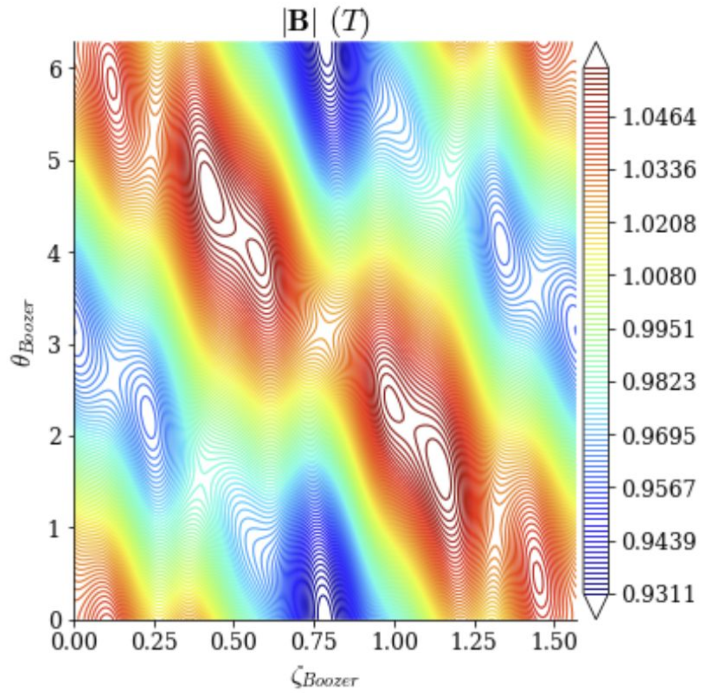


O(ϱ^1) Constraint in DESC - Example Solve where Fixed Surface Struggles (Example From E. Rodriguez)

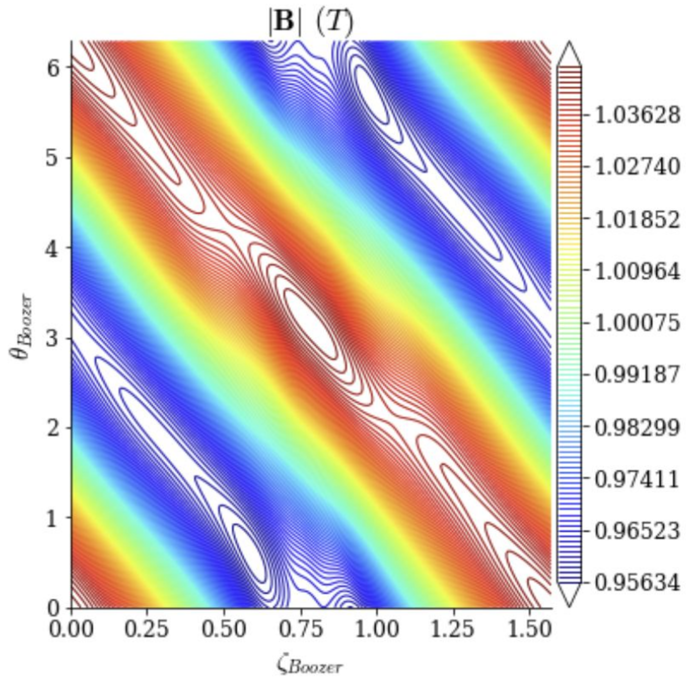


O(ϱ^1) Constraint in DESC - Example Solve where Fixed Surface Struggles - QS at ϱ=0.25

Fixed Surface

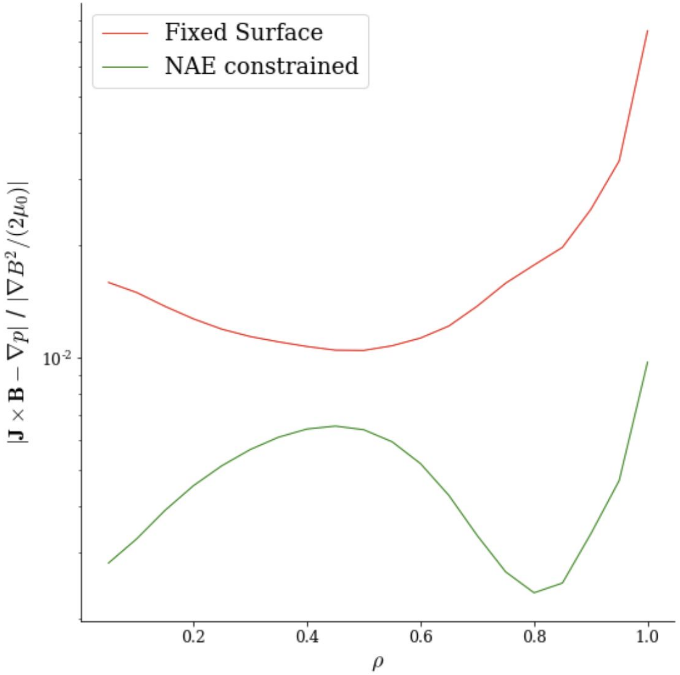


NAE Constrained

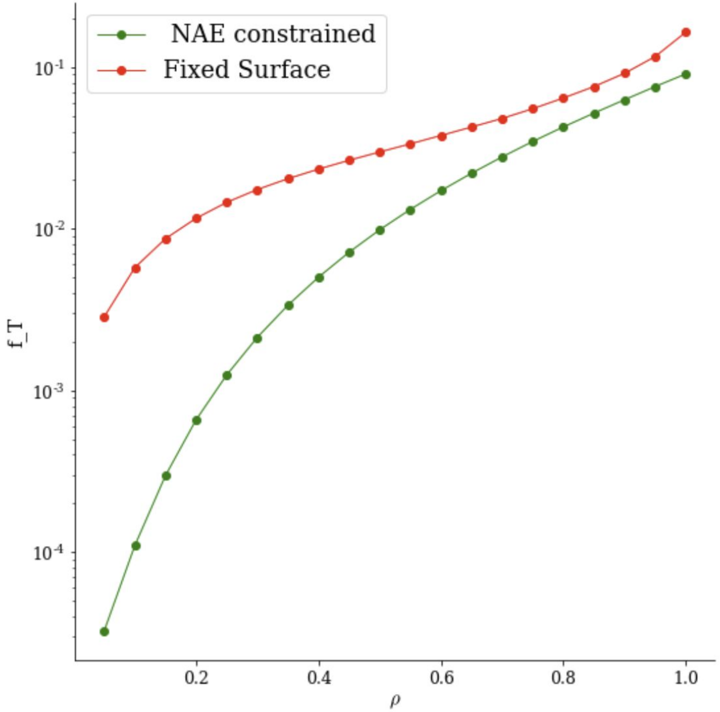


O(ϱ^1) Constraint in DESC - Example Solve where Fixed Surface Struggles - Force Error and QS

Force Error



QS Triple Product



Further Verification of NAE constraint Ongoing

Expected NAE Behavior of DESC solution to use for verification:

- Force error decreases towards axis ✅
- Surfaces near-axis match NAE ✅
- iota near-axis matches NAE
- |B| on axis matches NAE

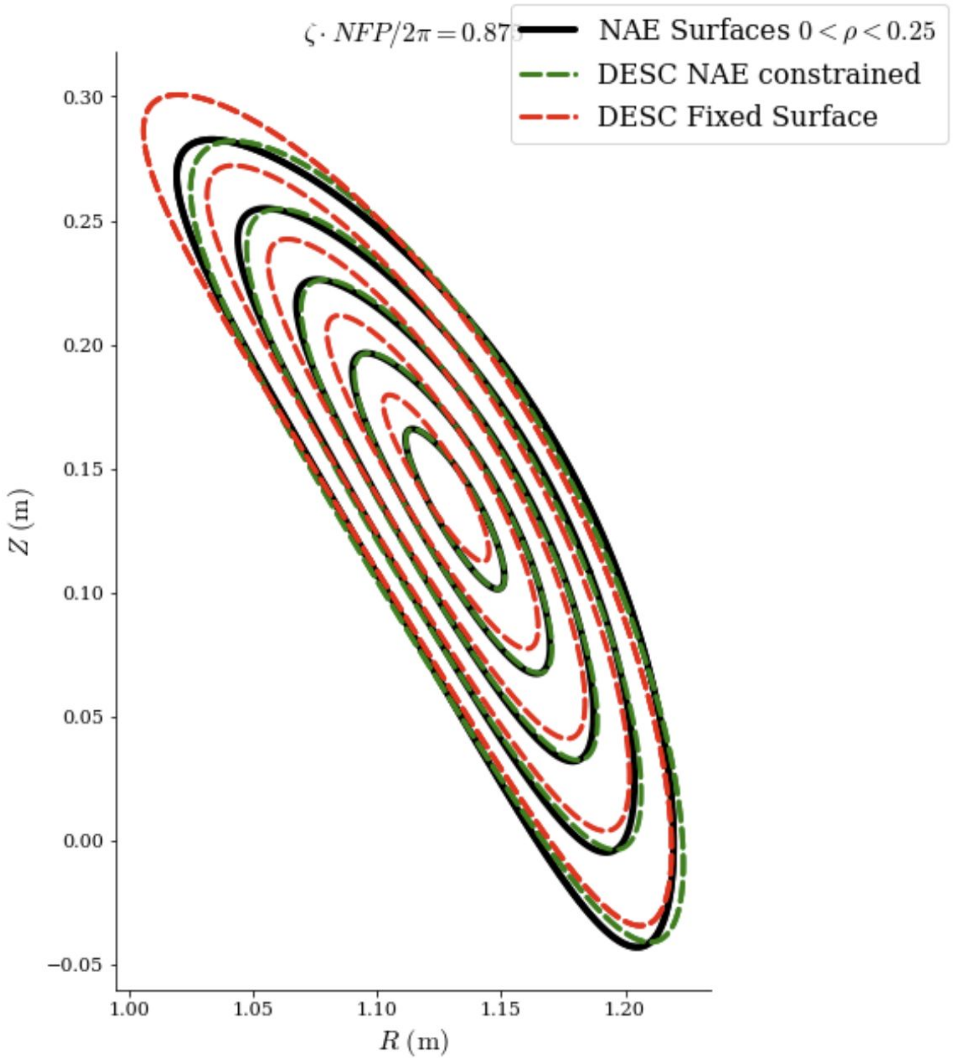
Others?

DESC Offers Unique Flexibility in Constraints that Open New Possibilities

- Poincare
  - requires much fewer number of the input coefficients as compared to the conventional last-closed-flux surface (ϱ=1) boundary condition
  - Could be utilized to optimize in a lower-dimensional subspace
  - Potentially restricts to only solutions with nested surfaces
- NAE Constraints
  - Can offer connection between rich NAE+QS theory and global solutions
  - Allow global solutions to be found matching NAE axes that otherwise could not be found traditionally
  - verification ongoing
  - future work to use with inequality constraint in DESC (O(r) constraint only enforced up to O(r))

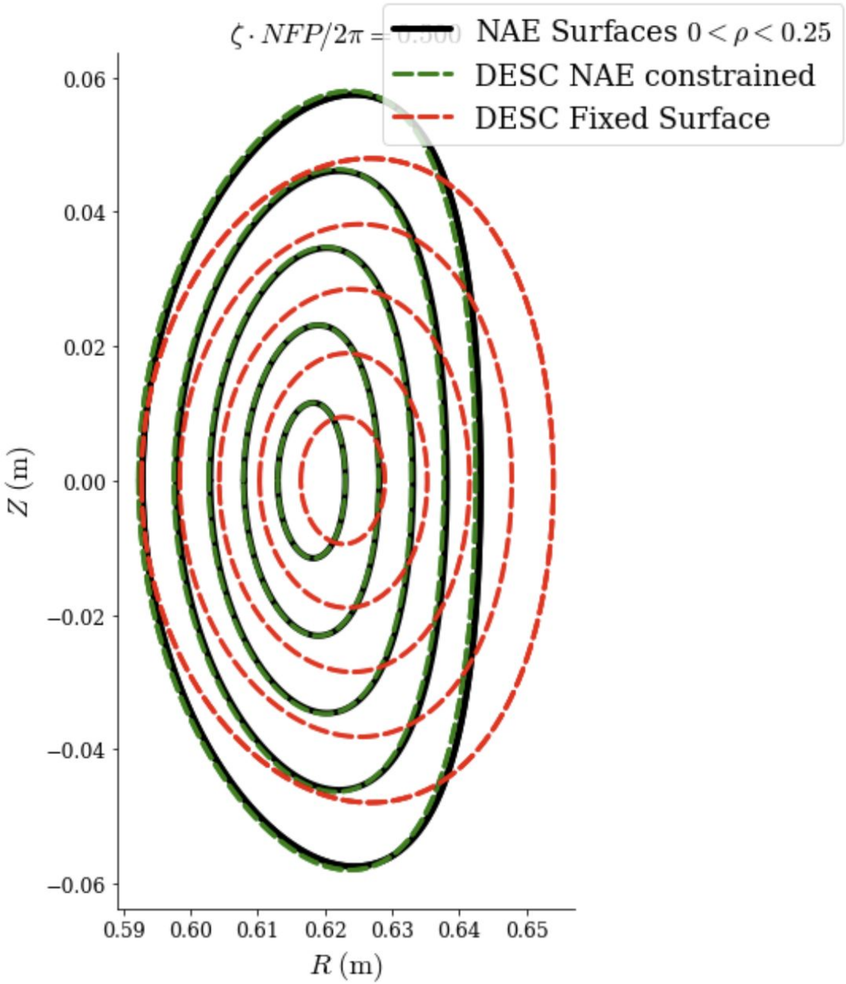
Backup

Closer look at flux surfaces near axis for Precise QA



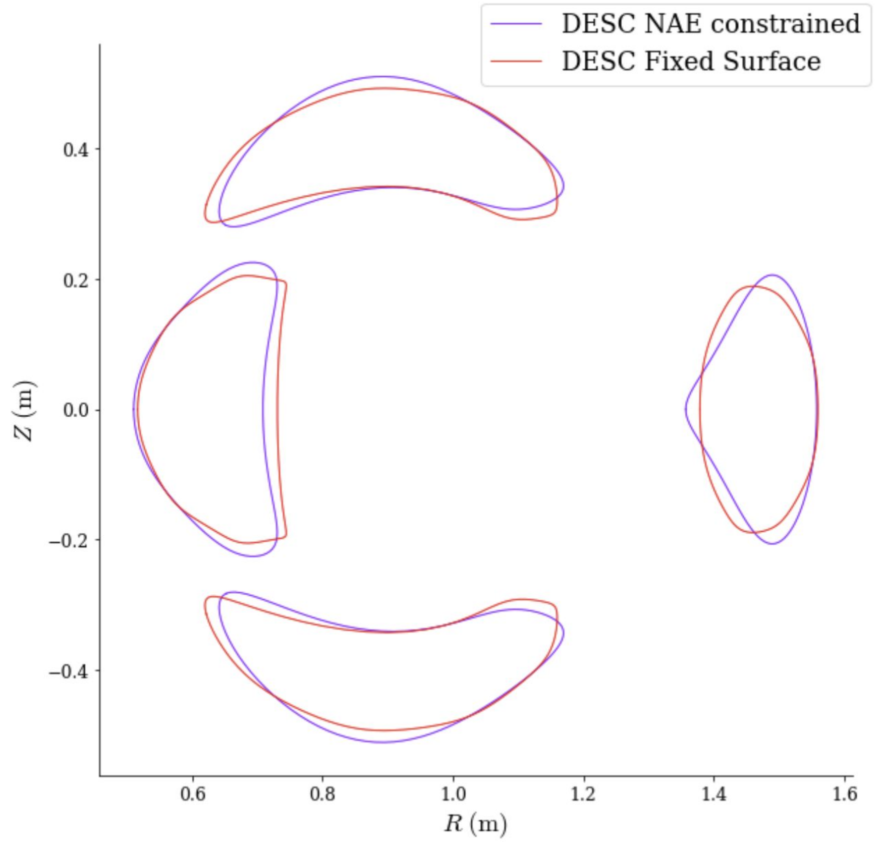
Closer look at flux surfaces near axis for difficult NAE (from E. Rodriguez)

```
rc = [1, 0.426, 0.044, -6.3646383583351e-11,
2.851584586653665e-05, 3.892992983405039e-08]

zs = [0.0, 0.4110168175146285, 0.04335427796015756,
6.530936323433338e-05, 1.3623898672936873e-05,
1.1620514629503932e-05]

etabar=1.64209358
B2c = 0.11293987662545873
B0=1
nfp = 4

qsc = Qsc(rc=rc, zs=zs, B0=B0, nfp=nfp, I2=0, B2c = B2c,
etabar=etabar, order = "r1", nphi = 201)

desc_eq= Equilibrium.from_near_axis(qsc,r=
r,L=9,M=9,N=N,ntheta=ntheta)
```

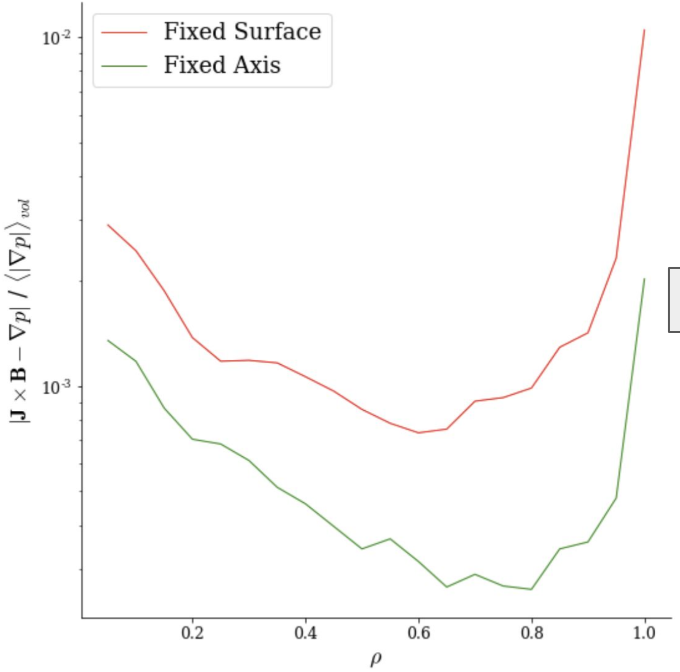


Closer look at LCFS for difficult NAE (from E. Rodriguez)

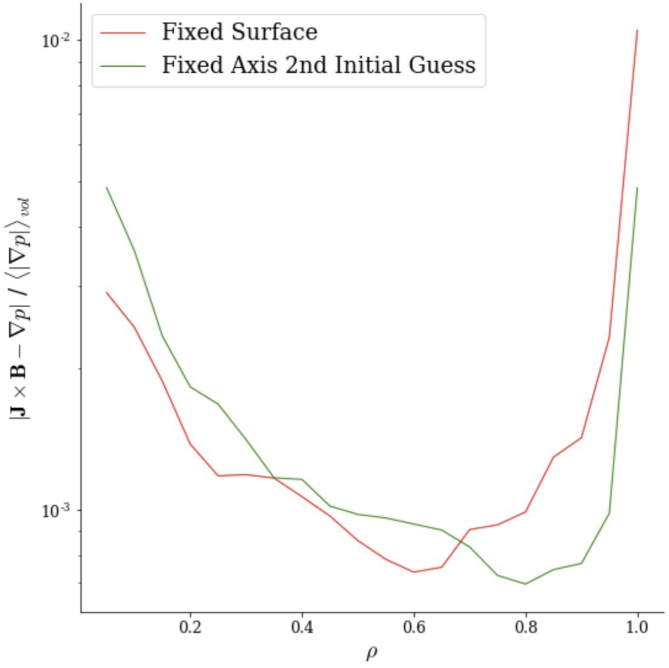
```
rc = [1, 0.426, 0.044, -6.3646383583351e-11,
2.851584586653665e-05, 3.892992983405039e-08]

zs = [0.0, 0.4110168175146285, 0.04335427796015756,
6.530936323433338e-05, 1.3623898672936873e-05,
1.1620514629503932e-05]

etabar=1.64209358
B2c = 0.11293987662545873
B0=1
nfp = 4

qsc = Qsc(rc=rc, zs=zs, B0=B0, nfp=nfp, I2=0, B2c = B2c,
etabar=etabar, order = "r1", nphi = 201)

desc_eq= Equilibrium.from_near_axis(qsc,r=
r,L=9,M=9,N=N,ntheta=ntheta)
```



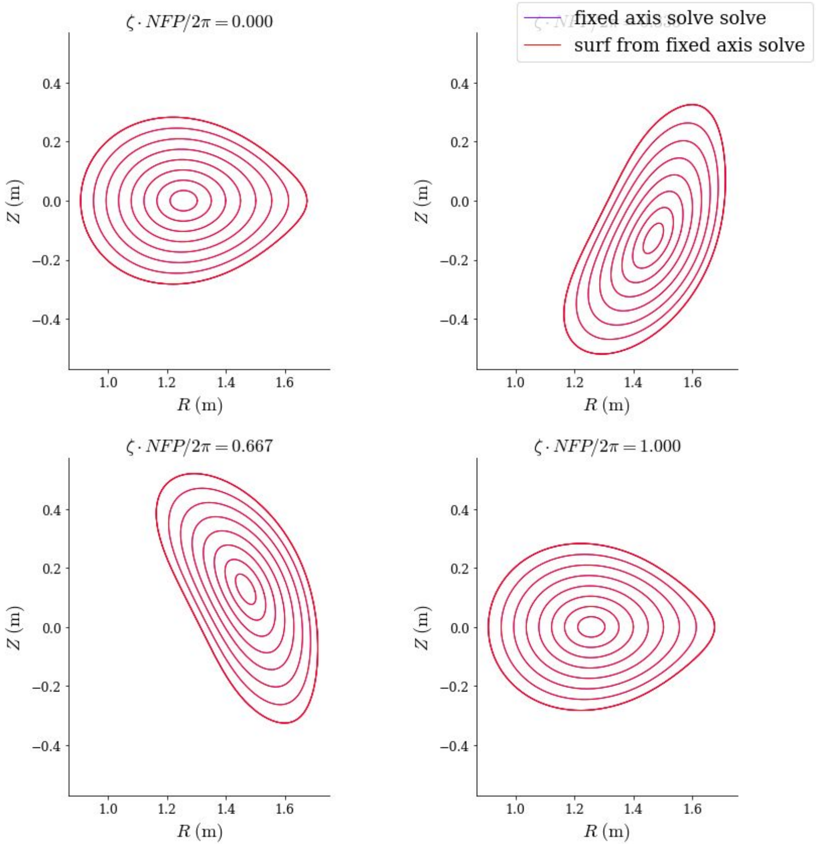
O(ϱ^0) (axis) Constraint in DESC - Under-constrained Problem

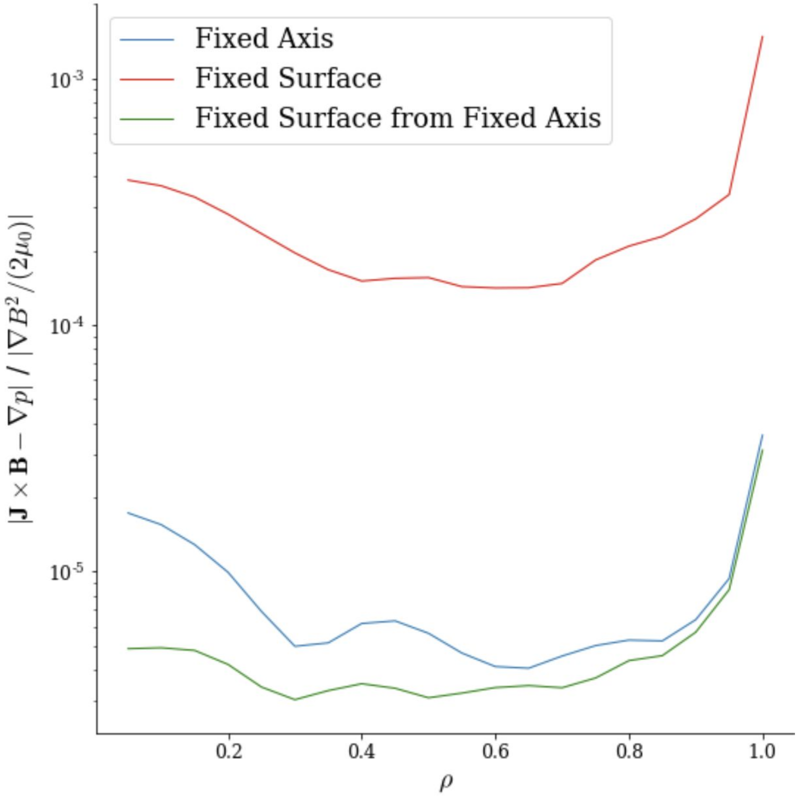


Different Initial Guess



Constraining with Surface from Fixed Axis solve yields same Surfaces





NAE Constraints in DESC

- First order are implemented
    - yields better agreement with NAE near the axis, but does not give better force error than a fixed-surface solve
- second order also implemented
    - requires larger number of toroidal harmonics to describe R_n, Z_n from NAE, since they decay slower with N