Flexible and Fast Stellarator Optimization with DESC Suite

E. Kolemen, Princeton University

Daniel Dudt, Rory Conlin, Dario Panici, Rahul Gaur,
Patrick Kim, Kaya Unalmis, Eduardo Rodriguez, Alan Goodman, Aza Jalalvand

DESC

Princeton Plasma Control
control.princeton.edu

Postdoc Opportunity!

- **Novel Stellarator Designs**
- **Stellarator/Divertor Topology**
- **Plasma SOL**
- **Plasma, Applied Math, CS welcome**
- **Work on DESC**

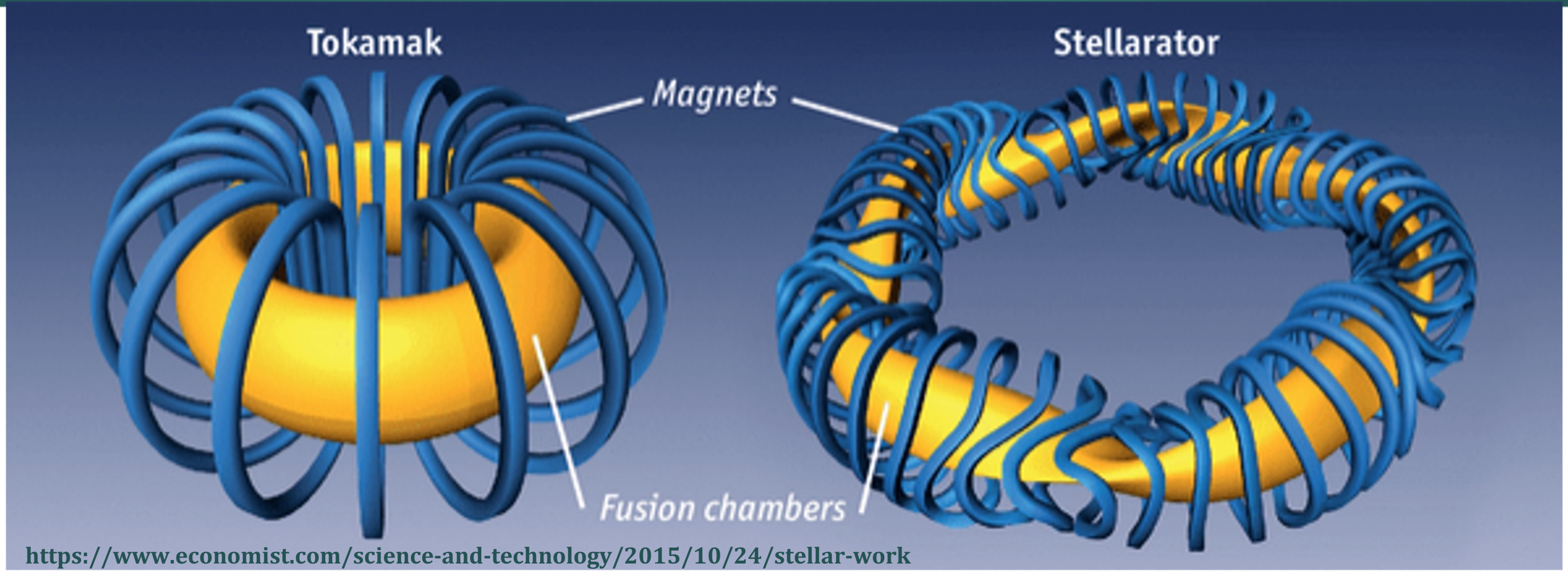
- **Email: ekolemen@princeton.edu (or talk to me)**

Tokamak Disruption (C-Mod)

Plasma Control

youtu.be/CUfR819hlDg

Magnetic Confinement Devices: Tokamak vs. Stellarator



Tokamaks

- **Axisymmetric**
  - Simpler geometry
  - Guaranteed particle confinement
- **Requires substantial plasma current**
  - Must be driven
  - NOT steady state!
- **Current leads to instabilities -> disruptions**

Stellarators

- **Inherently 3-D**
  - Complex geometry and coils
  - Confinement not guaranteed
    - certain fields exist which recover this (Quasisymmetry)
  - **Larger design space**
- **Does not need plasma current**
  - Steady state
  - No disruptions

What is the ideal way to optimize stellarators?

- **Constraints g(x):**
  - **MHD equilibrium**
  - ***Physicist insight: Analytical calculations (e.g. NEA)***
  - ***Engineer insight: e.g. A<5, …***
- **Objectives f(x):**
  - **Quasi-symmetry**
  - **Turbulence**
  - **…**
- ***Physicist/engineer insight: relative importance of f(x)***

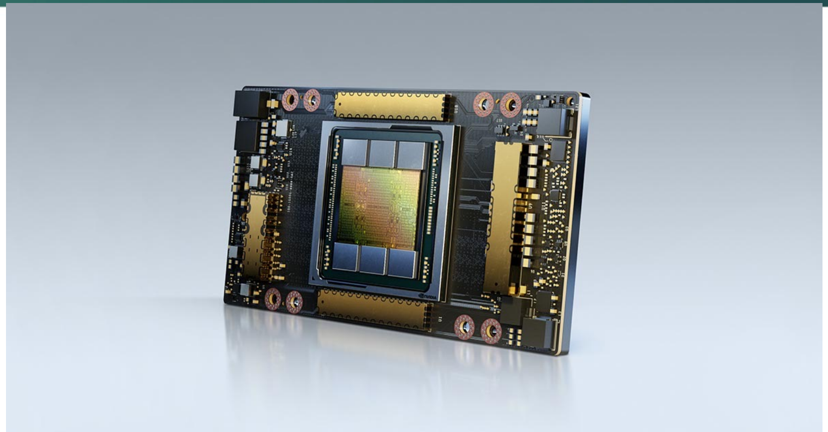
Plasma Control

Then Call A Fast Code

$$\min_x f(x)$$

$$\text{subject to} \quad g_{eq}(x) = 0$$
$$g_{ineq}(x) \geq 0$$

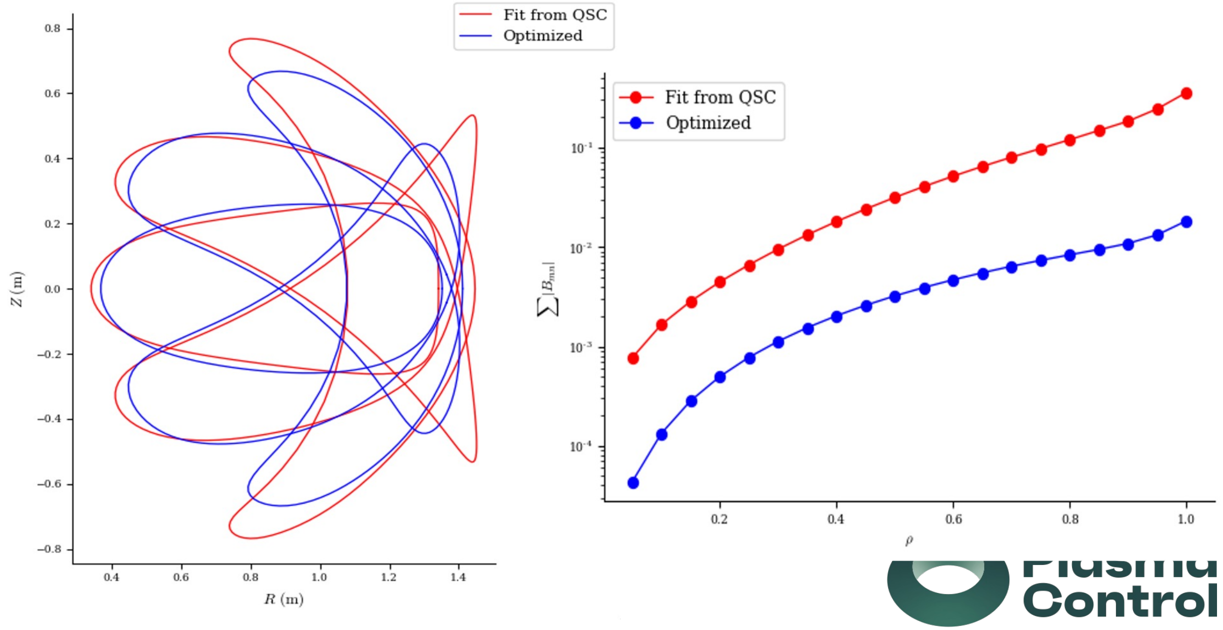
→



Fast= GPU + Jacobian

Constraints: $g_{physicist}$ =Fix NEA
+ g= MHD Eq.
Optimize remaining volume

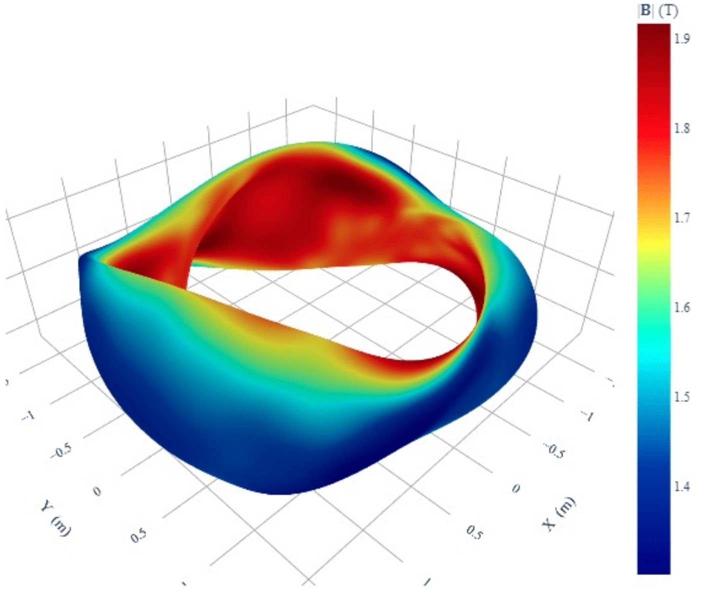
→



Plasma Control

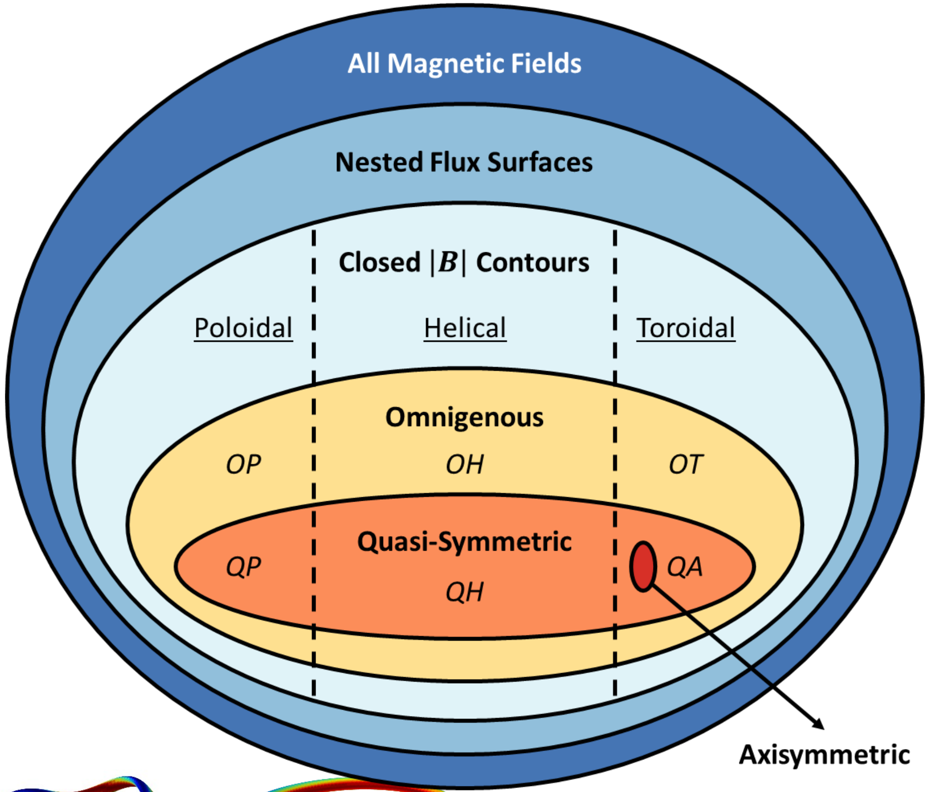
DESC is a new tool for stellarator optimization

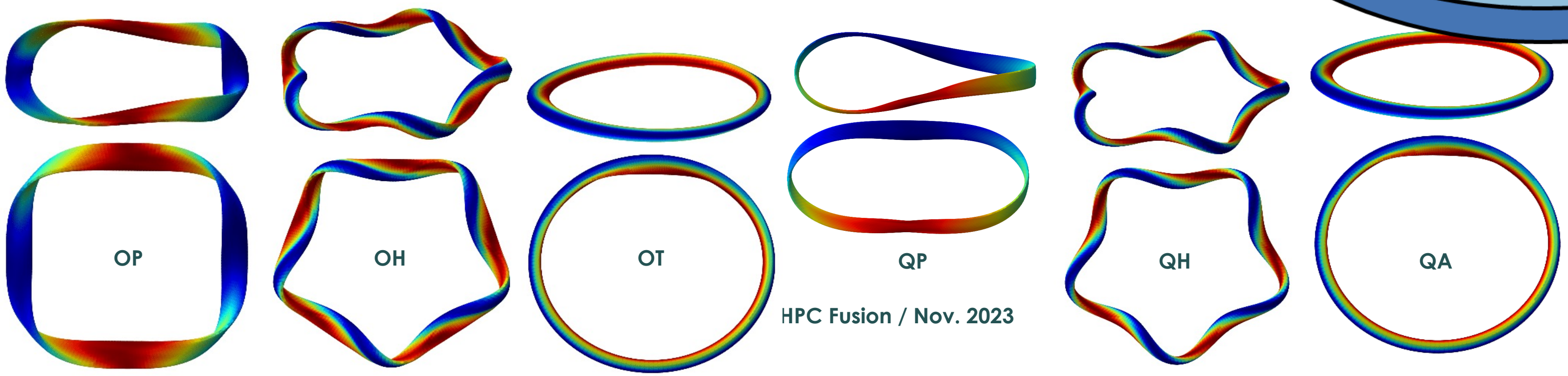
Accurate Equilibria



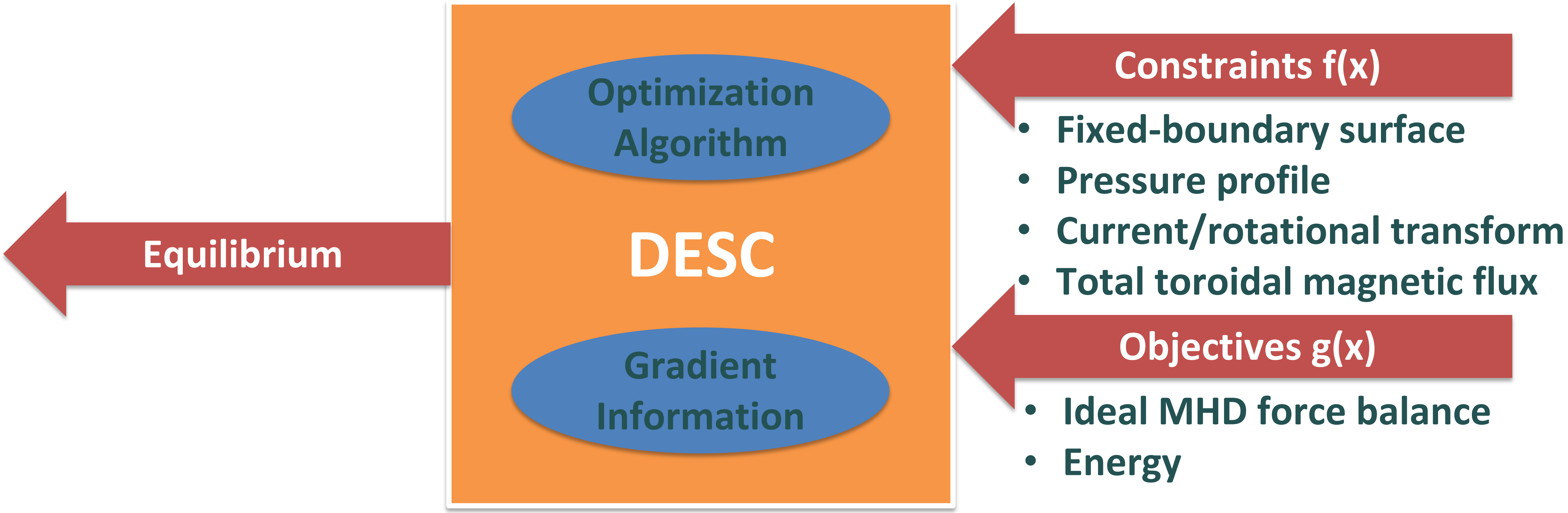
- **Stellarator equilibria are complicated**
- **Design space is much larger than tokamaks**

Fast Optimization





A flexible stellarator optimization suite



A flexible stellarator optimization suite

DESC

- Optimization Algorithm
- Gradient Information

← Optimized Stellarator

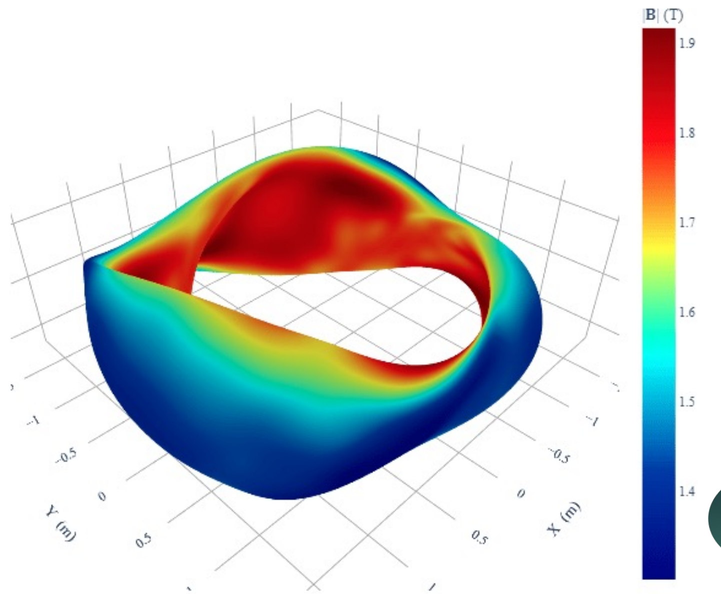
Constraints f(x) →
- Ideal MHD force balance
- Equilibrium profiles
- Some boundary modes

Objectives g(x) →
- Quasi-symmetry
- Mercier stability
- Aspect ratio
- etc.

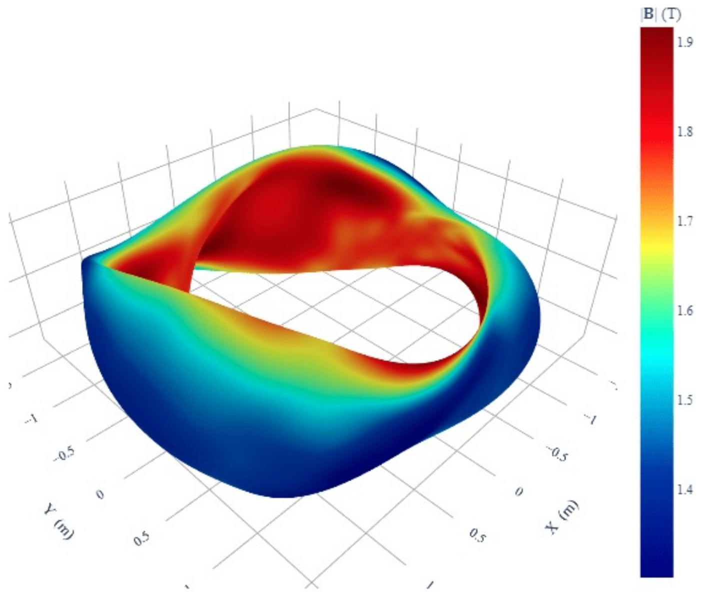
DESC Developed with the following design principles:

1. **Simple user interface**
   - Open-source Python code
   - Well documented
   - High test coverage
   - Easy to install
2. **Local error quantification**
   - Pseudo-spectral (collocation) methods
3. **Properly resolve the magnetic axis**
   - Global basis functions
   - Zernike polynomials
4. **Exact derivatives of all objectives**
   - Automatic differentiation (JAX)
5. **Hardware agnostic**
   - Run on CPUs, GPUs, and TPUs
6. **Extendable to new applications**
   - Modular & flexible code structure

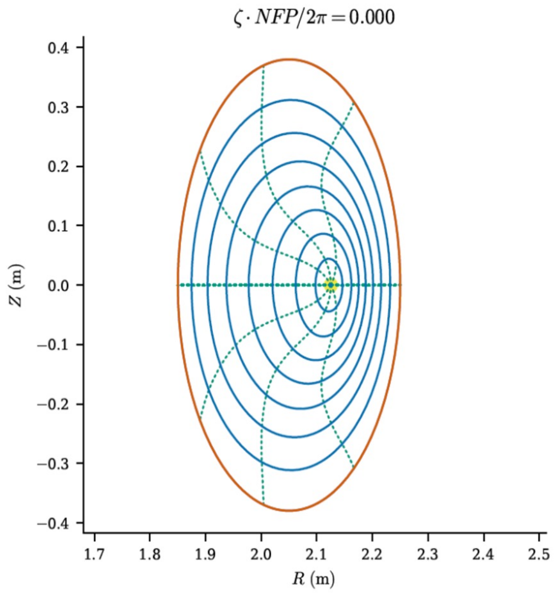
https://github.com/PlasmaControl/DESC

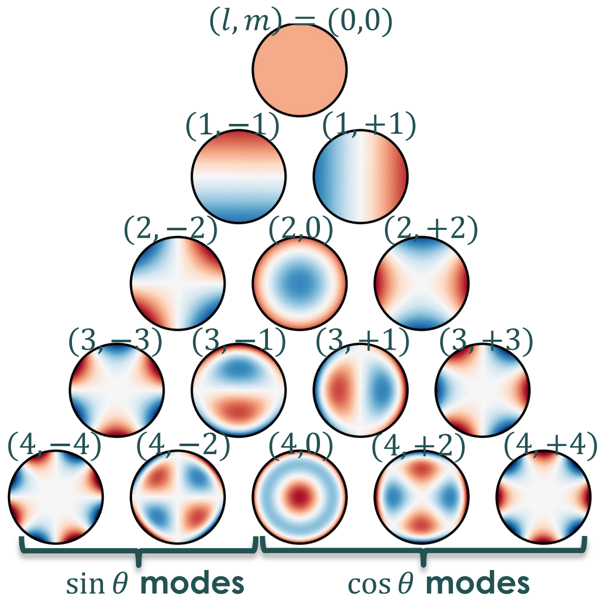


Zernike spectral basis better than concentric circle basis



Poincare section





spectral coefficients

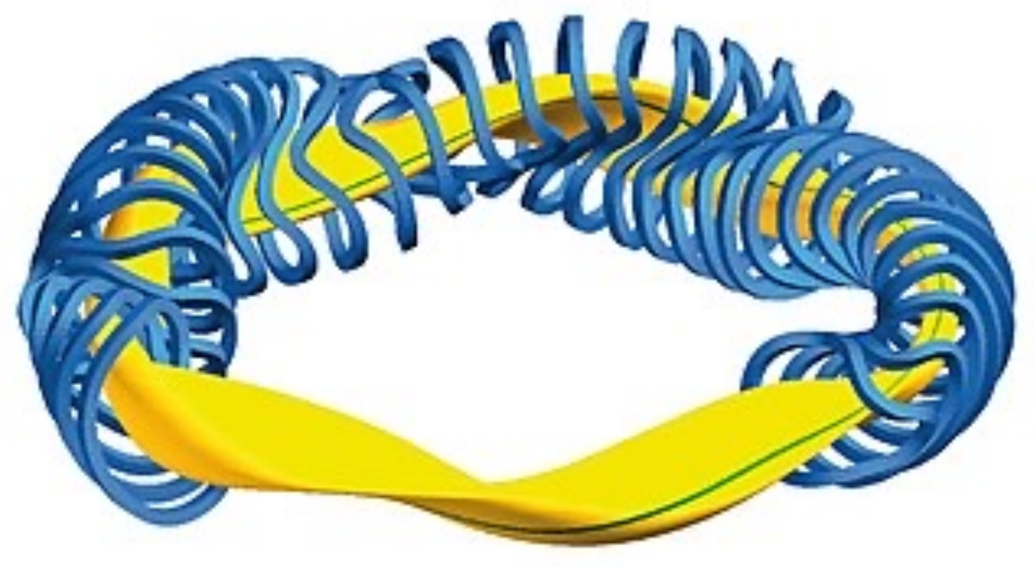
Zernike polynomials

$$X(\rho,\theta,\zeta) = \sum_{lmn} X_{lmn} \mathcal{Z}_l^m(\rho,\theta)\mathcal{F}^n(\zeta)$$

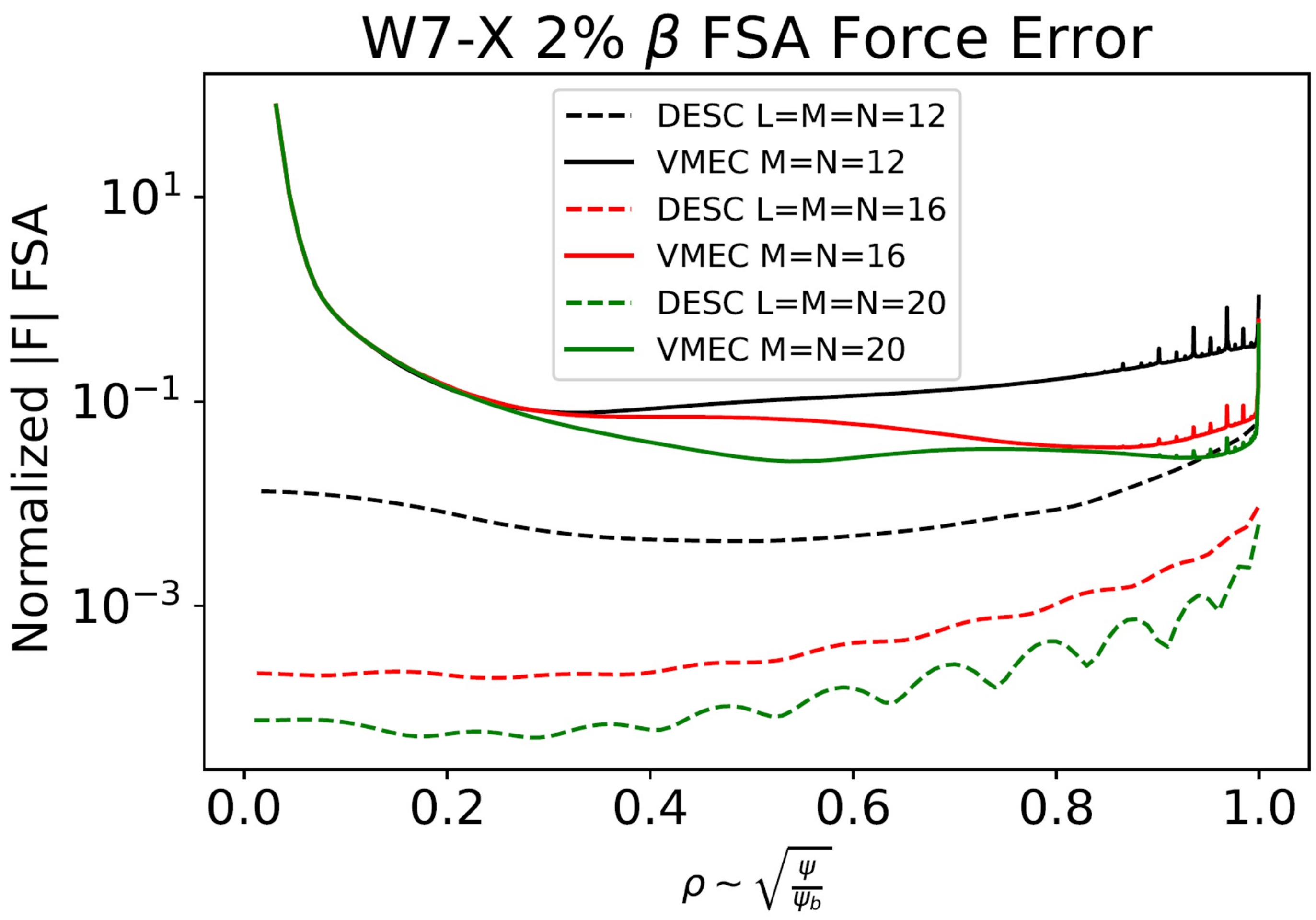
Fourier series

- **Satisfies analyticity conditions at the magnetic axis**
- **Exponential convergence (if solution exists and is smooth)**

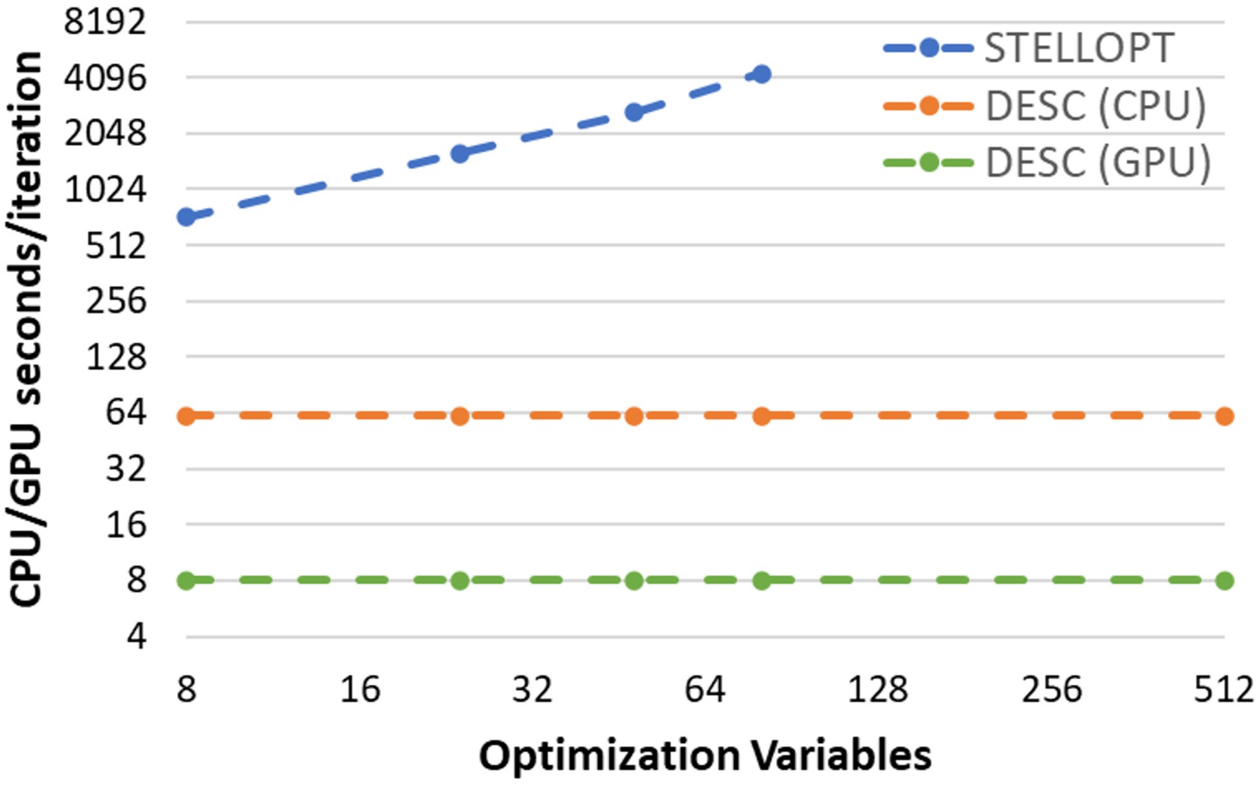
DESC spectral methods yield more accurate equilibrium solutions



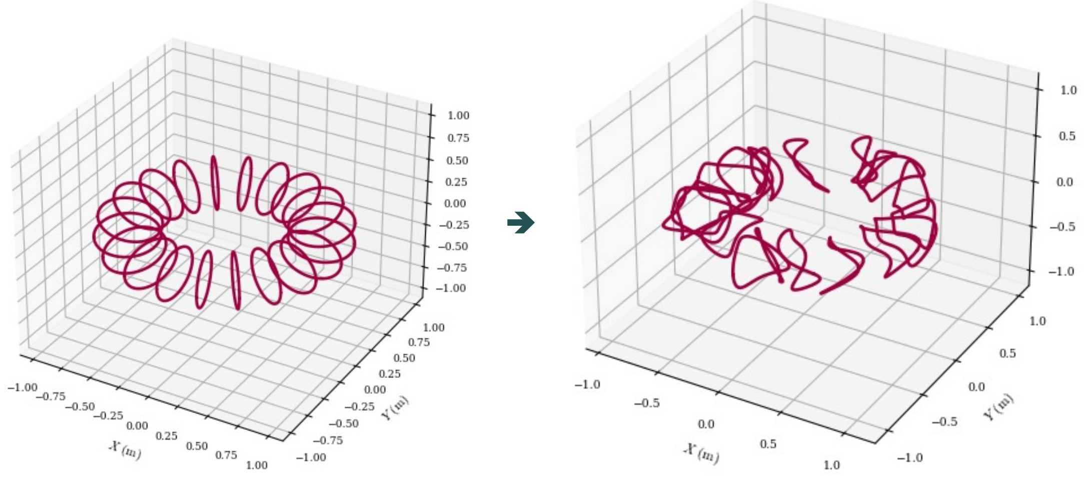
W7-X



DESC Allow Much Faster Stellarator Optimization



Example Coil Optimization



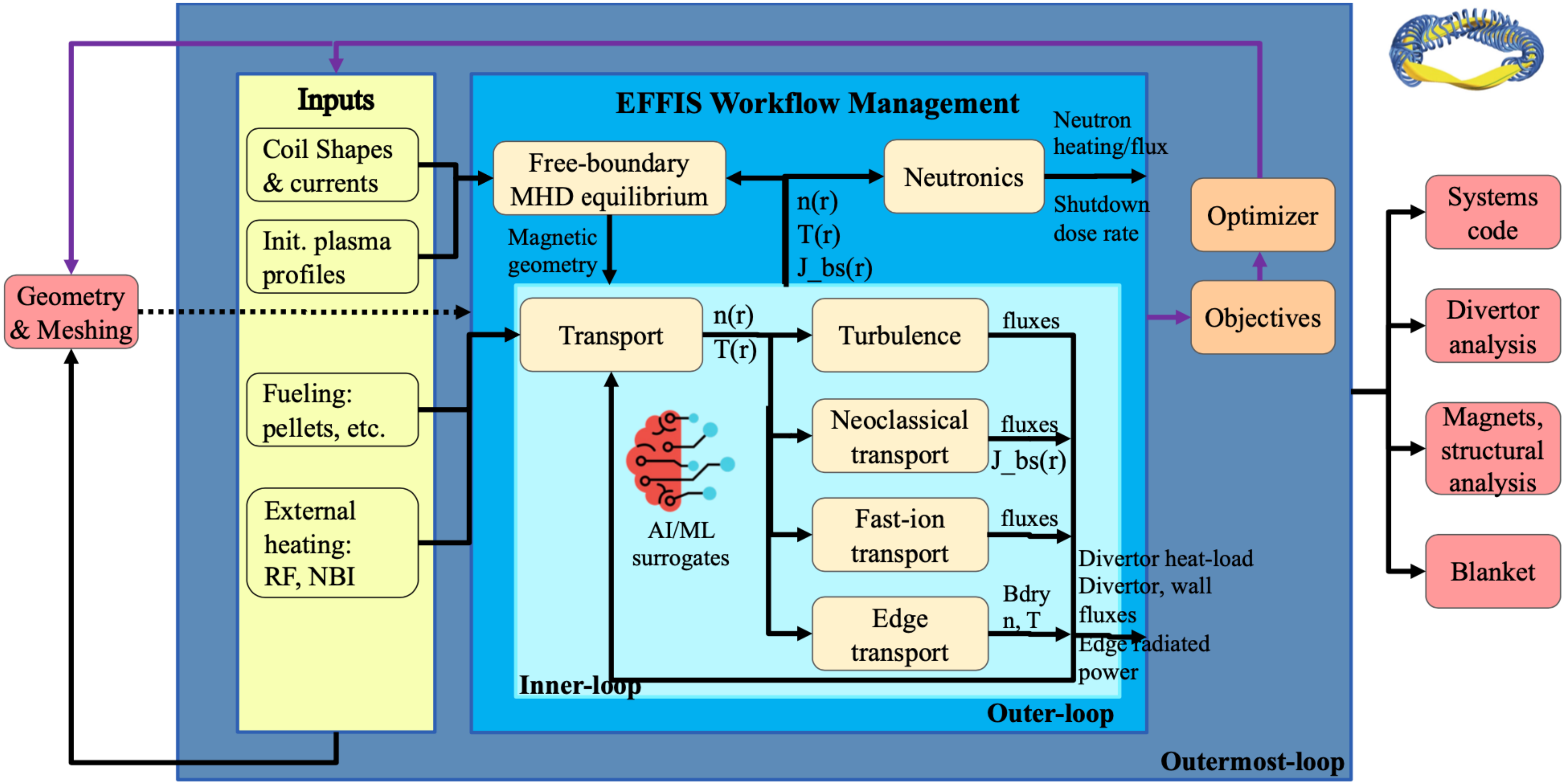
W7-X $\beta = 2\%; L = 24, M = N = 12$

Hardware	Run Time
Intel Cascade Lake CPU	48 min
NVIDIA A100 GPU	20 min

https://github.com/PlasmaControl/DESC

Plasma Control

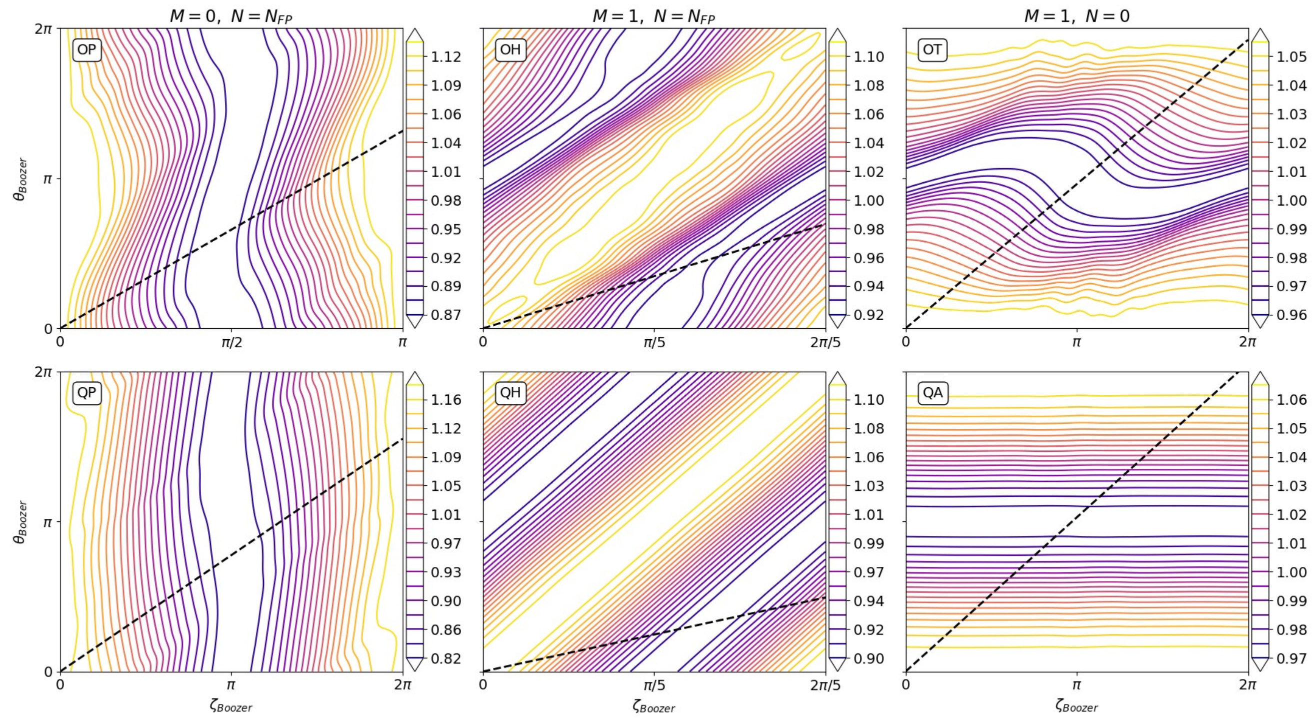
Next up: End-To-End Stellarator Optimization with AutoDiff

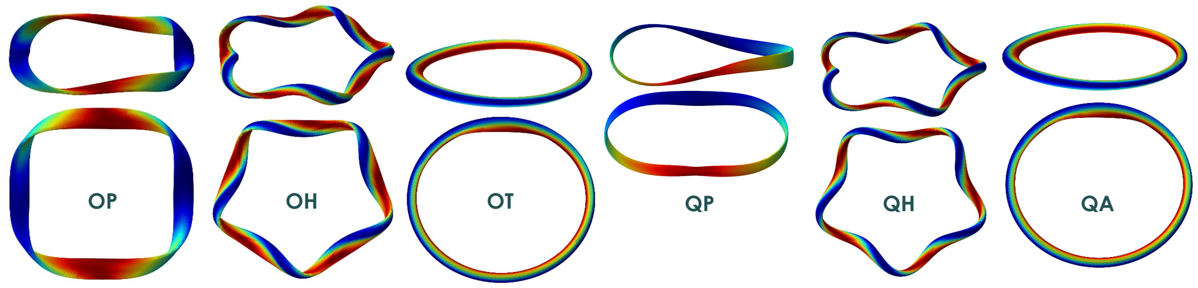


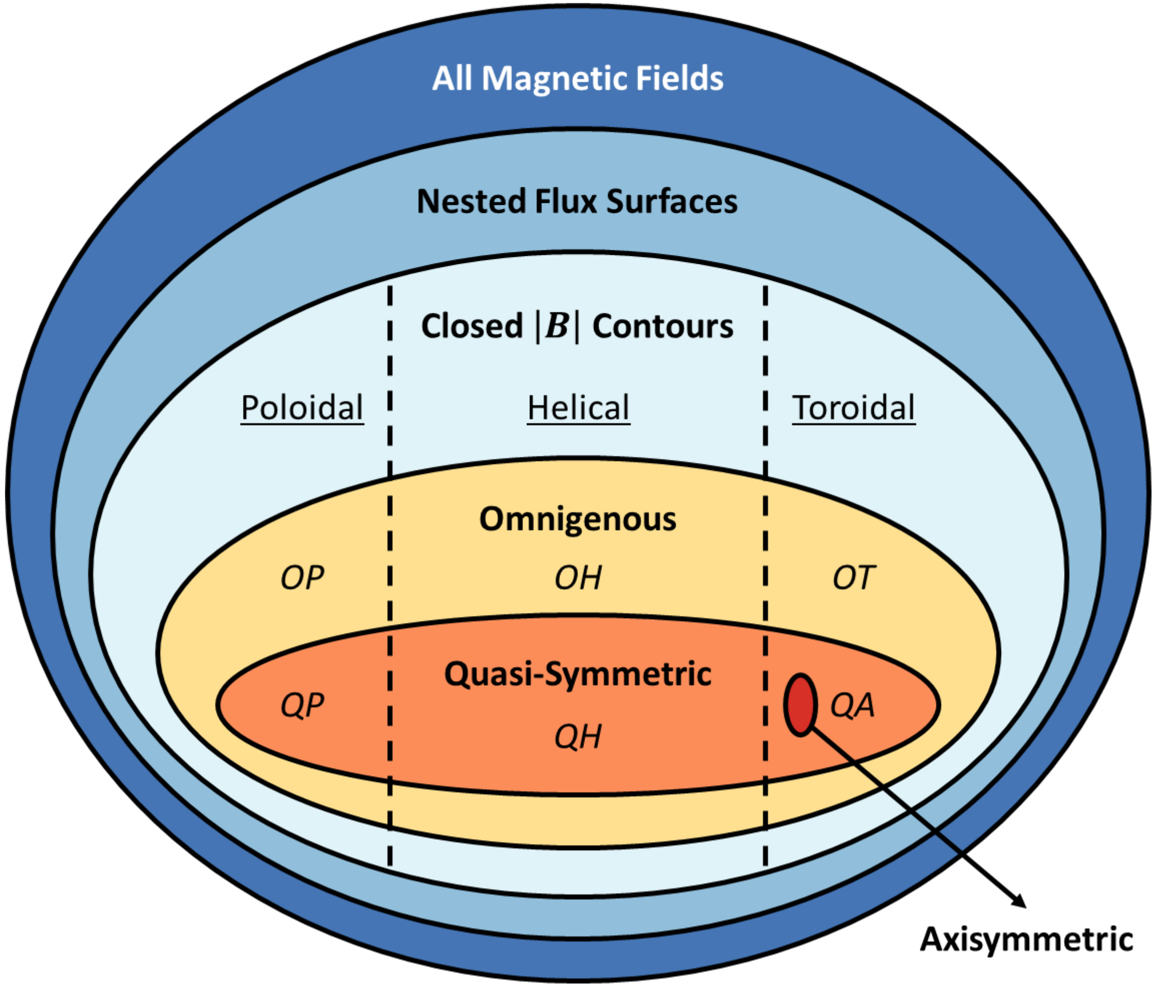
Scidac work M. Churchill PPPL

Plasma Control

Main Result: General Omnigenity Easily Obtained with DESC







Current optimization methods: Sum of Squares

Combine equality + inequality constraints

$$\min_x f(x) + w_1[g(x)]^2$$

Choose small weight for inequality constraints to enforce "approximately"

Choose large weight for equality constraints to penalize a lot

Limitations:

- **Hard to guess a-priori what weights should be**
- **Even small weights for "inequality" constraints can overly penalize things we don't care about**

Better methods: Augmented Lagrangian

- Combination of traditional Lagrangian + quadratic penalty

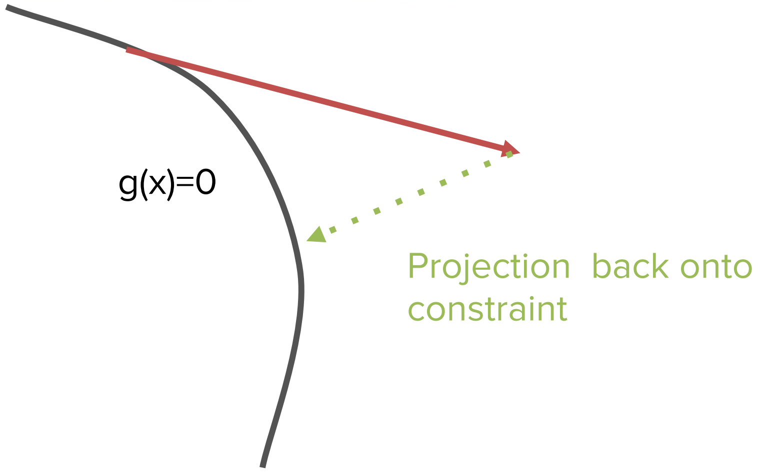
$$\mathcal{L}(x, \lambda, \mu) = f(x) + \lambda^T \mathbf{g}(x) + \mu g^2(x)$$

- Doesn't introduce any non-smooth terms
- "Exact" method - doesn't need μ → infinity
- Solve sequence of subproblems for increasing μ, λ
- Provides estimate of true Lagrange multipliers - useful information about trade-offs

- Open source packages available (LANCELOT, NLopt, etc). Also python/JAX version implemented in DESC

DESC Allow Combined Constraints + Optimization

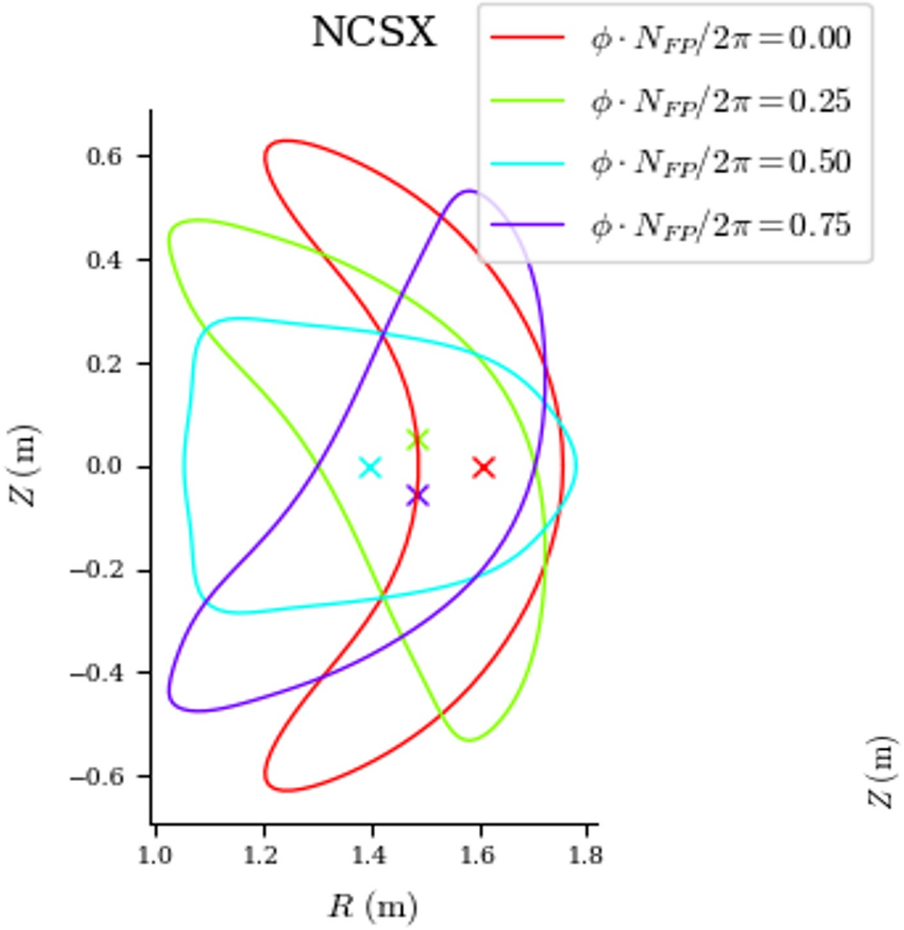
```
while optimization stopping criteria are not met:

    perturb equilibrium solution to improve objective

    while equilibrium stopping criteria are not met:

        solve equilibrium force balance
```



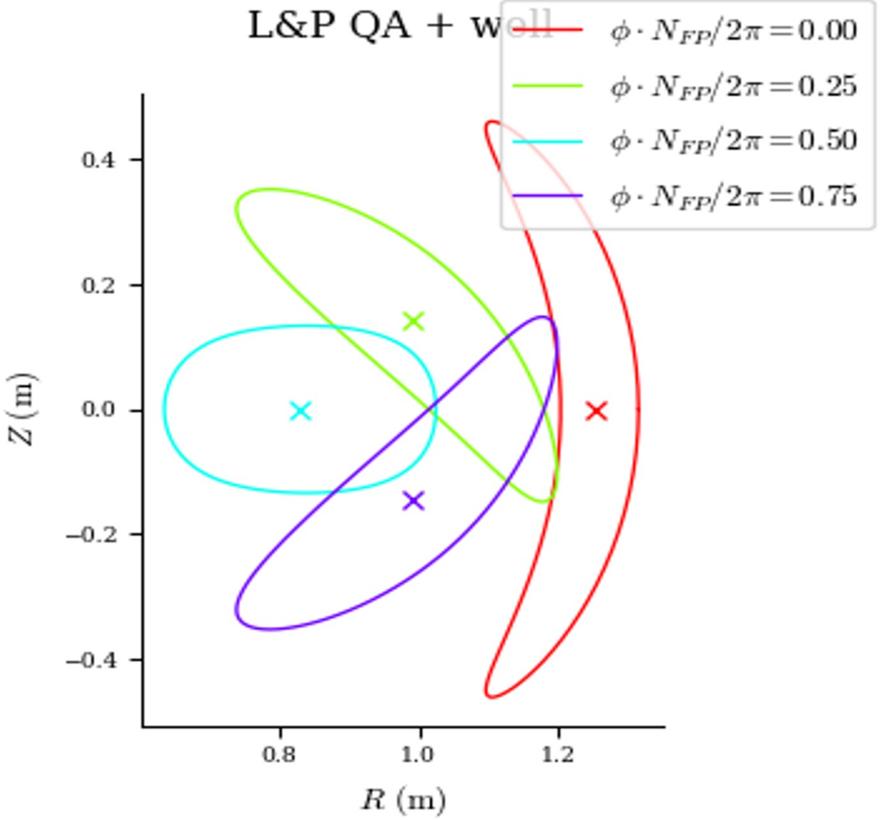
Find x_{k+1} = min $L_A(x,\lambda_k,\mu_k)$ → **Calculate constraint violation $c(x_k)$ and check tolerances** → **Depending on $c(x_k)$, update λ_k or μ_k** → (back to Find x_{k+1})

Plasma Control

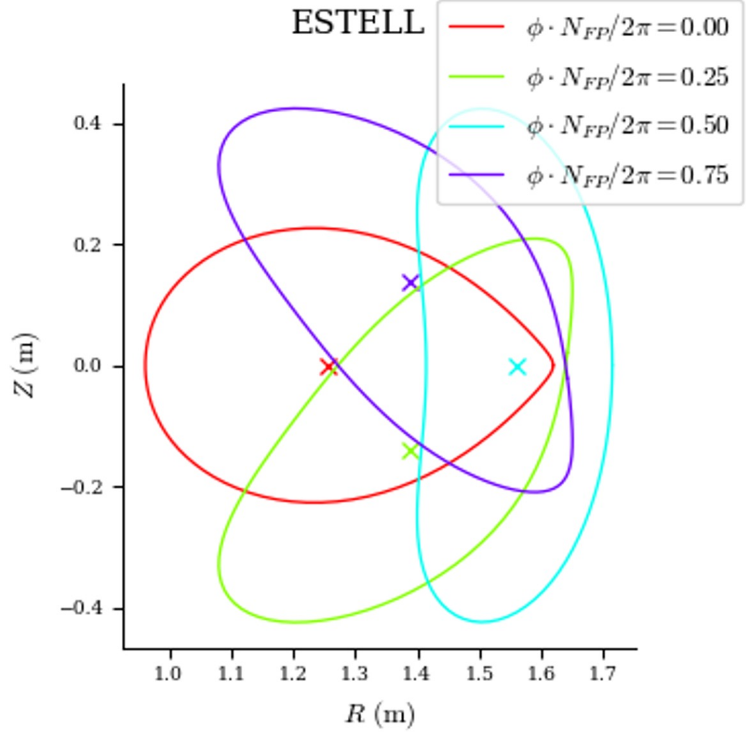
Application of Multi-Objective Optimization in DESC



NCSX + L&P precise QA: Magnetic well, but large concave regions



ESTELL: simpler geometry, but magnetic hill



Plasma Control

Application of Multi-Objective Optimization in DESC

Minimize f_{QS} ("two term" metric) subject to

Force balance: $$\mathbf{J} \times \mathbf{B} - \nabla p = 0$$

Iota: $$0.41 \leq \iota \leq 0.43$$

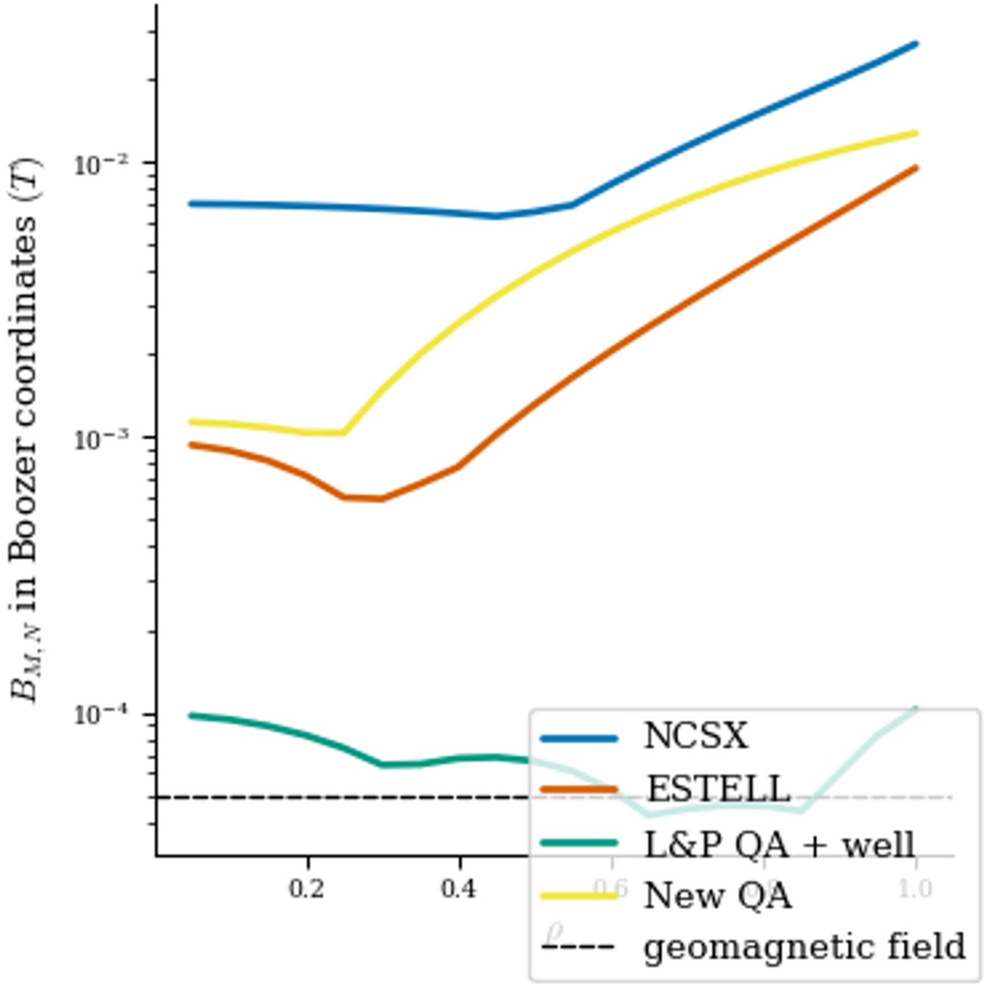
Magnetic Well: $$\frac{\partial_\rho V (2\mu_0 \partial_\rho p + \partial_\rho \langle B^2 \rangle)}{V \langle B^2 \rangle} > 0$$

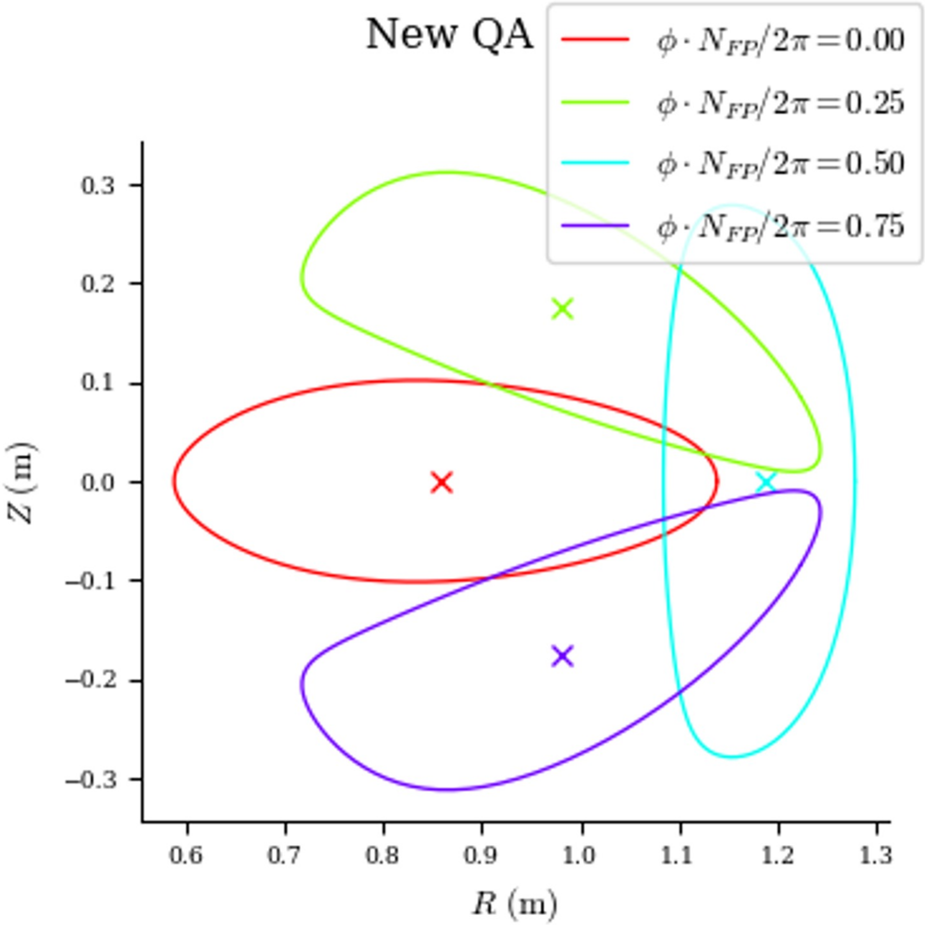
Mean Curvature: $$H = (\kappa_1 + \kappa_2)/2 < 0$$

κ_1, κ_2 eigenvalues of second fundamental form

Solve with least squares augmented Lagrangian method in DESC

QS better than NCSX + stability without concavity





Plasma Control

Free Boundary DESC

Free boundary constraints:

Flux surface:

$$\mathbf{B} \cdot \mathbf{n} = 0$$

Pressure balance:

$$B_{\rm out}^2 - B_{\rm in}^2 - 2\mu_0 p = 0$$

Tangential jump:

$$\mathbf{n} \times [\mathbf{B}_{\rm out} - \mathbf{B}_{\rm in}] - \mu_0 \mathbf{K} = 0$$

B_{in} = B from fixed boundary DESC

$B_{out} = B_{coil} + B_{VC} + B_K$

- **Parameterize unknown sheet current K, along with R, Z**
- **Use high order singular integration scheme (Malhotra et al 2019) to compute virtual casing + sheet current field $B_{VC} + B_K$**
- **Minimize all 3 equations simultaneously**

Free Boundary DESC

Free boundary constraints:

Pressure balance:

$$B_{\mathrm{out}}^2 - B_{\mathrm{in}}^2 - 2\mu_0 p = 0$$

B_{in} = B from fixed boundary DESC

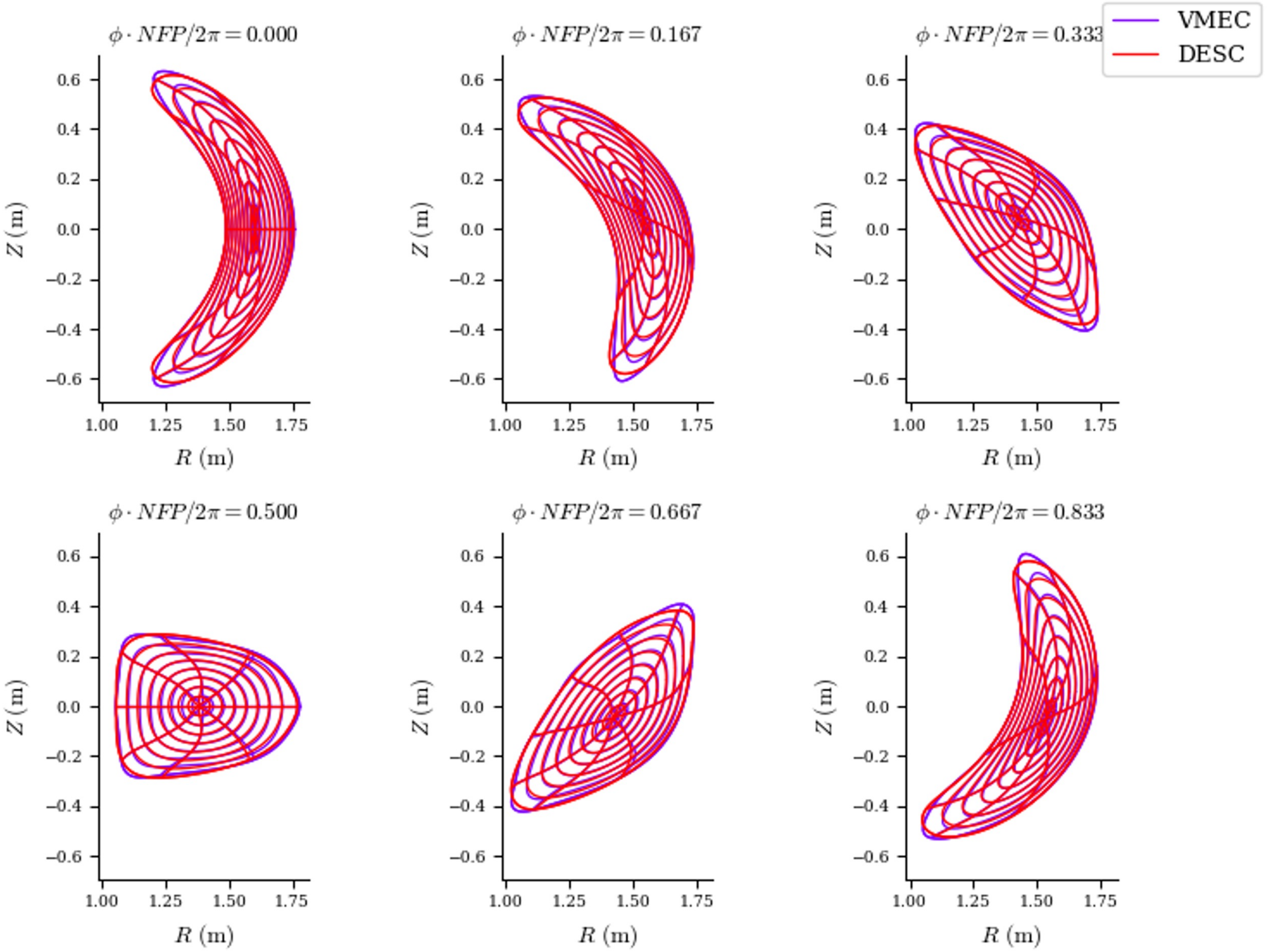
Virtual Casing: -nxB_{out} = K_{out}

Biot-Savart(K_{out})=B_{coil}

- **Alternatively:**
- **Use high order singular integration scheme (Malhotra et al 2019) to compute virtual casing for B_{out}**
- **Just check for pressure balance.**

Plasma Control

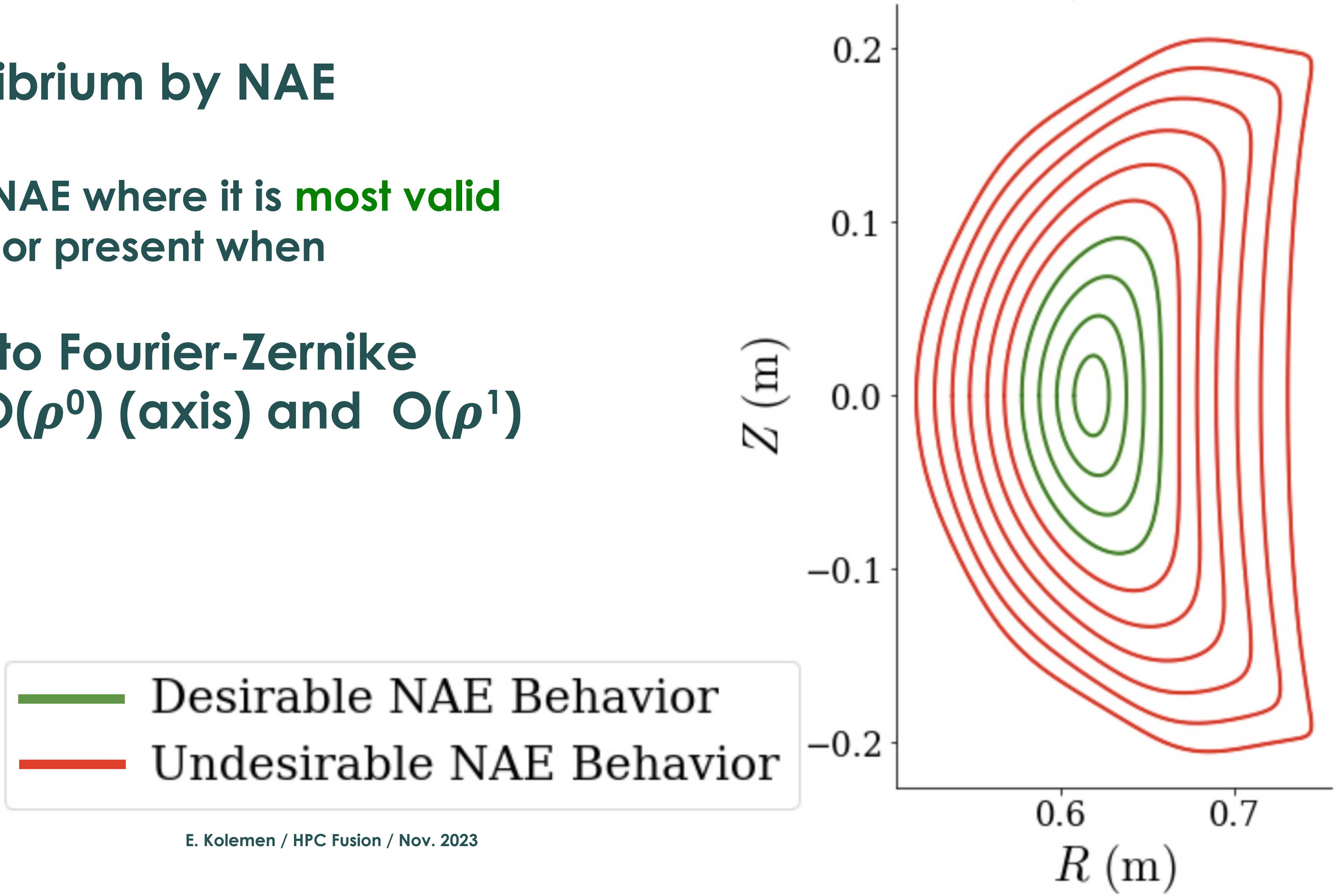
Free Surface DESC vs VMEC



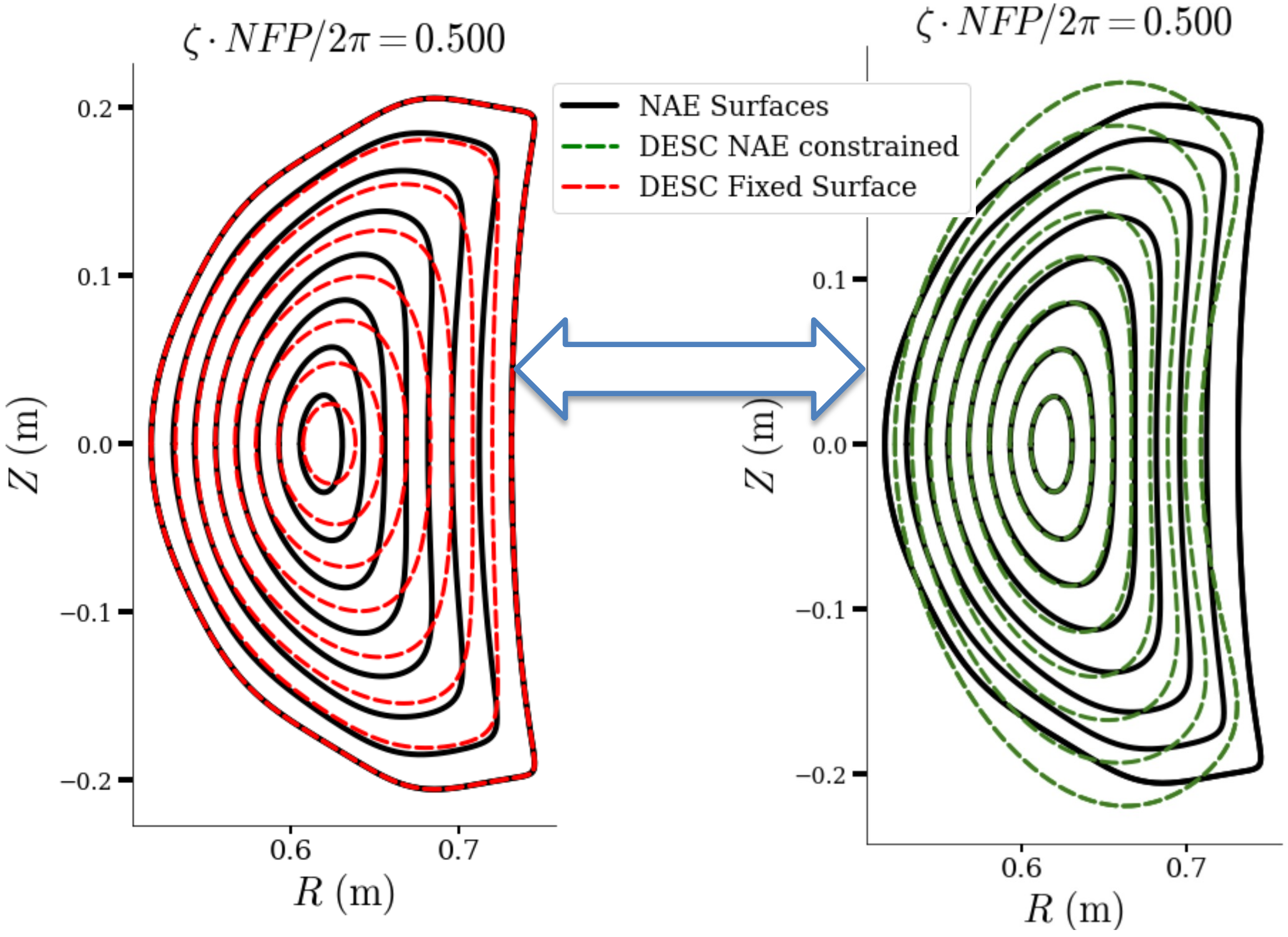
Near-Axis Expansion (NAE) Constraints in DESC

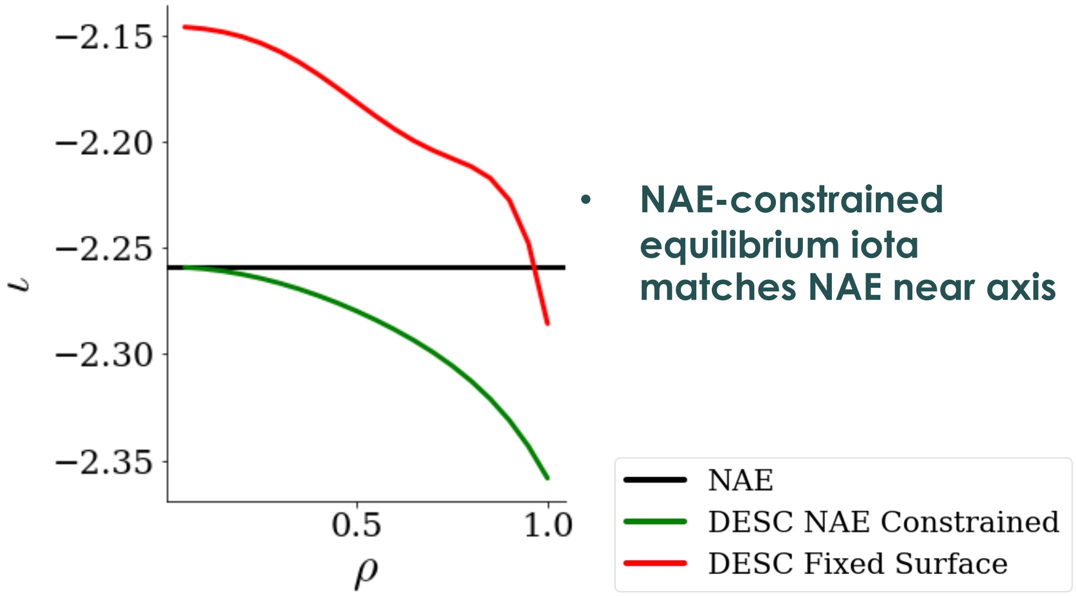
- Constrain global equilibrium by NAE behavior as $\rho \rightarrow 0$
  - Use information from NAE where it is most valid
  - Avoid singular behavior present when evaluating at large r
- Map NAE coefficients to Fourier-Zernike modes of DESC to fix $O(\rho^0)$ (axis) and $O(\rho^1)$ behavior

NAE equilibrium evaluated at r =0.1

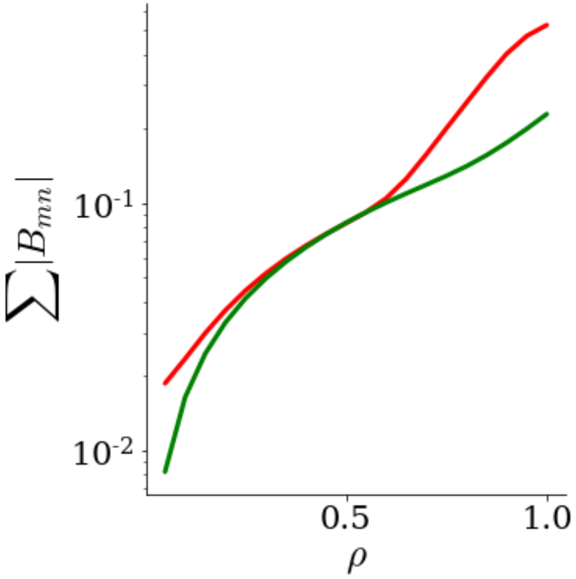


Near-Axis-Expansion Constrained Equilibria in DESC





- NAE-constrained equilibrium iota matches NAE near axis

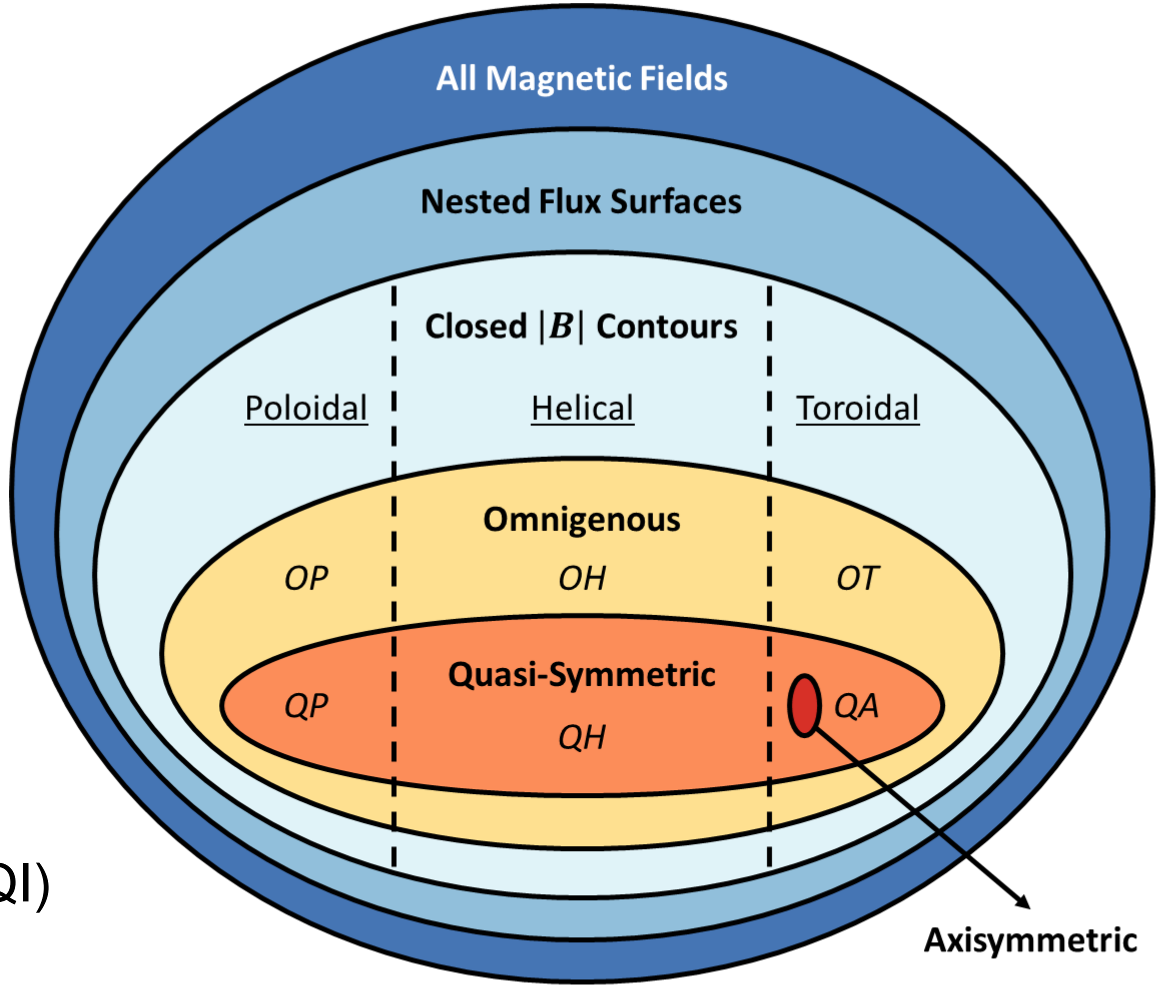


- NAE-constrained equilibrium maintains better QS near axis

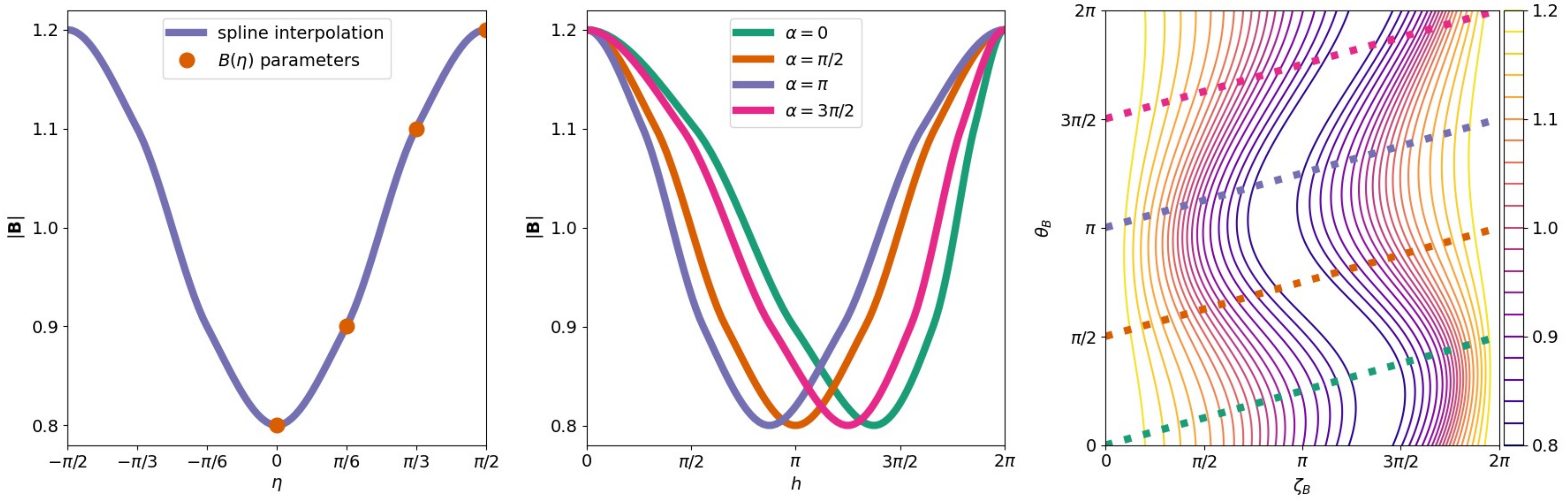
Plasma Control

General Omnigenity: confined particles without Quasi-Symmetry

- **Omnigenity:** the class of magnetic fields in which the bounce-averaged radial drifts of trapped particles vanish
  - $B = |\boldsymbol{B}|$ contours are closed curves
  - B_{max} contour is straight in Boozer coords
  - "Bounce distances" between consecutive B points are independent of the field line α
- Previous omnigenous equilibria were limited to:
  - Quasi-Axisymmetry (QA)
  - Quasi-Helical symmetry (QH)
  - Omnigenity with Poloidal contours (OP aka QI)
- General omnigenity = larger design space!



General omnigenity optimization is implemented in DESC



1. **Define a target magnetic well "shape" (in computational coordinate η)**
2. **Define a target "shift" on each field line (preserves constant bounce distances)**
3. **Optimize to minimize the errors: $B_{equilibrium} - B_{target}$**

Omnigenity is parametrized by a coordinate mapping

- Map between computational and Boozer coordinates:

helicity = (M, N)

$$h(\rho,\alpha,\eta) = h(\theta_B,\zeta_B)$$

$$2\eta + \pi + \sum_{l=0}^{L_\rho}\sum_{m=0}^{M_\eta}\sum_{n=-N_\alpha}^{N_\alpha} x_{lmn} T_l(2\rho-1)\mathcal{F}_m(\eta)\mathcal{F}_{nN_{FP}}(\alpha) = \begin{cases} N\zeta_B & \text{for } M = 0 \\ -\theta_B + \dfrac{N}{M}\zeta_B & \text{for } M \neq 0 \end{cases}$$

ρ = flux surface label
α = field line label
η = coord along field line

free parameters (x_{lmn}) — Chebyshev polynomials (T_l) — Fourier series ($\mathcal{F}_m$, $\mathcal{F}_{nN_{FP}}$)

- Constant bounce distances are preserved:

$$\delta \propto \Delta h = h(\rho,+\eta,\alpha) - h(\rho,-\eta,\alpha)$$

$$= 4\eta + \sum_{l=0}^{L_\rho}\sum_{m=0}^{M_\eta}\sum_{n=-N_\alpha}^{N_\alpha} x_{lmn}\left[T_l(2\rho-1)\mathcal{F}_{nN_{FP}}(\alpha)[\mathcal{F}_m(+\eta) - \mathcal{F}_m(-\eta)]\right]$$

$$= 4\eta$$

$$\therefore \frac{\partial\delta}{\partial\alpha} = 0$$

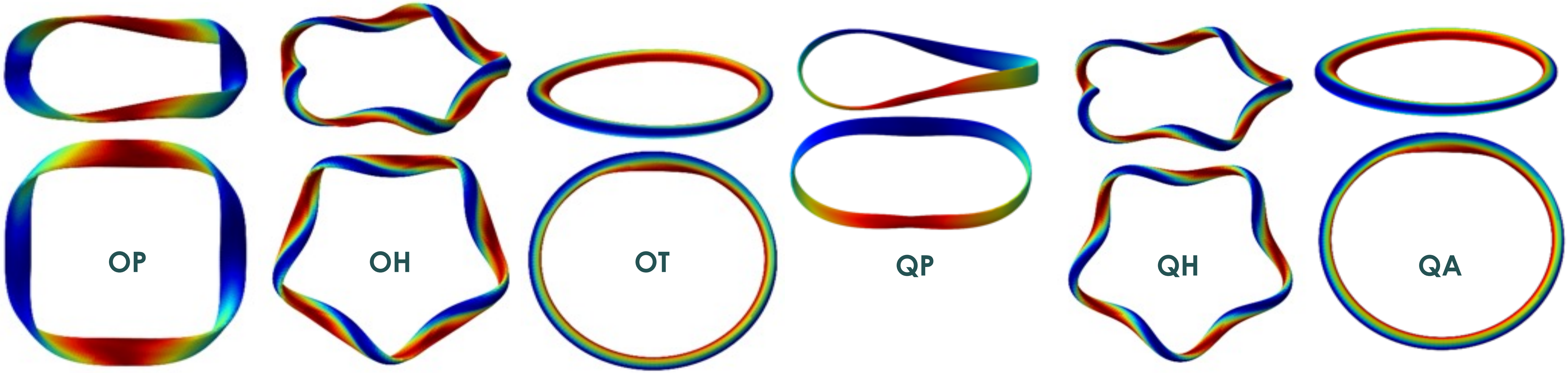
$= 0$ because $\sum_{m\geq 0}\mathcal{F}_m(\eta)$ is an even function of η

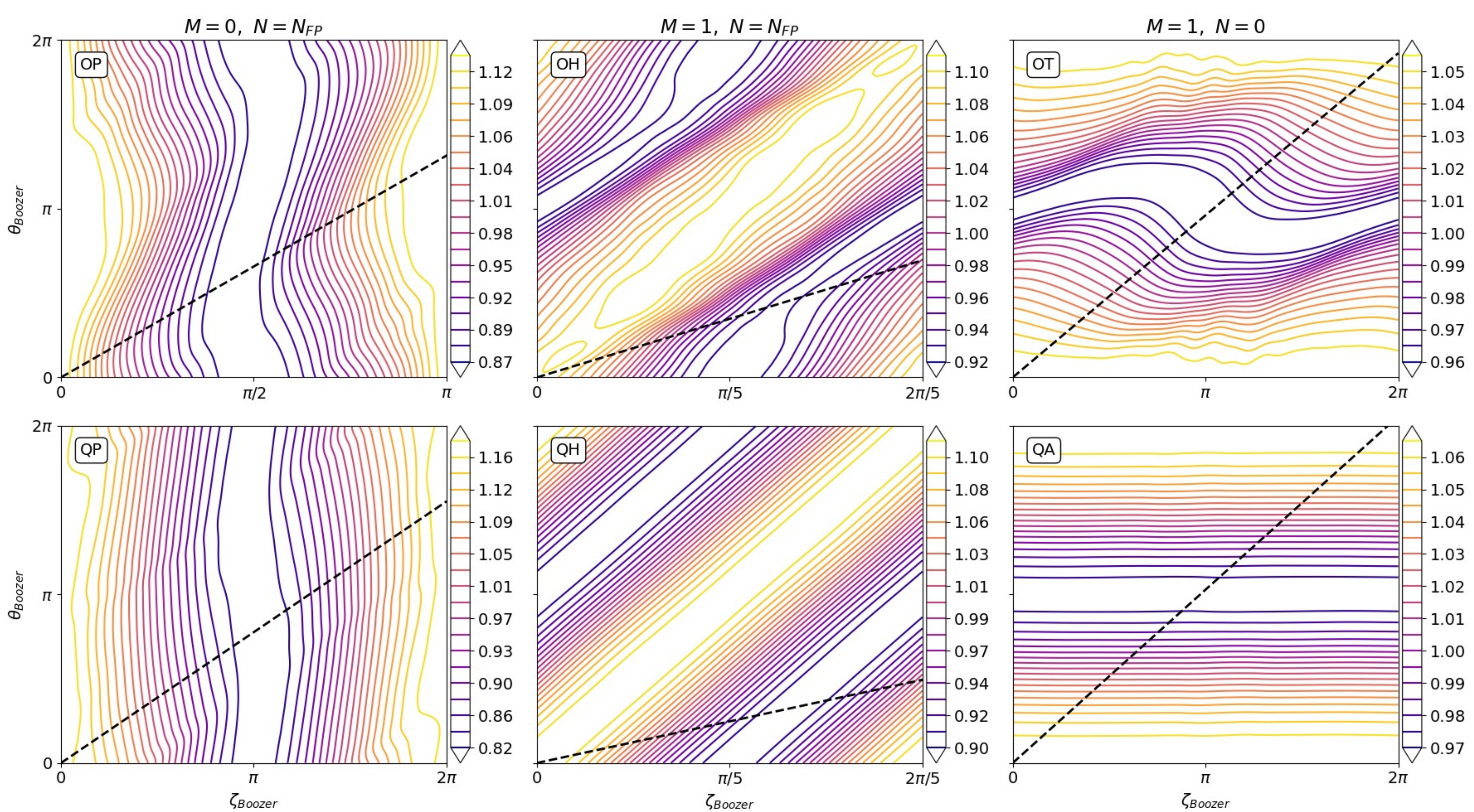
Examples of the different types of omnigenity

General Omnigenity with poloidally/helically/toroidally-closed B contours (OP, OH, OT)

Quasi-Symmetry is a subspace of omnigenity (QP, QH, QA)

- All solutions used near-axis expansions as an initial guess (from pyQIC and pyQSC)
- Optimized for omnigenity at five surfaces (no other objectives)
- Aspect ratios ≈ 20

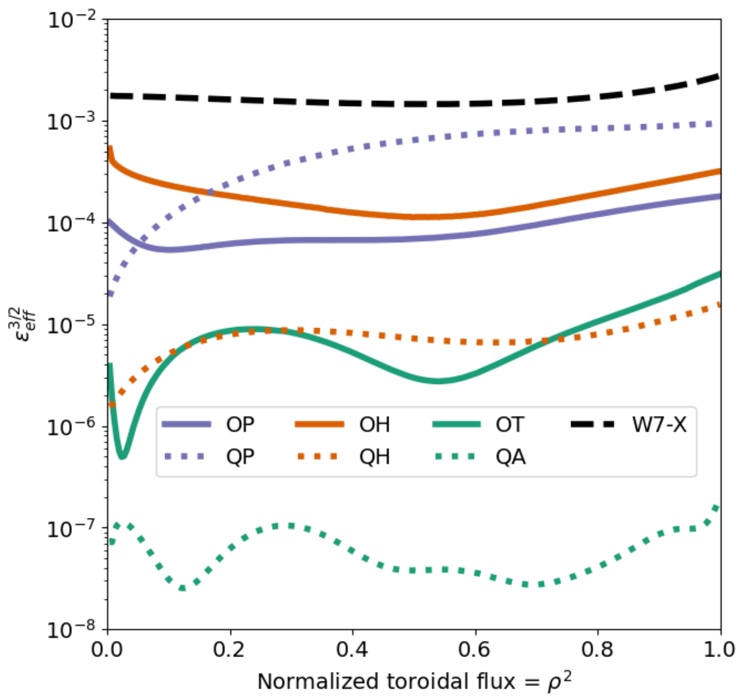


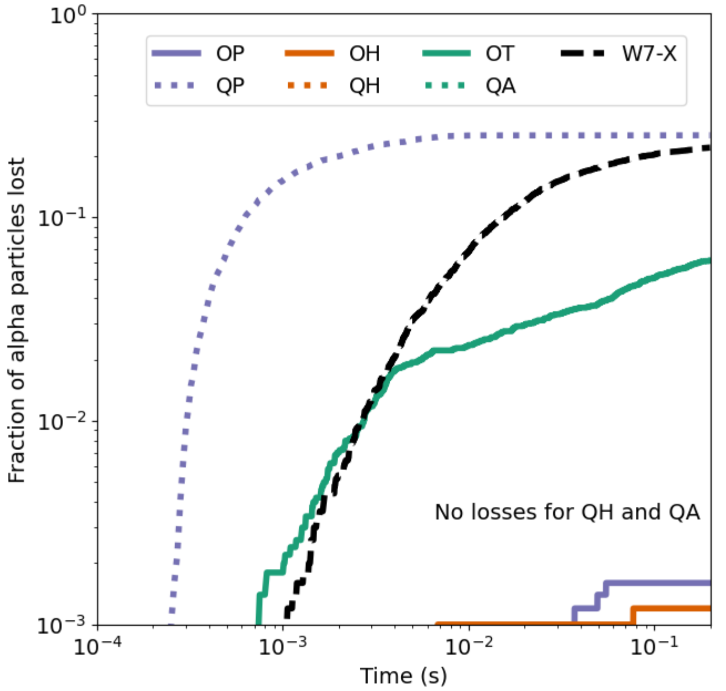


$M=0,\ N=N_{FP}$
$M=1,\ N=N_{FP}$
$M=1,\ N=0$
OP
OH
OT
QP
QH
QA
θ_{Boozer}
ζ_{Boozer}

Solutions show good particle confinement!

- **Top:** Neoclassical collisional transport magnitude
  - Computed by NEO
- **Bottom:** Collisionless losses of fusion-born alpha particles
  - Computed by SIMPLE
  - Particles initialized at $\rho = 0.5$
  - Configurations scaled to a and B of ARIES-CS
- Reference case is W7-X at $\beta = 4\%$
- Precise QP is difficult (impossible?) to achieve
- Higher alpha particle losses for OT case might be due to wide banana orbits?





GPU Allows Direct Optimization of Particles (Instead of proxies such as omnigenity)

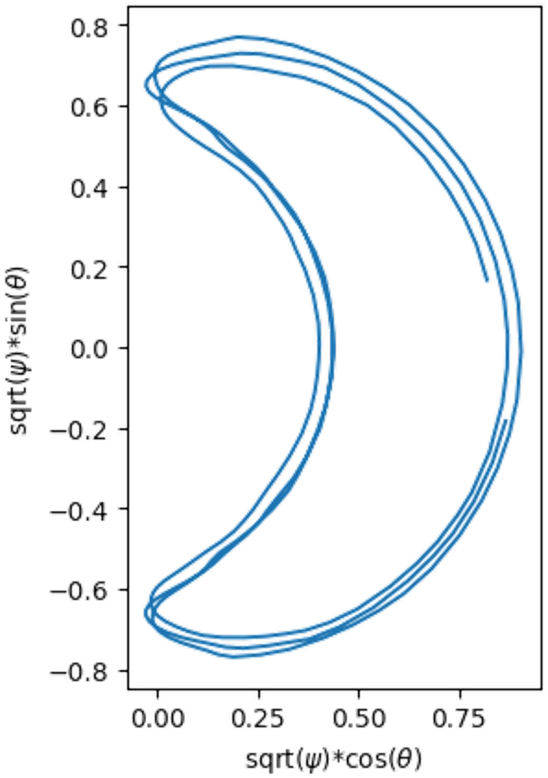
- **GPU has advantage in doing the same compute many times**
- ***Integrate lots of particle trajectories in DESC (guiding center equations of motion)***
- **JAX autodiff: Jacobian of trace particle trajectories wrt equilibrium**

$$\dot{\psi} = \frac{m}{qB^3}\left(\frac{v_\perp^2}{2} + v_\parallel^2\right) \boldsymbol{B} \times \nabla B \cdot \nabla \psi$$

$$\dot{\theta} = \frac{v_\parallel}{B}\boldsymbol{B} \cdot \nabla \theta \frac{m}{qB^3}\left(\frac{v_\perp^2}{2} + v_\parallel^2\right) \boldsymbol{B} \times \nabla B \cdot \nabla \theta$$

$$\dot{\zeta} = \frac{v_\parallel}{B}\boldsymbol{B} \cdot \nabla \zeta$$

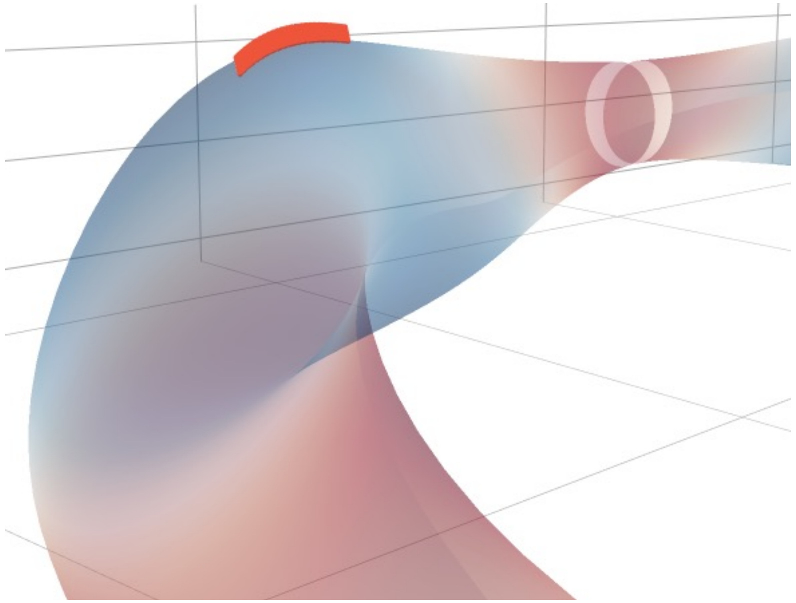
$$\dot{v_\parallel} = -\frac{v_\perp^2}{2B}\left(\frac{\boldsymbol{B}}{B} + \frac{m v_\perp^2}{2qB^3}\frac{1}{v_\parallel}\boldsymbol{B} \times \nabla B\right) \cdot \nabla B$$



Equilibrium optimization using a Particle Tracer

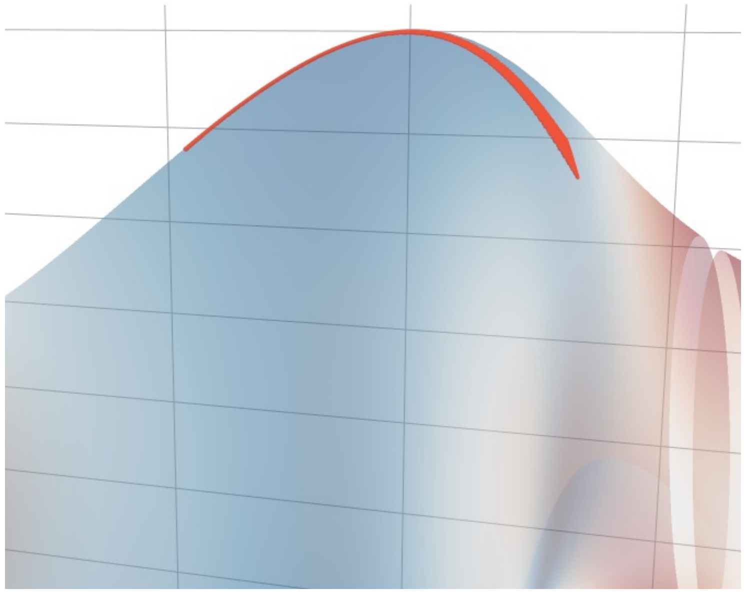
- Particle Tracer:
  - **Integrate Guiding Center EoM directly**
  - **Optimize the equilibrium from particle's trajectories using JAX autodiff**

Starting Equilibrium

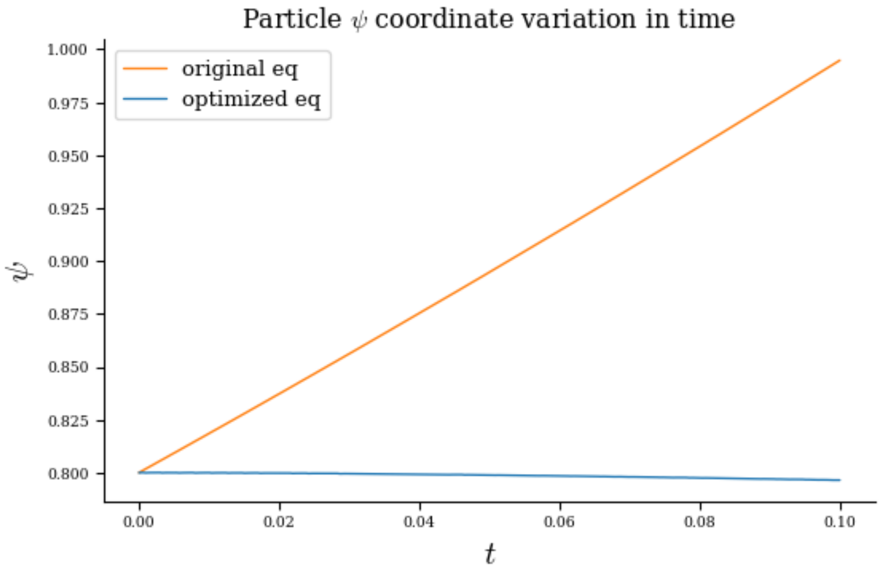


Particle drifting from flux surface ψ

Optimized Equilibrium



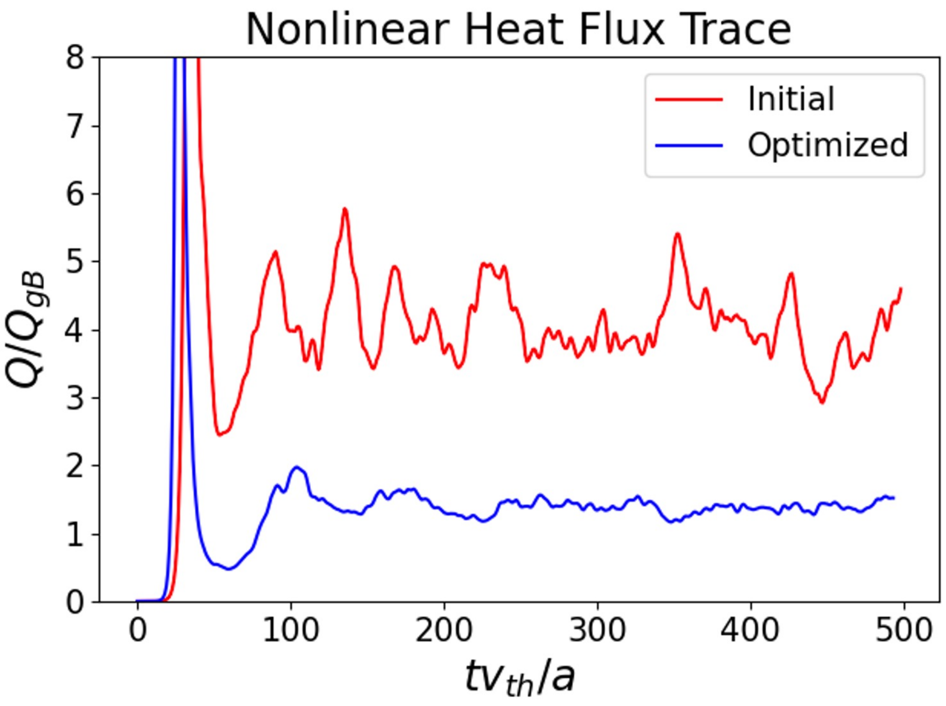
Particle confined in flux surface ψ

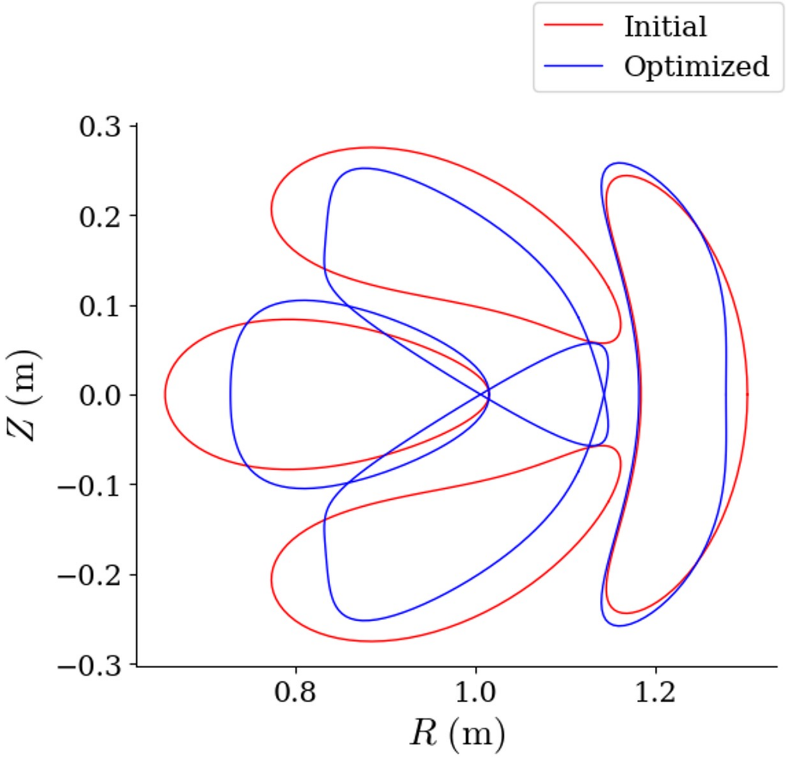


Plasma Control

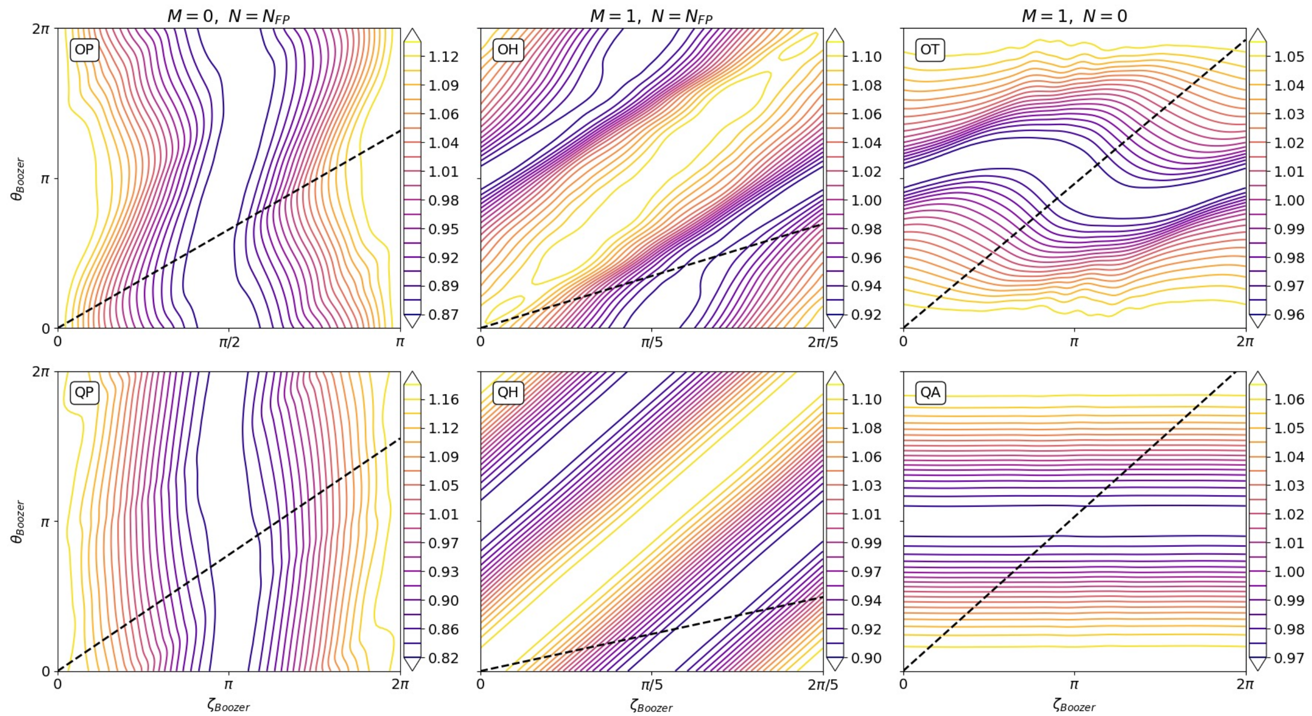
Turbulence + QS Optimization (Using GX+ DESC)

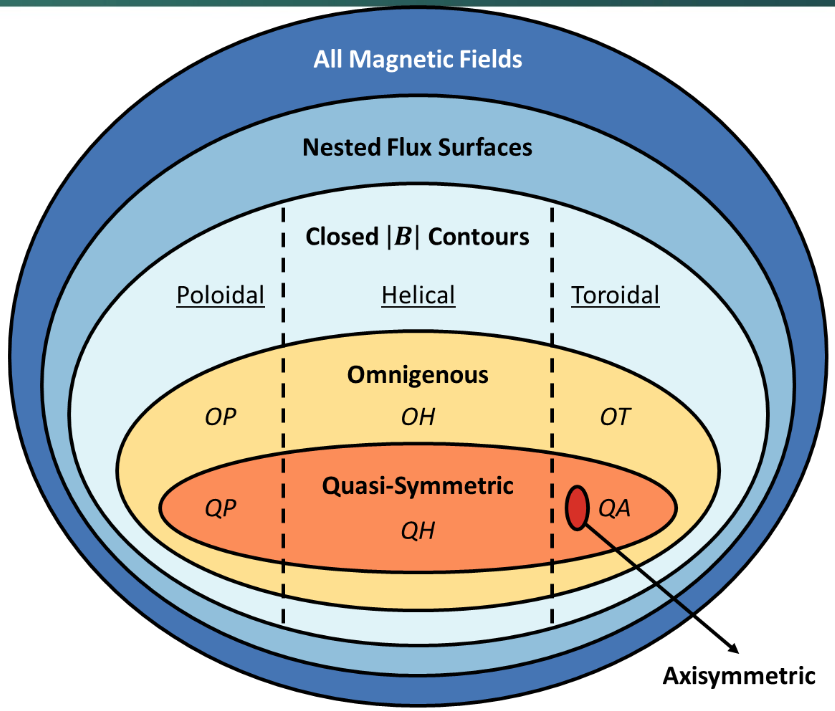
- GX + DESC coupling enables direct optimization of nonlinear heat fluxes with good quasi-symmetry.
- SPSA algorithm allows for cheaper gradient approximations for noisy objectives.
- Optimizer reduces nonlinear heat flux by a factor of 3, while maintaining good quasi-symmetry.

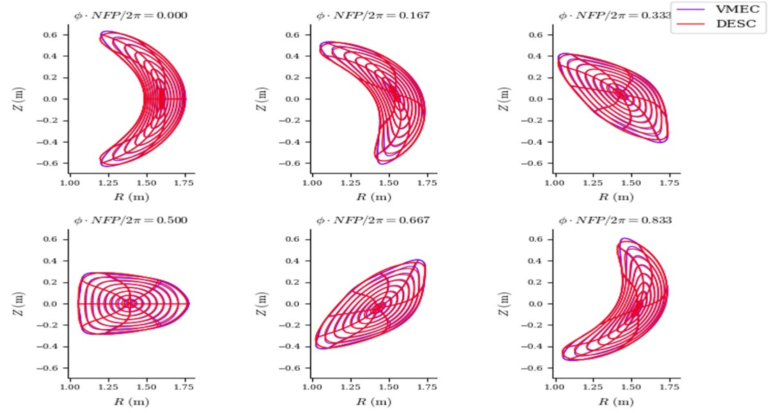


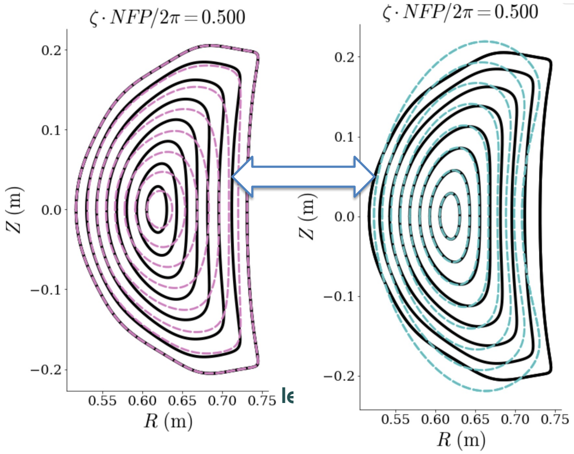


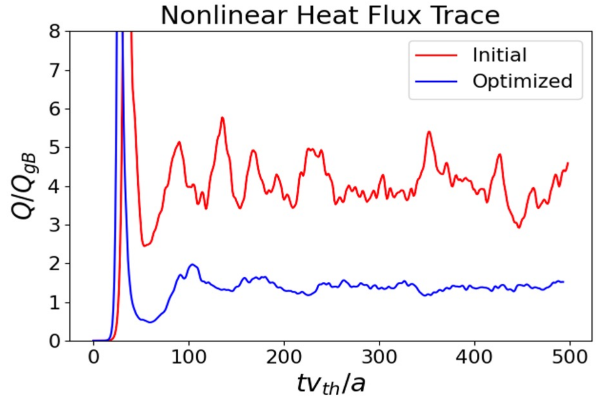
Conclusions: DESC multi-objective optimization applied to Turbulence Optimization, NEA, free surface stellarator & General Omnigenity











Postdoc Opportunity!

- **Novel Stellarator Designs**
- **Stellarator/Divertor Topology**
- **Plasma SOL**
- **Plasma, Applied Math, CS welcome**
- **Work on DESC**

- **Email: ekolemen@princeton.edu (or talk to me)**