EUROfusion

Energetic particle optimization of stellarator devices using near-axis magnetic fields

Rogerio Jorge

P. Rodrigues, J. Ferreira, A. Figueiredo, R. Coelho, D. Borba, P. Figueiredo

ipfn
INSTITUTO DE PLASMAS E FUSÃO NUCLEAR

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

Motivation

How to guarantee alpha particle heating in stellarator reactors?

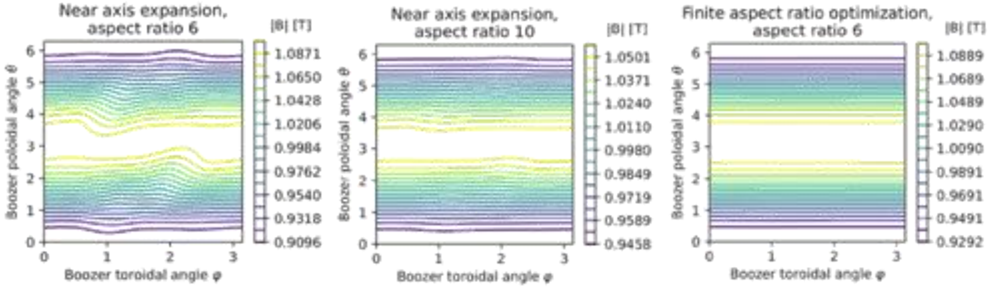
How to decrease first wall damage from unconfined particles?

- Particle confinement is usually considered the Achilles Heel of stellarators

- New proxies have changed this paradigm

- Analytical models (e.g., near-axis expansion) decrease degrees of freedom but have other tradeoffs

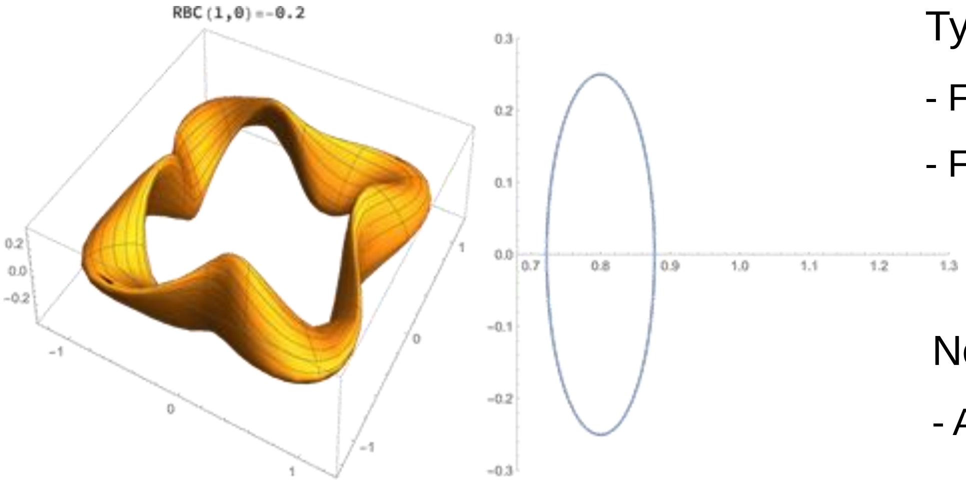
- The near-axis can also provide physical insight and guide future designs



(a) NAE for $A = 6$. (b) NAE for $A = 10$. (c) Optimized precise QA with $A = 6$.

Goals

Obtain reactor relevant stellarator shapes in a reliable and efficient manner



Typical degrees of freedom to solve $\boldsymbol{J} \times \mathbf{B} = \nabla \mathrm{P}$

- Fixed boundary (LCFS): ~100 Fourier coefficients
- Free boundary (coils): ~400 Fourier coefficients

Near-axis (high aspect ratio) degrees of freedom

- Axis + 1st order + 2nd order: ~10 Fourier coefficients

Is this gain worth it?

EnR Task Specification

The project is divided into 5 different tasks (WP)

- WP1 – Particle tracer code development (near-axis & full MHD)
- WP2 – Combine particle tracer and stellarator optimization codes

2022

- WP3 – Optimized stellarator equilibria (QS, QI and General)
- WP4 – Physics study of Nemov's criterion
- WP5 – Fast particle orbits in realistic magnetic fields

2023

With the following goals

- Create an open-source, user friendly, fully tested particle tracer (WP1, WP2)
- Perform the first direct fast particle optimization of a stellarator (WP3)
- Compare fast particle optimization with commonly used proxies (WP4)
- Extend the optimization to stochastic magnetic fields (WP5)

Publications

First Author	Title of work	Journal	DOI
Rogerio Jorge	Single-Stage Stellarator Optimization: Combining Coils with Fixed Boundary Equilibria	Plasma Physics and Controlled Fusion, 65 074003 (2023)	doi.org/10.1088/1361-6587/acd957
Paulo Figueiredo	Energetic Particle Tracing in Optimized Quasisymmetric Stellarator Equilibria	Journal of Plasma Physics, 90(2), 905900207 (2024)	doi.org/10.1017/S0022377824000400
Rogerio Jorge	Direct Microstability Optimization of Stellarator Devices	Physical Review E, 110(3), 035201 (2024)	doi.org/10.1103/PhysRevE.110.035201
Miguel Madeira	Tokamak to Stellarator Conversion using Permanent Magnets	Plasma Physics and Controlled Fusion, 66 085008 (2024)	doi.org/10.1088/1361-6587/ad5586
Pedro Curvo	Using Deep Learning to Design High Aspect Ratio Fusion Devices	Journal of Plasma Physics (submitted)	arXiv:2409.00564
Estêvão Gomes	Differentiable Single-Stage Optimization of Stellarator Coils Based on Alpha Particle Losses	Journal of Computational Physics (in preparation)	-

EnR Task Specification

The project is divided into 5 different tasks (WP)

- WP1 – Particle tracer code development (near-axis & full MHD)
- WP2 – Combine particle tracer and stellarator optimization codes
- WP3 – Optimized stellarator equilibria (QS, QI and General)
- WP4 – Physics study of near-axis expansion and Nemov's criterion
- WP5 – Fast particle orbits in realistic magnetic fields

2022 (WP1, WP2)

2023 (WP3, WP4, WP5)

With the following goals

- Create an open-source, user friendly, fully tested particle tracer (WP1, WP2)
- Perform the first direct fast particle optimization of a stellarator (WP3)
- Compare fast particle optimization with commonly used proxies (WP4)
- Extend the optimization to stochastic magnetic fields (WP5)

WP1

Development of the NEAT code

- ✓ Near-Axis Geometry
- ✓ Full MHD Geometry

Already being used by the stellarator optimization community

Critical gradient turbulence optimization toward a compact stellarator reactor concept

G. T. Roberg-Clark,* G. G. Plunk, P. Xanthopoulos, C. Nührenberg, S. A. Henneberg, and H. M. Smith
Max-Planck-Institut für Plasmaphysik, D-17491, Greifswald, Germany
(Dated: January 18, 2023)

Integrating turbulence into stellarator optimization is shown by targeting the onset for the ion-temperature-gradient mode, highlighting effects of parallel connection length, local magnetic shear, and flux surface expansion. The result is a compact quasihelically symmetric stellarator configuration, admitting a set of uncomplicated coils, with significantly reduced turbulent heat fluxes compared to a known stellarator. The new configuration combines low values of neoclassical transport, good alpha particle confinement, and Mercier stability at a plasma beta of almost 2%.

arXiv:2301.06773v1 [physics.plasm-ph] 17 Jan 2023

when rescaled to an ARIES-CS-equivalent [36] minor radius and volume averaged magnetic field strength, using the NEAT code [37, 38] (Fig. 2). Increased neoclassical

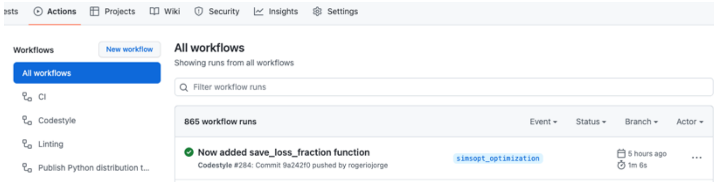
GitHub Open-source and automatic testing with GitHub actions

NEAT

rogeriojorge / NEAT Public

NEar-Axis opTimisation

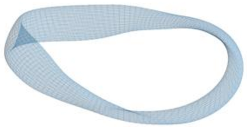
license GPL-3.0 | build passing | docs passing | codecov 61%

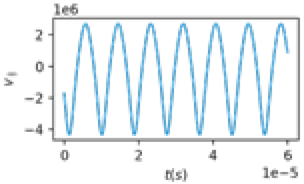


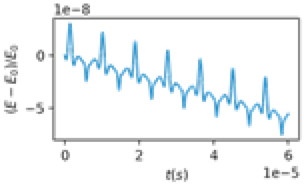
User-friendly example in near-axis geometry

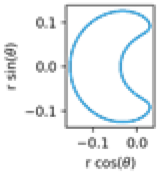
```python
from neat.fields import StellnaQS
from neat.tracing import ChargedParticle, ParticleOrbit
g_field = StellnaQS.from_paper(1)
g_particle = ChargedParticle()
g_orbit = ParticleOrbit(g_particle, g_field tfinal=1e-4)
g_orbit.plot_orbit()
```

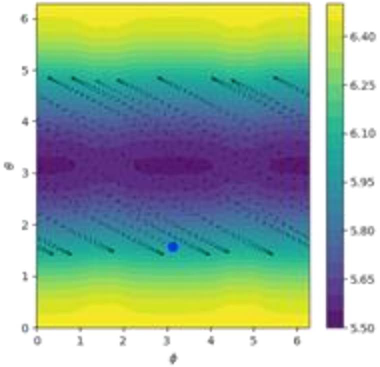
```python
print("Creating B contour plot")
g_orbit.plot_orbit_contourB()
```











WP1

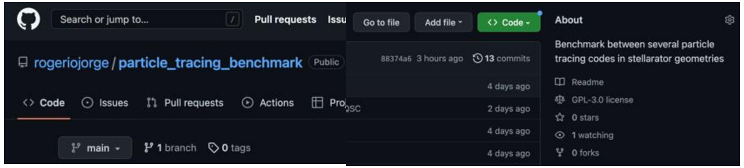
Flexible geometries

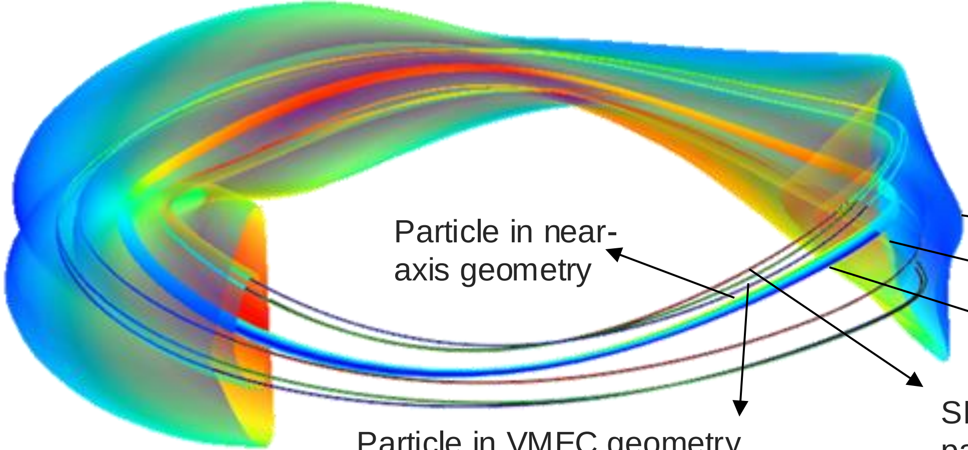
- VMEC output files
- Near-axis analytical model
- Near-axis quasisymmetry (exact/partial)
- Dommaschk potentials (magnetic islands)

Multiple tracers

- SIMPLE
- gyronimo
- SIMSOPT
- BEAMS3D

Benchmark available on
https://github.com/rogeriojorge/particle_tracing_benchmark





NEAT is fast as it uses C++ for trajectory calculations,

which are called via Python

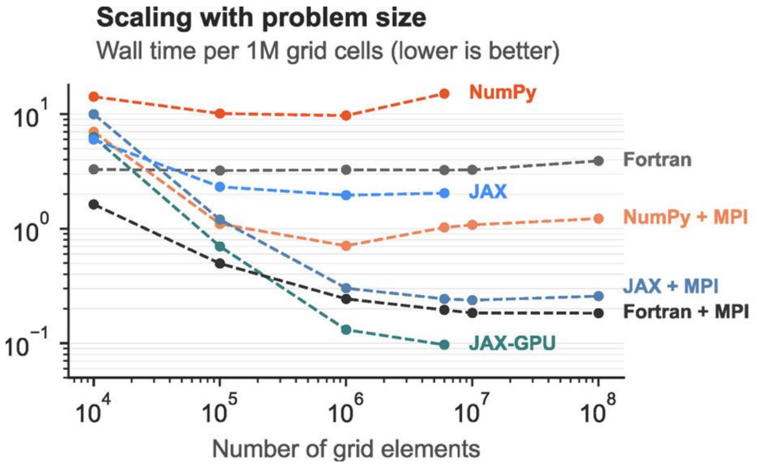
But can we simplify it further?

Use JAX!

JAX: High-Performance Array Computing

JAX is Autograd and XLA, brought together for high-performance numerical computing.

- JAX allows python scripts to run as fast as compiled code
- JAX provides derivatives of the output of the code with respect to the input (backpropagation)
- The same code can be run on CPUs and GPUs



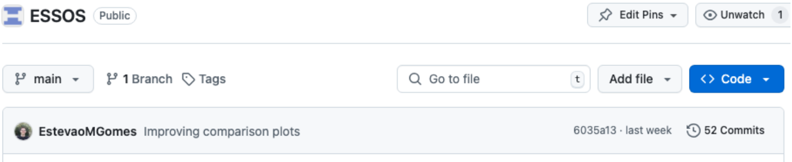
JAX performance compared to other compiled and parallelized backends [1]

[1] D. Hafner et al, "Fast, Cheap and Turbulent – Global Ocean Modeling with GPU Acceleration in Python, Journal of Advances in Modeling Earth Systems 13 (2021)

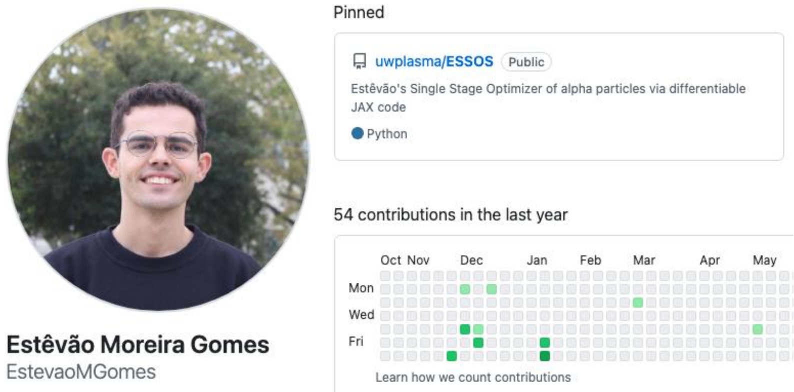
WP1

Particle tracer code ESSOS using:

- only Python
- hybrid OpenMP/MPI parallelization
- able to run on CPUs and GPUs



Developed by IST undergrad Student



Guiding Center Equations

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

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

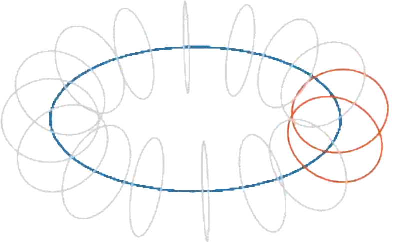
$$\dot{\zeta} = \frac{v_\parallel}{B} \boldsymbol{B} \cdot \nabla \zeta$$

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

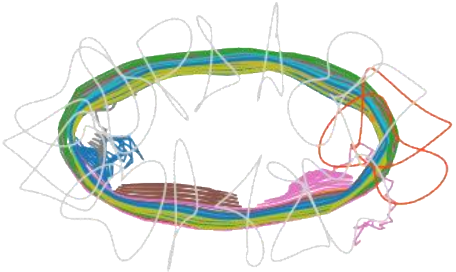
+

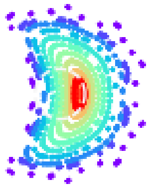
On Biot-Savart coil fields

$$\mathbf{B} = \frac{\mu_0}{4\pi}\int \mathbf{J}(\mathbf{r}') \times \frac{\mathbf{r} - \mathbf{r}'}{|\mathbf{r} - \mathbf{r}'|^3} dV'$$



Optimized stellarator (WP5)





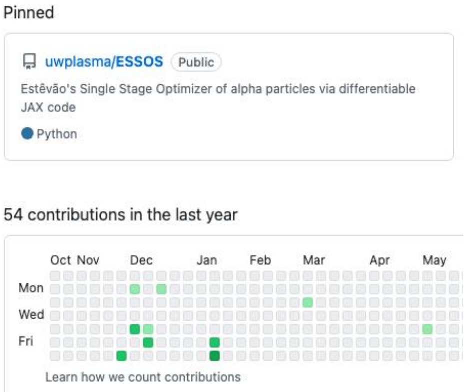
WP1

Particle tracer code ESSOS using:

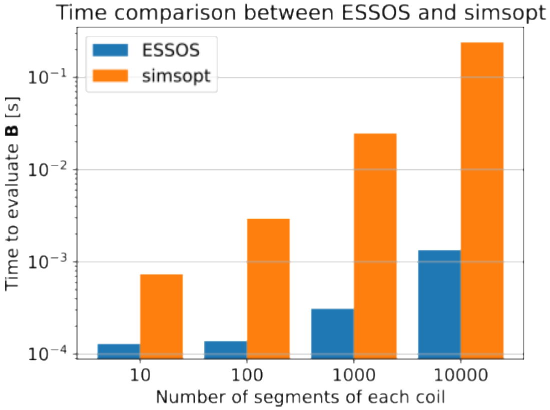
- only Python
- hybrid OpenMP/MPI parallelization
- able to run on CPUs and GPUs

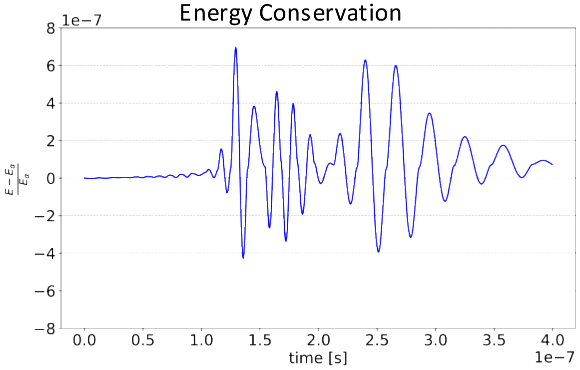
Developed by IST undergrad Student

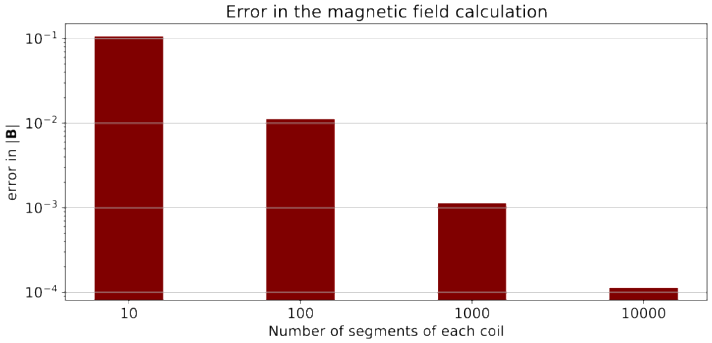
Estêvão Moreira Gomes
EstevaoMGomes



Benchmarks







EnR Task Specification

The project is divided into 5 different tasks (WP)

- WP1 – Particle tracer code development (near-axis & full MHD)
- WP2 – Combine particle tracer and stellarator optimization codes
- WP3 – Optimized stellarator equilibria (QS, QI and General)
- WP4 – Physics study of near-axis expansion and Nemov's criterion
- WP5 – Fast particle orbits in realistic magnetic fields

2022 (WP1–WP2)

2023 (WP3–WP5)

With the following goals

- Create an open-source, user friendly, fully tested particle tracer (WP1, WP2)
- Perform the first direct fast particle optimization of a stellarator (WP3)
- Compare fast particle optimization with commonly used proxies (WP4)
- Extend the optimization to stochastic magnetic fields (WP5)

WP2 – Combine particle tracer and stellarator optimization codes

2. Integration with stellarator optimization frameworks (WP2)

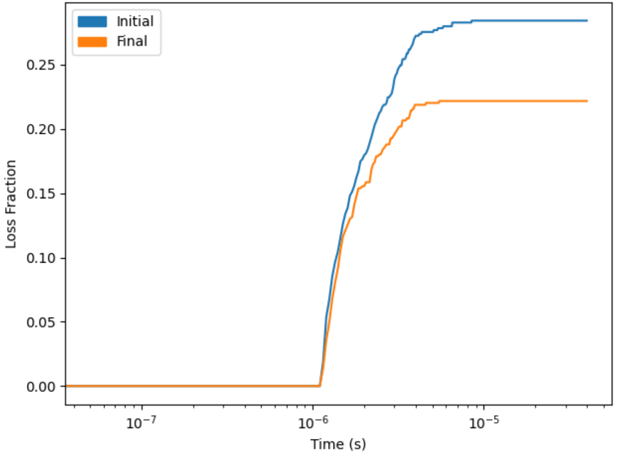
- scipy.optimize.minimize or SIMSOPT – near-axis
- SIMSOPT - full MHD
- DESC - full MHD (finalizing)

User-friendly example in near-axis geometry

```
from simsopt.objectives import LeastSquaresProblem
from simsopt.solve import least_squares_mpi_solve
from simsopt.util.mpi import MpiPartition
from neat.fields import StellnaQS
from neat.objectives import EffectiveVelocityResidual
from neat.tracing import ChargedParticleEnsemble
mpi = MpiPartition()
nsamples = 500
tfinal = 4e-5
g_particle = ChargedParticleEnsemble(r_initial=0.03,r_max=0.1,ntheta=5,nphi=5,nlambda_trapped=10,nlambda_passing=2)
g_field = StellnaQS.from_paper('r1 section 5.1', nphi=51, B0=5)
g_field.fix_all()
g_field.unfix("etabar")
g_field.unfix("rc(1)")
g_field.unfix("zs(1)")
residual = EffectiveVelocityResidual(g_field,g_particle,nsamples,tfinal,2,0.1)
prob = LeastSquaresProblem.from_tuples(
    [(residual.J, 0, 40),
     (g_field.get_elongation, 0.0, 0.5),
     (g_field.get_inv_L_grad_B, 0, 0.1)])
least_squares_mpi_solve(prob,mpi,grad=True,max_nfev=20,ftol=1e-5)
```

Choose particle ensemble
Choose geometry
Choose degrees of freedom
Choose objective function
Optimize

Reduction in loss of alpha particles



EnR Task Specification

The project is divided into 5 different tasks (WP)

- WP1 – Particle tracer code development (near-axis & full MHD)
- WP2 – Combine particle tracer and stellarator optimization codes

2022

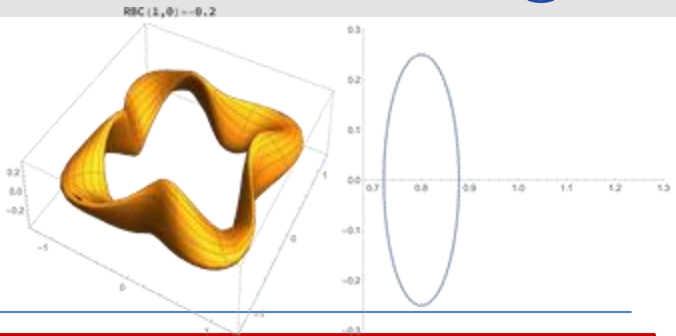
- WP3 – Optimized stellarator equilibria (QS, QI and General)
- WP4 – Physics study of near-axis expansion and Nemov's criterion
- WP5 – Fast particle orbits in realistic magnetic fields

2023

With the following goals

- Create an open-source, user friendly, fully tested particle tracer (WP1, WP2)
- Perform the first direct fast particle optimization of a stellarator (WP3)
- Compare fast particle optimization with commonly used proxies (WP4)
- Extend the optimization to stochastic magnetic fields (WP5)

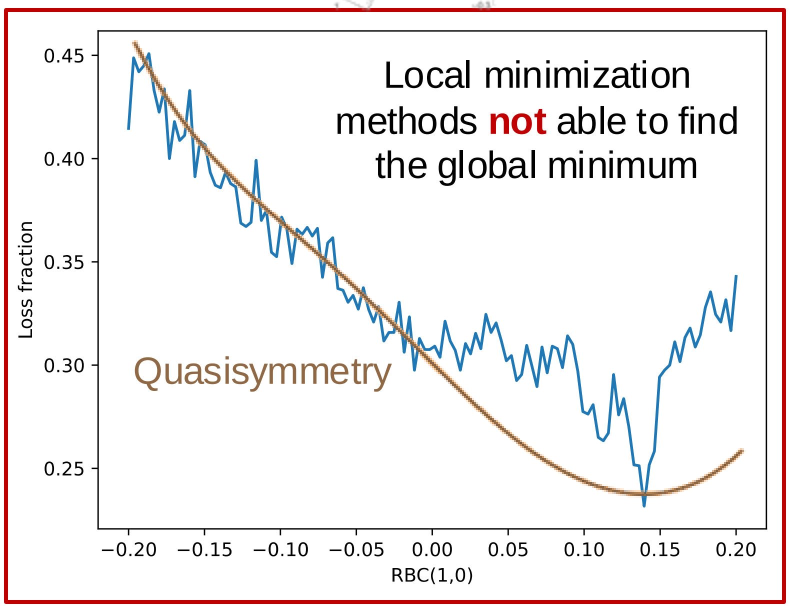
WP3 – Optimized stellarator equilibria



3. Optimized stellarator configurations (WP3)
   - Obtained Near-Axis Optimizations (previous slide)
   - Obtained full MHD Optimizations

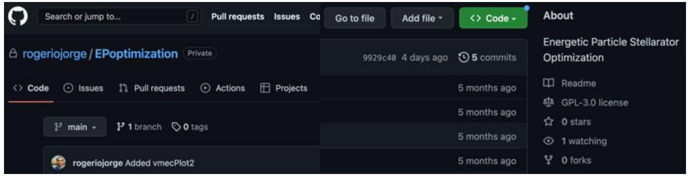
Minimal benchmark problem

- Trace 2400 particles for 5×10^{-4} s with the SIMPLE code
- Scale the minor radius and magnetic field to half of the ARIES-CS reactor
- Save the fraction of loss particles in an array for each RBC(1,0)
- Each point takes ~1 second on a laptop



WP3 – Optimized stellarator equilibria

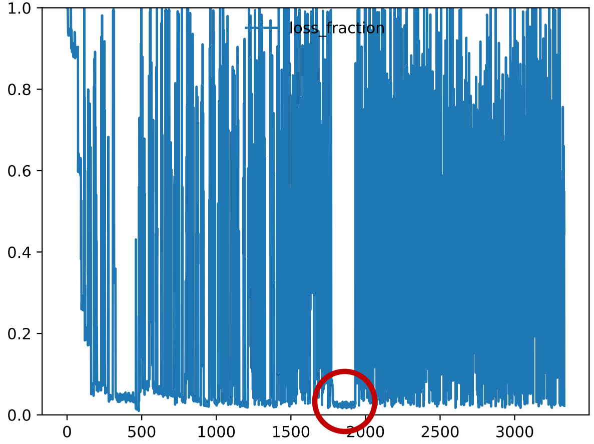
Scripts available on https://github.com/rogeriojorge/EPoptimization



3. Optimized stellarator configurations (WP3)

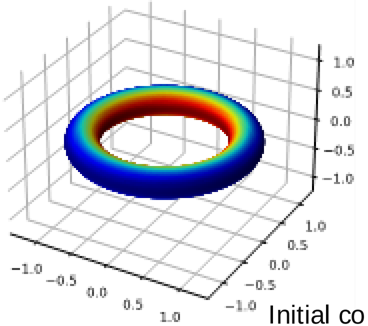
- Obtained Near-Axis Optimizations (previous slide)
- Obtained full MHD Optimizations

Stochastic optimization – generalized dual annealing



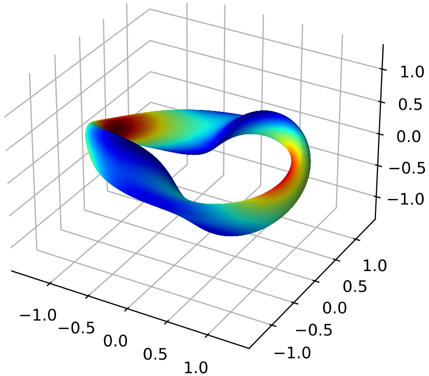
Number of iterations

Found a robust solution

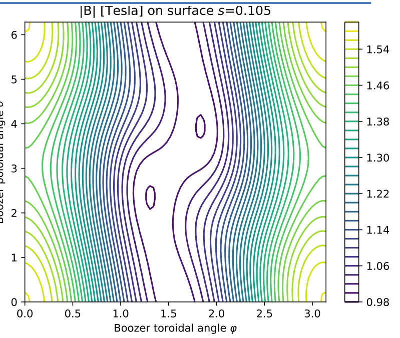


Initial condition (100% losses)

Optimization



Optimized solution (2.5% losses)



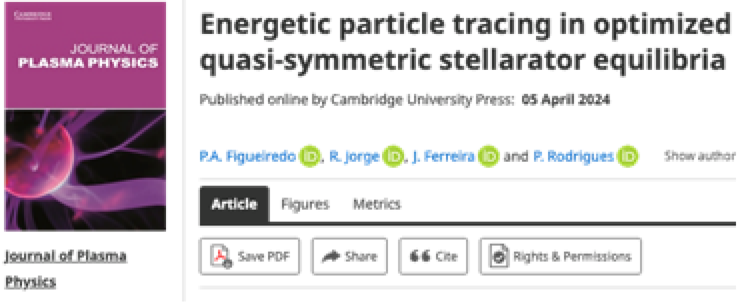
The optimizer found a quasi-isodynamic stellarator

WP3 – Optimized stellarator equilibria

Focus on the near-axis expansion

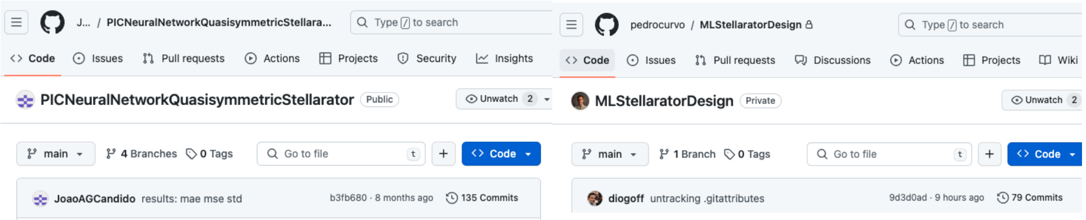
- Viability of near-axis solutions as good initial conditions for full MHD designs

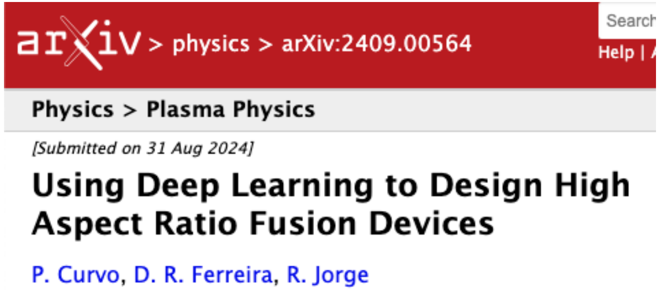
P. A. Figueiredo et al, JPP Volume 90, Issue 2, April 2024



- Near-axis database and machine learning model

J. Candido, Undergraduate Thesis (2022/2023)
P. Curvo, Undergraduate Thesis (2023/2024) – submitted to JPP





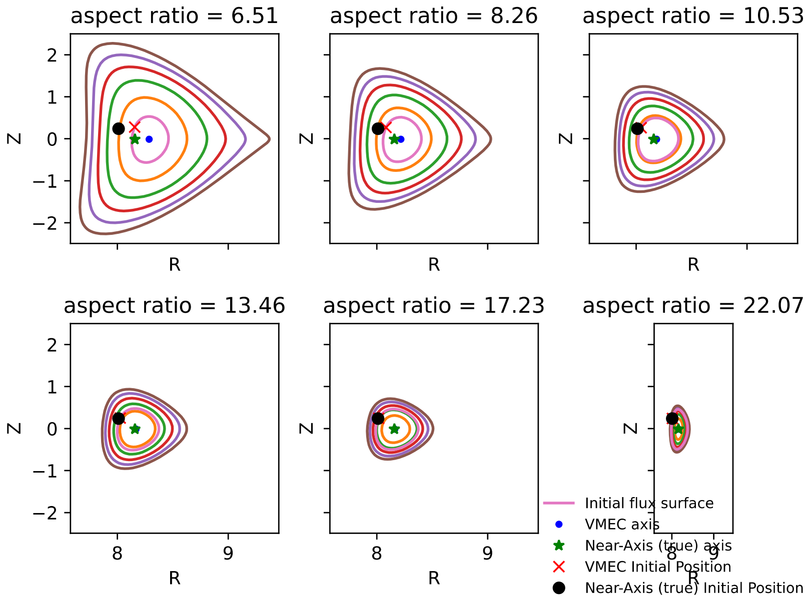
Focus on the near-axis expansion

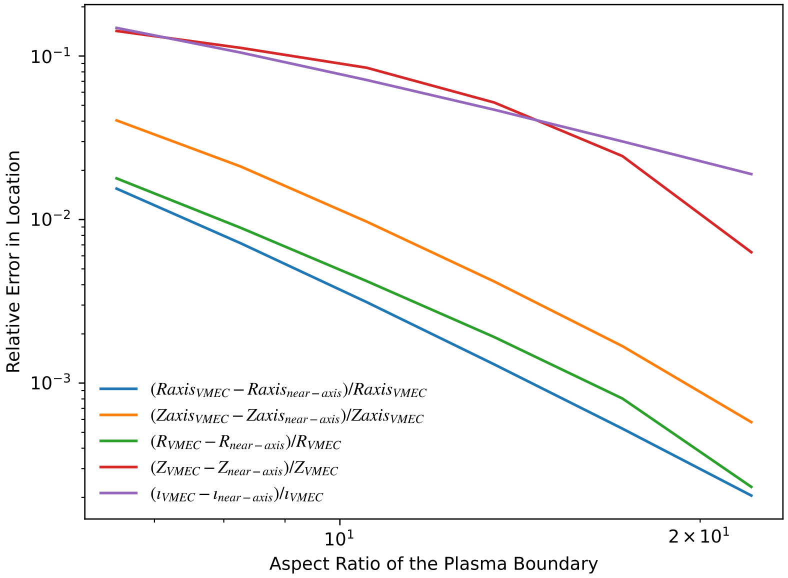
- Viability of near-axis solutions as good initial conditions for full MHD designs

P. A. Figueiredo et al, JPP Volume 90, Issue 2, April 2024

Create near-axis plasma boundaries and export them to VMEC.

Compare (R, Z) values of a near-axis vs. VMEC point





Focus on the near-axis expansion

- Viability of near-axis solutions as good initial conditions for full MHD designs

P. A. Figueiredo et al, JPP Volume 90, Issue 2, April 2024

1. Use NEAT to benchmark codes *(gyronimo vs SIMPLE)*

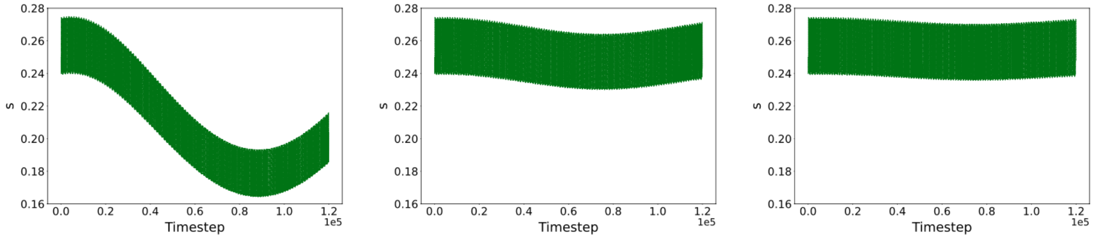
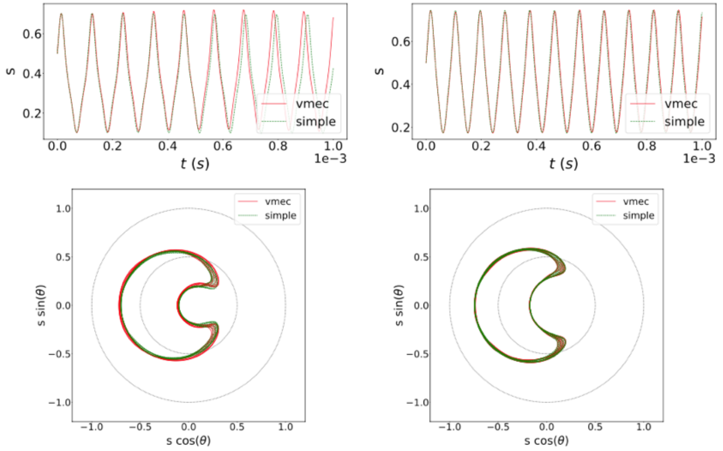


FIGURE 3. Orbits obtained with SIMPLE with $(s, \theta, \phi) = (0.25, 2.89, 1.84)$ and $v_{\parallel}/v = 0.44$ as

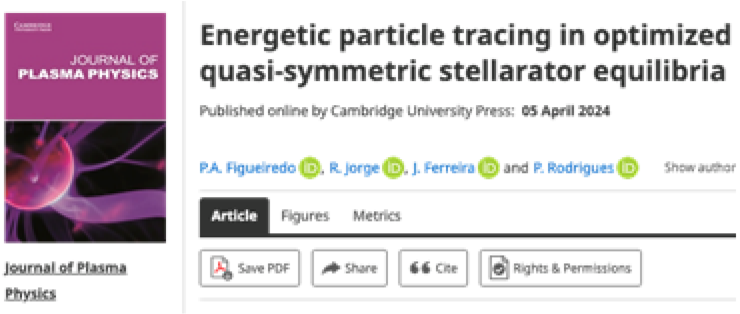


gyronimo and *SIMPLE* tracers for original *precise QA*, scaled to
$_A$ and B_0, the field at the plasma axis. With an initial position

Focus on the near-axis expansion

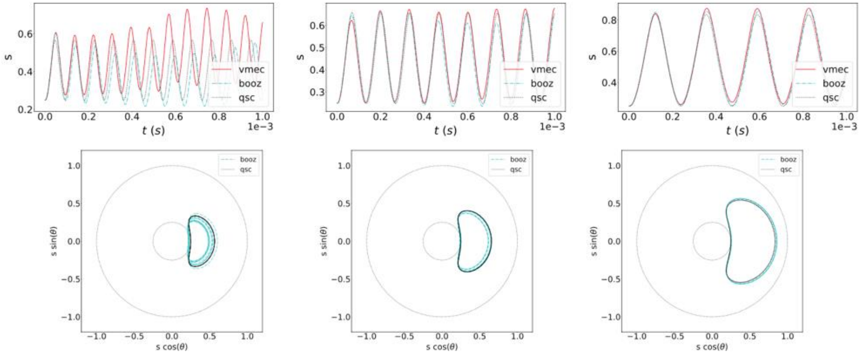
- Viability of near-axis solutions as good initial conditions for full MHD designs

P. A. Figueiredo et al, JPP Volume 90, Issue 2, April 2024

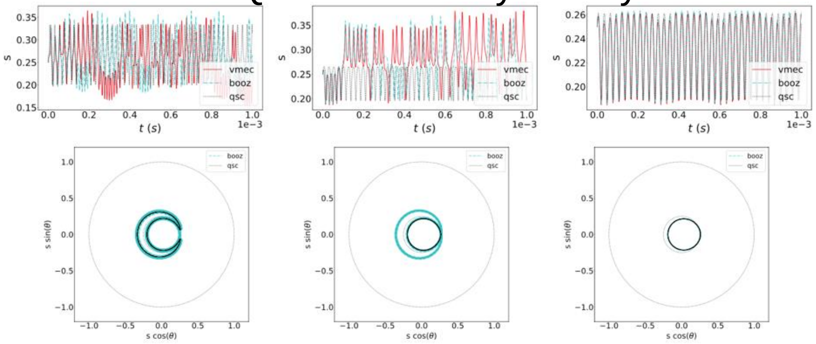


1. Use NEAT to benchmark codes
2. Compare near-axis vs. full MHD orbits

Quasi-Axisymmetry



Quasi-Helical Symmetry



WP3 – Optimized stellarator equilibria

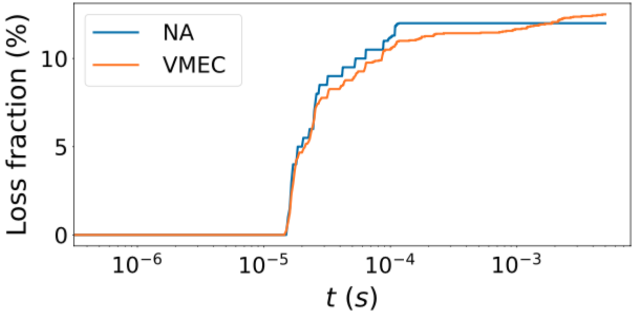
Focus on the near-axis expansion

- Viability of near-axis solutions as good initial conditions for full MHD designs

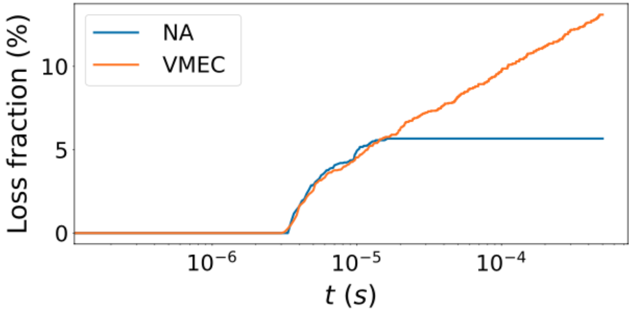
P. A. Figueiredo et al, JPP Volume 90, Issue 2, April 2024

1. Use NEAT to benchmark codes
2. Compare near-axis vs. full MHD orbits
3. Compare near-axis vs full MHD loss fractions

Quasi-Axisymmetry



Quasi-Helical Symmetry



WP3 – Optimized stellarator equilibria

Focus on the near-axis expansion

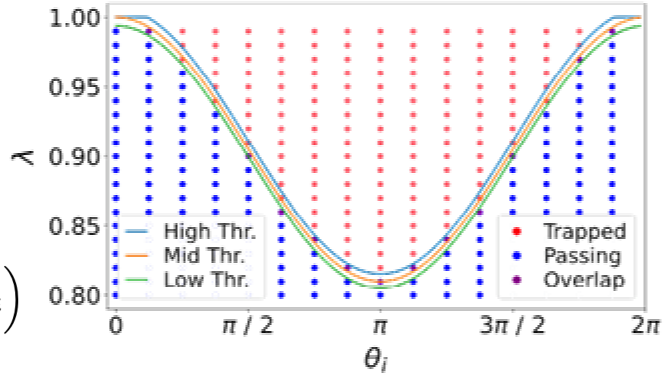
- Viability of near-axis solutions as good initial conditions for full MHD designs

P. A. Figueiredo et al, JPP Volume 90, Issue 2, April 2024

1. Use NEAT to benchmark codes
2. Compare near-axis vs. full MHD orbits
3. Compare near-axis vs full MHD loss fractions
4. Obtain analytical formulas for the trapped-passing boundary and banana width using near-axis expansion

$$\frac{(1+a_A\sqrt{s_i}\;\bar{\eta}\cos\theta_i)}{(1+a_A\sqrt{s_i+2\Delta s}\;|\bar{\eta}|)} \leqslant \lambda_s \leqslant \frac{(1+a_A\sqrt{s_i}\;\bar{\eta}\cos\theta_i)}{(1+a_A\sqrt{s_i-2\Delta s}\;|\bar{\eta}|)}$$

$$\Delta s = \frac{mvL\bar{\eta}}{\pi q\,\iota_{N_0} a_A B_0}\,\frac{1-\lambda B_0/(2B_i)}{\sqrt{1-\lambda B_0/B_i}}\left(2\sqrt{s_i}+\left(\sqrt{s_i+\Delta s_{avg}}-\sqrt{s_i}\right)\cos\theta_i\right)$$

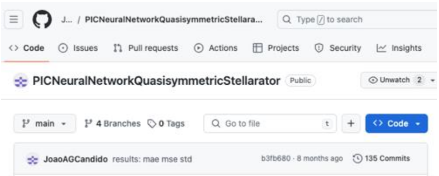


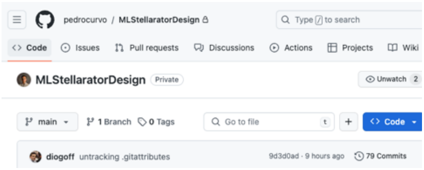
WP3 – Optimized stellarator equilibria

Focus on the near-axis expansion

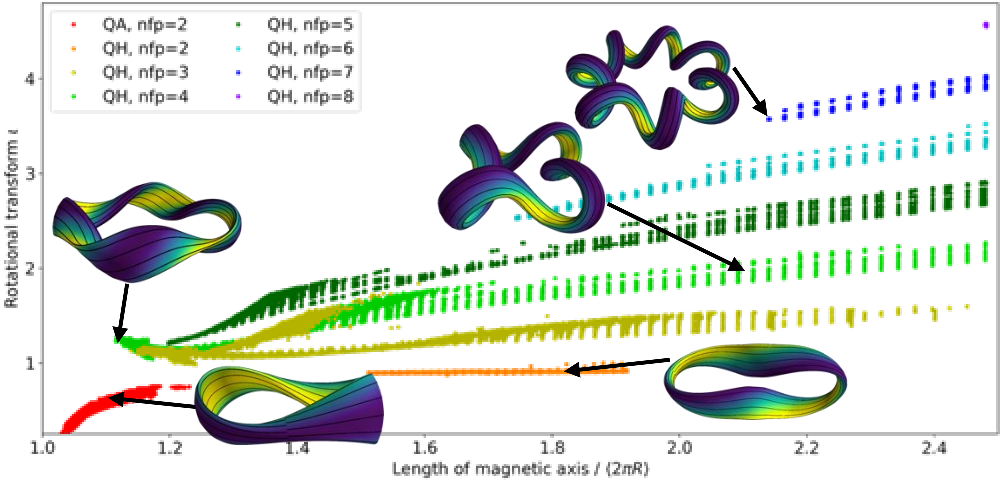
- Near-axis database and machine learning model

J. Candido, Undergraduate Thesis (2022/2023)
P. Curvo, Undergraduate Thesis (2023/2024)





1. Create a near-axis database similar to *M. Landreman, JPP 88(6), 2022*

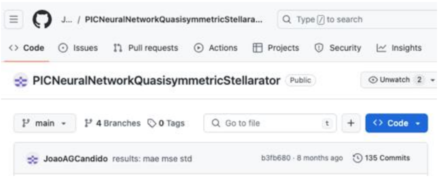


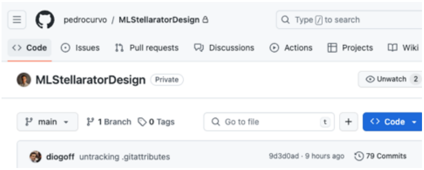
WP3 – Optimized stellarator equilibria

Focus on the near-axis expansion

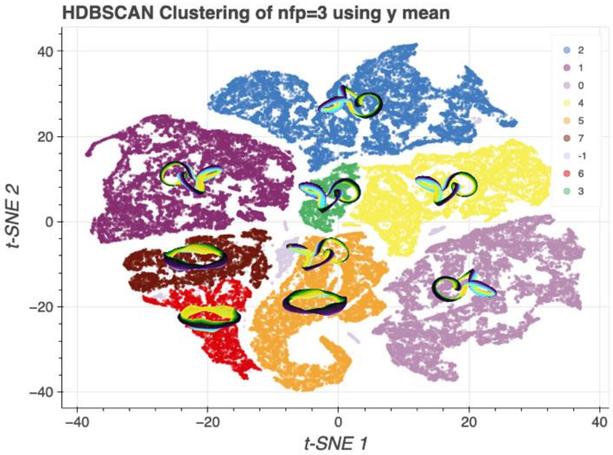
- Near-axis database and machine learning model

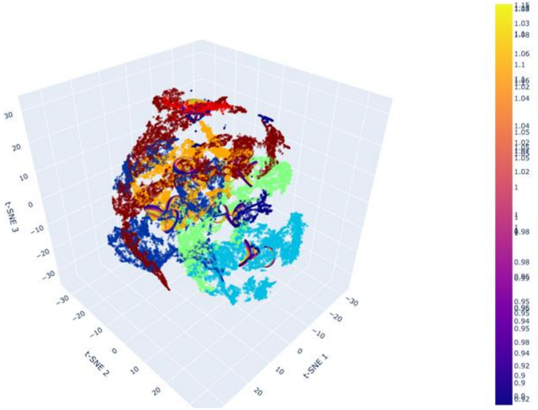
J. Candido, Undergraduate Thesis (2022/2023)
P. Curvo, Undergraduate Thesis (2023/2024)





1. Create a near-axis database similar to *M. Landreman, JPP 88(6), 2022*
2. Experiment with data-reduction and clustering methods (e.g., find division between QA and QH)



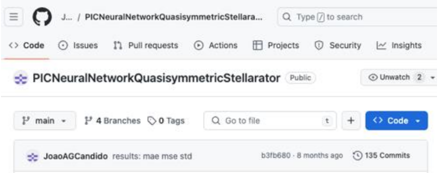


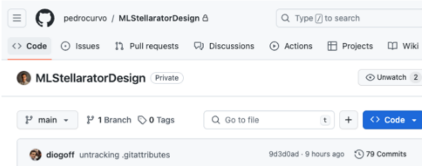
WP3 – Optimized stellarator equilibria

Focus on the near-axis expansion

- Near-axis database and machine learning model

J. Candido, Undergraduate Thesis (2022/2023)
P. Curvo, Undergraduate Thesis (2023/2024)





1. Create a near-axis database similar to *M. Landreman, JPP 88(6), 2022*
2. Experiment with data-reduction and clustering methods (e.g., find division between QA and QH)
3. Train neural network to reproduce forward and inverse solutions

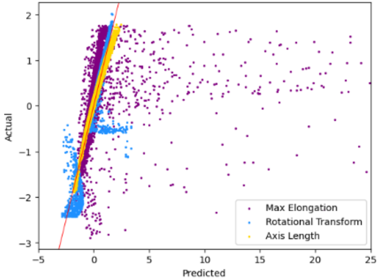
Forward model (easy to reproduce)

Model	Adjusted R-Squared	R-Squared	RMSE	Time Taken
ExtraTreesRegressor	0.66	0.66	0.60	3.84
RandomForestRegressor	0.64	0.64	0.62	11.98
XGBRegressor	0.64	0.64	0.62	4.36
BaggingRegressor	0.61	0.61	0.64	1.21
MLPRegressor	0.59	0.59	0.65	3.54
KNeighborsRegressor	0.51	0.51	0.72	0.18
DecisionTreeRegressor	0.29	0.29	0.87	0.20
ExtraTreeRegressor	0.24	0.24	0.90	0.04
RidgeCV	0.04	0.04	0.99	0.01
Ridge	0.04	0.04	0.99	0.01
Lars	0.04	0.04	0.99	0.01
TransformedTargetRegressor	0.04	0.04	0.99	0.01
LinearRegression	0.04	0.04	0.99	0.01
KernelRidge	0.04	0.04	0.99	12.93
OrthogonalMatchingPursuit	0.04	0.04	0.99	0.01
ElasticNet	-0.00	-0.00	1.01	0.01
Lasso	-0.00	-0.00	1.01	0.01
DummyRegressor	-0.00	-0.00	1.01	0.01
LassoLars	-0.00	-0.00	1.01	0.01

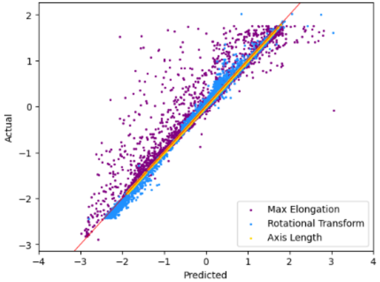
Inverse model (hard to reproduce)

alpha = 7.91e- 05, batch_size = 87, hidden_layer_sizes = [45, 45, 45, 45], learning_rate_init = 9.3 × 10^{-4}

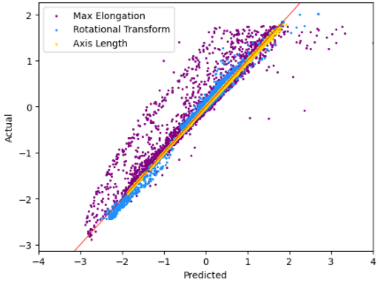
"Given an elongation, ι and elongation, what are its near-axis parameters?"



Linear Regressor



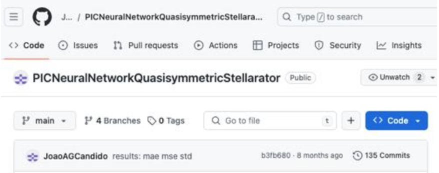
Polynomial Regressor

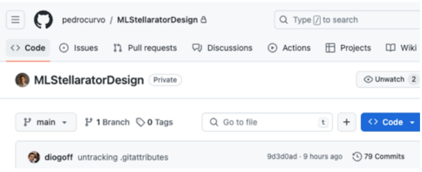


Neural Network

WP3 – Optimized stellarator equilibria

Focus on the near-axis expansion



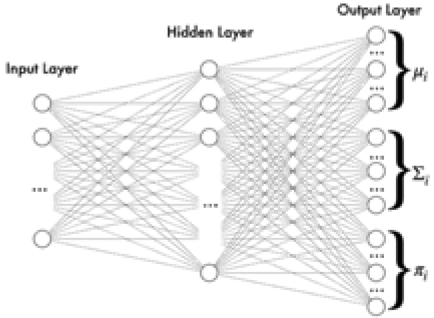


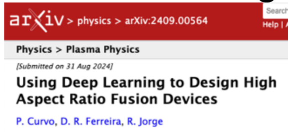
- Near-axis database and machine learning model

J. Candido, Undergraduate Thesis (2022/2023)
P. Curvo, Undergraduate Thesis (2023/2024)

1. Create a near-axis database similar to *M. Landreman, JPP 88(6), 2022*
2. Experiment with data-reduction and clustering methods (e.g., find division between QA and QH)
3. Train neural network to reproduce forward and inverse solutions
4. Train mixture density networks to solve the inverse design problem

Output Property	Range
axis length	> 0.0
$\lvert\iota\rvert$	$\geqslant 0.2$
max elongation	$\leqslant 10.0$
min $L_{\nabla B}$	$\geqslant 0.1$
min R_0	$\geqslant 0.3$
$r_{\text{singularity}}$	$\geqslant 0.05$
$L_{\nabla\nabla B}$	$\geqslant 0.1$
$B_{20\text{variation}}$	$\leqslant 5.0$
β	$\geqslant 10^{-4}$
$D_{\text{Merc}} \times r^2$	> 0.0



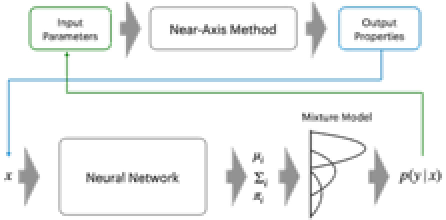

arXiv > physics > arXiv:2409.00564

Physics > Plasma Physics

[Submitted on 31 Aug 2024]

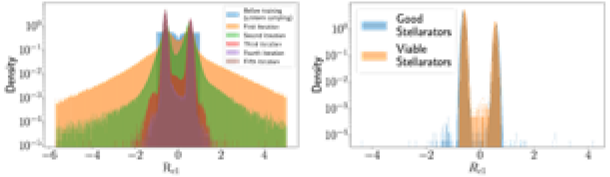
Using Deep Learning to Design High Aspect Ratio Fusion Devices

P. Curvo, D. R. Ferreira, R. Jorge



Dataset	Good Stellarators (%)
Before training (uniform sampling)	0.0018
After the first training iteration	0.0406
After the second training iteration	1.3788
After the third training iteration	9.0024
After the fourth training iteration	12.3903
After the fifth training iteration	20.2670

TABLE 6. Percentage of good stellarators in each iteration dataset.



EnR Task Specification

The project is divided into 5 different tasks (WP)

- WP1 – Particle tracer code development (near-axis & full MHD)
- WP2 – Combine particle tracer and stellarator optimization codes

2022

- WP3 – Optimized stellarator equilibria (QS, QI and General)
- WP4 – Physics study of near-axis expansion and Nemov's criterion
- WP5 – Fast particle orbits in realistic magnetic fields

2023

With the following goals

- Create an open-source, user friendly, fully tested particle tracer (WP1, WP2)
- Perform the first direct fast particle optimization of a stellarator (WP3)
- Compare fast particle optimization with commonly used proxies (WP4)
- Extend the optimization to stochastic magnetic fields (WP5)

Physics study of Nemov's criterion

Nemov Γ_c - minimize radial drift of trapped orbits

$$\Gamma_c = \frac{\pi}{\sqrt{8}} \lim_{L\to\infty}\left(\int_0^L \frac{dl}{B}\right)^{-1} \int_1^{B_{max}/B_{min}} db' \times \sum_{well_j} \gamma_c^2 \frac{v\tau_{b,j}}{4B_{min}\, b'^2}; \quad \gamma_c = \frac{2}{\pi}\arctan\frac{v_r}{v_\theta}$$

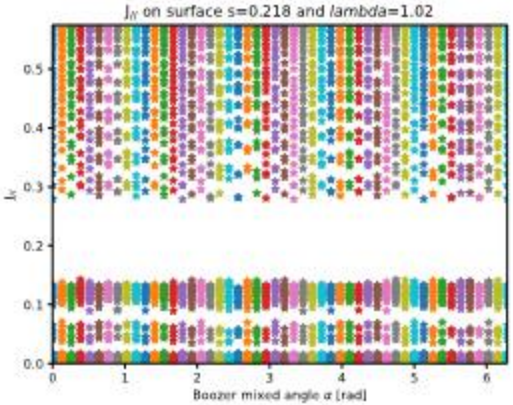
V_r - bounce average radial drift $\partial J/\partial\alpha$

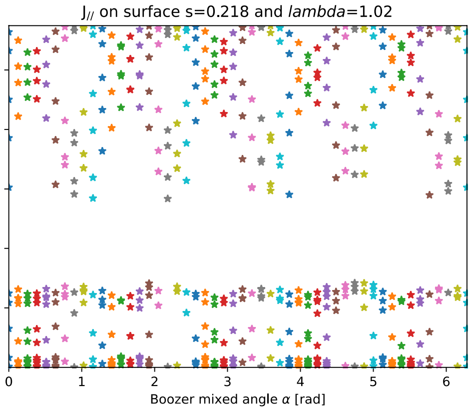
V_θ - bounce average poloidal drift $\partial J/\partial\alpha$

$J = \int_{bounce}\sqrt{1-\frac{|B|}{b'}}\,dl$ - adiabatic invariant

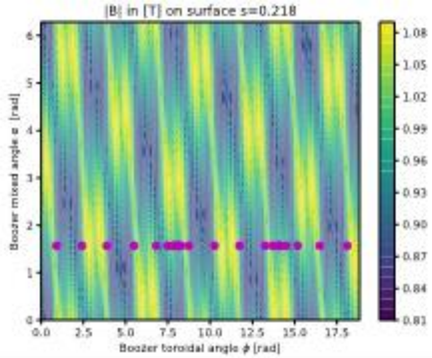
Calculate J at each surface and normalized magnetic moment $b'=1/\lambda$ by bounce averaging

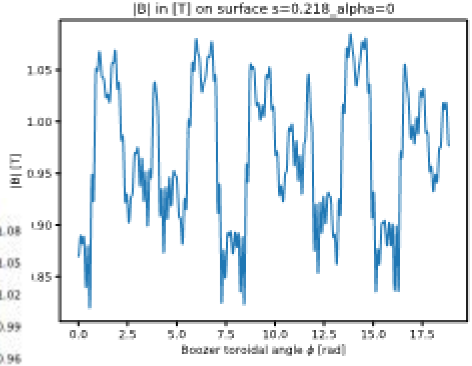
Higher resolution and longer field lines create discontinuities between wells leading to noise





Many turning points on unoptimized stellarators make calculation very complex





Calculation of Γ_c implemented but noise hinders optimization efforts *(R. Coelho, J. Rodrigues)*

EnR Task Specification

The project is divided into 5 different tasks (WP)

- WP1 – Particle tracer code development (near-axis & full MHD)
- WP2 – Combine particle tracer and stellarator optimization codes

2022

- WP3 – Optimized stellarator equilibria (QS, QI and General)
- WP4 – Physics study of near-axis expansion and Nemov's criterion
- WP5 – Fast particle orbits in realistic magnetic fields

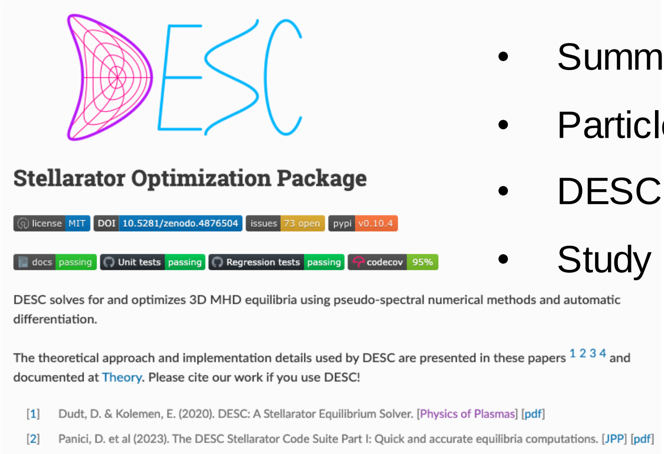
2023

With the following goals

- Create an open-source, user friendly, fully tested particle tracer (WP1, WP2)
- Perform the first direct fast particle optimization of a stellarator (WP3)
- Compare fast particle optimization with commonly used proxies (WP4)
- Extend the optimization to stochastic magnetic fields (WP5)

WP5 – Fast particle orbits in realistic magnetic fields

Implementation of particle tracing on DESC (collaboration with PPPL)


Stellarator Optimization Package

license MIT | DOI 10.5281/zenodo.4876504 | issues 73 open | pypi v0.10.4

docs passing | Unit tests passing | Regression tests passing | codecov 95%

DESC solves for and optimizes 3D MHD equilibria using pseudo-spectral numerical methods and automatic differentiation.

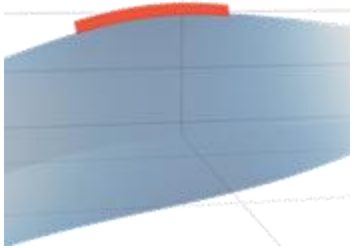
The theoretical approach and implementation details used by DESC are presented in these papers [1] [2] [3] [4] and documented at Theory. Please cite our work if you use DESC!

[1] Dudt, D. & Kolemen, E. (2020). DESC: A Stellarator Equilibrium Solver. [Physics of Plasmas] [pdf]

[2] Panici, D. et al (2023). The DESC Stellarator Code Suite Part I: Quick and accurate equilibria computations. [JPP] [pdf]

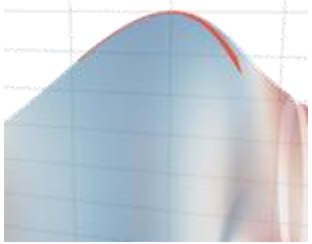
- Summer 2023 visit of João Biu to Princeton
- Particle tracing now implemented in DESC
- DESC uses automatic differentiation
- Study of direct particle tracing using automatic differentiation underway

Starting Equilibrium

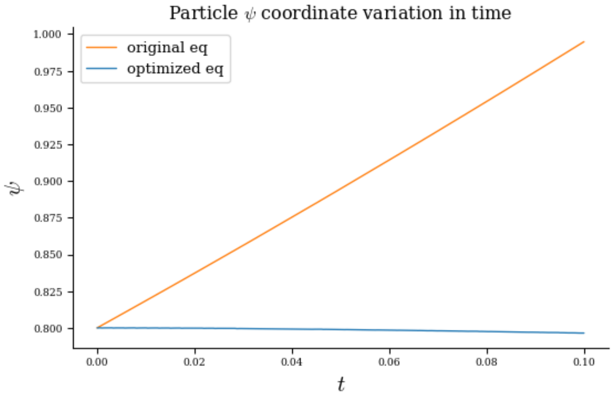


Particle drifting from flux surface ψ

Optimized Equilibrium



Particle confined in flux surface ψ



WP5 – Fast particle orbits in realistic magnetic fields

Dommaschk potentials (analytical B field in NEAT)

M. Pereira, Undergraduate Thesis (2022/2023)

$$U = I_{m,n}(Z,R)e^{\pm i m\phi},$$

$$I_{m,n} = \sum_{2k\le n}^{k=0} \frac{Z^{n-2k}}{(n-2k)!} C_{m,k}(R),\ k = 0,1,\dots n/2,$$

$$C^D_{m,k} = -[\alpha_j(\alpha^* ln(R) + \gamma^* - \alpha)_{k-m-j} - \gamma_j\alpha^*_{k-m-j} + \alpha_j\beta^*_{k-j}]R^{2j+m} + \dots$$

$$C^N_{m,k} = +[\alpha_j(\alpha ln(R) + \gamma)_{k-m-j} - \gamma_j\alpha_{k-m-j} + \alpha_j\beta_{k-j}]R^{2j+m} - \beta_j\alpha_{k-j}\dots$$

$$\frac{\partial}{\partial R}C^D_{m,k} = -[\alpha_j(\alpha^* ln(R) + \gamma^* - \alpha + \frac{\alpha^*}{2j+m})_{k-m-j} - \gamma_j\alpha^*_{k-m-j} + \alpha_j\beta^*_{\dots}\dots + \beta_j\alpha^*_{k-j}R^{2j-m-1}(2j-m),$$

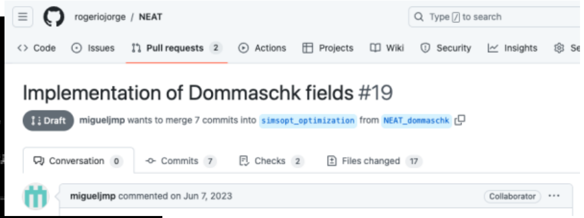
$$\frac{\partial^2}{\partial R^2}C^D_{m,k} = -[\alpha_j(\alpha^* ln(R) + \gamma^* - \alpha + \frac{\alpha^*(4j+2m+1)}{(2j+m)(2j+m-1)})_{k-m-j} - \gamma_j\alpha^*_{k-m-j} + \alpha_j\beta^*_{k-j}]R^{2j+m-2}(2j+m)(2j+m-1) + \beta_j\alpha^*_{k-j}R^{2j-m-2}(2j-m)(2j-m-1),$$

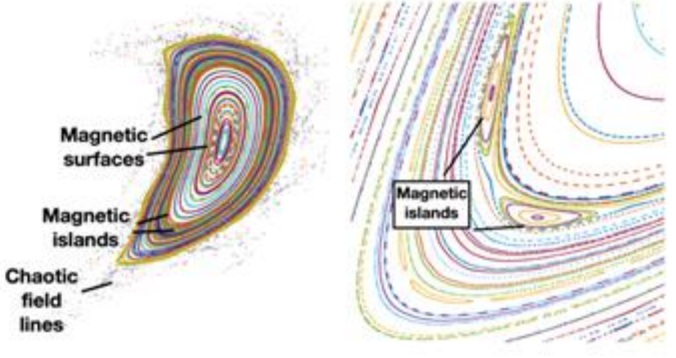
$$\frac{\partial}{\partial R}C^N_{m,k} = -[\alpha_j(\alpha ln(R) + \gamma + \frac{\alpha}{2j+m})_{k-m-j} - \gamma_j\alpha^{k-m-j} + \alpha_j\beta^{k-j}]R^{2j+m-1}(2j+m) - \beta_j\alpha^{k-j}R^{2j-m-1}(2j-m),$$

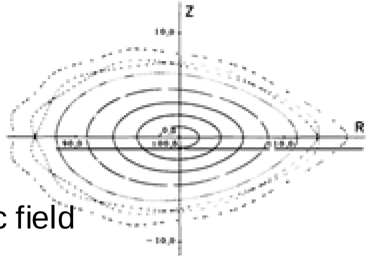
$$\frac{\partial^2}{\partial R^2}C^N_{m,k} = -[\alpha_j(\alpha ln(R) + \gamma + \frac{\alpha(4j+2m+1)}{(2j+m)(2j+m-1)})_{k-m-j} - \gamma_j\alpha_{k-m-j} + \alpha_j\beta_{k-j}]R^{2j+m-2}(2j+m)(2j+m-1) - \beta_j\alpha_{k-j}R^{2j-m-2}(2j-m)(2j-m-1),$$

$$\frac{\partial}{\partial Z}I_{m,n} = \sum_{2k\le n}^{k=0}\frac{Z^{n-2k-1}(n-2k)}{(n-2k)!}C_{m,k}(R),$$

$$\frac{\partial^2}{\partial Z^2}I_{m,n} = \sum_{2k\le n}^{k=0}\frac{Z^{n-2k-2}(n-2k)(n-2k-1)}{(n-2k)!}C_{m,k}(R),$$



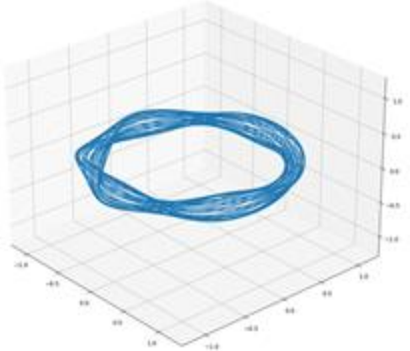


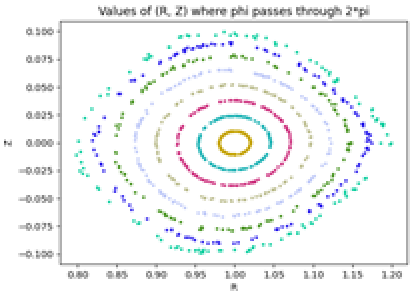


Poincaré plot of

Dommaschk magnetic field

Passing particles traced in such a field





Analysis of trapped trajectories in magnetic islands left for future work

End