

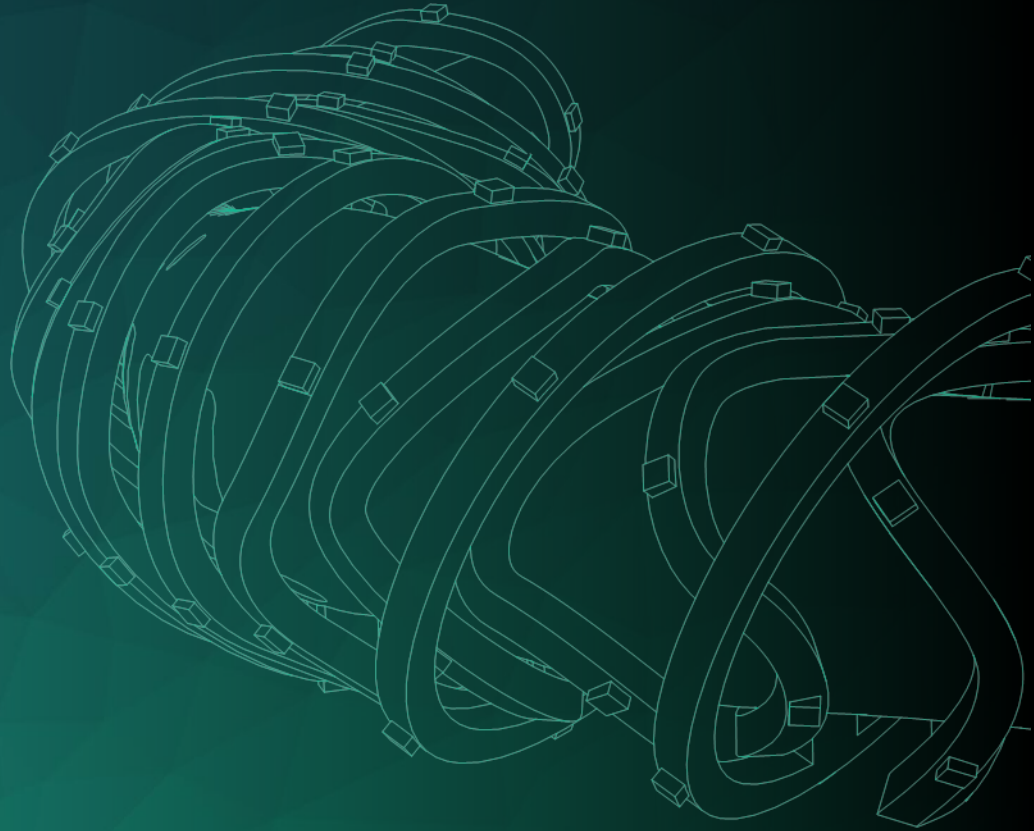


Proxima  
Fusion

# Software Supercharges Stellarator Design

Open Source @ Siemens 2026

Jonathan Schilling





# Fusion is the ultimate source of clean energy

Fusion is the process that powers the stars.

By forcing light nuclei (normally, heavy hydrogen) together, mass is transformed into energy.

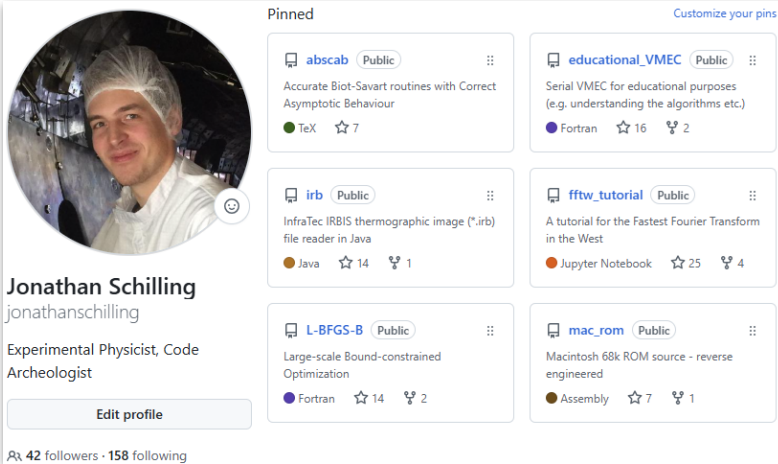
The Sun is a remarkably inefficient fusion reactor: it only works thanks to huge gravitational forces.

By harnessing fusion on Earth, we will directly access clean, safe, and effectively limitless energy.

**When we enter the fusion era, we will end energy scarcity on Earth.**

# About me...

- Co-Founder & Head of Labs @ Proxima Fusion
- PhD @ Max-Planck-Institute for Plasma Physics, Greifswald
  - modeling for Wendelstein 7-X stellarator experiment
- OSS interests: numerics, physics, reverse engineering
  - Tutorial for FFTW Fast Fourier Transform library (2021)
  - InfraTec IR camera image reader (2021)
  - 15-digit accurate magnetic field computation (2023)
  - Macintosh 68k ROM reverse engineering (2026)
  - VMEC cleanup, re-implementation (VMEC++) and documentation
- hands-on electrical & mechanical engineering
  - High Voltage experiments, e.g., 25kV transformer
  - Deckel FP1 milling machine overhaul



Jonathan Schilling  
jonathanschilling  
Experimental Physicist, Code Archeologist

42 followers · 158 following

**abscab** Public  
Accurate Biot-Savart routines with Correct Asymptotic Behaviour  
● TeX ☆ 7

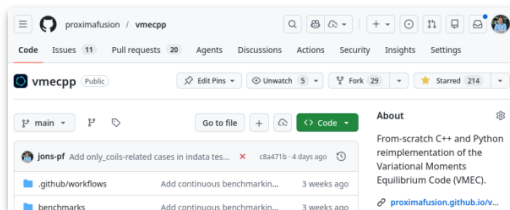
**educational\_VMEC** Public  
Serial VMEC for educational purposes (e.g. understanding the algorithms etc.)  
● Fortran ☆ 16 🍴 2

**irb** Public  
InfraTec IRBIS thermographic image (\*.irb) file reader in Java  
● Java ☆ 14 🍴 1

**fftw\_tutorial** Public  
A tutorial for the Fastest Fourier Transform in the West  
● Jupyter Notebook ☆ 25 🍴 4

**L-BFGS-B** Public  
Large-scale Bound-constrained Optimization  
● Fortran ☆ 14 🍴 2

**mac\_rom** Public  
Macintosh 68k ROM source - reverse engineered  
● Assembly ☆ 7 🍴 1



proximafusion / vmecpp

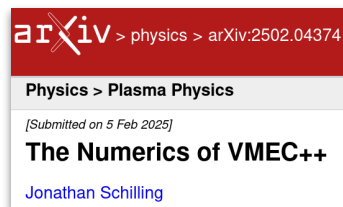
Code Issues 11 Pull requests 20 Agents Discussions Actions Security Insights Settings

**vmecpp** Public

main

About

From-scratch C++ and Python implementation of the Variational Moments Equilibrium Code (VMEC).  
proximafusion.github.io/v...



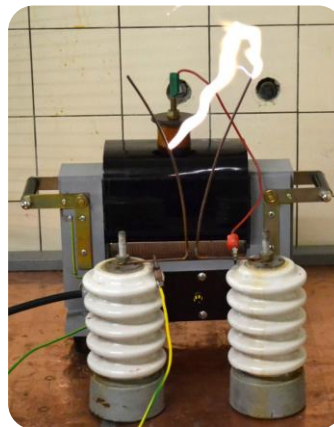
arXiv > physics > arXiv:2502.04374

Physics > Plasma Physics

[Submitted on 5 Feb 2025]

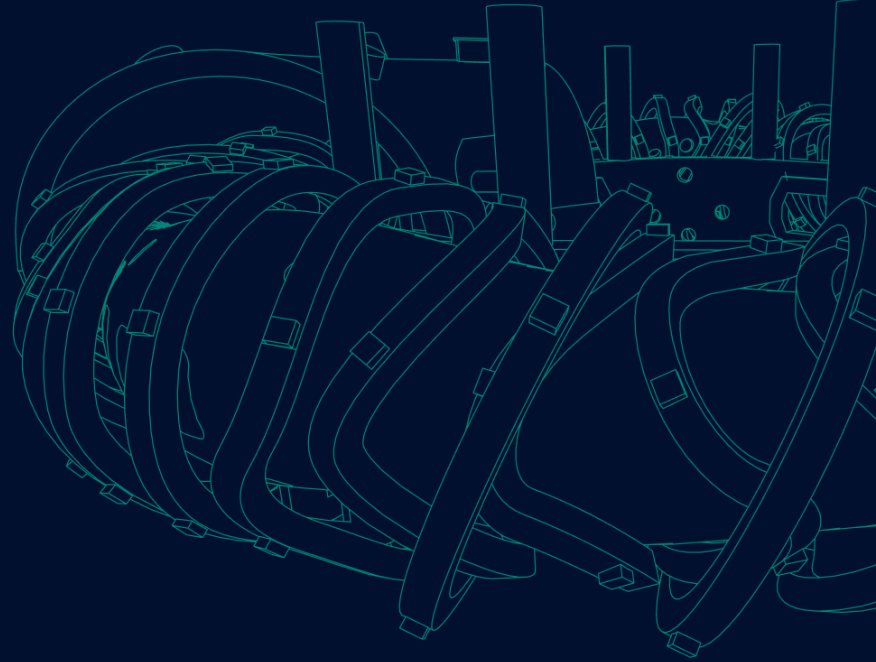
**The Numerics of VMEC++**

Jonathan Schilling



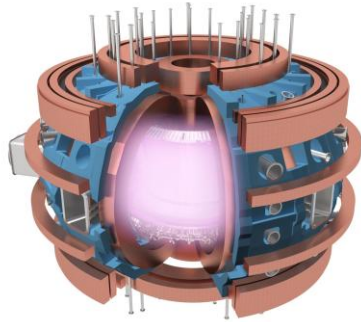
# Proxima Fusion

---



# Design, not control, holds the key to fusion

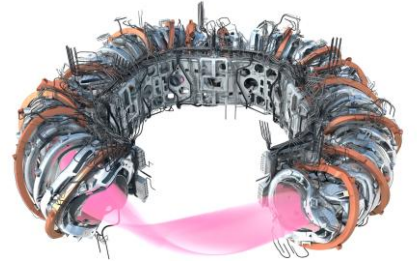
Tokamaks



**Tokamaks** create helical magnetic fields via external coils and a large plasma current, which leads to instabilities.

**Simple** to design, but **hard** to operate.

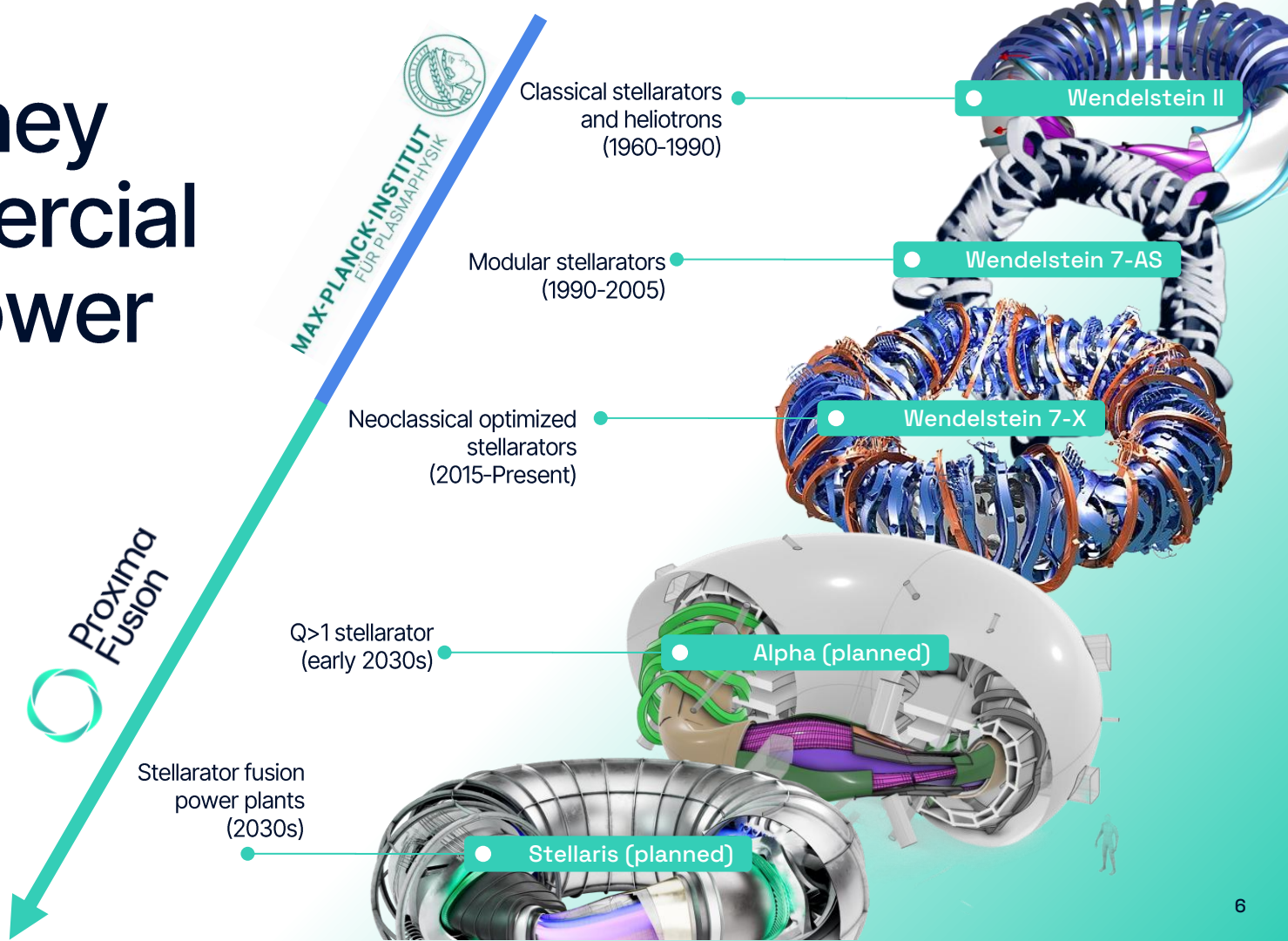
Stellarators

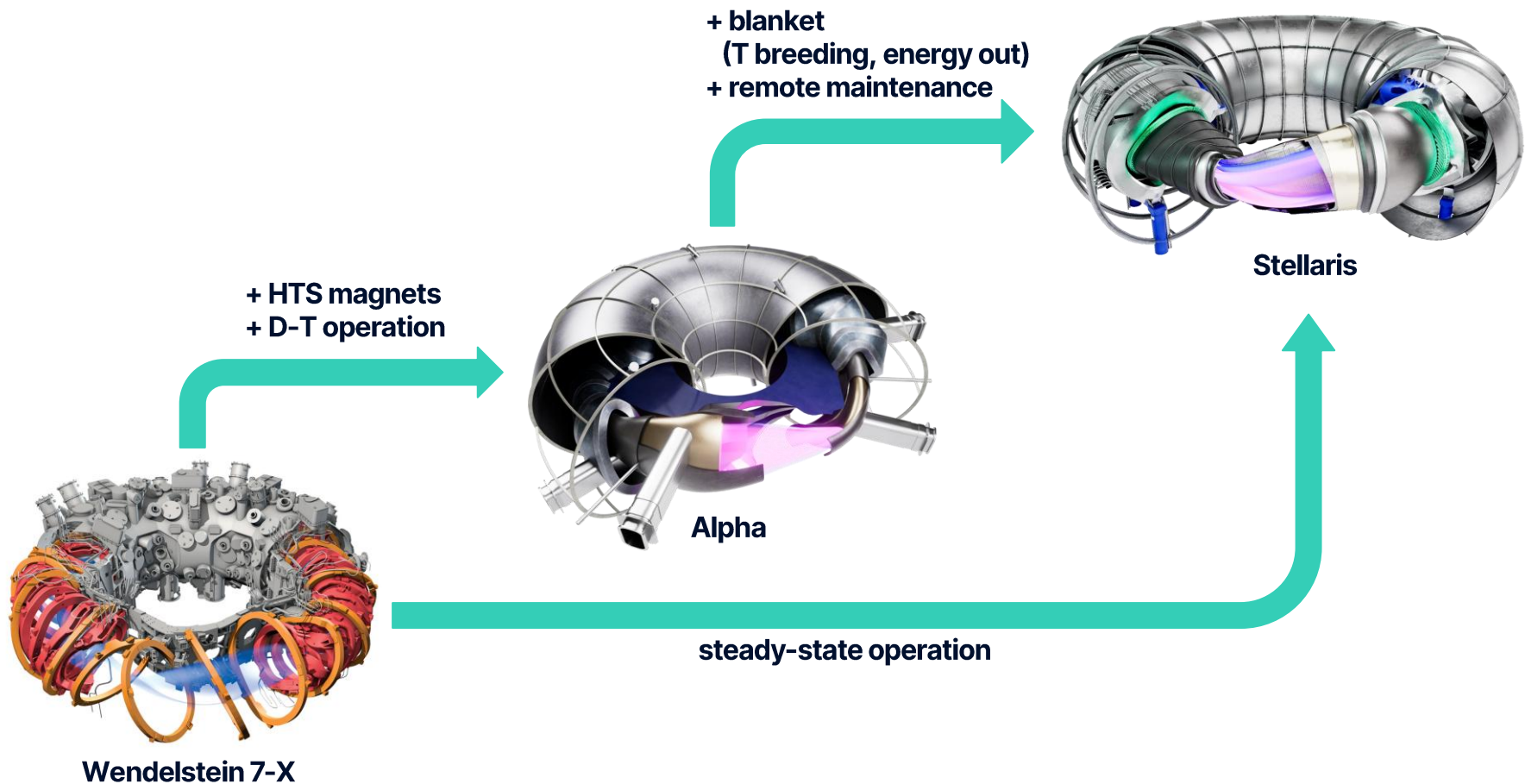


**Stellarators** create a helical magnetic field via only external coils, with potential for intrinsic stability.

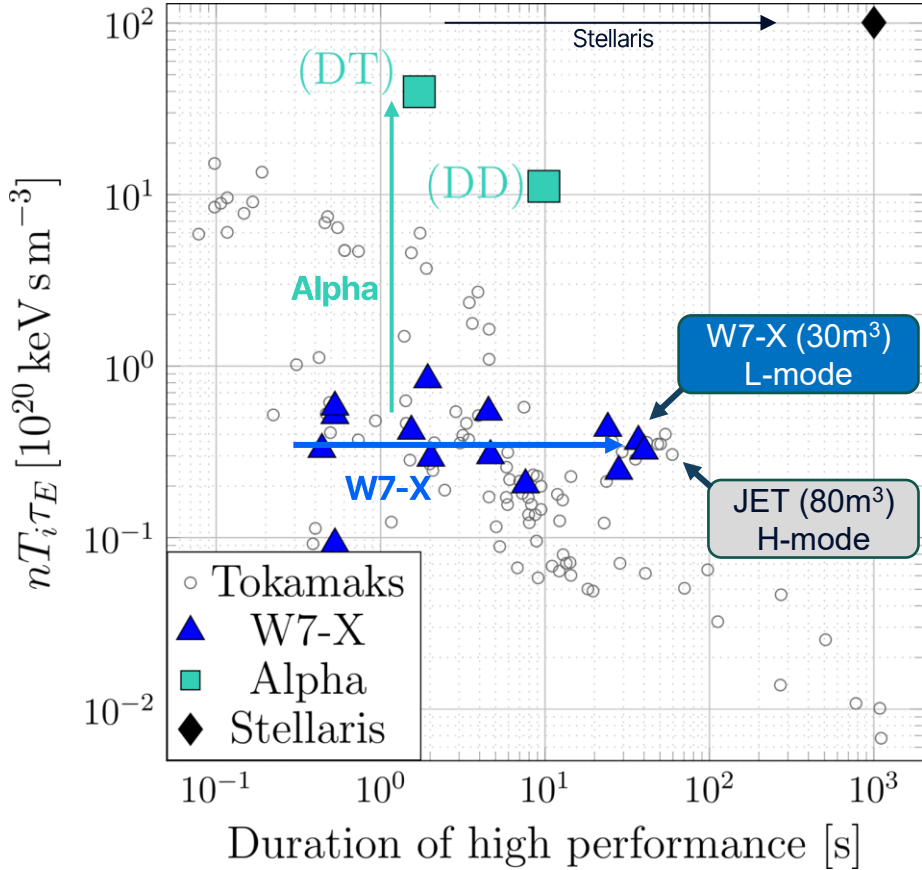
**Harder** to design, but **simple** to operate.

# The journey to commercial fusion power



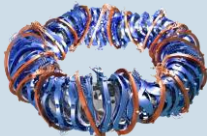


Fusion needs both high performance and stable, long-term plasma duration



# Proxima's path to a fusion power plant

2022



Operation of **W7-X**, the world's most advanced stellarator, at the Max Planck Institute for Plasma Physics (IPP)

MAX-PLANCK-INSTITUT  
FÜR PLASMAPHYSIK



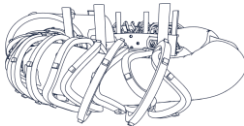
2023



Proxima Fusion

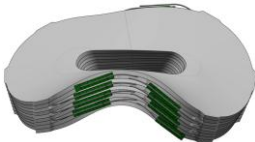
Spin-out of **Proxima Fusion from IPP**, to directly build on the record achievements of W7-X

2024



**Stellaris concept** – the world's first consistent design for a commercial stellarator power plant, published after peer review

2027



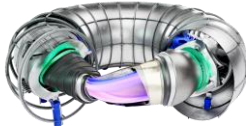
Completion of the demonstration magnet **Stellarator Model Coil (SMC)** and the design for the demonstration stellarator Alpha

Early 2030s



**Alpha** – The world's first stellarator that will demonstrate net energy gain

2030s

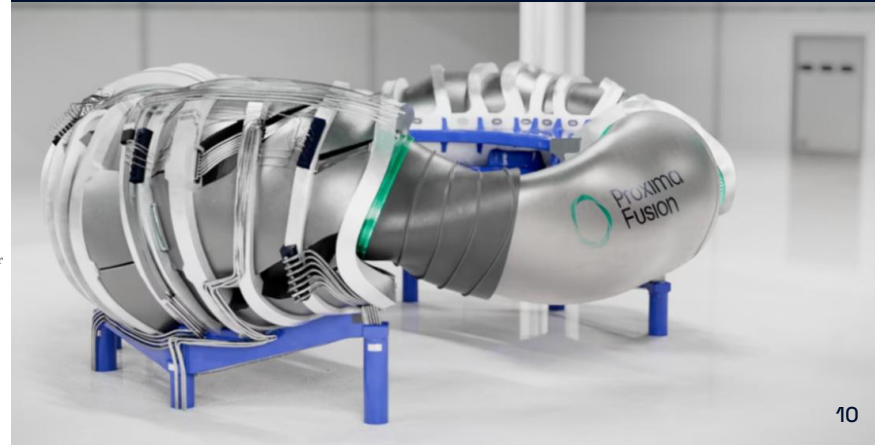


**Stellaris** - The first stellarator power plant that will feed fusion electricity into the grid

# Proxima Fusion In May 2026



- 150+ engineers, physicists and operators.
- Offices in Munich, Zurich and Oxford.
- \$230M+ of private and public financing + \$450M Bavarian commitment towards Alpha.

Peer-reviewed publication on Stellaris with academic partners in February 2025



# Stellaris: A high-field quasi-isodynamic stellarator for a prototypical fusion power plant

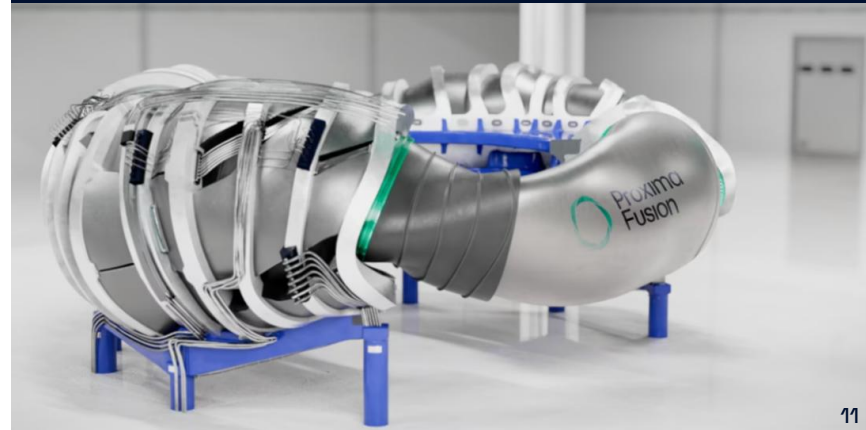
● Open access

J. Lion <sup>a</sup>  , J.-C. Anglès <sup>a</sup>, L. Bonauer <sup>a</sup>, A. Bañón Navarro <sup>b</sup>, S.A. Cadena Ceron <sup>a</sup>, ...

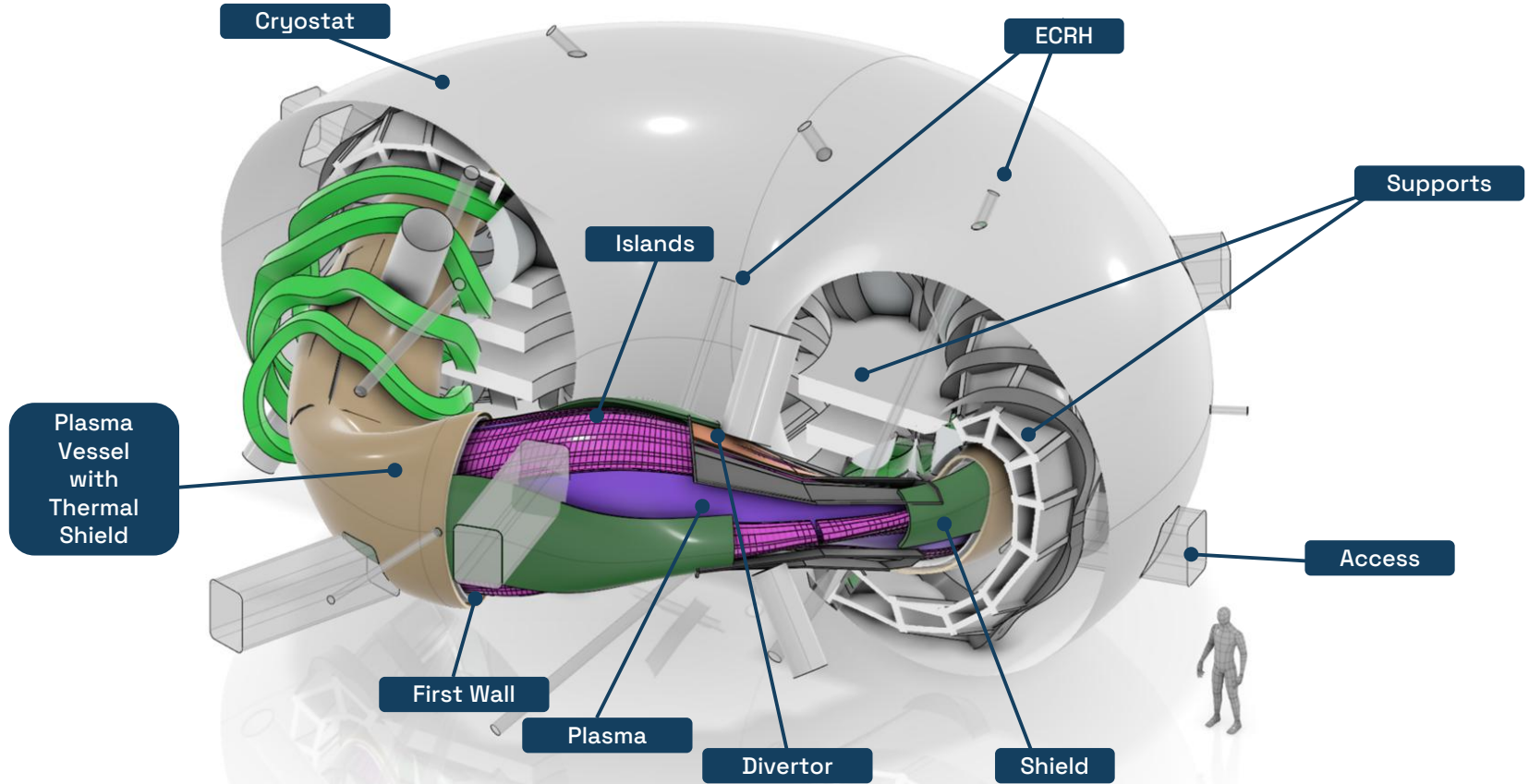
- QI Stellarator, based on SQUID configuration family (IPP)
- non-planar HTS magnets
- ECRH heating (50MW, 240 GHz), pellet fueling
- island divertor
- sector splitting for remote maintenance
  
- major radius . . . . . 12.7 m
- minor radius . . . . . 1.3 m
- plasma volume . . . . . 428 m<sup>3</sup>
- magnetic field on axis . . . 9 T
- peak triple product . . . . 124 · 10<sup>20</sup> keV s m<sup>-3</sup>
- peak fusion power . . . . . 2.7 GW
- peak electric power . . . . . 1 GW

“ The biggest development in fusion technology since the breakthrough in tokamaks a decade ago. ”

*D. Whyte (MIT), Wall Street Journal*



# Alpha - our net-energy demonstrator



## MARS De-risking Alpha assembly

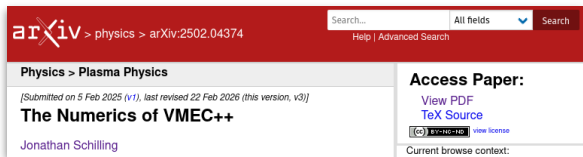
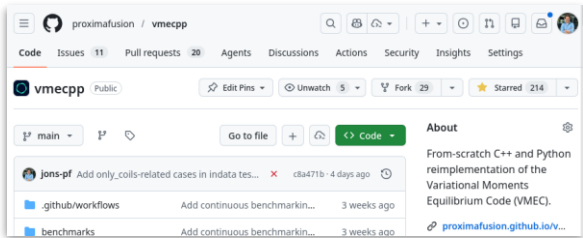
- The Maintenance & Assembly Research Sector (MARS) is our assembly testbed to prove we can manufacture and align real-scale and real-weight components.
- Proxima Fusion can leverage W7-X learnings like no other fusion company can do



# Open Science, Closed Engineering

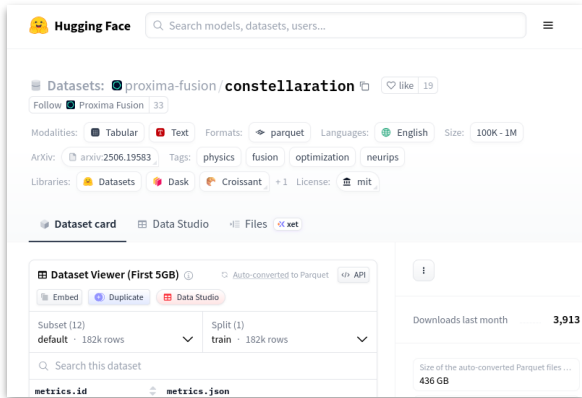
## VMEC++

Ideal MHD Equilibrium Solver



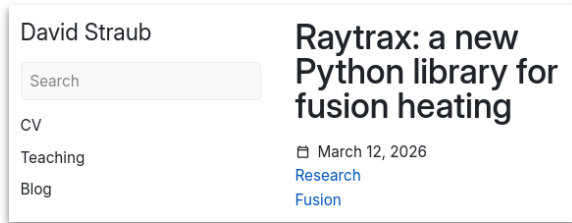
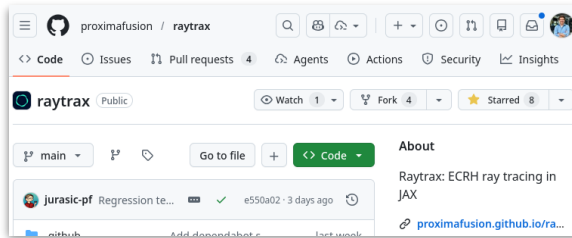
## ConStellation

Dataset of ~150k QI stellarators



## Raytrax

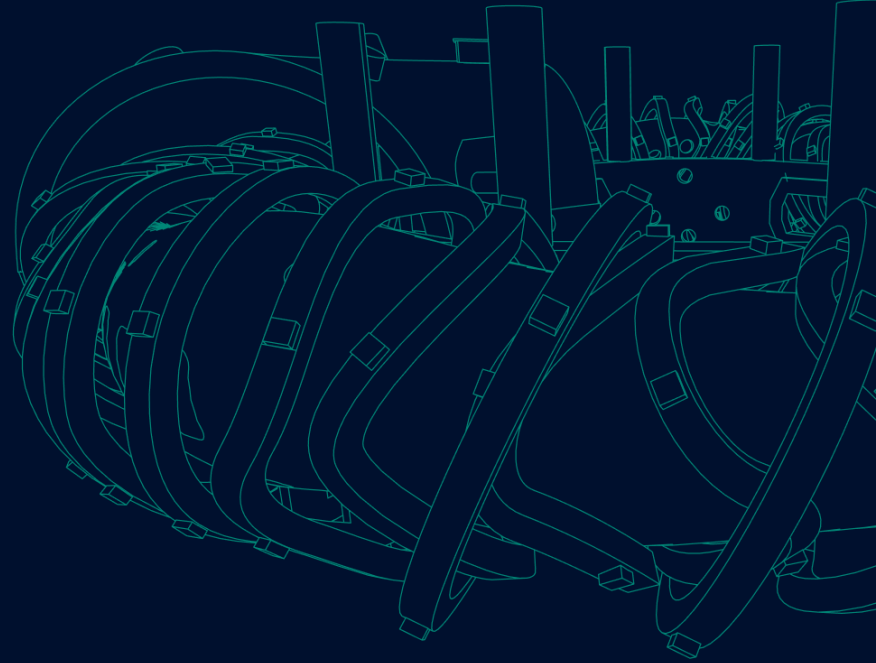
ECRH ray tracing in JAX



+ sponsorship of & presentations on [Open Source Software for Fusion Energy \(OSSFE\) conference](#)  
 + contributions to [OpenMC \(neutron transport\)](#), [cadquery \(CAD-as-code\)](#), [pants](#), [dagmc](#), [simsopt](#), ...  
 by Proxima Fusion team members + sponsorship

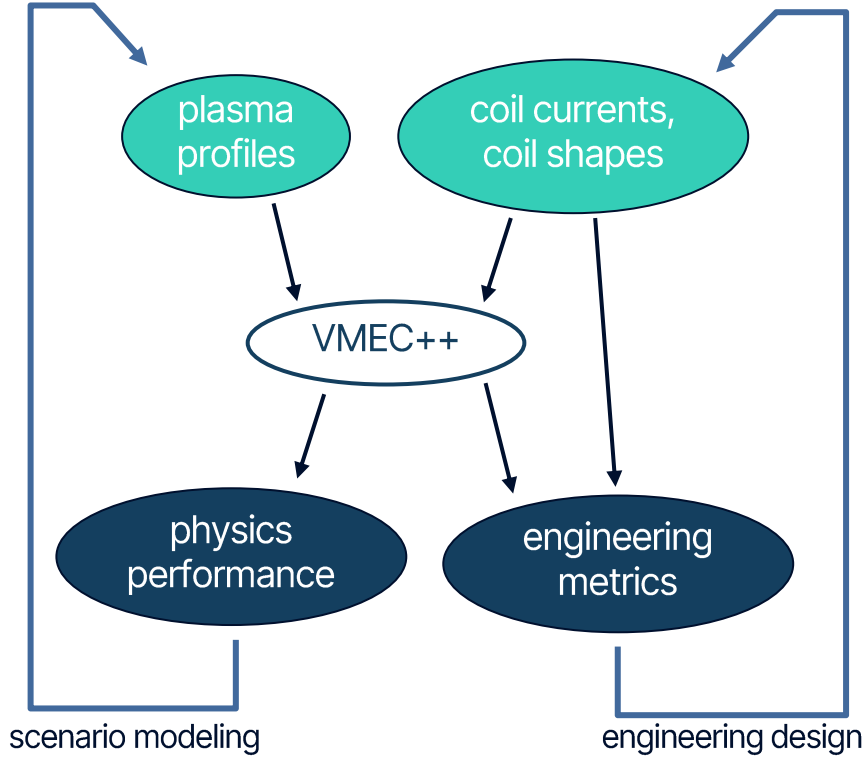
# VMEC+++

---

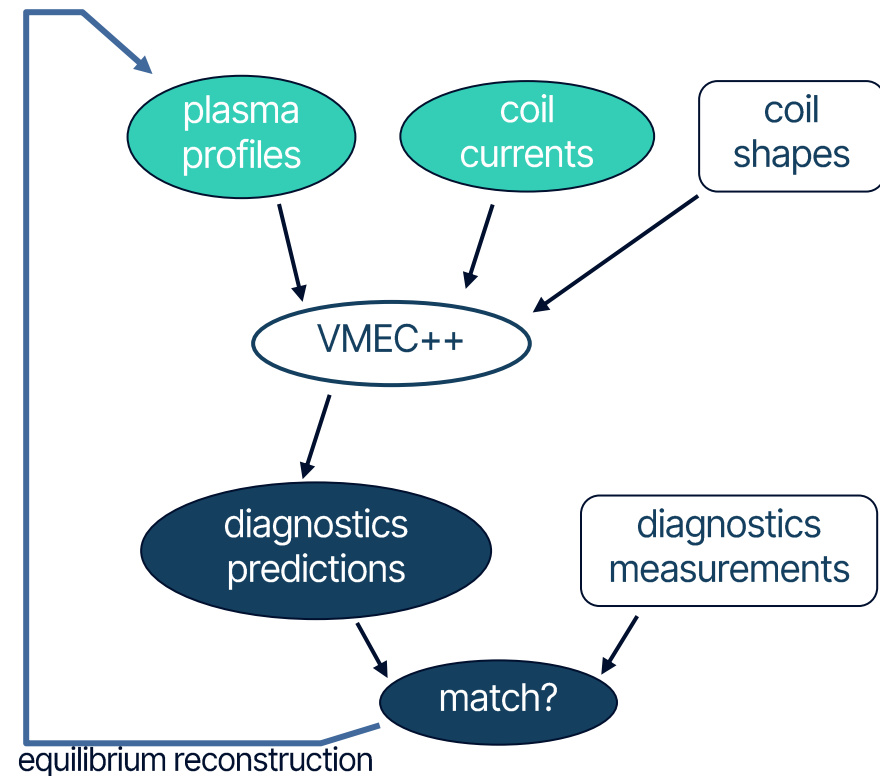


# Stellarators Design == Equilibrium Reconstruction ?

## Stellarator Design



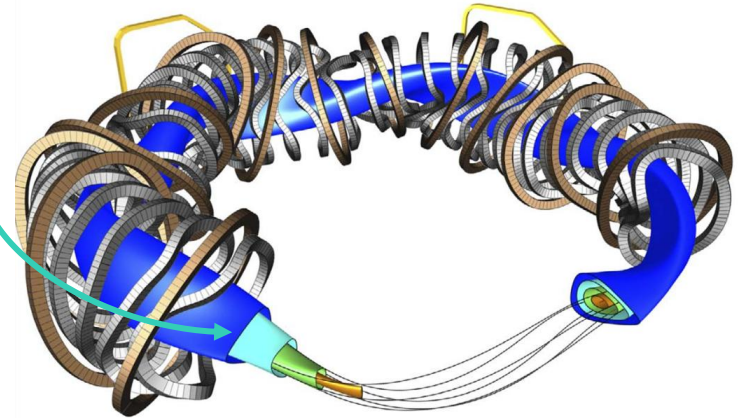
## Equilibrium Reconstruction

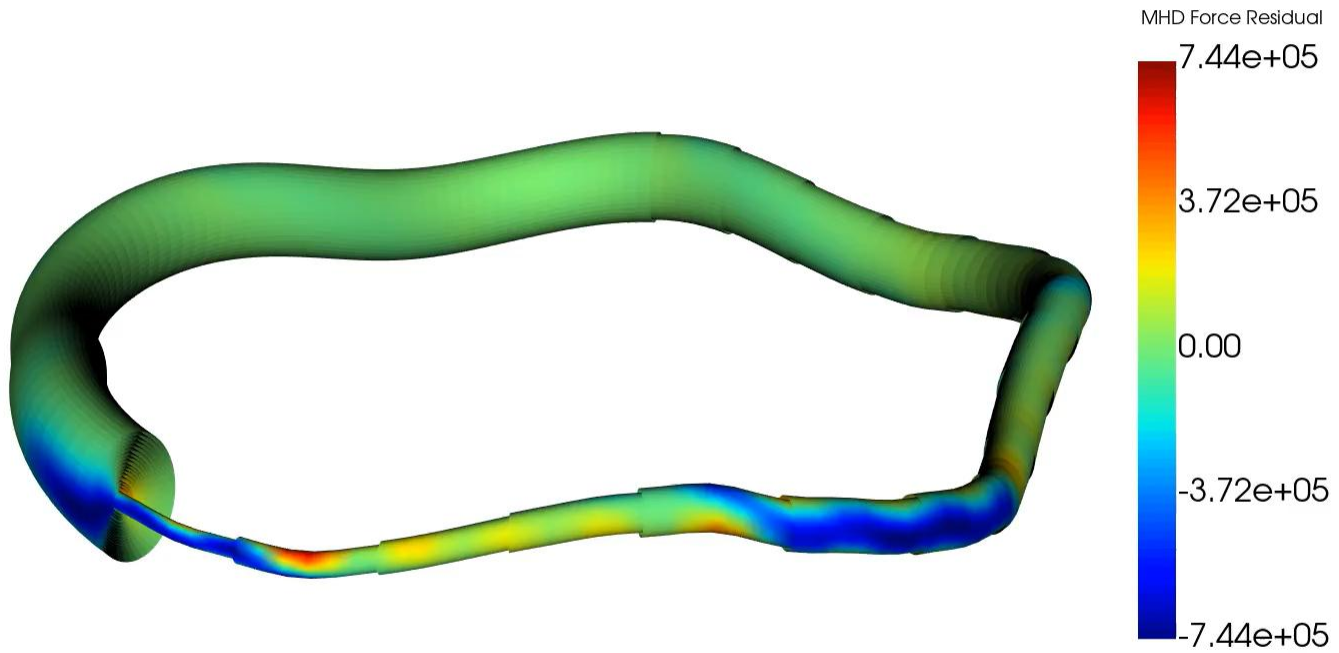


# VMEC: flux surfaces $\Rightarrow$ force balance

- Ideal MHD  $\Leftrightarrow$  nested flux surfaces exist
- goal: move flux surfaces into equilibrium  
 $\Rightarrow$  MHD force balance  $\nabla p = \mathbf{j} \times \mathbf{B}$
- Stellarators need numerical solvers for this, due to the inherent 3D geometry
- VMEC is most well-established tool for this; so far only a single legacy implementation

Coils and Plasma of Wendelstein 7-X







# How VMEC++ came about

## Status Quo (1983 - 2019)

- VMEC central to stellarator modeling toolchain
- only one legacy implementation in Fortran
- no documentation of core numerical approach
- memory management broken, random segfaults
- didn't want to build next-gen \$bln stellarator based on single black-box code


## The Journey (2020 - 2022)

- survive lots of push-back from PhD advisors
- put PhD topic aside, fully focus on VMEC re-write
- define reference cases, dump reference data from Fortran implementation
- 1:1 re-write in  Java
- clean port to  python™

## The Solution (2023-2025)

- co-founded Proxima Fusion
- again from scratch in C++ with OpenMP → VMEC++
- Enrico, Veronika & Philipp
  - performance optimization
  - rigorous integration tests
  - Verification & Validation
  - open-source release
- since 2024 in daily use at Proxima Fusion
- “The Numerics of VMEC++” docs released along code
- growing example collection

VMEC++ is a  
modern,  
Python-friendly  
VMEC implementation



# Easy to install

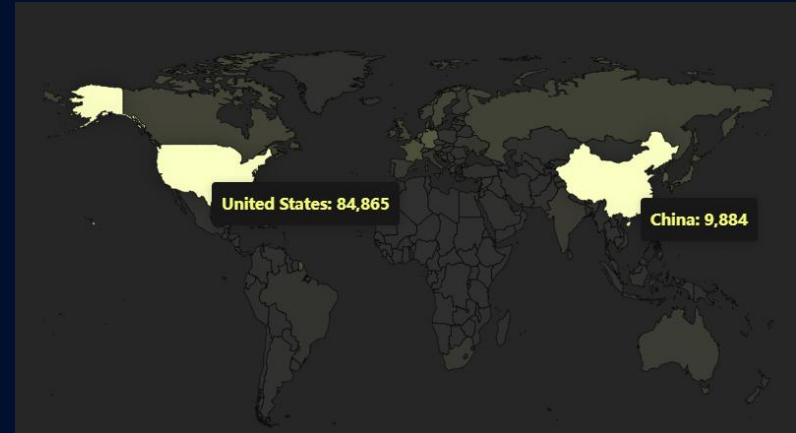
```
pip install vmecpp
```

- **All major Linux flavors** and **latest MacOS** versions are supported
- **Direct integration in C++ projects** via the CMake and Bazel build systems is also possible



Ruff code style black license MIT python 3.10 ☆ 222 stars  
DOI 10.5281/zenodo.14800158

Install and test Python package passing C++ core tests passing  
Publish wheels to PyPI passing



# Easy to use

```
vmec_output = vmecpp.run(vmec_input)
```

- **Python-friendly**, functional programming-style API
- Error reporting via **Python exceptions**  
(or C++ exceptions if invoking the C++ core directly);  
*no segfaults, no aborts, no C++ stacktraces in Python*
- **No global state**: plays well with outer parallelization
- **Integrates with [SIMSOPT](#)**, a popular stellarator optimization framework

```
def run(
    input: VmecInput,
    max_threads: int | None = None,
    verbose: bool = True,
    restart_from: VmecOutput | None = None,
) -> VmecOutput:
```

```
# Construct input from a classic Fortran input file
vmec_input = vmecpp.VmecInput.from_file("input.w7x")

# Run VMEC++.
vmec_output = vmecpp.run(vmec_input)

# Save results as a classic wout file
vmec_output.wout.save("wout_w7x.nc")
```

```
# Or via the SIMSOPT-friendly API:
vmec = vmecpp.simsopt_compat.Vmec("input.w7x")
print(f"Computed plasma volume: {vmec.volume()}")
```

# Well tested and documented

---

- **Extensive test coverage** for both the C++ core and the Python interfaces
- **Extensive documentation** of the numerical methods available as permissively licensed latex sources: [The Numerics of VMEC++](#)

✓ Test Linux distros (debian:12, false)	51s
✓ Test Linux distros (debian:12, true)	3m 6s
✓ Test Linux distros (debian:testing-20250224, false)	52s
✓ Test Linux distros (debian:testing-20250224, true)	3m 17s
✓ Test Linux distros (fedora:41, false)	1m 2s
✓ Test Linux distros (fedora:41, true)	2m 39s
✓ Test Linux distros (ubuntu:22.04, false)	53s
✓ Test Linux distros (ubuntu:22.04, true)	3m 8s
✓ Test Linux distros (ubuntu:24.04, false)	1m 1s

arXiv > physics > arXiv:2502.04374

Physics > Plasma Physics

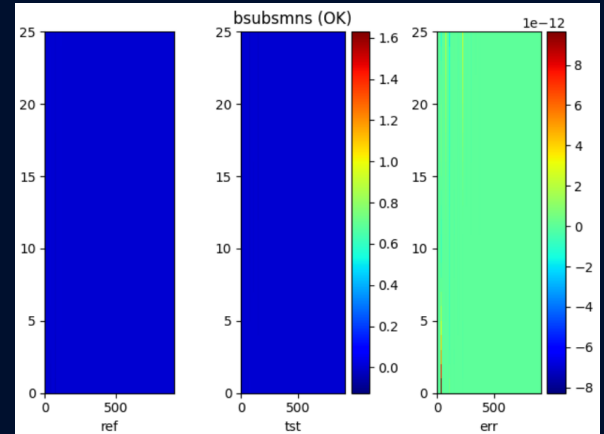
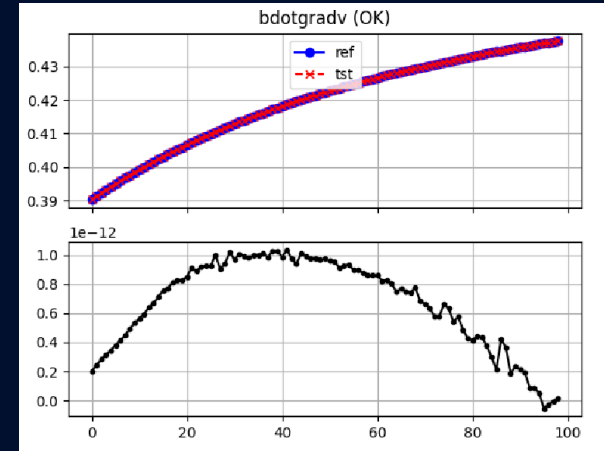
[Submitted on 5 Feb 2025]

**The Numerics of VMEC++**

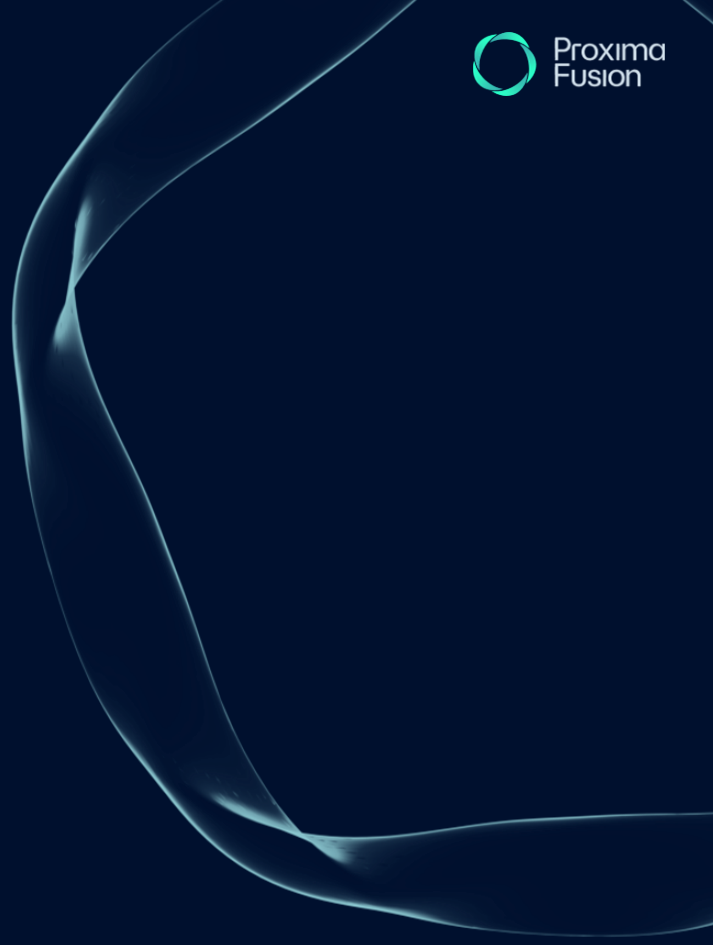
[Jonathan Schilling](#)

# Validated against the Fortran reference

- Both VMEC++ and VMEC 8.52 are run on **9 different configurations** with a parameter scan in plasma pressure, number of Fourier modes and number of flux surfaces, for a total of **219 test cases**, each checking  **$O(100)$  quantities**
- Source code is available at [github.com/proximafusion/vmecpp-validation](https://github.com/proximafusion/vmecpp-validation), together with a Docker image with the pre-built Fortran reference implementation for **full reproducibility**

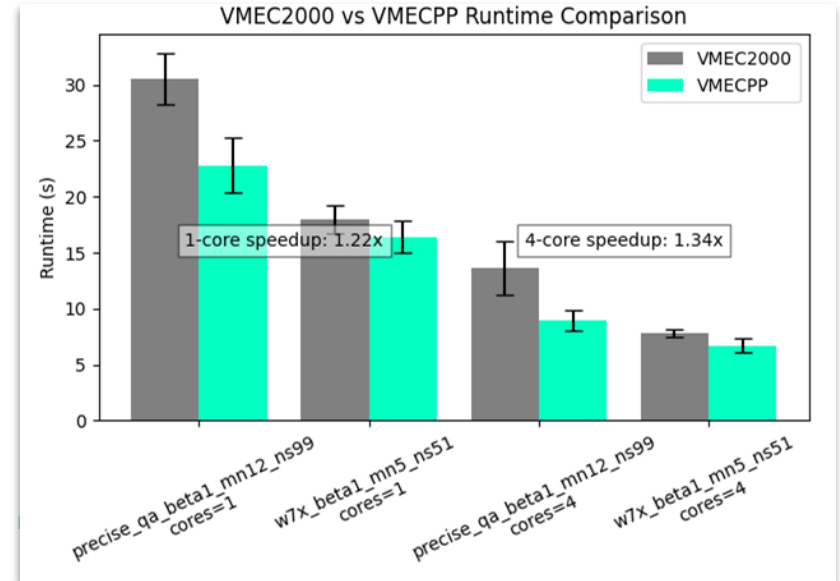


# Accelerating stellarator research with VMEC++



# Faster runtimes thanks to CPU vectorization

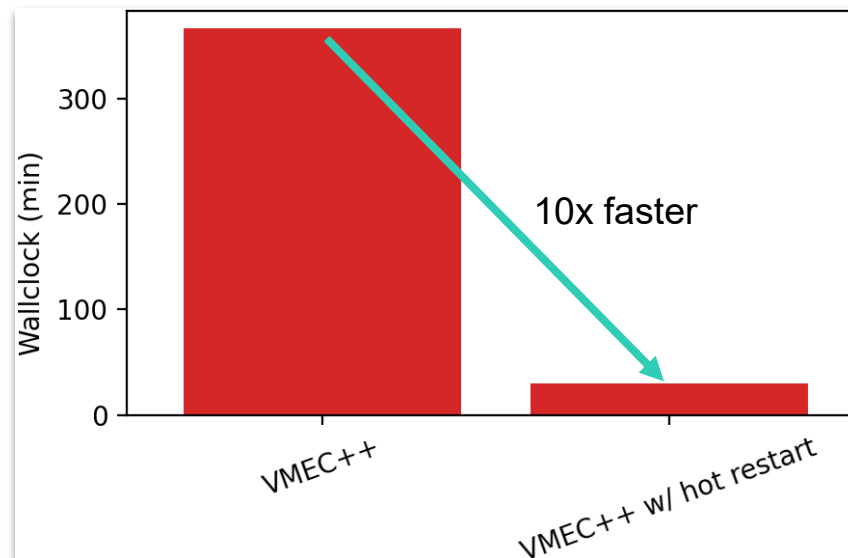
- VMEC++ internals have been optimized to be **friendly to modern CPU architectures**
- **Hot loops** have been rewritten so that modern compilers can **autovectorize** them
- The result is that **VMEC++ is typically faster than its Fortran predecessor**
- The benchmarks are publicly available at [github.com/proximafusion/vmecpp-benchmarks](https://github.com/proximafusion/vmecpp-benchmarks)



# Hot restart: 10x faster solutions with smart re-initialization

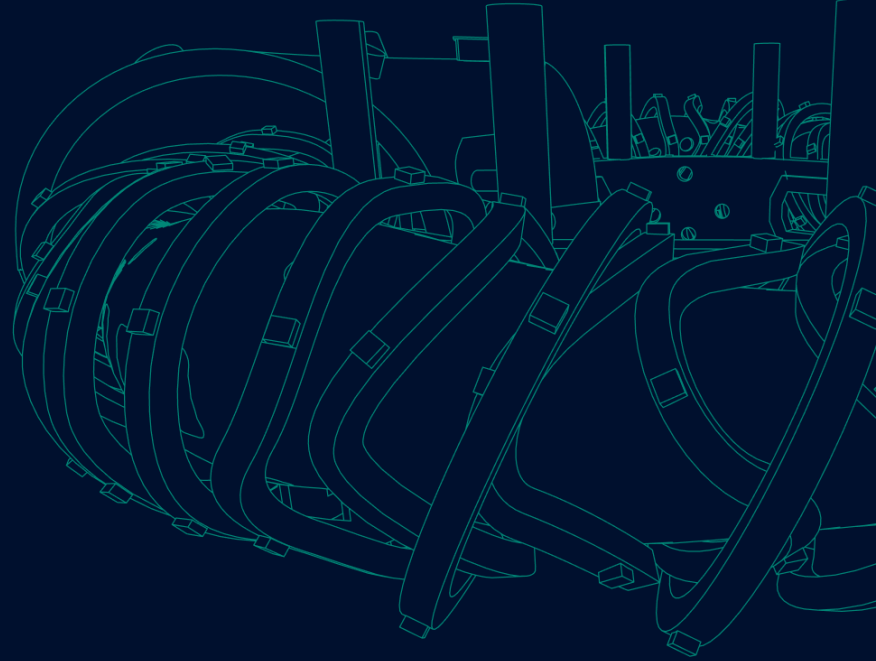
- Similar initial configurations converge to similar equilibria
- We can **reuse a previous solution** as “educated first guess” when solving a similar configuration!
- Incredibly useful for optimization: **gradient estimation involves many nearby evaluations** in finite-difference approximation
- “Hot restart” already known to speed up VMEC (e.g. from V3FIT equilibrium reconstructions) but non-trivial to integrate in Fortran VMEC

Runtime of a full optimization run



# ConStellaration

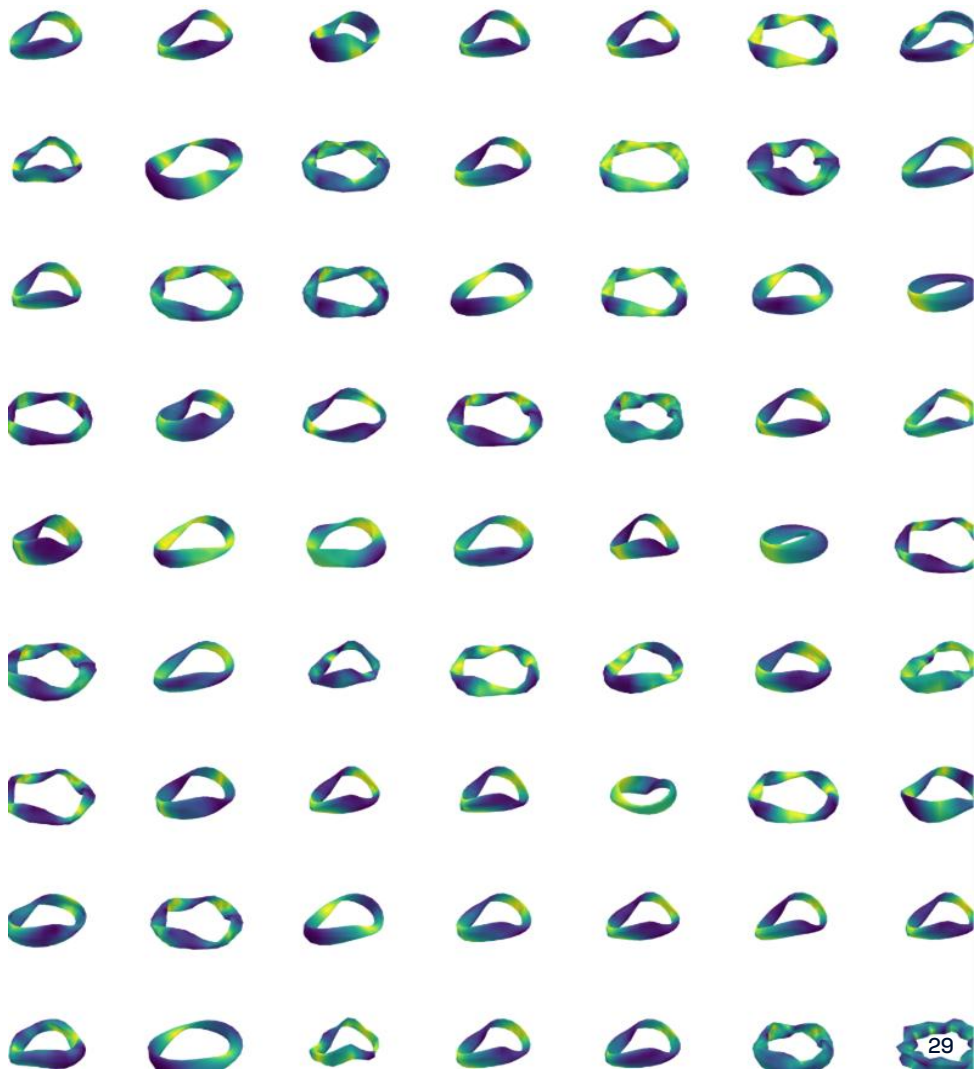
---



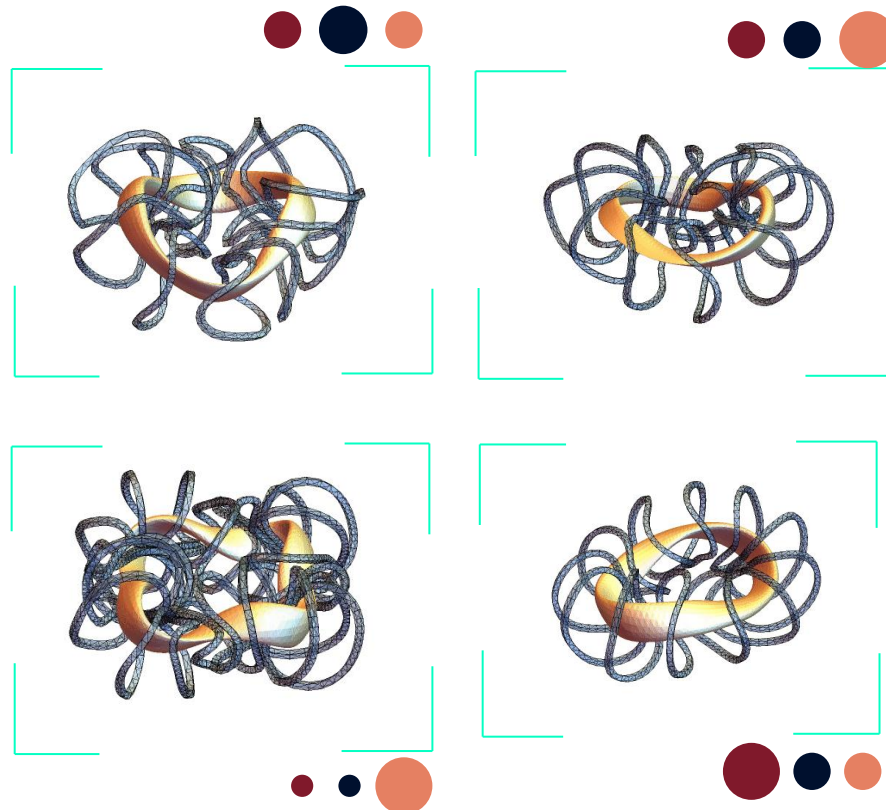
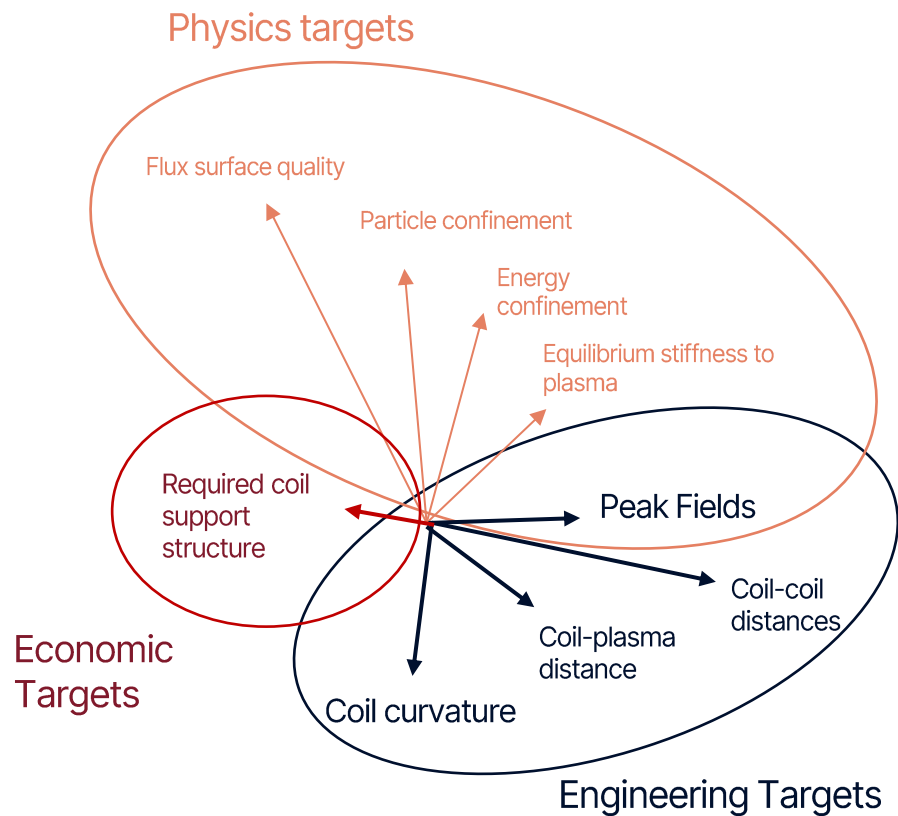
# What is ConStellaration?

- A ~160k dataset of QI-like stellarator plasma boundaries + equilibria + metrics:  
<https://huggingface.co/datasets/proxima-fusion/constellation>
- Three optimization benchmarks with tools and baselines:  
<https://github.com/proximafusion/constellation>  

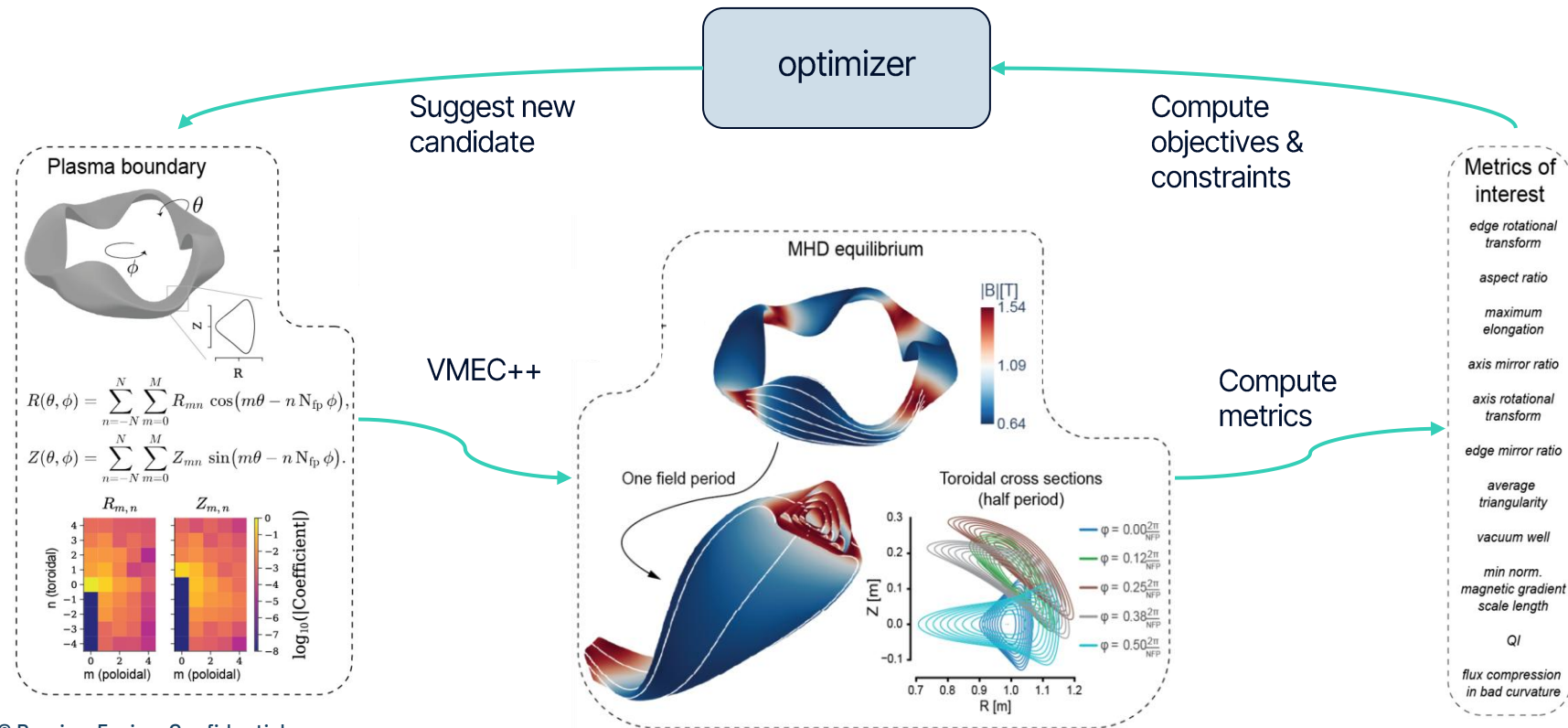
```
sudo apt-get install -y libnetcdf-dev  
pip install constellation
```
- A public leaderboard of the challenge with 🤖:  
<https://huggingface.co/spaces/proxima-fusion/constellation-bench>
- Paper: <https://arxiv.org/abs/2506.19583>



# Trade-offs exist between objectives



# Plasma boundary optimization using VMEC++ in the loop.



# Three Optimization benchmarks

## Geometric Problem

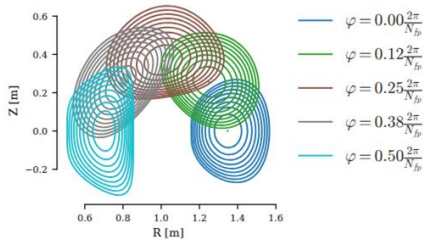
$$\min_{\Theta} \epsilon_{\max}$$

s.t.  $A \leq A^*$ ,

$$\bar{\delta} \leq \bar{\delta}^*$$

$$\tilde{t} \geq \tilde{t}^*$$

### Baselines



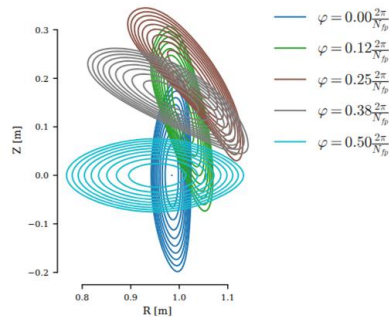
## Simple-to-build QI

$$\min_{\Theta} -\tilde{L}_{\nabla B}$$

s.t.  $\tilde{t} \geq \tilde{t}^*$ ,  $QI \leq QI^*$

$$\Delta \leq \Delta^*$$
,  $A \leq A^*$ 

$$\epsilon_{\max} \leq \epsilon_{\max}^*$$



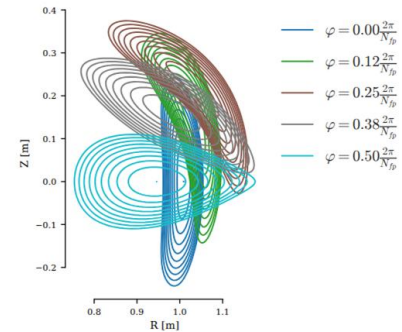
## MHD-stable QI (multi-objective)

$$\min_{\Theta} (-\tilde{L}_{\nabla B}, A)$$

s.t.  $\tilde{t} \geq \tilde{t}^*$ ,  $QI \leq QI^*$

$$\Delta \leq \Delta^*$$
,  $W_{\text{MHD}} \geq 0$ 

$$\langle \chi_{\nabla r} \rangle \leq \langle \chi_{\nabla r} \rangle^*$$



# Curious for more?



→ “Physics Lesson with Dr. Busch” – Apr 16<sup>th</sup> visit in Proxima Fusion Munich lab

# Curious for more?

- 
- Check out our OSS code, open issues/PRs: [github.com/proximafusion](https://github.com/proximafusion)
  - Jonathan Schilling: [jons@proximafusion.com](mailto:jons@proximafusion.com)
  - for ConStellation →  
Santiago Cadena: [scadena@proximafusion.com](mailto:scadena@proximafusion.com)

<https://www.proximafusion.com/careers>



Clean Energy,  
For Good.

