Optimizing the performance of fusion reactors at exascale

N. R. Mandell

Plasma Science and Fusion Center, MIT, Cambridge, USA

Massachusetts Institute of Technology

OAK RIDGE INSTITUTE FOR SCIENCE AND EDUCATION

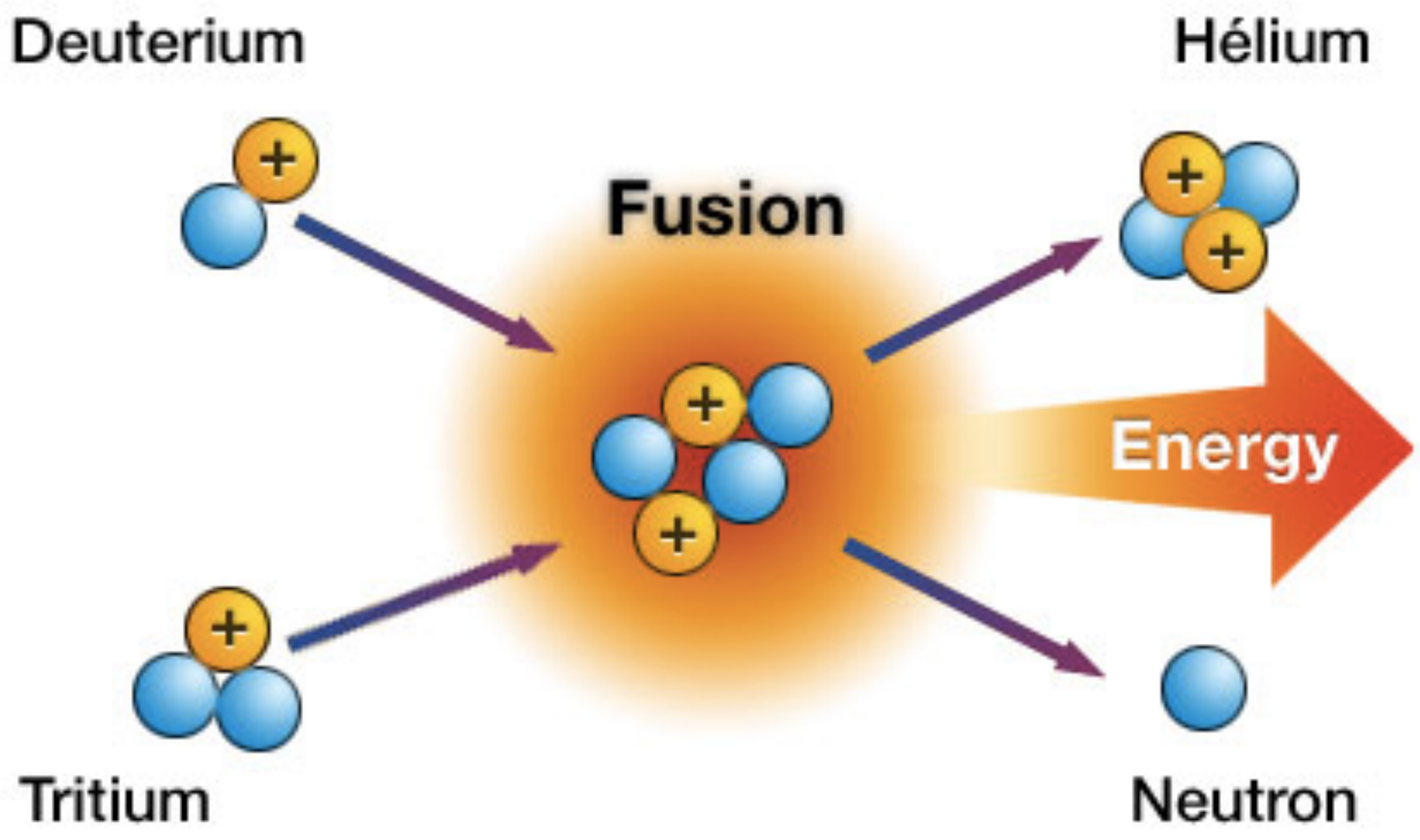
PPPL PRINCETON PLASMA PHYSICS LABORATORY

PSFC MIT Plasma Science & Fusion Center

Fusion 101: star power!

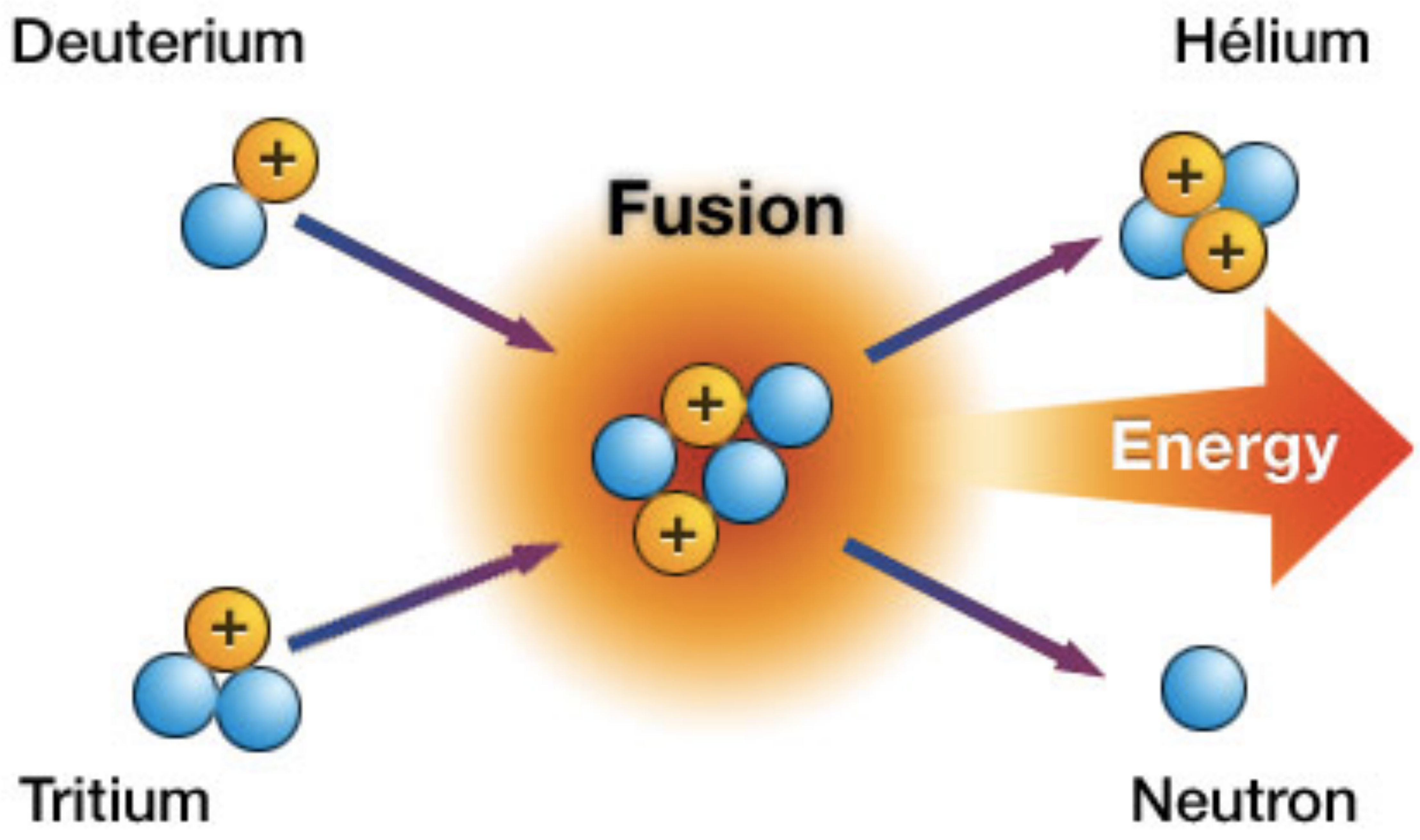
Fusion 101: star power!

- **Fusion** is the energy that powers the stars



Fusion 101: star power!

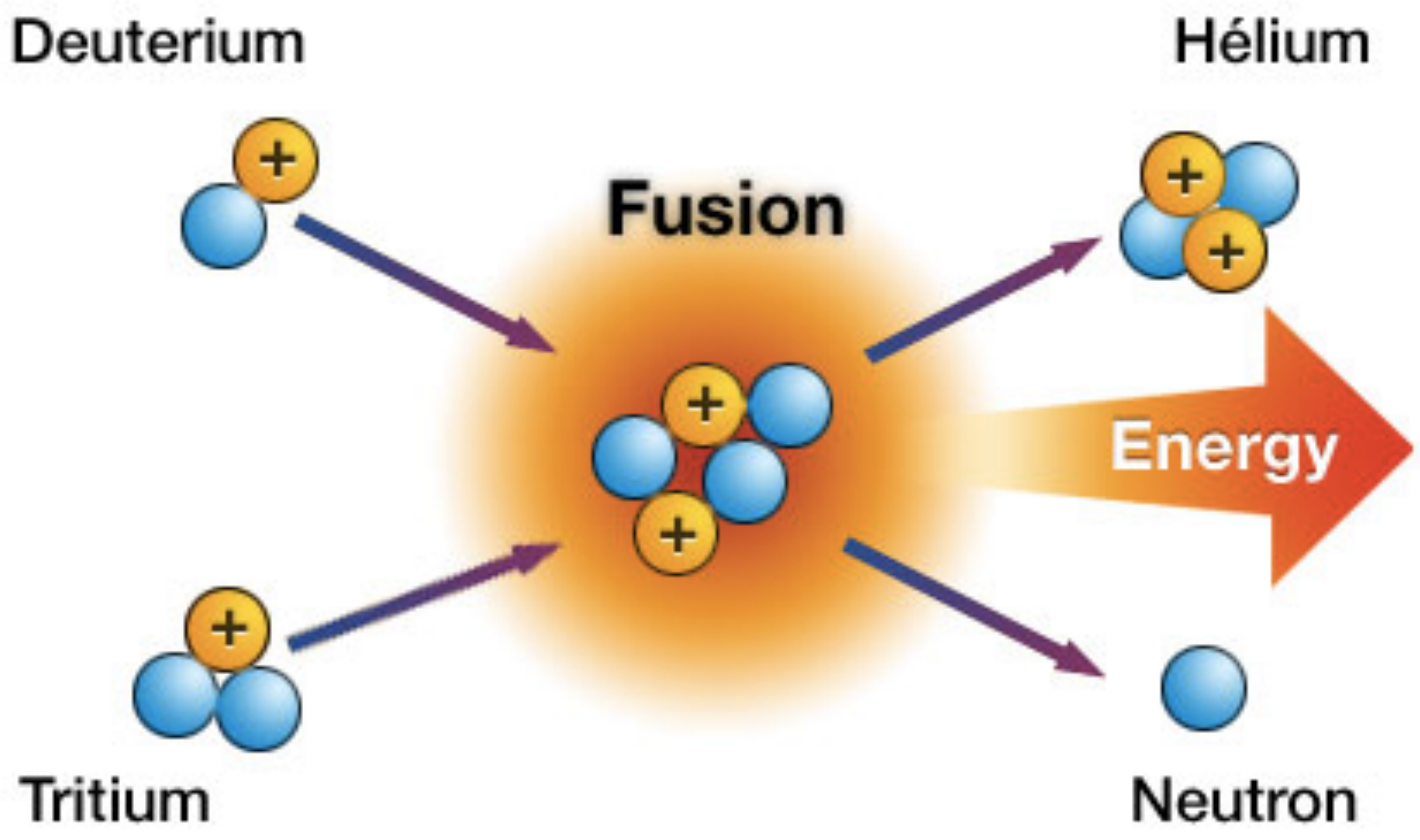
- **Fusion** is the energy that powers the stars



- In stars, enormous gravitational pressure produces the conditions necessary for fusion

Fusion 101: star power!

- **Fusion** is the energy that powers the stars

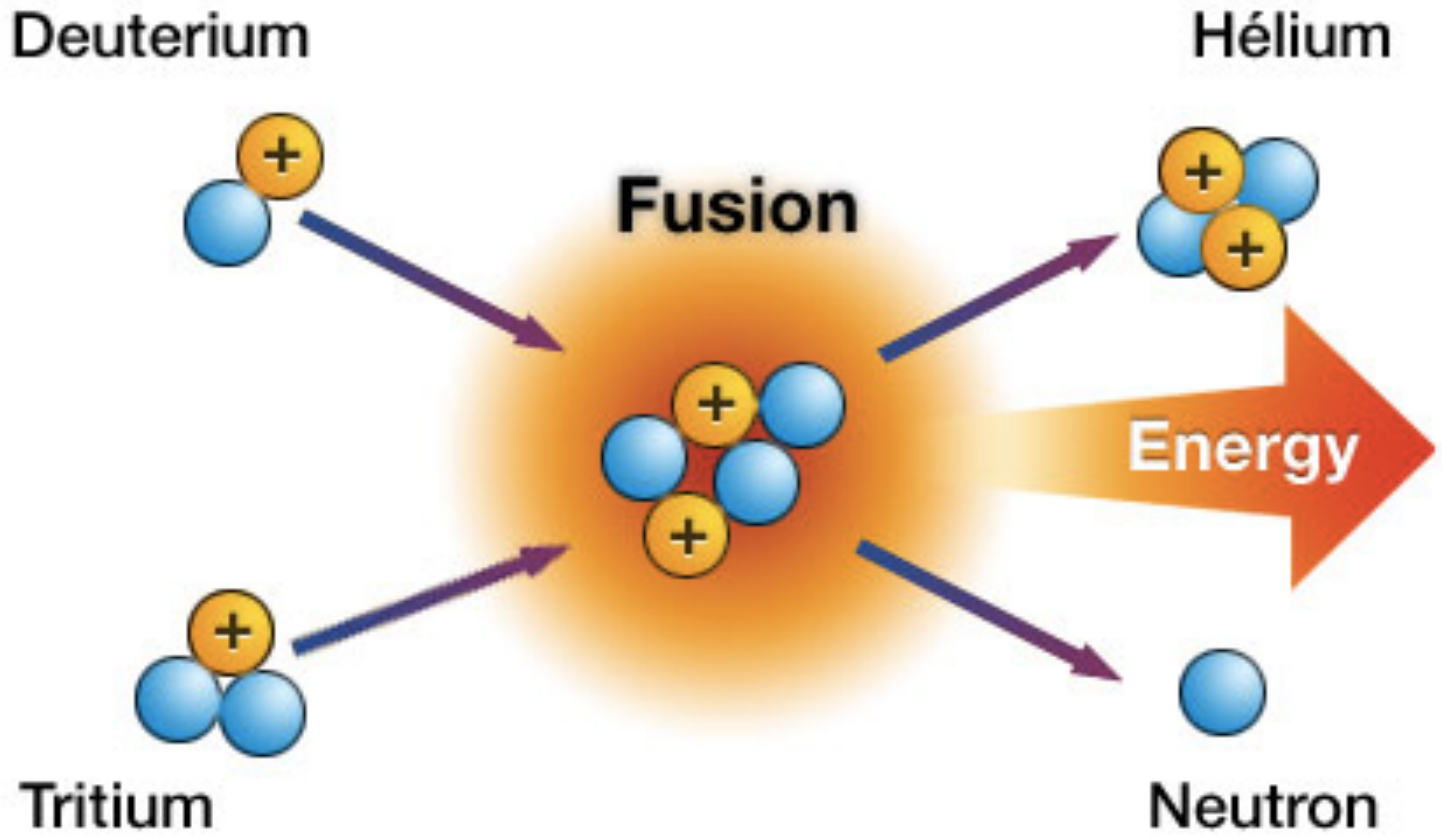


- In stars, enormous gravitational pressure produces the conditions necessary for fusion

- Here on Earth, we have to develop a different approach, which involves heating gas to 150 million C, 10x hotter than the surface of the sun, so that random collisions have enough energy to overcome the repulsive forces between ions

Fusion 101: star power!

- **Fusion** is the energy that powers the stars



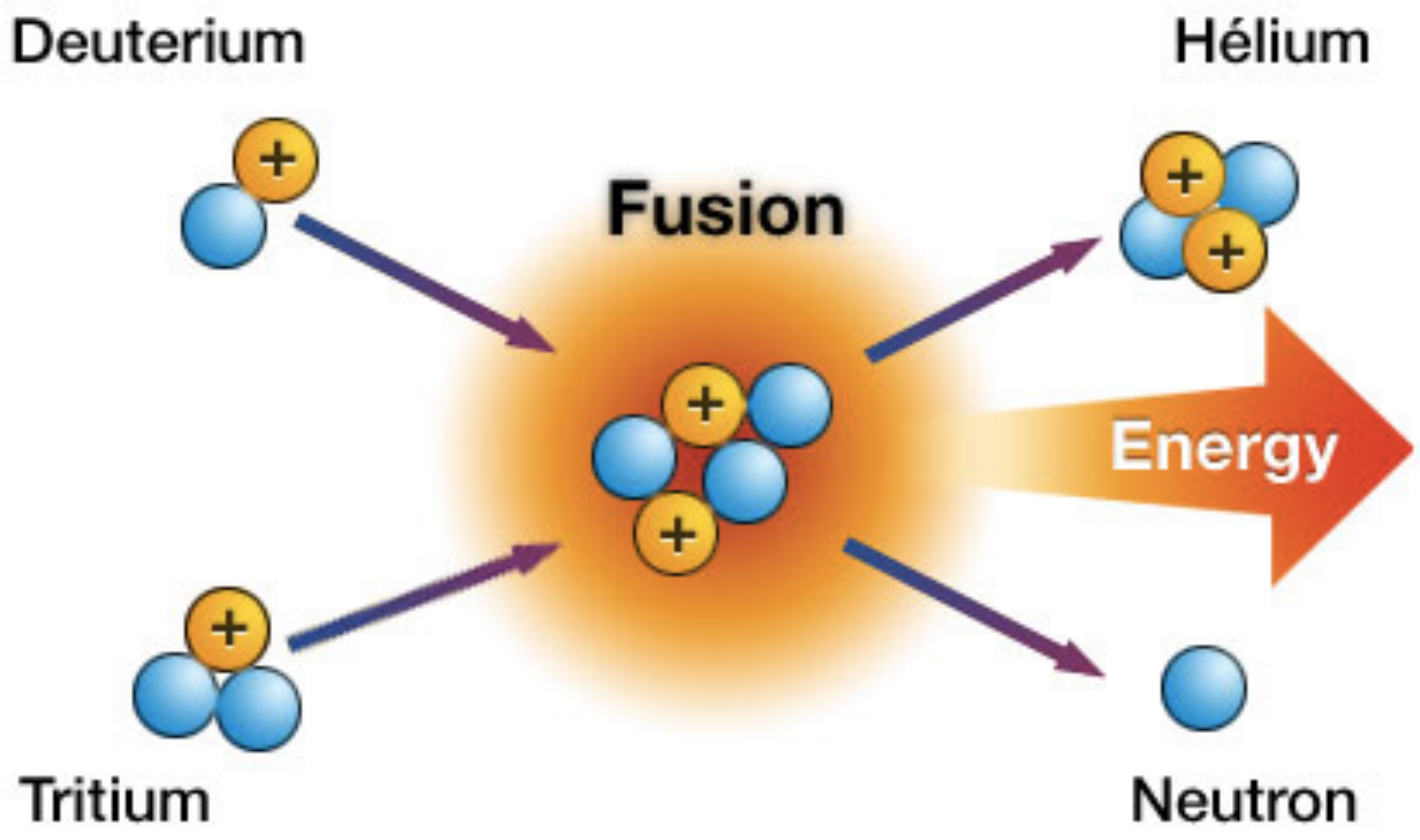
- In stars, enormous gravitational pressure produces the conditions necessary for fusion

- Here on Earth, we have to develop a different approach, which involves heating gas to 150 million C, 10x hotter than the surface of the sun, so that random collisions have enough energy to overcome the repulsive forces between ions

- The development of **fusion** as a viable commercial energy source will be a game-changer for the **health of the planet** in the coming decades, because it is a clean and virtually limitless energy source

Fusion 101: star power!

- **Fusion** is the energy that powers the stars



- In stars, enormous gravitational pressure produces the conditions necessary for fusion
- Here on Earth, we have to develop a different approach, which involves heating gas to 150 million C, 10x hotter than the surface of the sun, so that random collisions have enough energy to overcome the repulsive forces between ions
- The development of **fusion** as a viable commercial energy source will be a game-changer for the **health of the planet** in the coming decades, because it is a clean and virtually limitless energy source
    - Minimal carbon footprint, no long-lived radioactive byproducts, safe (no meltdown scenario)

Fusion 101: magnetic confinement

Fusion 101: magnetic confinement

Here on Earth, we have developed **magnetic confinement fusion**

Fusion 101: magnetic confinement

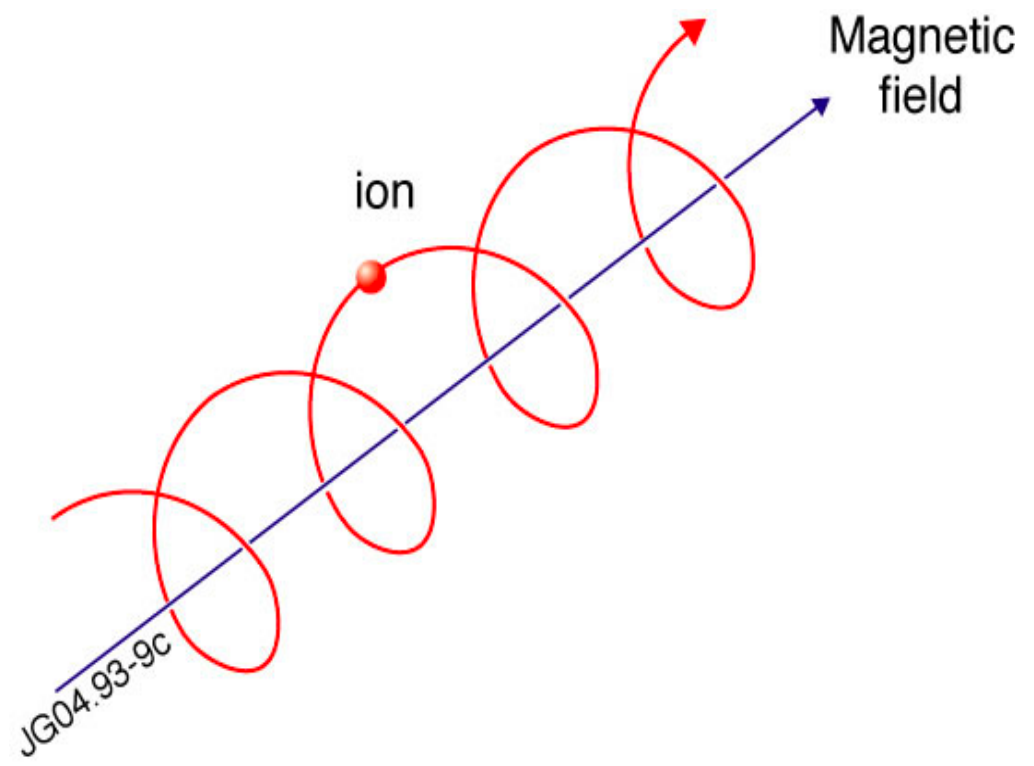
Here on Earth, we have developed **magnetic confinement fusion**

- At the extreme temperatures necessary for fusion, the fuel becomes a plasma (charged gas)

Fusion 101: magnetic confinement

Here on Earth, we have developed **magnetic confinement fusion**

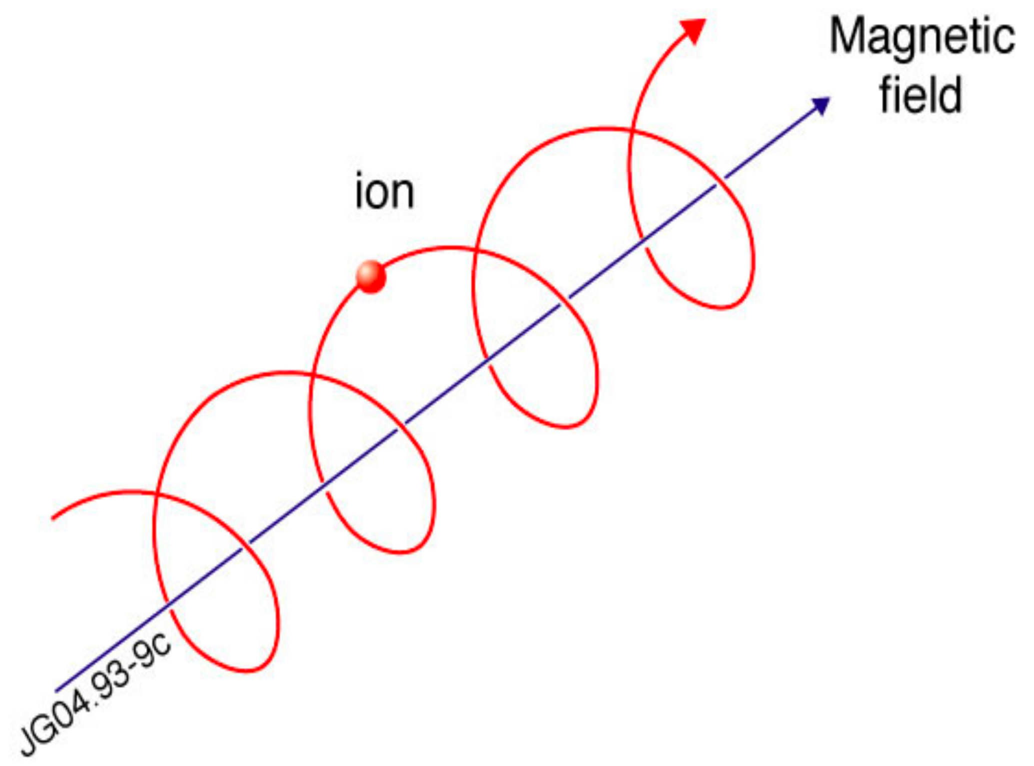
- At the extreme temperatures necessary for fusion, the fuel becomes a plasma (charged gas)
- Plasmas can be confined by strong magnetic fields, allowing fuel (and heat) to be confined in “magnetic insulation”

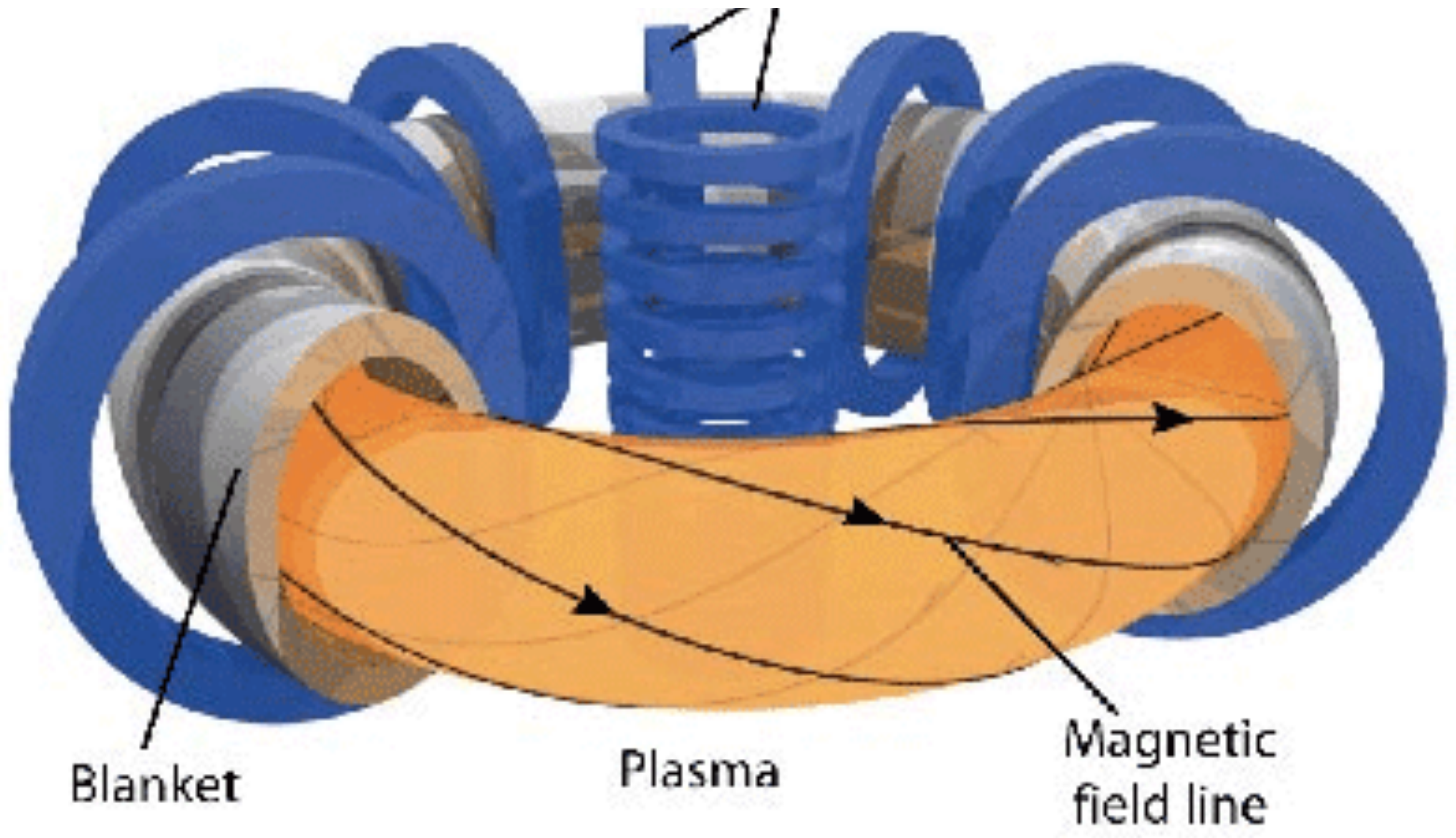


Fusion 101: magnetic confinement

Here on Earth, we have developed **magnetic confinement fusion**

- At the extreme temperatures necessary for fusion, the fuel becomes a plasma (charged gas)
- Plasmas can be confined by strong magnetic fields, allowing fuel (and heat) to be confined in “magnetic insulation”
- The most advanced designs are donut-shaped: tokamaks & stellarators

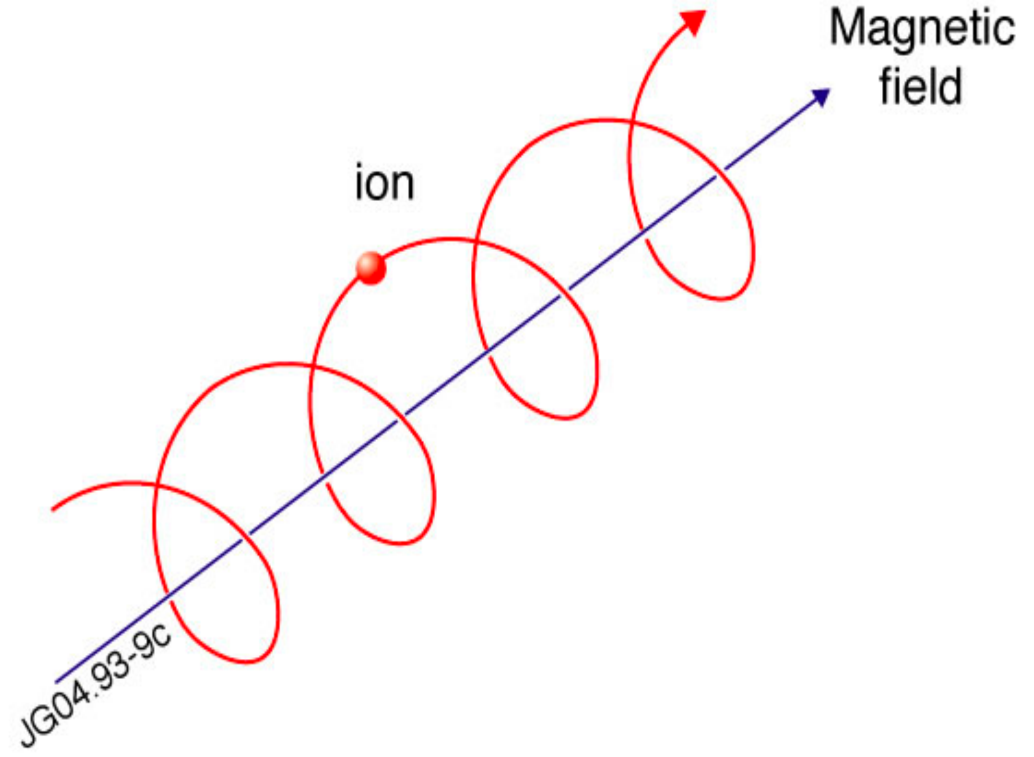


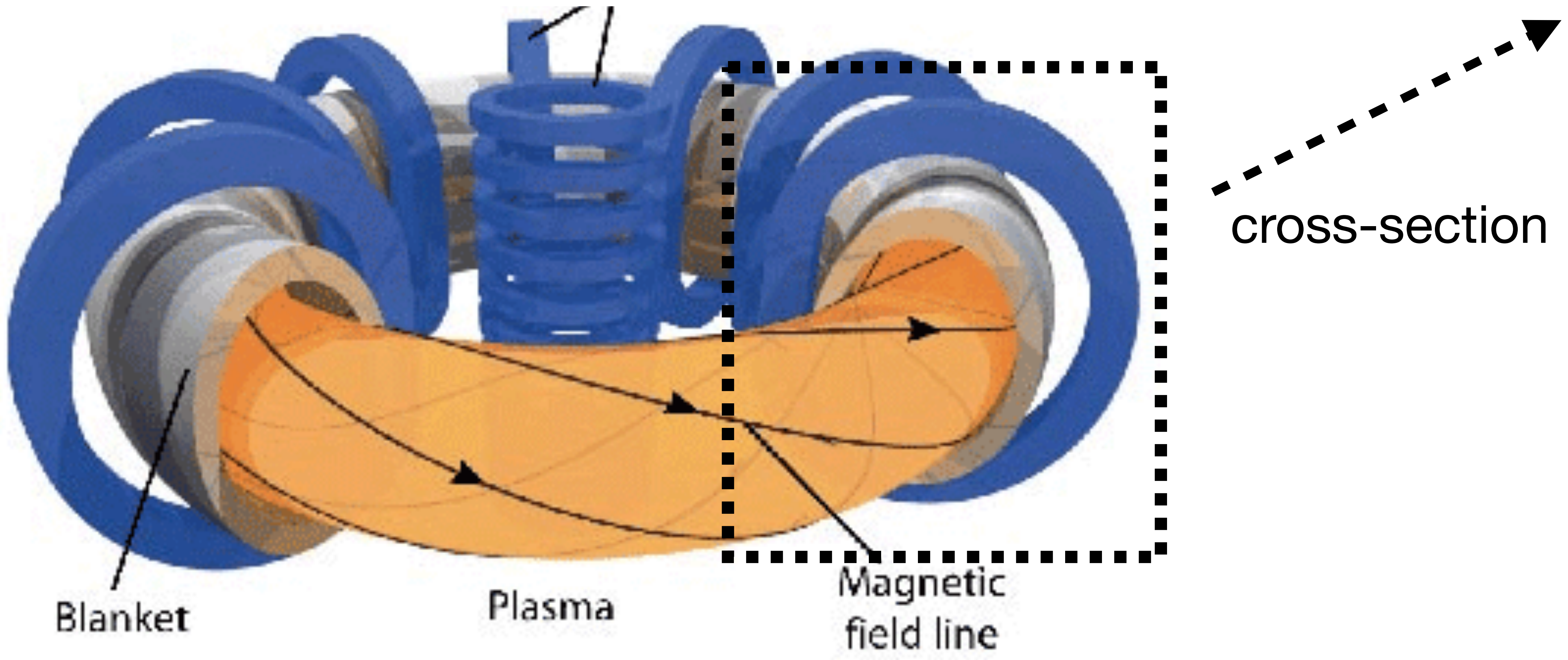


Fusion 101: magnetic confinement

Here on Earth, we have developed **magnetic confinement fusion**

- At the extreme temperatures necessary for fusion, the fuel becomes a plasma (charged gas)
- Plasmas can be confined by strong magnetic fields, allowing fuel (and heat) to be confined in “magnetic insulation”
- The most advanced designs are donut-shaped: tokamaks & stellarators
    - In the “core”, the magnetic field lines are “closed”, wrapping around the concentric surfaces of the donut





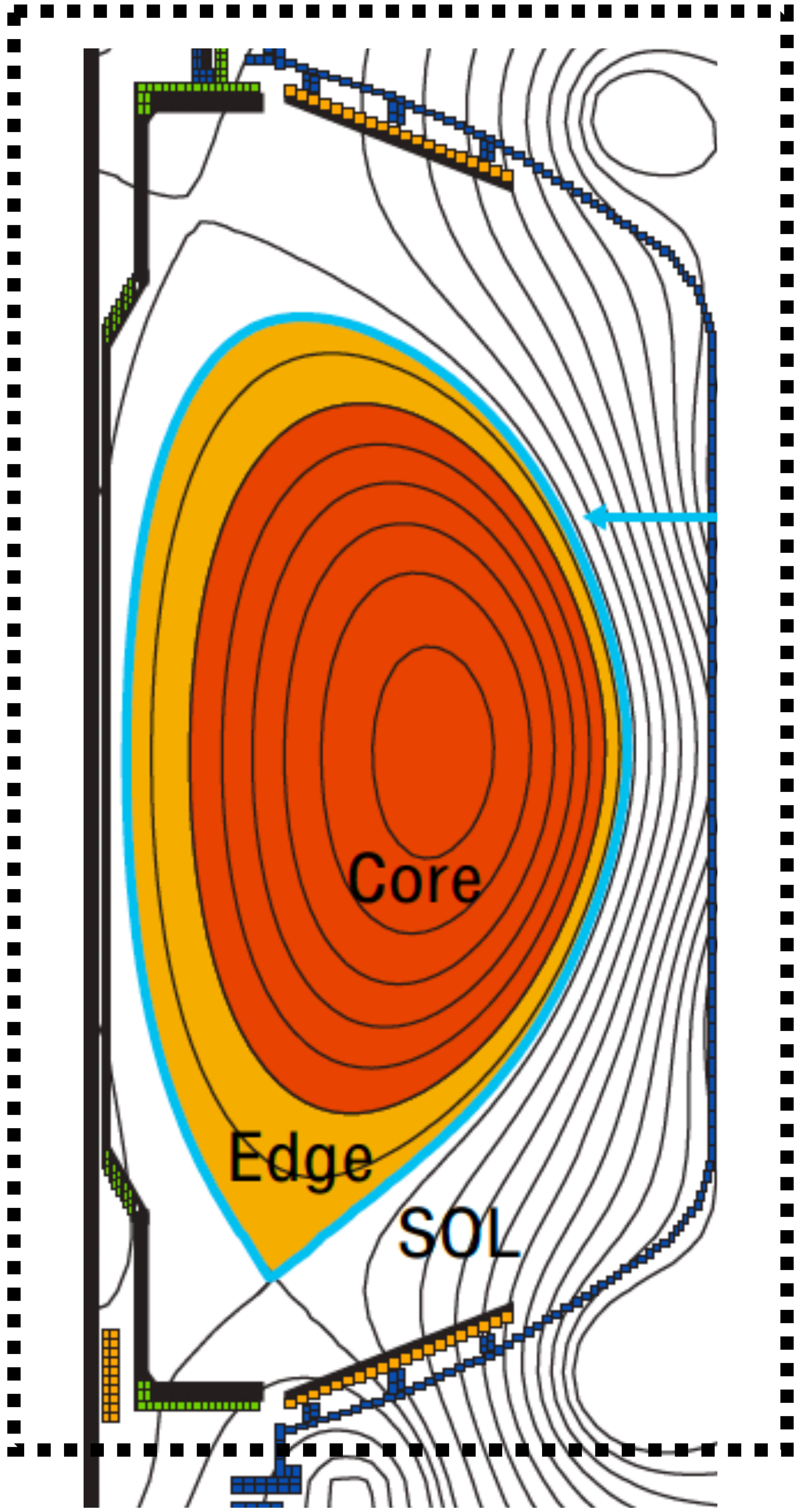
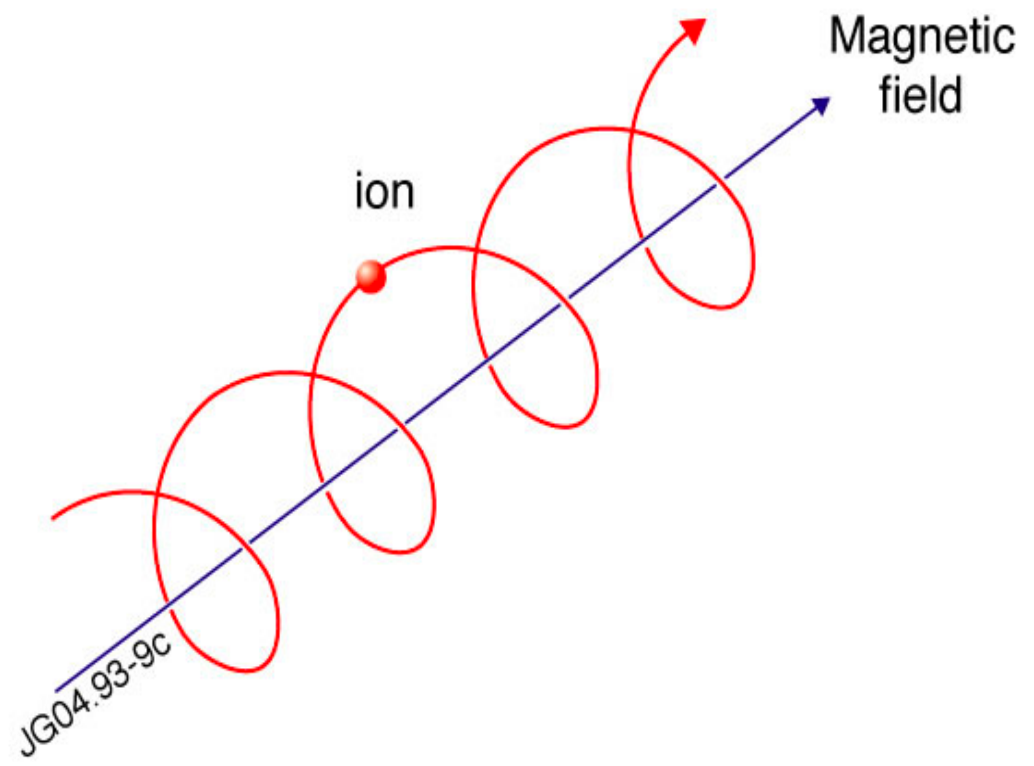


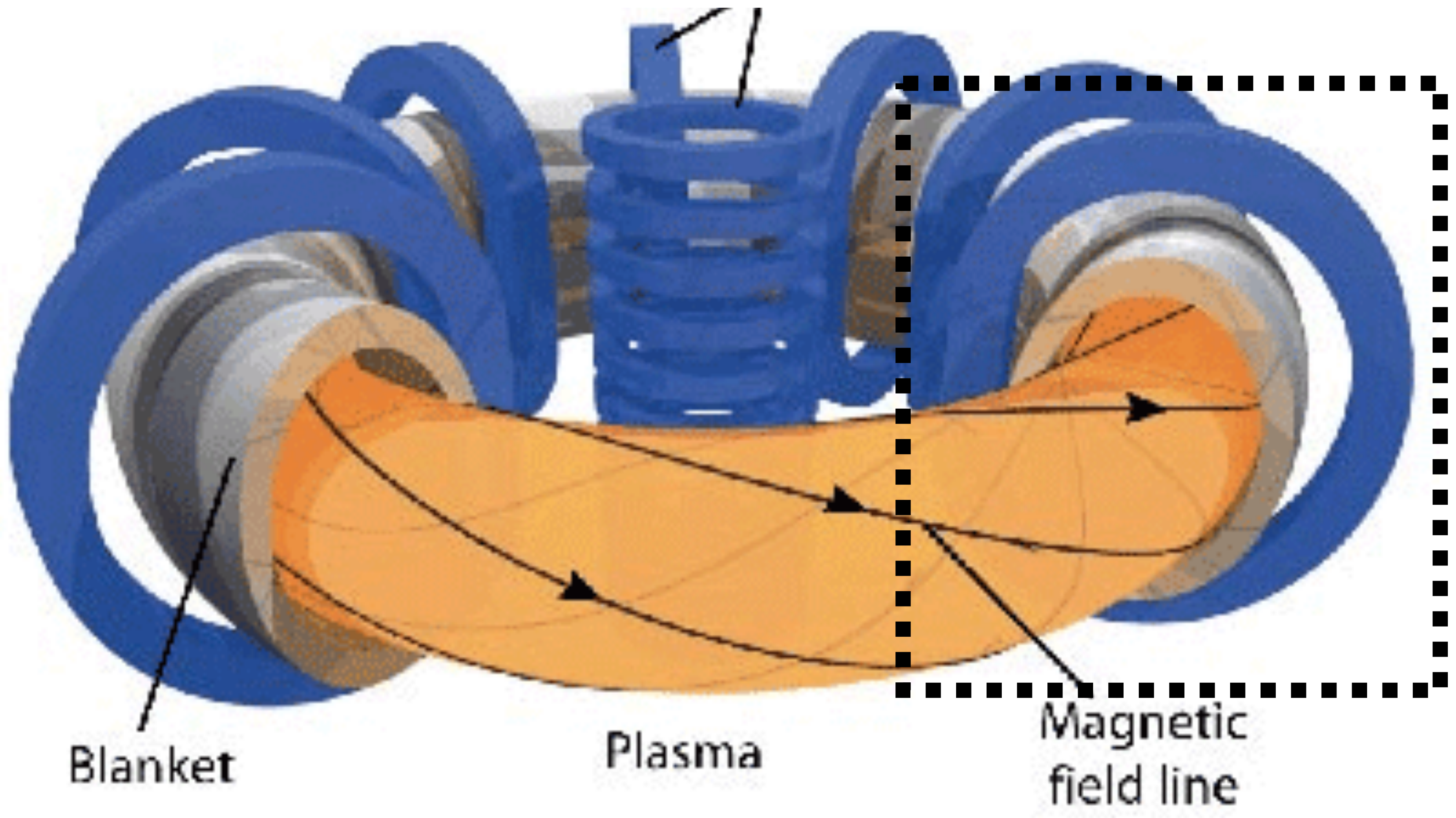
Figure adapted from Stoltzfus-Dueck (2009)

Fusion 101: magnetic confinement

Here on Earth, we have developed **magnetic confinement fusion**

- At the extreme temperatures necessary for fusion, the fuel becomes a plasma (charged gas)
- Plasmas can be confined by strong magnetic fields, allowing fuel (and heat) to be confined in “magnetic insulation”
- The most advanced designs are donut-shaped: tokamaks & stellarators
    - In the “core”, the magnetic field lines are “closed”, wrapping around the concentric surfaces of the donut
    - Temperature gradients between hot core and cool walls are more than 1000x steeper than gradients handled by reentry tiles on spacecraft





cross-section

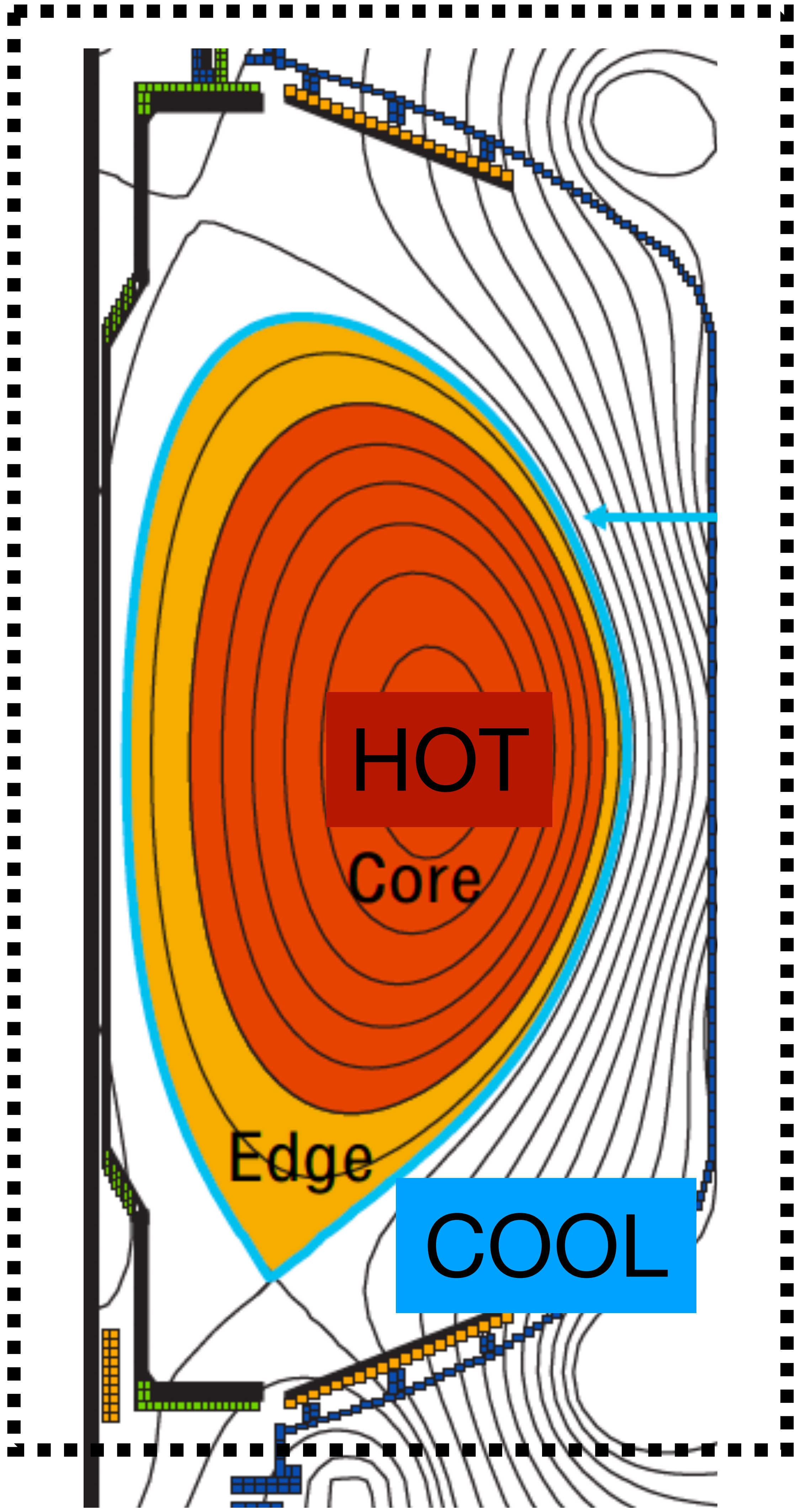
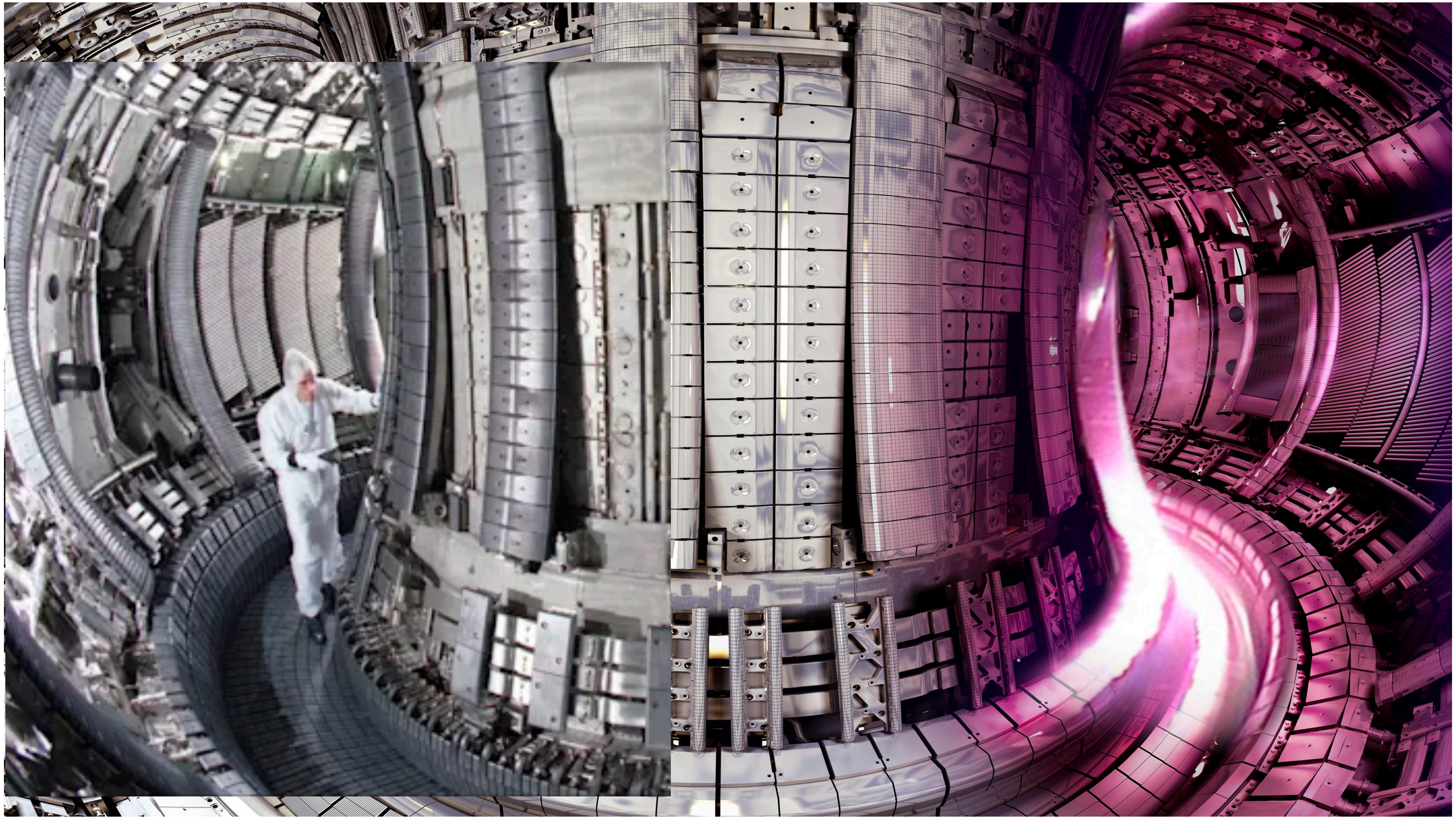


Figure adapted from
Stoltzfus-Dueck (2009)

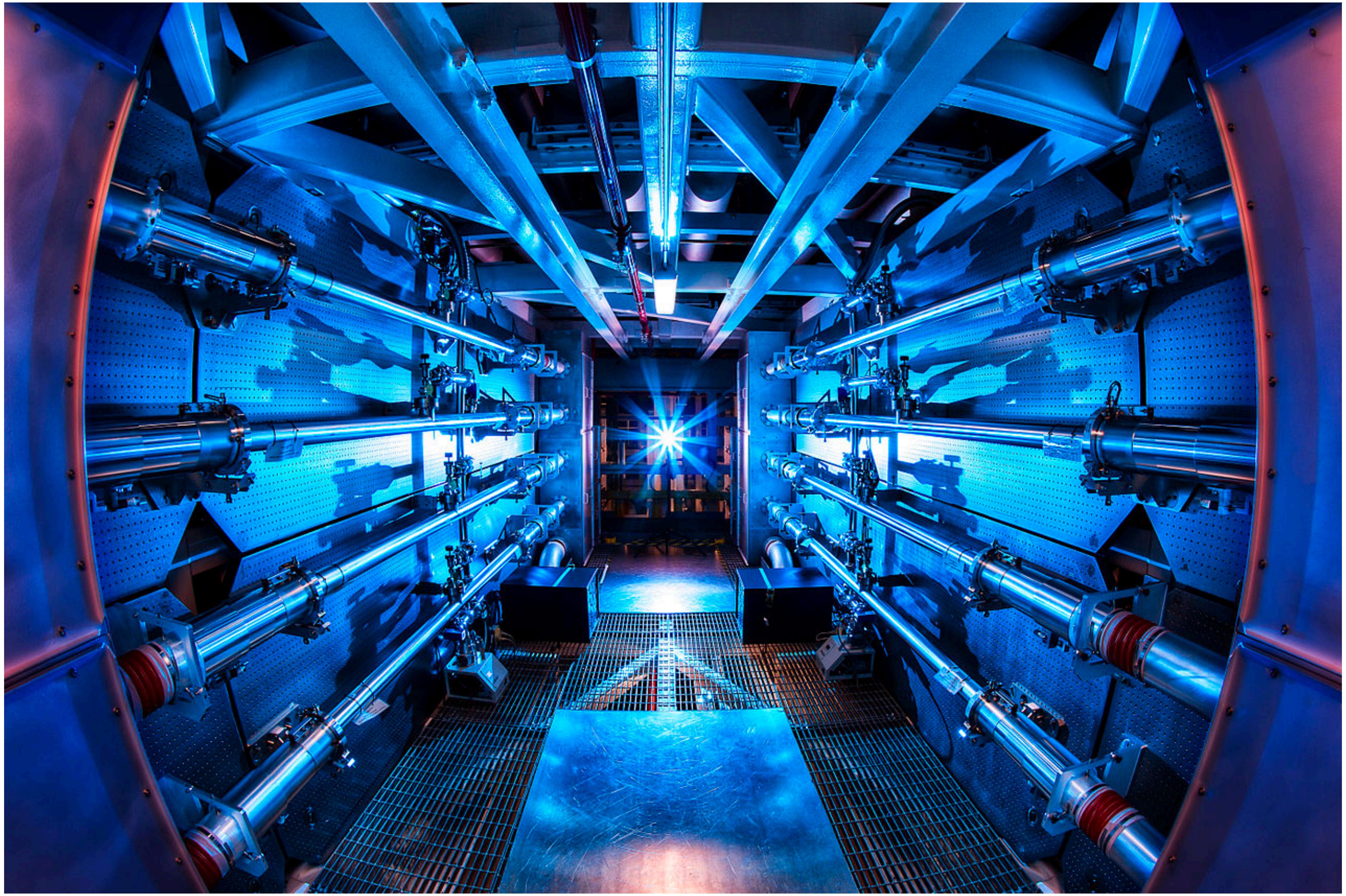
Exciting times in fusion: JET sets fusion power record

- The Joint European Torus (JET) tokamak recently set a record of 59 MJ of fusion power over 5 seconds (Q = 0.33)



Exciting times in fusion: NIF breaks even(?)

- National Ignition Facility (NIF) at LLNL uses inertial confinement fusion (shoots super-lasers at a tiny capsule to implode it)
- Produced 14 kJ of energy, more than the x-ray energy absorbed (break-even?), but still a small fraction of the 1.8 MJ laser energy



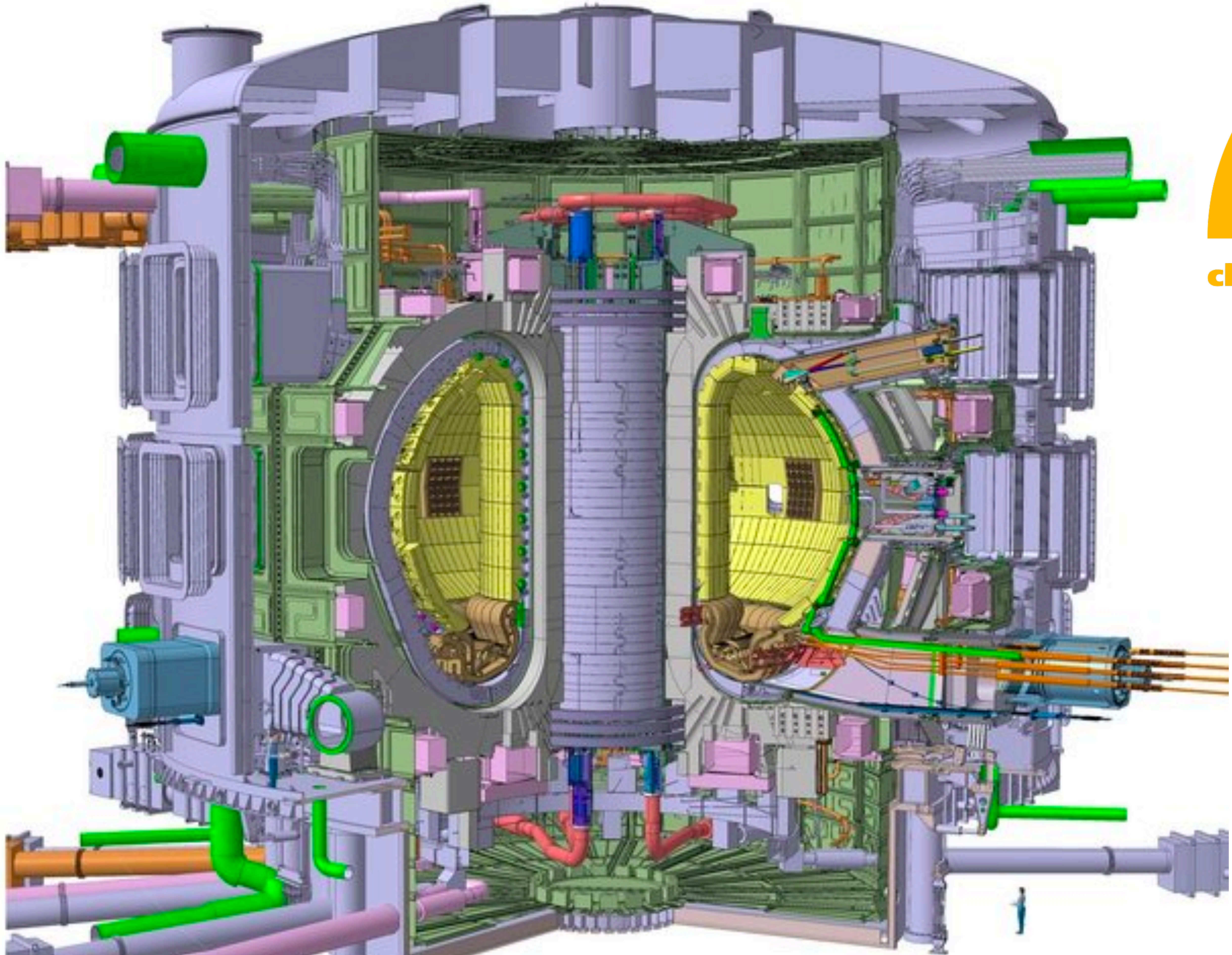
National Nuclear Security Administration

Exciting times in fusion: ITER on the way

- The ITER tokamak experiment will produce 500 MW of fusion power from 50 MW of heating power (Q = 10), with 400+ sec pulses
  - Under construction in France, projected to begin operation in 2025?, Q>1 in 2035. Massive international collaboration involving 35 nations. ~$25B

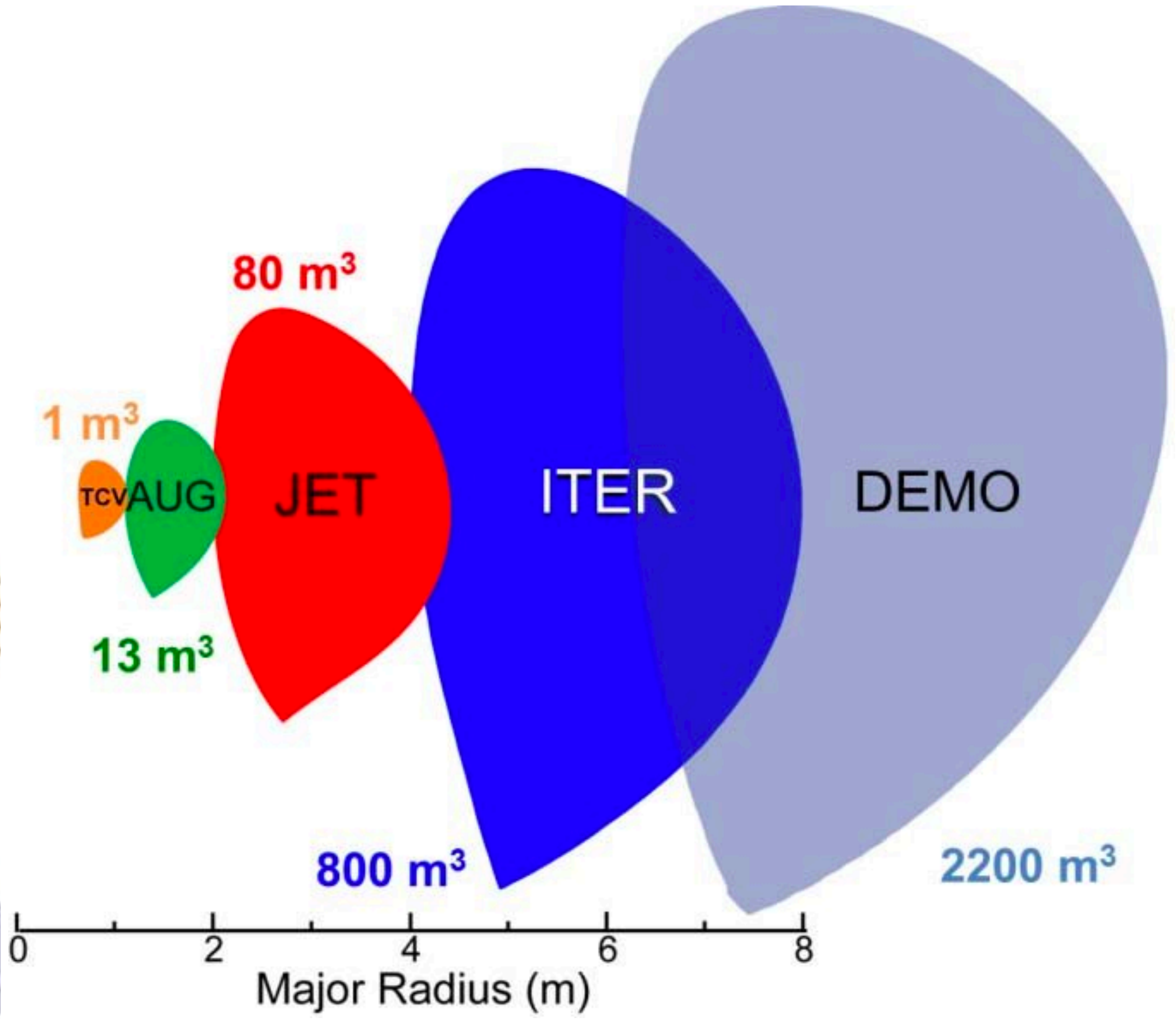
ITER = “The Way” in Latin

JET



ITER
(10x bigger by plasma volume than JET)

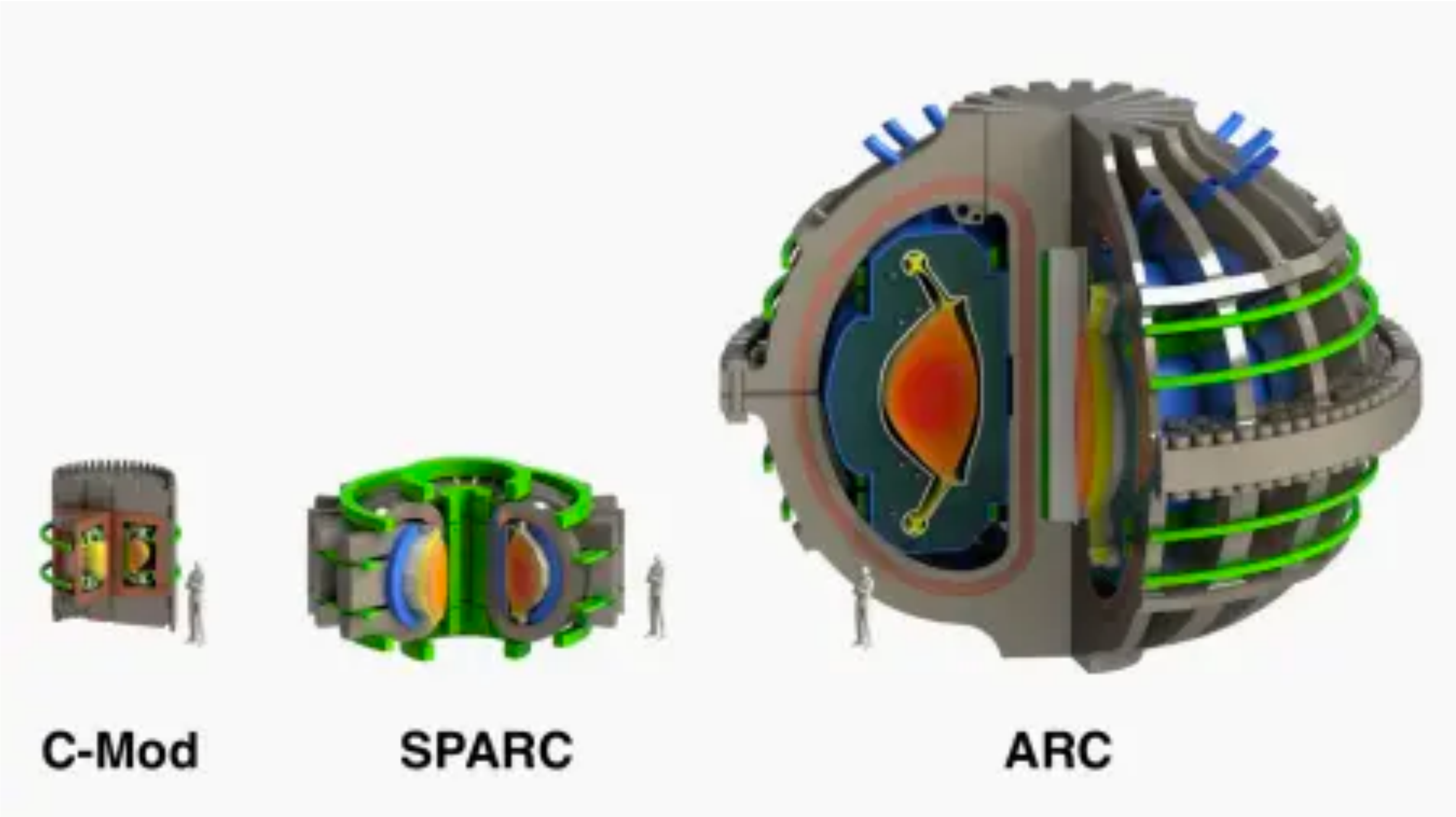
china eu india japan korea russia usa



Exciting times in fusion: SPARC/CFS go private

- MIT spin-off Commonwealth Fusion Systems (CFS) has begun building the SPARC tokamak
  - Stronger magnetic fields enabled by new high temperature superconducting magnets will give similar performance as ITER in a smaller device. Target is $Q>2$, could achieve $Q\sim10$.
    - Breakthrough: successful demonstration of 20T toroidal field magnet in late 2021.
  - Projected to begin operation in 2025, with $Q>1$ soon after.

Commonwealth Fusion Systems



Still work to do… fusion is still hard

Still work to do… fusion is still hard

- Need to achieve and maintain high pressures in the core of the reactor to reach self-sustaining “burning” conditions (10x hotter than the surface of the sun)

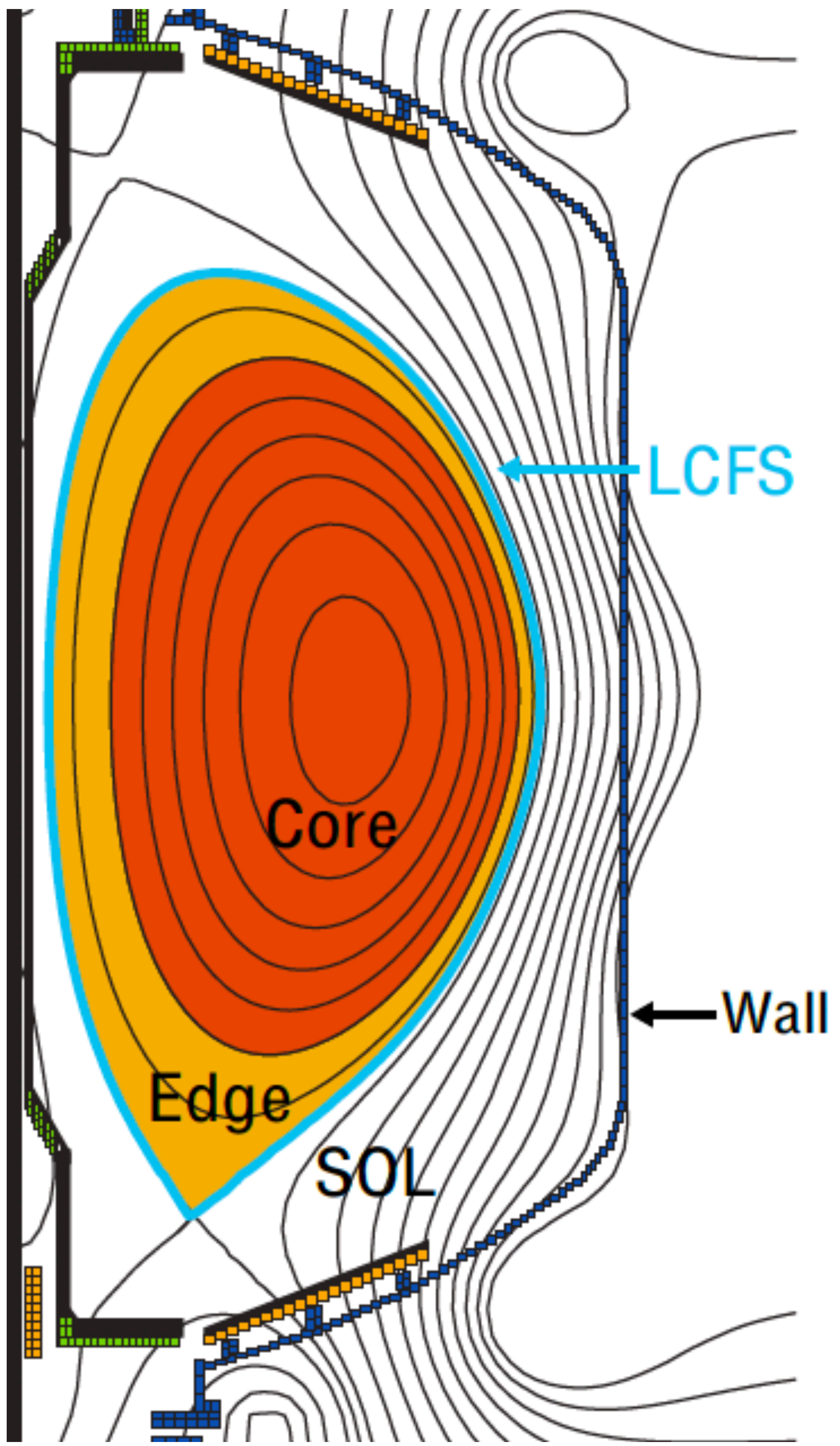


Figure adapted from Stoltzfus-Dueck (2009)

Still work to do… fusion is still hard

- Need to achieve and maintain high pressures in the core of the reactor to reach self-sustaining “burning” conditions (10x hotter than the surface of the sun)

- **Problem 1**: heat is lost too quickly from the core because of **turbulence**

    - Nature doesn’t like temperature gradients that steep!

    - Bigger reactors (e.g. ITER) or stronger magnetic fields (e.g. SPARC) can help

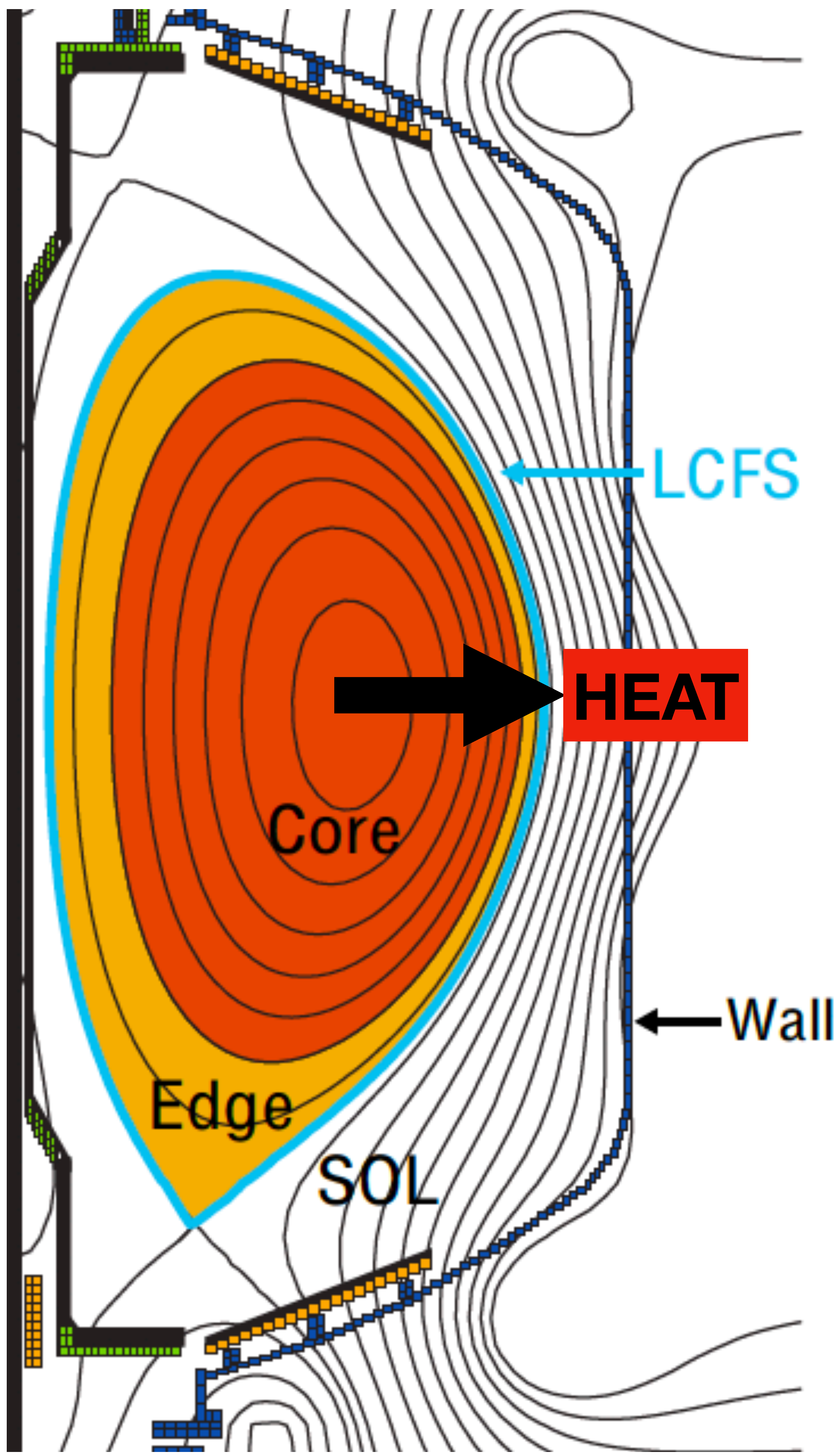


Figure adapted from Stoltzfus-Dueck (2009)

Still work to do… fusion is still hard

- Need to achieve and maintain high pressures in the core of the reactor to reach self-sustaining “burning” conditions (10x hotter than the surface of the sun)
- **Problem 1**: heat is lost too quickly from the core because of **turbulence**
  - Nature doesn’t like temperature gradients that steep!
  - Bigger reactors (e.g. ITER) or stronger magnetic fields (e.g. SPARC) can help
- Heat is exhausted from the core and handled in the boundary region, where the field lines are “open” and intersect the walls
- **Problem 2**: if the boundary plasma is too hot, the **heat exhaust** will be dangerous to the device walls

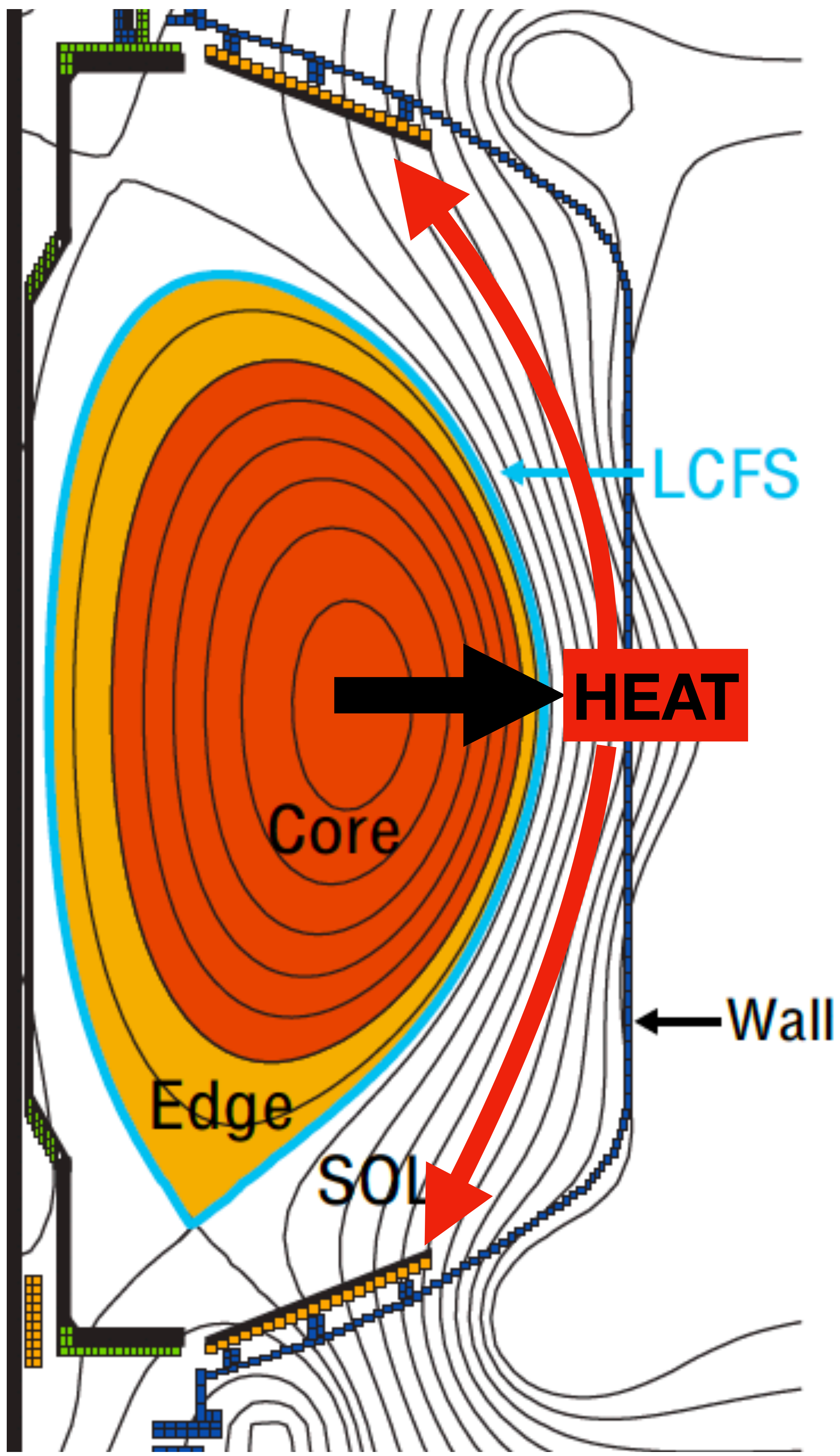


Figure adapted from Stoltzfus-Dueck (2009)

Still work to do… fusion is still hard

- Need to achieve and maintain high pressures in the core of the reactor to reach self-sustaining “burning” conditions (10x hotter than the surface of the sun)
- **Problem 1**: heat is lost too quickly from the core because of **turbulence**
  - Nature doesn’t like temperature gradients that steep!
  - Bigger reactors (e.g. ITER) or stronger magnetic fields (e.g. SPARC) can help
- Heat is exhausted from the core and handled in the boundary region, where the field lines are “open” and intersect the walls
- **Problem 2**: if the boundary plasma is too hot, the **heat exhaust** will be dangerous to the device walls
- Need a simultaneous **whole-device** solution to these coupled problems

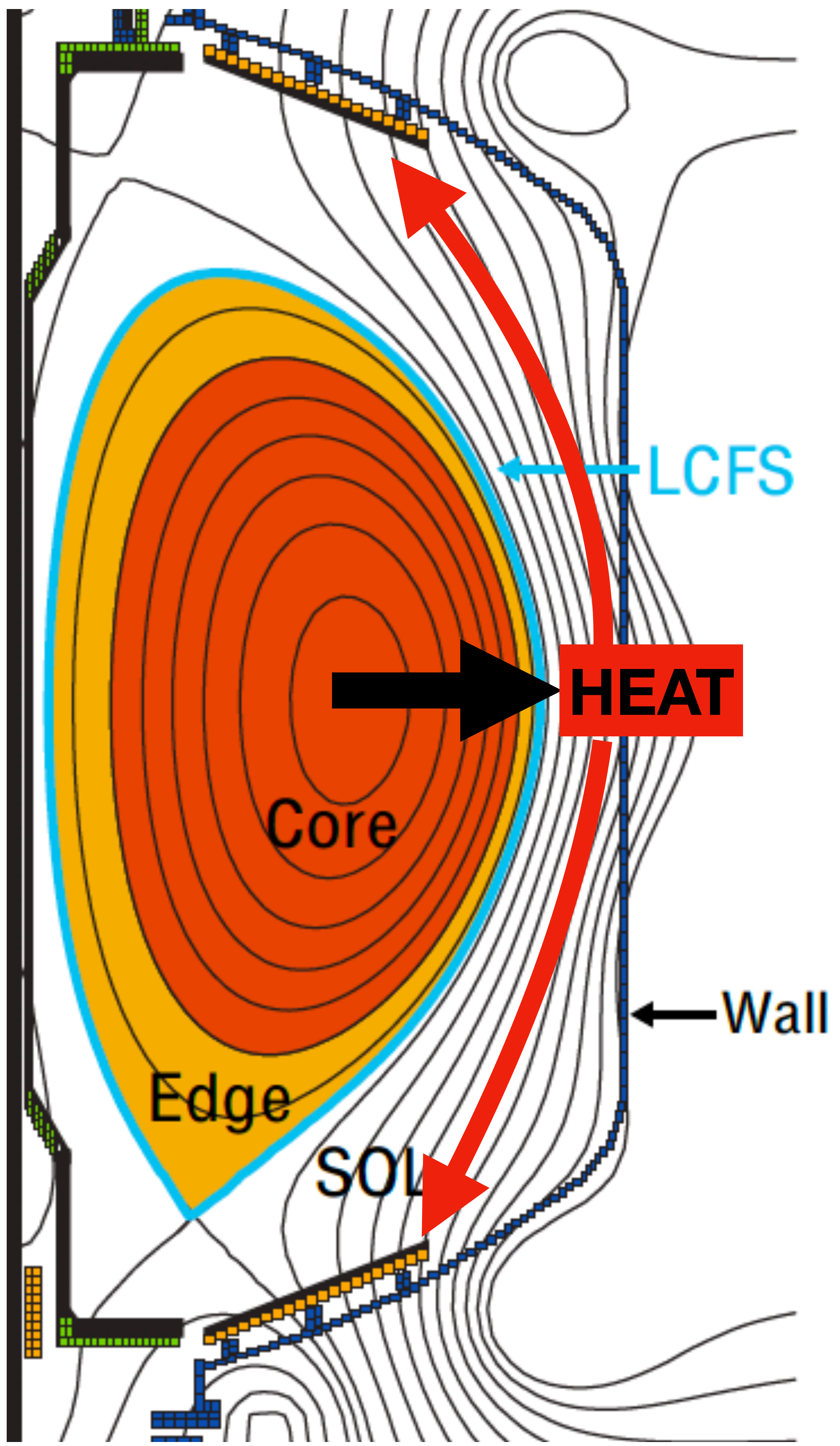
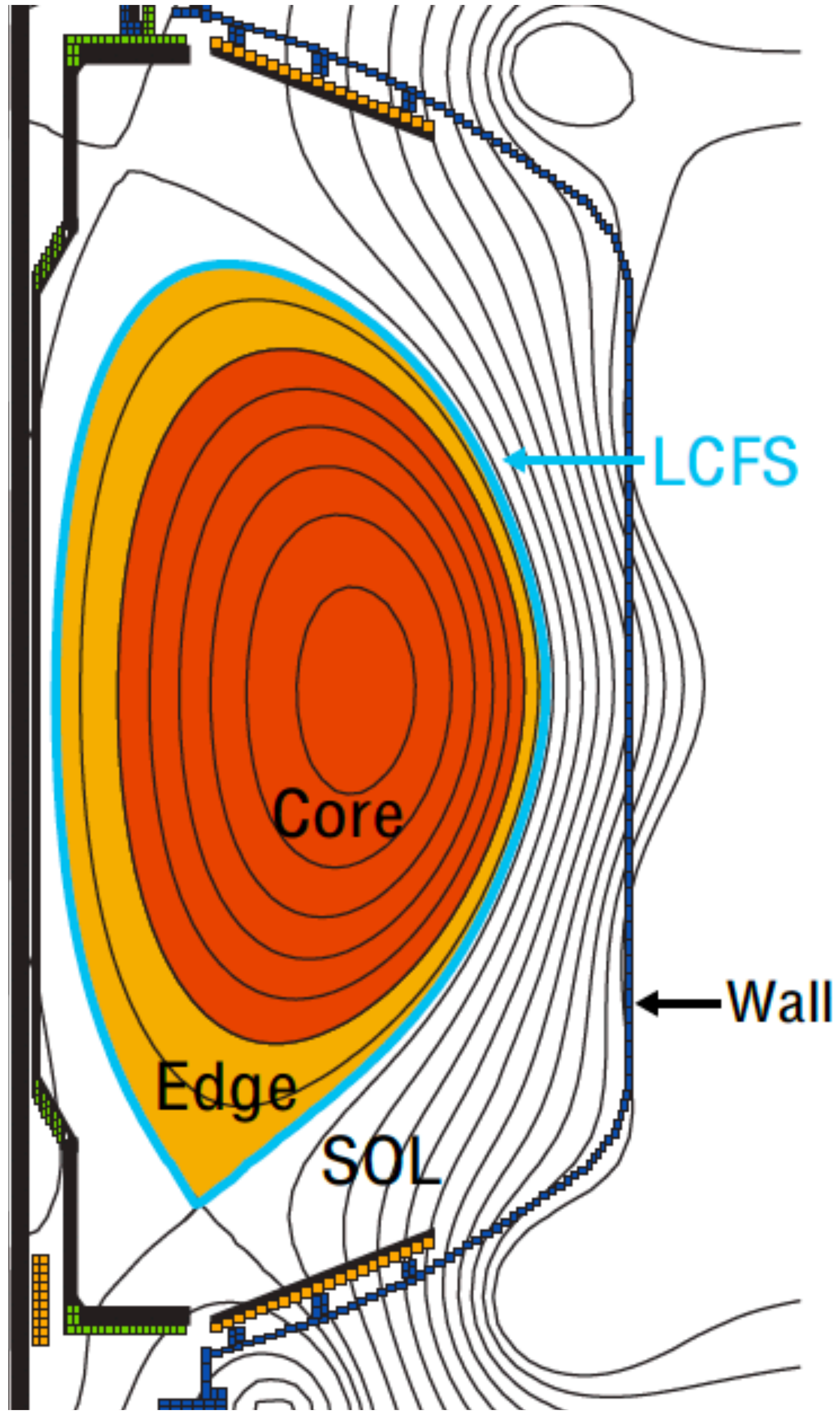


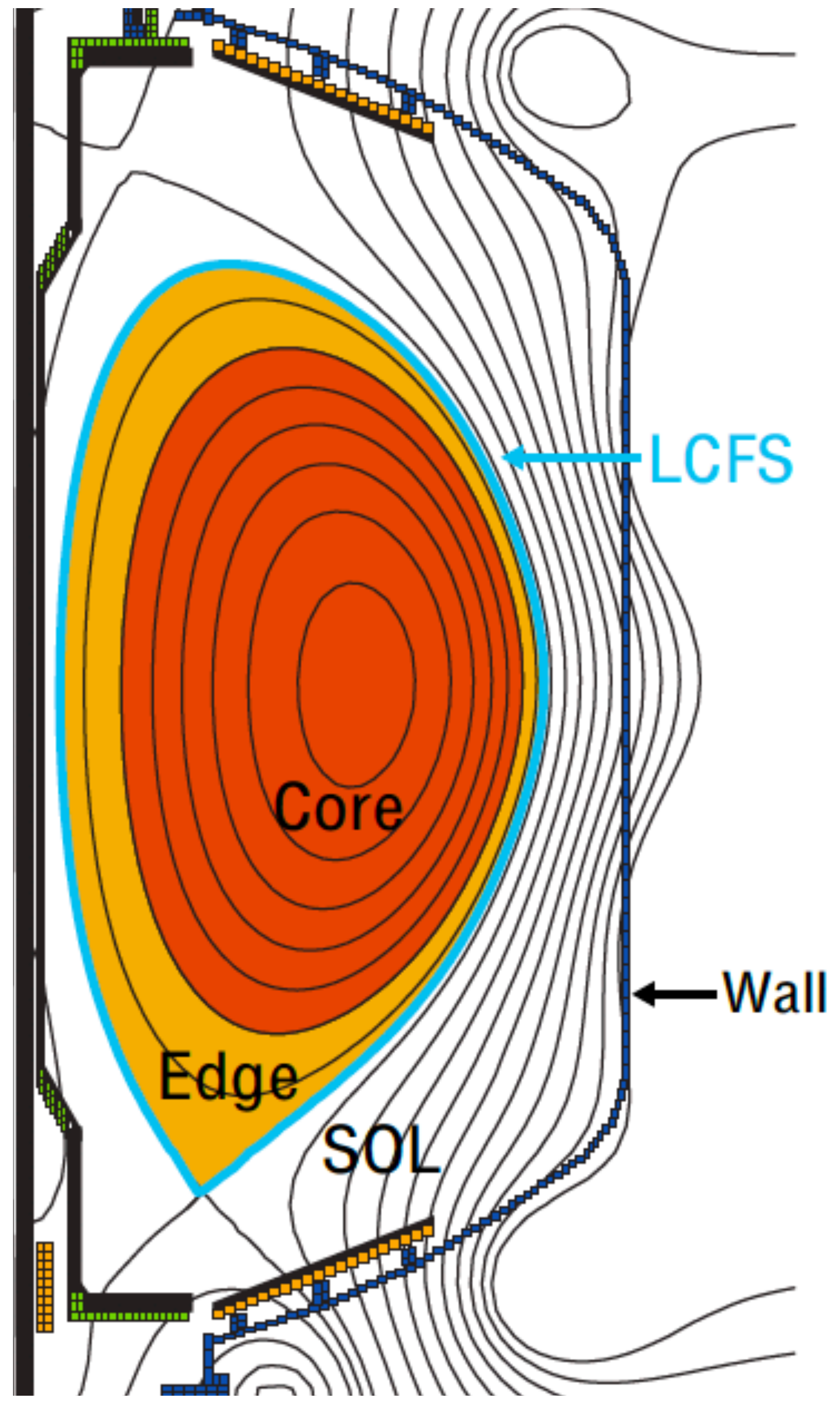
Figure adapted from Stoltzfus-Dueck (2009)

Whole-device modeling and optimization



Whole-device modeling and optimization

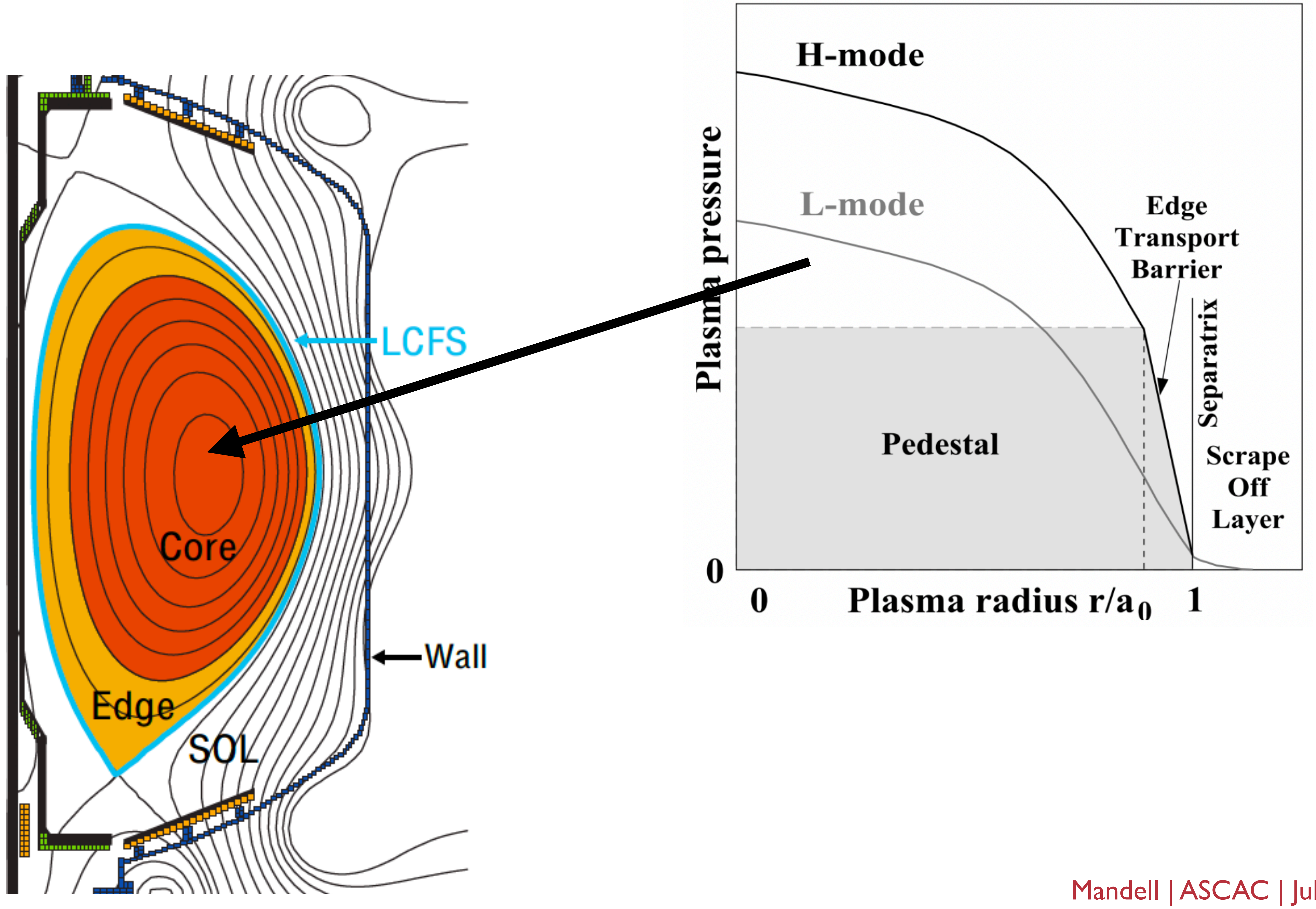
- **Whole-device turbulent transport modeling and optimization** are areas where **exascale high-performance computing** can make a large impact on the success of fusion by enabling the design of more efficient fusion reactors
- For a given reactor design, need to be able to model/predict:



Whole-device modeling and optimization

- **Whole-device turbulent transport modeling and optimization** are areas where **exascale high-performance computing** can make a large impact on the success of fusion by enabling the design of more efficient fusion reactors

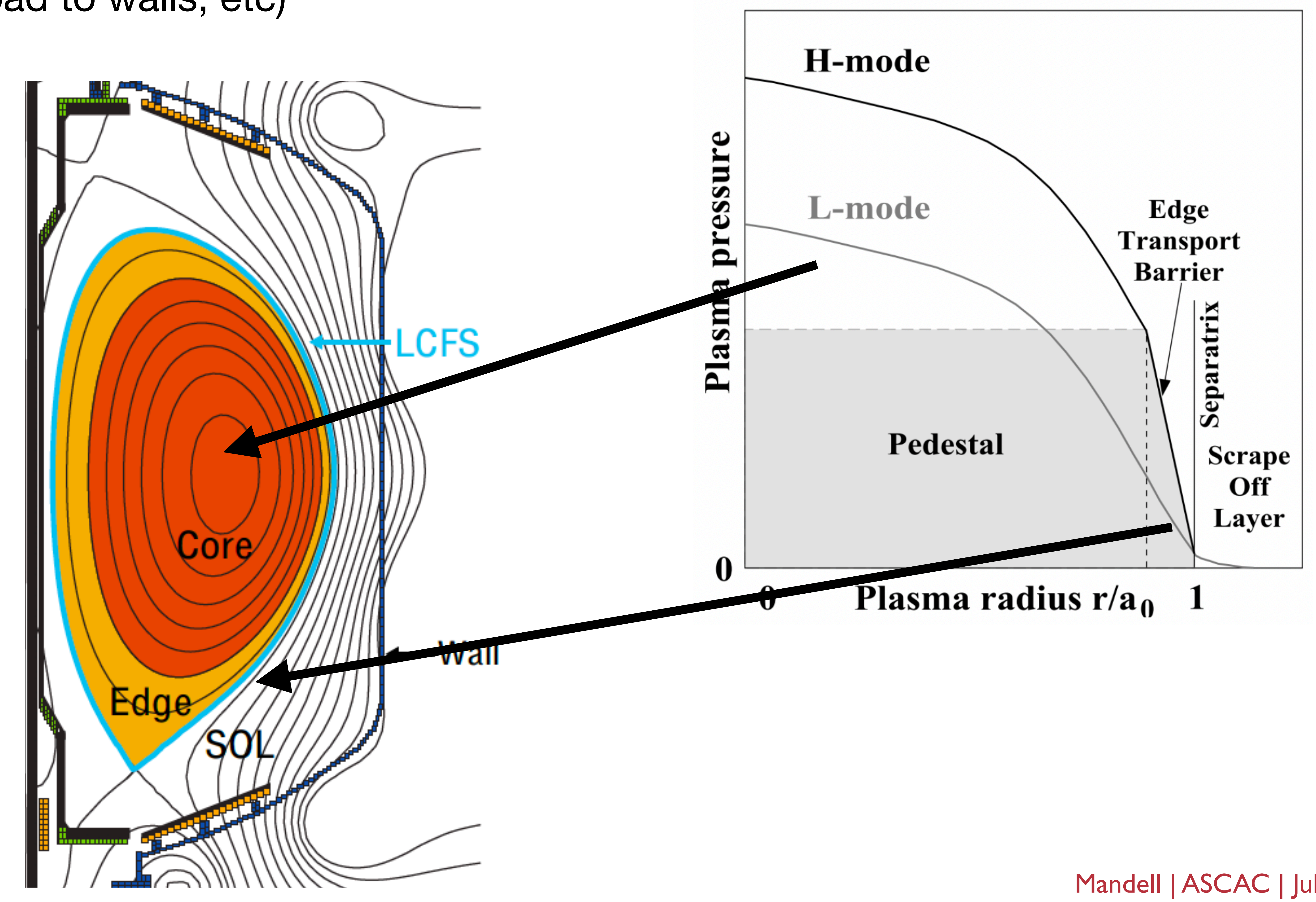
- For a given reactor design, need to be able to model/predict:
  - Fusion power, which is determined by macroscopic profiles of density and temperature in the core



Whole-device modeling and optimization

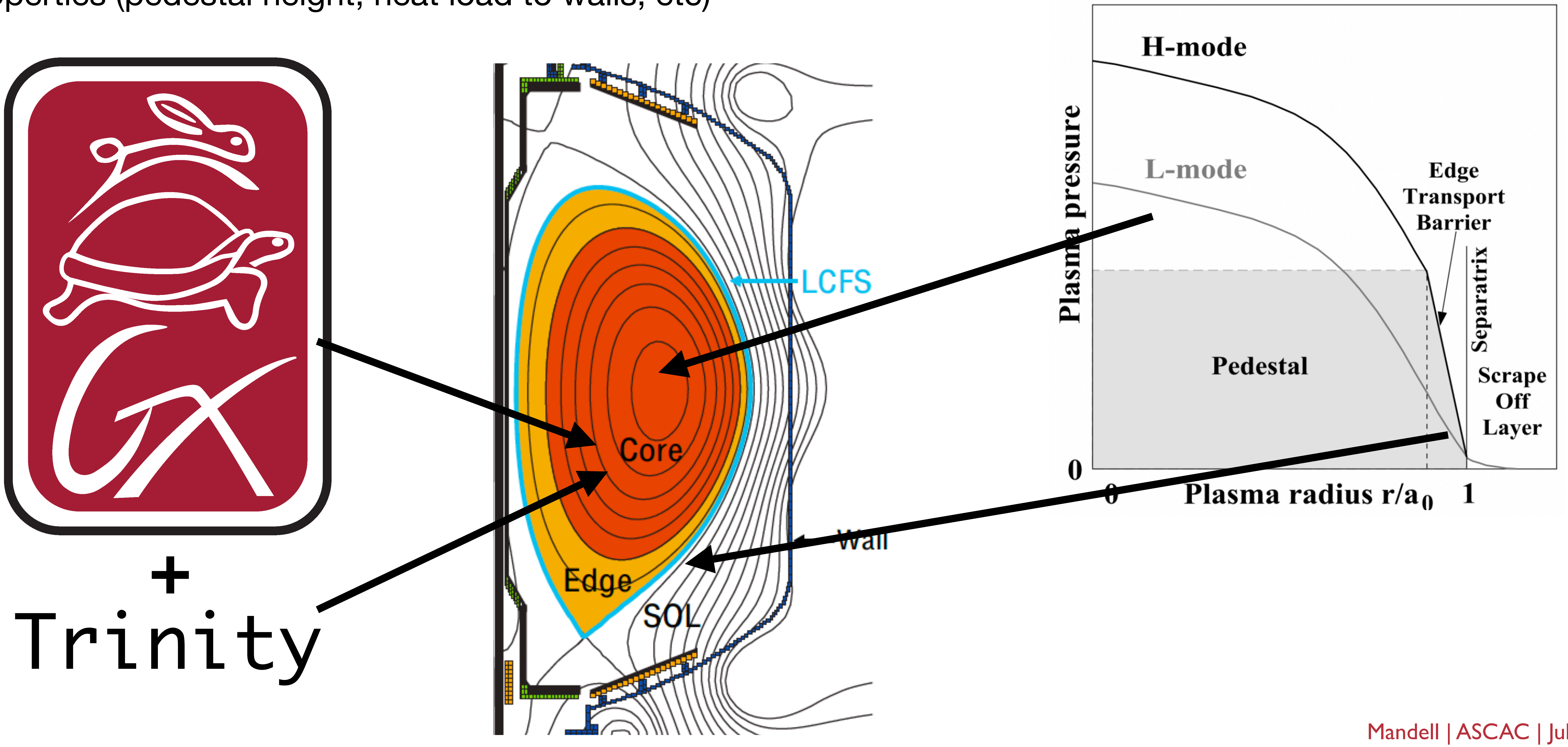
- **Whole-device turbulent transport modeling and optimization** are areas where **exascale high-performance computing** can make a large impact on the success of fusion by enabling the design of more efficient fusion reactors

- For a given reactor design, need to be able to model/predict:
  - Fusion power, which is determined by macroscopic profiles of density and temperature in the core
  - Boundary properties (pedestal height, heat load to walls, etc)



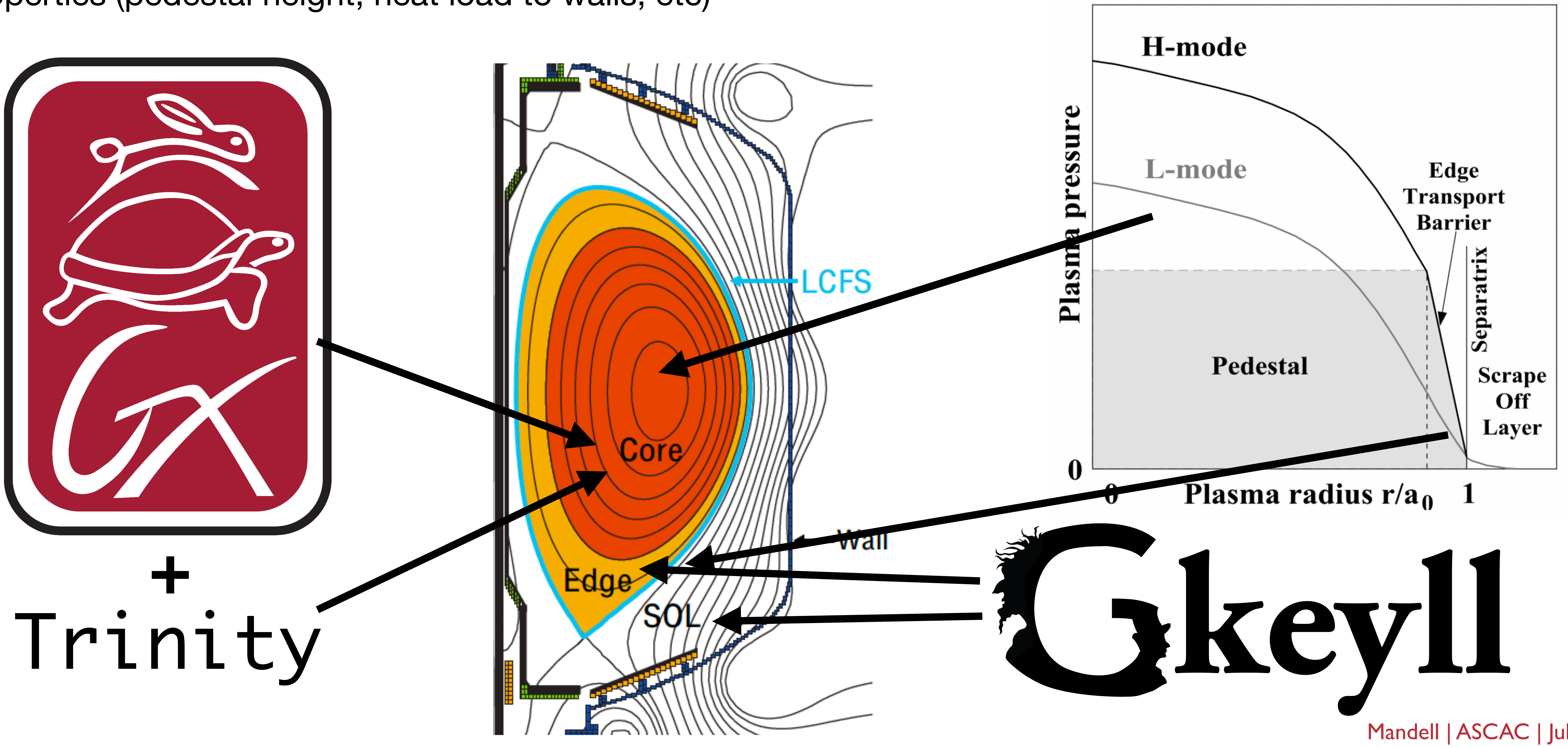
Whole-device modeling and optimization

- **Whole-device turbulent transport modeling and optimization** are areas where **exascale high-performance computing** can make a large impact on the success of fusion by enabling the design of more efficient fusion reactors
- For a given reactor design, need to be able to model/predict:
    - Fusion power, which is determined by macroscopic profiles of density and temperature in the core
    - Boundary properties (pedestal height, heat load to walls, etc)

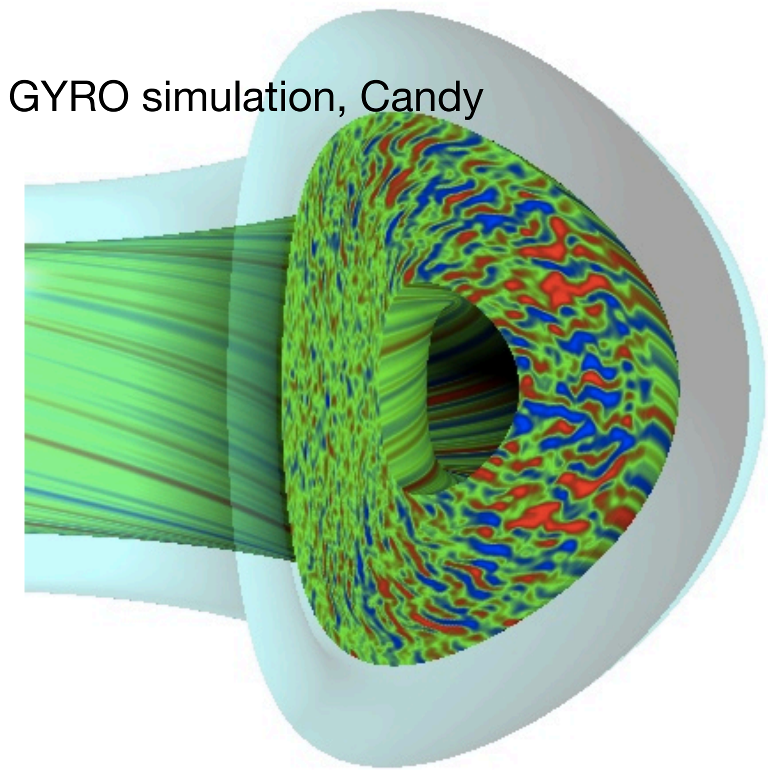


Whole-device modeling and optimization

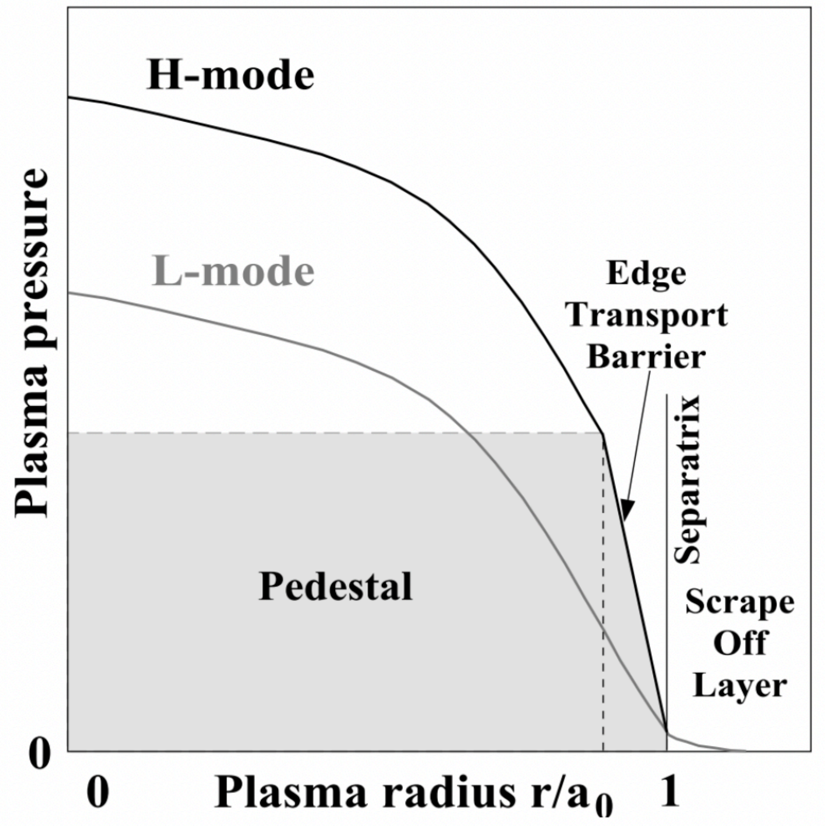
- **Whole-device turbulent transport modeling and optimization** are areas where **exascale high-performance computing** can make a large impact on the success of fusion by enabling the design of more efficient fusion reactors
- For a given reactor design, need to be able to model/predict:
  - Fusion power, which is determined by macroscopic profiles of density and temperature in the core
  - Boundary properties (pedestal height, heat load to walls, etc)



Fusion is hard; *modeling* a fusion plasma is hard, too



small-scale turbulence

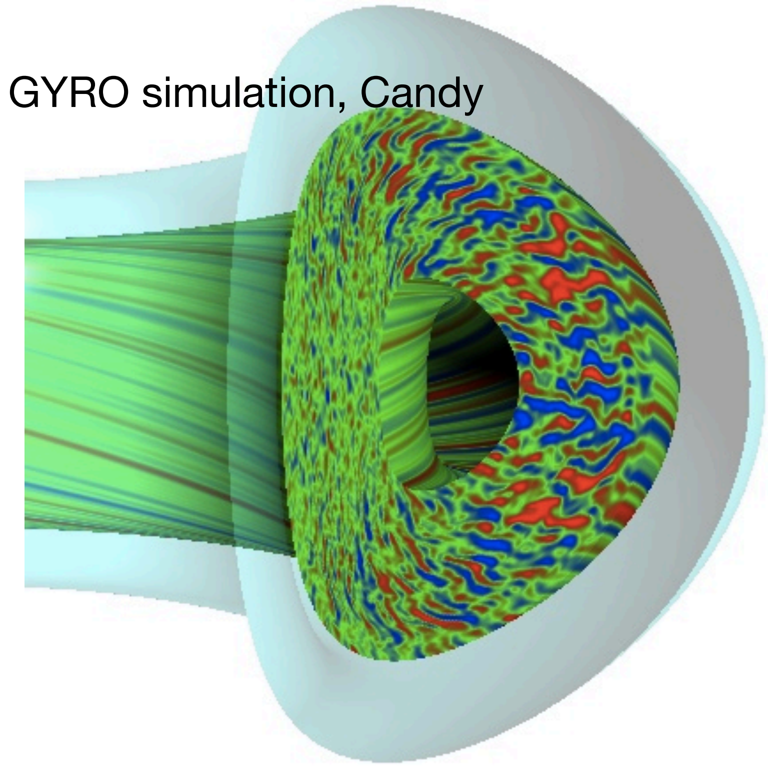


slowly-varying background profiles

Fusion is hard; *modeling* a fusion plasma is hard, too

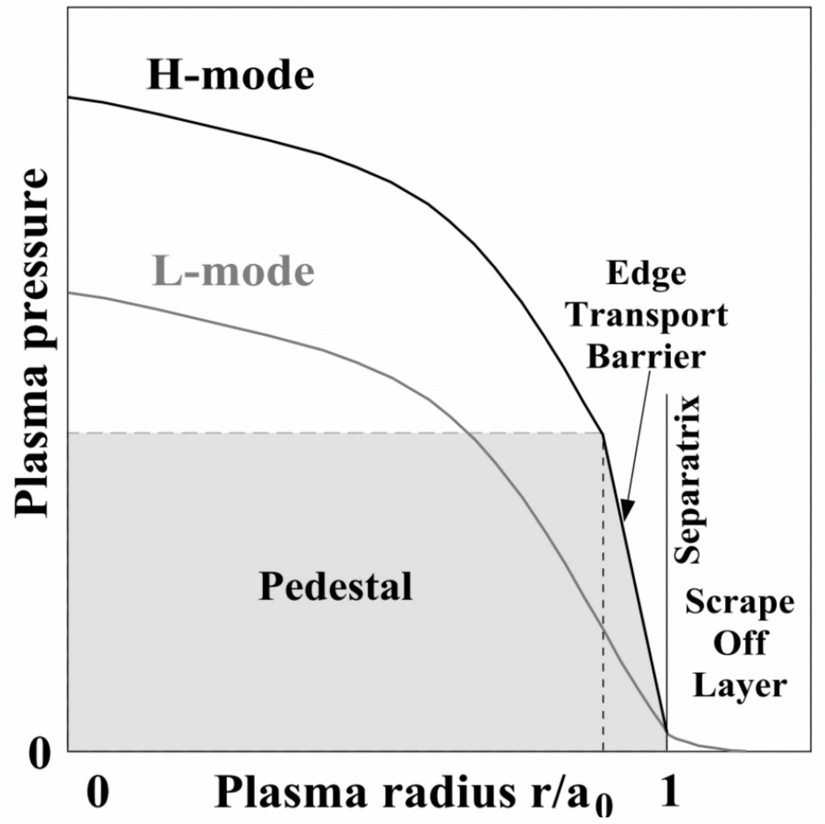
- A fusion reactor has an **enormous range of scales**. In the core, we have **turbulence at micro-scales** and background **profile evolution (transport) at macro-scales**, separated by several orders of magnitude

GYRO simulation, Candy



$$L_{\text{turb}} \sim \rho_i \sim 0.1 - 0.5 \text{ mm} \xleftrightarrow{\sim 10^3 - 10^4} L_{\text{transp}} \sim a \gtrsim 1 \text{ m}$$

$$\tau_{\text{turb}} \sim a/v_{\text{ti}} \sim 10\ \mu\text{s} \xleftrightarrow[\sim 10^5]{} \tau_{\text{transp}} \sim \tau_E \sim 1 \text{ s}$$



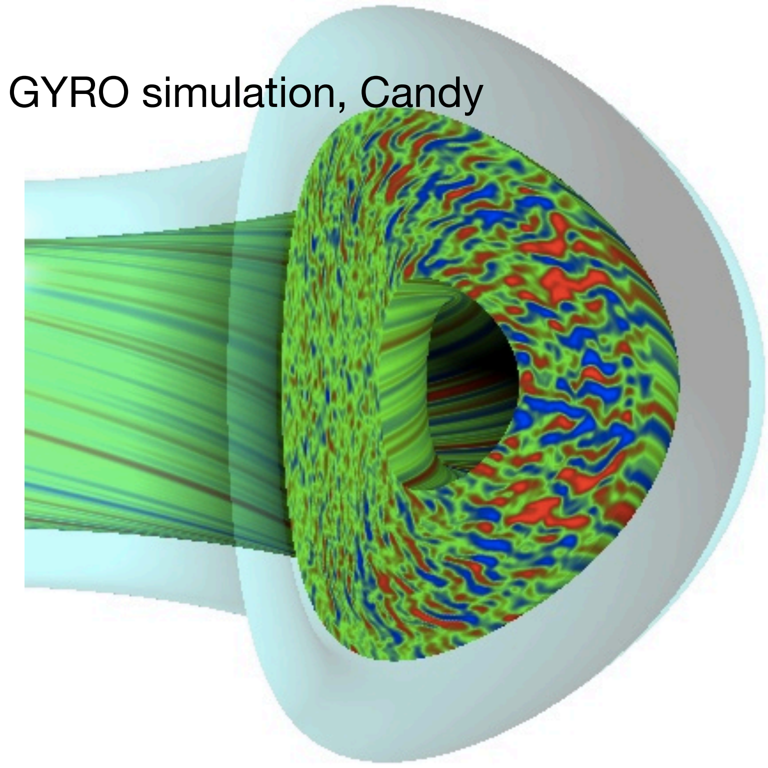
small-scale turbulence

slowly-varying background profiles

Fusion is hard; *modeling* a fusion plasma is hard, too

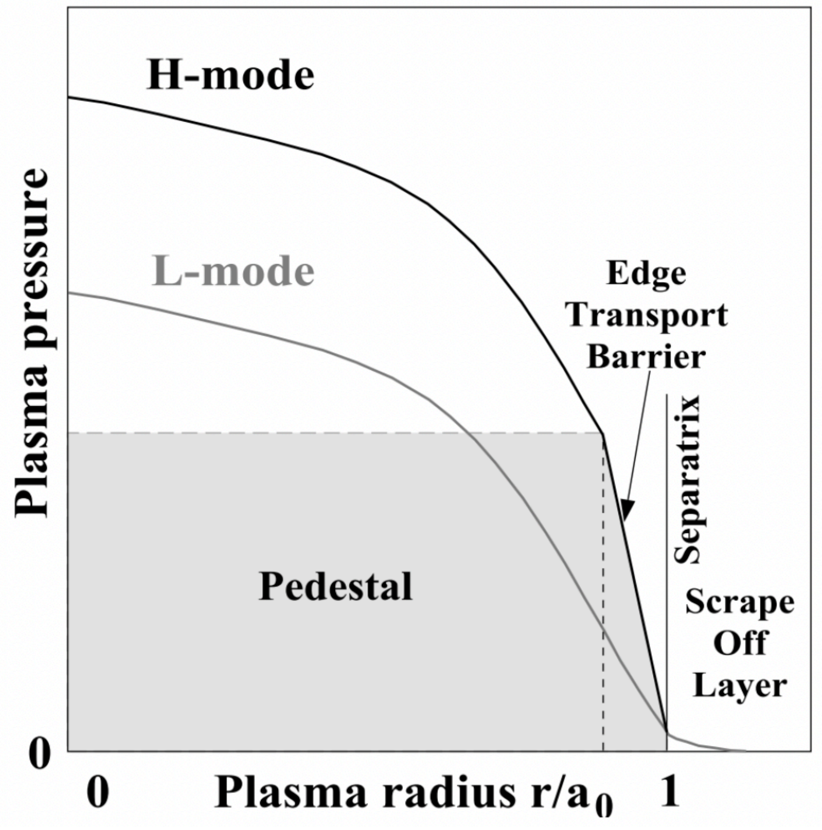
- A fusion reactor has an **enormous range of scales**. In the core, we have **turbulence at micro-scales** and background **profile evolution (transport) at macro-scales**, separated by several orders of magnitude

GYRO simulation, Candy



$$L_\text{turb} \sim \rho_i \sim 0.1 - 0.5 \text{ mm} \quad \xleftrightarrow{\sim 10^3 - 10^4} \quad L_\text{transp} \sim a \gtrsim 1 \text{ m}$$

$$\tau_\text{turb} \sim a/v_\text{ti} \sim 10\ \mu\text{s} \quad \xleftrightarrow[\sim 10^5]{} \quad \tau_\text{transp} \sim \tau_E \sim 1 \text{ s}$$



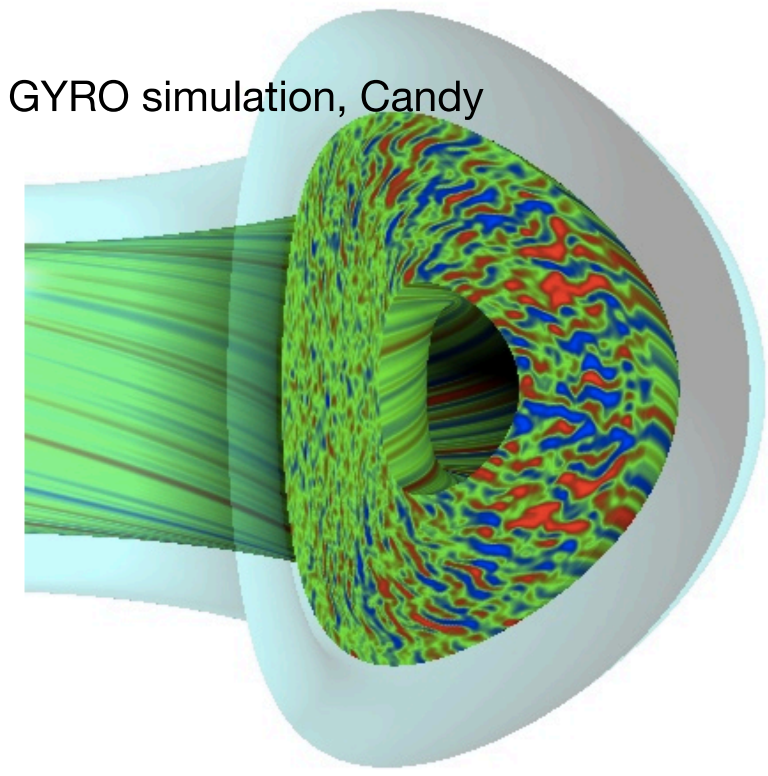
small-scale turbulence

slowly-varying background profiles

- Extreme temperatures require a **kinetic** description (instead of fluid), which means solving a PDE in 5D (3x+2v, after averaging over fast gyromotion to get **gyrokinetics**)

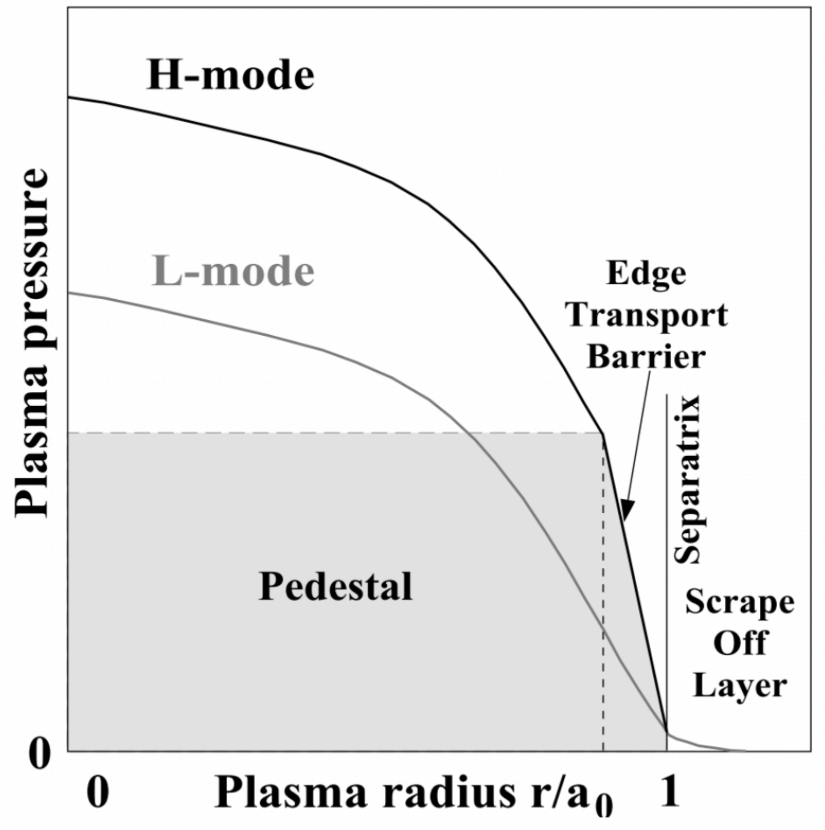
Fusion is hard; *modeling* a fusion plasma is hard, too

- A fusion reactor has an **enormous range of scales**. In the core, we have **turbulence at micro-scales** and background **profile evolution (transport) at macro-scales**, separated by several orders of magnitude


GYRO simulation, Candy

$L_{\text{turb}} \sim \rho_i \sim 0.1 - 0.5\ \text{mm} \longleftarrow \sim 10^3 - 10^4 \longrightarrow L_{\text{transp}} \sim a \gtrsim 1\ \text{m}$

$\tau_{\text{turb}} \sim a/v_{\text{ti}} \sim 10\ \mu\text{s} \longleftarrow \sim 10^5 \longrightarrow \tau_{\text{transp}} \sim \tau_E \sim 1\ \text{s}$



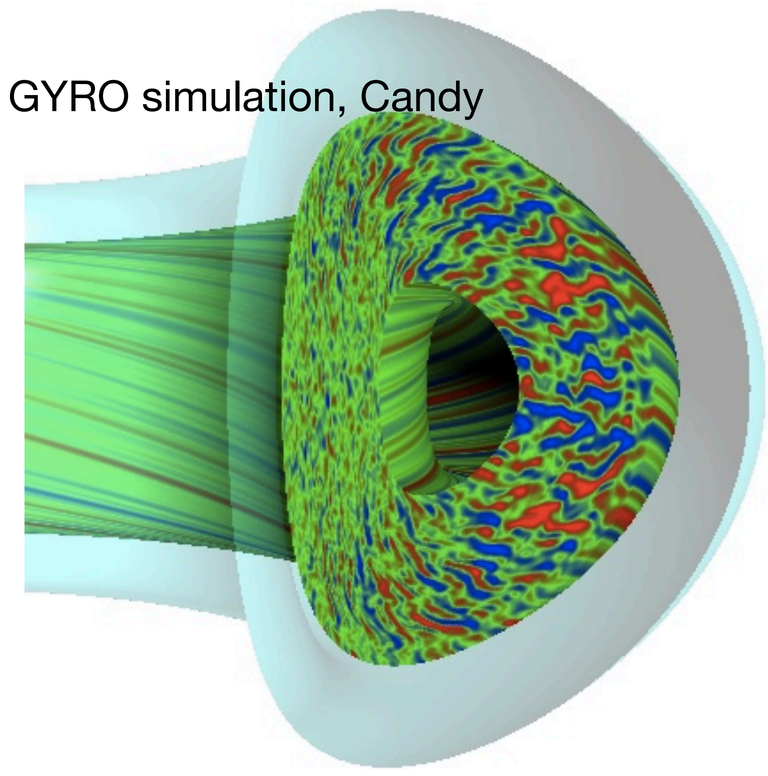
small-scale turbulence **slowly-varying background profiles**

- Extreme temperatures require a **kinetic** description (instead of fluid), which means solving a PDE in 5D (3x+2v, after averaging over fast gyromotion to get **gyrokinetics**)
- Modeling this range of scales with a brute-force 5D+time discretization would then require

$$N \sim \left(\frac{L_{\text{turb}}}{L_{\text{transp}}}\right)^3 (N_v \sim 10)^2 \sim 10^{14} \text{ phase-space grid points in 5D}$$

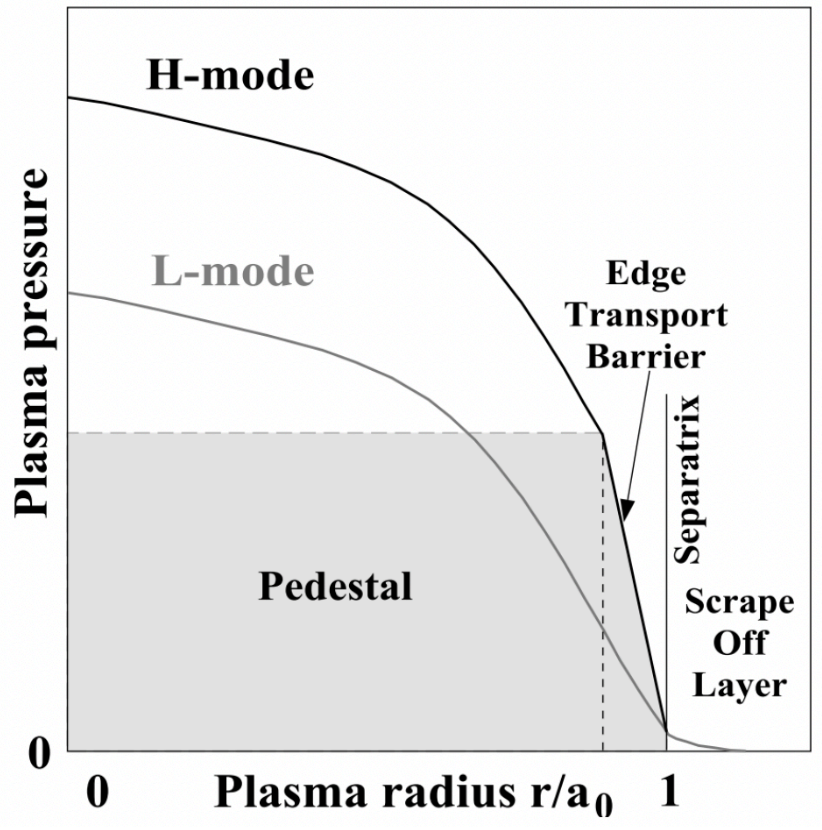
Fusion is hard; *modeling* a fusion plasma is hard, too

- A fusion reactor has an **enormous range of scales**. In the core, we have **turbulence at micro-scales** and background **profile evolution (transport) at macro-scales**, separated by several orders of magnitude



$L_{\mathrm{turb}} \sim \rho_i \sim 0.1 - 0.5\ \mathrm{mm}$ $\xleftrightarrow{\sim 10^3 - 10^4}$ $L_{\mathrm{transp}} \sim a \gtrsim 1\ \mathrm{m}$

$\tau_{\mathrm{turb}} \sim a/v_{\mathrm{ti}} \sim 10\ \mu\mathrm{s}$ $\xleftrightarrow[\sim 10^5]{}$ $\tau_{\mathrm{transp}} \sim \tau_E \sim 1\ \mathrm{s}$



small-scale turbulence **slowly-varying background profiles**

- Extreme temperatures require a **kinetic** description (instead of fluid), which means solving a PDE in 5D (3x+2v, after averaging over fast gyromotion to get **gyrokinetics**)

- Modeling this range of scales with a brute-force 5D+time discretization would then require

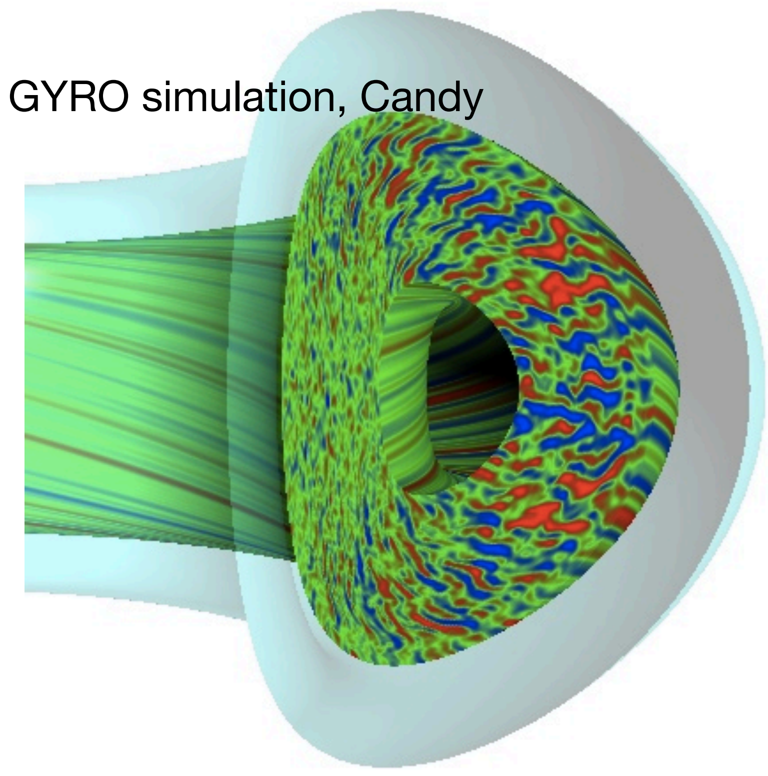
$$N \sim \left(\frac{L_{\mathrm{turb}}}{L_{\mathrm{transp}}}\right)^3 \left(N_v \sim 10\right)^2 \sim 10^{14} \text{ phase-space grid points in 5D}$$

$$\mathrm{NFLOP} \sim \mathcal{O}(N^2)\left(\frac{\tau_{\mathrm{turb}}}{\tau_{\mathrm{transp}}}\right) \sim 10^{33}\text{, which would require } 10^{15}\ \mathrm{s} \text{ on an exascale computer}$$

Fusion is hard; *modeling* a fusion plasma is hard, too

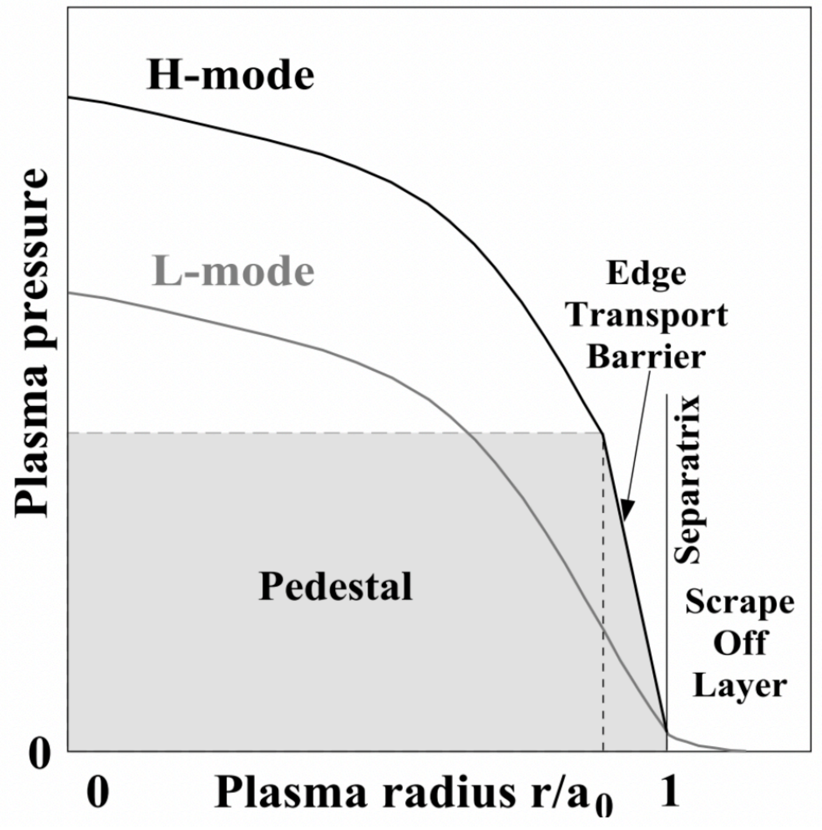
- A fusion reactor has an **enormous range of scales**. In the core, we have **turbulence at micro-scales** and background **profile evolution (transport) at macro-scales**, separated by several orders of magnitude

GYRO simulation, Candy



$$L_\mathrm{turb} \sim \rho_i \sim 0.1 - 0.5\ \mathrm{mm} \xleftrightarrow{\sim 10^3 - 10^4} L_\mathrm{transp} \sim a \gtrsim 1\ \mathrm{m}$$

$$\tau_\mathrm{turb} \sim a/v_\mathrm{ti} \sim 10\ \mu\mathrm{s} \xleftrightarrow[\sim 10^5]{} \tau_\mathrm{transp} \sim \tau_E \sim 1\ \mathrm{s}$$



small-scale turbulence **slowly-varying background profiles**

- Extreme temperatures require a **kinetic** description (instead of fluid), which means solving a PDE in 5D (3x+2v, after averaging over fast gyromotion to get **gyrokinetics**)
- Modeling this range of scales with a brute-force 5D+time discretization would then require

$$N \sim \left(\frac{L_\mathrm{turb}}{L_\mathrm{transp}}\right)^3 (N_v \sim 10)^2 \sim 10^{14} \text{ phase-space grid points in 5D}$$

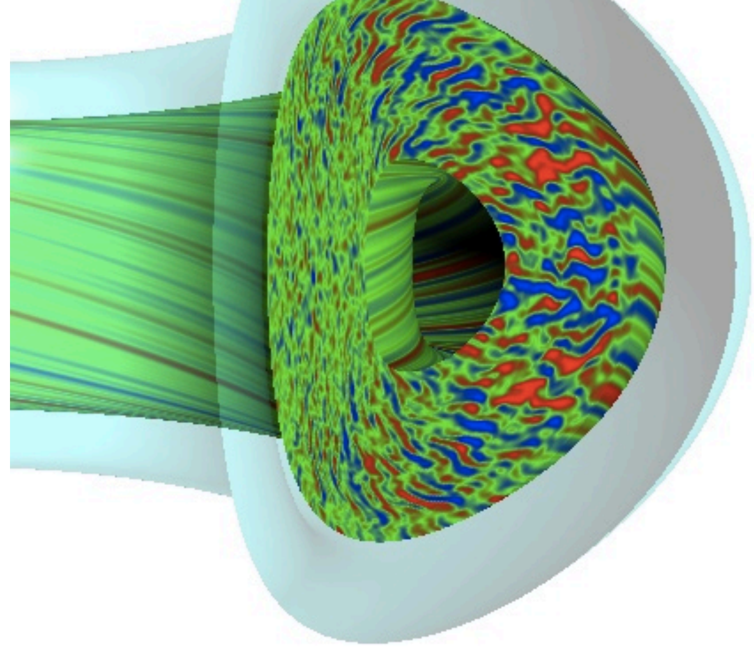
$$\mathrm{NFLOP} \sim \mathcal{O}(N^2)\left(\frac{\tau_\mathrm{turb}}{\tau_\mathrm{transp}}\right) \sim 10^{33}, \text{ which would require } 10^{15}\ \mathrm{s} \text{ on an exascale computer}$$

- These problems require **intense computational resources**, but also **clever multi-scale numerical algorithms and theory** to enable tractable calculations

Fusion is hard; *modeling* a fusion plasma is hard, too

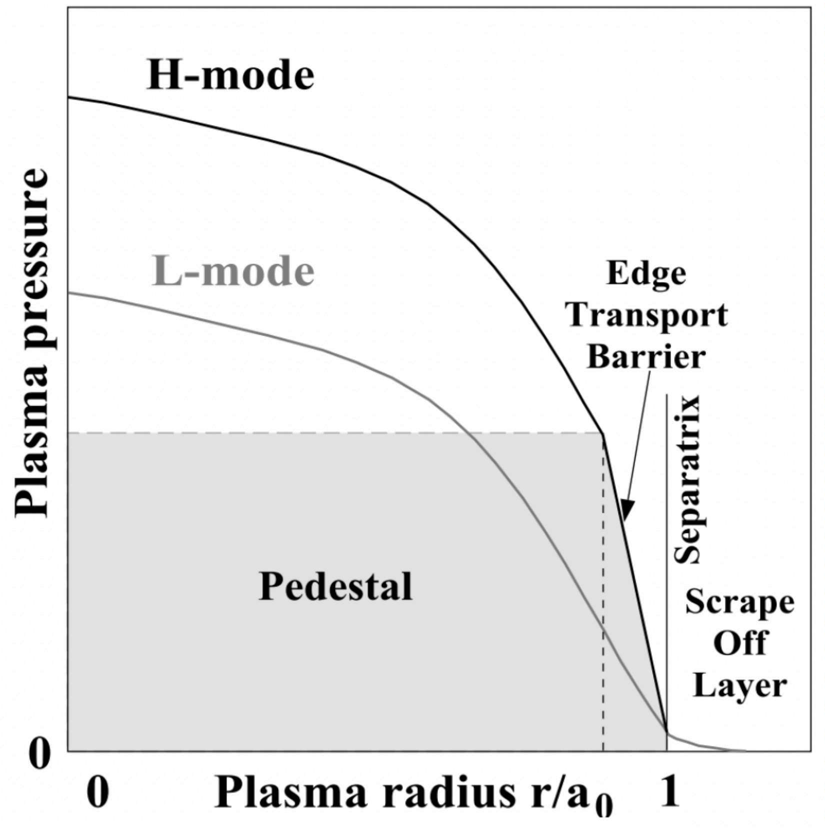
- A fusion reactor has an **enormous range of scales**. In the core, we have **turbulence at micro-scales** and background **profile evolution (transport) at macro-scales**, separated by several orders of magnitude

GYRO simulation, Candy



$$L_{\text{turb}} \sim \rho_i \sim 0.1 - 0.5 \text{ mm} \quad \xleftrightarrow{\sim 10^3 - 10^4} \quad L_{\text{transp}} \sim a \gtrsim 1 \text{ m}$$

$$\tau_{\text{turb}} \sim a/v_{\text{ti}} \sim 10\ \mu\text{s} \quad \xleftrightarrow[\sim 10^5]{} \quad \tau_{\text{transp}} \sim \tau_E \sim 1 \text{ s}$$



small-scale turbulence **slowly-varying background profiles**

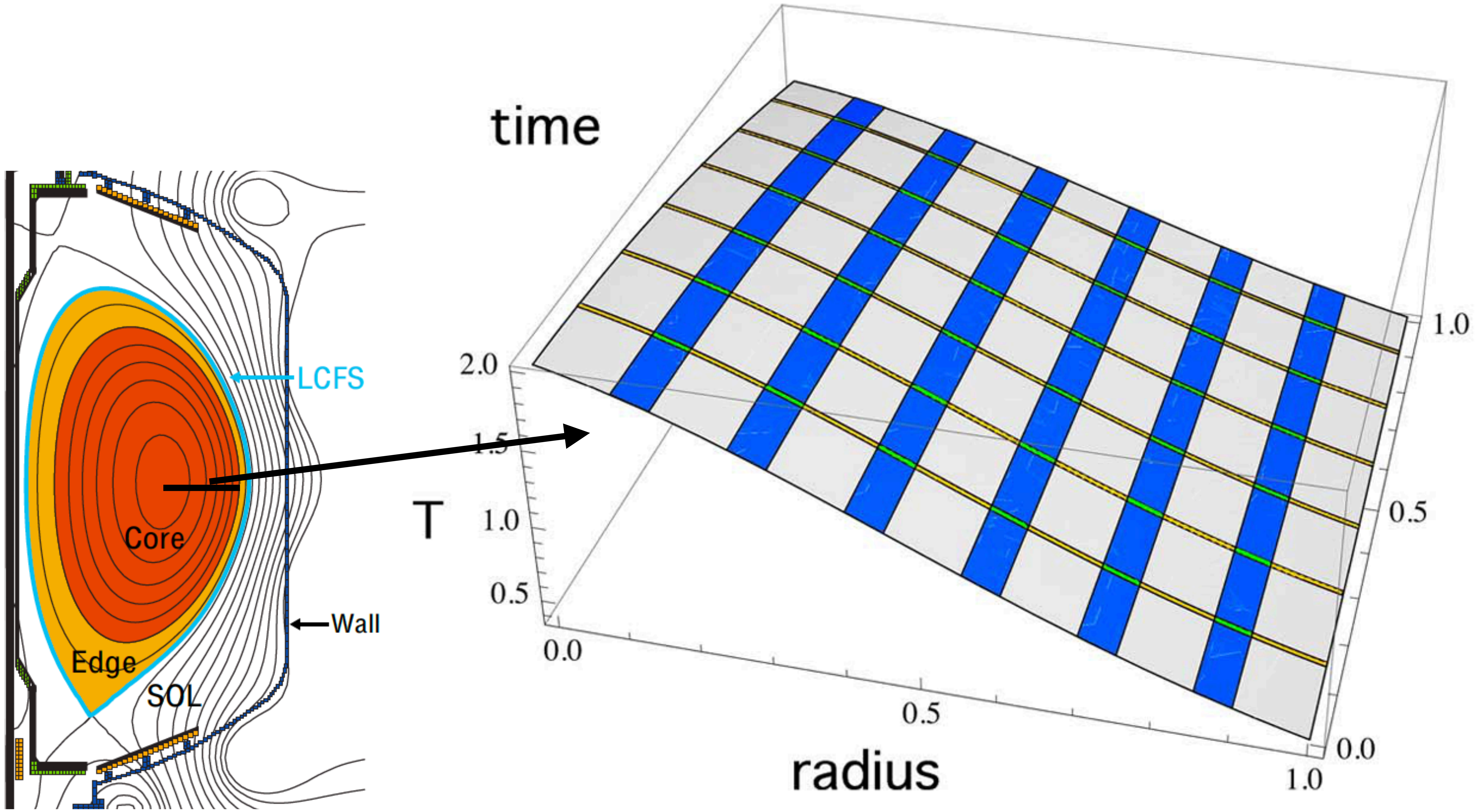
- Extreme temperatures require a **kinetic** description (instead of fluid), which means solving a PDE in 5D (3x+2v, after averaging over fast gyromotion to get **gyrokinetics**)
- Modeling this range of scales with a brute-force 5D+time discretization would then require

$$N \sim \left(\frac{L_{\text{turb}}}{L_{\text{transp}}}\right)^3 (N_v \sim 10)^2 \sim 10^{14} \text{ phase-space grid points in 5D}$$

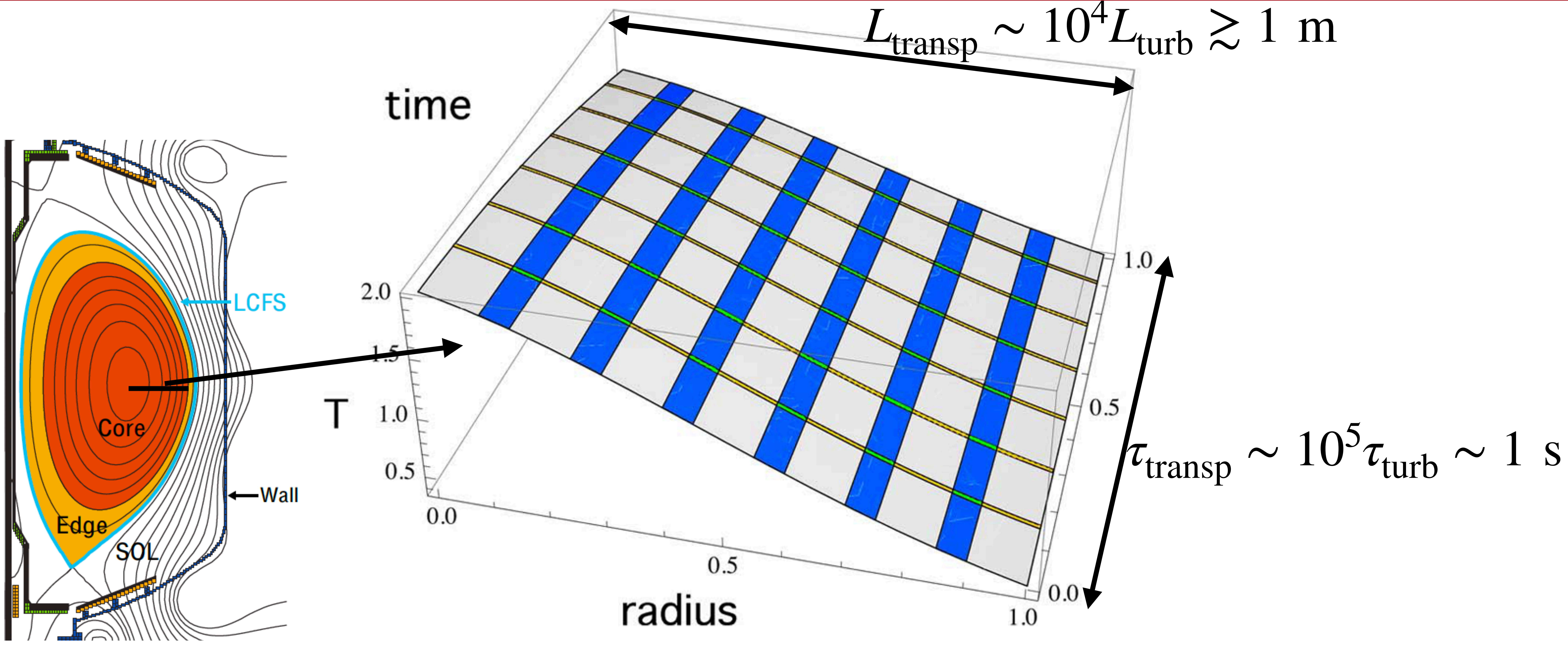
$$\text{NFLOP} \sim \mathcal{O}(N^2)\left(\frac{\tau_{\text{turb}}}{\tau_{\text{transp}}}\right) \sim 10^{33}, \text{ which would require } 10^{15} \text{ s on an exascale computer}$$

- These problems require **intense computational resources**, but also **clever multi-scale numerical algorithms and theory** to enable tractable calculations
- In **optimization** context, $\mathcal{O}(10^3)$ or more such calculations may be needed, so need additional acceleration to make tractable

Savings from using a multi-scale approach



Savings from using a multi-scale approach

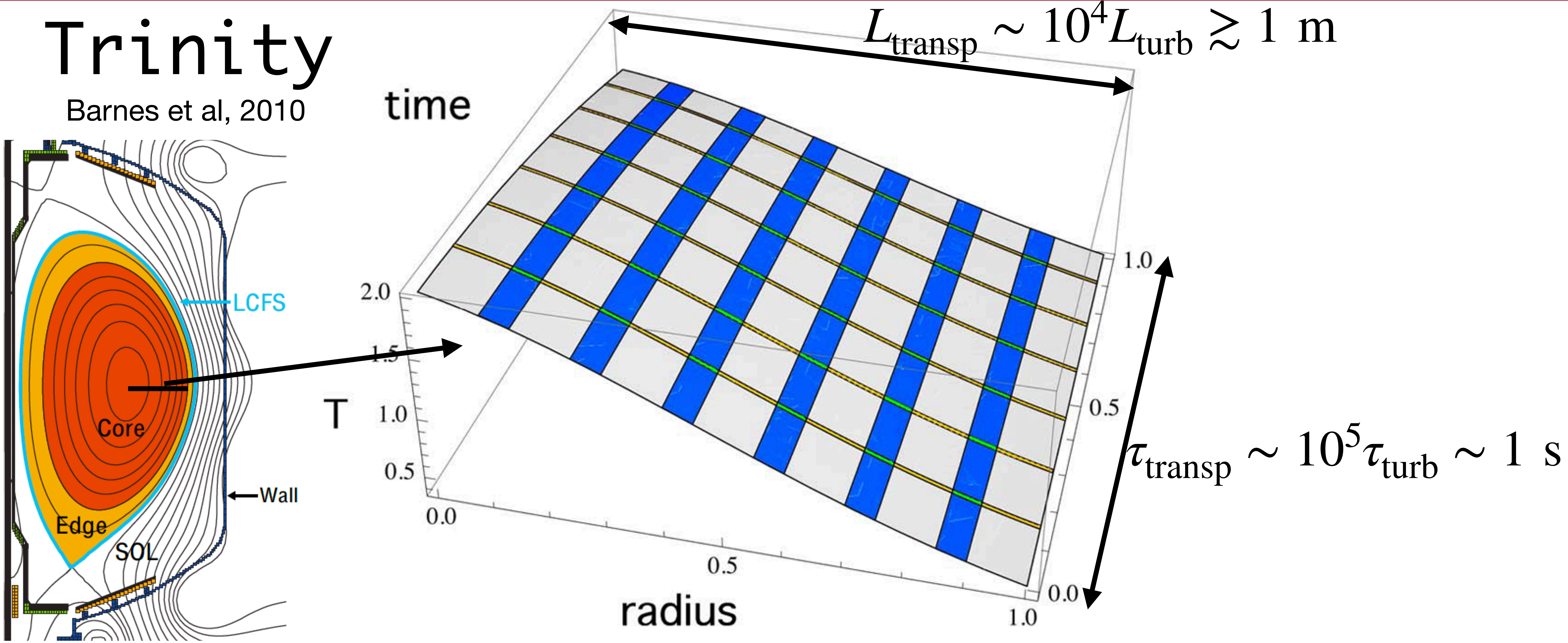


- Brute-force approach: fine space-time mesh for entire domain, with $\Delta x \sim L_{\text{turb}}$ and $\Delta t \sim \tau_{\text{turb}}$

Savings from using a multi-scale approach

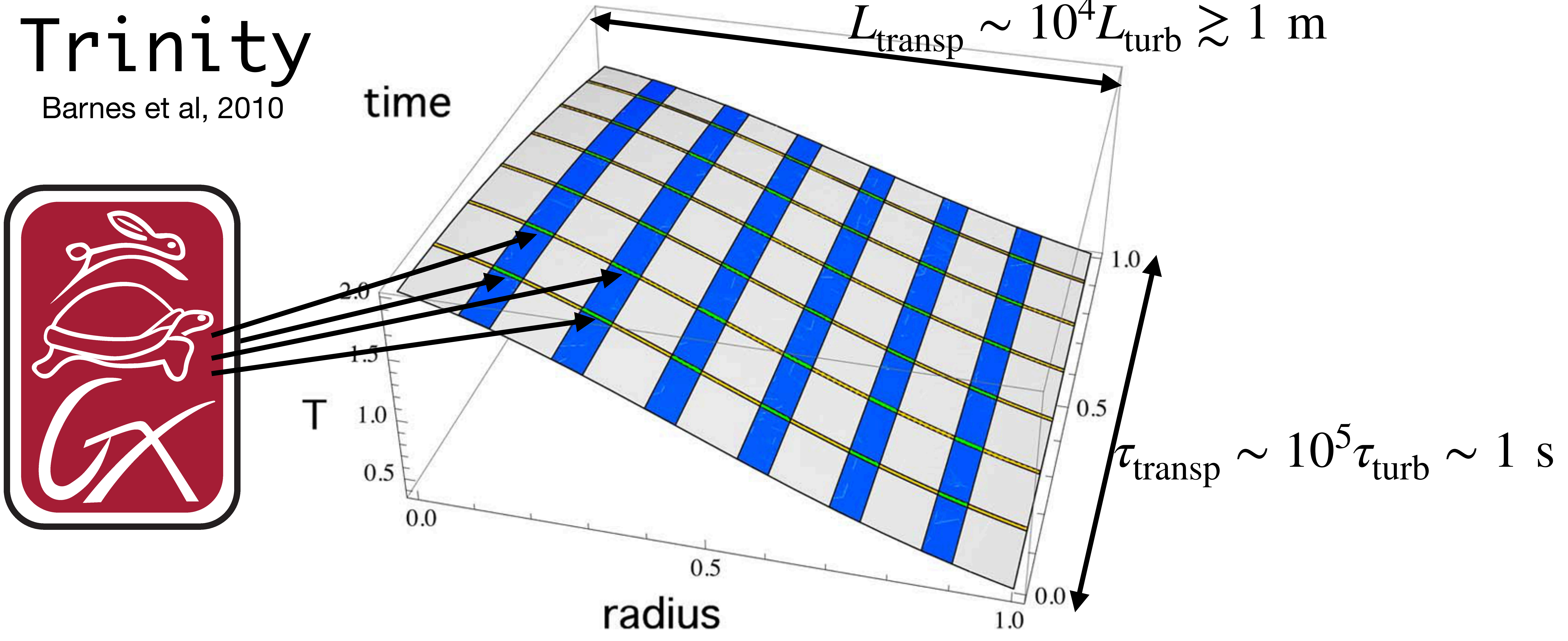
Trinity

Barnes et al, 2010



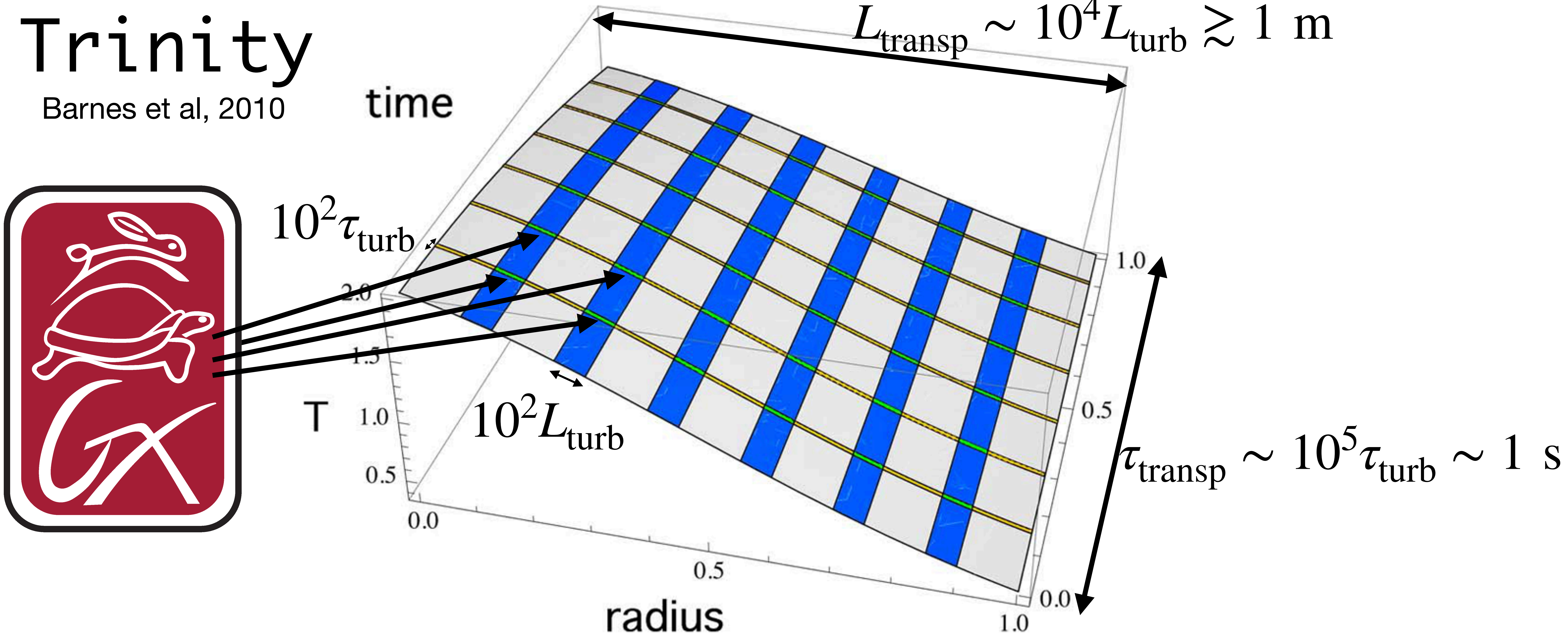
- Brute-force approach: fine space-time mesh for entire domain, with $\Delta x \sim L_{\mathrm{turb}}$ and $\Delta t \sim \tau_{\mathrm{turb}}$
- Trinity **multi-scale** approach (Barnes et al, 2010):

Savings from using a multi-scale approach



- Brute-force approach: fine space-time mesh for entire domain, with $\Delta x \sim L_{\rm turb}$ and $\Delta t \sim \tau_{\rm turb}$
- `Trinity` **multi-scale** approach (Barnes et al, 2010):
  - Can "zoom in" with small regions of fine grid (for turbulence) embedded in coarse grid (for slow profile evolution), in both space and time, dramatically reducing the domain over which a fine space-time mesh is needed

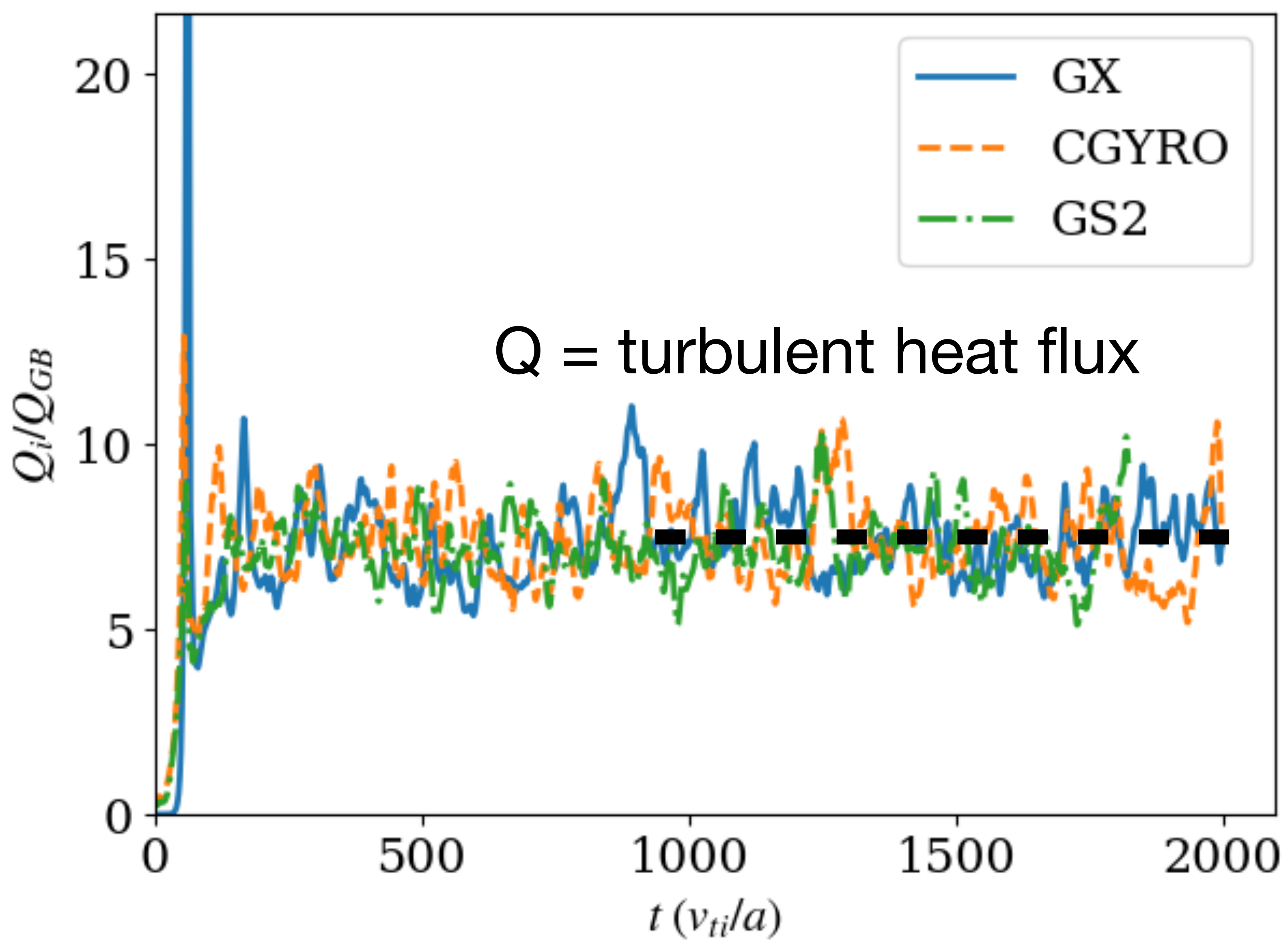
Savings from using a multi-scale approach

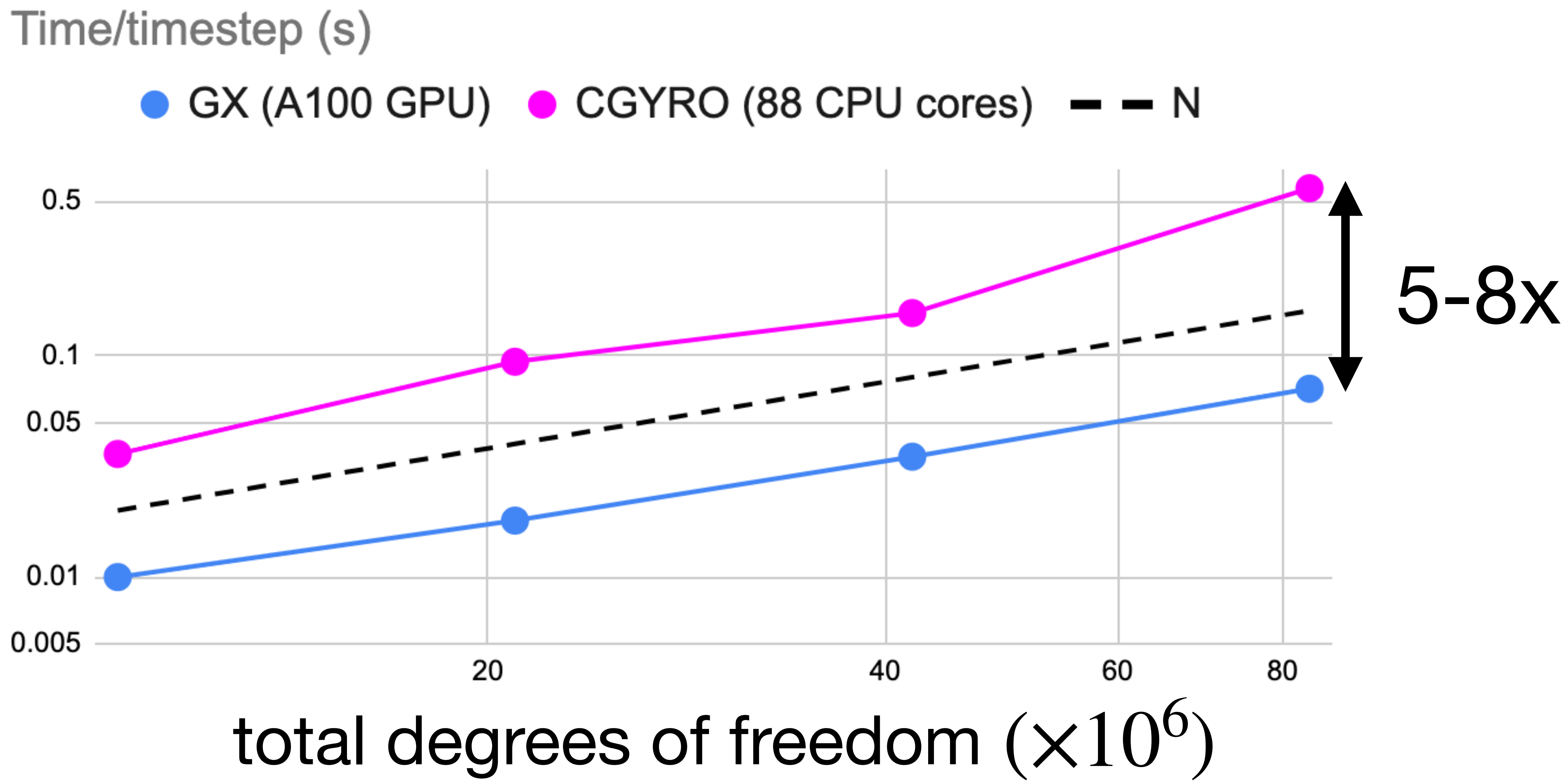


- Brute-force approach: fine space-time mesh for entire domain, with $\Delta x \sim L_{turb}$ and $\Delta t \sim \tau_{turb}$
- `Trinity` **multi-scale** approach (Barnes et al, 2010):
  - Can "zoom in" with small regions of fine grid (for turbulence) embedded in coarse grid (for slow profile evolution), in both space and time, dramatically reducing the domain over which a fine space-time mesh is needed

GX: a GPU-native pseudo-spectral kinetic turbulence code

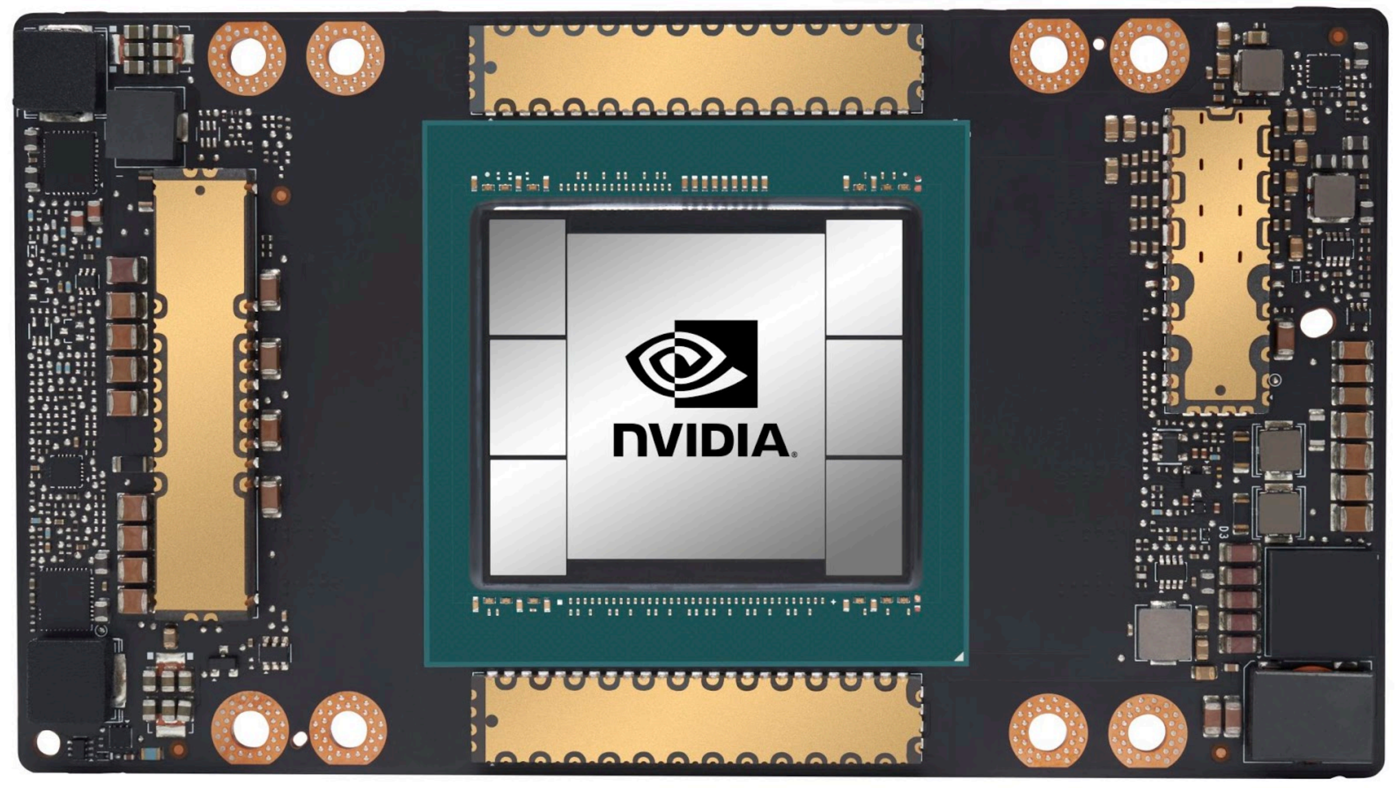
- GX models micro-scale turbulent fluctuations, which drive heat and particle losses in the reactor
  - Solves a nonlinear PDE in 5D+time
- Uses pseudo-spectral (Fourier-Hermite-Laguerre) methods
  - Ideal for GPUs (compute intensity is mostly in fast transforms and tensor operations)
- Designed and implemented directly in CUDA/C





Cyclone base case, adiabatic electrons

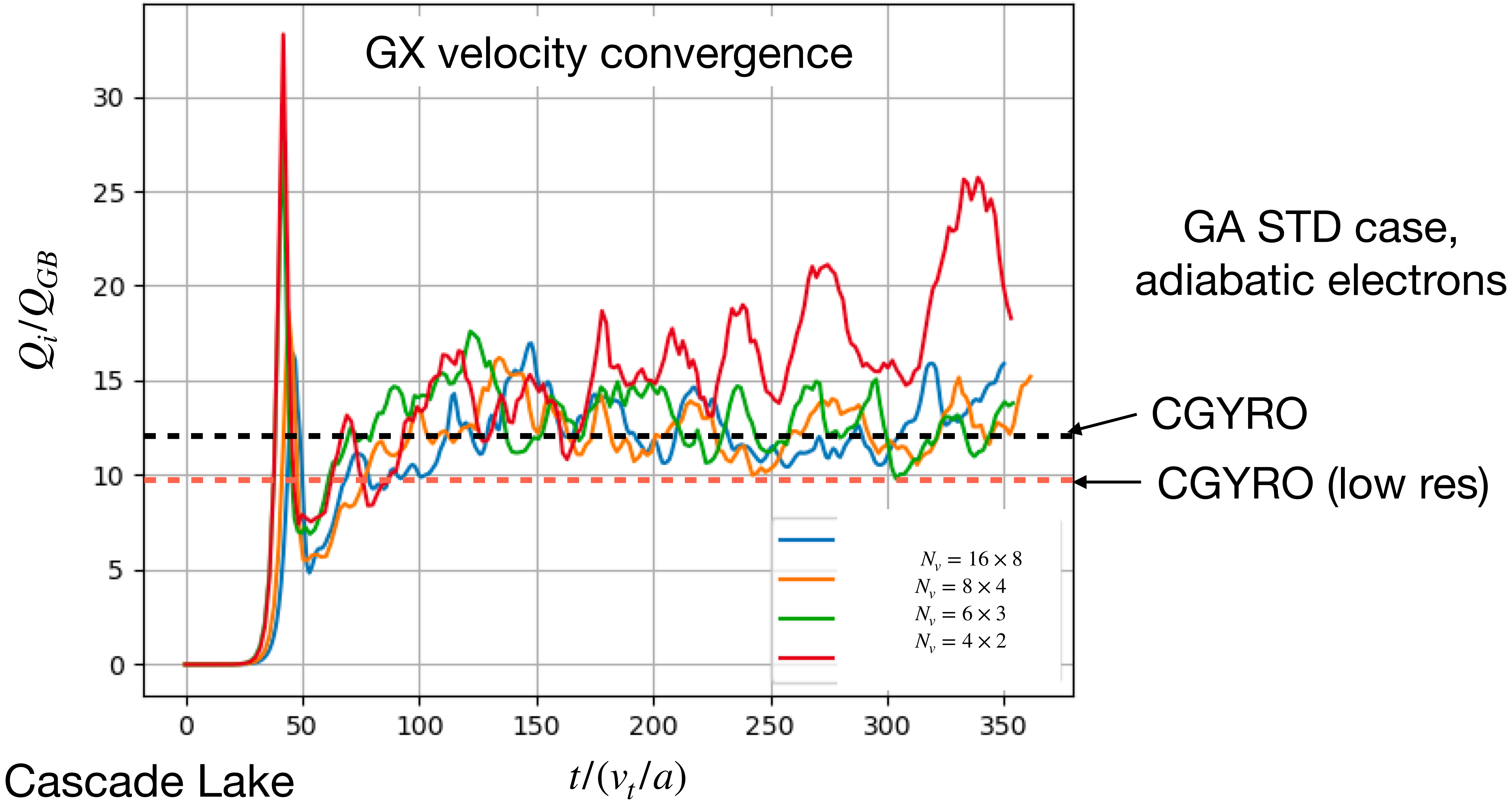
https://gx.readthedocs.io
https://bitbucket.org/gyrokinetics/gx



Combining algorithmic and hardware improvements

- GX's velocity basis appears to have **better convergence properties**, making GX still accurate even with 7x less velocity resolution than CGYRO
- **Combination of algorithmic and hardware** improvements makes GX **10-15x faster** than CGYRO on 256 cores!

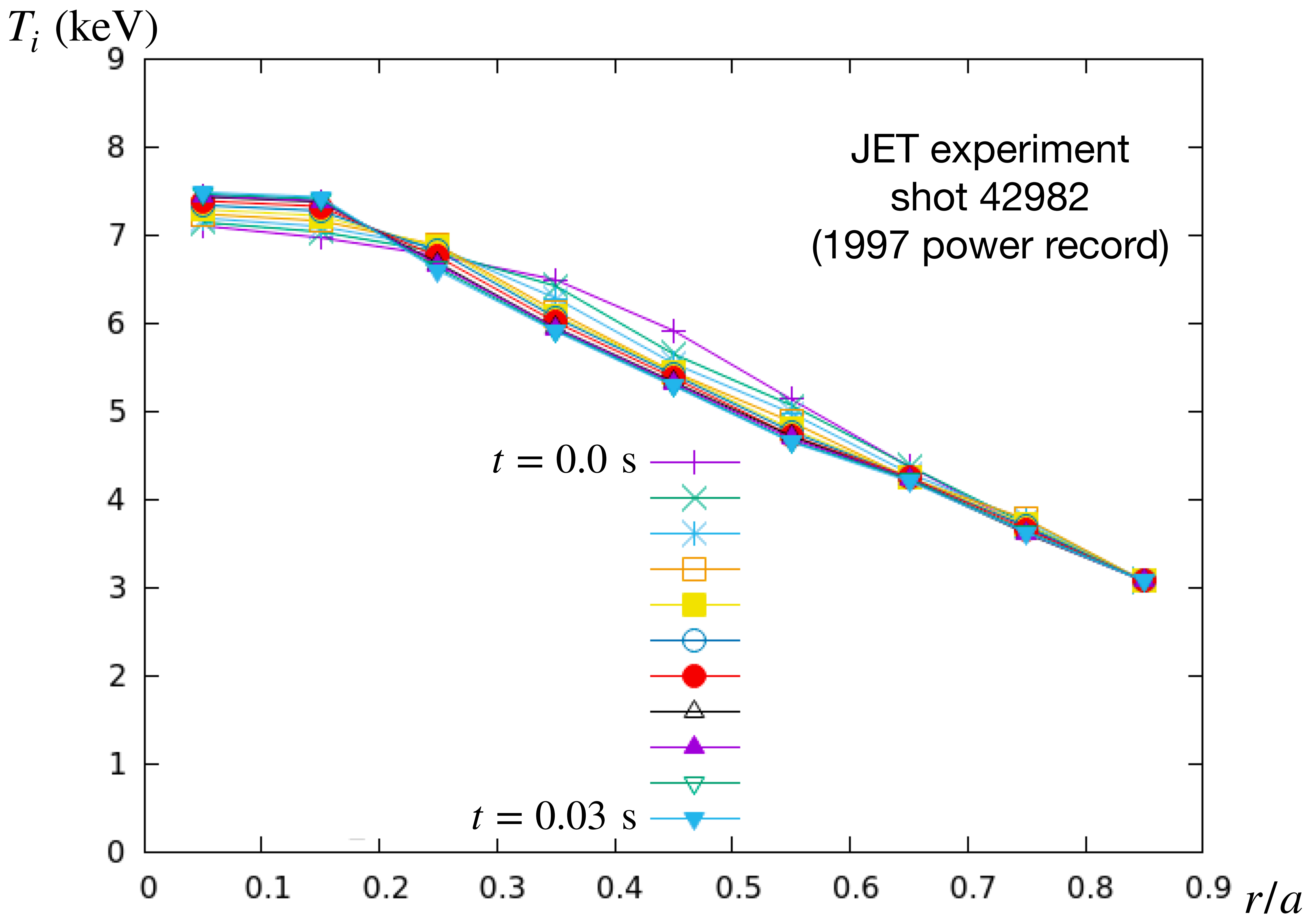
	GX (1 A100 GPU)		CGYRO (256 CPU cores†)	
Velocity resolution	Heat flux	Simulation time	Heat flux	Simulation time
16 x 8	12.5	17.5 min	12.1	31 min ~ 8000 CPU-min
8 x 4	12.3	3 min	9.8	9.5 min ~ 2400 CPU-
6 x 3	12.8	2.3 min	—	—
4 x 2	17.5	1.6 min	—	—



†CPU = 2.9 GHz Intel Cascade Lake

Preliminary GX + Trinity simulation

- The speed of GX makes once-daunting coupled turbulence-transport simulations manageable



8 radial cells

16 GX instances (2 per cell for a Jacobian calculation)

10 timesteps, 18 total evaluations (including intermediate Newton iterations)

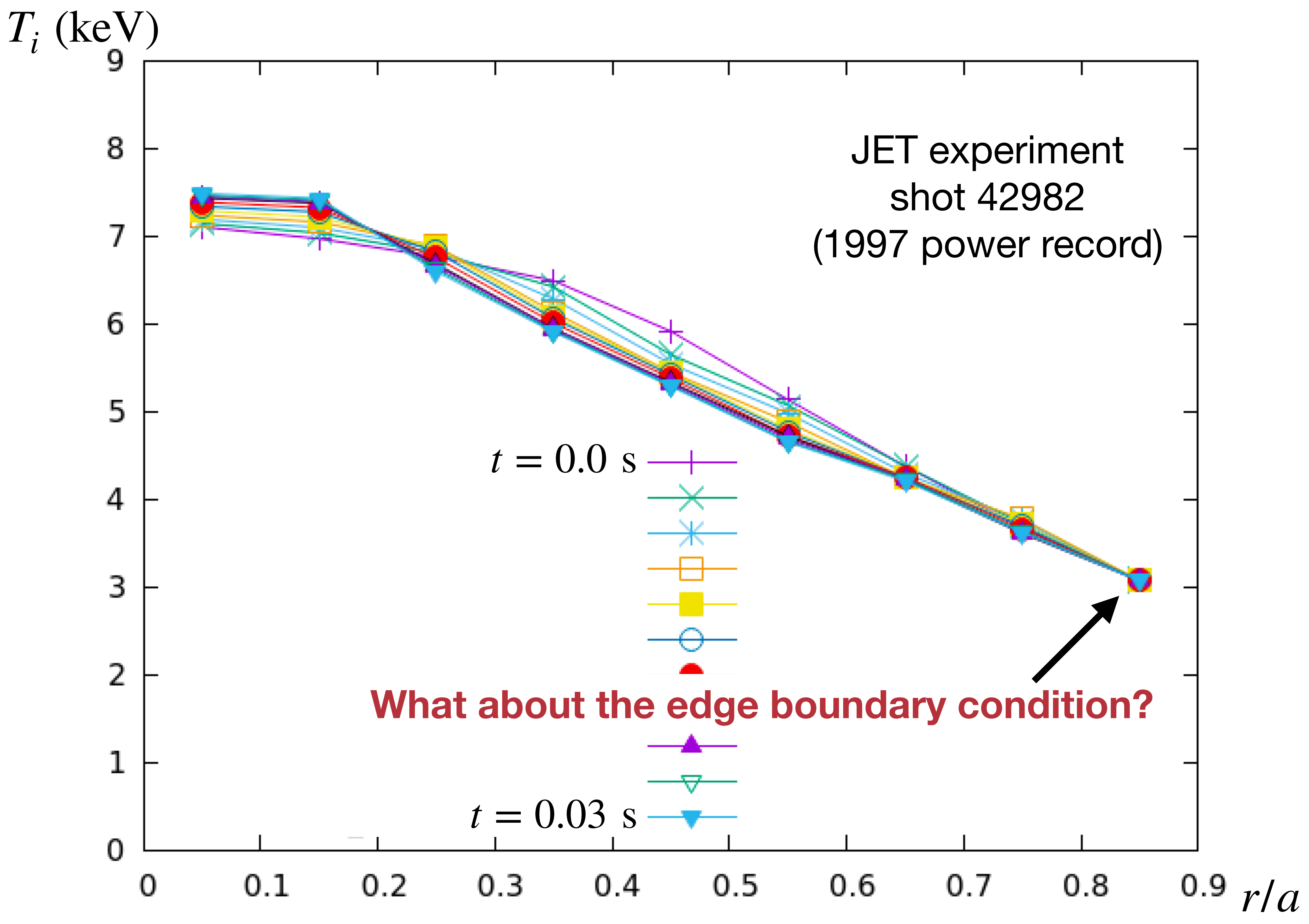
18 x 16 = 288 nonlinear turbulence evaluations

~2 hours total on 16 GPUs (V100)
(would be ~70% faster on A100)

(Might have taken CGYRO+Trinity ~12 hours on ~4000 cores)

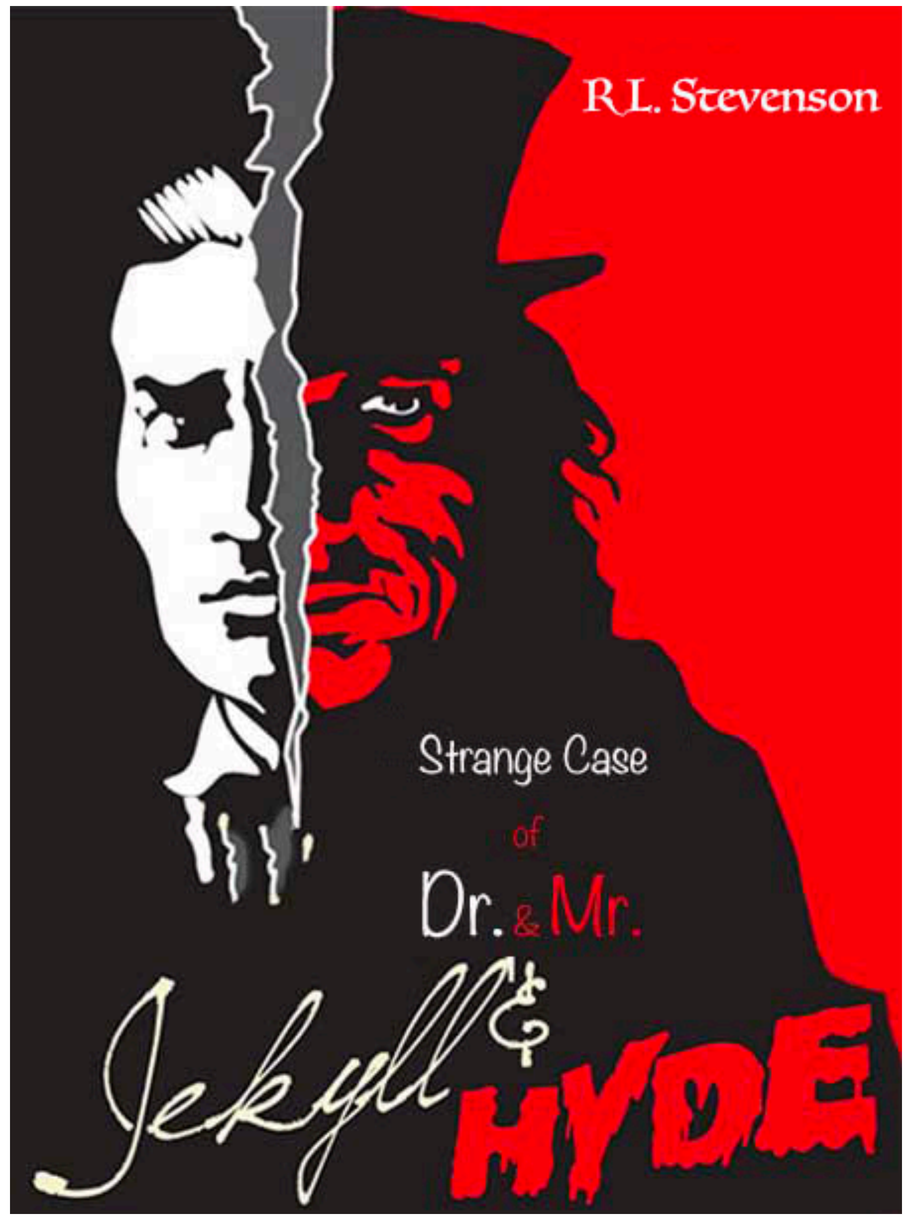
GX+Trinity transport model can be used to predict and optimize core profiles for future fusion experiments!

Thinking on the edge



Gkeyll: a gyrokinetic model for the boundary

- Modeling boundary requires specialized kinetic turbulence codes; can't use multi-scale approach from core
- Gkeyll is specialized for modeling kinetic turbulence in the tokamak boundary
  - Handles arbitrarily large fluctuations, special boundary conditions where plasma interacts with walls
- Energy-conserving discontinuous Galerkin (DG) discretization scheme for Hamiltonian systems (like GK)
- **Central contribution of my thesis work:** Novel scheme that models electromagnetic interactions between kinetic plasma turbulence and the confining magnetic field in tokamak boundary for the first time
- Massively-parallel implementation scales efficiently to ~1000 CPU cores; GPU implementation in progress!



https://github.com/ammarhakim/gkyl/

https://gkyl.readthedocs.io

https://gkyl.readthedocs.io/en/latest/gkyl/pubs.html

The discontinuous Galerkin method

- We use the **discontinuous Galerkin** (DG) method in Gkeyll
  - Class of finite-element methods with discontinuous basis functions to represent solution in each cell
  - Highly local, highly parallelizable, allows high-order accuracy, enforces local conservation laws
  - Can use limiters for stability (as in FV)

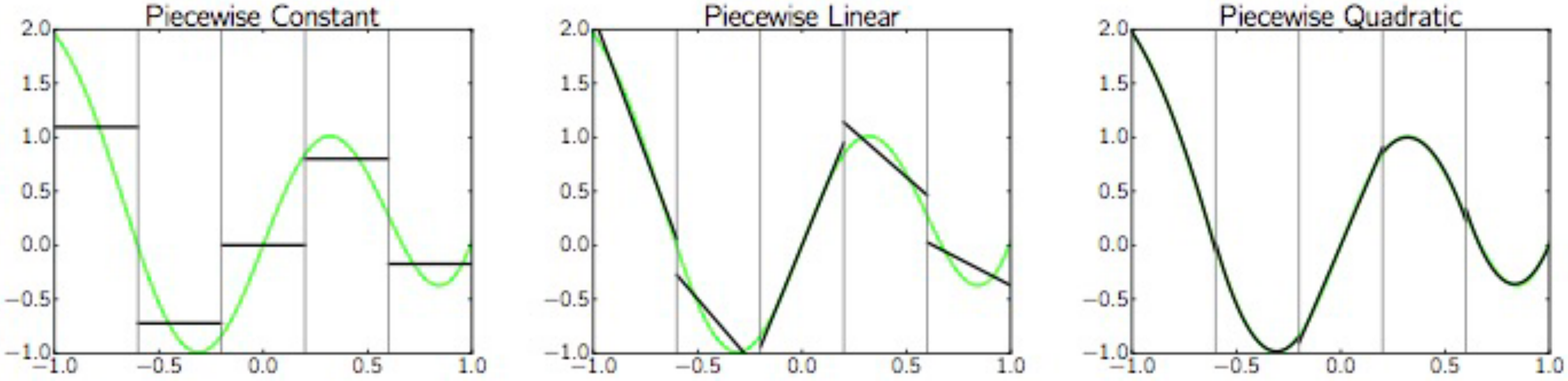


Figure: The best L_2 fit of $x^4 + \sin(5x)$ (green) using piecewise constant (left), linear (center), and quadratic (right) polynomials.

Orthonormal modal DG

- Our system can be expressed as a hyperbolic conservation law: $\frac{\partial f}{\partial t} + \nabla_{\vec{Z}} \cdot \left(\vec{\alpha} f\right) = 0$
- *Modal* expansion of solution in each cell: $f(\vec{Z},t) = \sum_{k}^{N_b} f_k(t) w_k(\vec{Z})$
- Fundamental DG operations can be expressed as tensor products, e.g. volume term:

$$\int_{C_m} d\vec{Z}\ f\vec{\alpha}\cdot\nabla w_i = \sum_{j,k} \underbrace{\left(\int_{C_m} d\vec{Z}\ w_j w_k \nabla w_i\right)}_{\overleftrightarrow{T}_{ijk}} \cdot \vec{\alpha}_j f_k$$

- Naively, this requires $\mathcal{O}(N_b^3)$ operations, same as quadrature (without aliasing)
- But if we choose basis functions to be *orthonormal*, $\overleftrightarrow{T}_{ijk}$ is sparse!
  - We use “Serendipity” Legendre polynomials as our orthonormal basis functions
- Use a computer algebra system (Maxima) to compute sparse tensor products *analytically* and generate solver kernels

Computer-generated DG kernels

$$\mathtt{out}_i = \sum_{j,k} \overleftrightarrow{T}_{ijk} \cdot \overrightarrow{\alpha}_j f_k \quad \Rightarrow$$

```
out[1] += 0.3061862178478971*(alphax[9]*f[9]+alphax[7]*f[7]+alphax[4]*f[4]+alphax[3]*f[3]+alphax[1]*f[1]
out[2] += 0.3061862178478971*(alphay[16]*f[16]+alphay[12]*f[12]+alphay[9]*f[9]+alphay[8]*f[8]+alphay[7]*
out[3] += 0.3061862178478971*(alphaz[1]*f[1]+alphaz[0]*f[0]);
out[4] += 0.3061862178478971*(alphav[26]*f[26]+alphav[23]*f[23]+alphav[19]*f[19]+alphav[18]*f[18]+alphav
out[6] += 0.3061862178478971*(alphax[9]*f[17]+(alphay[8]+alphax[7])*f[16]+f[8]*alphay[16]+alphay[5]*f[12
out[7] += 0.3061862178478971*(alphax[9]*f[18]+alphax[4]*f[11]+alphax[1]*f[7]+f[1]*alphax[7]+alphax[0]*f[
out[8] += 0.3061862178478971*(alphay[12]*f[21]+alphay[9]*f[18]+alphay[6]*f[16]+f[6]*alphay[16]+alphay[5]
out[9] += 0.3061862178478971*(alphav[19]*f[26]+f[19]*alphav[26]+alphav[15]*f[23]+f[15]*alphav[23]+(alpha
out[10] += 0.3061862178478971*(alphav[23]*f[28]+(alphav[18]+alphay[16])*f[26]+f[18]*alphav[26]+alphav[15
out[11] += 0.3061862178478971*(alphav[23]*f[29]+alphav[17]*f[26]+f[17]*alphav[26]+alphav[15]*f[25]+alpha
out[12] += 0.3061862178478971*(alphax[9]*f[23]+alphax[7]*f[21]+alphax[4]*f[15]+alphax[3]*f[14]+alphax[1]
out[13] += 0.3061862178478971*(alphay[16]*f[27]+alphay[9]*f[23]+alphay[8]*f[22]+alphay[7]*f[21]+alphay[6
out[14] += 0.3061862178478971*(alphaz[1]*f[12]+alphaz[0]*f[5]);
out[15] += 0.3061862178478971*(alphav[26]*f[31]+alphav[19]*f[30]+alphav[18]*f[29]+alphav[17]*f[28]+alpha
out[16] += 0.3061862178478971*(alphax[9]*f[26]+alphay[5]*f[21]+alphax[4]*f[19]+alphay[4]*f[18]+(alphay[2
out[17] += 0.3061862178478971*(alphav[15]*f[28]+(alphav[11]+alphay[8]+alphax[7])*f[26]+f[11]*alphav[26]+
out[18] += 0.3061862178478971*(alphav[15]*f[29]+alphav[10]*f[26]+f[10]*alphav[26]+alphav[23]*f[25]+alpha
out[19] += 0.3061862178478971*(alphav[23]*f[31]+alphav[15]*f[30]+alphay[12]*f[29]+alphav[12]*f[27]+(alph
out[20] += 0.3061862178478971*(alphax[9]*f[28]+(alphay[8]+alphax[7])*f[27]+alphax[4]*f[24]+alphay[4]*f[2
out[21] += 0.3061862178478971*(alphax[9]*f[29]+alphax[4]*f[25]+alphax[1]*f[21]+alphax[0]*f[14]+(alphax[7
out[22] += 0.3061862178478971*(alphay[9]*f[29]+alphay[6]*f[27]+alphay[4]*f[25]+alphay[2]*f[22]+alphay[1]
out[23] += 0.3061862178478971*(alphav[19]*f[31]+alphav[26]*f[30]+(alphav[11]+alphax[7])*f[29]+alphav[10]
out[24] += 0.3061862178478971*((alphav[18]+alphay[16])*f[31]+(alphav[11]+alphay[8])*f[30]+(alphav[26]+al
out[25] += 0.3061862178478971*(alphav[17]*f[31]+alphav[10]*f[30]+alphav[9]*f[29]+alphav[26]*f[28]+alphav
out[26] += 0.3061862178478971*(alphav[15]*f[31]+alphav[23]*f[30]+alphay[5]*f[29]+alphav[5]*f[27]+(alphav
out[27] += 0.3061862178478971*(alphax[9]*f[31]+alphax[4]*f[30]+alphay[4]*f[29]+(alphay[2]+alphax[1])*f[2
out[28] += 0.3061862178478971*((alphav[11]+alphay[8]+alphax[7])*f[31]+(alphav[18]+alphay[16]+alphax[3])*
out[29] += 0.3061862178478971*(alphav[10]*f[31]+alphav[17]*f[30]+(alphav[4]+alphax[1])*f[29]+alphav[19]*
out[30] += 0.3061862178478971*((alphav[9]+alphay[6])*f[31]+(alphav[4]+alphay[2])*f[30]+(alphav[17]+alpha
out[31] += 0.3061862178478971*((alphav[4]+alphay[2]+alphax[1])*f[31]+(alphav[9]+alphay[6]+alphax[0])*f[3
```

Computer-generated DG kernels

$$\mathtt{out}_i = \sum_{j,k} \overleftrightarrow{T}_{ijk} \cdot \overrightarrow{\alpha}_j f_k \quad \rightarrow$$

```
out[1] += 0.3061862178478971*(alphax[9]*f[9]+alphax[7]*f[7]+alphax[4]*f[4]+alphax[3]*f[3]+alphax[1]*f[1]
out[2] += 0.3061862178478971*(alphay[16]*f[16]+alphay[12]*f[12]+alphay[9]*f[9]+alphay[8]*f[8]+alphay[7]*
out[3] += 0.3061862178478971*(alphaz[1]*f[1]+alphaz[0]*f[0]);
out[4] += 0.3061862178478971*(alphav[26]*f[26]+alphav[23]*f[23]+alphav[19]*f[19]+alphav[18]*f[18]+alphav
out[6] += 0.3061862178478971*(alphax[9]*f[17]+(alphay[8]+alphax[7])*f[16]+f[8]*alphay[16]+alphay[5]*f[12
out[7] += 0.3061862178478971*(alphax[9]*f[18]+alphax[4]*f[11]+alphax[1]*f[7]+f[1]*alphax[7]+alphax[0]*f[
out[8] += 0.3061862178478971*(alphay[12]*f[21]+alphay[9]*f[18]+alphay[6]*f[16]+f[6]*alphay[16]+alphay[5]
out[9] += 0.3061862178478971*(alphav[19]*f[26]+f[19]*alphav[26]+alphav[15]*f[23]+f[15]*alphav[23]+(alpha
out[10] += 0.3061862178478971*(alphav[23]*f[28]+(alphav[18]+alphay[16])*f[26]+f[18]*alphav[26]+alphav[15
out[11] += 0.3061862178478971*(alphav[23]*f[29]+alphav[17]*f[26]+f[17]*alphav[26]+alphav[15]*f[25]+alpha
out[12] += 0.3061862178478971*(alphax[9]*f[23]+alphax[7]*f[21]+alphax[4]*f[15]+alphax[3]*f[14]+alphax[1]
out[13] += 0.3061862178478971*(alphay[16]*f[27]+alphay[9]*f[23]+alphay[8]*f[22]+alphay[7]*f[21]+alphay[6
out[14] += 0.3061862178478971*(alphaz[1]*f[12]+alphaz[0]*f[5]);
out[15] += 0.3061862178478971*(alphav[26]*f[31]+alphav[19]*f[30]+alphav[18]*f[29]+alphav[17]*f[28]+alpha
out[16] += 0.3061862178478971*(alphax[9]*f[26]+alphay[5]*f[21]+alphax[4]*f[19]+alphay[4]*f[18]+(alphay[2
out[17] += 0.3061862178478971*(alphav[15]*f[28]+(alphav[11]+alphay[8]+alphax[7])*f[26]+f[11]*alphav[26]+
out[18] += 0.3061862178478971*(alphav[15]*f[29]+alphav[10]*f[26]+f[10]*alphav[26]+alphav[23]*f[25]+alpha
out[19] += 0.3061862178478971*(alphav[23]*f[31]+alphav[15]*f[30]+alphay[12]*f[29]+alphav[12]*f[27]+(alph
out[20] += 0.3061862178478971*(alphax[9]*f[28]+(alphay[8]+alphax[7])*f[27]+alphax[4]*f[24]+alphay[4]*f[2
out[21] += 0.3061862178478971*(alphax[9]*f[29]+alphax[4]*f[25]+alphax[1]*f[21]+alphax[0]*f[14]+(alphax[7
out[22] += 0.3061862178478971*(alphay[9]*f[29]+alphay[6]*f[27]+alphay[4]*f[25]+alphay[2]*f[22]+alphay[1]
out[23] += 0.3061862178478971*(alphav[19]*f[31]+alphav[26]*f[30]+(alphav[11]+alphax[7])*f[29]+alphav[10]
out[24] += 0.3061862178478971*((alphav[18]+alphay[16])*f[31]+(alphav[11]+alphay[8])*f[30]+(alphav[26]+al
out[25] += 0.3061862178478971*(alphav[17]*f[31]+alphav[10]*f[30]+alphav[9]*f[29]+alphav[26]*f[28]+alphav
out[26] += 0.3061862178478971*(alphav[15]*f[31]+alphav[23]*f[30]+alphay[5]*f[29]+alphav[5]*f[27]+(alphav
out[27] += 0.3061862178478971*(alphax[9]*f[31]+alphax[4]*f[30]+alphay[4]*f[29]+(alphay[2]+alphax[1])*f[2
out[28] += 0.3061862178478971*((alphav[11]+alphay[8]+alphax[7])*f[31]+(alphav[18]+alphay[16]+alphax[3])*
out[29] += 0.3061862178478971*(alphav[10]*f[31]+alphav[17]*f[30]+(alphav[4]+alphax[1])*f[29]+alphav[19]*
out[30] += 0.3061862178478971*((alphav[9]+alphay[6])*f[31]+(alphav[4]+alphay[2])*f[30]+(alphav[17]+alpha
out[31] += 0.3061862178478971*((alphav[4]+alphay[2]+alphax[1])*f[31]+(alphav[9]+alphay[6]+alphax[0])*f[3
```

- Maxima generates thousands of lines of machine-written C code… no loops!
  - Takes advantage of significant sparsity in tensor contractions

Computer-generated DG kernels

5D piecewise linear basis = 32 basis functions

$$\mathtt{out}_i = \sum_{j,k} \overleftrightarrow{T}_{ijk} \cdot \overrightarrow{\alpha}_j f_k$$

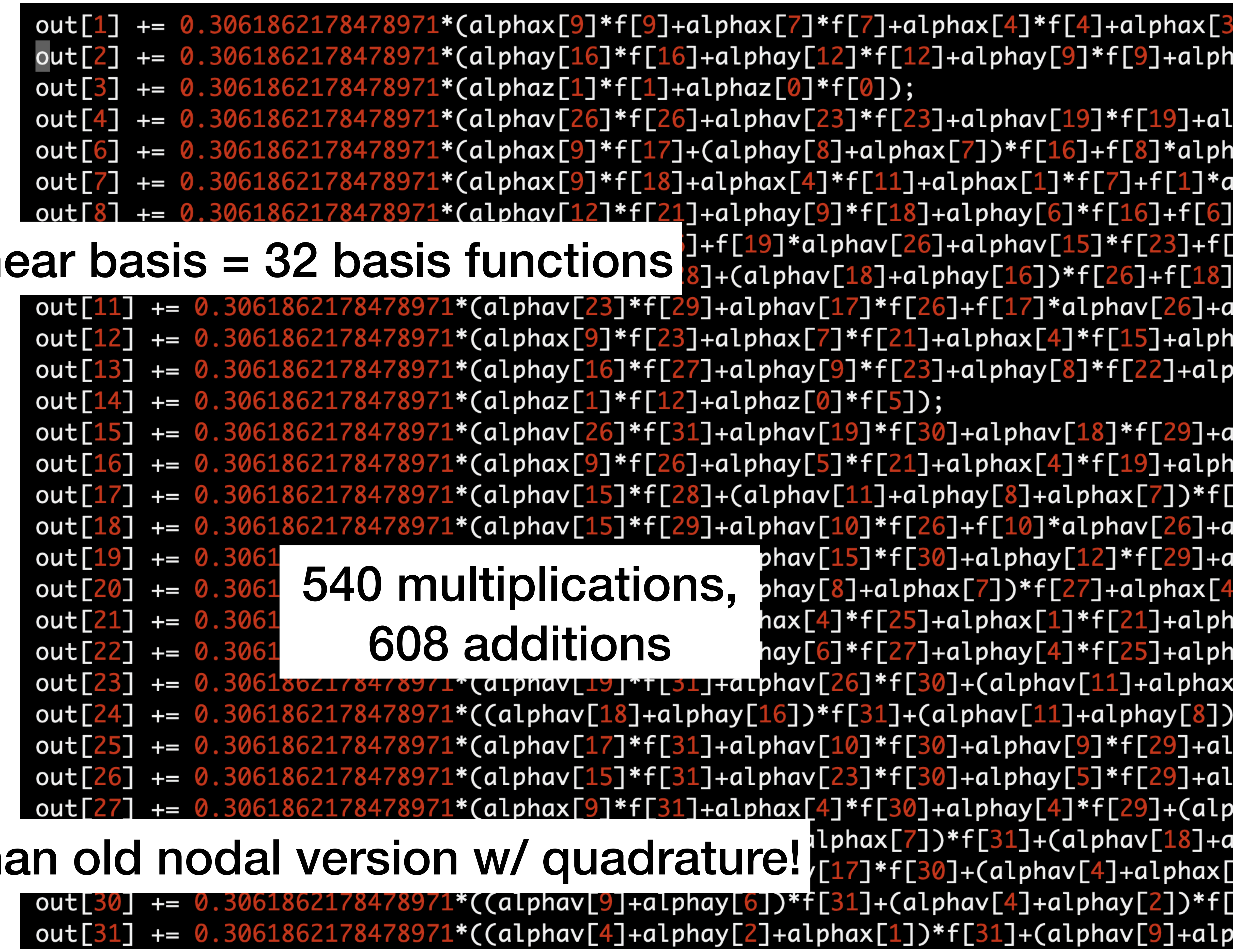
163,840 multiplications,
98,304 additions

```
out[1] += 0.3061862178478971*(alphax[9]*f[9]+alphax[7]*f[7]+alphax[4]*f[4]+alphax[3]*f[3]+alphax[1]*f[1]
out[2] += 0.3061862178478971*(alphay[16]*f[16]+alphay[12]*f[12]+alphay[9]*f[9]+alphay[8]*f[8]+alphay[7]*
out[3] += 0.3061862178478971*(alphaz[1]*f[1]+alphaz[0]*f[0]);
out[4] += 0.3061862178478971*(alphav[26]*f[26]+alphav[23]*f[23]+alphav[19]*f[19]+alphav[18]*f[18]+alphav
out[6] += 0.3061862178478971*(alphax[9]*f[17]+(alphay[8]+alphax[7])*f[16]+f[8]*alphay[16]+alphay[5]*f[12
out[7] += 0.3061862178478971*(alphax[9]*f[18]+alphax[4]*f[11]+alphax[1]*f[7]+f[1]*alphax[7]+alphax[0]*f[
out[8] += 0.3061862178478971*(alphay[12]*f[21]+alphay[9]*f[18]+alphay[6]*f[16]+f[6]*alphay[16]+alphay[5]
]+f[19]*alphav[26]+alphav[15]*f[23]+f[15]*alphav[23]+(alpha
8]+(alphav[18]+alphay[16])*f[26]+f[18]*alphav[26]+alphav[15
out[11] += 0.3061862178478971*(alphav[23]*f[29]+alphav[17]*f[26]+f[17]*alphav[26]+alphav[15]*f[25]+alpha
out[12] += 0.3061862178478971*(alphax[9]*f[23]+alphax[7]*f[21]+alphax[4]*f[15]+alphax[3]*f[14]+alphax[1]
out[13] += 0.3061862178478971*(alphay[16]*f[27]+alphay[9]*f[23]+alphay[8]*f[22]+alphay[7]*f[21]+alphay[6
out[14] += 0.3061862178478971*(alphaz[1]*f[12]+alphaz[0]*f[5]);
out[15] += 0.3061862178478971*(alphav[26]*f[31]+alphav[19]*f[30]+alphav[18]*f[29]+alphav[17]*f[28]+alpha
out[16] += 0.3061862178478971*(alphax[9]*f[26]+alphay[5]*f[21]+alphax[4]*f[19]+alphay[4]*f[18]+(alphay[2
out[17] += 0.3061862178478971*(alphav[15]*f[28]+(alphav[11]+alphay[8]+alphax[7])*f[26]+f[11]*alphav[26]+
out[18] += 0.3061862178478971*(alphav[15]*f[29]+alphav[10]*f[26]+f[10]*alphav[26]+alphav[23]*f[25]+alpha
out[19] += 0.3061 phav[15]*f[30]+alphay[12]*f[29]+alphav[12]*f[27]+(alph
out[20] += 0.3061 phay[8]+alphax[7])*f[27]+alphax[4]*f[24]+alphay[4]*f[2
out[21] += 0.3061 hax[4]*f[25]+alphax[1]*f[21]+alphax[0]*f[14]+(alphax[7
out[22] += 0.3061 hay[6]*f[27]+alphay[4]*f[25]+alphay[2]*f[22]+alphay[1]
out[23] += 0.3061862178478971*(alphav[19]*f[31]+alphav[26]*f[30]+(alphav[11]+alphax[7])*f[29]+alphav[10]
out[24] += 0.3061862178478971*((alphav[18]+alphay[16])*f[31]+(alphav[11]+alphay[8])*f[30]+(alphav[26]+al
out[25] += 0.3061862178478971*(alphav[17]*f[31]+alphav[10]*f[30]+alphav[9]*f[29]+alphav[26]*f[28]+alphav
out[26] += 0.3061862178478971*(alphav[15]*f[31]+alphav[23]*f[30]+alphay[5]*f[29]+alphav[5]*f[27]+(alphav
out[27] += 0.3061862178478971*(alphax[9]*f[31]+alphax[4]*f[30]+alphay[4]*f[29]+(alphay[2]+alphax[1])*f[2
out[28] += 0.3061862178478971*((alphav[11]+alphay[8]+alphax[7])*f[31]+(alphav[18]+alphay[16]+alphax[3])*
out[29] += 0.3061862178478971*(alphav[10]*f[31]+alphav[17]*f[30]+(alphav[4]+alphax[1])*f[29]+alphav[19]*
out[30] += 0.3061862178478971*((alphav[9]+alphay[6])*f[31]+(alphav[4]+alphay[2])*f[30]+(alphav[17]+alpha
out[31] += 0.3061862178478971*((alphav[4]+alphay[2]+alphax[1])*f[31]+(alphav[9]+alphay[6]+alphax[0])*f[3
```

540 multiplications,
608 additions

- Maxima generates thousands of lines of machine-written C code… no loops!
  - Takes advantage of significant sparsity in tensor contractions

Computer-generated DG kernels



5D piecewise linear basis = 32 basis functions

$$\mathtt{out}_i = \sum_{j,k} \overleftrightarrow{T}_{ijk} \cdot \overrightarrow{\alpha}_j f_k \quad \Rightarrow$$

163,840 multiplications,
98,304 additions

540 multiplications,
608 additions

Code is ~30x faster than old nodal version w/ quadrature!

- Maxima generates thousands of lines of machine-written C code… no loops!
  - Takes advantage of significant sparsity in tensor contractions

Computer-generated DG kernels

```
out[1] += 0.3061862178478971*(alphax[9]*f[9]+alphax[7]*f[7]+alphax[4]*f[4]+alphax[3]*f[3]+alphax[1]*f[1]
out[2] += 0.3061862178478971*(alphay[16]*f[16]+alphay[12]*f[12]+alphay[9]*f[9]+alph
out[3] += 0.3061862178478971*(alphaz[1]*f[1]+alphaz[0]*f[0]);
out[4] += 0.3061862178478971*(alphav[26]*f[26]+alphav[23]*f[23]+alphav[19]*f[19]+al
out[6] += 0.3061862178478971*(alphax[9]*f[17]+(alphay[8]+alphax[7])*f[16]+f[8]*alph
out[7] += 0.3061862178478971*(alphax[9]*f[18]+alphax[4]*f[11]+alphax[1]*f[7]+f[1]*a
out[8] += 0.3061862178478971*(alphay[12]*f[21]+alphay[9]*f[18]+alphay[6]*f[16]+f[6]
]+f[19]*alphav[26]+alphav[15]*f[23]+f[
8]+(alphav[18]+alphay[16])*f[26]+f[18]
out[11] += 0.3061862178478971*(alphav[23]*f[29]+alphav[17]*f[26]+f[17]*alphav[26]+a
out[12] += 0.3061862178478971*(alphax[9]*f[23]+alphax[7]*f[21]+alphax[4]*f[15]+alph
out[13] += 0.3061862178478971*(alphay[16]*f[27]+alphay[9]*f[23]+alphay[8]*f[22]+alp
out[14] += 0.3061862178478971*(alphaz[1]*f[12]+alphaz[0]*f[5]);
out[15] += 0.3061862178478971*(alphav[26]*f[31]+alphav[19]*f[30]+alphav[18]*f[29]+a
out[16] += 0.3061862178478971*(alphax[9]*f[26]+alphay[5]*f[21]+alphax[4]*f[19]+alph
out[17] += 0.3061862178478971*(alphav[15]*f[28]+(alphav[11]+alphay[8]+alphax[7])*f[
out[18] += 0.3061862178478971*(alphav[15]*f[29]+alphav[10]*f[26]+f[10]*alphav[26]+a
out[19] += 0.3061                phav[15]*f[30]+alphay[12]*f[29]+alphav[12]*f[27]+(alph
out[20] += 0.3061                phay[8]+alphax[7])*f[27]+alphax[4]*f[24]+alphay[4]*f[2
out[21] += 0.3061                hax[4]*f[25]+alphax[1]*f[21]+alphax[0]*f[14]+(alphax[7
out[22] += 0.3061                hay[6]*f[27]+alphay[4]*f[25]+alphay[2]*f[22]+alphay[1]
out[23] += 0.3061862178478971*(alphav[19]*f[31]+alphav[26]*f[30]+(alphav[11]+alphax[7])*f[29]+alphav[10]
out[24] += 0.3061862178478971*((alphav[18]+alphay[16])*f[31]+(alphav[11]+alphay[8])*f[30]+(alphav[26]+al
out[25] += 0.3061862178478971*(alphav[17]*f[31]+alphav[10]*f[30]+alphav[9]*f[29]+alphav[26]*f[28]+alphav
out[26] += 0.3061862178478971*(alphav[15]*f[31]+alphav[23]*f[30]+alphay[5]*f[29]+alphav[5]*f[27]+(alphav
out[27] += 0.3061862178478971*(alphax[9]*f[31]+alphax[4]*f[30]+alphay[4]*f[29]+(alphay[2]+alphax[1])*f[2
lphax[7])*f[31]+(alphav[18]+alphay[16]+alphax[3])*
[17]*f[30]+(alphav[4]+alphax[1])*f[29]+alphav[19]*
out[30] += 0.3061862178478971*((alphav[9]+alphay[6])*f[31]+(alphav[4]+alphay[2])*f[30]+(alphav[17]+alpha
out[31] += 0.3061862178478971*((alphav[4]+alphay[2]+alphax[1])*f[31]+(alphav[9]+alphay[6]+alphax[0])*f[3
```

5D piecewise linear basis = 32 basis functions

$$\mathtt{out}_i = \sum_{j,k} \overleftrightarrow{T}_{ijk} \cdot \overrightarrow{\alpha}_j f_k \quad \Rightarrow$$

163,840 multiplications,
98,304 additions

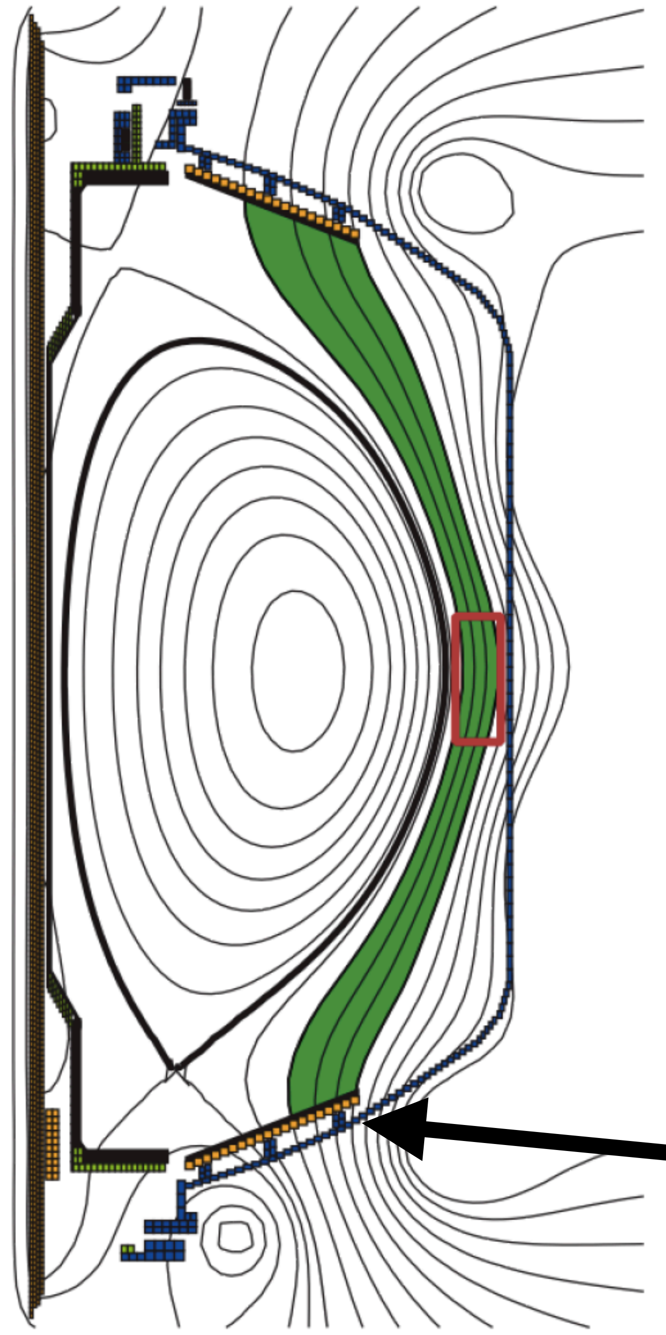
540 multiplications,
608 additions

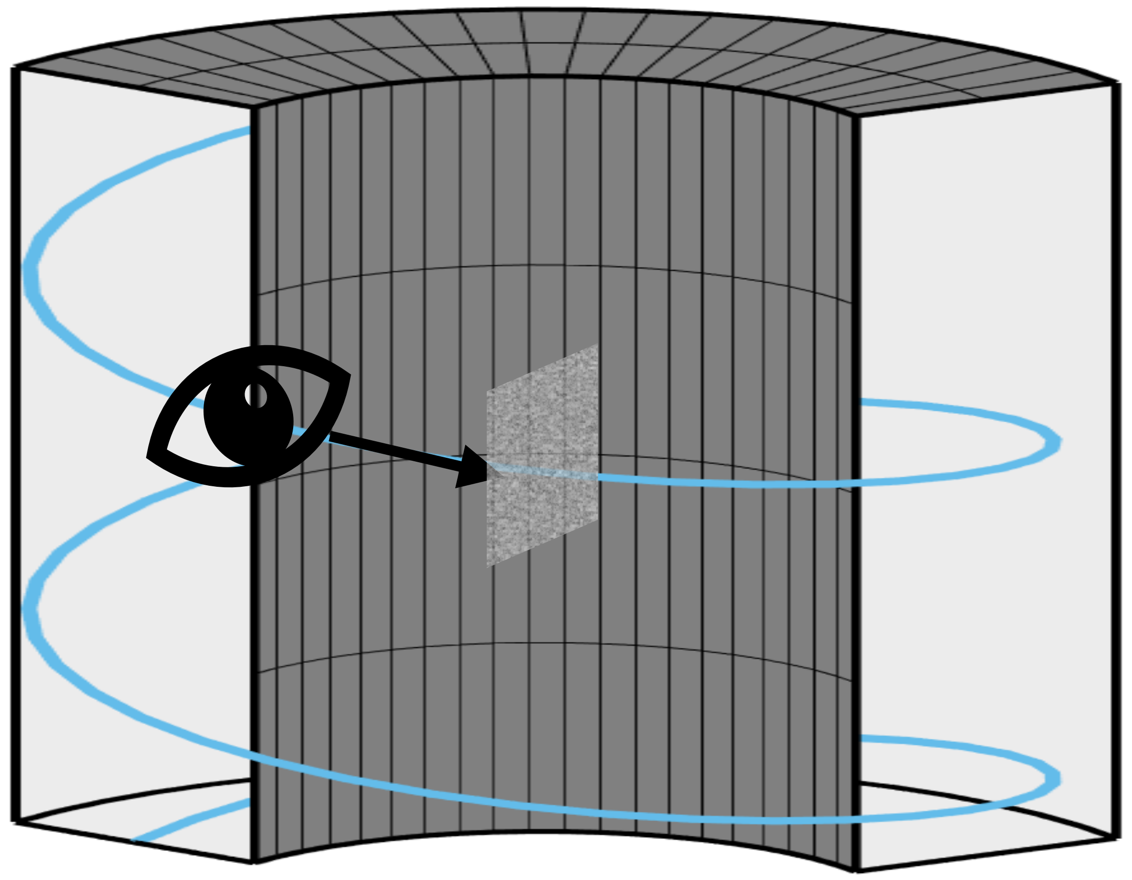
Code is ~30x faster than old nodal version w/ quadrature!

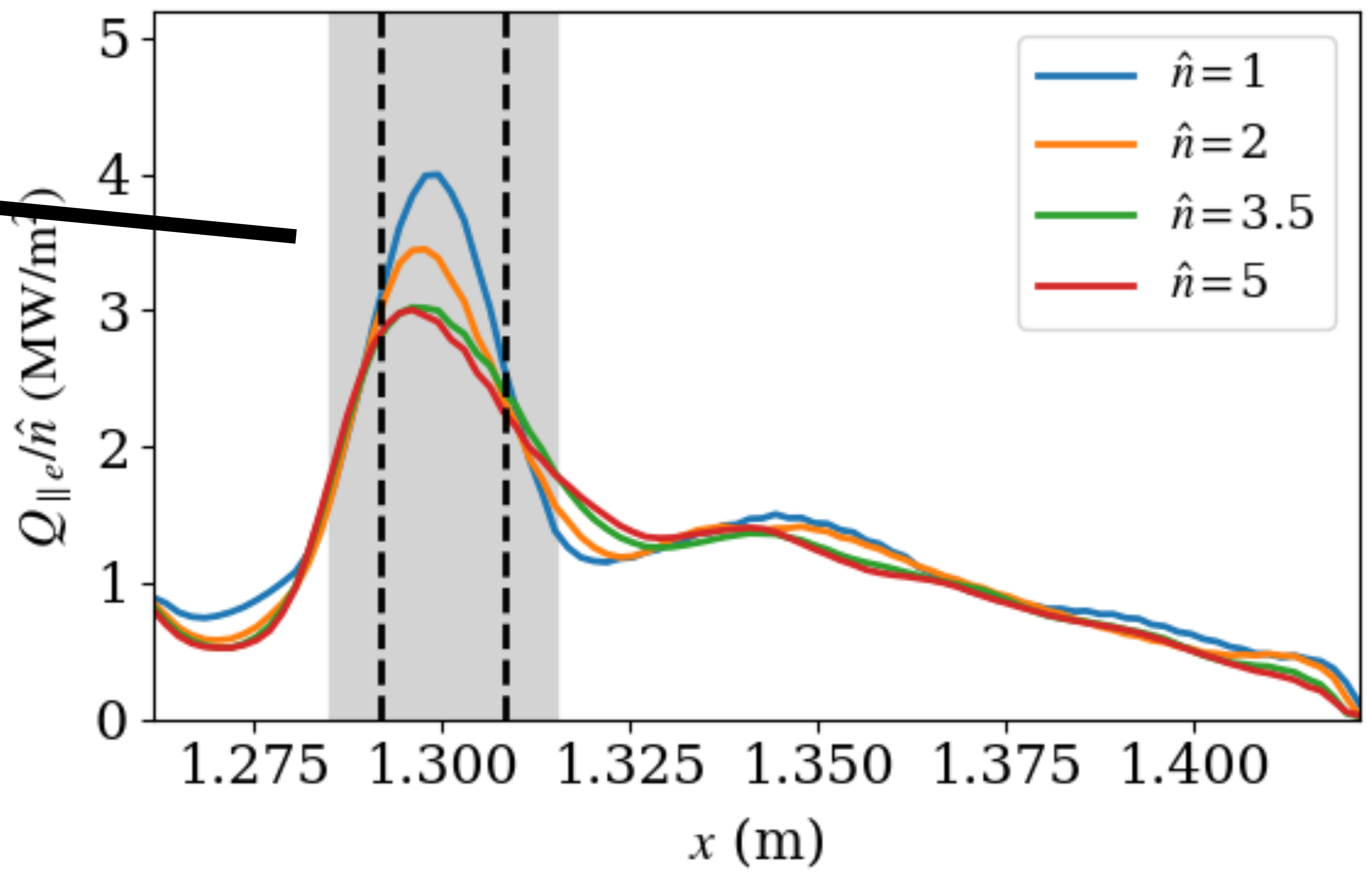
- Maxima generates thousands of lines of machine-written C code… no loops!
  - Takes advantage of significant sparsity in tensor contractions
- High arithmetic intensity is ideal for modern computing architectures
  - GPU version of Gkeyll nearly finished!

Blobby turbulence in the tokamak boundary

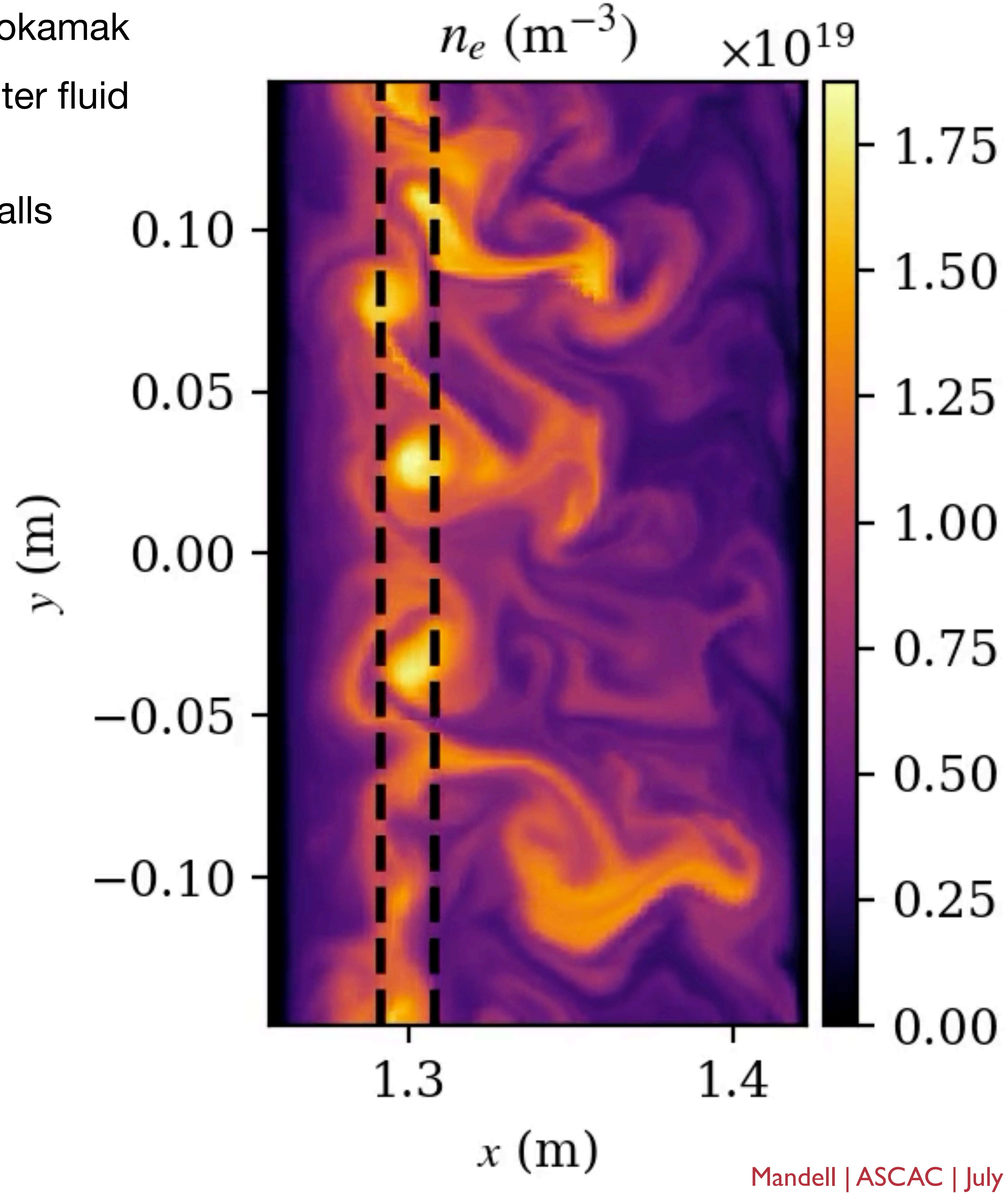
- Blobs of plasma propagate radially outwards in boundary region of tokamak
- Similar to Rayleigh-Taylor instability of heavy fluid falling through lighter fluid (flipped on its side)
- Turbulence can spread out heat flux “footprint” on device material walls





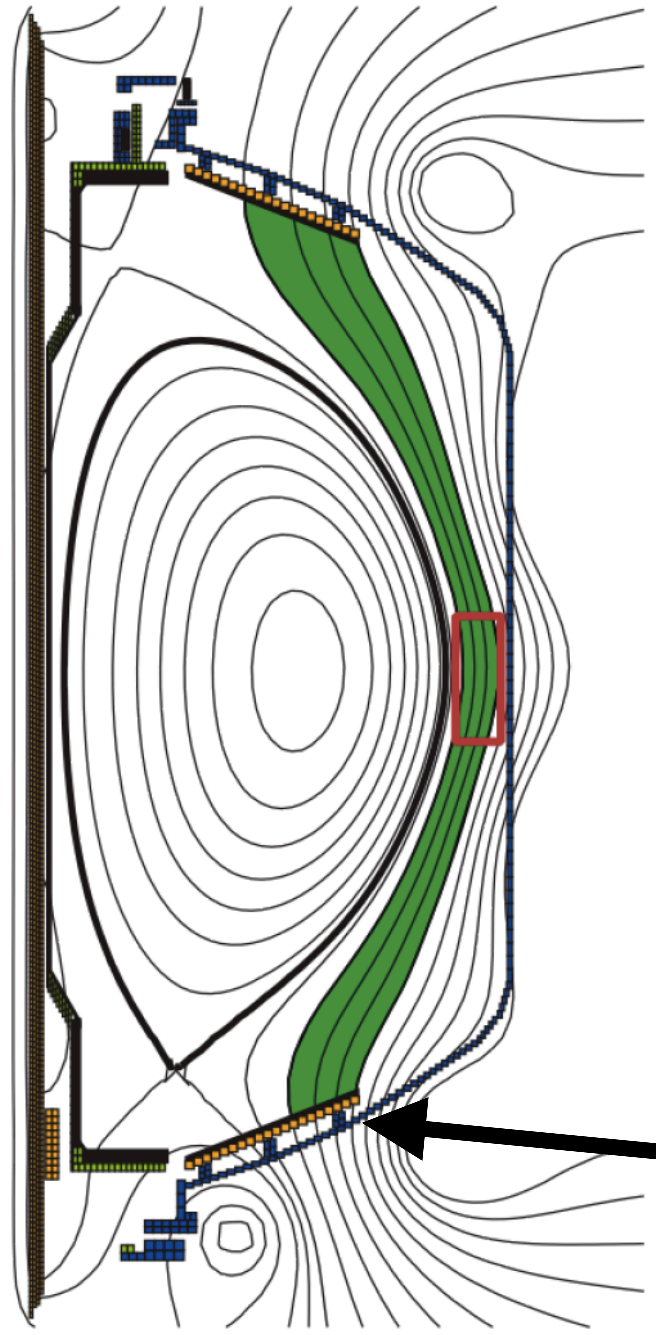


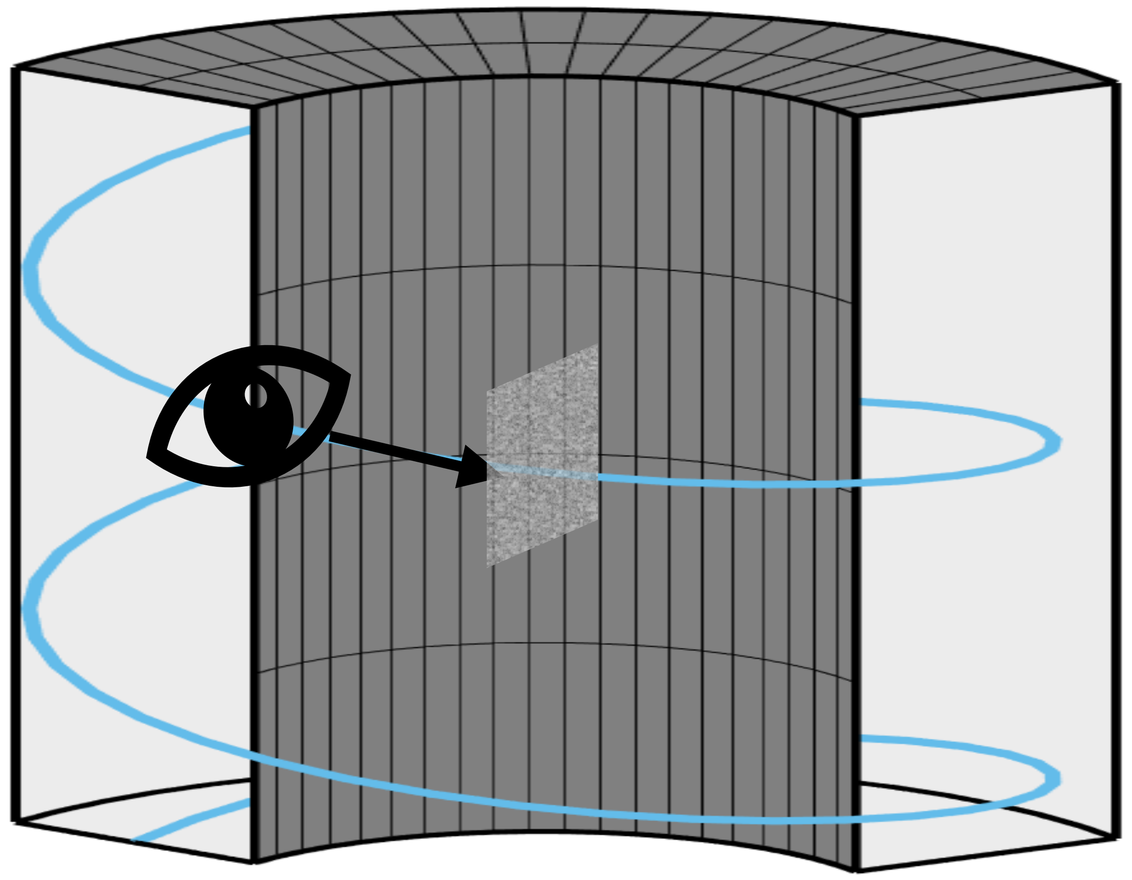
heat “footprint” on walls

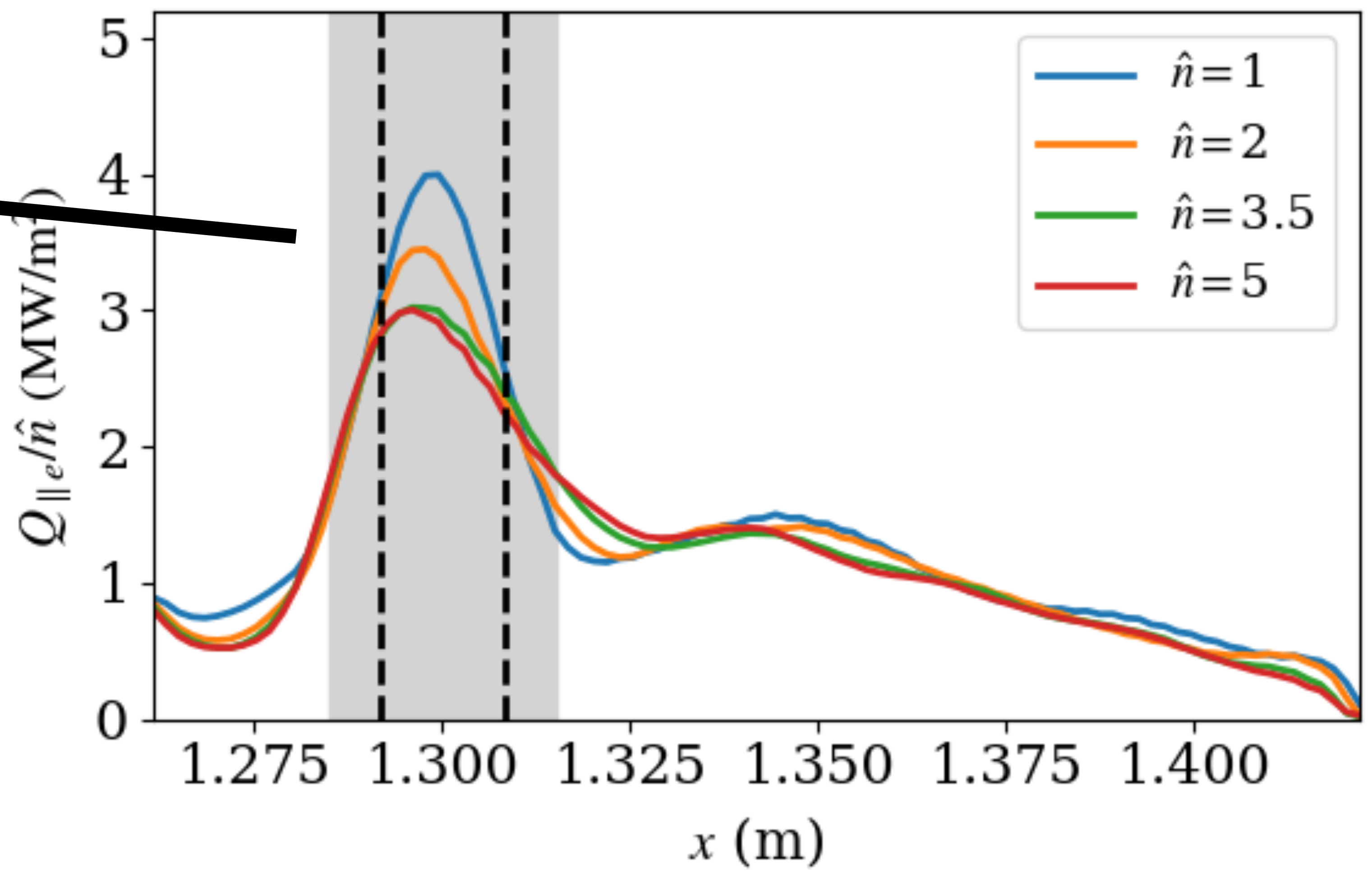


Blobby turbulence in the tokamak boundary

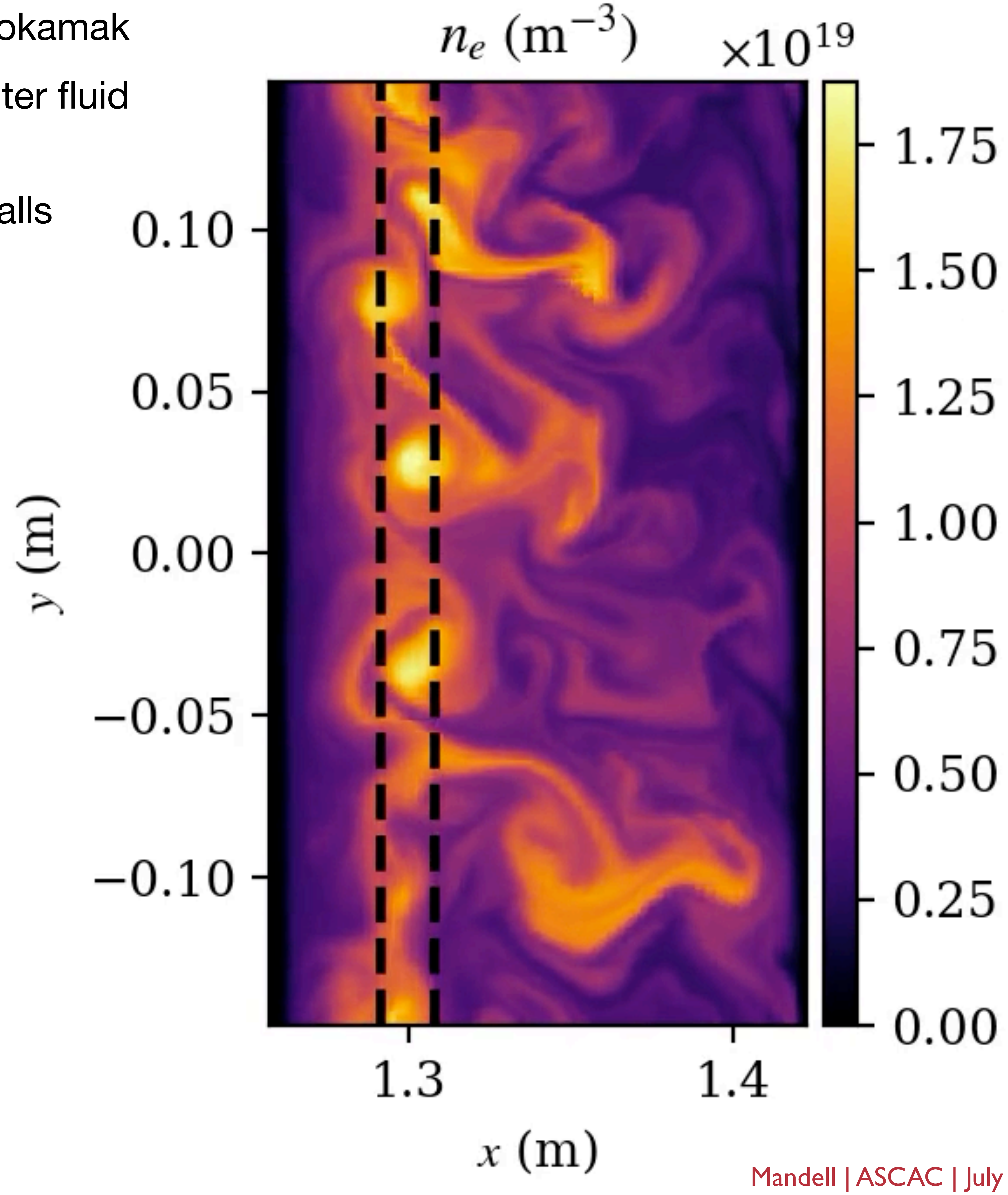
- Blobs of plasma propagate radially outwards in boundary region of tokamak
- Similar to Rayleigh-Taylor instability of heavy fluid falling through lighter fluid (flipped on its side)
- Turbulence can spread out heat flux “footprint” on device material walls



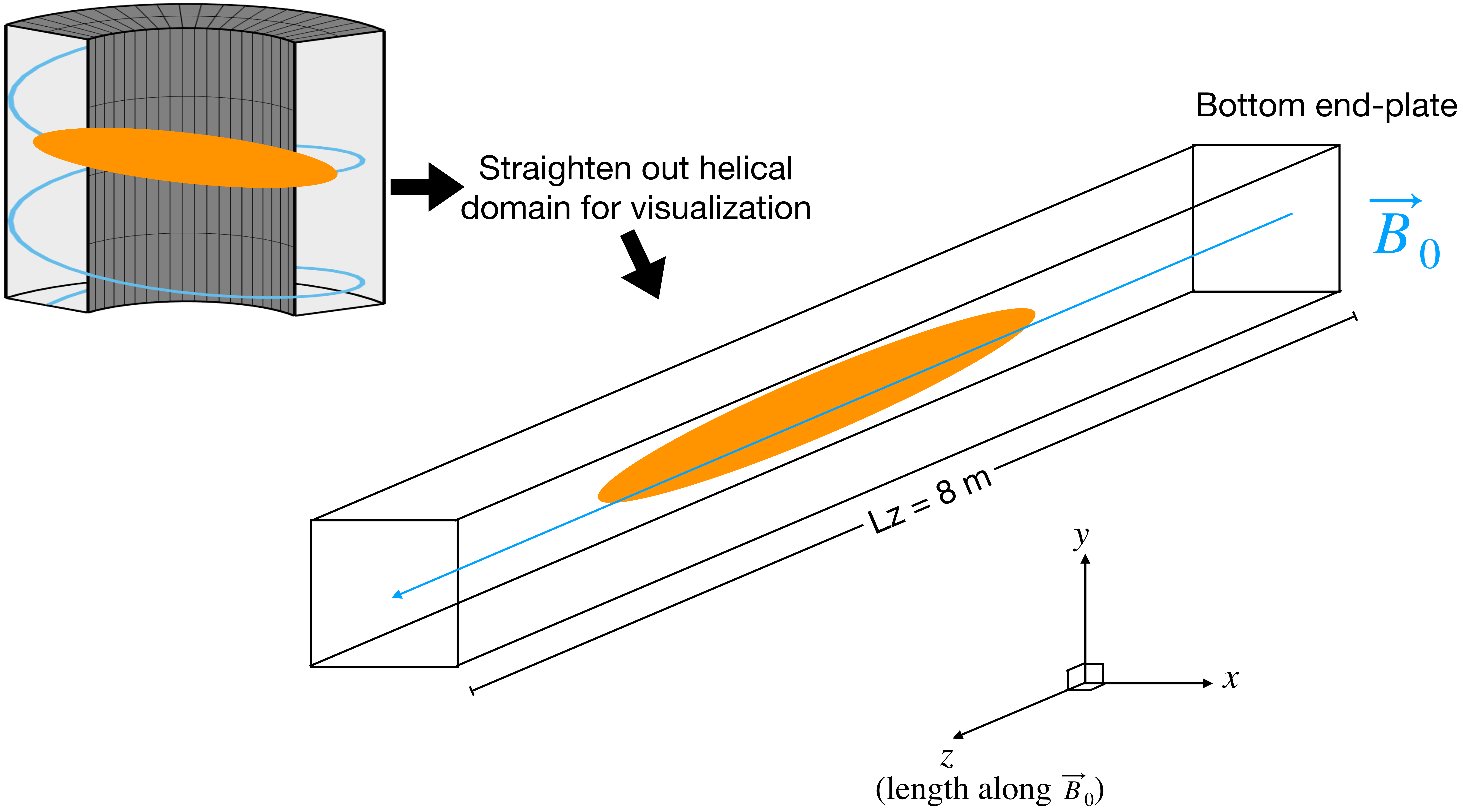




heat “footprint” on walls

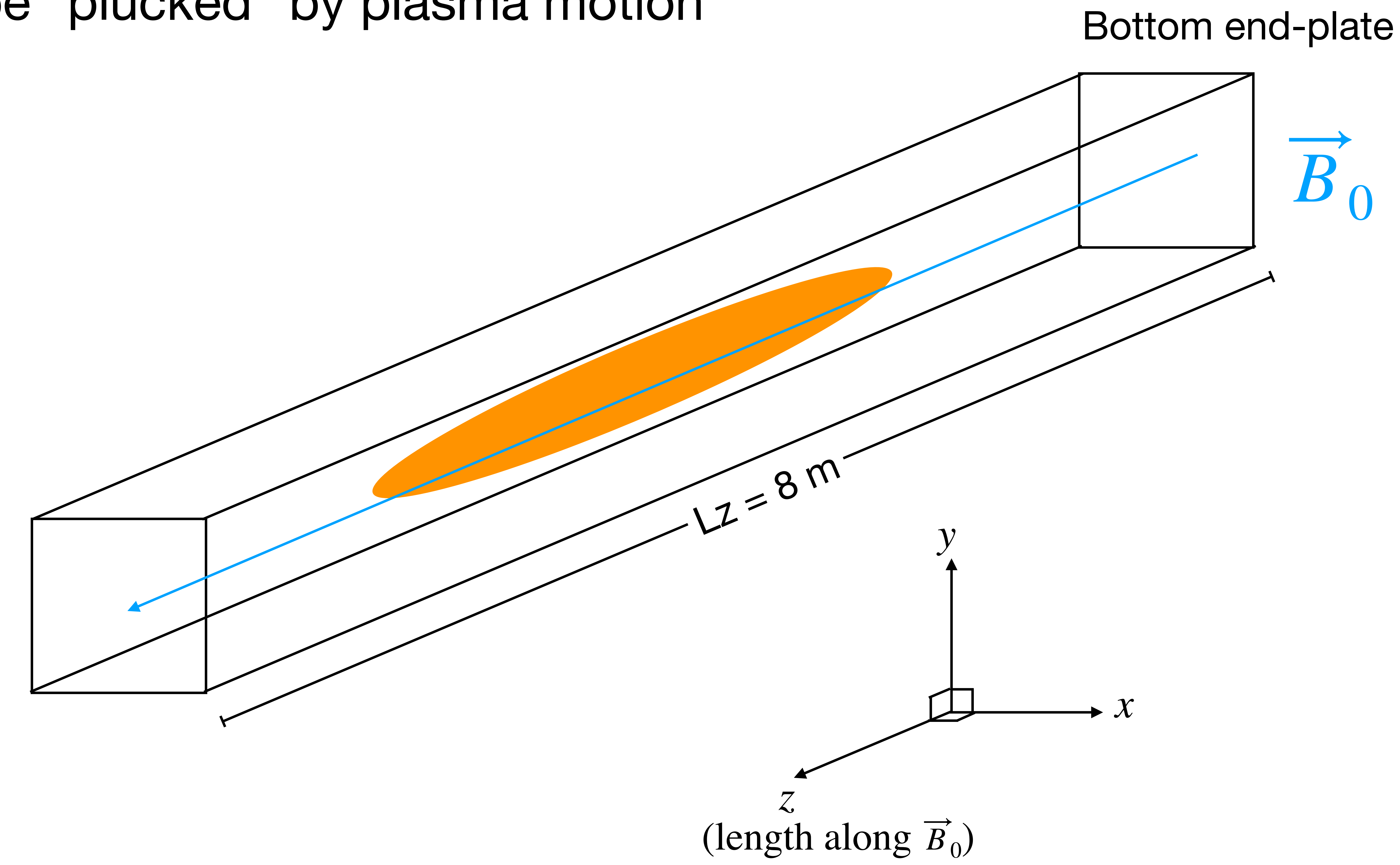


Interaction between magnetic field lines and plasma



Interaction between magnetic field lines and plasma

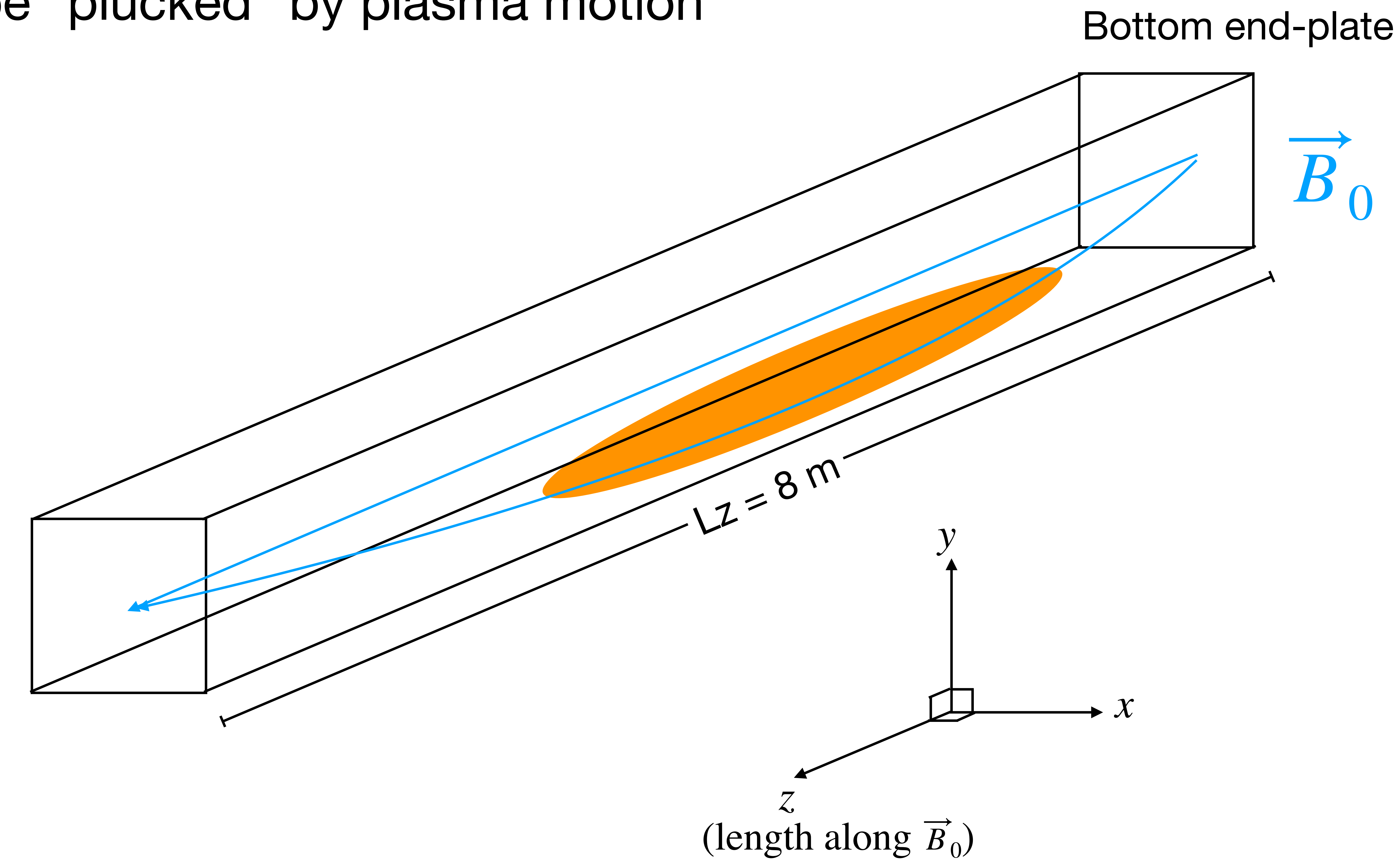
- Magnetic field lines in plasma behave like taut strings
- The field lines can be "plucked" by plasma motion



Interaction between magnetic field lines and plasma

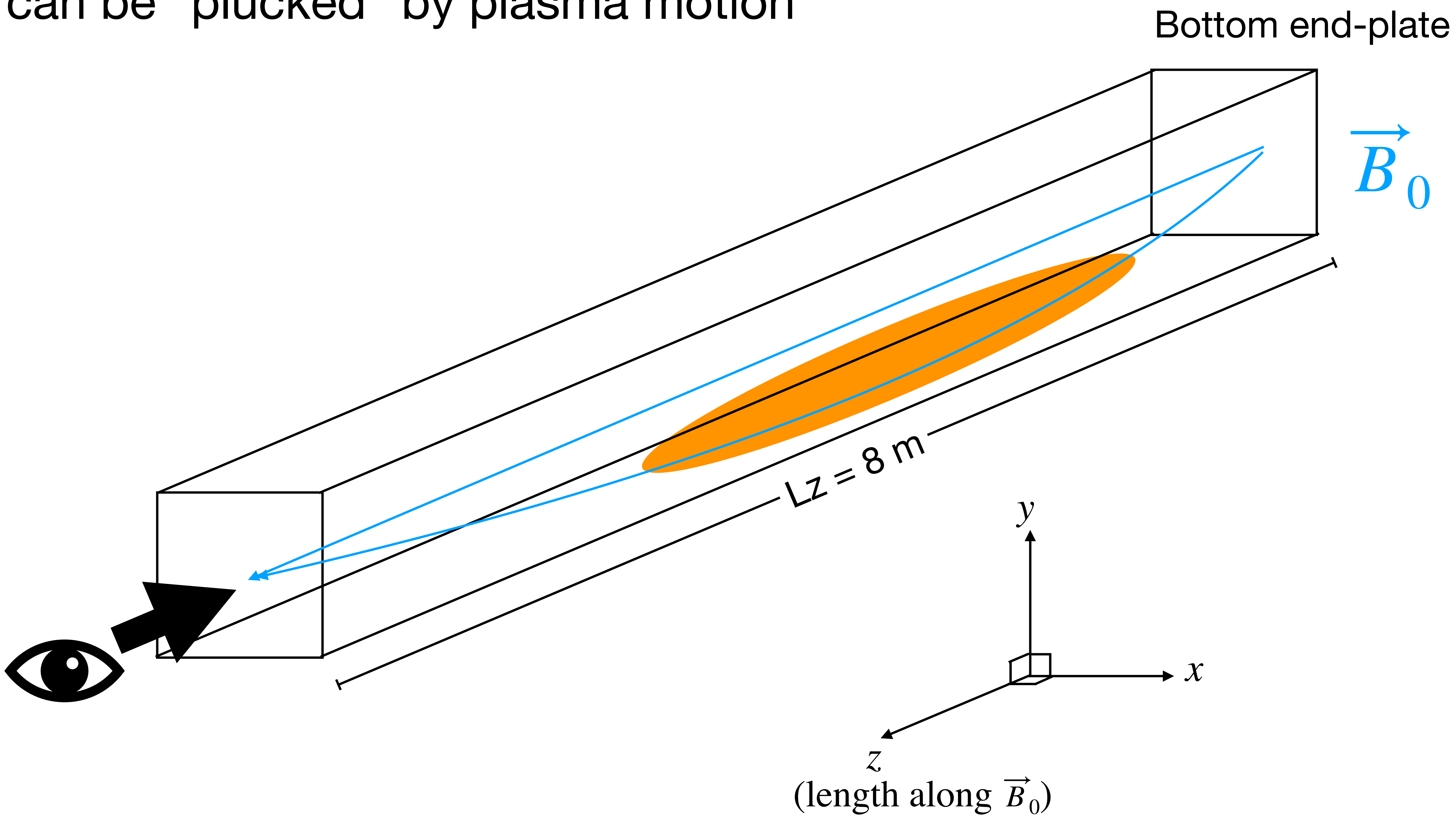
- Magnetic field lines in plasma behave like taut strings

- The field lines can be "plucked" by plasma motion



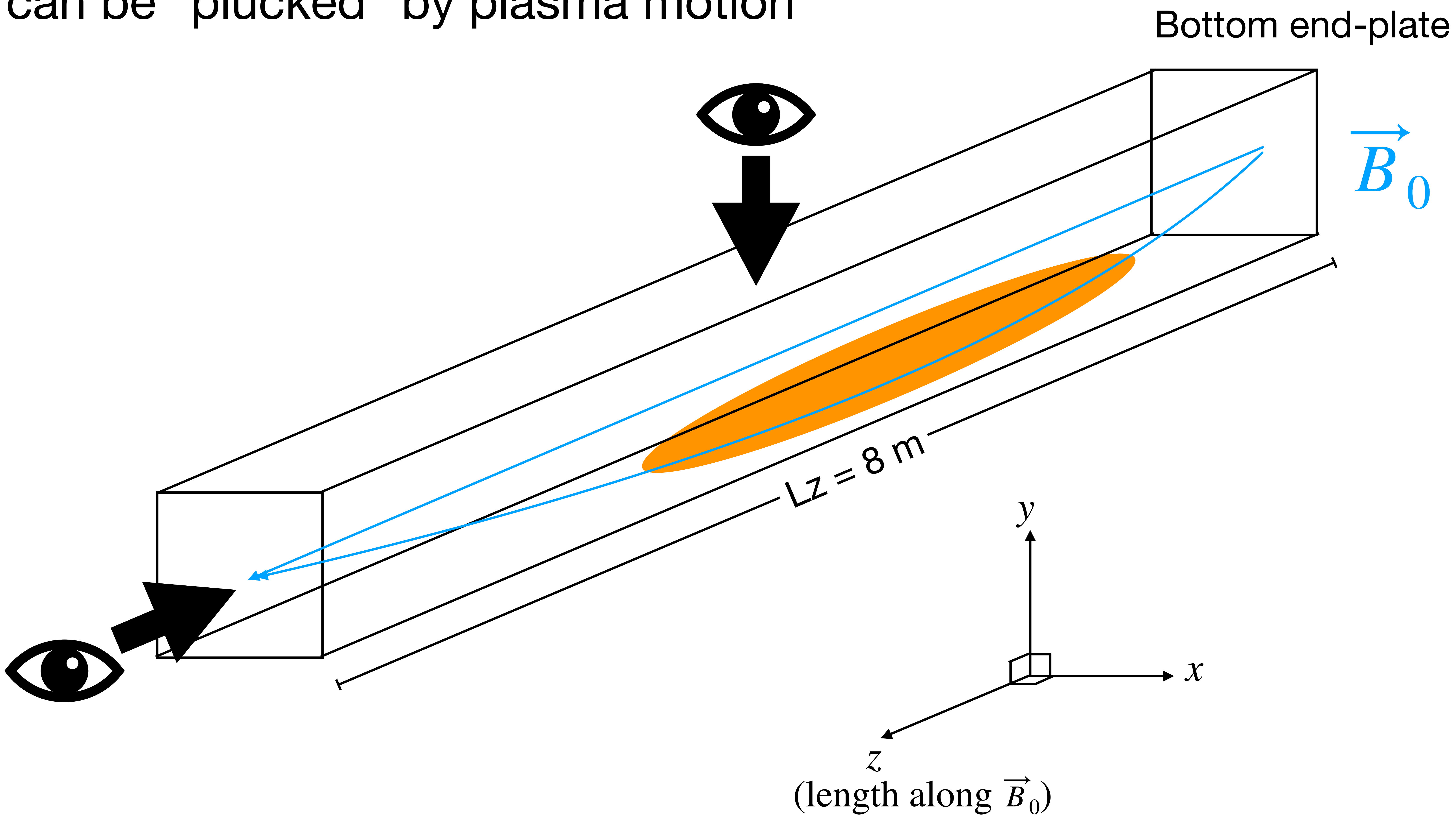
Interaction between magnetic field lines and plasma

- Magnetic field lines in plasma behave like taut strings
- The field lines can be "plucked" by plasma motion

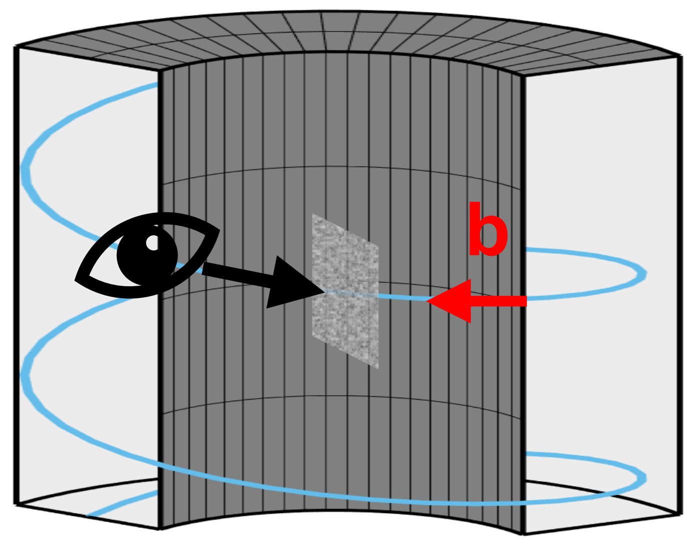


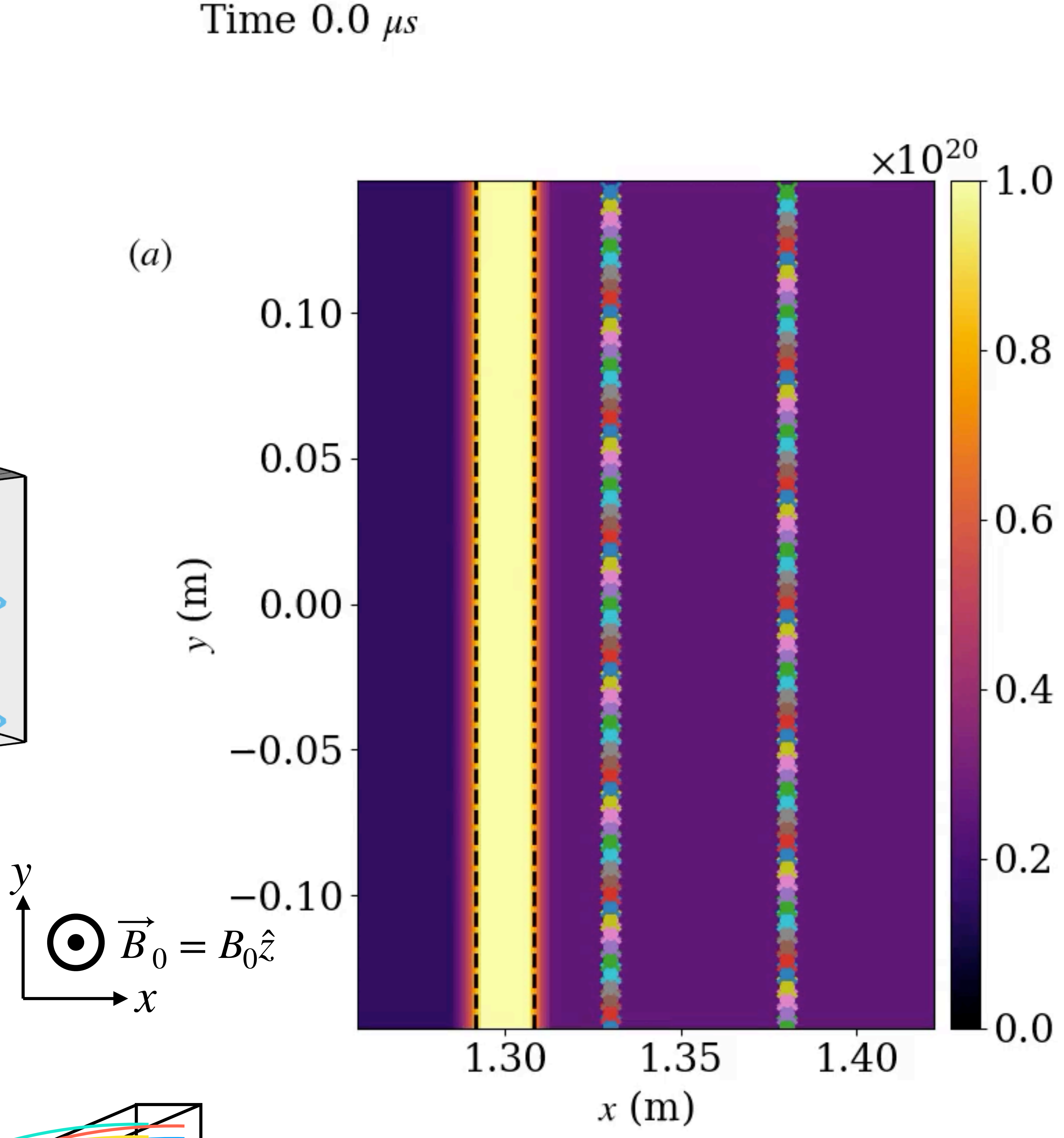
Interaction between magnetic field lines and plasma

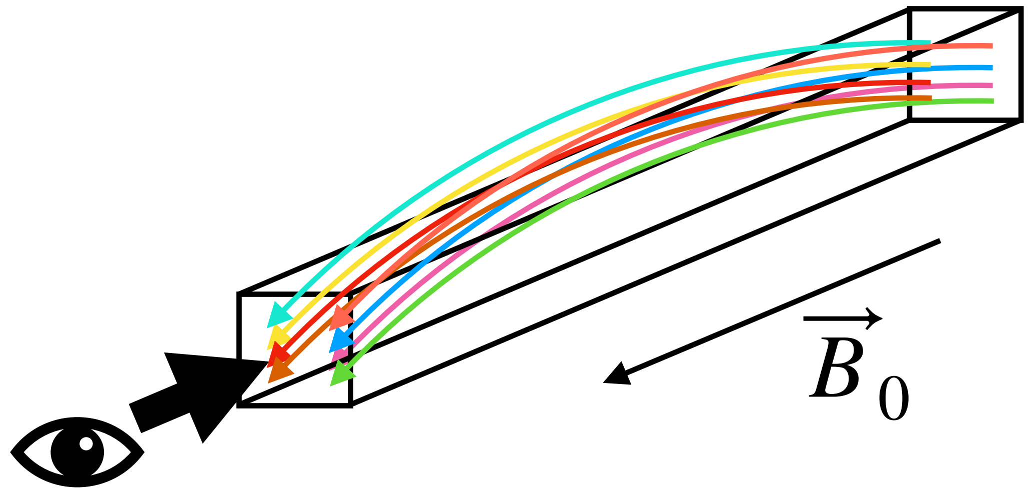
- Magnetic field lines in plasma behave like taut strings
- The field lines can be "plucked" by plasma motion

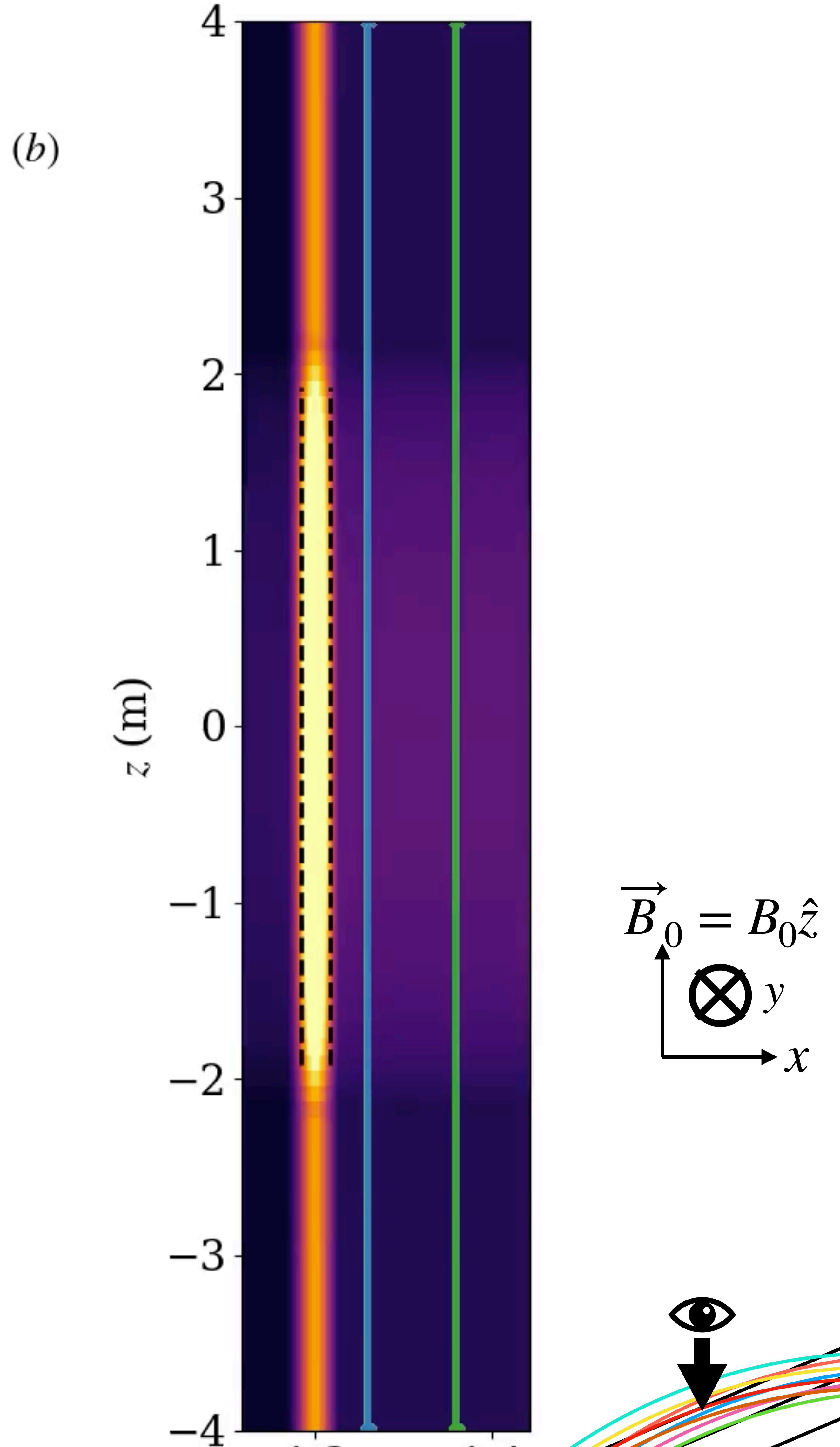


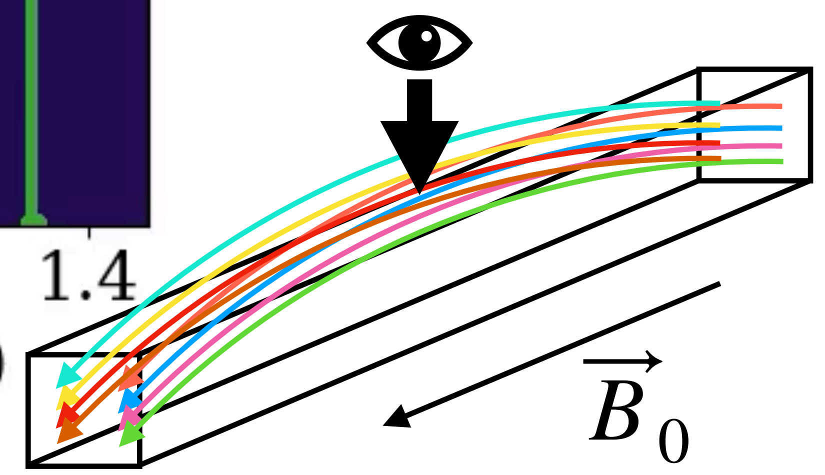
Visualizing Gkeyll's first-of-a-kind electromagnetic capabilities





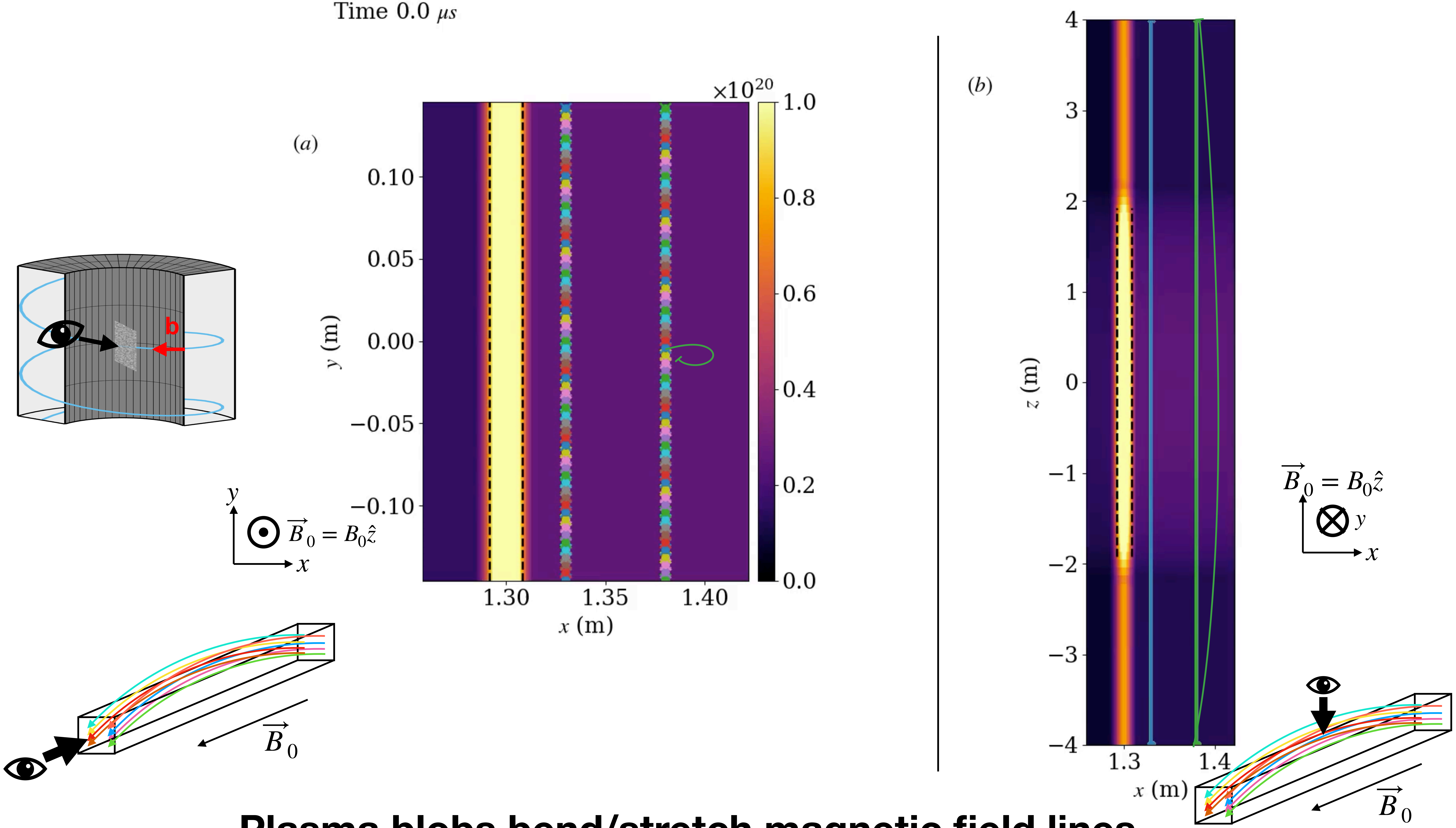






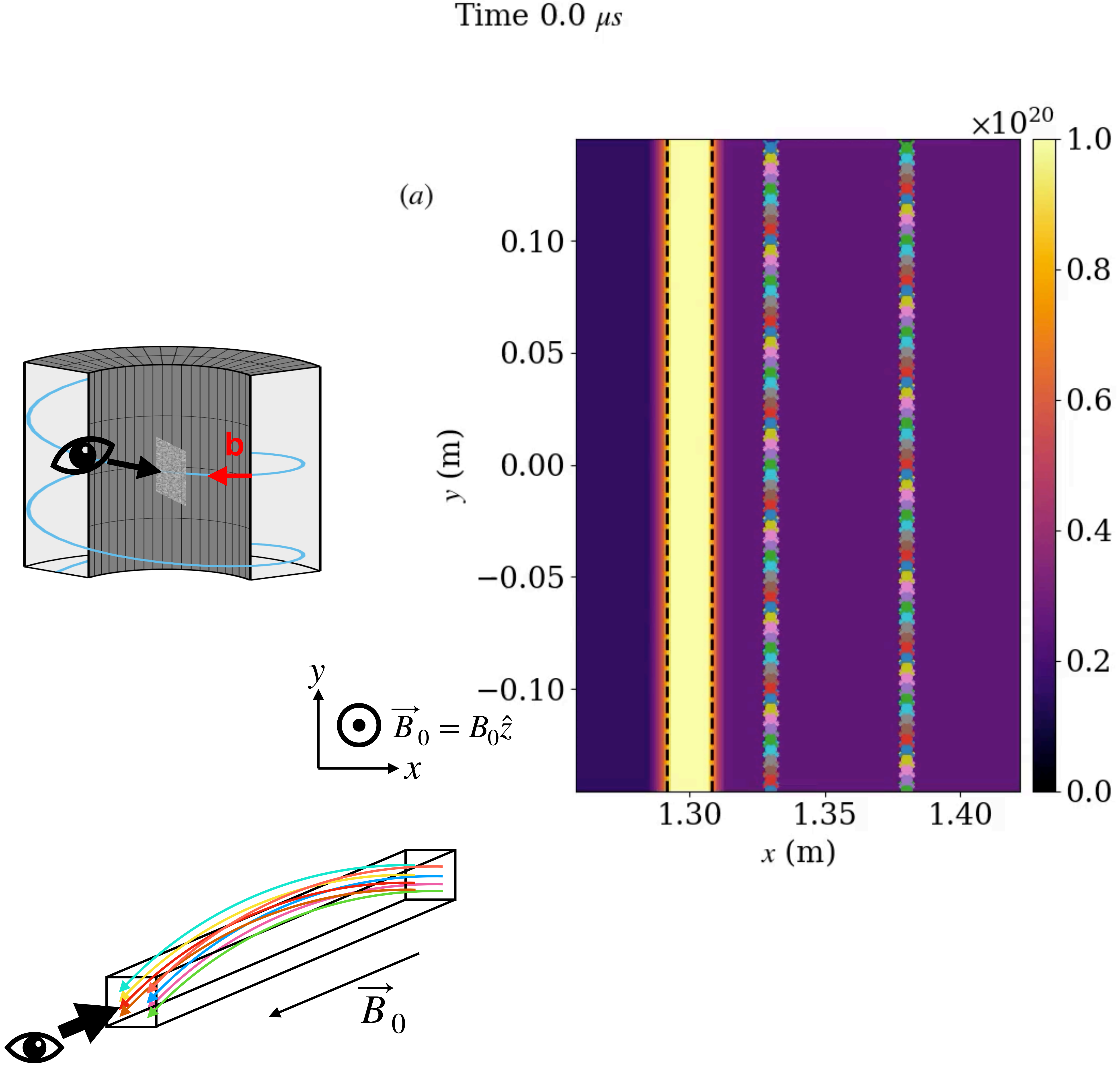
Plasma blobs bend/stretch magnetic field lines

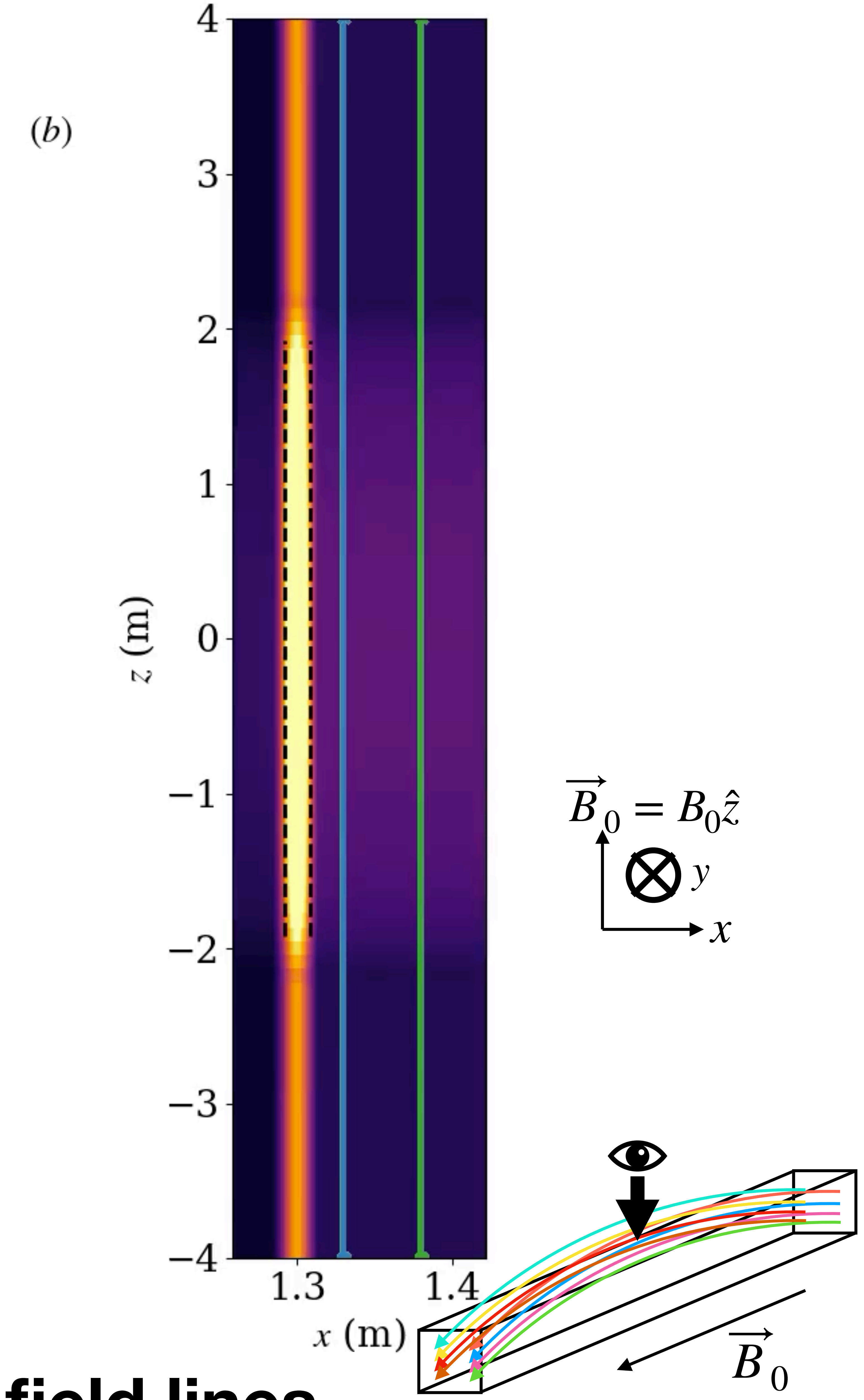
Visualizing Gkeyll's first-of-a-kind electromagnetic capabilities



Plasma blobs bend/stretch magnetic field lines

Visualizing Gkeyll's first-of-a-kind electromagnetic capabilities

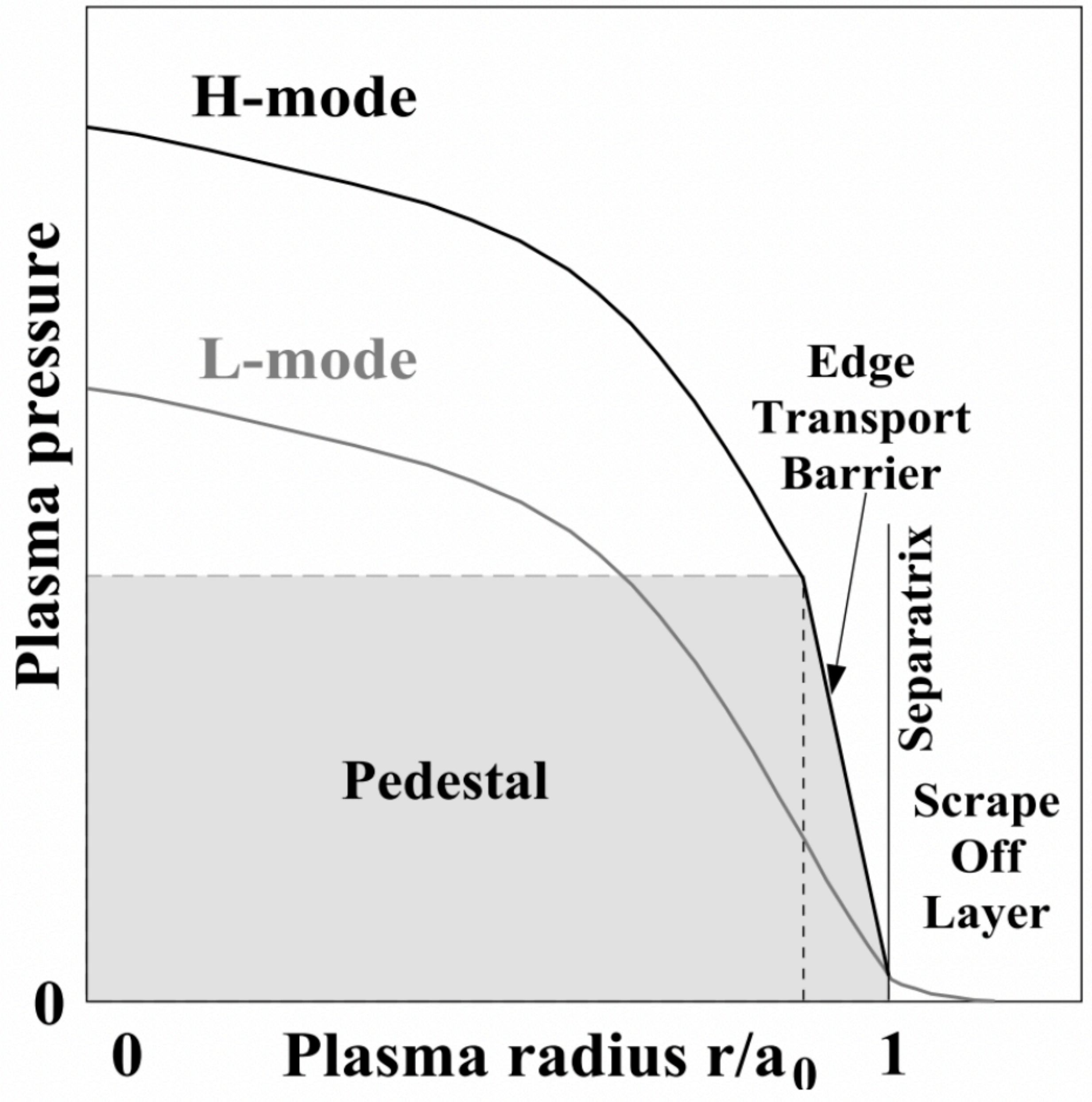


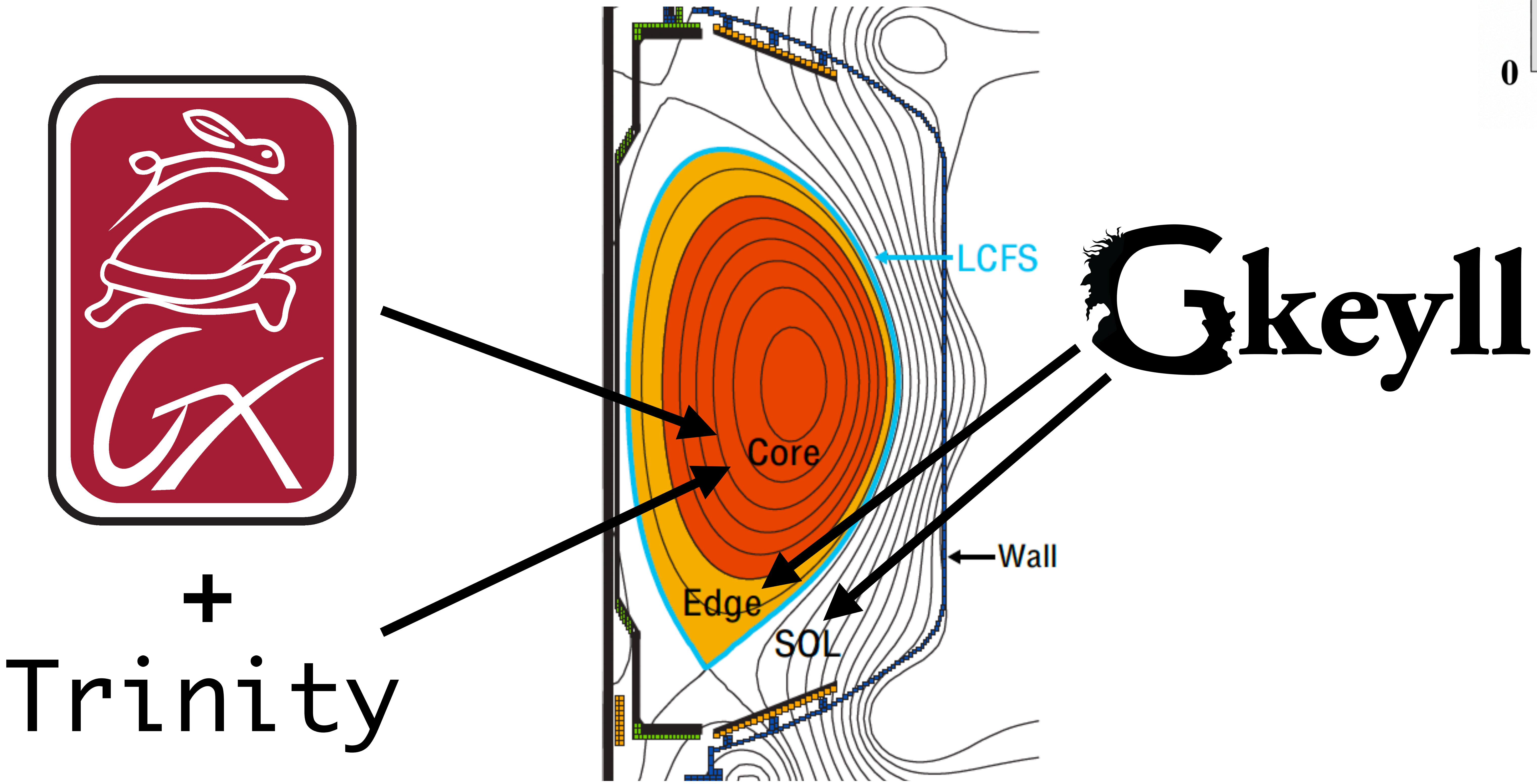


Plasma blobs bend/stretch magnetic field lines

A whole-device transport model

- Using GX+Trinity in the core and Gkeyll in the boundary, we have a whole-device transport model
  - Will be able to directly study impact of changes in core confinement on boundary exhaust
  - Will be able to directly study impact of boundary exhaust solutions (like detachment) on core confinement
  - As we add more physics to the models, we can achieve true **predict-first modeling capability**





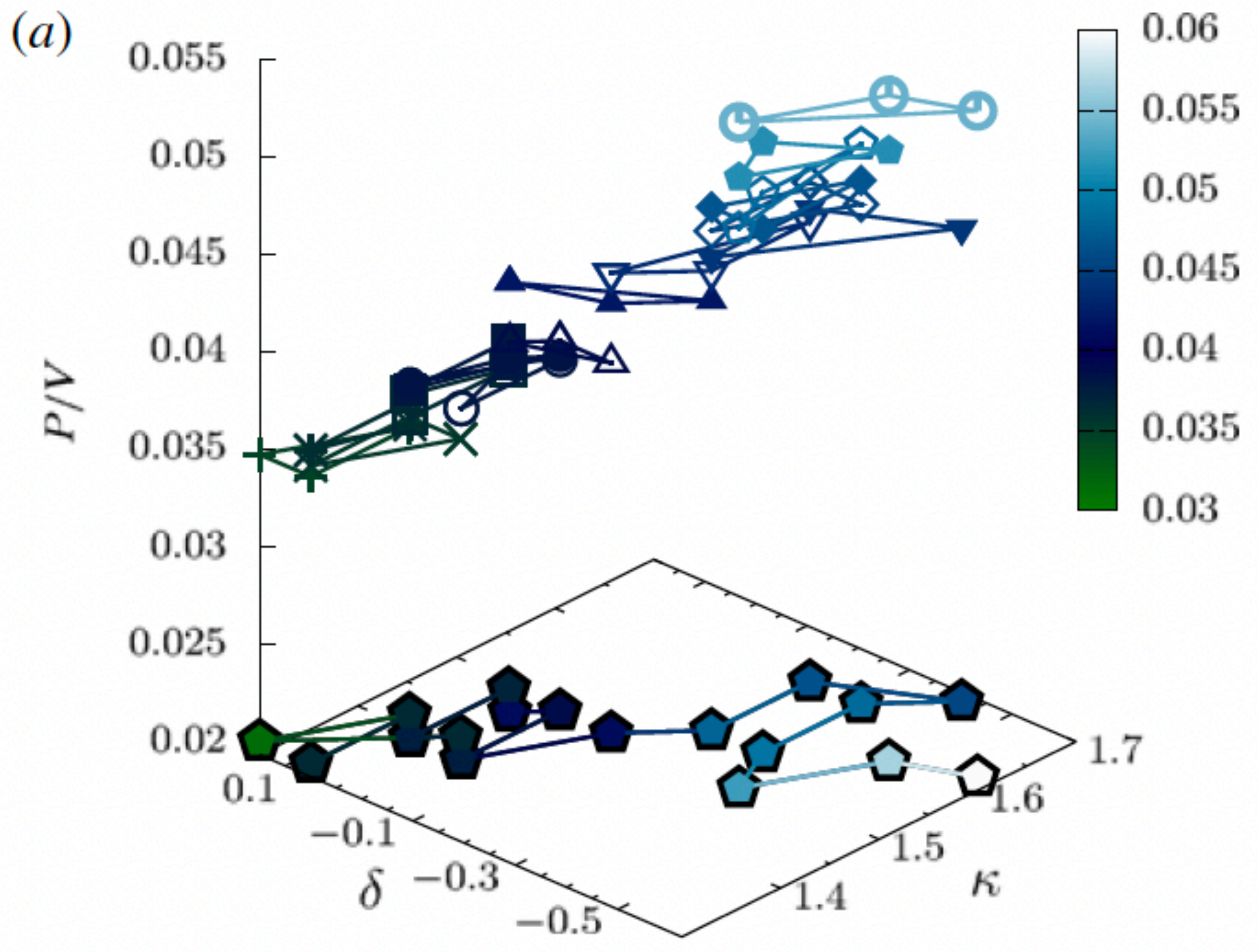
The goal: whole-device transport optimization

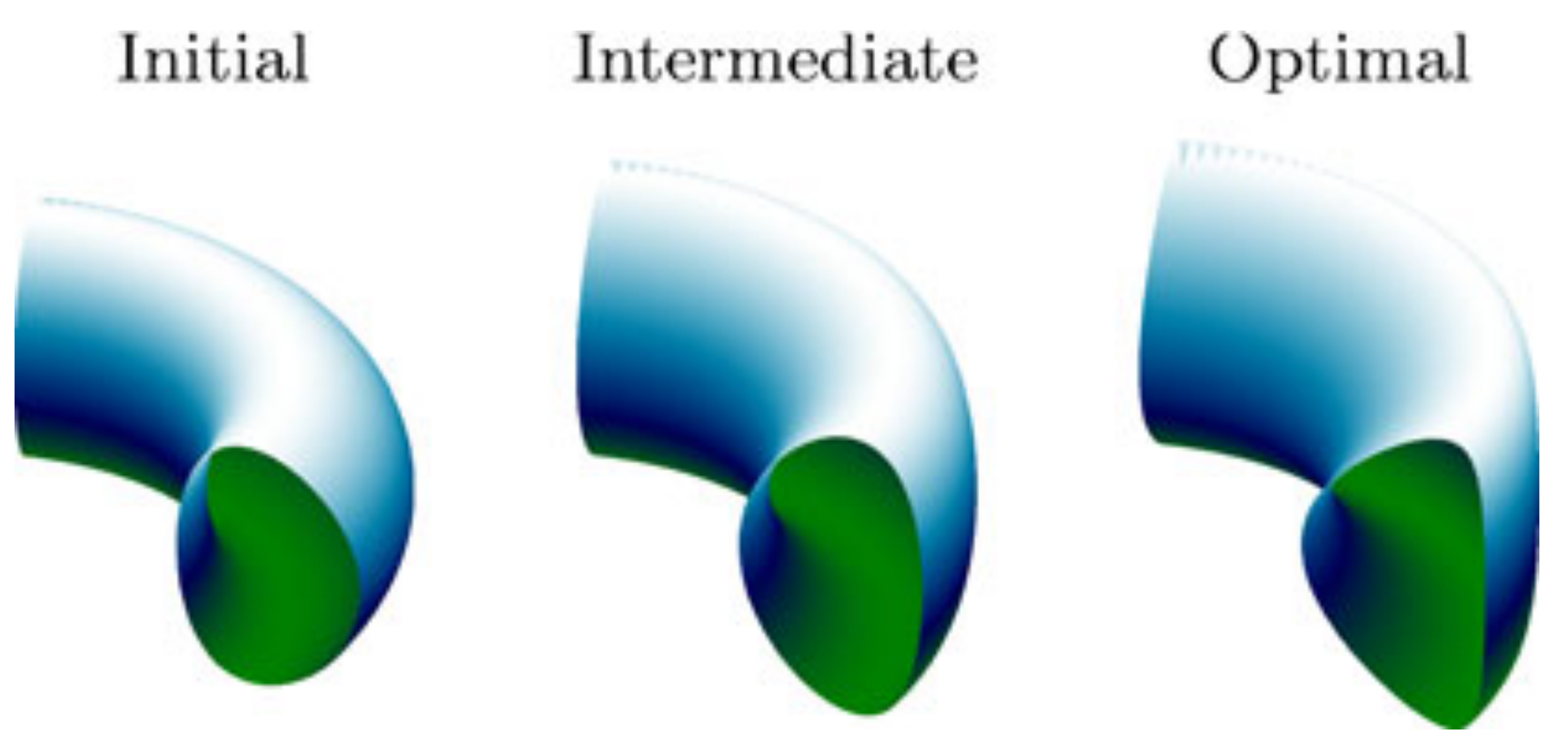
The goal: whole-device transport optimization

- Can we do **optimization at exascale?**
  - Need to ensure that a whole-device model calculation is sub-exascale (petascale?)
    - Requires both algorithmic advancements and effective use of latest hardware, like GPUs
  - Each whole-device calculation can compute key figures of merit to be optimized for a fusion reactor design
    - Total fusion power, energy confinement time, heat load to walls, etc
  - In a tokamak core, there are ~20 shaping parameters that could be varied simultaneously; in the boundary, could start with a few candidate divertor configurations
  - Parallel optimization algorithm could run O(10)-O(100) shapes simultaneously
  - Can use a hierarchy of models of varying speed and accuracy (incl. machine learning models) to progressively narrow design space
  - Could also build in economic (e.g. cost) and safety/environmental factors (e.g. minimize tritium onsite)

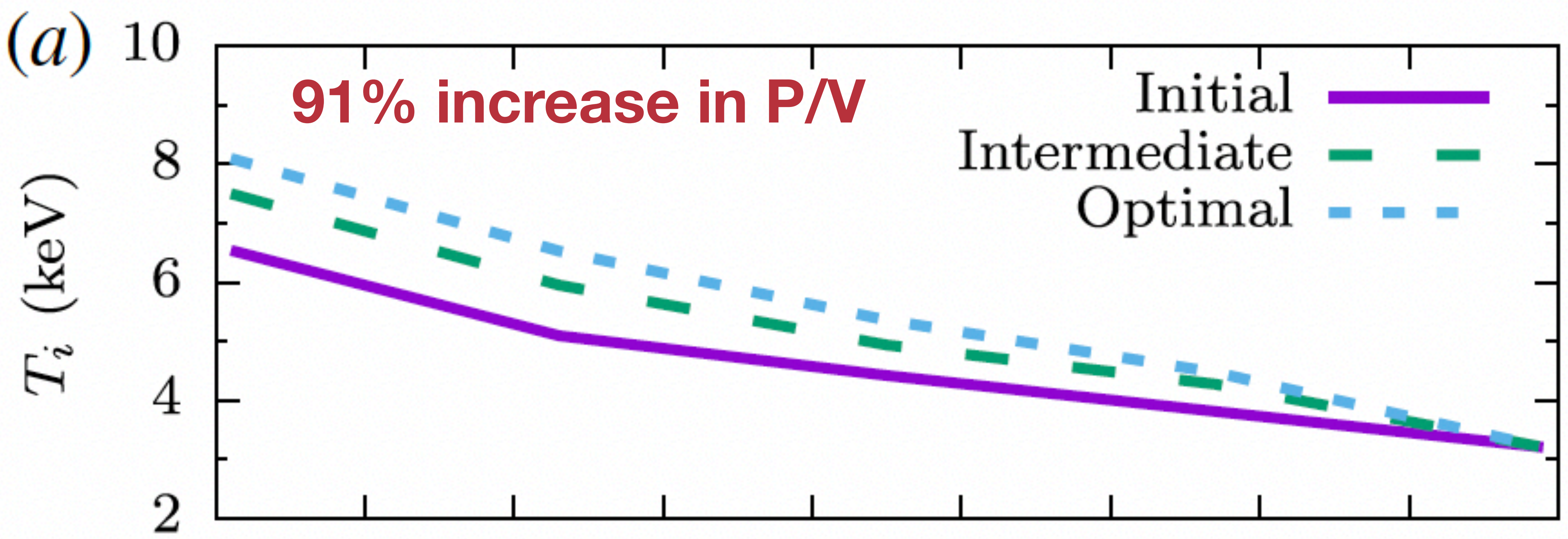
Transport optimization is possible!

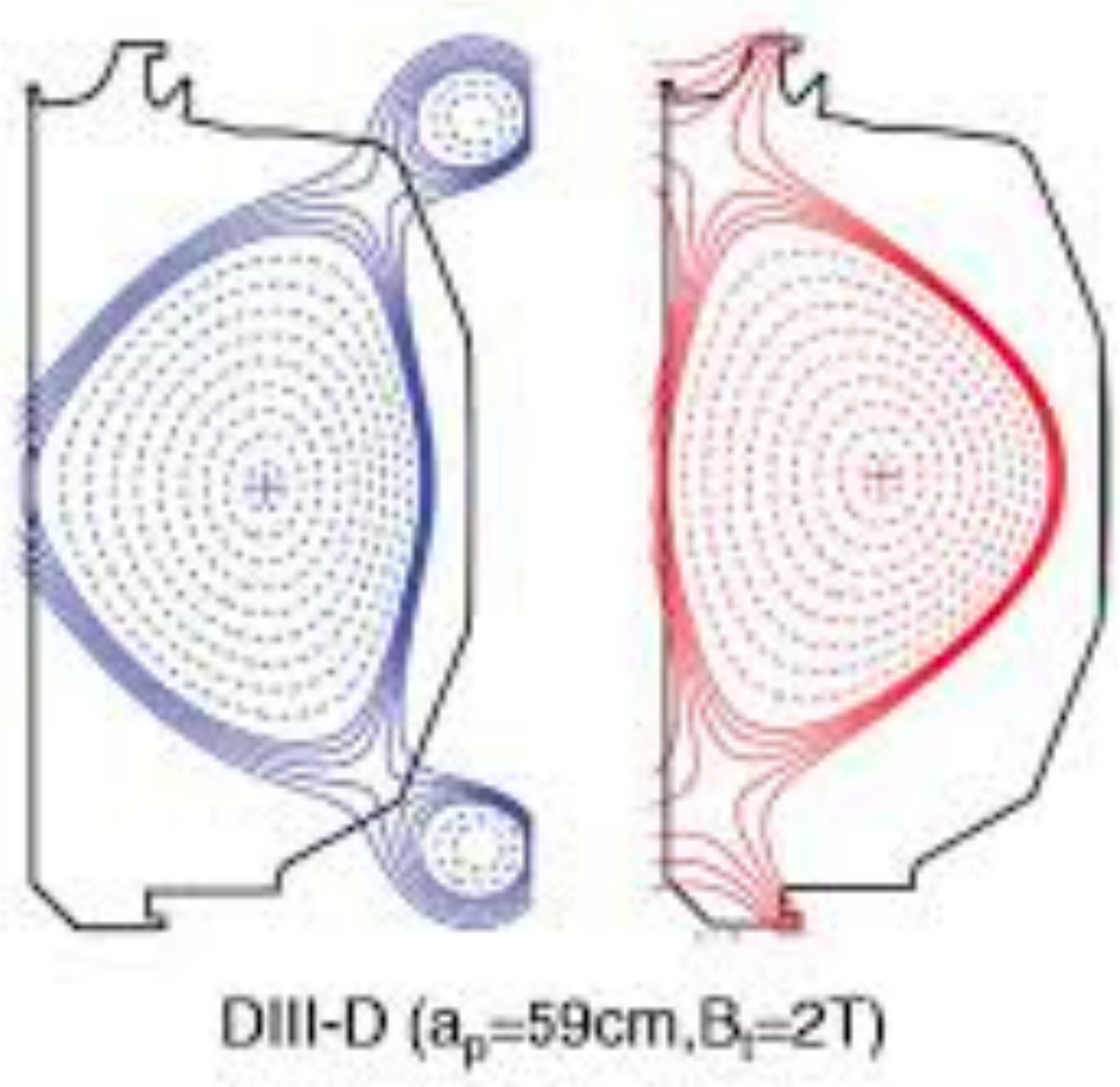
- Preliminary design study (**Highcock, Mandell, Barnes, Dorland (2018)**) found **negative triangularity** to be optimal in core
  - Used a lower-fidelity precursor to GX, coupled to Trinity
  - **18 shape design evaluations = 8680 nonlinear calculations = 3000 GPU node hours**
  - **91% improvement in fusion power per unit volume from going to negative triangularity**
  - Consistent with recent DIII-D experimental results showing confinement improvement with negative triangularity





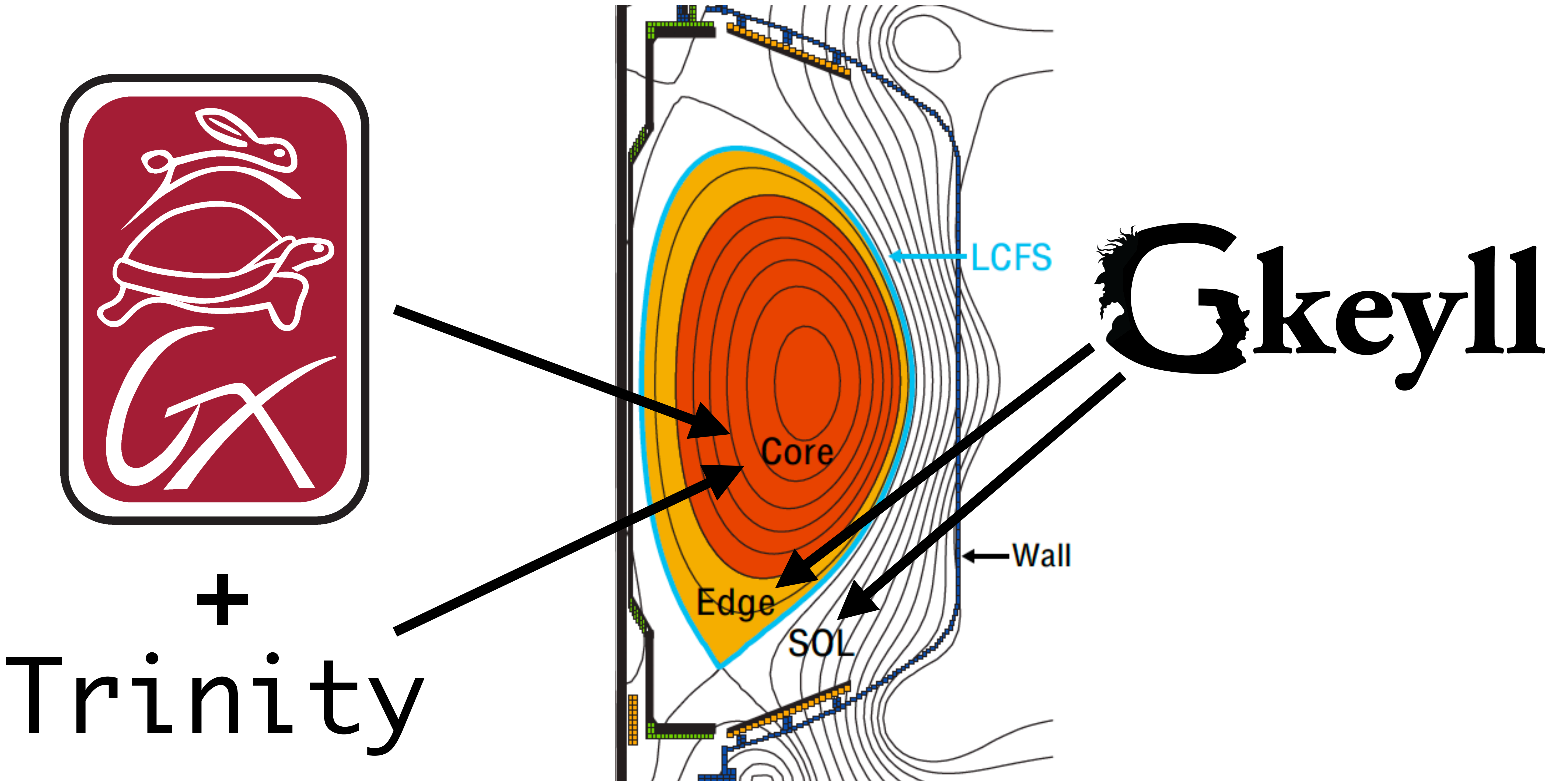
Highcock, Mandell, Barnes, Dorland (2018)





Summary

- I have presented a whole-device framework for modeling turbulence and transport in fusion reactors
  - **GX** models **core turbulence** with **spectral methods** on **GPUs**, and couples to a transport solver like **Trinity** to form a **multi-scale core transport model**
  - **Gkeyll** models **boundary turbulence** with **discontinuous Galerkin methods** and a first-of-a-kind **kinetic** scheme that includes **magnetic fluctuations**
- A number of key advances have been made, spanning theory, algorithms, and hardware
- Still work to do, but we can achieve game-changing **whole-device transport modeling and optimization** to **design better fusion reactors**



Acknowledgements

This work has benefitted from funding from DOE CSGF, the DOE SciDAC program, and the DOE Fusion Energy Sciences Postdoctoral Fellow program, administered by ORISE.

Massachusetts Institute of Technology

Oak Ridge Institute for Science and Education

PPPL Princeton Plasma Physics Laboratory

Princeton University

GX group:

- **William Dorland**
- Ian Abel
- Nate Barbour
- Braden Buck
- Sarah Fischer
- Rahul Gaur
- Patrick Kim
- Matt Landreman
- Tony Qian
- … and others!

Gkeyll group:

- **Ammar Hakim**
- **Greg Hammett**
- Tess Bernard
- Petr Cagas
- Mana Francisquez
- Jimmy Juno
- … and others!

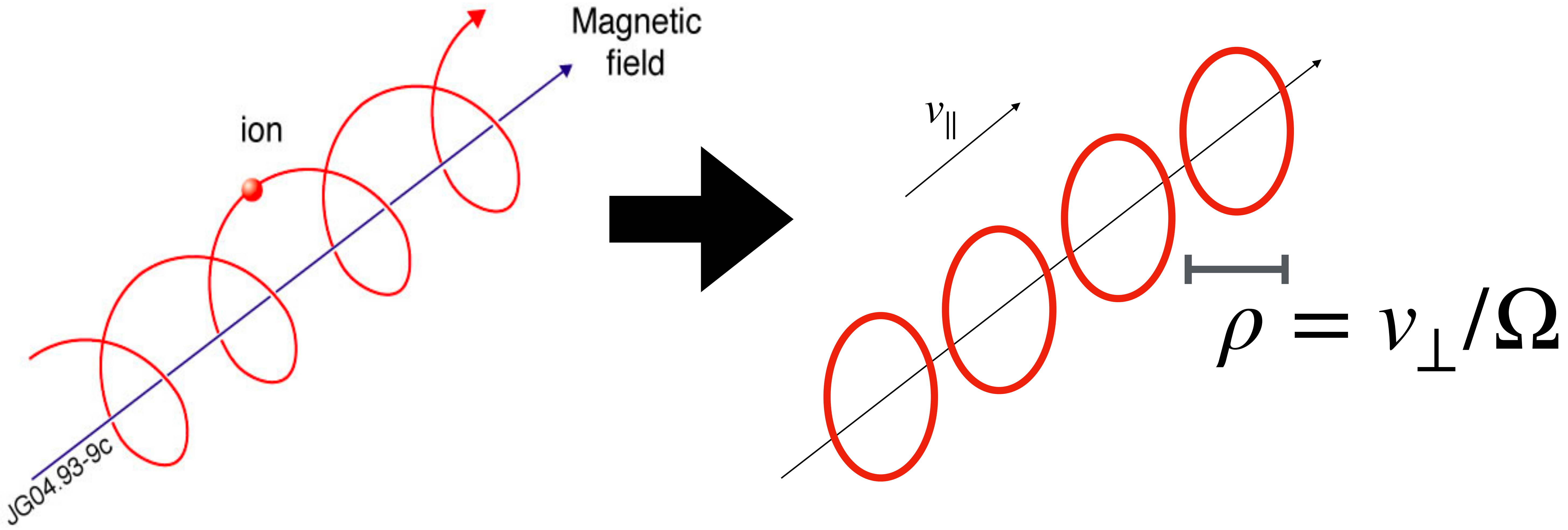
https://github.com/ammarhakim/gkyl/

https://gkyl.readthedocs.io/

Backup

Modeling turbulence in fusion plasmas

- Turbulence in a fusion plasma (both core and boundary) is well-described by **gyrokinetics**
  - A kinetic description (where particle positions *and* velocities are tracked) is necessary because fusion plasmas are very collisionless
  - A naive kinetic description would involve a 6D phase space, tracking PDF $f(x, y, z, v_x, v_y, v_z)$
    - E.g. Vlasov-Boltzmann-Maxwell system
  - We can reduce the dimensionality by one velocity dimension by averaging out the high frequency particle gyration, tracking PDF $f(x, y, z, v_{||}, v_\perp)$
    - Effectively a transformation from discrete charged particles → rings of charge
    - Still a 5D nonlinear PDE!



Typical turbulence scale length
~ ion gyroradius ρ_i

DG for general Hamiltonian system

$$\frac{\partial f}{\partial t} = \{H, f\}$$

Define phase-space velocity $\vec{\alpha} = \{\vec{Z}, H\}$, write in conservative form as

$$\frac{\partial f}{\partial t} + \nabla_{\vec{Z}} \cdot \left(\vec{\alpha} f\right) = 0$$

DG weak form:

- divide global phase-space domain into cells
- multiply GK eq. by a test function w_i and integrate (by parts) over cell C_m

$$\int_{C_m} d\vec{Z}\ w_i \frac{\partial f}{\partial t} + \oint_{\partial C_m} dS\ w_i^- \widehat{f \vec{\alpha} \cdot \vec{n}} - \int_{C_m} d\vec{Z}\ f \vec{\alpha} \cdot \nabla_{\vec{Z}} w_i = 0$$

- Particle conservation by taking $w = 1$
- Energy conservation by taking $w = H$, requires H to be continuous!

DG for full-f gyrokinetics

- GK is a Hamiltonian system, with $H = \frac{1}{2}mv_\parallel^2 + \frac{1}{2}mv_\perp^2 + q\phi$, and a non-canonical Poisson bracket:

$$\{F,G\} = \frac{\vec{B}^*}{mB_\parallel^*} \cdot \left(\nabla F \frac{\partial G}{\partial v_\parallel} - \frac{\partial F}{\partial v_\parallel}\nabla G\right) - \frac{\hat{b}}{qB_\parallel^*} \times \nabla F \cdot \nabla G$$

- Same DG weak form, recall $\vec{\alpha} = \{\vec{Z}, H\}$:

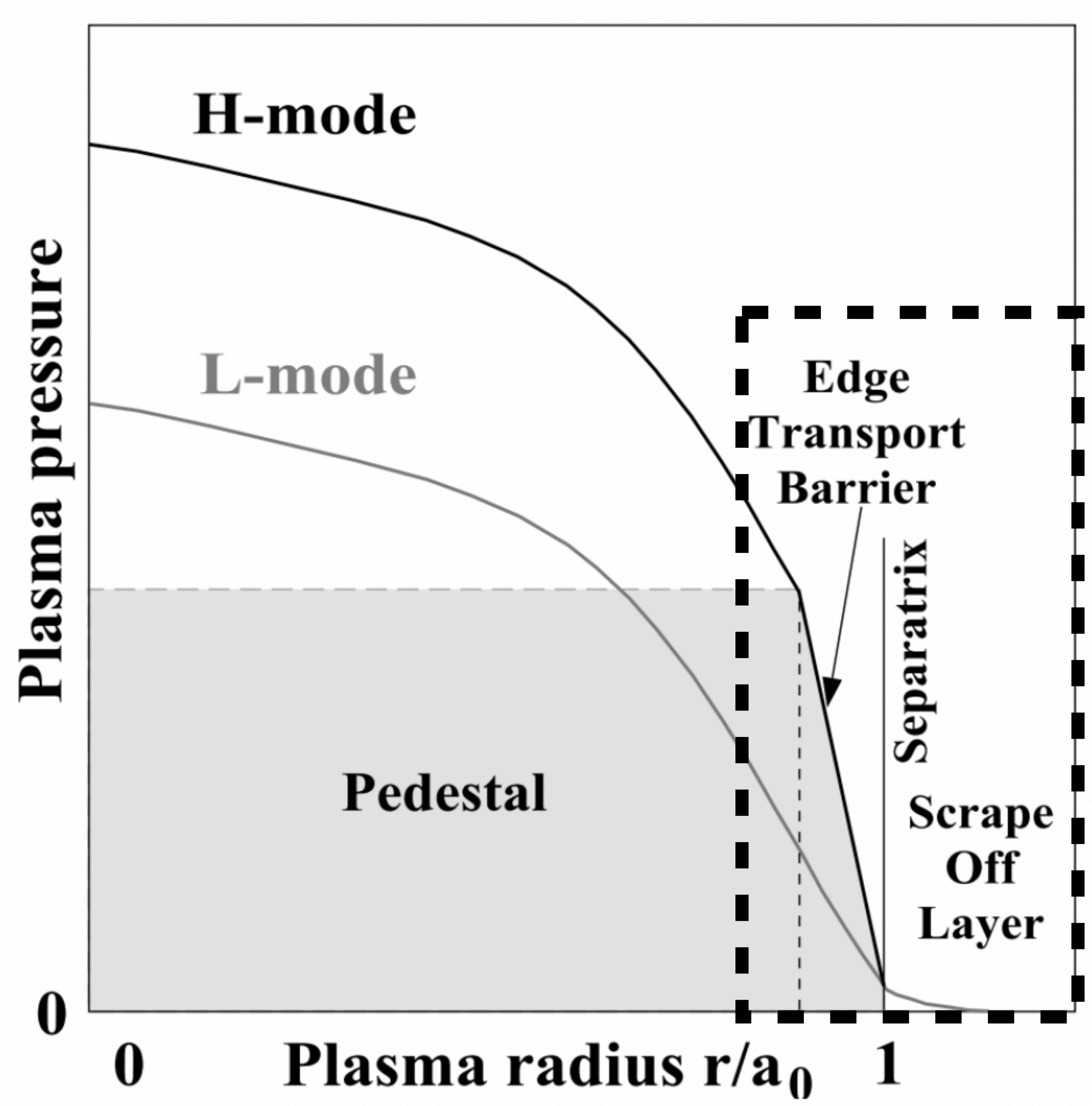
$$\int_{C_m} d\vec{Z}\; w_i \frac{\partial f}{\partial t} + \oint_{\partial C_m} dS\; w_i^- \widehat{f\vec{\alpha}\cdot\vec{n}} - \int_{C_m} d\vec{Z}\; f\vec{\alpha}\cdot\nabla_{\vec{Z}} w_i = 0$$

- Implicit conservation laws via integrals:
  - particle conservation $\int d\vec{Z}\, \frac{\partial f}{\partial t} = 0$ by taking $w = 1$
  - energy conservation $\int d\vec{Z}\, H\frac{\partial f}{\partial t} = 0$ by taking $w = H$, requires H continuous
  - since H must be continuous, ϕ must be continuous — use standard FEM for Poisson eq.
  - conservation laws require integrals to be computed exactly! (i.e. no aliasing errors)
  - exact integration with numerical quadrature $\sim \mathcal{O}(N_q N_b) \sim \mathcal{O}(N_b^3)$

The edge boundary condition is crucial to reactor performance

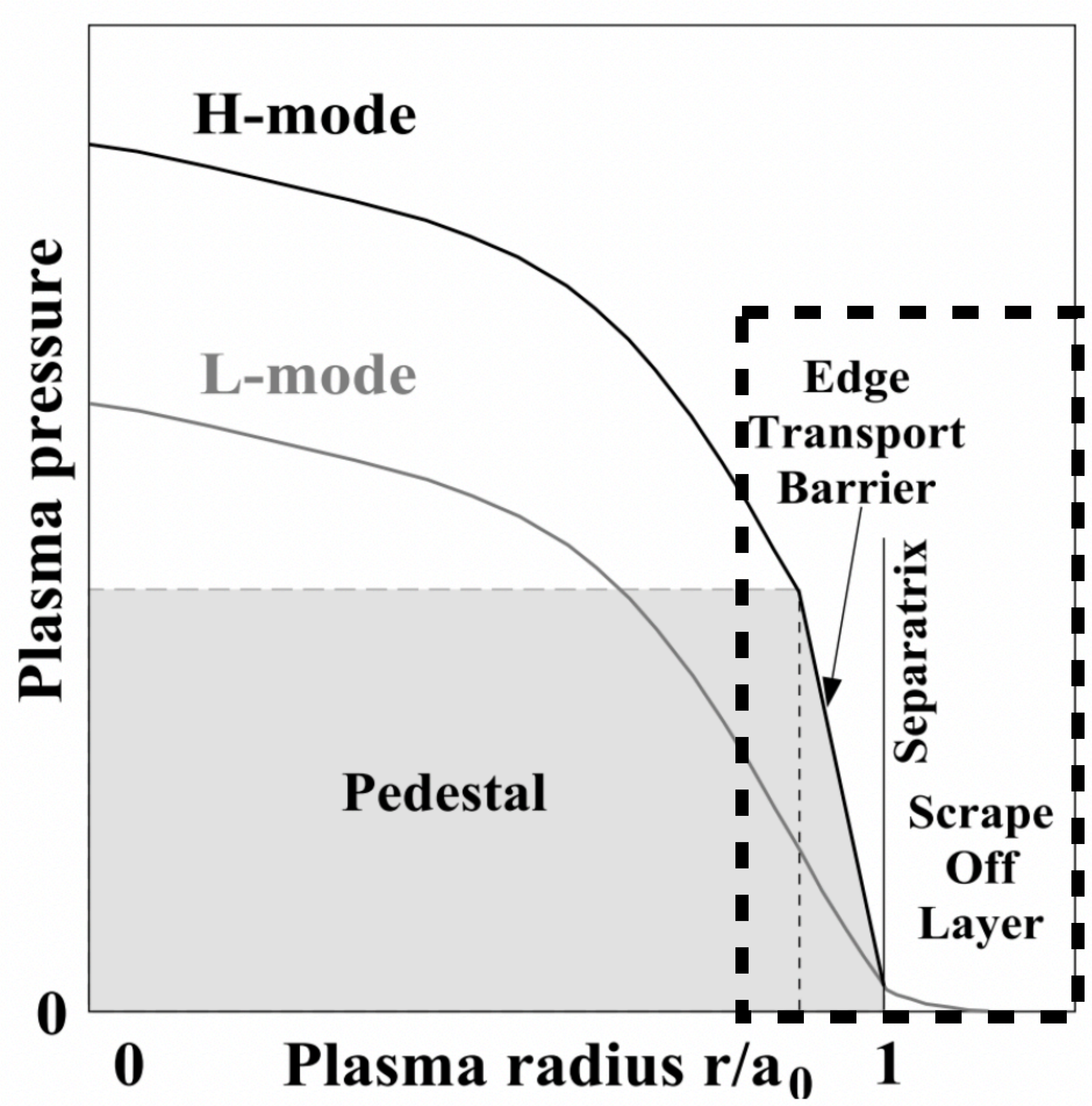
The edge boundary condition is crucial to reactor performance

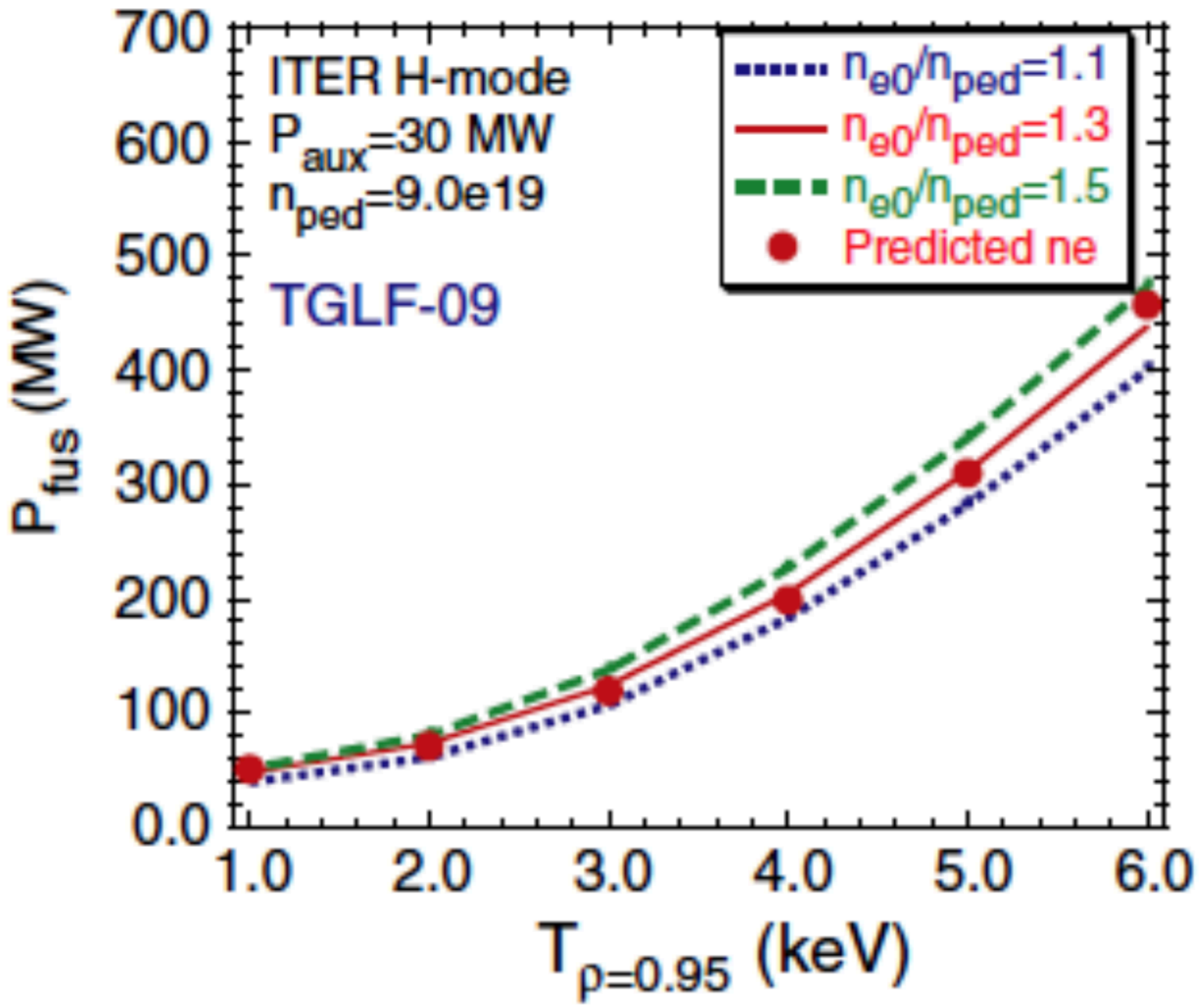
- In most of today's fusion experiments, achieving good performance requires "high-confinement-mode" (H-mode)
  - H-mode occurs when a transport barrier forms at the edge of the core, enabling a steep-gradient region that lifts up the pressure profile in the core (as if it were standing on a "pedestal")



The edge boundary condition is crucial to reactor performance

- In most of today's fusion experiments, achieving good performance requires "high-confinement-mode" (H-mode)
  - H-mode occurs when a transport barrier forms at the edge of the core, enabling a steep-gradient region that lifts up the pressure profile in the core (as if it were standing on a "pedestal")

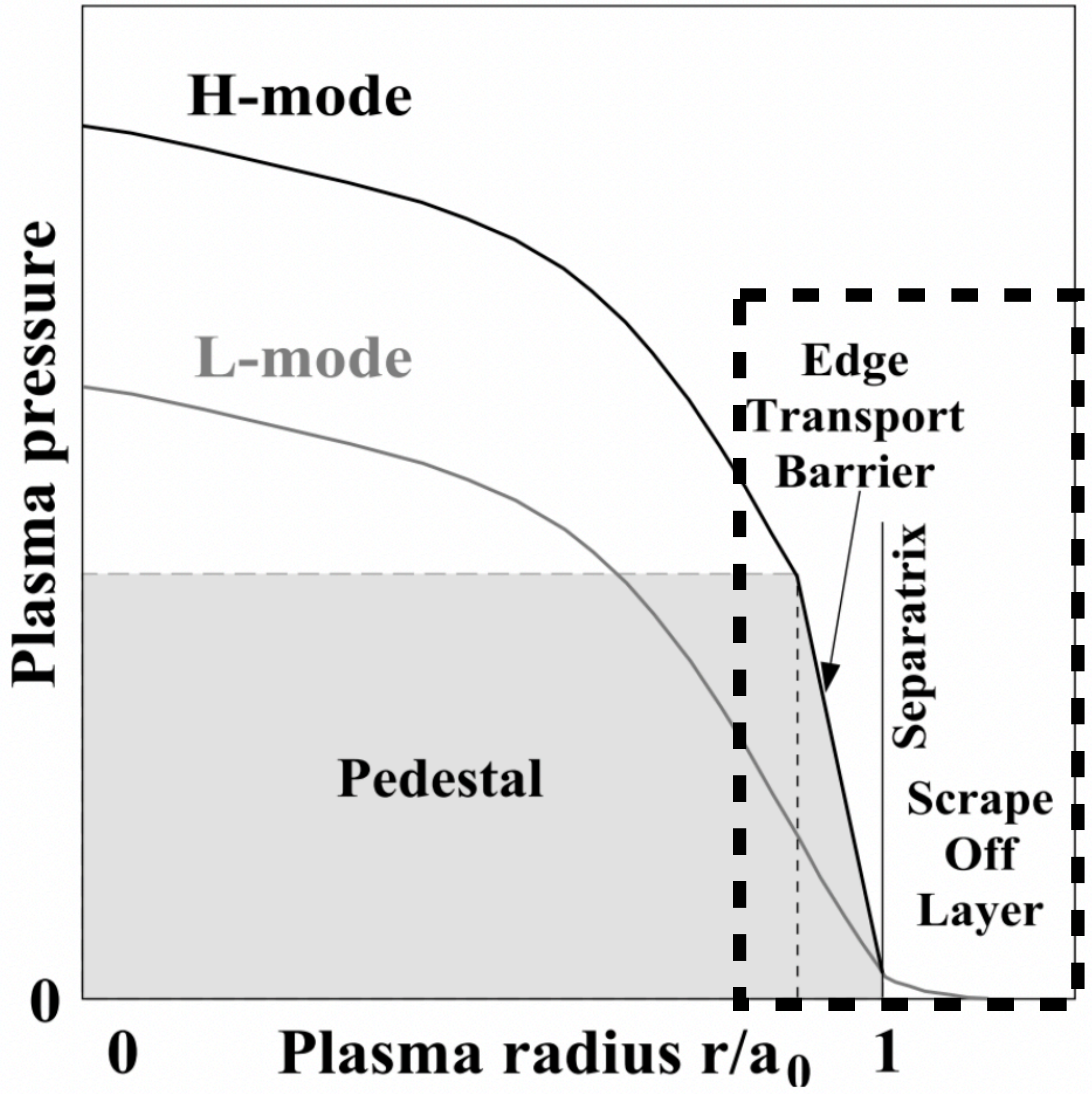


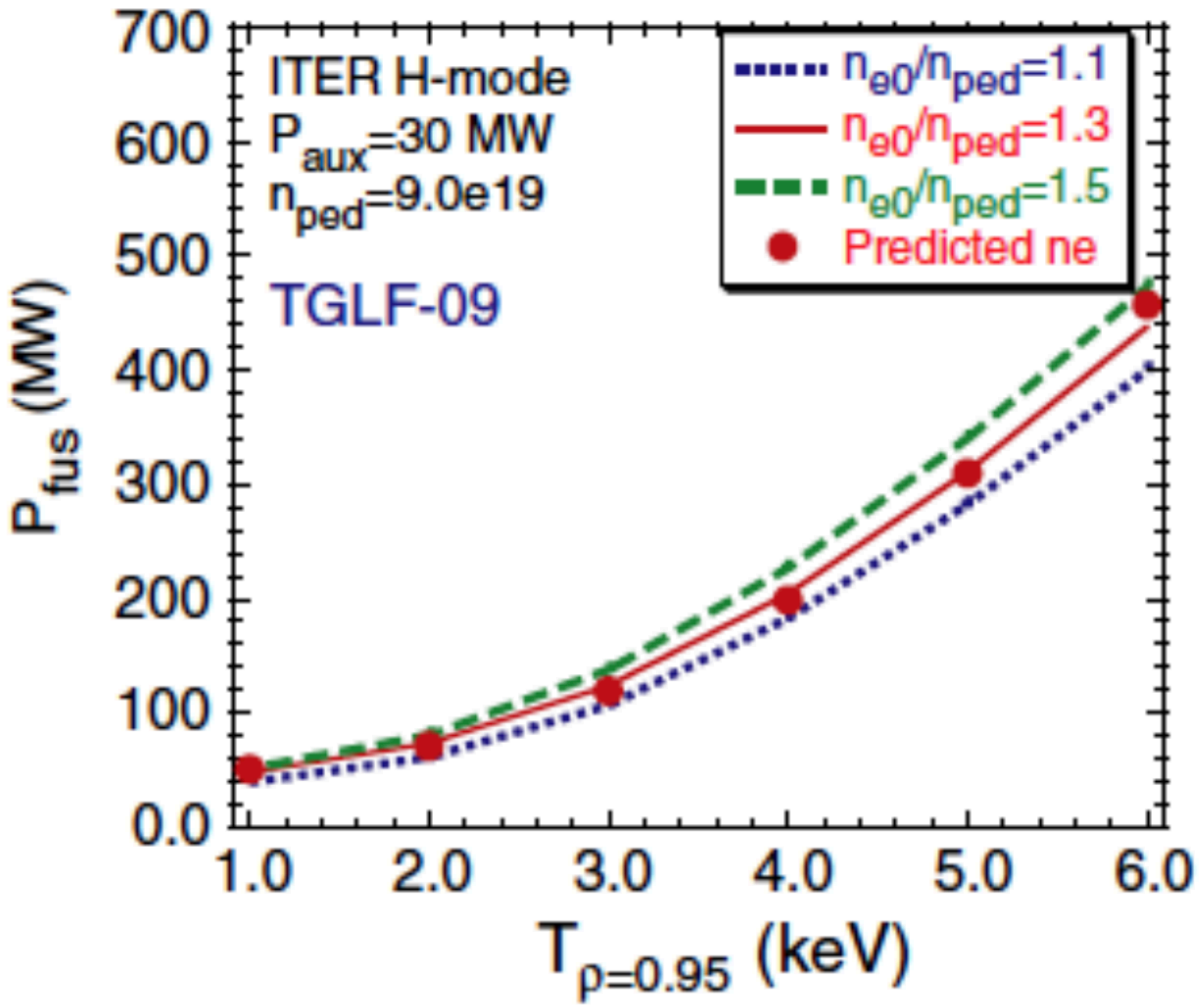


Kinsey et al. Nucl. Fus. 2011

The edge boundary condition is crucial to reactor performance

- In most of today's fusion experiments, achieving good performance requires "high-confinement-mode" (H-mode)
    - H-mode occurs when a transport barrier forms at the edge of the core, enabling a steep-gradient region that lifts up the pressure profile in the core (as if it were standing on a "pedestal")
- **Need to be able to confidently predict/optimize edge boundary condition (pedestal temperature) of reactor designs**





Kinsey et al. Nucl. Fus. 2011

The boundary heat exhaust problem is a potential show-stopper

- Heat exhausted in boundary could damage divertor plates if heat flux width is too narrow
  - Major problem at reactor scale (~500 MW)
- Turbulence in the boundary could help by broadening the width of the heat flux channel
- **Need first-principles kinetic models to model/optimize turbulent broadening of boundary heat flux**
- Modeling boundary requires specialized kinetic turbulence codes; can't use multi-scale approach from core
  - Boundary plasma has large-amplitude fluctuations, open field lines, plasma-wall interactions, X-point geometry, neutral/atomic physics, etc

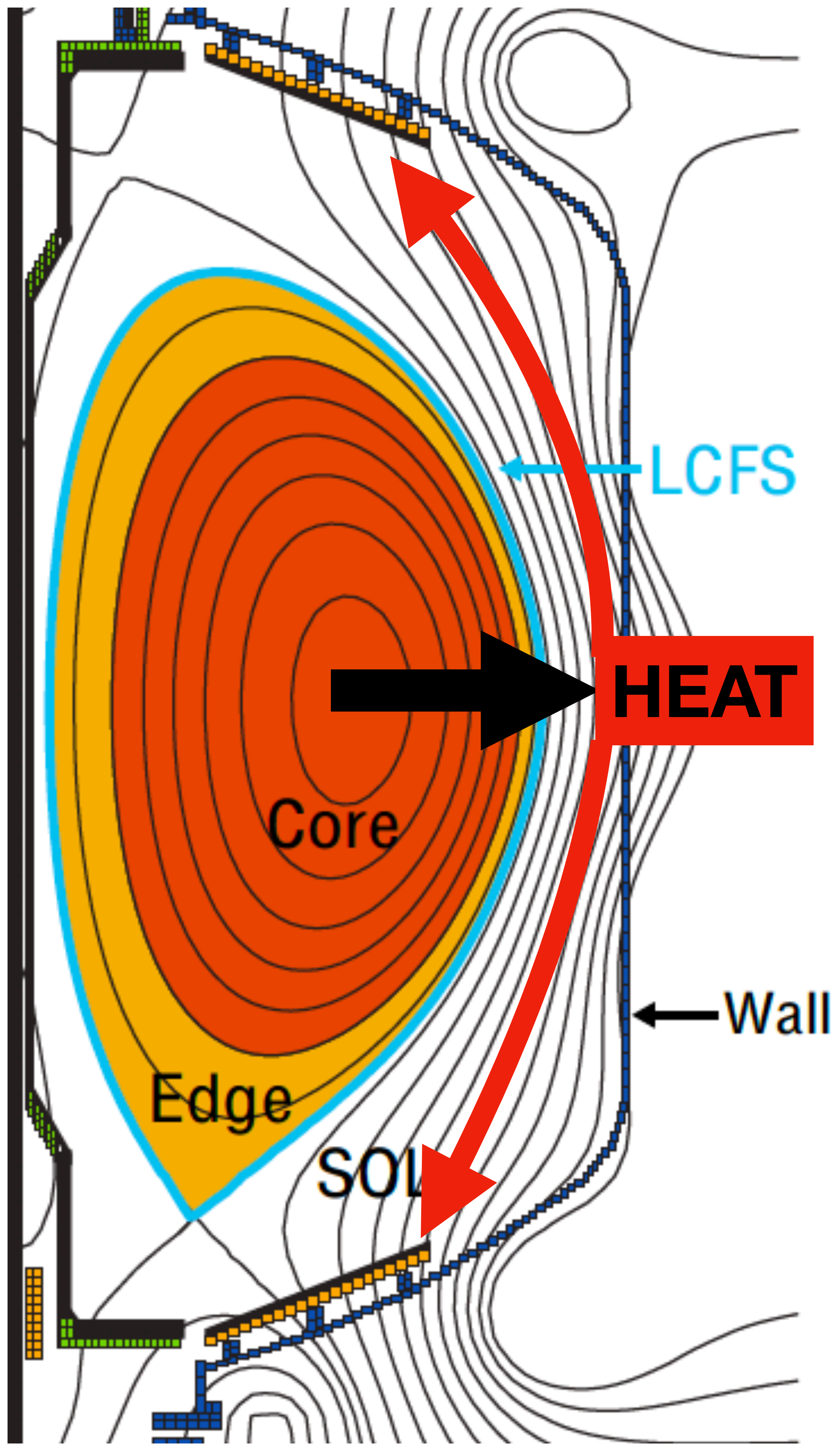


Figure adapted from Stoltzfus-Dueck (2009)