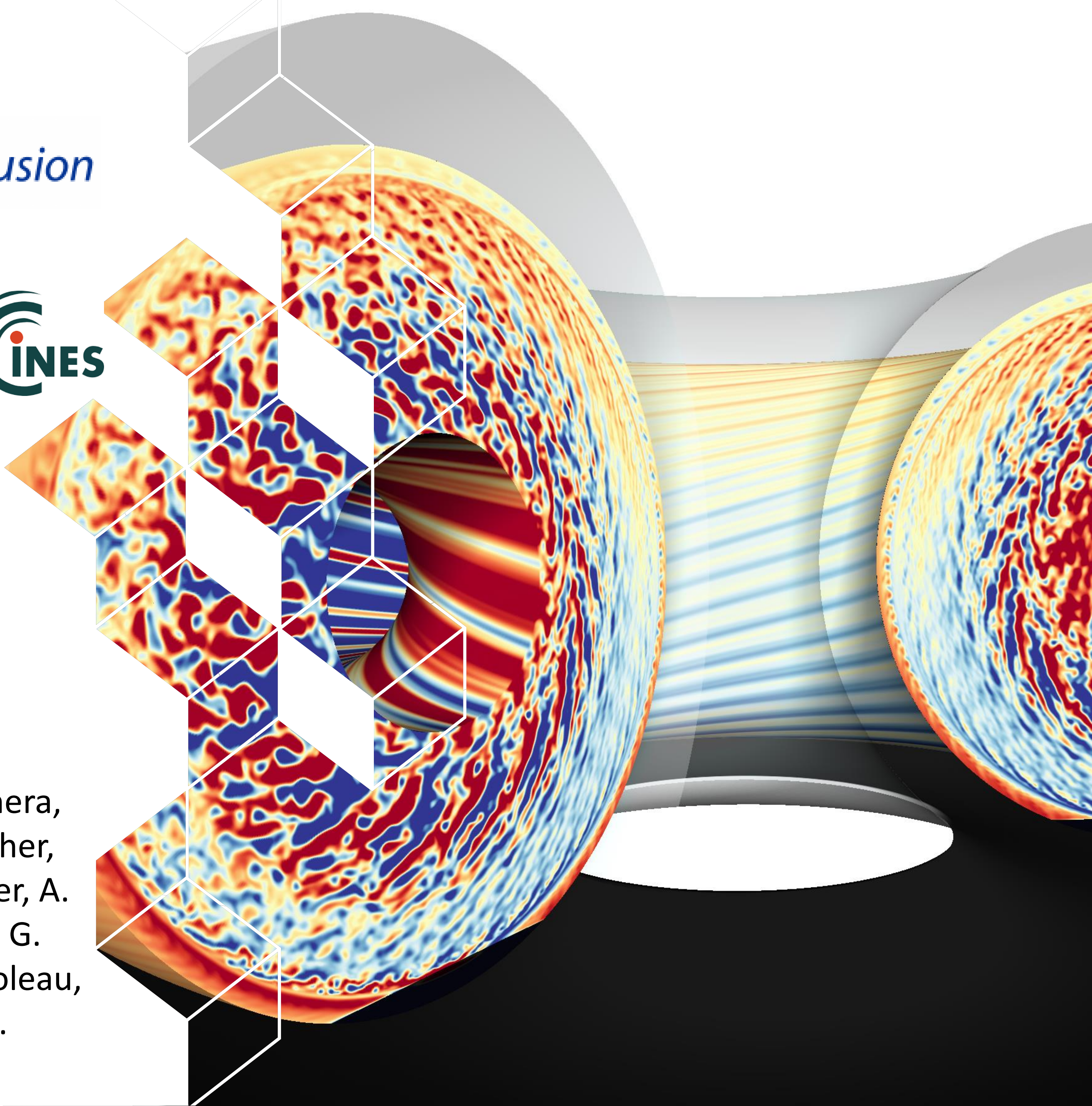
Exascale challenges for tokamak plasma turbulence simulations

Y. Asahi, R. Bigué, J. Bigot, E. Bourne, N. Bouzat, E. Caschera, Y. Cho, L. De Gianni, G. Dif-Pradalier, P. Donnel, C. Ehrlacher, X. Garbet, P. Ghendrih, C. Gillot, V. Grandgirard, E. Gravier, A. Hoffmann, A. Kara, G. Latu, B. Legouix, M. Lesur, K. Lim, G. Lo-Cascio, E. Malabeouf, Y. Munschy, K. Obrejan, T. Padioleau, C. Passeron, M. Peybernes, Z. Qu, T. Rouyer, Y. Sarazin, E. Sonnendrücker, R. Varennes, P. Vidal



Outline

1. General context

2. The GYSELA code (current status)

3. Roadmap for the future: upgrades to target Exascale simulations

4. Conclusion

Two strategies to obtain controlled fusion reactions in a lab: inertial fusion vs magnetic fusion

To sustain a plasma without external heating ($P_{fus} = P_{loss}$), one needs to fullfil the Lawson criterion

Ignition = $P_{\text{self-heating}} > P_{loss}$ $\Leftrightarrow$ $nT\tau_E \geq 3 \cdot 10^{21} \text{keV s/m}^3$

density

Temperature (Optimal $\sim 1.5 * 10^8$ K)

Energy confinement time

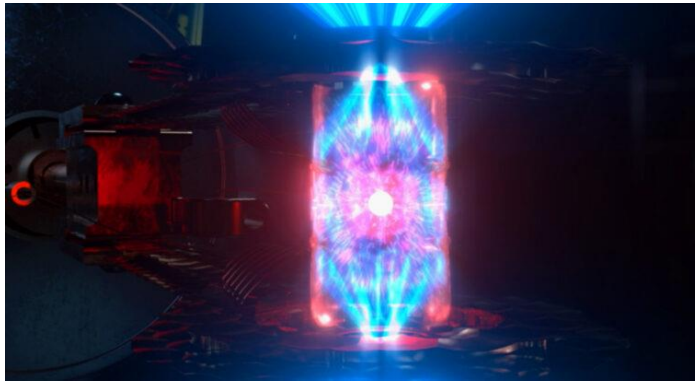
D+T $\rightarrow$ ^{4}He + n (+17,6MeV)

Plasma heating

- Tritium
- Energy prod.

inertial fusion

NIF

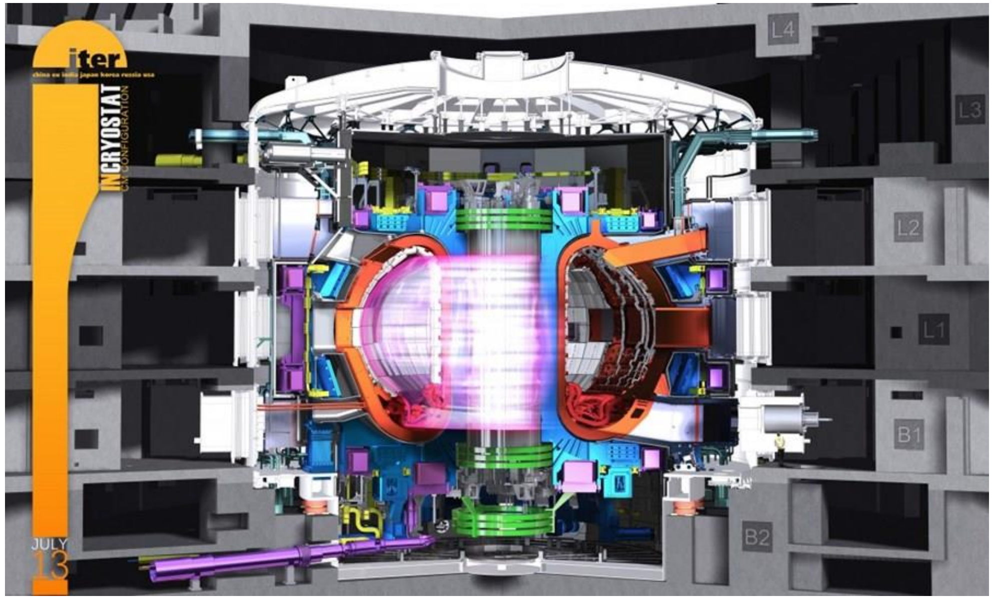


Ignition obtained for the first time! $P_{X-rays} < P_{fus}$ [H. Abu-Shawareb, PRL 2022]

- Historical physics result!
- But still far from Electricity production

  $P_{fus} \sim P_{lasers} << P_{grid}$ + low repetition rate

magnetic confinement



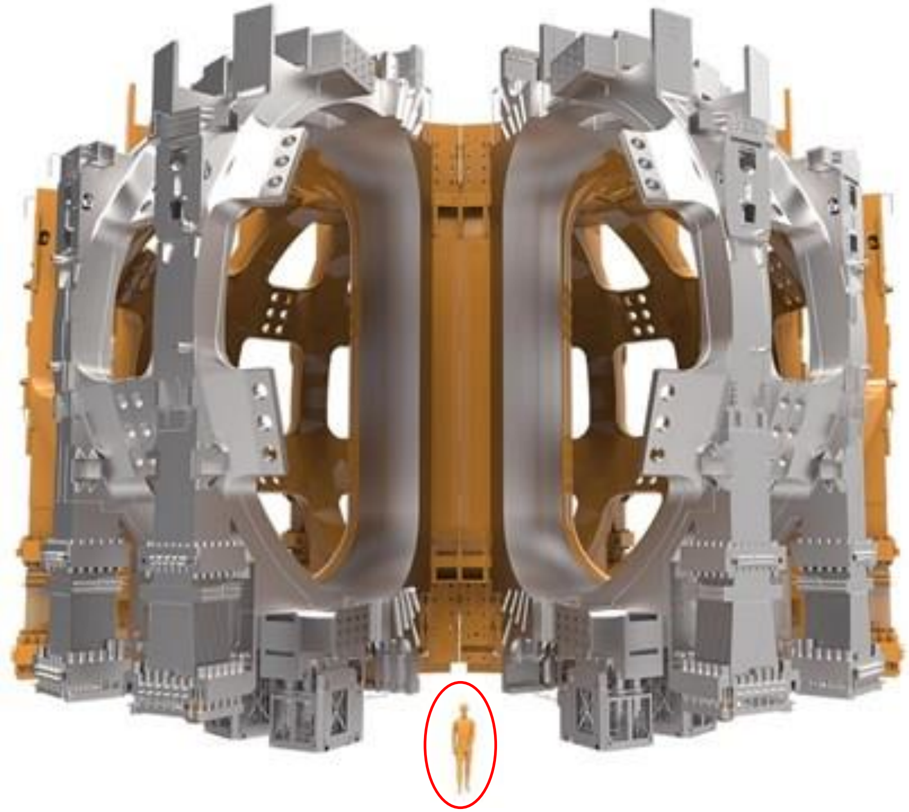
ITER

ITER project

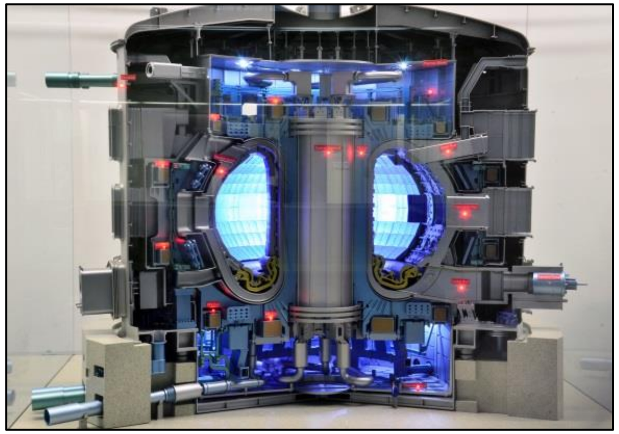
One of the largest scientific project of mankind history

International project: China, India, Japan, Korea, Russia, UE, USA

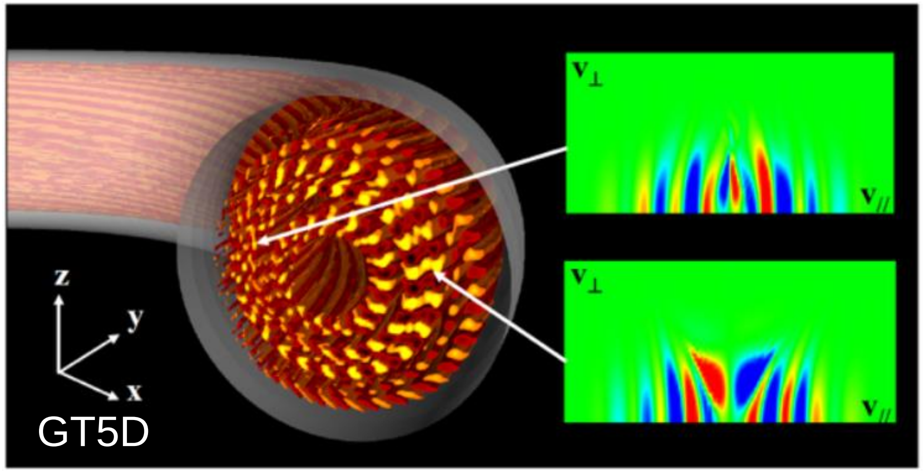
Under construction at Cadarache



Success of fusion requires understanding and prediction of transport in tokamaks



ITER project



- To optimize performance and minimize risks, each ITER scenario will have to be numerically validated.
- A complete chain of numerical tools will be required, ranging from scale models, which can be used in real time, to first-principles simulations, which are more costly but more reliable.
- Turbulent transport mainly governs confinement in Tokamaks
- Tokamak plasmas weakly collisional → Kinetic approach mandatory
  - Fusion plasma turbulence is low frequency → fast gyro-motion is averaged out
  - Gyrokinetic approach: phase space reduction from 6D to 5D

Outline

1. General context

2. The GYSELA code (current status)

3. Roadmap for the future: upgrades to target Exascale simulations

4. Conclusion

The GYSELA code

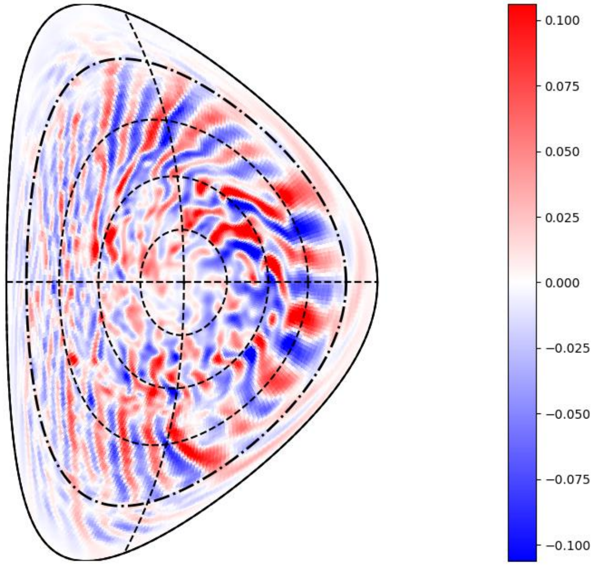
- GYSELA is the unique gyrokinetic code based on a **semi-Lagrangian** scheme
  - Fortran 90 + few C modules with hybrid MPI/OpenMP parallelisation optimized up for up to 750,000 cores
- Development in strong collaboration between physicists, mathematicians and computer scientists
  - Ex: **SELALIB** software library = **joint effort INRIA** Project Lab FRATRES - **CEA**/IRFM – MINGUS (INRIA/Rennes) and TONUS (INRIA/Strasbourg) + Max-Planck-Institut für Plasmaphysic (**IPP**/Garching) started in 2011.

→ SELALIB modules recently successfully coupled to GYSELA [Emily Bourne, PhD 2022]

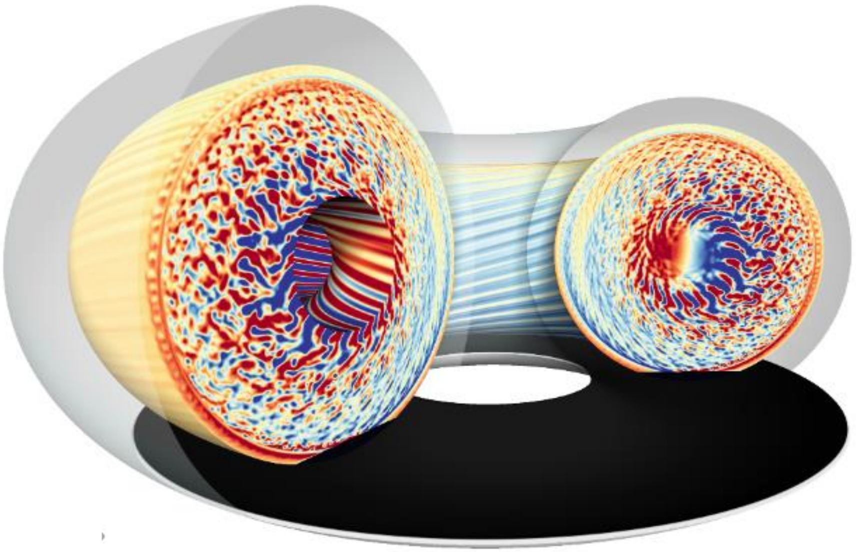
International collaboration France + Switzerland+ Germany + Singapour

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Improvements of the physics model in the last 10 years



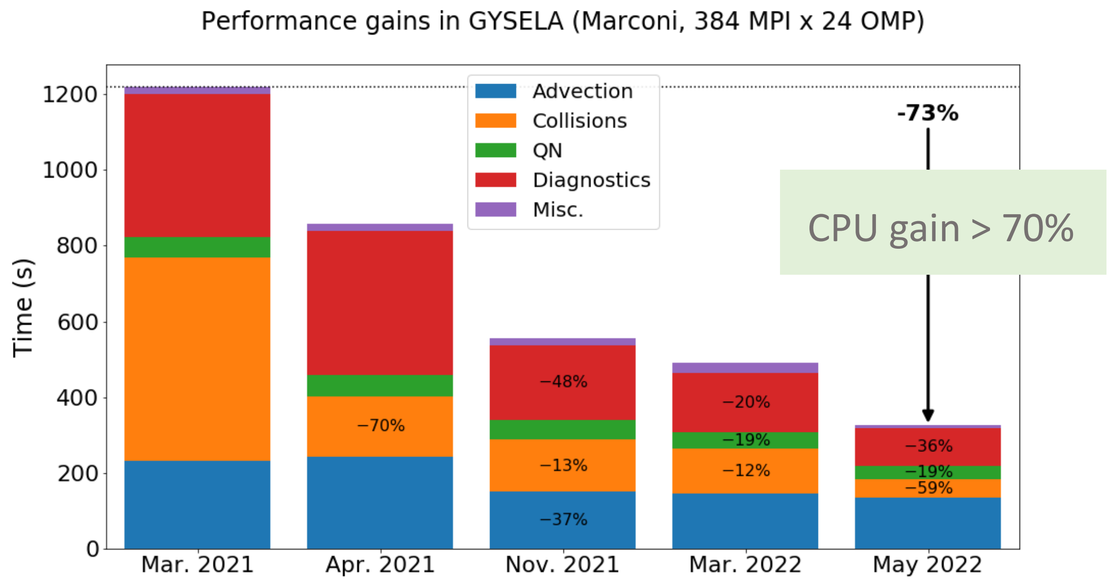
- Electrons: from adiabatic response (not simulated) to kinetic
  [C. Ehrlacher 2018, V. Grandgirard 2019]
- Simulation of impurities (= minority ion species)
  [D. Estève 2018, P. Donnel 2019, K. Lim 2023]
- Geometry: from circular to shaped plasmas + ripple
  [Bourne 2023, R. Varennes 2023, PhD L. De Gianni 2023-2026]
- Boundary conditions: from simple buffer to more realistic immersed boundaries
  [E. Caschera 2018, Dif-Pradalier 2022, PhD Y. Munschy 2021-2024]
- From electrostatic (constant magnetic field) to electromagnetic
  [PhD C. Gillot 2017-2020, PhD R. Bigué 2023-2026]
- More complex operators: collisions, sources…
  [P. Donnel 2018, G. Lo-Cascio 2022]

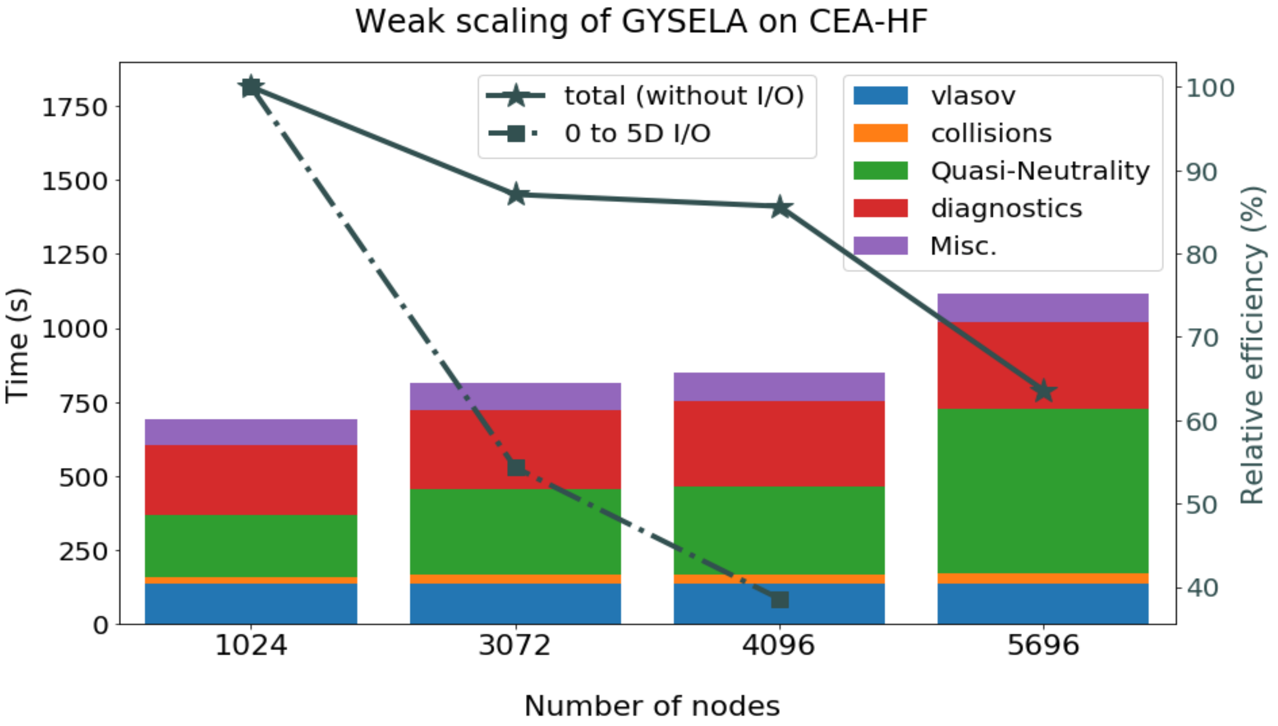


Need for constant code optimization to partly compensate for the increased complexity of the physics + adapation to new architectures

Numerical improvements in 2021-2022

- Vectorisation
- Blocking → Cache optimisation
- Asynchronous MPI communications





Relative efficiency of 85% on more than 500k cores and 63% on 729 088 cores

[V. Grandgirard et al., PASC 2022]

Outline

1. General context

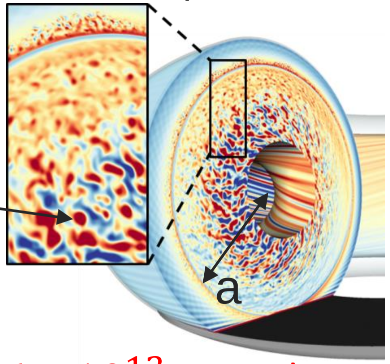
2. The GYSELA code (current status)

3. Roadmap for the future: upgrades to target Exascale simulations

4. Conclusion

Need for Exascale computing to simulate ITER

- The dominant transport in tokamaks is related to electromagnetic turbulence. A first principle code is intrinsically multi-scale as it needs to solve simultaneously small scale turbulence and large scale evolution of the plasma



- **Spatial scale**
    - Turbulent structures proportional to the ion Larmor radius $\rho_i = \frac{\sqrt{mk_bT}}{eB}$
    - Size of plasma ‘a’
    - The grid resolution scales as $\rho_*^{-2}$where $\rho_* = \frac{\rho_i}{a}$. WEST: $\rho_*^{-1} = 250$, ITER: $\rho_*^{-1} = 1000$

➢ Grid resolution needs to be increase by a factor ~ 16 from WEST ($\sim 10^{11}$points) to ITER ($\sim 1.6 * 10^{12}$points)

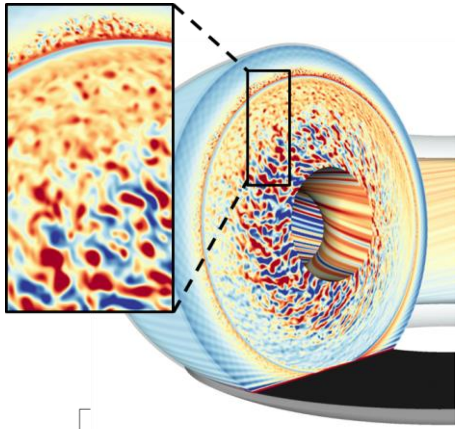
- **Temporal resolution**
    - Correlation time of turbulence proportionnal to $\Omega_{ci} = \frac{eB}{m}$
    - number of iterations scales as the confinement time / turbulent time scale, i.e. $N_{iter} \propto \tau_E \Omega_{ci} \propto \rho_*^{-2}$(pessimistic) or ρ_*^{-3}(optimistic)

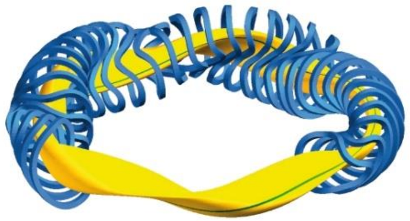
➢ Number of iterations increased by a factor 16 to 64 from WEST to ITER

- Well resolved simulations of WEST ~ 10 millions of CPU hours → impossible to simulate ITER with current code& HPC

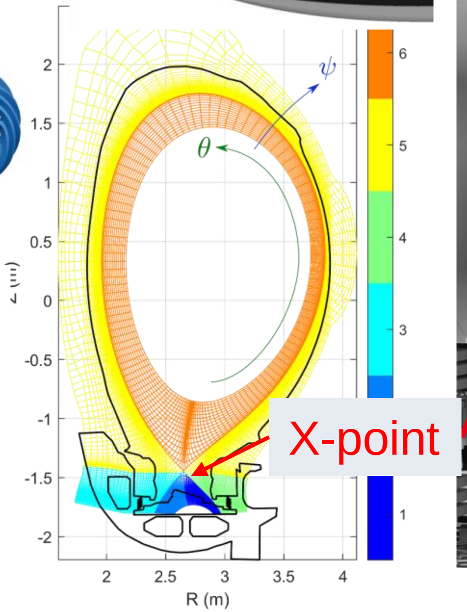
Roadmap for Gysela-X++ towards exascale
→ Why do we choose to rewrite GYSELA ?

- **Gysela-X++** = GYSELA in modern C++ with X-point for exascale ITER core-edge turbulence simulations (+ 3D via BPI / Renaissance Fusion)
  - **Rewritting of the code in modern C++ with MPI + Kokkos**
    - ➢ Portable code on new exascale architectures
  - **Non-uniform meshes**
    - ➢ relevant density & temperature gradients at edge-SOL
  - Semi-Lagrangian scheme for multi-patches
    - ➢ **X-point** geometry
  - Implementation of a **3D scalable Poisson solver**
    - ➢ X-point configuration + stellarator configuration
  - **Scalable I/O and in-situ diagnostics**

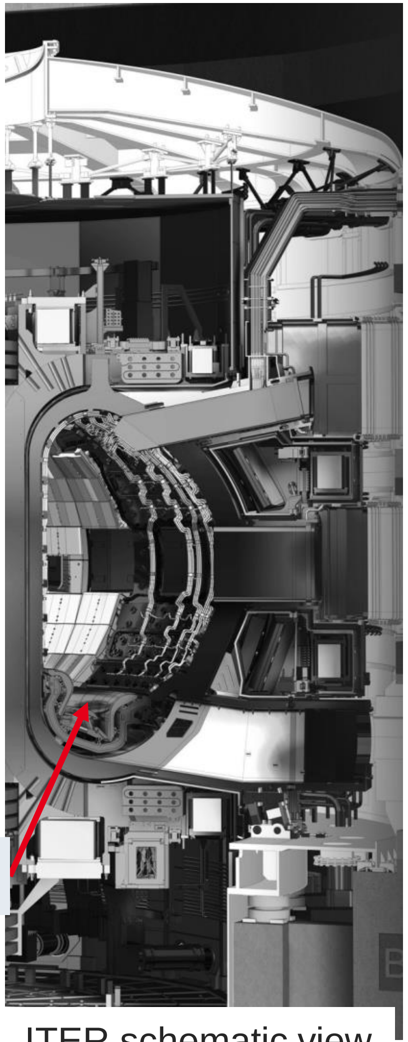




- Development of the new code supported by multiple projects/collab.
  - National projects: Moonshot CExA (2023-2025), PTC Dose (2023-2024), PTC Assist (2023-2024) + PEPR NUMPEX (2023-2028) + ANR AIM4EP (2022-2025)
  - EoCoE-III
  - BPI funding in kink with Startup Renaissance Fusion (2024-2027)
  - European projects: EUROfusion TSVV (2022-2026)
  - Joint research center (SAFE) Singapore Alliance with France for Fusion Energy



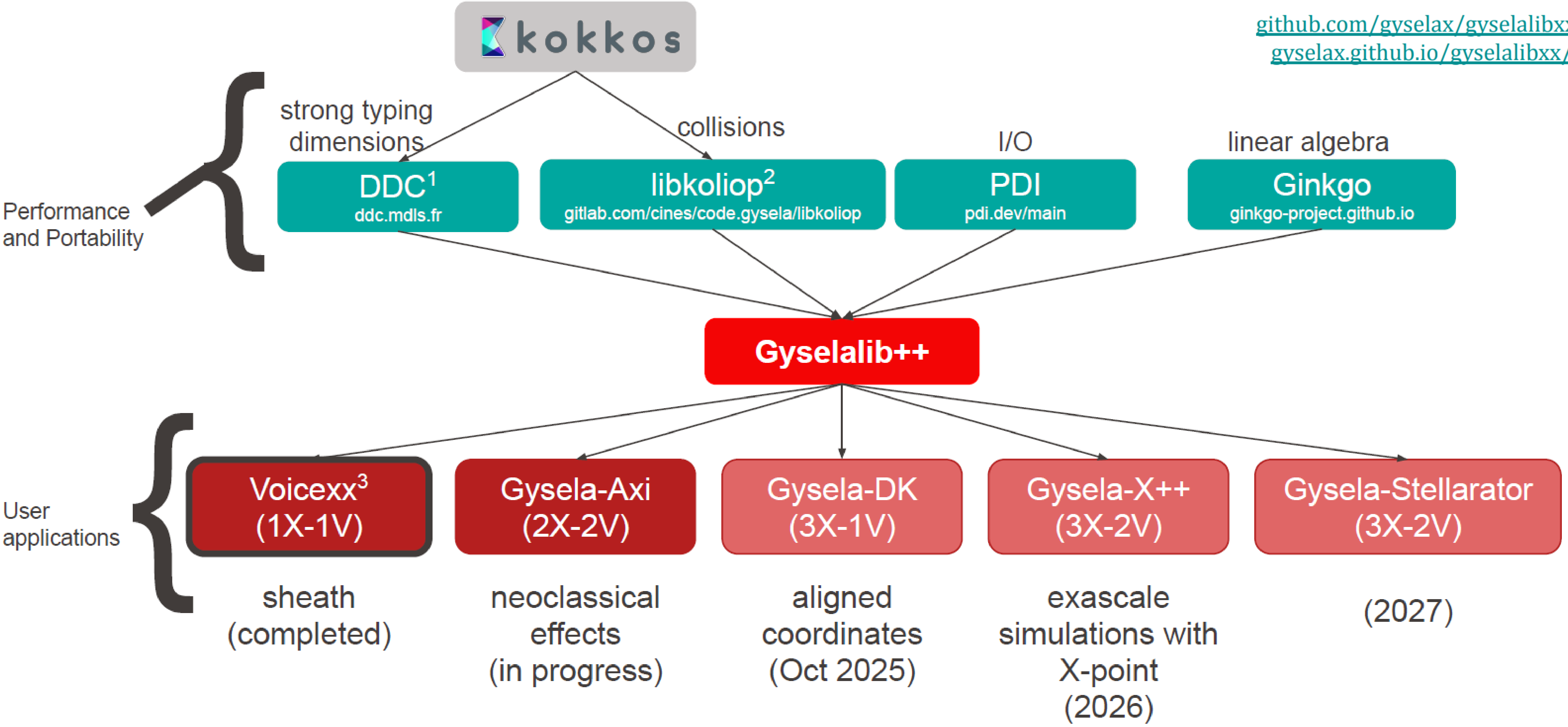
SOLEDGE-3X
X-point geometry



ITER schematic view

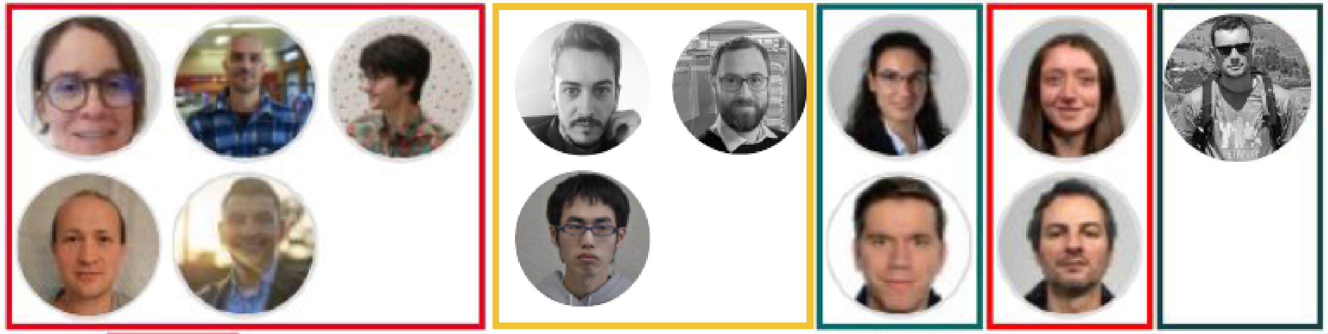
Gyselalib++: A new modular library in C++

github.com/gyselax/gyselalibxx/
gyselax.github.io/gyselalibxx/



1 T. Padioleau et al. DDC: The Discrete Domain Computation library, JOSS, submitted
2 P. Donnel et al. A multi-species collisional operator for full-F global gyrokinetics codes: Numerical aspects and verification with the GYSELA code, CPC 2019
3 E. Bourne et al. Non-uniform splines for semi-Lagrangian kinetic simulations of the plasma sheath, JCP 2023

Gyselalib++: A collaborative open source project



irfm

EPFL

CINES

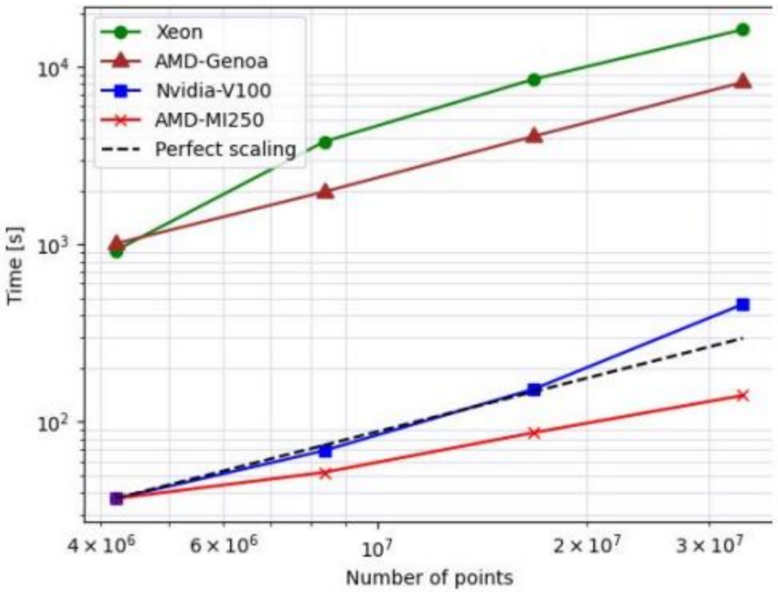
github.com/gyselax/gyselalibxx/
gyselax.github.io/gyselalibxx/

Funded and supported by multiple collaborations:

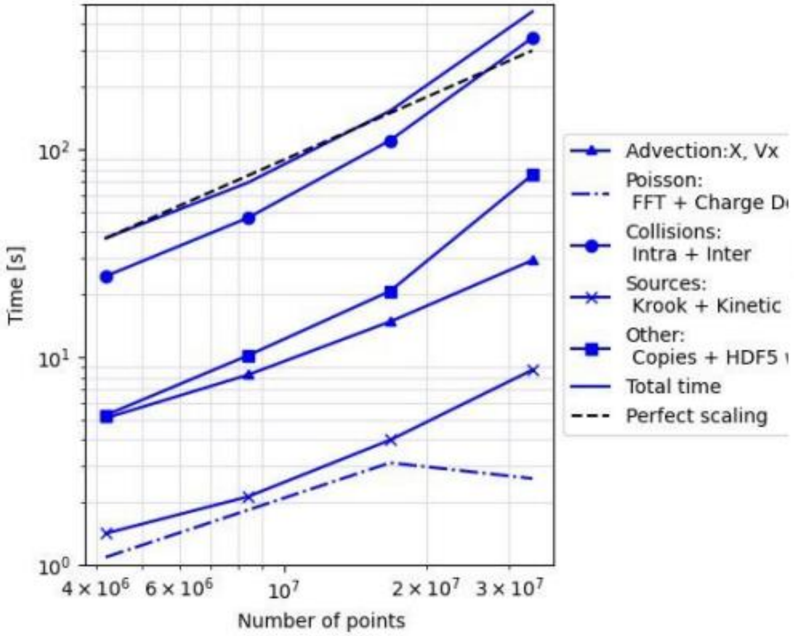
- Moonshot CExA → Development of new functionalities in Kokkos (e.g. KokkosFFT and KokkosKernels)
- EoCoE-III → Mathematics expertise (IPP), HPC Optimisation (FAU), Linear solvers (CERFACS)
- PEPR NumPEx (Exa-DoST) → I/O Optimisation, in situ diagnostics and dataworkflows for IA

Performance: 1X1V Sheath simulation

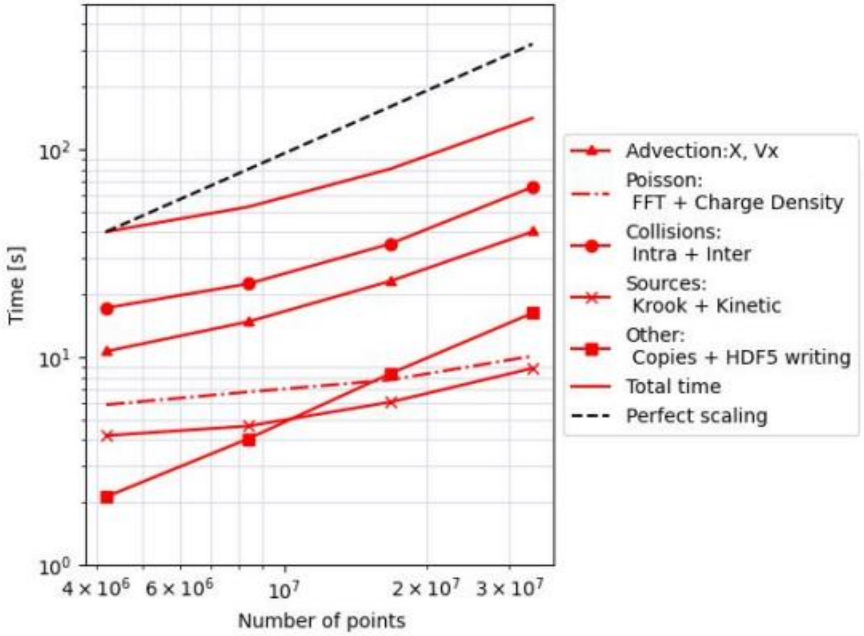
- See [E.Bourne et al. 2023] for numerics and [Y.Munschy et al. 2023] for physics
- Tests performed Oct. 2024
- Single GPU simulations (No MPI)



CPU versus GPU



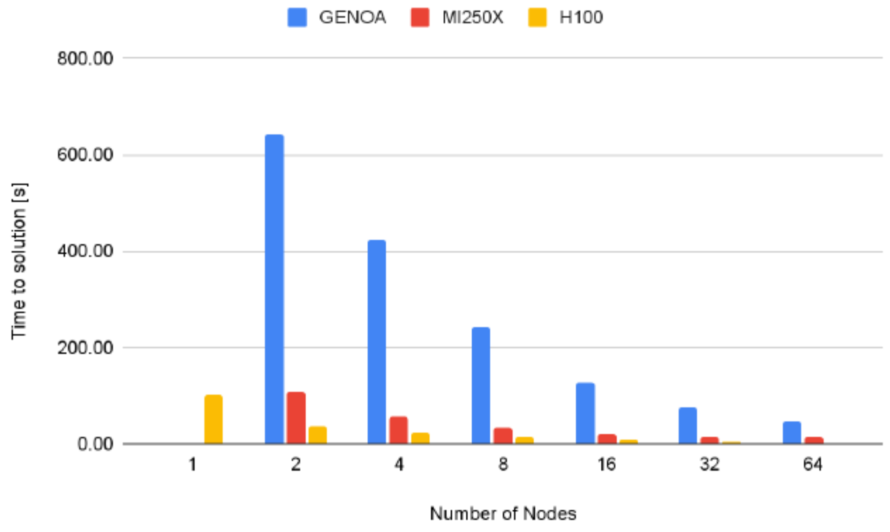
GPU: NVIDIA V100

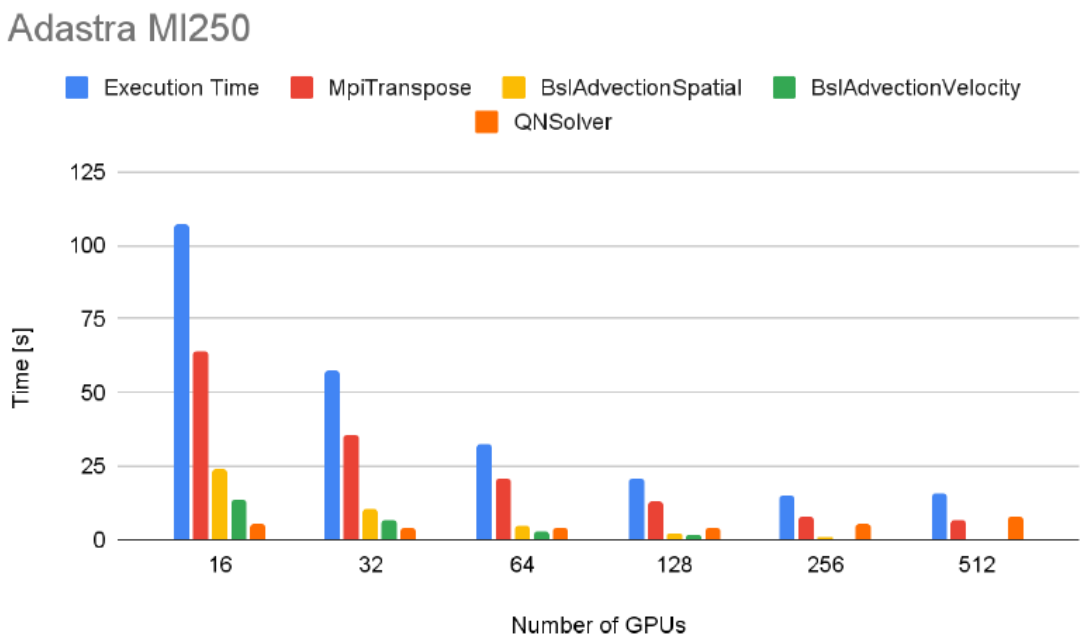


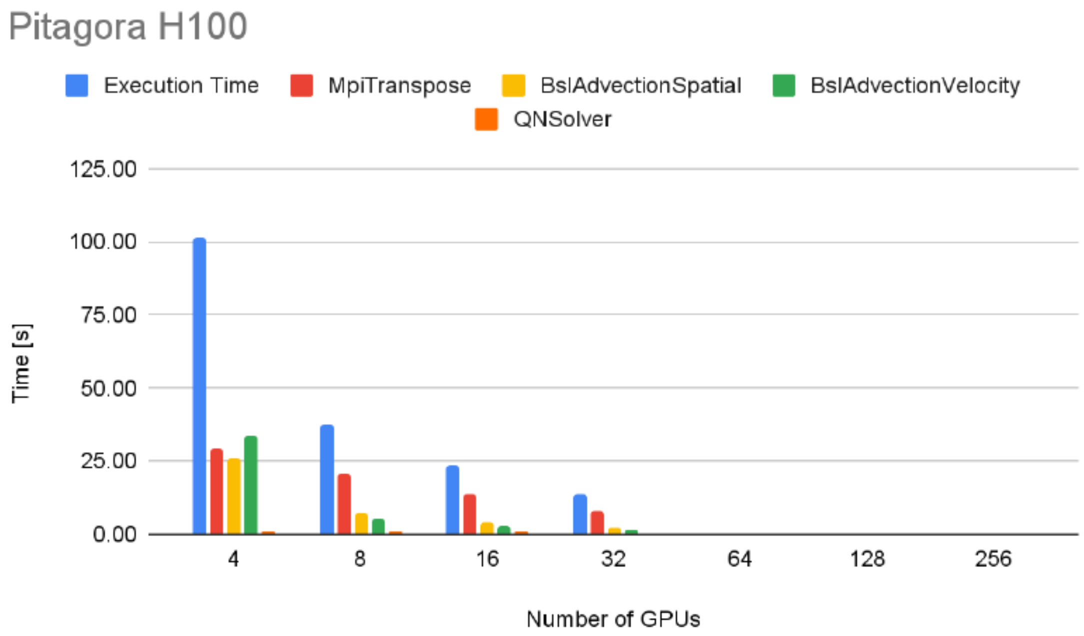
GPU: AMD-MI250X

Performance: 2X2V Landau damping

- Kokkos strategy → Code portable to various architectures :
  - AMD GENOA (CPU)
  - AMD MI250 (GPU)
  - Pitagora H100 (GPU)
- Preliminary scalability results
- Optimisation efforts will be performed on the 5D version





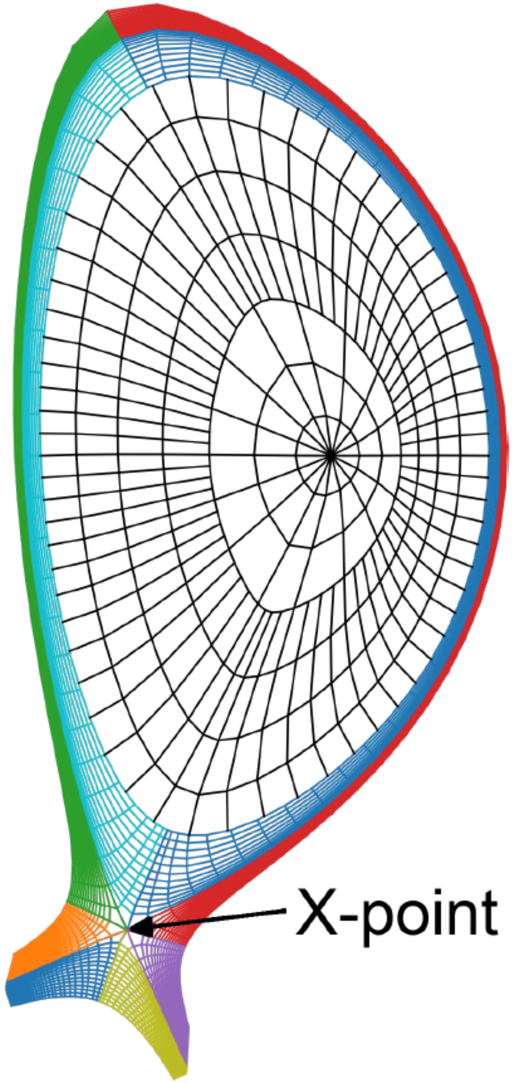


Handling an X-point

Patches are required to manage the X-point geometry along field lines.

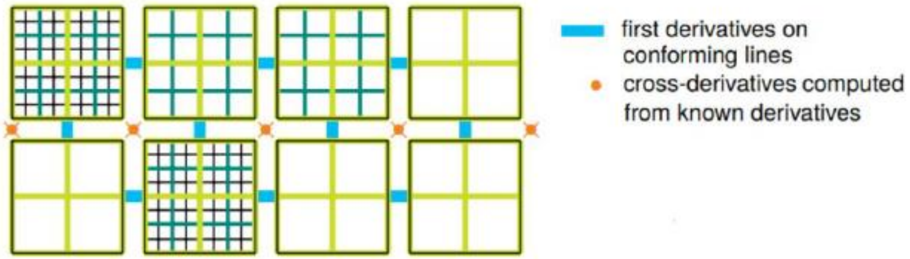
There are 2 main axes of research to handle this:

- Semi-Lagrangian Advection
  - P. Vidal (IPP) PhD 2022-2025
- Poisson solver
  - A. Hoffmann (IPP) PhD 2023-2026
    - Multi-patch (CONGA) FEEC solver
  - HyTeg A. Dasari (CERFACS) PhD 2024-2028
  - Without patches as a starting point:
    - GMGPolar coupling (CERFACS)
    - Porting of Zoni's polar spline solver E. Malaboeuf (CINES) PhD 2025-2028

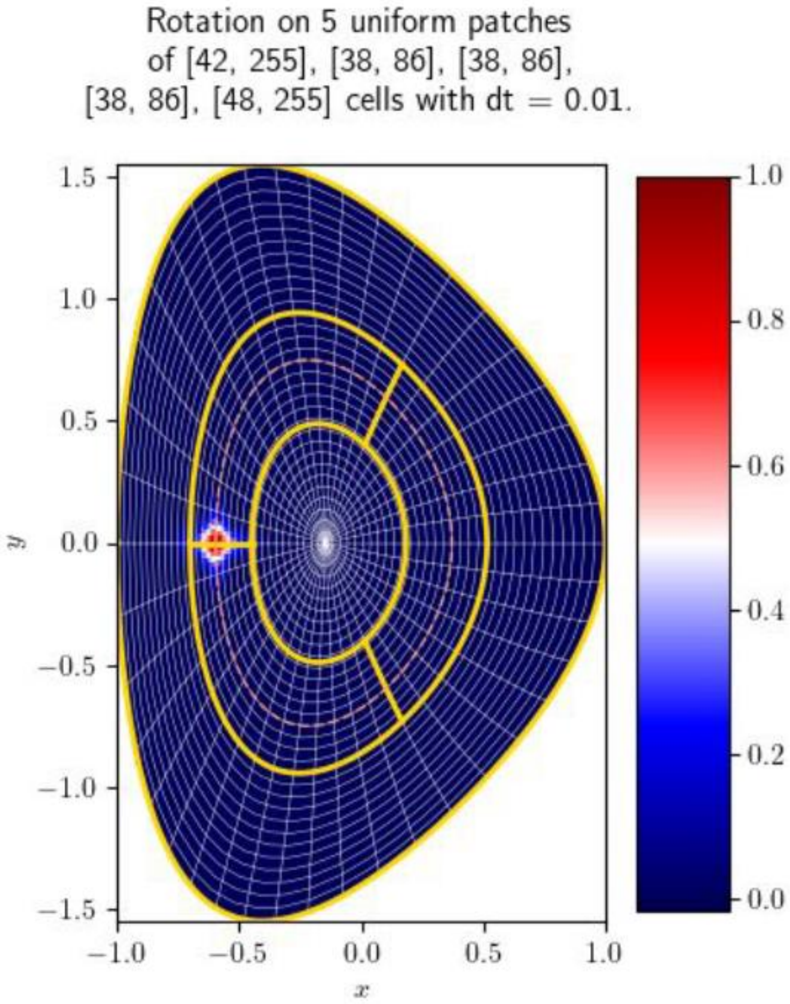


Semi-Lagrangian multi-patch advection

- Strang splitting is used to break the problem into 1D or 2D advections
- Patch-wise spline interpolation with treatment of interfaces
  P. Vidal et al. (2025) *Local cubic spline interpolation for Vlasov-type equations on a multi-patch geometry.* arXiv preprint arXiv:2505.22078



- Local spline interpolation is carried out by DDC
  - Batched LAPACK solvers upstreamed to Kokkos Kernels
    Asahi, Y. et al. (2024) *Development of performance portable spline solver for exa-scale plasma turbulence simulation.* SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE.
  - Ginkgo iterative solvers



[P. Vidal 2025]

Conclusions

- The GYSELA code at the era of pre-exascale for ion-scale turbulence simulations for current tokamaks
  - Optimized up to more than 500k cores on standard CPU architecture (ex: AMD milan)
  - Resource needs: more than 200 millions of CPU hours / year
  - Petabytes of data manipulated per simulation with huge reduction to limit the storage to few Terabytes

  → Lot of physics still to be explored with this version during the development of Gysela-X++ .

- Gysela-X++ : Rewritting in modern C++, more modular and scalable on different accelerated architectures
  - More realistic temperature gradients at the edge: Patches
  - More realistic geometry: X-point and stellarators
  - More physics: neutrals, fusion reactions…
  - Based on DDC library + Kokkos
  - In situ diagnostics foreseen

Backup slides

Two main magnetic topologies to confine a plasma

Tokamak

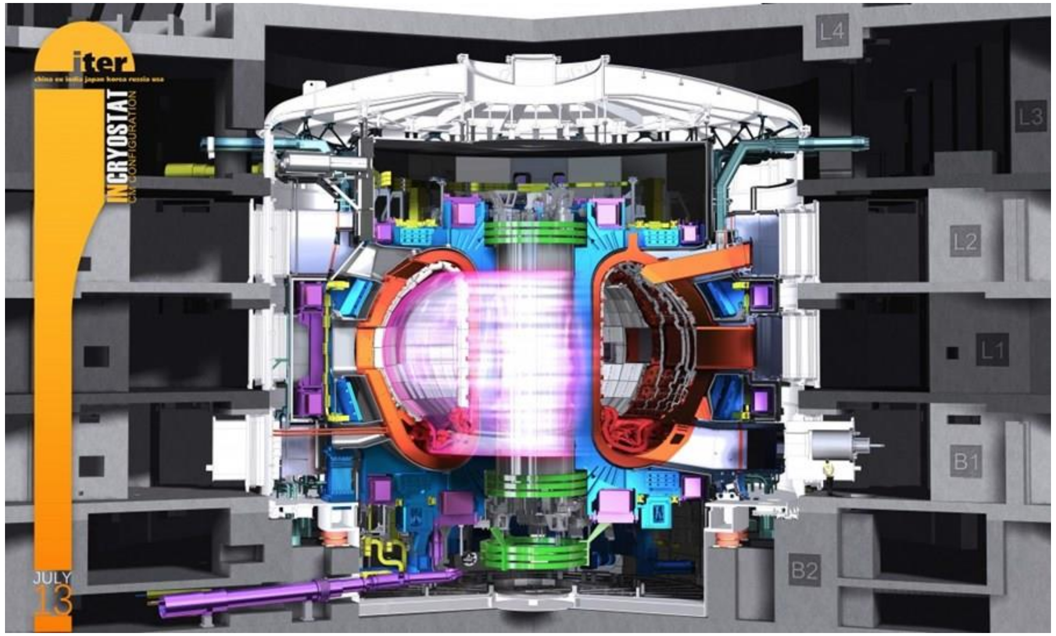
☺ Confinement

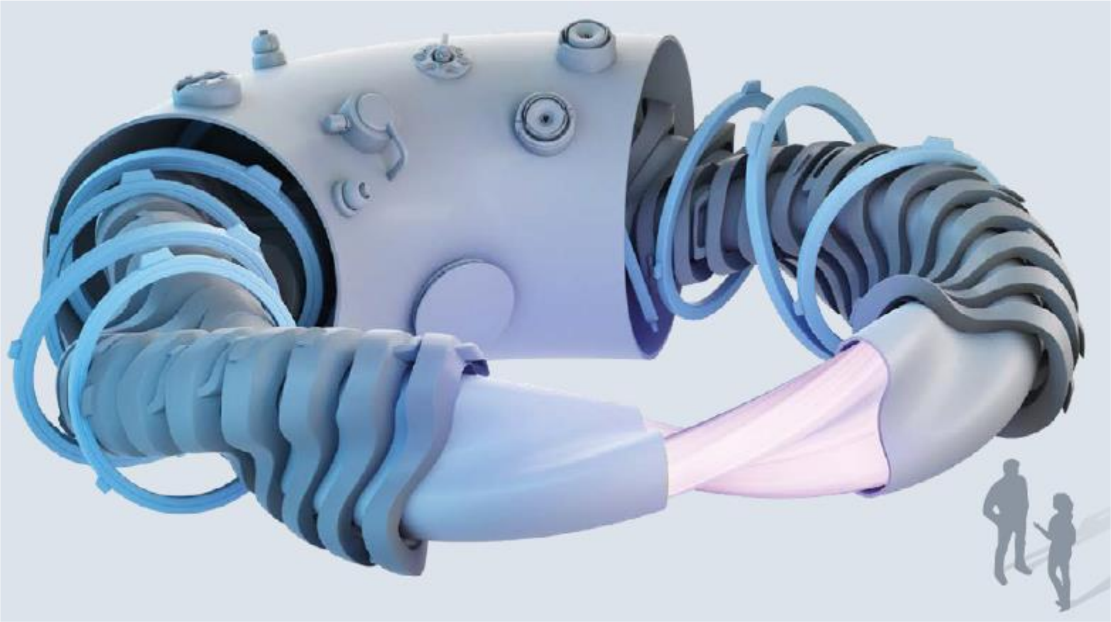
☹ Plasma current, Disruptions

Stellarator

☺ Stability

☹ α-confinement, Technology

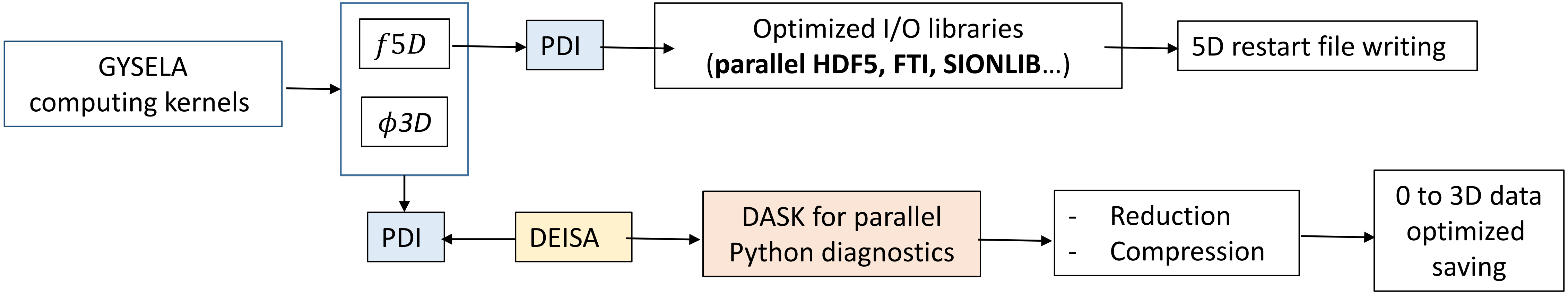




Work in progress : Optimized Data workflow

- Main idea : **Decouple I/O from computing kernels**
  - HPC/IA coupling not trivial : CPU or GPU for computing kernels (Fortran or C++) + GPU for diagnostics +AI (python)

- **Development of in-situ diagnostics framework based on PDI + DEISA + DASK**
  - PDI Data Interface for handling I/O (developed at MDLS) https://pdi.julien-bigot.fr/master/
  - DEISA (dask-enabled in situ analytics) library (developed at MDLS+INRIA) *[A. Gueroudji et al., 2023]*
  - DASK a flexible library for parallel computing in Python https://docs.dask.org/



Within national PEPR NUMPEX (2023-2027)

2028 objective: In-situ AI diagnostics

- Development of **in-situ AI diagnostics to optimize exascale simulations**:
  - Data compression
  - Automatic anomaly detection: Automatic stop of simulation → CPU consumption optimization
  - Automatic rare event detection: Optimisation of diagnostic saving → Memory storage reduction