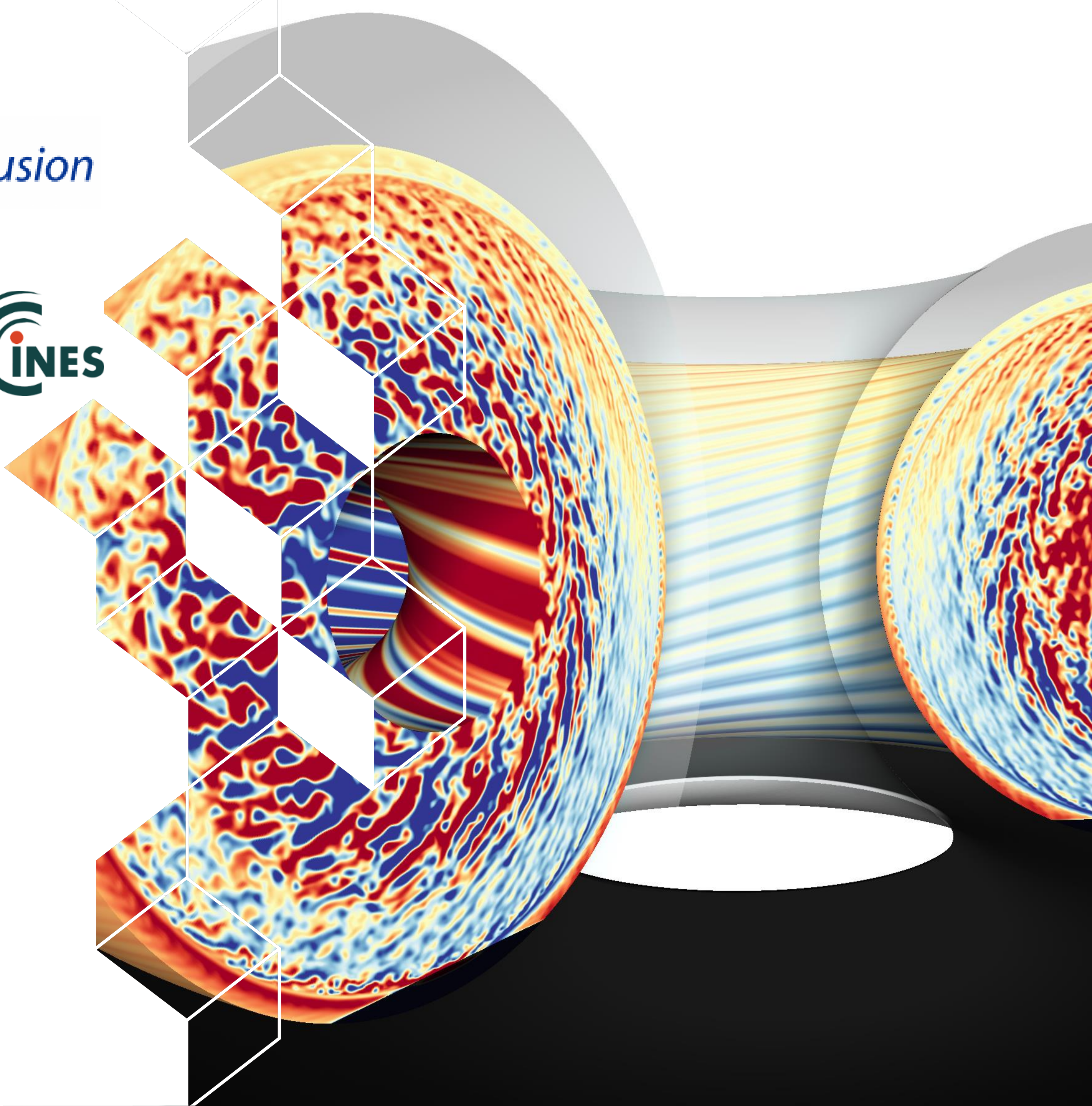
Gysela-axi: a flux-driven full-F gyrokinetic code to simulate axisymmetric tokamak plasmas



P. Donnel and the Gysela team

Outline

1. Context: development of Gysela-X++

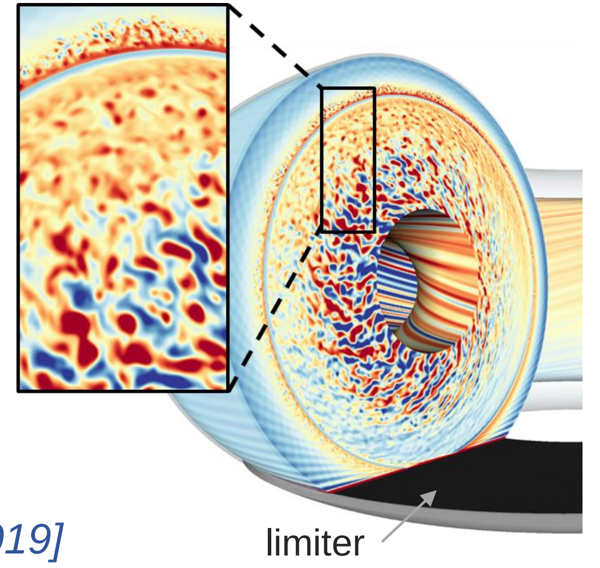
2. Gysela-axi a major milestone

3. Current status of the code and next steps

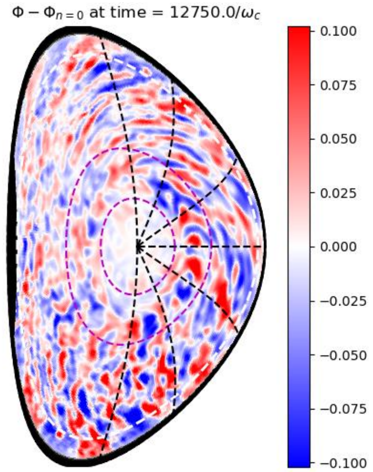
4. Conclusion

GYSELA: GYrokinetic SEmi-LAgrangian code [Grandgirard et al., CPC 2016]

- GYSELA (GYrokinetic SEmi-LAgrangian): first gyrokinetic code based on a **semi-Lagrangian scheme**
- From beginning, **accurate strategy but more HPC demanding**:
  - **Global**: simulate entire tokamak → boundary conditions (SOL-like, limiter)
  - **Full-f**: multi-scale physics
  - **Flux-Driven** (heat, momentum, … sources) → steady state on τ_E
  - **Collision operator** → synergy between neoclassical & turbulent transports



- Permanent improvements of the physics model
  - **Electrons:** From adiabatic response to **kinetic** *[C. Ehrlacher 2018, V. Grandgirard 2019]*
  - **Multi-ion species** → impurity transport *[D. Estève 2018, G. Lo-Cascio 2022, K. Lim 2023]*
  - Geometry: from circular to **shaped plasmas** + ripple *[E. 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, Z. Qu 2026]*
  - ECRH sources *[P. Donnel 2022]* & Kinetic plasma neutrals inelastic collisions *[PhD M. Protais 2024-2027]*

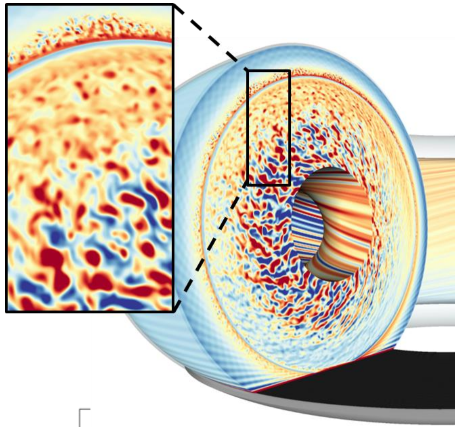


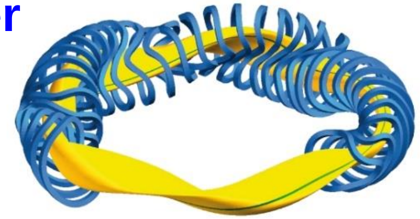
Roadmap for Gysela-X++ towards exascale

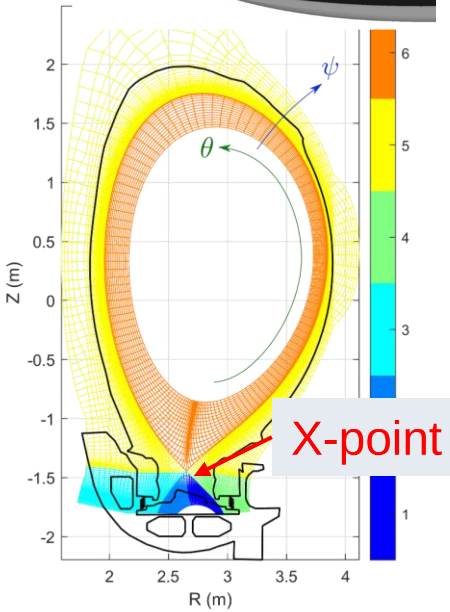
→ Why do we choose to rewrite GYSELA ?

- **Gysela-X++** = GYSELA in modern C++ with **X-point** for **exascale** ITER core-edge turbulence simulations
  (+ **Stellarator** via BPI / Renaissance Fusion)

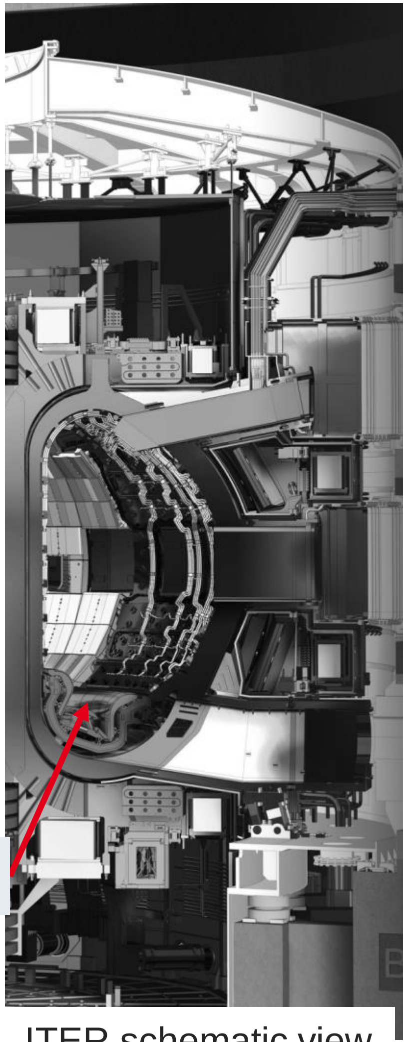
  - **Rewriting of the code in modern C++ with MPI + Kokkos**
    - Portable code on new exascale architectures
  - **Non-uniform meshes**
    - relevant density & temperature gradients at edge-SOL
  - Semi-Lagrangian scheme for **multi-patches**
    - **X-point** geometry
  - Implementation of a **3D scalable Poisson solver**
    - X-point & stellarator configuration
  - **Scalable I/O and in-situ diagnostics**





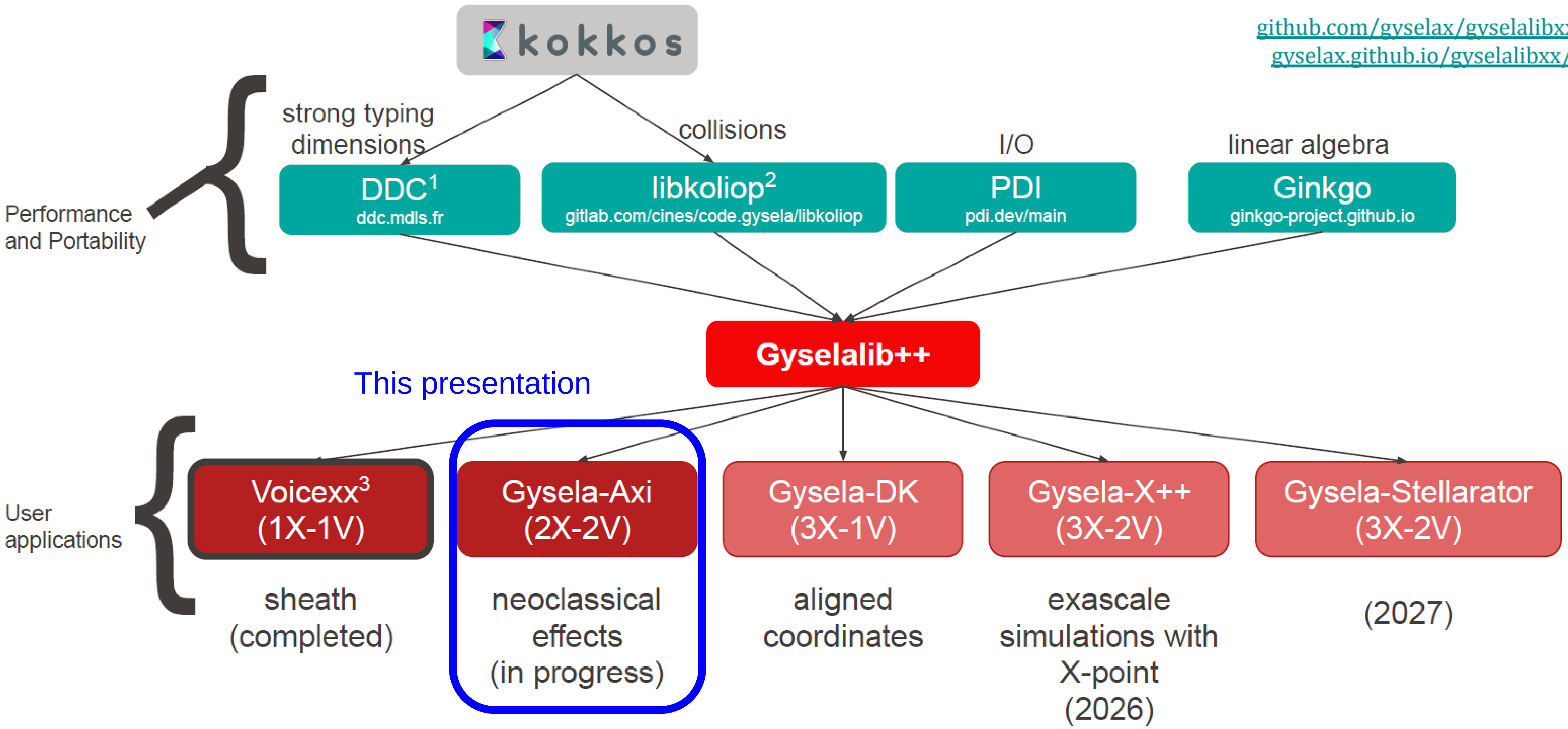


SOLEDGE-3X
X-point geometry


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

Outline

Advection equations = same as Gysela Fortran but assuming axi-symmetry (for now electrostatic)

- Spatial Advection

$$\frac{dx^i}{dt} = v_\parallel \boldsymbol{b}_s^\star \cdot \boldsymbol{\nabla} x^i + \boldsymbol{v}_{Ds} \cdot \boldsymbol{\nabla} x^i + \boldsymbol{v}_E \cdot \boldsymbol{\nabla} x^i$$

Drifts

$$\boldsymbol{v}_{Ds} \cdot \boldsymbol{\nabla} x^i = \left(\frac{m_s v_\parallel^2 + \mu B}{q_s B_{\parallel s}^\star B} \right) [B, x^i]$$

$$\boldsymbol{v}_E \cdot \boldsymbol{\nabla} x^i = \frac{1}{B_{\parallel s}^\star} [\phi, x^i]$$

Usage of general coordinates = ease the change of geometry (tokamak / stellarator)

$$[F, G] = \boldsymbol{b} \cdot (\boldsymbol{\nabla} F \times \boldsymbol{\nabla} G) = \frac{\epsilon^{ijk}}{J_x} \partial_i F \partial_j G b_k$$

- Advection in parallel direction

$$m_s \frac{dv_\parallel}{dt} = -\mu \boldsymbol{b}_s^\star \cdot \boldsymbol{\nabla} B - q_s \boldsymbol{b}_s^\star \cdot \boldsymbol{\nabla} \phi + \frac{m_s v_\parallel}{B} \boldsymbol{v}_E \cdot \boldsymbol{\nabla} B$$

Time-splitting for Boltzmann equation (= with given electric field)

- A time-splitting of Strang is applied to the 4D non-linear Boltzmann equation:

$$B^*_{\|s}\frac{\partial \bar{F}_s}{\partial t} + \nabla\cdot\left(\frac{d\mathbf{x_G}}{dt}B^*_{\|s}\bar{F}_s\right) + \frac{\partial}{\partial v_{G\|}}\left(\frac{dv_{G\|}}{dt}B^*_{\|s}\bar{F}_s\right) = \mathcal{C}(\bar{F}_s) + S$$

- Let us define three advection operators (with $\mathcal{X}_G = (r,\theta)$)

$$B^*_{\|s}\frac{\partial \bar{F}_s}{\partial t} + \boldsymbol{\nabla}\cdot\left(B^*_{\|s}\frac{d\mathcal{X}_G}{dt}\bar{F}_s\right) = 0 \qquad : (\tilde{\mathcal{X}}_G)$$

$$B^*_{\|s}\frac{\partial \bar{F}_s}{\partial t} + \frac{\partial}{\partial\varphi}\left(B^*_{\|s}\frac{d\varphi}{dt}\bar{F}_s\right) = 0 \qquad : (\tilde{\varphi})$$

Axi-symmetry

$$B^*_{\|s}\frac{\partial \bar{F}_s}{\partial t} + \frac{\partial}{\partial v_{G\|}}\left(B^*_{\|s}\frac{dv_{G\|}}{dt}\bar{F}_s\right) = 0 \qquad : (\tilde{v_{G\|}})$$

➠ Semi-Lagrangian scheme

When going to 5D, just need to add a 1D advection in the toroidal direction

- And the collision operator ($\tilde{\mathcal{C}}$) on a Δt : $\partial_t\bar{F}_s = \mathcal{C}(\bar{F}_s)$ ➠ Crank-Nicolson
- And the source operator ($\tilde{\mathcal{S}}$) on a Δt : $\partial_t\bar{F}_s = \mathcal{S}$ ➠ Crank-Nicolson
- Then, a Boltzmann solving sequence ($\tilde{\mathcal{B}}$) is performed:

$$(\tilde{\mathcal{B}}) \equiv (\tilde{\mathcal{S}},\, \tilde{\mathcal{C}})\left(\frac{\tilde{v_{G\|}}}{2},\, \frac{\tilde{\varphi}}{2},\, \tilde{\mathcal{X}}_G,\, \frac{\tilde{\varphi}}{2},\, \frac{\tilde{v_{G\|}}}{2}\right)$$

Coupling with quasi-neutrality :predictor-corrector scheme

- Prediction:
  - Computes $\mathrm{F}(t_n + 0.5\Delta\mathrm{t})$ solving the Boltzmann equation with $\Phi(t_n)$
  - Solves the quasi-neutrality equation to compute $\Phi(t_n + 0.5\Delta\mathrm{t})$
- Correction
  - Computes $\mathrm{F}(t_n + \Delta\mathrm{t})$ solving the Boltzmann equation with $\Phi(t_n + 0.5\Delta\mathrm{t})$
  - Solves the quasi-neutrality equation to compute $\Phi(t_n + \Delta\mathrm{t})$ for next time step

Quasi-neutrality equation (for now full kinetic electrons only)

$$-\boldsymbol{\nabla}_\perp \cdot \left(\alpha \boldsymbol{\nabla}_\perp \phi^{FKE}\right) = \sum_i e_i \delta \bar{n}_i - e \delta \bar{n}_e \qquad \alpha_s = \frac{m_s \langle \bar{n}_s \rangle_\varphi}{\langle B \rangle_\varphi^2}$$

Gysela-axi: a testbed for new developments

New features can be tested with a reduced cost/complexity:

- X-point geometry
- More realistic boundary conditions
- Neutrals (M. Protais's presentation)
- Sources (e.g. ECRH)

Theoretical predictions for GAM dynamics and neoclassical are done with axisymmetry: good framework for verification

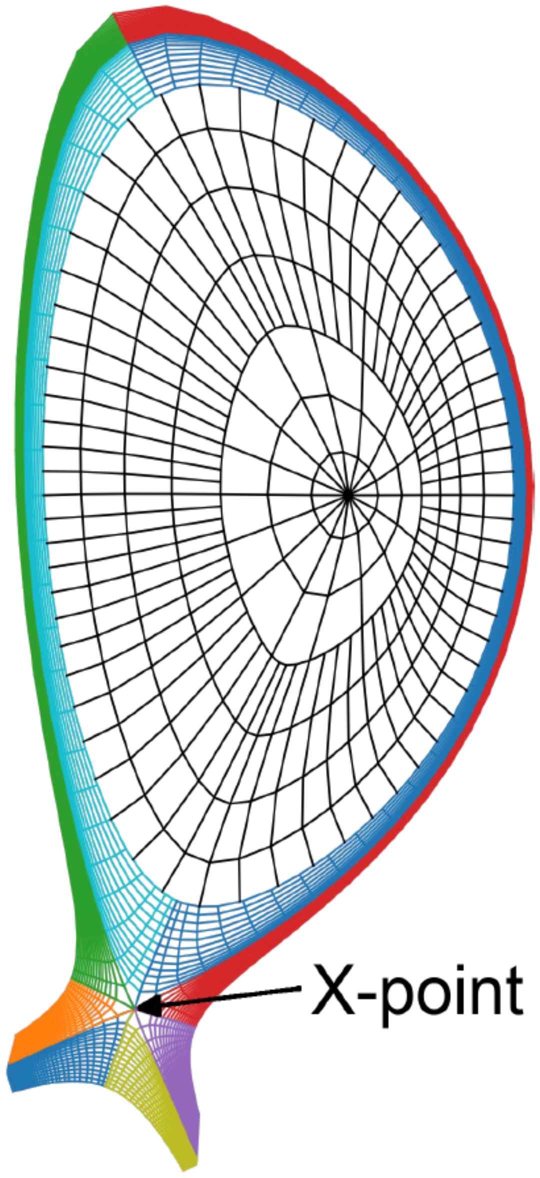
Also a good testbed for new algorithms / new solvers / improve numerical efficiency

Handling an X-point

Patches are required to handle 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



NB: Same grid generation as SOLEDGE

Gysela-axi as initial condition for Gysela-X++

One difficulty for global flux-driven gyrokinetic codes is to set the initial condition. If the choice is bad (= F0 far from steady state in the δf formalism), large transient can make the code crash. Even without a crash, the numerical cost to solve this transients in 5D is probably useless.

For electrostatic core simulation, we know a steady state solution: canonical Maxwellian

In presence of X-point / neutrals: no analytical expressions even without turbulence

→An axisymmetric code is probably good enough to solve the transients with a limited numerical cost

Outline

Status and next steps for Gysela-axi

A first electrostatic version with full kinetic electrons has been implemented (with B* = B). Collisions are already included [E. Malaboeuf 2025]

This version is currently being verified:

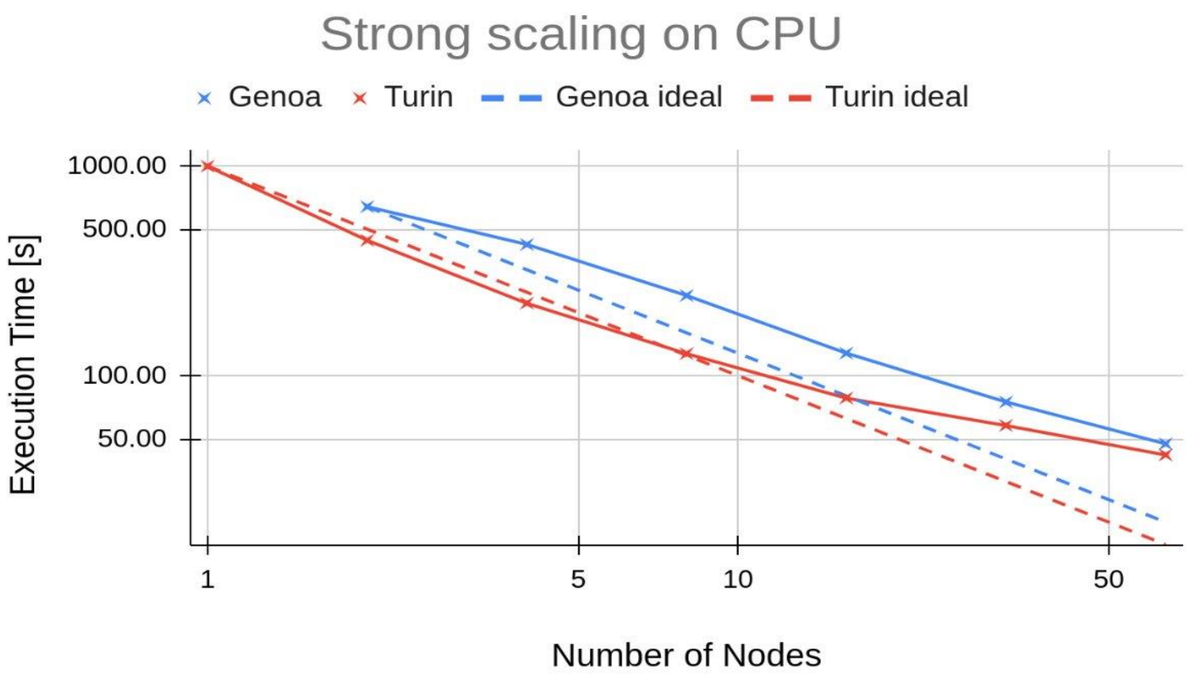
- GAM damping & residual (collisionless) → problem in the advection near the magnetic axis detected, under resolution
- Neoclassical physics

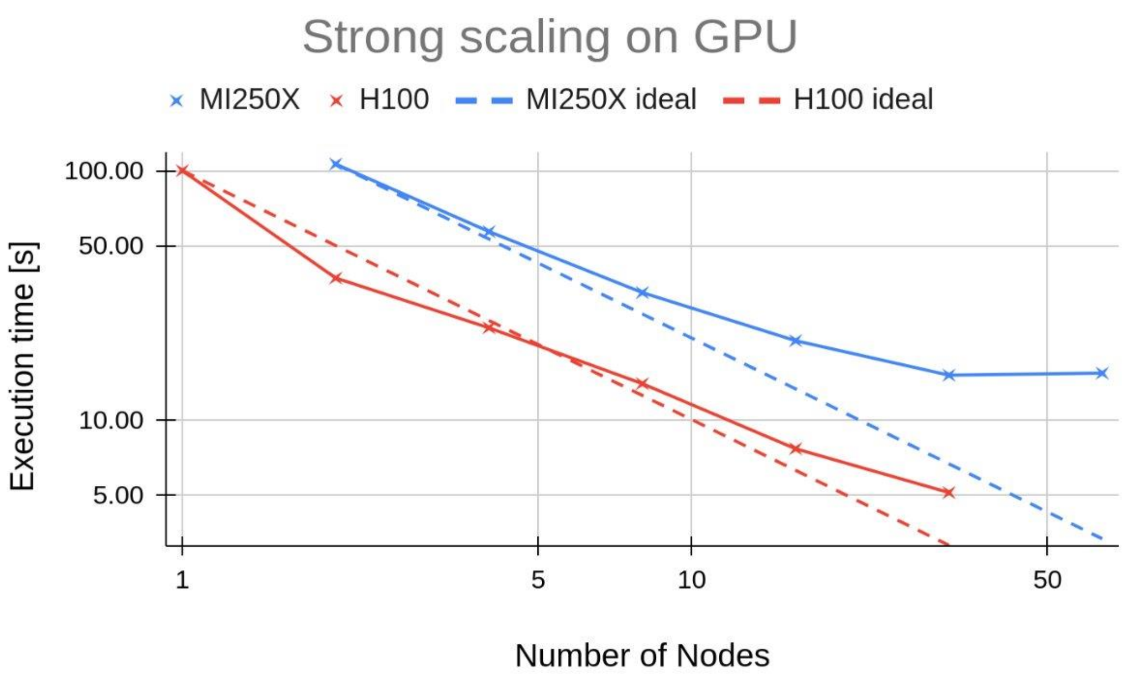
Next steps of implementation (should be quite fast as identical to the Fortran version):

- B* corrections
- Alternative models of electrons: adiabatic & hybrid [P. Donnel 2025]
- Simplified limiter version (immersed boundary conditions)
- Electromagnetic version (A// only) [Z. Qu 2026]

Gyselalib++ : Kokkos backend = **efficient portability on both CPU and GPU** architectures

- Kokkos strategie ensures efficient portability on both CPU and (AMD, NVIDIA) GPU architectures
- Promising results for Landau4D case (256 x 256 x 256 x 256) up to 64 nodes (512 GPU)
  - Adastra (CINES/France) : AMD CPU Genoa + AMD GPU MI250X
  - Pitagora (CINECA/Italy) : AMD CPU Turin + NVIDIA GPU H100





[E. Bourne, 2026]

Conclusions

- **Gysela-X++** = GYSELA in modern C++ with **X-point** for **exascale** ITER core-edge turbulence simulations
  - Require to harvest exascale supercomputers: usage of state of the art HPC libraries (Kokkos, DDC, PDI…)
  - Development of new capacities, hard to implement in the current Fortran version (X-point, neutrals, patches)

- **Gysela-axi** = a major milestone in this project
  - Test the implementation / strategy of development
  - Perfect testbed for the developments of new features with a limited numerical cost
  - Foreseen to be routinely used in pair with the 5D version to lower the overall numerical cost

- **Gysela-stellarator** = another code aiming at simulating core stellarator plasmas will also be developed in parallel to Gysela-X++

Acknowledgments

https://gyselax.github.io/

Gysela-X Team

Permanent Staff


Abdelhadi Kara
CEA
IRFM


Dr. Emily Bourne
EPFL
SCITAS


Dr. Julien Bigot
CEA
Maison de la Simulation


Dr. Kevin Obrejan
CEA
IRFM


Dr. Peter Donnel
CEA
IRFM


Dr. Philipp Krah
CEA
IRFM


Dr. Thomas Padioleau
CEA
Maison de la Simulation


Dr. Virginie Grandgirard
CEA
IRFM


Dr. Yuuichi Asahi
CEA
Maison de la Simulation


Dr. Kyungtak Lim
Nanyang Technological University


Dr. Xavier Garbet
CEA
IRFM
Nanyang Technological University


Dr. Yanick Sarazin
CEA
IRFM


Dr. Guilhem Dif-Pradalier
CEA
IRFM


Dr. Zhisong Qu
Nanyang Technological University

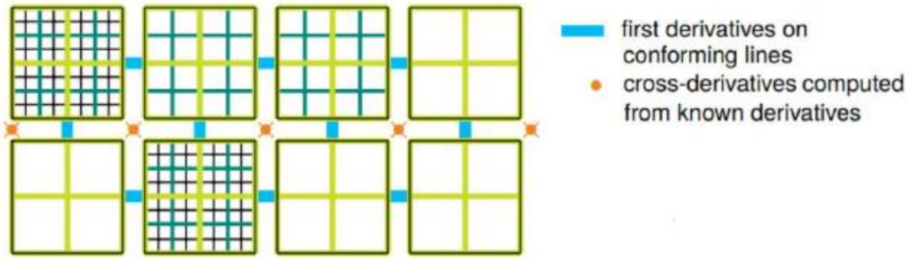
Non-permanent Staff

Currently: R. Bigué, A. De Vora, L. De Gianni, M. Go, A. Hoffmann, E. Malaboeuf, M. Protais, R. Varennes, P. Vidal, …

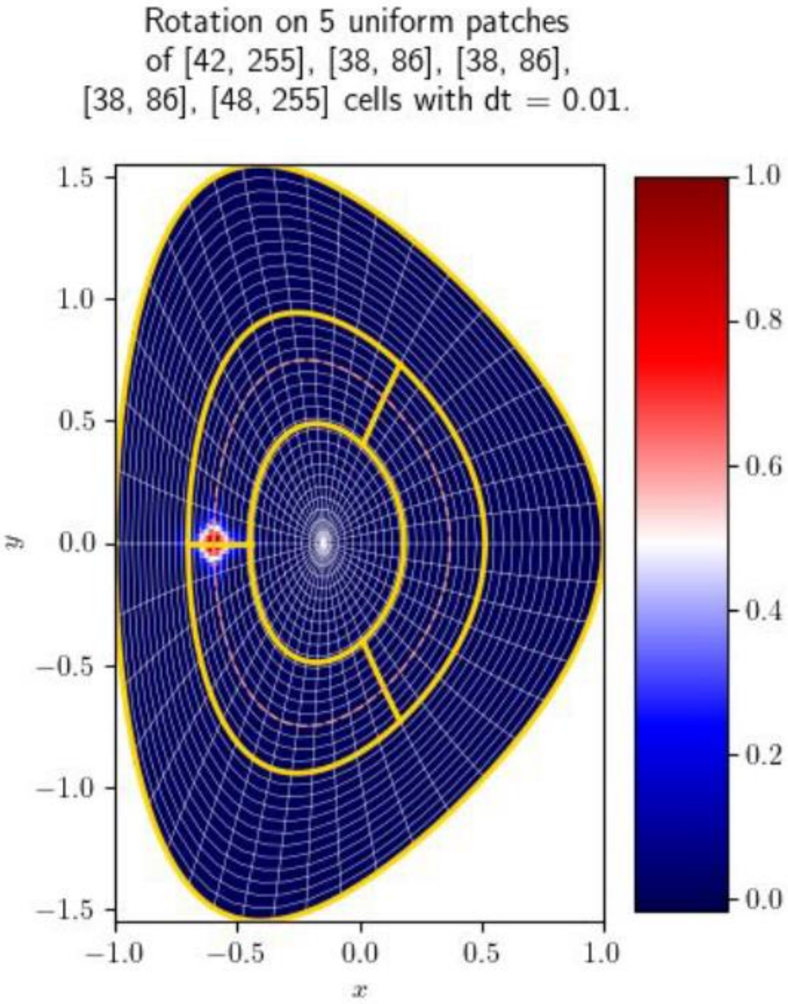
Backup slides

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 2022-2025]