MAX-PLANCK-INSTITUT
FÜR PLASMAPHYSIK

Vlasov-type equations solved with a backward semi-Lagrangian method on a multi-patch geometry

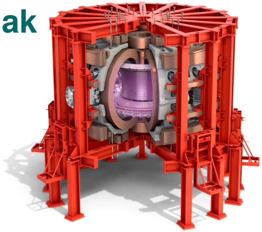
Pauline Vidal [1], Emily Bourne [2], Virginie Grandgirard [3], Michel Mehrenberger [4,5], and Eric Sonnendrücker [1,6].

[1] IPP, Max-Planck Institut for Plasma Physics
[2] EPFL, École Polytechnique Fédérale de Lausanne
[3] CEA, IRFM
[4] I2M, Institut de Mathématique de Marseille
[5] AMU, Aix-Marseille University
[6] TUM, Technische Universität München

EUROfusion

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

Context of application: plasma evolution in a tokamak



source: ASDEX, https://ipp.mpg.de

- Simulation of particle motion in a plasma with magnetic confinement.

- Equations: [LMG+18, GAB+16, GBB+06],

Time evolution of the particles governed by the **Vlasov equation**,
6D space: $(x, v) \in \mathbb{R}^6$, with $x \in \mathbb{R}^3, v \in \mathbb{R}^3$

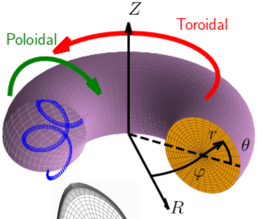
$$\partial_t f(t, \mathbf{x}, \mathbf{v}) + \mathbf{v} \cdot \nabla_{\mathbf{x}} f(t, \mathbf{x}, \mathbf{v}) + \frac{q}{m} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_{\mathbf{v}} f(t, \mathbf{x}, \mathbf{v}) = 0, \qquad (1)$$

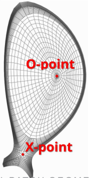
Gyrokinetic approximation, 5D space: $(x, v_{||}, \mu) \in \mathbb{R}^5$, with $x \in \mathbb{R}^3, v_{||}, \mu \in \mathbb{R}$

$$\partial_t f(t, \mathbf{x}, v_{||}, \mu) + \mathbf{v}_{GC}(t, \mathbf{x}, v_{||}, \mu) \cdot \nabla_{\mathbf{x}} f(t, \mathbf{x}, v_{||}, \mu) + a_{GC}(t, \mathbf{x}, v_{||}, \mu) \partial_{v_{||}} f(t, \mathbf{x}, v_{||}, \mu) = 0, \qquad (2)$$

Guiding-center, **2D space**: $(r, \theta) \in \mathbb{R}^2$,

$$\boxed{\partial_t f(t, r, \theta) + \mathbf{v}_{GC}(t, r, \theta) \cdot \nabla f(t, r, \theta) = 0,} \qquad (3)$$

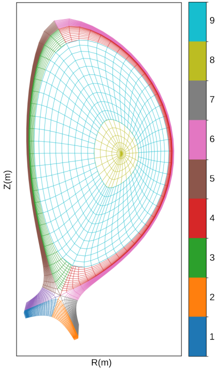




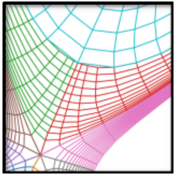
General motivation: special geometries and local refinements

Why using multi-patch geometry?

We align the mesh lines on the flux surfaces of the magnetic field $\Rightarrow$ separate the poloidal plane into different zones.

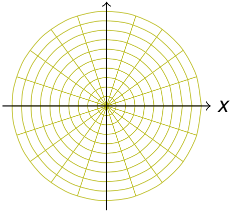


Zoom on the non-conforming mesh



Global domain split into 9 patches. Mesh for the ITER tokamak based on data from SOLEDGE3X and generated by K. Obrejan (`https://soledge3x.onrender.com`, article [RTM+25]).

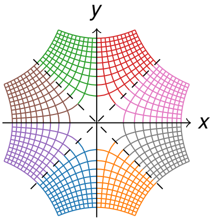
Zoom on a reference O-point in the core.



$$\begin{cases} x = r\cos(\theta) \\ y = r\sin(\theta) \end{cases}$$

→ Studied in [ZG19].
→ Studied in [Bou22].

Zoom on a reference X-point at the separatrix intersection.



$$\begin{cases} x = sgn(x)\sqrt{\sqrt{u^2+v^2} - u} \\ y = sgn(y)\sqrt{\sqrt{u^2+v^2} + u} \end{cases}$$

→ Work in progress.

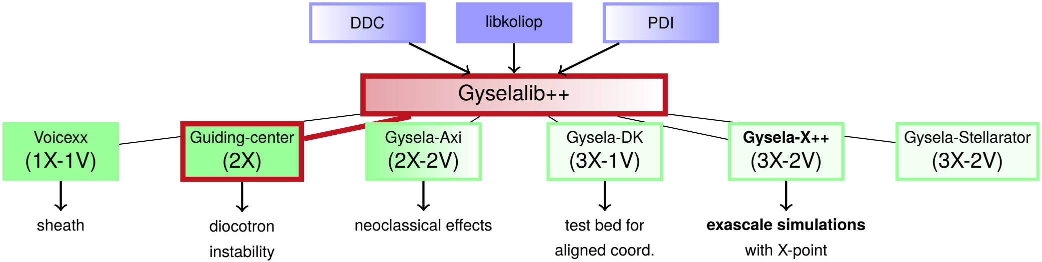
- E. Bourne. *Non-Uniform Numerical Schemes for the Modelling of Turbulence in the 5D GYSELA Code* [BMG+23]
- E. Zoni et al. *Solving hyperbolic-elliptic problems on singular mapped disk-like domains with the method of characteristics and spline finite elements*, [ZG19]

Implementation in Gyselalib++

Gyrokinetic Semi-Lagrangian code

- Fortran version: **GYSELA** [GAB+16]
  - **Gyrokinetic plasma turbulence simulations** to understand turbulent transport that mainly governs confinement in Tokamaks.
  - GYSELA developed at IRFM/CEA code over 20 years thanks to strong EU collaboration between physicists, mathematicians and HPC specialists. (**Petascale** simulations.)

- C++ version: **Gyselalib++** *Start with European project EoCoE-III (2024 - 2027)*.
  - **Exascale** needs for ITER plasma turbulence simulations.
  - + Modular + More portable on exascale architectures + More physics.



GitHub: `https://github.com/gyselax/gyselalibxx`

Plan

P. Vidal, E. Bourne, V. Grandgirard, M. Mehrenberger, E. Sonnendrücker,
Local cubic spline interpolation for Vlasov-type equations on a multi-patch geometry.
[SUBMITTED], Available on `http://arxiv.org/abs/2505.22078`
[VBG+25]

I. Semi-Lagrangian scheme on multi-patch

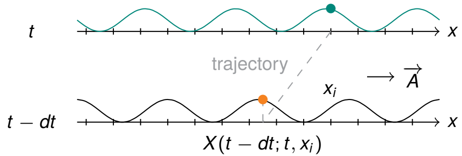
1. Backward semi-Lagrangian (BSL) method

Advection equation:

$$\begin{cases} \partial_t f(t,x) + A(t,x)\cdot\nabla f(t,x) = 0, & t \geq 0, x \in \Omega \\ f(t=0,x) = f_0(x), & x \in \Omega \end{cases} \tag{4}$$

Characteristics' equation: → conservation of f.

$$\begin{cases} \partial_s X(s;t,x) = A(s, X(s;t,x)), & s \geq 0, x \in \Omega \\ X(t;t,x) = x, & x \in \Omega \end{cases} \tag{5}$$

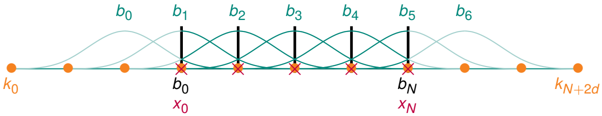


- **Solve backward** the characteristics' equation.
- **Interpolate** the function at $t - dt$ at the feet of the characteristics: $X(t-dt;t,x_i)$.

2. Interpolation with cubic B-splines

spline = piecewise cubic polynomials
1D formula:

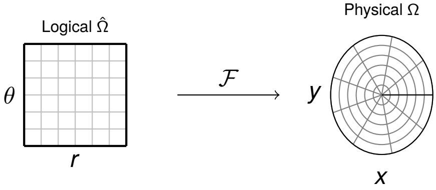
$$s(x) = \sum_{k=0}^{N+d} c_k b_k(x), \quad x \in \Omega \tag{6}$$



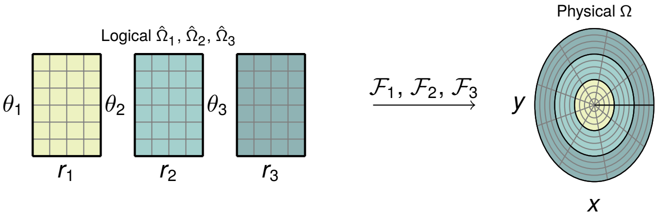
I.3. Logical space and physical space

We want to align the mesh lines on the flux surfaces of the **magnetic field**.

- Global domain:



- Multi-patch:



▷ 2D spline = tensor product between two 1D splines.

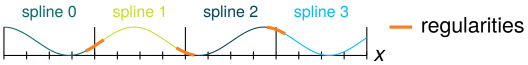
▷ B-splines defined on the logical domain.

▷ Advection computed in the physical domain.

▷ **Connection between patches?**

II. Semi-Lagrangian scheme on multi-patch

1. Local spline regularities at the interfaces and generalisation



! $\mathcal{C}^0 \rightarrow$ instabilities.

▶ $\mathcal{C}^1 \rightarrow$ Hermite boundary conditions.

▶ N. Crouseilles et al. *A parallel Vlasov solver based on local cubic spline interpolation on patches.* [CLS09]

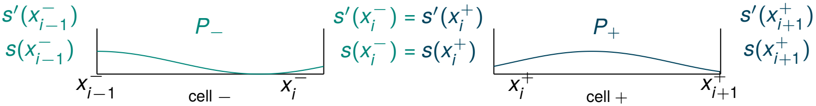
$$s'(x_i) = \sum_{j=-10}^{10} \frac{\tilde{\omega}_j}{\Delta x} f_{i+j}, \tag{7}$$

with given weigths $\{\tilde{\omega}_j\}_j$ ($\tilde{\omega}_0 = 0$), and Δx the uniform cell length.

Assumptions/limitations: ▶ Study only applied to **cubic** splines. ▶ {Break points} $\subset$ {interpolation points}.

Hermite interpolation polynomials: (H_0, H_1, K_0 and K_1 cubic polynomials.)

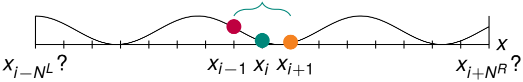
$$P(a + x(b-a)) = f(a)H_0(x) + f(b)H_1(x) + (b-a)f'(a)K_0(x) + (b-a)f'(b)K_1(x). \tag{8}$$



Imposing $\mathcal{C}^2$ continuity $P''_-(x_i^-) = P''_+(x_i^+)$ enforces the Hermite polynomials to be **equal** to an equivalent global cubic spline defined on both cells.

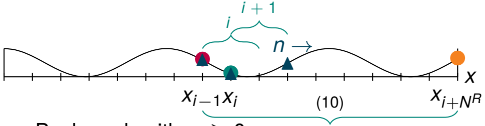
II.2. Generalisation of the formula with Hermite polynomial
Relation between three derivatives

$\forall N^L, N^R > 1$ number of cells,

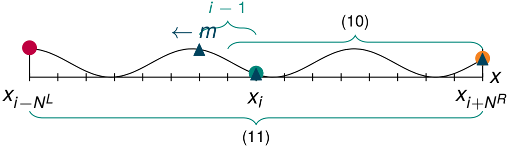


Recursivity:

- Forward: with $n \geq 0$,



- Backward: with $m \geq 0$



Imposing $\mathcal{C}^2$ continuity $P''_-(x_i^-) = P''_+(x_i^+)$ gives

$$s'(x_i) = \gamma_i + \alpha_i s'(x_{i+1}) + \beta_i s'(x_{i+1}), \tag{9}$$

with $\alpha_i, \beta_i, \gamma_i \in \mathbb{R}$, $\gamma_i = \sum_{k=-1}^{1} \gamma_{i,k} f_{i+k}$,

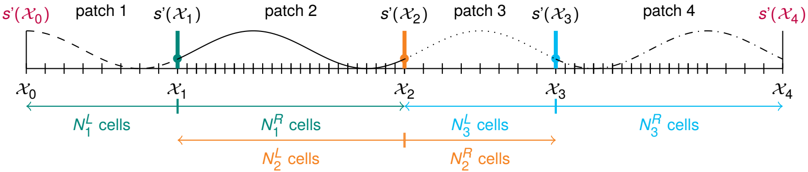
$$s'(x_i) = c^i_{1,n} + a^i_{1,n} s'(x_{i+n}) + b^i_{1,n} s'(x_{i-1}), \tag{10}$$

$$s'(x_i) = c^i_{m,N^R} + a^i_{m,N^R} s'(x_{i+N^R}) + b^i_{m,N^R} s'(x_{i-m}) \tag{11}$$

with $a^i_{m,n}, b^i_{m,n}, c^i_{m,n} \in \mathbb{R}$, $c^i_{m,n} = \sum_{k=-m}^{n} \omega^i_{k,m,n} f_{i+k}$,

II.2. Generalisation of the formula with Hermite polynomial
Application to multi-patch: use a matrix system

For each interface I, $$s'(\mathcal{X}_I) = c^I + a^I s'(\mathcal{X}_{I+1}) + b^I s'(\mathcal{X}_{I-1}) \quad (12)$$



(Notation: $a^I = a^I_{N_I^R, N_I^L}$.)

For $N_p = 4$ patches,
$N_p + 1 = 5$ interfaces,

$$\mathbf{S} = \begin{bmatrix} s'(\mathcal{X}_1) \\ \vdots \\ s'(\mathcal{X}_I) \\ \vdots \\ s'(\mathcal{X}_{N_p-1}) \end{bmatrix} = \begin{bmatrix} 1 & -a^1 & & \\ -b^2 & 1 & -a^2 & \\ & \ddots & \ddots & \ddots \\ & & -b^{N_p-1} & 1 \end{bmatrix}^{-1} \begin{bmatrix} c^1 + b^1 s'(\mathcal{X}_0) \\ \vdots \\ c^I \\ \vdots \\ c^{N_p-1} + a^{N_p-1} s'(\mathcal{X}_{N_p}) \end{bmatrix} = (\mathbb{I} - \mathbb{M})^{-1}\mathbf{C} \quad (13)$$

Matrix size $N_p - 1 \times N_p - 1 \rightarrow$ small matrix system!

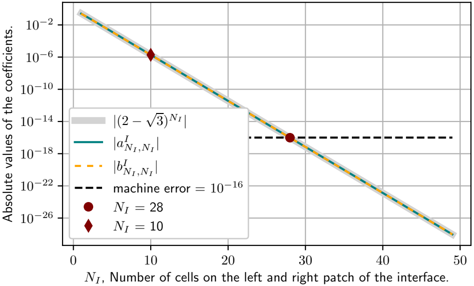
Exact derivatives!

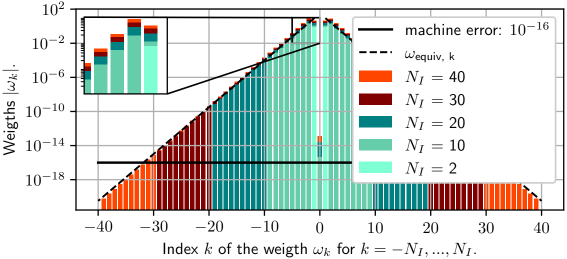
II.3. Approximation of the exact 1D formula

Behaviour of (c^I, a^I, b^I) when $N_I^L, N_I^R \to \infty$?

$$(12): \quad s'(\mathcal{X}_I) = c^I + a^I s'(\mathcal{X}_{I+1}) + b^I s'(\mathcal{X}_{I-1})$$

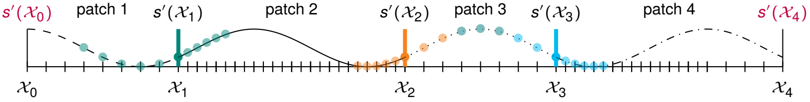
- For $N_I = N_I^L = N_I^R$, $\quad c^I = \sum_{k=-N_I^L}^{N_I^R} \omega^I_{k,N_I^L,N_I^R} f_{i+k} \longrightarrow c^{I,*} = \sum_{k=-N^L_{I,trunc}}^{N^R_{I,trunc}} \omega^I_{k,N^L_{I,trunc},N^R_{I,trunc}} f_{i+k}.$





II.3. Approximation of the exact 1D formula

$$(12): \quad s'(\mathcal{X}_l) = c^l + a^l s'(\mathcal{X}_{l+1}) + b^l s'(\mathcal{X}_{l-1})$$



For $N_l^L, N_l^R > 30, \forall l$,

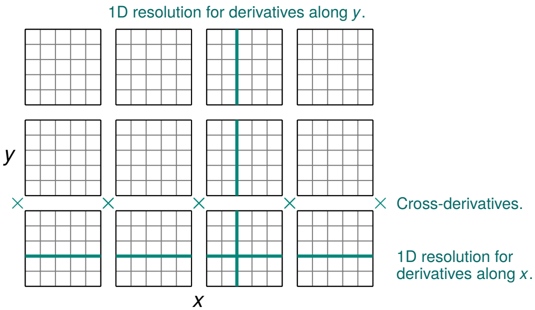
$$\mathbf{S} \to \mathbf{C}_{\text{trunc}} = \begin{bmatrix} c^{1,*} \\ \vdots \\ c^{N_p-1,*} \end{bmatrix}, \qquad s(\mathcal{X}_l) \simeq \sum_{k=-30}^{30} \omega^l_{k,30,30} f_{i+k} \tag{14}$$

→ **Pseudo-local systems.**

II. 4. 2D Multi-patch

Conforming case: 2D splines are a tensor product of 1D splines ⇒ direct application of the 1D case.

1D: $s'(\mathcal{X}_I) = c^I + a^I s'(\mathcal{X}_{I+1}) + b^I s'(\mathcal{X}_{I-1})$

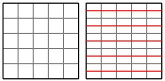


$$\partial_x s(\mathcal{X}_I, y_j) = c^I + a^I \partial_x s(\mathcal{X}_{I+1}, y_j) + b^I \partial_x s(\mathcal{X}_{I-1}, y_j)$$

$$\partial_y s(x_i, Y_J) = c^J + a^J \partial_y s(x_i, Y_{J+1}) + b^J \partial_y s(x_i, Y_{J-1})$$

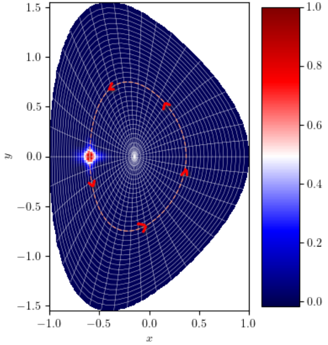
$$\partial_{xy} s(\mathcal{X}_I, y_j) = c^I_{\partial y} + a^I \partial_{xy} s(\mathcal{X}_{I+1}, y_j) + b^I \partial_{xy} s(\mathcal{X}_{I-1}, y_j)$$

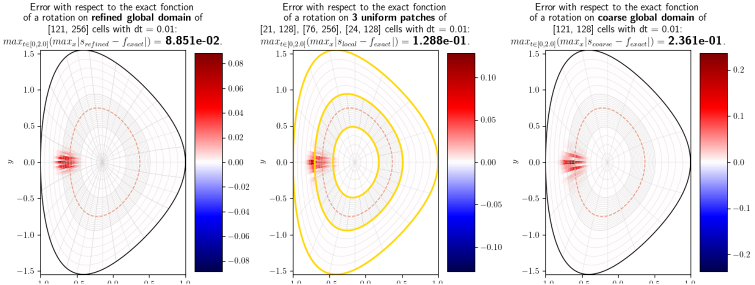
Non-conforming case: Not always an order to compute the derivatives and cross-derivatives using all the information.
⟶ Approximation may be a better solution for some cases.

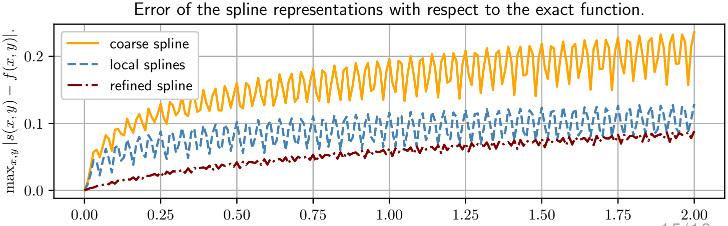


III. Numerical results

III. Numerical results
1. Advection on non-conforming mesh

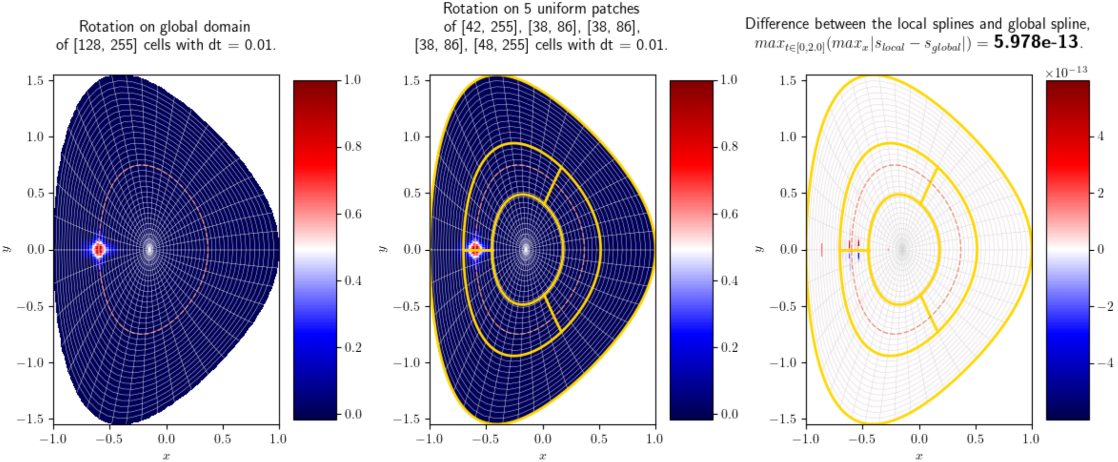






III. Numerical results
2. Advection on mesh with T-joints

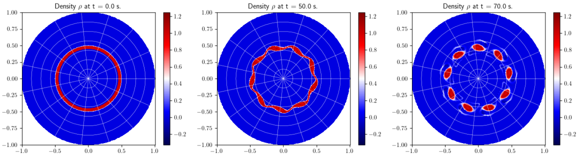
T-joint = connection between at least three patches, where an edge of one patch is not fully connected to the edge of another patch along its entire length.



Global spline Local splines Difference

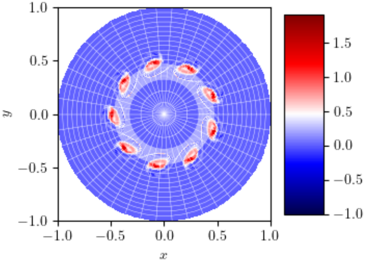
III. Numerical results
3.Diocotron on non-uniform mesh with T-joints

$$\begin{cases} \partial_t \rho + \dfrac{E \times B}{|B|^2} \cdot \nabla \rho = 0, \\ \Delta \phi = -\rho, \qquad \phi = 0 \text{ on } \partial\Omega \end{cases} \tag{15}$$

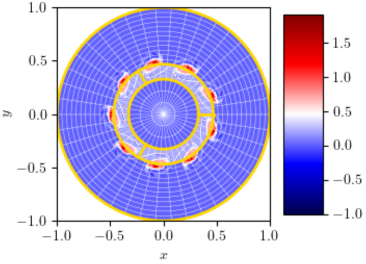


▶ Local refinement with T-joints + "ill-placed" interface.

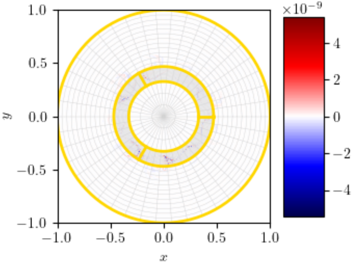
Diocotron instability simulation on global domain of [124, 255] cells with dt = 0.1.



Diocotron instability simulation on 5 uniform patches of [4, 255], [72, 85], [72, 85], [72, 85], [48, 255] cells with dt = 0.1.



Difference between the local splines and global spline, $max_{t\in[0,70]}(max_x|s_{local} - s_{global}|) =$ **5.419e-09**.



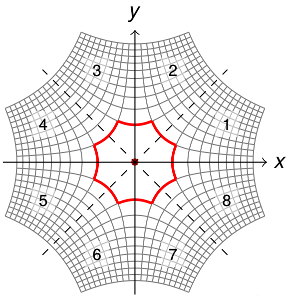
V. Conclusion and perspectives

- Advection on a multi-patch geometry (with BSL + B-splines) $\Rightarrow$ compute derivatives at the inner interfaces.
- Generalisation of formula $s'(x_i) = \sum_{j=-10}^{10} \frac{\tilde{\omega}_j}{\Delta x} f_{i+j}$ (from N.Crouseilles, G.Latu and E. Sonnendrücker's article. [CLS09])
- Provide exact and approximation formulas.
- Work on
  - uniform and **non-uniform** meshes.
  - conforming and **non-conforming** meshes.
  - meshes with T-joints.

P. Vidal et al. *Local cubic spline interpolation for Vlasov-type equations on a multi-patch geometry.* Submitted, available on `http://arxiv.org/abs/2505.22078`,

---

- Study of interpolation on geometries with **X-point**.
- Application to higher dimension equations
  (e.g. drift-kinetic equations (2X - 2V), ...)
- Optimization for exascale simulations
  (porting on GPU work in progress).



References

Emily Bourne, Yann Munschy, Virginie Grandgirard, Michel Mehrenberger, and Philippe Ghendrih, *Non-uniform splines for semi-lagrangian kinetic simulations of the plasma sheath*, Journal of Computational Physics **488** (2023), 112229.

Emily Bourne, *Non-uniform numerical schemes for the modelling of turbulence in the 5d gysela code*, Ph.D. thesis, ED 184 MATHEMATIQUES ET INFORMATIQUE, December 2022.

Nicolas Crouseilles, Guillaume Latu, and Eric Sonnendrücker, *A parallel vlasov solver based on local cubic spline interpolation on patches*, Journal of Computational Physics **228** (2009), no. 5, 1429–1446.

V. Grandgirard, J. Abiteboul, J. Bigot, T. Cartier-Michaud, N. Crouseilles, G. Dif-Pradalier, Ch. Ehrlacher, D. Esteve, X. Garbet, Ph. Ghendrih, G. Latu, M. Mehrenberger, C. Norscini, Ch. Passeron, F. Rozar, Y. Sarazin, E. Sonnendrücker, A. Strugarek, and D. Zarzoso, *A 5d gyrokinetic full-f global semi-lagrangian code for flux-driven ion turbulence simulations*, Computer Physics Communications **207** (2016), 35–68.

V. Grandgirard, M. Brunetti, P. Bertrand, N. Besse, X. Garbet, P. Ghendrih, G. Manfredi, Y. Sarazin, O. Sauter, E. Sonnendrücker, J. Vaclavik, and L. Villard, *A drift-kinetic semi-lagrangian 4d code for ion turbulence simulation*, Journal of Computational Physics **217** (2006), no. 2, 395–423.

Guillaume Latu, Michel Mehrenberger, Yaman Güçlü, Maurizio Ottaviani, and Eric Sonnendrücker, *Field-aligned interpolation for semi-lagrangian gyrokinetic simulations*, Journal of Scientific Computing **74** (2018), 1601–1650.

N. Rivals, P. Tamain, Y. Marandet, X. Bonnin, J.-S. Park, H. Bufferand, R.A. Pitts, G. Falchetto, H. Yang, and G. Ciraolo, *Soledge3x full vessel plasma boundary simulations of iter non-active phase plasmas*, Nuclear Fusion **65** (2025), no. 2, 026038.

Pauline Vidal, Emily Bourne, Virginie Grandgirard, Michel Mehrenberger, and Eric Sonnendrücker, *Local cubic spline interpolation for vlasov-type equations on a multi-patch geometry*, . (2025).

Thank you for your attention!

APPENDIX: General of the formula with Hermite polynomials

Exact relation of derivatives at three consecutives points

Imposing $\mathcal{C}^2$ continuity $P''_-(x_i^-) = P''_+(x_i^+)$ gives

$$\boxed{s'(x_i^-) = \gamma_i + \alpha_i s'(x_{i+1}^+) + \beta_i s'(x_{i-1}^-),} \tag{16}$$

with

$$\begin{cases} \gamma_i = \dfrac{3}{2}\dfrac{1}{\Delta x_i^+ + \Delta x_i^-}\left[\dfrac{\Delta x_i^-}{\Delta x_i^+} f_{i+1}^+ + \left(\dfrac{\Delta x_i^+}{\Delta x_i^-} - \dfrac{\Delta x_i^-}{\Delta x_i^+}\right) f_i^- - \dfrac{\Delta x_i^+}{\Delta x_i^-} f_{i-1}^-\right] \\ \alpha_i = \dfrac{-1}{2}\dfrac{\Delta x_i^-}{\Delta x_i^+ + \Delta x_i^-} \\ \beta_i = \dfrac{-1}{2}\dfrac{\Delta x_i^+}{\Delta x_i^+ + \Delta x_i^-} \end{cases} \tag{17}$$

with $\Delta x_i^+ = x_{i+1}^+ - x_i^+$ and $\Delta x_i^- = x_i^- - x_{i-1}^-$. $\rightarrow$ not especially $\Delta x_i^+ = \Delta x_i^-$.

- Derivative equal to the derivative of a global spline defined on the two cells.

APPENDIX: Recursive formula for the coefficients - Forward

- **Forward:** with $n \geq 0$,

$$s'(x_i) = c^i_{1,n+1} + a^i_{1,n+1} s'(x_{i+n+1}) + b^i_{1,n+1} s'(x_{i-1}), \tag{18}$$

$$\begin{cases} c^i_{1,1} = \gamma_i \\ a^i_{1,1} = \alpha_i \\ b^i_{1,1} = \beta_i \end{cases}$$

$$\begin{cases} c^i_{1,2} = \dfrac{1}{1 - \alpha_i \beta_{i+1}} [\gamma_i + \alpha_i \gamma_{i+1}] \\ a^i_{1,2} = \dfrac{\alpha_i \alpha_{i+1}}{1 - \alpha_i \beta_{i+1}} \\ b^i_{1,2} = \dfrac{\beta_i}{1 - \alpha_i \beta_{i+1}} \end{cases}$$

$$\begin{cases} c^i_{n+1} = \dfrac{1}{1 - \beta_{i+n} \frac{a^i_{1,n}}{a^i_{1,n-1}}} \left[c^i_n + a^i_{1,n} \gamma_{i+n} - \beta_{i+n} \dfrac{a^i_{1,n}}{a^i_{1,n-1}} c^i_{1,n-1} \right] \\ a^i_{1,n+1} = \dfrac{a^i_{1,n} \alpha_{i+n}}{1 - \beta_{i+n} \frac{a^i_{1,n}}{a^i_{1,n-1}}} \\ b^i_{1,n+1} = \dfrac{1}{1 - \beta_{i+n} \frac{a^i_{1,n}}{a^i_{1,n-1}}} \left[b^i_n - \beta_{i+n} \dfrac{a^i_{1,n}}{a^i_{1,n-1}} b^i_{1,n-1} \right] \end{cases} \tag{19}$$

The c coefficient can be seen as a sum of $\{f_k\}_k$: $c^i_{1,n+1} = \sum_{k=-1}^{n} \omega^i_{k,1,n+1} f_{i+k}$.

APPENDIX: Recursive formula for the coefficients - Backward

- **Backward:** with $m \geq 0$,

$$s'(x_i) = c^i_{m+1,N^R} + a^i_{m+1,N^R} s'(x_{i+N^R}) + b^i_{m+1,N^R} s'(x_{i-(m+1)}) \tag{20}$$

$$\begin{cases} c^i_{2,N^R} = \dfrac{1}{1 - b^i_{1,N^R}\alpha_{i-1}} \left[c^i_{1,N^R} + b^i_{1,N^R}\gamma_{i-1} \right] \\ b^i_{2,N^R} = \dfrac{b^i_{1,N^R}\beta_{i-1}}{1 - b^i_{1,N^R}\alpha_{i-1}} \\ a^i_{2,N^R} = \dfrac{a^i_{1,N^R}}{1 - b^i_{1,N^R}\alpha_{i-1}} \end{cases} \qquad \begin{cases} c^i_{m+1,N^R} = \dfrac{1}{1 - \alpha_{i-m}\frac{b^i_{m,N^R}}{b^i_{m-1,N^R}}} \left[c^i_{m,N^R} + b^i_{m,N^R}\gamma_{i-m} - \alpha_{i-m}\dfrac{b^i_{m,N^R}}{b^i_{m-1,N^R}} c^i_{m-1,N^R} \right] \\ b^i_{m+1,N^R} = \dfrac{b^i_{m,N^R}\beta_{i-m}}{1 - \alpha_{i-m}\frac{b^i_{m,N^R}}{b^i_{m-1,N^R}}} \\ a^i_{m+1,N^R} = \dfrac{1}{1 - \alpha_{i-m}\frac{b^i_{m,N^R}}{b^i_{m-1,N^R}}} \left[a^i_{N,m} - \alpha_{i-m}\dfrac{b^i_{m,N^R}}{b^i_{m-1,N^R}} a^i_{m-1,N^R} \right] \end{cases} \tag{21}$$

The c coefficient can be seen as a sum of $\{f_k\}_k$: $c^i_{m+1,N^R} = \sum_{k=-(m+1)}^{N^R} \omega^i_{k,m+1,N^R} f_{i+k}$.

APPENDIX: Explicit formula for uniform patches

- For uniform per patch mesh, to lighten the notation $m = N_l^L$, $n = N_l^R$,

$$
\begin{cases}
a^l_{m,n} = \dfrac{(-1)^{n-1} u_1 a^l_{1,1} u_m}{u_m u_n + u_m u_{n-1} a^l_{1,1} + u_n u_{m-1} b^l_{1,1}}, \\
b^l_{m,n} = \dfrac{(-1)^{m-1} u_1 b^l_{1,1} u_n}{u_m u_n + u_m u_{n-1} a^l_{1,1} + u_n u_{m-1} b^l_{1,1}}, \\
c^l_{m,n} = \displaystyle\sum_{k=-m}^{n} \omega^l_{k,m,n} f_{i+k},
\end{cases}
,
\quad
\begin{cases}
\omega^l_{k,m,n} = 3(-1)^k \dfrac{\frac{a^l_{1,1}}{\Delta x+} u_m u_1}{u_m u_n + u_m u_{n-1} a^l_{1,1} + u_n u_{m-1} b^l_{1,1}}, & k = n, \\
\omega^l_{k,m,n} = 3(-1)^k \dfrac{\frac{a^l_{1,1}}{\Delta x+} u_m (u_{n-k+1} - u_{n-k-1})}{u_m u_n + u_m u_{n-1} a^l_{1,1} + u_n u_{m-1} b^l_{1,1}}, & k = 1, \dots, n-1, \\
\omega^l_{k,m,n} = 3(-1)^k \dfrac{\frac{a^l_{1,1}}{\Delta x+} u_m (u_n - u_{n-1}) - \frac{b^l_{1,1}}{\Delta x-} u_n (u_m - u_{m-1})}{u_m u_n + u_m u_{n-1} a_{1,1} + u_n u_{m-1} b^l_{1,1}}, & k = 0, \\
\omega^l_{k,m,n} = 3(-1)^{k+1} \dfrac{\frac{b^l_{1,1}}{\Delta x-} u_n (u_{m+k+1} - u_{m+k-1})}{u_m u_n + u_m u_{n-1} a^l_{1,1} + u_n u_{m-1} b^l_{1,1}}, & k = -(m-1), \dots, -1, \\
\omega^l_{k,m,n} = 3(-1)^{k+1} \dfrac{\frac{b^l_{1,1}}{\Delta x-} u_n u_1}{u_m u_n + u_m u_{n-1} a^l_{1,1} + u_n u_{m-1} b^l_{1,1}}, & k = -m,
\end{cases}
\tag{22}
$$

and $a^l_{1,1} = -\dfrac{1}{2} \dfrac{\Delta x^-}{\Delta x^+ + \Delta x^-}$, $b^l_{1,1} = -\dfrac{1}{2} \dfrac{\Delta x^+}{\Delta x^+ + \Delta x^-}$, and $u_k = (2+\sqrt{3})^k - (2-\sqrt{3})^k$, $\forall k \in \mathbb{N}$.