Solving Vlasov-like equations using the Semi-Lagrangian scheme on a 2D hexagonal mesh

Laura Mendoza, Eric Sonnendrücker

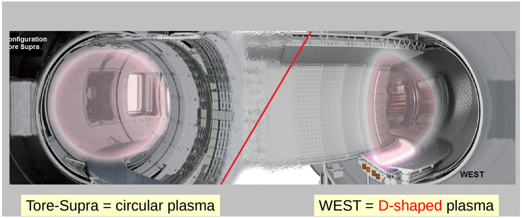
Max-Planck-Institut für Plasmaphysik

Friday 24th October, 2014

Table of contents

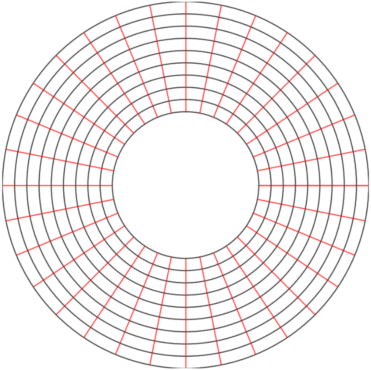
Motivation

The Gyrokinetic Semi-Lagrangian (**GYSELA**) code:



- **Gyrokinetic model**: 5D kinetic equation on the charged particules distribution
- 5 Dimensions: 2 in velocity space, 3 in configuration space
- **Simplified geometry**: concentric toroidal magnetic flux surfaces with circular cross-sections
- Based on the **Semi-Lagrangian** scheme

Motivation



Current representation of the poloidal plane :

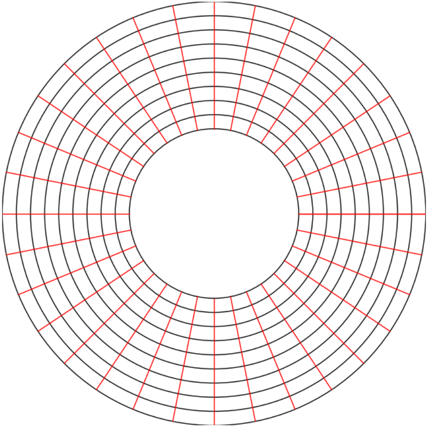
- Annular geometry
- **Polar mesh** (r, θ)

Some limitations of this choice :

- Geometric (and numeric) **singular point** at origin of mesh
- Unrepresented area and very costly to minimize that area
- Impossible to represent **complex geometries**

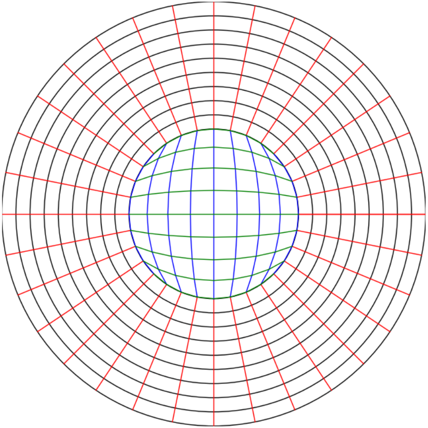
Multi-patch: the general idea

Our original mesh:



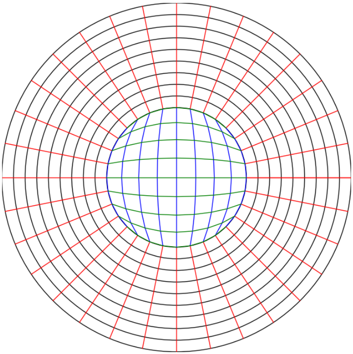
Multi-patch: the general idea

New representation of the poloidal plane:



Multi-patch: the general idea

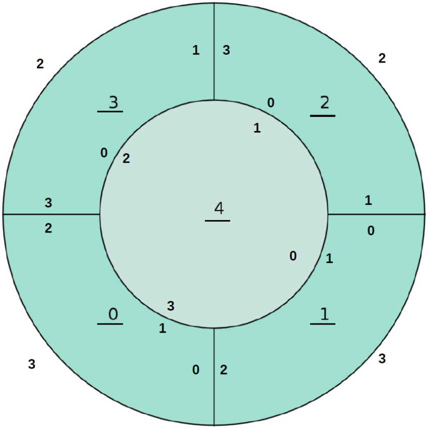
Specificities of the new geometry definition :



- **Additional patch(es)** with no singular point at origin
- Each patch defined as a **transformation** (or mapping) from uniform cartesian grid to new mesh
- Mappings defined with **NURBS** (Non-Uniform Rational B-Splines) $\Longrightarrow$ complex geometries
- **Coupling** between patches defined by boundary condition

The 5 patches configuration

External crown divided into 4 patches and the connectivity is defined as a patch-edge to patch-edge association (creation tool: **CAID**)



Advantages

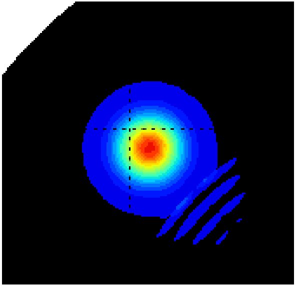
- Flexibility defining complex geometries
- Each patch can be treated separately
- No geometrical singularity

New challenges

- What is the best BC?
- How to treat particules which characteristics' origin are on another patch?
- **4 new numerical singularities**

Multi-patch: Some results

Results always showed instabilities near singular points. What we've tried to avoid them:

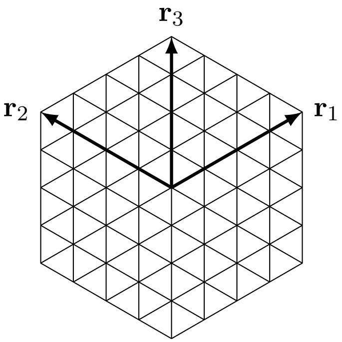


- Boundary conditions tested: strictly interdependent gradients and mean gradients between connecting patches
- Over-lapping: Impossible with interior patch and useless for others
- Squared internal mapping

Problem: Impossible to avoid singular points from mapping from a square to a circle

The hexagonal mesh

Idea: Use a new mapping: **hexagon** $\longrightarrow$ **circle**.
We define a tiling of triangles of a hexagon as our mesh for a 2D poloidal plane.



Some advantages:

- No singular points
- (Hopefully) no need of multiple patches for the core of the tokamak
- Twelve-fold symmetry $\Rightarrow$ more efficient programming
- Easy transformation from cartesian to hexagonal coordinates
- Easy mapping to a disk
  $\Rightarrow$ field aligned physical mesh

The Backward Semi-Lagrangian Method

We consider the advection equation

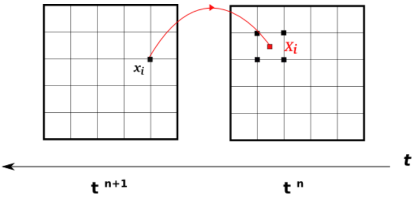
$$\frac{\partial f}{\partial t} + \mathbf{a}(x,t) \cdot \nabla_{\mathbf{x}} f = 0 \qquad (1)$$

The scheme:

- Fixed grid on phase-space
- Method of characteristics : ODE $\longrightarrow$ origin of characteristics
- Density f is conserved along the characteristics

$$\textit{i.e.} \qquad f^{n+1}(\mathbf{x}_i) = f^n(X(t_n; \mathbf{x}_i, t_{n+1})) \qquad (2)$$

- Interpolate on the origin using known values of previous step at mesh points (initial distribution f^0 known).



The guiding center model: general algorithm

We consider a reduced model of the gyrokinetic model – a simplified 2D Vlasov equation coupled with Poisson–:

$$\begin{cases} \dfrac{\partial f}{\partial t} + E_\perp \cdot \nabla_X f = 0 \\ -\Delta \phi = f \end{cases} \tag{3}$$

The global scheme:

- Known: initial distribution function f^0 and electric field E^0
- Solve (Leap frog, RK4, ...) ODE for origin of characteristics X
- For every time step :
  - Solve poisson equation $\Rightarrow E^{n+1}$
  - Interpolate distribution in $X^n \Rightarrow f^{n+1}$

Two different approaches for interpolation step:
Spline and Hermite Finite Elements interpolations.

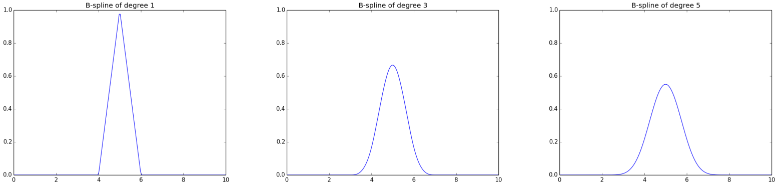
B(asis)-Splines basis*

B-Splines of degree d are defined by the **recursion** formula:

$$B_j^{d+1}(x) = \frac{x - x_j}{x_{j+d} - x_j} B_j^d(x) + \frac{x_{j+1} - x}{x_{j+d+1} - x_{j+1}} B_{j+1}^d(x) \tag{4}$$

Some important properties about B-splines:

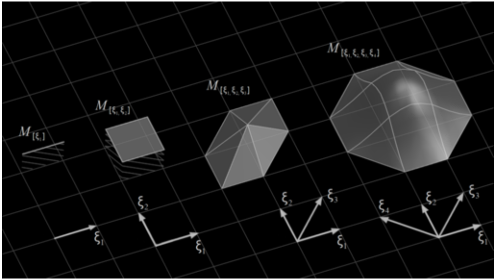
- Piecewise polynomials of degree d $\Rightarrow$ **smoothness**
- Compact support $\Rightarrow$ **sparse matrix system**
- Partition of unity $\sum_j Bj(x) = 1, \forall x$ $\Rightarrow$ **conservation laws**



Box-splines and quasi-interpolation

Box-Splines:

- Generalization of B-Splines
- Depend on the vectors that define the mesh
- Easy to exploit symmetry of the domain



A box-spline $B_\Xi : \mathbb{R}^d \to \mathbb{R}$ associated to the matrix $\Xi = [\eta_1, \eta_2, \ldots, \eta_N]$ is defined, when $N = d$ by

$$B_\Xi(x) = \frac{1}{|det\Xi|}\chi_\Xi(x)$$

else, by recursion

$$B_{\Xi\cup\eta}(x) = \int_0^1 B_\Xi(x - t\eta)$$

Box-splines and quasi-interpolation

Box-Spline's properties:

- Does not depend on the order of η_i in Ξ
- has the support $S = \Xi[0,1)^d$
- is positive on support S
- is symmetric

Quasi-interpolation:

- Distribution function known at mesh points
- Of order L if perfect reconstruction of a polynomial of degree $L-1$
- No exact interpolation at mesh points $f_h(x_i) = f(x_i) + O(\|\Delta x_i\|^L)$

$$f_h(x) = \sum_j c_j B_\Xi(x - x_j) \tag{5}$$

$\Rightarrow$ Additional freedom to choose the coefficients c_j

Computing the spline coefficients using pre-filters

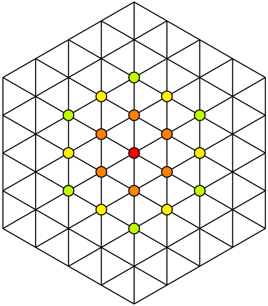
Idea: Coefficients obtained by discrete filtering of sample values $f(x_i)$

$$c = p * f = \sum_i f(x_i) p_i \qquad (6)$$

prefilters: Obtained by solving a linear system of L equations (quasi-interpolation conditions)

Example with $L = 2$:

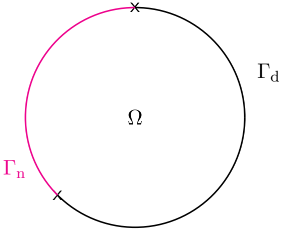
- We use information on two hexagons from point
- Points at same radius have same weight
- Error: $O(\| \Delta x \|^2)$



Poisson solver : FEM based solver

In cartesian coordinates:

$$\begin{cases} -\Delta_x \phi = f(t,x) & \text{in} \quad \Omega \\ \phi(t,x) = g_d(t,x) & \text{on} \quad \Gamma_\mathrm{d} \\ \nabla_x \phi(t,x) \cdot \mathbf{n} = g_n(t,x) & \text{on} \quad \Gamma_\mathrm{n} \end{cases}$$



Which we can write in general coordinates such as:

$$\nabla_\eta \cdot J^{-1}(J^{-1})^T \nabla_\eta \tilde{\phi}(\eta) = \tilde{f}(t,\eta) \tag{7}$$

And its weak formulation

$$-\int_\Omega (\nabla_\eta \tilde{\phi})^T \cdot J^{-1}(J^{-1})^T \nabla_\eta \psi \mid J(\eta) \mid \mathrm{d}\eta = \int_\Omega \tilde{f}(t,\eta)\psi \mid J(\eta) \mid \mathrm{d}\eta \tag{8}$$

with ψ test function, that we will define as a **box-spline**

Poisson solver : Discretization

We discretize the solution ϕ and the test function ψ using the splines (Box- or **B-splines**) denoted B_i as follows

$$\phi^h(\mathbf{x}) = \sum_i \phi_i B_i(\mathbf{x}), \qquad \rho^h(\mathbf{x}) = \sum_i \rho_i B_i(\mathbf{x})$$
$$\psi^h(\mathbf{x}) = B_j(\mathbf{x})$$

We obtain

$$\sum_{i,j} \phi_i \left(\int_\Omega \partial_x B_i \partial_y B_j + \int_\Omega \partial_y B_i \partial_y B_j \right) = -\sum_{i,k} \rho_i \int_\Omega B_i B_k \tag{9}$$

⇒ **SELALIB**'s general coordinate elliptic solver (developed by A. Back) or Jorek (**Django** version, developed by A. Ratnani) solver

Circular advection test case

In order to compare the two families' performances:

$$\partial_t f + y\partial_x f - x\partial_y f = 0 \tag{10}$$

Taking a gaussian pulse as an initial distribution function

$$f^n = \exp\left(-\frac{1}{2}\left(\frac{(x^n - x_c)^2}{\sigma_x^2} + \frac{(y^n - y_c)^2}{\sigma_y^2}\right)\right) \tag{11}$$

Constant CFL ($CFL = 2$) , $\sigma_x = \sigma_y = \frac{1}{2\sqrt{2}}$, hexagonal radius : 8.
Null Dirichlet boundary condition.

Hexagonal mesh: first results

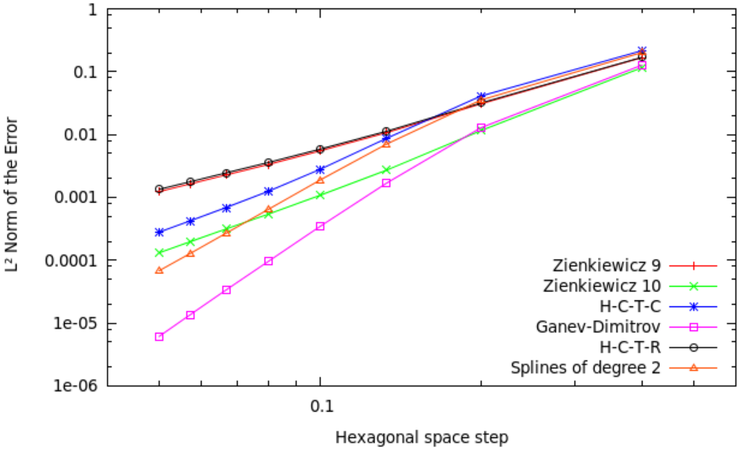
model	Points	a	dt	loops	L_2 error
On mesh points	17101	$0.$	0.025	1	4.99×10^{-6}
Constant advec.	17101	0.05	0.025	81	4.70×10^{-3}
Circular advec.	17101	$1.$	0.025	81	4.33×10^{-3}

Box-splines ($deg = 2$) for circular advection:

Cells	dt	loops	L_2 error	L_∞ error	points/μ-seconds
40	0.05	60	3.53E-2	7.74E-2	0.162
80	0.025	120	1.88E-3	4.66E-3	0.162
160	0.0125	240	6.77E-5	1.35E-4	0.162

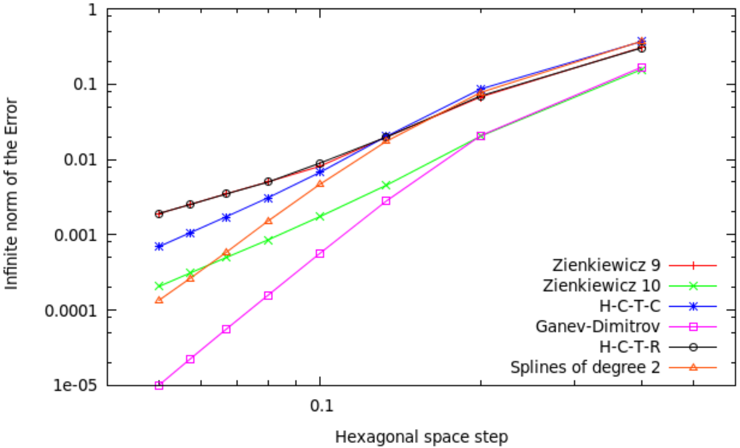
Comparing results with a FE method

As a project for the CEMRACS 2014, we decided to compare results with a FE scheme (joined work with Charles Prouveur, Michel Mehrenberger, Eric Sonnedrucker)



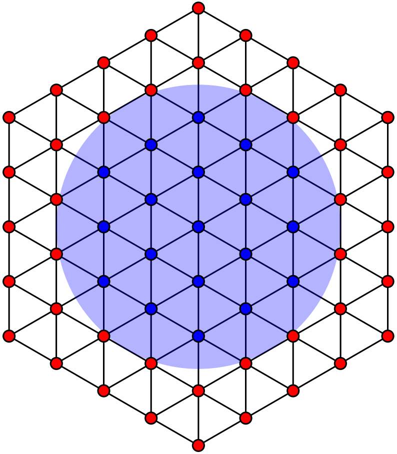
Comparing results with a FE method

As a project for the CEMRACS 2014, we decided to compare results with a FE scheme (joined work with Charles Prouveur, Michel Mehrenberger, Eric Sonnedrucker)



Handling boundary conditions : Main problem

Non interpolatory splines $\longrightarrow$ Problems with Dirichlet boundary conditions



We can differentiate three different types of elements:

- Interior elements
- Exterior elements
- Boundary elements

New questions arise:

- How to derive the equation such that BC intervene?
- Which elements should be considered as interior/exterior?

Dirichlet boundary conditions : Nitsche's method

Using Nitsche's method, we derive the variational form of the Poisson equation which yields[1]:

$$\int_\Omega \nabla\psi \cdot \nabla\phi \mathrm{d}\Omega - \int_{\Gamma d} \psi(\nabla\phi \cdot \mathbf{n})\mathrm{d}\Gamma_d - \int_{\Gamma d} \phi(\nabla\psi \cdot \mathbf{n})\mathrm{d}\Gamma_d + \alpha \int_{\Gamma d} \psi\phi \mathrm{d}\Gamma$$
$$= \int_\Omega \psi f \mathrm{d}\Omega + \int_{\Gamma n} \psi g_n \mathrm{d}\Gamma - \int_{\Gamma d} g_d(\nabla\psi \cdot \mathbf{n})\mathrm{d}\Gamma + \alpha \int_{\Gamma d} \psi g_d \mathrm{d}\Gamma$$

$\Rightarrow$ standard penalty method + additional integrals along Γ_d.

Solutions ϕ respect the boundary condition problem **under some conditions of the stabilization parameter** α

[1] Anand Embar, John Dolbow, and Isaac Harari. *International Journal for Numerical Methods in Engineering* 83.7 (2010), pp. 877–898. ISSN: 1097-0207.

Nitsche's method: coercivity study and the α parameter

We discretize the solution ϕ and the test function ψ using splines like before and we study $rhs(\psi^h, \phi^h)$ at (ψ^h, ψ^h):

$$rhs(\psi^h, \phi^h) = \int_\Omega \nabla\psi^h \cdot \nabla\psi^h \mathrm{d}\Omega - 2\int_{\Gamma d} \psi^h (\nabla\psi^h \cdot \mathbf{n}) \mathrm{d}\Gamma_d + \alpha \int_{\Gamma d} (\psi^h)^2 \mathrm{d}\Gamma$$

Using the definition of the L_2-norm : $\| \psi \| = \left(\int_\Omega \psi^2\right)^{1/2}$

$$rhs(\psi^h, \phi^h) = \| \nabla\psi^h \|^2 - 2\int_{\Gamma d} \psi^h (\nabla\psi^h \cdot \mathbf{n}) \mathrm{d}\Gamma_d + \alpha \| \psi^h \|^2$$

We define C such that $\| \nabla\psi^h \cdot \mathbf{n} \|^2_{\Gamma d} \leq C \| \nabla\psi^h \|^2$ and using Young's inequality we find that coercivity is ensured when

$$\boldsymbol{\alpha} > \frac{\mathbf{1}}{\mathbf{C}}$$

Conclusion and perspectives

Multi-patch:

- Schwartz iterative method: stabilize singular points
- May still be useful for more complex geometries
- Implementation in the **SELALIB** library

Hexagonal mesh:

- Results more encouraging than multi-patch results
- No numeric problems due to the mesh
- Efficiency to be compared
- More complex models to be tested
- Results have to be tested on a disk (and not a hexagon)
- Boundary conditions to be defined properly
- Box-MOMS (Maximal order minimal support box splines)