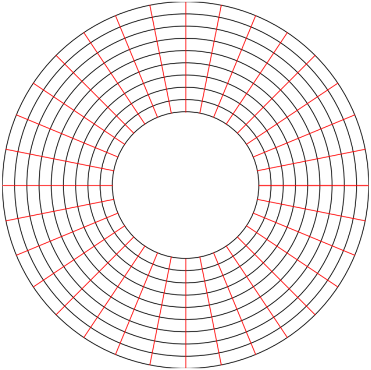
Solving the guiding center model using the Semi-Lagrangian scheme on a 2D hexagonal mesh (SelHex)

Michel Mehrenberger, Laura Mendoza, Charles Prouveur, Eric Sonnendrücker

Tuesday 9$^{\text{th}}$ September, 2014

Table of contents

Motivation



Current representation of the poloidal plane :

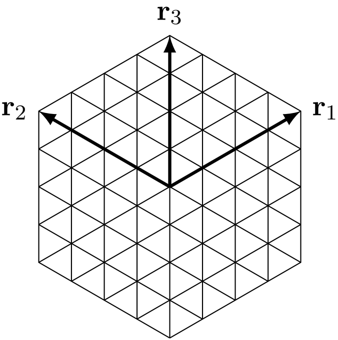
- Annular geometry
- **Polar mesh** (r, θ)

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**

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

The Backward Semi-Lagrangian Method

We consider the advection equation

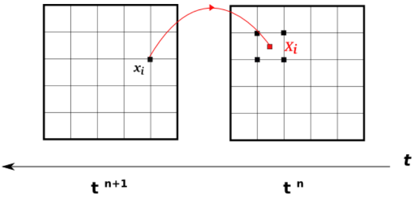
$$\frac{\partial f}{\partial t} + \mathbf{a}(x,t) \cdot \nabla_{\mathbf{x}} f = 0 \qquad (1)$$

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.

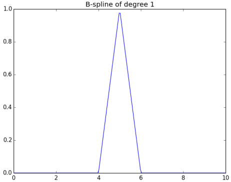
First approach: B(asis)-Splines basis*

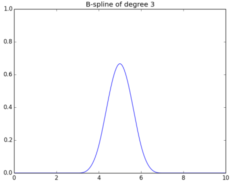
B-Splines of degree d are defined by the **recursion** formula:

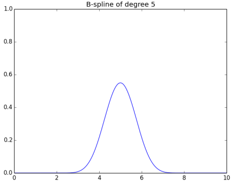
$$B_j^{d+1}(x) = \frac{x - x_j}{x_{j+d} - x_j} B_j^d(x) + \frac{x_{j+1} - x}{x_{j+d+1} - x_{j+1}} B_{j+1}^d(x) \tag{4}$$

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**







First approach: 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

$\Rightarrow$ More efficient interpolation

Quasi-interpolation:

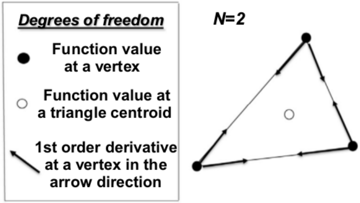
- Distribution function known at mesh points
- Of order L if perfect reconstruction of a polynomial of degree $L-1$
- No exact interpolation at mesh points $f_h(x_i) = f(x_i) + O(\|x_i\|^L)$

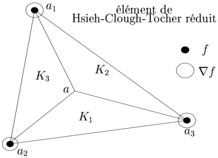
$\Rightarrow$ Additional freedom to choose the coefficients c_j

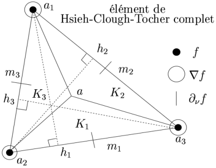
$$f_h(x) = \sum c_j \chi^L(x - x_j) \tag{5}$$

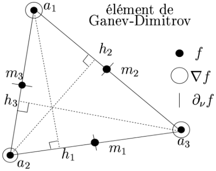
Second approach: Hermite finite elements

Computed from the value of the function and its derivatives in various directions at various points.
Five in total have been implemented and tested:









Circular advection test case

In order to compare the two families' performances:

$$\partial_t f + y\partial_x f - x\partial_y f = 0 \tag{6}$$

Taking a gaussian pulse as an initial distribution function

$$f^n = exp\left(-\frac{1}{2}\left(\frac{(x^n - x_c)^2}{\sigma_x^2} + \frac{(y^n - y_c)^2}{\sigma_y^2}\right)\right) \tag{7}$$

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

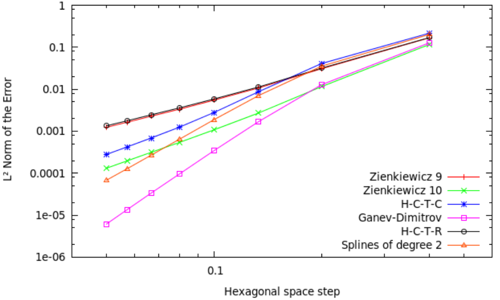
Box-splines ($deg = 2$) scheme:

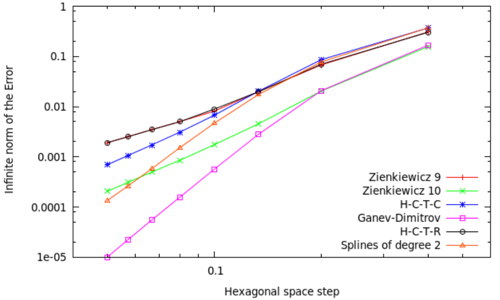
Points	**dt**	**loops**	L_2 **error**	L_∞ **error**	points/μ-seconds
40	0.5	60	3.53E-2	7.74E-2	0.105
80	0.025	120	1.88E-3	4.66E-3	0.105
160	0.0125	240	6.77E-5	1.35E-4	0.105

Hermite FE scheme:

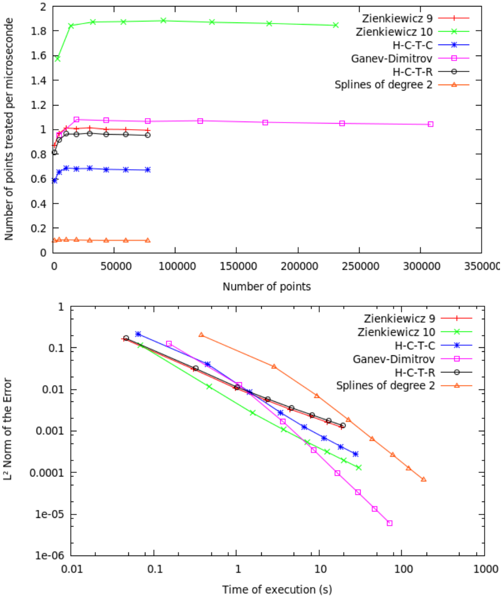
Points	**dt**	**loops**	**CFL**	L_2 **error**	L_∞ **error**	points/μ-seconc
40	dt	numloops	1.28E-2	error2	seconds	
80	dt	numloops	3.43E-4	error2	seconds	
160	dt	numloops	6E-6	error2	seconds	

Circular advection: results 1





Circular advection: results 2



Conclusion and perspectives

- **Hexagonal mesh for SELALIB**
- **Comparison between two families of finite elements.**

  Splines : good precision but poor efficiency.

  Amongst the other tested elements the complete Zienkiewicz's one seems to be the best choice in this case, at the moment.
- **Perspectives**

  Optimization.
  Abstract classes.
  Finite element solver of Poisson's equation on hexagonal mesh.