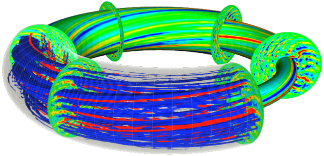
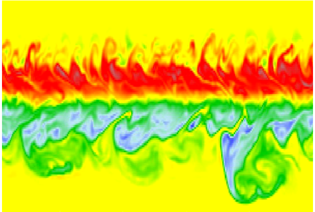
CEMRACS'10 project GYRONURBS

Access complex geometry in GYSELA with NURBS

GYSELA collaborators of **CEA-Cadarache**:
Jérémie Abiteboul, Simon Allfrey, Xavier Garbet, Philippe Ghendrih,
Virginie Grandgirard, Guillaume Latu,
Chantal Passeron, Yanick Sarazin, Antoine Strugarek

GYSELA collaborators of **INRIA CALVI & University of Strasbourg**:
Nicolas Besse, Jean-Philippe Braeunig, Nicolas Crouseilles,
Michel Mehrenberger, Ahmed Ratnani, Eric Sonnendrücker





Outline

1. Motivations
2. Introduction to NURBS
3. Using NURBS in GYSELA
4. Academic study: the paraxial beam using ISOLOSS
5. Conclusions and perspectives

Outline

1 Motivations

2 Introduction to NURBS

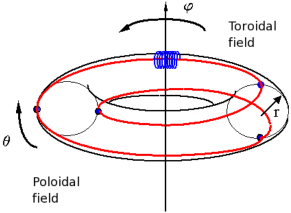
3 Using NURBS in GYSELA

4 Academic study: the paraxial beam using ISOLOSS

5 Conclusions and perspectives

Tokamak kinetic model:
6D phase space

- 3D in space:
  $\hookrightarrow$ toric geometry (r, θ, φ)
- 3D in velocity: $(v_\perp, \alpha, v_\parallel)$



Gyrokinetic theory:

- Adiabatic invariant $\mu = \frac{mv_\perp^2}{2B}$ replaces variables $(v_\perp, \alpha)$
  $\hookrightarrow$ gyrokinetic model 5D+t : Data $\bar{f}(r, \theta, \varphi, v_\parallel, \mu, t)$
- Vlasov equation to solve at each time step t

GYSELA 5D:

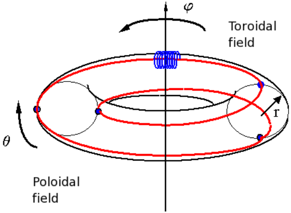
- Semi-Lagrangian method
- Strang splitting:
  - 1D advections in φ and $v_\parallel$
  - 2D advection in (r, θ)

GYSELA : a 5D gyrokinetic code

irfm
cea
cadarache

Tokamak kinetic model:
6D phase space

- 3D in space:
  $\hookrightarrow$ toric geometry (r, θ, φ)
- 3D in velocity: $(v_\perp, \alpha, v_\|)$



Gyrokinetic theory:

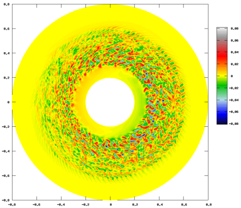
- Adiabatic invariant $\mu = \frac{mv_\perp^2}{2B}$ replaces variables $(v_\perp, \alpha)$
  $\hookrightarrow$ gyrokinetic model 5D+t : Data $\bar{f}(r, \theta, \varphi, v_\|, \mu, t)$
- Vlasov equation to solve at each time step t

GYSELA 5D:

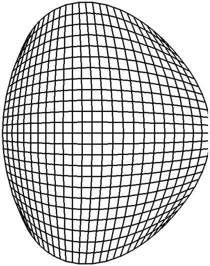
- Semi-Lagrangian method
- Strang splitting:
  - 1D advections in φ and $v_\|$
  - 2D advection in (r, θ)

irfm
cea
cadarache

- Up to now
  in the poloidal plane:
  - polar system (r, θ)
  - disc geometry
    (hole in the center)
    $r \in [r_{min}, r_{max}]$
    $r_{min} > 0$



- Aim: use NURBS in the poloidal plane
  - complex shapes accessible:
    ellipse, X-point, ...
  - avoid the hole in the center



Outline

1 Motivations

2 Introduction to NURBS

3 Using NURBS in GYSELA

4 Academic study: the paraxial beam using ISOLOSS

5 Conclusions and perspectives

Non-Uniform Rational B-Splines (NURBS)

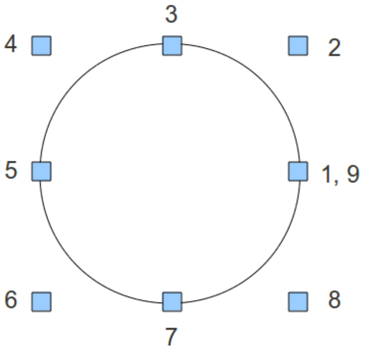
- Developed for applications in CAD
- Extension of B-splines (polynomials→rational functions)
- Provide an exact representation of geometrical shapes (including conic sections)
- A NURBS curve is defined by
  - its order p
  - a set of control points
  - a weight associated to each control point
  - a knot vector (→where the control points affect the curve)

Example of a 1D NURBS curve

- Order $p = 2$ (i.e. fractions of second degree polynomials)
- $N = 9$ control points P_i with weights ω_i

i	1	2	3	4	5	6	7	8	9
P_i	1;0	1;1	0;1	-1;1	-1;0	-1;-1	0;-1	1;-1	1;0
ω_i	1	$\frac{1}{\sqrt{2}}$	1	$\frac{1}{\sqrt{2}}$	1	$\frac{1}{\sqrt{2}}$	1	$\frac{1}{\sqrt{2}}$	1

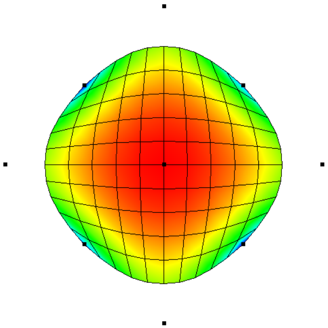
- Knot vector: $T = \left(0, 0, 0, \frac{1}{4}, \frac{1}{4}, \frac{1}{2}, \frac{1}{2}, \frac{3}{4}, \frac{3}{4}, 1, 1, 1\right)$

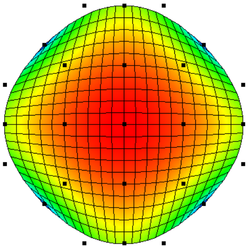


Tensor product of two NURBS curves

- $N^{(1)} = 3,\ p^{(1)} = 2$
- $N^{(2)} = 3,\ p^{(2)} = 2$
- 9 control points

- $N^{(1)} = 5,\ p^{(1)} = 2$
- $N^{(2)} = 5,\ p^{(2)} = 2$
- 25 control points





Outline

1 Motivations

2 Introduction to NURBS

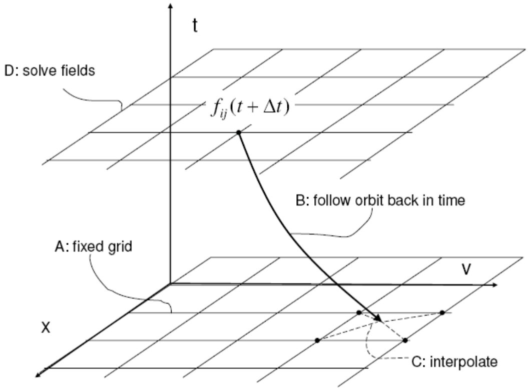
3 Using NURBS in GYSELA

4 Academic study: the paraxial beam using ISOLOSS

5 Conclusions and perspectives

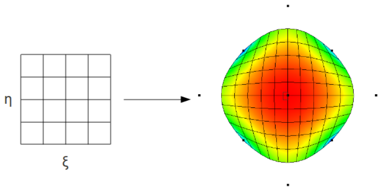
The Semi-Lagrangian method

- Fixed grid in phase space (∼Eulerian method)
- Follow characteristics back in time (∼PIC method)
- Interpolate foot of characteristic on the fixed grid



Using the ISOBOX library developed by A. Ratnani

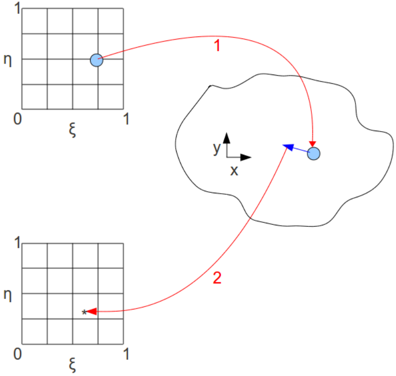
- Input
  - $N^{(1)}$, $p^{(1)}$, $N^{(2)}$, $p^{(2)}$
  - Knot vectors $T^{(1)}$, $T^{(2)}$
  - Control points
- Generates a 2D mesh parametrized by $(\xi, \eta) \in [0,1]^2$



- Direct mapping $(\xi, \eta) \to (x, y)$ is explicit (analytical expression)
- Inverse mapping $(x, y) \to (\xi, \eta)$ is not trivial

2 possible approaches

- Rewrite the advections in the (ξ, η) coordinates ➔ ISOPIC
- Compute the advections in real space (x, y)➔ GYRONURBS
  $\Rightarrow$ adds 2 steps to the usual SL method



- (1) Direct mapping
  $(\xi, \eta) \rightarrow (x, y)$
- Follow the characteristic backward
  $(x, y) \rightarrow (x^*, y^*)$
- (2) Inverse mapping
  $(x^*, y^*) \rightarrow (\xi^*, \eta^*)$
- Interpolate (ξ^*, η^*) on the fixed grid

Outline

1 Motivations

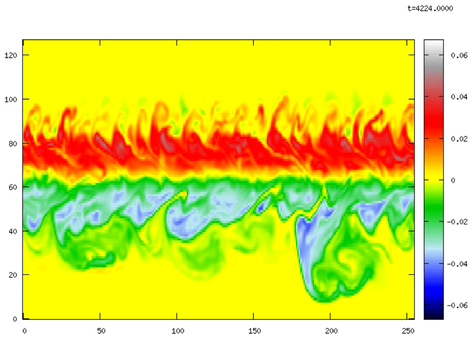
2 Introduction to NURBS

3 Using NURBS in GYSELA

4 Academic study: the paraxial beam using ISOLOSS

5 Conclusions and perspectives

- What happens in the poloidal plane during a GYSELA run?
- GYSELA 4D slab test case (cylinder geometry): one μ value
- Grid size is $N_r = 128, N_\theta = 256, N_\varphi = 32, N_{v_\parallel} = 64$
- Diagnostic shown (φ, $v_\parallel$ are fixed):
  slice $(f_t - f_0)$, abcissa θ, ordinate r



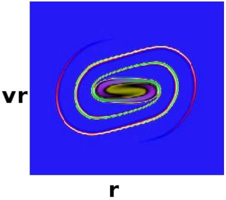
- Propagation of intense laser beam (accelerator physics)
- Vlasov-Poisson in cylindrical coordinates
- Constant propagation velocity along the optical axis
  $\Rightarrow$ distribution function in (r, v_r)
- System of equations

$$\partial_t f + v_r \partial_r f - E_{tot}(t,r)\partial_{v_r} f(t,r,v_r) = 0$$

$$\frac{1}{r}\partial_r(rE_{beam}(t,r)) = \rho(t,r) = 1 - \int_{-V_m}^{V_m} f(t,r,v)\,\mathrm{d}v$$

- Periodic focusing by an external electric field

$$E_{ext} = -\frac{1}{2}\left\{1+\cos\left(2\pi t\right)\right\} r$$



The ISOLOSS code

based on LOcal Spline Simulator code (Crouseilles-Latu 2006)

- 1D-1D Vlasov-Poisson solver
- Semi-Lagrangian method
- Local Cubic Spline interpolation
- Predictor-corrector scheme
- Strang splitting between spatial and velocity direction
- Describes Landau damping and two-stream instability

ISOLOSS *(this project)*:

- replace Strang splitting by 2D advection
  (closer to the situation in GYSELA)
- Implement the paraxial beam test case → film
- Create a non-cartesian grid using NURBS
- Adapt the Vlasov-Poisson solver to the NURBS formulation

irfm cea cadarache

Input : f^n on (ξ, η) grid, E^n
Output : f^{n+1} on (ξ, η) grid

for *each grid point* $(\xi, \eta) \in [0, 1]^2$ **do**
direct mapping $(\xi, \eta) \to (r, v_r)$;
Compute 2D advection $(r, v_r) \to (r^*, v_r^*)$;
Inverse mapping $(r^*, v_r^*) \to (\xi^*, \eta^*)$;
Interpolate (ξ^*, η^*) on the $[0, 1]^2$ grid (using local cubic splines);

Algorithm 1: Vlasov solver using NURBS grid

Inverse mapping

- In general, no analytical expression for $(r, v_r) \rightarrow (\xi, \eta)$
- surfaces defined by rational functions of 2 variables
  $\hookrightarrow$ high degree polynomials (>4 even for simple surfaces)
- Standard solution: iterative method (e.g. Newton scheme)
  $\Rightarrow$possibly high numerical cost
- Less costly solutions available for specific grids

- Polar grid: $(r, v_r) \rightarrow (\xi, \eta) \sim (\sqrt{r^2 + v_r^2}, \arctan(v_r/r))$

- for Bezier surfaces (specific category of NURBS): efficient inverse mapping using Bezier clipping + deCasteljau algorithm. *(work in progress)*
  $\Rightarrow$possible extension to the general case?

Poisson solver

- We need to compute $\rho(r) = \int f(r, v_r) \mathrm{d}v_r$ from $f(\xi, \eta)$ $\rightarrow$ a grid in the spatial coordinate r is required
- Naive approach: for each grid point in (ξ, η), "deposit" $f(\xi, \eta)$ at the closest radial grid point➔bad accuracy
- Method adopted: create a grid in velocity space as well

Input : f on (ξ, η) grid
Output : $E(r)$

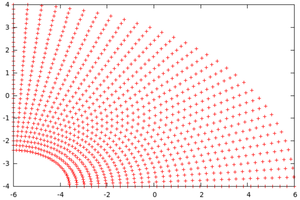
for *each grid point in r* **do**
  $\rho(r) = 0$;
  for *each grid point in v_r* **do**
    Inverse mapping $(r, v_r) \rightarrow (\xi, \eta)$;
    Interpolate on the $[0, 1]^2$ grid to find $f(\xi, \eta)$;
    $\rho(r) = \rho(r) + f(\xi, \eta)$

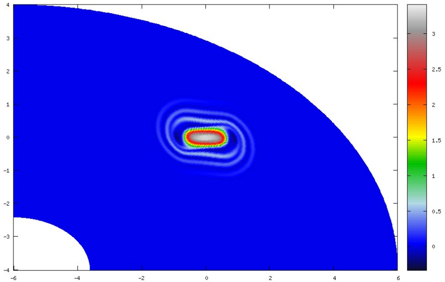
Compute E on radial grid points $(\partial_r(rE)/r = \rho)$

Algorithm 2: Poisson solver using NURBS grid

Results

- Quarter-ring grid defined by a polar parameterization
- $\hookrightarrow$ Simple analytical inverse mapping





- $\Rightarrow$ Filamentation in phase-space is well-described (➔ film)
- No extra numerical cost

We also considered a "tokamak-like" surface

- Mesh generated using Bezier curves
- *in progress:* inverse mapping with Bezier clipping algorithm

Outline

1 Motivations

2 Introduction to NURBS

3 Using NURBS in GYSELA

4 Academic study: the paraxial beam using ISOLOSS

5 Conclusions and perspectives

Conclusions and perspectives

Results on the 2D test case

- Generation of a NURBS grid using the ISOBOX library
- Vlasov-Poisson solver for this non-cartesian grid
- $\Rightarrow$ Correct description of filamentation in phase-space
- in progress: inverse mapping for non-analytical case

Perspectives for GYSELA 5D

- The ISOBOX interface has been modified and is now adaptable to the GYSELA code
- Next step: solve the 2D advection in GYSELA using NURBS
- Further perspective: solve the quasi-neutrality equation in GYSELA using NURBS