A conservative semi-Lagrangian method for the gyrokinetic Vlasov equation

Eric Sonnendrücker

IRMA, Université de Strasbourg & CNRS, France
Calvi project-team INRIA Nancy Grand Est

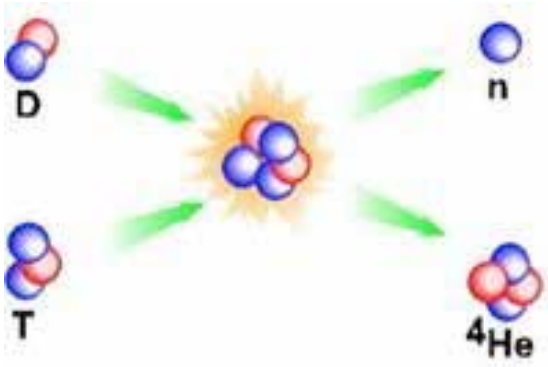
Collaborators:
J.P. Braeunig, N. Crouseilles, M. Mehrenberger, A. Ratnani (CALVI)
V. Grandgirard, G. Latu (CEA Cadarache)

ICERM, Providence, September 19, 2011

Outline

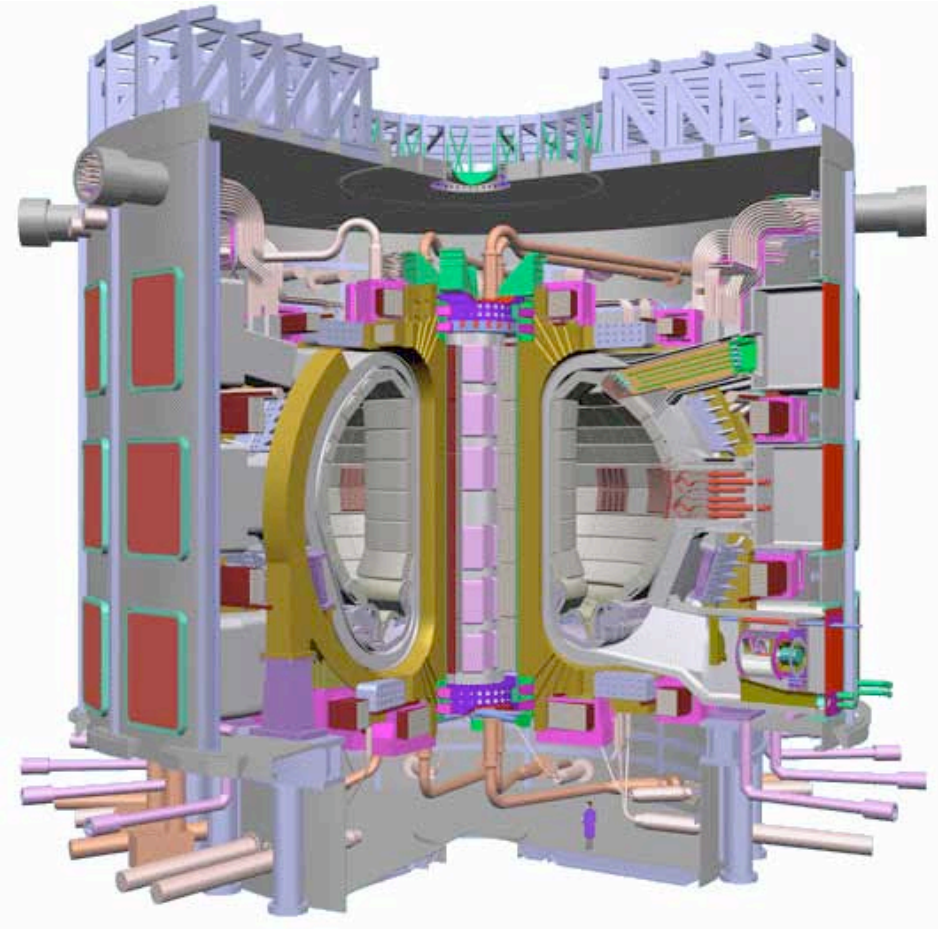
Outline

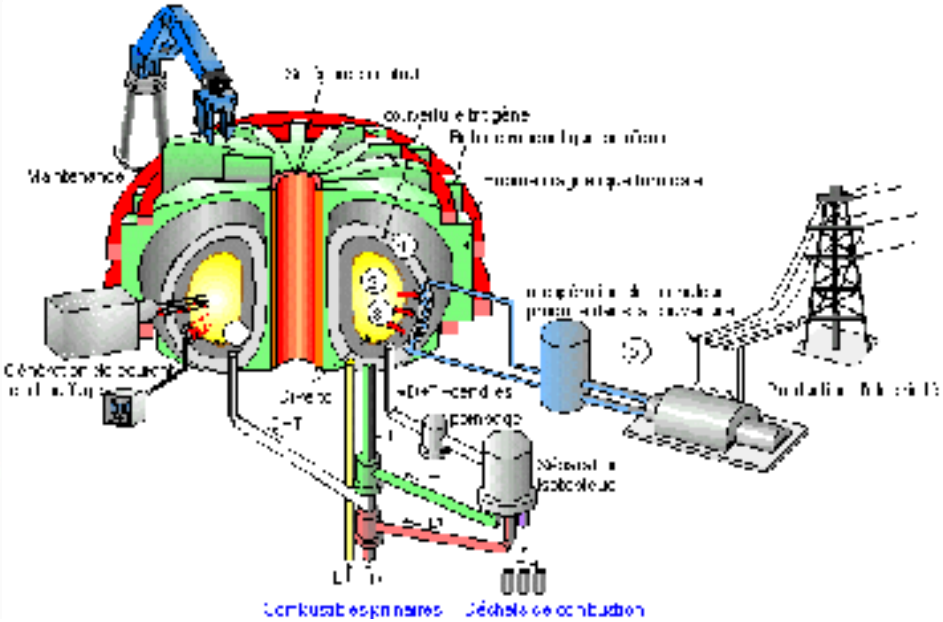
The Physics context : Controlled thermonuclear fusion



- Magnetic confinement (ITER)
- Inertial confinement
  - by laser (LMJ, NIF)
  - by heavy ions

The ITER project





ITER numerical simulations in France

- ITER being built in France: Several new research actions.
- INRIA Large scale initiative on magnetic fusion. Involve applied mathematicians and computer scientists in ITER.

- Mathematical understanding and development of reduced models
- Gyrokinetic simulations: GYSELA (CEA+Strasbourg)
- MHD simulations: JOREK (CEA+Bordeaux+Nice)
- Inovative numerical methods: Asymptotic Preserving, geometric..
- Plasma control: EQUINOXE (CEA+Nice)

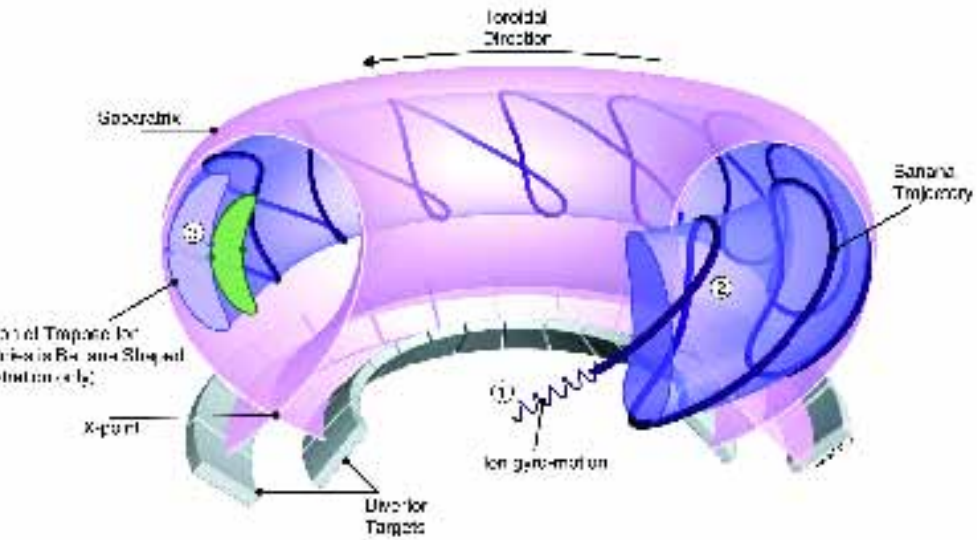
Outline

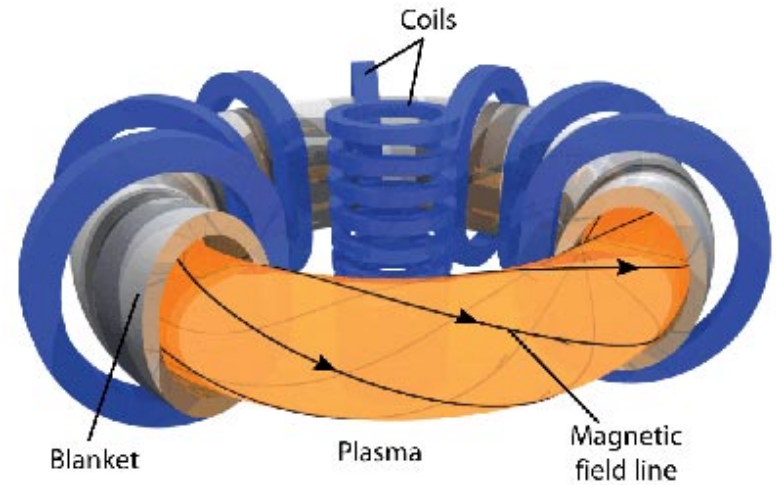
Turbulent transport in magnetized plasma

- Plasma not very collisional and far from fluid state
  $\Rightarrow$ Kinetic description necessary. Fluid and kinetic simulations of turbulent transport yield very different results.
- Vlasov (6D espace des phases) coupled to Maxwell or Poisson

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_x f + \frac{q}{m}(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_v f = 0.$$

- Toroidal geometry





The drift-kinetic model (in slab geometry)

- Assume a large magnetic field and small fluctuations of electric field on the scale of a Larmor radius. We then get the following dimensionless Vlasov equation

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_x f + (\mathbf{E} + \frac{1}{\varepsilon}\mathbf{v} \times \mathbf{B}) \cdot \nabla_v f = 0.$$

- taking the limit $\epsilon \to 0$ we get a 4D model $(r, \theta, \phi, v_\parallel)$

$$\frac{\partial f}{\partial t} + v_D \cdot \nabla_x f + v_\parallel \cdot \nabla_\parallel f + \frac{q}{m}\mathbf{E}_\parallel \cdot \nabla_v f = 0,$$

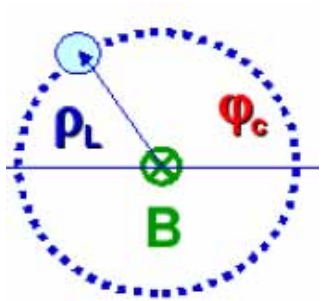
with $v_D = \frac{\mathbf{E}\times\mathbf{B}}{B^2}$. Poisson's equation is replaced by the quasi-neutrality relation

$$-\nabla_\perp \cdot (\frac{n_0(r)}{B\omega_c}\nabla_\perp \phi) + \frac{e\, n_0(r)}{T_e(r)}(\phi - \lambda\langle\phi\rangle) = n - n_0,$$

$\langle\phi\rangle$ representing the average of the potential over the magnetic flux lines.

The gyrokinetic model (in slab geometry) (1/2)

- Assume a large magnetic field and fluctuations of electric field on the scale of a Larmor radius. We then need to take into account finite Larmor radius effects. We then obtain the following dimensionless Vlasov equation



$$\frac{\partial f}{\partial t} + \mathbf{v}_{\parallel} \cdot \nabla_x f + \frac{1}{\varepsilon} \mathbf{v}_{\perp} \cdot \nabla_x f + (\mathbf{E} + \frac{1}{\varepsilon} \mathbf{v} \times \mathbf{B}) \cdot \nabla_v f = 0.$$

  - This model will consider several values of the magnitude of the transverse velocity which is an adiabatic invariant. It will be represented by the parameter $\mu = \frac{mv_{\perp}^2}{2B}$ which does not appear in the Vlasov equation, but needs to be takent into account when computing the charge density.

The gyrokinetic model (2/2)

- We obtain a 5D model describing the evolution of the guiding center distribution $f(r, \theta, \phi, v_\parallel, \mu)$

$$\frac{\partial f}{\partial t} + v_D \cdot \nabla_x f + v_\parallel \cdot \nabla_\parallel f + \frac{q}{m} \mathbf{E}_\parallel \cdot \nabla_v f = 0,$$

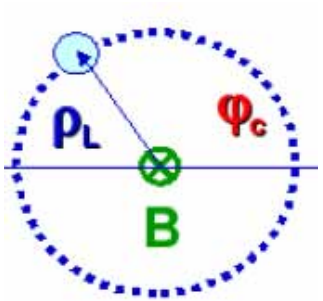
with $v_D = -\frac{\nabla J(\phi) \times \mathbf{B}}{B^2}$, coupled with the quasi-neutrality equation

$$-\nabla_\perp \cdot (\frac{n_0(r)}{B\omega_c} \nabla_\perp \phi) + \frac{e\, n_0(r)}{T_e(r)} (\phi - \lambda \langle \phi \rangle) = \int J(f)\, dv_\parallel d\mu - n_0.$$

- The gyroaverage operator J transforms the guiding-center distribution onto the actual particle distribution enabling to take into account the finite Larmor radius.

The gyroaverage operator

The gyroaverage operator is applied to a function g depending on the guiding center distribution. For a particle of velocity v, the Larmor radius is
$\vec{\rho} = \frac{\vec{v}_\perp}{\omega_c} = (\rho\cos\alpha, \rho\sin\alpha)$ where $\vec{v}_\perp = (-v_y, v_x)$



$$J(g)(\vec{x},\vec{v}) = \frac{1}{2\pi}\int_0^{2\pi} g(x+\rho\cos\alpha, y+\rho\sin\alpha)\,d\alpha.$$

A Fourier transform in $\vec{x}$ yields

$$\widehat{J(g)}(\vec{k},\vec{v}) = \frac{1}{2\pi}\int_0^{2\pi} e^{i\vec{k}\cdot\vec{\rho}}\,d\alpha\;\hat{g}(\vec{k}),$$

moreover denoting by $\vec{k}_\perp = (k_\perp\cos\beta, k_\perp\sin\beta)$,

$$\frac{1}{2\pi}\int_0^{2\pi} e^{i\vec{k}\cdot\vec{\rho}}\,d\alpha = \frac{1}{2\pi}\int_0^{2\pi} e^{i\rho k_\perp\cos(\alpha-\beta)}\,d\alpha = J_0(\rho k_\perp)$$

Justification of the limits

- The gyrokinetic models used in codes mostly obtained by Lie transform methods. Littlejohn, Hahm, Brizard, Qin
- Another approach (Sosenko) is based on the Boboliugov-Mitropolsky averaging technique.
- Rigorous mathematical justification not complete. Some recent results obtained by:
  - Bostan (2010): gyrokinetic Fokker-Planck
  - Bostan (2007,2009)
  - Han-Kwan (2010): 3D Finite Larmor Radius
  - Ghendrih, Hauray, Nouri (2009): Existence and uniqueness of stationary solution
  - Frénod (2006)
  - Golse, Saint-Raymond (1999, 2003)
  - Saint-Raymond (2000, 2002)
  - Frénod, ES (2001): 2D Finite Larmor Radius
  - Brenier (2000)

The gyrokinetic model (in torus)

- The Vlasov-Poisson gyrokinetic model reads

$$\frac{\partial f}{\partial t} + \frac{d\mathbf{X}}{dt} \cdot \nabla_x f + \frac{dV_\parallel}{dt} \frac{\partial f}{\partial v_\parallel} = 0,$$

with

$$\begin{aligned} B^* \frac{d\mathbf{X}}{dt} &= \mathbf{b} \times \nabla J(\phi) + \frac{1}{q}(m V_\parallel^2 \nabla \times \mathbf{b} + \mu \mathbf{b} \times \nabla B) + V_\parallel \mathbf{B} \\ B^* \frac{dV_\parallel}{dt} &= -(\mathbf{B} + \frac{m}{q} V_\parallel \nabla \times \mathbf{b}) \cdot (\frac{\mu}{m} \nabla B + \frac{q}{m} \nabla J(\phi)) \end{aligned}$$

and $B^* = B + \frac{m}{q} V_\parallel \nabla \times \mathbf{b} \cdot \mathbf{b}$.

- The gyroaverage operator J transforms the guiding center distribution to the distribution at the particle position which allows to take into account the finite Larmor radius effects.

Conservativity

- We have the relations

$$\nabla \cdot (B^* \frac{d\mathbf{X}}{dt}) = \nabla \cdot (\mathbf{b} \times \nabla J(\phi)) + \frac{1}{q} \nabla \cdot (\mathbf{b} \times \mu \nabla B)$$
$$= \nabla J(\phi) \cdot \nabla \times \mathbf{b} + \frac{\mu}{q} \nabla B \cdot \nabla \times \mathbf{b}$$

On the other hand

$$\frac{\partial}{\partial v_\parallel} (B^* \frac{dV_\parallel}{dt}) = -\nabla \times \mathbf{b} \cdot (\frac{\mu}{q} \nabla B + \nabla J(\phi)).$$

Hence the phase-space divergence vanishes, which leads to the conservatity.

Outline

Numerical solution of the gyrokinetic model

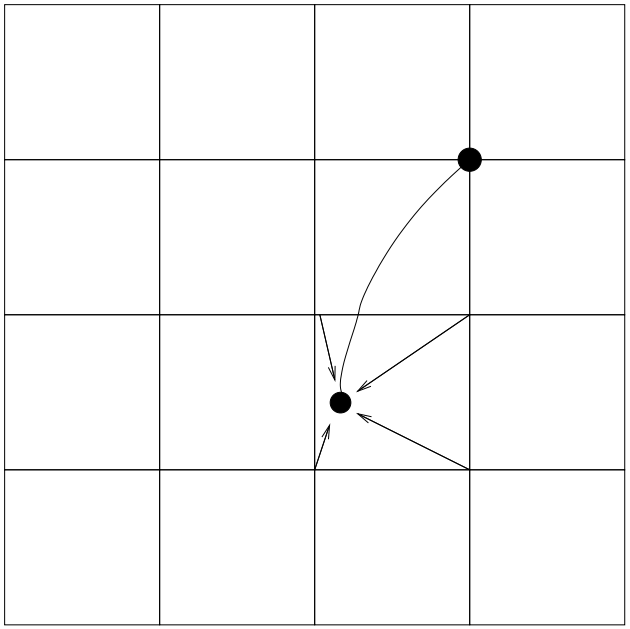
Difficulties: defined in phase-space (5D). Appearance of small scales

- Two approaches
  - Full f. The gyrokinetic equation is solved for the full distribution function.
  - δf. The perturbation with respect to the equilibrium function is solved, involving in some cases linearized equations.
- Two types of numerical methods:
  - Particle methods: often preferred for higher dimensions.
    - Often good qualitative results at lower cost.
    - Numerical noise and slow convergence $\rightarrow$ difficult to get precise results in some situations.
  - Methods based on phase-space grids:
    - Good grid resolution necessary for qualitatively correct results.
    - No numerical noise, but diffusion.

Massive parallelism

- New fusion dedicated supercomputers (Jülich, Germany, since last september - Japan 2012
- More that 10000 processors will be available for gyrokinetic runs.
- New constraints on numerical methods (with respect to < 100 processors) :
  - No more transfer from one to all.
  - No global data redistribution
  - Sophisticated adaptive methods less competitive
  - Load balance more complex, in particular for particle methods.
  - Advantage to local methods with static load balancing.

The classical backward semi-Lagrangian Method



- f conserved along characteristics
- Find the origin of the characteristics ending at the grid points
- Interpolate old value at origin of characteristics from known grid values $\rightarrow$ High order interpolation needed

- Typical interpolation schemes.
  - Cubic spline (Cheng-Knorr)
  - Cubic Hermite with derivative transport (Nakamura-Yabe)

A history of semi-Lagrangian schemes for the Vlasov equation

- Cheng-Knorr (JCP 1976): split method for 1D Vlasov-Poisson
- ES-Roche-Bertrand-Ghizzo (JCP 1998) : general semi-Lagrangian framework for Vlasov type equations.
- Nakamura-Yabe (CPC 1999): CIP semi-Lagrangian scheme with Hermite interpolation.
- Filbet-ES-Bertrand (JCP 2001): PFC semi-Lagrangian positive and conservative with Lagrange interpolation.
- Nakamura-Tanaka-Yabe-Takizawa (JCP 2001): Conservative 1D split CIP semi-Lagrangian.
- N. Besse - ES (JCP 2003) : SL schemes on unstructured grids.
- Crouseilles-Respaud-ES (CPC 2009) : Forward SL method.
- Crouseilles-Mehrenberger-ES (JCP 2010): conservative semi-Lagrangian spline PSM + new classes of positive filters.
- Qiu-Christlieb (JCP 2010): conservative SL WENO schemes.
- Qiu-Shu (U. Brown preprint 2010): DG SL schemes.

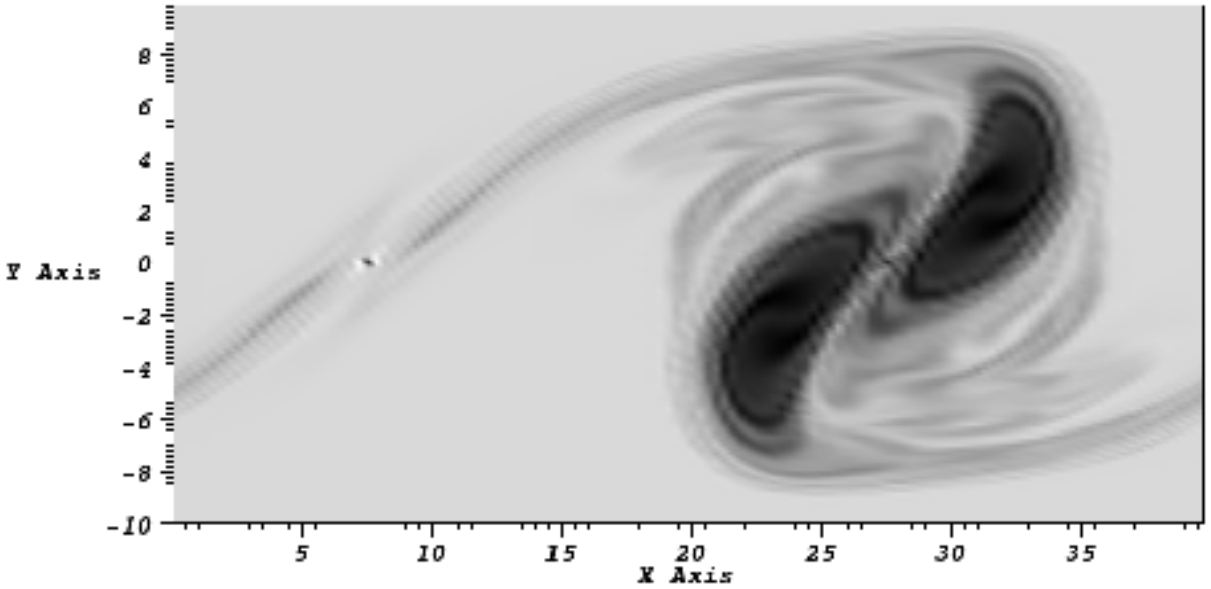
Convergence of semi-Lagrangian schemes

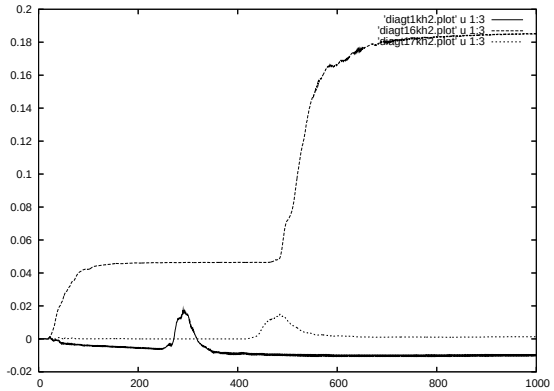
- Filbet (SINUM 2001): PFC method for Vlasov-Poisson
- N. Besse (SINUM 2003): semi-Lagrangian method with linear interpolation for Vlasov-Poisson.
- Campos Pinto - Mehrenberger (Numer. Math. 2008): adaptative SL method for Vlasov-Poisson
- N. Besse - Mehrenberger (Math of Comp 2008): split SL method for Vlasov-Poisson for symmetric high order interpolation Lagrange and spline interpolation.
- N. Besse (SINUM 2008): SL method with Hermite interpolation and propagation of gradients.
- Respaud - ES (Numer math, 2011): Forward SL for Vlasov-Poisson with linear interpolation.

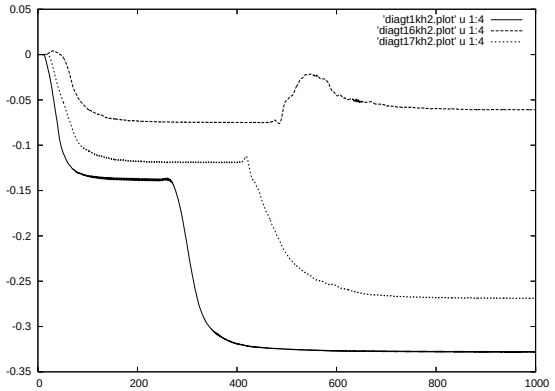
Problem with non conservative Vlasov solver

- When non conservative splitting is used for the numerical solver, the solver is not exactly conservative.
- Does generally not matter when solution is smooth and well resolved by the grid. The solver is still second order and yields good results.
- However: Fine structures develop in non linear simulations and are at some point locally not well resolved by the phase space grid.
- In this case a non conservative solvers can exhibit a large numerical gain or loss of particles which is totally unphysical.
- Lack of robustness.
- Note that by an adequate computation of the electric field from the distribution function, non split classical SL method based on cubic B-splines can be made conservative at lowest order (on displacements).
  very good but not exact conservation observed in actual simulations.

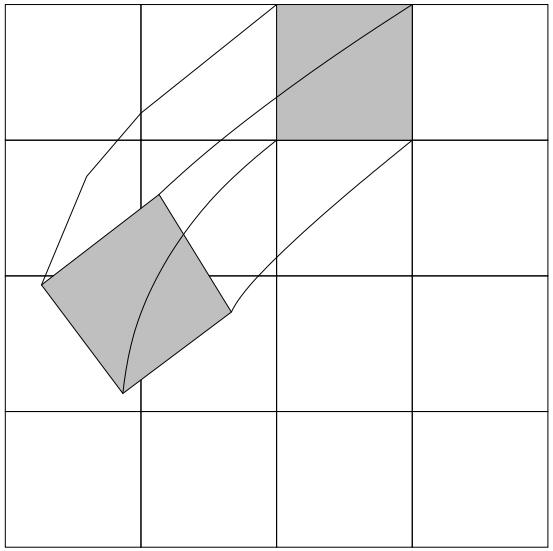
Vortex in Kelvin-Helmholtz instability







Conservative semi-Lagrangian method



- Start from conservative form of Vlasov equation

$$\frac{\partial f}{\partial t} + \nabla \cdot (f\mathbf{a}) = 0.$$

- $\int_V f\, dx\, dv$ conserved along characteristics
- Three steps:
  - High order polynomial reconstruction.
  - Compute origin of cells
  - Project (integrate) on transported cell.

- Efficient with splitting in 1D conservative equations as cells are then defined by their 2 endpoints. A lot more complex for 2D (or more) transport.
- Splitting on conservative form: always conservative.
- Filtering for positivity and damping of oscillations due to high-order interpolation.

Choice of interpolation

- What interpolation should be chosen for primitive?
- Lagrange interpolation with centered stencil (used in PFC: Filbet, ES, Bertrand JCP 2001).
- ENO type interpolation. Lagrange with varying stencil. Not efficient for Vlasov.
- WENO. Recent positive results (Carrillo-Vecil, Qiu-Christlieb).
- Cubic spline interpolation: cubic polynomial on each cell, globally C^2 $\rightarrow$ reconstructed function is then locally a quadratic polynomial and globally C^1. Linked to cubic spline interpolation for classical semi-Lagrangian method.
- Related but different method : SL DG uses a Galerkin formulation and high order polynomials on each cell (Qiu-Shu). Promising recent results.

Origin of cells and projection

- Compute cell origins:
  - In 1D cell and its origin determined by end points. Compute origin of end points like in classical semi-Lagrangian method.
  - Need to make sure end points do not cross $\rightarrow$ restriction on time step.
- Compute average value of f^{n+1} on cells using

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} f^{n+1}(x)\,dx = \int_{X(t_n;x_{i-\frac{1}{2}},t_{n+1})}^{X(t_n;x_{i+\frac{1}{2}},t_{n+1})} f^{n}(x)\,dx,$$

where $f^n(x)$ is the high order polynomial reconstruction.

Link between classical and conservative semi-Lagrangian methods

- For constant coefficient advections we can show that

$$\text{C-Lag(2d)} \Longleftrightarrow \text{SL-Lag(2d+1)}$$
$$\text{PSM} \Longleftrightarrow \text{SPL}$$

- Consequences:
  1. Classical semi-Lagrangian method is conservative for split schemes reducing to constant coefficients advection for each split step.
  2. PFC (Filbet-ES-Bertrand) corresponds to classical semi-Lagrangian method with cubic Lagrange interpolation.
  3. PSM corresponds to classical semi-Lagrangian method with cubic spline interpolation.
- Note that Collella and Woodward Finite Volume scheme is also algebraically equivalent for constant coefficient advections.

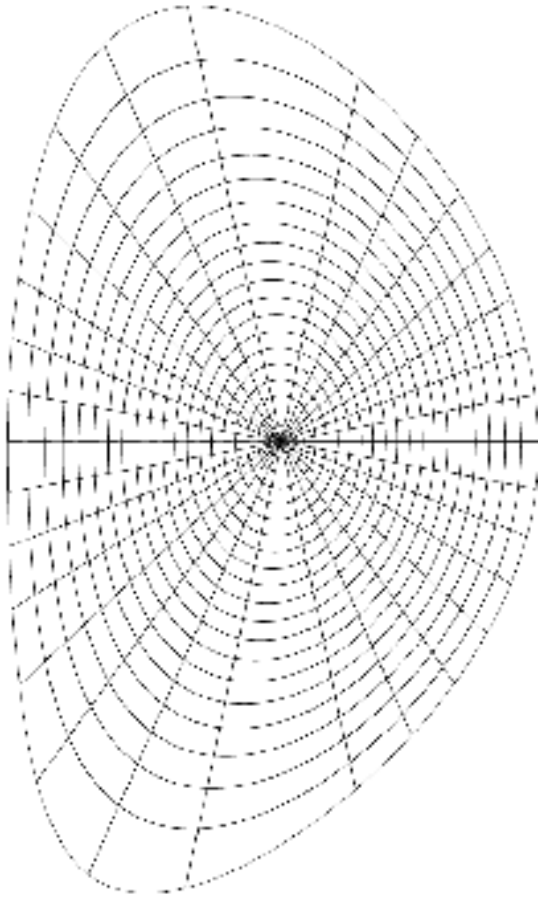
Outline

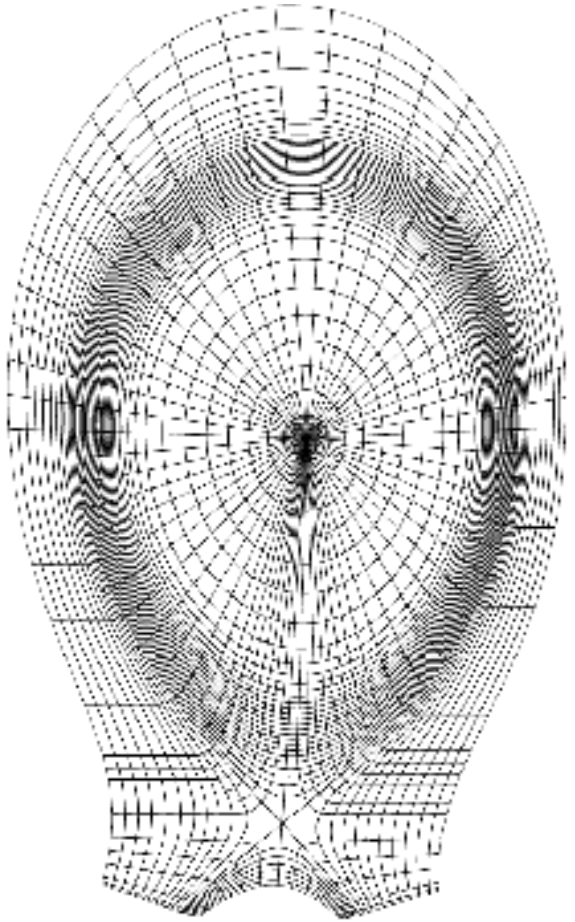
Motivation for curvilinear mesh for gyrokinetic simulations

- Due to confinement by large magnetic field particle travel a lot faster along magnetic field lines than across.
- Look at typical turbulence simulation shows that large scale structures form along field lines and small scale structures across.
- Significant anisotropy of flow
- For optimal resolution anisotropic mesh needed.
- To keep numerical efficiency of structured meshes: use mapped mesh which defines curvilinear coordinates almost aligned with magnetic field lines.

Structured mesh using Bézier splines (JOREK, G. Huysmans)

Mesh aligned on magnetic flux surfaces, without and with X-point.





Splines and NURBS

Splines are smooth piecewise polynomial functions.
Let $T = (t_i)_{1 \leqslant i \leqslant N+k}$ be a non-decreasing sequence of knots.

- The i-th B-Spline of order k is defined by the recurrence relation:

$$N_j^k = w_j^k N_j^{k-1} + (1 - w_{j+1}^k) N_{j+1}^{k-1}$$

$$\text{where} \qquad w_j^k(x) = \frac{x - t_j}{t_{j+k-1} - t_j}, \qquad N_j^1(x) = \chi_{[t_j, t_{j+1}[}(x)$$

- The i-th NURBS of order k associated to the knot vector T and the weights ω, is defined by

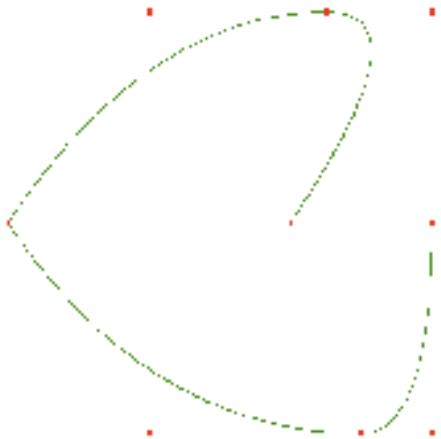
$$R_i^k = \frac{\omega_i N_i^k}{\sum_{j=1}^N \omega_j N_j^k}.$$

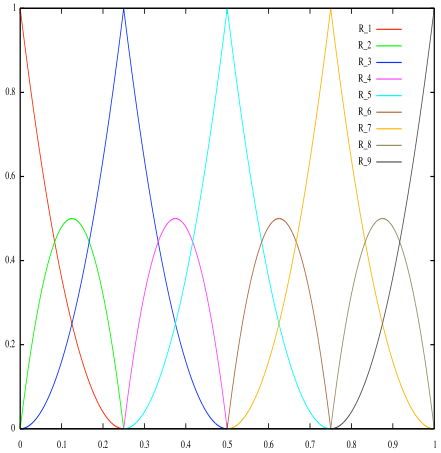
Spline curves

- B-splines (and NURBS) can be used to defined curves and surfaces
- The B-spline curve in $\mathbb{R}^d$ associated to $T = (t_i)_{1 \leqslant i \leqslant N+k}$ and $(\mathbf{P}_i)_{1 \leqslant i \leqslant N} \in \mathbb{R}^d$ is defined by :

$$\mathbf{M}(t) = \sum_{i=1}^{N} N_i^k(t) \mathbf{P}_i$$

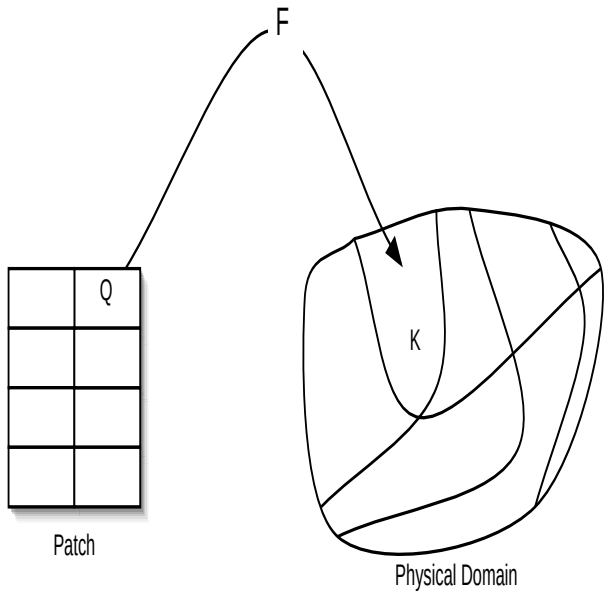
- Example: (left) A B-spline curve and its control points, (right) B-splines functions used to draw the curve. $N = 9$, $p = 2$, $T = \{000, \frac{1}{4}\frac{1}{4}, \frac{1}{2}\frac{1}{2}, \frac{3}{4}\frac{3}{4}, 111\}$





Mapped mesh

- A mapped mesh is defined by a mapping e.g. $F : \mathbb{R}^2 \to \mathbb{R}^2$ from a structured uniform logical mesh parametrized by (ξ_1, ξ_2) to the physical domain in cartesian coordinates $x_1 = F_1(\xi_1, \xi_2), \quad x_2 = F_2(\xi_1, \xi_2)$.
- Mapping can be analytical coordinate transformation or defined by spline or NURBS curves.



- We then have the contravariant (curvilinear) coordinates A^i: $A^i = A \cdot \nabla_x \xi_i$, with A velocity vector in cartesian coordinates.
- Conservative Vlasov equation in curvilinear coordinates then writes:

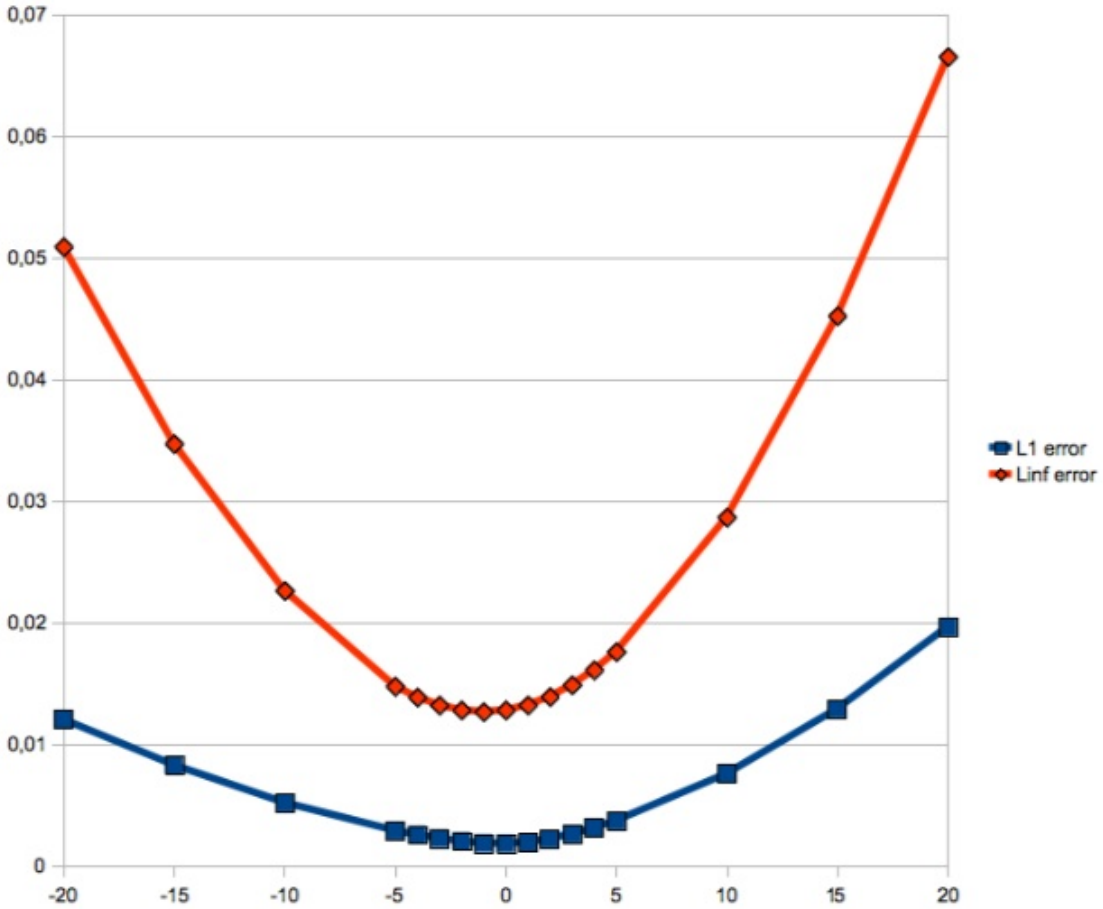
$$\sqrt{g}\frac{\partial f}{\partial t} + \sum_k \frac{\partial}{\partial \xi_k}\left(\sqrt{g}\ f\ A^k\right) = 0,$$

with the transformation matrix jacobian $\sqrt{g} = det\left(\frac{D(x_1, x_2)}{D(\xi_1, \xi_2)}\right)$.

Approximate alignment

L^1 and L^∞ norms of relative error for oblic advection (travel 10 times the domain) function of the angle between advection direction and mesh lines

direction:



Guiding-Center model

It is a reduced 2D model:

$$\frac{\partial \rho}{\partial t} + E^{\perp} \cdot \nabla_x \rho = 0$$

$$-\Delta \Phi = \nabla \cdot E = \rho.$$

with $E = -\nabla \Phi = (E_x, E_y)$ and $E^{\perp} = (-E_y, E_x)$.

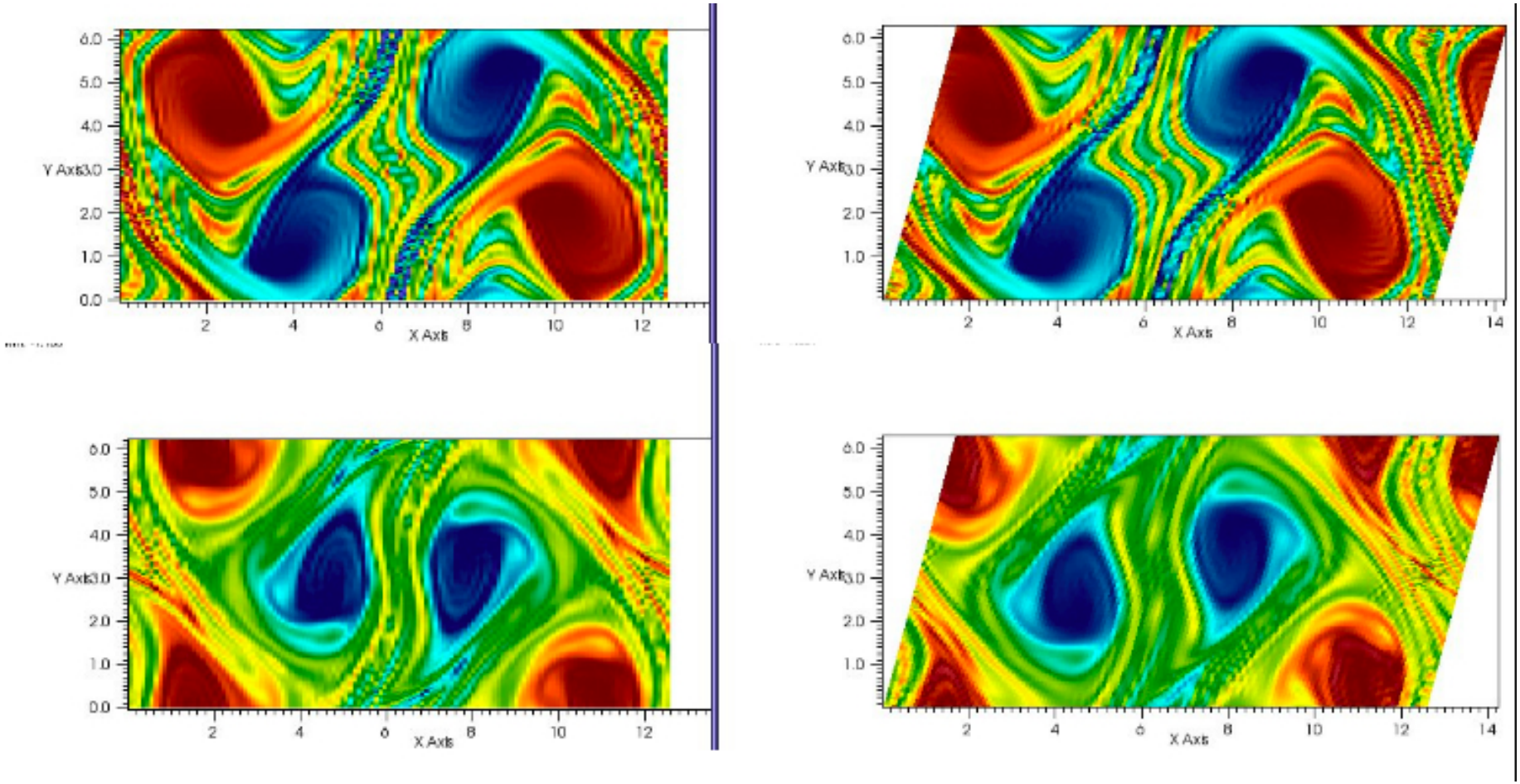
Since $\nabla \cdot E^{\perp} = -\partial_x \partial_y \Phi + \partial_y \partial_x \Phi = 0$,
one can write the Vlasov equation in a conservative formulation:

$$\frac{\partial \rho}{\partial t} + \nabla_X \cdot (\rho E^{\perp}) = 0.$$

Therefore a conservative directional splitting is possible.

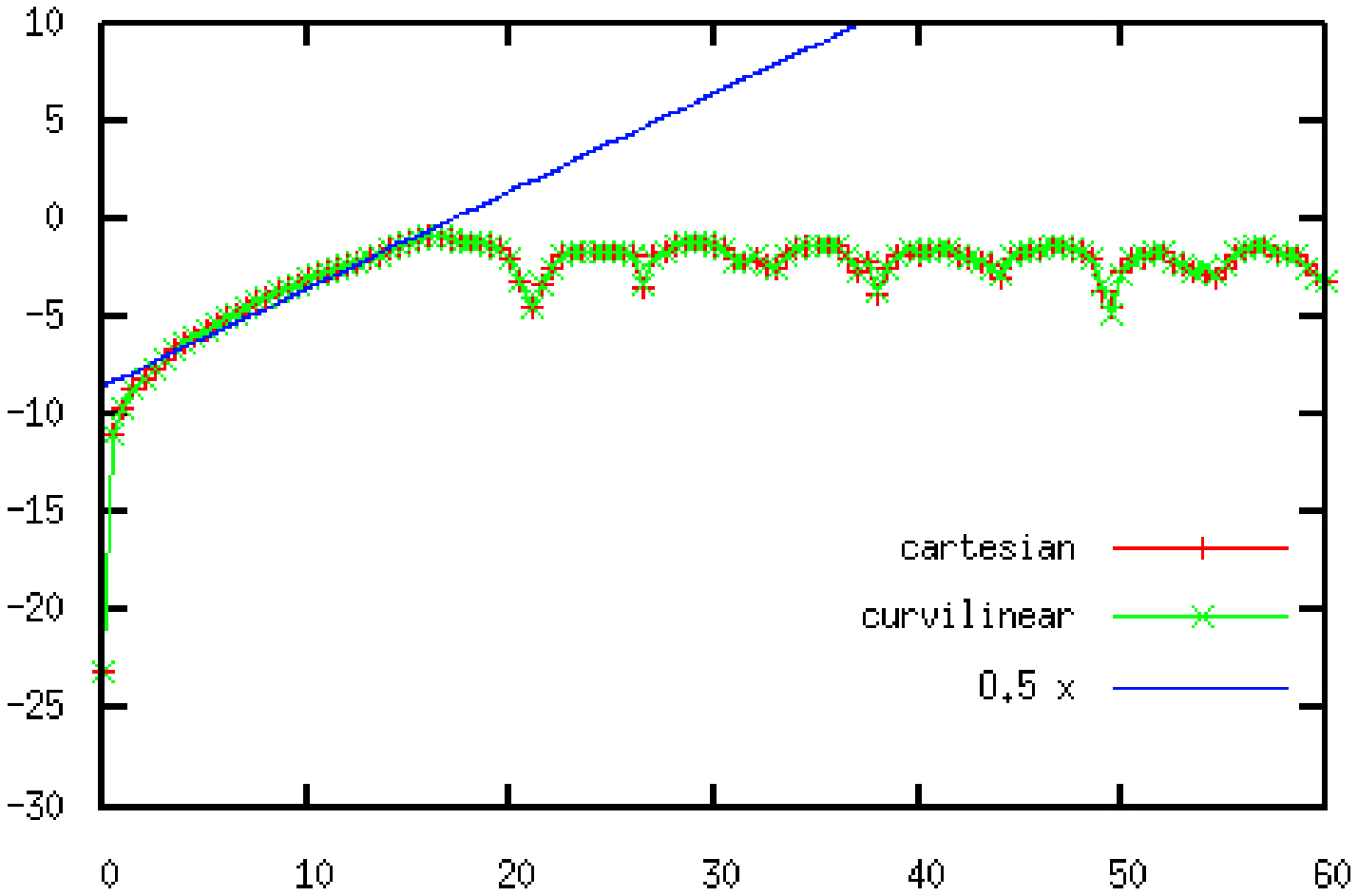
Mapped mesh GC model

Qualitative comparison between PSM computations on cartesian and curvilinear meshes of a periodic-periodic instability for the GC model.



Mapped mesh GC model

Fourier analysis of the solution:
The growth rate of the unstable mode is in good agreement in the linear phase with the expected value $Im(\omega) = 1/2$ for both meshes.



Mapped mesh Vlasov-Poisson for beam simulation

J. Abiteboul, G. Latu, V. Grandgirard, A. Ratnani, E. Sonnendrucker and A. Strugarek ESAIM Proceedings (to appear).

- 2D mapped grid based either on analytical mapping or Bézier mapping.
- Characteristics followed backward in physical space. Inverse mapping is used to bring points back to logical space for interpolation.
- Velocity integral needs interpolation. Sparse matrix can be constructed for this at initialization.
- Performance (1024 time steps, 512x512 mesh)

	Field solve	Spline coeff.	Advection	Total
no mapping	2.3	16.4	78	97
analytical	10.1	18.4	115	143
bézier	9.2	17.3	275	301

A few issues with semi-Lagrangian methods on mapped grids

- In semi-Lagrangian framework, computation of characteristics well split.
- Free stream preservation exact when characteristics are computed in physical space.
- No exact free stream preservation when characteristic computed using logical coordinates. However verified up to the accuracy of computation of characteristics, which can be very good and robust using adequate ODE solver.
- Velocity integral needs inverse mapping, but can be precomputed at initialization for interpolation points.
- Discrete divergence of advection field should exactly vanish.

Outline

4D drift-kinetic model

Similar model as gyrokinetic models with $\mu = 0$ in cylindrical geometry $\Rightarrow$ relevant for numerical tests.

The 4D velocity field $a = (v_{GC_r}, v_{GC_\theta}, v_\parallel, q/m_i\ E_z)^t$ is divergence free:

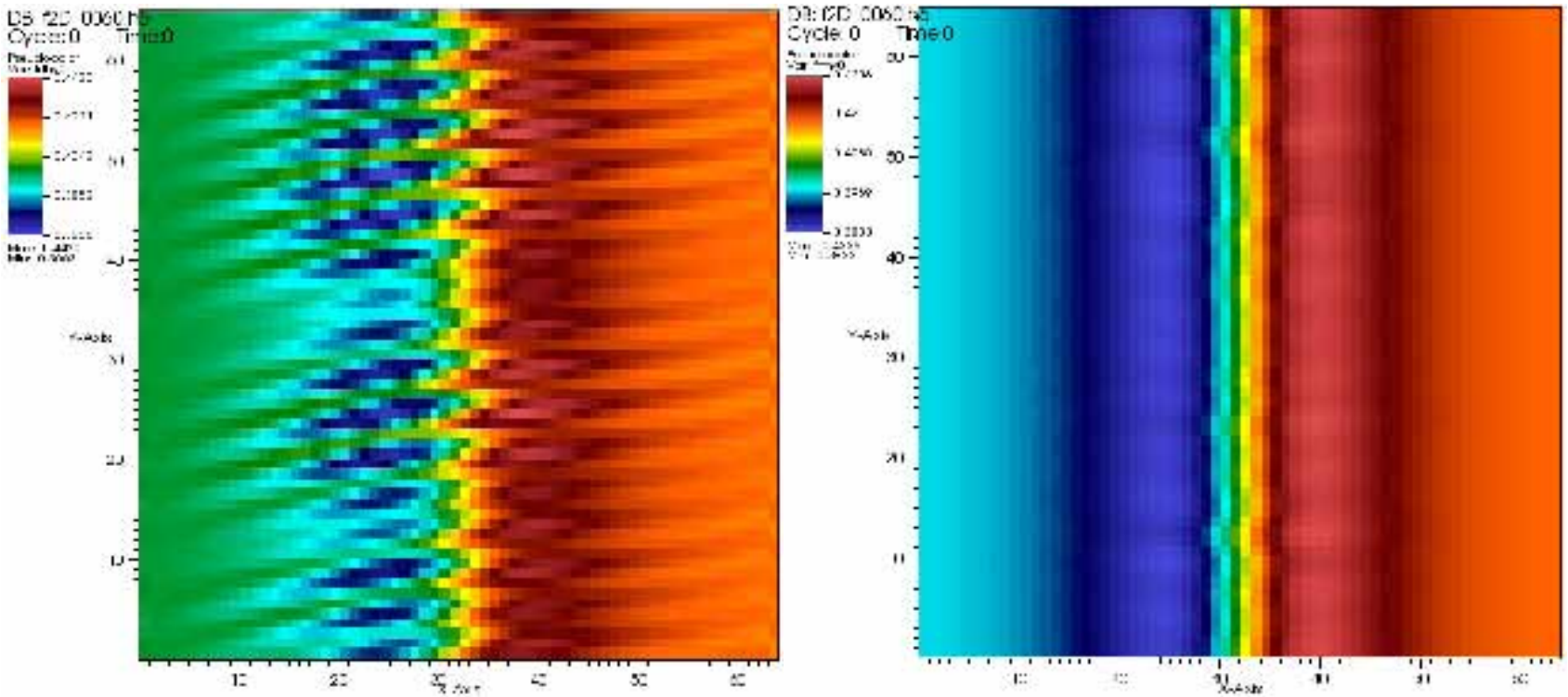
$$\nabla \cdot a = \frac{1}{r}\partial_r(r\ v_{GC_r}) + \frac{1}{r}\partial_\theta(v_{GC_\theta}) + \partial_z v_\parallel + \partial_{v_\parallel}(q_i/m_i\ E_z) = 0$$

with

$$v_{GC_r} = \frac{1}{B_z}\left(-\frac{1}{r}\partial_\theta \Phi\right) \quad \text{and} \quad v_{GC_\theta} = \frac{1}{B_z}(\partial_r \Phi)$$

This property should be satisfied at discrete level, to conserve cell volumes in the phase space and thus have a maximum principle.

4D drift-kinetic model



Advection field computed with cubic splines (left) and with one that satisfies the discrete divergence free condition (right).
4D simulation of 128x128x64x32 cells, result at time $t = 60$.

4D drift-kinetic model

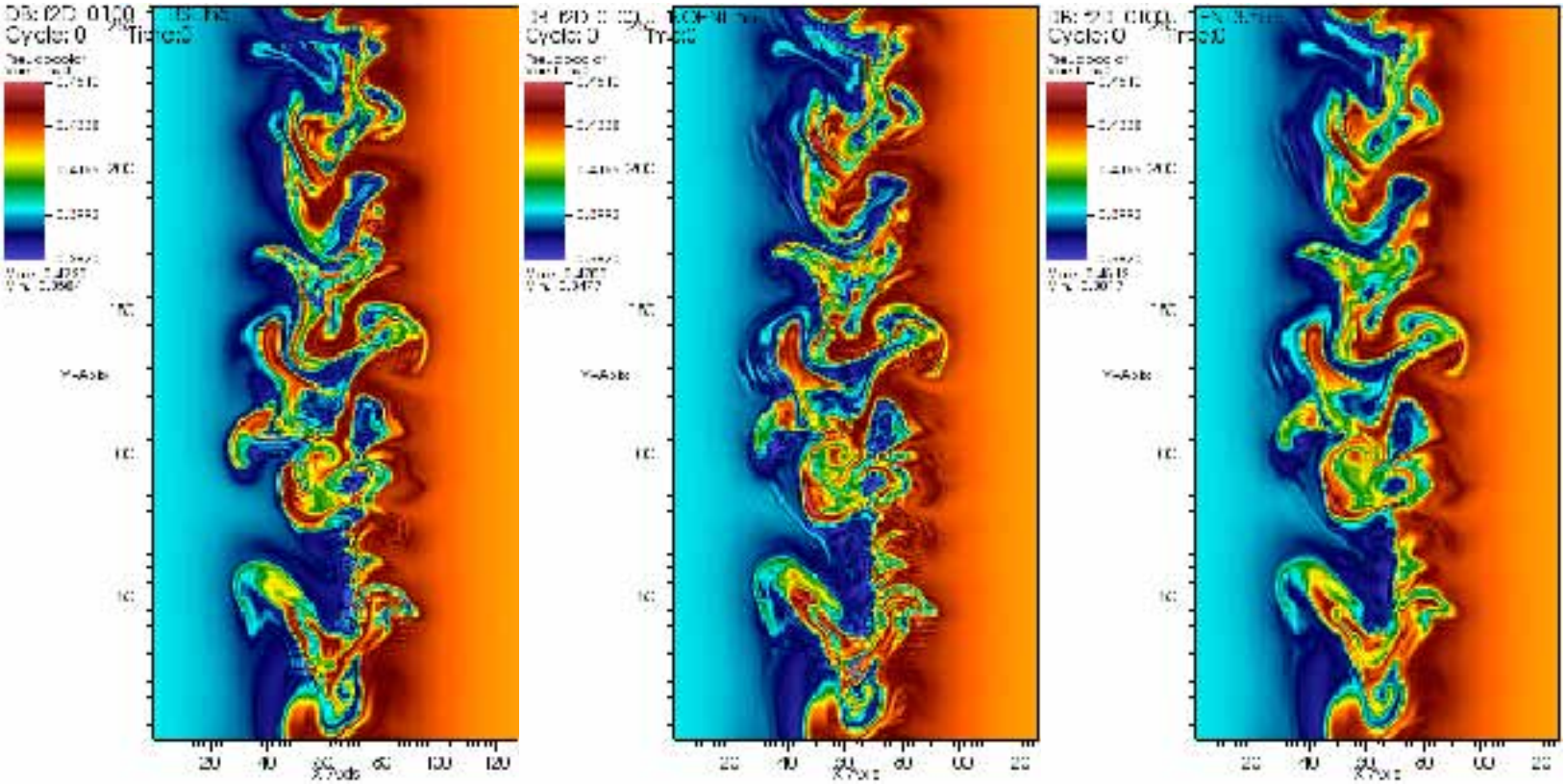
The PSM scheme does not ensure a maximum principle (positivity), which is crucial for a model without physical dissipation.
The following conditions should be satisfied at discrete level:

- cells volumes conservation in the phase space $Vol_i^n = Vol_i^{n+1}$, even when using a directional splitting.
- the average of the reconstructed function on an arbitrary volume satisfies the maximum principle.

$\Rightarrow$ we propose numerical limiters of two kinds:

- correction of the cubic spline reconstruction using monotonicity criterions (Crouseilles, Mehrenberger, Sonnendrücker, JCP 2010),
- "Entropic Flux Limiter" that evaluates the sign of the numerical diffusion and cuts off anti-diffusion when occurs (Braeunig et al, INRIA Report 2009).

4D drift-kinetic simulation



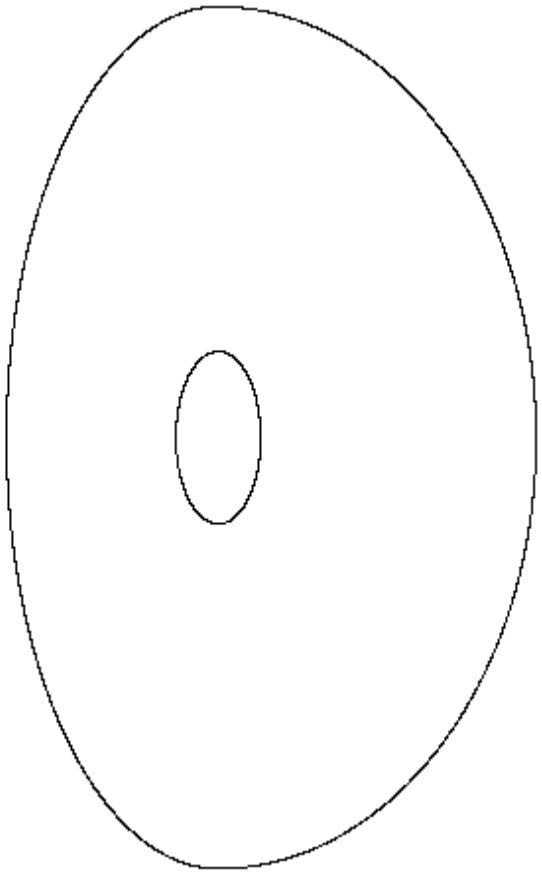
From left to right: BSL, PSM and PSM with entropic flux limiter.
4D simulation of 128x256x128x64 cells, time =2000.

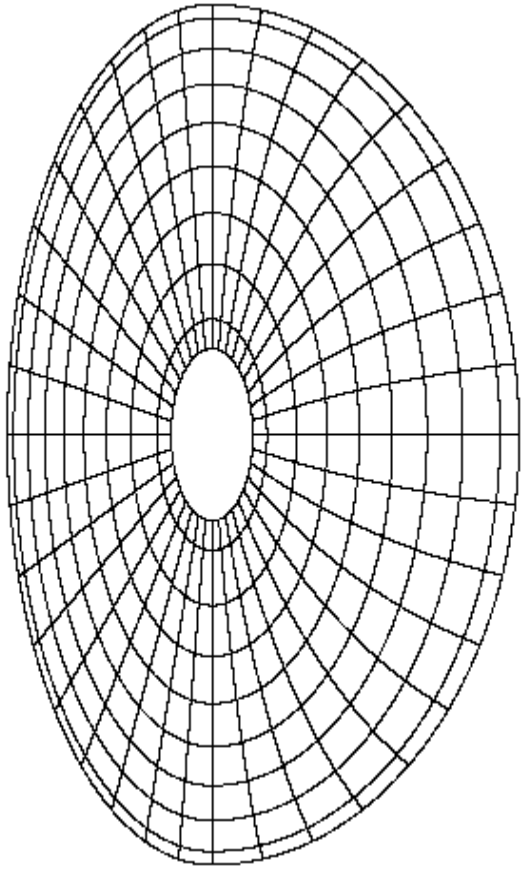
Outline

Principles of isogeometric analysis

- New concept recently developed by Hughes, Bazilevs and Cottrell as an alternative to finite element modeling.
- Finite Elements: define a mesh of the computational domain - associate degrees of freedom and basis functions to each cell.
- Isogeometric analysis: Computational domain defined by NURBS as is traditional in CAD. Use tensor product of knots to define logical mesh and use NURBS or splines as basis functions of discrete space for solution.
- NURBS allow an exact description of conics and a very good and smooth approximation of most domains.

NURBS mesh of the poloidal plane of a Tokamak





Quasi-neutral equation = standard elliptic equation

- The variational formulation is of the form:
  Find $u \in \mathcal{S}$ such as:

$$a(w,u) = l(w), \qquad \forall w \in \mathcal{V} \tag{1}$$

where

$$a(w,v) = \int_\Omega A\nabla v \cdot \nabla w,$$

and

$$l(w) = \int_\Omega fw + \int_{\Gamma_N} hw.$$

- Use NURBS mapping of computational domain to compute integrals like for standard Finite Elements.
- Integrals are computed using high order Gauss-Legendre quadrature.

First test case: Laplace equation in annulus

Order of convergence:

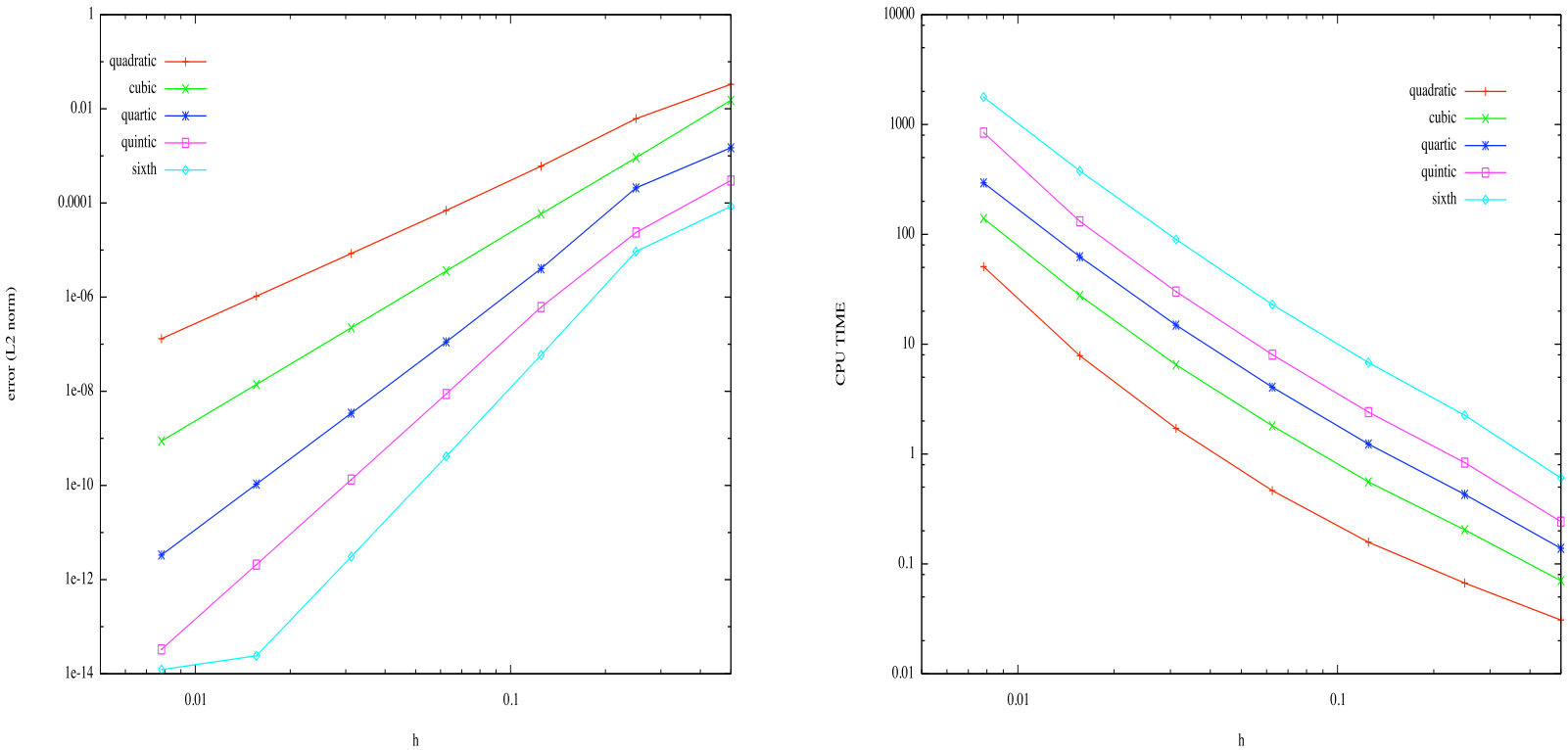


Figure: *Validation test* : (left) L^2 error norm, (right) CPU time.

Turbulent solution

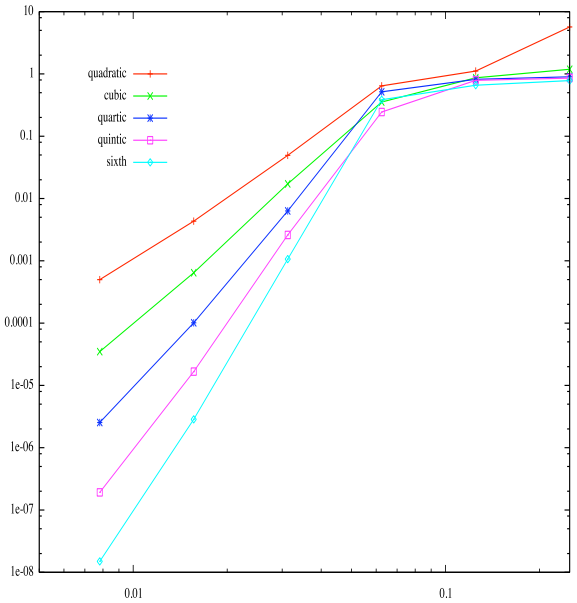
The analytical solution is :

$$\phi(r,\theta) = \sum_{m=-M_{max}}^{M_{max}} \sum_{l=-l_{max}}^{l_{max}} A_l(\delta_{m,1} + \epsilon B_m)\sin(2\pi m\kappa(x,y))\cos(l\theta + \Theta_l)$$

avec $\kappa(x,y) = \frac{r-r_{min}}{r_{max}-r_{min}}$
A_l and B_M are random numbers which range between 0 and 1,
the phase Θ_l is also given by a random number in $0, 2\pi$.
In our numerical experiments :

- $|M|, |l| (\leq 20$ for the first simulation and ≤ 40 for the second one).
- $\epsilon = 0.4$
- Dirichlet boundary condition at $r = r_{\text{min}} = 0.2$ and $r = r_{\text{max}} = 0.4$.

Order of convergence



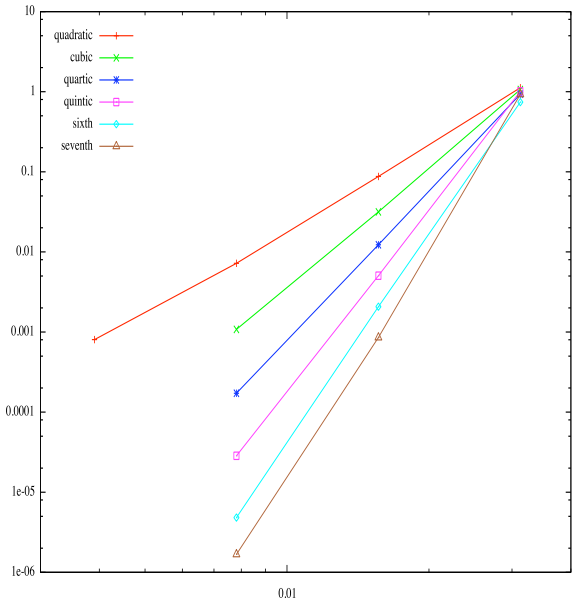


Table: L^2 error norm as a function of h; (left) for 20 modes, (right) 40 modes.

Error table for given mesh

Spline degree	Degrees of freedom	L^2 error norm
2	17408	$7.20\ 10^{-3}$
3	18060	$1.07\ 10^{-3}$
4	18720	$1.71\ 10^{-4}$
5	19388	$2.84\ 10^{-5}$
6	20064	$4.81\ 10^{-6}$
7	20748	$1.67\ 10^{-6}$

Table: Number of degree of freedom and L^2 norm of the error for each spline degree, with $N = 128$ in the case of 41 modes

Comments

Comparing with results of
Y. Nishimura, Z. Lin, J.L.V. Lewandowski, S. Ethier, *A finite element Poisson solver for gyrokinetic particle simulations in a global field aligned mesh*, J. Comput. Phys., **214**, pp. 657-671, (2006).

- we can reach same error order, with much fewer degrees of freedom, by increasing the spline order and reducing the knot multiplicity,
- to reach an error of 10^{-6}, Nishimura used a grid of $N = 154,560$, with our method, we only need 8 times less ($N = 20064$) degrees of freedom by using sixth or seventh order for spline basis.

Fast Isogeometric Analysis

- For domains with specific structure: cartesian, polar, ..., fast tensor product solvers can be used.
- Can be extented to a large class of domains.
- e.g. for polar domains: spline finite element matrices feature very specific structure as integrals in r and θ decouple:

$$K_{ar} U M_\theta + M_{ar} U K_\theta + M_{cr} U M_\theta = F,$$

$$K_{ar} = ((\int a(r)\varphi_i'(r)\varphi_j'(r)\,dr)), \quad M_{ar} = ((\int a(r)\varphi_i(r)\varphi_j(r)\,dr)),$$

$$K_\theta = ((\varphi_i'(\theta)\varphi_j'(\theta)\,d\theta)), \quad K_\theta = ((\int \varphi_i(\theta)\varphi_j(\theta)\,d\theta)),$$

- With unknown U and F given in matrix form

$$U = ((u_{i,j})), \qquad F = ((f_{i,j})).$$

Faster Isogeometric Analysis with periodic direction

- Periodicity and translation invariance of splines yield symmetric circulant matrices M_θ and K_θ,

$$C = \begin{pmatrix} c_0 & c_1 & c_2 & c_1 \\ c_1 & c_0 & c_1 & c_2 \\ c_2 & c_1 & c_0 & c_1 \\ c_1 & c_2 & c_1 & c_0 \end{pmatrix}.$$

- All symmetric circulant matrices are diagonalizable in same basis. Transformation matrix = Discrete Fourier Transform matrix. Eigenvalues known as functions of c_j:

$$\lambda_k = c_0 + 2\sum_{j=1}^{N/2} c_j \cos\frac{2\pi jk}{N}.$$

- Transformation implemented using FFT: $O(N_r N_\theta \log_2 N_\theta)$

A fast algorithm

- Start from $AU\overbrace{P\Lambda_M P^*}^{M=} + MU\overbrace{P\Lambda_K P^*}^{K=} = F$.
- Multiply on the right by P (i.e. apply FFT to lines of F)
- Denoting by $\hat{U} = UP$ and $\hat{F} = FP$, we have $A\hat{U}\Lambda_M + M\hat{U}\Lambda_K = \hat{F}$, i.e. set of decoupled banded systems involving columns (U_k and F_k)

$$\lambda_{M_k} A\hat{U}_k + \lambda_{K_k} M\hat{U}_k = \hat{F}_k.$$

- Compute U from $\hat{U}$ right-multiplying by P^*, i.e. applying inverse FFT to lines of U.

Computational cost

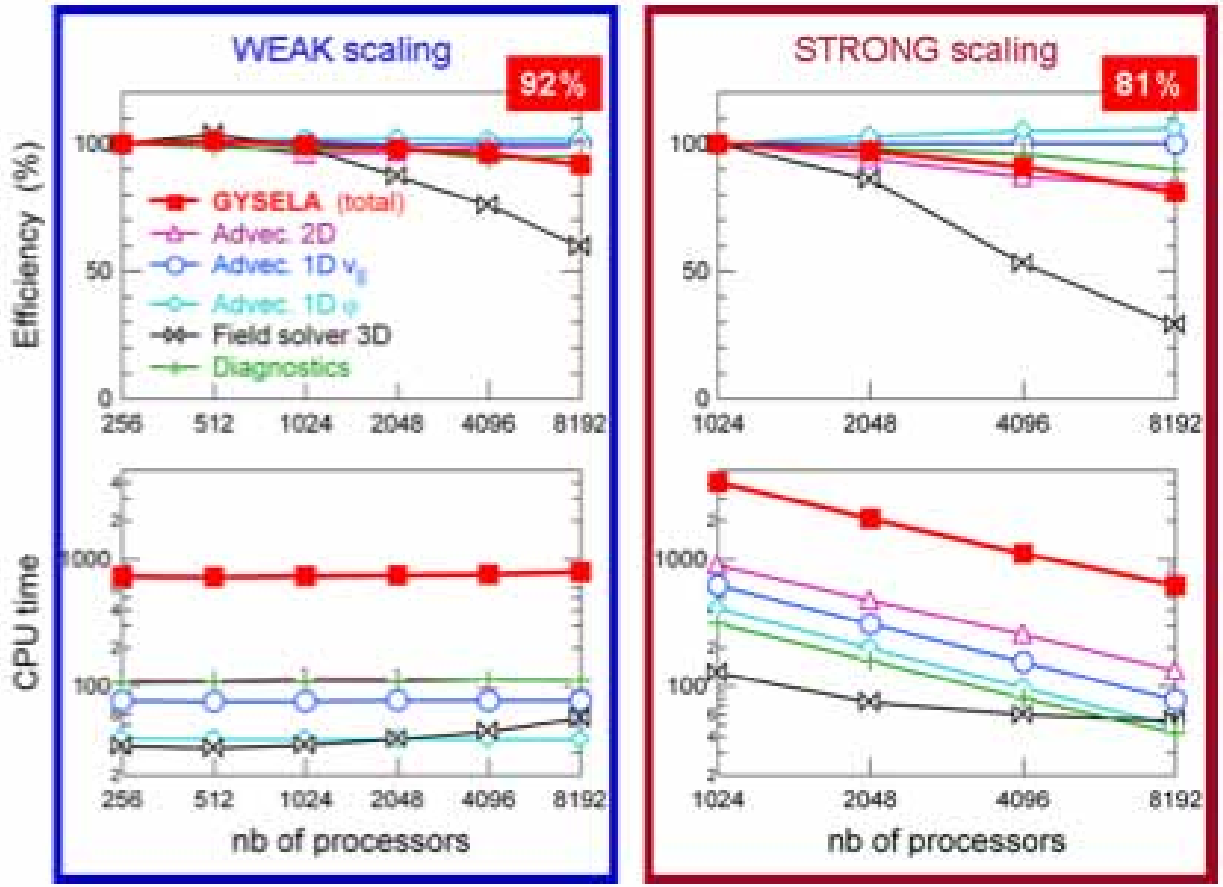
- Compared with superLU (standard direct sparse matrix solver).
- More fill in for superLU when spline degree increases.
- Small additional bandwith in fast solver does only marginally increase computation time when spline degree increases.
- 128×128 grid.

Spline degree	FIGA	SuperLU
1	0.012	0.5
2	0.014	4.58
3	0.014	11.55
4	0.013	20.59
5	0.015	32.62
6	0.015	52.54
7	0.015	80.08

Outline

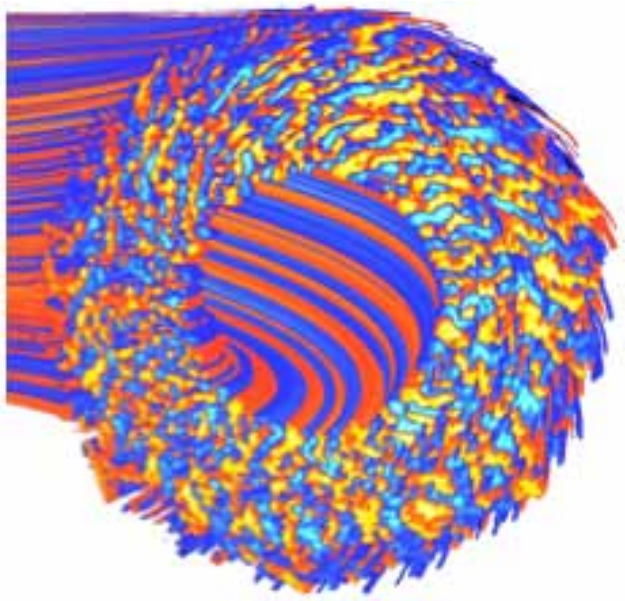
Efficiency of GYSELA parallelization

- Hybrid MPI/OpenMP parallelization.
- 92% efficiency on 8192 processors.



Biggest simulation ever run with GYSELA

- A simulation close to ITER-size scenario ($\rho^* = 1/512$) performed on $1/4$ torus with additional heating power of $60MW$ during 1 ms.
- JADE (SGI Altix ICE 8200, 23040 cores on 2880 nodes) at CINES (Montpellier)
- 5D mesh of $(r, \theta, \varphi, v_{\|}, \mu) =$ $1024 \times 1024 \times 128 \times 128 \times 16 =$ 272×10^9 points.
- 31 days on 8192 cores.
- 6.5 TBytes of ouput data.
- See J. Abiteboul EPS 2010, Y. Sarazin IAEA 2010.



Ion temp. fluctuations

Outline

Conclusions and perspectives

- Semi-Lagrangian method has proved to be a good choice for full-f gyrokinetic simulations.
- 5D gyrokinetic code efficiently running on several thousands of processors.
- Semi-Lagrangian idea very fruitful to remove the numerical time step constraint.
- Several flavours: backward, forward, finite volume.
- All flavours identical for constant coefficient advections (used in time-split Vlasov-Poisson simulations).
- What option to best handle anisotropy and long time accuracy at minimal cost ?
- Can accomodate easily and efficiently structured meshes based on mapped coordinates.
- For some applications like gyrokinetics different directions play very different roles. Only perpendicular plane features strong nonlinear effects. Consider different methods for different directions.