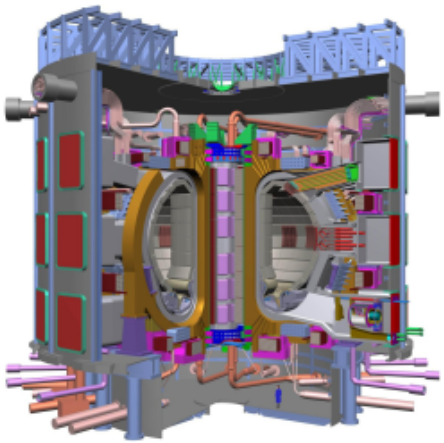
TArgeting Realistic GEometry in Tokamak code gysela



Collaborators of CEA:
Virginie Grandgirard, Guillaume Latu

Collaborators of INRIA and University of Strasbourg:
Michel Mehrenberger, Nicolas Bouzat

Collaborators of IPP Garching (Germany):
Camilla Bressan

25/08/2016

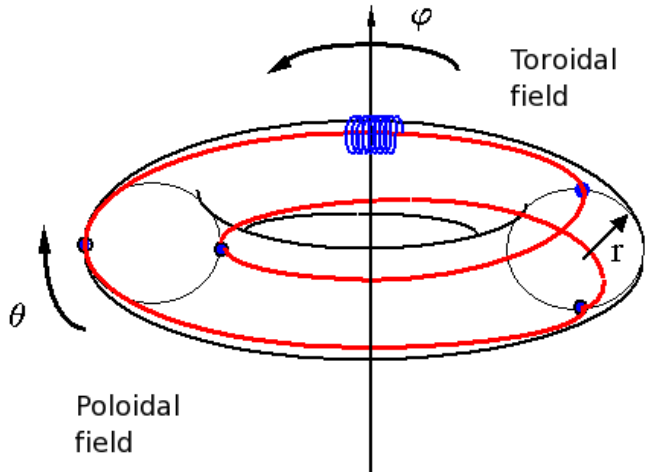
Outline

1. Introduction: Gyrokinetic setting
   - Gyrokinetic
   - Code integration
   - Improving the polar mesh
2. Experimental study: convergence, test cases
   - 2D-Lagrangian interpolation
   - Advection
   - Gyroaverage
3. Final words

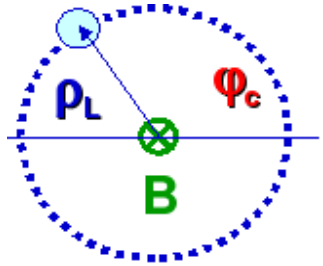
Kinetic approach, Gyrokinetic theory

Tokamak kinetic model: 6D phase space

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

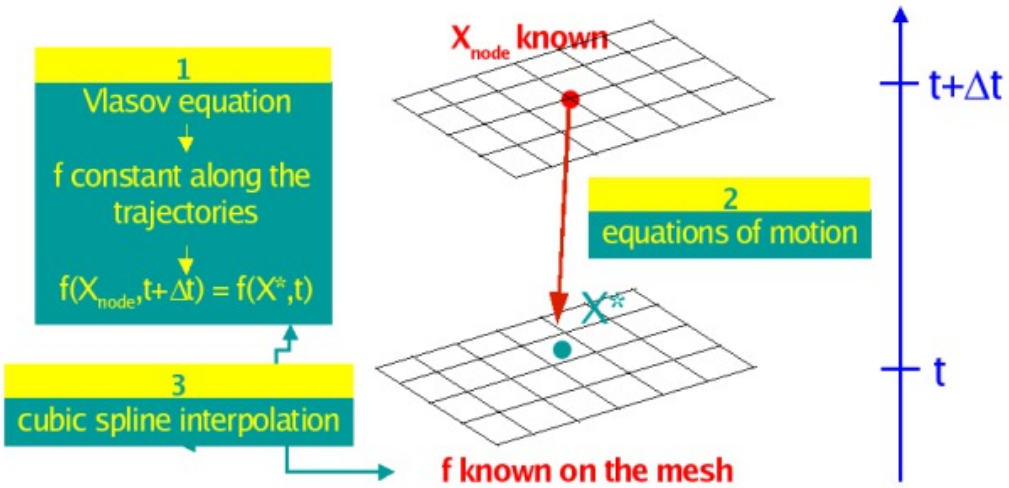


Time scale of particle gyration
$\ll$ Time scale of ITG turbulence



Gyrokinetic model 5D+t: Data $\bar{f}(r, \theta, \varphi, v_\parallel, \mu, t)$
Vlasov equation to solve at each time step t
$\Downarrow$
Large memory and CPU consumption
Aim of the project: improve **Gysela** and **Selalib** codes

The Semi-Lagrangian method (backward version)



- f conserved along characteristics
- Find the origin of the characteristics ending at the grid points
- Interpolate value at origin $X^\star$ of characteristics from known grid values $\rightarrow$ Interpolation method needed

- Typical interpolation schemes:
  - Cubic spline
  - Lagrange polynomial (high order)

Gysela 5D: parallel domain decomposition

Input : *Physics parameters*, $\bar{f}^0$
Output : *Diagnostics*

for *time step* $n \geq 0$ **do**

Integration: $\mathcal{N}_i^n(r,\theta,\varphi) = \int\int$ **Gyroaverage of** $\bar{f}^n$ $B(r,\theta)\, dv_\parallel d\mu$;
Solve **Field equation**: $\mathcal{N}_i^n(r,\theta,\varphi) \rightarrow$ **Gyroaverage of** $\Phi^n(r,\theta,\varphi)$;
Broadcast of **Gyroaverage of** Φ^n to all processors;
Diagnostics for time step n;
1D Advection in $v_\parallel$ $(\forall(\mu,r,\theta) = [local], \forall(\varphi, v_\parallel) = [*])$;
1D Advection in φ $(\forall(\mu,r,\theta) = [local], \forall(\varphi, v_\parallel) = [*])$;
Transposition of $\bar{f}$;
2D Advection in (r,θ) $(\forall(\mu,\varphi,v_\parallel) = [local], \forall(r,\theta) = [*])$;
Transposition of $\bar{f}$;
1D Advection in φ $(\forall(\mu,r,\theta) = [local], \forall(\varphi, v_\parallel) = [*])$;
1D Advection in $v_\parallel$ $(\forall(\mu,r,\theta) = [local], \forall(\varphi, v_\parallel) = [*])$;

- Gyroaverage, Poisson, 2D advection act on poloidal plane
- Aim: improve the poloidal mesh

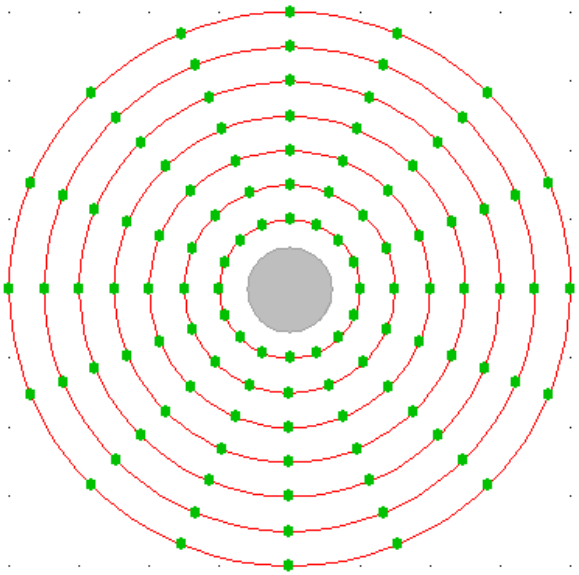
Parallel algorithm for a 2D advection (in r, θ directions)

Input : $\bar{f}^{\star}(r, \theta, \varphi, v_{\parallel}, \mu)$
Output : $\bar{f}^{\diamond}(r, \theta, \varphi, v_{\parallel}, \mu)$

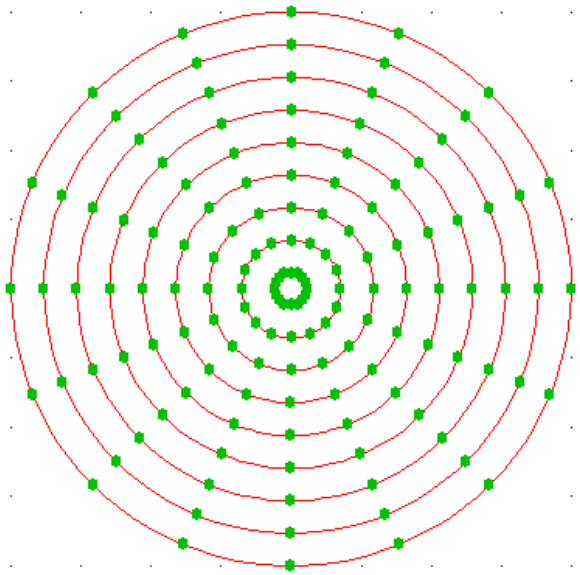
for μ // MPI **do in parallel**
  for $v_{\parallel}$ // MPI+OpenMP **do in parallel**
    for φ // MPI **do in parallel**
      for θ *and* r **do**
        $(\Delta r, \Delta\theta) \leftarrow$ *Compute_displacement*(*Electric field*, Δt)
      Prepare_interpolation (spline coeff if needed)
      for θ *and* r **do**
        $\bar{f}^{\diamond}(r, \theta, \varphi, v_{\parallel}, \mu) =$ *interpolate*$(\bar{f}^{\star}(r - \Delta r, \theta - \Delta\theta, \varphi, v_{\parallel}, \mu))$

Algorithm 1: Advection in variables along r, θ dimensions on $\bar{f}^{\star}$

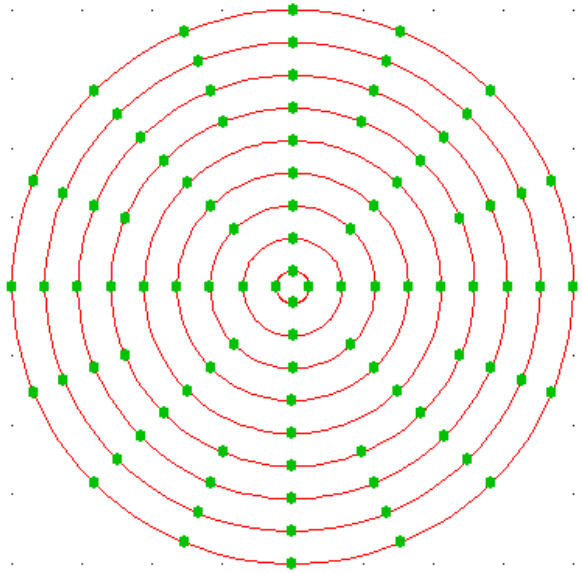
Improving the polar mesh



Old grid
(*hole* in the center)



New uniform grid
Good point:
2D Poisson solver available



New non-uniform grid
Good point:
Save points

2D-Lagrangian interpolation

- Simple Lagrange interpolation on r, θ neighbouring points.
- **Degree** choice has to be studied to achieve desired accuracy.
- Stencil size $\leq$ degree^2
- Accuracy highly drops near the **center** due to **stencil distortion**.

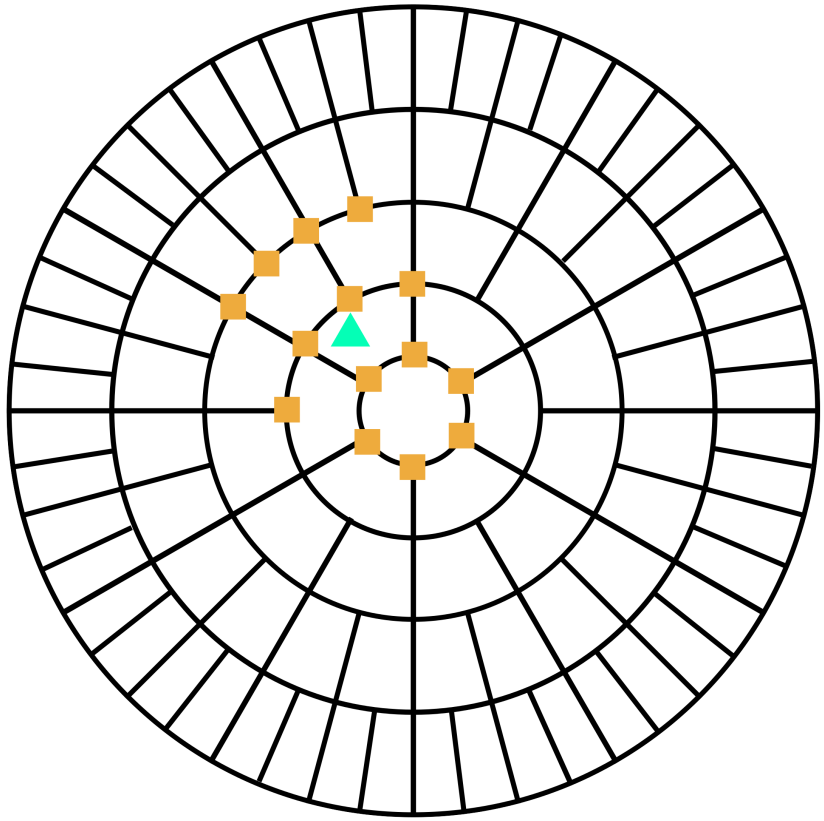
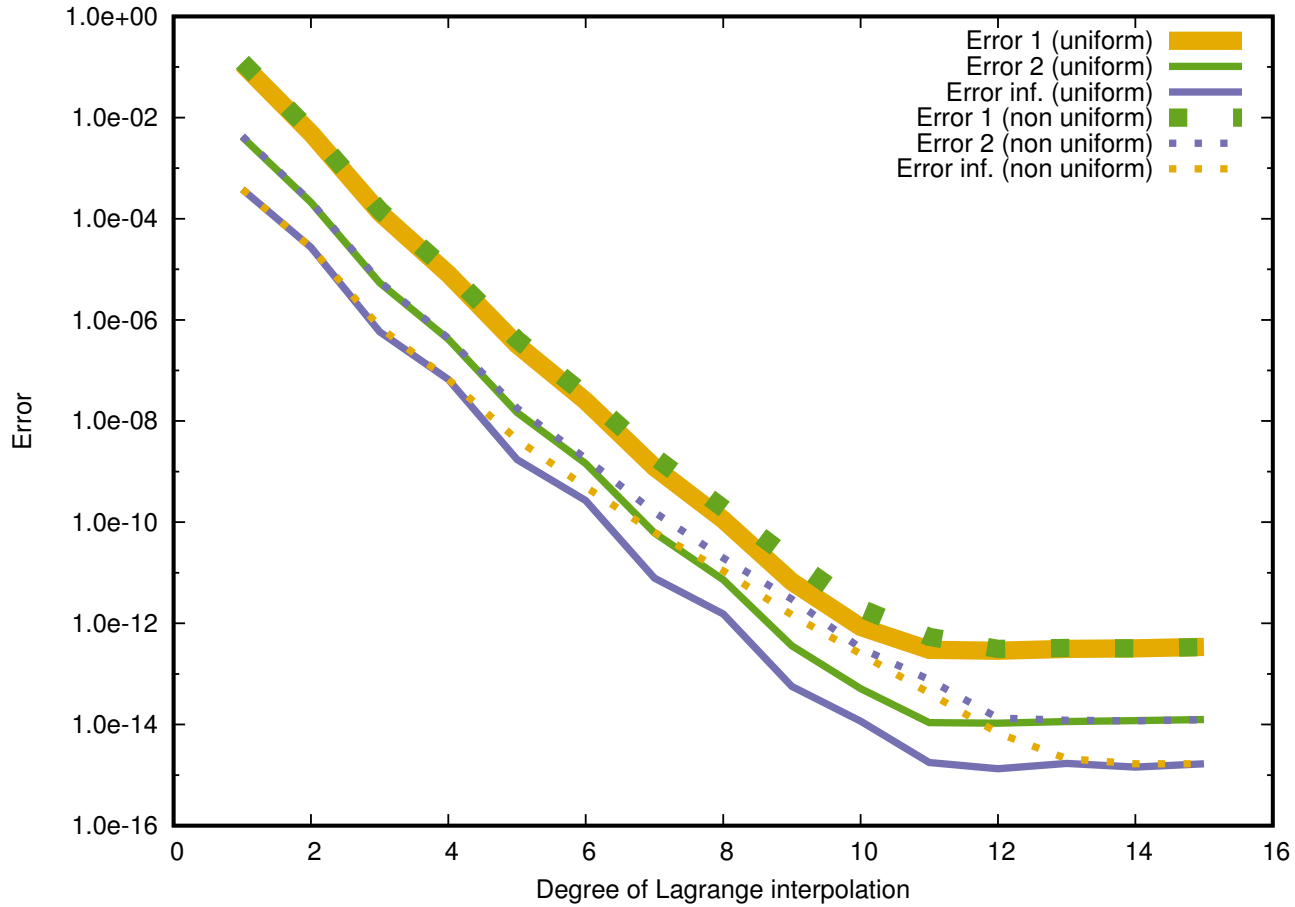


Figure: Lagrange of degree 3; stencil of 4 points

2D-Lagrangian interpolation

Convergence in degree

Test case : $f(x, y) = cos(mx) * sin(ny)$

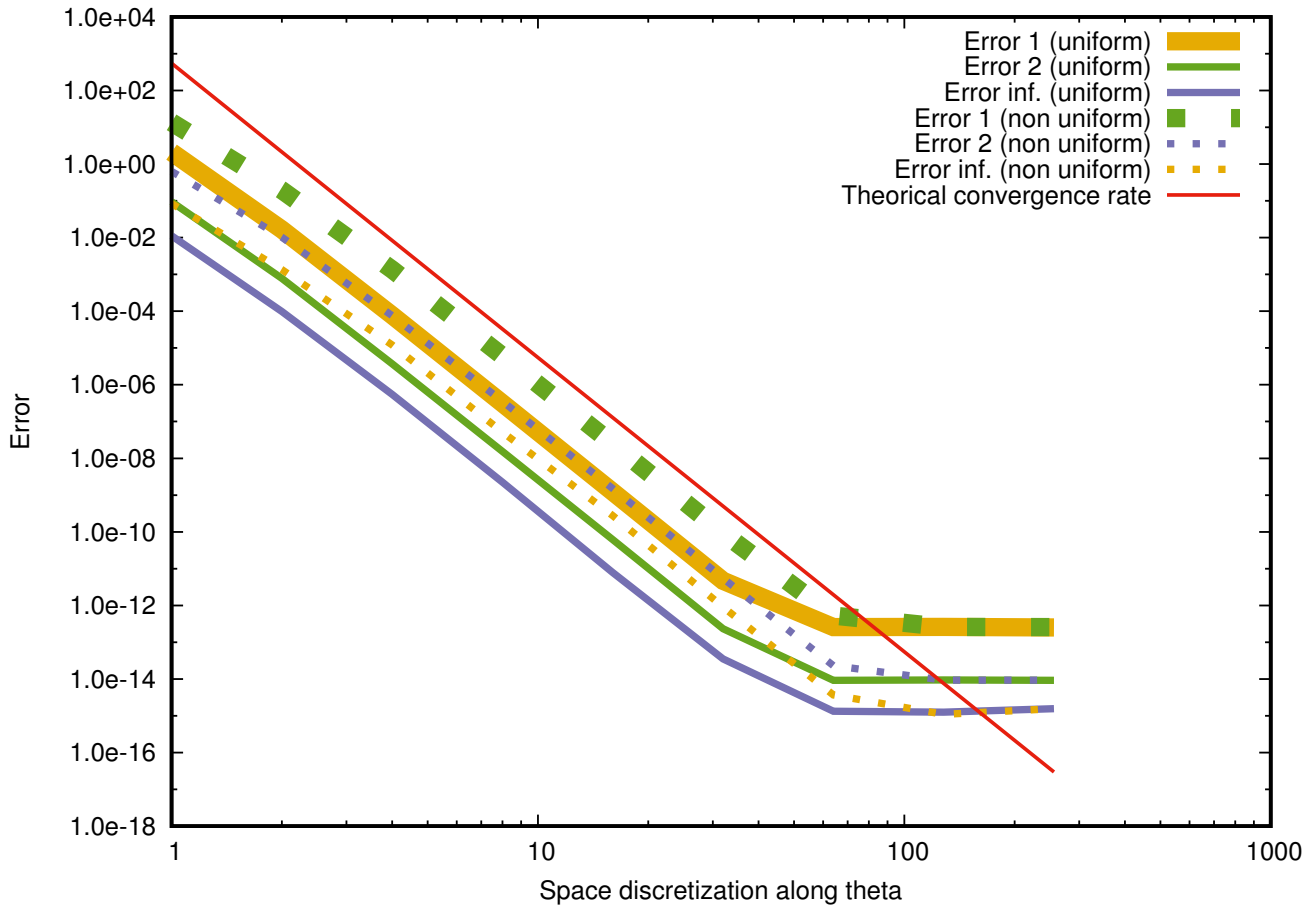


Uniform mesh : 65k points

Non-uniform mesh : 56k points

2D-Lagrangian interpolation

Convergence in space along r (degree 7)

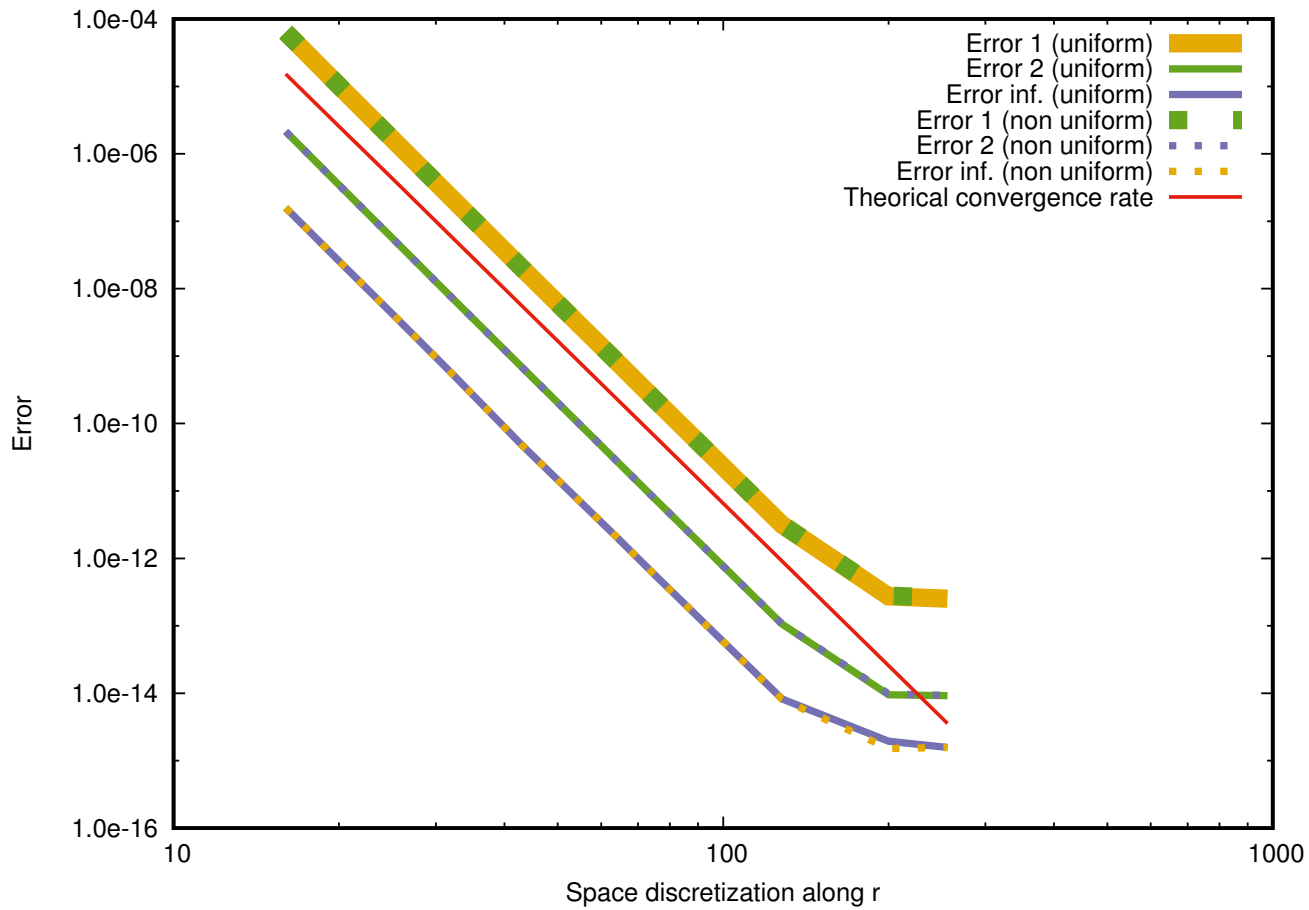


Uniform mesh : 4k points $\rightarrow$ 1 048k points

Non-uniform mesh : 2k points $\rightarrow$ 505k points

2D-Lagrangian interpolation

Convergence in space along *theta* (degree 7)

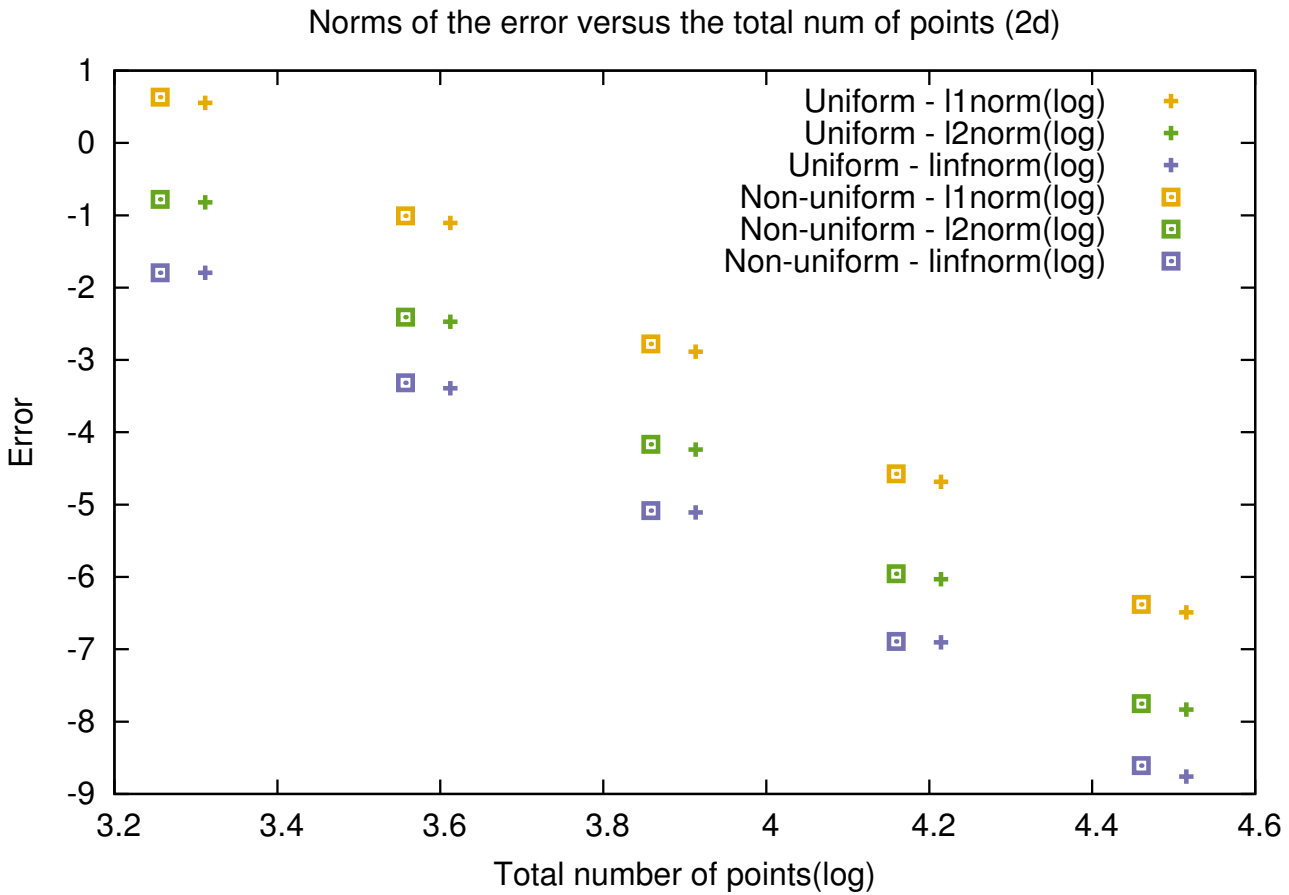


Uniform mesh : 4k points $\rightarrow$ 1 048k points

Non-uniform mesh : 2k points $\rightarrow$ 505k points

2D-Lagrangian interpolation

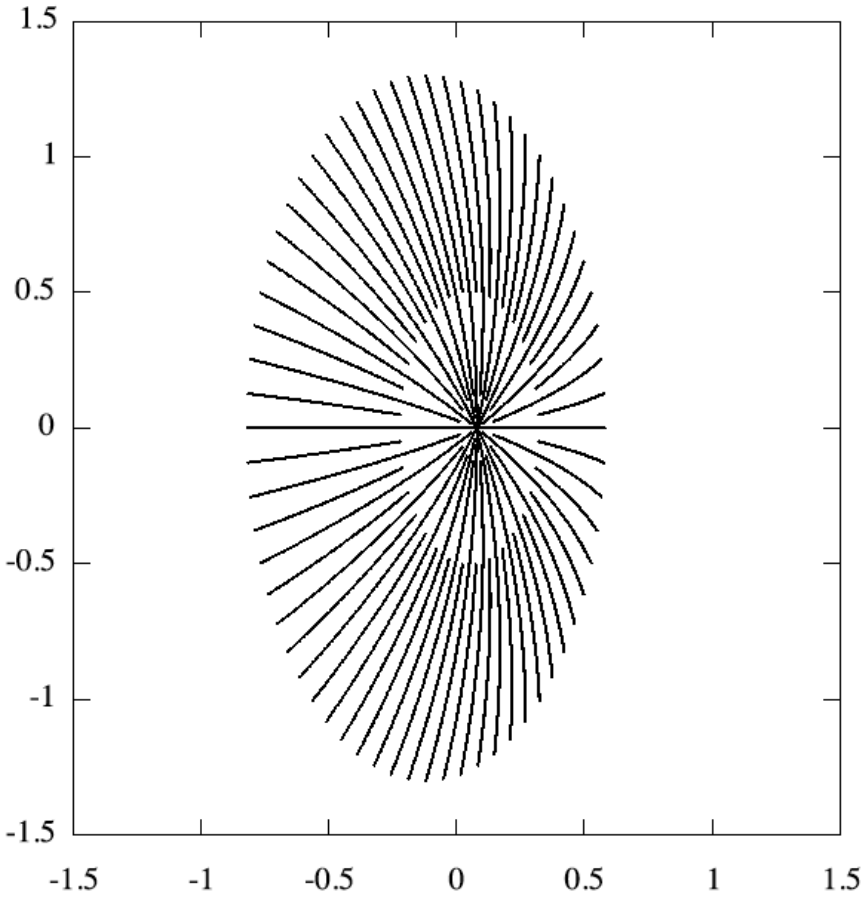
Uniform versus non-uniform grid

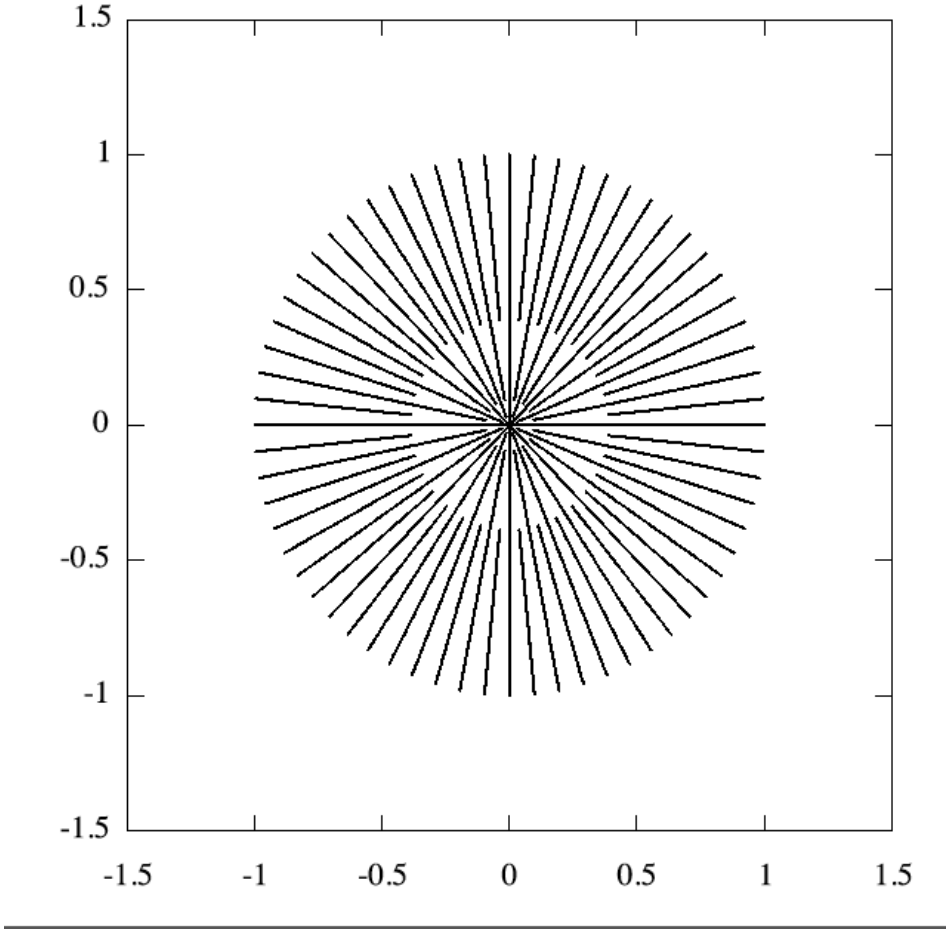


Non-uniform meshes (square) achieves roughly the same error than uniform (cross)

2D-Lagrangian interpolation

Mapping: Culham[1] (left) polar (right) mesh

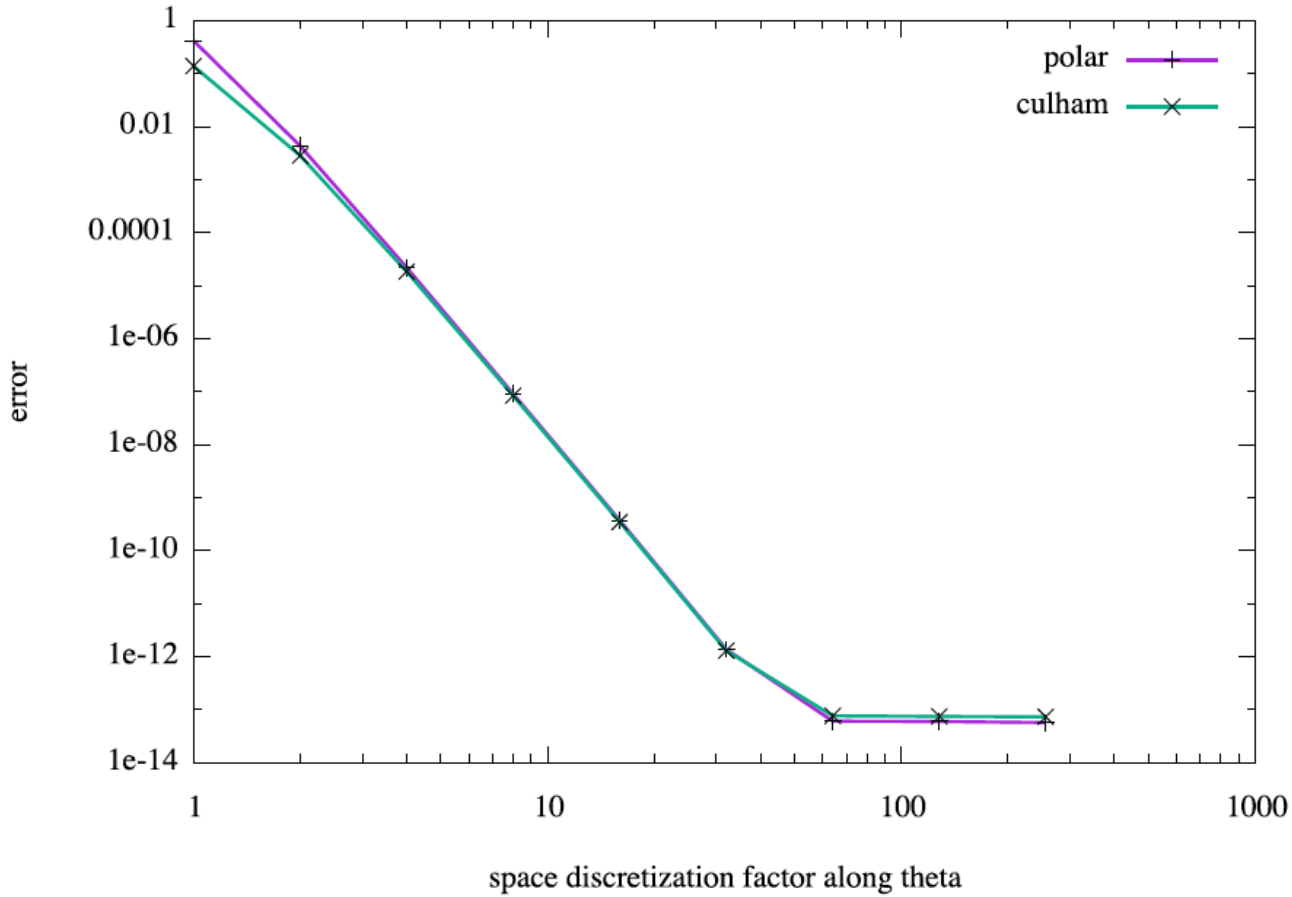




[1] $x = R_0 - \delta_0 r^2 + (1 - E_0) r \cos(\theta),\ y = (1 + E_0) r \sin(\theta)$
$E_0 = 0.3,\ R_0 = 0.08,\ \delta_0 = 0.2$

2D-Lagrangian interpolation

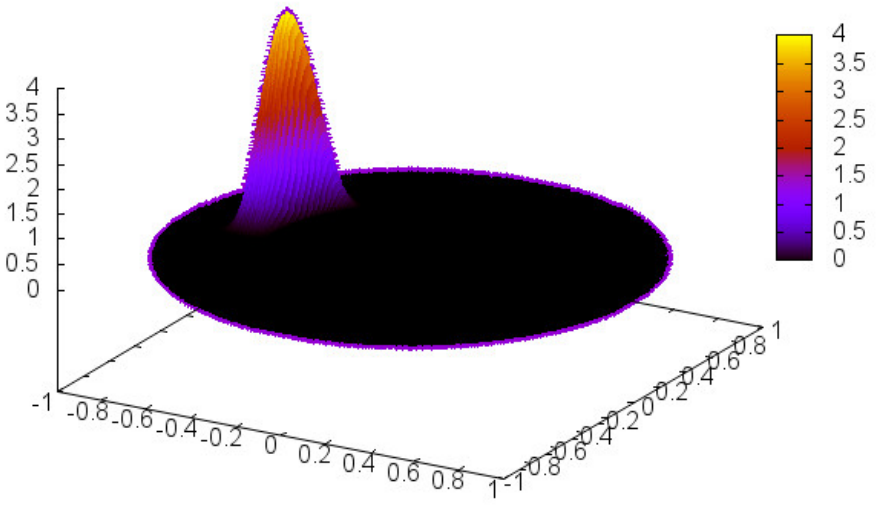
Mapping: convergence in θ



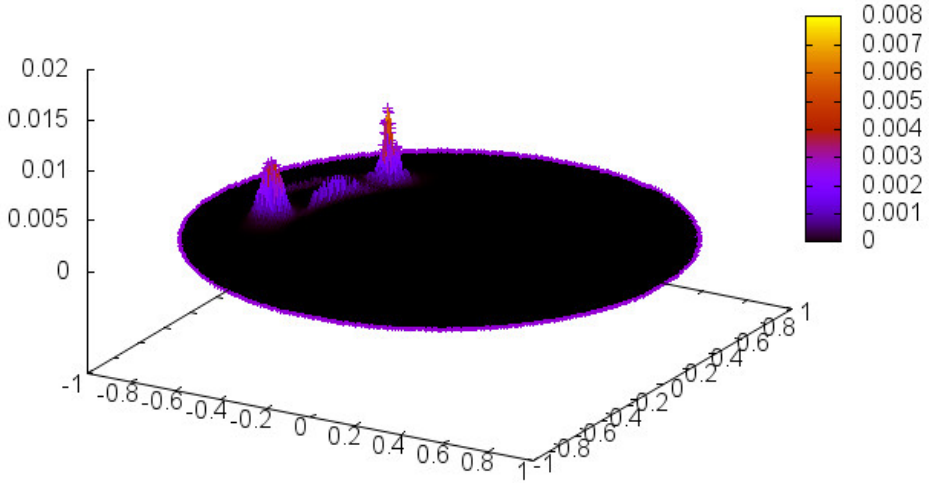
Advection along theta

Error evaluation on uniform mesh (16 384 points)

Degree : 7



Reference function



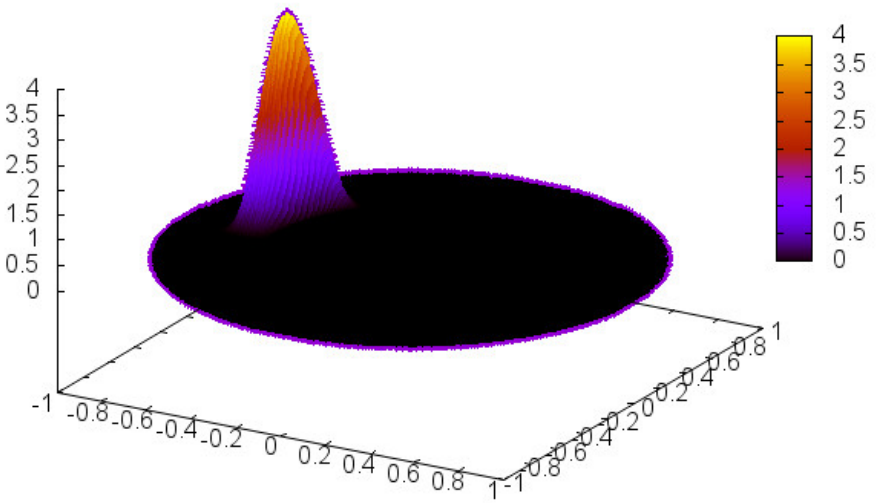
Error
(ref. value - interpolated value)

Advection along theta

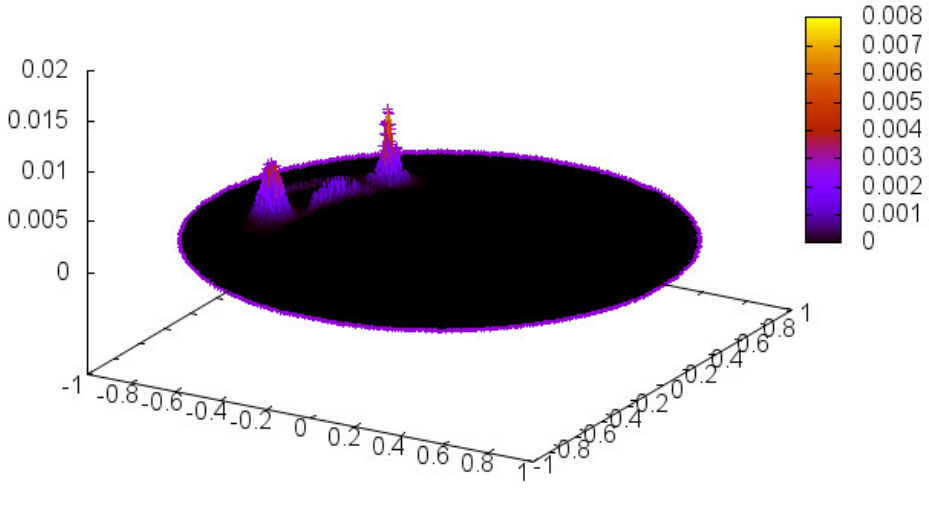
Error evaluation on non uniform mesh (14 480 points)

Degree : 7

Number of points by radius : $8 \times 16, 32 \times 32, 74 \times 64, 68 \times 128$



Reference function

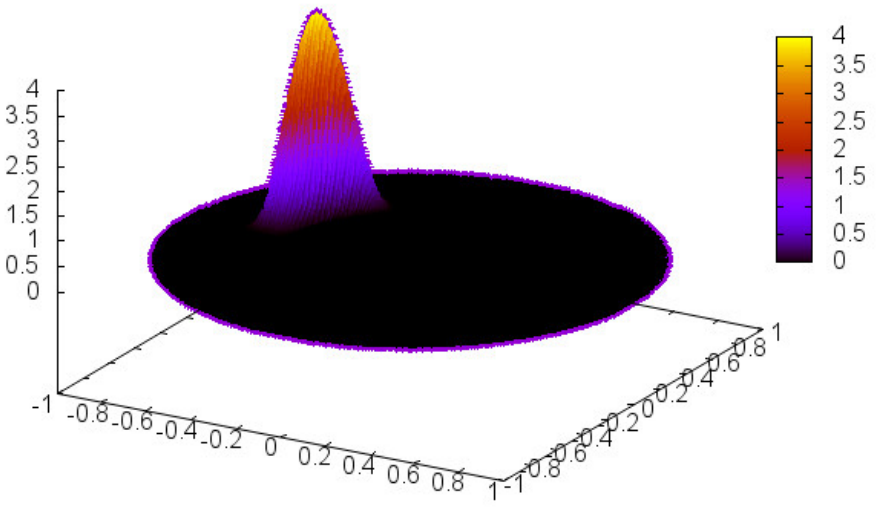


Error
(ref. value - interpolated value)

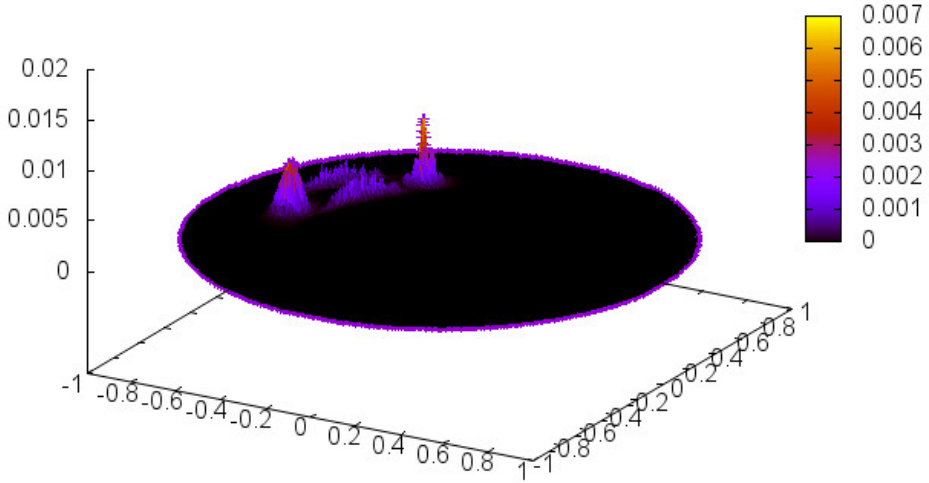
Advection through the center

Error evaluation on uniform mesh (16 384 points)

Degree : 7



Reference function



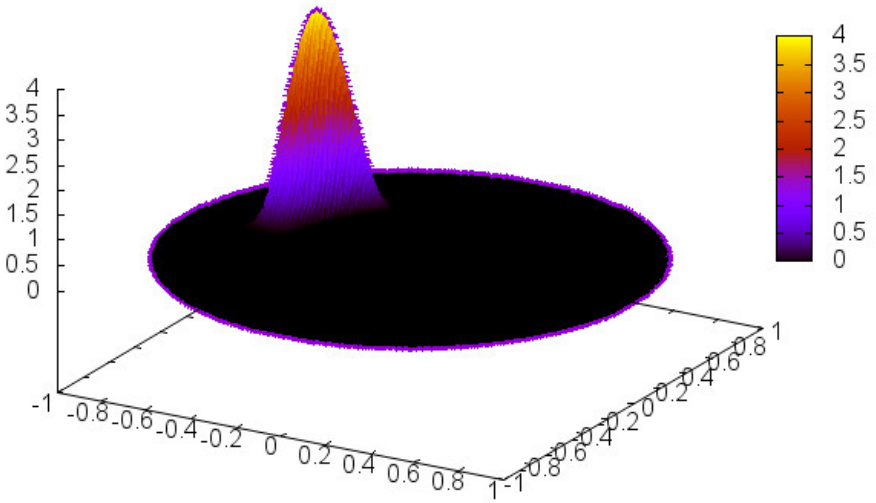
Error
(ref. value - interpolated value)

Advection through the center

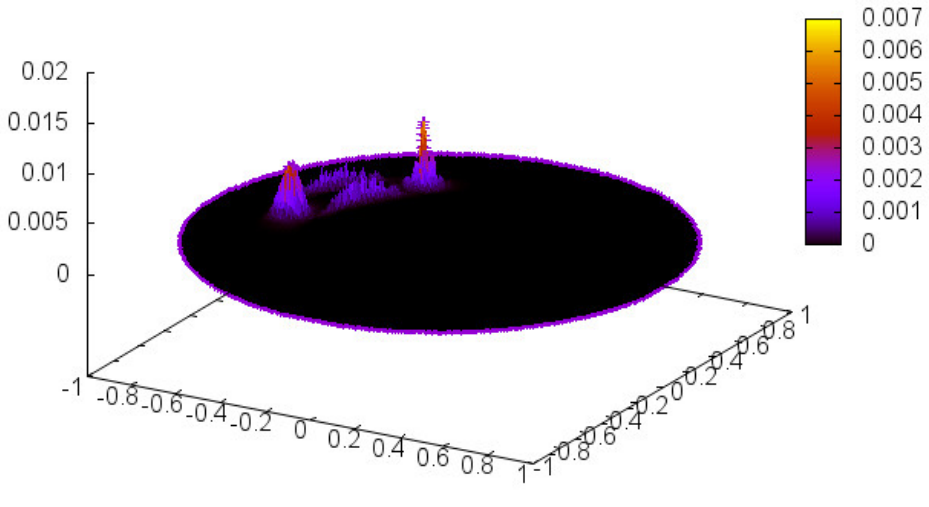
Error evaluation on non uniform mesh (14 480 points)

Degree : 7

Number of points by radius : $8 \times 16, 32 \times 32, 74 \times 64, 68 \times 128$



Reference function



Error
(ref. value - interpolated value)

The gyroaverage operator

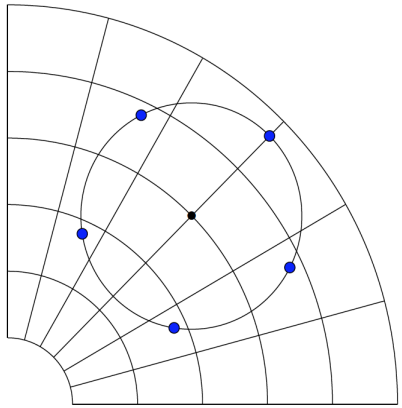
Given the definition of the gyroradius,

$$\rho = \rho(cos(\alpha)\mathbf{e}_\perp + sin(\alpha)\mathbf{e}_\perp)$$

the gyroaverage of a function can be written as:

$$J_\rho(f)(r,\theta) = \frac{1}{2\pi}\int_0^{2\pi} g(\mathbf{x}_g + \rho)d\alpha,$$

f and g being the function defined on a polar and cartesian grid respectively.



Different approaches available, we have explored a method based on interpolation:

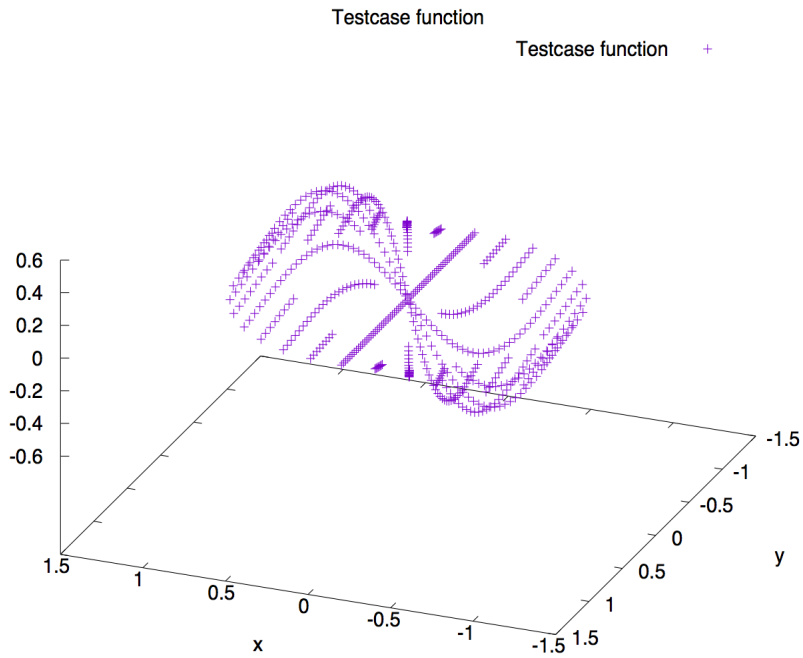
$$J_\rho(f)_{j,k} = \frac{1}{2\pi}\sum_0^{N-1} P(f)(r_j cos(\theta_k) + \rho cos(\alpha_l), r_j sin(\theta_k) + \rho sin(\alpha_l)) \cdot \Delta\alpha,$$

where P is a Lagrange or a cubic spline interpolator, $\alpha_l = l \cdot \Delta\alpha$, $\Delta\alpha = 2\pi/N$.

Testcases and parameters considered

Different testcases $f_1(r, \theta)$ have been considered:

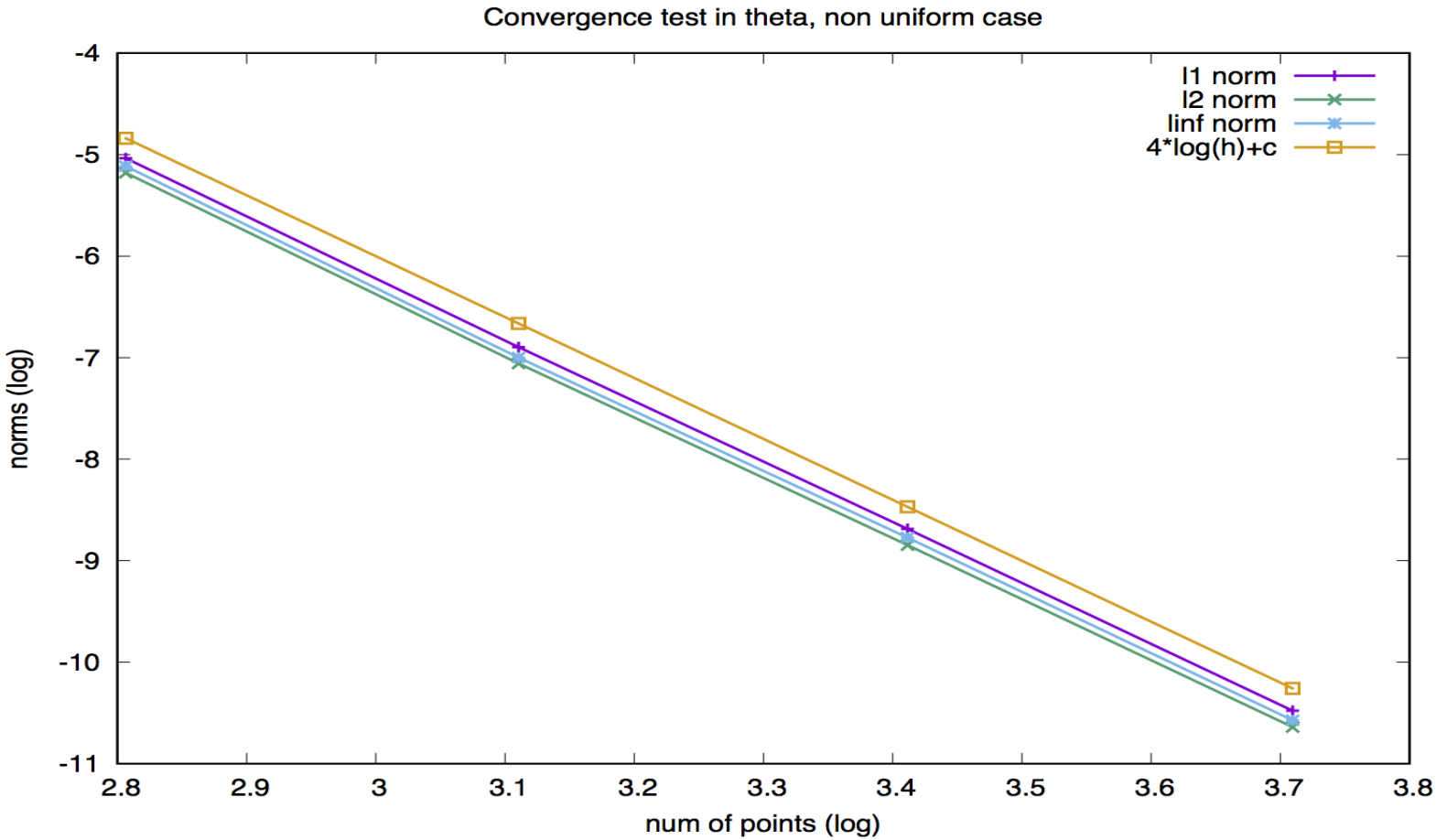
- $J_m\left(\frac{r \cdot j_{m,l}}{r_{max}}\right) exp(im\theta)$, with $m = 1$ as the Bessel function order and $j_{m,l}$ its l-th zero.



Analytical gyroaverage known:
$J_\rho(f_1)(r_0, \theta_0) = J_0\left(\frac{\rho \cdot j_{m,l}}{r_{max}}\right) f_1(r_0, \theta_0)$, with ρ as the gyroradius.

Convergence test

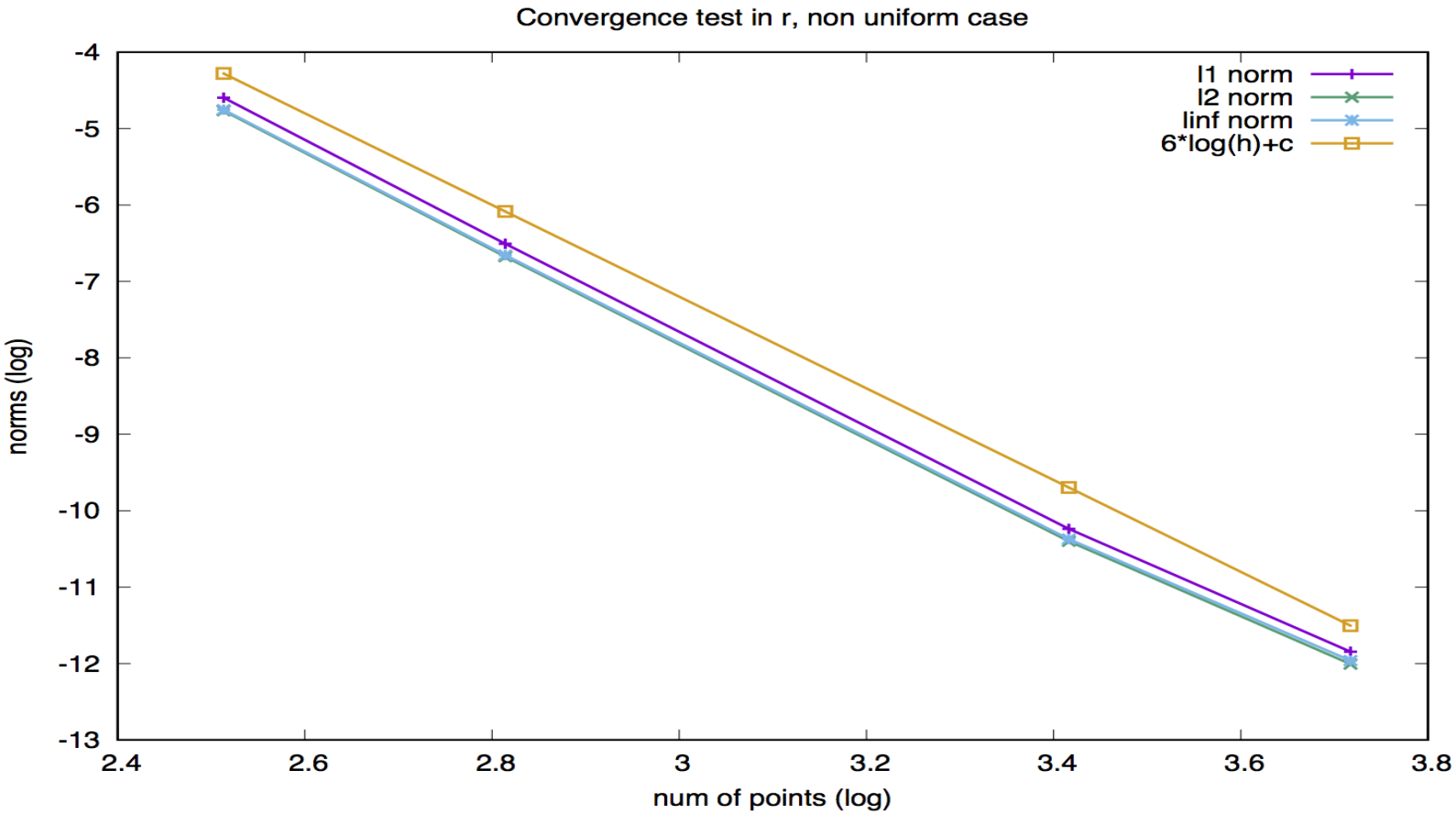
theta direction, non uniform grid



Lagrange degree = 5. Parameters of the Bessel function testcase: $\rho = 0.1$, $n_r = 256$, $n_{\text{gp}} = 128$.

Convergence test

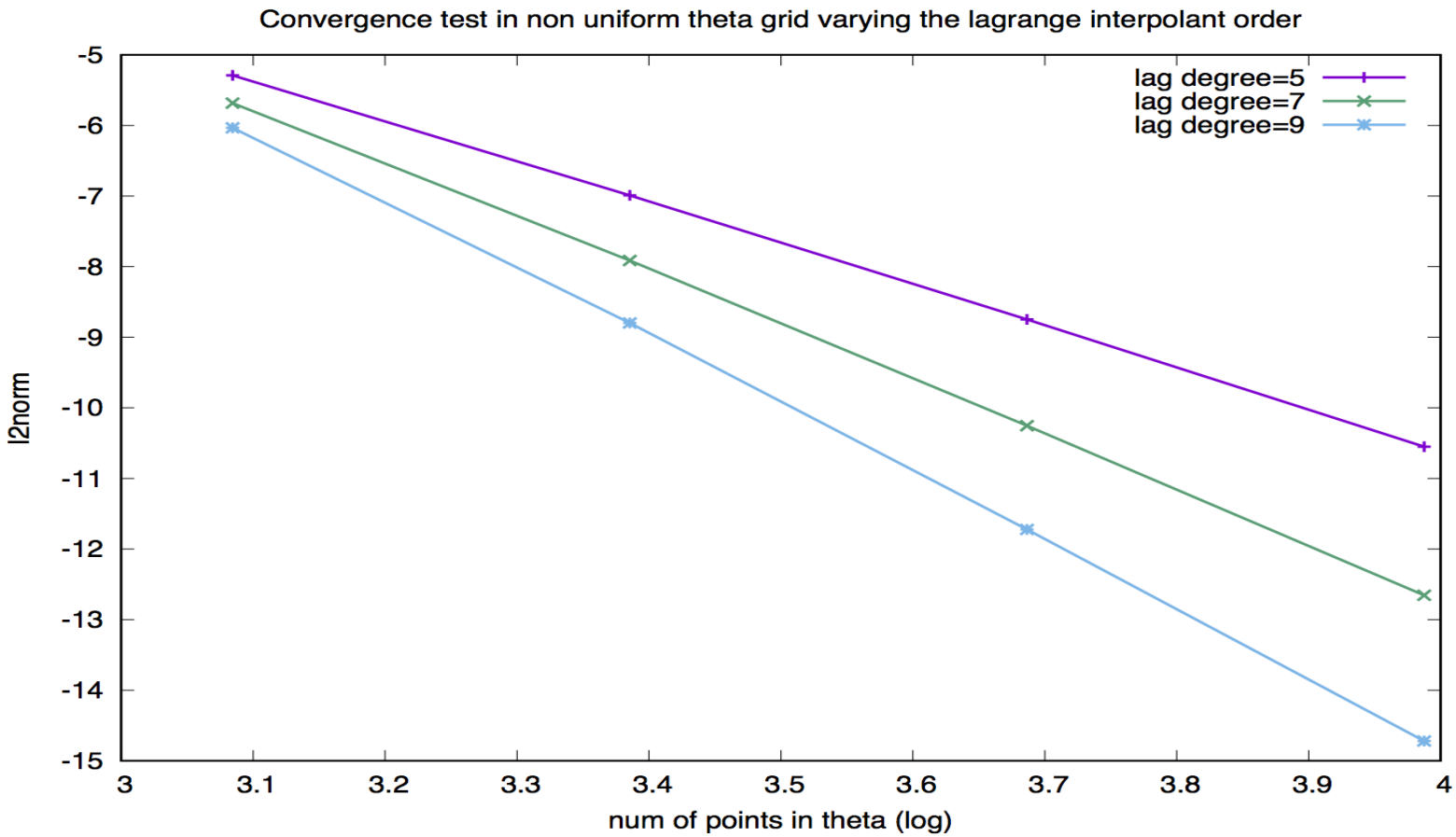
r direction, non uniform grid



Lagrange degree $= 5$. Parameters of the Bessel function testcase: $\rho = 0.1$, $n_\theta = 256$, $n_{\text{gp}} = 128$.

Convergence test

Lagrange order, non uniform grid



Parameters of the Bessel function testcase: $\rho = 0.1$, $n_r = 256$, $n_{\text{gp}} = 128$.

Conclusion, Perspectives

- DONE: convergence study for different operators
- Work in progress: cubic splines along theta direction
- Work in progress: detailed analysis for mapping (Miller, Culham)
- TODO: Poisson operator, porting to Gysela and Selalib

Everything worked fine !