# Compact AMR schemes for Conservation Laws

Erwan Deriaz

Équipe Plasmas Chauds

Institut Jean Lamour (Nancy), CNRS / Université de Lorraine

**Projet ANR Vlasix**

Avec: Nicolas Besse [1], Stéphane Colombi [2], Thierry Sousbie [2]

CANUM 2016

Obernai – 9-13 mai 2016

1. Observatoire de Nice
2. Institut d'Astrophysique de Paris

# Plan

## Vlasov-Poisson equations

Distribution function $f : \mathbb{R}^{2d+1} \to \mathbb{R}_+, (t, \mathbf{x}, \mathbf{v}) \mapsto f(t, \mathbf{x}, \mathbf{v})$

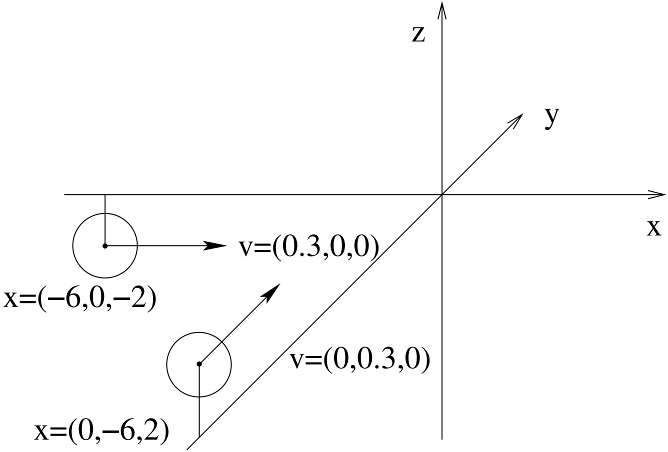
$$\partial_t f + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + F(t, \mathbf{x}) \cdot \nabla_{\mathbf{v}} f = 0 \tag{1}$$

with

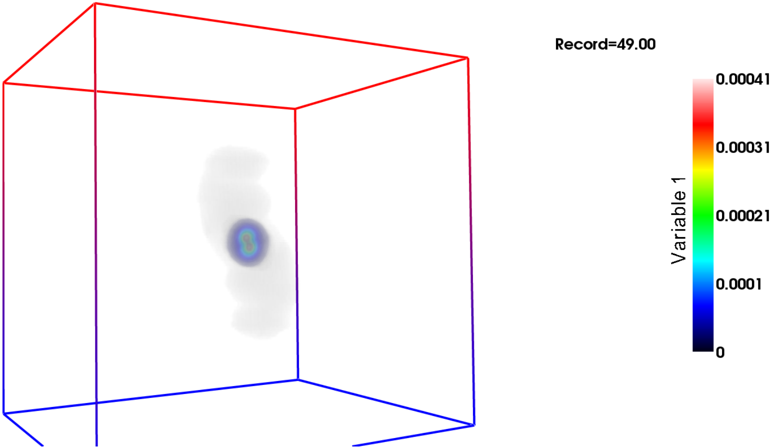
$$F(t, \mathbf{x}) = \pm \nabla_{\mathbf{x}} \phi(t, \mathbf{x}), \tag{2}$$

$$\Delta_{\mathbf{x}} \phi(t, \mathbf{x}) = \int_{\mathbf{v} \in \mathbb{R}^d} f(t, \mathbf{x}, \mathbf{v}) d\,\mathbf{v} - \int_{\mathbf{x}, \mathbf{v} \in \mathbb{R}^d} f(t, \mathbf{x}, \mathbf{v}) d\,\mathbf{v}\, d\mathbf{x}. \tag{3}$$

## Interaction of two Plummer models

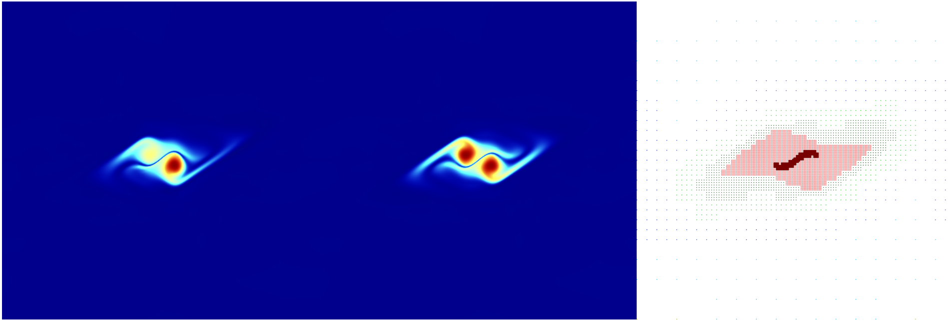


## Collision of two Plummer : 3D-3V, 3D-view



# Cut in phase space

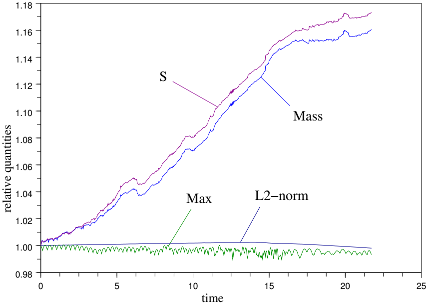
$512^6$ uniform grid equivalent accuracy

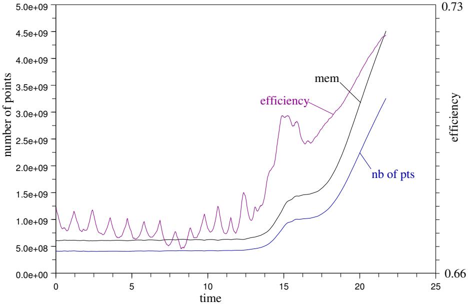


(z,w) at max | cut in (z,w) at zero | grid in (z,w) at zero

# Collision of two Plummer : 3D-3V

for $t \in [0, 21.7]$, number of time steps : 695
max number of points : 3,000,000,000 (on Curie supercomputer at IDRIS, Extra Large Node with 512 GB main memory).




conservations

point storage efficiency

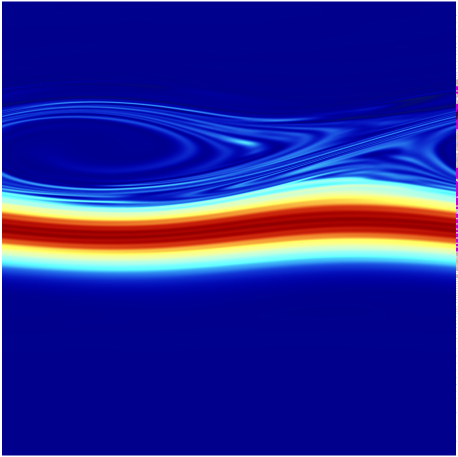
## Application to plasma physics : Bump-on-Tail in 1D-1V

Simulation box : $(x, v) \in [-\frac{10}{3}\pi, \frac{10}{3}\pi] \times [-10, 10]$

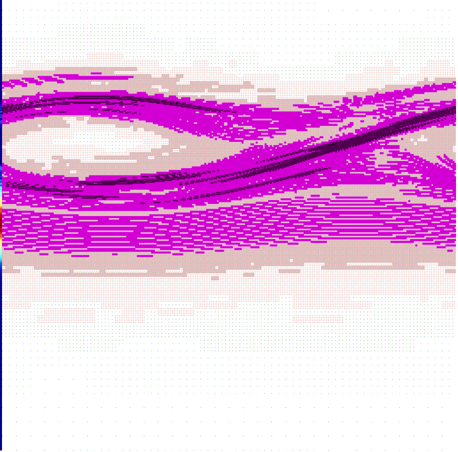
Initial condition Bump-on-Tail :

$$f_0(x, v) = \left( \frac{0.9}{\sqrt{2\pi}} e^{-\frac{v^2}{2}} + \frac{0.2}{\sqrt{2\pi}} e^{-4(v-4.5)^2} \right).$$

## 1D-1V



distribution function



grid

## 1D-1V

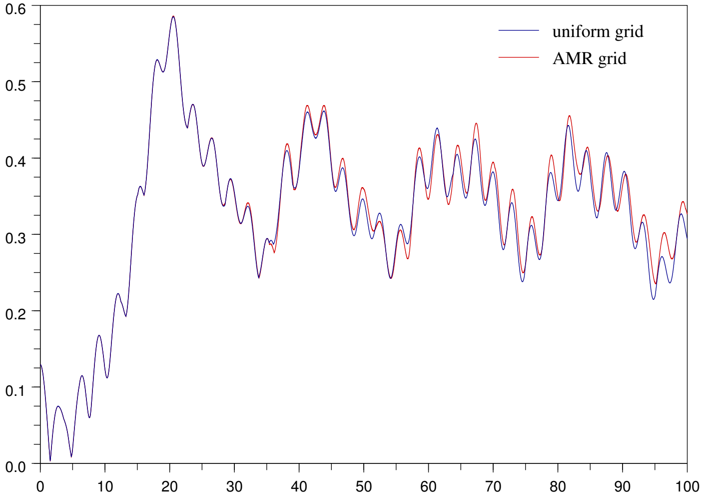



Figure: Plots of the maximum absolute value of the field $E$ for two instances of the bump-on-tail instability : with an uniform grid and with an AMR grid.

# 1D-1V

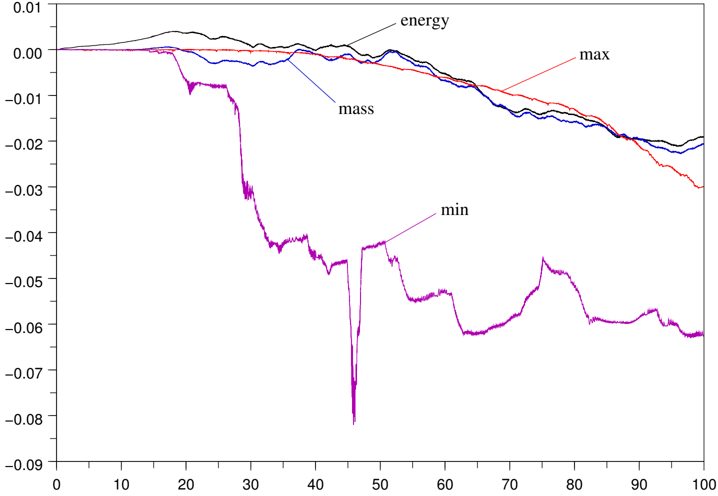



Figure: Relative variations ($\Delta f / f$) for the mass, the total energy and the maximum value of the distribution function. These should remain constant. For visualization purpose, the mass variation was multiplied by ten.

## 1D-1V

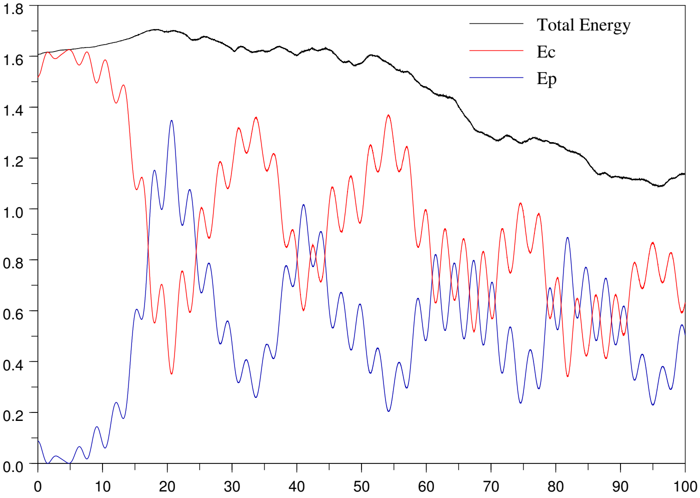


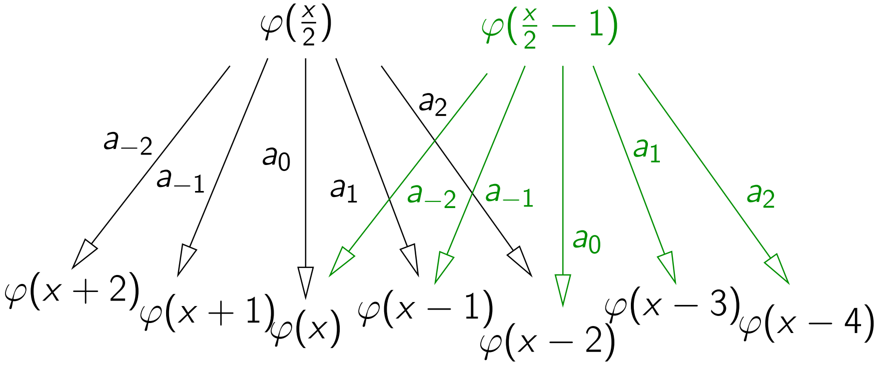

Figure: Variation of the kinetic (Ec) and potential (Ep) energies. The kinetic energy was vertically shifted by $-23$.

## Refinement filter

each refinement scheme issues a scaling function

$$\varphi\left(\frac{x}{2}\right) = \sum_{k\in\mathbb{Z}} a_k \varphi(x-k)$$

with $\sum_{k\in\mathbb{Z}} a_k = 2$.



## Gradation and compacity

- Key element of the refinement algorithms : an element can become active only if its parents (the elements from which it is interpolated) are already active. Hence the notion of **gradation**.
- The more non zero $a_k$ the larger the **gradation margin**, the needed memory and the complexity. Critical in many dimensions.
- It is possible to apply finite difference schemes to elements of the same level (identical elements).
- **Interest of the interpolet** scaling functions : many zero coefficients, easy to pass an element from a level to an other level since it corresponds to a point value.

## Special filters

- **Finite volume** elements

$$\overline{\varphi}\left(\frac{x}{2}\right) = \sum_{k\in\mathbb{Z}} b_k \overline{\varphi}(x-k)$$

  with $\forall k \neq 0,\ b_{2k} + b_{2k+1} = 0$, and $b_0 = b_1 = 1$ if symmetry

- **Interpolet** scaling functions

$$\dot{\varphi}\left(\frac{x}{2}\right) = \sum_{k\in\mathbb{Z}} a_k \dot{\varphi}(x-k)$$

  with $a_0 = 1$ and $\forall k \neq 0,\ a_{2k} = 0$ (lots of gaps)

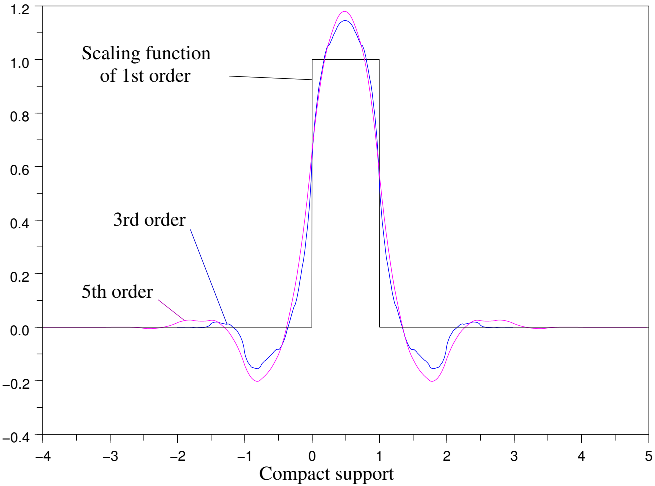
Any finite volume scaling function derives from an interpolet scaling function :

$$\dot{\varphi}'(x) = \overline{\varphi}(x) - \overline{\varphi}(x-1)$$

$$\forall k \quad a_k = \frac{b_k + b_{k+1}}{2}$$

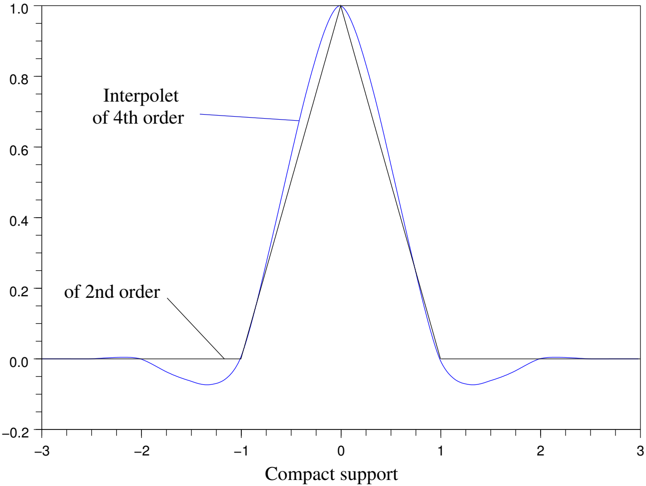
Interpolets are much smoother than finite volume scaling functions.

## Finite volume scaling functions



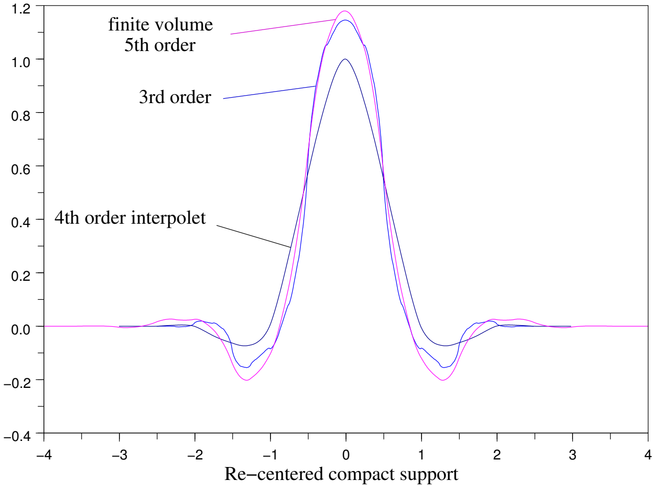

Scaling functions $\varphi$ corresponding to the finite volume schemes.

## Interpolets : finite difference scaling functions




Scaling functions $\varphi$ corresponding to the finite difference schemes.

## Comparison between these two types




Scaling functions $\varphi$ recentered for comparison.

## Mass calculation in AMR

Two instances when the mass $M = \int_\Omega u(x)\, dx$ is affected by the AMR scheme :

- when the grid changes : refined or **coarsed**.
- when solving the conservation law

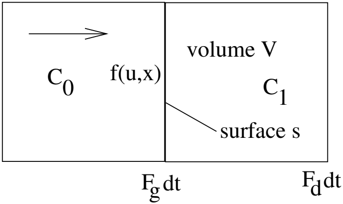
$$\partial_t u + \nabla \cdot f(u, x) = 0$$

  in the non uniform grid : $u_n \to u_{n+1}$.

In the case of the finite volumes, the volume of an element depends exclusively from its level and from the fact of being a leaf of not.
In the other cases, it depends on which descendents are activated.
Applying a wavelet transform concentrates all the mass on the coarsest level. It allows to modify the grid safely.

## Advection and fluxes

For the finite volumes, we have to compute the fluxes $F = f(u,x)$ along the the surface $S$ in the conservation law $\partial_t + \nabla \cdot f(u,x) = 0$.




then

$$du = \frac{F_g - F_d}{V} s \, dt$$

Interest : as the $F_g$ of $C_1$ equals the $F_d$ of $C_0$, the mass is strictly conserved.

# Making the scheme conservative

Let $(x_i, \ell_i)_i$ be a set of elements (points,weights) containing the information $(u_i)$ subject to a conservative equation $\partial_t u - \partial_x u = 0$.
We approximate the term $\partial_x u$ by a finite difference formula at order $p_i$ :

$$(\partial_x u)_i \sim \delta u_i = \frac{1}{\ell_i} \sum_j \alpha_{ij} u_j.$$

We can compute the flux going outside the element $(x_j, \ell_j)$ :

$$F_j = \left( \sum_i \alpha_{ij} \right) u_j.$$

These fluxes should be zero.
If it is not the case we substract them introducing correcting terms in some of the $(\alpha_{ij})$ stencils :

$$(\partial_x u)_{i_0} \sim \delta u_{i_0} = \frac{1}{\ell_{i_0}} \left( \sum_j \alpha_{i_0 j} u_j - \sum_j F_j \right) = \frac{1}{\ell_{i_0}} \sum_j \left( \alpha_{i_0 j} - \sum_i \alpha_{ij} \right) u_j.$$

## Points with volumes

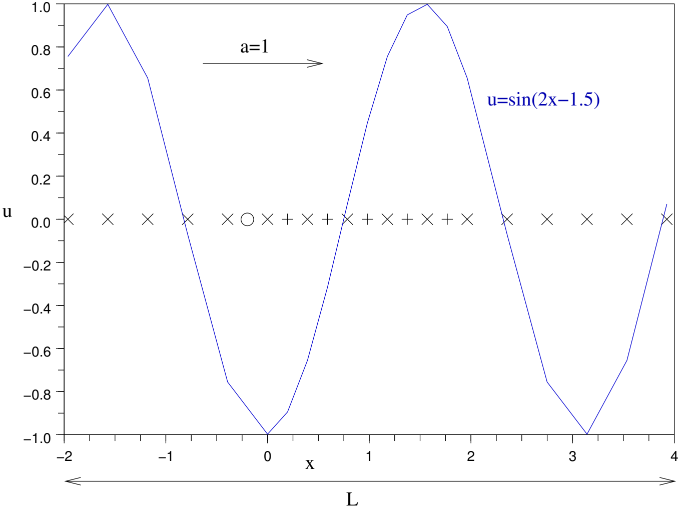
Using the refinement scheme for 4th order interpolet we derive the following ‘volumes’ for the points

$$2 \quad 2 \quad 2 \quad \frac{33}{16} \quad \frac{3}{2} \; 1 \; \frac{15}{16} \; 1 \; 1 \; 1 \; 1 \; 1 \; \frac{15}{16} \; 1 \; \frac{3}{2} \quad \frac{33}{16} \quad 2 \quad 2 \quad 2$$

$$\times \quad \times \quad \times \quad \times \; \bigcirc \; \times + \times + \times + \times + \times + \times \quad \times \quad \times \quad \times \quad \times$$

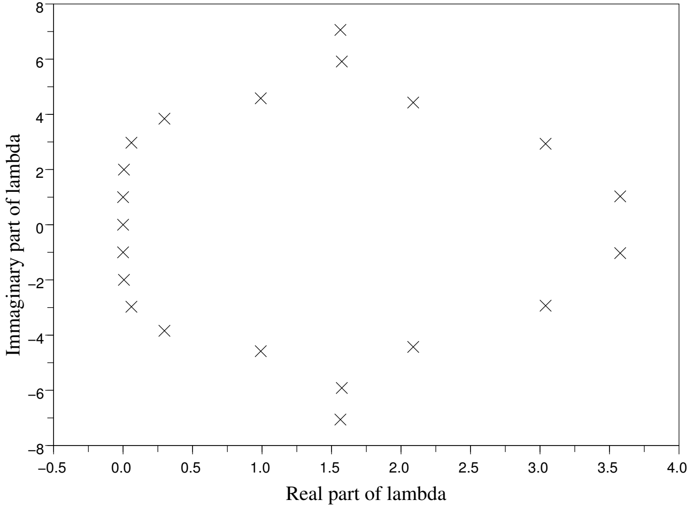
$$2 \quad 2 \quad \frac{33}{16} \quad \frac{3}{2} \; 1 \; \frac{15}{16} \; 1 \; 1 \; 1 \; 1 \; 1 \; 1 \; 1 \; \frac{15}{16} \; 1 \; \frac{3}{2} \quad \frac{33}{16} \quad 2 \quad 2 \quad 2$$

## Tranport in 1D

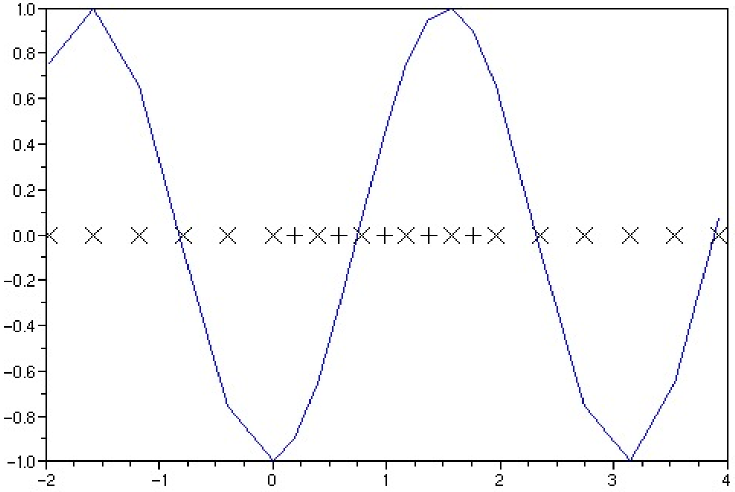



$\partial_t u + a\partial_x u = 0$ for $t \in [0, T]$ with $T = 4\,L$, on 21 points.

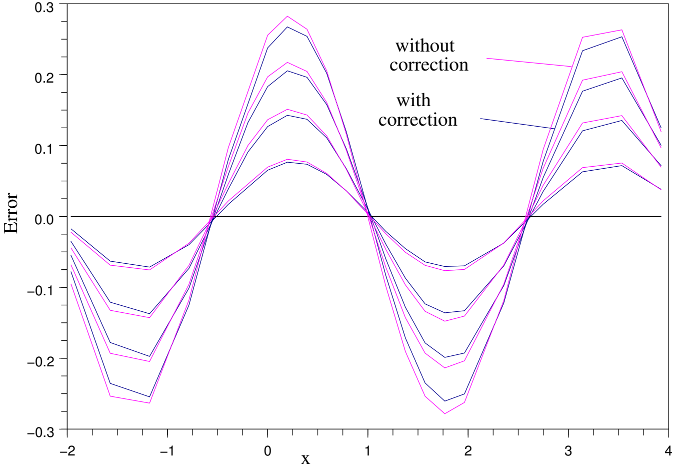
## Spectrum of the discrete operator : eighenvalues $\lambda$




$A\,u$ is our discrete approximation of $\partial_x u$, $\lambda \in \mathbb{C}$ are the eighenvalues of $A$.

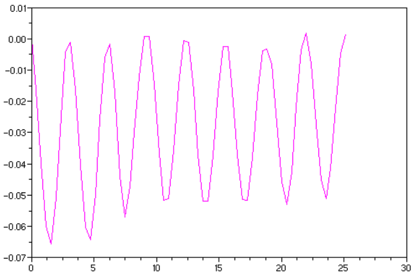

1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
−2
−1
0
1
2
3
4

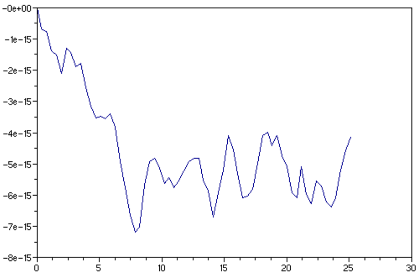
## No change in the error




Error at four instances: L, 2L, 3L and 4L

## Total mass variation during time




Mass without correction (left) and with correction (right)
The mass conservation is ok now

# Conclusion–Perspectives

Conclusion :

- application of finite volume principles in interpolant AMR,
- application of wavelet constructions through the considerations on scaling functions,
- 6D simulations demand a lot of memory, we pass from 100,000 to 300,000 the number of points necessary for a local refinement.

Perspectives :

- finish to implement the 6D code with these improvements,
- as soon as the numerical scheme is validated, implement a MPI parallelisation,
- test other schemes, lagrangian, Galekin discontinuous, *cf* Eric Madaule's PhD.