Semi-Lagrangian particles
or
6D Vlasov on single core

G.-H. Cottet
Univ. Grenoble-Alpes

LABORATOIRE
JEAN KUNTZMANN
MATHÉMATIQUES APPLIQUÉES - INFORMATIQUE

A brief and biased history of particle methods for Vlasov-Poisson and flow simulations

	Numerics	Vlasov	Incompressible flows	Compressible flows
80's		Numerical analysis (PIC, VIC, ..)		Design of SPH
	Numerical issues Noise / accuracy	Random init $n_{part} >> N_{cell}$	Location processing technique	Renormalization $h << \varepsilon$
	Field solve	Grid/FFT	Grid-free/Biot-Savart Fast N-body solvers	no field
90's	*Accuracy*		Particle remeshing	
2000	*Field solve*		FFT based	
2005-	*Accuracy*		-High order remeshing -Directional splitting -S.L particles -Multi-resolution particles -GPU implementation	

LABORATOIRE JEAN KUNTZMANN
MATHÉMATIQUES APPLIQUÉES - INFORMATIQUE

Vlasov-Maxwell equations

Distribution function for one specie of ions (or electrons) subject to electric and magnetic fields satisfy

$$f = f(\mathbf{x}, \mathbf{v}, \mathbf{t}) \in [\mathbf{0}, \mathbf{1}] \quad \mathbf{E} = \mathbf{E}(\mathbf{x}, \mathbf{t}) \quad \mathbf{B} = \mathbf{B}(\mathbf{x}, \mathbf{t})$$

Conservation of charge:

$$\frac{\partial f}{\partial t} + (\mathbf{v} \cdot \nabla_{\mathbf{x}})\mathbf{f} + ((\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_{\mathbf{v}})\, \mathbf{f} = \mathbf{0}$$

+ Maxwell equations coupling E and B to moments of f (density and charge):

$$\rho(\mathbf{x}, t) = q \int f(\mathbf{x}, \mathbf{v}, t)\, dv$$

$$i(\mathbf{x}, t) = q \int \mathbf{v}\, f(\mathbf{x}, \mathbf{v}, t)\, dv$$

Transport equation in phase space (x,v) of dimension up to 6

Advection field given by

$$\mathbf{U} = \begin{bmatrix} \mathbf{v} \\ E + \mathbf{v} \times \mathbf{B} \end{bmatrix}$$

satisfies

$$\mathrm{div}_{\mathbf{x},\mathbf{v}} \mathbf{U} = \mathbf{0}$$

conservative advection equation for f with velocity field U

-> conservation of all L^p norms of f

Computational complexity lies in

- space dimension (up to 6)
- transport equation where one wishes to conserve physical invariants and $f \in [0,1]$

➡ **however support of f occupies in general small part of phase space**

These features make natural the use of **Lagrangian particle methods** :

- replace f by macro-particles in the phase space
- follow them with local velocities
- compute E, B fields in self-consistent way with integral formulas or FFT-based grid solvers

Advantages and drawback of Lagrangian (grid-free) particle methods

Pros:

- calculations restricted to the support of f
- norms of distribution function f, bounds of f and entropy ok
- large time-steps (see later)

Cons :

convergence analysis (and practice) shows that it is important to assure $\Delta x \ll \varepsilon$

(or $n_{part} >> N_{cells}$)

➔ need many particles to limit numerical noise in field evaluation

➔ expensive

Can Eulerian methods provide reasonable alternative in d>1 ?

Yes, if used in multi-resolution modes

Recent work by Deriaz and Periani (SIAM MMS 2018) : multi-resolution method based on wavelet analysis and third order finite-difference methods (despite lack of conservativity, numerical diffusion, and CFL conditions)

Enables simulations of 6D gravitational systems with acceptable memory and CPU times requirements

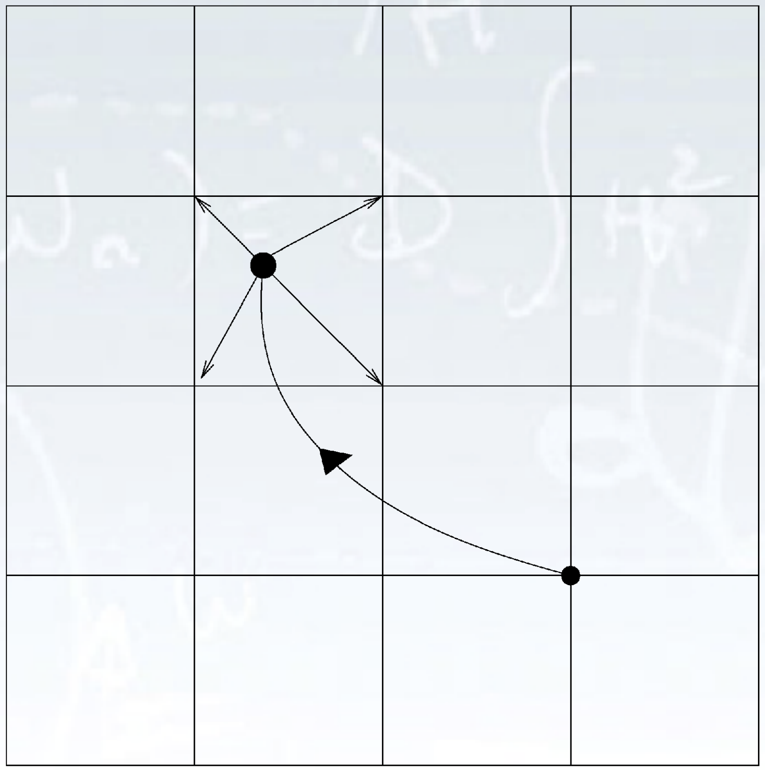
Suffers difficulties related to finite-difference solvers (in particular numerical diffusion)

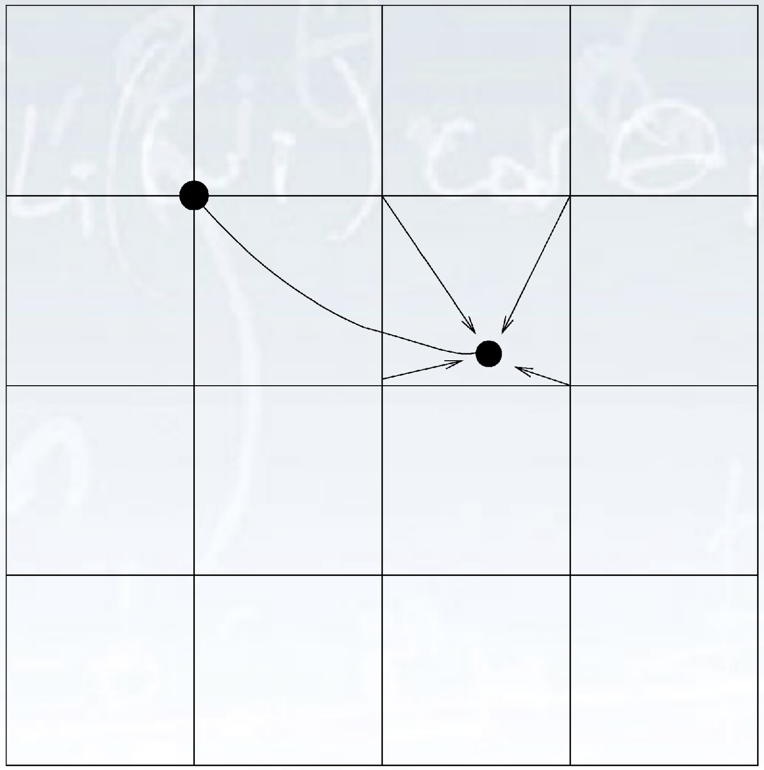
Is there room for methods in-between grid-free particle methods and eulerian methods with (bonus) multi-resolution capabilities ?

Semi-Lagrangian methods are good candidates :

- they are well-adapted to advection-dominated problems (low numerical diffusion, even for low order methods)
- not constrained by CFL conditions

Classical semi-Lagrangian methods for transport and Vlasov (Cruseilles et al 2009, Sonnendrucker et al 2010 ..):





Forward Semi-Lagrangian :
push trajectories,
deposit mass though interpolation
reconstruct grid values through B-splines

Backward Semi-Lagrangian :
go backward on trajectories,
interpolate from grid values

$$f(t,x,y) = \sum_{k,l} \omega_{k,l}^n S\big(x - X_1(t; x_k, y_l, t^n)\big) S\big(y - X_2(t; x_k, y_l, t^n)\big)$$

LABORATOIRE JEAN KUNTZMANN
MATHÉMATIQUES APPLIQUÉES - INFORMATIQUE

In (2D) FSL methods, to advance from time step t^n to t^{n+1}, the solution is represented on a B-spline basis :

$$f(t,x,y) = \sum_{k,l} \omega^n_{k,l} S\big(x - X_1\big(t; x_k, y_l, t^n\big)\big) S\big(y - X_2\big(t; x_k, y_l, t^n\big)\big)$$

and weights ω^{n+1} at time t^{n+1} are recovered by solving linear system :

$$f^{n+1}_{i,j} = \sum_{k,l} \omega^{n+1}_{k,l} S(x_i - x_k) S(y_j - y_l).$$

Conservative methods can be constructed along similar lines

Drawbacks:

- cost (linear system)
- memory requirements (like Eulerian methods, need full grids)
- high order ?

So far, used mostly (only ?) for low-dimensional (1+1D) systems

LABORATOIRE JEAN KUNTZMANN
MATHÉMATIQUES APPLIQUÉES - INFORMATIQUE

Other approach to Semi-Lagrangian methods: remeshed particle methods

- try to combine natural adaptivity of particles with accuracy of Semi-Lagrangian methods
- well established for 3D level-set methods and complex flow calculations but (almost) never used so far for Vlasov-Poisson

Principle:

- initialize particles in support of f and push them with local velocities (like regular particles)
- remesh **at each time-step** with high order interpolation formulas
- retain after remeshing only particles with strength above a given cut-off

LABORATOIRE JEAN KUNTZMANN
MATHÉMATIQUES APPLIQUÉES - INFORMATIQUE

Remeshed particle methods

Idea goes back to the 80's:
Krasny's 2D vortex sheet, Meiburg's 3D jets, and Chorin's and Leonard's hairpin removal
Insert fresh particles «in between» old particles when needed
Specific to problems with topology control

More generic approach : remesh particles on regular grids through standard 3D **interpolation** formulas.

Criterium for remeshing schemes :
conservation of the moments of the particle distribution: $\int f\,d\mathbf{x}\;,\;\int \mathbf{x}\,f\,d\mathbf{x}\;,\;\int \mathbf{x}^2\,f\,d\mathbf{x}\;\cdots$

Allowed first high resolution DNS of flow past cylinders at high Reynolds numbers (Koumoutsakos & Leonard, JFM 1995), … before spectral element calculations

LABORATOIRE JEAN KUNTZMANN
MATHÉMATIQUES APPLIQUÉES - INFORMATIQUE

Traditionally, work with tensor products of 1D formulas

Typical interpolation formulas :

•conservation of 3 moments (third order truncation error) use 3 points in each direction
smooth version uses an additional grid point -> 4 grid points

•conservation of 5 moments (5th order truncation error) use 5 points
smooth version spread particle on 6 nearest grid points

•resulting stencils in 3D : 27, 64, 125, 216 points

if advection of particles is split direction by direction, reduces to one-dimensional stencils

Until recently particle remeshing was considered as an ad-hoc fix, and momentum conservation properties as safeguard

Particle methods with remeshing at every time-step can be viewed and analyzed as **forward semi-lagrangian** methods (C. et al, M2AN 2014)

How they work:

1) particles on a grid
2) push particles with local velocity values
3) remesh particles on the grid, through interpolation

In 1 D for advection equation $\theta_t + u\,\theta_x = 0$

method can be described by the following equations:

$$x_i^{n+1} = x_i + \tilde{u}_i^n \Delta t \qquad \theta_i^{n+1} = \sum_j \theta_j^n \, \Gamma\left(\frac{x_j^{n+1} - x_i}{\Delta x}\right), i \in \mathbb{Z}^d$$

where $\tilde{u}_i^n$ depend on the time-stepping scheme and Γ is a piecewise polynomial kernel

LABORATOIRE JEAN KUNTZMANN
MATHÉMATIQUES APPLIQUÉES - INFORMATIQUE

Γ is defined by regularity and moment properties :

★ moment properties : $\sum_{k\in\mathbb{Z}} k^\alpha \Gamma(x-k) = x^\alpha,\ 0 \le \alpha \le p,\ x \in \mathbb{R}$

★ regularity : Γ is of class C^r and $\Gamma \in \mathcal{C}^\infty(]l, l+1[),\ l \in \mathbb{Z}$

★ **interpolation property :** $\Gamma(i) = \begin{cases} 1 \text{ if } i = 0, \\ 0 \text{ otherwise.} \end{cases}$

Convergence result (Cottet et al, M2AN 2014) **:**
1) the spatial order of the method is *inf(p,r)*
2) stability holds for a large class of kernels under the condition $\Delta t \le \|\vec{\nabla}\vec{u}\|_\infty^{-1}$

Remark : $\Delta t \le \|\vec{\nabla}\vec{u}\|_\infty^{-1}$ is sometimes called a Lagrangian CFL condition (LCFL) with LCFL 1

If at all times, each cell contains exactly one particle, order = p

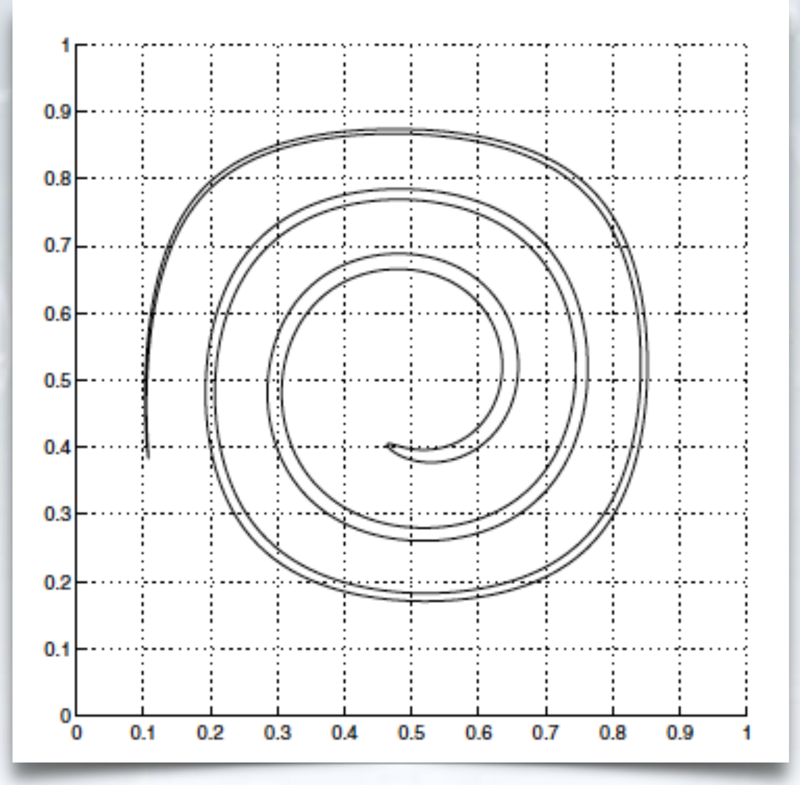
Examples of remeshing kernels (2nd and 6th order)

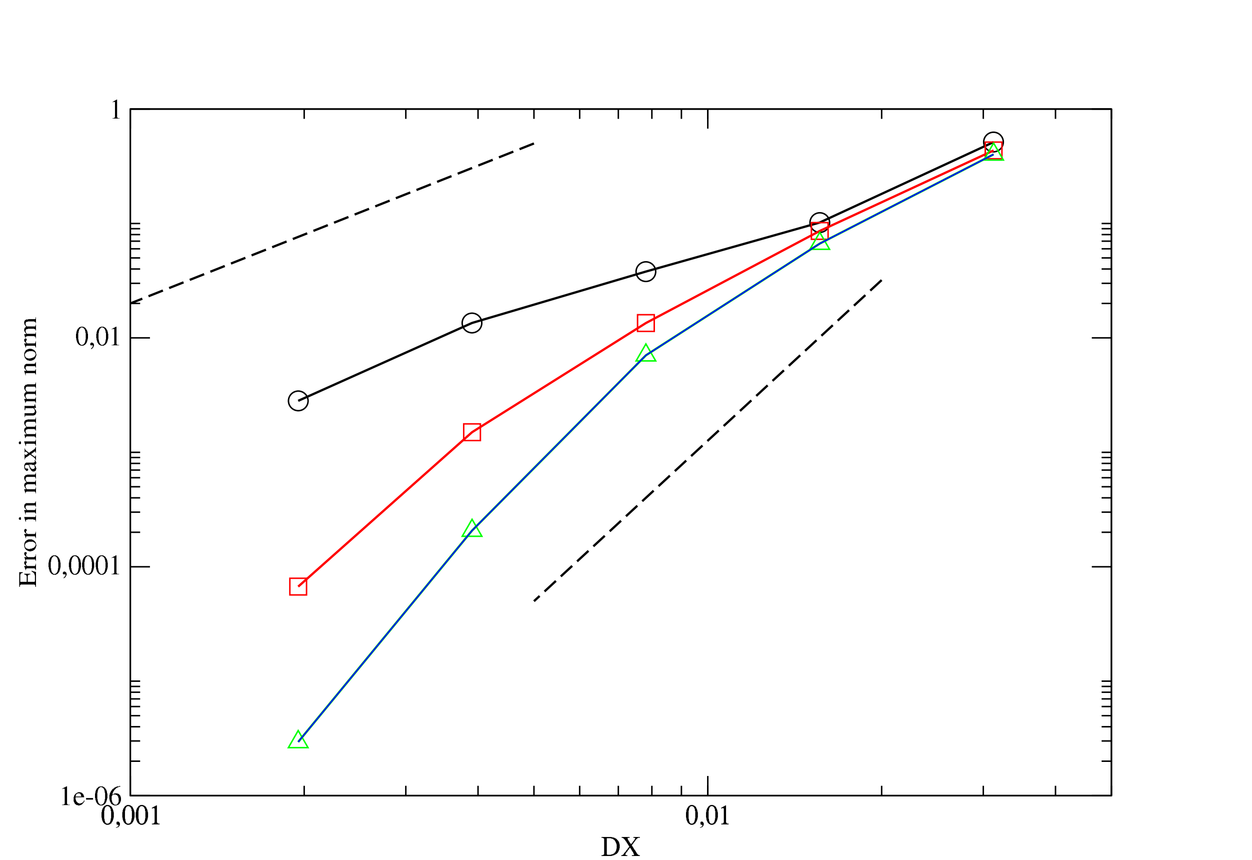
$$\Lambda_{4,2}(x)=\begin{cases}1-\frac{5}{4}|x|^2-\frac{35}{12}|x|^3+\frac{21}{4}|x|^4-\frac{25}{12}|x|^5 & 0\leqslant|x|<1\\ -4+\frac{75}{4}|x|-\frac{245}{8}|x|^2+\frac{545}{24}|x|^3-\frac{63}{8}|x|^4+\frac{25}{24}|x|^5 & 1\leqslant|x|<2\\ 18-\frac{153}{4}|x|+\frac{255}{8}|x|^2-\frac{313}{24}|x|^3+\frac{21}{8}|x|^4-\frac{5}{24}|x|^5 & 2\leqslant|x|<3\\ 0 & 3\leq|x|\end{cases}$$

$$\Lambda_{6,6}(x)=\begin{cases}1-\frac{49}{36}|x|^2+\frac{7}{18}|x|^4-\frac{1}{36}|x|^6-\frac{46109}{144}|x|^7+\frac{81361}{48}|x|^8-\frac{544705}{144}|x|^9+\frac{655039}{144}|x|^{10}\\ \qquad -\frac{223531}{72}|x|^{11}+\frac{81991}{72}|x|^{12}-\frac{6307}{36}|x|^{13}, & 0\leqslant|x|<1\\ -\frac{44291}{5}+\frac{1745121}{20}|x|-\frac{15711339}{40}|x|^2+\frac{32087377}{30}|x|^3-\frac{7860503}{4}|x|^4+\frac{38576524}{15}|x|^5\\ \qquad -\frac{24659323}{10}|x|^6+\frac{84181657}{48}|x|^7-\frac{74009313}{80}|x|^8+\frac{17159513}{48}|x|^9\\ \qquad -\frac{7870247}{80}|x|^{10}+\frac{438263}{24}|x|^{11}-\frac{81991}{40}|x|^{12}+\frac{6307}{60}|x|^{13}, & 1\leqslant|x|<2\\ 3905497-\frac{424679647}{20}|x|+\frac{3822627865}{72}|x|^2-\frac{2424839767}{30}|x|^3+\frac{3009271097}{36}|x|^4\\ \qquad -\frac{930168127}{15}|x|^5+\frac{305535494}{9}|x|^6-\frac{9998313437}{720}|x|^7+\frac{203720335}{48}|x|^8-\frac{137843153}{144}|x|^9\\ \qquad +\frac{22300663}{144}|x|^{10}-\frac{6126883}{360}|x|^{11}+\frac{81991}{72}|x|^{12}-\frac{6307}{180}|x|^{13}, & 2\leqslant|x|<3\\ -\frac{255622144}{5}+\frac{971097344}{5}|x|-\frac{15295867328}{45}|x|^2+\frac{5442932656}{15}|x|^3-\frac{2372571796}{9}|x|^4\\ \qquad +\frac{2064517469}{15}|x|^5-\frac{9563054381}{180}|x|^6+\frac{2210666335}{144}|x|^7-\frac{796980541}{240}|x|^8\\ \qquad +\frac{76474979}{144}|x|^9-\frac{43946287}{720}|x|^{10}+\frac{343721}{72}|x|^{11}-\frac{81991}{360}|x|^{12}+\frac{901}{180}|x|^{13} & 3\leqslant|x|<4\\ 0, & 4\leq|x|\end{cases}$$

Refinement study : case of a rotating patch in an off-center vorticity field

$$\boldsymbol{u}(x,t) = 2\,\cos(\pi\,t/T)\begin{pmatrix} -\sin^2(\pi\,x)\,\sin(\pi\,y)\,\cos(\pi\,y) \\ \sin^2(\pi\,y)\,\sin(\pi\,x)\,\cos(\pi\,x) \end{pmatrix}.$$





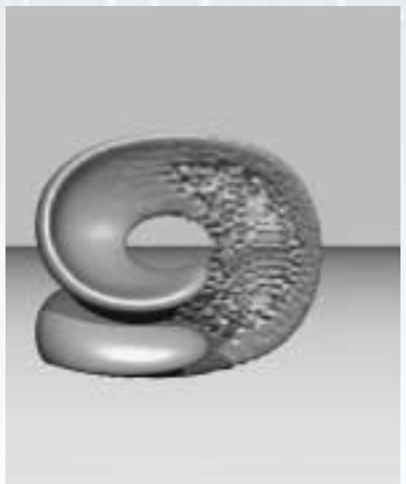
Kernel	Order of convergence
$\Lambda_{2,1}$	1.87
$\Lambda_{4,2}$	3.17
$\Lambda_{6,4}$	5.92

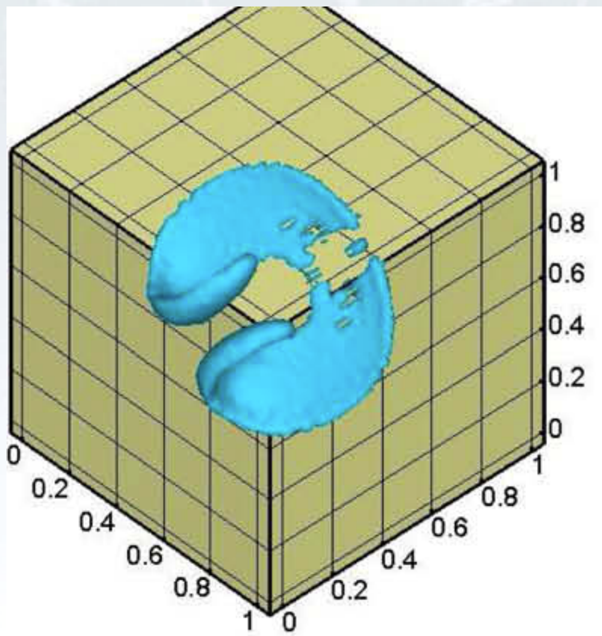
Error in maximum norm for different kernels (order 1_2, 2_4 and 4_6)

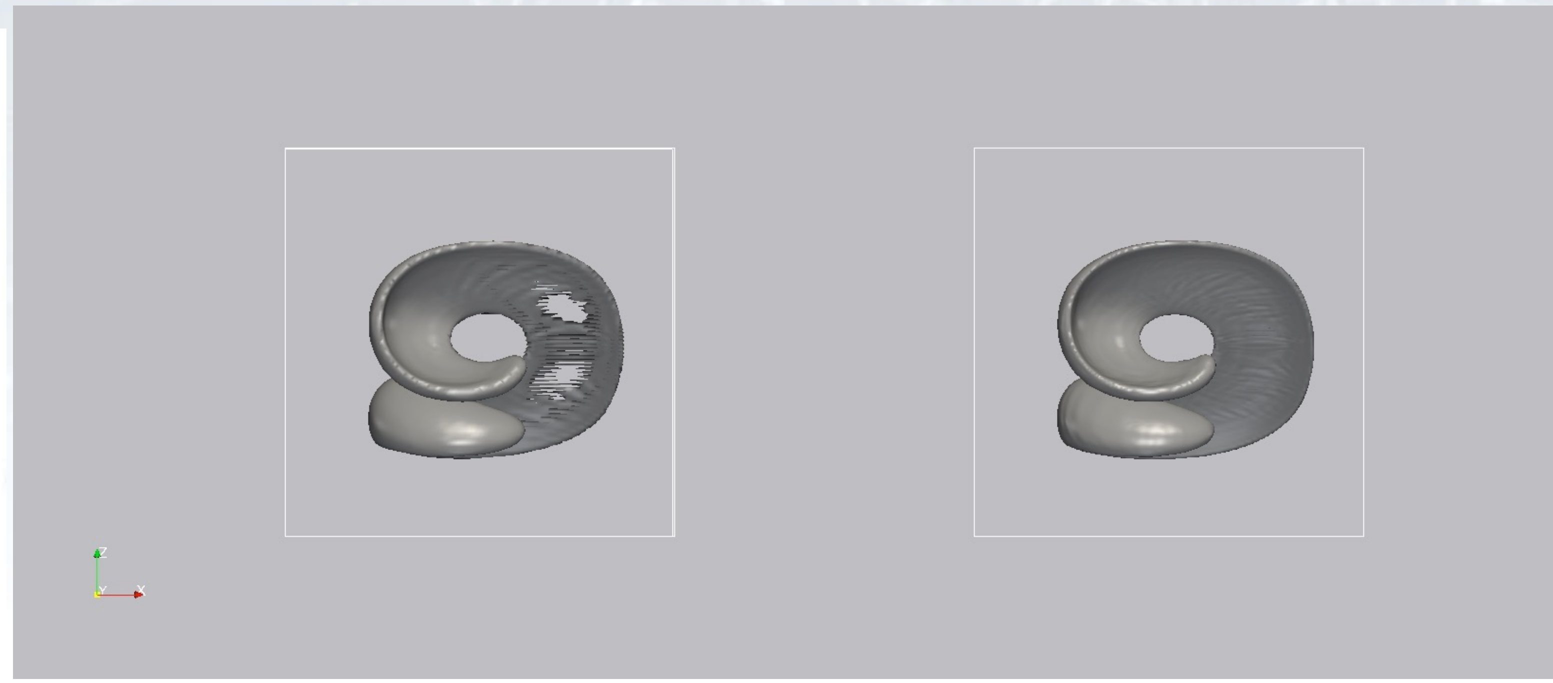
LABORATOIRE JEAN KUNTZMANN
MATHÉMATIQUES APPLIQUÉES - INFORMATIQUE

3D case : comparisons with Weno and VOF methods

Implementation of grid-based methods
with particles for corrections







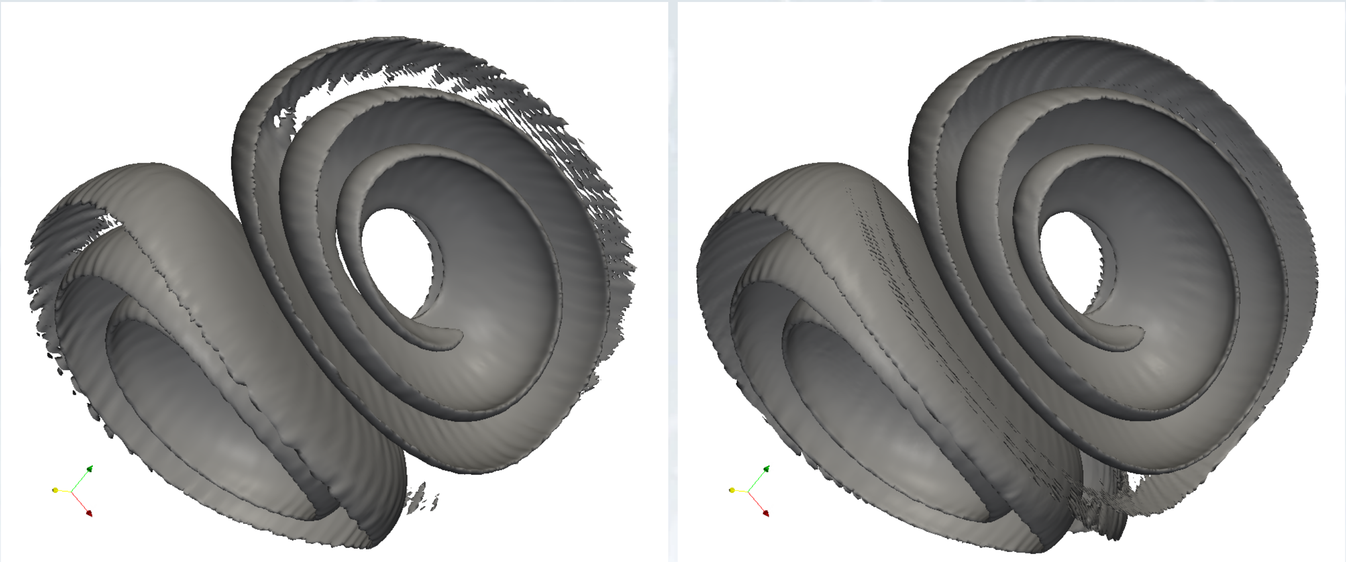
Enright et al, JCP 2002
3r order Weno
N=100 + 64 ppc
CFL=1 (?)
CPU =??

Vincent et al, JCP 2010
VOF
N=64 + 9 ppc
CFL=0.1

N=100, CFL=8

N=160, CFL=12
CPU time :
1 s per iteration

remeshed particle method, 4th order
remeshing,
2nd order in time



$N = 256$ and $CFL = 30$. Left picture: kernel $\Lambda_{2,1}$; right picture: kernel $\Lambda_{8,4}$.

LABORATOIRE JEAN KUNTZMANN
MATHÉMATIQUES APPLIQUÉES - INFORMATIQUE

Semi-Lagrangian (or remeshed) particle methods for Vlasov-Poisson

First attempt: Myers, Colella, Van Straalen (SIAM J. Sci. Comput., 2017) :

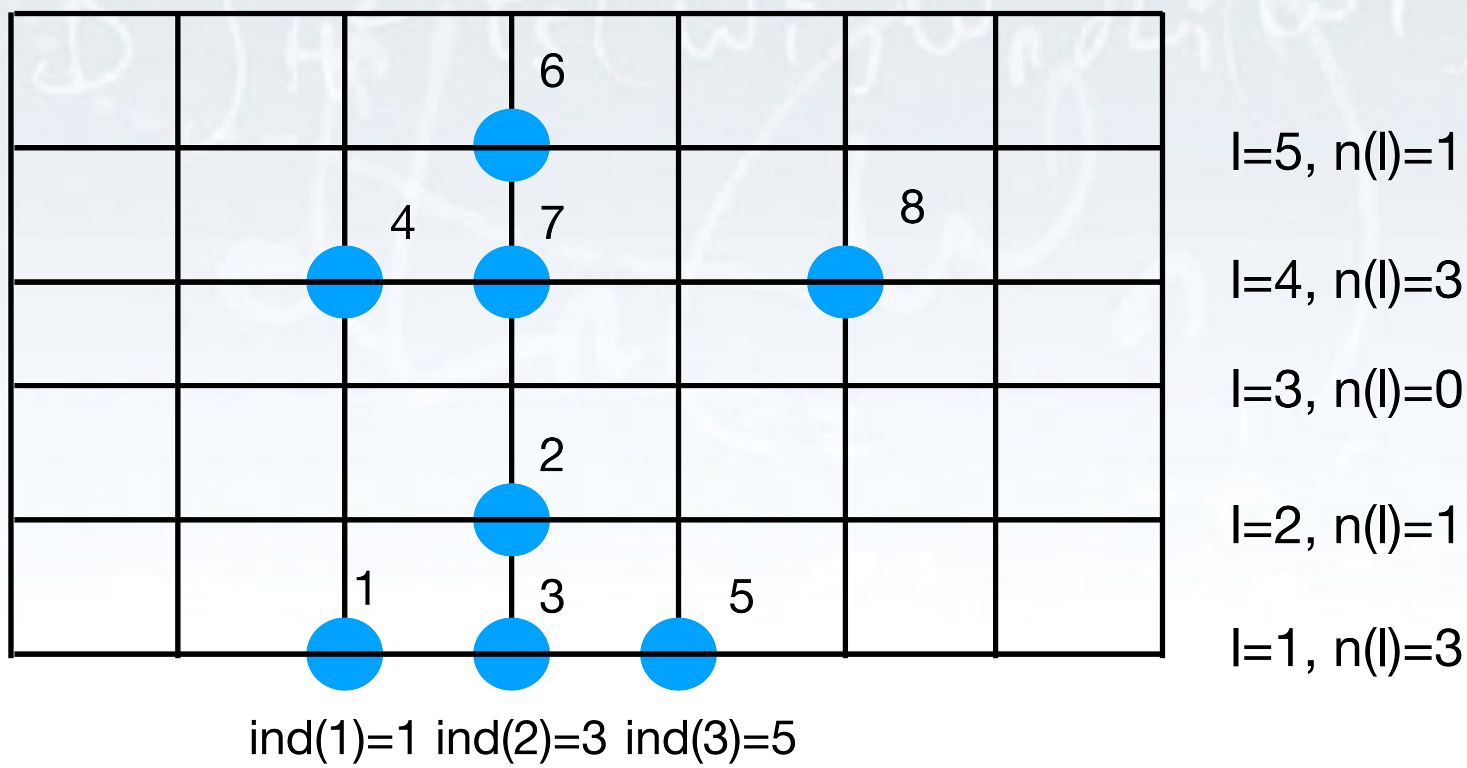
- Successful application to 2+2D Landau damping
- Use of 2nd and 4th order kernels
- **roadblock for higher dimension : need a full grid to remesh particles**
- **still n>N, and remeshing frequency to adjust**

roadblock can be removed by using link-lists of particles in lower dimensional spaces (C., JCP 2018)

LABORATOIRE JEAN KUNTZMANN
MATHÉMATIQUES APPLIQUÉES - INFORMATIQUE

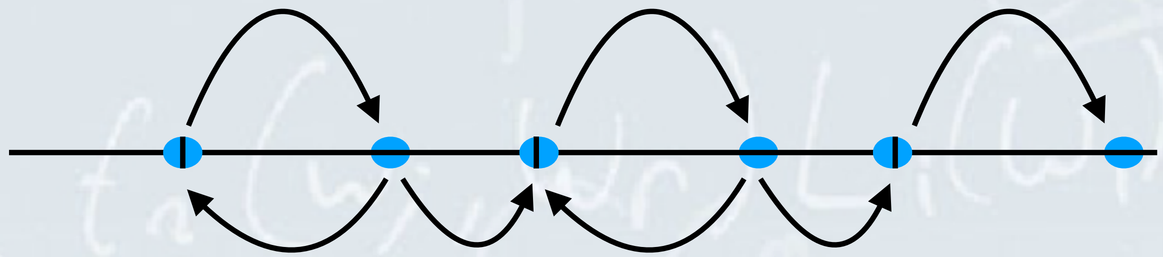
2D example : particles in (x,y) space, directional splitting of advection

1st sweep : horizontal advection



sort particles by horizontal lines :
each particle on the line gets an address in the original list

line l=1



-> i=1:n(1)=3,
push
xp(l(i))=xp(l(i))+dt * vx(1)

remesh
i=int(xp):int(xp)+1, ug(i)=ug(i)+up*Γ(xp-i*dh) (for a 2 points formula)

and reinitialize particles on this line where needed :
-> j=1:nx, if ug(i) > threshold ,
npart=npart+1, xp(npart)=i*dh, up(npart)=ug(i)

Good news : Only 1d array for values on the grid !
Bad news : Need to label lines in a 6D space : 5d arrays

Good compromise between sizes of label and grid value arrays :

sort particles in 3D spaces : (x,y,z) spaces then (u,v,w) spaces

Remarks :

- accumulation of charges to compute density in a given (x,y,z) plane and calculation of field are done simultaneously with particle sorting

- push-remesh line by line can still be (and is) done inside each 3D space to reduce computational cost of high order remeshing kernels. Important for high order (large stencils) kernels.

- uniform velocity in each line -> order of method given by number of moments of cut-off (no need of regularity)

Memory requirements for 6D algorithm with Np particles :

- 7 main arrays of size Np for positions, velocity, distribution function
- 7 auxiliary arrays for same quantities
- 2 arrays of size Np to store particles addresses in link-list algorithm
- several 3D arrays for E, density, link-list

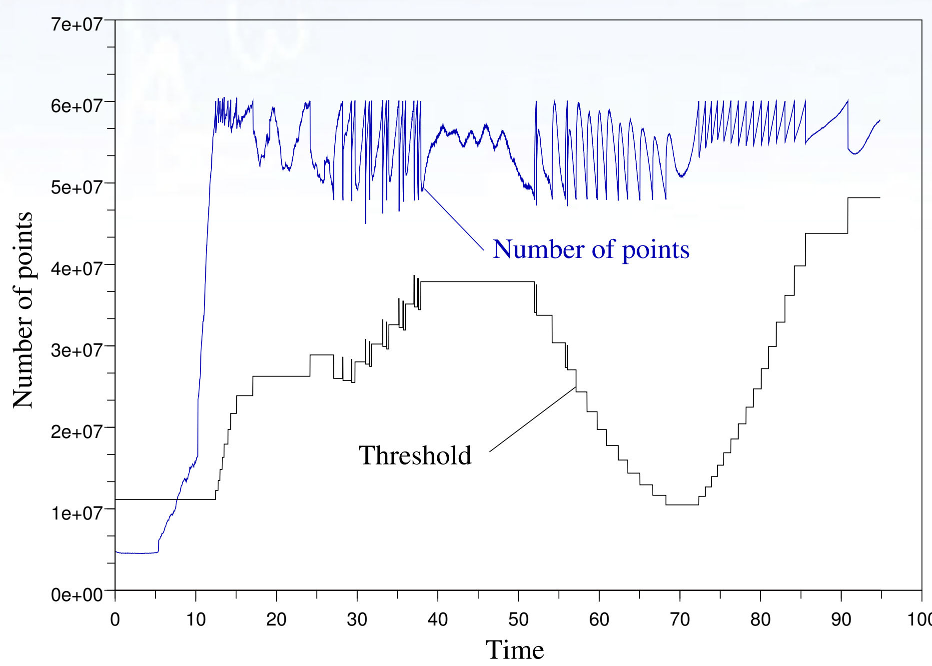
Goal of simulations :
test accuracy / efficiency of SL particles on uniform grids against MRA Eulerian methods and regular particle methods

Example 1: 2+2D plasma two beams instability, SL with 4th order kernel

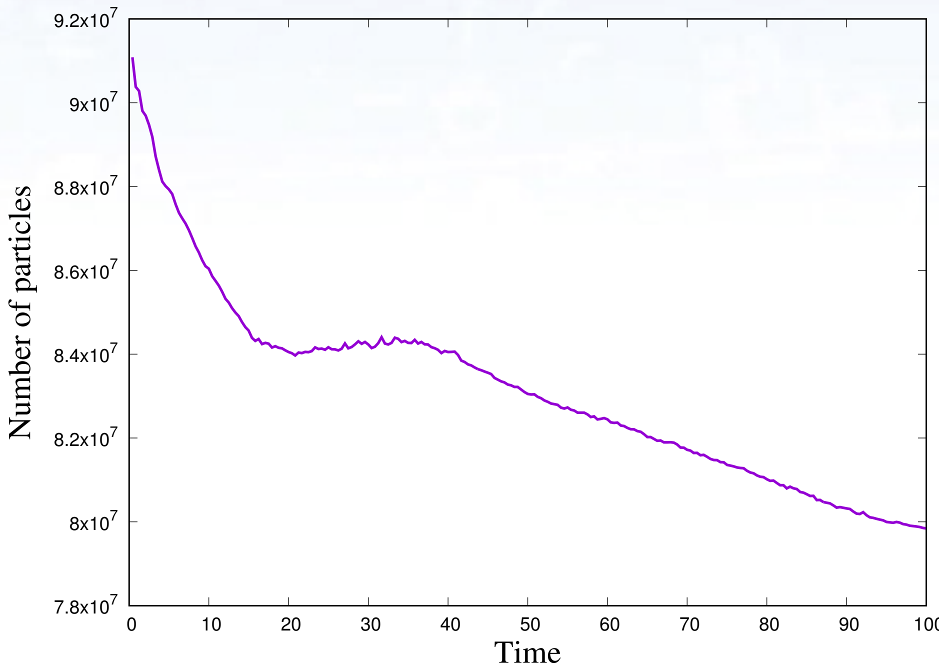
$$f_0(x,y,u,v) = \frac{7}{4\pi}\exp\left(-\frac{u^2+4v^2}{8}\right)\sin^2(\frac{u}{3})\,(1+0.05\cos(0.3x))$$

$$\Omega = [-\frac{10\pi}{3},\frac{10\pi}{3}]^2 \times [-3\pi,3\pi]^2$$

+ periodic boundary conditions for E.

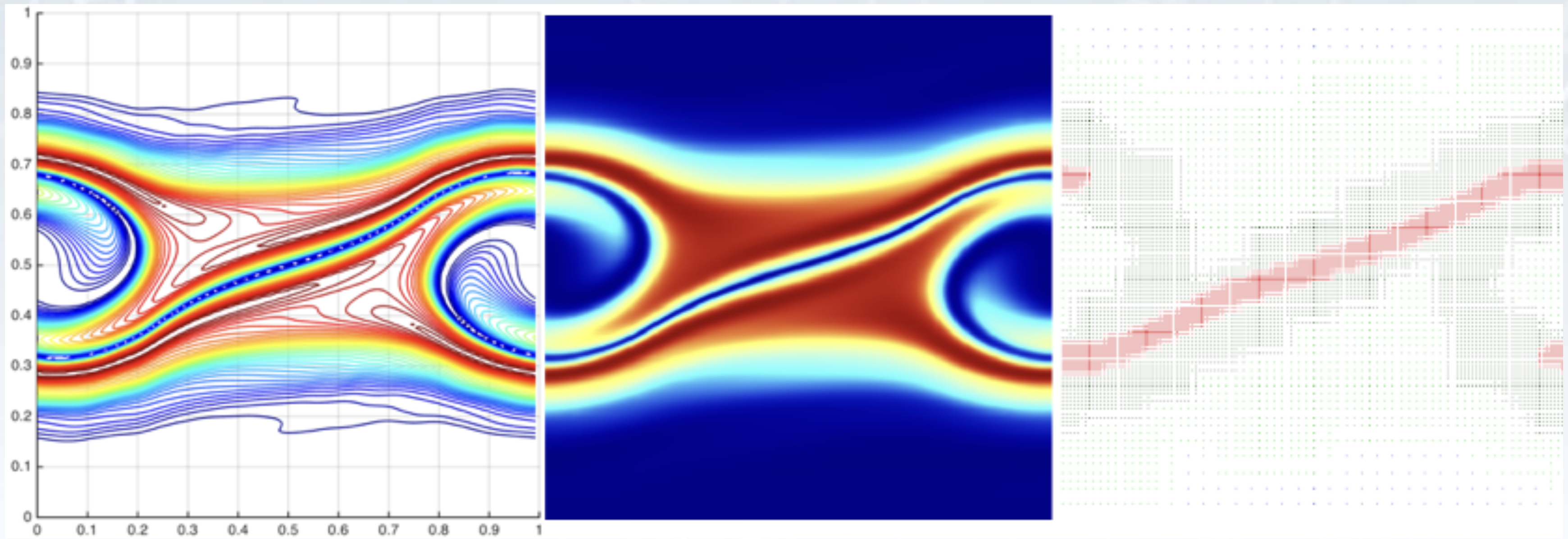


MRA grid (Deriaz-Periani) 32^4-256^4



number of particles on a 128^4 grid

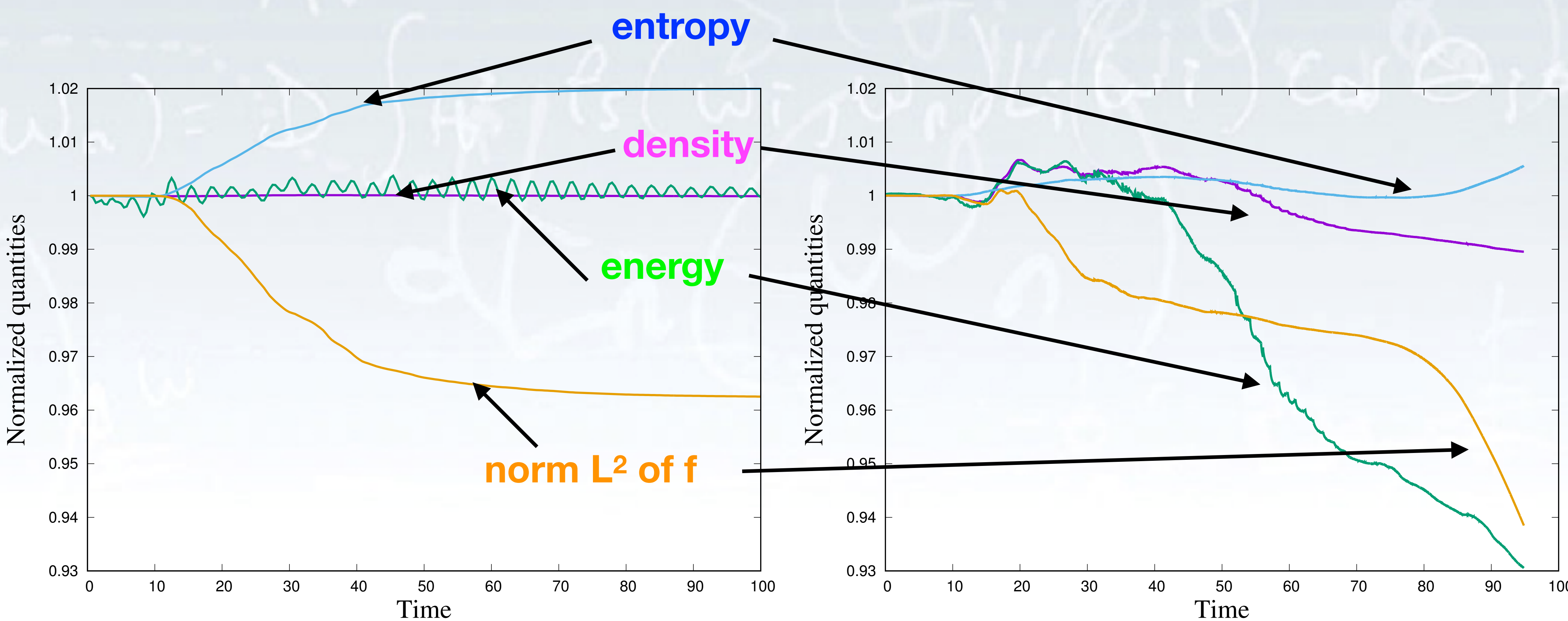
Qualitative comparison : cuts in (x,u) plane at t=12 (time of peak potential energy)



SL isolines of f　　MRA isolines of f　　MRA grid

Conservation properties :

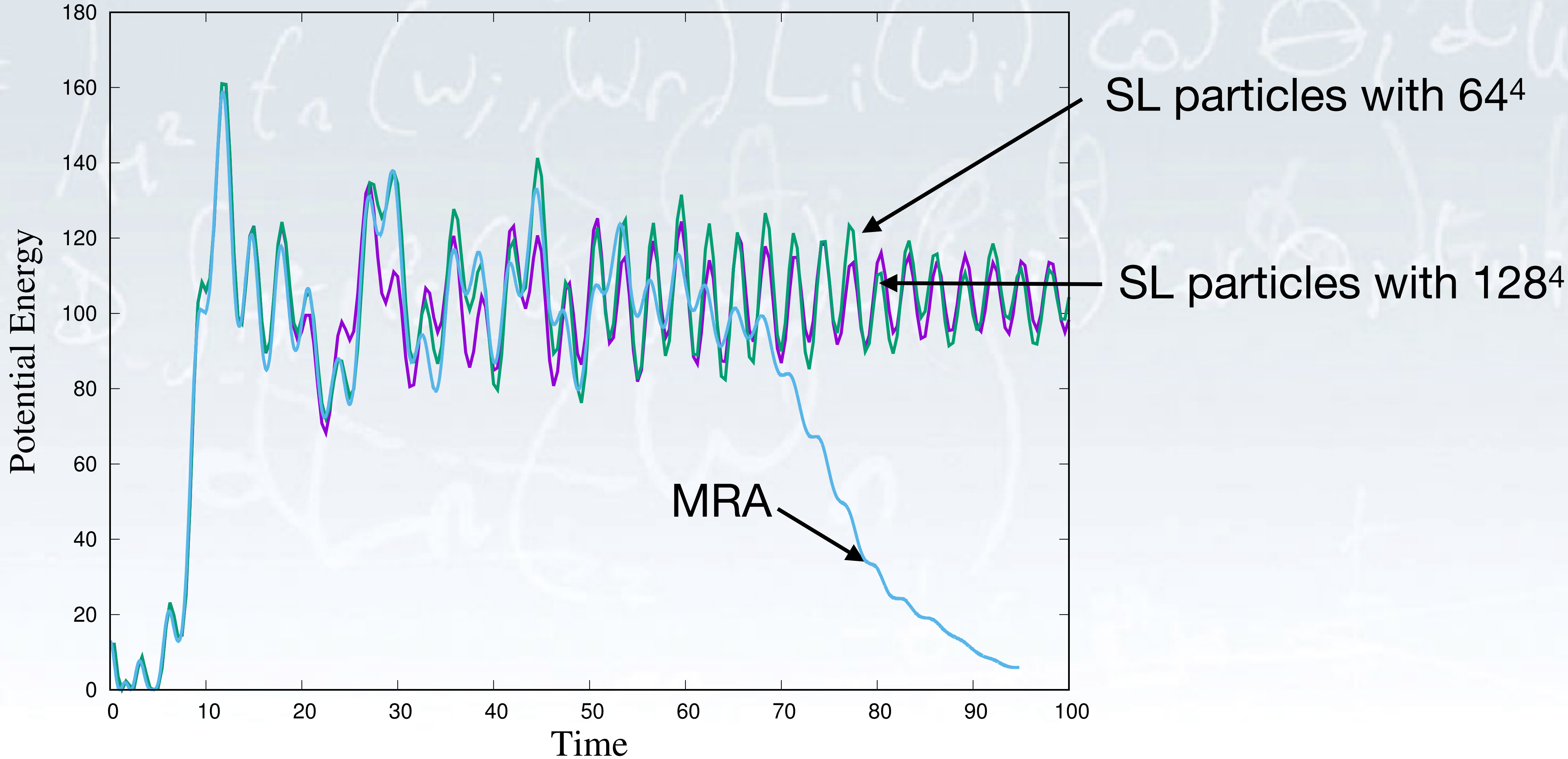
total density, L2 norm, entropy and total energy $\int v^2 f\,dxdv + \int E^2\,dx$



SL particle method grid with 128^4 grid

MRA grid (Deriaz-Periani) 32^4-256^4

LABORATOIRE JEAN KUNTZMANN
MATHÉMATIQUES APPLIQUÉES - INFORMATIQUE



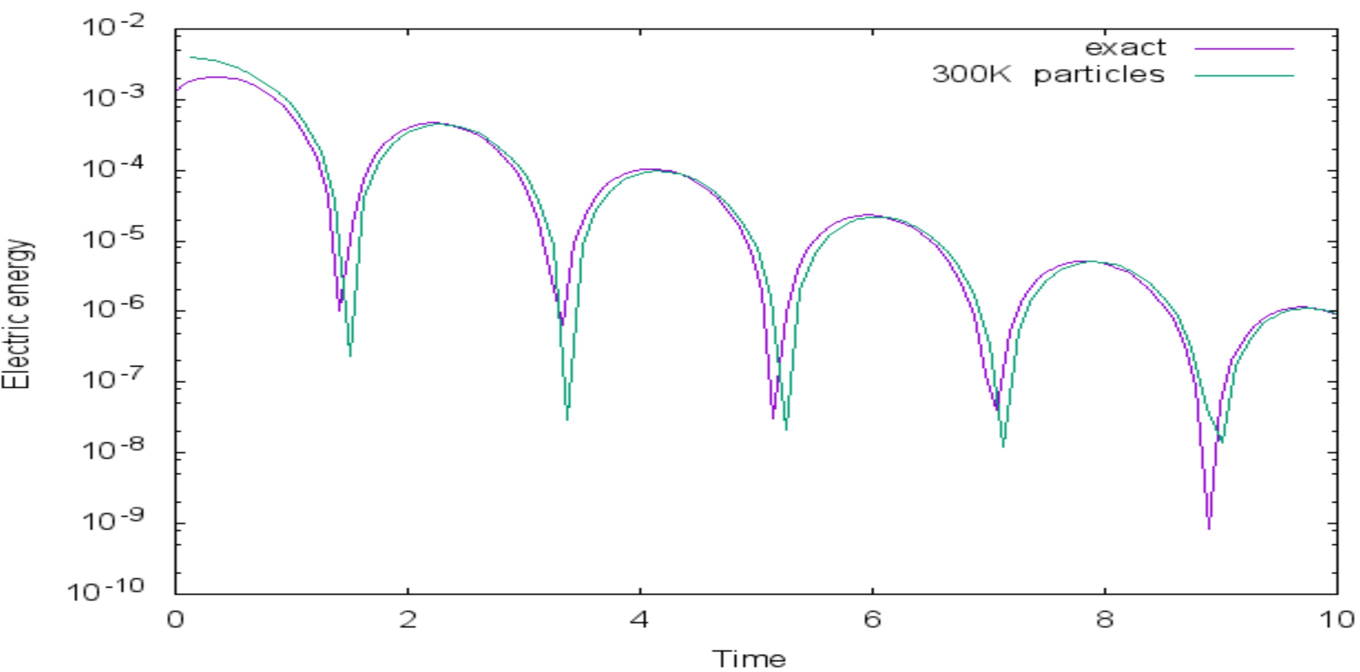
Comparison of potential energy $\int E^2$ obtained by MRA and SL particles at two resolutions

First conclusions on this case :

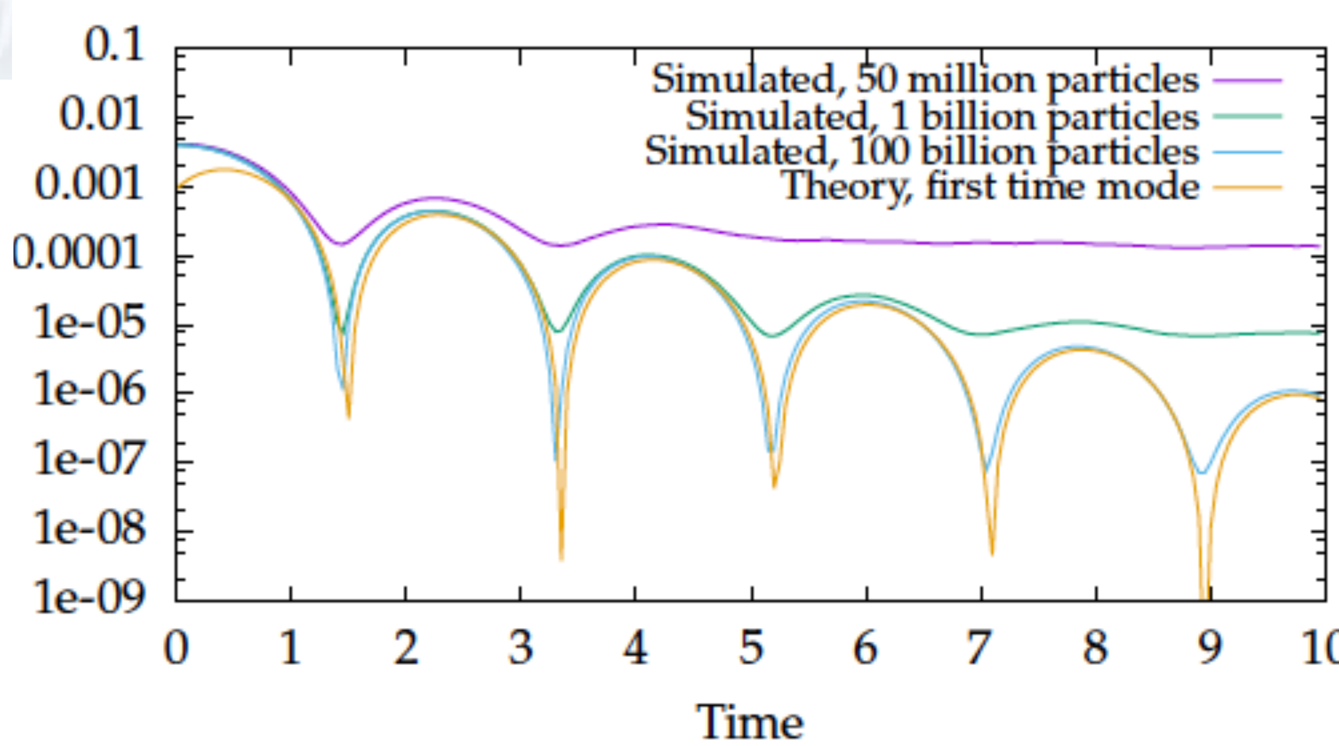
- **Accuracy of SL on uniform grids comparable to finite-difference MRA at higher local resolution**
- **Diffusive effects visible on Eulerian scheme, even at high (local) resolution, not seen on SL methods**

Comparison with pure PIC method on 2+2D Landau Damping

$$f_0(x,y,u,v) = \frac{1}{2\pi}\,\exp\left(-u^2 - v^2\right)\left(1 + 0.01\cos\frac{x}{2}\cos\frac{y}{2}\right)$$

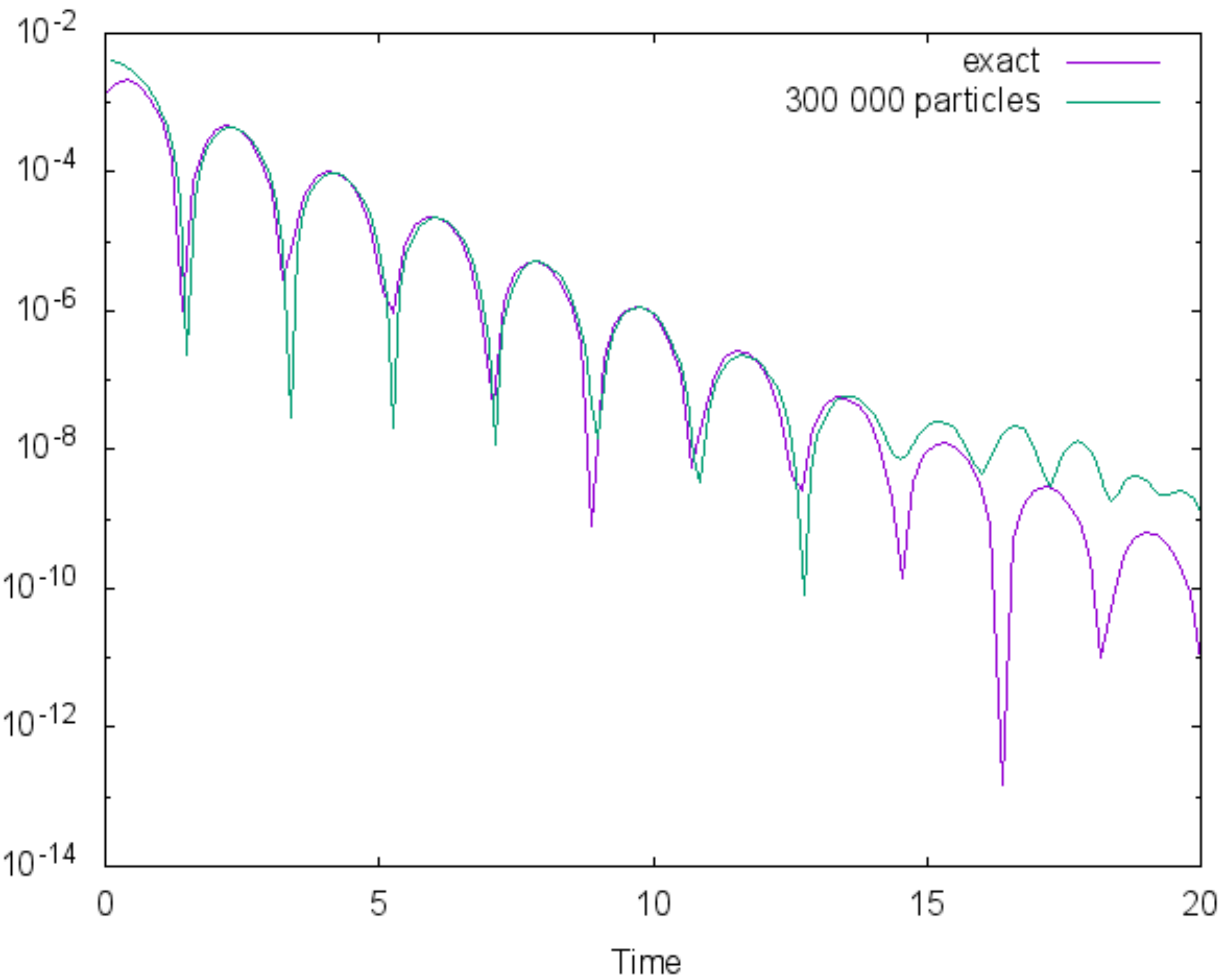


SL particles on 32^4 grid

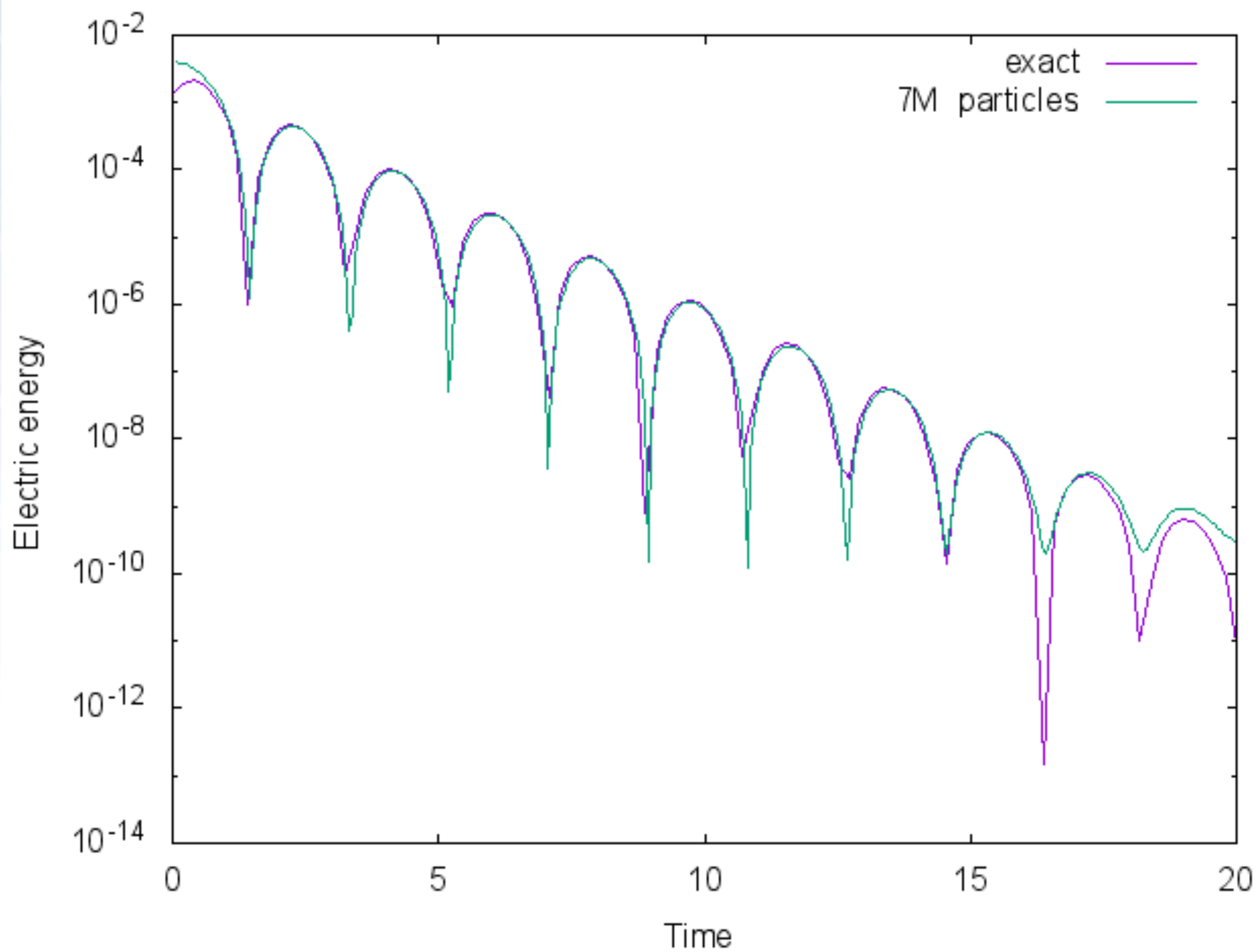


Classical PIC on 256^2 cells, with varying particle/cell ratio (from Y. Barsamian PhD thesis)

Good fit for longer times with higher resolution for SL particles



SL particles on 32^4 grid



SL particles on 64^4 grid

LABORATOIRE
JEAN KUNTZMANN
MATHÉMATIQUES APPLIQUÉES - INFORMATIQUE

Example 2 : 6D gravitational system

interaction of 2 Kipling spheres

$$f(r, \|\mathbf{v}\|) = \frac{3M}{7\pi^3 a^3}\left(2\left(1+\left(\frac{r}{a}\right)^2\right)^{-1/2} - \|\mathbf{v}\|^2\right)^{7/2}$$

$$\text{if} \quad 2\left(1+\left(\frac{r}{a}\right)^2\right)^{-1/2} - \|\mathbf{v}\|^2 \geq 0$$

$$\Omega = [-12, +12]^6$$

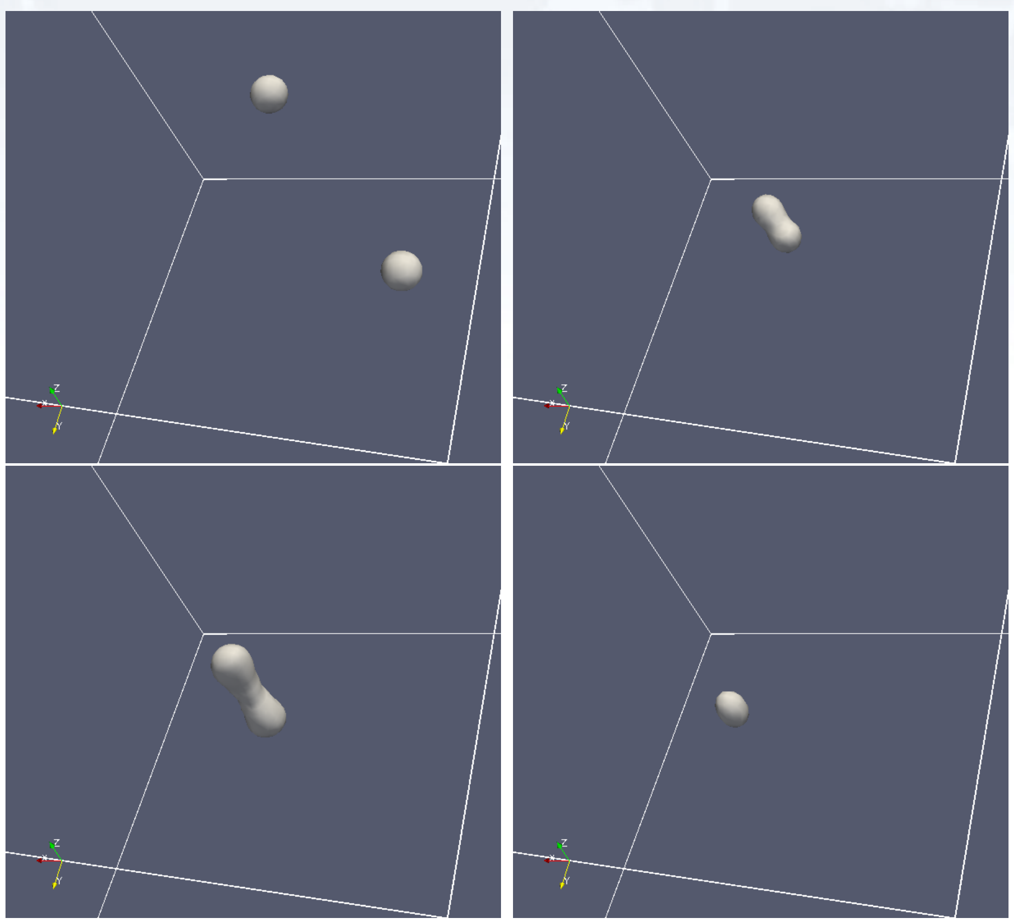
density isosurfaces

t=0.7

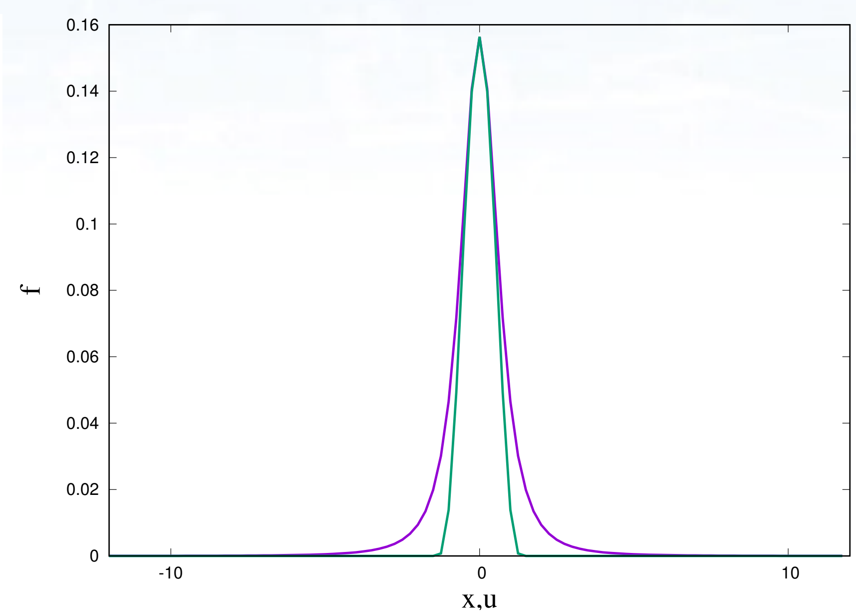
t=15.4
first collision

t=20

t=25
second collision

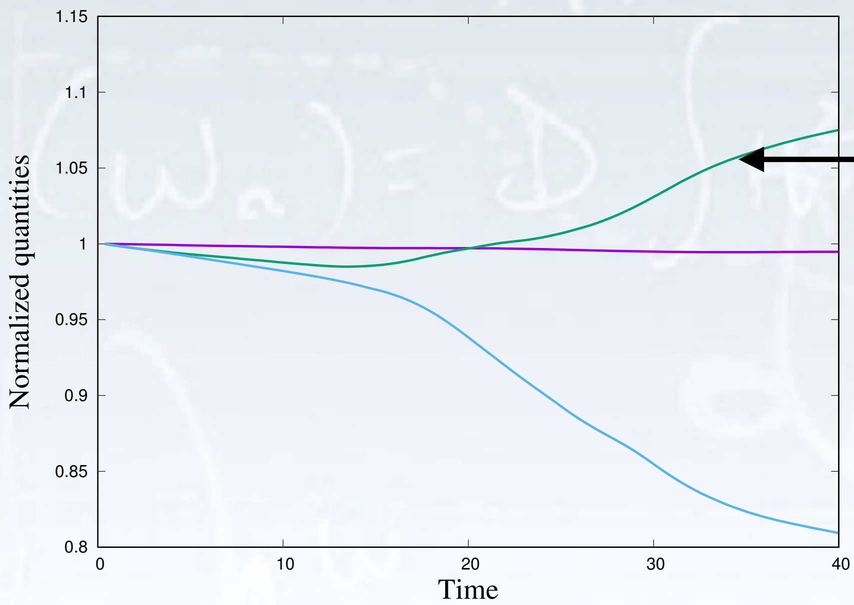


cuts of initial f-profile

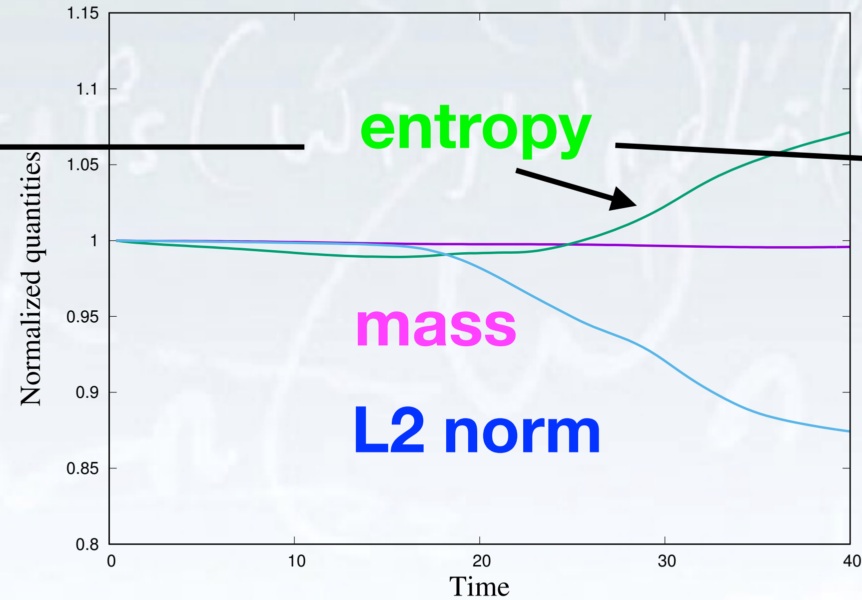


In that example, using a higher order SL particle method can be useful

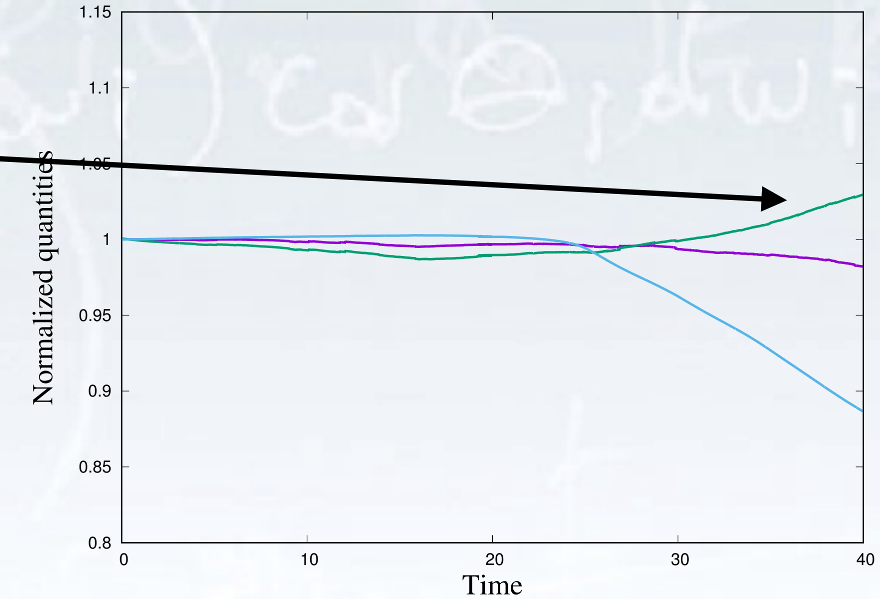
4th order SL method, 96^6



8th SL method, 96^6

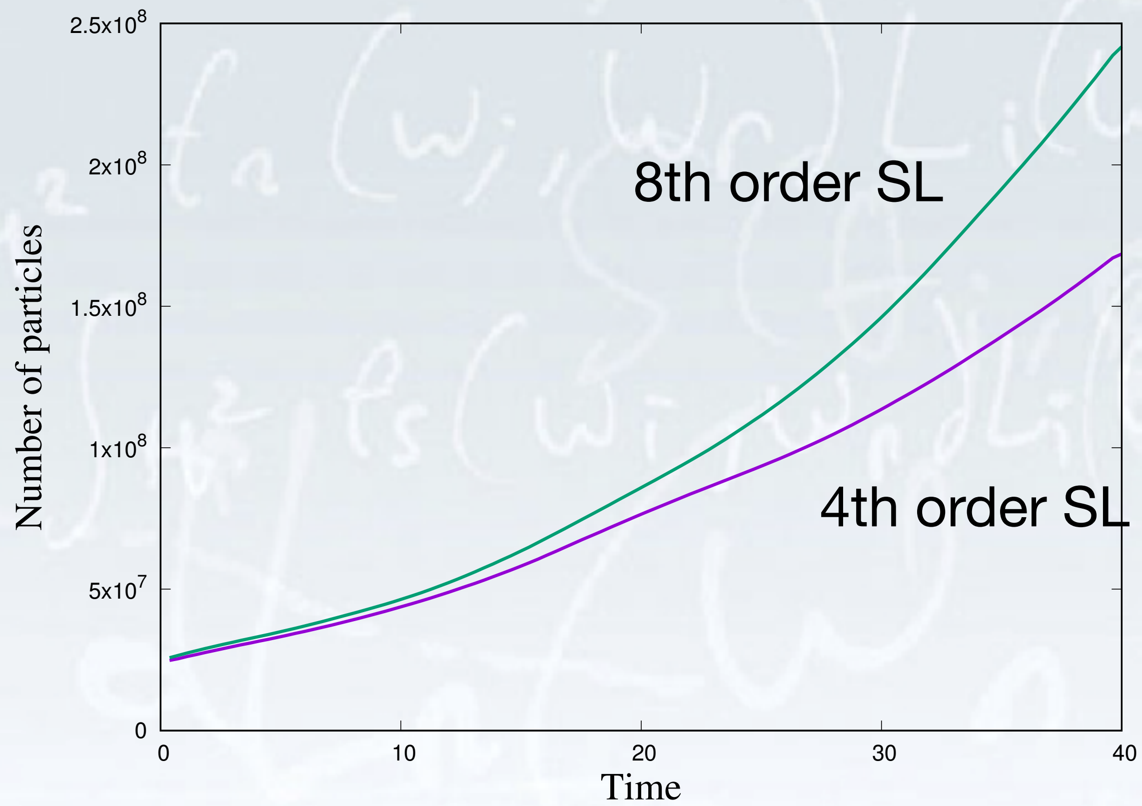


MRA 32^6-512^6



more visible with L2 norm at lower (64^6) resolution

LABORATOIRE
JEAN KUNTZMANN
MATHÉMATIQUES APPLIQUÉES - INFORMATIQUE

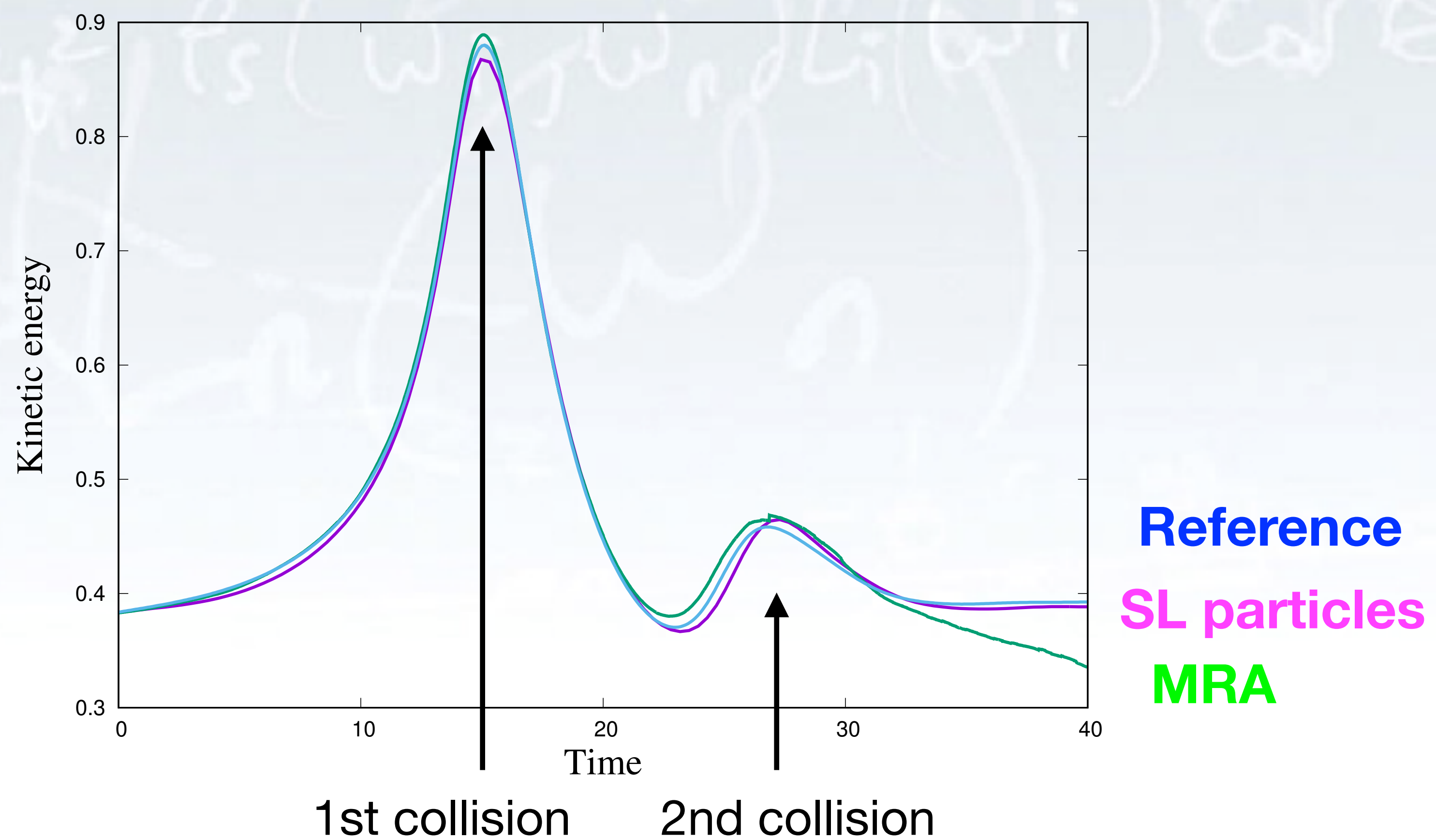


number of particles for SL methods on 96^6 grid

For comparison the 32^6-512^6 MR method of Deiraz-Peirani used up to 5 10^9 active grid points

LABORATOIRE JEAN KUNTZMANN
MATHÉMATIQUES APPLIQUÉES - INFORMATIQUE

Comparisons of kinetic energy :
SL particles vs MRA vs reference grid-free particles (code GADGET) using 500 M particles (fixed number)



Conclusion for this case :

-as in the plasma case for a SL particle methods using roughly the same number of particles on a uniform grid, profiles of invariants and kinetic energy are similar to Eulerian MRA code (with less numerical dissipation at late times)

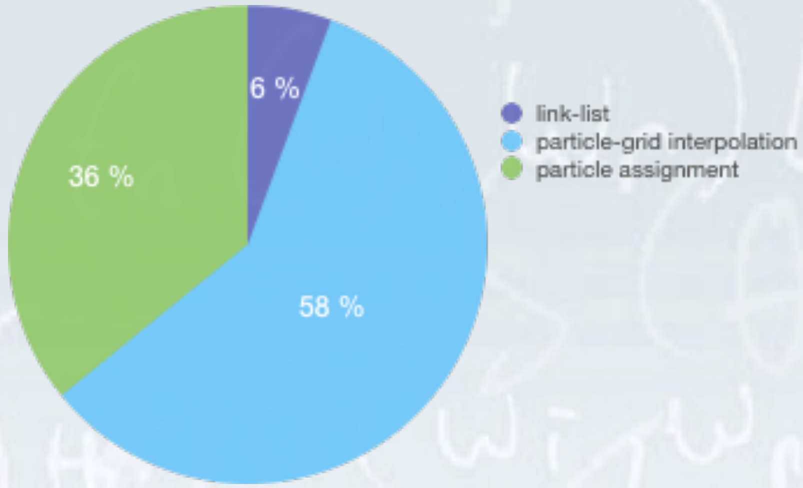
-results compare well with grid-free particle code using roughly same number of particles

So what ?

Cost ?

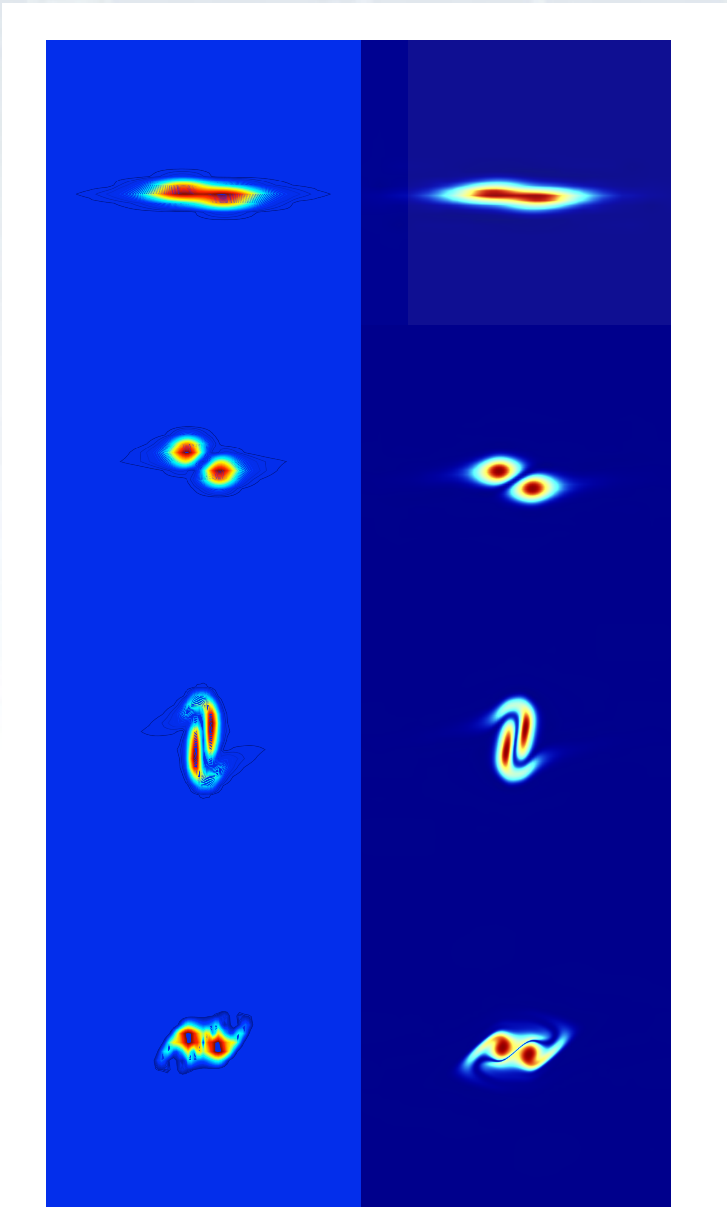
	4th order SL PM	8th order SL PM	Wavelet MRA [7]	GADGET [7]
Effective grid resolution	96	96	32 to 512	N.A.
Maximum number of active grid-points / particles	$1.8\,10^8$	$2.5\,10^8$	$5\,10^9$	$5\,10^8$
Number of time-steps	100	100	1349	N.A.
Wall clock CPU time	3.5 hours	5.8 hours	120 days	1 week
Hardware	1 Intel Xeon E5-2640 2.5 GHz	1 Intel Xeon E5-2640 2.5 GHz	32 Intel Xeon X5650 2.66GHz	500 cores

LABORATOIRE JEAN KUNTZMANN
MATHÉMATIQUES APPLIQUÉES - INFORMATIQUE



breakdown of main
steps of SL algorithm

SL particles MRA



The dark side :

- only very under-resolved simulations (can be seen on cuts of f)
- bounds of f ok for plasma case
  but (very) bad in astro case!

Conclusion :

There is room for simple methods on single cores even for complex problems

Combining SL particles with wavelets (already done for Navier-Stokes, Bergdorf et al SIAM MMS 2006) should be a killer

Wavelet-based multi-resolution particle methods (Bergdorf & Koumoutsakos, 2006)

At each time-step, wavelet-based MRA of (grid) quantities, based on interpolating wavelets :

$$q(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathcal{K}^0} c_{\boldsymbol{k}}^0\,\varphi_{\boldsymbol{k}}^0(\boldsymbol{x}) + \sum_{l=0}^{L-1}\sum_{\boldsymbol{k}\in\mathcal{K}^l}\sum_{\mu=1}^{2^d-1} d_{\boldsymbol{k}}^{l,\mu}\,\psi_{\boldsymbol{k}}^{l,\mu}(\boldsymbol{x})$$

where d is the dimension and scaling functions and wavelets are recursively given by filter operations

$$\varphi_{\boldsymbol{j}}^l = \sum_{\boldsymbol{k}} H_{\boldsymbol{j},\boldsymbol{k}}^l\,\varphi_{\boldsymbol{k}}^{l+1}, \qquad \psi_{\boldsymbol{j}}^{l,\mu} = \sum_{\boldsymbol{k}} G_{\boldsymbol{j},\boldsymbol{k}}^{l,\mu}\,\varphi_{\boldsymbol{k}}^{l+1}$$

LABORATOIRE JEAN KUNTZMANN
MATHÉMATIQUES APPLIQUÉES - INFORMATIQUE

Nested grids and grid adaptation based on thresholding detail coefficient
(Liandrat & Tchamitchian, 1990, Vasilyev 2003)

Particle method advects/remeshes scale solution at the successive scales, level by level, then add up results to reconstruct solution and perform MRA for next iteration

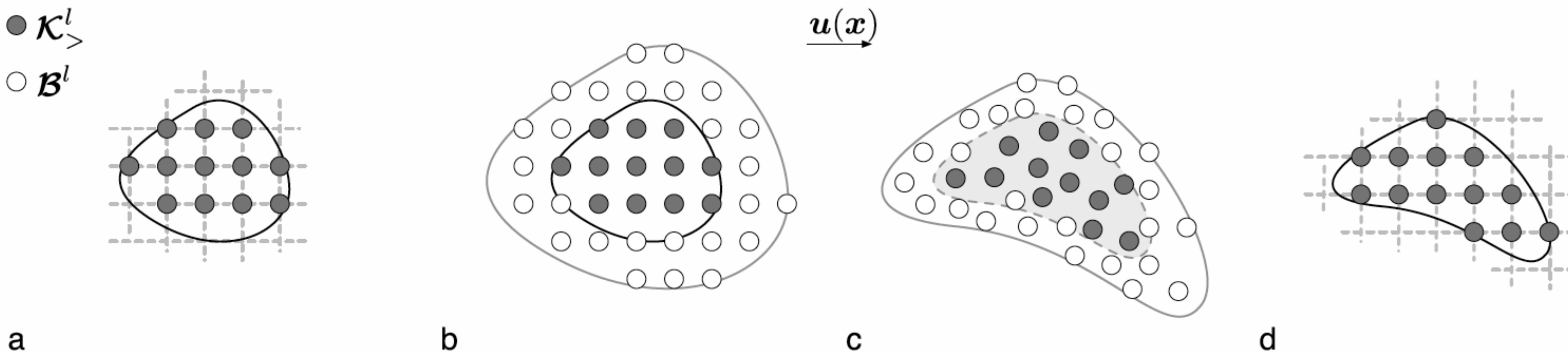
Nested grids, for wavelet coefficient above given threshold : $\{\mathcal{K}^l_>\}_{l=0}^L$

An additional buffer is created around particles activated at level l, with values obtained by interpolation form level l-1, to allow consistent remeshing

$$\boldsymbol{\mathcal{B}}^l = \left\{ \boldsymbol{k}' \,\middle|\, \min_{\boldsymbol{k}\in\mathcal{K}^l_>} |\boldsymbol{k}' - \boldsymbol{k}| \leq \lceil \tfrac{1}{2}\ \mathrm{supp}(\zeta) + \mathrm{LCFL} \rceil \right\}$$

Finally, like for grid-based methods, need to allow levels l+1 to appear from level l during advection

Algorithm for time advancement of particles at a given level l

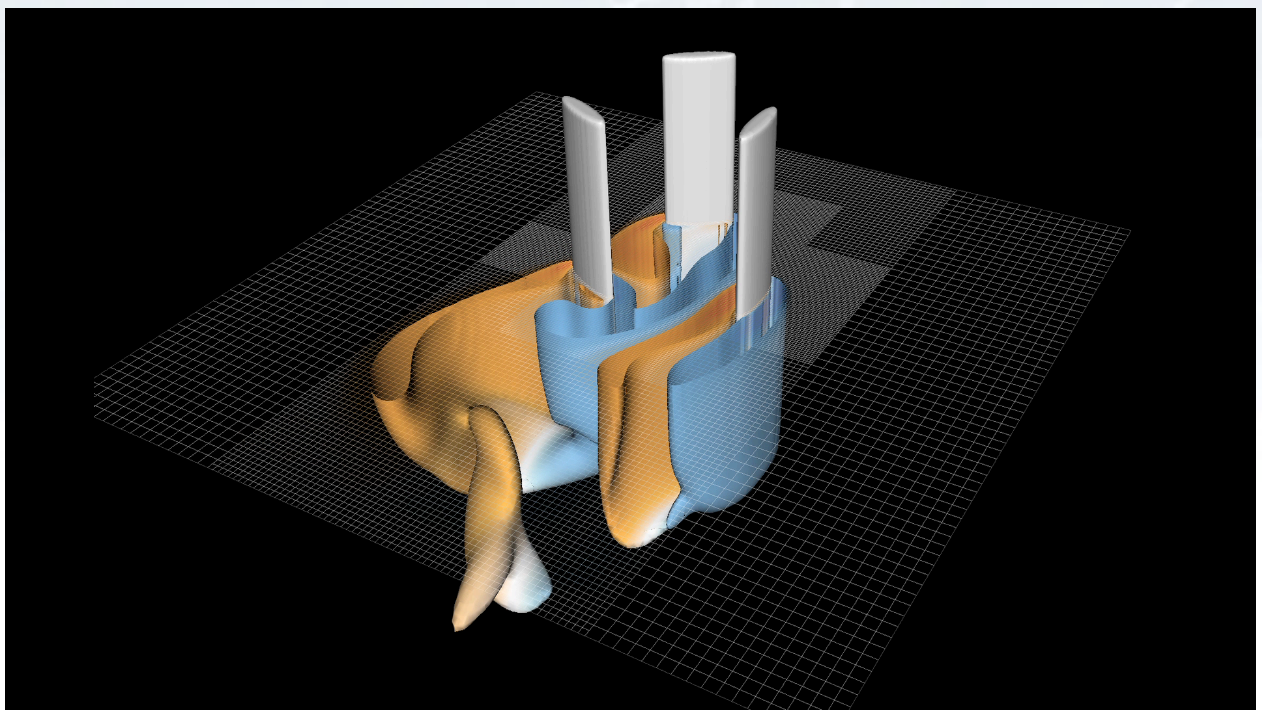


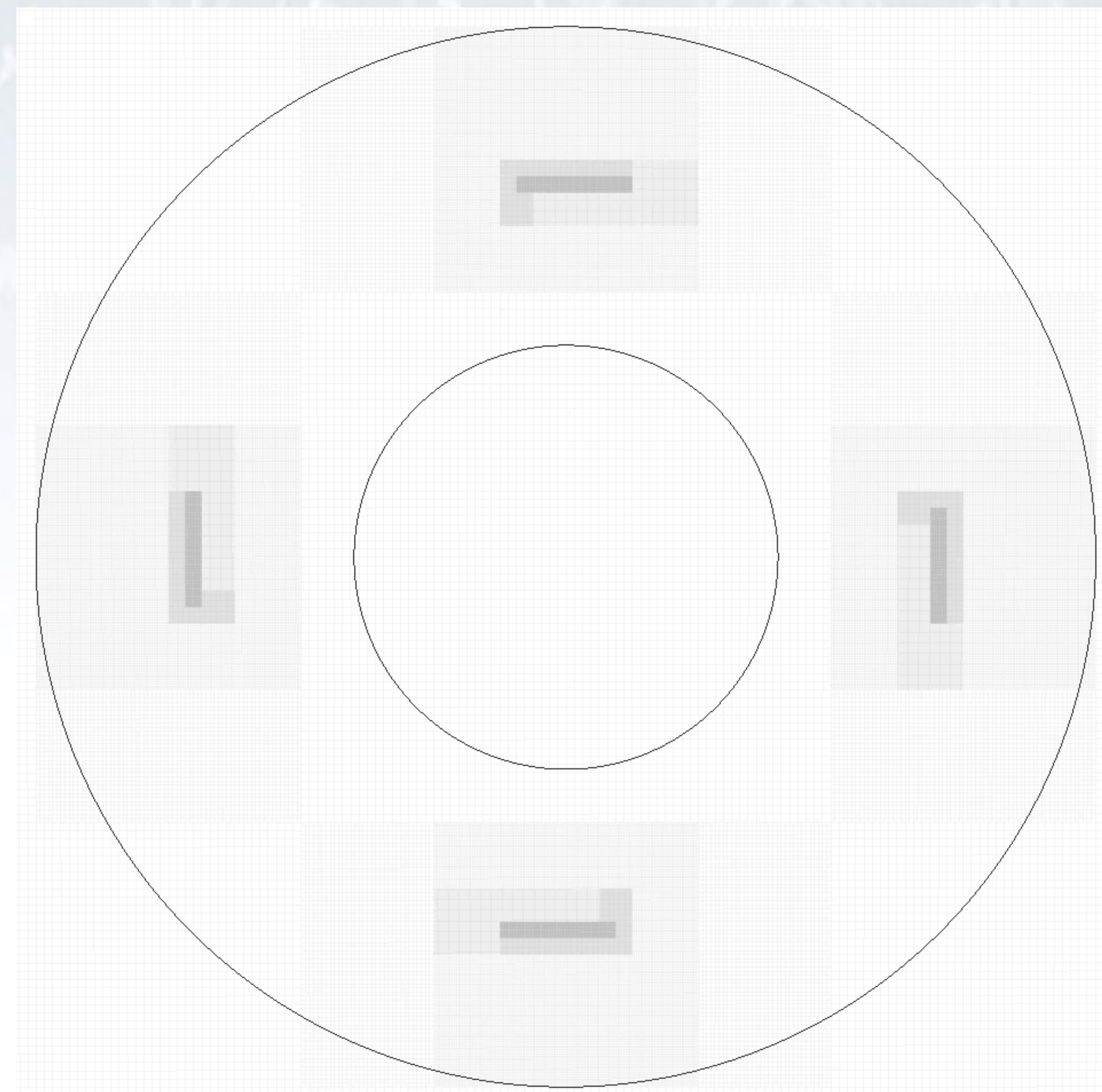
select «active» particles on the grid (with tag=1)	create a buffer around these particles (with tag=0)	advect particles and tag	remesh particles and tag; **keep particles with tag > 0**

additional trick : to allow levels ℓ+1 to appear through advection of level ℓ,
remesh level ℓ particles on grid ℓ+1
consistent with lagrangian CFL for time-step

Illustration of MRA SL particles for flow around a wind turbine (ETH group of Koumoutsakos)

Ingredients : wavelet-based particles for vorticity transport and
Brinkman penalization for non-slip boundary conditions (Angot et al., 1999, Coquerelle & Cottet, 2008)





LABORATOIRE JEAN KUNTZMANN
MATHÉMATIQUES APPLIQUÉES - INFORMATIQUE

Extension to Vlasov-Poisson « should » be straightforward

with (at least) two questions :

1) splitting (x,y,z)/(u,v,w) does not allow fine scales to appear dynamically

-> each level passively transported

-> is it a problem ?

-> should one choose other splitting strategy e.g. (x,u)/(y,v)/(z,w) ?

2) kinetic equations vs fluid models : worth it ?