Solving Vlasov-Poisson equations with a "metric" approach

S. Colombi & C. Alard
Institut d'Astrophysique de Paris

Colombi & Alard, 2017, Journal of Plasma Physics 83, 705830302

www.vlasix.org

Direct Vlasov solvers in the warm case

Vlasov-Poisson equations:

f : phase space density

Φ: gravitational potential

ρ : projected density

$$\frac{\partial f}{\partial t} + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} f - \nabla_{\boldsymbol{x}} \phi \cdot \nabla_{\boldsymbol{v}} f = 0,$$

$$\Delta_{\boldsymbol{x}} \phi = 4\pi G \rho, \qquad \rho(\boldsymbol{x}, t) \equiv \int f(\boldsymbol{x}, \boldsymbol{v}, t)\, \mathrm{d}\boldsymbol{v}$$

Warm case (e.g. *relaxed* dark matter halo, stars in galaxies): numerous methods mainly invented for plasma physics, mostly of *semi-Lagrangian* nature (e.g. Yoshikawa's, Grandgirard's and Imodura's talks, review of Besse). Among them:

- The *waterbag method* (DePackh 1962, Robert & Berk 1967, and followers, e.g. Colombi & Touma 2008, 2014)
- The *splitting algorithm* (Cheng & Knorr 1976) and its variants and improvements, e.g Discontinuous Galerkin methods (e.g. Mehrenberger's talk)
- But *many others*: finite differences (e.g., Domínguez Fernández's poster), finite elements, lattice dynamics (e.g. Mocz's talk), Schrödinger method (e.g., Kopp's talk) etc

Some bibliographic details: e.g. Alard & Colombi 2005, Sousbie & Scolombi 2016, Besse's talks, 2015 and 2017: http://www.vlasix.org/uploads/Main/Besse.pdf and BesseTalk.pdf

Classical semi-Lagrangian code: the splitting algorithm

The splitting algorithm of Cheng & Knorr (1976, JCP 22, 330) exploits Liouville theorem, namely conservation of phase-space density along characteristics: $f[\mathbf{r}(t), \mathbf{u}(t), t] = \text{constant}$

- The phase space distribution function is sampled on a grid
- A test particle is associated to each grid site and followed backwards in time to find its position at previous time step.
- *f* is interpolated from previous time step at the root of the characteristic using e.g. spline interpolation
- This is performed in a split fashion:

$$f^*(\boldsymbol{r}, \boldsymbol{u}) = f(\boldsymbol{r} - \boldsymbol{u}\Delta t/2, \boldsymbol{u}, t), \quad \text{Drift},$$

$$f^{**}(\boldsymbol{r}, \boldsymbol{u}) = f^*(\boldsymbol{r}, \boldsymbol{u} + \nabla_r \phi \Delta t), \quad \text{Kick},$$

$$f(\boldsymbol{r}, \boldsymbol{u}, t + \Delta t) = f^{**}(\boldsymbol{r} - \boldsymbol{u}\Delta t/2, \boldsymbol{u}), \quad \text{Drift},$$

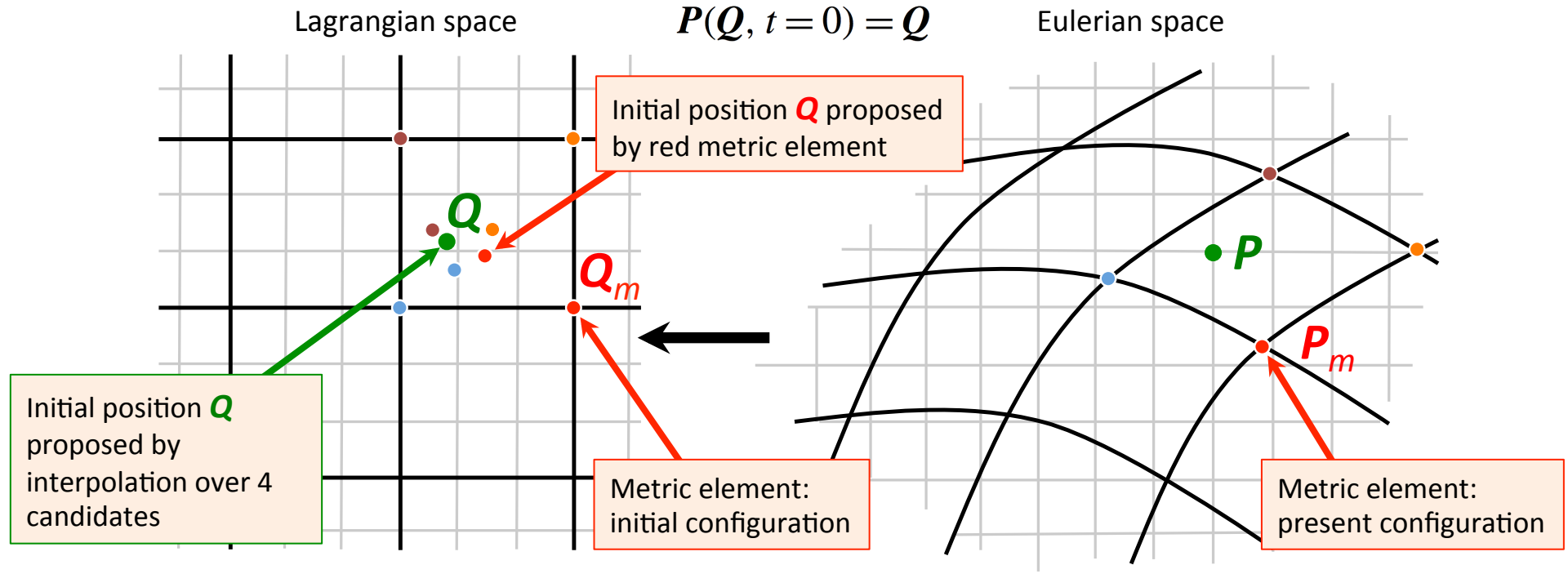
First applications in astrophysics: Fujiwara (1981), Nshida et al. (1981), Watanabe et al. (1981)

Improvement over the splitting algorithm: the ``metric'' approach

Colombi & Alard 2017, J. Plasma Phys. 83, 705830302

See also Campos Pinto & Charles, HAL-01385676

- A set of metric elements is used to follow locally the flow and its deformation



- Finding the root of the characteristics: **2nd order Lagrangian perturbation theory**

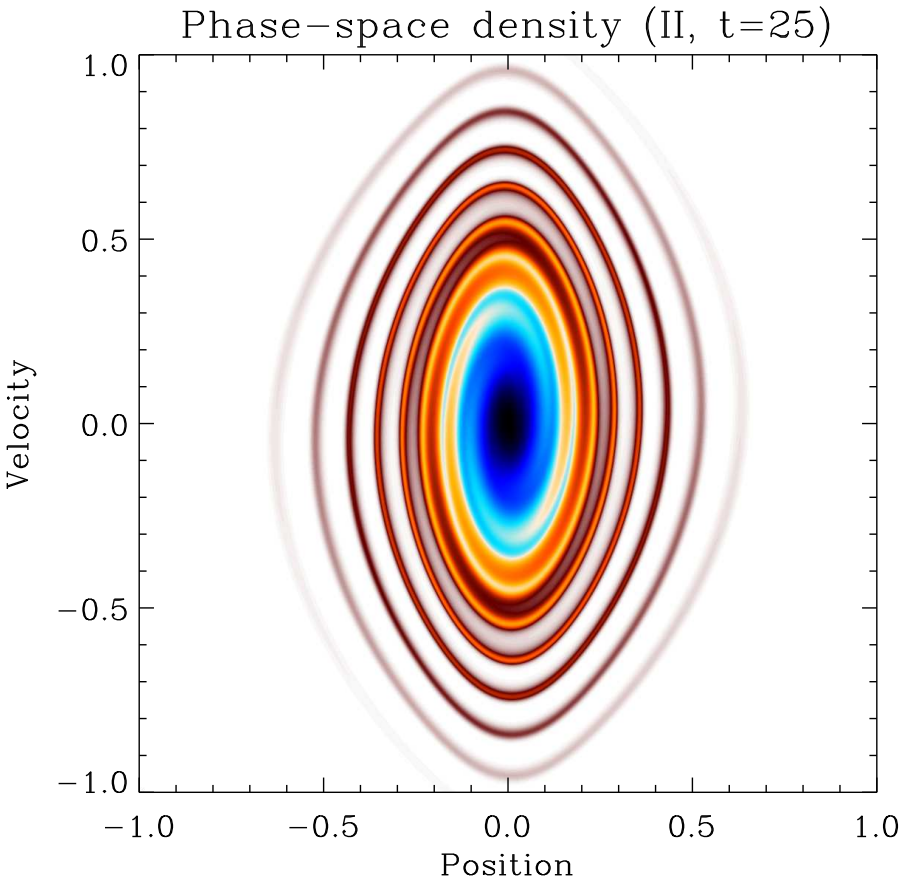
$$\boldsymbol{Q} \simeq \boldsymbol{Q}_m + \boldsymbol{T}_m^{-1}(\boldsymbol{P}-\boldsymbol{P}_m) - \tfrac{1}{2}\boldsymbol{T}_m^{-1}(\boldsymbol{P}-\boldsymbol{P}_m)^{\mathrm{T}}[\boldsymbol{T}_m^{-1}]^{\mathrm{T}}\boldsymbol{H}_m\boldsymbol{T}_m^{-1}(\boldsymbol{P}-\boldsymbol{P}_m)$$

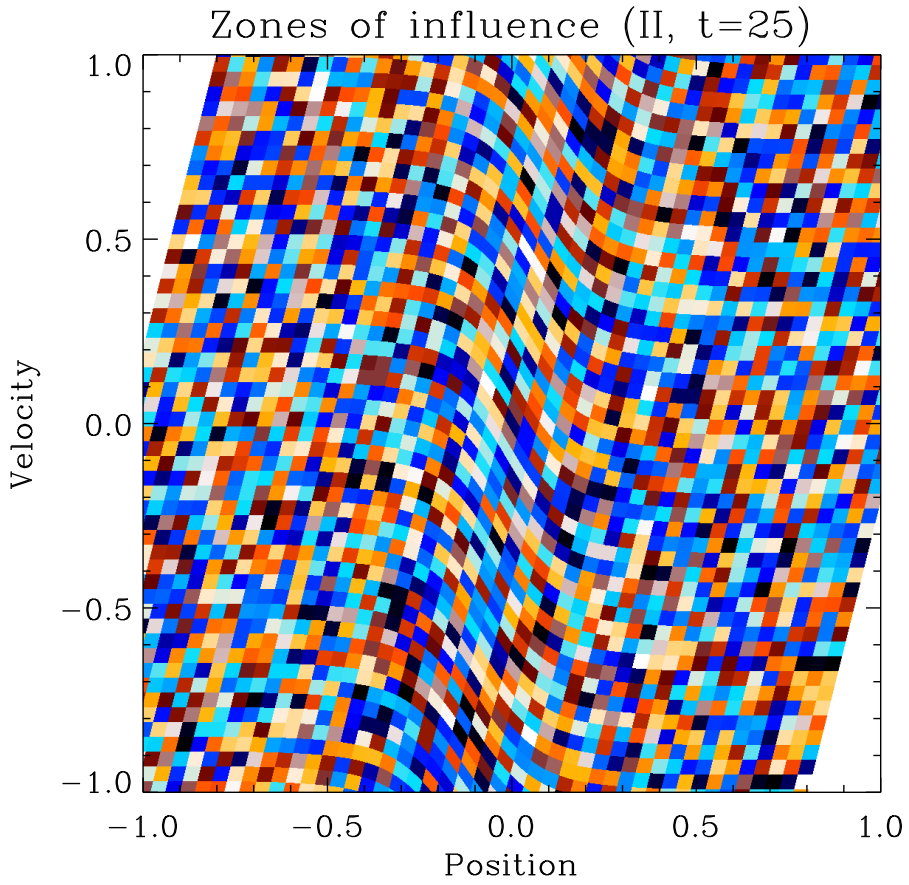
Deformation tensor: $\boldsymbol{T}(\boldsymbol{Q}, t) \equiv \dfrac{\partial \boldsymbol{P}}{\partial \boldsymbol{Q}}$

Hessian of the displacement: $\boldsymbol{H}(\boldsymbol{Q}, t) \equiv \dfrac{\partial^2 \boldsymbol{P}}{\partial \boldsymbol{Q}^2}$

Lagrangian regions of influence: Percolation algorithm

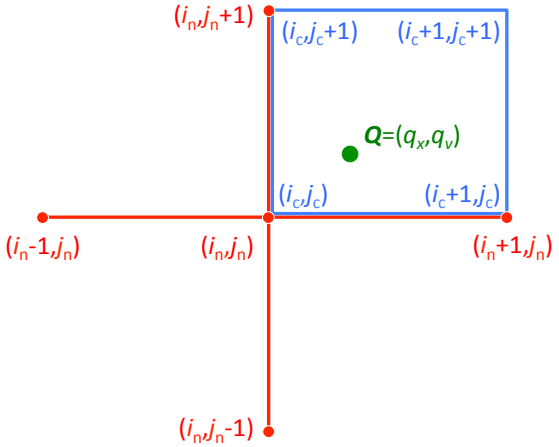
A percolation algorithm is implemented to compute accurately the Lagrangian region of influence of each metric element.





Improvement over the splitting algorithm: the ``metric'' approach

- One can find the initial (Lagrangian) position ***Q***(***P***) of any test particle P by expanding at second order the geometry of the motion in the vicinity of the closest metric element
- Reconstruction of the phase-space distribution function at each time step is performed using Liouville theorem, by interpolation of *f* on initial position ***Q(P)***, similarly as in the splitting method, however a lower 2nd order interpolation is used:



$$
\begin{aligned}
f(q_x, q_v) = {} & F(i_n, j_n)(1 - \mathrm{d}q_x^2 - \mathrm{d}q_v^2) \\
& + \tfrac{1}{2}F(i_n + 1, j_n)\,\mathrm{d}q_x(1 + \mathrm{d}q_x) + \tfrac{1}{2}F(i_n - 1, j_n)\,\mathrm{d}q_x(1 - \mathrm{d}q_x) \\
& + \tfrac{1}{2}F(i_n, j_n + 1)\,\mathrm{d}q_v(1 + \mathrm{d}q_v) + \tfrac{1}{2}F(i_n - 1, j_n)\,\mathrm{d}q_v(1 - \mathrm{d}q_v) \\
& + [F(i_c + 1, j_c + 1) + F(i_c, j_c) - F(i_c + 1, j_c) - F(i_c, j_c + 1)]\,\mathrm{d}q_x\,\mathrm{d}q_v
\end{aligned}
$$

$$
\begin{aligned}
(i_n, j_n) &= (\lfloor (q_x - q_{x,mid})/\Delta x \rceil,\ \lfloor (q_v - q_{v,mid})/\Delta v \rceil), \\
(i_c, j_c) &= (\lfloor (q_x - q_{x,mid})/\Delta x \rfloor,\ \lfloor (q_v - q_{v,mid})/\Delta v \rfloor), \\
\mathrm{d}q_x &= (q_x - q_{x,mid})/\Delta x - i_n, \\
\mathrm{d}q_v &= (q_v - q_{v,mid})/\Delta v - j_n,
\end{aligned}
$$

- When deformation of the metric elements is too high, new isotropic elements are set along with new initial conditions corresponding to the current state: cubic B-splines are used at the moment of resampling and ***Q***(***P***) is computed more accurately using interpolation between neighbouring metric elements.
- Because re-samplings of the phase-space distribution function are much more seldom, **the metric scheme is much less diffusive than the standard splitting algorithm**

Evolution of metric elements: 2nd order drift-kick-drift

- Drift step $\Delta t/2$

$$\begin{aligned}
\boldsymbol{x}(t+\Delta t/2) &= \boldsymbol{x}(t) + \tfrac{1}{2}\boldsymbol{v}(t)\Delta t, \\
T_{i,j}(t+\Delta t/2) &= T_{i,j}(t) + \tfrac{1}{2}T_{i+D,j}(t)\Delta t, \quad i \leqslant D, \\
H_{i,j,k}(t+\Delta t/2) &= H_{i,j,k}(t) + \tfrac{1}{2}H_{i+D,j,k}\Delta t, \quad i \leqslant D
\end{aligned}$$

- Calculation of phase-space density at $t+\Delta t/2$, hence acceleration $\boldsymbol{a}(t+\Delta t/2)$

- Kick step Δt

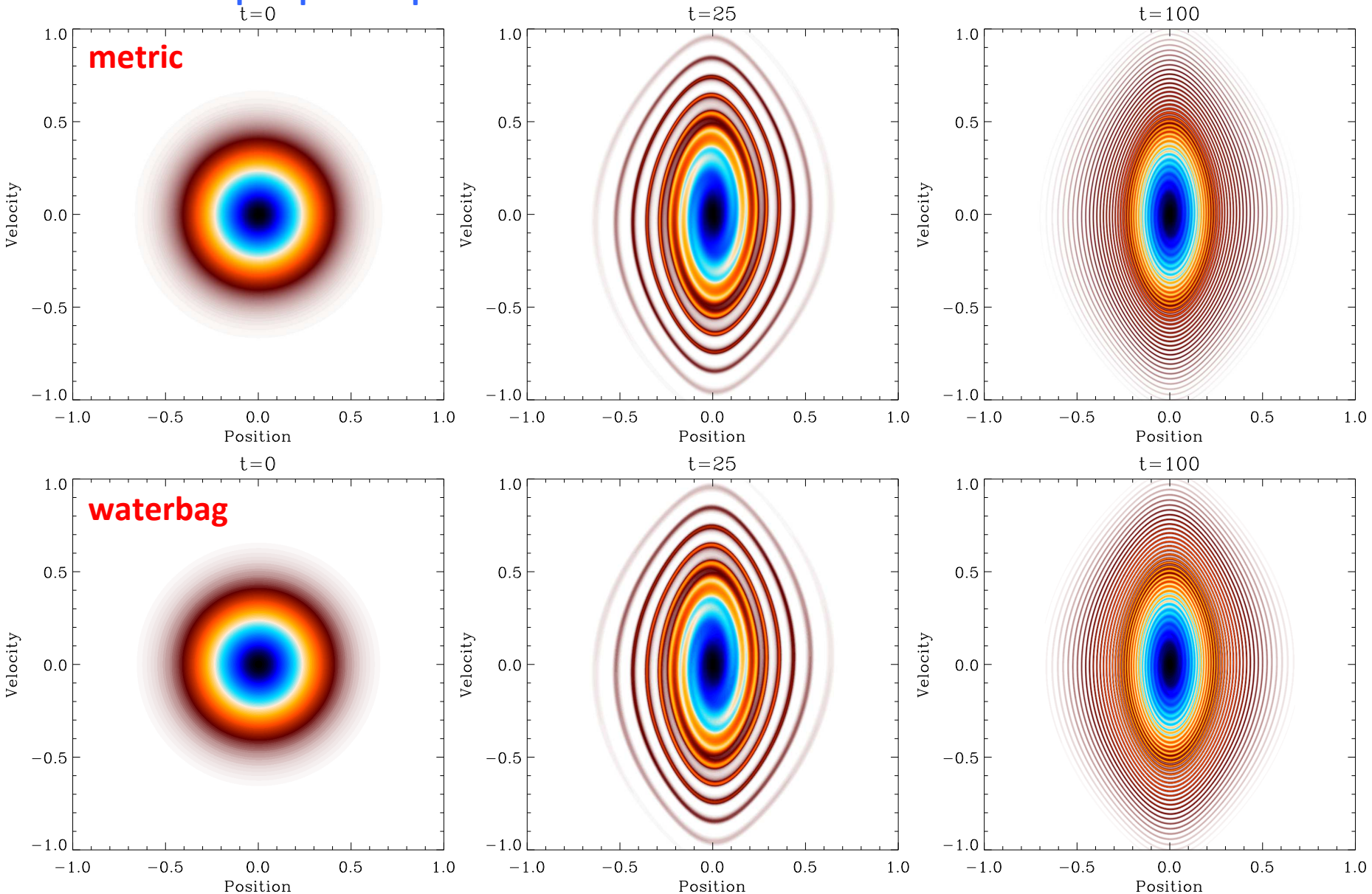
$$\begin{aligned}
\boldsymbol{v}(t+\Delta t) &= \boldsymbol{v}(t) + \boldsymbol{a}(t+\Delta t/2)\Delta t, \\
T_{i+D,j}(t+\Delta t) &= T_{i,j}(t) + \sum_{r=1}^{D} \frac{\partial a_i}{\partial x_r} T_{r,j}(t+\Delta t/2)\Delta t, \\
H_{i+D,j,k}(t+\Delta t) &= H_{i+D,j,k}(t) + \sum_{r=1}^{D} \frac{\partial a_i}{\partial x_r} H_{r,j,k}(t+\Delta t/2)\Delta t \\
&\quad + \sum_{r,s=1}^{D} \frac{\partial^2 a_i}{\partial x_r \partial x_s} T_{r,j}(t+\Delta t/2)\, T_{s,k}(t+\Delta t/2)\Delta t
\end{aligned}$$

- Drift step $\Delta t/2$

$$\begin{aligned}
\boldsymbol{x}(t+\Delta t) &= \boldsymbol{x}(t+\Delta t/2) + \tfrac{1}{2}\boldsymbol{v}(t+\Delta t)\Delta t, \\
T_{i,j}(t+\Delta t) &= T_{i,j}(t+\Delta t/2) + \tfrac{1}{2}T_{i+D,j}(t+\Delta t)\Delta t, \quad i \leqslant D, \\
H_{i,j,k}(t+\Delta t) &= H_{i,j,k}(t+\Delta t/2) + \tfrac{1}{2}H_{i+D,j,k}(t+\Delta t)\Delta t, \quad i \leqslant D
\end{aligned}$$

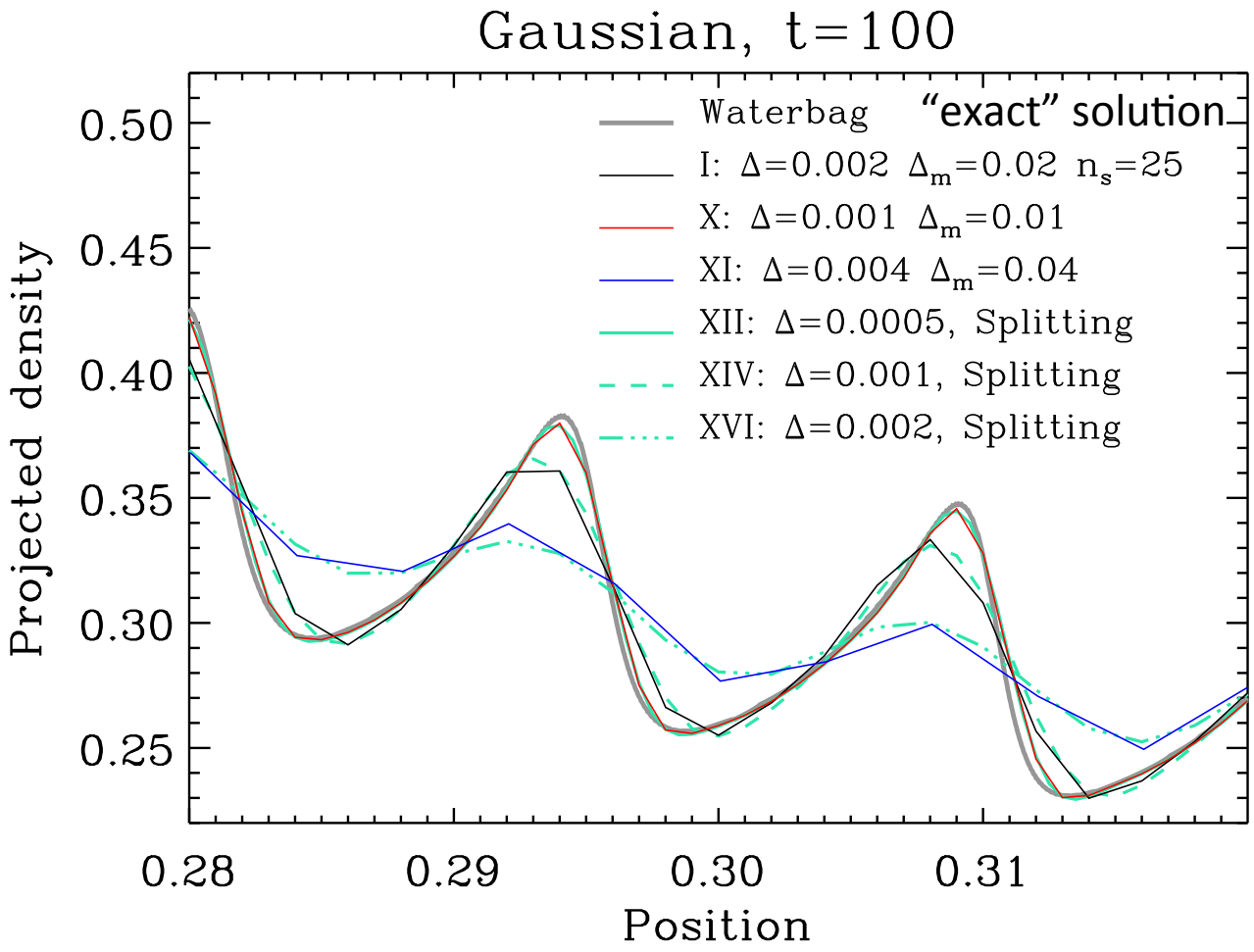
Metric versus waterbag

Example: phase-space of a 1D simulation with Gaussian initial conditions



Metric versus splitting (and waterbag)

Example: phase-space of a 1D simulation with Gaussian initial conditions



(Colombi & Touma 2014)

The metric scheme allows one to improve actual spatial resolution by a factor 2 or more for a computational cost of the same order

Performances in 1D: metric versus splitting

Designation	Δ	Δ_m	n_s	t_{CPU}
I	0.002	0.02	25	1.0
II	0.002	0.04	25	1.0
III	0.002	0.01	25	1.2
IV	0.002	0.005	25	1.4
V	0.002	0.02	100	1.1
VI	0.002	0.02	50	1.1
VII	0.002	0.02	10	1.2
VIII	0.002	0.02	5	1.5
IX	0.002	0.02	1	3.2
X	0.001	0.01	25	4.0
XI	0.004	0.04	25	0.4
XII	0.0005	—	—	8.8
XIII	0.00707	—	—	4.5
XIV	0.001	—	—	2.3
XV	0.00141	—	—	1.2
XVI	0.002	—	—	0.7

Increasing metric element density slightly increases computational cost

- Δ: spatial resolution
- Δ_m: inter-element of metric distance
- n_s: number of time step between full resamplings
- t_{CPU}: total CPU time spent in units of simulation I

$t_{CPU} \sim \Delta^{-2D}$ *D:* dimension of space

Metric code slightly more costly than splitting algorithm for D=1

Conclusions

Energy conservation: metric algorithm does slightly worse than splitting scheme but can conserve energy at a very good level (e.g. below the 10^{-5} level) if metric element density is sufficiently high (e.g. distance between metric elements of the order of 2.5 grid element size)

Information (entropy) conservation: metric algorithm does as well as splitting scheme but by using twice or even 4 times less resolution

Computational cost: splitting along dimensions is not possible in the metric algorithm, which makes resampling phase much more costly than in the splitting scheme (in 6D, a factor 171 for cubic B-splines) but this is more than compensated by the gain in effective resolution. In 6D, the metric method is expected to be much less costly than the splitting method with similar level of diffusion.

Extension to higher number of dimensions and parallel programming: extension to 6D is straightforward and the metric method not more complex to parallelize than the standard splitting scheme (e.g. Crouseilles et al. 2009) even with the percolation algorithm part.

Issues: improvements exploiting splitting for efficiency cannot be trivially implemented (e.g. Discontinuous Galerkin methods).