

Semi-Lagrangian particle methods for complex fluid and fluid-structure dynamics

G.-H. Cottet
Université Grenoble Alpes

support of Agence Nationale pour la Recherche (ANR)
and Institut Universitaire de France (IUF)

In particular with:

M. Coquerelle, A. Magni, J.-B. Lagaert, J.-M. Etancelin, C. Mimeau
G. Balarac, E. Maitre, I. Mortazavi, F. Perignon, C. Picard

and close collaboration with group of P. Koumoutsakos, ETH Zurich



Methods

- Multilevel, Semi-Lagrangian particle methods
- Immersed boundary technics for fluid-structure interactions

Applications

- Turbulent transport of passive and active scalars
- Simulation and optimization of flows around complex bodies

Issues

- Accuracy
- Flexibility
- Computational efficiency and parallel scalability

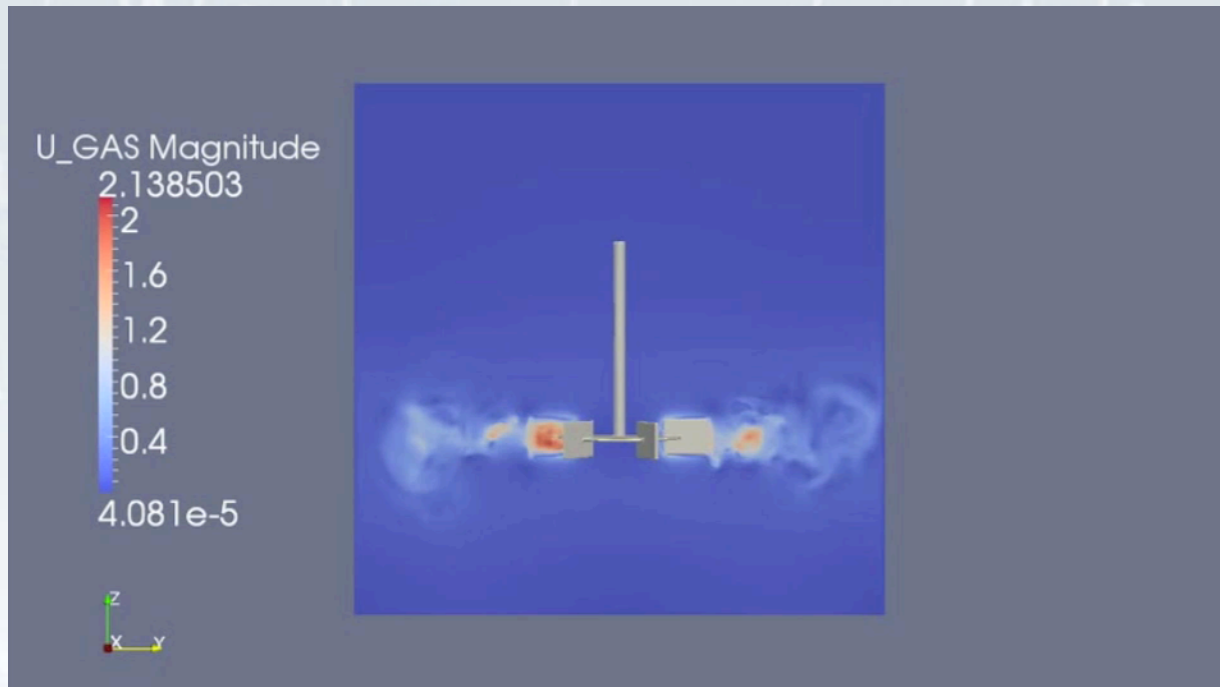


Outline

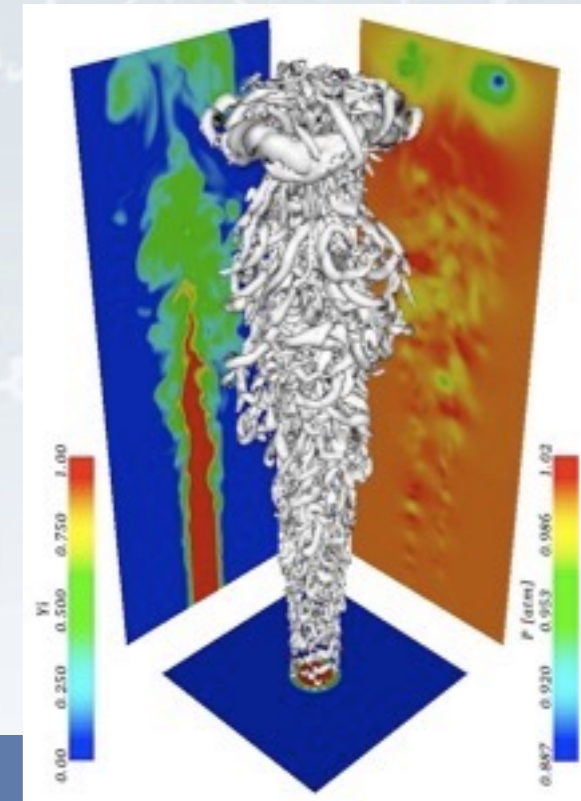
- Turbulent transport (I)
- Semi-Lagrangian particle methods
- Immersed boundary methods for fish optimization
- Multi-resolution semi-lagrangian particles
- Turbulent transport (II)
- Computational issues and hybrid computing



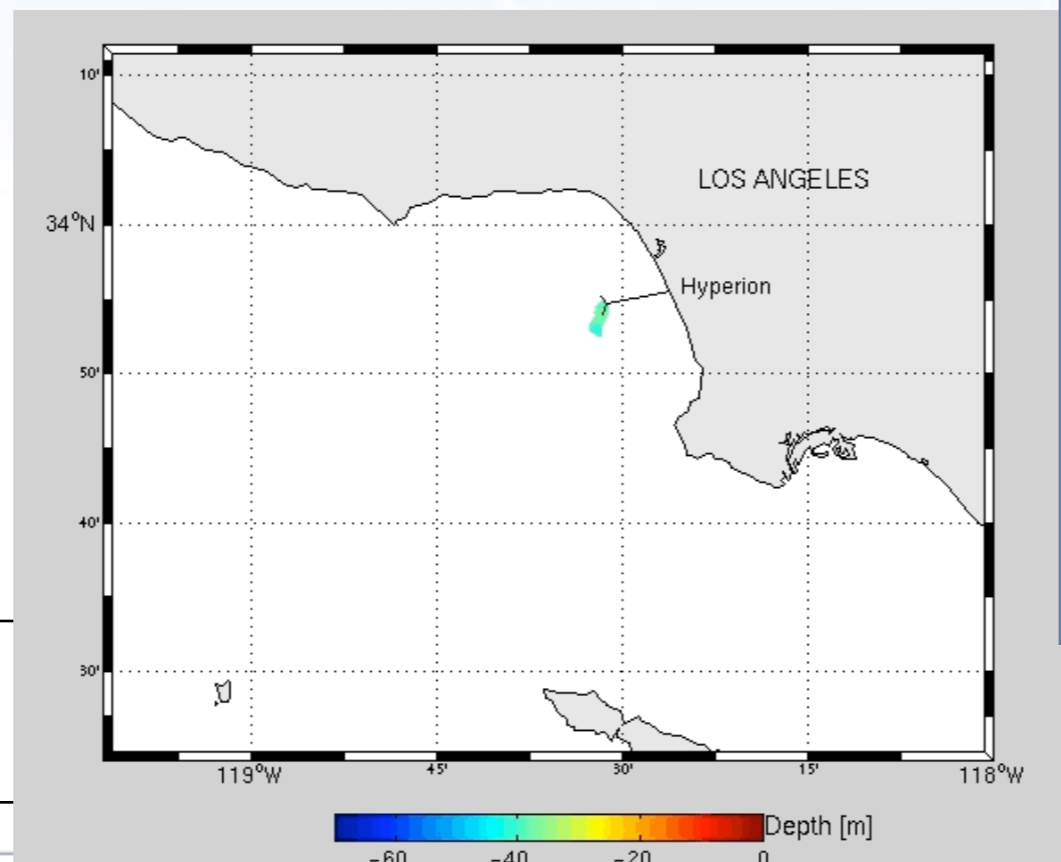
Accurate simulations of turbulent transport of scalar are of great importance in many applications



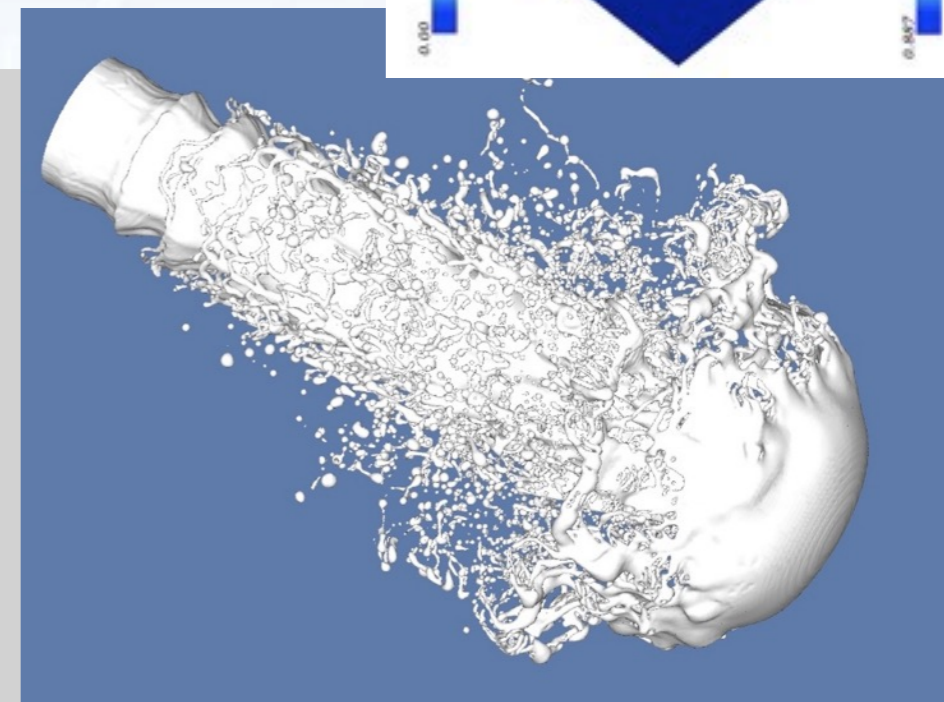
transport of species in combustion (Vervisch et al.)



mixing and polymer formation (Coria and Solvay)

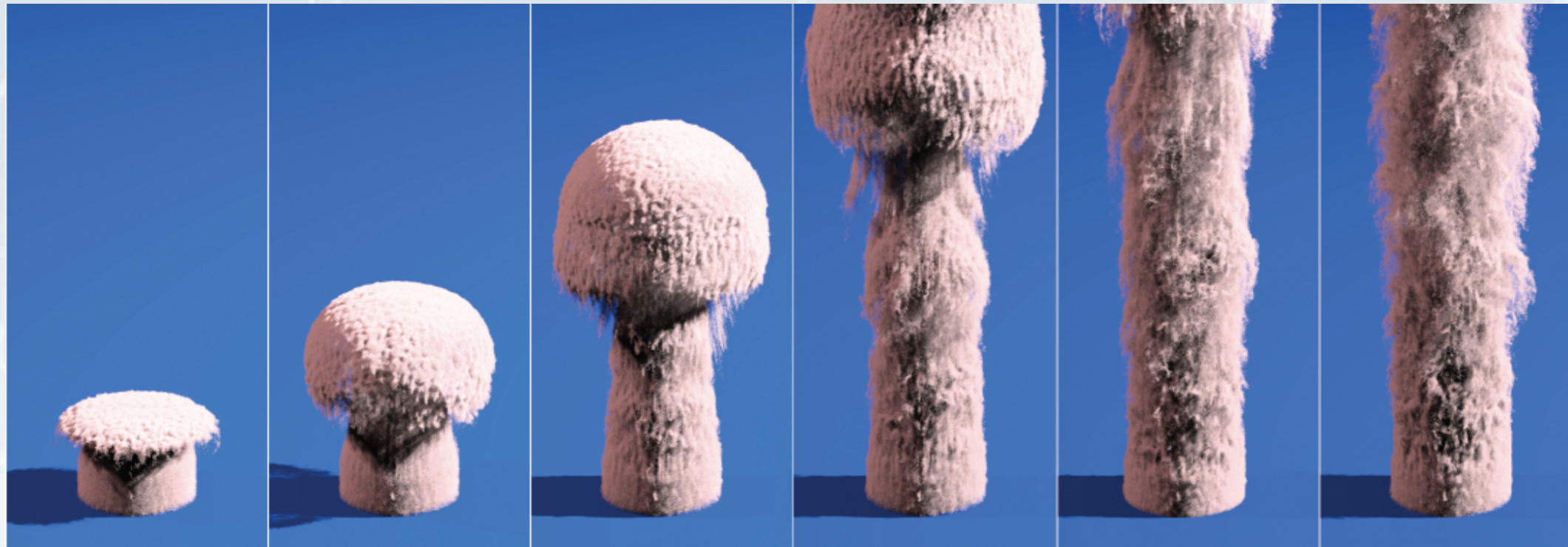


transport of pollutant in L.A. bay (Blayo et al)



transport of level set function in multiphase flows (Zaleski et al)

A 50x100x50 smoke simulation with eight turbulence bands added to synthesize an **effective resolution** of 12800x2560x12800



Kim et al, SIGGRAPH 2008

The hunt for (affordable) details
in computer graphics

rch

IN THE NEWS

- RRSP season
- Killer cats

Home World Canada Politics Business Health Arts & Entertainment Technology & Science Community

Technology & Science Quirks & Quarks Blog Photo Galleries

Tech achievement Oscar goes to Canadian software developers

The Canadian Press Posted: Jan 4, 2013 9:43 AM ET | Last Updated: Jan 4, 2013 6:04 PM ET

The team's Oscar-winning Wavelet Turbulence software is employed by visual effects artists to control the appearance of smoke and gas on film. It has been used in big-budget movies like Battleship. (Universal Pictures/Associated Press)

Equations of turbulent transport of a **passive** scalar:

Incompressible
Navier-Stokes equations

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \vec{\nabla} \vec{u} = \vec{\nabla} \cdot (\nu \vec{\nabla} \vec{u}) + \vec{\nabla} p, \quad \vec{\nabla} \cdot \vec{u} = 0$$

Transport equation

$$\frac{\partial \theta}{\partial t} + \vec{u} \cdot \vec{\nabla} \theta = \vec{\nabla} \cdot (\kappa \vec{\nabla} \theta)$$

with appropriate boundary conditions

The ratio of fluid viscosity to scalar diffusivity is called the **Schmidt number** : $S_c = \nu / \kappa$

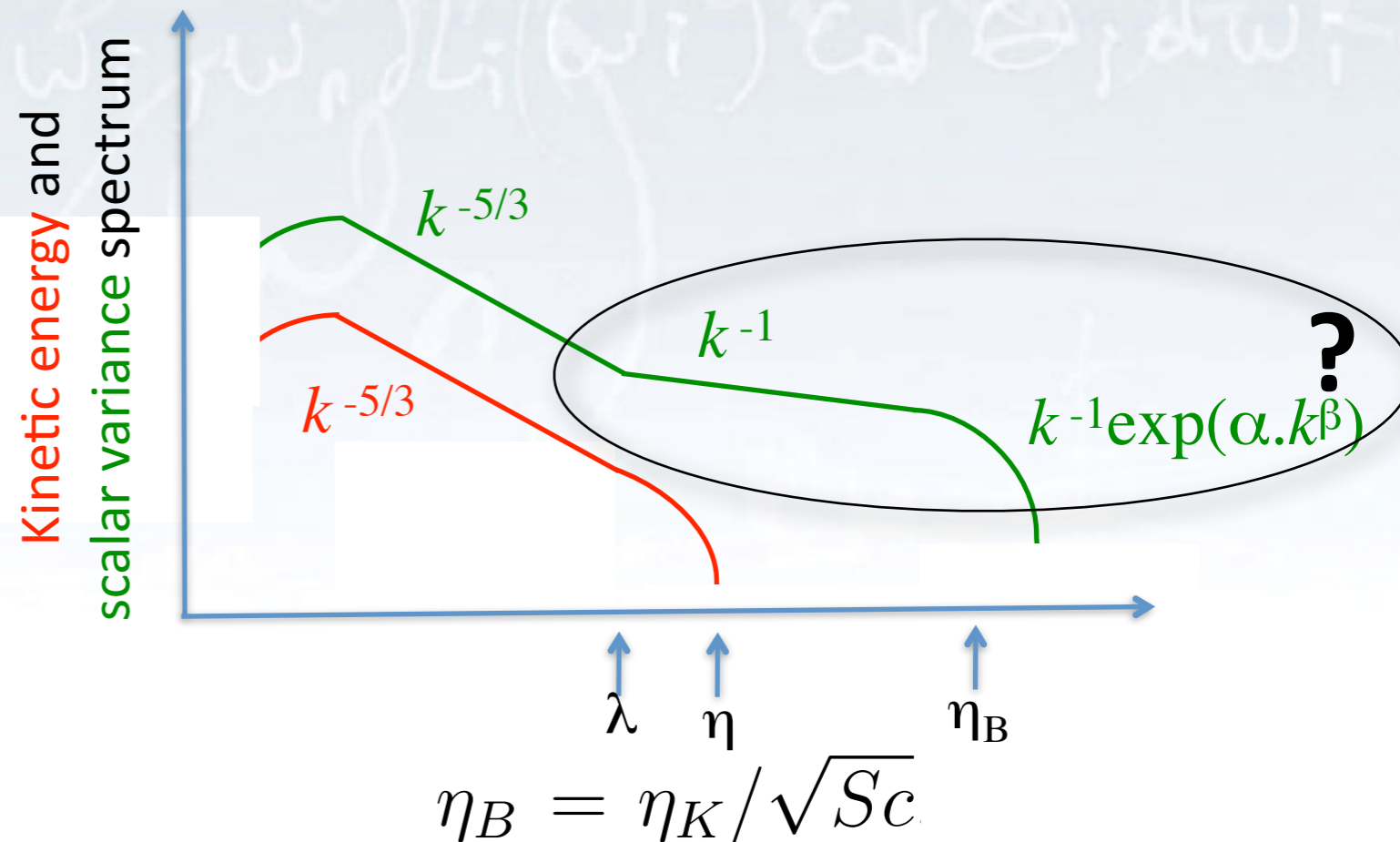
Challenge in DNS of turbulent transport:

Observation and theoretical prediction:

depending on the value of Sc , energetic scales for scalar can extend far beyond those of the momentum (the Kolmogorov scale), with a «slow» k^{-1} decay

Challenging numerical question:

- ✓ How to resolve in an affordable way the significant scales in the scalar
- ✓ Scaling of the scalar spectrum at the smallest scales?



Natural idea: decouple spatial resolutions for momentum and for scalar

$$\Delta x^\theta \simeq \frac{\Delta x^u}{\sqrt{Sc}} \Rightarrow N^\theta \simeq \sqrt{Sc} N^u$$

which solvers to choose for Navier-Stokes and transport equations ?

➡ Turbulent flow :

Spectral (periodic «academic» geometries) or Finite Volume (complex «engineering» geometries)

➡ Scalar equation:

we need a method which

- is conservative
- is high order accurate (spectral-like)
- has good parallel scalability
- limits as much as possible the computational overhead

-> choice of Lagrangian or Semi-Lagrangian method :

- well adapted to transport equations
- time-step will not be constrained by very fine scalar mesh size

Particle methods with remeshing at every time-step can be viewed and analyzed as **forward semi-lagrangian methods**

How they work:

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

In 1D method can be described by the following equations:

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

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

Γ is defined by regularity and moment properties :

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

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

★ 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 $\Delta t \leq \|\vec{\nabla} \vec{u}\|_\infty^{-1}$

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

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

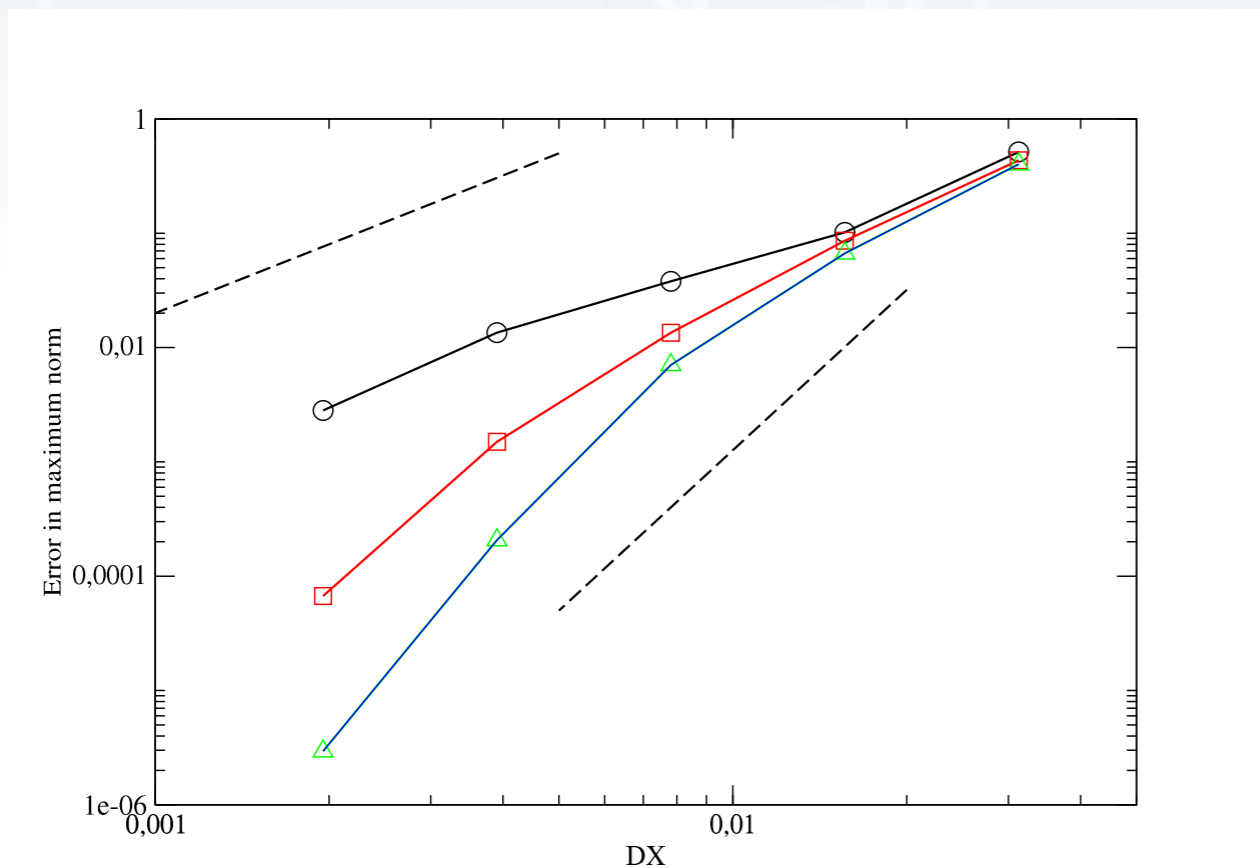
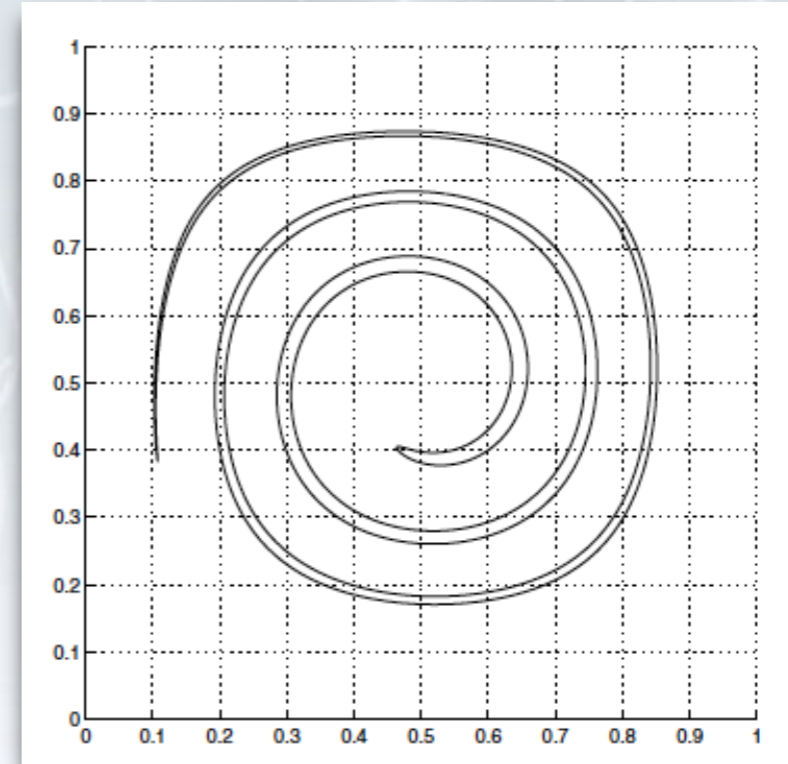
$$\Lambda_{4,2}(x) = \begin{cases} 1 - \frac{5}{4}|x|^2 - \frac{35}{12}|x|^3 + \frac{21}{4}|x|^4 - \frac{25}{12}|x|^5 & 0 \leq |x| < 1 \\ -4 + \frac{75}{4}|x| - \frac{245}{8}|x|^2 + \frac{545}{24}|x|^3 - \frac{63}{8}|x|^4 + \frac{25}{24}|x|^5 & 1 \leq |x| < 2 \\ 18 - \frac{153}{4}|x| + \frac{255}{8}|x|^2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

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



Accuracy assessment : case of a rotating patch in an off-center vorticity field

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

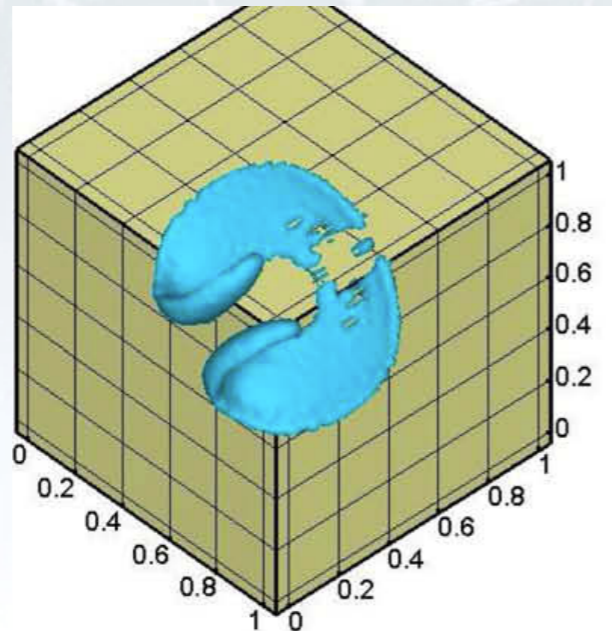
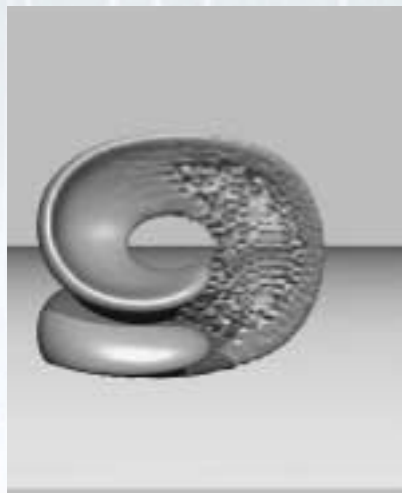


Kernel	Order of convergence
$\Lambda_{2,1}$	1.87
$\Lambda_{4,2}$	3.17
$\Lambda_{6,4}$	5.92

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

3D case : comparisons with Weno and VOF methods

Implementation of grid-based methods
with particles for corrections



Enright et al, JCP 2002

3rd order Weno
N=100 + 64 ppc
CFL=1 (?)
CPU =??

Vincent et al, JCP 2010

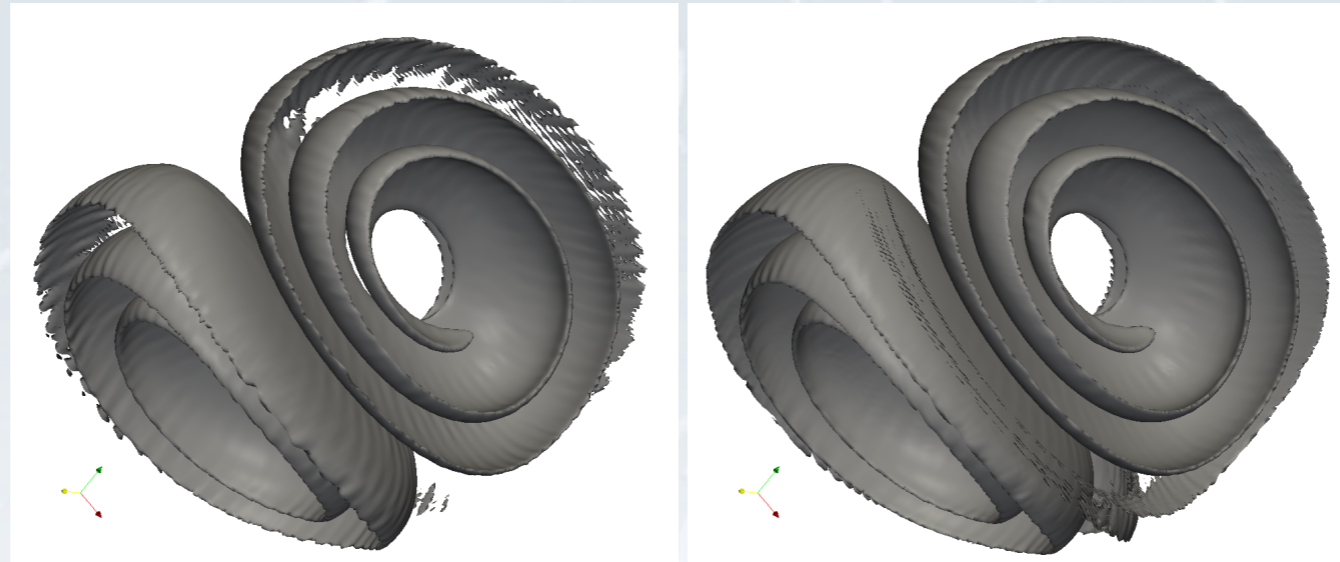
VOF
N=64 + 9 ppc
CFL=0.1

N=100, CFL=8

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

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



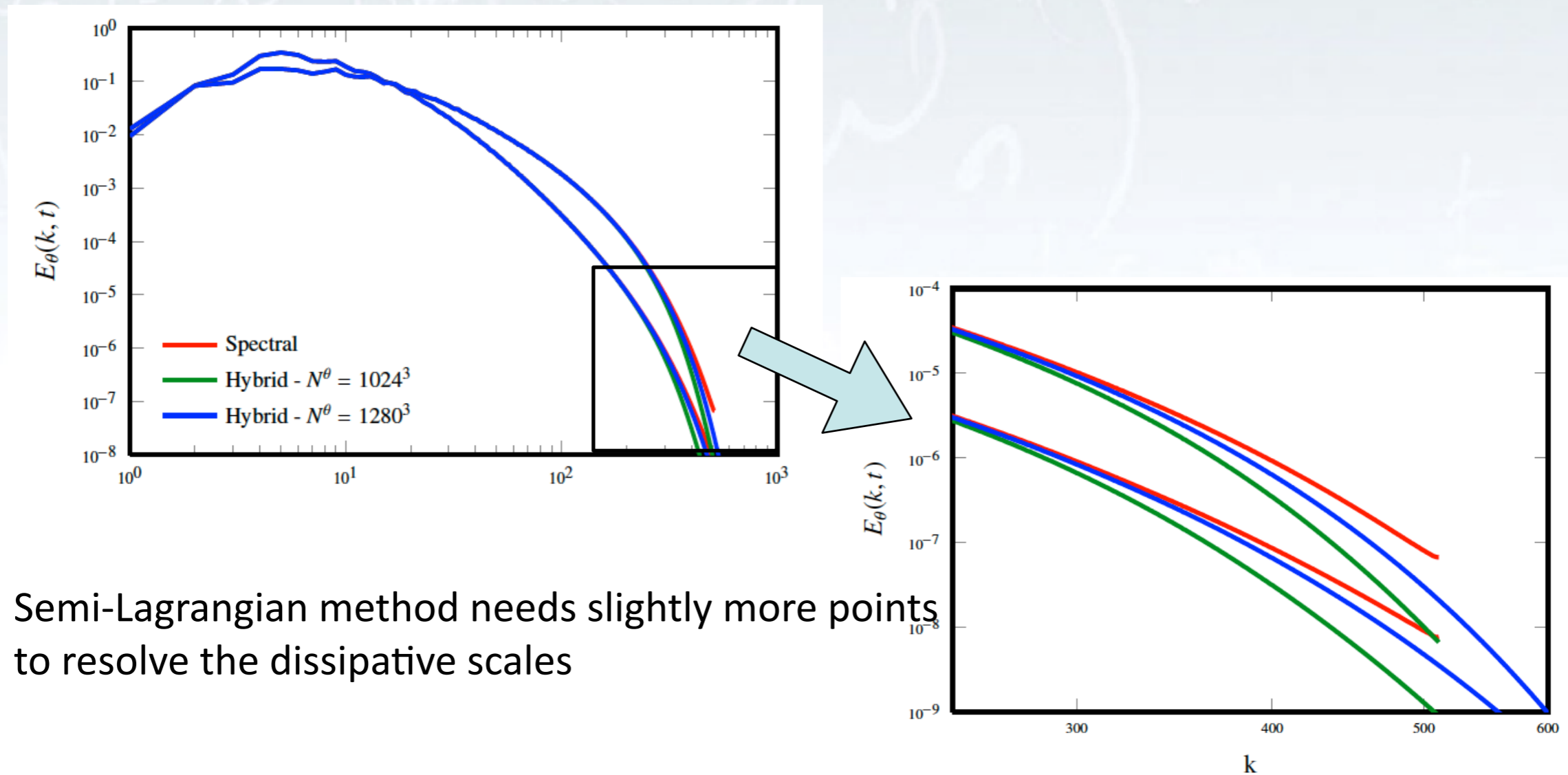


$N = 256$ and $CFL = 30$. Left picture: kernel $\Lambda_{2,1}$; right picture: kernel $\Lambda_{8,4}$.

Illustration of accuracy of the Semi-Lagrangian method compared to a spectral method

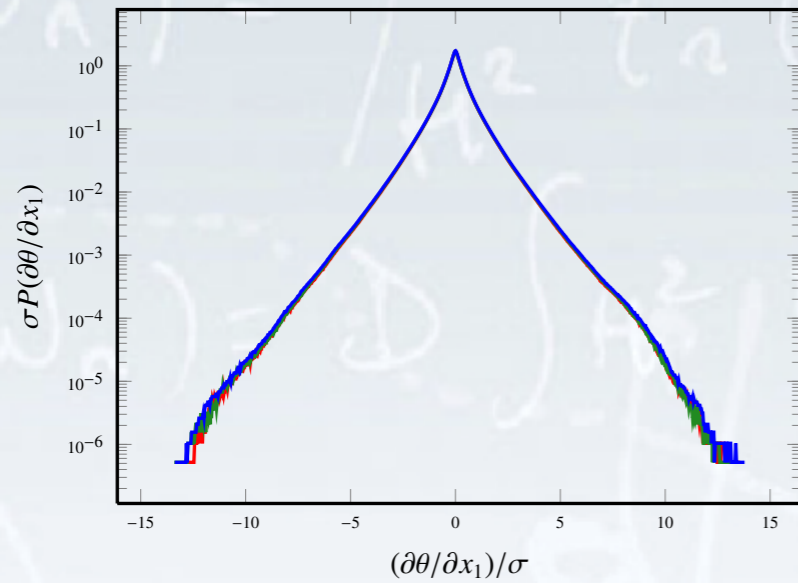
Scalar spectra given by spectral vs spectral/particle method in a decaying THI experiment with $Sc=50$, $N_u=256$, $N_\theta=1024$

Particle method uses second order kernel

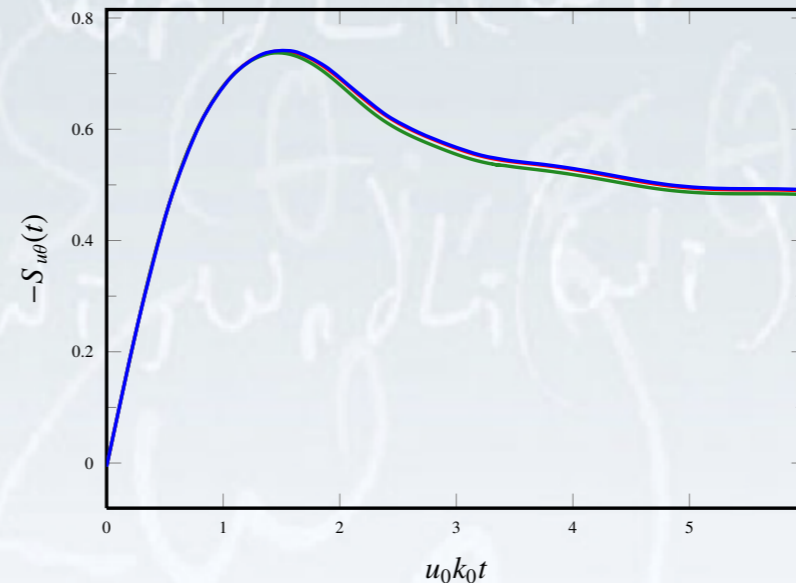


Semi-Lagrangian method needs slightly more points to resolve the dissipative scales

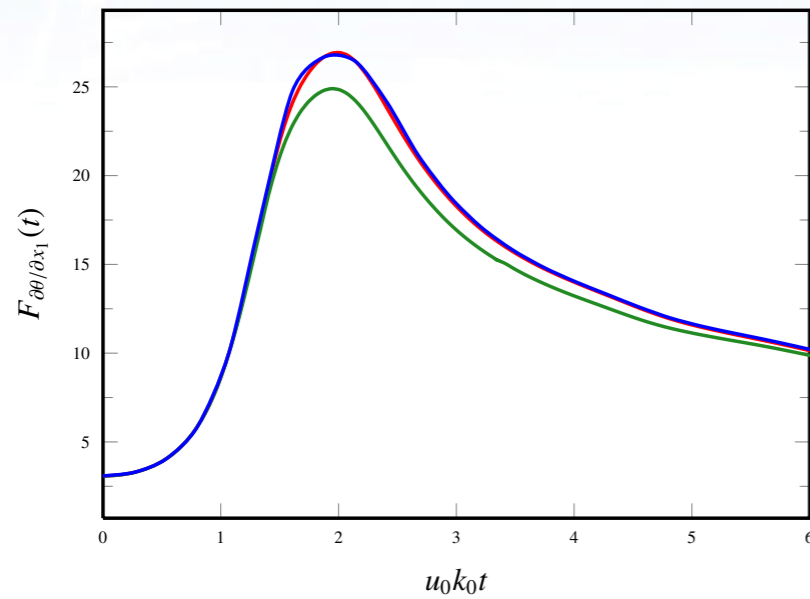
Additional diagnostics



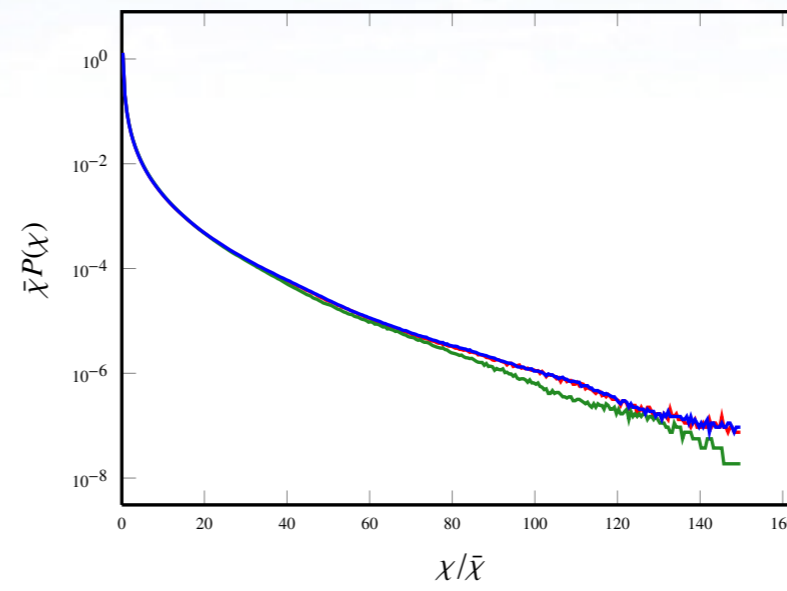
pdf of scalar gradient



time evolution of scalar skewness



time evolution of scalar flatness



pdf of scalar dissipation

Forced homogeneous turbulence:

Systematic quantitative study of spectra at **large, intermediate** and **small** scales for a wide range of Reynolds and Schmidt numbers

Samples of simulations performed during the CTR Summer Program 2012 (on up to 8000 cores with MPI)

- Ratio of scalar/momentum resolution roughly given by $Sc^{1/2}$

- Stability condition is independent of Δx^Z

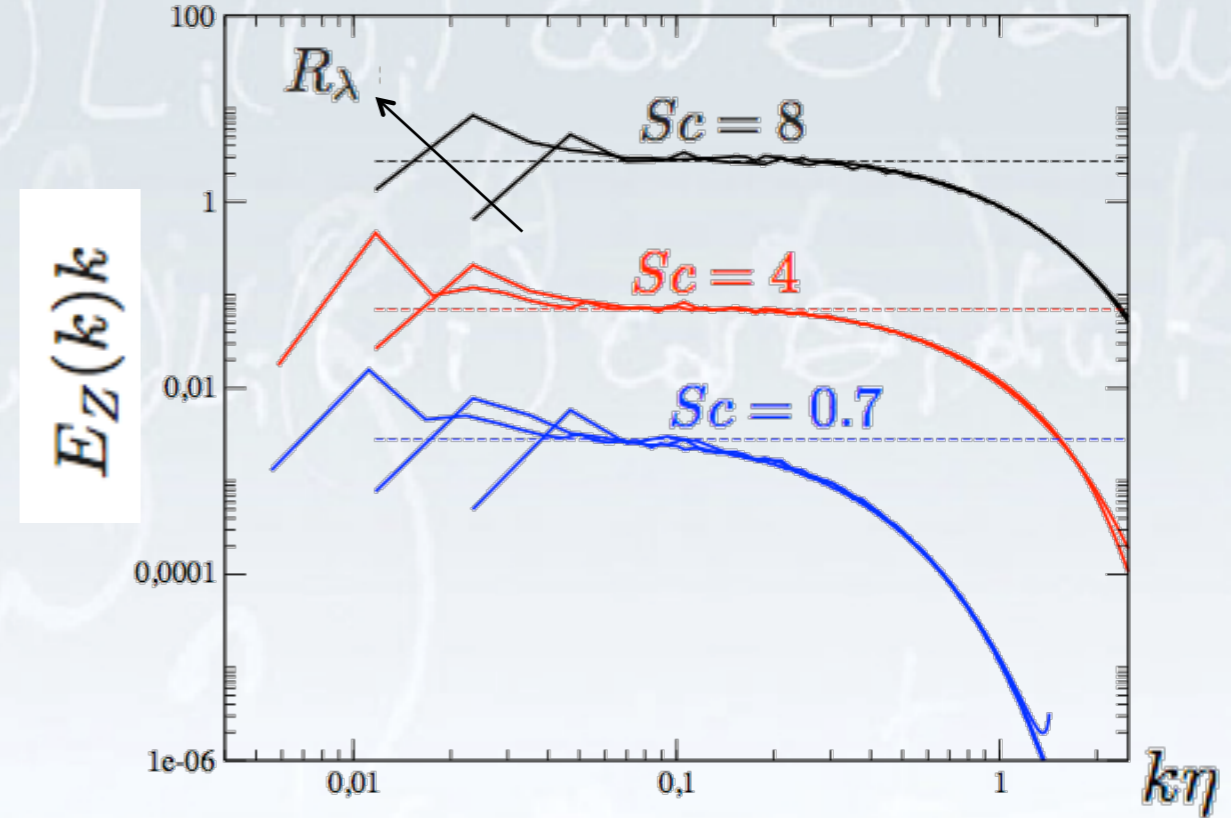
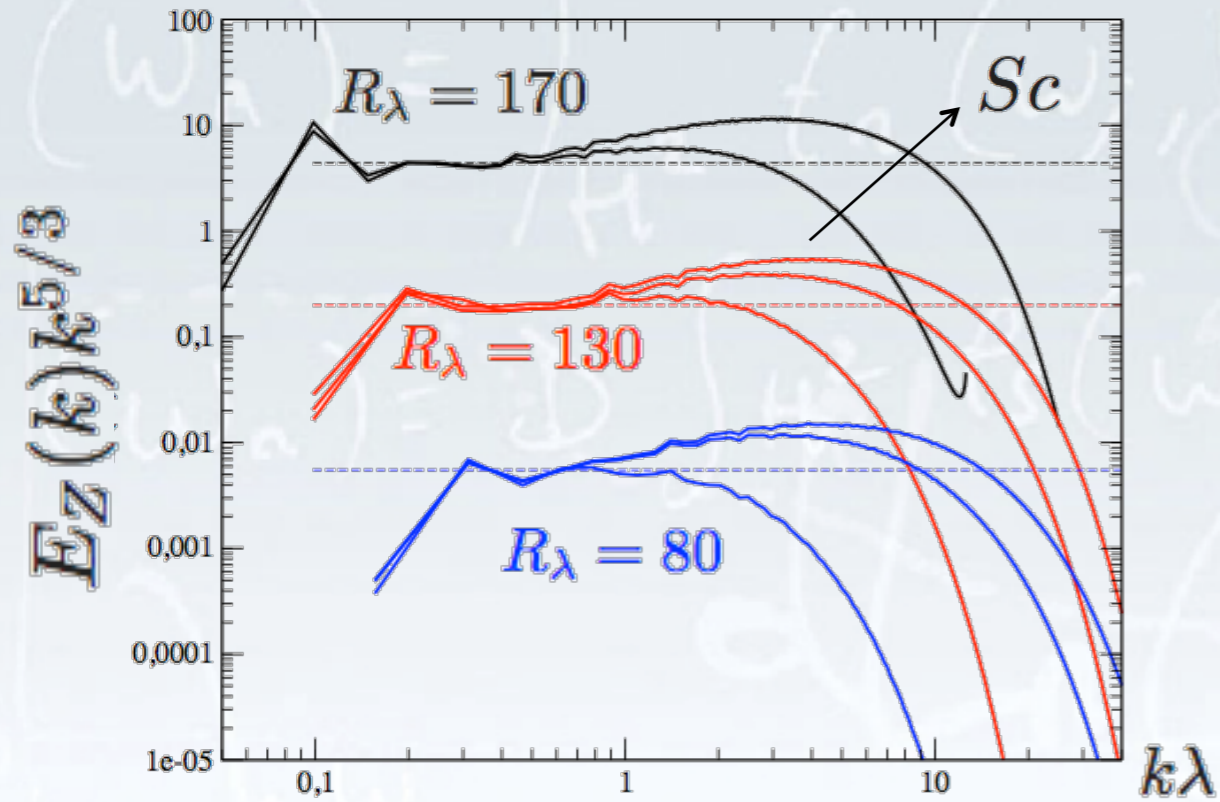
- Case $R_\lambda = 80$ and $Sc = 16$

established mixing dynamic takes **18h instead of ≈ 12 days** with full spectral method.

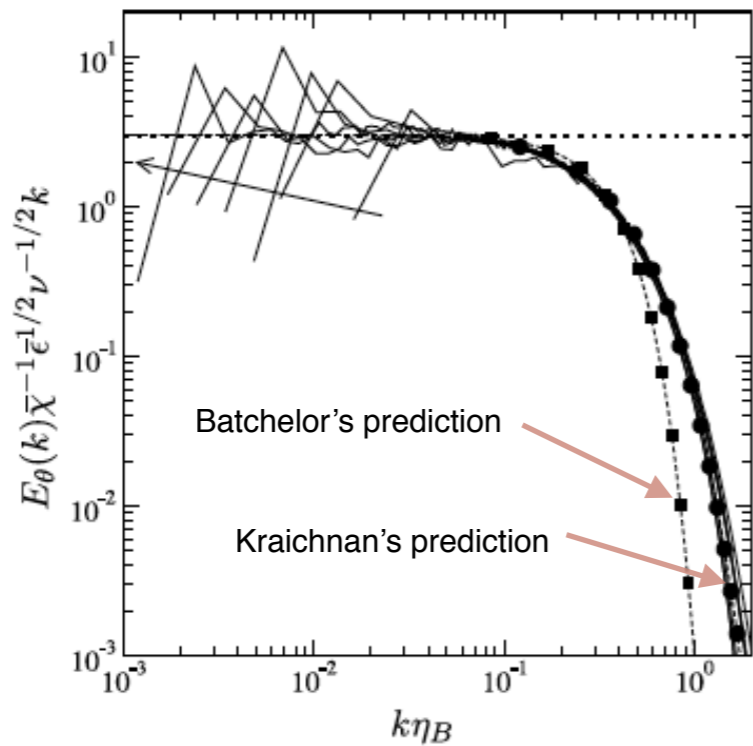
R_λ	N^u	$K_{\max}^u \eta_K$	Δt^u	Sc	N^θ	$K_{\max}^\theta \eta_B$	Δt^θ	$\Delta t_{\text{spec}}^\theta$
				0.7	512^3	-		$6e^{-3}$
				4	1024^3	3.39		$3e^{-3}$
				8	1024^3	2.45		$3e^{-3}$
130	256^3	1.73	$1.2e^{-2}$	16	1536^3	2.61	$8.6e^{-2}$	$2e^{-3}$
				32	1536^3	1.85		$2e^{-3}$
				64	2048^3	1.76		$1.5e^{-3}$
				128	3064^3	1.79		$1e^{-3}$
210	512^3	1.79	$3e^{-3}$	0.7	770^3	-	$2e^{-2}$	$2e^{-3}$
				4	1024^3	1.76		$1.5e^{-3}$



Scaling of the large and intermediate scales



Scaling of smallest scales



Scaling at large and intermediate scales

Depending on Reynolds and Schmidt numbers, $k^{-5/3}$ or k^{-1} or both scaling found in accordance with theoretical predictions

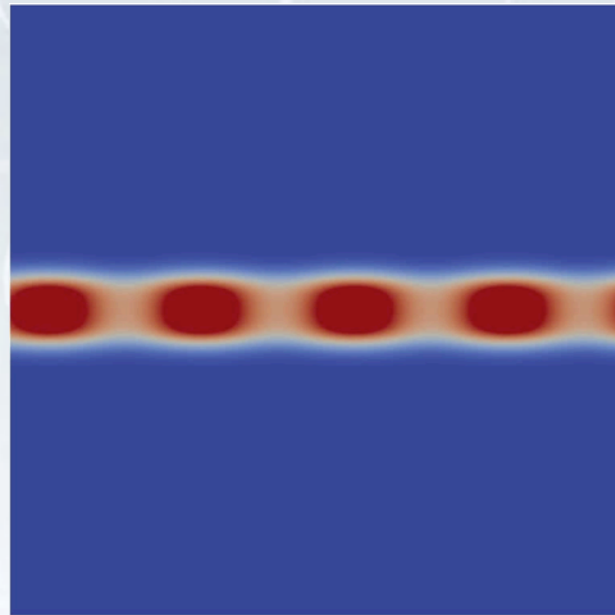
Scaling at the smallest scales

Numerical results in good agreement with Kraichnan's theory: $k^{-1} \exp(\alpha.k)$

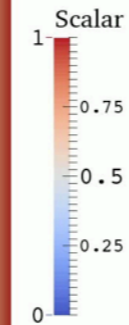
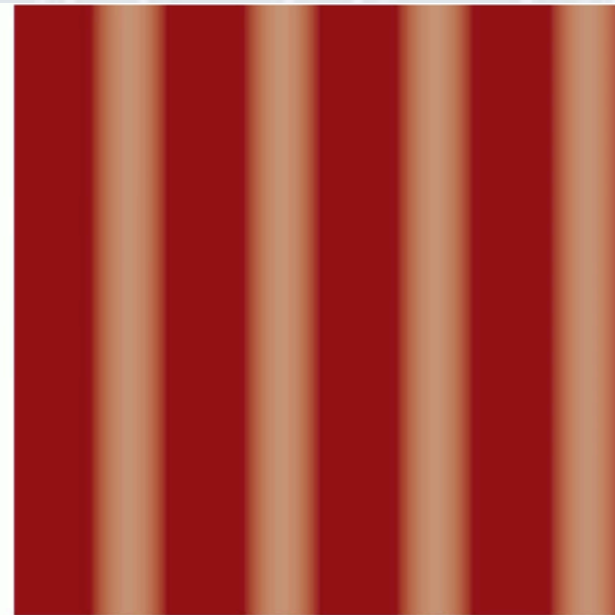
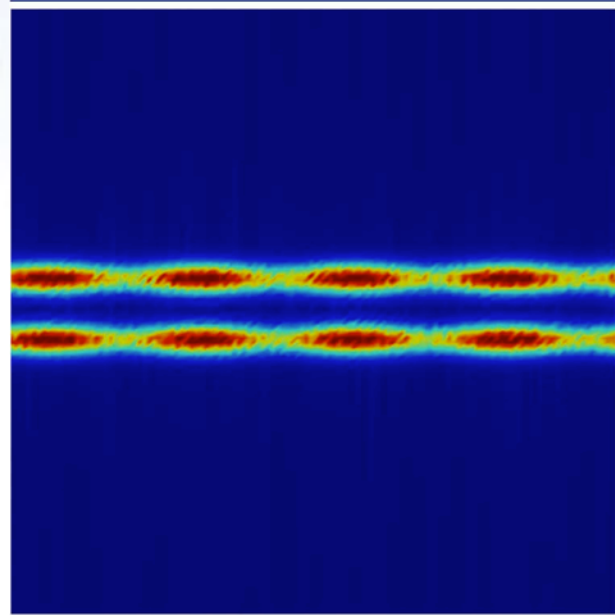
Example of jet simulation using **one billion** particles for the scalar on 8 GPUs at **1 fps**

Re=10⁴, Sc=10

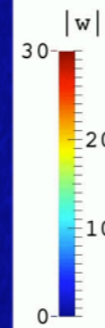
scalar



vorticity

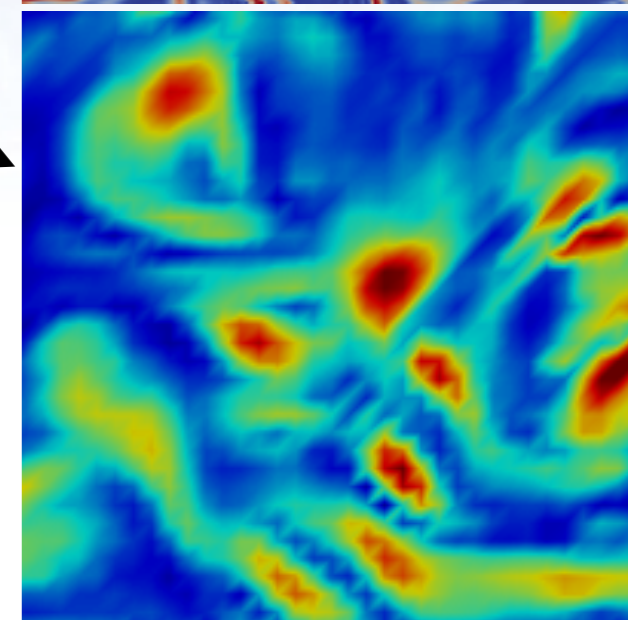
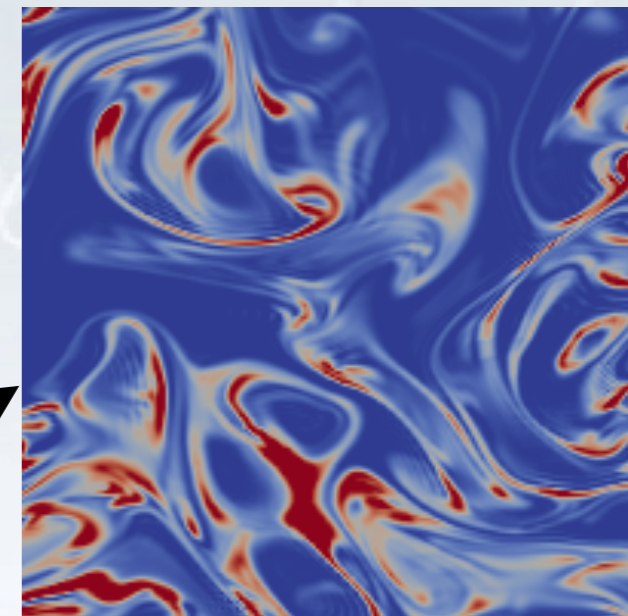


t=0.000



t=0.000

zoom of scalar and vorticity showing the different scales



side view

top view



Handling complex geometries and Fluid-Structure Interactions with Immersed boundary technics



Regular (Cartesian) grid with your (favorite) Navier-Stokes solver

elastic bodies

No-slip boundary conditions

Elastic stresses
Interfaces

S

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p - \nu(\phi) \Delta \mathbf{u} = \mathbf{F}(u, \phi, \nabla \phi)$$

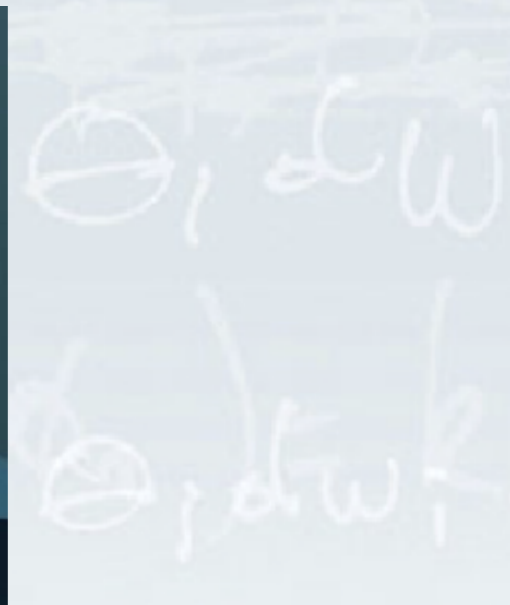
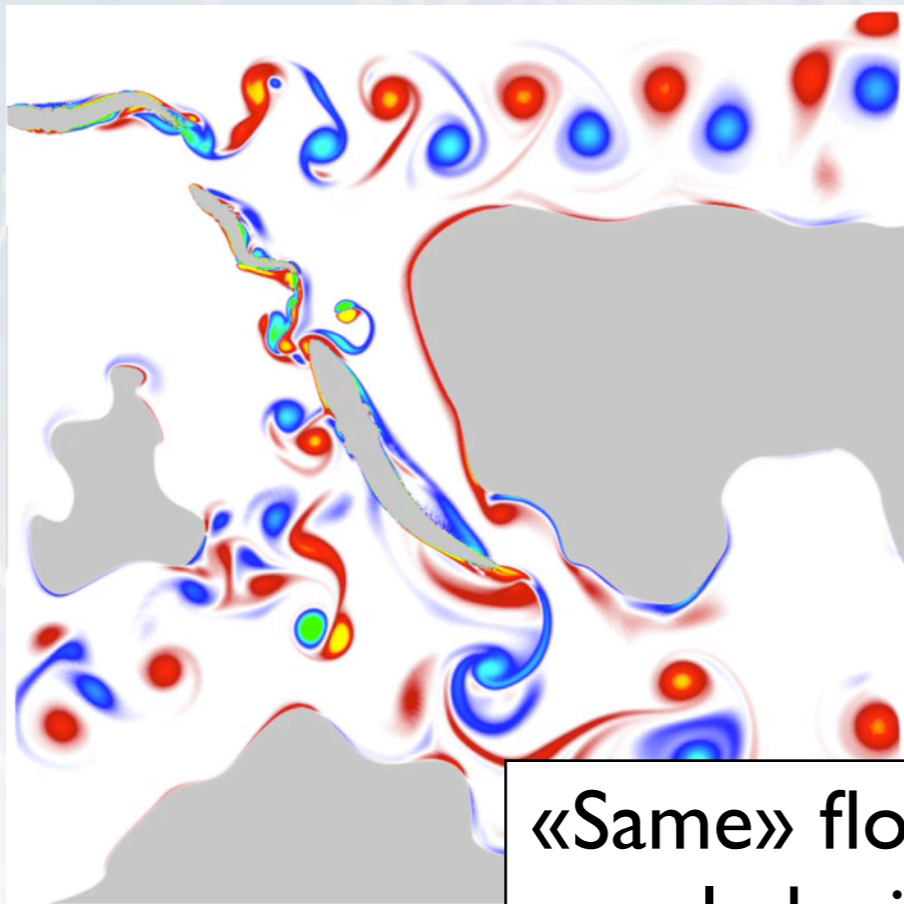
complex

$$\frac{\partial \phi}{\partial t} + (\mathbf{u} \cdot \nabla) \phi = 0$$

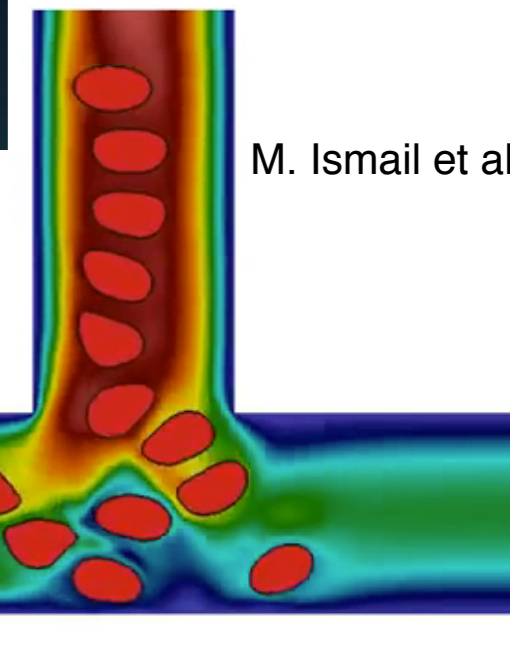
rigid bodies

collision/contact

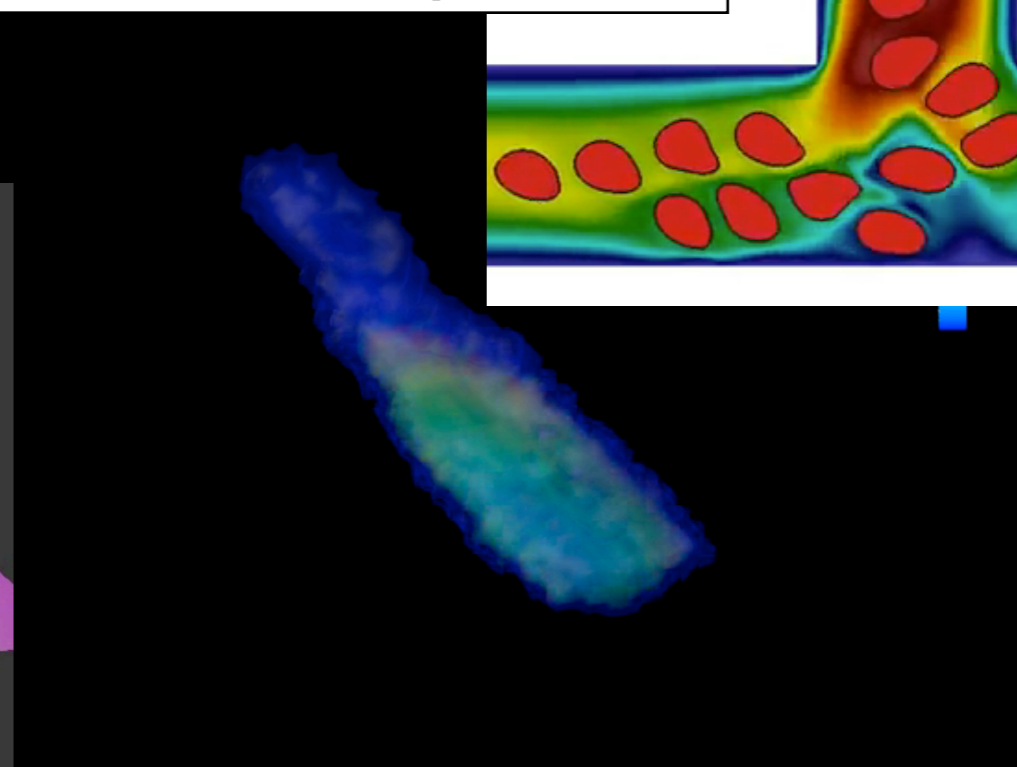
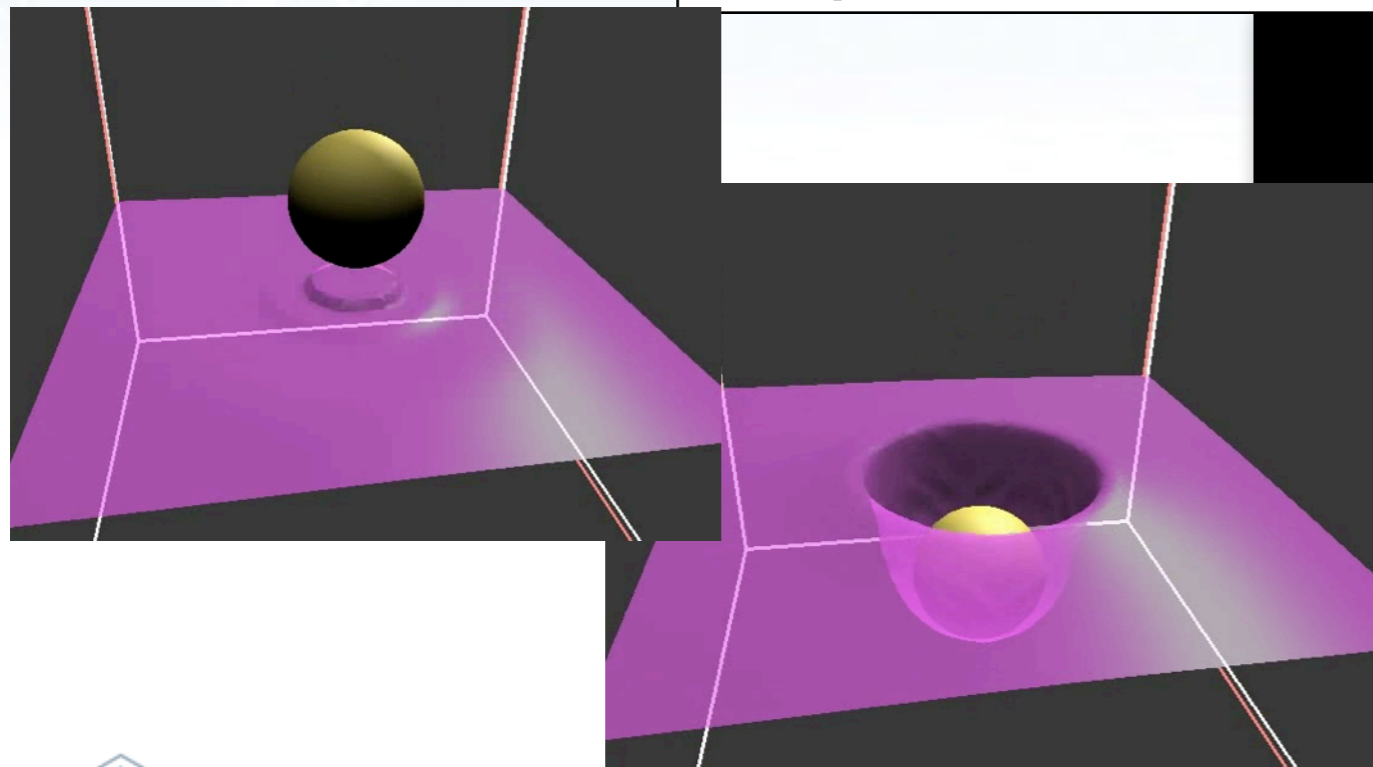




«Same» flow solver on a cartesian grid coupled with a few advection equations

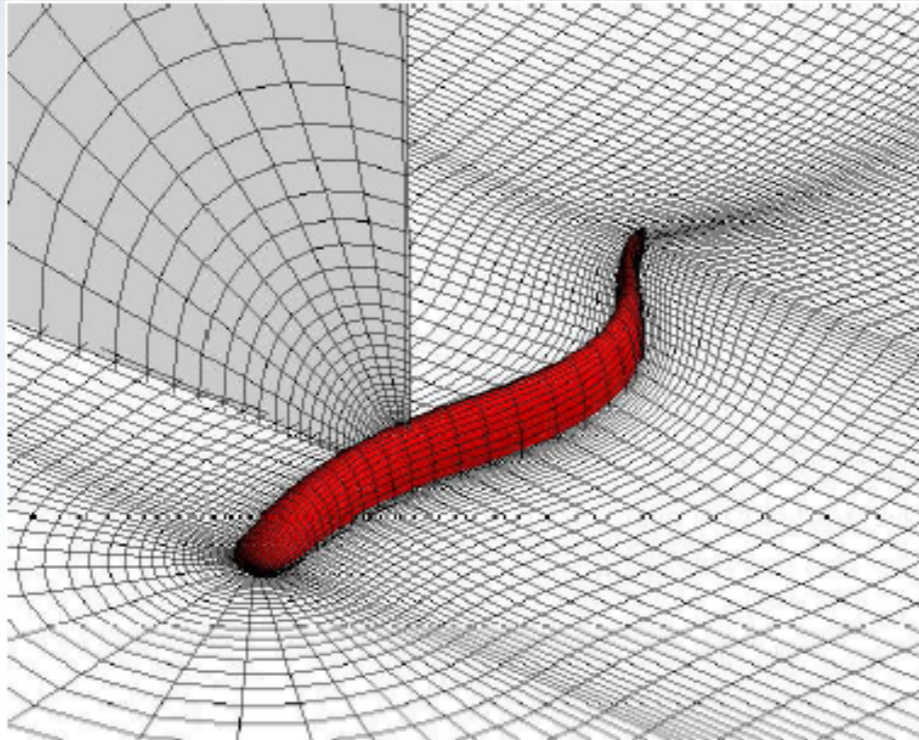


M. Ismail et al.

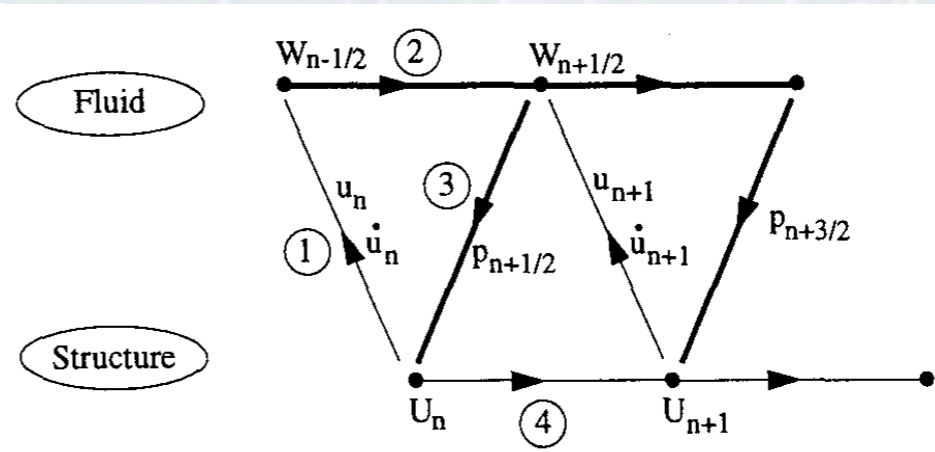


Application to the 3D simulation and optimization of swimming

Conventional approach would use body-fitted FV methods,
ALE mesh generation and coupling boundary conditions between fluid and body



Body-fitted mesh generation



Fluid-solid interface coupling and interface conditions

Time consuming and difficult to extend to several bodies

Alternative approach based on penalization method in cartesian grids

I) Penalization method for one-way interaction [Angot-Bruneau-Fabrie 1999]

- S the rigid body at a given time, subject to gravity forces, embedded in a computational box Ω
- \bar{u} = prescribed body velocity,
- $\chi_S = 1$ in the body, 0 outside and
- λ a (large) penalization parameter

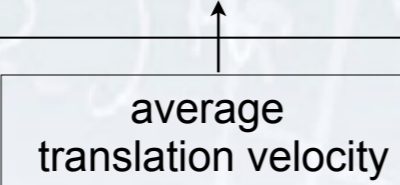
then :

$$\rho \left(\frac{\partial u}{\partial t} + (u \cdot \nabla)u \right) - \nu \Delta u + \nabla p = \rho g + \lambda \rho \chi_S (\bar{u} - u) \quad \text{in the computational box } \Omega$$

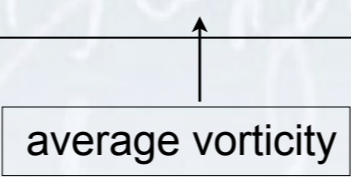
+ boundary conditions (e.g. periodic)

2) extension to two-way interaction [Coquerelle, C. 2008; Bost, C., Maitre 2009]: computation of solid velocities not prescribed but obtained by projection of flow velocity onto rigid body velocity:

$$\bar{u} = \frac{1}{|S|} \int_K \chi_S u \, dx + \left(J^{-1} \int_K \chi_S u \times (x - x_G) \, dx \right) \times (x - x_G)$$



average
translation velocity



average vorticity

where J is the inertia matrix of the body K , and x_G is its mass center.

S is captured by a level set function, advected with the rigid motion \bar{u} :

$$\frac{\partial \phi}{\partial t} + (\bar{u} \cdot \nabla) \phi = 0$$

In practice: solid motion allows to compute ϕ analytically

3) In a vorticity formulation, leads to

$$\frac{\partial \omega}{\partial t} + (u \cdot \nabla) \omega - (\omega \cdot \nabla) u - \nu \Delta \omega = + \lambda \nabla \times \chi_S (\bar{u} - u).$$

complemented by the usual system giving u from ω , and the coupling of \bar{u} with the flow and the contact forces

4) Extension to deforming bodies (Gazzola et al. 2011)

$$\bar{u} = u^t + u^r + u^{def}$$

translation deformation

rotation prescribed

result of body-flow interaction
→ computed from mean velocity/vorticity in the body

- Penalization methods offer flexibility and reduce computational cost, but at the expense of lower accuracy near the boundaries
- Requires to **locally refine** in adaptive manner

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

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

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

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

$$\varphi_j^l = \sum_{\mathbf{k}} H_{j,\mathbf{k}}^l \varphi_{\mathbf{k}}^{l+1}, \quad \psi_j^{l,\mu} = \sum_{\mathbf{k}} G_{j,\mathbf{k}}^{l,\mu} \varphi_{\mathbf{k}}^{l+1}$$

basis functions for **scales** spaces \mathcal{V}^l at level l and $l+1$ basis functions for **details** spaces \mathcal{W}^l at level l and $l+1$

$$\mathcal{V}^{l+1} = \mathcal{V}^l \oplus \mathcal{W}^l$$

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 combine 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 from level $l-1$, to allow consistent remeshing

$$\mathcal{B}^l = \left\{ \mathbf{k}' \mid \min_{\mathbf{k} \in \mathcal{K}_{>}^l} |\mathbf{k}' - \mathbf{k}| \leq \left\lceil \frac{1}{2} \text{supp}(\zeta) + \text{LCFL} \right\rceil \right\}$$

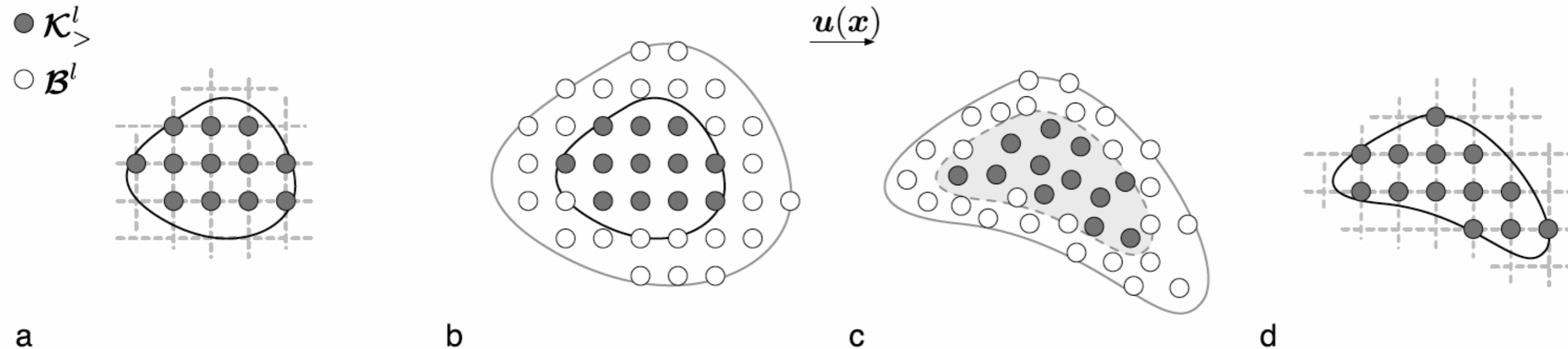
Finally, like for grid-based methods, need to allow levels $l+1$ to appear from level l during advection step.

Important: time steps given by $\text{LCFL} = 1$ are ok :

time scale on which scale $l+1$ appear from scale l is given by $dt_{l \rightarrow l+1} = \ln 2 / |\nabla u|$



Algorithm for time advancement of particles at a given level l



select «active»
particles
on the grid
(with tag=1)

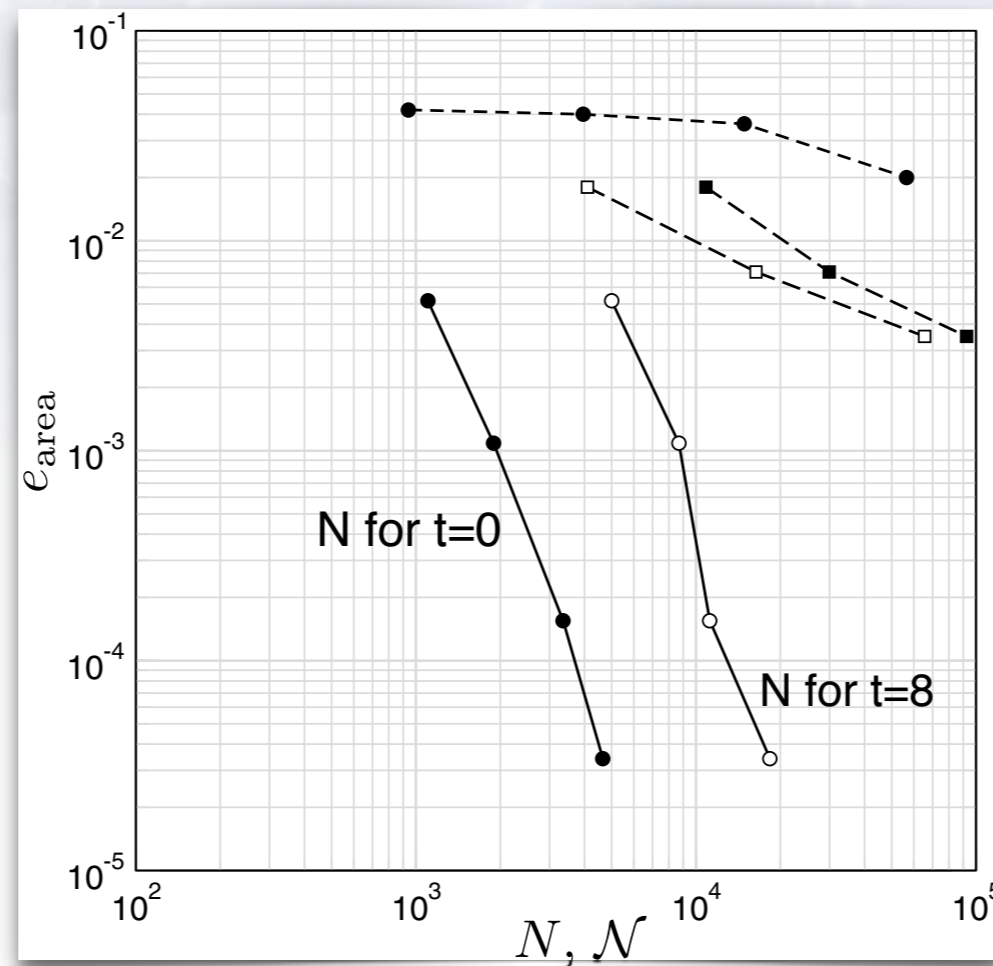
create a buffer
around these
particles
(with tag=0)

advect particles
and tag

remesh particles
and tag;
**keep particles
with tag > 0**

Back to the stretching-by-rotation patch

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



error in area enclosed by contour at time T=8

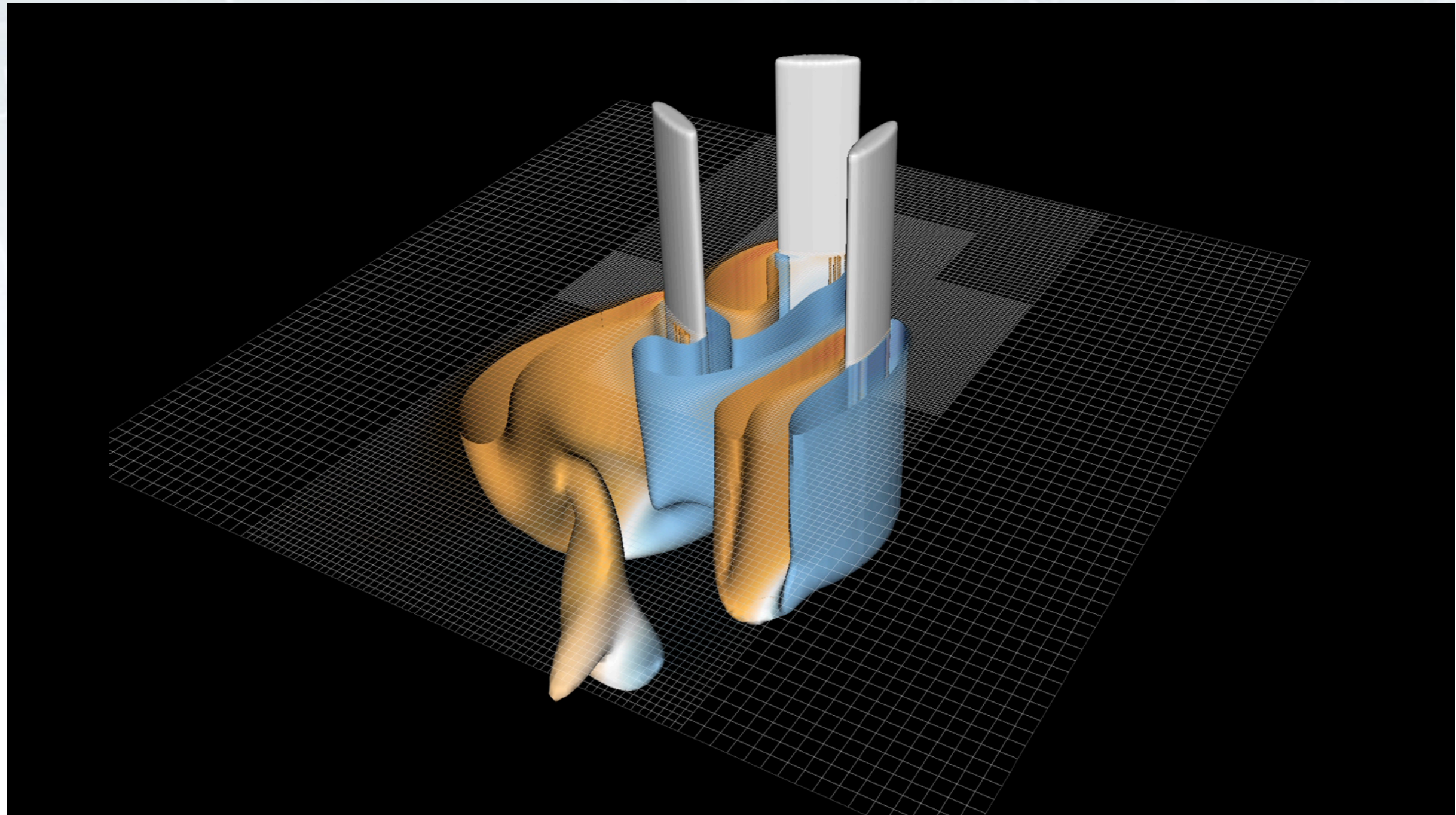
Particle level set, Enright et al. 2002

wavelet particles

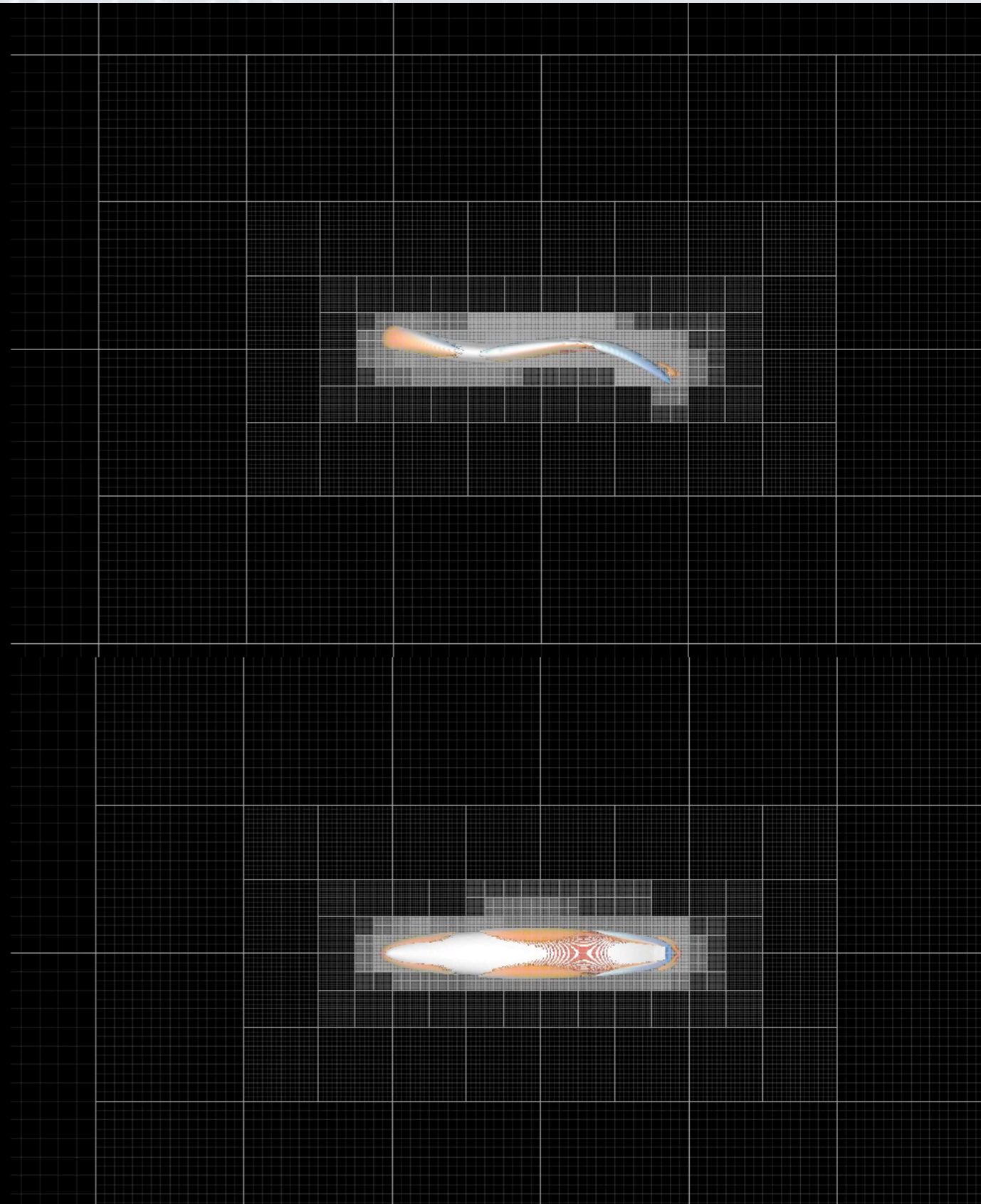
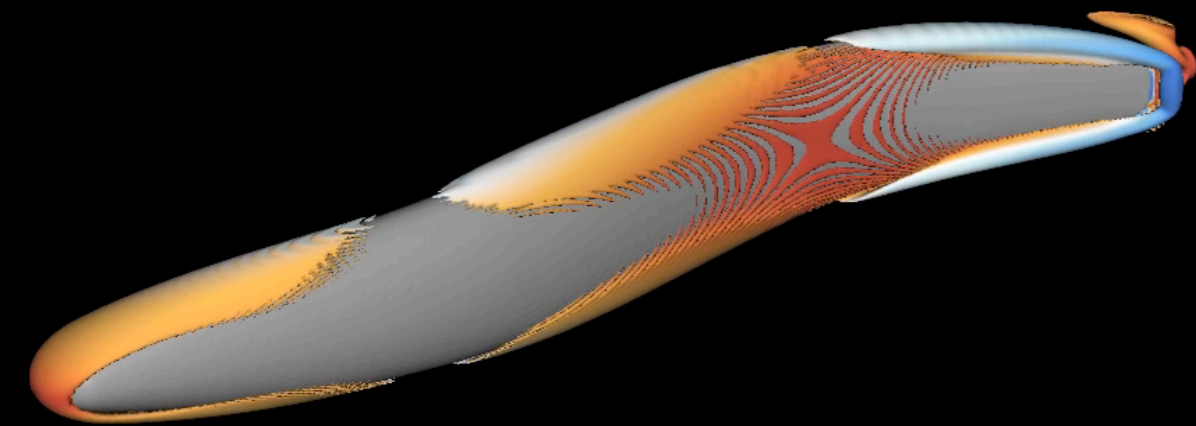
Number of active particles or grid points at time t=0 and t=8

Illustration of particle MRA 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)



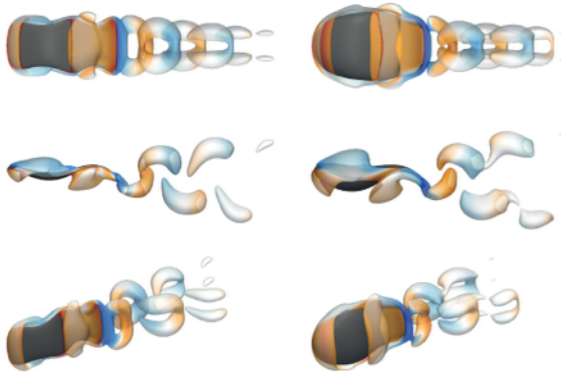
3D anguilliform swimming



Journal of Fluid Mechanics

VOLUME 722

VOLUME
722



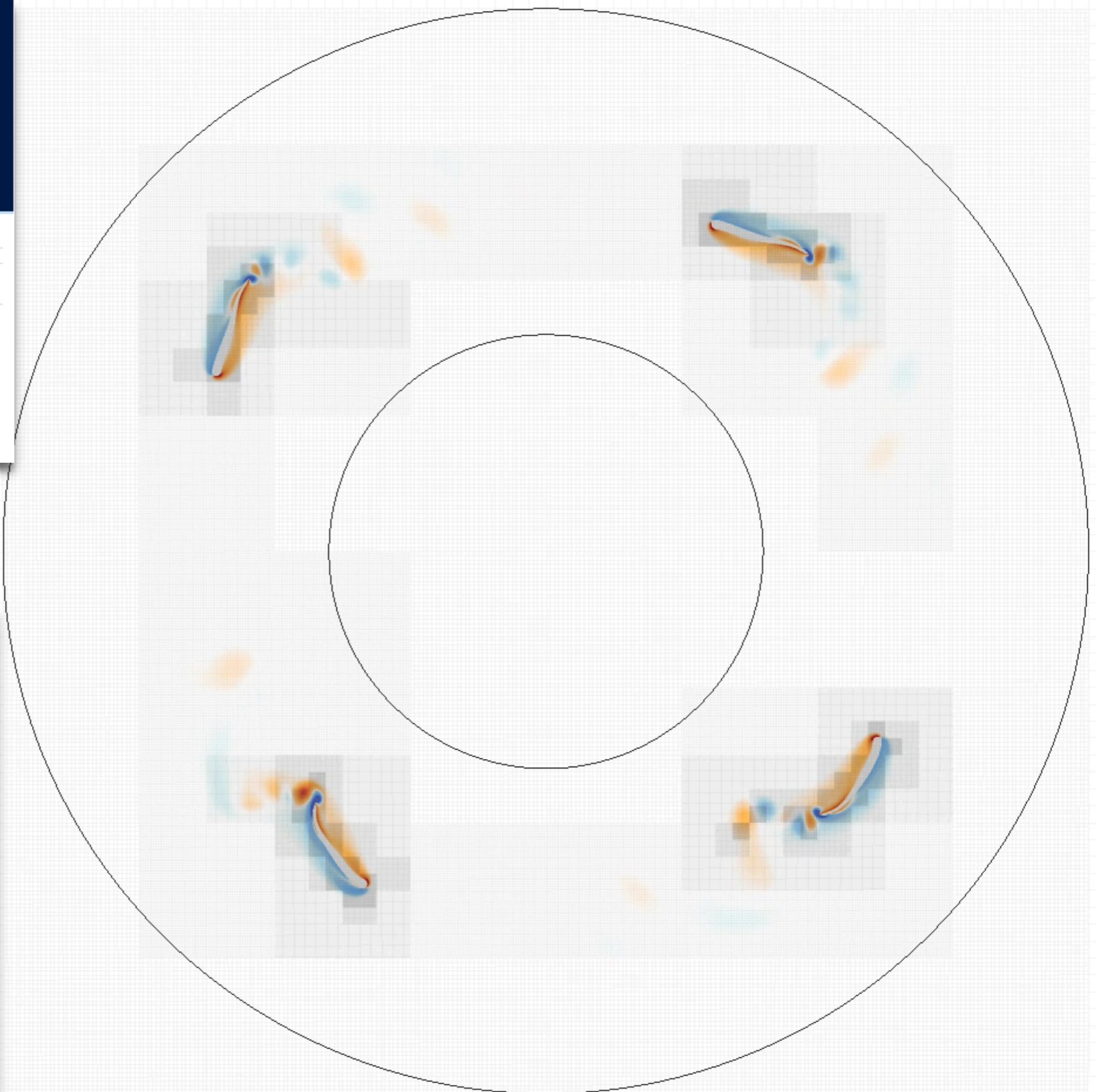
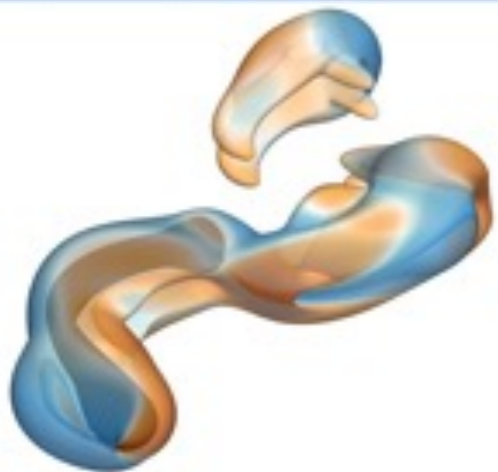
10 May
2013

CAMBRIDGE

Mattia Gazzola, Wim van Rees,
Andrew Tchieu, P. Koumoutsakos

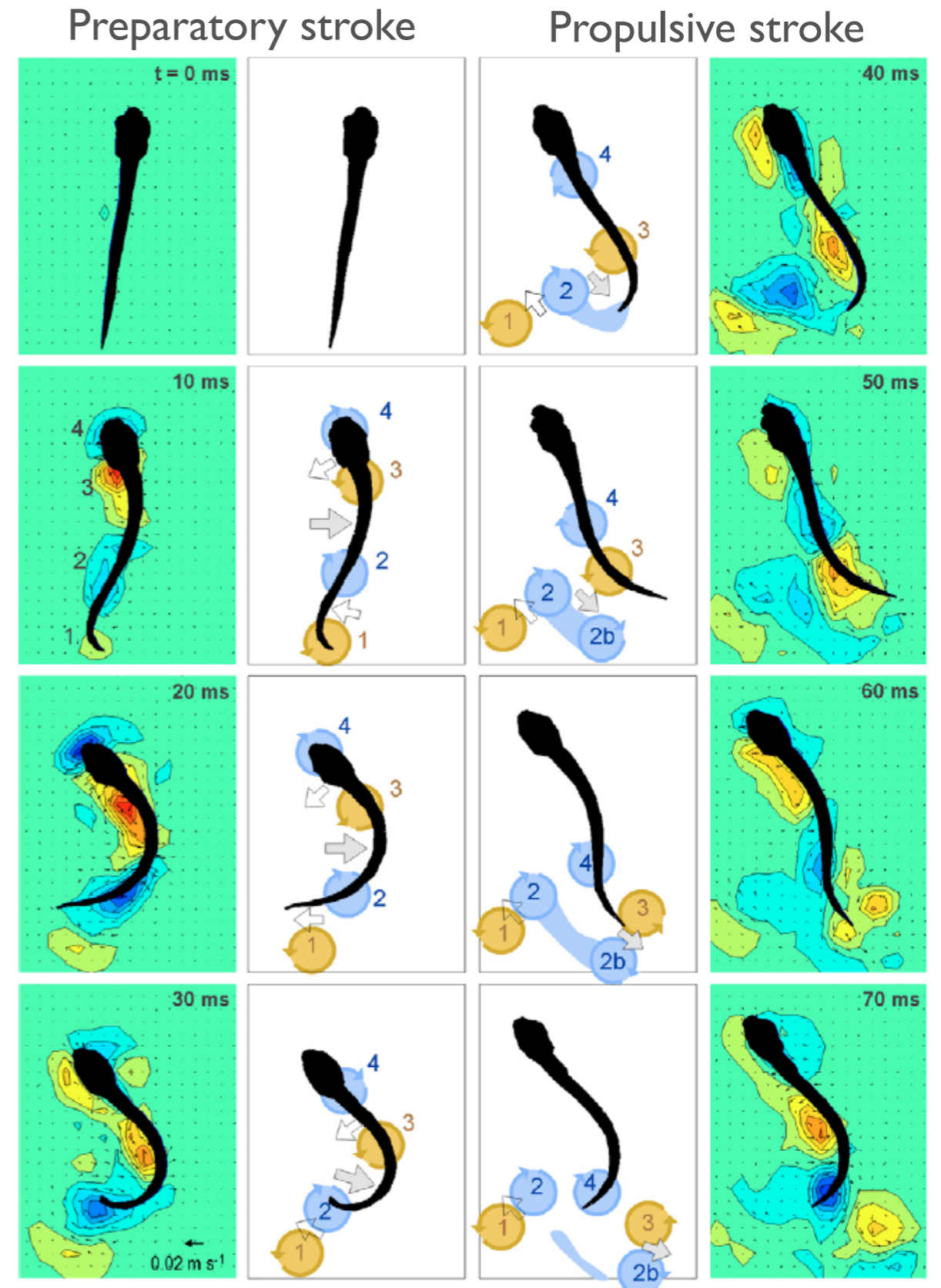
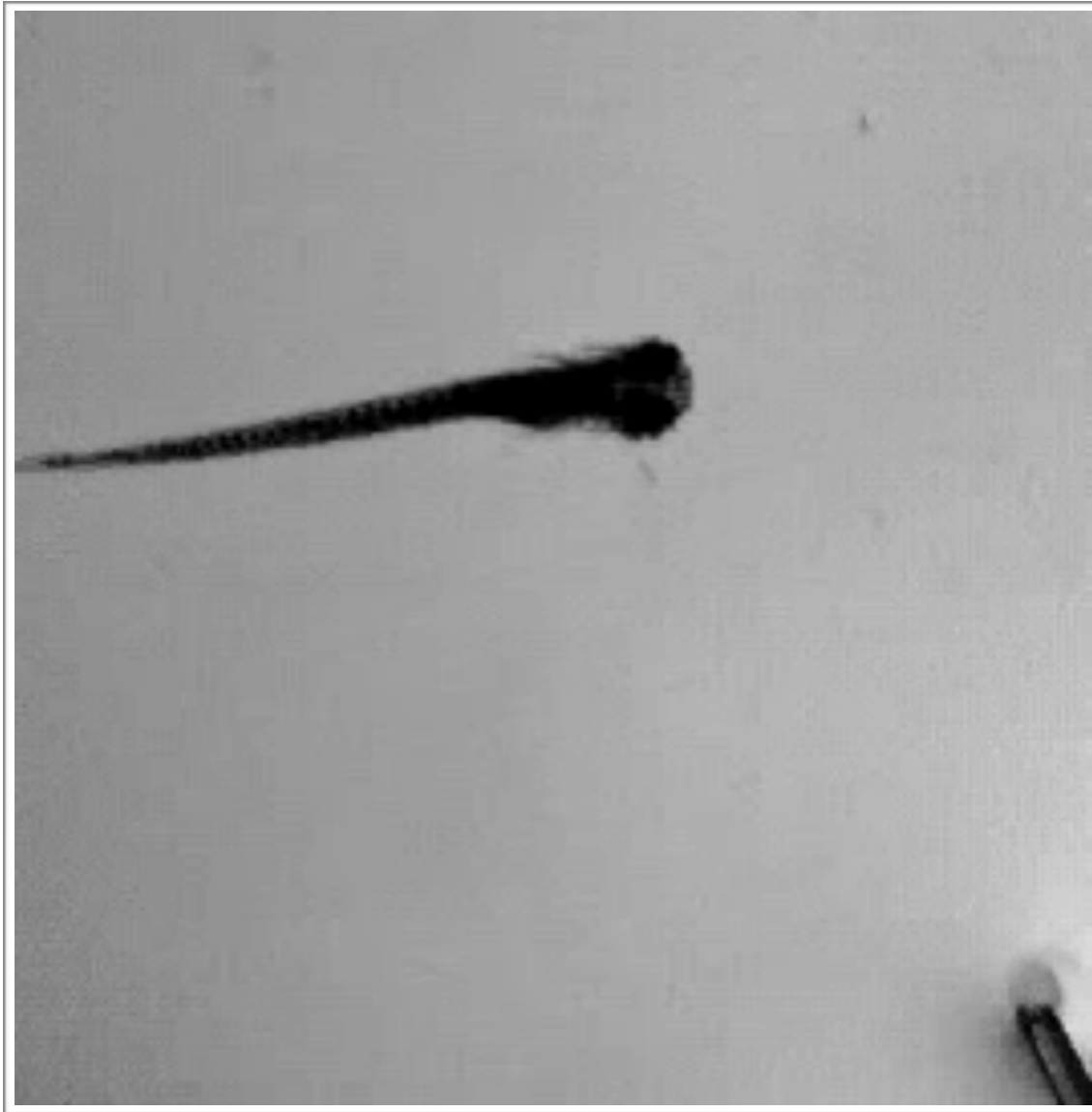
Journal of Fluid Mechanics

VOLUME 698



C-start is an **escape** motion pattern

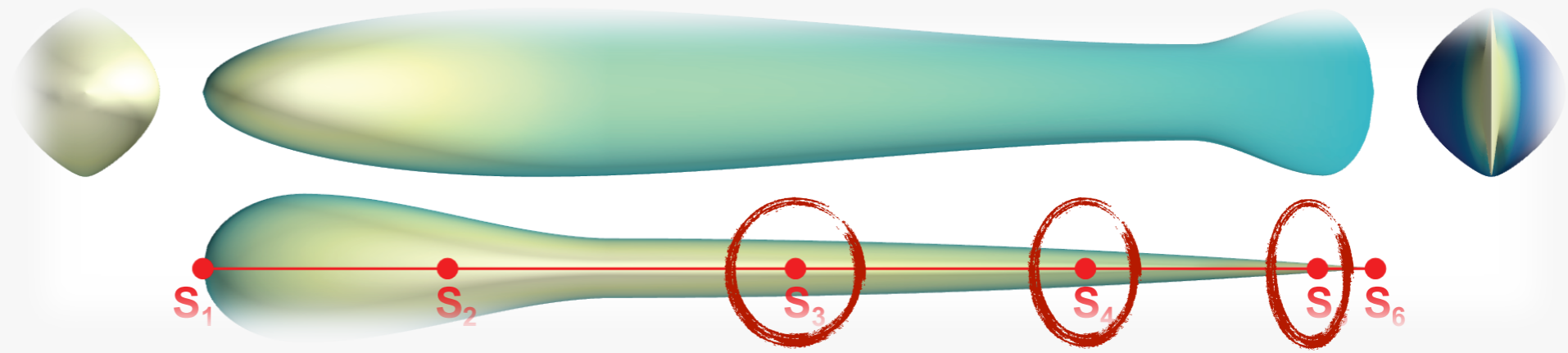
Is C-start **optimal**?



FLOW @ $Re = \frac{L^2 / T_{prop}}{\nu} = 550$

GEOMETRY

4.4mm long larva zebrafish of age 5 days post-fertilization



Muller, van den Boogaart, van Leeuwen. JEB, 2008
 Parichy et al. Developmental Dynamics, 2009
 Fontaine et al. JEB, 2008

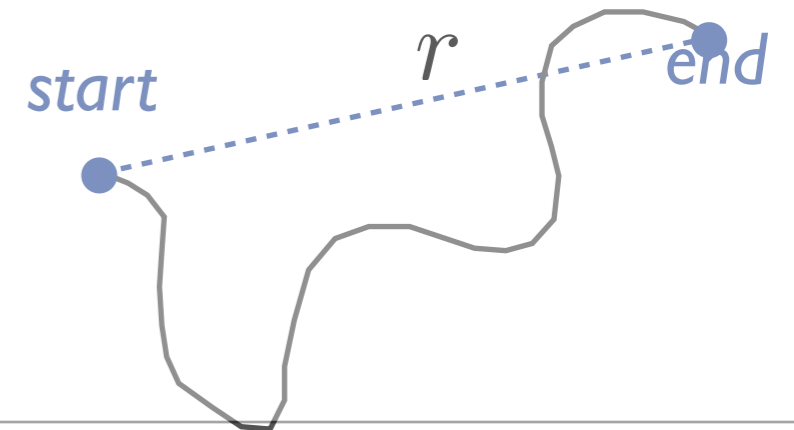
PARAMETERS

$$\kappa_s(s, t) = B(s) f\left(\frac{t}{T_{prep} + T_{prop}}\right) + K(s) \sin\left[2\pi\left(\frac{t}{T_{prop}} - \tau(s)\right) + \phi\right]$$

8 = 3 + 3 + 1 + 1

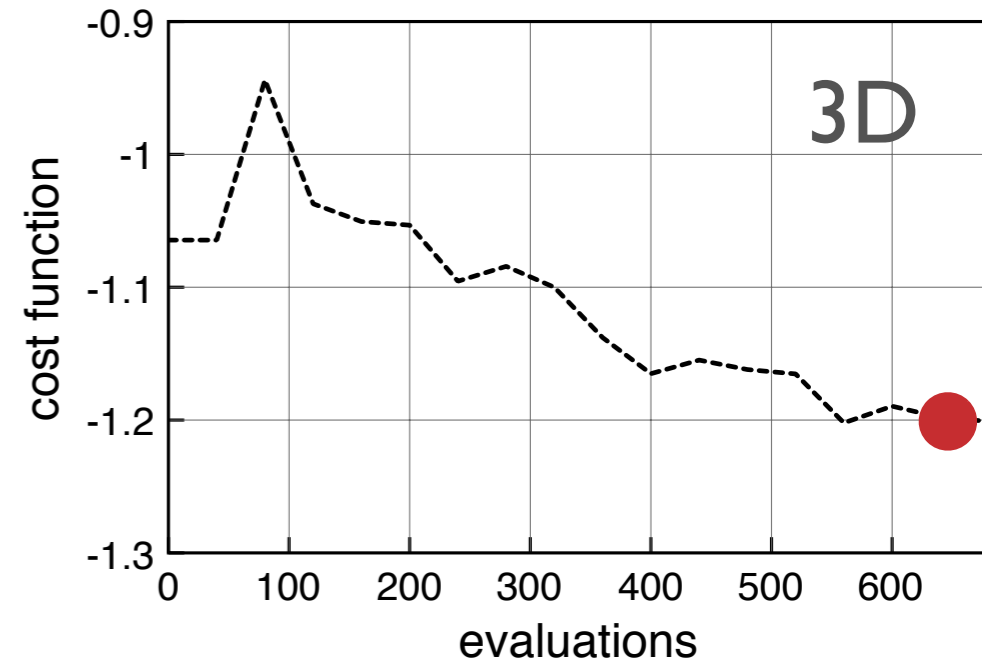
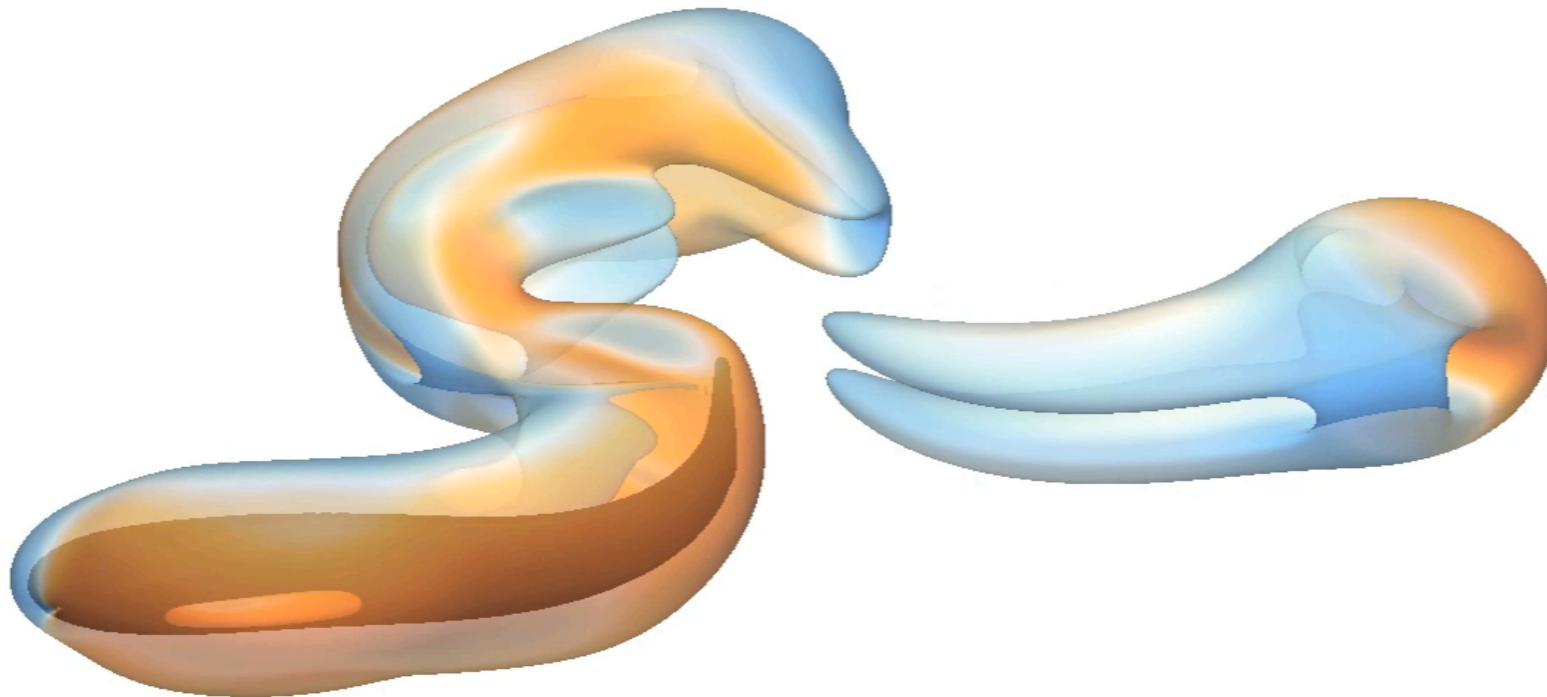
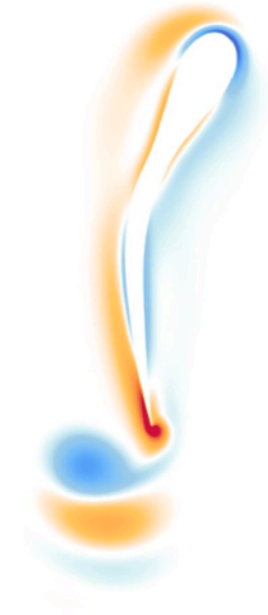
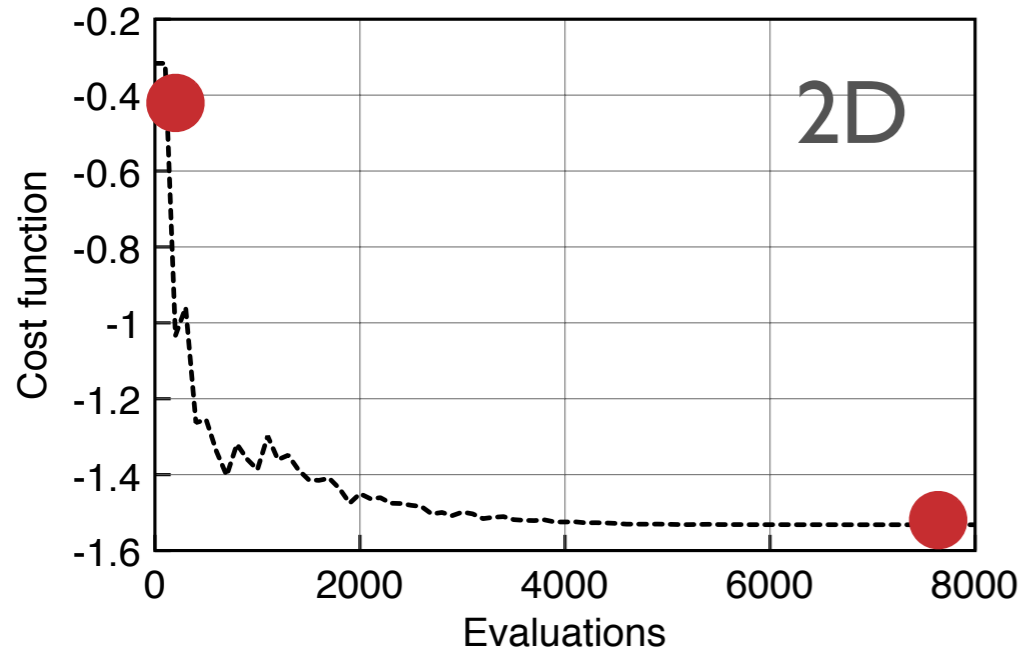
COST

$$f = -r \Big|_{T_{prep} + 2T_{prop}}$$



Optimization strategy : evolutionary algorithm + covariance matrix adaptation

C-start is **OUTCOME** of optimization



Pros and cons of the Wavelet-based multi-resolution particle method

Pros:

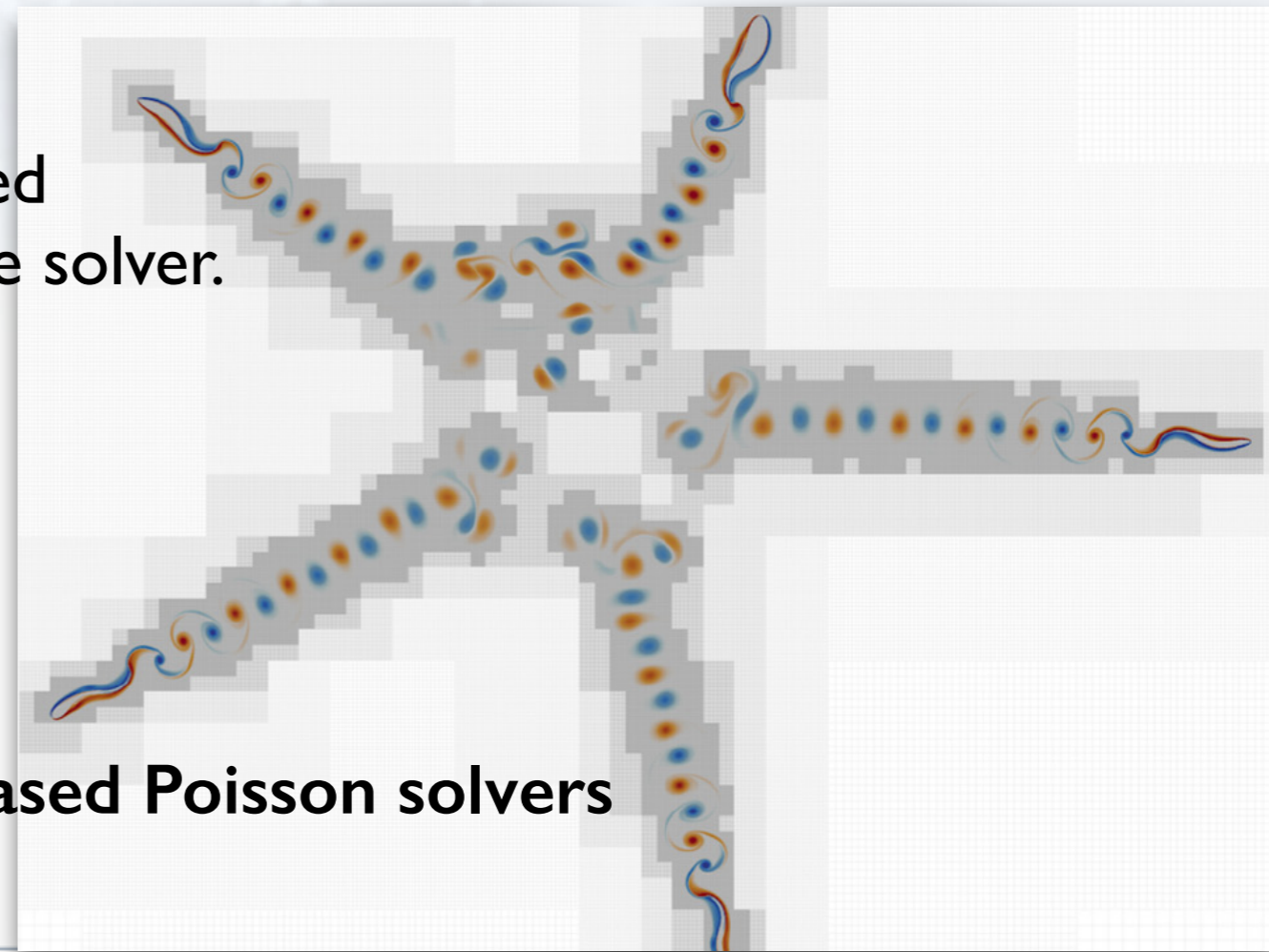
- great compression / efficiency boundary layer / wake
- well adapted to Lagrangian transport of scales

Cons :

- Scalability non optimal, load balancing difficult (not a problem for evolutionary optimization)
- In a vorticity particle method, velocities of particles have to be recovered through Biot-Savart law and Fast Multipole solver.

$$\mathbf{u}(\mathbf{x}_p) = \sum_{q \neq p} v_q \omega_q \mathbf{K}(\mathbf{x}_q - \mathbf{x}_p)$$

Much more expensive than FFT-based Poisson solvers



Alternative (more naive approach) : Bi-level formulation of vorticity transport

3D Euler equations in vorticity formulation
$$\frac{\partial \omega}{\partial t} + \operatorname{div}(\mathbf{u}\omega) - [\nabla \mathbf{u}]\omega = 0$$

Using higher resolution for vorticity than for velocity is equivalent to advecting vorticity with mollified velocity

$$\frac{\partial \omega}{\partial t} + \operatorname{div}(\bar{\mathbf{u}}\omega) - [\nabla \bar{\mathbf{u}}]\omega = 0$$

where $\operatorname{div} \bar{\mathbf{u}} = 0$, $\operatorname{curl} \bar{\mathbf{u}} = \bar{\omega}$, and $\bar{\omega} = \omega * \zeta_\varepsilon$

In practice means that

- ω computed (advection-diffusion equation) on **fine grid**
- $\bar{\mathbf{u}}$ computed on **coarse grid**, from filtered field $\bar{\omega}$

Problem : these modified NS equations induce enstrophy production at the sub-grid scale

This can be analyzed and overcome by appropriate “minimal” non-linear dissipation terms [C., JCP 1996]

This is reminiscent to subgrid-scale models involved in Large Eddy simulations

To compensate for this backscatter, need to introduce **anisotropic subgrid-scale dissipation** :

$$\frac{\partial \omega}{\partial t}(x) = C \int [\omega(x) - \omega(y)] [(u(x) - u(y)) \cdot \nabla \zeta(x - y)]_- dy$$

with linear cut-off, easily discretized on a 27-points stencil.

Constant C needs to be tuned (unfortunately)

Proof of concept on Taylor-Green vortex

[C. Mimeau, PhD thesis 2015]

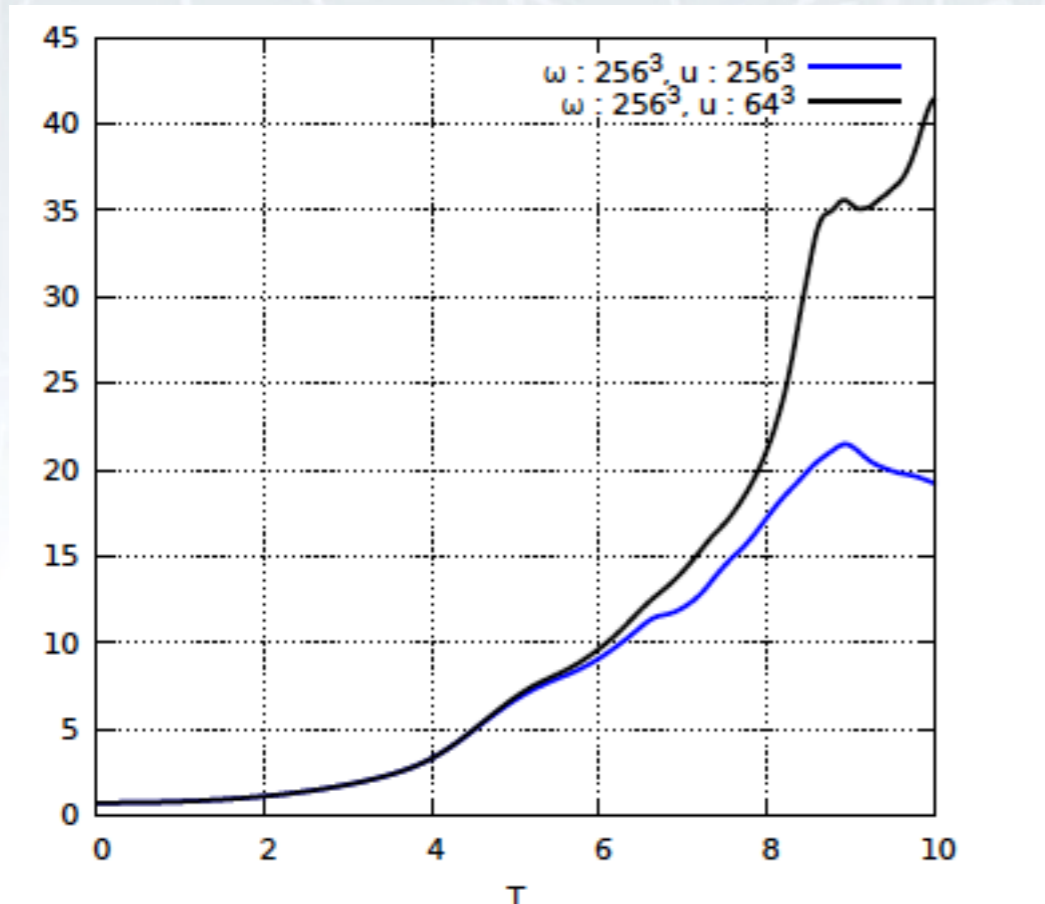
$$u_x(\mathbf{x}, t = 0) = \frac{2}{\sqrt{3}} \sin\left(\theta + \frac{2\pi}{3}\right) \sin(x) \cos(y) \cos(z),$$

$$u_y(\mathbf{x}, t = 0) = \frac{2}{\sqrt{3}} \sin\left(\theta - \frac{2\pi}{3}\right) \cos(x) \sin(y) \cos(z),$$

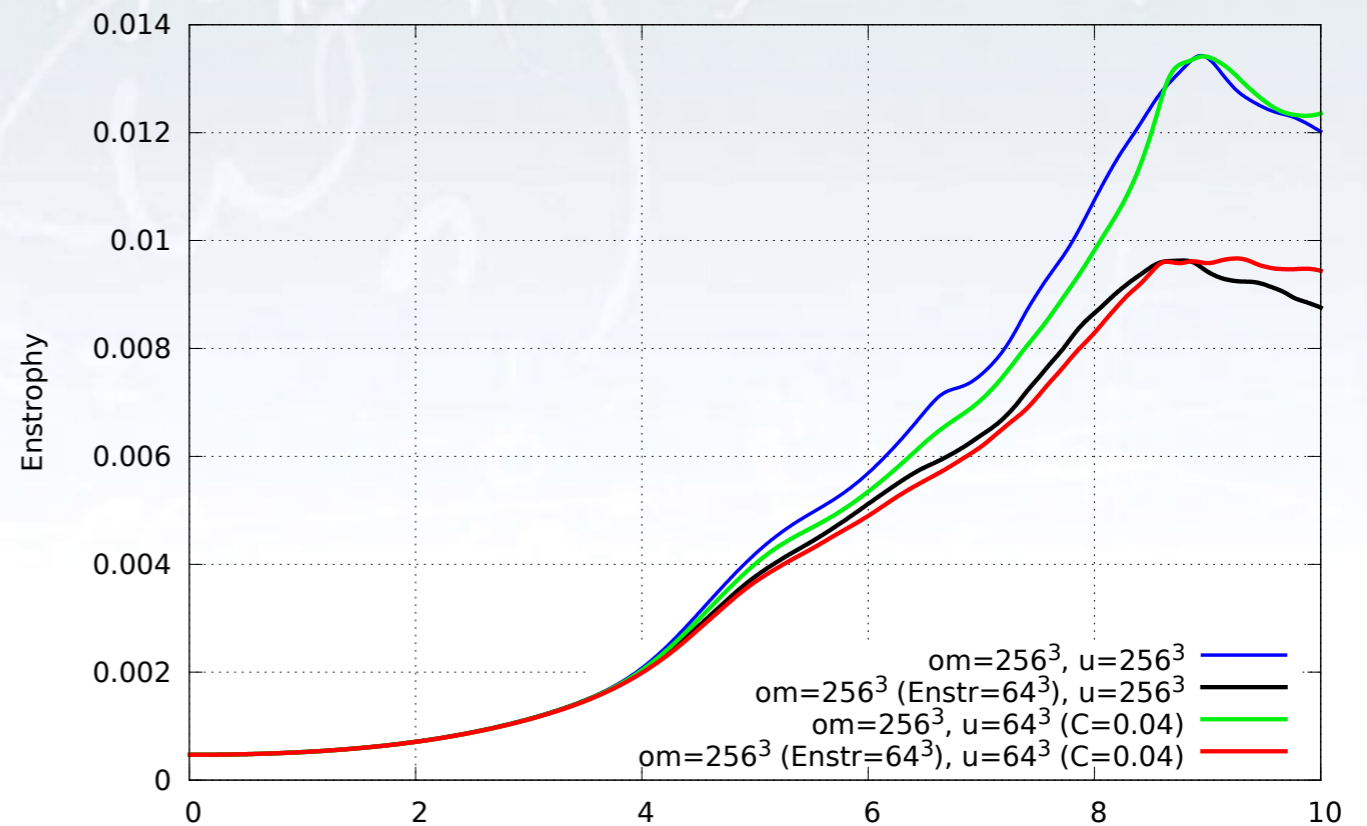
$$u_z(\mathbf{x}, t = 0) = \frac{2}{\sqrt{3}} \sin(\theta) \cos(x) \cos(y) \sin(z).$$



Comparison of enstrophy in a fully resolved 512³ simulation [van Rees et al., JCP 2011] and in a bi-level 64³/256³ simulation



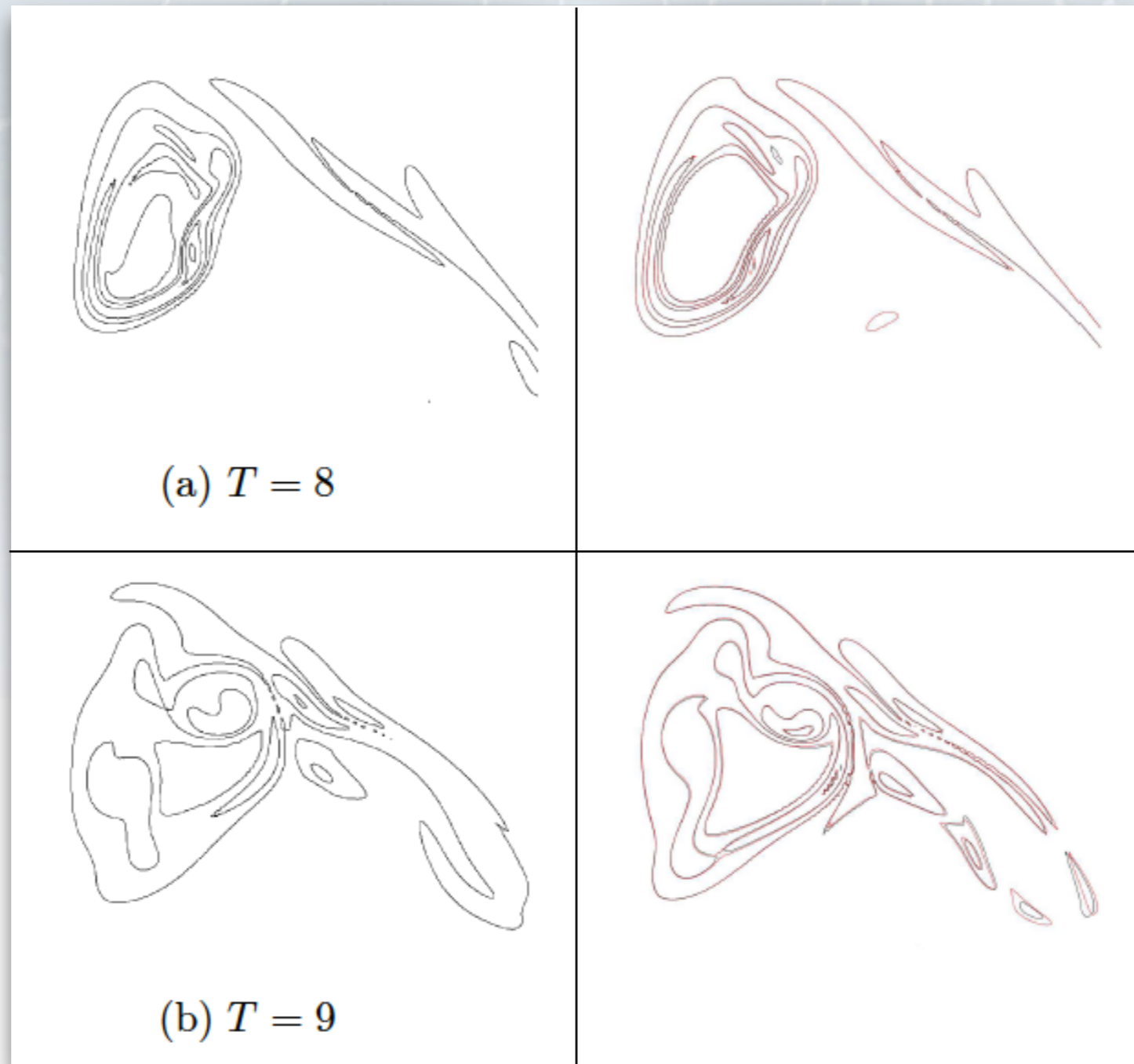
Case without subgrid-scale dissipation
and unphysical enstrophy growth



Case with subgrid-scale dissipation
with C=0.04

Temporal evolution of enstrophy

Contours of vorticity in a cross-section



$128^3 / 512^3$ simulation

512^3 simulation
(van Rees et al., JCP 2011)



Similar results with turbulent plane jets (**with same value of coefficient C**)

What is the point of doing of using different particle resolution for velocity and vorticity (is there any CPU gain at all) ?

Vorticity advancement implies only local operations (advection - remeshing -diffusion)

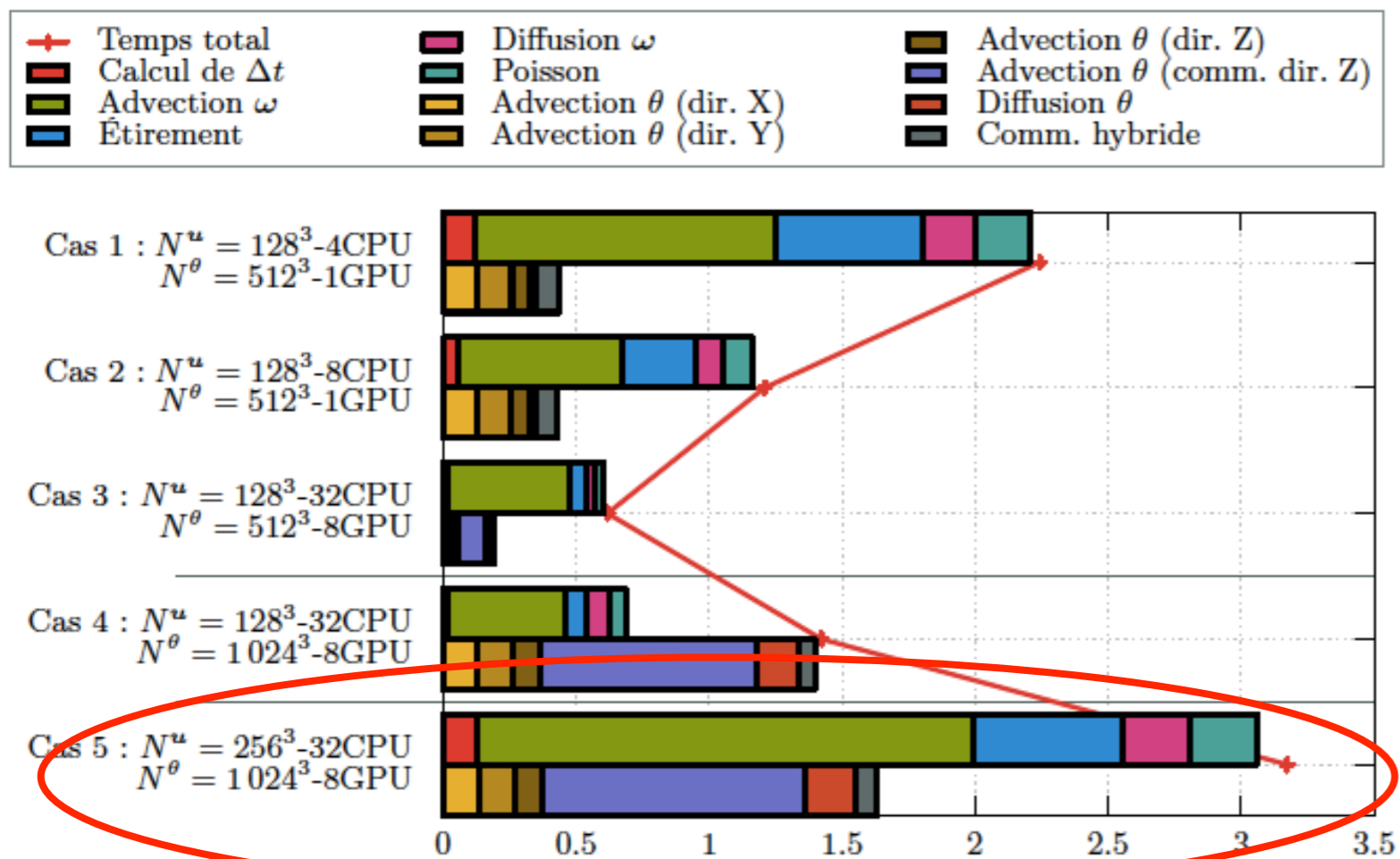
-> good parallel scalability

Velocity calculation implies non-local operations (Poisson solver, FFT ..) : fast but not highly parallel

Hybrid computing idea :

- distribute operation intensive parallel parts of the algorithm (vorticity) to GPU (or accelerators)
- keep less operation intensive (more communication demanding) on classical CPU.

Proof of concept of transport of passive scalar:
 Summary of performance for hybrid CPU /GPU implementations of
Navier Stokes / scalar transport



Computational time (s) / iteration
 (Etancelin, PhD thesis, 2014)

Performance equivalent to conventional MPI implementation of same method on several thousand Blue Gene cores

Ongoing : extension to Navier-Stokes (transport of vorticity)
DNS at the price of a LES ?

Conclusion and outlook

SL Particle methods combined with immersed boundary technics efficient in many CFD problems dominated by transport

Efficiency relies on

- high order algorithms, not constrained by CFL conditions
- ability to be combined with grid-based technics, including MRA
- high parallel scalability

Ongoing works:

- explore double-diffusion at high Sc with hybrid method, in Finite Volume NS solvers
- continue the validation of the vorticity/velocity bi-resolution approach on turbulent jets and boundary layers
- implement it on hybrid architectures
- extend the approach to (finite-difference, velocity-pressure)/(particles, level set) for **multiphase flows with surface tension**, using new semi-implicit time-discretization [Cottet-Maitre, JCP 2016]