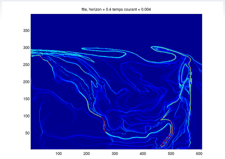
Fast FTLE computations

Computations of Lagrangian Coherent Structures



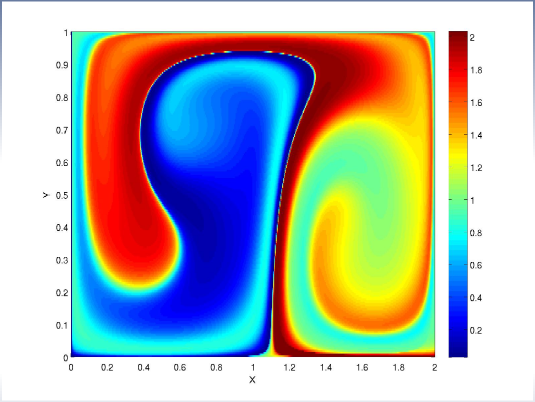
Florimond GUENIAT[1,2]
L.PASTUR[1], F. LUSSEYRAN[1], M. AMMI[1],

J. FALCOU,[3],R. MONTAGNE[3]

BIFD 2011 – 18 21 July

[1]LIMSI-CNRS,[2]Orsay University,[3]LRI-CNRS

✉ florimond.gueniat@limsi.fr

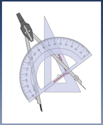


Representation of the flow over the x direction

✉ florimond.gueniat@limsi.fr

Contents

florimond.gueniat@limsi.fr



Lagrangian Coherent Structures 1/2

We are tracking invariant manifolds through the dynamical flow Φ, i.e. time dependent sets of hyperbolic points.
One can see these sets as some kinds of separatrix of the flow :

$\rightarrow$ Given a fluid particule willing to cross an invariant manifold, it will end stick on it, unable to get through the separatrix.

Therefore, finding these structures is somehow equivalent to compute the local maxima of the particules separation's rate.

Lagrangian Coherent Structures 2/2

Computing this rate means computing the Green-Cauchy strain Tensor $\boldsymbol{T}$. A way to exprime it is with the flow's Jacobian $\frac{\mathrm{d}\boldsymbol{\Phi}}{\mathrm{d}x}$:

$$\boldsymbol{T} = \frac{\mathrm{d}\boldsymbol{\Phi}}{\mathrm{d}x} \times \frac{\mathrm{d}\boldsymbol{\Phi}}{\mathrm{d}x}^{\dagger}$$

Then, the particules separation's rate is the maximum eigenvalue of $\boldsymbol{T}$, **i.e.** Finite-Time Lyapunov Exponent of the flows.

✉ florimond.gueniat@limsi.fr

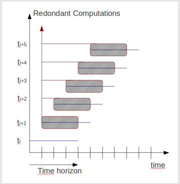
Redondancy (1/2)

Why we don't want trajectories computations

Jacobian of the flow ⇒ Cauchy-Green strain Tensor ⇒ 4 trajectories' computations ad minima around the point.
In a nutshell :
At time t, 1 information ⇒ 4 integrations.

To avoid these redondancies, we will look at trajectories through the flows.

Redondancy



How to avoid redondant trajectories computations ?

florimond.gueniat@limsi.fr

Redondancy (2/2)

Trajectory in physical space $\equiv$ evolution of a few components of the state vector.

$$\vec{\boldsymbol{X}}(t) = \Phi_0^t\left(\vec{\boldsymbol{X}}(0)\right)$$

$\Rightarrow$ Replace trajectory computations with a flow computation !

$$\Phi_{t_0}^{t_0+T}\left(\vec{\boldsymbol{X}}(t_0)\right) = \vec{\boldsymbol{X}}(t_0+T)$$

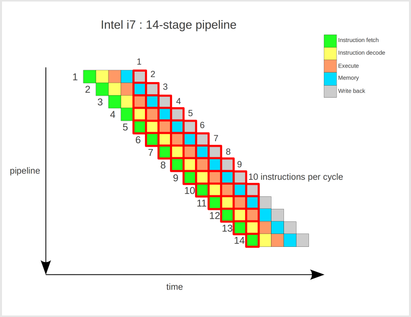
Therefore

$$\Phi_{t_A}^{t_C} = \Phi_{t_B}^{t_C} \circ \Phi_{t_A}^{t_B}$$

✉ florimond.gueniat@limsi.fr

Low level consideration (1/4)

SIMD and pipelines



SIMD : Single Instruction, Multiple Data. It is performing the same operation on multiple data simultaneously.
Thus, it is exploiting data level parallelism i.e. memory coalescent data, and processors pipelines.

Low level consideration (2/4)

Vectorization and parallelism

Vectorization of the code has 2 objectives and 2 consequences :

- → O1 : The Memory coalescence through linear indexing
- → O2 : The use of the pipelines' depth
- → C1 : SIMD !
- → C2 : Obvious parrallelism

florimond.gueniat@limsi.fr

Low level consideration (3/4)

Gain

The theoritical gain, for single dataset :

$$K_{th} = \underbrace{(NP-4)}_{\text{pipelines}} \cdot \underbrace{\frac{D}{8}}_{Depth} \cdot \underbrace{NC}_{Cores}$$

In our application, the theoritical gain is $K_{th} = (14-4) \cdot \left(\frac{128}{8}\right) \cdot 8 = 1280$. With Matlab (well known for its efficiency...), we typically have a gain $K_{exp} \approx 700$.

Low level consideration (4/4)

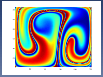
Drawbacks

1. SIMD implies Cartesian Grid, i.e. space increment has to be constant.
2. Elementary flows interpolation is time consuming.

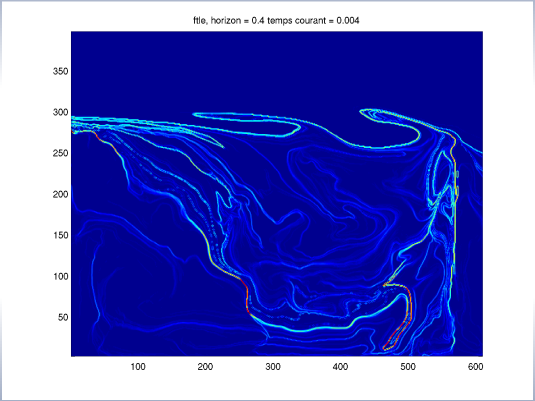
Solutions

1. A conformal transformation of the dataset solves that.
2. Interpolation is an hard-ware implementation on GPU.

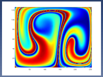
✉ florimond.gueniat@limsi.fr



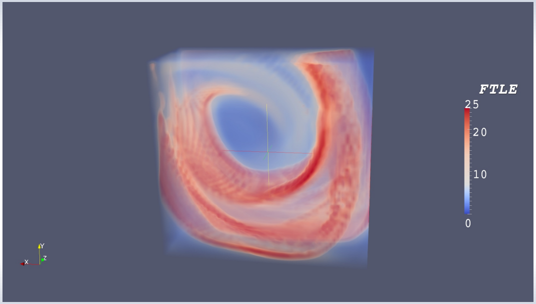
2D FTLE on a cavity flow



florimond.gueniat@limsi.fr



3D FTLE on a DNS flow



florimond.gueniat@limsi.fr

This is the end

Thank you for your patience and attentiveness !

✉ florimond.gueniat@limsi.fr

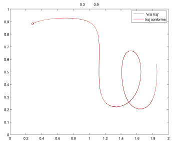
Annexes - Conformal transformations

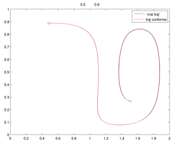
Problem of the code : good only for regular mesh (for the SIMD)
Always true in experiences
Never true for DNS computations !
Solution : conformal transformations ≡ ad hoc **inversible** deformation of the data.

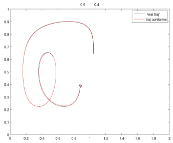
$$\vec{U}_{conf}\left(\boldsymbol{X}\right) \cdot \Delta x_{i\ reg} = \Delta t = \vec{U}\left(\boldsymbol{X}\right) \cdot \Delta x_i$$

✉ florimond.gueniat@limsi.fr

Annexes - Conformal transformations







Some examples in a chaotic but synthetic flow.

florimond.gueniat@limsi.fr