A blob tracking algorithm for the study of turbulence-flow interaction

G. Hornung[1], S. Hoste[1], G. Verdoolaege[1,2], Ph. Ghendrih[3], Y. Sarazin[3], J.-M. Noterdaeme[1,4]

[1] *Department of Applied Physics, Ghent University, B-9000 Ghent, Belgium*

[2] *LPP-ERM/KMS, B-1000 Brussels, Belgium*

[3] *CEA, IRFM, F-13108 Saint-Paul-lez-Durance, France*

[4] *Max Planck Institute for Plasma Physics, Boltzmannstr. 2, 85748 Garching, Germany*

Introduction

Mean flows attract a growing interest as they can suppress the plasma turbulence and are thus believed to play a crucial role in the transition to improved confinement regimes. Turbulent structures, often called blobs, are deformed through their propagation across the flow. Following the tilting and stretching of the blobs, the Reynolds stress transfers energy from the ambient turbulence to the flow, leading to a reduction of the fluctuation level. Given the continuous blob deformation during the stretching and splitting processes, a careful description of the evolution of the blob shape can yield useful information on the turbulence-flow interaction. A non-rigid object tracking algorithm, aiming at characterizing the blobs individually, has therefore been developed and benchmarked on simulated turbulent data.

In the following, we first briefly describe the simulation of turbulent data. The tracking algorithm is then presented and we finish by providing a glimpse of the possibilities offered by the algorithm in terms of physical studies.

Simulated turbulent data

The fluctuating density data presented hereafter were obtained with the scrape-off layer turbulence code TOKAM-2D [1]. The code solves the conservation equations for density and charge with cold ions and constant electron temperature in the $r-\theta$ plane. Fluctuations are generated by a magnetic curvature term and are assumed to have flute characteristics. Plasma-wall interaction is accounted for via Bohm conditions imposed at both ends of the field lines.The space and time coordinates are normalized to ρ_s and ω_{ci}, respectively. The size of the simulation box is $256\rho_s \times 256\rho_s$. The density and potential are normalized as follows, $n = n/n_0$, $\phi = e\phi/T_e$ where n_0 and T_e represent an arbitrary density and electron temperature, respectively.

Blob tracking algorithm

Extending previous work [2, 3], the algorithm has been designed to track blobs with arbitrary shapes and keeps record of the merging and splitting processes as illustrated in fig. 1. For each frame, the algorithm stores blob attributes such as the position of the center of mass, the shape, area or eccentricity.

The algorithm is divided in several steps. First, blobs are detected, where blobs are defined as regions of at least 50 connected pixels with density values larger than 1.5 local standard deviation.

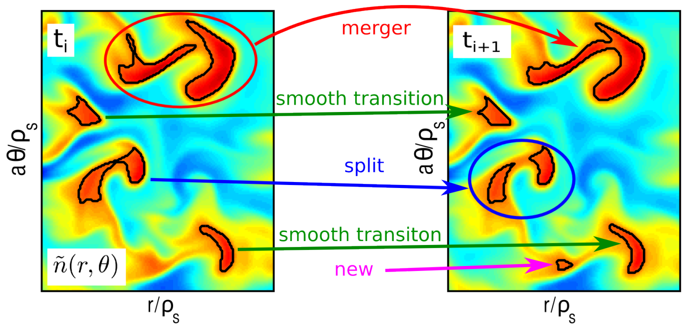

Figure 1: *Evolution of the simulated fluctuating density between two successive time frames. The blobs contours are highlighted by solid dark lines. Different scenarios are illustrated (smooth transition, new blob, split and merger).*

The second step consists of associating, in a pairwise manner, blobs between successive frames. This is done by minimizing the Euclidean distance between the position of the center of mass of two blobs. At this step, the association is required to be unique. When a conflict arises, namely multiple blobs at frame t_{i+1} can be associated to a single blob at frame t_i, the minimization is performed again but only between the blobs involved in the conflict. An ensemble of associated blobs defines a trajectory.

Up to this point, no attention was paid to situations where special events, i.e. split or merger of blobs, occur. To detect such events, we assume that the blob properties suddenly change in case of a split or merger. The event detection step therefore implies searching bifurcation in the existing trajectories. An event is defined at time t_i if the time derivative of at least 3 blob attributes F fulfill the criterium $dF/dt|_{t_i} > \Gamma dF/dt|_{t_{i-1}}$ & $dF/dt|_{t_i} > \Gamma dF/dt|_{t_{i+1}}$, Γ being a user-defined threshold.

Once events have been identified, it remains to build the blob genealogy. This is done during the event refinement step. We will consider the case of a split for explanatory purposes. Building blob genealogy consists of associating the parent (the blob which splits) to its children (the blobs resulting from the split). To do so, we assume that the momentum of the parent is approximately transferred to the children during the split. Consequently, the set of blobs with the total momentum that is the closest to that of the parent are considered as the children. A similar reasoning is used in the case of a merger.

At this point, we dispose of the full trajectories and the blobs genealogies but it remains to evaluate the algorithm performance. The outputs of the algorithm are therefore compared with a ground truth established for a given dataset. The ground truth is defined as the set of correct trajectories and blob genealogy obtained by processing manually the simulation frames. Several indicators have been built to evaluate the performance of the different steps of the algorithm. Both blob detection, blob association and event refinement steps have a large success rate (> 99%). The optimal performance for the event detection has been obtained by using a threshold $\Gamma = 2.6$. Even in this case, the false positive and true negative event detection rates remains high ($\approx 30\%$). These quit large inaccuracies in the event detection is due to the fact that during splitting, many blobs do not divide in equal parts but rather in a large and a small one, making the detection of trajectory bifurcation more difficult.

Investigation of blob velocities

Contrary to time-delay estimation methods based on correlation analysis [4], the tracking algorithm offers the possibility to unambiguously define the instantaneous blob velocity. The velocities have been estimated by the time derivative of the position of the center of mass. The radial velocities are observed to be widely spread around their mean values ($\langle v_r \rangle \approx \sigma_{v_r} \approx 0.017 c_s$) and do not show any radial dependence.

The potential relation between blob sizes and velocities, predicted by different analytical models [5], was investigated by constructing the joint distribution of the blob area and radial velocities as shown in fig. 2. There is no clear relation between the mean radial velocity and the area but it is possible to deduce interesting information from their joint distribution. Most of the blobs have small area and propagate with moderates velocities. However, only small blobs are observed to travel with either negative or large positive velocities. It is also interesting to note that the width of the distribution seems to decrease when the blob area increases.

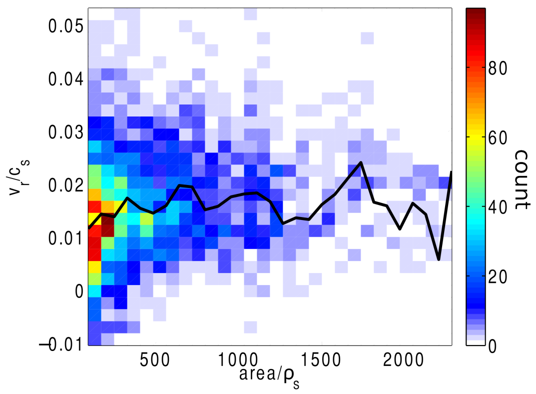

Figure 2: *Joint distribution of blob radial velocities and area. The solid dark line denotes the mean radial velocities.*

Effects of shear flows on blobs

A shear layer is obtained by polarizing the simulation box with a Gaussian potential homogenous along the poloidal region, which peaks at $r/\rho_s = 128$ and decreases with the characteristics length $L_r/\rho_s = 10$.

The impact of the shear flows on an individual blob is illustrated in fig. 3. While approaching the shear layer, the blob is convected downward. In the mean time, its area and mass slightly decrease. Inside the shear layer, the area and mass significantly increase up to the position of the maximum shear while the blob is strongly tilted and streched. Finally, the blob mass rapidly falls down and the blob disappears in the outer part of the shear region.

The effect of the shear layer on the blobs has also been analyzed by counting the number of natural births and deaths, i.e. those which do not originate from splits or mergers. Inward the shear region, the birth and death rate are very similar between biased and unbiased cases. On the contrary, the number of deaths increases by almost a factor 3 in the inner and outer part of the shear layer. Interestingly, the birth rate was also observed to significantly increase in the shear layer revealing that shear flows can both suppress and generate turbulent structures.

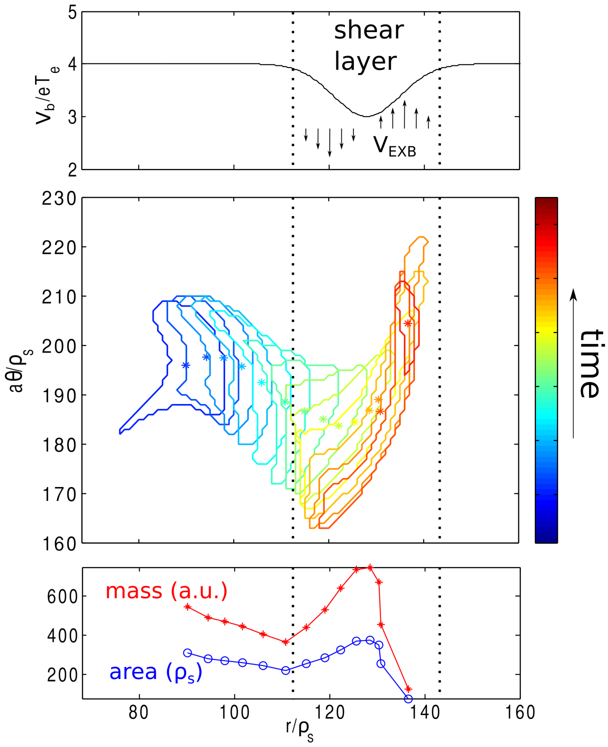

Figure 3: *Top panel: radial profile of the externally applied polarization. Center panel: evolution of the blobs shape while crossing the shear layer. The colorbar indicates the time direction. Bottom panel: variation of the blob area and mass inside the shear layer.*

Conclusion and perspectives

A blob tracking algorithm has been developed with the aim to provide advanced turbulence analysis capabilities. The algorithm has been benchmarked on a SOL turbulent simulated density and a taste of possible physical studies has been presented. In the near future, we plan to improve the event detection step as for now it is a weak point. The longterm goal is to use of the algorithm with gas-puff imaging experimental data [6]. This will probably require rethinking the blob detection step to handle the noise inherent to experimental data.

References

[1] Y Sarazin and P Ghendrih. Phys. Plasmas, **5** 12 (1998).

[2] Ph. Ghendrih *et al.*, J. Nucl. Mater. **337-339** (2005).

[3] J. Myra *et al.*, Nucl. Fusion **53** (2013).

[4] N. Fedorczak *et al.*, Phys. Plasmas, **19** 12 (2012).

[5] D. A. D'Ippolito, J. R. Myra, and S. J. Zweben. Phys. Plasmas, **18** 6 (2011).

[6] Y. Shesterikov *et al.*, Phys. Rev. Lett. **111** (2013).