Automatic and statistically robust spatio-temporal detection and tracking of fusion plasma turbulent fronts

Neil Shah[1,3,*], Yekaterina Shpanskaya[1,3,*], C-S Chang [2], Seung-Hoe Ku[2], Anatoli V. Melechko[4], and Nagiza F. Samatova[1, 3,†]

[1] Department of Computer Science, North Carolina State University, Raleigh, NC 27695, USA
[2] New York University, New York, NY 10012, USA
3Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831
4 Materials Science and Engineering Department, North Carolina State University, Raleigh, NC 27695

[*] Both authors contributed equally; [†] E-mail: samatovan@ornl.gov

Abstract. The controlled production of thermo-nuclear fusion energy is critical for providing an alternative, environmentally friendly, and renewable energy source on our planet. The technical challenge is to stabilize the dynamic turbulent flow of hot plasma in magnetic fields in a fusion energy reactor. More specifically, the issue is how to control fusion plasma instabilities, or turbulence–an analogy comparable to learning how to "Hold the Sun." To address this issue, we create computational and mathematical methodology to discover and track–both in space and time–the intricate patterns of dynamic plasma turbulence during fusion energy production in a reactor simulated on a supercomputer. In doing so, we establish several aims. First, from extreme scale data, we automatically discover the structure of fronts, or patterns where the turbulence starts, and analyze their dynamic behaviour including speed and direction of front propagation. To do this, we create an algorithm for automatic turbulent front detection, track and quantify front propagation in space and time, and assess the predictability of the proposed methodology. Comprehensively, this process can potentially predict the structure, dynamics, and function of fusion plasma turbulence. It could also enable similar analyses required in other fields, such as astrophysics and combustion.

1. Introduction

Few would argue that fusion energy has been the Holy Grail of renewable energy efforts. The success of this endeavor will have vast environmental, geopolitical, and economic impacts. The grand challenge is to produce more energy through a fusion reaction than that required to initiate the process in a reactor. A key bottleneck is the turbulence, or unstable motion, of the fusion plasma. Turbulence influences the degree of energy lost by plasma during the fusion process; therefore, controlling the turbulence is critical to viable energy production. Predicting plasma performance in a fusion reactor, such as the one aimed by the international ITER project [1] is a non-trivial task. Understanding turbulence dynamics is the key for this capability. A promising approach uses numerical simulations of plasma's dynamic behavior. The composite energy signal observed via these simulations is distributed in both space and time, or spatio-temporally.

Until recently, plasma turbulence has been considered as a local phenomenon, in which turbulence only occurs in a certain area for a certain time period. The emerging supercomputing simulations, such as XGC1, at extreme scale–with trillions of particles in 3D space and thousands of time steps–have revealed non-local and non-linear nature of turbulence propagation in the reactor with a toroidal magnetic field for confining plasma [2].

Discovery of dynamic turbulent patterns and trends from the data produced by a computer-simulated fusion reaction offers a potential to reveal ways to control the turbulence. Yet, it presents a challenge: how to effectively and efficiently analyze the massive amounts of data, which is inherently complex, noisy, and high-dimensional. To address this challenge, we create a supercomputing analytical methodology to discover, track, and statistically quantify–both in space and time–the intricate patterns of dynamic plasma turbulence fronts from extreme-scale fusion simulation data.

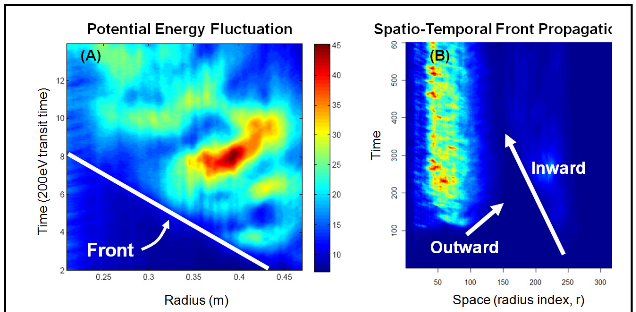

Figure 1. (A) A turbulent front illustration. (B) Spatio-temporal propagation of multiple fronts.

2. Problem statement

Informally, turbulent *fronts* are patterns where turbulence starts (Figure 1.A). Multiple fronts can propagate both in space and time, or spatio-temporally. This portrays the non-locality and complexity of turbulence, namely the impact that turbulence in one region can have in another region. For example, fronts may propagate *inward*, from larger to smaller toroidal radii, or *outward*, from smaller to larger toroidal radii (Figure 1.B).

Mathematically, fronts correspond to the points of the maximum positive curvature $c(r,t)$ of function $y(r,t)$ (Figure 2.A). For example, $y(r,t)$ may correspond to $\delta\varphi^2(r,t)$, the square of electrical potential fluctuation by turbulence. From a plasma physics perspective, propagation of turbulent fronts along the radial (r) spatial dimension over time in a toroidal fusion reactor is of primary interest. For each time step, t, spatial (radial, r_{front}) coordinates, where $\delta\varphi^2(r,t)$ attains its maximal positive curvature value are being sought:

$$r_{front}(t) = \underset{r:c(r,t)>0}{\arg\max}\, c(r,t) = \underset{r:c(r,t)>0}{\arg\max} \frac{d^2 y(r,t)}{dr^2}$$

Calculation of the curvature function from numerical simulation output data may produce noisy patterns that challenge finding the maxima (Figure 2.B). Likewise, visual front assessment is sensitive to the underlying visual color map. Also, front selection based on the specified threshold value requires domain knowledge and prevents unsupervised discovery. Thus, due to inherent complexity of simulation output data, automatic detection of turbulent fronts from the data calls for a statistically robust and predictive method.

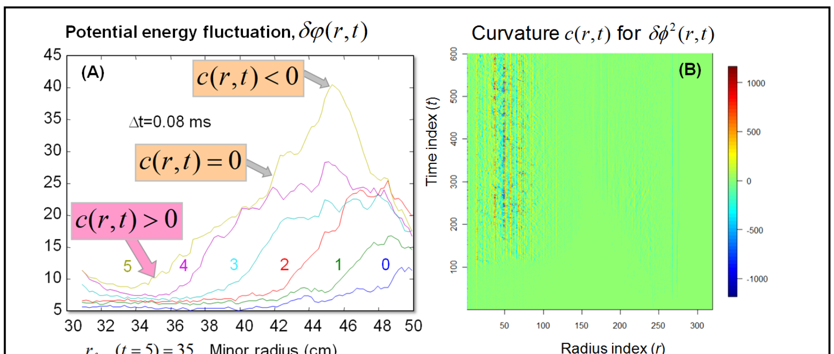

Figure 2. (A) Curvature values for different spatio-temporal points. (B) Noisy patterns of curvature values challenge finding maxima.

3. End-to-end data analysis pipeline

We introduce an end-to-end data analysis pipeline for detection and tracking, both in space and time, of turbulent fronts (Figure 3). Due to data noise, complexity, and size, the pipeline involves several critical steps, including data pre-processing, front detection, front tracking, and efficient pipeline execution using our parallel R [7] software on a supercomputer. We discovered an intelligent strategy for amplifying the signals and for reducing the noise. In a nutshell, our strategy rests on the following observation. For a fixed time-step, t_0, consider the approximation of $\delta\varphi^2(r,t_0)$ with line segments in a small $(r-\Delta r, r+\Delta r)$ spatial region around the point of interest, r. The points corresponding to the fronts are the points where the line segments change their slope from the direction almost parallel to the x-axis to the direction almost parallel to the y-axis (Figure 4). Based on this intuition, our strategy is to discover these transitions. The following subsections describe the key steps involved in this discovery process.

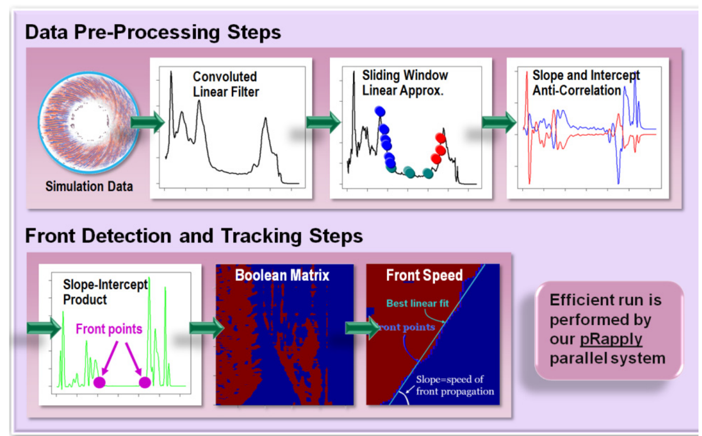

Figure 3. An end-to-end, multi-step analytical pipeline for spatio-temporal turbulent front detection and tracking.

3.*1. Data pre-processing*

Convoluted linear filtering: To reduce noise and amplify the signal, the data is first smoothed along both the temporal and spatial dimensions using a convoluted linear filtering algorithm [6]. Intuitively, the filter utilizes a "moving average" technique, where the value of $\delta\varphi^2(r,t)$ at each spatial point is the weighted average of the $\delta\varphi^2(r,t)$ values for the surrounding points; more distant points are weighted less than closer ones. Filtering along the time dimension is performed similarly.

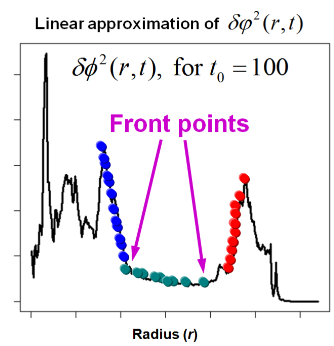

Figure 4. Front points are where the linear segments change their slope direction from *x*-parallel to *y*-parallel.

Sliding window linear approximation: The smoothed $\delta\varphi^2(r,t)$ values are then linearly approximated for each spatial window $(r-\Delta r, r+\Delta r)$ in a sliding fashion along the *r*-dimension (default, $\Delta r = 2$) (Figure 4). A linear model function is $l(r) = a \cdot r + b$ for each *r*-window of data. Since the slope (*a*) and intercept (*b*) values have different magnitudes, we normalize the values to the [-1, 1] range (Figure 5.B).

Slope-and-intercept anti-correlation: We observe that the slope and intercept will be anti-correlated (a positive slope would likely result in a negative intercept, and vice versa). The visual reasoning behind this anti-correlation is depicted in Figure 5.A, whereas the true anti-correlated data is depicted in Figure 5.B.

3.2. Front detection

Slope-intercept product: This strongly anti-correlated pattern suggests multiplying the slope and the intercept can effectively amplify the signal to be able to clearly see the curvature points of interest. We further normalize the product of the slope and the intercept to create an all-positive graph. This will further amplify the differences in the data's direction and magnitude, and thus will create a slope-intercept-product (**SIP**) "signal" representing the data at each time-step (Figure 6.A).

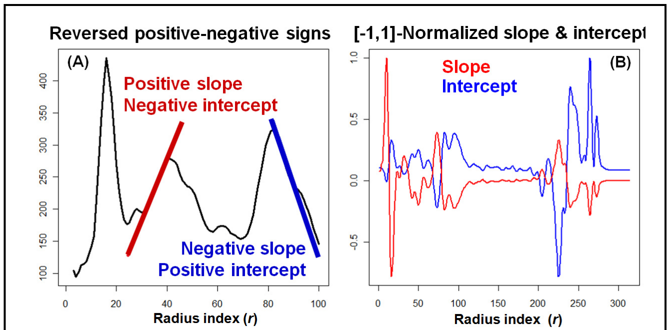

Figure 5. (A) Reversed signs for slope and intercept. (B) Normalized slope-and-intercept anti-correlation.

Boolean matrix: We establish a threshold value $T = 0.01$ to be utilized for all such signals through the series of t values. The thresholding operation converts the entire dataset of SIP signals to a matrix of Boolean TRUE/FALSE values, where TRUE corresponds to signal values v, $v \geq T$, and FALSE represents values for which $v < T$. We smooth this Boolean vector to ensure consistency in TRUE/FALSE sequences (e.g.: to prevent against detection of false-positives). This completes the detection process–the points at which sequences of Booleans change to their counterparts are the points contributing towards formation of fronts. For example, the two points at radius index *r*=110 and *r*=220 in Figure 6.B are such points for time index *t*=100.

3.3. Front Tracking

Quantifying the speed of front propagation in space and time: After creating this matrix of Booleans, we plot a heat-map by utilizing only 2 colors (dark red and blue) to represent TRUE and FALSE, respectively. This spatial-temporal heat-map of SIP signal values is depicted in Figure 6.B. As one can see, there are often multiple fronts propagating over time along the radial dimension. For example, one front protrudes outward in the region of r index values of 100 to 150 and t time index values of 100 to 180. Likewise, another front protrudes inward from r index values of 150 to 225 and t index values of 100 to 300.

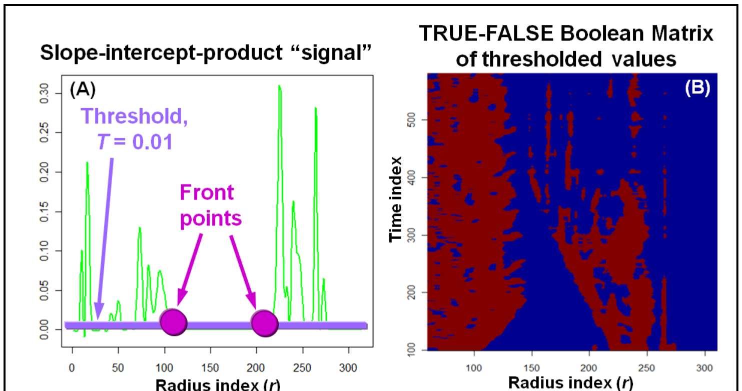

Figure 6. (A) Slope-intercept product. (B) Boolean matrix.

This visual representation of front points via SIP signal heat-map has enabled viewing the propagation of these radial fronts far more clearly than from the raw data. It is only at this point, the effective tracking of radial fronts has become possible. To customize the process to user's regions of interest, our algorithm accepts the specified r and t start-and-end values defining a rectangular region to analyze (Figure 7.A). The algorithm then automatically tracks the points, at which these Boolean series change from TRUE to FALSE (in the case of the outward front), and from FALSE to TRUE (in the case of the inward front). It creates a list of front coordinate points and performs linear regression on these points (Figure 7.B). By fitting a linear model, it derives the speed of radial propagation for each of these fronts.

The accuracy of our algorithm is consistent with both the results produced by the visual judgment of fusion scientists and with the theory [2-5]. It provides a robust method to track radial (along toroidal radius dimension) front propagation over time.

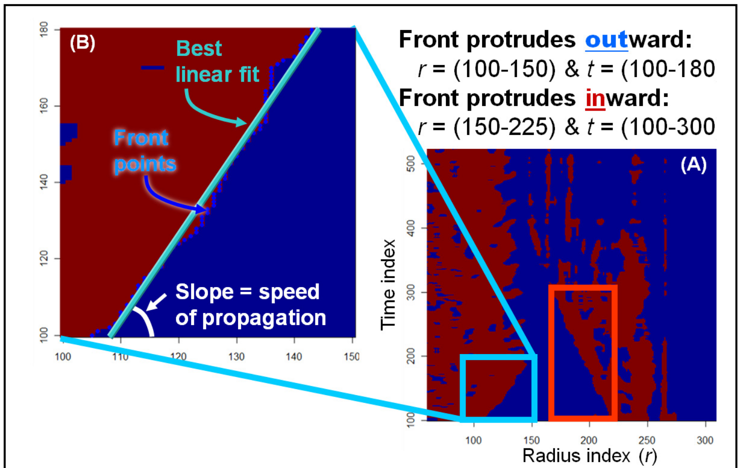

Figure 7. (A) Selection of user's regions of interest. (B) Linear regression on front points for speed of propagation.

4. Conclusion

Data produced by extreme-scale fusion plasma simulations are not only massive in size but also inherently complex due to the generally nonlinear, multi-scale and dynamic nature of the underlying physical phenomena. Yet, techniques for analyzing these complex signals are in their infancy. In particular, the dynamics of large, or meso-scale, turbulence patterns and structures has not been thoroughly addressed. These issues are relevant to galactic dynamics simulations, and so are of interest beyond magnetic fusion energy.

Toward this goal, we formulated and achieved two feasible objectives: (1) automatic detection of spatial-temporal turbulent fronts and (2) quantification of turbulence dynamic behavior, i.e. the speed and direction of dynamic turbulent front propagation. Using our advancements, we were able to automatically identify and characterize complex global turbulent frontal spatio-temporal structures. The results of our techniques were validated and accepted by fusion-science experts and agreed with the theory.

5. Acknowledgements

The work of NFS and AVM was partially supported by the SciDAC Scientific Data Management Center from U.S. Department of Energy, Office of Science. The work was performed in part at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC under Contract No. De-AC05-00OR22725.

6. References

1. ITER, http://www.iter.org/default.aspx.
2. C.S. Chang, S. Ku, and P. Diamond, Phys. Plasmas, 2009. 16(056108).
3. Yagi, M., T. Ueda, S. Itoh, M. Azumi, K. Itoh, P. Diamond, and T. Hahm, *Turbulence spreading in reversed shear plasmas.* Plasma physics and controlled fusion, 2006. 48(5).
4. X. Garbet, Y. Sarazin, F. Imbeaux, P. Ghendrih, C. Bourdelle, Ö. D. Gürcan, and P.H. Diamond, *Front propagation and critical gradient transport models.* Physics of Plasma, 2007. 14(122305).
5. Y. Sarazin, X. Garbet, Ph. Ghendrih, and S. Benkadda, *Transport due to front propagation in tokamaks.* Physics of Plasma, 2000. 7(4).
6. Brillinger, D.R., *Time Series: Data Analysis and Theory*. Second Edition ed. 1981: San Francisco: Holden-Day.
7. P. Breimyer, G. Kora, W. Hendrix, N. Shah, N. F. Samatova, pR: Automatic parallelization of data-parallel statistical computing codes for R in hybrid multi-node and multi-core environments. IADIS AC (2) 2009: 28-32.