Supplemental Material: Robustness-Based Simplification of 2D Steady and Unsteady Vector Fields

Primoz Skraba, Bei Wang, *Member, IEEE,* Guoning Chen, *Member, IEEE,* and Paul Rosen, *Member, IEEE*

Additional results of the example shown in Fig. 21(b) of the paper.

Fig. 1 of this supplemental document provides a number of sampled time steps, i.e., $t = 2, 5, 10, 13, 18$, respectively, of the unsteady flow shown in Fig. 21(b) of the paper. A group simplification has been applied to this vector field. The top row shows the vector fields before group simplification, while the bottom provides their corresponding simplification results. The highlighted purple contours illustrate the sublevel sets that were used to perform the group simplifications at the individual steps. There are a total of eight critical points involved in this group simplification. Although most of the times, only four critical points are canceled in a given time step (e.g., when $t = 10, 13, 18$ in Fig. 1), in the earlier time, the critical pairing switches occur more often, as can be seen that the canceled critical points range from two to six (e.g., $t = 2, 5$ in Fig. 1). Nonetheless, these involved critical points are identified as one group, and the proposed relax group simplification successfully removes them with only a small amount of local change, while the other parts of the flow are preserved.

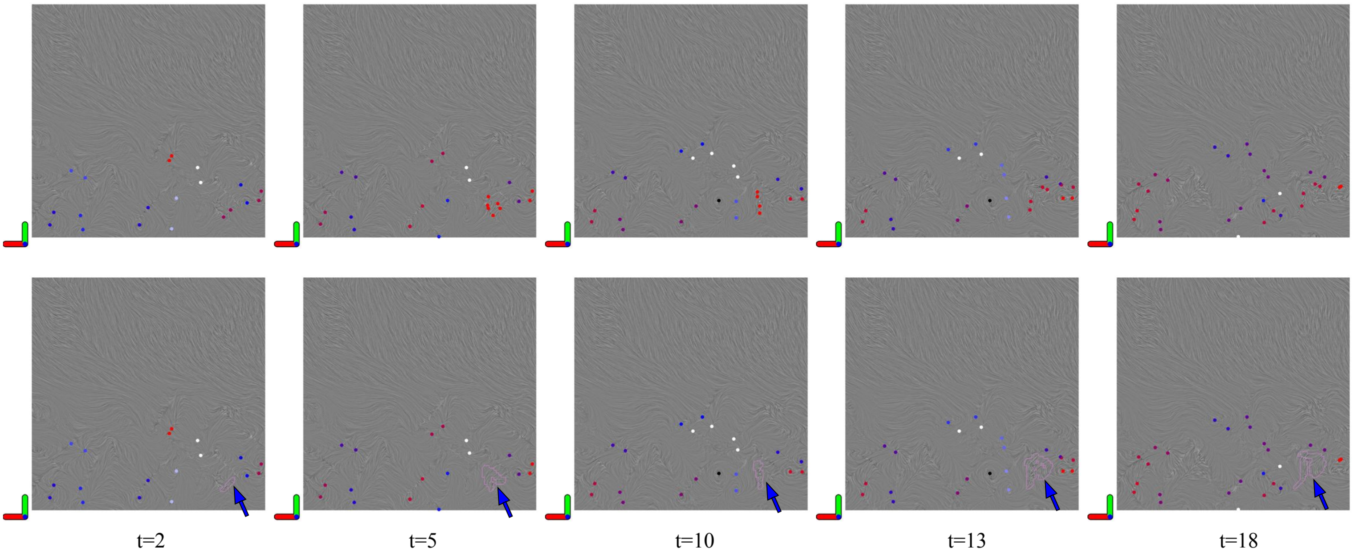

Fig. 1. A number of sampled time steps of the example shown in Fig. 21(b) of the paper. The top row shows the vector fields before simplification, while the bottom provides their simplified results. The highlighted purple contours indicate the sublevel sets used by the simplification at the individual steps.

Running time. The sampled running times of the various operations used in simplifying the vector fields based on our approach are shown in Table 1.

Bounded perturbations. By construction, the algorithm comes with theoretical guarantees on the amount of perturbation introduced to the vector field. We formally state the following theorem and its proof.

- *P. Skraba is with Jozef Stefan Institute, Slovenia.*
  E-mail: primoz.skraba@ijs.si
- *B. Wang and P. Rosen are with Scientific Computing and Imaging Institute, University of Utah.*
  E-mails: {beiwang, prosen}@sci.utah.edu
- *G. Chen is with University of Houston.*
  E-mail: chengu@cs.uh.edu

Dataset Name	Tile #	Region	# of vertices	# of edges	**Reading input** (sec)	**Smoothing** (sec)	**Cut** (sec)
OceanC	20904	(1,6)	120	285	0.007	0.647	0.014
		(8,9)	85	212	0.005	0.521	0.010
Synthetic		SyntheticC	1318	3767	0.100	19.993	0.195
Combustion	173	(10,13)	21662	64035	0.273	166.723	2.007
		(0,3)	1694	4852	0.020	0.1031	0.151
		(5,8)	3949	11474	0.084	0.423	0.542
		(18,20)	1106	3091	0.019	0.070	0.108

TABLE 1
The running times of the various operations used in simplifying the vector fields. For comparison, we show the time taken to load as inputs the respective vector fields. The code is implemented in MATLAB and the running time is given in seconds. The main bottleneck in the computation is often the Laplacian smoothing, whose running time is highly dependent on the chosen parameters, e.g., error threshold. The values shown are obtained with identical parameters to the ones which are used for the results in the paper. Finally, we note that in a more recent C++ implementation of the same algorithm, the times have been reduced by at least an order of magnitude and could be further improved with a more optimized implementation.

Theorem 4.1: Let x be a critical point of robustness r, with the corresponding component $C(x)$. Let f and $\hat{f}$ denote the vector fields before and after simplification respectively using **Unwrap** and **Cut** operations. Then $||f(p)-\hat{f}(p)||_\infty \leq r+\varepsilon$ for all $p \in C(x)$ and $f(p)=\hat{f}(p)$ for $p \notin C(x)$.

PROOF. The equality outside of $C(x)$ follows from the algorithm, as it only changes the interior of $C(x)$. If only the **Cut** operation is used, then the vector field only potentially changes during the projection, which moves the vector field value at a point by at most $r+\varepsilon$. Since all points that are projected lie within the disc of radius r, that is, their distances to the center are at most r. Also, since the distance between the line and the center is ε, the distance from any point to the line is bounded by the triangle inequality, at $r+\varepsilon$.

The only remaining case is when **Unwrap** is used. In this case, however, only the boundary values are altered (and later restored). This may alter which points are projected in the subsequent **Cut** operation, but the bound on perturbation remains the same. In the **Restore**, again only the boundary values are changed (i.e., restored) whereas the internal points remain unchanged, thereby maintaining the perturbation bound.

Note that this guarantee only applies to the algorithm without **Smoothing**. To the best of our knowledge there are no known non-trivial results to bound the amount of perturbation for Laplacian smoothing. Given the construction of the components such that the maximum magnitude of the vectors is r, we can only guarantee the trivial perturbation bound of $2r$, since after Laplacian smoothing the magnitude is still bounded by r. However, we note that simplification does not *require* the smoothing step and it was only used for aesthetic reasons.

Finally, Lemma 3.2 implies that there exists no simplification for $C(x)$ with a perturbation smaller than r (since by assumption the degree is non-zero). This implies that the simplification is optimal with respect to this metric.

Results for bounded perturbations. The main motivation for introducing Laplacian smoothing is to produce more visually appealing results. As shown in Fig. 2(b), the cutting procedure alone gives a correct, continuous but not visually appealing simplification result, compared to the vector field with Laplacian smoothing in Fig. 2(c). To further describe the amount of perturbation we introduce in practice for both our synthetic and real-world datasets, we include Table 2 below. In practice, the addition of Laplacian smoothing does increase the amount of perturbation but not significantly. For most datasets, the total amount of perturbation after **Cut** alone (the 5th column), as well as combining **Cut** and **Smoothing** (the 6th column) is roughly upper-bounded by the robustness of the critical points (e.g., the maximum magnitude of the vector field in the region of interest, the 7th column), as indicated by a ratio below 1.

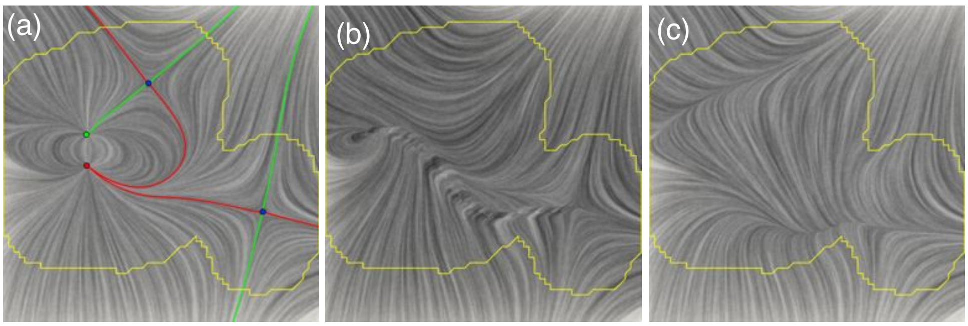

Fig. 2. SyntheticB. (a) the original vector field with its topological skeleton. (b) Simplification result by **Cut** only (without smoothing). (c) Simplification result by **Cut** and **Smoothing**.

Dataset Name	Tile #	Region	**Smoothing**	**Cut**	**Cut** and **Smoothing**	Max Magnitude
OceanA	20311	(6,9)	0.8863	0.4547	0.7162	4.2849
	21217	(8,9)	0.5902	0.5598	0.5727	5.0136
OceanB	20821	(2,4)	0.2869	0.2409	0.2409	8.0435
		(6,7)	—-	0.5432	0.5432	6.3153
OceanC	20904	(1,6)	0.1744	0.3095	0.3095	8.3236
		(8,9)	0.2740	0.5080	0.5080	9.8286
OceanD	20710	(4,5)	0.7971	0.5296	0.6801	8.3474
		(7,8)	0.3533	0.1923	0.2830	6.7341
	20715	(5,7)	1.2277	0.7006	0.9239	8.0544
		(8,9)	—-	0.5040	0.5040	10.7337
Synthetic		SyntheticA	1.0744	1.1701	1.0744	0.0059
		SyntheticB	1.2003	1.1706	1.2003	0.0059
		SyntheticC	1.4375	1.1706	1.2024	0.0059
Combustion	173	(3,6)	0.1490	0.0936	0.1490	0.1220
		(10,14)	—-	0.5944	0.6016	0.2542
		(12,13)	0.4614	0.2674	0.2699	0.0699
		(18,19)	—-	0.3626	0.3669	0.1998
		(10,13)	0.2775	0.9446	0.9448	0.4557
		(0,3)	—-	0.1439	0.1439	0.2583
		(5,8)	—-	0.4559	0.4559	0.3015
		(18,20)	0.0846	0.2661	0.2661	0.1109

TABLE 2

Amount of perturbation introduced during our simplification algorithm. The first three columns indicate the specific regions of interest, where the 3rd column includes the coordinates of the regions to be simplified. For a particular region C, the amount of perturbation (or vector field distortion) introduced by the simplification is shown as a ratio with respect to the radius of $\mathrm{im}(C)$ (which is approximately the robustness value), which is given by the maximum magnitude of the region C (7th column). The 4th column includes distortion introduced using just Laplacian **Smoothing**. The 5th column is for **Cut** procedure (possibly preceded by **Unwrap**) and the 6th column is for **Cut** and **Smoothing** (corresponds to results shown in the paper). For the synthetic datasets, for **Cut** operation only, the values could be brought arbitrarily close to 1 through an appropriate choice of ε (see Section 4.3). Some values are missing for **Smoothing** as these operations do not remove the critical points.