Eurographics Conference on Visualization (EuroVis) 2018
J. Heer, H. Leitte, and T. Ropinski
(Guest Editors)

Volume 37 (2018), Number 3

An Approximate Parallel Vectors Operator for Multiple Vector Fields

Tim Gerrits, Christian Rössl, and Holger Theisel

University of Magdeburg, Visual Computing Group, Germany

Abstract

The Parallel Vectors (PV) Operator extracts the locations of points where two vector fields are parallel. In general, these features are line structures. The PV operator has been used successfully for a variety of problems, which include finding vortex-core lines or extremum lines. We present a new generic feature extraction method for multiple 3D vector fields: The Approximate Parallel Vectors *(APV) Operator extracts lines where all fields are approximately parallel. The definition of the APV operator is based on the application of PV for two vector fields that are derived from the given set of fields. The APV operator enables the direct visualization of features of vector field ensembles without processing fields individually and without causing visual clutter. We give a theoretical analysis of the APV operator and demonstrate its utility for a number of ensemble data.*

CCS Concepts

•***Human-centered computing*** → *Scientific visualization;*

1. Introduction

The parallel vectors (**PV**) operator [PR99] is a concept that enjoys a high popularity in visualization and other communities because it is conceptually simple, generic, fast and easily computable, and applicable to a variety of problems. It has been used for the extraction of vortex core lines, for finding ridge structures, or finding bifurcation lines in flow fields. The **PV** operator yields all locations where two vector fields are parallel. These are structurally stable line structures. Different types of input vector fields open a variety of applications for the **PV** operator.

The application background of our approach are ensemble flow data sets, i.e., a number of vector fields in a common spatial domain, all describing the same flow phenomenon with slightly varying parameters. For this scenario, we search locations of vortex core lines that simultaneously describe the vortical behavior of all fields best.

This problem can be solved using one of two general strategies: either extract vortex core lines for each of the velocity fields along with a visual representation of the resulting multiple line sets – including line bundling, line clustering, or finding best representatives –, or directly extract line structures that represent the vortices of all fields best in an approximate sense.

In this paper, we present the – to the best of our knowledge – first approach using the second strategy. We introduce a new generic concept called the *Approximate Parallel Vectors (**APV**) Operator* that is applied to an arbitrary number of vector fields. Given n 3D vector fields $\mathbf{v}_1,\ldots,\mathbf{v}_n$, it is generally not useful to search for locations where all vectors are parallel. For $n > 2$, such structures – if there exist any at all – are structurally unstable: adding noise to the fields will destroy line structures with all parallel vectors. Instead, the new operator gives stable line structures at locations where all n fields are maximally – but generally not perfectly – parallel.

The **APV** operator is extremely simple in terms of computation: from the given $\mathbf{v}_1,\ldots,\mathbf{v}_n$ we compute two derived fields $\mathbf{a},\mathbf{b}$ and apply the **PV** operator to these. Despite its computational simplicity, the **APV** operator requires a rigorous mathematical analysis of its properties to make it applicable.

The paper proceeds as follows: Section 2 summarizes related work and reviews the **PV** operator. Section 3 introduces the new **APV** operator, Section 4 summarizes its properties. Section 5 explains the discretization and visualization, and in Section 6, we demonstrate the application of the **APV** operator on some test data sets and show results. Section 7 provides a discussion and compares to alternative approaches. Section 8 gives limitations and concludes the paper.

2. Background and Related Work

When working with vector fields, the visualization of extracted features enables the analysis and deeper understanding of such fields. Feature extraction has therefore grown to become a major field of research in scientific visualization. An in-depth discussion is beyond the scope of this work. We refer to Post et al. [PVH*03] for an extended overview. Features can be extracted directly from

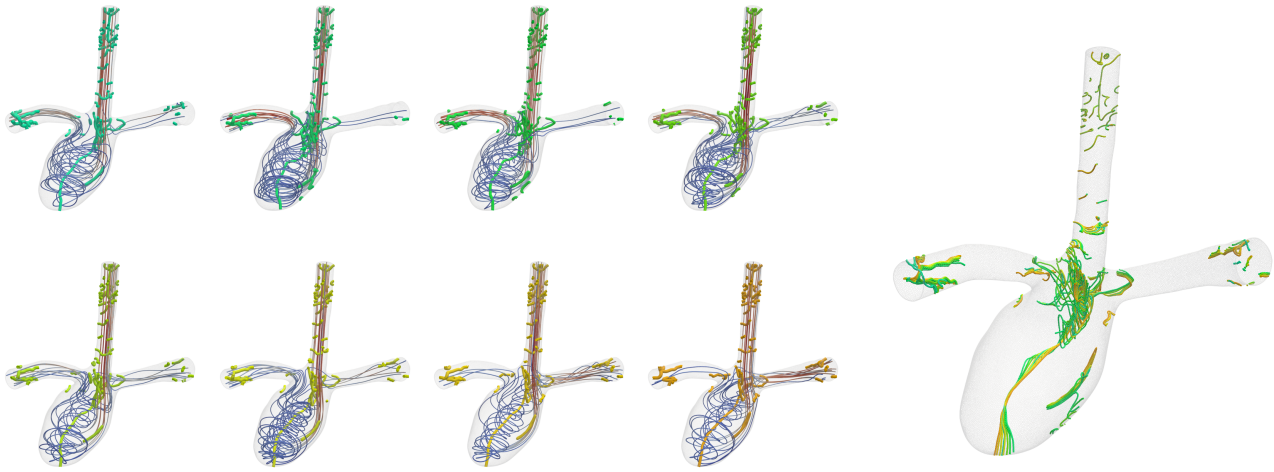


Figure 1: *Left: Eight different members of an ensemble dataset that represents different outcomes of CFD simulations of blood flow though an aneurysm with varying pressure parameters. Each member is visualized with several streamlines seeded at the flow inlet as well as parallel vector feature lines calculated from the derived acceleration field. Right: A spaghetti plot visualization of all parallel vectors lines is only of limited use for insight in the ensemble data.*

the given field itself, or they take into account additional derived information like the acceleration field.

The well-known *parallel vectors operator* was introduced by Peikert and Roth [PR99] and is used to find line-type features in a pair of vector or scalar fields. Features are characterized as locations where both fields are parallel, i.e., $\mathbf{v}_1 \parallel \mathbf{v}2$. This concept has been applied to a wide variety of problems. Roth and Peikert list several problems, including finding ridge and valley lines [Har83] or separation lines [Ken98]. A lot of effort has been put into finding the centers of vortices, or vortex-core lines. Whereas many methods differ in the extraction of solution points and their connection, most of them can be reformulated to be the result of the parallel vectors operator applied to the velocity field and a second, derived field. Sujudi and Haimes [SH95] look for structures where the velocity vector is parallel to the acceleration vector $\mathbf{v} \parallel \nabla\mathbf{v}\,\mathbf{v}$ and the Jacobian $\nabla\mathbf{v}$ has complex eigenvectors. Roth and Peikert [RP98] use higher-order derivatives to improve results and to find locations where velocity vectors are parallel to the second order derivatives of particles $\mathbf{v} \parallel (\nabla\mathbf{c})\mathbf{v}$ where $\mathbf{c} = \nabla\mathbf{v}\,\mathbf{v}$.

A number of works extend and improve the operator. In general, these approaches can be divided into two stages: first, finding solution points on a grid, and second, tracing feature lines from these locations. Banks and Singer [BS95] use a predictor-corrector scheme that uses pressure information for line tracing, which was later combined with the λ_2 method [JH95] by Stegmaier [SRE05]. Theisel et al. [TSW*05] show that tracing solution lines from extracted points or from seed points can be reformulated as a streamline integration in the feature flow field [TS03, WTVP11]. This flow characterizes the movement of critical points over time in time-dependent vector fields. This is extended for higher-order data by Pagot et al. [POS*11]. Sukharev et al. [SZP06] define an analytical tangent instead for tracing solution lines. A generalization of both of these is used by *PVSolve*, introduced by Van Gelder and Pang [VP09].

When one is given ensemble data, e.g., as produced by different Computational Fluid Dynamics (CFD) simulations, it is not only interesting to get an understanding of each individual field but rather to gain additional insight in similarities or dissimilarities of all ensemble members [JDKW15, LGY15, LGY17]. Defining, extracting and visualizing features of multifield data – either derived from one field or independent fields – is a challenging task with a variety of applications as presented by Verma and Pang [VP04]. Features can either be extracted from each single field or from information given by a combination of multiple or all fields as discussed by Obermaier and Peikert [OP14].

Ensembles of curves, like pathlines, that are extracted from each ensemble member can be collected and visualized collectively: this is known as spaghetti plots, which typically result in visual clutter and therefore pose an additional visualization challenge. Techniques have been developed to cluster curves into similar groups or to choose lines that are most representative for the whole ensemble. For example, Guo et al. [GYHZ13] create a variation field that is filtered for pathlines that best characterize the differences between different fields. Lui et al. [LGZY16] create a similar variation field, based on what they call the Longest Common Subsequences of pathlines. When pathlines pass through a shared location, several approaches make use of visualization techniques known from statistics. Ferstl et al. [FBW16] cluster such pathlines in major trends and provide a median as well as a region of confidence. Mirzargar et al. [MWK14] further summarize trends by introducing curve boxplots, additionally showing outliers and selected pathline members. An alternative approach is using features of a single field, which are then modulated by the additional data, as seen in modulated streamlines or streamtubes [USM96].

In this paper we focus on extracting one set of line-type features that take into account the information not only given by all ensemble members, but also possibly fields derived from each of these. Our

approach uses the **PV** operator, and we dedicate the final part of this section to a brief formal introduction.

The Parallel Vectors operator

Given two vector fields $\mathbf{v}_1(\mathbf{x}), \mathbf{v}_2(\mathbf{x})$ the *parallel vectors operator* [PR99] yields all locations where these fields are parallel, i.e.,

$$\mathbf{PV}(\mathbf{v}_1, \mathbf{v}_2) \;=\; \{\mathbf{x} \mid \mathbf{v}_1(\mathbf{x}) \parallel \mathbf{v}_2(\mathbf{x})\} \,.$$

The set $\mathbf{PV}(\mathbf{v}_1, \mathbf{v}_2)$ generally represents line structures also called *parallel vectors lines*.

For $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^3$, **PV** can be implemented as finding the roots of $\mathbf{v}_1(\mathbf{x}) \times \mathbf{v}_2(\mathbf{x})$. For $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^2$, the cross-product can be replaced by the determinant of the matrix $(\mathbf{v}_1 \,|\, \mathbf{v}_2)$.

There are different algorithmic approaches to numerical root finding, the Newton-Raphson method is one of the most well-known. Their success often depends on an initial guess and the behavior of the function like multiplicity of roots or crossing zero versus touching without change of sign. Root finding is a non-trivial numerical task in general, even if the vector fields are given as polynomials, i.e., from interpolated data.

The setting is simpler for an appropriate discretization of the domain: Peikert and Roth [PR99] give an analytic solution for piecewise linear vector fields. The domain is partitioned such that parallel vector locations are searched on triangles. For instance, a bounded domain $\Omega \subset \mathbb{R}^3$ is partitioned into tetrahedral pieces, and the search space is restricted to their triangular faces. Each triangle supports a linear piece of the vector fields. The restriction to a locally 2-dimensional search domain yields parallel vectors locations – if any – as isolated points, which are connected to line features in a post-process. Within each triangle Δ, solutions $\mathbf{PV}(\mathbf{v}_1(\mathbf{x}), \mathbf{v}_2(\mathbf{x}))|_{\mathbf{x}\in\Delta}$ are found by solving a (generalized) eigenvalue problem.

3. The Approximate Parallel Vectors operator

Given are n vector fields $\mathbf{v}_i(\mathbf{x})$ with $\mathbf{v}_i : \mathbb{R}^3 \to \mathbb{R}^3$ and $i = 1, \ldots, n$. We assume simultaneous evaluation at the same location $\mathbf{x}$ and write $\mathbf{v}_i$ for short.

The *parallel vectors operator* is defined for $n = 2$: $\mathbf{PV}(\mathbf{v}_1, \mathbf{v}_2)$ gives all locations where $\mathbf{v}_1(\mathbf{x})$ and $\mathbf{v}_2(\mathbf{x})$ are parallel. These are typically *line* structures. For $n > 2$ distinct fields, e.g., multiple fields of an ensemble, we would generally expect no such locations or just isolated points if we require that $\mathbf{v}_i \parallel \mathbf{v}_j$ for all $i \neq j$. The higher n the "more restrictive" is this condition. Our goal is the construction of a new operator that

- relaxes the condition and measures if n vector fields are *approximately* parallel, and
- does so in a parameter-free way by measuring if *two derived fields* are parallel using the *parallel vectors* operator **PV**.

We stack all vectors $\mathbf{v}_i$ as columns in the matrix $\mathbf{V} = (\mathbf{v}_1 \,|\, \ldots \,|\, \mathbf{v}_n)$ and compute the average

$$\mathbf{a} \;=\; \frac{1}{n}\sum_{i=1}^{n} \mathbf{v}_i \;=\; \frac{1}{n}\mathbf{V}\mathbf{1} \,,$$

where $\mathbf{1} \in \mathbb{R}^n$ is a column vector with all entries equal to 1. If we subtract the mean vector field from all fields we obtain columns $\mathbf{v}_i - \mathbf{a}$ in the matrix

$$\mathbf{D} \;=\; \mathbf{V} - (\mathbf{a} \,|\, \ldots \,|\, \mathbf{a}) \;=\; \mathbf{V} - \mathbf{a}\mathbf{1}^{\mathrm{T}} \,.$$

Then the symmetric operator

$$\mathbf{D}\mathbf{D}^{\mathrm{T}} \;=\; \sum_{i=1}^{n} (\mathbf{v}_i - \mathbf{a})(\mathbf{v}_i - \mathbf{a})^{\mathrm{T}} \;\in \mathbb{R}^{3\times 3}$$

measures the covariance of the vector fields and thus how much and in which directions the fields "spread away" from the average field. The quadratic form $\mathbf{D}\mathbf{D}^{\mathrm{T}}$ is positive definite if $\mathbf{v}_i$ are linearly independent. It's spectral decomposition gives the directions of minimum and maximum variance as eigenvectors. This is also known as the Principle Component Analysis (PCA).

We take the mean vector field $\mathbf{a}$ as a representative for the whole ensemble $\{\mathbf{v}_i\}$. We define $\{\mathbf{v}_i\}$ being *approximately parallel* if their variance obtains a maximum in direction of the mean $\mathbf{a}$. A necessary condition is that $\mathbf{a}$ must be an *eigenvector* of $\mathbf{D}\mathbf{D}^{\mathrm{T}}$, i.e.,

$$\mathbf{D}\mathbf{D}^{\mathrm{T}}\mathbf{a} \;=\; \lambda\mathbf{a} \,. \tag{1}$$

A further condition requires that the corresponding eigenvalue is maximal, i.e.,

$$\lambda = \lambda_{\max}(\mathbf{D}\mathbf{D}^{\mathrm{T}}) \,. \tag{2}$$

Approximate Parallel Vectors. We define the *Approximate Parallel Vectors* operator (**APV**) as follows: Let $\mathbf{b} = \mathbf{D}\mathbf{D}^{\mathrm{T}}\mathbf{a}$ then

$$\mathbf{APV}(\mathbf{v}_1, \ldots, \mathbf{v}_n) \;=\; \mathbf{PV}(\mathbf{a}, \mathbf{b}) \,.$$

In this definition the necessary condition (1) is expressed by the *parallel vectors operator* as $\mathbf{PV}(\mathbf{a}, \mathbf{b}) \Leftrightarrow \mathbf{a} \parallel \mathbf{b} \Leftrightarrow \mathbf{a} = \lambda\mathbf{b}$.

We reduced the definition of **APV** to the standard **PV** operator. This reduces the problem of finding **APV** lines to the application of **PV** and makes the implementation straightforward.

Filtered APV. The **APV** operator computes eigenvectors and uses only the necessary condition (1). Among all **APV** lines, we are only interested in those where the mean vector $\mathbf{a}$ is the major eigenvector that corresponds to the *largest eigenvalue*. We implement the missing condition (2) as a "filter" and define the *Filtered Approximate Parallel Vectors* operator (**fAPV**) as

$$\begin{aligned}\mathbf{fAPV}(\mathbf{v}_1, \ldots, \mathbf{v}_n) &= \{\mathbf{x} \,|\, \mathbf{D}\mathbf{D}^{\mathrm{T}}\mathbf{a} = \lambda_{\max}(\mathbf{D}\mathbf{D}^{\mathrm{T}})\,\mathbf{a}\} \\ &= \{\mathbf{x} \in \mathbf{PV}(\mathbf{a}, \mathbf{b}) \,|\, \mathbf{b} = \lambda_{\max}\mathbf{a}\} \\ &\subset \mathbf{APV}(\mathbf{v}_1, \ldots, \mathbf{v}_n) \,.\end{aligned}$$

Figure 2 shows – from left to right – examples of a non-feature point (no alignment), a point that is in **fAPV** and in **APV** (alignment with the major eigenvector), and a point that is in **APV** but not in **fAPV** (alignment with a minor eigenvector).

4. Properties of APV

In this section, we summarize a number of properties of the **APV** operator. We provide all proofs in the Appendix. In the remainder of this section, the matrix of stacked vector fields $\mathbf{V}$, the mean vector field $\mathbf{a}$ and the derived field $\mathbf{b}$ are defined as in Section 3.

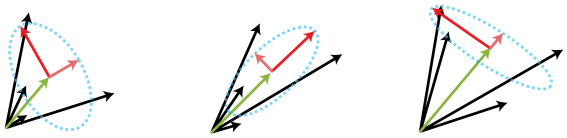


Figure 2: *A location* **x** *is part of an* ***APV*** *line if the mean vector* **a** *is an eigenvector of the covariance matrix* $\mathbf{D}\mathbf{D}^{\mathrm{T}}$. *The filtered* ***fAPV*** *operator requires additionally that this is the eigenvector corresponding to the largest eigenvalue.*

Independence of order. For any permutation π of $(1,2,\ldots,n)$:

$$\mathbf{APV}(\mathbf{v}_{\pi_i},\ldots,\mathbf{v}_{\pi_n}) \;=\; \mathbf{APV}(\mathbf{v}_i,\ldots,\mathbf{v}_n)\,. \qquad \textbf{(P1)}$$

Relation to PV. For $n=2$, **fAPV** and **PV** coincide:

$$\mathbf{fAPV}(\mathbf{v}_1,\mathbf{v}_2) \;=\; \mathbf{PV}(\mathbf{v}_1,\mathbf{v}_2)\,. \qquad \textbf{(P2)}$$

Dependence on scaling. The **PV** operator is invariant to scaling, i.e., $\mathbf{PV}(\mathbf{v}_1,\mathbf{v}_2)=\mathbf{PV}(s_1\mathbf{v}_1,s_2\mathbf{v}_2)$ for any nonzero scalars s_1,s_2. By construction, the **APV** operator depends on the scaling of the vector fields as different scales weight their contributions to the covariance matrix $\mathbf{D}\mathbf{D}^{\mathrm{T}}$. We study $\mathbf{APV}(\mathbf{v}_1,\ldots,\mathbf{v}_n,s\mathbf{v}_{n+1})$ for the edge cases $s=0$ and $s\to\infty$:

$$\mathbf{APV}(\mathbf{v}_1,\ldots,\mathbf{v}_n,0\,\mathbf{v}_{n+1}) \;=\; \mathbf{APV}(\mathbf{v}_1,\ldots,\mathbf{v}_n) \qquad \textbf{(P3)}$$

$$\lim_{s\to\infty}\mathbf{APV}(\mathbf{v}_1,\ldots,\mathbf{v}_n,s\,\mathbf{v}_{n+1}) \;=\; \mathbf{PV}(\mathbf{a},\mathbf{v}_{n+1}) \qquad \textbf{(P4)}$$

The second property (**P4**) is remarkable: If one single field is scaled extremely such that it "dominates" all other fields, and both the average **a** as well as **b** converge to $\mathbf{v}_{n+1}$, the **APV** operator gives a well-defined line.

Adding new vector fields. The **APV** operator is invariant to adding a scaled mean vector field or zero fields. For any scalar s:

$$\mathbf{APV}(\mathbf{v}_1,\ldots,\mathbf{v}_n,s\sum_i\mathbf{v}_i) \;=\; \mathbf{APV}(\mathbf{v}_1,\ldots,\mathbf{v}_n) \qquad \textbf{(P5)}$$

$$\mathbf{APV}(\mathbf{v}_1,\ldots,\mathbf{v}_n,\mathbf{0},\ldots,\mathbf{0}) \;=\; \mathbf{APV}(\mathbf{v}_1,\ldots,\mathbf{v}_n) \qquad \textbf{(P6)}$$

If we add the same field **w** extremely often, **APV** still yields a well-defined result:

$$\lim_{k\to\infty}\mathbf{APV}(\mathbf{v}_1,\ldots,\mathbf{v}_n,\underbrace{\mathbf{w},\ldots,\mathbf{w}}_{k\text{ times}}) \;=\mathbf{PV}(\mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{w}-n\,||\mathbf{w}||^2\,\mathbf{a},\mathbf{w})\,. \qquad \textbf{(P7)}$$

5. Discretization and Visualization

All datasets are given as sets of piecewise linear vector fields that are defined w.r.t. a tetrahedral partition of the domain. We apply the **APV** operator on all triangular faces of the tetrahedra. This way, we find point locations on faces that are connected by line segments within tetrahedra, which gives discrete **APV** lines. This is the same modus operandi as for parallel vectors **PV**.

At each feature point location **x** we can quantify the "spread" of vectors. For eigenvalues $\lambda_1\le\lambda_2\le\lambda_3$ of $\mathbf{D}\mathbf{D}^{\mathrm{T}}$, we measure the ratio

$$\varepsilon \;=\; \frac{\lambda_3-\lambda_2}{\lambda_1+\lambda_2+\lambda_3}\cdot\chi \quad\text{with}\quad \chi=\begin{cases}+1 & \text{if } \mathbf{x}\in\mathbf{fAPV}\\ -1 & \text{else}\end{cases}$$

We use the additional sign χ to distinguish between locations that are part of the filtered **fAPV** and locations the are non-filtered **APV** features but not part of **fAPV**. For the latter the mean vector is aligned to one of the minor eigenvectors corresponding to λ_1 or λ_2. We color code ε (see Figure 6) and we place spheres in regions where $\varepsilon\approx 0$, i.e., $\lambda_3\approx\lambda_2$ and thus the major eigenvector is undefined. As **PV** feature lines always form closed lines, so do **APV** lines. Figure 7 compares the different filtering options:

- Figure 7 (a) shows *all* closed feature lines *without* filtering.
- In Figure 7 (b), connected components are discarded if $\varepsilon<0$ for *all* locations, i.e., the remaining lines contain at least one **fAPV** location ($\varepsilon>0$). This is a non-local filter criterion on structures that maintains closed lines.
- In Figure 7 (c), all line segments which are spanned by a location with $\varepsilon<0$ are discarded. This is essentially the "pointwise" **fAPV** filter, which generally yields a set of open feature lines.

6. Applications and Results

In the following, we demonstrate the approximate parallel vector fields operator. We start with an analytic ensemble and then examine three different ensemble datasets from numerical simulations. For all shown applications, the data consists of a number of 3D velocity fields $\mathbf{v}_1,\ldots,\mathbf{v}_n$. We use the **APV** operator to analyze them in two ways:

- To analyze the alignment of the velocity fields, we compute $\mathbf{APV}(\mathbf{v}_1,\ldots,\mathbf{v}_n)$. This gives corelines of best alignment of the velocity fields. Along these lines the ensemble members have a locally maximal parallelity to each other.
- To analyze the alignment of the vortex core lines of all fields, we additionally consider the accelerations fields $\mathbf{c}_1=(\nabla\mathbf{v}_i)\,\mathbf{v}_i$ of all ensemble members. Instead of computing the vortex core lines $\mathbf{v}_i\parallel\mathbf{c}_i$ of each ensemble member, our approach gives the best approximated vortex core lines of all fields by considering $\mathbf{APV}(\mathbf{v}_1,\ldots,\mathbf{v}_n,\mathbf{c}_1,\ldots,\mathbf{c}_n)$.

The acceleration fields are estimated on the same tetrahedral partition as the velocity fields and represented as linear pieces in the tetrahedral cells.

6.1. Linear Vector Field Ensemble

A family of linear vector fields is given by

$$v(x,y,z) \;=\; \begin{pmatrix}0 & a & 0\\ -a & 0 & 0\\ 0 & 0 & b\end{pmatrix}\begin{pmatrix}x+x_0\\ y+y_0\\ z+z_0\end{pmatrix}.$$

We created an ensemble of 250 velocity fields with randomly chosen parameters $a,b,x_0,y_0,z_0\in[-1,1]$. Then each member describes a rotational flow around a vertical core line with random location, rotational speed and direction. Figure 3 displays some streamlines. For each member and its acceleration, their **PV** lines are vertical

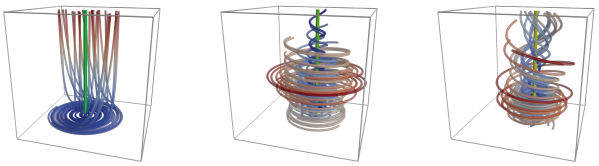


Figure 3: *Streamlines and corresponding **PV** lines for three different members of an ensemble of random linear flows rotating around a vertical axis with varying locations, rotational direction and speed.*

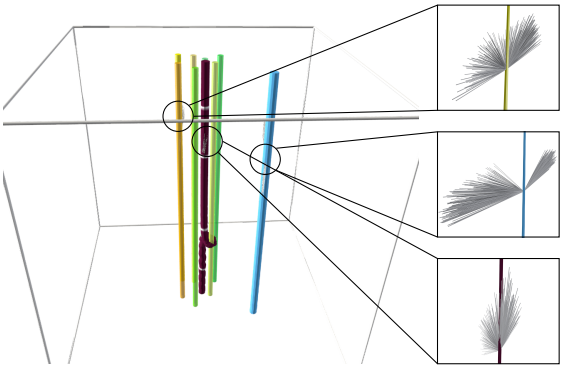


Figure 4: *Several **PV** feature lines of the ensemble are shown in different shades of green. The blue **PV** line of the mean velocity and its acceleration appears at a significant offset, while the red **APV** line is centered within the members' **PV** lines and runs closely through the origin. The close-ups right display velocity and acceleration vectors of all members displayed at three distinct points on a member, on **PV** and on **APV** The alignment of vectors seems best for the **APV** sample.*

lines that intersect the x-y-plane around the origin. Figure 4 shows seven of these **PV** lines, colored in different shades of green.

A naïve approach to finding feature lines of the ensemble, consists in applying the standard **PV** operator to the mean vector field and its acceleration field. In the example, this gives the blue line in Figure 4, which appears at a significant offset from the individual members' **PV** lines. In contrast, the **APV** line – displayed in red – intersects the x-y-plane as expected close to the origin, i.e., which is obviously the better representative or "mean feature line". The reason for the misalignment of the blue line lies in the fact that averaging the ensemble members results in a field with an ill-conditioned Jacobian, i.e., the matrix' determinant is close to zero. Finding extremal lines in linear vector fields with such Jacobians gives unstable results.

In order to compare and demonstrate alignment, we display glyphs for velocity and acceleration vectors of all members sampled at three different locations: on a **PV** solution line of a single member, on the **PV** line of the mean velocity field, and on the **APV** line. The **APV** solution indeed shows the smaller spread and hence the better alignment of vectors. This is indicator for the plausibility of our approach.

Figure 5 shows two scalar fields that are derived from the velocity

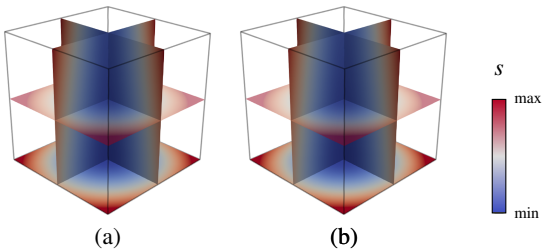


Figure 5: *Scalar fields s that measure local alignment of ensemble members. (a) Accumulated norm of the cross product of all vectors at a given location (see (4)). (b) Accumulated angle of all vectors at a given location (see (3)). Both fields indicate that locations of high alignment of all vectors lie vertically in the center of the domain.*

ensemble: the accumulated angles (3) and the accumulated norm of the cross products (4) and provide an alternative measure for "how parallel" vectors are at a domain point. The lowest values in both fields are found in the center of the domain around a vertical line which coincides with the location of the **APV** feature line. This fact again indicates plausibility of **APV** features. We will discuss in Section 7.2, why we prefer the definition of **APV** over alternative concepts of approximately parallel line features.

6.2. Aneurysm Ensemble

The Computational Fluid Dynamics Rapture Challenge 2013 [BRB*15, JBS*15] presented a velocity field ensemble data that was created by different hemodynamics simulations inside an aneurysm geometry. Eight different blood flow fields were simulated by varying the outlet boundary conditions. This included a zero-pressure condition for both outlets as well as seven simulations, where the pressure was split between the outlets and changed in steps of 10% from 80% to 20% and vice versa. Figure 1 shows streamlines of blood flow of each of the eight members of the ensemble as well as the feature lines extracted with the standard **PV**operator. Each member has a specific color assigned ranging from green to yellow so the corresponding feature line can be located in the combined spaghetti plot visualization on the right.

Figure 6 (a) shows **APV** feature lines derived from eight ensemble members. Near the inlet (top) there is no significant difference between members, and therefore many insignificant features are found. As the flow progresses, the members start to divert. The closeups (b)-(d) show single point locations $\mathbf{x}$ of different regions with vectors at $\mathbf{x}$ drawn as arrows: (b) $\mathbf{x} \in$ **fAPV**, (c) $\mathbf{x} \notin$ **APV**, and (d) $\mathbf{x} \in$ **APV** but $\mathbf{x} \notin$ **fAPV**. The latter is a mean vector aligned with a minor eigenvector and would be removed by filtering. The resulting feature lines are shown in Figure 6 (e).

The resulting **APV** lines give the locations, where all **PV** lines of the ensemble are best aligned. This behavior can be observed in Figure 8: We observe high ε near the locations where all **PV** lines are close to each other. However, in regions where the **PV** lines start to diverge from each other, ε decreases and eventually turns negative. In these locations, the mean vector is aligned with a minor eigenvector as seen in Figure 6 (d). Finally, we remark

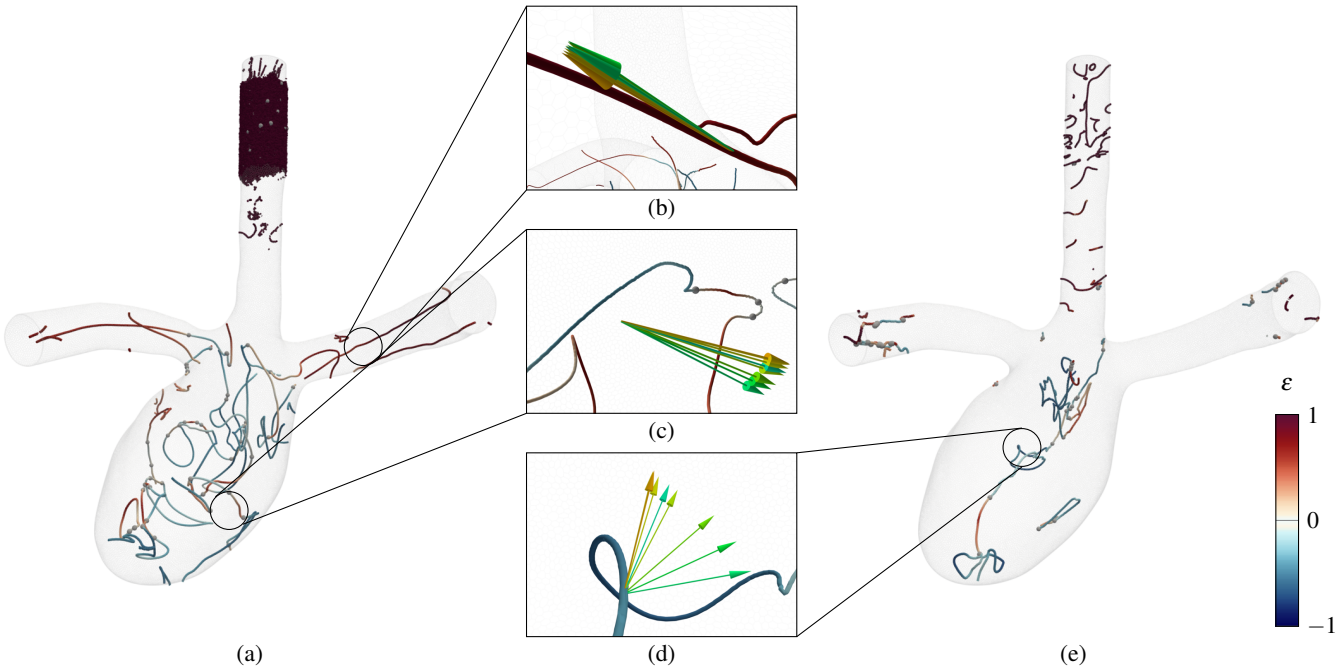


Figure 6: ***APV*** *lines for aneurysm ensemble: (a) using all eight velocity fields. (e) using all eight velocity fields and their acceleration fields. The closeups give examples for the color coding (right): (b) Locations with input vectors aligned closely with the major eigenvector are depicted in red color. (c) Is an example of a non-feature location. (d) Blue lines refer to locations with vectors aligned closely to one of the minor eigenvalues. They would be removed in filtered* ***fAPV*** *in Figure 7.*

Figure 7: *The* ***APV*** *operator finds structures of closed lines, which can be filtered: (a)* ***APV*** *lines w/o filtering. (b) Only lines with at least one* ***fAPV*** *segment. (c) Structures of* ***fAPV*** *line segments may be non-closed.*

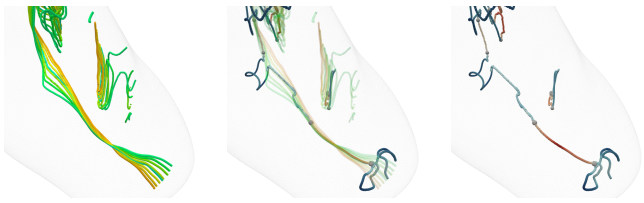


Figure 8: *(a) Spaghetti plot of all* ***PV*** *feature lines. (b) The* ***APV*** *operator finds locations where the* ***PV*** *feature lines are close to each other. (c)* ***APV*** *lines of all velocity and acceleration fields.*

that computing **PV** lines for each ensemble member and its derived acceleration field consumes significantly more time than one single application of the **APV** operator for the same data. To reduce visual clutter, we apply filtering as seen in Figure 7.

6.3. Helicopter in Ground Proximity

This dataset by Kutz et al. [KKKK12] simulates wind flow near a helicopter that is hovering over ground. Figure 9 shows streamlines for a single time step. We sampled the flow field uniformly in time such that the rotor revolution increases by 10° for each time step and collect six time steps in total in an ensemble. The difference between the ensemble members is relatively small. This can be seen by computing **PV** features for each member and its derived acceleration field as depicted in Figure 11 (a). Figure 10 (a) shows all **PV** feature lines in a combined visualization. We compute **APV** features on all velocity fields of the ensemble. This is shown in Figures 10 (b) and (c). Figures (d) and (f) include in addition the derived acceleration fields.

Note that this dataset is special, due to the lack of variance between **PV** features. In this case the **APV** features resemble the "mean locations" of **PV** features (although it is unclear how to av-

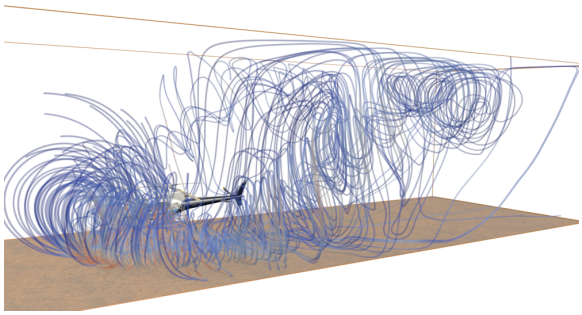


Figure 9: *Streamlines of the wind flow around a hovering helicopter for a fixed time step. Swirling behavior can be seen behind the helicopter.*

erage core lines). Indeed, the comparison of **PV** and **APV** features in Figure 11 suggests that the latter express the essence of the "ensemble" of **PV** core lines. The additional streamlines for one single ensemble member in Figure 10 (d) suggests that **APV** lines can yield locations similar to vortex core lines.

6.4. Rotating Mixer

The last ensemble was created by sampling a CFD simulation of flow inside a container with a rotating mixer. Six time steps were chosen such that the three blades inside the container rotate by 120° each time such that the blade geometry overlaps exactly for each ensemble member. This is a turbulent flow and the members vary greatly. The **PV** feature lines for one single member's velocity and acceleration shows already a complex behavior, which makes it difficult to identify interesting structures. This is shown in Figure 13 (a). Figure 13 (b) shows **PV** feature lines for all members. We compute **APV** features for velocity and acceleration fields. The result is shown in Figure 13 (c). Filtering reduces the feature regions significantly: 13 (c) shows **fAPV** features. The colors in (d) show regions with low "spread" of ensemble members near the blades of the mixer. Figures (e) and (f) show **APV** and **fAPV** features for only the given velocity members without additional acceleration fields.

6.5. Performance

Table 1 summarizes the sizes of the ensemble data and timings for feature extraction. All times were measured for an Intel Core i7-6700K CPU at 4GHz with 32GB RAM available, this was always enough memory to store the data. All algorithms were implemented in C++. The time that is required for computing the derived fields $\mathbf{a}$ and $\mathbf{b} = \mathbf{D}\mathbf{D}^{\mathrm{T}}\mathbf{a}$ depends linearly of input fields. For a moderate number of ensemble members this cost is not significant compared to the next step: The time for the subsequent computation of $\mathbf{PV}(\mathbf{a},\mathbf{b})$ for all triangular faces is constant for the given tetrahedral partition. Filtering (**fAPV** versus **APV**) and color coding require a spectral decomposition of $\mathbf{D}\mathbf{D}^{\mathrm{T}}$ at feature locations. The cost for their computation is negligible as the number of feature locations is small compared to the total number of triangular faces.

7. Discussion and Comparison

The **APV** operator computes core lines for ensembles of velocity fields. In this section, we compare the new approach with other algorithms for computing core lines for flow ensemble data.

7.1. Multiple line sets

We may treat each input field individually and compute core lines by applying the standard **PV** operator. Then a number of visual representations could be considered:

- One option is a simple *spaghetti plot* [CHL13]. Their limitations are discussed in several recent publications [WMK13, MWK14, FKRW16, FBW16].
- Another option is *clustering* line features and selection of representatives. There are different clustering methods. They all require the notion of distance, and many different distance measures are available. The reasons why these approaches are not applicable to our problem lies in the fact that the line features that result from the ensemble members may differ significantly in shape and topology: they may consist of several unconnected parts (due to the filtering of the **PV** lines), or some of the ensemble members may give no core lines at all. There is no straightforward answer to the question how to incorporate these cases into a stable clustering algorithm. While clustering algorithms have been successfully applied to sets of stream lines or path lines [OLK*14, OCJP16, RT12], we are not aware of existing work of line clustering of vortex core lines in ensemble flows.
- A *curve boxplot* approach similar to [WMK13, MWK14] may be applied to the set of core lines. These approaches rely either on implicit curve representations [WMK13] or on a common parametrization of the curves [MWK14]. Such parametrization does not exist for **PV** lines as input curves. Moreover, due to the unconnectedness and even non-existence of core lines in individual fields, we are not aware of straightforward approaches to construct a common parametrization. We conclude that also curve boxplots are not directly applicable to our problem.
- *Variability plots* [FKRW16, FBW16] may be applied to the sets of core lines. Similar to curve boxplots, variability plots either require implicit curve representations [FKRW16] or common parameterizations [FBW16]. And similarly, we are not aware of straightforward extensions of variability plots to sets of core lines.

7.2. Lines on derived fields

As introduced in Section 6.1, in order to find core lines of multiple fields, one may also consider *extremal curves* in a scalar field s derived from $\mathbf{v}_1,\ldots,\mathbf{v}_n$ such that summed angles or cross products of each pair of vectors are considered as a measure of how "parallel" the n fields are. The scalar field could be chosen as

$$s = \sum_{i=1}^{n}\sum_{j=1}^{n} |\angle(\mathbf{v}_i,\mathbf{v}_j)|, \quad \text{or} \tag{3}$$

$$s = \sum_{i=1}^{n}\sum_{j=1}^{n} \|\mathbf{v}_i \times \mathbf{v}_j\|. \tag{4}$$

Ridge lines of s may also be interpreted as lines of "maximal parallelity" of $\mathbf{v}_1,\ldots,\mathbf{v}_n$. However, we see the following potential problems with this approach:

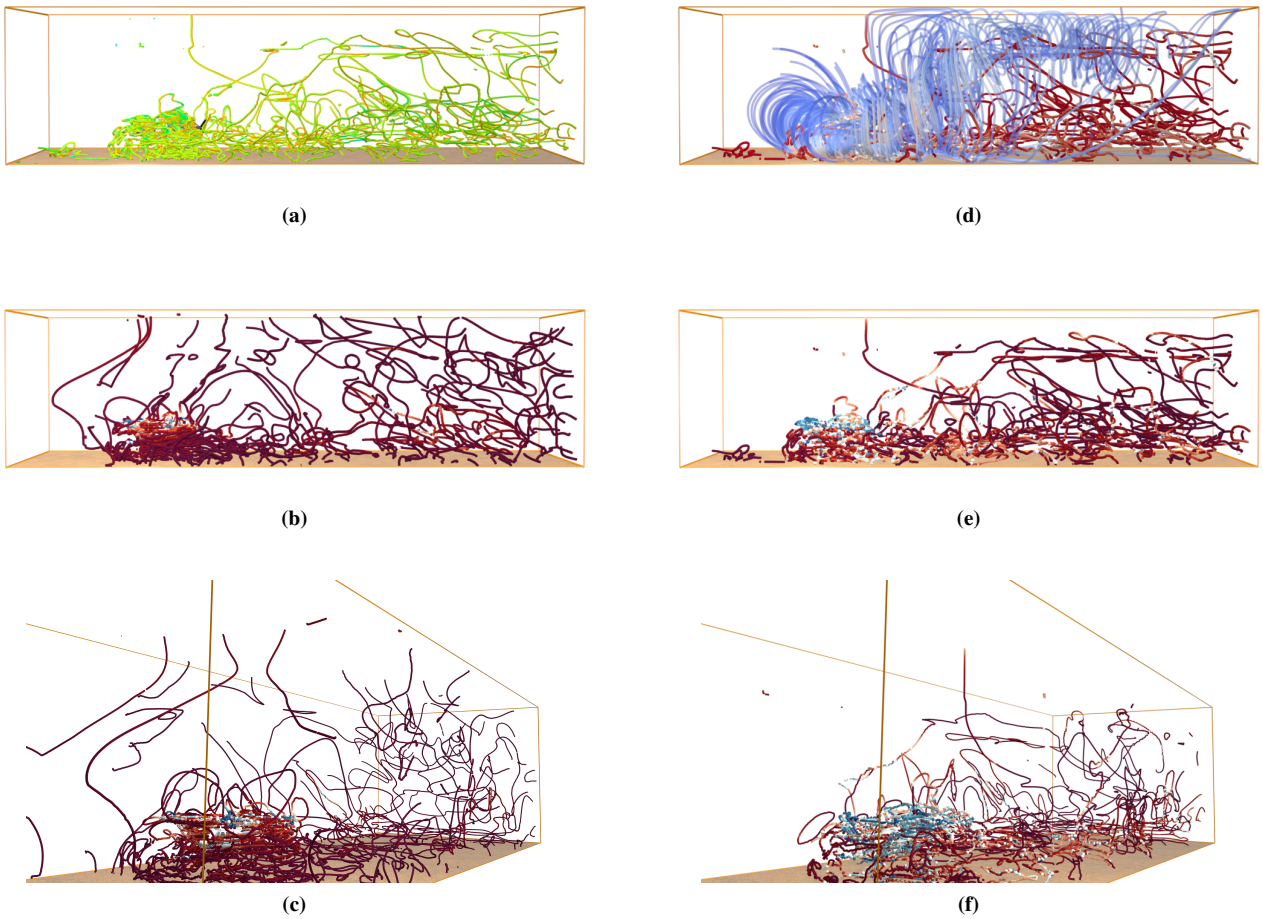


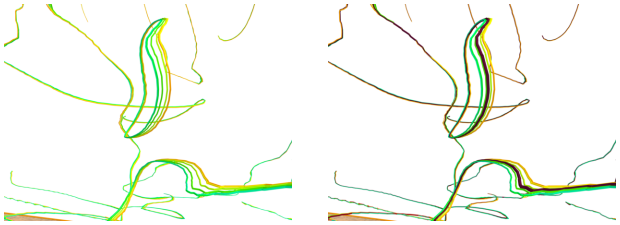
Figure 10: *Helicopter dataset. (a) All **PV** core lines from each ensemble member displayed in different shades of green. They lie very close to each other. (b)-(c) **APV** features for all velocity fields and derived acceleration fields. (d)-(f): **APV** features for only the velocity members. (d) Additionally shows streamlines for one of the members.*

Dataset	#Fields	#Vertices	#Vectors	**APV**	**PV**
Aneurysm Velocity	8	3,501,487	28,011,896	974ms	261,571ms
Aneurysm Velocity + Acceleration	16	3,501,487	56,023,792	1,538ms	261,668ms
Helicopter Velocity	6	4,810,000	28,860,000	1,231ms	310,260ms
Helicopter Velocity + Acceleration	12	4,810,000	57,720,000	1,884ms	288,672ms
Mixer Velocity	6	1,258,759	7,552,554	312ms	99,083ms
Mixer Velocity + Acceleration	12	1,258,759	15,105,108	465ms	97,529ms

Table 1: *Timings for the given ensemble data. The column **APV** refers to the computation of the derived fields, the mean* $\mathbf{a}$ *and* $\mathbf{b} = \mathbf{D}\mathbf{D}^{\mathrm{T}}\mathbf{a}$*. The **PV** column refers to the extraction of* $\boldsymbol{PV}(\mathbf{a}, \mathbf{b})$.

- *Scaling:* (3) gives unstable results in areas of small vectors $\mathbf{v}_i$.
- *Numerical Stability:* The numerical ridge extraction requires the gradient and Hessian of s, e.g., as input to the **PV** operator. The use of, possibly estimated, first and second order derivatives makes the feature extraction significantly more sensitive to noise in the data. Note that alternative ridge extraction methods are, although well-understood, generally less stable than applying the **PV** operator, because ridge extraction is based on searching for local extrema. In contrast, **PV** is based on searching for zero crossings of functions. In regions of strongly varying fields, zero crossing is less prone to missing results than searching for extrema.

We are not aware of any existing approaches of this category that find core lines of multiple velocity fields.



(a) (b)

Figure 11: *(a) A combined visualization of all core lines extracted with the **PV** operator from each ensemble member and its derived acceleration field. (b) Same with an additional overlay of **APV** feature lines for the same input data.*

Figure 12: *Streamlines for one time step of rotating mixer dataset.*

8. Limitations and Future Work

APV feature lines provide an intuitive interpretation for locations where the mean vector is aligned with the major eigenvector. In the visualization, this was shown in red ($\varepsilon > 0$, and ideally $\varepsilon \to 1$). For locations where we observe the alignment with one of the minor eigenvectors – these are missing in **fAPV** – there is no such interpretation (shown in blue as $\varepsilon < 0$). This is the reason why we offer a filter for them. It may be worthwhile to try to derive some information that is meaningful and helps understanding the data also from these locations.

In the future, the presented approach could be extended to an out-of-core method for ensembles with very large numbers of members such that members can be "streamed" into memory. This requires only an "online" update of the mean $\mathbf{a}$ and the matrix $\mathbf{D}\mathbf{D}^{\mathrm{T}}$ such that only mean vectors and covariance matrices are held in main memory. This type of update is certainly possible, however, the straightforward formulas suffer significantly from numerical round-off errors. An out-of-core method would require finding a balance between numerical efficiency and sufficient accuracy.

Finally, the domain of applications can be extended by applying **APV** to fields other than velocity and acceleration such as second derivatives of particle trajectories, or pressure gradients.

Appendix

In the following, we give proofs for the properties (P1)-(P7) summarized in Section 4.

Property (P1) holds by construction as neither the mean vector $\mathbf{a}$ nor the symmetric operator $\mathbf{D}\mathbf{D}^{\mathrm{T}} = \sum_{i=1}^{n}(\mathbf{v}_i - \mathbf{a})(\mathbf{v}_i - \mathbf{a})^{\mathrm{T}}$ (and hence $\mathbf{b}$) change on permutation of the columns $\mathbf{v}_i$ of $\mathbf{V}$. □

Property (P2). Assume the *parallel vectors condition* holds for $\mathbf{v}_1$ and $\mathbf{v}_2$, i.e., $\mathbf{v}_1 = \lambda\mathbf{v}_2$. Then with $\mathbf{V} = (\mathbf{v}_1|\lambda\mathbf{v}_1)$, $\mathbf{a} = \frac{1+\lambda}{2}\mathbf{v}_1$ and

$$\mathbf{D}\mathbf{D}^{\mathrm{T}} \;=\; (1+\lambda^2)\mathbf{v}_1\mathbf{v}_1{}^{\mathrm{T}} - \tfrac{1}{2}(1-\lambda)^2\mathbf{v}_1\mathbf{v}_1{}^{\mathrm{T}} \;=:\; \tilde{\lambda}\,\mathbf{v}_1\mathbf{v}_1{}^{\mathrm{T}}\,.$$

This implies $\mathbf{b} = \mathbf{D}\mathbf{D}^{\mathrm{T}}\mathbf{a} = \frac{1+\lambda}{2}\tilde{\lambda}\mathbf{a}$, or equally the **APV** condition. The symmetric matrix $\mathbf{D}\mathbf{D}^{\mathrm{T}}$ has rank 1 and one single nonzero eigenvalue $\tilde{\lambda}\mathbf{v}_1{}^{\mathrm{T}}\mathbf{v}_1$ therefore also the **filtered APV** condition holds. The reverse direction (**fAPV** $\Rightarrow$ **PV**) is straightforward to show using the same argument. □

The proofs of the remaining, more general properties are slightly more complex. The essential argument, however, is similar.

Let $\mathbf{1}_n \in \mathbb{R}^n$ denote the constant vector with all entries equal to 1, and let $\mathbb{1}_n = \mathbf{1}_n\mathbf{1}_n^{\mathrm{T}} \in \mathbb{R}^{n\times n}$ denote the constant matrix with all entries 1. For the sake of a concise notation, we may omit the dimension n whenever it is irrelevant or clear from the context.

Let $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_n)$ with $\mathbf{v}_i \in \mathbb{R}^m$. Then $\mathbf{a} = \frac{1}{n}\mathbf{V}\mathbf{1}_n$ the *mean* of column vectors for a matrix $\mathbf{V} \in \mathbb{R}^{m\times n}$, and $\frac{1}{n}\mathbf{V}\mathbb{1}_n$ gives the matrix $(\mathbf{a}, \ldots, \mathbf{a}) \in \mathbb{R}^{m\times n}$. So far we considered (w.l.o.g.) the 3D case $m = 3$. The following arguments apply for any dimension $m \geq 2$.

The proposed algorithm requires the matrix $\mathbf{D} = \mathbf{V} - \frac{1}{n}\mathbf{V}\mathbb{1}_n = \mathbf{V}\mathbf{P}$ with $\mathbf{P} = \mathbf{I} - \frac{1}{n}\mathbb{1}_n$. ($\mathbf{I}$ is the identity.)

In a slightly more general setting, we first summarize properties of symmetric matrices $\mathbf{P}_n(\alpha) := \mathbf{I} - \alpha\mathbb{1}_n$.

Lemma 1 (Spectral decomposition)**.** *For $\alpha > 0$, matrices $\mathbf{P}_n(\alpha)$ have* eigenvalues $\lambda_i = 1 - \alpha\mu_i$ *for* $\mu_i = 0, \ldots, 0, n$, *and they share* eigenvectors *up to order and orientation.*

Proof. The following holds for eigenvalues λ and eigenvectors $\mathbf{x}$.

$$(\mathbf{I} - \alpha\mathbb{1})\mathbf{x} = \lambda\mathbf{x} \;\Leftrightarrow\; \mathbf{x} - \alpha\mathbb{1}\mathbf{x} = \lambda\mathbf{x} \;\Leftrightarrow\; \alpha\mathbb{1}\mathbf{x} = (1-\lambda)\mathbf{x}$$
$$\mathbb{1}\mathbf{x} = \mu\mathbf{x} \quad \text{with eigenvalues} \quad \mu = \tfrac{1}{\alpha}(1-\lambda)$$

The matrix $\mathbb{1} = \mathbb{1}_n$ has rank 1 and eigenvalue 0 with multiplicity $n-1$, and the remaining eigenvalue must be equal to n. It is straightforward to confirm that the first $n-1$ *eigenvectors* of $\mathbb{1}_n$ provide an orthonormal basis of the kernel $\{\mathbf{x}\,|\,\mathbb{1}_n\mathbf{x} = \mathbf{0}\}$, and the remaining last (unit length) eigenvector is the constant vector $\mathbf{q} := \frac{1}{\sqrt{n}}\mathbf{1}_n$. The eigenvalues are shifted for $\mathbf{P}(\alpha)$, and the eigenvectors are the same as for $\mathbb{1}$ up to order and orientation. □

Note that the constant eigenvector $\mathbf{q}$ of $\mathbf{P}(\alpha)$ appears for the smallest eigenvalue due to the shift of eigenvalues. In the following, we write the eigenvectors as columns of the orthogonal matrix $\mathbf{Q} = (\mathbf{q}|\hat{\mathbf{Q}})$, where the columns of $\hat{\mathbf{Q}}$ span the kernel of $\mathbb{1}$ (or equally the image of $\mathbf{P}(\alpha)$).

For the *special case* $\mathbf{P} = \mathbf{I} - \frac{1}{n}\mathbb{1}_n$, this gives eigenvalues $\lambda_i = 0, 1, \ldots, 1$. The symmetric matrix $\mathbf{P}$ has rank $n-1$ and is *idempotent* (i.e., $\mathbf{P}\mathbf{P} = \mathbf{P}$). It can be written in terms of its spectral decomposition as $\mathbf{P} = \mathbf{Q}\,\mathrm{diag}(0, 1, \ldots, 1)\,\mathbf{Q}^{\mathrm{T}} = \hat{\mathbf{Q}}\hat{\mathbf{Q}}^{\mathrm{T}}$.

In the following, we consider the cases of n and $n+1$ vectors:

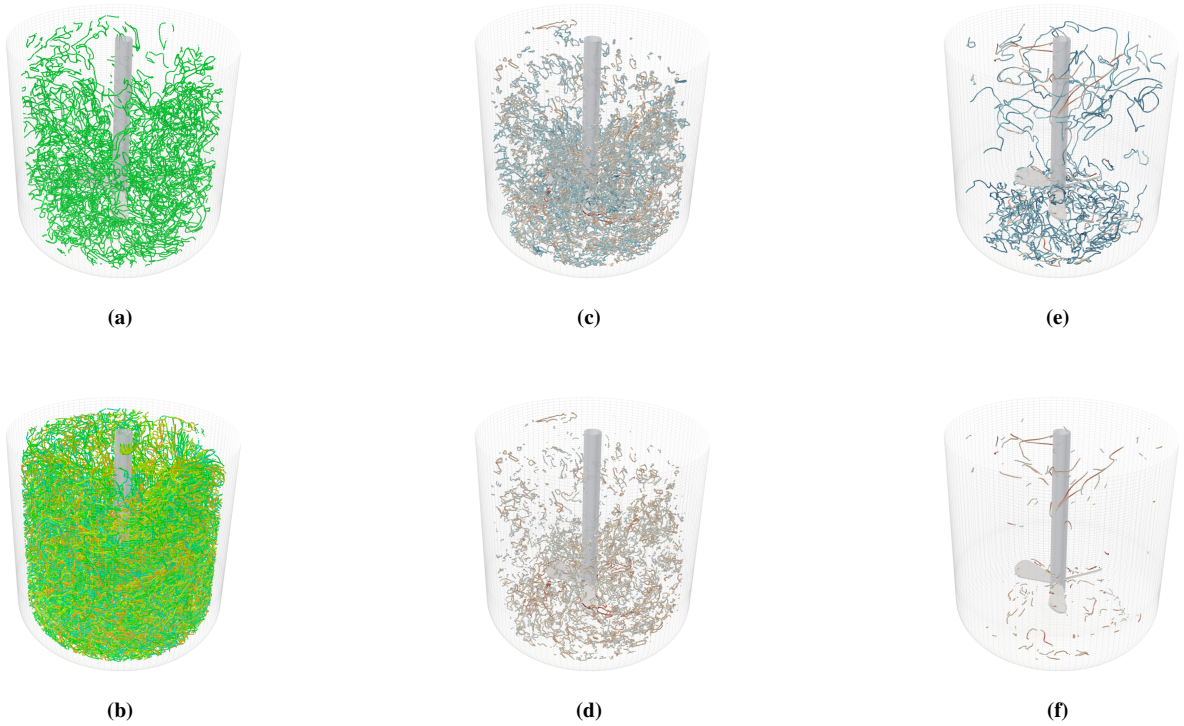


Figure 13: *Flow inside a rotating mixer. Left: (a) **PV** core lines extracted from one member and the derived acceleration field. (b) Combined visualization of all **PV** core lines extracted from each ensemble member and derived acceleration fields. Center: **APV** features for the same data, unfiltered (c) and filtered (d). Right: **APV** features for velocity members only, unfiltered (e) and filtered (f).*

Let $\mathbf{V} = (\mathbf{v}_1|\dots|\mathbf{v}_n) \in \mathbb{R}^{3\times n}$ and $\bar{\mathbf{V}} = (\mathbf{v}_1|\dots|\mathbf{v}_{n+1}) \in \mathbb{R}^{3\times(n+1)}$. Likewise, we denote

$$\begin{aligned} n\,\mathbf{a} &= \mathbf{V}\mathbf{1}_n \quad \text{and} \quad \mathbf{b} = \mathbf{D}\mathbf{D}^{\mathrm{T}}\mathbf{a} = \mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}}\mathbf{V}\mathbb{1}_n, \quad \text{and} \\ (n+1)\,\bar{\mathbf{a}} &= \bar{\mathbf{V}}\mathbf{1}_{n+1} \quad \text{and} \quad \bar{\mathbf{b}} = \bar{\mathbf{D}}\bar{\mathbf{D}}^{\mathrm{T}}\bar{\mathbf{a}} = \bar{\mathbf{V}}\bar{\mathbf{P}}\bar{\mathbf{V}}^{\mathrm{T}}\bar{\mathbf{V}}\mathbb{1}_{n+1}\,. \end{aligned}$$

For the sake of a concise notation, the bar over a quantity denotes dimension $n+1$, e.g., $\bar{\mathbf{D}} \in \mathbb{R}^{3\times(n+1)}$ and $\bar{\mathbf{P}} \in \mathbb{R}^{(n+1)\times(n+1)}$, and we omit explicit subscripts.

We show how the *parallel vectors condition* $\bar{\mathbf{a}} = \lambda\bar{\mathbf{b}}$ can be expressed in terms of $\mathbf{a}$ and $\mathbf{b}$ using block decompositions of $\bar{\mathbf{V}} = (\mathbf{V}|\mathbf{v}_{n+1})$ and

$$\bar{\mathbf{P}} = \left(\begin{array}{c|c} \hat{\mathbf{P}} & -\frac{1}{n+1}\mathbf{1} \\ \hline -\frac{1}{n+1}\mathbf{1}^{\mathrm{T}} & \frac{n}{n+1} \end{array}\right) \quad \text{with} \quad \hat{\mathbf{P}} = \mathbf{P}_n(\tfrac{1}{n+1}) = \mathbf{I} - \tfrac{1}{n+1}\mathbb{1}_n\,.$$

Then

$$\begin{aligned} \bar{\mathbf{D}}\bar{\mathbf{D}}^{\mathrm{T}} = \bar{\mathbf{V}}\bar{\mathbf{P}}\bar{\mathbf{V}}^{\mathrm{T}} &= \left(\mathbf{V}\hat{\mathbf{P}} - \tfrac{1}{n+1}\mathbf{v}_{n+1}\mathbf{1}^{\mathrm{T}} \;\middle|\; \tfrac{n}{n+1}\mathbf{v}_{n+1} - \tfrac{1}{n+1}\mathbf{V}\mathbf{1}\right)(\mathbf{V}|\mathbf{v}_{n+1})^{\mathrm{T}} \\ &= \mathbf{V}\hat{\mathbf{P}}\mathbf{V}^{\mathrm{T}} - \tfrac{1}{n+1}\mathbf{v}_{n+1}\mathbf{1}^{\mathrm{T}}\mathbf{V}^{\mathrm{T}} + \tfrac{1}{n+1}\left(n\mathbf{v}_{n+1} - \mathbf{V}\mathbf{1}\right)\mathbf{v}_{n+1}^{\mathrm{T}}\,. \end{aligned} \quad (5)$$

Using Lemma 1, we can express $\hat{\mathbf{P}}$ as a rank-1 update of $\mathbf{P}$ — the zero eigenvalue becomes $\frac{1}{n+1}$ — and obtain

$$\hat{\mathbf{P}} \;=\; \mathbf{P} + \tfrac{1}{n+1}\mathbf{q}\mathbf{q}^{\mathrm{T}} \;=\; \mathbf{P} + \tfrac{1}{(n+1)n}\mathbb{1}\,.$$

We use (5) to show properties (P3), (P4) and (P5), and we use a similar block decomposition to show (P7).

Property (P3). Let $\mathbf{v}_{n+1} = \mathbf{0}$. Then (5) reduces to $\bar{\mathbf{D}}\bar{\mathbf{D}}^{\mathrm{T}} = \mathbf{V}\hat{\mathbf{P}}\mathbf{V}^{\mathrm{T}}$. With $(n+1)\,\bar{\mathbf{a}} = n\,\mathbf{a}$ the *parallel vectors condition* $\bar{\mathbf{b}} = \lambda\bar{\mathbf{a}}$ becomes

$$\begin{aligned} \mathbf{V}\hat{\mathbf{P}}\mathbf{V}^{\mathrm{T}}\mathbf{V}\mathbf{1} &= \lambda\mathbf{V}\mathbf{1} \quad \Leftrightarrow \\ \mathbf{V}\left(\mathbf{P} + \tfrac{1}{(n+1)n}\mathbb{1}\right)\mathbf{V}^{\mathrm{T}}\mathbf{V}\mathbf{1} &= \lambda\mathbf{V}\mathbf{1} \quad \Leftrightarrow \\ \mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}}\mathbf{V}\mathbf{1} \;+\; \tfrac{1}{(n+1)n}\mathbf{V}\mathbb{1}\mathbf{V}^{\mathrm{T}}\mathbf{V}\mathbf{1} &= \lambda\mathbf{V}\mathbf{1} \quad \Leftrightarrow \\ \mathbf{b} \;+\; \tfrac{1}{(n+1)n}\mathbf{V}\mathbb{1}\mathbf{V}^{\mathrm{T}}\mathbf{a} &= \lambda\mathbf{a}\,. \end{aligned}$$

Assume that $\mathbf{b} = \mu\mathbf{a}$ holds. Then there exists an eigenvalue

$$\lambda = \mu + \frac{n}{n+1}\mathbf{a}^{\mathrm{T}}\mathbf{a}$$

for which the above equation becomes true because

$$\mathbf{b} + \tfrac{1}{(n+1)n}\mathbf{V}\mathbb{1}\mathbf{V}^{\mathrm{T}}\mathbf{a} = \lambda\mathbf{a} \quad \Leftrightarrow \quad \mu\mathbf{a} + \tfrac{1}{(n+1)n}n^2\mathbf{a}(\mathbf{a}^{\mathrm{T}}\mathbf{a}) = \lambda\mathbf{a}\,.$$

It is straightforward to show the reverse by fixing λ rather than μ. □

Property (P4). Assume $\bar{\mathbf{b}} = \lambda\bar{\mathbf{a}}$ holds, i.e., an eigenvalue λ exists. We consider the the left-hand-side of the equation $\bar{\mathbf{b}} - \lambda\bar{\mathbf{a}} = \mathbf{0}$ and

substitute $\mathbf{v}_{n+1} = s\,\mathbf{v}$. With $\mathbf{V1} = n\,\mathbf{a}$, this gives

$$\begin{aligned}\bar{\mathbf{b}} - \lambda\bar{\mathbf{a}} =& \tfrac{1}{n+1}\bar{\mathbf{D}}\bar{\mathbf{D}}^{\mathrm{T}}(n\mathbf{a}+s\mathbf{v}) - \lambda\tfrac{1}{n+1}(n\mathbf{a}+s\mathbf{v})\\ =& \tfrac{n}{n+1}\mathbf{V}\hat{\mathbf{P}}\mathbf{V}^{\mathrm{T}}\mathbf{a} - \tfrac{sn^2}{(n+1)^2}\mathbf{v}\mathbf{a}^{\mathrm{T}}\mathbf{a} - \tfrac{sn^2}{(n+1)^2}\mathbf{a}\mathbf{v}^{\mathrm{T}}\mathbf{a} + \tfrac{s^2n^2}{(n+1)^2}\mathbf{v}\mathbf{v}^{\mathrm{T}}\mathbf{a}\\ &+ \tfrac{s}{n+1}\mathbf{V}\hat{\mathbf{P}}\mathbf{V}^{\mathrm{T}}\mathbf{v} - \tfrac{s^2n}{(n+1)^2}\mathbf{v}\mathbf{a}^{\mathrm{T}}\mathbf{v} - \tfrac{s^2n}{(n+1)^2}\mathbf{a}\mathbf{v}^{\mathrm{T}}\mathbf{v} + \tfrac{s^3n}{(n+1)^2}\mathbf{v}\mathbf{v}^{\mathrm{T}}\mathbf{v}\\ &- \lambda\tfrac{n}{n+1}\mathbf{a} - \lambda\tfrac{s}{n+1}\mathbf{v}\,.\end{aligned}$$

We choose and substitute

$$\lambda = \tfrac{n}{n+1}(s^2 - 2\mu s)\,\mathbf{v}^{\mathrm{T}}\mathbf{v}\,,$$

factor powers of s and obtain

$$\begin{aligned}\bar{\mathbf{b}} - \lambda\bar{\mathbf{a}} =& s^2\left(-\tfrac{n}{(n+1)^2}\mathbf{v}^{\mathrm{T}}\mathbf{v}\mathbf{a} - \tfrac{n^2}{(n+1)^2}\mathbf{v}^{\mathrm{T}}\mathbf{v}\mathbf{a} + \tfrac{n^2}{(n+1)^2}\mathbf{a}^{\mathrm{T}}\mathbf{v}\mathbf{v} - \tfrac{n}{(n+1)^2}\mathbf{a}^{\mathrm{T}}\mathbf{v}\mathbf{v}\right.\\ &\left.+ \tfrac{2\mu n}{(n+1)^2}\mathbf{v}^{\mathrm{T}}\mathbf{v}\mathbf{v}\right) + O(s)\\ =& s^2\tfrac{1}{(1+n)^2}\left((n^2-n)\mathbf{a}^{\mathrm{T}}\mathbf{v}\mathbf{v} - (n^2+n)\mathbf{v}^{\mathrm{T}}\mathbf{v}\mathbf{a} + 2\mu n\mathbf{v}^{\mathrm{T}}\mathbf{v}\mathbf{v}\right) + O(s)\,.\end{aligned}$$

The computation of the above term is somewhat lengthy but elementary. We give a few remarks: First, the choice of eigenvector λ requires a term that is *linear* in s. Second, s^3 is the highest power that appears in the the derivation. However, the cubic terms sum to zero and disappear in the result. (This is independent of the particular choice of λ. There the additional linear term generates a term in s^2 that is required to ensure no solution is lost.)

We can now evaluate the limit $\lim_{s\to\infty}\frac{1}{s^2}(\bar{\mathbf{b}} - \lambda\bar{\mathbf{a}})$ and obtain the equation

$$(n^2-n)\mathbf{a}^{\mathrm{T}}\mathbf{v}\mathbf{v} - (n^2+n)\mathbf{v}^{\mathrm{T}}\mathbf{v}\mathbf{a} + 2\mu n\mathbf{v}^{\mathrm{T}}\mathbf{v}\mathbf{v} = \mathbf{0}\,,$$

which holds iff the *parallel eigenvectors condition* $\mathbf{a} = \mu\mathbf{v}$ is satisfied. □

Property (P5). Let $\mathbf{v}_{n+1} = s\sum_{i=1}^{n}\mathbf{v}_i = s\,\mathbf{V1}$. Then

$$\begin{aligned}\bar{\mathbf{D}}\bar{\mathbf{D}}^{\mathrm{T}} &= \mathbf{V}\hat{\mathbf{P}}\mathbf{V}^{\mathrm{T}} - \tfrac{s}{n+1}\mathbf{V}\mathbb{1}\mathbf{V}^{\mathrm{T}} + \tfrac{s^2n}{n+1}\mathbf{V}\mathbb{1}\mathbf{V}^{\mathrm{T}} - \tfrac{s}{n+1}\mathbf{V}\mathbb{1}\mathbf{V}^{\mathrm{T}}\\ &= \mathbf{V}\hat{\mathbf{P}}\mathbf{V}^{\mathrm{T}} + \frac{n^2(s^2n-2s)}{n+1}\mathbf{a}\mathbf{a}^{\mathrm{T}}\\ &= \mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}} + \tfrac{1}{(n+1)n}\mathbf{V}\mathbb{1}\mathbf{V}^{\mathrm{T}} + \frac{n^2(s^2n-2s)}{n+1}\mathbf{a}\mathbf{a}^{\mathrm{T}}\\ &= \mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}} + \frac{n(1-sn)^2}{n+1}\mathbf{a}\mathbf{a}^{\mathrm{T}}\,.\end{aligned}$$

With

$$\bar{\mathbf{a}} = \tfrac{1}{n+1}(n\mathbf{a}+s\mathbf{V1}) = \tfrac{1}{n+1}(n\mathbf{a}+sn\mathbf{a}) = \tfrac{1+s}{n+1}\mathbf{a}$$

we obtain

$$\bar{\mathbf{D}}\bar{\mathbf{D}}^{\mathrm{T}}\bar{\mathbf{a}} = \lambda\bar{\mathbf{a}} \quad\Leftrightarrow$$

$$\mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}}\mathbf{a} + \frac{n(1-sn)^2}{n+1}\mathbf{a}(\mathbf{a}^{\mathrm{T}}\mathbf{a}) = \lambda\mathbf{a}\,,$$

which holds if $\mathbf{b} = \mu\mathbf{a}$. The argument is similar as for (P3) and uses $\mathbf{b} = \mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}}\mathbf{a} = \mu\mathbf{a}$. □

Property (P6) follows immediately from (P3) as

$$\mathbf{APV}(\mathbf{v}_1,\ldots,\mathbf{v}_n,\underbrace{\mathbf{0},\ldots,\mathbf{0},\mathbf{0}}_{k+1}) = \mathbf{APV}(\mathbf{v}_1\ldots,\mathbf{v}_n,\underbrace{\mathbf{0},\ldots,\mathbf{0}}_{k}) = \ldots = \mathbf{APV}(\mathbf{v}_1\ldots,\mathbf{v}_n)\,.$$

□

Property (P7). Let $\mathbf{W} = \mathbf{w1}_k = (\mathbf{w},\ldots,\mathbf{w}) \in \mathbb{R}^{m\times k}$. We consider $\bar{\mathbf{V}} = (\mathbf{V}|\mathbf{W}) \in \mathbb{R}^{m\times(n+k)}$. Similarly as before we define

$$\bar{\mathbf{P}} = \left(\begin{array}{c|c}\mathbf{P}_n & -\alpha\mathbb{1}_{n\times k}\\ \hline -\alpha\mathbb{1}_{k\times n} & \mathbf{P}_k\end{array}\right) \in \mathbb{R}^{(n+k)\times(n+k)}$$

where $\alpha = \frac{1}{k+n}$ and $\mathbf{P}_n = \mathbf{P}_n(\alpha)$ and $\mathbf{P}_k = \mathbf{P}_k(\alpha)$. Then

$$\begin{aligned}\bar{\mathbf{D}}\bar{\mathbf{D}}^{\mathrm{T}} =& \bar{\mathbf{V}}\bar{\mathbf{P}}\bar{\mathbf{V}}^{\mathrm{T}} = \mathbf{V}\mathbf{P}_n\mathbf{V}^{\mathrm{T}} - \alpha\mathbf{W}\mathbb{1}_{k\times n}\mathbf{V}^{\mathrm{T}} - \alpha\mathbf{V}\mathbb{1}_{n\times k}\mathbf{W}^{\mathrm{T}} + \mathbf{W}\mathbf{P}_k\mathbf{W}^{\mathrm{T}}\\ =& \mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}} + \alpha kn\mathbf{A} \quad \in \mathbb{R}^{m\times m}\end{aligned}$$

with $\mathbf{A} = \mathbf{a}\mathbf{a}^{\mathrm{T}} - \mathbf{a}\mathbf{w}^{\mathrm{T}} - \mathbf{w}\mathbf{a}^{\mathrm{T}} + \mathbf{w}\mathbf{w}^{\mathrm{T}}$ and $\mathbf{P} = \mathbf{P}_n(\frac{1}{n})$. To confirm this equivalence we study each term of the quadratic form:

From $\mathbf{P}_n = \mathbf{P} + \frac{k}{k+n}\mathbf{q}\mathbf{q}^{\mathrm{T}} = \mathbf{P} + \frac{k}{n(k+n)}\mathbb{1}$ (rank update) we obtain

$$\begin{aligned}\mathbf{V}\mathbf{P}_n\mathbf{V}^{\mathrm{T}} =& \mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}} + \frac{k}{n(k+n)}\mathbf{V}\mathbb{1}\mathbf{V}^{\mathrm{T}} = \mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}} + \frac{kn}{k+n}\mathbf{a}\mathbf{a}^{\mathrm{T}}\\ =& \mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}} + \alpha kn\,\mathbf{a}\mathbf{a}^{\mathrm{T}}\,.\end{aligned}$$

We have

$$-\alpha\mathbf{W}\mathbb{1}\mathbf{V}^{\mathrm{T}} = -\alpha k\mathbf{w}\mathbf{1}_n^{\mathrm{T}}\mathbf{V}^{\mathrm{T}} = -\alpha k\mathbf{w}(\mathbf{V1}_n)^{\mathrm{T}} = -\alpha kn\,\mathbf{w}\mathbf{a}^{\mathrm{T}}$$

and $-\alpha\mathbf{V}\mathbb{1}\mathbf{W}^{\mathrm{T}} = -\alpha kn\,\mathbf{a}\mathbf{w}^{\mathrm{T}}$.

And finally, $\mathbf{W}\mathbf{P}_k\mathbf{W}^{\mathrm{T}} = \mathbf{w}\mathbf{1}^{\mathrm{T}}\mathbf{P}_k\mathbf{1}\mathbf{w} = \frac{kn}{k+n}\mathbf{w}\mathbf{w}^{\mathrm{T}} = \alpha kn\,\mathbf{w}\mathbf{w}^{\mathrm{T}}$ using $\mathbf{1}^{\mathrm{T}}\mathbf{P}_k\mathbf{1} = (1-\alpha k)k = \alpha kn$.

With the mean $\bar{\mathbf{a}} = \alpha(n\,\mathbf{a} + k\mathbf{w})$ we write the condition

$$\bar{\mathbf{D}}\bar{\mathbf{D}}^{\mathrm{T}}\bar{\mathbf{a}} = \lambda\bar{\mathbf{a}} \quad\Leftrightarrow\quad (\mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}} + \alpha kn\mathbf{A})\,\mathbf{a} = \lambda\,\mathbf{a}\,,$$

which gives

$$\begin{aligned}&\frac{n}{k+n}\mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}}\mathbf{a} + \frac{k}{k+n}\mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}}\mathbf{w}\\ &\quad+ \frac{kn^2}{(k+n)^2}\mathbf{A}\mathbf{a} + \frac{k^2n}{(k+n)^2}\mathbf{A}\mathbf{w} - \frac{\lambda n}{k+n}\mathbf{a} - \frac{\lambda k}{k+n}\mathbf{w} = \mathbf{0}.\end{aligned}$$

In the limit $k\to\infty$ this reduces to the condition

$$\mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}}\mathbf{w} + n\left(\mathbf{a}\mathbf{a}^{\mathrm{T}} - \mathbf{a}\mathbf{w}^{\mathrm{T}} - \mathbf{w}\mathbf{a}^{\mathrm{T}} + \mathbf{w}\mathbf{w}^{\mathrm{T}}\right)\mathbf{w} - \lambda\mathbf{w} = \mathbf{0}\,,$$

and with $\mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}} = \mathbf{D}\mathbf{D}^{\mathrm{T}} = \mathbf{V}\mathbf{V}^{\mathrm{T}} - n\,\mathbf{a}\mathbf{a}^{\mathrm{T}}$ we obtain

$$\begin{aligned}&\mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{w} + n\left(\mathbf{w}\mathbf{w}^{\mathrm{T}} - \mathbf{a}\mathbf{w}^{\mathrm{T}} - \mathbf{w}\mathbf{a}^{\mathrm{T}}\right)\mathbf{w} - \lambda\mathbf{w} = \mathbf{0} \quad\Leftrightarrow\\ &\mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{w} - n||\mathbf{w}||^2\mathbf{a} = \mu\mathbf{w} \quad \text{for} \quad \mu = \lambda + n(\mathbf{w}^{\mathrm{T}}\mathbf{a} - ||\mathbf{w}||^2)\,,\end{aligned}$$

which is the postulated *parallel vectors condition*. □

Acknowledgments

This work was supported by DFG grant TH 692/13-1.

References

[BRB*15] BERG P., ROLOFF C., BEUING O., VOSS S., SUGIYAMA S.-I., ARISTOKLEOUS N., ANAYIOTOS A. S., ASHTON N., REVELL A., BRESSLOFF N. W., ET AL.: The computational fluid dynamics rupture challenge 2013–phase II: variability of hemodynamic simulations in two intracranial aneurysms. *Journal of biomechanical engineering 137*, 12 (2015), 121008. 5

[BS95] BANKS D. C., SINGER B. A.: A predictor-corrector technique for visualizing unsteady flow. *IEEE Transactions on Visualization and Computer Graphics 1*, 2 (1995), 151–163. 2

[CHL13] COX J., HOUSE D., LINDELL M.: Visualizing uncertainty in predicted hurricane tracks. *International Journal for Uncertainty Quantification 3*, 2 (2013), 143–156. 7

[FBW16] FERSTL F., BÜRGER K., WESTERMANN R.: Streamline variability plots for characterizing the uncertainty in vector field ensembles. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2016), 767–776. 2, 7

[FKRW16] FERSTL F., KANZLER M., RAUTENHAUS M., WESTERMANN R.: Visual analysis of spatial variability and global correlations in ensembles of iso-contours. In *Computer Graphics Forum* (2016), vol. 35, pp. 221–230. 7

[GYHZ13] GUO H., YUAN X., HUANG J., ZHU X.: Coupled ensemble flow line advection and analysis. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2733–2742. 2

[Har83] HARALICK R. M.: Ridges and valleys on digital images. *Computer Vision, Graphics, and Image Processing 22*, 1 (1983), 28–38. 2

[JBS*15] JANIGA G., BERG P., SUGIYAMA S., KONO K., STEINMAN D.: The computational fluid dynamics rupture challenge 2013—phase i: Prediction of rupture status in intracranial aneurysms. *American Journal of Neuroradiology 36*, 3 (2015), 530–536. 5

[JDKW15] JAREMA M., DEMIR I., KEHRER J., WESTERMANN R.: Comparative visual analysis of vector field ensembles. In *IEEE Visual Analytics Science and Technology (VAST)* (2015), pp. 81–88. 2

[JH95] JEONG J., HUSSAIN F.: On the identification of a vortex. *Journal of fluid mechanics 285* (1995), 69–94. 2

[Ken98] KENWRIGHT D. N.: Automatic detection of open and closed separation and attachment lines. In *IEEE Visualization* (1998), pp. 151–158. 2

[KKKK12] KUTZ B. M., KOWARSCH U., KESSLER M., KRÄMER E.: Numerical investigation of helicopter rotors in ground effect. In *30th AIAA Applied Aerodynamics Conference* (2012), vol. 7. 6

[LGY15] LIU R., GUO H., YUAN X.: A bottom-up scheme for user-defined feature comparison in ensemble data. In *SIGGRAPH Asia: Visualization in High Performance Computing* (2015), p. 10. 2

[LGY17] LIU R., GUO H., YUAN X.: User-defined feature comparison for vector field ensembles. *Journal of Visualization 20*, 2 (2017), 217–229. 2

[LGZY16] LIU R., GUO H., ZHANG J., YUAN X.: Comparative visualization of vector field ensembles based on longest common subsequence. In *IEEE Pacific Visualization Symposium (PacificVis)* (2016), pp. 96–103. 2

[MWK14] MIRZARGAR M., WHITAKER R. T., KIRBY R. M.: Curve boxplot: Generalization of boxplot for ensembles of curves. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 2654–2663. 2, 7

[OCJP16] OELTZE-JAFRA S., CEBRAL J. R., JANIGA G., PREIM B.: Cluster Analysis of Vortical Flow in Simulations of Cerebral Aneurysm Hemodynamics. *IEEE Transactions on Visualization and Computer Graphics 22 (1)*, 1 (2016), 757–766. 7

[OLK*14] OELTZE S., LEHMANN D. J., KUHN A., JANIGA G., THEISEL H., PREIM B.: Blood Flow Clustering and Applications in Virtual Stenting of Intracranial Aneurysms. *IEEE Transactions on Visualization and Computer Graphics 20(5)* (2014), 686–701. 7

[OP14] OBERMAIER H., PEIKERT R.: Feature-based visualization of multifields. In *Scientific Visualization*. Springer, 2014, pp. 189–196. 2

[POS*11] PAGOT C., OSMARI D., SADLO F., WEISKOPF D., ERTL T., COMBA J.: Efficient parallel vectors feature extraction from higher-order data. In *Computer Graphics Forum* (2011), vol. 30, pp. 751–760. 2

[PR99] PEIKERT R., ROTH M.: The "Parallel Vectors" Operator - A Vector Field Visualization Primitive. *IEEE Visualization* (1999), 263–270,532. 1, 2, 3

[PVH*03] POST F. H., VROLIJK B., HAUSER H., LARAMEE R. S., DOLEISCH H.: The state of the art in flow visualisation: Feature extraction and tracking. In *Computer Graphics Forum* (2003), vol. 22, Wiley Online Library, pp. 775–792. 1

[RP98] ROTH M., PEIKERT R.: A higher-order method for finding vortex core lines. In *IEEE Visualization* (1998), pp. 143–150. 2

[RT12] RÖSSL C., THEISEL H.: Streamline embedding for 3d vector field exploration. *IEEE Transactions on Visualiztion and Computer Graphics 18-3* (2012), 407–420. 7

[SH95] SUJUDI D., HAIMES R.: Identification of swirling flow in 3-d vector fields. In *12th Computational Fluid Dynamics Conference* (1995), pp. 792–799. 2

[SRE05] STEGMAIER S., RIST U., ERTL T.: Opening the can of worms: An exploration tool for vortical flows. In *IEEE Visualization* (2005), pp. 463–470. 2

[SZP06] SUKHAREV J., ZHENG X., PANG A.: Tracing parallel vectors. In *Electronic Imaging* (2006), International Society for Optics and Photonics, pp. 606011–606011. 2

[TS03] THEISEL H., SEIDEL H.-P.: Feature flow fields. In *VisSym* (2003), vol. 3, pp. 141–148. 2

[TSW*05] THEISEL H., SAHNER J., WEINKAUF T., HEGE H.-C., SEIDEL H.-P.: Extraction of parallel vector surfaces in 3d time-dependent fields and application to vortex core line tracking. In *IEEE Visualization* (2005), IEEE, pp. 631–638. 2

[USM96] UENG S.-K., SIKORSKI C., MA K.-L.: Efficient streamline, streamribbon, and streamtube constructions on unstructured grids. *IEEE Transactions on Visualization and Computer Graphics 2*, 2 (1996), 100–110. 2

[VP04] VERMA V., PANG A.: Comparative flow visualization. *IEEE Transactions on Visualization and Computer Graphics 10*, 6 (2004), 609–624. 2

[VP09] VAN GELDER A., PANG A.: Using PVsolve to analyze and locate positions of parallel vectors. *IEEE Transactions on Visualization and Computer Graphics 15*, 4 (2009), 682–695. 2

[WMK13] WHITAKER R. T., MIRZARGAR M., KIRBY R. M.: Contour boxplots: A method for characterizing uncertainty in feature sets from simulation ensembles. *IEEE Trans. Vis. Comput. Graph. 19*, 12 (2013), 2713–2722. 7

[WTVP11] WEINKAUF T., THEISEL H., VAN GELDER A., PANG A.: Stable feature flow fields. *IEEE Transactions on Visualization and Computer Graphics 17*, 6 (2011), 770–780. 2