J Vis (2018) 21:147–161
https://doi.org/10.1007/s12650-017-0438-2

REGULAR PAPER

Wentao Wang · Wenke Wang · Sikun Li

Detection and classification of critical points in piecewise linear vector fields

Received: 19 October 2016 / Accepted: 14 April 2017 / Published online: 29 June 2017

Abstract Topological methods are important for flow visualization, and many of them involve the detection and classification of critical points. Traditionally, the critical points are detected and classified by error-prone numerical methods, until Bhatia et al. (Topological methods in data analysis and visualization III, Springer, 2014) proposed the robust method for detecting simplices containing critical points. In this paper, we will extend Bhatia's idea to compute the Poincaré index of critical points in piecewise linear vector fields. All kinds of simplical complexes are considered, including 2D/3D triangulated meshes, and also triangulated surfaces. We test our algorithm on both synthetic and simulation data sets, which show the efficiency and accuracy of our methods.

Keywords Flow visualization · Critical points · Poincaré index · Brouwer degree · Simulation of simplicity

1 Introduction

Topological visualization is an important topic for flow visualization. Since the first introduction of critical point analysis to visualization community by Helman and Hesselink (1989), there have been a lot of topological methods developed (Pobitzer et al. 2010; Laramee et al. 2007; Wang et al. 2016), most of them related to critical points.

There are mainly two aspects of the critical points for the flow visualization, i.e., the detection and the classification. The detection of critical points is trivial in the mathematical sense; however, there are numerical problems related to the interpolated vector fields. By the approximation nature of numerical techniques, we are only interested in stable critical points, so a meaningful critical point is a zero point with meaningful neighboring structures. The most recent method of robust detection of critical points was proposed by Bhatia et al. (2014).

Another aspect of critical points is the classification. The most widely adopted classification measure is the Poincaré index. We can always compute the index by the eigenvalues of Jacobian matrix; however, we will be bothered by numerical problems again.

W. Wang · S. Li
College of Computer, National University of Defense Technology, Changsha 410073, China
E-mail: wwthunan@gmail.com

W. Wang · S. Li
State Key Laboratory of High Performance Computing, National University of Defense Technology, Changsha 410073, China

W. Wang (✉)
Institute of Ocean Science and Engineering, National University of Defense Technology, Changsha 410073, China
E-mail: wangwenke@nudt.edu.cn

In this paper, we proposed a method to robustly and efficiently compute the Poincaré index for critical points in piecewise linear vector fields, which is compatible with the method proposed by Bhatia et al. (2014). In addition, we also discuss some related topics.

Our main contributions can be summarized as follows:

- We clarify the related topological concepts for piecewise linear vector fields.
- We improve Bhatia's detection method for critical point.
- We derive the formula for the computation of Poincaré index for piecewise linear vector fields.
- We discuss the case of triangulated surfaces.

The remaining text is organized as follows. In Sect. 2, some related works and theoretical background are introduced. In addition, then, we give the main results of this paper in Sect. 3, including some propositions and proofs, which lead to the algorithms in Sect. 4. In Sect. 5, we show the experimental results, and finally conclude in Sect. 6.

2 Background

2.1 Related works

The critical points, first introduced to the visualization community by Helman and Hesselink (1989), play a central role in the topological visualization of vector fields. In Helman's method (Helman and Hesselink 1989), a stable (or hyperbolic) critical point is detected as the location with zero vector value, and the classification is determined by the eigenvalues of its Jacobian matrices.

The detection of isolated critical points is trivial by numerical methods. For example, there are methods (Lavin et al. 1998; Batra and Hesselink 1999) to compute the location of zero vectors by solving a linear system. For non-isolated critical points, Mann and Rockwood (2002) used the geometric algebra to detect non-isolating critical points based on Poincaré–Hopf theorem. Because the neighboring structures are more important than critical points themselves, there are some other indirect methods, for example, Morse decomposition (Chen et al. 2007), Hodge decomposition (Wang and Deng 2014), combinatorial vector field topology (Reininghaus and Hotz 2011), etc. More detailed survey can be found in Wang et al. (2016). The problem of numerical methods is the instability when handling degenerate cases. Recently, Bhatia et al. (2014) proposed robust and efficient methods to detect the simplices containing critical points based on Brouwer degree theory. Our results are mainly motivated by this paper.

The full classification of critical points can be done by eigenvalues of Jacobian matrices. However, computation of eigenvalues is neither efficient nor robust. A more widely used classification measure is the Poincaré index. Garth et al. (2004) proposed a method to get the 3D Poincaré index by computing the areas of the spherical triangles, which involves complicated numerical computation. As far as we know, there is still no robust and efficient method to compute the index.

In this paper, motivated by the detailed formulation of linear vector field on simplex in Wang et al. (2016), we extend the method of Bhatia et al. (2014) to compute the Poincaré index of the critical points in piecewise linear vector fields.

2.2 Vector fields

A vector field $\boldsymbol{v} : \Omega \to \mathbb{R}^n$ on $\Omega \subset \mathbb{R}^n$ can be viewed as a special case of an initial value problem:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{v}(\boldsymbol{x}, t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0,$$

where $\boldsymbol{v}(\boldsymbol{x}, t) = \boldsymbol{v}(\boldsymbol{x})$, i.e., not directly related to time t. For a vector field $\boldsymbol{v}(\boldsymbol{x})$, a *critical point* $\boldsymbol{x}_0$, i.e., a point $\boldsymbol{x}_0$ with $\boldsymbol{v}(\boldsymbol{x}_0) = \boldsymbol{0}$, is characterized by the eigenvalues of the Jacobian matrix:

$$J(\boldsymbol{v}, \boldsymbol{x}_0) = \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{x}} \boldsymbol{v} \bigg|_{\boldsymbol{x}_0}.$$

A special case is the *hyperbolic* critical points, i.e., the critical points with eigenvalues having non-zero real parts. Hyperbolic critical points are stable, which are important in both theory and practice. By Hartman–Grobman theorem (Hsu 2006), the neighboring structure around a hyperbolic critical point $\boldsymbol{x}_0$ is topologically equivalent to its linearized system:

$$\boldsymbol{v}(\boldsymbol{x}) = J(\boldsymbol{v}, \boldsymbol{x}_0)\boldsymbol{x}.$$

By the eigenvalues of Jacobian matrix, we can classify the hyperbolic critical points. Intuitively, a real eigenvalue λ corresponds to repelling ($\lambda > 0$) or attracting ($\lambda < 0$) behaviour along the direction of its eigenvector. A pair of complex eigenvalues $\alpha \pm \beta \boldsymbol{i}$ corresponds to the swirling behaviour in the plane defined by its complex eigenvectors, and the real part α still contains repelling/attracting information. The critical points with pure imaginary eigenvalues are called *centers*. In practice, it is difficult to exactly extract centers because of interpolation scheme and numerical errors. However, centers carry the important swirling motions, and we can always treat them as degenerated cases of hyperbolic critical points.

2.3 Simplices and simplicial complexes

Data sets of vector fields are defined mainly on regular or irregular grids. The most important irregular grid are the simplicial complexes or triangulated meshes.

Simplicial complex consists of simplices. An n-dimensional *simplex* (or n-simplex) is the convex hull of $n+1$ affinely independent points $\{\boldsymbol{a}_0, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n\} \subset \mathbb{R}^n$. Affine independence means that these $n+1$ points are not in the same $(n-1)$-dimensional hyper-plane or the volume of the convex hull is non-zero. When we shuffle the order of these points, we get two *orientations*, defined as the sign of the permutation. If the number of swaps needed to change one order to another is even, these two orders have the same orientation. The orientation can be computed as the sign of the signed volume of the simplex Bhatia et al. (2014):

$$\det(\boldsymbol{a}_0, \ldots, \boldsymbol{a}_n) \triangleq \begin{vmatrix} \boldsymbol{a}_{0,0} & \boldsymbol{a}_{1,0} & \cdots & \boldsymbol{a}_{n,0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{a}_{0,n-1} & \boldsymbol{a}_{1,n-1} & \cdots & \boldsymbol{a}_{n,n-1} \\ 1 & 1 & \cdots & 1 \end{vmatrix}. \tag{1}$$

Notice that, we may abuse the notation of det by considering it as a function det(.) as defined in Eq. (1) or an operator on a square matrix as shown in Eqs. (2) or (4).

Triangular meshes consisting of 2-simplices and tetrahedral meshes consisting of 3-simplices are of main interest in applications. The triangular meshes may be either planar meshes or surface meshes, and the latter one embedded in $\mathbb{R}^3$ is more difficult to handle with.

2.4 Linearly interpolated vector fields

In practice, vector field data sets are sampled fields with topology defined by geometrical grids. Therefore, if we need to get the vector values in arbitrary positions other than the vertices, we have to interpolate it according to neighboring vertices.

There are numerous interpolation schemes, while the simplest and most widely adopted one for simplicial complex in visualization community is the linear interpolation. If the corresponding vector values on an n-simplex $\{\boldsymbol{a}_0, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n\}$ are $\{\boldsymbol{v}_0, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$, then we can always formulate the linearly interpolated vector field in this simplex as

$$\boldsymbol{v}(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{b},$$

where A and $\boldsymbol{b}$ are constant, as shown in Proposition 5.

2.5 Brouwer degree and Poincaré index

The eigenvalues of Jacobian matrix are sufficient to classify critical points. However, they are computationally intensive and not topologically invariants. Poincaré index, defined by Brouwer degree, is more important in various applications, although it is more coarse than eigenvalues with regard to classification.

For a smooth map $f : \Omega \to \mathbb{R}^n$, where $\Omega \subset \mathbb{R}^n$ is a connected and closed bounded region, the $p \in \mathbb{R}^n$ is called a *regular value* if $f^{-1}(p)$ is a finite set. The *Brouwer degree* of f of Ω relative to a regular value p is defined as

$$\deg(f,\Omega,p) = \sum_{x\in f^{-1}(p)} \operatorname{Sign}\det Df_{\boldsymbol{x}}, \tag{2}$$

where $Df_{\boldsymbol{x}}$ is the derivative of f at $\boldsymbol{x}$, usually represented as a matrix under given frames.

Poincaré index is defined as the Brouwer degree of a special function. For the smooth map $f:\Omega\to\mathbb{R}^n$ if there is an isolated zero point $\boldsymbol{x}_0$ in the interior of Ω, i.e., $f(\boldsymbol{x}_0)=\boldsymbol{0}$, then by selecting a small enough sphere S centered at $\boldsymbol{x}_0$, we can define the *Poincaré index* at $\boldsymbol{x}_0$ as the degree of $u:S\to S^{n-1}$, where $u(x)=f(x)/|f(x)|$ and S^{n-1} is the unit sphere in $\mathbb{R}^n$. In Proposition 1, we will show that the Poincaré index of a critical point can be directly computed as the Brouwer degree of any neighborhood around this critical point relative to $\boldsymbol{0}$ in linear vector fields.

2.6 Simulation of simplicity

Simulation of Simplicity (SoS), proposed by Edelsbrunner and Mücke (1990), is a programming technique for the predicate problem in computational geometry. Its main idea is to perturb the data set to eliminate all degenerate cases and, thus, simplify the programming in computational geometry.

The predicate problem in computational geometry is to classify a quantity to one of $\{+1,0,-1\}$. For example, we usually want to determine the relative position between a plane and a point. Therefore, we can let $+1$ to represent the point on one side of the plane, -1 the other side, and 0 in the plane. In specific, many of these problems can be reduced to the computation of the sign of a determinant, as shown in Eq. (1). If the signed volume of a simplex $\{\boldsymbol{a}_0,\boldsymbol{a}_1,\ldots,\boldsymbol{a}_n\}$ is positive, then point $\boldsymbol{a}_0$ lies on one side of the hyper-plane determined by points $\{\boldsymbol{a}_1,\ldots,\boldsymbol{a}_n\}$, and on the other side if negative. When the volume is zero, the $n+1$ points are coplanar (in an $(n-1)$-dimensional hyper-plane), which is viewed as a degenerated case and can be eliminated by SoS.

To avoid the zero determinant of a matrix, the SoS technique perturbs all the n points symbollically. In specific, there is a designed perturb function $f(\varepsilon,i,j)=\varepsilon^{2^{i\delta-j}},\delta\geq n$ added to the jth component of the ith point to ensure that the resulting determinant, a polynomial of ε, approaches to original one as the symbolic variable $\varepsilon\to 0$. However, the determinant is never zero as long as $\varepsilon>0$. Therefore, the ε is just symbolic and the function is not actually computed. We only need to care about the coefficient of the polynomial expression of the determinant. Notice that, because this perturb function is only related to the indices i and j, so in fact, the SoS introduce a global perturbation and thus lead to a globally consistent result.

Bhatia et al. (2014) applied the SoS to the detection of critical points to solve the problem of inconsistency due to the numerical errors. As shown in Fig. 1, there is a critical point O on the edge of BC. If we apply traditionally numerical methods, O may belongs to both ABC and BCD, or none of them. Point O on the edge of BC is considered to be degenerated, so O must be on either side of the edge BC after applying SoS. By the global consistency, the relative location of O is uniquely determined.

2.7 Robust detection of critical points

The robust detection method of critical points is based on the following theorem proposed by Bhatia et al. (2014).

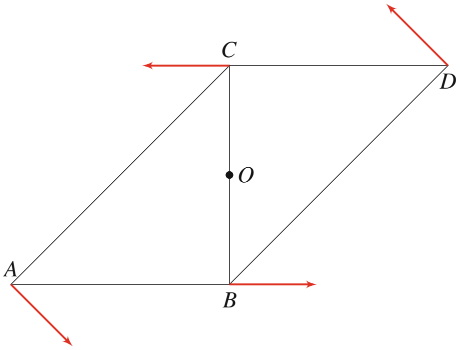


Fig. 1 Degenerate case for the detection of critical points

Theorem 1 *Bhatia et al.* (2014) Let $\mathbb{S} = \{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_n\} \subset \mathbb{R}^n$ *be a simplex, and* $\boldsymbol{v}$ *be an interpolated vector field defined by vectors* $\boldsymbol{v}_0, \ldots, \boldsymbol{v}_n$ *on the vertices of* $\mathbb{S}$. *Then,* $\mathbb{S}$ *contains a critical point if and only if* $\boldsymbol{0}$ *is in the interior of the convex hull of* $\{\boldsymbol{v}_0, \ldots, \boldsymbol{v}_n\}$.

Then, Bhatia et al. reduce the detection problem to determine if the signs of $n+2$ determinants are the same, that is

$$\mathrm{Sign}\det(\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_n) = \mathrm{Sign}\det(\boldsymbol{v}_0, \ldots, \boldsymbol{0}_k, \ldots, \boldsymbol{v}_n),$$

where $\boldsymbol{0}_k, k = 0, 1, \ldots, n$, means $\boldsymbol{v}_k$ is replaced by $\boldsymbol{0}$.

3 Main results

3.1 Brouwer degree versus Poincaré index

In this section, we further clarify the relation between Brouwer degree and Poincaré index. Under the assumption of linear interpolation, their computations are the same, regardless of the choice of neighborhood.

Proposition 1 *If a linear vector field* $\boldsymbol{v}(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{b}$ *has a hyperbolic critical point* $\boldsymbol{x}_0$ *with a bounded and closed neighborhood N, then the Poincaré index of* $\boldsymbol{x}_0$ *is the same with the Brouwer degree of N if the frame field of the sphere is compatible to the surrounding Euclidean space.*

Proof Because the hyperbolic critical point $\boldsymbol{x}_0 = A^{-1}\boldsymbol{b}$ is the only critical point of this vector field, and due to the hyperbolic nature, we know the coefficient matrix A is invertible. Therefore, for any regular value $\boldsymbol{v}_1$, there is only one point in its pre-image $A^{-1}(\boldsymbol{v}_1 - \boldsymbol{b})$. According to the definition as shown in Eq. (2), the Brouwer degree of N about the critical point is just $\mathrm{Sign}\det A$.

On the other hand, consider normalized map $f(\boldsymbol{x}) = (A\boldsymbol{x} + \boldsymbol{b})/|A\boldsymbol{x} + \boldsymbol{b}|$. Let $g : \mathbb{R}^n - \{0\} \to S^{n-1}$ be the extension of $f : S \to S^{n-1}$, where S is a small sphere centered at $\boldsymbol{x}_0$, then $f = g|_S$. It is easy to show that the derivative of g is

$$\mathrm{d}g = \frac{1}{|f|}(I - \boldsymbol{arf}\,\boldsymbol{arf}^T)A$$

under usual Euclidean frame, where $\boldsymbol{arf} = (A\boldsymbol{x} + \boldsymbol{b})/|A\boldsymbol{x} + \boldsymbol{b}|$. Therefore

$$\mathrm{d}f = \mathrm{d}g|_{T_xS},$$

a linear transformation restricted on the tangent plane $T_{\boldsymbol{x}}S$, which is actually the restriction of $\boldsymbol{x} \mapsto A\boldsymbol{x}$ on the tangent plane $T_{\boldsymbol{x}}S$. By taking frames compatible with $\mathbb{R}^n$ for both $T_{\boldsymbol{x}}S$ and $T_{f(\boldsymbol{x})}S^{n-1}$, we know that $\det df = |A|$.

We still have to prove there is only one point in the pre-image of each regular value of f. If there are $\boldsymbol{x}_1, \boldsymbol{x}_2 \in S$, such that $f(\boldsymbol{x}_1) = f(\boldsymbol{x}_2)$, then there is a $k > 0$, such that $A\boldsymbol{x}_1 + \boldsymbol{b} = k(A\boldsymbol{x}_2 + \boldsymbol{b})$. Thus, $\boldsymbol{x}_1 - k\boldsymbol{x}_2 = (1-k)A^{-1}\boldsymbol{b} = (1-k)\boldsymbol{x}_0$, where $\boldsymbol{x}_0$ is the center of the sphere S. Therefore, we must have $k = 1$, and thus, $\boldsymbol{x}_1 = \boldsymbol{x}_2$. □

3.2 Poincaré Index versus eigenvalues

In this section, we discuss the relation between Poincaré index and eigenvalues. As shown in Proposition 4, the Poincaré index is actually the number of positive eigenvalues (or positive real parts of complex eigenvalues). To prove Proposition 4, we need some results (Proposition 2 and 3) from linear algebra.

Proposition 2 *For an n-dimensional real square matrix A, the product of its eigenvalues is* $|A|$.

Proof Let $f(\lambda) = |A - \lambda I| = 0$, so f is a polynomial with order n, and coefficient of λ^n is $(-1)^n$, and the constant item of $f(\lambda)$ is $f(0) = |A|$. According to Fundamental Theorem of Algebra (Waerden and Leendert), there must be n (complex) roots of $f(\lambda) = 0$ (counted with multiplicity), denoted as $\lambda_1, \lambda_2, \ldots, \lambda_n$, and they are the eigenvalues of A. So $(-1)^n \prod_i^n(-\lambda_i) = f(0)$, and thus

$$\prod_{i}^{n} \lambda_i = f(0) = |A|.$$

□

Proposition 3 *For an invertible $n \times n$ matrix A, if the number of eigenvalues with positive real parts is k, then*

$$\text{Sign}|A| = (-1)^{n-k}.$$

Proof According to Sect. 5.5 of Hirsch et al. (2013), we can always find an invertible matrix T, such that

$$T^{-1}AT = \begin{bmatrix} R_1 & & & & & \\ & \ddots & & & & \\ & & R_r & & & \\ & & & C_1 & & \\ & & & & \ddots & \\ & & & & & C_\ell \end{bmatrix},$$

where R_i and C_j are matrix blocks (possibly with different sizes) with forms

$$R_i = \begin{bmatrix} \lambda_i & 1 & & \\ & \ddots & \ddots & \\ & & \lambda_i & 1 \\ & & & \lambda_i \end{bmatrix}, C_j = \begin{bmatrix} \Lambda_j & I_2 & & \\ & \ddots & \ddots & \\ & & \Lambda_j & I_2 \\ & & & \Lambda_j \end{bmatrix},$$

where

$$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \Lambda_j = \begin{bmatrix} \alpha_j & \beta_j \\ beta_j & \alpha_j \end{bmatrix}.$$

Then, A has eigenvalues $\lambda_i, \alpha_j \pm \beta_j \boldsymbol{i}$, $i = 1, \ldots, r, j = 1, \ldots, \ell$ counted with multiplicity. We also know

$$n = \sum_{i=1}^{r} \text{dim} R_i + \sum_{j=1}^{\ell} \text{dim} C_j,$$

and the number of distinct real eigenvalues is r, and the number of distinct complex eigenvalues is 2ℓ.

By Proposition 2, |*A*| is the product of all the eigenvalues counted with multiplicity. Then, all the contributions to the sign of |*A*| come from the $n - k$ negative eigenvalues (or complex eigenvalue with negative parts) counted with multiplicity. Obviously, we have

$$\text{Sign}|A| = (-1)^{n-k}.$$

□

Proposition 4 *For a hyperbolic critical point $\boldsymbol{x}_0$ in an n-dimensional vector field, if the dimension of its unstable manifold is k, then its Poincaré index is $(-1)^{n-k}$.*

Proof For an n-dimensional vector field $f : \mathbb{R}^n \to \mathbb{R}^n$ with hyperbolic critical point $\boldsymbol{x}_0$, according to Hartman–Grobman theorem (Hsu 2006), we only need to consider its linearized system:

$$\dot{\boldsymbol{x}} = (Df)\boldsymbol{x},$$

where Df is the Jacobian matrix of f. By assumption of the k-dimensional unstable manifold, we know that there are k eigenvalues of Df with positive real parts.

By the definition of Poincaré index, by choosing any non-zero vector value $\boldsymbol{a} \neq \boldsymbol{0}$, we always get the only pre-image $\boldsymbol{x}_0 = (Df)^{-1}\boldsymbol{a}$, and the Jacobian matrix at $\boldsymbol{x}_0$ is always the constant Df. Therefore, the Poincaré index of this critical point is

$$\operatorname{Sign}\det Df = (-1)^{n-k},$$

according to Proposition 3. □

3.3 Poincaré index for critical points in linear vector fields

In this section, we give the method to compute the Poicnaré index in an easy way, as shown in Theorem 2. As a preliminary, we need to get the explicit formula (Proposition 5) for linear vector field on a simplex.

Proposition 5 *For a linear interpolated vector fields on an n-simplex, with vertices* $\boldsymbol{a}_0, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ *and corresponding vectors* $\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$*, the analytic form of this vector field is*

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + \boldsymbol{b},$$

where

$$\begin{aligned} A &= V_n A_n^{-1}, \\ \boldsymbol{b} &= \boldsymbol{v}_n - A\boldsymbol{a}_n, \\ V_n &= [\boldsymbol{v}_0 - \boldsymbol{v}_n, \boldsymbol{v}_1 - \boldsymbol{v}_n, \ldots, \boldsymbol{v}_{n-1} - \boldsymbol{v}_n], \\ A_n &= [\boldsymbol{a}_0 - \boldsymbol{a}_n, \boldsymbol{a}_1 - \boldsymbol{a}_n, \ldots, \boldsymbol{a}_{n-1} - \boldsymbol{a}_n]^{-1}. \end{aligned}$$

Proof Define the velocity $\boldsymbol{v}(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}^n$ as $\boldsymbol{v}(\boldsymbol{x}) = \dot{\boldsymbol{x}}$. By linear interpolation, we mean each component of $\boldsymbol{v}$ is

$$\boldsymbol{v}^{(i)} = \sum_{j=0}^{n} c_j \boldsymbol{v}_j^{(i)} = [\boldsymbol{v}_0^{(i)}, \boldsymbol{v}_1^{(i)}, \ldots, \boldsymbol{v}_n^{(i)}][c_0, c_1, \ldots, c_n]^T,$$

where $c_j : \mathbb{R}^n \to \mathbb{R}$ is a linear function about $\boldsymbol{x}$, that is

$$c_j = \boldsymbol{y}_j^T \begin{pmatrix} \boldsymbol{x} \\ 1 \end{pmatrix}$$

for some $\boldsymbol{y}_j \in \mathbb{R}^{n+1}$. Let $V = [\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_n]$, and $Y = [\boldsymbol{y}_0^T, \boldsymbol{y}_1^T, \ldots, \boldsymbol{y}_n^T]$ then

$$\boldsymbol{v}(\boldsymbol{x}) = V[c_0, c_1, \ldots, c_n]^T = VY\begin{pmatrix} \boldsymbol{x} \\ 1 \end{pmatrix}.$$

Because $\boldsymbol{v}(\boldsymbol{a}_i) = \boldsymbol{v}_i$ at each vertex $\boldsymbol{v}_i$, if we denote

$$\mathcal{A} = \begin{bmatrix} \boldsymbol{a}_0 & \boldsymbol{a}_1 & \ldots & \boldsymbol{a}_n \\ 1 & 1 & \ldots & 1 \end{bmatrix},$$

then we have

$$V = VY\mathcal{A}.$$

Because c_j is not related to $\boldsymbol{v}_0, \ldots, \boldsymbol{v}_n$, so we must have $Y = \mathcal{A}^{-1}$. Thus

$$\boldsymbol{v}(\boldsymbol{x}) = V\mathcal{A}^{-1}\begin{pmatrix} \boldsymbol{x} \\ 1 \end{pmatrix}.$$

We can compute $\mathcal{A}^{-1}$ explicitly as

$$\begin{aligned}
\mathcal{A}^{-1} &= \left(\begin{bmatrix} I & \boldsymbol{a}_n \\ \boldsymbol{0}^T & 1 \end{bmatrix} \begin{bmatrix} A_n & \boldsymbol{0} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \begin{bmatrix} I & \boldsymbol{0} \\ \boldsymbol{1}^T & 1 \end{bmatrix} \right)^{-1} \\
&= \begin{bmatrix} I & \boldsymbol{0} \\ -\boldsymbol{1}^T & 1 \end{bmatrix} \begin{bmatrix} A_n^{-1} & \boldsymbol{0} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \begin{bmatrix} I & -\boldsymbol{a}_n \\ \boldsymbol{0}^T & 1 \end{bmatrix} \\
&= \begin{bmatrix} A_n^{-1} & -A_n^{-1}\boldsymbol{a}_n \\ -\boldsymbol{1}^T A_n^{-1} & 1 + \boldsymbol{1}^T A_n^{-1}\boldsymbol{a}_n \end{bmatrix}.
\end{aligned}$$

By reformulating V as

$$V = [\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_n] = [V_n, \boldsymbol{v}_n] \begin{bmatrix} I & \boldsymbol{0} \\ \boldsymbol{1}^T & 1 \end{bmatrix},$$

we finally have

$$\begin{aligned}
\boldsymbol{v}(\boldsymbol{x}) &= V\mathcal{A}^{-1} \begin{pmatrix} \boldsymbol{x} \\ 1 \end{pmatrix} \\
&= [V_n, \boldsymbol{v}_n] \begin{bmatrix} I & \boldsymbol{0} \\ \boldsymbol{1}^T & 1 \end{bmatrix} \begin{bmatrix} A_n^{-1} & -A_n^{-1}\boldsymbol{a}_n \\ -\boldsymbol{1}^T A_n^{-1} & 1 + \boldsymbol{1}^T A_n^{-1}\boldsymbol{a}_n \end{bmatrix} \begin{pmatrix} \boldsymbol{x} \\ 1 \end{pmatrix} \\
&= [V_n, \boldsymbol{v}_n] \begin{bmatrix} A_n^{-1} & -A_n^{-1}\boldsymbol{a}_n \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{pmatrix} \boldsymbol{x} \\ 1 \end{pmatrix} \\
&= V_n A_n^{-1}\boldsymbol{x} + \boldsymbol{v}_n - V_n A_n^{-1}\boldsymbol{a}_n.
\end{aligned}$$

□

Now, we are ready to give the main result of this paper, which compute the Poincaré index in a robust and succinct way.

Theorem 2 *If the vector field defined in Proposition 5 has a hyperbolic critical point, then its Poincaré index is equal to the sign of*

$$|V_n|\,|A_n| = \det(\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_n)\det(\boldsymbol{a}_0, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_n). \tag{3}$$

Proof According to Proposition 5, the formula for this linearly interpolated vector field is

$$\boldsymbol{v}(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{b},$$

then the Jacobian matrix is $A = V_n A_n^{-1}$. Because there is a hyperbolic critical point, say $\boldsymbol{x}_0$, then all the eigenvalues of A have non-zero real parts, and thus, A must be invertible. Therefore, $\boldsymbol{v} : \mathbb{R}^n \to \mathbb{R}^n$ is a bijection map, and we have

$$\mathrm{Index}(\boldsymbol{x}_0) = \deg(\boldsymbol{x}_0) = \mathrm{Sign}\det A$$

according to the Propositions 4 and 5. Therefore

$$\begin{aligned}
\mathrm{Index}(\boldsymbol{x}_0) &= \mathrm{Sign}\det A = \mathrm{Sign}\det V_n \det A_n^{-1} \\
&= \mathrm{Sign}\det V_n \det A_n.
\end{aligned}$$

□

3.4 Revisit of the robust detection

We have reviewed the detection algorithm proposed by Bhatia et al. in Sect. 2.7, which involves the computation of $n+2$ determinants, but now, we claim that at most $n+1$ determinants need to be computed.

Proposition 6 *For* $\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_n \in \mathbb{R}^n$ *, we have*

$$\det(\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_n) = \sum_{k=0}^{n} \det(\boldsymbol{v}_0, \cdots, \boldsymbol{0}_k, \cdots, \boldsymbol{v}_n),$$

where $\boldsymbol{0}_k, k = 0, 1, \ldots, n$, *means* $\boldsymbol{v}_k$ *is replaced by* $\boldsymbol{0}$.

Proof Take $k = 0$ for example

$$\begin{aligned} \det(\boldsymbol{0}, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_n) &= \begin{vmatrix} \boldsymbol{0} & \boldsymbol{v}_1 \\ 1 & 1 \end{vmatrix} \\ &= \begin{vmatrix} \boldsymbol{0} & \boldsymbol{v}_1 \\ 1 & 0 \end{vmatrix} = \begin{vmatrix} \boldsymbol{v}_0 & \boldsymbol{v}_1 \\ 1 & 0 \end{vmatrix}. \end{aligned}$$

Thus, for $k = 0, 1, \ldots, n$, each of these $n+1$ determinants can be reformulated to a determinant with the same first-n-rows and the last zero row except the kth element. Therefore, according to the multiple linearity of determinant, we add all these $n+1$ determinants by the last row, and get the

$$\det(\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_n) = \begin{vmatrix} \boldsymbol{v}_0 & \boldsymbol{v}_1 \\ 1 & 1 \end{vmatrix}.$$

□

According to Proposition 6, the sign of $\det(\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_n)$ is totally determined by other $n+1$ determinants. Furthermore, since there is a $\boldsymbol{0} \in \mathbb{R}^n$ in each of these $(n+1) \times (n+1)$ determinants, so at most $n+1$ $n \times n$ determinants need to be computed, other than original $n+2$ $(n+1) \times (n+1)$ determinants.

We may also notice that the whole detection algorithm is determined by the faces of the simplex, and the computation can be surely reused by the two neighboring simplices. Therefore, if the topology information is available, we can always predicate the neighboring simplices by results of current simplex. For example, simplices around a critical one are probably non-critical. However, because the detection algorithm itself is efficient enough, it may be not necessary to complicate the programming.

3.5 About the triangulated surfaces

Now, we will extend our discussion to the triangulated surfaces. The following proposition is a direct generalization of Theorem 1.

Proposition 7 *For a 3D triangle* $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3\}$ *with corresponding vectors* $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$, *under linear interpolation, there is a critical point in the interior of the triangle if and only if the signs of the following 3 determinants are the same:*

$$\det[\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{n}], \quad \det[\boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{n}], \quad \det[\boldsymbol{v}_3, \boldsymbol{v}_1, \boldsymbol{n}], \tag{4}$$

where $\boldsymbol{n}$ *is the normal of the triangle, and*

$$\boldsymbol{n} = \pm(\boldsymbol{a}_2 - \boldsymbol{a}_1) \times (\boldsymbol{a}_3 - \boldsymbol{a}_1).$$

Proof According to Theorem 1, we know the sufficient and necessary condition for the existence of critical point in a tetrahedron. Thus, we can construct a vector field on a tetrahedron generated by extruding the 3D triangle.

For a small enough $\varepsilon > 0$, consider the tetrahedron $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3 - \varepsilon\boldsymbol{n}, \boldsymbol{a}_3 + \varepsilon\boldsymbol{n}\}$, with corresponding vectors $\{\boldsymbol{v}'_1, \boldsymbol{v}'_2, \boldsymbol{v}'_3 - \varepsilon\boldsymbol{n}, \boldsymbol{v}'_3 + \varepsilon\boldsymbol{n}\}$, where $\boldsymbol{v}'_i = (1 - \boldsymbol{arnarn}^T)\boldsymbol{v}_i, i = 1, 2, 3,$ and $\boldsymbol{arn} = \boldsymbol{n}/|\boldsymbol{n}|$. Obviously, $\boldsymbol{v}'_i$ is the projection of $\boldsymbol{v}_i$ to the triangle, and $\{\boldsymbol{v}'_i\}$ are coplanar and $\boldsymbol{v}'_i \perp \boldsymbol{n}$. Because the triangle is in the tetrahedron, and the convex hull of $\{\boldsymbol{v}'_i\}$ is in the convex hull of the vectors on the vertices of the tetrahedron, it is easy to verify that there is a critical point in the triangle if and only if there is a critical point in the tetrahedron when $\varepsilon \to 0$.

According to Proposition 6, we need to compute the signs of 4 determinants:

$$\begin{aligned}D_1 &= -\ \det[\boldsymbol{v}_2', \boldsymbol{v}_3' - \varepsilon\boldsymbol{n}, \boldsymbol{v}_3' + \varepsilon\boldsymbol{n}],\\ D_2 &= \det[\boldsymbol{v}_1', \boldsymbol{v}_3' - \varepsilon\boldsymbol{n}, \boldsymbol{v}_3' + \varepsilon\boldsymbol{n}],\\ D_3 &= -\ \det[\boldsymbol{v}_1', \boldsymbol{v}_2', \boldsymbol{v}_3' + \varepsilon\boldsymbol{n}],\\ D_4 &= \det[\boldsymbol{v}_1', \boldsymbol{v}_2', \boldsymbol{v}_3' - \varepsilon\boldsymbol{n}].\end{aligned}$$

By $\det[\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}] = (\boldsymbol{a} \times \boldsymbol{b}) \cdot \boldsymbol{c}$ and $\boldsymbol{v}_i' \perp \boldsymbol{n}$, we get the following results:

$$\begin{aligned}D_1 &= \ -2\varepsilon(\boldsymbol{v}_2' \times \boldsymbol{v}_3') \cdot \boldsymbol{n},\\ D_2 &= \ -2\varepsilon(\boldsymbol{v}_3' \times \boldsymbol{v}_1') \cdot \boldsymbol{n},\\ D_3 &= \ -\varepsilon(\boldsymbol{v}_1' \times \boldsymbol{v}_2') \cdot \boldsymbol{n},\\ D_4 &= \ -\varepsilon(\boldsymbol{v}_1' \times \boldsymbol{v}_2') \cdot \boldsymbol{n}.\end{aligned} \tag{5}$$

Because $\bar{n}\bar{n}^T\boldsymbol{a} \,//\, \bar{n}\bar{n}^T\boldsymbol{b} \,//\, \boldsymbol{n}$, obviously, we have

$$\left[(I - \bar{n}\bar{n}^T)\boldsymbol{a} \times (I - \bar{n}\bar{n}^T)\boldsymbol{b}\right] \cdot \boldsymbol{n} = (\boldsymbol{a} \times \boldsymbol{b}) \cdot \boldsymbol{n}. \tag{6}$$

Since $\varepsilon > 0$, the signs of these determinants are irrelevant to ε, but only depends on

$$\begin{aligned}\det[\boldsymbol{v}_1', \boldsymbol{v}_2', \boldsymbol{n}] &= (\boldsymbol{v}_1' \times \boldsymbol{v}_2') \cdot \boldsymbol{n} = (\boldsymbol{v}_1 \times \boldsymbol{v}_2) \cdot \boldsymbol{n},\\ \det[\boldsymbol{v}_2', \boldsymbol{v}_3', \boldsymbol{n}] &= (\boldsymbol{v}_2' \times \boldsymbol{v}_3') \cdot \boldsymbol{n} = (\boldsymbol{v}_2 \times \boldsymbol{v}_3) \cdot \boldsymbol{n},\\ \det[\boldsymbol{v}_3', \boldsymbol{v}_1', \boldsymbol{n}] &= (\boldsymbol{v}_3' \times \boldsymbol{v}_1') \cdot \boldsymbol{n} = (\boldsymbol{v}_3 \times \boldsymbol{v}_1) \cdot \boldsymbol{n},\end{aligned}$$

according to Eqs. (5) and (6), that is

$$\det[\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{n}], \quad \det[\boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{n}], \quad \det[\boldsymbol{v}_3, \boldsymbol{v}_1, \boldsymbol{n}].$$

□

This result can be easily extended to any higher dimension, and it is actually the general case of planar triangular mesh, where the xy-plane is embedded in $\mathbb{R}^3$ with constant normal $\boldsymbol{n} = (0, 0, 1)^T$.

Therefore, we generalize the Bhatia's result from planar meshes to triangulated surfaces. Our result can be used as a standard technique for the detection of critical simplices.

However, we should be aware of that SoS does not apply any more for the case of surfaces, because a vector on an edge will be projected to different vectors on the neighboring triangles. Therefore, other criteria are still needed to robustly avoid the ambiguous situation that critical points lies on edge. For example, we can set a threshold to test the determinants, and recheck the neighboring triangles to remove redundancies.

Next, we need to compute the Poincaré index of the critical points with following proposition.

Proposition 8 *For a 3D triangle $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3\}$ with corresponding vectors $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$, if there is a critical point in the interior of the triangle, then under linear interpolation, the Poincaré index of the critical point is one of*

$$\text{Sign}\det[\boldsymbol{n}, \boldsymbol{v}_2, \boldsymbol{v}_3], \quad \text{Sign}\det[\boldsymbol{v}_1, \boldsymbol{n}, \boldsymbol{v}_3], \quad \text{Sign}\det[\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{n}],$$

where $\boldsymbol{n}$ is the normal of the triangle, and

$$\boldsymbol{n} = (\boldsymbol{a}_2 - \boldsymbol{a}_1) \times (\boldsymbol{a}_3 - \boldsymbol{a}_1).$$

Proof By choosing two orthogonal vectors $\boldsymbol{e}_1, \boldsymbol{e}_2$ in the plane of the triangle, together with $\boldsymbol{n}$, we can construct a coordinate system. In this new coordinate system, according to Proposition 5, we can get the two matrices A_n and V_n:

$$\begin{aligned}A_n &= [\boldsymbol{e}_1 \quad \boldsymbol{e}_2]^T[\boldsymbol{a}_1 - \boldsymbol{a}_3 \quad \boldsymbol{a}_2 - \boldsymbol{a}_3],\\ V_n &= [\boldsymbol{e}_1 \quad \boldsymbol{e}_2]^T[\boldsymbol{v}_1' - \boldsymbol{v}_3' \quad \boldsymbol{v}_2' - \boldsymbol{v}_3'],\end{aligned}$$

where $\boldsymbol{arn} = \boldsymbol{n}/|\boldsymbol{n}|$, and $\boldsymbol{v}_i' = (I - \boldsymbol{arnarn}^T)\boldsymbol{v}_i$. So the Poincaré index of the critical point is

$$\text{Sign}|A_n|\text{Sign}|V_n|.$$

If we reformulate A_n and V_n as

$$A'_n = [\,\boldsymbol{e}_1 \quad \boldsymbol{e}_2 \quad \boldsymbol{n}\,]^T [\,\boldsymbol{a}_1 - \boldsymbol{a}_3 \quad \boldsymbol{a}_2 - \boldsymbol{a}_3 \quad \boldsymbol{n}\,],$$
$$V'_n = [\,\boldsymbol{e}_1 \quad \boldsymbol{e}_2 \quad \boldsymbol{n}\,]^T [\,\boldsymbol{v}'_1 - \boldsymbol{v}'_3 \quad \boldsymbol{v}'_2 - \boldsymbol{v}'_3 \quad \boldsymbol{n}\,],$$

we get

$$\begin{aligned}
&\mathrm{Sign}|A_n|\mathrm{Sign}|V_n| = \mathrm{Sign}|A'_n|\mathrm{Sign}|V'_n| \\
&= \mathrm{Sign}\det[\boldsymbol{a}_1 - \boldsymbol{a}_3, \boldsymbol{a}_2 - \boldsymbol{a}_3, \boldsymbol{n}]\mathrm{Sign}\det[\boldsymbol{v}'_1 - \boldsymbol{v}'_3, \boldsymbol{v}'_2 - \boldsymbol{v}'_3, \boldsymbol{n}] \\
&= \mathrm{Sign}\det[\boldsymbol{v}'_1 - \boldsymbol{v}'_3, \boldsymbol{v}'_2 - \boldsymbol{v}'_3, \boldsymbol{n}] \\
&= \mathrm{Sign}(((\boldsymbol{v}'_1 - \boldsymbol{v}'_3) \times (\boldsymbol{v}'_2 - \boldsymbol{v}'_3)) \cdot \boldsymbol{n}) \\
&= \mathrm{Sign}((\boldsymbol{v}'_1 \times \boldsymbol{v}'_2) \cdot \boldsymbol{n}) = \mathrm{Sign}((\boldsymbol{v}_1 \times \boldsymbol{v}_2) \cdot \boldsymbol{n}) \\
&= \mathrm{Sign}\det[\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{n}].
\end{aligned}$$

According to Proposition 7, we finish the proof. □

4 Algorithms

4.1 Critical points on simplicial complexes

In this section, we extent the algorithm of Bhatia et al. (2014) to detect the critical points and compute the Poincaré index of it.

Algorithm 1 Positive$_n(\boldsymbol{a}_{j_0}, \cdots, \boldsymbol{a}_{j_n})$ [1]

```
$s \leftarrow$ Sort$(j_0, \cdots, j_n)$: $s$ is the number of swaps need to sort
$d \leftarrow$ Sign det$(\boldsymbol{a}_{j_0}, \cdots, \boldsymbol{a}_{j_n})$
if $odd(s)$ then
    $d \leftarrow -d$
end if
return $d$
```

Algorithm 1 is the same with Bhatia's algorithm. Function Positive$_\mathrm{n}(.)$ is used to compute the sign of the signed volume of simplex $[\boldsymbol{a}_{j_0}, \boldsymbol{a}_{j_1}, \ldots, \boldsymbol{a}_{j_n}]$. Notice that the sort of indices is necessary, because the perturbation of SoS depends on the global indices of the points. In implementation, this dependency implies that we should insist on the same indices for each point throughout the program. For example, if we index each point with its C++ pointer address, then two different pointers storing the same point are considered as different points. This implication also influence the Algorithm 2, where the points $\boldsymbol{a}_i$ or vectors $\boldsymbol{v}_i$ with a fixed pointer address should be used throughout the program.

Algorithm 2 Poincaré Index of a critical point in a simplex

```
Input: $\{\boldsymbol{a}_0, \cdots, \boldsymbol{a}_n, \boldsymbol{v}_0, \cdots, \boldsymbol{v}_n\}, \boldsymbol{a}_i, \boldsymbol{v}_i \in \mathbb{R}^n$
  $s \leftarrow$ Positive$_n(\boldsymbol{0}, \cdots, \boldsymbol{v}_n)$
  for $i = 1$ to $n$ do
      $(\boldsymbol{v}'_0, \cdots, \boldsymbol{v}'_n) \leftarrow (\boldsymbol{v}_0, \cdots, \boldsymbol{v}_n)$
      $\boldsymbol{v}'_i \leftarrow \boldsymbol{0}$
      $s_i \leftarrow$ Positive$_n(\boldsymbol{v}'_0, \cdots, \boldsymbol{v}'_n)$
      if $s \neq s_i$ then
          return $0$
      end if
  end for
  if Positive$_n(\boldsymbol{a}_0, \cdots, \boldsymbol{a}_n) \neq s$ then
      return $-1$
  end if
  return $+1$
```

Algorithm 2 is used to detect the critical point and compute its Poincaré index. In Bhatia's algorithm, a critical point is in the interior of a simplex, if and only if the origin is in the interior of the simplex constructed by the vectors on the vertices. Therefore, we first compute the orientation of simplex

$[\boldsymbol{0}, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_n]$, and replace $\boldsymbol{v}_k$ to $\boldsymbol{0}$, $k = 1, \ldots, n$, to compute the remaining n sign of the signed volumes. If all the $n+1$ signs are the same, then there is a critical point in the simplex, otherwise return 0. Then, we compute the orientation of simplex $[\boldsymbol{a}_0, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_n]$, if it is equal to the orientation of the simplex $[\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_n]$, then the Poincaré index is $+1$, otherwise -1.

4.2 Critical points on surfaces in $\mathbb{R}^3$

For the triangulated surfaces in $\mathbb{R}^3$, we cannot apply SoS technique to robustly detect the critical triangles; however, we can still use the standard procedure to compute in a conservative way.

In specific, we can use a threshold ε to control the computation procedure, as shown in Algorithm 3. When the signed volume relative to normal $\boldsymbol{n}$, e.g., $s_1 = \det[\boldsymbol{n}, \boldsymbol{v}_2, \boldsymbol{v}_3]$, is small, i.e., $-\varepsilon < s_1 < \varepsilon$, the sign of s_1 can be viewed as either -1 or $+1$.

Clearly, this algorithm is not satisfying, because multiple simplices sharing a single critical point can be all detected as critical simplices. Therefore, we may need further clean-up in specific applications.

Algorithm 3 Poincaré Index of a critical point in a surface triangle

```
Input: $\varepsilon > 0, \quad \{\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\} \subset \mathbb{R}^3$
  $\boldsymbol{n} = (\boldsymbol{a}_2 - \boldsymbol{a}_1) \times (\boldsymbol{a}_3 - \boldsymbol{a}_1)$
  $s_1 = \det[\boldsymbol{n}, \boldsymbol{v}_2, \boldsymbol{v}_3]$
  $s_2 = \det[\boldsymbol{v}_1, \boldsymbol{n}, \boldsymbol{v}_3]$
  if ($s_1 > \varepsilon$ and $s_2 < -\varepsilon$) or ($s_1 < -\varepsilon$ and $s_2 > \varepsilon$) then
      return 0
  end if
  $s_3 = \det[\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{n}]$
  if ($s_1 \geq -\varepsilon$ and $s_2 \geq -\varepsilon$ and $s_3 \geq -\varepsilon$) then
      return 1
  end if
  if ($s_1 \leq \varepsilon$ and $s_2 \leq \varepsilon$ and $s_3 \leq \varepsilon$) then
      return $-1$
  end if
  return 0
```

5 Experiments

We apply our algorithm to both the synthetic and simulation data sets. The experiment environment is: Linux-64bit operating system on an i7-1.347GHz laptop with 8G memory, and the algorithm is implemented with C++. In the following experiments, we ensure the results to be in accordance with the results of the Bhatia's results.

5.1 2D Tayler–Green vortex flow

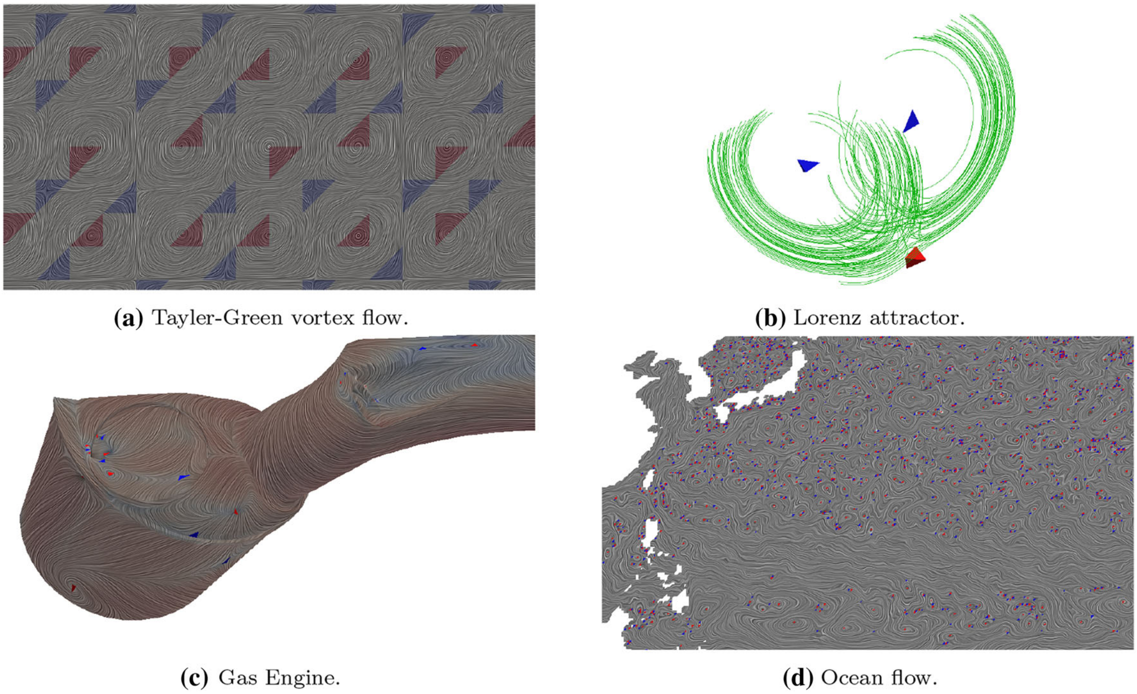
The Tayler–Green vortex flow (Stommel 1949; Maxey 1987) is a 2D vector field with equation:

$$\begin{aligned} u_x(x,y) = &\quad -U_{\max}\cos\left(\tfrac{\pi x}{L}\right)\sin\left(\tfrac{\pi y}{L}\right), \\ u_y(x,y) = &\quad U_{\max}\sin\left(\tfrac{\pi x}{L}\right)\cos\left(\tfrac{\pi y}{L}\right), \end{aligned}$$

where $U_{\max}$ is the maximum magnitude of the speed. If we choose $U_{max} = 10$ and $L = 1$, then the critical points are $C = \{(m_1, m_2) \mid m_1, m_2 \in \mathbb{Z}\}$ and $S = \{(m_1 + \frac{1}{2}, m_2 + \frac{1}{2}) \mid m_1, m_2 \in \mathbb{Z}\}$, where C is the set of centers (with purely imaginary eigenvalues) and S are the set of saddle points. Because of the interpolation scheme and numerical errors, the centers are always classified as hyperbolic critical points.

The result is shown in Fig. 2a, the red triangles contain critical point with Poincaré index $+1$, while the blue ones contain critical point with Poincaré index -1.

Because index of -1 represents the saddle point in 2D vector fields, we have actually proposed a robust and accurate method to extract 2D saddle points. Because saddle points are probably not swirling, we can use this robustly computed indices as a criteria for vortex core detection in 2D flow.


(a) Tayler-Green vortex flow.

(b) Lorenz attractor.

(c) Gas Engine.

(d) Ocean flow.

Fig. 2 Experimental results for the computation of Poincaré index. The *blue* and *red colors* represent indices -1 and $+1$, respectively.

5.2 Lorenz attractor

To verify the applicable to any higher dimensional spaces, we apply our algorithm to a 3D vector field, i.e., the Lorenz attractor with equation:

$$\begin{aligned}\dot{x} &= \sigma(y - x),\\ \dot{y} &= \rho x - y - xz,\\ \dot{z} &= -\beta z + xy,\end{aligned}$$

where $\sigma = 10, \beta = 8/3., \rho = 28$. There are 3 critical points including (0, 0, 0) and

$$(\pm\sqrt{\beta(\rho-1)}, \pm\sqrt{\beta(\rho-1)}, \rho - 1) = (\pm 6\sqrt{2}, \pm 6\sqrt{2}, 27).$$

It is easy to verify that the origin (0, 0, 0) is a critical point with Poincaré index $+1$ while other two have the indices of -1.

The result is shown in Fig. 2b, we get a red tetrahedron at origin and two blue ones in the center of two arms of the attractor.

5.3 Gas engine

To validate our results on the triangulated surfaces, we test Algorithms 3 on a gas engine data set from automotive engine simulation (Laramee et al. 2004). The result is shown in Fig. 2c.

This is a data set with 13,151 vertices and 26,298 triangles. In our experiments, it costs 0.00791 s to detect the critical simplices by traditional projection and interpolation. And our method needs only 0.0058 s to detect the critical simplices and compute the Poincaré index. Our algorithm is more efficient and the programming is much easier.

5.4 Ocean flow

We test our algorithm on an ocean flow data set, which are simulated from the HYbrid Coordinate Ocean Model (HYCMO) sponsored by the National Ocean Partnership Program (NOPP). The data files used in our experiments are rarchv.2008_028_00_3zu.nc and rarchv.2008_028_00_3zu.nc, which are velocity components on a 4500 × 2170 graticule grid.

If we only detect critical point by Bhatia's method, it needs 5.07 s. However, we can compute the Poincaré index in about 4.38 s by our method. If we detect critical points first, and then compute eigenvalues, it will cost 11.42 s.

5.5 Turbulence simulation flow

We also test our algorithm on the MHD1024 turbulence simulation dataset (Eyink et al. 2013) from Johns Hopkins Turbulence Databases. We use a tetrahedral grid with 82,906,875 cells and 16,777,216 points. There are 12,077 critical cells with index +1, and 12,072 critical cells with index −1.

The original Bhatia's method costs 20.3125 s, the Gauss method costs 22.8817 s and our method costs 17.0115. Thus, our method is more efficient to detect the critical points. At the same time, we get additional Poincaré indices (Fig. 3).

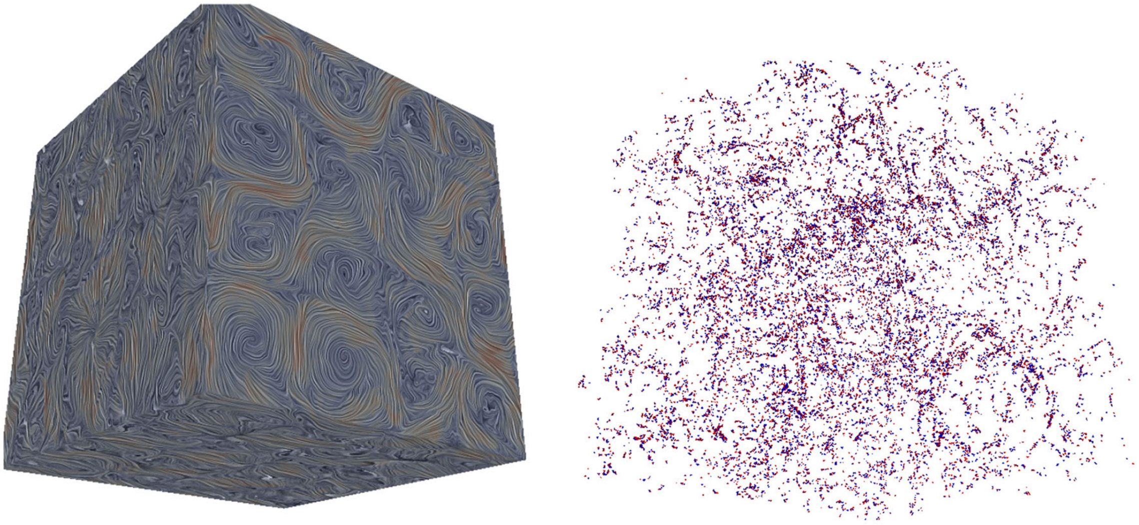
6 Conclusion

In this paper, we discuss the detection and classification of critical points in piecewise linear vector field, which can be viewed as the generalization of Bhatia's work (Bhatia et al. 2014).

In conclusion, we summarise the main points in this paper as follows:

- We discuss the relationship between Poincaré index and Brouwer degree in the case of piecewise linear vector fields (Proposition 1).
- We discuss the relationship between Poincaré index and eigenvalues (Proposition 4).
- We prove the simple formula for computation of Poincaré index (Theorem 2).
- We improve the detection method for critical simplices proposed by Bhatia et al. (Proposition 6).
- We discuss the case of triangulated surfaces (Propositions 7 and 8), and propose the simple Algorithm 3.

All these results are proved, and we test the algorithms on both synthetic and experimental data sets, which verify their accurate and simplicity. Although the traditional algorithms are efficient enough, thus, our



(a) Turulence simulation flow.

(b) Extracted critical cells.

Fig. 3 Experimental results for the turbulence data sets. The *blue* and *red colors* represent indices −1 and +1, respectively

methods are not significant in the improvement of efficiency. However, our formulations make them much simpler, and lead to less ambiguities.

As far as we know, we are the first to propose a simple method to compute the Poincaré index during the detection, without much additional effort. In addition, furthermore, except the surface case, the robust computation of Poincaré indices implies topological consistence, which may used as criteria in other applications with topological sensitivity.

In the future, we can use it to help to detect and verify features in vector fields, for example, the vortex core. On the other hand, the classification of Poincaré index is still not fine enough, so we should develop robust techniques for other topological invariants, e.g., the Conley index (Chen et al. 2012).

Acknowledgements This work is supported by Chinese 973 Program (2015CB755604) and the National Science Foundation of China (61202335). We would like to thank Dr. He Ouyang for his guidance.

References

Bhatia H, Gyulassy A, Wang H, Bremer P-T, Pascucci V (2014) Robust detection of singularities in vector fields. In: Topological methods in data analysis and visualization III. Springer, pp 3–18

Helman J, Hesselink L (1989) Representation and display of vector field topology in fluid flow data sets. Computer 22(8):27–36

Pobitzer A, Peikert R, Fuchs R, Schindler B, Kuhn A, Theisel H, Matkovic K, Hauser H (2010) On the way towards topology-based visualization of unsteady flow-the state of the art. H. und E. Reinhard (Hrsg.), Eurographics

Laramee RS, Hauser H, Zhao L, Post FH (2007) Topology-based flow visualization, the state of the art. In: Hauser H, Hagen H, Theisel H (eds) Topology-based methods in visualization. Mathematics and visualization. Springer, Berlin Heidelberg, pp 1–19

Wang W, Wang W, Li S (2016) From numerics to combinatorics: a survey of topological methods for vector field visualization. J Vis 19(2):727–752

Lavin Y, Batra R, Hesselink L (1998) Feature comparisons of vector fields using earth mover's distance. In: IEEE proceedings of visualization'98, pp 103–109

Batra R, Hesselink L (1999) Feature comparisons of 3-d vector fields using earth mover's distance. In: Proceedings of the conference on Visualization'99: celebrating ten years. IEEE Computer Society Press, pp 105–114

Mann S, Rockwood A (2002) Computing singularities of 3d vector fields with geometric algebra. In: Proceedings of the conference on Visualization'02. IEEE Computer Society, pp 283–290

Chen G, Mischaikow K, Laramee RS, Pilarczyk P, Zhang E (2007) Vector field editing and periodic orbit extraction using morse decomposition. IEEE Trans Vis Comput Graph 13(4):769–785

Wang H, Deng J (2014) Feature extraction of complex ocean flow field using the helmholtz-hodge decomposition. In: 2014 IEEE international conference on multimedia and expo workshops (ICMEW), pp 1–6

Reininghaus J, Hotz I (2011) Combinatorial 2d vector field topology extraction and simplification. In: Topological methods in data analysis and visualization. Springer, pp 103–114

Garth C, Tricoche X, Scheuermann G (2004) Tracking of vector field singularities in unstructured 3d time-dependent datasets. In: Proceedings of the conference on Visualization'04. IEEE Computer Society, pp 329–336

Wang W, Wang W, Li S (2016) Batch advection for the piecewise linear vector field on simplicial grids. Comput Graph 54:75–83

Hsu S-B (2006) Ordinary differential equations with applications, 2nd edn. World scientific, Singapore

Edelsbrunner H, Mücke EP (1990) Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. ACM Trans Graph (TOG) 9(1):66–104

van der Waerden, B. Leendert, Algebra i (7th ed.)

Hirsch MW, Smale S, Devaney R (2013) Differential equations, dynamical systems, and an introduction to chaos, 3rd edn. Elsevier, New York

Stommel H (1949) Trajectories of small bodies sinking slowly through convection cells. J Mar Res 8(11):24–29

Maxey M (1987) The motion of small spherical particles in a cellular flow field. Phys Fluids 30(7):1915–1928

Laramee RS, Weiskopf D, Schneider J, Hauser H (2004) Investigating swirl and tumble flow with a comparison of visualization techniques. In: IEEE visualization, 2004, pp 51–58

Eyink G, Vishniac E, Lalescu C, Aluie H, Kanov K, Bürger K, Burns R, Meneveau C, Szalay A (2013) Flux-freezing breakdown in high-conductivity magnetohydrodynamic turbulence. Nature 497(7450):466–469

Chen G, Deng Q, Szymczak A, Laramee RS, Zhang E (2012) Morse set classification and hierarchical refinement using conley index. IEEE Trans Vis Comput Graph 18(5):767–782