IADIS International Journal on Computer Science and Information Systems
Vol. 18, No. 2, pp. 178-193
ISSN: 1646-3692

A VISUAL ANALYSIS METHOD FOR VECTOR FIELDS DEFINED ON CURVED SURFACES

Fukun Wu[1,2], Yi Cao[2] and Qingli Dong[1,2]
[1]*CAEP Software Center for High Performance Numerical Simulation, Beijing, China*
[2]*Institute of Applied Physics and Computational Mathematics, Beijing, China*

ABSTRACT

We introduce a surface streamline generation approach for visualizing vector fields defined on curved surfaces. This approach performs the intersection calculation on curved surfaces with complex topology to achieve the highly detailed underlying vector. Then, an extended Runge-Kutta streamline integration technology is applied for performing streamline tracing on an unstructured mesh, where the adaptive stepsize strategy and intersection acceleration structure are presented for sake of simplicity and efficiency. Finally, this algorithm applies the ball feature to improve its visual intuitiveness and is integrated into the general visual analysis platform. Experimental results show that our method can generate continuous and consistent geometric surface streamlines by tracing streamlines on curved surfaces.

KEYWORDS

Surface Streamline, Extended Runge-Kutta, Adaptive Stepsize Strategy

1. INTRODUCTION

The analysis of flow on polyhedral surfaces is of particular importance for design and optimization of complex geometry in computational fluid dynamics, which plays a key role in visual analysis of electromagnetic shielding effects, surface currents on electronic equipment and flow characteristics inside engine combustion chambers from physical sciences and engineering.

A significant body of research is dedicated to vector field visualization such as LIC (Battke et al., 1997; Cabral and Leedom, 1993; Mao et al. 1997), ISA (Laramee et al. 2003), or IBFVS (van Wijk, 2003), in order to support exploration of flow on large and unstructured polygonal meshes. Although texture-like representations allow to increase the spatial resolution and to depict small details accurately, it is still challenging to visualize the flow feature on complex models for engineering designer. Covering an image with a set of evenly spaced streamlines is a good way to visualize the flow features. However, image quality enhancement needs to be achieved by using streamline placement algorithms which optimize the placement of a set of streamlines according to an image-based criterion. In other words, these methods can generate sparse or dense representations of vector fields. However, the generation and advection of texture properties in object space is finally projected to image space.

Directly performing streamline tracing on 3D curved surfaces is interesting due to the visual intuitiveness from the perspective of a designer. The work on streamline tracing visualization has been concentrated on Runge-Kutta integration methods which can't adapt to the constraints of complex geometric surfaces due to the intrinsic properties. It is difficult to generate the continuous and consistent streamlines on curved surfaces with the challenge of the robust intersection testing and numerically stable methods. Polthier and Schmies (2006) introduced straightest geodesics in geometry to solve the initial value problem for geodesics on polyhedral surfaces. However, their algorithm only uses intrinsic geometric properties of polyhedral surfaces, not considering underlying discrete triangulation of surfaces in vector field.

This paper proposes an efficient streamline generation algorithm for the analysis of flow on curve surfaces. Different from previous work, we introduce the high-precision polygon intersection and interpolation algorithm to achieve vector field extraction on surfaces with complex topology which keeps the underlying detail of geometric surfaces in vector field. Then, we present surface streamline integration technology to perform streamline tracing on unstructured mesh where the fourth-order Runge-Kutta integrator is extended to 2D surfaces.

2. RELATED WORK

Visualizing vector fields defined on curved surfaces or manifolds is of particular importance for engineering designer in computational fluid dynamics and has received much attention. Due to the complex topology of CAD geometries with holes and discontinuities, to use a technique based on surface parametrization would be especially difficult. The research on the visualization of vector fields defined on surfaces focuses on texture-based approaches (Laramee, 2004; Stalling and Hege, 1997; Li et al. 2008). In the early 1990s, Spot Noise (van Wijk, 1991) and LIC (Cabral and Leedom, 1993; Forssell and Cohen, 1995) were presented to generate dense representations based on textures which are limited to curvilinear surface in 2D. An extension of LIC for arbitrary surfaces in 3D were proposed by Mao et al. (Mao et al., 1997) where the convolution of noise image with filter kernels are performed only at visible ray-surface intersections. See Stalling and Hege (1997) for more comprehensive overviews of LIC techniques applied to surfaces. In order to overcome computation time hurdle introduced by LIC, two representative approaches, ISA (Laramee et al., 2004) and IBFVS (van Wijk, 2003), were proposed to generate dense representation of flow on complex surfaces at fast frame rates. Although texture properties are advected on boundary surfaces in 3D, they essentially realized texture advection in image space in 2D by projecting the surface geometry and its associated vector field to image space and then applying texturing.

Covering an image with a set of streamlines is also a good way to visualize the flow features. A significant body of research is dedicated to using image-based approach to generate streamlines with different streamline seeding strategies for vector fields defined on curved surfaces (Li and Shen, 2007; Mattausch et al., 2003; McLoughlin et al., 2010; Spencer et al., 2009; Verma et al., 2000; Ye et al., 2005). These methods can generate sparse or dense representations of vector fields, and image quality enhancement can be achieved by using streamline placement algorithms based on an image criterion. However, the generation and advection of texture properties is still confined to image space. It is still difficult to generating surface streamlines directly on complex CAD models, and the quality is affected by various factors such as vector field projection deformation and seed point distribution.

Generating geometric streamlines in object space is an efficient means for engineering designers to perform flow field analysis directly based on complex geometric models due to their visual intuitiveness. This can help them analyze the flow feature of arbitrary positions on curved surfaces in order to aid the design of computational fluid dynamics models. However, it is very difficult to perform streamline tracing on an unstructured mesh with the challenge of the robust intersection testing and numerically stable methods of handing special cases (Reshetnyak, 1989). Geodesic curves in geometry solve the initial value problem on polyhedral surfaces on smooth surfaces and play a key role to streamline generation (Polthier and Schmies, 2006; Spencer et al., 2009; Alexander and Bishop, 1996). For instance, Polthier and Schmies presented an efficient straightest geodesics method to generate streamlines on polyhedral surface and arbitrary manifolds, which allows to move uniquely on a polyhedral surface in a given direction along a straightest geodesic. Their work only uses geometric properties of the polyhedral surface and not considers the underlying discrete triangulation of the surface.

At the same time, the special or general post-processing software such as Tecplot, ParaView and VisIt rely on output results of the solvers, which makes it difficult for users to directly obtain the dynamic trend of the flow field on curved surfaces from complex models. Inspired by these works, we achieve the highly detailed underlying vector by surface extraction and present a novel and efficient method of directly generating streamlines on curve surfaces from complex models to make up for the deficiencies of commercial software.

3. ALGORITHM

The streamline generation algorithm proposed in this paper is applied to vector fields defined on complex curved surfaces. The key to performing streamline tracing is to accurately calculate the projection of the streamline on curved surfaces and the integral value of the streamline at different steps.

Our algorithm consists of three parts. First, it performs the vector extraction operation by applying the polygon intersection and interpolation algorithms of geometric surfaces with vector field in order to achieve a high-precision vector field defined on curved surfaces. Then, it applies the surface streamline integration technology to generate streamlines with the aid of specific optimization operations. Finally, this algorithm is incorporated into the visual analysis platform with ball feature rendering algorithm to achieve expressive dynamic simulation of flow field on surfaces.

3.1 Surface Extraction

The streamline generation algorithm proposed in this paper is applied to vector fields defined on complex curved surfaces. The key to performing streamline tracing is to accurately calculate the integral value of the streamline on curved surfaces. From the viewpoint of engineering designer in computation fluid dynamics, the surface streamline generation algorithm is essentially different from the traditional image-based streamline visualization algorithms. It is limited by the geometry model with high-order, discontinuous and multi-scale characteristics, and firstly requires the high-precision acquisition of the vector field on the geometric surface.

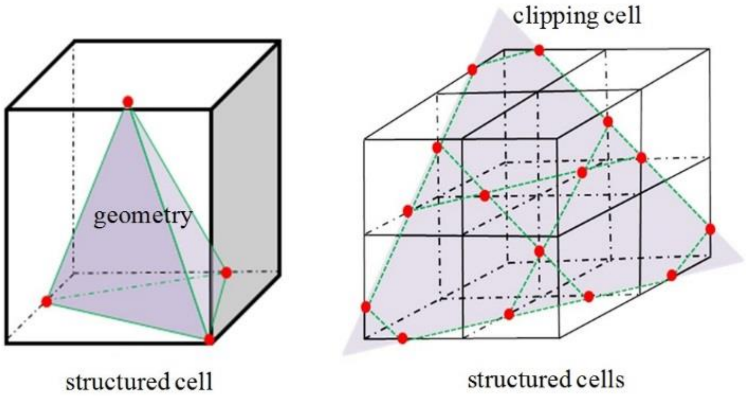


Figure 1. Two cases that a geometry and its surface clipping cell intersecting with volumetric cells in vector field

Figure 1 illustrates the intersection cases of geometry and vector fields where red points denote calculated intersection points and polygons surrounded by green lines are extracted valid polygons. Here, our algorithm extends the previous polygon clipping approaches presented by Greiner and Horman (1998) and Vatti (1992) to 3D curved surfaces to achieve the high-precision surface extraction which keeps the underlying detailed structures of geometric surface on vector field.

Considering the fact that vector field may be composed of structured or unstructured grids, this paper applies the grid-related interpolation algorithms to calculate the vector field on extracted surfaces. For unstructured grids, we adopt the trilinear interpolation method to calculate the vector on arbitrary point on curved surfaces. For structured grids, domain-related interpolation methods are applied.

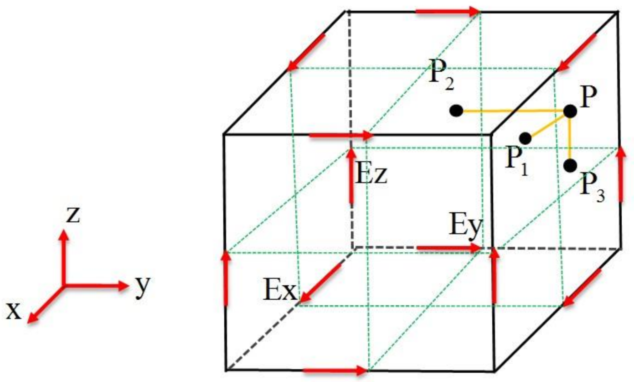


Figure 2. Geometric structure of vector field *E* defined on the hexahedral cell

Figure 2 shows an example of the structured grid generated by the numerical solver. The physical quantity *E* in vector field is registered as the center of the grid cell, but the components of the vector *E* are located at the edge center of the grid cell. In order to calculate the vector value of any node *P* on the complex geometric surface, three reference planes, as shown in the area enclosed by the three groups of green dashed lines in Figure 3, are introduced for vector field defined on structured grids. The projection points of the point P on the three reference planes are P_1, P_2, and P_3 respectively. P_1 is used to calculate the x-axis component Ex of the vector field E associated with P. P_2 and P_3 are used to calculate y-axis component Ey and z-axis component Ez respectively.

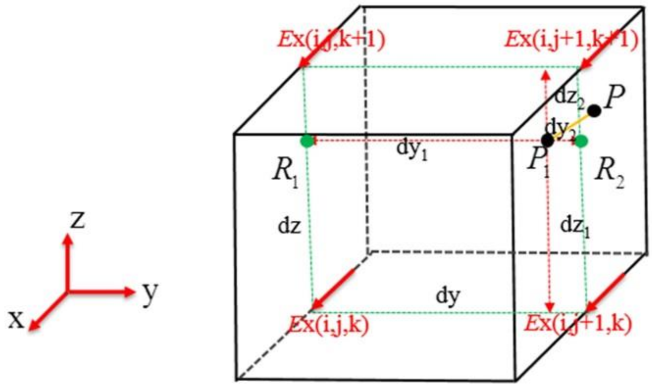


Figure 3. Geometric structure of vector field E defined on the hexahedral cell

Figure 3 shows the example of calculating the projection point P_1, and the complete linear interpolation formula is:

$$Ex(P_1) = \frac{dy_1}{dy} * Ex_{R_1} + \frac{dy_2}{dy} * Ex_{R_2} \tag{1}$$

where,

$$\begin{aligned} Ex_{R_1} &= \frac{dz_1}{dz} * Ex(i, j, k) + \frac{dz_2}{dz} * Ex(i, j, k+1) \\ Ex_{R_2} &= \frac{dz_1}{dz} * Ex(i, j+1, k) + \frac{dz_2}{dz} * Ex(i, j+1, k+1) \end{aligned} \tag{2}$$

Substituting the Ex_{R_1} and Ex_{R_2} in Equation 2 into Equation 1 can obtain the component of Ex(P_1). Ey(P_2) and Ez(P_3) can be obtained in the same way.

3.2 Surface Streamline Generation

Through extracting vector field using curved surfaces, we can obtain a high-precision surface vector field which provides the pre-processing data for streamline generation. Then, according to the initial seed point specified interactively by engineering designer, a streamline moving along the geometric surface needs to be generated. However, due to the limitation from the complex curved surfaces, the classical Runge-Kutta integrators is no longer valid. Focusing on this central challenge, this paper adopts a new and extended Runge-Kutta method based on surface projection for integration calculation, with the aid of an adaptive stepsize strategy and streamline-surface intersection acceleration structure, to accomplish the continuous and consistent surface streamline generation.

3.2.1 Runge-Kutta Integral

The core idea of the surface streamline generation algorithm is to track and calculate the motion trajectory of these seed points on curved surfaces in vector field. The trajectory is a curve, namely a streamline. For each streamline $x(t)$, its motion equation satisfies:

$$\frac{dx_t}{dt} = V(t, x) \tag{3}$$

Where $V(t, x)$ is the value of vector field at the position $x(t)$, and for the static field, $V(t, x) = V(x)$.

Considering the case that surface streamlines are constrained by geometric shapes, this paper defines an extended Runge-Kutta streamline integral formula as

$$x(t+h) = x(t) + (Proj_{\Omega}(\int_{t}^{t+h} V(t, x)dt)) \tag{4}$$

Where Ω represents the geometric surface and $Proj_{\Omega}$ represents the projection of the integral formula on the geometric surface. The choice of h is more important. If h is too small, the calculation overhead will be too large. If h is too large, it will bring greater errors and reduce the accuracy of streamline generation.

For generating continuous surface streamline in vector field, we need to iteratively calculate the integral points of the streamline according to the Equation 4, and finally obtain a complete streamline that can show the move trend of flow field. We adopt the fourth-order Runge-Kutta method used for concrete calculation. This method is used as the basis of solving the derivative and initial values in differential equations. Specifically, the next step value x_{i+1} of the streamline is determined by the current value x_i plus the projection on the surface of the product of the step interval h and an estimated slope, formulated by

$$x_{i+1} = x_i + Proj_{\Omega}(h(K_1 + 2K_2 + 2K_3 + K_4)/6) \tag{5}$$

Where

$$\begin{aligned} &K(1)=V(t_i, x_i) \\ &K(2)=V(t_i + \frac{h}{2}, x_i + \frac{hK_1}{2}) \\ &K(3)=V(t_i + \frac{h}{2}, x_i + \frac{hK_2}{2}) \\ &K(4)=V(t_i + h, x_i + hK_3) \end{aligned} \tag{6}$$

3.2.2 Stepsize Adaptive Adjustment

When iteratively performing streamline integration, the stepsize is comprehensively weighed according to the calculation accuracy and efficiency, and it has a decisive influence on the local error. When using a fixed step size, the complexity of the projection calculation becomes a time-consuming problem. As shown in Figure 4, s_1, s_2, and s_3 represent three polygonal cells on the geometric surface, and p_1, p_2, and p_3 represent the projection points of streamline integration on surface. The structural diversity of curved surfaces increases the complexity of generating surface streamlines after projection.

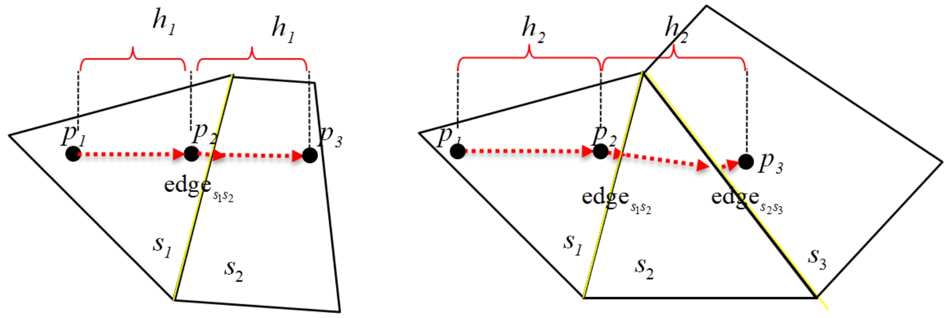


Figure 4. Surface projection under two types of integration with fixed steps h1 (left) and h2 (right)

For solving this problem, we construct an adaptive stepsize method based on the grid size and shape to calculate and generate surface streamline as shown in Figure 5.

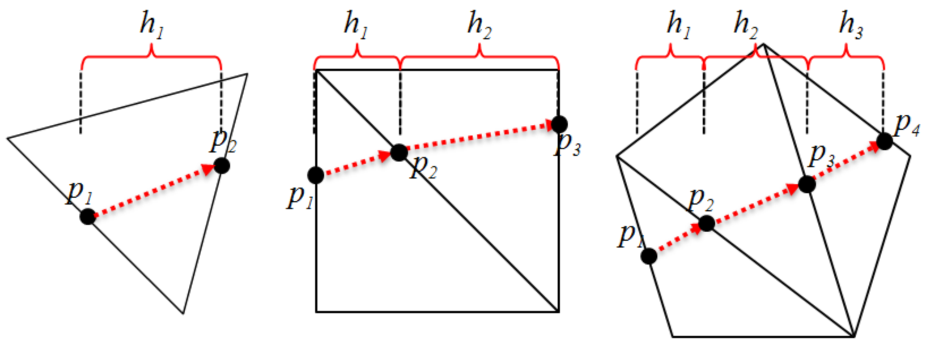


Figure 5. Classification of the adaptive stepsize on a polyhedral surface according to the shape of polygon

According to the type of polygon cells on curved surface, different stepsizes are processed during surface integration. For a triangle, the integral projection directly crosses the internal field of the triangle, and the step length is only h_1. The quadrilateral or pentagon is triangulated separately to form two step sets h_1h_2 and $h_1h_2h_3$, where there is still only one integration step inside each triangle. This algorithm avoids the complicated calculations when the streamlines are projected on the geometric surface, and at the same time takes into account the grid shape and the changes of cell volume. When visualizing streamlines, we synthesize surface streamlines orderly according to the adaptively generated projection pointsets.

3.2.3 Intersection Acceleration

Surface streamline generation algorithm involves how to accurately calculate the projection point of the streamline on curved surfaces. When the resolution of the dataset grows, the geometric surface extraction operation will generate a large number of surface polygons. Therefore, a lot of time will be consumed in the calculation of the streamline integral during the surface projection.

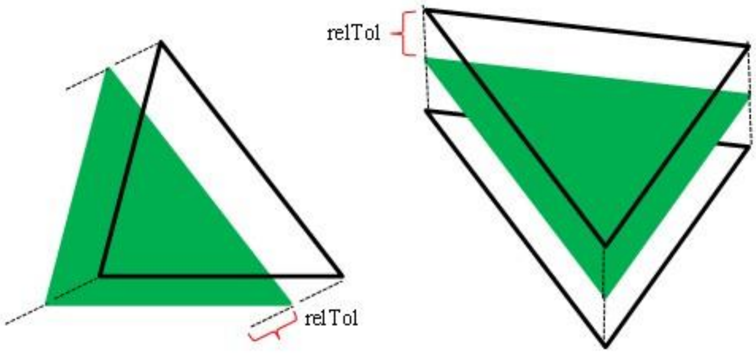


Figure 6. Notion of constructing bounding box of geometric surface with the parameter relTol

Considering the surface topology, we introduce the spatial threshold relTol in the vertical direction of the surface to construct the surface-based volume cell, as shown in Figure 6. Based on this, an octree-based acceleration structure of the geometric surface is constructed, where the spatial volume represented by each polygon represents a valid node information in the octree.

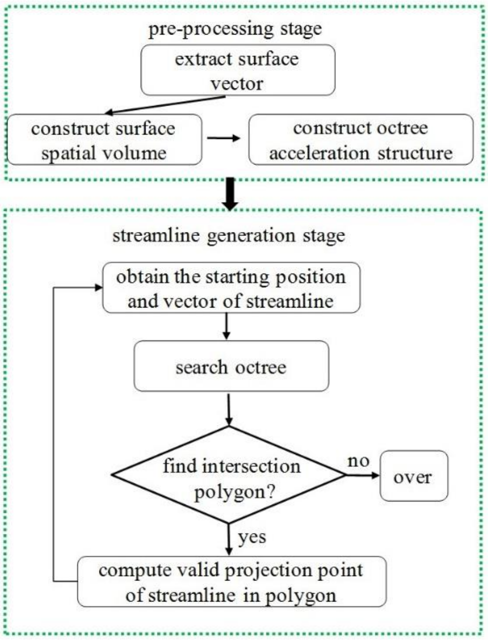


Figure 7. An overview diagram of generating surface streamlines using octree acceleration structure

The complete process of performing the surface streamline generation is shown in Figure 7. It includes two main stages, namely the pre-processing stage and the streamline generation stage. In the preprocessing stage, its core purpose is to extract vector fields from large-scale simulation data based on the geometric model provided by the user. Here, domain related interpolation methods for structured and unstructured grids will be applied, resulting in high-precision distribution of these vector fields on the geometric surface, and their accuracy is consistent with that of the data field. The purpose of the streamline generation stage is to perform two-dimensional integration calculations dependent on the geometric surface. To avoid a large amount of time consumption, the octree acceleration structure will be constructed to improve the intersection efficiency. At the same time, cycle to calculate the next integration point after calculating one integral point.

3.3 Dynamic Feature

Generally, the geometric form of each streamline is a polyline, which can be drawn directly or by a pipe surface. This paper introduces a feature ball to depict the flow trend in the way of moving small balls.

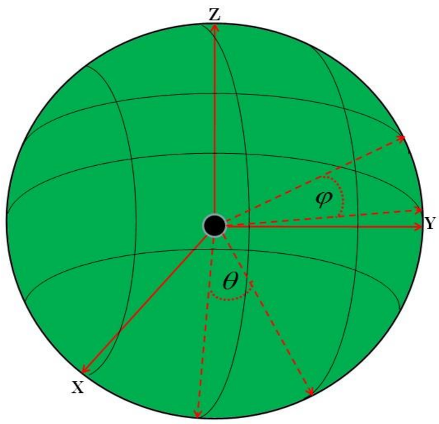


Figure 8. Geometric diagram of ball generation dependent on latitude and longitude constraints

Figure 8 shows the geometric diagram of generating the feature ball, where the longitude θ and latitude φ control the polygon density on the surface of the ball. The small ball starts from the seed point and moves along the streamline, cyclically and repeatedly, in order to dynamically display the trend of the flow field. In particular, our rendering algorithm supports two types of coloring modes based on the physical field value and the number of integration steps.

3.4 Limitation

Numerical robustness is an issue to deal with floating-point computations in computational geometry, and many methods have been proposed to overcome the problems of finite precision and inexact data. Generally, there exit some degeneracies where each vertex of one polygon does not accurately lie on an edge of the other polygon. In the work of Greiner and Hormann (1998), they assume that if the perturbation is less than pixel width, the output on the screen will be correct. However, it requires very high accuracy for our extracting model, and also provides interactive features such as zooming that allows the user to observe local details. Here we do a small modification by setting $perturbation = min\{0.1 \cdot \Delta w, 0.01 \cdot \Delta p\}$ according to the actual data resolution where Δp denotes minimum diagonal length of volume cells and Δw denotes pixel width. In addition, because of our extracting model being built on the intersection of plane and volume, the cell type of volumetric data is limited to tetrahedron, hexahedron, voxel, wedge or pyramid.

4. EXPERIMENTAL RESULTS AND ANALYSIS

We implemented our surface streamline generation algorithm and integrate it into a general visual analysis platform, namely VisIt (Childs et al., 2005), in the form of plugin-in component. We also demonstrated its validity by applying our method to representative datasets for visualizing the surface field of objects with regular and irregular geometry. All of the images in this work were produced on a Dell T7600 workstation with a 2.40 GHz Intel Xeon CPU E5-2609.

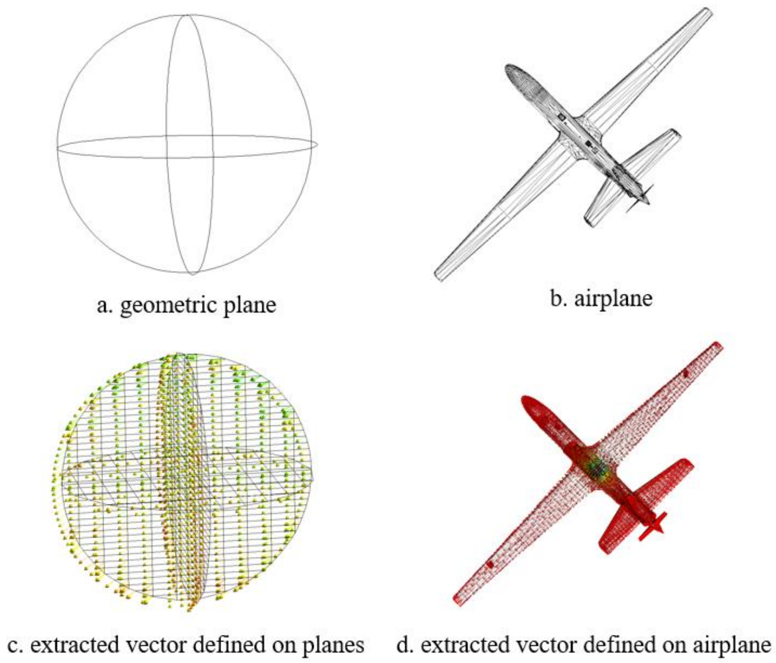


Figure 9. Visualization results of extracting high-precision surface vectors defined on planes and aircraft

Calculating vector field defined on the curved surfaces is the first step of generating surface streamlines. For geometric models with arbitrary topological structures, this paper applies the high-precision extraction algorithm to obtain the surface vector field with the same mesh density and accuracy as original 3D vector field, meanwhile to maintain the profile of the geometric model. Figure 9 shows the visualization results of the extracted surface vector field using plane and aircraft models. Figure 9a and Figure 9b show the geometric structures of planes and airplane respectively, and Figure 9c and Figure 9d show the extracted vector defined on planes and airplane. As shown in the Figure 9, our vector field extraction algorithm is not limited by the complex topology of the geometric model, and can generate vector field with high accuracy.

In addition, Table 1 provides a concise overview of performance by extracting surface vector field using cylinder and sphere models from two different types of datasets under 6 CPU cores, where the consumed time can be further reduced with the increase of the CPU cores.

Table 1. Performance comparison of extracting two types of datasets using cylinder and sphere under 6 CPU cores

Type of dataset	Num of cells	Time(cylinder)	Time(sphere)
Structured grid	36000	0.257 s	0.189 s
Unstructured grid	45618700	3.872 s	3.591 s

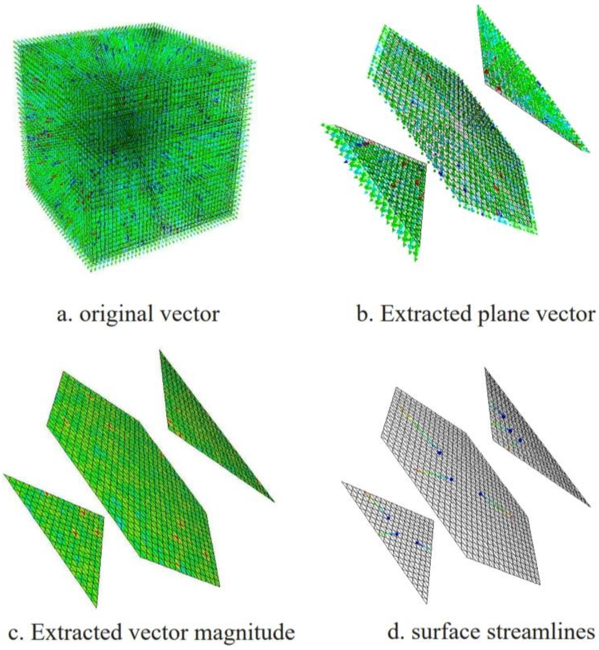


Figure 10. Visualization of generating surface streamlines using geometric plane

Figure 10 shows the complete process of generating surface streamline. Figure 10a denotes a structured vector field composed of 36 grid patches and a total of 36,000 hexahedrons. Figure 10b shows the visualization result of surface extraction using three planes where the vector field is retained with high accuracy through our vector field interpolation algorithm related to structured grid. Figure 10c shows the visualization result of the magnitude of the extracted vector field. Figure 10d presents the streamlines generated by the proposed surface streamline algorithm with 9 initial seed points. It can be seen that our algorithm can accurately generate continuous and consistent surface streamlines constrained by the geometric surface. The direction of generated streamlines is consistent with the vector field.

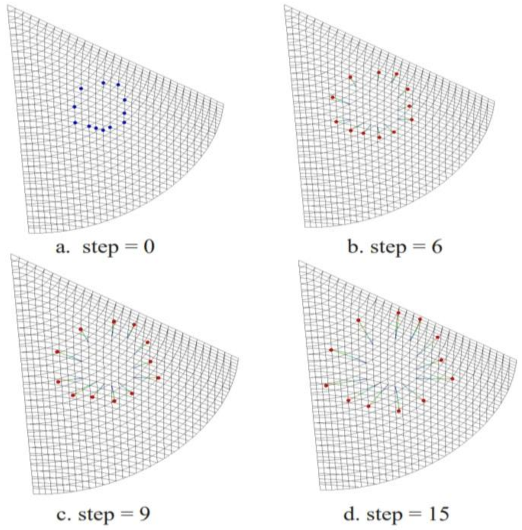


Figure 11. Visualization of generating surface streamlines under different integration step

Numerical integration is the central feature of surface streamline generation. This paper sets the number of integration steps to be 0, 6, 9 and 15 respectively, and generates the corresponding surface streamlines as illustrated in Figure 11. The coloring method of the streamline adopts the steps as the color mapping standard. It can be seen from Figure 11 that the streamline integral points generated by our streamline generation algorithm fully fit the surface of the geometric model and are not constrained by the geometric model. Its own trajectory is also consistent with the direction of vector field defined on the surface, which is determined by the extended Runge-Kutta equation.

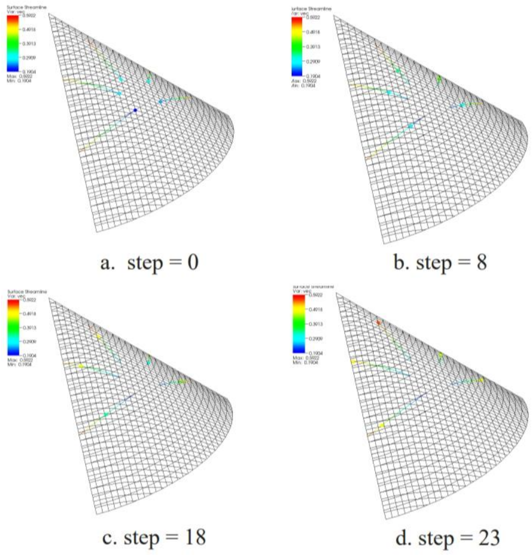
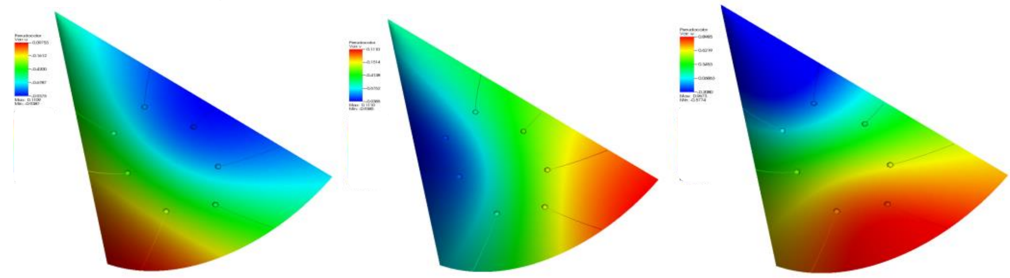


Figure 12. Visualization of moving small balls along streamline direction with different integration step

In order to enhance the visual effect of dynamic generation of streamline, this paper introduces the small ball feature to identify the moving position under different integration steps in order to dynamically describe the moving trend of the vector field defined on the curved surfaces. Figure 12 shows the visual effects of the moving small ball along the streamline at the integration steps of 0, 8, 18, and 23 respectively. Users can understand the surface flow field characteristics through the movement of these small balls.



a. variable = u   b. variable = v   c. variable = w

Figure 13. Visualization of coloring surface streamline using different physical variable. The comparison with the underlying scalar field shows the expected better approximation quality

In addition to the coloring method based on streamline distance corresponding to Figure 11, A very important feature in streamline rendering is to support the visualization of streamline colors based on arbitrary physical field values. Figure 13 shows the surface streamline coloring effects based on the four types of scalar fields u, v, and w, with the corresponding scalar field coupled in the background. It can be concluded that our surface streamline generation algorithm can effectively describe the flow characteristics of the vector field and the changing trend of physical quantities.

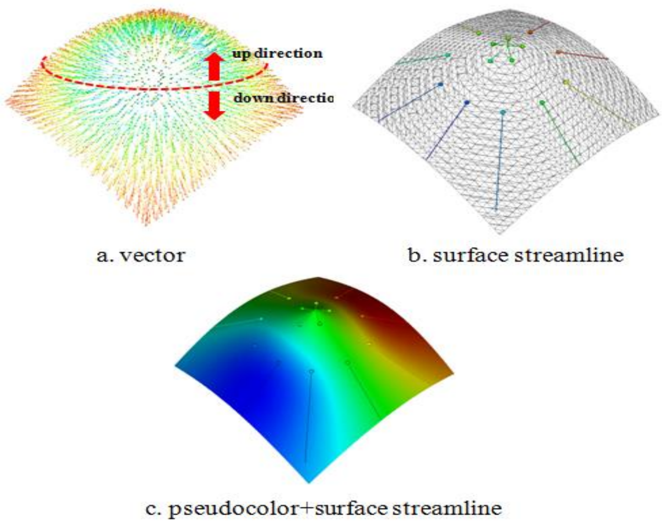
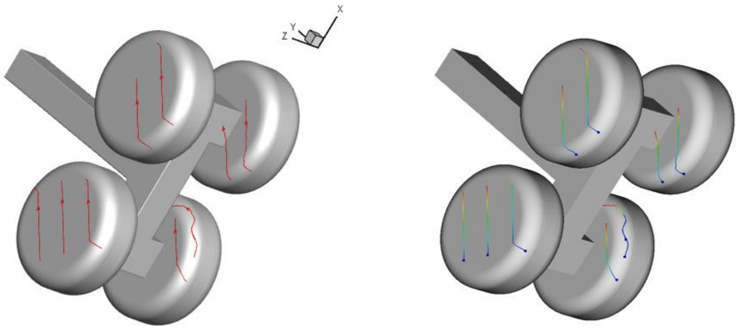


Figure 14. Proximity comparison: (top row) between the underlying direction of motion in vector field and the generated streamlines using our algorithm, (bottom row) between scalar field and our coloring method. The underlying structure of the flow mesh is now more clearly reflected by the streamlines

The complexity of the geometric surface models varies with numerical devices simulated in the field of engineering. As shown in Figure 14, this paper selects another common geometric surface model for surface streamline generation and verification. Figure 14a shows the high-precision vector field obtained after surface extraction. The upper half of the vector diverges upward, and the lower half of the vector diverges downward, that is, there should be two completely different directions of surface flow on the entire surface. Figure 14b shows the continuous and consistent surface streamlines generated using 14 seed points, which is the same as the trend of the surface flow field, where the color is generated based on the physical field u. Figure 14c is a schematic diagram of the coupling of the pseudocolor diagram and surface streamlines. Figure 14c verifies the flexibility and accuracy of the algorithm in this paper when coloring the surface streamline. In particular, the above experiments further prove that the proposed surface streamline method is not limited by the shape of the geometric model and the characteristics of the vector field.

In addition, a comparison with Tecplot's streamline generation results as illustrated in Figure 15 also demonstrates the expected high approximation quality of our proposed algorithm.



a. Tecplot b. our work

Figure 15. Comparison of generating surface streamlines on curved surface using Tecplot and our method

5. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a streamline generation algorithm for vector field defined on curved surfaces. This algorithm achieves the underlying discrete feature of curved surfaces by applying the polygon intersection and the grid-related high-precision interpolation algorithms. Then, we present the surface streamline integral technology, combined with the adaptive stepsize strategy and acceleration structure, in order to support continuous and consistent surface streamline generation, which solves the bottleneck problem of performing streamline tracing directly on curved surfaces. We have shown that this work can aid engineering designer for the design and optimization of major devices by analyzing flow on curved surfaces.

In future, we focus on the adaptive adjustment of seed points to enhance the interaction and friendliness from the viewpoint of designer. We will also study the asynchronous calculation mechanism in order to improve the efficiency of parallel generation of surface streamlines in unsteady flow visualization.

REFERENCES

Alexander, S. and Bishop, R. L. (1996). Comparison theorems for curves of bounded geodesic curvature in metric spaces of curvature bounded above. *Differential Geometry and its Applications*, Vol. 6, No. 1, pp. 67-86.

Battke, H. et al. (1997). Fast line integral convolution for arbitrary surfaces in 3D. In H. C. Hege and K. Polthier (eds) *Visualization and Mathematics*. Springer, Berlin, Heidelberg, pp. 181-195.

Cabral, B. and Leedom, L. C. (1993). Imaging vector fields using line integral convolution. *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, California, USA, pp. 263-270.

Childs, H. et al. (2005). *VisIt User's Manual*. Lawrence Livermore National Laboratory, USA.

Forssell, L. K. and Cohen, S. D. (1995). Using line integral convolution for flow visualization: curvilinear grids, variable-speed animation, and unsteady flows. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 1, No. 2, pp. 133-141.

Greiner, G. and Hormann, K. (1998). Efficient clipping of arbitrary polygons. *ACM Transactions on Graphics*, Vol. 17, No. 2, pp. 71-83.

Laramee, R. S. (2004). *Interactive 3D Flow Visualization Using Textures and Geometric Primitives*. Vienna University of Technology, Institute for Computer Graphics and Algorithms, Vienna, Austria.

Laramee, R. S. et al. (2003). Image space based visualization of unsteady flow on surfaces. *Proceedings of Visualization*. Los Alamitos, USA, pp. 131-138.

Laramee, R. S. et al. (2004). The state of the art in flow visualization: Dense and texture-based techniques. *Computer Graphics Forum*, Vol. 23, No. 2, pp.203-221.

Li, L. and Shen, H.-W. (2007). Image-based streamline generation and rendering. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 13, No. 3, pp. 630-640.

Li, G. et al. (2008). Flow charts: visualization of vector fields on arbitrary surfaces. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 14, No. 5, pp. 1067-1080.

Mao, X. et al. (1997). Line integral convolution for 3D surfaces. *Proceedings of Eurographics*. Vienna, Austria, pp. 57-69.

Mattausch, O. et al. (2003). Strategies for interactive exploration of 3D flow using evenly spaced illuminated streamlines. *Proceedings of Spring Conference on Computer Graphics*, pp. 213-222.

McLoughlin, T. et al. (2010). Over two decades of integration-based, geometric flow visualization. *Computer Graphics Forum*, Vol. 29, No. 6, pp. 1807-1829.

Polthier, K. and Schmies, M. (2006). *Straightest geodesics on polyhedral surfaces*. Association for Computing Machinery, New York, USA, pp. 30-38. https://www.math.kit.edu/iag5/lehre/semgeometrie2016s/media/geodesics.pdf

Reshetnyak, Y. G. (1989). Two-dimensional manifolds of bounded curvature. *Encyclopaedia of Mathematical Sciences*, Vol. 70, pp. 3-164.

Spencer, B. et al. (2009). Evenly spaced streamlines for surfaces: An image-based approach. *Computer Graphics Forum*, Vol. 28, No. 6, pp. 1618-1631.

Stalling, D. and Hege, H.-C. (1997). LIC on surfaces. *Proceedings of Computer Graphics and Interactive Techniques*. Los Angeles, USA, pp. 51-64.

van Wijk, J. J. (1991). Spot noise-texture synthesis for data visualization. *ACM SIGGRAPH Computer Graphics*, Vol. 25, No. 4, pp. 309-318.

van Wijk, J. J. (2003). Image based flow visualization for curved surfaces. *Proceedings of Visualization*. Los Alamitos, USA, pp. 123-130.

Vatti, B. R. (1992). A generic solution to polygon clipping. *Communications of the ACM*, Vol. 35, No. 7, pp. 56-63.

Verma, V. et al. (2000). A flow-guided streamline seeding strategy. *Proceedings of Visualization*. Los Alamitos, USA, pp. 163-170.

Ye, X. et al. (2005). Strategy for seeding 3D streamlines. *Proceedings of Visualization*. Los Alamitos, USA, pp. 471-478.