Conformal Flattening VTK Algorithm

Wenyang Zhou

Oct 3rd, 2019

**Abstract**

This paper describes the Visualization Toolkit (VTK) Conformal Flattening Filter: ConformalFlatteningFilter. This VTK Polydata Algorithm is an implementation of a paper by Sigurd Angenent, et al., “On the Laplace-Beltrami Operator and Brain Surface Flattening” [1]. This filter performs an angle preserving map of any genus zero (i.e. no handles) surface to the sphere or, alternatively, to the plane. In this paper, we describe the code and provide the user with enough details to reproduce the results which we present in this paper. This filter has a variety of applications including the flattening of brain surfaces, which was the initial motivation for this work.

**Contents**

**1 Algorithm Details**………………………………………………………………………………………………………… 2

1.1 Mathematical manipulation ……………………………………………………………………………… 2

1.2 Numerical scheme

**2 User’s Guide**

2.1 Basic usage

2.2 More about APIs

**3 The Filter Test**

**4 Conclusions**

The folding and wrapping of brain surface present researchers with difficulties in obtaining a more intuition-based surface rendering. In recent years, a number of methods have been proposed to map the brain surface to a plane or a sphere. Most of the previous methods are derived to preserve local area or length. Like in the work of [3] and [4], a parameterized deformable surface whose topology is mappable to a sphere is fitted. Thus, the brain surface can be represented on a planar map using spherical coordinates. Also, as in [2] and [5], quasi-isometrics and quasi-conformal flattening schemes are used to flatten the brain surface. Those methods are more functional minimizing schemes and hence bijectivity cannot be guaranteed. In the paper [1], a bijective angle preserving conformal flattening scheme is proposed.

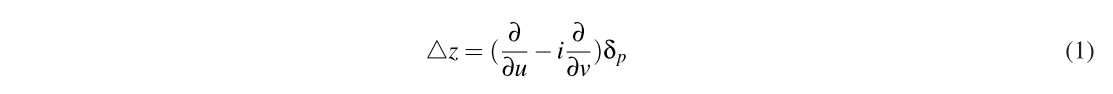
The algorithm obtains the explicit form of the flattening map by solving a partial differential equation on the surface using the finite element method (FEM). The mapping is a bijection thus the triangles do not flip, and the original structure can be restored by inverse mapping.

**1 Algorithm Details**

**1.1 Mathematical manipulation**

Denote the genus zero surface which is to be flattened as . can be mapped to a plane using the algorithm proposed in [1]. The mapping is conformal thus the angles are preserved. Furthermore, the plane can be mapped to a sphere, using standard stereographic projection.

It is proven in the appendix of [1] that the mapping , defined on , satisfying is the following:

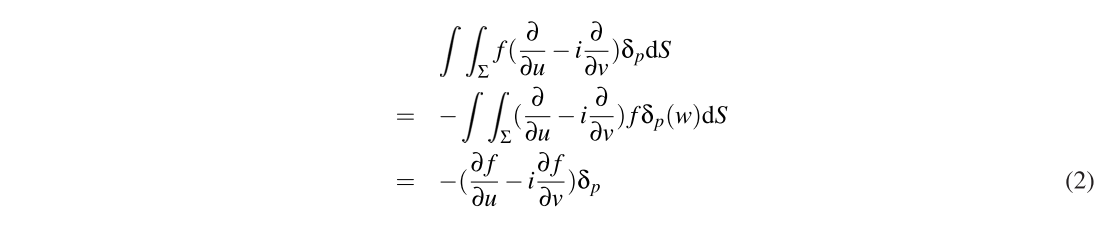


where is an (arbitrary) point on . The function z maps to the complex plane . Then by standard stereographic projection, the complex plane is mapped to a sphere excluding only the north pole.

**1.2 Numerical scheme**

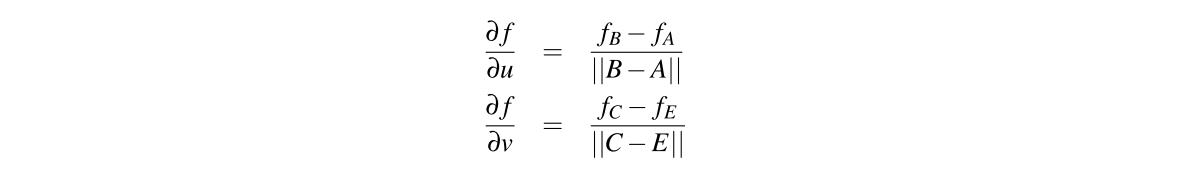
The mapping is defined on the surface. In the numerical manipulation where the surface is denoted as a triangulated mesh, solving of the equation (1) is carried out by FEM.

In the discrete configuration, is approximated by a piecewise linear function on the triangulated mesh. Denote the finite-dimensional space of piecewise linear functions on . To solve equation (1), first its right side should be approximated. For any function smooth in a neighborhood of ,



Thus , given that the point lies in the cell composed by three points , , and , the quantity of (2) is determined by the values of on , , and .

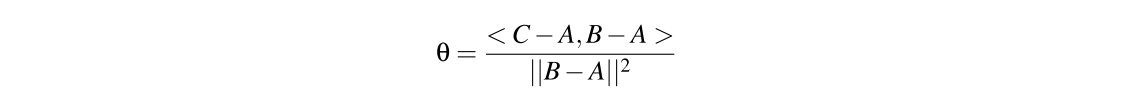
On the triangle , choose the and axes such that and are along axis and the positive direction of axis is pointing to the point . So,



in which is the orthogonal projection of on and can be computed by:

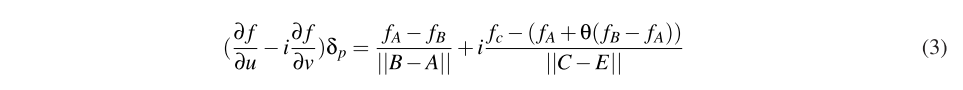


where



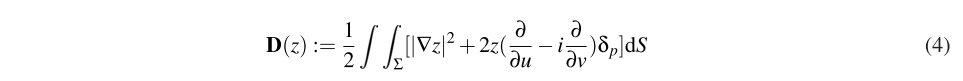
in which <,> denotes inner product.

Thus ,

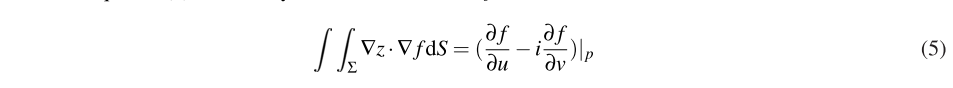


Next, we are going to perform the finite element solution of the mapping in the function space .

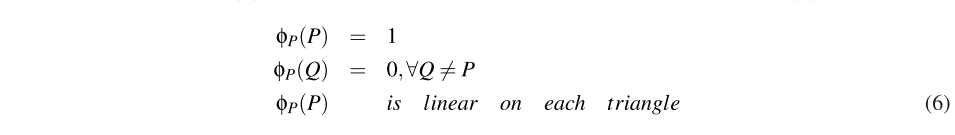
FEM theory indicates that the solver function for equation (1) is the minimizer of the Dirichlet functional:



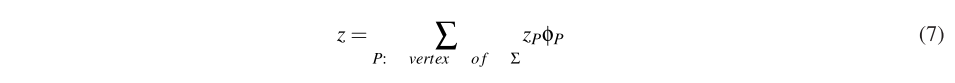
satisfies equation (1) if and only if smooth function ,



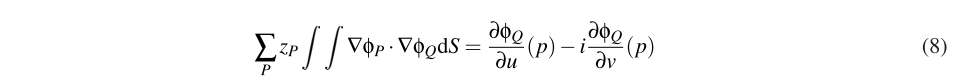
To find the function in satisfying equation (5), we define a set of basis in the as,



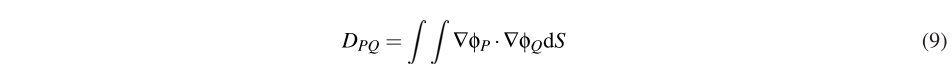
Then function can be expressed as a linear combination of the basis:



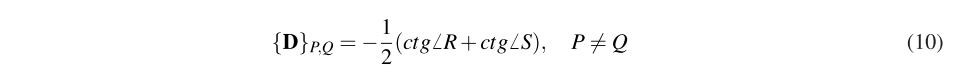
By this we convert the original problem into solving for the complex vector :



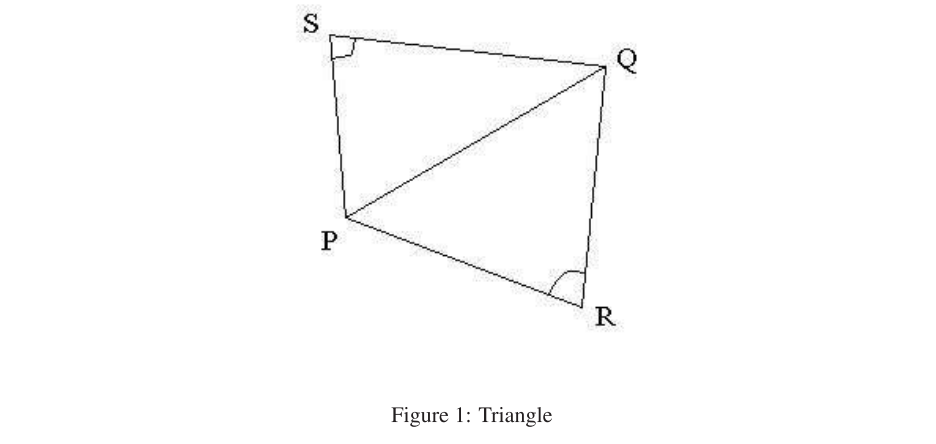
Denote matrix as



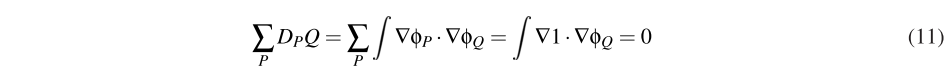
The off-diagonal elements of can be computed by:



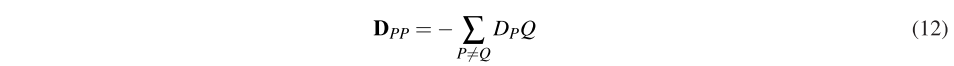
The points are illustrated in figure (1).



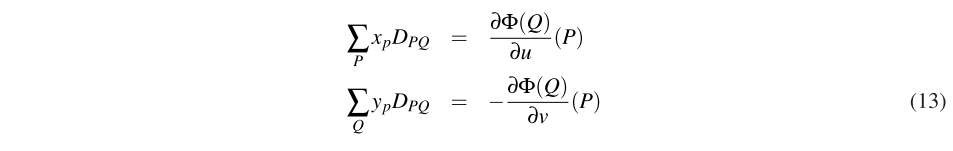
Also, since



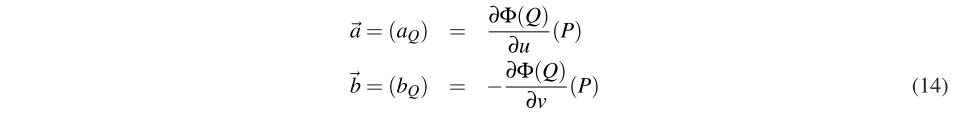
Thus, the diagonal elements of matrix can be obtained by:



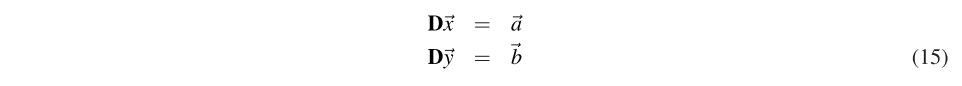
Write each as , then equation (8) can be written separately as:



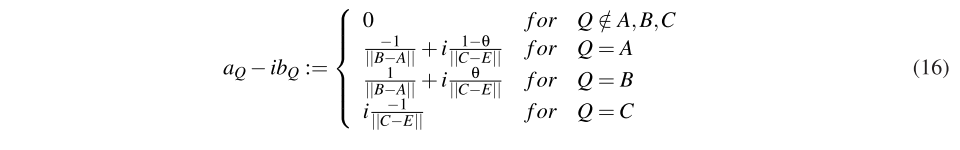
denote:



so we have:



point :



With vectors , , and matrix , we can solve the linear equation to obtain the mapping. The matrix is sparce, symmetric and positively defined, hence is suitable to being solved by the conjugate gradient method.

**2 User’s Guide**

The conformal flattening filter takes an VTK PolyData as input and will generate another VTK PolyData as output. The usage is basically the same as other PolyDataAlgorithms in VTK.

**2.1 Basic usage**

The filter is instantiated by:

*vtkSmartPointer<vtk::ConformalFlatteningMeshFilter> filter =*

*vtkSmartPointer<vtk::ConformalFlatteningMeshFilter>::New();*

Then the input can be set and results can be obtained by:

*filter->SetInputData(inputMesh);*

and

*newPolyData = filter->GetOutput();*

**2.2 More about APIs**

The filter has several APIs for further manipulation of the output.

1. setPointP function. On the right side of equation (1), the function depends on the location of the point . Basically, this point will be mapped to infinity on the plane and the north-pole of the sphere. Hence the selection of the point determines which patch on the original mesh is mapped up to the north pole.

The API setPointP takes an integer as input indicating the cell number in which the point lies. It’s a good choice to set the point where the local surface is relatively flat, i.e., having a small local curvature. If setting the point at some flat area is crucial, we suggest that user first use vtkCurvatures to obtain the number of cells having low curvatures and then call this function using one of the cells with a low curvature.

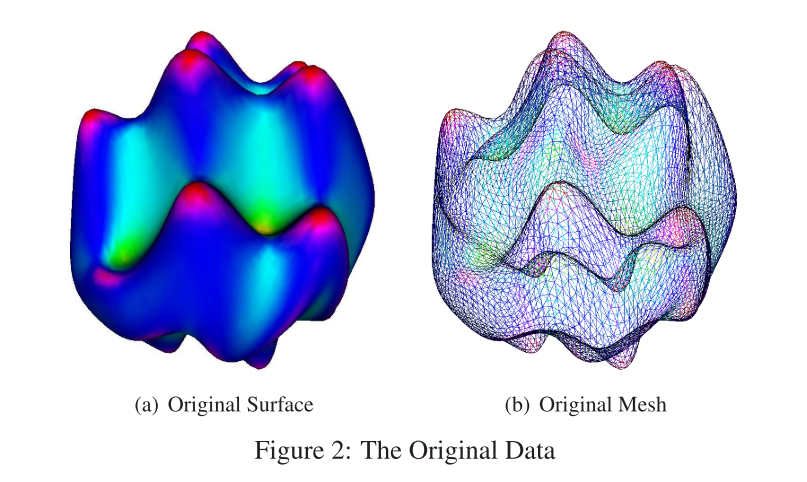
1. The switch functions mapToPlane and mapToSphere determine the output to be either a plane or a sphere, the sphere being the default. The difference between the two mappings is simply a stereographic projection from the plane to the sphere. Simply by

*filter->mapToSphere( );*

or

*filter->mapToPlane( );*

users can switch between two different outputs.



1. setScale function. The mapping, calculated from the equation (1), is dependent on the number of the nodes within the mesh. Given a mesh of a large number of nodes and cells, the image of the flattening mapping, is constrained in a small range around origin. To make it cover a sufficient area of the plane and further get a reasonable result from stereographic projection, re-scale of the flattened plane is needed. This function is used to set the scale factor, by:

*filter->setScale( scale factor );*

For a mesh of around several thousands of nodes, a factor of 100 is likely to get a good result. The factor should grow as the number of nodes grows.

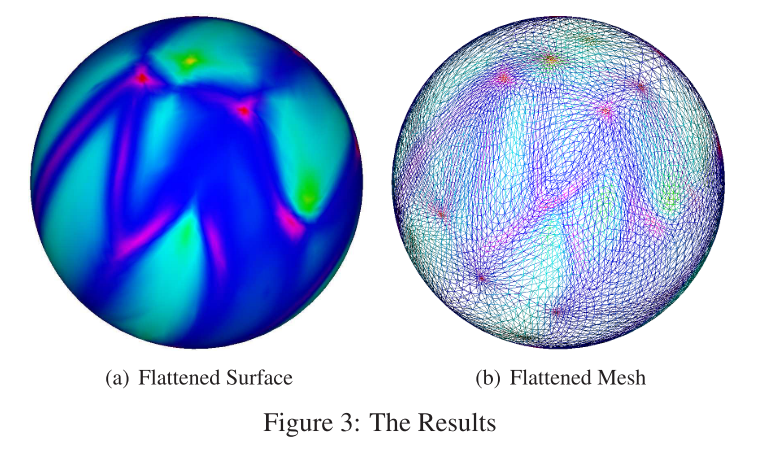
**3 The Filter Test**

Here we explain how the reader may reproduce our results. This section contains details about the test we include with the submission of this vtkPolyDataAlgorithm. We have provided *nice.vtk* which is a synthetic genus zero mesh. We set this as the first parameter to the vtkConformalFlatteningFilterTest and we set the output filename for the flattened mesh as the second parameter and the option “mapToSphere” as the third (represented by 1), as follows:

*D:\> VtkConformalFlatteningMesh nice.vtk niceFlat.vtk 1*

In Figure 2, we show the original *nice.vtk* data, which is colored according to mean curvature. In Figure 2(a), we show the surface view and in Figure 2(b), we show the triangulated mesh.

In Figure 3, we show the conformally flattened result of the filter. The coloring provides a convenient visual mapping from the original mesh to the sphere. In Figure 3(a), we show the surface view of the sphere and in Figure 3(b), we show the triangulated mesh on the sphere.



In order to reproduce our results, the reader should compile and run our code with the following packages

*CMake 3.14.3*

*ITK 4.13.1*

*VTK 8.1.2*

We have also included the results presented here as the *niceFlat.vtk* file included in this submission. One can verify the angle preserving nature of this filter by comparing the relative angle proportions of angles emanating from each vertex in the mesh.

**4 Conclusions**

In this paper, we have described the Visualization Toolkit (VTK) Conformal Flattening filter: ConformalFlatteningFilter. This VTK Polydata Algorithm is an implementation of a paper by Sigurd Angenent, et al., “On the LaplaceBeltrami Operator and Brain Surface Flattening” [1]. This filter performs an angle preserving map of any genus zero (i.e. no handles) triangulated mesh to the sphere or, alternatively, to the plane.

We have provided the user with details to be able to reproduce our results. We have also given suggestions as to how to exploit the full functionality of this filter.

**Acknowledgements**

We would like to thank the support from the National Natural Science Foundation of China No. 61601302 and Shenzhen peacock plan (NO. KQTD2016053112051497).

**References**

[1] S. Angenent, S. Haker, A. Tannenbaum, and R. Kikinis. On the Laplace-Beltrami operator and brain surface flattening. Medical Imaging, IEEE Transactions on, 18(8):700–711, 1999. (document), 1.1, 5

[2] G. J. Carman, H. A. Drury, and D. C. Van Essen. Computational methods for reconstructing and unfolding the cerebral cortex. Cereb. Cortex, 5(6):506C517, 1995. (document)

[3] C. Davatzikos and R. N. Bryan. Using a deformable surface model to obtain a shape representation of the cortex. Medical Imaging, IEEE Transactions on, 15(6):785C795, 1996. (document)

[4] D. MacDonald, D. D. Avis, and A. C. Evans. Multiple surface identification and matching in magnetic resonance images. In R. Robb, editor, Visualization Biomed. Comput., page 160C169. New York: SPIE, 1994. (document)

[5] E. Schwartz, A. Shaw, and E. Wolfson. A numerical solution to the generalized mapmakers problem: Flattening nonconvex polyhedral surfaces. Pattern Anal. Machine Intell., IEEE Transactions on, 11:1005C1008, 1989. (document)