

2010

Topological visualization of tensor fields using a generalized Helmholtz decomposition

Lierong Zhu
West Virginia University

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

Recommended Citation

Zhu, Lierong, "Topological visualization of tensor fields using a generalized Helmholtz decomposition" (2010). *Graduate Theses, Dissertations, and Problem Reports*. 2988.
<https://researchrepository.wvu.edu/etd/2988>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

TOPOLOGICAL VISUALIZATION OF TENSOR FIELDS USING A GENERALIZED HELMHOLTZ DECOMPOSITION

Lierong Zhu

**Thesis submitted to the
College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the Degree of**

**Master of Science
in
Electrical Engineering**

**Committee Members
Tim McGraw, Ph.D., Chair
Arun A. Ross, Ph.D.
Xin Li, Ph.D.**

Lane Department of Computer Science and Electrical Engineering

**Morgantown, West Virginia
2010**

**Keywords: Visualization, Vector, Diffusion Tensor, PDE, Topology, Helmholtz
Decomposition, Curl, Gradient, Divergent**

ABSTRACT

Topological Visualization of Tensor Fields Using a Generalized Helmholtz Decomposition

Lierong Zhu

Analysis and visualization of fluid flow datasets has become increasingly important with the development of computer graphics. Even though many direct visualization methods have been applied in the tensor fields, those methods may result in much visual clutter. The Helmholtz decomposition has been widely used to analyze and visualize the vector fields, and it is also a useful application in the topological analysis of vector fields. However, there has been no previous work employing the Helmholtz decomposition of tensor fields. We present a method for computing the Helmholtz decomposition of tensor fields of arbitrary order and demonstrate its application. The Helmholtz decomposition can split a tensor field into divergence-free and curl-free parts. The curl-free part is irrotational, and it is useful to isolate the local maxima and minima of divergence (foci of sources and sinks) in the tensor field without interference from curl-based features. And divergence-free part is solenoidal, and it is useful to isolate centers of vortices in the tensor field. Topological visualization using this decomposition can classify critical points of two-dimensional tensor fields and critical lines of 3D tensor fields. Compared with several other methods, this approach is not dependent on computing eigenvectors, tensor invariants, or hyperstreamlines, but it can be computed by solving a sparse linear system of equations based on finite difference approximation operators. Our approach is an indirect visualization method, unlike the direct visualization which may result in the visual clutter. The topological analysis approach also generates a single separating contour to roughly partition the tensor field into irrotational and solenoidal regions. Our approach will make use of the 2nd order and the 4th order tensor fields. This approach can provide a concise representation of the global structure of the field, and provide intuitive and useful information about the structure of tensor fields. However, this method does not extract the exact locations of critical points and lines.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr. Tim McGraw, who has offered me great help in completing this thesis. My supervisor shared with me a lot of his valuable expertise and research. And throughout the duration of my research and thesis-writing, he provided encouragement, thoughtful suggestions, good teaching, and a lot of good ideas. Without his help, my research and thesis would have been impossible to be successfully completed.

Secondly, I also would like to express my appreciation for the instruction of other members of my committee, Dr. Xin Li and Dr. Arun A. Ross. I am truly thankful for their sound advice, valuable time enormous support, and constructive comments.

Last but not least, I would like to thank my colleagues Matthew Madden and Swetha Danda, my friends Yunfei Zheng, Congxia Dai, Takamitsu Kawai, Chuck Maggio, and my family for their abundant help and support,

TABLE OF CONTENTS

TITLE-----	i
ABSTRACT-----	ii
ACKNOWLEDGEMENTS-----	iii
TABLE OF CONTENTS-----	iv
LIST OF TABLES-----	vi
LIST OF FIGURES-----	vii
CHAPTER 1: INTRODUCE-----	1
1.1. Motivation-----	1
1.2. Thesis Outline-----	3
CHAPTER 2: VECTOR CALCULUS AND DIFFERENTIAL OPERATORS-----	5
2.1. The 2 nd Order Cartesian Tensors-----	6
2.2. Higher Order Cartesian Tensors-----	9
CHAPTER 3: VECTOR CALCULUS AND DIFFERENTIAL OPERATORS-----	14
3.1. Gradient of Scalar Fields-----	15
3.2. Vector Field Operator-----	15
3.2.1. Vector Divergence-----	16
3.2.2. Vector Curl-----	18
3.3. Tensor Field-----	20
3.3.1. Tensor Divergence-----	21
3.3.2. Tensor Curl-----	22
CHAPTER 4: THE HELMHOLTZ DECOMPOSITION OF VECTOR AND TENSOR FIELDS -----	24
4.1. The Helmholtz Decomposition of Vector Fields-----	24
4.2. The Helmholtz Decomposition for tensor field-----	26
CHAPTER 5: NUMERICAL IMPLEMENTATION-----	28
5.1. Taylor Series Expansion-----	28
5.2. Kronecker product-----	29
5.3. Center Difference-----	31

5.4. Gradient of the 2nd Order Tensor Field G -----	36
5.5. Curl of the 2nd Order Tensor Field C -----	37
5.6. Numerical Implementation of the Helmholtz Decomposition-----	39
5.7 Conjugate Gradient Algorithm-----	40
CHAPTER 6: TOPOLOGICAL VISUALIZATION-----	47
6.1 Critical Points and Vortices-----	47
6.2 Degenerate Points and Separatrices-----	49
6.3 Topological Segmentation-----	49
CHAPTER 7: EXPERIMENT RESULTS-----	51
7.1. Experiment 1: simple 2 nd and 4 th synthetic tensor fields-----	51
7.2. Experiment 2: a 2 nd order MRI tensor data of human being-----	56
7.3. Experiment 3: more different 2 nd and 4 th order synthetic tensor fields-----	58
7.4. Experiment 4: generating 100 2 nd and 4 th order random tensor fields-----	66
CHAPTER 8: CONCLUSIONS AND FUTURE WORK-----	70
8.1 Conclusions-----	70
8.2 Future Work-----	71
BIOGRAPHY-----	72

LIST OF TABLES

2.1. Distinct Elements of the Diffusion Tensor up to the 4th order, and their response multiplicities-----	13
5.1. The matrix size of G and C for the 2nd and the 4th order tensors-----	41
5.2. Pseudocode: the algorithm for the topological visualization using the Helmholtz decomposition -----	46
7.1. Comparison of the various 2nd order synthetic tensor fields-----	61
7.2. Comparison of the various 4th order synthetic tensor fields-----	65
7.3. The comparison of the mean, variance, and max of the 2nd order random tensor fields-----	67
7.4. The comparison of the men, variance, max of the 4th order random fields-----	68

LIST OF FIGURES

1.1. Result of 3D texture-based billboard visualization of a human brain MRI-----	2
2.1. Image indication of a 3D Diffusion tensor three principal axis directions-----	8
2.2. The diffusion tensor images resulting from various typical eigenvalues-----	8
3.1. (a) and (b) indicate the source and its expanding direction, (b) and (c) represent the sink and its shrinking direction-----	17
4.1. The results of Helmholtz decomposition of vector fields-----	26
5.1. A simple example of the comparison of Kronecker (X, Y) and Kronecker (Y, X) ----	30
5.2. (a) Plot of the relationship of the target neighbor pixels in one dimension. (x) is the target pixel. (b) Transform the vector to one column vector-----	31
5.3. The result of plotting Δ^x operator based on $m = 16$ -----	32
5.4. (a) Plot of the relationship of the target neighbor pixels in two dimension. (x) is the target pixel. (b) Transform the 2D matrix data to one column vector-----	32
5.5. The result of plotting Δ^x, Δ^y operators based on $m \times n = 4 \times 4$ -----	33
5.6. (a) Plot of the relationship of the target neighbor pixels in three dimensions. (x) is the target pixel. (b) Transform the 3D matrix data to one column vector-----	34
5.7. The result of plotting $\Delta^x, \Delta^y, \Delta^z$ operators based on $m \times n \times p = 4 \times 4 \times 4$ -----	35
5.8. Plot of the gradient of the 2 nd order tensor field, $m \times n \times p = 4 \times 4 \times 4$ -----	37
5.9. Plot of the Curl of the 2 nd order tensor field, $m \times n \times p = 4 \times 4 \times 4$ -----	38
5.10. The image of method for the conjugate gradients and gradient descent-----	42
6.1. Vortices, (a) is the source, (b) is the sink-----	48
7.1. (a) and (d) illustrate vortices while (b) and (c) illustrate a source and sink respectively, which are used to construct a synthetic 2 nd order tensor field-----	52
7.2. Helmholtz decomposition results for the 2 nd order synthetic tensor field $D = \text{grad}(\phi) + \text{curl}(\psi) + H$ -----	54
7.3. Helmholtz decomposition results for the 4 th order synthetic tensor field $D = \text{grad}(\phi) + \text{curl}(\psi) + H$ -----	55
7.4. Helmholtz decomposition results for the 2 nd order DT-MRI datasets of human brain-----	57

7.5. The left side images are various 2 nd order synthetic tensor fields, and the right side images are the result of the topological visualization of left side tensor data-----	60
7.6. The left side images are various 4 th order synthetic tensor fields, and the right side images are the result of the topological visualization of left side tensor data-----	64
7.7. (a) is a synthetic 2 nd order random tensor field, (b) is the topological visualization of the left side 2nd order random tensor field-----	67
7.8. (a) is a synthetic 4 th order random tensor field, (b) is the topological visualization of left side 4 th order random tensor field-----	68
7.9. The comparison of time consuming for different size and order tensor fields-----	69

CHAPTER 1: INTRODUCTION

1.1. Motivation

Analysis and visualization of fluid flow datasets has become increasingly important with the development of computer graphics. Many scientists are interested in performing the research and make great contribution in visualization of tensor fields. There are many different methods to visualize the tensor fields. Even though Volume rendering [3], particle tracing, tensor probe and similar illuminated field lines methods have proved to be the most suitable approaches for the flow field visualization, those traditional visualization approaches often provide a rather coarse spatial resolution. The fast 2D texture-based billboard visualization by McGraw et al. [2] is combined with the current glyph, particle and texture-based visualization techniques. The resulting image is blended textured billboards, which are rendered at the center of each voxel in the dataset, and rendering polygons always face the viewer. This approach has very few parameters, and the speed of the billboard method makes parameter turning much faster since the image can be recomputed in real-time. Furthermore, the texture-based approach can flexibly map diffusion tensor field properties to visual characteristics such as color, texture orientation, texture frequency, tensor animation, and other characteristics. It is effective and suitable for a large spectrum of applications. My previous work involved applying this method into three dimensions. The result is presented in Figure (1.1).

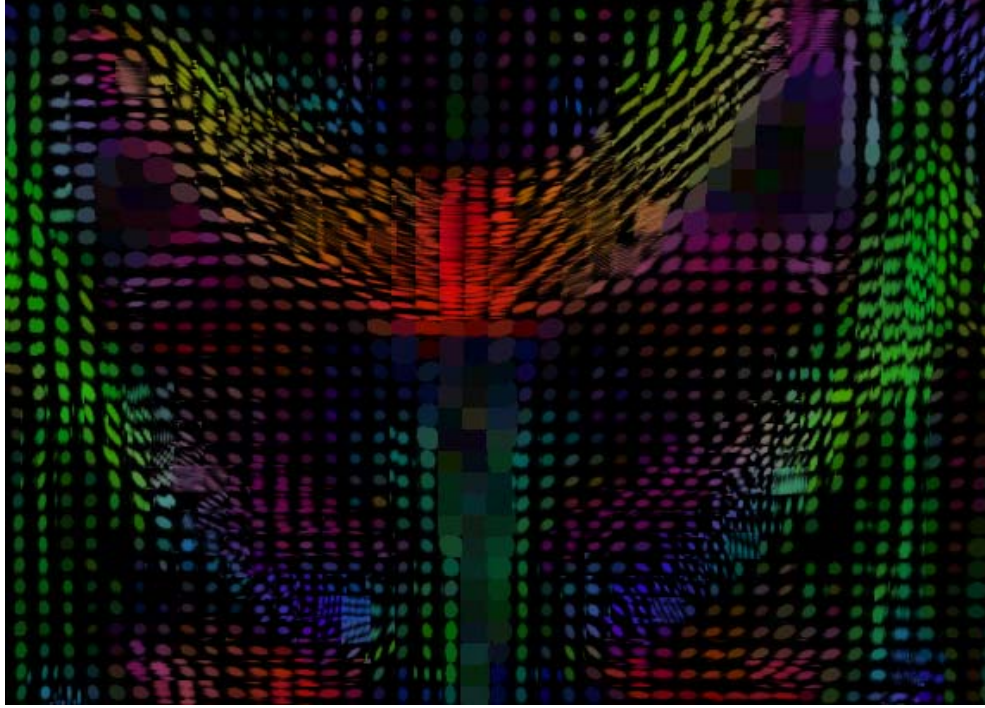


Fig. 1.1. Result of 3D texture-based billboarding visualization of a human brain MRI

The results obtained using the billboarding approach show some visual artifacts between voxels. And the visualization approaches mentioned are all direct visualizations of flow fluid datasets. However, all direct visualization approaches of the flow fluid datasets may still include the visual clutter if the original datasets are cluttered, which makes it difficult to analyze the tensor fields correctly. How to effectively visualize the flow fluid dataset still presents a great challenge.

The Helmholtz decomposition has been widely used to resolve many problems of fluid mechanics and electromagnetism, and it has also been applied in the topological analysis of vector fields. Polthier and Preuss [34, 35] use a discrete Helmholtz decomposition to robustly locate singularities in vector fields. Because they are dealing with vector fields defined on irregular meshes they must carefully define divergence and curl operators. Tong et al. [29] describe vector fields in a multiscale framework by defining a vector field scale space in terms of the separate scale spaces of the solenoidal and irrotational parts of the field. The vector field is then smoothed by separately smoothing these constituent fields. It can be seen that this approach better preserves singularities than

smoothing the input field directly. They also enhance features of the field by separately amplifying the components of the decomposed field. Li et al. [30] used the Helmholtz decomposition to segment 2D discrete vector fields, and use Green's function method to compute the decomposition. Then for each critical point they find the region of influence using graph-cuts. In their work the critical points are defined in terms of scalar stream and potential functions. Even though the Helmholtz decomposition has been proved useful in the analysis of the vector field, there has been no previous work applying this decomposition to tensor fields. Our approach will be to use finite difference approximations of the differential operators in regular Cartesian coordinates.

There is a long history of the topological analysis of surfaces and scalar fields. Applications include surface editing, segmentation, and shape matching. Reeb graphs [22] have been useful in describing the topology of surfaces in terms of a skeleton. Nodes of the Reeb graph represent critical points where isocontours of a scalar function are defined over the surface change topology. The contour tree [13, 26] is a special case of a Reeb graph - one which contains no cycles. It is often applied to images, or scalar fields of arbitrary dimension [18].

Our approach is based on the topological visualization and Helmholtz decomposition. The goal of this work is to develop a simple, but powerful representation of the complex phenomena described by the data.

1.2. Thesis Outline

Chapter 1 provides an overview of this thesis, the motivation to choose the topic of topological visualization of tensor fields using a generalized Helmholtz decomposition, the contribution of this thesis for diffusion tensor fields, and the thesis outline.

Chapter 2 introduces the mathematical equations and properties of the 2nd order Cartesian tensors and higher order Cartesian tensors.

Chapter 3 introduces several very important concepts which will be applied in the following chapters, including the properties of scalar field gradient, vector field divergence, vector field curl, tensor field divergence and tensor field curl.

Chapter 4 illustrates the properties of Helmholtz decomposition method and how the Helmholtz decomposition method works in the vector fields. Also, the Helmholtz decomposition is applied to the tensor fields extending from vector fields by employing the definitions of tensor field differential operators.

Chapter 5 introduces the numerical implementation, and thus illustrates how to realize the idea described in chapter 4 by using sparse matrices and Kronecker product to compute the curl-free and divergence-free components of a tensor field.

In Chapter 6, topological concepts, such as critical points, vortices, etc., will define the signed error function which will be used to segment tensor fields.

In Chapter 7 we demonstrate the results of experiments which apply the Helmholtz decomposition to synthetic tensor fields and diffusion tensor MRI of the human brain and the topological visualization of synthetic tensor fields based on signed error function.

Chapter 8 represents the conclusion of the thesis and recommendation for future work. Although this method appears to work very well, there is still room for a great deal of improvement to this method.

CHAPTER 2:

The 2nd ORDER AND HIGHER ORDER CARTESIAN TENSOR

Tensors have become a very important topic, since they have been introduced into mathematics, physics and the engineering field to extend the notion of scalars, vectors and matrices. Tensor fields, especially the 2nd order tensor fields, are useful in many medical, mechanical and physical applications such as: fluid dynamics, molecular dynamics, biology, geophysics, MRI and other applications. Tensors provide a natural and concise mathematical framework for formulating and solving problems in these areas of research.

The order of the tensor is the number of numerical indices which can specify an individual component of a tensor. If the tensor order is l and d represents the dimension of the tensor, then the tensor has l indices and d^l components. Each index can take one of d different values. In 2 dimensions, an l th order tensor then has 2^l components. Similarly in 3 dimensions, if the order of the tensor is 2, the tensor has 9 components, and if the order of the tensor is 4, then the tensor has 81 components.

Thus the 0th order tensor is a scalar, which is a single component. Tensors of the 1st order are represented as vectors, and tensors of the 2nd order are represented as matrices and so on. Tensors are generalizations of scalars (that have no indices), vectors (that have exactly one index), and matrices (that have two or more indices) to an arbitrary number of indices. The notation of a tensor is similar to a matrix and has an arbitrary number of indices. A general tensor may also have two different types of indices, covariant denoted by subscripts and contravariant denoted by superscripts. The distinction between covariant and contravariant indices must be made for general tensors. The two indices are equivalent for tensors in three-dimensional Euclidean space. Such tensors are known as Cartesian tensors. Therefore, indices will be denoted using only subscripts.

2.1. The 2nd Order Cartesian Tensors

Diffusion tensor MRI is a source of tensor field data which is common in medical imaging applications. Diffusion tensor imaging scans comprise at least six gradient directions. A diffusion tensor MRI is measured via the magnetic field variations in the MRI magnet. It is sufficient to calculate, for each voxel, a diffusion tensor D . The 2nd order tensor in 3 dimensions will have 9 components represented by a 3×3 positive symmetric matrix given by,

$$D = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yz} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{bmatrix} \quad (2.1)$$

The diagonal elements D_{xx} , D_{yy} and D_{zz} represent the apparent diffusion coefficient (ADC) along the x , y and z axes in the laboratory frame. Furthermore, the tensor is symmetric, so $D^T = D$.

The MRI intensity, S , at each voxel depends on the properties of the diffusion-encoding gradient, b , and the apparent diffusion tensor, D , at that location. The Stejskal-Tanner formula indicates their relationship is given as:

$$S_0 \exp \left(- \sum_{i=1}^3 \sum_{j=1}^3 b_{ij} D_{ij} \right) \quad (2.2)$$

In Equation (2.2), S_0 is the signal intensity without the diffusion weighting, S is the signal with the gradient, b is a matrix characterizing the gradient pulse sequence, and D is the diffusion constant.

Equation (2.2) can be expanded to yield the following equation:

$$S = S_0 \exp(-b_{xx}D_{xx} - b_{yy}D_{yy} - b_{zz}D_{zz} - 2b_{xy}D_{xy} - 2b_{yz}D_{yz} - 2b_{xz}D_{xz}) \quad (2.3)$$

Then we can log-linearize the Equation (2.3) to obtain the following equation,

$$\ln S = (-b_{xx}D_{xx} - b_{yy}D_{yy} - b_{zz}D_{zz} - 2b_{xy}D_{xy} - 2b_{yz}D_{yz} - 2b_{xz}D_{xz}) \ln S_0 \quad (2.4)$$

For each slice image, there are seven unknown parameters which represent the six independent components of symmetric diffusion tensor, D , and S_0 . According to the method of least squares linear regression, we acquire a diffusion-weighted image with 7 gradient directions which provide eight equations for S in each voxel in order to fully generate a diffusion tensor. The system of equations can be solved using the following Equation (2.5),

$$\begin{bmatrix} \ln S^1 \\ \vdots \\ \ln S^m \end{bmatrix} = \begin{bmatrix} 1 & -b_{xx}^1 & -b_{yy}^1 & -b_{zz}^1 & -b_{xy}^1 & -b_{yz}^1 & -b_{xz}^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & -b_{xx}^m & -b_{yy}^m & -b_{zz}^m & -b_{xy}^m & -b_{yz}^m & -b_{xz}^m \end{bmatrix} \begin{bmatrix} \ln S_0 \\ D_{xx} \\ D_{yy} \\ D_{zz} \\ D_{xy} \\ D_{yz} \\ D_{xz} \end{bmatrix} \quad (2.5)$$

The Equation (2.5) can compute each component of tensor. Linear algebra is used to generate eigenvalues and eigenvectors of this tensor D ,

$$D = U\Lambda U^T \quad (2.6)$$

In Equation (2.6), $\Lambda = \text{diag} [\lambda_1, \lambda_2, \lambda_3]$, $U = [e_1, e_2, e_3]$. Sorting the eigenvalues (λ_1, λ_2 and λ_3), let $\lambda_1 > \lambda_2 > \lambda_3 > 0$ be the eigenvalues of the symmetric tensor D , and let e_i be the normalized eigenvector corresponding to λ_i . Then D can be expressed as the following Equation (2.7),

$$D = [e_1 \quad e_2 \quad e_3] \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \quad (2.7)$$

Expanding the Equation (2.7), therefore, the tensor equation can also be expressed as the Equation (2.8),

$$D = \lambda_1 e_1 e_1' + \lambda_2 e_2 e_2' + \lambda_3 e_3 e_3' \quad (2.8)$$

The eigenvalues represent the magnitude of diffusion in the direction of their corresponding eigenvectors. One purpose of diffusion tensor MRI is to assess the

properties of white matter in the central nervous system. The fiber direction is indicated by the tensor's principal diffusion direction (PDD). A popular representation for describing anisotropic diffusion is the diffusion ellipsoid. This following ellipsoid image (Fig. 2.1.) is the unit sphere under the transformation defined by a tensor D .

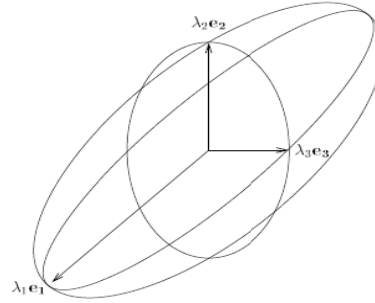


Fig. 2.1. Image indication of a 3D Diffusion tensor three principal axis directions

For anisotropic diffusion $\lambda_1 \gg \lambda_2 = \lambda_3$, the diffusion ellipsoid is linear.

For anisotropic diffusion $\lambda_1 = \lambda_2 \gg \lambda_3$, the diffusion ellipsoid is planar.

For isotropic diffusion $\lambda_1 = \lambda_2 = \lambda_3$, the diffusion ellipsoid is spherical.

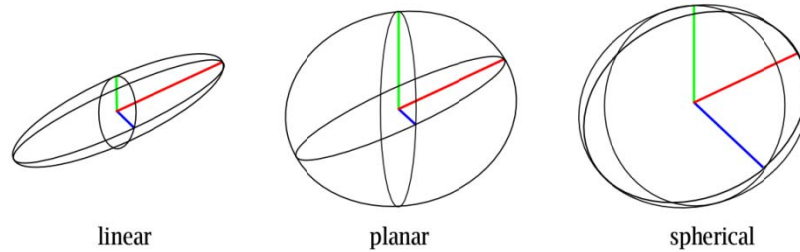


Fig. 2.2. The diffusion tensor images resulting from various typical eigenvalues

Fractional anisotropy (FA) is commonly used to quantify anisotropy, which is greatly influenced by the magnitude and orientation of white matter tracts of the brain. This results in significantly large normal variations of FA throughout the brain, in which the FA is high in areas of prominent white matter pathways, for example. Therefore, this property can be used to map the course of fiber tracts. However, one of the limitations of FA imaging is that it does not contain information about the directionality of anisotropy and it is therefore difficult to identify white-matter tracts only through the FA images

alone. Fortunately, color-coded FA images can composite images containing information about both magnitude and direction of diffusion anisotropy.

If full tensor data acquisition is applied, the fractional anisotropy can be calculated from the eigenvalues λ_1 , λ_2 and λ_3 of the diffusion tensor. The equation is given as,

$$FA = \frac{\sqrt{3}}{\sqrt{2}} \frac{\sqrt{(\lambda_1 - \lambda)^2 + (\lambda_2 - \lambda)^2 + (\lambda_3 - \lambda)^2}}{\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}} \quad (2.9)$$

where λ_1 is the mean of the eigenvalues ($\frac{\lambda_1 + \lambda_2 + \lambda_3}{3}$) and λ_1 , λ_2 and λ_3 are the eigenvalues of diffusion tensor D sorted in decreasing order.

FA takes on values between 0 (perfectly isotropic diffusion) and 1 (the hypothetical case of an infinite cylinder) and is thus directly comparable between subjects.

2.2. Higher Order Cartesian Tensors

Higher-order tensors have recently been used to describe diffusion in the context of medical imaging by Özarslan et al. [9, 10].

From the 2nd order tensors Equation (2.6), the diffusion coefficient can be assumed as,

$$D(g) = \sum_{i=1}^3 \sum_{j=1}^3 D_{ij} g_i g_j \quad (2.10)$$

If the order indices are permuted, the result is the Equation (2.11)

$$D(g) = \sum_{i=1}^3 \sum_{j=1}^3 D_{ij} g_i g_j = \sum_{i=1}^3 \sum_{j=1}^3 D_{ij} g_j g_i = \sum_{i=1}^3 \sum_{j=1}^3 D_{ji} g_i g_j = \sum_{i=1}^3 \sum_{j=1}^3 D_{ji} g_j g_i \quad (2.11)$$

For all vectors $g_i g_j = g_j g_i$, and the symmetric the 2nd order tensor becomes

$$D_{ij} = D_{ji} \quad (2.12)$$

The number of distinct components reduces from 9 to 6. Clearly, imposing a symmetry constraint on D can significantly reduce the number of distinct components and eliminate advective transport from the equation.

Similarly, for the l th order tensors, symmetry implies that components whose indices are permutations of each other are equal,

$$D_{i_1 i_2 \dots i_l} = D_{(i_1 i_2 \dots i_l)} \quad (2.13)$$

where $(i_1 i_2 \dots i_l)$ stands for all permutations of the l indices. For instance, if l is 4th order tensor,

$$D_{xxxx} = D_{xxyx} = D_{xyxx} = D_{yxxx} \quad (2.14)$$

The number of unique components of a tensor is independent of the order of indices, so the symmetry reduces the number of unique components from 3^l to

$$N_3^l = \binom{l+2}{2} = \frac{(l+2)(l+1)}{2} \quad (2.15)$$

where N_3^l is the number of unique components and l is the tensor order. For example, if $l = 4$, then the number of distinct components $N_3^4 = 15$, which is much lower than the number of total components $3^4 = 81$.

The previous section illustrated the Stejskal-Tanner formula for 2nd order tensor. The generalized Stejskal-Tanner formula for order l tensor is given by

$$S = S_0 \exp \left(-b \sum_{i_1=1}^3 \sum_{i_2=1}^3 \dots \sum_{i_j=1}^3 D_{i_1 i_2 \dots i_j} g_{i_1} g_{i_2} \dots g_{i_j} \right) \quad (2.16)$$

where g_{i_j} is one of the basic vectors. The Equation (2.16) can help us to calculate all the components of the diffusion tensor (DT) of general order by means of a simple multilinear regression. The diffusion coefficient along the gradient direction is given by

$$D(g) = \sum_{i_1=1}^3 \sum_{i_2=1}^3 \dots \sum_{i_3=1}^3 D_{i_1 i_2 \dots i_j} g_{i_1} g_{i_2} \dots g_{i_j} \quad (2.17)$$

the generalized Stejskal-Tanner formula for order l tensor will be simplified into the following Equation (2.18)

$$S = S_0 \exp(-bD(g)) \quad (2.18)$$

Given g , a negative value, the diffusion coefficient will have the following Equation (2.19),

$$D(-g) = \begin{cases} -D(g), & l \text{ is odd} \\ D(g), & l \text{ is even} \end{cases} \quad (2.19)$$

However, the diffusion coefficient has a physical meaning, which implies that it can not have a negative value. Therefore, the l is forced to be an even number ($l=2, 4, \dots$). The result is the antipodal symmetrical diffusivity, given as

$$D(-g) = D(g), l \text{ is even} \quad (2.20)$$

The Equation (2.17) shows us all the tensor components. Therefore the distinct components of the l th order DT can be extracted, which provides the number of times a given element is repeated. The number of permutations of each set of indices is equal to the number of times that component is duplicated. The repetition number, μ , of the component is called the multiplicity, and is given by

$$\mu = \binom{l}{n_x} \binom{l-n_x}{n_y} = \frac{l!}{n_x! n_y! n_z!} \quad (2.21)$$

where l is the order of tensors, n_x , n_y and n_z are individually the number of the x , y , and z indices included in the full sequence of subscripts defining the component of the tensors, and $l = n_x + n_y + n_z$. Therefore, the general Stejskal-Tanner formula can be rewritten as,

$$S = S_0 \left(-b \sum_{k=1}^{N_3^l} \mu_k D_k \prod_{\mu=1}^l g_{k(p)} \right) \quad (2.22)$$

Where D_k is the k^{th} unique element of the tensor, $g_{k(p)}$ is the component of the gradient direction specified by the p^{th} index of the k^{th} unique element of the generalized tensor D , and μ_k is the multiplicity of the component. Table (2.1) lists the unique components of a generalized DT up to the 4th order, along with their response multiplicities.

When writing an expression containing tensors, the Einstein summation convention will be used. This means that repeated indices are to be multiplied pairwise, and summed over all possible values,

$$A_{i_1 i_2 \dots i_l} B_{i_1 i_2 \dots i_l} = \sum_{i_1=1}^d \sum_{i_2=1}^d \dots \sum_{i_l=1}^d A_{i_1 i_2 \dots i_l} B_{i_1 i_2 \dots i_l} \quad (2.23)$$

where A and B are two same size tensors. If both tensors are symmetric, the number of terms in the summation can be greatly reduced by simply summing over the unique components

$$A_{i_1 i_2 \dots i_l} B_{i_1 i_2 \dots i_l} = \sum_{k=1}^{N^l} \mu_k A_k B_k \quad (2.24)$$

where A_k and B_k are the k^{th} independent components of A and B respectively, μ_k is the multiplicity of that component, N^l is the number of unique components without repeating, and l is the order of tensor A or B .

0^{th} Order 0		4^{th} Order			
$\mu = 1$	D	$\mu = 1$	D_{xxxx}	$\mu = 6$	D_{xxyy}
			D_{yyyy}		D_{yyzz}
2^{nd} Order			D_{zzzz}		D_{zzxx}
$\mu = 1$	D_{xx} D_{yy} D_{zz}	$\mu = 4$	D_{xxxy}	$\mu = 12$	D_{xxyz}
			D_{xxxz}		D_{yyxz}
			D_{yyyx}		D_{zzxy}
$\mu = 2$	D_{xy}		D_{yyyz}		
	D_{yz}		D_{zzzx}		
	D_{xz}	D_{zzzy}			

Table 2.1. Distinct Elements of the Diffusion Tensor up to the 4th order, and their response multiplicities

CHAPTER 3:

VECTOR CALCULUS AND DIFFERENTIAL OPERATORS

This chapter introduces the basics of vector calculus and the differential operators, which will be used in the subsequent sections. A vector can be thought of as a directed arrow with magnitude and direction in Cartesian space. A vector field is a vector, each of whose components is a scalar field. We first introduce one of most important and useful mathematical constructs called the ‘*del* operator’, *del* is a vector differential operator, usually denoted by the nabla symbol, ∇ . The gradient of a scalar field, and the divergence and curl of vector field will be denoted by the *del*. *Del* is defined as components of a vector which are partial differentiations with respect to the three principle directions of a Cartesian coordinate system.

Assuming the unit vectors in the x, y, z directions are denoted by i, j and k respectively, the *del* operator in the three-dimensional Cartesian coordinate system is given as

$$\nabla = \frac{\partial}{\partial x}i + \frac{\partial}{\partial y}j + \frac{\partial}{\partial z}k \quad (3.1)$$

This can also be expressed as

$$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right) \quad (3.2)$$

Or

$$\nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} \quad (3.3)$$

3.1. Gradient of Scalar Fields

At any point of a scalar field, the gradient of the scalar field is a vector field which points in the direction of the greatest rate of change, and whose magnitude is the greatest rate of change.

For a scalar field $f(x, y, z)$ in Cartesian space, the gradient is given as

$$\text{grad}(f) = \nabla f = \frac{\partial f}{\partial x} i + \frac{\partial f}{\partial y} j + \frac{\partial f}{\partial z} k = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right) \quad (3.4)$$

where each of the partial derivatives is evaluated at the point (x, y, z) . The gradient of the scalar field becomes a vector field. The gradient of a scalar field is simply the product of a scalar field such as $f(x, y, z)$ by the *del* operator. Here, $f(x, y, z)$ can be referred to as the potential field.

3.2. Vector Field Operator

Besides the usual operations of vector-vector addition and scalar-vector multiplication, there are two types of multiplication of vectors by other vectors. These include the dot product and cross product.

The dot product of vector field is the sum of the products of corresponding parts of vector components. The result of the dot product of two vectors is a scalar. In Cartesian space, the dot product of two vectors $u = (u_1, u_2, u_3)$ and $v = (v_1, v_2, v_3)$ is defined as:

$$u \cdot v = u_1 v_1 + u_2 v_2 + u_3 v_3 \quad (3.5)$$

Also the dot product can be obtained via transposition and matrix multiplication,

$$u \cdot v = u^T v \quad (3.6)$$

The cross product results in another vector which is perpendicular to the plane containing the two input vectors. This is also referred as the vector product because it yields a new vector.

$$u \times v = \begin{vmatrix} i & j & k \\ u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \end{vmatrix} = (u_2 v_3 - u_3 v_2)i - (u_1 v_3 - u_3 v_1)j + (u_1 v_2 - u_2 v_1)k \quad (3.7)$$

where i, j and k are the basis of unit vectors. Here, $u \times v$ is always perpendicular to both u and v , with the orientation is determined by the right-hand rule. The cross product of two vectors contains more valuable information about the two vectors themselves.

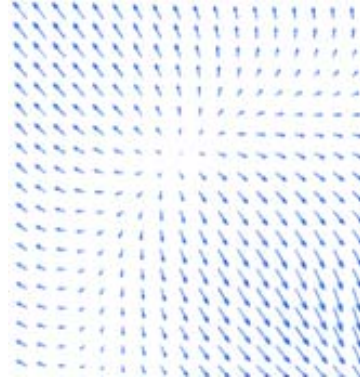
If we multiply a vector field by the *del* operator, we will get two different multiplication results. One is called the divergence of a vector field, and other is called the curl of vector field. Divergence and curl will be introduced in detail in the following two sections.

3.2.1. Vector Divergence

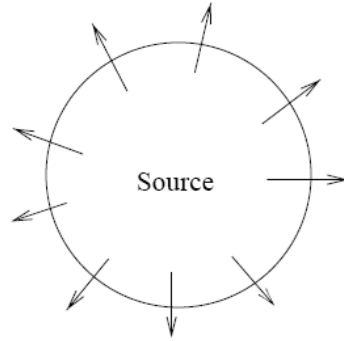
The physical meaning of divergence is flux density, which reflects the rate of flux expansion and contraction at a point. We can define divergence as the rate of increase or decrease in the direction of the vector at a point of a vector field. The divergence of a vector field is the dot product of the *del* operator and the vector field. Therefore, the divergence of a vector field is a scalar. Given a vector field $v(x, y, z) = (v_x i + v_y j + v_z k)$, the divergence of a vector field v is given as

$$\begin{aligned} \text{div}(v) &= \nabla \cdot v = \left(\frac{\partial}{\partial x} i + \frac{\partial}{\partial y} j + \frac{\partial}{\partial z} k \right) \cdot (v_x i + v_y j + v_z k) \\ &= \left(\frac{\partial}{\partial x} \right) v_x + \left(\frac{\partial}{\partial y} \right) v_y + \left(\frac{\partial}{\partial z} \right) v_z \\ &= \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} \end{aligned} \quad (3.8)$$

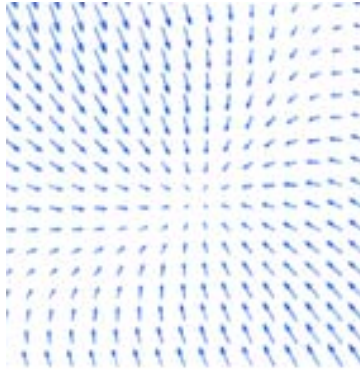
The divergence of the vector field becomes a scalar field by applying the dot product of vector operator ∇ to the vector field. At a given point, the divergence of a vector field is a single number that represents how much the flow is expanding at that point.



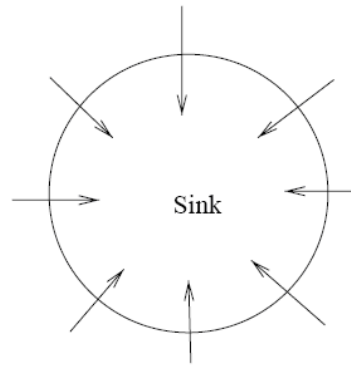
(a) $\text{div}(v) > 0$, expanding



(b) source



(c) $\text{div}(v) < 0$, shrinking



(d) sink

Fig 3.1. (a) and (b) indicate the source and its expanding direction, (b) and (c) represent the sink and its shrinking direction

A positive value of $\nabla \cdot v$ at a point means that the integral lines of the vector field v diverge there. We can also say that v has a source there, whose volume density is $\nabla \cdot v$.

A negative value of $\nabla \cdot v$ at a point means that the integral lines of the vector field v converge there. We say that v has a sink, whose volume density is $\nabla \cdot v$.

If $\nabla \cdot v = 0$, v has neither a source nor a sink. A vector field v , where $\nabla \cdot v = 0$ at all points is referred to as solenoidal. A solenoidal field is also sometimes called incompressible.

We use the velocity field to illustrate the divergence of vector field. The divergence of a vector field is relatively easy to understand intuitively. Imagine that the vector field v (Figure (3.1.a)) provides the velocity of a fluid flow. It appears that the fluid is expanding outward from the origin. Its expansion of fluid flowing with velocity field v is captured by the divergence of v , which we denote $div(v)$ ($\nabla \cdot v$). The divergence of the above vector field is positive ($\nabla \cdot v > 0$) since the flow is expanding. In contrast, the vector field (Figure (3.1.c)) represents fluid flowing so that it compresses as it moves toward the origin. Since this compression of fluid is the opposite of expansion, the divergence of this vector field is negative ($\nabla \cdot v < 0$).

Divergence is distributive, and therefore satisfies $\nabla \cdot (v_1 + v_2) = \nabla \cdot v_1 + \nabla \cdot v_2$. If v is an incompressible fluid, it satisfies $\nabla \cdot v = 0$.

3.2.2. Vector Curl

Curl (also called *rotor* or *rotation*) is defined as the rate of increase or decrease perpendicular to the vector field, which reflects the rotational tendency at a point of a vector field. The curl of a vector field is the cross product of the *del* operator and a vector field, and therefore the curl of a vector field is still a vector field. Given a vector field $v(x, y, z) = (v_x i + v_y j + v_z k)$, the curl of a vector field is given as

$$\begin{aligned}
rl(v) = \nabla \times v &= \begin{vmatrix} i & j & k \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ v_x & v_y & v_z \end{vmatrix} = \left(\frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z} \right) i - \left(\frac{\partial v_z}{\partial x} - \frac{\partial v_x}{\partial z} \right) j + \left(\frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y} \right) k \\
&= \begin{vmatrix} 0 & -\frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & -\frac{\partial}{\partial x} \\ -\frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{vmatrix} v
\end{aligned} \tag{3.9}$$

where v is the vector field, v_x, v_y, v_z are functions of the variables x, y and z , and i, j and k are the unit basis vectors. In fluid mechanics, the curl of the fluid velocity field is called the rotation because it measures the field's degree of rotation around a given point.

Based on the above definitions of divergence and curl definition, we can analyze the properties of the divergence and curl of a vector field.

$$\begin{aligned}
(\nabla \times v) \cdot \nabla &= \nabla \cdot \left(\left(\frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z} \right) i - \left(\frac{\partial v_z}{\partial x} - \frac{\partial v_x}{\partial z} \right) j + \left(\frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y} \right) k \right) \\
&= \frac{\partial}{\partial x} \left(\frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z} \right) - \frac{\partial}{\partial y} \left(\frac{\partial v_z}{\partial x} - \frac{\partial v_x}{\partial z} \right) + \frac{\partial}{\partial z} \left(\frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y} \right) = 0
\end{aligned} \tag{3.10}$$

The divergence of the curl of a vector field is always zero. Therefore, the curl of a vector field is described as divergent-free or solenoidal.

Similarly,

$$\begin{aligned}
\nabla \times (\nabla v) &= \begin{vmatrix} i & j & k \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \end{vmatrix} = \left(\frac{\partial^2 v}{\partial y \partial z} - \frac{\partial^2 v}{\partial z \partial y} \right) - \left(\frac{\partial^2 v}{\partial z \partial x} - \frac{\partial^2 v}{\partial x \partial z} \right) + \left(\frac{\partial^2 v}{\partial x \partial y} - \frac{\partial^2 v}{\partial y \partial x} \right) \\
&= 0
\end{aligned} \tag{3.11}$$

The curl of the gradient of a scalar field is always zero. The gradient field is described as irrotational, or curl-free.

3.3. Tensor Field

The properties of a tensor have been described in Chapter2. A tensor field, a concept which has been applied in mathematics, physics and engineering, is defined as a Cartesian space, each point of which has a tensor value. Tensor divergence and tensor curl will be related through discrete partial derivatives, which are denoted by the symbol ∂ ,

$$\partial_i = \frac{\partial}{\partial i} \quad (3.12)$$

Therefore, tensor equations such as divergence and curl can be expressed in a simple and compact manner based on the above equation.

Using Equation (3.4), the gradient tensor for the 2nd order is simply the product of the *del* operator and tensor, which is given as

$$\begin{aligned} grad(D^2) = \nabla D^2 &= \left(\frac{\partial}{\partial x} i + \frac{\partial}{\partial y} j + \frac{\partial}{\partial z} k \right) (D_x i + D_y j + D_z k) = \begin{bmatrix} \frac{\partial D_x}{\partial x} & \frac{\partial D_y}{\partial x} & \frac{\partial D_z}{\partial x} \\ \frac{\partial D_x}{\partial y} & \frac{\partial D_y}{\partial y} & \frac{\partial D_z}{\partial y} \\ \frac{\partial D_x}{\partial z} & \frac{\partial D_y}{\partial z} & \frac{\partial D_z}{\partial z} \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial D_{xx}}{\partial x} & \frac{\partial D_{xy}}{\partial x} & \frac{\partial D_{xz}}{\partial x} \\ \frac{\partial D_{yx}}{\partial y} & \frac{\partial D_{yy}}{\partial y} & \frac{\partial D_{yz}}{\partial y} \\ \frac{\partial D_{zx}}{\partial z} & \frac{\partial D_{zy}}{\partial z} & \frac{\partial D_{zz}}{\partial z} \end{bmatrix} \end{aligned} \quad (3.13)$$

or

$$grad(D^2) = \partial_i D_{jk} \quad (3.14)$$

where superscripts denote tensor order. Similarly, the gradient tensor for the 4th order can be expressed as,

$$grad(D^4) = \partial_i D_{jklm} \quad (3.15)$$

where the notation indicates that for all possible values of index i, the tensor components are differentiated with respect to that index.

3.3.1. Tensor Divergence

In general, the divergence of an order n tensor field is an order (n-1) tensor field. If the order of tensor field is 2,

$$(\nabla \cdot D)_\alpha = \nabla_\beta D_{\beta\alpha} = \frac{\partial D_{x\alpha}}{\partial x} + \frac{\partial D_{y\alpha}}{\partial y} + \frac{\partial D_{z\alpha}}{\partial z} \quad (3.16)$$

Therefore, the divergence of the 2nd tensor field is given as,

$$div(D) = \nabla \cdot \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{bmatrix} = \begin{bmatrix} \frac{\partial D_{xx}}{\partial x} + \frac{\partial D_{xy}}{\partial y} + \frac{\partial D_{xz}}{\partial z} \\ \frac{\partial D_{yx}}{\partial x} + \frac{\partial D_{yy}}{\partial y} + \frac{\partial D_{yz}}{\partial z} \\ \frac{\partial D_{zx}}{\partial x} + \frac{\partial D_{zy}}{\partial y} + \frac{\partial D_{zz}}{\partial z} \end{bmatrix} \quad (3.17)$$

Therefore, the divergence of the tensor field can be expressed in Einstein notation as,

$$div(D^2) = \partial_i D_{ij} \quad (3.18)$$

Similarly, if the order of tensor field is 4, the divergence of the tensor field can be expressed in Einstein notation as,

$$div(D^4) = \partial_i D_{ijkl} \quad (3.19)$$

Where the notation indicates that for all possible values of index i , the tensor components are differentiated with respect to that index and summed over. If the field consists of totally symmetric tensors, the divergence tensor will also be completely symmetric.

If we apply the product rule for differentiation, the divergence for Dv will be,

$$\text{div}(Dv) = \partial_i(D_{ij}v_j) = v_j\partial_i D_{ij} + D_{ij}v_j \partial_i \quad (3.20)$$

From the Equation (3.20), we find the divergence of Dv is split into two parts. The first term $v_j\partial_i D_{ij}$ relies on the divergence of $D(\partial_i D_{ij})$. The second component $D_{ij}\partial_i v_j$ depends on the gradient of v ($\partial_i v_j$). The divergence of D contains all of the information regarding how spatially varying D influences the $\text{div}(Dv)$.

The divergence often appears in conservation laws such as Fick's second law,

$$\text{div}(D\nabla c) = \text{div}(D)\nabla c + D\nabla^2 c = 0 \quad (3.21)$$

which is a statement of conservation of mass for the diffusion process governed by the concentration gradient and diffusion tensor field D . Expanding $\text{div}(D\nabla c)$ using the product rule, it becomes apparent that if the concentration gradient is nonzero and constant, then mass can only be conserved if the divergence of the tensor field is zero.

3.3.2. Tensor Curl

Similarly, the curl of a tensor field can be derived from the product rule. The curl of a tensor field is a rotation operator. The magnitude of each point of the tensor field is the rotation magnitude, and the direction of curl is perpendicular to the rotation plane at each point. The curl of tensor field is denoted $\text{curl } \psi$ or $\nabla \times \psi$.

$$\text{curl}(D) = \nabla \times D = \begin{bmatrix} \frac{\partial D_{xz}}{\partial y} - \frac{\partial D_{xy}}{\partial z} & \frac{\partial D_{yz}}{\partial y} - \frac{\partial D_{yy}}{\partial z} & \frac{\partial D_{zz}}{\partial y} - \frac{\partial D_{yz}}{\partial z} \\ \frac{\partial D_{xx}}{\partial z} - \frac{\partial D_{xz}}{\partial x} & \frac{\partial D_{xy}}{\partial z} - \frac{\partial D_{yz}}{\partial x} & \frac{\partial D_{xz}}{\partial z} - \frac{\partial D_{zz}}{\partial x} \\ \frac{\partial D_{xy}}{\partial x} - \frac{\partial D_{xx}}{\partial y} & \frac{\partial D_{yy}}{\partial x} - \frac{\partial D_{xy}}{\partial y} & \frac{\partial D_{yz}}{\partial x} - \frac{\partial D_{xz}}{\partial y} \end{bmatrix} \quad (3.22)$$

The curl of the 2nd order case in 3 dimensions is defined as

$$\text{curl}(D) = \varepsilon_{ijk}(\partial_j D_{kl}) \quad (3.23)$$

Also the curl of the 4th order case in 3 dimensions is defined as

$$\text{curl}(D) = \varepsilon_{ijk}(\partial_j D_{klmn}) \quad (3.24)$$

where ε_{ijk} is the permutation tensor (Levi-Civita symbol),

$$\varepsilon_{ijk} = \begin{cases} +1, & (i, j, k) \text{ is an even permutation of indices} \\ -1, & (i, j, k) \text{ is an odd permutation of indices} \\ 0, & \text{otherwise} \end{cases} \quad (3.25)$$

The permutation tensor is often used to define the vector cross product $u \times v = \varepsilon_{ijk} u_j v_k$.

Therefore, the curl is usually used to characterize vortices and shear in flows.

CHAPTER 4:

THE HELMHOLTZ DECOMPOSITION OF VECTOR AND TENSOR FIELDS

4.1. The Helmholtz Decomposition of Vector Fields

A general vector field may contain features such as sources and sinks which are characterized by divergence. They may also contain other features, such as vortices, which are characterized by the curl. If we can separate the field into two components, in which one part contains the sources and sinks, and the other part contains the vortices we can simplify the processing and visualization of the field.

Fortunately, in the mathematical theory of fluid dynamics, a method exists which can separate the divergence-free and curl-free components. This method is the well known Helmholtz decomposition (also referred to as the Helmholtz-Hodge decomposition). Hermann von Helmholtz found that any sufficiently smooth, rapidly decaying vector field in 3 dimensions can be decomposed into three components, including a curl-free vector field, a divergence-free vector field, and a harmonic vector field. The equation representing this is given as,

$$v = \nabla\phi + \nabla \times \psi + h \quad (4.1)$$

where v is a vector, $\nabla\phi$ is the gradient of a scalar potential field ϕ , $\nabla \times \psi$ is the curl of a vector stream field ψ , and h represents a harmonic vector field.

Chapter 3 describes the characteristics of gradient, divergence, and curl of vector fields. The properties of these concepts can be applied in Equation (4.1). The gradient of a scalar potential field ϕ is curl free, and therefore we have

$$\text{curl}(\nabla\phi) = \nabla \times \nabla\phi = 0 \quad (4.2)$$

Similarly, the curl of a vector stream field ψ is divergent free, and therefore we have

$$\text{div}(\nabla \times \psi) = \nabla \cdot (\nabla \times \psi) = 0 \quad (4.3)$$

If a vector field possesses both solenoidal and potential features, it is said to be a Laplacian field, which is common in vector analysis. The symbol of the Laplacian operator is $\nabla \cdot \nabla$, also written as Δ . The Laplacian of a scalar field is given as,

$$v = \nabla \cdot \nabla v = \frac{\partial^2 v_x}{\partial x^2} + \frac{\partial^2 v_y}{\partial y^2} + \frac{\partial^2 v_z}{\partial z^2} \quad (4.4)$$

The Laplacian field satisfies the conditions $\text{curl}(v)=0$ and $\text{div}(v)=0$.

Harmonic vector field h , is both solenoidal and irrotational, so the harmonic vector field also has the following properties,

$$\text{curl}(h) = \text{div}(h) = 0 \quad (4.5)$$

$\nabla\phi$ is called the curl-free or divergence component, $\nabla \times \psi$ is the divergence-free part or rotation component, and the remaining h is the harmonic component. Also ϕ can be referred to as the rotation potential and ψ can be referred to as the divergence potential. If the curl operator is applied to the Helmholtz decomposition equation, the result is

$$\text{curl}(v) = \text{curl}(\nabla\phi) + \text{curl}(\nabla \times \psi) + \text{curl}(h) = \text{curl}(\nabla \times \psi) \quad (4.6)$$

The curl of the divergence-free component is the same as the curl of the original vector field. The curl-free and harmonic components are eliminated from the equation.

If the divergence operator is also applied to the Helmholtz decomposition equation, the result is

$$\text{div}(v) = \text{div}(\nabla\phi) + \text{div}(\nabla \times \psi) + \text{div}(h) = \text{div}(\nabla\phi) \quad (4.7)$$

The divergence of the curl-free part is equal to the divergence of the original vector field. The divergence-free part and harmonic components are eliminated from the equation.

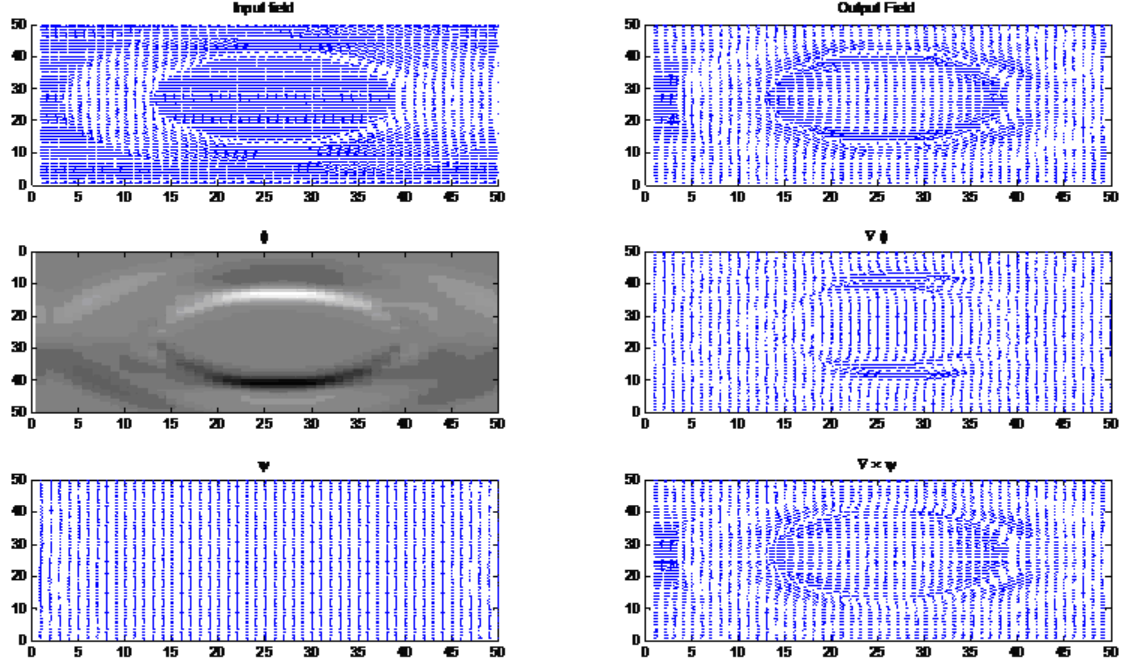


Fig 4.1. The results of Helmholtz decomposition of vector fields,

4.2. The Helmholtz Decomposition for tensor field

The above section has described the Helmholtz decomposition for vector fields in detail. However, the Helmholtz decomposition of tensor fields can be extended from the vector field to a tensor field. The Helmholtz decomposition can split a tensor field into three parts: divergence-free, curl-free and harmonic tensor field components. The Helmholtz decomposition of a tensor field is represented by,

$$D = \nabla\phi + \nabla \times \psi + H \quad (4.8)$$

where D is a tensor field, $\nabla\phi$ is the gradient of a scalar potential field ϕ , $\nabla \times \psi$ is the curl of a tensor stream field ψ and H is a harmonic tensor field. Similarly with the vector field, $\nabla\phi$ is irrotational, $\nabla \times \psi$ is solenoidal, and H is both solenoidal and irrotational with a typically small amplitude. The tensor field has the following properties, which will greatly aid in computing the values of ϕ and ψ in future computations,

$$\text{div}(\text{curl}(\psi)) = \nabla \cdot (\nabla \times \psi) = 0 \quad (4.9)$$

$$\text{curl}(\text{grad}(\phi)) = \nabla \times (\nabla \phi) = 0 \quad (4.10)$$

$$\nabla \cdot H = \nabla \times H = 0 \quad (4.11)$$

Combining Chapter 3 definitions of tensor divergence, curl and gradient of Cartesian tensors given by Heinbockel [36] with the above equation for Helmholtz decomposition of tensor field, the Helmholtz decomposition of the 2nd order tensor fields is given in Einstein notation as,

$$D_{ij} = \partial_i \phi_j + \varepsilon_{imn} (\partial_m \psi_{nj}) + H_{ij} \quad (4.12)$$

Similarly, the equation for the Helmholtz decomposition of the 4th order tensor field in Einstein notation is given as,

$$D_{ijkl} = \partial_i \phi_{jkl} + \varepsilon_{imn} (\partial_m \psi_{njkl}) + H_{ijkl} \quad (4.13)$$

The formulation of the Helmholtz decomposition for tensor fields can be made for tensors of arbitrary order, however, we only present the 2nd order and the 4th order decompositions due to the fact that these provide the basis for the experiments in Chapter 7.

CHAPTER 5:

NUMERICAL IMPLEMENTATION

5.1. Taylor Series Expansion

We have indicated how to extend the Helmholtz decomposition from the vector fields to the tensor fields in Chapter 4. From the description of Equation (4.12) and (4.13), we find that both the 2nd order tensor and the 4th order tensor have first derivative. In order to compute the first derivative of the tensor data, the Taylor series expansion for approximating a derivative will be applied. This concept provides the basis for many numerical methods, such as numerical differentiation. Taylor series is a series expansion of a function about a point with derivative of continuously increasing order derivatives. Therefore, we can apply Taylor series expansion for approximating the derivative. The Taylor series is given by

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \cdots + \frac{1}{n!}(x - x_0)^n f^{(n)}(x_0) + \cdots \quad (5.1)$$

where $n!$ denotes the factorial of n and $f^{(n)}(x_0)$ denotes the n th derivative of f evaluated at the point x_0 . And the first order Taylor series expansion can be expressed as,

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + o(x - x_0) \quad (5.2)$$

Evaluating at $x = x_0 + h$ yields,

$$f(x_0 + h) = f(x_0) + hf'(x_0) + o(x - x_0) \quad (5.3)$$

Replacing h with $-h$ in the previous equations gives,

$$f(x_0 - h) = f(x_0) - hf'(x_0) + o(x - x_0) \quad (5.4)$$

Using Equation (5.3) to subtract Equation (5.4) provides the following approximation,

$$f(x_0 + h) - f(x_0 - h) \approx 2hf'(x_0) \quad (5.5)$$

Equation (5.5) divided by $2h$ provides the centered difference:

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0 - h)}{2h} \quad (5.6)$$

5.2. Kronecker product

In order to represent the discretized operators as block matrices, where the blocks correspond to finite difference operators applied to a single tensor component. We use the Kronecker tensor product, denoted by symbol \otimes .

If given A is a $m \times n$ matrix and B is a $k \times l$ matrix,

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}_{m \times n} \quad \& \quad B = \begin{bmatrix} b_{11} & \dots & b_{1l} \\ \vdots & \ddots & \vdots \\ b_{k1} & \dots & b_{kl} \end{bmatrix}_{k \times l} \quad (5.7)$$

Then the Kronecker product $A \otimes B$ is the $mk \times nl$ block matrix as following,

$$A \otimes B = [a_{ij}B] = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix}_{mk \times nl} \quad (5.8)$$

Now, when we extend every element of matrix B in the above Equation (5.8), we will get the Kronecker product of A and B as following Expression (5.9),

$$A \otimes B = \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & \dots & a_{11}b_{1l} & \dots & a_{1n}b_{11} & a_{1n}b_{12} & \dots & a_{1n}b_{1l} \\ a_{11}b_{21} & a_{11}b_{22} & \dots & a_{11}b_{2l} & \dots & a_{1n}b_{21} & a_{1n}b_{22} & \dots & a_{1n}b_{2l} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{11}b_{k1} & a_{11}b_{k2} & \dots & a_{11}b_{kl} & \dots & a_{1n}b_{k1} & a_{1n}b_{k2} & \dots & a_{1n}b_{kl} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{11} & a_{m1}b_{12} & \dots & a_{m1}b_{1l} & \dots & a_{mn}b_{11} & a_{mn}b_{12} & \dots & a_{mn}b_{1l} \\ a_{m1}b_{21} & a_{m1}b_{22} & \dots & a_{m1}b_{2l} & \dots & a_{mn}b_{21} & a_{mn}b_{22} & \dots & a_{mn}b_{2l} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{k1} & a_{m1}b_{k2} & \dots & a_{m1}b_{kl} & \dots & a_{mn}b_{k1} & a_{mn}b_{k2} & \dots & a_{mn}b_{kl} \end{bmatrix} \quad (5.9)$$

The Kronecker product is a special case of the tensor product, and therefore, it is bilinear and associative. The Kronecker product follows several laws similar to scalars and vectors,

Distributive Law: $A \otimes (B + C) = A \otimes B + A \otimes C$

$$(A + B) \otimes C = A \otimes C + B \otimes C$$

k is a scalar: $(kA) \otimes B = A \otimes (kB) = k(A \otimes B)$

Associative Law: $(A \otimes B) \otimes C = A \otimes (B \otimes C)$

Transposition: $(A \otimes B)^T = A^T \otimes B^T$

Mixed-product property: $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$

However, the Kronecker product is not commutative, so the Kronecker product $A \otimes B$ and $B \otimes A$ are different matrices. Therefore, $A \otimes B \neq B \otimes A$

A simple example can be used to illustrate that the Kronecker products $A \otimes B$ and $B \otimes A$ are different matrices,

$$X = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \& Y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Plot the Kronecker (X, Y) , Kronecker (Y, X)

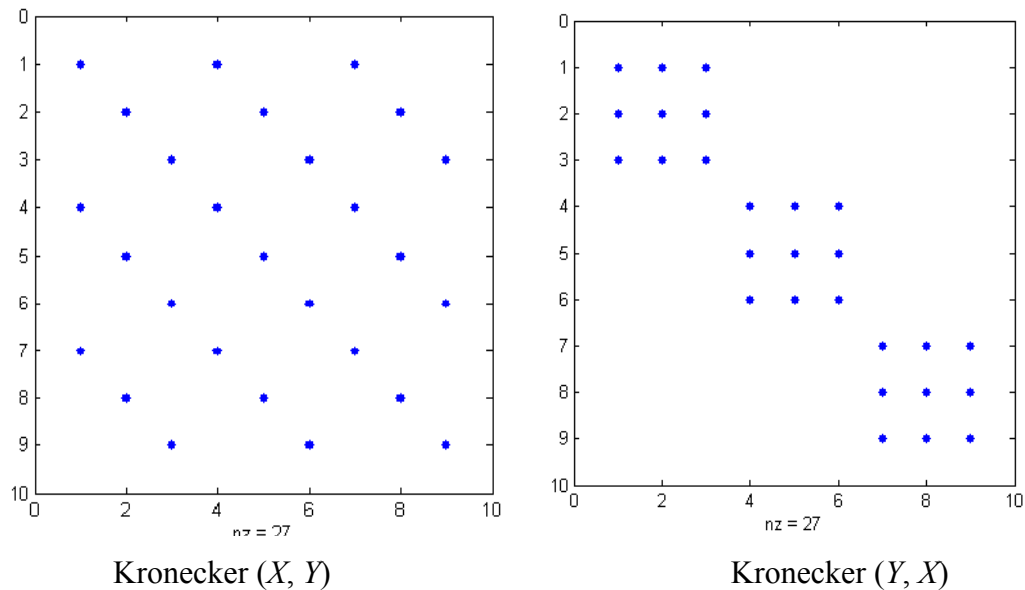
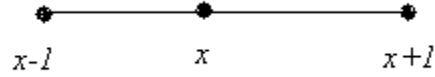


Fig. 5.1. A simple example of the comparison of Kronecker (X, Y) and Kronecker (Y, X)

5.3. Center Difference

The above section applies Taylor series to compute the first derivative by using centered difference. In the case of $1 \times n$ vector data, the following illustration can be used to demonstrate the computation of the 1-dimensional gradient. Using matrices, the data will be transformed into a $n \times 1$ column vector,



(a)

$$\begin{bmatrix} \vdots \\ d_{x-1} \\ d_x \\ d_{x+1} \\ \vdots \end{bmatrix}$$

(b)

Fig. 5.2. (a) Plot of the relationship of the target neighbor pixels in one dimension. (x) is the target pixel. (b) Transform the vector to one column vector

$$\Delta = \frac{1}{2} \begin{bmatrix} 0 & +1 & 0 & \dots & 0 \\ -1 & 0 & +1 & \ddots & \vdots \\ 0 & -1 & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & +1 \\ 0 & \dots & 0 & -1 & 0 \end{bmatrix} \quad (5.10)$$

$\Delta^x = \Delta$ is the operator to compute the first derivative of one dimension data, and the Figure (5.3) can show us the operator.

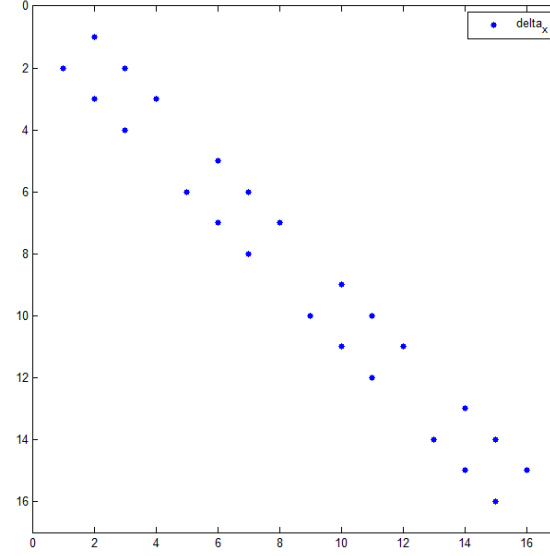
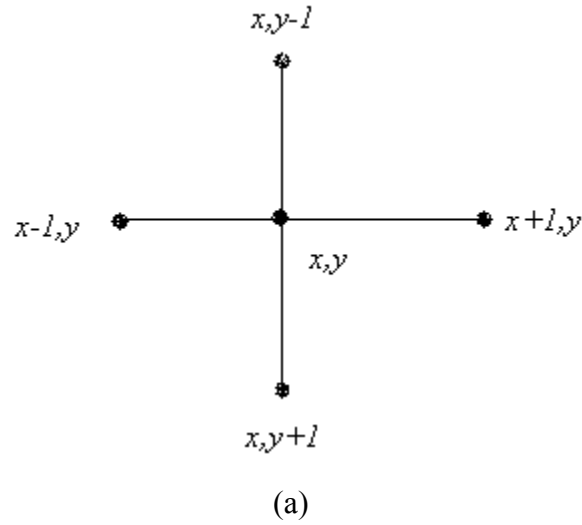


Fig. 5.3. The result of plotting Δ^x operator based on $m = 16$

Now we use a 2-dimensional pixel map to explain how to design the matrix to compute the 2D gradient, divergence, and curl. The pixel relationship of the image can be expressed by a simple map. The density of any pixel is $d_{x,y}$. The center pixel is the output pixel (x, y) . It depends on its neighbor pixels in 4 directions. The following coordinate table shows the relationship of the target neighbor pixels,



$$\begin{bmatrix} \vdots \\ d_{x,y-1} \\ \vdots \\ d_{x-1,y} \\ d_{x,y} \\ d_{x+1,y} \\ \vdots \\ d_{x,y+1} \\ \vdots \end{bmatrix}$$

(b)

Fig. 5.4. (a) Plot of the relationship of the target neighbor pixels in two dimensions. (x) is the target pixel. (b) Transform the 2D matrix data to one column vector

Still we need to transform the $m \times n$ matrix data to a $mn \times 1$ rows vector. Also,

$$\Delta^x = I_{m \times m} \otimes \Delta_{n \times n} \quad (5.11)$$

$$\Delta^y = \Delta_{m \times m} \otimes I_{n \times n} \quad (5.12)$$

Δ^x and Δ^y are the operators to compute the first derivatives of x and y directions of two dimensions data, and the Figure (5.5) can show us the two operators in one map.

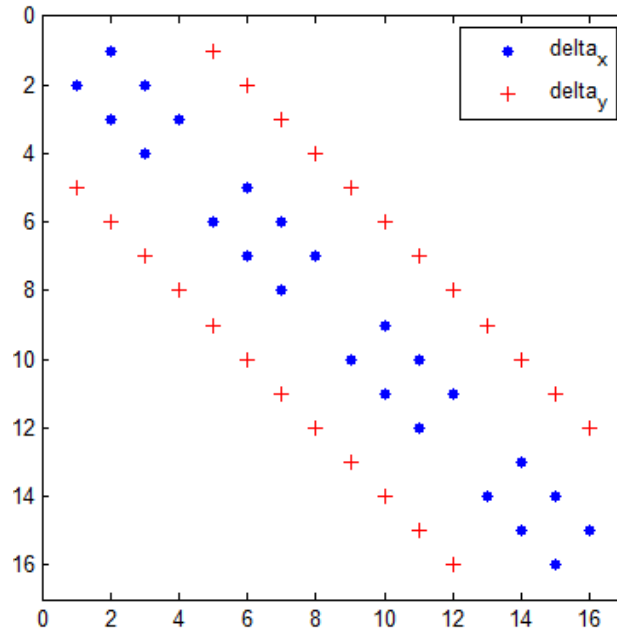
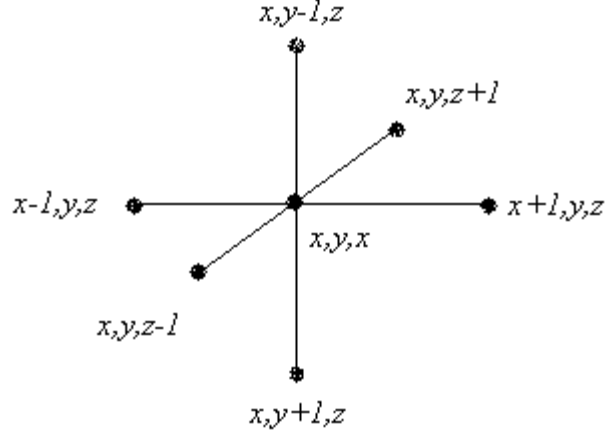


Fig. 5.5. The result of plotting Δ^x, Δ^y operators based on $m \times n = 4 \times 4$



(a)

$$\begin{bmatrix} \vdots \\ d_{x,y,z-1} \\ \vdots \\ d_{x,y-1,z} \\ \vdots \\ d_{x-1,y,z} \\ d_{x,y,z} \\ d_{x+1,y,z} \\ \vdots \\ d_{x,y+1,z} \\ \vdots \\ d_{x,y,z+1} \\ \vdots \end{bmatrix}$$

(b)

Fig. 5.6. (a) Plot of the relationship of the target neighbor pixels in three dimensions. (x) is the target pixel. (b) Transform the 3D matrix data to one column vector

To compute the 3-dimensional matrix data,

$$\Delta^x = I_{p \times p} \otimes I_{m \times m} \otimes \Delta_{n \times n} \quad (5.13)$$

$$\Delta^y = I_{p \times p} \otimes \Delta_{m \times m} \otimes I_{n \times n} \quad (5.14)$$

$$\Delta^z = \Delta_{p \times p} \otimes I_{m \times m} \otimes I_{n \times n} \quad (5.15)$$

where $I_{n \times n}$ is a $n \times n$ identity matrix, and $\Delta_{n \times n}$ is a $n \times n$ finite difference matrix, and \otimes is the Kronecker product,

Δ^x , Δ^y , and Δ^z are the operators to compute the first derivatives of x, y, and z directions of three dimensions data, and the Figure (5.7) can show us the three operators in one map.

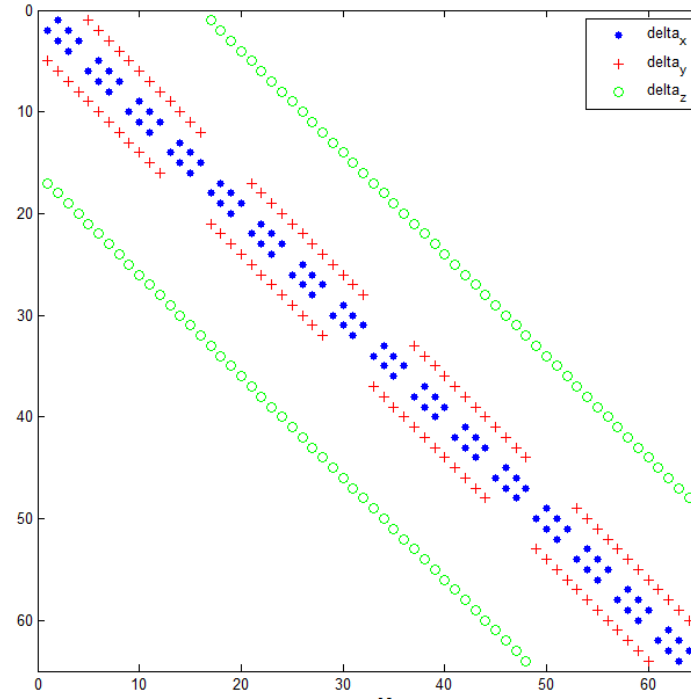


Fig. 5.7. The result of plotting Δ^x , Δ^y , Δ^z operators based on $m \times n \times p = 4 \times 4 \times 4$

In order to illustrate how the numerical implementation works, the following sections will use the 2nd order tensor data to indicate the entire numerical implementation procedure.

For computational convenience, we will reshape the 2nd tensor fields where

$$D = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{bmatrix}$$

into a column vector to arrive at the following Equation (5.16)

$$D = [D_{xx} \quad D_{xy} \quad D_{xz} \quad D_{yx} \quad D_{yy} \quad D_{yz} \quad D_{zx} \quad D_{zy} \quad D_{zz}]^T \quad (5.16)$$

It becomes apparent that the column has 9 rows. The above expression tells us that both the elements (for example, D_{xx}, \dots) of each tensor component (for example, $[D_{xx}, D_{yy}, D_{zz}, \dots]$) and the tensor components are vectorized in lexical order of the spatial coordinates (x, y, z) . If the size of the input, the 2nd order tensor field, is $m \times n \times p$, the result is $9 \times m \times n \times p$ rows.

5.4. Gradient of the 2nd Order Tensor Field G

According to the central differences for approximately derivatives Equation (5.6), we can compute the gradient of the 2nd order tensor field ϕ

$$\begin{bmatrix} G_{xx} \\ G_{xy} \\ G_{xz} \\ G_{yx} \\ G_{yy} \\ G_{yz} \\ G_{zx} \\ G_{zy} \\ G_{zz} \end{bmatrix} = G \begin{bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{bmatrix} \quad (5.17)$$

G represents the gradient matrix, $G = \nabla \cdot$. Multiplying the gradient matrix by ϕ will get result in the gradient of the tensor field, ϕ .

$$G = \begin{bmatrix} \Delta^x & 0 & 0 \\ 0 & \Delta^x & 0 \\ 0 & 0 & \Delta^x \\ \Delta^y & 0 & 0 \\ 0 & \Delta^y & 0 \\ 0 & 0 & \Delta^y \\ \Delta^z & 0 & 0 \\ 0 & \Delta^z & 0 \\ 0 & 0 & \Delta^z \end{bmatrix} \quad (5.18)$$

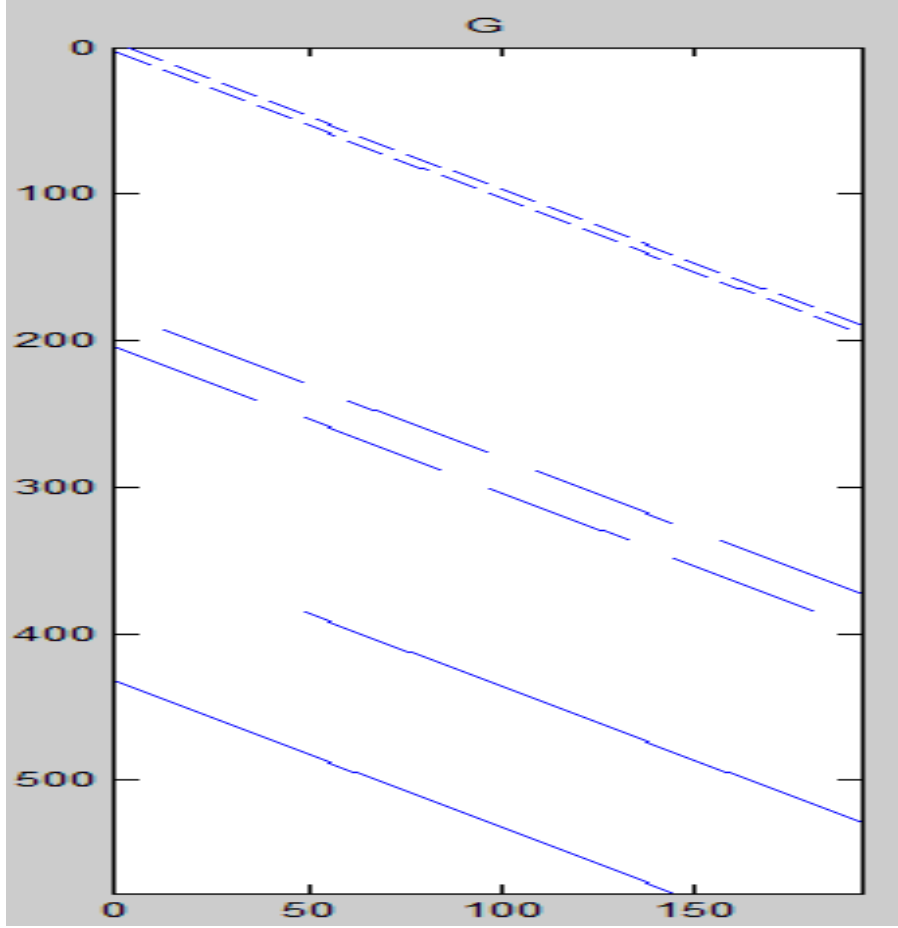


Fig. 5.8. Plot of the gradient of the 2nd order tensor field, $m \times n \times p = 4 \times 4 \times 4$

5.5 Curl of the 2nd Order Tensor Field C

Similarly, we can compute the curl of the second tensor field ψ based on the operators described above,

$$\begin{bmatrix} C_{xx} \\ C_{xy} \\ C_{xz} \\ C_{yx} \\ C_{yy} \\ C_{yz} \\ C_{zx} \\ C_{zy} \\ C_{zz} \end{bmatrix} = C \begin{bmatrix} \psi_{xx} \\ \psi_{xy} \\ \psi_{xz} \\ \psi_{yx} \\ \psi_{yy} \\ \psi_{yz} \\ \psi_{zx} \\ \psi_{zy} \\ \psi_{zz} \end{bmatrix} \quad (5.19)$$

C represents for the curl matrix $C = \nabla \times$, which is the sparse block matrix. Also, the above equation can be rewritten as $C\psi = \nabla \times \psi$. Using the product of the curl matrix and ψ will provide the curl of the tensor field ψ .

$$C = \begin{bmatrix} 0 & 0 & 0 & -\Delta^z & 0 & 0 & \Delta^y & 0 & 0 \\ 0 & 0 & 0 & 0 & -\Delta^z & 0 & 0 & \Delta^y & 0 \\ 0 & 0 & 0 & 0 & 0 & -\Delta^z & 0 & 0 & \Delta^y \\ \Delta^z & 0 & 0 & 0 & 0 & 0 & -\Delta^x & 0 & 0 \\ 0 & \Delta^z & 0 & 0 & 0 & 0 & 0 & -\Delta^x & 0 \\ 0 & 0 & \Delta^z & 0 & 0 & 0 & 0 & 0 & -\Delta^x \\ -\Delta^y & 0 & 0 & \Delta^x & 0 & 0 & 0 & 0 & 0 \\ 0 & -\Delta^y & 0 & 0 & \Delta^x & 0 & 0 & 0 & 0 \\ 0 & 0 & -\Delta^y & 0 & 0 & \Delta^x & 0 & 0 & 0 \end{bmatrix} \quad (5.20)$$

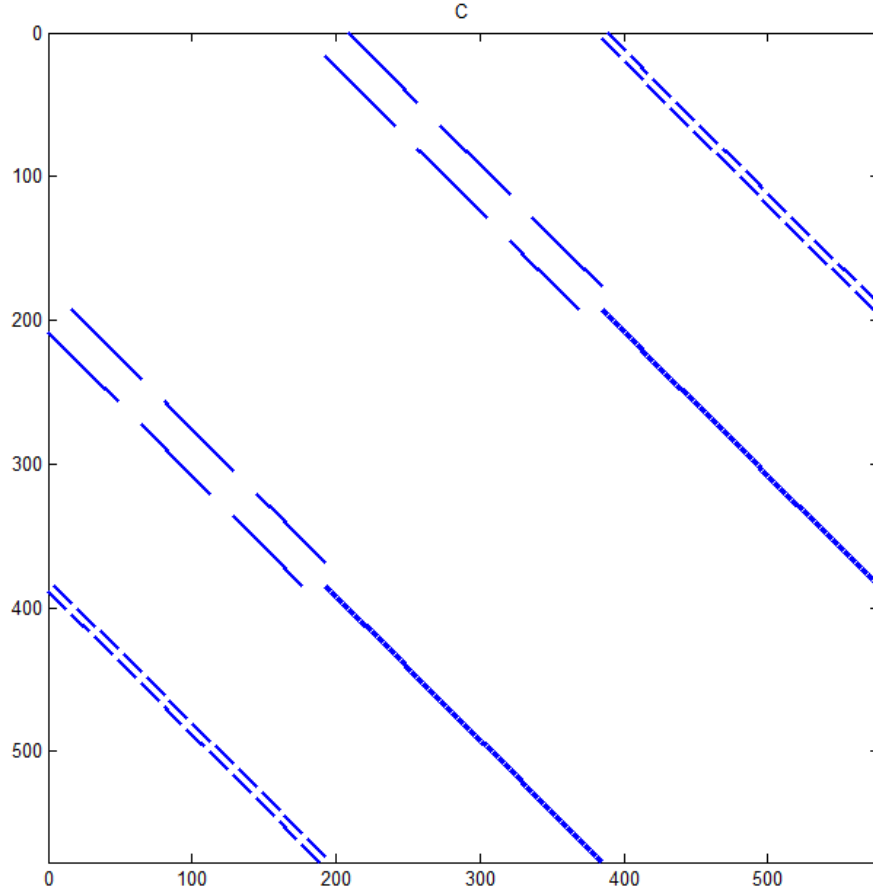


Fig. 5.9. Plot of the Curl of the 2nd order tensor field, $m \times n \times p = 4 \times 4 \times 4$

As a matter of fact, the 4th order tensor data can also be computed by using the discretized operators, which are similar to the discretized operators of the 2nd order tensor. Because each tensor contains 81 rows, the matrices will not be provided, however, codes will be used to realize the 4th order tensor decomposition.

5.6. Numerical Implementation of the Helmholtz Decomposition

Using the above definitions for the C and G matrices we can numerically implement the Helmholtz decomposition by solving for ϕ and ψ . If we simply change the Helmholtz decomposition expression, then we will get the following equation,

$$H = D - G\phi - C\psi \quad (5.21)$$

Now the Helmholtz decomposition is given by the solution to the least squares problem,

$$\min_{\phi, \psi} \|D - G\phi - C\psi\|_F^2 \quad (5.22)$$

Where $\|\cdot\|_F$ denotes the Frobenius norm of the tensor, and $\|X_{ik}\|_F = \text{trace}(X_{ij}X_{jk})$.

$$\|r\|_F^2 = r^T r \quad (5.23)$$

Therefore, Equation (5.20) can be converted to

$$\begin{aligned} H(\phi, \psi) &= \|D - G\phi - C\psi\|_F^2 = (D - G\phi - C\psi)^T (D - G\phi - C\psi) \\ &= (D^T - \phi^T G^T - \psi^T C^T) (D - G\phi - C\psi) \\ &= D^T D - D^T (G\phi + C\psi) - (\phi^T G^T + \psi^T C^T) D \\ &\quad + (\phi^T G^T G\phi + \phi^T G^T C\psi + \psi^T C^T G\phi + \psi^T C^T C\psi) \end{aligned} \quad (5.24)$$

Because G is the compact operator of the gradient, and C is the compact operator of curl, and stemming from the fact $\text{div}(\text{curl}(\psi)) = 0$, and $\text{curl}(\text{grad}(\phi)) = 0$, we obtain the following fact,

$$C^T G = G C^T = 0 \quad (5.25)$$

Therefore, Equation (5.24) can be simplified to,

$$H(\phi, \psi) = D^T D - D^T (G\phi + C\psi) - (\phi^T G^T + \psi^T C^T) D + (\phi^T G^T G \phi + \psi^T C^T C \psi) \quad (5.24)$$

The goal is to find values of ϕ and ψ which minimize the error. In multivariable calculus this requires us to find the values of (ϕ, ψ) such that

$$\begin{cases} \frac{\partial H}{\partial \phi} = 0 \\ \frac{\partial H}{\partial \psi} = 0 \end{cases} \quad (5.26)$$

Applying Equation (5.26) to Equation (5.24), and considering that ϕ and ψ are independent, two new equations are used in order to calculate ϕ and ψ individually,

$$\begin{cases} \frac{\partial H}{\partial \phi} = -2G^T D + 2G^T G \phi = 0 \\ \frac{\partial H}{\partial \psi} = -2C^T D + 2C^T C \psi = 0 \end{cases} \quad (5.27)$$

Simplifying the Equation (5.27) will obtain the new expression

$$\begin{cases} G^T G \phi = G^T D \\ C^T C \psi = C^T D \end{cases} \quad (5.28)$$

5.7 Conjugate Gradient Algorithm

Section 5.5 has described how to calculate the matrix equation of ϕ and ψ . The format of the matrix equations of ϕ and ψ can be generally described using,

$$Ax = b \quad (5.29)$$

Usually, we will use the inverse matrix transformation to compute the unknown vector x , due to the fact that $A^{-1}A = I$

$$x = A^{-1}b \quad (5.30)$$

Given a matrix A equal to

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad (5.31)$$

The inverse matrix is given by,

$$A^{-1} = \frac{1}{|A|} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \quad (5.32)$$

If the size of matrix A is small, the computation speed is fast. If the size of matrix A is large or the matrix is ill-conditioned, the matrix inversion method does not work. For example,

tensors dataset size is $64 \times 64 \times 5(n \times m \times p)$			
Order	Elements No.	Size of G	Size of C
2	9	$(9nmp) \times (3nmp)$	$(9nmp) \times (9nmp)$
		1.1325e+010	3.3974e+010

Table 5.1. The matrix size of G and C for the 2nd and the 4th order tensors

From Table (5.2), we find that the sizes of $grad(G)$ and $curl(C)$ of tensor fields are very large. If we still use the inverse matrix method to compute ϕ and ψ , the computation will be extremely time-consuming. If the order is 4, the sizes of G and C will be even greater than those of the 2nd order.

Therefore solving large systems of linear Equations (5.25) can be agonizing pain for many. Fortunately, Hestenes and Stiefel [12] created the Conjugate Gradient Method (CG), and the JR Shewchuk [31] describes the algorithm, which is effective to solve large systems of equations.

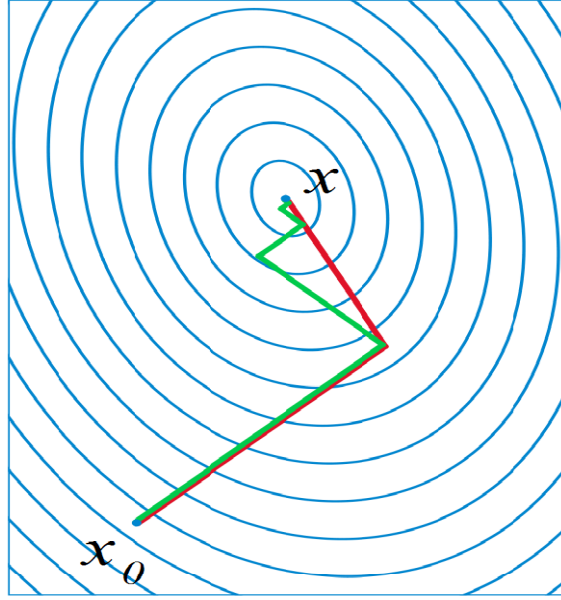


Fig. 5.10. The image of method for the conjugate gradients and gradient descent (wiki)

Green lines: iterations of gradient descent.

- Subsequent search directions , v , are perpendicular v_i
- $v_i^T v_{i+1} = 0$

Red lines: iterations of conjugate gradient method.

- In CG methods the search directions are conjugate v_i
- $v_i^T A v_{i+1} = 0$

Now we can resolve the linear system equation $Ax = b$ by using conjugate gradient method, where A is a $n \times n$ symmetric and positive-definite matrix.

Suppose that $\{v_k\}$ is a sequence of n mutually conjugate directions, therefore, they are orthogonal each other and linear independent. We can expand them

$$x = \sum_{i=1}^n \alpha_i v_i \quad (5.33)$$

and

$$b = Ax = \sum_{i=1}^n \alpha_i Av_i \quad (5.34)$$

Using the character of mutually conjugate of v_k to obtain the following equation,

$$v_k^T b = Ax = \sum_{i=1}^n \alpha_i v_k^T Av_i = \alpha_k v_k^T Av_k \quad (5.35)$$

Therefore,

$$\alpha_k = \frac{v_k^T b}{v_k^T Av_k} \quad (5.36)$$

We have describe how conjugate gradient method works, in fact conjugate gradients is an iterative method, the result of which approaches real value gradually. The name “Conjugate Gradients” is a bit of a misnomer, because the gradients are not conjugate, and the conjugate directions are not all gradients. “Conjugated Gradients” would be more accurate.

We can give an initial guess x_0 to search for the solution and stop when the value is close to the real solution.

Let r_k be the residual at the k th iteration:

$$r_k = b - Ax_k \quad (5.37)$$

Because the conjugate constraint is an orthonormal type constraint, the Gram-Schmidt orthonormalization gives the following equation:

$$v_{k+1} = r_k - \sum_{i \leq k} \frac{v_i^T Ar_k}{v_i^T Av_i} v_i \quad (5.38)$$

The next optimal point following this direction is given as,

$$v_{k+1} = x_k + \alpha_k v_k \quad (5.39)$$

$$\alpha_{k+1} = \frac{r_k^T r_k}{v_k^T Av_k} \quad (5.40)$$

If A is a symmetric, positive –definite matrix, the conjugate gradient method works very well. (A is real positive definite if and only if for all $x \neq 0$, $x^T A x > 0$)

However, the conjugate gradient method is not suitable for asymmetric linear systems. Therefore, the biconjugate gradient (BiCG) method designed to solve asymmetric linear systems, can be used. The BiCG method delivers the same results as the conjugate gradient method for symmetric positive definite systems, but at twice the cost per iteration.

BiCG can be used to solve systems where A is not symmetric, positive-definite, square. A solution to the least squares problem

$$\min_x \|Ax - b\|^2 \quad (5.41)$$

can be found by setting the derivative of Equation (5.41) to zero:

$$A^T A x = A^T b \quad (5.42)$$

If A is square and nonsingular, the solution to Equation () is the solution to $Ax = b$. If A is not square, and $Ax = b$ is overconstrained, --that is, has more linearly independent equations than variables -- then there may or may not be a solution to $Ax = b$. However, it is always possible to find a value of x that minimizes Equation (5.42), the sum of the squares of the errors of each linear equation.

$A^T A$ is symmetric and positive (for any x , $x^T A^T A x = \|Ax\|^2 > 0$). If $Ax = b$ is not underconstrained, then $A^T A$ is nonsingular, and methods CG can be used to solve Equation (5.28). The only nuisance in doing so is that the condition number of $A^T A$ is the square of that of A , so convergence is significantly slower.

Given the inputs A , b , a starting vector value $x_0, \widetilde{x}_0, \widetilde{b}$, the number of iterations *loops*, and an error tolerance $\varepsilon < 1$, unlike the conjugate gradient method, biconjugate gradient method needs to perform multiplications matrix A . and residuals value r will reflect how the approximate solutions x close to the real solution. The conjugate gradient and biconjugate gradient methods are mature technique, I refer to this technique from internet to apply in my method to compute asymmetric and non-positive definite matrix from linear equation system.

Table 5.2. Pseudocode: the algorithm for the topological visualization using the Helmholtz decomposition

<i>Algorithm: topological visualization for diffusion tensor field</i>	
<i>Using generalized Helmholtz Decomposition</i>	

<i>Initializing</i>	
<i>Compute the $\Delta^x, \Delta^y, \Delta^z$ Matrix</i>	
<i>Compute the C Matrix based on $\Delta^x, \Delta^y, \Delta^z$ Matrix</i>	
<i>Compute the G Matrix based on $\Delta^x, \Delta^y, \Delta^z$ Matrix</i>	
<i>Iterations</i>	
	<i>biconjugate gradients method phi = BiCG(G, D)</i>
	<i>biconjugate gradients method psi = BiCG(C, D)</i>
<i>End</i>	
 <i>Subfunction BiCG()</i>	
$r_0 = b - Ax_0$	
$\tilde{r}_0 = \tilde{b} - \tilde{x}_0 A$	
$v_0 = r_0$	
$\tilde{v}_0 = \tilde{r}_0$	
$k=0;$	
<i>while</i> ($\frac{r_k}{v_k} > \text{tolerant value}$)	
	$\alpha_k = \frac{r_k \tilde{r}_k}{v_k^T A v_k}$
	$x_{k+1} = x_k + \alpha_k v_k$
	$\tilde{x}_{k+1} = \tilde{x}_k + \alpha_k \tilde{v}_k$
	$r_{k+1} = r_k - \alpha_k A v_k$
	$\tilde{r}_{k+1} = \tilde{r}_k - \tilde{\alpha}_k A^T \tilde{v}_k$
	$\beta_k = \frac{\tilde{r}_{k+1}^T r_{k+1}}{\tilde{r}_k^T r_k}$
	$v_{k+1} = r_{k+1} + \beta_k v_k$
	$\tilde{v}_{k+1} = \tilde{r}_{k+1} + \beta_k A^T \tilde{v}_k$
	$k+1$
	<i>if</i> $k = \text{max iteration}$ <i>then quit</i>
<i>end</i>	

CHAPTER 6:

TOPOLOGICAL VISUALIZATION

Topological concepts are present in many areas of mathematics. Topology is a beautiful field of mathematics dealing with the shapes of objects, without regard to distances. Topology can be used to abstract inherent connectivity of objects while ignoring their detailed form.

The 2nd and 4th order tensors describe a wide range of physical phenomena. Tensors can represent diffusivity [4, 19], the fiber orientation distribution function (ODF) in DT-MRI [3], mechanical stress and strain [13], and so on. More abstractly, tensors may represent covariance, skew, kurtosis and higher order moments of multivariate probability distributions as well as homogeneous polynomials. In our experiments we explore the topology of the diffusion tensor field (as opposed to the field of diffusion ODFs) due to the fact that the divergence operator has an important physical meaning when applied to the diffusion tensor. Zero divergence of the tensor field implies conservation of mass under a constant concentration gradient. However, analysis of the ODF field represented as the 4th order tensors, as described by Barmpoutis et al. [3] is a topic for future investigation.

6.1 Critical Points and Vortices

In Chapter 4, we described how the Helmholtz decomposition can separate a tensor field into divergence-free and curl-free components. The curl-free component is irrotational, which can be used to isolate local maxima and minima of divergence (foci of sources and sinks) in tensor fields. The divergence-free component is solenoidal, which can be used to isolate the centers of vortices in tensor field. Using critical points (as mentioned in Chapter 4) and a few separating curves (separatrices) to visualize the tensor fields can reduce a dense set of input data to provide a simple structure of the field.

The critical points are called singularities, which are the only locations that streamlines in a vector field can intersect. Critical points also are the locations at which the magnitude of the vector vanishes. A first order critical point can be classified according to the eigenvalue of the Jacobian matrix, and for the purpose of the experiments, we only focus on the linear case. If the real part of an eigenvalue is positive, it indicates a repelling nature in which the field is expanding from that point. This critical point is referred to as a sink. If the real part of an eigenvalue is negative, it indicates an attracting nature in which the field is shrinking to that point. This critical point is referred to as a source.

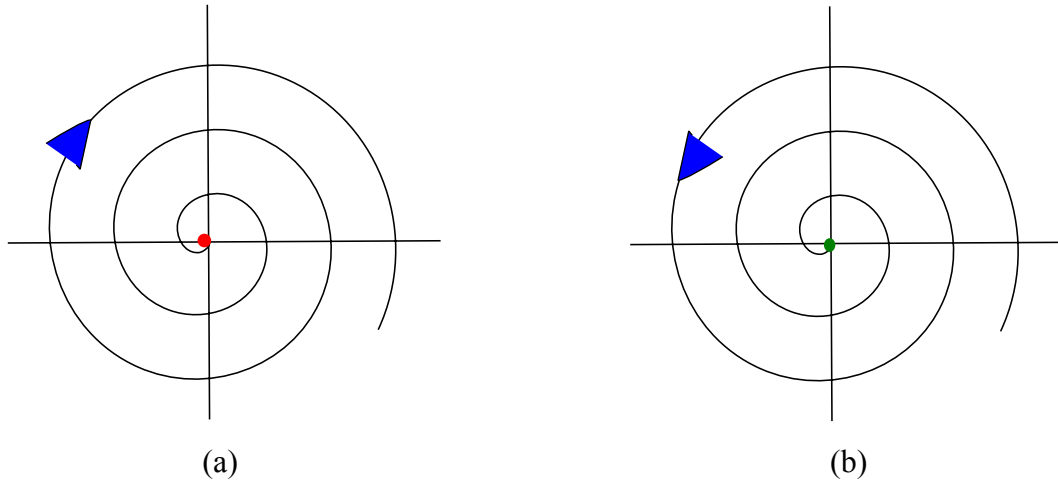


Fig. 6.1 Vortices, (a) is the source, (b) is the sink

A vortex is defined as motion of a fluid which is swirling rapidly around a center which is a critical point. If the critical point is a source, the vortex is expanding. Otherwise, the critical point is a sink, the vortex is shrinking. Therefore, the critical points and vortex can effectively be represented by the main structure of vector fields. We propose to extend the same concepts to tensor fields.

6.2. Degenerate Points and Separatrices

The streamlines in vector fields only intersect at critical points, and the hyperstreamlines in tensor fields meet only at degenerate points. Degenerate points play an analogous role to critical points of vector fields in the topological analysis of tensor fields. These points are the basic singularities in the topology of tensor fields. Thierry and Lambertus's [20] define degenerate points, '*A point x_0 is a degenerate point of the tensor field D if the two eigenvalues of D are equal to each other at x_0 , ie $\lambda_1(x_0) = \lambda_2(x_0)$* '. We can use this idea to degenerate critical points.

We propose to extend the concept of degenerate points to tensor fields of arbitrary order by defining them in terms of local extrema of the magnitude of divergence and curl.

6.3. Topological Segmentation

Helmholtz decomposition can be used to separate a tensor field into divergence-free and curl-free components, which can simplify the rich and complex datasets and assist our analysis. In order to apply the topological visualization, we classify critical points by using the signed error function,

$$e = \|D - C\psi\|_F^2 - \|D - G\psi\|_F^2 \quad (6.1)$$

Analyzing the above Equation (6.1) provides the following results. When $e < 0$, $\|D - G\psi\|^2$ becomes the dominant part, so the field is better approximated by curl. When $e > 0$, $\|D - C\psi\|^2$ becomes the dominant part, and the field is therefore better approximated by $grad(\psi)$. Li et al. [16] used a similar decision criterion through, b , which is a ratio of the two errors. Here the contour line, $e = 0$, separates the field into relatively low divergence and low curl regions. Therefore, this function can be used to roughly segment the field into two regions. Within the low divergence region we characterize critical points in

terms of the curl magnitude $\|curl(curl(\psi))\|$, and in the low curl region we investigate the divergence magnitude $\|div(grad(\phi))\|$.

Even though topological visualization can be used to extract the skeleton of the datasets, our approach is used to display the field in a way which reveals the topological features. To reiterate, the Helmholtz decomposition can separate a dense tensor field into simple divergence-free and curl-free parts, which is used to simplify the complex datasets into two simple fields. We then use signed error functions to classify the low curl and low divergence region, and it roughly splits this into irrotational and solenoidal regions. These features give a concise representation of the global structure of the tensor field.

CHAPTER 7:

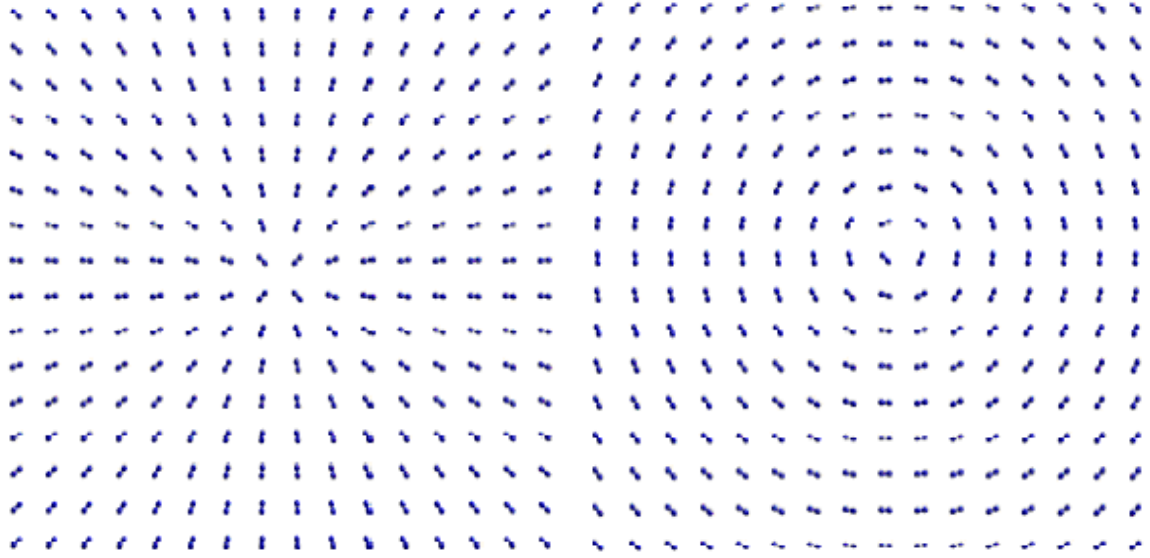
EXPERIMENTAL RESULTS

The generalized Helmholtz decomposition was implemented in Matlab and run on a system with Intel Quad Core QX6700 2.66 GHz CPU and 4 GB RAM. The algorithm was applied to the synthetic and real datasets as described below. The fitting quality of the decomposition can be analyzed in terms of the magnitude of the harmonic term $\|D - grad(\phi) - curl(\psi)\|$ which should be near zero.

7.1. Experiment 1: simple 2nd and 4th synthetic tensor fields

A synthetic 2nd order tensor field was generated from one source, one sink and two vortices shown in Figure (7.1). The input data for test are generated by computing $D = (D_1 + D_2 + D_3 + D_4)^2$.

The Figure (7.1) is the synthetic data for the 2nd order tensor fields, which generate the results of Figure (7.2). The tensor fields in Figure (7.1) and Figure (7.2) are visualized by plotting the radial surfaces $r(v) = D_{ij}v_i v_j$ for the basis of unit vectors v . The surface is colored blue when r is positive and red when r is negative. The results of the generalized Helmholtz decomposition are shown in Figure (7.2).



(a) $D1$

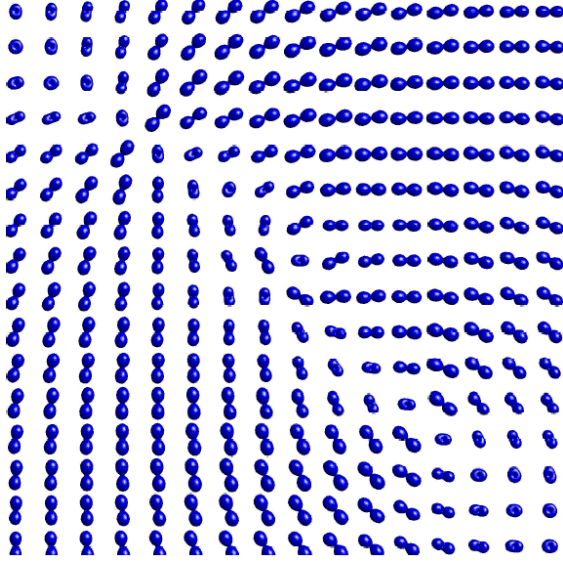
(b) $D2$



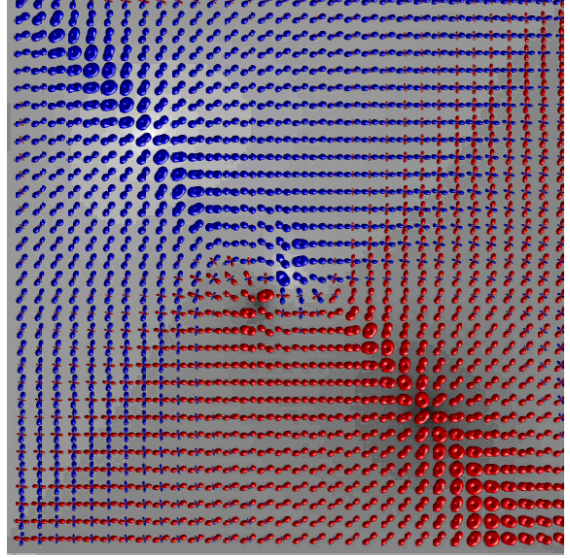
(c) $D3$

(d) $D4$

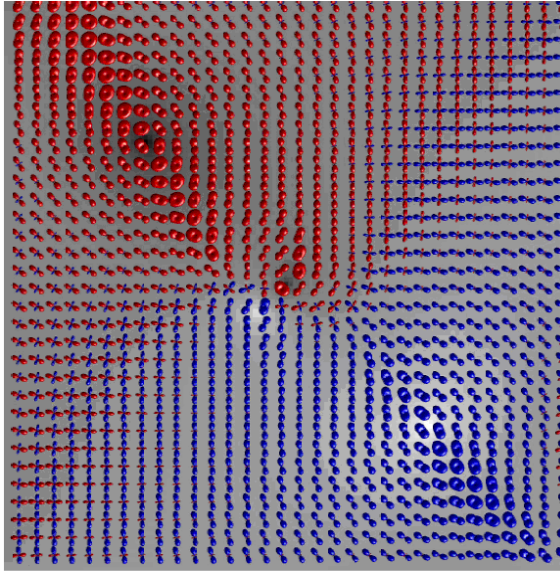
Fig. 7.1. (a) and (d) illustrate vortices while (b) and (c) illustrate a source and sink respectively, which are used to construct a synthetic 2nd order tensor field.



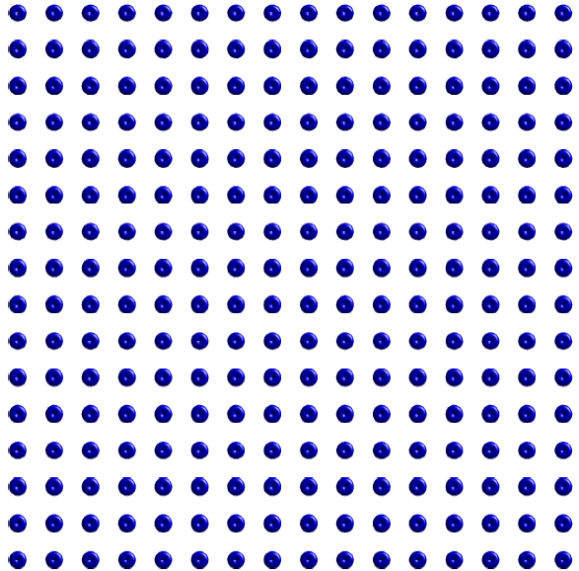
(a) Synthetic field $D = (D_1 + D_2 + D_3 + D_4)^2$



(b) $grad(\phi)$



(c) $curl(\psi)$



(d) Harmonic term H

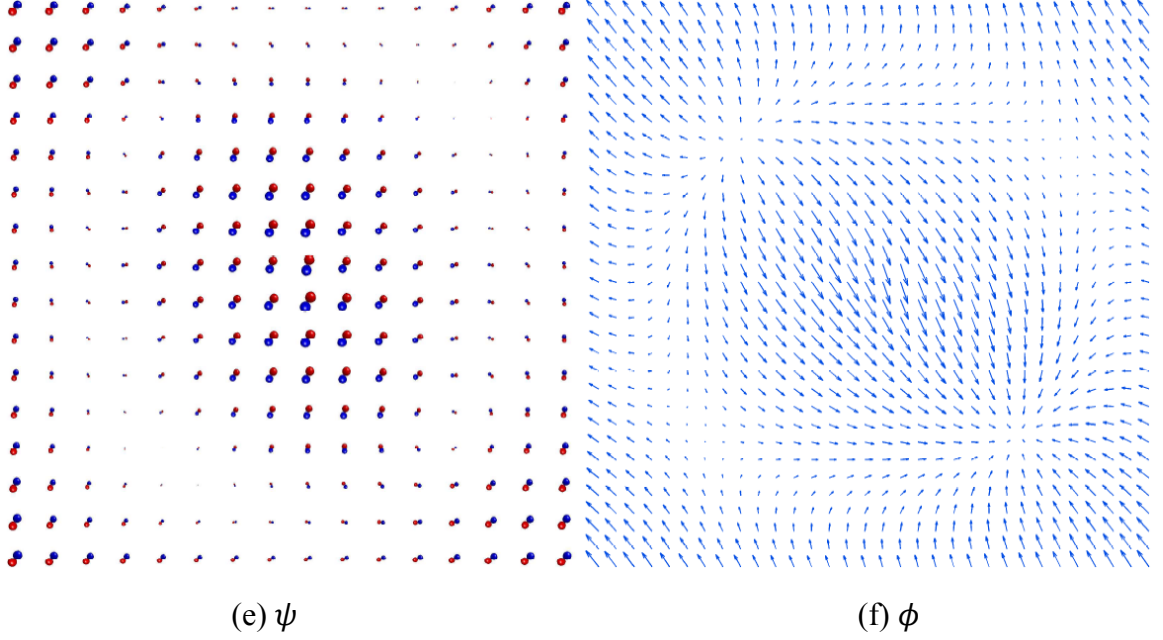
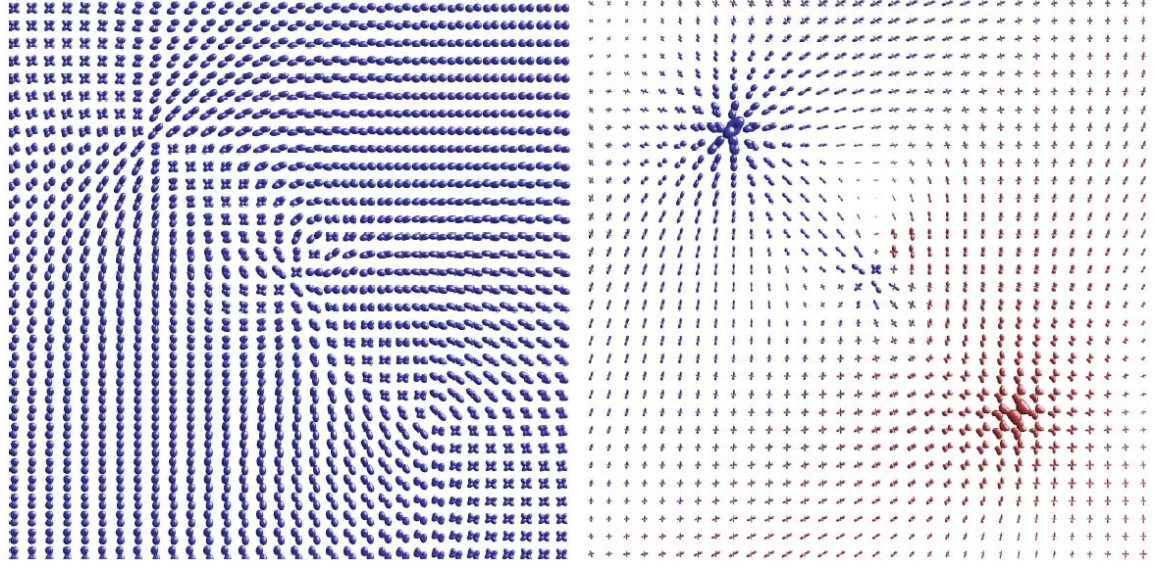


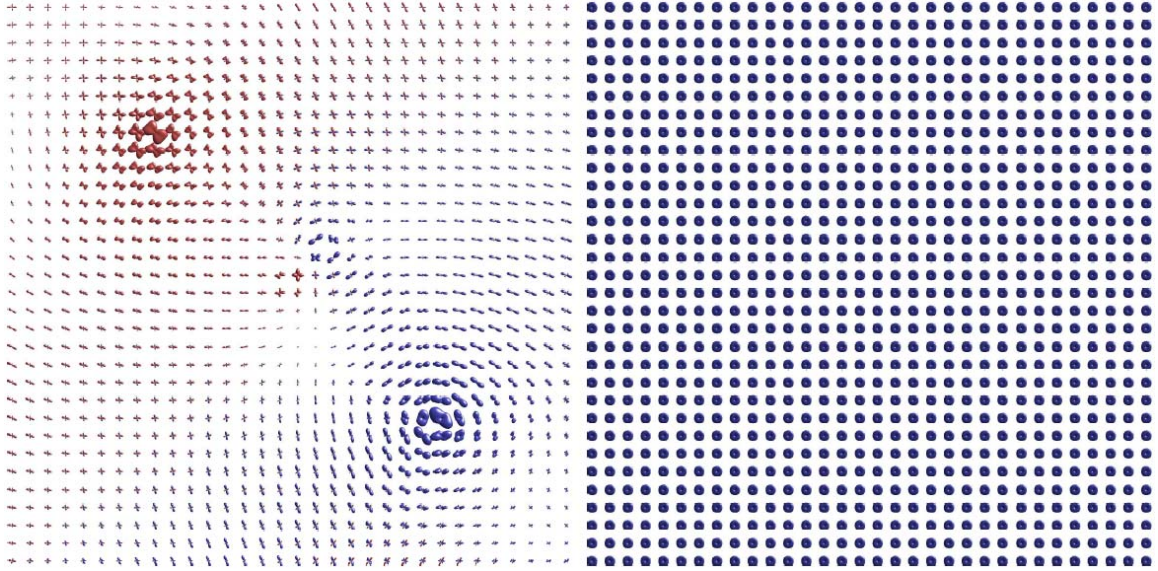
Fig. 7.2. Helmholtz decomposition results for the 2nd order synthetic tensor field $D = \text{grad}(\phi) + \text{curl}(\psi) + H$

Also, the synthetic 4th order tensor field was constructed from one source, one sink and two vortices similar to those shown in Figure (7.1). The results of the generalized Helmholtz decomposition of this field are shown in Figure (7.3). The tensor fields also are visualized by plotting the radial surfaces $r(v) = D_{ijkl}v_i v_j v_k v_l$ for the basis of unit vectors v . The surface is also colored blue when r is positive and red when r is negative. From the resulting images of ϕ and ψ , we find that ψ has two vortices. Similarly, ϕ has one source and one sink and flow direction.



(a) Synthetic field $D = (D_1 + D_2 + D_3 + D_4)^2$

(b) $\text{curl}(\psi)$



(c) $\text{grad}(\phi)$

(d) Harmonic term H

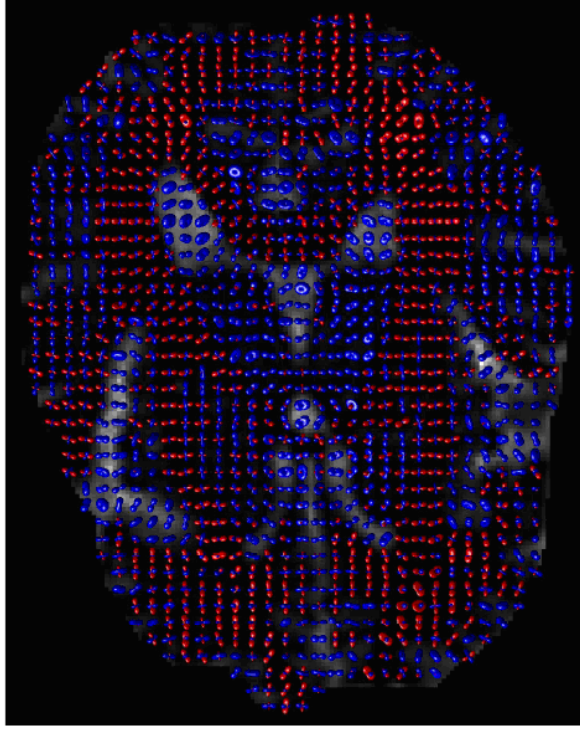
Fig. 7.3. Helmholtz decomposition results for the 4th order synthetic tensor field $D = \text{grad}(\phi) + \text{curl}(\psi) + H$

Several interesting observations can be made from above results. As evidenced from the negative (red-colored) lobes in Figure (7.2) and (7.3). It is obvious that the decomposition does not only preserve positivity. The critical points in the original field shown in Figure (7.2.a) and (7.3.a) are not clearly visible. However, they are quite evident in the

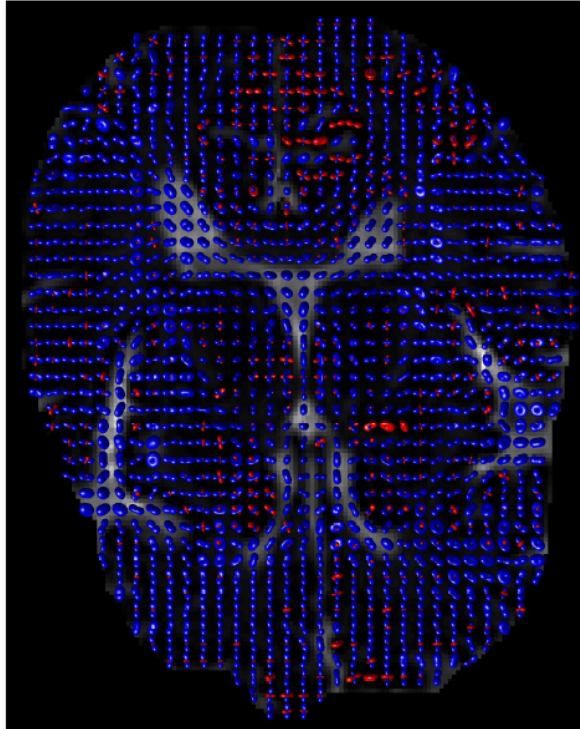
decomposed fields. Also there seems to be a correspondence between sources of positive-definite tensors and vortices of negative-definite tensors in the decomposition fields. The harmonic field, which is typically of small magnitude for vector field decompositions, can be substantial in terms of the tensor trace, but it is extremely smooth – nearly constant in all of our synthetic field experiments.

7.2. Experiment 2: a 2nd order MRI tensor data of human being

The Helmholtz decomposition was also applied to real datasets of a diffusion tensor MRI of the human brain. The data were acquired on a 3.0 Tesla General Electric Medical Systems Horizon LX imaging system with a diffusion weighted spin echo pulse sequence. Imaging parameters were: effective TR = 9000 ms, TE = 78 ms, NEX = 1. Diffusion-weighted images were acquired with 25 different gradient directions with $b = 1000 \frac{s}{mm^2}$ and a single image was acquired with $b \approx 0$. The image field of view was 24×24 cm and the acquisition matrix was $256 \times 256 \times 30$. The 2nd order tensors were computed from the diffusion weighted images by performing a least squares fit to the logarithm of the signal attenuation. Similarly, the tensor fields in Figure (7.4) are visualized by plotting the radial surfaces $r(v) = D_{ij}v_i v_j$ for the basis of unit vectors v . The surface is colored blue when r is positive and red when r is negative.



(a) $\text{curl}(\psi)$

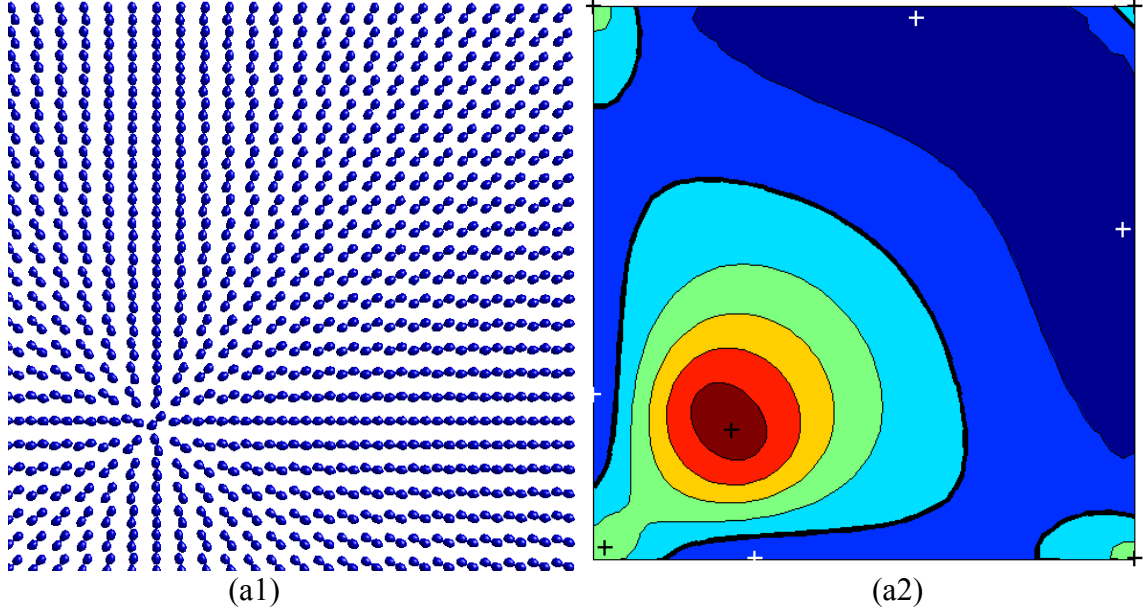


(b) $\text{grad}(\phi)$

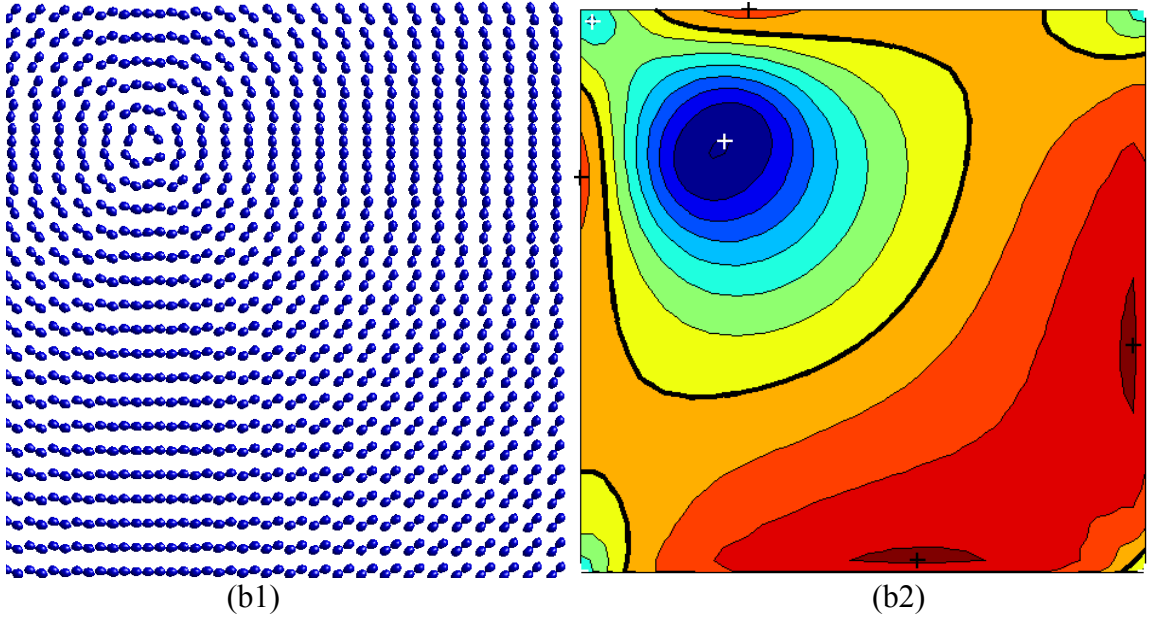
Fig. 7.4. Helmholtz decomposition results for the 2nd order DT-MRI datasets of human brain

7.3. Experiment 3: more different 2nd order and the 4th order synthetic tensor fields

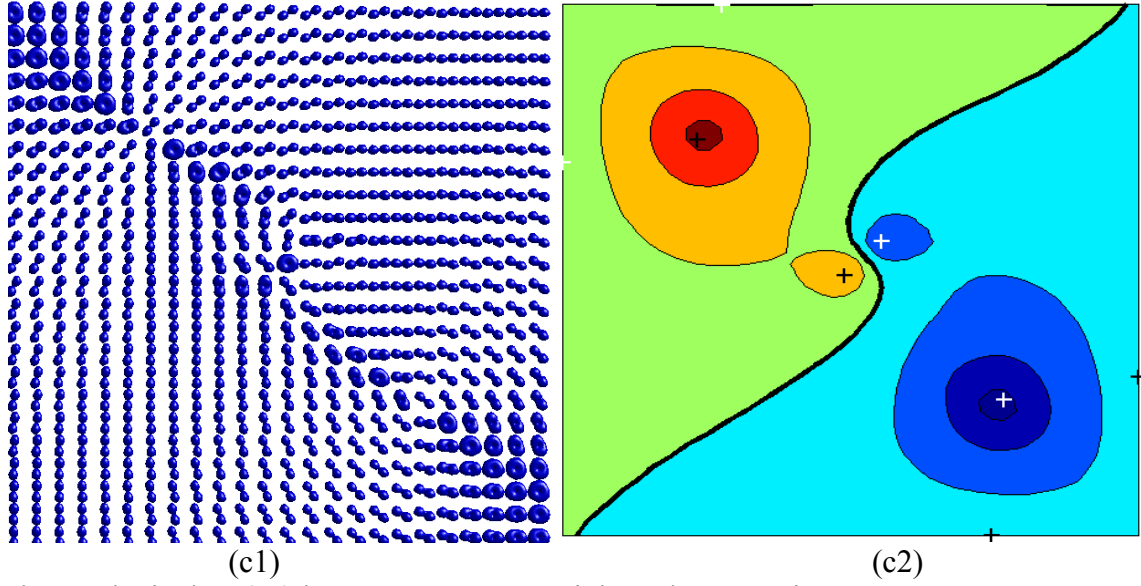
We display a few experiment results for the 2nd order and 4th order synthetic tensor datasets to test our idea and algorithms.



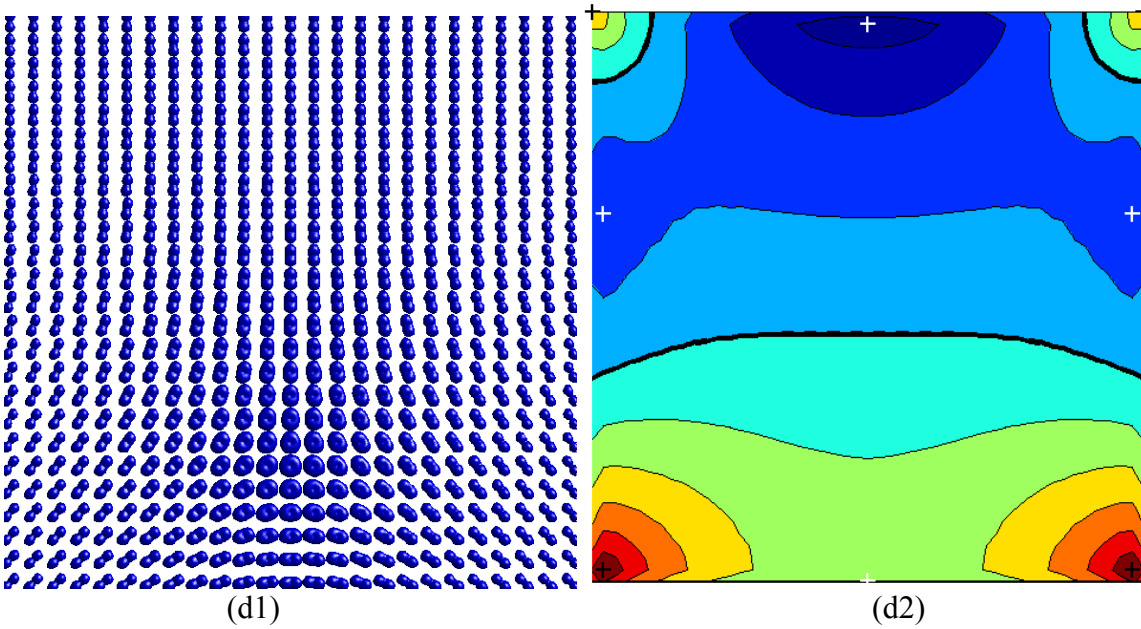
The synthetic tensor data (a1) only has one source, which is divergent from the critical point



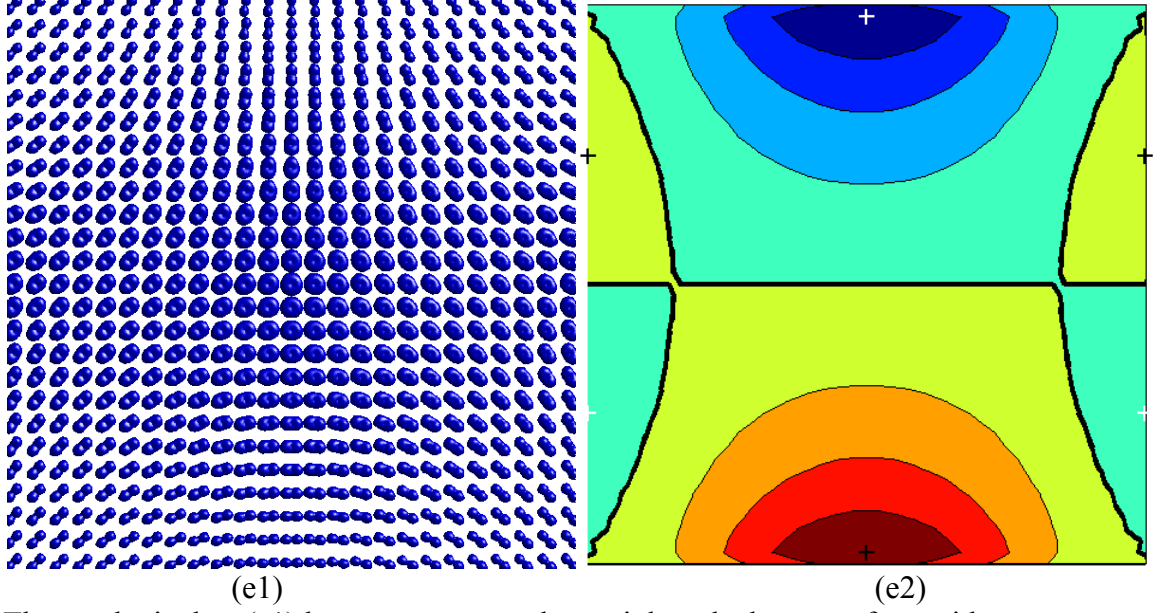
The synthetic data (b1) only has one vortex, which is rotation at the critical point



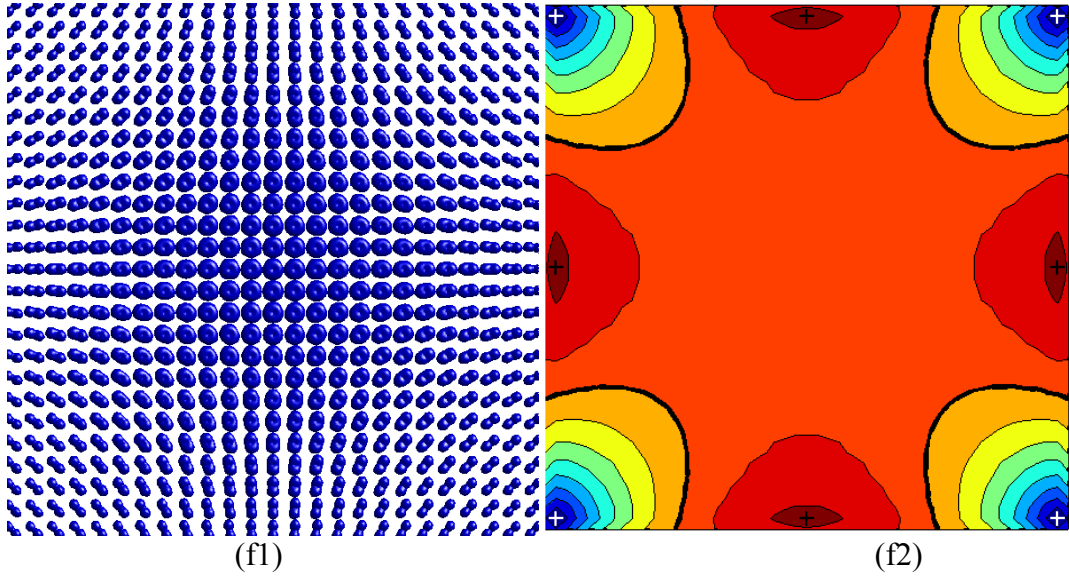
The synthetic data (c1) has one source, one sink, and two vortices,



The synthetic tensor data (d1) has two sources and two vortices at the four corners.



The synthetic data (e1) has one source and one sink at the bottom of two sides individually.



The synthetic tensor data (f1) has four sources at the bottom of four sides and four vortices at the four corners.

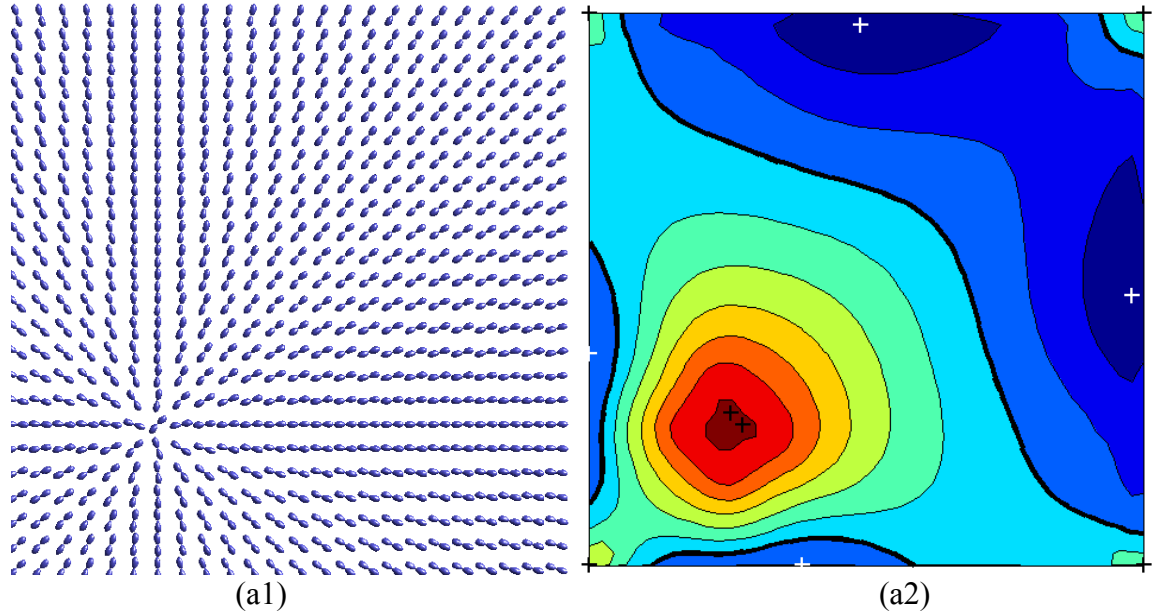
Fig. 7.5. The left side images are various 2^{nd} order synthetic tensor fields, and the right side images are the result of the topological visualization of left side tensor data

The magnitude of the residual after least-squares fitting to quantify how well the decomposition fits the input data since the harmonic part, H , is defined as that residual.

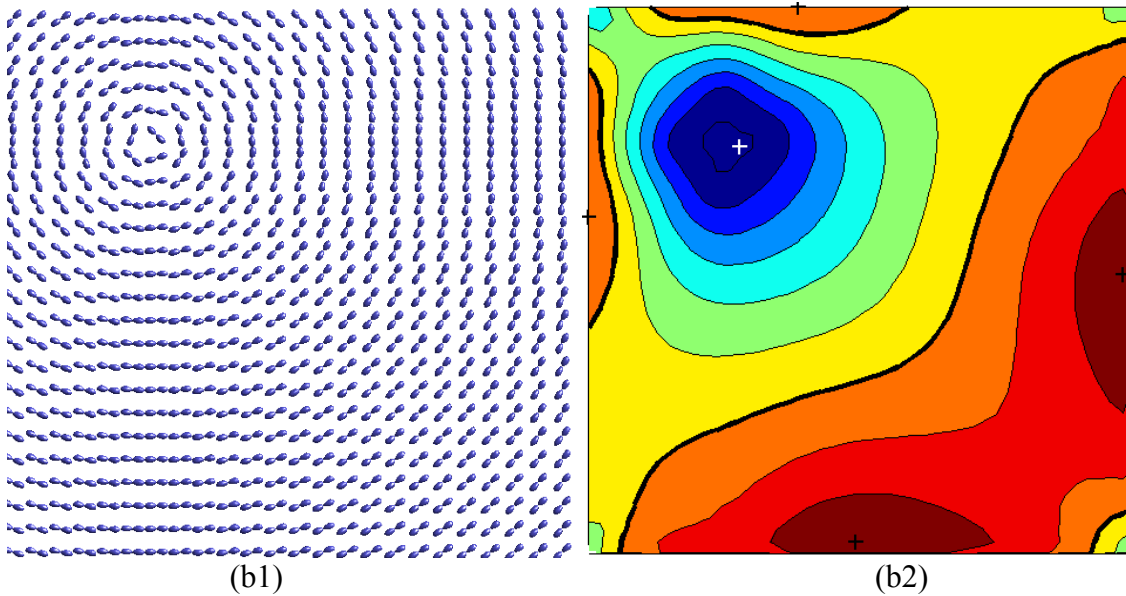
As a result we have zero fitting error. Instead, we analyze the quality of the fit by examining how irrotational $\nabla\phi$ is, how solenoidal $\nabla\psi$ is, and harmonic H . Ideally all of the quantities in the table below should be zero. This compare and contrast methods have been applied in the 2nd order and 4th order tensor fields.

D	$\frac{ div(curl(\psi)) }{ div(D) }$	$\frac{ curl(grad(\phi)) }{ curl(D) }$	$\frac{ div(H) }{ div(D) }$	$\frac{ curl(H) }{ curl(D) }$
a	7.86E-017	8.66E-017	2.03E-015	1.99E-015
b	7.89E-017	8.42E-017	2.54E-015	2.61E-015
c	8.54E-017	8.63E-017	2.21E-015	2.39E-015
d	5.07E-017	4.71E-017	9.67E-015	1.05E-014
e	5.50E-017	4.65E-017	1.01E-014	1.02E-014
f	5.48E-017	4.82E-017	7.81E-015	7.81E-015

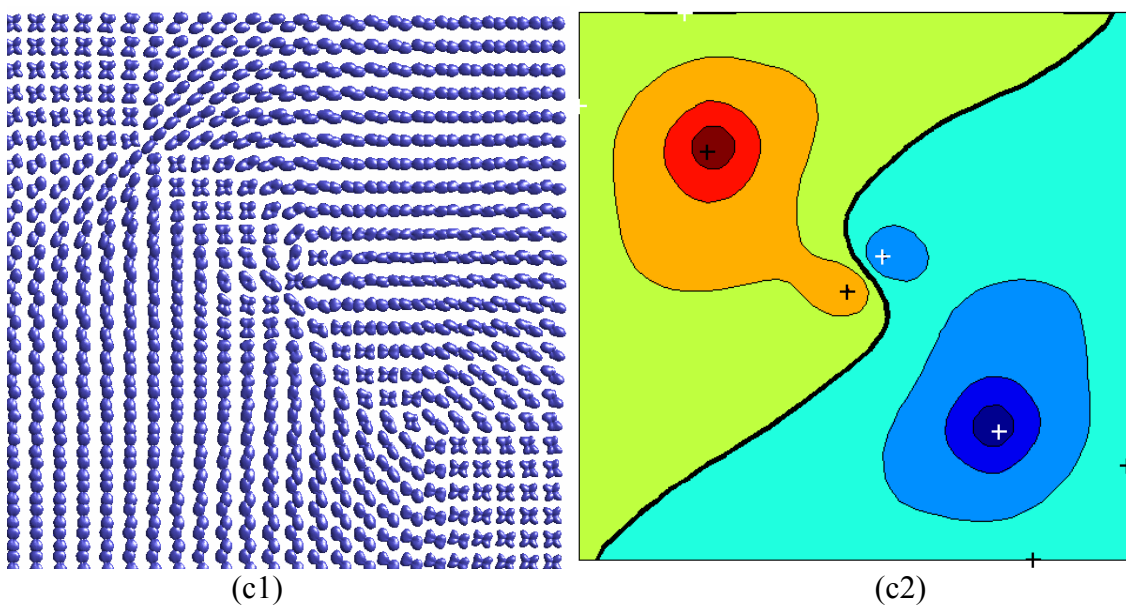
Table 7.1. Comparison of the various 2nd order synthetic tensor fields



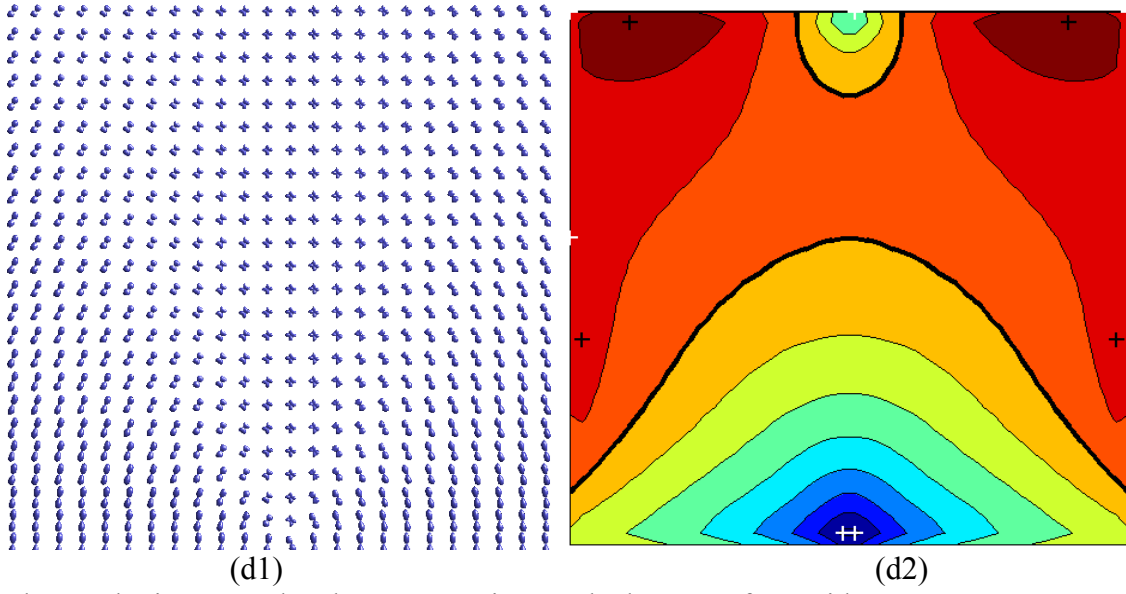
The synthetic tensor data (a1) has two source points



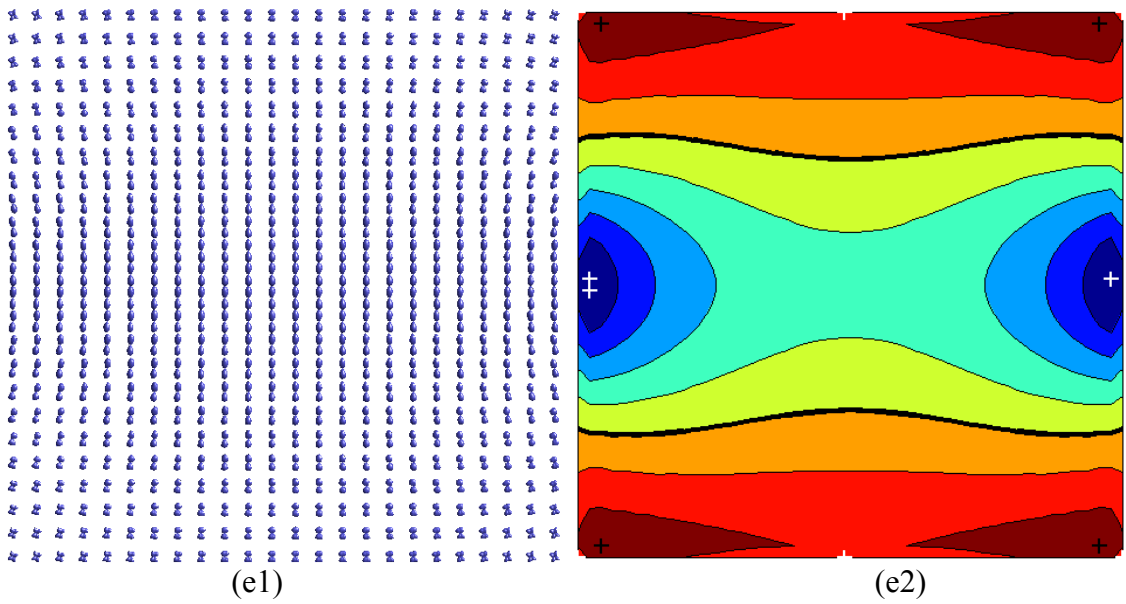
The synthetic tensor data (b1) has one vortex



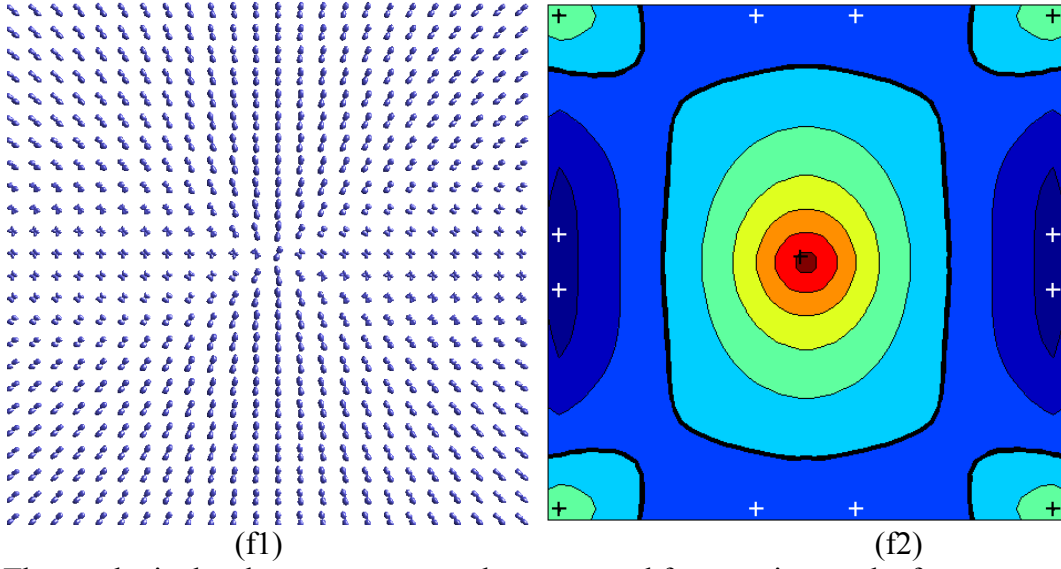
The synthetic tensor data (c1) has one source, one sink and two vortices



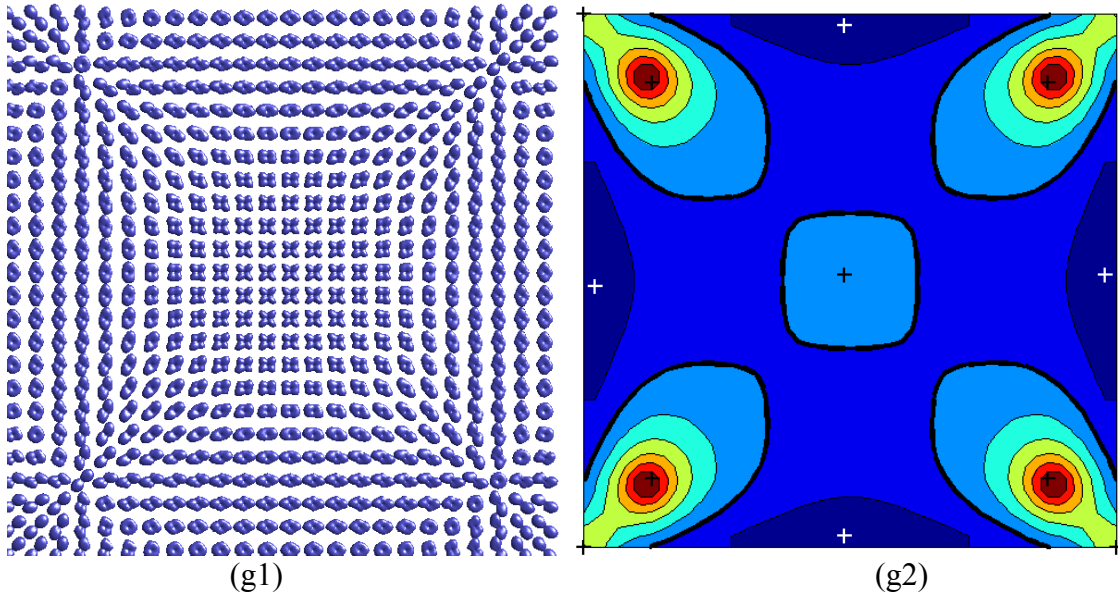
The synthetic tensor data has two vortices at the bottom of one side



The synthetic tensor data (e1) has two vortices at the bottom of two sides individually



The synthetic data has one source at the center and four vortices at the four corners



The synthetic tensor data (g1) has four sources at the four corners and one vortex at the center.

Fig. 7.6. The left side images are various 4th order synthetic tensor fields, and the right side images are the result of the topological visualization of left side tensor data

For the 4th order tensors we assess fitting quality in terms of the magnitude of the divergence and curl where the magnitude is taken to be the Frobenius norm of the tensor.

D	$\frac{ div(curl(\psi)) }{ div(D) }$	$\frac{ curl(grad(\phi)) }{ curl(D) }$	$\frac{ div(H) }{ div(D) }$	$\frac{ curl(H) }{ curl(D) }$
a	5.19E-008	3.24E-008	9.56E-007	9.59E-007
b	3.24E-008	5.19E-008	9.58E-007	9.56E-007
c	3.88E-008	3.63E-008	1.04E-006	1.02E-006
d	3.22E-008	4.53E-008	1.15E-006	9.34E-007
e	3.60E-008	5.98E-008	1.39E-006	1.11E-006
f	5.26E-008	4.09E-008	7.76E-007	1.13E-006
g	4.60E-008	2.13E-008	5.58E-007	7.10E-007

Table 7.2. Comparison of the various 4th order synthetic tensor fields

The results of the signed error function given by Equation (6.1) are shown in Figure (7.5, 7.6) as a filled contour plot. Note that the 'hot' (red, orange, yellow) colored regions represent the solenoidal part of the field and the 'cool' (blue, green) colors represent the irrotational part of the field. The smaller contour curves encircle the critical point in the field. The contour at $e = 0$ was used to successfully segment the field into two regions, and separate the nearby critical points in the center of the field. This function can be seen as a simple classifier for the critical points, separating extrema of divergence from extrema of curl. The $e = 0$ isocontour should not be interpreted as a hyperstreamline, but instead as the boundary between solenoidal and irrotational regions in the field. Black crosses denote local maxima in divergence magnitude (centers of sources / sinks), and white crosses denote local maxima in curl (centers of vortices). Several synthetic 2nd order and 4th order tensor datasets are shown above, as well as the generated response of the topological visualization based on the Helmholtz decomposition.

From the above images, we can found the critical points appear in original data image are not visible and clear, however it is very visible in the results of topological visualization based on the Helmholtz decomposition method both in the 2nd order and 4th order tensor data image.. Compared the results of the 2nd order and 4th order tensor data, we can obtain

that the results 4th order tensor data provide more detailed local information than those of 2nd order tensor data.

Both of the above tables reflect a high quality of fit. For all datasets the decomposition obeys the expected Helmholtz properties. The divergence of the solenoidal part is many orders of magnitude smaller than the divergence of the input field. Likewise, the curl of the irrotational part of the field is many orders of magnitude smaller than the input field. The resulting harmonic part has very small divergence and curl. The same observations hold when considering numerous randomly generated fields. The variance of the fitting parameters is very small, suggesting that the decomposition has consistent behavior over a large number of fields.

The Helmholtz decomposition is not unique. We may add any constant tensor field to $\text{curl}(\psi)$ or $\text{grad}(\phi)$, and subtract that field from H , and obtain new fields which satisfy the conditions of the decomposition.

The decomposition also does not preserve positivity, however we can make $\text{curl}(\psi)$ or $\text{grad}(\phi)$ positive by adding an isotropic tensor field to each, and subtracting the isotropic tensor fields from H .

7.4. Experiment 4: generating 100 2nd order and 4th order random tensor fields

We can also assess the fitting quality by generating 100 2nd order and 4th order random fields individually, with uniformly distributed tensor components in the range $[-1, 1]$. The mean and variance of the each fitting parameter is shown in the table below.

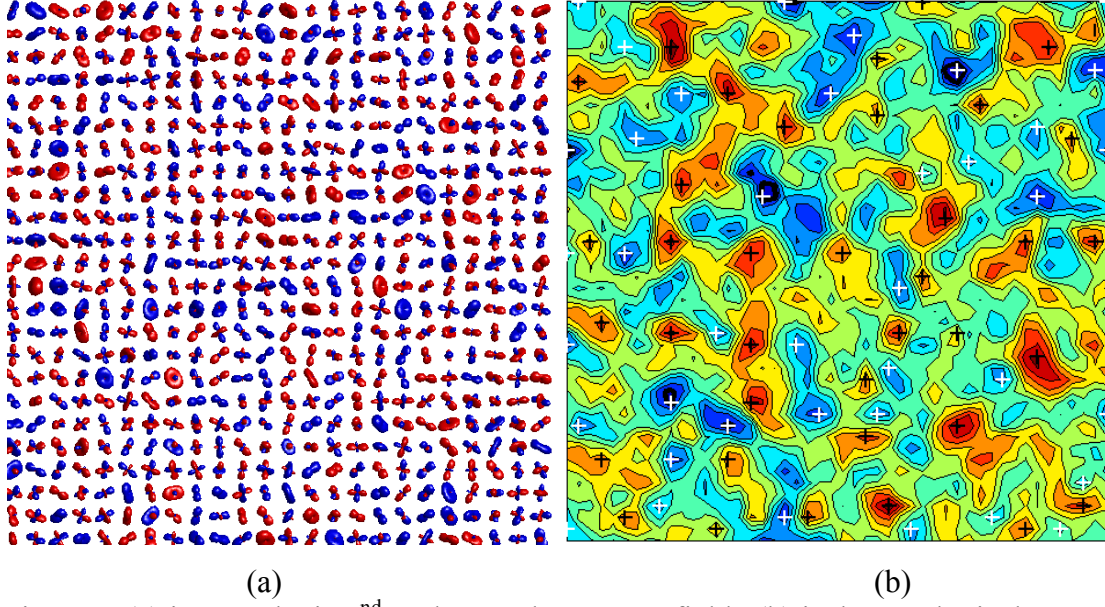


Fig. 7.7. (a) is a synthetic 2nd order random tensor field, (b) is the topological visualization of the left side 2nd order random tensor field.

D	$\frac{ div(curl(\psi)) }{ div(D) }$	$\frac{ curl(grad(\phi)) }{ curl(D) }$	$\frac{ div(H) }{ div(D) }$	$\frac{ curl(H) }{ curl(D) }$
mean	5.73E-017	3.09E-017	6.07E-016	5.11E-016
variance	7.12E-038	5.46E-038	4.23E-035	3.99E-035
max	5.77E-017	3.11E-017	6.17E-016	5.21E-016

Table 7.3. The comparison of the mean, variance, and max of the 2nd order random tensor fields

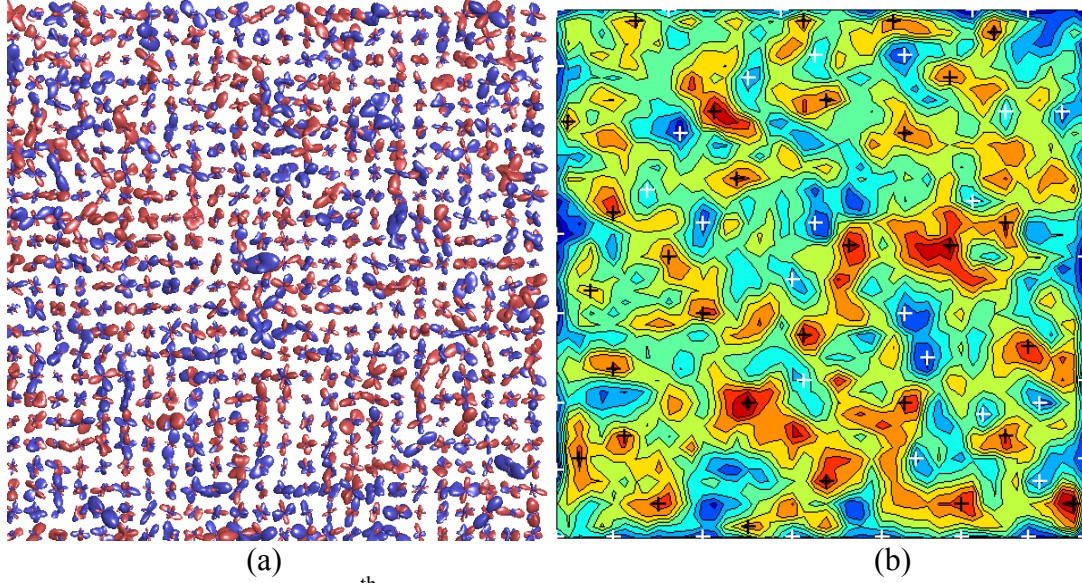


Fig. 7.8. (a) is a synthetic 4th order random tensor field, (b) is the topological visualization of left side 4th order random tensor field.

D	$\frac{ div(curl(\psi)) }{ div(D) }$	$\frac{ curl(grad(\phi)) }{ curl(D) }$	$\frac{ div(H) }{ div(D) }$	$\frac{ curl(H) }{ curl(D) }$
mean	2.66E-008	1.35E-008	1.69E-007	1.48E-007
variance	2.80E-021	5.11E-022	2.13E-019	8.02E-020
max	2.67E-008	1.35E-008	1.70E-007	1.48E-007

Table 7.4. The comparison of the men, variance, max of the 4th order random fields

It is apparent from these tables that the fitting quality of the generalized Helmholtz decomposition for the 4th order tensor fields are worse than the 2nd order tensor fields, but still show a great reduction in divergence and curl compared to the input data.

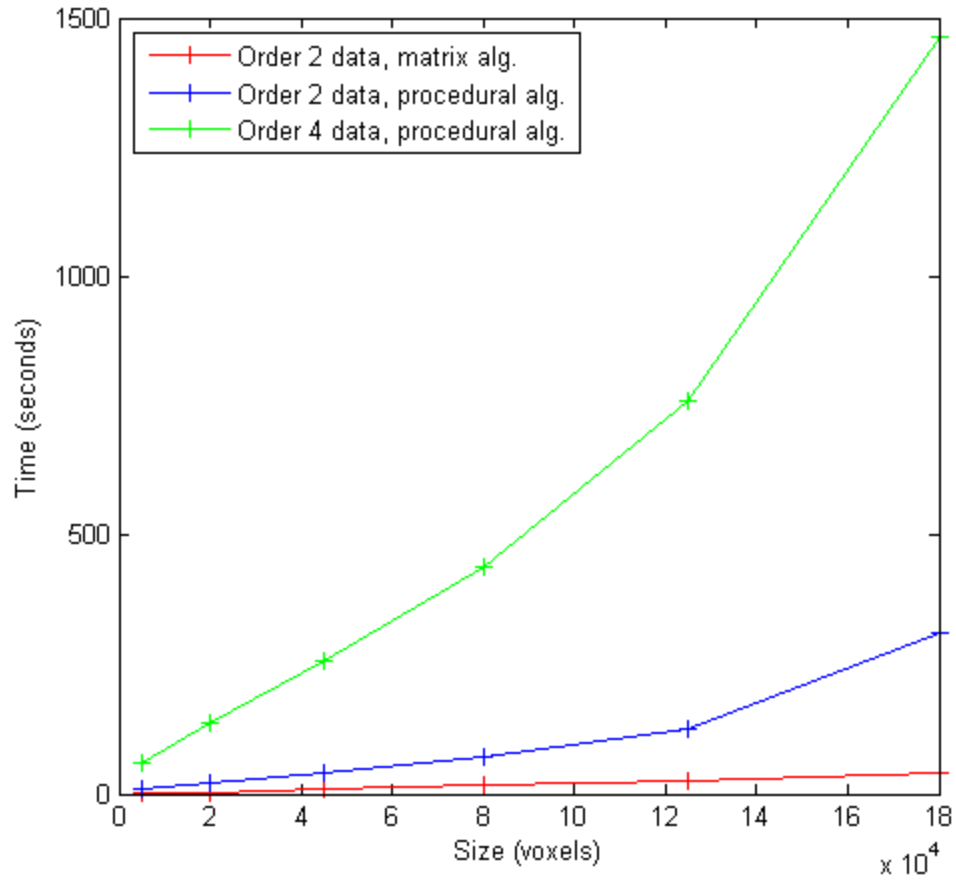


Fig. 7.9. The comparison of time consuming for different size and order tensor fields

A summary of timing results is presented in Figure (7.9). The red line shows the time to compute the decomposition of a second-order tensor field by forming the sparse matrices C and G in memory, and solving the normal equations using BiCG. The blue line shows the computation time for the same data when implementing matrix multiplications by C and G procedurally. For the 4th order tensor field we do not attempt to construct the matrices since they are so large.

CHAPTER 8: CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

Topological approaches attempt to reduce a dense set of input data to a simple representation of the structure of the field by using generalized Helmholtz Decomposition for tensor field visualization. The Helmholtz Decomposition of a tensor field on Cartesian grids allows a flow field to be separated into divergence-free, curl-free and harmonic components. The curl-free component is irrotational, so it is useful for isolating features such as local maxima and minima of divergence (foci of sources and sinks) in tensor fields without interference from curl-based features. Similarly, divergence-free component is solenoidal, and is useful for isolating centers of vortices in tensor field. The harmonic term is both solenoidal and irrotational, and the amplitude of harmonic term is typically small. Therefore, such methods can emphasize the locations of critical points (foci of sources and sinks, or the centers of vortices) and a few separating streamlines (separatrices) to generate a skeleton description of the field [11]. Such flow skeletons can be concise and provide intuitive representations of tensor fields. In contrast, direct visualization of the vector field may result in a great deal of visual clutter.

Topological feature extraction methods based on tensor field decomposition have been developed in this thesis. Decomposition splits the field into constituents with fundamentally different differential behavior, and permits the field to be segmented in terms of this behavior. The separating curve and local extrema of differential operators form a sparse skeleton representation of the field which still conveys much global structure. Topological methods for analyzing and visualizing tensor fields hold promise for simplifying these rich and complex datasets. Some developments in topological tensor field visualization have proceeded by generalizing the concepts of vector field topology. Degenerate points (in 2D tensor fields) and degenerate lines (in 3D tensor fields) have

commonly been defined in terms of eigenvectors of the tensors. Separatrices in the tensor case are hyperstreamlines, or integral curves of the eigenvector field.

Furthermore, our approach to topological tensor field visualization differs from previous approaches in several other ways. We do not depend on the computation of eigenvalues or eigenvectors. We do not trace deterministic or probabilistic hyperstreamlines. Instead we generate scalar fields based on the Helmholtz decomposition and render images which emphasize these features. This method does not need any preconditioner and has relatively few parameters.

We also generate a single separating contour which roughly partitions the tensor field into irrotational and solenoidal regions, which has the effect of classifying the critical points in each region. Even though some critical points are very close to each other, the signed error function still can effectively separate them. However, we do not address the problem of extracting the exact locations of critical points or lines.

8.2. Future Work

We recommend that future investigations attempt to develop more efficient large-scale optimization methods for decomposing higher order tensor fields. We also recommend investigating regularized decomposition methods which can also impose smoothness constraints. In analysis of DT-MRI it would also be interesting to compare the topological features of the diffusion tensor field with the topology of the tensor field representing fiber orientation distributions. We also wish to explore the potential field ϕ and stream field ψ to see if useful information can be extracted directly from them. Processing time for 4th order tensor fields is high, and future work should investigate exploiting symmetry to reduce the computational complexity of the problem. Moreover, we can try to involve exploiting symmetry to reduce the computational complexity of the problem.

BIOGRAPHY

- [1] T. McGraw, T. Kawai, I. Yassine, and L. Zhu. New Scalar Measures for Diffusion-Weighted MRI Visualization. Proceedings of the 5th International Symposium on Advances in Visual Computing: Part I, Las Vegas, Nevada, Section: ST: Visualization Enhanced Data Analysis for Health Applications, Pages: 934 - 943, 2009.
- [2] T. McGraw, and M. Nadar. Fast texture-based tensor field visualization for DR-MRI. United States Patent 7602180, 2007
- [3] R. A. Drebin, L. Carpenter, and P. Hanrahan. Volume Rendering. In *Computer Graphics, Proceedings of SIGGRAPH 88*, vol. 22, pp. 65–74, 1988.
- [4] C.-F. Westin, S. E. Maier, H. Mamata, A. Nabavi, F. A. Jolesz, and R. Kikinis. Processing and visualization of diffusion tensor MRI. *Medical Image Analysis*, vol. 6, no. 2, pp. 93–108, 2002.
- [5] M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, vol. 8, no. 3, pp. 29–37, 1988.
- [6] H. K. Tuy, and L. T. Tuy. Direct 2-D display of 3-D objects. *IEEE Computer Graphics and Applications*, vol. 4, no. 10, pp. 29–33, Oct. 1984.
- [7] A. Vilanova, S. Zhang, G. Kindlmann, and D. Laidlaw. An Introduction to Visualization of Diffusion Tensor Imaging and its Applications. *Visualization and Processing of Tensor Fields*, pp. 121-153, 2006.
- [8] T. McGraw, B.C. Vemuri, Y. Chen, M. Rao, and T. Mareci. DT-MRI Denoising and Neuronal Fiber Tracking. *Medical Image Analysis*, Pages 95-111, Volume 8, Issue 2, June 2004.
- [9] E. Özarslan, and T. Mareci. Generalized diffusion tensor imaging and analytical relationships between diffusion tensor imaging and high angular resolution diffusion imaging. *Magnetic Resonance in Medicine*, 50(5):955–965, 2003.
- [10] E. Özarslan, B. C. Vemuri, and T. H. Mareci. Generalized scalar measures for diffusion MRI using trace, variance and entropy. *Magnetic Resonance in Medicine*, 53(4): 866–876, 2005.

- [11] J. R. Shewchuk. An Introduction to the Conjugate Gradient Method without the Agonizing Pain. In *Technical report, School of Computer Science, Carnegie Mellon University*, Volume: CMU-CS-94-125, 1994.
- [12] M. R. Hestenes, and E. Stiefel. Methods of Conjugate Gradients for Solving Linear Systems. *Journal of Research of the National Bureau of Standards* 49 (6), December 1952.
- [13] C. Bajaj, V. Pascucci, and D. Schikore. The contour spectrum. In *Proceedings of the 8th conference on Visualization'97*. IEEE Computer Society Press Los Alamitos, CA, USA, 1997.
- [14] A. Barmpoutis, B. C. Vemuri, and J. R. Forder. Fast displacement probability profile approximation from HARDI using 4th-order tensors. *IEEE International Symposium on Biomedical Imaging*, pages 911-914, 14 -17 May 2008.
- [15] P. Basser, J. Mattiello, R. Turner, and D. Le Bihan. Diffusion tensor echoplanar imaging of human brain. In *Proceedings of the SMRM*, volume 56, pages 584–562, 1993.
- [16] P. Basser and C. Pierpaoli. Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor MRI. *Journal of Magnetic Resonance, Series B*, 111(3):209–219, 1996.
- [17] I. Beju and P. Teodorescu. Euclidean tensor calculus with applications. Taylor & Francis, 1983.
- [18] H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. In *Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*, pages 918–926. Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 2000.
- [19] G. Kindlmann, X. Tricoche, and C.-F. Westin. Anisotropy creases delineate white matter structure in diffusion tensor MRI. In *Ninth International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'06)*, Lecture Notes in Computer Science 4190, pages 126–133, Copenhagen, Denmark, October 2006.

- [20] T. Delmarcelle and L. Hesselink. The topology of symmetric, second-order tensor fields. In *Proceedings of the conference on Visualization '94*, pages 140–147. IEEE Computer Society Press Los Alamitos, CA, USA, 1994.
- [21] A. Gyulassy, P. Bremer, B. Hamann, and V. Pascucci. A Practical Approach to Morse-Smale Complex Computation: Scalability and Generality. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1619–1626, 2008.
- [22] Y. Shinagawa and T. Kunii. Constructing a Reeb graph automatically from cross sections. *IEEE Computer Graphics and Applications*, 11(6):44–51, 1991.
- [23] J. Helman and L. Hesselink. Visualization of vector field topology in fluid flows. *IEEE Computer Graphics and Applications*, 11(3):36–46, 1991.
- [24] L. Hesselink, Y. Levy, and Y. Lavin. The topology of symmetric, second order 3D tensor fields. *IEEE Transactions on Visualization and Computer Graphics*, 3(1):1–11, 1997.
- [25] T. Schultz, H. Theisel, and H.-P. Seidel. Topological visualization of brain diffusion MRI data. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1496–1503, 2007.
- [26] M. van Kreveld, R. van Oostrum, C. Bajaj, V. Pascucci, and D. Schikore. Contour trees and small seed sets for isosurface traversal. In *Proceedings of the thirteenth annual symposium on Computational geometry*, pages 212–220. ACM New York, NY, USA, 1997.
- [27] G. Kindlmann, X. Tricoche, and C.-F. Westin. Delineating white matter structure in diffusion tensor MRI with anisotropy creases. *Medical Image Analysis*, 11(5):492–502, October 2007.
- [28] S. Morita. Geometry of differential forms. *American Mathematical Society*, 2001.
- [29] Y. Tong, S. Lombeyda, A. Hirani, and M. Desbrun. Discrete multiscale vector field decomposition. *ACM Transactions on Graphics*, 22(3):445–452, 2003.
- [30] H. Li, W. Chen, and I. Shen. Segmentation of Discrete Vector Fields. *IEEE Transactions on Visualization and Computer Graphics*, pages 289–300, 2006.
- [31] X. Tricoche, G. Kindlmann, and C. Westin. Invariant Crease Lines for Topological and Structural Analysis of Tensor Fields. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1627–1634, 2008.

- [32] X. Zheng and A. Pang. Topological lines in 3D tensor fields. In *Proceedings of the conference on Visualization'04*, pages 313–320. IEEE Computer Society Washington, DC, USA, 2004.
- [33] X. Zheng, B. Parlett, and A. Pang. Topological lines in 3D tensor fields and discriminant Hessian factorization. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):395–407, 2005.
- [34] K. Polthier and E. Preuss. Variational approach to vector field decomposition. In *In Proc. Eurographics Workshop on Scientific Visualization*, pages 147–156. Springer Verlag, 2000.
- [35] K. Polthier and E. Preuss. Identifying vector fields singularities using a discrete hodge decomposition. *Visualization and Mathematics*, 3:113–134, 2003.
- [36] J. Heinbockel. Introduction to tensor calculus and continuum mechanics. Trafford Publishing, 2001.