Automated Feature Extraction in Oceanographic Visualization

by

Da Guo

Submitted to the Department of Ocean Engineering
in partial fulfillment of the requirements for the degrees of
Master of Science in Ocean Engineering
and
Master of Science in Electrical Engineering and Computer Science
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
February 2004

Author . . . .
Department of Ocean Engineering
December 22, 2003

Certified by . . . .
Nicholas M. Patrikalakis
Kawasaki Professor of Engineering
Thesis Supervisor

Certified by . . . .
Seth Teller
X-Consortium Associate Professor of Computer Science & Engineering
Thesis Reader

Certified by . . . .
Michael S. Triantafyllou
Chairman, Department Committee on Graduate Students

Certified by . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Automated Feature Extraction in Oceanographic Visualization

by

Da Guo

Submitted to the Department of Ocean Engineering
on December 22, 2003, in partial fulfillment of the
requirements for the degrees of
Master of Science in Ocean Engineering
and
Master of Science in Electrical Engineering and Computer Science

Abstract

The ocean is characterized by a multitude of powerful, sporadic biophysical dynamical events; scientific research has reached the stage that their interpretation and prediction is now becoming possible. Ocean prediction, analogous to atmospheric weather prediction but combining biological, chemical and physical features is able to help us understand the complex coupled physics, biology and acoustics of the ocean.

Applications of the prediction of the ocean environment include exploitation and management of marine resources, pollution control such as planning of maritime and naval operations. Given the vastness of ocean, it is essential for effective ocean prediction to employ adaptive sampling to best utilize the available sensor resources in order to minimize the forecast error. It is important to concentrate measurements to the regions where one can witness features of physical or biological significance in progress. Thus automated feature extraction in oceanographic visualization can facilitate adaptive sampling by presenting the physically relevant features directly to the operation planners. Moreover it could be used to help automate adaptive sampling.

Vortices (eddies and gyres) and upwelling, two typical and important features of the ocean, are studied. A variety of feature extraction methods are presented, and those more pertinent to this study are implemented, including derived field generation and attribute set extraction. Detection results are evaluated in terms of accuracy, computational efficiency, clarity and usability.

Vortices, a very important flow feature is the primary focus of this study. Several point-based and set-based vortex detection methods are reviewed. A set-based vortex core detection method based on geometric properties of vortices is applied to both classical vortex models and real ocean models. The direction spanning property, which is a geometric property, guides the detection of all the vortex core candidates, and the conjugate pair eigenvalue method is responsible for filtering out the false positives from the candidate set. Results show the new method to be analytically accurate and practically feasible, and superior to traditional point-based vortex detection methods.

Detection methods of streamlines are also discussed. Using the novel cross method or the winding angle method, closed streamlines around vortex cores can be detected. Therefore, the whole vortex area, i.e., the combination of vortex core and surrounding streamlines, is detected. Accuracy and feasibility are achieved through automated vortex detection requiring no human inspection. The detection of another ocean feature, upwelling, is also discussed.

Thesis Supervisor: Nicholas M. Patrikalakis
Title: Kawasaki Professor of Engineering

Thesis Reader: Seth Teller
Title: X-Consortium Associate Professor of Computer Science & Engineering

Acknowledgments

I would like to express my heartfelt thanks to the following people for their support in this research project:

Professor Nicholas M. Patrikalakis, for his expert advice on my research, instructive guidance on my academic study and words of encouragement. His influence goes beyond this study – he helped me to shape both my academic and life perspectives. Truly the best mentor one can ever imagine.

Professor Seth Teller, for taking time to read my thesis and providing valuable feedback.

Dr. Constantinos Evangelinos, for his acute academic insights and continuous guidance in my research work. His devotion to research work is contagious to me and my labmates.

I am also grateful to Dr. P. Haley and Dr. P. F. J. Lermusiaux, who provided me HOPS data models and were generous and patient to answer my questions.

Special thanks go to Mr. Fred Baker, for providing a stable hardware environment for my thesis work, and to Dr. Qiang Zhu, for his valuables perspectives regarding vortex detection.

I also want to thank my fiancee, Kun, for her love, understanding and support during my study at MIT; and to my friends, including Dr. Kwang Hee Ko and Dr. Yingbin Bao, many thanks for their valuable ideas and suggestions.

Lastly, my parents, who have been offering me unyielding support as always – to them this work is dedicated.

This work was funded in part from NSF/ITR under grant No. EIA-0121263 and NOAA/Sea Grant under grant No. NA86RG0074.

Contents

1 Introduction **21**

1.1 Poseidon Project 21

1.2 Harvard Ocean Prediction System (HOPS) 22

1.2.1 Harvard Primitive Equation Model 23

1.2.2 Grid Types 26

1.3 Real Time Regional Forecasting 31

1.4 Adaptive Sampling 33

1.5 Problem Statement 34

1.6 Thesis Outline 35

2 Feature Extraction and Visualization **37**

2.1 Feature Extraction 37

2.1.1 Oceanographic Features 38

2.2 Scientific Visualization 39

2.2.1 Volume Visualization Techniques 39

2.2.2 Scientific Dataset Format 41

2.2.3 Visualization Pipeline 42

2.2.4 HOPS Visualization Pipeline 43

2.2.5 Feature-Based Visualization 45

3 Vortex Detection **47**

3.1 Vortices in Fluid Flow 47

3.1.1 Definition 47

3.1.2 Classical 2D Vortices 48

3.1.3 Other 3D Vortices 52

3.2 Point-Based Vortex Detection Methods 54

3.2.1 Pressure Magnitude 54

3.2.2 Helicity 54

3.2.3 Vorticity Magnitude 55

3.2.4 Conjugate Pair Eigenvalues 55

3.2.5 Negative Eigenvalue λ_2 56

3.2.6 Positive Second Invariant of Jacobian 56

3.2.7 Conclusions on Point-based Methods 57

3.3 Set-Based Vortex Detection Methods 58

3.3.1 Curvature Center Density (CCD) 58

3.3.2 Winding Angle 59

3.3.3 Conclusions on Set-Based Methods 60

3.4 A Geometric-Based Vortex Core Detection Method 60

3.4.1 Direction Labeling 61

3.4.2 2D Algorithm 61

3.4.3 Post-Processing – Cleanup 62

3.4.4 Advantages of this Method 62

3.4.5 Classical Model Results 63

3.4.6 Real Ocean Model Results 67

3.4.7 Conclusions on the Vortex Core Detection Method 109

3.5 Detection of Streamlines and Closed Streamlines 109

3.5.1 Streamline Detection in Flows 109

3.5.2 Closed Streamline Detection 113

3.6 A Novel Cross Method to Detect Closed Streamlines 118

3.6.1 Cross Property 118

3.6.2 Distance Criterion 119

3.6.3 2D Algorithm 121

3.6.4 3D Algorithm 122

3.6.5 Quantization of Range and Axis 124
3.6.6 Classical Model Results 125
3.6.7 Real Ocean Model Results 125
3.6.8 Complexity Analysis 129
3.6.9 Conclusion on the “Cross” Method 131

4 Detection of Upwelling 133

4.1 Algorithm on Detecting Upwelling 134
4.2 Real Ocean Model Results 136

5 Conclusions 139

5.1 Conclusions 139
5.2 Future Work 140

List of Figures

1-1 System schematic of the HOPS. Chl, and PAR stand for chlorophyll, and Photosynthetically Active Radiation. 24

1-2 Horizontal grids in HOPS: Cartesian, spherical and rotated spherical. 28

1-3 Vertical grids in HOPS: (a)Single σ coordinates. (b-c) Hybrid coordinates. (d-e) Double σ coordinates. 30

1-4 (a) Horizontal layout. (b) Vertical layout. 31

3-1 Tangential velocity V_θ in Rankine vortex as a function of distance to the rotation axis 49

3-2 tangential velocity V_θ in potential vortex as a function of distance to the rotation axis 50

3-3 Tangential velocity V_θ in Oseen vortex as a function of distance to the rotation axis 51

3-4 Tangential velocity V_θ in Taylor vortex as a function of distance to the rotation axis 52

3-5 Streamlines with centers of curvature, adapted from [50]. 58

3-6 The winding angle α_w of a piecewise linear curve, adapted from [50]. . 59

3-7 The four equally spaced direction ranges correspond to direction labeling of the rectangular cell 61

3-8 The two grid points satisfy the direction spanning property. 62

3-9 Streamlines of Rankine Vortex 67

3-10 The horizontal velocity magnitude in Massachusetts Bay. 70

3-11 The vertical velocity magnitude in Massachusetts Bay. 71

3-12 The result of the vorticity magnitude method with superposed velocity vectors in Massachusetts Bay. 72

3-13 The boolean values of the vorticity magnitude method with the threshold "mean of vorticity field" with superposed velocity vectors in Massachusetts Bay. The deeper color shows the vortex area. 73

3-14 The boolean values of the vorticity magnitude method with the threshold "1.5×mean of vorticity field" with superposed velocity vectors in Massachusetts Bay. The deeper color shows the vortex area. 74

3-15 The boolean values of the vorticity magnitude method with the threshold "2×mean of vorticity field" with superposed velocity vectors in Massachusetts Bay. The deeper color shows the vortex area. 75

3-16 The boolean values of the vorticity magnitude method after the threshold "3×mean of vorticity field" with superposed velocity vectors in Massachusetts Bay. The deeper color shows the vortex area. 76

3-17 The result of the conjugate pair eigenvalues method in Massachusetts Bay. The deeper color shows the vortex area. 77

3-18 The result of the negative λ_2 method with the threshold "0" in Massachusetts Bay. The deeper color shows the vortex area. 78

3-19 The result of the negative λ_2 method with the threshold "-100" in Massachusetts Bay. The deeper color shows the vortex area. 79

3-20 The result of the negative λ_2 method with the threshold "-500" in Massachusetts Bay. The deeper color shows the vortex area. 80

3-21 The result of the negative λ_2 method with the threshold "-1000" in Massachusetts Bay. The deeper color shows the vortex area. 81

3-22 The result of the negative λ_2 method with the threshold "-5000" in Massachusetts Bay. The deeper color shows the vortex area. 82

3-23 The result of the positive Q method with the threshold "0" in Massachusetts Bay. The deeper color shows the vortex area. 83

3-24 The result of the positive Q method with the threshold "100" in Massachusetts Bay. The deeper color shows the vortex area. 84

3-25 The result of the positive Q method with the threshold "1000" in Massachusetts Bay. The deeper color shows the vortex area. 85

3-26 The result of the positive Q method with the threshold "5000" in Massachusetts Bay. The deeper color shows the vortex area. 86

3-27 The result of the positive Q method with the threshold "10^8" in Massachusetts Bay. The deeper color shows the vortex area. 87

3-28 The result of the geometric method without cleanup in Massachusetts Bay. 88

3-29 The result of the geometric method with the cleanup condition "conjugate pair eigenvalues" (condition A) in Massachusetts Bay. See next Figure for selected closeups of A, B, C, D, and E. 89

3-30 The closeups of last Figure. 90

3-31 The result of the geometric method with the cleanup condition "negative λ_2" (condition B) in Massachusetts Bay. 91

3-32 The result of the geometric method with the cleanup condition "positive Q" (condition C) in Massachusetts Bay. 92

3-33 The result of the geometric method with the cleanup condition "$A \cap B$" in Massachusetts Bay. 93

3-34 The result of the geometric method with the cleanup condition "$A \cap C$" in Massachusetts Bay. 94

3-35 The result of the geometric method with the cleanup condition "$B \cap C$" in Massachusetts Bay. 95

3-36 The result of the geometric method with the cleanup condition "$A \cup B \cup C$" in Massachusetts Bay. 96

3-37 The result of the geometric method with the cleanup condition "$A \cap B \cap C$" in Massachusetts Bay. 97

3-38 The result of the geometric method with the cleanup condition "$A \cap (B \cup C)$" in Massachusetts Bay. 98

3-39 The HOPS horizontal velocity magnitude and direction field in the Western Mediterranean. 101

3-40 The Boolean values of the vorticity magnitude with the threshold of twice the mean of the vorticity field, with superposed velocity vectors in the Western Mediterranean. The deeper color denotes the vortex area. 102

3-41 The result of the conjugate pair eigenvalues method in the Western Mediterranean. The deeper color shows the vortex area. 103

3-42 The result of the negative λ_2 method with superposed velocity in the Western Mediterranean. 104

3-43 The result of the positive Q method with superposed velocity in the Western Mediterranean. 105

3-44 The result of the geometric method without cleanup in the Western Mediterranean. 106

3-45 The result of the geometric method with cleanup in the Western Mediterranean. See next Figure for closeups of A, B, C and D. 107

3-46 The closeups of last Figure. 108

3-47 Inverse distance weighting 111

3-48 Transformation between *P-space* and *C-space*, adapted from [51]. . . . 112

3-49 The shortest distance (red) is the line where seed points are located. . 114

3-50 A streamline approaching a limit cycle has to reenter cells, taken from [61]. 116

3-51 If a real exit can be reached, the streamline will leave the cell cycle, taken from [61]. 116

3-52 If no real exit can be reached, the streamline will approach a limit cycle, taken from [61]. 117

3-53 In a vortex core, four axes can be drawn which are parallel to domain boundaries. 119

3-54 Four possible situations of a streamline. 120

3-55 The streamline E could not "bypass" vortex core O if O is the only vortex core in the region. 121

3-56 Streamline A is a closed streamline but streamline B is not. 121

3-57 A superposition of a source or sink and a vortex is not a perfect vortex. 122

3-58 Upwelling in a confined vortex, taken from [3]. 123
3-59 Swirling plane and core direction in 3D vortex core region, adapted from [25]. 123
3-60 Quantization of range and axis. 124
3-61 The result of winding angle method in the Western Mediterranean. . 126
3-62 The result of the "cross" method in the Western Mediterranean. . . . 127
3-63 The result of the "cross" method in the Western Mediterranean. . . . 128

4-1 Coastal upwelling 134
4-2 Equator upwelling 135
4-3 Upwelling in Massachusetts Bay. 137
4-4 Upwelling in Massachusetts Bay. 138

List of Tables

3.1 Timings of vortex detection methods. 131

List of Symbols

∇_{xy}	horizontal gradient operator
$\nabla_{xy}\cdot$	horizontal divergence operator
$\frac{\delta}{\delta t}$	material derivative
p	pressure
ρ	density
f	Coriolis parameter
τ	stress tensor
ν	kinematic viscosity
t	time
Γ	circulation
$\mathbf{v}$	velocity vector
V_r	radial (r) component of velocity
V_θ	tangential (θ) component of velocity
V_z	vertical (z) component of velocity
h	helicity
ω	vorticity
λ	eigenvalues
$\nabla\mathbf{v}$	velocity gradient tensor
J	Jacobian (velocity gradient tensor)
S	symmetric part of J
Ω	antisymmetric part of J
P, Q, R	three invariants of J

$\mathbf{x}$	position (x, y, z)
d	distance

Chapter 1

Introduction

1.1 Poseidon Project

Research on realistic nonlinear interdisciplinary processes has considerably expanded the knowledge of the complex coupled physics, biology and acoustics of the oceans. In order to effectively manage coastal marine resources and their impact on climate, powerful and sporadic biophysical events must be scientifically understood and predicted. To achieve this end, ocean forecasting, a process involving biological, chemical and physical aspects, is needed [36].

Essential to ocean forecasting is the continual melding of new observations into predictive dynamical models, and such observations must be obtained at the right time and space in the episodic ocean, thus calling for adaptive sampling. In spite of effective omission and aggregation efforts, the physical, biological and acoustical state variables and parameters remain large in number; this necessitates the evolution of the dynamical model parameters and structures during the prediction through adaptive modelling. The objective of the Poseidon project is to achieve efficient multiscale interdisciplinary ocean prediction with real-time objective adaptive sampling, assimilation of multiple streams of interdisciplinary data, and autonomous adaptive modelling through an effective union of information technology and ocean science.

Facilitated by the widely available, high-performance computing hardware, networking infrastructure, tools and interfaces for access, it is possible to construct a distributed ocean prediction computing system and using networked, scientific computing to effectively address a wide range of computationally intensive problems. The Poseidon project is developing an information system that can address the needs of interdisciplinary ocean scientists, making it possible to construct and execute complex domain-specific workflows in a simple and robust way.

The Poseidon project is based on a series of Observation System Simulation Experiments (OSSEs) with the Harvard Ocean Prediction System (HOPS) for three regions of the (near) coastal northwestern Atlantic which exhibit coastal upwelling, spring and tidal mixing plankton blooms, and the mesoscale eddy injection of nutrients into the upper ocean [36]. Acoustic tomographic data and acoustical and optical backscatter data are incorporated into the data streams for assimilation. As the system is developed and tested, the OSSEs are iterated from both ocean and information sciences approaches. The Poseidon project attempts to enhance scientific productivity, cultivate a new generation of interdisciplinary researchers, and contribute to the prediction and monitoring of marine resources in the multiuse coastal waters.

1.2 Harvard Ocean Prediction System (HOPS)

The Harvard Ocean Prediction System (HOPS) is a system of integrated software for multidisciplinary oceanographic research [63] [1]. It is designed to provide:

- "the naval ocean forecaster with accurate estimates of ocean fields in a timely and reliable manner";
- "the physical oceanography scientist with realistic simulations of the ocean in order to study fundamental dynamical mesoscale processes";
- "the acoustical oceanography scientist with tools to obtain reliable representations of the mesoscale sound speed variability";
- "the bio-geochemical oceanography scientist with an integrated environment to carry out coupled physical-bio-geochemical model simulations".

HOPS is configured as a nowcast/forecast system and is designed to meet shipboard operational needs requiring adaptability to diverse data streams and dominant physics in a given region. It is able to be exercised fully as a stand-alone system utilizing ship gathered information only. HOPS, a research and training tool for physical oceanography scientists, can facilitate the generation of physical fields suitable for process studies and allow efficient exploration and exploitation of new algorithms and ideas. The configuration of HOPS for interdisciplinary research serves as a laboratory for the investigation of the influence of the physical environment on the evolution of the oceanic bio-geochemical mass and sound propagation.

The distributed version of HOPS includes the software structured to work in an integrated fashion especially with regard to inputs and outputs. At the heart of HOPS is a primitive equation dynamical model supported by a series of routines, including data gridding routines, initialization and assimilation field preparation routines, visualization software, data preparation codes and topography conditioning software. The overall system schematic is shown in Figure 1-1 on page 24.

1.2.1 Harvard Primitive Equation Model

Models included in the Harvard Dynamical Modelling System are a Primitive Equation Model [52], a Quasi-Geostrophic Model [35] and a Surface Boundary Layer Model [58]. The emphasis here is placed on the the Harvard Primitive Equation Model for Coastal and Deep water, which is applicable to any oceanic region, with arbitrary coastline and/or open boundary segments. The GFDL (Geophysical Fluid Dynamics Laboratory) model offers the dynamical integration algorithm. The deep ocean version of the Harvard model was then developed by adding open boundary conditions, terrain following coordinates, and subgrid scale physics appropriate for mesoscale resolution [32].

HOPS can provide accurate computations in the deep ocean with steep topography, the shelf break and the shelf region.

Bryan [8] and Cox [12] both documented the GFDL algorithms, and Spall and Robinson [52] further expounded the details of the deep ocean model developments.

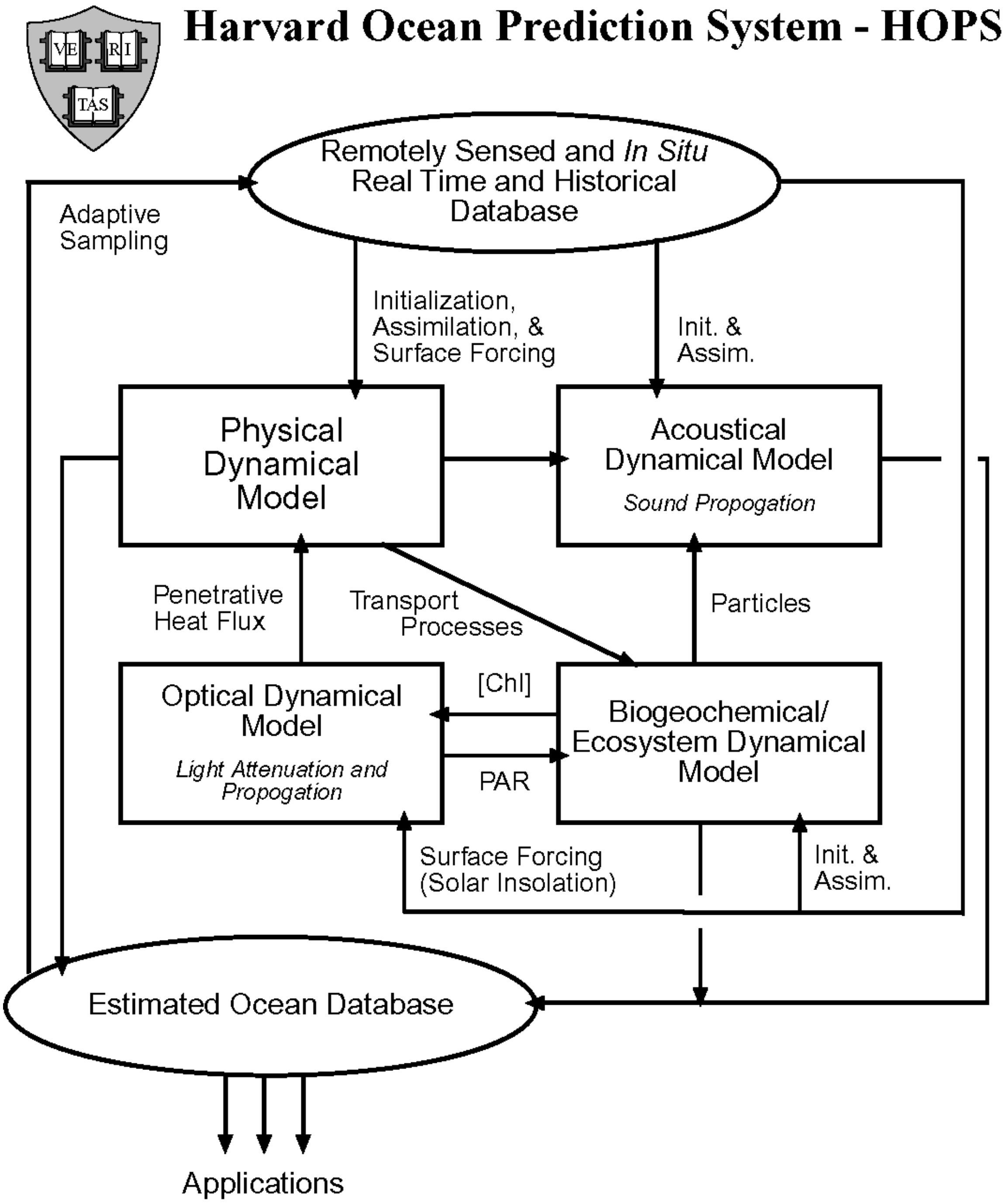

Harvard Ocean Prediction System - HOPS
VE RI TAS
Remotely Sensed and In Situ Real Time and Historical Database
Adaptive Sampling
Initialization, Assimilation, & Surface Forcing
Init. & Assim.
Physical Dynamical Model
Acoustical Dynamical Model
Sound Propogation
Penetrative Heat Flux
Transport Processes
Particles
Optical Dynamical Model
Light Attenuation and Propogation
[Chl]
PAR
Biogeochemical/ Ecosystem Dynamical Model
Surface Forcing (Solar Insolation)
Init. & Assim.
Estimated Ocean Database
Applications

Figure 1-1: System schematic of the HOPS. Chl, and PAR stand for chlorophyll, and Photosynthetically Active Radiation.

What follows is a description of recent developments. These and future developments are facilitated by modular codes managed under the UNIX source control system and code pre-processing options allowing for flexible support of diverse hardware platforms, physics and algorithm options.

Model Physics

According to [31], "the basis for the Primitive Equation Model is the Navier-Stokes equations in a rotating reference frame with the two main simplifying assumptions, appropriate for most deep ocean and shelf regions. One assumption is the Boussinesq approximation, which retains density variations only in the buoyancy terms. The other is the hydrostatic approximation. Further approximations based on dimensional analysis are included in the canonical primitive equations. In particular, the geopotentials are assumed to lie on concentric spheres; the gravitational field is taken to be constant throughout the domain; only the vertical component of the earth rotation vector is retained; the solid boundaries are assumed to be rigid and impermeable; and the heat and salt fluxes normal to the solid boundaries are assumed to be null. In the future, a free surface option will be provided."

With these assumptions, the canonical primitive equations are [37]:

$$\begin{aligned}
\nabla_{xy} \cdot \vec{u} + \tfrac{\partial w}{\partial z} &= 0 \\
\tfrac{\delta \vec{u}}{\delta t} + f\vec{e}_z \times \vec{u} &= -\nabla_{xy} p + Div\ \tau + \vec{F}_m \\
\tfrac{\partial p}{\partial z} &= -g\rho \\
\tfrac{\partial T}{\partial t} &= Div\ \vec{q}_T + F_T \\
\tfrac{\partial S}{\partial t} &= Div\ \vec{q}_S + F_S \\
\rho &= \rho(T, S, p)
\end{aligned} \tag{1.1}$$

where, Div is the three-dimensional divergence operator; $\nabla_{xy}\cdot$ is the horizonal divergence operator; ∇_{xy} is the horizontal gradient; $\vec{e}_z$ is a unit vector in the vertical direction; $\delta/\delta t$ is the material derivative; $\vec{u}$, w the horizontal and vertical velocities; p is the pressure; ρ is the density; f is the Coriolis parameter; τ is the stress tensor; S is the salinity; T is the temperature; $\vec{F}_m$, F_T and F_S are the chosen parameterizations for additional subgrid scale physics and $\vec{q}_T$, $\vec{q}_S$ are the heat and salinity flux vectors.

1.2.2 Grid Types

Horizontal Coordinates

There are three types of horizontal domains currently supported by HOPS: Cartesian, spherical and rotated spherical (Figure 1-2 on page 28) [32] [63].

1. Cartesian: The Cartesian grids are local tangent plane approximations to a sphere, often used for a small regions where spherical effects are relatively unimportant. The Cartesian transformation is defined by three parameters:

   (lat_c, lon_c): The "center" of the transformation. Geographic coordinates are converted to Cartesian based on the distance to this point.

   θ_{rot}: The rotation angle. The Cartesian coordinates, (x,y), are rotated from their geographic counterparts, (lat, lon), by the angle θ_{rot}.

   The Cartesian domains are further defined with six additional parameters:

   (nx, ny): The number of grid points in each direction.

   (dx, dy): The grid spacing in each direction.

   $(\Delta x, \Delta y)$: The offset (meters) between the center of the domain and the center of the transformation.

2. Spherical: The spherical grids are geographic latitudes/longitudes. They are often used where spherical effects are important and it is more convenient to keep geographic orientation. There is no transformation, so lat_c, lon_c and θ_{rot} are all zero. The domains are defined with six parameters:

   (nx, ny): The number of grid points in each direction.

   (dx, dy): The grid spacing in each direction.

   $(\Delta x, \Delta y)$: The offset (degrees) between the center of the domain and the center of the transformation. For the spherical grid, $(\Delta x, \Delta y)$ become the (lat, lon) of the center of the grid.

3. Rotated Spherical: A spherical grid in which the equator had been moved to reduce model grid distortions. It is often used for regions where spherical effects are important and reducing model grid distortions outweighs any benefit of maintaining geographic orientation. The rotated spherical transformation is defined by three parameters:

   (lat_c, lon_c): The "center" of the transformation, which is the (0,0) point of the rotated spherical system.

   θ_{rot}: The rotation angle. The angle between the rotated coordinates and the geographic coordinates at the"center" of the transformation.

   The rotated spherical domains are defined with six additional parameters:

   (nx, ny): The number of grid points in each direction.

   (dx, dy): The grid spacing in each direction.

   $(\Delta x, \Delta y)$ The offset (degrees) between the center of the domain and the center of the transformation.

Vertical Coordinates

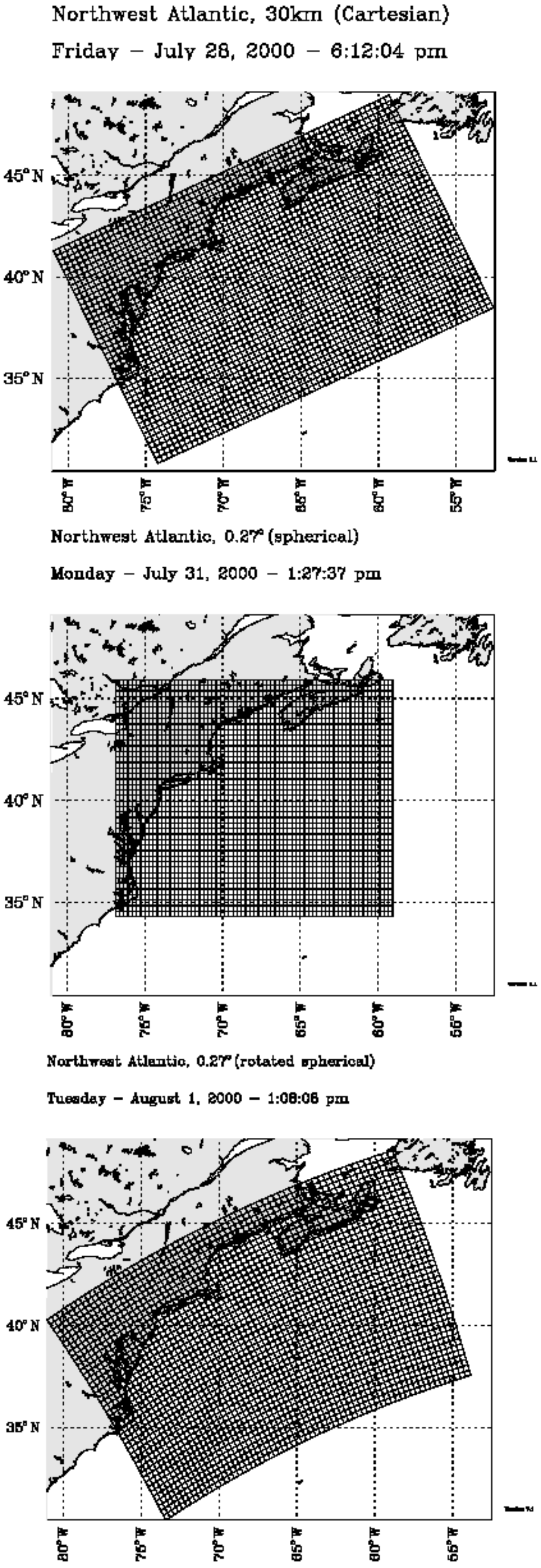
There are three types of vertical domains currently supported by HOPS: σ Coordinates, Hybrid Coordinates and Double σ Coordinates (as in Figure 1-3 on page 30).

1. σ Coordinates: The depth of a model is a fixed fraction of the total water column.

$$z_{i,j,k} = \sigma_k * h_{i,j} \tag{1.2}$$

   The σ coordinates are most useful in regions with high topography and where maintaining near surface resolution is not necessarily critical.

2. Hybrid Coordinates: A generalization of (1). The top k_c levels are flat, the remainders are terrain that follows,



Northwest Atlantic, 30km (Cartesian)

Friday – July 28, 2000 – 6:12:04 pm

Northwest Atlantic, 0.27° (spherical)

Monday – July 31, 2000 – 1:27:37 pm

Northwest Atlantic, 0.27° (rotated spherical)

Tuesday – August 1, 2000 – 1:08:06 pm

Figure 1-2: Horizontal grids in HOPS: Cartesian, spherical and rotated spherical.

$$z_{i,j,k} = \begin{cases} z_k & (k \leq k_c) \\ z_c + \sigma_k * (h_{i,j} - z_c) & (k > k_c) \end{cases} \tag{1.3}$$

Where z_c is the depth of the flat interface between the two vertical coordinate systems and the convention $0 \leq \sigma_k \leq 1$ is used.

The hybrid coordinates are often used in deep ocean where maintaining near surface resolution is important.

3. Double σ Coordinates: A generalization of (2). The constant coordinate interface, z_c, is replaced by one that varies in the horizontal to ride above topography. The variable coordinate interface, $f_{i,j}$, is given by:

$$f_{i,j} = \frac{z_{c_1} + z_{c_2}}{2} + \frac{z_{c_1} - z_{c_2}}{2} \tanh\{\frac{2\alpha}{z_{c_1} - z_{c_2}}[h_{i,j} - h_{ref}]\} \tag{1.4}$$

Where,

z_{c_1}: The shallow limit for the coordinate interface.

z_{c_2}: The deep limit for the coordinate interface.

h_{ref}: The“switching” depth. $f_{i,j}$ is halfway between z_{c_1} and z_{c_2} at $h_{i,j} = h_{ref}$.

α: The slope parameter. How fast f responds to changes in the topography, h. Also a limit on the slope of f, $\|\nabla f\| \leq \alpha\|\nabla h\|$.

The vertical system is: a σ system referenced between the surface and $f_{i,j}$ above a second σ system referenced between $f_{i,j}$ and the bottom.

$$z_{i,j,k} = \begin{cases} \sigma_k * f_{i,j} & (k \leq k_c) \\ f_{i,j} + (\sigma_k - 1) * (h_{i,j} - f_{i,j}) & (k > k_c) \end{cases} \tag{1.5}$$

where, the (arbitrary) convention $0 \leq \sigma_k \leq 1$ for $k \leq k_c$ and $1 \leq \sigma_k \leq 2$ for $k > k_c$ has been used. The double σ coordinates are most useful in regions with shallow topography and where it is important to maintain near surface resolution.

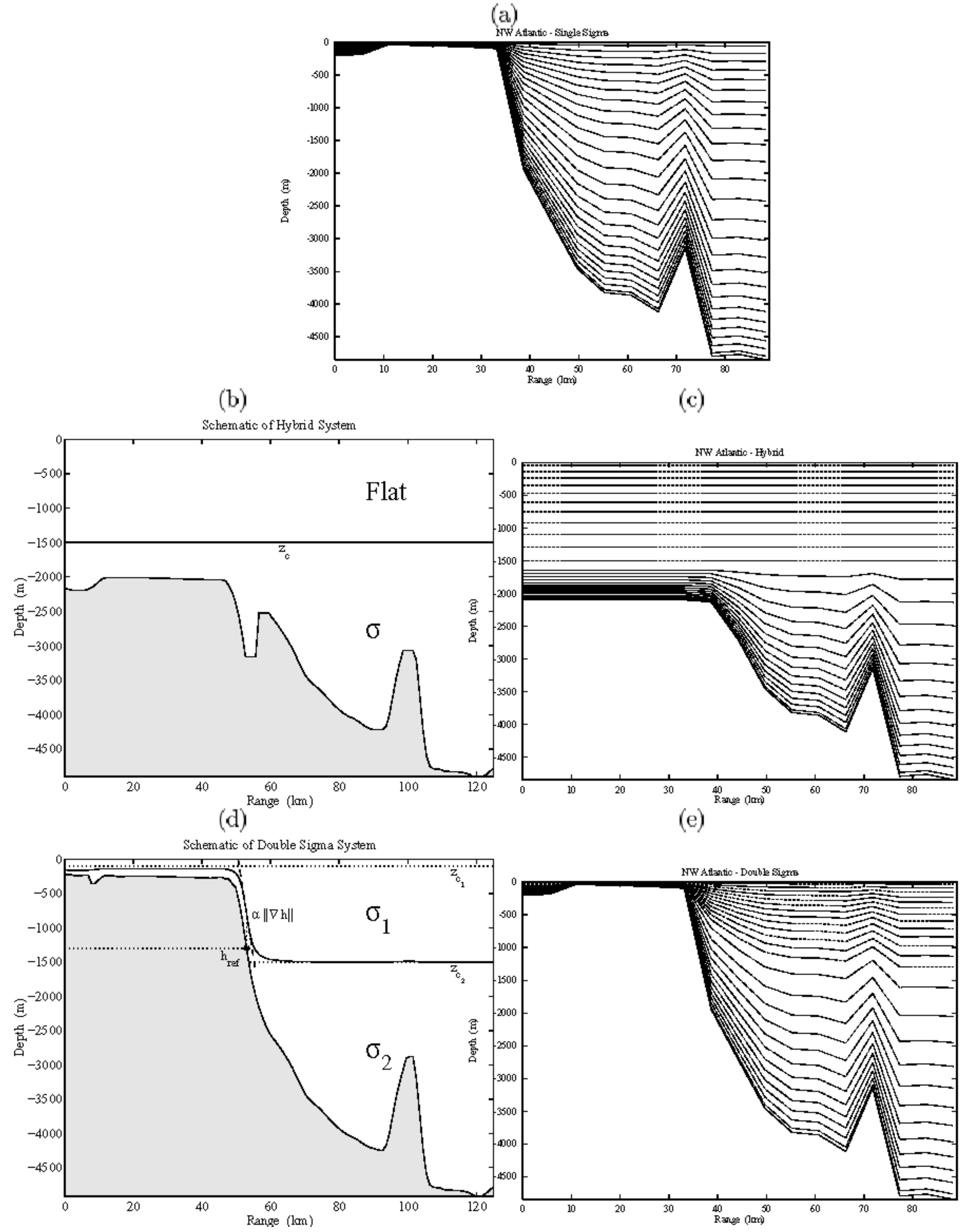


Figure 1-3: Vertical grids in HOPS: (a)Single σ coordinates. (b-c) Hybrid coordinates. (d-e) Double σ coordinates.

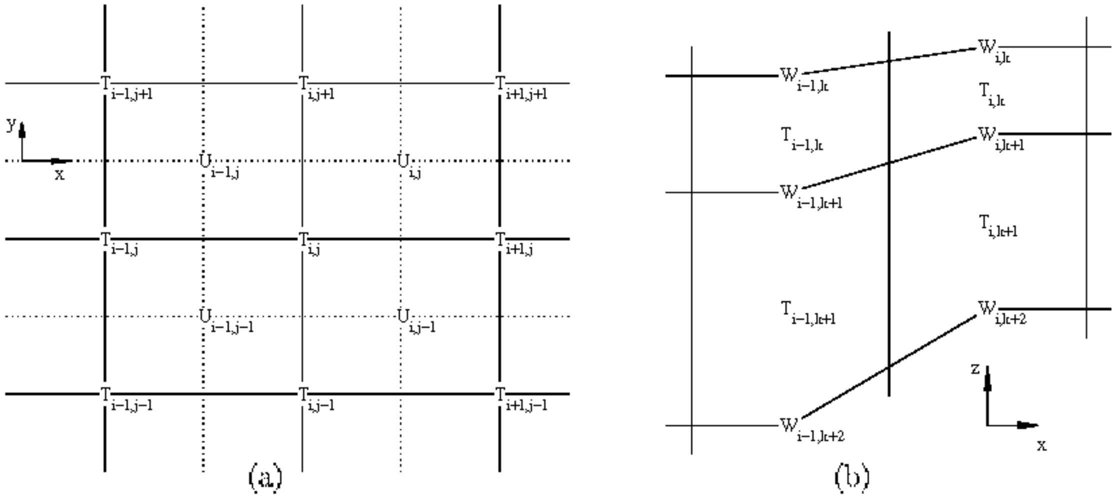


Figure 1-4: (a) Horizontal layout. (b) Vertical layout.

An example of horizontal and vertical layout of a grid is as in Figure 1-4 on page 31. In horizontal layout, T stands for tracers (temperature, salinity, etc) and streamfunctions. U stands for both horizontal velocity components. In vertical layout, Tracer vertical boxes are shown, velocity vertical boxes have the same layout, merely shifted 1/2 grid space horizontally. W stands for vertical velocities.

In this paper we present our initial approach to vortex detection in ocean flows, applying 2D algorithms to the flow at multiple constant depths. Such methods, while less comprehensive compared to their 3D counterparts, allow us to efficiently identify horizontal structures in the ocean. Especially in the case of stratified ocean flows (which are more prevalent during the summer months) feature extraction in the horizontal dimensions can be very useful. As an added benefit, they are considerably less expensive and less sensitive to the vertical grid irregularities of most ocean models than fully 3D algorithms.

1.3 Real Time Regional Forecasting

Being intermittent, eventful and episodic, the ocean is a turbulent fluid in essence and its circulation is characterized by a myriad of dynamic processes occurring over a vast range of nonlinearly interacting scales both in time and space. Ocean forecasting allows us to effectively and efficiently manage operations on and within the sea. Such forecasting has been used for military operations as well as scientific research [41].

Forecasting usually starts with observations that can initialize dynamical forecasting models; further observations are then assimilated into the models as the forecasts progress in time. As a matter of fact, such observations are usually difficult, costly and incomprehensive. It goes without saying that if a region of the ocean were to be sampled uniformly over an arbitrary space-time grid, only a small subset of those observations would have significant impact on the accuracy of the forecasts. In contrast, accurate forecasts target the intermittent energetic synoptic dynamical events and form the basis for designing a sampling scheme that can observe the ocean state. Adaptive sampling of features of greatest impact increases efficiency as it can reduce the observational requirements by several orders of magnitude.

The oceans evolve as time goes by, responding to both external surface and body forces, and internal dynamical processes. Some examples of external surface and body forces include tidal forces, winds and surface fluxes of heat and fresh water. Given the importance of air-sea interaction, ocean forecasts requires an accurate meteorological forecast. Analogous to atmospheric weather phenomena, ocean internal instabilities and resonances, including meanders of currents, fronts, eddies and wave propagation, are called the internal weather of the sea. Their spatial scales are short with respect to global scale, which requires the ocean forecasts to be carried out regionally as opposed to globally.

Furthermore, the regional forecast problem has additional forces appearing as fluxes through horizontal boundaries, a representation of both larger scales of direct forcing, remote internal dynamical events and land-sea interactions in the littoral zone. To develop a regional forecast system and capability, both the scales and processes of interest and the dominant scales and processes in the region are essential.

Some of the fields of forecast interest are physical, acoustical, biological, optical, chemical and sedimentological state variables. Examples include velocities, temperature, sound speed, irradiance, plankton concentrations, scattering profiles and suspended particle concentrations. Multi-disciplinary sampling schemes are constrained by the interdisciplinary compatibility requirements. Some variables are of direct interest whereas others are helpful in interdisciplinary field estimation such as the case of acoustic travel times in the estimation of temperature gradients. As a result, the scope of ocean forecasting has expanded, which in turn leads to rise of a challenging adaptive sampling problem, i.e., the design of sampling schemes for the acquisition of multi-scale compatible interdisciplinary data sets based on real time observations and realistic forecasts.

1.4 Adaptive Sampling

Adaptive sampling can help to collect the most useful data of an ocean field and help compute error forecasts through the use of quantitative criteria or goals. Current adaptive sampling methods [29] [44] [43] combine field and error forecasts with *a priori* experience to intuitively choose the future sampling. This approach can conceptually be then automated: quantitative adaptive sampling utilizes forecasts as inputs to mathematical criteria whose real-time optimization produces the sampling pattern [42].

Current forecast capabilities involve the future evolution of ocean fields, of their variabilities and uncertainty statistics [28]. Various constraints remain, including practical considerations (platforms and sensors availability, ship speeds, autonomous underwater vehicle (AUV) range, weather conditions), dynamical motives (search for precursors of the primary phenomenon, dynamical model verifications), and cost penalties (fuel, batteries, labor costs). Moreover, scientific and technical constraints exist for the measurement model to generate the actual state variables to be assimilated. Finally, in varied representations, several goals or criteria are also a possibility. One type of optimum can be the sampling that minimizes the forecast of the field error variance confined by cost penalties and practical constraints. Others belong to the best skill evaluation samplings that perfectly define specific properties of future dynamics, independent of past data and other constraints. Assimilation studies also use data assimilation criteria to develop the goal accordingly. Thus, the assimilation study aiming at minimizing the field error with a variance metric would set the adaptive sampling goal as determining the future sampling that minimizes the trace of the error covariance matrix.

For linear models, the optimization can be implemented independent of future data values, making use of the dynamical and measurement models and their uncertainties. For non-linear models, data values make a difference and a forecast OSSE needs to be done during the optimization process. Research on optimal control and estimation theory can be applied and extended by interdisciplinary oceanographers to obtain the most useful sampling methods.

The process of feature extraction discussed in this thesis facilitates user-assisted adaptive sampling and could eventually be used to help automate this optimization procedure.

1.5 Problem Statement

Given the vastness of ocean, it is essential for effective ocean prediction to employ adaptive sampling to best utilize the available sensor resources in order to minimize the forecast error. It is important to concentrate measurements to the regions where one can witness features of physical or biological significance in progress. Thus, automated feature extraction in oceanographic visualization can facilitate adaptive sampling by presenting the physically relevant features directly to the operation planners. Moreover it could be used to help automate adaptive sampling. Therefore, the study of automated feature extraction in oceanographic visualization is introduced.

A series of Observation System Simulation Experiments (OSSEs) with the Harvard Ocean Prediction System (HOPS) provide the basis, the large-scale HOPS datasets in different ocean regions, for this study. The task is to detect typical oceanographic features from these large-scale datasets.

Vortices (eddies and gyres) and upwelling, two typical and important features of the ocean, are studied. A variety of feature extraction methods are presented, and those more pertinent to this study are implemented, including derived field generation and attribute set extraction. Detection results are evaluated in terms of accuracy, computational efficiency, clarity and usability.

Vortices, a very important flow feature is the primary focus of this study. Several point-based and set-based vortex detection methods are reviewed. A set-based vortex core detection method based on geometric properties of vortices is applied to both classical vortex models and real ocean models. The direction spanning property, which is a geometric property, guides the detection of all the vortex core candidates, and the conjugate pair eigenvalue method is responsible for filtering out the false positives from the candidate set. Detection methods of streamlines are also discussed. Using the novel cross method or winding angle method, closed streamlines around vortex cores can be detected. Therefore, the entire vortex area, i.e., the combination of vortex core and surrounding streamlines, is detected. Accuracy and feasibility are achieved through automated vortex detection requiring no human inspection.

The detection of upwelling using integration methods is also discussed.

1.6 Thesis Outline

The organization of this thesis is as follows.

- Chapter 2 provides an overview of feature extraction and scientific visualization techniques.
- Chapter 3 discuss vortex detection. First, different vortex models are presented. Second, the point-based vortex detection methods and set-based vortex detection methods are reviewed. Finally, a geometric-based vortex core detection method and a novel “cross” method to detect closed streamlines are introduced.
- Chapter 4 describes upwelling detection.
- Chapter 5 provides conclusions and suggestions for future work.

‘

Chapter 2

Feature Extraction and Visualization

2.1 Feature Extraction

Features are regions of interest in a data set, be it a single grid location or a set of nodes from the data set. There are several methods available to extract the feature-sets from the data sets. For example, in the area of vortex identification, there are two categories of algorithms, point-based and set-based methods, which will be discussed later. The means of feature extraction are feature criteria, which depend on the scientific domain as well as the specific features of interest. For example, it could be a Boolean criterion function, consisting of a logical combination of scalar thresholds, applied to data values or to derived data values such as velocity gradients [53] [57].

Attribute sets are defined as sets of characteristic values computed from the extracted features in the data set. They can involve a combination of scalars, tensors and vectors. When a feature goes beyond a single grid position or cell, relevant attributes including volume, mean data value, center and various moments can be obtained by using volume integrals to do an integration over this region. Attributes fall into two categories, either purely geometric or involving a combination of geometry and the underlying data values.

In short, features are defined to be thresholded, connected regions. The following steps required to identify these regions:

1. Find the nodes which meet certain threshold criteria.
2. Partition the nodes into connected regions.
3. Extract the bounding surface of each connected region.

2.1.1 Oceanographic Features

There are many oceanographic features in the ocean, such as upwelling, fronts, jets/meanders, vortices, geostrophic turbulence etc. These oceanographic features are caused by combined rotation and stratification effects. For instance, coastal upwelling is a phenomenon resulting from wind forcing, Ekman dynamics and geostrophy [13].

A front is a relatively narrow region of intensified gradients of fluid properties between two main fluid masses. Under the action of Coriolis forces, the process of geostrophic adjustment is at work, leading to a relatively intense flow aligned with the front. The much weaker density gradients in the main part of each fluid mass confine the motion to the frontal region, and the flow exhibits the form of a jet. The jet has a velocity maximum, coinciding more or less with the location of the maximum density gradient, on both sides of which the velocity decays. Meanders and shed vortices can also exist in a jet area. Meanders on a jet do not remain stationary but propagate, usually downstream and rarely upstream. A vortex, or eddy, is defined as a closed circulation that is relatively persistent, which means that the turnaround time of a fluid particle embedded in the structure is shorter than the time during which the structure remains identifiable. When several eddies are present and are not too distant from each other, interactions are inevitable. Vortices shear and peel off the sides of their neighbors and, at times, merge to create larger vortices. The sheared elements either curl onto themselves, forming new, smaller vortices, or dissipate under the action of viscosity. The net result is a combination of consolidation and destruction. The state of many interacting vortices under the influence of Coriolis effects is geostrophic turbulence.

Research in detecting these features would help understanding the complex coupled physics, biology and acoustics of the oceans and could be used to help automate adaptive sampling.

2.2 Scientific Visualization

Visualization literally means making things visible. In general, it is concerned with providing visual images to understand something that can not be directly seen under normal circumstances. In the context of this paper, the meaning of visualization is confined to the visualization of data stored in a computer, the goal of which is to explore and understand data sets. As a subcategory of computer graphics, visualization refers to algorithms, programs, techniques and systems that produce images.

A review of the field of visualization in computer graphics can be found in [34]. Visualization could again be divided into two sub-fields − information visualization and scientific visualization. Information visualization deals with discrete data often with high dimensionality and depicts properties of a large but finite number of data items. Scientific visualization, on the other hand, is good at handling continuous data often with low dimensionality. Such data represents a continuous field, though in practice it is usually sampled on a discrete set of locations. The two subfields sometimes merge.

2.2.1 Volume Visualization Techniques

There are a great variety of volume visualization techniques able to probe 3D fields, and they differ on the type of field being investigated, be it a scalar, vector or tensor, and the type of information sought. Isosurface and volume rendering are two major 3D visualization methods to explore scalar volume datasets.

Isosurface

An isosurface is defined as a surface of a constant threshold within the dataset and contains points where $p(x, y, z) = \alpha$ for a particular threshold value α. To approximate an isosurface, polygonal primitives such as triangles can be generated by surface fitting algorithms and then projected onto a 2D display.

As the most frequently seen isosurface technique, the marching cubes algorithm [30] constructs a contour surface for every cell in the volume. A case table is set up that enumerates all possible topological states of a cell, taking consideration of combinations of scalar values at the cell vertices. The number of topological states relies on the number of cell vertices, and the number of vertex-to-contour value relationships. When the vertex's scalar value is larger than that of the contour line, the vertex is considered within the isosurface. In the same vein, vertices with scalar values lower than the contour value are defined to be outside the isosurface. The resulting binary encoding creates indices for a case table, which denote how the contour intersects the cell. The algorithm marches through the volume one cell after another, and linear interpolations are used to calculate the isosurface locations. The isosurface is complete after all the cells are visited. Since the marching cube algorithm tends to produce a large number of triangles, a triangle decimation algorithm is useful in reducing this number.

Volume Rendering

An isosurface is only a partial representation of the dataset. For a complete view of 3D data, a class of techniques known as volume rendering offers an inside picture of the data. Volume rendering refers to an approximate simulation of the propagation of light through a participating medium represented by the volume. In comparison to the traditional surface rendering techniques, this offers a more effective way to convey information within volumetric data as it can project the entire 3D volume directly onto a 2D plane.

Object-order and image-order are two basic approaches of volume rendering [17]. The former uses a forward mapping scheme to project voxels onto the image plane whereas the latter uses a backward mapping where rays are cast from each pixel in the image plane through the volume to determine the pixel color results. Some researchers have also combined the two techniques. Volume rendering algorithms involve a transfer function to blend or composite data onto a 2D image plane. This transfer function has to be chosen with deliberation to be able to offer segmentation and generate surface like objects. The vector and tensor fields can be visualized by a set of arrows, other point icons or glyphs, which is discussed in detail in [46] [62].

2.2.2 Scientific Dataset Format

The HOPS data are stored in the NetCDF format [1]. The Network Common Data Form, or NetCDF, is an interface to a library of data access programs for storing and retrieving scientific data [4]. NetCDF is an abstraction that supports a view of data as a collection of self-describing, network-transparent objects that can be accessed through a simple interface. Collections of named multidimensional variables can be randomly accessed, without knowing details of how the data are stored. Auxiliary information about the data, such as what units are used, can be stored with the data. Generic utilities and application programs have been written that access arbitrary NetCDF files and transform, combine, analyze, or display specified fields of the data. The development of such applications was led to improved accessibility of data and reusability of software for scientific data management, analysis, and display. The NetCDF interface is supported for the C, FORTRAN and Java computer languages, and for UNIX and Windows operating systems. A NetCDF file has *dimensions*, *variables*, and *attributes*. These components can be used together to capture the meaning of data and relations among data fields in a scientific data set. A NetCDF file is constructed as:

```
netCDF name {
dimensions: ...
variables: ...
data: ...
}
```

NetCDF dimension declarations appear after the dimensions keyword, NetCDF variables and attributes are defined after the variables keyword, and variable data assignments appear after the data keyword.

Data can be categorized into ***scalars***, ***vectors***, and ***tensors***. Scalar data is a magnitude without a direction, for example, temperature, salinity, density, transport streamfunction, etc. Vector data consists of magnitude and directions, which are usually represented as an n-tuple, where n is the dimension of the space. In 3D, it is represented as a triplet of values (x,y,z). Total velocity, geostrophic velocity and vorticity are examples of vectors. Tensors are complex mathematical generalizations of scalars, vectors and matrices. A tensor of rank 0 is a scalar, rank 1 a vector, rank 2 a matrix, and a tensor of rank 3 is a three-dimensional rectangular array. Tensors of higher rank are k-dimensional rectangular arrays. Stress and strain are examples of symmetric tensors. However, in HOPS NetCDF data files, there is no tensor field available.

In HOPS, datasets are sampled on 3D grids. In general, grids are organized to resemble the actual geometry under approximation. The space where the physical phenomena are defined is referred to as the physical space, where data are sampled on the grid nodes and stored in discrete forms of numerical values. On the other hand, in computational space, data are organized into a rectangular uniform mesh in a logical order to ease numerical computation. Between the two spaces, mapping in both directions must be done. Numerical computations performed in the computational space generate results that are mapped back into the physical space for visualization purposes.

2.2.3 Visualization Pipeline

The process of obtaining an image from a data set can be roughly classified as four stages. The final image is achieved through four stages of operations on the original data. The first stage is **data input** during which a data set is read in or generated, resulting in the original field data in the form of a grid. The second stage is **filtering**, where several filters may be applied on the data. Both the input and output is a grid in most cases. The third stage is **mapping** during which the grid data is transformed into geometrical primitives. In this key visualization step, the data are converted into a collection of triangles, lines, points or icons. The final stage is **rendering**, where an image is finally generated by rendering the geometrical primitives according to a particular computer graphics model.

Most algorithms targeted at scientific visualization are associated with the mapping stage. Marching Cubes [30], an approximation of an iso-surface, is the most prominent visualization algorithm. Other commonly used visualization techniques include scatter plots, contour plots, arrow plots, height field representation and direct volume rendering techniques [14].

2.2.4 HOPS Visualization Pipeline

Visualization in HOPS is achieved in the HOPS Plotting Package [1]. The HOPS data are stored in NetCDF format. In addition, an input script *.in* file which includes plotting commands, the parameters input file, the color palette input file and the contour parameters input file are also needed in order to get visualization results. The plotting package has five executable programs:

- cnt_ncar: horizontal contours, non-color version.
  input script: pe_cnt.in or qg_cnt.in

- ccnt_ncar: horizontal contours, color version.
  input script:pe_ccnt.in or qg_ccnt.in

- ccntmd_ncar: horizontal contours, color version, multiple domains[1].
  input script: ccntmd_ncar.in

- sec_ncar: cross-section contours, non-color version.
  input script: pe_sec.in or qg_sec.in

- csec_ncar: cross-section contours, color version.
  input script: pe_csec.in or qg_csec.in

[1] HOPS can handle nested domains.

The prefixes ”pe” and ”qg” in these script file names indicate the type of field classification used: either PE (Primitive Equation) or QG (Quasi-Geostrophic) field identifiers.

The data fields in all “horizontal” levels (discrete constant σ-value isosurfaces) in a HOPS data file can be visualized. The data fields at an actual depth, which are interpolated from existing levels, can also be visualized. Nowcasts and forecasts can be obtained.

The filtering is done by HOPS Fortran 77 routines and our newly developed feature extraction Fortran 77 routines. Feature extraction routines are specialized in detecting features of interest. For instance, in vortex detection, the whole algorithm which includes several filters would be applied to the total velocity field. With the help of NCAR Graphics Library [3], basic and advanced visualization can be achieved. The plotting package uses NCAR Graphics' GKS [1] [3] utilities to help accomplish mapping and rendering, that is, the HOPS Fortran routines call corresponding NCAR's Fortran subroutines to convert data into a collection of lines, points or icons. Some visualization techniques, such as contours, iso-surface and volume rendering, are also easily done by calling NCAR's routines. Therefore, scalar fields and vector fields in HOPS data files can be properly visualized.

NCAR Graphics Library

The NCAR Graphics Library [3] is a Fortran and C based software package for scientific visualization which contains a large number of Fortran/C utilities for drawing contours, maps, vectors, streamlines, weather maps, surfaces, histograms, X/Y plots, annotations, etc. It enables users to apply advanced visualization and analysis techniques to their data. These techniques can be applied to help users gain new insights into data from applications in a wide variety of fields. The NCAR graphics is discipline-independent and easily adapts to new applications and data.

2.2.5 Feature-Based Visualization

Research on feature-based visualization focuses on providing feasible visualization techniques that are semi-automatic and require less user time and little a-priori knowledge of the feature locations. A feature is defined in this context as anything contained in the data set that might be of interest to the user. A feature-based data representation is a high-level data description able to replace the original representation in a succinct and meaningful manner [46].

Less complexity, more information content and a good match to the concepts of the application area, are some of the goals of feature-based methods. By focusing on the user-defined effects or events, feature-based visualization is intended to improve the efficiency of the visualization process. Apart from the normal visualization pipeline, feature-based visualization adds on an automated analysis performed directly on the data set in question and able to get a set of features that are typically represented as icons at the location of the feature.

The demand for feature-based techniques has been growing with increasing availability of simulation data sets. In fluid dynamics, for example, the size of data sets exploded with the wider adoption of time-dependent simulations, which reflected an increase in the resulting data of several orders of magnitude over a steady simulation. So far, only a few research groups have been actively engaging in feature-based visualization research [46] [18] [38] [40] [61] [64].

Examples of a feature-based techniques in oceanography are the extraction of eddies/gyres (vortices), upwelling and jets from real ocean fields. For visualization purposes, these features can be represented by different lines and icons either alone or combined with classical visualization methods. Features can also serve as guides to classical visualization techniques, an example of which is the placement of streamline seed points close to a vortex core. Moreover, the high-level data description of the features can be used for selective visualization. For example, a user can pick a set of features to be visualized instead of displaying all features found in the data set. The representation of point and line features does not require many graphic primitives, and this makes feature-based techniques well suited for virtual reality environments, though limiting the geometry to be drawn remains a main concern.

Feature-based visualization could be utilized for adaptive sampling. One goal of adaptive sampling is to direct measurements for the purpose of providing accurate and efficient forecasts and for evaluating and further developing relevant and useful predictive skill metrics. Therefore, adaptive sampling is often applied to regions bearing phenomena of interest. After the location and size of one feature in the ocean is detected and visualized, the observational pattern in the next step of adaptive sampling can be decided upon with some objective criteria. Thus, feature-based visualization facilitates adaptive sampling and can help minimize the selected error measure in field forecasts.

Chapter 3

Vortex Detection

3.1 Vortices in Fluid Flow

Vortices are among the most important features of fluid flows. It is easy to have an intuitive conception of a vortex, a mental picture of a tornado-like object. However, dissent remains on the formal definition of vortex.

3.1.1 Definition

The fluid dynamicists have proposed several vortex definitions, but none is without problems. The generality of such definitions remains unresolved. Among the first definitions was the one proposed by Lugt[33]:

- "A vortex is the rotating motion of a multitude of material particles around a common center."

In addition to its vagueness, this definition lacks the potential to guide practical algorithms. Targeting this problem, Robinson[45] constructed a more concrete definition of vortex:

- "A vortex exists when instantaneous streamlines mapped onto a plane normal to the vortex core exhibit a roughly circular or spiral pattern, when viewed from a reference frame moving with the center of a vortex core."

Such a definition, even though is more concrete, it is self-referential. Moreover, it is hard to detect the correct reference frame for all types of flows.

Using set theory and differential geometry, Portela [39] developed a set of mathematically rigorous definitions for vortices. Although his 3D definition is not as strong as the 2D version, the intuition behind his definitions is insightful. According to [39], a vortex is made up of a central core region surrounded by swirling streamlines. In this definition, two integral parts, the core and the streamlines, can be identified. Both are of equal importance and both have to be covered by the vortex algorithms. In this thesis, both aspects will be discussed.

3.1.2 Classical 2D Vortices

Rankine Vortex

Rankine Vortex [48] takes consideration of the two limit behaviors. The Rankine Vortex is a simple vortex model, yet there is no mechanism that can produce a Rankine vortex in the atmosphere or the ocean. The Rankine vortex is often called a Rankine combined vortex for the reason that it is built from two separate flow fields. The interior flow field is one that has only a tangential velocity V_θ which increases from zero along the central axis to a maximum value at a radius r_0, which is called a reference radius. This interior region rotates like a solid body even though it is a fluid. The outer flow is also purely tangential and begins with its maximum velocity at radius r_0 and the velocity declines inversely with radius. The Rankine vortex is mathematically described by

$$V_r = 0 \tag{3.1}$$

$$\begin{cases} V_\theta = \omega r \quad (r \leq r_0) \\ V_\theta = \frac{\Gamma}{2\pi r} = \omega \frac{r_0^2}{r} \quad (r > r_0) \end{cases} \tag{3.2}$$

where r = radius from the rotation axis (z-axis) (as in Figure 3-1), ω = angular velocity of any fluid particle within r_0, Γ (circulation) is the vortex strength defined

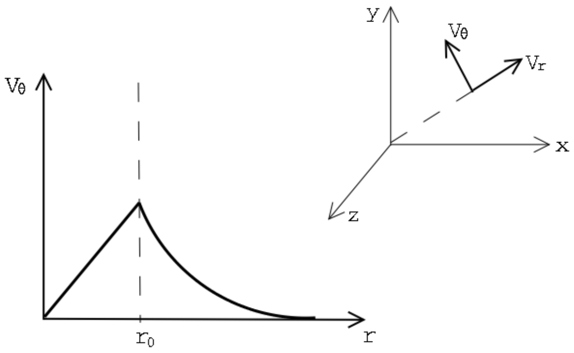


Figure 3-1: Tangential velocity V_θ in Rankine vortex as a function of distance to the rotation axis

by $\Gamma = \int_0^{2\pi} \omega r^2 \, \mathrm{d}\theta = 2\pi\omega r_0^2$ (the same definition is also used in other vortex models). The inner velocity distribution of the vortex ($r \leq r_0$) has constant vorticity 2ω in z direction and the outer velocity distribution ($r > r_0$) has zero vorticity. The advantage of the Rankine vortex, which is not a solution of the Navier-Stokes equations, is that it clarifies the separation between the central region and the potential decay region.

Figure 3-1 on page 49 shows the tangential velocity V_θ of Rankine vortex as a function of the radius from the vortex core.

Potential Vortex

The potential vortex in 2D can be taken as the outer velocity distribution of Rankine vortex [48]. Especially, when r_0 is made to approach zero and ω is increased to keep the product ωr_0^2 constant, the velocity profile becomes,

$$V_r = 0 \tag{3.3}$$

$$V_\theta = \frac{\Gamma}{2\pi r} = \omega \frac{r_0^2}{r} \tag{3.4}$$

where, Γ (circulation) is the vortex strength defined by $\Gamma = \int_0^{2\pi} \omega r^2 \, \mathrm{d}\theta = 2\pi\omega r_0^2$ and r is the radius from the rotation axis.

So the flow approaches zero when going away from the rotation axis, but the downside is that the flow speed approaches infinity when moving closer to the ro-

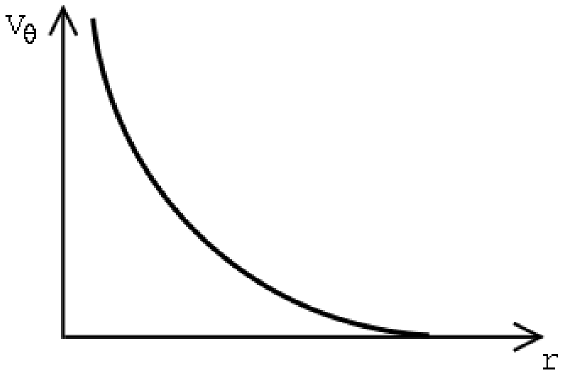


Figure 3-2: tangential velocity V_θ in potential vortex as a function of distance to the rotation axis

tation axis. The potential vortex has the marked property that its vorticity is zero (irrotational) everywhere except at the axis (which is a singularity). Compared to this, the solid body rotation has a constant non-zero curl. The vorticity of the solid body rotation is 2ω in z direction.

Figure 3-2 on page 50 shows the velocity V_θ of a potential vortex as a function of the radius from the rotation axis.

Oseen Vortex

The Oseen vortex is an exact solution of the Navier-Stokes equations [39]. Its velocity profile is given by,

$$V_r = 0 \tag{3.5}$$

$$V_\theta = \frac{\Gamma}{2\pi\sqrt{4\nu t}}\frac{1}{\eta}(1 - e^{-\eta^2}) \tag{3.6}$$

where $\eta = r/\sqrt{4\nu t}$, Γ = vortex strength, t = time, ν = kinematic viscosity.

The Oseen vortex corresponds to the decay of a potential vortex with tangential velocity V_θ given by:

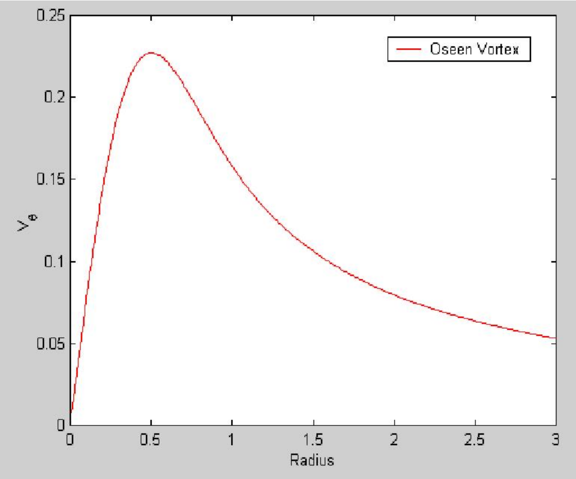


Figure 3-3: Tangential velocity V_θ in Oseen vortex as a function of distance to the rotation axis

$$V_\theta = \frac{\Gamma}{2\pi r} \tag{3.7}$$

In essence, the Oseen vortex can be viewed as a model for a potential decay type of vortex, that is, one that has a central region where the velocity grows with the radius and an outer region where the velocity decays with the radius, with the limit behavior shown below:

$$\begin{cases} r \to 0: & V_\theta \propto r \\ r \to \infty: & V_\theta \propto \frac{1}{r} \end{cases} \tag{3.8}$$

Figure 3-3 on page 51 shows the tangential velocity V_θ of Oseen vortex as a function of the radius from the vortex core.

Taylor Vortex

The Taylor vortex is also an exact solution of the Navier-Stokes equation [39]. Its velocity profile is given by:

$$V_r = 0 \tag{3.9}$$

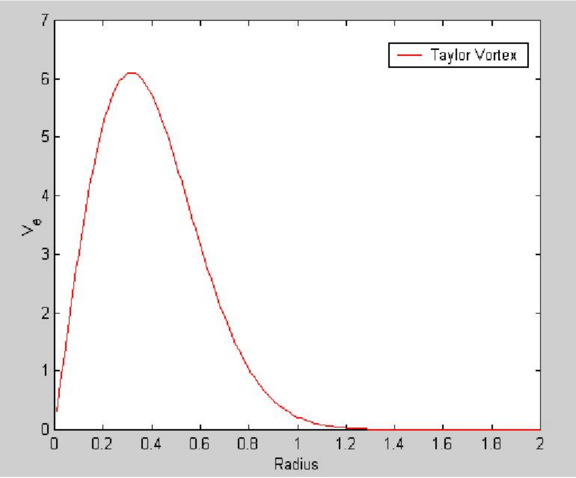


Figure 3-4: Tangential velocity V_θ in Taylor vortex as a function of distance to the rotation axis

$$V_\theta = H\frac{\sqrt{4\nu t}}{\nu t^2}\eta e^{-\eta^2} \tag{3.10}$$

The Taylor vortex corresponds to the decay of a vortex with a finite amount of angular momentum (H is proportional to the angular momentum).

Figure 3-4 on page 52 shows the tangential velocity V_θ of Taylor vortex as a function of the radius from the vortex core.

Other more elaborate models for vortex do exist, however, none is general enough to be taken as a rigorous definition of a vortex. In most cases, some 2D flow plus a tiny component in the third direction is employed whereas in reality vortices can assume a variety of shapes and the vortex axis may not necessarily be a straight line.

3.1.3 Other 3D Vortices

Burgers-Rott Vortex

This vortex has a central axis like the Rankine vortex around which there is a tangential flow. However, unlike the Rankine vortex, a Burgers-Rott vortex has radial and vertical flows as well [9] [47]. “Air spirals in toward the axis and then flows upward. Unlike the Rankine vortex, there is a mechanism for making a Burgers-Rott vortex in the atmosphere, as it results from suction at great height above a plane surface. In the atmosphere a thunderstorm driven by intense convection could provide suction to draw air toward a point on the ground below and draw the air upward into the cloud. Therefore, a Burgers-Rott vortex could approximate the air flow beneath the central region of a large thunderhead. As a model of a real tornado, the Burgers-Rott vortex suffers from the deficiency of placing the tornado beneath the center of a symmetric updraft. Thunderstorms are not symmetric, and their associated tornados rarely occur below their central axis." [2]

The Burgers-Rott vortex is an exact solution to Navier-Stokes equations, and mathematically assumes the form:

$$\begin{aligned} V_r &= -ar \\ V_\theta &= [\Gamma/(2\pi r)](1 - e^{-ar^2/2\nu}) \\ V_z &= 2az \end{aligned} \tag{3.11}$$

where, V_r, V_θ, V_z are the radial, tangential and vertical components of velocity in a cylindrical coordinate system, ν = kinematic viscosity, Γ = vortex strength, a = constant.

Sullivan Vortex

The Sullivan vortex is also an exact solution to the Navier-Stokes equations [56]. It shares some similarities with the Burgers-Rott vortex, but it is a two-celled vortex.

"It has an inner cell in which air flow descends from above and flows outward to meet a separate air flow that is converging radially. Both air flows rise at the point of meeting. What is interesting about the Sullivan vortex is that it is the simplest vortex that can describe the flow in an intense tornado with a central downdraft." [2] The mathematical form of the Sullivan vortex is:

$$\begin{aligned} V_r &= -ar + 6(\nu/r)(1 - e^{-ar^2/2\nu}) \\ V_\theta &= \tfrac{\Gamma}{2\pi r} H(ar^2/2\nu)/H(\infty) \\ V_z &= 2az(1 - 3e^{-ar^2/2\nu}) \end{aligned} \tag{3.12}$$

where, Γ = vortex strength, $H(x) = \int_0^x e^{f(m)}\,\mathrm{d}m$, $f(t) = (-t+3)\cdot\int_0^t (1-e^{-y})/y\,\mathrm{d}y$, ν = kinematic viscosity, a = constant.

3.2 Point-Based Vortex Detection Methods

One category of vortex detection criteria is point-based [50]. It requires the calculation of *physical* quantities at each point in a flow field and then the classification of the values based on thresholds. The assumption behind this is that characteristics of the flow patterns in an infinitely small area around a point can be determined by quantities at that point. The family of point-based methods include pressure magnitude [23], vorticity magnitude [22], helicity [50], and various other quantities derived from the Jacobian $\nabla\mathbf{v}$ (velocity gradient tensor).

3.2.1 Pressure Magnitude

One can use an area of low pressure to define a vortex region, so the vortex covers the region where

$$p \leq p_{threshold} \tag{3.13}$$

The rationale behind this is that a pressure gradient with a minimum of pressure at the center of the vortex produces the centripetal force of a rotating motion. An arbitrary threshold is required for this methods, and viscous forces acting on the field are neglected in this rationale.

3.2.2 Helicity

High helicity can be used as the criterion to define a vortex as the region in which

$$\frac{\mathbf{v}\cdot\omega}{|\mathbf{v}|\cdot|\omega|} \geq h_{threshold} \tag{3.14}$$

where, $\omega = \nabla \times \mathbf{v}$ is vorticity.

This method fails in some simple situations. Since helicities are all zero in 2D flows, this method is useless for 2D flows. Another counter example is that shear flow along a curved boundary has high helicity due to shear effects, but vortices may not exist in it.

3.2.3 Vorticity Magnitude

This commonly used method uses vorticity as the definition of rotating motion and defines a vortex as a region of high vorticity. According to this method, a vortex is a region where the vorticity is greater than or equal to some threshold ω:

$$|\nabla \times \mathbf{v}| \geq \omega_{threshold} \tag{3.15}$$

Like the pressure magnitude method, this method demands some arbitrary value for $\omega_{threshold}$. In addition, the absence of vorticity does not necessarily guarantee that there is no vortex, as can be shown in the case of a potential vortex.

3.2.4 Conjugate Pair Eigenvalues

Chong *et al.* [11] used eigenvalues of the velocity gradient tensor to classify the local streamline pattern around any point in a flow in a reference frame moving with the velocity of that point. They proposed that a vortex core is a region with complex eigenvalues of velocity gradient tensor $\nabla\mathbf{v}$. Complex eigenvalues imply that the local streamline pattern is closed or spiral in a reference frame moving with that point. Complex eigenvalues are essentially the necessary condition of swirling flows in the vortex area.

The eigenvalues, λ, of $\nabla\mathbf{v}$ satisfy the characteristic equation:

$$\lambda^3 - P\lambda^2 + Q\lambda - R = 0 \tag{3.16}$$

where $P \equiv v_{i,i} = 0$ (incompressible flow), $Q \equiv \frac{1}{2}(v_{i,i}^2 - v_{i,j}v_{j,i}) = -\frac{1}{2}v_{i,j}v_{j,i}$ and $R = Det(v_{i,j})$ are the three invariants of $\nabla\mathbf{v}$. Complex eigenvalues will occur when the discriminant ($\triangle$) is positive, i.e.,

$$\triangle = (\frac{1}{3}Q)^3 + (\frac{1}{2}R)^2 > 0 \tag{3.17}$$

This definition, derived from $\nabla\mathbf{v}$, is the Galilean invariant [23].

3.2.5 Negative Eigenvalue λ_2

The λ_2 method proposed by Jeong and Hussain [23] is a widely used vortex detection method. First, the local Jacobian J is decomposed into a symmetric part S and its antisymmetric part Ω:

$$S = \frac{J + J^T}{2} \tag{3.18}$$

$$\Omega = \frac{J - J^T}{2} \tag{3.19}$$

where, Jacobian $J = \nabla\mathbf{v}$.

Next, the eigenvalues of the matrix $S^2 + \Omega^2$ are determined. All three eigenvalues are real because of the symmetric nature of this matrix. The λ_2 criterion then defines a vortex as the region where at least two of the three values are negative, which is identical to the condition that the second largest eigenvalue is negative $\lambda_2 < 0$.

3.2.6 Positive Second Invariant of Jacobian

This is based on the second invariant of a 3D matrix m_{ij}:

$$Q = m_{11}m_{22} - m_{12}m_{21} + m_{11}m_{33} - m_{13}m_{31} + m_{22}m_{33} - m_{23}m_{32} \tag{3.20}$$

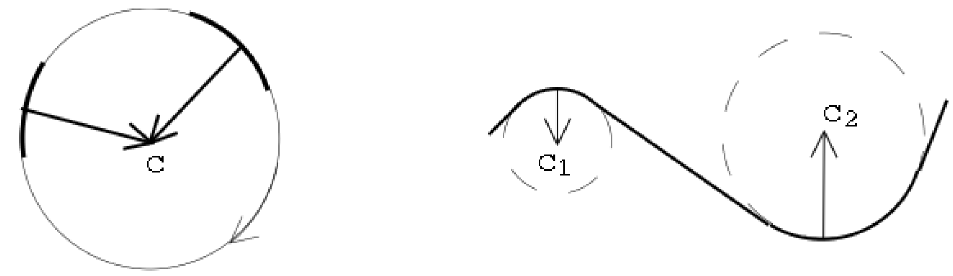
A vortex is the region where Q is positive [23]. Another vortex definition proposed [21], required that the second invariant Q of the Jacobian J is positive and that local pressure is smaller than the surrounding pressure. Jeong and Hussian [23] have observed that the second condition is always true if the first one is, thus reducing the condition to only demanding a positive second invariant $Q > 0$.

3.2.7 Conclusions on Point-based Methods

The point-based methods have been used in many applications. As long as the vortices have strong rotation speed around vortex cores and different vortices are sufficiently isolated from each other, point-based methods can be used and fairly good results will be achieved. But none of these point-based methods is able to reliably detect all vortices in large-scale real ocean flows. Here, there are some typical problems:

In some cases, although stronger vortices which have high angular velocity are usually found, weaker vortices with low angular velocity remain undetected by point-based methods. In other cases, some flows with seemingly swirling properties are detected. However, they are not necessarily vortices. They could be shear flows near the coastlines. Thus, false positives are introduced to the results. Moreover, if two vortical structures are close together, they will often be lumped together by point-based techniques, even if they for example have opposite direction of rotation. In fact, a vortex structure has only one vortex core. If detection methods indicate that two vortices merge into one, it should be further examined whether this is actually the case of two vortices simply moving too close to each other to be clearly distinguished, but still having two different vortex cores.

Last but not least, arbitrary thresholds may need to be used in point-based techniques. Thresholds are based on field values, which make thresholds hard to be determined. In different cases, threshold values need to vary in order to get good results. Thus, user interaction is required, and the whole process is not fully automated. In a nutshell, a vortex is essentially a macroscopic or regional phenomenon, yet the point values underlying the point-based criteria do not always translate into regional characteristics.



(a) Circular streamline with coinciding curvature centers C

(b) Non-circular streamline with scattered curvature centers C1, C2

Figure 3-5: Streamlines with centers of curvature, adapted from [50].

3.3 Set-Based Vortex Detection Methods

The other category of vortex detection criteria is set-based [49]. The criteria are based on *geometric* characteristics of streamlines, as represented by the shape or curvature of instantaneous streamlines. They build on the intuitive idea of a swirling pattern of flow around the vortex core in a vortex area.

3.3.1 Curvature Center Density (CCD)

The curvature center methods [27] attempts to detect a vortex in 2D by sampling the field at many points and often at all grid nodes. For each point sampled, the center of curvature is determined which is the center of the osculating circle of the streamline through that point [15]. The center of curvature should accumulate at a point in vortical regions of the field (as in Figure 3-5(a) on page 58). The samples taken on this circular streamline all project to the same center of curvature whereas the centers of curvature in the non-vortical regions of the field are scattered (as in Figure 3-5(b) on page 58).

Thus, a set of curvature center points can be obtained and then accumulated into a new grid. The number of curvature centers in each cell forms a new scalar field called the curvature center density (CCD) field.

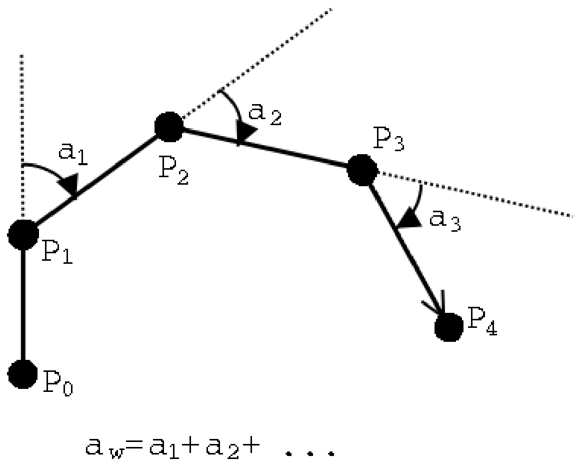


Figure 3-6: The winding angle α_w of a piecewise linear curve, adapted from [50].

3.3.2 Winding Angle

Portela's research [39] also inspired another geometric method for detecting vortex in 2D which elaborated on the intuitive idea of a swirling pattern around a central set of points. This methods attempts to locate a vortex by selecting and clustering closed streamlines. A winding-angle criterion simplified by Sadarjoen [49] and a distance criterion is used in the selection process.

Let S_i be a 2D streamline, consisting of points $P_{i,j}$ and line segments ($P_{i,j}$, $P_{i,j+1}$), and let $\angle(A, B, C)$ denote the angle between line segments AB and BC. Then, the winding angle $a_{\omega,i}$ of streamline S_i is defined as the cumulative change of direction of the streamline segments (see Figure 3-6 on page 59):

$$a_{\omega,i} = \Sigma_{j=1}^{N-1} \angle(P_{i,j-1}, P_{i,j}, P_{i,j+1}) \tag{3.21}$$

Signed angles are used in calculation, with positive rotation for a counterclockwise-rotating curve and negative rotation for a clockwise-rotating curve. Apparently, if two conditions are satisfied, namely that $a_{\omega,i} = \pm 2\pi$ and the "final" point in a streamline lies near its initial point, the streamline is fully closed. In real applications, lower values, such as 1.9π, 1.8π, may be used to find swirling streamlines which do not achieve a full revolution.

The two criteria used by the selection process are: the winding angle of a streamline should be very close to 2π, and the distance between the starting and ending point of the streamline should be relatively small.

3.3.3 Conclusions on Set-Based Methods

The set-based methods begin by covering the full domain with a large number of streamlines, and then selecting the curves with circular or closed geometry. The CCD method uses local accumulations of curvature centers which may indicate that many streamlines are swirling around a cluster of center points. The winding angle method detects circular patterns of streamlines by checking the cumulative changes of streamline direction, as represented by the winding angle. The results of set-based techniques look promising, especially that of the winding angle method. It is very effective in detecting weak vortices. This is because the set-based criteria are based on *geometric* properties of streamlines, they are largely insensitive to the angular velocity of a vortex. However, both methods are computationally more expensive. Thresholds are based on geometries for set-based methods while they are based on field values for point-based methods. Since field values vary considerably from case to case, it is difficult to determine thresholds. Geometry-based thresholds on the other hand are a function of mesh properties. Thus, automated vortex detection can be achieved.

3.4 A Geometric-Based Vortex Core Detection Method

This local, set-based algorithm was originally introduced by Jiang *et al.* [25] [24]. It is applied to the first step of vortex detection in this study: detecting the vortex core. Here, it is tested on real large-scale ocean models. In every case, the vortex core regions detected were consistent with similar results from other vortex detection methods and validated through human visual inspection.

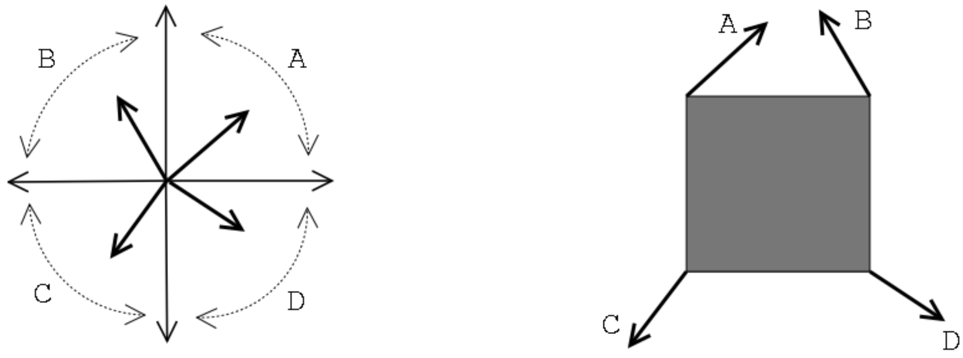


Figure 3-7: The four equally spaced direction ranges correspond to direction labeling of the rectangular cell

3.4.1 Direction Labeling

Direction labelling assigns labels to vectors which point to different direction ranges. As shown in Figure 3-7 on page 61, one can consider four direction ranges named A, B, C, D. If a vector points to direction range A, then label A is given to that vector. Thus, each velocity vector is labelled according to the direction to which it points. The number of directions can be set as big as desired. But a very high level of quantization does not improve results much, while the work of checking for direction-derived properties will increase. In Cartesian grids, four direction ranges is a good choice since they correspond to the grid's geometry. In our experiments, a level of quantization of four is sufficient to produce accurate results.

3.4.2 2D Algorithm

Since velocity vectors around core regions represent certain flow patterns that have circulatory properties, it is sufficient to examine the immediate neighborhood of a grid point for the existence of those flow patterns. The concept of the *direction spanning property* is introduced here. In the 2D case, the neighboring four vectors of a grid point are checked. If any three of four vectors point to three different direction ranges, i.e, any three of four vectors have 3 different labels, then the condition of the direction spanning property is satisfied and the surrounding flow of the grid point exhibits the swirling flow property. Therefore, that grid point is a vortex core candidate. In

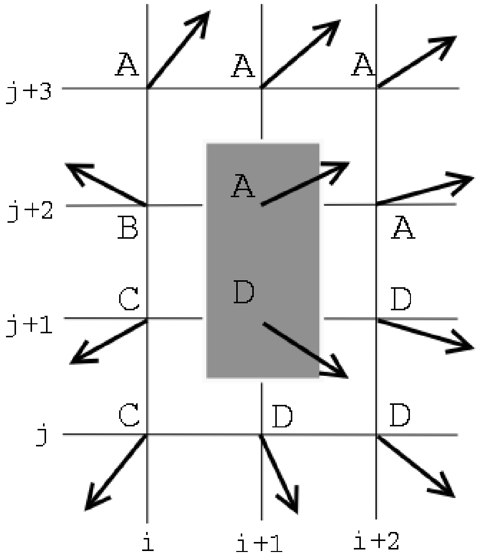


Figure 3-8: The two grid points satisfy the direction spanning property.

Figure 3-8 on page 62, the immediate neighboring vectors of points (i+1,j+1) and (i+1,j+2) point to 3 different direction ranges, so points (i+1,j+1) and (i+1,j+2) are vortex core candidates.

3.4.3 Post-Processing – Cleanup

The grid points which satisfied the direction-spanning property are vortex core candidates since flows around them have swirling property. However, shear flows and switching flows may also have swirling characteristics. These two kinds of flows do not exist in a vortex area. So a post-processing step, i.e., cleanup work, is required. A necessary condition for swirling flows in a vortex area is that the velocity gradient tensor has a pair of conjugate eigenvalues [23]. During the cleanup process, the vortex core candidates which do not meet this condition are eliminated. Vortex core candidates are then filtered and only the real vortex cores remain.

3.4.4 Advantages of this Method

This method is capable of detecting all vortex core candidates because their surrounding streamlines must satisfy the direction-spanning property. Then the false positives can be removed using the conjugate eigenvalues condition. After the vortex cores are detected, the detection of full vortex areas also becomes easier. Since an entire vortex region is made up of a vortex core and its surrounding closed streamlines, we can check closed streamlines around every vortex core instead of extracting all streamlines passing through grid points and detecting the closed ones, which makes the process much more efficient and accurate since the vortex core candidate dataset is much smaller than the grid point dataset. Detection can be done using the winding angle method.

3.4.5 Classical Model Results

We use the Rankine vortex model to check the validity of several vortex detection methods. The velocity profile of Rankine vortex are given by,

$$V_r = 0 \tag{3.22}$$

$$V_\theta = \begin{cases} kr \quad (r \leq r_0) \\ \frac{\Gamma}{2\pi r} = k\frac{r_0^2}{r} \quad (r > r_0) \end{cases} \tag{3.23}$$

Vorticity Magnitude

$$\omega = \omega_z = \frac{1}{r}\frac{\partial(rV_\theta)}{\partial r} - \frac{1}{r}\frac{\partial V_r}{\partial \theta} \tag{3.24}$$

Therefore,

$$\omega_z = \begin{cases} 2k \quad (r \leq r_0) \\ 0 \quad (r > r_0) \end{cases} \tag{3.25}$$

The flow is circulatory but irrotational outside r_0 and has uniform vorticity $2k$ inside r_0. If α ($0 < \alpha \leq 2k$) is used as the threshold of the vorticity magnitude method, only the forced vortex ($r \leq r_0$) is detected as a vortex area, while the free vortex $r > r_0$ is not detected. So the high vorticity magnitude partially fails in this example.

Negative Eigenvalue λ_2

The velocity gradient in a cylindrical coordinate system is given by,

$$\nabla V = \begin{bmatrix} \frac{\partial V_r}{\partial r} & \frac{\partial V_\theta}{\partial r} \\ \frac{1}{r}\frac{\partial V_r}{\partial \theta} - \frac{V_\theta}{r} & \frac{1}{r}\frac{\partial V_\theta}{\partial \theta} + \frac{V_r}{r} \end{bmatrix} \tag{3.26}$$

Case 1 ($r \leq r_0$):

$$\nabla V = \begin{bmatrix} 0 & k \\ -k & 0 \end{bmatrix} \tag{3.27}$$

and,

$$(\nabla V)^T = \begin{bmatrix} 0 & -k \\ k & 0 \end{bmatrix} \tag{3.28}$$

Therefore,

$$S = \frac{1}{2}(\nabla V + (\nabla V)^T) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{3.29}$$

and,

$$\Omega = \frac{1}{2}(\nabla V - (\nabla V)^T) = \begin{bmatrix} 0 & k \\ -k & 0 \end{bmatrix} \tag{3.30}$$

$$\Omega^2 = \begin{bmatrix} -k^2 & 0 \\ 0 & -k^2 \end{bmatrix} \tag{3.31}$$

Thus,

$$S^2 + \Omega^2 = \begin{bmatrix} -k^2 & 0 \\ 0 & -k^2 \end{bmatrix} \tag{3.32}$$

Solve the following equation to get eigenvalues.

$$\begin{vmatrix} -k^2-\lambda & 0 \\ 0 & -k^2-\lambda \end{vmatrix} = 0 \tag{3.33}$$

The solution is $\lambda_1 = \lambda_2 = -k^2$. Since $k \neq 0$, both eigenvalues are negative.

Case 2 ($r > r_0$):

$$\nabla V = \begin{bmatrix} 0 & -kr_0^2 \cdot \frac{1}{r^2} \\ kr_0^2 \cdot \frac{1}{r^2} & 0 \end{bmatrix} \tag{3.34}$$

and,

$$(\nabla V)^T = \begin{bmatrix} 0 & kr_0^2 \cdot \frac{1}{r^2} \\ -kr_0^2 \cdot \frac{1}{r^2} & 0 \end{bmatrix} \tag{3.35}$$

Therefore,

$$S = \frac{1}{2}(\nabla V + (\nabla V)^T) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{3.36}$$

and,

$$\Omega = \frac{1}{2}(\nabla V - (\nabla V)^T) = \begin{bmatrix} 0 & -kr_0^2 \cdot \frac{1}{r^2} \\ kr_0^2 \cdot \frac{1}{r^2} & 0 \end{bmatrix} \tag{3.37}$$

$$\Omega^2 = \begin{bmatrix} -(kr_0^2 \cdot \frac{1}{r^2})^2 & 0 \\ 0 & -(kr_0^2 \cdot \frac{1}{r^2})^2 \end{bmatrix} \tag{3.38}$$

Thus,

$$S^2 + \Omega^2 = \begin{bmatrix} -(kr_0^2 \cdot \frac{1}{r^2})^2 & 0 \\ 0 & -(kr_0^2 \cdot \frac{1}{r^2})^2 \end{bmatrix} \tag{3.39}$$

Solve the following equation to get eigenvalues.

$$\begin{vmatrix} -(kr_0^2 \cdot \frac{1}{r^2})^2-\lambda & 0 \\ 0 & -(kr_0^2 \cdot \frac{1}{r^2})^2-\lambda \end{vmatrix} = 0 \tag{3.40}$$

The solution is $\lambda_1 = \lambda_2 = -(kr_0^2 \cdot \frac{1}{r^2})^2$. Since $k \neq 0$, both eigenvalues are negative. Therefore, both the forced vortex ($r \leq r_0$) and the free vortex ($r > r_0$) are detected. The negative λ_2 method is valid in this example.

Second Invariant of Jacobian

Case 1 ($r \leq r_0$):

$$|\Omega| = k^2 \tag{3.41}$$

and,

$$|S| = 0 \tag{3.42}$$

Thus,

$$Q = \frac{1}{2}(|\Omega|^2 - |S|^2) = \frac{1}{2}k^4 \tag{3.43}$$

Since $k \neq 0$, Q is positive.

Case 2 ($r > r_0$):

$$|\Omega| = (kr_0^2 \cdot \frac{1}{r^2})^2 \tag{3.44}$$

and,

$$|S| = 0 \tag{3.45}$$

Thus,

$$Q = \frac{1}{2}(|\Omega|^2 - |S|^2) = \frac{1}{2}(kr_0^2 \cdot \frac{1}{r^2})^4 \tag{3.46}$$

Since $k \neq 0$, Q is positive.

Therefore, both the forced vortex ($r \leq r_0$) and the free vortex ($r > r_0$) are detected. The positive Q method is valid in this example.

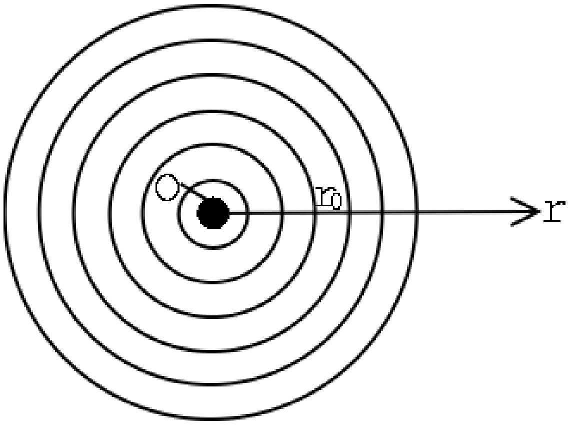


Figure 3-9: Streamlines of Rankine Vortex

Geometric-Based Vortex Core Detection

Streamlines of the Rankine vortex model are shown in Figure 3-9 on page 67. The only point which satisfies the direction spanning property is the center O. Therefore, the point O is detected as a vortex core. It is truly the vortex core for the Rankine vortex model. Thus, the geometric-based vortex core detection method is valid in this example.

3.4.6 Real Ocean Model Results

Massachusetts Bay

The main attributes of the Massachusetts Bay HOPS dataset from June, 18, 1997 are:

tlat (number of tracer points in the latitude direction) = 118 ;

tlon (number of tracer points in the longitude direction) = 64 ;

vlat (number of velocity points in the latitude direction) = 118 ;

vlon (number of velocity points in the longitude direction) = 64 ;

level (input vertical levels or depths) = 11 ;

outlev (output vertical levels or depths) = 11 ;

time (the number of days since initialization) = 15;

rlngd (transformation centroid longitude) = -70.45^o (West 70.45 degree) ;

rlatd (transformation centroid latitude) = 42.18^o (North 42.18 degree) ;

thetad (domain rotation angle) = 38^o ;

coord (flag for type of horizontal grid) = 0 (Cartesian (tangent plane) grid) (see Chapter 1.2.2 for explanations);

The visualization of the HOPS horizontal velocity field in Massachusetts Bay is shown in Figure 3-10 on page 70. Using the horizontal velocity field, the 2D vorticity and velocity gradient tensor can be calculated.

The visualization of the HOPS vertical velocity field in Massachusetts Bay is shown in Figure 3-11 on page 71.

Vortex detection in the Massachusetts Bay using flow fields generated by HOPS and employing the vorticity magnitude method is shown in Figure 3-12 on page 72. The minimum value of the vorticity magnitude is 0.0 at point (1, 1), the maximum value is 2.09170947×10^3 at point (38, 37), the mean value is 1.26739182×10^2 and the standard deviation is 1.574661712×10^2. Applying a threshold, we can get a cleaner picture of boolean values of vorticity. The boolean values of the vorticity magnitude method using thresholds "mean of vorticity field", "1.5×mean of vorticity field", "2×mean of vorticity field", and "3×mean of vorticity field" are shown in Figure 3-13 on page 73, Figure 3-14 on page 74, Figure 3-15 on page 75, and Figure 3-16 on page 76. Although some vortex regions are detected, there are many false positives in the domain, and the boundaries of the vortex regions are not clear. Moreover, selecting a good threshold will improve results significantly. The vorticity magnitude method requires an arbitrary threshold so human interaction is needed.

Vortex detection in the Massachusetts Bay employing the conjugate pair eigenvalues method is shown in Figure 3-17 on page 77. Almost all vortex regions are detected, but still, there are many false positives. Note that the condition of conjugate pair eigenvalues is a necessary condition for a vortex, but not a sufficient one.

Vortex detection in the Massachusetts Bay employing the negative λ_2 method with thresholds "0", "-100", "-500", "-1000", and "-5000" are shown in Figure 3-18 on page 78, Figure 3-19 on page 79, Figure 3-20 on page 80, Figure 3-21 on page 81, and Figure 3-22 on page 82. The minimum value of λ_2 is -6.52877875×10^5 at point (37, 35), the maximum value is 2.49484625×10^6 at point (63, 93), the mean value is 9.88204980×10^3 and the standard deviation is 7.78657266×10^4. Although some vortex regions are detected, there are many false positives in the domain. In addition, different vortices which are close to each other could not be distinguished. Selecting a suitable threshold is required in this method. Therefore, it's not a automated process.

Vortex detection in the Massachusetts Bay employing the positive Q method with thresholds "0", "100", "1000", "5000", and "1×10^8" are shown in Figure 3-23 on page 83, Figure 3-24 on page 84, Figure 3-25 on page 85, Figure 3-26 on page 86, and Figure 3-27 on page 87. The minimum value of Q is $-6.42370109 \times 10^{11}$ at point (40, 39), the maximum value is 9.99087997×10^6 at point (38, 37), the mean value is 2.01903504×10^8 and the standard deviation is $1.78935439 \times 10^{10}$. Similar problems exist as with the result of the negative λ_2 method. An arbitrary threshold is required in the process.

In practice, for the λ_2 method and positive Q method, the exact threshold 0 can not be used because of too much noise.

Figure 3-28 on page 88 shows the result of the geometric-based vortex core detection method without cleanup. Note that all the above methods detect vortices, whereas the geometric method detects vortex cores. The result before cleanup is fairly good but there are still some false positives, which are caused by shear flows and switching flows. We applied conjugate pair eigenvalues (defined as A), negative λ_2 (defined as B), and positive Q (defined as C) criteria and set operations between their results as cleanup methods. The results are shown in Figure 3-29 on page 89 to Figure 3-38 on page 98. Since swirling flows in the vortex area must have conjugate pair eigenvalues in the velocity gradient tensor, the false positives that do not have this characteristic are filtered in the cleanup work. So the conjugate pair eigenvalues, which is the necessary condition for a vortex area, can be singly used as the cleanup method. Our results show that more invloved cleanup filters do not improve the quality of the result. Eventually, only vortex cores are detected in Figure 3-29 on page 89. We see that even when two vortex cores are very close to each other, they

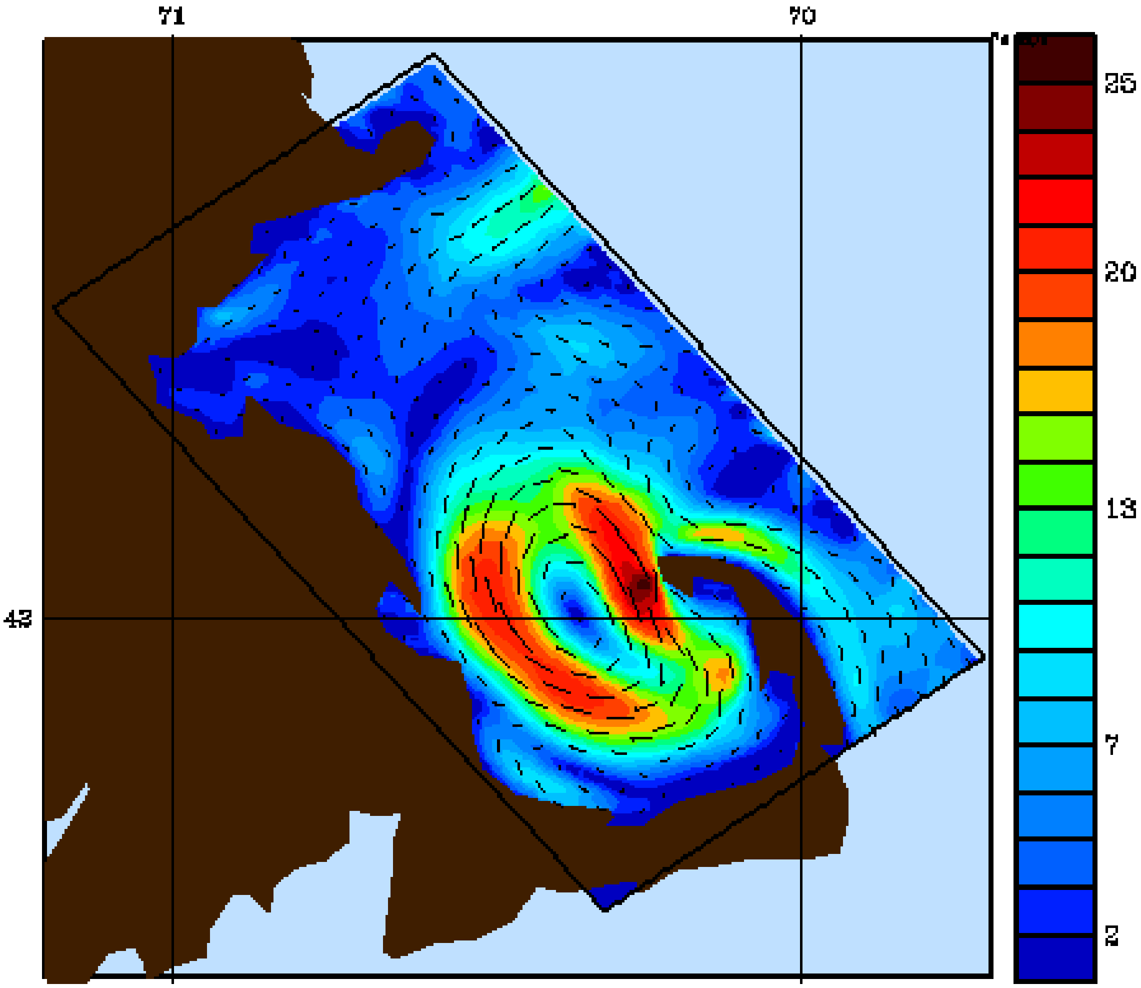


Figure 3-10: The horizontal velocity magnitude in Massachusetts Bay.

are still detected as two different vortex cores.

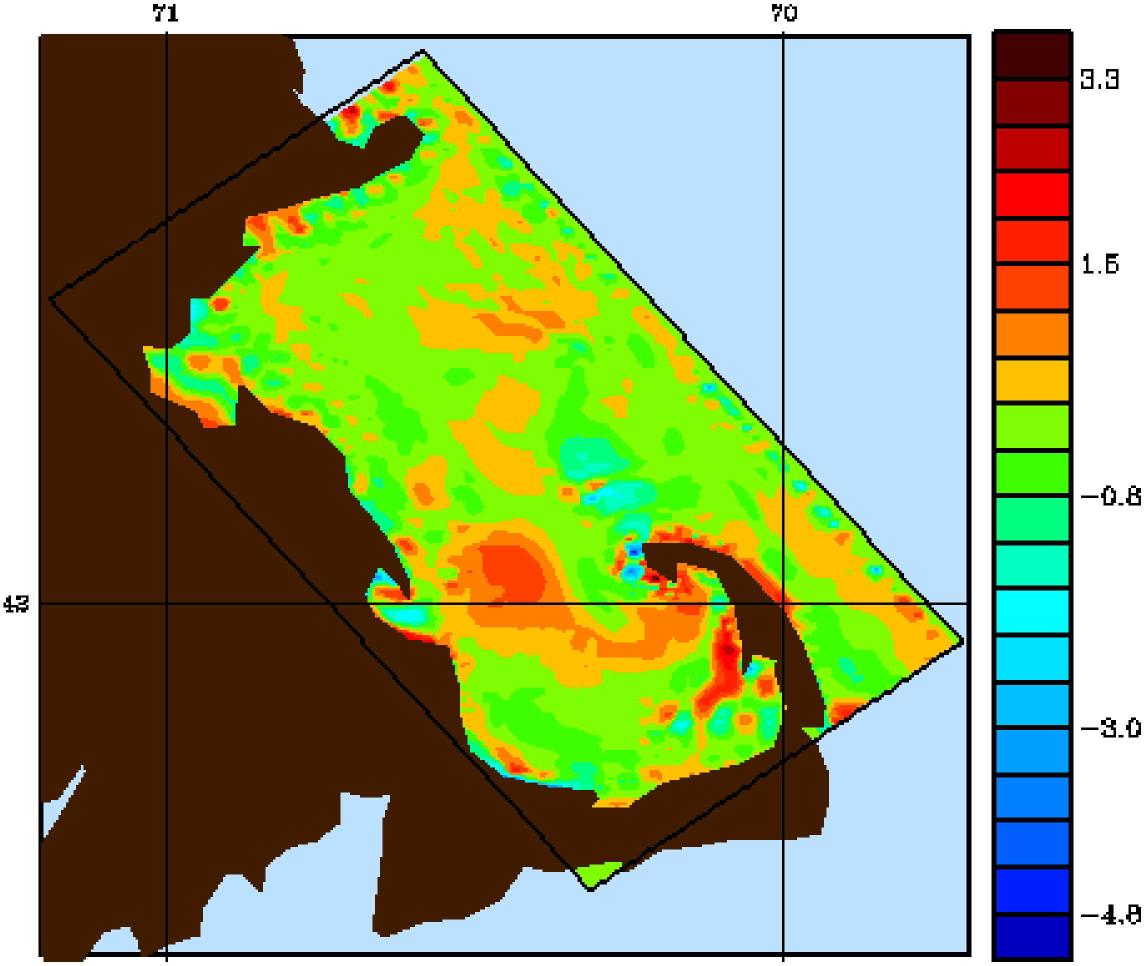


Figure 3-11: The vertical velocity magnitude in Massachusetts Bay.

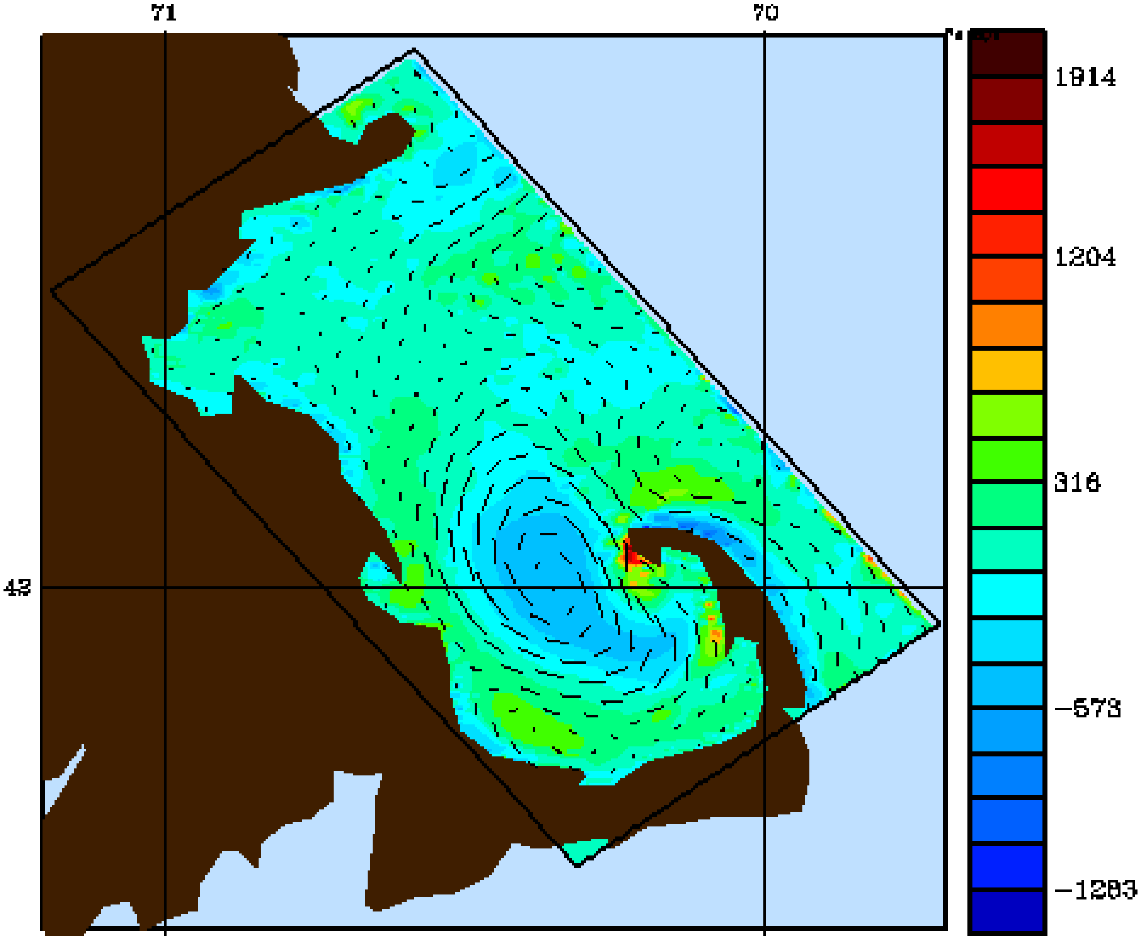


Figure 3-12: The result of the vorticity magnitude method with superposed velocity vectors in Massachusetts Bay.

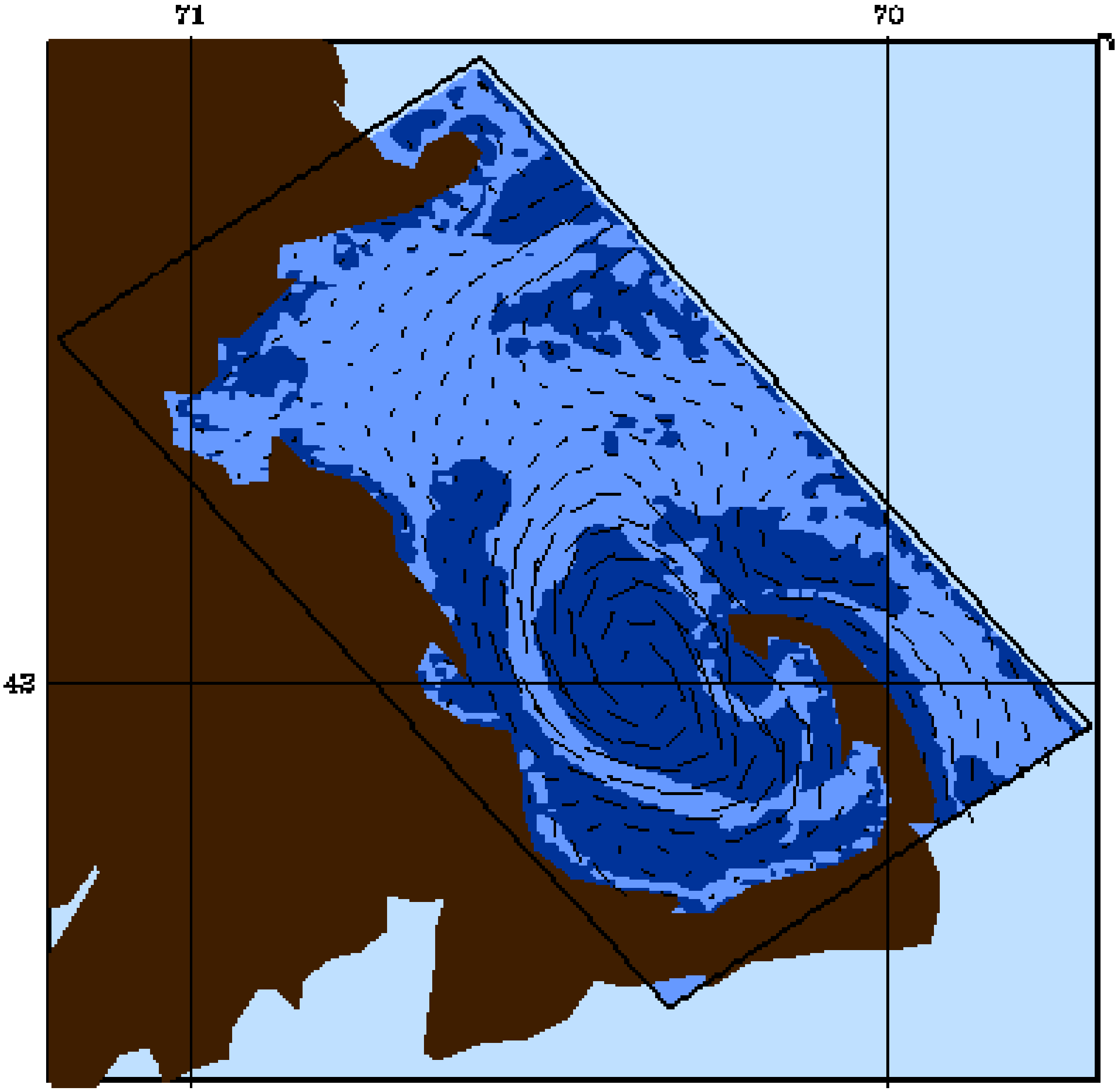


Figure 3-13: The boolean values of the vorticity magnitude method with the threshold "mean of vorticity field" with superposed velocity vectors in Massachusetts Bay. The deeper color shows the vortex area.

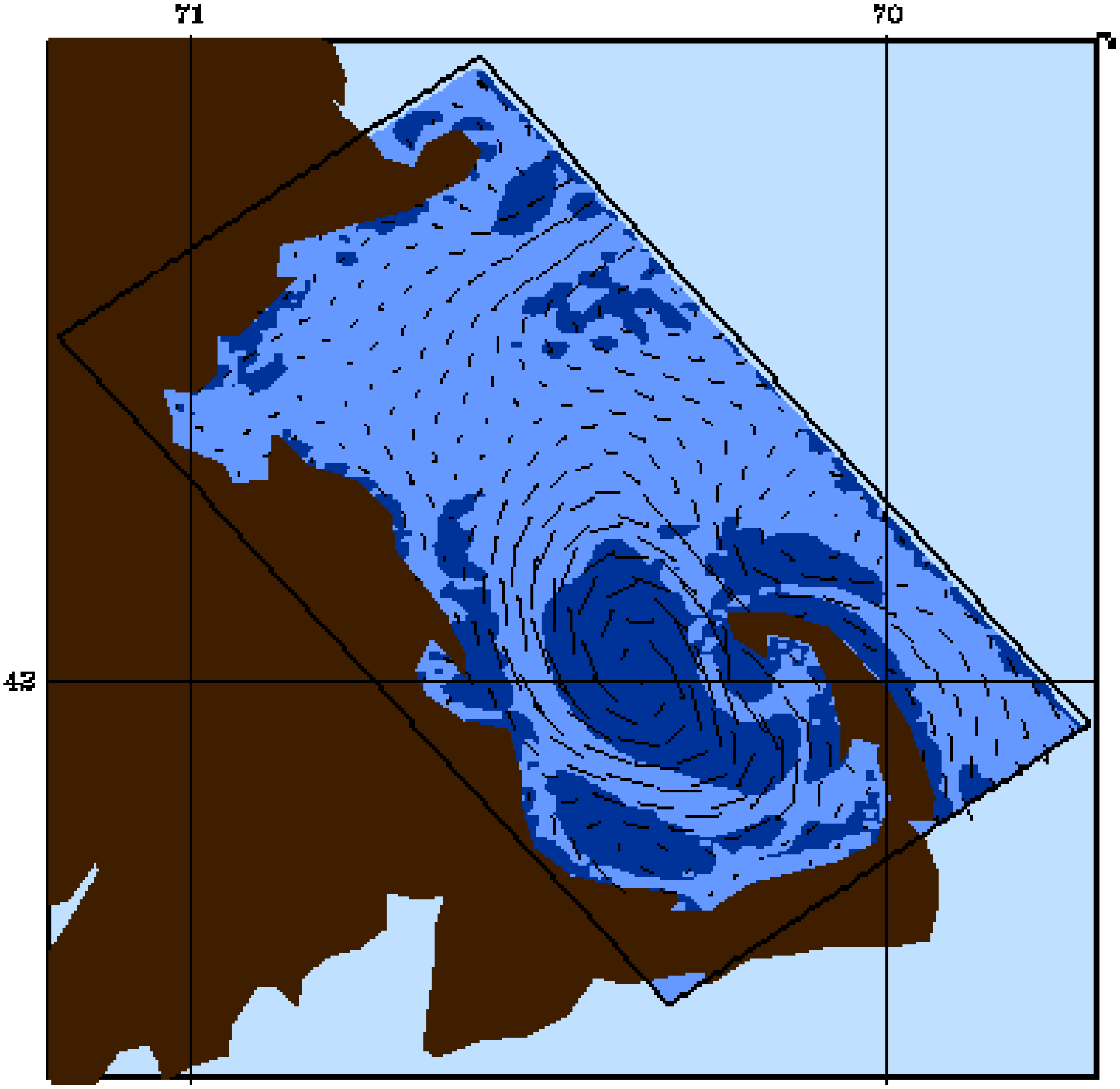


Figure 3-14: The boolean values of the vorticity magnitude method with the threshold "1.5×mean of vorticity field" with superposed velocity vectors in Massachusetts Bay. The deeper color shows the vortex area.

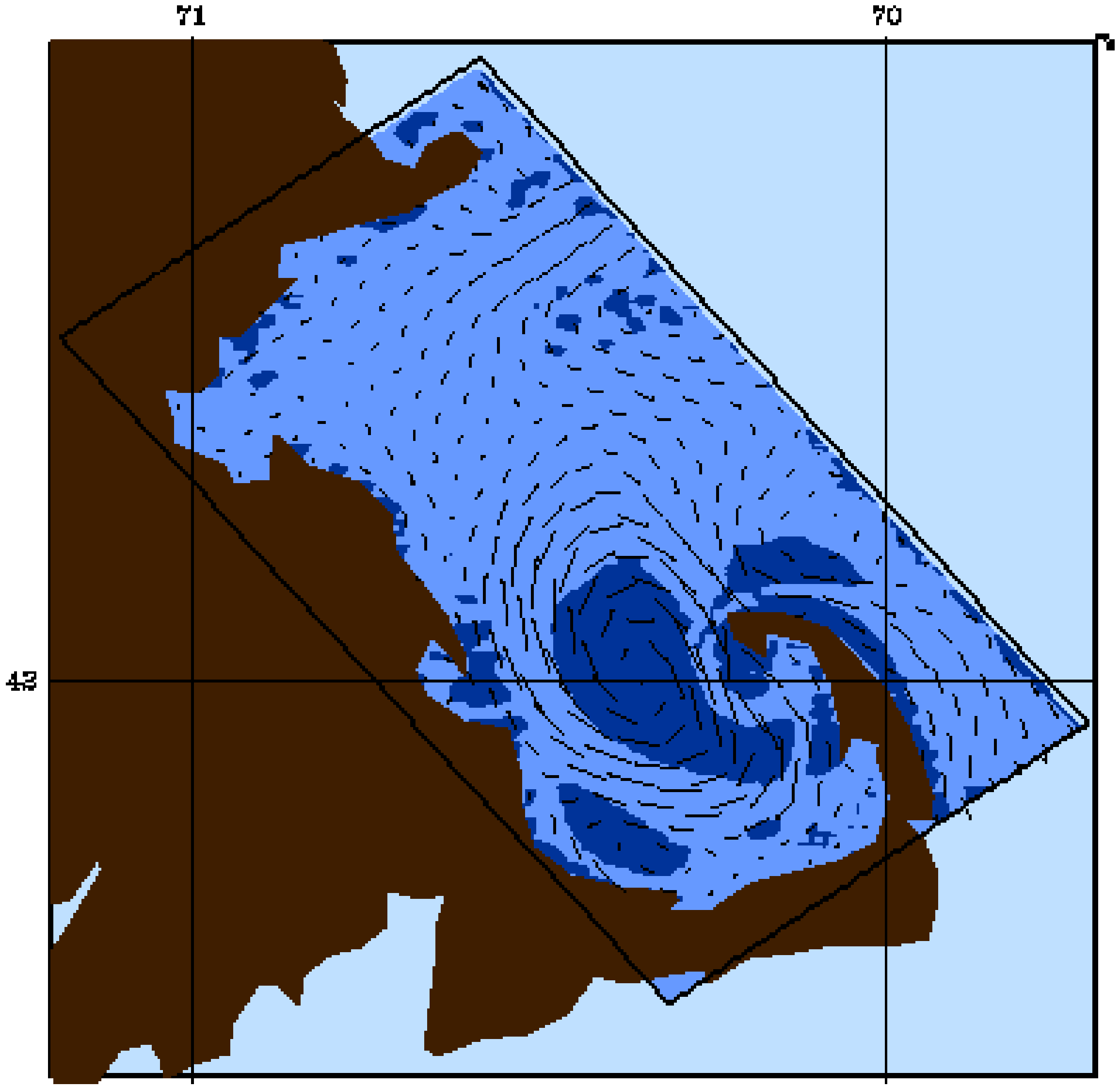


Figure 3-15: The boolean values of the vorticity magnitude method with the threshold "2×mean of vorticity field" with superposed velocity vectors in Massachusetts Bay. The deeper color shows the vortex area.

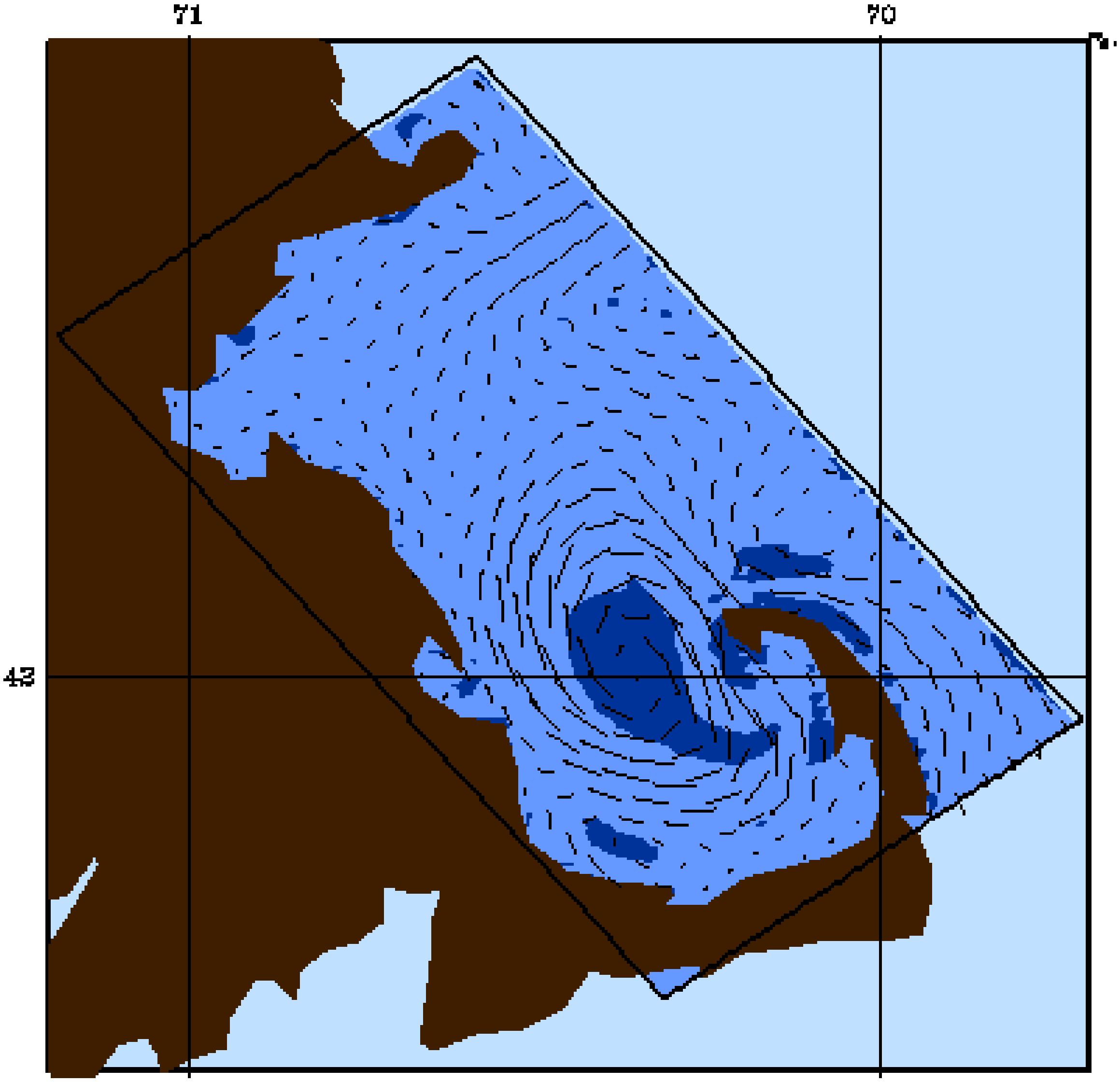


Figure 3-16: The boolean values of the vorticity magnitude method after the threshold “3×mean of vorticity field” with superposed velocity vectors in Massachusetts Bay. The deeper color shows the vortex area.

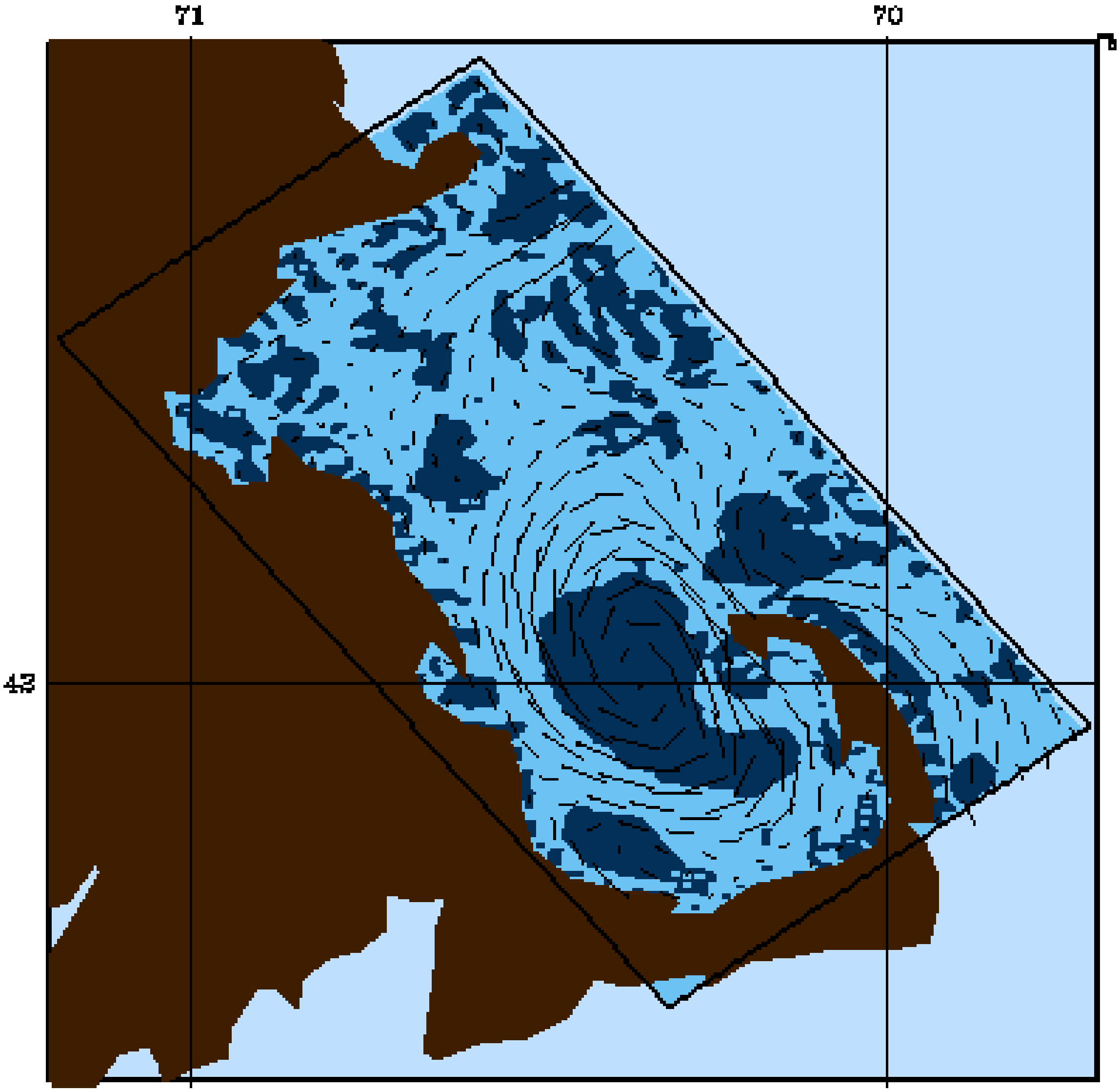


Figure 3-17: The result of the conjugate pair eigenvalues method in Massachusetts Bay. The deeper color shows the vortex area.

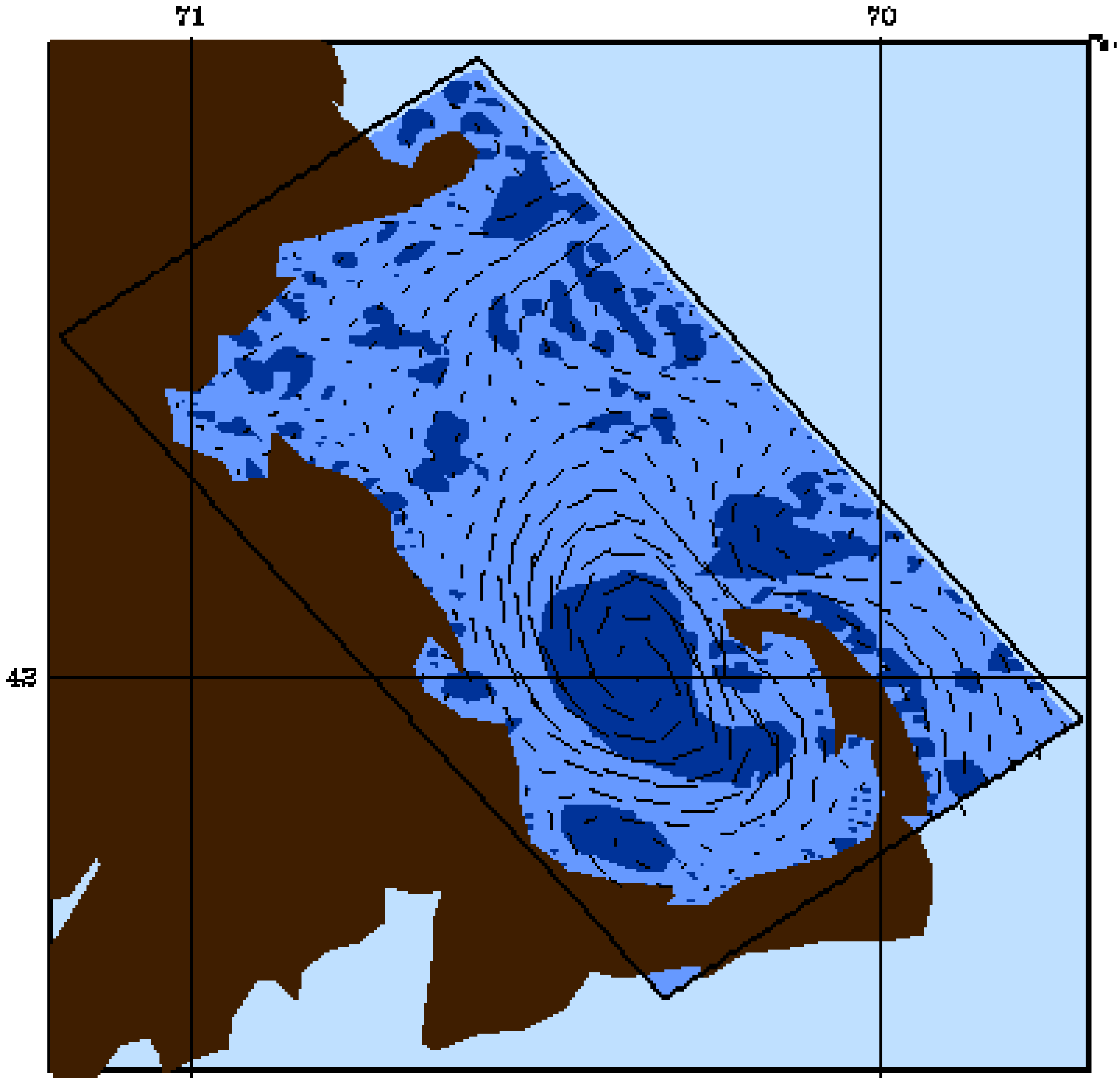


Figure 3-18: The result of the negative λ_2 method with the threshold “0” in Massachusetts Bay. The deeper color shows the vortex area.

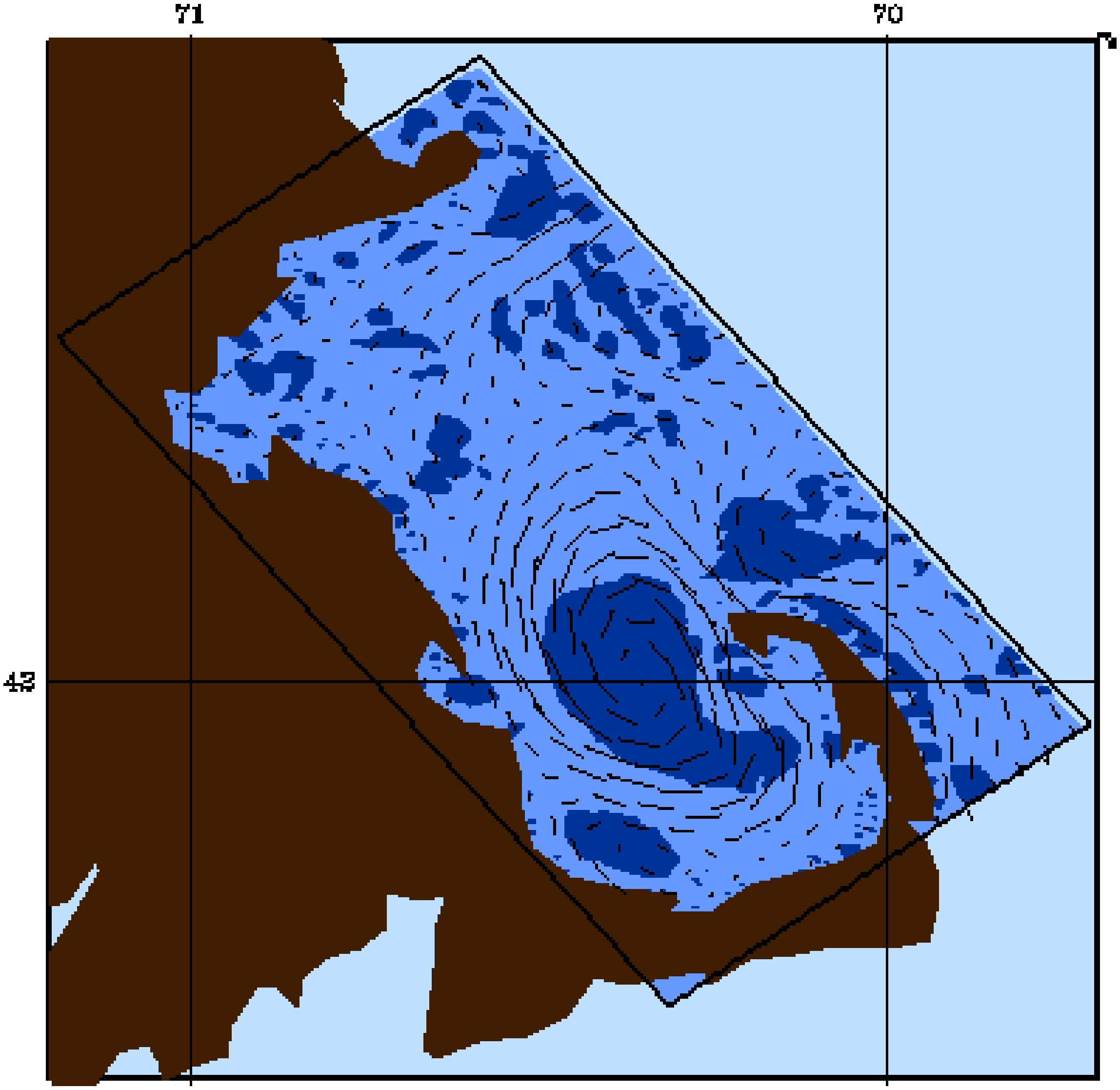


Figure 3-19: The result of the negative λ_2 method with the threshold "-100" in Massachusetts Bay. The deeper color shows the vortex area.

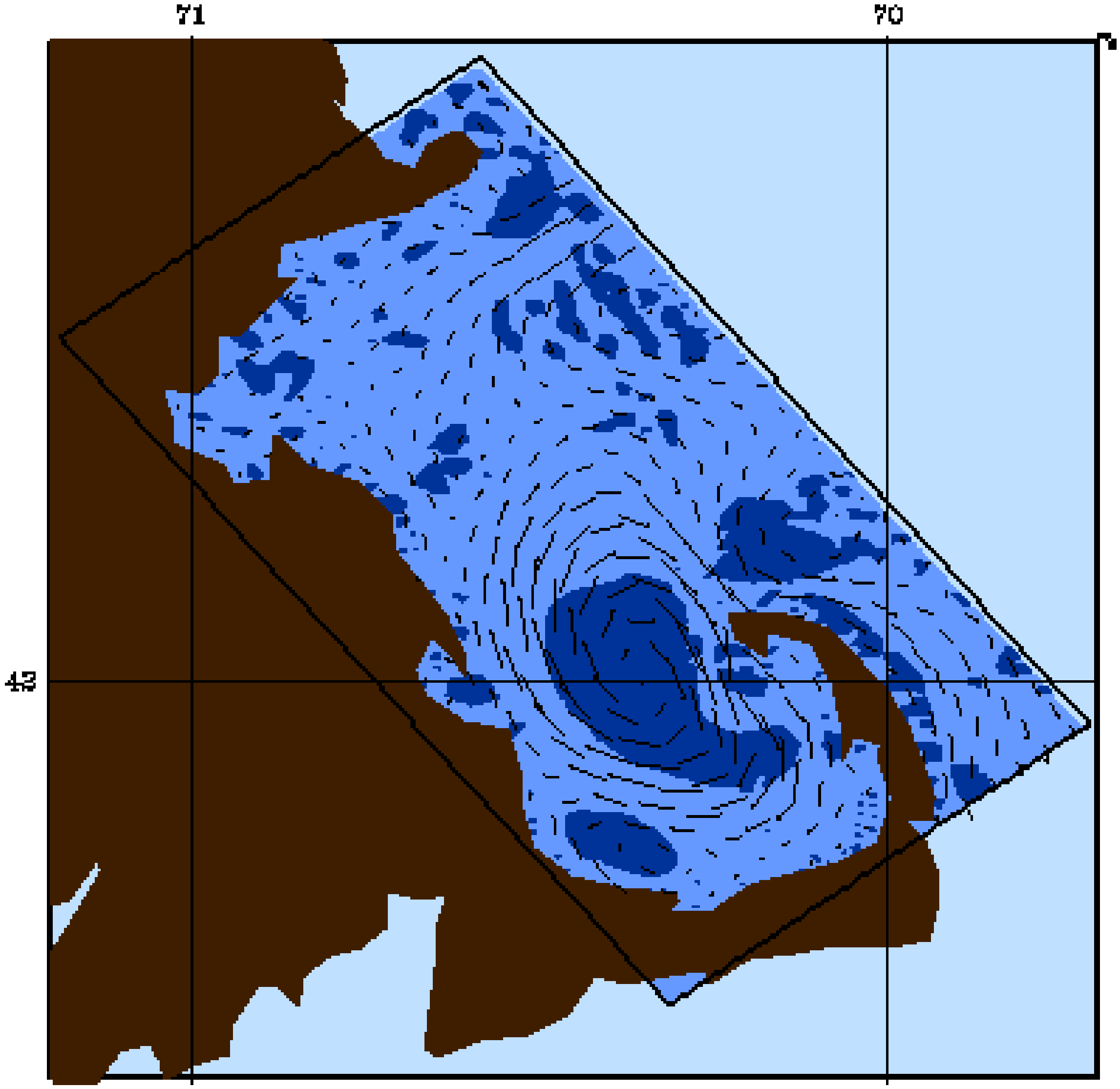


Figure 3-20: The result of the negative λ_2 method with the threshold "-500" in Massachusetts Bay. The deeper color shows the vortex area.

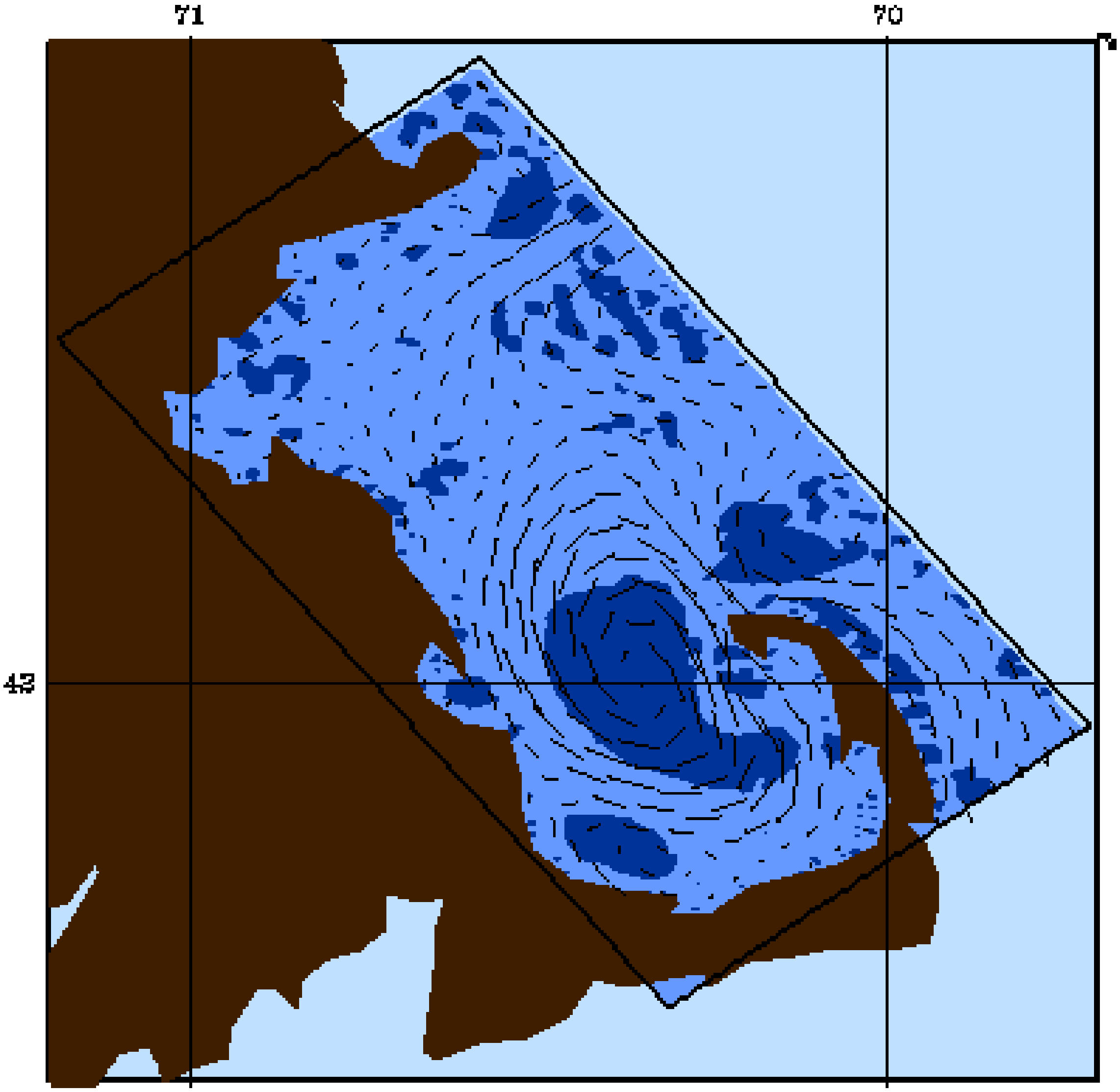


Figure 3-21: The result of the negative λ_2 method with the threshold "-1000" in Massachusetts Bay. The deeper color shows the vortex area.

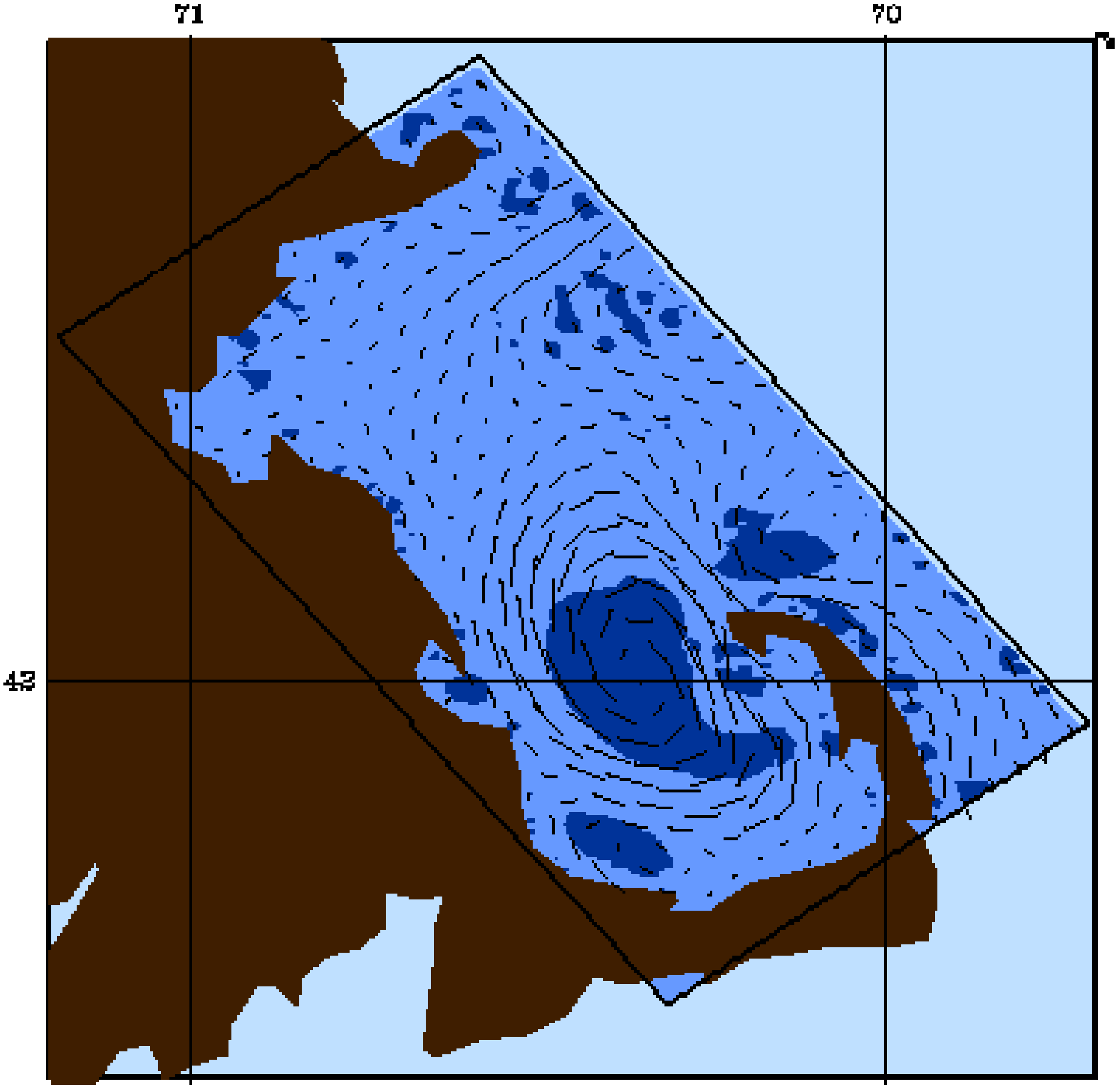


Figure 3-22: The result of the negative λ_2 method with the threshold "-5000" in Massachusetts Bay. The deeper color shows the vortex area.

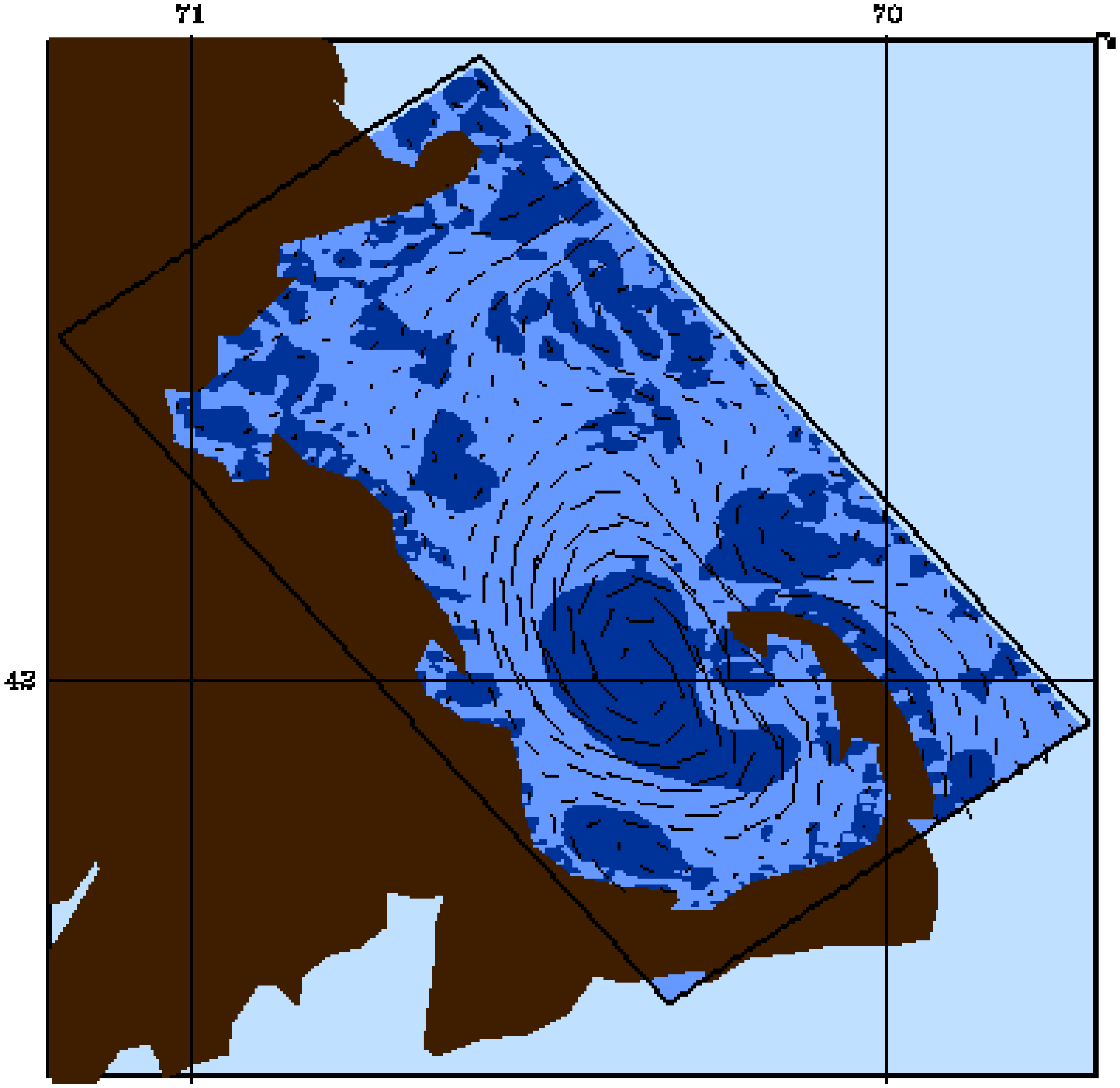


Figure 3-23: The result of the positive Q method with the threshold “0” in Massachusetts Bay. The deeper color shows the vortex area.

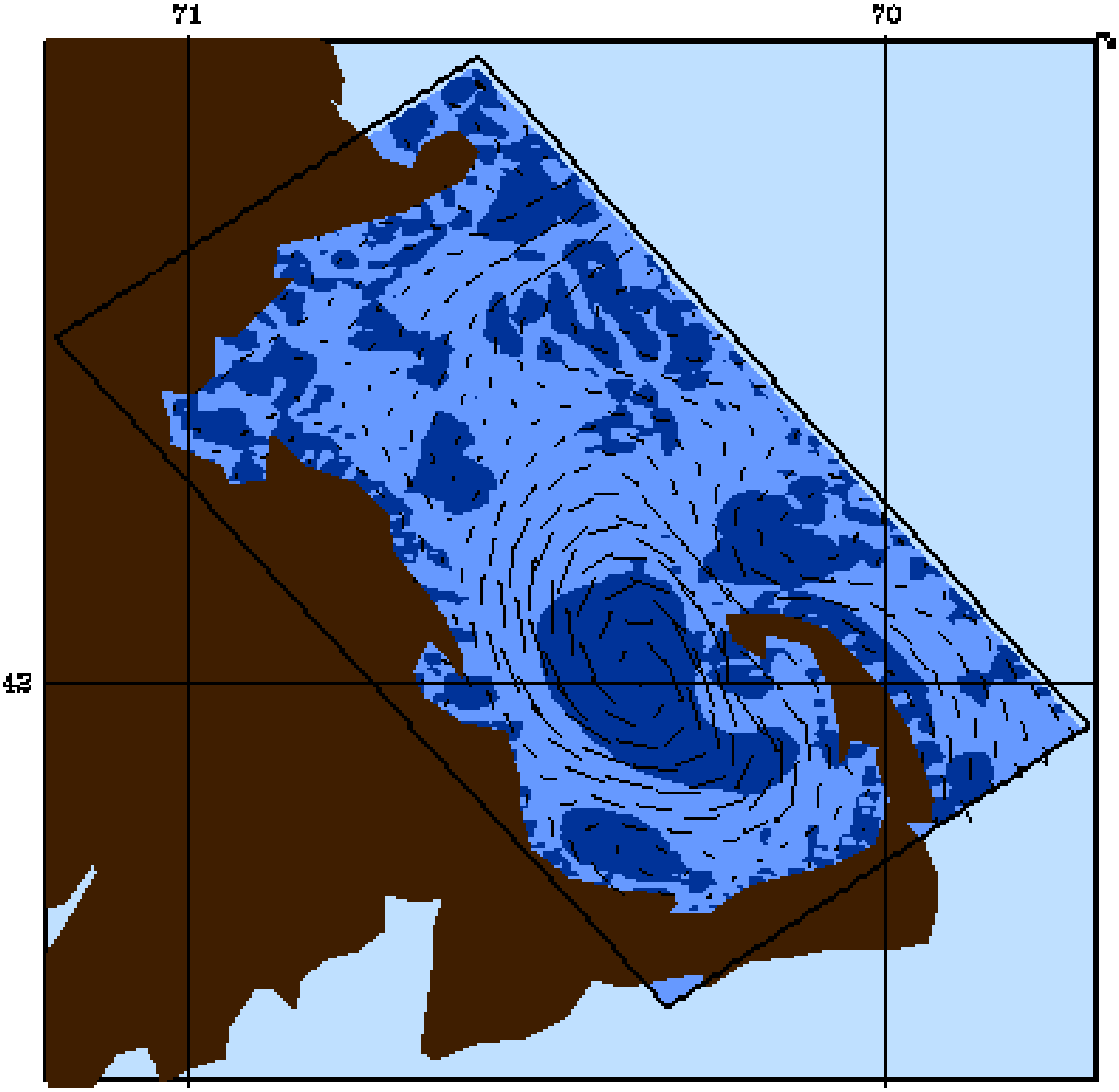


Figure 3-24: The result of the positive Q method with the threshold "100" in Massachusetts Bay. The deeper color shows the vortex area.

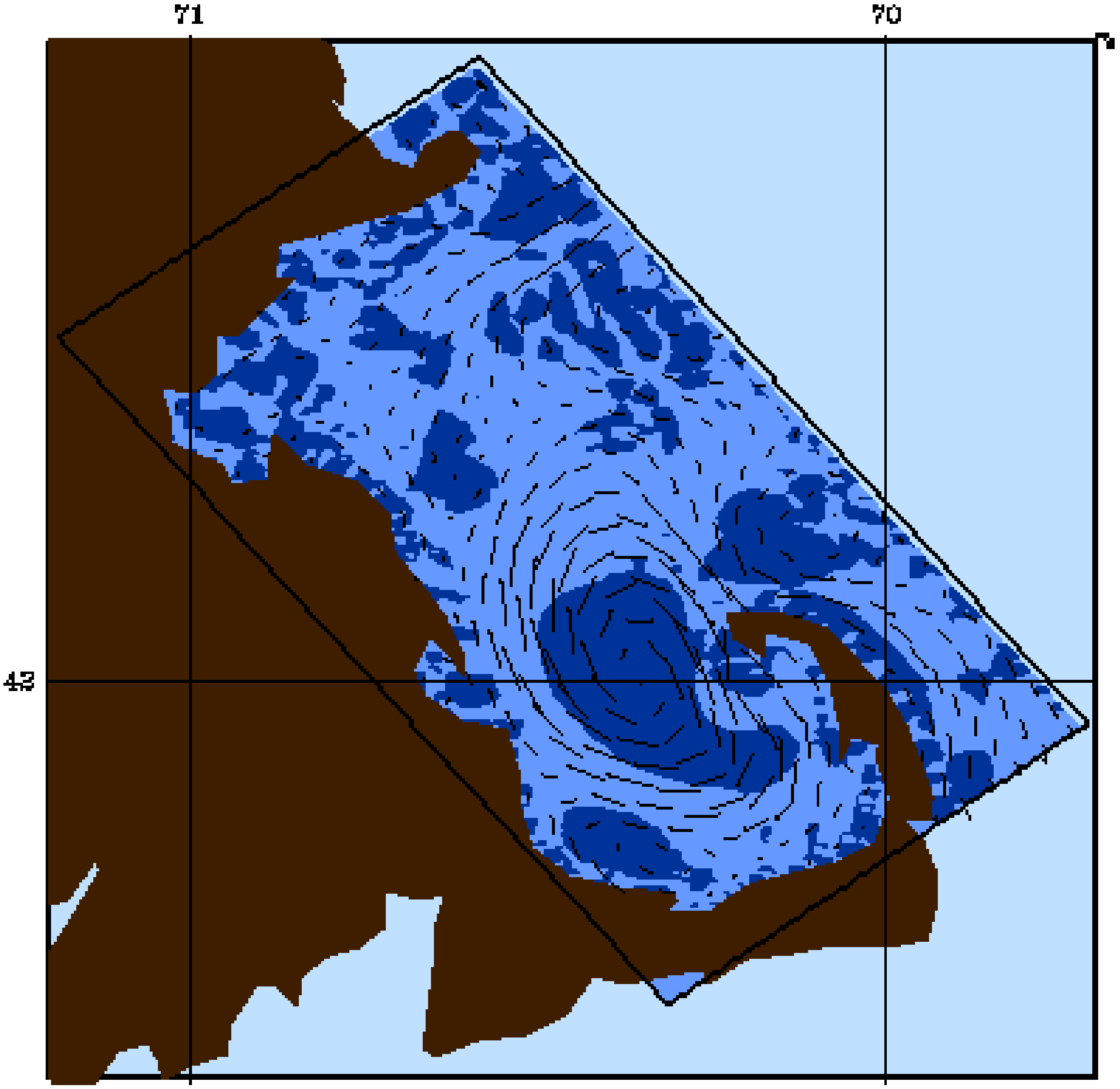


Figure 3-25: The result of the positive Q method with the threshold "1000" in Massachusetts Bay. The deeper color shows the vortex area.

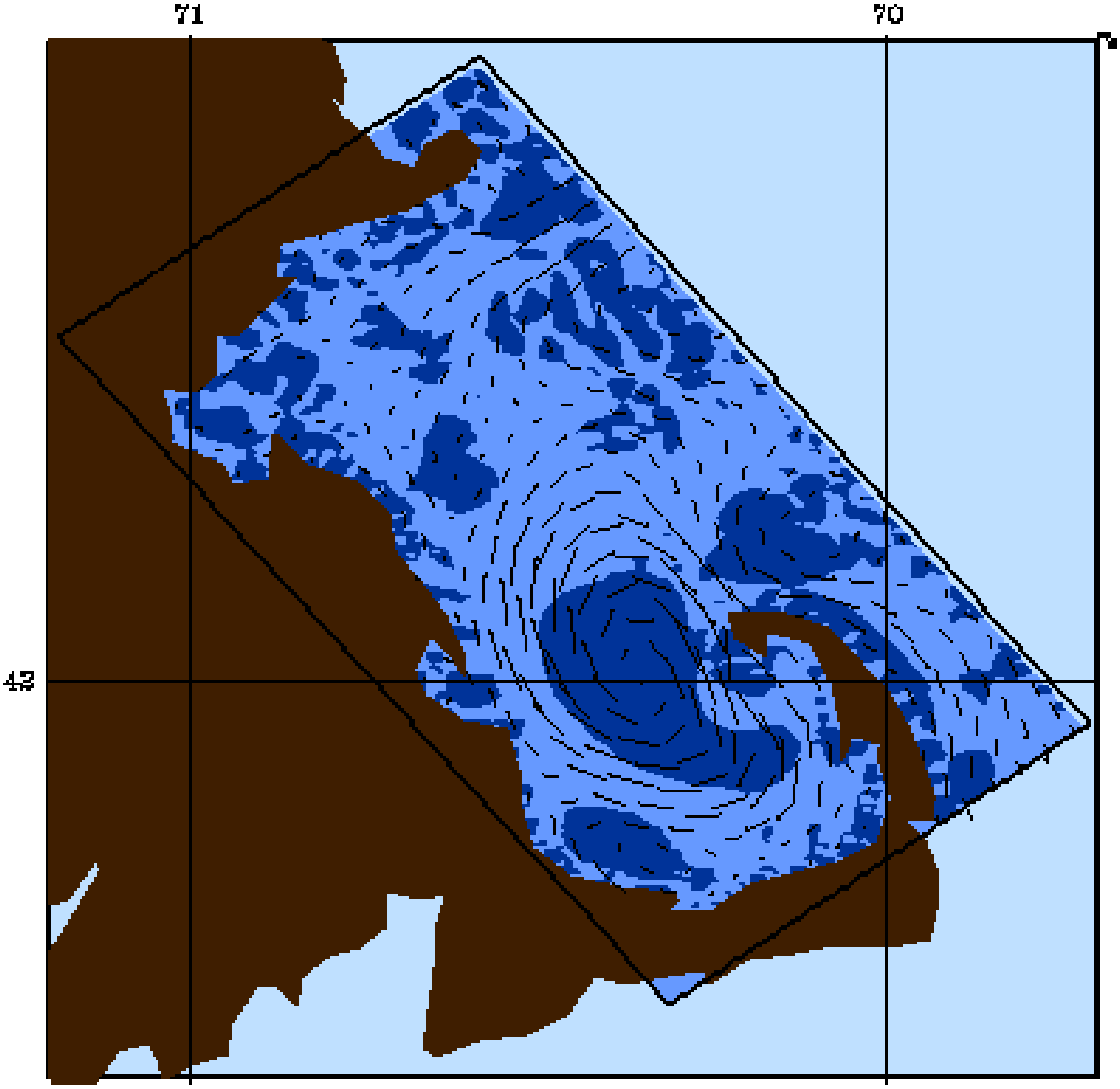


Figure 3-26: The result of the positive Q method with the threshold “5000” in Massachusetts Bay. The deeper color shows the vortex area.

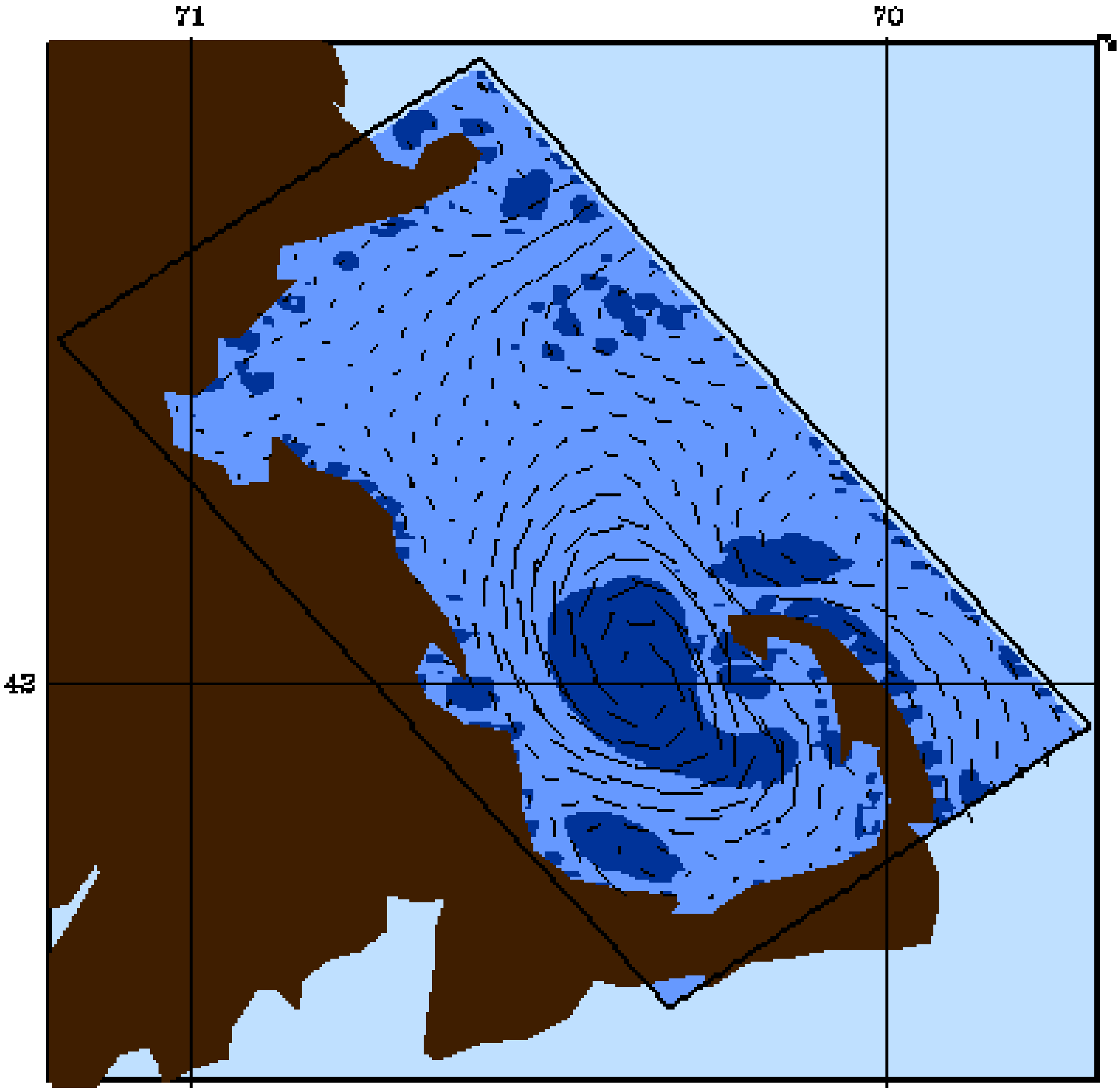


Figure 3-27: The result of the positive Q method with the threshold "10^8" in Massachusetts Bay. The deeper color shows the vortex area.

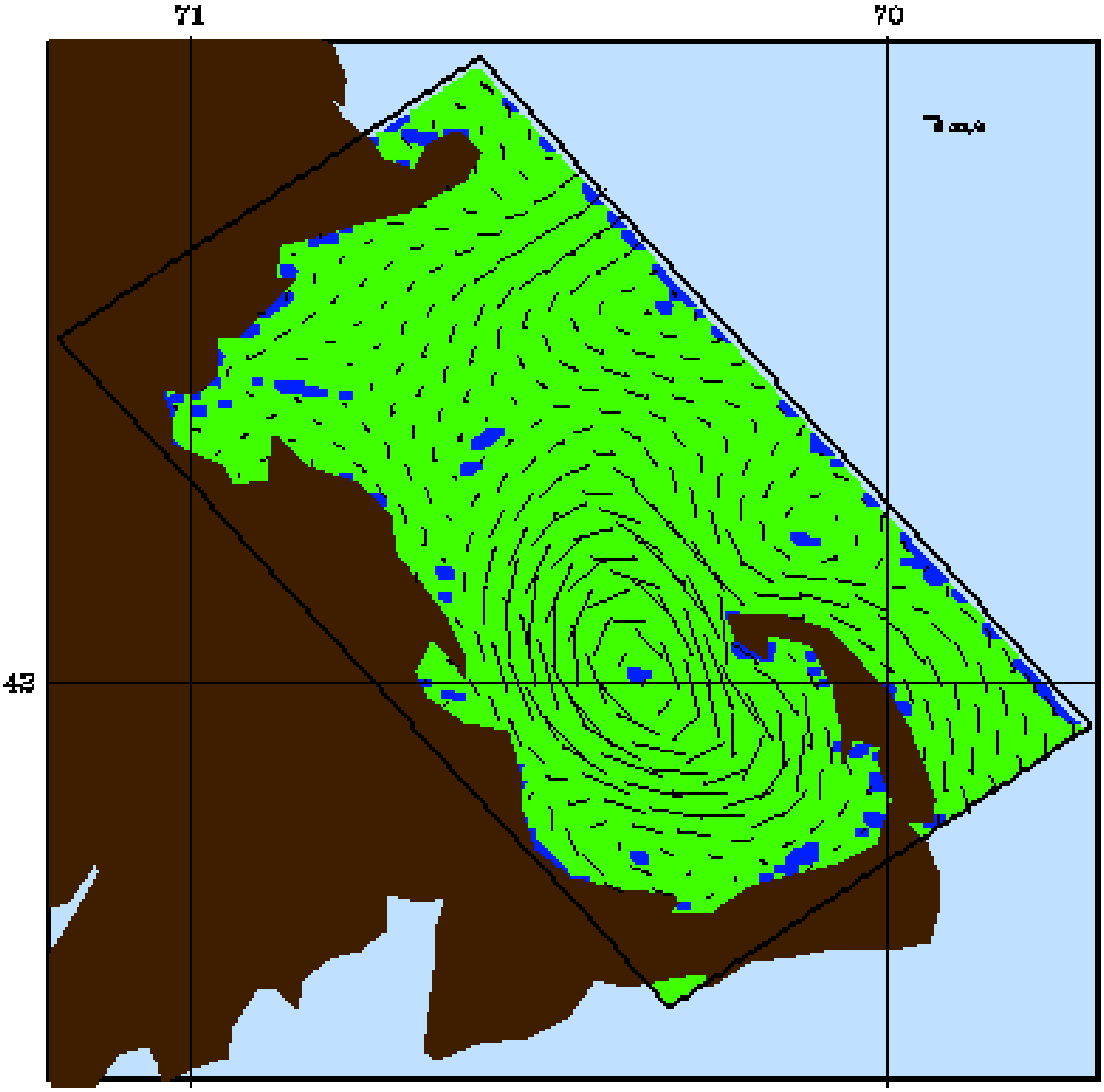


Figure 3-28: The result of the geometric method without cleanup in Massachusetts Bay.

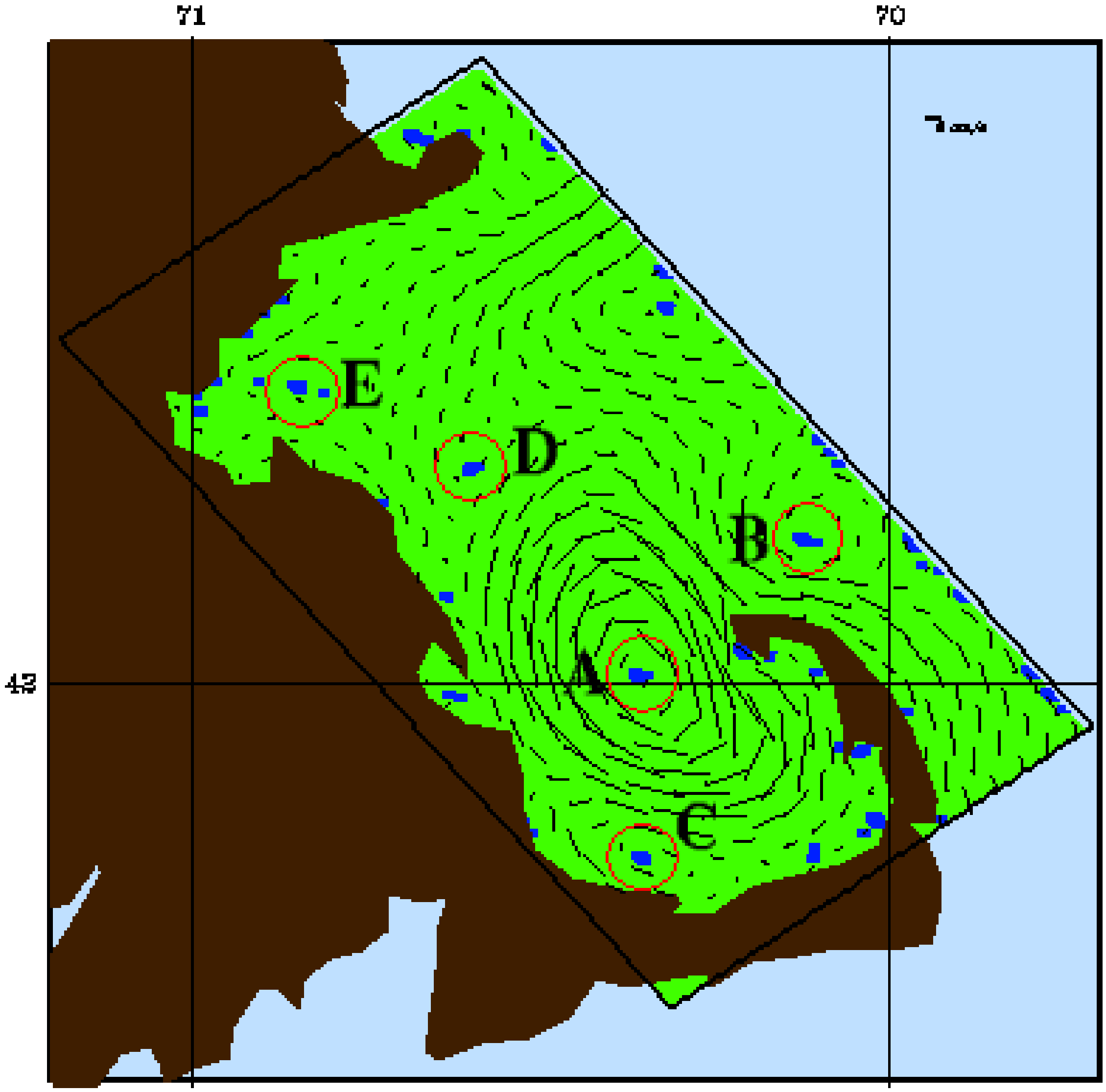


Figure 3-29: The result of the geometric method with the cleanup condition "conjugate pair eigenvalues" (condition A) in Massachusetts Bay. See next Figure for selected closeups of A, B, C, D, and E.

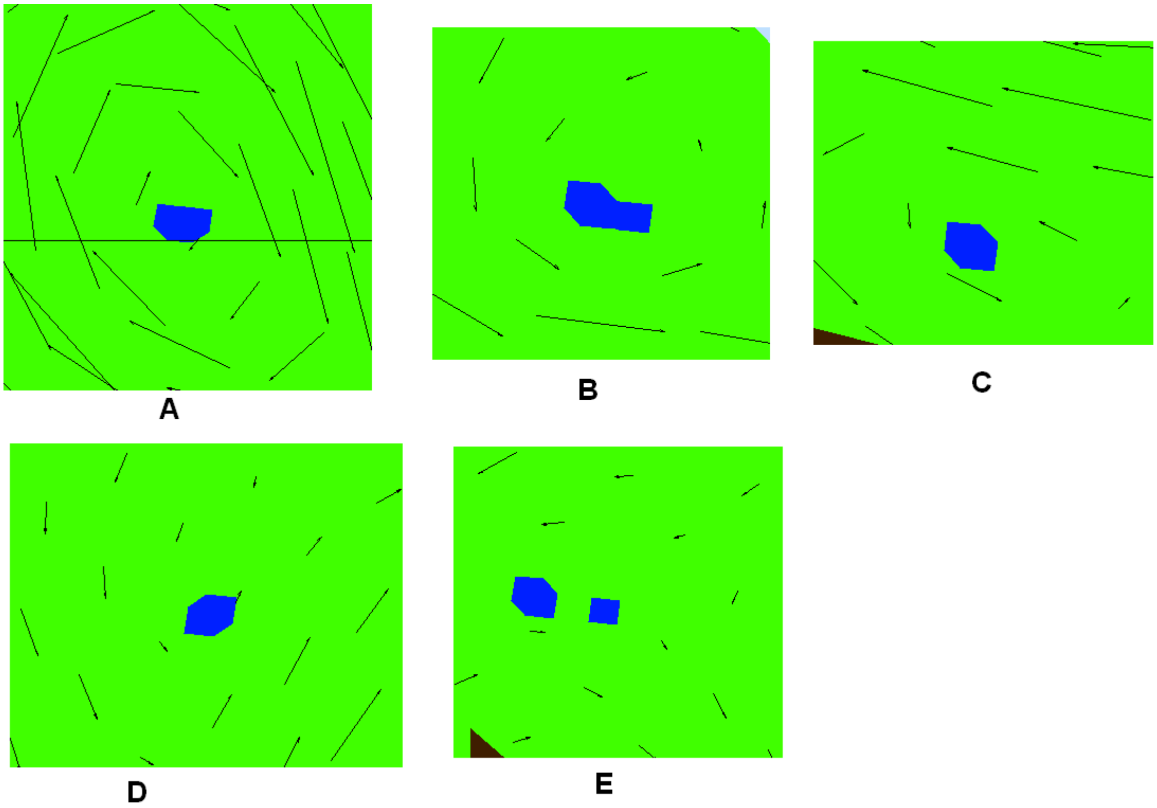


Figure 3-30: The closeups of last Figure.

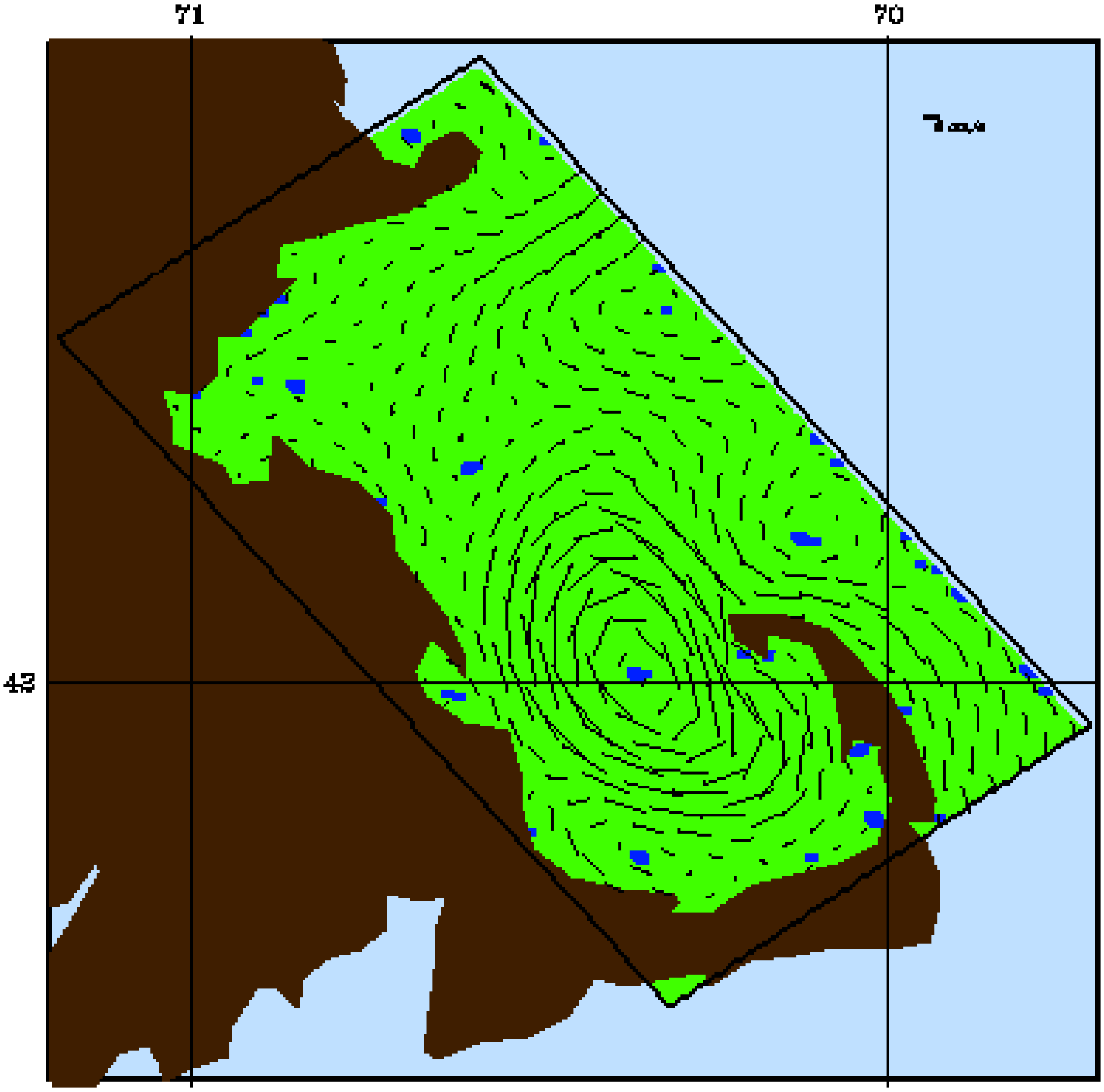


Figure 3-31: The result of the geometric method with the cleanup condition "negative λ_2" (condition B) in Massachusetts Bay.

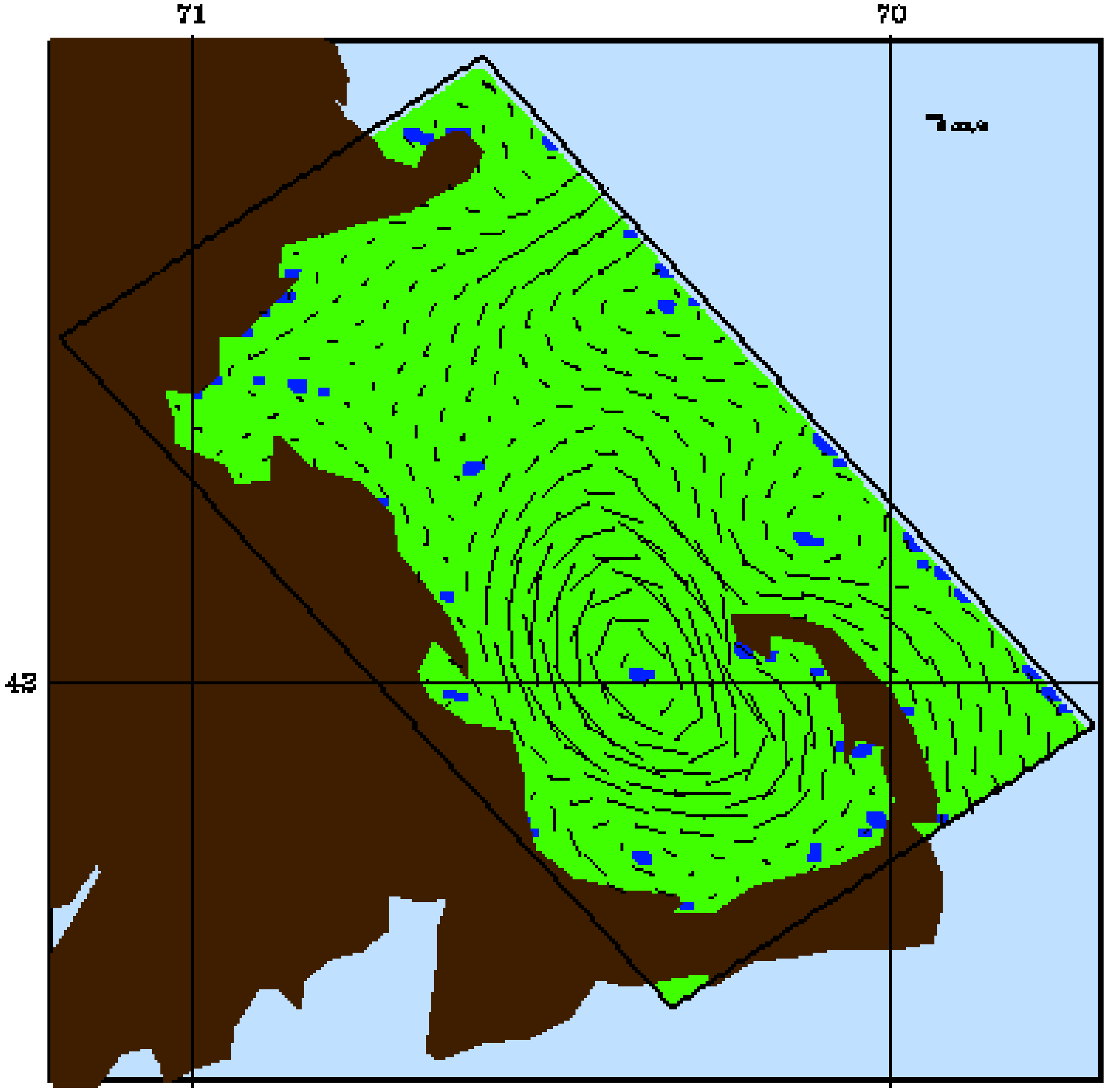


Figure 3-32: The result of the geometric method with the cleanup condition "positive Q" (condition C) in Massachusetts Bay.

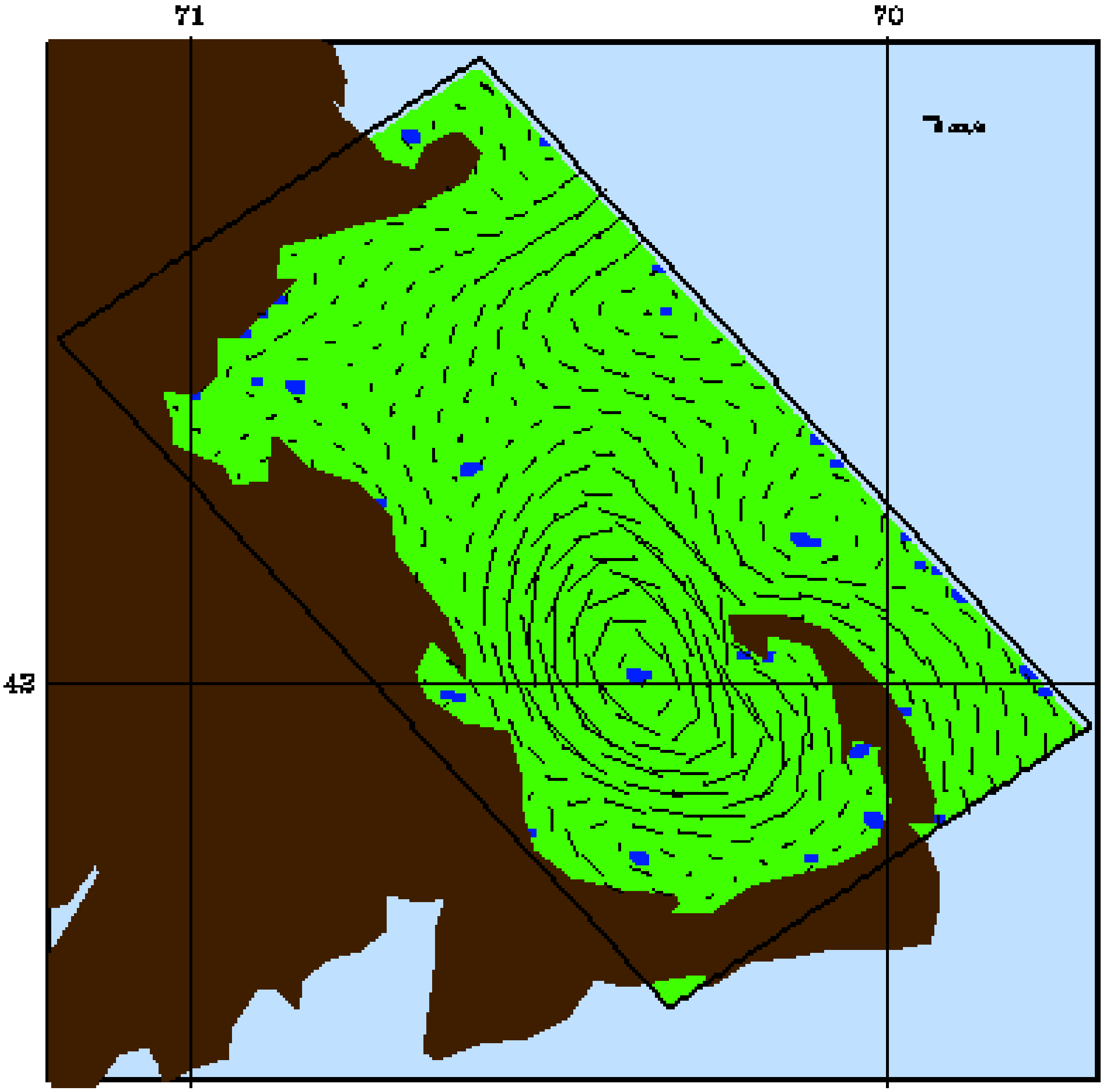


Figure 3-33: The result of the geometric method with the cleanup condition "$A \cap B$" in Massachusetts Bay.

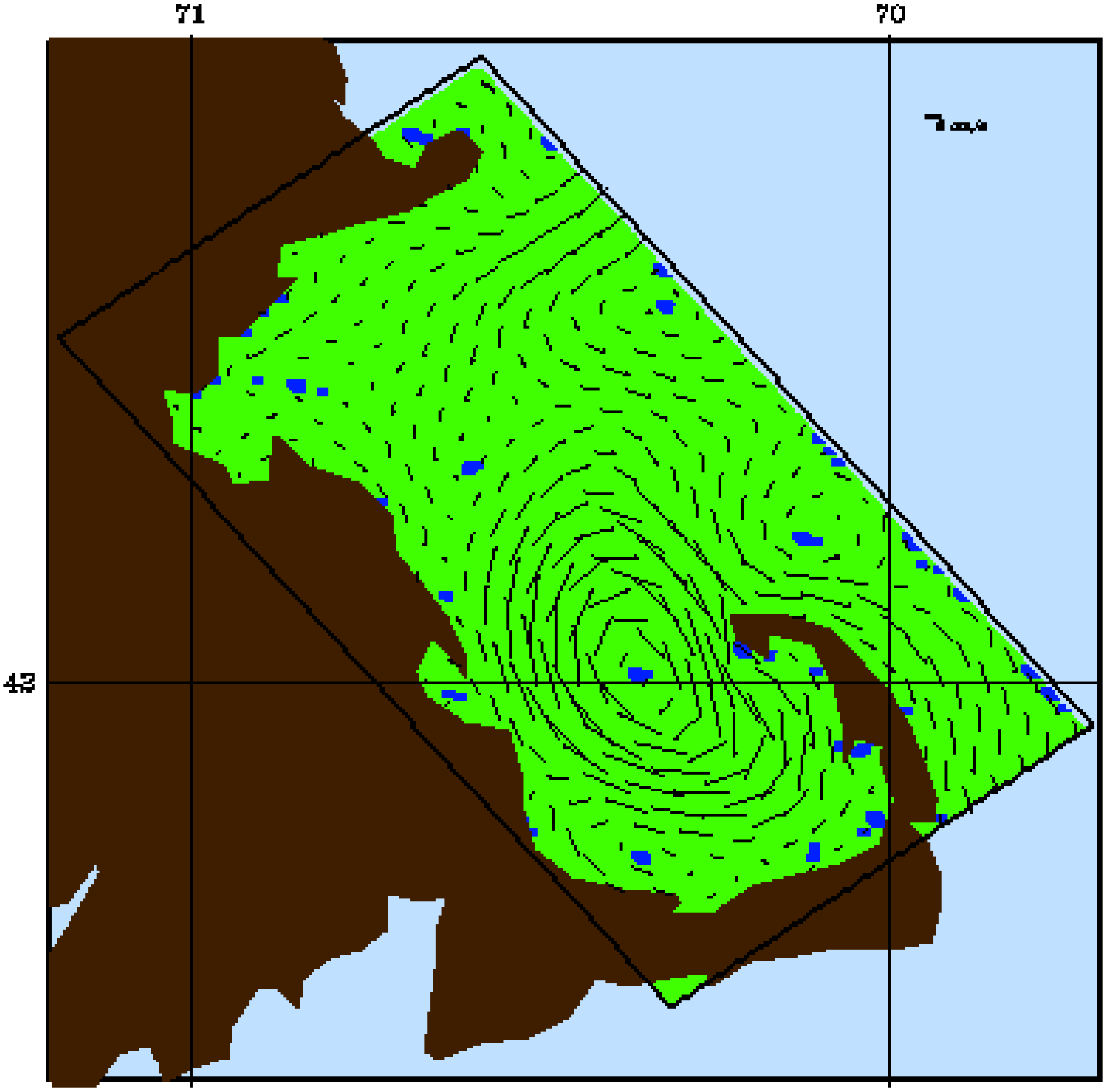


Figure 3-34: The result of the geometric method with the cleanup condition "$A \cap C$" in Massachusetts Bay.

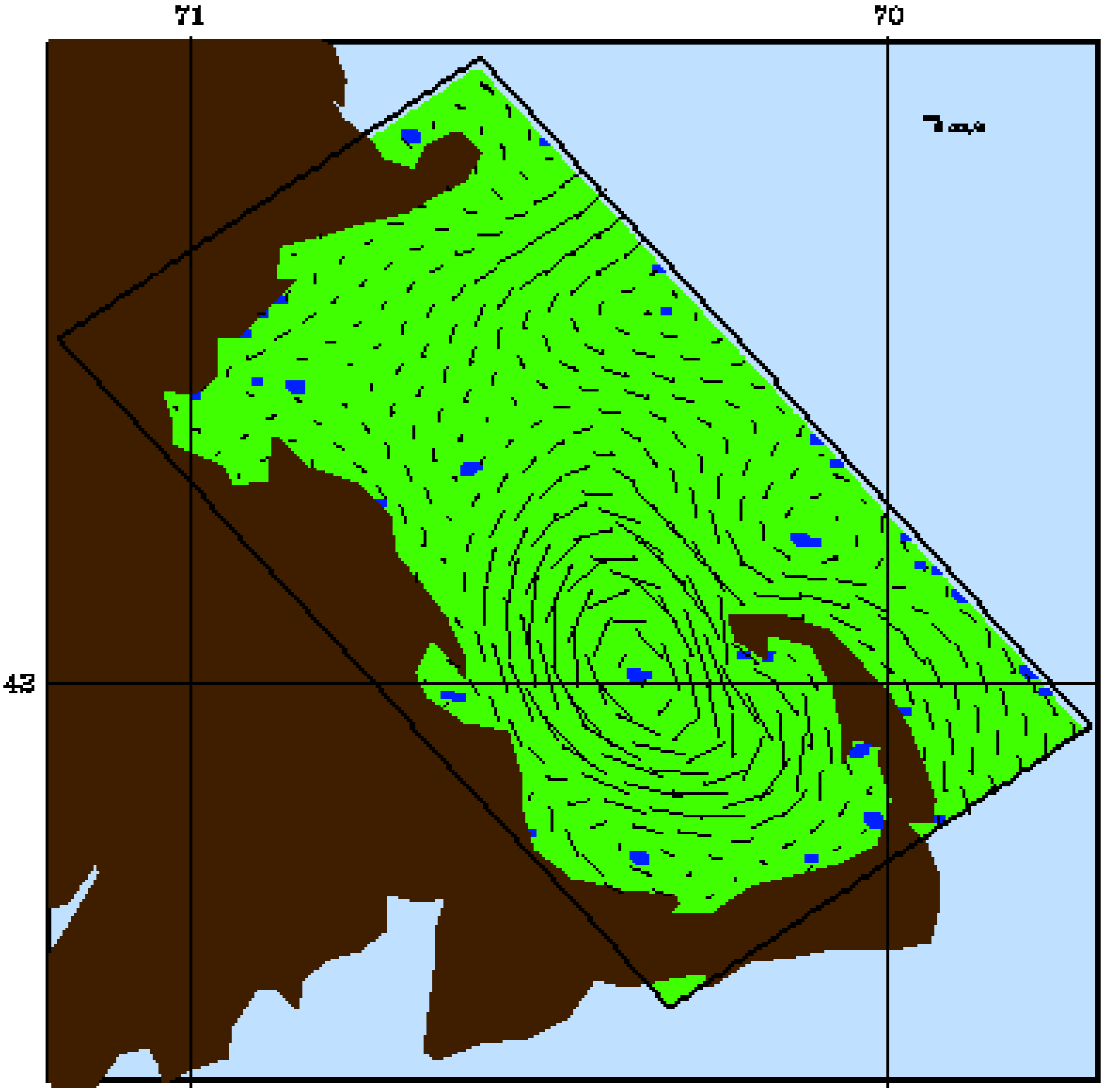


Figure 3-35: The result of the geometric method with the cleanup condition "$B \cap C$" in Massachusetts Bay.

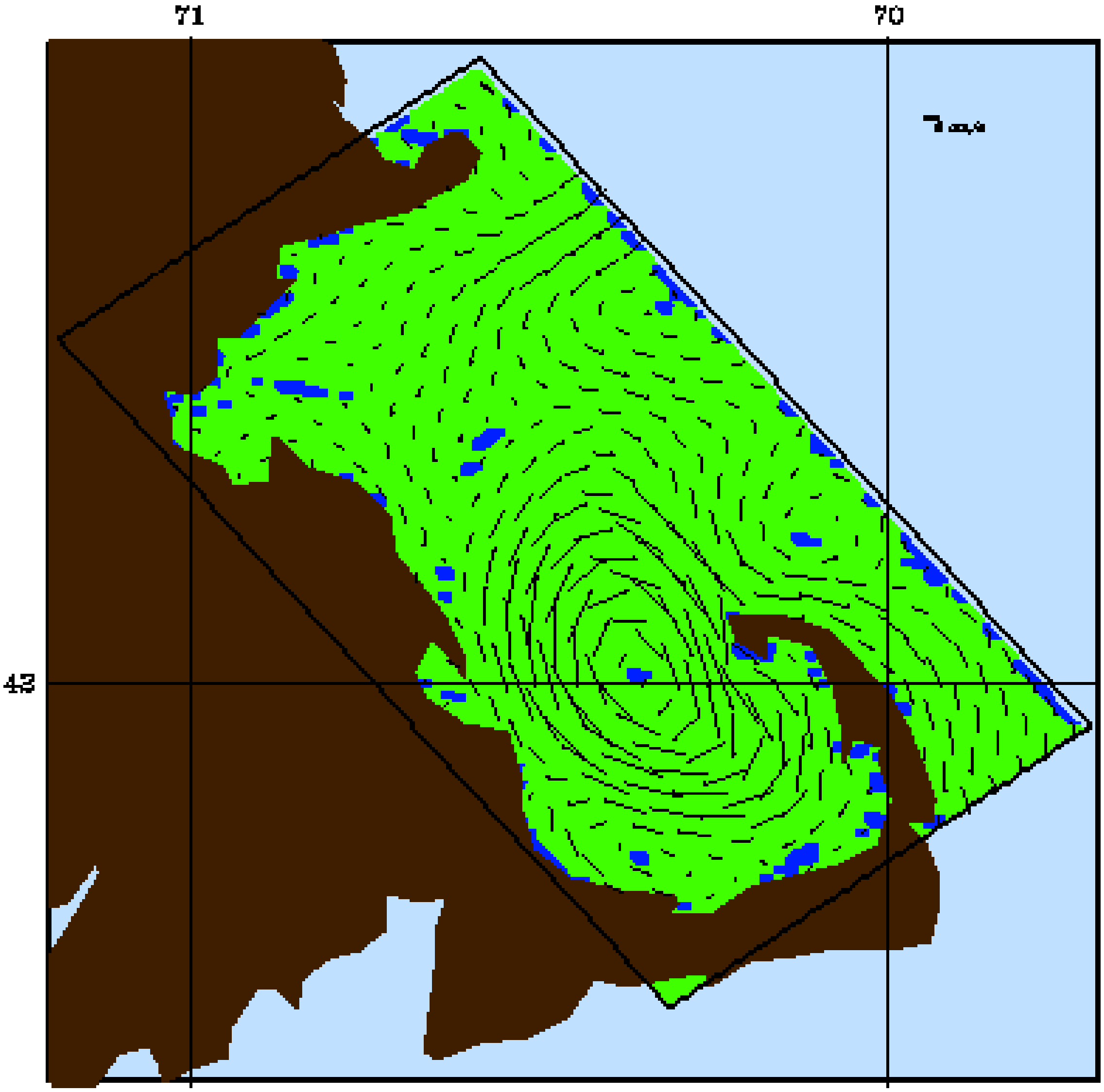


Figure 3-36: The result of the geometric method with the cleanup condition "$A \cup B \cup C$" in Massachusetts Bay.

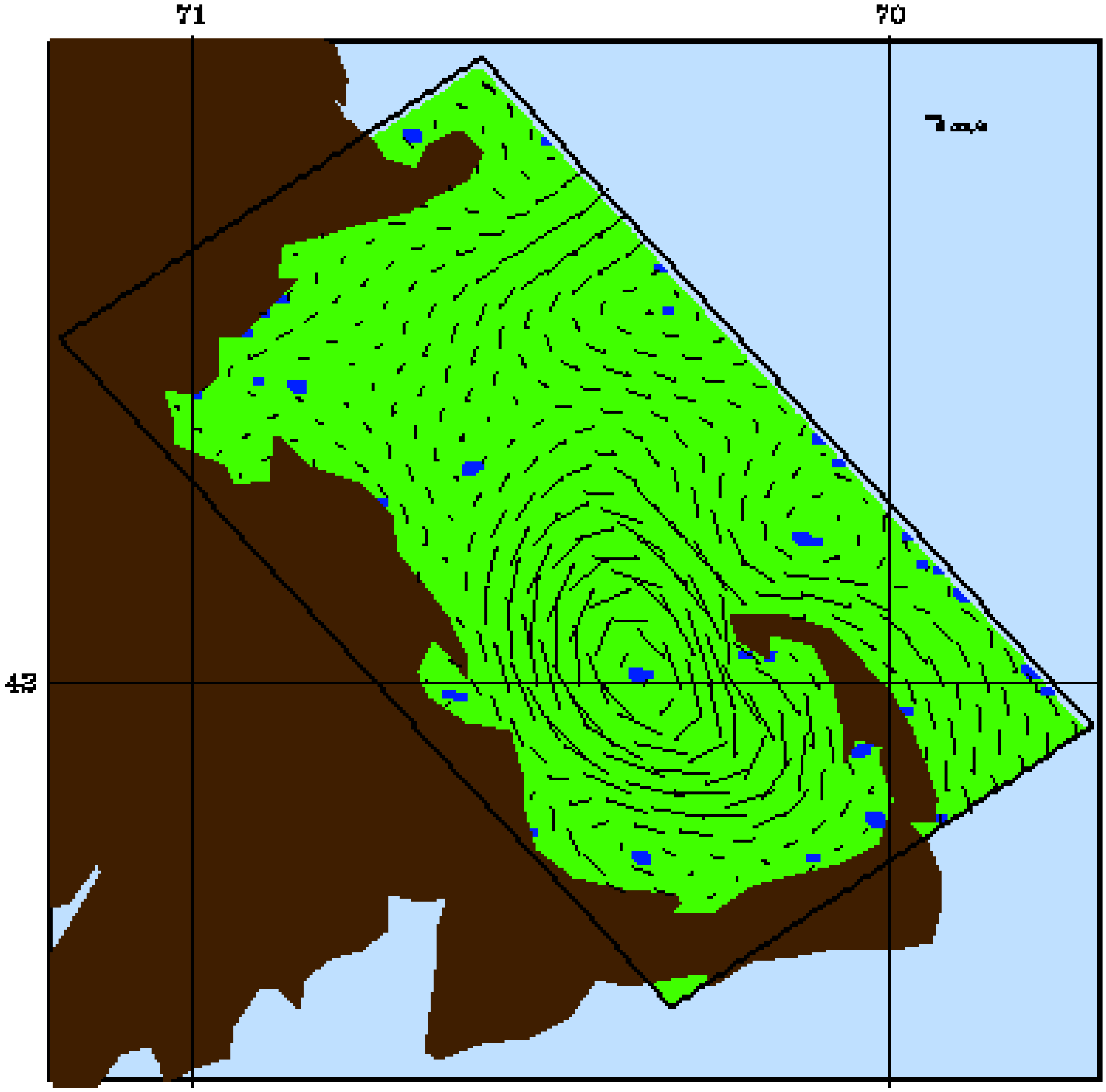


Figure 3-37: The result of the geometric method with the cleanup condition “$A \cap B \cap C$” in Massachusetts Bay.

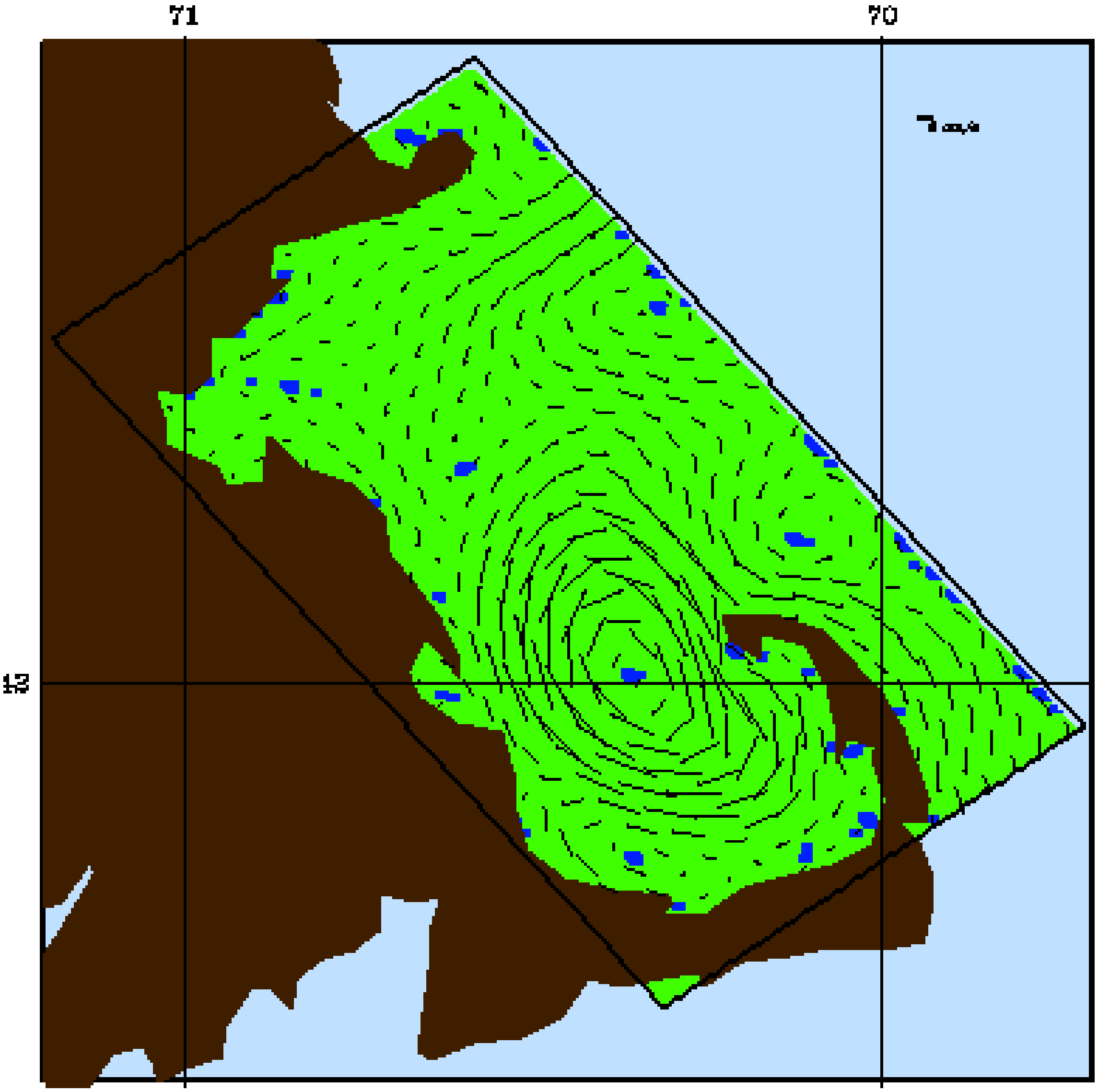


Figure 3-38: The result of the geometric method with the cleanup condition "$A \cap (B \cup C)$" in Massachusetts Bay.

Western Mediterranean

The main attributes of the Western Mediterranean HOPS dataset from May 7, 2002 are:

tlat (number of tracer points in the latitude direction) = 114 ;

tlon (number of tracer points in the longitude direction) = 149 ;

vlat (number of velocity points in the latitude direction) = 114 ;

vlon (number of velocity points in the longitude direction) = 149 ;

level (input vertical levels or depths) = 20 ;

outlev (output vertical levels or depths) = 20 ;

time (the number of days since initialization) = 21;

rlngd (transformation centroid longitude) = 10.16338^o (East 10.16338 degree) ;

rlatd (transformation centroid latitude) = 42.94479^o (North 42.94479 degree) ;

thetad (domain rotation angle) = 0^o ;

coord (flag for type of horizontal grid) = 0 (Cartesian (tangent plane) grid) ;

The visualization of the HOPS horizontal velocity field in the Western Mediterranean is shown in Figure 3-39 on page 101.

Vortex detection in the Western Mediterranean using flow fields generated by HOPS and employing the vorticity magnitude method is shown in Figure 3-40 on page 102, where the threshold is twice the mean of the vorticity field. Many false positives are detected in addition to vortex regions. The minimum value of the vorticity magnitude is 0.0 at point (1, 1), the maximum value is 3.11248486×10^3 at point (41, 105), the mean value is 7.78437653×10 and the standard deviation is 1.29923126×10^2.

Vortex detection in the Western Mediterranean employing the conjugate pair eigenvalues method is shown in Figure 3-41 on page 103. Again, all vortices and some false positives are detected since conjugate eigenvalues are only a necessary condition for a vortex.

Vortex detection employing the negative λ_2 method is shown in Figure 3-42 on page 104, where dark blue parts denote detected vortex regions when using the threshold -1000, while red parts denote detected additional vortex regions when using the theoretical threshold 0. Still, many false positives exist in the domain. In addition, it is hard to distinguish different vortices. The minimum value of λ_2 is -3.02418250×10^5 at point (36, 64), the maximum value is 1.83720953×10^5 at point (53, 9), the mean value is 6.09442566×10^2 and the standard deviation is 5.73457568×10^3.

Figure 3-43 on page 105 shows the result of the positive Q method, where dark blue parts denote detected vortex regions when using the threshold 10^6, while red parts denote detected additional vortex regions when using the threshold 0. Similar problems exist as in the negative λ_2 method. The minimum value of Q is -6.69282611×10^9 at point (37, 64), the maximum value is $1.96848943 \times 10^{11}$ at point (36, 64), the mean value is 2.44103080×10^7 and the standard deviation is 1.58065344×10^9.

Figure 3-44 on page 106 shows the result of the geometric-based vortex core detection method without cleanup, which detects vortex core regions. Some false positives are detected as well as all vortex regions in the domain.

Figure 3-45 on page 107 shows the result of the geometric-based vortex core detection method with cleanup. False positives are removed by using the conjugate eigenvalues condition. Also, two vortex cores close to each other are well distinguished.

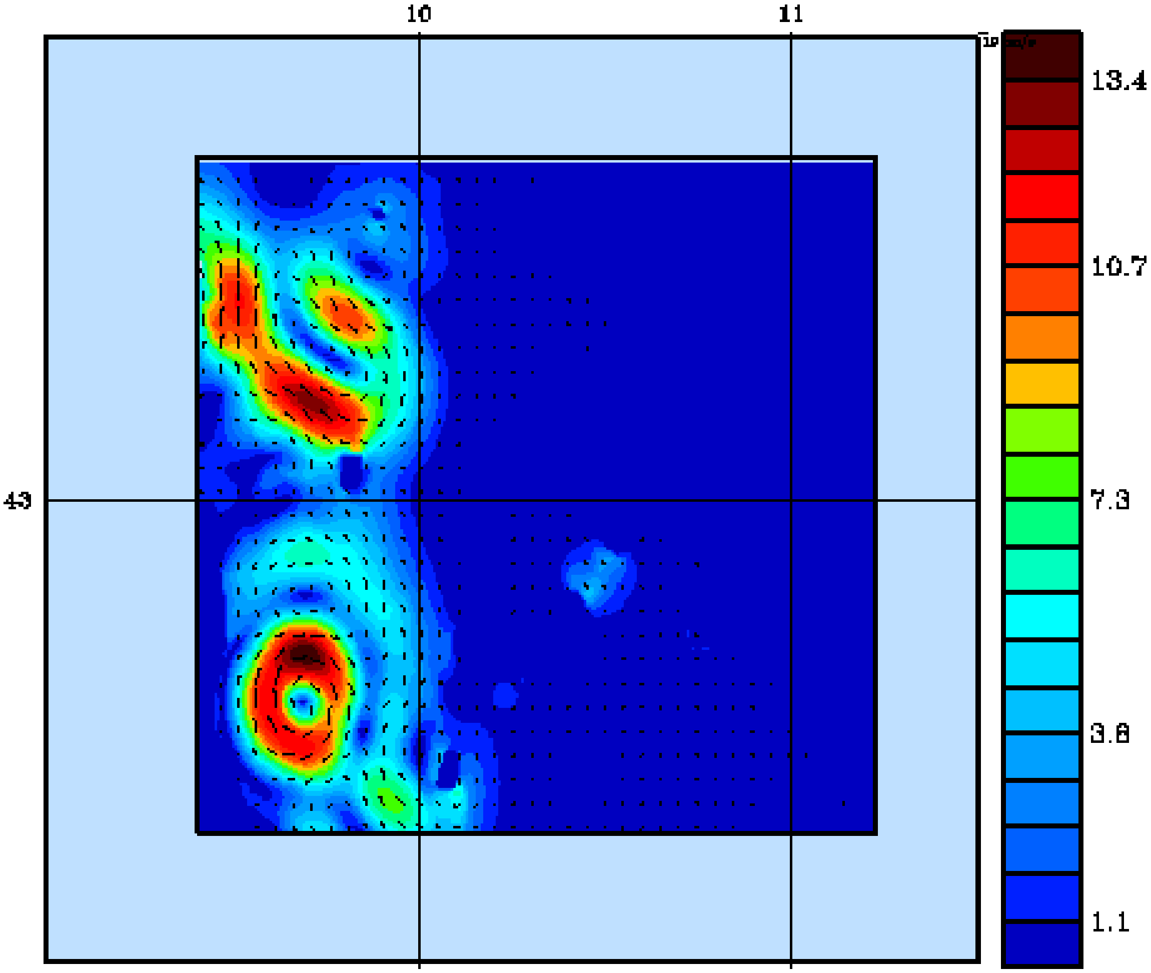


Figure 3-39: The HOPS horizontal velocity magnitude and direction field in the Western Mediterranean.

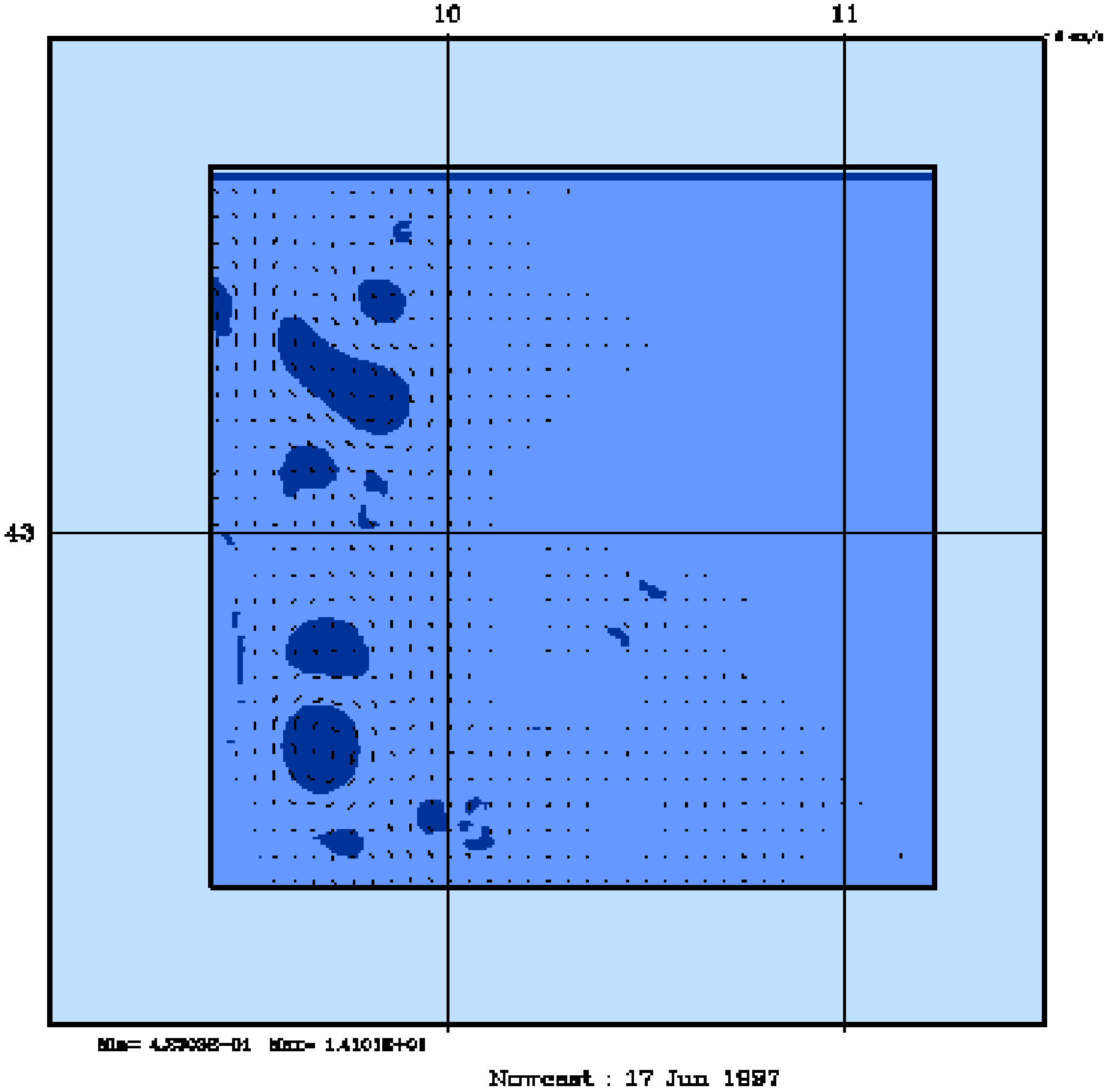


Figure 3-40: The Boolean values of the vorticity magnitude with the threshold of twice the mean of the vorticity field, with superposed velocity vectors in the Western Mediterranean. The deeper color denotes the vortex area.

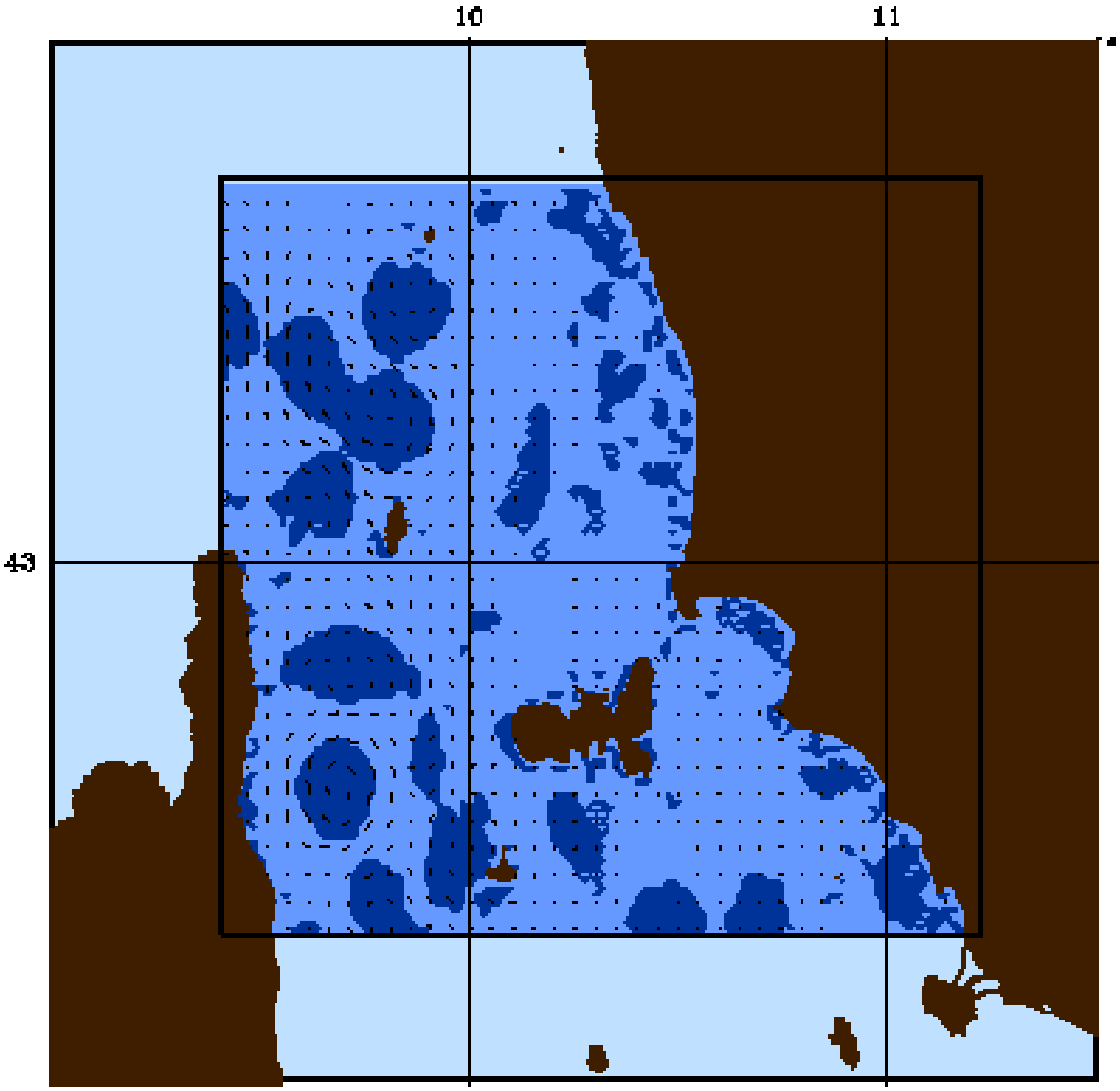


Figure 3-41: The result of the conjugate pair eigenvalues method in the Western Mediterranean. The deeper color shows the vortex area.

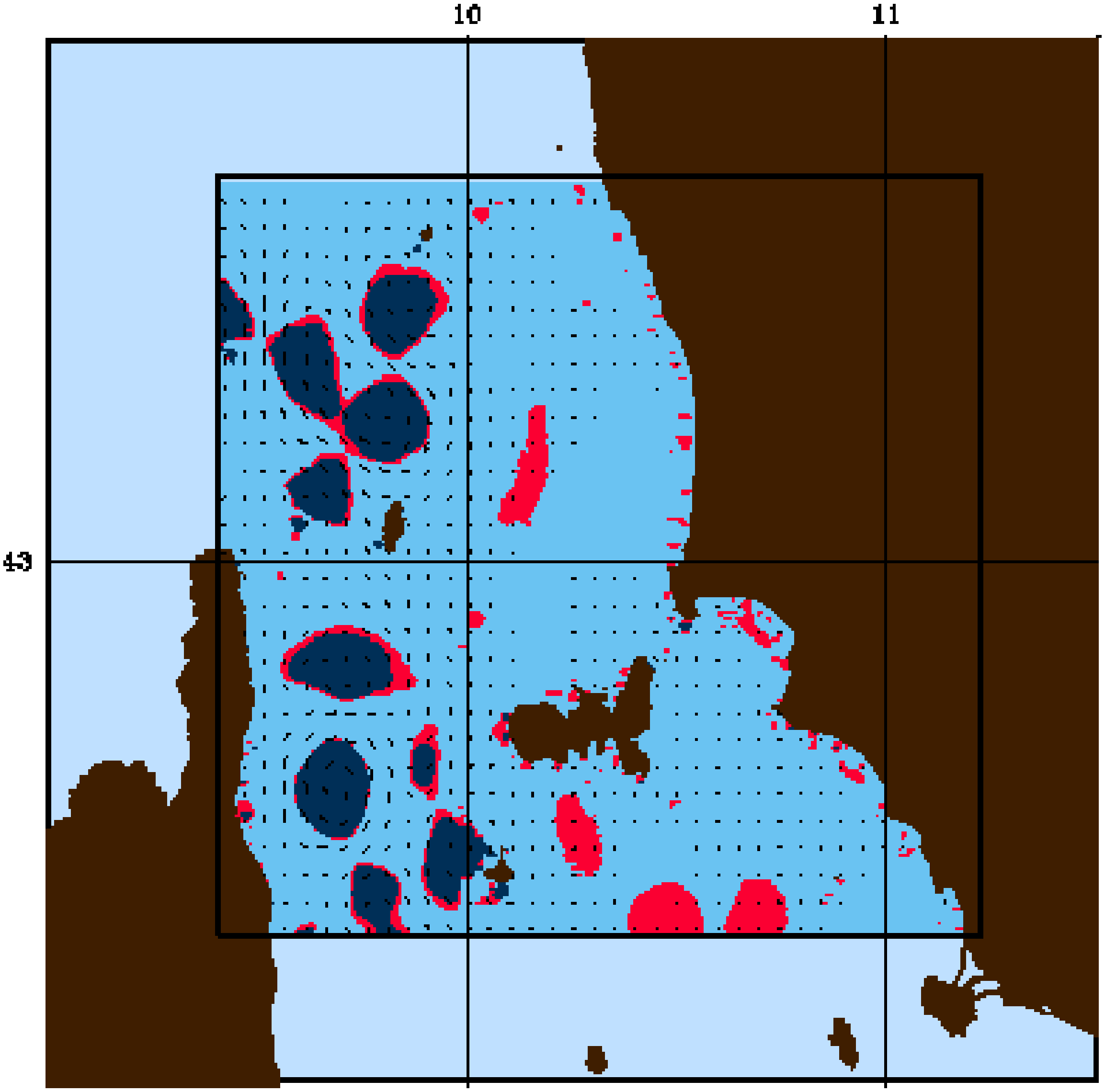


Figure 3-42: The result of the negative λ_2 method with superposed velocity in the Western Mediterranean.

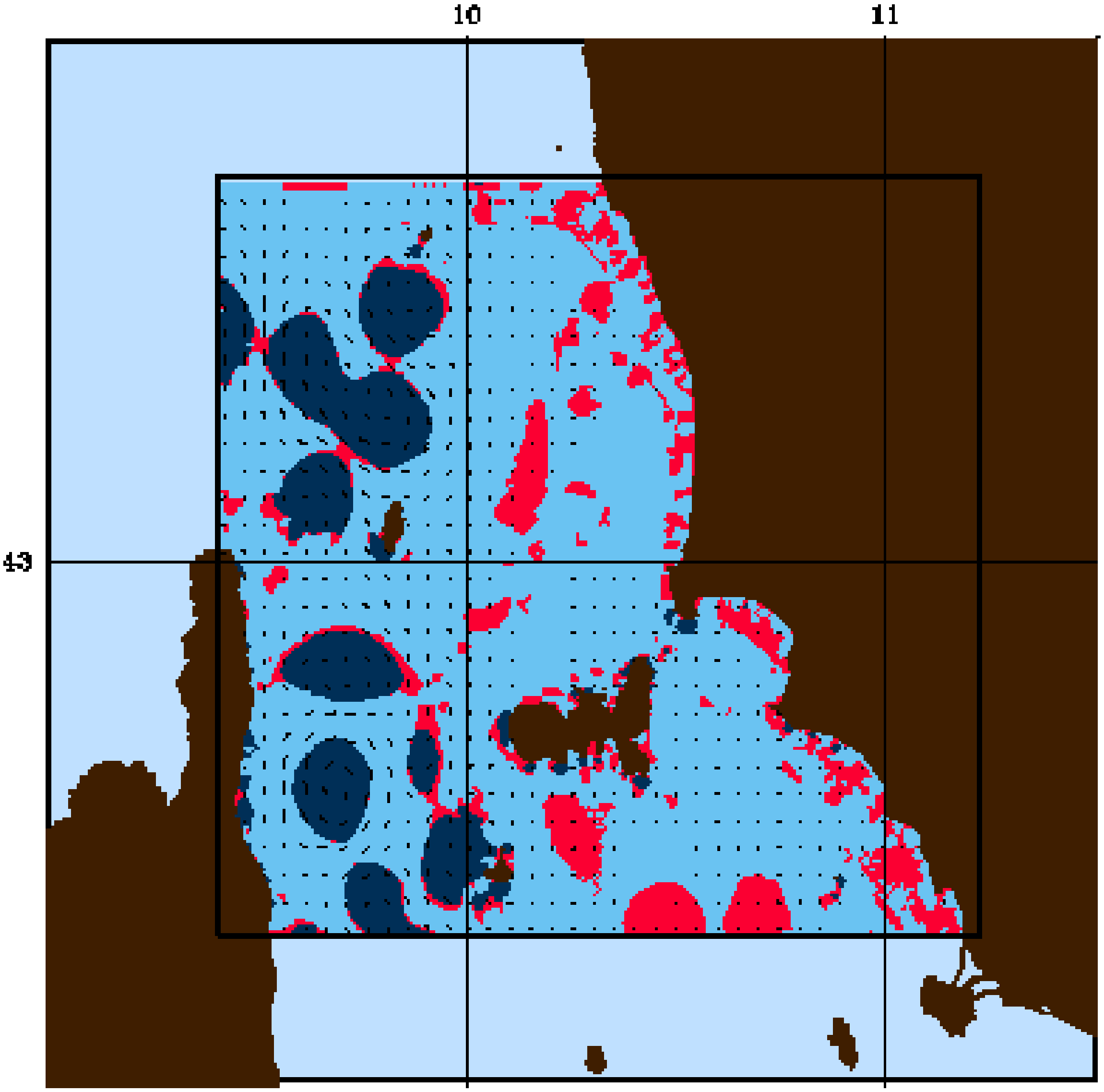


Figure 3-43: The result of the positive Q method with superposed velocity in the Western Mediterranean.

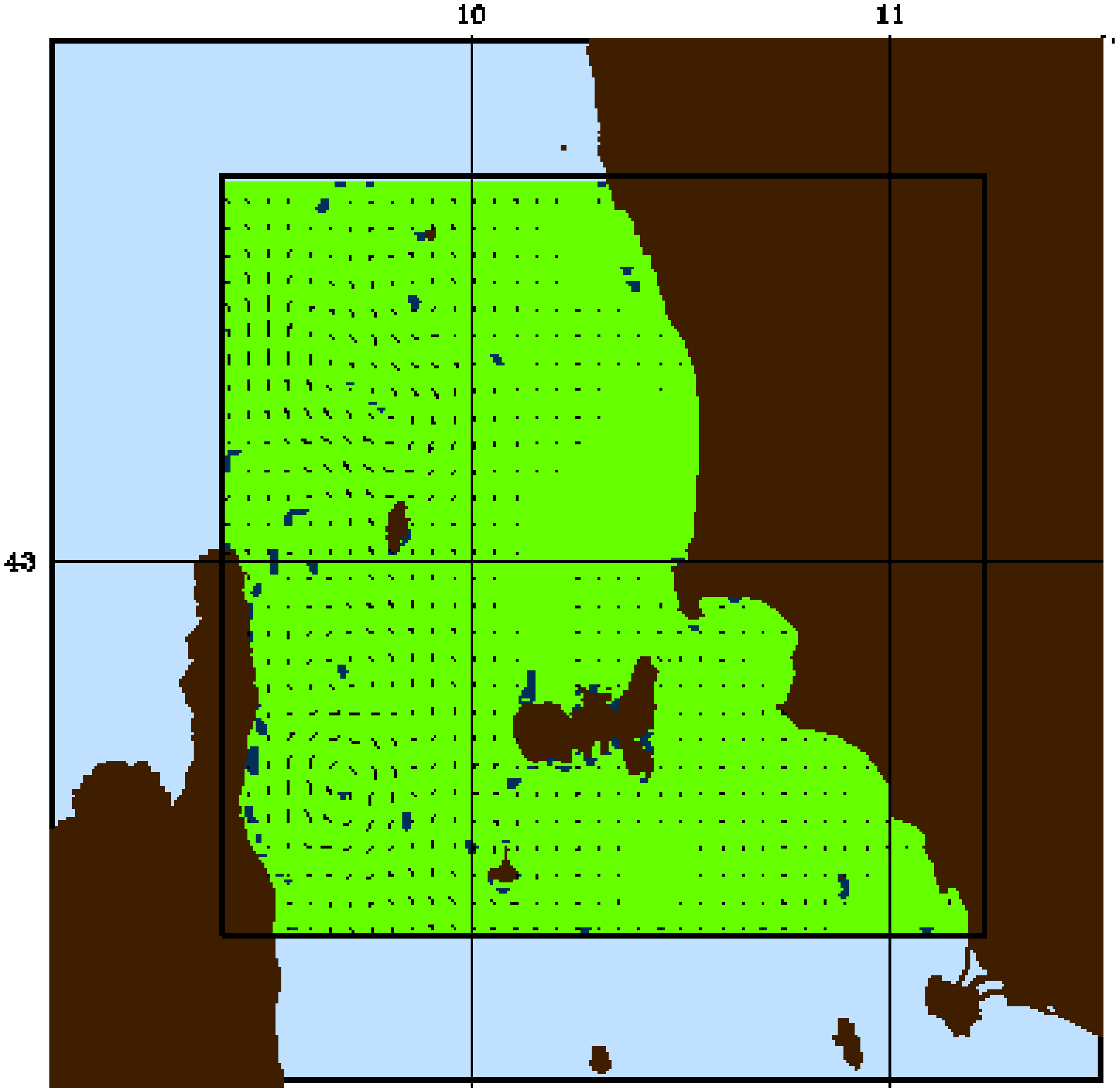


Figure 3-44: The result of the geometric method without cleanup in the Western Mediterranean.

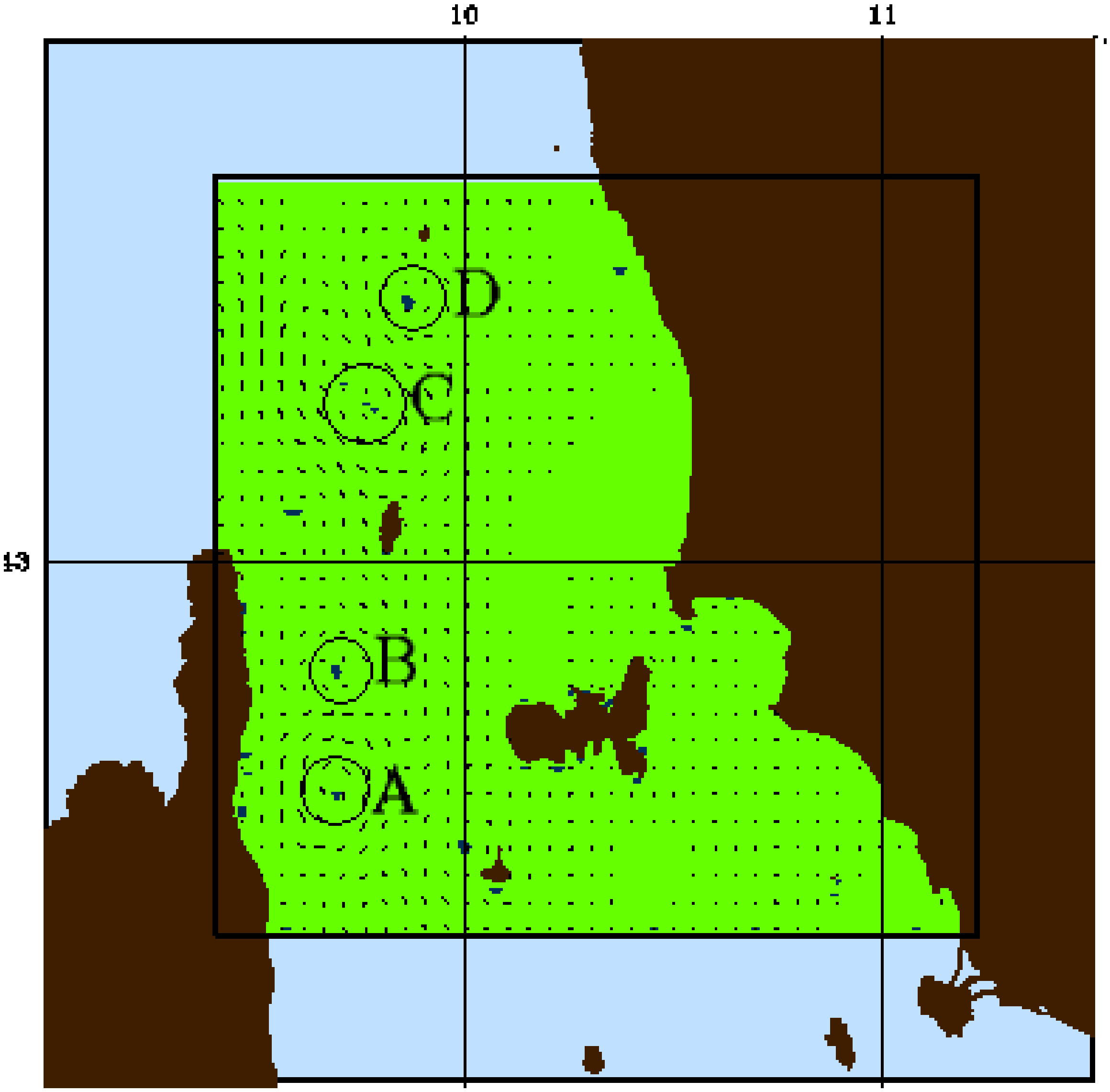


Figure 3-45: The result of the geometric method with cleanup in the Western Mediterranean. See next Figure for closeups of A, B, C and D.

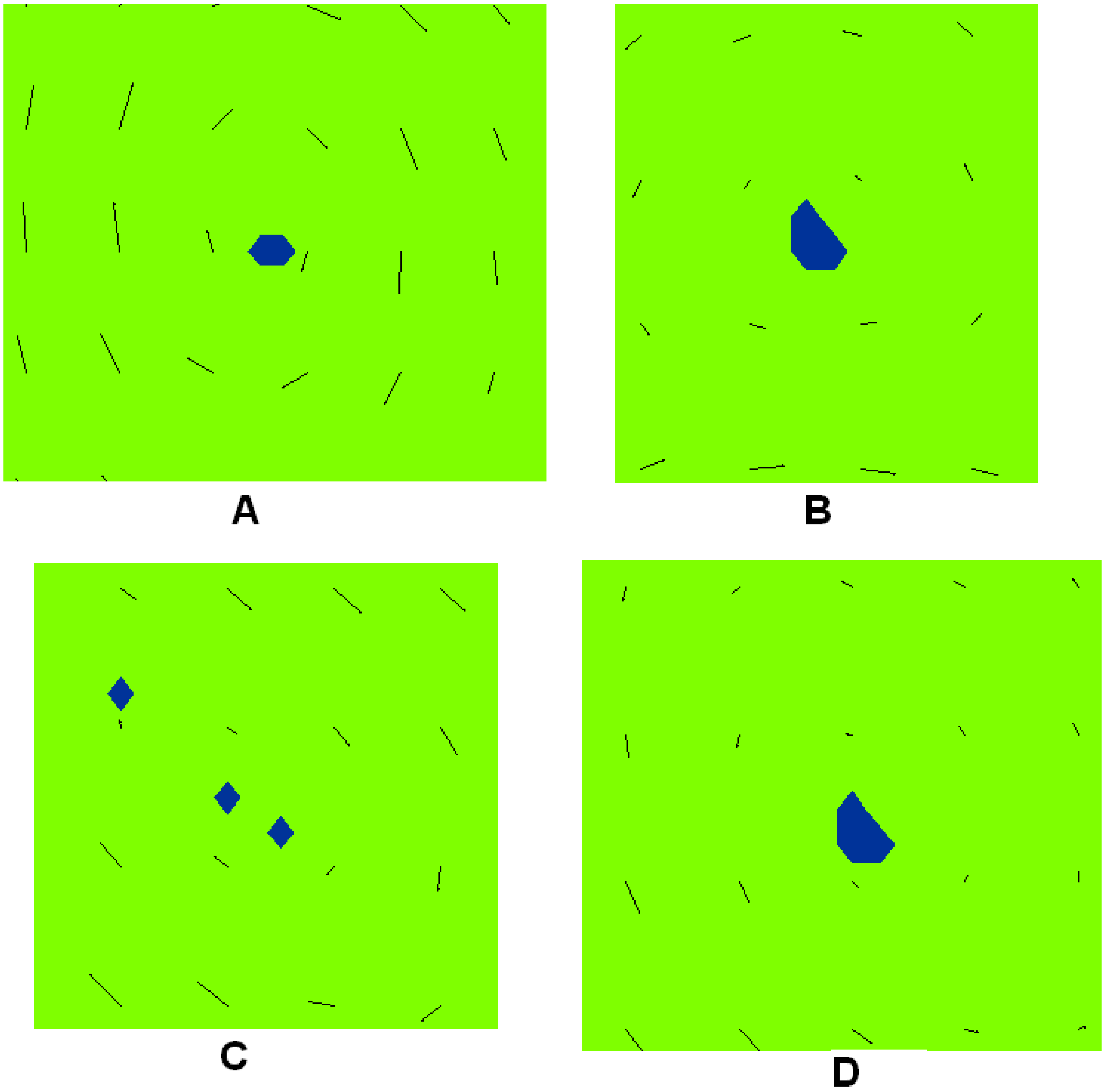


Figure 3-46: The closeups of last Figure.

3.4.7 Conclusions on the Vortex Core Detection Method

The intuitive logic behind this method is that swirling flows around a grid point can be detected by the direction-spanning property of the neighboring vectors around that point. The vortex core detection method can extract the vortex core regions individually. Even for two vortex structures very close to each other, the method can clearly detect two distinct vortex cores. Compared to the set-based method, the computation is less expensive. Moreover, since it does not require a threshold in the process of checking the direction-spanning property, human interaction or visual inspection is eliminated, and automated vortex core detection is achieved.

3.5 Detection of Streamlines and Closed Streamlines

3.5.1 Streamline Detection in Flows

Cartesian grids

The computation of a particle path is based on a numerical integration of the ordinary differential equation.

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = v(\mathbf{x}, t) \tag{3.47}$$

where t denotes time, $\mathbf{x}$ denotes the particle position and $v(\mathbf{x}, t)$ denotes the velocity field. The initial condition is provided by the starting point $\mathbf{x}_0$:

And the solution is a sequence of particle positions $(\mathbf{x}(t_0), \mathbf{x}(t_1), ...)$.

A particle tracing algorithm comprises of the following steps [51]. First, a search is conducted for the cell containing the initial position of the particle. Next the velocities in the cell corners are interpolated to determine the velocity in this point. Then, the next position of the particle can be calculated during the integration step, leading to another new search. The process of interpolation, integration and point location loops until the particle leaves the grid.

- **Point location**

  It refers to the process of determining which cell contains a specified point. The coordinates of a point can be divided into the integer and fractional parts [51]:

$$\mathbf{x} = (x, y, z) \Rightarrow [i, j, k] + (\alpha, \beta, \gamma) \tag{3.48}$$

  where, i, j, k are integers, which are indices, and $\alpha, \beta, \gamma \in [0, 1]$, which are offsets.

  In some cases, point location in a Cartesian grid is as simple as truncating the coordinates to their integer parts $[i, j, k]$. (α, β, γ) are offsets and also need to be determined [51].

- **Interpolation**

  In order to obtain a value of the velocity field in points other than the grid points, determining an interpolated value from the nodes surrounding the point becomes necessary. One way to do this is *Inverse Distance Weighting* [60] [59]. Let $(\mathbf{x}_0, \cdots, \mathbf{x}_7)$ be the coordinates of the eight corner nodes of a hexahedral cell, and let $(\mathbf{v}_0, \cdots, \mathbf{v}_7)$ be the velocities in those nodes. Then, the interpolated value $\mathbf{v}$ in point $\mathbf{X}$ is calculated as a weighted average of the corner values:

$$\mathbf{v} = \omega_0 \mathbf{v}_0 + \omega_1 \mathbf{v}_1 + \cdots + \omega_7 \mathbf{v}_7 \tag{3.49}$$

  The weight of each data point is calculated as a function of the Euclidian distance between $\mathbf{x}_i$ and $\mathbf{X}$:

$$d_i = \parallel \mathbf{x}_i - \mathbf{X} \parallel \tag{3.50}$$

$$\omega_i = \frac{\frac{1}{(d_i)^2}}{\sum_{j=0}^{7} \frac{1}{(d_j)^2}} \tag{3.51}$$

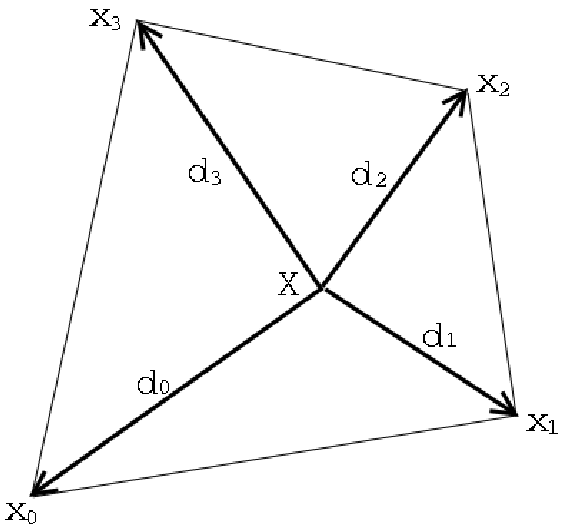


Figure 3-47: Inverse distance weighting

In a 2D cell consisting of four nodes, the same principle can be applied (see Figure 3-47 on page 111).

In the case that **X** is close to a corner node, the distance d_i would be close to zero and the value ω_i would be very large. So for reasons of floating point accuracy it is needed to first check whether the distance d_i is close to zero, in which case the weight of that corner ω_i is set to 1, and the weights of all other nodes are set to zero.

- **Integration**

  Many integration methods exist in the literature [16], from the simple first-order Euler scheme to the higher-order methods. Known as a second-order Runge-Kutta scheme, Heun's scheme is a widely used method. Starting from position $\mathbf{x}_n$ at time $t = t_n$, the position $\mathbf{x}_{n+1}$ at time $t = t_{n+1}$ is calculated in two steps:

$$\mathbf{x}^*_{n+1} = \mathbf{x}_n + \triangle t \cdot \mathbf{v}(\mathbf{x}_n) \tag{3.52}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \triangle t \cdot \frac{1}{2}\{\mathbf{v}(\mathbf{x}_n) + \mathbf{v}(\mathbf{x}^*_{n+1})\} \tag{3.53}$$

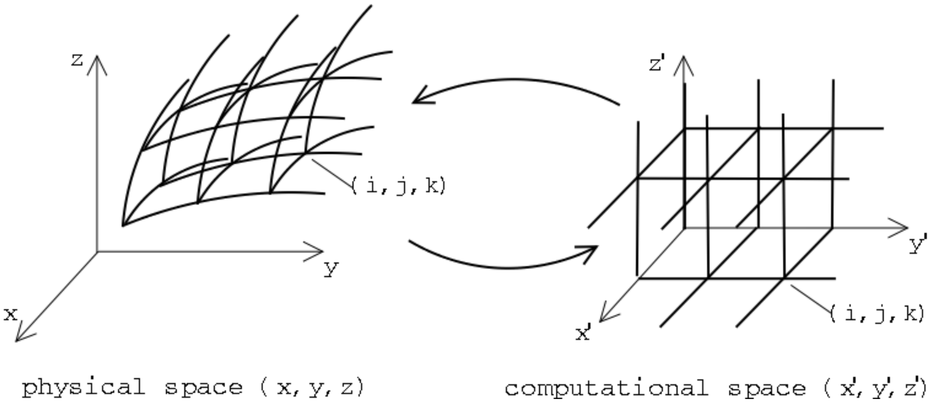

z
(i, j, k)
y
x
physical space (x, y, z)
z'
y'
(i, j, k)
x'
computational space (x', y', z')

Figure 3-48: Transformation between *P-space* and *C-space*, adapted from [51].

Curvilinear Grids

In real practice, to handle more complex geometries, many applications use grids other than Cartesian grids, such as curvilinear grids, also known as structured grids due to their regular topological structure, as opposed to unstructured grids which are used in Finite Element Methods.

Although curvilinear grids are applicable to a large variety of geometries, the trade-off is that numerical procedures are much more complicated in curved grids. That explains why the curvilinear grid in the physical domain is often internally transformed to a Cartesian grid in a new domain. The physical domain is called *P-space* and the new domain is called computational space or *C-space* [51]. A simple Cartesian grid in C-space enables much more efficient calculations.

Similar problems arise in particle tracing in a curvilinear grid, particularly with point location and interpolation. This suggests the use of a similar procedure in determining particle paths for visualization purposes as for flow solving. A transformation is done from physical space to computational space so that the curvilinear grid becomes a Cartesian grid (as in Figure 3-48 on page 112).

After point location and interpolation is carried out in Cartesian grid as previously discussed, the positions of the complete path back to the physical domain have to be transformed in order to be able to visualize it. Still, the use of a Cartesian computational domain may increase the efficiency of the algorithm.

An alternative way is to calculate the particle path directly in *P-space*. This makes point location and interpolation more complicated, but eliminates the work of grid transformation. However, in most case this method is less efficient.

3.5.2 Closed Streamline Detection

Essentially, a vortex consists of central vortex core region and surrounding closed streamlines. After we detected vortex core regions in an ocean domain, the next work is to detect the neighboring closed streamlines around vortex cores. We use two methods to detect closed streamlines. The first one is the winding angle method. In the winding angle method [39], closed streamlines can be recognized by determining if the winding angle of a streamline has reached a threshold, such as 2π for a full revolution, and the end point lies near the initial point. We also developed another method – the “cross”” method to detect closed streamlines. It is simple and efficient and has both a 2D and a 3D implementation. The Limit cycle method [61] to detect closed streamlines is also introduced.

Seed Points of Streamlines

In order to detect closed streamlines around every vortex core, the first step is to determine the seed points of these streamlines, without which the work of detecting streamlines could not be continued. As shown in Figure 3-49(a) on page 114, there are two vortex cores O_1 and O_2 which are already detected in the previous step of vortex core detection. To find the seed points around vortex core O_1, the distances from the vortex core O_1 to the four boundaries and that between two vortex cores are compared. The shortest distance O_1O_2 should be selected where seed points are located. In Figure 3-49(b) on page 114, the line O_1B is shorter than O_1O_2. So O_1B is selected as the place where seed points are located. The underpinning logic is that by selecting the shortest line, we can detect as many streamlines as possible within the domain. In contrast, if we select a line other than the one with the shortest distance as the location of seed points, it is more likely that some streamlines starting from

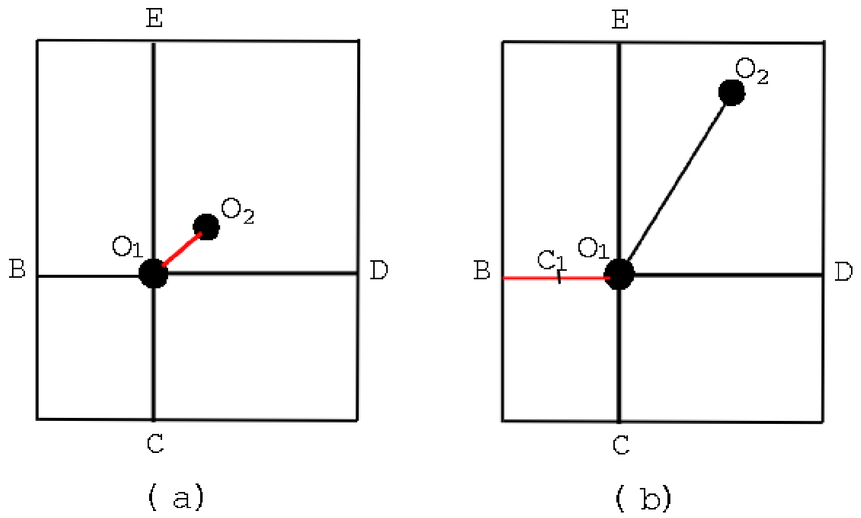


Figure 3-49: The shortest distance (red) is the line where seed points are located.

these seed points may exit the domain, and this definitely has to be avoided. Another advantage is that streamlines in between could still be detected in the case of two close vortex cores, separating the two vortices clearly. Thus, we can deal with the problem in point-based vortex detection methods when two close vortices get mixed up. For every vortex core candidate, a bisection algorithm is applied to the line where seed points are located to find suitable seed points for the closed streamlines, minimizing the number of streamlines needed to locate the bounds of a vortex region. The algorithm is as follows: in the shortest distance O_1B (in Figure 3-49(b)), the center point C_1 is selected as the first seed point. If a closed streamline is detected, then the center point C_2 of the line C_1B, which is far from the vortex core O_1, is selected as the second seed point; if no closed streamline is detected, then the center point C_2 of the line O_1C_1, which is close to the vortex core O_1, is selected as the second seed point. Repeat this process until when the distance $|C_nC_{n-1}| < \epsilon$, where ϵ is the termination criterion for the bisection method. At this time, if the streamline starting from the seed point C_n is a closed streamline, then it is also the boundary of the vortex with the vortex core O_1.

Winding Angle

As described in Section 3.3.2, the winding angle method is very reliable in detecting closed streamlines. Using the streamline detection method discussed in Section 3.3.2, a dense set of streamlines in an ocean domain are detected. Then the winding angle method is used to filter closed streamlines from the larger set of streamlines in the ocean domain. This process becomes much more efficient with the help of the vortex core detection method. As the result of the vortex core detection, all vortex cores in an ocean domain are detected. Then the next step is to detect closed streamlines around every vortex core. Thus, the number of streamlines that need to be checked by the winding angle method is much smaller. The combination of central core region and the swirling streamlines is the entire vortex area.

Limit Cycle

Another method to detect closed streamlines is a Limit Cycle method proposed by Wischgoll and Scheuermann [61]. According to this method, a streamline approaching a limit cycle has to reenter cells (Figure 3-50 on page 116). If a real exit can be reached, the streamline will leave the cell cycle (Figure 3-51 on page 116). If no real exit can be reached, the streamline will approach a limit cycle (Figure 3-52 on page 117). Their conclusion is that the closed streamline under detection is actually the limit cycle, where the initial point of a streamline has zero distance to its end point.

The Limit Cycle algorithm is comprised of two parts. The first part is the detection of closed streamlines as follows,

1. Streamline integration: From an initial point, calculate a streamline until a cell cycle is found. Here, cell circle is defined as cells crossed by the streamline have not changed for the last two turns.

2. Select potential exit: Potential points on each edge in cells of the cell cycle need to be found. The outflow out of the cell cycle may occur near these points. These points are identical with the vertices of the edge and points where the vector field is tangential to the edge.

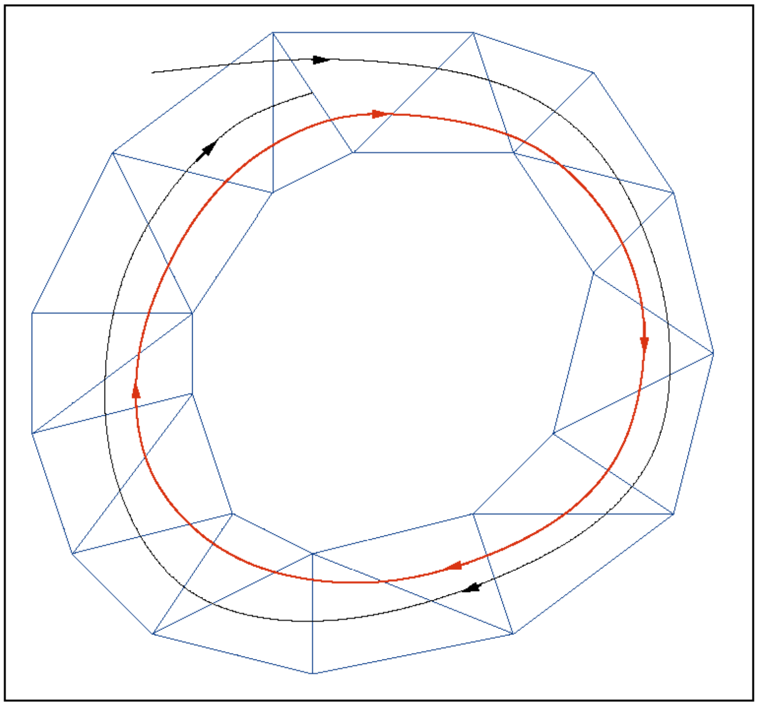


Figure 3-50: A streamline approaching a limit cycle has to reenter cells, taken from [61].

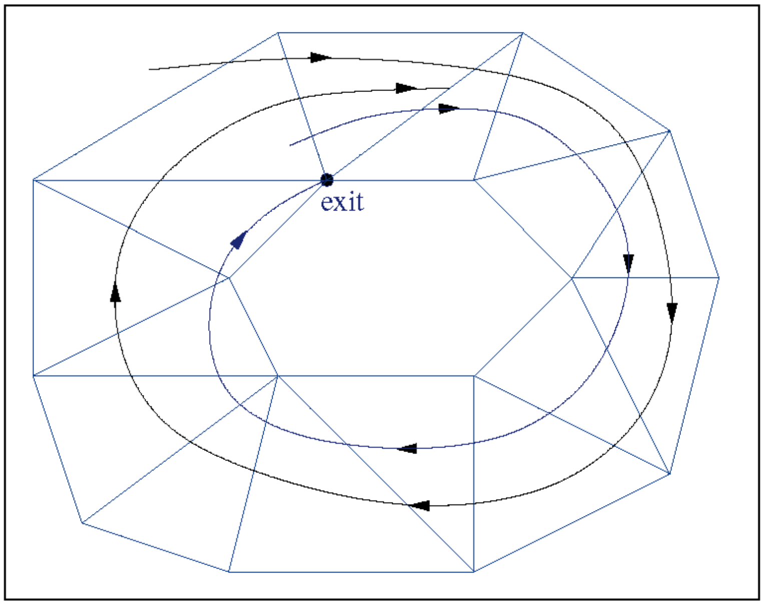


Figure 3-51: If a real exit can be reached, the streamline will leave the cell cycle, taken from [61].

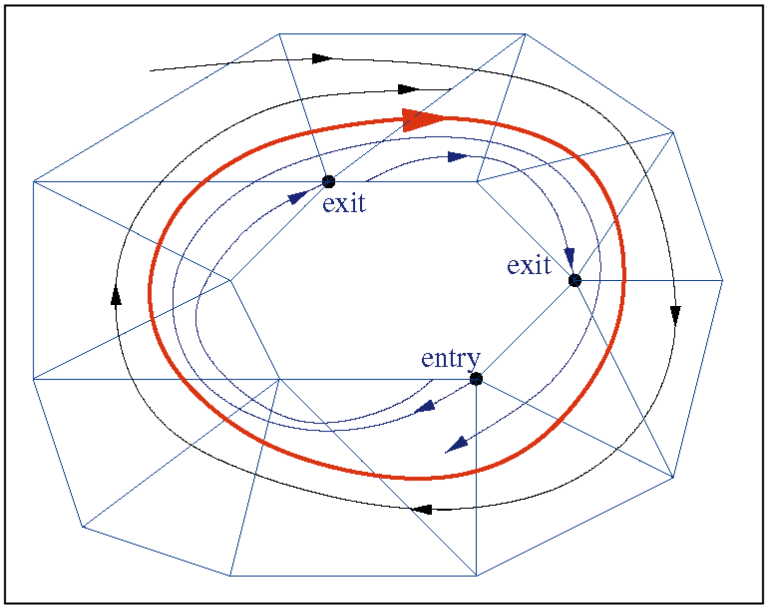


Figure 3-52: If no real exit can be reached, the streamline will approach a limit cycle, taken from [61].

3. Validating real exit: Integrate backward a curve from the potential exit. If this curve leaves the cell cycle, i.e, diverges from the investigated streamline, then this exit is a false exit. Otherwise, that is, this backward integration converges to the investigated streamline, the exit is a real exit.

4. If a real exit can be reached, then the streamline will leave the cell cycle. If no real exit, the streamline will approach a limit cycle.

The second part of the Limit Cycle method is the determination of the exact location of closed streamlines. Once a limit cycle is detected, work is continued by using one edge with which the investigated streamline intersects in the cell cycle. The edge is then divided into two parts at the midpoint of the edge, and we check which part gets intersected by the streamline. Then this binary search moves on until one point of the limit cycle is located. Then a streamline can start from this point, and get to the whole limit cycle (closed streamline).

This Limit Cycle algorithm is complicated. The whole process is: streamline integration ⇒ finding cell cycle ⇒ finding limit cycle ⇒ locate limit cycle. In fact, this method is more strict than the winding angle method in terms of detecting closed streamlines. Only streamlines which approach the limit cycle will be detected as closed streamlines. Also, it needs to check every streamline in a domain in order to find all closed streamlines. So it is computationally expensive. Moreover, cell cycle depends on the size of grids. The method requires relatively fine grids. In a nutshell, this method is good only for checking whether one of a few streamlines in one data field are closed streamlines or not.

3.6 A Novel Cross Method to Detect Closed Streamlines

We developed a new method to detect closed streamlines in a large-scale real ocean domain. We have named it the "cross" method since a closed streamline must cross some ranges and axis. Regardless of shape, size and direction of streamline, the above property is always observed. Another reason to name it the "cross" method is that in practice, four ranges and four semi-axis are most frequently used in this new method, and the two axes form a "cross". Examples prove that the method can get as good results as the winding angle method, while it involves much less computation work than the winding angle method.

3.6.1 Cross Property

In a vortex core, four axes can be drawn parallel to domain boundaries dividing the domain into four ranges (as in Figure 3-53 on page 119). A streamline starting from a point on one axis around this vortex core is studied. Some possible shapes of the streamline are as in Figure 3-54 on page 120: it could be located only in one range (A), or expand to two ranges (B), or transcend three ranges (C) or pass through all four ranges (D). None of the streamlines A, B and C could be recognized as closed streamlines, though some of them may bear some kinds of swirling property. Only situation (D) satisfies the "cross" property. We can simplify the condition of "cross"

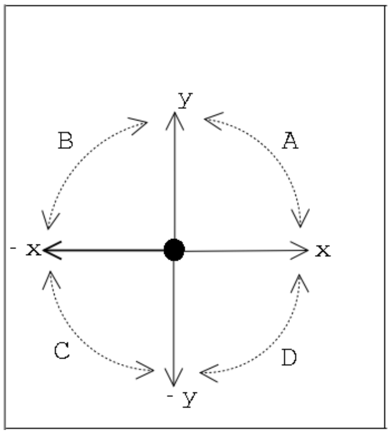


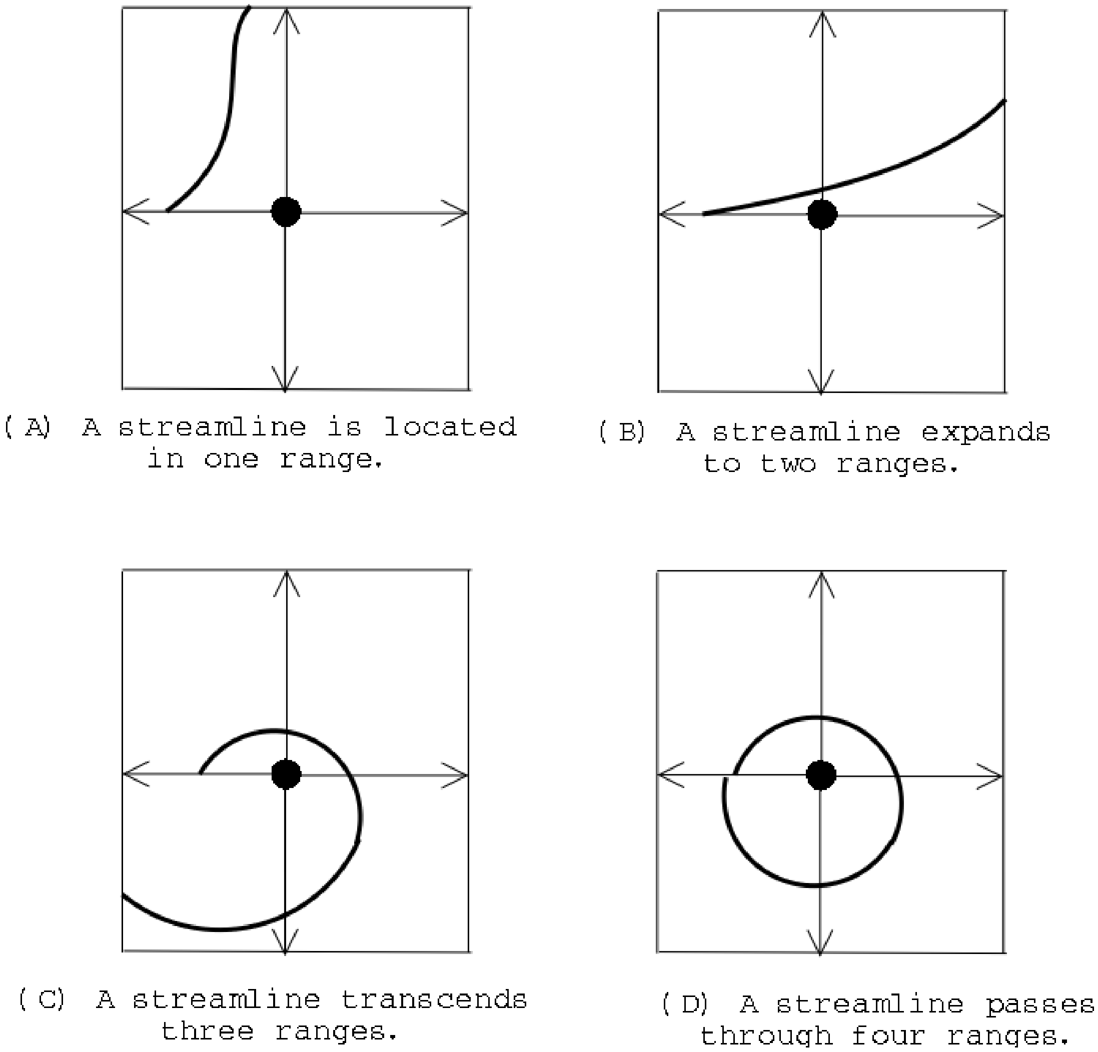
Figure 3-53: In a vortex core, four axes can be drawn which are parallel to domain boundaries.

property as the following: if a streamline starts from negative x axis and proceeds clockwise, then it has to pass through positive y axis, positive x axis, and then negative y axis again in order to satisfy the “cross” property.

Note that if there is only one vortex core O in an area shown as in Figure 3-55 on page 121, the ”closed” streamline E could not exist outside vortex core O. The reason is that according to the definition of vortex, vortex area consists of the central vortex core and the closed streamlines surrounding it. So ”closed” streamlines that ”bypass” the core are eliminated from consideration. In fact, the streamline E must enclose another vortex core in the same domain.

3.6.2 Distance Criterion

Satisfying the “cross” property does not guarantee a streamline to be closed. Another necessary condition here is that the end point of a streamline must lie near the initial point. In Figure 3-56 on page 121, streamline A is a closed streamline, but streamline B could not be recognized as a closed streamline. In practice, the position of end point do not need to be the same as that of initial point. So far as they are close to each other, we can recognize the streamline as the closed streamline. Therefore, a distance criterion is introduced here. An appropriate threshold for the distance between initial point of a streamline and its end point needs to be set. The distance



(A) A streamline is located in one range.

(B) A streamline expands to two ranges.

(C) A streamline transcends three ranges.

(D) A streamline passes through four ranges.

Figure 3-54: Four possible situations of a streamline.

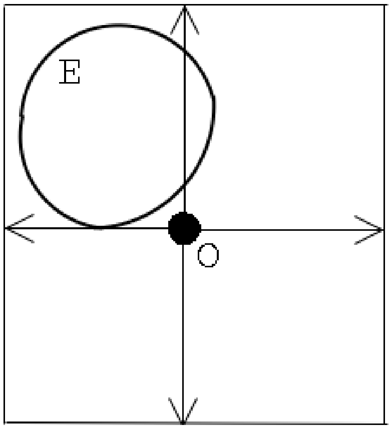


Figure 3-55: The streamline *E* could not "bypass" vortex core *O* if *O* is the only vortex core in the region.

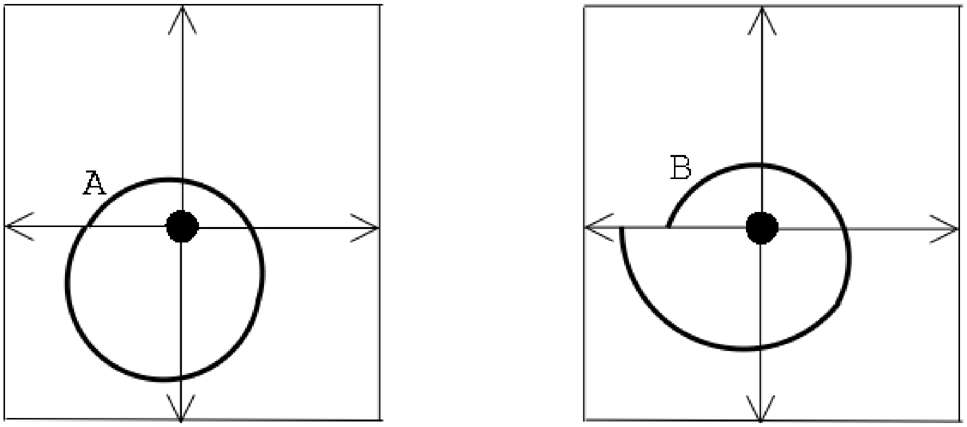


Figure 3-56: Streamline A is a closed streamline but streamline B is not.

criterion can be expressed as,

$$|\vec{P}_{end} - \vec{P}_{start}| < d_{threshold} \tag{3.54}$$

where, $\vec{P}_{end}$ and $\vec{P}_{start}$ refer to end point position and start point position.

3.6.3 2D Algorithm

The 2D algorithm is really simple. The "cross" property, combined with the distance criterion, determines if a streamline is a closed streamline around a vortex core. In 2D applications, we notice that if the "cross" property is satisfied for a streamline

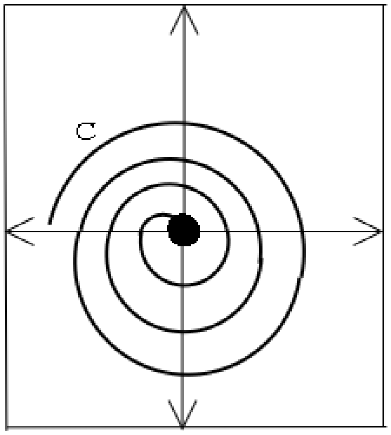


Figure 3-57: A superposition of a source or sink and a vortex is not a perfect vortex.

but the distance criterion are not satisfied, the streamline is still not closed, which means that the streamline is not in the vortex area. As in Figure 3-57 on page 122, streamline C is not in a 2D vortex area. In fluid flow, it could be recognized as a superposition of a source or a sink and a vortex. Although it has some kind of swirling property, it is not the same as vortex and thus not included in the content of vortex detection. However, in 3D application, streamline C may be in a vortex region. For example, Figure 3-58 on page 123 shows an upwelling in a 3D confined vortex. In top view, the vortex structure is the same as in Figure 3-57 on page 122, which is a 3D vortex. In fact, if vertical velocity exceeds a threshold in some regions, we can relax the distance criterion for streamline detection. The 3D case will be further studied in the 3D feature extraction and visualization for HOPS datasets.

3.6.4 3D Algorithm

Up to now, the “cross” method has only been implemented in 2D. But there is an alternative way to apply it in 3D. The idea is adapted from Portela’s [39] winding angle method for 3D: surrounding streamlines could be projected to a plane. This plane is referred to as the swirl plane since, as pointed out by [45] [39], instantaneous streamlines projected onto this plane exhibit a swirling pattern. Then the 2D algorithm can be applied to the projected streamlines on the swirl plane. Then the

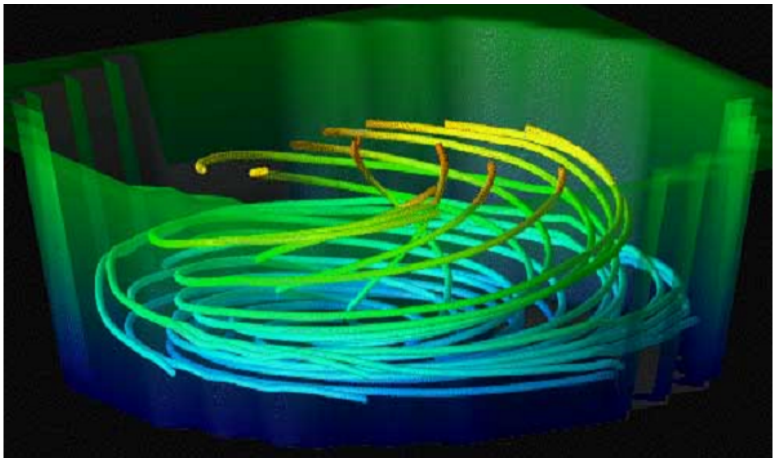


Figure 3-58: Upwelling in a confined vortex, taken from [3].

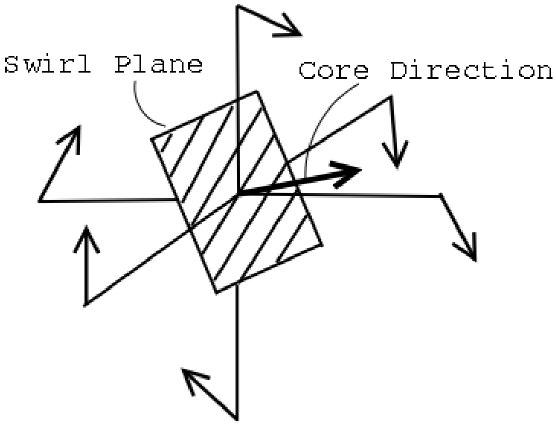


Figure 3-59: Swirling plane and core direction in 3D vortex core region, adapted from [25].

task is simplified into finding the normal vector of the swirling plane, which is called the vortex core direction since both of them should go though the vortex core (as in Figure 3-59 on page 123).

There are several existing methods to compute the core direction, all of which are related with the velocity gradient tensor. They range from the least computationally intensive, i.e., the vorticity vector [7] [54], to the most computationally intensive,i.e., the real eigenvector [55] [19]. It should be noted that [55] [11] have shown the latter to be the correct core direction, and others are only approximations to the core direction. Once the core direction is found, the swirling plane which is normal

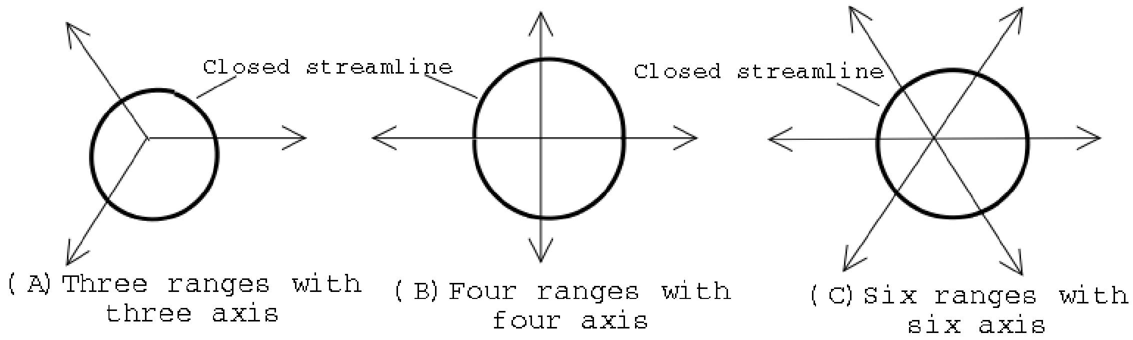

Closed streamline
Closed streamline
(A) Three ranges with three axis
(B) Four ranges with four axis
(C) Six ranges with six axis

Figure 3-60: Quantization of range and axis.

to the core direction is generated. Then the surrounding swirling streamlines could be projected to that plane. After that, the 2D algorithm can be applied. If the 2D algorithm detects that the "cross" property and the distance criterion are satisfied, then the detected streamline is a closed streamline.

To sum up, the 3D algorithm can be conducted in 3 steps: First, computing the core direction and swirl plane; second, projecting neighboring streamlines onto a swirl plane which is normal to the core direction; and third, applying the 2D algorithm to the projected streamlines.

3.6.5 Quantization of Range and Axis

Quantization of range and axis refers to the number of possible ranges and axes in which a streamline is checked. It needs to be predetermined in this method. In different applications, the regular grid may be different. It could be a cartesian grid, which has four ranges with four semi-axes for every grid point, or a complicated finite element grid, which has n ranges with n semi-axes (as in Figure 3-60 on page 124) for every grid point. When we check the "cross" property, the accuracy is actually the same for four ranges with four semi-axes or more ranges with more semi-axes. But an exponential increase will occur in computation when checking for the "cross" property of streamlines. Since we can always map data from the physical space to the computational space, four ranges with four axes are selected as a Cartesian grid is what is used in computational space for HOPS.

3.6.6 Classical Model Results

We still use the Rankine vortex model to check the validity of the winding angle method and the "cross" method. Streamlines of Rankine vortex model are shown in Figure 3-9 on page 67. Every streamline makes a full revolution and satisfies the distance criterion since the end point is in the same position of the initial point. Therefore, winding angle method detects all streamlines as closed streamlines. Also, the "cross" property is satisfied for every streamline in Rankine vortex. Therefore, our "cross" method detects all streamlines as closed streamlines. The whole Rankine vortex, which is the combination of Rankine vortex core O and all streamlines, is a vortex area. Thus, both the winding angle method and the "cross" method are valid in this example.

3.6.7 Real Ocean Model Results

When we use the winding angle or the "cross" method to detect closed streamlines around vortex cores in Massachusetts Bay, no closed streamline is detected. Although some streamlines around vortex cores show some swirling property, they don't satisfy the distance criterion. However, if vertical velocity (shown in Figure 3-11 on page 71) exceeds a threshold in some regions, we can relax the distance criterion for streamline detection. Since vertical velocities increase along the vertical direction of vortex cores, it's quite possible that both the sink and the vortex exist in the vortex area, which could form 3D vortices. This issue will be further studied in 3D results of feature extraction and visualization for HOPS datasets.

The result of the winding angle method applied to a flow field of the Western Mediterranean produced by HOPS is shown in Figure 3-61 on page 126. The closed streamlines around the vortex cores are detected. The combination of vortex core regions and the surrounding closed streamline regions are entire vortex areas. Figure 3-62 on page 127 shows the result of the "cross" method in the Western Mediterranean. The "cross" method gets very similar result with the winding angle method. Both methods get fairly good results. Figure 3-63 on page 128 shows the attribute

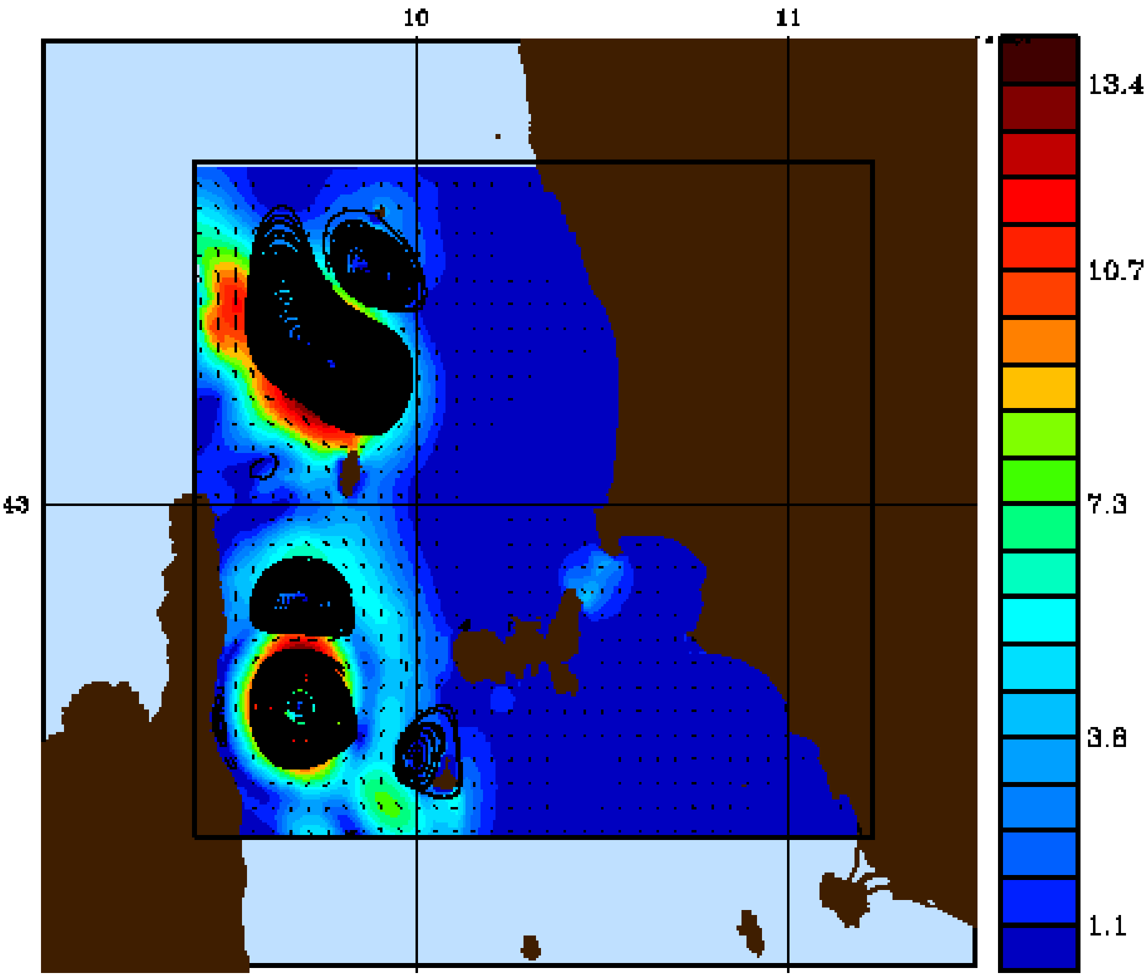


Figure 3-61: The result of winding angle method in the Western Mediterranean.

set of vortex direction in the Western Mediterranean, where dark blue parts denote clockwise rotating vortices, and red parts denote counter-clockwise rotating vortices.

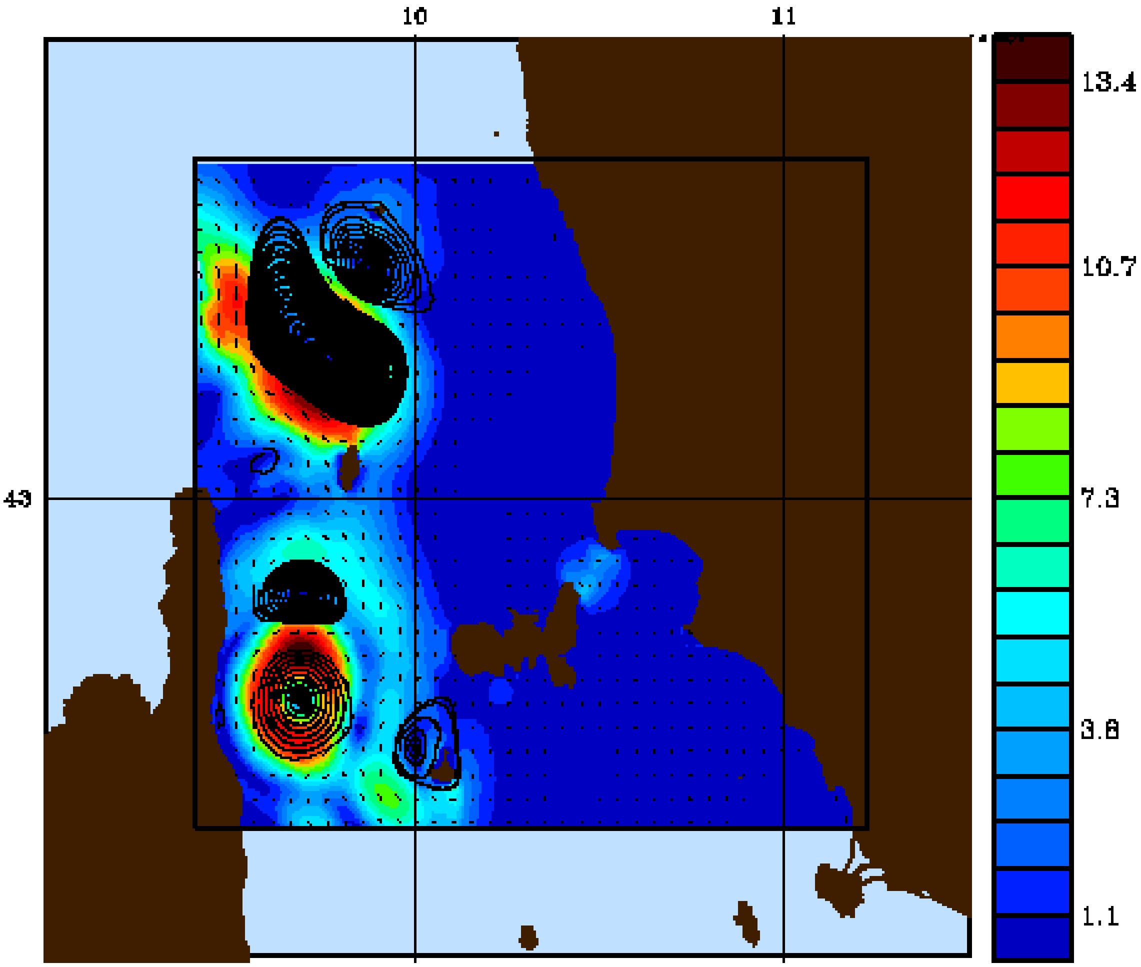


Figure 3-62: The result of the “cross” method in the Western Mediterranean.

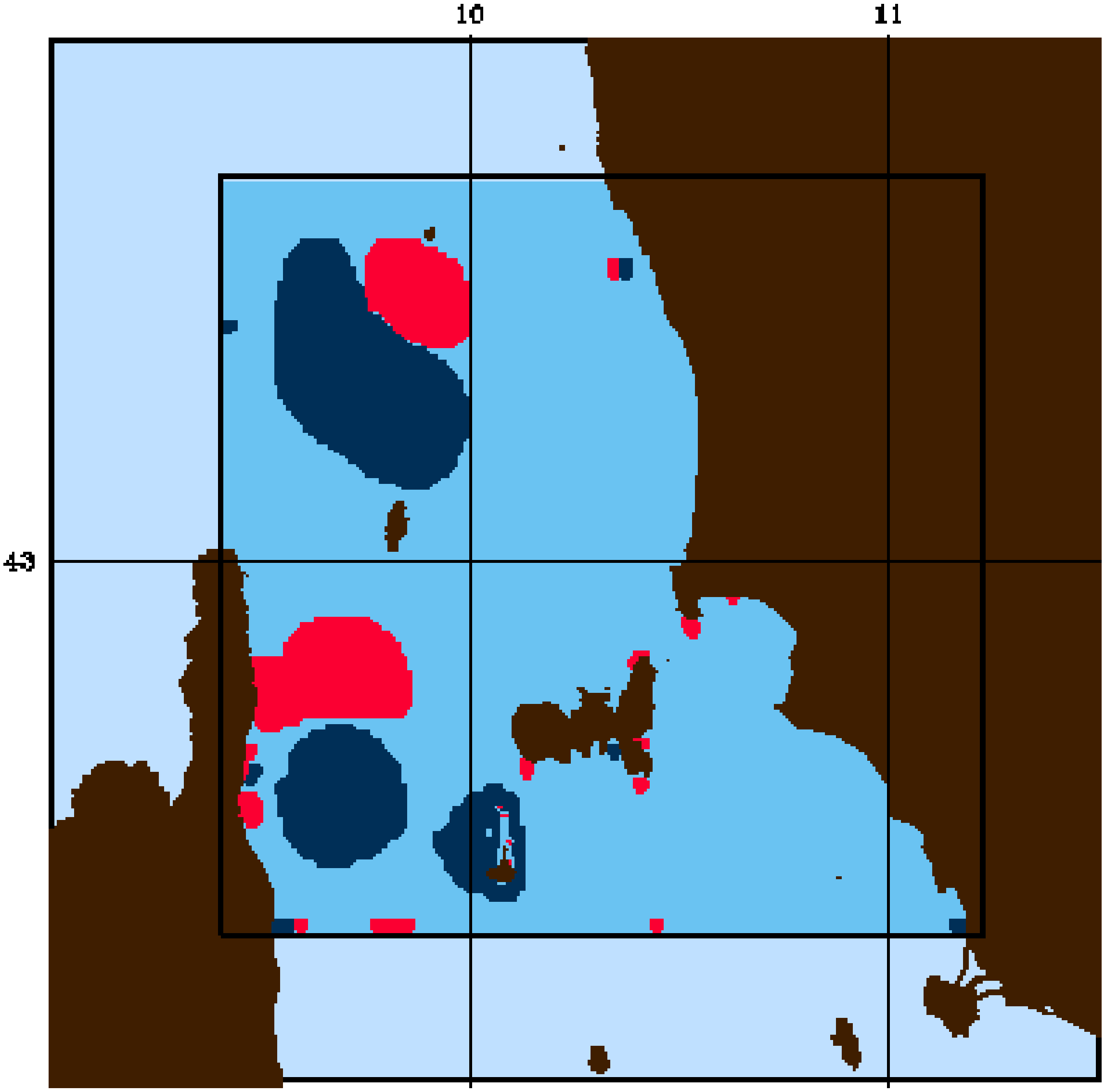


Figure 3-63: The result of the “cross” method in the Western Mediterranean.

3.6.8 Complexity Analysis

The cost of point-based methods is independent of the number of vortex cores and scales with the number of grid points as $O(N)$. The cost of the "cross" method and the winding angle method is independent of the number of grid points and scales with the number of vortex core candicates (N_C) as $O(N_C)$. For every vortex core candidate, a bisection algorithm is used to find suitable seed points for the closed streamlines, minimizing the number of streamlines needed to find the bounds of a vortex region. So the total time for detecting closed streamlines T_{tot} is

$$T_{tot} = O(N) + \sum_{j=1}^{N_C} C_{Bj} \tag{3.55}$$

where, $O(N)$ denotes the cost for transform between physical space and computational space. C_{Bj} denotes the time of the bisection algorithm for vortex core candidate j. The cost per vortex core candidate C_{Bj} should scale as $O(\log_2 \Delta_j)$ where Δ_j is the minimum distance between a given candidate j and the nearest boundary or other vortex core. Therefore, C_{Bj} can be expressed as

$$C_{Bj} = \sum_{i=1}^{\lceil log_2 \frac{\Delta_j}{\epsilon} \rceil} C_{Si}^j \tag{3.56}$$

where, $\epsilon > 0$ is the termination criterion for the bisection method. C_{Si}^j denotes the time for extracting one streamline. In streamline extraction, the Inverse Distance Weighting algorithm using the four corners of a cell is used as the interpolation method and a second-order Runge-Kutta scheme (Heun's scheme) is used as the integration method. So C_{Si}^j can be expressed as,

$$C_{Si}^j = (C_I + C_R + C_C) \cdot \frac{\tau_i^j}{\delta t} \tag{3.57}$$

where, C_I denotes the time per step of the Inverse Distance Weighting algorithm, C_R denotes the time per step of Heun's scheme and C_C denotes the time for checking for the "cross" or winding angle property per step. They all have approximately constant cost per step. τ_i^j denotes the time needed for a streamline i to complete a loop, effectively stagnate or exit the domain. τ_i^j is different from streamline to streamline for the same vortex core candidate as well as for different vortex core candidates. δt denotes the step size.

Combining the above three formulae, we have:

$$T_{tot} = O(N) + \sum_{j=1}^{N_C} \sum_{i=1}^{\lceil log_2 \frac{\Delta_j}{\epsilon} \rceil} (C_I + C_R + C_C) \cdot \frac{\tau_i^j}{\delta t} \tag{3.58}$$

Factoring out the constant terms,

$$T_{tot} = O(N) + \frac{(C_I + C_R + C_C)}{\delta t} \cdot \sum_{j=1}^{N_C} \sum_{i=1}^{\lceil log_2 \frac{\Delta_j}{\epsilon} \rceil} \tau_i^j \tag{3.59}$$

For a characteristic Δ_j, say Δ^{ch} and a characteristic τ_i^j, say τ^{ch}, in the "cross" and the winding angle methods, the total time for detecting closed streamlines T_{tot} is expected to scale as $O(N + \frac{N_C \cdot log_2 \Delta^{ch} \cdot \tau^{ch}}{\delta t})$. Moreover, the "cross" method uses the simple "cross" property to detect closed streamlines, while the winding angle method uses the $atan()$ function in order to calculate every α_i, which is computationally more expensive. Therefore, the winding angle method is expected to take more time than the "cross" method as $C_{C\,cross} < C_{C\,winding}$, the difference becoming considerable for small δt and large N_C.

Timings

The timing tests were performed on a PC which has two Athlon MP@1.4GHz processors, 512MB memory, running under the Red Hat 7.2 Linux. However, software used only one processor. We did three tests for each method applied in Massachusetts Bay and in the Western Mediterranean respectively. The mean values of the tests are listed in Table 3.1 on page 131, where user time I is for Massachusetts Bay and user time II is for the Western Mediterranean. Note that the "cross" method and the winding angle method need to detect streamlines, which involves interpolation and integration. We were able to get satisfactory results with an integration step size of $\delta t = 0.1$. It is reasonable that these two methods take more time than other methods.

Table 3.1: Timings of vortex detection methods.

	user time I (seconds)	user time II (seconds)
vortex core detection	0.00195	0.00390
cross method	1.13281	2.08984
winding angle	1.21132	2.32812
positive Q	0.00195	0.00390
negative λ_2	0.00390	0.00585
conjugate eigenvalues	0.00195	0.00390
vorticity magnitude	0.00195	0.00195

However, both methods produce much more accurate results than other methods in our ocean models. The "cross" method uses less time than the winding angle method in order to get results of similar accuracy.

3.6.9 Conclusion on the "Cross" Method

We have presented a "cross" method to detect closed streamlines in the vortex area. The algorithm is both simple and efficient. We can conclude with some certainty whether or not a streamline around a vortex core is a closed streamline and belongs to the vortex area by examining its "cross" property and distance criterion. We have tested our algorithms on real large-scale ocean models. In every case, the vortex regions including the vortex core and the surrounding closed streamlines detected by our algorithms either concurred with similar results from other methods, such as the winding angle method, or were validated through human visual inspection. While the "cross" method can get as good results as the winding angle method, it requires less computational work which can also be used for the second part.

Our vortex detection method, unlike many other methods, divides the work into two steps. First, detecting vortex core regions; and second, detecting closed streamlines around vortex core regions. According to Portela's [39] work, a vortex is made up of a central core region surrounded by swirling streamlines. So, both parts, the core and the streamlines, are integral parts of a vortex, their combination making an entire vortex area. This strategy has many merits. The first part, detecting vortex cores, is not computationally expensive as explained in the algorithm; for the second part, detecting surrounding closed streamlines, instead of checking all the streamlines in an ocean domain, only those streamlines in the neighborhoods of vortex cores need to be checked. Moreover, results can not only show vortex areas, but also clearly show the vortex core regions, which could be important in some oceanographic applications. Finally, the method has both 2D and 3D implementations.

Chapter 4

Detection of Upwelling

Upwelling is a process by which cold water is brought up from the depths of the ocean. In the USA, along the California and Oregon states, wind from the northwest pulls warmer surface water away from shore, and cold water moves up to take its place [5]. The reason the wind does not just push that water on down south has to do with physics, friction in particular, and the Coriolis effect, which combine into a process known as Ekman transport. Cold water appears off the shore from about April to September. Fog results when cold air comes in contact with the moist warm air flowing off the land. But fog is only one aspect of the upwelling phenomenon. That cold water also brings nutrients, nourishing the very bottom of the food web: phytoplankton and zooplankton. And plankton, in turn, supports larger marine creatures. When the winds come and push the warm water away from shore, the cold water rising from the depths brings those nutrients to the surface. The plant life thrives - plankton and all the other seaweeds. As they convert light and nutrients into food and oxygen, they manufacture most of the organic compounds required by marine animals: zooplankton first, which are eaten by fish, which are eaten by larger fish and birds and marine mammals. Last but not least, the oxygen given off by the marine plants benefits the land animals as well.

The reason why upwelling occurs has to do with the Ekman transport [13] [5]. The flow of most surface currents in the ocean is driven by wind. When wind blows over water, the surface of that water is not pushed directly in front of the wind, but moves

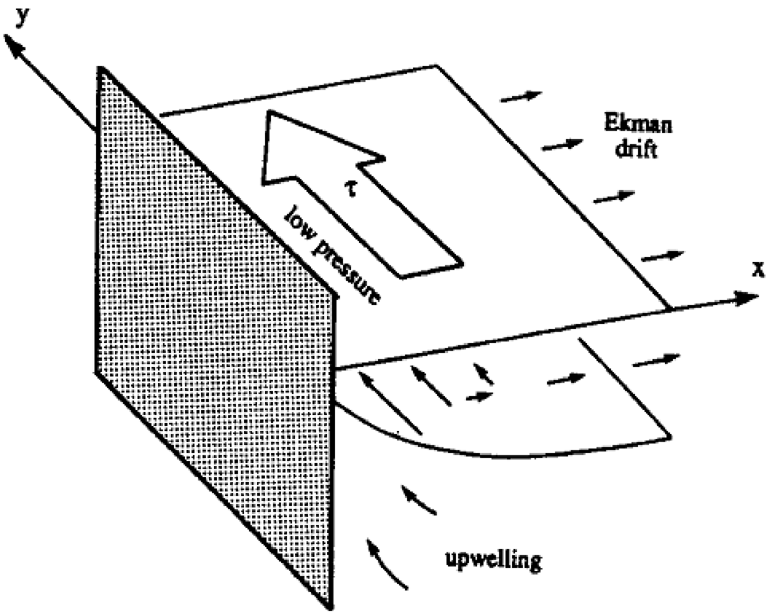


Figure 4-1: Coastal upwelling

at about 45 degrees to the right of the wind's motion in the Northern Hemisphere, or to the left in the Southern Hemisphere, thanks to the Coriolis force, an effect of the rotation of the earth. As one descends in the water, the direction of flow continues to be deflected rightward (or leftward), until ultimately a three-dimensional spiral is formed vertically in the water. The net transport of water, as explained by the Ekman transport, is at an angle of roughly 90 degrees to the direction of the wind. In short, water, to a depth of a few hundred meters, is pulled directly off the coast - leaving room for deep water to rise and replace the displaced water. Upwelling is generally associated with strong cold currents. The diagrams of two typical upwelling, coastal upwelling and equator upwelling, are shown in Figure 4-1 on page 134 and Figure 4-2 on page 135.

4.1 Algorithm on Detecting Upwelling

Judging from the definition, upwelling is a process by which cold water is brought up from the depths of the ocean, which prompts the intuitive thinking to detect upwelling. We can integrate vertical velocity from bottom to top at every grid point

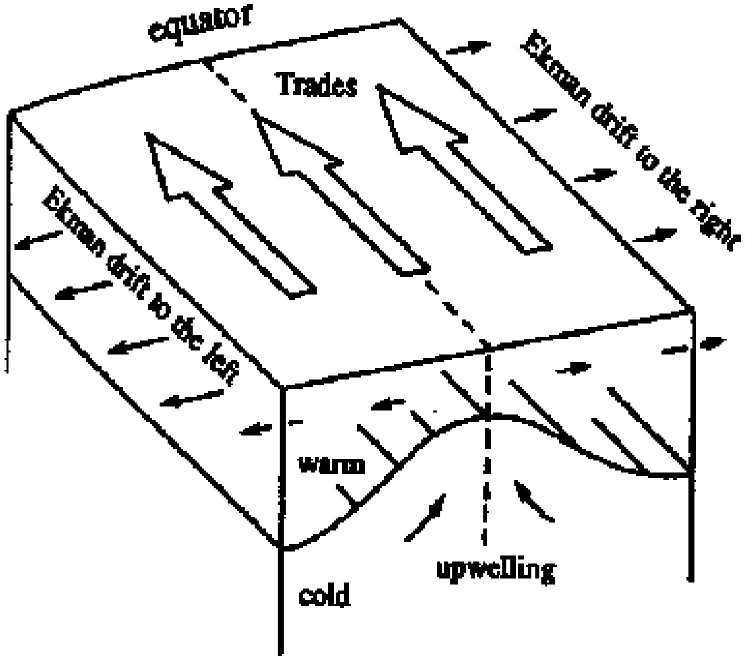


Figure 4-2: Equator upwelling

along a vertical line so we will get upwelling magnitude at that grid point on horizontal plane. In HOPS system, instead of tracer points, we use velocity points as grid points because vertical velocity is located at every velocity points.

We have used the trapezoidal rule and Simpson's rule to do vertical velocity integration. Simpson's rule has higher precision since the trapezoidal rule gives the exact value of the integral for linear polynomials, whereas the Simpson's rule gives the exact value of the integral for polynomials of degree three or less. After integration, we can get the upwelling magnitude at every grid point on the horizontal plane. Then a threshold is needed to determine if every grid point is in upwelling area or not. If the following condition is satisfied, that grid point is in upwelling area.

$$V_{point} \geq V_{threshold} \tag{4.1}$$

where, V_{point} is upwelling magnitude at a grid point and $V_{threshold}$ is a threshold.

Finally, the grid points that are in the upwelling area need to be classified together and visualized as in the case of the vortex area. We can summarize the algorithm of detecting upwelling into three steps: First, integrating vertical velocity from bottom to top at every grid point along a vertical line; second, comparing upwelling magnitude at every grid point on a horizontal plane with the upwelling threshold; and third, clustering grid point in the upwelling area and visualizing them.

4.2 Real Ocean Model Results

Figure 4-3 on page 137 shows the the result of Nowcast (Jun, 17, 1997) in Massachusettes Bay. The green part represents upwelling area. The threshold here is 5 m/s. The integrated vertical velocity in green part is bigger than the value of threshold, which indicates that there is greater possibility of upwelling in that region than in other region. The different threshold values could be defined in order to get different orders of intensity of upwelling. Figure 4-4 on page 138 shows the result of three days Forecast (Jun, 20, 1997) in Massachusettes Bay. The green part represents upwelling area.

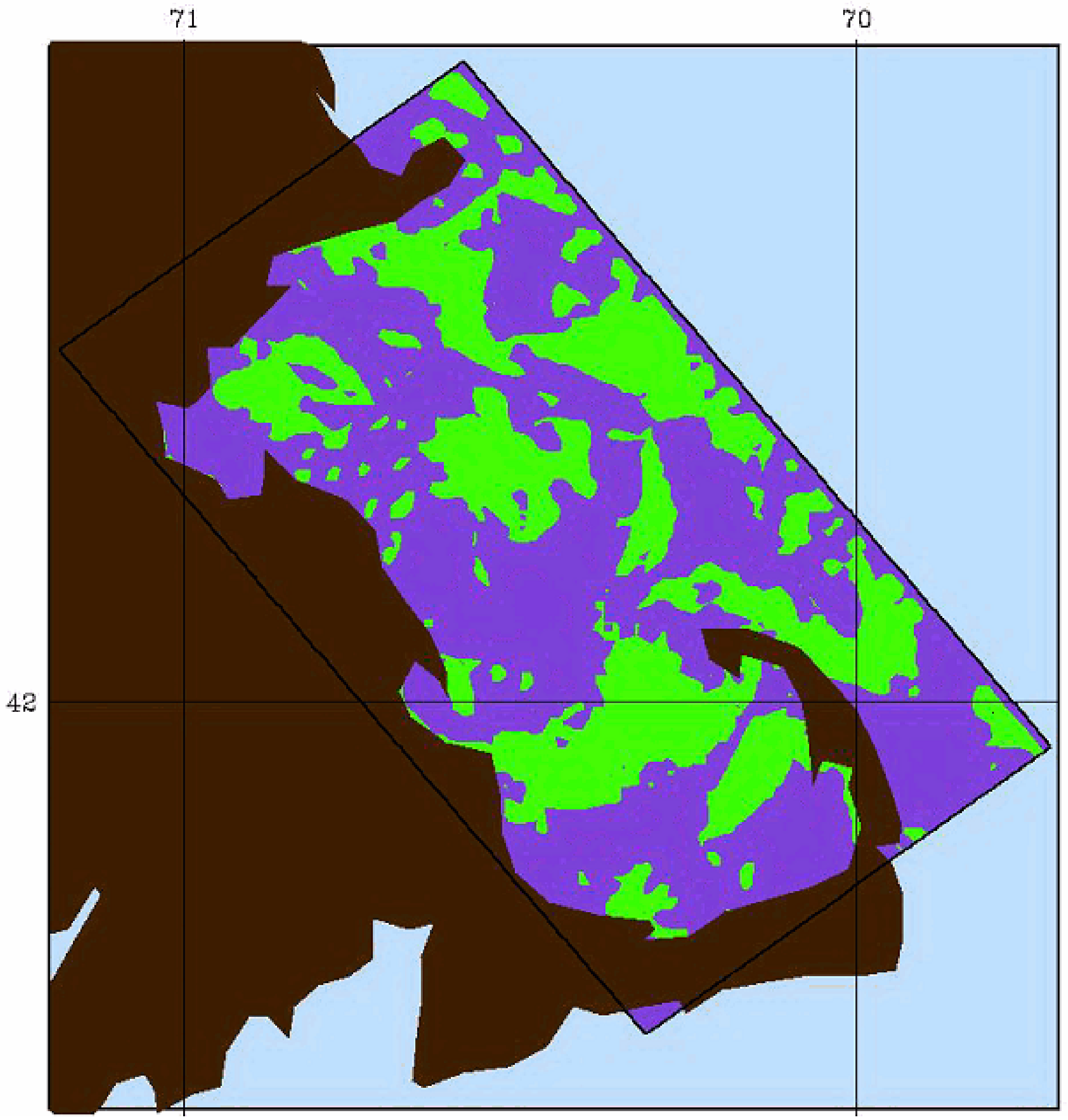


Figure 4-3: Upwelling in Massachusetts Bay.

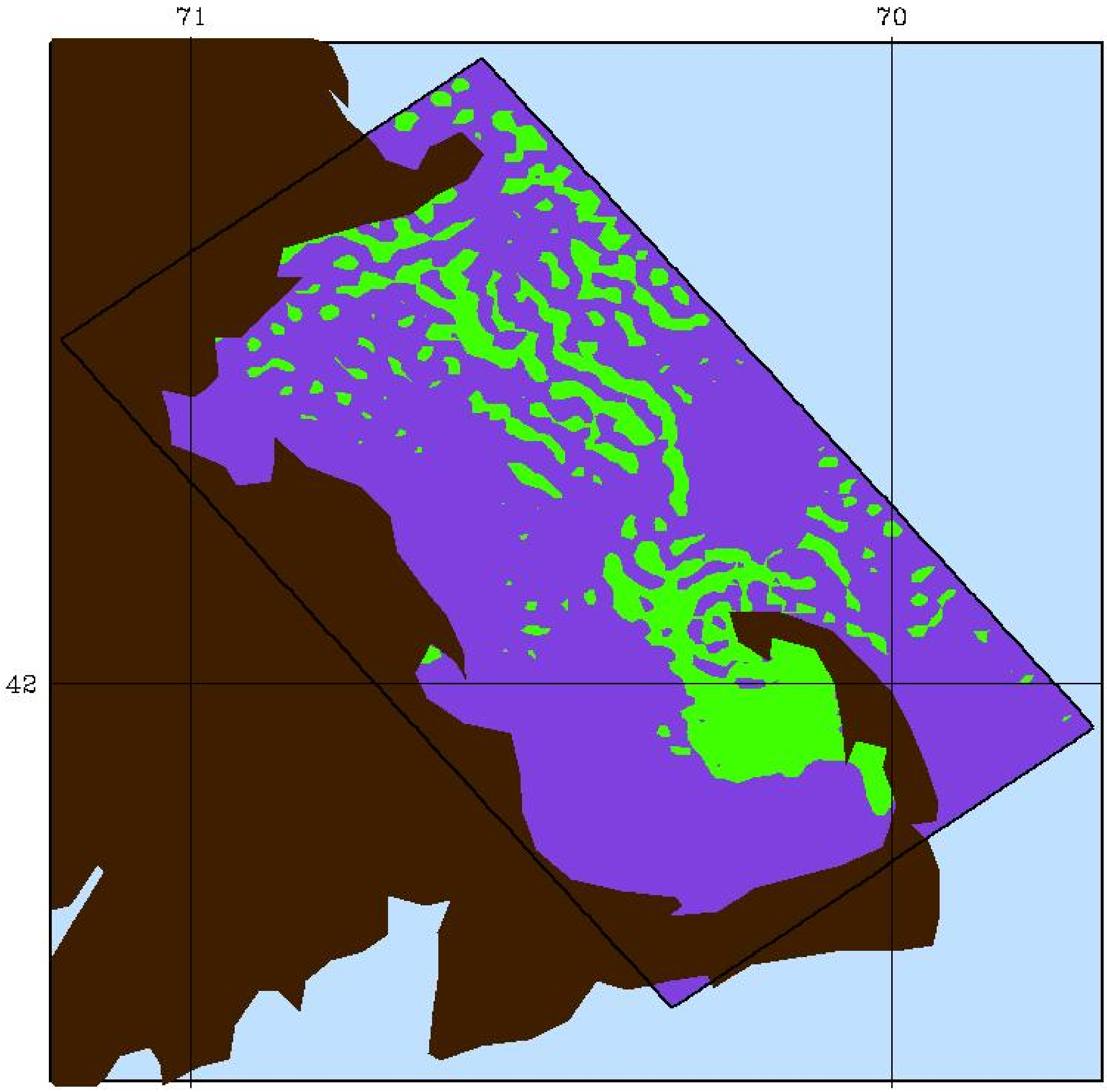


Figure 4-4: Upwelling in Massachusetts Bay.

Chapter 5

Conclusions

5.1 Conclusions

In this thesis, feature extraction and scientific visualization methods are discussed. The approaches for detection of two typical oceanographic features, vortex, upwelling, are presented. In vortex detection, the algorithm consists of two parts. The first part is vortex core detection. A geometry-based vortex core detection method is used to detect vortex cores. The second part is closed streamline detection. A novel cross method is used to detect closed streamlines around vortex cores. Finally, the combinations of vortex core regions and their surrounding closed streamline regions form the entire vortex regions. In upwelling detection, the algorithm of integrating vertical velocity from the bottom to the top in the ocean is used. These detection algorithms have been applied to real large-scale ocean models. In every case, vortices or upwelling detected by our algorithms either concurred with similar results from other methods, such as the winding angle method, or were validated through human visual inspection.

As a result, feature-based visualization could be served for adaptive sampling in the Poseidon project. It will facilitate manual adaptive sampling and eventually could be used to help automate adaptive sampling.

5.2 Future Work

Future work can be focused on the following areas,

1. The 3D feature extraction and visualization for HOPS datasets. The related work can be seen in [20].

2. The introduction of iconic representation for combined visualizations of features and other quantities.

3. Visualizing errors and uncertainty in feature extraction and visualization. The related work can be seen in [26].

We would like to introduce the above in the context of a more modern visualization framework (IBM OpenDX [6] [10]), which has a better user interface and more powerful visualization functions. Using OpenDX Java Explorer (applet-based tool), the visualization programs can be accessed through the web.

Bibliography

[1] Harvard ocean prediction system (HOPS). http://oceans.deas.harvard.edu/HOPS/.

[2] Model vortexes. http://www.kilty.com/t_model.htm.

[3] NCAR graphics. http://ngwww.ucar.edu/ng4.3/.

[4] Netcdf. http://www.unidata.ucar.edu/packages/netcdf/.

[5] Ocean upwelling. http://www.ocean98.org/cacoast2.htm.

[6] OpenDX. http://www.opendx.org/.

[7] D. C. Banks and B. A. Singer. A predictor-corrector technique for visualizing unsteady flow. *Visualization and Computer Graphics*, 1(2):151–163, 1995.

[8] K. Bryan. A numerical method for the study of the circulation of the world ocean. *J. of Comp. Phys.*, (4):347–376, 1969.

[9] J. M. Burgers. A mathematical model illustrating the theory of turbulence. *Adv. Appl. Mech.*, 1:197–199, 1948.

[10] Robert C. Chang. The encapsulation of legacy binaries using an XML-based approach with applications in ocean forecasting. Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2003.

[11] M. S. Chong, A. E. Perry, and B. J. Cantwell. A general classification of three-dimensional flow field. *Physics of Fluids*, 2:765–781, 1990.

[12] M. D. Cox. A primitive equation, three-dimensional model of the ocean. Technical Report 1, GFDL Ocean Group, Princeton NJ, 1984.

[13] B. Cushman-Roisin. *Introduction to Geophysical Fluid Dynamics*. Prentice Hall, 1994.

[14] K. W. Brodlie et al. *Scientific Visualization – Techniques and applications*. Springer–Verlag, 1992.

[15] G. Farin. *Curves and surfaces for computer aided geometric design*. Academic Press, New York, 1990.

[16] L. Fausett. *Numerical methods: algorithms and applications*. Prentice Hall, New Jersey, 2003.

[17] Chengwei Feng. A user interface design for feature extraction and tracking with application to oceanography. Master's thesis, Rutgers, the State University of New Jersey, Department of Electrical and Computer Engineering, 2000.

[18] A. Globus, C. Levit, and T. Lasinski. A tool for visualizing the topology of 3D vector fields. In *Proc. of IEEE Visualization '91*, pages 33–39, San Diego, CA, 1991.

[19] R. Haimes and D. Kenwright. On the velocity gradient tensor and fluid feature extraction. In *AIAA Paper 99–3288*, 1999.

[20] E. Heiberg, T. Ebbers, L. Wigstrom, and M. Karlsson. Three-dmensional flow characterization using vector pattern matching. *Visualization and Computer Graphics*, 9(3):313–319, 2003.

[21] J. C. R. Hunt, A. A. Wray, and P. Moin. Eddies, streams, and convergence zones in turbulent flow fields. Technical Report CTR-S88, Center for Turbulence Research, 1988.

[22] F. Hussain and M. Hayakawa. Education of large-scale organized structure in a turbulent plane wake. *Journal of Fluid Mechanics*, 180:193–204, 1987.

[23] J. Jeong and F. Hussain. On the identification of a vortex. *Journal of Fluid Mechanics*, 285:69–94, 1995.

[24] M. Jiang, R. Machiraju, and D. Thompson. Geometric verification of swirling features in flow fields. In *Proceedings of the Conference on Visualization '02*, pages 307–314, Boston, MA, 2002.

[25] M. Jiang, R. Machiraju, and D. Thompson. A novel approach to vortex core region detection. In *Joint Eurographics – IEEE TCVG Symposium on Visualization*, pages 217–225, Barcelona, Spain, May 2002.

[26] C. R. Johnson and A. R. Sanderson. Visualization viewpoints. *Computer Graphics and Applications*, 23(5):6–10, 2003.

[27] W. C. Leeuw and F. H. Post. *A statistical view on vector fields.* Visualization in Scientific Computing. Springer-Verlag, 1995.

[28] P. F. J. Lermusiaux. Data assimilation via error subspace statistical estimation, part II: Mid-atlantic bight shelfbreak front simulations, and ESSE validation. *Monthly Weather Review*, 127(8):1408–1432, 1999.

[29] P. F. J. Lermusiaux. Evolving the subspace of the three-dimensional multiscale ocean variability: Massachusetts Bay. *Journal of Marine Systems, Special issue on "Three-dimensional ocean circulation: Lagrangian measurements and diagnostic analyses"*, 29:385–422, 2001.

[30] W. Lorensen and H. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *ACM Siggraph Proceedings*, number 21, pages 163–166, 1987.

[31] C. J. Lozano, P. Haley, H. Arango, and Q. Sloan. Harvard coastal/deep water primitive equation model. Harvard Open Ocean Model Reports 52, Harvard University, Department of Earth anf Planetary Sciences, Harvard University, MA, July 1994.

[32] C. J. Lozano, P. Haley, H. Arango, Q. Sloan, and A. R. Robinson. Harvard coastal/deep water primitive equation model. Technical Report 52, Harvard University, Cambridge MA, 1994.

[33] H. Lugt. *Vortex Flow in Nature and Technology.* Wiley, 1972.

[34] B. H. McCormick, T. A. DeFanti, and D. Brown. Visualization in scientific computing. *Computer Graphics*, 21(6):1–4, November 1987.

[35] R. N. Miller, A. R. Robinson, and D. B. Haidvogel. A baroclinic quasi-geostrophic open ocean model. *J. of Comp. Phys*, (50):38–70, 1983.

[36] N. M. Patrikalakis. Rapid real-time interdisciplinary ocean forecasting: Adaptive sampling and adaptive modeling in a distributed environment. http://czms.mit.edu/poseidon/.

[37] J. Pedlosky. *Geophysical Fluid Dynamics.* Springer–Verlag, New York, second edition, 1987.

[38] A. E. Perry and M. S. Chong. A description of eddying motions and flow patterns using critical-point concepts. *Annual Review of Fluid Mechanics*, 19:125–155, 1987.

[39] L. M. Portela. *Identification and Characterization of Vortices in the Turbulent Boundary Layer.* PhD thesis, Stanford University, School of Mechanical Engineering, 1997.

[40] F. H. Post, B. Vrolijk, H. Hauser, R. Laramee, and H. Doleisch. Feature extraction and visualization of flow fields. In *EUROGRAPHICS 2002 STAR – State of the Art Report.* Eurographics, 2002.

[41] A. R. Robinson. Forecasting and simulating coastal ocean processes and variabilities with the Harvard Ocean Prediction System. In *Coastal Ocean Prediction*, AGU Coastal and Estuarine Studies Series, pages 77–100. American Geophysical Union, 1999.

[42] A. R. Robinson and S. M. Glenn. Adaptive sampling for ocean forecasting. *Naval Research Reviews*, (51):28–38, 1999.

[43] A. R. Robinson and J. Sellschopp. *Rapid assessment of the coastal ocean environment in ocean forecasting: Conceptual Basis and Applications*. Springer-Verlag, 2000.

[44] A. R. Robinson and the LOOPS Group. Realtime forecasting of the multidisciplinary coastal ocean with the littoral ocean observing and predicting system (LOOPS). In *Preprint Volume of the Third Conference on Coastal Atmospheric and Oceanic Prediction and Processes*, pages 30–35, New Orleans, LA, November 1999. American Meteorological Society.

[45] S. K. Robinson. Coherent motions in the turbulent boundary layer. *Annual Review of Fluid Mechanics*, 23:601–639, 1991.

[46] Martin Roth. *Automatic extraction of vortex core lines and other line-type features for scientific visualization*. PhD thesis, Swiss Federal Institute of Technology Zurich, 2000.

[47] N. Rott. On the viscous core of a line vortex. *Z. Agnew Math. Mech.*, 9:543–553, 1958.

[48] R. H. Sabersky, A. J. Acosta, E. G. Hauptmann, and E. M. Gates. *Fluid Flow*. Prentice Hall, New Jersey, fourth edition, 1999.

[49] A. Sadarjoen and F. H. Post. Detection, quantification, and tracking of vortices using streamline geometry. *Visualization and Computer Graphics*, 24:333–341, 2000.

[50] A. Sadarjoen, F. H. Post, B. Ma, D. C. Banks, , and H-G Pagendarm. Selective visualization of vortices in hydrodynamics flows. In *Proc. Visualization '98*, pages 419–422, Research Triangle Park, NC, October 1998.

[51] A. Sadarjoen, T. Walsum, A. J. S. Hin, and F. H. Post. Particle tracing algorithms for 3D curvilinear grids. In *Proc. 5th EuroGraphics Workshop on Visualization in Scientific Computing*, Rostock, Germany, 1994.

[52] M. A. Spall and A. R. Robinson. A new open ocean, hybrid coordinate primitive equation model. *Math. and Comp. in Sim.*, (31):241–269, 1989.

[53] R. Spence, L. Tweedie, H. Dawkes, and H. Su. Visualization for functional design. In *Proc. Int'l Symp. Information Visualization (InfoVis '95)*, pages 4–10, 1995.

[54] R. Strawn, D. Kenwright, , and R. Haimes. Computer visualization of vortex wake systems. *AIAA Journal*, 37(4):511–512, 1999.

[55] D. Sujudi and R. Haimes. Identification of swirling flow in 3-D vector fields. In *AIAA Paper 95–1715*, 1995.

[56] R. D. Sullivan. A two-cell vortex solution of the Navier-Stokes equations. *J. Aerosp. Sci.*, 26:767–768, 1959.

[57] J. J. van Wijk and R. D. van Liere. Hyperslice. In *Proc. Visualization '93*, pages 119–125, 1993.

[58] L. J. Walstad and A. R. Robinson. A coupled surface boundary layer – quasi-geostrophic ocean model. *Dyn. Atmos. and Oceans*, (18):151–207, 1993.

[59] D. F. Watson. *Contouring: A Guide to the Analysis and Display of Spatial Data*, volume 10 of *Computer Methods in the Geosciences*. Pergamon Press, 1992.

[60] J. Wilhelms, J. Challinger, N. Alper, and S. Ramamoorthy. Direct volume rendering of curvilinear volumes. *Computer Graphics*, 24(5):41–47, 1990.

[61] T. Wischgoll and G. Scheuermann. Detection and visualization of closed streamlines in planar flows. *Visualization and Computer Graphics*, 7(2):165–171, April-June 2001.

[62] C. M. Wittenbrink, A. T. Pang, and S. K. Lodha. Glyphs for visualizing uncertainty in vector fields. *Visualization and Computer Graphics*, 2(3):266–279, 1996.

[63] Z. Yu. *Harvard Ocean Prediction System (HOPS) User's Guide*. Harvard University, Cambridge MA, October 2001.

[64] Z. F. Zhu and R. J. Moorhead. Extracting and visualization ocean eddies in time-varying flow fields. In *Proceedings of the 7th International Conference on Flow Visualization*, Seattle, WA, 1995.