

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Diplomarbeit Nr. 3131

Analyse von FTLE-Feldern (Analysis of FTLE fields)

Tom Schilli

Studiengang: Informatik

Prüfer: Prof. Dr. Daniel Weiskopf

Betreuer: Dipl. Inf. Marcel Hlawatsch
Dipl. Inf. Markus Üffinger

begonnen am: 20. Januar 2011

beendet am: 22. Juli 2011

CR-Klassifikation: I.3.3, I.4.10, I.6.0

Inhaltsverzeichnis

1	Einleitung	1
2	Verwandte Arbeiten	3
2.1	FTLE	3
2.2	Clustering	3
3	Grundlagen	5
3.1	Fachbegriffe	5
3.2	FTLE - finite-time Lyapunov exponent	7
3.2.1	Berechnung des FTLE	7
3.2.2	Berechnung der Gradienten mittels finiter Differenzen	8
3.2.3	Bestimmung des größten Eigenwertes	9
4	Caching und Performance-Steigerung	11
4.1	Zugriff auf Strömungsdaten: Speedup durch Hardwareinterpolation	11
4.2	Zweck der Integration	11
4.3	Zugriff auf die Strömungsdaten	11
4.4	Normalisierung der FTLE-Werte auf der GPU	12
4.5	Caching der integrierten Strömungsdaten: flow map	14
4.5.1	Erstellung der flow maps	14
4.6	Caching der flow maps	15
4.6.1	Versuch: Einsparung von Kernel-Calls	16
4.6.2	Beschleunigte Berechnung der Diagramm-Daten	17
5	FTLE-Werteverlauf	19
6	Implementierung	23
6.1	Berechnung der Diagramme	23
6.2	Validierung	24
6.2.1	Validierung der flow map	24
6.2.2	Validierung des RK4-Integrators	25
6.2.3	Validierung des Gesamtsystems	25
6.2.4	Diskretisierungsprobleme	26

7 Auswertungsmethoden	29
7.1 Transformation der Kurven	29
7.1.1 Berechnug des FSR	29
7.2 Betrachtung des Gradienten	30
7.2.1 Fourier-Transformierte des Werteverlaufs	33
7.3 Clustering	34
7.4 Visuell vereinfachte Untersuchung von Symmetrien durch Clustering	38
7.5 Kriterium: FTLE-Wert fällt über ein gewisses Intervall unter Max	43
7.6 Kriterium: FTLE-Wert steigt über ein gewisses Intervall über Min	43
7.7 Schwierigkeit: Überschreiten der Datensatzgrenzen	46
7.7.1 Lösungsansätze	46
7.8 Kombination beider Kriterien	49
7.9 Resonanzen und Drehgeschwindigkeiten von Wirbeln	49
7.9.1 FFT	52
8 Auswertung	55
8.1 Strömungsdatensätze	55
8.1.1 Analytisch vorgegebene Strömungsdatensätze	55
8.1.2 Strömungsdatensätze auf regulären Gittern	56
8.1.3 Datensatz: elektrisches Feld	56
8.1.4 Mit Matlab als partielle Differentialgleichung erstellte Datensätze	58
8.2 Circle-Datensatz	61
8.3 Bouyancy-Datensatz	64
8.4 Elektro3-Datensatz	64
9 Zusammenfassung	73
9.1 Ausblick	74
Literaturverzeichnis	75
Abbildungsverzeichnis	81

Kapitel 1

Einleitung

Die Berechnung des FTLE (finite-time Lyapunov exponent) bietet die Möglichkeit, die Separation von Partikeln in Strömungen zu visualisieren. Dies bedeutet nicht die Trajektorie des einzelnen, masselosen Partikels darzustellen, sondern zu ermitteln, ob eine Teilchengruppe innerhalb eines Betrachtungszeitraumes nahe beieinander bleibt (geringe Separation) oder sich voneinander entfernt (starke Separation).

Um diese Separation der Teilchen darzustellen wird zuerst ein Startpunkt vorgegeben. An diesem wird begonnen, die Positionsänderung der Teilchen über einen Zeitraum zu verfolgen. Statt nun nur den Endzustand zu betrachten wird in dieser Arbeit die Veränderung der Werte über den kompletten Integrationszeitraum betrachtet. Hierbei kann zwischen Teilchen unterschieden werden, die sich über den kompletten Zeitraum immer weiter entfernen - andere nähern sich nach einer gewissen Zeit einander wieder an. Diese Phänomene können nur bei Betrachtung über den kompletten Betrachtungszeitraum erkannt werden. In einem einzelnen FTLE-Bild, bei welchem lediglich die Separation der Teilchen zum Endzeitpunkt betrachtet wird, liegt diese Information nicht vor. Teilchen, die sich zuerst entfernen und dann wieder annähern, treten zum Beispiel beim Umströmen von Hindernissen auf. Ebenfalls können solche Effekte in Wirbeln beobachtet werden. Bei Wirbeln können sogar wiederkehrende, periodische Muster im Verlauf des FTLE-Wertes erkannt werden, die Periodendauer gibt Aufschluss über die Drehgeschwindigkeit des Wirbels. Die Analyse des FTLE-Werteverlaufes ist somit ein interessantes Instrument, um noch mehr Informationen über ein Strömungsfeld zu erhalten.

In dieser Arbeit werden die zeitlichen Vorgänge in zweidimensionalen FTLE-Feldern untersucht. Hierzu wird eine Programm entworfen, mit welchem die FTLE-Werte der Strömungsdaten angezeigt werden können. Innerhalb gewählter Bildbereiche können Diagramme über die (zeitlichen) Veränderungen der Werte generiert werden. Ebenfalls werden auf Basis dieser Kurvenverläufe Operationen wie Clustering oder Detektion sinkender beziehungsweise steigender Kurven durchgeführt. Um eine für die Durchführung der Arbeiten angenehme Berechnungsgeschwindigkeit zu erreichen werden Berechnungen auf der Grafikkarte durchgeführt. Für die Validierung des entwickelten Algorithmus wurden Datensätze erzeugt.

Kapitel 2

Verwandte Arbeiten

2.1 FTLE

Eine sehr anschauliche Einführung in die Eigenschaften und Berechnung des Finite Time Lyapunov Exponent (FTLE) liefert [Ill05]. Die grundlegenden Arbeiten wurden von Haller et al. [HP98], [Hal02], [HY00] sowie Shadden et al. [SLM05] geleistet. [SLM05] beschreibt auch die Verwendung einer flow map. Die Verwendung einer flow map wurde auch in dieser Arbeit aufgegriffen und durch das Caching der flow map erweitert. Detailliertere Untersuchungen der Beschleunigung durch Reduktion redundanter Partikelintegrationen mittels flow map führen [BR10] durch. Hierbei werden auch auf Speicherverbrauch und erreichter Geschwindigkeitszuwachs betrachtet.

Weitere Erläuterungen des zur Berechnung des im FTLE enthaltenen Cauchy Green Tensors enthält eine Arbeit von Ihlenburg [Ihl11]. Einen bebilderten Vergleich zwischen der Visualisierung mit LIC und der Visualisierung mittels FTLE liefert [STW⁺08].

Hlawatsch et al. beschreiben in [HSW11] eine Methode zur Beschleunigung der Berechnung von Gruppen von Trajektorien in Vektorfeldern und führen mit dieser Performancemessungen von FTLE und LIC auf der GPU durch. Die Analyse von Lagrangian Coherent Structures basiert auf Berechnung des Lyapunov Exponents und wird in [Hal01] und [HVS11] erläutert. Eine Abwandlung der Lagrangian Coherent Structures zu 2-dimensionalen Vektorfeldanalyse verwenden Sadlo und Weiskopf in [SW10]. Sadlo et al. behandeln die zeitabhängige Visualisation von Lagrange Coherent Structures mittels Grid Advection [SRP09].

Die zur Berechnung der Gradienten verwendete Methode der finiten Differenzen wird in [KR10] sowie [NM08] behandelt. Finite Differenzen werden in dieser Arbeit zur Berechnung des Gradienten auf diskreten Gitterzellen der flow map verwendet. Detaillierte Analysen zur Erkennung von Strukturen in Strömungsdaten führten bereits de Leeuw und van Liere [LL] durch. Eine Beschleunigung der Berechnung mittels CUDA beschreibt [JV09]. [GLT⁺] kombiniert FTLE und LIC und führt die Berechnung ebenfalls auf der Graphics Processing Unit (GPU) durch. In dieser Arbeit wird ebenfalls CUDA verwendet, um die Berechnungen auf der GPU durchzuführen.

2.2 Clustering

In [MCDZL05] führen Ma et al. Clustering auf sich zeitlich verändernden Gendaten durch. Hierbei wird statt des k-means Algorithmus ein Ansatz mit Approximation von Splines angewendet. Des weiteren wird auf die Wahl der Clusteranzahl eingegangen.

[AGDJ09] behandelt das Segmentieren von Funktionsfeldern mittels Range-Space Segmentation und vergleicht diese Ergebnisse mit Vector Quantization Clustering. Beim Clustern eines hochdimensionalen

Feldes tritt starkes Rauschen auf, auf welches auch in [JMF99] eingegangen wird. K-Means Clustering wird von [PP09] mit range trees kombiniert, um eine schnelle Suche von Daten unter Vorgabe von Wertebereichen zu ermöglichen. Als Anwendungsgebiet wird personalisiertes Clustern von Daten in e-commerce-Anwendungen angegeben. Karch vergleicht GPU-beschleunigte Clusteringalgorithmen in [Kar10].

Die Grundlagen des k-Means Algorithmus wurden 1967 von J.B. MacQueen [Mac67] veröffentlicht. Dabei wird bereits auf N-Dimensionale Felder eingegangen. In dieser Arbeit wird der k-Means Algorithmus für Clustering auf 2-dimensionalen Daten eingesetzt.

Kapitel 3

Grundlagen

3.1 Fachbegriffe

Pixel, Voxel und Texel

Computer können Daten nur in diskretisierter Form verarbeiten. Ein Pixel ist der kleinste Bildbereich eines zweidimensionalen Computerbildes. Sowie ein Pixel den Wert eines Feldes in einem zweidimensionalen Bild darstellt ist ein Voxel der Wert einer Zelle eines dreidimensionalen Datensatzes. Von einem Texel spricht man bei einem Bildpunkt auf einer Textur, unabhängig von deren Dimensionalität.

Vektorfeld

In einem Vektorfeld wird jedem Punkt im Raum ein Vektor zugeordnet. Im Folgenden werden nur 2-Dimensionale Vektorfelder betrachtet. Dies entspricht beispielsweise einem 2D-Strömungsdatensatz oder einer Ebene innerhalb eines 3D-Strömungsdatensatzes. Ein Vektor stellt bei der Strömungsvisualisierung die momentane Strömung an einem Punkt dar. Für das LIC-Verfahren in dieser Arbeit werden Vektoren nur für den zweidimensionalen Fall betrachtet.

CPU

Central Processing Unit (CPU) bedeutet Hauptprozessor. In jedem Computer befindet sich ein Prozessor, der für die Abarbeitung des laufenden Programms zuständig ist. Seine Stärke liegt in der raschen Verarbeitung sequentieller Programme und Abläufe. Trotz zunehmender Parallelisierung durch mehrere Cores bei DualCore (2 Cores) bzw. QuadCore (4 Cores) - Prozessoren liegen diese bei massiv parallelen Berechnungen leistungsmäßig hinter den aktuellen Grafikkarten.

GPU

Die Graphic Processing Unit (GPU) kann in Form einer Grafikkarte (Einsteckkarte) oder eines Grafikchips (direkt auf der Hauptplatine) verbaut sein. Durch die rasche Zunahme an Rechenleistung, Grafikspeicher und Flexibilität hinsichtlich der Programmierung werden diese Karten nicht mehr nur für Grafikausgabe sondern auch für Berechnungen auf großen Datenmengen verwendet. Geschwindigkeitsvorteile gegenüber der CPU werden erreicht, wenn auf vielen Datensätzen die gleiche Operation durchgeführt wird (SIMD). Dies ist unter anderem bei Wissenschaftlichen Berechnungen wie Strömungssimulationen oder dem hier behandelten FTLE-Verfahren der Fall.

SIMD

Von single instruction multiple data (SIMD) wird gesprochen, wenn auf mehrere Datensätze die gleiche Operation angewendet wird - beispielsweise wenn zu jedem Feld in einer Matrix ein Wert einer anderen Matrix addiert wird.

Eine ältere CPU (Single Core, ohne MMX, SSE etc.) muss hier sequentiell jede Multiplikation nacheinander durchführen. Eine GPU kann eine große Anzahl dieser Einzeloperationen in einem Zeitschritt abarbeiten.

Core

Core bedeutet „Kern“. Ein Prozessorkern besteht aus der Arithmetisch-Logischen Einheit (ALU), Registern, Cache und Steuerelektronik. In einer aktuellen CPU im Consumer-Sektor sind meist 4 Cores im Einsatz (Stand 2010). In Serven können CPUs mit einer höheren Anzahl Cores zum Einsatz kommen, wie beispielsweise der AMD Opteron 6164HE mit 12 Cores [AMD10a]. Eine aktuelle Grafikkarte wie die ATI FirePro™ V9800 hat 1600 Cores [AMD10b]. Die ALU eines Grafikkarten-Cores hat hingegen einen geringeren Befehlssatz als die ALU eines CPU-Cores. Auch sind die von NVidia angegebenen CUDA-Cores aufgrund des Architekturunterschieds nur schwer mit den Stream Processors von ATI vergleichbar. Ein besseres Kriterium zum Vergleich der Leistung ist die Anzahl Floating-Point Operationen pro Sekunde: eine Intel Core i7-975 - CPU mit 4 Cores leistet 55.36 GFlops (double precision, peak) [Int10], eine NVidia GTX275 leistet hingegen 1011 GFlops (single precision) [Spo09].

CUDA

Compute Unified Device Architecture (CUDA) bezeichnet sowohl eine Architektur als auch eine Programmierschnittstelle (API). CUDA ist eine Entwicklung von NVidia und stellt das Konkurrenzprodukt zu ATI Stream dar. Ziel beider Architekturen ist es, das Programmieren der Grafikkarte transparenter zu gestalten. Bei der früheren GPU-Programmierung mussten Shader und deren Besonderheiten für die Entwicklung berücksichtigt werden. CUDA hingegen erlaubt eine Programmierung in gewohnter C++-Umgebung, auch Speichertransfer zwischen Grafikspeicher und Hauptspeicher wird durch vorgefertigte Routinen vereinfacht. In dieser Arbeit werden die Strömungsdaten in den Texturspeicher der GPU kopiert, um von dort unter Verwendung von Hardwareinterpolation zugreifen zu können. Die Transformationsoperationen auf diesen Daten wird mit CUDA durchgeführt.

3.2 FTLE - finite-time Lyapunov exponent

Der finite-time Lyapunov exponent, FTLE, stellt ein Maß für die Separation innerhalb eines zeitabhängigen Strömungsdatensatzes dar. Anschaulich lässt sich dieses auf einem Gitter mit diskreten Partikel-Abständen zeigen (Abbildung 3.1): je weiter die Partikel, die sich zum Startzeitpunkt in direkter Nachbarschaft zum Partikel innerhalb eines vorgegebenen Zeitschritts voneinander entfernen, desto höher wird der resultierende FTLE-Wert. Der FTLE-Wert gibt lediglich eine Information über die Separation der Teilchen, nicht jedoch über deren Trajektorie oder Richtung. Eine ridge bedeutet, dass entlang dieser Kante eine hohe Separation herrscht (zum Beispiel an den Grenzen zweier Strömungen mit unterschiedlicher Strömungsrichtung) - ein Partikel, das sich auf der ridge befindet muss sich jedoch nicht entlang dieser bewegen.

Der FTLE-Wert $\sigma(\vec{x})_{T_0}^{\Delta t}$ muss für jede Position \vec{x} im Zielbild berechnet werden. Da diese Berechnungen für jedes \vec{x} unabhängig durchgeführt werden können wird eine parallele Berechnung der Werte auf der GPU implementiert. Einen Vergleich der Berechnungsdauer des FTLE zwischen CPU und GPU bietet [JV09]. Es wird gezeigt, dass eine Beschleunigung um mehr als Faktor 10 erreicht werden kann.

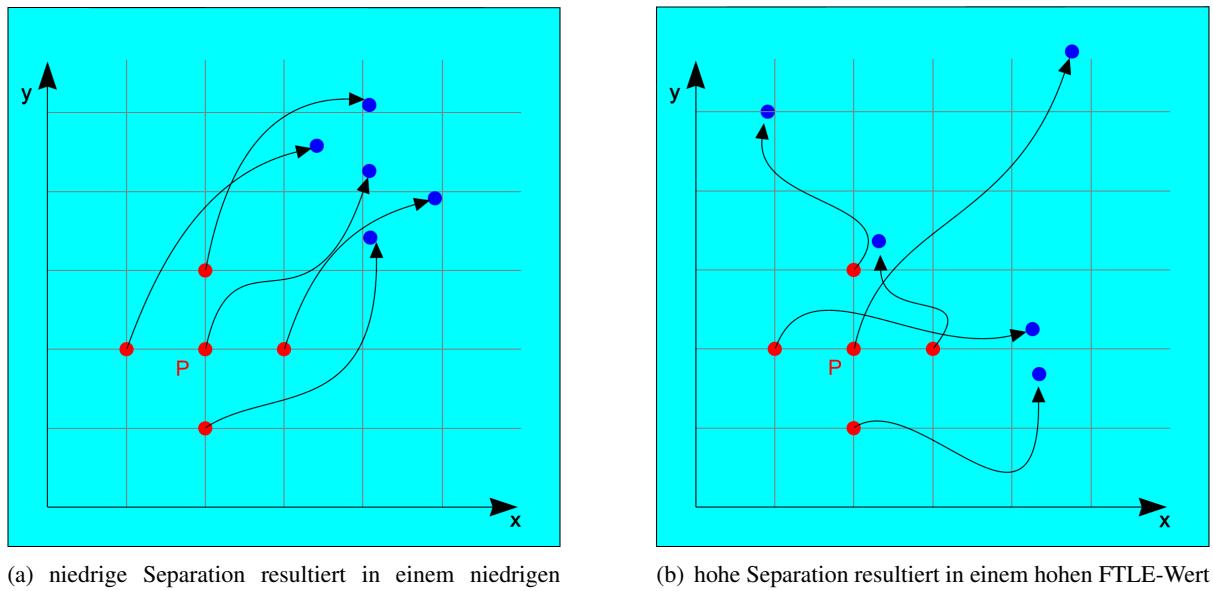


Abbildung 3.1: Veranschaulichung der Separation anhand der Trajektorie von Partikeln.

3.2.1 Berechnung des FTLE

Die mathematische Berechnung des FTLE wird in [SLM05] erläutert. Der FTLE-Werte $\sigma(\vec{x})_{T_0}^{\Delta t}$ an der Position \vec{x} mit den Parametern Startzeit und Integrationslänge ist durch folgende mathematische Vorschrift gegeben:

$$\sigma_{T_0}^{\Delta t}(\vec{x}) = \frac{1}{|\Delta t|} \ln \sqrt{\lambda_{max}(\Delta)} \quad (3.1)$$

Hierbei stellt $\lambda_{max}(\Delta)$ den maximalen Eigenwert von Δ dar. Δ ist der rechte Cauchy-Green Verzerrungstensor der Jakobimatrix $\nabla\chi$ und wird nach der Vorschrift

$$\Delta = \nabla\chi^T * \nabla\chi \quad (3.2)$$

ermittelt. Cauchy-Green Verzerrungstensoren liefern Auskunft über Verzerrungen und Deformationen und werden in [Ihl11] behandelt. Es ist zu beachten, dass Δ sowie $\nabla\chi$ von \vec{x} , T_0 und Δt abhängig sind. Der Übersichtlichkeit und Lesbarkeit wegen wird hier jedoch die verkürzte Notation Δ statt $\Delta(x; T_0; \Delta t)$ verwendet.

Die Jakobi-Matrix $\nabla\chi$ für den 2-dimensionalen Fall setzt sich aus den partiellen Ableitungen der Einträge der flow map $\xi_{T_0}^{T_0 + \Delta t}$ zusammen. Zur Erstellung der flow map werden die Strömungsdaten ausgehend vom Startzeitpunkt T_0 über einen Zeitraum Δt advektiert. Dieser Arbeitsschritt wurde hier in Form des Runge-Kutta-Verfahrens 4. Ordnung implementiert. Die Schrittweite des Integrators wird vom Benutzer vorgegeben. Alternativ sind auch andere Integratoren möglich. Da die Größe der vorgegebenen Schrittweite je nach Datensatz stark unterschiedlich ist kommt als Erweiterung ein Integrator mit adaptiver Anpassung der Schrittweite in Frage: hier müsste lediglich der maximale Fehler vorgegeben werden. Hierauf wurde jedoch verzichtet.

Für eine flow map mit 2-Dimensionalen Einträgen der Form

$$\xi(\vec{x}) = \begin{pmatrix} \xi_u(\vec{x}) \\ \xi_v(\vec{x}) \end{pmatrix} \quad (3.3)$$

ergibt sich eine Jakobimatrix der Form

$$J(\vec{x}, T_0, \Delta t) = \begin{pmatrix} \frac{\partial \xi_u}{\partial u} & \frac{\partial \xi_u}{\partial v} \\ \frac{\partial \xi_v}{\partial u} & \frac{\partial \xi_v}{\partial v} \end{pmatrix} \quad (3.4)$$

Für die Aufstellung der Jakobimatrix müssen die partiellen Ableitungen ermittelt werden. Da die Flow Map aus diskreten Werten auf einem rechteckigen Gitter besteht, wird zur Ermittlung der partiellen Ableitungen die Methode der finiten Differenzen angewandt.

3.2.2 Berechnung der Gradienten mittels finiter Differenzen

Bei der Berechnung der Gradienten mit der Methode der finiten Differenzen wird auf die Punkte in der Nachbarschaft des zu untersuchenden Punktes zugegriffen. Hierzu wird auf benachbarte Gitterpunkte in x- und y-Richtung entsprechend Abbildung 3.2 zugegriffen.

Zur Auswertung der benachbarten Partikel wurde ein kreuzförmiger stencil 1. Ordnung gewählt: hierbei werden 4 Partikel in direkter Nachbarschaft entsprechend Abbildung 3.2 betrachtet. Prinzipiell sind zur Berechnung des FTLE auch beliebig komplexere stencils denkbar. Da die Untersuchung des Einflusses des stencils jedoch nicht Gegenstand dieser Arbeit ist, wird der einfachst mögliche stencil verwendet. Bei der Ermittlung der Gittergröße d_{grid} in physikalischer Einheit muss die Zellgröße der Gitterdiskretisierung des Strömungsdatensatzes beachtet werden. Die Berechnung der Ableitungen mittels finiter Differenzen wird in [KR10] und [Nat04] erläutert - benötigt wird die zentrale Differenz (central difference).

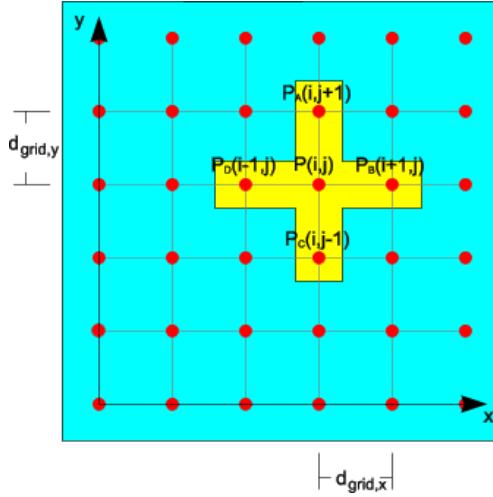


Abbildung 3.2: Für die Berechnung der finiten Differenzen gewählter stencil.

3.2.3 Bestimmung des größten Eigenwertes

Da die Bestimmung des maximalen Eigenwertes zur Berechnung des FTLE ebenfalls auf der GPU ausgeführt werden muss ist es sinnvoll, diese Berechnung soweit wie möglich zu vereinfachen und auf bezüglich der Berechnungsgeschwindigkeit teure Verzweigungen zu verzichten. Zur Bestimmung der Eigenwerte ist allgemein folgende Gleichung zu lösen:

$$\det(\Delta - \lambda \vec{E}) = 0 \quad (3.5)$$

Durch Einsetzen und Berechnung der Determinante kann dies in folgende Form gebracht werden:

$$\det \begin{pmatrix} f_{xx} - \lambda & f_{xy} \\ f_{yx} & f_{yy} - \lambda \end{pmatrix} = \lambda^2 - \lambda(f_{xx} + f_{yy}) + f_{xx}f_{yy} - f_{xy}f_{yx} \stackrel{!}{=} 0 \quad (3.6)$$

Eine 2x2-Matrix hat 2 Eigenwerte - diese Eigenwerte lauten:

$$\lambda_{1,2} = \frac{f_{xx} + f_{yy} \pm \sqrt{(f_{xx} + f_{yy})^2 - 4 * (f_{xx}f_{yy} - f_{xy}f_{yx})}}{2} \quad (3.7)$$

Für $\det \Delta \neq 0$ ist der rechte Cauchy-Green Tensor laut [Ihl11] symmetrisch und positiv definit und hat somit reelle Eigenwerte. Dadurch ist der maximale Eigenwert:

$$\lambda_{max} = \frac{f_{xx} + f_{yy} + \sqrt{(f_{xx} + f_{yy})^2 - 4 * (f_{xx}f_{yy} - f_{xy}f_{yx})}}{2} \quad (3.8)$$

Da nicht zuerst beide Eigenwerte bestimmt werden müssen und daraus der größere gewählt wird kann dieser verkürzte Rechenweg verwendet werden und auf ein auf der GPU teures if-statement verzichtet werden.

Kapitel 4

Caching und Performance-Steigerung

4.1 Zugriff auf Strömungsdaten: Speedup durch Hardwareinterpolation

4.2 Zweck der Integration

Der Strömungsdatensatz stellt Strömungsvektoren an diskreten Positions-/Zeitschritten zu Verfügung. Für die Berechnung des FTLE ist es jedoch notwendig, die Bewegung eines virtuellen, masselosen Partikels durch das Strömungsfeld zu erfassen: Zum gewählten Startzeitpunkt T_0 werden für jede zu berechnende Position ein virtueller Partikel gestartet, und dessen Position nach einer Integrationsdauer von ΔT Sekunden ermittelt. Als Integrator wurde das Runge-Kutta-Verfahren 4. Ordnung (RK4) gewählt.

4.3 Zugriff auf die Strömungsdaten

Der RK4-Integrator benötigt einen Zugriff auf beliebige Positionen im Strömungsfeld. Dieser Zugriff wird über eine 3-dimensionale Textur auf der Grafikkarte realisiert. Für diese Textur steht somit die schnelle Hardware-Interpolation (trilineare Interpolation zwischen den Voxeln) zu Verfügung. Am Rand der Textur muss ein Offset von $\frac{1}{2}$ Texel beachtet werden. Im 2D-Fall ist der Zugriff in Abbildung 4.1 dargestellt. Bei einem Strömungsdatensatz mit der Dimension $\vec{dim}_{flow} = \begin{pmatrix} dim_{flow,x} \\ dim_{flow,y} \end{pmatrix}$ erfolgt die Umrechnung zwischen normalisierter Position im Strömungsdatensatz $\vec{P}_f = \begin{pmatrix} P_{f,x} \\ P_{f,y} \end{pmatrix}$ und Texturzuggriffskoordinate P_t wie folgt:

$$\vec{P}_f = \begin{pmatrix} x_f \\ y_f \end{pmatrix} = \begin{pmatrix} P_{f,x} * \frac{dim_{flow,x}-1}{dim_{flow,x}} \\ P_{f,y} * \frac{dim_{flow,y}-1}{dim_{flow,y}} \end{pmatrix} + \begin{pmatrix} \frac{1}{2dim_{flow,x}} \\ \frac{1}{2dim_{flow,y}} \end{pmatrix} \quad (4.1)$$

Problematisch sind allerdings die Randgebiete des Datensatzes. Wenn in x- bzw. y-Richtung außerhalb des Definitionsbereichs des Datensatzes zugegriffen wird, so wird durch clamping (Abschneiden) auf einen Randpixel des Datensatzes zugegriffen. Falls jedoch in z-Richtung, welche der Zeitachse im Datensatz entspricht, ausserhalb des gültigen Bereichs zugegriffen wird, darf nicht mit dem in Hardware implementierten Clamping zugegriffen werden. Hierbei würde sonst auf den Daten des letzten Zeitschritts weitergerechnet, was zwar den Eindruck erweckt es würde weitergerechnet, allerdings keine sinnvoll interpretierbaren Daten liefert. Deshalb wird beim Überschreiten des definierten Zeit-Bereichs eine Strömung von $\vec{v} = \vec{0}$ zurückgegeben.

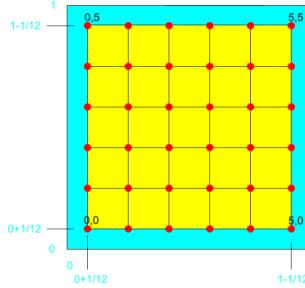
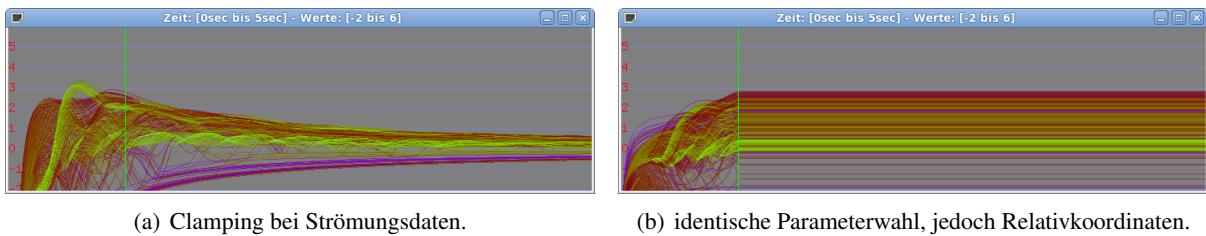


Abbildung 4.1: Koordinatentransformation beim Zugriff auf die Strömungsdaten-Textur.



(a) Clamping bei Strömungsdaten.

(b) identische Parameterwahl, jedoch Relativkoordinaten.

Abbildung 4.2: Textur-Clamping beim Strömungsdaten-Zugriff.

Einen Eindruck über das Verhalten liefern die in Abbildung 4.2 gezeigten Diagramme der FTLE - Werteverläufe: Bei dem 16 Sekunden langen Datensatz wurde bei $T0 = 15sec$ mit der Berechnung begonnen. Der grüne Strich markiert $\Delta T = 1sec$ und damit das Ende des definierten Bereichs. Im ersten Ansatz 4.2(a) wurde die Hardware-Clamping-Funktion verwendet, und somit auf dem letzten definierten Zeitschritt weiterintegriert. Die Schwankungen der FTLE-Werte hinter $\Delta T = 1sec$ entstehen lediglich durch Überschreiten von Zellgrenzen in x- und y-Richtung, resultieren jedoch alle aus dem selben Zeitschritt im Datensatz. Im zweiten Diagramm 4.2(b) wird bei Überschreiten des definierten Zeitbereichs $\vec{v} = \vec{0}$ erzwungen, wodurch ein Anhalten des virtuellen Partikels auf dem Integrationsweg erfolgt. Hierdurch bleiben die FTLE-Werte ab diesem Punkt konstant - dies ist deutlich intuitiver.

4.4 Normalisierung der FTLE-Werte auf der GPU

Um die FTLE-Daten anzuzeigen müssen diese normalisiert werden. Für eine Normalisierung ist die Berechnung des Minimums (Maximums) über alle FTLE-Daten notwendig. Diese Berechnung zu parallelisieren und somit zu beschleunigen ist mit Hilfe eines divide and conquer - Ansatzes möglich: Parallel wird von mehreren Teilbereichen des Arrays das Minimum (Maximum) bestimmt. Das Ergebnis der Teilbereiche wird daraufhin im nächsten Durchlauf verglichen - so lange, bis nur noch ein globales Minimum bzw. Maximum übrig bleibt.

Hierbei müssen jedoch die Eigenschaften der Hardware beachtet werden: Einen CUDA-Kernel zu starten ist im Vergleich zu dessen Ausführungszeit sehr zeitaufwändig, die Normalisierung soll deshalb mit nur einem Kernel-Aufruf vollständig durchführbar sein. Zusätzlich kommt erschwerend hinzu, dass eine Synchronisation während eines Kernel-Aufrufes nur zwischen den Threads eines Blockes möglich ist, nicht jedoch zwischen Blocks. Um nur ein globales Minimum (Maximum) nach dem Aufruf von nur einem Kernel zu erhalten, ist es also notwendig nur einen Block, dafür jedoch möglichst vielen Threads

zu starten. Auf der GPU kann bei der Berechnung in Teilbereichen die Lokalität der Daten genutzt werden: Für den CUDA-Block wird ein Speicherbereich für ein Min/Max-Paar im schnelleren shared memory der Grafikkarte reserviert. Dieser Speicherbereich wird für den Datenaustausch der Threads dieses Blocks verwendet.

Abbildung 4.3 zeigt den Ablauf der effizienten Min/Max-Suche auf der GPU. Es wird nur ein Block gestartet. Jeder der Threads untersucht einen Bereich der FTLE-Daten und schreibt diesen Wert in einen Speicherbereich des shared memory. Daraufhin werden die CUDA-Threads synchronisiert: Hierdurch wird sichergestellt, dass alle Threads abgeschlossen sind und somit alle Minima (Maxima) im shared memory gespeichert wurden. Abschließend detektiert ein einzelner Thread das globale Minimum bzw. Maximum aus dem Array des shared memory und schreibt dieses zurück ins global memory.

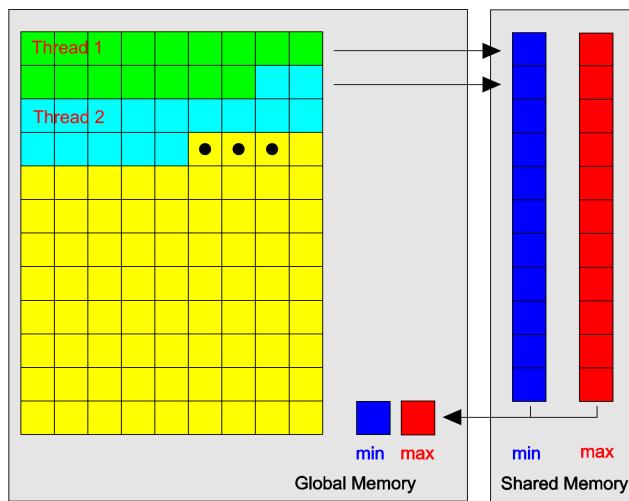


Abbildung 4.3: Berechnung des globalen Minimums/Maximums auf der GPU.

Da für jeden Thread ein Speicherblock im shared memory reserviert werden muss ist die Größe des shared memories eine obere Schranke für die Anzahl der einsetzbaren Threads. Grafikkarten mit Compute Capability 1.x erlauben eine Verwendung von 16kb shared memory, ab Compute Capability 2.0 werden bereits 48kb unterstützt (siehe [cud10], Appendix G.1). Bei Verwendung von float-Variablen (4 byte) können also maximal 2048 Min-Max-Paare gespeichert werden. Dies bedeutet, dass maximal 2048 Threads gestartet werden können. Bei einem FTLE-Feld mit 512x512 Werten müssen pro Thread lediglich 128 FTLE-Werte bearbeitet werden.

Bei Laufzeitmessungen wurde ermittelt, dass die Ermittlung von Min/Max mit dieser Methode bei einem Feld mit 512x512 Werten im Durchschnitt 0,8 Millisekunden benötigte und somit im Vergleich zur übrigen Berechnung praktisch vernachlässigbar ist.

Bei Datensätzen mit großer Dynamik können die Werte zwischen Minima und Maxima so verteilt sein, dass ein großer Teil der Werte sehr nahe an einem der Extreme liegt - und nur ein sehr kleiner Bildbereich den vollen Dynamikumfang ausnutzt. Hierdurch werden große Bildbereiche völlig schwarz (bzw. völlig weiß) dargestellt. Um gewünschte Details dieser Bereiche dennoch anzeigen zu können müssten Minima bzw. Maxima dann manuell vom Benutzer vorgegeben werden. Auch hat es sich als vorteilhaft erwiesen, negative Werte zu ignorieren indem das Minimum auf 0 gesetzt wird.

4.5 Caching der integrierten Strömungsdaten: flow map

4.5.1 Erstellung der flow maps

Eine flow map speichert für jedes Teilchen, dass sich zum Zeitpunkt T_0 an der Position \vec{x} befindet die neue Position \vec{x}' , an der es sich zum Zeitpunkt $t = k * \Delta t_{flowMapDist}$ befindet.

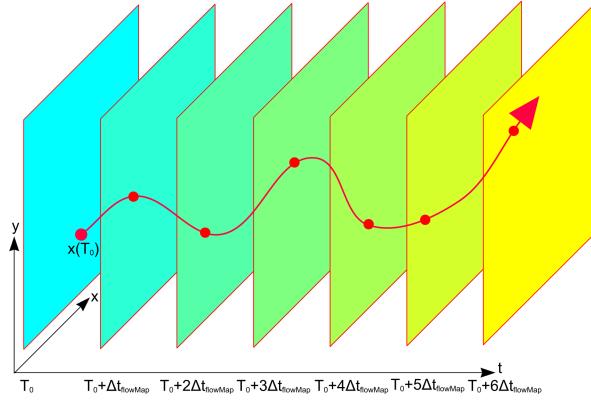


Abbildung 4.4: Aufbau der gecacheten flow map

Die hier benötigten flow maps sind 2-Dimensional. Als Größe der flow map wurde die Größe des Ausgabebildes gewählt.

Die Integration erfolgt unter Verwendung des Runge-Kutta Verfahrens 4. Ordnung. Die Schrittweite des Runge-Kutta-Integrators darf nicht mit dem für FTLE festgesetzten Zeitraum Δt verwechselt werden. Bei einer zu großen Schrittweite des Integrators bilden sich Artefakte aus, da die Lösung der Pfadlinien-Differentialgleichung zu unpräzise approximiert wird. Bei zu geringer Schrittweite wird der Rechenaufwand größer, ohne dass eine weitere Verbesserung des Ergebnisses eintritt. In Abbildung 4.5 ist dieser Effekt verdeutlicht: Bei einer sehr großen Schrittweite von einem slice des Bouyancy-Datensatzes benötigte die Berechnung der flow map lediglich 18.3 Millisekunden. Hierbei sind jedoch deutliche Artefakte entlang der ridges ausgebildet. In Abbildung 4.5(c) bilden sich wellenartige Verzerrungen aus. Wird eine präzisere Berechnung mit einer Integrator-Schrittweite von lediglich $\frac{1}{26}$ slice durchgeführt (siehe Abbildung 4.5(b)), so verschwinden die Artefakte. Die Berechnung benötigt nun jedoch 403 Millisekunden. Die Steigerung der Rechenschritte um den Faktor 26 hat einen Anstieg der Berechnungszeit um etwas weniger als Faktor 26 zur Folge, da das Kopieren der Daten einen konstanten Faktor darstellt der unabhängig von der Anzahl Integrationsschritte durchgeführt wird. In Abbildung 4.5(c) wurde eine noch geringere Schrittweite des RK4-Integrators gewählt. In Bezug zu Abbildung 4.5(b) ist keine Veränderung bzw. Verbesserung des Bildes erkennbar - die Berechnungszeit beträgt nun jedoch 4,7 Sekunden.

Eine automatische Wahl der Schrittweite kann unter Verwendung adaptiver Verfahren wie z.B. dem adaptiven Runge-Kutta Integrationsverfahren [PTVF] erreicht werden. Dabei muss vom Benutzer lediglich der maximal zulässige Fehler vorgegeben werden. Jedoch haben diese Verfahren bei der Implementierung den Nachteil, dass einzelne Berechnungen eine unterschiedliche Anzahl Schritte benötigen. Dies bringt bei paralleler Berechnung auf der GPU größeren Aufwand mit sich. Für die gewünschten Untersuchungen erwies sich eine Vorgabe der Schrittweite des Integrators durch den Benutzer als ausreichend.

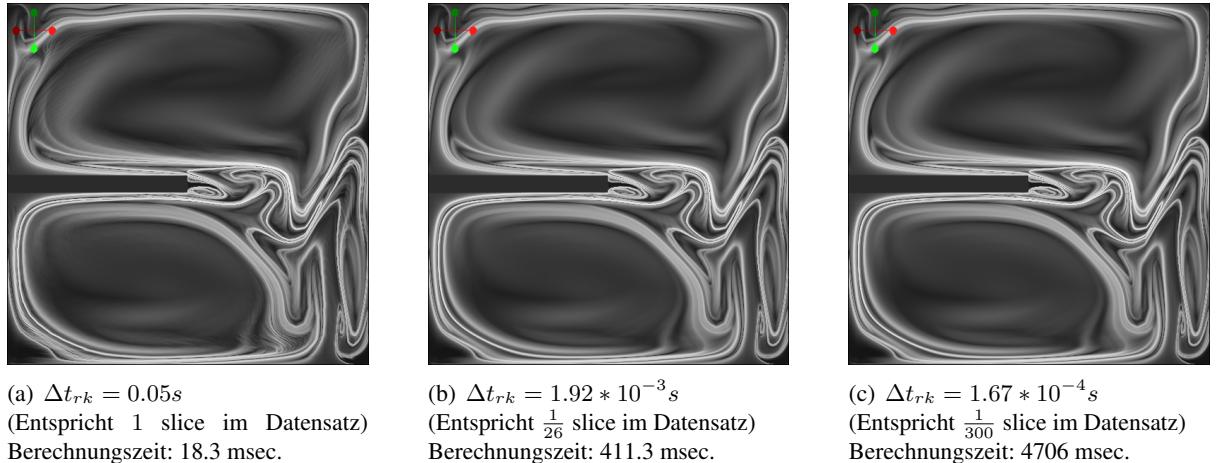


Abbildung 4.5: Vergleich unterschiedlicher RK4-Schrittweiten.

4.6 Caching der flow maps

Ziel ist es, innerhalb des Programms über einen Schieberegler die Integrationslänge Δt (nicht zu verwechseln mit der Schrittweite des RK4-Integrators Δt_{rk}) zu verändern um die gewünschte Einstellung interaktiv zu finden. Bei vollständiger Berechnung einer flow map nimmt die Berechnungszeit mit der Integrationslänge zu. Die Berechnungsdauer des FTLE-Wertes auf Basis der flow map sowie die Normalisierung sind hingegen unabhängig von Δt . Tabelle 4.1 wurde mit dem Bouyancy-Datensatz und einer Auflösung von 512x512 Feldern erstellt. Der Startzeitpunkt wurde auf $T_0 = 8.0s$ gesetzt, die Schrittweite des RK4-Integrators fest auf $\Delta t_{rk} = 1.92 * 10^{-3}s$ ($\frac{1}{26}$ slice). Als Berechnungsdauer ist die der flow map bezeichnet, zusätzliche Rechenzeit für FTLE und Normalisierung ist nicht inbegriffen. Für die Zeitmessung wurden jeweils 5 Messwerte gesammelt und gemittelt. Die Streuung zwischen

Δt	Berechnungsdauer
0.1s	52 msec
0.2s	101 msec
0.4s	202 msec
0.8s	411 msec
1.6s	874 msec

Tabelle 4.1: Berechnungsdauer der flow maps.

den Durchläufen betrug jeweils unter 3 msec. Aus den Messdaten geht hervor, dass bereits bei einer Integrationslänge größer als $\Delta t = 0.4s$ die Interaktivität nicht mehr gegeben ist.

Dieses Problem kann durch Caching der flow map gelöst werden: Es wird einmalig ein Array mit n flow maps mit $\Delta t_{flowMap,k} = (k+1) * \frac{T_{max}}{n+1}$ für $k = 0..n$ erstellt. Der Abstand zwischen 2 flow maps beträgt somit $\Delta t_{flowMapDist} = \frac{T_{max}}{n+1}$. Zur Berechnung einer flow map für ein angefordertes Δt wird auf die naheliegendste vorberechnete flow map zugegriffen und von dort aus weiterintegriert. Da auch rückwärts integriert werden kann beträgt somit der maximal zu integrierende Zeitbereich $\Delta t_{diff} = \frac{\Delta t_{flowMapDist}}{2}$. Aus Speicherplatzgründen können die flow maps nur für das jeweils aktuelle T_0 vorberechnet werden:

Eine einzelne flow map hat bei einer Auflösung von 512x512 Gitterpunkten und dem Datentyp float4 eine Größe von $s = 512 * 512 * 4 * 32\text{Bit} = 4\text{MB}$. Auf einer Grafikkarte mit 1GB Grafikspeicher passen somit theoretisch maximal 256 Felder. In der Praxis ist dieser Wert allerdings nicht erreichbar, da für weitere Berechnungen sowie die Darstellung des aktuellen Monitorbildes bereits Grafikspeicher reserviert ist. Es werden deswegen nur die flow maps für das aktuell gewählte T_0 vorberechnet - bei einer Veränderung des Parameters T_0 müssen die flow maps neu vorberechnet werden.

Um die optimale Anzahl vorberechneter flow maps in Abhängigkeit der Hardwareleistung zu ermitteln wird über ein Testintervall Δt_1 eine flow map vorberechnet und die Berechnungszeit t_{test} gemessen. Aus Tabelle 4.1 geht hervor, dass die Berechnungszeit weitgehend proportional zu Δt ist. Wenn als Parameter vom Benutzer eine maximale Berechnungszeit t_{bMax} vorgegeben wurde kann die optimale Distanz zwischen den flow maps bestimmt werden: $\Delta t_{flowMapDist, opt} = \frac{t_{bMax} * \Delta t_1}{2 * t_{test}}$.

Für die Vorberechnung von $n = 50$ flow maps bei $T_{max} = 8.0\text{s}$ benötigt der Referenzcomputer 7 Sekunden. Daraufhin kann der Schieberegler zur Veränderung von Δt jedoch ohne merkliche Verzögerung über den kompletten Bereich bewegt werden. Da eine Vorberechnungszeit in der Größenordnung von 7 Sekunden nach jeder Änderung von T_0 den Bedienungskomfort stark einschränkt wird die flow map nur auf Anforderung des Benutzers erstellt. Erst nach Auswahl des gewünschten Startzeitpunktes klickt der Benutzer auf den „create flow maps“-Button um die Vorberechnung anzustoßen.

4.6.1 Versuch: Einsparung von Kernel-Calls

Die Vorberechnung der flow map für Δt aus der gecacheten Flow map und die Berechnung des FTLE werden in getrennten Kernels durchgeführt. Beim Aufruf des FTLE-Kernels muss der Kernel zur Berechnung der flow map bereits vollständig abgeschlossen sein. Um eine Berechnung in einem Durchgang zu ermöglichen, müssen von einem Thread 4 Partikel-Positionen berechnet werden. Aufgrund der Lokalität der Daten können über das schnellere shared memory vorberechnete Partikelpositionen zwischen den Threads getauscht werden. Dem Nachteil des mehrfachen Rechenaufwandes steht somit der Vorteil der Einsparung eines Kernel-Calls sowie die Nutzung des shared memories entgegen. Um herauszufinden, welche Rechenmethode performanter ist, wurde ein Kernel entworfen, der beide Rechenschritte auf Einmal durchführt, und die Ergebnisse verglichen. Der Vergleich der Rechendauer wird mit Bouyancy-Datensatz bei einer Auflösung von 512x512 FTLE-Werten durchgeführt: Vorberechnet wurden $n = 150$ flow maps. Die Schrittweite für den RK4-Integrator beträgt $\Delta t_{rk} = \frac{1}{1000}\text{s}$. Der Startzeitpunkt liegt bei $T_0 = 4.57\text{s}$. Die Ergebnisse der Messreihe befinden sich in Tabelle 4.2.

Δt	Berechnungsduer getrennte Kernel	Berechnungsduer kombinierter Kernel
0.801s	88msec	264msec
1.1sec	129msec	442msec

Tabelle 4.2: Vergleich des FTLE mit unterschiedlichen Kernels

Der Rechenaufwand für die Integration (vorberechnete flow map bis zum gewünschten Δt) wird trotz Nutzung des shared memory praktisch vervierfacht - hingegen ein Kernel-Call eingespart. Der benötigte Zeitaufwand für die Berechnung wird ebenfalls vervierfacht. Die Integration in einem getrennten Kernel ist der Einsparung eines Kernels somit vorzuziehen.

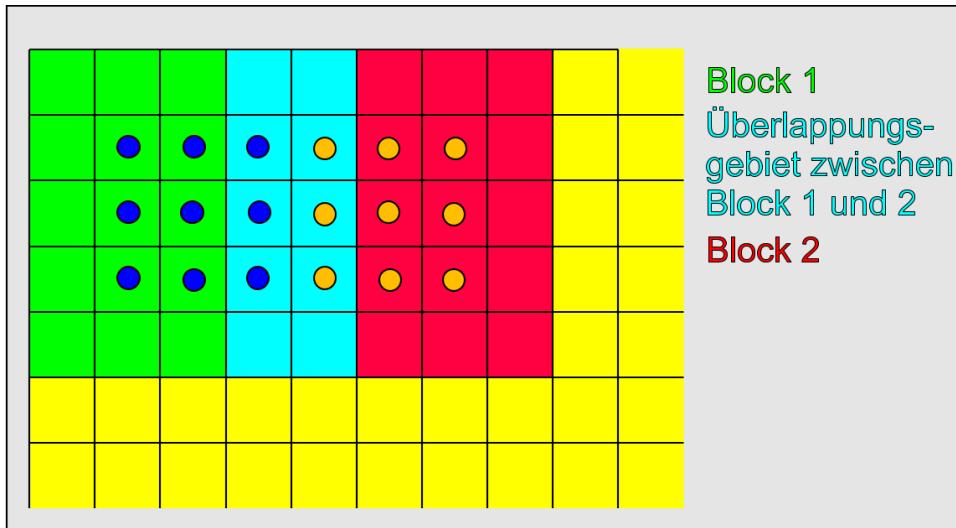


Abbildung 4.6: Berechnung der Diagrammdaten.

Ein Kernel-Aufruf verursacht laut [TSM] mit ca. $9\mu s$ overhead. Dies ist relativ klein im Vergleich zur Berechnungszeit im zweistelligen ms -Bereich. Durch eine Reduktion der Anzahl Kernel-Calls kann somit hier keine weitere Beschleunigung erzielt werden.

4.6.2 Beschleunigte Berechnung der Diagramm-Daten

Um die Diagrammdaten zu speichern wird ein 3-dimensionales Array verwendet. Die x- und y-Achse stellt hierbei die Koordinate im selektierten Bereich des FTLE-Feldes dar. Entlang der t-Achse sind die einzelnen Zeitschritte angeordnet. Mit Hilfe der flow map kann die Berechnung der FTLE-Werte für einen einzelnen gewünschten Zeitschritt beschleunigt werden. Da die t-Achse des Diagramms den kompletten Zeitbereich des Strömungsdatensatzes abdeckt kann durch die gecacheten flow maps keine Beschleunigung erreicht werden. Da von jeder jeweils am nächsten liegenden vorberechneten flow map bis zum aktuellen t des Diagramms integriert werden muss, kann hierbei gleich vom zuletzt berechneten Zeitschritt im Diagramm weiterintegriert werden. Dies hat auch zum Vorteil, dass wiederum das schnellere shared memory zum Einsatz kommen kann. Die Größe des shared memory ist hier ein begrenzender Faktor: Maximal 16x16 Zellen können gleichzeitig berechnet werden. Die Blockgröße wird dem entsprechend auf 16x16 Threads festgelegt. Aufgrund des für die finiten Differenzen benötigten kreuzförmigen stencils können auf Basis dieser lokalen 16x16 Felder-flow map lediglich 14x14 FTLE-Werte berechnet werden. Dementsprechend müssen sich die Berechnungsgebiete der einzelnen Blocks überlappen. Diese Überlappung veranschaulicht Abbildung 4.6.

Zur Vereinfachung wurde in der Abbildung die Block-Größe auf 5x5 reduziert. In grün ist das Gebiet dargestellt, auf dem Block 1 die Strömungsdaten integriert. Diese Werte werden im shared memory dieses Blocks hinterlegt. Daraus werden die FTLE-Werte für die Positionen berechnet, an denen die Kreise mit blauem Kern eingetragen sind. Der zweite Block (rot) erzeugt die FTLE-Werte mit den orangenen Kreiskernen. Im türkis gefärbten Gebiet überlappen sich die Gebiete, für die der jeweilige Kernel die Strömungsdaten integriert. Dieser Mehraufwand muss in Kauf genommen werden.

Falls die Berechnung eines Diagrams über einen zu großen Bereich des FTLE-Feldes angefordert wird

reicht der Speicher der GPU nicht aus, um das komplette Array des Diagramms auf einmal zu berechnen. In diesem Fall wird der angeforderte Bereich automatisch mittels divide&conquer in mehrere kleinere Bereiche aufgeteilt und getrennt berechnet. Nach jeder Teilberechnung wird das erzeugte Array in den Arbeitsspeicher verschoben, um den GPU-Speicher frei zu halten. Von dort aus wird auf die Daten zum Zeichnen per OpenGL Display List und zur weiteren Auswertung zugegriffen.

Kapitel 5

FTLE-Werteverlauf

Um den in dieser Arbeit im Mittelpunkt stehenden, zeitlichen Verlauf der FTLE-Werte darstellen zu können, wird dieser in ein Diagramm gezeichnet. In der grafischen Oberfläche wird als erster Schritt der Bereich vorgegeben, für dessen FTLE-Werte das Diagramm gezeichnet wird (Abbildung 5.1). Um die zeitlichen Vorgänge im Diagramm mit dem Resultat der Visualisierung der FTLE-Werte für die gewählte Integrationslänge vergleichen zu können wird Δt durch einen grünen Balken im Diagramm dargestellt.

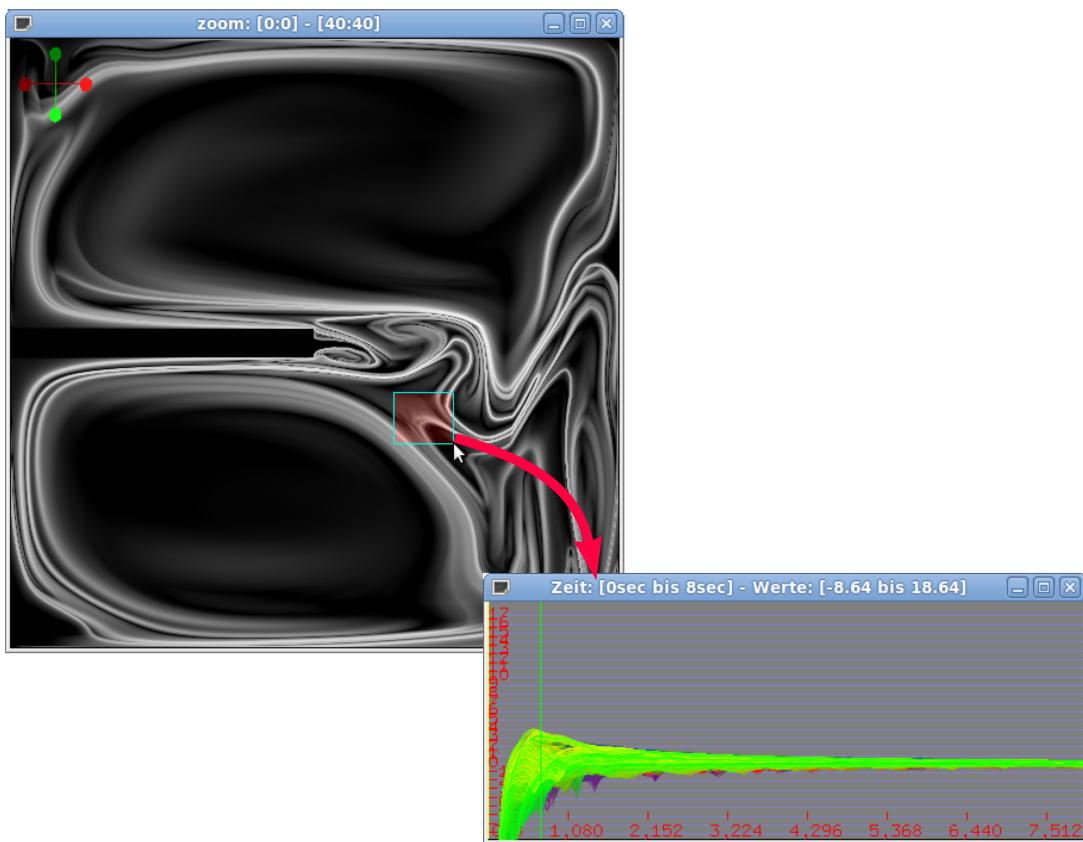


Abbildung 5.1: Vorgabe des Bereichs, über den der zeitliche Verlauf des FTLE-Wertes dargestellt wird.

Zur präziseren Betrachtung einzelner Bereiche des Diagramms kann in diese Bereiche hineingezoomt werden (Abbildung 5.2). Ein Zoomstack speichert hierbei den letzten Zustand, sodass zu diesem zurückgekehrt werden kann.

Bei der Darstellung der Diagramme kann zwischen FTLE-Wert, Gradienten des FTLE-Wertes und

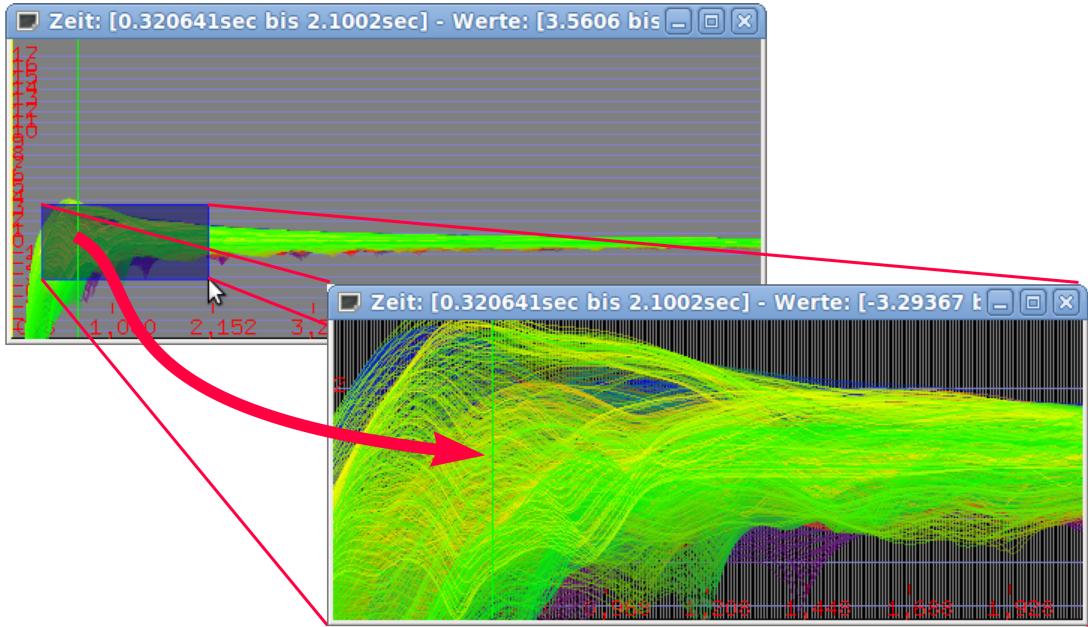


Abbildung 5.2: Betrachtung eines Ausschnitts mittels Zoom-Funktion.

Fourier-Transformierter gewählt werden (Abbildung 5.3).

Innerhalb des Diagramms ist es möglich, eine Selektion für den zu betrachtenden Zeitbereich zu definieren (Abbildung 5.4, Ziffer 1). Dieser Zeitbereich definiert das Gebiet, auf dem Clustering sowie die Anwendung von Kriterien, wie der Detektion von steigenden bzw. fallenden FTLE-Werten, durchgeführt werden.

Beim Clustering und der Selektion von Kurven nach Bereichen bzw. Kriterien können die jeweils ausgewählten Kurven optional hervorgehoben bzw. ausgeblendet werden. Die Kurven der FTLE-Werte gesetzter Particle Tracer werden in hellgrau und einer dickeren Liniendicke dargestellt (Abb. 5.4, Ziffer 3). Um neue Particle Tracer zu setzen kann die gewünschte Linie im Diagramm selektiert werden. Hierbei wird ein optisches Feedback in Form einer Hervorhebung der Kurve durchgeführt, die sich am nächsten am Mauszeiger befindet (Ziffer 4). Die Repräsentanten von Clustern werden in blau (Ziffer 2) dargestellt, falls dieses Cluster aktuell im FTLE-Bild angezeigt wird. Die übrigen Clusterrepräsentanten werden im Diagramm weiß (Ziffer 5) gezeichnet.

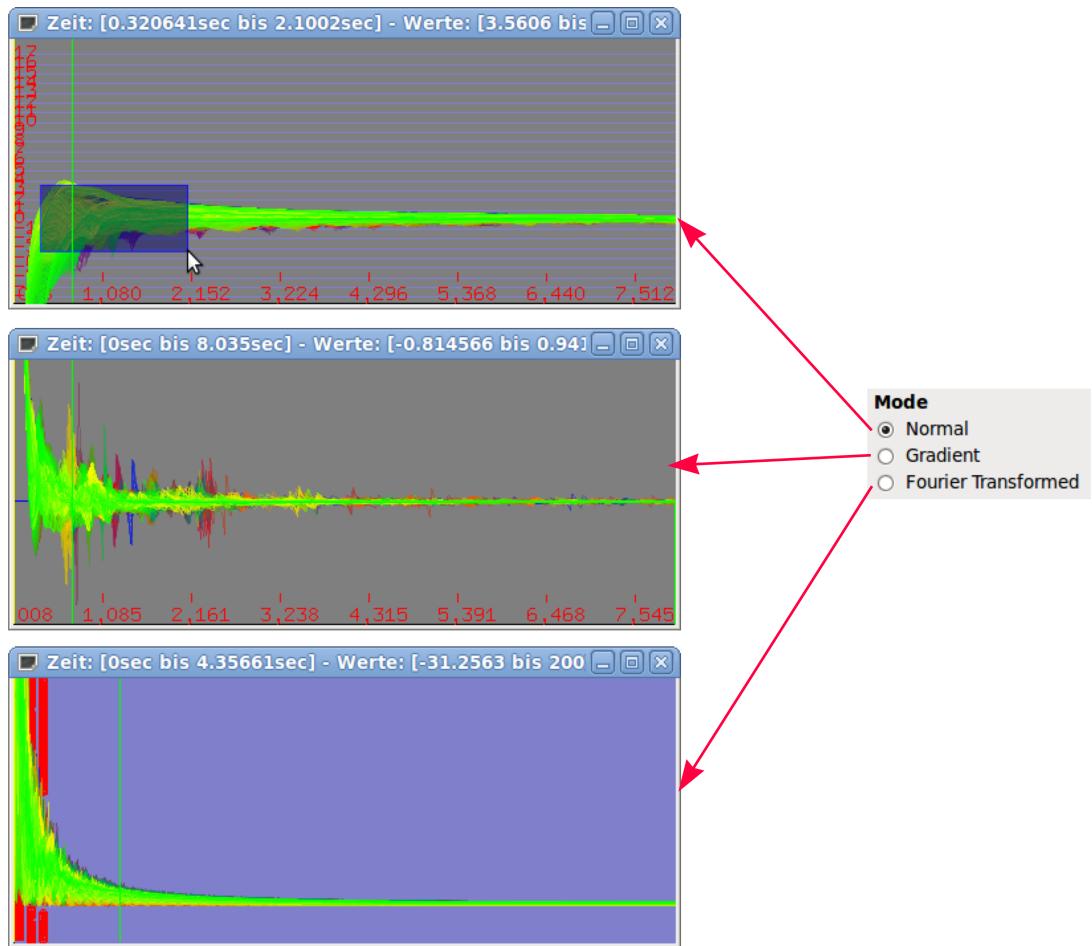


Abbildung 5.3: Unterschiedliche Modi: FTLE-Wert, Gradient oder Fourier-Transformierte.

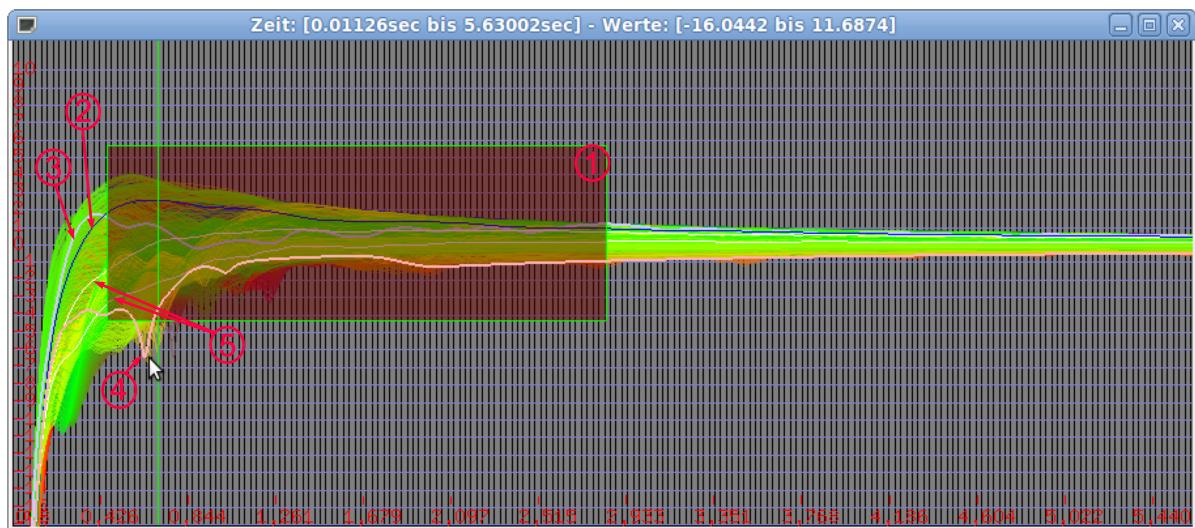


Abbildung 5.4: Features der Diagrammdarstellung.

Kapitel 6

Implementierung

Das Programm zur Durchführung der Auswertung wurde in `c++` geschrieben. Die grafische Oberfläche wurde mittels `Qt4` erstellt. Berechnungen auf der GPU wurden mit `CUDA` implementiert. `OpenGL` wird zum Zeichnen der Diagramme eingesetzt.

6.1 Berechnung der Diagramme

Um Rechenzeit und Speicheraufwand gering zu halten werden diese Kurven aus einzelnen Liniensegmenten zusammengesetzt. Die Koordinaten dieser Liniensegmente werden in einem Array gespeichert. Bei beispielsweise einer betrachteten Fläche von 512x512 Werten im FTLE-Feld und 1000 Zeitschritten (399 Liniensegmente pro FTLE-Wert) ergeben sich 262,144 Millionen zu speichernde FTLE-Werte. Bei Speicherung mit dem Datentyp `float` werden 4 byte pro Eintrag verwendet - in diesem Beispiel sind also bereits 1GB Speicherplatz notwendig. Da der Speicher der GPU hier schnell an seine Grenzen kommt und für die Speicherung des Strömungsdatensatzes, der flow map und andere Berechnungen ebenfalls GPU-Speicher frei bleiben muss, wird das Array mit dem Zeitverlauf der FTLE-Werte im RAM gespeichert. Die Berechnung erfolgt in Teilbereichen auf der GPU. Näheres zur Implementierung dieser schrittweisen Berechnung befindet sich in Kapitel 4.6.2.

Der Werteverlauf wird im Programm in einem Fenster wie in Abbildung 6.1 dargestellt. In Abbildung 6.1(a) wurden lediglich 38 Zeitschritte zur Approximation des Kurvenverlaufs verwendet. Sehr deutlich sind die Liniensegmente zu sehen. In vertikaler Richtung verläuft die Zeitachse: vertikale graue Linien zeigen an, dass an diesem Zeitpunkt eine Stützstelle der gezeichneten Kurven ist. Am unteren Rand des Diagramms befinden sich in roter Schrift die Zeit für Integrationslänge in Sekunden. Bei der Darstellung der Zeitwerte wird auf einen Minimalabstand zwischen den Achsenbeschriftungen geachtet - beim Vergrößern des Fensters erscheinen entsprechend mehr Einträge. Um ein sicheres Zuordnen der Zeiteinträge zu den vertikalen Linien zu gewährleisten wird oberhalb der roten Beschriftung ein kleines Segment der sonst grauen Linie in rot markiert. In horizontale Richtung ist der FTLE-Wert (nicht normalisiert) eingetragen. Die Achsenbeschriftung sowie die grauen, horizontalen Linien haben einen konstanten Abstand von 1. In Abbildung 6.1(b) wurde der selbe Ausschnitt des FTLE-Bildes mit einer höheren Auflösung von 1000 Zeitschritten abgetastet. Die Kurvenform wird hier deutlich genauer approximiert, die größere Anzahl Datenpunkte ist auch durch die häufigeren vertikalen Linien gut zu erkennen. Eine Linie im Diagramm ist dicker und in grauer Farbe gezeichnet: für diese Linie wurde im FTLE-Bild ein Partikel-Tracer gesetzt. Die vom Benutzer eingestellte Advektionsdauer der Partikel wird durch die gelbe vertikale Linie im Diagramm markiert. Die grüne vertikale Linie markiert die momentan eingestellte Integrationslänge Δt des Diagramms. In der Titelleiste des Fensters erscheint der aktuell im Diagramm sichtbare Werte- und Zeitbereich. Bei der Darstellung in Diagrammform

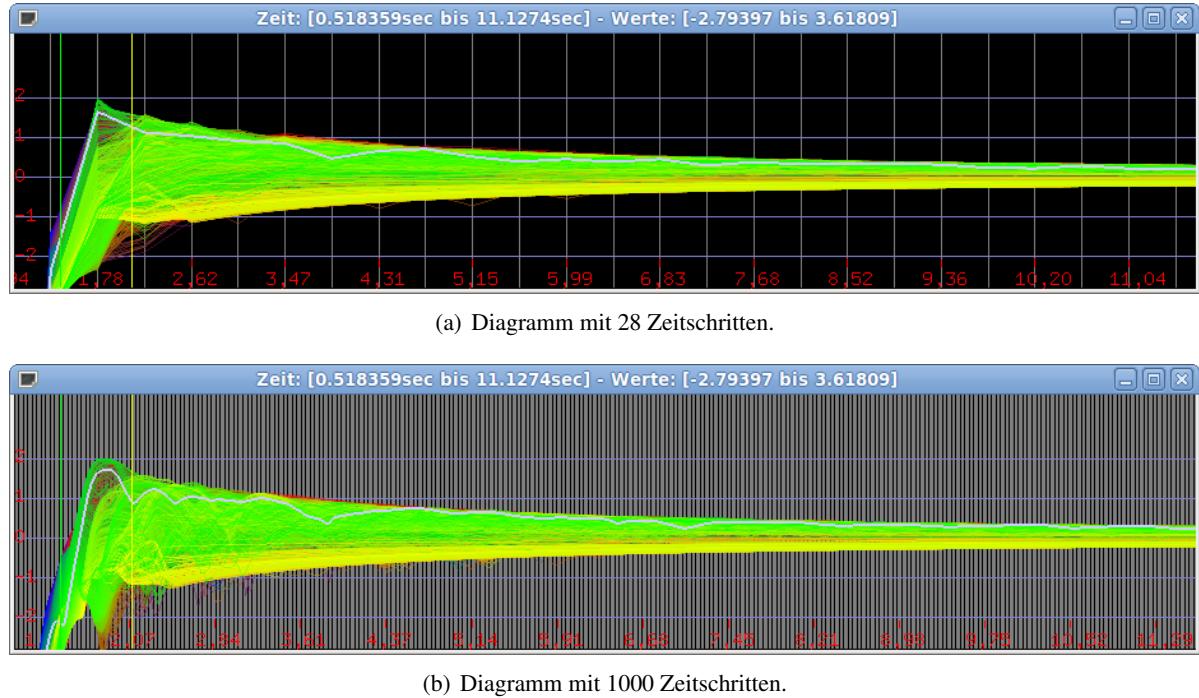


Abbildung 6.1: Diagramm-Darstellung des FTLE-Werteverlaufs.

werden Transformationen wie die in Kapitel 7.2 erläuterte Gradientenform bzw. die in Kapitel 7.2.1 angewendete Fouriertransformation berücksichtigt. Ebenfalls wird bei markierter Option „nicht durch Δt teilen“ (siehe Kapitel 7.1.1) der FTLE-Wert ohne entsprechende Division im Diagramm dargestellt.

6.2 Validierung

6.2.1 Validierung der flow map

Statt des FTLE-Wertes wurde zur optischen Prüfung die flow map dargestellt. Hierbei wurden in mehreren Durchläufen x und y-Wert bzw. die Länge des Vektors gezeichnet. Es konnten anfangs bereits die durch die begrenzte Präzision des float-Datentyps entstandenen Probleme (siehe Abschnitt 6.2.4) sichtbar gemacht werden. Als Darstellungsform wurden Graustufen gewählt. Auch wird hierbei sofort sichtbar, wenn Array-Indizes beim Zugriff auf die Strömungsdaten falsch berechnet wurden da das Bild dann entweder deformiert oder völlig unkenntlich erscheint. Erst wenn mit der optischen Prüfung keine offensichtlichen Fehler sichtbar sind kann zum zweiten Schritt weitergegangen werden, der Validierung gegen externe Daten. Hierbei wurde anhand des Bouyancy-Datensatzes eine flow map erstellt, und gegen einen Referenzdatensatz verglichen. Hierzu wurde $T_0 = 8.05\text{sec}$ und $\Delta t = 0.55\text{sec}$ gewählt, dies entspricht einem Start auf slice 161 des Bouyancy-Datensatzes. Die Auflösung der verglichenen flow maps beträgt $512 * 512$ Felder.

Beim Vergleich muss darauf geachtet werden, dass die hier erzeugten Relativkoordinaten der flow map wieder in Absolutkoordinaten umgewandelt werden müssen. Es wurde eine durchschnittliche quadrati-

sche Abweichung von $\varepsilon_{avg} = 0,0001$ festgestellt. Diese Abweichungen erscheinen bei der Berechnung mit Float-Werten als realistische, numerische Abweichungen.

6.2.2 Validierung des RK4-Integrators

Um die Funktion des RK4-Integrators zu prüfen wird ein synthetischer Datensatz verwendet, bei dem die Teilchen eine Drehung mit Umlaufdauer $T = 2\pi$ durchführen. Dieser Datensatz ist durch die Gleichung $\vec{v} = (x, -y)$ bestimmt, wobei \vec{v} die Strömung an der Position (x, y) darstellt. Dieser Datensatz wird über $\Delta t = 2\pi$ Sekunden integriert, wodurch jedes Teilchen wieder an seiner Ursprungposition angelangt ist, somit also in jedem Gitterpunkt der flow map die Koordinate dieses Punkts steht.

Eine weitere Integritätsprüfung stellt eine Integration mit einer darauffolgenden Integration gleicher Länge, jedoch entgegengesetzter Integrationsrichtung dar. Auf unterschiedlichen Datensätzen wird die Integration über eine Schleife n Schritte durchgeführt. Daraufhin folgt eine weitere Integration in die entgegengesetzte Richtung (t-Achse invertiert) mit der selben Anzahl Schritte. Rechnerisch befindet man sich wieder am Ausgangspunkt, die flow map muss somit an jeder Stelle einen Nullvektor-Eintrag haben. Bis auf die numerischen Schwankungen (7. Stelle hinter dem Komma) wurde dies erfüllt.

6.2.3 Validierung des Gesamtsystems

Um das Gesamtsystem und die Oberfläche zu validieren muss die Integrität der dargestellten Achsenbeschriftungen kontrolliert werden. Hierfür wird ein Datensatz geladen und die Parameter T_0 sowie Δt vorgegeben. Im FTLE-Bild wird ein beliebiger Bereich selektiert, damit ein Diagramm erzeugt wird. Zusätzlich wird ein Partikel-Tracer platziert. Nun kann die Zeitachse im Diagramm gegen die Einstellregler der Oberfläche kontrolliert werden. Der grüne Balken im Diagramm von Abbildung 6.2 zeigt das aktuell gewählte Δt an. Im Diagramm muss die Achsenbeschriftung (hier: 0.48) exakt mit dem im Hauptprogramm gewählten Wert übereinstimmen. Die Position des gelben Balkens, der die Integrationslänge des Particle-Tracers im Diagramm markiert, wird auf die selbe Weise überprüft. Beim Verändern der Schieberegler müssen die Balken entsprechend mitlaufen. Beim Zoomen im Diagramm kann mit Hilfe der farbigen Balken die korrekte Skalierung ebenfalls geprüft werden. Beim Zoomen im FTLE-Bild dürfen sich die Selektion sowie die verfolgten Partikel relativ zu markanten Stellen im Datensatz nicht verschieben.

Beim Selektieren eines hellen Gebietes im FTLE-Bild muss die Kurve im Diagramm höhere Werte als bei Selektion eines dunklen Gebietes aufweisen. Zusätzlich werden in diesen Gebieten Partikel verfolgt. Anhand des Verlaufes der 4 farbigen Punkte wird überprüft, ob die Werte plausibel sind: bei einem niedrigen Diagrammwert dürfen sich die Punkte nicht weit voneinander entfernt haben. Bei einem hohen Diagrammwert hingegen muss hinreichende Separation vorliegen. Dies wird in Abbildung 6.3 verdeutlicht. Bei der Selektion unterschiedlicher Bereiche ist darauf zu achten, dass die Kurven stets einen stetigen Verlauf haben. Bei einem Strömungsdatensatz, der keine Sonderfälle wie not a number (NaN), unendlich große Werte, harte Kanten die sich über die Zeit verschieben oder ähnliches aufweist sind Sprünge nicht zu erwarten. Dies gilt ebenfalls für die Repräsentantenkurven beim Clustering, welches in Kapitel 7.3 beschrieben wird. Unter Verwendung eines homogenen Datensatzes mit $\vec{v} = \vec{0}$ bzw. $\vec{v} = \vec{1}$ darf im Diagramm nur eine Kurve erscheinen, da alle Werte zusammenfallen.

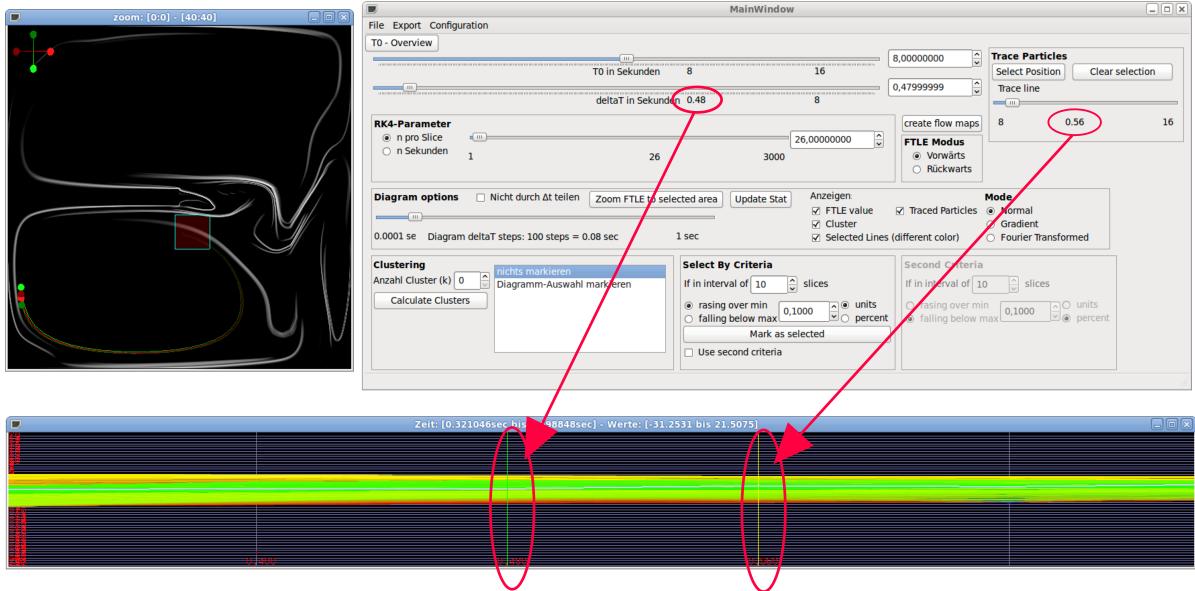


Abbildung 6.2: Überprüfen der Oberfläche.

6.2.4 Diskretisierungsprobleme

Bei der Validierung mit dem Datensatz, bei dem an jedem Punkt der selbe Fluss $\vec{v} = (1, 1)$ angegeben war, traten statt des erwarteten homogen schwarzen FTLE-Bildes ein gitterartiges Muster auf, dass sich bei Modifikation der Integrationslänge Δt veränderte. Bei Untersuchung mit einem Datensatz ohne Strömung ($\vec{v} = (0, 0)$) blieb das FTLE-Resultat erwartungsgemäß einheitlich schwarz.

Für die Artefakte bei Untersuchung des Datensatzes mit $\vec{v} = (1, 1)$ waren leichte Schwankungen der Werte in der flow map verantwortlich, die sich dann durch FTLE-Berechnung und die darauffolgende Normalisierung zu deutlich sichtbaren Mustern verstärkten. Die Schwankungen wurden durch Diskretisierungs- und Rundungsfehler verursacht: Ein Eintrag in der flow map entspricht der Position eines Teilchens nach der Advektion. Die Advektion und die Daten des Strömungsdatensatzes sind meist sehr gering bezogen auf die Dimension des Strömungsfeldes. So konnte an Position (0, 0) die Advektion noch mit voller float-Präzision gespeichert werden - an Position (10, 10) hingegen wurde durch den höheren Exponenten bereits Nachkommastellen abgeschnitten.

In Abbildung 6.4(a) sind die Artefakte bei der Darstellung des Bouyancy-Datensatzes zu sehen. Hierbei kam als zusätzlicher Störungsfaktor hinzu, dass unnötig oft zwischen den Koordinatensystemen der flow map, des Strömungsdatensatzes und physikalischen Koordinaten transformiert wurde und hierbei weitere Rundungsfehler zum Tragen kamen. In Abbildung 6.4(b) und 6.4(c) wurde diese Problematik mit Hilfe eines Datensatz mit konstantem Fluss von $\vec{v} = (1, 1)$ näher untersucht. Für Abbildung 6.4(b) wurde die Anzahl Koordinatensystemtransformationen bereits auf das Minimum reduziert. Erst mit der Speicherung relativer Koordinaten in der flow map verschwanden die Artefakte vollständig, wie in Abbildung 6.4(c) zu erkennen ist. Bei der Speicherung mittels relativer Koordinaten wird nur die Verschiebung eines Teilchens gespeichert - der Offset wird bei der Berechnung als getrennte Variable behandelt. Der Offset muss nicht zwingend gespeichert werden: da die Größe des Datensatzes bekannt ist, kann der Offset jederzeit wieder berechnet werden.

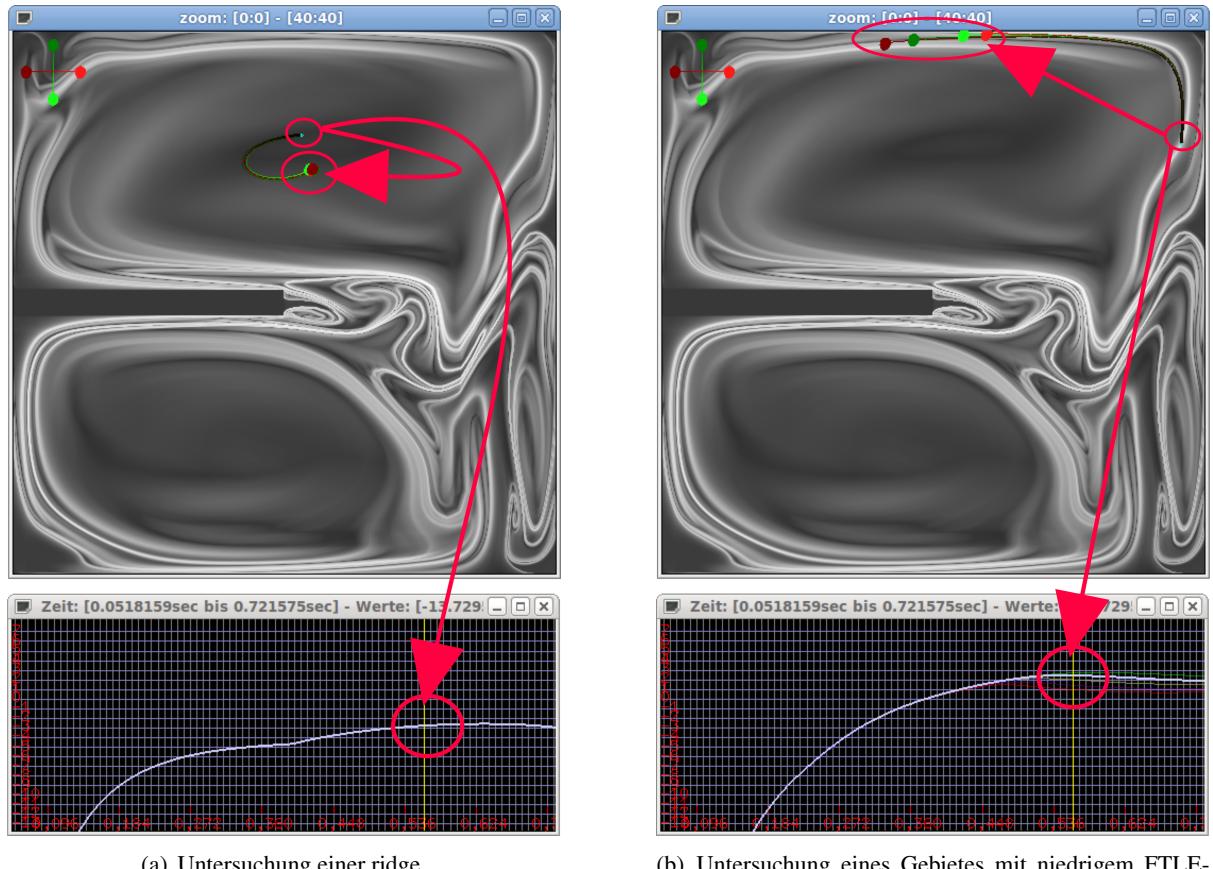


Abbildung 6.3: Vergleich unterschiedlicher RK4-Schrittweiten.

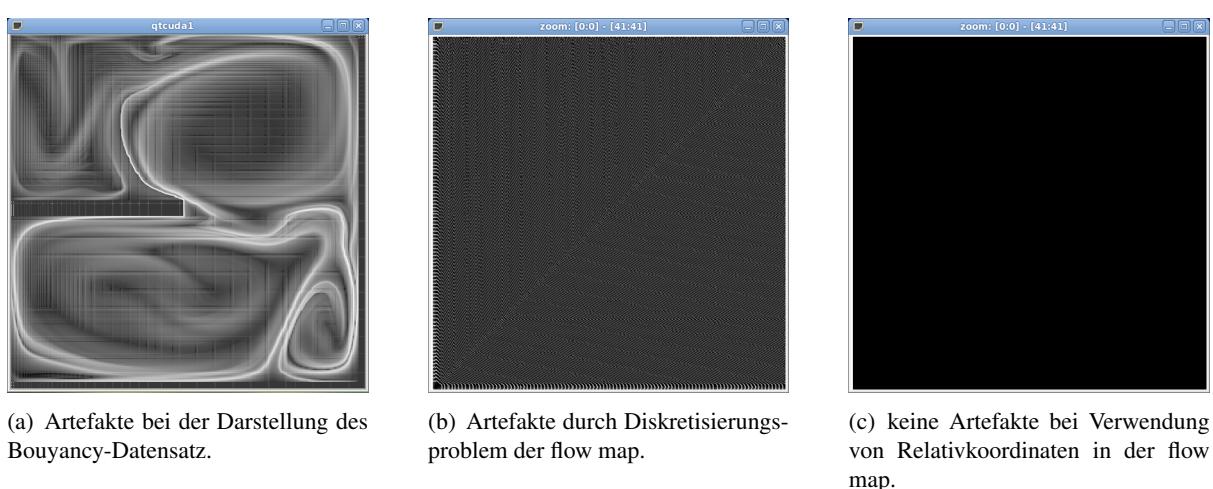


Abbildung 6.4: Flowmap: Diskretisierungsproblem und Lösung durch Relativkoordinaten.

Kapitel 7

Auswertungsmethoden

7.1 Transformation der Kurven

7.1.1 Berechnung des FSR

Bei der Berechnung des FTLE wird die in Abschnitt 3.2.1 behandelte Formel

$$\sigma_{T0}^{\Delta t}(\vec{x}) = \frac{1}{|\Delta t|} \ln \sqrt{\lambda_{max}(\Delta)} \quad (7.1)$$

angewendet. Da in dieser Arbeit ausschließlich räumlich begrenzte Datensätze verwendet werden ist ein exponentieller Anstieg der Separation über einen unendlichen Zeitraum ausgeschlossen. Somit gilt in diesem Fall, dass für große Δt der FTLE-Wert sich 0 annähert, da Δt im Nenner des Bruchs steht. Somit sinken die FTLE-Werte bei Erhöhung des Integrationszeitraumes Δt , auch wenn sich die Teilchen nicht annähern. Um die Annäherung von Partikeln ohne störenden Einfluss dieses Effekts untersuchen zu können, wurde die Oberfläche um die Möglichkeit erweitert, auf die Division mit Δt zu verzichten. Hlawatsch et al. definieren in [HVS11] den finite separation ratio (FSR). Für den Spezialfall $m = 1$ lautet dieser:

$$\tilde{\sigma}_{T0}^{\Delta t} = \sqrt{\lambda_{max}(\Delta)} \quad (7.2)$$

Um die Skalierung bei großen Werten zu erhalten wird der Logarithmus des FSR verwendet. Dies entspricht der Berechnung des FTLE ohne die Division durch Δt :

$$\sigma, noDivision_{T0}^{\Delta t}(\vec{x}) = \ln \sqrt{\lambda_{max}(\Delta)} \quad (7.3)$$

Der Unterschied im Verlauf der Kurvenschaar zwischen Abbildung 7.1(a) und Abbildung 7.1(b) ist signifikant. Für die Erstellung dieser Kurven wurde der selbe Ausschnitt des quadGyre-Datensatz verwendet. Wenn durch Δt geteilt wird ist beim gewählten Zoomfenster bereits ab $\Delta t = 6s$ ein Unterscheiden der einzelnen Kurven praktisch nicht mehr möglich, da diese sich zu weit angenähert haben. Auch ist das Annähern an den Grenzwert 0 gut zu beobachten. Verzichtet man hingegen auf das Teilen durch Δt , so wird deutlich, dass die FTLE-Werte auch bei größeren Integrationslängen stark unterschiedlich bleiben. Auch ist ein kontinuierlicher Anstieg der Werte zu betrachten: die Separation wächst bei steigendem Δt . Für die Suche nach sich annähernden Teilchen und die Betrachtung der Separation unabhängig von der Integrationsdauer ist die Darstellung des FTLE-Wertes ohne die Division durch Δt somit ein wichtiges Werkzeug. Auch auf das Ergebnis des Clusterings hat diese Transformation Einfluss: Bei normaler Berechnung nähern sich mit steigendem Δt die FTLE-Werte 0 an. Somit weisen diese auch gegeneinander immer geringere Differenzen auf, wodurch beim Clustering der Bereich mit niedrigem Δt dominant wirkt. Wird auf die Division jedoch verzichtet sind auch bei größeren Integrationszeiträumen noch Differenzen in den FTLE-Werten ausschlaggebend. Eine nähere Betrachtung findet in Kapitel 7.3 statt.

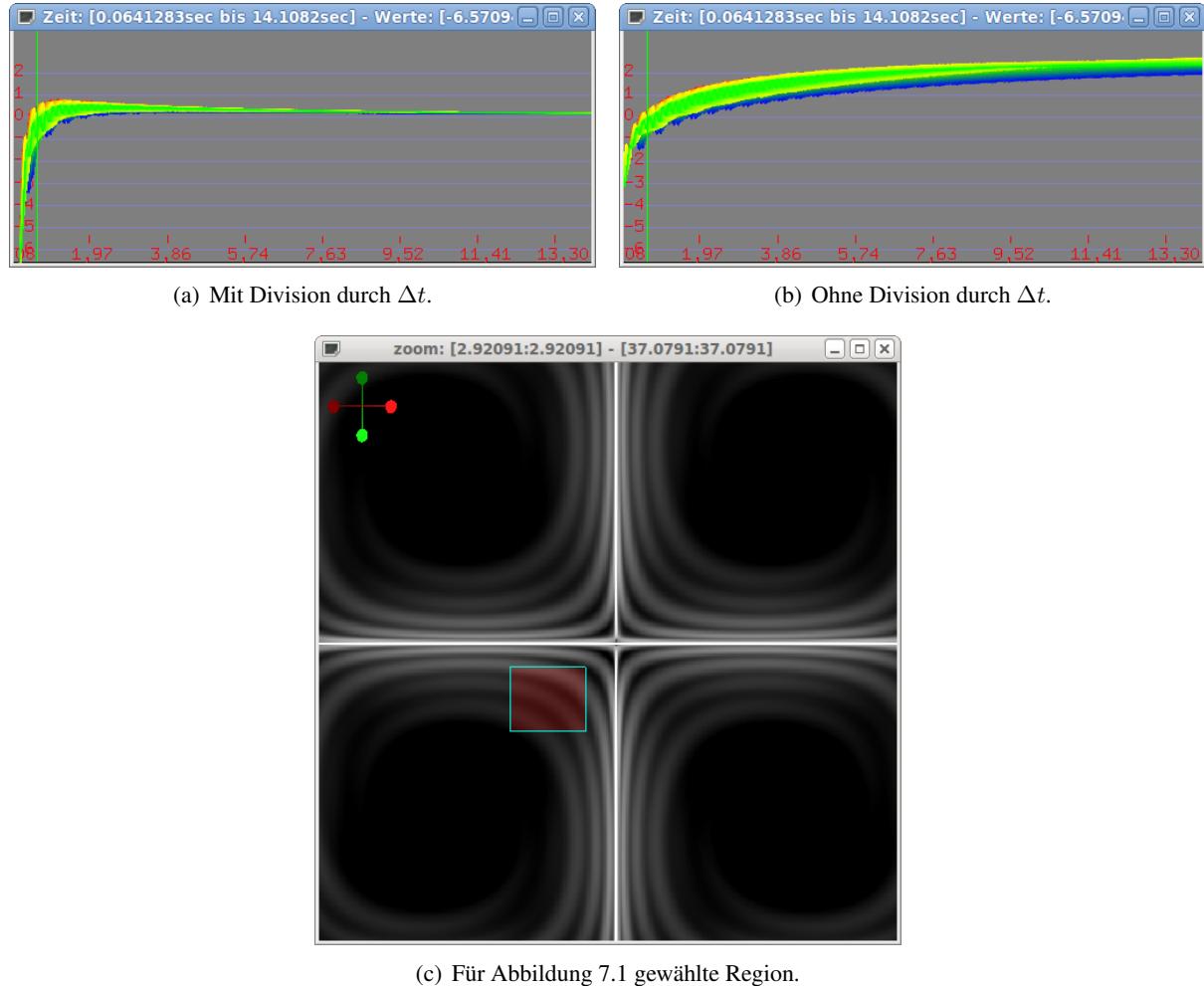


Abbildung 7.1: Vergleich der Kurvenschaar mit und ohne Division durch Δt .

7.2 Betrachtung des Gradienten

Um ein Sinken beziehungsweise Steigen der FTLE-Werte besser untersuchen zu können liegt es nahe, den Gradienten des FTLE-Werteverlaufs zu betrachten. Zur Berechnung des Gradienten wird zuerst das Array mit den FTLE-Werten des selektierten Bereichs wie in Abschnitt 4.6.2 berechnet. Daraufhin folgt ein zweiter Durchlauf, in dem der Gradient mit der Methode der finiten Differenzen berechnet wird. Bei diesem Berechnungsmodus kann die Division durch Δt ebenfalls wie in Abschnitt 7.1.1 unterdrückt werden.

Der wing-Datensatz (siehe Abschnitt 8.1.4) wurde zur Veranschaulichung der Funktionsweise gewählt, da an der Vorder- und Hinterseite des Tragflächenprofils starke, aber kurze Schwankungen des FTLE-Wertes zu erwarten sind. Der Gradient der Kurven dieses Datensatzes kann in Abbildung 7.2(a) betrachtet werden. Wenn sich der FTLE-Wert nicht ändert beträgt dessen Gradient 0. Ein positiver Gradient wird durch einen ansteigenden FTLE-Wert verursacht, ein negativer Gradient deutet auf einen abfallenden FTLE-Wert hin. Änderungen treten in diesem Datensatz beim Erreichen bzw. Verlassen der Tragfläche

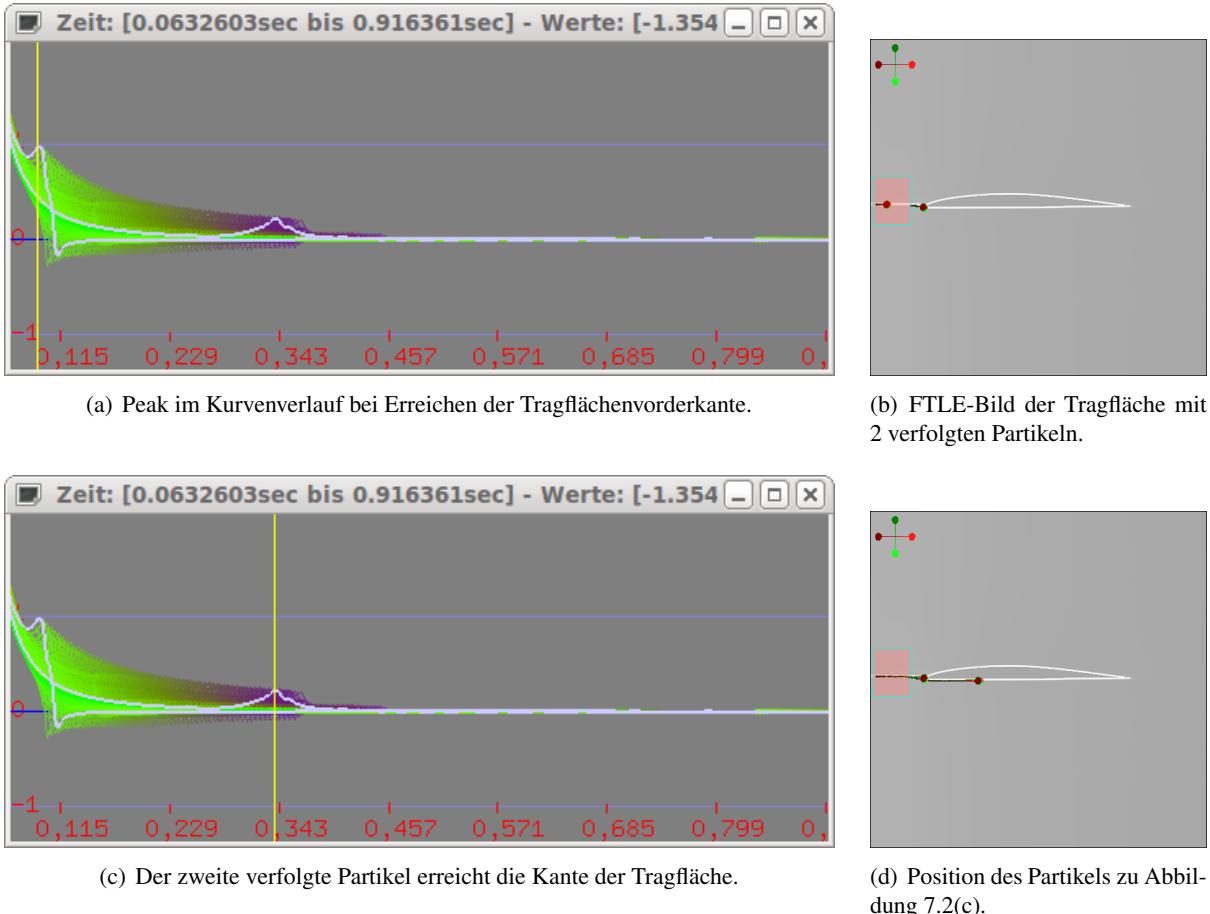
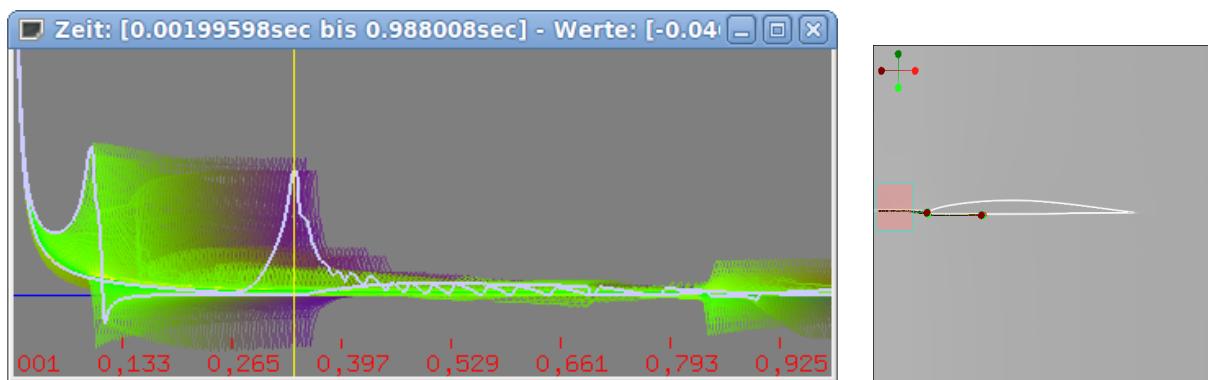


Abbildung 7.2: Betrachtung des Gradienten der Diagrammkurven.

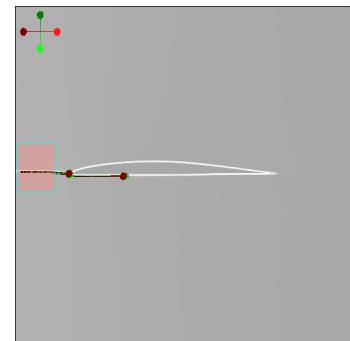
auf. Dieser Effekt ist in Abbildung 7.2 zu sehen. Auch wird sichtbar, dass auch die Werte des Gradienten mit steigendem Δt zurückgehen. Die Peaks beim Verlassen des Profils sind in dieser Zoom-Einstellung bereits nicht mehr zu sehen.

Möchte man auch auf diese Details nicht verzichten bietet es sich an, wie bereits in Kapitel 7.3 angewendet auf das Teilen durch Δt bei der Berechnung des FTLE-Wertes zu verzichten. Die Diagramme in Abbildung 7.3 zeigen diese Ergebnisse: Die Peaks bei $\Delta t = 0.3s$ sind fast genauso hoch wie die Peaks der Partikel, die die Tragfläche als erstes erreichen. Das nochmalige Ansteigen des Gradienten hinter $\Delta t = 0.8s$ entsteht beim Verlassen des Tragflächenprofils. Dies ist in Abbildung 7.3(c) anhand eines einzelnen Partikels sichtbar.

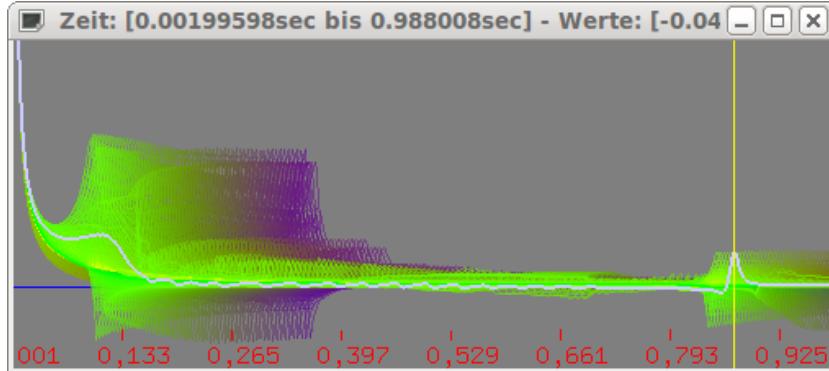
Beim Clustering spielt es ebenfalls eine Rolle, ob auf den Kurven der FTLE-Werte oder auf dessen Gradienten gearbeitet wird. Abbildung 7.4 stellt den Vergleich zwischen normalem Clustering und dem Clustering auf dem Gradienten dar. In beiden Fällen wurde das Teilen durch Δt unterdrückt, um auch die Peaks beim Verlassen des Tragflächenprofils ausreichend zu berücksichtigen. Als Zeitbereich wurde $\Delta t = 0.6s$ bis $\Delta t = 0.98s$ gewählt. Die Form und Lage der 5 Cluster unterscheidet sich maßgeblich: Beim normalen Clustering spielt der Absolutwert hier eine größere Rolle. Deshalb ist für die Zuordnung der Cluster der Zeitpunkt, zu dem ein Teilchen die Tragflächenkante erreicht der dominante Faktor. Beim Clustern auf den Gradienten hingegen ist die Stärke der Änderung zu betrachten. Diese ist abhängig von



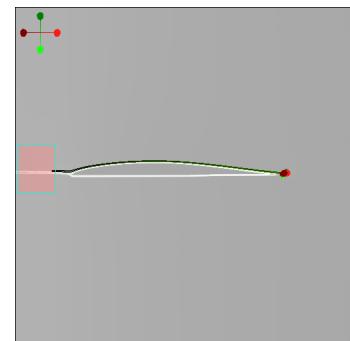
(a) Darstellung des Gradienten ohne Division durch Δt mit den gleichen Partikeln wie in Abbildung 7.2.



(b) Position des Partikels zum mit der gelben Markierung in Abbildung 7.3(a) markierten Zeitpunkt.



(c) Darstellung des Gradienten ohne Division durch Δt . Es wurde 1 Partikel verfolgt, bei dem der Peak beim Verlassen der Tragfläche sehr deutlich zu erkennen ist.



(d) Position des Partikels zum mit der gelben Markierung in Abbildung 7.3(a) markierten Zeitpunkt.

Abbildung 7.3: Betrachtung des Gradienten der Diagrammkurven.

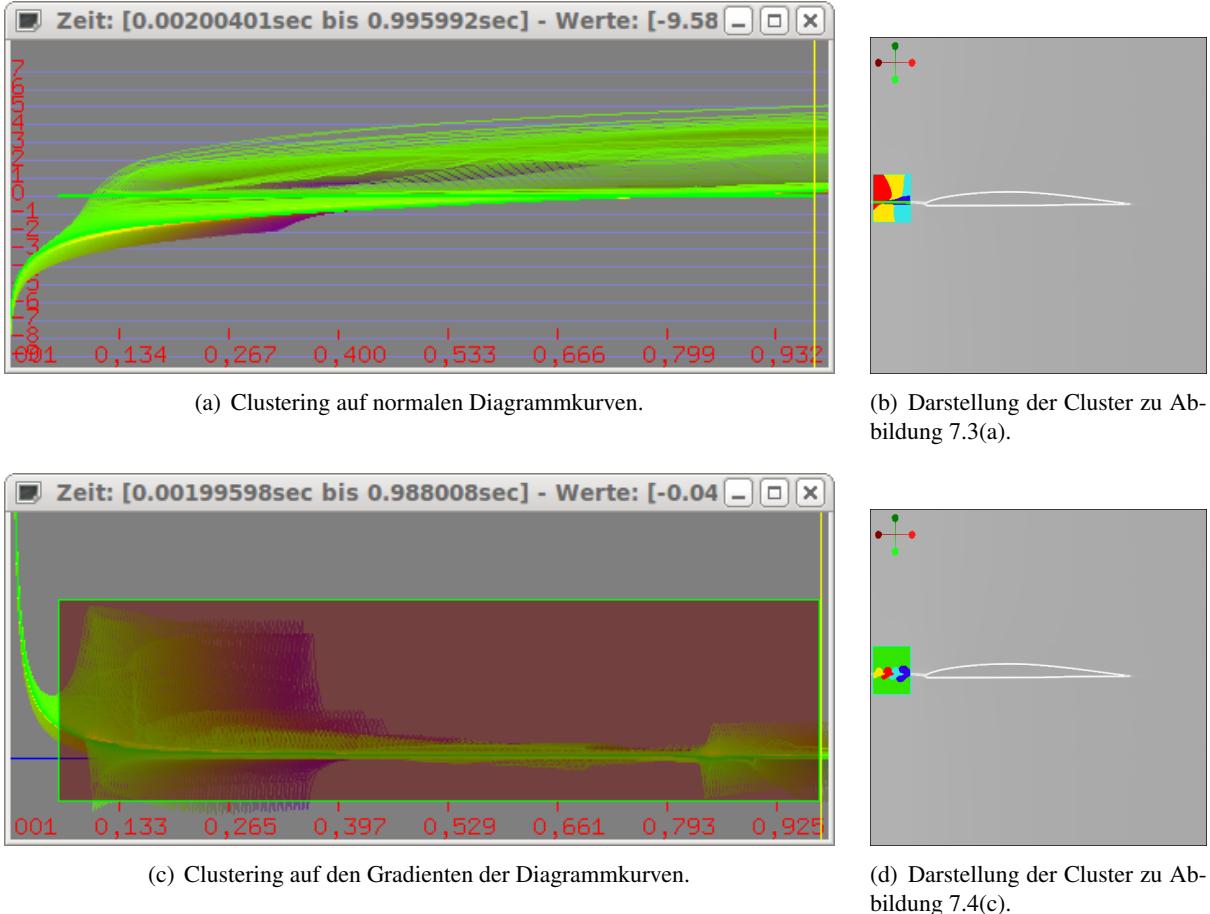
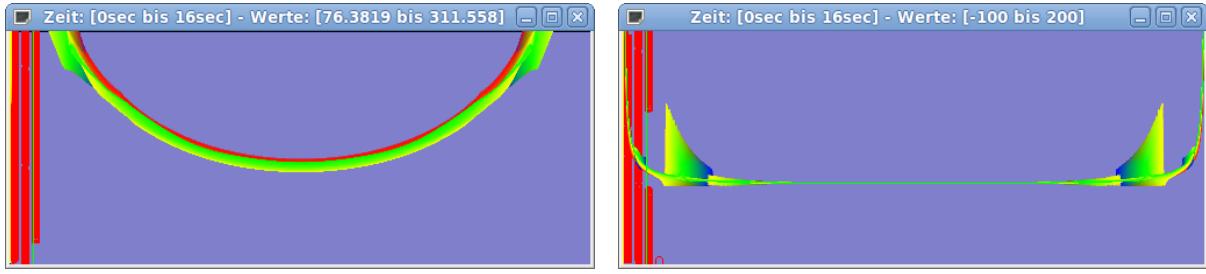


Abbildung 7.4: Clustering auf FTLE-Werten und deren Gradienten.

der Entfernung zur Tragfläche während des Vorbeiströmens. Teilchen mit großer Entfernung erfahren nur eine geringe Änderung der Separation und werden dem grünen Cluster zugeordnet. Die Teilchen die die Tragfläche mit geringerem Abstand umstreifen und somit eine größere Änderung des FTLE-Wertes erfahren, befinden sich in den übrigen 4 Clustern. Die Gebiete dieser Cluster unterschieden sich hauptsächlich durch den Zeitpunkt, wann das Erreichen der Tragflächenkante (und damit das Auftreten des Peaks) stattfand.

7.2.1 Fourier-Transformierte des Werteverlaufs

Auf der Suche nach wiederkehrenden Mustern und periodischen Effekten, wie sie z.B. in Wirbeln auftreten können ist die Fourier-Transformation zur Betrachtung des Frequenzspektrums ein mächtiges Werkzeug. Mit Hilfe der in der FFTW-Library [FJ09] enthaltenen Fast-Fourier-Transformation (FFT) werden die Spektren der einzelnen Kurven der FTLE-Daten berechnet. Diese Berechnung wird auf der CPU durchgeführt. Bei der Betrachtung der Diagramme ist zu beachten, dass der k -te Wert (von links nach rechts) der Frequenz k/N entspricht. N ist hierbei die gesamte Anzahl Samplingpunkte der Kurve. In der linken Hälfte des Diagramms sind die positiven Frequenzen dargestellt, in der rechten Hälfte



(a) Frequenzspektrum der FTLE-Wertekurve. (b) Frequenzspektrum der FTLE-Wertekurve ohne durch Δt .

Abbildung 7.5: Gradienten der in Abbildung 7.1 gewählten Region Kurven Δt .

die negativen Frequenzen. Dies bedeutet auch, dass in der Mitte des Diagramms die höchste Frequenz dargestellt wird und diese nach außen hin abfällt. Am linken bzw. rechten Rand des Diagramms ist der Gleichanteil eingezeichnet. Da die Ergebnisse in diesem Modus lediglich Qualitativ beurteilt werden sowie die Ergebnisse des Clusterings auf dem Frequenzspektrum untersucht wird wird auf die Bemaßung der Frequenz im Diagramm verzichtet.

Bei der Betrachtung des Frequenzspektrums muss stets beachtet werden, dass bei Frequenzen $|k| > \frac{1}{2}N$ Artefakte auftreten können. Näheres über das Sampling Theorem und dessen Einflüsse können [Fro07] entnommen werden. Auf die hier gewählte Diagrammdarstellung bezogen bedeutet dies, dass lediglich das linke bzw. rechte Viertel des Diagramms untersucht werden darf. Die roten Streifen am linken Rand der Diagramme von Abbildung 7.5 sind die Beschriftungen der y-Achsen, die sich aufgrund des großen Wertebereichs der Fouriertransformierten überschneiden.

7.3 Clustering

Als Verfahren für das Clustering wird k-means eingesetzt. Für den eindimensionalen Fall wird das Vorgehen in [JMF99] erläutert. Dies kann jedoch sehr einfach auf den zweidimensionalen Fall übertragen werden: Bei diesem Verfahren werden zuerst zufällige Kurven als Repräsentant eines Clusters gesetzt. Im zweiten Schritt wird jede Kurven des FTLE-Diagramms dem Cluster zugeordnet, zu dem sie den jeweils geringsten Abstand haben. Im nächsten Schritt werden Repräsentanten des Clusters aktualisiert: Jeder Stützpunkt des Repräsentanten zum Zeitpunkt t ist der Mittelwert der Stützpunkte zum Zeitpunkt t aller in diesem Cluster enthaltenen Kurven. Das Zuordnen der Kurven und Aktualisieren der Repräsentanten wird so lange wiederholt, bis keine Veränderungen in der Clusterzuordnung mehr stattfinden. Als Metrik zur Bestimmung des Abstandes zwischen Clusterrepräsentanten und der zu vergleichenden Kurve wird hier die Summe des quadratischen Abstands der Werte verwendet.

$$d(v_{centroid}, v_{curve}) = \sum_t (v_{centroid}(t)) - v_{curve}(t))^2 \quad (7.4)$$

Eine Eigenschaft des k-Means Clustering ist, dass je nach (zufällig gewählten) Startwerten andere Cluster-Zuordnungen auftreten können. In der Praxis hat sich jedoch gezeigt, dass beim Wiederholen des Clusterings lediglich die Reihenfolge (IDs) der Cluster variiert, die Inhalte der Cluster jedoch bis auf wenige Ausnahmen gleich bleiben.

Der Clustervorgang kann über die Anzahl Cluster k gesteuert werden. Ebenfalls hat die Wahl des

Zeitbereiches, über den die Werte verglichen werden einen Einfluss auf das Ergebnis.

Die Auswahl des Zeitbereichs wird mit Hilfe des Datensatzes „bottleneck assymetric“ (siehe Abschnitt 8.1.4) und 4 Clustern untersucht. Dieser Datensatz bietet durch die Engstelle ein starkes Schwanken des FTLE-Wertes über die Zeit. Trotzdem ist das Verhalten von Teilchen in diesem Datensatz sehr überschaubar, es gibt keine nennenswerten Verwirbelungen. Durch die dominante Strömungsrichtung von links nach rechts ist auch die Überprüfung, ob Teilchen bereits das Betrachtungsfeld verlassen haben einfach möglich. Clustering wird über einen Zeitbereich durchgeführt die Berechnung des FTLE-Bildes hingegen visualisiert lediglich die Separation zum Zeitpunkt $T0 + \Delta t$. Es können sich unterschiedliche Strukturen ausbilden, die nicht zwingend vergleichbar sind. Bei der ersten Gruppe in Abbildung 7.6 werden für das Clustering lediglich die FTLE-Werte zwischen $\Delta t = 0s$ und $\Delta t = 0.3s$ betrachtet. Hierzu gehört das Ansteigen der Werte zu Beginn der Betrachtung sowie ein Abfallen, nachdem die Teilchen die Engstelle passiert haben. Zum Zeitpunkt $\Delta t = 0.3s$ haben bereits alle Partikel, die sich innerhalb des im FTLE-Bild selektierten Bereichs befinden die Engstelle durchlaufen. Hierbei spielt bei der Zuordnung der Cluster hauptsächlich der Zeitpunkt wann die Teilchen die Engstelle passieren eine Rolle. Während der Partikel die Engstelle durchläuft herrscht dort eine sehr hohe Geschwindigkeit und damit verbunden eine hohe Separation. Diese äußert sich in einem kurzen, aber starkem Anstieg des FTLE-Wertes und fällt sofort hinter dem Engpass wieder steil ab. In Abbildung 7.7(a) wurden diese Peaks mit roten Kreisen markiert. Da der Zeitpunkt, an dem die Engstelle durchschritten wird, direkt von der Entfernung zu dieser abhängt, bilden sich die Cluster ringförmig um die Engstelle herum aus. Beim zweiten gewählten Zeitbereich, bei dem die Kurven nur zwischen $\Delta t = 0.3s$ und $\Delta t = 0.6s$ für das Clustering berücksichtigt werden, weicht die Form der Cluster stark von der zuletzt beschriebenen ab. Am Anfang des betrachteten Bereiches passieren die in der Selektion des FTLE-Bildes ganz links liegenden Partikel noch den Engpass. Deswegen ist die linke Grenze zwischen rotem und grünem Cluster in Abbildung 7.6(d) noch in Form einer Ellipse um den Engpass ausgebildet. Bis $\Delta t = 0.6s$ entfernen sich die Partikel immer weiter von der Engstelle, wobei die Änderung der Separation und somit die Änderung des FTLE-Wertes immer geringer wird. Am Ende des Betrachtungsbereiches mit $\Delta t = 0.6s$ hat jedoch noch kein Partikel den rechten Rand des Datensatzes überschritten.

In Abbildung 7.6(f) und dem zugehörigen Diagramm 7.6(e) erreichen hingegen Partikel aus allen Clustern bereits den rechten Rand. Überwiegend die auf der y-Achse in der Mitte liegenden Teilchen überschreiten die Datensatzgrenze. Teilchen die den Rand erreichen sind in jedem Cluster enthalten, keines der Cluster enthält ausschließlich Teilchen die am Rand ankommen. Da Teilchen, die die Datensatzgrenze überschreiten dort „eingefroren“ und nicht mehr weiter advektiert werden, stellt sich dort eine Art stationärer Zustand ein: der Wert beim Verlassen des Feldes wird somit maßgeblicher Faktor zur Auswahl des Clusters. Die Höhe dieses Wertes wird in diesem Datensatz hauptsächlich durch die y-Position mit der der Partikel die Engstelle passiert bestimmt. Auch wenn die Form der Cluster symmetrisch und gleichmäßig erscheint ist ein praktischer Nutzen der Analyse von Kurven, deren Partikel zu unterschiedlichen Zeiten das Feld verlässt zweifelhaft oder zumindest mit Vorsicht zu betrachten. Jedoch steht außer Frage, dass die Auswahl des für das Clustering zu betrachtenden Zeitbereiches wesentlichen Einfluss auf das Ergebnis und die Aussage des Clusterings hat.

Es muss ebenfalls untersucht werden, welche Rolle die Länge des Zeitbereichs hat, auf dem geclustert wird. Hierzu wird der Startzeitpunkt fest auf $\Delta t = 0s$ gelegt und lediglich der Endzeitpunkt variiert. Clustering auf nur einem Slice zu Beginn des Datensatzes liefert das in Abbildung 7.8 gezeigte Resultat. In Abbildung 7.8(a) ist anzumerken, dass zur Zuordnung der einzelnen Kurven zum jeweiligen Cluster nur der im Diagramm selektierte Zeitbereich betrachtet wird - das Berechnen der Repräsentanten erfolgt hingegen der Anschaulichkeit wegen über den kompletten Zeitbereich. Ansonsten wären lediglich 4

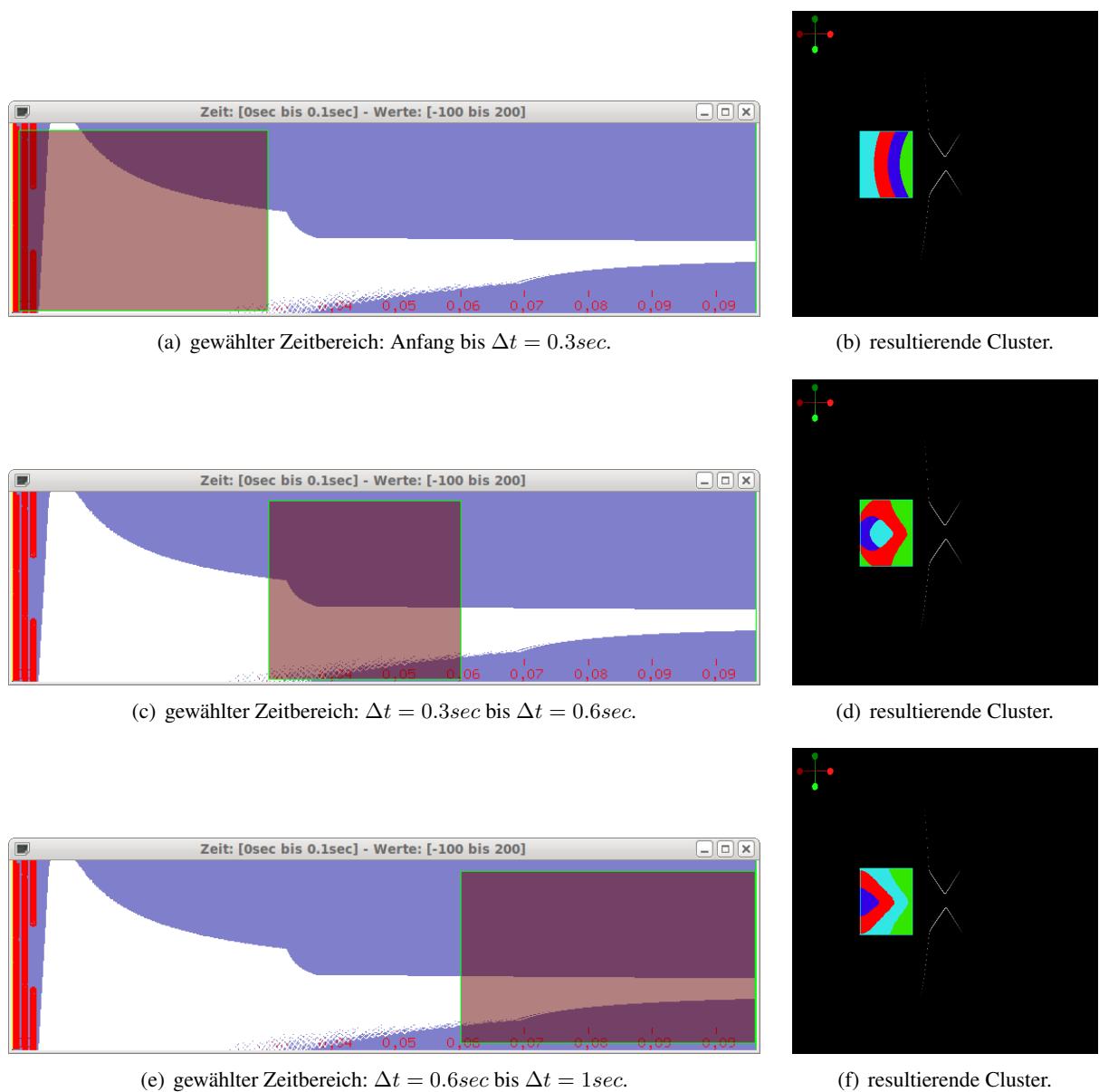
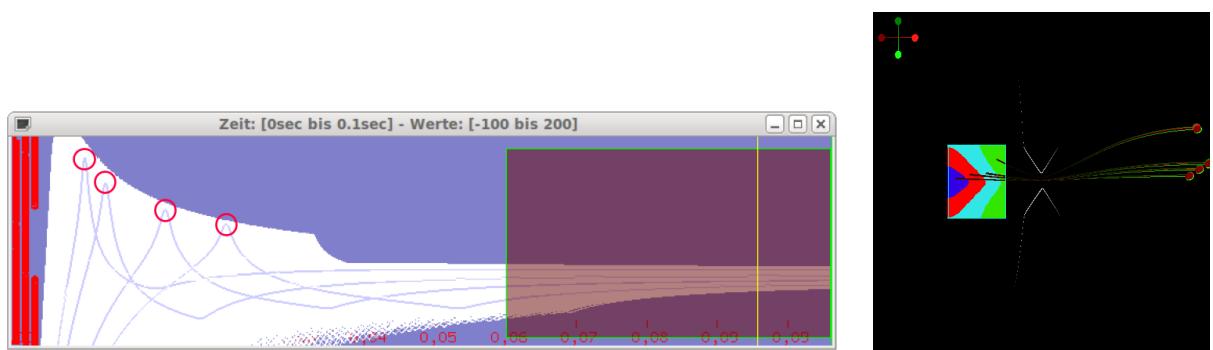


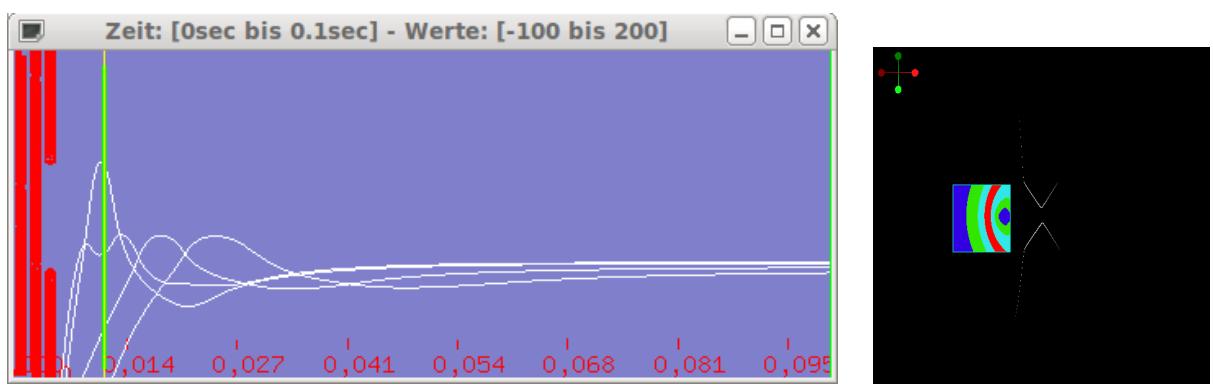
Abbildung 7.6: Diagramm-Darstellung des FTLE-Werteverlaufs.



(a) Zeitpunkt der in Abbildung 7.7(b) dargestellten Partikel (gelbe Linie). Die Peaks des FTLE-Wertes beim Durchströmen der Engstelle wurden mit roten Kreisen markiert.

(b) in unterschiedlichen Clustern gestartete Partikel.

Abbildung 7.7: Tracen von Partikeln in den Clustern aus Abbildung 7.6(e) bzw. 7.6(f).



(a) Clustering auf nur 1 Slice: Dargestellt sind lediglich die Repräsentanten der 4 Cluster.

(b) Anordnung der Cluster.

Abbildung 7.8: Variation des Zeitbereichs, auf dem geclustert wird.

Punkte statt der 4 Kurven darstellbar. Auf die Zuordnung der Kurven zu ihren Clustern hat diese zusätzliche Berechnung jedoch keinen Einfluss. Auffällig ist hierbei, dass die Cluster hierbei nicht zusammenhängen. Der blaue Cluster in Abbildung 7.8(b) tritt ganz links sowie ganz rechts auf, darauf folgen von außen nach innen der grüne und der türkisfarbene Cluster. Lediglich der rote Cluster in der Mitte tritt nur einmal auf. Den Grund für dieses Verhalten liefert eine Betrachtung von Partikeln, die in den jeweiligen Feldern gestartet werden. In Abbildung 7.9 wurden in mehreren Clustern Partikel gestartet. Im Diagramm sind die Cluster-Repräsentanten (weiß) sowie die Kurven der verfolgten Partikel (hellgrau, breitere Linien) eingezeichnet. Zu erkennen ist, dass die Partikel des linken blauen Clusters die Engstelle noch nicht passiert haben, die des rechten blauen Clusters befinden sich hingegen bereits dahinter. Partikel des grünen Clusters sind ebenfalls noch vor der Engstelle (linkes grünes Cluster) bzw. dahinter (rechtes grünes Cluster), jedoch deutlich näher am Cluster. Nur die Partikel, die im roten Clusters gestartet wurden befinden sich zum Zeitpunkt des für das Clustering berücksichtigten Slices direkt in der Engstelle. Da beim Clustering auf einem einzelnen Zeitschritt lediglich die Momentanwerte berücksichtigt werden und nicht der Verlauf des Wertes fallen hier die Partikel mit gleichem FTLE-Wert in ein Cluster, unabhängig davon ob dieser vor- oder nach dem beim Passieren der Engstelle auftretenden Peak auftritt.

Wird hingegen über einen längeren Zeitraum geclustert, so fällt der zeitliche Verlauf stärker ins Gewicht. Abbildung 7.10(b) zeigt die Cluster, die unter Berücksichtigung des Kurvenverlaufs zwischen $\Delta t = 0.3s$ bis $\Delta t = 0.8s$ erstellt wurden. Durch Setzen von Testpartikeln wurde überprüft, dass auch am Ende der Betrachtungszeit noch keine Teilchen den rechten Bereich des Datensatzes verlassen haben. Wie zu erwarten spielt es bei der Betrachtung über diesen längeren Zeitraum nun eine Rolle, zu welchem Zeitpunkt das Teilchen die Engstelle passiert - nicht zusammenhängende Cluster treten nicht mehr auf. Ebenfalls verändert der Verzicht auf die Division durch Δt bei der Berechnung des FTLE-Wertes die Resultate. Da alle Δt bei diesem Zeitfenster kleiner als 1 sind sinken die Werte deutlich, weshalb die Zoomeinstellung im Diagramm angepasst wurde. Durch das Auslassen dieser Division, die die Dämpfung der Werte mit wachsendem Δt reduziert, wächst hierbei der Einfluss der Werte gegen Ende des Betrachtungszeitraumes. Die hierdurch resultierende Aufteilung der Cluster zeigt Abbildung 7.10(d).

7.4 Visuell vereinfachte Untersuchung von Symmetrien durch Clustering

Das Auffinden von Symmetrien sowie von unsymmetrischen Bereichen wird anhand des Datensatzes „bottleneck“ (siehe Abschnitt 8.1.4) durchgeführt. Hierzu wird mit Hilfe von Matlab ein modifizierter Datensatz erzeugt, bei dem der untere Teil des Hindernisses nach rechts verschoben ist (Abbildung 7.11(b)). Der Versatz beträgt gerade einmal 10% des Abstands zwischen oberer und unterer Hindernisspitze und ist im FTLE-Bild mit bloßen Auge nur schwer erkennbar.

In Abbildung 7.12 wird das Bild mittels Clustering untersucht. Der für das Clustering verwendete Zeitbereich geht über die komplette Datensatzlänge von $\Delta t = 0sec$ bis $\Delta t = 0.1sec$, da gegen Ende des Zeitbereiches keine extremen Schwankungen auftraten. Bei einem Clustering mit 4 Clustern (Abbildung 7.12(b)) sind an den Grenzen des rot gefärbten Bereichs bereits Auswirkungen der Asymmetrie erkennbar: der türkisfarbige Einschnitt ist im unteren Bereich deutlich größer als im oberen. Bei einer Erhöhung der Clusterzahl auf 10 werden wie in Abbildung 7.13(b) ersichtlich noch mehr Details deutlich: der Rand zwischen hellgrünem und gelbem Cluster hat einen unterschiedlich großen Einschnitt. Ebenfalls ist der rechte, rosafarbige Cluster etwas unterhalb der Mitte stark eingeschnitten.

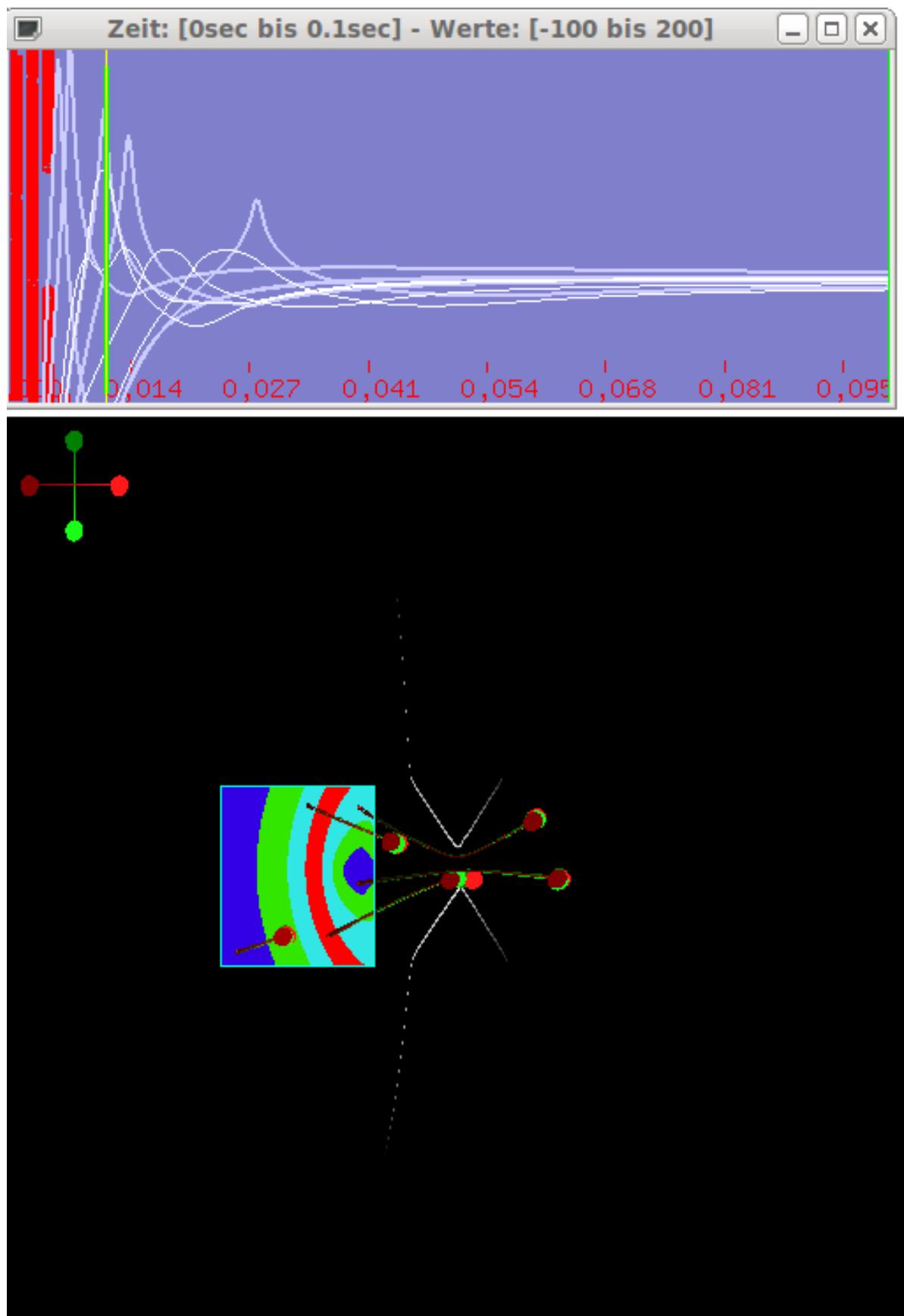
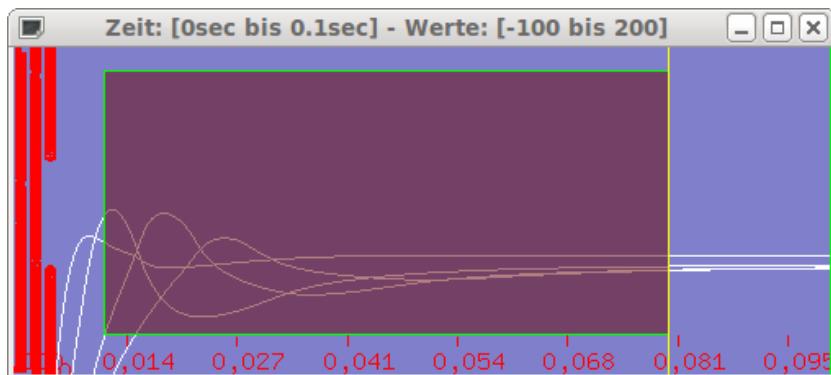
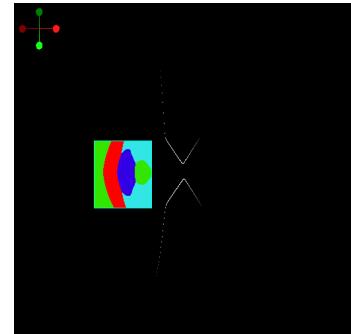


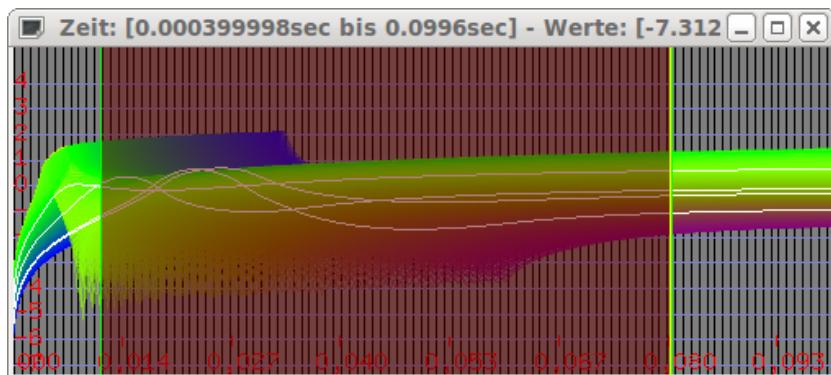
Abbildung 7.9: Untersuchung der Clusteraufteilung durch Verfolgung von Partikeln.



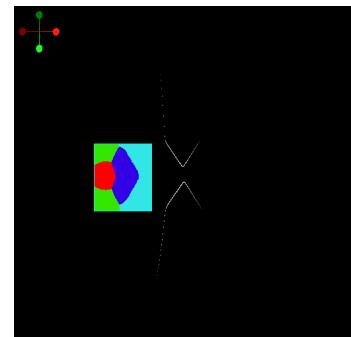
(a) Kurven der 4 Cluster-Repräsentanten. Um diese besser erkennen zu können wurden die übrigen Kurven ausgeblendet.



(b) Resultierende Cluster zu Abbildung 7.10(a).

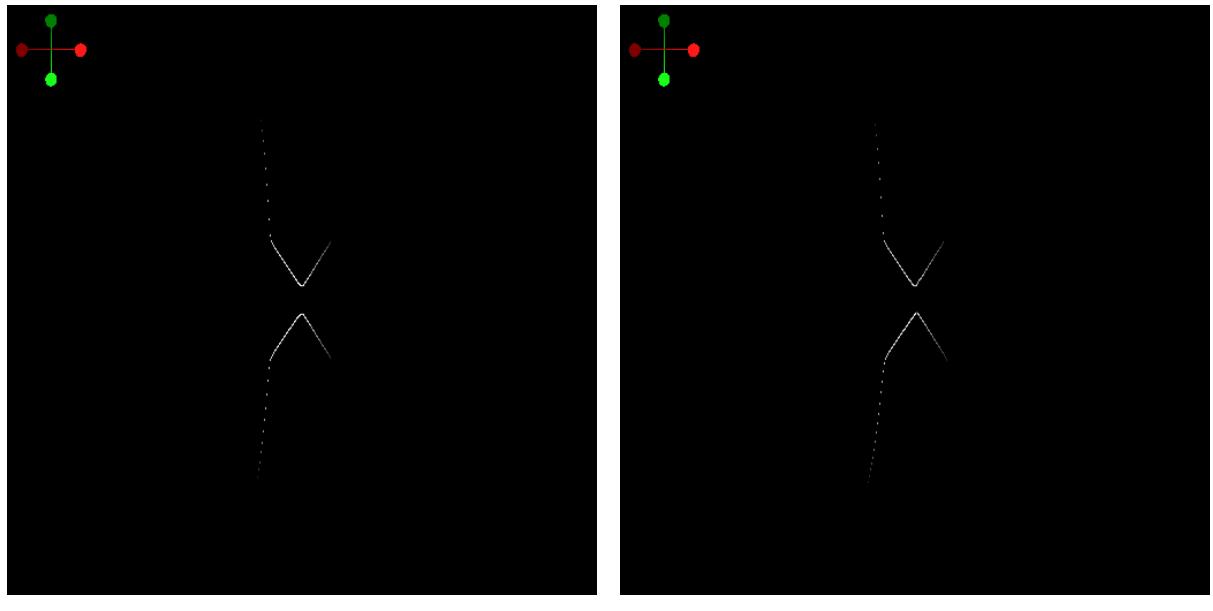


(c) Diagramm der FTLE-Werte und der 4 Cluster-Repräsentanten. Hier wurde die Division des FTLE-Wertes durch Δt unterdrückt.



(d) Resultierende Cluster zu Abbildung 7.10(c).

Abbildung 7.10: Durchführung des Clusterings auf einem größeren Intervall von $\Delta t = 0.3s$ bis $\Delta t = 0.8s$.

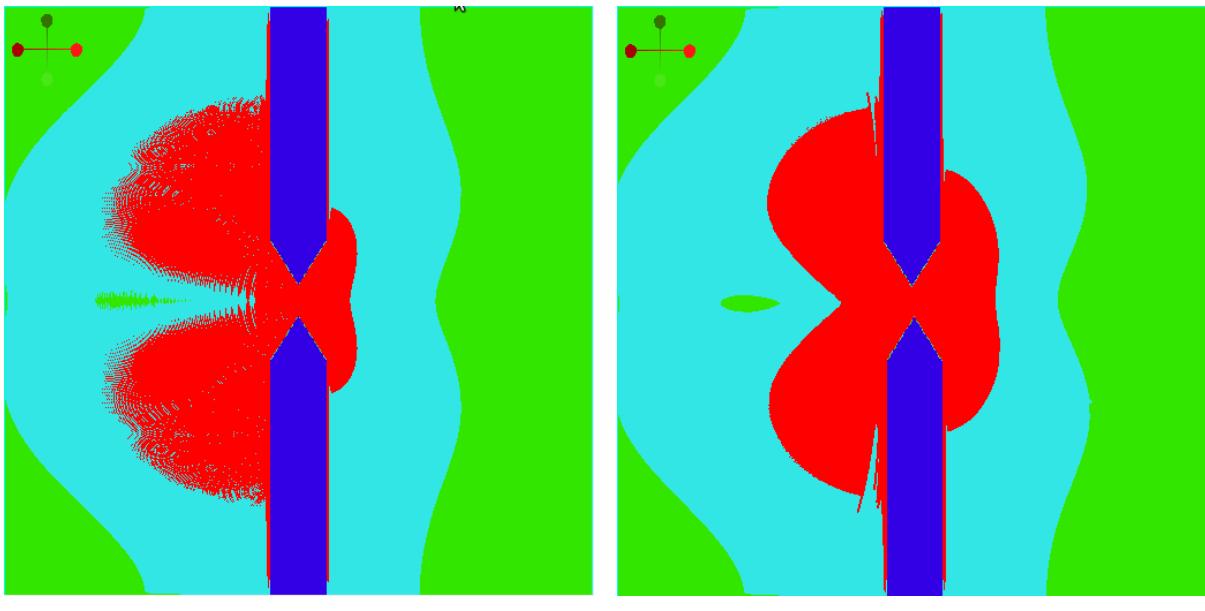


(a) Symmetrische Form des Hindernisses im Datensatz „bottleneck1“. (b) modifizierter Datensatz mit assymetrischem Hindernis.

Abbildung 7.11: Vergleich des FTLE-Bildes zwischen originalem und modifiziertem Datensatz.

Beim symmetrischen Bild (Abbildung 7.12(a)) hingegen sind die Cluster absolut symmetrisch ausgebildet - sogar die Ausreißer des rechten, gelben Clusters sind oben und unten gleich weit ausgebildet. Auffällig ist auch, dass die Cluster nahe der Engstelle beim assymetrischen Bild deutlich schärfer abgegrenzt sind als beim symmetrischen, obwohl der Zeitbereich sowie die Schrittweite des Runge-Kutta-Integrators unverändert blieben. Dieses Verhalten blieb auch bei Verkürzung des Zeitbereichs und mehrmaligem Neusetzen der (zufällig gewählten) Startwerte des K-Means Clusterings bestehen. Ein Vergrößern der Schrittweite des RK4-Operators verstärkt den Effekt. Es ist also anzunehmen, dass beim symmetrischen Datensatz die bei der Integration entstehenden Fehler stärker ins Gewicht fallen als beim assymetrischen.

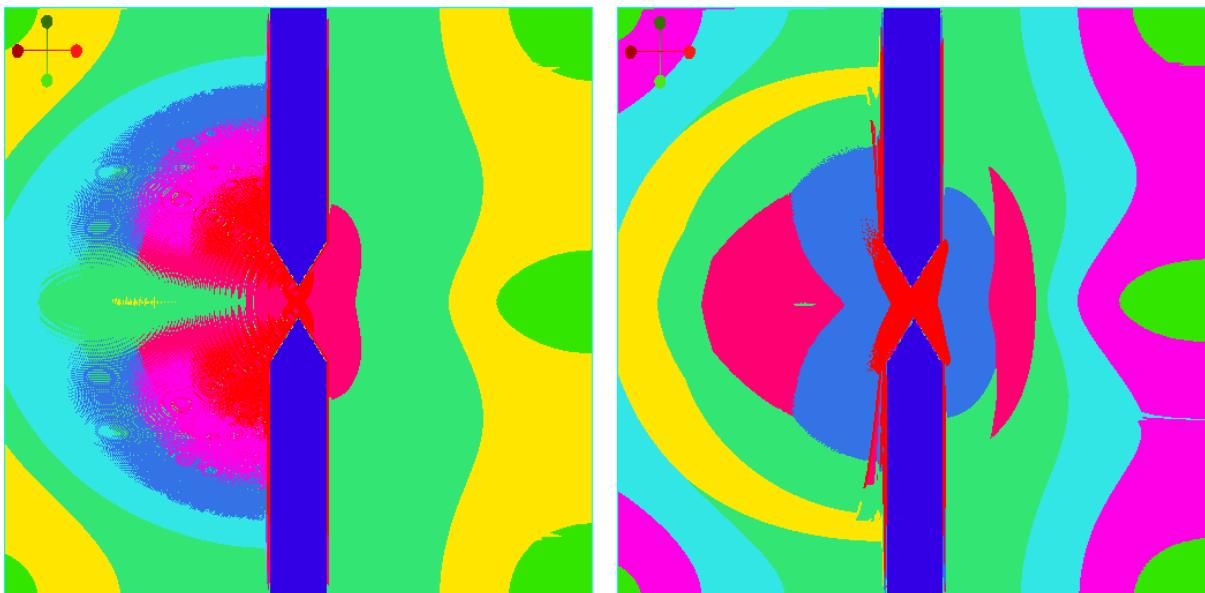
Der Vorteil dieses Ansatzes zur Erkennung von assymetrischen Bereichen ist, dass nicht nur die Separation zu einem konkreten Zeitpunkt betrachtet wird, sondern die Veränderung der FTLE-Werte über einen größeren Zeitraum eine Rolle spielt - und sich kleinere Unterschiede zu einer größeren Wirkung aufsummieren können. Auch wird hierbei die volle Bildfläche zur Darstellung von Information ausgenutzt: gerade in den bei diesem Datensatz sonst einheitlich schwarz erscheinenden Bereichen erscheinen die für den Benutzer wichtigsten Details in Form von ausgeprägten Clustergrenzen.



(a) Symmetrische Form des Hindernisses im Datensatz „bottleneck1“.

(b) modifizierter Datensatz mit assymetrischem Hindernis.

Abbildung 7.12: 4 Cluster.



(a) Symmetrische Form des Hindernisses im Datensatz „bottleneck1“.

(b) modifizierter Datensatz mit assymetrischem Hindernis.

Abbildung 7.13: 10 Cluster.

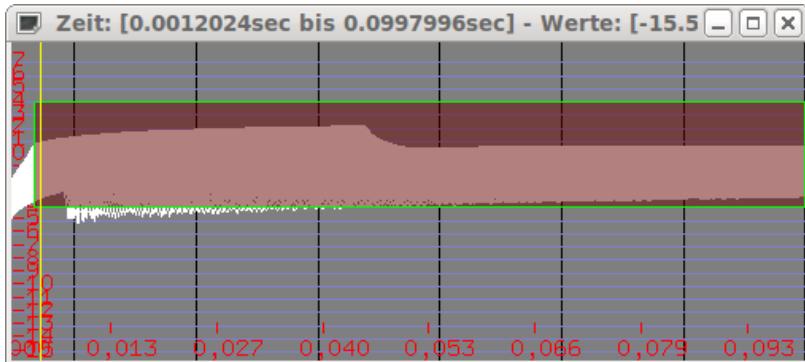
7.5 Kriterium: FTLE-Wert fällt über ein gewisses Intervall unter Max

Beim Betrachten sinkender Werte ist es notwendig, die Division durch Δt zu deaktivieren, um nicht fälschlicherweise das Abfallen der Werte durch die Division zu detektieren. Der „bottleneck“-Datensatz eignet sich zur Untersuchung hervorragend, da die Werte bei Annäherung an den Engpass steigen und dahinter rasch abfallen. Der Startzeitpunkt für die Anwendung des Kriteriums wurde auf $\Delta t = 0.004s$ gelegt. Zu diesem Zeitpunkt ist der initiale Anstieg der FTLE-Werte bereits abgeschlossen, jedoch hat noch kein Teilchen den Engpass erreicht. In Abbildung 7.14 wurde von Abbildung 7.14(a) bis Abbildung 7.14(d) ein Interval von 10 slices gewählt. Das Diagramm besteht hierbei aus 500 slices, wodurch das Interval eine Länge von 0.002 Sekunden hat. Bei Abbildung 7.14(e) und 7.14(h) wurde das Interval auf 70 slices (0.014 Sekunden) verlängert. Dies bedeutet, dass der niedrige Wert über eine längere Zeit gehalten werden muss, wodurch Werte die zu spät abfallen nicht mehr berücksichtigt werden. Auffällig ist, dass dies erst Einfluss auf den Selektionsbereich in der Nähe der Engstelle hat, der linke Bereich der Selektion bleibt unverändert gegenüber 7.14(b). Grund ist, dass die Teilchen, die sich zum Startzeitpunkt bereits nahe an der Engstelle befinden zum Ende der Betrachtungszeit bereits den Datensatz auf der rechten Seite verlassen haben und deren Position dort eingefroren wird. Dieser Effekt ist in den meisten Fällen nicht erwünscht, muss jedoch berücksichtigt werden und ist hier sehr deutlich zu erkennen. Detailliert wird diese Problematik in Kapitel 7.7 behandelt.

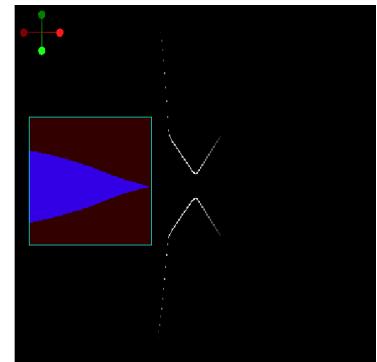
Weiterhin besteht die Möglichkeit, den mindest-Abfall des FTLE-Wertes nicht absolut sondern in Prozent anzugeben. Dies ist sinnvoll, wenn Kurven die mit sehr großer Streuung gleichzeitig auf fallende Werte untersucht werden sollen. Kurven mit hohem Maximum müssen verhältnismäßig stärker abfallen um dem Kriterium zu entsprechen als Diagrammkurven mit geringerem Maximum. Zu sehen ist diese veränderte Form des selektierten Bereichs in Abbildung 7.14(h): Werte mit geringerem Maximum und geringerem Abfall des Wertes befinden sich vorwiegend oberhalb bzw. unterhalb des Selektionsgebietes. Dort erweitert sich das Gebiet, das dem Kriterium entspricht, und wurde ebenfalls blau gefärbt.

7.6 Kriterium: FTLE-Wert steigt über ein gewisses Intervall über Min

Eine andere Auswahlmöglichkeit ist das Aufspüren von Werten, die über ein festgelegtes Intervall über dem Minimum des FTLE-Wertes innerhalb der Selektion liegen. Abbildung 7.15 zeigt die Anwendungen des Kriteriums. Wird die Höhe des Wertes, um die das Minimum überschritten wird vergrößert, so entsprechen weniger Diagrammkurven diesem Kriterium. Der blau markierte Bereich der FTLE - Werteverläufe die diesem Kriterium entspricht befindet sich in desto größerer Entfernung, je höher die Schwelle angesetzt wird. Die Ursache liegt darin, dass die FTLE-Werte mit größerer Entfernung zu Beginn einen niedrigeren Wert aufweisen und somit ein geringeres Minimum besitzen. Es fällt jedoch auf, dass sich in Abbildung 7.15(a) eine Art Zunge in der Nähe der Engstelle ausbildet. In Abbildung 7.16(b) wird die Trajektorie eines Partikels an dieser Position verfolgt. Hierbei wird deutlich, dass die FTLE-Werte in dieser Region kurz hinter der Engstelle sehr stark abfallen und dadurch das Überschreiten des hierbei gesetzten Minimums erleichtert ist. Die gelbe Zeitmarkierung in Diagramm 7.16(b) markiert den Zeitpunkt, zu dem das in der Zunge gestartete Teilchen direkt in der Engstelle steht (Peak des FTLE-Wertes), um die Partikel zuordnen zu können. Dem Vergleich zwischen Abbildung 7.15(b) und Abbildung 7.16(b) kann man entnehmen, dass beim Vergrößern des zu erfüllenden Intervalls von 10 auf 30 slices Werte mit großer Entfernung zur Engstelle ausgefiltert werden.



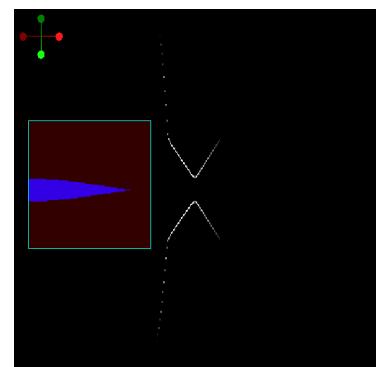
(a) Dargestellt sind Kurven, die über 10 Slices um 3 Einheiten unter dem Maximalen FTLE-Wert liegen.



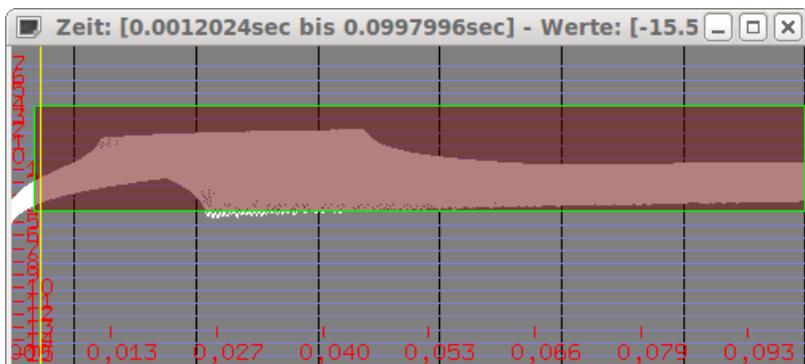
(b) Detekterter Bereich im FTLE-Bild zu Abbildung 7.14(a).



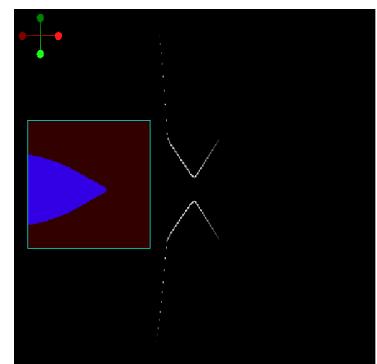
(c) Dargestellt sind Kurven, die über 10 Slices um 4 Einheiten unter dem Maximalen FTLE-Wert liegen.



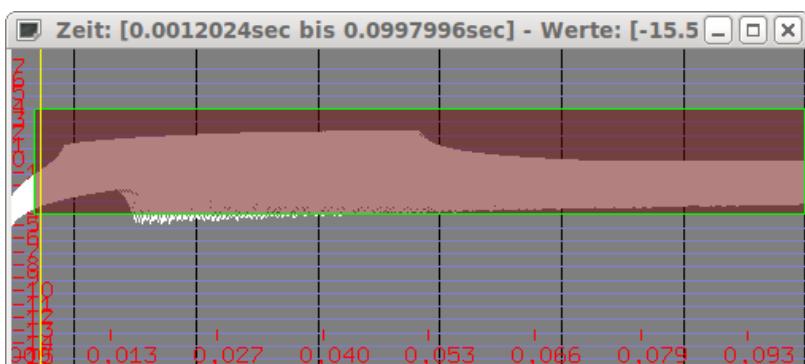
(d) Detekterter Bereich im FTLE-Bild zu Abbildung 7.14(c).



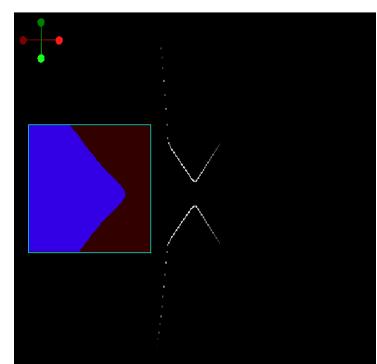
(e) Dargestellt sind Kurven, die über 70 Slices um 3 Einheiten unter dem Maximalen FTLE-Wert liegen.



(f) Detekterter Bereich im FTLE-Bild zu Abbildung 7.14(e).

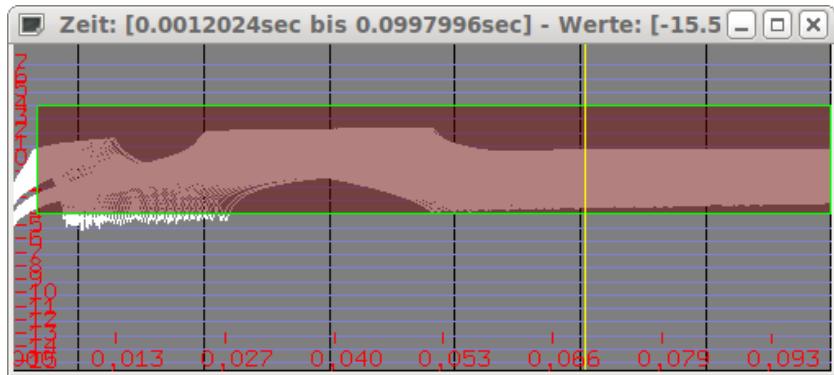


(g) Dargestellt sind Kurven, die über 70 Slices um 400% unter dem Maximalen FTLE-Wert liegen.

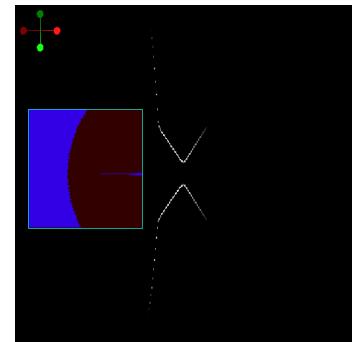


(h) Detekterter Bereich im FTLE-Bild zu Abbildung 7.14(g).

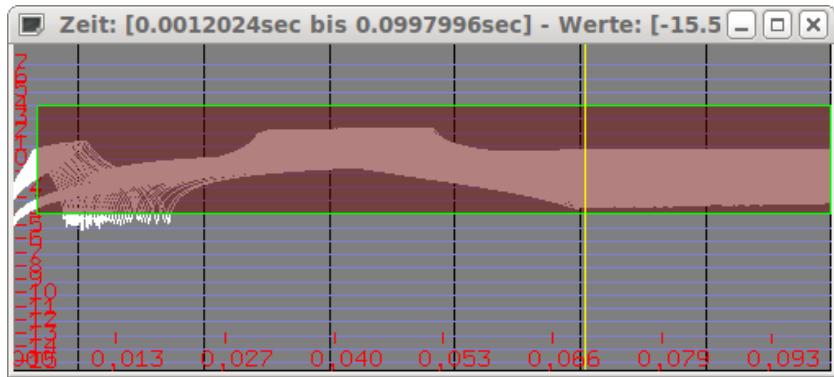
Abbildung 7.14: Detektion fallender FTLE-Werte.



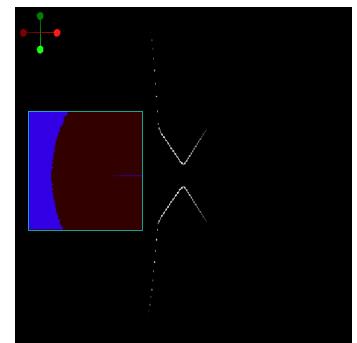
(a) Die selektierten Kurven liegen über ein Intervall von 10 slices um mindestens 4,0 Einheiten über dem Minimum.



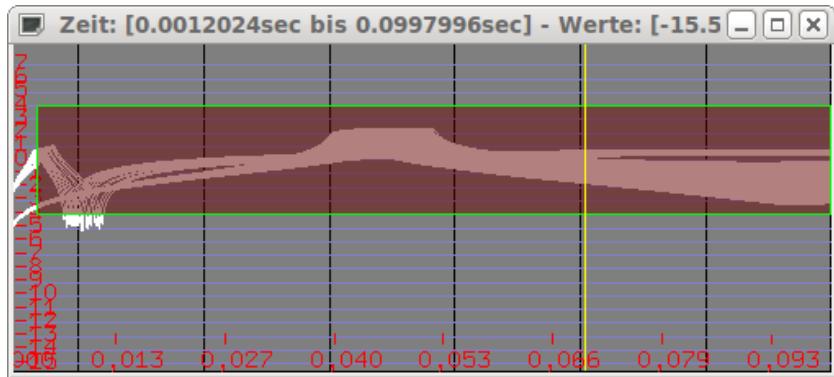
(b) Selektierter Bereich im FTLE-Bild zu Abbildung 7.15(a).



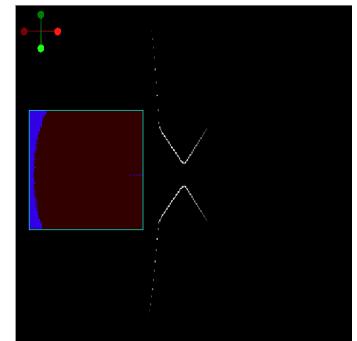
(c) Die selektierten Kurven liegen über ein Intervall von 10 slices um mindestens 4,5 Einheiten über dem Minimum.



(d) Selektierter Bereich im FTLE-Bild zu Abbildung 7.15(c).

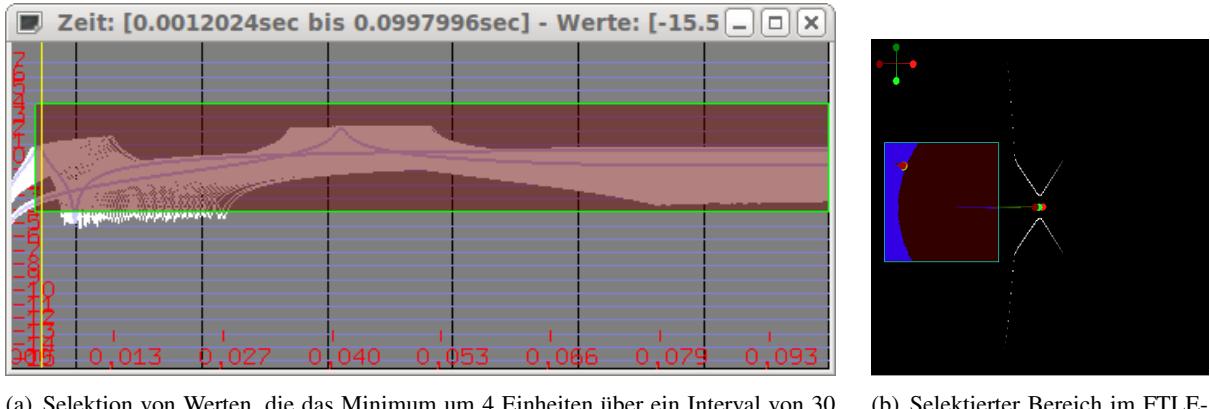


(e) Die selektierten Kurven liegen über ein Intervall von 10 slices um mindestens 5,0 Einheiten über dem Minimum.



(f) Selektierter Bereich im FTLE-Bild zu Abbildung 7.15(e).

Abbildung 7.15: Betrachtung von Gebieten, deren FTLE-Wert über ein Intervall von 10 slices (0.002 Sekunden) um den vorgegebenen Betrag über dem Minimum der jeweiligen Kurve liegt.



(a) Selektion von Werten, die das Minimum um 4 Einheiten über ein Intervall von 30 slices überschreiten.
 (b) Selektierter Bereich im FTLE-Bild zu Abbildung 7.16(a).

Abbildung 7.16: Wie Abbildung 7.15, jedoch Vergrößerung des zu erfüllenden Intervalls auf 30 slices.

7.7 Schwierigkeit: Überschreiten der Datensatzgrenzen

Vorsicht ist auch bei der Wahl des richtigen Zeitbereichs im Bezug auf den zu untersuchenden FTLE-Bildbereich: Falls der Zeitbereich zum Ende hin zu groß gewählt wird können Teilchen das Feld bereits verlassen. Anschaulich gezeigt werden kann dies am Tragflächen-Datensatz: In Abbildung 7.17 ist das FTLE-Bild der Tragfläche gezeigt. Das Gebiet wurde so gewählt, dass auch ein an der rechten Kante des selektierten Bereichs liegendes Partikel innerhalb des entsprechend Abbildung 7.18 gewählten Zeitbereichs noch nicht die Grenzen des Datensatzes verlässt. Somit ist sichergestellt, dass der nahe der verfolgten Partikelgruppe blau markierte Bereich nicht fälschlicherweise aufgrund von FTLE-Werteschwankungen, die beim Verlassen der Datensatzgrenze entstehen, selektiert wurde. Der FTLE-Wert fällt beim Passieren der Hinterkante des Flügels kurzzeitig ab: hier treffen die Partikel oberhalb der Tragfläche mit den Partikeln, die unterhalb des Flügels vorbei advektiert wurden wieder zusammen - und die Partikelgruppe wurde in der vertikalen Achse etwas näher zusammengedrückt.

Für Abbildung 7.19 wurde hingegen ein deutlich größerer Zeitbereich untersucht - der Plot der entsprechenden Kurven befindet sich in Abbildung 7.20. Man beachte, dass die Endposition der verfolgten Partikel (gelber Balken im Diagramm) innerhalb der Selektion liegt, die Teilchen aber bereits auf der rechten Seite den Bildbereich verlassen haben. Es lässt sich dadurch nicht mehr klar erkennen, ob der Abfall des FTLE-Wertes durch das Verlassen des definierten Datensatzbereiches oder beim Vorbeiströmen an der Hinterkante der Tragflächenkante verursacht wurde. Da hier alle Teilchen fast gleichzeitig das Feld verlassen, ist die Schwankung des FTLE-Wertes im Diagramm nicht sichtbar.

7.7.1 Lösungsansätze

Ein vom Benutzer durchführbares Mittel zur Plausibilitätsprüfung ist das Setzen von Partikeltracern an den Grenzen des selektierten Bereiches bzw. auch vorwiegend in den bereits durch das Anstiegs- oder Abfallkriterium markierten Bereiches. Hierdurch kann festgestellt werden, ob und wann Teilchen die Datensatzgrenze verlassen. Daraufhin kann falls nötig im Diagramm der gewählte Zeitbereich angepasst werden. Diese Anpassung des Betrachtungsbereichs sowie die Auswahl des Zeitbereiches ist jedoch nicht trivial. Beim Tragflächen-Datensatz ist eine Eingrenzung über den Zeitbereich noch machbar, da

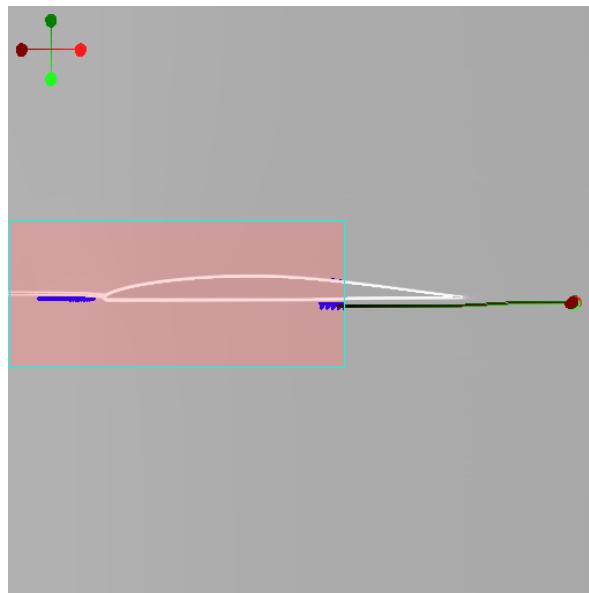


Abbildung 7.17: FTLE-Bild mit selektiertem Bereich. Blau markiert ist das Gebiet, indem das Kriterium (angestiegener FTLE-Wert) zutrifft. Unterhalb des Flügels wurde die Trajektorie einer Partikelgruppe gezeichnet

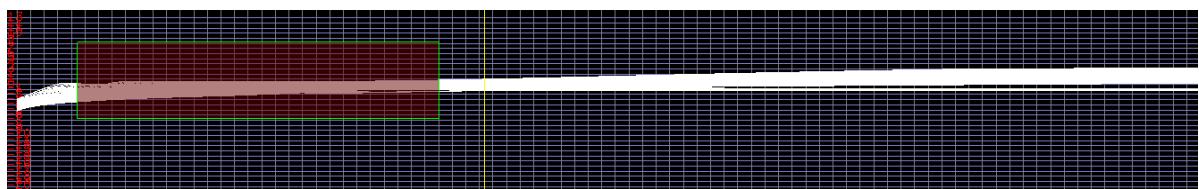


Abbildung 7.18: Diagramm der selektierten Kurven bei richtiger Begrenzung von Δt : die gelbe Linie hinter dem selektierten Bereich zeigt die aktuelle Position des in Abbildung 7.17 gezeigten Partikels.

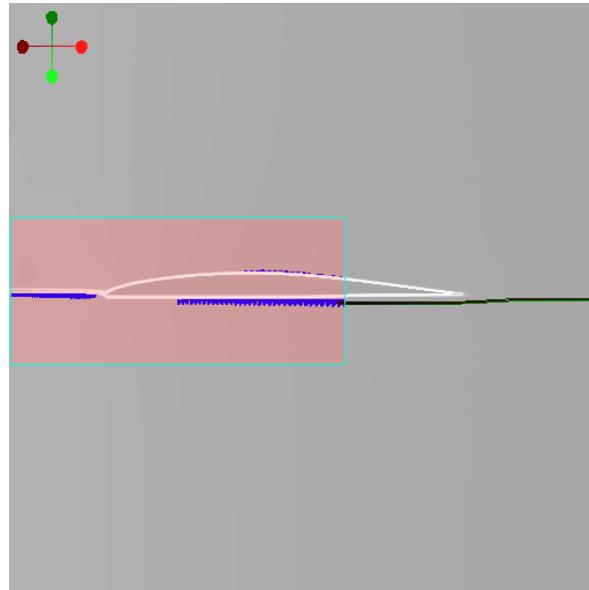


Abbildung 7.19: Der selektierte Bereich sowie die Partikel-Startposition entspricht der Konfiguration in Abbildung 7.17. Die Endposition der verfolgten Partikel liegt jedoch bereits außerhalb des Datensatzes.

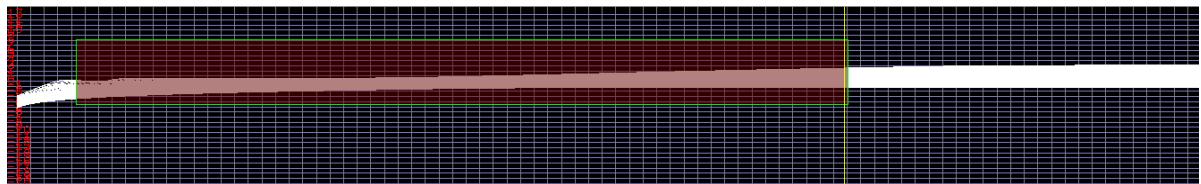


Abbildung 7.20: Diagramm der selektierten Kurven zu Abbildung 7.19.

nur eine dominante Strömungsrichtung (nach rechts) vorliegt und Abweichungen/Verwirbelungen gegenüber der horizontalen Bewegung vernachlässigbar sind. Jedoch ist auch hier bereits die Problemstellung gegeben, dass weiter links liegende Teilchen erst zu einem späteren Zeitpunkt den rechten Bildrand verlassen - wenn ein an der rechten Tragflächenkante gestartetes Teilchen bereits die Datensatzgrenze verlässt erreicht ein ganz links gestartetes Teilchen ggf. noch nicht einmal die rechte Flügelkante. Bei sehr weit links liegenden Teilchen werden hier bei kurzen Zeitbereichen Features übersehen, bei rechts liegenden hingegen besteht die Gefahr, Randgebieterscheinungen als Feature misszudeuten. Bei komplexeren Datensätzen ist die Auswahl des Zeitbereiches desto komplexer bzw. führt nicht zum gewünschten Ergebnis.

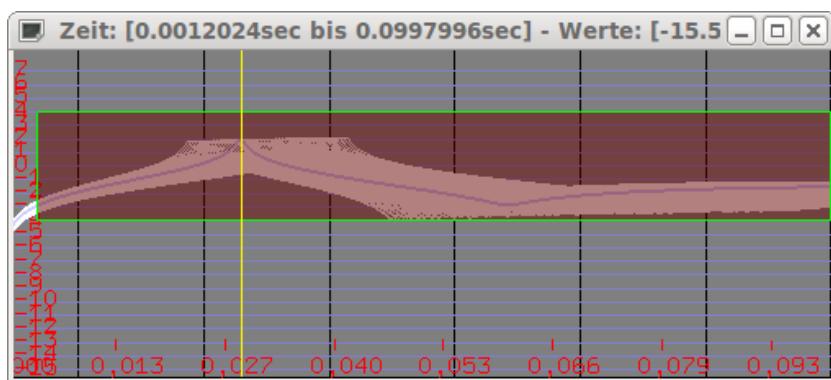
Gelöst werden kann dies durch ein Abbruchkriterium, dass sobald ein Teilchen den Bildrand verlässt (und die Kurve bis dahin noch nicht als markiert galt) eine weitere Untersuchung dieser einzelnen Kurve unterbindet. Somit können beliebig lange Zeitbereiche untersucht werden, ohne dass ungewollte Sprünge durch das Verlassen der Datensatzgrenze einbezogen werden. Ein Verwerfen der kompletten Kurve ist hingegen nicht sinnvoll, da die Werte vor dem Verlassen der Datensatzgrenze gültig sind - und bei chaotischen Datensätzen nach hinreichender Integrationszeit sehr viele Startpositionen zu einer Endposition außerhalb des Datensatzes führen können.

7.8 Kombination beider Kriterien

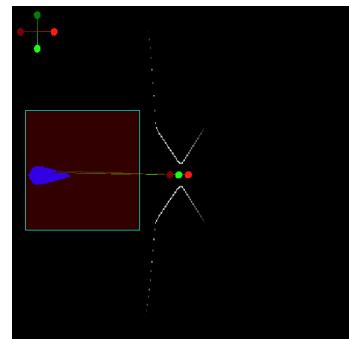
Um Bereiche zu finden, bei denen der FTLE-Wert zuerst ansteigt und daraufhin wieder abfällt (bzw. umgekehrt) kann das Kriterium auch ein zweites Mal mittels „und“-Verknüpfung angewendet werden. Die Erfüllung des zweiten Kriteriums wird jedoch nur auf dem Bereich, der zeitlich nach der Erfüllung des ersten Kriteriums liegt, geprüft. Würde dies nicht berücksichtigt, so könnte nicht unterschieden werden ob ein Wert zuerst über das Minimum ansteigt und dann unter das Maximum abfällt oder zuerst abfällt und dann ansteigt. In Abbildung 7.21(a) muss der FTLE-Wert ein Minimum haben, über das er dann wieder ansteigt. Zur Erfüllung der Kriterien in Abbildung 7.21(c) ist hingegen ein Maximum mit darauffolgendem Abfallen der Werte notwendig.

7.9 Resonanzen und Drehgeschwindigkeiten von Wirbeln

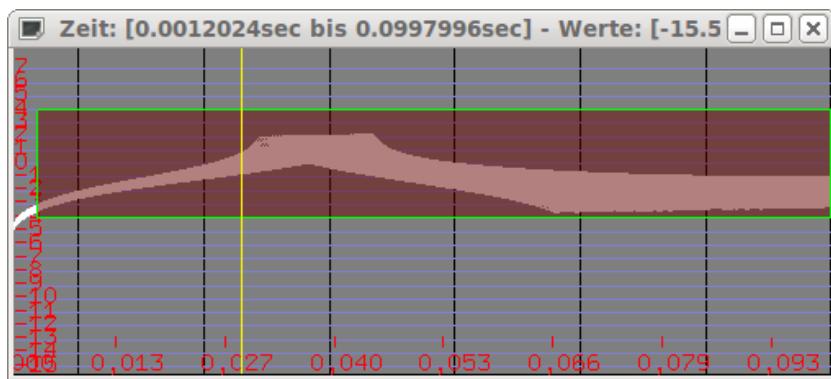
Der QuadGyre Datensatz (siehe Abschnitt 8.1.1) besteht aus 4 Wirbeln. Die Drehrichtung dieser Wirbel ist in Abbildung 7.22 mit gelben Pfeile eingezeichnet. Im Diagramm werden durch diese Wirbel periodische Schwingungen hervorgerufen (Abbildung 7.23(b)). Bei der Erstellung des Diagramms wurde die Division durch Δt deaktiviert, um auch gegen Ende des Betrachtungszeitraumes noch Wertschwankungen verfolgen zu können. Um die Umlaufdauer zu untersuchen wird ein Partikel an einer Stelle verfolgt, die ein deutliches Oszillieren des FTLE-Wertes im Diagramm aufweist. In Abbildung 7.24 wurden die Integrationslängen jeweils so gewählt, dass im Diagramm an dieser Stelle ein Minimum vorliegt. Dies ist jeweils beim Erreichen des Ausgangspunktes der Fall, also nach vollständigen Umdrehungen. Da die Trajektorien des Particle Tracers durch lineare Abschnitte interpoliert werden und die einstellbaren Zeitschritte diskret sind muss hier 1 Zeitschritt vor dem Tiefpunkt sowie 1 Zeitschritt nach dem Tiefpunkt untersucht werden. Falls diese weiter vom ursprünglichen Startpunkt entfernt sind als die Position am Tiefpunkt, so kann die Position am Tiefpunkt als vollständige Umdrehung angesehen werden. Nach der 5. Umdrehung wurde am verfolgten Partikel eine Integrationslänge von $t_{tracer} = 1.728s$ ermittelt. Die



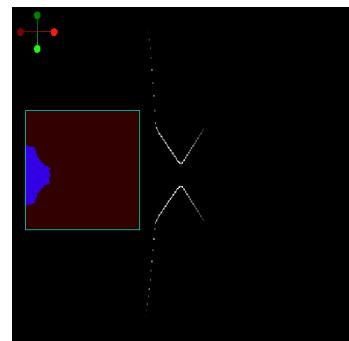
(a) Zuerst abfallen um 4 über 10 slices, dann ansteigen um 0,1 über 10 slices.



(b) Zusätzlich wurde ein Partikel eingestreut, der den Peak nach unten verdeutlicht.



(c) Zuerst abfallen um 4 über 20 slices, dann Ansteigen um 3,1 über 20 slices.



(d) FTLE-Bild zu Abbildung 7.21(c).

Abbildung 7.21: Verknüpfung der Detektion abfallender und ansteigender Werte.

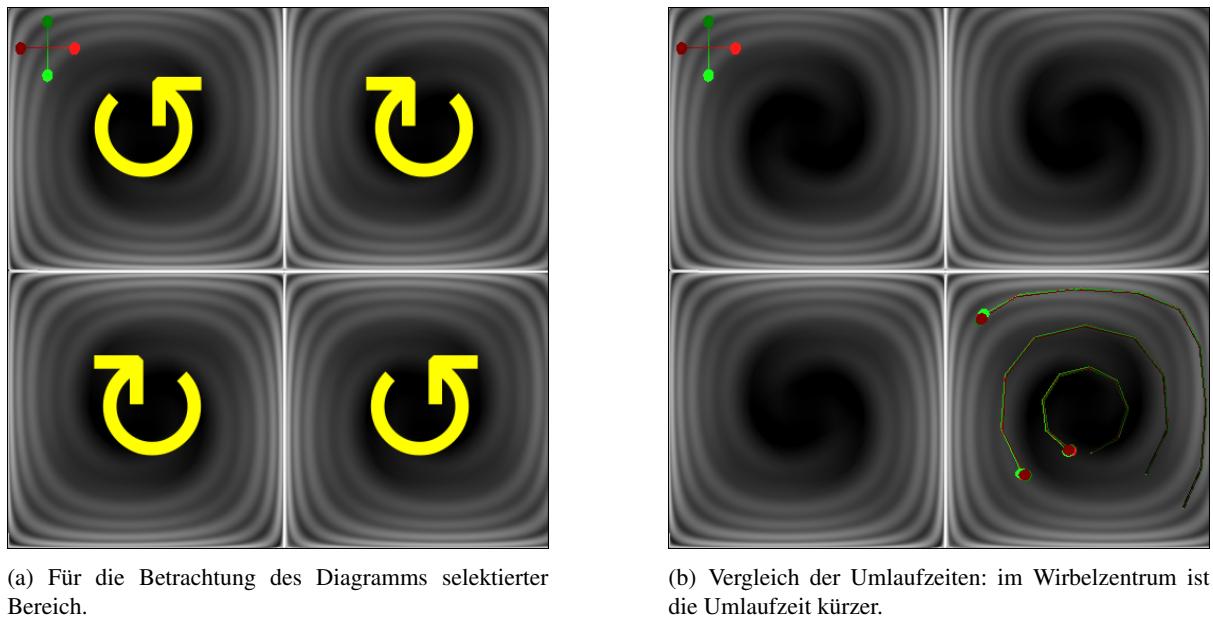


Abbildung 7.22: Drehrichtung und Umlaufzeiten eines Wirbels des QuadGyre-Datensatzes.

Umlaufdauer des Teilchens beträgt somit $t_r = \frac{1.728s}{5}$ In Abbildung 7.22(b) ist anhand der verfolgten Partikel sichtbar, dass die Umlaufdauer im Zentrum des Wirbels geringer als am Rand des Wirbels ist. Im Diagramm äußert sich dies durch ein Auseinanderdriften der Position der Tiefpunkte: Zum Zeitpunkt der Abbildung 7.24(b) fallen noch alle Tiefpunkte auf die gleiche Position auf der Z-Achse. Beim fünften Wirbel (Abbildung 7.24(j)) hingegen ist der Zeitpunkt, zu dem der Tiefpunkt auftritt bereits je nach Position des Teilchens deutlich früher oder später.

Beim Clustering fallen Kurven mit ähnlichem Auftrittszeitpunkt des Tiefpunktes in das gleiche Cluster. Hierdurch bilden sich beim Clustering kreisförmige Ringe als Clusterform aus.

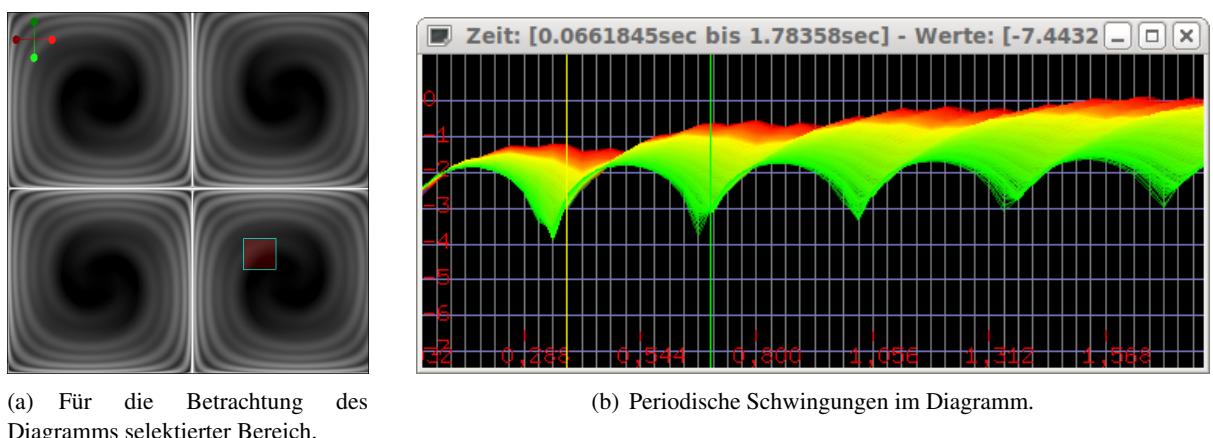


Abbildung 7.23: Untersuchung des Diagramms eines Wirbels.

7.9.1 FFT

In Abbildung 7.26 wird die Fourier-Transformierte des FTLE-Werteverlaufs betrachtet. Mittels Zoom ist der Betrachtungsbereich des Diagramms in Abbildung 7.25(b) auf das erste Viertel beschränkt. Die Frequenzen sind im Diagramm aufsteigend von links nach rechts angeordnet. Der deutliche Peak bildet sich durch das zyklische Steigen und Fallen des FTLE-Wertes aufgrund der Rotation im Wirbel aus.

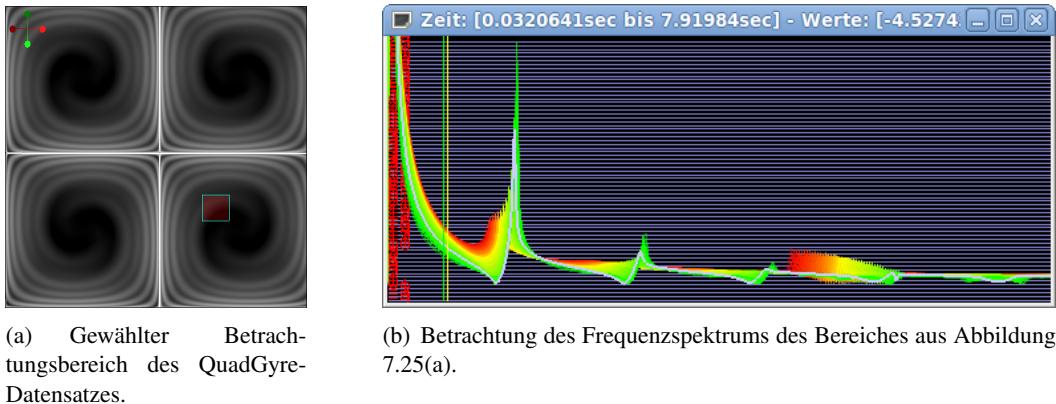
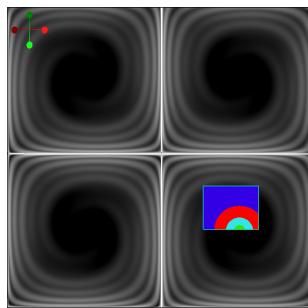
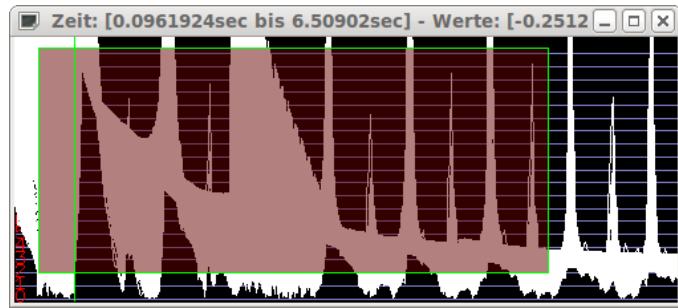


Abbildung 7.25: Betrachtung der Fourier-Transformierten.

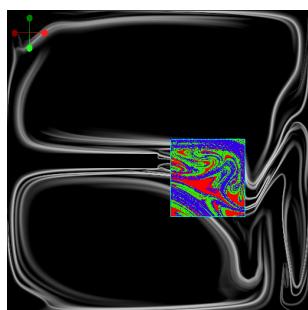
Wenn wie in Abbildung 7.25(b) nur ein kleines Gebiet selektiert wird, ist sind die Umlaufdauer aller Teilchen ähnlich. Wird jedoch ein größerer Bereich nahe des Wirbelzentrums gewählt, fallen Partikel mit unterschiedlicher Umlaufdauer in diesen Bereich. Das Frequenzspektrum enthält mehrere, weniger scharf begrenzte Peaks (Abbildung 7.26(b)). Clustering führt hier zu einer Gruppierung der Partikeln mit ähnlicher Umlaufdauer. Aus diesem Grund bilden sich in Abbildung 7.26(b) kreisförmige Cluster aus. Beim Bouyancy-Datensatz hingegen herschen viele Verwirbelungen, die zu einem chaotisch wirkenden Frequenzspektrum führen. Erwartungsgemäß liefert Clustering hierauf eine sehr verrauschte Clusteraufteilung, dies ist in den Abbildungen 7.26(c) und 7.26(d) ersichtlich.



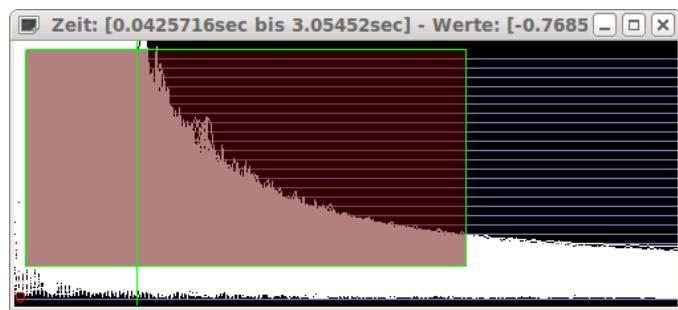
(a) Gewählter Betrachtungsbereich des QuadGyre-Datensatzes. Die 4 Cluster bilden kozentrische Kreise.



(b) Betrachtung des Frequenzspektrums des Bereiches aus Abbildung 7.26(a).

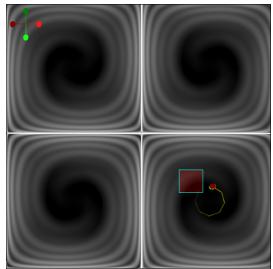


(c) Im Bouyancy-Datensatz sind die Cluster weniger zusammenhängend.



(d) Betrachtung des Frequenzspektrums des Bereiches aus Abbildung 7.26(c).

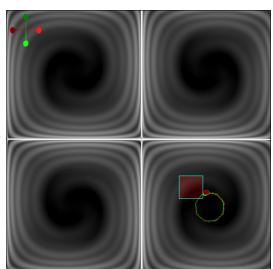
Abbildung 7.26: Betrachtung der Fouriertransformierten und Clustering.



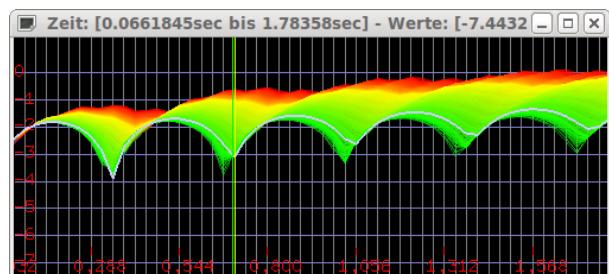
(a) 1. Umdrehung.



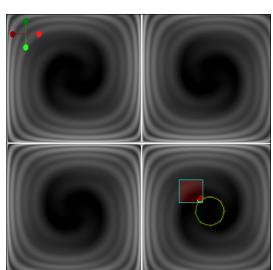
(b) Periodische Schwingung im Diagramm.



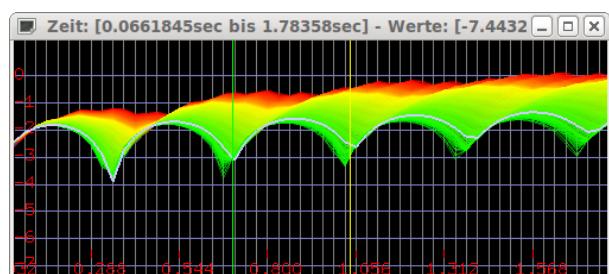
(c) 2. Umdrehung.



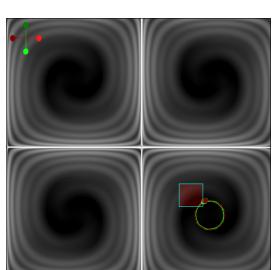
(d) Periodische Schwingungen im Diagramm.



(e) 3. Umdrehung.



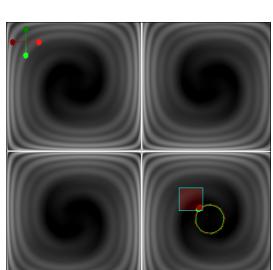
(f) Periodische Schwingungen im Diagramm.



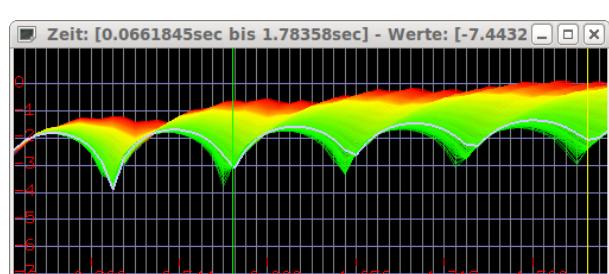
(g) 4. Umdrehung.



(h) Periodische Schwingungen im Diagramm.



(i) 5. Umdrehung.



(j) Periodische Schwingungen im Diagramm.

Abbildung 7.24: Untersuchung der Tiefpunkte im Diagramm.

Kapitel 8

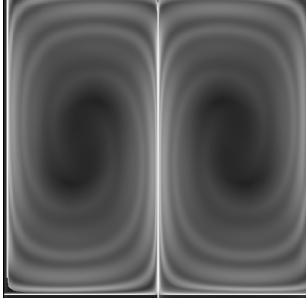
Auswertung

8.1 Strömungsdatensätze

Strömungsdatensätze können entweder analytisch oder in Form von diskretisierten Werten auf einem Gitter vorgegeben werden. Die Wahl der Speicherungsmethode hängt maßgeblich von der Herkunft der Daten ab - analytische Simulationsdaten können sowohl als Formel gespeichert werden als auch in diskrete Werte umgewandelt werden. Die Umwandlung von Gitterbasierten Daten einer Simulation in ein analytische Form ist jedoch nur durch Approximation und Interpolation (z.B. durch lineare Interpolation oder Bézierskurven) möglich - und ergibt in vielen Fällen unhandlich große Gleichungssysteme.

8.1.1 Analytisch vorgegebene Strömungsdatensätze

Bei analytischer Vorgabe eines Strömungsdatensatzes wird der Geschwindigkeitsvektor $\vec{v}(\vec{P}, t)$ eines Teilchens an der Position P zum Zeitpunkt t mit Hilfe eines Gleichungssystems vorgegeben. Diese Vorgehensweise ist nur für simulierte Strömungsfelder verwendbar, da bei Messungen diskrete Messwerte statt einer Gleichung vorliegen. Der Vorteil eines analytischen Datensatzes liegt darin, dass keine Interpolation zwischen Messwerten notwendig ist. Problematischer ist allerdings die Einbindung solcher Datensätze in das Auswertungsprogramm, da der Datensatz entweder bereits zum Zeitpunkt des Compilierens in den Quelltext eingebunden werden muss - oder ein Interpreter für mathematische Formeln verwendet werden muss. Analytische Datensätze wurden hier lediglich zur Validierung verwendet und für diesen Verwendungszweck fest in den Quelltext einprogrammiert. Die zur Auswertung verwendeten analytisch erstellten Datensätze wurden auf einem kartesischen Gitter mit 41x41 Punkten diskretisiert.

	<p>DoubleGyre</p>	<p>Auf einem 41x41x321 großen Feld werden 2 Wirbel simuliert. Die Formel zur Erstellung des Datensatzes lautet $\vec{v}(x, y, t) = \left(-\pi A \sin(\pi f(x, t)) \cos(\pi y), \pi A \cos(\pi f(x, t)) \sin(\pi y) \frac{df}{dx} \right)$ mit $f(x, t) = a(t)x^2 + b(t)x$, $a(t) = \epsilon \sin(\omega t)$, $b(t) = 1 - 2\epsilon \sin(\omega t)$ und stammt aus [SLM05]. Als Parameter wurden $\epsilon = 0.25$ und $\omega = 2\pi$ gewählt. In dieser Abbildung beträgt $T_0 = 1.0\text{sec}$, $\Delta t = 0.81\text{sec}$.</p>
---	-------------------	---

	QuadGyre	<p>Vierfacher Wirbel.</p> <p>Der Datensatz wurde mit der gleichen Formel wie der DoubleGyre erstellt, allerdings wurde die y-Achse auf einem Bereich von [0..2] statt [0..1] skaliert. Für die Darstellung wurde wie beim DoubleGyre $T0 = 1.0sec$ und $\Delta t = 0.81sec$ verwendet.</p>
--	----------	--

8.1.2 Strömungsdatensätze auf regulären Gittern

Die vorliegenden Datensätze liegen in einem Dateiformat vor, das diskrete Werte auf einem regulären Gitter beinhaltet. Prinzipiell sind auch andere Datensatzformen denkbar: Beispielsweise könnte statt des regulären Gitters ein an Regionen mit komplexem Strömungsverhalten feiner aufgelöstes Gitter verwendet werden, um dort die Strömungsdaten detaillierter erfassen zu können. Auch wäre ein rundes statt eines rechteckigen Gebietes denkbar - diese Möglichkeiten werden hier jedoch nicht behandelt, es wird stets ein reguläres Gitter mit equidistanter Unterteilung angenommen.

	Bouyancy	<p>Der von Sadlo in [SW10] verwendete Datensatz simuliert eine geschlossene Kammer, in der ein Gas angeregt durch unterschiedliche Temperaturniveaus zirkuliert. In der Mitte dieser Kammer befindet sich ein für das Gas nicht durchdringbares Hindernis.</p> <p>Die Größe des Datensatzes beträgt $41x41$ Einträge in x-/y-Richtung und 321 Zeitschritte. Abgebildet ist der Datensatz zum Zeitpunkt $T0 = 4.5sec$ mit einer Integrationslänge von $\Delta t = 0.8sec$.</p>
--	----------	--

8.1.3 Datensatz: elektrisches Feld

Bei diesen Datensätzen wird die Wirkung eines elektrischen Feldes auf eine Probeladung simuliert. Zwischen im Raum verteilten Punktladungen bildet sich ein elektrisches Feld aus. Es wurden zwei positive und eine negative Punktladung verwendet. Die Probeladung hat eine negative Ladung von $q_{probe} = 10e^-$.¹

Beim Eletreco3-Datensatz werden 1 negative und 2 positive Punktladungen auf einer 2-dimensionalen Ebene platziert. Die positiven Ladungen sind Q_1 und Q_2 mit $q_1 = 400C$ und $q_2 = 200C$. Diese befinden sich bei den Koordinaten $\vec{P}_{Q_1} = (1.01cm|1.01cm)$ bzw. $\vec{P}_{Q_2} = (8.01cm|1.01cm)$. Die Negative Punktladung Q_3 an Position $\vec{P}_{Q_3} = (7.01cm|8.01cm)$ ist mit $q_3 = -200C$ geladen. Bei der Erstellung

¹ e^- ist die Konstante für die Ladung eines Elektrons, wird in Coulomb angegeben und beträgt $e^- \approx -1.60E^{-19}C$ [NIS].

weiterer Felder wurde die Position der Punktladungen, deren Anzahl und Ladung variiert. Die Kraft, die zwischen Probeladung und einer Ladung Q_n wirkt, wird mit Hilfe der Coulombschen Gesetze [MIT10] berechnet und ist durch die Formel

$$\vec{F}_{probe,n} = \frac{q_{probe} * q_n}{4\pi\epsilon_0\epsilon_r} \frac{P_{probe} - P_n}{|P_{probe} - P_n|^3} \quad (8.1)$$

beschrieben. Die resultierende Kraft unter Berücksichtigung aller 3 Punktladungen wird mittels Superposition ermittelt:

$$\vec{F}_{probe,res} = \sum_{n=1}^3 \vec{F}_{probe,n} \quad (8.2)$$

$\vec{F}_{probe,res}$ wird für jede Gitterzelle im Datensatz berechnet. Hierfür wird die Position der virtuellen Probeladung auf den jeweiligen Gitterpunkt gesetzt und die resultierende Kraft ermittelt. Es ist zweckmäßig, die Koordinaten der Punktladungen nicht exakt auf einen Gitterpunkt des Datensatzes zu legen, da sonst $|P_{probe} - P_n|^3 = 0$ wird und dies in Gleichung 8.2 eine Division durch 0 erzeugt. Dies wiederum bringt ein unerwünschtes *NaN* im erstellten Datensatz mit sich, welches bei der Darstellung im FTLE ein schwarzes Kästchen zur Folge hat, da jeder Zugriff auf diese Strömungsdatensatz-Zelle einen Fehler hervorruft.

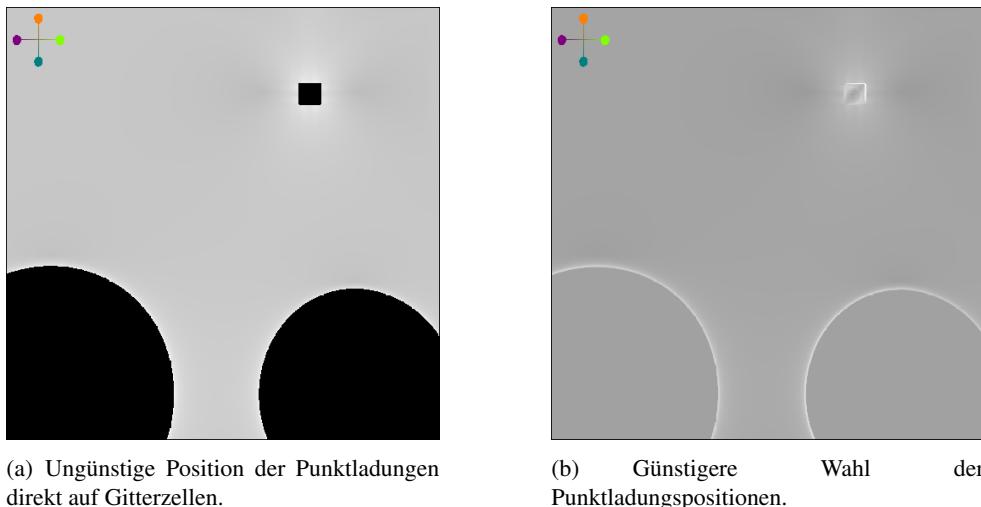
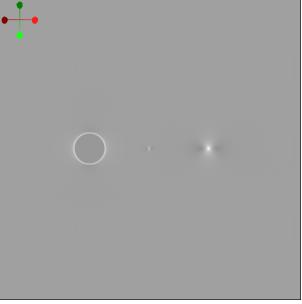
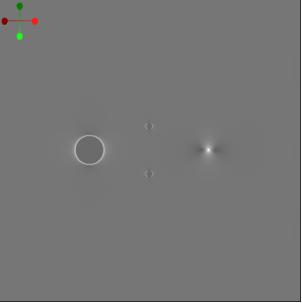
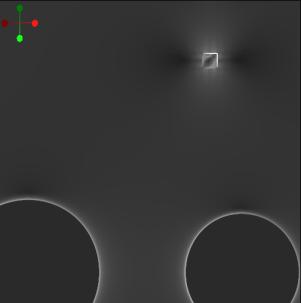


Abbildung 8.1: Positionierung der Punktladungen bezüglich des Strömungsdatensatzgitters.

Folgende Datensätze wurden erstellt:

	Electro1	Auf der linken Seite befindet sich eine positiv Ladung, auf der rechten Seite eine negative. Zwischen den beiden Ladungen befindet sich eine weitere negative Ladung. Somit wird eine negative Probeladung, die von der rechten zur linken Punktladung gezogen wird, von der mittleren Punktladung zusätzlich abgestoßen.
	Electro2	Im Gegensatz zum Electro1-Datensatz sind zur Ablenkung 2 positive Punktladungen angebracht. Durch die zusätzliche Anziehung zu diesen Punktladungen kann bei der Verfolgung eines Partikels von rechts nach links ein temporärer Anstieg des FTLE-Wertes verzeichnet werden.
	Electro3	Im oberen Bereich befindet sich eine negative Ladung, im unteren Bereich sind 2 positive Punktladungen. Bei der Verfolgung der Trajektorie negativer Probeladungen kann die Ausbildung eines Grenzbereichs zwischen den beiden positiven Ladungen beobachtet werden. Im Detail wird dies in Abschnitt 8.4 behandelt.

8.1.4 Mit Matlab als partielle Differentialgleichung erstellte Datensätze

Matlab verfügt über eine GUI zur grafischen Eingabe und Visualisierung von partiellen Differentialgleichungen, der Partial Differential Equation Toolbox. Dieses Tool ist in Abbildung 8.2 abgebildet und dient der Lösung partieller Differentialgleichungen. Mit der Erstellung von Diffusionsgleichungen können bei Betrachtung des Gradientenfeldes jedoch auch Fluid-ähnliche Bewegungen simuliert werden. Bei der verwendeten Version 7.12.0 von Matlab kann diese GUI mit dem Befehl *pdetool* aufgerufen werden. Mit Hilfe der Maus können Ellipsen, Rechtecke und Polygone platziert werden - die Kanten dieser Primitive lassen sich zu Randbedingungen der partiellen Differentialgleichung modellieren.

Als Bedingungstyp stehen Neumann- und Dirichletbedingung zur Verfügung. Die Neumann-Bedingung ist zur Modellierung undurchlässiger Hindernisse geeignet, an der Kante des Hindernisses wird ein Gradient von 0 erzwungen, was bedeutet dass weder Teilchen in das Hindernis gelangen, noch welche aus dem Hindernis ausströmen. Die Dirichlet-Bedingung hingegen ist zur Modellierung der Feldkanten geeignet, der Parameter „c“ entspricht der Partikelkonzentration, die an diesen Stellen herrscht. Berechnet wird die Lösung der partiellen Differentialgleichung auf einem Dreiecksgitter. Zur Verwendung in dem für diese Diplomarbeit erstellten Programmes müssen die Daten noch auf ein kartesisches Gitter

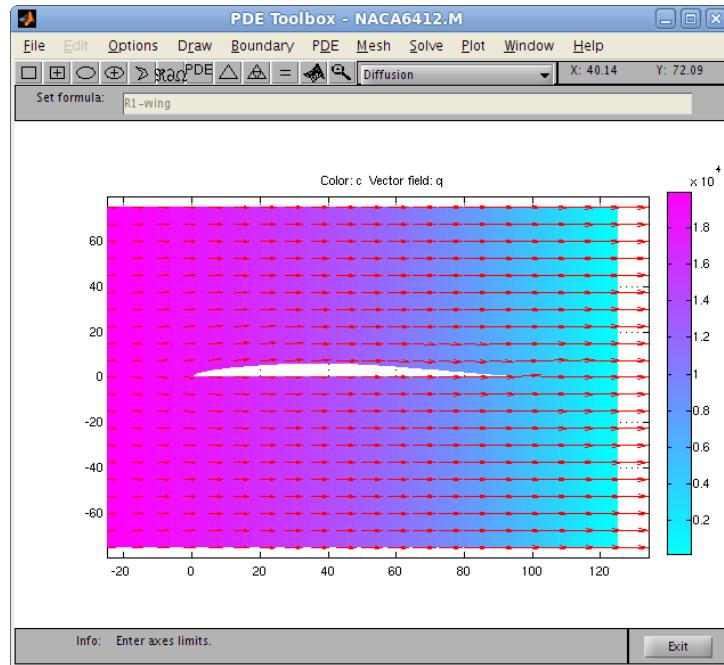
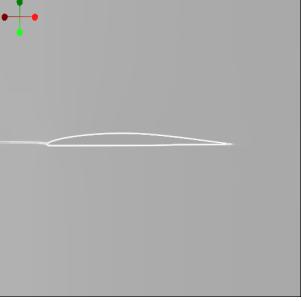
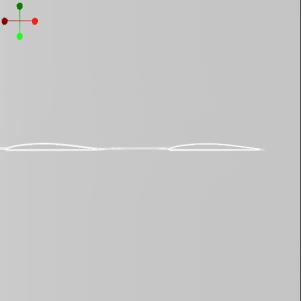
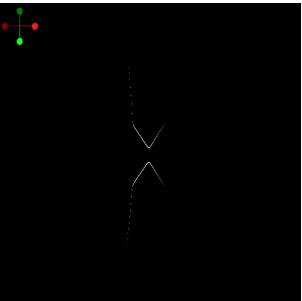
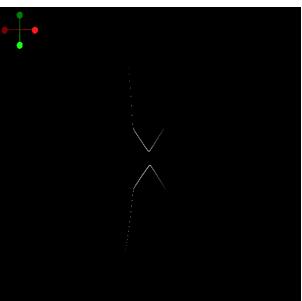
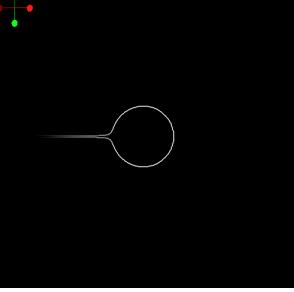
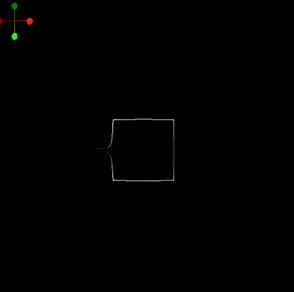


Abbildung 8.2: Partial Differential Equation Toolbox in Matlab: grafische Eingabe und Visualisierung von partiellen Differentialgleichungen.

umgerechnet und in das für das Auswertungsprogramm erforderliche Format konvertiert werden. Die Umrechnung und Konvertierung geschieht mit Hilfe eines von Markus Üffinger zu Verfügung gestellten Matlab-Scripts.

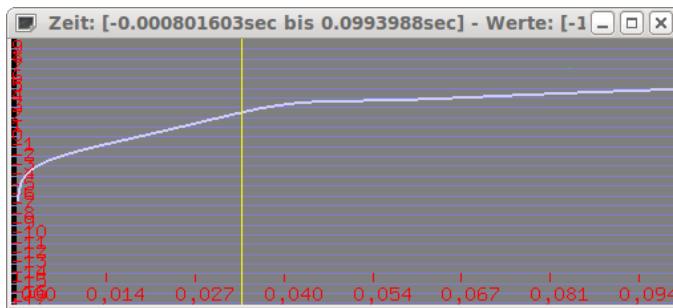
	Wing	Nachbildung eines NACA6412 Tragflächenprofils [AID11]. Auf der linken Seite wurde eine hohe Partikelkonzentration modelliert, sodass eine Strömung von links nach rechts vorliegt. Dargestellt ist $T0 = 0s$, $\Delta t = 1s$.
	DoubleWing	Um mehrmaliges Ansteigen und Abfallen des FTLE-Wertes untersuchen zu können wurden 2 NACA6412-Tragflächen hintereinander modelliert. Dargestellt ist ebenfalls $T0 = 0s$, $\Delta t = 1s$.
	Bottleneck	An der Engstelle sinken die FTLE-Werte ab und steigen kurz dahinter wieder an. Die Darstellung erfolgte hier mit $T0 = 0s$ und $\Delta t = 0.1s$
	Bottleneck Assymetric	Gleiche Eigenschaften wie der Bottleneck-Datensatz, allerdings ist die untere Spitz um 10% des Abstands zwischen oberer und unterer Hindernisspitze nach rechts verschoben.

	Circle	Umströmen eines kreisförmigen Hindernisses. Das FTLE-Bild wurde mit $T0 = 0s$, $\Delta t = 0.1s$ erzeugt.
	Square	Umströmen eines Quadrates. Die höchste Konzentration wurde auf der linken Seite vorgegeben, sodass sich ein Fluss von links nach rechts einstellt. Parameter zur Darstellung waren hier $T0 = 0s$ und $\Delta t = 0.1s$

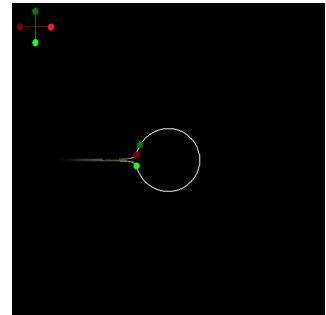
8.2 Circle-Datensatz

Die Trajektorie eines direkt vor dem Hindernis gestarteten Partikels ist Abbildung 8.3(a) bis 8.3(h) dargestellt. Zum Zeitpunkt $\Delta t = 0.075s$ ist ein Absinken des FTLE-Wertes zu erwarten, jedoch steigt der Wert kontinuierlich weiter. Ursache ist ein Problem bei der Integration in der Nähe des Hindernisses: Da innerhalb des Hindernisses die Strömung konstant $\vec{v} = \vec{0}$ ist endet die Bewegung eines Partikels, sobald er das Hindernis berührt. Eine Lösung wäre die Modifikation des Integrators: Die Hindernisse müssten durch ein Flag bzw. einen Eintrag von *NaN* (Not a Number) im Strömungsdatensatz gekennzeichnet werden. Falls ein Partikel nach der Integration in ein so markiertes Gebiet eintritt wird es vom Integrator auf die letzte Position außerhalb des Hindernisses zurückgesetzt.

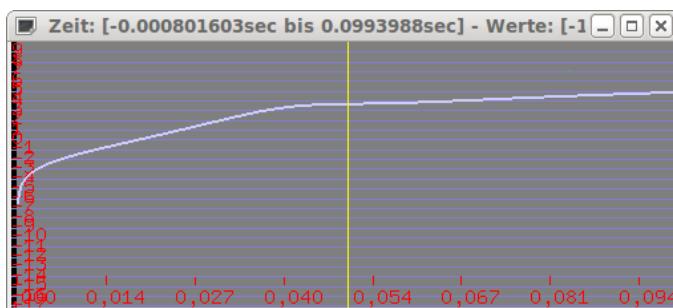
Das Clustering auf einem Bereich neben dem Hindernis liefert homogene Cluster (Abbildung 8.4(b)). Es wurde darauf geachtet, dass weder einzelne Partikel bereits die Grenzen des Datensatzes überschreiten noch Partikel einbezogen werden, die am Hindernis stehen bleiben. Abbildung 8.4(c) und Abbildung 8.4(e) zeigen die Diagramme mit dem Verlauf der FTLE-Werte. Die in Abbildung 8.4(d) und Abbildung 8.4(f) gesetzten Particle Tracer sind auch in den dazugehörigen Diagrammen eingezeichnet. Die Diagramme beinhalten jeweils nur die Kurven der FTLE-Werte, die in dieses Cluster fallen. Bei dem Cluster, dass am weitesten vom Hindernis entfernt ist, findet lediglich ein gleichmäßiger Anstieg der Werte statt. Die im grün gefärbten Cluster liegenden Werte steigen hingegen deutlich stärker an und fallen hinter dem Hindernis wieder leicht ab. Die Kurven der FTLE-Werte haben eine geringe Streuung, es sind keine Ausreißer feststellbar.



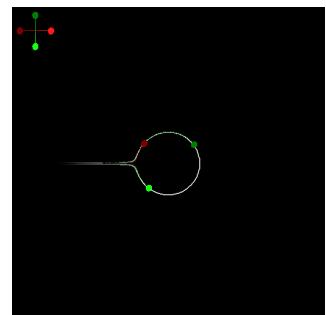
(a) $\Delta t = 0.034s$: Anstieg des FTLE-Wertes durch Separation am Hindernis.



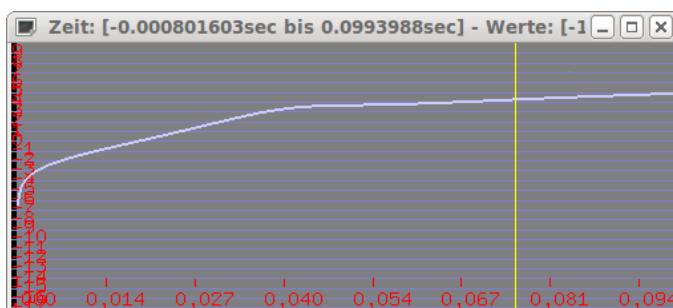
(b) FTLE und Trajektorien zu Abbildung 8.3(a).



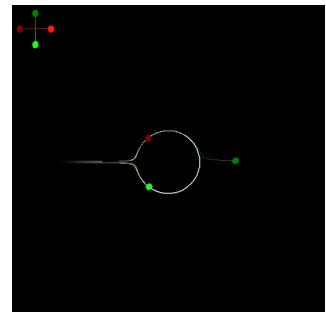
(c) $\Delta t = 0.05s$: Ein Partikel (hellgrün) tangiert den Rand des Hindernisses und bleibt dort stehen.



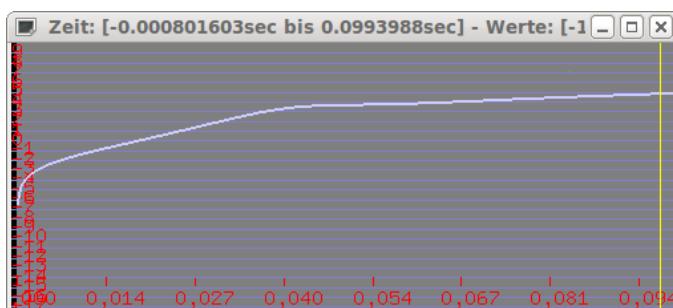
(d) FTLE und Trajektorien zu Abbildung 8.3(c).



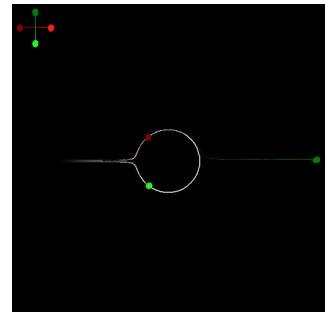
(e) $\Delta t = 0.075s$: die Trajektorie des roten Partikels endet ebenfalls.



(f) FTLE und Trajektorien zu Abbildung 8.3(e).



(g) Bei $\Delta t = 0.097s$ bewegt sich nur noch 1 Partikel.



(h) FTLE und Trajektorien zu Abbildung 8.3(e).

Abbildung 8.3: Trajektorie von Partikeln im Circle-Datensatz.

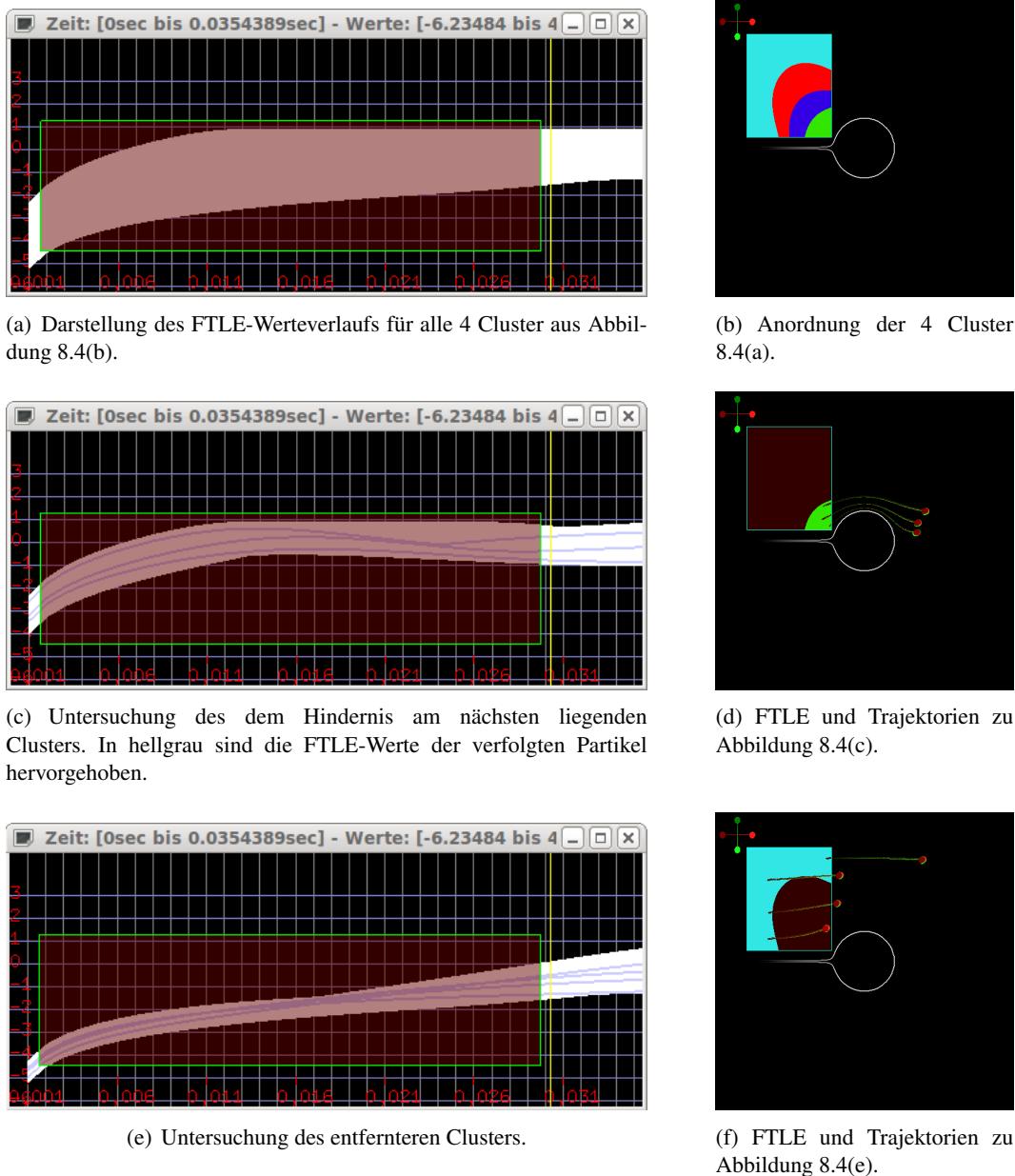


Abbildung 8.4: Clustering im Circle-Datensatz.

8.3 Bouyancy-Datensatz

Am Beispiel des Bouyancy-Datensatzes können turbulente Regionen mit starken Verwirbelungen untersucht werden. Die Unterschiede im Verlauf der FTLE-Werte zwischen Bereichen mit homogener Strömung und verwirbelten Gebieten sind im Diagramm klar zu unterscheiden: Abbildung 8.5(a) zeigt den Verlauf im Zentrum eines Wirbels. Im Diagramm liegen die einzelnen Kurven bis $\Delta t = 2.0s$ sehr eng beieinander. Erst dahinter divergieren die Kurven stärker, da die Trajektorien einzelner Partikel in verwirbeltere Regionen führen. Bei Betrachtung des Gradienten in Abbildung 8.5(c) ist dieser Effekt ebenfalls ersichtlich.

Eine Betrachtung eines turbulenteren Gebietes zeigen Abbildung 8.5(e) bis 8.5(h). In Abbildung 8.5(f) ist das betrachtete Gebiet zusätzlich durch einen roten Kreis hervorgehoben. In diesem Gebiet befinden sich viele, eng beieinander liegende ridges. Im Diagramm (Abbildung 8.5(e)) äußert sich dies durch eine hohe Divergenz zwischen den Kurven. Der Gradient in Abbildung 8.5(f) lässt zudem erkennen, dass das Steigen und Sinken der Werte zu unterschiedlichen Zeitpunkten stattfindet. Dies erschwert die Extraktion von Features: das Clustering neigt zum Rauschen. Je höher die Clusterzahl, desto stärker: Bei 3 Clustern (Abbildung 8.6(b)) sind diese noch überwiegend zusammenhängend, bei 8 Clustern (Abbildung 8.6(d)) bilden sich sehr ungleichmäßige Ränder aus. In homogenen Regionen wie im Zentrum eines Wirbels liegen die Kurven im Diagramm am Anfang des Integrationsbereiches zusammen. Da die Trajektorien einiger Partikel aus der Wirbelmitte jedoch in turbulenteren Gebieten führen treten für längere Integrationszeiten größere Unterschiede zwischen den einzelnen Werteverläufen auf. In Turbulenten Regionen sind die FTLE-Werte bereits bei geringen Integrationszeiten sehr breit gefächert. Zusätzlich zu den vielen Minima und Maxima aus Abbildung 8.5(e) ist in Abbildung 8.5(g) zu erkennen, dass diese für die einzelnen Kurven zu verschiedenen Zeitpunkten auftreten. Da das Clustering auf einem Zeitbereich durchgeführt wird, die Berechnung des FTLE-Bildes jedoch lediglich die Separation zum Zeitpunkt $T_0 + \Delta t$ visualisiert können sich unterschiedliche Strukturen ausbilden, die nicht zwingend vergleichbar sind.

Das Resultat der Detektion fallender und steigender FTLE-Werte ist ebenfalls verrauscht. Aufgrund des großen Selektionsbereichs in Abbildung 8.6(e) und der damit verbundenen hohen Anzahl Liniensegmente im Diagramm wird automatisch die Anzahl der Liniensegmente reduziert, um die Speicherauslastung zu senken. Auf die vom Detektionsalgorithmus zur Extraktion fallender und steigender FTLE-Werte verwendeten Daten hat diese Darstellungsform jedoch keinen Einfluss. Steigende Werte treten überwiegend auf ridges auf (Abbildung 8.6(h)). Jedoch konnten keine Parameter ermittelt werden, bei denen die ridges vollständig markiert werden, jedoch noch keine Punkte, die nicht auf ridges liegen, hinzukommen. Fallende Werte hingegen befinden sich überwiegend im Bereich zwischen den ridges (Abbildung 8.6(f)). Um ridges zu detektieren führt jedoch auch eine Invertierung dieser Markierung nicht zum Ziel, da diese ebenfalls von Rauschen überlagert ist.

8.4 Elektro3-Datensatz

Dieser Datensatz basiert auf eine 2-dimensionale Simulation der Kräfte elektrischer Ladungen und ist in Abschnitt 8.1.3 beschrieben. Aufgrund der Anziehung der beiden sich im unteren Bereich befindlichen positiven Ladungen auf die negative Probeladung wird die Probeladung je nach Startpunkt zu einer der beiden positiven Ladungen advektiert. Hierbei bildet sich eine Grenze zwischen den Partikeln, die die linke positive Ladung erreichen und Partikeln, deren Trajektorie in der rechten positiven Ladung

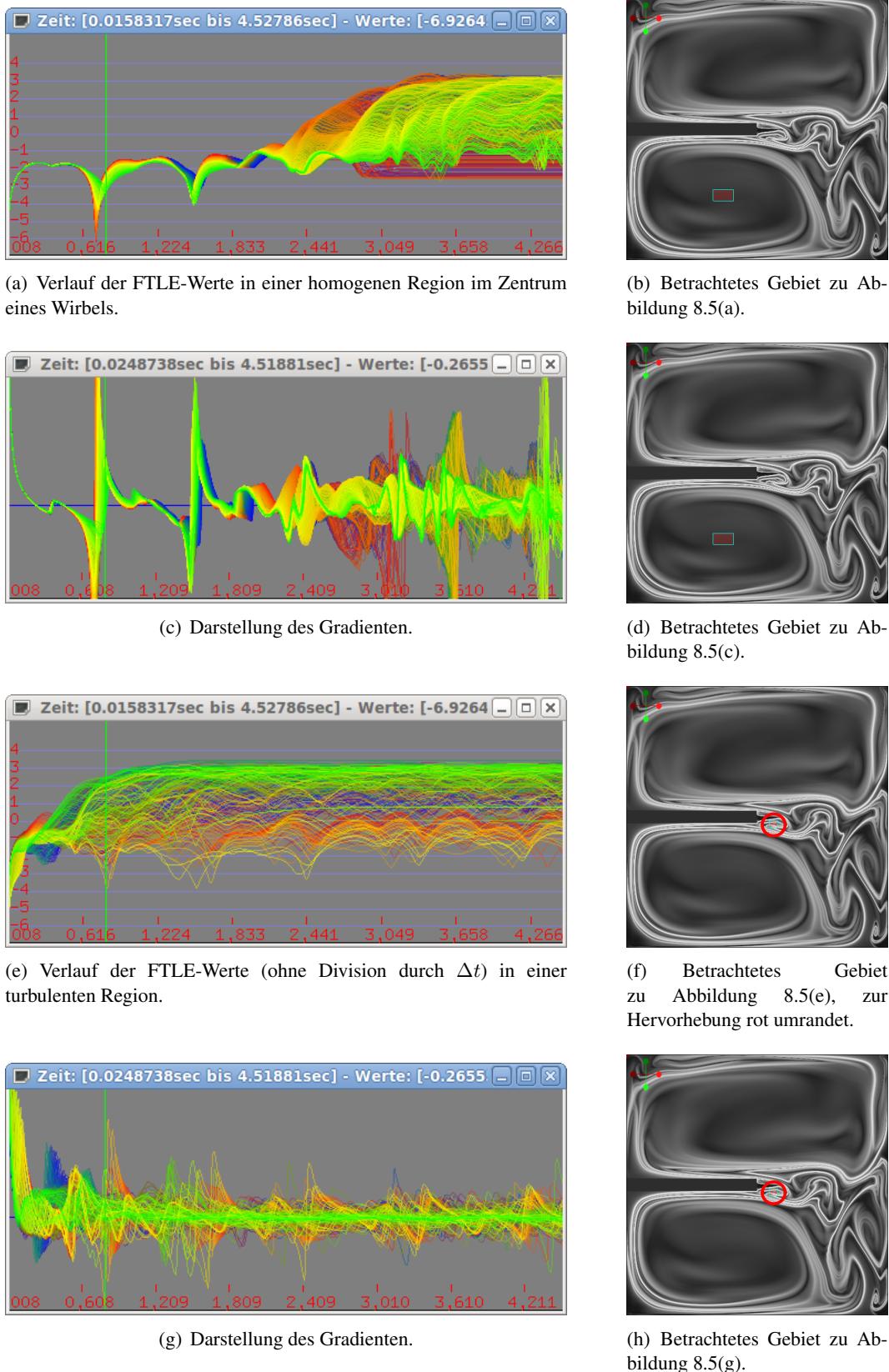
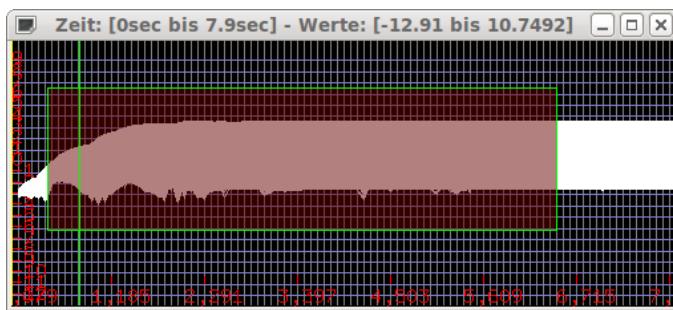
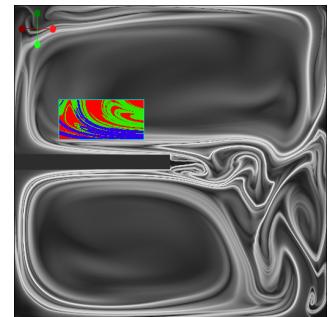


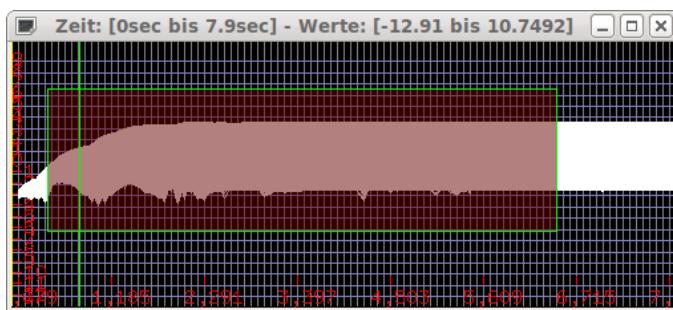
Abbildung 8.5: FTLE-Werte und deren Gradienten in Gebieten des Bouyancy - Datenstzes.



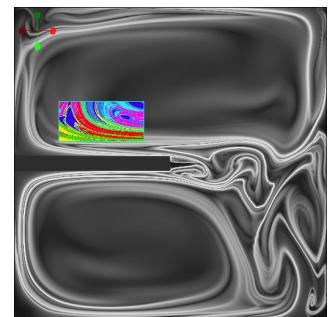
(a) Clustering mit 3 Clustern.



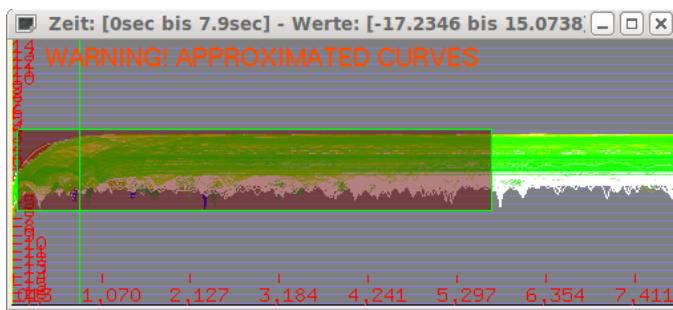
(b) FTLE und Trajektorien zu Abbildung 8.6(a).



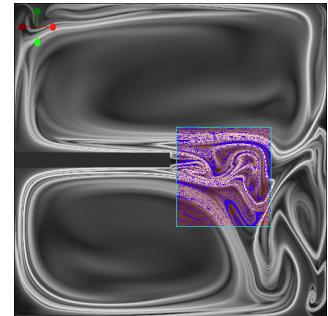
(c) Clustering mit 8 Clustern.



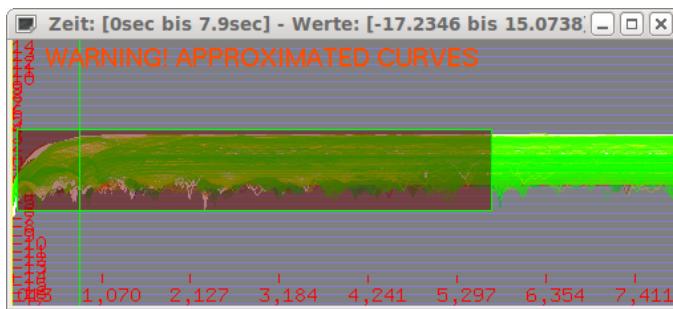
(d) FTLE und Trajektorien zu Abbildung 8.6(c).



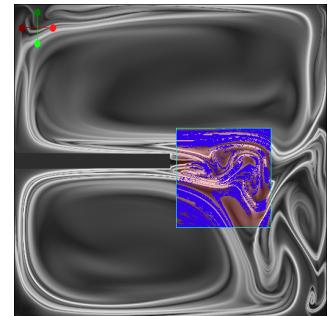
(e) Zeitliche Einschränkung des Gebietes, auf dem nach abfallenden FTLE-Werten gesucht wird.



(f) FTLE und Trajektorien zu Abbildung 8.6(e).



(g) Detektion steigender Werte: Zeitbereich.



(h) FTLE und Trajektorien zu Abbildung 8.6(g).

Abbildung 8.6: Clustering und Detektion von fallenden bzw. steigenden Werten im Bouyancy - Datensatz.

endet. Dies verdeutlichen die Particle Tracer in Abbildung 8.7. Zu erkennen ist, dass fast jede Partikelgruppe die gleiche positive Ladung als Ziel hat. Lediglich ein Particle Tracer wurde exakt auf der Grenze gestartet: Bei dieser Partikelgruppe ist in Abbildung 8.7(d) die Aufteilung in unterschiedliche Richtungen zu erkennen. Mit sinkender Distanz zur positiven Ladung wird die Beschleunigung des Partikels immer größer. Dies hat auch einen Anstieg der Separation zur Folge, was in Abbildung 8.8 ersichtlich ist. Bei Abbildung 8.8(h) ist die Division durch Δt bei der Berechnung des FTLE-Wertes unterdrückt. Das Gleichbleiben des FTLE-Wertes nach Erreichen der positiven Ladung wird hierdurch noch besser ersichtlich.

Das Grenzgebiet, in dem sich die Partikel wie in Abbildung 8.7(d) zwischen den beiden positiven Ladungen aufteilen, kann mittels Clustering sehr gut extrahiert werden. Dies ist in Abbildung 8.9 ersichtlich. Das Clustering funktioniert auch bei Veränderung der Größe des Bereiches, auf dem das Clustering angewendet wird. Es wurden 3 Cluster eingesetzt. Auf der zusammenhängenden Fläche des grünen Clusters sind Ausreißer, die dem blauen Cluster zugeordnet wurden. Diese entstehen durch Sprünge im FTLE-Wert, wenn ein Partikel die positive Ladung erreicht. Aufgrund der sehr hohen Strömungsgeschwindigkeit im Datensatz integriert der RK4-Integrator hier teilweise über das Ziel hinaus. Eine Vergrößerung der Schrittweite des Integrators verstärkt diesen Effekt. Die Schrittweite des RK4-Integrators kann allerdings nicht beliebig verringert werden, da die Berechnungszeit entsprechend ansteigt. Die Verwendung eines Integrators mit adaptiver Schrittweite wäre hier vorteilhaft.

Die Detektion der Grenze mittels Clustering ist der Betrachtung des FTLE-Wertes zu einem festen Zeitpunkt überlegen: Die FTLE-Felder mit verschiedenen Integrationslängen werden in Abbildung 8.10 verglichen. Beim Clustering bildet die ridge ein eigenes Cluster. Die ridge kann auch durch Binärisierung des FTLE-Bildes durch Setzen eines Grenzwertes separiert werden. Allerdings ist hierzu die Vorgabe eines Grenzwertes sowie der Integrationslänge durch den Benutzer erforderlich (siehe Abbildung 8.10(e)).

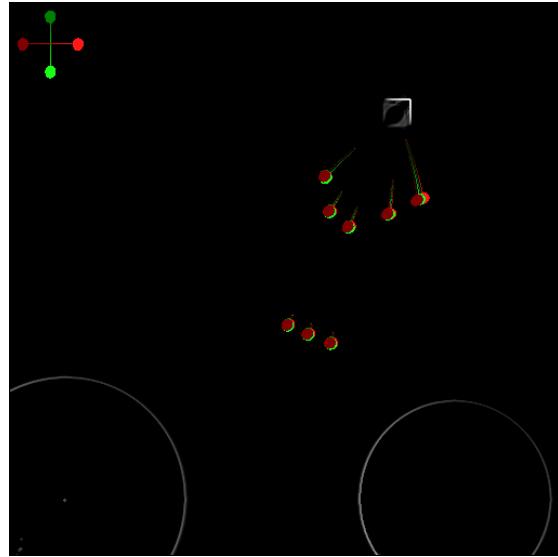
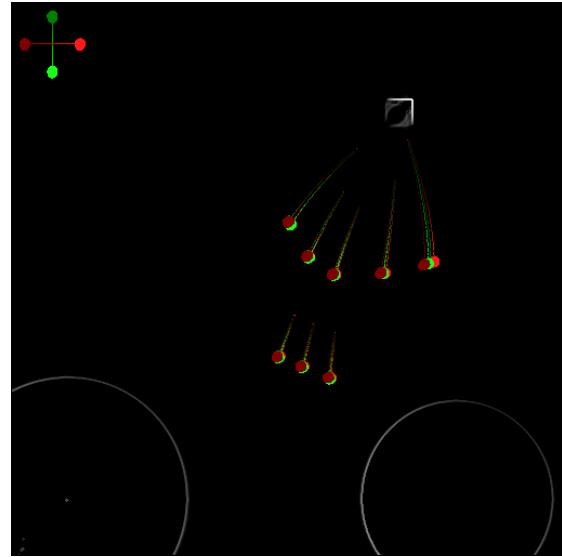
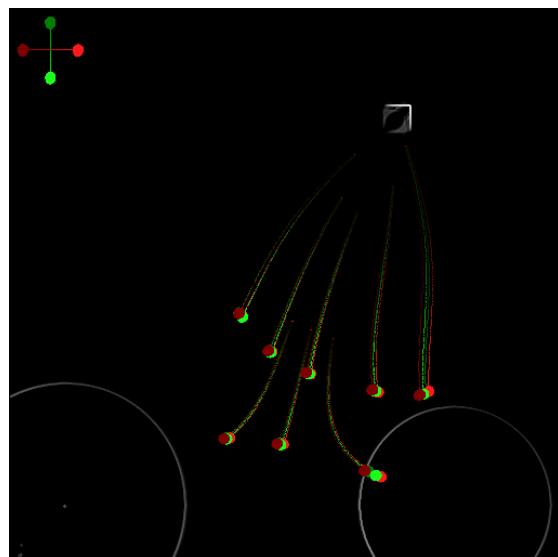
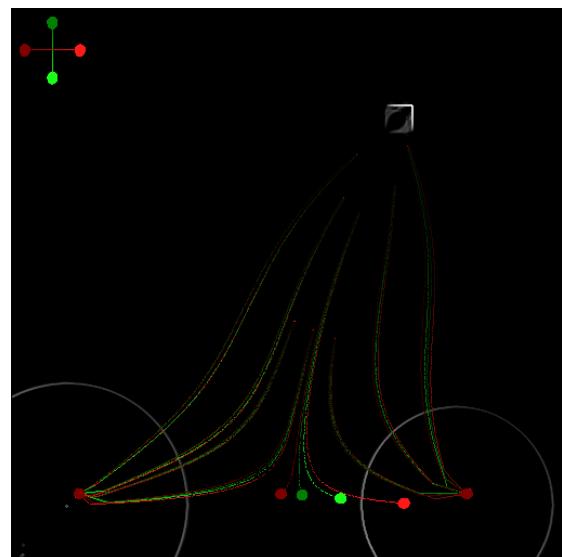
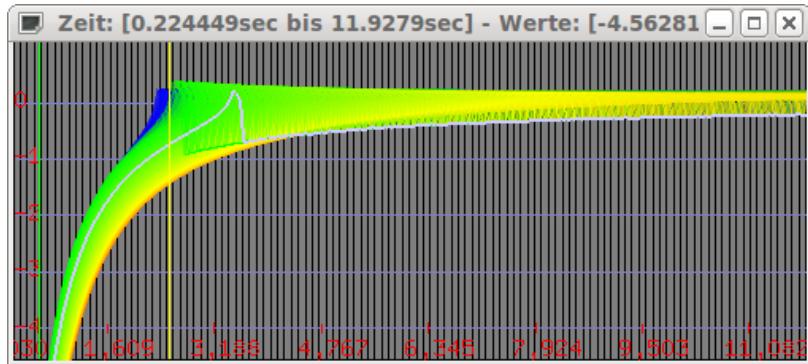
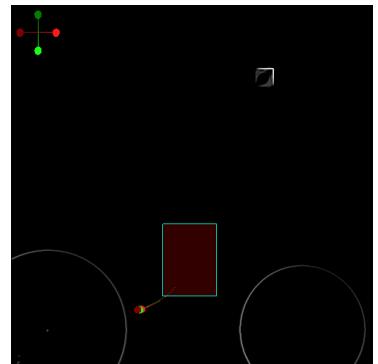
(a) Position der Partikel für $\Delta t = 0.5s$.(b) Position der Partikel für $\Delta t = 2s$.(c) Position der Partikel für $\Delta t = 6s$.(d) Position der Partikel für $\Delta t = 15.1s$.

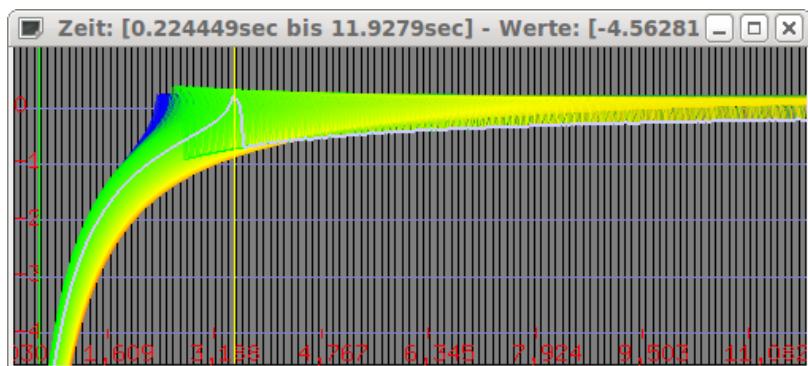
Abbildung 8.7: Trajektorien von Partikeln im Elektro3-Datensatz zu unterschiedlichen Zeitpunkten.



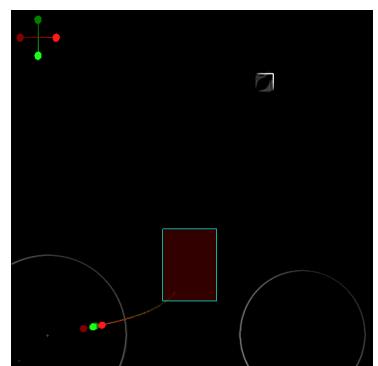
(a) Bei Annäherung an die positive Ladung steigt die Separation.



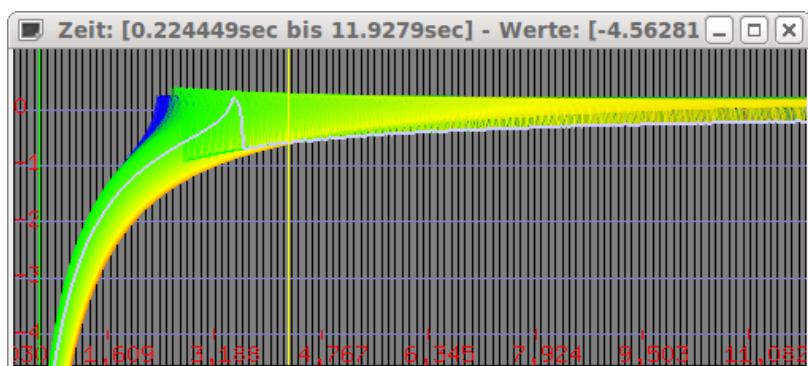
(b) Trajektorien zu Abbildung 8.8(a).



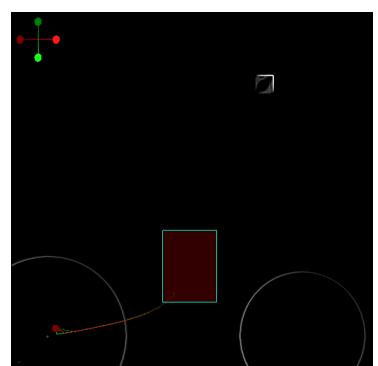
(c) Maximale der Separation kurz vor Erreichen der positiven Ladung.



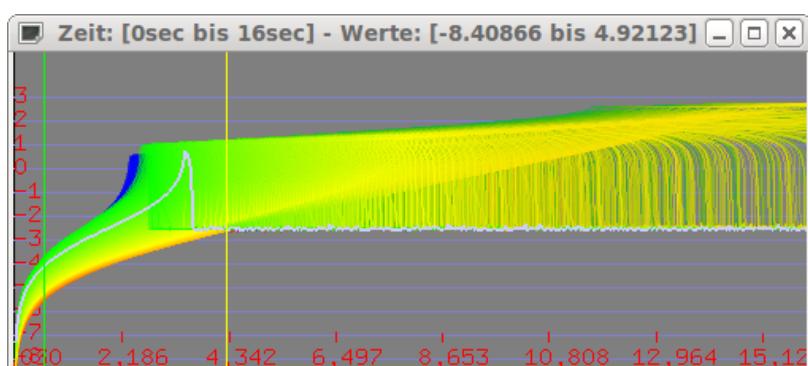
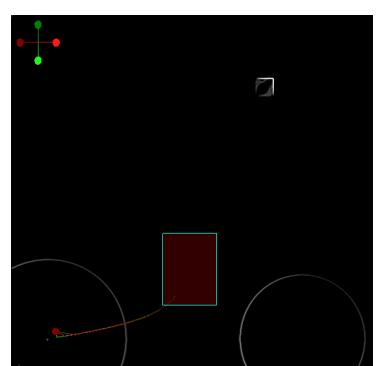
(d) Trajektorien zu Abbildung 8.8(c).



(e) Nach Erreichen des Zentrums der positiven Ladung herrscht geringe Separation.

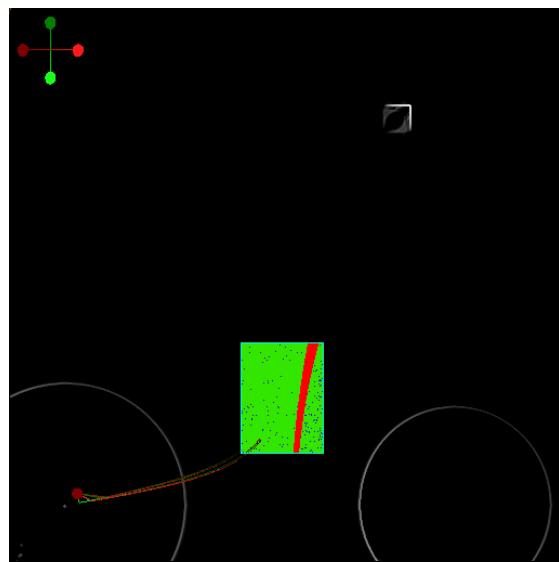


(f) Trajektorien zu Abbildung 8.8(e).

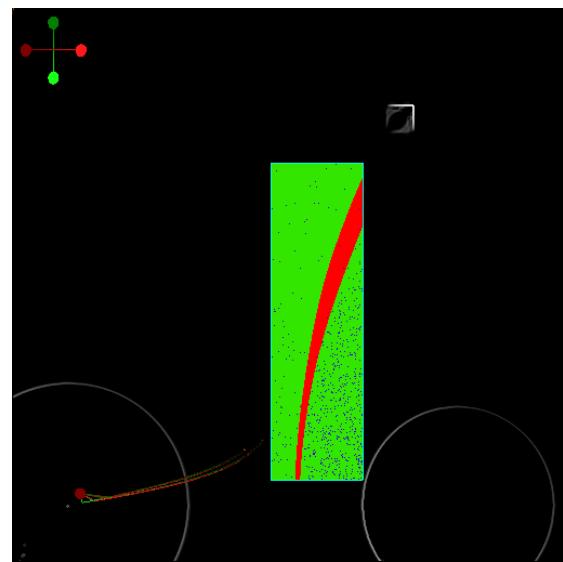
(g) Betrachtung des FTLE-Wertes ohne Division durch Δt .

(h) Trajektorien zu Abbildung 8.8(g).

Abbildung 8.8: Trajektorien von Partikeln im Elektro3-Datensatz zu unterschiedlichen Zeitpunkten.

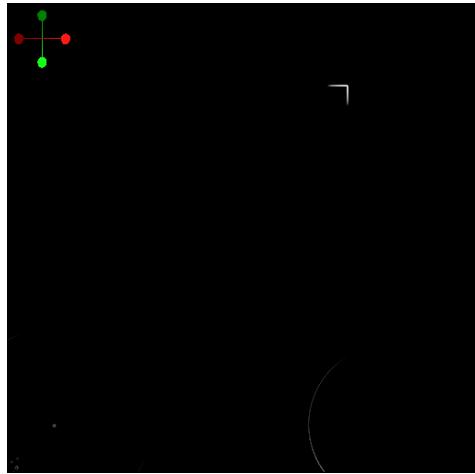


(a) Durch Clustering wird das Grenzgebiet deutlich.

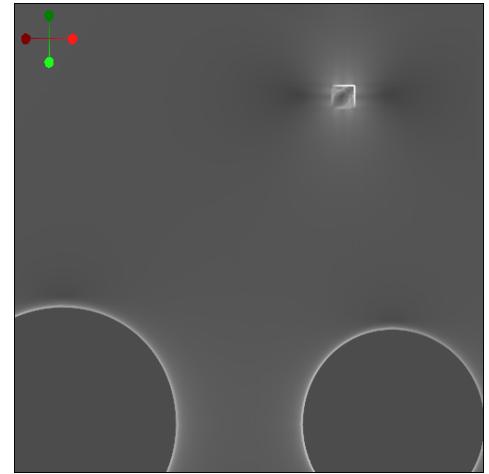


(b) Clustering über ein größeres Gebiet.

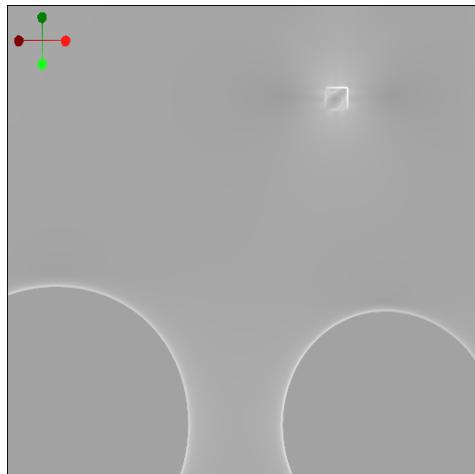
Abbildung 8.9: Detektion des Grenzgebiets durch Clustering.



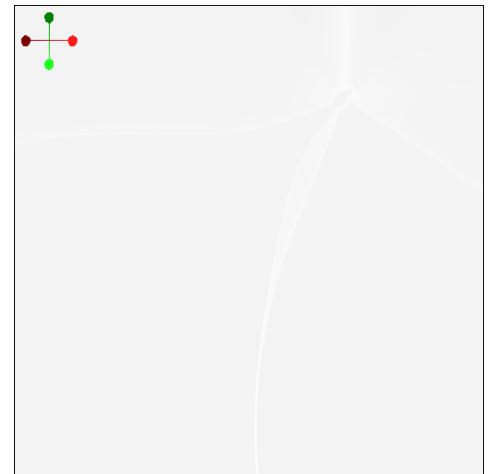
(a) Bei $\Delta t = 0.5s$ sind die ridges um die Pole nur schwach zu erkennen.



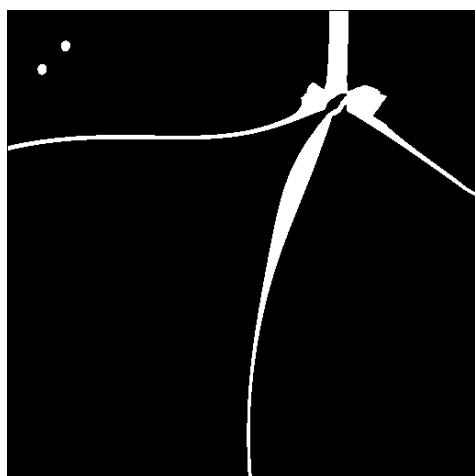
(b) Bei $\Delta t = 0.9s$ sind die ridges noch getrennt.



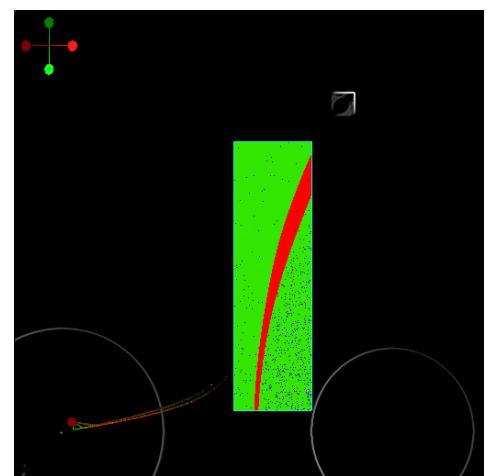
(c) Bei $\Delta t = 3s$ sind die ridges noch getrennt.



(d) Bei $\Delta t = 12s$ berühren sich die ridges, der Kontrast lässt aufgrund der automatischen Normalisierung jedoch stark nach.



(e) Abbildung 8.10(d) wurde mit Hilfe der Schwellwert-Funktion eines Bildbearbeitungsprogramms in Schwarzweiss umgewandelt. Die ridge ist nun deutlich zu erkennen.



(f) Ergebnis des Clusterings aus Abbildung 8.9(b) nochmals zum Vergleich.

Abbildung 8.10: Vergleich der Ausbildung der ridges bei verschiedenen Integrationslängen.

Kapitel 9

Zusammenfassung

Die Betrachtung der zeitlichen Veränderung des FTLE-Wertes erweitert die Visualisierung der Separation nach einem festen Intervall. Bei der Betrachtung des zeitlichen Verlaufs im Diagramm können markante Stellen bereits identifiziert werden. Durch Transformation der Kurven können weitere Features extrahiert werden.

Caching der flow map

Das Caching der flow map stellt einen großen Vorteil dar, da es dem Benutzer ein interaktives Verändern der Integrationslänge erlaubt. Der Benutzer kann sich beim Verändern des Parameters in Echtzeit einen Eindruck über die Veränderung des FTLE-Bildes verschaffen. Nachteilig ist die Berechnungszeit, die einmalig für die Vorberechnung der flow map benötigt wird.

Particle Tracer

Weder aus dem FTLE-Wert noch aus dem zeitlichen Verlauf können Aussagen über die Richtung einer Strömung getroffen werden. Das setzen eines Probepartikels und die Verfolgung dessen Trajektorie gibt diese Information. Die Betrachtung der Nachbarpartikel und deren Separation gibt Aufschluss über die Vorgänge und hilft zur Erkennung der Ursache einer Werteänderung.

Gradient

Die Betrachtung des Gradienten des FTLE-Werteverlaufs unterdrückt den Einfluss unterschiedlicher Startwerte. Dies bietet die Möglichkeit, die Änderung statt dem absoluten Wert zu betrachten und erleichtert die Erkennung dieser. Auch Clustering auf dem Gradienten liefert gute Ergebnisse.

Clustering

Clustering wird eingesetzt, um Bereiche mit ähnlichem Verlauf der FTLE-Werte zu gruppieren. Gerade bei einem Datensatz mit einer scharfen Trennkante (siehe Abschnitt 8.4) konnte diese mittels Clustering besser hervorgehoben werden als durch Betrachtung des FTLE zu einem festen Zeitpunkt. Auf Datensätzen mit vielen Wirbeln und feinen Features liefert Clustering hingegen keine sinnvollen Resultate.

Detektion sinkender Werte

Die Detektion sinkender FTLE-Werte dient der Auffindung von Regionen mit sinkender Separation. Diese treten an Engstellen sowie an der Hinterkante eines modellierten Tragflächenprofils auf. Bei übersichtlichen Datensätzen liefert dies gute Ergebnisse, bei starken Verwirbelungen schwanken die Werte jedoch sehr stark, was zur Folge hat, dass die detektierten Bereiche immer weniger zusammenhängen und zufällig wirken.

Fourier Transformation

Durch die Transformation der Kurven des FTLE-Werteverlaufs ins Frequenzspektrum können periodische Änderungen anhand eines Auftretens einer dominanten Frequenz beobachtet werden. Dies ist jedoch nur bei Betrachtung eines einzelnen Wirbels (zum Beispiel aus dem QuadGyre-Datensatz) der Fall. Bei größeren Betrachtungsbereichen oder anderen Datensätzen ist die Darstellung der Fouriertransformierten zu unübersichtlich, da viele unterschiedlichen Frequenzanteile auftreten.

9.1 Ausblick

Modifikation des Integrators

Bei der Integration der Trajektorien von Partikeln, die sich in der Nähe von Hindernissen befinden bleiben diese aufgrund zu großer Schrittweiten teilweise am Hindernis stehen. Dieses Problem kann durch einen abgewandelten Integrationsalgorithmus gelöst werden: Im Strömungsdatensatz werden Hindernisse durch NaN (not a number) statt dem Nullvektor gekennzeichnet. Dadurch kann dieses bei der Integration erkannt werden. Falls bei der Integration ein Partikel in ein mit NaN belegtes Feld advektiert wird, so muss er auf das nächstgelegene gültige Feld zurückgesetzt werden.

Adaptive Integrationslänge

In dieser Arbeit wird ein Runge-Kutta Integrator 4. Ordnung mit fester Schrittweite verwendet. Die Schrittweite wird vom Benutzer als Parameter vorgegeben. Es hat sich gezeigt, dass die Größenordnung dieses Parameters je nach Datensatz stark variiert. Bei einer adaptiven Integrationslänge muss der Benutzer lediglich einen maximalen Fehler als Parameter vorgeben. Bei Datensätzen, die große Flächen mit geringer Strömung enthalten kann ein Integrationsverfahren mit adaptiver Schrittweite auch Performancevorteile bieten, da weniger Rechenschritte durchgeführt werden.

Weitere Auswertungen auf der Zeitachse

Die hier beschriebene Implementierung detektiert Bereiche anhand festgelegter Kriterien im FTLE-Feld. Eine denkbare Erweiterung ist, die Features bereits vorab auf der Zeitachse zu markieren. Hierbei steht nicht im Vordergrund, an welcher Position im FTLE-Feld das Kriterium erfüllt wird sondern wann auf der Zeitachse. Auch Diagramme, welche die Variation eines Parameters eines Selektionskriteriums der Anzahl selektierter Bereiche gegenüberstellt, können interessante Informationen beinhalten.

Alternative Clustering-Algorithmen

In dieser Arbeit wurde k-Means eingesetzt. Bereits die Modifikation der Abstandsmetrik beeinflusst die Zuordnung der Punkte bzw. Kurven zum jeweiligen Cluster. Auch die Untersuchung mit Clusteringalgorithmen, die unterschiedlich auf Verschiebung oder Streckung entlang der Zeitachse reagieren bieten sich an.

Wavelet-Analyse

Die verwendete Fourier-Transformation dient der Betrachtung im Frequenzbereich, wobei als Basis Sinus- und Cosinusschwingungen verwendet werden. Als Alternative bietet sich die Wavelet-Analyse an. Hier kann die Form der Wavelets auch dem erwarteten Verlauf der FTLE-Werte angepasst werden. Es bieten sich Untersuchungen an, welchen Einfluss das Filtern bestimmter Frequenzbereiche hat und ob so zum Beispiel das Rauschen beim Clustering oder der Detektion steigender oder fallender Werte verringert werden kann.

Anwendung in der Physik

Bei dieser Auswertung wurden Datensätze erstellt, die bestimmte Eigenschaften wie fallende FTLE-Werte oder zyklische Muster aufweisen. Diese Datensätze sind jedoch stark idealisiert. Um den Nutzen in realistischer Umgebung prüfen zu können sollten Datensätze physikalischen Ursprungs wie zum Beispiel Messungen aus einem Windkanal verwendet werden. Hier muss untersucht werden, inwiefern sich Ungenauigkeiten bei der Messwerterfassung und der Auftritt gesuchter FTLE-Werteänderungen auseinanderhalten lassen.

Literaturverzeichnis

- [AGDJ09] ANDERSON, J. C.; GOSINK, L. J.; DUCHAINEAU, M. A. ; JOY, K. I.: Interactive Visualization of Function Fields by Range-Space Segmentation. In: *Computer Graphics Forum* 28 (2009), Nr. 3, 727–734. <http://dx.doi.org/10.1111/j.1467-8659.2009.01480.x>. – DOI 10.1111/j.1467-8659.2009.01480.x. – ISSN 1467–8659
- [AID11] AID AIRFOIL INVESTIGATION DATABASE: *naca-6412*. Website, 2011. – Available online at <http://www.worldofkrauss.com/foils/1417>
- [AMD10a] AMD ADVANCED MICRO DEVICES, INC.: *AMD OpteronTM Processor Solutions*. Website, 2010. – Available online at <http://www.amd.com/us/products/server/processors/6000-series-platform/pages/6000-series-platform.aspx>
- [AMD10b] AMD ADVANCED MICRO DEVICES, INC.: *ATI FireProTM V9800 Data Sheet*. Website, 2010. – Available online at http://www.amd.com/de/Documents/ATI_FirePro_V9800_DataSheet.pdf
- [BR10] BRUNTON, S. L.; ROWLEY, C. W.: *Fast computation of finite-time Lyapunov exponent fields for unsteady flows*. CHAOS 20, 017503, 2010
- [cud10] NVIDIA CUDA C Programming Guide Version 3.2. 09 2010
- [FJ09] FRIGO, M.; JOHNSON, S. G.: *FFTW - Fastest Fourier Transform in the West*. WEBSITE, 2009. – Available online at <http://www.fftw.org/>
- [Fro07] FROEHLICH, L.: *FFT Crash Course*. WEBSITE, 2007. – Available online at tesla.desy.de/~lfroehli/tutorial/download/fourier.pdf
- [GLT⁺] GARTH, C.; LI, G.-S.; TRICOCHE, X.; HANSEN, C. D. ; HAGEN, H.: *Visualization of Coherent Structures in Transient 2D Flows*. University of Kaiserslautern, University of Utah, Springer-Verlag,
- [Hal01] HALLER, G.: Distinguished material surfaces and coherent structures in three-dimensional fluid flows. In: *Phys. D* 149 (2001), March, 248–277. [http://dx.doi.org/10.1016/S0167-2789\(00\)00199-8](http://dx.doi.org/10.1016/S0167-2789(00)00199-8). – DOI 10.1016/S0167-2789(00)00199-8. – ISSN 0167-2789

- [Hal02] HALLER, G.: Lagrangian coherent structures from approximate velocity data. 14 (2002), Nr. 6, 1851–1861. <http://dx.doi.org/DOI:10.1063/1.1477449>. – DOI DOI:10.1063/1.1477449. – ISSN 10706631
- [HP98] HALLER, G.; POJE, A.C.: *Finite-time transport in aperiodic flows*. Division of Applied Mathematics, Brown University, 1998
- [HSW11] HLAWATSCH, M.; SADLO, F. ; WEISKOPF, D.: Hierarchical Line Integration. In: *IEEE Transactions on Visualization and Computer Graphics* 17 (2011), Nr. 8 , S. 1148–1163
- [HVS11] HLAWATSCH, M.; VOLLRATH, J. E.; SADLO, F. ; WEISKOPF, D.: Coherent Structures of Characteristic Curves in Symmetric Second Order Tensor Fields. In: *IEEE Transactions on Visualization and Computer Graphics* 17 (2011), Nr. 6 , S. 781–794
- [HY00] HALLER, G.; YUAN, G.: Lagrangian coherent structures and mixing in two-dimensional turbulence. In: *Phys. D* 147 (2000), December, 352–370. [http://dx.doi.org/10.1016/S0167-2789\(00\)00142-1](http://dx.doi.org/10.1016/S0167-2789(00)00142-1). – DOI 10.1016/S0167-2789(00)00142-1. – ISSN 0167-2789
- [Ihl11] IHLENBURG, F.: *Kinematik großer Verformungen*. Website, 04 2011. – Available online at www.mp.haw-hamburg.de/pers/Ihl/HFEM/PDF/kinematik_3D.pdf
- [Ill05] ILLINOIS INSTITUTE OF TECHNOLOGY: *Lagrangian Coherent Structures - Analysis of time-dependent dynamical systems using finite-time Lyapunov exponents*. Website, 04 2005. – Available online at <http://mmae.iit.edu/shadden/LCS-tutorial>
- [Int10] INTEL CORPORATION: *Intel microprocessor export compliance metrics*. Website, 2010. – Available online at <http://www.intel.com/support/processors/sb/cs-023143.htm>
- [JMF99] JAIN, A.K.; MURTY, M.N. ; FLYNN, P.J.: *Data clustering: a review*. ACM Computing Surveys, Vol. 31, No. 3, 1999
- [JV09] JIMENEZ, R.; VANKERSCHAVER, J.: *Optimization of FTLE Calculations Using nVidia's CUDA*. California Institute of Technology, 2009
- [Kar10] KARCH, G. K.: *GPU-based acceleration of selected clustering techniques*, Diplomarbeit, 2010
- [KR10] KATIA, M.Sc.; ROCHA, M.C.: *Numerical Techniques for Real Options*. Website, 2010. – Available online at <http://www.puc-rio.br/marco.ind/katia-num.html#finite-differences>
- [LL] LEEUW, W.; LIERE, R.: *Visualization of Global Flow Structures using Multiple Levels of Topology*. Center for Mathematics and Computer Science CWI,
- [Mac67] MACQUEEN, J. B.: Some Methods for Classification and Analysis of MultiVariate Observations. In: CAM, L. M. L. (Hrsg.); NEYMAN, J. (Hrsg.): *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability* Bd. 1, University of California Press, 1967, S. 281–297

- [MCDZL05] MA, P.; CASTILLO-DAVIS, C. I.; ZHONG, W. ; LIU, J. S.: *Curve Clustering to Discover Patterns in Time Course Gene Expression Data*. Department of Statistics, Harvard University, 2005
- [MIT10] MIT OPENCOURSEWARE: *8.02SC Physics II: Electricity and Magnetism*. Website, 2010. – Available online at http://ocw.mit.edu/courses/physics/8-02sc-physics-ii-electricity-and-magnetism-fall-2010/electric-fields-and-discrete-charge-distributions/MIT8_02SC_notes3to4.pdf
- [Nat04] NATIONAL UNIVERSITY OF SINGAPORE: *Gradient and Grid Perturbation*. Website, 2004. – Available online at <http://www.ims.nus.edu.sg/Programs/wbfst/files/siva6.pdf>
- [NIS] NIST REFERENCE ON CONSTANTS, UNITS AND UNCERTAINTY: *Fundamental Physical Constants - elementary charge e*. – Available online at <http://physics.nist.gov/cgi-bin/cuu/Value?e>
- [NM08] NEUWEILER; MILBRADT: *Finite Differenzen Methode*. Leibniz Universität Hannover, Institut für Bauinformatik, 2008
- [PP09] PAPAMICHAIL, G. P.; PAPAMICHAIL, D. P.: *The k-means range algorithm for personalized data clustering in e-commerce*. Eurographics/ IEEE-VGTC Symposium on Visualization, 2009
- [PTVF] PRESS, W. H.; TEUKOLSKY, S. A.; VETTERLING, W. T. ; FLANNERY, B. P.: *Numerical Recipes in C: The art of scientific computing*, ISBN 0-521-43108-5. Cambridge University Press,
- [SLM05] SHADDEN, S. C.; LEKIEN, F. ; MARSDEN, J. E.: Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows. In: *Physica D: Nonlinear Phenomena* 212 (2005), Nr. 3-4, 271–304. <http://linkinghub.elsevier.com/retrieve/pii/S0167278905004446>
- [Spo09] SPOOREN, F.: *Finale Geforce GTX 275-Spezifikationen*. Website, 2009. – Available online at <http://www.hardware-infos.com/news.php?news=2822>
- [SRP09] SADLO, F.; RIGAZZI, A. ; PEIKERT, R.: Time-Dependent Visualization of Lagrangian Coherent Structures by Grid Advection. In: *Proceedings of TopoInVis 2009 (to appear)*, Springer, 2009
- [STW⁺08] SHI, K.; THEISEL, H.; WEINKAUF, T.; HEGE, H.-C. ; SEIDEL, H.-P.: Finite-Time Transport Structures of Flow Fields. In: *Proc. IEEE Pacific Visualization 2008*. Kyoto, Japan, March 2008, 63–70
- [SW10] SADLO, F.; WEISKOPF, Daniel: Time-Dependent 2-D Vector Field Topology: An Approach Inspired by Lagrangian Coherent Structures. In: *Computer Graphics Forum* 29 (2010), Nr. 1, S. 88–100

- [TSM] TARJAN, D.; SKADRON, K. ; MICIKEVICIUS, P.: *The Art of Performance Tuning for CUDA and Manycore Architectures.* WEBSITE, . – Available online at <http://www.docstoc.com/docs/21525105/The-Art-of-Performance-Tuning-for-CUDA-and-Manycore>

Abbildungsverzeichnis

3.1	Veranschaulichung der Separation anhand der Trajektorie von Partikeln.	7
3.2	Für die Berechnung der finiten Differenzen gewählter stencil.	9
4.1	Koordinatentransformation beim Zugriff auf die Strömungsdaten-Textur.	12
4.2	Textur-Clamping beim Strömungsdaten-Zugriff.	12
4.3	Berechnung des globalen Minimums/Maximums auf der GPU.	13
4.4	Aufbau der gecacheten flow map	14
4.5	Vergleich unterschiedlicher RK4-Schrittweiten.	15
4.6	Berechnung der Diagrammdaten.	17
5.1	Vorgabe des Bereichs, über den der zeitliche Verlauf des FTLE-Wertes dargestellt wird. .	19
5.2	Betrachtung eines Ausschnitts mittels Zoom-Funktion.	20
5.3	Unterschiedliche Modi: FTLE-Wert, Gradient oder Fourier-Transformierte.	21
5.4	Features der Diagrammdarstellung.	21
6.1	Diagramm-Darstellung des FTLE-Werteverlaufs.	24
6.2	Überprüfen der Oberfläche.	26
6.3	Vergleich unterschiedlicher RK4-Schrittweiten.	27
6.4	Flowmap: Diskretisierungsproblem und Lösung durch Relativkoordinaten.	27
7.1	Vergleich der Kurvenschaar mit und ohne Division durch Δt	30
7.2	Betrachtung des Gradienten der Diagrammkurven.	31
7.3	Betrachtung des Gradienten der Diagrammkurven.	32
7.4	Clustering auf FTLE-Werten und deren Gradienten.	33
7.5	Gradienten der in Abbildung 7.1 gewählten Region Kurven Δt	34
7.6	Diagramm-Darstellung des FTLE-Werteverlaufs.	36
7.7	Tracen von Partikeln in den Clustern aus Abbildung 7.6(e) bzw. 7.6(f).	37

7.8	Variation des Zeitbereichs, auf dem geclustert wird.	37
7.9	Untersuchung der Clusteraufteilung durch Verfolgung von Partikeln.	39
7.10	Durchführung des Clusterings auf einem größeren Intervall von $\Delta t = 0.3s$ bis $\Delta t = 0.8s$	40
7.11	Vergleich des FTLE-Bildes zwischen originalem und modifiziertem Datensatz.	41
7.12	4 Cluster.	42
7.13	10 Cluster.	42
7.14	Detektion fallender FTLE-Werte.	44
7.15	Betrachtung von Gebieten, deren FTLE-Wert über ein Intervall von 10 slices (0.002 Sekunden) um den vorgegebenen Betrag über dem Minimum der jeweiligen Kurve liegt.	45
7.16	Wie Abbildung 7.15, jedoch Vergrößerung des zu erfüllenden Intervalls auf 30 slices.	46
7.17	FTLE-Bild mit selektiertem Bereich. Blau markiert ist das Gebiet, indem das Kriterium (angestiegener FTLE-Wert) zutrifft. Unterhalb des Flügels wurde die Trajektorie einer Partikelgruppe gezeichnet	47
7.18	Diagramm der selektierten Kurven bei richtiger Begrenzung von Δt : die gelbe Linie hinter dem selektierten Bereich zeigt die aktuelle Position des in Abbildung 7.17 gezeigten Partikels.	47
7.19	Der selektierte Bereich sowie die Partikel-Startposition entspricht der Konfiguration in Abbildung 7.17. Die Endposition der verfolgten Partikel liegt jedoch bereits außerhalb des Datensatzes.	48
7.20	Diagramm der selektierten Kurven zu Abbildung 7.19.	48
7.21	Verknüpfung der Detektion abfallender und ansteigender Werte.	50
7.22	Drehrichtung und Umlaufzeiten eines Wirbels des QuadGyre-Datensatzes.	51
7.23	Untersuchung des Diagramms eines Wirbels.	51
7.25	Betrachtung der Fourier-Transformierten.	52
7.26	Betrachtung der Fouriertransformierten und Clustering.	53
7.24	Untersuchung der Tiefpunkte im Diagramm.	54
8.1	Positionierung der Punktladungen bezüglich des Strömungsdatensatzgitters.	57
8.2	Partial Differential Equation Toolbox in Matlab: grafische Eingabe und Visualisierung von partiellen Differentialgleichungen.	59
8.3	Trajektorie von Partikeln im Circle-Datensatz.	62
8.4	Clustering im Circle-Datensatz.	63
8.5	FTLE-Werte und deren Gradienten in Gebieten des Bouyancy - Datenstzes.	65
8.6	Clustering und Detektion von fallenden bzw. steigenden Werten im Bouyancy - Datensatz.	66
8.7	Trajektorien von Partikeln im Elektro3-Datensatz zu unterschiedlichen Zeitpunkten.	68
8.8	Trajektorien von Partikeln im Elektro3-Datensatz zu unterschiedlichen Zeitpunkten.	69

8.9 Detektion des Grenzgebiets durch Clustering.	70
8.10 Vergleich der Ausbildung der ridges bei verschiedenen Integrationslängen.	71

Anmerkung zu Bildern

Die in dieser Arbeit abgedruckten Bilder sind für die Darstellung am Monitor optimiert. Die sehr feinen Strukturen sehen beim Druck oft verschwommener aus, als diese am Monitor erscheinen. Die Auflösung der im PDF eingebetteten Bilder wurde so gewählt, dass diese bei 400% Vergrößerung betrachtet werden können.

Erklärung

Hiermit versichere ich, diese Arbeit
selbstständig verfasst und nur die
angegebenen Quellen benutzt zu haben.

(Tom Schilli)