Physics-based velocity field simplification for flow visualization

A. Pobitzer
University of Bergen, Norway

Minisymposium on Analysis and Representation of Large Data Sets
E.T.S.I. Aeronáuticos, Madrid 15/02/2012

- From an introduction to continuum mechanics…

Past and present

Traditional modeling	Computational modeling
model formulation governed by available analytical solution techniques	model formulation governed by numerical solution methods and available software
tendency towards over-simplified models because of limited solution techniques	more general models for phenomena of more industrial/scientific relevance
often sloppy treatment of initial and boundary conditions	numerics demands complete specification of initial-boundary value problems
result as symbolic expressions	result as visualizations

[H.P. Langtangen
J. Sundnes 2012]

CFD and Visualization

- From an introduction to continuum mechanics…

Past and present

Traditional modeling	Computational modeling
model formulation governed by available analytical solution techniques	model formulation governed by numerical solution methods and available software
tendency towards over-simplified models because of limited solution techniques	more general models for phenomena of more industrial/scientific relevance
often sloppy treatment of initial and boundary conditions	numerics demands complete specification of initial-boundary value problems
result as symbolic expressions	result as visualizations

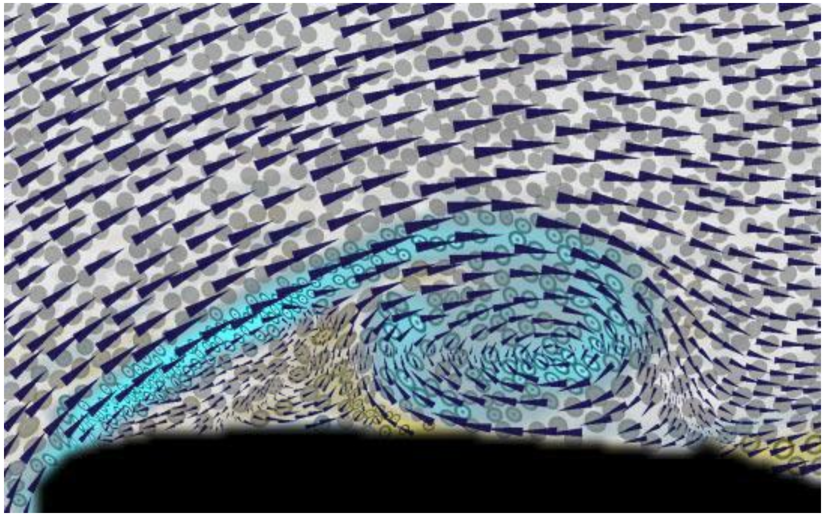
↓ result = symbolic expression

↓ visualization = velocity field + X

[H.P. Langtangen J. Sundnes 2012]

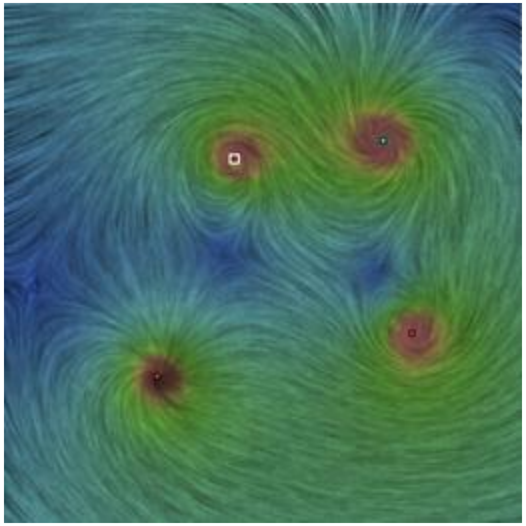
X = Visualization Techniques

- Direct flow visualization



[Kirby et al. 1999]

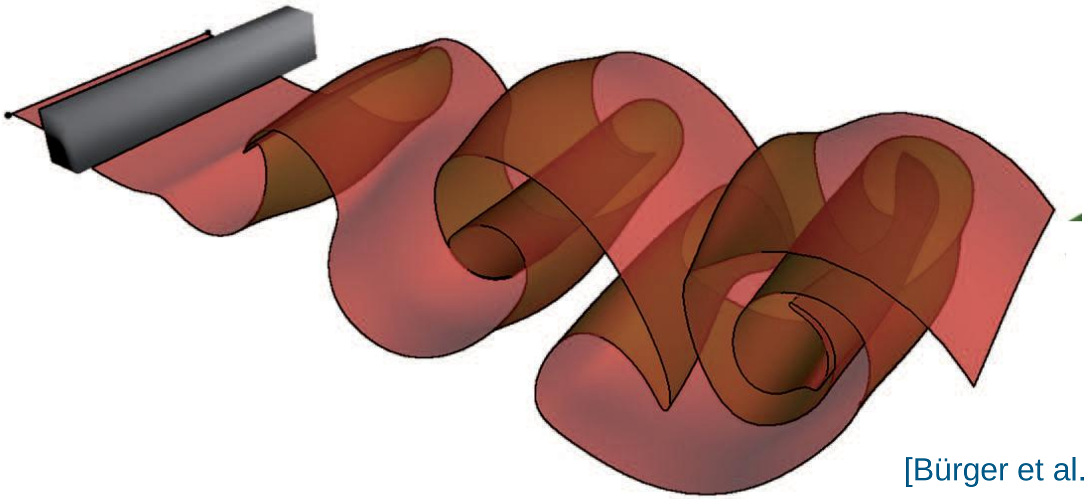
- Dense, texture-base flow visualization



[v. Wijk et al. 1999]

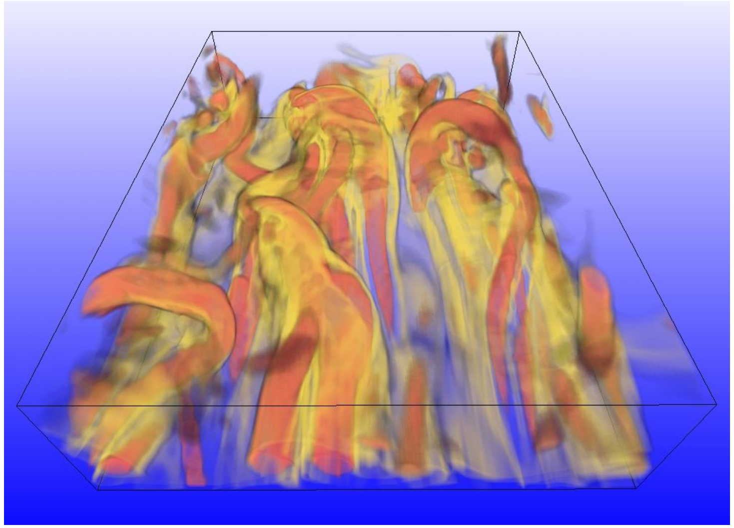
X = Visualization Techniques

- Integration-based flow visualization



[Bürger et al. 2009]

- Feature-based flow visualization



[Helgeland et al. 2007]

X = Feature extraction + Postprocessing

- Complex flow patterns (e.g., turbulence) lead to rich output of feature detectors → often simplification wanted
- Usual simplification works like this
  - extract all features according to a certain detector
  - remove according to geom. criteria (vicinity, size, length,…)
  - problem: immediate relation to underlying flow patterns is destroyed

X = Preprocessing + Feature extraction

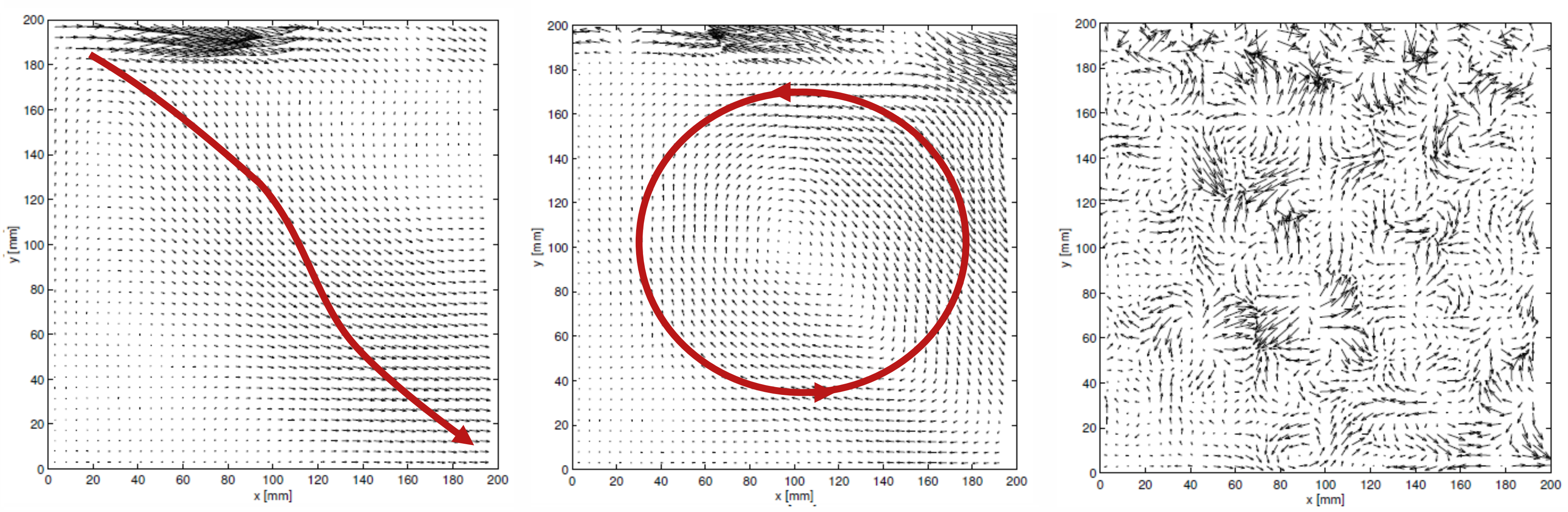
- Conserves 1 to 1 relationship, but…
- Does the preprocessed velocity field describe the same physical process as the original?
  - e.g. still incompressible fluid?
  - boundary conditions?
  - …
- Smoothing the field (low pass filtering) can be problematic [Velte et al. 2010]
  - changes the data
  - destroys the physical scales

Outline

1. "Physics based velocity field simplification…"
    1. turbulence energy cascade
    2. main idea of our approach
    3. computational aspects
2. "… for flow visualization"
    1. application for feature extraction
    2. examples for important classes of feature detectors
    3. discussion of results

Velocity field as Superposition of Scales

- Turbulence energy cascade
  - velocity field = superposition of energy-scales
  - different energy-scales = different roles in the flow
    - scales are ranked by turbulence kinetic energy
    - transport to dissipation



[Pedersen 2003]

Physics-based simplification

- Main idea: removing some low TKE scales
  - should capture physics of the flow “at large”…
  - but remove details that are unimportant for the chosen level of observation
- Advantages:
  - energy-scales describe all the same physical process
  - turbulence energy cascade depends on problem only (no user parameters to choose)
- Open questions (for now):
  - how to compute energy-scales?
  - how to remove low TKE energy-scales?

Computing the Energy-scales

- Proper orthogonal decomposition (POD)
  - Lumley: Introduced of POD in 1967
    - continuous formulation as Fredholm integral equation
    - rather theoretical value
  - Sirovich: discrete (snapshot) formulation in 1987
    - applicable to measurements
    - with today's computing power: also large DNS
- Technically (discrete formulation):
  - Find orthonormal basis of the function space spanned by the measured/simulated time steps

Practical POD computation [Sirovich 1987]

- For data $\mathbf{u} = (\mathbf{u}(\mathbf{x}, t_n))_{n=0}^{N}$, the equations to solve are given by

$$\sigma^i = \underset{\sigma \in L^2 \setminus \langle \sigma^1, \ldots, \sigma^{i-1} \rangle}{\arg\max} \frac{1}{N} \sum_{n=1}^{N} \langle \sigma, \mathbf{u}_n \rangle^2 \quad \text{with} \quad \mathbf{u}_n(\mathbf{x}) = \mathbf{u}(\mathbf{x}, t_n)$$

- Equivalent to eigenvalue problem

$$\sum_{m=1}^{N} C_{n,m} a_m^{(i)} = \lambda_{(i)} a_n^{(i)} \quad \text{with} \quad C_{n,m} = \frac{1}{N} \langle \mathbf{u}_n, \mathbf{u}_m \rangle$$

- Then $\sigma^i = \dfrac{\sum_{n=1}^{N} a_n^i \mathbf{u}_n}{|| \sum_{n=1}^{N} a_n^i \mathbf{u}_n ||}$

Properties of POD modes

- Every snapshot has the exact representation

$$\mathbf{u}_n(\mathbf{x}) = \sum_{i=1}^{N} \left\langle \mathbf{u}_n, \sigma^i \right\rangle \sigma^i(\mathbf{x})$$

- Every mode σ^i
  - fulfills boundary conditions (periodic, heterogeneous,…)
  - Same physical properties as orig. flow (e.g., divergence-freeness,…)
    [Sirovich 1987]
- The relative TKE a mode accounts for is given by

$$\frac{\lambda_i}{\sum_{i=1}^{N} \lambda_i}$$

Usage for feature detection

- Truncated reconstructions = removal of the respective energy-scales
- Choose the relative amount of TKE to include, i.e. i_p such that $(\sum_{i=1}^{i_p} \lambda_i)/(\sum_{i=1}^{N} \lambda_i) \leq p$ for a chosen p
- Then

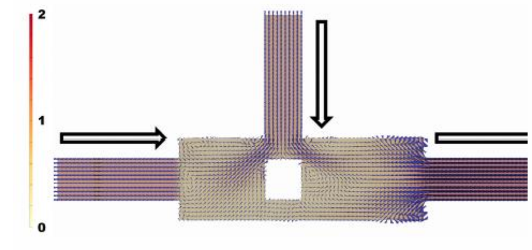
$$\mathbf{u}_p(\mathbf{x},t) = \sum_{i=1}^{i_p} \left\langle \mathbf{u}_t, \sigma^i \right\rangle \sigma^i(\mathbf{x})$$

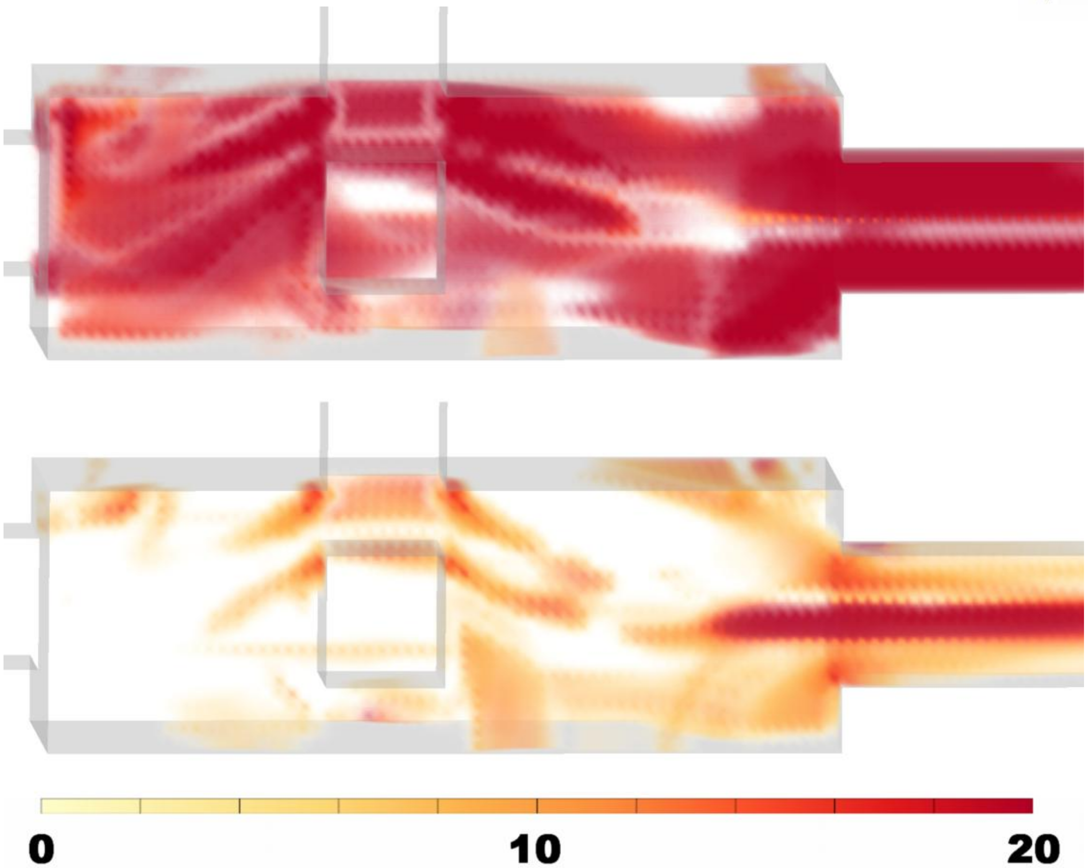
  includes automatically the right amount of energy-scales *("…as simple as possible, but no simpler." Einstein's razor)*
- Apply feature detection on $\mathbf{u}_p(\mathbf{x},t)$

Feature detectors

- Two important classes
  - Local feature detection (Eulerian)
  - Integration based feature detection (Lagrangian)
- Examples for Application
  - vorticity thresholding
  - finite-time Lyapunov exponents (FTLE)

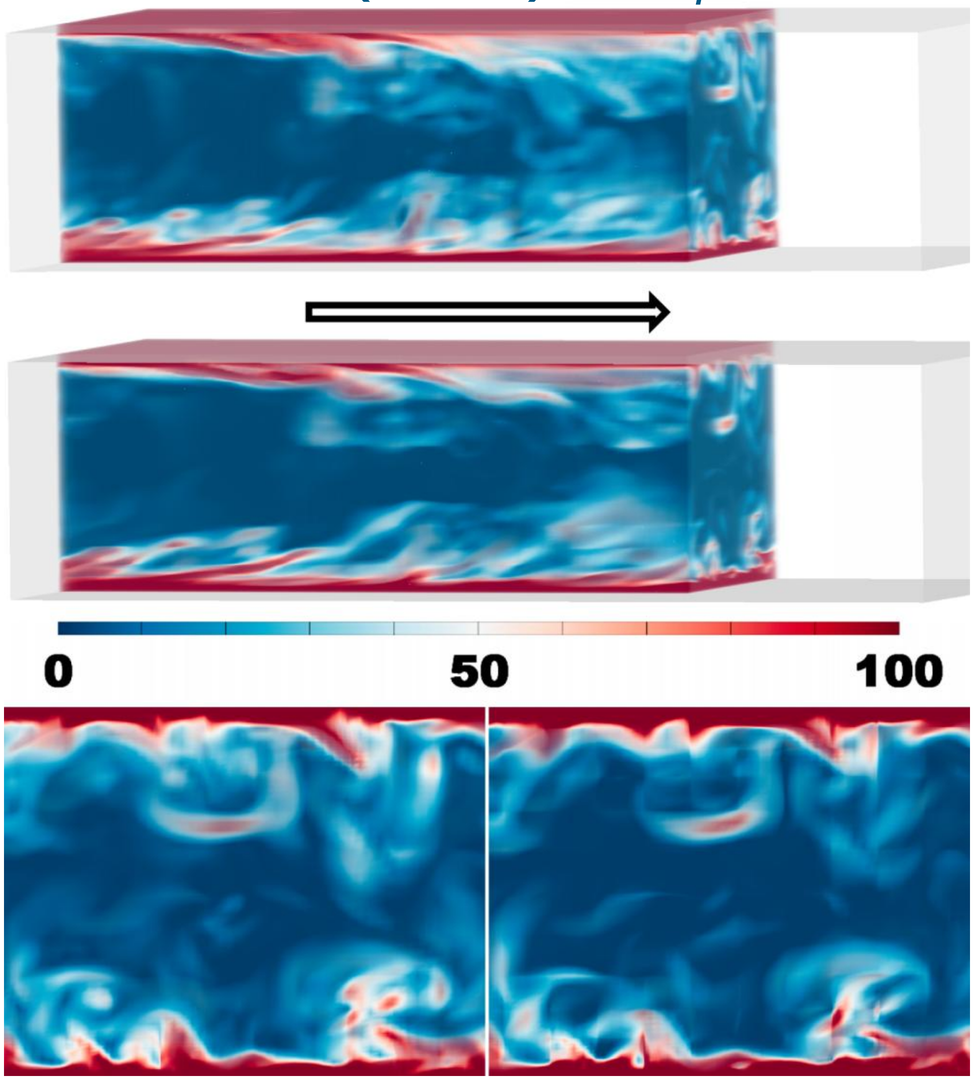
- Flow in a T-junction
  - vorticity: $\omega = \nabla \times \mathbf{v}$





Examples: vorticity

- Turbulent channel flow (DNS), $Re_\tau=180$

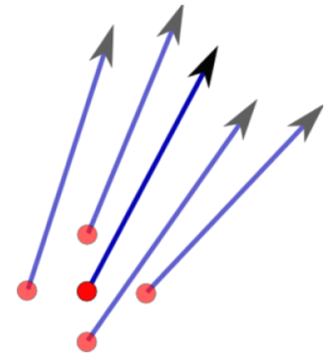


Basic Concept of FTLE

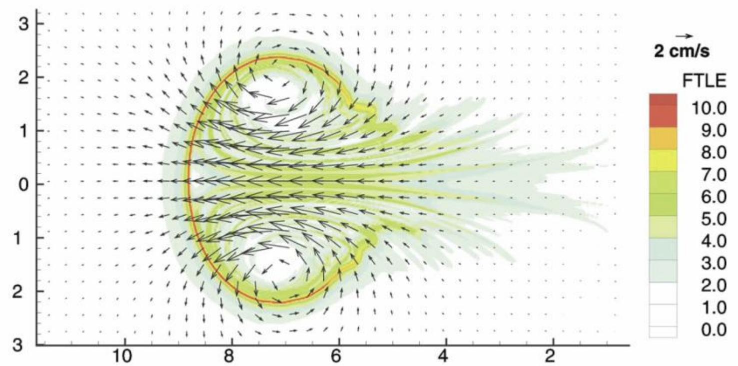
- Measure for separation rate of particles
  - Assumption: separation after advection = initial separation x exp(rate * integration time)
  - Hence: maximal separation rate = log(maximal sep. after advection/initial sep.)
- For relatively long integration times, we expect rather large deformations
- Measure of large deformation: (right) Cauchy-Green tensor
- Maximal deformation rate of infinitesimal sphere = largest eigenvale of Cauchy-Green tensor
  (largest singular value of flow map gradient)

FTLE in velocity fields

- Practical computation
  - seed particles to estimate flow map/displacement gradient F
  - Calculate Cauchy-Green tensor $\mathrm{C} = \mathrm{F}^{\mathrm{T}}\mathrm{F}$
  - the FTLE value is then



$$\mathrm{FTLE} = \frac{\log\left(\sqrt{\lambda_{\max}(\mathrm{C})}\right)}{|t - t_0|}$$



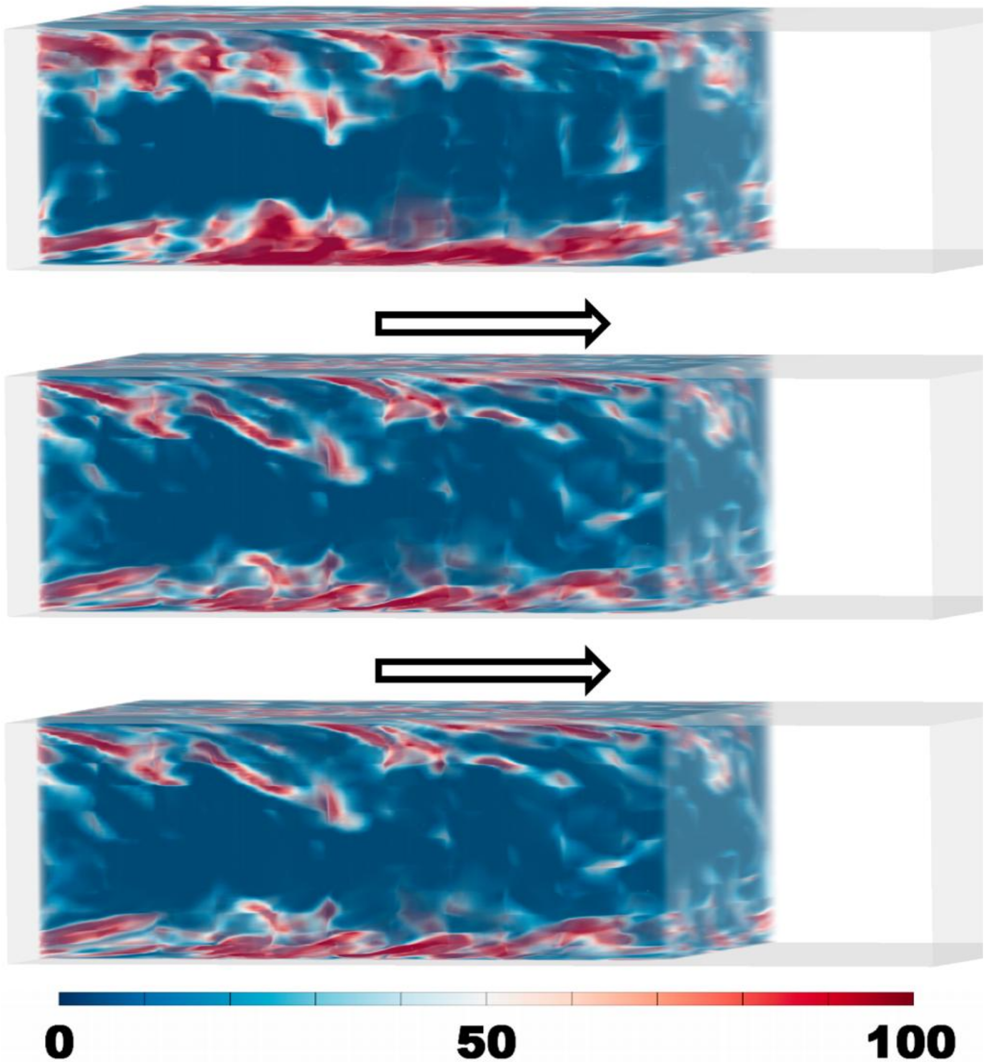
[Shadden 2006]

- FTLE ridges ~ Lagrangian coherent structures

[Haller 2001]

Example: FTLE

- Turbulent channel flow (DNS)



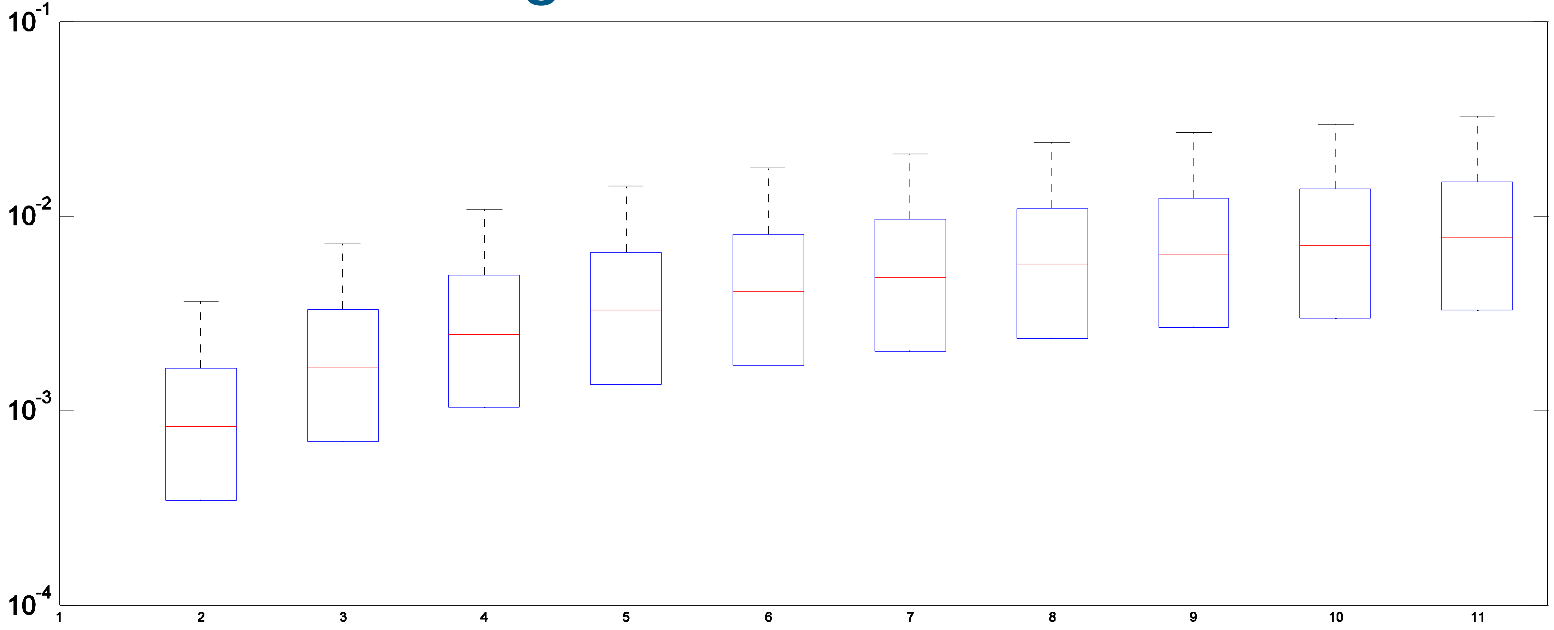
FTLE of orig. field

FTLE of recon.
Based on 2 modes
(79% of TKE)

FTLE of recon.
Based on 4 modes
(93% of TKE)

Integration error analysis

- Relative error integration in reconstructed field



- For longer integration times, large energy-scales determinant (transport)
- Small energy-scales introduce noise in gradient computation

Conclusion

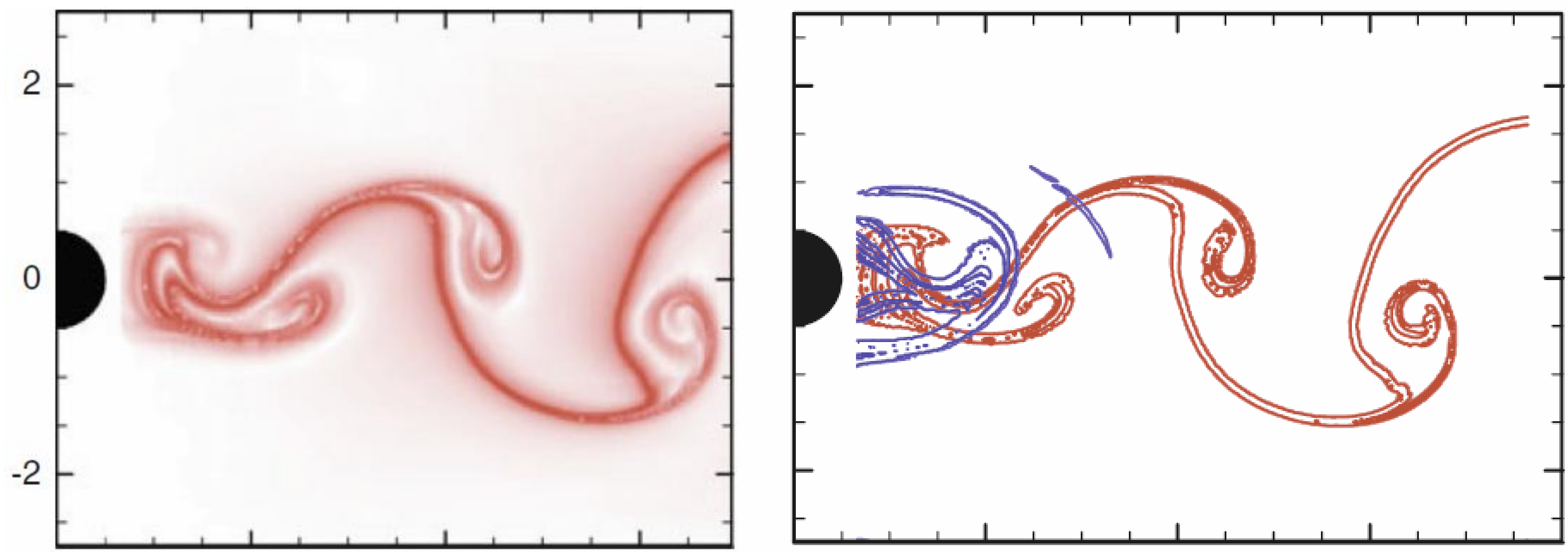
- POD-based feature extraction
  - simplification without changing physics
  - simplification without user parameters
  - visualization = velocity fields
  - can uncover hidden feature
  - denoises output if removed energy-scales do not related to observed phenomenon

Current Limitations

- At low Reynolds numbers, bad separation of scales
  - One energy-scale “smeared out” over several POD modes
  - No cascade, but “slide”, where to truncate?
- If too few snapshots available, multiple energy-scales in one POD mode
- Problems with not statistically converged flow (e.g., accelerating flow)
- At current no systematic testing of behaviour with larger diversity of feature detectors

Remark on LCS and POD

- Conclusions similar to the here presented for experimental data in [L. Kourentis, E. Konstantinidis: **Uncovering large-scale coherent structures in natural and forced turbulent wakes by combining PIV, POD, and FTLE, Experiments in Fluids** (2011)]



Acknowledgments

- Special thanks to:
  Murat Tutkun (Norwegian Defence Research Establishment, University of Lille - Nord de France) and **Øyvind Andreassen** (Norwegian Defence Research Establishment), **Raphael Fuchs** and **Ronald Peikert** (ETH Zürich), and **Helwig Hauser** (Universtiy of Bergen)
- **Carl Erik Wasberg** (Norwegian Defence Research Establishment) and **AVL List** GmbH, Austria
- **SimVis** GmbH, Austria

The project SemSeg acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number 226042.

Thanks for your attention!

Questions?

Based on [A. Pobitzer, M. Tutkun, Ø. Andreasen, R. Fuchs, R. Peikert, and H. Hauser: **Energy-scale Aware Feature Extraction for Unsteady Flow Visualization, Computer Graphics Forum**(2011)]