Piecewise constant vector field topology

Andrzej Szymczak

Colorado School of Mines

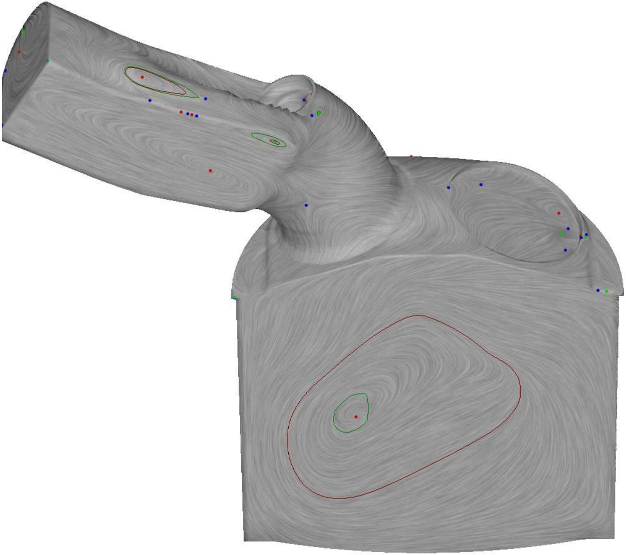
July 5, 2012

- Robust Morse Decompositions of Piecewise Constant Vector Fields, A.S and E. Zhang, IEEE TVCG 18(6), 938-951, 2012
- Nearly Recurrent Components in 3D Piecewise Constant Vector Fields, A.S. and N. Brunhart-Lupo, EuroVis 2012/Computer Graphics Forum, 31(3), 1115-1124, 2012
- Stable Morse Decompositions for Piecewise Constant Vector Fields on Surfaces, A.S., EuroVis 2011/Computer Graphics Forum 30(3), 851-860, 2011
- Morse Connection Graphs for Piecewise Constant Vector Fields on Surfaces, A.S., Computer Aided Geometric Design, in press
- Hierarchy of Stable Morse Decompositions, A.S., IEEE TVCG, in press

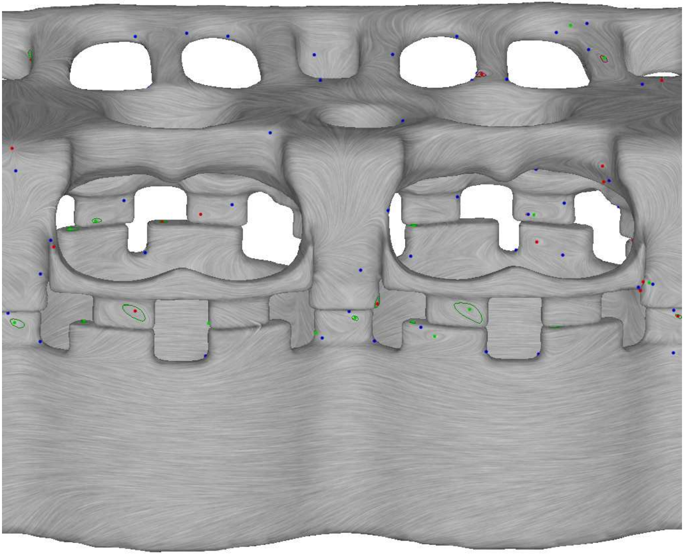
Vector field topology: prior work

- Popular topic in scientific visualization
- Focused on computing classical features
  - Stationary points (sinks, sources, saddles)
  - Periodic trajectories
  - Connecting trajectories/separatrices
- Problems with consistency because of wrong assumptions
  - Perform numerical integration and treat the resulting curves as trajectories
- Inconsistency motivated research on combinatorial approaches
  - Combinatorial vector fields
  - Edge maps
  - Morse decompositions

Inconsistency



Inconsistency



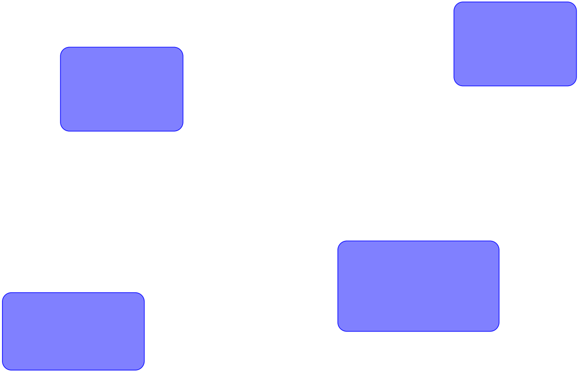
Contributions

- Use a non-standard (PC, Piecewise Constant) approximation scheme that easily supports:
  - Efficient computation of vector field features (Morse sets)
  - No need for numerical integration
  - Consistency (if implemented carefully)
  - Feature stability analysis
  - Hierarchy of features based on stability
- Why stability of features?
  - More stable features more likely to be correct if data is imprecise
    - The known imprecise data is a small perturbation of the unknown precise data
    - ... and the other way around
  - More stable features less numerous
    - Stability can be used as a filter for features
    - Topology simplification
    - Hierarchy of feature sets based on stability

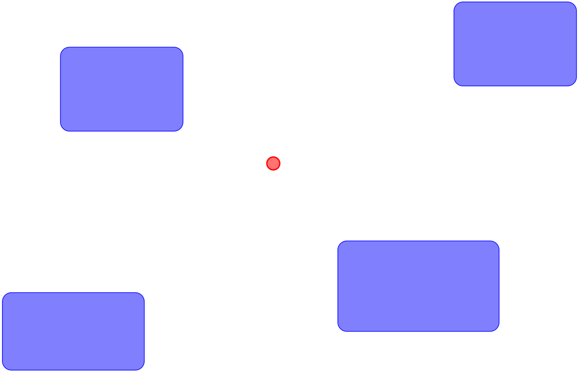
Background: Morse decomposition

- Morse decomposition is a family of Morse sets, i.e. disjoint compact sets such that:
  1. Any trajectory σ that is not contained in the union of Morse sets connects two different Morse sets
     - σ connects M_1 to M_2 if it converges to M_1 when followed backward, and to M_2 when followed forward
  2. No cycles exist in the 'is connected to' relation on the family of Morse sets

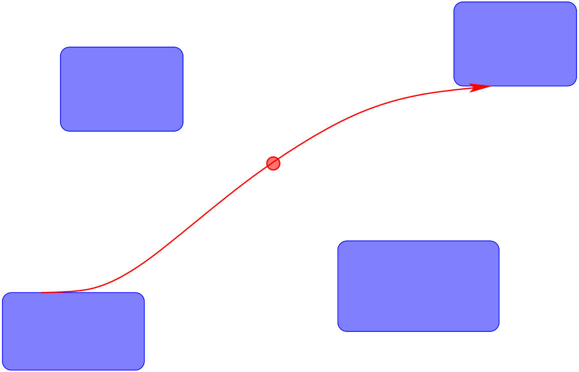
Morse decomposition



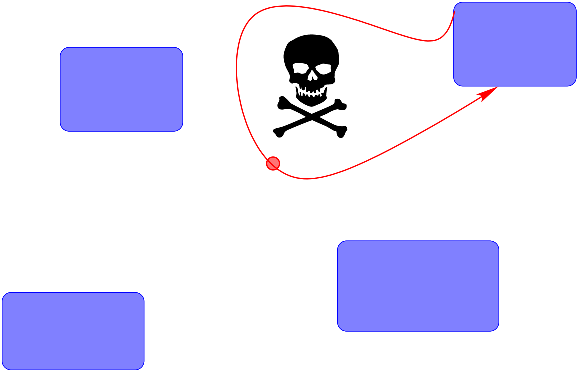
Morse decomposition



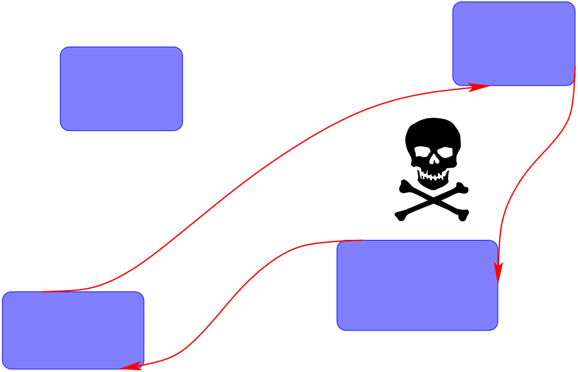
Morse decomposition



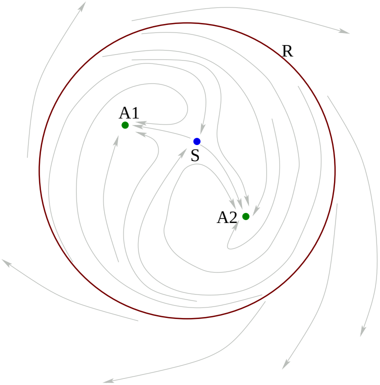
Morse decomposition

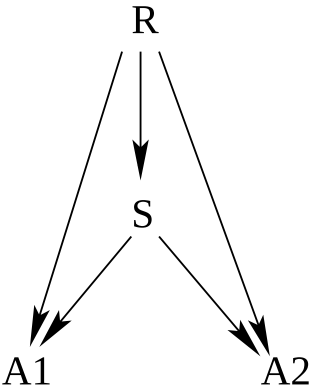


Morse decomposition

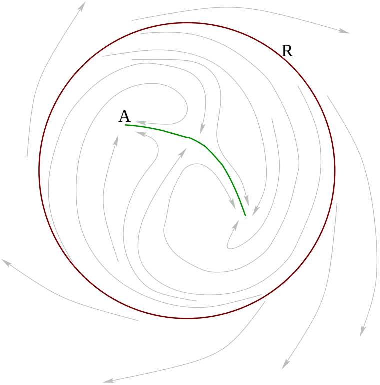


Morse decomposition





Morse decomposition

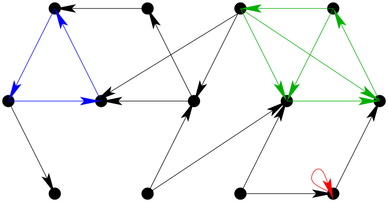


Background: Morse decomposition

- Properties
  - Flow gradient like outside Morse sets
  - Morse sets capture all recurrent dynamics (including stationary points and periodic trajectories)
  - Not unique (unlike the traditional features)
    - Coarse: more stable, fewer Morse sets but some could be complex
    - Fine: Morse sets are likely to correspond to elementary vector field features, but more Morse sets
    - Natural support for multi-scale analysis

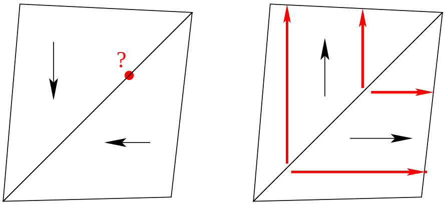
Morse decompositions: computation

- Build a finite directed graph $\mathcal{G}$ that *represents* all trajectories
  - Any trajectory can be encoded by a path in the graph
- Strongly connected components of $\mathcal{G}$ *represent* Morse sets
  - Strongly connected component: maximal set of vertices on a single loop
  - Morse set defined by a component A consist of trajectories encoded by paths in A
- A few ways to build the graph are around



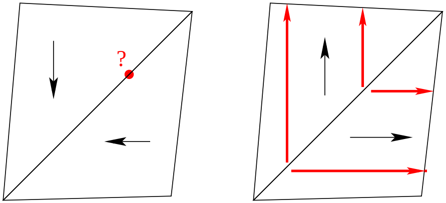
PC vector fields

- Piecewise constant vector fields
  - Constant, $= f(\Delta)$, in the interior of every triangle Δ
  - $f(\Delta)$ is required to be parallel to Δ
- Trajectories?



PC vector fields

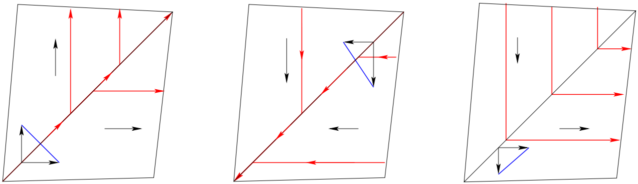
- Piecewise constant vector fields
  - Constant, $= f(\Delta)$, in the interior of every triangle Δ
  - $f(\Delta)$ is required to be parallel to Δ
- Trajectories?



- Flow complex
- Control theory
- Allow trajectories to move along the 'problem' edges

Trajectories along mesh edges

- Exploding, imploding, crossing edges



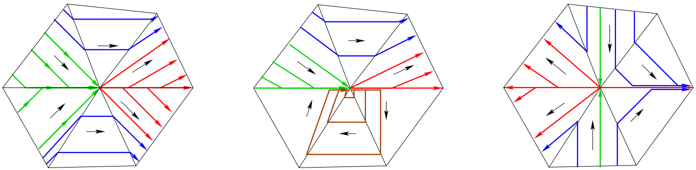
- Trajectories can follow exploding and imploding edges
  - For best results, velocity determined by vectors assigned to incident triangles
    - Make the two triangles coplanar by rotating one around the edge
    - Intersect linear space parallel to the edge with the line segment connecting the two vectors

Trajectories at mesh vertices

- Similar issues: mesh vertices
- Question: To make stationary or not to make stationary?
- Want to create flow with topological properties similar to a single-valued continuous flow
  - Upper semicontinuous (limit of trajectories also a trajectory)
  - Nonempty & acyclic set of trajectories leaving any point
  - Inspired by differential inclusions

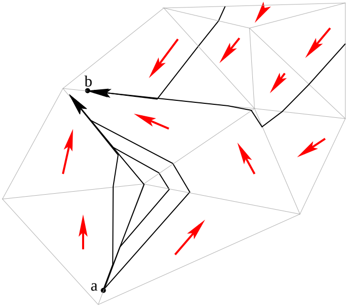
Trajectories at mesh vertices

- Perform sector analysis of a vertex
- Count elliptic (E) and stable/unstable parabolic sectors (SP/UP)
- Non-stationary $\iff [E = 0 \text{ and } SP = UP = 1]$
- Left: non-stationary
- Middle: stationary, $E \neq 0$
- Right: stationary, $UP = 2$



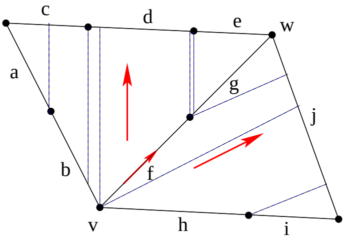
Trajectories of a PC vector field

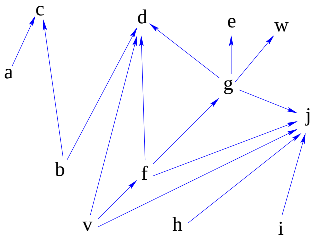
- Constant velocity in the interior of a triangle
- Polygonal lines with knots on 1-skeleton
- Can be obtained by concatenating *simple segments*, i.e. constant velocity segments contained in a single triangle
- No numerical integration needed
- Simulate an infinitesimal perturbation so that stationary points are only at the vertices



Transition graph

- Transition graph
  - Finite representation of trajectories (discretization)
  - Nodes: n-sets
    - Mesh vertices
    - Edge pieces: obtained by subdividing mesh edges
  - Arcs
    - Connect n-sets connected by a simple segment



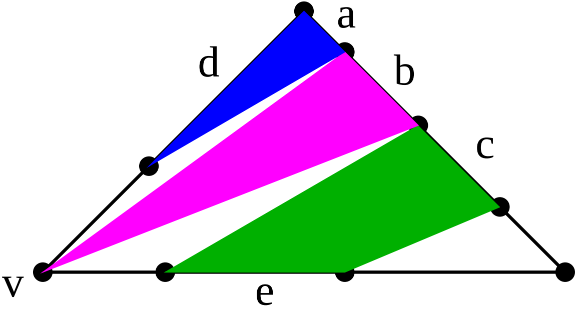


Transition graph

- Nice properties
  - Easy to adaptively refine
    - Refinement: split an edge piece f into two f_1 and f_2
    - Construct arcs out of/into f_i based on arcs out of/into f
  - Easy to coarsen
    - Coarsening: merge two adjacent edge pieces f_1 and f_2 of the same edge into one, f
    - Arcs into/out of f: built based on arcs into/out of f_i
- Adaptive refinement
  - Build coarse graph (edge pieces = mesh edges)
  - Iteratively refine edge pieces in strongly connected components

Geometric model of Morse sets

- Morse sets defined by strongly connected components
- Geometric model of a Morse set defined by a component A: union of convex hulls of pairs of n-sets in A connected by an arc



Morse set classification

- Morse set classification
  - $(i, +)$: fixed point index i and repelling
  - $(i, -)$: fixed point index i and attracting
  - $(i, 0)$: fixed point index i and neither repelling nor attracting

Common Morse set types

- Most common types of Morse sets and their classical counterparts
  - $(0, -)$; attracting periodic orbit
  - $(0, +)$; repelling periodic orbit
  - $(1, +)$; source
  - $(1, -)$; sink
  - $(-1, 0)$; saddle; in some cases, may look like a loop – this indicates a possible homoclinic orbit
  - $(0, 0)$; $\emptyset$, trivial: may contain no recurrent features or contains features that cancel
- Complex Morse set: of neither of the above types
- Classical counterpart: simplified model for the Morse set
- Guarantees
  - Nonzero index $\Rightarrow$ fixed point
  - Type $(0, +)$ or $(0, -)$, possible to flatten & no stationary point $\Rightarrow$ periodic orbit

CVPC (Convex Valued PC) vector fields

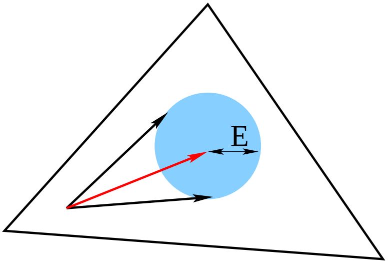
- Motivation
  - Stability measures for Morse sets
  - Making results valid for continuous vector fields (e.g. PL vector fields)
- Similar to PC vector fields, but assign a convex set of vectors $F(\Delta)$, parallel to Δ, to each mesh triangle Δ
- A PC vector field f is *feasible* $\iff f(\Delta) \in F(\Delta)$ for each Δ
- Goal: Morse decomposition valid for all feasible vector fields
  - Determined from super-transition graph, representing all trajectories of all feasible vector fields

Transition graph vs super-transition graph

- Almost the same
- Transition graph: n-sets connected by simple trajectory segments are connected by arcs
- Super-transition graph: n-sets connected by *feasible segments* are connected by arcs
  - Feasible segment: simple trajectory segment for a feasible vector field
  - Constant velocity segment contained in a single triangle
  - If moving through the interior of Δ, velocity in $F(\Delta)$
  - Can also move along a mesh edge
  - Or stay at a stationary vertex (zero velocity)

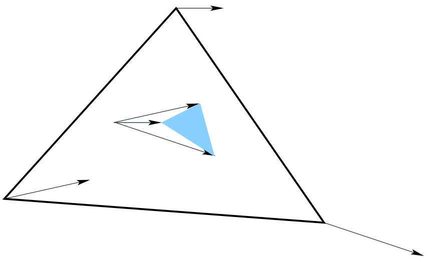
CVPC vector fields: example

- f : PC vector field
- $F(\Delta)$ = Disk of radius E centered at $f(\Delta)$
  - Feasible vector fields: perturbations of f by no more than E
  - Output Morse sets valid for all PC vector fields no more than E away from f
- Performance
  - Much slower than PC vector field analysis
  - More feasible segments ⇒ larger graph



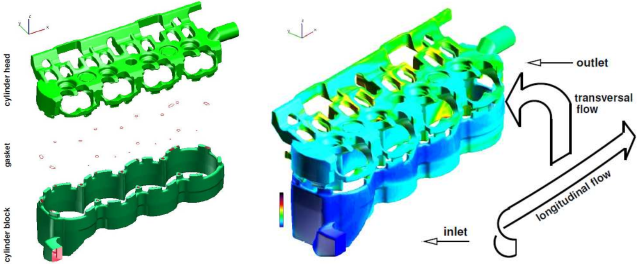
CVPC vector fields: examples

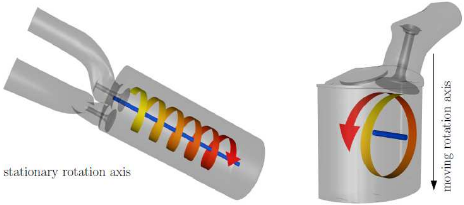
- Envelope for a PL vector field
  - $F(\Delta)$ is the convex hull of vectors assigned to vertices of Δ, projected to Δ's plane.
  - Output Morse sets valid for the piecewise linear vector field (on flat patches)



Input vector fields

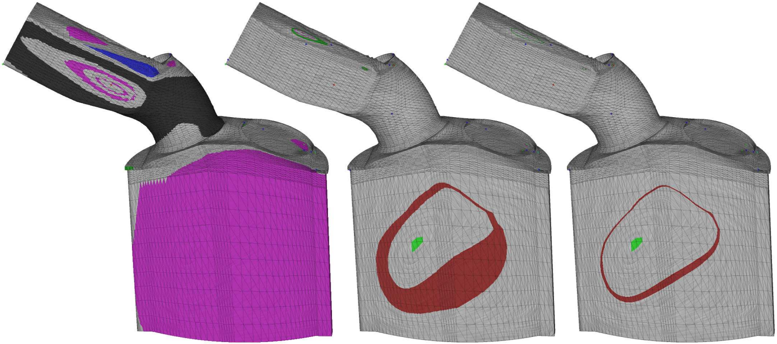
- Fluid simulations
- Velocity extrapolated to the boundary





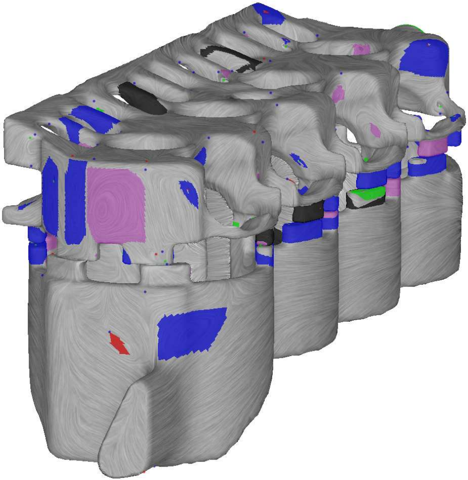
Morse decompositions: examples

- About 26k triangles
- 2,6,9 refinement iterations; 0.92, 1.49, 1.82 seconds



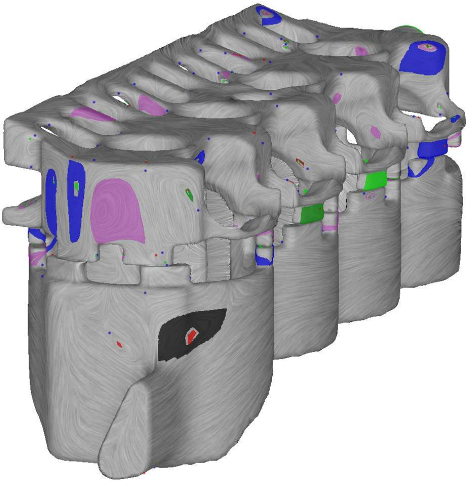
Morse decompositions: cooling jacket dataset

- About 228k triangles
- 2 subdivision iterations (8 s)



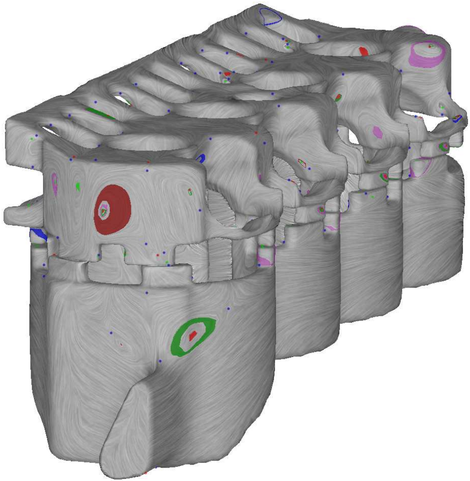
Morse decompositions: cooling jacket dataset

- 4 subdivision iterations (12 s)



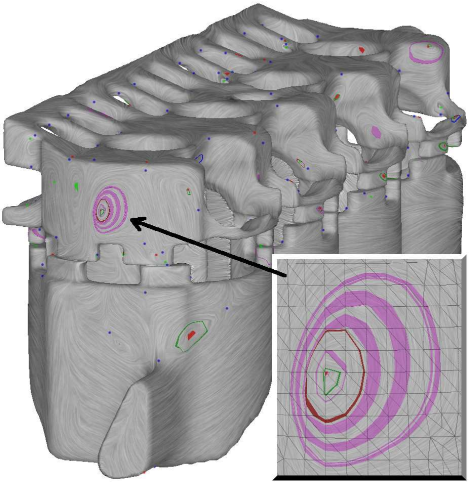
Morse decompositions: cooling jacket dataset

- 6 subdivision iterations (17 s)



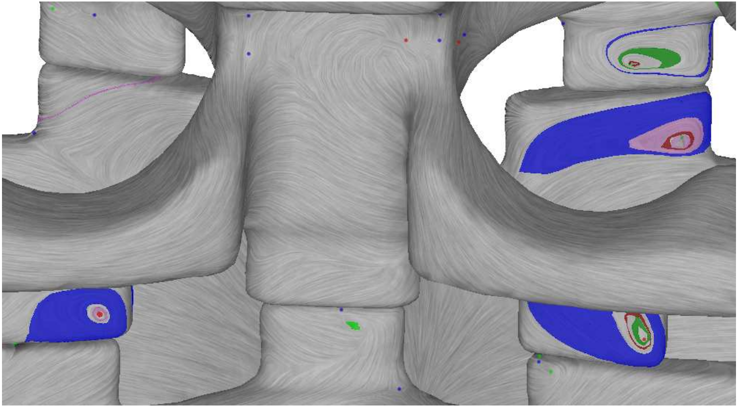
Morse decompositions: cooling jacket dataset

- 8 subdivision iterations (22 s)



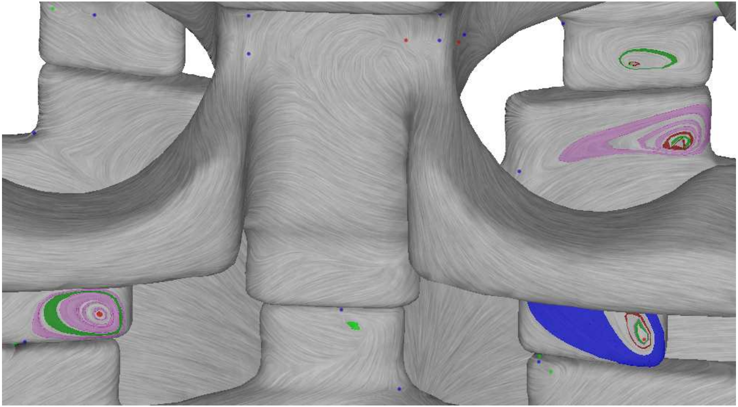
Morse decompositions: cooling jacket dataset

- 6 subdivision iterations (17 s); Note the blue rings



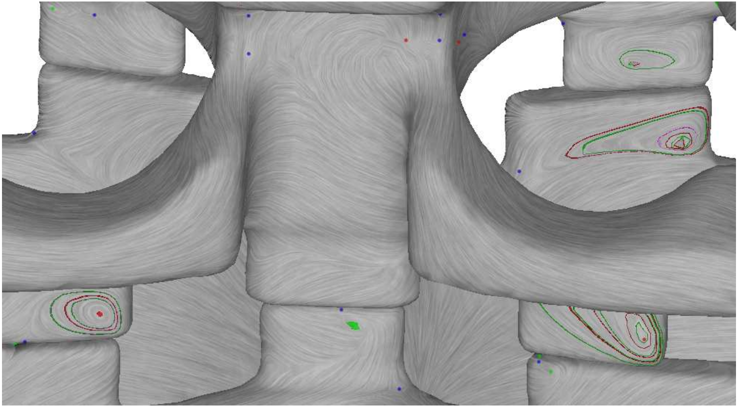
Morse decompositions: cooling jacket dataset

- 7 subdivision iterations (19 s)



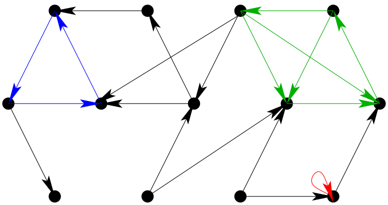
Morse decompositions: cooling jacket dataset

- 11 subdivision iterations (30 s)



Morse connection graph

- Morse connection graph (MCG)
  - Nodes: nontrivial Morse sets
  - Arcs: Connections represented by paths connecting different strongly connected components of the transition graph
- Similar, but not the same as linkage graph
- Not all paths in the graph represent trajectories (some MCG arcs "false positives")

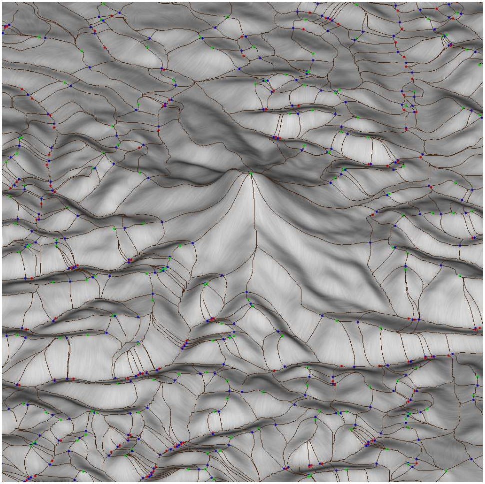


Morse connection graph

- Goal: make MCG *closer* to the linkage graph
- Basic idea: refine edge pieces on paths in the transition graph originating from/ending at nontrivial Morse sets that are neither attracting nor repelling
- Refinement depth driven by the ability to *prove* that the path in the graph defines a string of connecting trajectories
  - Refine along paths that would define "structurally unstable" connections (i.e. between Morse sets that are neither attracting nor repelling)
  - Parameters: minimum/maximum subdivision depth for edge pieces

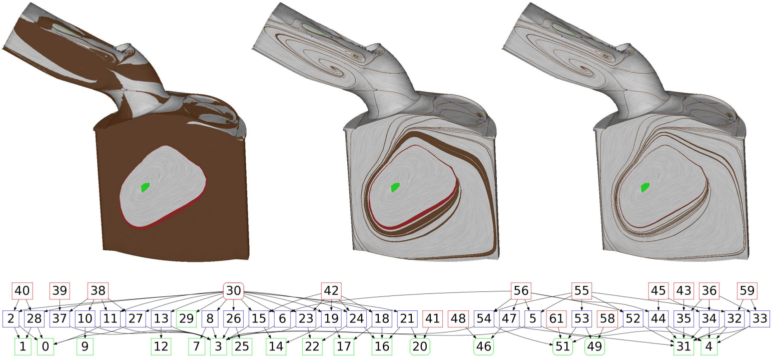
Gradient vector fields

- Related to the Morse complex



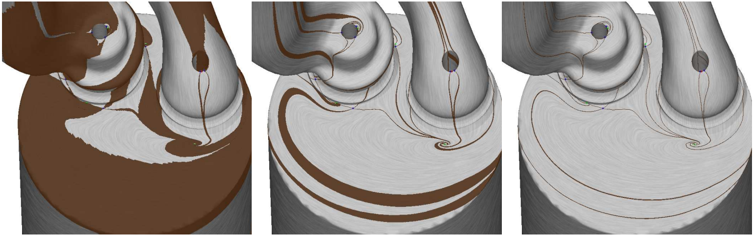
MCG: examples

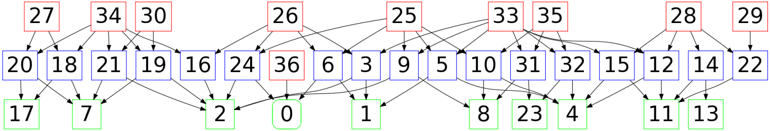
- Gas engine, 4, 12 and 80 seconds
- Brown regions: represented by paths starting/ending at a nontrivial Morse set that is neither repelling nor attracting



MCG: examples

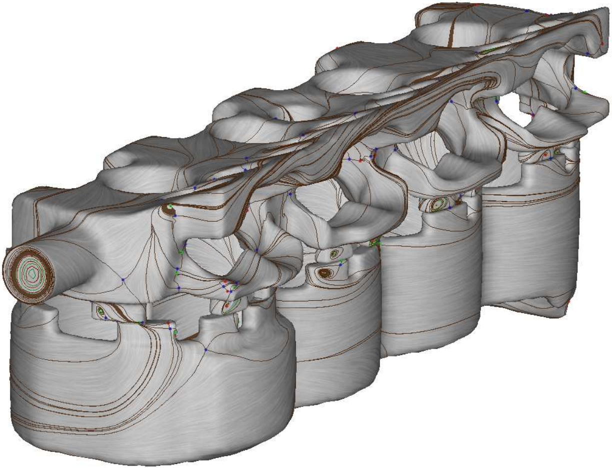
- Diesel engine (222k triangles), 11, 16 and 28 seconds





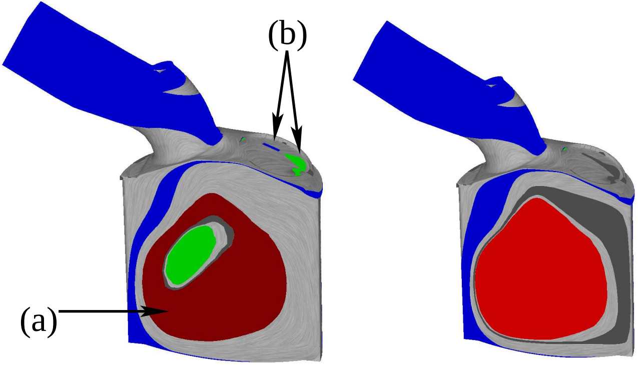
MCG: examples

- Cooling jacket, 8 minutes (1 connection not verified)



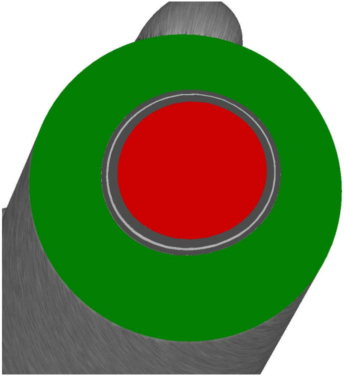
Gas engine dataset

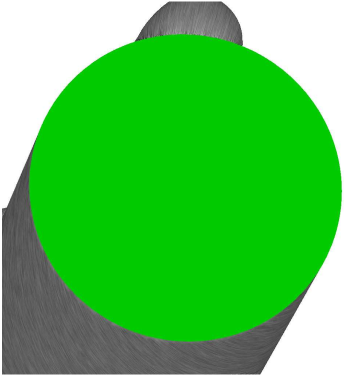
- Morse decompositions valid for 10% and 11% perturbations [% of the mean magnitude]
- Need at least 10% perturbation to remove periodic orbit(s) from (a)
- Morse sets grow and merge as perturbation is increased
- (b): saddle-sink cancellation



Diesel engine dataset

- 2.5% and 2.6% perturbations
- Running times: 3.5 – 4.5 minutes





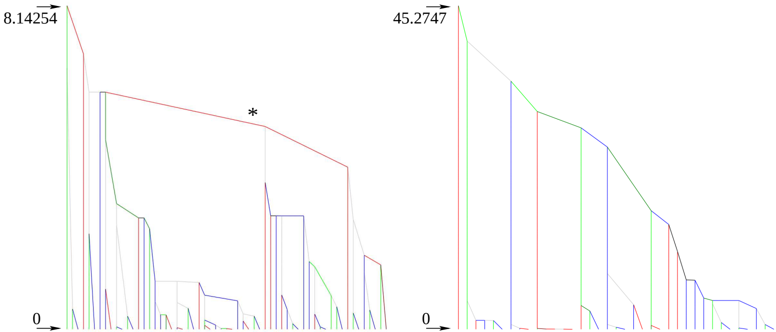
Morse hierarchy

- Hierarchy of Morse decompositions based on stability
- Can be constructed using a sweep algorithm
  - Gradually increase E, keeping track of graph and Morse set changes
  - Slow ($6.5 - 154$ minutes for the three datasets)
- However, the results can be explored interactively
- Videos

Gas engine
Diesel engine
Cooling jacket

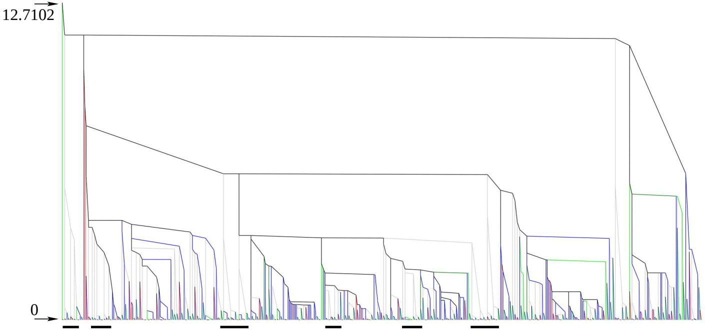
Morse merge tree

- Morse sets grow and merge
- Morse merge tree: representation of the mergers; can be used to explore the hierarchy
- Height of a vertex: E for which the merger took place
- Gas and diesel engine

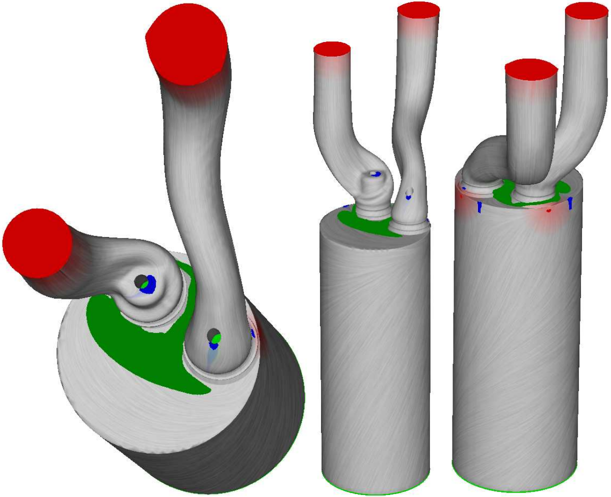


Morse merge tree

- Cooling jacket
- Notice relatively few attracting and repelling sets higher up

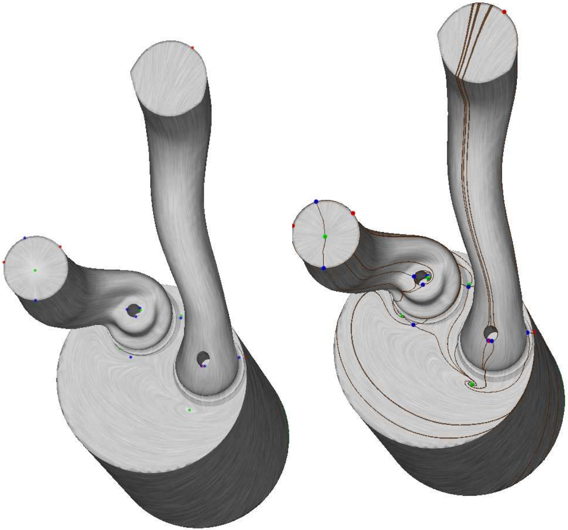


Diesel engine: stable Morse decomposition

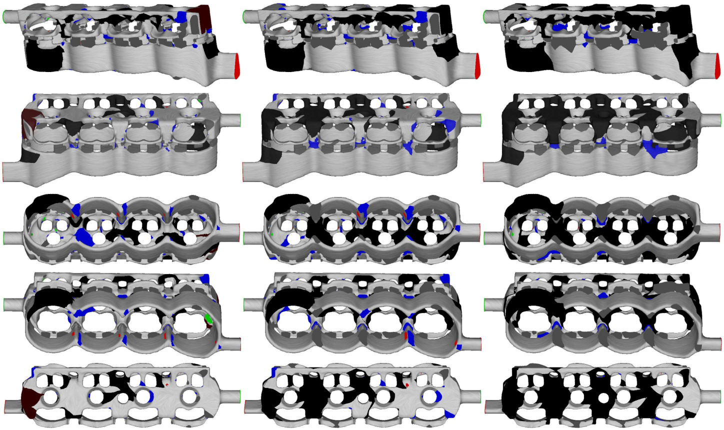


Diesel engine: fine Morse decomposition

- More features, their physical meaning not clear



Cooling jacket: stable Morse decomposition

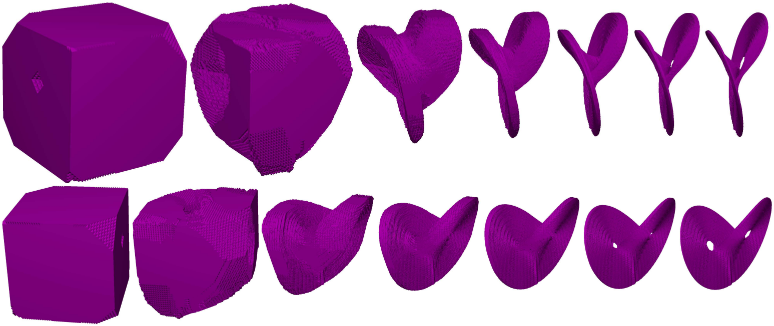


3D case

- Current implementation: regular grids
- Use transition graph to compute *nearly recurrent components*
  - Regions of close to circulating flow
  - Represented by strongly connected components of the graph
  - No stability yet
- Trajectories: $\dot{x}(t) \in F(x(t))$, where $F(x)$ is defined as the convex hull of the set consisting of vectors assigned to all 3D cells containing x
- Transition graph nodes: face pieces, edge pieces and vertices

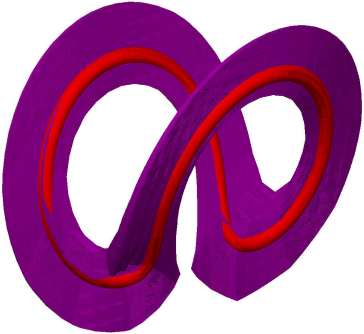
Lorenz attractor

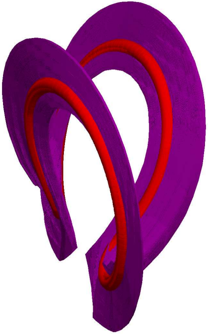
- 48^3 grid
- Computation times between 16 seconds and 10 minutes
- Note the holes at stationary points



Lorenz system, non-standard parameter values

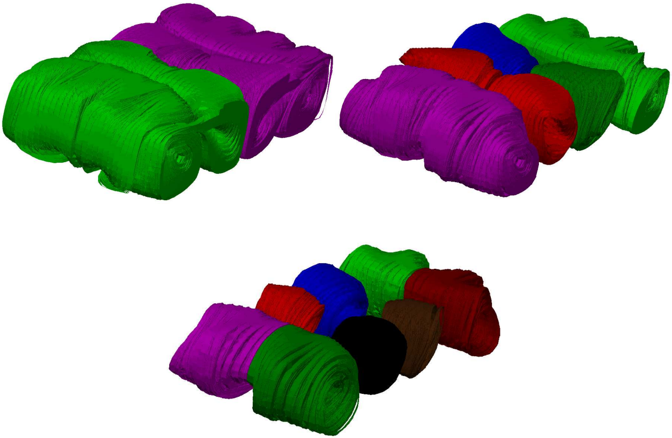
- Nearly recurrent set containing a periodic trajectory proven to exist in [Mrozek & Pilarczyk 2002]





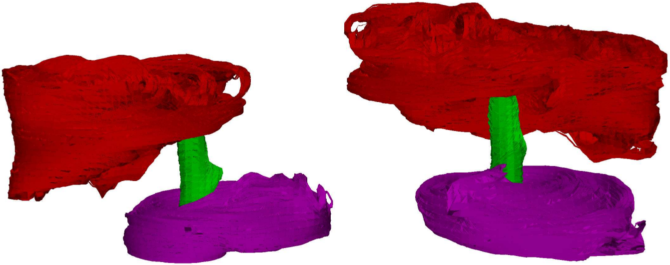
Example: Bérnard convection dataset

- $64 \times 16 \times 32$, 4 refinement iterations, $3 - 8$ minutes
- Remove layers of grid elements near the floor and ceiling
- Output represents circulating regions that stay away from floor and ceiling



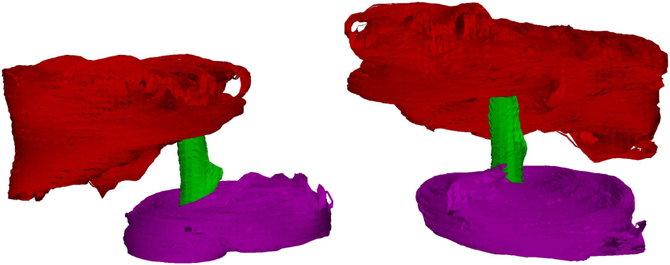
Example: Hurricane Isabel dataset

- Restrict computation to neighborhood of the hurricane

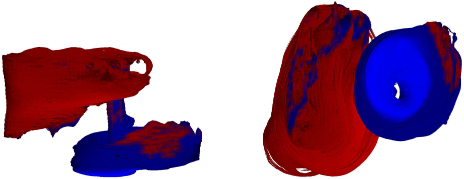


Example: Hurricane Isabel dataset

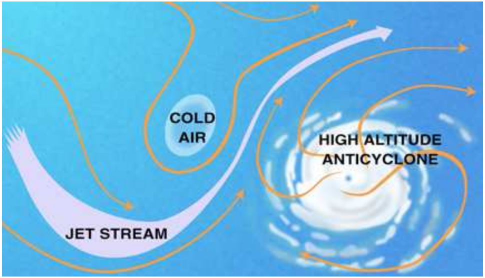
- Restrict computation to neighborhood of the hurricane

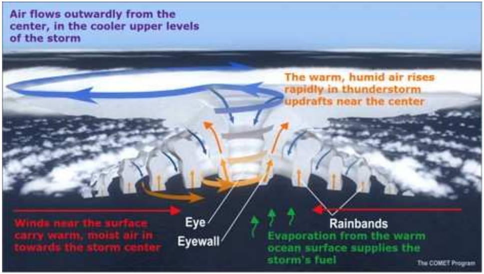


- Top component: CW-spinning winds
- Bottom component: CCW-spinning winds



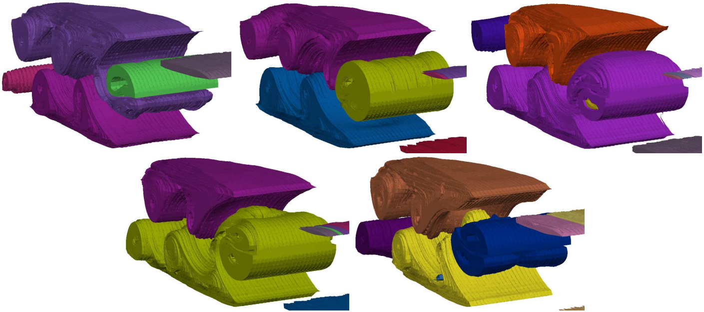
Example: Hurricane Isabel dataset





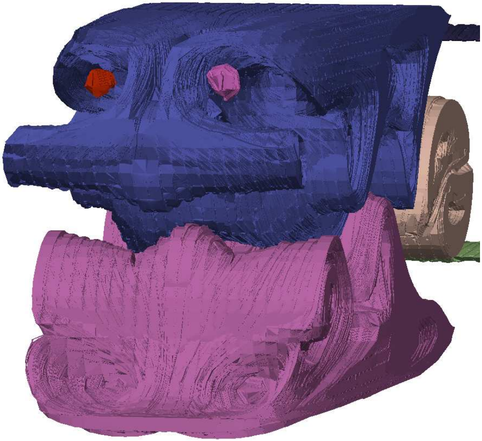
Flow around square cylinder

- Resampled $96 \times 32 \times 24$ grid
- Computation times around 8 min per frame
- Frame of reference change (subtract average velocity)



Flow around square cylinder

- View from downstream direction, end of the simulation



Future work

- What do the Morse sets converge to as number of refinement iterations goes to infinity?
  - Strong evidence that the limit is the chain-recurrent set of the PC vector field in the single valued PC case
- Morse decompositions and the Conley index
  - Index pairs from the transition graph?

Future work

- Reduce the size of the transition graph, particularly in the CVPC case
  - Filippov-Ważewski relaxation theorems

    can steer a car using  instead of 

  - Most likely, few arcs of the super-transition graph actually matter (perhaps only the ones corresponding to extreme directions of the CVPC vector field)
- Direct computation (is transition graph really necessary?)

Questions

?