Simplifying Unstructured Grids for Oceanographic Visualizations

Ole Tytlandsvik

December 6, 2024

Oceanbox

- Tromso-based
- Interactive oceanographic simulations
- Oceanography as a Service
- Web-Based Geographic Information System (Web GIS)
- Digital twin of the coastal ocean

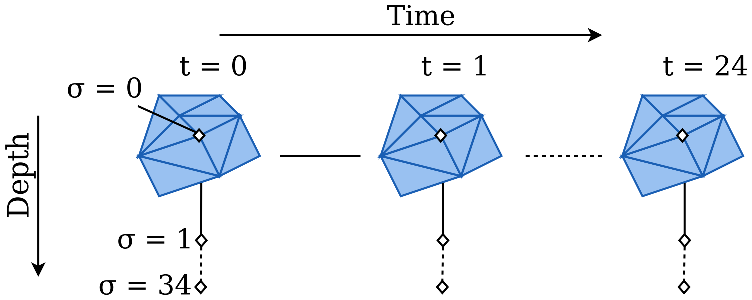
The Problem

- Large data sets
  - High resolution (millions of spatial points)
  - Multi-dimensional
  - Payloads of 20Mb+
  - Unresponsive web application
  - Increased bandwidth costs

The Solution: Lossy Compression

- Traditional approaches tricky
  - Accuracy of coordinates are important
  - Tiling/Multi-resolution not trivial with *unstructured grids*
  - Inflated data size should be smaller on the client
- Grid simplification
  - Remove vertices/nodes
  - Maintain visualization quality
  - **Angle bound half-edge collapse**

FVCOM grids



- Unstructured grid
  - Variable resolution
- Multiple dimensions
  - Coordinates, depth, time

Related Work

Rasterization-based approaches

- Good compression rates (90%)
- Requires *structured* grid layout

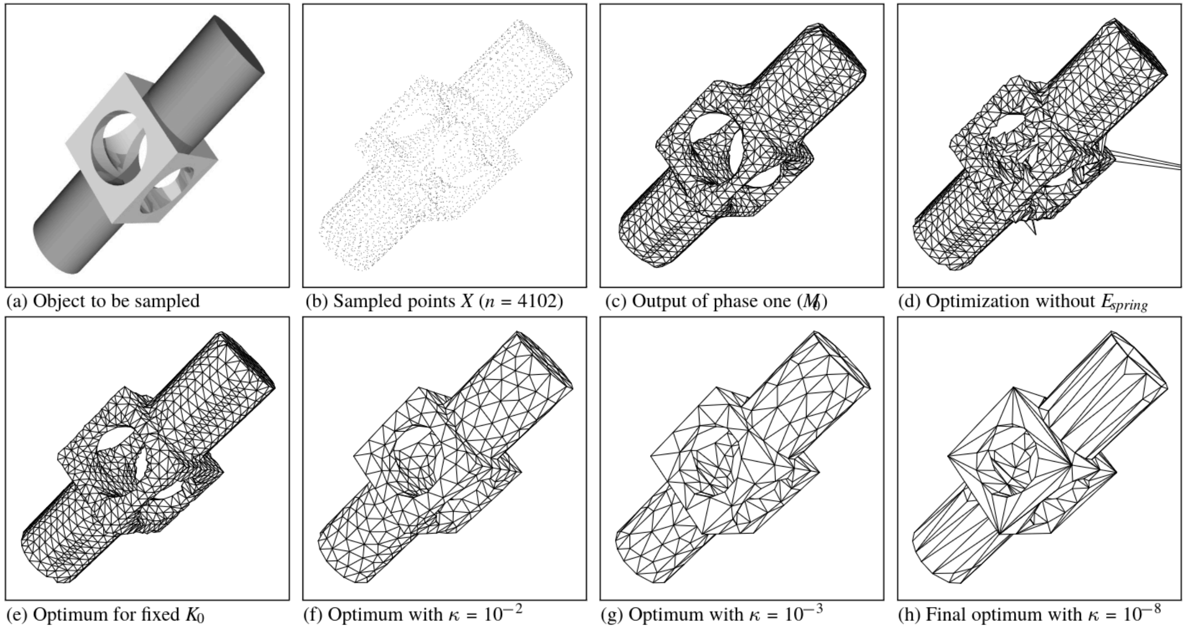
Tiling approaches

- Load only what you need
- Varying resolution
- Also requires structured grids

Mesh simplification

- More suited for unstructured grids
- Popular in literature: *3D mesh decimation*
- Not necessarily directly applicable…

Related Work: Mesh Simplification



- Approximate a surface
- Preserve topology
- Good reduction[1]

However

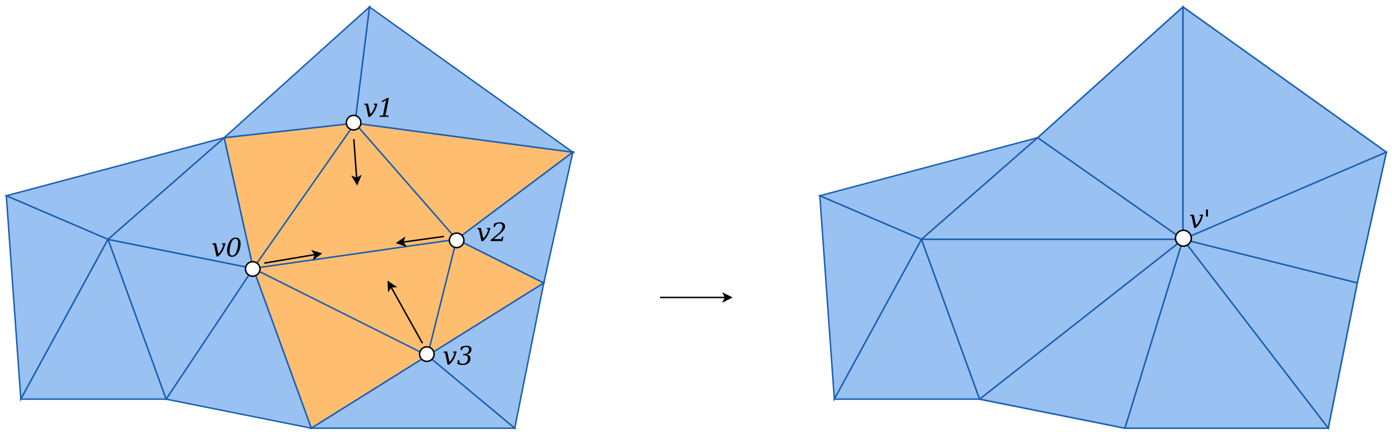
- Not ideal for 2D grids
- Need even resolution

[1]Figure from Hoppe et al. “Mesh Optimization”, *ACM*, 1993, pp. 19-26

Related Work: Mesh Simplification Operators

Vertex Clustering

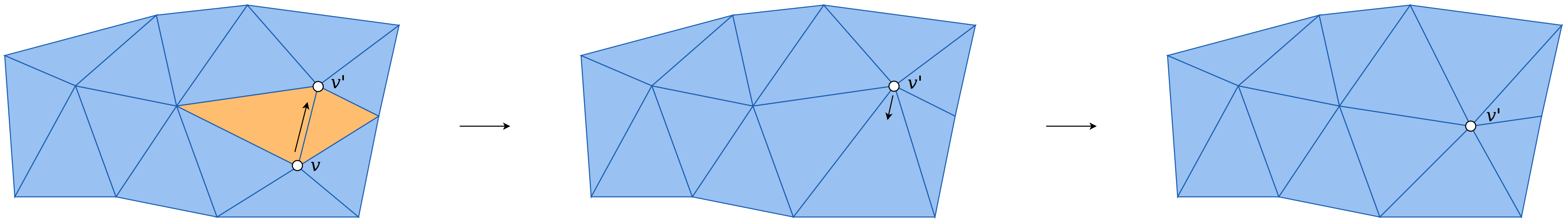
Identify a “cluster” of vertices and represent them all with one vertex



Related Work: Mesh Simplification Operators

Edge Collapse

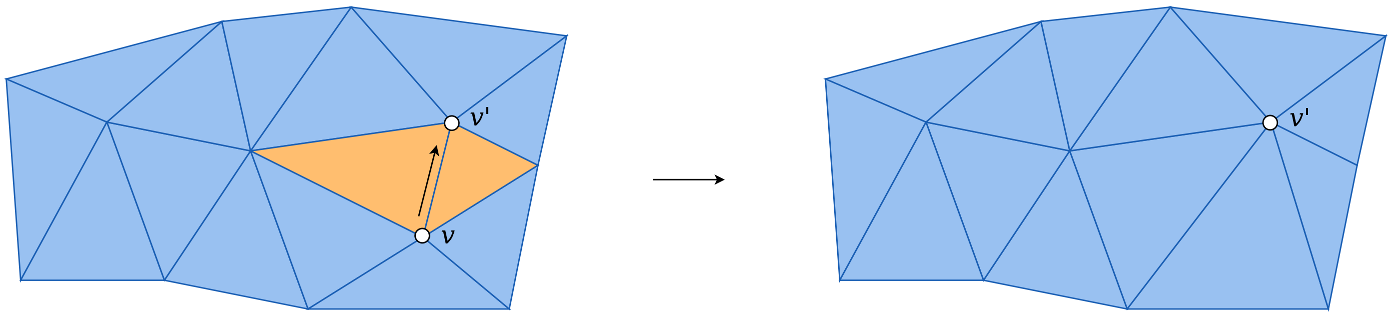
Collapse an edge between two vertices, representing them with one vertex



Our Approach: Angle Bound Half-edge Collapse

Adaptation from previous work.[1]

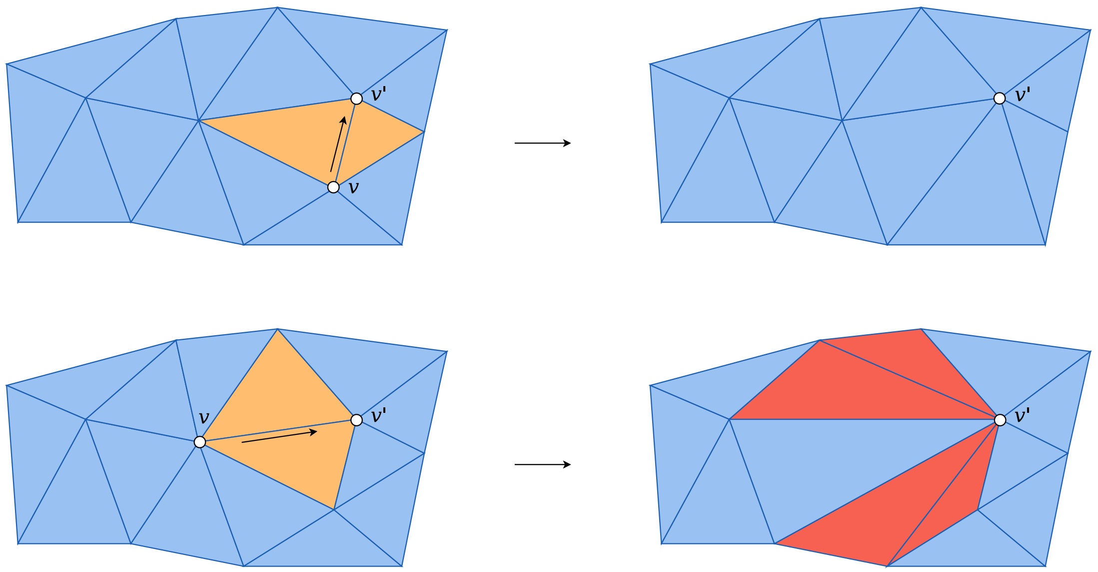
Half-edge collapse with a minimum angle criterion to inner angles



[1]Hinderink et al. "Angle-Bounded 2D Mesh Simplification." *Computer Aided Geometric Design*, vol 95, May 2022, p. 102085

Our Approach: Angle Bound Half-edge Collapse

- We collapse v into v' by collapsing the half-edge $v \rightarrow v'$
- We define a strict angle bound θ
- We ensure inner angles θ_n respect $\theta_n > \theta$



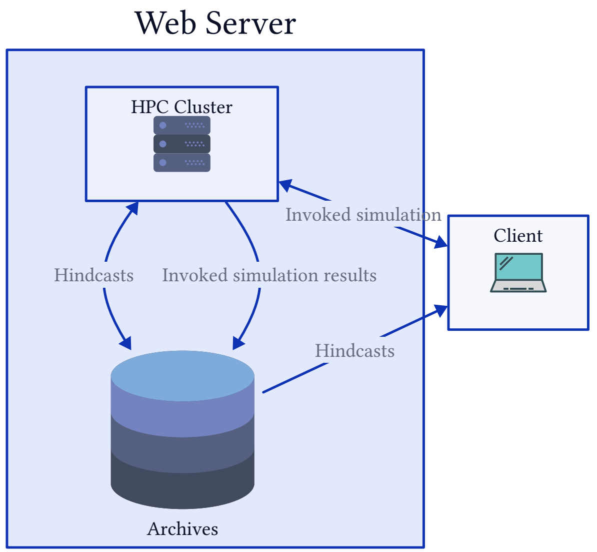
Architecture Overview

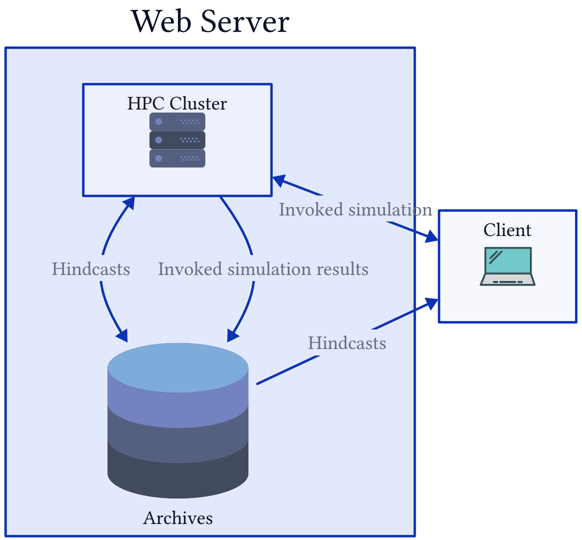
We concentrate on the data flow of *Archives*.

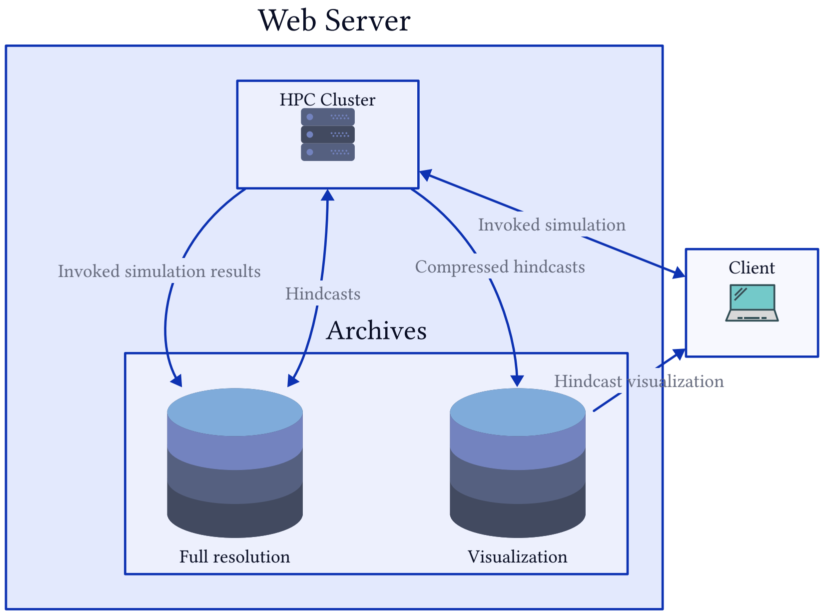
- *Hindcast* simulations
- *User invoked* simulations

Hindcasts are periodically produced, and are the basis of visualizations.

These are the archives we aim to compress/reduce.




Web Server
HPC Cluster
Invoked simulation
Client
Hindcasts
Invoked simulation results
Hindcasts
Archives


Web Server
HPC Cluster
Invoked simulation
Client
Invoked simulation results
Compressed hindcasts
Hindcasts
Archives
Hindcast visualization
Full resolution
Visualization

Implementation Details

Two-fold design:

Grid Simplification

- Once per grid geometry
  - Can be slow
- Boundary nodes preserved

Archive Application

- Picking out values from full res -> compressed
- Vertices a proper subset of original vertices
- Also truncate depth dimension

Evaluation

Visualization similarity

- Inspection of raster images

Geometric Error

- Angle distribution
- Triangulation inspection

Compression/Speedup

- Compression ratio of payloads
- Transfer speed

For the Master:

- Pixel-by-pixel comparison
- Client execution time
- Hausdorff distance

Evaluation

Grids

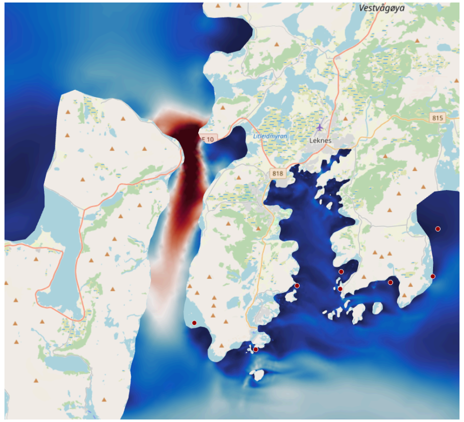
- Buksnes Waste (test)
- PO5 (prod)
- PO6 (prod)

Comparison

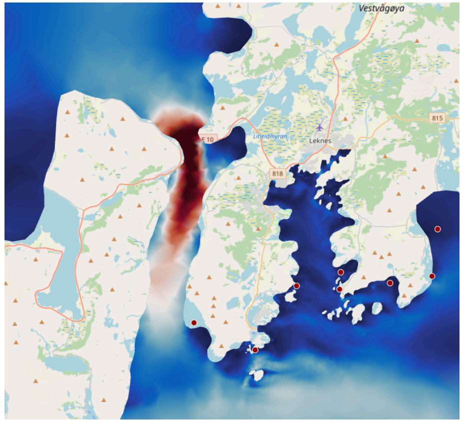
1. Original grid
2. Randomly reduced[1]
3. Angle bound, $\theta = 28°$
4. Angle bound, $\theta = 30°$

[1]Triangulated with Delaunay

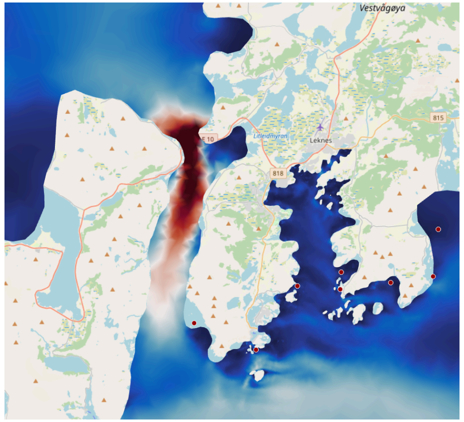
Visualization Similarity: Speed, Buksnes Waste



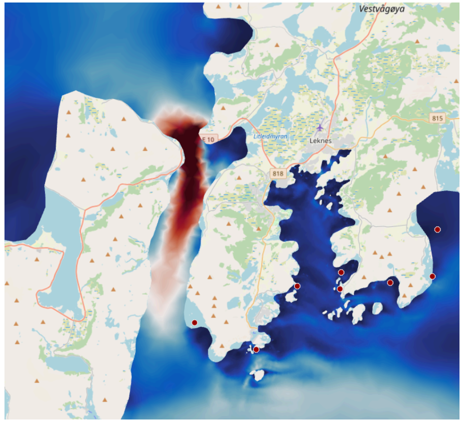
Original grid



Random reduction

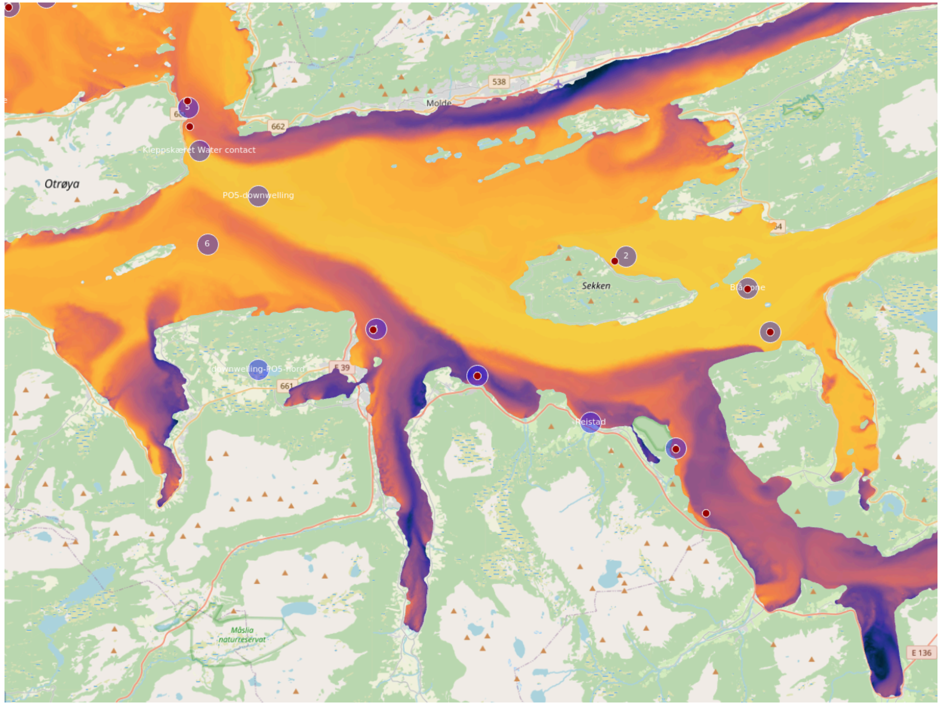


Angle bound, 28 °

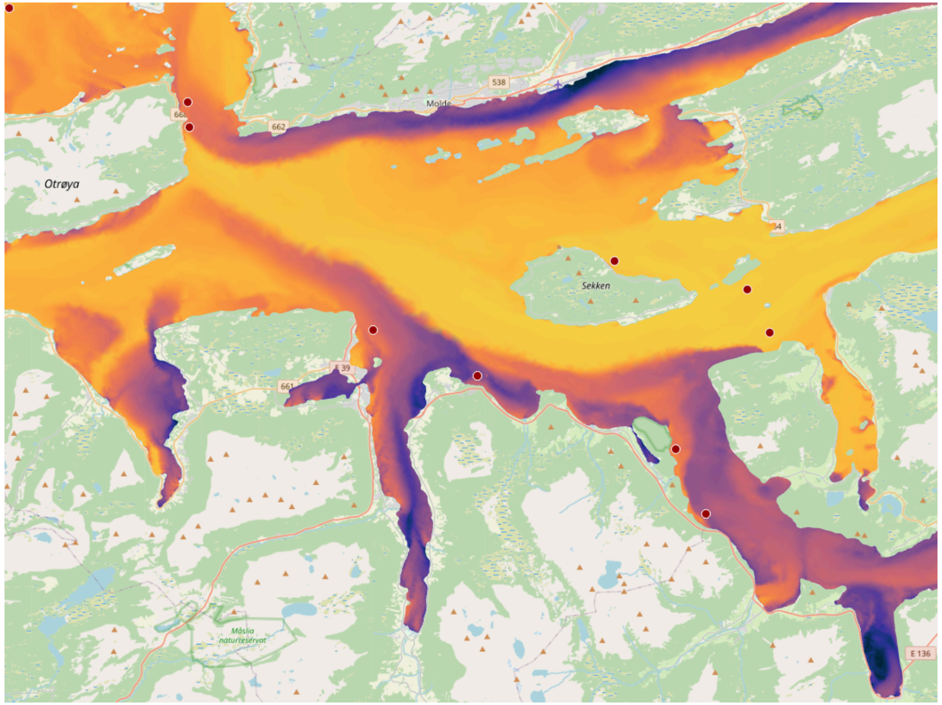


Angle bound, 30 °

Visualization Similarity: Temperature, PO5

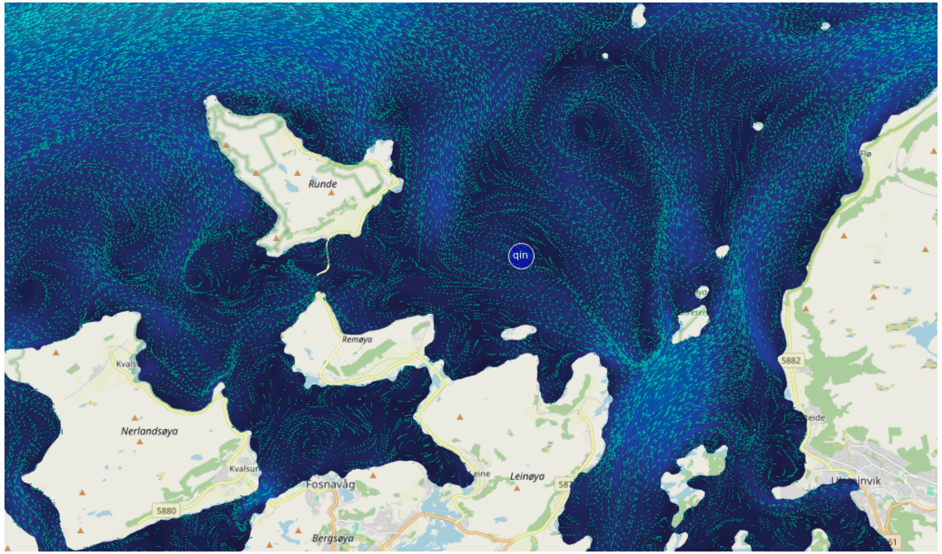


Original grid

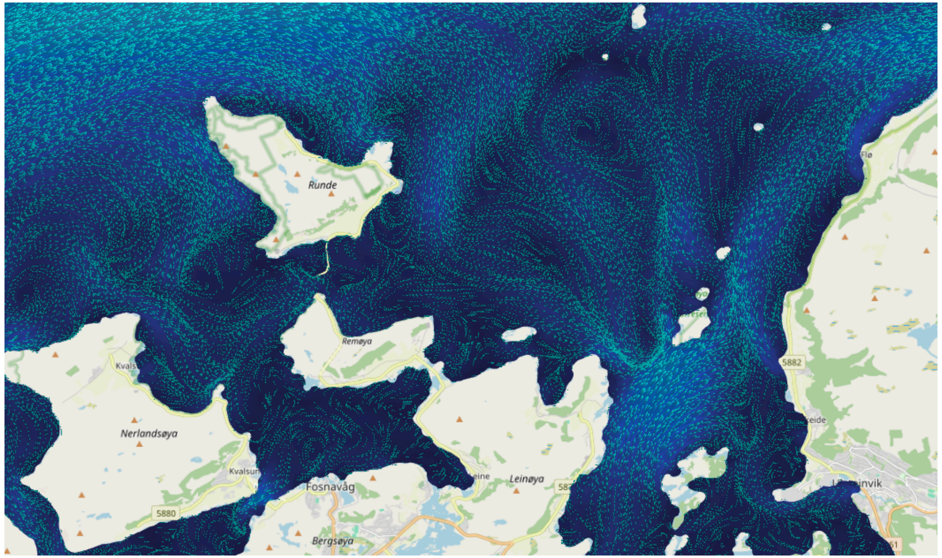


Angle bound, 28°

Visualization Similarity: Streams, PO5

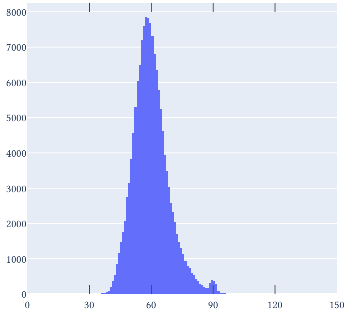


Original grid

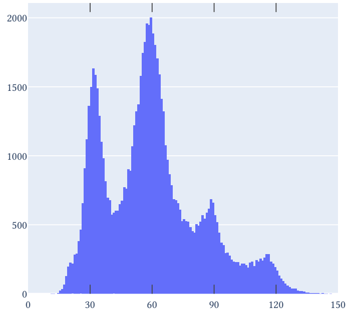


Angle bound, 28°

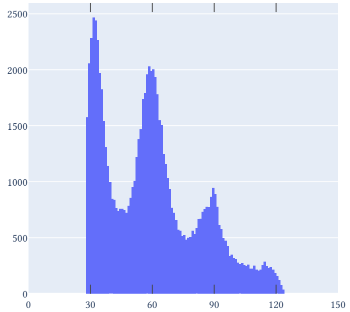
Geometric Similarity: Angle Distribution



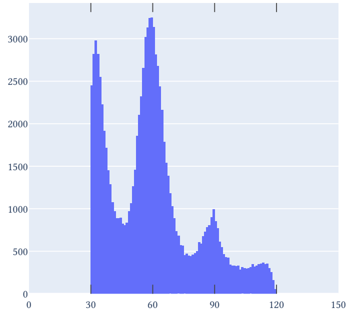
Original grid



Random reduction

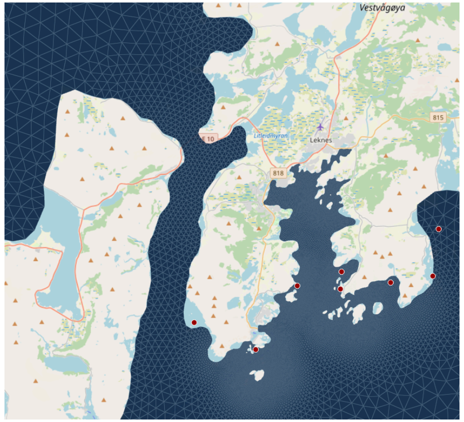


Angle bound, 28°



Angle bound, 30°

Geometric Similarity: Triangulation



Original grid

Random reduction

Angle bound, 28°

Angle bound, 30°

Compression Ratio

Data set	Size / Compression Ratio				
	Nodes	Elements	Geometry	Nodal variable	On disk
Buksnes Waste	25 136	48 332	762 KiB	98 KiB	475 686 KiB
Random	1.87	1.93	1.91	1.88	29.27
SHAVER 28°	1.71	1.76	1.74	1.72	26.62
SHAVER 30°	1.43	1.45	1.45	1.44	22.05
PO5	459 242	869 324	13 669 KiB	1 793 KiB	8 473 272 KiB
SHAVER 28°	1.67	1.74	1.71	1.67	26.03
PO6	1 691 194	3 251 577	51 317 KiB	6 606 KiB	32 002 309 KiB
SHAVER 28°	1.64	1.69	1.67	1.64	25.58

Compression Ratio

- External factors
  - Lossless compression
- Nodal variable closer to theoretic

Data set	Size	
	Geometry	Nodal variable
Buksnes Waste	363 KiB	126 KiB
Random	71.2%	98.4%
SHAVER 28°	93.1%	98.8%
SHAVER 30°	93.8%	99.3%
PO5	7 250 KiB	2 300 KiB
SHAVER 28°	93.0%	100.6%
PO6	26 200 KiB	8 460 KiB
SHAVER 28°	91.6%	100%

Speedup

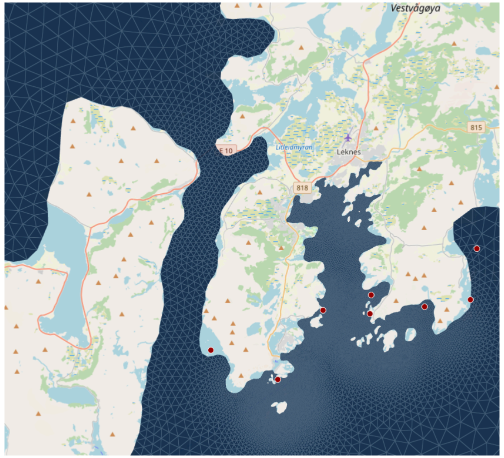
Data set	Speed / Speedup	
	Geometry	Nodal variable
Buksnes Waste	140 ms	22 ms
Random	3.68	2.44
SHAVER 28°	3.5	1.57
SHAVER 30°	2.92	1.47
PO5	2 700 ms	340 ms
SHAVER 28°	1.99	1.29
PO6	10 000 ms	950 ms
SHAVER 28°	1.85	0.84

- Unexpected results
  - Geometry vs Nodal variable
  - Buksnes vs PO5/PO6

Theory: Discrepancies in NetCDF *Chunk Size*

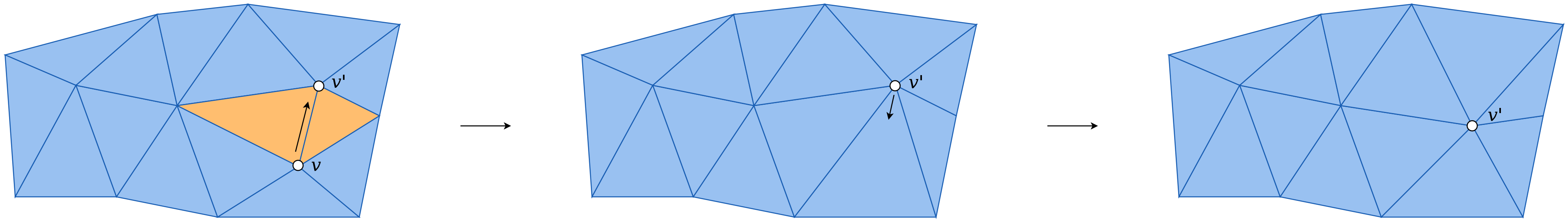
Summary

- Compression by Grid Simplification
- Angle bound half-edge collapse
- Varying results
  - Jagged visualizations
  - Only 1.7x compression



Future Work

- Angle bound *edge* collapse
- Investigate other methods
  - Spring constant
- “Conventional” lossy compression for variables



Questions