JTS Topology Suite
State of the Lib

Martin Davis

October 2015

What is JTS?

- API for representing and processing **2D linear vector Geometry**
- Implemented in Java; licensed under LGPL
- Provides the full OGC **Simple Features for SQL** geometry specification:
  - Points, Linestring, Polygons, collections
  - **Metrics:** Length, Area, Distance
  - **Predicates:** intersects, contains, etc.; relate for DE-9IM
  - **Overlay:** intersection, union, difference, symDifference
  - **Algorithms:** Convex Hull, Buffer
- Other features:
  - Validation, Polygonization, Simplification, Linear Referencing, etc.

Project History

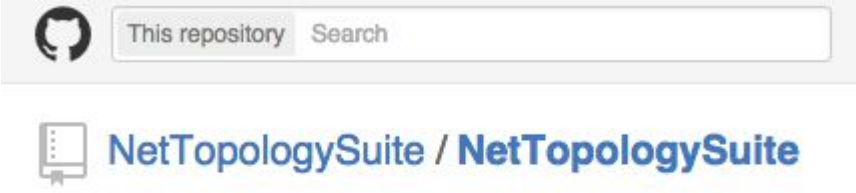
- **Version 1.0** - May 2001

  ...
- **Version 1.9** - January 2008
- **Version 1.10** - December 2008
- **Version 1.11** - March 2010
- **Version 1.12** - June 2011
- **Version 1.13** - December 2012
- **Version 1.14** - *Coming Soon!*

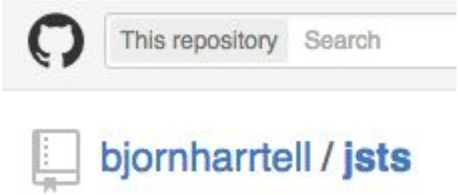
JTS Ports & Bindings

- **Ports**
  - ***GEOS*** (C++)
  - ***Net Topology Suite*** (C#)
  - ***JSTS*** (JavaScript)

GEOS Geometry Engine Open Source


This repository Search

NetTopologySuite / **NetTopologySuite**

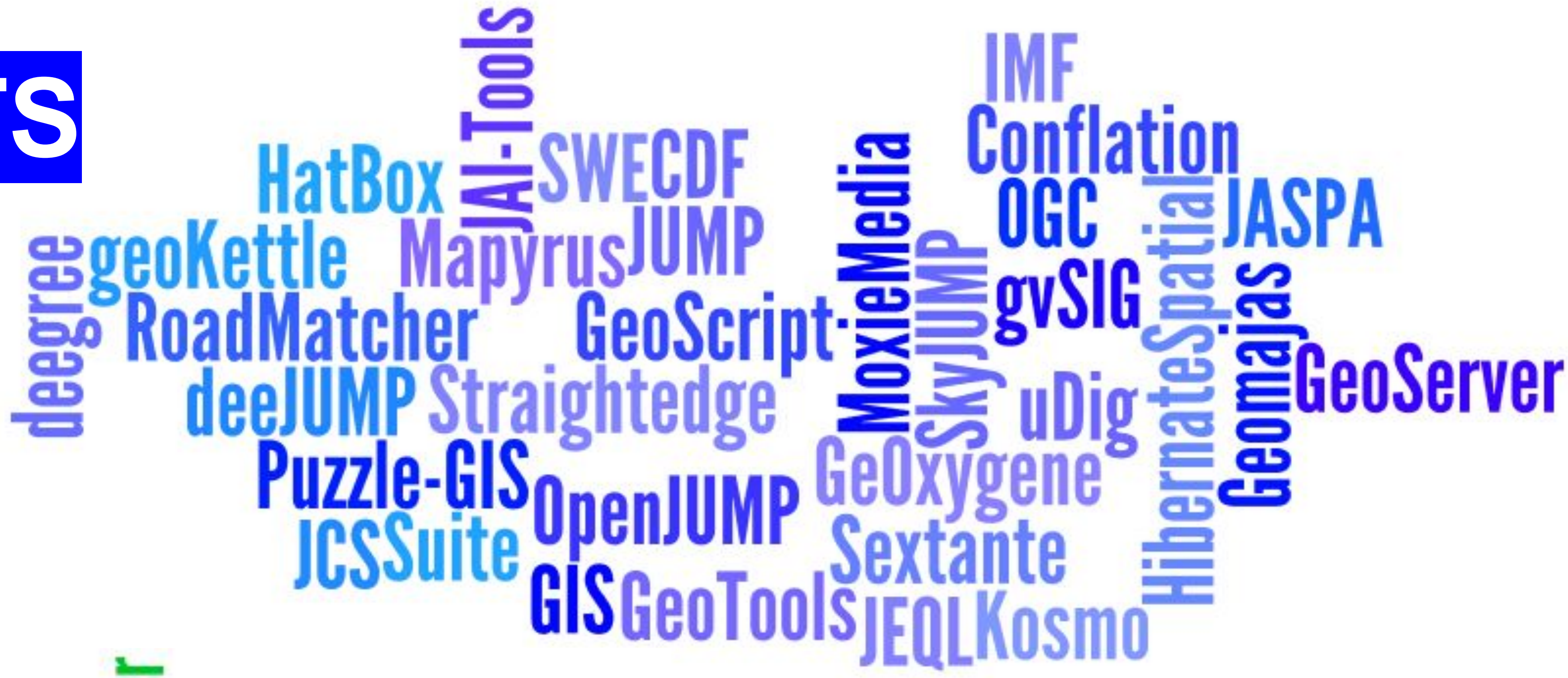

This repository Search

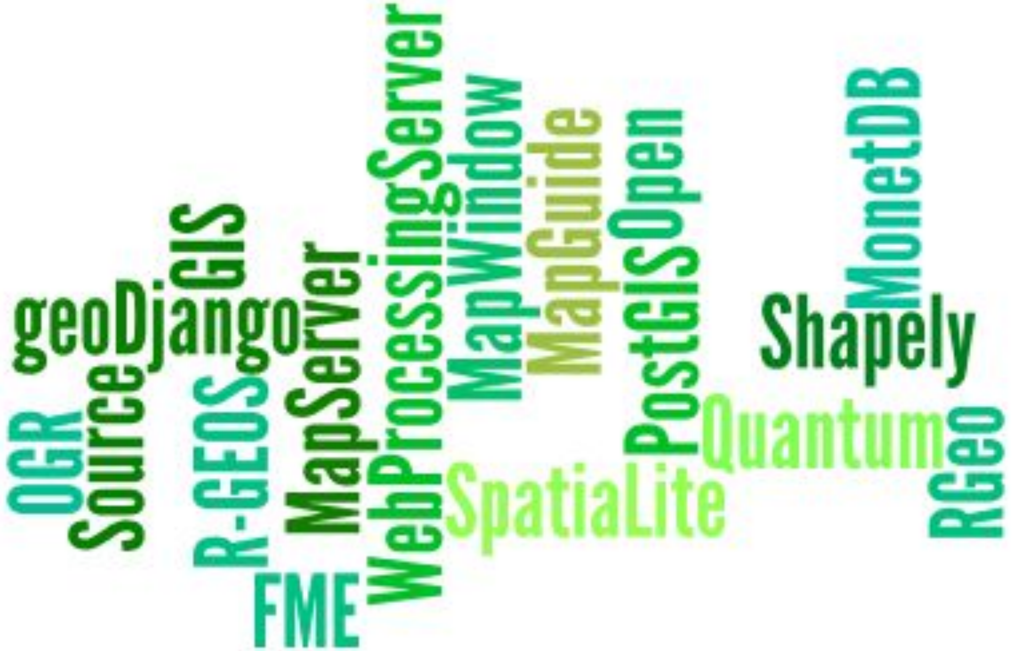
bjornharrtell / **jsts**

- **Bindings *(on JVM)***
  - Groovy, Scala, Jython, JRuby, Clojure, etc
- **Bindings *(to GEOS)***
  - Shapely (Python)
  - RGeo (Ruby)
  - R-GEOS (R)

Where is JTS used ?

JTS



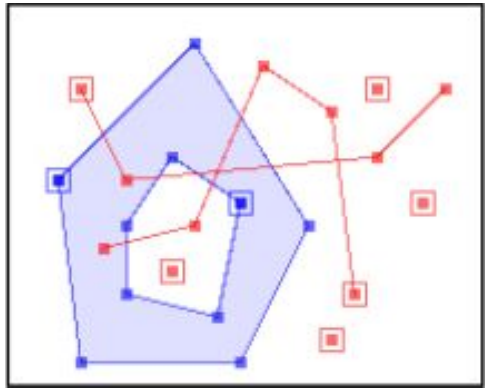


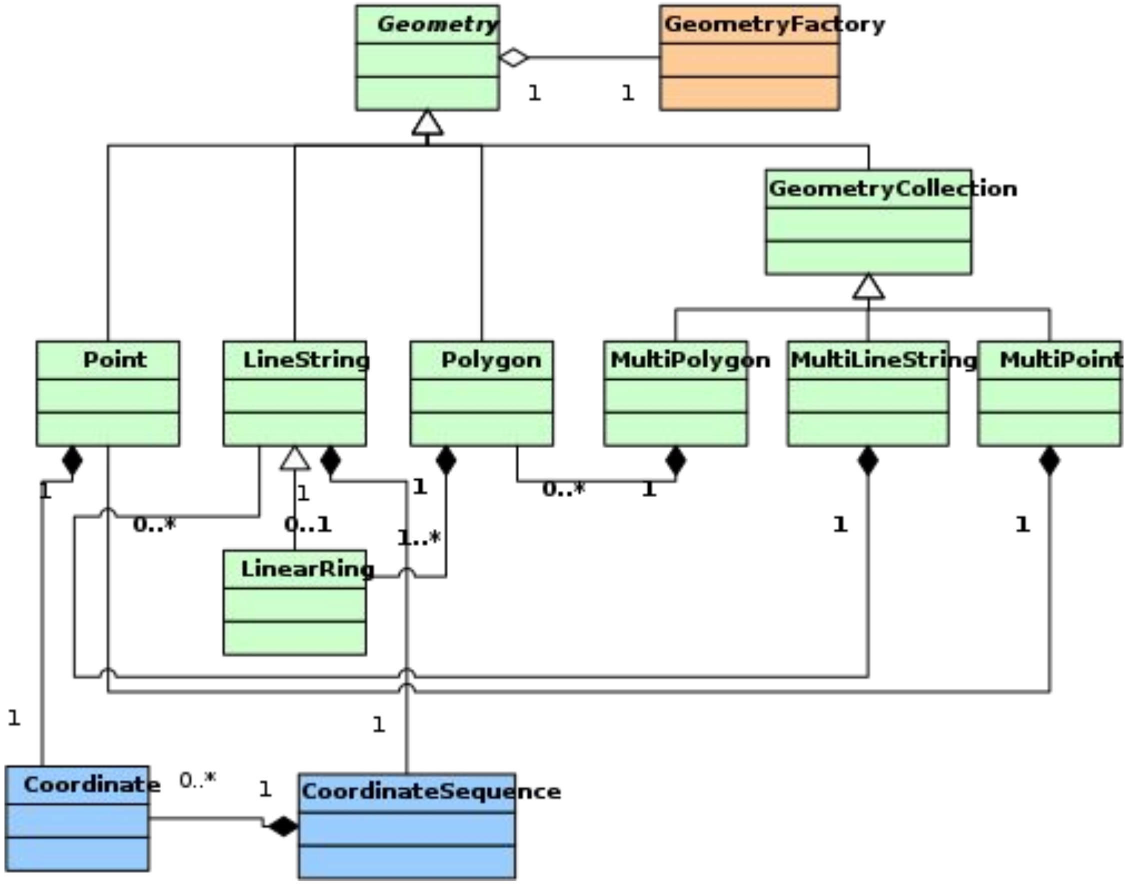
GEOS

Overview of **JTS**

Geometry Model

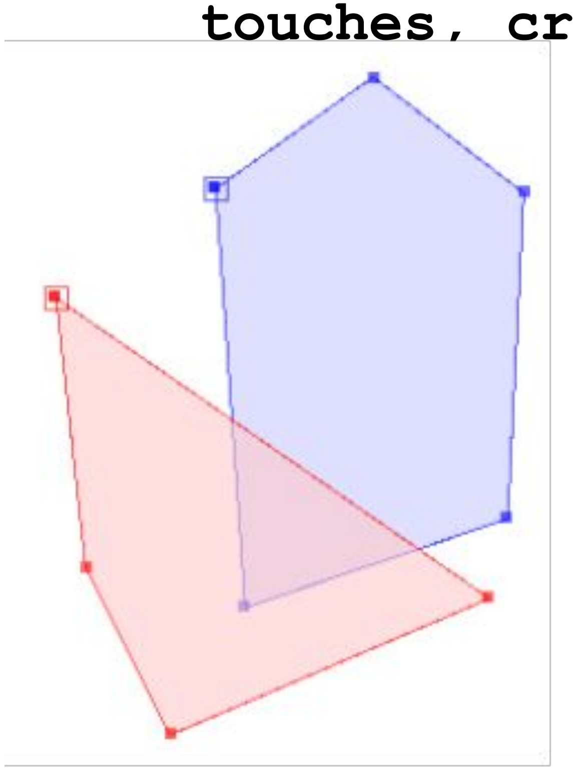
- **Complete model for 2-D linear geometry (OGC SFS model)**
  - ○ Point
  - ○ LineString, LinearRing
  - ○ Polygon (with holes)
  - ○ MultiPoint, MultiLineString, MultiPolygon
  - ○ GeometryCollection
- **User-defined coordinate representation**

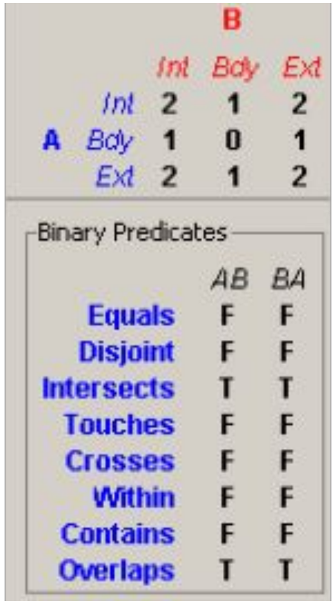




Spatial Predicates

- **Determines the spatial relationship of two Geometries**
- **Uses the *Dimensionally Extended 9-Intersection Model (DE-9IM)***
  - Computes dimension of intersection of Interior, Boundary, Exterior
- **General function**
  - `relate( IMpattern )`
- **Named predicates**
  - **`intersects, contains, within, equals, disjoint, touches, crosses, overlaps, covers, coveredBy`**

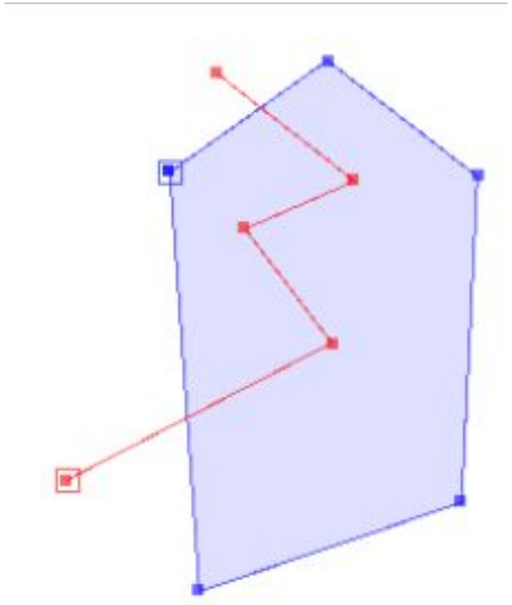


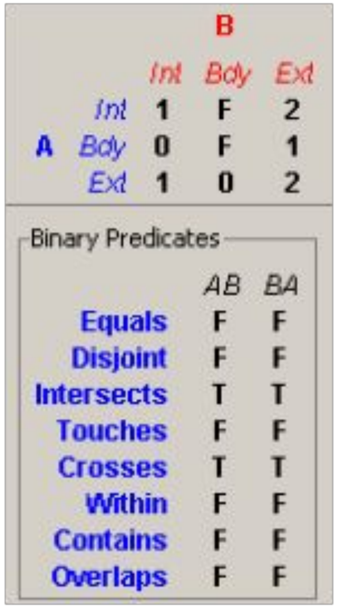


			B	
		Int	*Bdy*	*Ext*
	Int	2	1	2
A	*Bdy*	1	0	1
	Ext	2	1	2

Binary Predicates

	AB	*BA*
Equals	F	F
Disjoint	F	F
Intersects	T	T
Touches	F	F
Crosses	F	F
Within	F	F
Contains	F	F
Overlaps	T	T





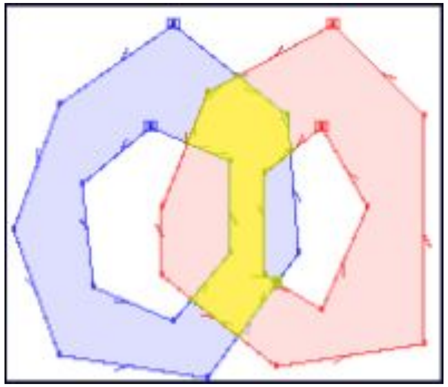
			B	
		Int	*Bdy*	*Ext*
	Int	1	F	2
A	*Bdy*	0	F	1
	Ext	1	0	2

Binary Predicates

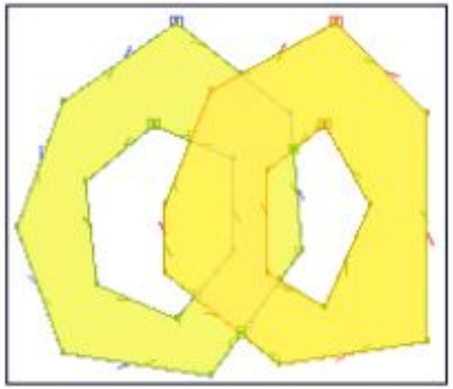
	AB	*BA*
Equals	F	F
Disjoint	F	F
Intersects	T	T
Touches	F	F
Crosses	T	T
Within	F	F
Contains	F	F
Overlaps	F	F

Overlay functions

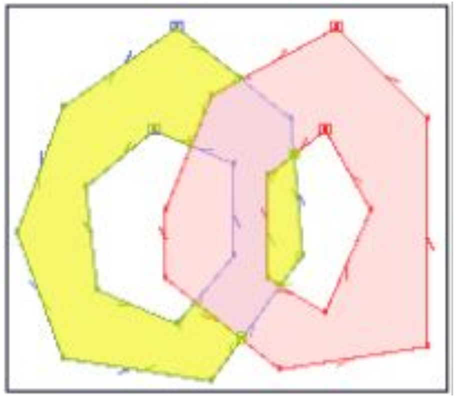
- **AKA Boolean functions, Set-theoretic functions**



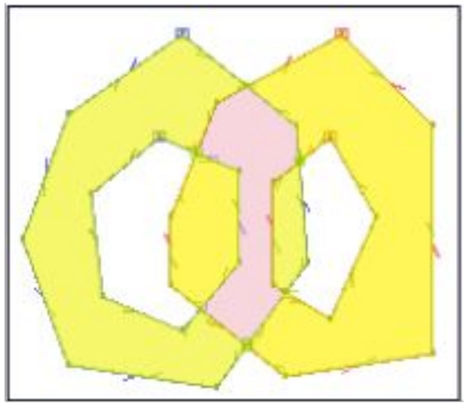
Intersection



Union

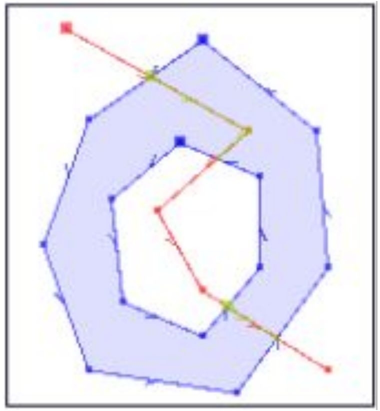


Difference



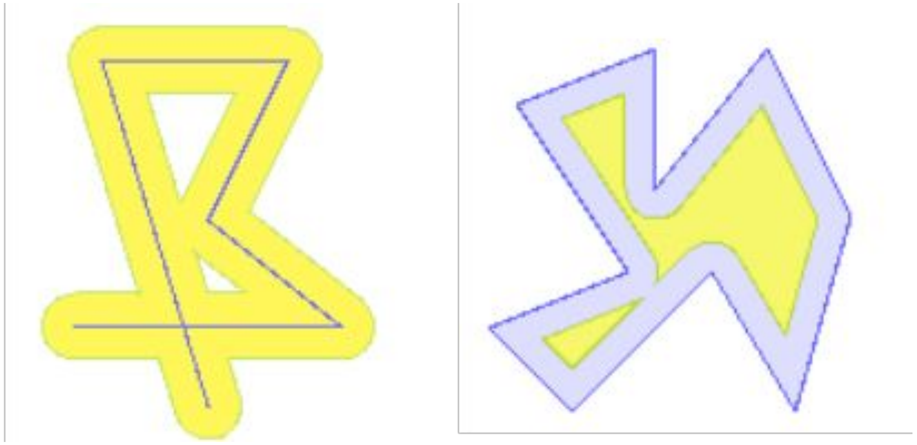
Symmetric Difference

- **Heterogeneous – all geometry types supported**

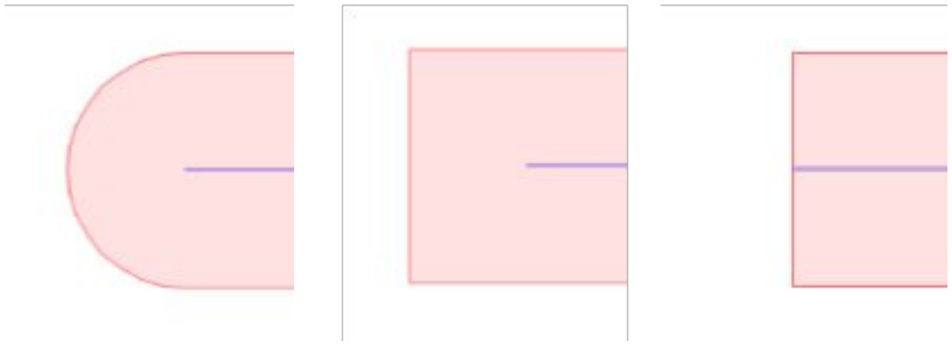


Buffers

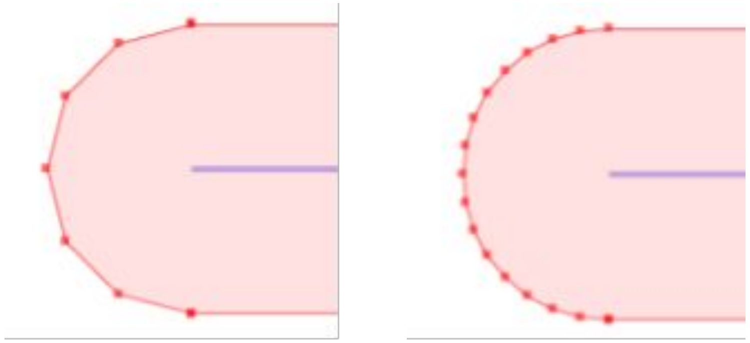
- **Positive & Negative buffers**
  - All Geometry types
  - Robust, efficient algorithm



- **Choice of End Cap Styles**
  - Round, Square, Butt

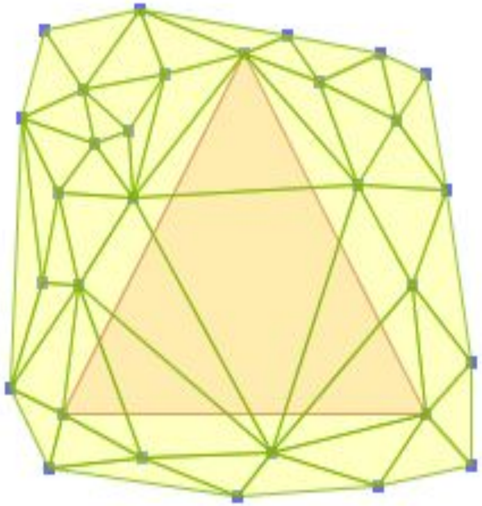


- **Curve Quantization is user-controllable**

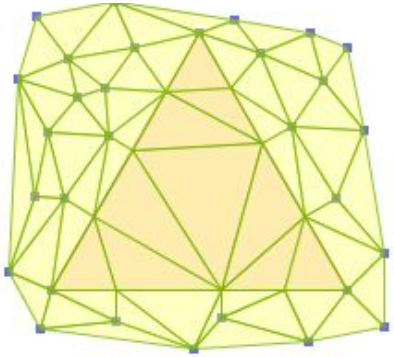


Delaunay Triangulation, Voronoi Diagram

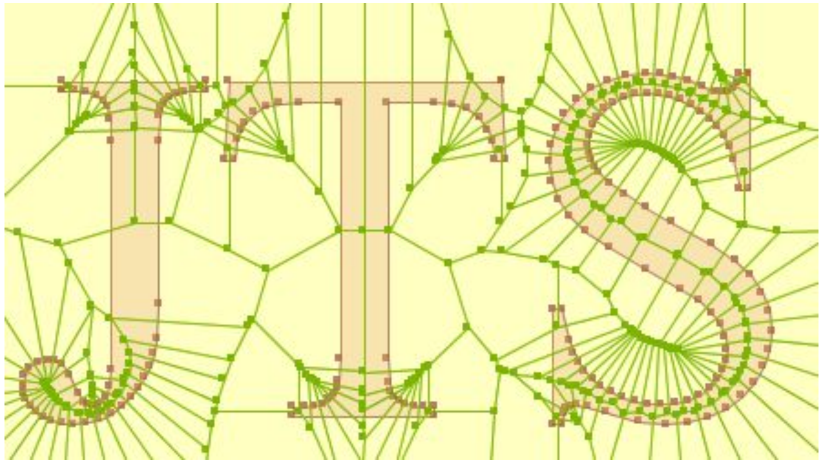
- **Delaunay Triangulation**
  - Optimal triangulation of point sets
  - Efficient, robust algorithm



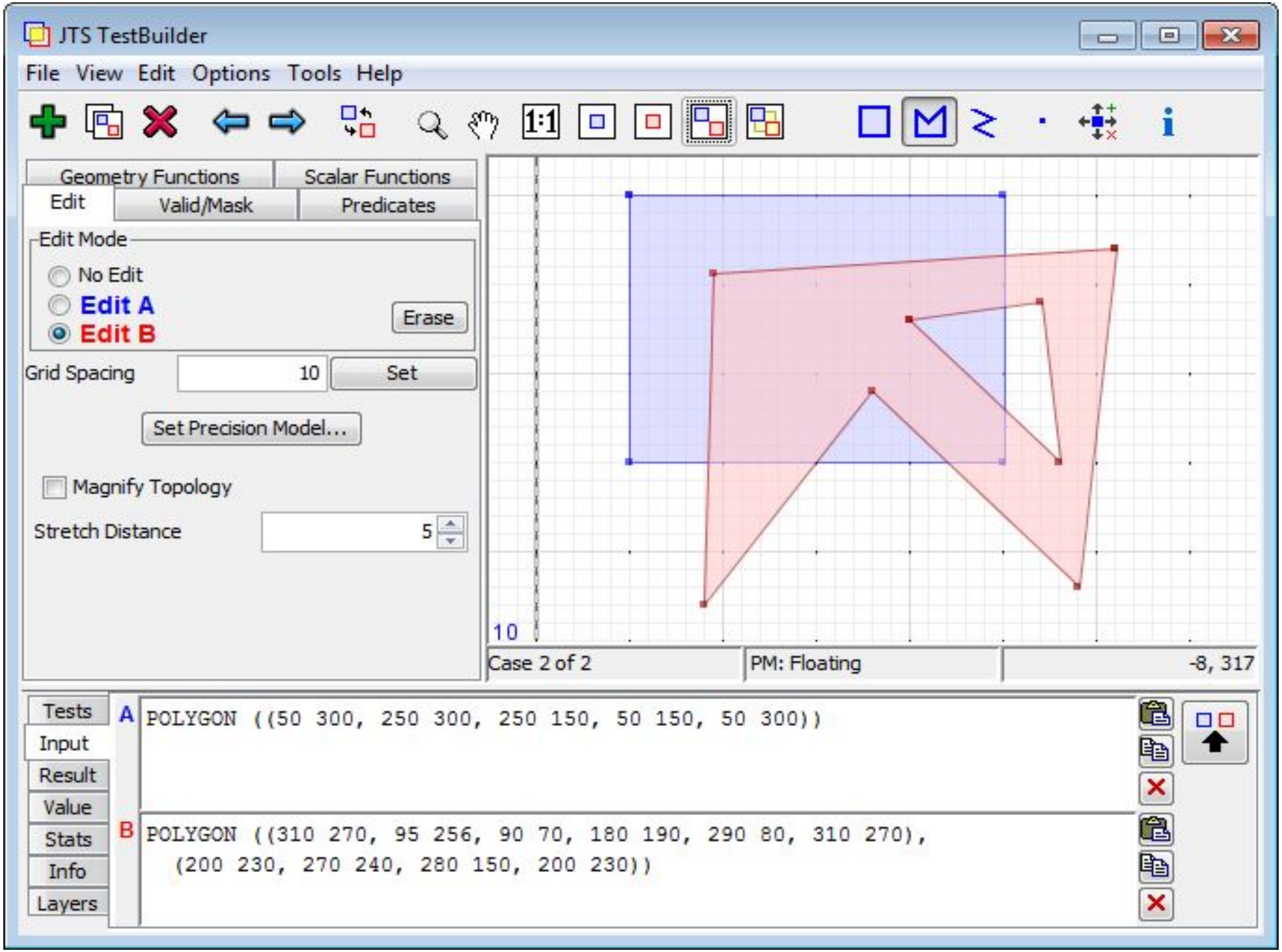
- **Conforming Delaunay Triangulation**
  - includes (approximated) linear constraints



- **Voronoi Diagram**
  - dual of Delaunay



JTS TestBuilder



Edit Mode

○ No Edit

○ Edit A

● Edit B

Grid Spacing 10

□ Magnify Topology

Stretch Distance 5

Case 2 of 2 | PM: Floating | -8, 317

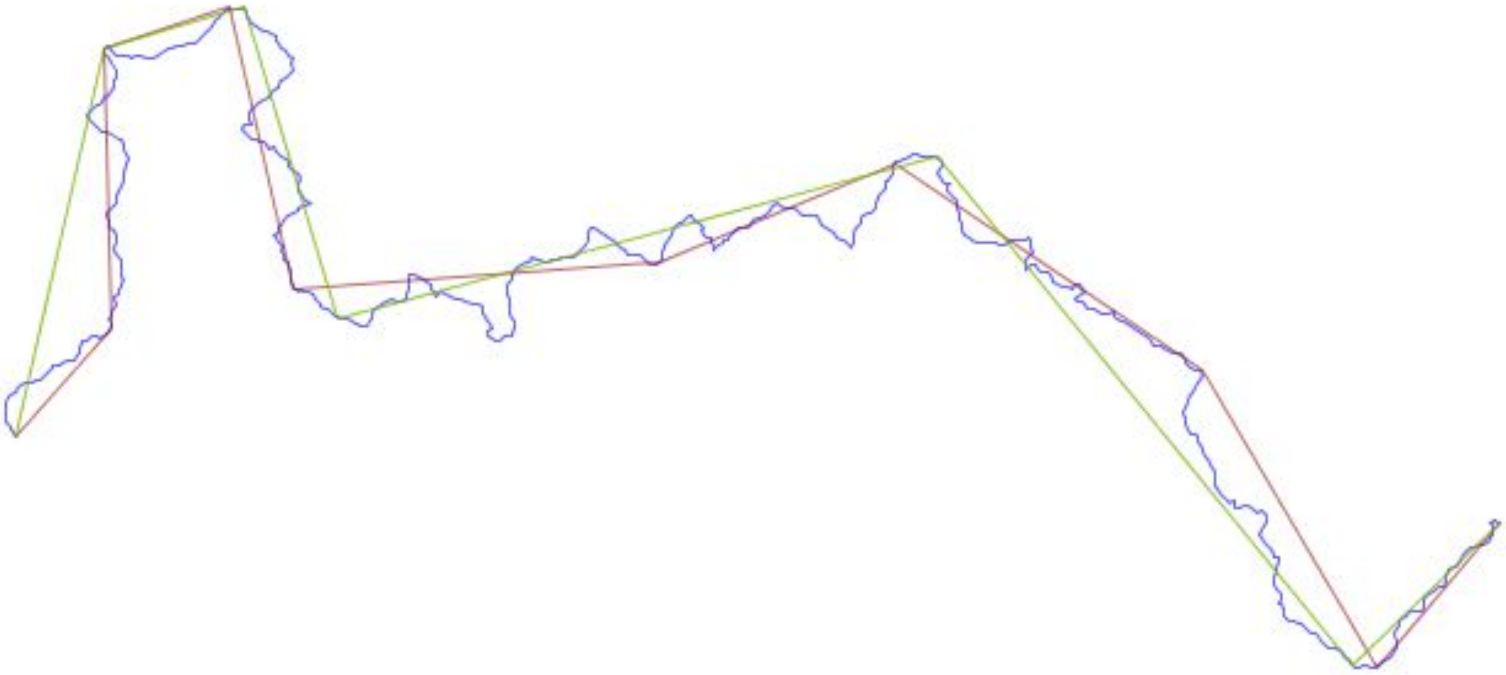
A
```
POLYGON ((50 300, 250 300, 250 150, 50 150, 50 300))
```

B
```
POLYGON ((310 270, 95 256, 90 70, 180 190, 290 80, 310 270),
  (200 230, 270 240, 280 150, 200 230))
```

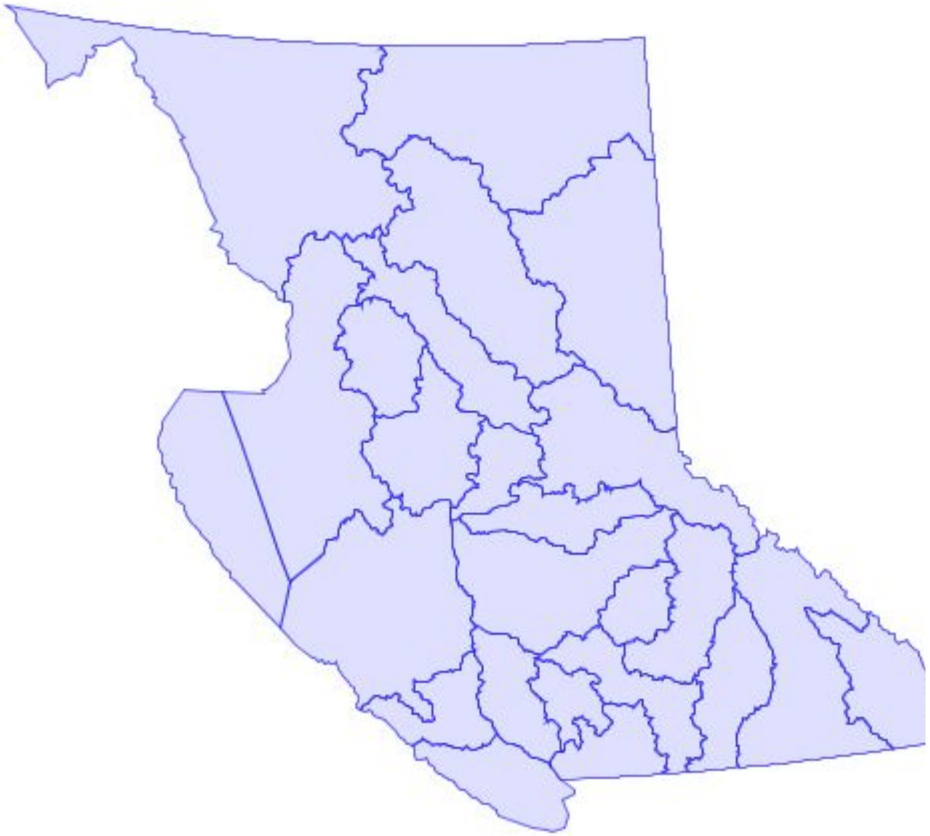
What's New in **JTS**

Visvalingam-Whyatt Simplifier

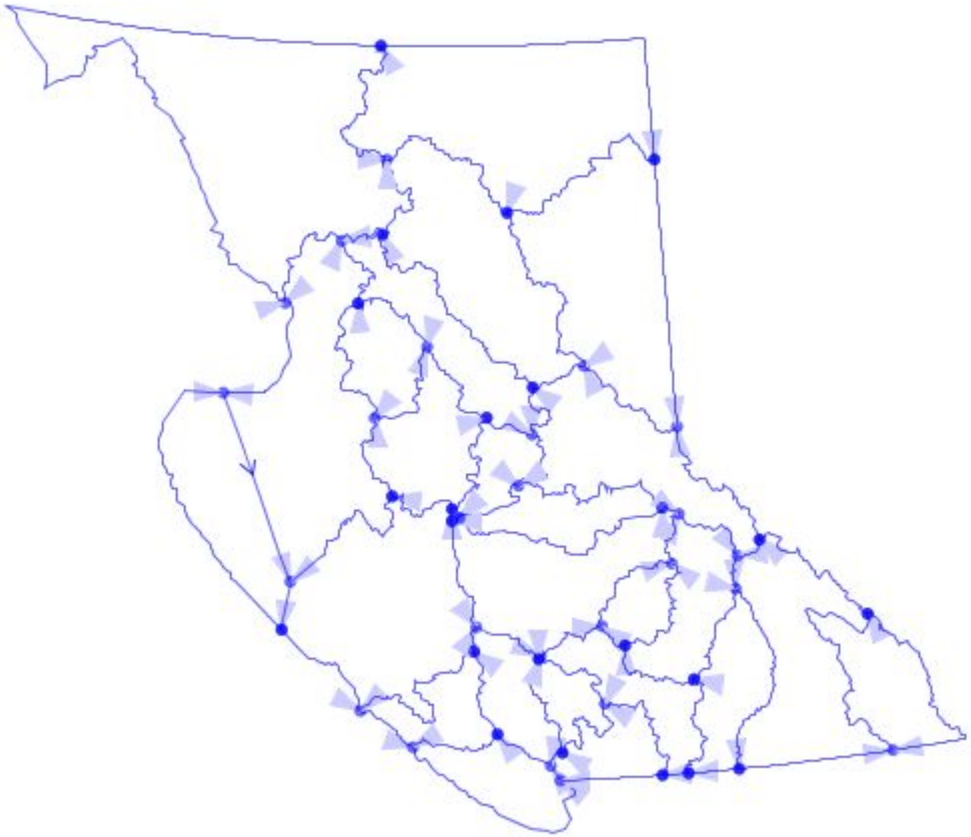
Visvalingam-Whyatt VS Douglas-Peucker



Line Dissolver



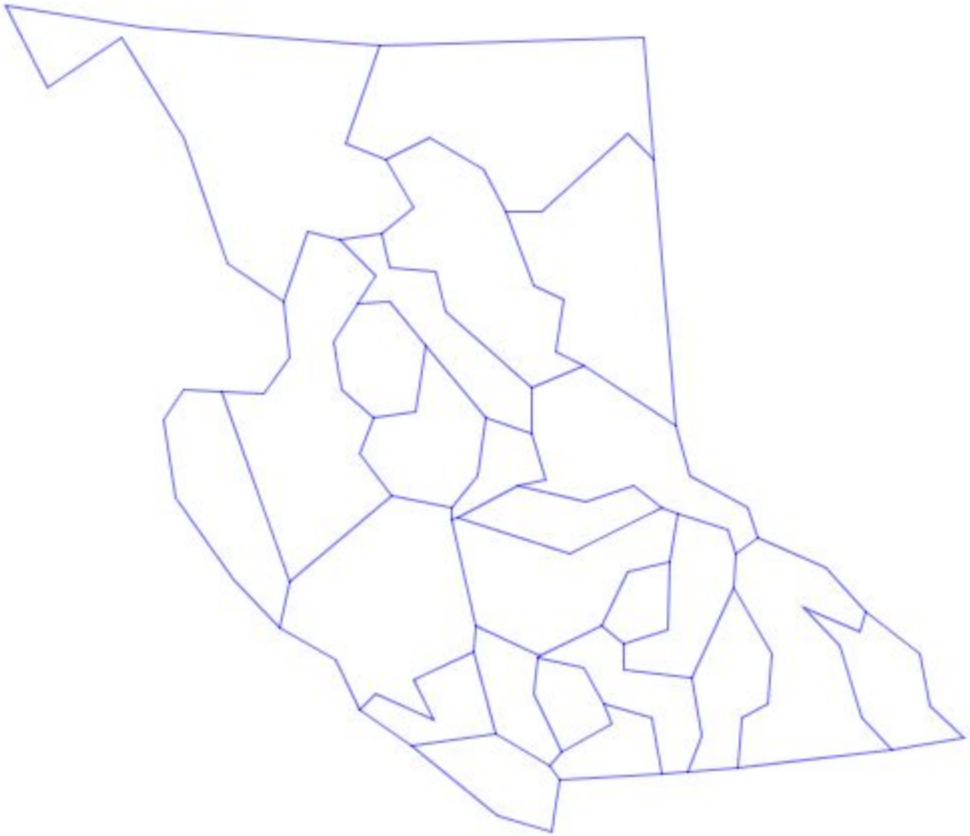
25 Polygons
949,625 vertices



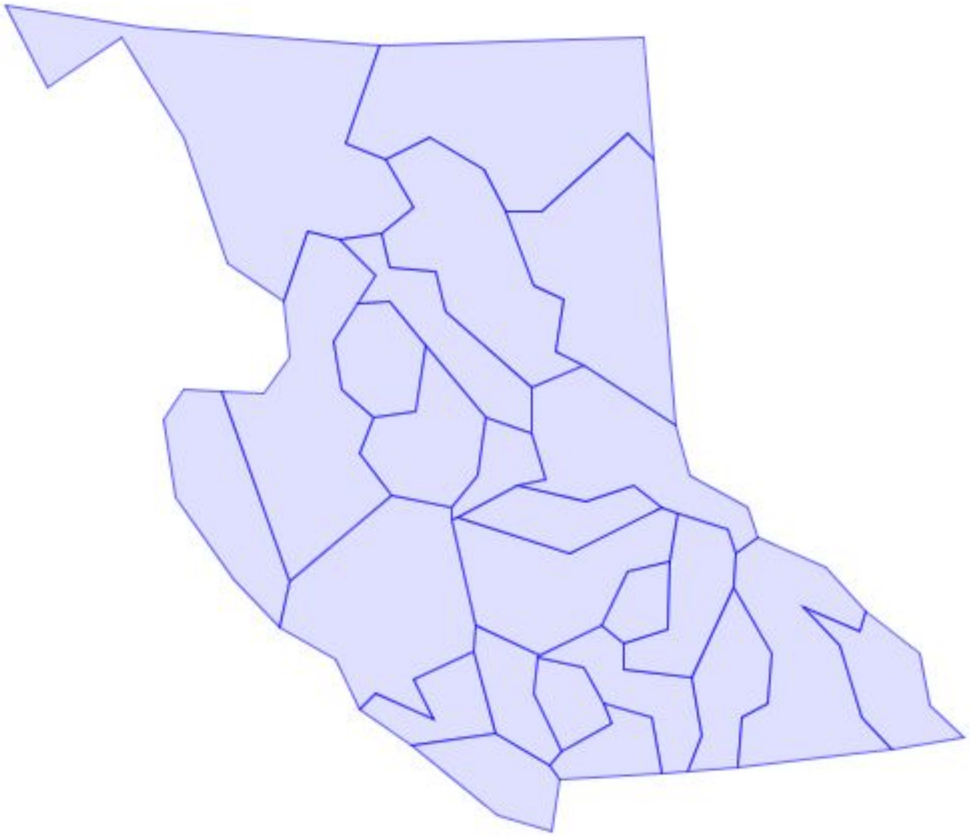
72 LineStrings
505,615 vertices

Example: Polygonal Coverage Simplification

- **Line Dissolve -> VW Simplify -> Polygonize**



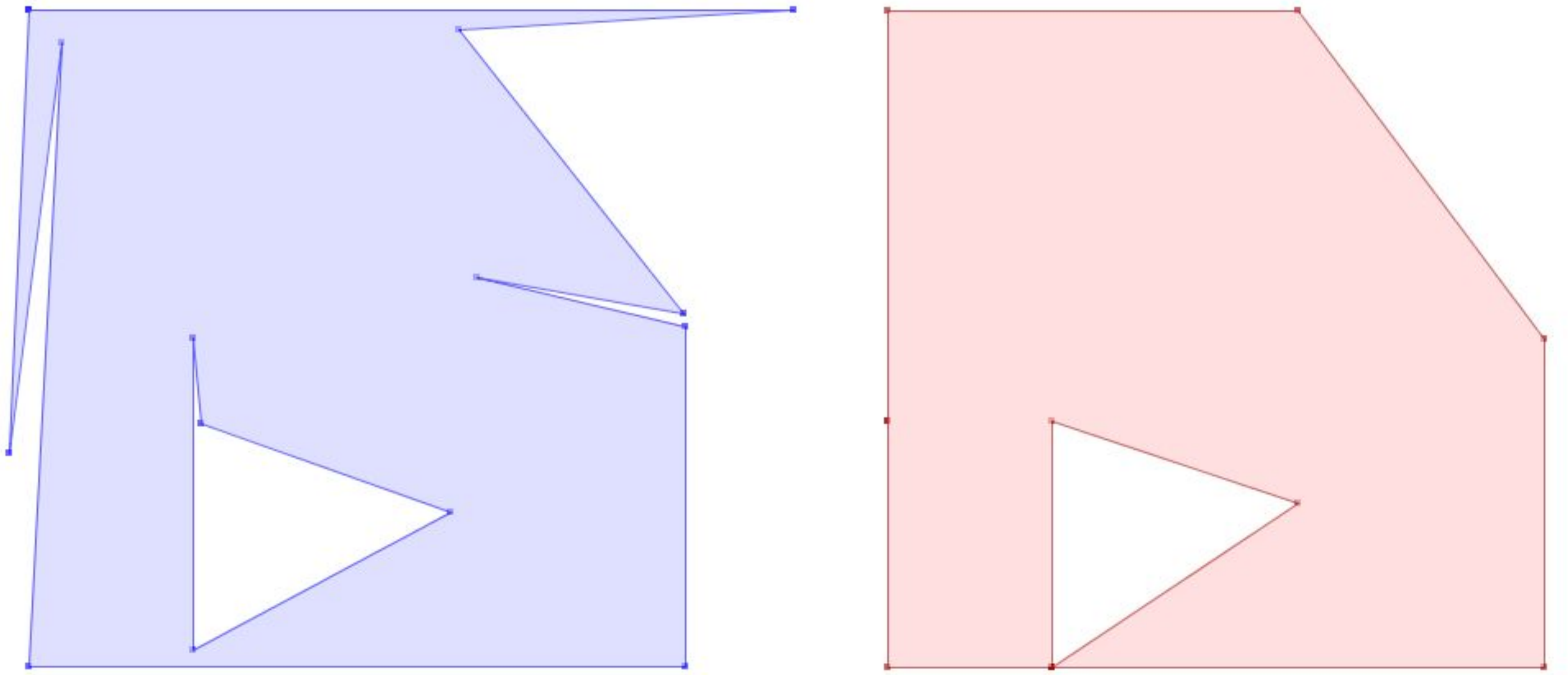
72 LineStrings
209 vertices



25 Polygons
262 vertices

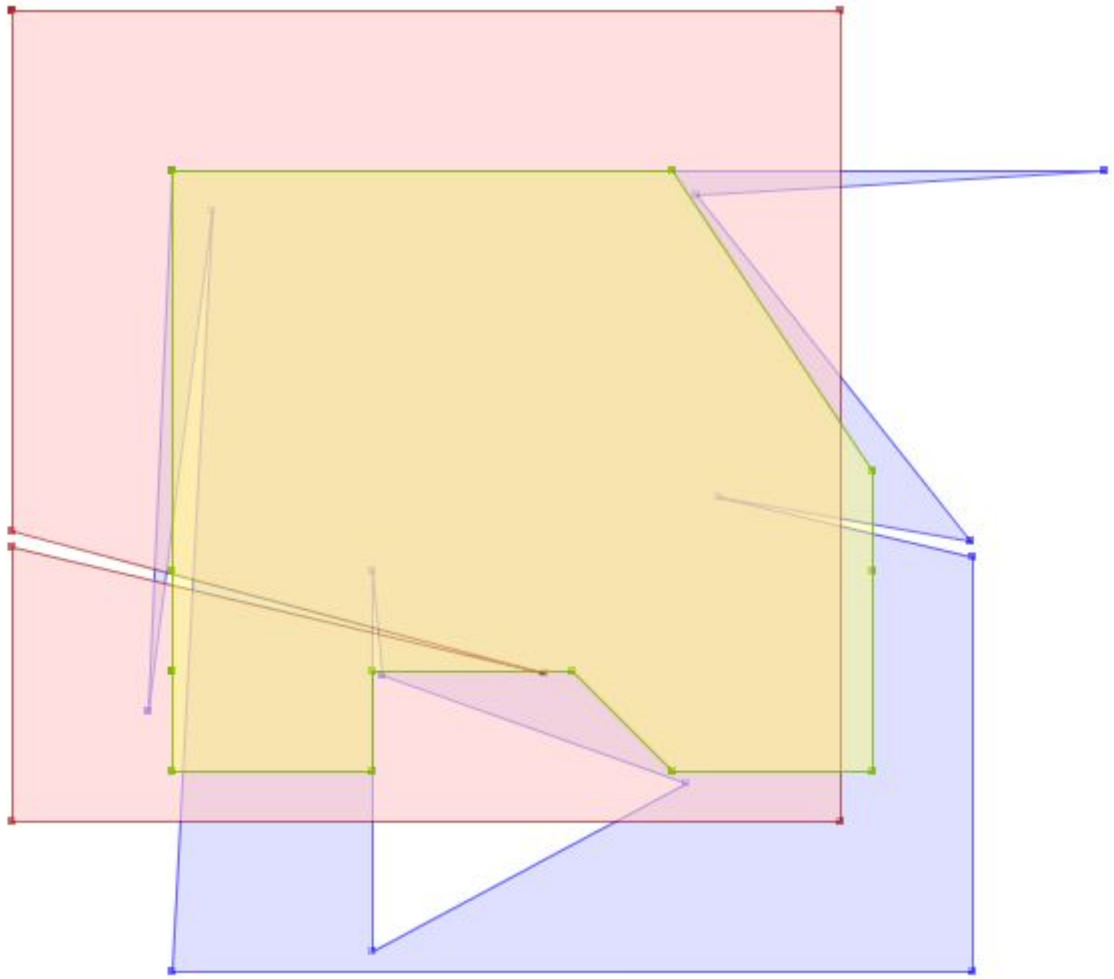
Snap-Rounded Geometry

- Snap-round geometry to precision grid
- Topology collapses are cleaned so output is valid



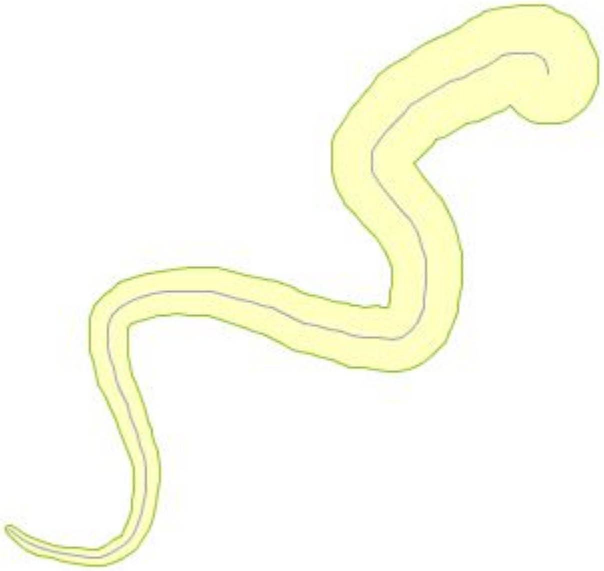
Snap-Rounded Overlay

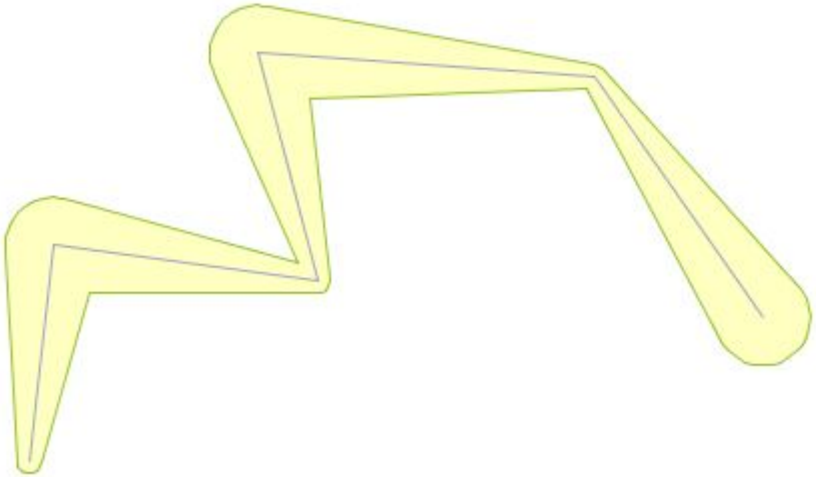
- 100% Robust !



Variable-Width Buffer

- **Variable-Width Buffer**
  - e.g. for styling linear river networks





Future Plans

- **Functionality**
  - Computation in Geodetic coordinate systems
  - Measures on coordinates
- **Deployment**
  - Split packaging into Core and Algorithms
  - Move to Maven
- **Governance**
  - Move to LocationTech
  - License change to BSD + EPL
- **JTS 2.0…**
  - Refactor `Geometry` classes to use interfaces
  - allows alternate geometry representations

Distribution & Support

- **JTS available from SourceForge**

`http://sourceforge.net/projects/jts-topo-suite/`

- **Mailing List**

`https://lists.sourceforge.net/lists/listinfo/jts-topo-suite-user`

- **Other JTS resources**
  - Javadoc
  - References
  - FAQ

`http://tsusiatsoftware.net/jts/main.html`