



Data QC– Fixing misalignments, slivers and duplicates

Miriam Schmidts

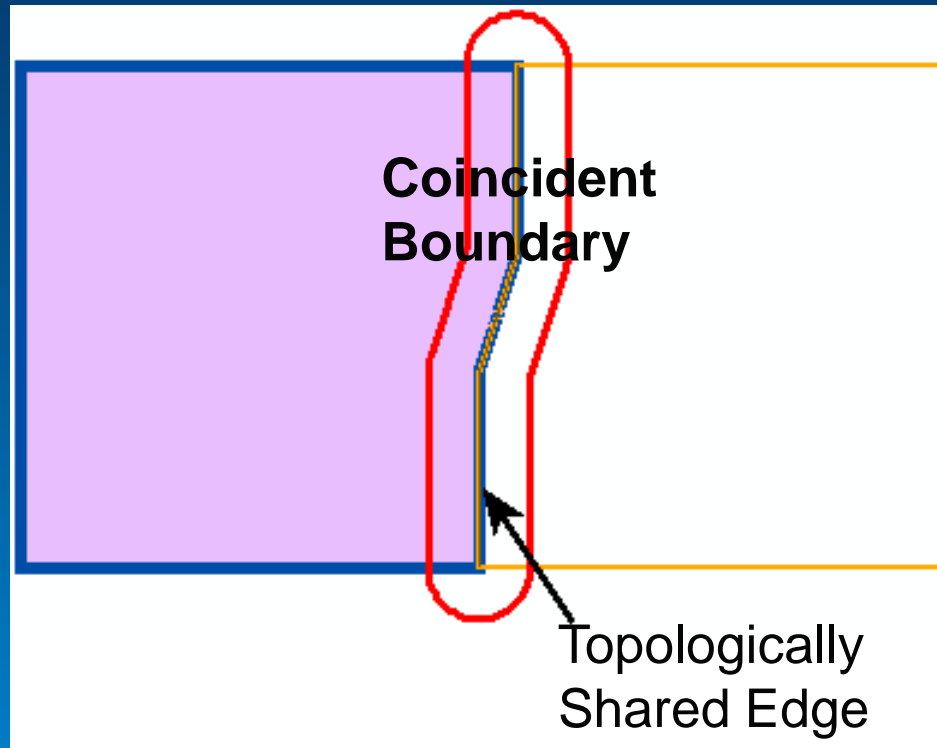
Lesson 4



Learning objectives

- **Align features using**
 - **Align to Shape**
 - **Map topology**
 - **Geodatabase topology**

Coincident and shared geometry



Using map topology

- **Maintains coincidence during an edit session**
 - Between features in one feature class
 - Between features in different feature classes
- **Cluster tolerance**
 - Makes vertices coincident
- **During edit session only**
- **For simple features only**
 - No annotations, dimensions, geometric networks
- **For geodatabase feature classes or shapefiles**

Why map topology?

- **Fast**
 - Create on-the-fly when needed
 - Specify feature classes to participate
- **Works with shapefiles or geodatabase feature classes**
 - Shapefiles must be in the same folder
 - Feature classes must be in the same geodatabase
 - Feature classes may participate in a geodatabase topology
- **Available with all license levels**

Map topology workflow

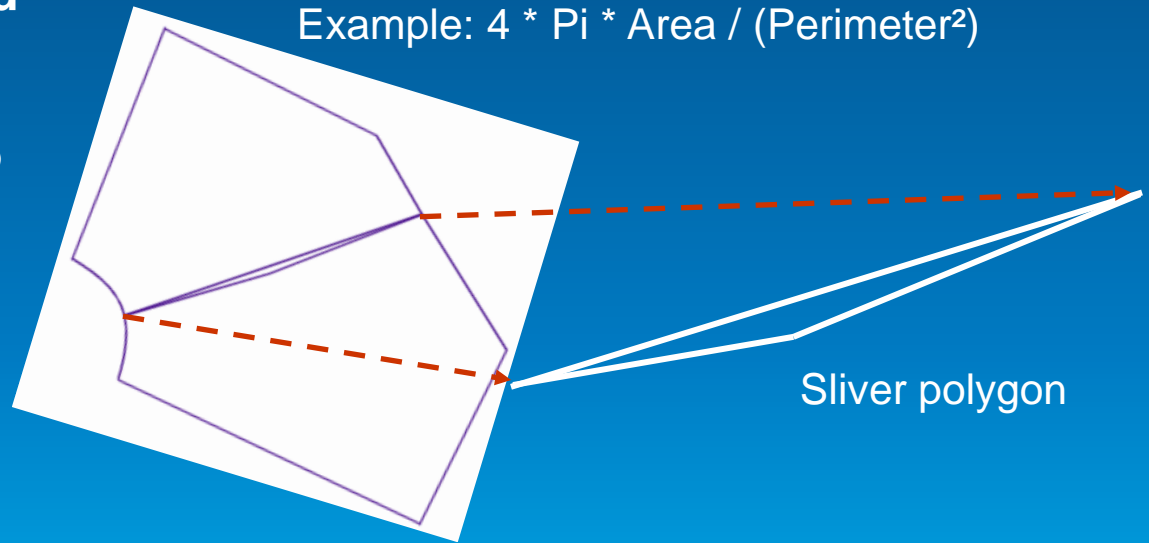
- **Start an edit session**
- **Build the map topology**
 - **Select layers**
 - **Set cluster tolerance**
- **Set topology node snapping as needed**
- **Select topology element to edit**
 - **Remove shared features as needed**
- **Perform the edit**
- **QA/QC**
 - **Edit features if necessary**

Checking for tiny and sliver polygons

- May be valid features, but should be inspected
- Tiny polygons
 - Query features based on a minimum area threshold
- Sliver polygons
 - Calculate thinness ratio for each polygon

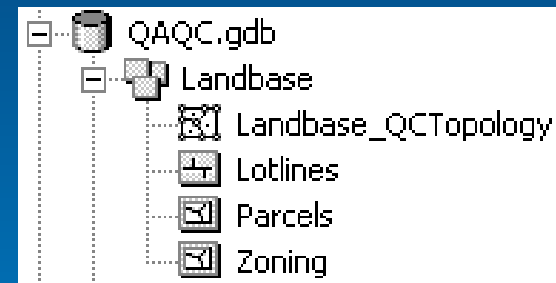
$$\text{Pi} = 3.14159265$$

$$\text{Example: } 4 * \text{Pi} * \text{Area} / (\text{Perimeter}^2)$$



Geodatabase Topology

- **Topology has four parameters**
 - Cluster tolerance
 - Participating feature classes
 - Rules
 - Ranks
- **Validation is used to discover topology errors**
 - Validation process is iterative
- **Participating feature classes must be in the same feature dataset**



Topology rules

Define valid spatial relationships

Between features in a single feature class

Between feature classes

Between subtypes

Set as properties of the topology

Apply to dataset during validation

Must not overlap

Polymons must not overlap within a feature class or subtype. Polymons can be disconnected or touch at a point or touch along an edge.



Polygon errors are created from areas where polygons overlap.

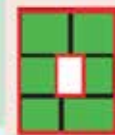
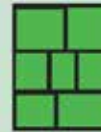
Use this rule to make sure that no polygon overlaps another polygon in the same feature class or subtype.



A voting district map cannot have any overlaps in its coverage.

Must not have gaps

Polymons must not have a void between them within a feature class or subtype.



Line errors are created from the outlines of void areas in a single polygon or between polygons. Polygon boundaries that are not coincident with other polygon boundaries are errors.

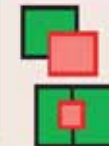
Use this rule when all of your polygons should form a continuous surface with no voids or gaps.



Soil polygons cannot include gaps or form voids—they must form a continuous fabric.

Must be covered by

Polymons in one feature class or subtype must be covered by a single polygon from another feature class or subtype.



Polygon errors are created from polygons from the first feature class or subtype that are not covered by a single polygon from the second feature class or subtype.

Use this rule when you want one set of polygons to be covered by some part of another single polygon in another feature class or subtype.



Courties must be covered by states.

Demo

Align features



Exercise introduction

- **Align features using**
 - **Align to Shape**
 - **Align edge**
- **Use geodatabase topology to find and fix spatial errors**
- **Find sliver polygons using the thinness ratio**