Maps as Numbers: Data Models

- **Reality**
  - Nodes (S, E)
  - Vertices (S, E)
- **Conceptual Models**
  - Wells are points, rock units are polygons (both are objects)
  - Well A penetrates Fm. 1; produces oil. Well B penetrates Fm. 3; produces gas. Fm 3 overlies Fm. 1.
  - Store well locations with a particular file structure, production stats. in a dBase table. Associate table with location.
- **Logical Models**
- **Physical Models**

**The Task**

- An accurate, registered, digital map that can be queried and analyzed.

**Translate:**
- Real World, Paper Map ➔ Computer Files
  - Spatial Data Models, Topology
  - Entity Info. ➔ Queriable Database Files
  - Relational or Object-Oriented Databases
- Relate Spatial Coordinates to Entity Info.

“**Spatial DBMS**” software = GIS software!

**Data Models**

- **How is reality abstracted and codified?**
  - **Reality**
  - **Conceptual Models**
    - Wells produce from rocks that contain oil and gas
    - Well A penetrates Fm. 1; produces oil. Well B penetrates Fm. 3; produces gas. Fm 3 overlies Fm. 1.
  - **Logical Models**
  - **Physical Models**

**Conceptual Models**

- Characterized all features or phenomenon as:
  - **Discrete objects; e.g. wells, roads, rock bodies, etc.**
    - **Object-based models**
  - **Continuous phenomena; e.g. gravity, magnetic intensity, topography, temperature, snowfall, soil pH, etc.**
    - **Field-based models**
**Logical Models**

**VECTOR MODEL**
- Discrete objects are represented by points and vectors, continuous fields by irregular tessellations of triangles (TINs)

**RASTER MODEL**
- Discrete objects and continuous fields are represented by an array of square cells (pixels)

**How should discrete objects be coded?**

**Continuous phenomena as surfaces**

**Raster Topography**
- Regular tessellations, e.g. DEM

**Vector Topography**
- Irregular tessellations, e.g. T.I.N.
Maps as Numbers

Physical Models

Simple Vector Data Structure

Vector Line

Table of Points

<table>
<thead>
<tr>
<th>ID</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>503200</td>
<td>3200522</td>
</tr>
<tr>
<td>P2</td>
<td>503250</td>
<td>3200522</td>
</tr>
<tr>
<td>P3</td>
<td>503300</td>
<td>3200460</td>
</tr>
<tr>
<td>P4</td>
<td>503245</td>
<td>3200410</td>
</tr>
<tr>
<td>P5</td>
<td>503350</td>
<td>3200410</td>
</tr>
</tbody>
</table>

(in UTM coordinates)

Physical Models

Simple Raster Data Structure:

Equivalent Binary Flat File

```
1 1 0 0 0 0
0 0 1 0 0 0
0 0 0 1 0 0
0 0 1 0 0 0
0 1 1 1 1
```

Physical Models

Vector Models

- “Graphical”
- Topologic/georelational
- T.I.N.
- Network

“Graphical” Vector Model

- Lines have arbitrary beginning and end, like spaghetti on a plate
- Common lines between adjacent polygons duplicated
- Can leads to “slivers” of unassigned area = “sliver polygons”
Maps as Numbers

“Graphical” Vector Model

- Shortcomings for maps:
  - No real world coordinates required
  - No identification of individual objects; no way to attach attributes
  - Details of relationships among objects (e.g. what’s adjacent) not stored, but needed for spatial analysis

Graphical Vector Structure

- Contains no explicit information about adjacency, containment or contiguity i.e.
  - Which polygons are adjacent?
  - Which polygons are contained within other polygons?
  - Which lines are connected? Where are they connected? Where do lines begin and end?

= “Spaghetti Data Model”

Topological Vector Model

- Store pts. as x,y geographic coordinates
- Store lines as paths of connected pts.
- Store polygons as closed paths

  Also explicitly store ....

- Where lines start and end (connectivity)
- Which polygons are to the right and left side of a common line (adjacency)

Topology

- The geometric relationship(s) between entities (e.g. points, lines, areas); where is one thing with respect to another?
Topological Properties

- Spatial characteristics that are unchanged by transformations like scaling, rotation and translation are topologic
  - **Non-topological**: x, y coordinates, area, distance, orientation
  - **Topological**:
    - Contiguity – what’s adjacent
    - Connectivity – what’s connected
    - Containment – what’s inside or outside of a region

Unchanged by translation, scaling, rotation

Maintaining Topology: Planar Enforcement

- One and only one feature at every x, y location
  - Lines cross at nodes; polygons space-filling, exhaustive, mutually exclusive (no overlaps or gaps)
  - Sum of the area of all individual polygons equals the area of extent of all polygons
  - Common boundaries stored only once

- A PLANAR GRAPH meets these conditions
- Allows spatial queries for adjacency, containment and rapid what-is-where
- All raster data is of this sort

Non-Planar vs. Planar Graphs

<table>
<thead>
<tr>
<th>Spaghetti</th>
<th>Topologic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Survey A</td>
<td>1</td>
</tr>
<tr>
<td>Survey B</td>
<td>3</td>
</tr>
<tr>
<td>Survey C</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

After Bonham-Carter, 1994

<table>
<thead>
<tr>
<th>Polynomials</th>
<th>Survey A</th>
<th>Survey B</th>
<th>Survey C</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Maps as Numbers

9/6/2016

M. Helper, GEO 327G/386G, UT Austin

6

Lines: Graphic vs. Topologic

*Graphic (Spaghetti)*

- vertices
- overshoot

**Topologic (with meat-balls)**

- vertices
- Table of (x,y) coordinates

S – Start node
E – End node

Lines: Arc-Node Topology

**Table of (x,y) coordinates**

- S
- E

**Table of arcs with IDs, starting and ending nodes**

- F
- T

Arc Table

<table>
<thead>
<tr>
<th>ID</th>
<th>FID</th>
<th>Arc ID</th>
<th>Poly</th>
<th>Start</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>A1</td>
<td>P1</td>
<td>N1</td>
<td>N2</td>
</tr>
<tr>
<td>2</td>
<td>102</td>
<td>A2</td>
<td>P1</td>
<td>N2</td>
<td>N1</td>
</tr>
<tr>
<td>3</td>
<td>103</td>
<td>A3</td>
<td>P2</td>
<td>World</td>
<td>N2</td>
</tr>
</tbody>
</table>

Why bother with topology?

* Provides a way of error trapping and geometry validation after data entry
  - All lines must meet at nodes, all polygons must close, polygons can’t overlap, all lines in a network must join
* Permits spatial queries, precise measurements

Polygons: Polygon-Arc Topology

**Arc Table**

- Poly ID
- FID
- Arcs

**Polygon Table**

- Arc ID
- Poly
- Start
- End

**Arc Coordinates Table**

- Arc
- Start
- Vertices
- End

- A1
- N1: v7, v11, ...
- N2

- A2
- N2: v8
- N1

- A3
- N2: v1, v2, ...
- N1

- P1
- N1

- P2
- N2

- P3
- N2
What kind of queries does topology permit?

- **Connectivity**
  - What is shortest path between features or locations? (networks, flow)
  - Find all fault trace intersections

- **Contiguity**
  - What’s adjacent: e.g. Show all granite/limestone contacts
  - Combine all contiguous units with a specific attribute (e.g. lithology) into a single unit

- **Containment (= “Area Definition”)**
  - What proportion of an area is underlain by a specific rock type?
  - What is spatial density of specific feature(s)?

Vector Models

- **Graphical ✓**
- **Topologic/“georelational” ✓**
- **T.I.N.**
- **Network**

Triangulated Irregular Network - TIN

- Topological 3-D model for representing continuous surfaces using a tessellation of triangles

Triangular Irregular Network

- Network ("tessellation") of interlocking triangles made from irregularly spaced points with x, y and z values
- Density of triangles varies with density of data points (e.g. spacing of contours) - c.f. raster with uniform data density
- Triangle sides are constructed by connecting adjacent points so that the minimum angle of each triangle is maximized (see "Delaunay Triangulation" for details)
- Can render faces, calculate slope, aspect, surface shade, hidden-line removal, etc.
TIN Topology

Node Table

<table>
<thead>
<tr>
<th>Node</th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>12</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>13</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>7</td>
<td>10</td>
</tr>
</tbody>
</table>

Tin Topology Table

<table>
<thead>
<tr>
<th>Triangle</th>
<th>Node-list</th>
<th>Neighbors</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1, 2, 6</td>
<td>C, E</td>
</tr>
<tr>
<td>B</td>
<td>2, 3, 4</td>
<td>C</td>
</tr>
<tr>
<td>C</td>
<td>2, 4, 6</td>
<td>B, D, A</td>
</tr>
<tr>
<td>D</td>
<td>4, 5, 6</td>
<td>E, C, A</td>
</tr>
<tr>
<td>E</td>
<td>5, 1, 6</td>
<td>A, C, D</td>
</tr>
</tbody>
</table>

After Zeiler, Modeling our World, p. 165

TIN for Seiad Valley, CA

3-D TIN Scenes of Seiad Valley fault

3-D TINS, Grand Canyon

3-D TINS, Grand Canyon at Bright Angel Creek

Bright Angel Trail
Vector Models

- Graphical ✓
- Topologic/"georelational" ✓
- T.I.N. ✓
- Network - not discussed, see Help files