Dr. Jim Graham and Chris Muhl
Humboldt State University, 2014
The Need

• Error varies greatly between projections and their settings
• Computing error in projections can be time-consuming
• Projections can be difficult for students to appreciate
Projections greatly distort area, distance, and/or shape (form).

Mercator
0 to 50 area and distance distortion
• The nature of distortion can be difficult to describe in text
Clipping bounds are also needed.

Cassini Soldner Projection

Default

Limited to 0 to 2x distance distortion
Previously, to Compute Area Distortion

• Create “Fishnet” of polygons
• Project to equal-area projection
• Compute “Exact” areas
• Project to desired projection
• Compute projected areas
• Divide the exact by projected area values
  – <1: area was made much smaller than expected
  – >1: area was made larger than expected
Today:

- Selection projection
- Enter desired settings
- Press “Update”
- “OK” to add layers
Albers Equal Area Conic

Preserves Area and Form

Distorts Distance
The Approach

- **BlueSpray**
  - Created by SchoonerTurtles, Inc.
  - Provided free under a beta testing agreement

- **GDAL, GeoTools**
  - Open source projection engines

- **Also:**
  - Java from Oracle
  - Java Topology Suite
  - NetBeans
• Create grid of points along lines of latitude and longitude (parallels and meridians)
• Compute:
  – Great circle area
  – Great circle distances (along meridians and parallels)
  – Angles are at intersections are 90 degrees except for the poles
• Project the grid of points to desired projection
• Compute area of “cells” between points
• Divide by expected area
• Compute the length of each line segment
• Divide by expected length
• Compute the average change in angle at each point
  – Sum the angle between all the line segments at each point
  – Divide by the number of angles to find the average angle
  – Divide by expected value of 90 degrees
Finding the bounds

• Start at the center (0,0)
• Moving left and right two cells then up and down one cell:
  – Add cells that are within the specified tolerances
  – Check for overlapping points
  – Check for intersecting lines
Adding “Cells”

Assumed the first four cells were within tolerances
Adding “Cells”

<table>
<thead>
<tr>
<th>1</th>
<th></th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td>4</td>
</tr>
</tbody>
</table>

Add cells to left and right that are within tolerance
Adding “Cells”

Add additional cells to the left and right since the world is twice as wide as it is call (with Geographic data)
Adding “Cells”

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>10</td>
<td>9</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>1</td>
<td></td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td></td>
<td>4</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>16</td>
<td>15</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
</tbody>
</table>

Add cells along the top and bottom. Keep repeating the cycle until no more cells are added.
Disallow Intersections and Overlaps
Limit Distortion

0 to 2x on area and distance
Details

• Accessing vector data from an applet
  – Used BlueSprays “stx” format

• Java Topology Suite is very picky

• Projected Systems can extend beyond ±180
to ±90 degrees
  – Used a 360*3, 180*3 sized grid for analysis
Orthographic

Polar

Stereographic

Robinson

Polyconic

Orthographic
Available At:

• Applet:
  – HSU Geospatial Web Site
  – www.humboldt.edu/gsp -> Links

• BlueSpray:
  – SchoonerTurtles web site:
  – www.schoonerturtles.com
Future Steps

• Add the ability to project from any layer
  – Not just the globe
• Finish projection engine within BlueSpray
  – Uses the Projection Explorer to set the bounds
Acknowledgments

• Jake Nelson, Dr. Bernhard Jenny at Oregon State University
• Nick Ramirez at Humboldt State University
• Greg Newman at Colorado State University
• All the folks that support open source GIS software: GDAL, GeoTools, Proj4, JTS, NetBeans, etc.