# Spatial Reference Systems

Definitions (OGC/ISO-19100 series)

- **Coordinate operation**: a change of coordinates, based on a one-to-one relationship, from one coordinate reference system to another.
- **Conversion**: is a coordinate operation in which both coordinate reference systems are based on the same datum, e.g. change of units of coordinates.
- **Transformation**: is a coordinate operation in which the two coordinate reference systems are based on different datums, e.g. a change from a global reference frame to a regional frame.
- **Projection**: is a coordinate conversion from an ellipsoidal coordinate system to a plane.
  - **Note**: Although projections are simply conversions according to the standard, they are treated as separate entities in PROJ as they make up the vast majority of operations in the library.

Projection surfaces

Map Projection Families
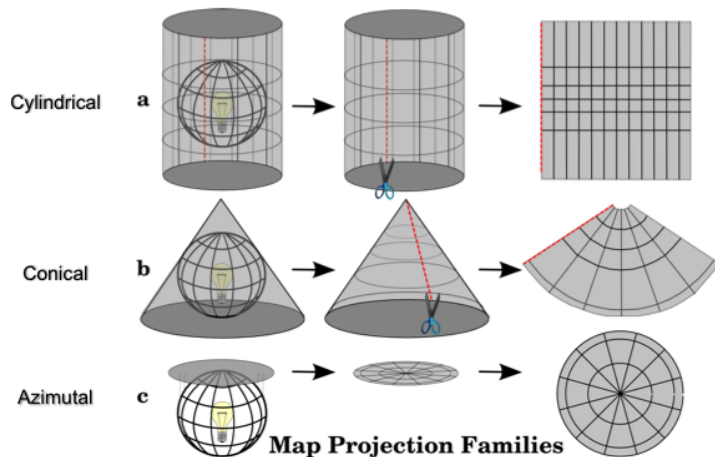
Choice of projection

- **Purpose of the map**
  - **Area covered** (USA, Australia, Europe, Polar regions, Equatorial regions, linear shape, …)
  - **Scale of representation** (continent, state, city)
  - **Use** – what distortions we allow/not allow
    - Equal area projections: for measuring/comparing areas
    - Conformal projection: measuring local angles (cadastral app)
    - Gnomonic: great circles display as straight lines
    - Equidistant projection: preserves proportions of distances and some distances along parallels

Conversion: Cartesian to Geographical CRS

- **Cartesian** 3D Coordinates
  - Earth centered (X,Y,Z),
- **Polar 3D coordinates**:
  - Latitude: angle north or south of the equatorial place
  - Longitude: angle east or west from the prime meridian
  - Height: distance above the surface of sphere/spheroid/geoid
  - → Parallels and Meridians
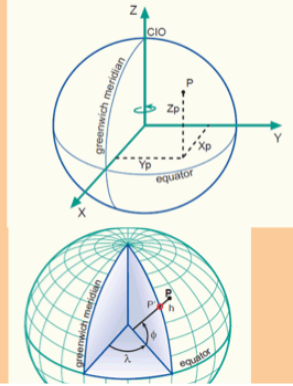
Transformation

- **Transformation is the conversion (mapping) of the values of coordinates in one reference system into another;**
- **The values of the coordinates are <u>numerically</u> different;**
- **Affine transformations - linear mapping method, preserving straight lines and planes (7 parameters):**
  - Rotation (3 parameters – rotation around x, y and z)
  - Translation (in 3D – translation vector x,y,z)
  - Scaling (1 parameter)

There are hundreds of SRS…

Defining parameters of transformations for each pair would be infeasible (m:n)

We (Proj) define the parameters from the current SRS to a world geographic SRS: WGS84
- *SRS1 → WGS84 → (-1) SRS2*

In PostGIS: ST_Transform(geom, targetSRS)

Most geodetic transformations are a series of elementary operations, the output of one operation being the input of the next.

Coordinate transformations – going from one system to the other (similar type)
- May involve datum change
- From polar to 3D earth centred cartesian
- From polar to 2D cartesian (Map **projection**)
- Map re-projections (from one 2D cartesian to another – via the polar)

Point

A point in the Cartesian plane $\mathbb{R}^2$ is associated with a unique pair of real number $a = (x, y)$ measuring distance from the origin in the x and y directions. It is sometimes convenient to think of the point a as a vector.

- Scalar: Addition, subtraction, and multiplication, e.g.,
  $(x_1, y_1) - (x_2, y_2) = (x_1 - x_2, y_1 - y_2)$
- Norm: $||a|| = \sqrt{(x^2 + y^2)}$
- Distance: $|ab| = ||a - b||$
- Angle between vectors:
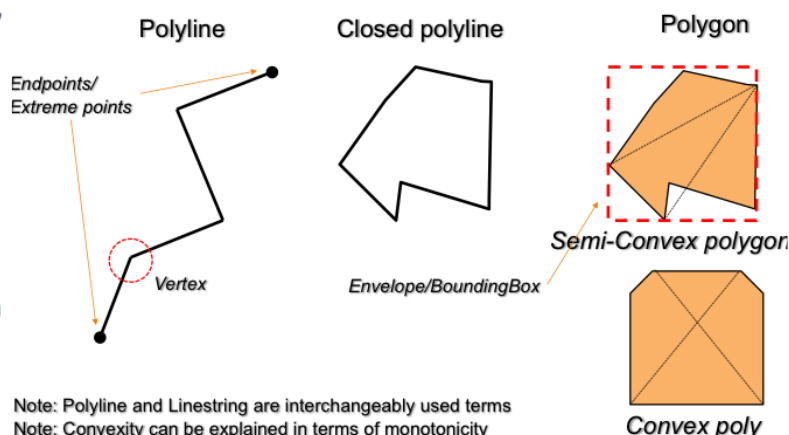 $\cos \alpha = \frac{x_a x_b + y_a y_b}{||a|| \cdot ||b||}$
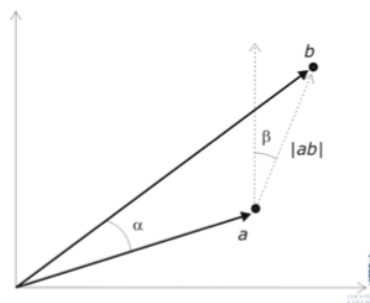
Line

- The *line* incident with a and b is defined as the point set $\{\lambda a + (1 - \lambda)b | \lambda \in \mathbb{R}\}$.
- The *line segment* between a and b is defined as the point set $\{\lambda a + (1 - \lambda)b | \lambda \in [0, 1]\}$.
- The *half line* radiating from b and passing through a is defined as the point set $\{\lambda a + (1 - \lambda)b | \lambda \geq 0\}$.
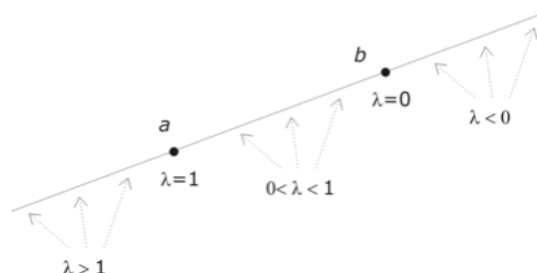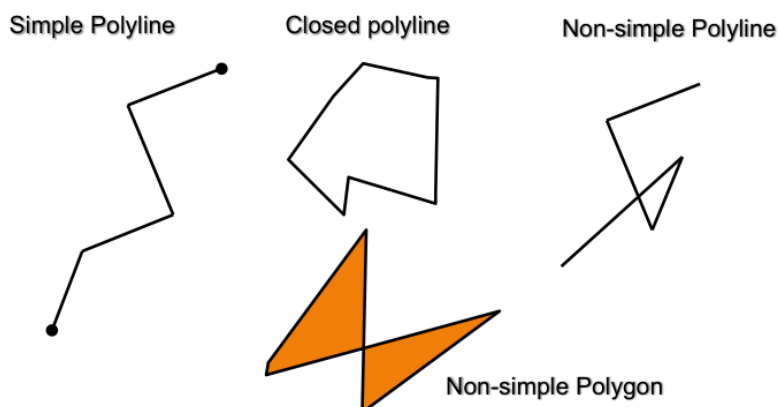
Polyline vs Polygon

- A *polyline* in $\mathbb{R}^2$ is a finite set of line segments (called *edges*) such that each edge end-point is shared by exactly two edges, except possibly for two points, called the extremes of the polyline.
- If no two edges intersect at any place other than possibly at their end-points, the polyline is simple.
- A polyline is closed if it has no extreme points.
- A (*simple*) *polygon* in $\mathbb{R}^2$ is the area enclosed by a simple closed polyline. This polyline forms the boundary of the polygon. Each end-point of an edge of the polyline is called a vertex of the polygon.
- A *convex* polygon has every point intervisible.
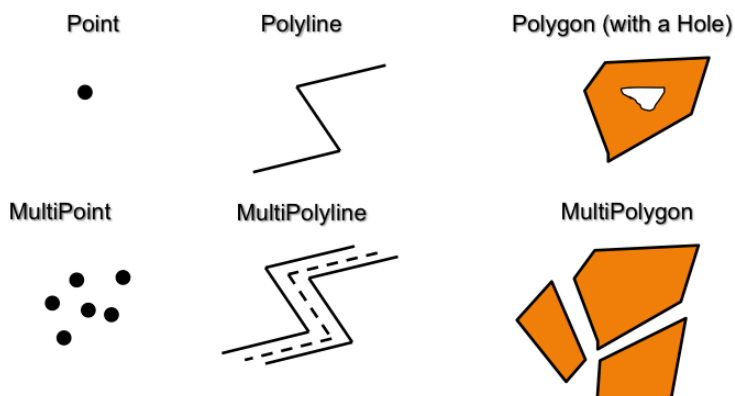- A *star-shaped* or semi-convex polygon has at least one point that is intervisible

Polyline vs Polygon (and other terms)

Polyline   Closed polyline   Polygon

Endpoints/ Extreme points

Vertex   Envelope/BoundingBox

Semi-Convex polygon

Convex poly

Note: Polyline and Linestring are interchangeably used terms
Note: Convexity can be explained in terms of monotonicity

Simple vs Non-Simple Geometry

Simple Polyline   Closed polyline   Non-simple Polyline

Non-simple Polygon

Single vs Multi-Geometry

Point   Polyline   Polygon (with a Hole)
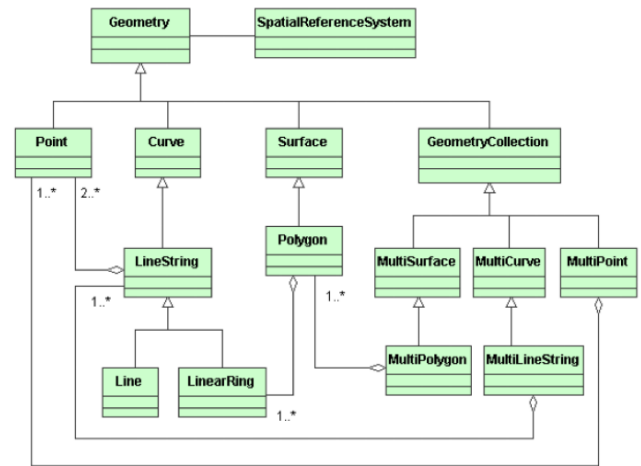
MultiPoint   MultiPolyline   MultiPolygon

Representing geometries as text
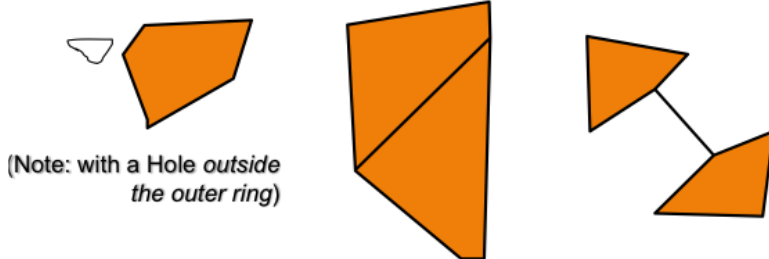
- □ **WKT: Well-Known <u>Text</u> representation:**
  - ▣ Point: "`POINT(70 30)`"
  - ▣ Polyline: "`LINESTRING(30 10,20 50,60 70)`"
  - ▣ Polygon: "`POLYGON((30 10,20 50,60 70, 30 10))`"
  - ▣ Polygon with hole:
    "`POLYGON((30 10,20 50,60 70, 30 10),(30 15, 25 45,55 65,30 15))`"
  - ▣ Multipolygon: "`MULTIPOLYGON (((40 40, 20 45, 45 30, 40 40)), ((20 35, 10 30, 10 10, 30 5, 45 20, 20 35), (30 20, 20 15, 20 25, 30 20)))`"

Invalid polygons

(Note: with a boundary *overlapping itself*)

(Note: with a Hole *outside the outer ring*)

Geometry Validation

- □ **Extremely important step when importing data from unknown sources;**
- □ **Invalid geometries can break spatial analysis!**
- □ **It can be a tedious process to fix data...**
- □ **ST_isSimple(geom)**
- □ **ST_isValid(geom) and ST_isValidReason(geom)**
- □ **ST_makeValid(geom) (and the expert trick: *ST_Buffer(geom,0.00001)***
- □ *Constraint:*
  ```
  ALTER TABLE public.my_valid_table
      ADD CONSTRAINT enforce_valid_geom CHECK (st_isvalid(geom));
  ```

Operations on Spatial entities

Query types

- □ **Thematic (as in non-spatial databases):**
  - ▣ Return all areas where landuse is "crop"
  - ▣ Return all parcels where owner is "Joe Blocks"

  Standard DB – already covered
  - ▣ → we get a subset of spatial entities that satisfy a non-spatial condition
- □ **Geometric (you have experienced some, and we will do some more soon):**
  - ▣ What is the distance from NY to MEL?
  - ▣ Return the area of 100 km around Melbourne.
  - ▣ What is the region where pine trees grow in this forest
  - ▣ Is Bendigo north of Melbourne?
- □ **Topological:**
  - ▣ Is Parkville adjacent to Carlton?
  - ▣ What are the suburbs surrounding the suburb where Melbourne uni is?
  - ▣ What is the tram route between Melbourne Uni and St Kilda Beach
- □ **Set-theoretic operations (as distinguished by some authors)**
  - ▣ Union/Intersection/Difference – create geometry based on topological test outomces
  - ▣ What is the part of the Yarra river that is within Melbourne

50