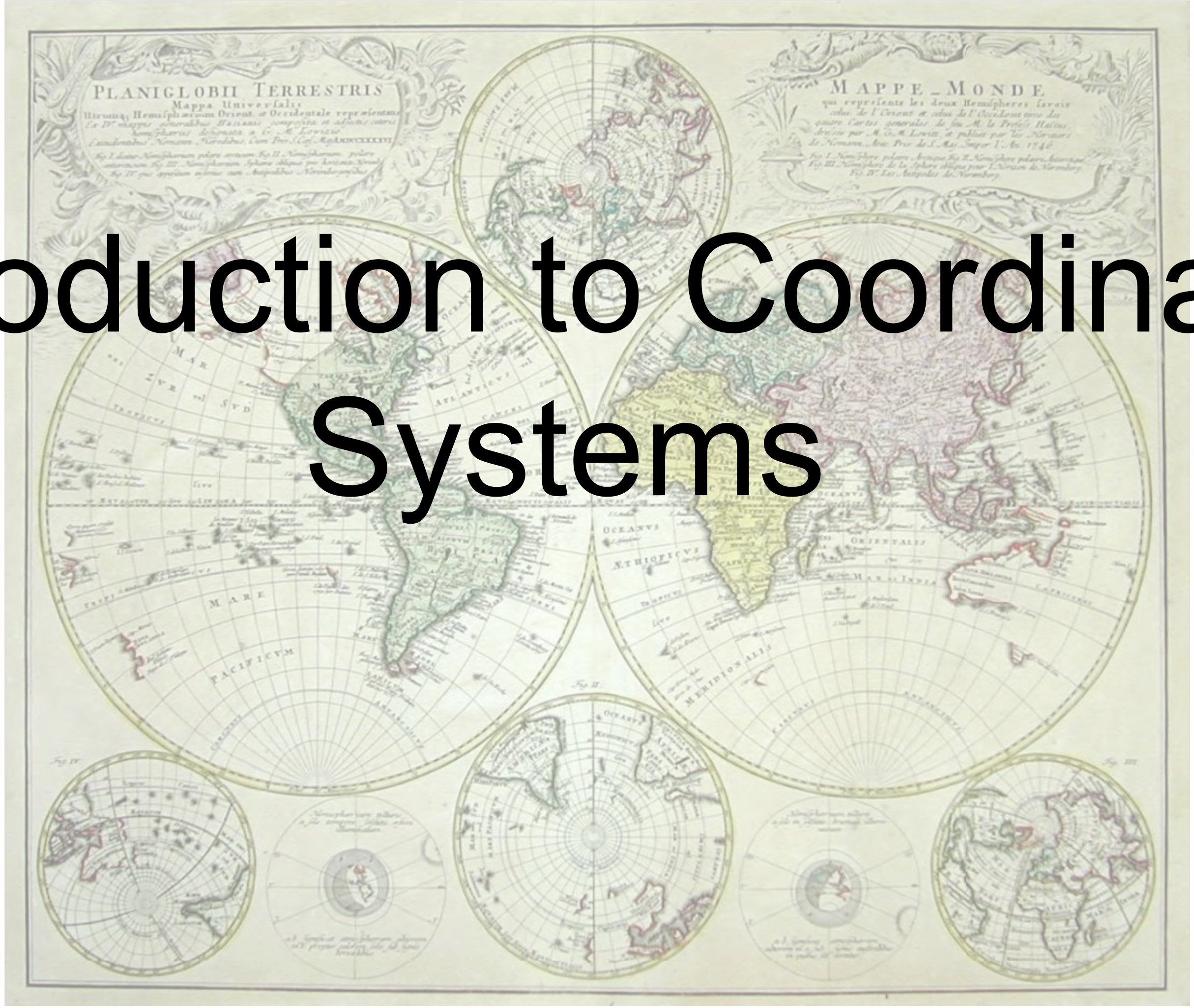
※ The Huntington

Introduction to Coordinate Systems



Introduction to Coordinate Systems

… what's doable in 20 minutes.

FOSS4G

Prizren, 2023

Javier Jimenez Shaw

PROJ contributor.

Civil Engineer and Software Developer.

Technical Coordinator of SRS team at Pix4D.

https://github.com/jjimenezshaw/

Content

- Why do we need CRS?
- Geographic Coordinate (Reference) Systems
- Projections
- UTM / LCC
- Projected Coordinate Reference Systems
- Examples: Europe, State Plane
- WKT
- EPSG
- PROJ

© discworld.com permission granted for this presentation

Why do we need Coordinate Reference Systems?

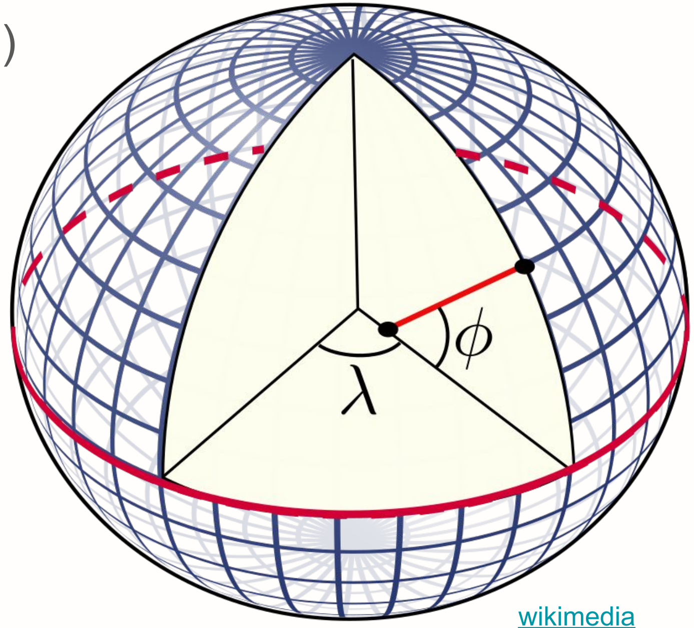
- We need to locate points on the Earth (2D, 3D, 4D?) respect to a known reference.
- Common understanding of the **reference(s)**.

- Coordinates without a CRS are meaningless numbers.
- Coordinates with the wrong CRS are very confusing.
- When getting the wrong coordinate reference system makes a lake go away.

Geographic Coordinate System

- Define the earth as Ellipsoid (of revolution) / Spheroid
- Coordinates as (geodetic) latitude (ϕ), longitude (λ)

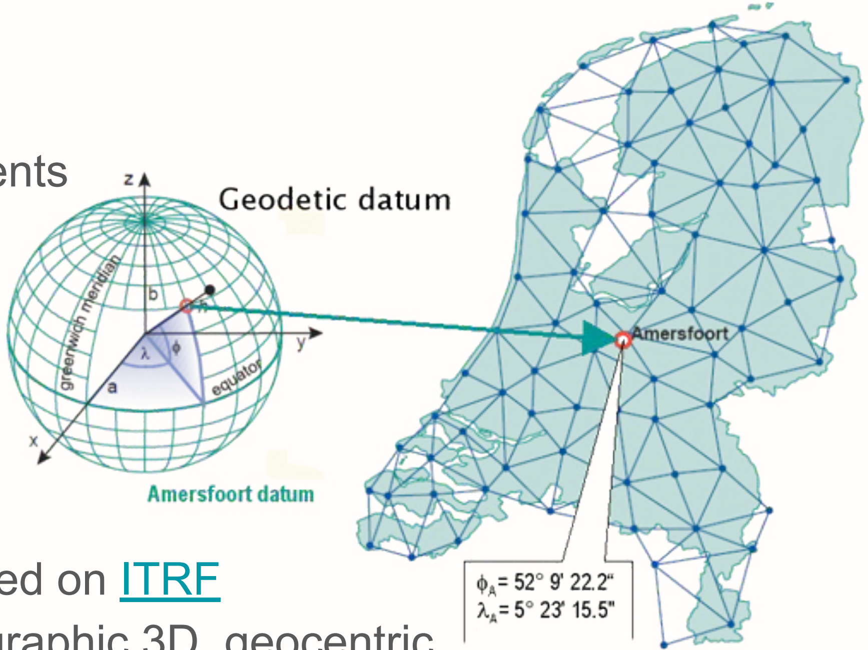
We need a well defined **Reference**!



wikimedia

Geographic Coordinate (Reference) System

- **Datum**
  - Spheroid with size defined (R, e)
  - Location based on station measurements
- **Prime meridian** (usually Greenwich)
- **Unit** (usually degree)
- Examples:
  - WGS84 (EPSG:4326) - World
  - NAD83(2011) (EPSG:6318) - USA
  - ETRS89 (EPSG:4258) - Europe
- New ones (like the new NATRF2022) are based on ITRF
- May have 3 flavours: Geographic 2D, geographic 3D, geocentric



credit un.org

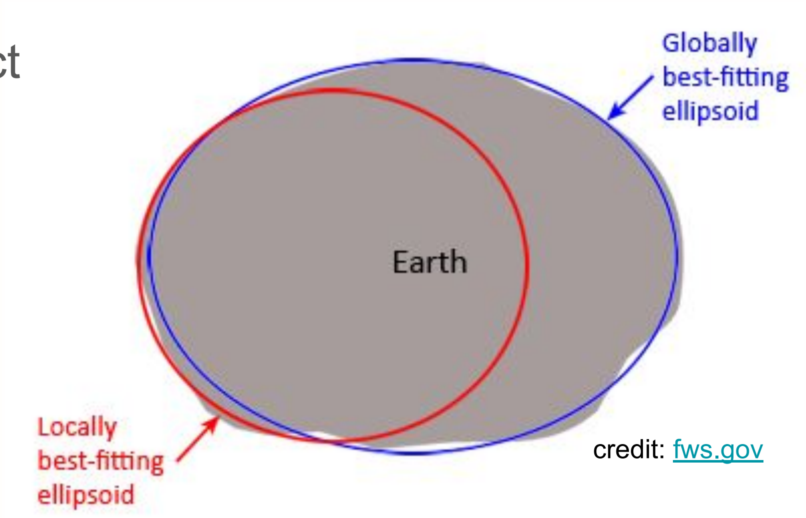
Local or global ellipsoid

- Fitting an ellipsoid to the Real World™ is not easy
- A datum that works in one location may not work somewhere else in the globe
- Datum transformations are not always exact

➢ CH1903+ (Swiss CRS):
  - Uses Bessel 1841 Ellipsoid
    - a = 6,377,397.155 m
    - b = 6,356,078.963 m
  - Axes about 700 m shorter
  - Offset of approx {674, 15, 405} m from WGS84

➢ Amersfoort (Dutch CRS)
  - Bessel Ellipsoid
  - Offset: {565, 50, 465} m and some rotation

➢ DHDN (German CRS)



credit: fws.gov

Projections

It is impossible to show a spherical surface on a flat plane without distortion.

© Charles Preppernau

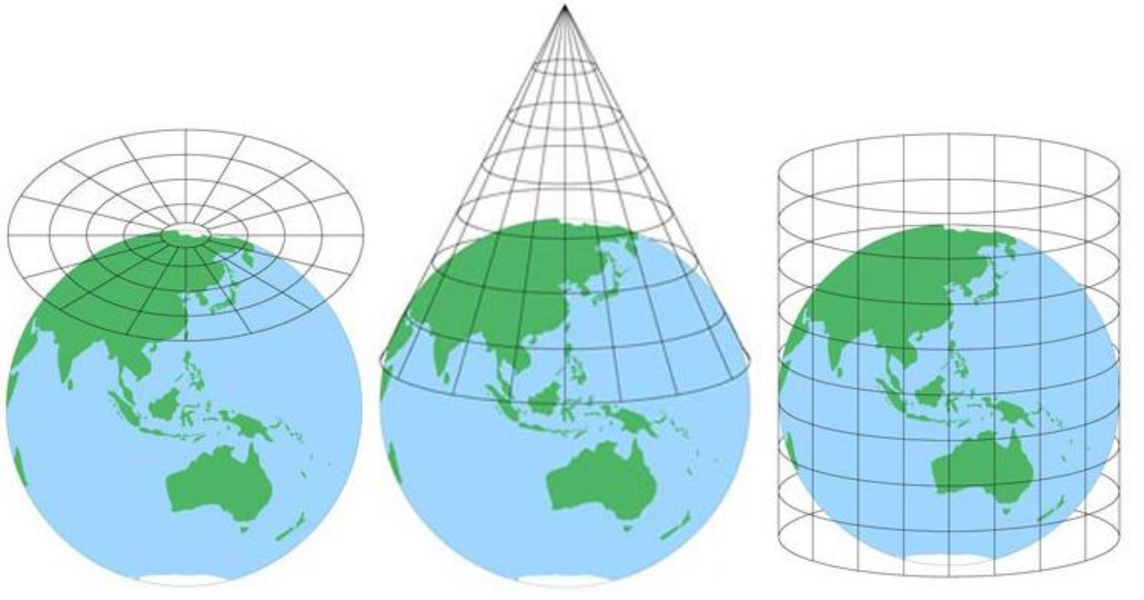
Projections

- Transform from 3D model (using only 2 coordinates) to 2D **flat** surface:
  We can't please everyone.
- Planar
- **Conical**
- **Cylindrical**
- ...other

It may conserve

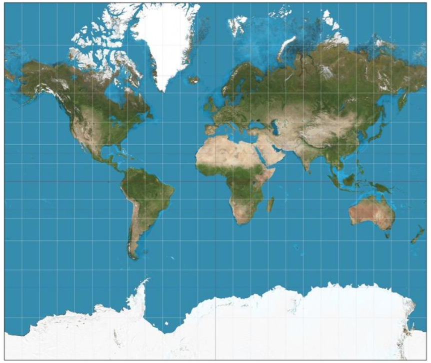
- Areas
- Local angles (conformal)
- Distances (from one/two points)

[List of projections]



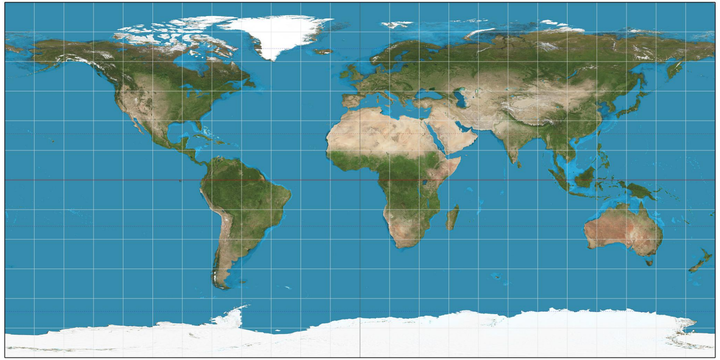
atlasandboots.com

Projections

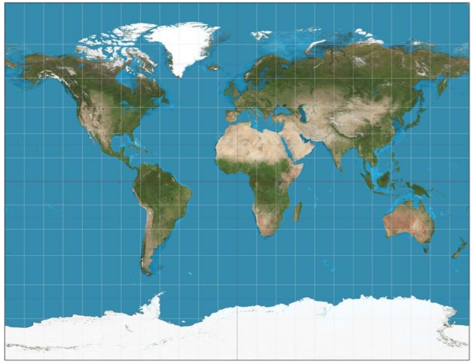


Mercator (1569)

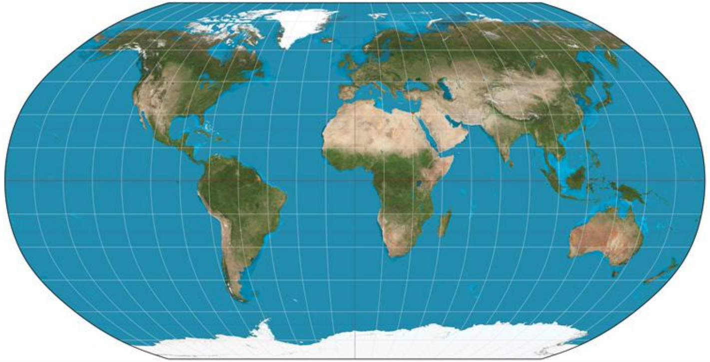
Plate Carrée (AD 100)



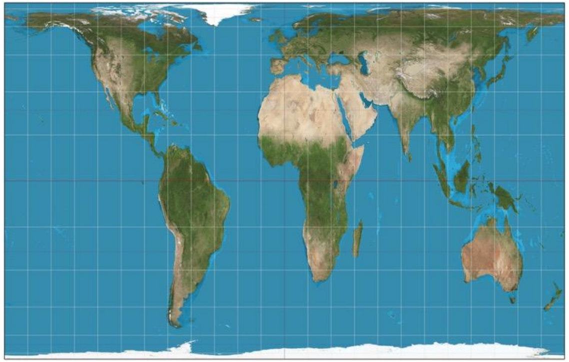
Gall Stereographic (1855)



Lambert conformal conic (1772)

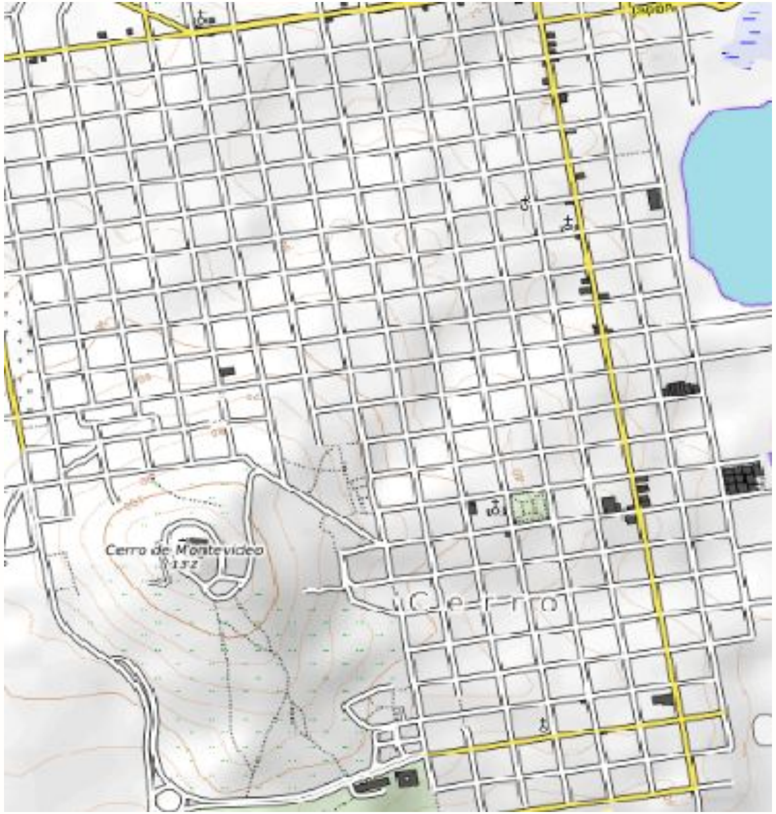


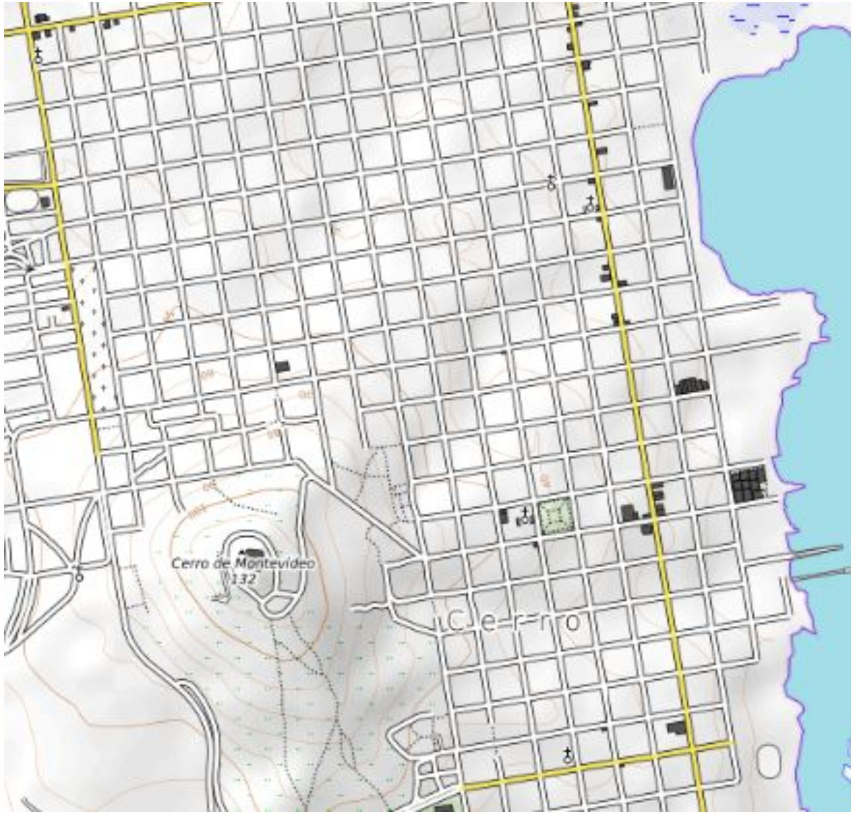
Robinson (1963)



Gall-Peters (1855)

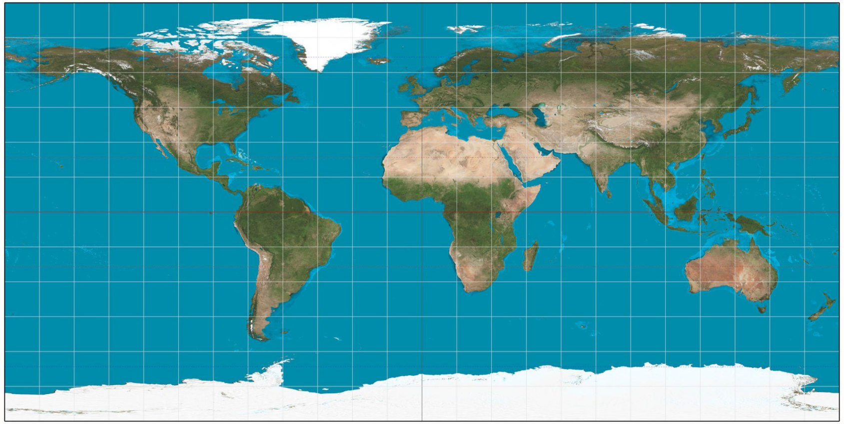
Conformal Projections: `Plate Carrée` vs `Mercator`





Equirectangular projection

- Plate Carrée is a particular case.
- Neither equal area nor conformal.



© Strebe

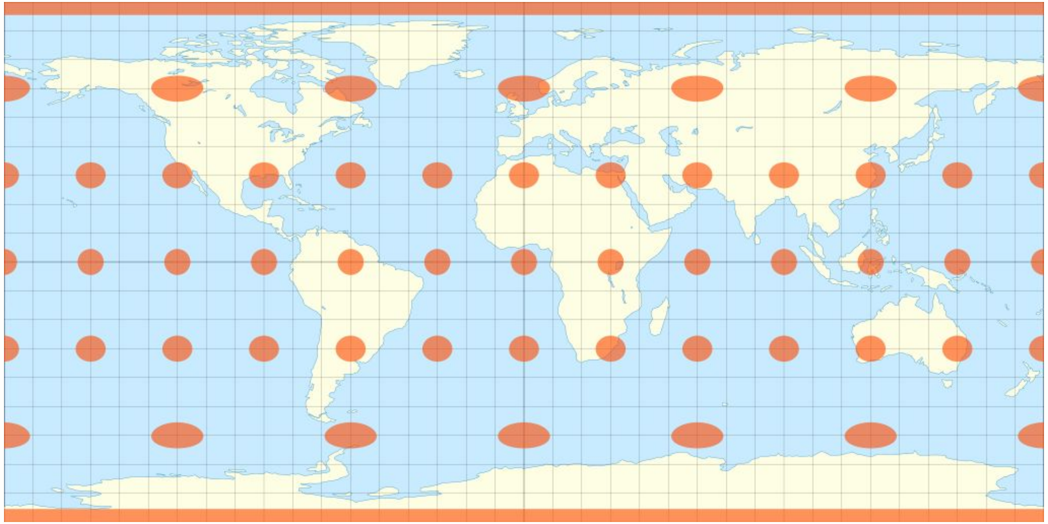
- Formulas (sphere)
  - $x = R(\lambda - \lambda_0)\cos\varphi_1$
  - $y = R(\varphi - \varphi_0)$
  - $\lambda = \lambda_0 + x / (R\cos\varphi_1)$
  - $\varphi = \varphi_0 + y / R$

- Simplified formulas ($\lambda_0=0$, $\varphi_0=0$, $\varphi_1=0$, $R=1$)
  - $x = \lambda$
  - $y = \varphi$

  - QGIS uses this to display geographic systems

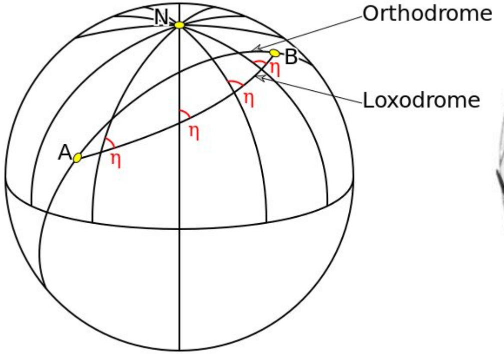
Tissot’s indicatrix

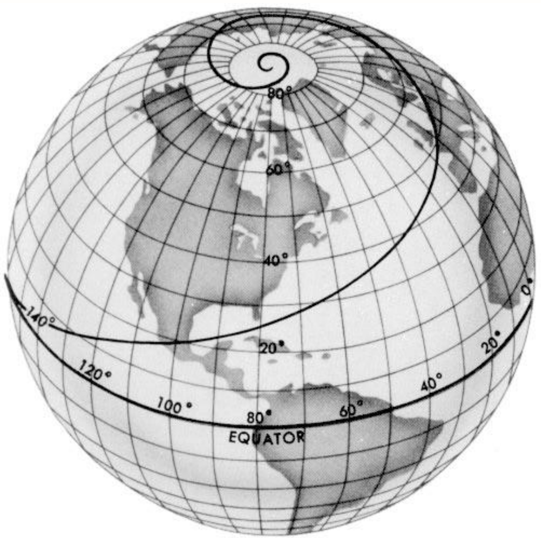
Tissot’s indicatrix. All those ellipses represent the same circular shape and area in the Real World™

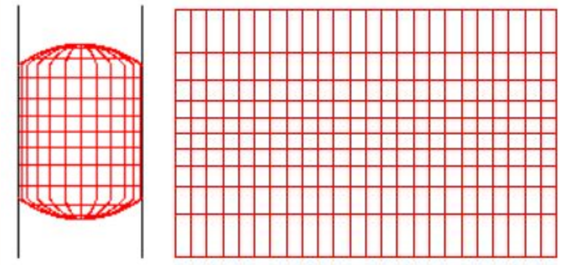


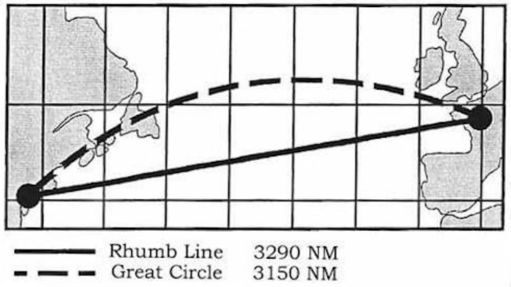
Mercator projection

- Done in 1569 by Gerardus Mercator.
- Loxodromic (rhumbline) curves are straight lines.
- Great for navigation!
- Poles are located at infinity.
- Conformal (keep angles).
- North is “up” for every point.
- https://xkcd.com/2082/



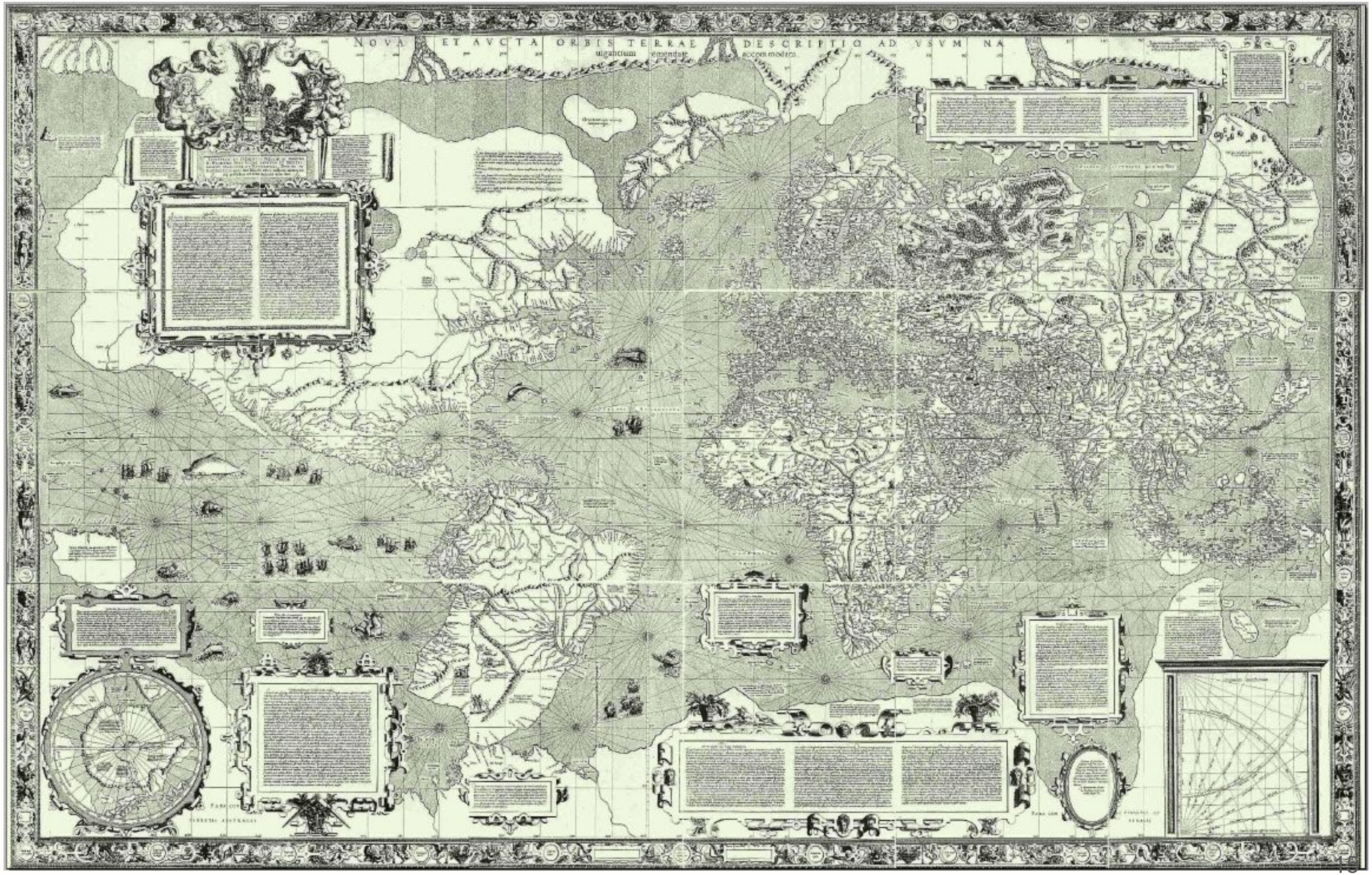






Robert Israel

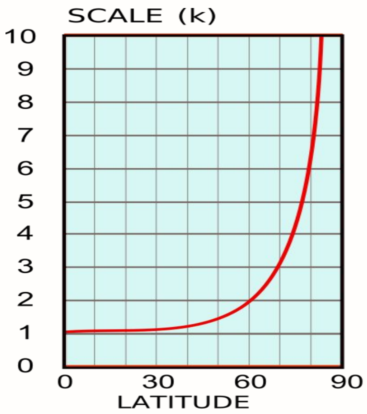
Mercator’s map from 1569



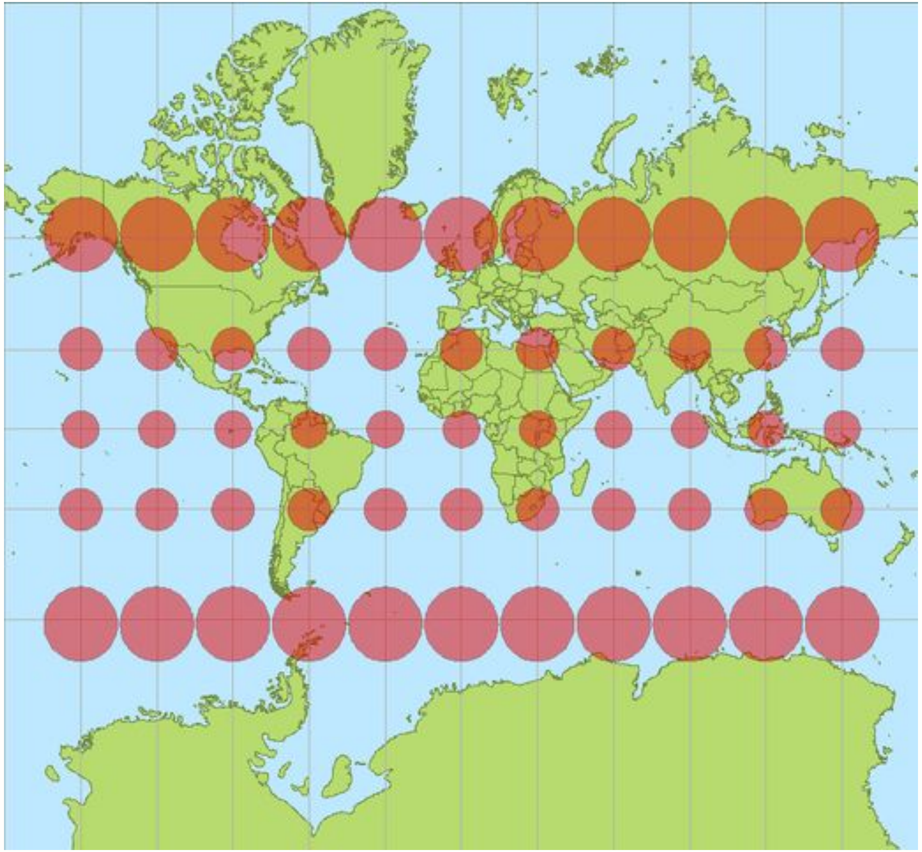
Mercator projection

Tissot's indicatrix.

k = 1/cos(lat)



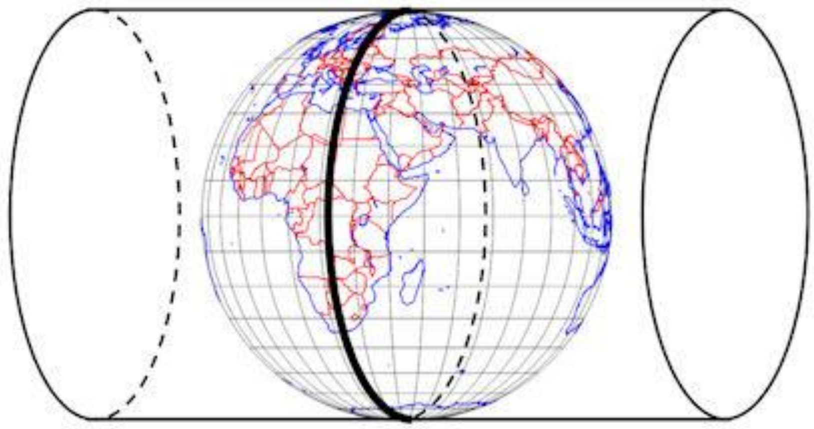
Play TheTrueSize

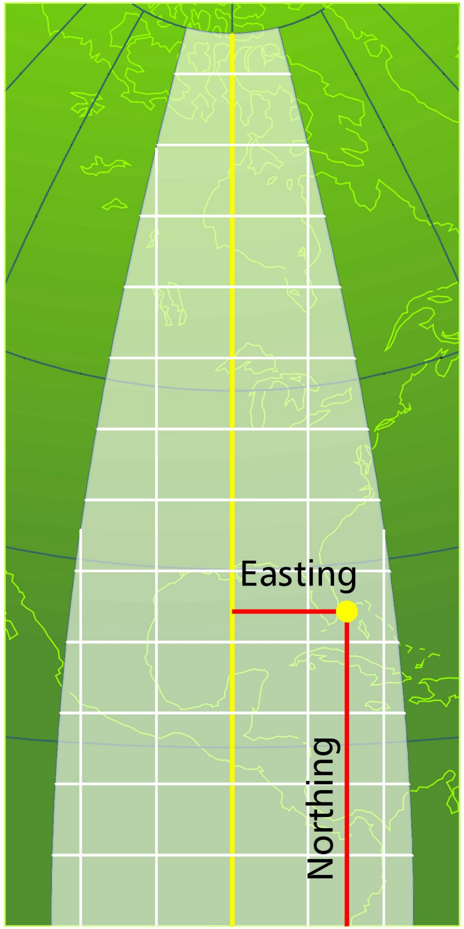


© Stefan Kühn

Transverse Mercator Projection

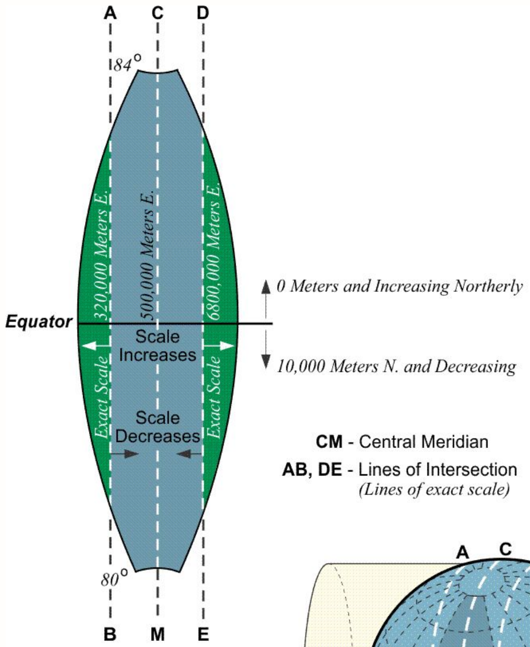
- Like Mercator, but tangent at a meridian, not at the equator.
- Accurate near tangent line.
- Y axis is north only at central meridian or equator.

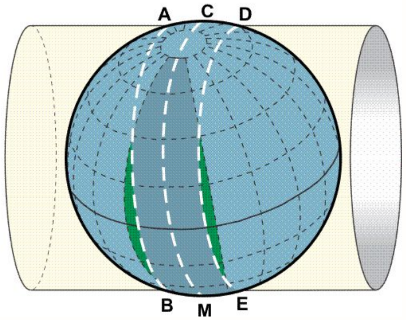


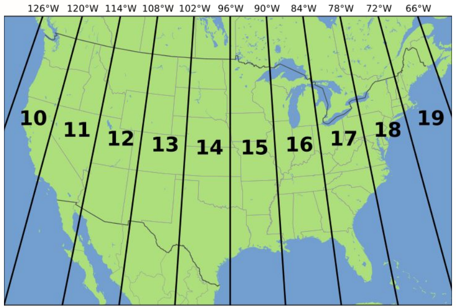


UTM / Universal Transverse Mercator

- Transverse Mercator with 60 zones around the globe.
- 6° wide each zone.
- Not only in WGS84, but many other datums.
- To avoid negative values uses false northing and easting.
- Scale factor of 0.9996
- Valid between 84°N and 80°S.
- UPS in the poles.

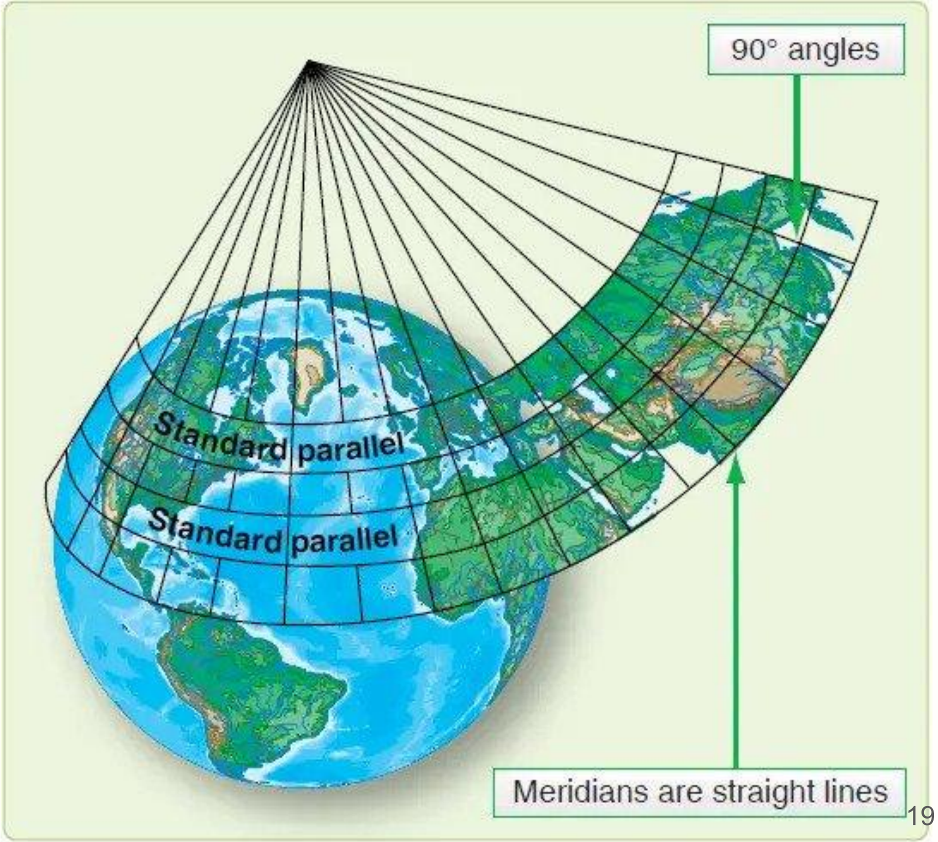






LCC / Lambert Conformal Conic

- One or two standard parallels
- Good for E-W maps
- Conformal: keep angles
- Small distance distortion
- Used in many State Plane CS
- Developed by Johann Lambert
- Used in aviation: a straight line approximates a great circle



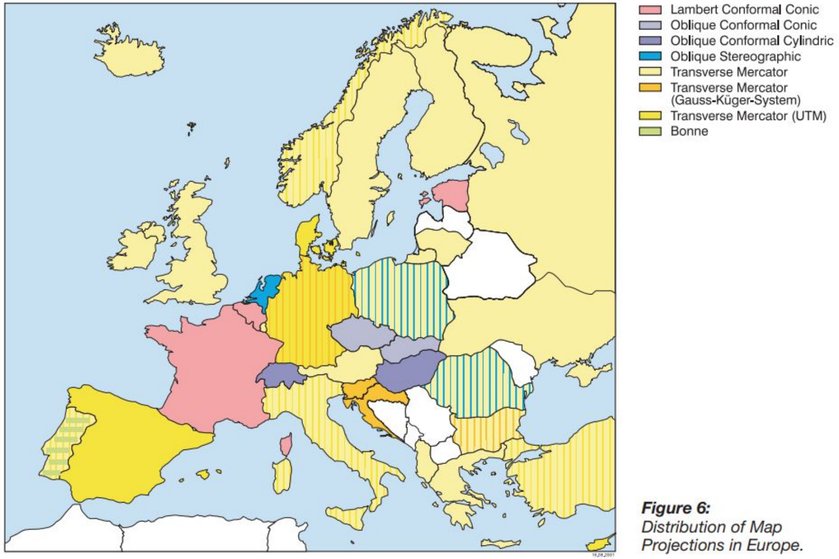
flightliteracy.com

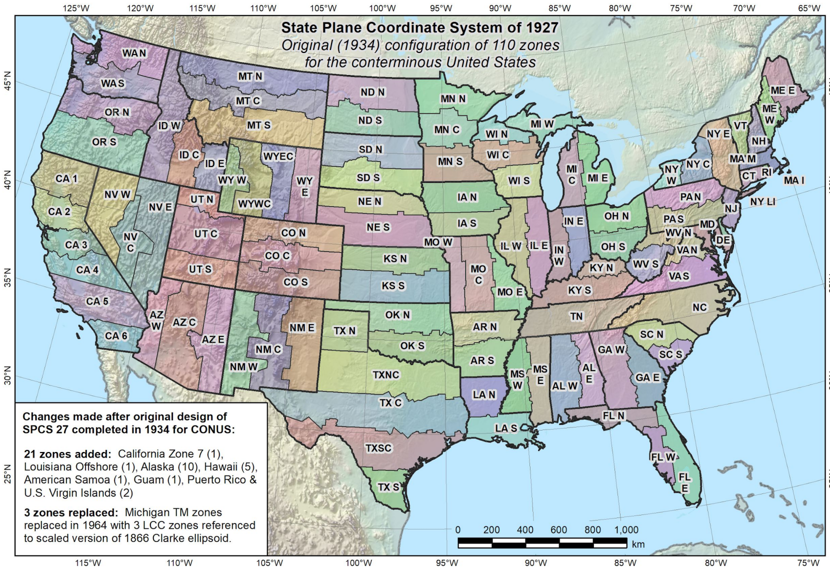
Projected Coordinate Reference System

- Applies a projection with specific parameters **on a Geographic CRS**
  - v.g. Geographic ETRS89 + UTM projection with central meridian -123, false northing 0
- Different Geographic CRS produce different Projected CRS
- Length Units (m, ft, ftUS)
- Named as “{GeogCRS} / {Projection or Zone or Whatever} - VertCRS”
  - ETRS89 / UTM zone 30N
  - NAD83 / California zone 3 (ftUS)
  - KOSOVAREF01 / Balkans zone 7

freepik.com

Examples - Europe, USA





Projections used in Europe (source)

State planes ngs.noaa.gov

EPSG

“European Petroleum Survey Group” https://epsg.org/

*The IOGP's EPSG **Geodetic Parameter Dataset** is a collection of definitions of coordinate reference systems and coordinate transformations. Maintained by the Geodesy Subcommittee of the IOGP Geomatics Committee.*

Database is updated regularly, even with new CRSs.
PROJ has 7300 PCRS, 5000 from EPSG; 1700 GCRS, 1000 from EPSG

- *EPSG:4326* WGS 84
- *EPSG:4258* ETRS89
- *EPSG:4269* NAD83
- *EPSG:6317* NAD83(2011)
- *EPSG:9140* KOSOVAREF01

- *EPSG:32632* WGS 84 / UTM zone 32N
- *EPSG:25830* ETRS89 / UTM zone 30N
- *EPSG:26910* NAD83 / UTM zone 10N
- *EPSG:6420* NAD83(2011) / California zone 3 (ftUS)
- *EPSG:9141* KOSOVAREF01 / Balkans zone 7

WKT / Well Known Text 📖

- Text to describe the CRS (there are WKT for other entities, like geometries)
- Versions WKT, WKT2, and some minor variants.
- Different software supports different versions.

- Axes chaos
  - Latitude - longitude vs X - Y vs Y - X.
  - Easting-Northing, Northing-Easting, Westing-Southing…
  - Some formats define the axes order (E-N), regardless anything else.

WKT / Examples - GEOGCS 🌐

```
GEOGCS["ETRS89",
    DATUM["European_Terrestrial_Reference_System_1989"
        SPHEROID["GRS 1980",6378137,298.257222101,
            AUTHORITY["EPSG","7019"]],
        AUTHORITY["EPSG","6258"]],
    PRIMEM["Greenwich",0,
        AUTHORITY["EPSG","8901"]],
    UNIT["degree",0.0174532925199433,
        AUTHORITY["EPSG","9122"]],
    AUTHORITY["EPSG","4258"]]
```

WKT / Examples - PROJCS

```
PROJCS["WGS 84 / UTM zone 32N",
    GEOGCS["WGS 84",
        DATUM["WGS_1984",
          SPHEROID["WGS 84",6378137,298.257223563,
                AUTHORITY["EPSG","7030"]],
            AUTHORITY["EPSG","6326"]],
        PRIMEM["Greenwich",0,
            AUTHORITY["EPSG","8901"]],
        UNIT["degree",0.0174532925199433,
            AUTHORITY["EPSG","9122"]],
        AUTHORITY["EPSG","4326"]],
    PROJECTION["Transverse_Mercator"],
    PARAMETER["latitude_of_origin",0],
    PARAMETER["central_meridian",9],
    PARAMETER["scale_factor",0.9996],
    PARAMETER["false_easting",500000],
    PARAMETER["false_northing",0],
    UNIT["metre",1,
        AUTHORITY["EPSG","9001"]],
    AXIS["Easting",EAST],
    AXIS["Northing",NORTH],
    AUTHORITY["EPSG","32632"]]
```

```
PROJCS["NAD83 / California zone 3 (ftUS)",
    GEOGCS["NAD83",
        DATUM["North_American_Datum_1983",
            SPHEROID["GRS 1980",6378137,298.257222101,
                AUTHORITY["EPSG","7019"]],
            AUTHORITY["EPSG","6269"]],
        PRIMEM["Greenwich",0,
            AUTHORITY["EPSG","8901"]],
        UNIT["degree",0.0174532925199433,
            AUTHORITY["EPSG","9122"]],
        AUTHORITY["EPSG","4269"]],
    PROJECTION["Lambert_Conformal_Conic_2SP"],
    PARAMETER["latitude_of_origin",36.5],
    PARAMETER["central_meridian",-120.5],
    PARAMETER["standard_parallel_1",38.43333333333333],
    PARAMETER["standard_parallel_2",37.06666666666667],
    PARAMETER["false_easting",6561666.667],
    PARAMETER["false_northing",1640416.667],
    UNIT["US survey foot",0.3048006096012192,
        AUTHORITY["EPSG","9003"]],
    AXIS["Easting",EAST],
    AXIS["Northing",NORTH],
    AUTHORITY["EPSG","2227"]]
```

WKT / Examples - WKT2 😎

```
PROJCRS["NAD83 / California zone 3 (ftUS) ",
    BASEGEOGCRS["NAD83",
        DATUM["North American Datum 1983",
            ELLIPSOID["GRS 1980",6378137,298.257222101,
                LENGTHUNIT["metre",1]]],
        PRIMEM["Greenwich",0,
            ANGLEUNIT["degree",0.0174532925199433]],
        ID["EPSG",4269]],
    CONVERSION["SPCS83 California zone 3 (US Survey feet)",
        METHOD["Lambert Conic Conformal (2SP)",
            ID["EPSG",9802]],
        PARAMETER["Latitude of false origin",36.5,
            ANGLEUNIT["degree",0.0174532925199433],
            ID["EPSG",8821]],
        PARAMETER["Longitude of false origin",-120.5,
            ANGLEUNIT["degree",0.0174532925199433],
            ID["EPSG",8822]],
        PARAMETER["Latitude of 1st standard
parallel",38.4333333333333,
            ANGLEUNIT["degree",0.0174532925199433],
            ID["EPSG",8823]],
        PARAMETER["Latitude of 2nd standard
parallel",37.0666666666667,
            ANGLEUNIT["degree",0.0174532925199433],
            ID["EPSG",8824]],
        PARAMETER["Easting at false origin",6561666.667,
            LENGTHUNIT["US survey foot",0.304800609601219],
            ID["EPSG",8826]],
        PARAMETER["Northing at false origin",1640416.667,
            LENGTHUNIT["US survey foot",0.304800609601219],
            ID["EPSG",8827]]],
    ...
```

```
PROJCRS["NAD83 / California zone 3 (ftUS)",
    ...
    CS[Cartesian,2],
        AXIS["easting (X)",east,
            ORDER[1],
            LENGTHUNIT["US survey foot",0.304800609601219]],
        AXIS["northing (Y)",north,
            ORDER[2],
            LENGTHUNIT["US survey foot",0.304800609601219]],
    USAGE[
        SCOPE["unknown"],
        AREA["USA - California - SPCS - 3"],
        BBOX[36.73,-123.02,38.71,-117.83]],
    ID["EPSG",2227]]
```

PROJ

https://proj.org/

https://github.com/OSGeo/PROJ

“PROJ is a generic coordinate transformation software that transforms geospatial coordinates from one coordinate reference system (CRS) to another. This includes cartographic projections as well as geodetic transformations.”

https://crs-explorer.proj.org/

PROJ

- C/C++ library, used by other software like GDAL (used by QGIS)
- Python bindings with pyproj
- There was a big change in PROJ 6 respect to PROJ4. Update!

- Information about CRS (CLI: projinfo)
- Transformations and projections: (CLI: cs2cs, projinfo, cct, proj)
- Transformations using grid files (PROJ-data) 600 MB and growing
- Helpful mailing list

Thanks for watching!

Javier Jimenez Shaw

https://github.com/jjimenezshaw/