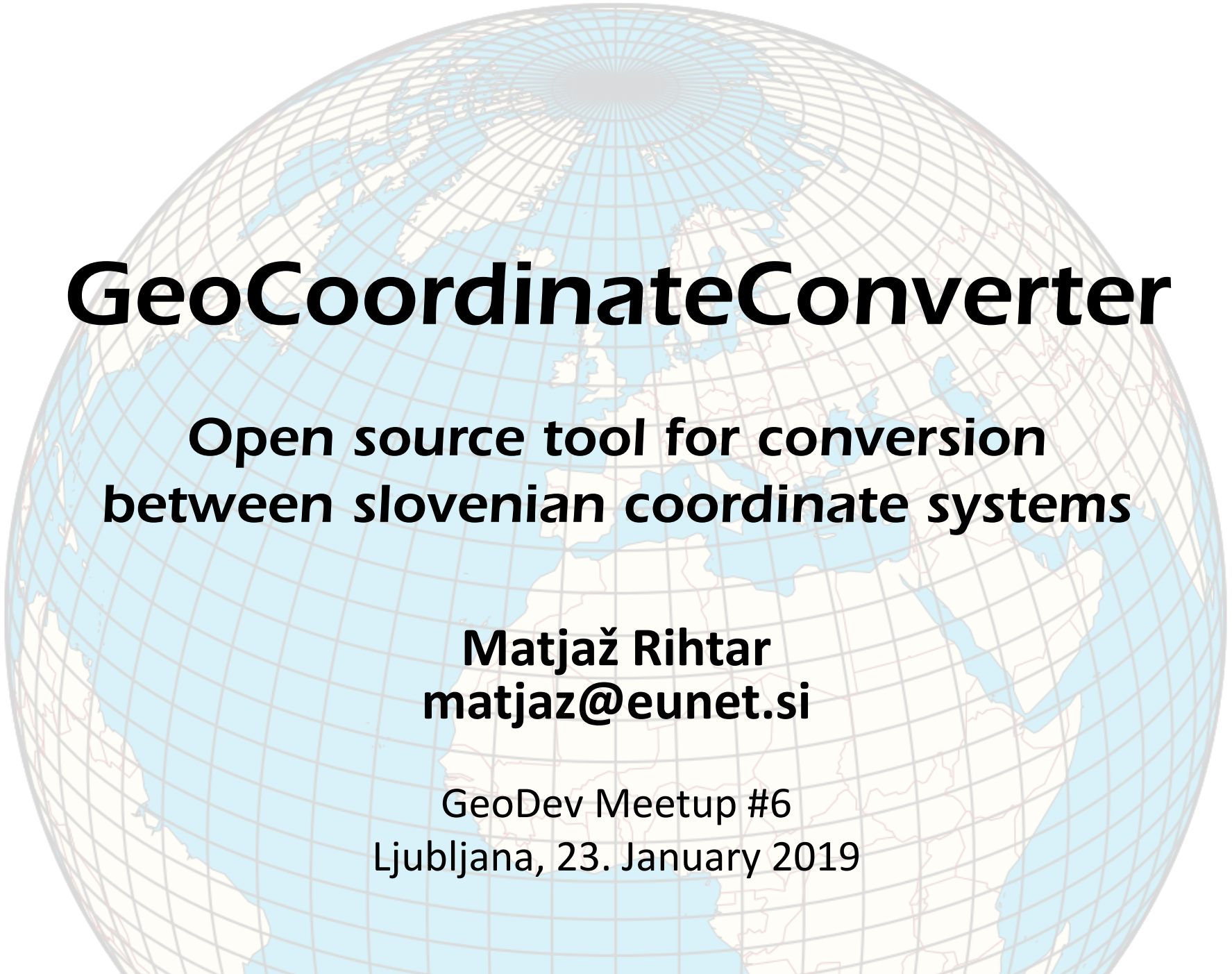


GeoCoordinateConverter

Open source tool for conversion between slovenian coordinate systems

Matjaž Rihtar
matjaz@eunet.si

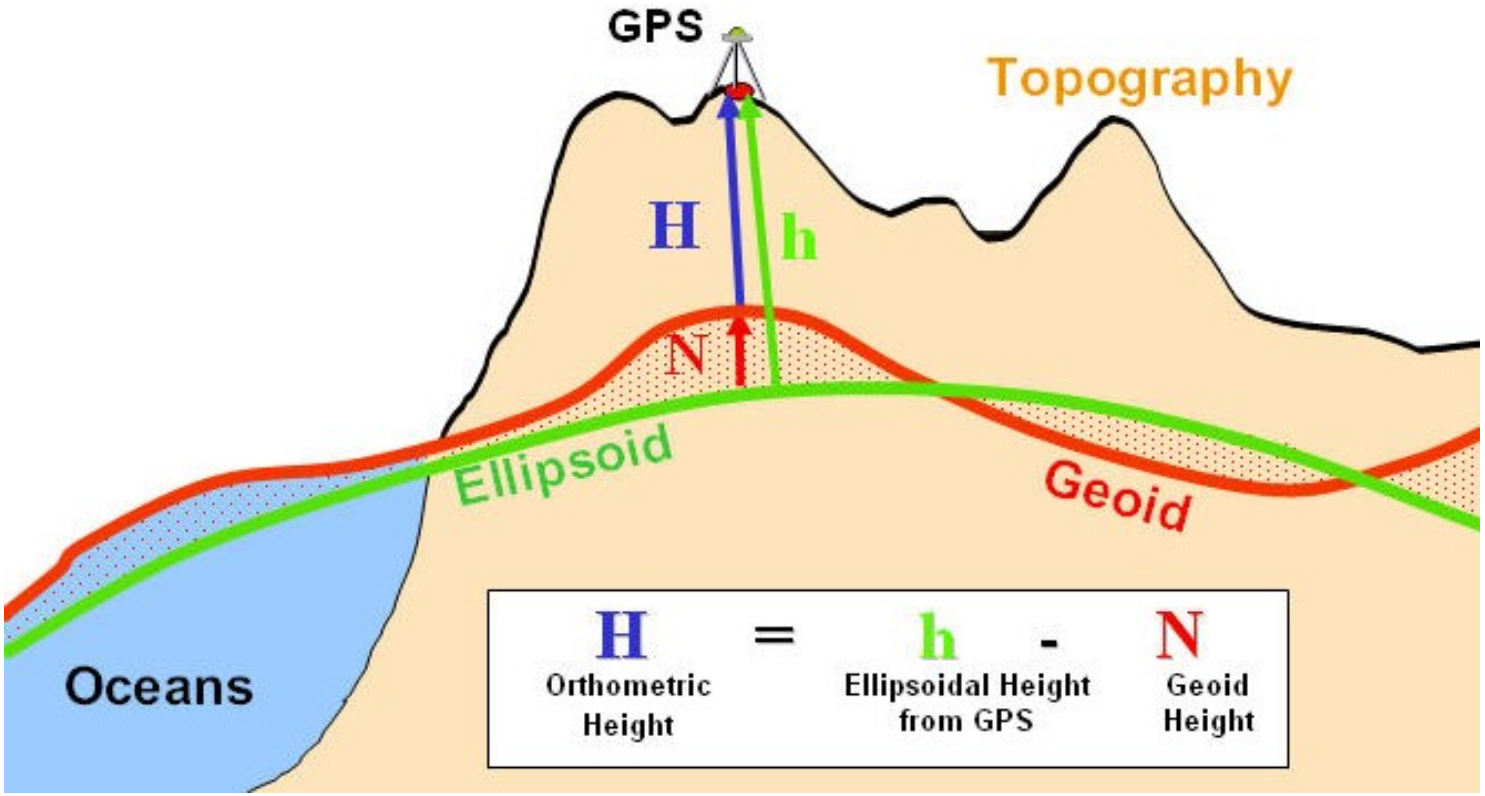
GeoDev Meetup #6
Ljubljana, 23. January 2019

Outline

- Height calculation
- Coordinate systems in Slovenia
- Conversion
- GeoCoordinateConverter tool

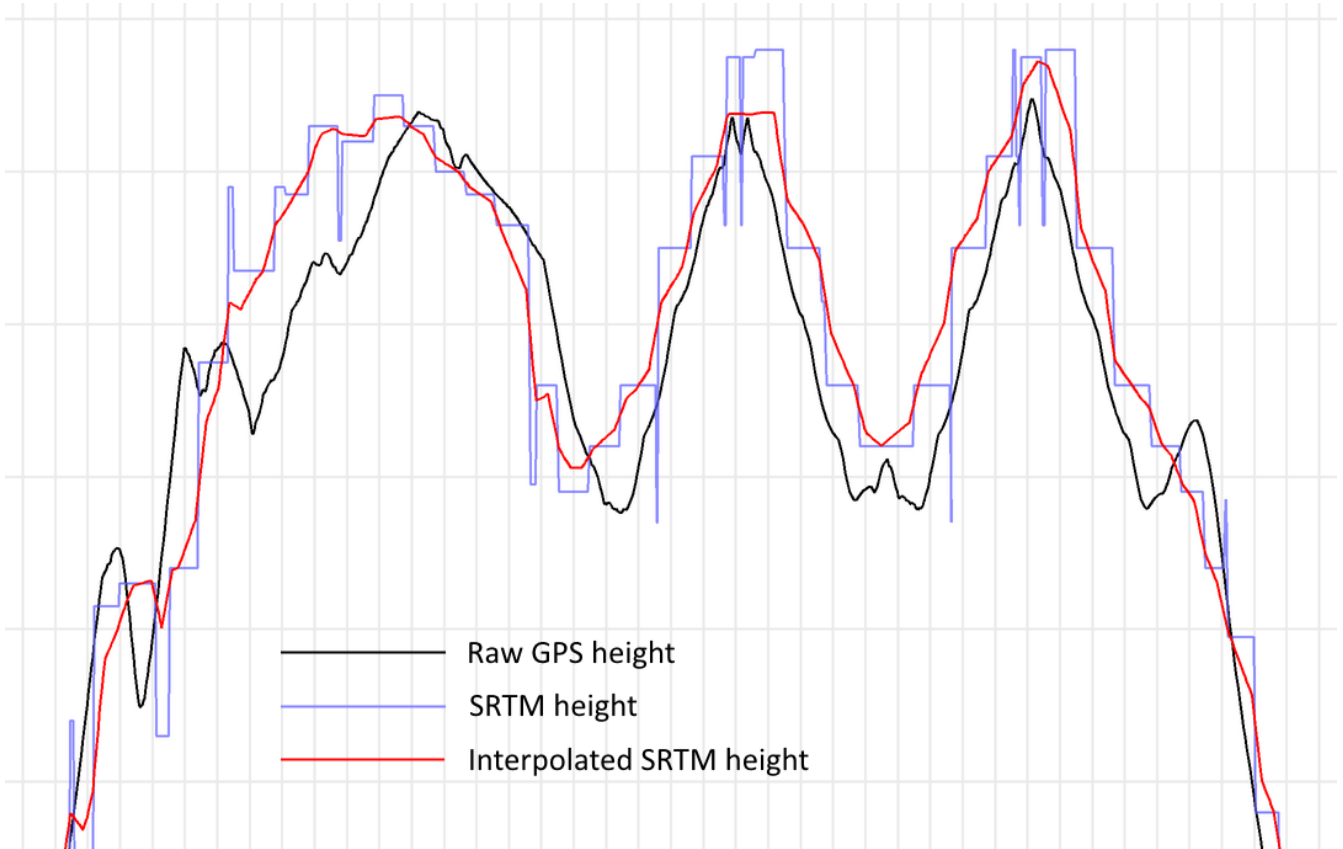
Heights

- Elipsoid vs. orthometric vs. geoid height



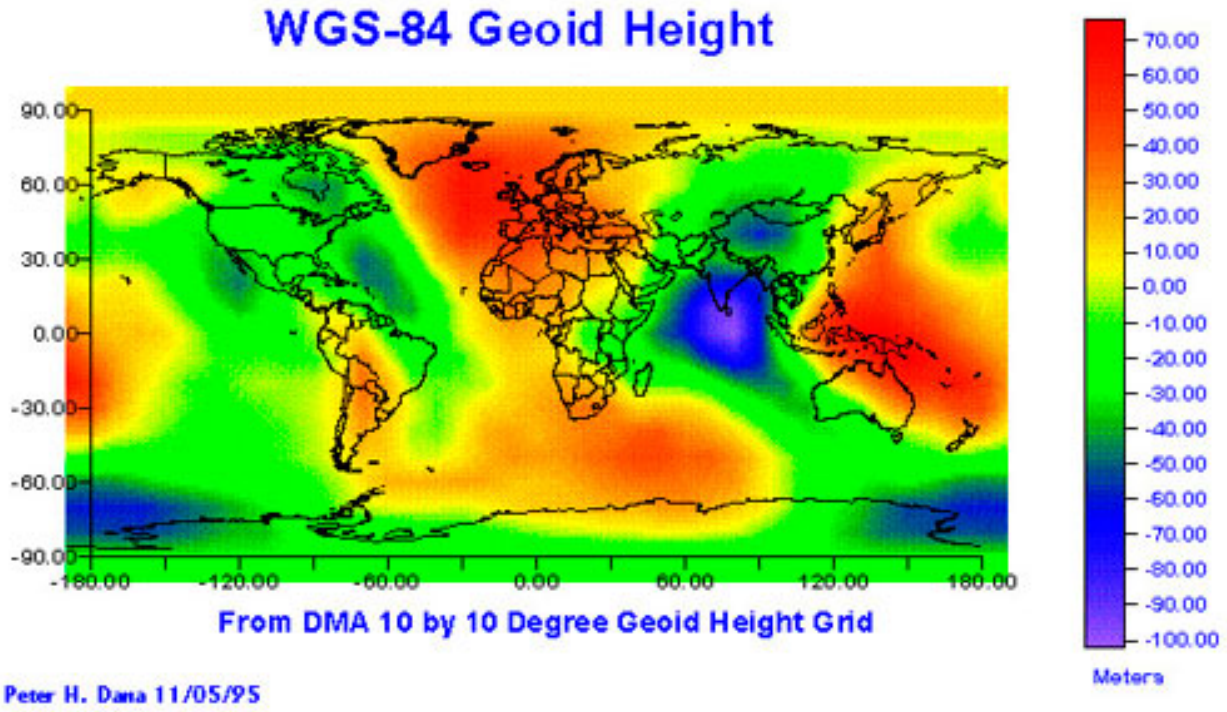
GPS height correction

- With geoid models
- With DEM models



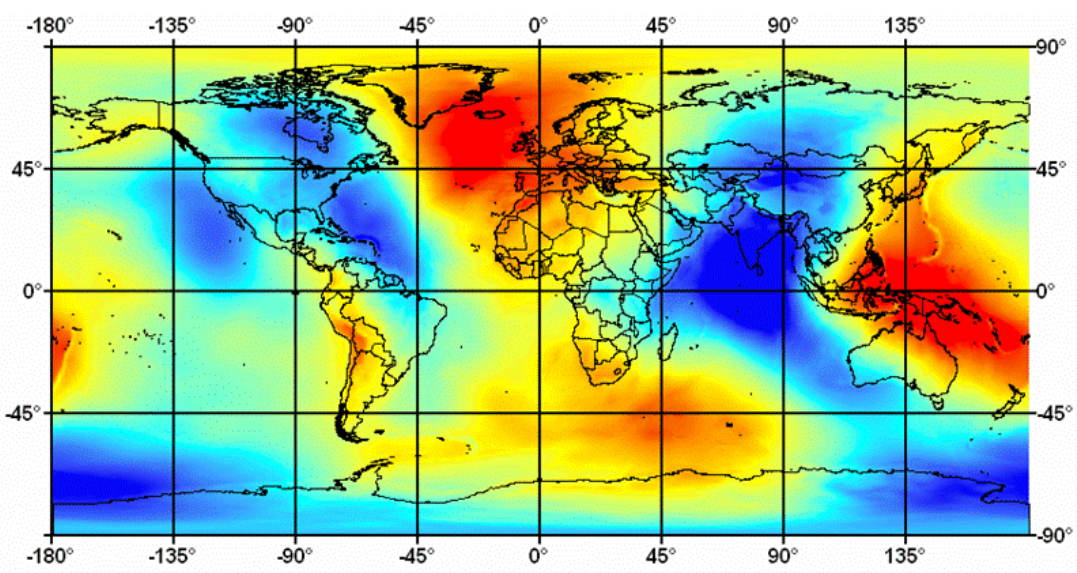
World geoid model

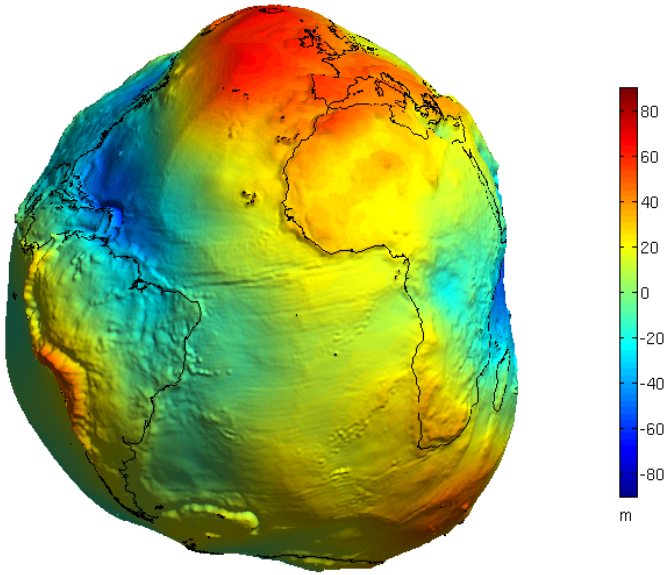
- WGS-84 (1996, revised in 2004)



World geoid model

- EGM2008

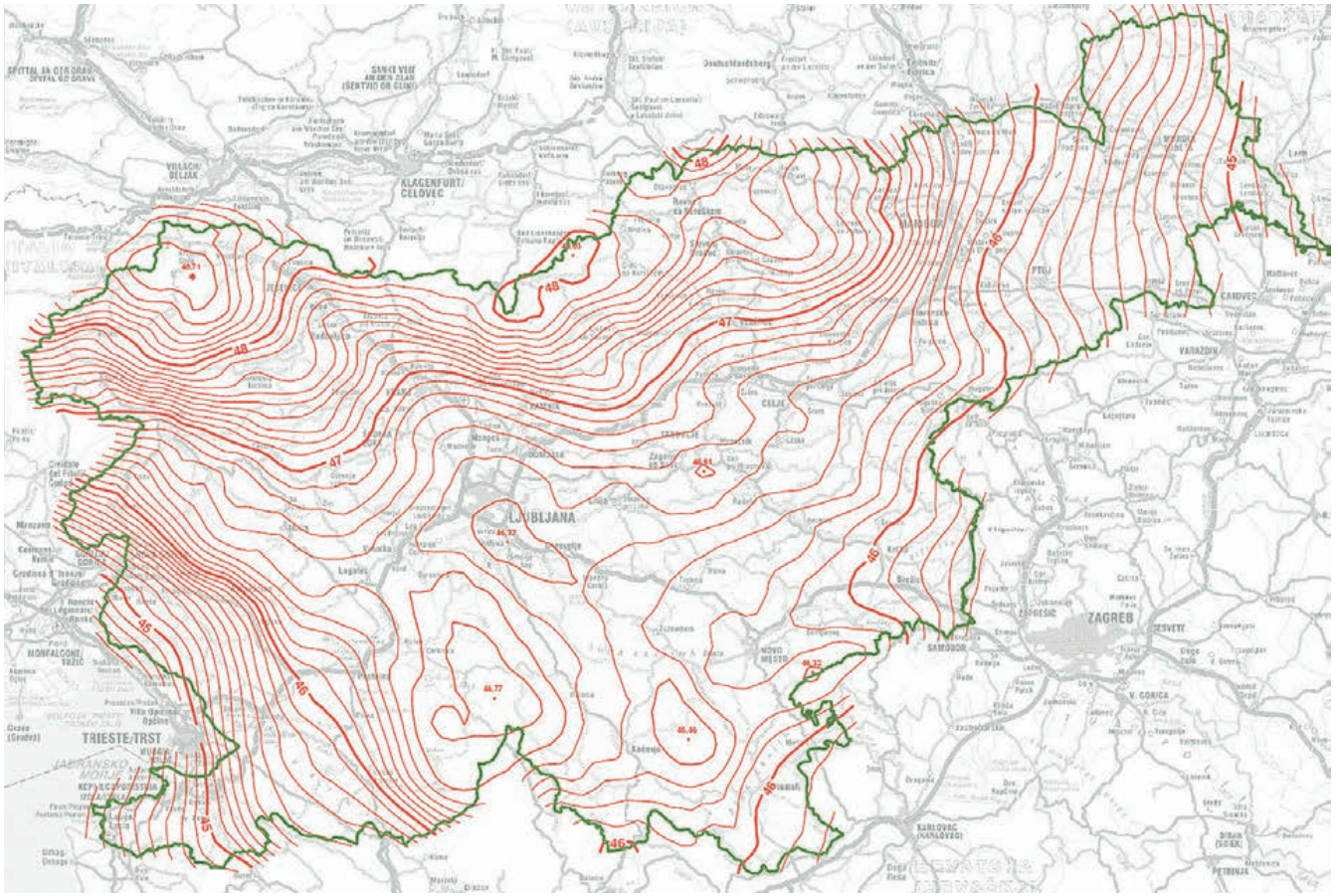




Geoid height (EGM2008, nmax=500)

Slovenian geoid model

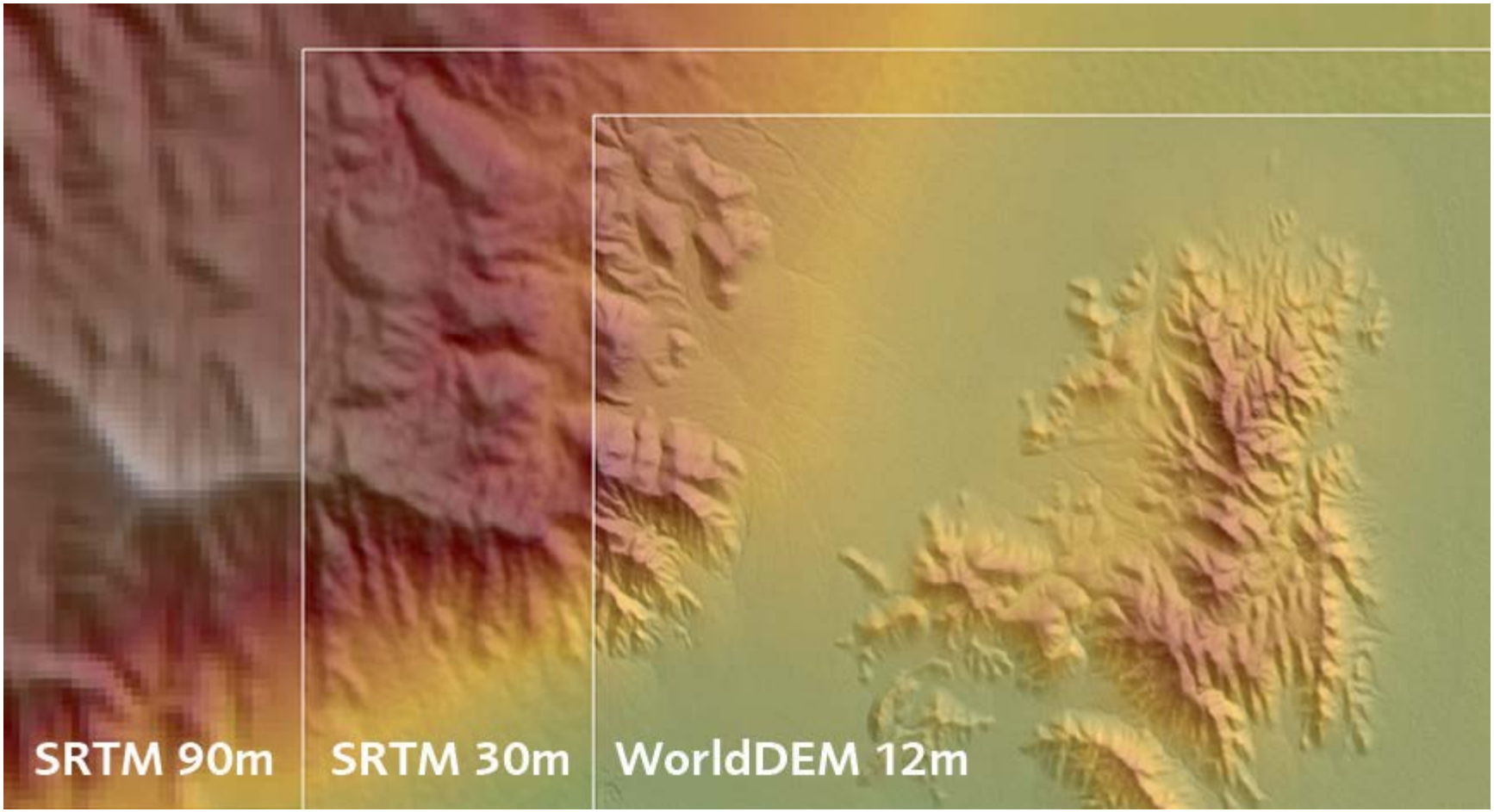
- From 2000, new model begin evaluated



Digital Elevation Models

- Space Shuttle Radar Topography Mission (SRTM)
  - originally 90m (except USA), now 30m, shadows
- ASTER Global Digital Elevation Model (GDEM2)
  - 90m, USA 30m, not so good
- JAXA's Global ALOS 3D World
  - 30m, best quality
- Astrium WorldDEM (Airbus, payable)
  - 12m
- Light Detection and Ranging (LiDAR) models
  - 1m

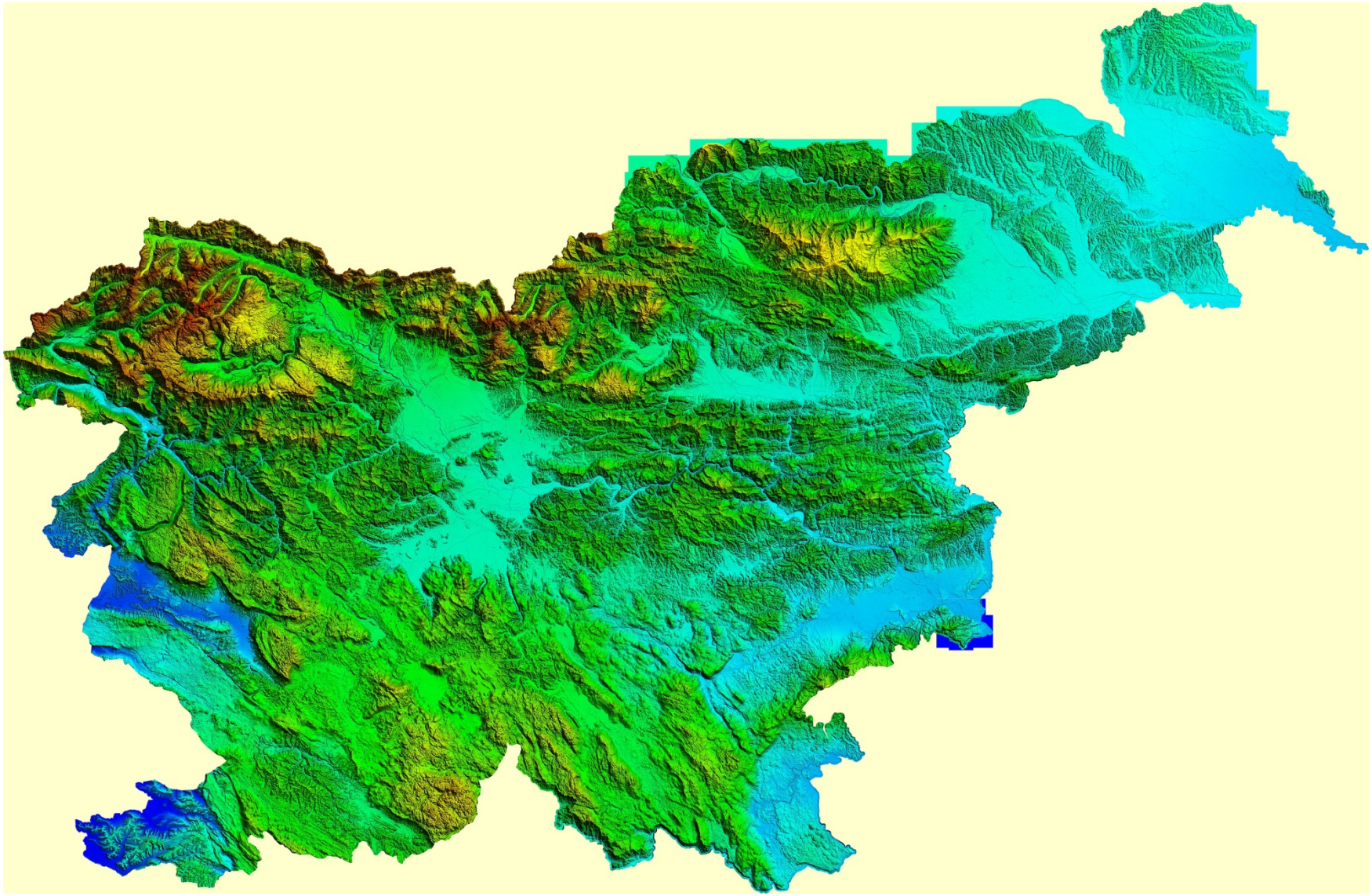
Digital Elevation Models



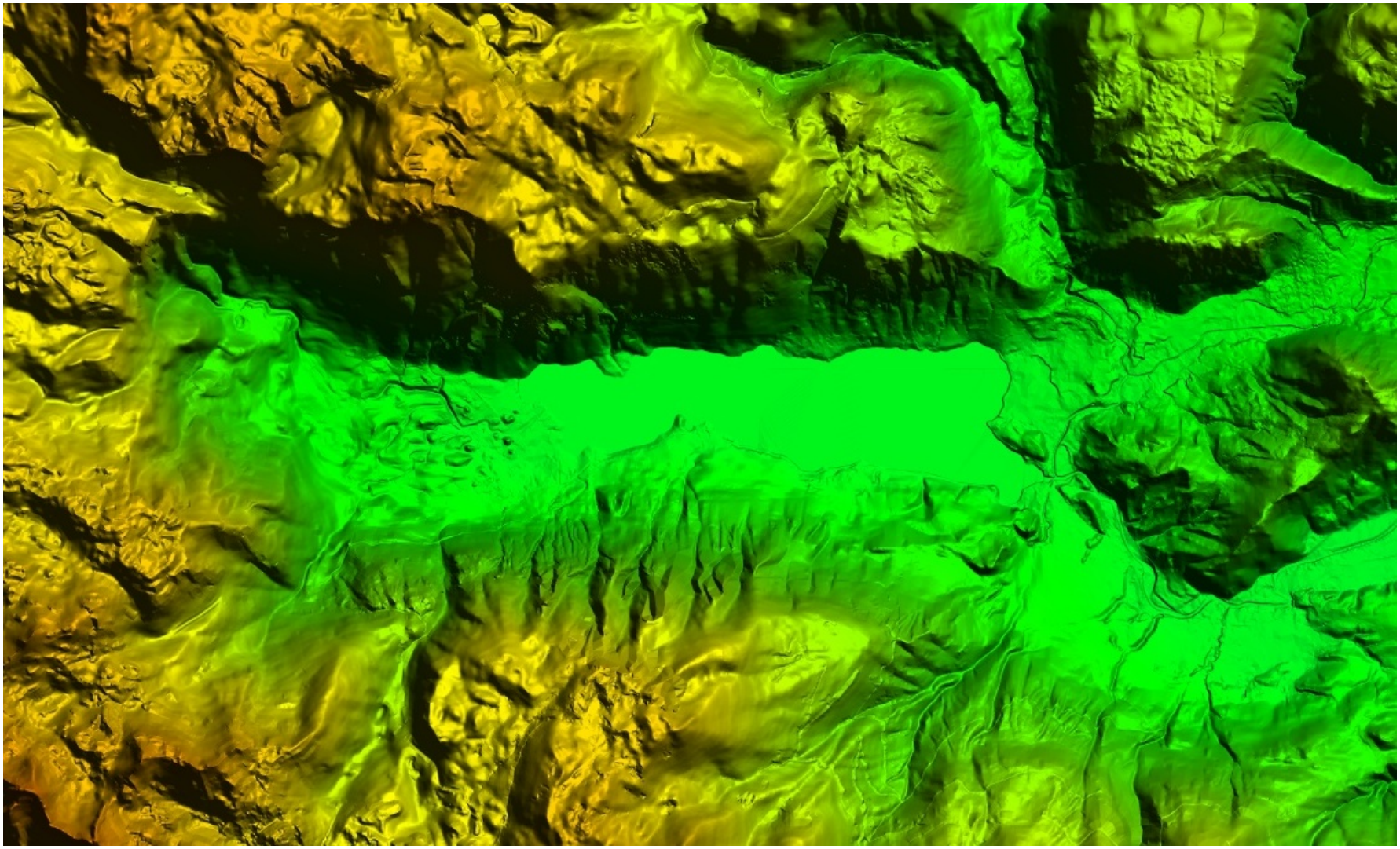
Slovenian DMV models

- Older:
  - DMV 5m (2011)
  - DMV 12.5m, DMV 25m and DMV 100m
  - In Gauss-Krüger (D48) coordinate system
- LiDAR (2015)
  - 1m
  - In Gauss-Krüger (D48) and Transverse Mercator (D96)

Slovenian DMV 12.5m



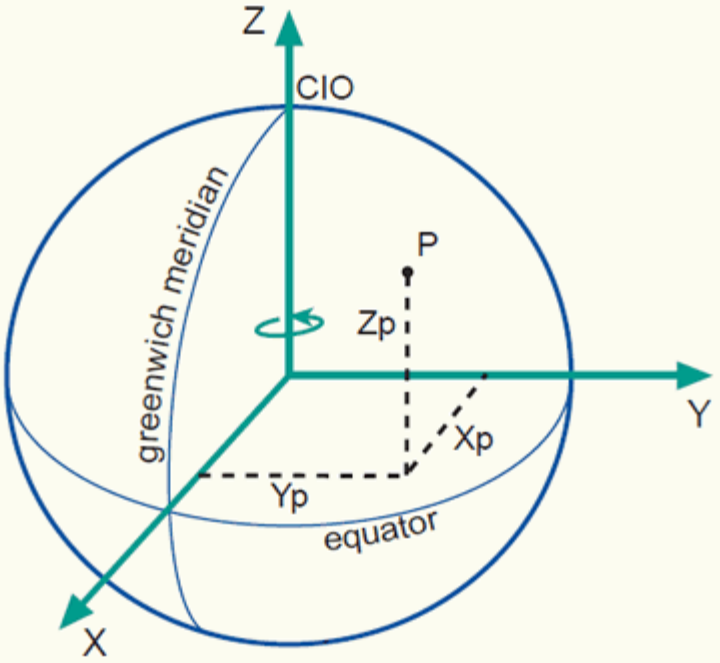
Slovenian DMV 12.5m



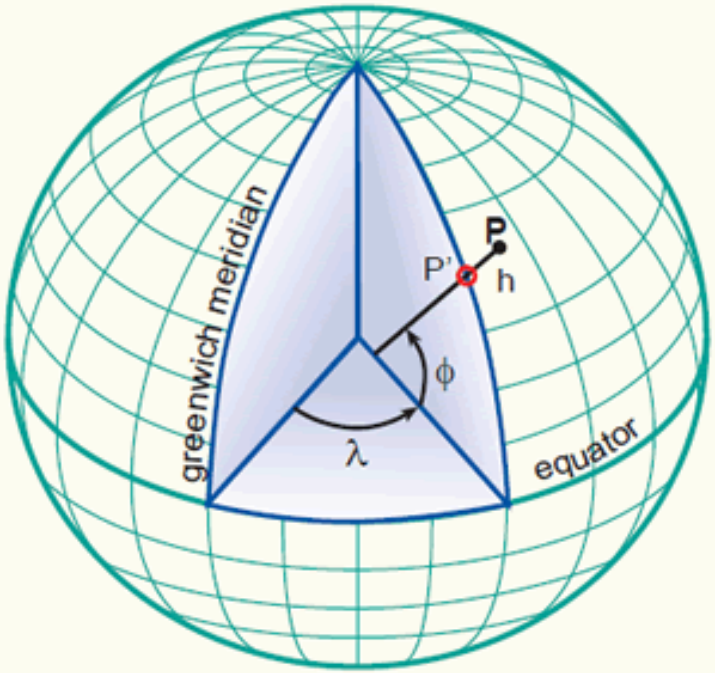
Coord. systems in Slovenia

- Old: D48/GK
  - Geodetic datum 1948
  - Bessel's elipsoid 1841
  - Gauss-Krüger projection
  - EPSG:3787, EPSG:3912
- New: D96/TM
  - Geodetic datum 1996 (slovenian realization of ETRS89)
  - Elipsoid GRS80 (≈ WGS84)
  - Transverse Mercator projection
  - EPSG:3794

Coordinates



3D geocentric coordinates



3D geographic coordinates

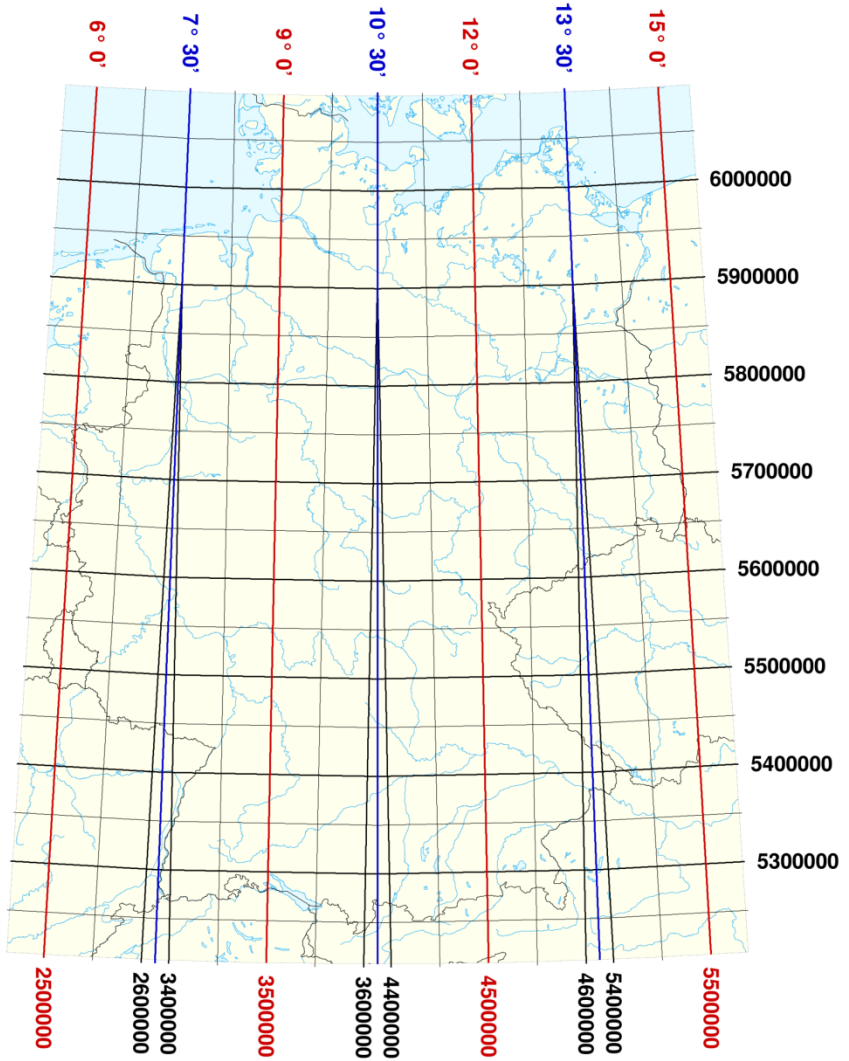
Conversion method

- G-K (y,x,H) ↔ Bessel (φ,λ,h) ↔ Bessel (X,Y,Z)
  ↕*
  UTM (y,x,H) ↔ WGS84(φ,λ,h) ↔ WGS84 (X,Y,Z)

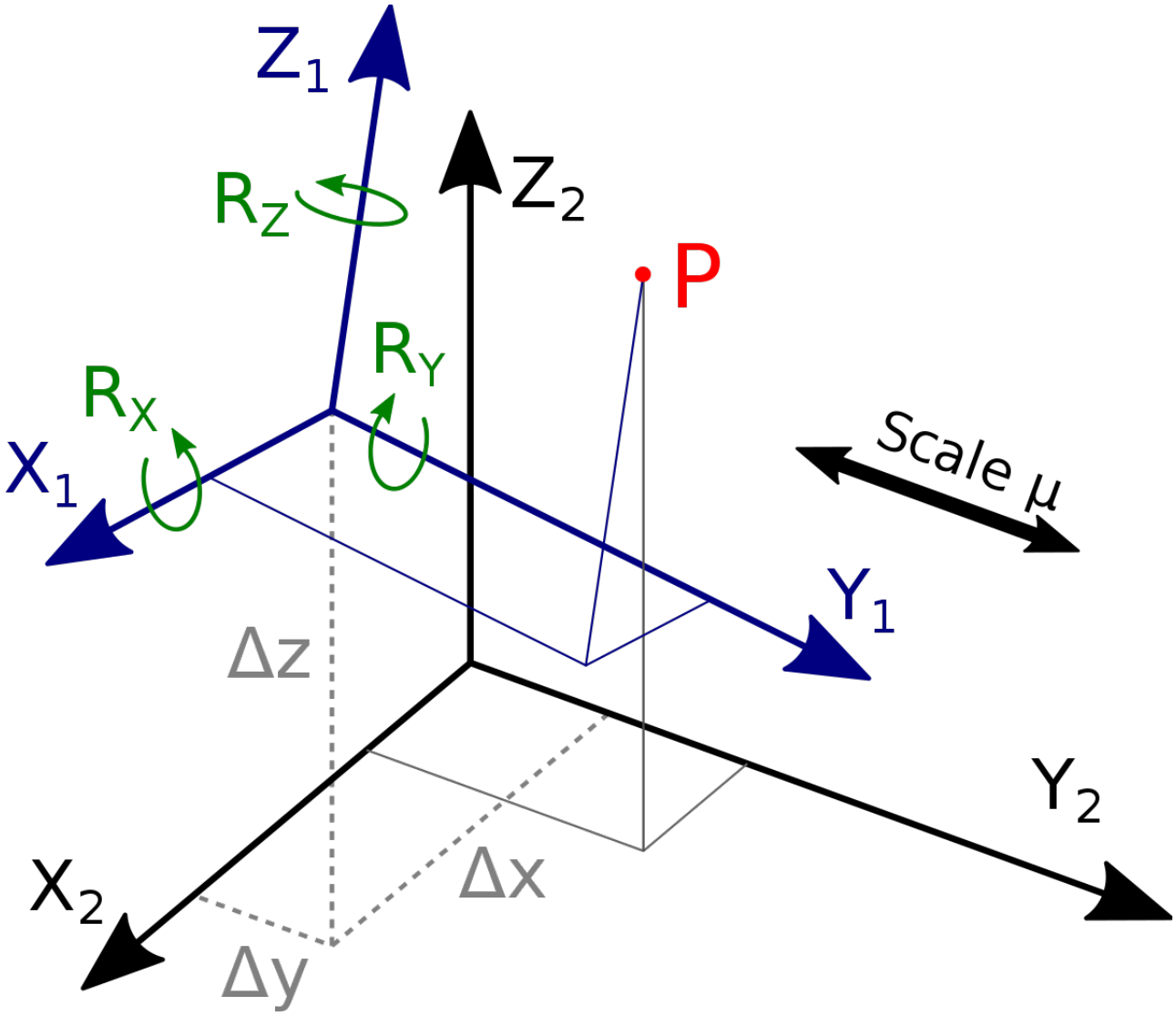
- Transformation of geodetic datum*:
  - 7-parameters Helmert transformation
  - Affine/triangle based transformation
- Heights are calculated separately (using geoid)

Gauss-Krüger zones

- 3° degrees wide
- Slovenia: zone 5 (13° - 17° E)

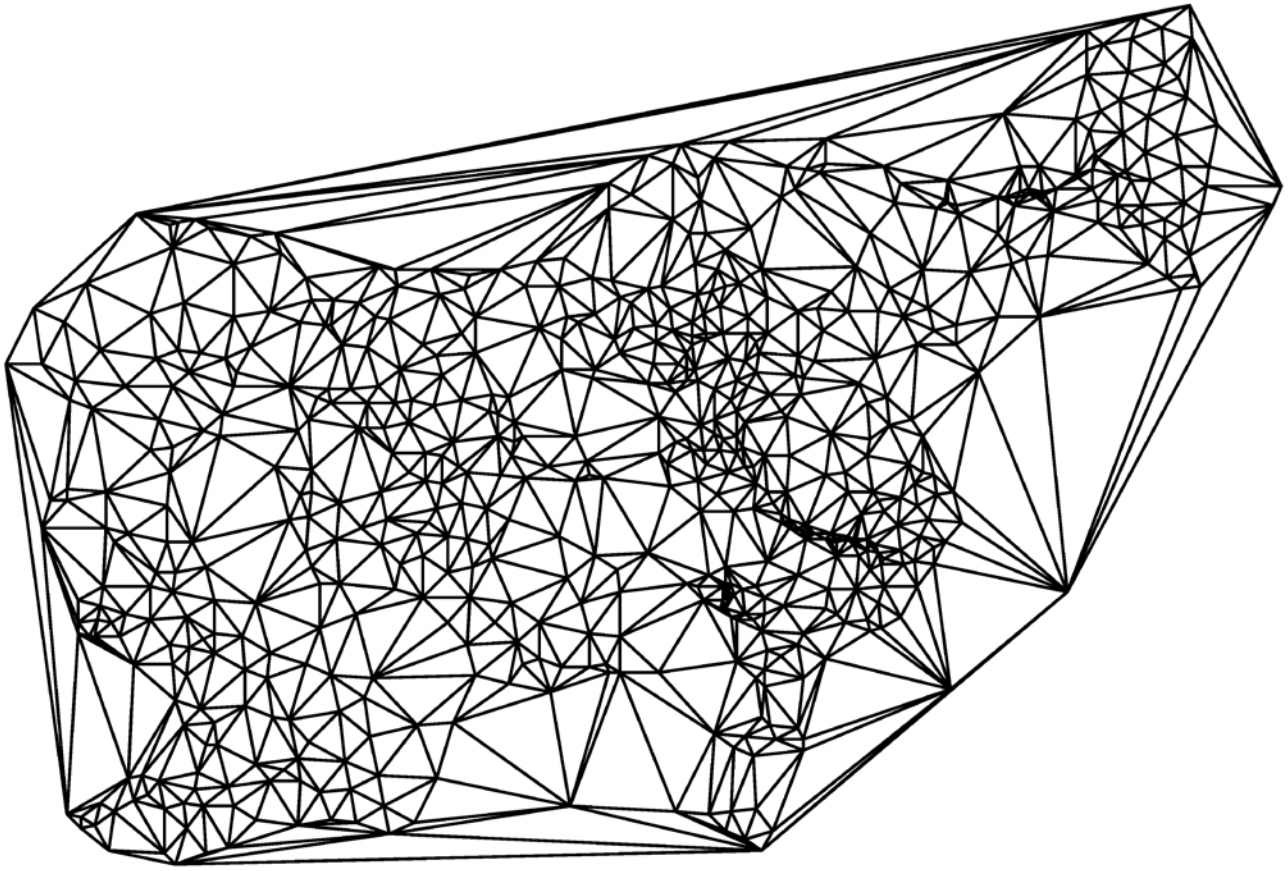


Helmert transformation



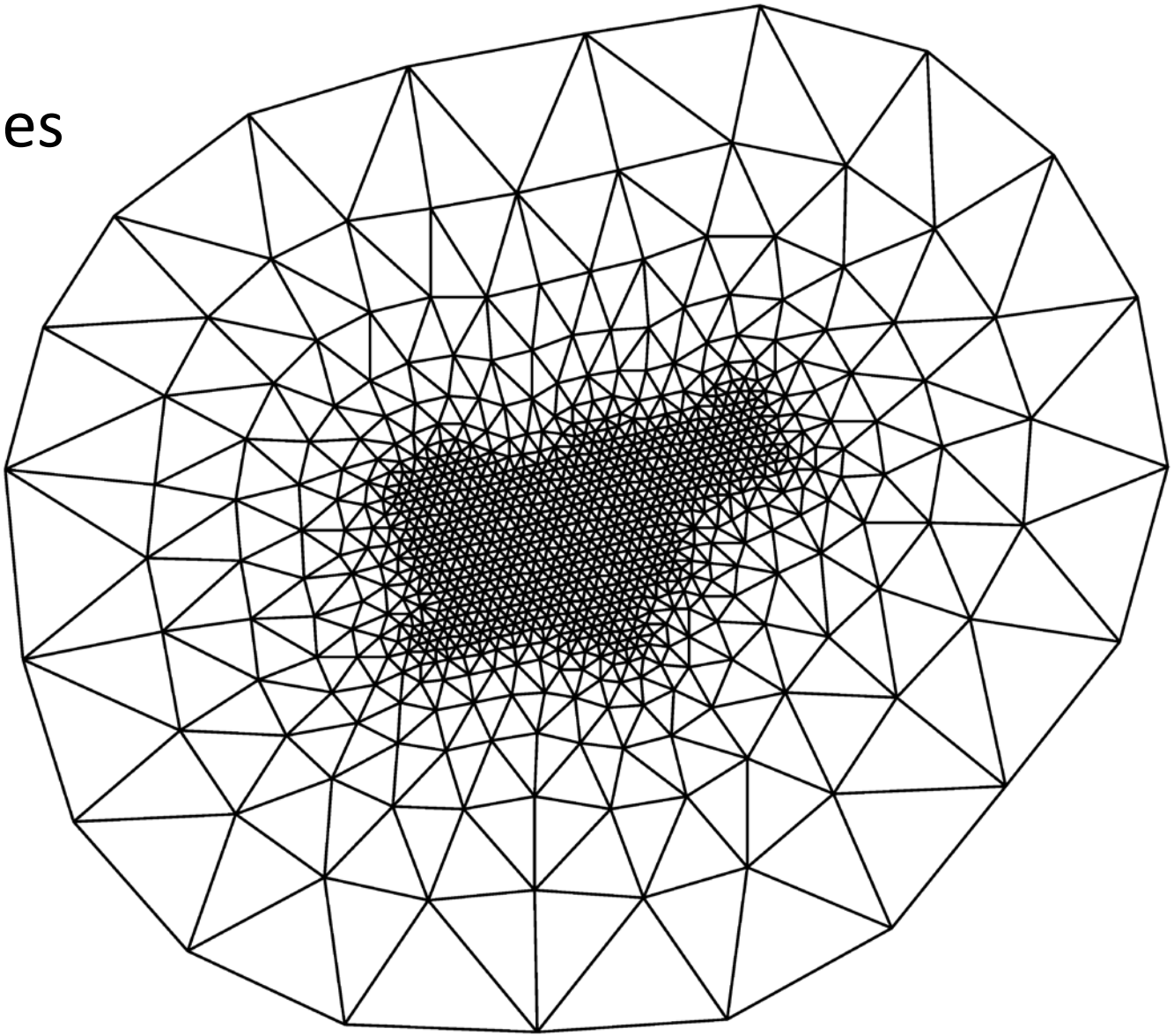
Affine/triangle transformation

- Precalculated from 899 reference virtual tie points (residing in both coordinate systems)



Affine/triangle transformation

- 1776 triangles



GK-SLO tool

- https://geocoordinateconverter.tk
- Source (LGPL v2.1): https://github.com/mrihtar/GeoCoordinateConverter
- Supports all conversions:
  - xy (D96/TM) ↔ ϕλ (ETRS89)
  - xy (D48/GK) ↔ xy (D96/TM) with Helmert transf.
  - xy (D48/GK) ↔ ϕλ (ETRS89) with Helmert transf.
  - xy (D48/GK) ↔ xy (D96/TM) with Affine transf.
  - xy (D48/GK) ↔ ϕλ (ETRS89) with Affine transf.

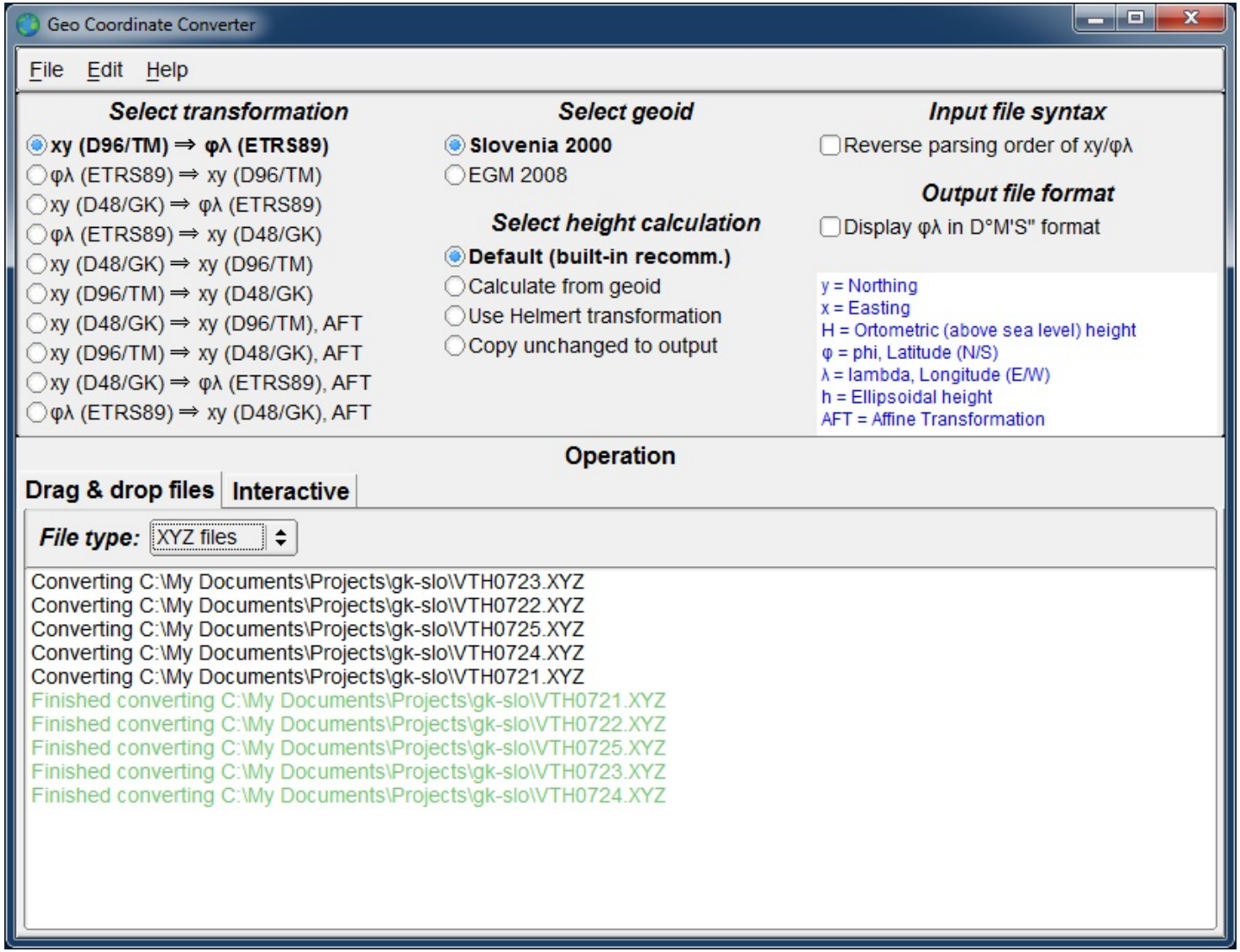
GK-SLO tool

- Input data format:
  - ASCII files like SiTra .xyz or LiDAR .asc
    `[<label> ]<fi|x> <la|y> <h|H>`
    `[<label>;]<fi|x>;<la|y>;<h|H>`
  - ESRI shapefiles .shp (ArcGIS)
- Two geoid models
  - Slo 2000
  - EGM 2008
- Written in C (in 2014)
  - Very fast!
  - Uses Shapelib, FLTK and Deelx libraries (supplied)
- Same precision as official tools!
- Works on all platforms (Windows, Unix, MacOS)
- Command line and GUI version

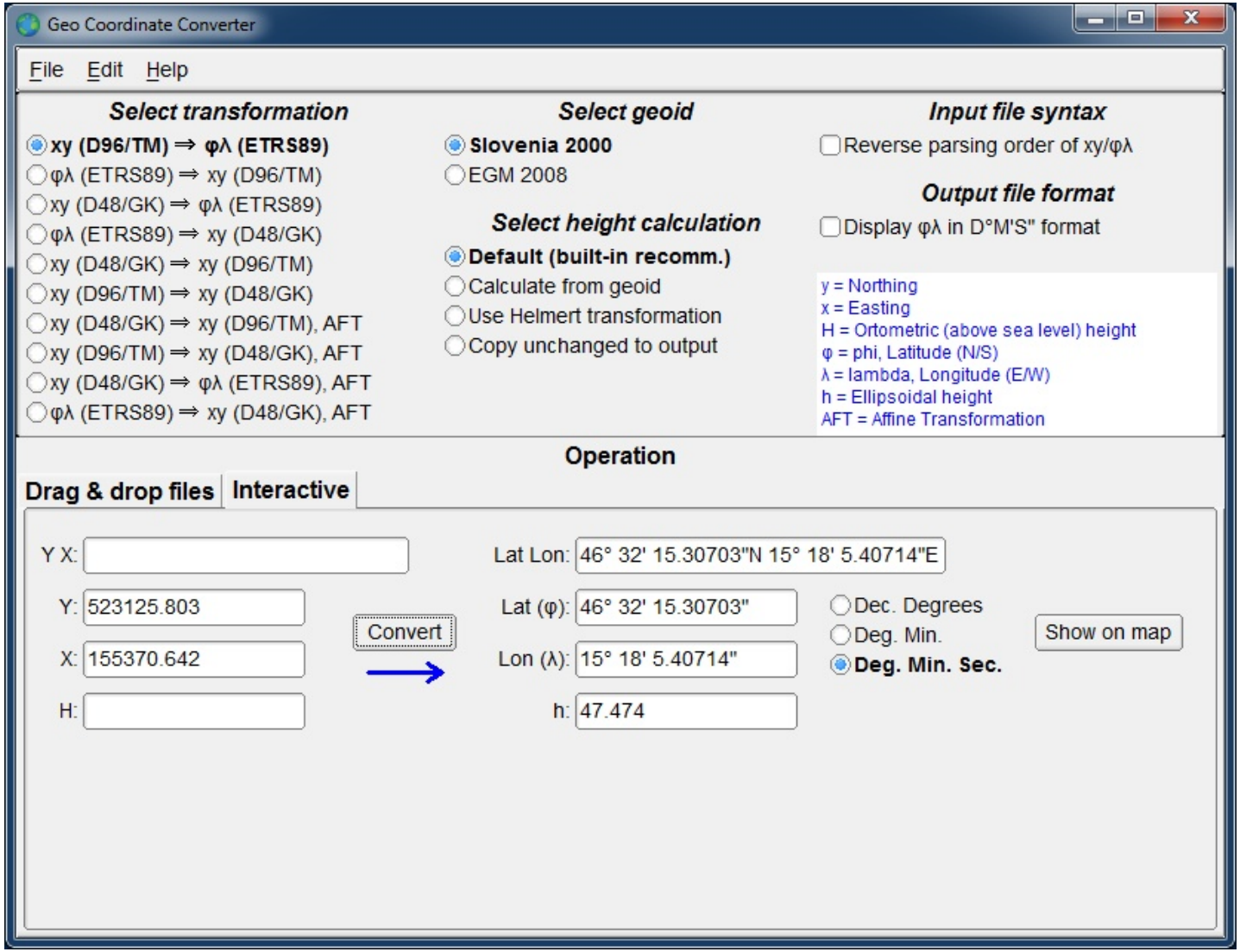
Typical code example

```
in.fi = in.fi*PI/180.0;
in.la = in.la*PI/180.0;

dl = in.la - tm.lambda0;
dl2 = pow(dl,2); dl3 = pow(dl,3); dl4 = pow(dl,4); dl5 = pow(dl,5);
dl6 = pow(dl,6); dl7 = pow(dl,7); dl8 = pow(dl,8);

sinFi = sin(in.fi);
sin2Fi = pow(sinFi,2);
cosFi = cos(in.fi);
cos2Fi = pow(cosFi,2); cos3Fi = pow(cosFi,3); cos4Fi = pow(cosFi,4);
cos5Fi = pow(cosFi,5); cos6Fi = pow(cosFi,6); cos7Fi = pow(cosFi,7);
cos8Fi = pow(cosFi,8);
tanFi = tan(in.fi);
tan2Fi = pow(tanFi,2); tan4Fi = pow(tanFi,4); tan6Fi = pow(tanFi,6);

out->x = L
  + tanFi/2.0*N*cos2Fi*dl2
  + tanFi/24.0*N*cos4Fi*(5.0 - tan2Fi + 9.0*ni2 + 4.0*ni4)*dl4
  + tanFi/720.0*N*cos6Fi*(61.0 - 58.0*tan2Fi + tan4Fi + 270.0*ni2 - 330.0*tan2Fi*ni2)*dl6
  + tanFi/40320.0*N*cos8Fi*(1385.0 - 3111.0*tan2Fi + 543.0*tan4Fi - tan6Fi)*dl8;
out->y = N*cosFi*dl
  + 1.0/6.0*N*cos3Fi*(1.0 - tan2Fi + ni2)*dl3
  + 1.0/120.0*N*cos5Fi*(5.0 - 18.0*tan2Fi + tan4Fi + 14.0*ni2 - 58.0*tan2Fi*ni2)*dl5
  + 1.0/5040.0*N*cos7Fi*(61.0 - 479.0*tan2Fi + 179.0*tan4Fi - tan6Fi)*dl7;
```

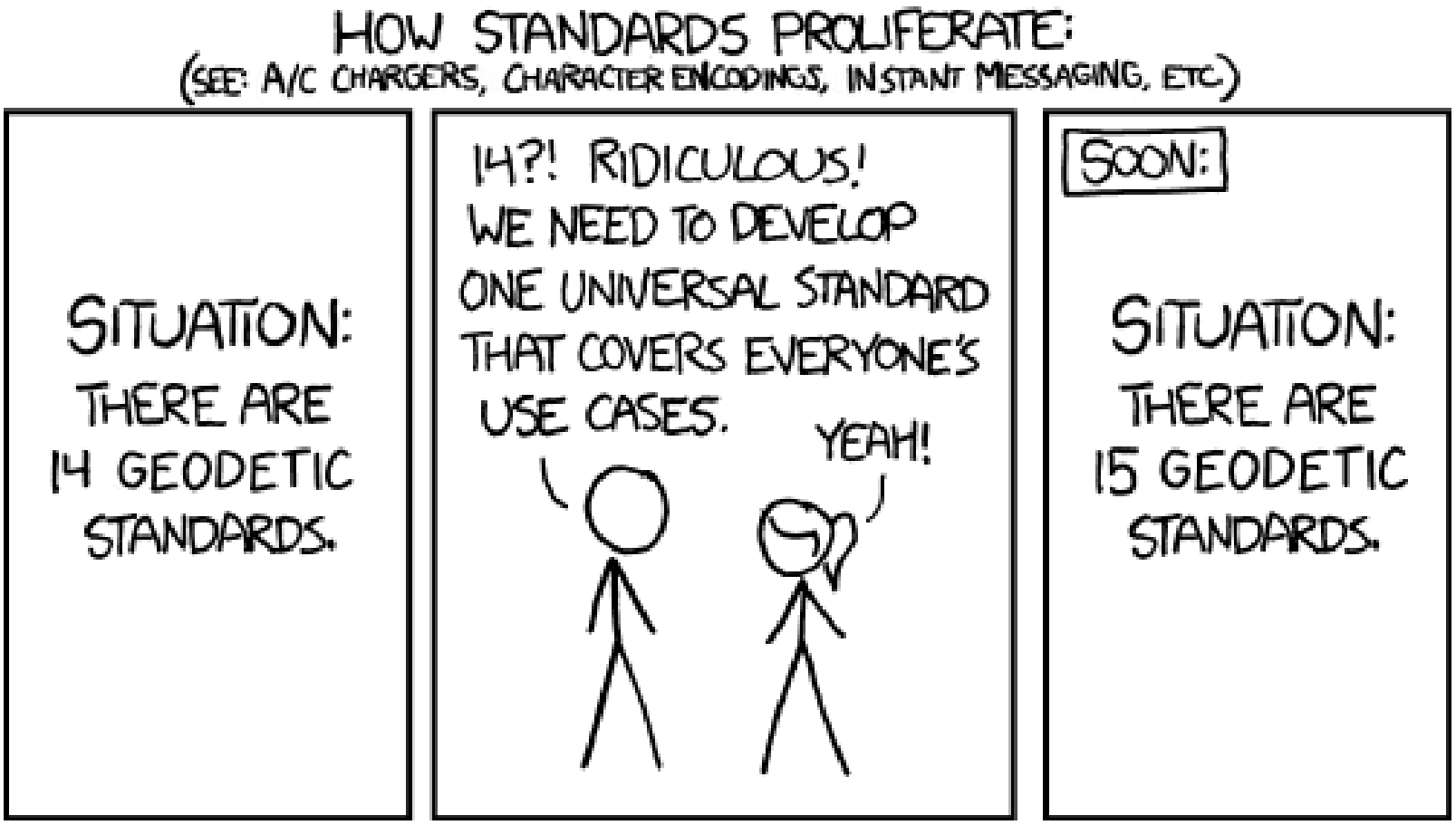
GK-SLO GUI


Geo Coordinate Converter
File Edit Help
Select transformation
xy (D96/TM) ⇒ φλ (ETRS89)
φλ (ETRS89) ⇒ xy (D96/TM)
xy (D48/GK) ⇒ φλ (ETRS89)
φλ (ETRS89) ⇒ xy (D48/GK)
xy (D48/GK) ⇒ xy (D96/TM)
xy (D96/TM) ⇒ xy (D48/GK)
xy (D48/GK) ⇒ xy (D96/TM), AFT
xy (D96/TM) ⇒ xy (D48/GK), AFT
xy (D48/GK) ⇒ φλ (ETRS89), AFT
φλ (ETRS89) ⇒ xy (D48/GK), AFT
Select geoid
Slovenia 2000
EGM 2008
Select height calculation
Default (built-in recomm.)
Calculate from geoid
Use Helmert transformation
Copy unchanged to output
Input file syntax
Reverse parsing order of xy/φλ
Output file format
Display φλ in D°M'S" format
y = Northing
x = Easting
H = Ortometric (above sea level) height
φ = phi, Latitude (N/S)
λ = lambda, Longitude (E/W)
h = Ellipsoidal height
AFT = Affine Transformation
Operation
Drag & drop files
Interactive
File type: XYZ files
Converting C:\My Documents\Projects\gk-slo\VTH0723.XYZ
Converting C:\My Documents\Projects\gk-slo\VTH0722.XYZ
Converting C:\My Documents\Projects\gk-slo\VTH0725.XYZ
Converting C:\My Documents\Projects\gk-slo\VTH0724.XYZ
Converting C:\My Documents\Projects\gk-slo\VTH0721.XYZ
Finished converting C:\My Documents\Projects\gk-slo\VTH0721.XYZ
Finished converting C:\My Documents\Projects\gk-slo\VTH0722.XYZ
Finished converting C:\My Documents\Projects\gk-slo\VTH0725.XYZ
Finished converting C:\My Documents\Projects\gk-slo\VTH0723.XYZ
Finished converting C:\My Documents\Projects\gk-slo\VTH0724.XYZ

GK-SLO GUI


Geo Coordinate Converter
File Edit Help
Select transformation
xy (D96/TM) ⇒ φλ (ETRS89)
φλ (ETRS89) ⇒ xy (D96/TM)
xy (D48/GK) ⇒ φλ (ETRS89)
φλ (ETRS89) ⇒ xy (D48/GK)
xy (D48/GK) ⇒ xy (D96/TM)
xy (D96/TM) ⇒ xy (D48/GK)
xy (D48/GK) ⇒ xy (D96/TM), AFT
xy (D96/TM) ⇒ xy (D48/GK), AFT
xy (D48/GK) ⇒ φλ (ETRS89), AFT
φλ (ETRS89) ⇒ xy (D48/GK), AFT
Select geoid
Slovenia 2000
EGM 2008
Select height calculation
Default (built-in recomm.)
Calculate from geoid
Use Helmert transformation
Copy unchanged to output
Input file syntax
Reverse parsing order of xy/φλ
Output file format
Display φλ in D°M'S" format
y = Northing
x = Easting
H = Ortometric (above sea level) height
φ = phi, Latitude (N/S)
λ = lambda, Longitude (E/W)
h = Ellipsoidal height
AFT = Affine Transformation
Operation
Drag & drop files
Interactive
Y X:
Y: 523125.803
X: 155370.642
H:
Convert
Lat Lon: 46° 32' 15.30703"N 15° 18' 5.40714"E
Lat (φ): 46° 32' 15.30703"
Lon (λ): 15° 18' 5.40714"
h: 47.474
Dec. Degrees
Deg. Min.
Deg. Min. Sec.
Show on map

Questions?


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)
SITUATION: THERE ARE 14 GEODETIC STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 GEODETIC STANDARDS.