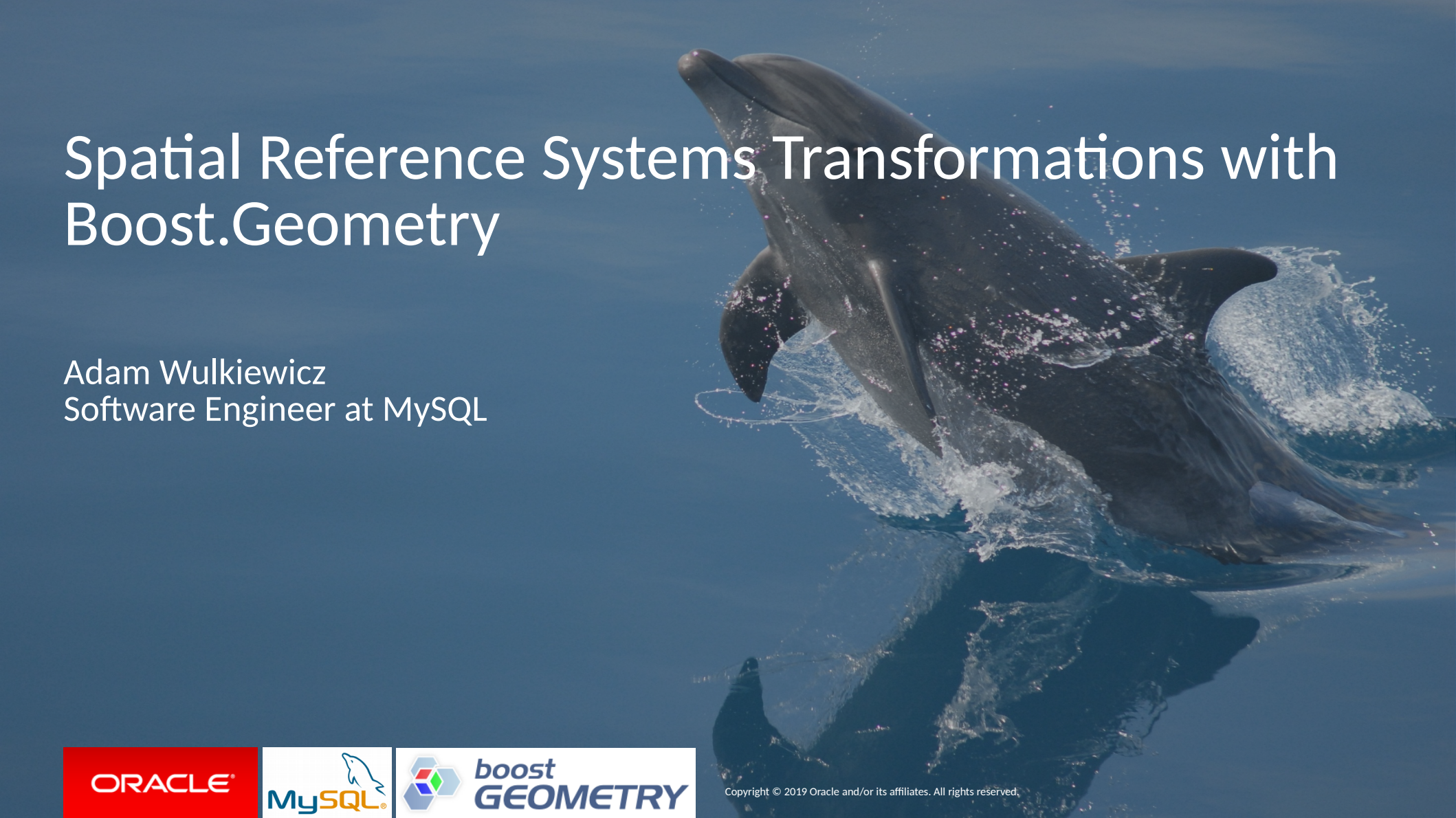
Spatial Reference Systems Transformations with Boost.Geometry

Adam Wulkiewicz
Software Engineer at MySQL



ORACLE MySQL boost GEOMETRY

Spatial reference system

A spatial reference system (SRS) or coordinate reference system (CRS) is a coordinate-based local, regional or global system used to locate geographical entities. A spatial reference system defines a specific map projection, as well as transformations between different spatial reference systems.

en.wikipedia.org/wiki/Spatial_reference_system

Spatial reference system
SRID

Spatial reference systems can be referred to using a SRID integer.

- EPSG
  International Association of Oil & Gas Producers
  (European Petroleum Survey Group)
- ESRI
  Environmental Systems Research Institute
- IAU2000
  International Astronomical Union

Spatial reference system Resources

- www.epsg.org
- epsg.io
- spatialreference.org

Spatial reference system
WKT – EPSG:4326 / WGS 84

```
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",0.0174532925199328,
        AUTHORITY["EPSG","9122"]],
    AUTHORITY["EPSG","4326"]]
```

Spatial reference system
proj4 – EPSG:4326 / WGS 84

```
+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs
```

proj4

- Version:
  5.2.0
- Documentation:
  proj4.org
- GitHub:
  github.com/OSGeo/proj.4

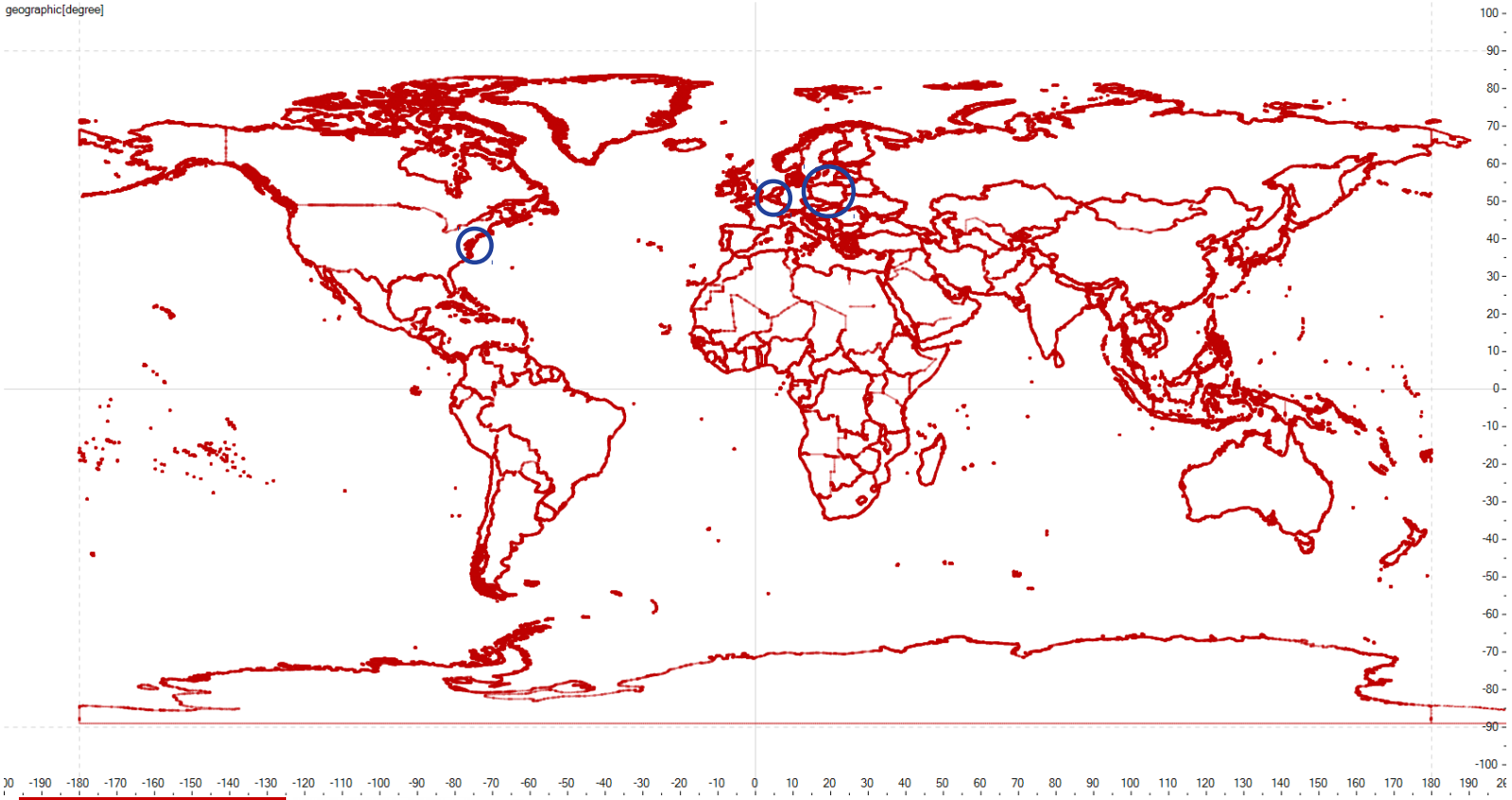
Boost.Geometry

- Version:
  1.69.0 / 1.70.0
- Documentation:
  www.boost.org/libs/geometry
- Mailing list:
  lists.boost.org/geometry
- GitHub:
  github.com/boostorg/geometry

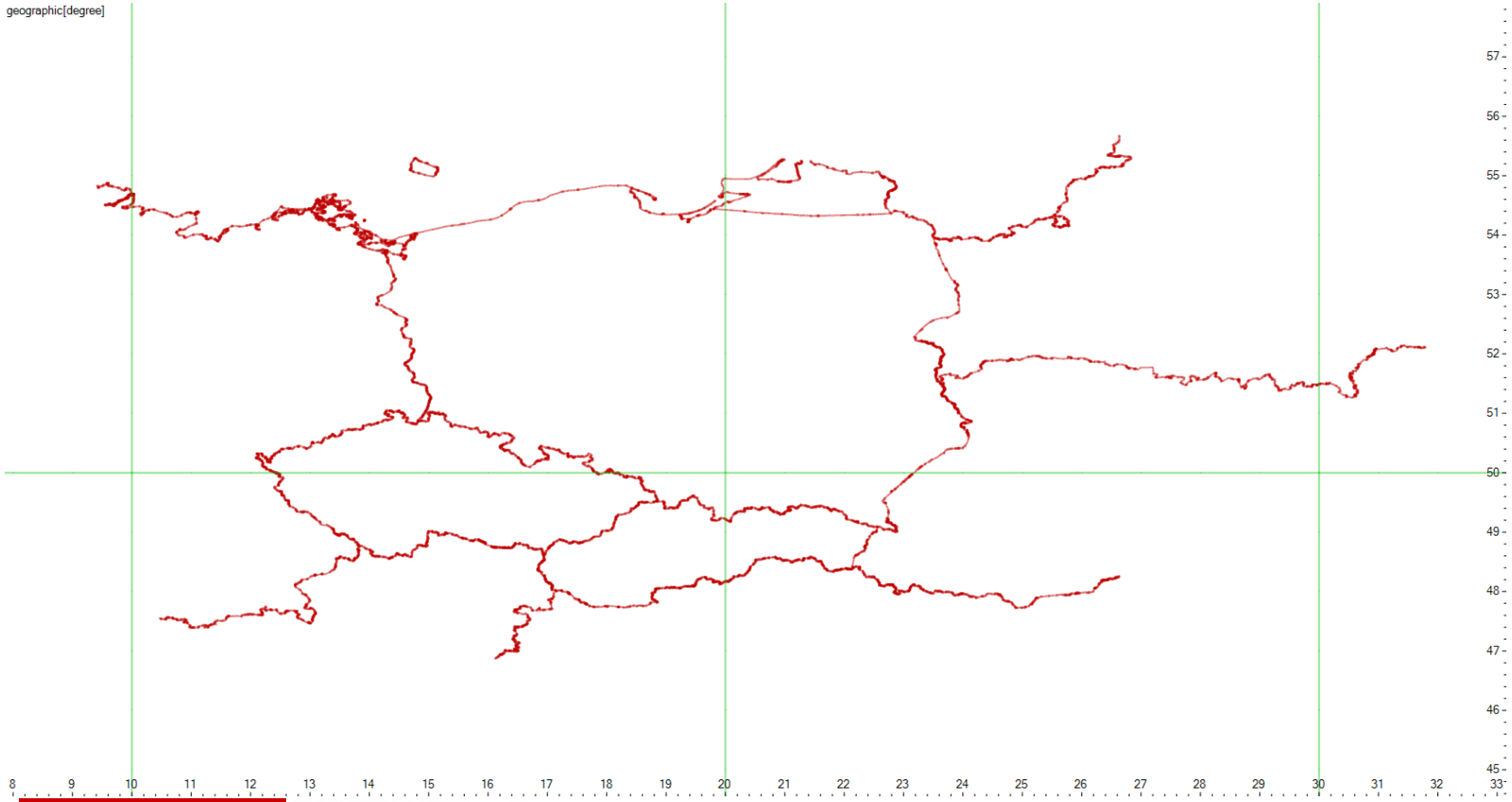
Data

- Source: ec.europa.eu/eurostat/web/gisco
- Geodata > Reference data > Countries 2016 > 1:1 Million > SHP > CNTR_BN_01M_2016_4326
- CNTR_BN_01M_2016_4326.shp

Data
EPSG:4326 / WGS 84

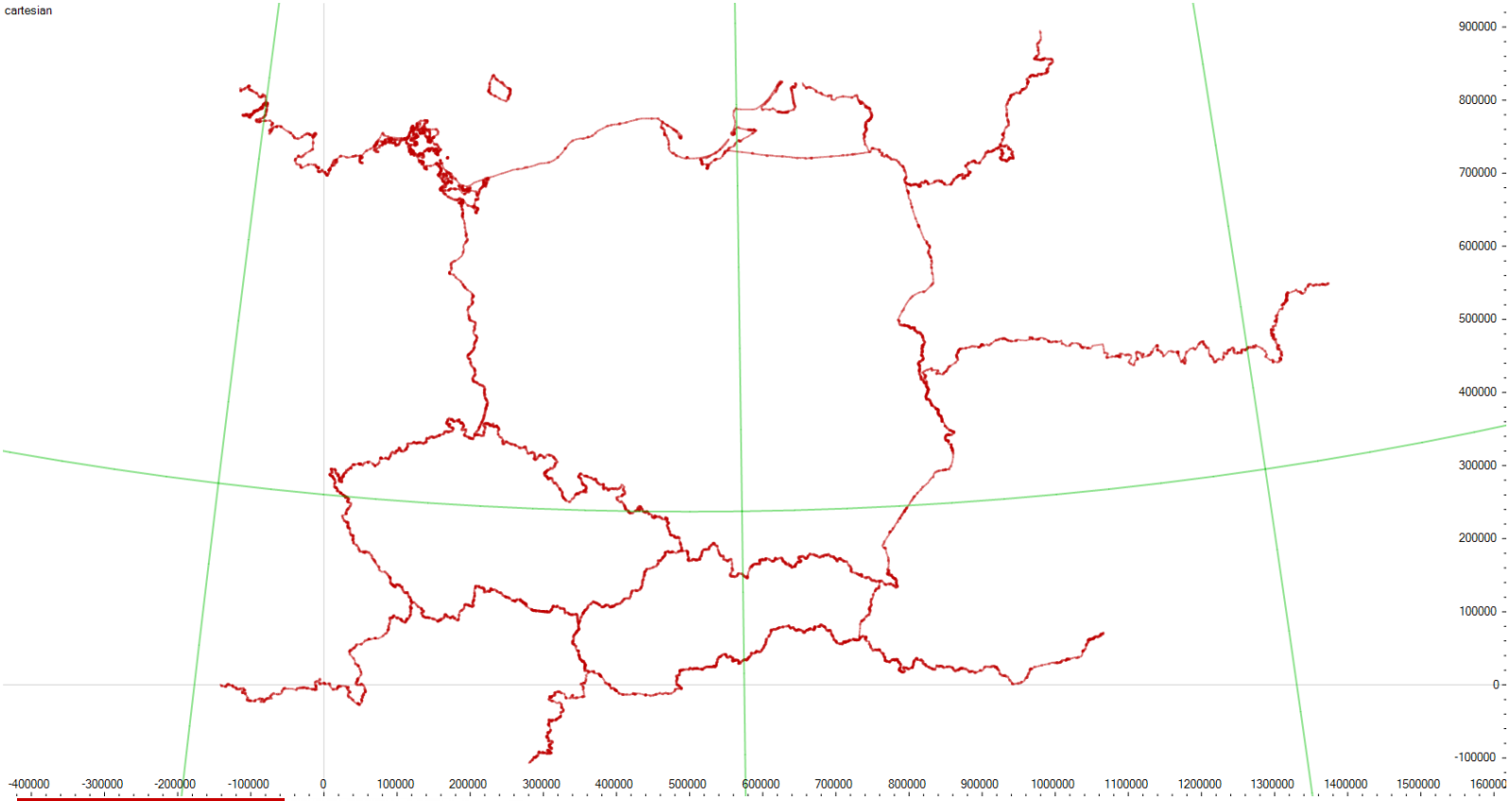


Example 1 – Poland
EPSG:4326 / WGS 84



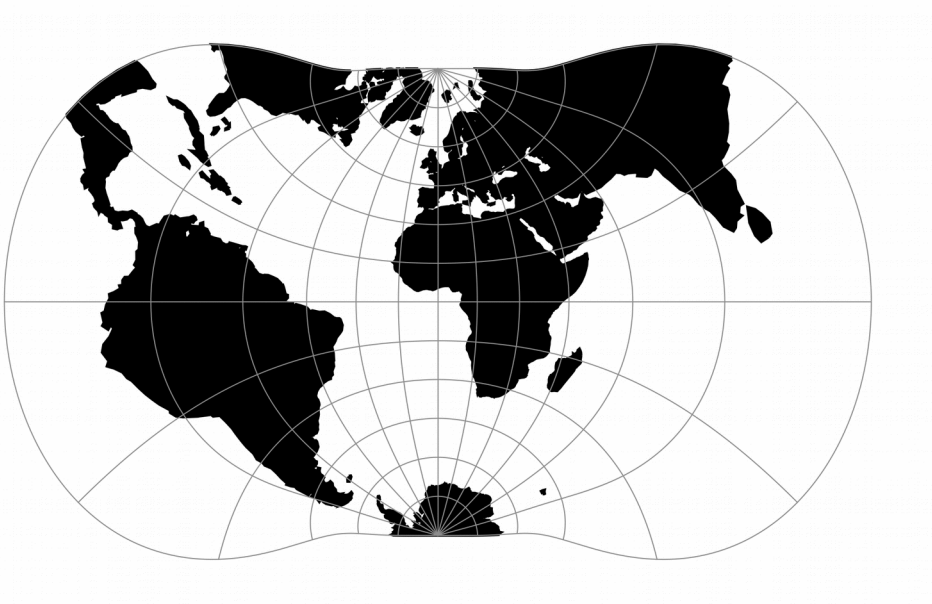
ORACLE MySQL boost GEOMETRY

Example 1 – Poland
EPSG:2180 / ETRS89 / Poland CS92



Example 1 – Poland
EPSG:2180 / ETRS89 / Poland CS92

```
PROJCS["ETRS89 / Poland CS92",
    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",0.01745329251994328,
            AUTHORITY["EPSG","9122"]],
        AUTHORITY["EPSG","4258"]],
    UNIT["metre",1,
        AUTHORITY["EPSG","9001"]],
    PROJECTION["Transverse_Mercator"],
    PARAMETER["latitude_of_origin",0],
    PARAMETER["central_meridian",19],
    PARAMETER["scale_factor",0.9993],
    PARAMETER["false_easting",500000],
    PARAMETER["false_northing",-5300000],
    AUTHORITY["EPSG","2180"],
    AXIS["y",EAST],
    AXIS["x",NORTH]]
```



proj4.org/operations/projections/tmerc.html

Example 1 – EPSG:2180 proj4

```
#include <proj_api.h>

// ...

projPJ from = pj_init_plus("+proj=latlon +datum=WGS84 +no_defs");
projPJ to   = pj_init_plus("+proj=tmerc +lat_0=0 +lon_0=19 +k=0.9993 "
                           "+x_0=500000 +y_0=-5300000 +ellps=GRS80 "
                           "+units=m +no_defs");

std::vector<double> vec_xy;

// Fill vector with data in radians:
// lon0, lat0, lon1, lat1, ...

pj_transform(from, to, vec_xy.size() / 2, 2, &vec_xy[0], &vec_xy[1], NULL);

pj_free(from);
pj_free(to);
```

Example 1 – EPSG:2180
Boost.Geometry

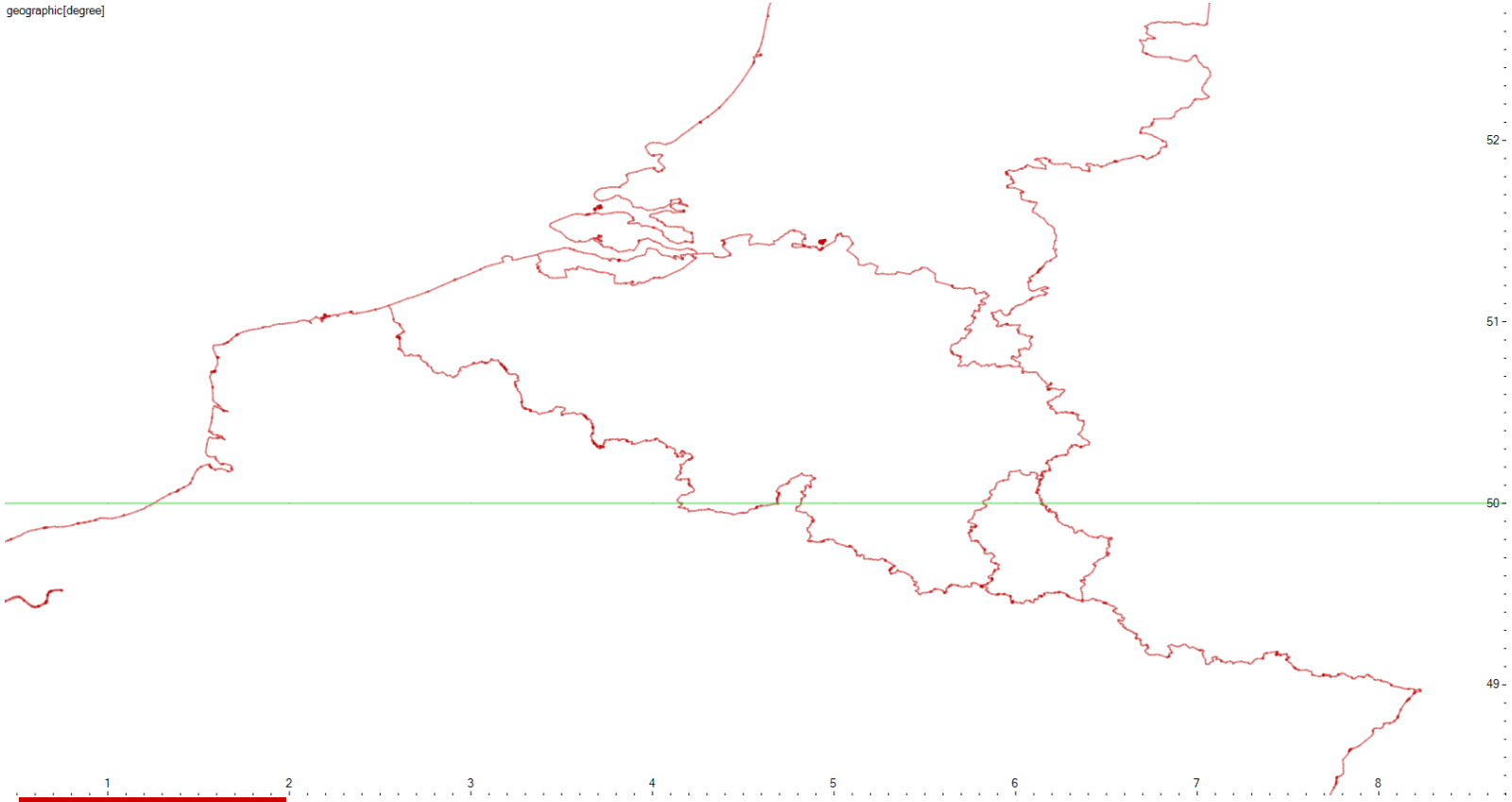
```
#include <boost/geometry.hpp>
#include <boost/geometry/geometries/geometries.hpp>

// ...

namespace bg = boost::geometry;

using point_geo = bg::model::point<double, 2,
                                   bg::cs::geographic<bg::degree>>;
using linestring_geo = bg::model::linestring<point_geo>;
using mlinestring_geo = bg::model::multi_linestring<linestring_geo>;

using point_car = bg::model::point<double, 2, bg::cs::cartesian>;
using linestring_car = bg::model::linestring<point_car>;
using mlinestring_car = bg::model::multi_linestring<linestring_car>;
```

ORACLE MySQL boost GEOMETRY

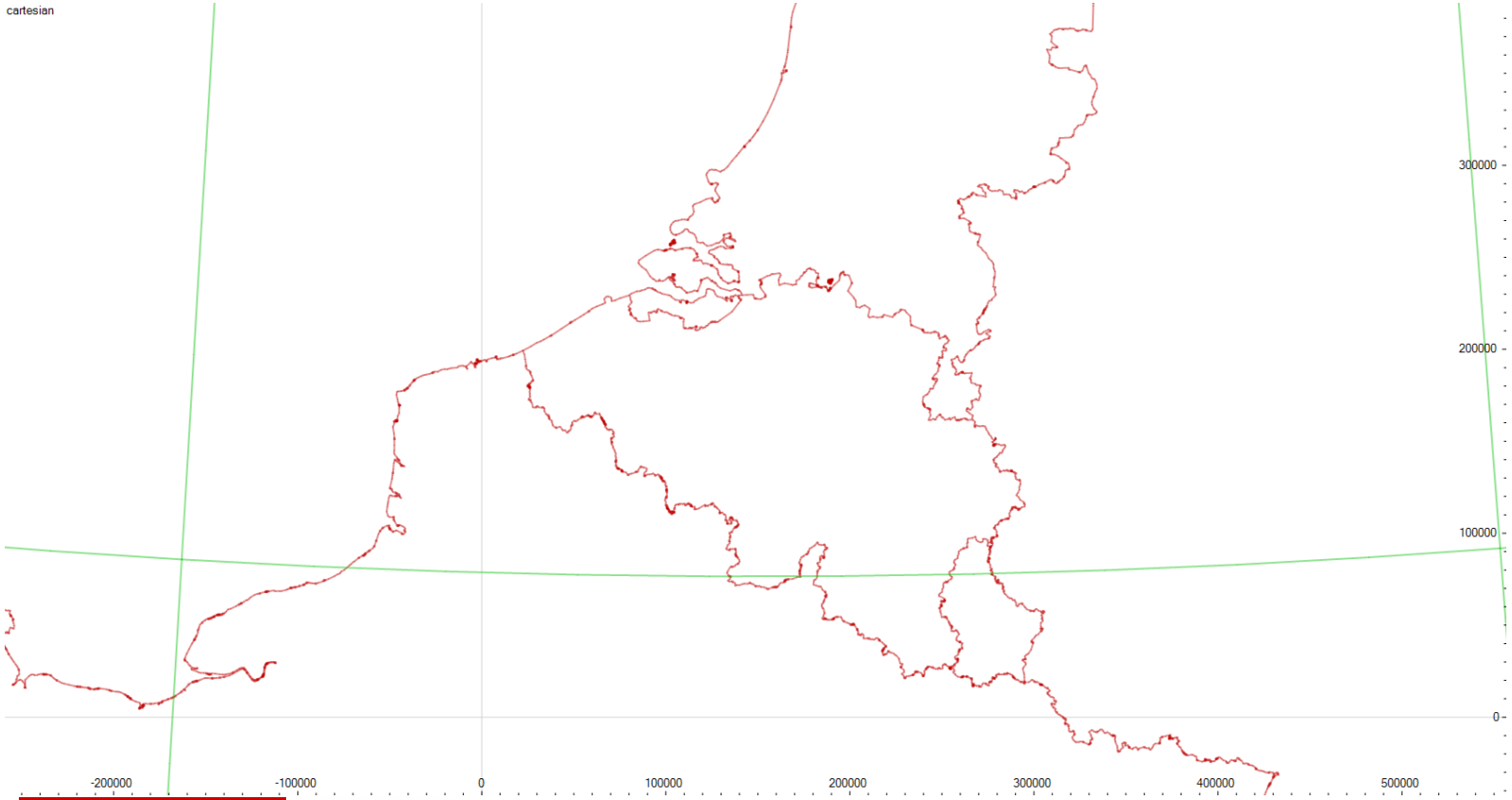
Example 1 – EPSG:2180 Boost.Geometry

```
#include <boost/geometry/srs/transformation.hpp>

// ...

bg::srs::transformation<> tr{
    bg::srs::proj4("+proj=latlon +datum=WGS84 +no_defs"),
    bg::srs::proj4("+proj=tmerc +lat_0=0 +lon_0=19 +k=0.9993 "
                   "+x_0=500000 +y_0=-5300000 +ellps=GRS80 "
                   "+units=m +no_defs") };

mlinestring_geo mls_geo;

// Fill multi-linestring with linestrings containing points

mlinestring_car mls_car;

tr.forward(mls_geo, mls_car);
```

Example 2 – Belgium
EPSG:4326 / WGS 84



Example 2 – Belgium
EPSG:31370 / Belge 1972 / Belgian Lambert 72



Example 2 – Belgium
EPSG:31370 / Belge 1972 / Belgian Lambert 72

```
PROJCS["Belge 1972 / Belgian Lambert 72",
    GEOGCS["Belge 1972",
        DATUM["Reseau_National_Belge_1972",
            SPHEROID["International 1924",6378388,297,
                AUTHORITY["EPSG","7022"]],
            TOWGS84[-106.869,52.2978,-103.724,0.3366,-0.457,1.8422,-1.2747],
            AUTHORITY["EPSG","6313"]],
        PRIMEM["Greenwich",0,
            AUTHORITY["EPSG","8901"]],
        UNIT["degree",0.0174532925199433,
            AUTHORITY["EPSG","9122"]],
        AUTHORITY["EPSG","4313"]],
    PROJECTION["Lambert_Conformal_Conic_2SP"],
    PARAMETER["standard_parallel_1",51.16666723333333],
    PARAMETER["standard_parallel_2",49.8333339],
    PARAMETER["latitude_of_origin",90],
    PARAMETER["central_meridian",4.367486666666666],
    PARAMETER["false_easting",150000.013],
    PARAMETER["false_northing",5400088.438],
    UNIT["metre",1,
        AUTHORITY["EPSG","9001"]],
    AXIS["X",EAST],
    AXIS["Y",NORTH],
    AUTHORITY["EPSG","31370"]]
```

proj4.org/operations/projections/lcc.html

Example 2 – Belgium Boost.Geometry

```
bg::srs::transformation<> tr{

    bg::srs::proj4("+proj=latlon +datum=WGS84 +no_defs"),

    bg::srs::proj4("+proj=lcc +lat_1=51.16666723333333 +lat_2=49.8333339 "
                   "+lat_0=90 +lon_0=4.367486666666666 "
                   "+x_0=150000.013 +y_0=5400088.438 +ellps=intl "
                   "+towgs84=-106.869,52.2978,-103.724,0.3366,"
                           "-0.457,1.8422,-1.2747 "
                   "+units=m +no_defs")
};
```

Example 2 – Belgium
Boost.Geometry

```
using namespace bg::srs::dpar;

bg::srs::transformation<> tr{

  parameters<>(proj_latlon)(datum_wgs84),

  parameters<>(proj_lcc)(ellps_intl)
    (towgs84, {-106.869,52.2978,-103.724,0.3366,-0.457,1.8422,-1.2747})
    (lat_1, 51.16666723333333)(lat_2, 49.8333339)
    (lat_0, 90)(lon_0, 4.367486666666666)
    (x_0, 150000.013)(y_0, 5400088.438)
    (units_m)(no_defs)

};
```

Example 2 – Belgium
Boost.Geometry

```
using namespace bg::srs::spar;

using proj1_t = parameters<proj_latlon, datum_wgs84>;
using proj2_t = parameters<proj_lcc, ellps_intl, towgs84<>, lat_1<>, lat_2<>,
                           lat_0<>, lon_0<>, x_0<>, y_0<>, units_m, no_defs>;

bg::srs::transformation<proj1_t, proj2_t> tr{

  proj1_t(),

  proj2_t(proj_lcc(), ellps_intl(),
          towgs84<>(-106.869,52.2978,-103.724,0.3366,-0.457,1.8422,-1.2747),
          lat_1<>(51.16666723333333), lat_2<>(49.8333339),
          lat_0<>(90), lon_0<>(4.367486666666666),
          x_0<>(150000.013), y_0<>(5400088.438),
          units_m(), no_defs())

};
```

Example 2 – Belgium Boost.Geometry

```
#include <boost/geometry/srs/epsg.hpp>

// ...

bg::srs::transformation<> tr{
    bg::srs::epsg(4326),
    bg::srs::epsg(31370)
};

bg::srs::transformation
  <
    bg::srs::static_epsg<4326>,
    bg::srs::static_epsg<31370>
  > tr;
```

Example 2 – Belgium Benchmarks

Dataset: CNTR_BN_01M_2016_4326 polylines intersecting BOX(2 49, 7 52), 7433 points
Results: time (s)

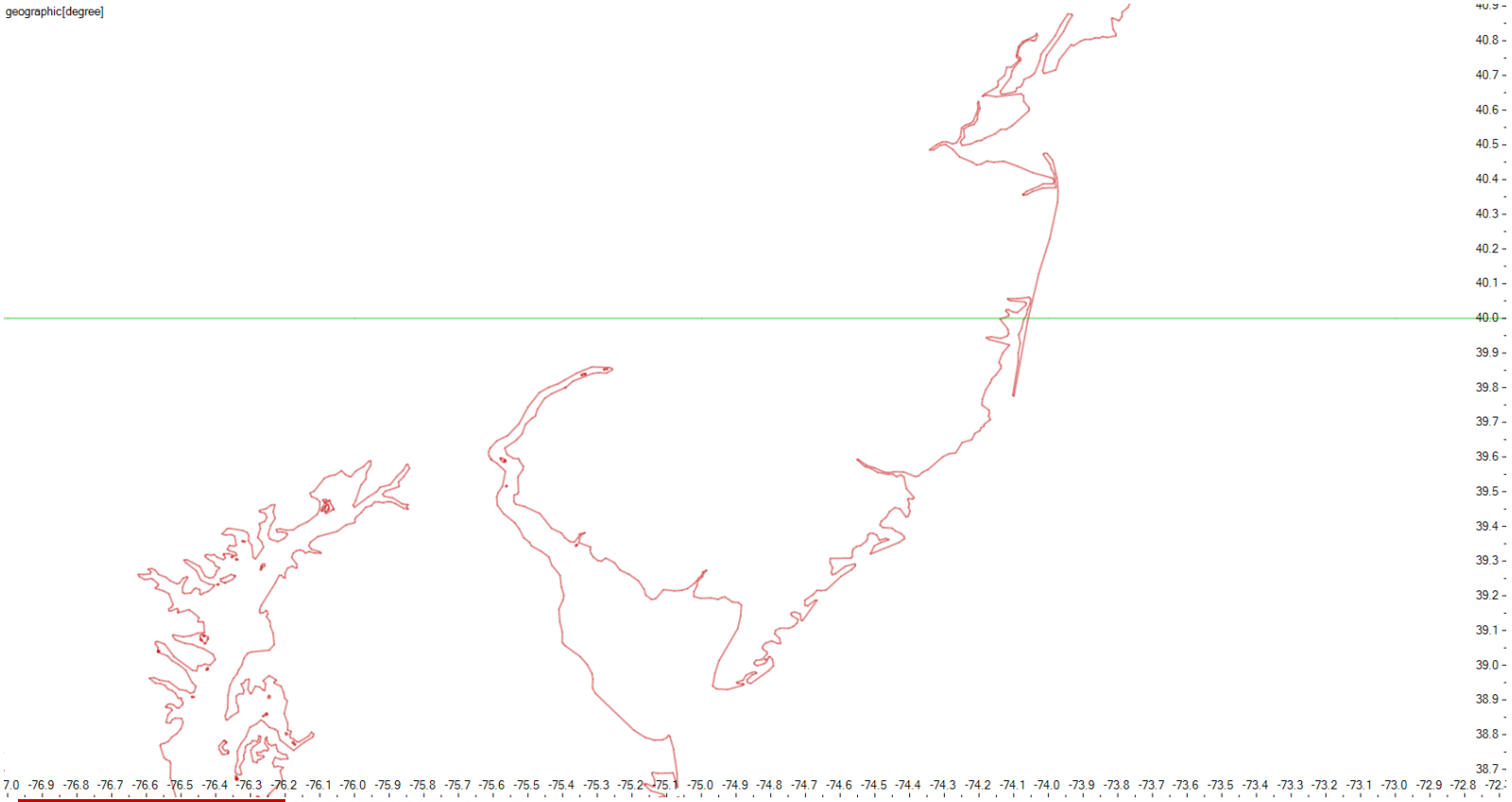
Construct: 5000 creations of transformation followed by 1 forward transformation

	proj4	BG proj4	BG dpar	BG spar
Vc++-14.0 release	0.677	0.158	0.025	0.012
gcc-5.5 -O3	0.164	0.086	0.022	0.014
clang-3.8 -O3	0.127	0.082	0.022	0.015

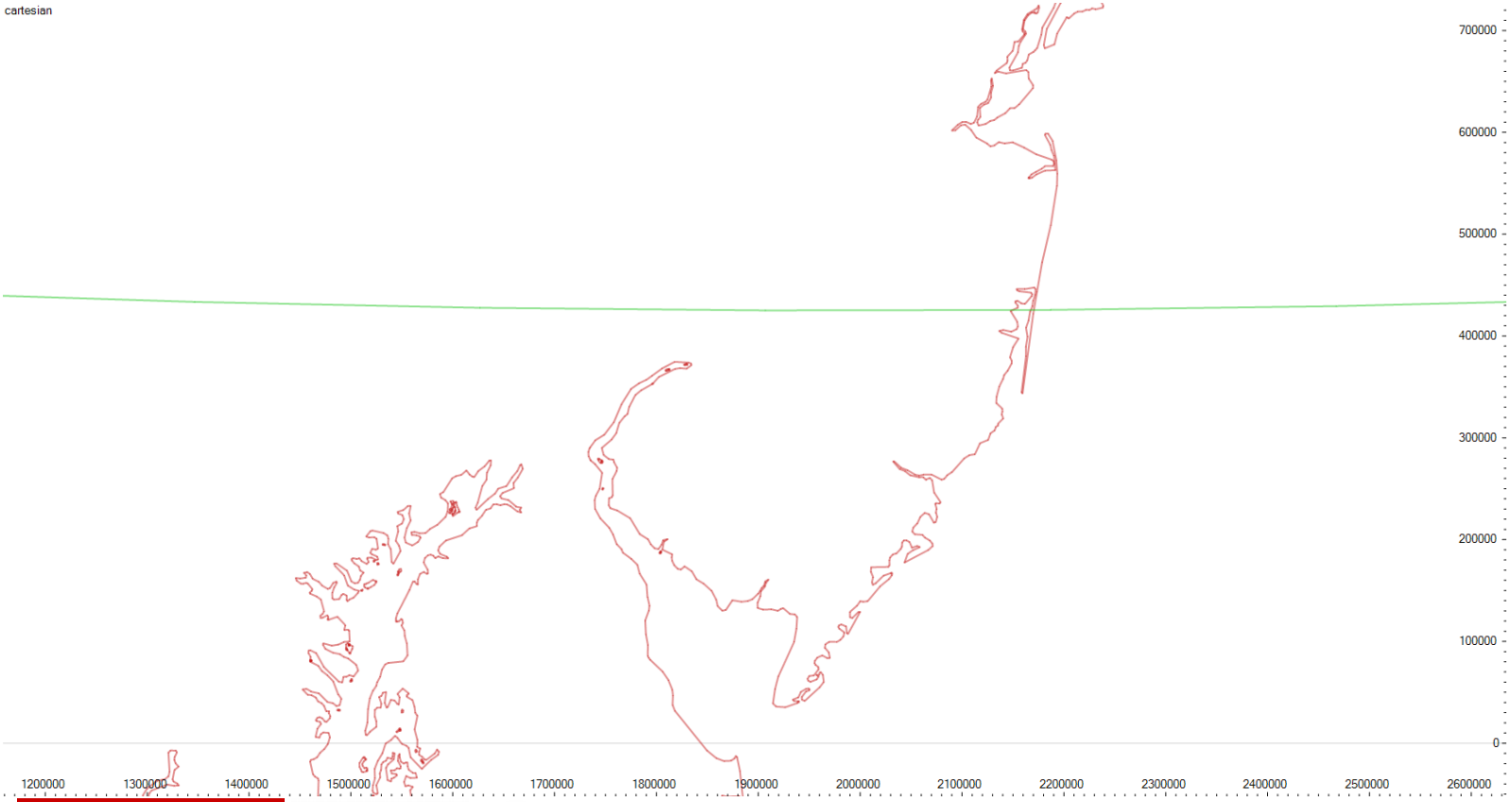
Transform: 1 creation of transformation followed by 500 forward transformations

	proj4	BG proj4	BG dpar	BG spar
Vc++-14.0 release	2.389	2.277	2.277	2.268
gcc-5.5 -O3	3.162	2.870	2.861	2.836
clang-3.8 -O3	4.796	4.501	4.494	4.489

Example 3 – New Jersey
EPSG:4326 / WGS 84

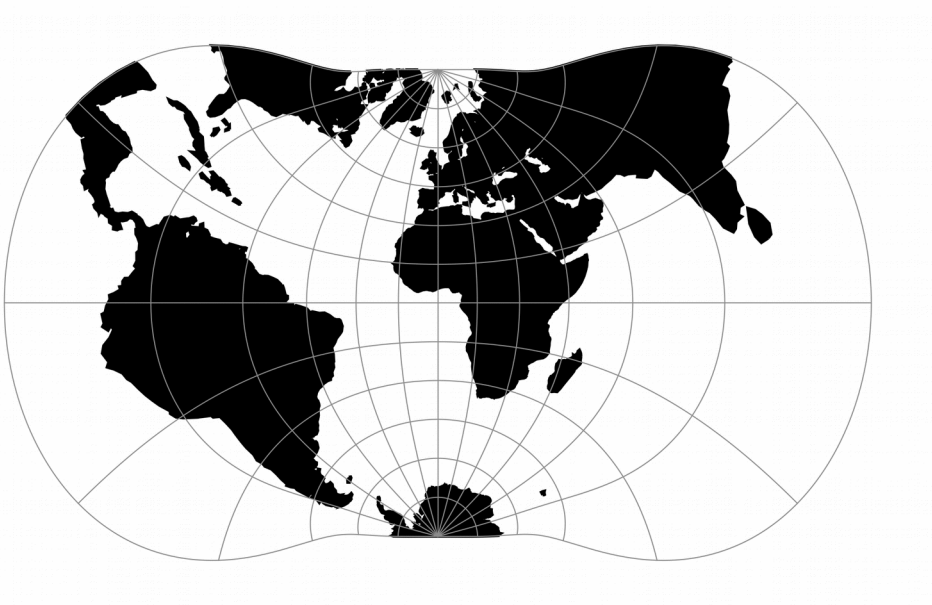


Example 3 – New Jersey
EPSG:32011 / NAD27 / New Jersey



Example 3 – New Jersey
EPSG:32011 / NAD27 / New Jersey

```
PROJCS["NAD27 / New Jersey",
    GEOGCS["NAD27",
        DATUM["North_American_Datum_1927",
            SPHEROID["Clarke 1866",6378206.4,294.9786982139006,
                AUTHORITY["EPSG","7008"]],
            AUTHORITY["EPSG","6267"]],
        PRIMEM["Greenwich",0,
            AUTHORITY["EPSG","8901"]],
        UNIT["degree",0.0174532925199433,
            AUTHORITY["EPSG","9122"]],
        AUTHORITY["EPSG","4267"]],
    PROJECTION["Transverse_Mercator"],
    PARAMETER["latitude_of_origin",38.83333333333334],
    PARAMETER["central_meridian",-74.66666666666667],
    PARAMETER["scale_factor",0.999975],
    PARAMETER["false_easting",2000000],
    PARAMETER["false_northing",0],
    UNIT["US survey foot",0.3048006096012192,
        AUTHORITY["EPSG","9003"]],
    AXIS["X",EAST],
    AXIS["Y",NORTH],
    AUTHORITY["EPSG","32011"]]
```



proj4.org/operations/projections/tmerc.html

Example 2 – New Jersey Boost.Geometry

```
bg::srs::transformation<> tr{
    bg::srs::epsg(4326), bg::srs::epsg(32011)
};

tr.forward(data_geo, data_car);
```

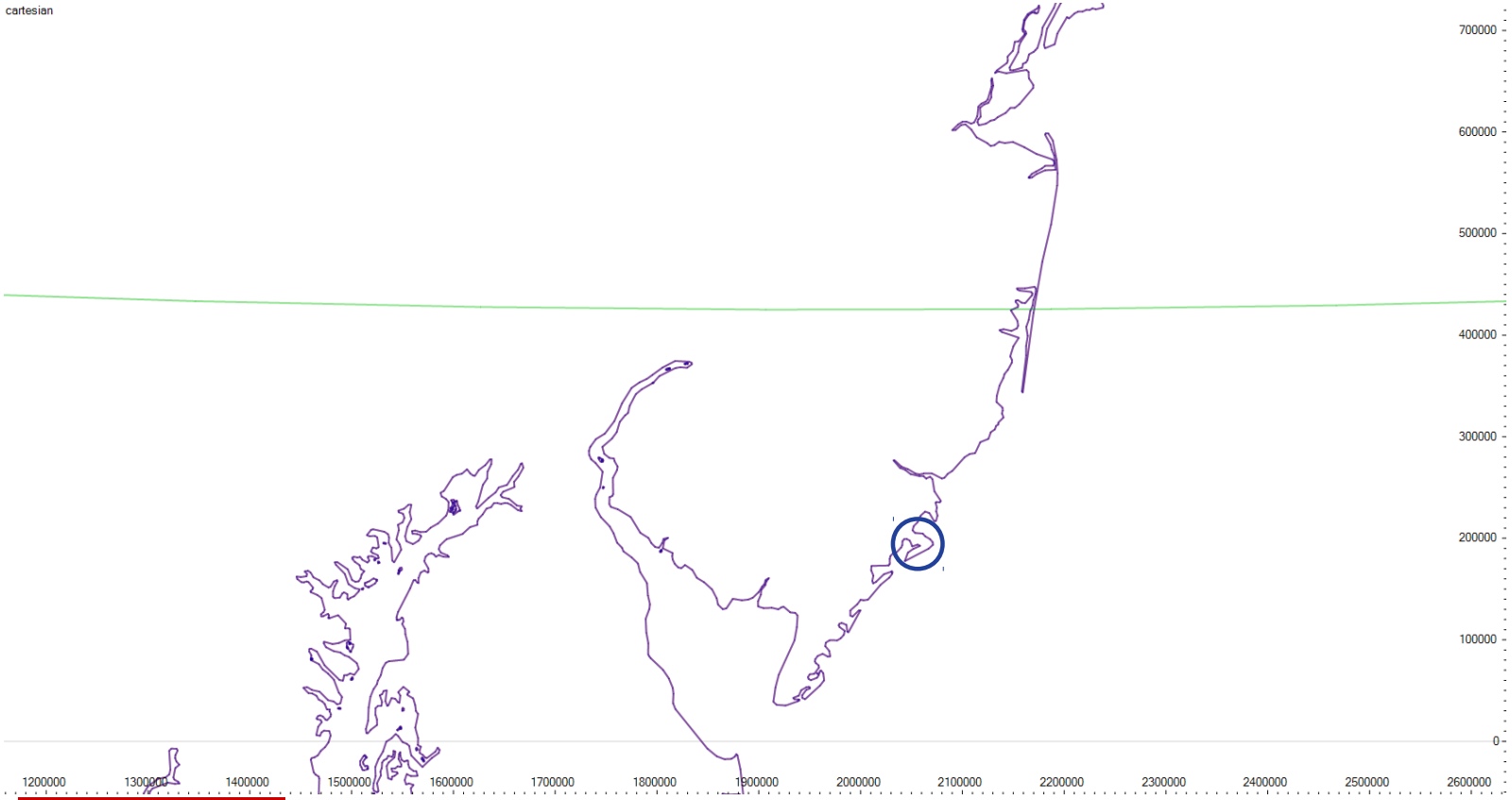
Example 2 – New Jersey Boost.Geometry

```
bg::srs::grids_storage
  <
    bg::srs::ifstream_policy, bg::srs::grids
  > gs;

bg::srs::transformation<> tr{
    bg::srs::epsg(4326), bg::srs::epsg(32011)
};

auto g = tr.initialize_grids(gs);

tr.forward(data_geo, data_car, g);
```

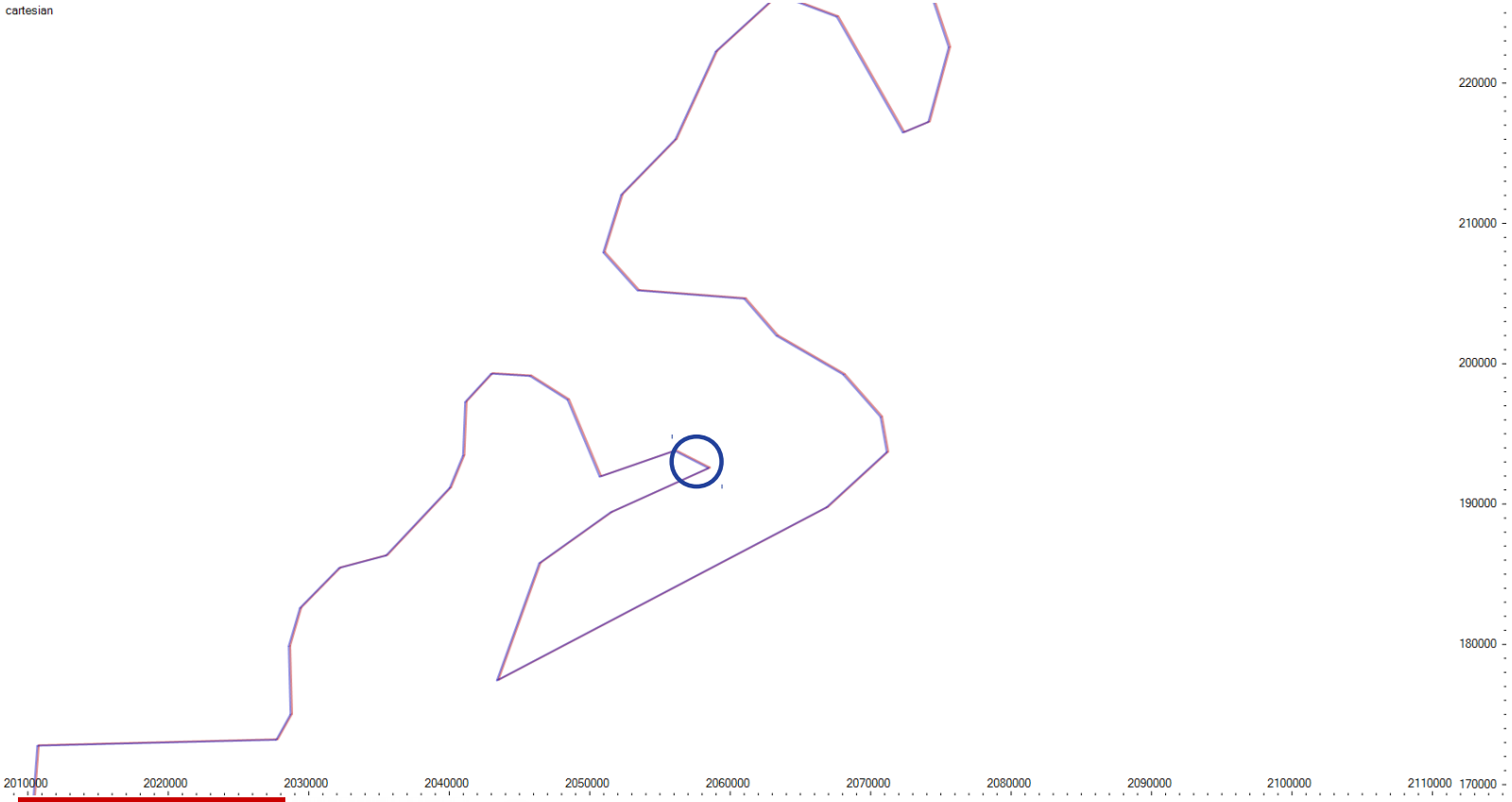
Example 2 – New Jersey Boost.Geometry

```
#include <boost/geometry/srs/shared_grids.hpp>

// ...

bg::srs::grids_storage<bg::srs::ifstream_policy, bg::srs::shared_grids> sgs;

bg::srs::transformation<> tr{ bg::srs::epsg(4326), bg::srs::epsg(32011) };

auto g = tr.initialize_grids(sgs);

boost::thread t1{ [&]() {
    tr.forward(geometry1_geo, geometry1_car, g);
  } };
boost::thread t2{ [&]() {
    tr.forward(geometry2_geo, geometry2_car, g);
  } };

t1.join();
t2.join();
```

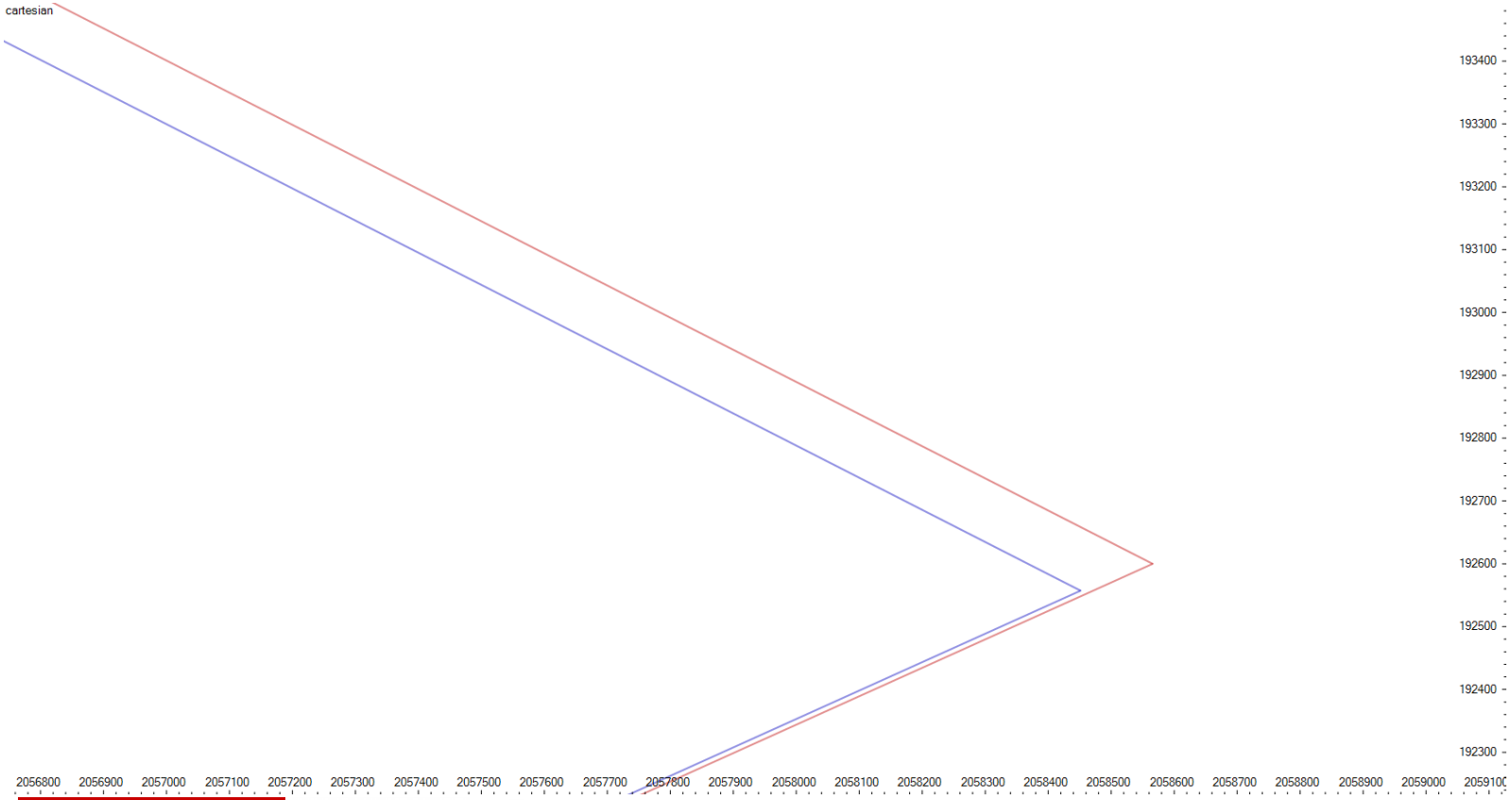
Example 3 – Atlantic City
EPSG:32011 / NAD27 / New Jersey



Example 3 – Atlantic City
EPSG:32011 / NAD27 / New Jersey



Example 3 – Atlantic City
EPSG:32011 / NAD27 / New Jersey



Example 3 – New Jersey Benchmarks

Dataset: CNTR_BN_01M_2016_4326 polylines intersecting BOX(-77 37, -75 43), 6562 points

Creation: 5000 creations of transformation and initializations of grids followed by 1 forward transformation

	proj4 no grids	proj4 shr grids	BGp4 no grids	BGp4 grids	BGp4 shr grids
vc++-14.0 rel	0.663	0.654	0.111	0.124	0.126
gcc-4.4 -O3	0.126	0.123	0.058	0.104	0.106
clang-3.8 -O3	0.105	0.101	0.052	0.098	0.102

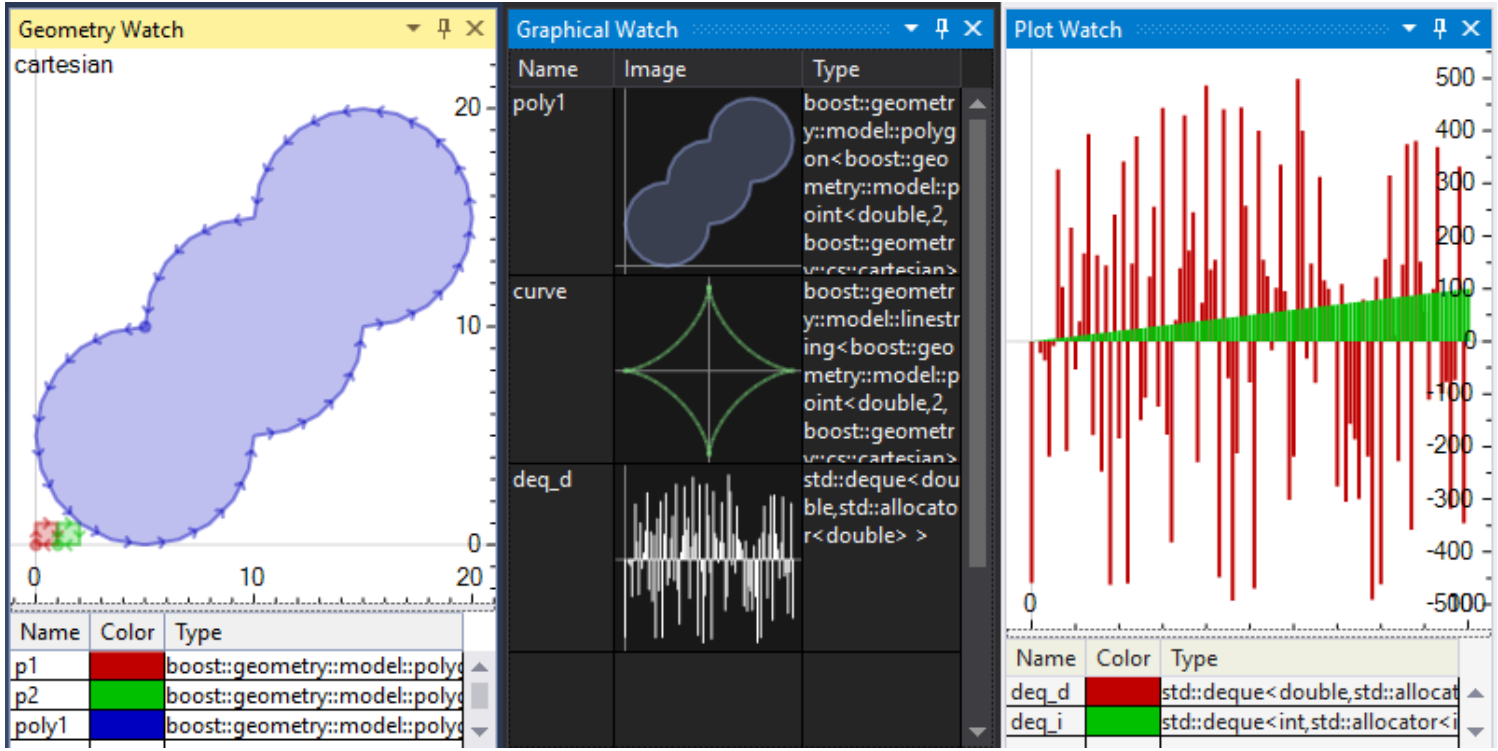
Transform: 1 creation of transformation followed by 500 forward transformations

	proj4 no grids	proj4 shr grids	BGp4 no grids	BGp4 grids	BGp4 shr grids
vc++-14.0 rel	0.845	1.045	0.411	0.925	0.938
gcc-4.4 -O3	0.460	0.478	0.231	0.233	0.232
clang-3.8 -O3	0.886	0.897	0.686	0.686	0.687

Boost.Geometry vs proj4

- User- defined coordinate type
- More flexible
- Faster
- Easier thread safety
  (see: proj4.org/development/threads.html)
- Less transformations (no deformation, vertical grids, etc.)
- 2d only
- No pipeline operator: +proj=pipeline +step
- No parameters: +init +axis

Graphical Debugging extension for Visual Studio 2013, 2015 and 2017



`github.com/awulkiew/graphical-debugging`

Thanks!