



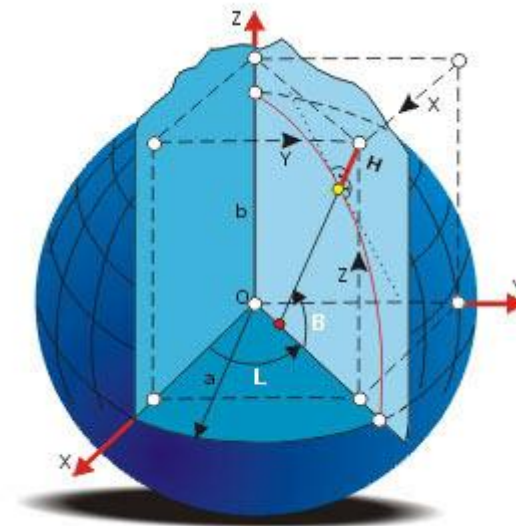
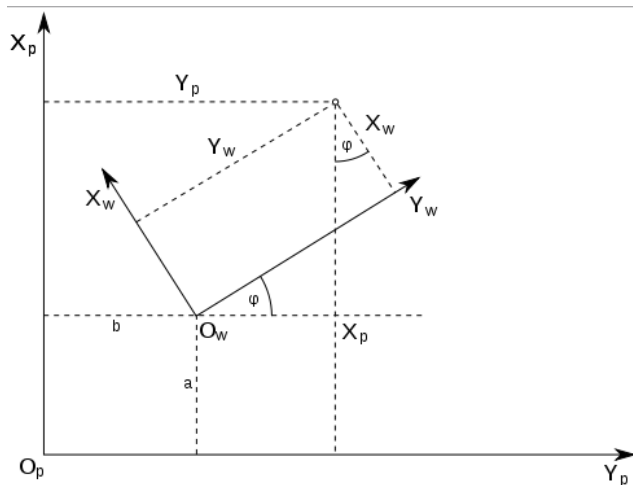
**AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE**

Wykorzystanie pakietu R w transformacji współrzędnych – package „vec2dtransf”, „proj4”

**mgr inż. Krzysztof Butryn
Wydział Geodezji Górniczej i Inżynierii Środowiska
Katedra Geomatyki**

Kraków, 21 grudnia 2015 r.

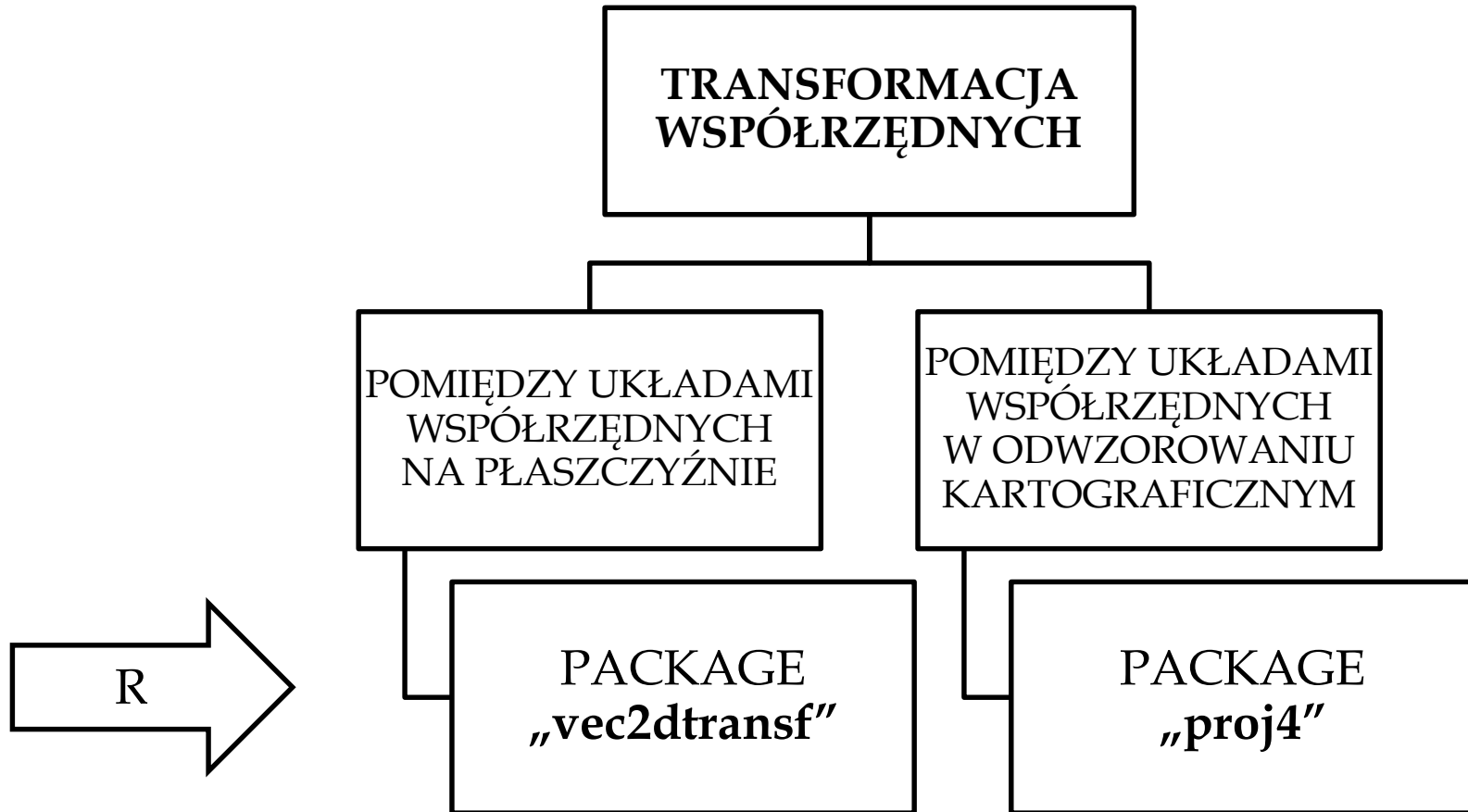
- Zagadnienie transformacji (przeliczania) współrzędnych pomiędzy różnymi układami to jeden z najczęściej spotykanych problemów praktyki geodezyjnej.
- Różne narzędzia i metodyka w zależności od natury danych.



Projekt inżynierski:

„Problematyka transformacji współrzędnych między układami „1965” i „2000” na obszarze Katowic”

Transformacja współrzędnych



Obszar zastosowań biblioteki:

Transformacja współrzędnych prostokątnych płaskich

- pakiet oferuje możliwość wykonania transformacji współrzędnych jako transformacji afinicznych oraz przez podobieństwo
- zastosowanie dla obiektów w przestrzeni wektorowej
- transformacja zadawana jest w oparciu o punkty, na podstawie których wyliczane są współczynniki lub wprost przez podanie współczynników transformacji
- w przypadku nadmiarowości danych wejściowych możliwa ocena dokładności procesu transformacji



AGH

Biblioteka vec2dtransf

Kiedy stosować? (wg autora biblioteki)

Podobieństwa i przekształcenia afiniczne są przydatne podczas integracji danych przestrzennych z różnych źródeł.

Istota – integracja dwóch zestawów „A” i „B”, gdzie zestaw „A” dostosowywany jest do zestawu „B”.

- Dla zbioru danych „A” brak danych dotyczących odwzorowania
- Zbiór „A” pierwotnie odwzorowany w układzie lokalnym, po migracji do WGS84 odstaje od zbioru „B”.
- Zbiór danych „A” ma dowolny układ współrzędnych, zintegrowany z układem krajowym
- Zbiór danych „A” został zdigitalizowany w oparciu o zniekształcony obraz, np. stara mapa
- Zbiór danych „A” został zdigitalizowany z błędnego georeferencyjnie zdjęcia satelitarne (OpenStreetMap).



AGH

Transformacja afiniczna

Transformacja afiniczna:

- umożliwia przesunięcie, obrót i zmianę skali (różną dla x i y)
- minimalna liczba punktów to 3
- zachowuje: równoległość linii i środki odcinków
- zmienia: długości odcinków i wartości kątów

$$x' = ax + by + c$$

$$y' = dx + ey + f$$

x', y' - współrzędne punktu w układzie wtórnym

x, y - współrzędne punktu w układzie pierwotnym

a, b, c, d - parametry obrotu

e, f - parametry przesunięcia o wektor

AffineTransformation

tworzenie obiektów klasy `AffineTransformation`. Procedura realizowana na dwa sposoby:

- **controlPoints** – parametry transformacji afinicznej wyznaczane w oparciu o punkty dostosowania posiadające współrzędne w układzie pierwotnym i wtórnym
Xsource, Ysource, Xtarget, Ytarget
- **parameters** – definicja transformacji poprzez zadanie wektora parametrów transformacji (a, b, c, d, e, f)
$$x' = ax + by + c$$
$$y' = dx + ey + f$$

Definicja przez „**controlPoints**” oraz przez „**parameters**” – stosowane wymiennie

```
at <- AffineTransformation(parameters=c(1,2,3,4,5,6))
```

AffineTransformation jako klasa:

tworzenie obiektów klasy `AffineTransformation`. Procedura realizowana na dwa sposoby:

- **controlPoints** - obiekt klasy „`data.frame`” zawierający współrzędne punktów dostosowania zorganizowane w czterech kolumnach (jak wcześniej)
- **parameters** - obiekt klasy „`numeric`” zawierający parametry transformacji afinicznej (wektor 6-elementowy)
- **residuals** - występuje jedynie przy nadmiarowej liczbie punktów dostosowania i zastosowaniu MNK (różnice pomiędzy przetransformowanymi współrzędnymi źródłowymi a współrzędnymi docelowymi)
- **rmse** - pierwiastek błędu średniokwadratowego wpasowania (charakterystyka dokładnościowa)



AGH

Transformacja przez podobieństwo

Transformacja przez podobieństwo:

- umożliwia przesunięcie obiektu o wektor
- umożliwia obrót obiektu o kąt
- umożliwia rozciągnięcie układu pierwotnego względem wtórnego w kierunkach osi x i y przy założeniu jednakowej skali
- minimalna liczba punktów to 2 (cztery parametry transformacji)

$$x' = ax + by + c$$

$$y' = ay - bx + d$$

x', y' - współrzędne punktu w układzie wtórnym

x, y - współrzędne punktu w układzie pierwotnym

a, b, c, d - parametry transformacji

`SimilarityTransformation`

tworzenie obiektów klasy `SimilarityTransformation`. Procedura realizowana na dwa sposoby:

- **`controlPoints`** – parametry transformacji afinicznej wyznaczone w oparciu o punkty dostosowania posiadające współrzędne w układzie pierwotnym i wtórnym
`Xsource`, `Ysource`, `Xtarget`, `Ytarget`
- **`parameters`** – definicja transformacji poprzez zadanie wektora parametrów transformacji (a, b, c, d)
$$x' = ax + by + c$$
$$y' = ay - bx + d$$

Definicja przez „**`controlPoints`**” oraz przez „**`parameters`**” – stosowane wymiennie

```
st <- SimilarityTransformation(parameters=c(1,2,3,4))
```

`SimilarityTransformation` jako klasa:

tworzenie obiektów klasy `AffineTransformation`. Procedura realizowana na dwa sposoby:

- **`controlPoints`** - obiekt klasy „`data.frame`” zawierający współrzędne punktów dostosowania zorganizowane w czterech kolumnach (jak wcześniej)
- **`parameters`** - obiekt klasy „`numeric`” zawierający parametry transformacji afinicznej (wektor 4-elementowy)
- **`residuals`** - występuje jedynie przy nadmiarowej liczbie punktów dostosowania i zastosowaniu MNK (różnice pomiędzy przetransformowanymi współrzędnymi źródłowymi a współrzędnymi docelowymi)
- **`rmse`** - pierwiastek błędu średniokwadratowego wpasowania (charakterystyka dokładnościowa)

W przypadku realizacji transformacji na podstawie punktów dostosowania, należy uprzednio obliczyć parametry transformacji za pomocą metody **`calculateParameters`**, przykładowo:

`calculateParameters`

`signature (object=„AffineTransformation“)`

`calculateParameters`

`signature (object=„SimilarityTransformation“)`

Po zdefiniowaniu klasy transformacji zaleca się jej podgląd:

`showClass ("AffineTransformation")`

`showClass ("SimilarityTransformation")`

getParameters

wyświetlenie uprzednio wyznaczonych parametrów transformacji

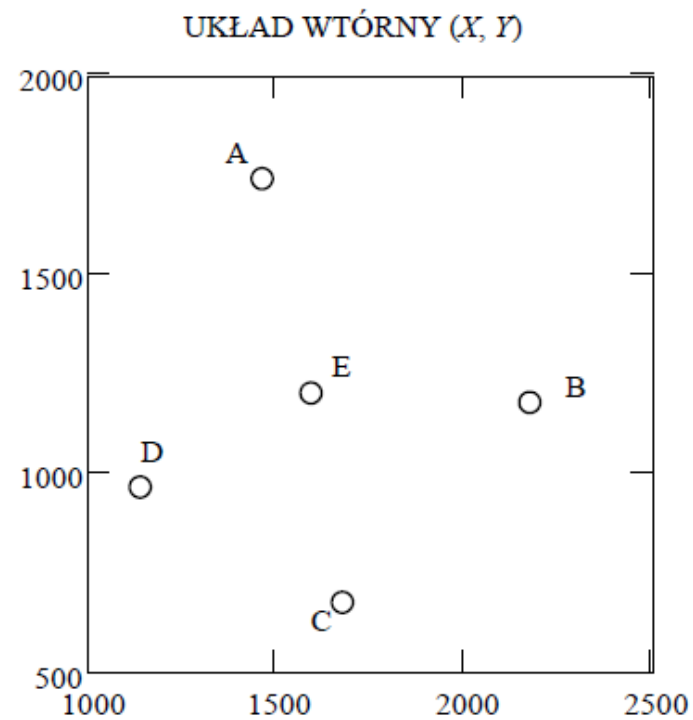
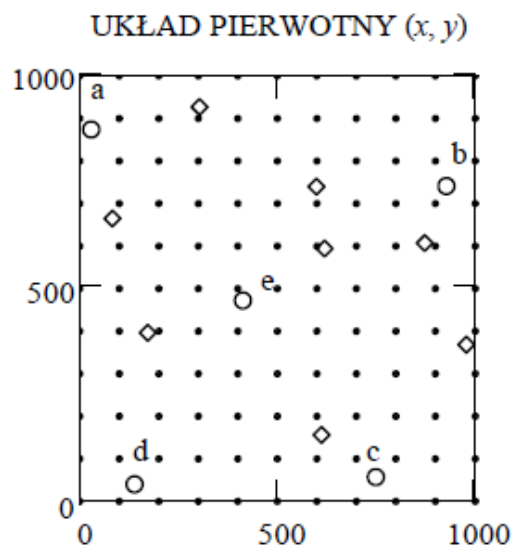
getResiduals

wyświetlenie obliczonych odchyłek (pod warunkiem, że do obliczeń przyjęto nadliczbowe punkty dostosowania)

getRMSE

wyświetlenie charakterystyki dokładnościowej pod postacią obliczonego pierwiastka błędu średniokwadratowego (pod warunkiem, że do obliczeń przyjęto nadliczbowe punkty dostosowania)

Przykład 1 - transformacja regularnej siatki





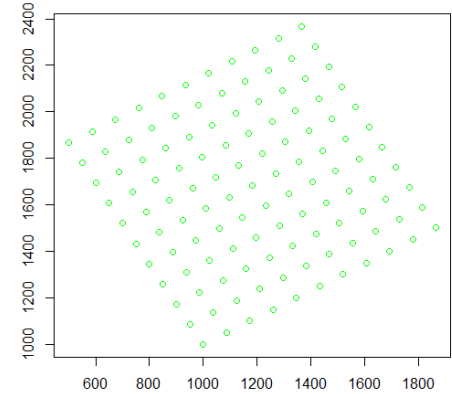
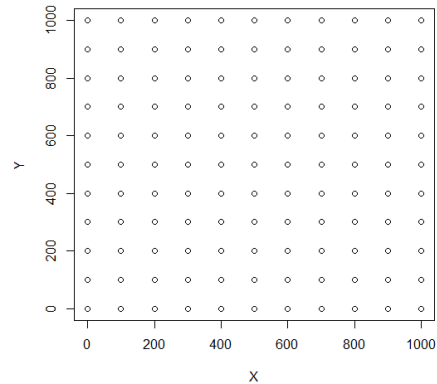
AGH

Przykład 1 - transformacja regularnej siatki

```

> pierw=read.table("pierw.txt")
> plot(pierw)
> names(pierw)=c("X", "Y")
> plot(pierw)
> library(vec2dtransf)
> dost=read.table("wtor.txt")
> dost
> st=SimilarityTransformation(dost)
> calculateParameters(st)
> getParameters(st)
> S <- SpatialPoints(pierw)
> trS <- applyTransformation(st,S)
> plotGridTransformation(st, bbox(S), 120)

```



```

1 873.91 30.00 1742.198 1463.317
2 740.61 927.12 1178.366 2173.414
3 56.25 749.02 674.665 1677.178
4 40.00 139.02 965.374 1140.807
5 471.20 412.83 1201.102 1593.498
> at=AffineTransformation(dost)
> calculateParameters(at)
> getParameters(at)

```

	a	b	c	d	e	f
	0.8661010	-0.4996867	1000.0372432	0.4998751	0.8658576	1000.4807568

```

> getResiduals(at)

```

	X_Target	Y_Target
1	0.2570749	0.01469282
2	0.1552386	-0.03313348
3	0.1848892	0.03460843
4	0.1591599	-0.04028365
5	-0.7563626	0.02411589

```

> getRMSE(at)
[1] 0.381183

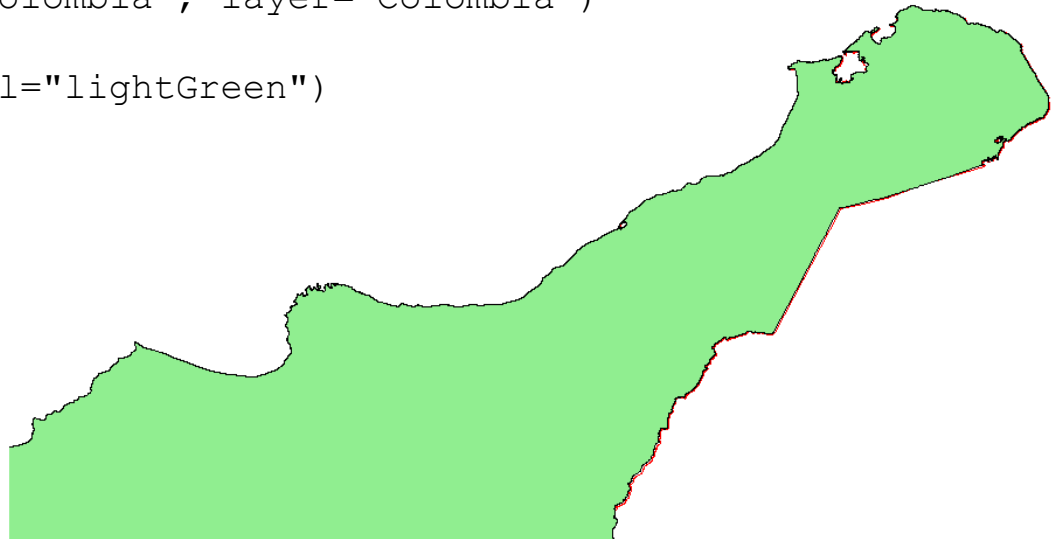
```

Przykład 2 - fragment wybrzeża Ameryki Płd.

Zbiór danych „A”: linia brzegowa (obiekt liniowy)

Zbiór danych „B”: fragment poligonu pokrywający się zasięgiem z w/w linią (obiekt poligonowy)

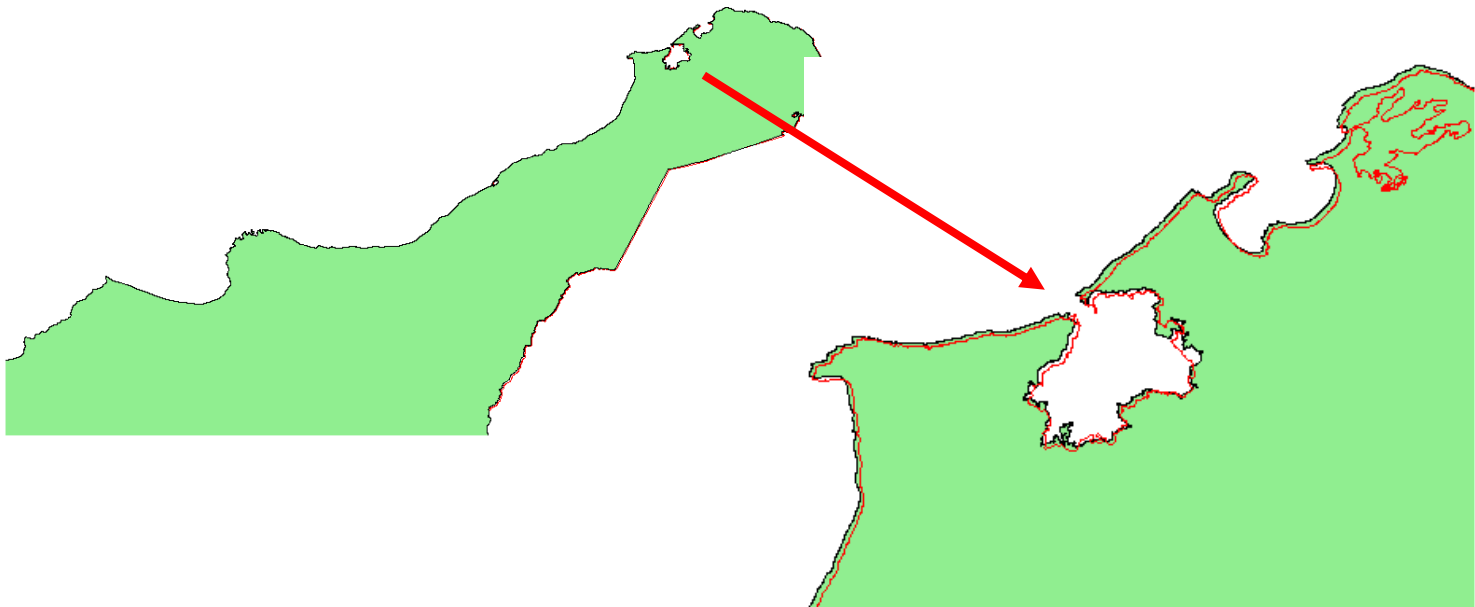
```
>require("rgdal")  
>l = readOGR(dsn="la_guajira", layer="La_Guajira")  
>r = readOGR(dsn="colombia", layer="Colombia")  
>plot(l,col="red")  
>plot(r,add=TRUE,col="lightGreen")
```



Przykład 2 - fragment wybrzeża Ameryki Płd.

Przybliżenie do wybranego fragmentu wybrzeża:

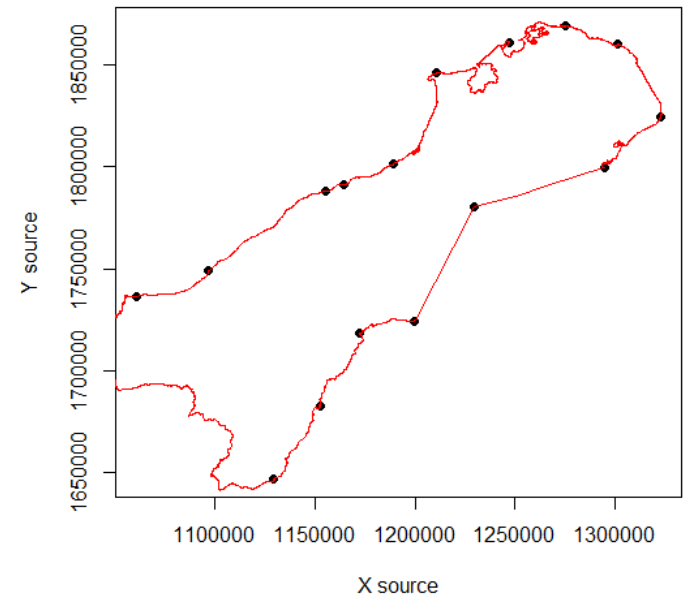
```
plot(r,col="lightGreen",xlim=c(1206309,1264449),ylim=c(1832839,1864827))  
plot(l,col="red",add=TRUE)
```





Przykład 2 - fragment wybrzeża Ameryki Płd.

```
> require("vec2dtransf")
> data(control.points)
> options(digits=10)
> control.points
  ID   X source   Y source   X target   Y target
1  21 1155435.738 1788033.340 1155226.925 1788399.626
2  20 1172614.978 1718170.440 1172166.670 1718421.167
3  19 1294237.492 1799488.935 1293812.519 1799926.624
4  17 1096966.295 1748773.531 1096550.401 1749046.362
5  15 1189372.949 1801408.533 1188968.824 1801745.270
6  13 1152591.958 1682719.038 1152198.231 1683076.117
7  11 1300854.445 1859903.983 1300481.740 1860215.505
8  10 1274769.607 1868758.771 1274382.705 1869069.549
9   8 1246851.546 1860836.296 1245897.667 1861410.277
10  7 1229707.294 1780574.284 1229232.915 1780619.843
11  6 1199565.945 1724303.607 1199216.274 1724586.285
12  5 1164257.772 1790905.328 1163404.642 1791332.734
13  4 1210479.122 1846091.924 1210049.768 1846299.716
14  3 1061465.218 1736328.976 1061151.587 1736734.430
15  2 1129726.749 1647315.999 1128964.546 1647933.372
16  1 1322531.689 1824367.779 1322119.595 1824669.653
> plot(control.points[2:3],pch=19)
> plot(l,add=TRUE,col="red")
```



Przykład 2 - fragment wybrzeża Ameryki Płd.

```

aft = AffineTransformation(control.points[2:5])
calculateParameters(aft)
getParameters(aft)
getResiduals(aft)
getRMSE(aft)
plotGridTransformation(aft, bbox(1), 100)
plot(1, add=TRUE, col="red")

```

```

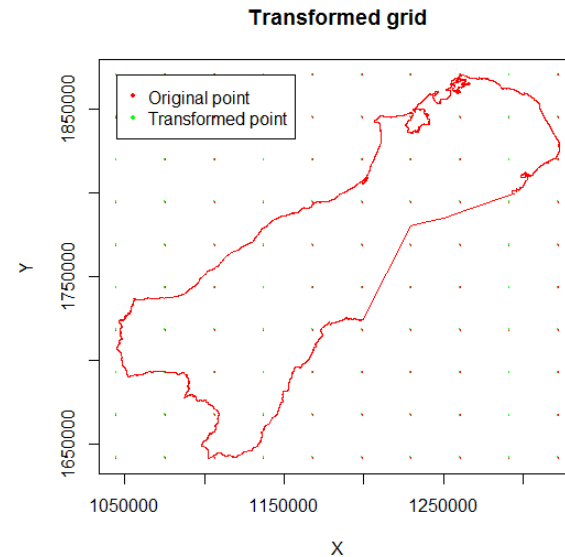
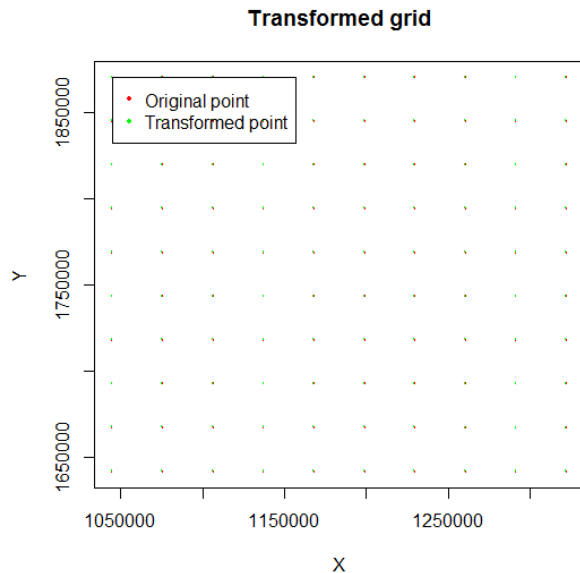
> getParameters(aft)
      a              b              c              d
9.999615493e-01 -2.747970753e-05 -3.801820475e+02 -1.177985405e-04
      e              f
9.997213391e-01  9.814604351e+02

```

```

> getRMSE(aft)
[1] 234.1379378

```



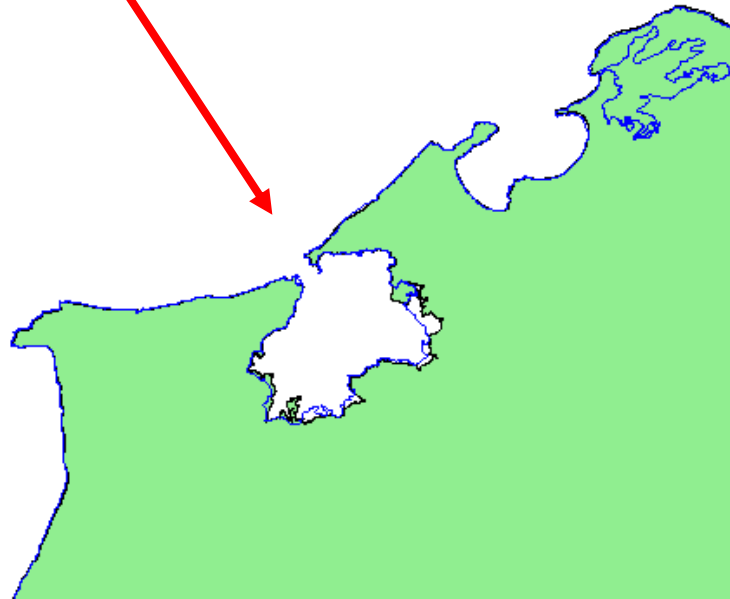
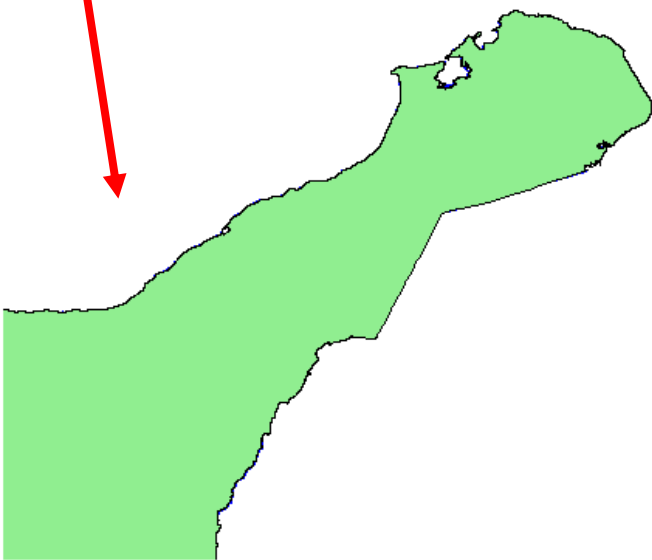


AGH

Przykład 2 - fragment wybrzeża Ameryki Płd.

```
newLines = applyTransformation(aft,1)  
plot(newLines,col="blue,,")  
plot(r,add=TRUE,col="lightGreen")
```

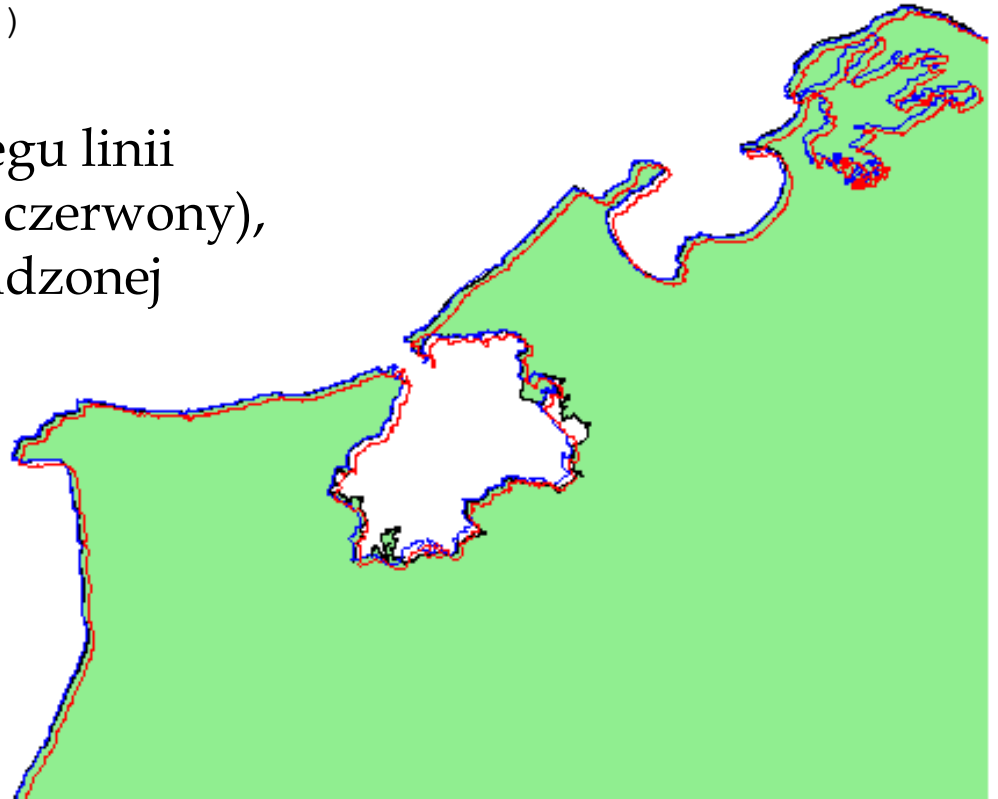
```
plot(r,col="lightGreen",xlim=c(1206309,1264449),ylim=c(1832839,1864827))  
plot(newLines,col="blue",add=TRUE)
```



Przykład 2 – fragment wybrzeża Ameryki Płd.

```
plot(1, col="red", add=TRUE)
```

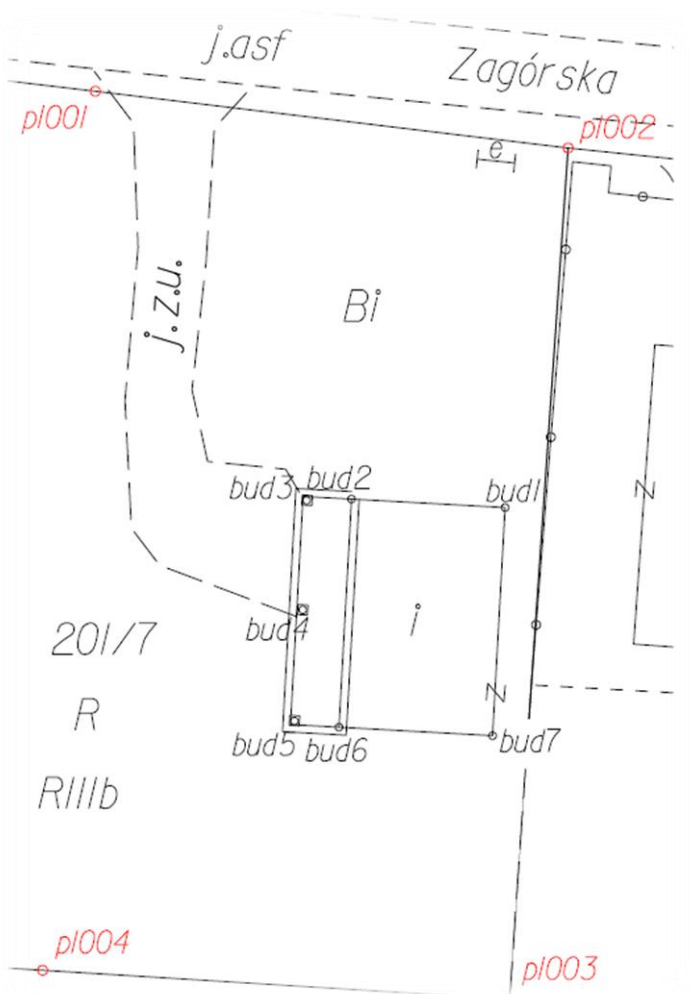
Wynik: porównanie przebiegu linii
sprzed dopasowania (kolor czerwony),
do przebiegu po przeprowadzonej
transformacji.





AGH

Przykład 3 – geodezja codzienna



Dla potrzeb tyczenia obrysu budynku pod wykop. Transformacja z układu lokalnego do układu wsp. „2000” – wprowadzenie współrzędnych do odbiornika RTK GPS.

p1001	136.18	83.51
p1002	133.92	102.33
p1003	100.00	100.00
p1004	101.16	81.41
bud1	119.59	99.82
bud2	119.93	93.71
bud3	119.88	91.92
bud4	115.52	91.76
bud5	111.09	91.44
bud6	110.84	93.21
bud7	110.50	99.32



p1001	5535544.68	7420579.10
p1002	5535542.44	7420597.93
p1003	5535508.51	7420595.64
p1004	5535509.65	7420577.04



bud1	5535528.10	7420595.43
bud2	5535528.44	7420589.32
bud3	5535528.38	7420587.53
bud4	5535524.02	7420587.38
bud5	5535519.59	7420587.06
bud6	5535519.34	7420588.83
bud7	5535519.01	7420594.94

Przeliczanie współrzędnych pomiędzy układami - package „proj4”

Czym jest „proj4”?

Jest to prosty interfejs stworzony dla potrzeb transformacji współrzędnych geograficznych pomiędzy różnymi układami odniesienia.

Cel powstania

Powstała dla potrzeb obsługi popularnej biblioteki odwzorowań kartograficznych o nazwie **PROJ.4**.

Twórca biblioteki

Simon Urbanek, programista AT&T Research Labs

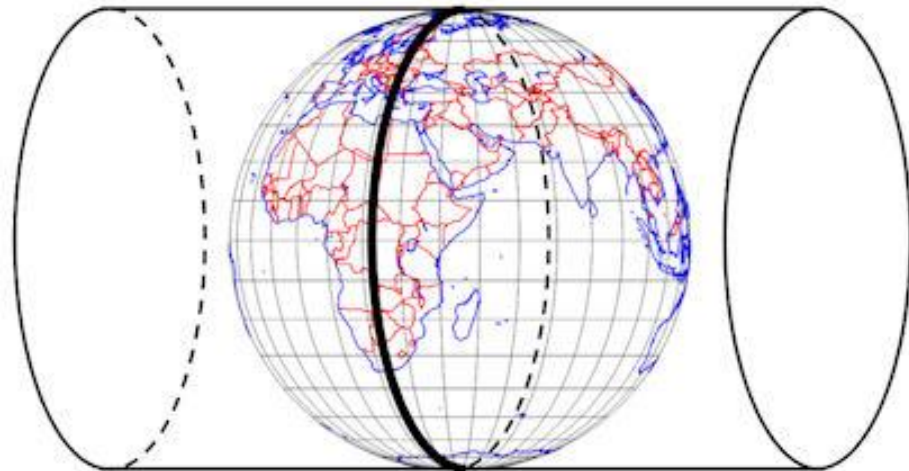
PROJ.4 (lub proj) – biblioteka dedykowana do wykonywania konwersji (przeliczeń) pomiędzy odwzorowaniami kartograficznymi.

Na stronie: http://www.remotesensing.org/geotiff/proj_list/

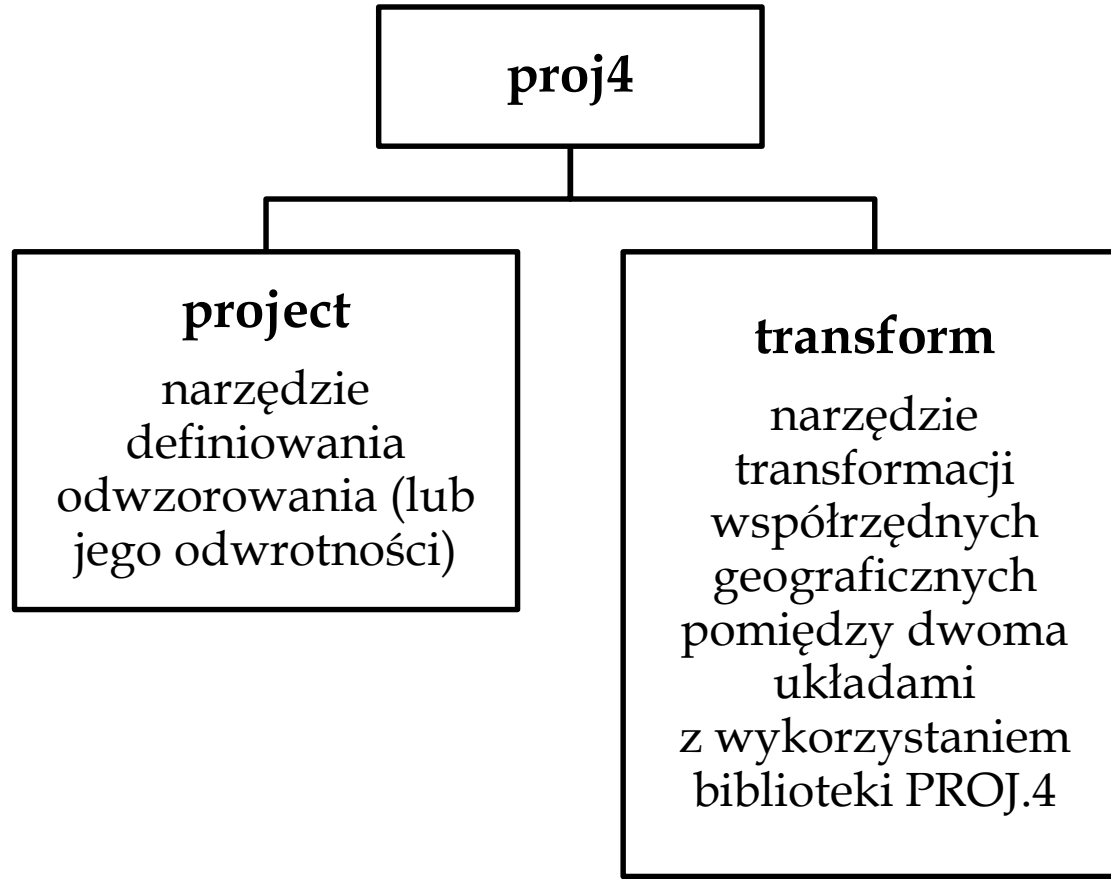
Lista najczęściej wykorzystywanych odwzorowań:

Projections Transform List

- [Albers Equal-Area Conic](#)
- [Azimuthal Equidistant](#)
- [Cassini-Soldner](#)
- [Cylindrical Equal Area](#)
- [Eckert IV](#)
- [Eckert VI](#)
- [Equidistant Conic](#)
- [Equidistant Cylindrical](#)
- [Equirectangular](#)
- [Gauss-Kruger](#)
- [Gall Stereographic](#)
- [GEOS - Geostationary Satellite View](#)
- [Gnomonic](#)
- [Hotine Oblique Mercator](#)
- [Krovak](#)
- [Laborde Oblique Mercator](#)
- [Lambert Azimuthal Equal Area](#)
- [Lambert Conic Conformal \(1SP\)](#)
- [Lambert Conic Conformal \(2SP\)](#)
- [Lambert Conic Conformal \(2SP Belgium\)](#)
- [Lambert Cylindrical Equal Area](#)



Przeliczanie współrzędnych pomiędzy układami - package „proj4”





AGH

project / ptransform

```
project(xy, proj, inverse = FALSE, degrees = TRUE, silent = FALSE,  
        ellps.default="sphere")
```

xy - dane wejściowe (lista, wektor lub macierz)

proj - definicja odwzorowania

inverse - domyślnie współrzędne geogr. rzutowane na siatkę kartograficzną

degrees - dla TRUE kąty podane w stopniach, dla FALSE w radianach

silent - dla TRUE wyłączone ostrzeżenia

ellps.default - domyślna elipsoida, chyba że proj zawiera inne ustalenia

```
ptransform(data, src.proj, dst.proj, silent=TRUE)
```

xy - dane wejściowe (lista, wektor lub macierz)

src.proj - definicja odwzorowania wejściowego (źródłowego)

dst.proj - definicja odwzorowania docelowego (wtórnego)

silent - dla TRUE wyłączone ostrzeżenia



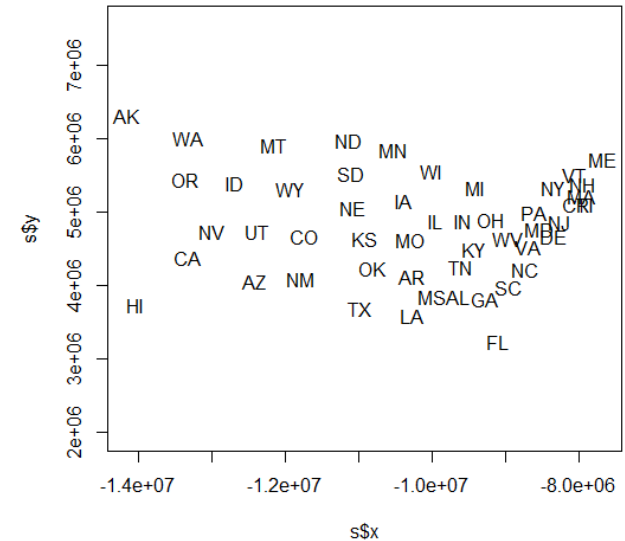
AGH

Przykłady

Wykorzystanie „project”:

```
> data(state)
> s <- project(state.center, "+proj=merc")
> plot(s, type='n', asp=1)
> text(s,, state.abb)
```

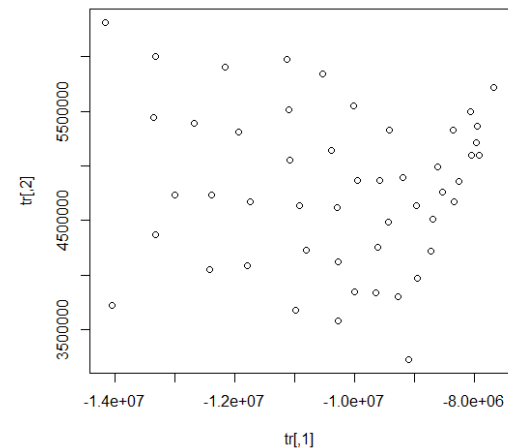
Początki nazw stanów wyświetlone w środkach odpowiednich Stanów, odwzorowanie Mercatora.



Wykorzystanie „transform”:

```
> data(state)
> sc <- cbind(state.center$x, state.center$y)
> tr <- ptransform(sc/180*pi, '+proj=latlong
+ellps=sphere', '+proj=merc +ellps=sphere')
> plot(tr)
```

Zmiana jednostek; wyświetlane środki punktów odpowiednich Stanów.



Wykaz źródeł

- <https://cran.r-project.org/web/packages/proj4/proj4.pdf>
- <https://cran.r-project.org/web/packages/vec2dtransf/vec2dtransf.pdf>
- Iliffe, J. and Lott, R. Datums and map projections: For remote sensing, GIS and surveying. Section 4.5. pp.109-117,135-137, 2008.
- Osada R., Transformacje, georeferencja i rektyfikacja map i obrazów, Wydawnictwo Naukowe DSW, Wrocław 2009
- UC Davis Soil Resource Laboratory. Case Study: Fixing Bad TIGER Line data with R and PostGIS. 2007.
<URL: <http://casoilresource.lawr.ucdavis.edu/drupal/node/433>>
- Knippers, R. 2D Cartesian coordinate transformations. Section 5.4. 2009.
<URL: <http://kartoweb.itc.nl/geometrics/Coordinate%20transformations/coordtrans.html>>