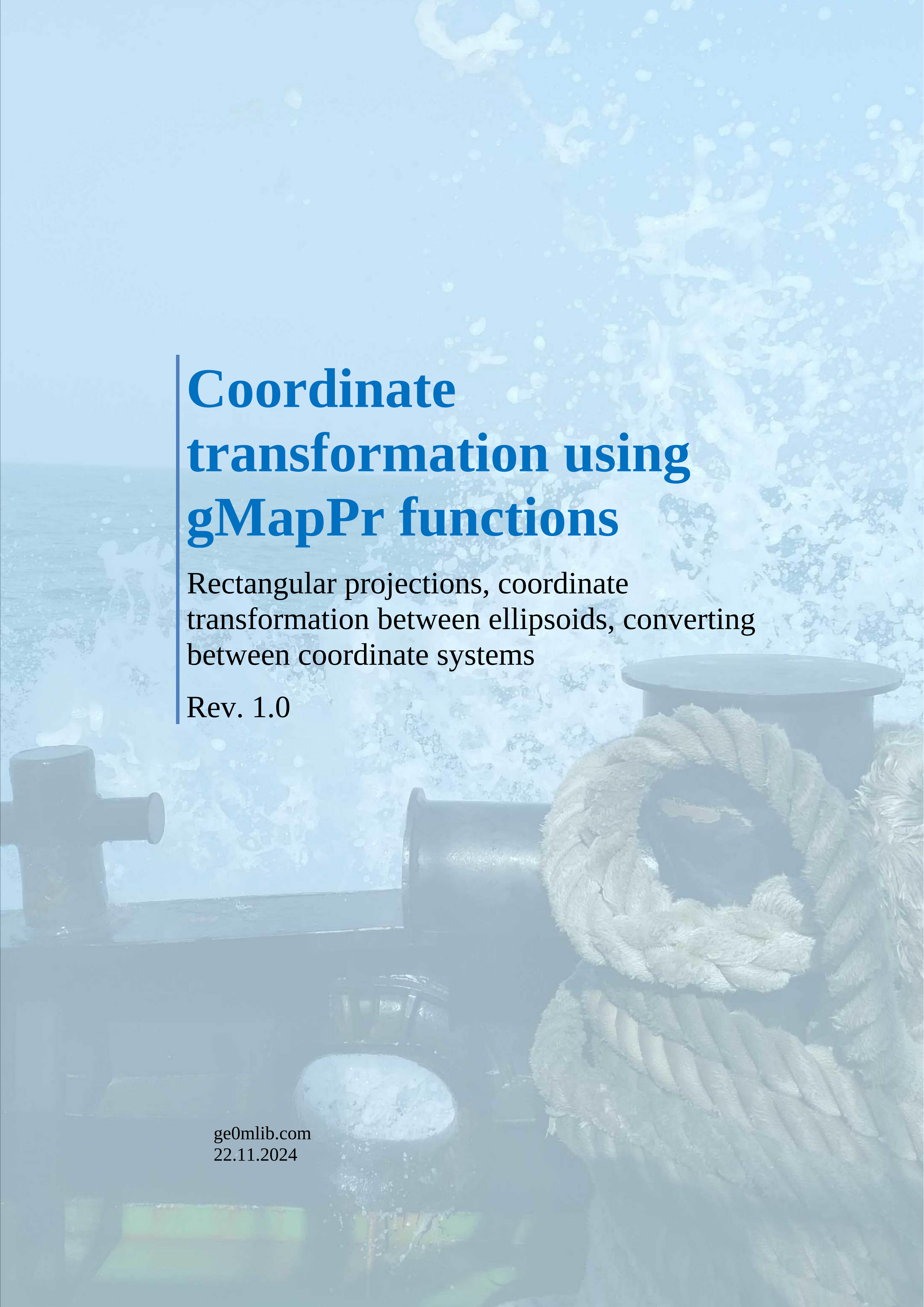

Coordinate transformation using gMapPr functions

Rectangular projections, coordinate transformation between ellipsoids, converting between coordinate systems

Rev. 1.0

ge0mlib.com
22.11.2024

Contents

Revision History

Version	Date	Revision description
1.0	2024-11-22	Created the document // Ivan D.

List of Figures

List of Tables

Introduction

The gMapPr library is designed for coordinate transformations such as:

-- converting between different coordinate systems (geographic, rectangular, topocentric, geocentric);

-- calculations for 24 rectangular projections (Azimuthal, Mercator, etc.), each of them may include several different types of input parameters;

-- transformation between ellipsoids (7-parameter Helmert transformation, polynomial transformations, etc.).

The library aims to perform coordinate transformations when working with seg-y, jsf, xtf, etc. files directly in Matlab, avoiding the use of external geographic information systems or geodetic calculators. The functions are written based on the Guidance Note 7-2 of International Association of Oil & Gas Producers: “IOGP Report 373-07-02, EPSG Guidance Note number 7, part 2, July 2024”. The functions have been tested in MatLab 2018a and GNU Octave 9.2; the functions are the part of the ge0mlib library.

The functions have been verified on the examples given in the “IOGP Report 373-07-02, EPSG Guidance Note number 7, part 2, July 2024”; extended function testing was not performed. For support and bug reports, use mail@ge0mlib.com, with gMapPr in the subject line.

1. General description of the gMapPr functions

The following types of data and parameters are used when working with the library (*Figure 1.1*):

-- **«Horizontal» coordinates**. Depending on the current coordinate system, they can be represented by latitude-longitude, rectangular coordinates (E,N), topocentric or geocentric (X,Y) coordinates;

-- **«Vertical» coordinate**. Depending on the current coordinate system, it can be represented by ellipsoidal height (h), geocentric “height” (Z) or topocentric height (W);

-- Date, time, distance (on the ellipsoid or plane). The current version of the library does not use these data types;

-- **Ellipsoid parameters** (EllipParam variable). In all the functions they are specified as the length of the semi-major axis and eccentricity;

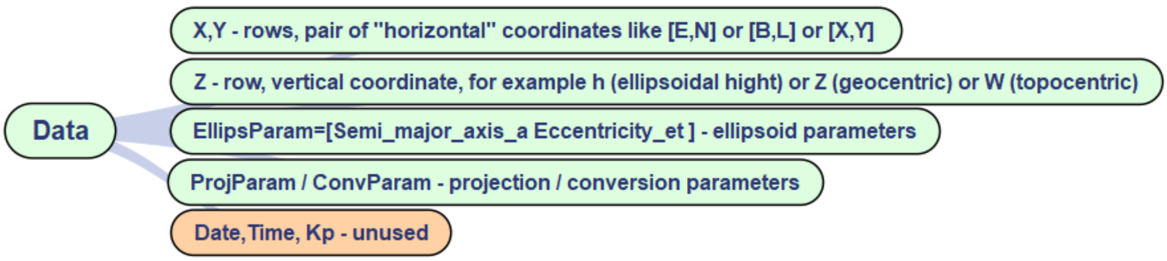

Figure 1.1 Data and parameter types used in gMapPr

-- **Projection parameters** (ProjParam variable). Used when converting between different types of coordinate systems (geographic, rectangular, topocentric); in particular, to convert between geographic and rectangular coordinates;

-- **Coordinate transformation/conversion parameters** (ConvParam variable). Used when converting between ellipsoids and different types of coordinate systems.

The following data dimensions are used when working with the library:

-- **meters** for (E,N), (X,Y) coordinates and distance;

-- **degrees** for Latitude-Longitude and direction;

-- **seconds** for time (the current version of the library does not use the time value).

The functions performing coordinate transformations and their corresponding datasets (coordinates) are shown in ***Figure 1.2***: each of the two columns corresponds to a different ellipsoid; blocks of dark green color describe datasets; blocks of light green color contain function names.

Within a single ellipsoid, one can perform converting between the following types of coordinate systems: **engineering rectangular** (axes rotated and shifted about the E, N axes), **rectangular** (E,N), **geographic** (B,L), **geocentric** (X,Y,Z), **topocentric** (U,V,W). To convert between rectangular and geographical systems (EN&BL block in ***Figure 1.2***), library contains 47 functions describing different types of rectangular projections (***Figure 1.3***, ***Table 3.1***).

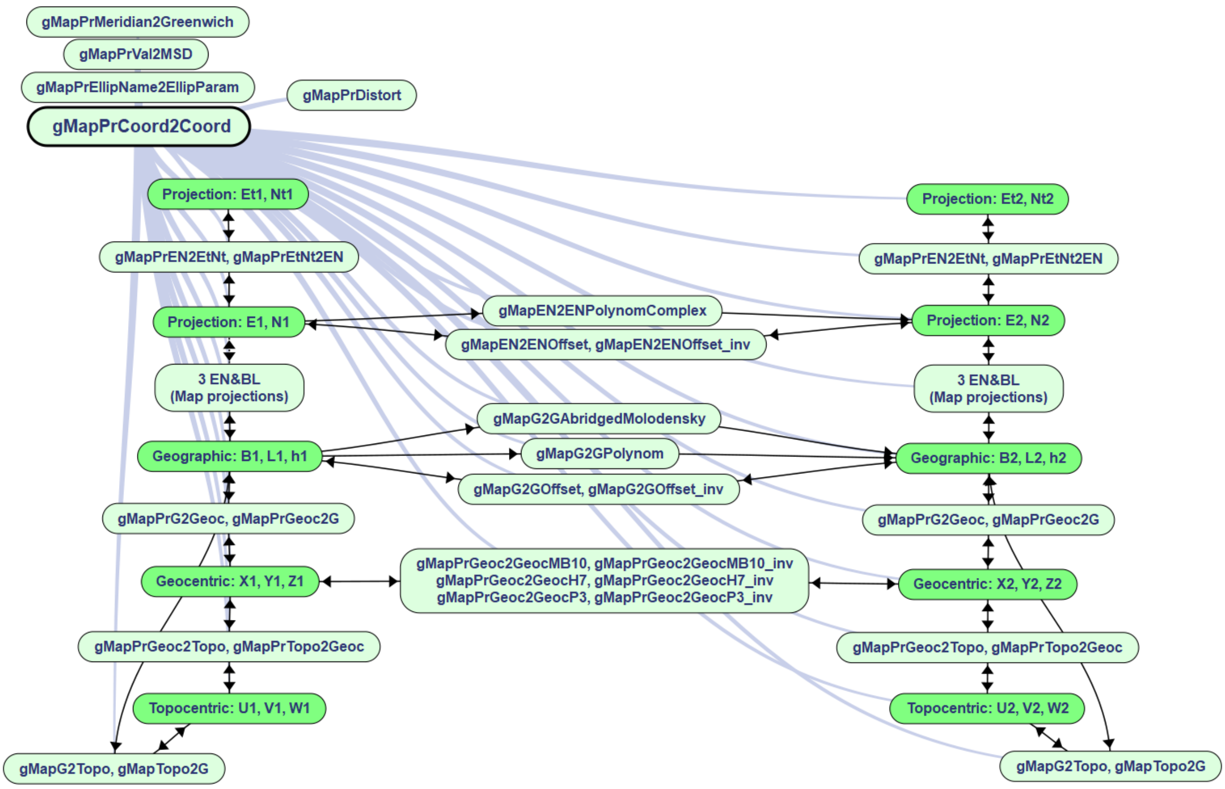

Figure 1.2 The library functions for coordinate transformation and data types

One can perform a transformation between two ellipsoids by converting between geocentric coordinate systems (the 7-parameter Helmert, 3-parameter, or 10-parameter Molodensky-Badekas transformations). The library also contains the functions to convert between geographic coordinates of two ellipsoids and convert between rectangular coordinates of two ellipsoids (blocks in the center part of ***Figure 1.2***, ***Table 3.2***).

The specific function group is shown in the left upper part of ***Figure 1.2*** and labeled as the "Main" group in ***Figure 1.3***. **The primary function for working with coordinates is gMapPrCoord2Coord**; the remaining functions are considered "auxiliary" and not directly related to coordinate transformation.

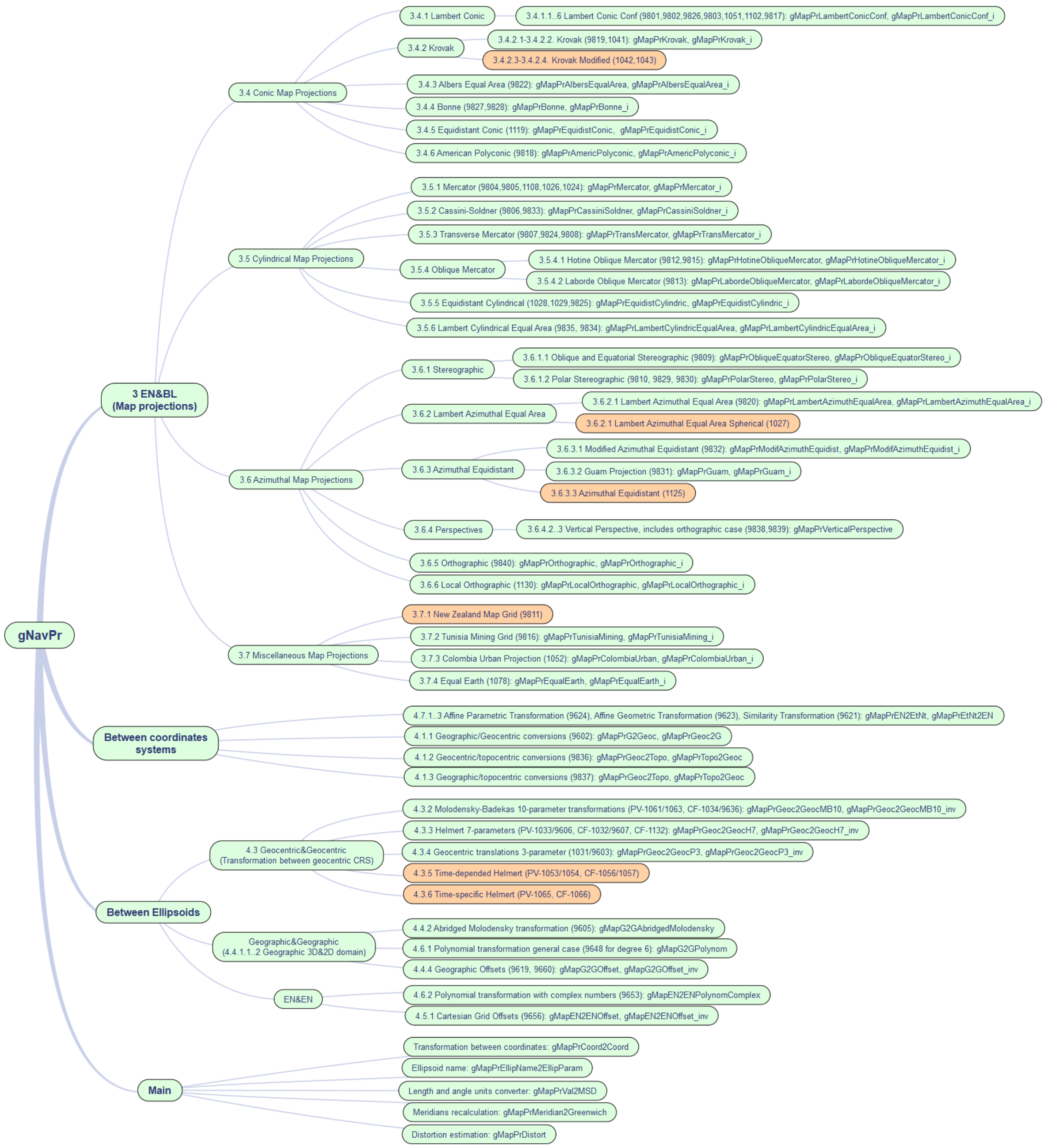

Figure 1.3 The gMapPr library functions (the blocks are signed with the chapter numbers from the "IOGP Report 373-07-02, EPSG Guidance Note number 7, part 2, July 2024")

2. Description of the “Main” group functions

2.1 gMapPrCoord2Coord function

Each gMapPr function performs a single coordinate transformation. For example, if one needs to convert geographic latitude and longitude to rectangular UTM projection coordinates, the following commands must be executed in MatLab:

```
>> EllipParam=[6378137 0.081819190842];ProjParam=[0 142 0.9996 500000 0 1];
>> [GpsE,GpsN]=gMapPrTransMercator(51:.1:52,142:.1:143,GpsEllipParam,GpsProjParam);
```

Where:

EllipParam – semi-major axis and eccentricity of the ellipsoid;

ProjParam – UTM projection parameters;

gMapPrTransMercator – function for converting latitude and longitude (51:.1:52,142:.1:143) to UTM rectangular projection coordinates.

Consider a more complex transformation: convert geographic coordinates on the first ellipsoid into engineering rectangular coordinates on the second one. According to the available geodetic datum, the following sequence of transformations should be performed in order to do this:

1) convert geographic coordinates to geocentric coordinates on the ellipsoid 1;
2) convert from ellipsoid 1 to ellipsoid 2 using the 7-parameter Helmert transformation;
3) convert geocentric coordinates on ellipsoid 2 to geographic coordinates;
4) convert geographic coordinates to rectangular UTM projection coordinates;
5) convert to engineering rectangular coordinates (shift and rotation of rectangular coordinates).

The aforementioned sequence would necessitate the creation of a separate MatLab script or function comprising five transformations. In order to avoid designing separate functions for each “composite” transformation, the gMapPrCoord2Coord function was developed. It consists of a cell array which has the sequence of transformations represented as a list of "functions for coordinate recalculation" and "recalculation parameters". The aforementioned sequence of transformations is implemented via the following code:

```
%cell{1}: geographic >> geocentric_1
>>NavD{1}={'gMapPrG2Geoc',[6378245.0 0.08181333]};
% cell{2}: geocentric_1 >> geocentric_2
>> NavD{2}={'gMapPrGeoc2GeocH7_inv',[23.57 -140.95 -79.8 0.000 -0.35 -0.79 -0.22e-6 1]};
% cell{3}: geocentric_2 >> geographic
>> NavD{3}={'gMapPrGeoc2G',[6378137 0.081819190842]};
% cell{4}: geographic >> E,N (UTM)
>> NavD{4}={'gMapPrTransMercator',[6378137 0.081819190842],[0 142 0.9996 500000 0 1]};
% cell{5}: E,N >> Et,Nt
>> NavD{5}={'gMapPrEN2EtNt',[1.00000155 0.8333333333333 -129.549 -208.185 3]};
%Calculations
```

```
>> [X,Y,Z]=gMapPrCoord2Coord(51:.1:52,142:.1:143,0,NavD);
```

The transformation steps are carried out in accordance with the transitions between adjacent blocks in ***Figure 1.2***. In fact, ***Figure 1.2*** is a flowchart that enables the construction of a command sequence to convert from any coordinate system on the initial ellipsoid to any coordinate system on the subsequent one and back. The list of functions for transforming from a rectangular system to a geographic system and back is not included in ***Figure 1.2***. Instead, these functions are presented in ***Figure 1.3*** and ***Table 3.1***.

2.2 gMapPrEllipName2EllipParam

The auxiliary function that enables the ellipsoid name or EPSG code to be employed instead of the primary semi-major axis and eccentricity values. For example:

```
>> ProjParam=[0 142 0.9996 500000 0 1];
>> [GpsE,GpsN]=gMapPrTransMercator(51:.1:52,142:.1:143,'WGS 84',GpsProjParam);
>> [GpsE,GpsN]=gMapPrTransMercator(51:.1:52,142:.1:143,'7030',GpsProjParam);
```

2.3 gMapPrVal2MSD

The auxiliary function that enables the values of different dimensions to be converted into meters-degrees-seconds, employing the appropriate EPSG code. For example:

```
>> a=gMapPrVal2MSD([1260 852],'1025');
>> b=gMapPrVal2MSD([1.260 0.852],'1025i');
>> a=gMapPrVal2MSD([126 85],'Clarke''s chain');
>>b=gMapPrVal2MSD([2534.69405952 1709.9126592],'9038i');
```

In the case of a "reverse conversion" from degrees-meters-seconds, the letter “i” is added to the EPSG code.

2.4 gMapPrMeridian2Greenwich

The auxiliary function that enables the longitude values to be converted from different zero meridians to the Greenwich meridian. The value of the current meridian is given to the function using the EPSG code or meridian designation. For example:

```
>> a=gMapPrMeridian2Greenwich(12.6,'Bern');
>> a=gMapPrMeridian2Greenwich(20.0395833333333333,'8907i');
```

In the case of a "reverse conversion" the letter “i” is added to the EPSG code.

2.5 gMapPrDistort

The auxiliary function that enables the calculation of the change of scale and rotation of coordinate axes for coordinate systems, as well as the change of height for coordinates specified at the input and output of the gMapPrCoord2Coord function.

The calculations are performed by setting the increments [dx, dy] to the coordinates [X, Y], with subsequent calculations for four points using the gMapPrCoord2Coord function. The results of the calculations are employed to estimate the scale and azimuth for the X+dx and Y+dy segments obtained in the new coordinate system subsequent to the aforementioned transformations.

Description of the gMapPrDistort function:

function [mx,my,angx,angy,dh]=gMapPrDistort(X,Y,Z,NavD,dd,key),where

X,Y - rows, a pair of the horizontal coordinates, for example [E,N] or [B,L] or [X,Y];

Z - row, the vertical coordinate, for example h (ellipsoidal hight) or Z (geocentric) or W (topocentric);

NavD - cell array with the function names and conversion parameters as for the function call;

dd=[dd1 dd2] - steps along X and Y axes;

mx,my - scale; multiple coefficients for X and Y axes after transformation;

angx,angy - rotation angles for X and Y axes after transformation;

dh - Z-value changes after transformation (calculated for central point);

key - target axis "swap" against the source one: 0 - not inverted BL>>BL or EN>>EN; 1 - inverted BL>>EN or EN>>BL.

The key is introduced because in the rectangular system the [X,Y] axes correspond to [E(**horizontal** axis),N(vertical axis)], whereas in the geographic system the axes are "swapped" with [X,Y] corresponding to [B(**vertical** axis),L(horizontal axis)]. Consequently, if key=1 is not used, the rotation angle of the axes will be augmented by ±90 degrees when converting from the geographic system to the rectangular system (and back).

3. List of the functions for coordinate transformation

The list of functions, accompanied by EPSG codes, for converting geographic coordinates to rectangular projections and back is provided in ***Table 3.1***. The list of functions for converting between different coordinate systems is provided in ***Table 3.2***. The list of functions for converting between coordinates on different ellipsoids is provided in ***Table 3.3***.

Table 3.1 Rectangular projections

Function name	Comments
gMapPrAlbersEqualArea gMapPrAlbersEqualArea_i	Albers Equal Area projection. EPSG code 9822.
gMapPrAmericPolyconic gMapPrAmericPolyconic_i	American Polyconic projection. EPSG code 9818.
gMapPrBonne gMapPrBonne_i	Bonne projection. EPSG codes 9827, 9828.
gMapPrCassiniSoldner gMapPrCassiniSoldner_i	Cassini-Soldner projection. EPSG codes 9806, 9833.
gMapPrColombiaUrban gMapPrColombiaUrban_i	Colombia Urban projection. EPSG code 1052.
gMapPrEqualEarth gMapPrEqualEarth_i	Equal Earth projection. EPSG code 1078.
gMapPrEquidistConic gMapPrEquidistConic_i	Equidistant Conic projection. EPSG code 1119.
gMapPrEquidistCylindric gMapPrEquidistCylindric_i	Equidistant Cylindrical projection. EPSG codes 1028, 1029, 9825.

Function name	Comments
gMapPrGuam gMapPrGuam_i	Guam projection. EPSG code 9831.
gMapPrHotineObliqueMercator gMapPrHotineObliqueMercator_i	Hotine Oblique Mercator projection. EPSG codes 9812, 9815.
gMapPrKrovak gMapPrKrovak_i	Krovak projection. EPSG codes 1041, 9819.
gMapPrLabordeObliqueMercator gMapPrLabordeObliqueMercator_i	Laborde Oblique Mercator projection. EPSG code 9813.
gMapPrLambertAzimuthEqualArea gMapPrLambertAzimuthEqualArea_i	Lambert Azimuthal Equal Area projection. EPSG code 9820.
gMapPrLambertConicConf gMapPrLambertConicConf_i	Lambert Conic Conformal projection. EPSG codes 9801, 9802, 9826, 9803, 1051, 1102, 9817.
gMapPrLambertCylindricEqualArea gMapPrLambertCylindricEqualArea_i	Lambert Cylindrical Equal Area projection. EPSG codes 9835, 9834.
gMapPrLocalOrthograph gMapPrLocalOrthograph_i	Local Orthographic projection. EPSG code 1130.
gMapPrMercator gMapPrMercator_i	Mercator projection. EPSG codes 9804, 9805, 1044, 1026, 1024.
gMapPrModifAzimuthEquidist gMapPrModifAzimuthEquidist_i	Modified Azimuthal Equidistant projection. EPSG code 9832.
gMapPrObliqueEquatorStereo gMapPrObliqueEquatorStereo_i	Oblique Equatorial Stereographic projection. EPSG code 9809.
gMapPrOrthograph gMapPrOrthograph_i	Orthographic projection. EPSG code 9840.
gMapPrPolarStereo gMapPrPolarStereo_i	Polar Stereographic projection. EPSG codes 9810, 9829, 9830.
gMapPrTransMercator gMapPrTransMercator_i	Transverse Mercator projection. EPSG codes 9807, 9824, 9808.
gMapPrTunisiaMining gMapPrTunisiaMining_i	Tunisia Mining Grid projection. EPSG code 9816.
gMapPrVerticalPerspective	Vertical Perspective projection. EPSG codes 9838, 9839.

Table 3.2 Converting between coordinate systems

Function name	Comments
gMapPrEN2EtNt gMapPrEtNt2EN	Convert between [E,N] source coordinates and [Et,Nt] target coordinates. EPSG codes 9624, 9623, 9621.
gMapPrG2Geoc gMapPrGeoc2G	Convert between Geographic coordinates and Geocentric coordinates. EPSG code 9602.
gMapPrGeoc2Topo gMapPrTopo2Geoc	Convert between Geocentric coordinates and Topocentric coordinates. EPSG code 9836.
gMapPrG2Topo gMapPrTopo2G	Convert between Geographic coordinates and Topocentric coordinates. EPSG code 9837.

Table 3.3 Coordinate transformation between ellipsoids

Function name	Comments
gMapPrEN2ENPolynomComplex	Convert [E1,N1] coordinates in the source coordinate system to [E2,N2] coordinates in the target coordinate system using Polynomial transformation with complex numbers. EPSG code 9653.
gMapPrEN2ENOffset gMapPrEN2ENOffset_inv	Convert between [E1,N1] coordinates in the source coordinate system and [E2,N2] coordinates in the target coordinate system using Cartesian Grid Offsets. EPSG code 9656.
gMapPrG2GAbridgedMolodensky	Convert [B1,L1,h1] source Geographic coordinates to [B2,L2,h2] target Geographic coordinates using Abridged Molodensky transformation. EPSG code 9605.
gMapPrG2GPolynom	Convert [B1,L1] source Geographic coordinates to [B2,L2] target Geographic coordinates using General polynomial transformation. For General polynomial of degree 6, EPSG code 9648.
gMapPrG2GOffset gMapPrG2GOffset_inv	Convert between [B1,L1,h1] Geographic coordinates in the source reference system and [B2,L2,h2] Geographic coordinates in the target reference system using Geographic Offsets. EPSG codes 9660, (9619).
gMapPrGeoc2GeocP3 gMapPrGeoc2GeocP3_inv	3-parameter transformation. Convert between [X1,Y1,Z1] coordinates in the source geocentric reference system and [X2,Y2,Z2] coordinates in the target geocentric reference system. EPSG codes 1031,9603.
gMapPrGeoc2GeocH7 gMapPrGeoc2GeocH7_inv	7-parameter Helmert transformation. Convert between [X1,Y1,Z1] coordinates in the source geocentric reference system and [X2,Y2,Z2] coordinates in the target geocentric reference system. The Coordinate Frame Rotation (geocentric domain) is used. EPSG codes 1032, (1033), 1132.
gMapPrGeoc2GeocMB10 gMapPrGeoc2GeocMB10_inv	10-parameter Molodensky-Badekas transformation. Convert between [X1,Y1,Z1] coordinates in the source geocentric reference system and [X2,Y2,Z2] coordinates in the target geocentric reference system. The Coordinate Frame Rotation (geocentric domain) is used. EPSG codes 1034, (1061), 9636, (1063).

4. Accuracy of coordinate transformation

In order to estimate the error of coordinate transformation and test the functions' operation, one sequentially performed "forward" and "reverse" coordinate transformations, followed by calculation of the difference between the obtained values and the original coordinates. The calculations were performed based on the test examples given in [1].

The geographic coordinate projections and other "degrees-to-meters" conversions had their calculations performed sequentially in the following form:

[E,N] >> [B,L] >> [Es,Ns]; dE=Es-E, dN=Ns-N

or

[X,Y,Z] >> [B,L,h] >> [Xs,Ys,Zs]; dX=Xs-X, dY=Ys-Y, dZ=Zs-Z.

The "meter-to-meter" conversions had their calculations performed sequentially as follows:

[E,N] >> [Et,Nt] >> [Es,Ns]; dE=Es-E, dN=Ns-N

or

[X1,Y1,Z1] >> [X2,Y2,Z2] >> [X1s,Y1s,Z1s]; dX=X1s-X1, dY=Y1s-Y1, dZ=Z1s-Z1.

or

[X,Y,Z] >> [U,V,W] >> [Xs,Ys,Zs]; dX=Xs-X, dY=Ys-Y, dZ=Zs-Z.

The transformations described must have the source and "calculated" coordinates coincide (the difference value must be equal to zero). The discrepancy between the coordinates indicates the "coordinate transformation" formulas' error. Obviously, the error will also depend on the input data and transformation parameters; therefore, the results obtained are not complete, but only give an idea of the formulas' error for the "special case". The results of calculations are summarized in ***Table 4.1***; the case identifies the input parameter combination number for the m-file (see the comments at the beginning of each m-file).

Table 4.1 Values of the difference between the coordinates calculated using the "forward and reverse" transformation and the source coordinates

No	Function	Case	dE or dX (m)	dN or dY (m)	dZ (m)
1	gMapPrAlbersEqualArea	1	4.4 e-05	-6.2 e-04	
		2	1.5 e-04	1.6 e-03	
2	gMapPrAmericPolyconic		9.3 e-09	4.2 e-03	
3	gMapPrBonne	1	0	0	
		2	0	0	
4	gMapPrCassiniSoldner	1	4.8 e-08	6.4 e-06	
		2	-1.8 e-09	-2.9 e-07	
5	gMapPrColombiaUrban		1.0 e-04	-3.6 e-04	
6	gMapPrEN2ENOffset		0	0	
7	gMapPrEN2ENPolynomComplex		+	+	
8	gMapPrEN2EtNt	1	2.9 e-11	0	
		2	-5.8 e-11	-1.9 e-09	
		3	0	0	
9	gMapPrEqualEarth	1	-1.2 e-04	-1.1 e-03	
		2	0	0	
10	gMapPrEquidistConic		-4.7 e-10	1.9 e-09	
11	gMapPrEquidistCylindric	1	0	-9.3 e-10	
		2.1	0	0	
		2.2	0	0	
		3	0	0	

No	Function	Case	dE or dX (m)	dN or dY (m)	dZ (m)
12	gMapPrG2GAbridgedMolodensky	1	+	+	
	(approximate reverse)	2	+	+	
13	gMapPrG2Geoc		0	2.9 e-11	0
14	gMapPrG2GOffset		0	0	0
15	gMapPrG2GPolynom		+	+	
16	gMapPrG2Topo		0	0	0
17	gMapPrGeoc2GeocH7	1	0	2.9 e-11	0
		2	0	0	0
18	gMapPrGeoc2GeocMB10	1/1	0	0	0
		2/1	4.4 e-06	1.6 e-05	-1.7 e-06
		3/1	4.4 e-06	1.6 e-05	-1.7 e-06
	(approximate reverse)	1/2	5.6 e-04	1.2 e-0.2	6.4 e-03
	(approximate reverse)	2/2	5.6 e-04	1.2 e-0.2	6.4 e-03
	(approximate reverse)	3/2	5.6 e-04	1.2 e-0.2	6.4 e-03
19	gMapPrGeoc2GeocP3		0	0	0
20	gMapPrGeoc2Topo		2.9 e-11	0	0
21	gMapPrGuam		2.2 e-11	6.0 e-08	
22	gMapPrHotineObliqueMercator	1	3.0 e-08	-7.1 e-06	
		2.1	3.0 e-08	-7.1 e-06	
		2.2	3.8 e-06	-5.0 e-06	
23	gMapPrKrovak	1	-4.7 e-10	2.3 e-10	
		2	4.7 e-10	-2.3e-10	
24	gMapPrLabordeObliqueMercator		0	0	
25	gMapPrLambertAzimuthEqualArea	1	-2.3 e-05	-3.4 e-04	
		2	-2.8 e-05	-3.2 e-04	
		3	1.0 e-05	-1.2 e-04	
26	gMapPrLambertConicConf	1	0	3.7 e-09	
		2	-1.2 e-09	0	
		3	0	3.7 e-09	
		4	0	0	
		5	0	0	
		6	0	0	
		7	0	0	
27	gMapPrLambertCylindricEqualArea	1	0	0	
		2	0	0	
	(warning)	3	0	0	
		4	0	0	
		5	0	0	
		6	-7.5 e-09	1.8 e-09	
		7	0	0	
28	gMapPrLocalOrthograph		1.7 e-10	-3.3 e-10	
29	gMapPrMercator	1	0	4.3 e-06	
		2	0	-2.5 e-07	
		3	0	-2.5 e-07	
		4	0	0	
		5	0	0	
30	gMapPrModifAzimuthEquidist		4.8 e-07	3.7 e-09	
31	gMapPrObliqueEquatorStereo		5.8 e-11	0	
32	gMapPrOrthograph		-1.6 e-06	7.1 e-08	
33	gMapPrPolarStereo	10	2.4 e-02	2.4 e-02	
		11	-5.3 e-04	5.5 e-04	
		20	-5.4 e-04	-4.5 e-04	
		21	8.7 e-04	-7.3 e-04	
		30	-1.0 e-06	-8.3 e-04	
		31	9.9 e-07	-8.0 e-04	
34	gMapPrTransMercator	1	2.0 e-08	-1.7 e-07	
		2	2.7 e-08	-1.7 e-07	
		3	-6.7 e-09	-2.3 e-07	

No	Function	Case	dE or dX (m)	dN or dY (m)	dZ (m)
35	gMapPrTunisiaMining		0	0	
36	gMapPrVerticalPerspective	1	+	+	
		2	+	+	

Cells with an error value greater than 1 mm as well as cells corresponding to “approximate calculations” are marked in red.

The functions without “reverse transformations” (gMapPrEN2ENPolynomComplex, gMapPrG2GAbridgedMolodensky, gMapPrG2GPolynom, gMapPrVerticalPerspective) had no their calculations, described above, performed. If the calculation results coincided with the test examples in [1] (to the nearest rounding error), the results were marked with a “+” sign.

References

1) IOGP Report 373-07-02, EPSG Guidance Note number 7, part 2, July 2024.

2) Snyder, John Parr. Map projections – a working manual. U.S. Geological Survey professional paper; 1395.