56
Review paper
Doi 10.7251/stp202401035DJ
ISSN 2566-4484

Đorđe Đermanović, Military Geographical Institute, djordjedjermanovic99@gmail.com
Novak Roganović, Military Geographical Institute, novakroganovic1999@gmail.com
Vujadin Stanojković, Military Geographical Institute, vuja755@gmail.com
Jelena Savić, Military Geographical Institute, jelenasavic1903@gmail.com
Siniša Drobnjak, Military Geographical Institute, sinisa.drobnjak@vs.rs

COMPARATIVE ANALYSIS OF DIFFERENT VARIANTS OF AZIMUTHAL CONFORMAL PROJECTION FOR THE TERRITORY OF THE REPUBLIC OF SERBIA

Abstract

The paper analyzes four variants of conformal azimuthal projections for the territory of the Republic of Serbia. In the procedure, a double conformal mapping was applied. As it is common practice to perform mapping in azimuthal projections from a sphere to a plane, it was initially necessary to perform a conformal mapping from the ellipsoid to the sphere, after which a conformal mapping from the sphere to the plane was executed. The central point of the projection is a point with a value of 44° north latitude and 21° east longitude. The mapping territory is trapezoidal in shape, defined by the values of the geographic latitudes and longitudes of the southernmost (41°53'), northernmost (46°11'), westernmost (18°49'), and easternmost (23°) points in the territory of the Republic of Serbia. The mapping variants are defined based on different values of linear scale at the central point of the projection. Data processing was carried out within the Jupyter Notebook component, part of the Anaconda software package, using the Python programming language. The results are presented numerically, graphically, and visualized using AutoCAD software.

Keywords: Azimuth projection, conformal mapping, linear scale, linear deformations

КОМПАРАТИВНА АНАЛИЗА РАЗЛИЧИТИХ ВАРИЈАНТИ АЗИМУТНЕ КОНФОРМНЕ ПРОЈЕКЦИЈЕ ЗА ТЕРИТОРИЈУ РЕПУБЛИКЕ СРБИЈЕ

Сажетак

У раду су анализиране четири варијанте конформних азимутних пројекција за територију Републике Србије. У поступку рада је примијењено двоструко конформно пресликавање. Како је пракса да се пресликавање код азимутних пројекција врши са сфере на раван, првобитно је било потребно извршити конформно пресликавање елипсоида на сферу, након чега је извршено конформно пресликавање са сфере на раван. Централна тачка пројекције јесте тачка са вриједношћу 44° сјеверне географске ширине и 21° источне географске дужине. Територија пресликавања је трапезастог облика, дефинисана вриједностима географских ширина и дужина најјужније (41°53`), најсјеверније (46°11`), најзападније (18°49`,) и најисточније (23°) тачке на територији Републике Србије. Варијанте пресликавања су дефинисане на основу различитих вриједности линеарног размјера у централној тачки пројекције. Обрада података је вршена у оквиру компоненте *Jupyter Notebook*, софтверског пакета *Anaconda* употребом *Python* програмског језика. Резултати су приказани нумерички, графички и визуализовани су помоћу програмског софтвера *AutoCAD*.

Кључне ријечи: Азимутна пројекција, конформно пресликавање, линеарни размер, линеарне деформације

1. INTRODUCTION

The first cartographic projections (from the group of perspective and azimuthal projections) date back to the ancient period and are attributed to ancient Greek scientists, who proposed them for astronomical and other maps. It should be noted that with the appearance of these maps, a stage in the development of cartography begins in which cartography, along with mathematical cartography, takes on the characteristics of scientifically based disciplines [1].

In azimuthal projections, the plane of projection touches the Earth at a chosen point [2]. Depending on the latitude of the point where the plane of projection touches the sphere, azimuthal projections can be divided into three categories:

- vertical (straight) or polar where $\varphi_\circ = 90°$;
- oblique where $90° < \varphi_\circ < 90°$ and
- transverse (equatorial) where $\varphi_\circ = 0°$.

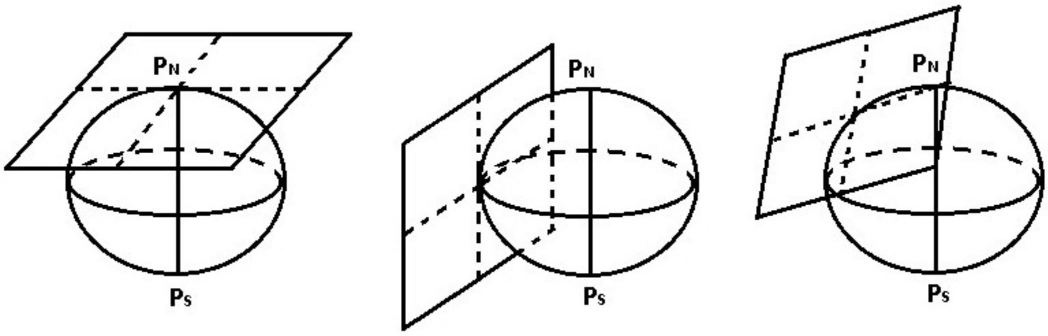


Figure 1. Vertical, transverse, and oblique azimuthal projection

According to the position of the projection point, azimuth projections are divided into (Figure 2) [3]:

- Gnomonic (projection center is in the center of the Earth);
- Stereographic (the projection center is on the opposite pole) i
- Orthographic (center of projection is at infinity)

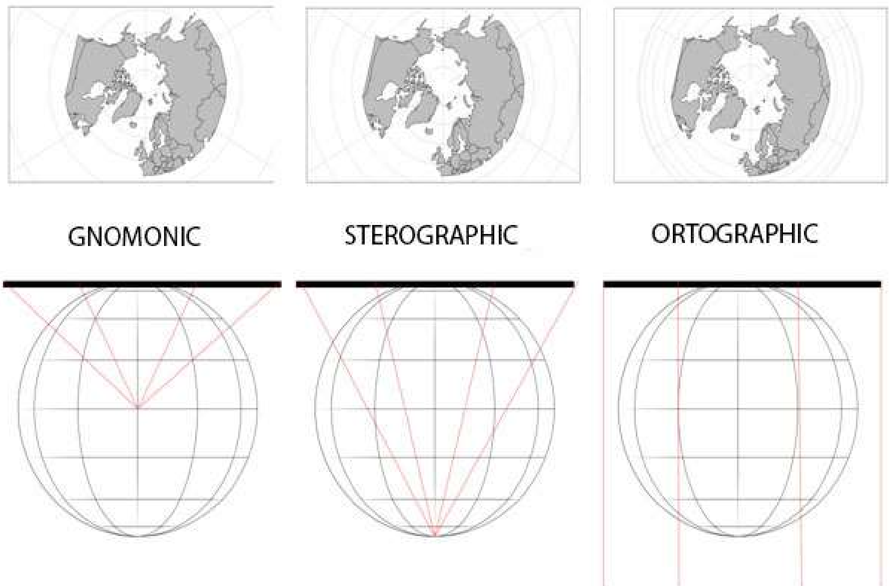


Figure 2. Division of azimuth projections in relation to the point of projection [3]

In the study, an oblique azimuthal stereographic projection was used with a central point at 44° north latitude and 21° east longitude because this point is located at the center of the territory to be mapped. As it is common practice in azimuthal projections to map from a sphere to a plane, it was necessary to first conformally map the ellipsoid onto a sphere [4]. The WGS84 ellipsoid was chosen for this

purpose, and conformal mapping was performed from it onto a sphere. After conformally mapping from the ellipsoid to the sphere, a conformal mapping from the sphere to the plane was carried out [5]. This process involved a double mapping [6].
The mapping area includes the entire territory of the Republic of Serbia and is trapezoidal in shape. Since the aim of the study is to analyze linear deformations in various variants of azimuthal conformal projection, what defines the variants and input parameters for data processing is the linear dimension at the central point of the projection. The linear dimension represents the relationship between corresponding linear elements on the projection surface and the original surface.
The variants that will be considered in the study are as follows:

- Variant 1: $\mu = 0.9996$
- Variant 2: $\mu = 0.9997$
- Variant 3: $\mu = 0.9998$
- Variant 4: $\mu = 0.9999$

2. DATA PROCESSING

The first step in data processing is to define the radius of the sphere that is most suitable for mapping from the WGS84 ellipsoid depending on the parallel where the central point of the projection is located and conformally mapping the ellipsoid to the sphere, which is defined by the following formulas [1]:

$$R = a * (1 - \frac{e^2 * sin^2(44°)}{2}) \tag{1}$$

$$\varphi' = \varphi - \frac{1}{2} \mathrm{e}^2 sin^2 \varphi \tag{2}$$

$$\lambda' = \lambda \tag{3}$$

Then it was necessary to determine the integration constant that appears in the formula for calculating the linear dimension, as well as the deformation as a function of latitude. The integration constant (c) is calculated on the basis of the following formula in which the unit z figures as the value of the degree distance from the central point of the projection:

$$\mu = \frac{C}{2 * R} sec^2 \frac{z}{2} \tag{4}$$

When using this formula, the value of the linear scale varied depending on the variant under consideration. The value of the integration constant for the first variant in the code was determined as follows:

```
ckonst=(0.9996*2*R)
```

Figure 3. The line of code for calculating value of integration constant

After the equation for the linear scale is equal to one, the distances from the central parallel are obtained where the values of the linear deformations d are equal to zero (d = |1-μ|), where cen represents the value of the geodetic latitude of the central point of the projection. This is written in the code for the first variant as follows [7]:

```
rast=2*acos(sqrt(ckonst/(2*R)))

rast1=rad2DMS(cen-rast)

rast2=rad2DMS(rast+cen)
```

Figure 4. The lines of code for calculating distances from the central parallel where the values of linear deformations are equal to zero

In the code, two lists are defined with values of geographic latitudes and longitudes, and they have the following values:

- Geographic latitudes: 41°53`, 42°, 42°30`, 43°, 43°30`, 44°, 44°30`, 45°, 45°30`, 46°, 46°11`
- Geographic longitudes: 18°49`, 19°, 19°30`, 20°, 20°30`, 21°, 21°30`, 22°, 22°30`, 23°

The defined values of geographic latitudes and longitudes on the ellipsoid are input data for calculating spherical geographic coordinates[8]. After obtaining these values, mapping from the ellipsoid to the sphere was performed, enabling further processing of conformal mapping from the sphere to the plane and obtaining rectangular coordinates, linear scale values, and linear deformation values. The formulas used in the process of calculating these values are as follows (variant 1) [9]:

$$\mu = \frac{C}{2*R} sec^2 \frac{z}{2} \tag{5}$$

$$x = 2 * R * 0.9996 * \frac{sin\phi cos\phi_{\circ} - cos\phi sin \;\; _{\circ}cos\lambda}{1+(sin\phi sin \;\; _{\circ}+cos\phi cos\phi_{\circ}cos\lambda)} \tag{6}$$

$$y = 2 * R * 0.9996 * \frac{cos\phi sin\lambda}{1+(sin\phi sin\phi_{\circ}+cos\phi cos \;\; _{\circ}cos\lambda)} \tag{7}$$

$$d = \left|1 - \mu\right| \tag{8}$$

Since we are dealing with variants of conformal azimuthal projection, the values of angular deformation are equal to zero, considering that the condition of conformity is defined during the derivation of the formulas used in the study.

3. ANALYSIS OF RESULTS

The following tables show the values of linear scales and deformations depending on the geographic latitude in four variants:

Table 1. Values of linear scale and deformation depending on the geodetic latitude- Variant 1

Geodetic Latitude	Linear Scale	Deformation Value
41°53`	0.9999389	0.0000611
42°00`	0.9999025	0.0000975
42°30`	0.9997702	0.0002298
43°00`	0.9996756	0.0003244
43°30`	0.9996189	0.0003811
44°	0.9996	0.0004
44°30`	0.9996189	0.0003811
45°00`	0.9996756	0.0003244
45°30`	0.9997702	0.0002298
46°00`	0.9999025	0.0000975
46°11`	0.9999605	0.0000395

Table 2. Values of linear scale and deformation depending on the geodetic latitude- Variant 2

Geodetic Latitude	Linear Scale	Deformation Value
41°53`	1.0000389	0.0000389
42°00`	1.0000026	0.0000026
42°30`	0.9998702	0.0001298
43°00`	0.9997756	0.0002244
43°30`	0.9997189	0.0002811
44°	0.9997	0.0003
44°30`	0.9997189	0.0002811
45°00`	0.9997756	0.0002244
45°30`	0.9998702	0.0001298
46°00`	1.0000026	0.0000026
46°11`	1.0000606	0.0000606

Table 3. Values of linear scale and deformation depending on the geodetic latitude- Variant 3

Geodetic Latitude	Linear Scale	Deformation Value
41°53`	1.0001389	0.0001389
42°00`	1.0001026	0.0001026
42°30`	0.9999702	0.0000298
43°00`	0.9998756	0.0001244
43°30`	0.9998189	0.0001811
44°	0.9998	0.0002
44°30`	0.9998189	0.0001811
45°00`	0.9998756	0.0001244
45°30`	0.9999702	0.0000298
46°00`	1.0001026	0.0001026
46°11`	1.0001606	0.0001606

Table 4. Values of linear scale and deformation depending on the geodetic latitude- Variant 4

Geodetic Latitude	Linear Scale	Deformation Value
41°53`	1.000239	0.000239
42°00`	1.0002026	0.0002026
42°30`	1.0000702	0.0000702
43°00`	0.9999756	0.0000244
43°30`	0.9999189	0.0000811
44°	0.9999	0.0001
44°30`	0.9999189	0.0000811
45°00`	0.9999756	0.0000244
45°30`	1.0000702	0.0000702
46°00`	1.0002026	0.0002026
46°11`	1.0002606	0.0002606

In the following tables, values of geodetic latitudes where the deformation value is equal to zero are presented for each variant (Values are calculated based on formula (4), where the unit μ is equal to one):

Table 5. Values of geodetic latitudes without deformation - Variant 1

d=0	o	‘	‘‘
Latitude 1	41	42	28.85761
Latitude 2	46	17	31.14238

Table 6. Values of geodetic latitudes without deformation - Variant 2

d=0	o	‘	‘‘
Latitude 1	42	0	54.42020
Latitude 2	45	59	5.579794

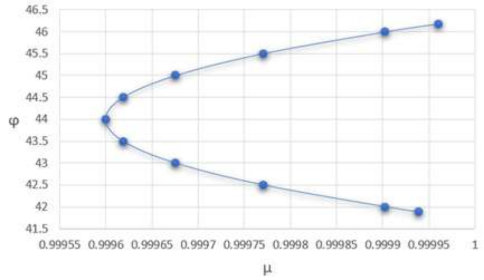
Table 7. Values of geodetic latitudes without deformation - Variant 3

d=0	o	‘	‘‘
Latitude 1	42	22	45.75578
Latitude 2	45	37	14.24421

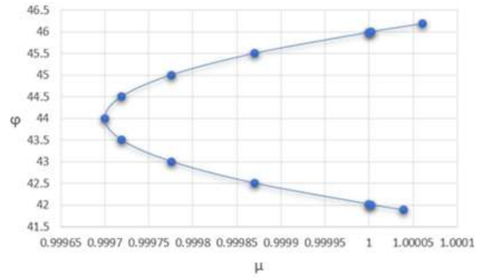
Table 8. Values of geodetic latitudes without deformation - Variant 4

d=0	o	‘	‘‘
Latitude 1	42	51	14.63511
Latitude 2	45	8	45.36488

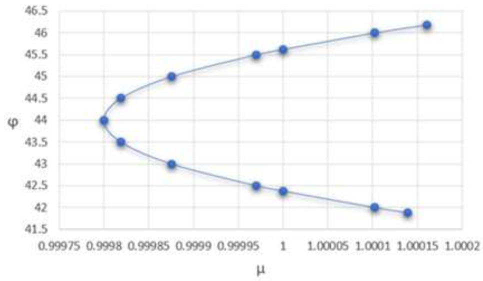
The following graphs represent the relationship between linear scale and the value of geodetic latitude for all four variants:



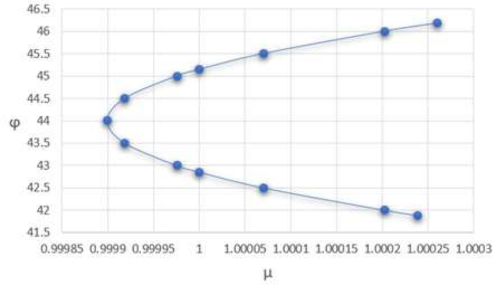
Graph 1: Dependence of linear scale value on geodetic latitude - Variant 1



Graph 2: Dependence of linear scale value on geodetic latitude - Variant 2



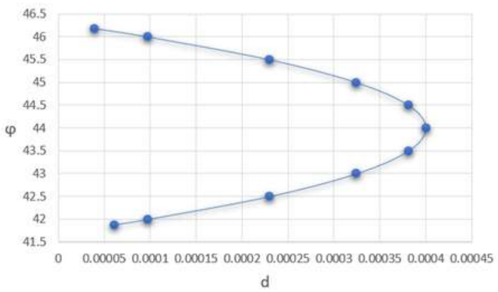
Graph 3: Dependence of linear scale value on geodetic latitude - Variant 3



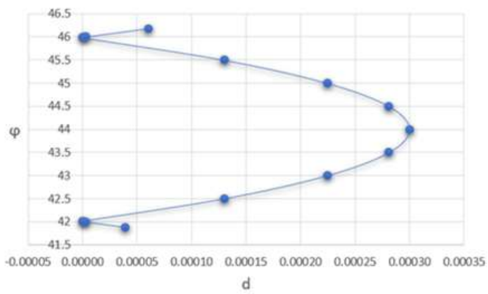
Graph 4: Dependence of linear scale value on geodetic latitude - Variant 4

Figure 5. Dependence of linear scale value on geodetic latitude

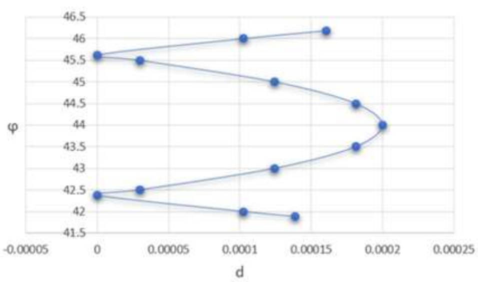
The following graphs show the values of deformations depending on geodetic latitudes:



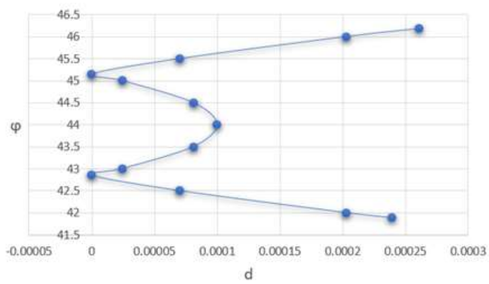
Graph 5: Dependence of deformation values on geodetic latitude - Variant 1



Graph 6: Dependence of deformation values on geodetic latitude - Variant 2



Graph 7: Dependence of deformation values on geodetic latitude - Variant 3



Graph 5: Dependence of deformation values on geodetic latitude - Variant 4

Figure 6. Dependence of deformation values on geodetic latitude

The following images show the generated coordinate grid in the *AutoCAD* environment with a visual representation of deformations for the given territory in all four variants under consideration. Red areas represent regions where the deformation value is between 3 dm/km and 4 dm/km, orange areas represent regions with deformations between 2 dm/km and 3 dm/km, lighter orange areas represent regions with deformations between 1 dm/km and 2 dm/km, while green areas represent regions where the deformation value is less than 1 dm/km.

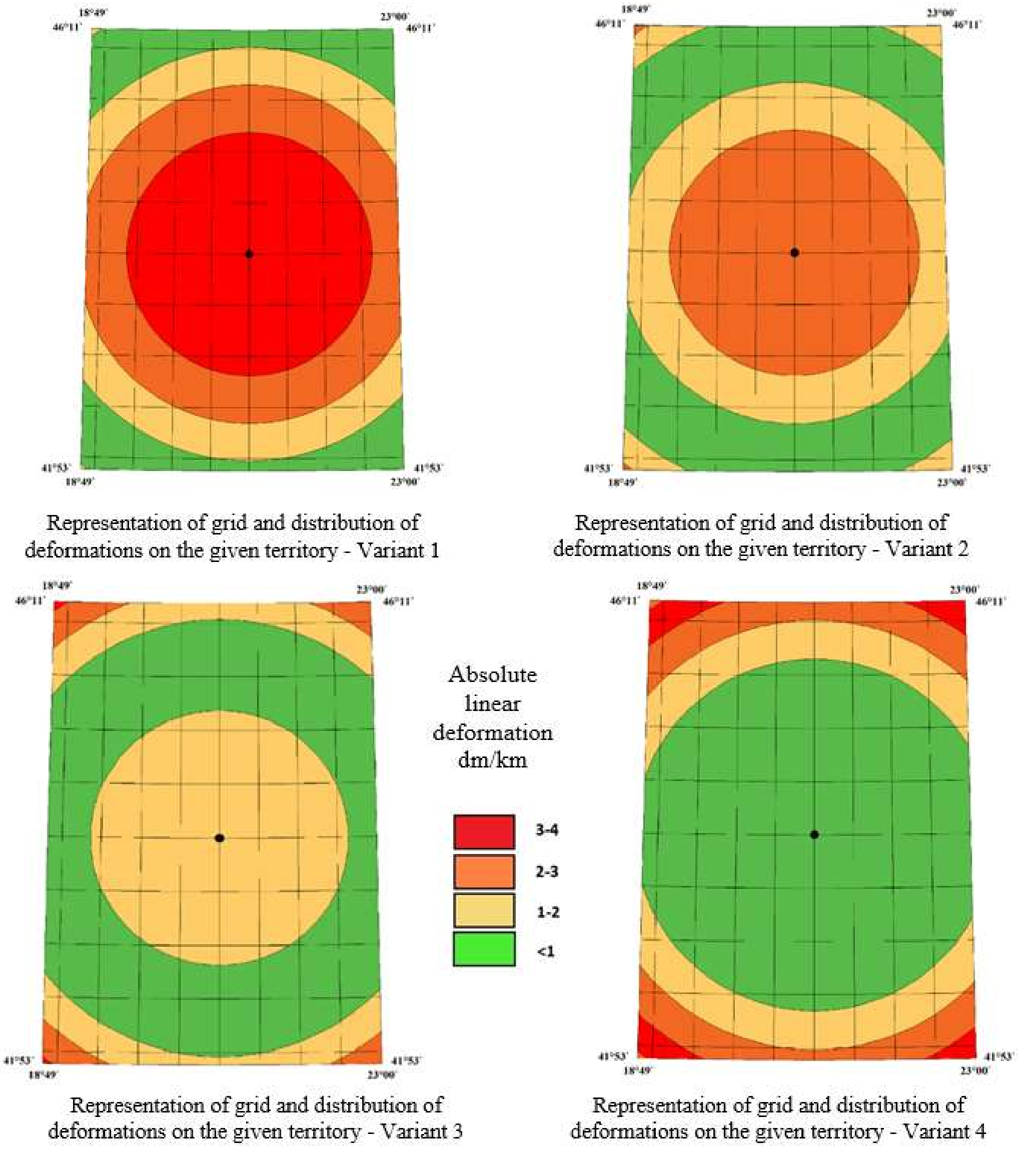


Figure 7. Absolute linear deformation dm/km

4. CONCLUSION

After examining the results presented numerically, in tables, in the form of diagrams, and graphically, certain conclusions can be drawn. When considering the values of linear deformations in all four variants, it can be observed that in the first variant, a larger part of the territory is affected by linear deformations whose value exceeds 2 dm/km, with a large percentage of areas where deformations are even greater than 3 dm/km. In the second variant, deformations between 2 dm/km and 3 dm/km are prevalent in the central part of the territory, with no areas having deformations exceeding 3 dm/km. The values of deformations in the third variant are mostly in the range between 0 dm/km and 2 dm/km, except for border areas where they become larger. In the fourth variant, it is noticeable that the central part of the territory is affected by deformations up to 1 dm/km, while deformations significantly increase towards the border of the territory, reaching values up to 4 dm/km. Depending on the desired accuracy and the allowed maximum value of linear deformations, as well as defining areas of interest within the mapping territory, it is necessary to choose optimal variant of azimuthal projection [10], [11]. Considering the need to ensure conformal mapping where the boundary of linear deformations is defined not to exceed the value of 3 dm/km, with equal importance of the central part of the mapping territory as well as the boundary parts, then the second variant is optimal. Additionally, the second variant is optimal if there is a requirement for linear deformations at the boundary parts of the mapping territory to be minimal, while their maximum value in the central part of the territory does not exceed 3 dm/km. If it is required that linear deformation on the largest part of the mapping territory be less than 1 dm/km, then the fourth variant is optimal, despite significant deformations occurring as one moves away from the central projection point. If the allowed boundary of linear deformations is increased from 1 dm/km to 2 dm/km, so that the largest part of the territory is covered by deformations up to this boundary value, then the third variant is optimal.

LITERATURE

[1] V. Jovanović, *Matematička kartografija*. Beograd: Vojnogeografski institut, 1983.

[2] B. Borčić, Matematička kartografija (Kartografske projekcije). Zagreb: Tehnička knjiga, 1955.

[3] M. Kilibarda, Matematička kartografija sa primerima u R-u i Pajtonu. Beograd: Akademska misao, 2023.

[4] R. J. Lisle and P. R. Leyshon, Stereographic projection techniques for geologists and civil engineers. Cambridge University Press, 2004.

[5] M. Borisov, B. Božić, and S. Drobnjak, “Mathematical models of rhumb line and their application,” *Tehnika*, vol. 66, no. 4, pp. 551–556, 2011.

[6] W. A. P. Smith and E. R. Hancock, “Modelling surface normal distribution using the azimuthal equidistant projection,” Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3604 LNCS, pp. 381–394, 2005, doi: 10.1007/11537908_23/COVER.

[7] J. R. Gott, C. Mugnolo, and W. N. Colley, “Map Projections Minimizing Distance Errors,” *https://doi.org/10.3138/carto.42.3.219*, vol. 42, no. 3, pp. 219–234, Jan. 2008, doi: 10.3138/CARTO.42.3.219.

[8] M. Lapaine and N. Frančula, “Map projection aspects,” *International Journal of Cartography*, vol. 2, no. 1, pp. 38–58, Jan. 2016, doi: 10.1080/23729333.2016.1184554.

[9] A. Dimitrijević, A. Milosavljević, and D. Rančić, “Efficient Distortion Mitigation and Partition Reduction in Mapping Global Geodata: Dual Orthogonal Equidistant Cylindrical Projection Approach,” *ISPRS International Journal of Geo-Information 2023, Vol. 12, Page 289*, vol. 12, no. 7, p. 289, Jul. 2023, doi: 10.3390/IJGI12070289.

[10] M. Borisov, V. M. Petrović, and M. Vulić, “Optimal map conic projection - a case study for the geographic territory of Serbia,” *Tehnicki Vjesnik-Technical Gazette*, vol. 22, no. 2, pp. 391–399, 2015, doi: 10.17559/TV-20140707123107.

[11] B. Jenny, “Adaptive composite map projections,” *IEEE Trans Vis Comput Graph*, vol. 18, no. 12, pp. 2575–2582, 2012, doi: 10.1109/TVCG.2012.192.

5. APPENDIX 1 - CODE IN JUPYTER NOTEBOOK

```
#loading the necessary modules
from math import *
from konverzijaUglova import *
import numpy as np
#WGS84 ellipsoid parameters
a=6378137
b=6356752.314
e=sqrt((a**2-b**2)/a**2)
cen=DMS2rad([44,0,0])
#calculation of the most suitable value of R
R=a*(1-1/2*e**2*pow(sin(cen),2))
ckonst=0.9997*2*R
#lists with the geodetic latitude and longitude values
fi=[DMS2rad([41,53,0]),DMS2rad([42,0,0]),DMS2rad([42,30,0]),DMS2rad([43,0,0]),DMS2rad([43,30,0]),D
MS2rad([44,0,0]),DMS2rad([44,30,0]),DMS2rad([45,0,0]),DMS2rad([45,30,0]),DMS2rad([46,0,0]),DMS2ra
d([46,11,0])]
fii=[[41,53,0],[42,0,0],[42,30,0],[43,0,0],[43,30,0],[44,0,0],[44,30,0],[45,0,0],[45,30,0],[46,0,0],[46,11,0]]
la=[DMS2rad([18,49,0]),DMS2rad([19,0,0]),DMS2rad([19,30,0]),DMS2rad([20,0,0]),DMS2rad([20,30,0]),D
MS2rad([21,0,0]),DMS2rad([21,30,0]),DMS2rad([22,0,0]),DMS2rad([22,30,0]),DMS2rad([23,0,0])]
laa=[[18,49,0],[19,0,0],[19,30,0],[20,0,0],[20,30,0],[21,0,0],[21,30,0],[22,0,0],[22,30,0],[23,0,0]]
fi_R=[]
fi_R1=[]
#loop for calculating spherical geographic latitudes
for i in range(11):
    fi_l=fi[i]-(1/2)*e**2*pow(sin(fi[i]),2)
    fi_R.append(fi_l)
    fi_R1.append(rad2DMS(fi_l))
lan=la[5]
fin=fi_R[5]
fin1=fi_R[0]
#distances where deformations are equal to zero
rast=2*acos(sqrt(ckonst/(2*R)))
rast1=rad2DMS(cen-rast)
rast2=rad2DMS(rast+cen)
x=[]
y=[]
n=[]
d=[]
fiix=[]
laax=[]
fiip1=[]
#double loop for calculating coordinates of intersection points, linear scale and deformations
for i in range(11):
    fix=fi_R[i]
    fip=fi[i]
    fip1=rad2deg(fip)
    for j in range(10):
        lax=la[j]
        xix=(2*R*0.9997*(sin(fix)*cos(fin1)-cos(fix)*sin(fin1)*cos(lax-
lan)))/(1+sin(fix)*sin(fin1)+cos(fix)*cos(fin1)*cos(lax-lan))
        yix=(2*R*0.9997*cos(fix)*sin(lax-lan))/(1+sin(fix)*sin(fin1)+cos(fix)*cos(fin1)*cos(lax-lan))
        nix=0.9997/(pow(cos((((fin-fix)))/2),2))
        dix=abs(1-nix)
        x.append(xix)
        y.append(yix)
        n.append(nix)
        d.append(dix)
        fiip1.append(fip1)
        fiix.append(rad2deg(fix))
        laax.append(rad2deg(lax))
tabela1 = np.column_stack(( fiix,laax,n,d))
np.savetxt('rezultati.csv', tabela1, delimiter=',', fmt='%.7f')

#code in the imported module konverzijaUglova
```

```
import math
def DMS2deg(ugaoDMS):
    ugaoDeg = ugaoDMS[0]+(ugaoDMS[1]/60)+(ugaoDMS[2]/3600)
    return (ugaoDeg)
def DMS2rad(ugaoDMS):
    ugaoRad = math.radians(ugaoDMS[0]+(ugaoDMS[1]/60)+(ugaoDMS[2]/3600))
    return (ugaoRad)
def deg2DMS(ugaoDeg):
    negative = ugaoDeg < 0
    dd = abs(ugaoDeg)
    minutes,seconds = divmod(dd*3600,60)
    degrees,minutes = divmod(minutes,60)
    if negative:
        if degrees > 0:
            degrees = -degrees
        elif minutes > 0:
            minutes = -minutes
        else:
            seconds = -seconds
    return [int(degrees), int(minutes), seconds]
def rad2deg(ugaoRad):
    return math.degrees(ugaoRad)
def rad2DMS(ugaoRad):
    ugaoDMS = deg2DMS(rad2deg(ugaoRad))
    return (ugaoDMS)
```