Maria Mańk 311590

Zadanie 4 – Przeliczenia między układami oraz elementarna skala długości i zniekształcenia

Prezentacja wyników – z przubliżeniami.

1. zestawienie współrzędnych						
	Xgk	Ygk	X2000	Y2000	X1992	Y1992
1	5570120.597	124812.228	5568256.030	7482170.562	266221.512	624724.859
2	5542315.026	125464.201	5540450.350	7482077.452	238435.405	625376.376
3	5571077.960	160469.907	5568256.030	7517829.438	267178.206	660357.578
4	5543273.892	161308.283	5540450.350	7517922.548	239393.600	661195.368
5	5556666.777	143014.239	5554323.110	7500000.000	252777.111	642914.129
6	5556698.104	143059.987	5554353.190	7500046.555	252808.416	642959.845

2. zestawienie pól powierzchni (km^2)			
P elipsoidalne	P gk	P 20000	P1992
994.265196	994.760761	994.108282	993.368584

3. elementarna skala długości i znekrztałcenia na 1km						
	mgk	Kgk(1km)	m2000	K2000(1km)	m1992	K1992(1km)
1	1.000193	0.19	0.999927	-0,07	0.999493	-0,51
2	1.000195	0.19	0.999927	-0,07	0.999494	-0,51
3	1.000319	0.32	0.999927	-0,07	0.999618	-0,38
4	1.000322	0.33	0.999927	-0,07	0.999621	-0,38
5	1.000253	0.25	0.999923	-0,08	0.999553	-0,45
6	1.000253	0.26	0.999923	-0,08	0.999554	-0,45

4. elementarna skala długości i znekrztałcenia na 1ha						
	mgk^2	Kgk^2(1ha)	m2000^2	K2000^2(1ha)	m1992^2	K1992^2(1ha)
1	1.000386	3,86	0.999854	-1,46	0.998986	-10,14
2	1.000389	3,89	0.999854	-1,46	0.998989	-10,11
3	1.000637	6,37	0.999854	-1,46	0.999236	-7,63
4	1.000643	6,43	0.999854	-1,46	0.999243	-7,57
5	1.000506	5,06	0.999846	-1,54	0.999105	-8,94
6	1.000506	5,06	0.999846	-1,54	0.999106	-8,94

Dane:

Współrzędne punktów pochodzą z wyników poprzedniego zadania i zostały zapisane w tablicy. Współrzędne punktu środkowego nie zostały przybliżone, aby otrzymać większą dokładność wynków

```
punktyfi = [50.25, 50.0, 50.25, 50.0, 50.125, 50.125270449027546]        #punkty wejściowe
punktylm = [20.75, 20.75, 21.25, 21.25, 21.0, 21.00065108883011]
p2000 = []                      #tablice wyników xy, xy itd
p1992 = []
pgk2 = []
pgk0 = []
mk = []
m2k2 = []
```

Utworzyłam również tablice do zapisywania wyników

Dodatkowo jako zmienne globalne zapisałam sobie parametry elipsoidy GRS80

```
a = 6378137 #metry
e2 = 0.00669438002290
```

Biblioteki:

Do uzyskania zamierzonego efektu wykorzystałam dwie biblioteki. Numpy posłużyła mi do przeliczeń matematycznych, a shapely do określenia pola w układzie 92.

```
import numpy as np
import shapely.geometry as shp
```

Poszczególnie funkcje znajdujące się w programie:

Funkcje pomocnicze zwracające M, N i elementarną skalę długości.

```
def Np(f, a, e2):
    N = a / np.sqrt(1 - e2 * (np.sin(f) ** 2))
    return N
def Mp(f, a, e2):
    M = (a * (1 - e2)) / ((np.sqrt(1 - e2 * np.sin(f) ** 2)) ** 3)
    return M
def m(y, r):
    mp = 1 + (y**2)/(2*(r**2)) + (y**4)/(24*(r**4))
    return mp
```

Kolejna funkcja pomocnicza zwracająca południk osiowy dla danego lambda w układzie 2000 (trzy - stopniowym):

```python
def l(lm):
    if lm >22.5 and lm < 25.5:
        lm0 = 24
    elif lm >19.5 and lm < 22.5:
        lm0 = 21
    elif lm < 19.5 and lm > 16.5:
        lm0 = 18
    elif lm < 13.5 and lm < 16.5:
        lm0 = 15
    return lm0
```

Funkcja zwracająca ze współrzędnych geodezyjnych współrzędne xy w odwzorowaniu Gaussa – Krügera:

```python
def geo2GK(fi, lm, lm0): #coś nie tak
    fi = np.deg2rad(fi)
    lm = np.deg2rad(lm)
    lm0 = np.deg2rad(lm0)
    ep2 = 0.0067394909677548 #kwadrat drugiego mimośrodu
    e2 = 0.006694380022900  #kwarat pierwszego mimośrodu
    a = 6378137
    N = a/np.sqrt(1 - e2*(np.sin(fi))**2)
    t = np.tan(fi)
    n2 = ep2*(np.cos(fi)**2)
    l = lm - lm0
    a0 = 1 - (e2/4) - (3*e2**2)/64 - (5*e2**3)/256
    a2 = (3/8)* ((e2 + (e2**2)/ 4) + (15 * (e2 ** 3)) / 128)
    a4 = (15/256)*(e2**2+(3*e2**3)/4)
    a6 = (35*(e2**3))/3072
    sigma = a*(a0*fi - a2*np.sin(2*fi)+ a4*np.sin(4*fi) - a6*np.sin(6*fi))
    x = sigma + ((l**2)/2)*N*np.sin(fi)*np.cos(fi)*(1+((l**2)/12)*(np.cos(fi)**2)*(5 - t**2 + 9*n2 + 4*(n2)**2) \
        + ((l**4)/360)*(np.cos(fi)**4)*(61 - 58*t**2 + t**4 + 270*n2 - 330*n2*(t**2)))
    y = l*N*np.cos(fi)*(1+((l**2)/6)*(np.cos(fi)**2)*(1 - t**2+n2) +
                        ((l**4)/120)*(np.cos(fi)**4)*(5 - 18*(t**2) + t**4 + 14*n2 - 58*n2*(t**2)))
    return x, y
```

Funkcje przeliczające współrzędne GK na poszczególne układu i na odwrót.

```python
def gk2u92 (xk, yk):
    m = 0.9993
    x = xk*m - 5300000
    y = yk*m + 500000
    return x, y
def gk2u00 (xk, yk, lm):
    m = 0.999923
    lm0 = l(lm)
    c = lm0/3
    x = m*xk
    y = m*yk + 500000 + c*1000000
    return x, y
```

```python
def u922gk (x, y):
    m = 0.9993
    xk = (x + 5300000)/m
    yk = (y - 500000)/m
    return xk, yk
def u002gk (x, y,lm0):
    m = 0.999923
    c = lm0/3
    xk = x/m
    yk = (y - c*1000000 - 500000)/m
    return xk, yk
```

Funkcja przeliczając współrzędne GK na elipsoidalne.

```python
def gk2el (xk, yk,lm0, a, e2):
    f = 1/(298.257)
    b = (a - (f*a))
    e22 = ((a**2 - b**2)/b**2)
    ep = 1
    lm0 = np.deg2rad(lm0)
    a0 = 1 - (e2 / 4) - (3 * e2 ** 2) / 64 - (5 * e2 ** 3) / 256
    a2 = (3 / 8) * ((e2 + (e2 ** 2) / 4) + (15 * (e2 ** 3)) / 128)
    a4 = (15 / 256) * (e2 ** 2 + (3 * e2 ** 3) / 4)
    a6 = (35 * (e2 ** 3)) / 3072
    fi0 = xk/(a*a0)
    sigma = a*(a0*fi0 - a2*np.sin(2*fi0)+ a4*np.sin(4*fi0) - a6*np.sin(6*fi0))
    eps = np.deg2rad(0.00001/3600)
    while ep> eps:
        fi1 = fi0 + (xk - sigma)/(a* a0)
        if abs(fi1 - fi0) < eps:
            break
        else:
            fi0 = fi1
        sigma = a * (a0 * fi1 - a2 * np.sin(2 * fi1) + a4 * np.sin(4 * fi1) - a6 * np.sin(6 * fi1))

    N = Np(fi1, a, e2)
    M = Mp(fi1, a, e2)
    t = np.tan(fi1)
    n2 = e22*(np.cos(fi1)**2)
    fi = fi1 - (((yk**2)*t)/(2*M*N) )* (1 - (yk**2)/(12*(N**2))*(5 + 3*(t**2) + n2 - 9*n2*(t**2) - 4*(n2**2))
                                    + (yk**4)/(360*(N**4))*(61 + 90 * (t**2) + 45*(t**4)))
    lm = lm0 + (yk/(N*np.cos(fi1))) * (1 - (yk**2)/(6*(N**2)) * (1 + 2 * (t**2)+n2)
         + ((yk**4)/(120*(N**4)))* (5 + 28* (t**2) + 24*(t**4) + 6* n2 + 8*n2*(t**2)))
    fi = np.rad2deg(fi)
    lm = np.rad2deg(lm)
    return fi, lm
```

Funkcja licząca elementarną skalę długości, pól, zniekształcenia na km i ha

```
def mkappa (x, y, a, e2, u, lmm, fi):
    if u == '92':
        m0 = 0.9993
        gk = u922gk(x, y)
        yk = gk[1]
        N = Np(fi, a, e2)
        M = Mp(fi, a, e2)
        R = np.sqrt(M*N)
        m1 =m(yk, R)*m0
        m2 = (m(yk, R)**2)*(m0**2)
        k = (m1 - 1)*1000
        k2 = (m2 - 1)*10000
    elif u == '00':
        m0 = 0.999923
        gk = u002gk(x, y, 21)
        yk = gk[1]
        N = Np(fi, a, e2)
        M = Mp(fi, a, e2)
        R = np.sqrt(M*N)
        m1 =m(yk, R)*m0
        m2 = (m(yk, R)**2)*(m0**2)
        k = (m1 - 1)*1000
        k2 = (m2 - 1)*10000
    elif u == 'None':
        N = Np(fi, a, e2)
        M = Mp(fi, a, e2)
        R = np.sqrt(M*N)
        m1 = m(y, R)
        m2 = m1**2
        k = (m1 - 1)*1000
        k2 = (m2 - 1) * 10000
    return m1, k, m2, k2
```

W programie znajduje się również funkcja z zadania 3 liczaca pole elipsoidalne

Najważniejsza część funkcji wywoływanie() odpwoeidzailnej za wywoływanie innych funkcji przedstawia się tak:

```
for i in punktyfi:
    fi = i
    lm = punktylm[p]
    lm0 = l(lm)
    gkk = geo2GK(fi, lm, lm0)       #GK dla 00  - odpowiedni poludnik osiowy
    gkkk = geo2GK(fi, lm, 19)       #GK dla 92 - poludnik osiowy 19
    pgk0.append([gkk[0], gkk[1]])
    pgk2.append([gkkk[0], gkkk[1]])
    u200 = gk2u00(gkk[0], gkk[1], lm0) #GK00 ---> 2000
    m00 = mkappa(u200[0], u200[1], a, e2, '00', lm0, fi)
    p2000.append([u200[0], u200[1]])
    u1992 = gk2u92(gkkk[0], gkkk[1])   #GK92---> 1992
    m92 = mkappa(u1992[0], u1992[1], a, e2, '92', 19, fi)
    p1992.append([u1992[0], u1992[1]])
    u92kk = u922gk(u1992[0], u1992[1]) #92 --->GK92
    u20kk = u002gk(u200[0], u200[1], lm0) #2000 --->GK00
    mGK0 = mkappa(u20kk[0], u20kk[1], a, e2, 'None', lm0, fi)
    mGK2 = mkappa(u92kk[0], u92kk[1], a, e2, 'None', 19, fi)
    gkqeo = gk2el(u92kk[0], u92kk[1], 19, a, e2) #GK92---> fi lm GRS80
    gkqeo1 = gk2el(u20kk[0], u20kk[1], lm0, a, e2) #GK00 ---> fi lm GRS80
    mk.append([mGK0[0],mGK0[1], mGK2[0], mGK2[1], m00[0], m00[1], m92[0], m92[1]])
    m2k2.append([mGK0[2], mGK0[3], mGK2[2], mGK2[3], m00[2], m00[3], m92[2], m92[3]])
    line = str(p + 1) + '\t' + str(fi) + '\t' + str(lm) + '\t' + str(round(gkk[0], 3))+'\t'+str(round(gkk[1], 3))+ '\t' \
           + str(round(gkkk[0], 3))+ '\t'+str(round(gkkk[1], 3)) + '\t' + str(round(u200[0], 3))+ '\t'+str(round(u200[1], 3)) + \
           '\t'+ str(round(u1992[0], 3))+ '\t'+str(round(u1992[1], 3))
    print(line)
    p+=1
print('------------------------------ZESTAWIENIE PÓL POWIERZCHNI--------------------------------')
pol = pole(a, e2, punktylm[0], punktylm[3], punktyfi[0], punktyfi[3]) / 1000000
pp92 = shp.Polygon([(p1992[0][0], p1992[0][1]),(p1992[1][0], p1992[1][1]), (p1992[3][0], p1992[3][1]), (p1992[2][0], p1992[2][1])])
pol92 = pp92.area/1000000
pol00 = (np.abs(p2000[3][0] - p2000[0][0])*np.abs(p2000[0][1] - p2000[3][1]))/ 1000000
```

Wynik działania programu (efekt funkcji):

```
------------------------------ZESTAWIENIE WSPÓŁRZĘDNYCH---------------------------------
pkt fi      lambda  X GK00      Y GK00      X GK92      Y GK92      X 2000      Y 2000      X 1992      Y 1992
1   50.25   20.75   5568684.819 -17830.811  5570120.597 124812.228  5568256.03  7482170.562 266221.512  624724.859
2   50.0    20.75   5540876.997 -17923.929  5542315.025 125464.201  5540450.35  7482077.452 238435.405  625376.376
3   50.25   21.25   5568684.819 17830.811   5571077.96  160469.907  5568256.03  7517829.438 267178.206  660357.578
4   50.0    21.25   5540876.997 17923.929   5543273.892 161308.283  5540450.35  7517922.548 239393.6    661195.368
5   50.125  21.0    5554750.826 0.0 5556666.777 143014.239  5554323.11  7500000.0   252777.111  642914.129
6   50.125270449027546  21.00065108883011   5554780.908 46.559  5556698.104 143059.987  5554353.19  7500046.555 252808.416  642959.845
------------------------------ZESTAWIENIE PÓL POWIERZCHNI--------------------------------
pole elip [km^2]    pole GK92 [km^2]    pole GK00 [km^2]    pole 2000 [km^2]    pole 1992 [km^2]
994.265196080189    994.760761498441    994.261392073965    994.108281714585    993.368583865118
-------------------------ELEMENTARNA SKALA DŁUGOŚCI I ZNIEKSZTAŁCENIA NA KM-------------------------
pkt mGK00       KGK00   mGK92       KGK92       m00     K00     m92     K92
0 1.0000039340555271 0.0039340555271216715 1.000192764143413 0.1927641434129601 0.9999269337526049 -0.07306624739511491 0.9994926292085126 -0.5073707914874381
1 1.0000039716022244 0.003971602224428494 1.0001946044568362 0.19460445683616712 0.9999269712964111 -0.07302870358893987 0.9994944682337163 -0.5055317662836512
2 1.0000039340555271 0.0039340555271216715 1.000318645869732 0.3186458697319594 0.9999269337526049 -0.07306624739511491 0.9996184228176231 -0.38157718237685767
3 1.0000039716022244 0.003971602224428494 1.000321688599201 0.3216885992010088 0.9999269712964111 -0.07302870358893987 0.9996214634171815 -0.37853658281850056
4 1.0 0.0 1.0002530243338323 0.25302433383234124 0.999923 -0.07699999999999374 0.9995528472167986 -0.44715278320139884
5 1.0000000000268159 2.6815882847586l8e-08 1.0002531864966853 0.25318649668526483 0.9999230000268138 -0.07699997318622032 0.9995530092661375 -0.44699073386245125
-------------------------ELEMENTARNA SKALA DŁUGOŚCI I ZNIEKSZTAŁCENIA NA HA-------------------------
pkt mGK00^2     KGK00^2 mGK92^2     KGK92^2     m00^2       K00^2       m92^2       K92^2
0 1.000007868126531 0.07868126530974351 1.0003855654448408 3.8556544484080746 0.9998538728438862 -1.4612715611383997 0.998985515842145 -10.14484157854989
1 1.0000079432202225 0.07943220222461633 1.000389246784567 3.892467845669767 0.9998539479260136 -1.4605207398643572 0.9989891920297994 -10.10807970200589
2 1.000007868126531 0.07868126530974351 1.0006373932746542 6.373932746541744 0.9998538728438862 -1.4612715611383997 0.9992369912363923 -7.6300876360768655
3 1.0000079432202225 0.07943220222461633 1.0006434806819569 6.434806819568539 0.9998539479260136 -1.4605207398643572 0.9992430701243076 -7.569298756924248
4 1.0 0.0 1.0005061126889783 5.061126889782663 0.999846005929 -1.5399407100002982 0.9991058943792088 -8.941056207911746
5 1.0000000000536318 5.363176569517236e-07 1.0005064370967727 5.064370967726717 0.9998460059826235 -1.5399401737647977 0.9991062183329913 -8.937816670087084
```

Wnioski:

- Największe zniekształcenia są gdy punkt znajduje się najdalej od południka osiowego.
- Na podstawie wielkości współrzędnych można oszacować w jakim układzie mamy dany punkt