

## Zadanie 5 – Transformacja między elipsoidami

### Zadanie:

Przetransformować dane współrzędne z ćwiczenia 4 z elipsoidy GRS80 na elipsoidę Krasowskiego.  
flH(GRS80) -> XYZ -Transformacja-> XYZ(Krasowski) -> flH  
Zaprezentować różnicę pomiędzy elipsoidami.

### Dane:

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

```
#---PARAMETRY GRS80
a = 6378137 #metry
e2 = 0.00669438002290
#---PARAMETRY KRASOWSKI
a2 = 6378245
e22 = (0.0818133340169)**2

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]
```

Dodatkowo jako zmienne globalne zapisałam sobie parametry elipsoidy GRS80 i elipsoidy Krasowskiego

### Biblioteki:

Do uzyskania zamierzonego efektu wykorzystałam jedną bibliotekę. Numpy posłużyła mi do przeliczeń matematycznych.

```
import numpy as np
```

### Poszczególne funkcje znajdujące się w programie:

Funkcja przeliczająca współrzędne geodezyjne na współrzędne xyz pochodzi z zadania 1.

Kolejna funkcja to funkcja odpowiadająca za transformacje pomiędzy elipsoidami – przeliczenie współrzędnych xyz GRS80 na współrzędne xyz elipsoidy Krasowskiego.

```
def xyzg2xyzk (xg, yg, zg):
    Tx = -33.4297
    Ty = 146.5746
    Tz = 76.2865
    m = 1 + 0.8407728 / 10**6
    ex = -1.7388854 / 10**6
    ey = -0.2561460 * 10**(-6)
    ez = 4.0896031 * 10**(-6)
    xk = m * xg + ez * yg - ey * zg + Tx
    yk = (-ez) * xg + m * yg + ex * zg + Ty
    zk = ey * xg - ex * yg + m * zg + Tz
    return xk, yk, zk
```

Gdy mamy już przeliczone współrzędne na współrzędne xyz Krasowskiego, to korzystając z algorytmu Hirvonen możemy wyliczyć  $\phi$  i  $\lambda$  elipsoidalne na elipsoidzie Krasowskiego.

```
def Hirvonen(x, y, z, a, e2):
    ep = np.deg2rad(0.000001/3600)
    r = np.sqrt(x**2+y**2)
    f1 = np.arctan(z/(r*(1-e2)))
    f2 = 2*f1
    while np.abs(f1 - f2) > ep:
        f2 = f1
        N = Np(a, e2, f1)
        h = r / np.cos(f1) - N
        f1 = np.arctan(z/(r*(1-e2*N/(N+h))))
    lam = np.arctan(y/x)
    N1 = Np(a, e2, f1)
    h1 = r / np.cos(f1) - N1
    f2 = np.rad2deg(f2)
    lam = np.rad2deg(lam)
    return f2, lam, N1, h1
```

```
def flg2flk():
    p = 0
    line = 'pkt' + '\t' + 'fi G..' + '\t' + 'lam G' + '\t' + 'H G\t' + 'X GRS80' + '\t' + 'Y GRS80' + \
          '\t' + 'Z GRS80' + '\t' + 'X Krasowski' + '\t' + 'Y Krasowski' + '\t' + 'Z Krasowski' + '\t' + 'fi Kras...' + \
          '\t' + 'lambda Kras' + '\t' + 'H Kras'
    print(line)
    for i in punktyfi:
        fi = i
        lm = punktylm[p]
        N = Np(a, e2, fi)
        h = 100
        xg, yg, zg = geo2xyz(fi, lm, h, a, e2) #fi, lambda GRS80 na xyz na GRS
        xk, yk, zk = xyzg2xyzk(xg, yg, zg) #xyz na GRS na xyz na Krasowkim
        fik, lmk, Nk, hk = Hirvonen(xk, yk, zk, a2, e22) #xyz Krasowski na fi lambda krasowski
        xg, yg, zg = rounding([xg, yg, zg], 3) # fi, lambda GRS80 na xyz na GRS
        xk, yk, zk = rounding([xk, yk, zk], 3) # xyz na GRS na xyz na Krasowkim
        fik, lmk = rounding([fik, lmk], 6)
        hk = rounding([hk], 3)
        if p!=5:
            line = str(p+1) + '\t' + str(round(fi,6)) + '\t' + str(round(lm,6)) + '\t' + str(h) + '\t' + str(xg) + '\t' + str(yg) + '\t' + str(zg) + \
            else:
                line = str(p + 1) + '\t' + str(round(fi, 6)) + '\t' + str(round(lm, 6)) + '\t' + str(h) + '\t' + str(
                    xg) + '\t' + str(yg) + '\t' + str(zg) + '\t' + str(xk) + '\t' + str(yk) + '\t' + str(zk) + '\t' + str(
                        fik) + '\t' + str(lmk) + '\t' + str(*hk)
        p+=1
    print(line)
```

Funkcja odpowiadająca za zaprezentowanie danych, której wynik prezentuje się następująco:

pkt	fi G..	lam G	H G X	GRS80	Y GRS80	Z GRS80	X Krasowski	Y Krasowski	Z Krasowski	fi Kras...	lambda Kras	H Kras
1	50.25	20.75	100	3821511.432	1447841.166	4880693.944	3821488.386	1447964.842	4880775.873	50.250293	20.751736	67.633
2	50.0	20.75	100	3841468.458	1455402.206	4862865.642	3841445.455	1455525.838	4862947.564	50.000296	20.751726	67.501
3	50.25	21.25	100	3808731.283	1481134.591	4880693.944	3808708.363	1481258.348	4880775.934	50.250285	21.251734	68.464
4	50.0	21.25	100	3828621.567	1488869.5	4862865.642	3828598.691	1488993.213	4862947.625	50.000288	21.251724	68.336
5	50.125	21.0	100	3825128.783	1468329.369	4871791.336	3825105.822	1468453.064	4871873.292	50.12529	21.00173	67.984
6	50.12527	21.000651	100	3825090.544	1468364.563	4871810.623	3825067.583	1468488.257	4871892.579	50.125561	21.002381	67.985

I jest to zestawienie współrzędnych we wszystkich fazach transformacji.

## Zestawienie współrzędnych w kolejnych fazach transformacji pomiędzy elipsoidami:

Zestawienie współrzędnych												
pkt	fi GRS80	lam GRS80	H GRS80	X GRS80	Y GRS80	Z GRS80	X Krasowski	Y Krasowski	Z Krasowski	fi Krasowski	lambda Krasowski	H Krasowski
1	50.25	20.75	100	3821511.432	1447841.166	4880693.944	3821488.386	1447964.842	4880775.873	50.250293	20.751736	67.633
2	50.0	20.75	100	3841468.458	1455402.206	4862865.642	3841445.455	1455525.838	4862947.564	50.000296	20.751726	67.501
3	50.25	21.25	100	3808731.283	1481134.591	4880693.944	3808708.363	1481258.348	4880775.934	50.250285	21.251734	68.464
4	50.0	21.25	100	3828621.567	1488869.5	4862865.642	3828598.691	1488993.213	4862947.625	50.000288	21.251724	68.336
5	50.125	21.0	100	3825128.783	1468329.369	4871791.336	3825105.822	1468453.064	4871873.292	50.12529	21.00173	67.984
6	50.12527	21.000651	100	3825090.544	1468364.563	4871810.623	3825067.583	1468488.257	4871892.579	50.125561	21.002381	67.985

## Wnioski:

- Współrzędne fi i lambda nieznacznie różnią się między elipsoidami. Duża różnica występuje jednak we współrzędnej H.