VIŠA GEODEZIJA

TEMA 6

Studenti:

Bajčeta Nikola GG4/2017 Mrvoš Kristijan GG11/2017

Doknić Ilija GG35/2017

Predmetni profesor:

Borisov dr. Mirko

Asisteniti:

Vrtunski dr. Milan

Bugarinović Željko

Gavrilović Milan

Tema 6: Primenom java/python programskog jezika izvrsiti redukciju proizvoljnog broja geodetskih merenja na osnovu plana opažanja (dužina,pravaca i uglova) za željeni elipsoid.





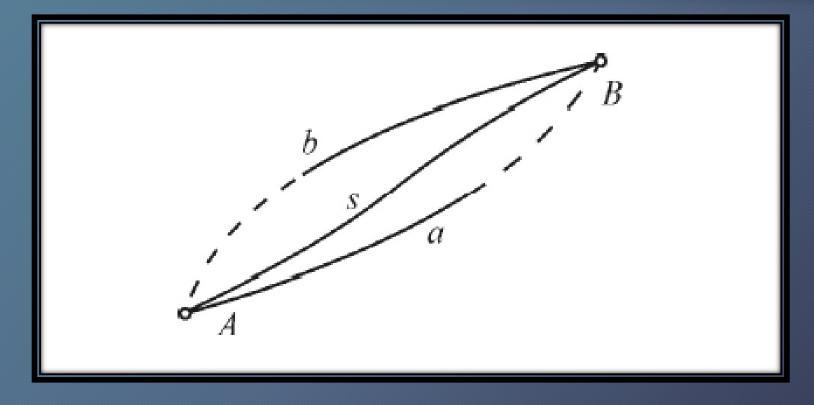




 Geodetska merenja vrše se na fizičkoj površi Zemlje, u trodimenzionalnom prostoru i u realnom polju sile teže. Međutim, da bi se ta merenja mogla kasnije računski obrađivati potrebno je izvršiti njihovu redukciju na referentni elipsoid. Takođe, razvoj satelitske geodezije koji je uveo u širu upotrebu globalne elipsoide (npr. WGS84, GRS80) koji se od lokalnih mogu znatno razlikovati, dovodi do potrebe za ovim proračunima.

1.1 Redukcija pravaca i horizontalnih uglova





Slika 1 - Normalni preseci I geodetska linija

1.1 Redukcija pravaca i horizontalnih uglova



Popravka 1 - C1

popravka za zamenu normalnih preseka geodetskom linijom.

Popravka 2 - C2

Popravka za visinu vizurne tacke

Popravka 3 - C3

Popravka za skretanje vertikala

$$c_1 = -\rho'' \frac{S^2 e^2}{12a^2} \cos^2 B_m \sin 2A$$

$$c_2 = \rho'' \cdot \frac{he^2}{2a} \cdot \cos^2 B_m \sin 2A$$

$$c_3 = (\eta \cos A - \xi \sin A) \cdot \frac{\Delta h}{S} = (\eta \cos A - \xi \sin A) \cdot \cot Z^0$$

1.1 Redukcija pravaca i horizontalnih uglova



Gde je:

ρ" = 206264,8 (vrednost radijana u seksagezimalnim sekundama)

a,e - elementi elipsoida, mala poluosa i prvi ekscentricitet

Bm - srednja vrednost latituda tačaka A i B

A - azimut sa stanice na vizuru

 ξ , η - komponente skretanja vertikale

h - apsolutna visina vizurne tačke

Δh - visinska razlika stanice i vizurne tačke

1.1 Redukcija pravaca i horizontalnih uglova



Nakont toga neophodno je definisati vrednost δ (suma popravaka) po formuli :

$$\delta = c1 + c2 + c3$$

Redukovani pravci se računaju tako što se opažani pravac popravi za vrednost δ.

Redukcija horizontalnih uglova

Redukcija merenog horizontalnog ugla se vrši tako što se računaju razlike popravaka pravaca $\delta 2$ - $\delta 1$ gde je $\delta 2$ desni a $\delta 1$ levi pravac.

1.Uvod1.2 Redukcija zenitnog ugla



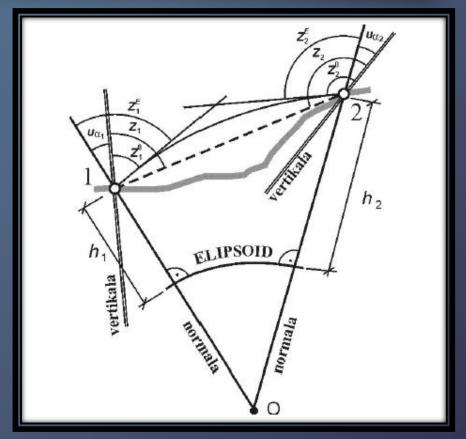
 Prilikom redukcije zenitnih uglova primenjujemo formule :

$$z_1^E = Z_1^0 + u_{\alpha 1}$$

$$z_2^E = Z_2^0 + u_{\alpha 2}$$

Odakle sledi da je vrednost u_a :

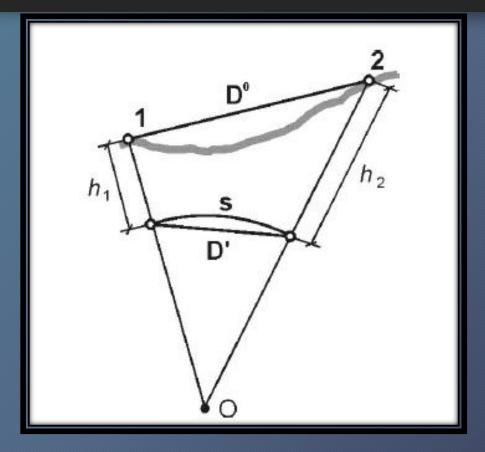
$$u_a = \xi \cos a + \eta \sin a$$



Slika 2 - Redukcija zenitnog ugla

1.Uvod1.3 Redukcija Dužina





Slika 3 - Redukcija dužina

1.Uvod 1.3 Redukcija Dužina



Korak 1

Svesti na tetivu - D'

Korak 1.2

Odredjivanje radijusa Zemlje - R_A

Korak 1.2.1

Određivanje radijusa zakrivljenosti duž M

$$D' = \sqrt{\frac{(D^0)^2 - \Delta h^2}{(1 + \frac{h_1}{R_A})(1 + \frac{h_2}{R_A})}}$$

$$R_A = \frac{MN}{N\cos^2 A + M\sin^2 A}$$

$$M = \frac{a(1 - e^2)}{(1 - e^2 \sin^2 B_m)^{3/2}}$$

1.Uvod1.3 Redukcija Dužina



Korak 1.2.2

Određivanje radijusa zakrivljenosti duž N

Prelazak sa tetive na dužinu redukovanu na normalan presek - s

Korak 3

Redukcija dužine sa normalnog preseka na geodetsku liniju - s_G

$$N = \frac{a}{(1 - e^2 \sin^2 B_m)^{1/2}}$$

$$s = 2R_A \arcsin \frac{D'}{2R_A}$$

$$s_G = s - \frac{e^4}{360a^4} s^5 \cos^4 B_m \sin^2 2A$$

2. PYTHON



Python je programski jezik visokog nivoa opšte namene.

Podržava, u prvom redu implementivni, objektno-orjentisani I funkcionalni stil programiranja.

Sintaksa jezika Python omogućava pisanje veoma preglednih programa.





2. PYTHON JUPYTER NOTEBOOK



Jupiter Notebook je veb aplikacija otvorenog koda koja vam omogućava da kreirate i delite dokumente koji sadrže živi kod, jednačine, vizuelizacije i narativni tekst. Upotrebe uključuju: čišćenje i transformaciju podataka, numeričku simulaciju, statističko modeliranje, vizuelizaciju podataka, mašinsko učenje i još mnogo toga.



2. PYTHON NUMPY I PANDAS BIBLIOTEKE



NumPY je biblioteka za programski jezik Python, dodajući podršku za velike, višedimenzionalne nizove i matrice, zajedno sa velikom kolekcijom matematičkih funkcija na visokom nivou za rad na tim nizovima.

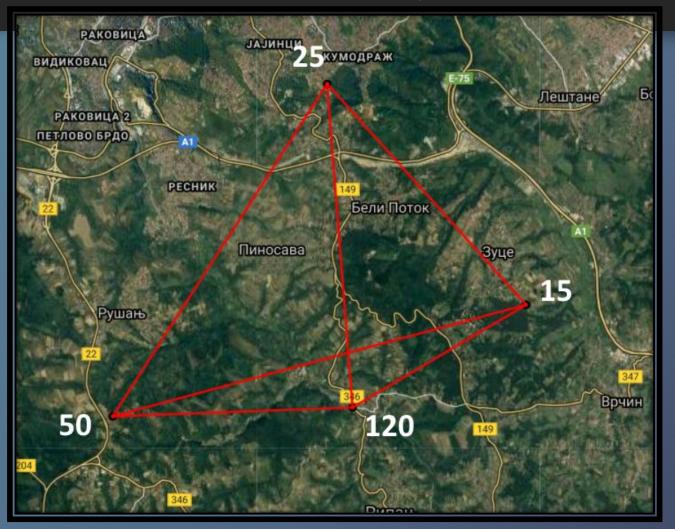




pandas je softverska biblioteka napisana za programski jezik Python za manipulaciju i analizu podataka. Posebno nudi strukture podataka i operacije za manipulisanje numeričkim tabelama i vremenskim serijama.

2. PYTHON PRIKAZ TERENA





Slika 4 Prikaz terena od interesa u Google Maps-u

2. PYTHON PRAKTIČAN DEO - KOD





Slika 5 Importovanje numpy-ja I pandasa I definisanje parametara GRS80 elipsoida

```
def degree2dms(a):
    minuta = (a%1)*60
    sekunda =(minuta%1)*60

    return[int(a),int(minuta),round(sekunda,5)]
```

Slika 6 konvertovanje u decimalni oblik

```
def dms2rad(deg):
    return np.radians(dms2degree(deg))

def deg2rad(deg):
    return np.radians(deg)
```

Slika 7 Konvertovanje u radijane







```
def redukcijaDuzina(koor1,koor2, Do, azimut):
   deltah= koor2[2] - koor1[2]
   phi1 = koor1[0]
   phi2 = koor2[0]
   Bm= (phi1 + phi2) / 2
   M = (a * (1 - e2)) / (m.sqrt(m.pow((1 - e2 * m.pow(m.sin(Bm) ,2)), 3)))
   N = a / (m.sqrt(1-e2*m.pow(m.sin(Bm),2)))
   RA = (M * N) / (N * m.pow(m.cos(azimut), 2) + M * m.pow(m.sin(azimut), 2))
   Dprim = m.sqrt((m.pow(Do, 2) - m.pow(deltah, 2)) / ((1 + koor1[2] / RA) * (1 + koor2[2] / RA)))
   s = 2 * RA * m.asin(Dprim / (2 * RA))
   return s - (m.pow(e2, 2) * m.pow(s, 5) * m.pow((m.cos(Bm)), 4) * m.pow((m.sin(2 * azimut)), 2)) / (360 * m.pow(a,4))
```



data = pd.read_excel('TestPodaci_dodatniZadatak.xlsx')



Slika 10 Pozivanje podataka merenja

In [306]:	dat	a													
Out[306]:		Stanica	Vizura	В	Unnamed: 3	Unnamed: 4	L	Unnamed: 6	Unnamed: 7	н	Kosa duzina	Pravac	Azimut	Ksi	Eta
	0	NaN	NaN	٥	¢	2	0	¢	2	[m]	[m]	[dec]	[dec]	2	2
	1	120.0	50.0	44	39	42.5891	20	30	29.6421	191.53	5920.97	0	268.277	1.05	3.15
	2	120.0	25.0	44	39	42.5891	20	30	29.6421	191.53	7972.15	87.5344	355.811	1.12	3.08
	3	120.0	15.0	44	39	42.5891	20	30	29.6421	191.53	4977.86	151.462	59.739	1.08	3.09
	4	15.0	120.0	44	41	4.65815	20	33	44.2184	253.37	4977.86	0	239.739	1.09	3.14
	5	15.0	50.0	44	41	4.65815	20	33	44.2184	253.37	10562.9	15.5297	255.266	1.07	3.1
	6	15.0	25.0	44	41	4.65815	20	33	44.2184	253.37	7310.42	78.3659	318.102	1.1	3.13
l	7	50.0	25.0	44	39	35.5782	20	26	1.1255	366.06	9788.73	0	33.0769	1.09	3.1
	8	50.0	120.0	44	39	35.5782	20	26	1.1255	366.06	5920.97	55.0017	88.2767	1.06	3.09
	9	25.0	50.0	44	44	0.0396	20	30	0.9958	352.66	9788.69	0	213.077	0.9	3.1
	10	25.0	120.0	44	44	0.0396	20	30	0.9958	352.66	9788.69	37.4639	175.613	0.91	3.12

```
def preprocessData(data):
    data = data.drop(0,axis=0)
    data = data.rename(columns ={data.columns[3]:'B_min','Unnamed: 4':'B_sec','Unnamed: 6':'L_min','Unnamed:
7':'L_sec',data.columns[9]:'kosa_duz'})
    data = data.astype(float)
    data[['Stanica','Vizura','B','B_min','L_min','L']] = data[['Stanica','Vizura','B','B_min','L_min','L']].astype(int)
    koordinate = data.groupby('Stanica').first().loc[:,['B','B_min','B_sec','L','L_min','L_sec','H']]
    koordinate['phi_dms'] =koordinate[['B','B_min','B_sec']].values.tolist()
    koordinate['lambda_dms'] =koordinate[['L','L_min','L_sec']].values.tolist()
    koordinate['phi']=koordinate['phi_dms'].apply(dms2degree).apply(deg2rad)
    koordinate['lambda']=koordinate['lambda_dms'].apply(dms2degree).apply(deg2rad)
    koor = koordinate[['phi','lambda','H']].copy()
    podaci_1 = data.loc[:,['Stanica','Vizura','Pravac','Azimut','Ksi','Eta','kosa_duz']].copy()
    podaci 1['azimut'] = data['Azimut'].apply(deg2rad)
    podaci 1['pravac'] = data['Pravac'].apply(deg2rad)
    podaci_1['ksi'] = (data['Ksi']/3600).apply(deg2rad)
    podaci 1['eta'] = (data['Eta']/3600).apply(deg2rad)
    merenja = podaci_1.drop(['Pravac', 'Azimut', 'Ksi', 'Eta',],axis=1).copy()
    merenja =merenja.set_index(['Stanica','Vizura'])
    merenja = merenja[['kosa duz', 'pravac', 'azimut', 'ksi', 'eta']]
    return koor, merenja
```



```
In [307]: koor
Out[307]: phi lambda H
Stanica

15 0.779894 0.358880 253.37
25 0.780744 0.357797 352.66
50 0.779462 0.356634 366.06
120 0.779496 0.357936 191.53
```



In [308]:	merenja	ì					
Out[308]:			kosa_duz	pravac	azimut	ksi	eta
	Stanica	Vizura					
		50	5920.965	0.000000	4.682312	0.000005	0.000015
	120	25	7972.146	1.527763	6.210075	0.000005	0.000015
		15	4977.861	2.643515	1.042642	0.000005	0.000015
		120	4977.864	0.000000	4.184234	0.000005	0.000015
	15	50	10562.878	0.271044	4.455231	0.000005	0.000015
		25	7310.424	1.367743	5.551930	0.000005	0.000015
	50 25	25	9788.726	0.000000	0.577302	0.000005	0.000015
		120	5920.969	0.959961	1.540720	0.000005	0.000015
		50	9788.693	0.000000	3.718894	0.000004	0.000015
		120	9788.693	0.653869	3.065025	0.000004	0.000015

```
def redukcijaMerenja(merenja,koor):
    duzine = []
    pravci = []
    for a,b in merenja.index:
        duzina = redukcijaDuzina(koor.loc[a],koor.loc[b],merenja.loc[(a,b),'kosa_duz'],merenja.loc[(a,b),'azimut'])
        pravac = redukcijaPravaca(koor.loc[a],koor.loc[b],merenja.loc[(a,b),'kosa_duz']
                                  ,merenja.loc[(a,b),'pravac'],merenja.loc[(a,b),'azimut'],
                                  merenja.loc[(a,b),'ksi'],merenja.loc[(a,b),'eta'])
        duzine.append(duzina)
        pravci.append(pravac)
    idx resenja = merenja.index
    resenja = pd.DataFrame({'Redukovane duzine':duzine,'Redukovani pravci':pravci},index =idx_resenja )
    resenja[['Stepen','Minut','Sekund']] = np.degrees(resenja['Redukovani pravci']).apply(degree2dms).values.tolist()
    index = pd.MultiIndex.from product([['Redukovani pravac'],['Stepen','Minut','Sekund']])
    izlaz = pd.DataFrame(resenja,columns = index)
    izlaz['Redukovane duzine'] = resenja['Redukovane duzine']
    izlaz['Redukovani pravac'] = resenja[['Stepen','Minut','Sekund']]
    izlaz[('Redukovani pravac', 'Stepen')] = izlaz[('Redukovani pravac', 'Stepen')].astype(int)
    izlaz[('Redukovani pravac','Minut')] = izlaz[('Redukovani pravac','Minut')].astype(int)
    return resenja.drop(['Stepen','Minut','Sekund'],axis=1),izlaz
```

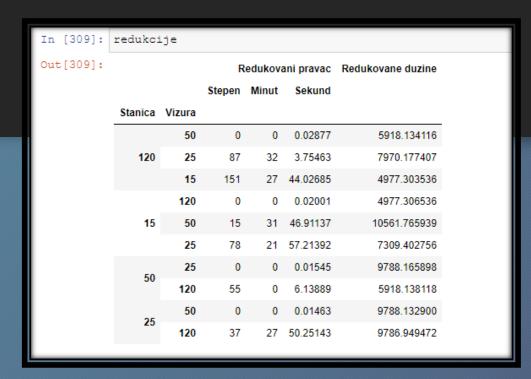


```
def horizontalniUglovi(res):
    rez= []
    for stanica, df in res.groupby(level=0):
        df = res.loc[stanica]
        for vizura in range(len(df.index)):
            for vizura_2 in range(vizura+1,len(df.index)):
                uqao= res.loc[(stanica)].iloc[vizura_2,1]-res.loc[(stanica)].iloc[vizura,1]
                rez.append([stanica,res.loc[(stanica)].iloc[vizura].name,res.loc[(stanica)].iloc[vizura 2].name,ugao])
    horizontalni_uqlovi = pd.DataFrame(rez)
    horizontalni_uglovi.columns = ['Stanica','Levi pravac','Desni pravac','Ugao']
    horizontalni_uglovi.set_index(['Stanica','Levi pravac','Desni pravac'],inplace= True)
    horizontalni_uglovi[['Stepen','Minut','Sekund']] =
np.degrees(horizontalni_uglovi['Ugao']).apply(degree2dms).values.tolist()
    horizontalni uglovi[['Stepen','Minut']]=horizontalni uglovi[['Stepen','Minut']].astype(np.int32).copy()
    return horizontalni_uglovi
```

Slika 16 Funkcija za računanje redukcije horizontalnih uglova

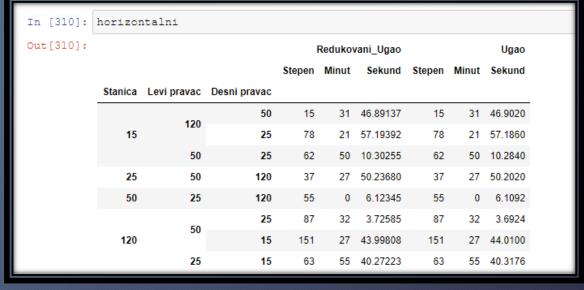






Slika 18 Tabelarni prikaz redukovanih pravaca I duzina





Slika 19 Tabelarni prikaz redukovanih I ne redukovanih horizontalnih uglova

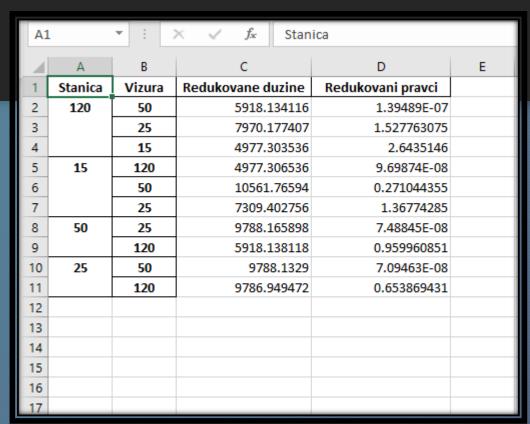


```
def resenjeZadatka(data):
    koor,merenja = preprocessData(data)
    res,izlaz = redukcijaMerenja(merenja,koor)
    hori = horizontalniUglovi(merenja)
    hori_redu = horizontalniUglovi(res)
    predj = poredjenjeHorizonta(hori,hori_redu)
    dfs = {'Koordinate tacaka':koor, 'Izvrsena merenja':merenja,'Redukovana merenje':res,'Horizontalni uglovi':predj}
    writer = pd.ExcelWriter('Izvrsene redukcije.xlsx', engine='xlsxwriter')
    for sheet_name in dfs.keys():
        dfs[sheet_name].to_excel(writer, sheet_name=sheet_name)

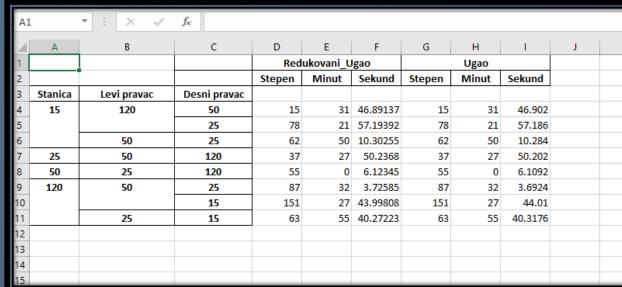
writer.save()

return koor,merenja,izlaz,predj
```

Slika 20 Funkcija za exportovanje dobijenih rešenja u exel fajl







Slika 21 | 22 Prikaz eksportovanih redukcija pravaca duzina | uglova u exel-u

3. Zaključak

