

ausgleichung.py

```
001 # Ausgleichung
002 # #####
003
004 # Authors:
005 # Joshua Wolf
006 # Silas Teske
007 # Lasse Zeh
008 # Christopher Mahn
009
010 # #####
011
012 # Import of Libraries
013 # -----
014
015 # import math as m
016 # import string as st
017 # import random as r
018 import numpy as np
019 import os
020
021
022 # -----
023 # Debugging-Settings
024
025 verbose = True # Shows more debugging information
026
027
028 # Functions
029 # -----
030
031
032 # Classes
033 # -----
034
035
036 # Beginning of the Programm
037 # -----
038
039 if __name__ == '__main__':
040
041     # Import der x-Vektor Punktbeschreibung
042     file = open(os.path.join("data", "punktnummern.txt"))
043     data = file.readlines()
044     file.close()
045     for i, e in enumerate(data):
046         data[i] = e.strip()
047
048     # Aufstellen des x-Vektors
049     x_vektor = []
050     for i in data:
051         x_vektor.append([i])
052     x_vektor = np.array(x_vektor)
053     if(verbose):
054         print(f"x_vektor:\n{x_vektor}\n")
055
056     # Import der Messwerte des Höhennetzes
057     file = open(os.path.join("data", "messungen.txt"))
058     data = file.readlines()
059     file.close()
060     for i, e in enumerate(data):
061         data[i] = e.strip().split(";")
062
063     # Aufstellen des Beobachtungsvektors
064     l_vektor = []
065     for i in data:
066         l_vektor.append([float(i[0])])
067     l_vektor = np.array(l_vektor)
068     if(verbose):
069         print(f"l_vektor:\n{l_vektor}\n")
070
071     # Aufstellen der A-Matrix
072     a_matrix = np.zeros([len(l_vektor), len(x_vektor)])
073     for i, e in enumerate(data):
074         a_matrix[i][int(e[1])] = 1
075         if(int(e[2]) != -1):
076             a_matrix[i][int(e[2])] = -1
077     if(verbose):
078         print(f"a_matrix:\n{a_matrix}\n")
079
080     # Berechnung der Redundanz
```

```

081     redundanz = len(l_vektor) - len(x_vektor)
082     if(verbose):
083         print(f"redundanz:\n{redundanz}\n")
084
085     # Berechnung des x-Vektors
086     x_vektor_berechnet = np.linalg.inv(np.transpose(a_matrix)@a_matrix)@np.transpose(a_matrix)@l_vektor
087     if(verbose):
088         print(f"x_vektor_berechnet:\n{x_vektor_berechnet}\n")
089
090     # Normalisieren des x-Vektors
091     average = np.mean(x_vektor_berechnet[0:49])
092     x_vektor_normalisiert = x_vektor_berechnet[0:49] - average
093     print(x_vektor_normalisiert)
094
095     # Export
096     file = open(os.path.join("data", "export.txt"), "w")
097     for i in x_vektor_normalisiert:
098         i = float(i[0])
099         file.writelines(f"{i:+.6f}\n")
100     file.close()

```