

POLITECHNIKA WARSZAWSKA

Wydział Geodezji i Kartografii
Kierunek Geodezja i Kartografia

Informatyka geodezyjna

Projekt numer 1

Prowadzący:
mgr inż. Andrzej Szeszko

Autorzy:
Dominik Sawczuk 319372
Karol Pawłowski 319354
Grupa 3b

Warszawa 2023

Spis treści

1. Cel ćwiczenia	3
2. Wymagania systemowe	3
3. Przebieg ćwiczenia	3
3.1. Charakterystyka funkcji	3
3.2. Charakterystyka elipsoid	4
3.3. Przebieg	4
4. Wnioski	4
Literatura	5

1. Cel ćwiczenia

Program Transformacje.py ma na celu przeprowadzenie transformacji pomiędzy układami współrzędnych takimi jak:

- **XYZ** → **BLH**
- **BLH** → **XYZ**
- **XYZ** → **NEU**
- **BL** → **XY2000**
- **BL** → **XY1992**

Ponadto program powinien obsługiwać elipsoidy:

- **GRS80**
- **WGS84**
- **Krasowskiego**

2. Wymagania systemowe

W celu implementacji transformacji zalecane jest wykorzystanie systemu operacyjnego Windows 11 oraz oprogramowania python v3.10 wraz z zainstalowanymi bibliotekami takimi jak:

- **numpy** – dodaje obsługę wielowymiarowych tablic i macierzy oraz dostarcza wiele funkcji matematycznych i operacji algebraicznych, które pozwalają na efektywne wykonywanie operacji na dużych zbiorach danych numerycznych. Wykorzystując tę bibliotekę uzyskujemy schludniejszy kod oraz precyzyjne wyniki [1]
- **argparse** – umożliwia stworzenie przyjaznych dla użytkownika interfejsów wiersza poleceń. Program określa, jakich argumentów potrzebuje, a *argparse* je przeanalizuje. Moduł *argparse* automatycznie generuje komunikaty pomocy i użytkownika. [2]
- **math** – dostarcza zestaw funkcji matematycznych w języku Python, umożliwiających przeprowadzanie różnych obliczeń numerycznych. Biblioteka ta jest dostępna w standardowej instalacji języka Python i nie wymaga żadnych dodatkowych instalacji. [3]

3. Przebieg ćwiczenia

3.1. Charakterystyka funkcji

- **XYZ** → **BLH** – funkcja transformująca współrzędne kartezjańskie **XYZ** na elipsoidzie do współrzędnych geocentrycznych **BLH**. [8]
- **BLH** → **XYZ** – funkcja transformująca współrzędne geocentryczne **BLH** do współrzędnych kartezjańskich **XYZ** na elipsoidzie. [8]
- **XYZ** → **NEU** – funkcja transformująca współrzędne kartezjańskie **XYZ** na elipsoidzie do macierzy obrotu **NEU** [7]
- **BL** → **XY2000** – funkcja transformująca współrzędne geocentryczne **BLH** do współrzędnych **XY** Gausa Krygera, a następnie otrzymane współrzędne przelicza do współrzędnych **XY** w układzie **pl2000** [7]
- **BL** → **XY1992** – funkcja transformująca współrzędne geocentryczne **BLH** do współrzędnych **XY** Gausa Krygera, a następnie otrzymane współrzędne przelicza do współrzędnych **XY** w układzie **pl1992** [7]

3.2. Charakterystyka elipsoid

1. **GRS80** – jeden z najczęściej używanych modeli geodezyjnych Ziemi. Elipsoida GRS80 jest wykorzystywana do określania dokładnego kształtu Ziemi oraz do dokładnego obliczania odległości, powierzchni i objętości na jej powierzchni. Jest to elipsoida obrotowa, czyli jej kształt jest uzyskiwany przez obrót elipsy wokół jednej z osi.

Podstawowe parametry elipsoidy

— $a = 6378137\text{m}$

— $b = 6356752.31414036\text{m}$

[4]

2. **WGS84** – jeden z najczęściej używanych modeli geodezyjnych Ziemi. Elipsoida ta została opracowana w celu zapewnienia globalnej jednolitości systemu pozycjonowania GPS oraz umożliwienia spójnego modelowania geoidy na całym świecie. Elipsoida WGS84 jest elipsoidą obrotową, co oznacza, że jej osie nie są prostopadłe do siebie, ale są przesunięte względem siebie. Elipsoida ta jest uważana za dokładną reprezentację kształtu Ziemi, co pozwala na precyzyjne określenie położenia punktów na powierzchni Ziemi.

Podstawowe parametry elipsoidy:

— $a = 6378137\text{m}$

— $b = 6356752.31424518\text{m}$

[5]

3. **Krasowskiego** – jeden z modeli geoidy opisujący kształt Ziemi. Elipsoida Krasowskiego jest elipsoidą obrotową, co oznacza, że jej kształt jest uzyskiwany poprzez obrót elipsy wokół jednej z jej osi. Elipsoida ta była wykorzystywana w Polsce jako standardowy model geoidy w geodezji i kartografii do lat 90. XX wieku, ale obecnie została zastąpiona nowszymi modelami.

Podstawowe parametry elipsoidy:

— $a = 6378245\text{m}$

— $b = 6356863.019\text{m}$

[6]

3.3. Przebieg

Ćwiczenie należało rozpocząć od utworzenia zdalnego repozytorium przez jednego członka z grupy i udostępnienia dostępu do repozytorium drugiemu członkowi. Na repozytorium zostały utworzone trzy gałęzie odpowiadające elementom projektu. Gałąź **README** i **LaTeX** zostały utworzone w celu umieszczania na nich odpowiednio pliku README i sprawozdania w LaTeX. Na gałęzi **main** została utworzona klasa *Transformacje* zawierająca implementację. Implementację zostały utworzone w oparciu o funkcję wykonującą transformację z przedmiotu Geodezja Wyższa w 3 semestrze. Przy pomocy metody *init* zostały wykonane przeliczenia współrzędnych z różnych modeli elipsoid, których zmienne zostały zapisane podczas użycia odwołania *self* w taki sposób aby implementację mogły wykorzystać parametry elipsoidy wybranej przez użytkownika. W ćwiczeniu zastosowano wyrażenie warunkowe *if __name__ == "__main__"* zapewniającą większą czytelność kodu, oraz biblioteka *argparse*, która umożliwiła użytkownikowi wywołanie programu z poziomu terminala.

Program został stworzony przy wykorzystaniu oprogramowania Spyder 5.4.3 w sposób umożliwiający zaimportowanie i zapis pliku w formacie *.txt*. Program został napisany oraz sprawdzony na urządzeniu obsługującym system operacyjny Windows 11, za pomocą języka python w wersji 3.10, w związku z brakiem możliwości przeprowadzenia testów oprogramowania na urządzeniach posiadających starszą wersję systemu operacyjnego nie byliśmy w stanie stwierdzić zgodności z takowymi systemami. Przy pomocy klauzuli *except* uniemożliwiliśmy uruchomienie programu dla przypadków podania przez użytkownika danych w złym zapisie, wybrania błędnego modelu elipsoidy oraz błędnej metody. W chwili, gdy użytkownik popełni któryś z wymienionych błędów, zostanie on automatycznie poinformowany na którym etapie go wykonał.

4. Wnioski

Praca nad projektem nauczyła nas korzystania ze zdalnego repozytorium *GitHub*, dzięki któremu praca nad kodem stała się sprawniejsza. Mogliśmy podzielić obowiązki w grupie, co przyspieszyło tempo i efektywność

pracy, ponieważ każda edycja pliku była widoczna. Dodatkowo nauczyliśmy się obsługi biblioteki *argparse*. Ponadto udało nam się zwiększyć umiejętności w:

- pisaniu kodu obiektowego w Pythonie
- implementowaniu algorytmów pochodzących ze źródeł zewnętrznych
- tworzeniu dokumentów w latex
- tworzeniu narzędzi w interfejsie tekstowym (cli) potrafiących przyjmować argumenty przy wywołaniu
- pisaniu użytecznej dokumentacji

Link do repozytorium

<https://github.com/Grabarzd/Projekt1>

Literatura

- [1] <https://numpy.org>.
- [2] <https://docs.python.org/pl/3.6/library/argparse.html>.
- [3] <https://docs.python.org/pl/3.6/library/math.html>.
- [4] https://en.wikipedia.org/wiki/Geodetic_Reference_System_1980.
- [5] https://en.wikipedia.org/wiki/World_Geodetic_System.
- [6] http://uriasz.am.szczecin.pl/naw_bezp/elipsoida.html#_Toc66971422.
- [7] ASG EUPOS. Strona systemu ASG-EUPOS. www.asgeupos.pl, 2021.
- [8] prof. dr hab. inż. Roman Kadał Akademia Rolnicza w Krakowie. Strona systemu ASG-EUPOS. https://ewmapa.pl/dane/wytyczne_g-1.10.pdf, 2001.