**Cel ćwiczenia:**

Stworzenie skryptu implementującego następujące transformacje:

* (XYZ TO flh) Transformacja współrzędnych geocentrycznych X, Y, Z na współrzędne elipsoidalne (φ, λ, h) fi, lambda, h.
* (flh TO XYZ) Transformacja współrzędnych elipsoidalnych (φ, λ, h) fi, lambda, h na współrzędne geocentryczne X, Y, Z.
* (XYZ TO NEU) Transformacja współrzędnych X,Y,Z do układu topocentrycznego Northing, Easting, Up.
* (FL [GRS80, WGS84, ew. Krasowski] to 2000) Transformacje współrzędnych elipsoidalnych () fi, lambda z systemów odniesienia GRS80, WGS84, ew. Krasowski do układy PL2000.
* (FL [GRS80, WGS84, ew. Krasowski] to 1992) Transformacje współrzędnych elipsoidalnych () fi, lambda z systemów odniesienia GRS80, WGS84, ew. Krasowski do układy PL1992.

**Wykorzystane narzędzia i materiały potrzebne do replikacji ćwiczenia:**

Podczas wykonania ćwiczenia korzystaliśmy z takich programów: Spyder 5.4.3 (Anakonda3), Python 3.10.11, Git BASH, TeXstudio (LaTEX editor), JabRef, wyszukiwarka Google, SO Windows 11.

Wykorzystując takie materiały pomocnicze: udostępnione przez prowadzącego zajęcia linki, własne napisane wcześniej kody, przykłady kodów udostępnione na innych zajęciach, materiały z ćwiczeń z Informatyki I i Geodezji Wyższej I.

**Przebieg ćwiczenia:**

Zdecydowałyśmy pisać skrypt w środowisku programistycznym Spyder (ostatniej mogła być aktualizowana w trakcie wykonania ćwiczenia) dla języka Python, które jest dostępne w pakiecie Anaconda, ponieważ oferuje wiele funkcji, takich jak edytor kodu z podświetlaniem składni, przeglądarkę zmiennych, system pomocy oraz wiele innych, które ułatwiają tworzenie i testowanie programów w języku Python.

Następnym krokiem było zaimportowanie biblioteki NumPy. Głównymi zaletej której są łatwe wykonywanie operacji arytmetycznych, algebraicznych, logicznych i innych operacji matematycznych na wielowymiarowych tablicach oraz łatwe indeksowanie i przetwarzanie danych, co ułatwia operacje na dużych zbiorach danych.

Na początek napisałam dwie definicje (funkcje) **XYZ\_to\_flh** i **flh\_to\_XYZ**, korzystając z prezentacji Pana Dominika Próchniewicza, w której był podany algorytm obliczeniowy. Następnie dodałam definicje do klasy „Transformacje” (class Transformacje).

Funkcja XYZ\_to\_flh wykonuje transformację ze współrzędnych kartezjańskich (XYZ) na elipsoidę geodezyjną (flh). Wewnątrz funkcji, a i e2 reprezentują wielkość i spłaszczenie elipsoidy geodezyjnej, p jest odległością punktu od osi obrotu elipsoidy, f jest szerokością geodezyjną, a h jest wysokością geodezyjną. Funkcja wykorzystuje iteracyjny algorytm do rozwiązania równań geodezyjnych i obliczenia wartości f i h dla danych wartości X, Y i Z. Funkcja flh\_to\_XYZ wykonuje transformację z elipsoidy geodezyjnej (flh) na współrzędne kartezjańskie (XYZ). Wewnątrz funkcji, a i e2 reprezentują wielkość i spłaszczenie elipsoidy geodezyjnej, N reprezentuje promień krzywizny normalnej w danym punkcie, a X, Y i Z reprezentują współrzędne kartezjańskie. Funkcja wykorzystuje wzory do obliczenia wartości X, Y i Z dla danych wartości f, l i h.

Po dodaniu do klasy postanowiłam sprawdzić poprawne działanie kodu. Dla tego zaimportowałam bibliotekę argparse (jak było podane w zagadnieniu do ćwiczenia) i utworzyłam skrypt, który akceptuje argumenty z wiersza poleceń i zwraca wynik transformacji. Kod sprawdza, czy skrypt jest uruchamiany jako plik główny (czyli bezpośrednio, a nie importowany jako moduł) poprzez sprawdzenie, czy name == 'main'. Następnie tworzony jest obiekt ArgumentParser, który pozwala zdefiniować argumenty wiersza poleceń. Argumenty te są zdefiniowane jako podprogramy, w tym przypadku 'XYZ' i 'flh'. Każdy podprogram definiuje argumenty, które są wymagane dla danej transformacji. Następnie funkcja parse\_args() jest wywoływana, aby przetworzyć argumenty wiersza poleceń. W zależności od wybranej operacji, transformacja jest wykonywana przez obiekt Transformacje, a wynik jest drukowany na ekranie.

Po napisaniu tej części kodu zauważałam, że jest domyślnie ustawiony branch „main”, więc utworzyłam branch „master”. Branch „main” nie został usunięty, co może być mylące, ponieważ reszta commitów będzie się znajdowała na innej gałęzi.

Żeby wywołać funkcję w wierszu poleceń bezpośrednio przez polecenie „python”, trzeba mieć ustawione środowisko zmienne.

.

.

.

Następnie została dopisana pomocnicza funkcja, która przelicza wartość z radianów na stopnie, minuty, sekundy.

Do napisanych definicji została dodana klauzula \_\_name\_\_, na przykładzie tego samego algorytmu.

Korzystając z wymienionych wcześniej materiałów, została dodana funkcja XYZ\_to\_neu, która przyjmuje cztery argumenty: dX, X, Y i Z. Funkcja wykorzystuje te argumenty do obliczenia macierzy transformacji R, która przekształca wektor przesunięcia dX, wyrażony w układzie współrzędnych kartezjańskich z punktem początkowym w (X, Y, Z), na lokalny układ współrzędnych płaszczyzny stycznej zdefiniowany przez wektor normalny w punkcie (X, Y, Z). I dopisanie do tej funkcji części \_\_name\_\_.

.

.

.

Następnym krokiem było napisanie dokumentacji do kodu w pliku README.md, który będzie się wyświetlał na GitHub. Jest to plik tekstowy, który zawiera podstawowe informacje na temat projektu. W naszym przypadku to: nazwa, opis metod transformacji (do czego służy, dane wejściowe i wyjściowe) , sposób instalacji, przykłady używania funkcji i inne ważne informacje, które mogą być przydatne dla użytkowników.

Przy napisaniu wystąpił problem ze śledzeniem plików, ponieważ dodałam do repozytorium plik .png, który wykorzystałam w dokumentacji. Przy rozwiązaniu tego problemu stworzył się w śledzonym folderze clon naszego repozytorium, co powodowało zapisywanie zmian do nowego pliku. Za pomocą prowadzącego ten problem został rozwiązany, w wyniku czego nie było negatywnych skutków.

.

.

.

Żeby pisać sprawozdanie z projektu przy pomocy programu LaTeX był stworzony pomocniczy plik, zawierający informacje na temat zadania, czyli cel ćwiczenia, przebieg i tp. Ponieważ nie miałyśmy jeszcze do czynienia z tym programem i chciałyśmy widzieć, jak musi wyglądać ostateczny plik.

.

.

.