



Wydział Geodezji i Kartografii

POLITECHNIKA WARSZAWSKA

TRANSFORMACJE MIĘDZY RÓŻNYMI UKŁADAMI GEODEZYJNYMI

INFORMATYKA GEODEZYJNA II
SEM. IV, ĆWICZENIA, ROK AKAD. 2023-2024

ALICJA ŁUBIANKA

NUMER INDEKSU: 325786, GRUPA 3

E-MAIL: 01179176@PW.EDU.PL

MAGDALENA STERNIK

NUMER INDEKSU: 325835, GRUPA 3

E-MAIL: 01179227@PW.EDU.PL

WYDZIAŁ GEODEZJI I KARTOGRAFII, POLITECHNIKA WARSZAWSKA
Warszawa, 12 maja 2024

Spis treści

1	Cel ćwiczenia	3
2	Wykorzystane narzędzia i materiały potrzebne do replikacji ćwiczenia	3
2.1	Wybrany język programowania i interpreter Spyder	3
2.2	System operacyjny	3
2.3	Potrzebne biblioteki i pliki	3
3	Przebieg ćwiczenia	3
3.1	Utworzenie klasy Transformacja	3
3.2	Algorytm hirvonena	4
3.3	flh2XYZ	4
3.4	flh2PL1992	4
3.5	flh2PL2000	4
3.6	xyz2neu	4
3.7	Wczytywanie i zapisywanie pliku	5
3.8	Kalkulator transformacji i zapis ich wyników do Kalkulatora	5
3.9	Dodanie możliwości wczytania pliku w argparse	5
4	Podsumowanie	5
4.1	Rezultaty	5
4.2	Umiejętności nabyte	5
4.3	Spostrzerzenia, problemy i ich rozwiązania:	6
5	Literatura	7

1 Cel ćwiczenia

W ramach ćwiczenia opracowano skrypt w języku Python w postaci klasy zawierającej metody służące do transformacji współrzędnych pomiędzy układem kartezjańskim (x, y, z) a geodezyjnym (ϕ, λ, H). Aby dokonać transformacji pomiędzy układami należy zastosować odpowiednie algorytmy. Poniżej przedstawiona jest lista utworzonych algorytmów:

- XYZ (geocentryczne) -> BLH (elipsoidalne ϕ , λ , h)
- BLH -> XYZ
- XYZ -> NEU (topocentryczne northing, easting, up)
- BL (GRS80, WGS84, ew. Krasowski) -> PL2000
- BL (GRS80, WGS84, ew. Krasowski) -> PL1992

2 Wykorzystane narzędzia i materiały potrzebne do replikacji ćwiczenia

2.1 Wybrany język programowania i interpreter Spyder

Do napisania skryptu tego ćwiczenia posłużył nam język programowania Python, a za środowisko odpowiadał Spyder zawierający edytor kodu, interpreter, konsolę, a także inne funkcje.

2.2 System operacyjny

Plik został utworzony w systemie operacyjnym Microsoft (Windows 11).

2.3 Potrzebne biblioteki i pliki

Do wykonania ćwiczenia należy użyć następujących bibliotek:

- Numpy - biblioteka w języku Python służąca do obliczeń numerycznych i analizy danych. Zapewnia narzędzia do pracy z wielowymiarowymi tablicami danych oraz operacji matematycznych i statystycznych na tych tablicach. Numpy nie jest wbudowany w Pythona, lecz jest dostarczany z Anacondą, co ułatwia jego dostępność.
- Argparse - biblioteka w języku Python do parsowania argumentów linii poleceń. Jest częścią standardowej biblioteki Pythona, co oznacza, że jest wbudowana w standardową instalację Anacondy.
- Os - biblioteka standardowa w języku Python zapewniająca interfejs do operacji na systemie operacyjnym, takich jak dostęp do plików, zarządzanie procesami, zmiana katalogu roboczego, itp.

Należy pobrać plik tekstowy o nazwie "input_file.txt", który znajduje się na naszym zdalnym repozytorium GitHub. Umożliwi nam on wczytanie oraz wykonanie transformacji z zawierającymi danymi. Kolejno zapisze te dane do pliku wyjściowego.

3 Przebieg ćwiczenia

3.1 Utworzenie klasy Transformacja

Utworzono klasę Transformację oraz konstruktor `__init__`, w którym przekazywane są parametry elipsoidy (a , e^2), niezbędne do wykonywania obliczeń. Ważne, aby pamiętać, gdy się odwołujemy do powyższych parametrów, musimy zastosować `self`. Dodatkowo, pokazano sytuację, gdy podana zostanie nieprawidłowa nazwa elipsoidy, co skutkuje wyświetleniem błędu.

3.2 Algorytm hirvonena

Algorytm Hirvonena przekształca współrzędne kartezjańskie (X,Y,Z) na współrzędne geodezyjne (ϕ , λ , H). W funkcji implementującej ten algorytm, używana jest pętla while, która wykonuje odpowiednią liczbę iteracji, aby uzyskać dokładność na poziomie 1 mm. Dodatkowo zaimplementowany został warunek „output” za pomocą instrukcji warunkowych elif, if, else, pozwalający na wybór formatu wyniku. Do jednej z metod przekształceniowej (stopnie, minuty sekundy) powstał dodatkowy algorytm „dms”. Prócz tego utworzono funkcję „np”, obliczającą promień przekroju w pierwszym wertykale. Wyniki działania transformacji zostały zweryfikowane z wynikami uzyskanymi w zaliczonym sprawozdaniu z poprzedniego semestru.

3.3 flh2XYZ

Transformacja flh konwertuje współrzędne geodezyjne (ϕ , λ , h) na współrzędne kartezjańskie (x,y,z). W implementacji tej transformacji zastosowano trzy wzory, z których każdy odpowiada jednej ze współrzędnych kartezjańskich. Dodatkowo wykorzystana została funkcja „Np”. Wyniki działania transformacji zostały zweryfikowane z wynikami uzyskanymi w zaliczonym sprawozdaniu z poprzedniego semestru.

3.4 flh2PL1992

Transformacja fl21992 przekształca współrzędne geodezyjne (ϕ , λ) na współrzędne w układzie 1992 (x1992, y1992). W implementacji tej transformacji dodano warunek ograniczający współrzędne ϕ i λ do obszaru wyłącz- nie Polski, aby uniknąć błędów. Następnie obliczono wartości x1992, y1992 przy użyciu odpowiednich wzorów. Wyniki działania transformacji zostały zweryfikowane z wynikami uzyskanymi w zaliczonym sprawozdaniu z poprzedniego semestru.

3.5 flh2PL2000

Transformacja fl2PL2000 konwertuje współrzędne geodezyjne (ϕ , λ) na współrzędne w układzie 2000 (x2000, y2000). Implementacja tej transformacji przebiega niemal identycznie jak transformacja fl2PL1992, z wyjątkiem zmiany skali oraz ustaleniem odpowiedniej strefy (5,6,7,8) dla naszych danych. Po zamianie współrzędnych otrzymano x2000, y2000. Wyniki działania transformacji zostały zweryfikowane z wynikami uzyskanymi w zaliczonym sprawozdaniu z poprzedniego semestru.

3.6 xyz2neu

Transformacja xyz2neu przekształca współrzędne kartezjańskie (x, y, z) do układu neu. W implementacji tej transformacji stworzono trzy definicje. Pierwsza dotyczyła macierzy obrotu (renu), druga obliczała macierz różnic między dwoma punktami, a trzecia obliczała macierz neu. Pierwsza kolumna (odpowiadała współrzędnej n, druga współrzędnej e, a trzecia u. Wyniki działania transformacji zostały zweryfikowane z wynikami uzyskanymi w zaliczonym sprawozdaniu z poprzedniego semestru.

3.7 Wczytywanie i zapisywanie pliku

Aby odczytać i zapisać plik stworzono jedną funkcję. Służy ona do odczytu pliku w formacie txt oraz transformuje zmienne głównie do stringów, aby wszystkie miały taką samą długość. Do tego celu użyto funkcji zamian float na string:

- f2s;
- f2s_fl;
- f2s_rad,

które działają na tej samej zasadzie, czyli dodają spację przed liczbą za pomocą pętli while, a warunek kończy się, gdy string osiągnie odpowiednią długość. Nasza funkcja ostatecznie zapisuje plik w postaci tabelki z nagłówkiem. Dla punktów spoza Polski w wynikach (x1992, y1992, x2000, y2000) stosowane są myślniki '-'.

3.8 Kalkulator transformacji i zapis ich wyników do Kalkulatora

W celu wykonania transformacji współrzędnych do różnych układów, został stworzony kalkulator o nazwie „kalkulator_xyz2reszta.py”. Korzysta z importowanych transformacji z pliku głównego oraz biblioteki argparse, wykorzystując ArgumentParser. Ma to na celu możliwość podawania przez użytkownika współrzędnych geodezyjnych (ϕ , λ , h) oraz elipsoidy. Następnie są liczone do układu kartezjańskiego (x, y, z), PL1992 i PL2000.

W pliku głównym skrypt.py stworzyliśmy specjalne funkcje, które zapisują wyniki kalkulatora. W nich przeprowadzono niezbędne operacje na zmiennych, na przykład zamienienie z typu float na string, zachowując odpowiednią liczbę miejsc po przecinku. Celem było to, aby podany plik wynikowy miał odpowiednią postać. Również użyto biblioteki os, aby sprawdzić, czy dany plik już znajduje się na naszym komputerze. Dzięki temu nie jest tworzony nowy dokument z nagłówkami, a do istniejącego dokumentu dodawane są kolejne linijki.

3.9 Dodanie możliwości wczytania pliku w argparse

Dodatkowo w głównym pliku dodaliśmy funkcję umożliwiającą wczytywanie i zapisanie pliku wynikowego za pomocą biblioteki argparse (możliwość wczytania i zapisania pliku z cmd). Dzięki temu użytkownik może teraz użytkownik może teraz wczytać plik z wiersza poleceń. WAŻNE! - aby to zadziałało należy pamiętać, żeby wczytywany plik miał odpowiednią formę taką jak plik "input_file.txt (znajduję się on na naszym zdalnym repozytorium GitHub).

4 Podsumowanie

4.1 Rezultaty

Poniżej znajduje się link do naszego zdalnego repozytorium GitHub:

https://github.com/AlicjaLubianka/projekt_1.git

W repozytorium znajduje się:

- Skrypt.py – główny plik, w którym mieszczą się wszystkie algorytmy transformujące współrzędne oraz wywołanie przykładowego pliku txt wraz z zapisem do pliku o nazwie wyniki.txt
- input_.file.txt - zawiera przykładowe dane, które możemy przeliczyć.
- kalkulator_xyz2reszta.py - plik importujący bibliotekę argparse. Za pomocą wiersza poleceń (cmd) podać dane do obliczenia.

4.2 Umiejętności nabyte

- Efektywne tworzenie plików tekstowym w LaTeX
- Poprawa umiejętności pisania kodu w Pythonie oraz bliższe poznanie wielu bibliotek tj. argparse, os, numpy
- Zdobywanie doświadczenia w pracy zespołowej, dzięki platformie GitHub
- Nauka pisania pliku README, który jest ważnym elementem każdego projektu lub repozytorium

4.3 Spostrzerzenia, problemy i ich rozwiązania:

Spostrzeżenia i trudności napotkane wraz z tworzeniem ćwiczenia:

- Program nie jest kompletny i idealny, ponieważ te algorytmy można na wiele przypadków rozpatrywać, dodawać warunki, czy nowe definicje, jednak wybiegałoby to daleko, jeśli chodzi o zakres projektu i prawdopodobnie nasze umiejętności.
- Początkowym problemem było posługiwanie się platformą GitHub, która była dosyć dużą nowością dla nas. Podczas korzystania z tego portalu napotkaliśmy wiele problemów, bądź konfliktów, które dzięki materiałom z zajęć oraz dodatkowym źródłom informacji udało nam się zażegnać. Po dłuższym użytkowaniu możemy powiedzieć, że usprawnia to proces tworzenia pracy grupowej, każdy użytkownik ma zawsze zaktualizowaną wersję, a prace są w jednym folderze.
- Nie wszystko potrafiliśmy przewidzieć w trakcie tworzenia koncepcji pracy. Dodatkowo musieliśmy implementować warunki czy nowe definicje, by nasza praca była bardziej przejrzysta, bądź wskazywała dobre wyniki. Przykładowo wystąpił błąd przy utworzeniu definicji PL2000 I PL 1992 dla (ϕ, λ) nie leżącego na terenie Polski, stąd musieliśmy dodać warunki dla współrzędnych, aby niespełnione kryteria zostały uwzględnione w notatniku jako myślnik.
- Mialiśmy również problem, z przetestowaniem naszego kalkulatora w wierszu poleceń, jednak po analizie notatek z zajęć Informatyka Geodezyjna oraz pomocy Internetu udało nam się pomyślnie przeprowadzić kontrolę naszego programu.

5 Literatura

- Borkowski, Przybylski (2015), Książka kucharska LaTeX, Springer Wien New York
- Roman J. Kadaj(2002), Polskie układy współrzędnych polskie układy współrzędnych polskie układy współrzędnych formuły transformacyjne, algorytmy i programy http://www.geonet.net.pl/images/2002_12_uklady_wspolrz.pdf
- Paweł Chaniewski(2018), Python argparse - przekazywanie parametrów (argumentów) wiersza poleceń <https://cws1.pl/python/tutorial/python-argparse-przekazywanie-parametrow-argumentow-uruchomieniowy?fbclid=IwAR0ctjScWQ55Jeyhv17gVSP01rGLeeKYItZFLiZINUfwEhES1Bn00NEaf3E>
- Vincent Stevenson(2021), Command Line Parsing Arguments in Python with Argparse - Intro and Demo, https://www.youtube.com/watch?v=53H_082uqfY