Sprawozdanie z laboratorium 3

Temat: Triangulacja wielokąta monotonicznego

Data: 02.12.2023

Algorytmy geometryczne

Wojciech Kaźmierczak

Nr albumu: 416692

Gr. 4, czwartek 11:20, tydzień A

Informatyka, Wydział Informatyki

AGH UST

Procesor: AMD Ryzen 7 5700U 8 rdzeni

RAM: 16GB

System: Windows 11 Home

Użyty język: Python, Biblioteki: numpy, pandas, matplotlib

Środowisko: Jupyter Notebook

1. Opis ćwiczenia

Celem ćwiczenia było sprawdzenie czy dany wielokąt jest y-monotoniczny, zaimplementowanie algorytmu przyporządkowującego dany punkt do podgrupy o danych cechach (opisane później), implementacja funkcji pozwalającej na tworzenie wielokątów za pomocą myszki oraz zaimplementowanie algorytmu realizującego triangulacje wielokąta y-monotonicznego oraz graficzne przedstawienie działania owego algorytmu.

2. Porządkowanie punktów według kategorii

W nawiasach zapisane kolory jakimi zaznaczono wierzchołki należące do danej kategorii

Wierzchołki naszego wielokąta możemy podzielić na parę kategorii:

- **początkowe (zielony)**, gdy obaj jego sąsiedzi leżą poniżej i kąt wewnętrzny ma mniej niż 180 stopni, to wierzchołki, w których wielokąt zaczyna się monotoniczny spadek

- **końcowe (czerwony)**, gdy obaj jego sąsiedzi leżą powyżej i kąt wewnętrzny ma mniej niż 180 stopni. To wierzchołki, w których monotoniczność wielokąta się zmienia, czyli na przykład zaczyna się monotoniczny wzrost, jeśli wcześniej był spadek, lub na odwrót.

Wierzchołki startowe i końcowe są ważne w kontekście algorytmów przetwarzania wielokątów monotonicznych, takich jak algorytmy dziel i zwyciężaj oraz triangulacji.

- **dzielący (błękitny)**, gdy obaj jego sąsiedzi leżą poniżej i kąt wewnętrzny ma więcej niż 180 stopni. To wierzchołki, które wyznaczają przekątne (linie łączące), tworzące trójkąty podczas triangulacji.

- **łączący (ciemno niebieski)**, gdy obaj jego sąsiedzi leżą powyżej i kąt wewnętrzny ma więcej niż 180 stopni. To wierzchołki, które są połączone liniami (przekątnymi) wewnątrz wielokąta, tworząc trójkąty.

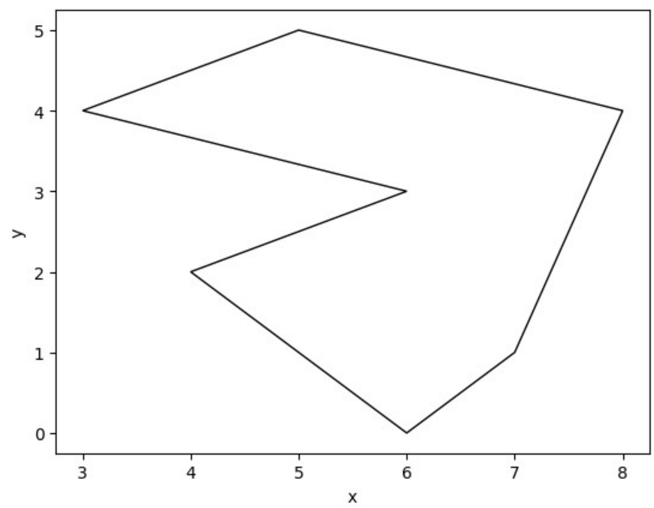
Wierzchołki **łączące** i **dzielące** odgrywają kluczową rolę w procesie triangulacji wielokątów, pozwalając na podział figury na trójkąty w sposób bezkolizyjny.

- **prawidłowy (brązowy)**, pozostałe przypadki, jeden sąsiad powyżej drugi poniżej

3. Wykonanie ćwiczenia

3.1. Sprawdzenie czy y-monotoniczny

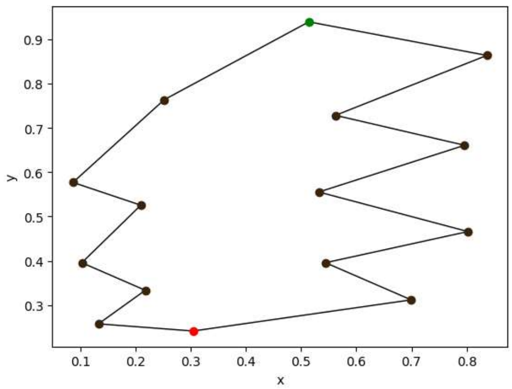
W celu sprawdzenia czy wielokąt jest monotoniczny szukamy punktu o najmniejszej wartości współrzędnej y oraz punktu o największej współrzędnej y. Wędrując po punktach zgodnie z ruchem wskazówek zegara sprawdzamy czy każdy kolejny ma współrzędną y większą od poprzednika, aż do momentu gdy napotkamy wierzchołek o największym y. Wtedy sprawdzamy czy każdy kolejny wierzchołek ma współrzędną mniejszą od poprzednika. Jeśli nie będzie sytuacji przeciwnej to po wykonaniu całego „kółka” po punktach wiem, że dany wielobok jest monotoniczny.



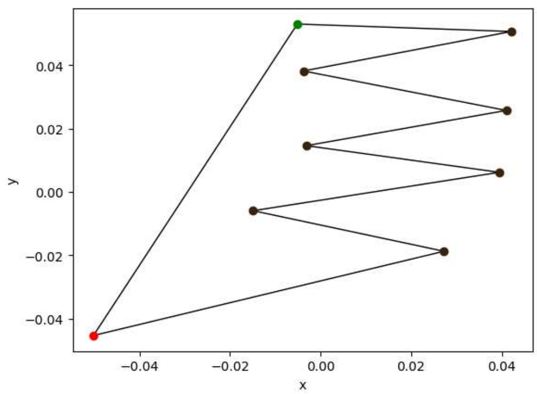
Wykres 1 Przykład wieloboku monotonicznego

3.2. Sklasyfikowanie punktów do kategorii opisanych w punkcie 2.

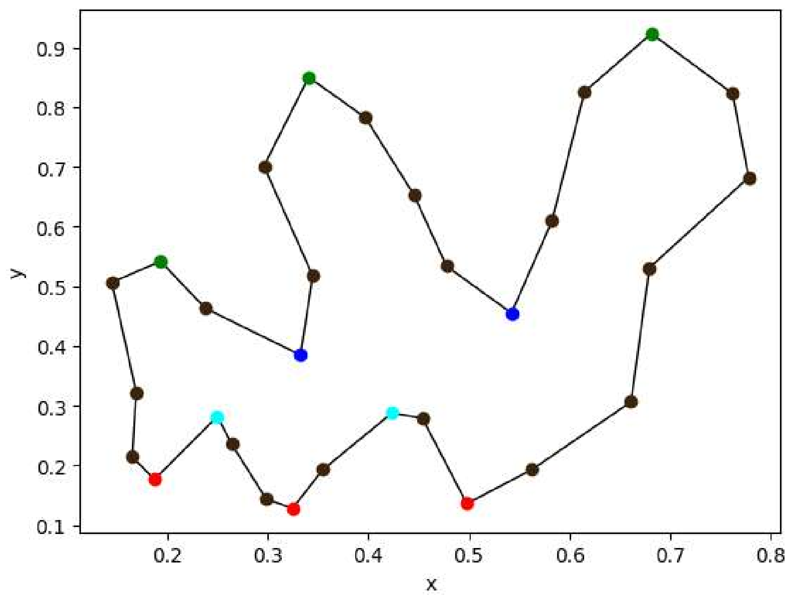
Punkty klasyfikowane są według kategorii opisanych w punkcie 2. do weryfikacji kąta wykorzystany jest wyznacznik 2x2 własnoręcznie zaimplementowany przyjęty.



Wykres 2 Przykładowa figura



Wykres 3 Przykładowa figura



Wykres 4 Przykładowa figura

3.3. Implementacja funkcji do tworzenia wielokątów myszką

Do tego celu użyta została biblioteka matplotlib oraz funkcja ginput() pozwala ona na dobieranie dowolnych punktów, które podawane przeciwnie do ruchu wskazówek zegara pozwalają na utworzenie wielokąta. Rezultat zapisywane jest jako lista punktów w postaci współrzędnych.

3.4. Triangulacja

Triangulację trójkąta monotonicznego można opisać w następujących krokach:

- Sprawdzenie czy wielokąt y-monotoniczny, gdyż tylko dla takich nasza implementacja działa

- Wyznaczenie lewego i prawego łańcucha wielokąta względem kierunku monotoniczności. W naszym przypadku do prawego łańcucha należy ten z najmniejszą współrzędną y oraz każdy kolejny znajdujący się powyżej poprzednika, natomiast punkt o największej współrzędnej y jest pierwszym punktem lewego łańcucha i każdy kolejny następujący po nim względem kierunku monotoniczności też należy do łańcucha, aż do wierzchołka początkowego.

- Porządkujemy wierzchołki wzdłuż kierunku monotoniczności zaczynając od tego z najmniejszą wartością y.

- Wkładamy dwa pierwsze wierzchołki na stos.

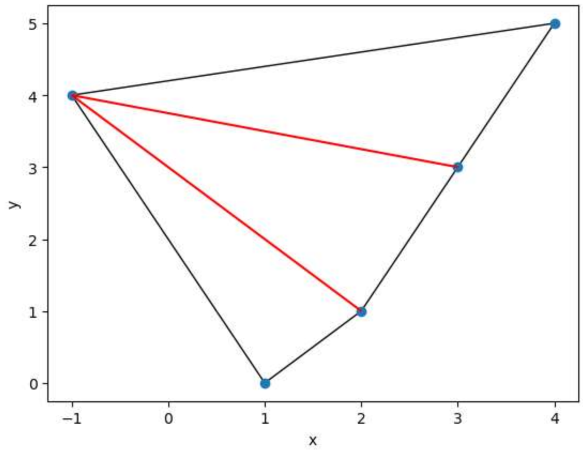
- Jeśli kolejny wierzchołek należy do innego łańcucha niż wierzchołek stanowiący szczyt stosu, to możemy go połączyć ze wszystkimi wierzchołkami na stosie. Na stosie zostają dwa wierzchołki, które były „zamiatane” ostatnie. Zamiatamy od góry schodząc miotłą po wartościach y malejąco.

- Jeśli kolejny wierzchołek należy do tego samego łańcucha co wierzchołek ze szczytu stosu, to analizujemy kolejne trójkąty, jakie tworzy dany wierzchołek z wierzchołkami zdejmowanymi ze stosu:
  - Jeśli trójkąt należy do wielokąta, to dodajemy przekątną i usuwamy odpowiedni wierzchołek ze szczytu stosu
  - w przeciwnym przypadku umieszczamy badane wierzchołki na stosie.

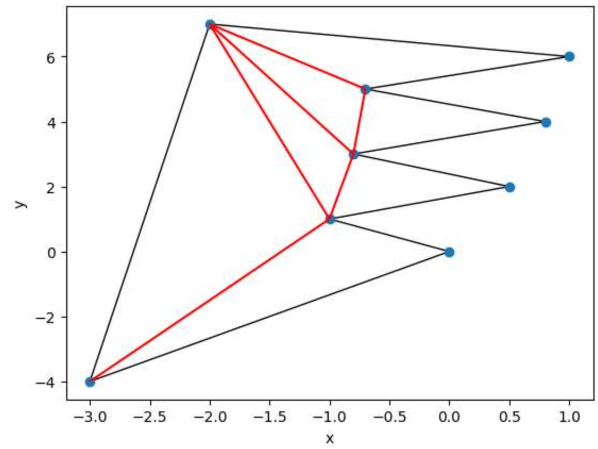
Sprawdzanie czy dany przekątna leży wewnątrz wieloboku realizowane jest za pomocą funkcji is_in_polygon(), która na podstawie wyznacznika oraz informacji o łańcuchu dla danego epsilona (przyjęty 10**(-12)) stwierdza ten fakt.

W każdym kroku dodajemy utworzony trójkąt do listy trójkątów. Przechowywany jest tam w formie 3-elementowej krotki punktów będących wierzchołkami. Przechowujemy także przekątne wewnątrz wielokąta tworzące te trójkąty, głównie do testów, gdyż znacznie bardziej przydatna jest wiedza o utworzonych trójkątach, a nie samych przekątnych. Program drukuje na wyjście kolejne stany stosu dla lepszej obserwacji działania algorytmu. Po zakończonej triangulacji drukowana jest lista trójkątów oraz widzimy wizualizację triangulacji wielokąta. Przykładowe triangulacje poniżej.

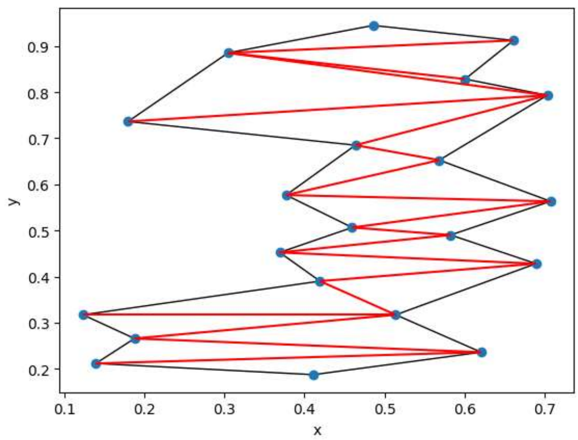
Wizualizacje:



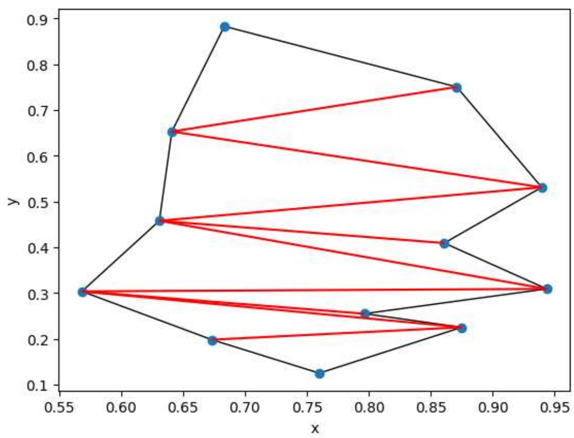
Wykres 5 Triangulacja pięciokąta



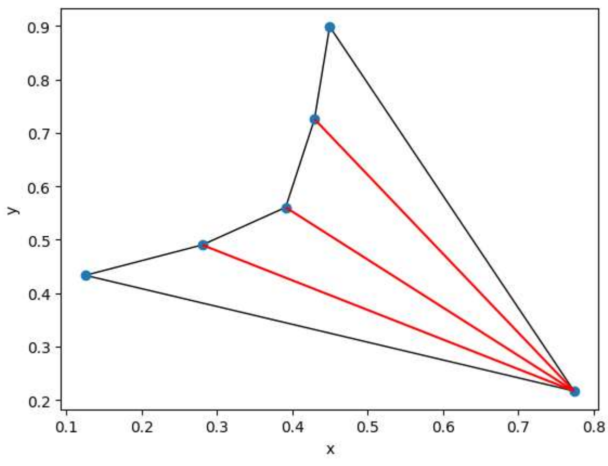
Wykres 6 Triangulacja dziewięciokąta



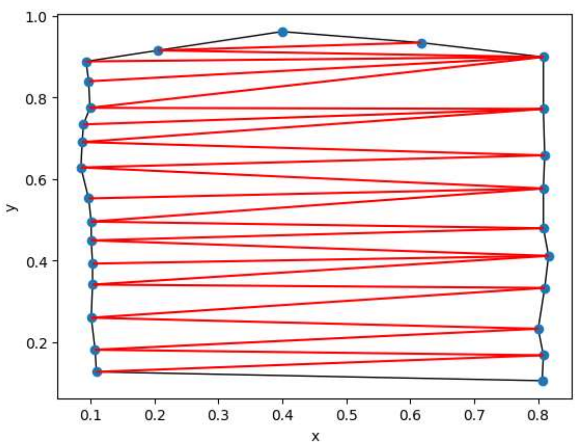
Wykres 7 Triangulacja 21-kąta



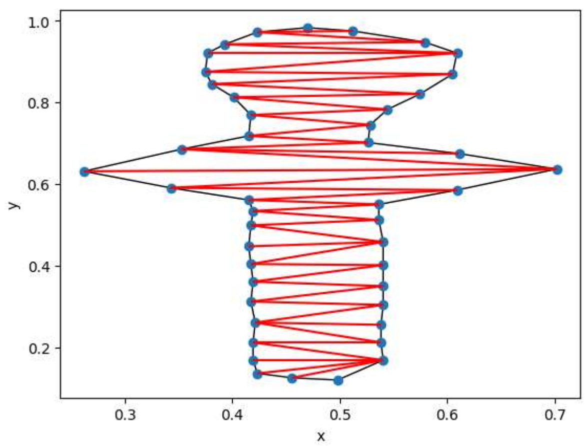
Wykres 8 Triangulacja dwunastokąta



Wykres 9 Triangulacja lotki



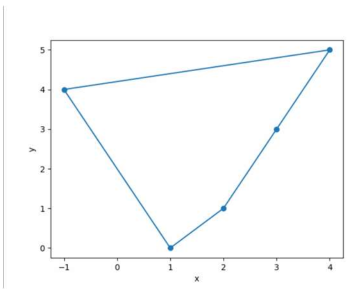
Wykres 10 Triangulacja „kromki chleba”



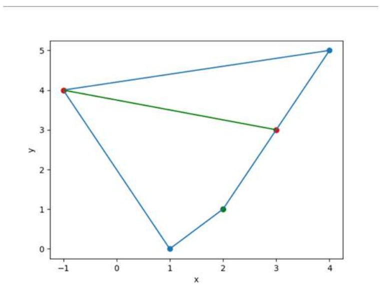
Wykres 11 Triangulacja „ludzika”

4. Wizualizacja procesu triangulacji

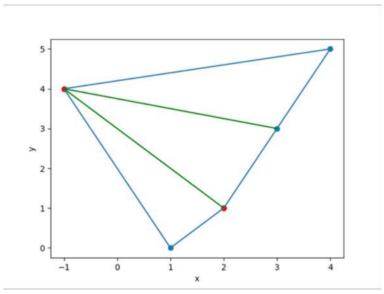
Poniżej zamieszczę kilka przykładowych scen z procesu triangulacji, gify oraz kolejne stany stosu dostępne są w jupyterze. Wizualizacje utworzone zostały przy pomocy dostarczonego narzędzia, a figury rysowane były za pomocą zaimplementowanej funkcji do tworzenia wielokątów przy użyciu myszki (w 5 ostatnich przypadkach, wykresy od 6 do 10). Wierzchołki zaznaczone na czerwono to wierzchołki aktualnie znajdujące się na stosie, a zaznaczony na zielono to ten aktualnie rozpatrywany idąc po wierzchołkach posortowanych malejąco według współrzędnej y.



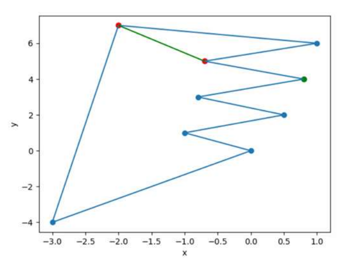
Wykres 12.1 Etap 1



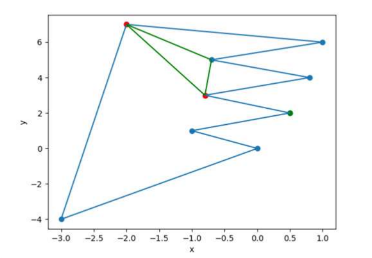
Wykres 12.2 Etap 2



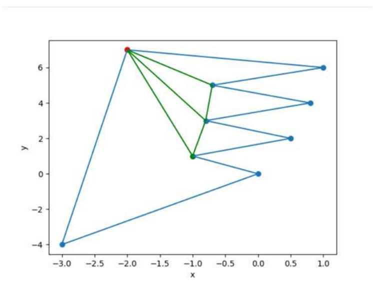
Wykres 12.3 Etap 3



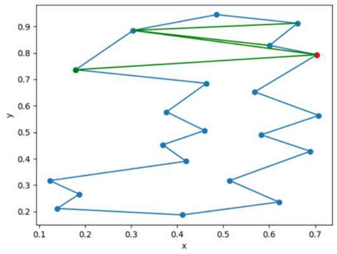
Wykres 13.1 Etap 1

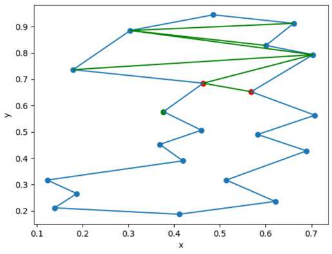


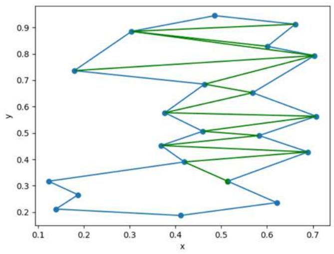
Wykres 13.2 Etap 2

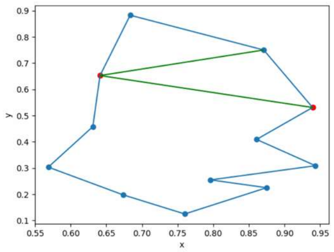


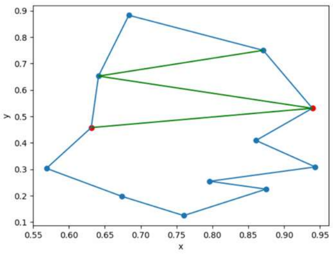
Wykres 13.3 Etap 3

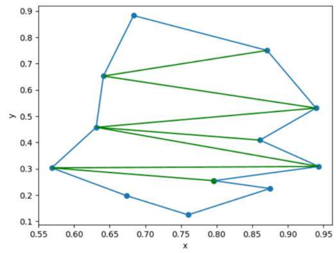

Wykres 14.1 Etap 1

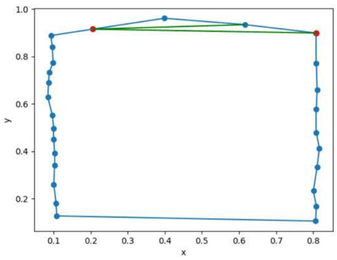

Wykres 14.2 Etap 2

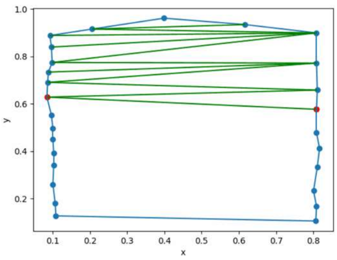

Wykres 14.3 Etap 3

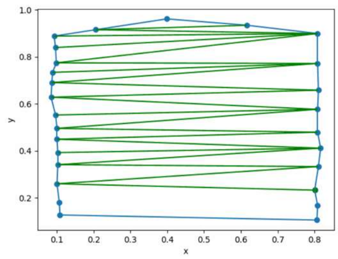

Wykres 15.1 Etap 1

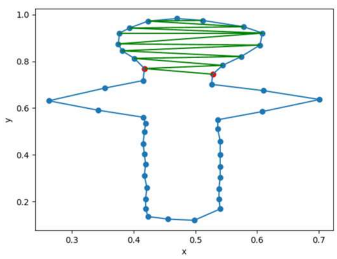

Wykres 15.2 Etap 2

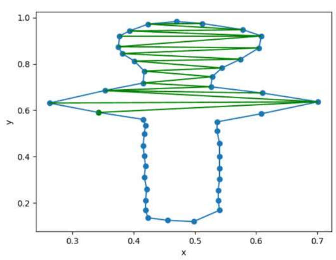

Wykres 15.3 Etap 3

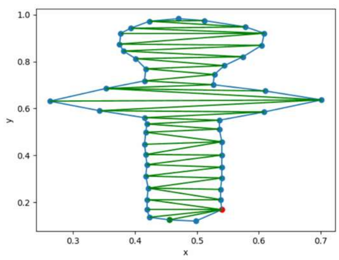

Wykres 16.1 Etap 1

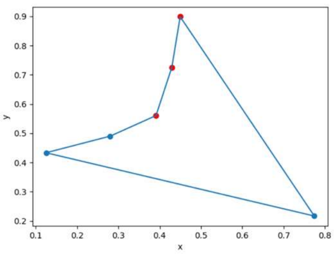

Wykres 16.2 Etap 2


Wykres 16.3 Etap 3

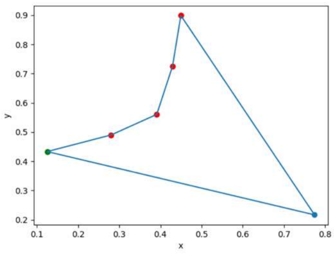

Wykres 17.1 Etap 1


Wykres 17.2 Etap 2

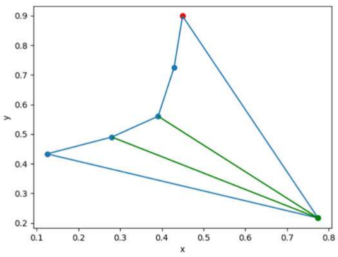

Wykres 17.3 Etap 3



Wykres 18.1 Etap 1



Wykres 18.2 Etap 2



Wykres 18.3 Etap 3

5. Wnioski

Zaimplementowane algorytmy przechodzą wszystkie dostarczone testy oraz wizualizacje zgadzają się z oczekiwaniami. Zmiany stosu także następują wedle oczekiwań, a zwracane trójkąty w postaci listy krotek dobrze obrazują powstałą triangulacje. Można uznać zatem, że zaimplementowane algorytmy działają poprawnie. Wymyślając przykłady wielokątów starałam się sprawdzić działanie algorytmu dla przypadków skrajnych, takich jak wiele wcięć, bardzo małe trójkąty triangulacji, wszystkie punkty oprócz jednego na jednym łańcuchu (tworzące wklęsłą lotkę wykres 9) oraz skrzydło z wcięciami (wykres 6) , dla wszystkich powyższych przykładów algorytm poradził sobie wedle oczekiwać. Kolorowanie wierzchołków odbyło się wedle ustalonych zasad dla wszystkich testowanych wielokątów. Algorytm triangulacji wieloboku monotonicznego jest całkiem ciekawy w swoim działaniu, jednak patrząc globalnie ma on swoje wady takie jak wymaganie dotyczące wielokąta monotonicznego oraz fakt, że powstała triangulacja nie dąży kształtem trójkątów do trójkątów równobocznych tylko dzieli w dowolny sposób. Jednak swoje zadania spełnia, a dla zredukowania potencjalnych minusów można użyć innych znanych algorytmów triangulacji wieloboku.