Algorytmy geometryczne

Anna Gut, 28.10.2020r., grupa nr 7 (czwartek 16:15 A)

Sprawozdanie z ćwiczenia nr 3 – triangulacja wielokąta monotonicznego

1.Wprowadzenie

1.1.Cel ćwiczenia

Celem ćwiczenia było:

- dostosowanie aplikacji graficznej tak, aby istniała możliwość zadawania wielokątów przy użyciu myszki, ich zapis oraz odczyt
- implementacja procedury sprawdzającej czy zadany wielokąt jest y-monotoniczny
- implementacja procedury klasyfikującej wierzchołki wielokąta na: początkowe, końcowe, łączące, dzielące i właściwe
- implementacja procedury dokonującej triangulacji wielokątów monotonicznych
- poddanie zadanych wielokątów działaniu powyższych algorytmów oraz wizualizacja ich przebiegu

1.2.Wstęp teoretyczny

1) Monotoniczność wielokątów

Wielokąt nazywamy ściśle monotonicznym względem prostej l wtedy i tylko wtedy, gdy jest możliwe podzielenie tego wielokąta na dwa spójne łańcuchy takie, że każda prosta prostopadła do prostej l przecina każdy z łańcuchów co najwyżej w jednym punkcie. Wielokąt ściśle monotoniczny względem osi OY nazywamy wielokątem y-monotonicznym.

Przebieg algorytmu sprawdzającego czy dany wierzchołek jest y-monotoniczny:

- wyznaczamy wierzchołek wielokąta o największej współrzędnej y i oznaczamy go jako *highest*
- przechodzimy po poprzednikach tego wierzchołka, dopóki każdy poprzednik ma współrzędną y mniejszą lub równą niż jego następnik. Ostatni wierzchołek spełniający ten warunek oznaczamy jako *lowest*
- w analogiczny sposób przechodzimy po następnikach wierzchołka *highest* i jeżeli napotkamy wierzchołek oznaczony wcześniej jako *lowest*, oznacza to, że badany wielokąt jest y-monotoniczny
- w przeciwnym przypadku badany wielokąt nie jest y-monotoniczny

2) Klasyfikacja wierzchołków

- początkowe – obaj jego sąsiedzi leżą poniżej i kąt wewnętrzny mniejszy od π
- końcowe – obaj jego sąsiedzi leżą powyżej i kąt wewnętrzny większy od π
- łączące – obaj jego sąsiedzi leżą powyżej i kąt wewnętrzny większy od π
- dzielące – obaj jego sąsiedzi leżą poniżej i kąt wewnętrzny mniejszy od π
- właściwe – pozostałe przypadki

3) Triangulacja wielokątów monotonicznych

Triangulacją wielokąta nazywamy jego podział na elementy takie, że:

- każdy z nich ma niepuste wnętrze
- wnętrza różnych elementów są rozłączne
- przecięcie dwóch elementów może być punktem, krawędzią lub zbiorem pustym

Przebieg algorytmu triangulacji wielokąta monotonicznego:

- przy sprawdzaniu monotoniczności realizujemy podział wielokąta na dwa łańcuchy - poprzedników i następników wierzchołka o największej współrzędnej y
- sortujemy wierzchołki wielokąta względem największej współrzędnej y
- tworzymy stos i umieszczamy na nim dwa pierwsze wierzchołki z posortowanej listy
- przechodzimy po kolejnych wierzchołkach posortowanej listy:
  - jeżeli aktualnie rozpatrywany wierzchołek nie należy do tego samego łańcucha, co wierzchołek znajdujący się na szczycie stosu, tworzymy przekątne wychodzące z aktualnie rozpatrywanego wierzchołka do wierzchołków na stosie i usuwamy wierzchołki ze stosu. Następnie umieszczamy na stosie wierzchołki aktualnie i poprzednio rozpatrywane
  - w przeciwnym przypadku usuwamy wierzchołek ze stosu i tworzymy przekątną pomiędzy wierzchołkiem na szczycie stosu, a aktualnie rozpatrywanym wierzchołkiem. Następnie dodajemy do stosu aktualnie rozpatrywany punkt

2.Uwagi techniczne dotyczące sprzętu

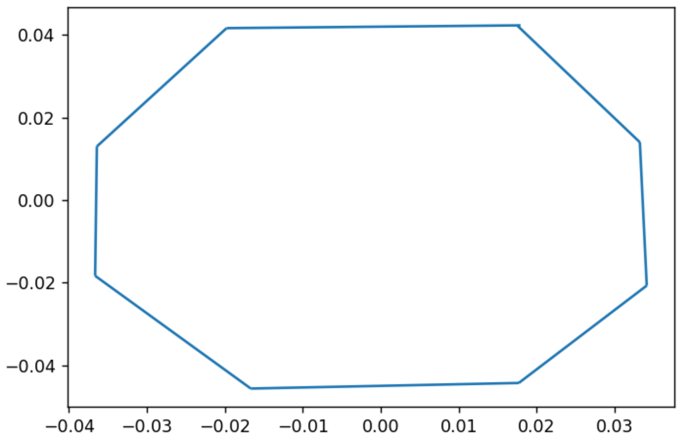
Ćwiczenie zostało przeprowadzone z wykorzystaniem sprzętu o następujących parametrach:

- oprogramowanie Windows 8.1 Pro
- procesor Intel i5
- pamięć RAM 8 GB
- 64-bitowy system operacyjny
- zainstalowane narzędzie graficzne oparte o ProjektJupyter

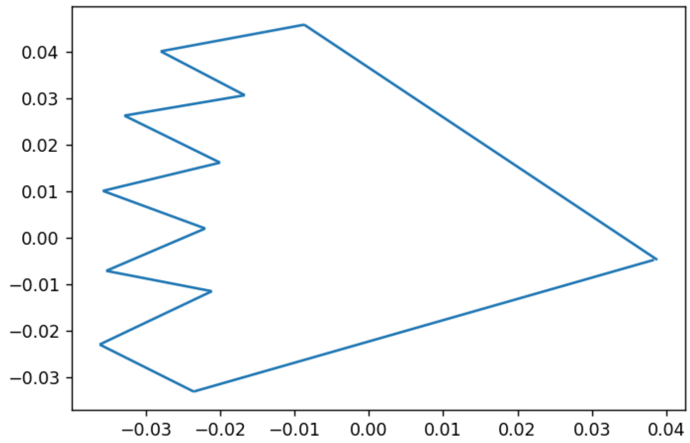
3.Przebieg ćwiczenia

Przy użyciu myszki zostały narysowane następujące wielokąty:

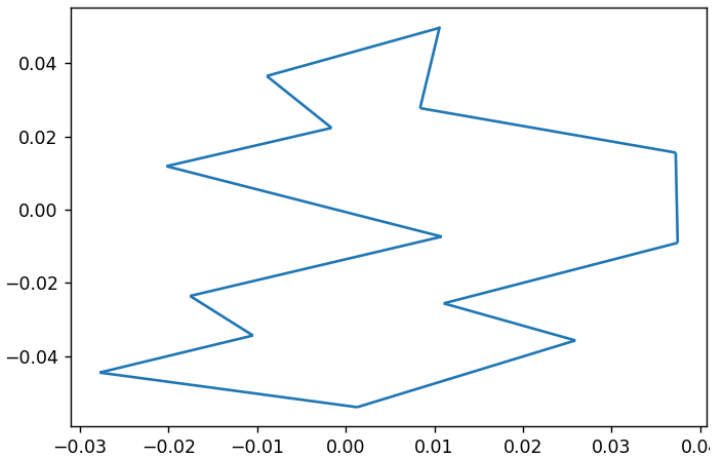
Rysunek nr 1. Wielokąt A



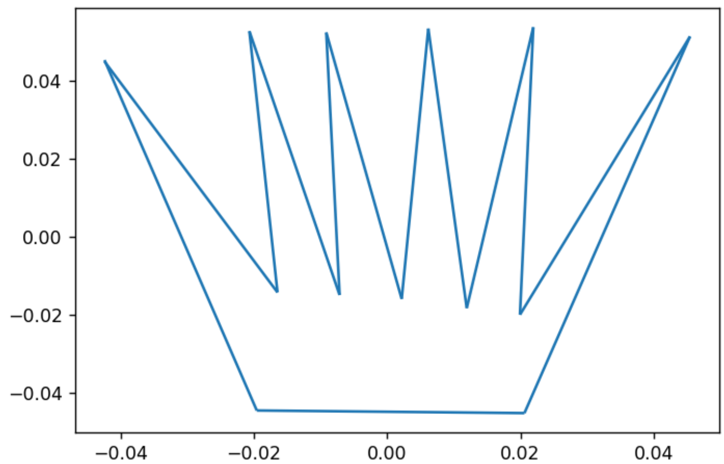
Rysunek nr 2. Wielokąt B



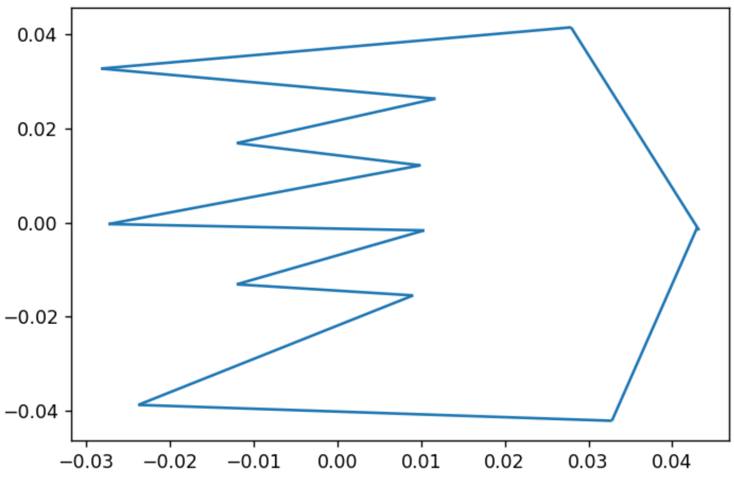
Rysunek nr 3. Wielokąt C



Rysunek nr 4. Wielokąt D



Rysunek nr 5. Wielokąt E



Wielokąty w ćwiczeniu zostały dobrane w taki sposób, aby większość z nich (poza wielokątem D) były monotoniczne i by dało się na nich przeprowadzić procedurę triangulacji. Wielokąty A oraz C cechują się porównywalną ilością wierzchołków w każdym łańcuchu, natomiast w wielokątach B oraz E jeden z łańcuchów składa się tylko z jednego wierzchołka, a drugi z pozostałych. Dzięki temu można zaobserwować odmienny przebieg algorytmu dla różnych wielokątów oraz zweryfikować całkowitą poprawność jego działania. Wykorzystany w ćwiczeniu wielokąt D (niemonotoniczny) służył weryfikacji poprawności działania algorytmu sprawdzania y-monotoniczności.

Po dokonaniu zapisu narysowanych figur, zaimplementowane zostały klasy *Line*, *Point* i *Polygon*. W klasie *Polygon* została dodana możliwość przechowywania zadanych wielokątów za pomocą zbioru wierzchołków połączonych ze sobą przeciwnie do ruchu wskazówek zegara oraz krawędzi. Taka struktura pozwala na łatwy dostęp do poprzednika oraz następnika danego wierzchołka, co pozwala na jednoznaczny podział wielokąta na łańcuchy.

W klasie *Polygon* zostały dodane procedury: sprawdzająca y-monotoniczność danego wielokąta, klasyfikująca punkty w zależności od położenia ich poprzedników oraz następników oraz dokonująca triangulacji, w przypadku, gdy rozpatrywany wielokąt jest monotoniczny.

Poprzez odczyt zadanych figur, utworzone zostały obiekty klasy *Polygon* reprezentujące narysowane wielokąty, a następnie na każdym z nich zostały wykonane wyżej wymienione procedury dostępne wewnątrz tej klasy. Uruchomiona została również wizualizacja graficzna klasyfikacji punktów oraz triangulacji.

4.Klasyfikacja użytych metod

Podczas wykonywania ćwiczenia zostały zaimplementowane następujące struktury oraz metody:

1) Struktury służące do przechowywania wielokąta:

- **Point** – struktura przechowująca wierzchołki wielokąta
- **Line** – struktura przechowująca krawędzie wielokąta
- **Polygon** – struktura przechowująca wielokąt

2) Procedury pozwalające na określenie wzajemnego położenia punktów oraz prostych

- **det** – procedura obliczająca wyznacznik macierzy kwadratowej 3x3
- orient – procedura pozwalająca na określenie, po której stronie prostej leży punkt
- **check_if_belongs** – procedura pozwalająca na określenie czy dany odcinek leży wewnątrz wielokąt
- **classify_points** – procedura dzieląca wierzchołki wielokąta na: początkowe, końcowe, dzielące, łączące oraz właściwe

3) Procedury sprawdzające monotoniczność oraz dokonujące triangulacji

- **y_monotony** – procedura sprawdzająca czy zadany wielokąt jest y-monotoniczny
- **triangulate** – procedura dokonująca triangulacji zadanego wielokąta

W powyższych implementacjach korzystałam z bibliotek functools oraz math.

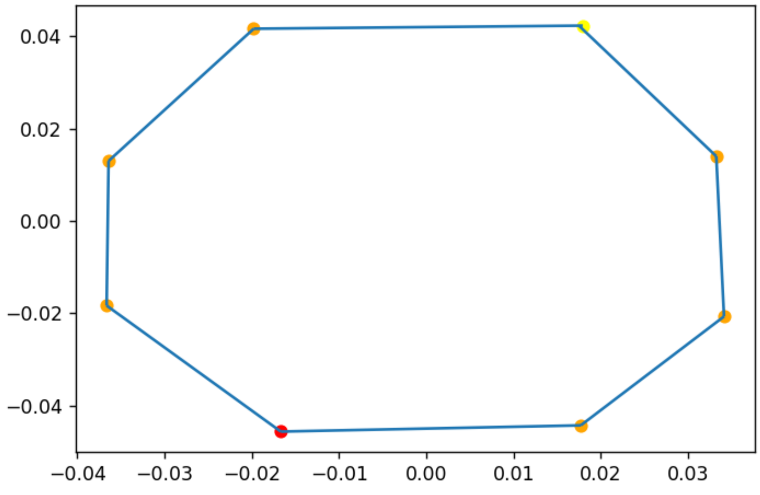
5.Wizualizacja graficzna przebiegu algorytmów

5.1.Klasyfikacja punktów

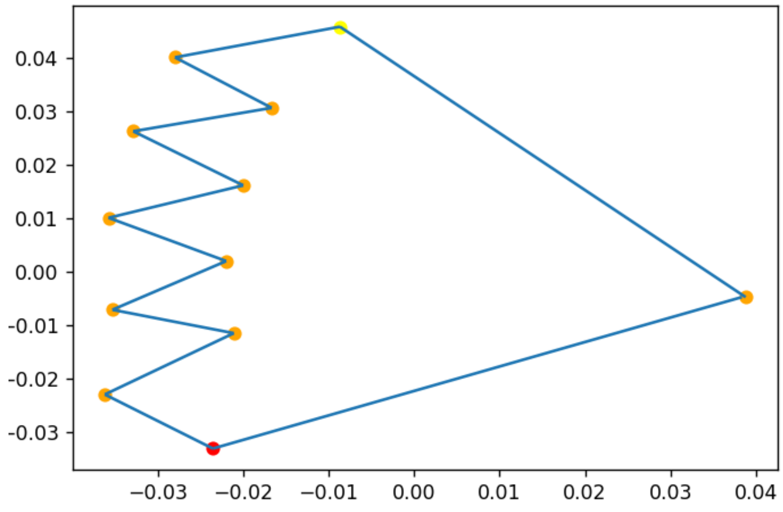
Przedstawione poniżej rysunki prezentują podział wierzchołków w zadanych wielokątach na:

- wierzchołki początkowe – kolor żółty
- wierzchołki końcowe – **kolor czerwony**
- wierzchołki łączące – **kolor zielony**
- wierzchołki dzielące – **kolor czarny**
- wierzchołki prawidłowe – **kolor pomarańczowy**

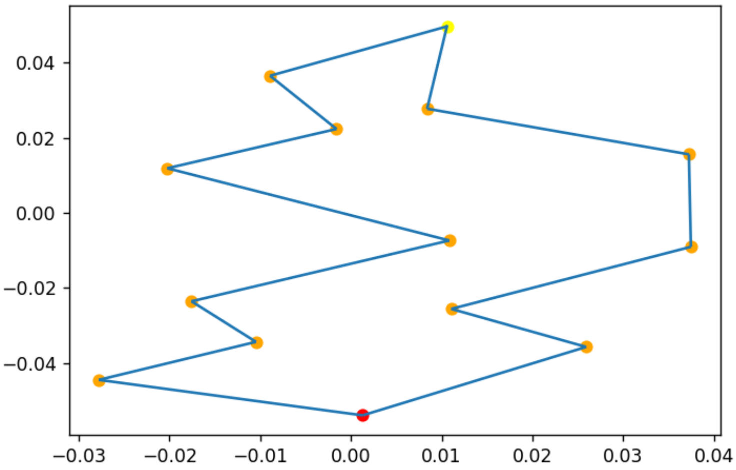
Rysunek 6. Klasyfikacja wierzchołków wielokąta A



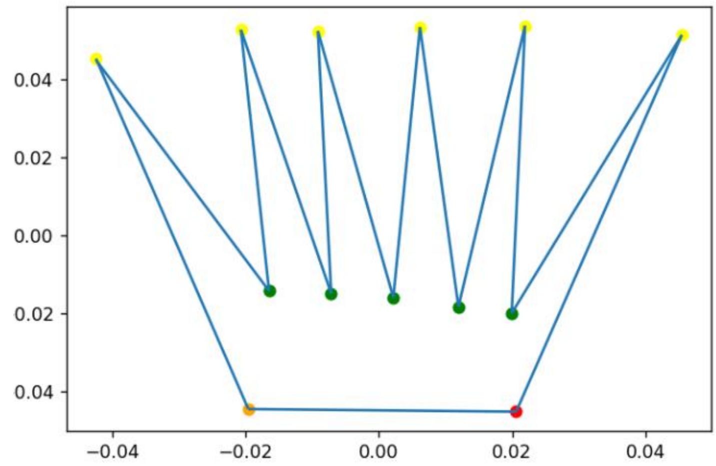
Rysunek 7. Klasyfikacja wierzchołków wielokąta B



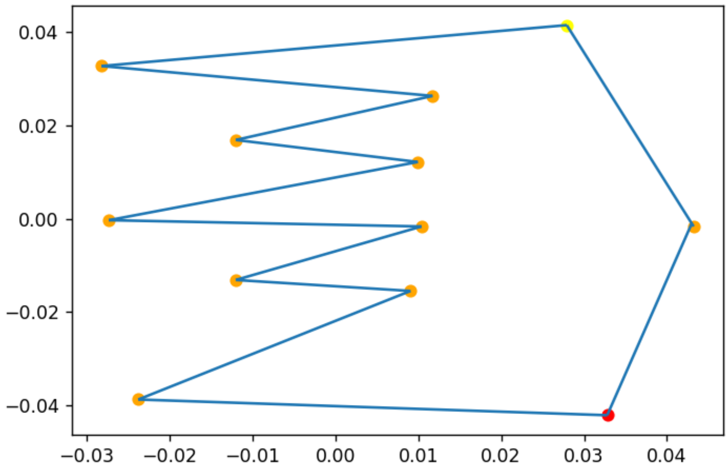
Rysunek 8. Klasyfikacja wierzchołków wielokąta C



Rysunek 9. Klasyfikacja wierzchołków wielokąta D



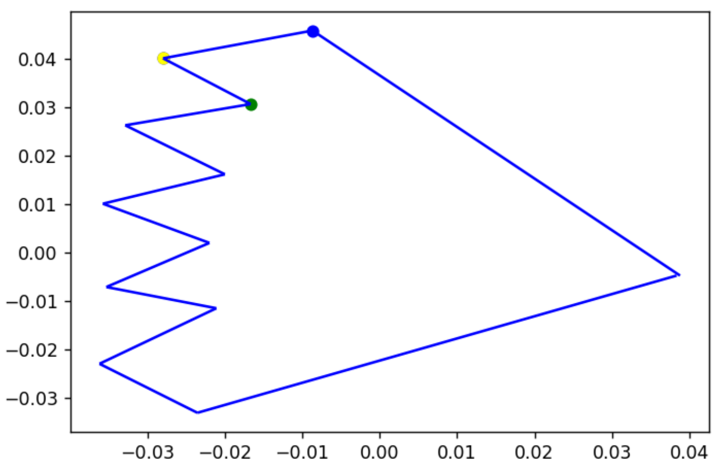
Rysunek 10. Klasyfikacja wierzchołków wielokąta E



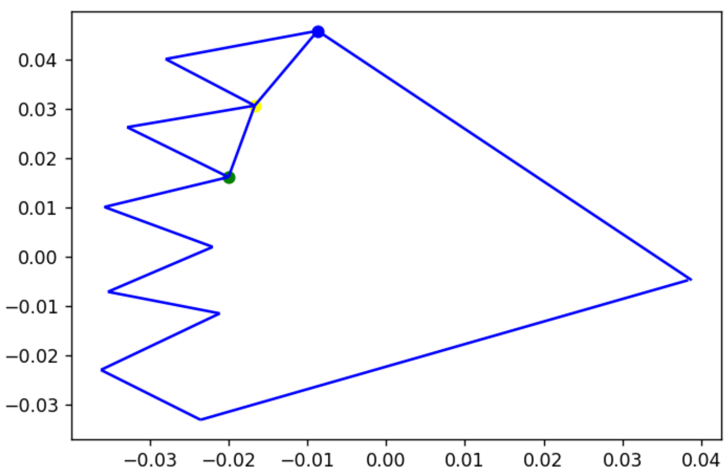
5.2.Triangulacja wielokątów monotonicznych

Poniższe rysunki prezentują przebieg algorytmu triangulacji dla wielokątów B oraz C. Przyjęte są następujące oznaczenia: **niebieski** – punkty aktualnie znajdujące się na stosie, **żółty** – punkt znajdujący się na szczycie stosu, **zielony** – aktualnie rozpatrywany punkt. Kolorem **niebieskim** oznaczane są również kolejne przekątne.

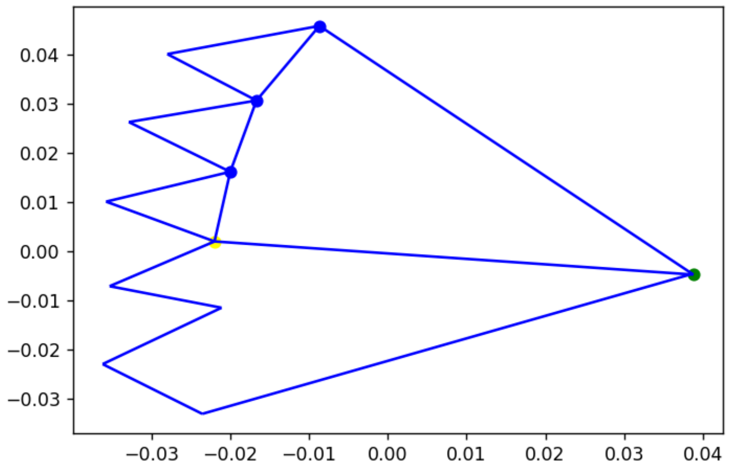
Rysunek nr 11. Wielokąt B



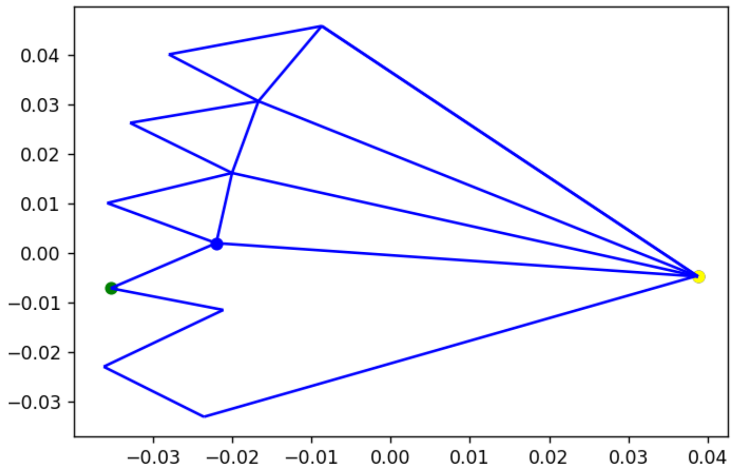
Rysunek nr 12.



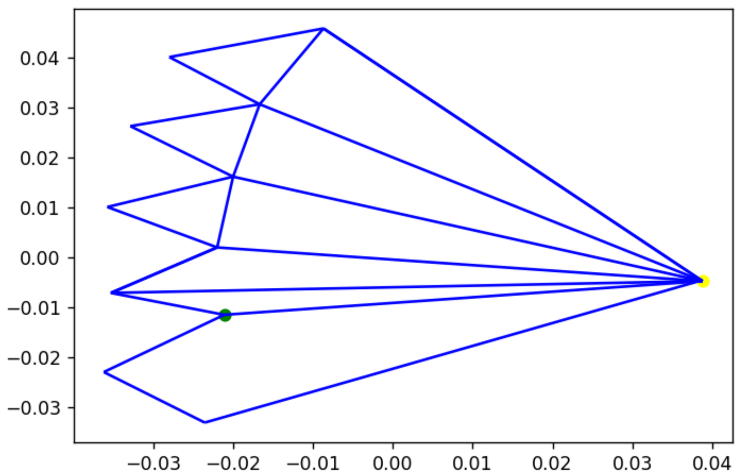
Rysunek nr 13.



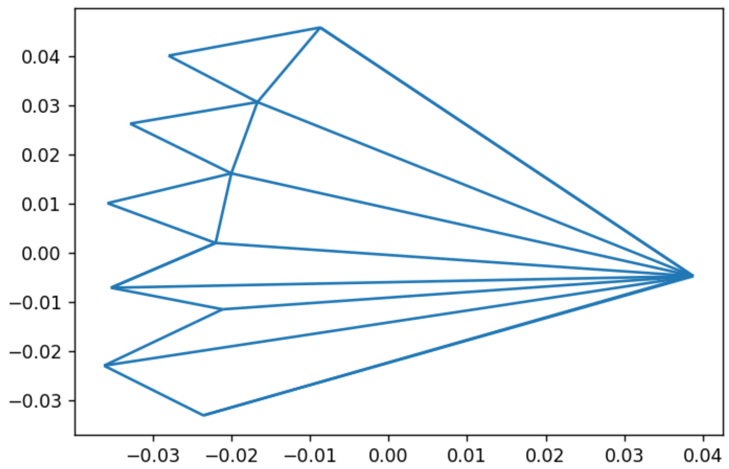
Rysunek nr 14.



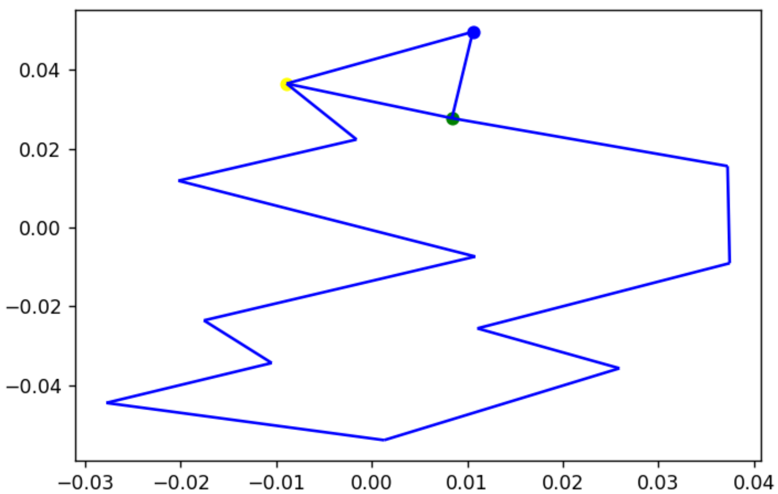
Rysunek nr 15.



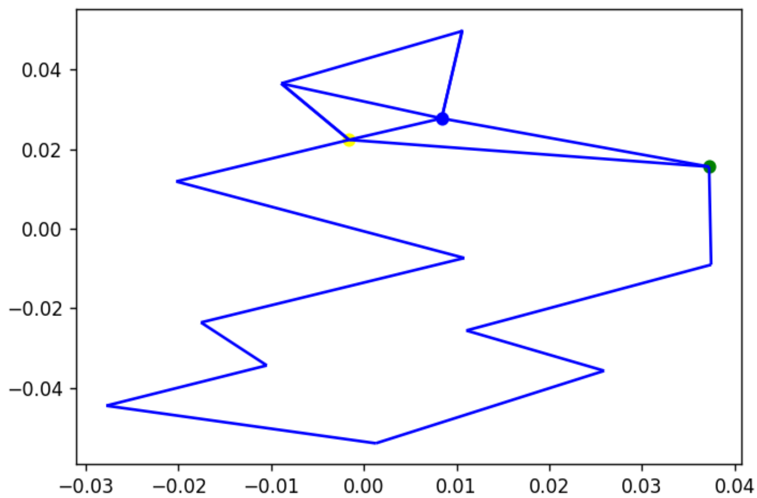
Rysunek nr 16.



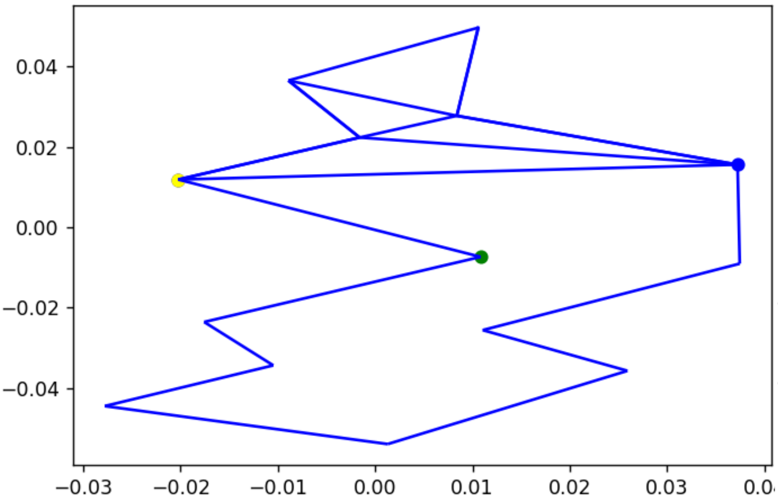
Rysunek nr 17. Wielokąt C



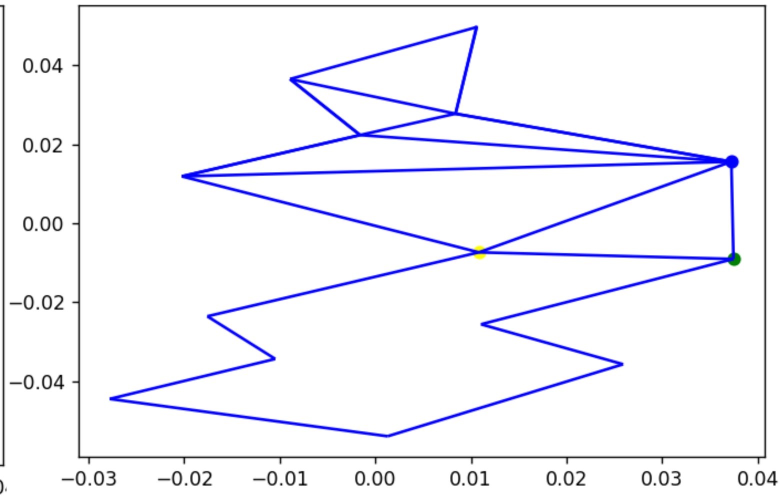
Rysunek nr 18.



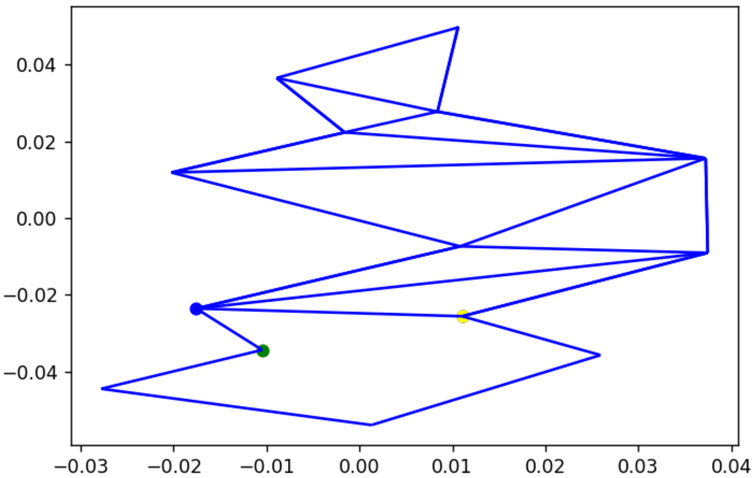
Rysunek nr 19.



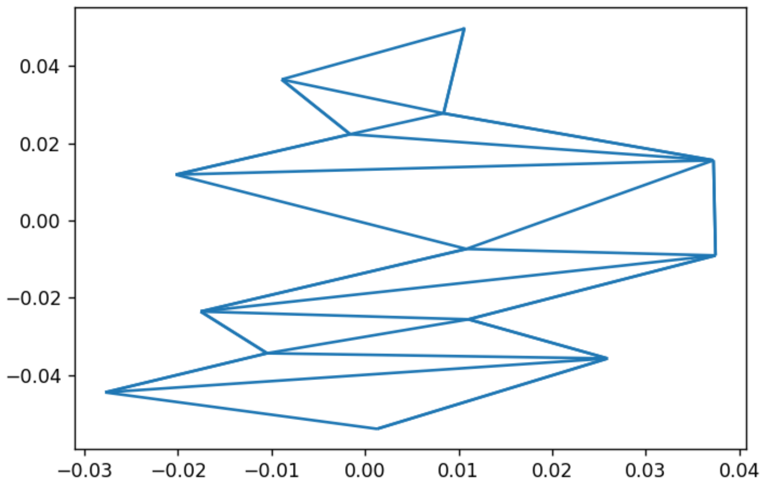
Rysunek nr 20.



Rysunek nr 21.

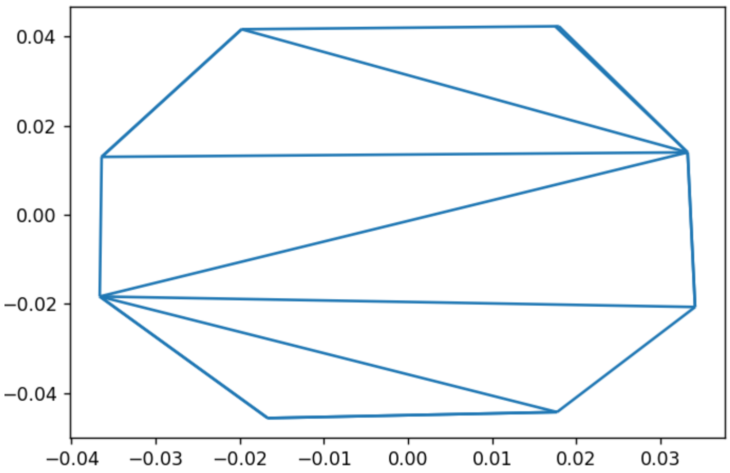


Rysunek nr 22.

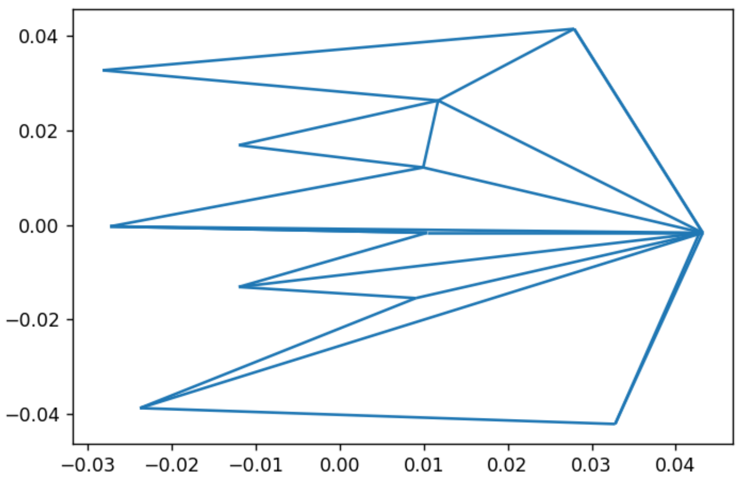


Rysunki 23 oraz 24 przedstawiają triangulację wielokątów A oraz D

Rysunek nr 23.



Rysunek nr 24.



6.Podsumowanie i wnioski

- Analizując otrzymane wyniki, można stwierdzić, że zaimplementowane w ćwiczeniu algorytmy zadziałały poprawnie dla wszystkich testowanych wielokątów. Narysowane figury były na tyle nieskomplikowane, że można było jednoznaczne określić, jaki powinien być rezultat wykonywanych na nich procedur
- Zaimplementowana w zadaniu struktura przechowująca wielokąt okazała się trafnym wyborem, ponieważ umożliwiała ona łatwy podział wierzchołków w wielokącie na dwa rozłączne łańcuchy poprzez dostęp do jego poprzednika oraz następnika w czasie stałym
- Zadane w ćwiczeniu wielokąty monotoniczne były zróżnicowane pod względem ilości wierzchołków w każdym łańcuchu oraz położenia łańcuchów w figurach. Pozwalało to na zaobserwowanie różnic w przebiegu algorytmu triangulacji w zależności od rodzaju wielokąta oraz na dokładniejsze zweryfikowanie poprawności implementacji
- Zadany w ćwiczeniu wielokąt niemonotoniczny pozwalał na zweryfikowanie poprawności algorytmu sprawdzania y-monotoniczności