Algorytmy geometryczne - Ćwiczenie 2. Paulina Jędrychowska

1. Dane techniczne urządzenia i wykorzystane narzędzia:

System Windows 10 x64

Procesor i5-9300H

Pamięć RAM 16GB

Środowisko Jupyter Notebook

Język programowania Python 3.0

2. Tworzenie figur:

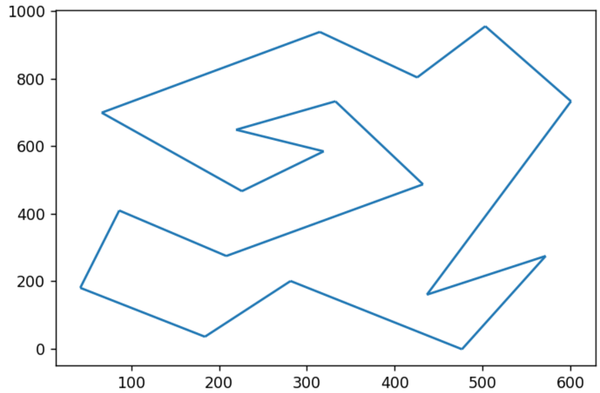
Aplikacja umożliwia tworzenie figur: za pomocą myszki zaznaczamy kolejne punkty w kierunku przeciwnym do ruchu wskazówek zegara. Następnie figurę można zapisać do pliku typu json, a zapisane figury można odczytywać.

Stworzone figury w celu przetestowania algorytmu:

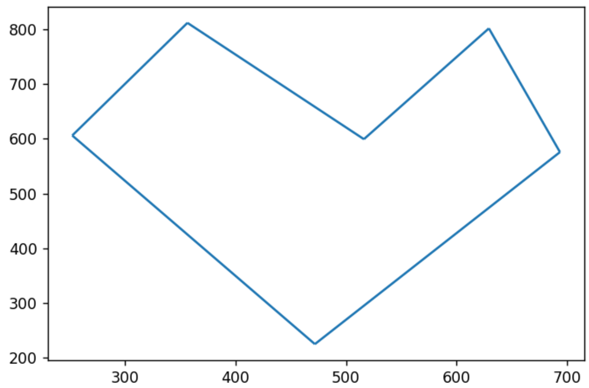
- Figura niemonotoniczna z wykładu („snail_z_wykladu")
- Serce („heart")
- Figura z wykładu, na której pokazywano przebieg algorytmu triangulacji („triangulacja_z_wykladu")
- Strzałka zaproponowana na ćwiczeniach („arrow")
- Lustrzane odbicie strzałki zaproponowanej na ćwiczeniach („reverse_arrow")
- Ptak („bird")
- Okrąg („circle")
- Figura z trzema punktami współliniowymi („collinear")

Wszystkie figury zostały zwizualizowane przez dostarczone narzędzie graficzne oparte o bibliotekę matplotlib.

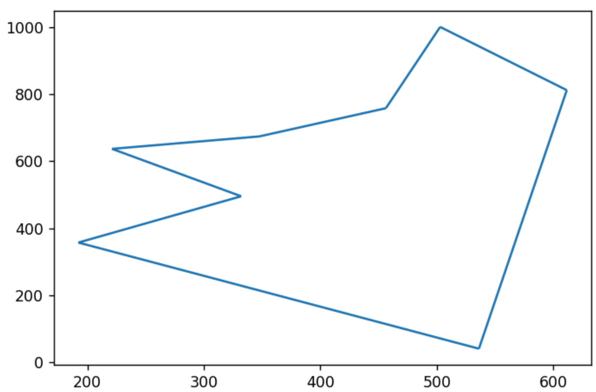
Wykres 1.1. Figura niemonotoniczna z wykładu



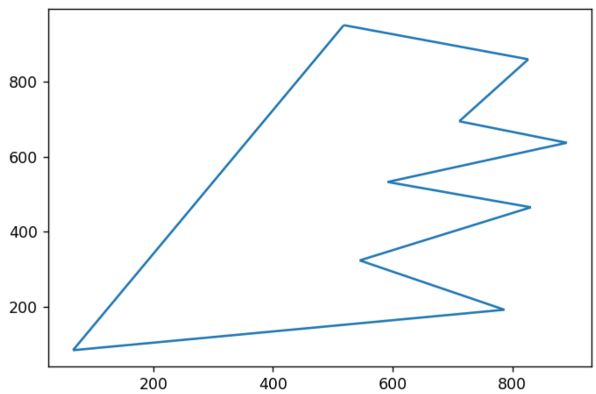
Wykres 1.2. Serce



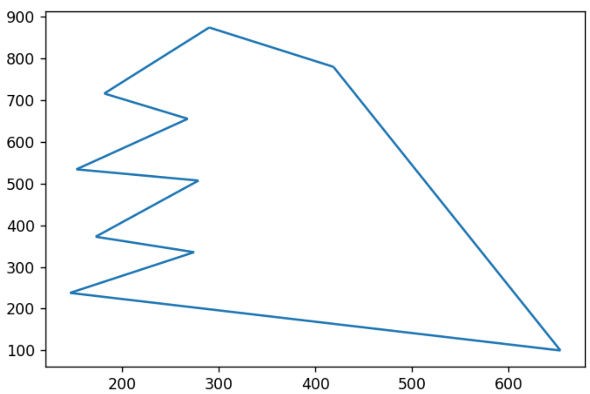
Wykres 1.3. Figura z wykładu



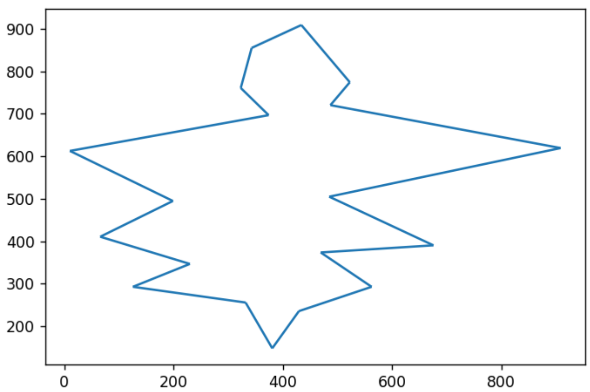
Wykres 1.4. Strzałka z ćwiczeń



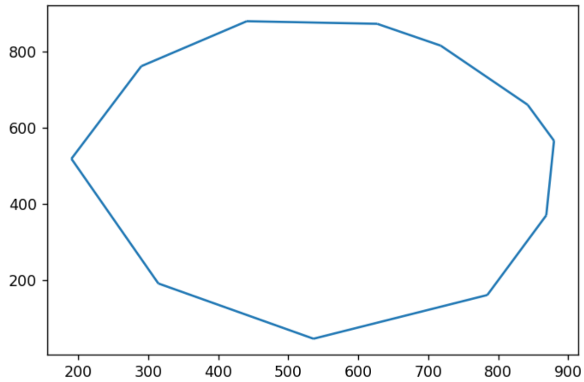
Wykres 1.5. Lustrzane odbicie strzałki z ćwiczeń



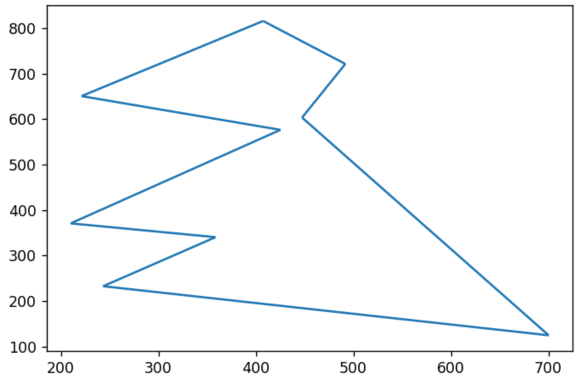
Wykres 1.6. Ptak



Wykres 1.7. Okrąg



Wykres 1.8. Figura z trzema punktami współliniowymi



3. Sprawdzanie y-monotoniczności figury:

Do sprawdzenia monotoniczności figura jest zamieniana na listę wierzchołków. Następnie znajdowane są wierzchołki o największej i najmniejszej współrzędnej y. W algorytmie przechodzimy od najwyższego do najniższego wierzchołka. Idąc najpierw lewym, a następnie prawym łańcuchem, sprawdzamy czy kolejny wierzchołek znajduje się niżej niż poprzedni. Jeżeli w którymś punkcie nastąpi zamiana kierunku, to znaczy, że figura nie jest monotoniczna i zwracany jest odpowiedni komunikat.

4. Obliczanie wyznacznika i tolerancja:

Do wyliczenia wyznacznika użyto wyznacznika 3x3 własnej implementacji.

Wzór 1. Wyznacznik 3x3

$$\det(a,b,c) = \begin{vmatrix} a_x & a_y & 1 \\ b_x & b_y & 1 \\ c_x & c_y & 1 \end{vmatrix}$$

Funkcja orient() zwraca znak wyznacznika, przy określonej tolerancji. Jako tolerancję dla zera przyjęto 10^{-14}.

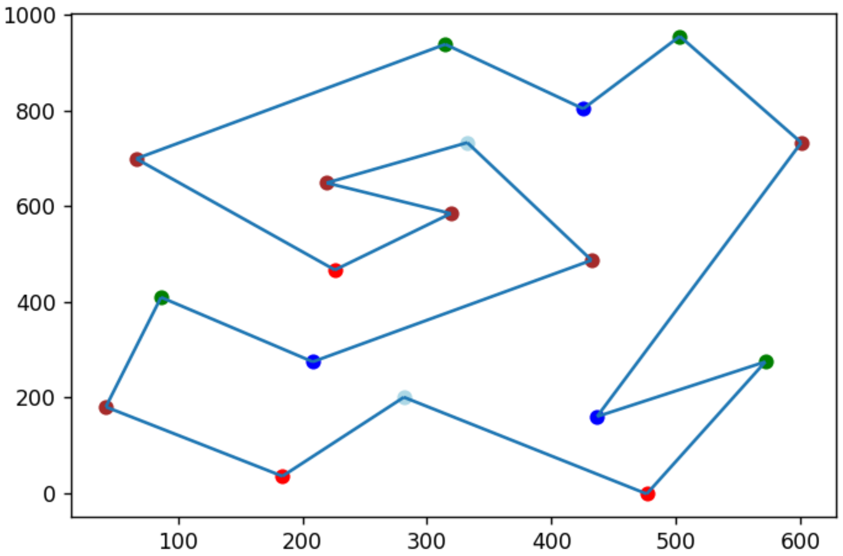
5. Algorytm klasyfikacji wierzchołków:

Algorytm dzieli wierzchołki na początkowe, końcowe, łączące, dzielące i prawidłowe. Sprawdzane są kolejne wierzchołki i określane jest ich położenie względem sąsiadów. Kat wewnętrzny między trzema wierzchołkami wyznaczony jest za pomocą wyznacznika.

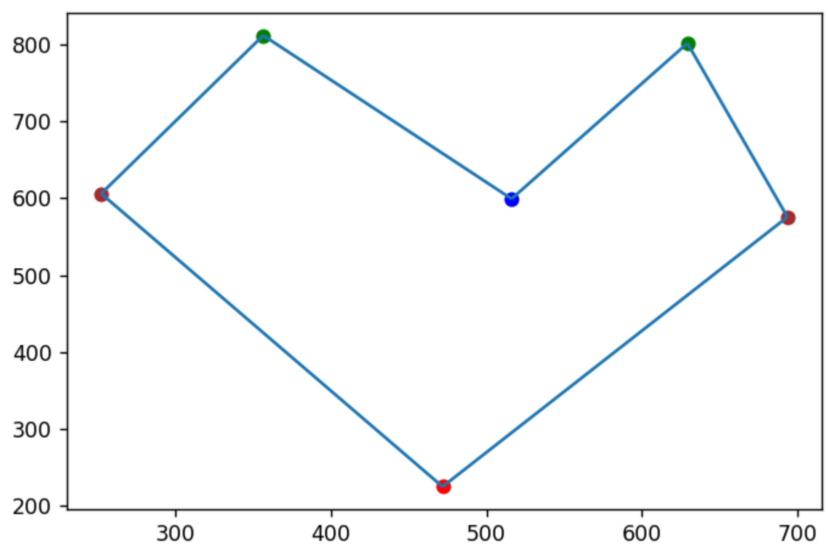
Wierzchołki były klasyfikowane i zaznaczane w następujący sposób:

- Początkowy, gdy obaj jego sąsiedzi leżą poniżej i kąt wewnętrzny < π – kolor zielony
- Końcowy, gdy obaj jego sąsiedzi leżą powyżej i kąt wewnętrzny < π – kolor czerwony
- Łączący , gdy obaj jego sąsiedzi leżą powyżej i kąt wewnętrzny > π – kolor niebieski
- Dzielący, gdy obaj jego sąsiedzi leżą poniżej i kąt wewnętrzny > π – kolor jasnoniebieski
- Prawidłowy, gdy ma jednego sąsiada powyżej drugiego powyżej – kolor brązowy

Rysunek 1.1. Klasyfikacja punktów dla figury niemonotonicznej z wykładu

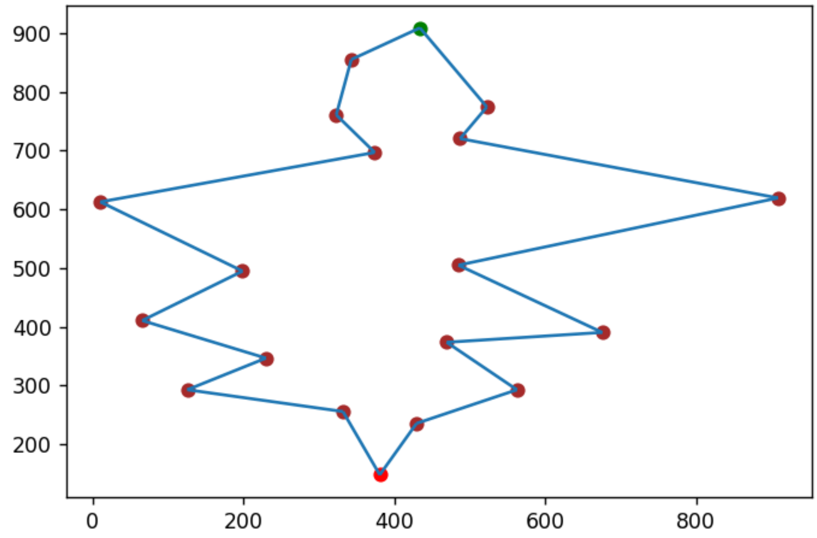


Rysunek 1.2. Klasyfikacja punktów dla serca



Rysunek 1.3. Klasyfikacja punktów dla figury y-monotonicznej

(ptak)



Zgodnie z przewidywaniami dla figury y-monotonicznej wszystkie punkty zakwalifikowane są jako prawidłowe oprócz punktu najwyższego i najniższego które są uznane za wierzchołek początkowy i końcowy.

5. Algorytm triangulacji wielokąta monotonicznego:

Algorytm zmienia figurę na listę wierzchołków i sprawdza czy figura jest monotoniczna. Jeżeli nie jest to algorytm się kończy i zwracana jest pusta lista i pusty graf.

W przeciwnym przypadku znajdowany jest wierzchołek najwyższy i najniższy, dla pozostały wierzchołków określane jest czy należą do lewego czy prawego łańcucha.

Do określenia czy wierzchołek należy do lewego czy prawego łańcucha został użyty podobny algorytm, co przy sprawdzaniu y-monotoniczności: Przechodzimy najpierw prawym, a potem lewym łańcuchem. Wierzchołki w grafie zmieniamy tak, że dla każdego z nich trzymana jest dodatkowa informacja czy leży on na lewym czy na prawym łańcuchu. Najniższy wierzchołek został zakwalifikowany do lewego, a najwyższy do prawego, natomiast nie ma to znaczenia, gdyż położenie najwyższego wierzchołka nigdy nie będzie rozpatrywane, a najniższy jest rozpatrywany osobno. Następnie wierzchołki są sortowane od najwyższego do najniższego.

Przed rozpoczęciem głównej pętli dwa pierwsze wierzchołki wkładane są na stos. Następnie w głównej pętli rozważane są kolejne wierzchołki (oprócz ostatniego). Dla każdego wierzchołka sprawdzamy czy należy do tego samego łańcucha co wierzchołek ze szczytu stosu.

- Jeżeli porównywane wierzchołki należą do różnych łańcuchów, ściągamy ze stosu wszystkie wierzchołki. Tworzymy przekątne między rozpatrywanym wierzchołkiem a wszystkimi (oprócz ostatniego) punktami ściągniętymi ze stosu. Następnie na stos dodawane są dwa ostatnio rozważane wierzchołki.

- Jeżeli porównywane wierzchołki należą do tego samego łańcucha, to sprawdzamy trójkąty jakie tworzą oba wierzchołki z kolejnym wierzchołkiem ze stosu. Sprawdzenia dokonujemy za pomocą wyznacznika. Jeżeli sprawdzone punkty leżą na lewym łańcuchu, to trójkąt należeć będzie do wielokąta, jeżeli wyznacznik będzie dodatni, dla punktów leżących na prawym łańcuchu wyznacznik powinien być ujemny.
    a) Jeżeli trójkąt należy do wielokąta to tworzymy przekątną między wierzchołkiem rozpatrywanym a przedostatnim ze stosu. Usuwamy wierzchołek ze szczytu stosu i rozpatrujemy kolejne trójkąty.
    b) W przeciwnym przypadku umieszczamy rozpatrywany wierzchołek na szczycie stosu.

Po zakończeniu głównej pętli osobno rozpatrywany jest najniższy wierzchołek. Łączymy go z wszystkimi wierzchołkami aktualnie znajdującymi się na stosie, oprócz pierwszego i ostatniego.

Zwracamy rezultat jako parę: lista utworzonych przekątnych oraz wykres kolejnych kroków algorytmu dla danej figury.

6. Wizualizacja przebiegu algorytmu:

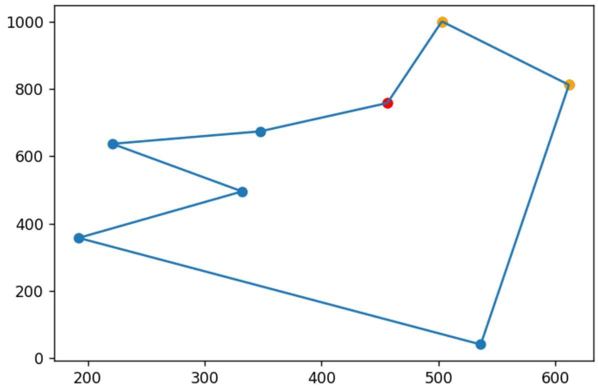
Legenda oznaczeń wykresów:

- Granatowy - punkty i krawędzie należące do wielokąta
- Żółty - punkty znajdujące się na stosie
- Czerwony – punkt aktualnie rozpatrywany
- Zielony – przekątne wielokąta

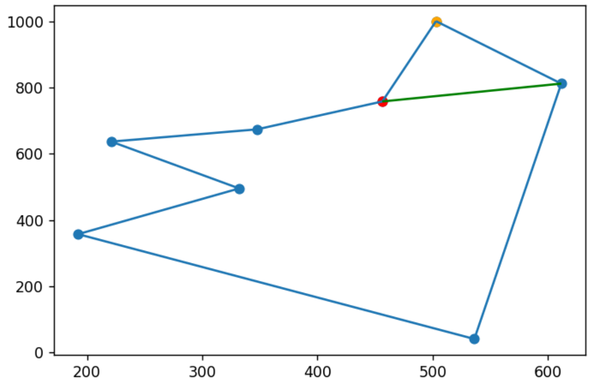
Wykres 2. Wizualizacja algorytmu triangulacji dla

figury z wykładu

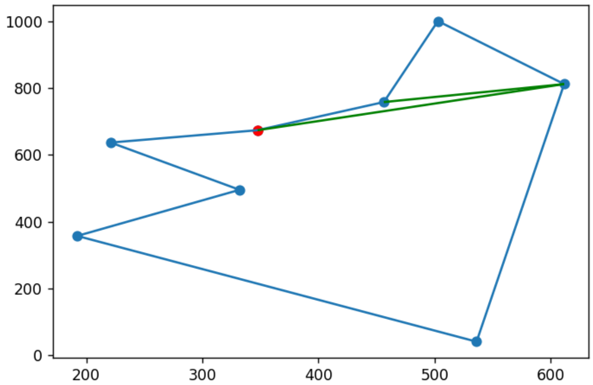
Wykres 2.1. Dwa pierwsze punkty znajdują się na stosie



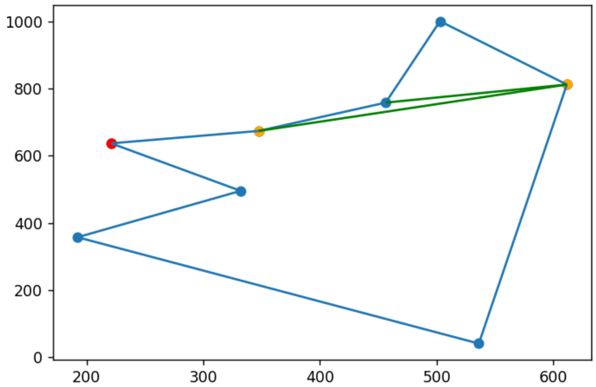
Wykres 2.2. Dodanie pierwszej przekątnej dla punktów z dwóch różnych łańcuchów



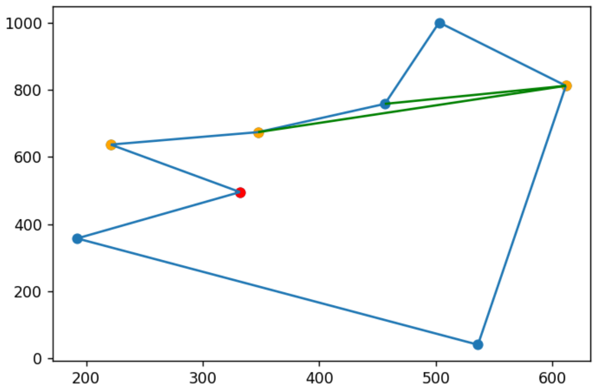
Wykres 2.3. Dodanie drugiej przekątnej dla punktów tworzących trójkąt wewnętrzny



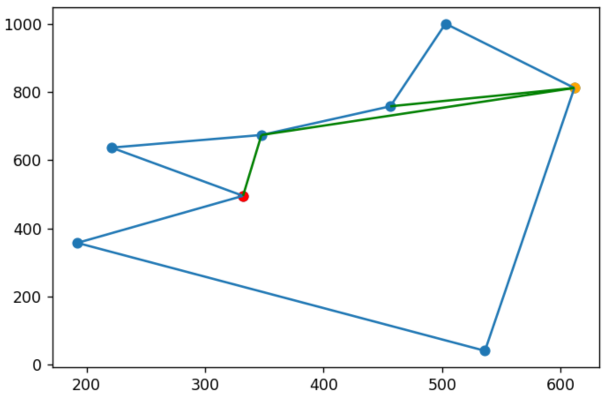
Wykres 2.4. Rozpatrywany jest kolejny punkt, kolejna trójka punktów jest współliniowa



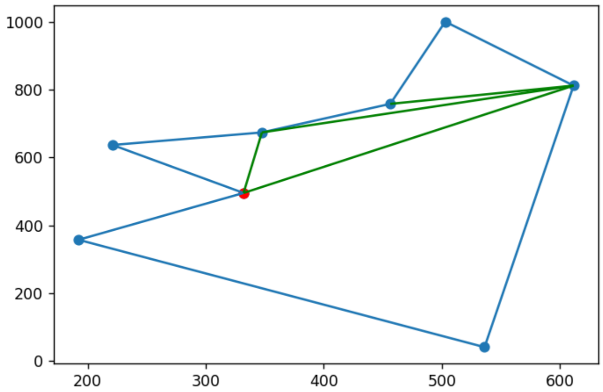
Wykres 2.5. Rozpatrywany jest kolejny punkt



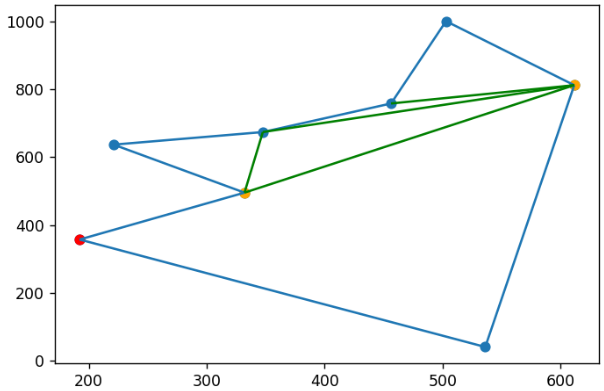
Wykres 2.6. Tworzony jest kolejny trójkąt dla punktów tworzących trójkąt wewnętrzny



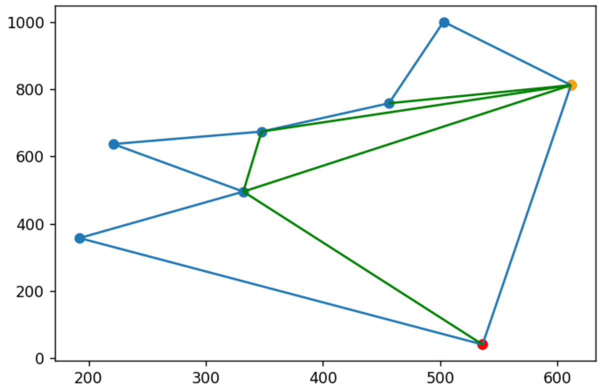
Wykres 2.7. Tworzony jest trójkąt dla następnego punktu ze stosu



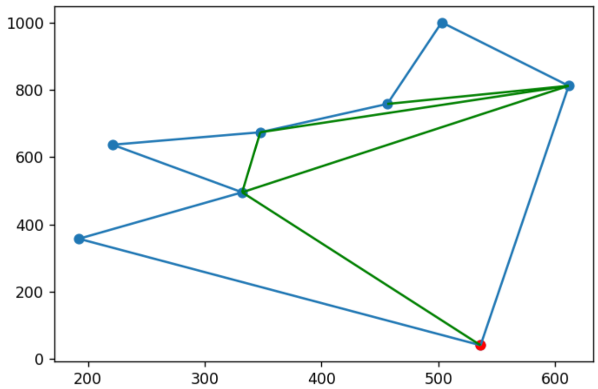
Wykres 2.8. Rozpatrywany jest kolejny punkt kolejna trójka punktów jest współliniowa



Wykres 2.9. Tworzenie przekątnych dla ostatniego punktu

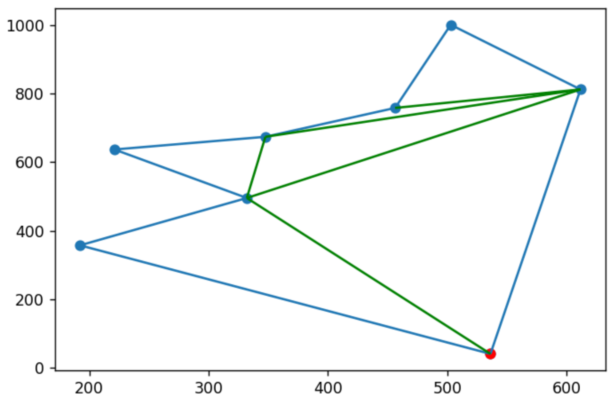


Wykres 2.10. Wynik algorytmu triangulacji



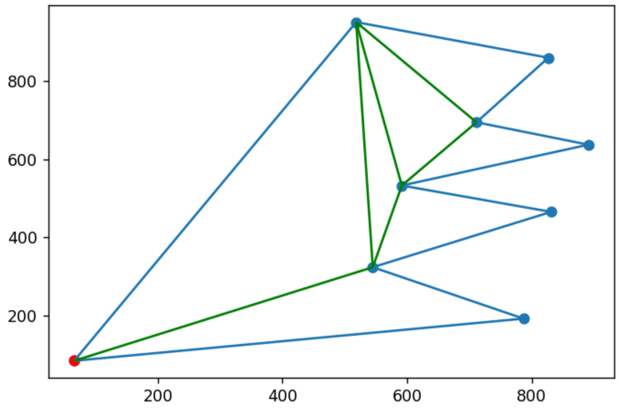
Wyniki triangulacji dla przykładowych figur:

Wykres 3.1. Wynik triangulacji dla figury z wykładu



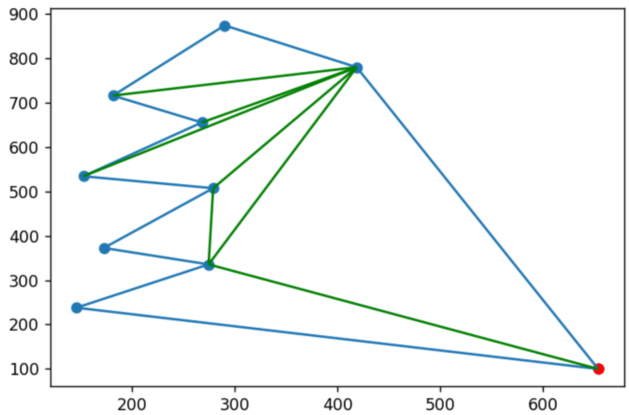
Wykres 3.2. Wynik triangulacji dla

strzałki z ćwiczeń



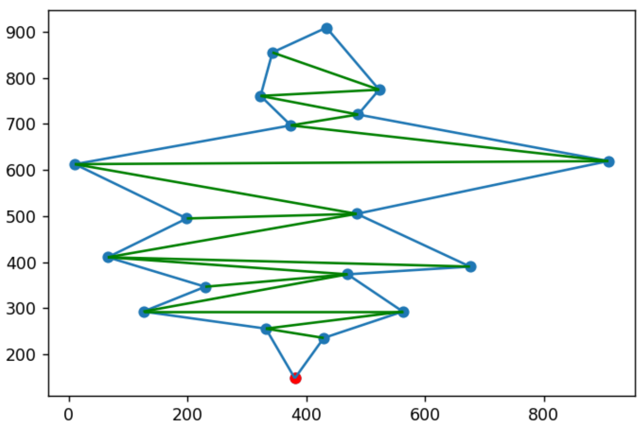
Wykres 3.3. Wynik triangulacji dla

lustrzanego odbicia strzałki z ćwiczeń



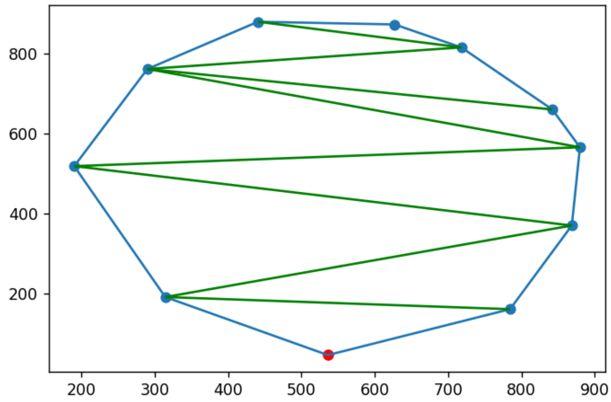
Wykres 3.4. Wynik triangulacji dla

ptaka



Wykres 3.5. Wynik triangulacji dla

okręgu



Wykres 3.6. Wynik triangulacji dla

figury z trzema punktami współliniowymi

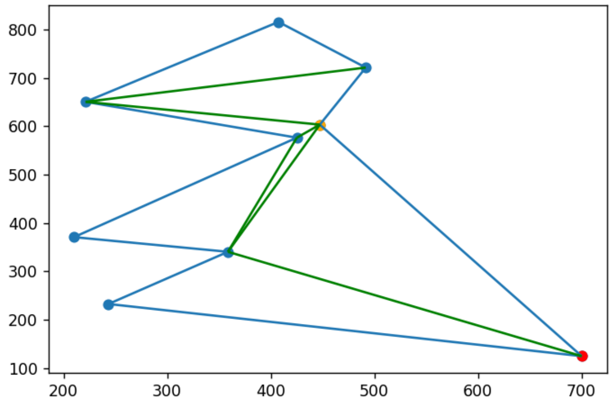


Tabela 1. Ilość przekątnych dla każdej figury

Wybrana figura	**Liczba przekątnych**
Figura z wykładu	5
Strzałka z ćwiczeń	6
Lustrzane odbicie strzałki z ćwiczeń	7
Ptak	16
Okrąg	8
Figura z trzema punktami współliniowymi	6

Wybór figur do testowania:

Okrąg jest figurą wypukłą, a więc testuje ona podstawowy przypadek triangulacji, gdzie wszystkie trójkąty są w środku. Pozostałe figury nie są wypukłe. Strzałki z ćwiczeń testują algorytm, gdy większość punktów znajduje się na jednym z łańcuchów. Natomiast dla ptaka sprawdzam działanie gdy punkty są w obu łańcuchach. Figura z trzema punktami współliniowymi została stworzona do przetestowania klasyfikacji punktów względem prostej i punkty zostały zaklasyfikowane poprawnie przy danej tolerancji (nie stworzono z nich trójkąta).

Dla wszystkich wielokątów triangulacja przebiegała poprawnie.

7. Wnioski:

Wszystkie algorytmy działają poprawnie. Zapisywanie i odczytywanie figur z pliku znacznie ułatwia analizę danych. Wyznacznik macierzy okazuje się być przydatny w bardzo wielu sytuacjach, gdzie potrzebujemy znaleźć położenie punktów względem siebie albo kąta między nimi. Struktury, w których przechowywano wielokąt umożliwiały szybki dostęp do potrzebnych danych. Wybrane figury były różnorodne i testowały różne kroki algorytmu. Algorytmy działają sprawnie i nie zauważono większych problemów w ich działaniu.