

Marceli Jędryka

Algorytmy Geometryczne – Laboratorium 3

Środowisko oraz sprzęt:

→ Wszystkie ćwiczenia zostały wykonane w Jupyter Notebook przy użyciu języka python oraz bibliotek Numpy i Matplotlib.

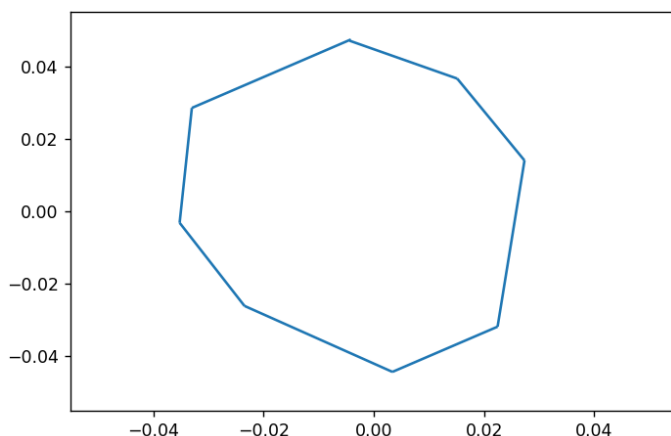
→ Obliczenia przeprowadzone na systemie operacyjnym Windows 10 x64 z procesorem Intel Core i5-10210U CPU 2.11 GHz.

1. Zadawanie wielokątów w aplikacji

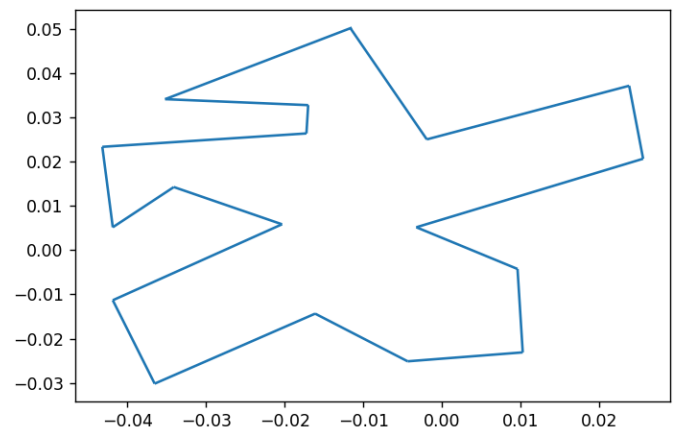
Zaimplementowana została sekcja plot, przy wywołaniu której istnieje możliwość narysowania figury, a następnie zapisania jej do pliku. Ponadto funkcja load_file jest odpowiedzialna za wczytywanie wcześniej stworzonych wielokątów.

2. Sprawdzanie y-monotoniczności wielokąta

Wielokąt jest y-monotoniczny wtedy, gdy każda prosta prostopadła do osi y i przechodząca przez ten wielokąt przecina go co najwyżej w dwóch punktach. Algorytm sprawdzania takiej zależności korzysta ze sposobu rysowania wielokąta (odwrotnie do ruchu wskazówek zegara). Zaczynając od punktu o najwyższej współrzędnej y sprawdzany jest lewy i prawy łańcuch punktów. W każdym wywołaniu pętli rozpatrywane są dwa punkty wielokąta. Jeśli następny punkt będzie wyżej niż swój poprzednik, to taki wielokąt nie jest y-monotoniczny - zwracana jest wartość False.



rys 2.1



rys 2.2

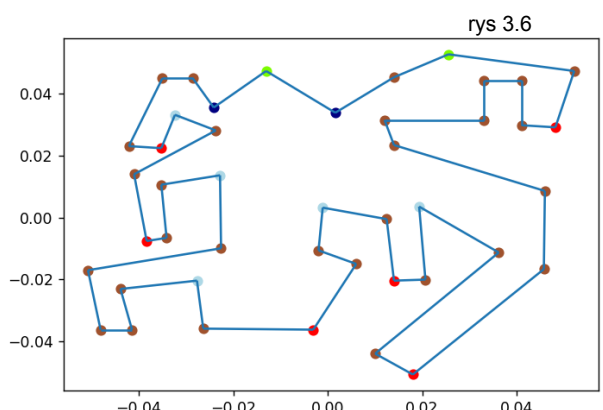
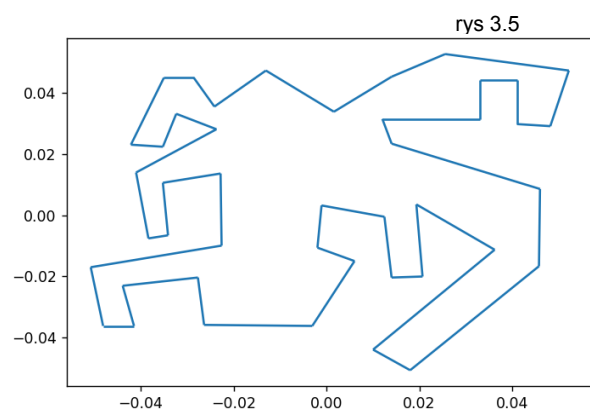
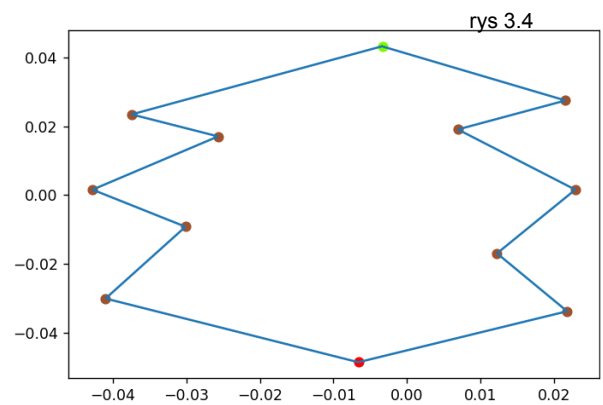
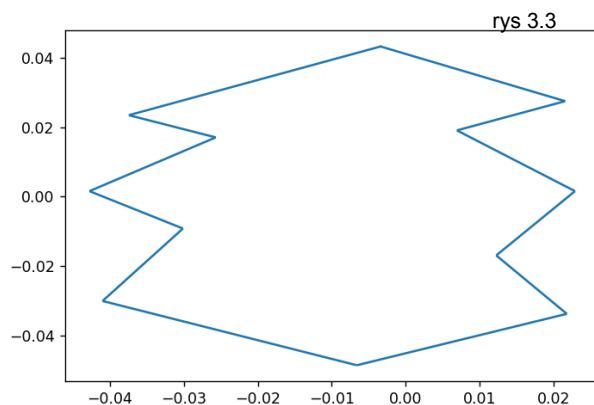
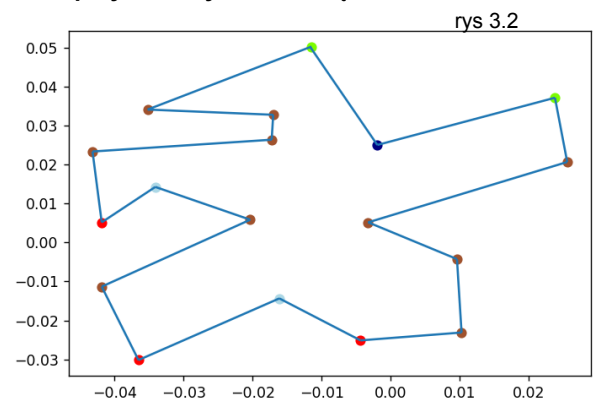
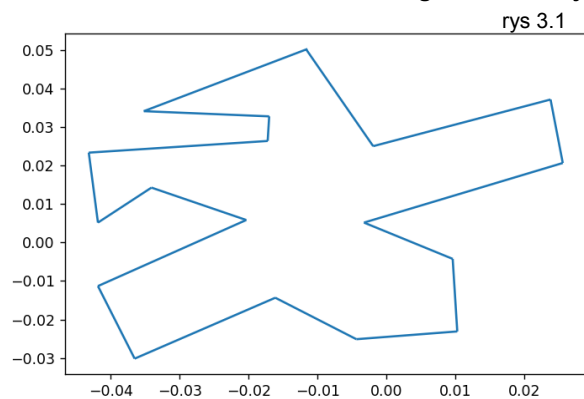
Powyższe grafiki przedstawiają przykłady wielokąta y-monotonicznego (rys 2.1) oraz wielokąta, który nie jest y-monotoniczny (rys 2.2)

3. Klasyfikacja punktów wielokąta ze względu na ich rodzaj

Algorytm klasyfikujący punkty wielokąta polega na rozpatrywaniu położenia trzech kolejnych punktów względem siebie według poniższego wzorca:

- **początkowy** (zielone) - gdy obaj jego sąsiedzi leżą poniżej i kąt wewnętrzny $< \pi$
- **końcowy** (czerwone) - gdy obaj jego sąsiedzi leżą powyżej i kąt wewnętrzny $< \pi$
- **łączyący** (granatowe) - gdy obaj jego sąsiedzi leżą powyżej i kąt wewnętrzny $> \pi$
- **dzielący** (niebieskie) - gdy obaj jego sąsiedzi leżą poniżej i kąt wewnętrzny $> \pi$
- **prawidłowy** (brązowe) - , w pozostałych przypadkach (ma jednego sąsiada powyżej, drugiego – poniżej)

Poniższe grafiki obrazują klasyfikację punktów dla przykładowych wielokątów



4.1 Triangulacja wielokąta monotonicznego

Triangulacja to rozkład wielokąta na trójkąty za pomocą jego przekątnych. Algorytm zaimplementowany został zgodnie z poniższym schematem:

→ wyznaczenie prawego i lewego łańcucha wielokąta względem kierunku monotoniczności

→ uporządkowanie wierzchołków względem współrzędnej kierunku monotoniczności

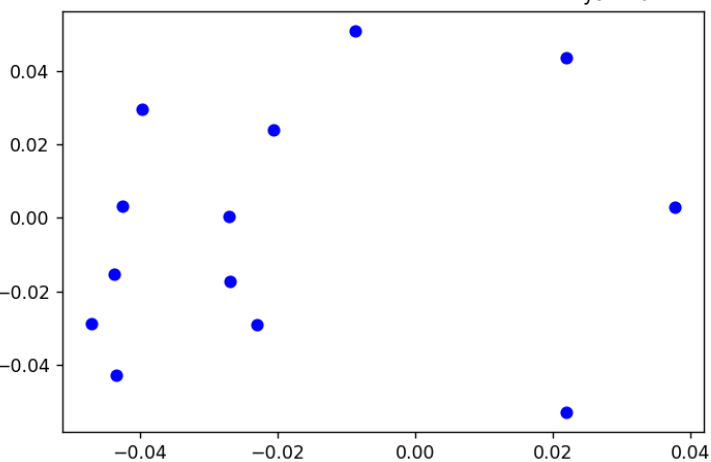
→ Jeśli kolejny wierzchołek należy do innego łańcucha niż wierzchołek stanowiący szczyt stosu, to możemy go połączyć ze wszystkimi wierzchołkami na stosie. Na stosie zostają dwa wierzchołki, które były „zamiatane” ostatnie.

→ 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.

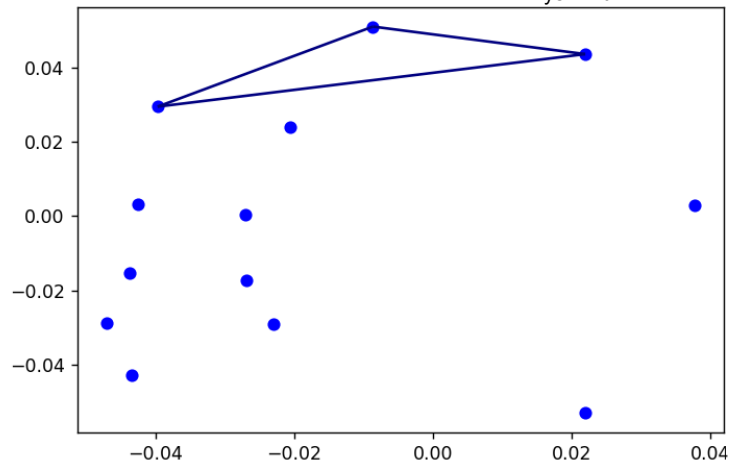
→ Jeśli trójkąt należy do wielokąta, to usuwamy wierzchołek ze szczytu stosu, w przeciwnym przypadku umieszczamy badane wierzchołki na stosie.

Poniższe grafiki pokazują proces działania algorytmu

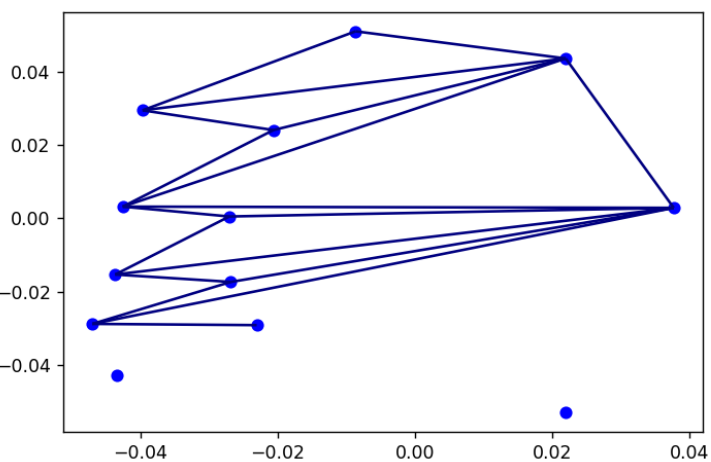
rys 4.1a



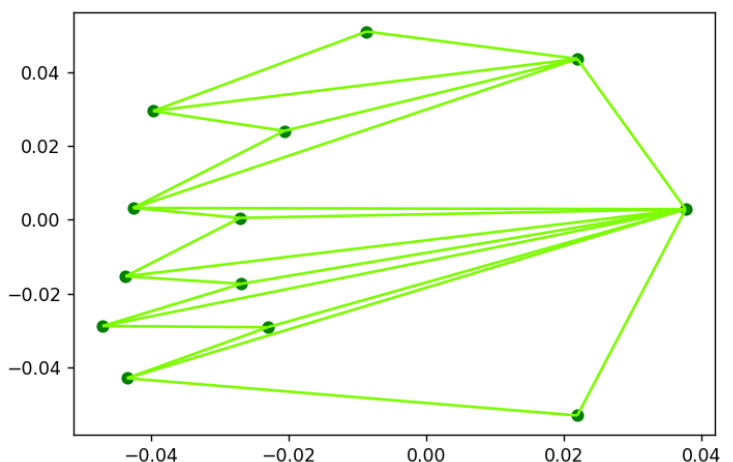
rys 4.1b



rys 4.1c

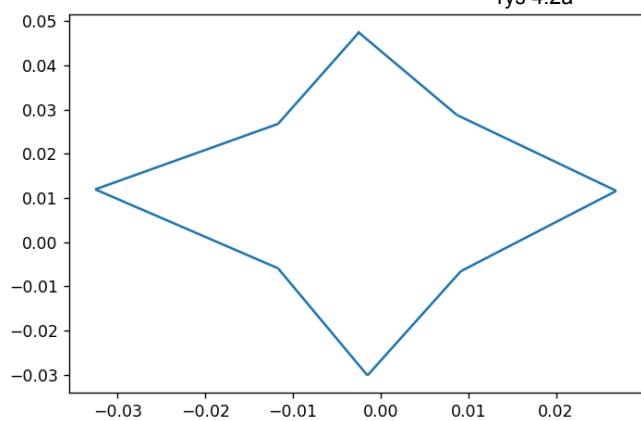


rys 4.1d

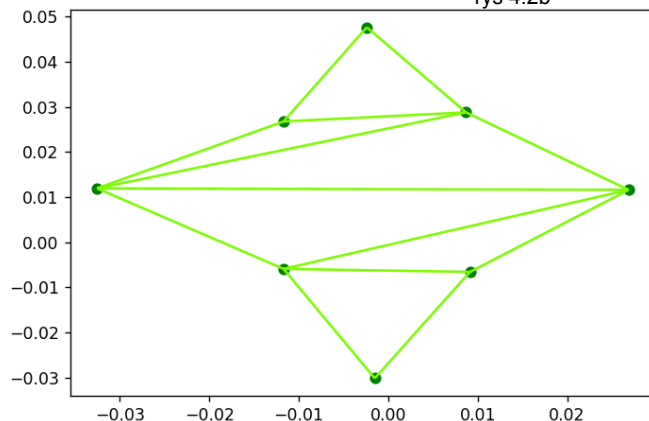


Poprawność algorytmu testowana była m.in. na poniższych wielokątach

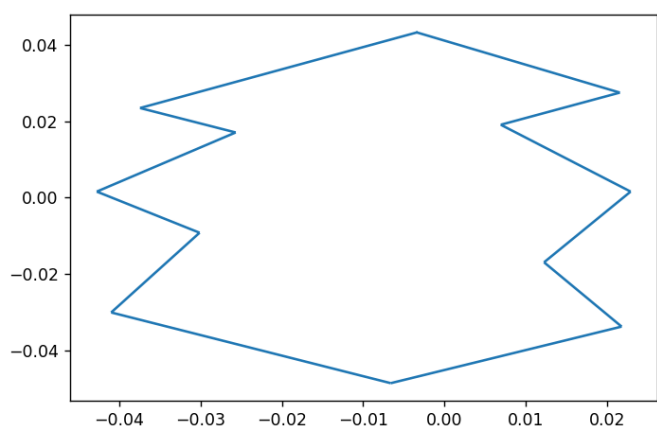
rys 4.2a



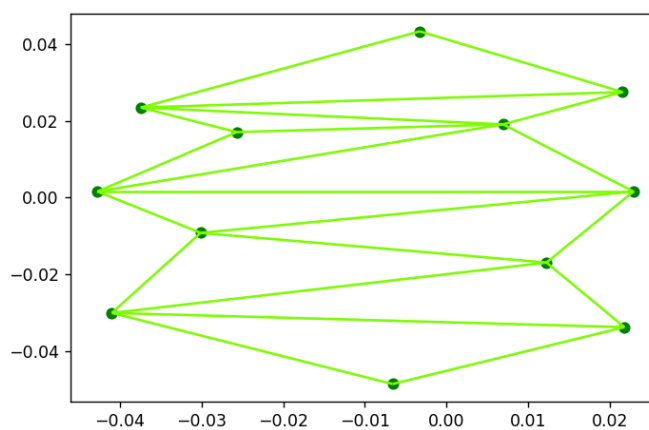
rys 4.2b



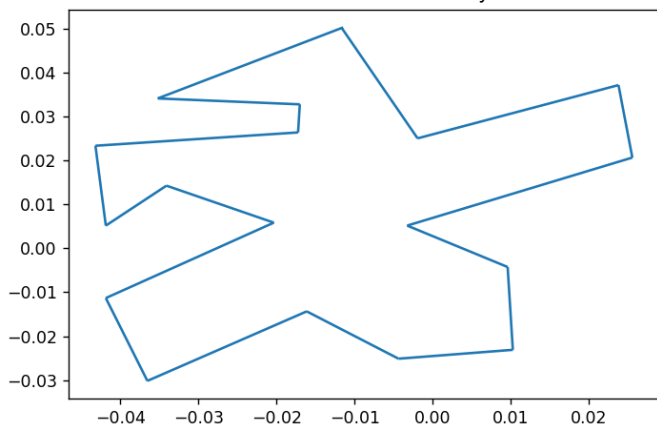
rys 4.3a



rys 4.3b

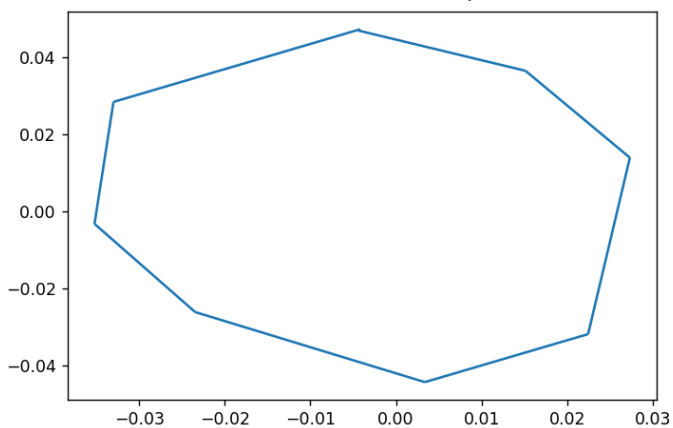


rys 4.4a

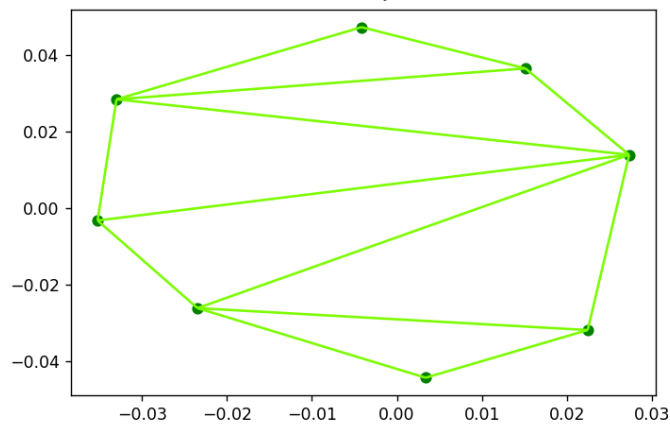


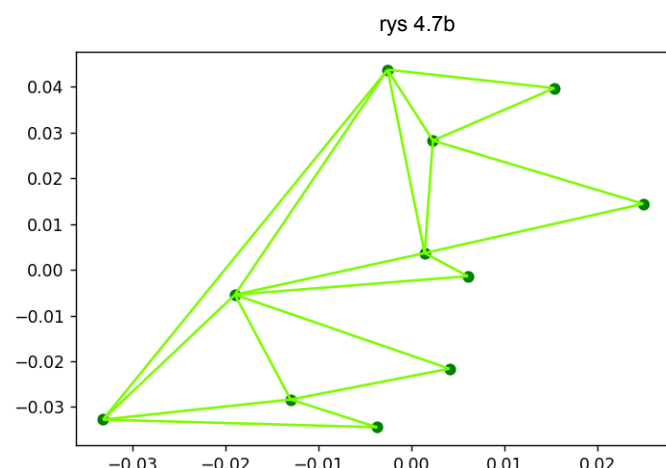
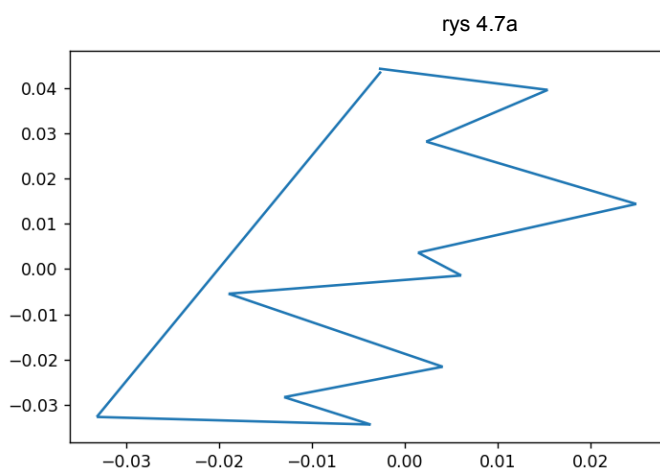
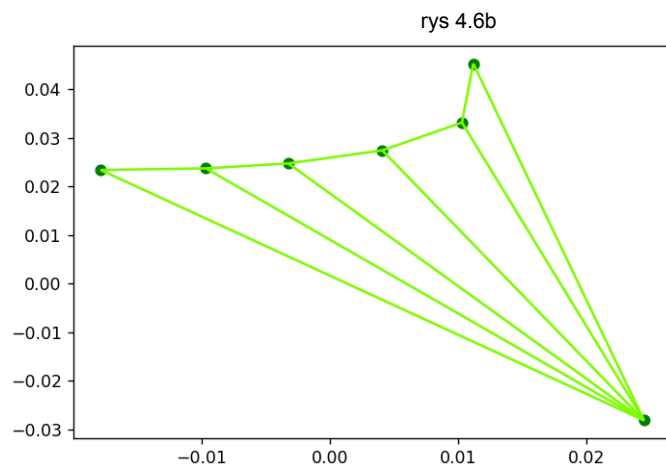
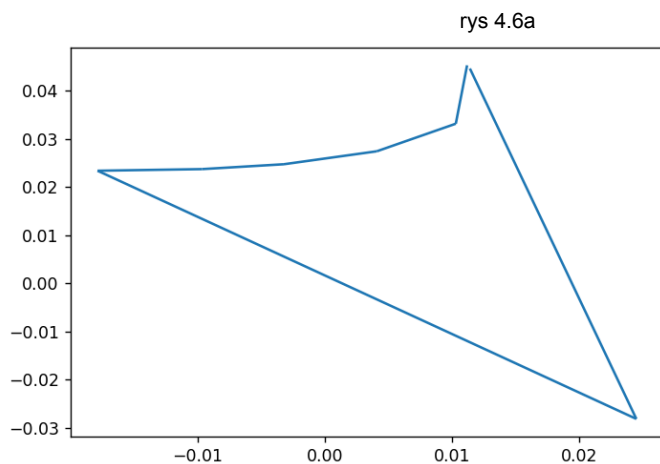
Wielokąt nie jest monotoniczny, brak możliwości triangulacji

rys 4.5a



rys 4.5b





4.2 Podsumowanie

Powyższe wielokąty zostały wybrane do testowania poprawności algorytmu, ponieważ zróżnicowana budowa każdego z nich pozwalała na sprawdzenie prawidłowego działania poszczególnych części algorytmu takich jak sprawdzanie czy potencjalna linia będzie znajdować się wewnątrz wielokąta oraz jego wyjątków np. monotoniczności rozpatrywanej figury.

Podczas pracy algorytmu wielokąt oraz utworzona triangulacja przechowywana jest w listach ze względu na jego implementację, wymagającą m.in. przetwarzania zbioru punktów wielokąta lub dostęp do punktów znajdujących się w poszczególnych indeksach. Wybrana struktura w prosty sposób zapewnia możliwość stosowania wymienionych wyżej operacji. Algorytm zwraca strukturę umożliwiającą wizualizację w narzędziu graficznym oraz listę boków i przekątnych wielokąta powstałych w procesie triangulacji.

Implementacja wszystkich wymienionych wyżej funkcji i algorytmów oraz wizualizacja procesu triangulacji znajduje się w pliku załączonym ze sprawozdaniem.