Algorytmy Geometryczne

Laboratorium 2 – Sprawozdanie

Jan Smółka

1. Sposób wykonania zadania

1.1. Kod programu

Do rozwiązania zadania posłużono się językiem programowania Julia, wraz z wbudowaną biblioteką do tworzenia wykresów.
Kod został podzielony na pakiety *generators.jl, utilities.jl, graham.jl, jarvis.jl* i *hull.jl.* Odpowiadają one za odpowiednio generowanie punktów, sortowanie i inne geometryczne procedury, implementację algorytmów Grahama i Jarvisa oraz integrację wszystkich poprzednich procedur i rysowanie wykresów.
W implementacjach przyjęto sprzętową tolerancję dla 0, implikowaną zastosowanym typem reprezentacji liczb rzeczywistych Float64.
Implementacje operują wyłącznie na zbiorach punktów o dodatnich rzędnych. W celu uzyskania takich danych, losowo generowane punkty o zadanych kształtach są przekształcane w translacji o wektor, a skonstruowana na przetransformowanych punktach otoczka jest przekształcana w translacji odwrotnej, stając się otoczką pierwotnego zbioru.

1.2. Metoda badania

Każdy z zadanych zbiorów punktów zbadano przy użyciu obu algorytmów. Nie zaobserwowano rozbieżności w poprawności działania, za wyjątkiem zbioru D, w którym przyjęcie sprzętowej tolerancji prowadzi do uzyskania osobliwych wyników. Zwiększenie tolerancji rozwiązuje ten problem.
Pomiary czasu działania wykonano za pomocą wbudowanej biblioteki Benchmark.

1.3. Wykresy

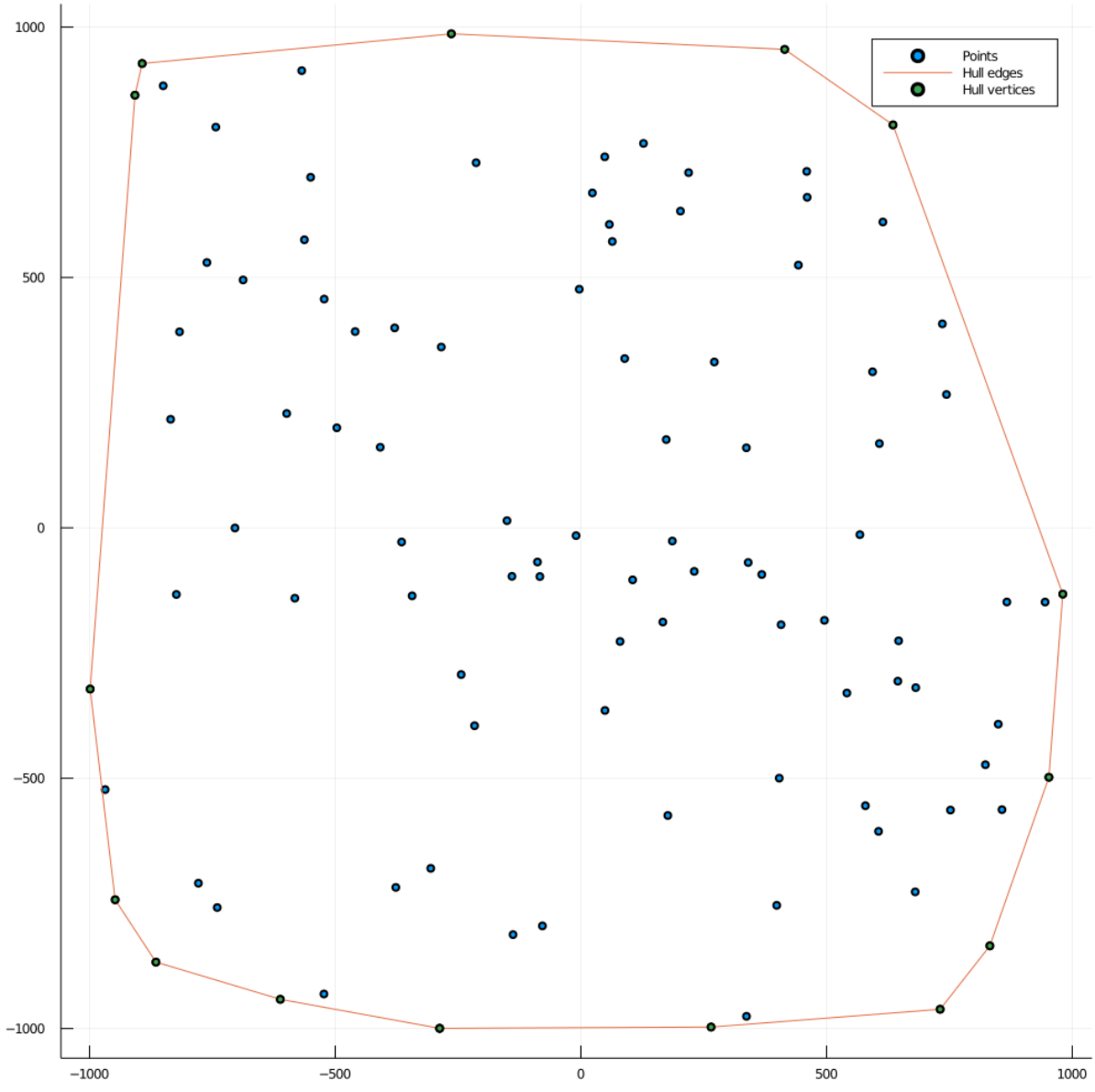
Dla każdego zbioru punktów sporządzono jeden wykres, uzyskany przy pomocy jednej metody wyznaczania otoczki.

1.4. Środowisko programistyczne

Do wykonania zadania użyto języka Julia w wersji 1.4, kompilowanego dedykowanym przez twórców kompilatorem. Obliczenia wykonano na komputerze z systemem Windows 11, z procesorem Intel Core i7, o częstotliwości taktowania 2 GHz.

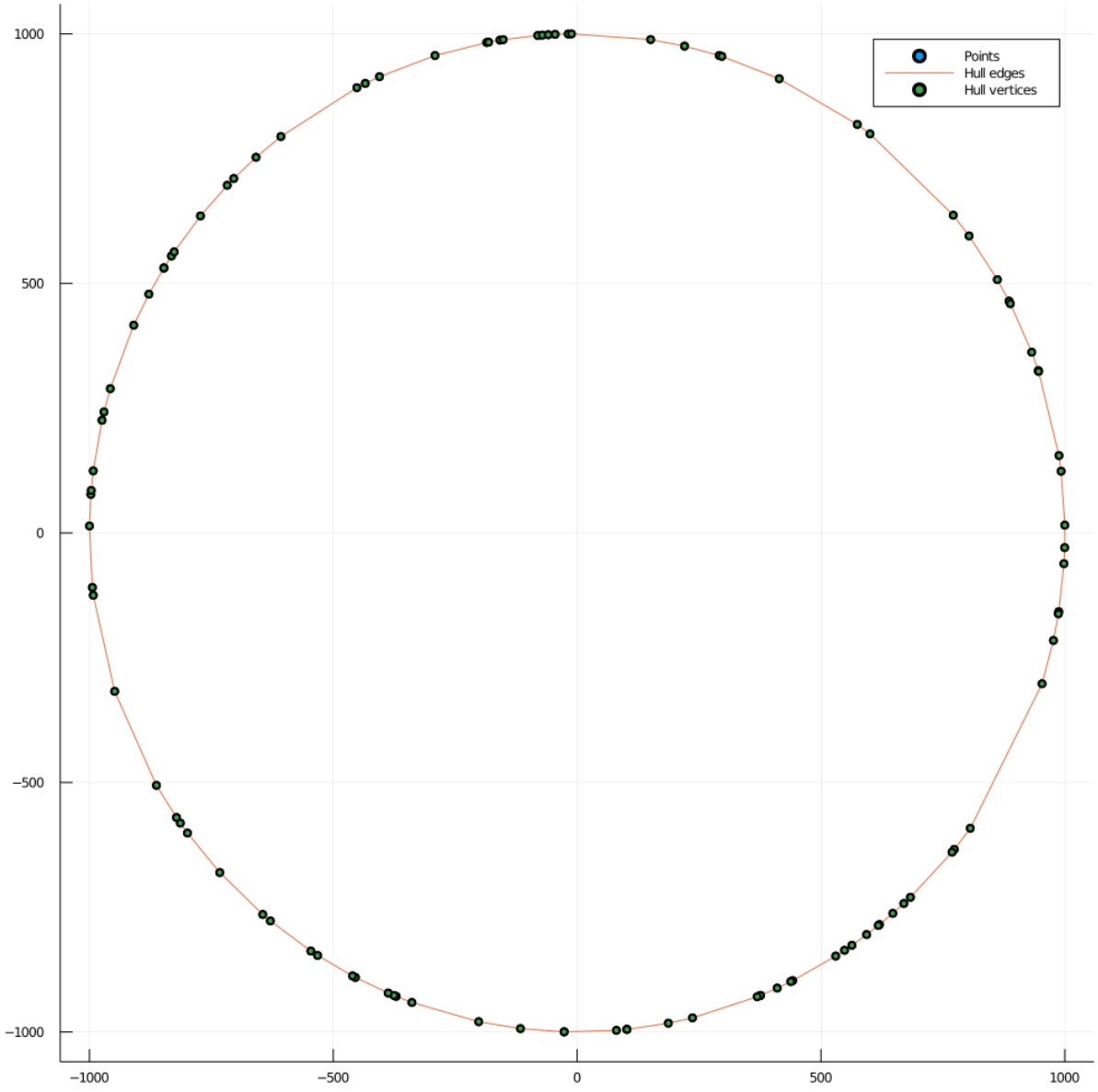
2. Wykresy

2.1. Zbiór A



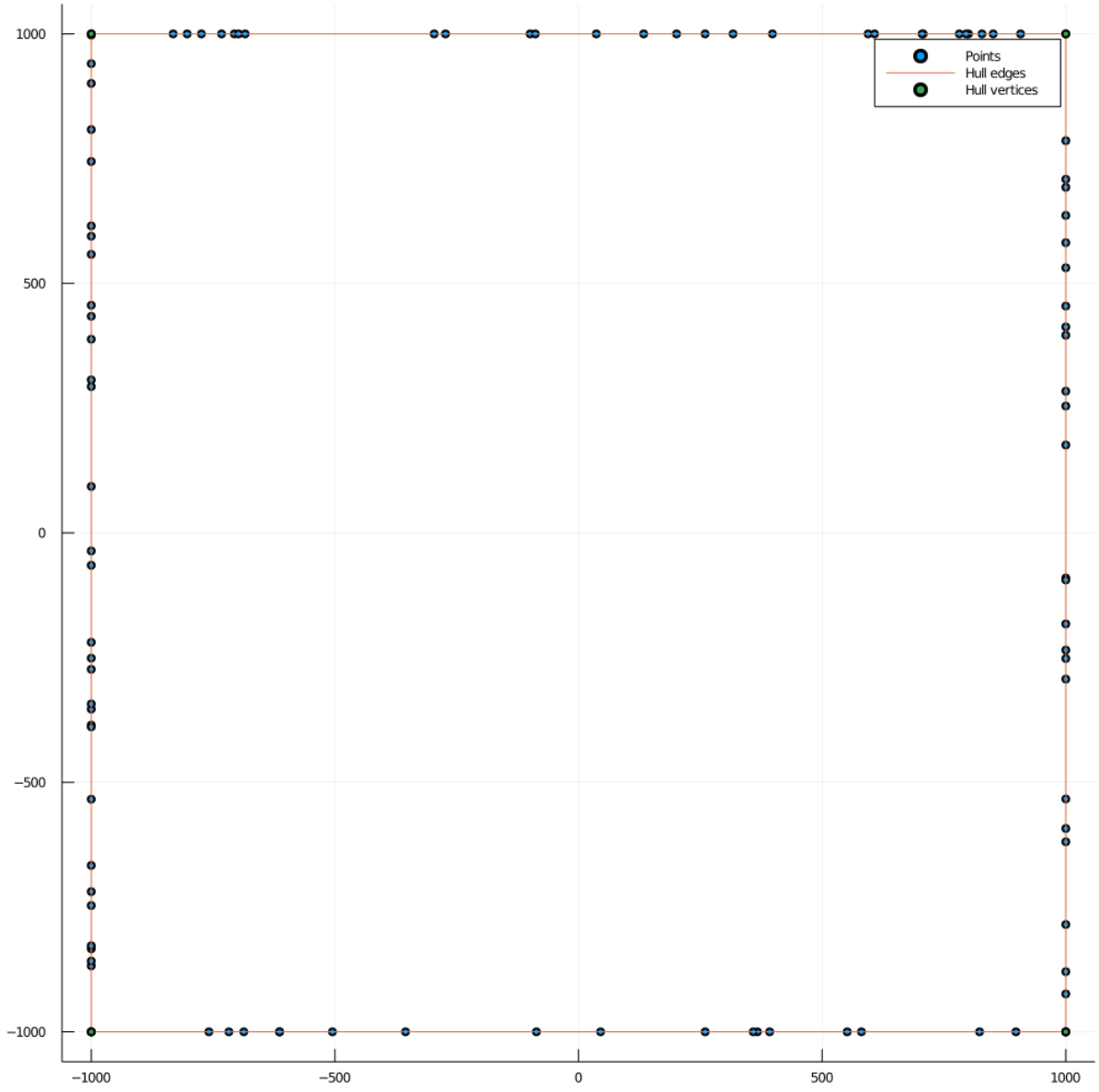
Rysunek 1 Otoczka wypukła zbioru A

2.2. Zbiór B



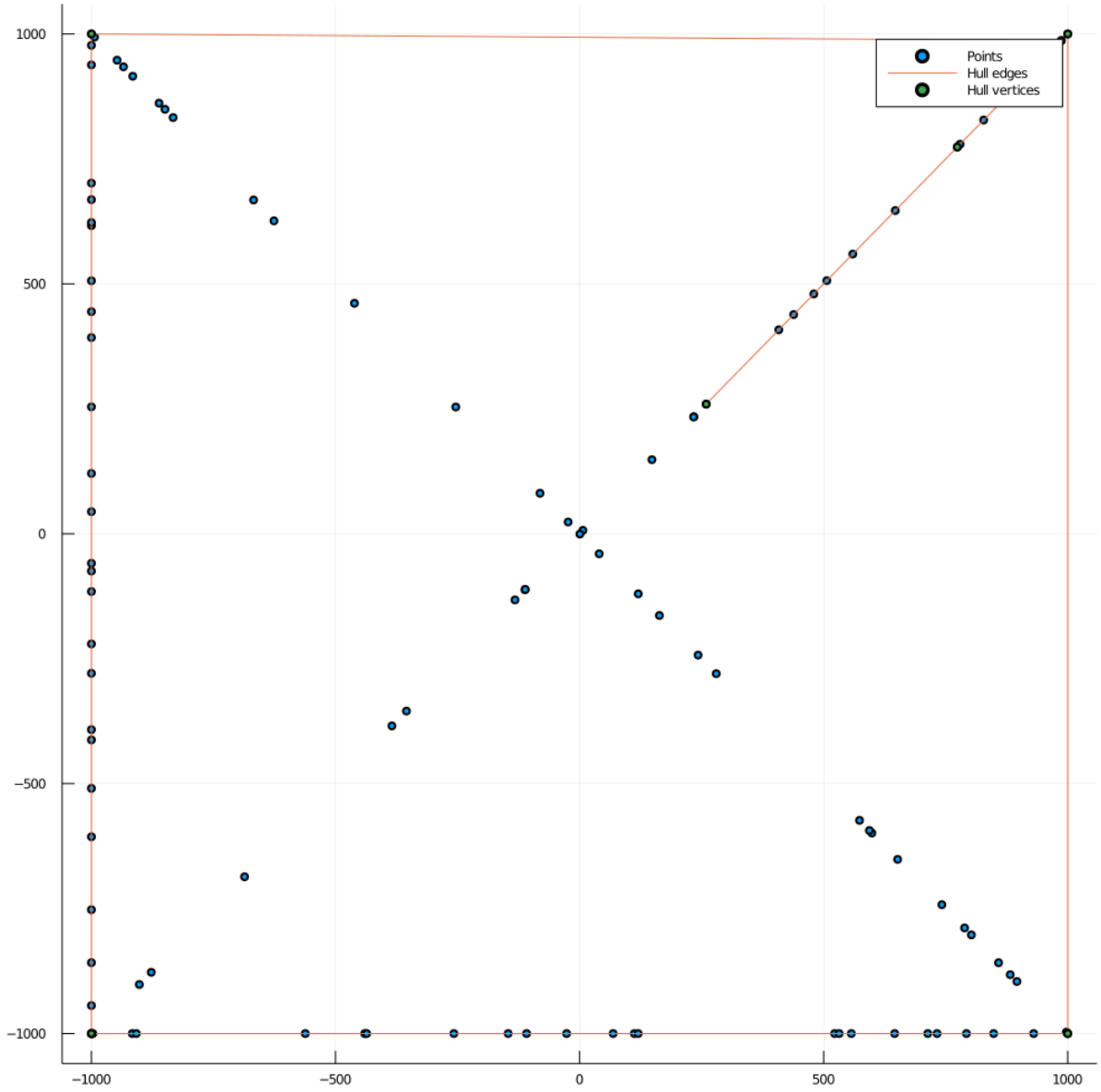
Rysunek 2 Otoczka wypukła zbioru B

2.3. Zbiór C



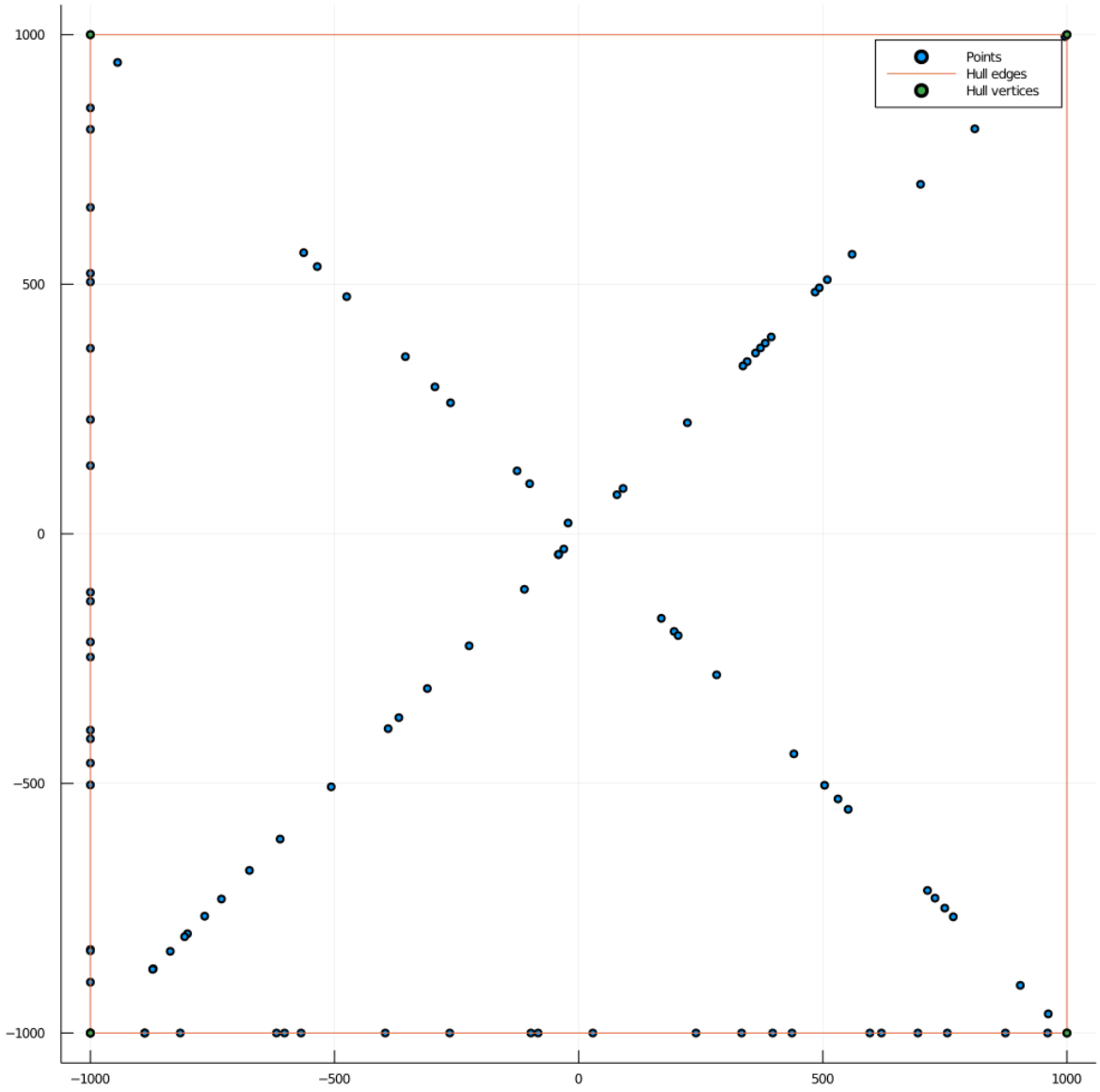
Rysunek 3 Otoczka wypukła zbioru C

2.4. Zbiór D – Algorytm Grahama z tolerancją 0.0



Rysunek 4 Otoczka wypukła zbioru D – błąd spowodowany wyborem tolerancji

2.5. Zbiór D – Algorytm Grahama z tolerancją $1 \cdot 10^{-10}$



Rysunek 5 Otoczka wypukła zbioru D - poprawny wynik algorytmu

3. Analiza działania algorytmów

3.1. Jakość generowanych wyników

Dla zbiorów A i B, oba algorytmy działają niezawodnie. Analiza wykresów nie dostarcza żadnych ciekawych wniosków. Warto jedynie zauważyć, że wszystkie punkty okręgu należą do otoczki wypukłej zbioru B, niezależnie od przyjętej ilości punktów i średnicy zbioru.
Dla efektywnego działania w zbiorze C zmodyfikowano algorytm Grahama, zmieniając jego zachowanie w sytuacji, gdy bieżący punkt jest współliniowy z dwoma ostatnimi znajdującymi się na stosie. Wówczas ostatni punkt ze stosu jest zastępowany bieżącym punktem, a algorytm przechodzi do analizy następnego punktu posortowanej listy. Po dokonaniu tego usprawnienia zaobserwowano znaczne polepszenie jakości otrzymywanych wyników.
Dodatkowo można zauważyć, że sporadycznie punkty otoczki należą do boków kwadratu. Wynika to z przyjętej dokładności względem 0.
W przypadku użycia algorytmu Grahama w zbiorze D zaobserwować można błędy w wyznaczaniu otoczki – punkty leżące na przekątnych kwadratu są błędnie kwalifikowane jako należące do niej. Problem ten wynika z przyjętej sprzętowej tolerancji dla 0. Przyjęcie innej, większe, zapobiega występowaniu tego rodzaju osobliwości. Różnicę można zaobserwować na rysunku 4. i 5. Problem ten nie występuje w algorytmie Jarvisa, dlatego działa on z arbitralną tolerancją sprzętową.

3.2. Czas działania

Tabela 1. – Porównanie czasu działania algorytmów

Zbiór, liczność		Czas działania algorytmu Grahama	Czas działania algorytmu Jarvisa
A	$1 \cdot 10^4$	0.35 s	0.40 s
	$1 \cdot 10^5$	4.03 s	3.78 s
	$1 \cdot 10^6$	56.24 s	57.24 s
B	$1 \cdot 10^4$	0.40 s	134.93 s
	$1 \cdot 10^5$	4.76 s	-
	$1 \cdot 10^6$	79.86 s	-
C	$1 \cdot 10^4$	0.48 s	0.11 s
	$1 \cdot 10^5$	5.67 s	1.10 s
	$1 \cdot 10^6$	46.52 s	7.90 s
D	$1 \cdot 10^4$	0.34 s	0.10 s
	$1 \cdot 10^5$	3.76 s	0.69 s
	$1 \cdot 10^6$	39.72 s	5.62 s

3.3. Dobór algorytmu do zbioru

Już na poziomie teoretycznym, analizując złożoność czasową algorytmów, można spekulować, iż algorytm Grahama lepiej sprawdzi się dla zbiorów dwuwymiarowych, takich jak B, a algorytm Jarvisa dla takich zbiorów, w których mniejszość punktów należy do otoczki. Powyższe ustalenia teoretyczne znajdują potwierdzenie w czasie działania implementacji.