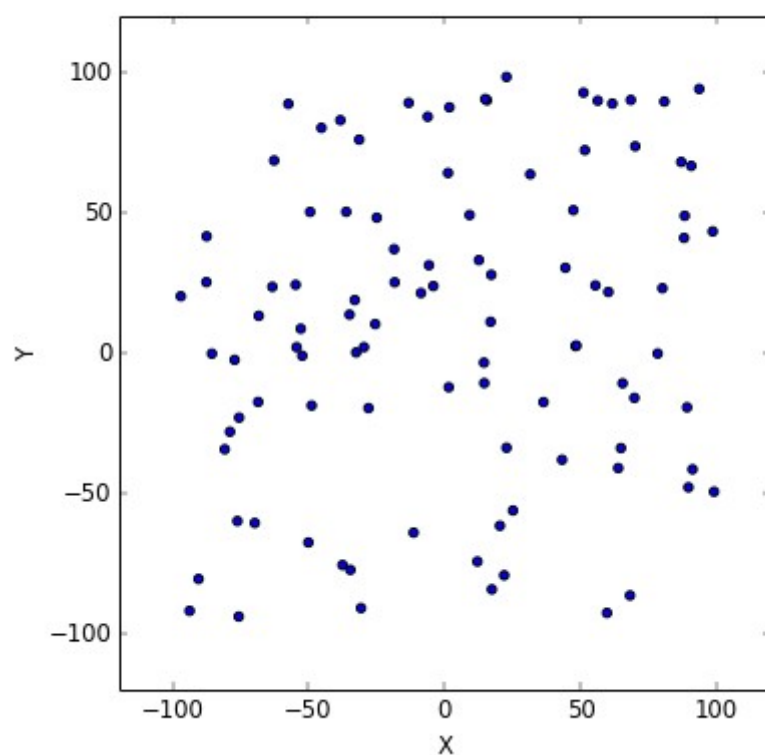


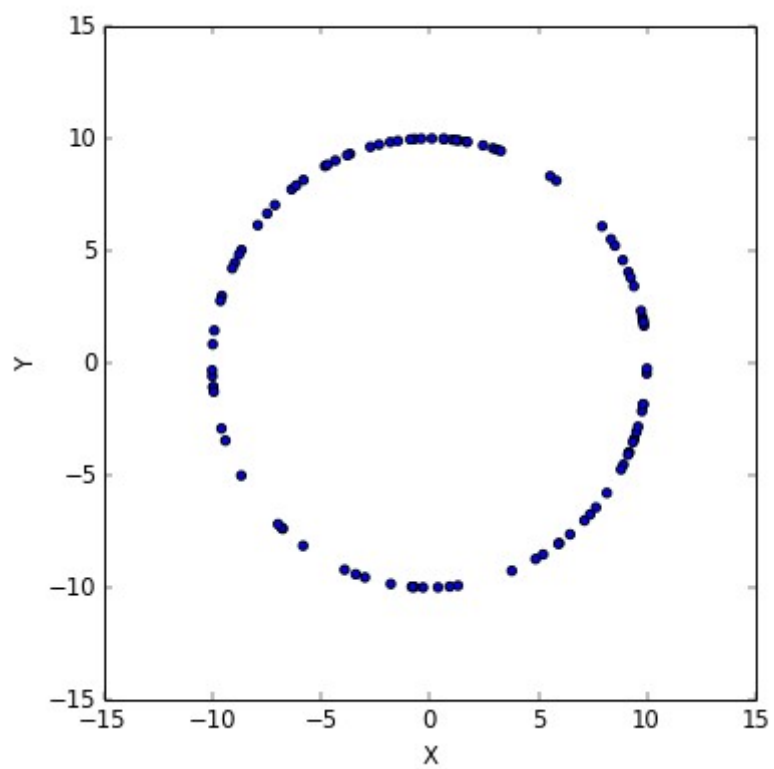
# Geometria obliczeniowa

Ćwiczenie drugie  
Piotr Oramus

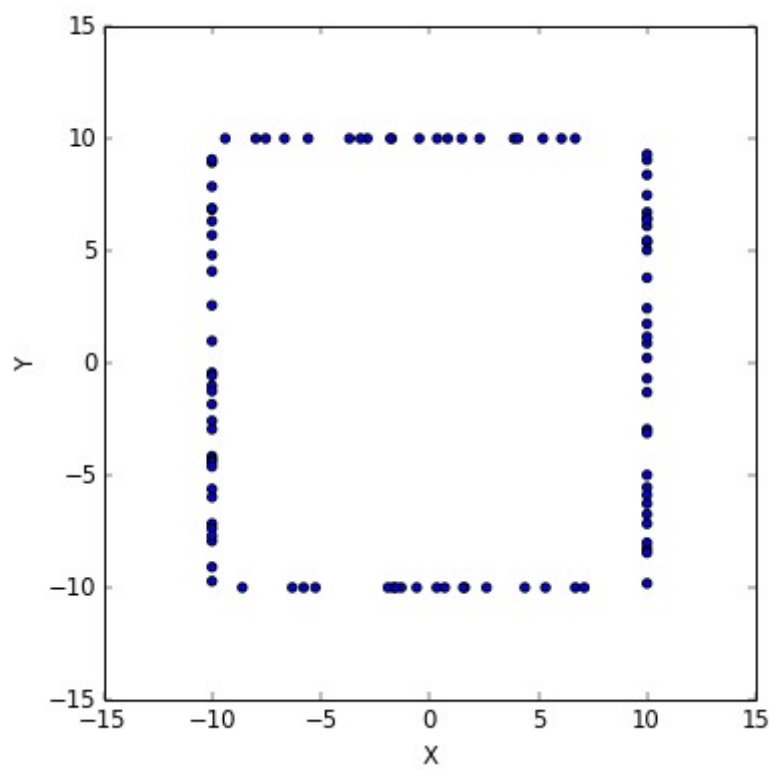
## 1. Ilustracja wygenerowanych punktów



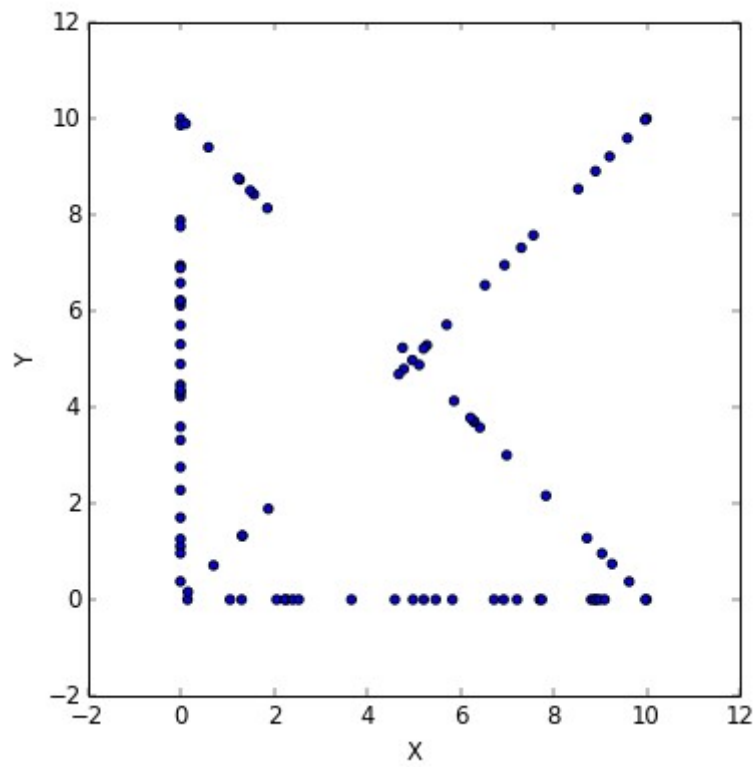
*Rysunek 1: 100 losowo wygenerowanych punktów o współrzędnych z przedziału  $[-100;100]$*



*Rysunek 2: 100 losowo wygenerowanych punktów leżących na okręgu o środku (0,0) i promieniu 10*

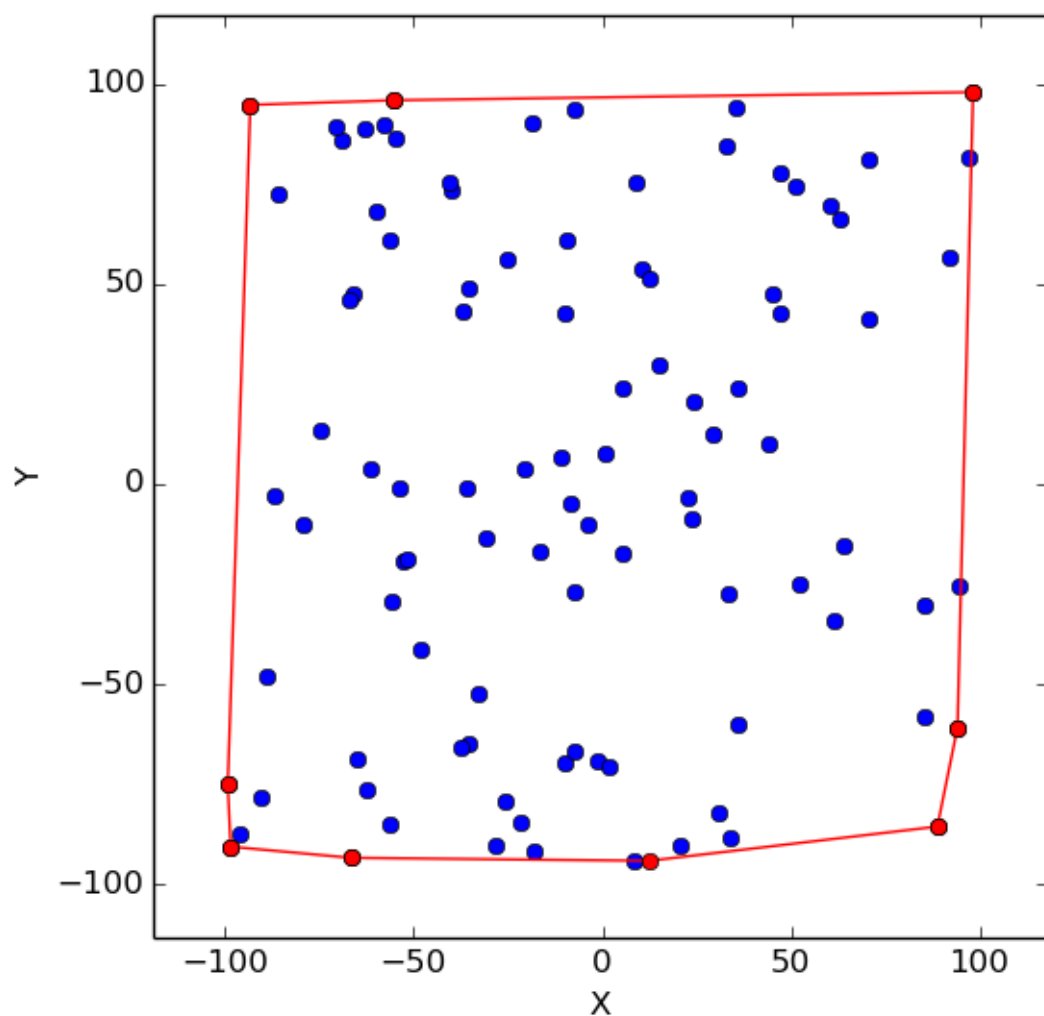


*Rysunek 3: 100 losowo wygenerowanych punktów leżących na bokach czworokąta*

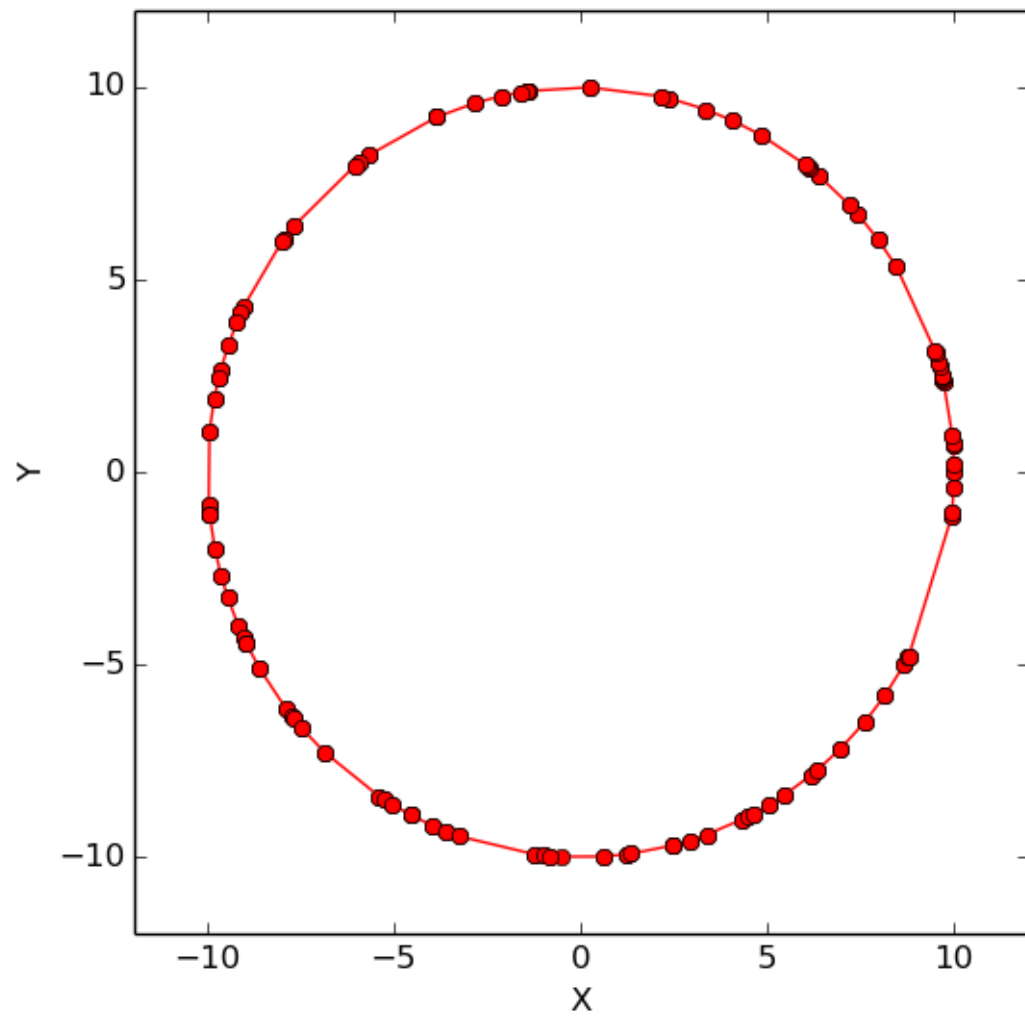


*Rysunek 4: Losowo wygenerowane punkty zawierające wierzchołki kwadratu, punkty na osiach  $XY$  oraz punkty na przekątnych kwadratu*

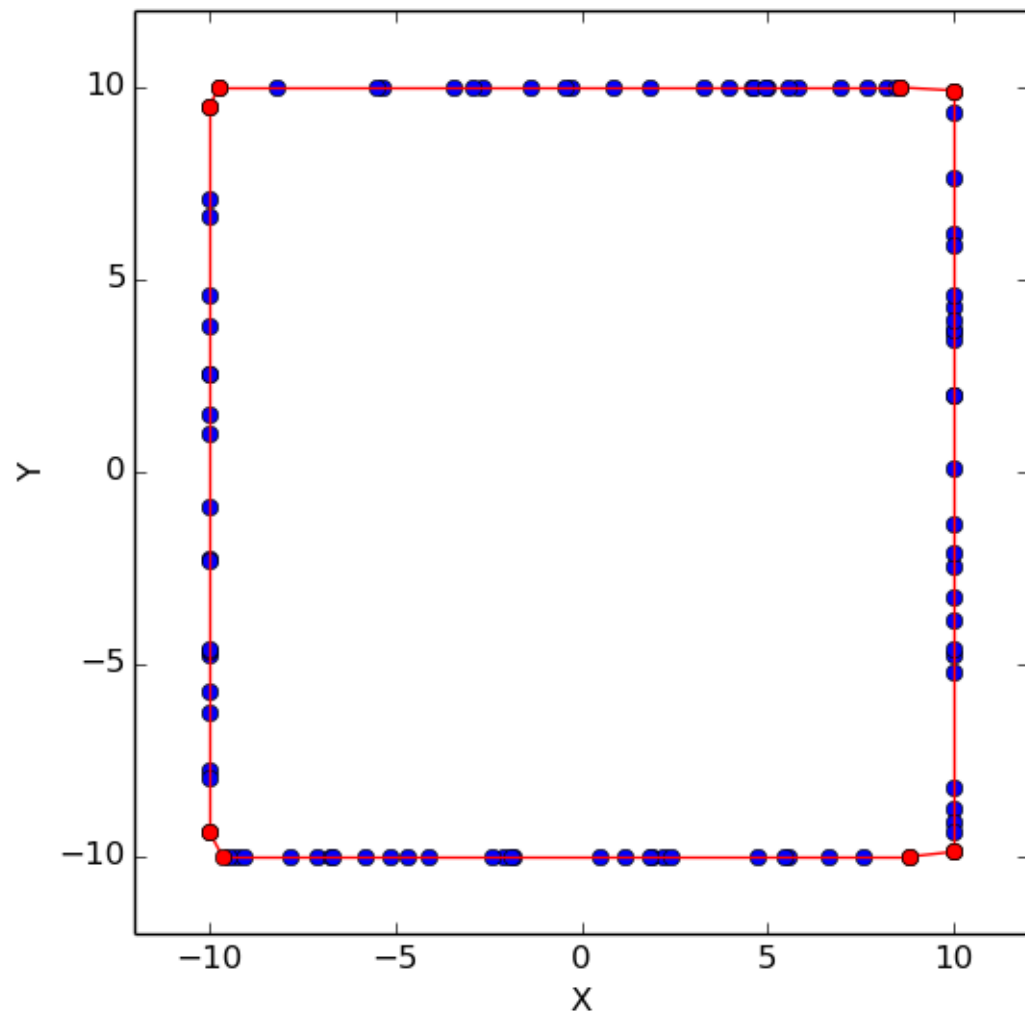
## 2. Otoczka wypukła wyznaczona algorytmem Grahama



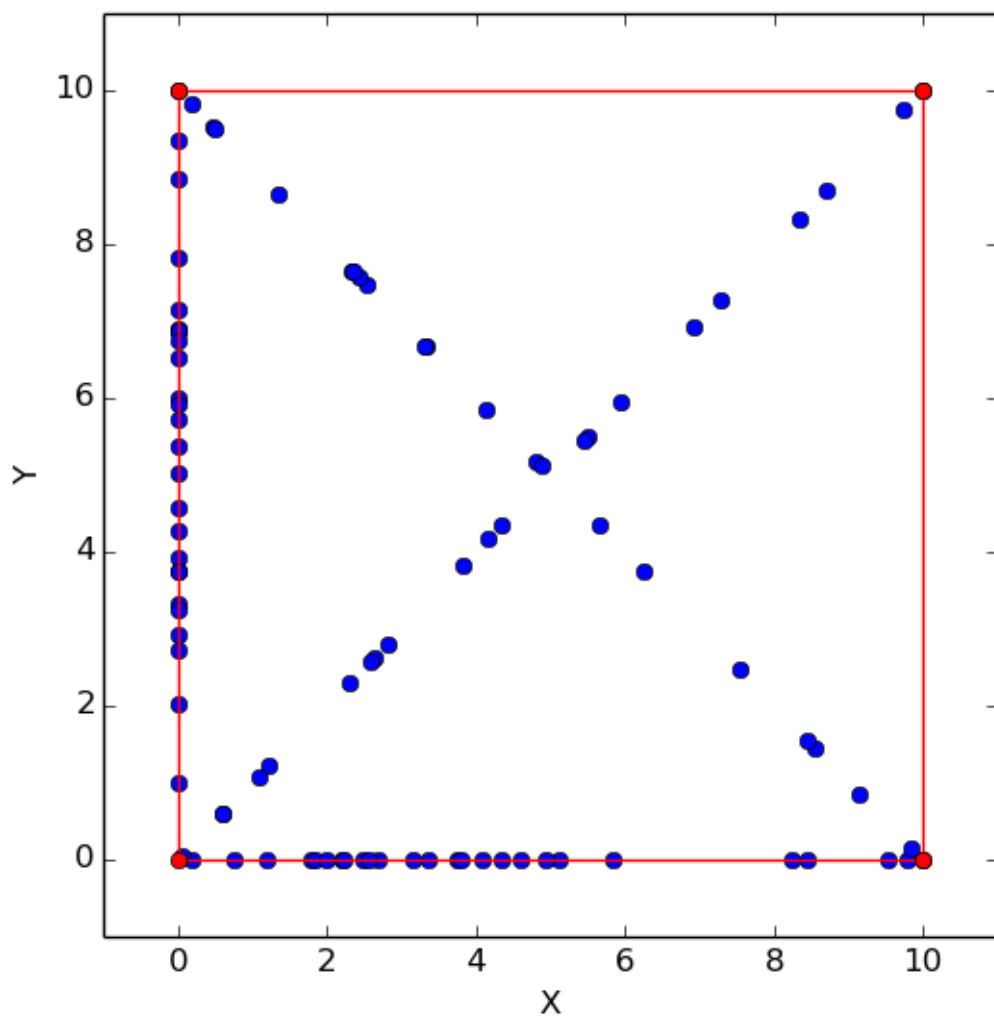
Rysunek 5: Otoczka wypukła wyznaczona dla 100 losowo wygenerowanych punktów o współrzędnych z przedziału  $[-100;100]$  algorytmem Grahama



*Rysunek 6: Otoczka wypukła wyznaczona dla 100 losowo wygenerowanych punktów leżących na okręgu o środku (0,0) i promieniu 10 algorytmem Grahama*



Rysunek 7: Otoczka wypukła wyznaczona dla 100 losowo wygenerowanych punktów leżących na bokach czworokąta algorytmem Grahama

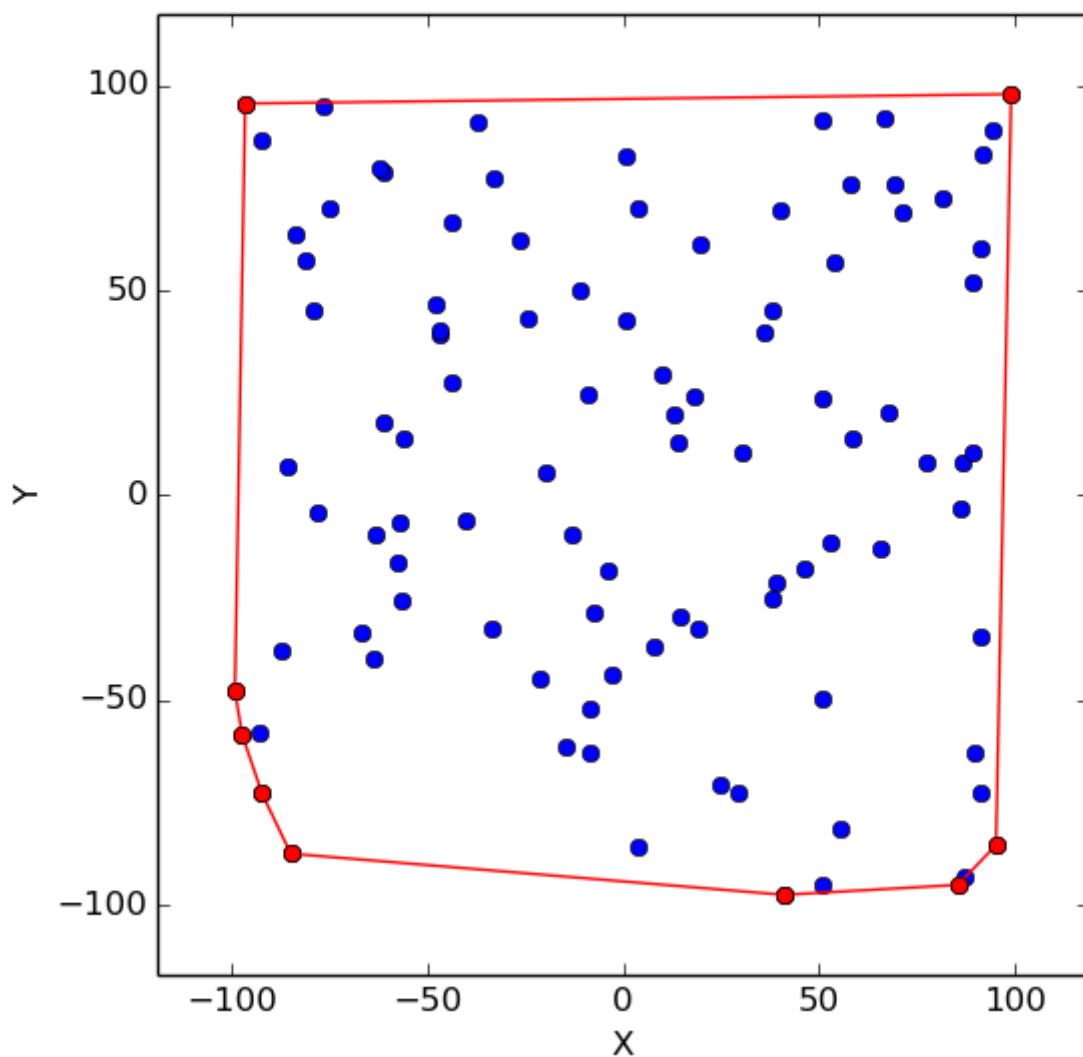


*Rysunek 8: Otoczka wypukła wyznaczona dla losowo wygenerowanych punktów zawierające wierzchołki kwadratu, punkty na osiach  $XY$  oraz punkty na przekątnych kwadratu algorytmem Grahama*

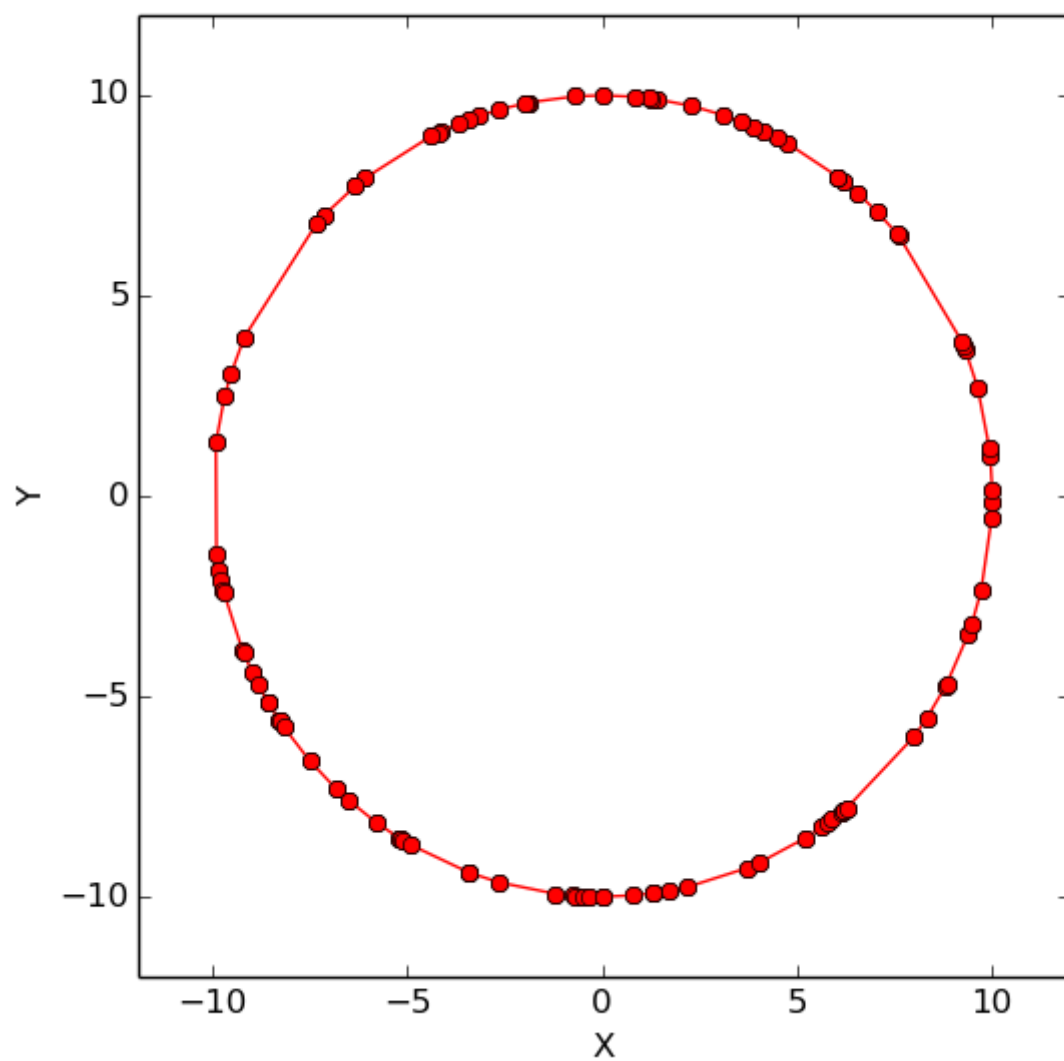


### 3. Otoczka wypukła wyznaczona algorytmem Jarvisa

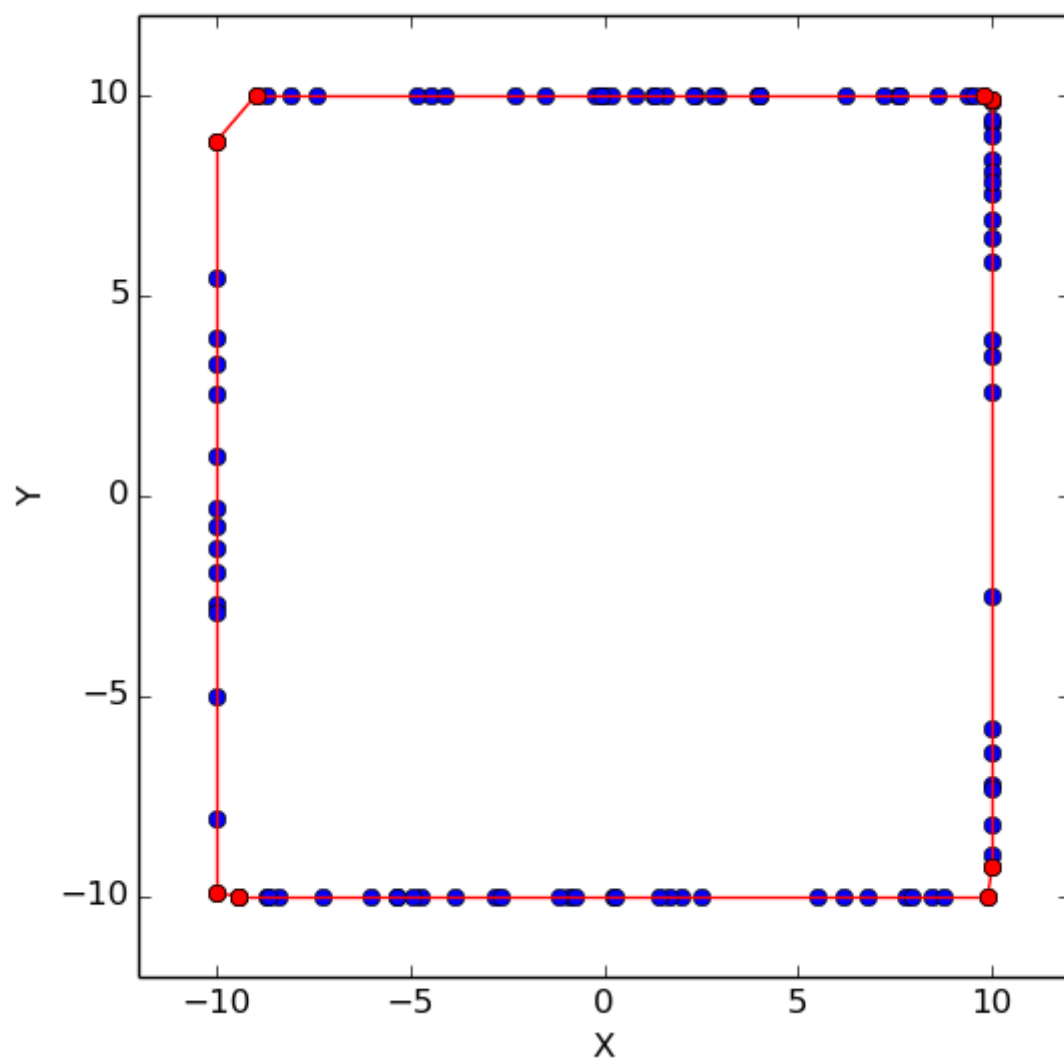
*Uwaga: punkty, których wizualizacja widnieje poniżej, nie są tymi samymi punktami, na których wizualizowany był algorytm Grahama.*



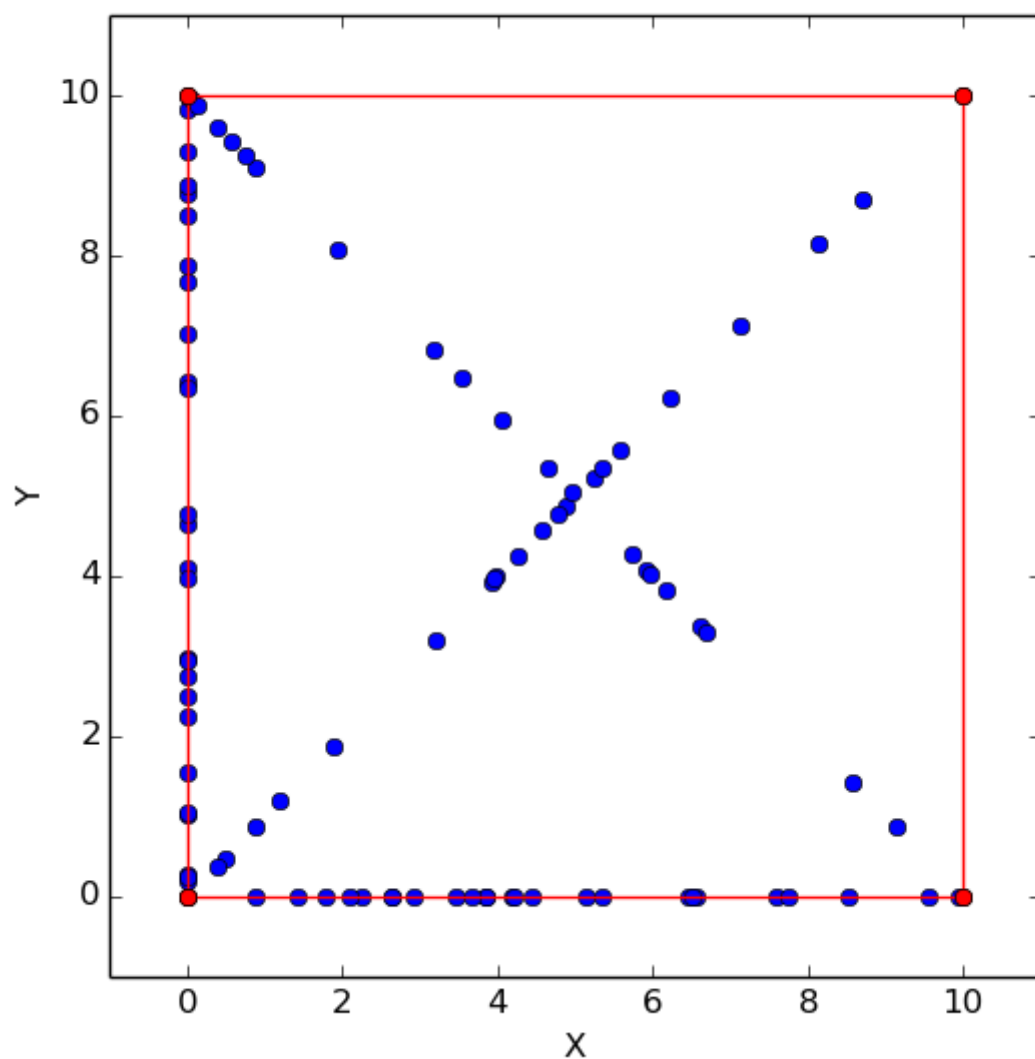
Rysunek 9: Otoczka wypukła wyznaczona dla 100 losowo wygenerowanych punktów o współrzędnych z przedziału  $[-100;100]$  algorytmem Jarvisa



Rysunek 10: Otoczka wypukła wyznaczona dla 100 losowo wygenerowanych punktów leżących na okręgu o środku  $(0,0)$  i promieniu 10 algorytmem Jarvisa



Rysunek 11: Otoczka wypukła wyznaczona dla 100 losowo wygenerowanych punktów leżących na bokach czworokąta algorytmem Jarvisa



*Rysunek 12: Otoczka wypukła wyznaczona dla losowo wygenerowanych punktów zawierające wierzchołki kwadratu, punkty na osiach XY oraz punkty na przekątnych kwadratu algorytmem Jarvisa*

#### 4. Porównanie czasów działania algorytmów Grahama i Jarvisa

Uwaga: w tym przypadku algorytmy działają z dokładnie tych samych danych. Dodatkowo program nie próbuje niczego rysować, więc czas wykonania jest wyłącznie czasem wyznaczania otoczki.

A – 100 punktów z przedziału  $[-100;100]$

B – 100 punktów okręgu  $(0,0)$ ,  $R = 10$

C – 100 punktów leżących na boku czworokąta  $(-10,10)$ ,  $(-10,-10)$ ,  $(10,-10)$ ,  $(10,10)$

D – wierzchołki kwadratu, 25 punktów na dwóch bokach oraz 20 punktów na osiach kwadratu  $(0,0)$ ,  $(0,10)$ ,  $(10,0)$ ,  $(10,10)$

	Algorytm Grahama [s]	Algorytm Jarvisa [s]
A	0.00131607055664	0.000828981399536
B	0.00114393234253	0.000869035720825
C	0.00110602378845	0.000808954238892
D	0.00119209289551	0.000807046890259

Dla większej ilości punktów w przypadku A oraz B wyniki są proporcjonalne. Niestety z niewiadomych dla mnie powodów program nie jest w stanie wyznaczyć otoczki dla dużej ilości punktów ( $> 1000$ ) w wariantach C oraz D.

#### 5. Wnioski

Pewnym zaskoczeniem może być, że Algorytm Jarvisa jest szybszy od algorytmu Grahama, pomimo większej teoretycznej złożoności obliczeniowej. Dzieje się tak dla każdych testowanych danych. Być może wynika to ze sposobu implementacji.

Dla danych z punktu 1. tematu zadania zarówno algorytm Grahama jak i Jarvisa działały poprawnie. Pierwszy zestaw dobrze pokazuje działanie obu algorytmów (w przypadku animacji). Drugi zestaw jest przypadkiem szczególnym, gdzie wszystkie punkty należą do otoczki, co dobrze testuje algorytm. Trzeci zestaw punktów dobrze obrazuje, czy nasz algorytm “umie” pomijać punkty współliniowe. Do tego czwarty dokłada “zakłócenia” w postaci punktów na przekątnych. Te szczególne przypadki pozwalają w kompleksowy sposób przetestować poprawność algorytmów.