

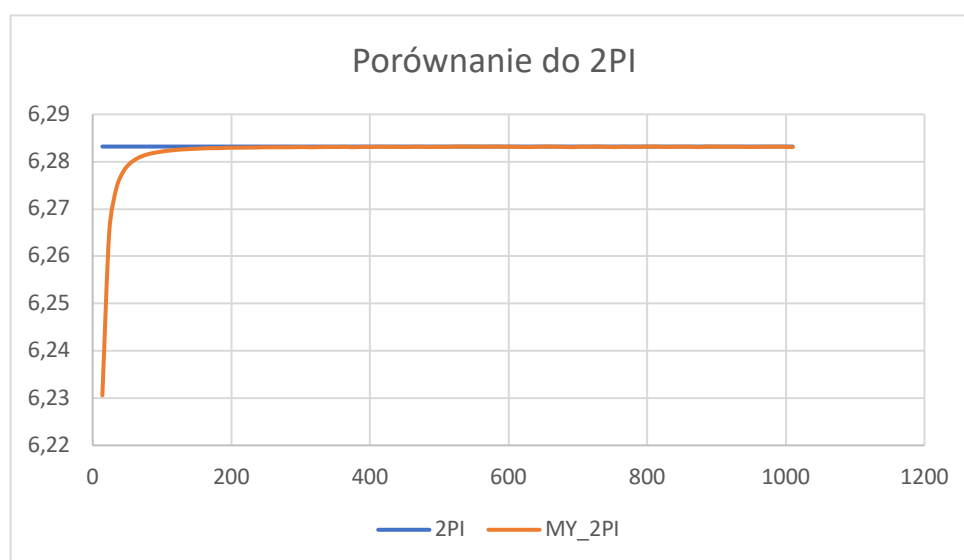
Programowanie Liniowe
Projekt 1
Michał Safuryn 288574

H1. Zwiększając n można uzyskać obwód dowolnie bliski liczbie 2Π .

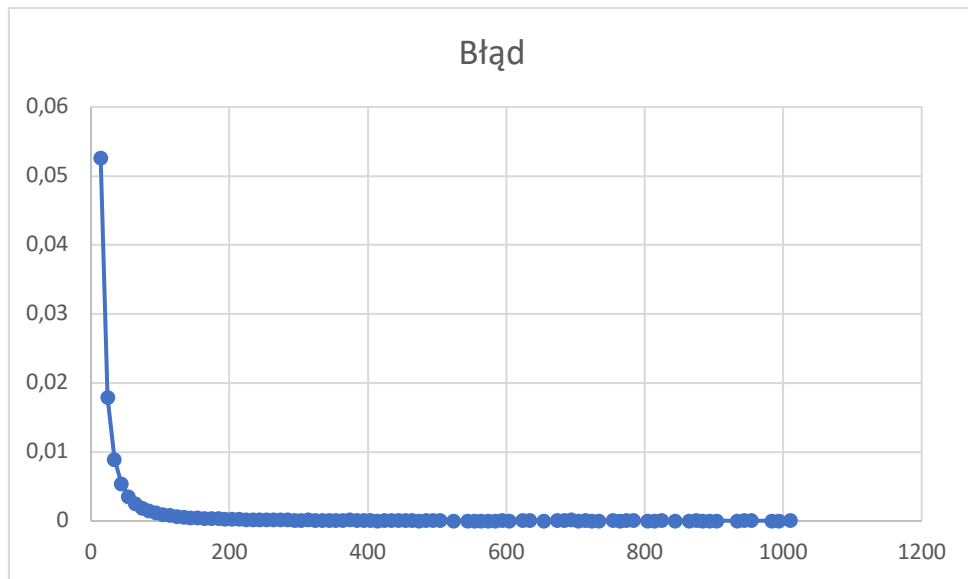
Nie zwiększając n można uzyskać obwód dowolnie bliski liczbie 2Π .

Przykład danych:

N	2PI	MY_2PI
14	6,28319	6,23059
24	6,28319	6,26526
34	6,28319	6,27425
44	6,28319	6,27785
54	6,28319	6,27964
64	6,28319	6,28066
74	6,28319	6,2813
84	6,28319	6,28172
94	6,28319	6,282
104	6,28319	6,28223
114	6,28319	6,28238



Powyższy wykres przedstawia jak zmienia się 2Π wyliczone z sumy długości wektorów. Można zauważyć, że od około 200-wierzchołkowego wielokąta 2Π jest bardzo blisko tej stałej. Jednak dla większych N to Π zaczyna się zmieniać i oscyluje w okolicy 6.24-6.32



Na wykresie powyżej można zobaczyć błędy, różnicę między stałą 2π , a wyliczaną. Podsumowując, Nie hipoteza jest prawdziwa.

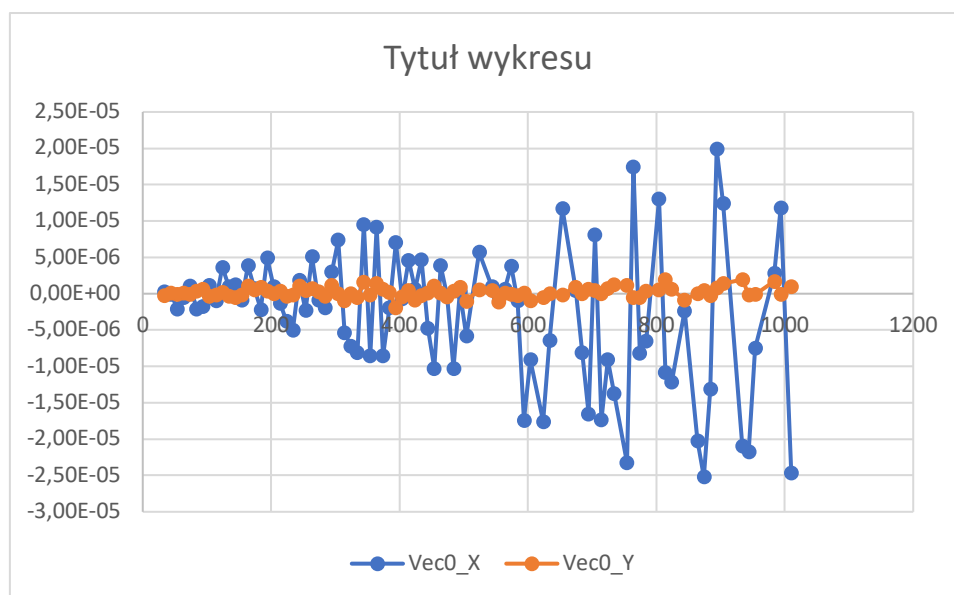
H2. Suma wszystkich wektorów w_i daje dokładnie wektor zerowy.

Nie, nie daje ona dokładnie wektora zerowego. Daje ona natomiast bardzo blisko wektorowi zerowemu

Przykład danych:

N	Vec0_X	Vec0_Y
14	-1,64E-07	2,98E-07
24	-3,35E-08	5,96E-08
34	2,70E-07	-2,83E-07
44	-1,16E-07	5,96E-08
54	-2,08E-06	-1,12E-07
64	-5,46E-07	5,96E-08
74	1,06E-06	-7,45E-08
84	-2,11E-06	3,50E-07
94	-1,72E-06	6,03E-07
104	1,16E-06	-3,39E-07
114	-9,99E-07	-1,45E-07

Jak można zauważyć zmienne nie są równe dokładnie $[0, 0]$, ale są one bardzo blisko.

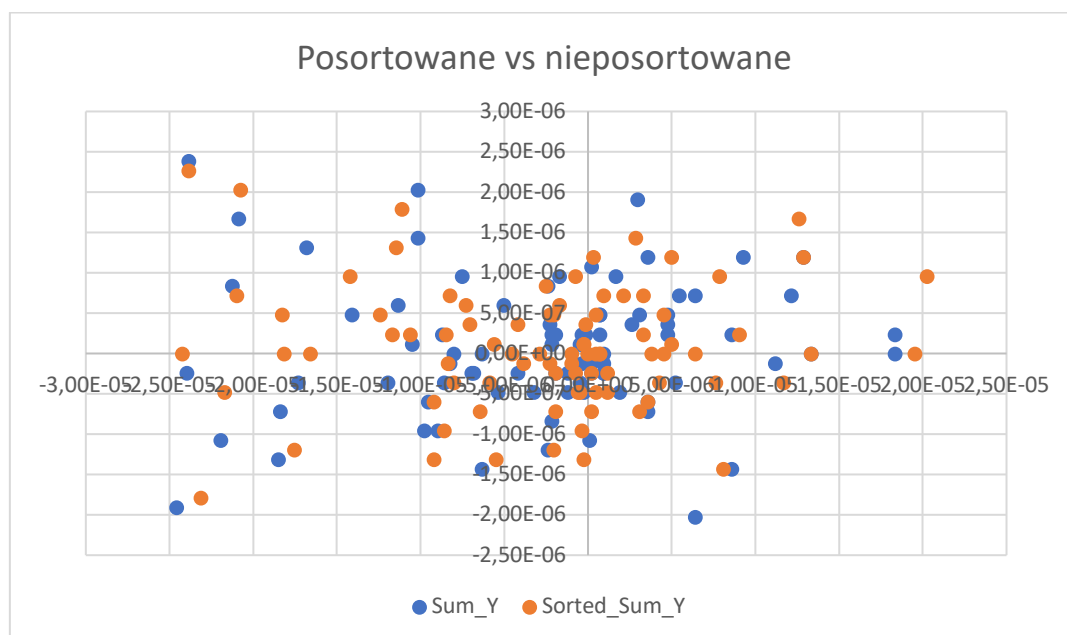


Wykres pokazuje że oscylują one w granicy 0,0. Jednak, gdy N rosną również błędy

Podsumowując, NIE, nie dają dokładnie wektora zerowego.

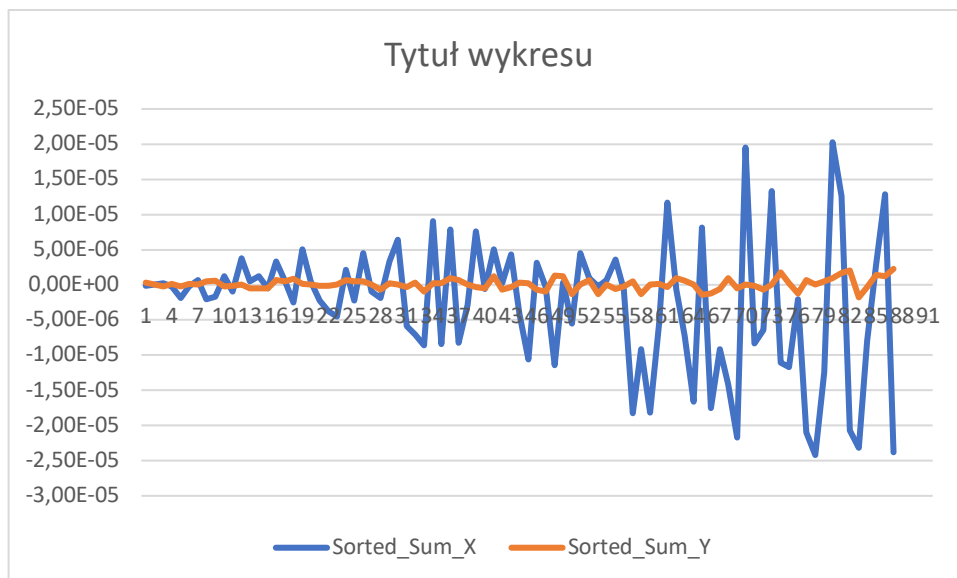
H3. Sumy współrzędnych wektorów w_i można policzyć osobno, a następująca zmiana kolejności sumowania sprawi, że wynik będzie bliższy wektorowi zerowemu.

Dla moich danych TAK.



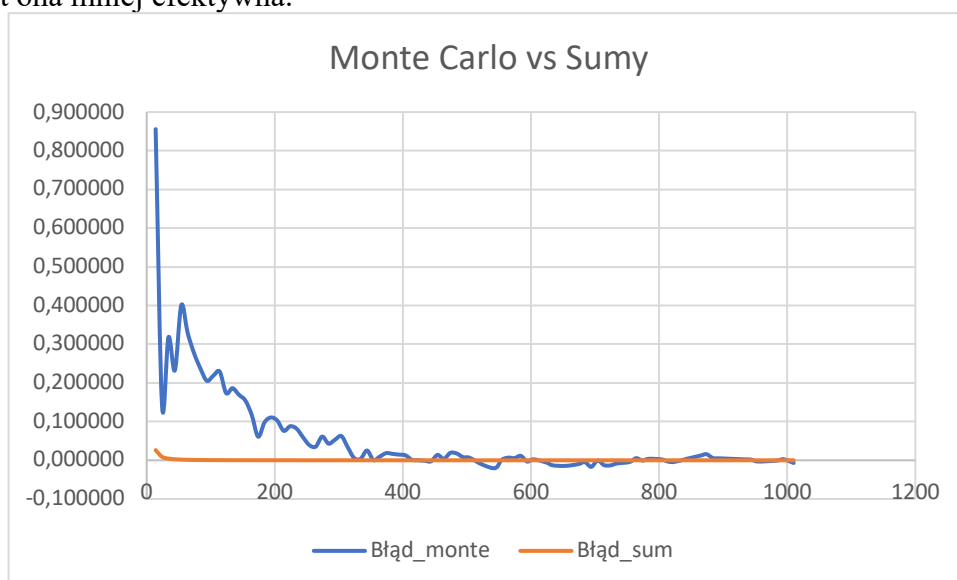
Pomarańczowe kropki przedstawiają posortowane dane i jest ich więcej bliższych [0, 0] niż tych nieposortowanych.

Podsumowując: Tak, zmieni.



H4. Opisane zastosowanie metody Monte Carlo jest mniej efektywne niż metoda oparta o sumowanie wektorów.

TAK, jest ona mniej efektywna.



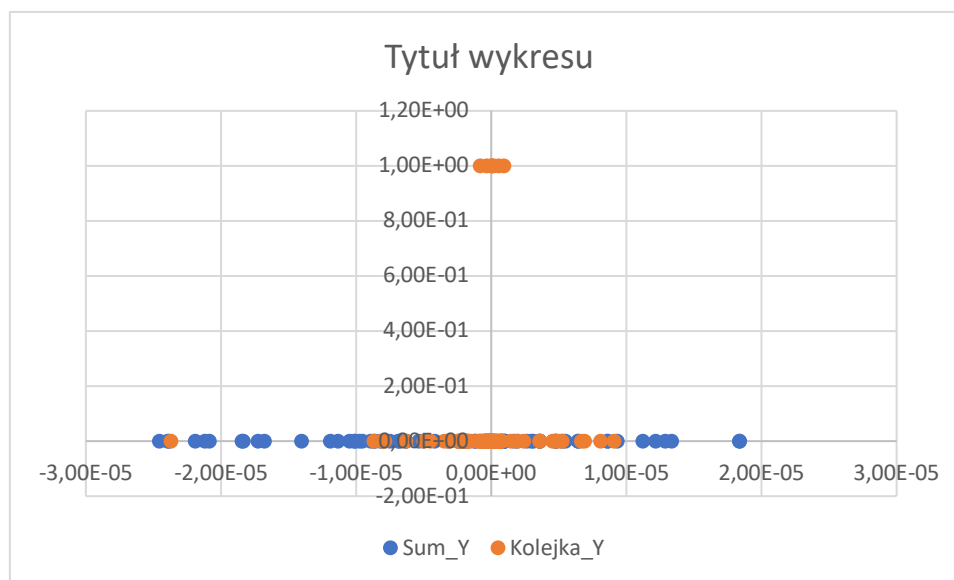
Jak łatwo można zauważyć na wykresie błędy generowane przez Monte Carlo są dużo większe dla małych danych. Dla większych stają się one porównywalne do tych wygenerowanych przez sumowanie. Jednakże, to że dla danych wielkości +/-500 generuje ona dość duże błędy można powiedzieć, że jest ona mniej efektywna.

Przykładowe dane:

N	PI	PI_Monte	Points In	MY_PI	Błąd_monte	Błąd_sum
14	3,141593	2,28571	8	3,115295	0,855883	0,026298
24	3,141593	3	18	3,13263	0,141593	0,008963
34	3,141593	2,82353	24	3,137125	0,318063	0,004468
44	3,141593	2,90909	32	3,138925	0,232503	0,002668
54	3,141593	2,74074	37	3,13982	0,400853	0,001773
64	3,141593	2,8125	45	3,14033	0,329093	0,001263
74	3,141593	2,86486	53	3,14065	0,276733	0,000943
84	3,141593	2,90476	61	3,14086	0,236833	0,000733
94	3,141593	2,93617	69	3,141	0,205423	0,000593
104	3,141593	2,92308	76	3,141115	0,218513	0,000478
114	3,141593	2,91228	83	3,14119	0,229313	0,000403

H5. Podobnie jak w H3 ale w celu sumowania każdego ze zbiorów wybieramy dwa najmniejsze (albo największe) elementy a sumę wstawiamy z powrotem do zbioru.

Używając kolejki możemy dostać lepsze wyniki, bliższe wektorowi zerowemu niż sum.



Z testowanych pkt 58/87 było bliżej 0.

