Maciej Mionskowski
Nr albumu: 296628

1. Sumowanie liczb pojedynczej precyzji

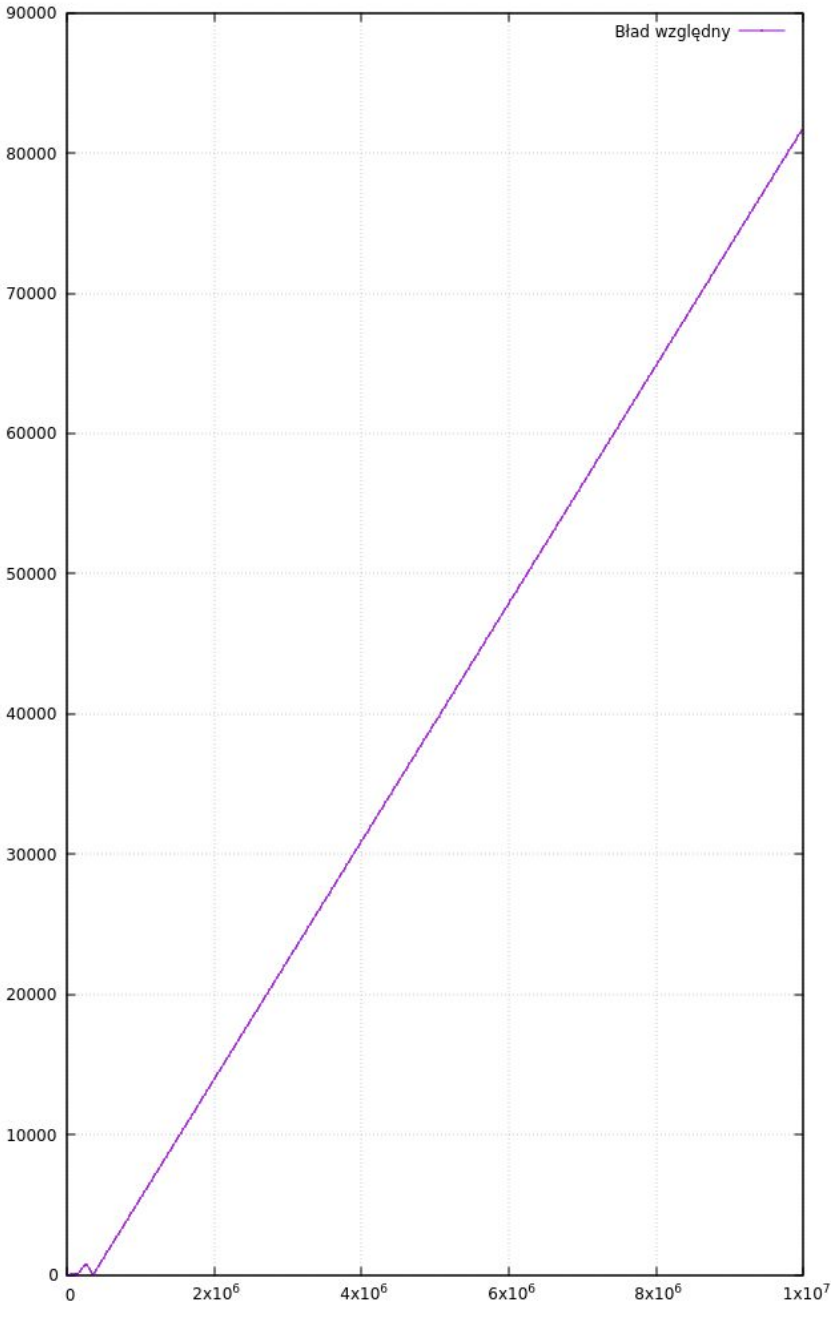
Analiza

Wykonana dla: float FILL = 0.25214f; Oczekiwany wynik: `2521399.8`

Algorytm: Standard	Algorytm: Kahan	Algorytm: Rekurencyjny
Czas wykonania: 20ms	Czas wykonania: 57ms	Czas wykonania: 42ms
Wynik: 2500762.0	Wynik: 2521399.8	Wynik: 2521399.8
Błąd bezwzględny: **20637.75**	Błąd bezwzględny: 0.0	Błąd bezwzględny: 0.0
Błąd względny: **81850.37**	Błąd względny: 0.0	Błąd względny: 0.0

Wnioski

1.1, 1.3 Duży błąd względny standardowego algorytmu wynika z faktu dodawania małej liczby do dużej. Float jako typ pojedynczej precyzji ma dokładność do 24 bitów. Większe liczby reprezentowane są przez dużą cechę, mantysa natomiast dąży do jak najbliższej reprezentacji danego wyniku, ale czym większa różnica pomiędzy liczbami tym większy błąd.
Przy każdej operacji dodawania błąd się sumuje.



1.5 Algorytm rekurencyjny idealnie nadaje się do tablic wartości o podobnym rzędzie wielkości. Sumuje on wartości parami przypominając strukturę drzewiastą. Korzeniem drzewa zostaje suma. Błąd zmalał, gdyż zawsze dodajemy do siebie liczby o tym samym rzędzie wielkości, przez co nie jest tracona precyzja.

1.6 Algorytm kahana i rekurencyjny są około 2-3 razy wolniejsze od standardowego sumowania.

1.7
0.43423915
Wynik: 4342391.0
Błąd bezwzględny: 1820991.2
Błąd względny: 4193521.5

2. Algorytm Kahana

```
Algorytm: Kahan
Czas wykonania: 57ms
Wynik: 2521399.8
2.1 Błąd bezwzględny: 0.0
2.1 Błąd względny: 0.0
```

2.2 zmienna err przechowuje estymację błędu, który pojawi się przy danej operacji. Wartość ta zostanie odjęta od kolejnej dodawanej wartości co spowoduje, że błąd będzie bardzo mały.
2.3 Patrz **1.6**

3. Sumy częściowe

3.1 Dzeta

	50	100	200	500	1000
2	1.19E-07	1.19E-07	2.38E-07	0	3.58E-07
3.666699886	-3.58E-07	-2.38E-07	-1.67E-06	-1.91E-06	-1.91E-06
5	-2.38E-07	-2.38E-07	-2.38E-07	-2.38E-07	-2.38E-07
7.199999809	0	0	0	0	0
10	0	0	0	0	0

Pojedyncza precyzja (wprzód - wstecz) (rozstęp wyniku)

	50	100	200	500	1000
2	2.22E-16	6.66E-16	2.22E-16	2.22E-16	-1.78E-15
3.666699886	-4.44E-16	-6.66E-16	4.44E-16	-2.22E-16	-2.89E-15
5	-1.11E-15	-2.22E-15	-1.55E-15	-4.44E-16	-1.78E-15
7.199999809	2.22E-16	4.44E-16	1.55E-15	2.22E-15	2.22E-15
10	-2.22E-16	-2.22E-16	-2.22E-16	-2.22E-16	-2.22E-16

Podwójna precyzja (wprzód - wstecz) (rozstęp wyniku)

3.2 Dirichlet ETA

	50	100	200	500	1000
2	5.96E-08	0	-1.19E-07	-2.98E-07	-3.58E-07
3.666699886	0	1.19E-07	1.19E-07	1.19E-07	1.19E-07
5	-5.96E-08	-5.96E-08	-5.96E-08	-5.96E-08	-5.96E-08
7.199999809	-5.96E-08	-5.96E-08	-5.96E-08	-5.96E-08	-5.96E-08
10	0	0	0	0	0

Pojedyncza precyzja (wprzód - wstecz) (rozstęp wyniku)

	50	100	200	500	1000
2	-5.55E-16	-4.44E-16	-1.11E-16	8.88E-16	1.33E-15
3.666699886	-2.22E-16	1.11E-16	-6.66E-16	-7.77E-16	-1.22E-15
5	-2.22E-16	-3.33E-16	-6.66E-16	-1.33E-15	-3.33E-16
7.199999809	0	-2.22E-16	1.11E-16	1.11E-16	1.11E-16
10	-2.22E-16	-2.22E-16	-2.22E-16	-2.22E-16	-2.22E-16

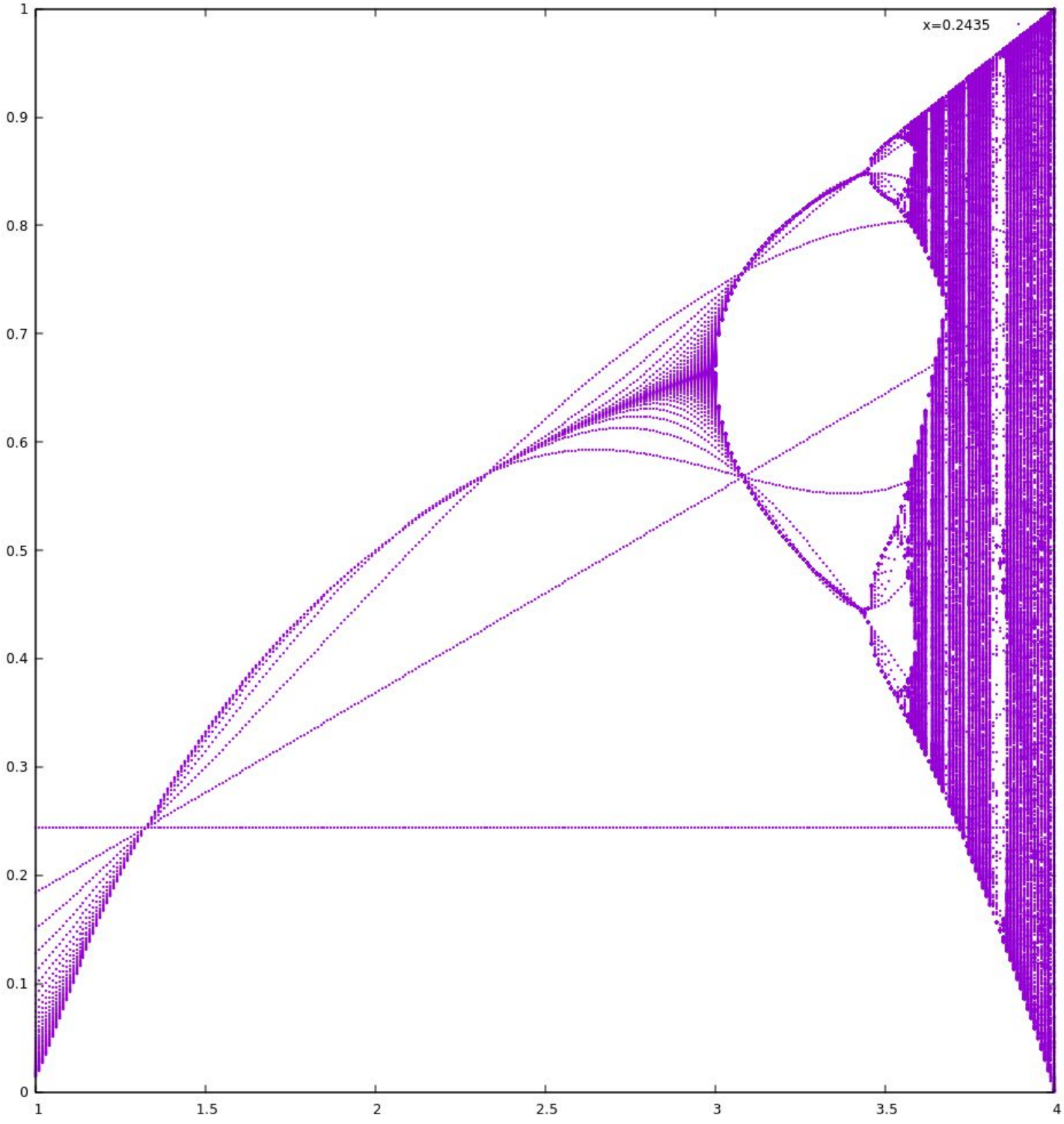
Podwójna precyzja (wprzód - wstecz) (rozstęp wyniku)

Zauważamy, że różnica znacząco zmalała po zastosowaniu liczb podwójnej precyzji.

Różnica pomiędzy liczeniem wstecz i wprzód wynika z niemożliwości dokładnego przedstawienia wyników. Licząc od przodu dodajemy do coraz większej sumy coraz mniejsze liczby. Licząc od tyłu dodajemy coraz większe liczby do coraz większej sumy, co jest zwykle sytuacją korzystniejszą.

4. Błędy zaokrągleń i odwzorowanie logistyczne.

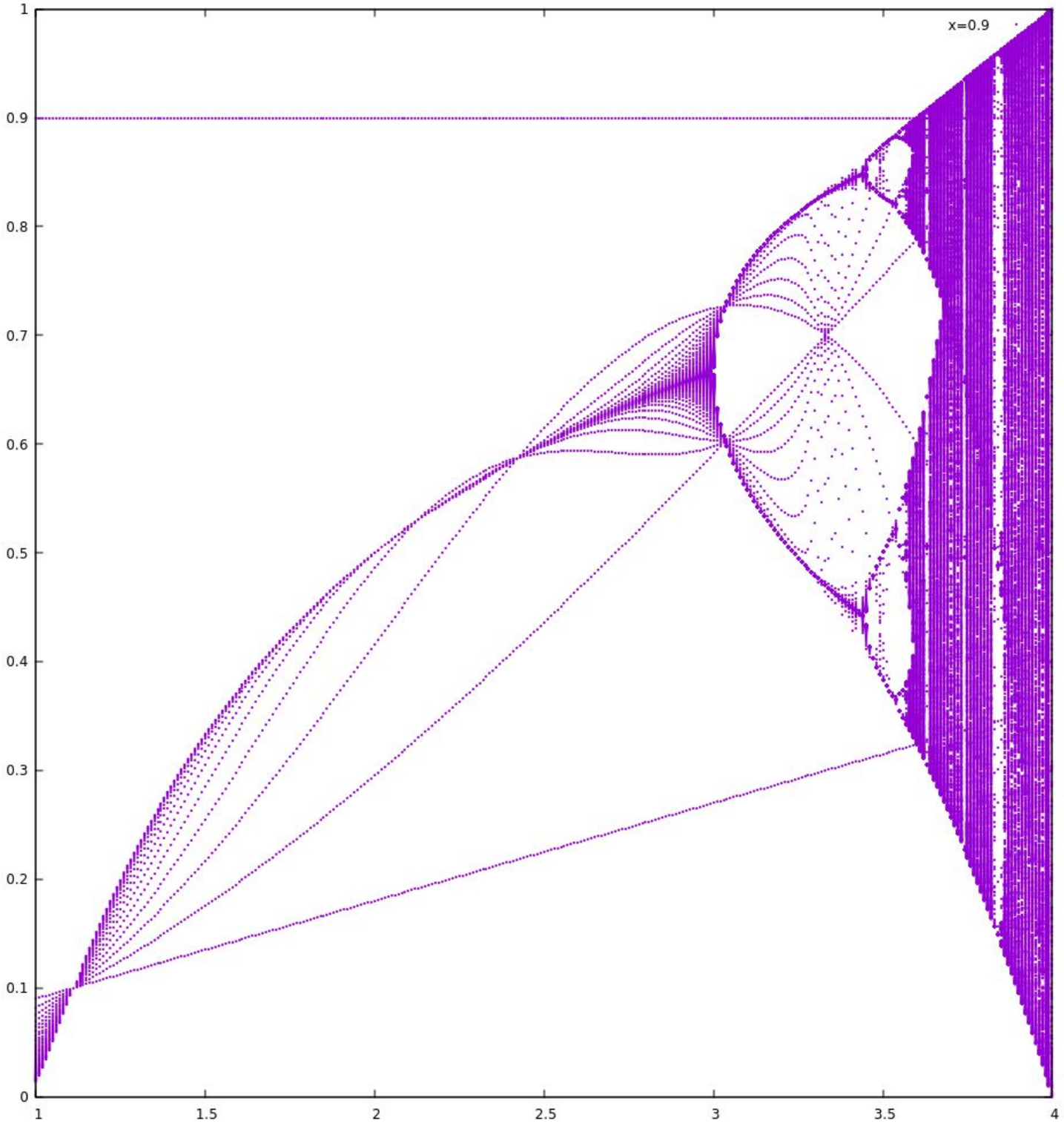
x - oś pionowa,
r - oś pozioma

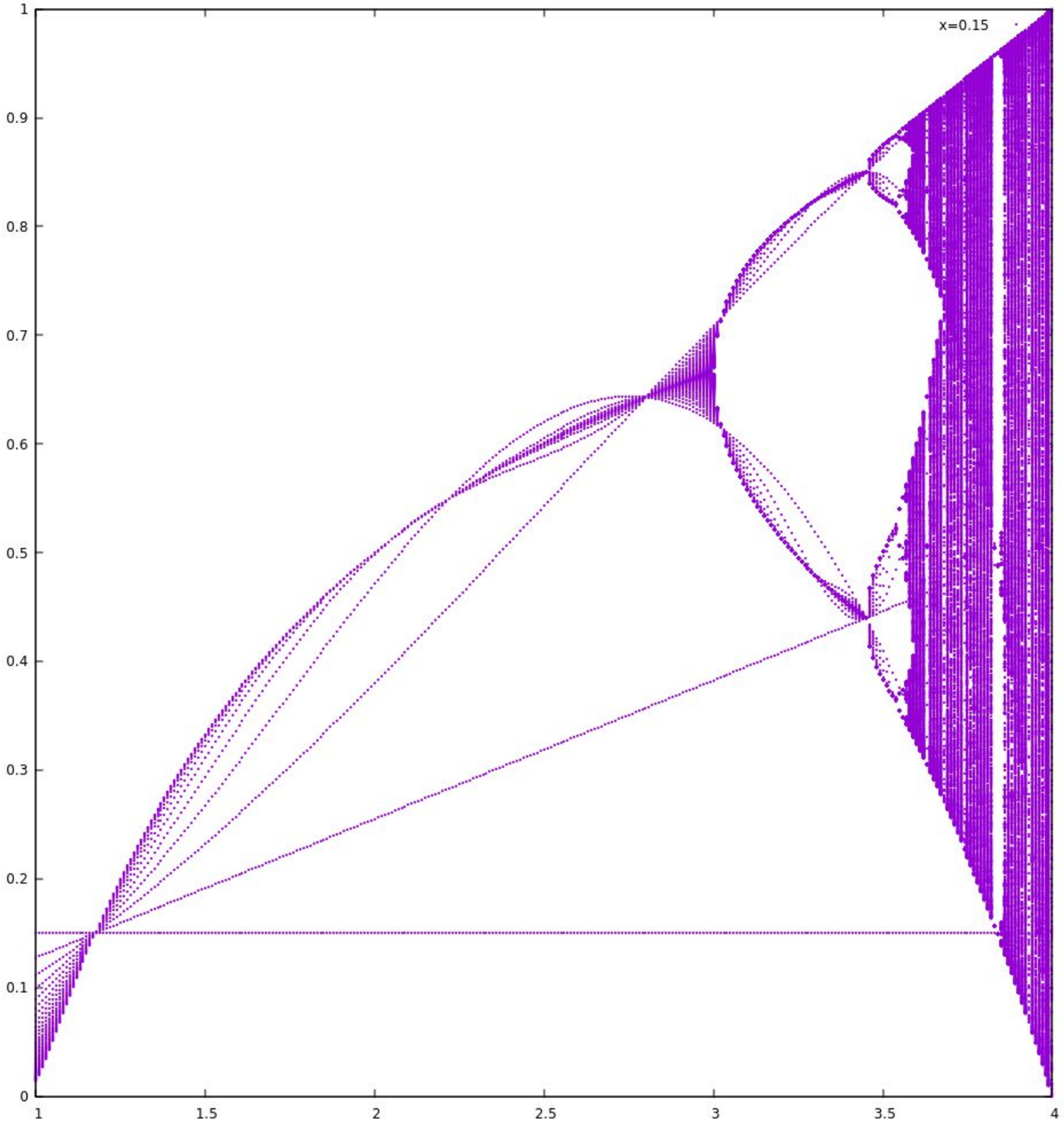


Maciej Mionskowski
Nr albumu: 296628

MOWNIT 2, środa 17:50
06.03.2019

0.2435





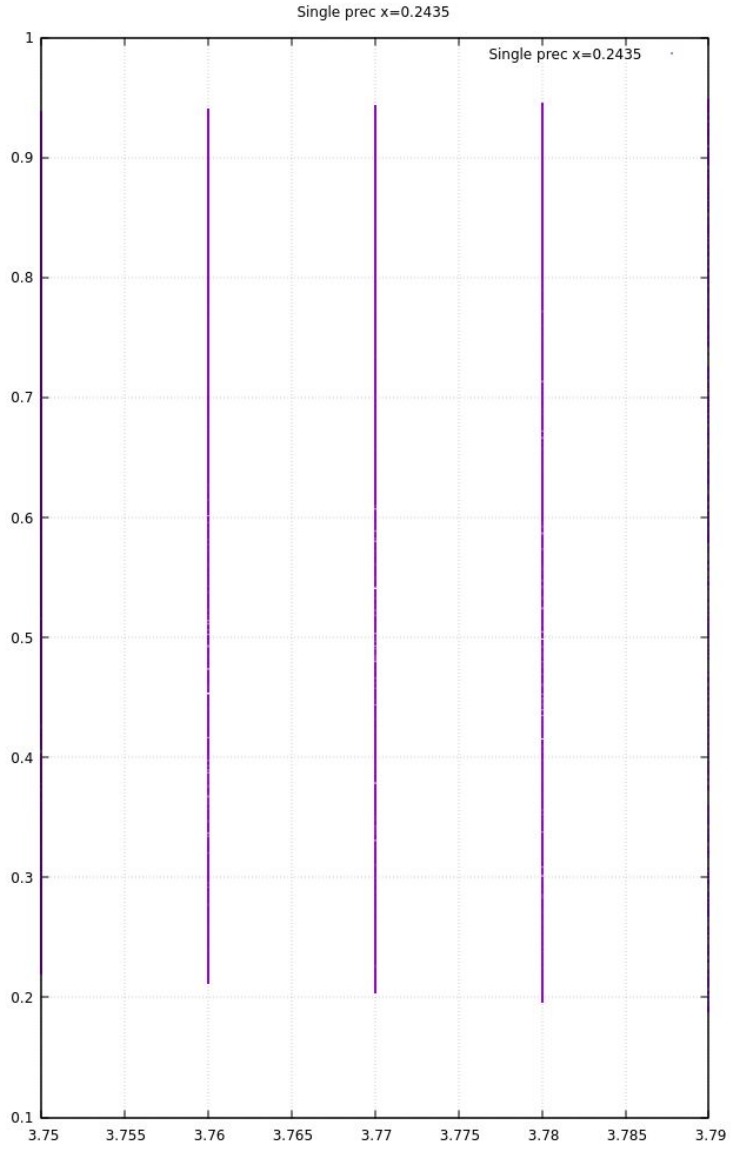
4a. Analiza

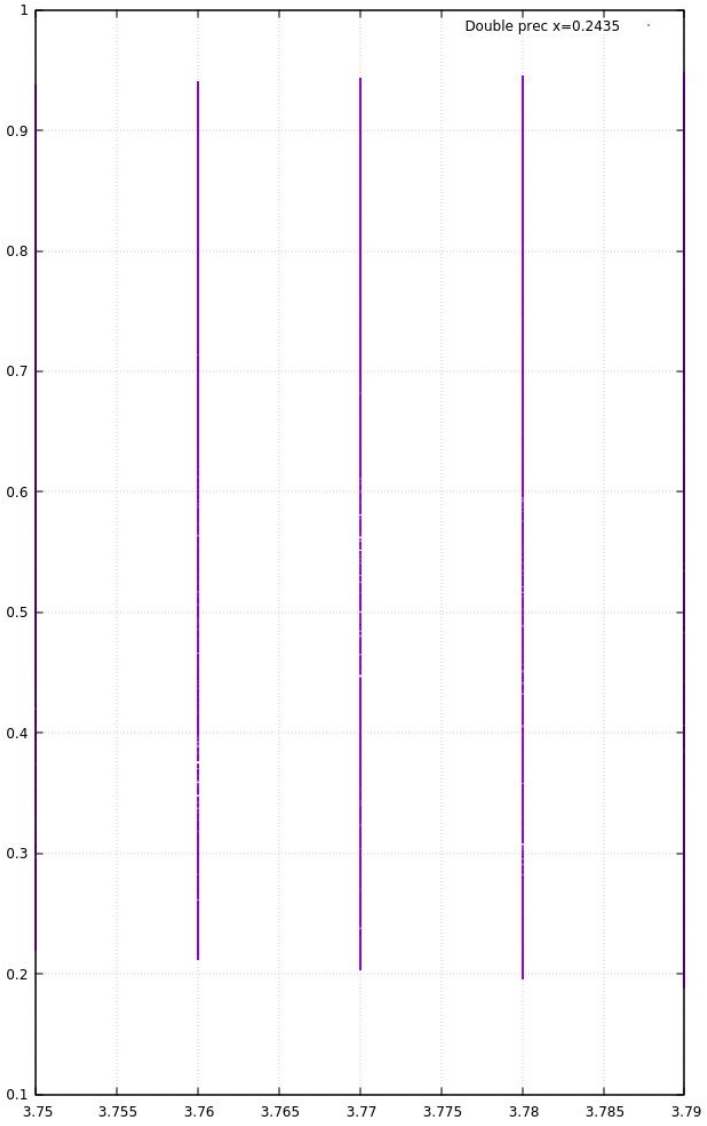
Diagram bifurkacyjny obrazuje czułość algorytmu na delikatna zaburzenia danych. Czym r jest większe tym algorytm jest bardziej podatny na drobne zaburzenia Można zauważyć stabilne trajektorie. Punkty, w który punkty się rozchodzą to forki/rozwidlenia.

Maciej Mionskowski
Nr albumu: 296628

4b. Trajektorie

Nie zauważyłem znaczącej różnicy między pojedynczą, a podwójną precyzją.

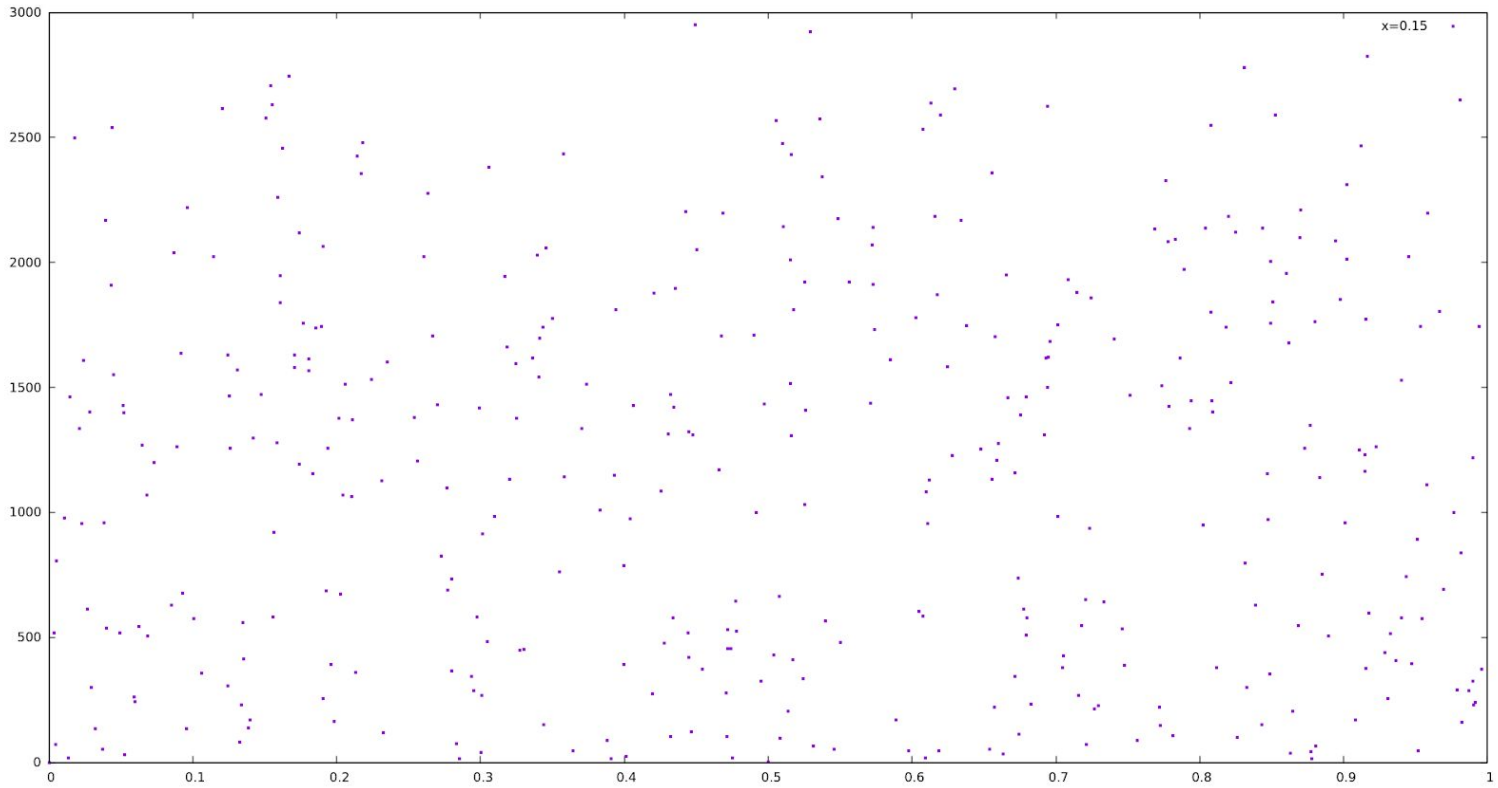




Rezultat jest podobny dla wielu wartości x. Można zauważyć większą regularność przy typie podwójnej precyzji.

Maciej Mionskowski
Nr albumu: 296628

4c Dla r=4. Przyjąłem, że wartość = 0 gdy Math.abs(x) < 0.000000002f.



Nie zauważyłem regularności. Dla niektórych wartości (0.25) ciąg nie zbiega do 0.