28.05.2024

**Sprawozdanie MSD – Lista 3**

Dmytro Zavhorodnii

**1 Wstęp**

*Krótki opis przedstawianego zjawiska*

Wahadło matematyczne jest prostym, ale ważnym modelem używanym do badania ruchu oscylacyjnego. Reprezentuje ono punkt poruszający się po okręgu w płaszczyźnie pionowej pod wpływem pola grawitacyjnego. Jego równanie ruchu opisuje zależność kąta od czasu i pozwala nam przewidzieć jego zachowanie.

Sympy i scipy to biblioteki Pythona przeznaczone odpowiednio do obliczeń symbolicznych i obliczeń naukowych. Za ich pomocą można przeprowadzić analizę wahadła matematycznego, w tym rozwiązać jego równanie ruchu, zbadać zależność okresu oscylacji od długości struny lub amplitudy, a nawet wykreślić wykresy.

Sprawozdanie obejmuje symulację wahadła przy użyciu sympy i scipy oraz analizę różnic tych podejść.

*Niezbędne wzory i warunki początkowe*

Wahadło matematyczne to punkt materialny poruszający się po okręgu w płaszczyźnie pionowej w jednorodnym polu grawitacyjnym. Równanie ruchu wahadła określa wzór:

gdzie:

* Θ(t) - kąt odchylenia wahadła od pionu w chwili t , przy czym kąt ten przyjmują wartości dodatnie np. dla odchyleń w prawo, a ujemne dla odchyleń w lewo
* g - przyspieszenie ziemskie
* l - długość nici.

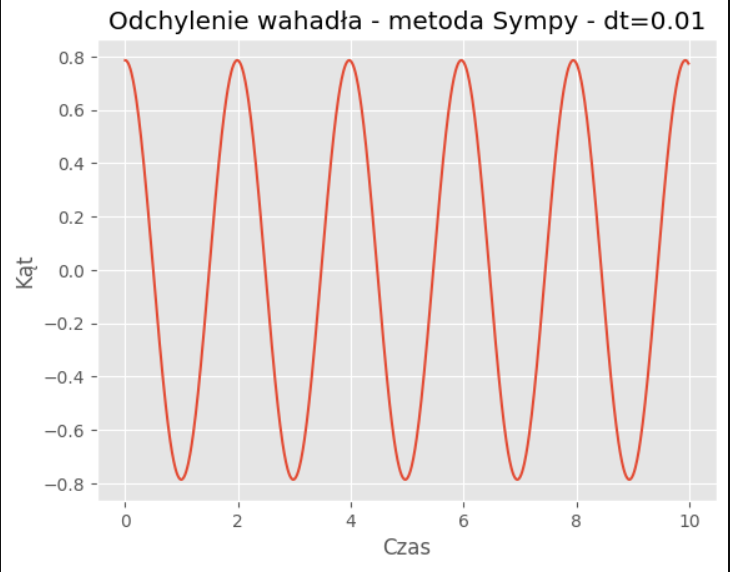
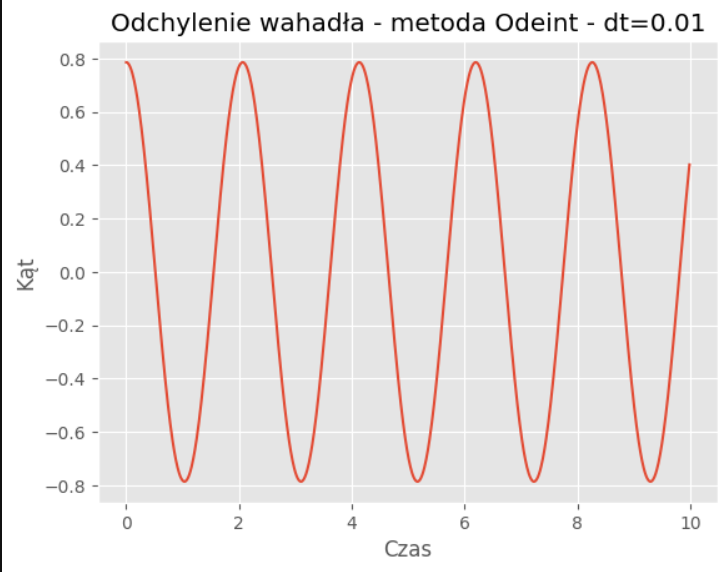
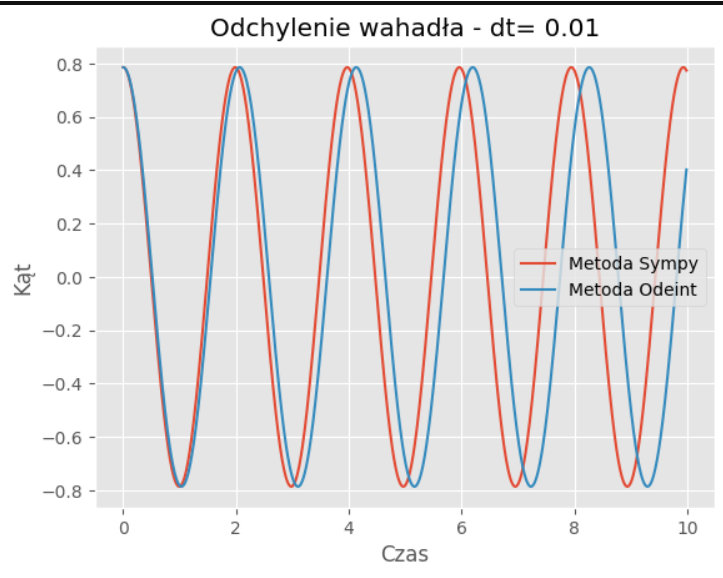
Warunki początkowe przyjęte w symulacji:

* Θ(0) =

**3 Wyniki**

Wynik linearyzacji równania wahadła otrzymany przy pomocy sympy:

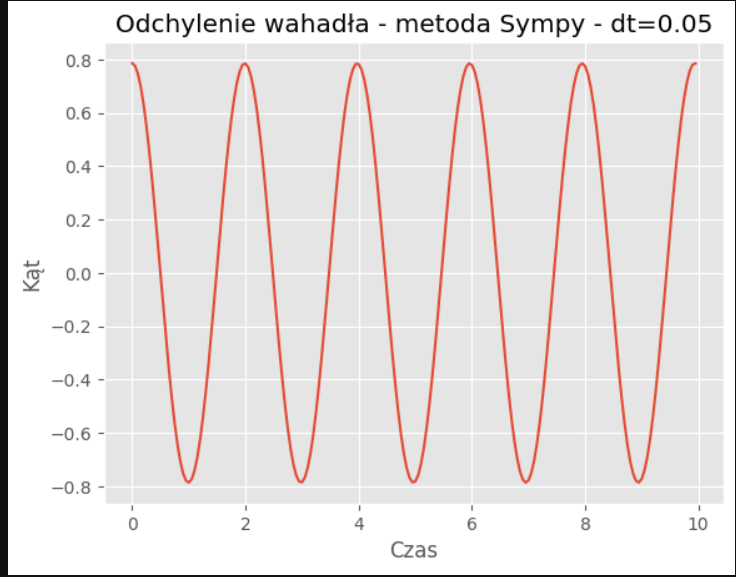
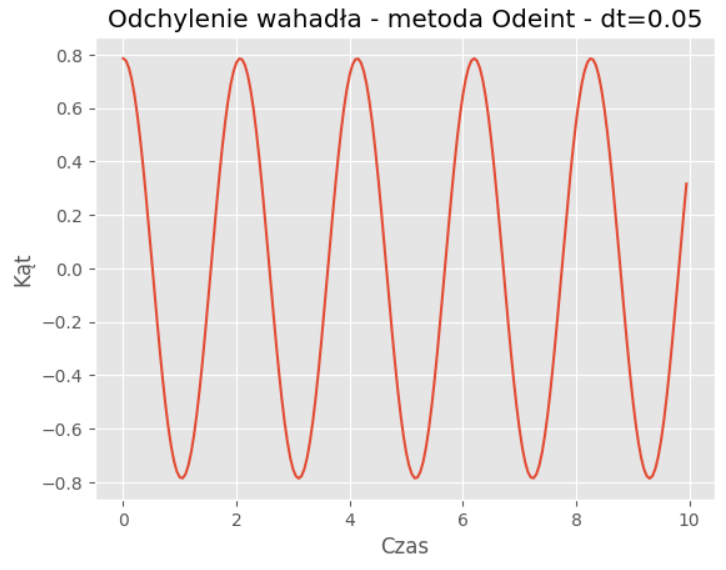
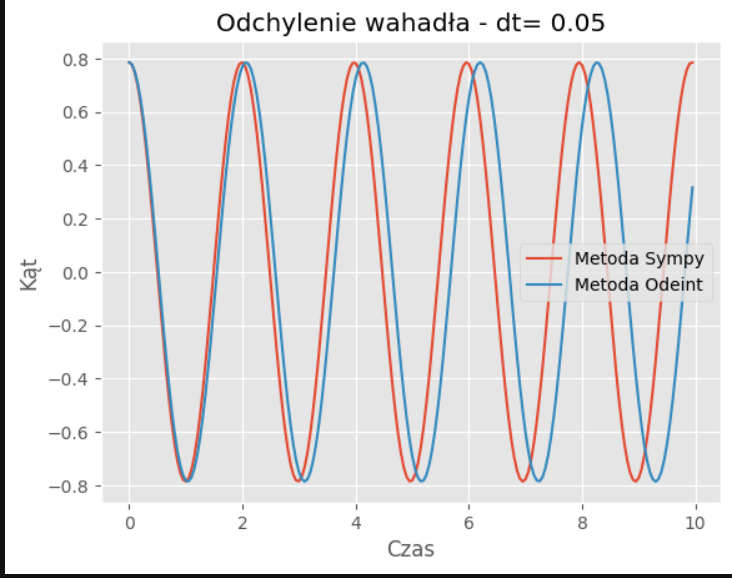
gdzie C1 i C2 dowolne konstanty.

***wykresy:*** 

Rysunek 2

Rysunek 1

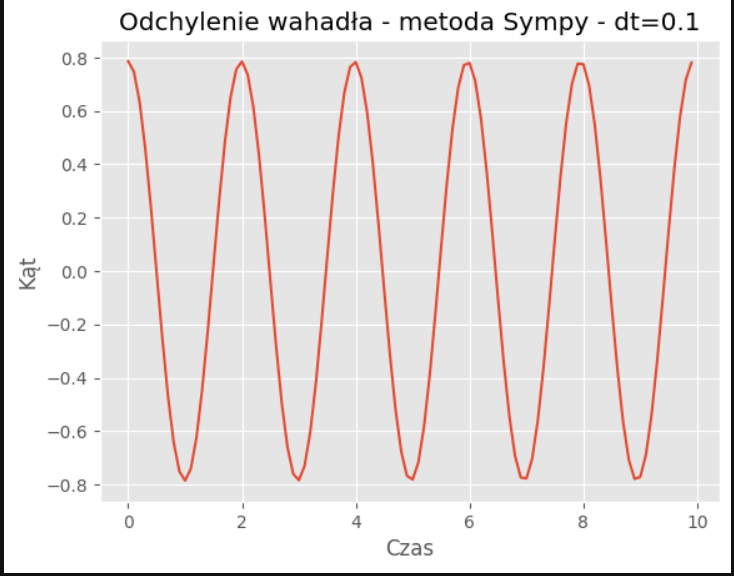
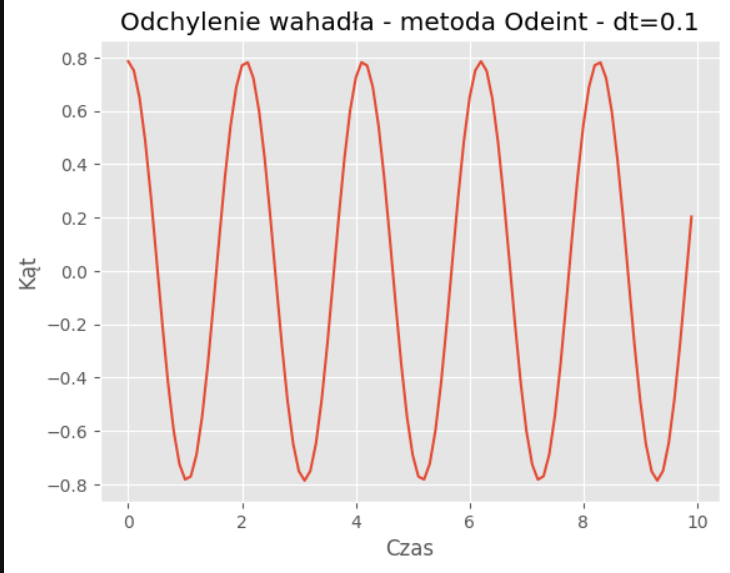
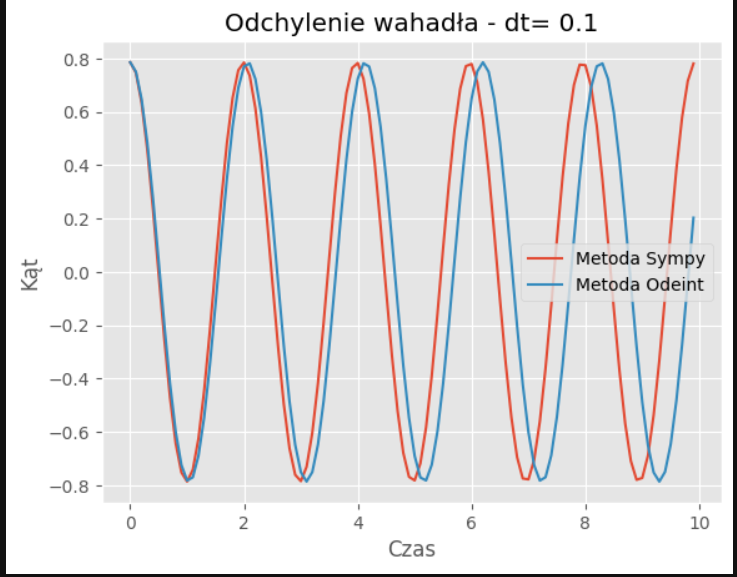
Rysunek 3



Rysunek 6

Rysunek 5

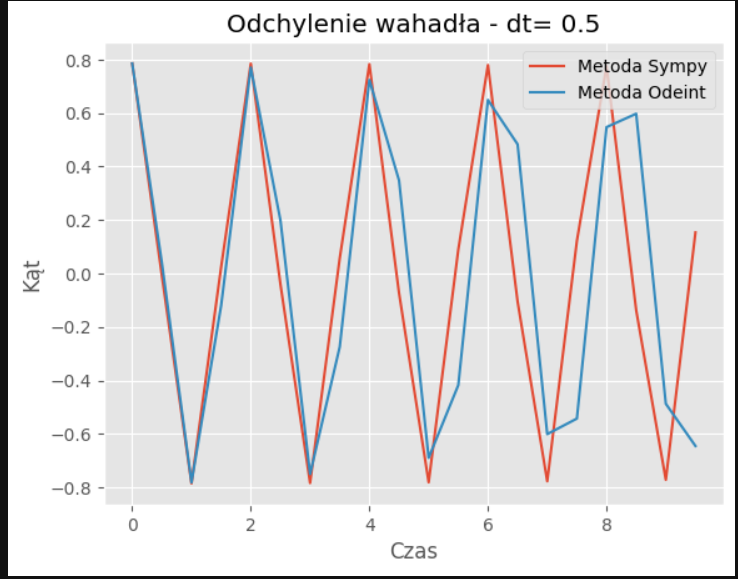
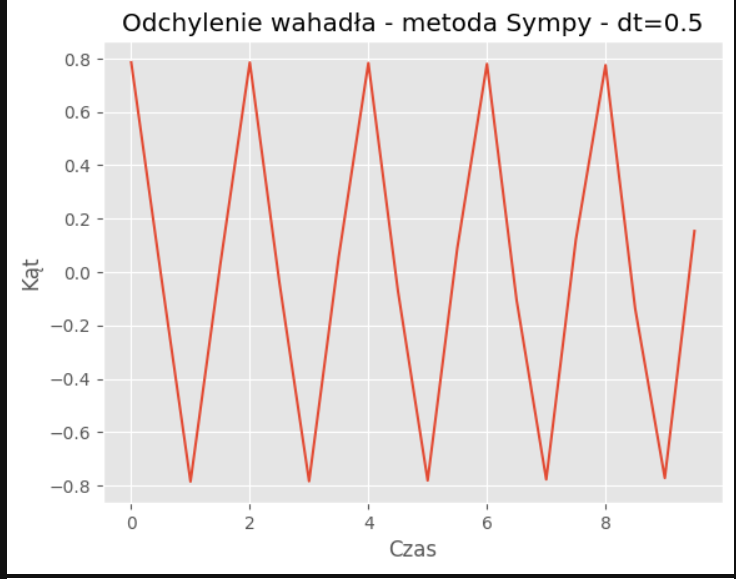
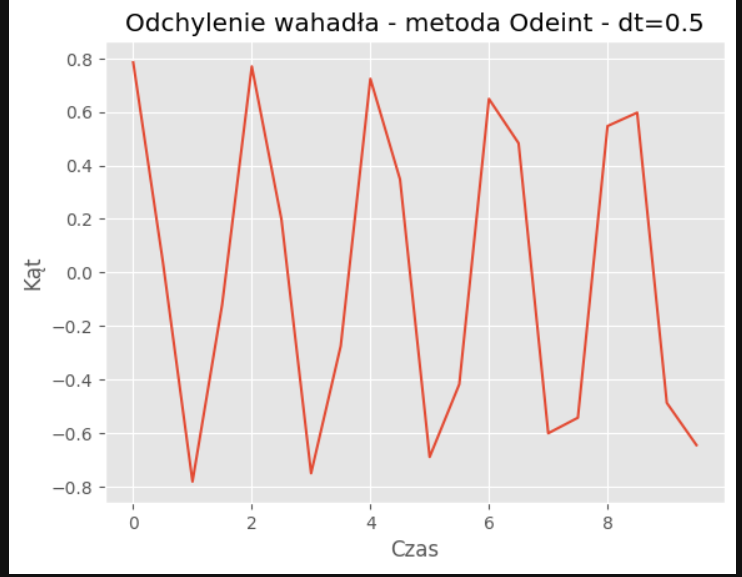
Rysunek 4



Rysunek 9

Rysunek 8

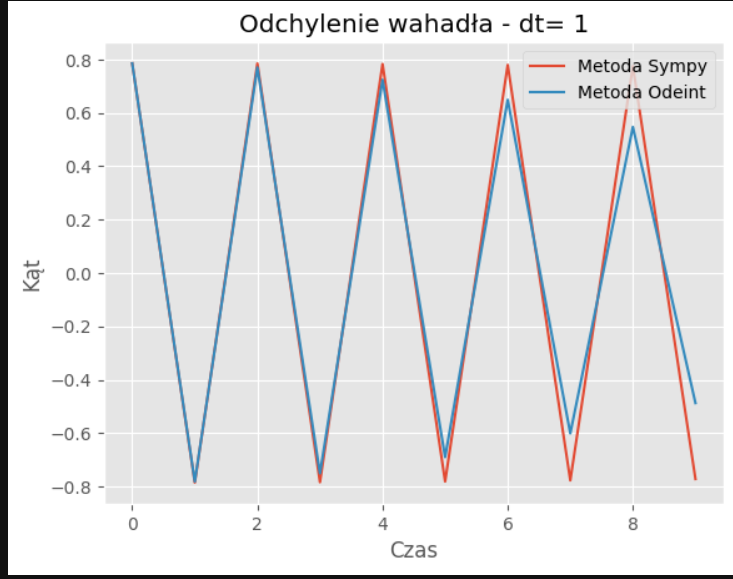
Rysunek 7



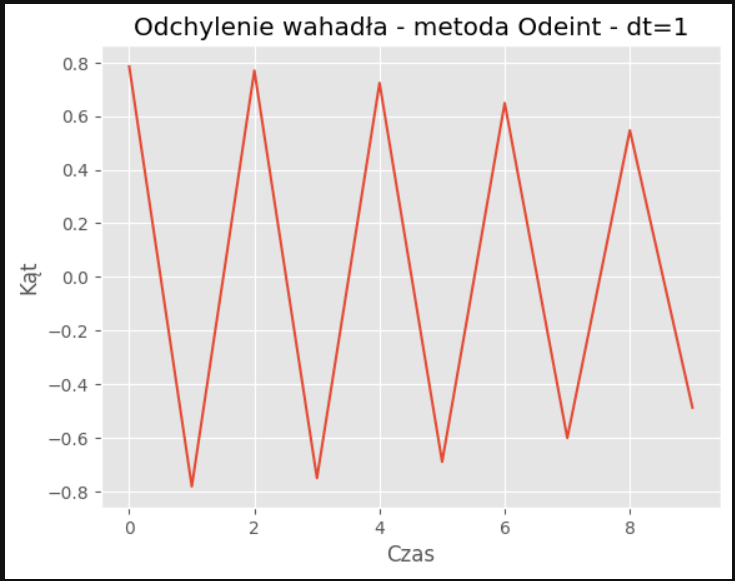
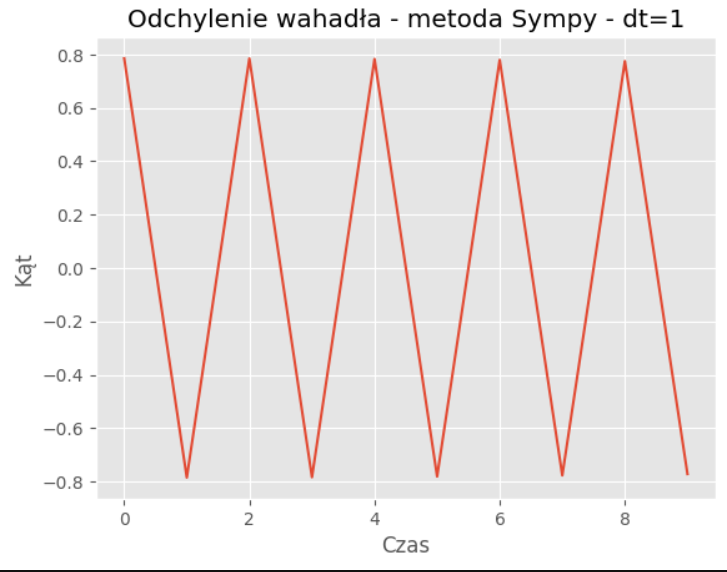
Rysunek 11

Rysunek 10

Rysunek 12

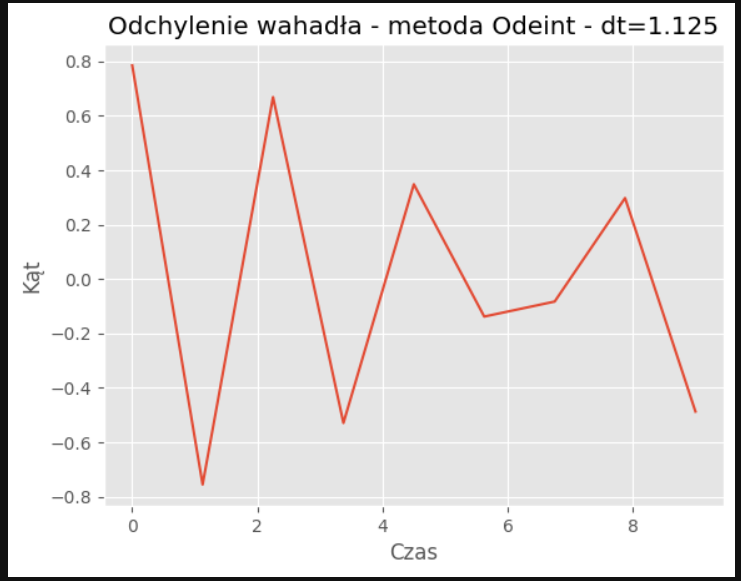
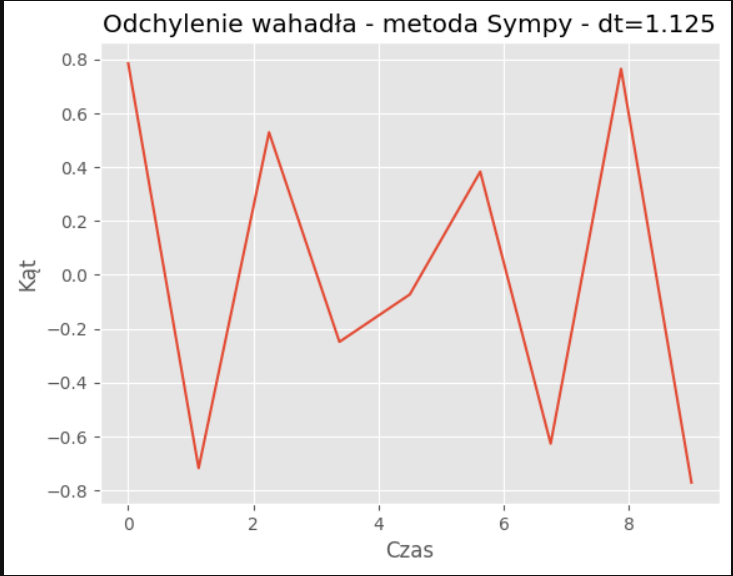
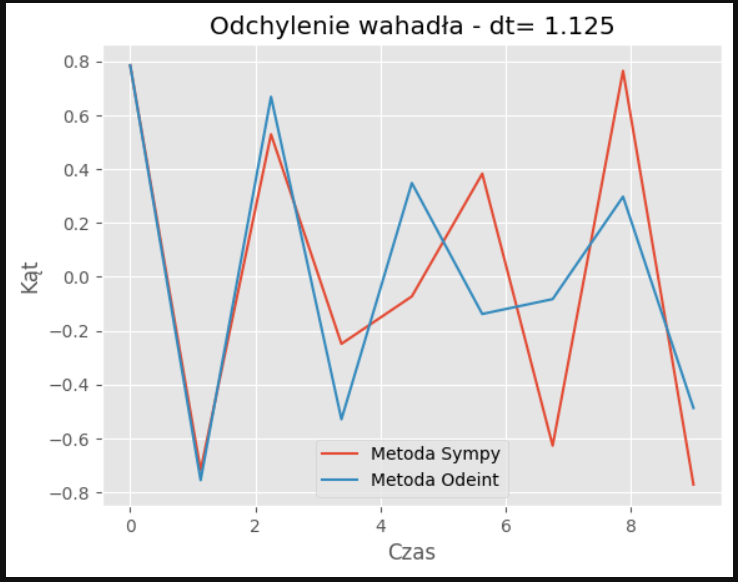


Rysunek 15



Rysunek 14

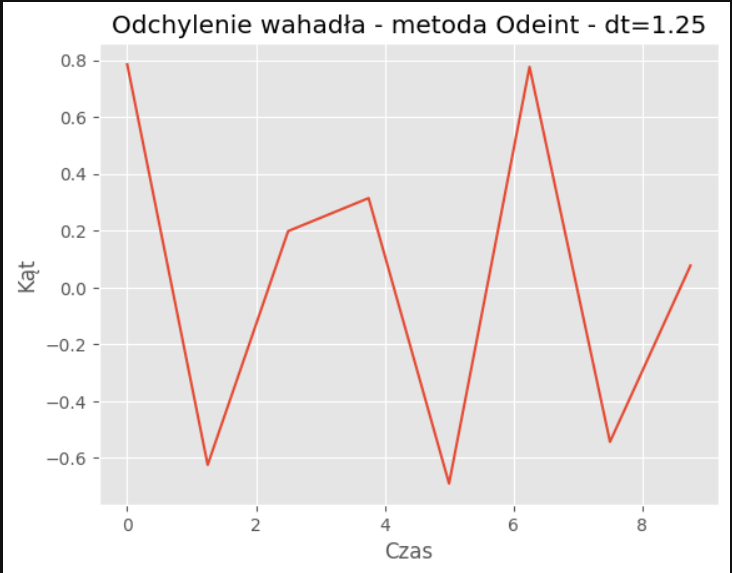
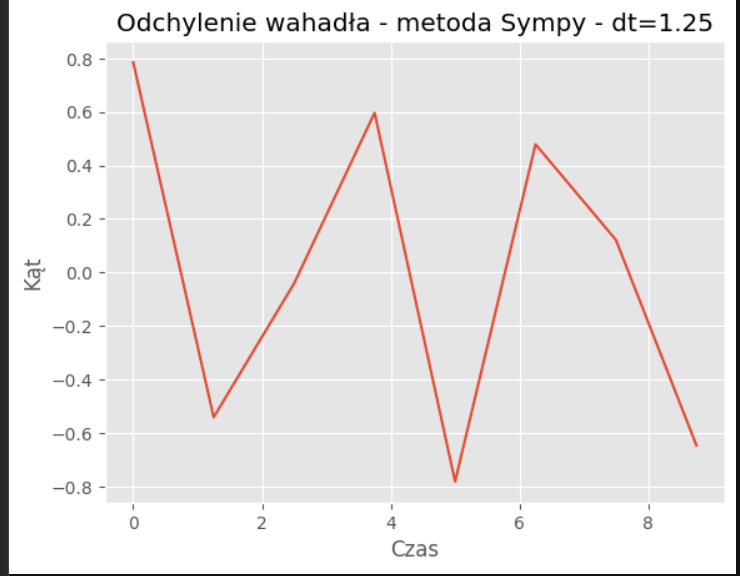
Rysunek 13



Rysunek 18

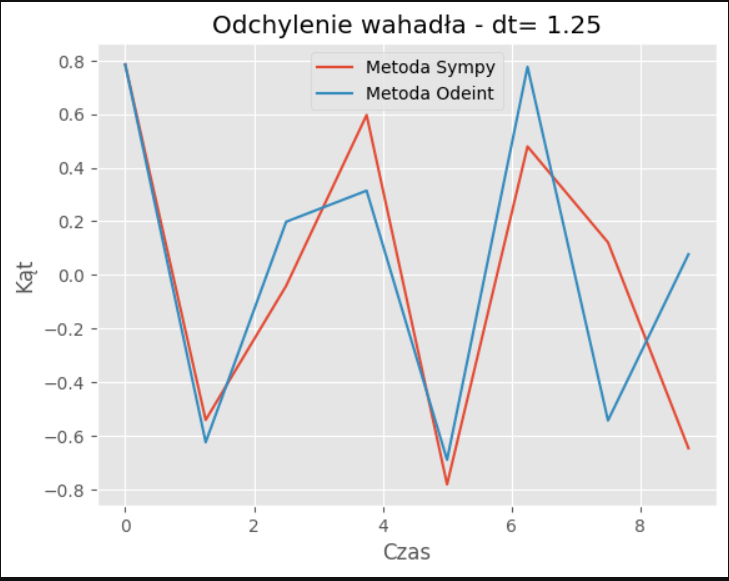
Rysunek 16

Rysunek 17

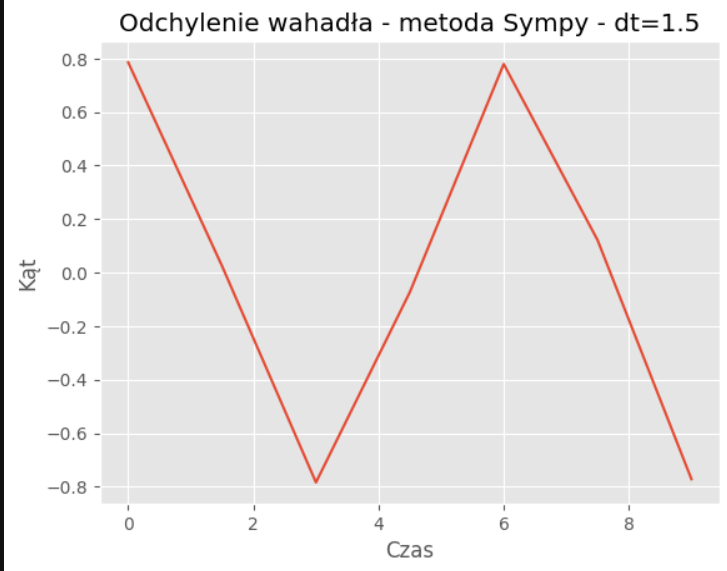
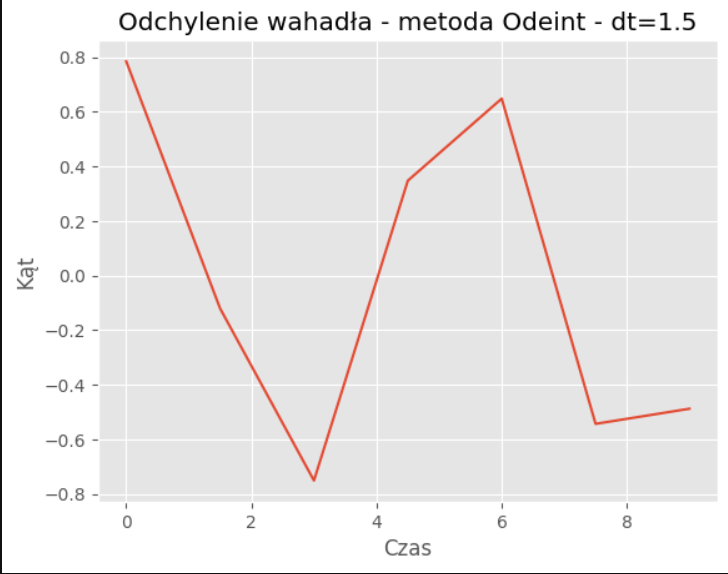


Rysunek 19

Rysunek 20

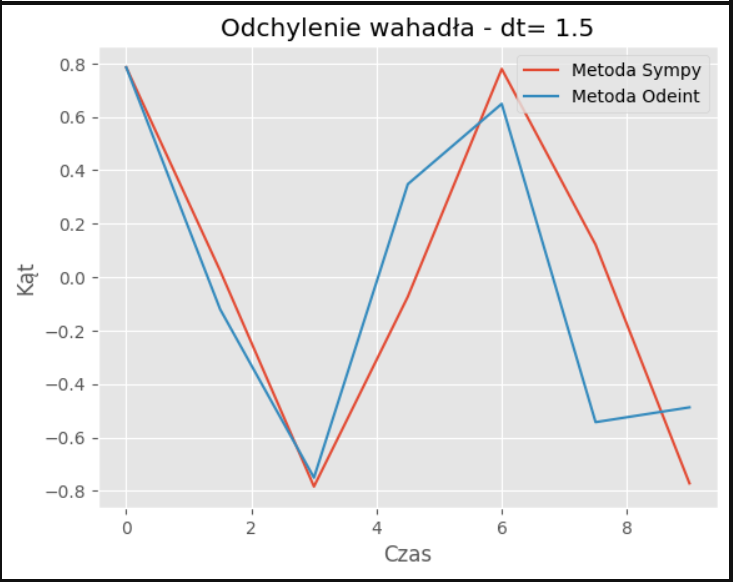


Rysunek 21

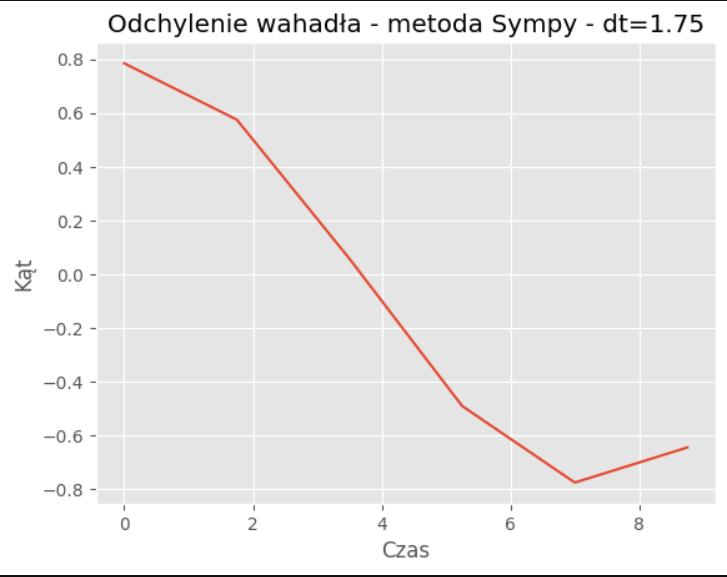
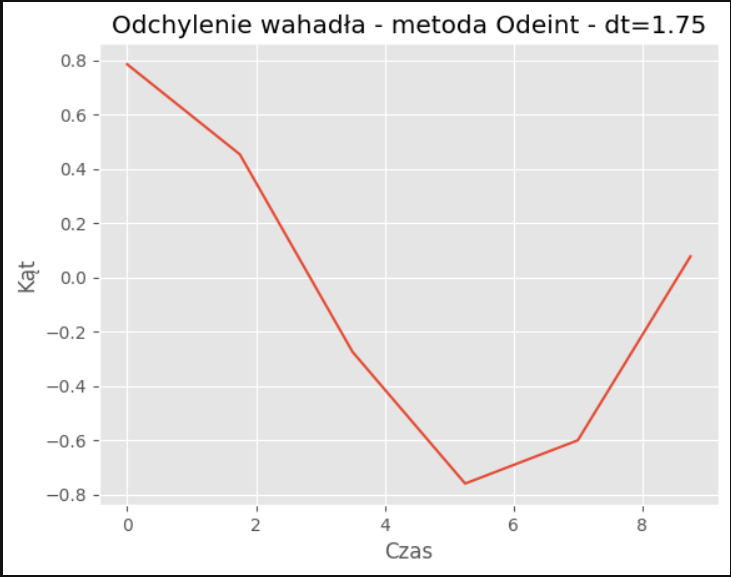


Rysunek 23

Rysunek 22

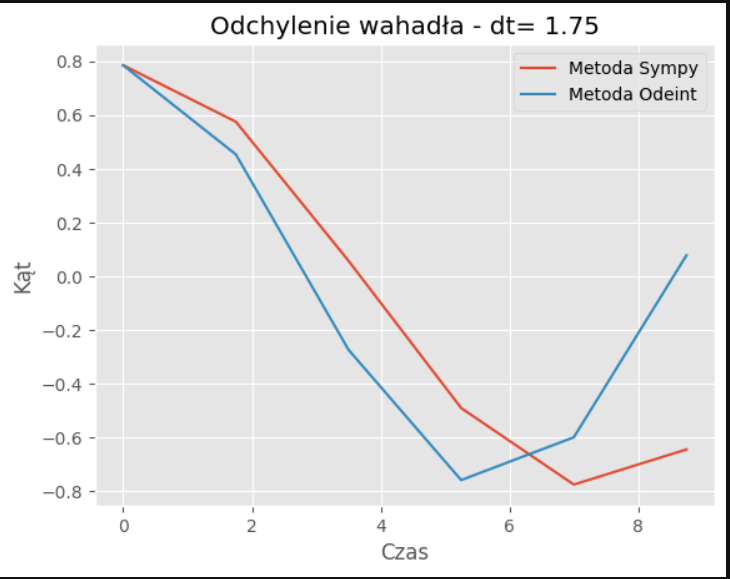


Rysunek 24

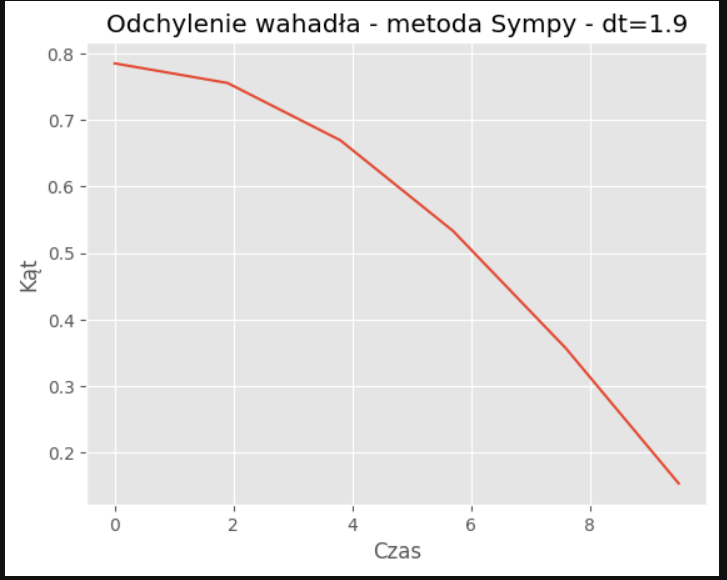
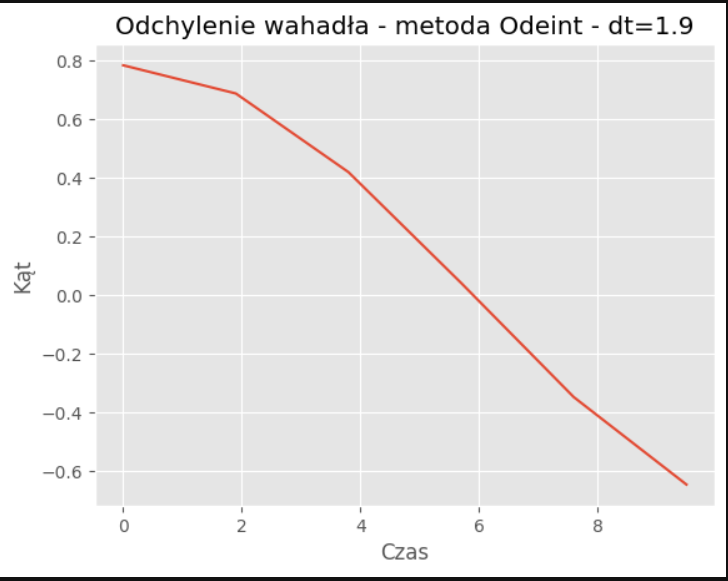


Rysunek 26

Rysunek 25

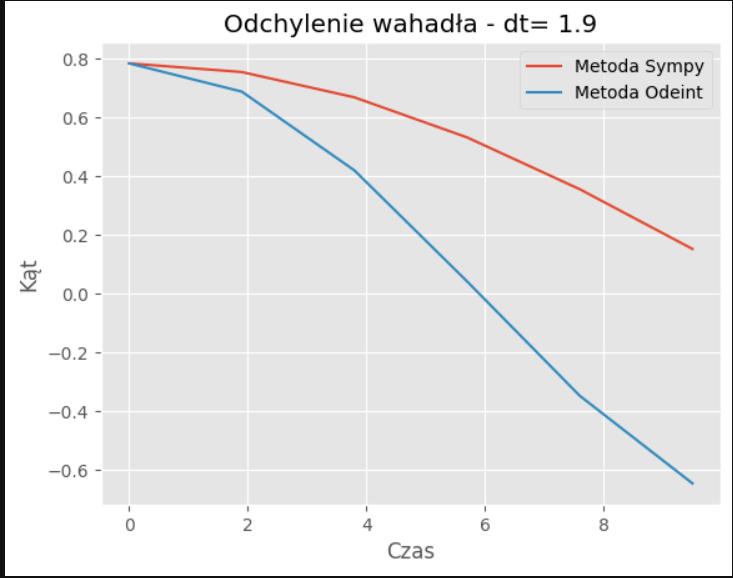


Rysunek 27

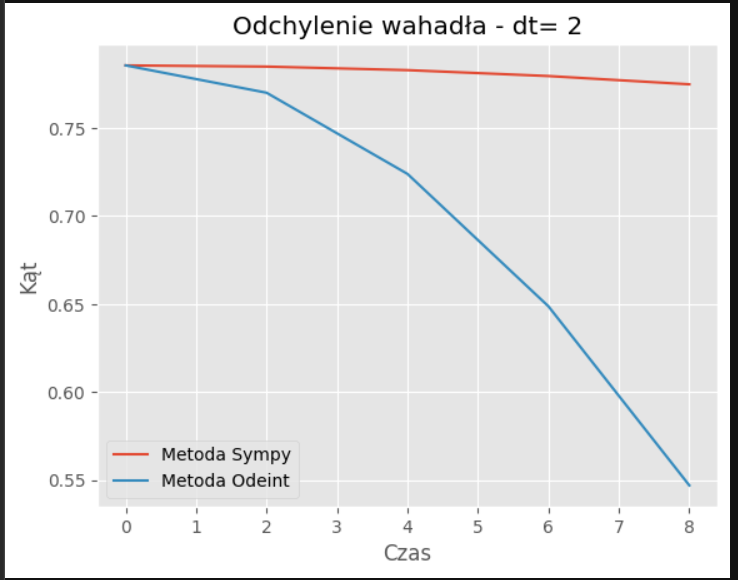


Rysunek 29

Rysunek 28



Rysunek 30



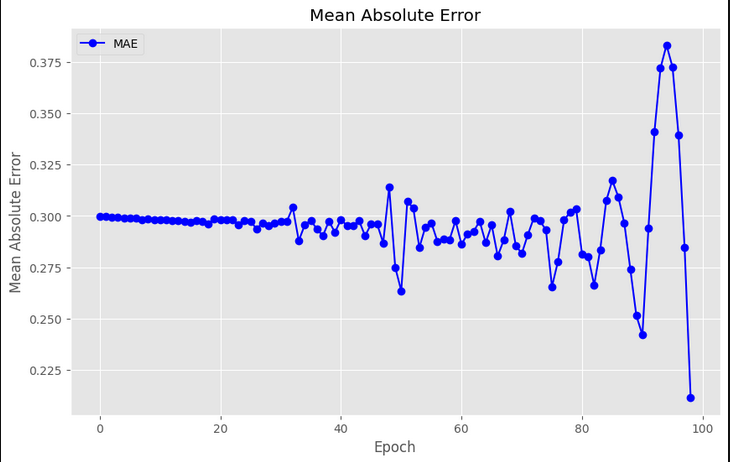
Rysunek 31

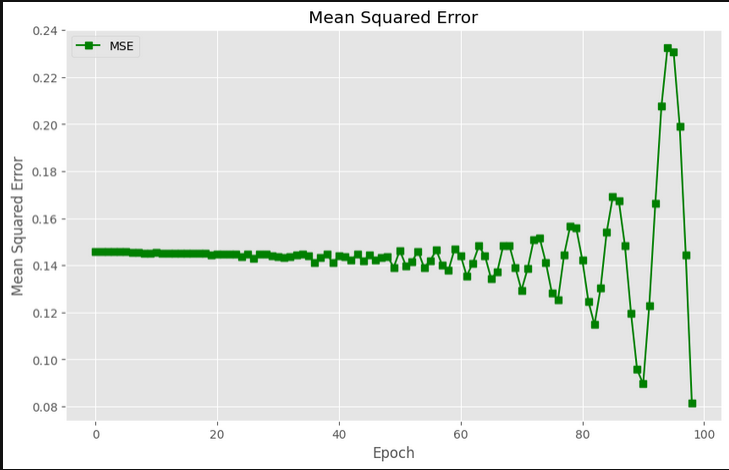
Tabela blędów absolutnego i kwadratowego, wyliczonych w sytuacjach przedstawionych na wykresach:

| **dt** | **MAE** | **MSE** |
| --- | --- | --- |
| 0.01 | 0.0864 | 0.0145 |
| 0.05 | 0.2998 | 0.1460 |
| 0.1 | 0.2990 | 0.1457 |
| 0.5 | 0.2977 | 0.1453 |
| 1 | 0.2749 | 0.1392 |
| 1.125 | 0.1021 | 0.0194 |
| 1.25 | 0.2995 | 0.1271 |
| 1.5 | 0.2976 | 0.1506 |
| 1.75 | 0.2397 | 0.1055 |
| 1.9 | 0.2703 | 0.1252 |
| 2 | 0.3848 | 0.2401 |

Wizualizację obu otrzymanych funkcji oraz osobny wykres błędu dla każdego kroku

𝑡{0, 𝑑𝑡, 2𝑑𝑡, … , 𝑇}



Rysunek 32

Rysunek 33

**4 Wnioski**

**Porównanie metod**: Porównanie wyników symulacji dla metody SymPy i odeint pokazuje, że obie metody generują podobne trajektorie wahadła dla różnych wartości kroku czasowego (dt). Jednakże widać pewne różnice w szczegółach trajektorii, szczególnie dla większych wartości dt.

**Wpływ kroku czasowego**: Analiza błędów (MAE i MSE) dla różnych wartości dt wykazała, że mniejsze wartości dt prowadzą do mniejszych błędów. To potwierdza, że dokładność symulacji wzrasta przy mniejszych krokach czasowych.

**Zbieżność wyników**: Dla odpowiednio małych wartości dt, obie metody zbiegają do podobnych wyników, co jest zgodne z oczekiwaniami. Jednakże dla większych wartości dt różnice między wynikami obu metod mogą być bardziej zauważalne.

**Optymalny krok czasowy**: Analiza błędów dla różnych wartości dt może pomóc w określeniu optymalnego kroku czasowego dla symulacji. Wartość optymalna będzie zależała od wymagań dokładnościowych oraz zasobów obliczeniowych.

**Kontrola błędów**: Śledzenie błędów (MAE i MSE) w zależności od czasu pozwala zrozumieć, jak zmienia się dokładność symulacji wraz z postępem czasu. To może być przydatne przy analizie stabilności i długoterminowej dokładności modelu.

Wnioski te mogą być użyteczne przy dalszym doskonaleniu symulacji oraz przy wyborze odpowiednich parametrów dla konkretnych zastosowań. Dzięki analizie błędów można zoptymalizować wydajność i dokładność modelu, co jest kluczowe w wielu dziedzinach nauki i inżynierii.