

MNUM Projekt - 3

Bartosz Latosek 310790

Grudzień 2022

1 Obliczanie przebiegu trajektorii ruchu punktu

1.1 Wstęp

Zadanie polega na obliczeniu przebiegu trajektorii ruchu punktu w podanym przedziale czasowym, dla zadanego położenia początkowego. Ruch punktu opisany jest za pomocą dwóch równań różniczkowych przedstawionych w plikach *src/f.m*, *src/f2.m*:

$$\frac{dx_1}{dt} = x_2 + x_1 (0.4 - x_1^2 - x_2^2)$$

$$\frac{dx_2}{dt} = -x_1 + x_2 (0.4 - x_1^2 - x_2^2)$$

Rysunek 1: Równania opisujące ruch punktu

Dla podanego zadania, badany jest ruch punktu w przedziale $T = [0, 20]$. Położenie początkowe opisane jest jako $x_1(0) = -8, x_2(0) = -7$. Zadanie zostanie rozwiązane przy pomocy metody Rungego–Kutty-Fehlberga drugiego rzędu przy zmiennym kroku z szacowaniem błędu techniką pary metod włożonych (RK23).

1.2 Algorytm RKF23

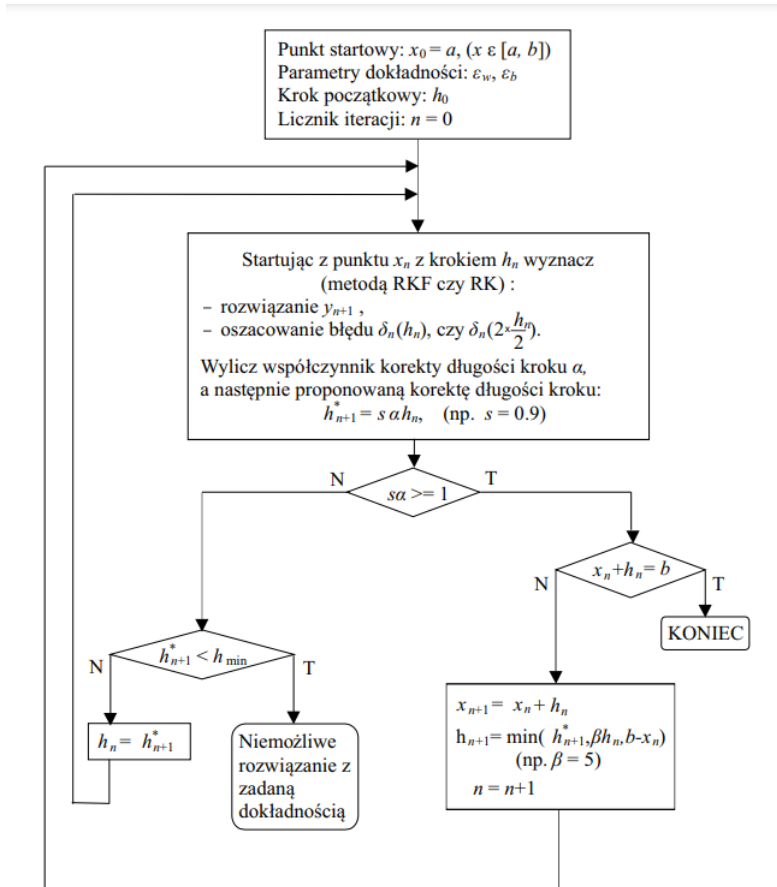
Implementacja przedstawionego poniżej algorytmu znajduje się w pliku *src/RKF23.m*

Algorithm 1 Metoda Rungego–Kutty–Fehlberga drugiego rzędu

```
1: function RKF23( $f, x_0, t_k, h_0, h_{min}, \delta$ )           ▷ Where  $f$  - tested function,  $x_0$  – pointpositionatime(0)
    ▷  $t_k$  - end of time interval,  $h_0$  - initial step,  $h_{min}$  - minimal step,  $\delta$  - minimal acceptable error
     $x = x_0$                                              ▷ Current point position
     $h = h_0$                                              ▷ Current step size
     $t = 0$                                               ▷ Current time
     $i = 1$ 
2: while  $t < t_k$  do
     $k_1 = f(t_n, x_n)$                                 ▷ Wyznaczenie współczynników metodą Rungego-Kutty-Fehlberga
     $k_i = f(t_n + c_i * h + x_n + h * \sum_{j=1}^{i-1} a_{ij} k_j)$ 
     $x_{p,n+1} = x_n + h * \sum_{i=1}^m w_i^* k_i$            ▷ Calculating next position of point using method of p and p+1 order
     $x_{p+1,n+1} = x_n + h * \sum_{i=1}^m w_i k_i$ 

     $err = |x_{p+1,n+1} - x_{p,n+1}|$                    ▷ Calculating error using embedded method of approximation
     $\alpha = (\delta/err)^{1/(p+1)}$                        ▷ Calculating step correction factor
     $h_{opt} = s * h * \alpha$                            ▷ Calculating step correction
    if  $s * \alpha \geq 1$  then
        if  $t + h = t_k$  then
            break
        else
             $x = x_{p+1,n+1}$ 
             $t = t + h$ 
             $i = i + 1$ 
             $h = \min(h_{opt}, \beta * h, t_k - t)$ 
        endif
    else
        if  $h_{opt} < h_{min}$  then
            Its impossible to solve with given  $\delta$ 
        else
             $h = h_{opt}$ 
        endif
    endif
3: end while
4: end function
```

W przedstawionym powyżej algorytmie został wykorzystany przedstawiony w materiałach schemat blokowy:



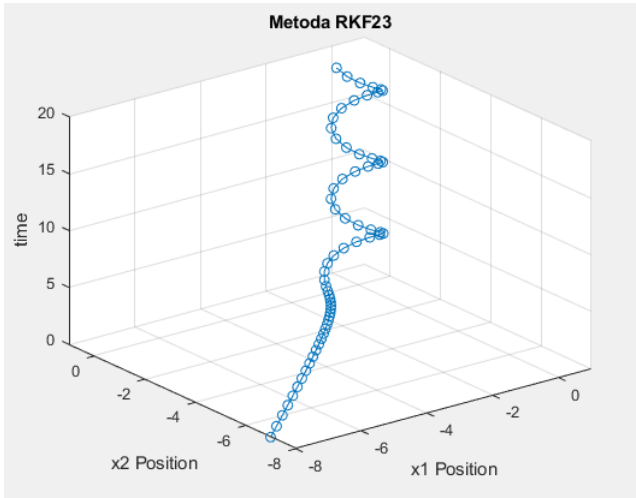
Rysunek 2: Schemat blokowy

1.3 Przykładowe działanie metody i porównanie z metodą ode45

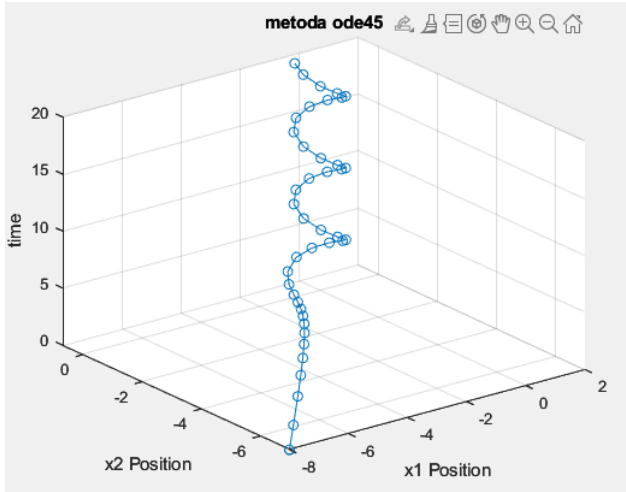
Zaprezentowane zostanie przykładowe działanie metody RKF23 dla zadanych parametrów początkowych:

$$x_0 = [-8 - 7], t_k = 20, h_0 = 0.01, h_{min} = 1e - 4, \delta = 1e - 4$$

1.3.1 Wykresy $t = f(x_1, x_2)$

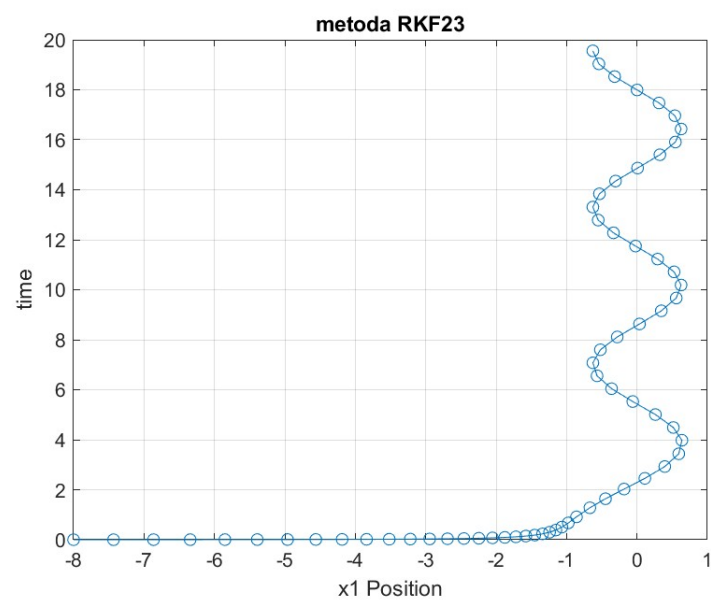


Rysunek 3: Metoda RKF23

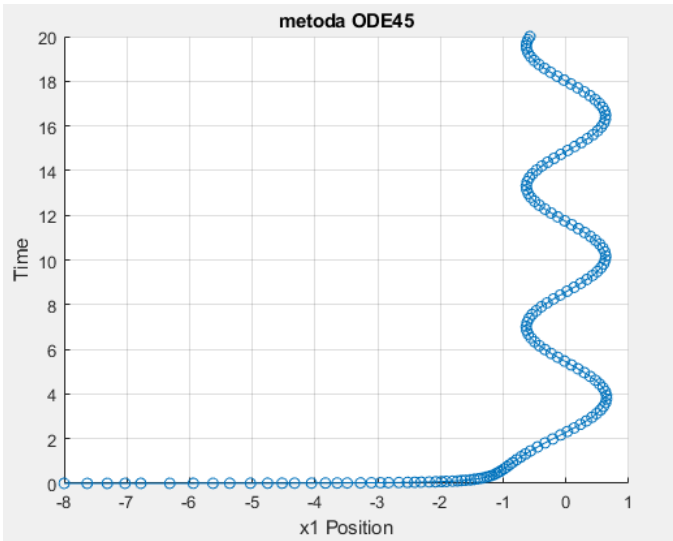


Rysunek 4: Metoda ode45

1.3.2 Wykresy $t = f(x_1)$

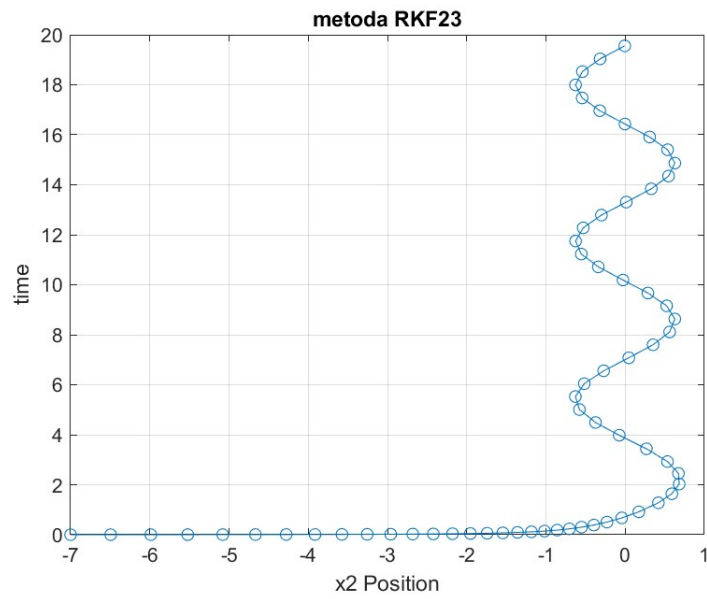


Rysunek 5: Metoda RKF23

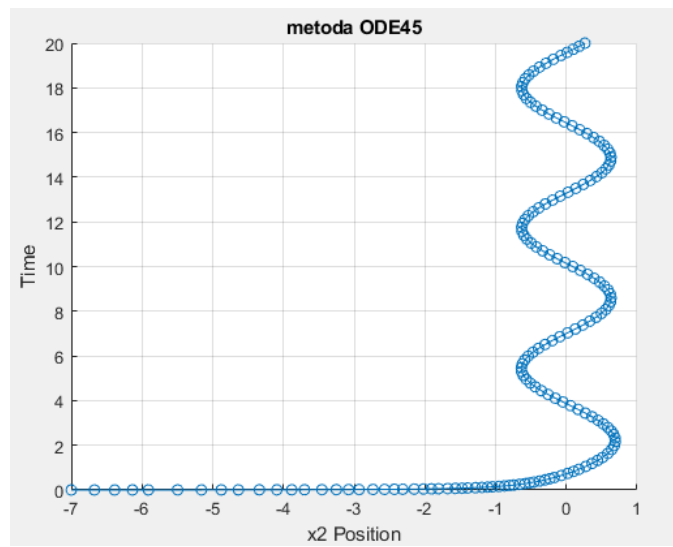


Rysunek 6: Metoda ode45

1.3.3 Wykresy $t = f(x_2)$

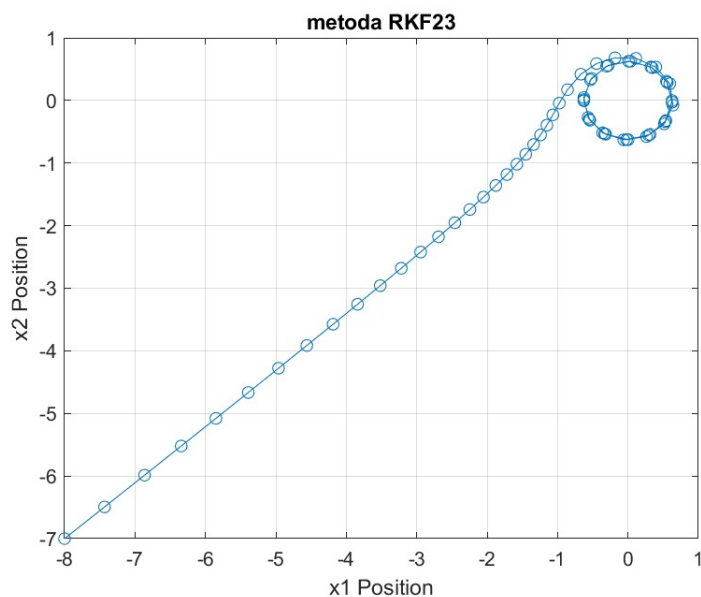


Rysunek 7: Metoda RKF23

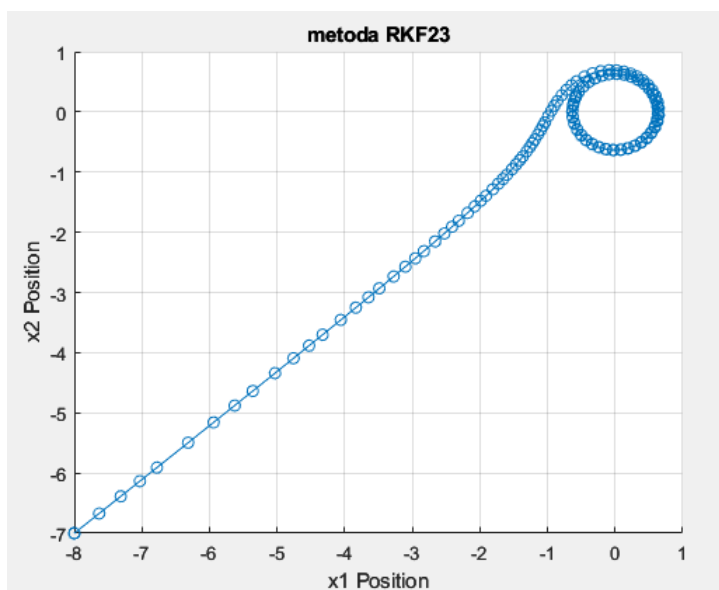


Rysunek 8: Metoda ode45

1.3.4 Wykresy położenia punktu



Rysunek 9: Metoda RKF23



Rysunek 10: Metoda ode45

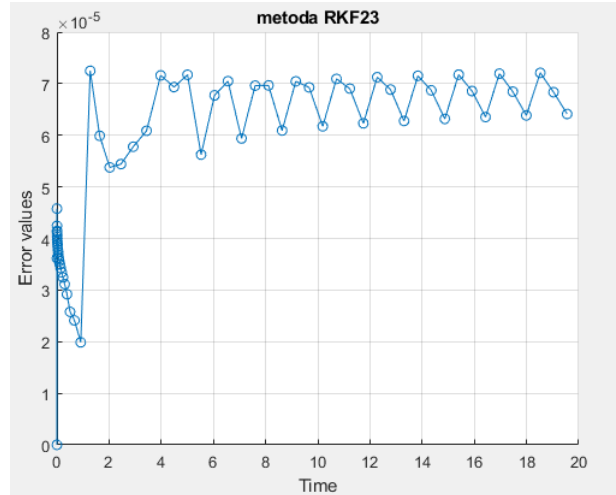
Metoda ode45 jest metodą Runge-Kutta o stałym kroku, która używa do aproksymacji metody rzędu 4 i 5. Cechuje się ona zwykle większą skutecznością i wydajnością niż pokrewne metody, ale wymaga też większych nakładów obliczeniowych.

Metoda RKF23 jest metodą Runge-Kutta-Fehlberga o dynamicznym kroku, używającej do aproksymacji metody rzędu 2 i 3. Krok ustawiany jest w zależności od estymaty błędu obliczonej metodą par metod włożeniowych.

W przypadku metody ode45, dobierane kroki są znacznie mniejsze i dokładniejsze niż w przypadku metody RKF23, w związku z czym czas obliczeń może być nieznacznie większy. Wykresy generowane za pomocą funkcji ode45 wydają się być znacznie bardziej gładkie, ale krok dobierany jest równomiernie. Metoda ode45 przewiduje

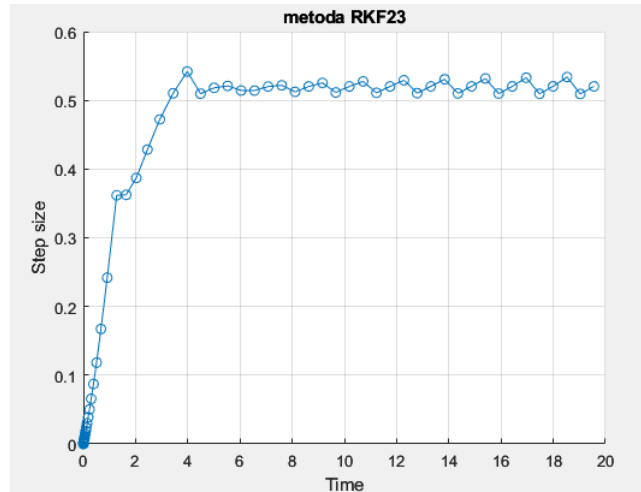
ruch punktów dokładniej, od metody RKF23, chociaż metoda RKF23 jest bardziej elastyczna i przy doborze odpowiednich parametrów wejściowych możemy uzyskać satysfakcjonujące nas wyniki.

1.3.5 Wykresy zależności estymaty błędu i kroku od czasu



Rysunek 11: Wygenerowany wykres zależności estymaty błędu od czasu

Z wykresu widzimy, że estymata błędu stabilizuje się w czasie i przyjmuje wartość zbliżoną do zadanego parametru δ - minimalnego dopuszczanego błędu, a po osiągnięciu tej wartości możemy zaobserwować oscylacje wynikające ze sposobu dobierania długości kroku.

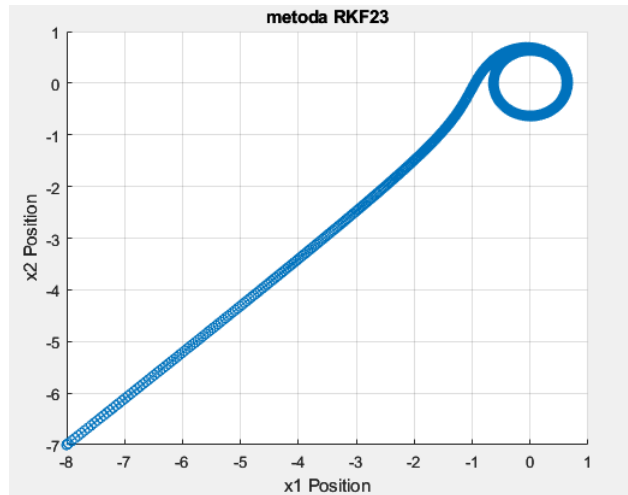


Rysunek 12: Wygenerowany wykres zależności długości kroku od czasu

Na powyższym wykresie można zauważyć, że długość kroku najpierw rośnie, a po wejściu w 'spiralkę' maleje i oscyluje wokół optymalnej wartości, w celu uzyskiwania jak najdokładniejszej estymaty położenia punktu.

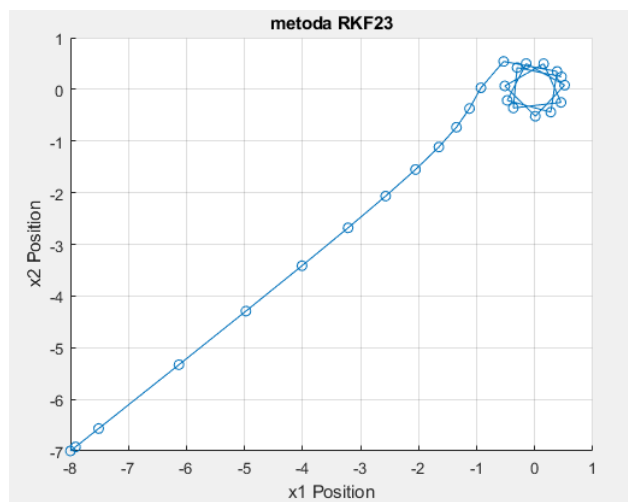
1.3.6 Testy dla różnej wartości minimalnego kroku

Wraz ze zmianą minimalnego kroku, zmianie ulega też kształt otrzymanego wykresu. I tak dla np. $h_{min} = 1e-8$ oraz $\delta = 1e-8$ otrzymujemy znacznie wykres o wyraźnie większym zagęszczeniu punktów.



Rysunek 13: Wygenerowany wykres położenia punktu w 2d

Dla znacznie większej tolerancji błędów oraz $h_{min} = 1e - 3$ wykres przedstawia znacznie rzadziej rozmieszczone punkty (średnia wartość kroku wzrosła).



Rysunek 14: Wygenerowany wykres położenia punktu w 2d