



Tytuł prelekcji: “Numeryczna estymacja i zastosowania pochodnej ułamkowej w inżynierii mechanicznej”



Prelegent: Dr hab. inż. Paweł Olejnik, prof. uczelni

Katedra Automatyki, Biomechaniki i Mechatroniki
Wydział Mechaniczny PŁ



Zawartość prezentacji

Definicja, algorytm obliczania i przykład zastosowania pochodnej ułamkowej w identyfikacji zmiennego współczynnika tłumienia swobodnego oscylatora mechanicznego z tarciem.



Plan prezentacji

- ▶ Czym jest i gdzie ma zastosowanie pochodna ułamkowa?
- ▶ Pierwsze pojawienie się i definicja tego pojęcia w matematyce.
- ▶ Jedno z wcześniejszych zastosowań w obliczeniach na komputerze.
- ▶ Wybrane metody numeryczne i pakiety programistyczne służące do jej obliczania.
- ▶ Lewostronna pochodna ułamkowa w sensie Riemanna-Liouville'a.
- ▶ Procedura numeryczna przybliżająca pochodną ułamkową ciągłej funkcji zdefiniowanej na przedziale obustronnie domkniętym.
- ▶ Przykład obliczenia pochodnej rzędu $\alpha \in \mathbb{R}_{[0,1]}$ funkcji x^2 .
- ▶ Przykład zastosowania pochodnej ułamkowej do wyznaczenia charakterystyki tłumienia oscylatora o jednym stopniu swobody z tarciem w przegubie obrotowym.
- ▶ Podsumowanie.



Czym jest i gdzie ma zastosowanie pochodna ułamkowa?

Pochodna ułamkowa lub inaczej różniczkowanie ułamkowe jest operacją, która pozwala na wyznaczenie pochodnej rzędu niecałkowitego funkcji.

Jest to koncepcja stosunkowo nowa w matematyce, która zyskuje na popularności w różnych dziedzinach nauki, takich jak fizyka, inżynieria czy ekonomia.

Ciekawym zastosowaniem różniczkowania ułamkowego jest na przykład modelowanie dynamiki ciał w środowiskach niejednorodnych, gdzie tradycyjne pochodne nie są w stanie opisać zjawisk występujących w takich układach.

W przypadku modelowania ruchu ciał w wodzie, zastosowanie różniczkowania ułamkowego pozwala uwzględnić wpływ zmian gęstości i lepkości wody na ruch ciała.

W systemach tłumienia drgań w silnikach i maszynach, gdzie różniczkowanie ułamkowe jest stosowane do modelowania tłumienia drgań w układach sprężystych i niejednorodnych.

Innym przykładem zastosowania jest regulacja temperatury w piecach przemysłowych, gdzie różniczkowanie ułamkowe modeluje nieliniową zależność temperatury i prędkości zmian temperatury.



Pierwsze pojawienie się i definicja tego pojęcia w matematyce

Koncepcję pochodnych ułamkowych można prześledzić wstecz do XVII wieku, kiedy matematycy tacy jak Leibniz i L'Hôpital rozważali możliwość rozszerzenia pojęcia różniczkowania na rzędy niecałkowite.

Jednak dopiero w XIX wieku matematycy tacy jak Liouville i Riemann wprowadzili pierwsze ścisłe definicje pochodnych ułamkowych. Od tego czasu rachunek ułamkowy był przedmiotem intensywnych badań i znalazł zastosowanie w wielu dziedzinach nauki i inżynierii.



Jedno z wcześniejszych zastosowań w obliczeniach na komputerze

Pierwsze numeryczne obliczenie pochodnych ułamkowych przypisuje się matematykowi, fizykowi i inżynierowi Oldhamowi, który na początku lat siedemdziesiątych użył algorytmu numerycznego opartego na transformatach Fouriera do obliczenia pochodnych ułamkowych niektórych funkcji. Podejście to zostało później poprawione przez Caputo, który wprowadził bardziej ogólną definicję pochodnych ułamkowych, znaną jako pochodna Caputo.

Podejście Caputo jest szeroko stosowane w obliczeniach numerycznych pochodnych ułamkowych, ponieważ ma kilka zalet w porównaniu z innymi definicjami, takimi jak pochodna Riemanna-Liouville'a.

Oldham, K. B. (1974). The Fractional Calculus. *Mathematical Gazette*, 58(404), 169-177.



Metody numeryczne i pakiety programistyczne służące do jej obliczania

Obecnie dostępnych jest kilka metod numerycznych i wiele pakietów oprogramowania do dokładnego obliczania pochodnych ułamkowych. Wśród nich wymienimy:

Fractional Calculus Toolbox, FractionalOrder – pakiety do obliczania pochodnych ułamkowych w języku MATLAB;

FractionalTools, differint – pakiety do obliczania pochodnych ułamkowych w Pythonie;

ApproxFun.jl, FractionalDiffEq.jl – pakiety Julii do obliczania pochodnych ułamkowych i rozwiązywania równań różniczkowych ułamkowych;

FFC++, OpenFVM – zastosowanie odpowiednio do ww. pary pakietów w C++;

FractionalDerivative – funkcja programu Mathematica.



Lewostronna pochodna ułamkowa w sensie Riemanna-Liouville'a

Aproksymacja lewostronnej pochodnej ułamkowej w sensie Riemanna-Liouville'a:

$${}_a D_t^\alpha x(\cdot) = A(\alpha)(t-a)^{-\alpha}x(t) + B(\alpha)(t-a)^{1-\alpha}\dot{x}(t) - \sum_{p=2}^{\infty} C(\alpha, p)(t-a)^{1-\alpha-p}V_p(x(t)),$$

Moment funkcji i współczynniki zależne od rzędu pochodnej:

$$V_p(x(t)) = (1-p) \int_a^t (\tau-a)^{p-2} x(\tau) d\tau,$$

- Coupled oscillators in identification of nonlinear damping of a real parametric pendulum', *Mechanical Systems and Signal Processing*, 98, 2018, 91-107.

$$A(\alpha) = \frac{1}{\Gamma(1-\alpha)} \left[1 + \sum_{p=2}^{\infty} \frac{\Gamma(p-1+\alpha)}{\Gamma(\alpha)(p-1)!} \right],$$

$$B(\alpha) = \frac{1}{\Gamma(2-\alpha)} \left[1 + \sum_{p=1}^{\infty} \frac{\Gamma(p-1+\alpha)}{\Gamma(\alpha-1)p!} \right],$$

$$C(\alpha, p) = \frac{\Gamma(p-1+\alpha)}{\Gamma(2-\alpha)\Gamma(\alpha-1)(p-1)!}.$$



Procedura numeryczna

```
def fractional_x_dot(n, alpha, N, t, i_a, h, x, x_dot):  
    N += 1  
    x_fac_dot = numpy.zeros(n, float)  
    v_p = numpy.zeros((N - 2, n), float)  
    g3 = numpy.zeros(N - 2, float)  
    g1 = coeff_A(alpha, N).evalf() # Funkcje obliczające  
    g2 = coeff_B(alpha, N).evalf() # współczynniki A, B, C  
  
    for P in range(0, N - 2):  
        v_p[P] = moment(t, x, P + 2, h, n, i_a=i_a)  
        g3[P] = coeff_C(alpha, P + 2)
```




Procedura numeryczna

```
...
for i in range(i_a, n):
    sum_1 = 0
    dt = t[i] - t[i_a]
    for P in range(2, N):
        g4 = dt ** (1 - P - alpha)
        g5 = numpy.trapz(v_p[P - 2][0:i], dx=h)
        sum_1 += (g3[P - 2] * g4 * g5)
        # estymacja poszukiwanej pochodnej:
        x_fac_dot[i] = g1 * dt ** (-alpha) * x[i] + g2 * dt ** (1
            - alpha) * x_dot[i] - sum_1

return x_fac_dot
```



Procedura numeryczna

```
def moment(t, x, P, h, n, i_a=0): # z def. V_p(x) :
    return [(1 - P) * (t[i] - t[i_a]) ** (P - 2) * x[i] for i in
            numpy.arange(i_a, n)]

def coeff_A(alpha, N): # wsp. A:
    g1 = sympy.gamma(1 - alpha) # funkcja gamma
    g2 = sympy.gamma(p - 1 + alpha)
    g3 = sympy.gamma(alpha) * sympy.factorial(p - 1) # silnia
    sum_A = sympy.Sum(g2 / g3, (p, 2, N)) # suma
    return 1 / g1 * (1 + sum_A)
```

Współczynnik B i C obliczamy w bardzo podobny sposób.



Przykład obliczenia pochodnej rzędu $\alpha \in \mathbb{R}_{[0,1]}$ funkcji x^2 – założenia

We wzorze na aproksymację:

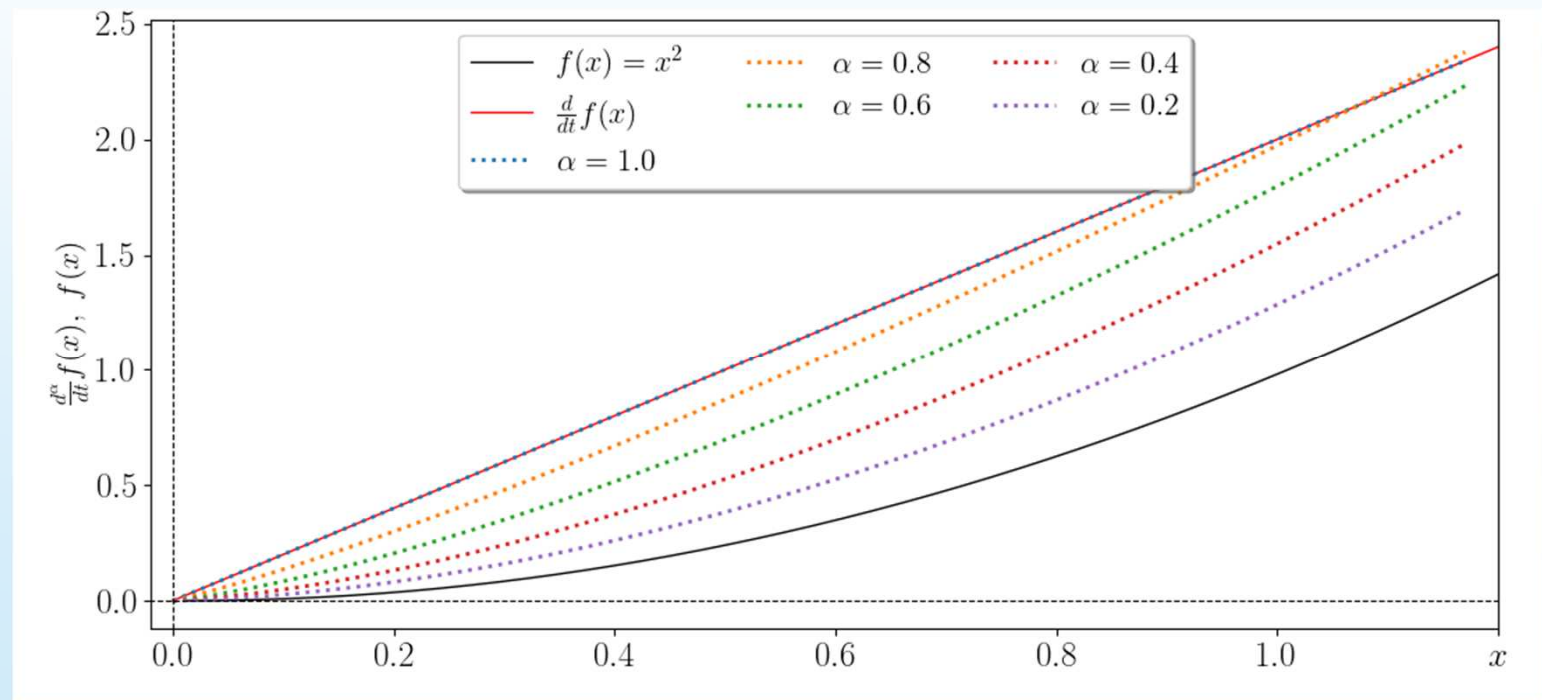
$${}_a D_t^\alpha x(\cdot) = A(\alpha)(t-a)^{-\alpha} x(t) + B(\alpha)(t-a)^{1-\alpha} \dot{x}(t) - \sum_{p=2}^{\infty} C(\alpha, p)(t-a)^{1-\alpha-p} V_p(x(t)),$$

kładziemy $x(t) = x^2$, przyjmujemy $N_p = 1000$ i obliczamy pochodne przy $\alpha_i \in \{1; 0,8; 0,6; 0,4; 0,2\}$.

Uruchomienie 1. programu i prezentacja wyniku

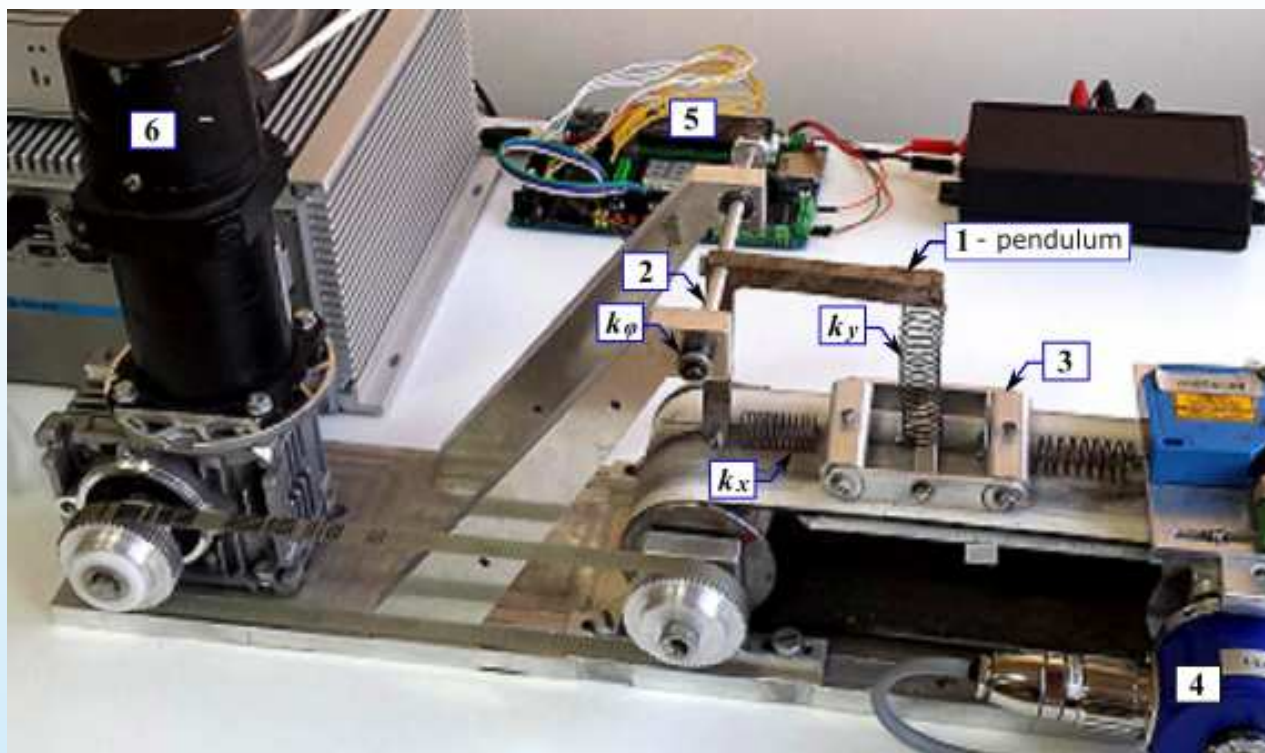


Przykład obliczenia pochodnej rzędu $\alpha \in \mathbb{R}_{[0,1]}$ funkcji x^2 – symulacja i wyniki





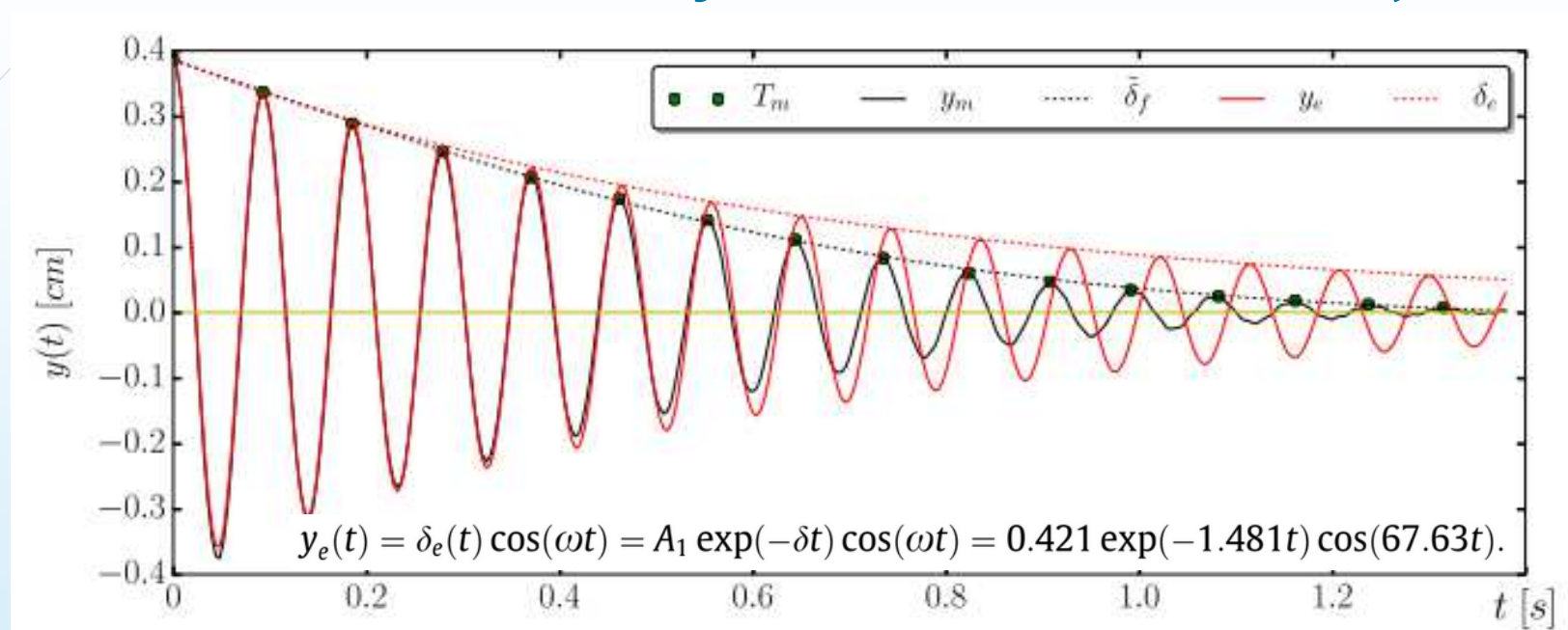
Zastosowanie w inżynierii mechanicznej



Przykład zastosowania pochodnej ułamkowej do wyznaczenia charakterystyki tłumienia oscylatora o jednym stopniu swobody z tarciem w przegubie obrotowym (1).



Zastosowanie w inżynierii mechanicznej cd..

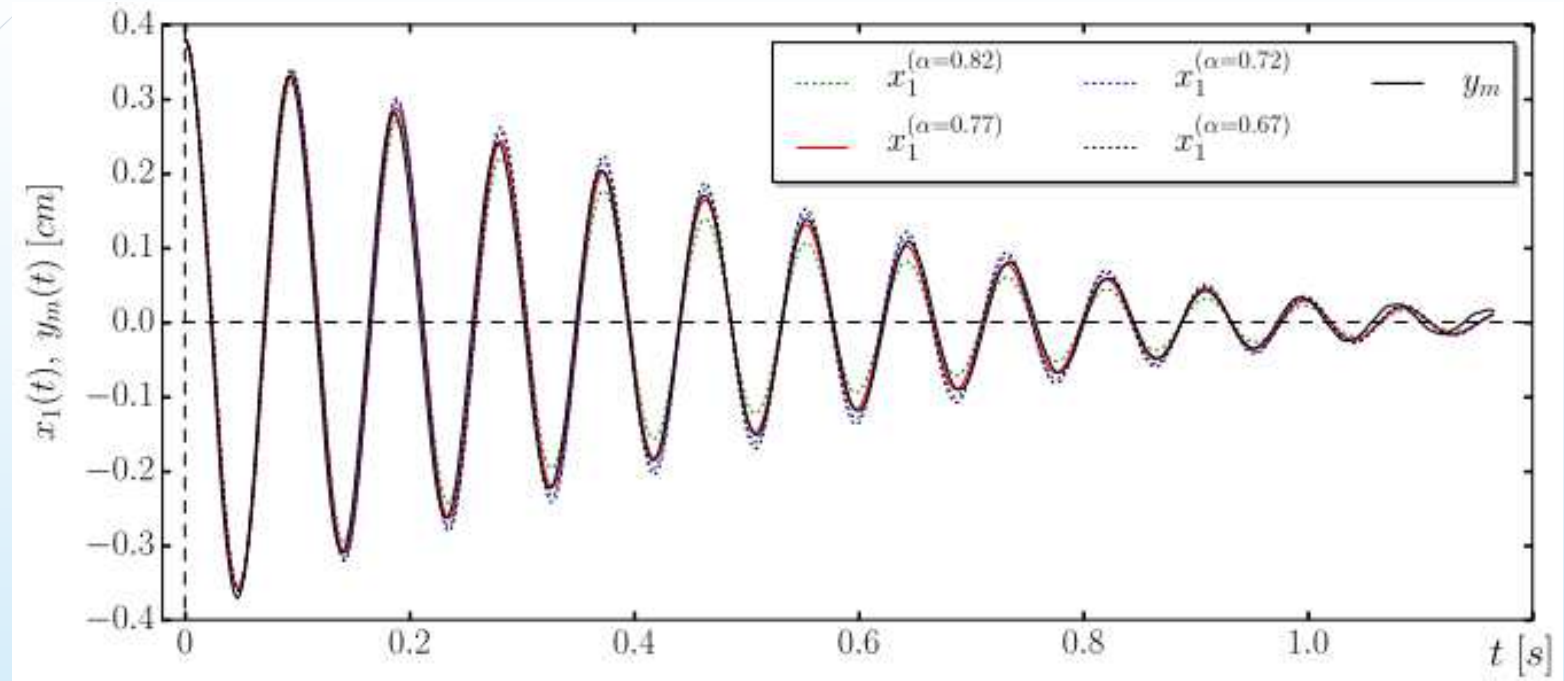


Porównanie dwóch serii czasowych, pokazujących rozbieżność między rzeczywistą trajektorią y_m , a dokładnym rozwiązaniem analitycznym y_e .

Uruchomienie 2. programu i prezentacja wyniku



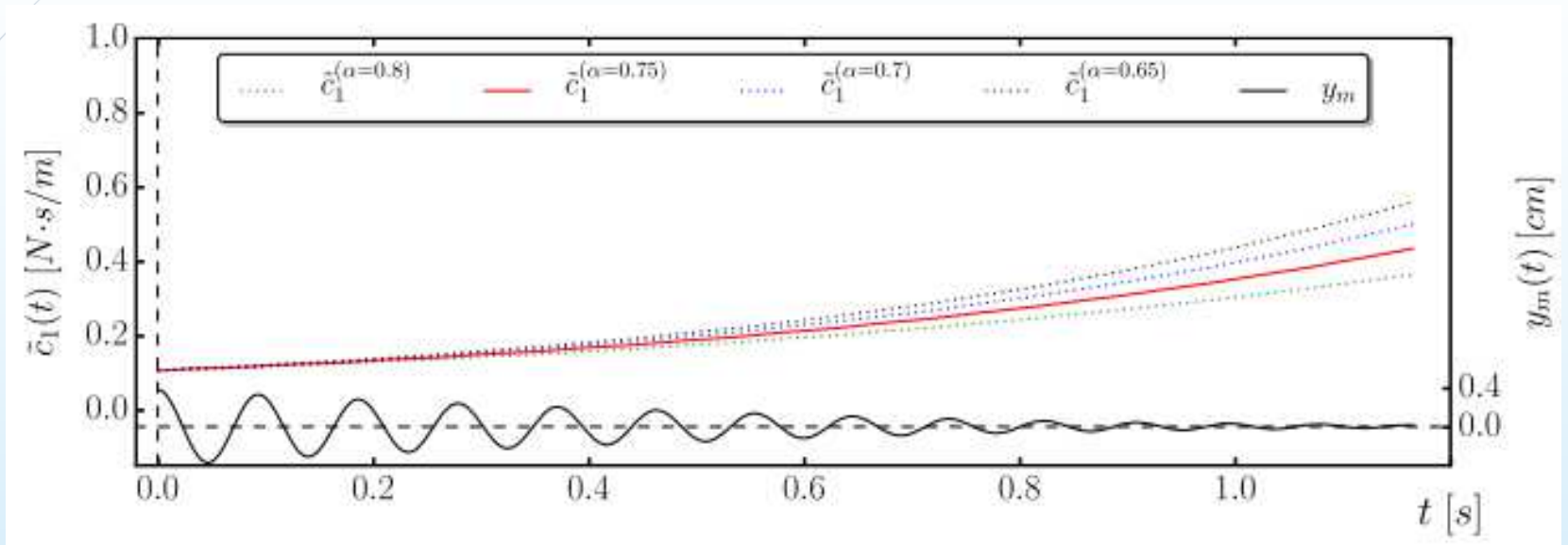
Zastosowanie w inżynierii mechanicznej cd..



Rozwiązanie numeryczne $x_1(t)$ w układzie oscylatorów sprzężonych opisanych modelem z różniczkowaniem ułamkowym na tle serii pomiarowej $y_m(t)$. Optymalne dopasowanie znajdujemy przy $\alpha = 0.77$.



Zastosowanie w inżynierii mechanicznej cd..



Uzyskana nieliniowa charakterystyka tłumienia.



Podsumowanie

- Symulacje numeryczne układów dynamicznych często wymagają opracowania dedykowanego kodu numerycznego, a czasem rozszerzenia istniejących bibliotek.
- Tłumaczenie formuł matematycznych na kod numeryczny bywa niełatwe, ale elastyczność i różnorodność procedur czyni te bardziej przejrzystymi.
- W sensie parametrycznym, złożoność przedstawionych parametrycznych układów dynamicznych jest niewielka, gdyż ze względu na skalowalność procedur numerycznych (zwiększenie liczby powtórzeń pętli, wielkość tablic itp.) można symulować i rozwiązywać również bardziej złożone układy.
- Chociaż przedmiotem tej prezentacji jest mniej złożony sprężysty układy o masie skupionej w punkcie i jednym stopniu swobody, to rozwiązywanie równań różniczkowych zwyczajnych i dalsza analiza ich rozwiązań z wykorzystaniem niektórych technik zaimplementowanych w języku programowania Python może być użyteczna w bardziej złożonych modelach zachowania kolektywnego różnych układów fizycznych.
- Niezależnie od tego, czy badania są prowadzone w obszarze nauk przyrodniczych, inżynierskich czy technicznych, niektóre metody numeryczne mogą tworzyć wspólny wątek porozumienia w zakresie analizy dynamicznej, modelowania i identyfikacji układów fizycznych.



Dziękuję za uwagę.