

# Laboratorium 10

## Równania różniczkowe - spectral bias

Mateusz Król

06/06/2024 r.

### Zadanie 1.

Dane jest równanie różniczkowe zwyczajne

$$\frac{du(x)}{dx} = \cos(\omega x) \text{ dla } x \in \Omega,$$

gdzie:

$x, \omega, u \in \mathbb{R}$ ,

$x$  to położenie,

$\Omega$  to dziedzina, na której rozwiązujemy równanie,

$\Omega = \{x \mid -2\pi \leq x \leq 2\pi\}$ ,

$u(\cdot)$  to funkcja, której postaci szukamy.

Warunek początkowy zdefiniowany jest następująco:

$$u(0) = 0.$$

Analityczna postać rozwiązania równania z warunkiem początkowym jest następująca:

$$u(x) = \frac{1}{\omega} \sin(\omega x)$$

Rozwiąż powyższe zagadnienie początkowe. Do rozwiązania użyj sieci neuronowych PINN (ang. *Physics-informed Neural Network*) Można wykorzystać szablon w pytorch-u lub bibliotekę DeepXDE.

Warstwa wejściowa sieci posiada 1 neuron, reprezentujący zmienną  $x$ .

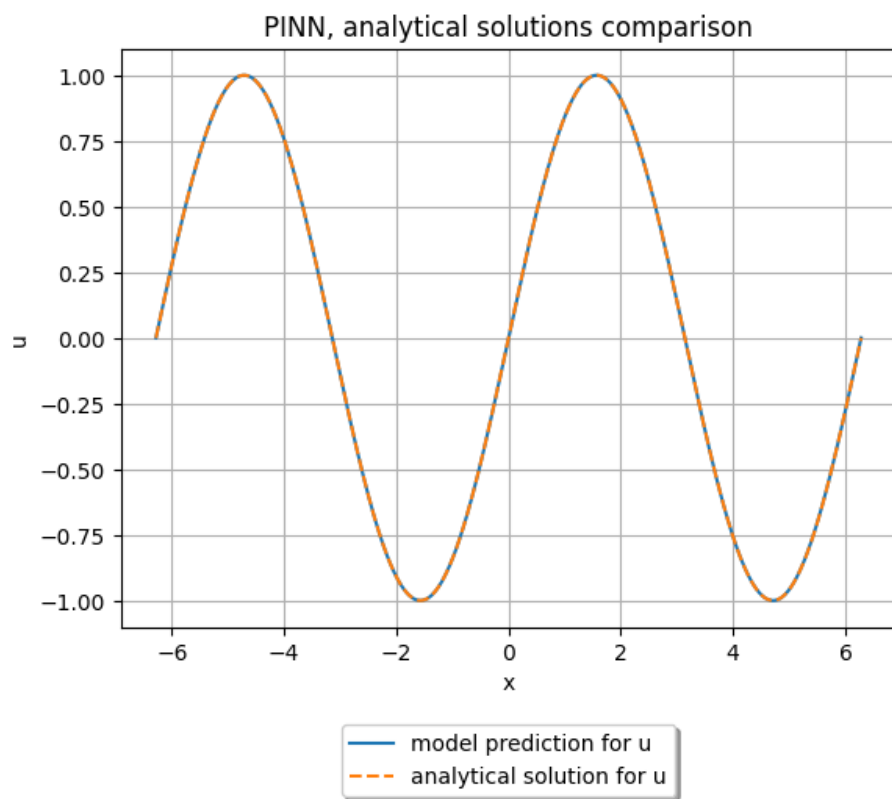
Warstwa wyjściowa także posiada 1 neuron, reprezentujący zmienną  $\hat{u}(x)$ .

Uczenie trwa przez 50000 kroków algorytmem *Adam* ze stałą uczenia równą 0.001. Jako funkcję aktywacji przyjmij tangens hiperboliczny,  $\tanh$ .

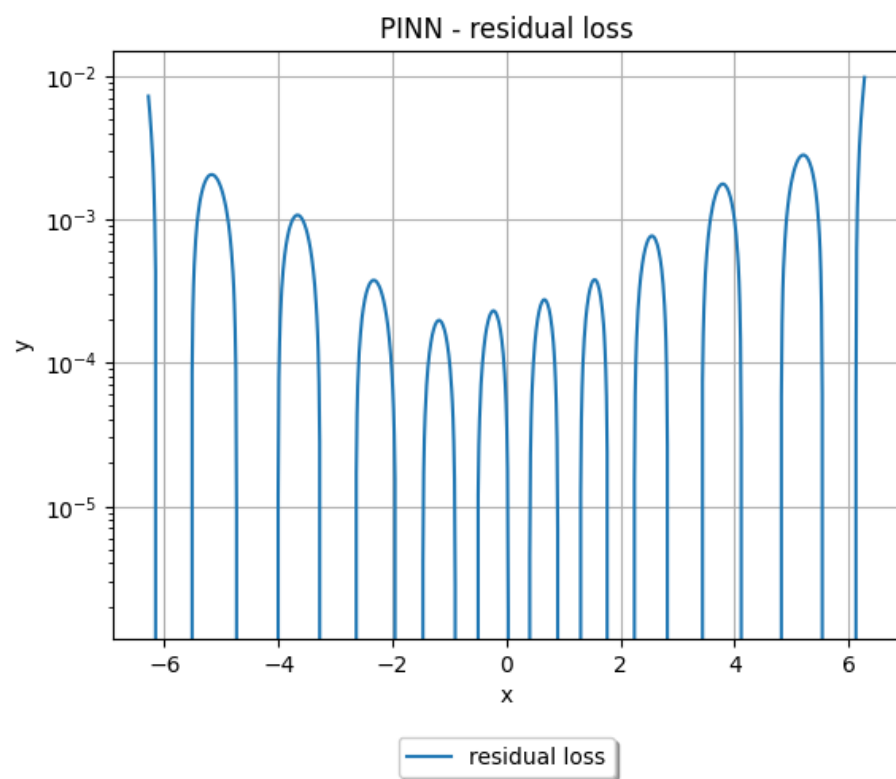
W poniższych rozwiązaniach wykorzystałem bibliotekę *DeepXDE*.

Wykres rozwiązania zadanego równania różniczkowego dla danych modelu:

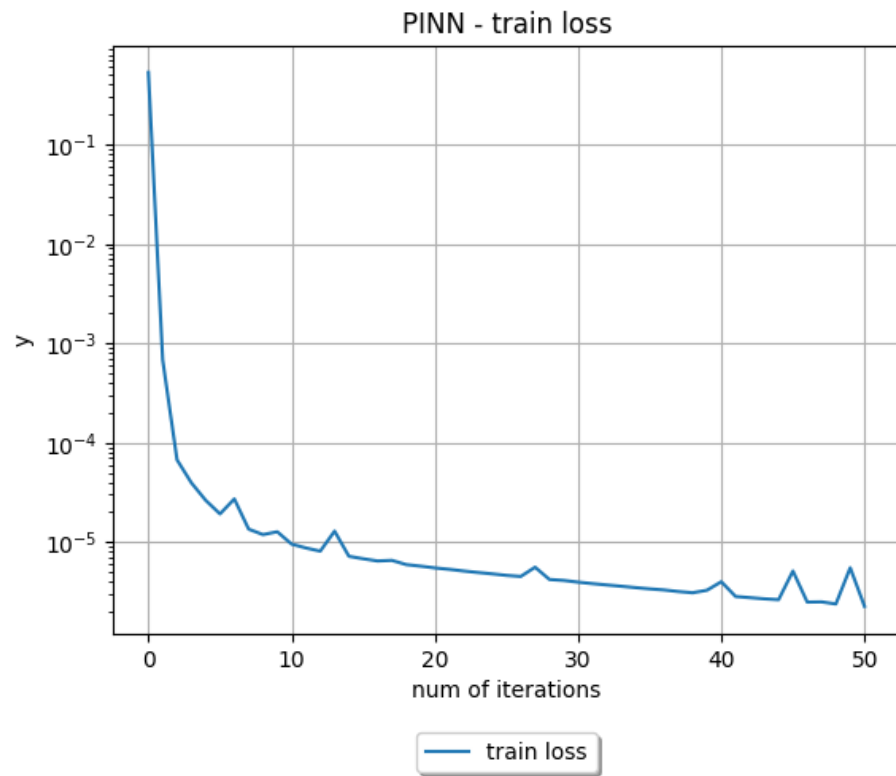
- $\omega = 1$
- 2 warstwy ukryte, 16 neuronów w każdej warstwie
- liczba punktów treningowych: 200
- liczba punktów testowych: 1000



Wykres błędu rezydualnego obliczonego rozwiązania:



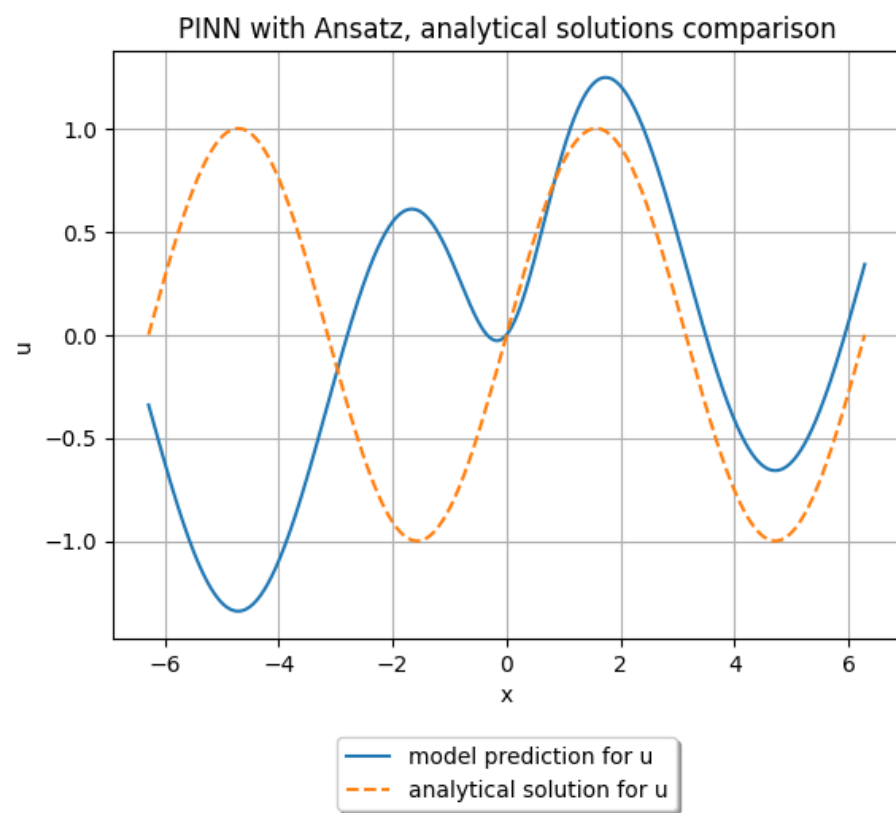
Wykres zmieniających się wartości funkcji kosztu danego modelu: (w zależności od  $1000 \cdot \text{num of iterations}$ )



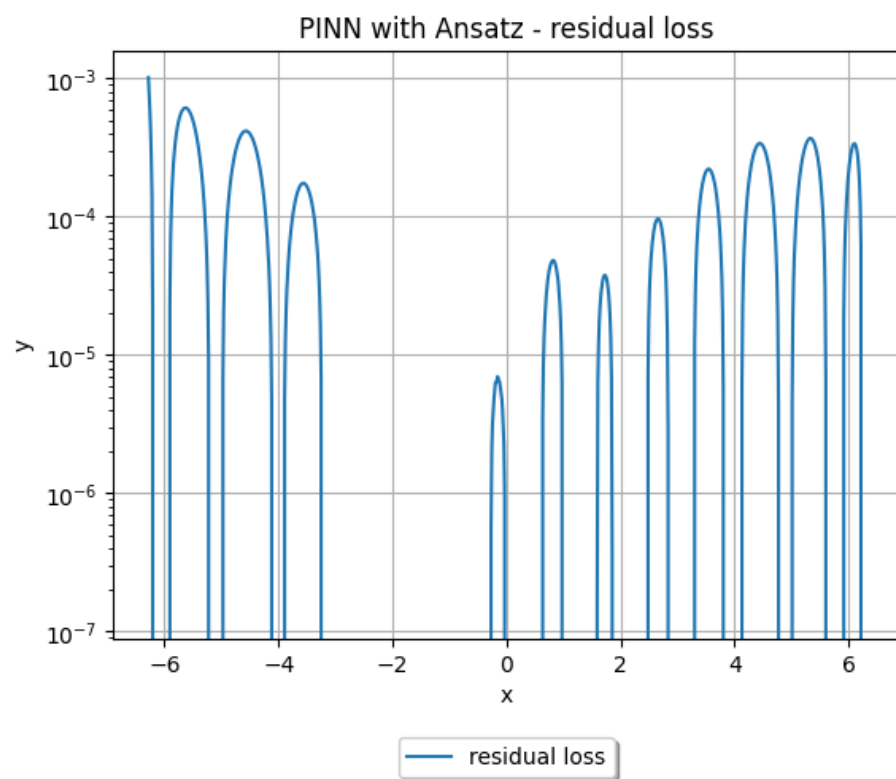
Analogiczne wykresy dla tych samych danych modelu, ale z *Ansatz*'em:

$$u(x) = \tanh(\omega x) \cdot NN(x)$$

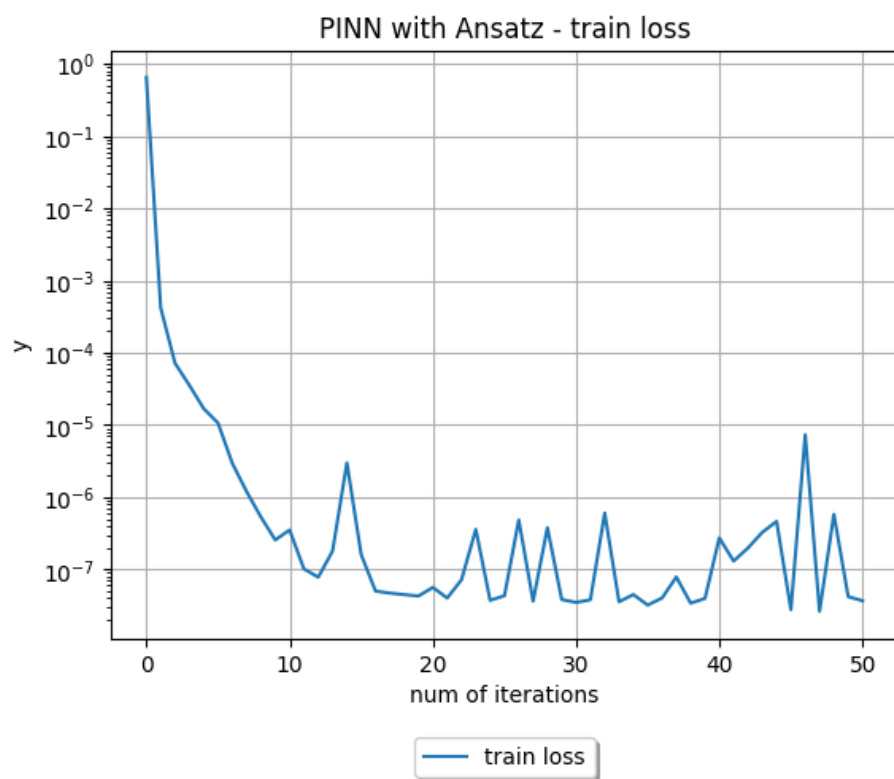
Wykres rozwiązania zadanego równania różniczkowego:



Wykres błędu rezydualnego obliczonego rozwiązania:

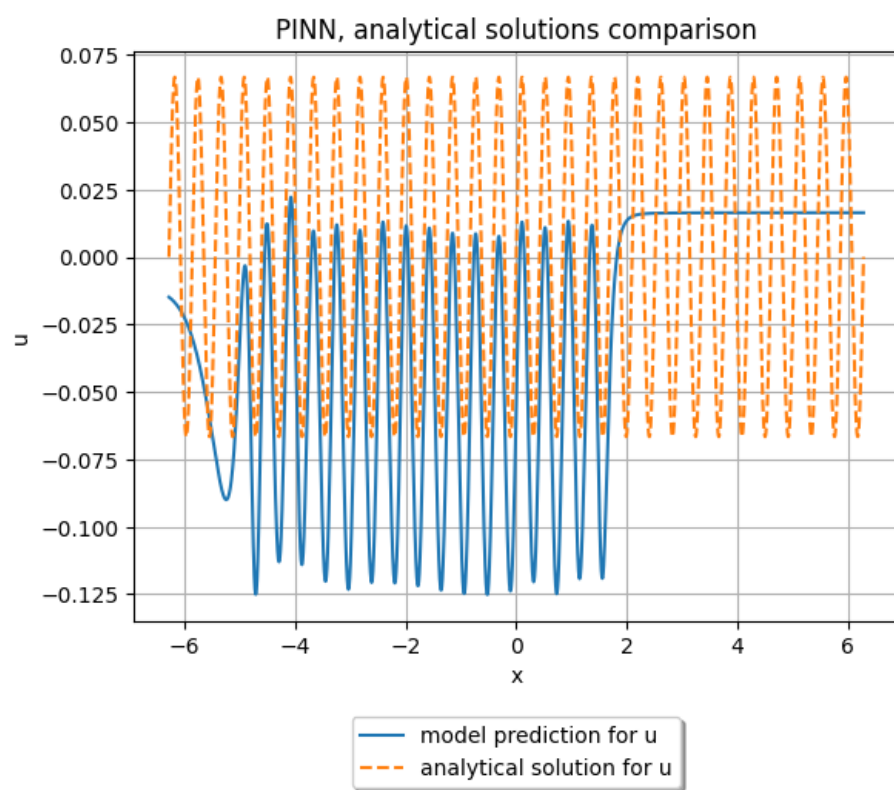


Wykres zmieniających się wartości funkcji kosztu danego modelu: (w zależności od  $1000 \cdot \text{num of iterations}$ )



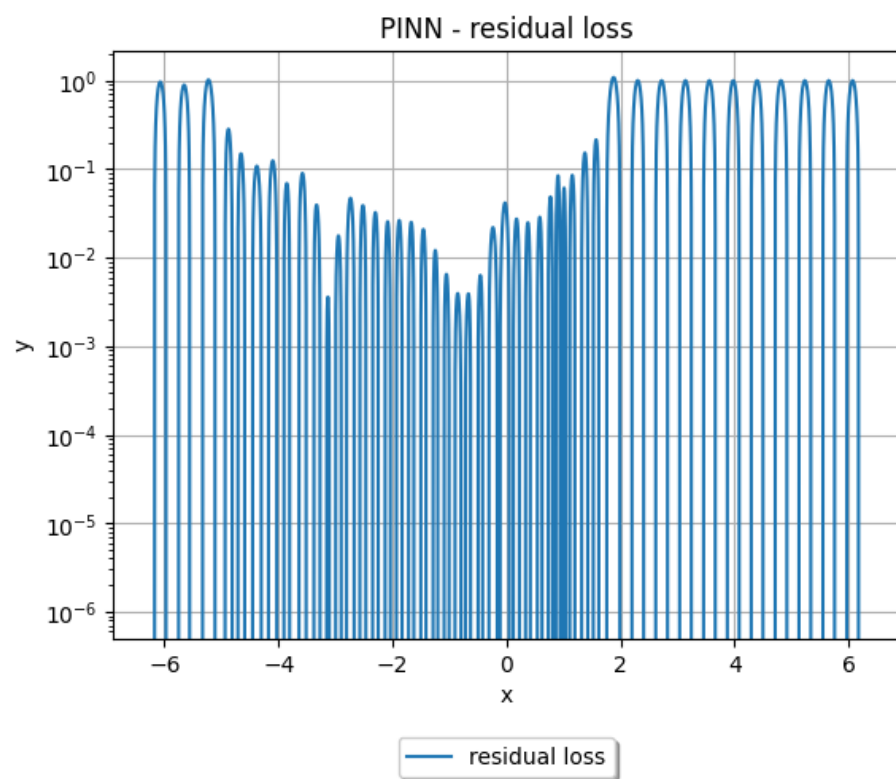
Wykres rozwiązania zadanego równania różniczkowego dla danych modelu:

- $\omega = 15$
- 2 warstwy ukryte, 16 neuronów w każdej warstwie
- liczba punktów treningowych: 3000
- liczba punktów testowych: 5000

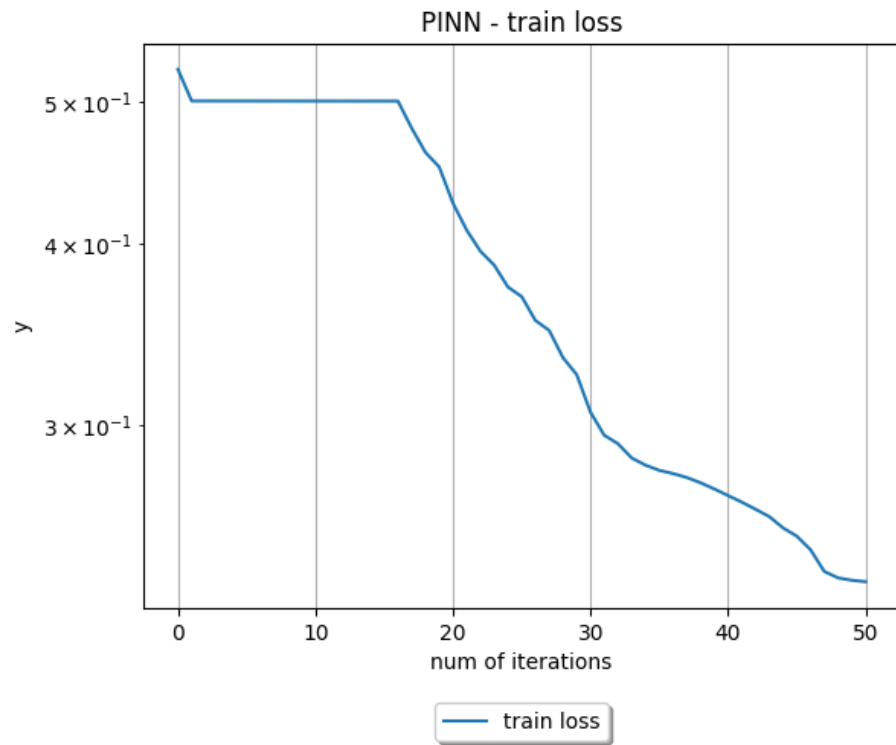


Wykres błędu rezydualnego obliczonego rozwiązania:





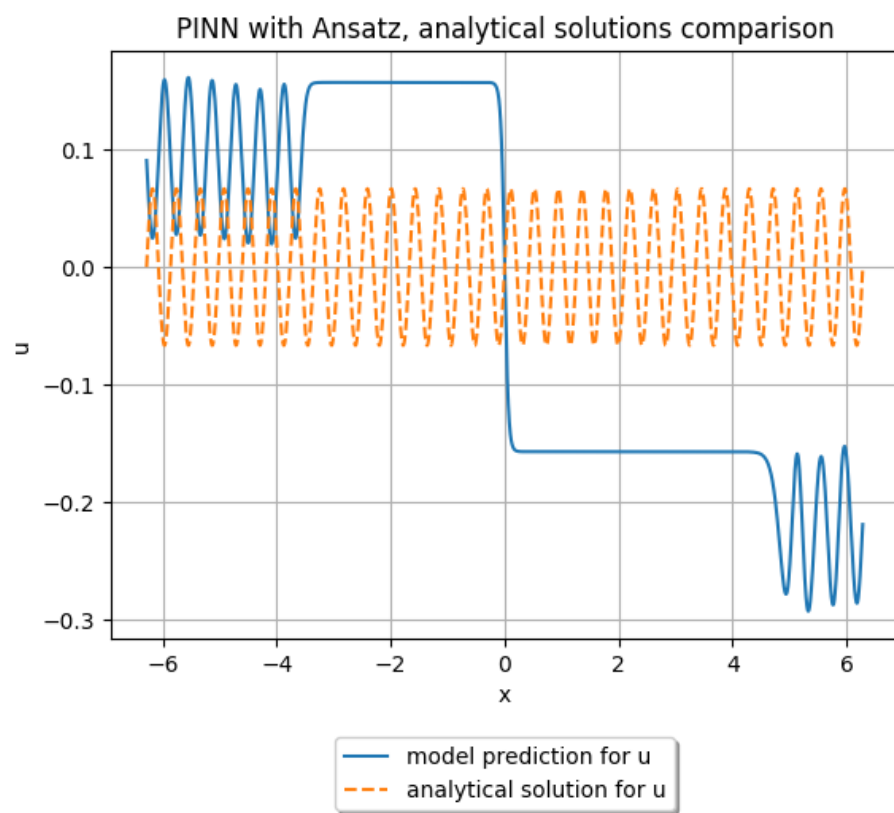
Wykres zmieniających się wartości funkcji kosztu danego modelu: (w zależności od  $1000 \cdot \text{num of iterations}$ )



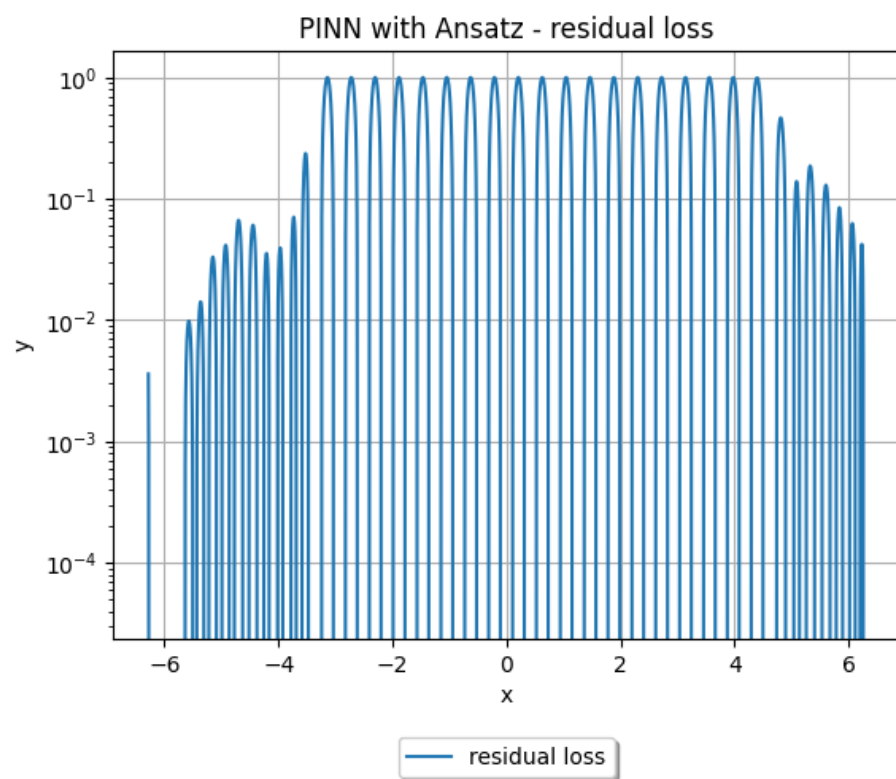
Analogiczne wykresy dla tych samych danych modelu, ale z *Ansatz*'em:

$$u(x) = \tanh(\omega x) \cdot NN(x)$$

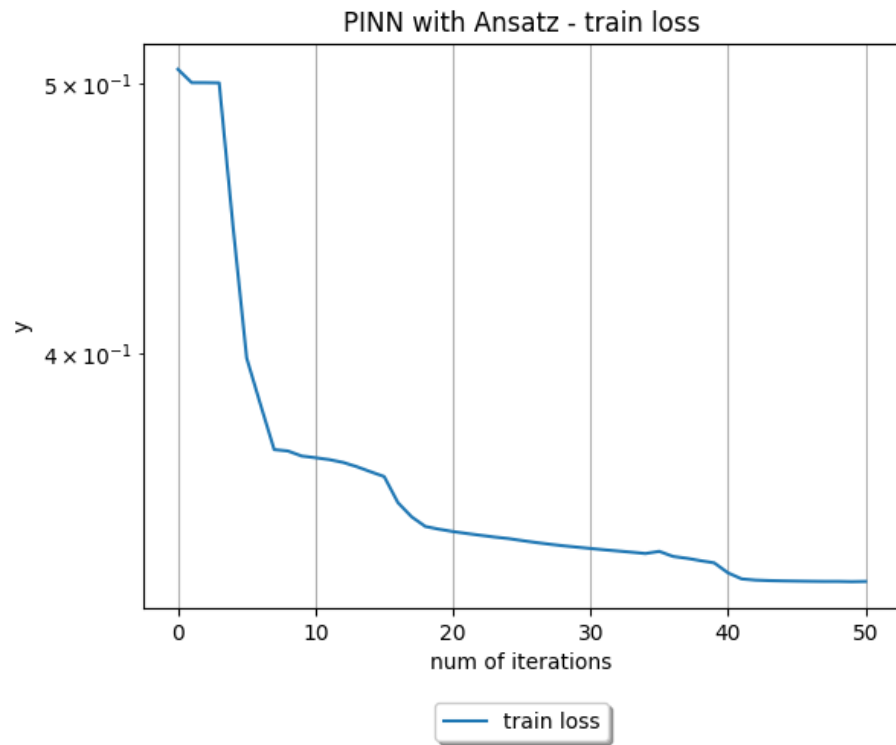
Wykres rozwiązania zadanego równania różniczkowego:



Wykres błędu rezydualnego obliczonego rozwiązania:

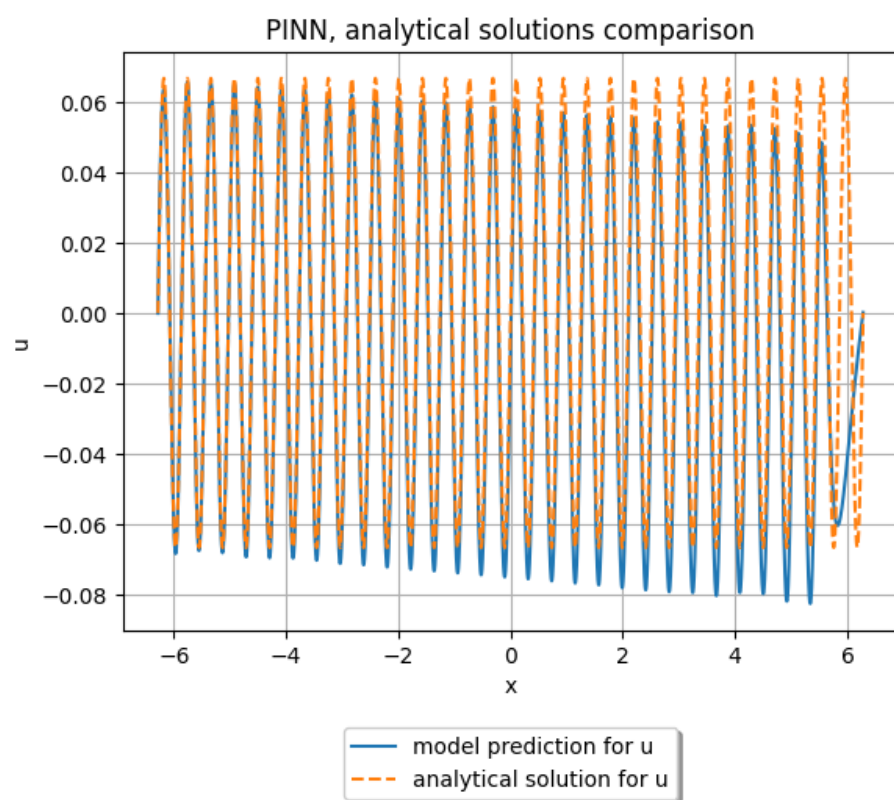


Wykres zmieniających się wartości funkcji kosztu danego modelu: (w zależności od  $1000 \cdot \text{num of iterations}$ )

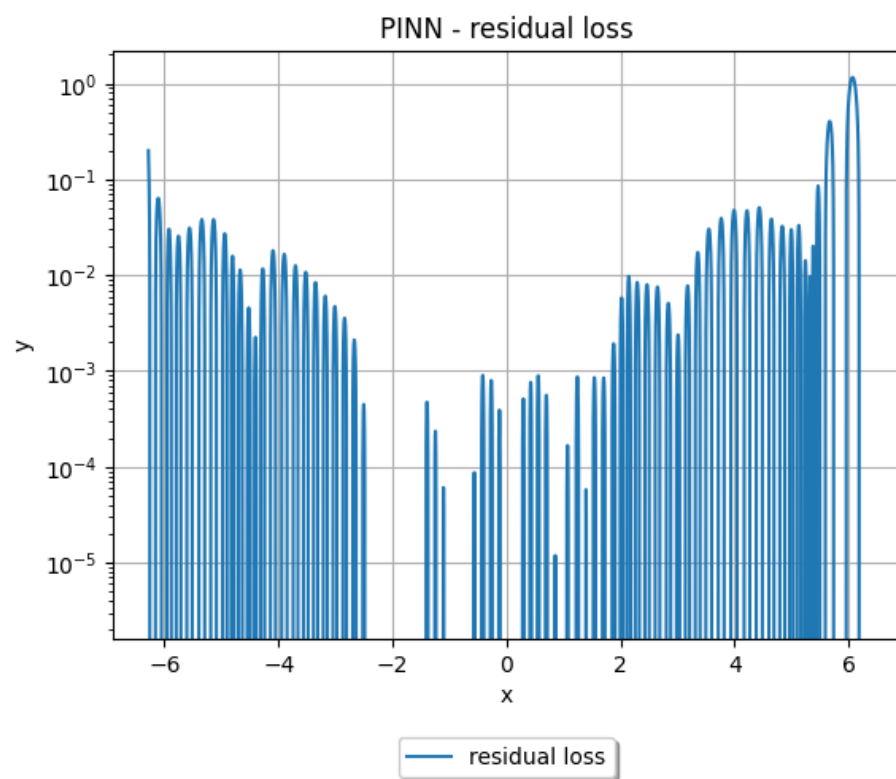


Wykres rozwiązania zadanego równania różniczkowego dla danych modelu:

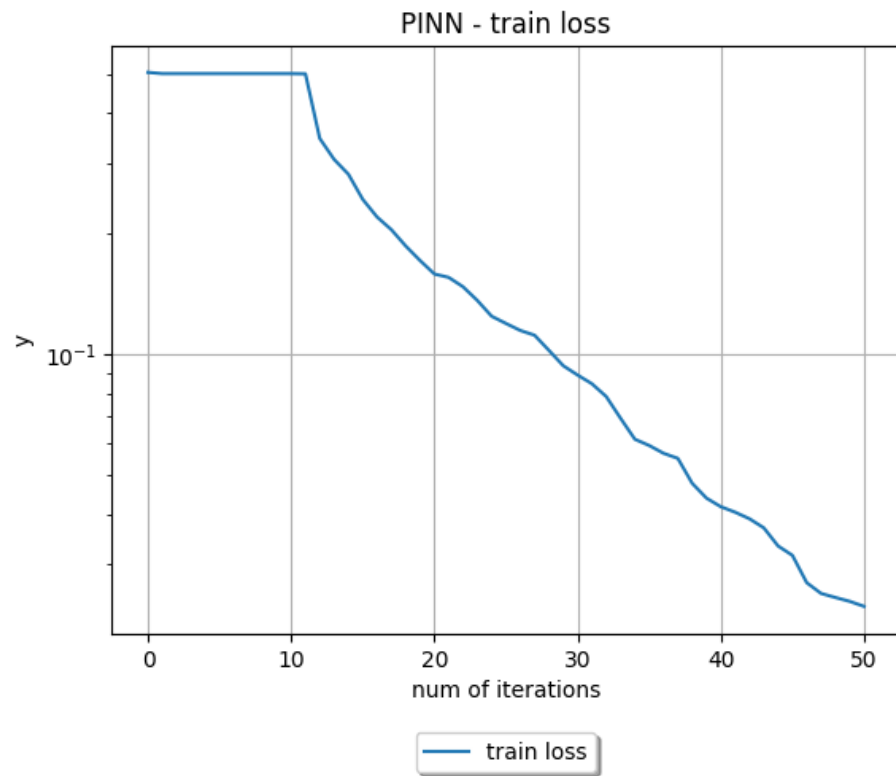
- $\omega = 15$
- 4 warstwy ukryte, 64 neuronów w każdej warstwie
- liczba punktów treningowych: 3000
- liczba punktów testowych: 5000



Wykres błędu rezydualnego obliczonego rozwiązania:



Wykres zmieniających się wartości funkcji kosztu danego modelu: (w zależności od  $1000 \cdot \text{num of iterations}$ )

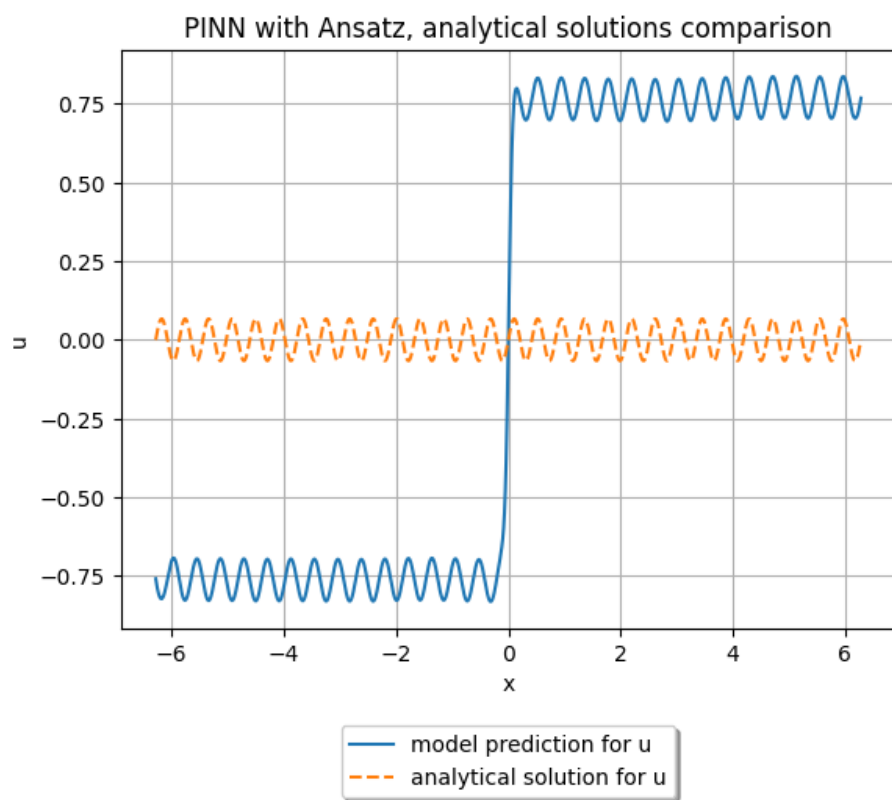


Analogiczne wykresy dla tych samych danych modelu, ale z *Ansatz*'em:

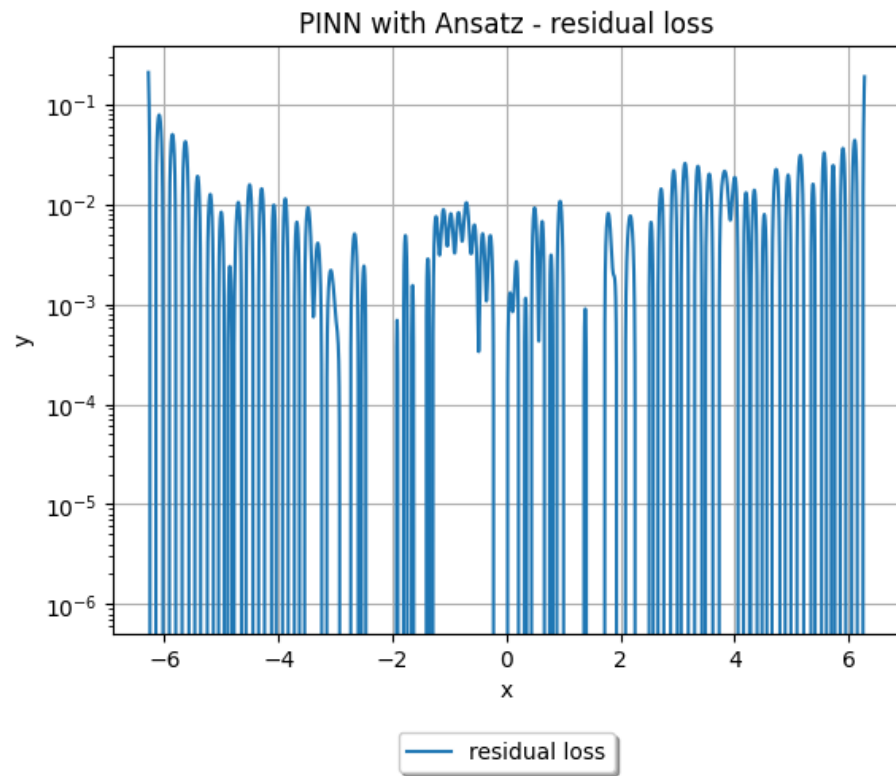
$$u(x) = \tanh(\omega x) \cdot NN(x)$$

Wykres rozwiązania zadanego równania różniczkowego:

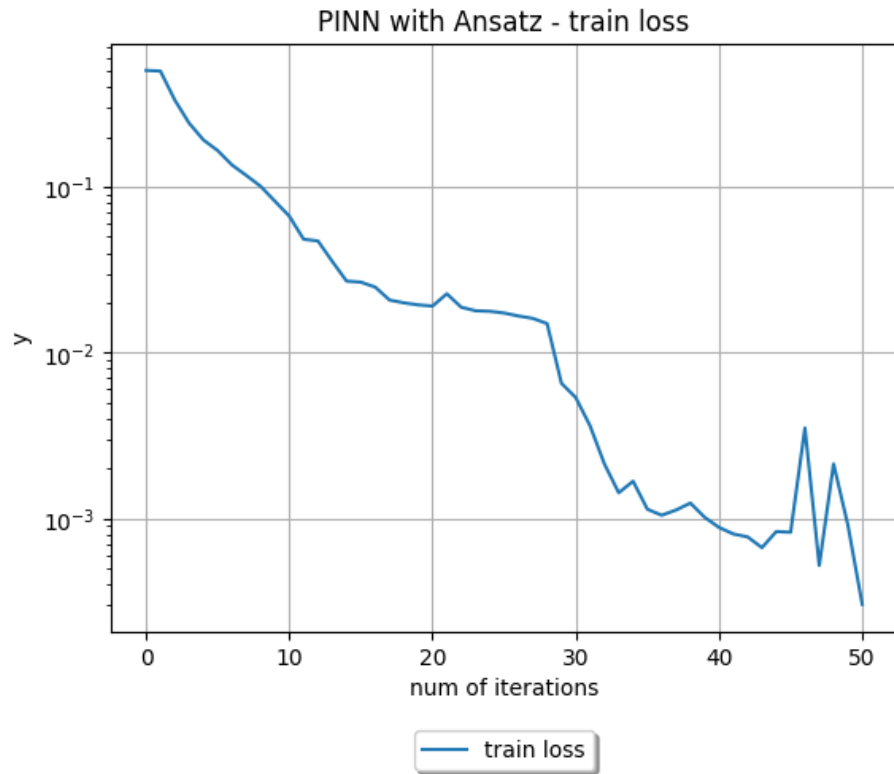




Wykres błędu rezydualnego obliczonego rozwiązania:



Wykres zmieniających się wartości funkcji kosztu danego modelu: (w zależności od  $1000 \cdot \text{num of iterations}$ )



## Wnioski

Wykorzystany model w każdej testowanej konfiguracji, nie umożliwił otrzymania akceptowalnych wyników wykorzystując dodatkowe założenie *Ansatz* w celu pozbycia się warunku *Dirichlet*'a.

Model pierwszy oraz model trzeci (w kolejności testowania), zapewniły satysfakcjonujące wyniki.

W pierwszym z testowanych modeli, wartości funkcji kosztu okazały się być zdecydowanie mniejsze dla modelu z *Ansatz*'em.