

Łukasz Stępień

25.05.2023r.

Laboratorium 10

Równania różniczkowe cząstkowe

1. Temat zadania:

Zadanie 1. Cząsteczka w dwuwymiarowej studni potencjału. Cząsteczka odbija się od ścian dwuwymiarowej nieskończonej studni potencjału o szerokości L . Zachowanie cząsteczki opisane jest bezczasowym równaniem Schrödingera

$$\frac{1}{2} \left(\frac{\partial^2 \psi(x, y)}{\partial x^2} + \frac{\partial^2 \psi(x, y)}{\partial y^2} \right) + \frac{(n_1^2 + n_2^2)\pi^2}{2L^2} \psi(x, y) = 0 \text{ dla każdego } x \in \Omega, \quad (1)$$

gdzie:

$(x, y) \in \mathbb{R}^2$ to położenie cząstki,
 Ω to dziedzina równania, $\Omega = \{(x, y) \mid -\frac{L}{2} \leq x \leq \frac{L}{2} \text{ oraz } -\frac{L}{2} \leq y \leq \frac{L}{2}\}$,
 $\psi(x, y)$ to funkcja falowa, której postaci szukamy,
 L to szerokość studni potencjału,
 n_1, n_2 to liczby kwantowe.

Jednostki dobrano w ten sposób, że iloraz $\frac{\hbar}{m}$ jest równy 1. Przyjmij $L = 2$. Warunków brzegowe zdefiniowane są następująco:

$$\psi(x, y) = 0 \text{ dla } |x| = \frac{L}{2} \text{ lub } |y| = \frac{L}{2}. \quad (2)$$

Analityczna postać rozwiązania równania (1) z warunkami brzegowymi (2) jest następująca:

$$\psi(x, y) = \begin{cases} \frac{2}{L} \sin\left(\frac{n_1\pi(x + \frac{L}{2})}{L}\right) \sin\left(\frac{n_2\pi(y + \frac{L}{2})}{L}\right) & \text{dla } x \in \Omega \\ 0 & \text{w przeciwnym wypadku} \end{cases} \quad (3)$$

Rozwiąż powyższe zagadnienie brzegowe (1),(2) dla $(n_1, n_2) \in \{1, 2\} \times \{1, 2\} = \{(1, 1), (1, 2), (2, 1), (2, 2)\}$. Do rozwiązania użyj sieci neuronowych PINN (ang. *Physics-informed Neural Network*), wykorzystując bibliotekę **DeepXDE**. Warstwa wejściowa sieci powinna posiadać 4 neurony, $L_0 = (x, y, n_1, n_2)$, kodujące odpowiednio położenie cząstki (x, y) oraz liczby kwantowe n_1, n_2 . Jako funkcję aktywacji przyjmij tangens hiperboliczny, \tanh . Przykładowe wartości hiperparametrów można znaleźć w [2, str. 221].

Stwórz następujące wykresy:

- Wykres funkcji kosztu w zależności od liczby epok
- Wykres konturowy funkcji $\psi(x)$, tj. dokładnego rozwiązania
- Wykres konturowy funkcji $\hat{\psi}(x)$, tj. rozwiązania znalezionego przez sieć neuronową
- Wykres konturowy funkcji błędu względnego $L_2: |\psi(x) - \hat{\psi}(x)|^2$.

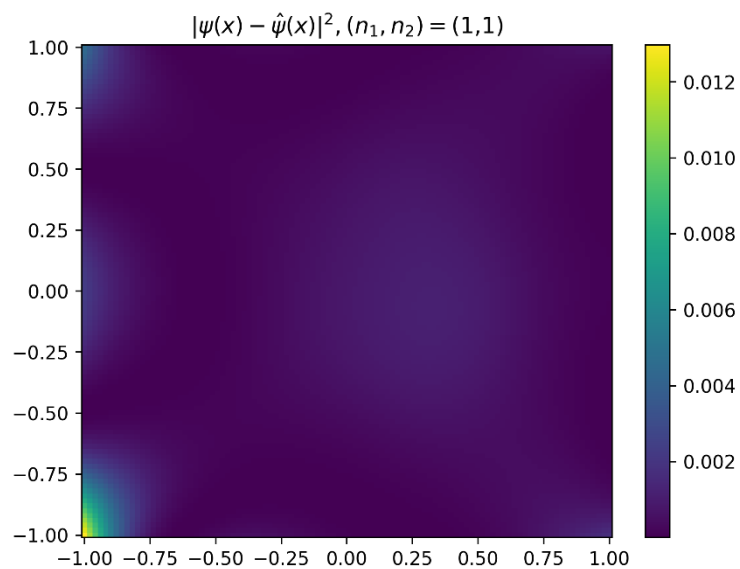
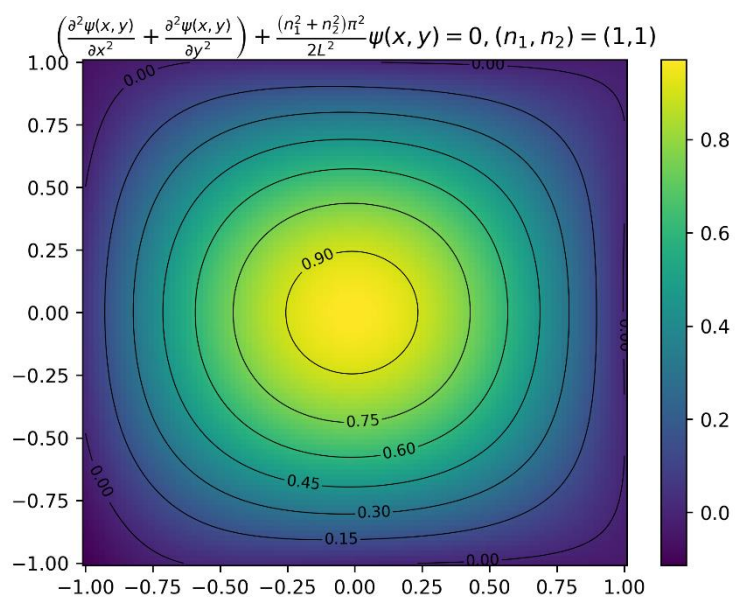
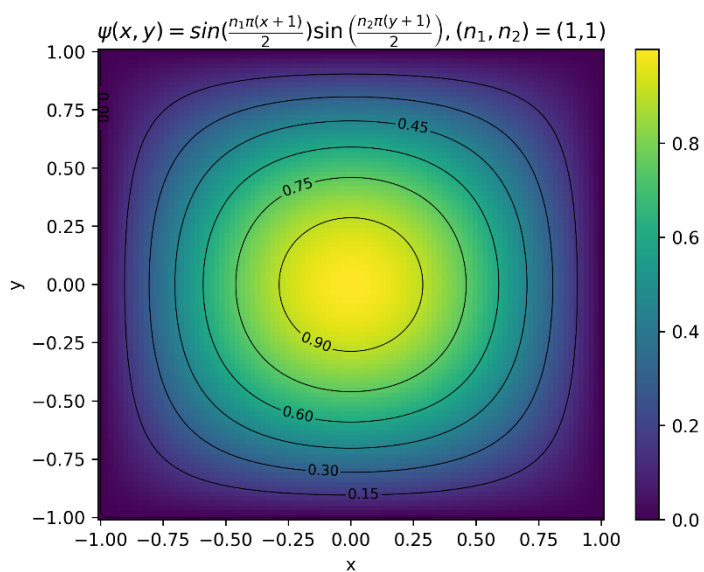
Uwaga. W przypadku wykorzystania backendu **tensorflow** należy użyć wersji tensorflow v1.

2. Implementacja:

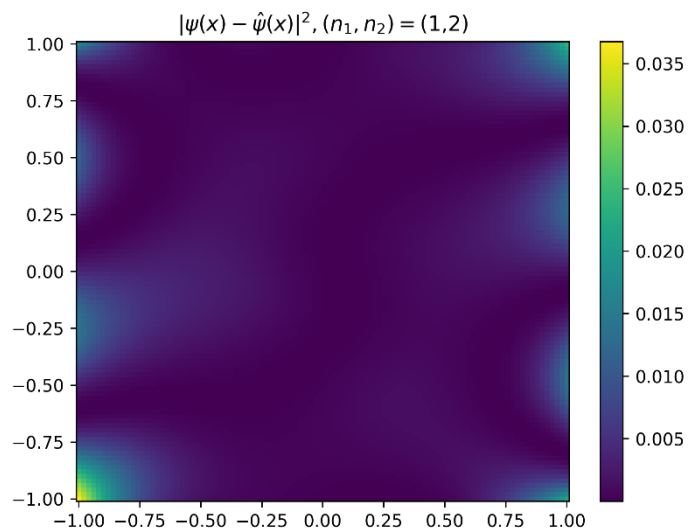
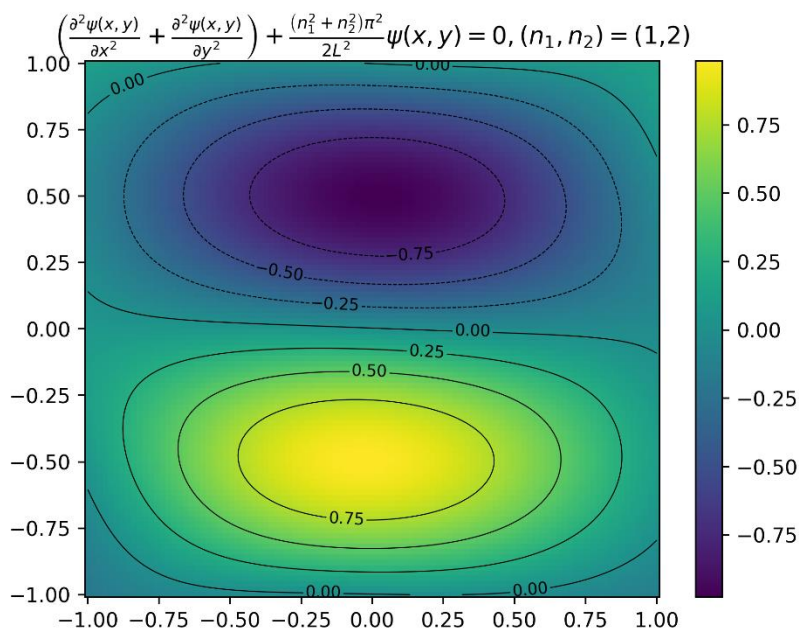
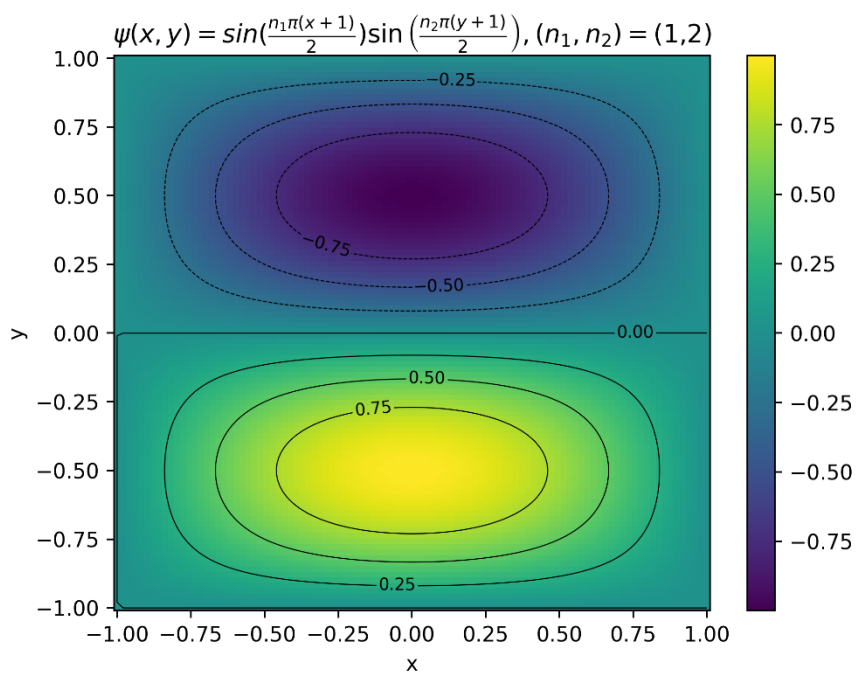
- Funkcja `psi(x, y, L, n1, n2)` oblicza wartość funkcji falowej `psi` dla podanych argumentów.
- Funkcja `show_exact(n1, n2)` generuje wykres funkcji falowej `psi` dla konkretnych wartości parametrów `n1` i `n2`. Wykorzystuje ona funkcję `psi` do obliczenia wartości funkcji `psi` dla różnych punktów w dziedzinie. Następnie rysuje wykres z użyciem biblioteki `matplotlib`.
- Następnie zdefiniowane są funkcje `pde`, `pde11`, `pde12`, `pde21`, `pde22`, które opisują równania różniczkowe cząstkowe (PDE) dla konkretnych wartości parametrów `n1` i `n2`. Funkcja `pde` oblicza wartość równania PDE dla danej funkcji `psi` i jej pochodnych. Funkcje `pde11`, `pde12`, `pde21`, `pde22` są specjalizacjami funkcji `pde` dla konkretnych wartości parametrów `n1` i `n2`.
- Funkcja `solve` rozwiązuje równanie PDE dla podanej funkcji `pde` oraz punktów i wartości brzegowych. Tworzona jest geometria w postaci prostokąta, a następnie definiowane są warunki brzegowe w postaci brzegów Dirichleta oraz punkty, które są wyliczane z oryginalnej funkcji. Wykorzystuje ona klasę `PDE` z biblioteki `deepxde.data` do zdefiniowania danych wejściowych. Tworzony jest również model sieci neuronowej, który jest kompilowany i trenowany z użyciem danych wejściowych. Funkcja zwraca model oraz stan treningu.
- Funkcja `draw_model` generuje wykres rozwiązania równania PDE uzyskanego z modelu. Oblicza wartości `psi` na siatce, a następnie rysuje wykres z użyciem biblioteki `matplotlib`.
- Następnie dla każdej kombinacji wartości parametrów `n1` i `n2`, wywoływana jest funkcja `solve` i `draw_model` dla odpowiedniego równania PDE. Wyniki są przedstawiane na wykresach.
- Funkcja `draw_error` oblicza błąd między rozwiązaniem dokładnym (funkcją `psi`) a rozwiązaniem uzyskanym z modelu. Następnie generuje wykres błędu na podstawie obliczonych wartości.

3. Wyniki:

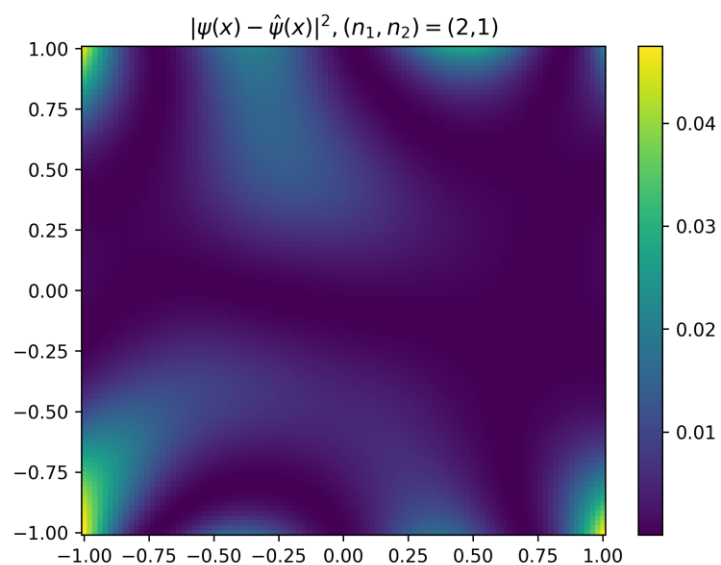
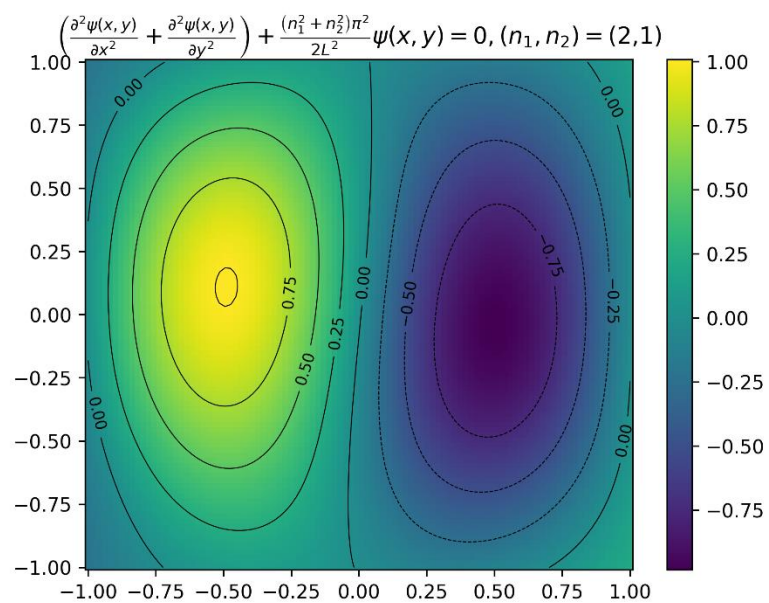
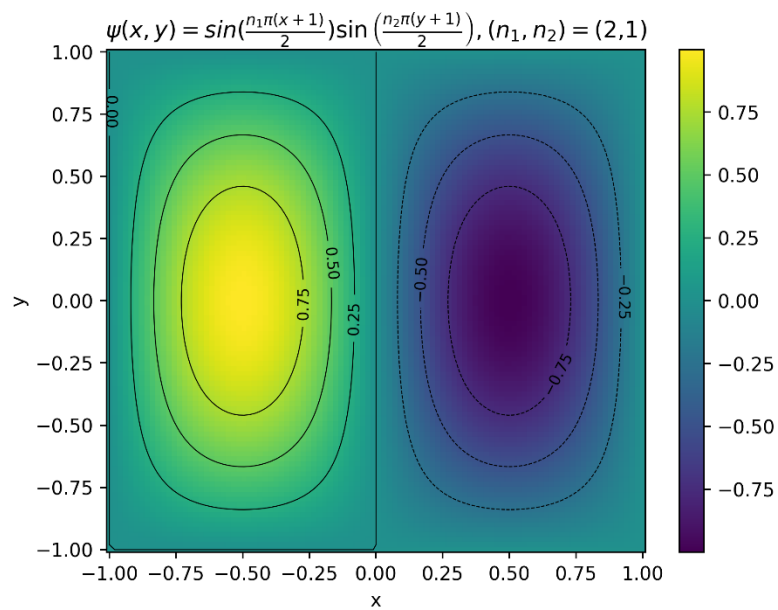
- Wykresy dla $(n_1, n_2) = (1, 1)$:



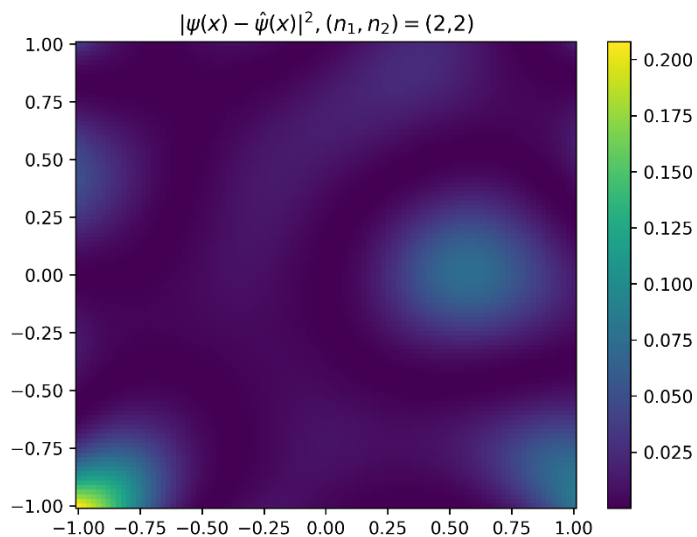
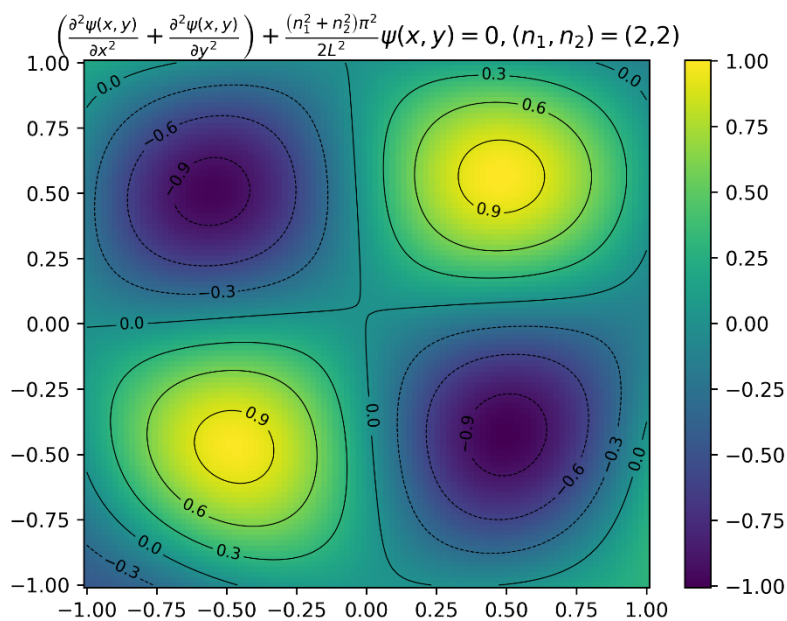
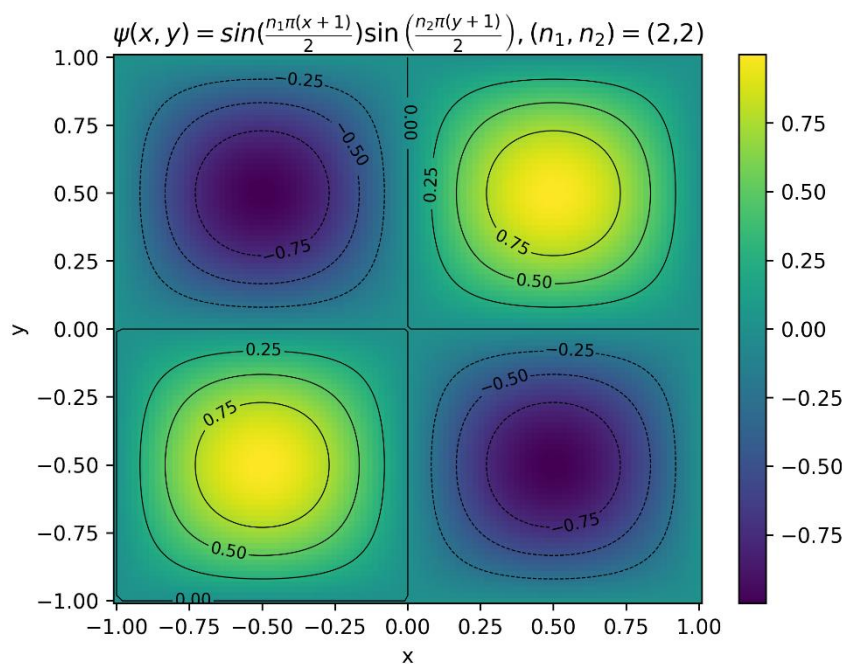
- Wykresy dla $(n_1, n_2) = (1, 2)$:



- Wykresy dla $(n_1, n_2) = (2, 1)$:



- Wykresy dla $(n_1, n_2) = (2, 2)$:



4. Wnioski:

- Z przedstawionych wykresów można wnioskować, że dzięki bibliotece DeepXDE udało się uzyskać wiarygodne wyniki w każdym przypadku obliczanej funkcji falowej. Z wykresów błędu można zauważyć, że jest on niewielki.
- Bez dodania początkowych punktów otrzymywano rozwiązanie trywialne w postaci funkcji tożsamościowo równej 0. Dzięki dodaniu zadanych punktów na początku obliczeń wyniki stały się już zbliżone do nietrywialnego rozwiązania tego równania.

5. Bibliografia:

- <https://deepxde.readthedocs.io/en/latest/>