

# Dokumentacja techniczna projektu

## Wersja 2

Hubert Rączkiewicz, Mikołaj Suszek

22 grudnia 2025

## 1 Wstęp

Odkryte przez Erwina Schrödingera równanie falowe legło u podstaw nowego działu fizyki - mechaniki kwantowej. Dokonany przełom znalazł zastosowanie w wielu dziedzinach nauk ścisłych.

## 2 Cel projektu

Celem projektu jest utworzenie interaktywnej animacji wizualizującej poszczególne stany energetyczne w atomie wodoru, zarówno dla ustalonych liczb kwantowych jak i stanów przejściowych. Pozwoli to na lepsze zrozumienie konfiguracji elektronowych i wynikających z nich właściwości chemicznych i fizycznych wodoru.

## 3 Opis zjawiska fizycznego

Mianem chmury elektronowej w atomie określa się zagęszczenie prawdopodobieństwa znalezienia elektronu w danym punkcie w przestrzeni. Jej kształt uzależniony jest od trzech liczb kwantowych: głównej, pobocznej i magnetycznej. Przy czym, liczby główna i poboczna wpływają na radialny rozkład prawdopodobieństwa, zaś rozkład kątowy określają liczby poboczna i magnetyczna.

Ilościowo prawdopodobieństwo znalezienia elektronu w danym punkcie opisuje kwadrat modułu funkcji falowej elektronu. Istotą problemu jest analityczne lub numeryczne wyznaczenie funkcji falowej, będącej rozwiązaniem równania Schrödingera. Obecnie, równanie to zostało rozwiązane analitycznie wyłącznie dla atomu wodoru. Otrzymana funkcja falowa dana jest wzorem:

$$\Psi(r, \theta, \varphi, t) = r^l \left( \frac{2}{na_0} \right)^{l+1} L_{n-l-1}^{2l+1} \left( \frac{2r}{na_0} \right) e^{-r/na_0} \cdot (-1)^m \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} P_{l,m}(\cos \theta) e^{im\varphi} \cdot e^{\frac{iEt}{\hbar}} \quad (1)$$

## 4 Opis wykorzystywanych narzędzi

Kod projektu będzie napisany w języku Python zgodnie ze standardem wersji: 3.9. Kod będzie podzielony pod względem funkcjonalności na moduły. Rozwój projektu i te-

sty zostaną przeprowadzone w zintegrowanym środowisku programistycznym PyCharm. Wstępnie do bezpośredniego wykorzystania wytypowano biblioteki:

1. NumPy, SciPy - do przeprowadzania obliczeń numerycznych
2. SymPy - do interpolacji danych
3. PyQt, PyQtGraph - do stworzenia graficznego interfejsu użytkownika i rysowania wykresów

## 5 Ogólny opis projektu

Elementy bazowe:

- Moduł wprowadzający typy i klasy pomocnicze.
- Moduł obliczający wartości funkcji falowej atomu wodoru i na ich podstawie generujący dane do wykresów.
- Moduł odpowiedzialny za graficzny interfejs użytkownika, pozwalający w przystępny sposób zmieniać parametry wejściowe programu.
- Moduł wizualizujący na wykresach obliczone wartości.

Elementy dodatkowe:

- Moduł interwału czasowego umożliwiający dynamiczne odświeżanie wykresu dla (niestacjonarnych) stanów przejściowych atomu wodoru.
- Zestaw klas i funkcji pomocniczych optymalizujących szybkość działania programu.
- Plik wykonywalny, pakujący ogólną funkcjonalność programu i uruchamiający go poza konsolą.

## 6 Specyficzne wymagania

a) wymagania funkcjonalne, program:

- oblicza wartości funkcji falowej atomu wodoru dla danej siatki przestrzennej oraz określonych liczb kwantowych
- wizualizuje gęstość prawdopodobieństwa na trójwymiarowym, interaktywnym wykresie punktowym lub objętościowym
- dynamicznie odświeża wykres dla stanów niestacjonarnych
- umożliwia tworzenie migawek wykresów (zapis chwilowego stanu w postaci pliku graficznego na dysku)

b) wymagania niefunkcjonalne, program:

- pozwala na zmianę parametrów wejściowych z poziomu graficznego interfejsu użytkownika
- umożliwia płynne generowanie obrazu na poziomie 20 klatek na sekundę na przeciętnym urządzeniu
- pakuje zawarte funkcjonalności w pojedynczy plik wykonywalny

## 7 Harmonogram pracy nad projektem

Tydzień 1 (8.12.2025 - 14.12.2025)	Opracowanie szkieletu projektu. Utworzenie repozytorium Git <sup>1</sup> , poszczególnych modułów wraz z zarysem ich funkcjonalności.
Tydzień 2 (15.12.2025 - 4.01.2026)	Praca nad modułem wyliczającym wartości funkcji faleowej, modułem wizualizującym dane na wykresie i modułem dynamicznie aktualizującym dane.
Tydzień 3 (5.01.2026 - 11.01.2026)	Dalsza praca nad wizualizacją danych. Dodanie graficznego interfejsu użytkownika. Implementacja technik optymalizacyjnych - obliczanie wartości stałych, odwleczenie wykonania, przechowywanie wartości obliczonych ( <i>caching</i> ), ucinanie (pomijanie) klatek.
Tydzień 4 (12.01.2026 - 18.01.2026)	Dalsza optymalizacja kodu. Podniesienie niezawodności na błędy w trakcie wykonywania. Utworzenie pliku wykonywalnego programu.

## 8 Zmiany względem wersji 1

W związku z częściową zmianą koncepcji realizacji projektu i koniecznością poprawy opisu wymagań funkcjonalnych i niefunkcjonalnych sporządzono wersję 2 dokumentacji technicznej projektu. Względem wersji 1, dokonano następujących zmian:

1. Poprawiono opis wymagań funkcjonalnych i niefunkcjonalnych programu
2. Zmieniono opis narzędzi planowanych do wykorzystania w zakresie bibliotek interfejsu graficznego.
3. Zmodyfikowano harmonogram wykonywania projektu w zakresie zadań niezrealizowanych. Wybrane zagadnienia przesunięto do szybszego wykonania, precyzyjniej rozdzielono złożone zadania. Dokonane zmiany nie powinny wpływać na czas zakończenia prac nad projektem.
4. Dodano link do repozytorium Github
5. Wybrane fragmenty zmieniono pod względem stylistycznym, aby bardziej akuratnie odzwierciedlały założenia projektu

---

<sup>1</sup>Link do repozytorium Github: <https://github.com/Serious-Physicists-Inc/ProjektWDMZ>