

Praca Dyplomowa Inżynierska

Adrian Rostek
205860

Wykorzystanie technologii webowych i języka Python do stworzenia aplikacji edukacyjnej z mechaniki kwantowej

Utilizing web technologies and Python language to create quantum physics
educational application

Praca dyplomowa na kierunku:
Informatyka

Praca wykonana pod kierunkiem
dr Andrzeja Zembrzuskiego
Instytut Informatyki Technicznej
Katedra Sztucznej Inteligencji

Warszawa, rok 2024



SZKOŁA GŁÓWNA
GOSPODARSTWA
WIEJSKIEGO

Wydział Zastosowań
Informatyki
i Matematyki

Oświadczenie Promotora pracy

Oświadczam, że niniejsza praca została przygotowana pod moim kierunkiem i stwierdzam, że spełnia ona warunki do przedstawienia tej pracy w postępowaniu o nadanie tytułu zawodowego.

Data

Podpis promotora

Oświadczenie autora pracy

Świadom/a odpowiedzialności prawnej, w tym odpowiedzialności karnej za złożenie fałszywego oświadczenia, oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami prawa, w szczególności z ustawą z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (Dz. U. 2019 poz. 1231 z późn. zm.)

Oświadczam, że przedstawiona praca nie była wcześniej podstawą żadnej procedury związanej z nadaniem dyplomu lub uzyskaniem tytułu zawodowego.

Oświadczam, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną. Przyjmuję do wiadomości, że praca dyplomowa poddana zostanie procedurze antyplagiatowej.

Data

Podpis autora pracy

Streszczenie

Wykorzystanie technologii webowych i języka Python do stworzenia aplikacji edukacyjnej z mechaniki kwantowej

...Treść streszczenia...

Słowa kluczowe – Edukacja, Fizyka kwantowa, Funkcja falowa, Wizualizacja

Summary

Utilizing web technologies and Python language to create quantum physics educational application

...Treść angielskiego streszczenia...

Keywords – Education, Quantum physics, Wave function, Visualization

Spis treści

1	Wstęp	9
1.1	Cel i motywacja pracy	9
1.2	Tematyka i struktura pracy	10
2	Wykorzystane technologie	11
2.1	Popularne technologie webowe - HTML, CSS i TypeScript	11
2.2	Biblioteki Chart.js i MathJax	11
2.3	Język Python	11
2.4	Framework Tauri	12
3	Podstawy teoretyczne	13
3.1	Zagadnienia matematyki wyższej	13
3.2	Falowa natura materii	15
3.3	Równanie Schrödingera	15
3.4	Znajdowanie równania funkcji falowej	16
4	Budowa i struktura aplikacji	17
4.1	Struktura aplikacji	17
4.2	Interfejs	17
4.3	Typescript i manipulacja DOM	17
4.4	Obliczenia fizyczne w Pythonie	17
4.5	Interfejs pomiędzy TypeScriptem i Pythonem	18
4.6	Wdrożenie i dystrybucja aplikacji	18
4.7	Problemy, ograniczenia i możliwości rozwoju	18
5	Interfejs aplikacji	19
5.1	Ekran główny	19
5.2	Interaktywna wizualizacja	19
5.3	Transkrypcja	19

6	Podsumowanie i wnioski	20
7	Bibliografia	21

1 Wstęp

Stworzenie teorii mechaniki kwantowej w 1925 r.[przypis?] okazało się podstawą dzisiejszej cywilizacji[przypis]. Zawdzięczamy jej m.in. tranzystory tworzące komputery, mikroskopy tunelowe o niebywalej precyzji oraz reaktory jądrowe, bez których ciężko wobrazić sobie dzisiejszą energetykę[przypis]. Mimo to jest to bardzo nieintuicyjny i przez większość ludzi niezrozumiały dział fizyki[przypis]. Na temat ten napisane zostały liczne publikacje[przypis], jednak profesjonalny język i matematyka wyższa mogą sprawić dużo trudności w zrozumieniu nawet podstawowych konceptów tej teorii.

1.1 Cel i motywacja pracy

Celem pracy jest stworzenie aplikacji, która ma ułatwić naukę zagadnień z zakresu mechaniki kwantowej. Zagadnienia przedstawiane są w interaktywny celom podtrzymania uwagi i zainteresowania tym nietrywialnym tematem. Osiągnięte to zostało poprzez wykorzystanie licznych symulacji, na których efekt końcowy bezpośredni wpływ ma użytkownik, jednocześnie stosując samouczek, który te efekty odpowiednio tłumaczy.

Osobiście temat mechaniki kwantowej uważam za niezwykle ciekawy, więc napisanie tej pracy motywowane jest chęcią poszerzenia swojej wiedzy w jakotym obszarze, jak i zastosowaniu nabytej wiedzy informatycznej w stworzeniu praktycznego narzędzia. Za interesujące również uważam symulację funkcji falowej w przeciwieństwie do przypatrywania się statycznym jej wykresom na papierze czy w plikach pdf. Aplikacja kierowana jest do osób chcących nauczyć się wstępnych zagadnień mechaniki kwantowej, jednak bez konieczności sięgania po profesjonalną literaturę. Do pełnego zrozumienia wszystkich zagadnień potrzebna jest znajomość matematyki wyższej, jednak nawet bez tej wiedzy użytkownik może wynieść z aplikacji dużo nowych informacji. Może ona być więc użyteczna zarówno dla osób nie będących ściśle związanych z naukami matematycznymi i fizycznymi, jak i **studentów kierunków fizycznych?**.

1.2 Tematyka i struktura pracy

Aplikacja przytacza kontekst historyczny dziedziny fizyki jaką jest mechanika kwantowa jak i tłumaczy falowo korpuskularną naturę cząstek. Główna część jednak skupia się na typowych rozwiązaniach równania Schrödingera niezależnego od czasu, a dokładniej:

- Cząstki swobodnej
- Nieskończonej studni potencjału
- Skończonej studni potencjału
- Progu potencjału
- Bariery potencjału

Wy tłumaczone są też zjawiska tunelowe i skwantowanych stanów energetycznych jako konsekwencja dotychczas przyswojonych zagadnień. Jest to często stosowana kolejność wprowadzania do tych zagadnień[przypis?], ponieważ każdy kolejny przypadek bazuje na poprzednim, wprowadzając jednak stopniowo coraz to nowsze elementy.

W rozdziale drugim opisane zostały zastosowane technologie, charakterystyka ich działania oraz wytłumaczone zostało czym motywowany był wybór akurat ich do stworzenia aplikacji.

W rozdziale trzecim szerzej **wyjaśniłem(forma osobowa czy utrzymać bezosobową narrację?)** dokładny zakres zagadnień zawartych w aplikacji oraz uzasadniłem powód upraszczania niektórych z nich i poświęcanie większej uwagi na pozostałe.

Rozdział czwarty skupia się na technicznych aspektach budowy aplikacji, omawia szczegóły implementacji wymaganych rozwiązań i problemy z tym związane. Omówione również zostały ograniczenia zastosowanych technologii, jak i otwartość aplikacji na rozwój.

W rozdziale piątym zaprezentowane są zrzuty ekranu z działania aplikacji, wytłumaczona zostaje budowa i działanie interfejsu oraz jak spełnione zostało wstępne wymaganie aplikacji, czyli ułatwianie nauki.

2 Wykorzystane technologie

2.1 Popularne technologie webowe - HTML, CSS i TypeScript

2.2 Biblioteki Chart.js i MathJax

2.3 Język Python

Python pomimo, że znacząco wolniejszy od wielu innych języków programowania [3], posiada bogatą kolekcję bibliotek do przetwarzania danych i obliczeń matematycznych m.in. NumPy, SciPy i SymPy. Biblioteki te najczęściej napisane w C lub Fortranie, są już z kolei bardzo znacznie szybsze od czystego Pythona[4]. Charakterystyczna jest też jego przejrzysta składnia, zawierająca dużo lukru składniowego[5].

Głównymi powodami, dla których zdecydowałem się na Pythona, jest właśnie dostępność bibliotek, a szczególnie SymPy oraz wbudowana obsługa liczb zespolonych. Liczby zespolone są dostępne za pośrednictwem biblioteki Math w TypeScript'cie[?], którego i tak używam w aplikacji, jednak przez składnię kod zajmuje znacznie więcej miejsca niż w Pythonie. SymPy z kolei nie został wykorzystany w żadnym miejscu w implementacji aplikacji, więc wydać się może nietypowym argumentem za wyborem języka. SymPy jest biblioteką umożliwiającą obliczenia w matematyce symbolicznej[?], co najważniejsze pozwala na znajdowanie analitycznych rozwiązań dla równań różniczkowych. W mojej opinii ułatwia to rozwój aplikacji i wzbogacenie jej o bardziej zaawansowane przypadki rozłożenia potencjału w przestrzeni.

Dostępność Matplotlib i podobnych bibliotek umożliwia rysowanie wykresów do wizualizacji danych, wymagając do tego bardzo małej ilości kodu, co może okazać się niezwykle przydatne przy sprawdzaniu poprawności implementacji obliczeń bardziej złożonych wzorów matematycznych. Przy pisaniu pracy niejednokrotnie z takiej możliwości skorzystałem, na co najczęściej potrzebne były 3-4 linie kodu.

2.4 Framework Tauri

Była opcja WebAssembly ale jest niedoropacowane(napisać szczegóły) + vite + npm

3 Podstawy teorytyczne

3.1 Zagadnienia matematyki wyższej

Niewątpliwie matematykę, a szczególnie rachunek różniczkowy, można nazwać językiem fizyki. Do opisu zawartych w pracy zagadnień fizycznych, wymagana jest pewna znajomość zagadnień matematyki wyższej.

Liczby zespolone są nadzbiorem liczb rzeczywistych, wzbogacone o jednostkę urojoną i definiowaną jako:

$$i^2 = -1$$

Dowolną liczbę zespoloną możemy więc zapisać w postaci algebraicznej, będącej sumą liczb rzeczywistej i urojonej tj. takiej będącej rzeczywistą wielokrotnością jednostki urojonej:

$$z = a + bi$$

gdzie $z \in \mathbb{C}$, $a \in \mathbb{R}$, $b \in \mathbb{R}$

Do graficznego opisu liczby zespolonej poza osią rzeczywistą, potrzebna jest dodatkowa, pionowa oś urojona. Liczba zespolona będzie punktem o współrzędnych a i b w kartezjańskim układzie współrzędnych. Moduł definiowany jako odległość liczby z od liczby 0, będącej środkiem układu współrzędnych, jest więc definiowana poprzez zastosowanie twierdzenia Pitagorasa:

$$|z| = \sqrt{a^2 + b^2}$$

gdzie: $z = a + bi$

$z \in \mathbb{C}$, $a \in \mathbb{R}$, $b \in \mathbb{R}$

Inną postacią liczby zespolonej jest postać wykładnicza, przedstawiana jako:

$$z = |z|e^{i\phi}$$

gdzie: $z \in \mathbb{C}$

e – podstawa logarytmu naturalnego

i – jednostka urojona

ϕ – kąt pomiędzy osią części rzeczywistej, a ramieniem poprowadzonym ze środka układu współrzędnych do liczby z

Szczegóły funkcji falowej ψ wytłumaczone są w dalszej części pracy, jednak na szczególną uwagę zasługuje wartość kwadratu jej modułu, określająca gęstość prawdopodobieństwa znalezienia cząstki w danym położeniu. Oznacza to, że prawdopodobieństwo znalezienia cząstki w obszarze $[x_1, x_2]$ będziemy wyznaczać jako:

$$P(X) = \int_{x_2}^{x_1} |\psi(x)|^2 dx$$

gdzie:

X – zdarzenie losowe, że cząstka znajduje się w obszarze

x_1, x_2 – początek i koniec obszaru

Równanie różniczkowe to równanie funkcyjne zawierające pochodne nieznannej funkcji. Znajdowanie rozwiązań takich równań znacząco wykracza poza wiedzę wymaganą do zrozumienia omawianych zagadnień fizycznych, więc nie będzie temu przywiązywana szczególna uwaga. Wybrany zakres zagadnień wymaga jedynie zapamiętania rozwiązań dwóch rodzajów równań:

$$f(x)'' + f(x) = 0 \iff f(x) = A\cos x + B\sin x$$

$$f(x)'' + K^2 f(x) = 0 \iff f(x) = A\cos(Kx) + B\sin(Kx)$$

gdzie:

$f(x)$ i $f(x)''$ – funkcja i jej pochodna względem zmiennej x

$K \in \mathbb{C}$

$A \in \mathbb{R}, B \in \mathbb{R}$ – stałe wyznaczone z warunków początkowych albo brzegowych

Warto zaznaczyć, że pomimo tego że zagadnienia te są niezbędne do pełnego zrozumienia mechaniki kwantowej, to nie są wymagane do zdobycia pewnej teoretycznej wiedzy na jej temat, nawet jeśli mocno nieintuicyjnej.

3.2 Falowa natura materii

Wyjaśnione przez Alberta Einsteina zjawisko fotoelektryczne ukazało, że światło zachowuje się nie tylko jak fala, ale też jak cząstką. Nośnik światła przekazywał energię porcjami – kwantami przez co wprowadzone zostało pojęcie fotonu jako cząstki światła.

Jako konsekwencja tego odkrycia Louis de Broglie zapostulował, że cząstki materii, takie jak elektron, muszą podzielać falowe zachowanie fotonów. Korzystając z pracy Einsteina określił długość tych fal, zwanych falami materii, wzorem:

$$\lambda = \frac{h}{p}$$

gdzie:

λ – długość fali materii cząstki

h – stała Plancka

p – pęd cząstki

Dzięki falom materii możemy wyjaśnić m.in. sposób rozchodzenia się elektronów np. przez siatkę dyfrakcyjną, co jest niemożliwe przy przyjęciu elektronów jako kul lub punktów w przestrzeni.

Dokładniejszego opisu tych fal dokonał Erwin Schrödinger proponując równanie funkcji falowej ψ o zespolonym zbiorze wartości.

3.3 Równanie Schrödingera

Funkcja falowa stanowi fundament zagadnień poruszanych w aplikacji. Wynika to z faktu, że jest ona niezbędna do opisu ruchu dowolnej cząstki. Erwin Schrödinger zawarł funkcję falową w równaniu nazwanym od jego nazwiska równaniem Schrödingera. Jeżeli skupimy się na odizolowanych układach fizycznych tj. takich, które nie oddziałują z otoczeniem, rozwiązać należy równanie Schrödingera niezależne od czasu o postaci:

$$-\frac{\hbar}{2m} \frac{d^2\psi(x)}{dx^2} + V(x)\psi(x) = E\psi(x)$$

gdzie:

\hbar – zredukowana stała Plancka

$\psi(x)$ – funkcja falowa w położeniu x

E – energia całkowita ciała

$V(x)$ – potencjał w położeniu x

m – masa ciała

Wartość funkcji falowej $\psi(x)$ nie posiada fizycznej interpretacji, do tego potrzebny jest kwadrat jej modułu, który określa gęstość prawdopodobieństwa znalezienia cząstki w położeniu x .

3.4 Znajdowanie równania funkcji falowej

Warunki brzegowe:

- Skończona wartość ψ
- Ciągłość ψ
- Ciągłość pochodnej $\frac{d\psi}{dx}$ (Przy skończonych potencjałach)

4 Budowa i struktura aplikacji

4.1 Struktura aplikacji

domyślne ustawienie Tauri, zasoby, pliki html, css, typescript, python

4.2 Interfejs

html + css, BEM, struktura stron

4.3 Typescript i manipulacja DOM

4.4 Obliczenia fizyczne w Pythonie

Implementacja obliczeń zagadnień cząstki swobodnej(rysunek), nieskończonej studni potencjału(rysunek) i progu potencjału, jest trywialnym zagadnieniem i nie wymaga dłuższego tłumaczenia.

Sytuacja jest już odmienna dla pozostałych zagadnień, zaczynając od skończonej studni potencjału. Istnieją w niej bowiem, tak jak w studni nieskończonej, skwantowane stany energii dla cząstki związanej, jednak wyznaczenie ich nie wyraża się prostym wzorem. Wiadomo, że $\psi(x)$ wyraża się wzorem: $\psi_1(x) = Ce^{\gamma x}$

$$\psi_2(x) = A\sin(kx) + B\cos(kx)$$

$$\psi_3(x) = De^{-\gamma x}$$

gdzie indeksy opisują obszar, którego dotyczy wzór funkcji. Stałe A , B , C i D wyznaczone są z warunków brzegowych, jednak do obliczenia γ potrzebna jest wartość k .

Ze względu na szybkość początkowo chciałem zastosować metodą Newtona, jednak eksperymentalnie stwierdziłem, że przez charakter funkcji, znalezienie miejsca zerowego tą metodą jest bardzo wolne. Znacznie lepsze efekty przyniosła metoda bisekcji i ona wykorzystywana jest w ostatecznej implementacji.

4.5 Interfejs pomiędzy TypeScriptem i Pythonem

Języki programowania tak odmienne jak TypeScript i Python uruchamiane są w zupełnie innych środowiskach. Framework Tauri daje tutaj wyjątkową możliwość obsługując TypeScriptu przy użyciu WebView[?] i Pythona wykorzystując lokalnie zainstalowany interpreter[?].

Problemem jest jednak przesyłanie danych między tymi językami, o czym nie myślałem przy ich tworzeniu. Zastosowałem więc serializację danych do formatu JSON, który jest tekstową reprezentacją obiektu w TypeScript'cie. Wyniki obliczeń fizycznych są więc serializowane(rysunek), wysyłane jako wyjście w powłoce systemowej i przekazywane przez Tauri do skryptu napisanego w TypeScript'cie(rysunek).

4.6 Wdrożenie i dystrybucja aplikacji

o budowie aplikacji, pakowanych zasobach, interpreterze pythona, platformach

4.7 Problemy, ograniczenia i możliwości rozwoju

5 Interfejs aplikacji

5.1 Ekran główny

5.2 Interaktywna wizualizacja

5.3 Transkrypcja

6 Podsumowanie i wnioski

7 Bibliografia

- [1] M.R. Wehre, H.A. Enge, J.A. Richards, *Wstęp do fizyki atomowej*, Państwowe Wydawnictwo naukowe, Warszawa 1983
- [2] David Flanagan, *JavaScript: The Definitive Guide. Master the World's Most-Used Programming Language. 7th Edition*, O'Reilly Media, 2020
- [3] Wes McKinney *Python for Data Analysis*, str. 3
- [4] Wes McKinney *Python for Data Analysis*, str. 2
- [5] Wes McKinney *Python for Data Analysis*, str. 418

Wyrażam zgodę na udostępnienie mojej pracy w czytelniach Biblioteki SGGW w tym
w Archiwum Prac Dyplomowych SGGW.

.....
(czytelny podpis autora pracy)

