AGH
Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie
Wydział Fizyki i Informatyki Stosowanej

Praca inżynierska

Olaf Schab

kierunek studiów: **informatyka stosowana**

Symulacja komputerowa zaniku sygnału luminescencyjnego w skaleniach

Opiekun: **dr inż. Grzegorz Gach**

Kraków, styczeń 2017

Oświadczam, świadomy(-a) odpowiedzialności karnej za poświadczenie nieprawdy, że niniejszą pracę dyplomową wykonałem(-am) osobiście i samodzielnie i nie korzystałem(-am) ze źródeł innych niż wymienione w pracy.

.....
(czytelny podpis)

Merytoryczna ocena pracy przez opiekuna

Merytoryczna ocena pracy przez recenzenta

Spis treści

Rozdział 1

Wstęp

Do określenia wieku próbek w archeologii i geologii wykorzystywane są skalenie z uwzględnieniem stymulowanej luminescencji. Już w latach 60- tych XX wieku powstał pomysł wykorzystania termoluminescencji, która dzięki badaniom znalazła wiele zastosowań w archeologii i naukach o Ziemi. Jednym z głównych tematów badań jest datowanie z wykorzystaniem skaleni przy użyciu ich termoluminescencji oraz atermicznym zanikiem luminescencji powodującym zaniżanie daty próbek. Atermiczny zanik związany jest z faktem zachodzenia rekombinacji zlokalizowanej, głownie poprzez zachodzące zjawisko tunelowania elektronów z pułapek elektronowych oraz centrów rekombinacyjnych.

1.1 Założenia projektu

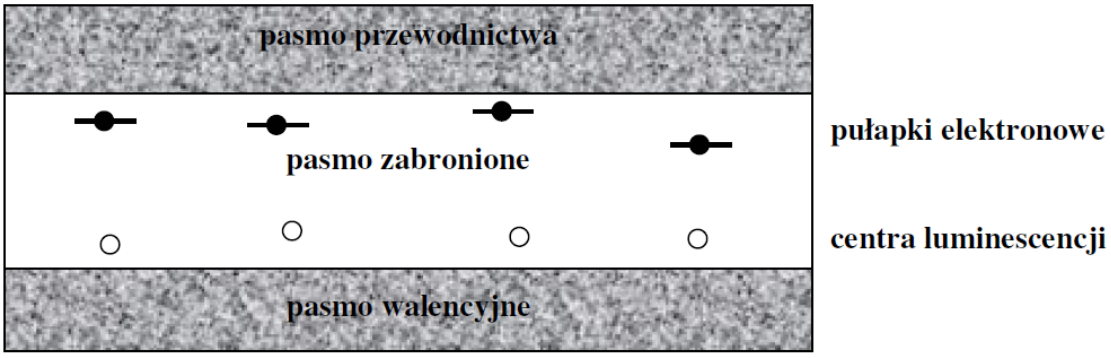
Celem projektu było stworzenie programu symulującego zanik sygnału luminescencyjnego w skaleniach. Głównym założeniem było otrzymanie wykresu obrazującego zmianę ilości elektronów znajdujących się w stanie wzbudzonym (znajdujących się w pułapkach) wraz z upływem czasu.

1.2 Zjawisko luminescencji w skaleniach

Jednym ze składników środowiska naturalnego jest promieniowanie jonizujące. Najważniejszym jego źródłem są izotopy promieniotwórcze zawarte w skorupie ziemskiej, atmosferze i biosferze, emitujące promieniowanie α, β i γ. W ostatnim okresie, bardzo krótkim w sensie geologicznym oraz historycznym, pojawiły się nowe jego źródła, związane z działalnością człowieka w zakresie zbrojeń atomowych i energetyki jądrowej. W dalszym jednak ciągu najważniejszym źródłem promieniowania jonizującego w środowisku są izotopy promieniotwórcze, które weszły w skład Ziemi w okresie formowania się układu słonecznego. Są to przede wszystkim długożyciowe izotopy uranu U^{238}, U^{235} i toru Th^{232}. Promieniowanie to niesie energię, którą pochłaniają wszystkie substancje występujące w środowisku - w tym skalenie. I to właśnie pochłanianie

promieniowania jonizującego przez te kryształy jest związane z luminescencją.

Rzeczywisty kryształ nie jest idealny. Zawiera on zawsze nieregularności i defekty struktury sieci krystalicznej. Część z nich znajdująca się w paśmie wzbronionym może mieć charakter pułapek, które są zdolne do wychwytywania elektronów z pasma przewodnictwa i przetrzymywania ich przez długi czas, do momentu, w którym otrzymają one energię niezbędną do wzbudzenia do pasma przewodnictwa. Stan kryształu, w którym część lub wszystkie pułapki są zapełnione schwytanymi elektronami, charakteryzuje się nadwyżką energii w porównaniu ze stanem podstawowym. Nadwyżka ta może zostać wyzwolona i wyemitowana np. w postaci światła poprzez dostarczenie dodatkowej energii (podgrzanie kryształu lub jego oświetlenie). Właśnie to zjawisko wykorzystywane jest w metodach datowania obiektów archeologicznych i osadów geologicznych do pomiarów dawek pochłoniętych naturalnego promieniowania jonizującego

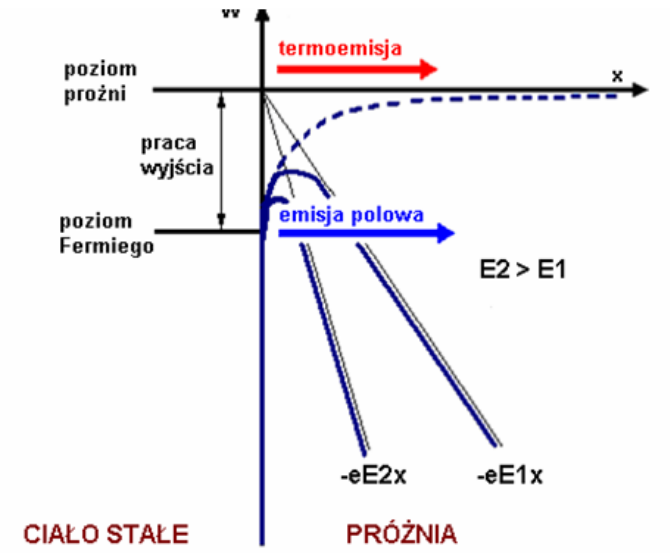


Rysunek 1.1: Struktura pasmowa skalenia.

Zdarza się jednak, że mimo iż uwięziony elektron nie otrzymał dodatkowej energii, będzie w stanie zrekombinować z dziurą. Jest to jest możliwe dzięki **zjawisku tunelowania**.

1.3 Zjawisko tunelowania (Zjawisko emisji polowej)

Zjawisko emisji polowej, zwane tunelowaniem Fowlera – Nordeheima, to proces kwantowo-mechaniczny w którym cząstka ma niezerowe prawdopodobieństwo przejścia przez barierę potencjalną nawet, gdy energia cząstki jest mniejsza od wysokości bariery potencjału.



Rysunek 1.2: Ilustracja mechanizmu emisji polowej.

Analizując powyższy wykres można stwierdzić, że im wyższa wartość przyłożonego zewnętrznego pola tym węższa staje się bariera potencjału, a prawdopodobieństwo tunelowania wzrasta.

W przypadku skaleni prawdopodobieństwo, że elektron nie przetuneluje do centrum rekombinacji wyrażone jest wzorem:

$$P = e^{\frac{-t}{\tau}} \tag{1.1}$$

$$\tau = S^{-1} e^{\alpha r} \tag{1.2}$$

gdzie:

- t - czas
- S - stała częstotliwość prób ucieczki
- α - stała zależna od różnicy energii pułapki i dziury
- r - odległość do przetunelowania

Rozdział 2

Wybór stosowanych technologii

2.1 Język C++11 oraz wybór kompilatora

Aplikacja została napisana z użyciem języka C++11. Język ten wybrano z kilku powodów. Skorzystano tu z jego natury jako języka zorientowanego obiektowo. Dzięki temu podczas wykonania programu stworzono klasy odpowiadające obiektom w świecie realnym - elektrony, pułapki elektronowe itp. Kolejnym powodem dla którego wykorzystano język C++ jest jego wydajność. Kod napisany w języku C++ jest kompilowany bezpośrednio i w pełni do kodu maszynowego wykonywanego przez procesor komputera. Dzięki takiemu otrzymujemy maksymalną możliwą wydajność inaczej niż gdy używamy innych obiektowych języków programowania wysokiego poziomu, które są kompilowanye do kodu zarządzanego (tzw. kodu bajtowego na przykład przypadku Javy) gdzie dodatkowy narzut generowany jest przez warstwę pośrednią w postaci maszyny wirtualnej.

Do kompilacji użyto darmowego kompilatora GCC (Gnu Compiler Collecion), konkretnie jego implementacji dla platformy Windows - MinGW w wersji 3.2.2 (Minimalist GNU for Windows).

2.2 Środowisko programistyczne

Program był tworzony z użyciem wieloplatformowego zintegrowanego środowiska programistycznego języków C/C++ - **CLion** (wersja 2016.2.3) produkcji spółki JetBrains. Było to podyktowane głównie znajomością produktów firmy JetBrains. CLion jest produktem komercyjnym, jednak skorzystano tu z darmowej licencji studenckiej. IDE produkcji JetBrains obsługuje również system zarządzania kompilacją **CMake**, którego główną cechą jest niezależność od używanego kompilatora oraz platformy sprzętowej - CMake nie kompiluje programu samodzielnie, lecz tworzy pliki z regułami kompilacji dla konkretnego środowiska.

2.3 System kontroli wersji Git

W procesie tworzenia aplikacji wykorzystano system kontroli wersji Git oraz darmowy hostingowy serwis internetowy GitHub. Głównym powodem wykorzystania tego systemu była jego popularność oraz prostota użytkowania. System kontroli wersji okazał się być niezwykle użytecznym narzędziem pozwalającym na śledzenie zmian w kodzie, wprowadzanie testowych rozwiązań bez ryzyka zniszczenia kodu. Cały projekt można pobrać ze strony:

`https://github.com/Sharkuu/Dissertation`

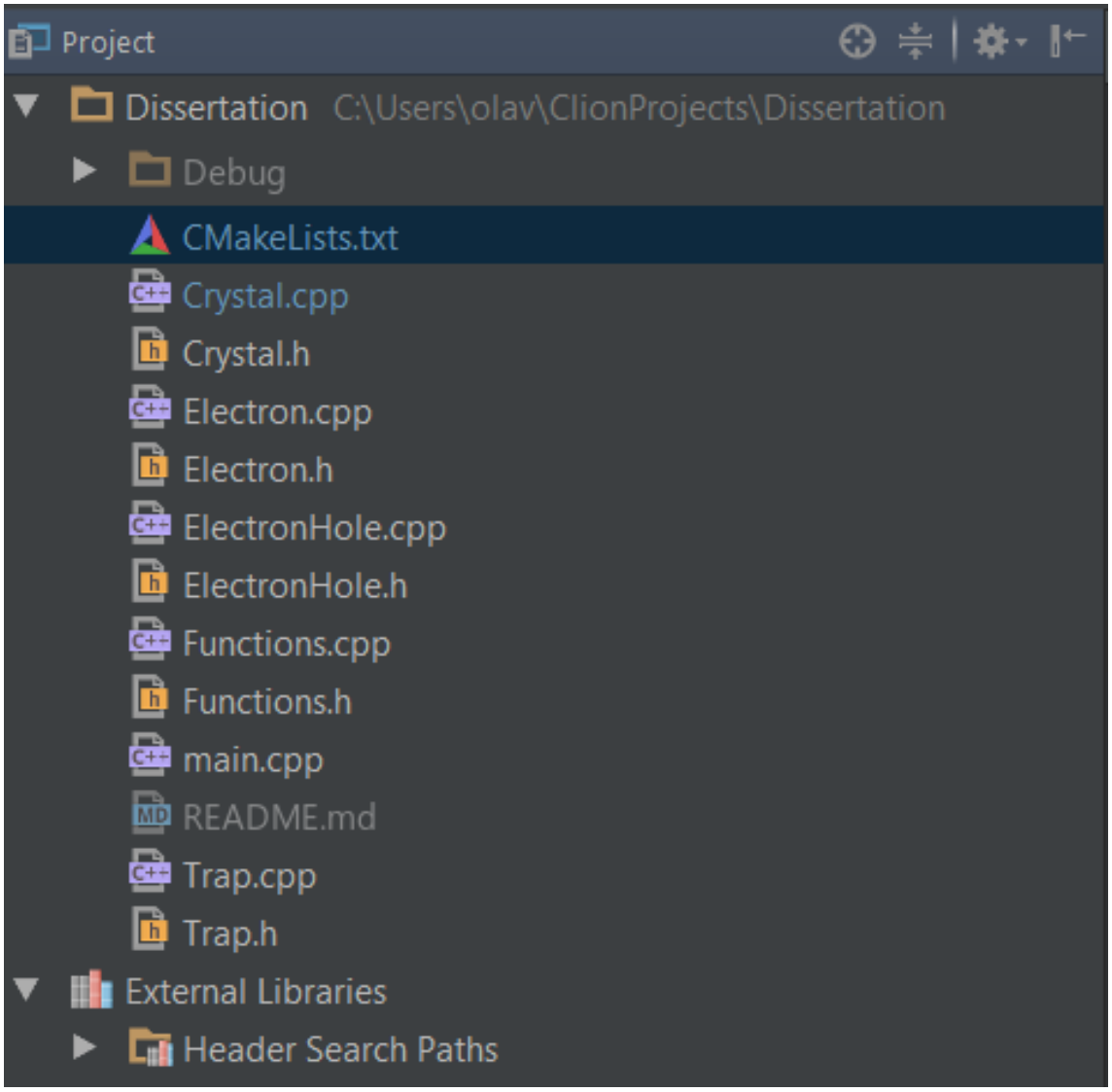
2.4 Wizualizacja wyników - Gnuplot

Po wygenerowaniu danych, aby z wizualizować wykres obrazujący zmianę ilości elektronów znajdujących się w stanie wzbudzonym (znajdujących się w pułapkach) wraz z upływem czasu skorzystano z programu **Gnuplot**.

Rozdział 3

Implementacja

Głównym założeniem projektu było stworzenie go w taki sposób, aby kod jak najtrafniej odzwierciedlał rzeczywistość. W tym celu stworzono klasy obiektów reprezentujące rzeczywiste byty w świecie realnym.



Rysunek 3.1: Struktura projektu.

3.1 Klasy

Każda z klas posiada własne metody w zależności od swojego przeznaczenia.

Klasa *Electron* odzwierciedla cząstkę elementarną - elektron. Zmiana ilości tych cząstek w stanie wzbudzonym tj. znajdujących się w pułapce jest głównym celem wykonania tej symulacji.

ElectronHole jest reprezentacją dziury elektronowej, z którą to elektron po wykorzystaniu zjawiska tunelowego zrekombinuje.

Klasa *Trap* odpowiada defektom w sieci krystalicznej czyli pułapkom, które mogą przechwycić elektron lub dziurę elektronową. Na potrzeby wykonania tej symulacji założono, że na samym jej początku wszystkie obiekty reprezentujące elektrony oraz dziury elektronowe znajdują się już w pułapkach. Dodatkowo, w symulacji ustalono, że dany elektron jeśli spełni warunek wystąpienia zjawiska tunelowego - po jego zajściu i rekombinacji z dziurą - nie może przetunelować ponownie. Zostało to podyktowane zmniejszeniem oczekiwanego czasu działania programu.

Klasa *Crystal* reprezentuje rzeczywisty kryształ, który będzie poddany symulacji zaniku sygnału luminescencyjnego. Zawiera on w sobie inne obiekty takie jak:

- Elektrony
- Dziury elektronowe
- Obiekty odpowiadające defektom sieci krystalicznej - tzw. pułapki

3.2 Opis działania programu

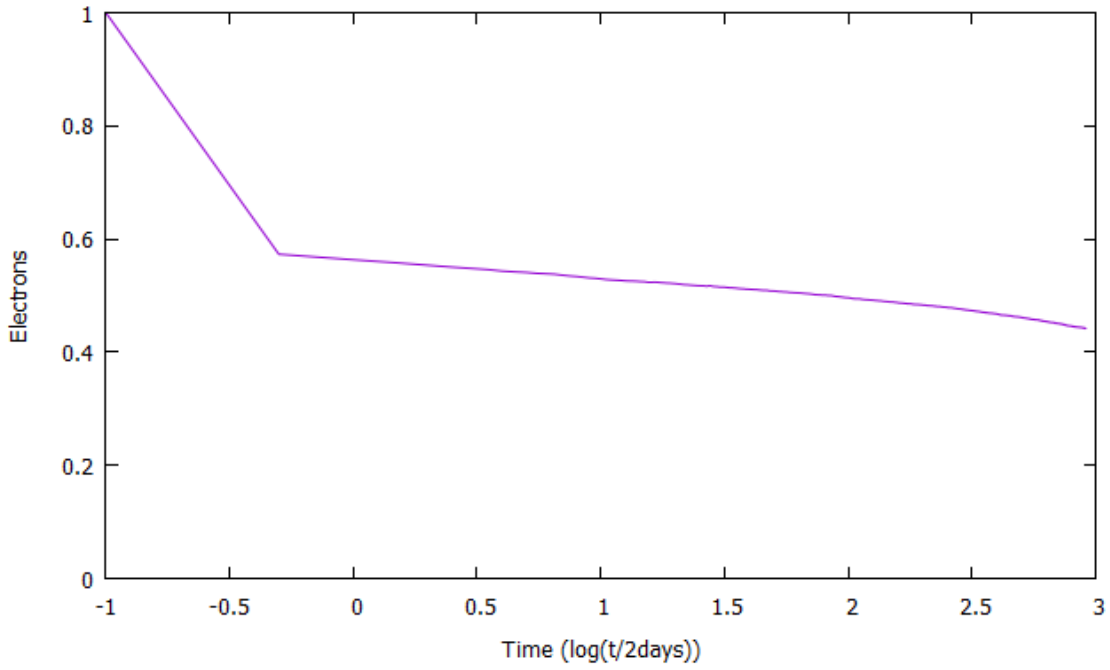
W celu otrzymania danych niezbędnych do wygenerowania wykresu zależności między ilością elektronów w pułapkach a upływem czasu uruchomiono stworzony program. Obiekt *Crystal* w swoim konstruktorze generuje podaną ilość dziur elektronowych, pułapek oraz elektronów, a następnie przechowuje je w oddzielnych kontenerach. Dla każdej z tych cząstek wywoływany jest konstruktor (ustawiający współrzędne położenia) z argumentami, które są losowane z podanego wcześniej przedziału (wartości są podane w Angstremach tj. 1 Å $= 10^{-10}$m. Jednostka ta nie jest jednostką układu SI, lecz jest stosowana w fizyce przy opisywaniu obiektów i zjawisk zachodzących w skali atomowej). Jak podkreślono wcześniej, symulacja zakłada, że na jej starcie każdy elektron znajduje się w pułapce. Powoduje to, że para elektron - pułapka ma identyczne współrzędne położenia, a także obiekt klasy *Trap* przyjmuje wskaźnik na uwięziony w nim elektron.

Następnie za pomocą metody *startSimulation(int time)* obiektu klasy *Crystal* rozpoczynana jest symulacja. Symulowany upływ czasu zależy od wartości argumentu *time*, który jest wyrażony w dniach tj. wywołanie *startSimulation(365)* oznacza rozpoczęcie działania symulacji symulującej efekt zaniku sygnału luminescencyjnego w czasie 1 roku.

Podczas wykonywania symulacji każda pułapka elektronowa jest sprawdzana czy przechowuje w sobie uwięziony elektron. Jeśli warunek jest spełniony, to dla elektronu oraz dziur elektronowych znajdujących się w pułapkach ,za pomocą wzoru (1.1) sprawdzane jest prawdopodobieństwo zajścia efektu tunelowego. Jeśli ma ono miejsce elektron opuszcza swoją pułapkę (wskaźnik obiektu klasy *Trap* na elektron ustawiany jest na wartość *NULL*), a następnie zmienia swoje dotychczasowe położenie na współrzędne do których przetunelował (współrzędne centrum rekombinacyjnego/dziury). Powtarzane jest to tak długo, aż program za symuluje podany mu wcześniej czas trwania symulacji. Jak wynika z założeń projektu wspomnianych wcześniej - jeśli elektron przetunelował do centrum rekombinacji przynajmniej raz, nie będzie on brał więcej udziału w obliczaniu prawdopodobieństwa zajścia efektu tunelowego.

Po skończeniu wykonywania symulacji, program zapisuje do pliku otrzymane wyniki w formacie *czas;ilość elektronów w stanie wzbudzenia*, gdzie czas wyrażony jest w $\log(\frac{t}{2dni})$, a ilość elektronów jest kwantowana do 1. Za pomocą skryptu *wykres.plt* w folderze bin/Debug generowany jest odpowiedni wykres ilustrujący otrzymane dane.

Przykładowo wygenerowany wykres (analiza otrzymanych wykresów znajduje się w dalszej części pracy):



Rysunek 3.2: Przykładowa wizualizacja otrzymanych danych

Rozdział 4

Analiza wyników

Program był tworzony, testowany oraz wykonywany na komputerze o podanej specjalizacji:

- Procesor: Intel Core i7-6700HQ (4 rdzenie, od 2.60 GHz do 3.50 GHz, 6 MB cache)
- Pamięć: 8 GB (SO-DIMM DDR4, 2133MHz)

4.1 Wygenerowane wykresy

W celu analizy otrzymanych danych, program został uruchomiony pewną ilość razy, za każdym razem z innymi danymi wejściowymi. Starano się tak dopasować ich wartości, aby gęstość rozkładu cząstek była stała.

———————

TU WYKRESY Z PODPISAMI JAKIE BYŁY ARGUENTY, CZAS WYKONANIA ,ILE PAMIECI UZYWALO(?)

————

4.2 Analiza wykresów

Analizując otrzymane wykresy można zauważyć znaczny spadek ilości elektronów w pułapkach na samym początku wykonywania programu. Jest to spowodowane tym, że wszystkie współrzędne cząstek są losowane z podanego zakresu. Oznacza to, że na początku symulacji wiele pułapek elektronowych zawierających elektrony znajduje się bardzo blisko centrów rekombinacji. Jak widać w równaniach (1.1) oraz (1.2) odległość między nimi znacząco wpływa na prawdopodobieństwo zajścia efektu tunelowego, stąd też wynika zaobserwowany znaczący spadek ilości wzbudzonych elektronów.

Rozdział 5

Wnioski