

## IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

## KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

## TWÓJ KOSZYK

DODAJ DO KOSZYKA

## CENNIK I INFORMACJE

ZAMÓW INFORMACJE  
O NOWOŚCIACH

ZAMÓW CENNIK

## CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

# Symulacje komputerowe w fizyce

Autor: Maciej Matyka  
ISBN: 83-7197-713-1  
Format: B5, stron: 194  
Zawiera CD-ROM



Książka prezentuje algorytmy umożliwiające komputerowe symulowanie rozmaitych procesów fizycznych. Po przedstawieniu metody różnicowej rozwiązywania zwyczajnych równań różniczkowych, Autor pokazuje, w jaki sposób zastosować ją przy rozwiązywaniu konkretnych problemów fizycznych.

Opisano między innymi:

- Symulowanie zjawisk mechaniki klasycznej w oparciu o prawa dynamiki Newtona
- Rozwiązanie numeryczne równania falowego
- Symulacje dynamiki cieczy nieściśliwej
- Rozwiązanie numeryczne równania Schrödingera

„Symulacje komputerowe w fizyce” to książka, którą powinni zainteresować się nie tylko fizycy: niektóre przedstawione tu algorytmy znajdują zastosowanie w aplikacjach CAD/CAM, a nawet przy tworzeniu gier komputerowych.



# Spis treści

|                                                                                             |           |
|---------------------------------------------------------------------------------------------|-----------|
| <b>Wstęp .....</b>                                                                          | <b>7</b>  |
| <b>Rozdział 1. Schematy różnicowe rozwiązywania równań różniczkowych zwyczajnych..</b>      | <b>11</b> |
| 1.1. Równania różniczkowe zwyczajne i różnice skończone.....                                | 12        |
| 1.2. Równania różniczkowe zwyczajne i rachunek całkowy .....                                | 13        |
| 1.2.1. Schemat różnicowy Eulera.....                                                        | 14        |
| 1.2.2. Rozwiązanie równania rozpadu promieniotwórczego.....                                 | 16        |
| 1.2.3. Metoda skokowa z wstępnymi obliczeniami Eulera .....                                 | 19        |
| 1.2.4. Wahadło matematyczne .....                                                           | 24        |
| 1.2.5. Punkt materialny przymocowany do sprężyny .....                                      | 32        |
| 1.3. Dokładniejsze metody wyznaczania rozwiązań równań różniczkowych .....                  | 38        |
| 1.3.1. Metoda punktu środkowego (ang. MidPoint) drugiego rzędu .....                        | 38        |
| 1.3.2. Metoda Rungego-Kutty czwartego rzędu.....                                            | 41        |
| 1.4. Zestawienie poznanych schematów rozwiązywania<br>równań różniczkowych zwyczajnych..... | 43        |
| 1.5. Podsumowanie .....                                                                     | 43        |
| <b>Rozdział 2. Dynamika według sir Isaaca Newtona.....</b>                                  | <b>45</b> |
| 2.1. Rachunek wektorowy .....                                                               | 45        |
| 2.1.1. Klasa Wektor .....                                                                   | 46        |
| 2.1.2. Operacje na wektorach .....                                                          | 47        |
| 2.1.3. Rachunek wektorowy — podsumowanie.....                                               | 52        |
| 2.2. Zasady dynamiki Newtona .....                                                          | 53        |
| 2.2.1. Pierwsza zasada dynamiki Newtona .....                                               | 53        |
| 2.2.2. Druga zasada dynamiki Newtona .....                                                  | 54        |
| 2.2.3. Trzecia zasada dynamiki Newtona.....                                                 | 54        |
| 2.3. Model fizyczny dynamiki układów punktów materialnych .....                             | 54        |
| 2.3.1. Elementy składowe modelu.....                                                        | 55        |
| 2.4. Punkt materialny .....                                                                 | 58        |
| 2.4.1. Przechowywanie danych. Lista jednokierunkowa .....                                   | 58        |
| 2.4.2. Równania ruchu pojedynczego punktu materialnego .....                                | 61        |
| 2.5. Kolizje .....                                                                          | 65        |
| 2.5.1. Prosta metoda wykrywania kolizji punkt-ściana.....                                   | 65        |
| 2.5.2. Nieruchoma sfera kolizji .....                                                       | 67        |
| 2.6. Oddziaływania między punktami materialnymi .....                                       | 75        |
| 2.6.1. Prawo powszechnego ciążenia .....                                                    | 75        |
| 2.6.2. Oddziaływanie sprężyste pary punktów.....                                            | 77        |
| 2.7. Konstruowanie obiektów złożonych.....                                                  | 81        |
| 2.7.1. Model dwuwymiarowego sznura .....                                                    | 81        |
| 2.7.2. Symulacja trójwymiarowych tkanin.....                                                | 84        |

|                                                                                                         |            |
|---------------------------------------------------------------------------------------------------------|------------|
| 2.7.3. Konstrukcja bryły sztywnej.....                                                                  | 85         |
| 2.7.4. Konstrukcja modelu poruszającej się postaci.....                                                 | 87         |
| 2.8. Podsumowanie .....                                                                                 | 90         |
| <b>Rozdział 3. Rozwiązanie numeryczne równania falowego .....</b>                                       | <b>91</b>  |
| 3.1. Co to jest fala?.....                                                                              | 91         |
| 3.2. Klasyczne równanie falowe .....                                                                    | 92         |
| 3.3. Równanie falowe w jednym wymiarze .....                                                            | 92         |
| 3.3.1. Podział równania falowego na układ dwóch sprzężonych równań różniczkowych pierwszego rzędu ..... | 93         |
| 3.3.2. Siatka różnicowa Eulera w jednym wymiarze .....                                                  | 94         |
| 3.3.3. Rozwiązanie algorytmiczne układu równań sprzężonych.....                                         | 94         |
| 3.3.4. Algorytm programu realizującego równanie falowe 1D .....                                         | 96         |
| 3.3.5. Efekty działania przedstawionego algorytmu .....                                                 | 101        |
| 3.4. Równanie falowe w dwóch i więcej wymiarach przestrzennych .....                                    | 106        |
| 3.4.1. Siatka różnicowa Eulera w dwóch wymiarach.....                                                   | 106        |
| 3.4.2. Realizacja symulacji równania falowego w dwóch wymiarach.....                                    | 110        |
| 3.5. Podsumowanie .....                                                                                 | 113        |
| <b>Rozdział 4. Symulacje cieczy nieściśliwej .....</b>                                                  | <b>117</b> |
| 4.1. Równanie Naviera-Stokesa dla cieczy nieściśliwej .....                                             | 117        |
| 4.1.1. Warunek nieściśliwości cieczy.....                                                               | 118        |
| 4.1.2. Pola wektorowe .....                                                                             | 119        |
| 4.1.3. Analiza równania Naviera-Stokesa .....                                                           | 121        |
| 4.2. Rozwiązanie uproszczone równań NS.....                                                             | 124        |
| 4.2.1. Równanie płytkiej wody.....                                                                      | 124        |
| 4.2.2. Warunek zachowania masy .....                                                                    | 125        |
| 4.2.3. Końcowa postać równania dla płytkiej wody.....                                                   | 126        |
| 4.2.4. Przybliżenie dyskretne .....                                                                     | 126        |
| 4.2.5. Efekty działania.....                                                                            | 130        |
| 4.3. Pełne rozwiązanie równań NS dla cieczy nieściśliwej.....                                           | 132        |
| 4.3.1. Reprezentacja cieczy .....                                                                       | 133        |
| 4.3.2. Schematy różnicowe dla równania NS.....                                                          | 139        |
| 4.3.3. Warunki brzegowe .....                                                                           | 148        |
| 4.3.4. Algorytm programu .....                                                                          | 152        |
| 4.3.5. Wizualizacja rezultatów obliczeń.....                                                            | 165        |
| 4.4. Podsumowanie .....                                                                                 | 169        |
| <b>Rozdział 5. Równanie Schrödingera .....</b>                                                          | <b>171</b> |
| 5.1. Funkcja falowa — wektor stanu układu kwantowego .....                                              | 171        |
| 5.2. Ewolucja w czasie stanu układu kwantowego .....                                                    | 172        |
| 5.3. Dyskretna postać operatora ewolucji w czasie.....                                                  | 173        |
| 5.4. Schemat rozwiązania różnicowego .....                                                              | 174        |
| 5.5. Stan początkowy układu .....                                                                       | 175        |
| 5.6. Implementacja .....                                                                                | 175        |
| 5.6.1. Algorytm programu .....                                                                          | 176        |
| 5.6.2. Konstrukcja stanu początkowego .....                                                             | 176        |
| 5.6.3. Pętla obliczeniowa.....                                                                          | 178        |
| 5.7. Rezultaty .....                                                                                    | 180        |
| 5.8. Podsumowanie .....                                                                                 | 181        |
| <b>Bibliografia .....</b>                                                                               | <b>183</b> |
| <b>Skorowidz.....</b>                                                                                   | <b>189</b> |

## Rozdział 5.

# Równanie Schrödingera

Mechanika kwantowa jest bardzo zaawansowaną dziedziną fizyki współczesnej. Jej zrozumienie jest zadaniem na lata i czytelnik raczej nie znajdzie tu systematycznego wykładu z tej dziedziny. Postaramy się natomiast przedstawić kompletne rozwiązanie numeryczne równania Schrödingera<sup>1</sup>, którego interpretację fizyczną i analizę znajdzie czytelnik w wykładach z mechaniki kwantowej w pracach [18], [19], [20] czy [21]. W literaturze tematu, oprócz świetnej pozycji [19], raczej trudno znaleźć ciekawe wizualizacje efektów kwantowych. Rozwiązania zagadnień mechaniki kwantowej są trudne do wyobrażenia i sprzeczne z naszą intuicją. Zazwyczaj autorzy pozycji traktujących o MK poprzestają na ogromnej ilości wzorów, których piękno samo w sobie dostrzec można dopiero po latach wytężonej pracy. W tym rozdziale postaramy się przedstawić graficznie rozwiązania numeryczne równania Schrödingera w jednym i dwóch wymiarach przestrzennych. Jak się okaże, numeryczne rozwiązywanie zagadnień mechaniki kwantowej nie jest niczym strasznym, a rezultaty są niesamowite i piękne. Wizualizacja rozwiązań będzie analogiczna do wizualizacji rozwiązań z rozdziału trzeciego; wszak rozwiązaniem naszym jest też fala...

## 5.1. Funkcja falowa — wektor stanu układu kwantowego

Aby rozpocząć rozważania na temat rozwiązania numerycznego równania Schrödingera, musimy poznać niezbędne minimum wiedzy na jego temat, czyli określić, czego tak naprawdę szukamy. Jakich rozwiązań mamy się spodziewać? Jaką interpretację fizyczną mają rozwiązania tego równania?

Odpowiedź na te pytania jest w zasadzie prosta, lecz interpretacja jej rozwiązań — już nie. Zagadnienie funkcji falowej w mechanice kwantowej ma bezpośredni związek z pojęciem prawdopodobieństwa<sup>2</sup>. Weźmy pojedynczą cząstkę — elektron. W przypadku

---

<sup>1</sup> Na płycie CD została użyta pisownia nazwiska w formie Schroedinger.

<sup>2</sup> Zakładamy, że pojęcie prawdopodobieństwa nie jest czytelnikowi obce.

klasycznym opisujemy go, podając jego położenie i pęd. W przypadku kwantowym — stan elektronu opisuje funkcja falowa zwana też często wektorem stanu. W takim razie nawet najprostszy układ, jakim jest pojedynczy elektron, musi mieć odpowiadającą mu funkcję falową  $\Psi$ . Skoro tak, to jaka jest funkcja falowa poruszającego się elektronu? Czy jest to po prostu punkt w płaszczyźnie dwuwymiarowej, który porusza się w określonym kierunku<sup>3</sup>?

I tu pojawia się problem zwany nieoznaczonością Heisenberga. Niestety — nawet dla pojedynczej cząstki kwantowej nie możemy jednoznacznie określić jednocześnie jej położenia i pędu. W takim razie funkcja falowa (zwana też wektorem w przestrzeni Hilberta stanu układu kwantowego) musi być „rozmyta” w przestrzeni. Funkcja falowa określa nam prawdopodobieństwo znalezienia elektronu w danym miejscu przestrzeni. Rozwiązując równanie Schrödingera zależne od czasu, dla każdego kroku czasowego otrzymamy rozkład prawdopodobieństwa znalezienia cząsteczki w konkretnym miejscu w przestrzeni. Doskonale widoczne będzie też rozmywanie w czasie stanu układu, tak że wreszcie wskazanie „gdzie znajduje się elektron w danym momencie” nie będzie możliwe.

## 5.2. Ewolucja w czasie stanu układu kwantowego

Chcielibyśmy wprowadzić kilka standardowych pojęć występujących w mechanice kwantowej. Mamy nadzieję, że czytelnik nie zrazi się do tego, bo w gruncie rzeczy zostanie poprowadzony za rękę i nie powinien mieć problemów ze zrozumieniem tekstu. Ważna jest znajomość pewnych podstaw matematycznych (równanie cząstkowe, działania na liczbach zespolonych). Warto też, aby czytelnik zajrzał do wspomnianej we wstępie literatury tematu.

Interesującym nas zagadnieniem jest ewolucja stanu układu kwantowego. Ewolucja w czasie opisywana jest przez równanie różniczkowe cząstkowe drugiego rzędu zwane równaniem Schrödingera z czasem:

$$i\hbar \frac{\partial \Psi(\vec{r}, t)}{\partial t} = H\Psi(\vec{r}, t) \quad (5.1)$$

gdzie odpowiednio:

$$\hbar = \frac{h}{2\pi} \quad \text{jest stałą Plancka dzieloną przez } 2\pi,$$

$$\Psi(\vec{r}, t) \quad \text{jest wektorem (przestrzeni Hilberta) stanu układu kwantowego,}$$

<sup>3</sup> Cały czas mówić będziemy o reprezentacji położeniowej funkcji falowej; z innymi reprezentacjami stanu układów kwantowych czytelnik może zapoznać się we wspomnianej literaturze.

$H$  jest hamiltonianem kwantowym układu (operatorem Hamiltona),  
 $i$  oznacza część urojoną liczby zespolonej (tu — część urojoną równania).

Naszym zadaniem jest znaleźć ewolucję wektora stanu  $\Psi(\vec{r}, t)$ . Aby to zrobić, należy przede wszystkim rozwinąć hamiltonian do postaci jawnej. Dla swobodnej paczki falowej reprezentującej np. swobodny elektron hamiltonian zapisujemy w postaci:

$$H = -\frac{\hbar^2}{2m}\nabla^2 + V(\vec{r}) \quad (5.2)$$

gdzie

$$-\frac{\hbar^2}{2m}\nabla^2$$

jest kwantowym operatorem pędu, a  $V(\vec{r})$  określa niezależny od czasu potencjał. Tak zdefiniowany hamiltonian, po podstawieniu do równania (5.1) i wprowadzeniu specjalnych jednostek, dla których  $\hbar = 1$  i  $m=1/2$ , sprowadza równanie (5.1) do postaci<sup>4</sup>:

$$i\frac{\partial\Psi(\vec{r},t)}{\partial t} = (-\nabla^2 + V(\vec{r}))\Psi(\vec{r},t) \quad (5.3)$$

Dla wprowadzonych jednostek ogólna postać rozwiązania równania (5.3) jest znana:

$$\Psi(\vec{r},t) = e^{-iHt}\Psi(\vec{r},0) \quad (5.4)$$

gdzie wyraz  $e^{-iHt}$  nazywa się operatorem ewolucji czasowej, a wektor  $\Psi(\vec{r},0)$  określa stan początkowy układu w chwili  $t=0$ . Oznacza to, że dla zadanego stanu początkowego możemy wyznaczyć stan układu w dowolnej chwili  $t>0$ , korzystając z operatora ewolucji czasowej.

## 5.3. Dyskretna postać operatora ewolucji w czasie

Zadanie nasze w praktyce sprowadzi się do przybliżenia różnicowego operatora ewolucji czasowej oraz rozwiązania tak zadanego równania różniczkowego. Dyskretna postać tego operatora nadająca się do rozwiązania różnicowego nosi nazwę postaci Caley'a:

$$e^{-iHt} \approx \frac{1 - iH\Delta t/2}{1 + iH\Delta t/2} \quad (5.5)$$

<sup>4</sup> Patrz pozycja [9], strona 851. Postępujemy tak jedynie dla uproszczenia równania Schrödingera, które można byłoby rozwiązywać wraz z wszystkimi stałymi. Podejście takie stosuje się często w celu uproszczenia formy zapisu równań opisujących zjawiska fizyczne.

Jest to przybliżenie funkcji  $\exp$  do wyrazu  $\Delta t^2$  włącznie. Tak zdefiniowany operator różnicowy jest unitarny, czyli zachowuje normę<sup>5</sup> funkcji falowej. Jak przedstawiono w pracy [18], operator ewolucji czasowej po podstawieniu hamiltonianu rozbić można na dwa wyrazy<sup>6</sup>:

$$e^{-iH\Delta t} = e^{-i(-\nabla^2 + V)\Delta t} = e^{-i(-\nabla^2)\Delta t} e^{-iV\Delta t}$$

z których pierwszy wyraz zostanie przybliżony schematem (5.5). W takim układzie otrzymujemy kompletną reprezentację różnicową operatora ewolucji czasowej, którą zastosujemy w równaniu (5.4):

$$\Psi(\vec{r}, t) = \frac{1 + i(\Delta t/2)\nabla^2}{1 - i(\Delta t/2)\nabla^2} e^{-iV\Delta t} \Psi(\vec{r}, 0) \quad (5.6)$$

Postać tę wykorzystamy bezpośrednio w tej formie w rozwiązaniu różnicowym równania Schrödingera.

## 5.4. Schemat rozwiązania różnicowego

Równanie (5.6) rozwiążemy metodą iteracyjną Jacobiego, zaczynając od postaci<sup>7</sup>:

$$\Psi(\vec{r}, t + \Delta t) = \left( \frac{2}{1 - i(\Delta t/2)\nabla^2} - 1 \right) \cdot e^{-iV\Delta t} \Psi(\vec{r}, t) \quad (5.7)$$

Równanie to jest tylko inaczej zapisanym (5.6) — wprowadźmy dla wygody nowe oznaczenie:

$$\Psi^? = \frac{2 \cdot e^{-iV\Delta t} \Psi(\vec{r}, t)}{1 - i(\Delta t/2)\nabla^2} \quad (5.8)$$

Równanie (5.7) przyjmie teraz bardziej zwartą formę:

$$\Psi(\vec{r}, t + \Delta t) = \Psi^? - e^{-iV\Delta t} \cdot \Psi(\vec{r}, t) \quad (5.9)$$

<sup>5</sup> O normie funkcji falowej powiemy więcej przy interpretacji otrzymanych rozwiązań, na razie czytelnik powinien wiedzieć, że zachowanie normy funkcji falowej jest niezbędne do otrzymania dobrych (fizycznych) rozwiązań.

<sup>6</sup> We wzorach różnicowych nie będziemy już wpisywać jawnie zależności funkcji od zmiennych.

<sup>7</sup> Metodę tę zaproponował dr G. Jastrzębski w pozycji [18].

Naszym zadaniem jest rozwiązanie zagadnienia dla nieznanej funkcji  $\Psi^?$  zdefiniowanej wzorem (5.8). Bezpośrednio z (5.8) rozwiązanie na funkcję  $\Psi^?$  zapisujemy wzorem iteracyjnym<sup>8</sup>:

$$\Psi_{n+1}^? = 2 \cdot e^{-iV\Delta t} \Psi(\vec{r}, t) + i \cdot \Delta t \cdot \nabla^2 \Psi_n^? \quad (5.10)$$

gdzie operator  $\nabla^2$  przybliżamy odpowiednio wzorem pięcio- lub dziewięciopunktowym, w zależności od rozmiarów siatki w poziomie i w pionie<sup>9</sup>. Indeks dolny  $n/n+1$  oznacza w tym kontekście przynależność do kroku iteracyjnego. Za warunek startowy iteracji przyjmujemy<sup>10</sup>  $\Psi_0^? = 0$ . Ilość iteracji ma bezpośredni wpływ z jednej strony na czas rozwiązania, z drugiej — na jego dokładność.

## 5.5. Stan początkowy układu

W prezentowanym programie jako warunek początkowy wybraliśmy zgodnie z sugestią w [4] dobrze określony fizycznie stan gaussowski unormowany, który zapisać możemy wzorem:

$$\Psi(x, y, t) = N e^{-(x-px)^2 - (y-py)^2 - ikx} \quad (5.11)$$

gdzie  $N$  jest stałą normalizacyjną dla tego stanu. Tak zapisany stan układu wyznacza gaussian o wektorze falowym  $k$ .

## 5.6. Implementacja

Teoretyczne wyprowadzenie schematu rozwiązania równania (5.1) było niezbędne i choć w tym momencie może wydawać się dosyć skomplikowane, to mając tak wprowadzoną teorię, sprowadzamy implementację do zastosowania kilku podstawowych zasad. Chodzi głównie o operacje na liczbach zespolonych oraz zastosowanie siatki różnicowej do reprezentacji stanu układu kwantowego. Siatkę różnicową potrafimy już dobrze określić (rozdziały 3., 4.). W tym przypadku na siatce różnicowej umieszczamy w centrum wartości funkcji falowej  $\Psi$  określającej stan układu kwantowego. Różnicą w stosunku do przedstawionych w poprzednich rozdziałach siatek różnicowych będzie to, iż siatka w naszym przypadku będzie zespolona. Do stworzenia siatki użyjemy konstrukcji `complex` z biblioteki standardowej<sup>11</sup> C++. Deklaracja siatki różnicowej o danych zespolonych, wielkości  $NX$  na  $NY$ , reprezentującej funkcję falową `psi`, przyjmie następującą postać:

```
complex<double> psi[NX][NY];
```

<sup>8</sup> Patrz pozycje [18], [11].

<sup>9</sup> Zagadnienia przybliżania operatora Laplace'a na siatkach różnicowych były już dyskutowane w rozdziale 3.

<sup>10</sup> Taki start iteracji zaproponowany został w pozycji [11].

<sup>11</sup> Użycie gotowego szablonu dla danych zespolonych bardzo uprości kod programu.



Zapisany w nawiasach trójkątnych typ danych określa, jak zapisane będą odpowiednio część rzeczywista i część urojona liczby zespolonej. Użycie standardowego typu danych pozwoli nam korzystać z własności liczb zespolonych bez wprowadzania dodatkowych definicji metod i operatorów<sup>12</sup>.

Wizualizacja danych przebiegać będzie analogicznie do tej znanej już z równania falowego. Dla każdego kroku czasowego<sup>13</sup> rysujemy rozkład funkcji  $\Psi$  wychyleń w trzech wymiarach przestrzennych. Zagadnieniem wizualizacji nie będziemy się więc bliżej zajmować, bo było to już omawiane w rozdziale 3.

### 5.6.1. Algorytm programu

Dalej w znanej już formie diagramu przedstawiamy na rysunku 5.1 propozycję schematu ogólnego działania programu rozwiązującego numerycznie równanie Schrödingera. Algorytm nie jest zbyt złożony i jak się okaże, przełożenie kolejnych punktów na kod programu nie stanowi większego problemu, szczególnie jeśli wykorzystujemy dobrze napisaną klasę liczb zespolonych (a taką niewątpliwie jest klasa z biblioteki standardowej C++).

W przedstawionym algorytmie zapisaliśmy w nawiasach równania, z których będziemy korzystać przy kolejnych częściach kroku czasowego. Jak widać, zagadnienie sprowadza się do zastosowania tylko trzech wzorów, które wyprowadziliśmy wcześniej.

### 5.6.2. Konstrukcja stanu początkowego

Zgodnie z wzorem (5.11) konstruujemy stan początkowy, wpisując odpowiednie wartości funkcji  $\Psi$  w siatkę różnicową. Niech stan początkowy wyznaczy paczka falowa o szerokości (double) SPaczk, w pozycji (int) px, py na siatce. Korzystając z tego, że istnieje w bibliotece standardowej funkcja `exp(class complex<T>)`, wyliczenie funkcji  $\exp$  we wzorze (5.11) nie jest trudne i sprowadza się do kilku wierszy kodu w C++.

Korzystamy bezpośrednio ze wzoru (5.11), pamiętając o współczynniku normalizacyjnym ( $tu = SPaczk^2$ ):

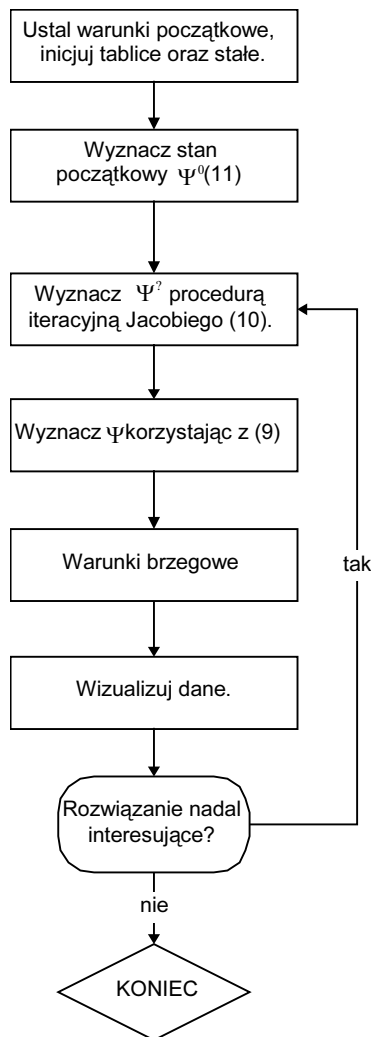
```
double exp_w1, exp_w2;
for(j=0; j<NY; j++)
{
    exp_w1=pow(j-SPaczk-py,2); // pierwsza część wykładnika ze wzoru (5.11)
    for(i=0; i<NX; i++)
    {
        exp_w2=pow(i-SPaczk-px,2); // druga część wykładnika
        exp_w2=(-exp_w1-exp_w2)/pow(SPaczk,2); // suma i normowanie
        psi_t[i][j] = exp(complex<double>(exp_w2,i));
    }
}
```

<sup>12</sup> Zakładamy, że czytelnik zna pojęcie liczby zespolonej.

<sup>13</sup> Przypominamy, że w rozdziale tym zajmujemy się tylko przypadkiem dwuwymiarowym.

**Rysunek 5.1.**

*Algorytm  
działania programu  
Schrödinger  
(rysunek znajduje się  
na płycie CD)*



Widać, że przedstawiony schemat konstrukcji stanu początkowego nie jest złożony. Dużym ułatwieniem było tu wykorzystanie standardowej biblioteki C++ i funkcji `exp()` zdefiniowanej dla klasy liczb zespolonych. Na rysunku 5.2 przedstawiamy stan początkowy przykładowej paczki falowej.

Rysunek 5.2 przedstawia stan określony parametrami:

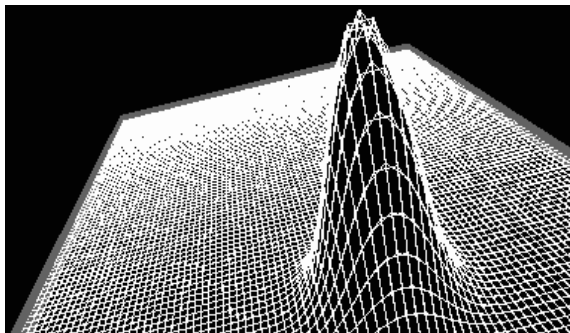
```

SPaczki=NX/10;    // szerokość paczki
px = 4;           // pozycja x
py = 30;          // pozycja y
  
```

gdzie  $NX$  oznacza liczbę komórek na siatce w kierunku osi  $X$  (w tym konkretnym przypadku — 70).

**Rysunek 5.2.**

Przykładowy stan  
początkowy układu  
(rysunek znajduje się  
na płycie CD)



### 5.6.3. Pętla obliczeniowa

Zgodnie z algorytmem z rysunku 5.1 pętlę obliczeniową zaczynamy od wyznaczenia nieznanej funkcji  $\Psi$ <sup>?</sup>.

W osobnej tablicy `**Flag` przechowywać będziemy informacje o potencjale w przestrzeni. Przy założeniu, że potencjał może być albo zerowy, albo mieć stałą wartość niezależną od czasu, wystarczy pamiętać flagi — czy potencjał jest zerowy, czy też wynosi  $e^{-iV\Delta t}$ . Załóżmy, że w pierwszym kroku algorytmu komórki, w których wyróżniliśmy potencjał, mają ustawioną flagę `C_BND`. Ustawiona flaga `C_BND` w danej komórce oznacza, że w tym miejscu potencjał jest niezerowy. Oznacza to, że wyraz  $\exp(-v*dt)$  jest stały w trakcie działania symulacji, dlatego możemy wyliczyć go wcześniej i używać bez dokonywania powtórnych obliczeń. Wyznamy zatem zespoloną liczbę  $e^{-iV\Delta t}$ :

```
exp_vdt=exp(complex<double>(0,-DT*v));
```

Działanie operatora  $e^{-iV\Delta t}$  sprowadzi się w naszym przypadku do wymnożenia przez `exp_vdt`.

Schemat (5.10) podzielimy na dwie części. Najpierw wymnożmy komórki odpowiednio przez 2, jeśli komórka ma potencjał równy 0, lub przez wyliczoną (i podwojoną) wcześniej wartość `2*exp_vdt`, jeśli komórka jest typu `C_BND` (potencjał niezerowy). Zadanie to realizuje podany kod<sup>14</sup>, w którym stosujemy oznaczenia:

$$psi\_t \equiv \Psi$$

$$psi \equiv \Psi^?$$

```
for(i=1;i<NX-1;i++)
for(j=1;j<NY-1;j++)
    if(Flag[i][j] & C_BND) // v != 0?
        psi_t[i][j]=((double)2*exp_vdt)*psi_t[i][j];
    else
        psi_t[i][j]=(double)2*psi_t[i][j];
```

<sup>14</sup> Jawna konwersja zwykłych liczb na typ `double` spowodowana jest istnieniem przeciążonych operatorów tylko dla liczb, przez które sparametryzowany został typ `complex`.

Kolejnym krokiem jest właściwa procedura iteracyjna, która dzięki wcześniejszemu wymnożeniu funkcji `psi_t` przez odpowiednie stałe staje się o wiele prostsza. Procedurę tę wykonujemy `IloscIteracji` razy — każdą iterację dwukrotnie, raz zwiększając, raz zmniejszając indeksy tablicy<sup>15</sup>:

```
for(l=1;l<IloscIteracji;l++)
{
    for(i=1;i<NX-1;i++)
        for(j=1;j<NY-1;j++)
        {
            lapl = (psi[i+1][j]+psi[i-1][j]+psi[i][j-1]+psi[i][j+1]-
                ➡(double)4*psi[i][j]);
            psi[i][j] = psi_t[i][j] + (double)DT *c_i * lapl;
        }

    for(i=NX-1;i>1;i--)
        for(j=NY-1;j>1;j--)
        {
            lapl = (psi[i+1][j]+psi[i-1][j]+psi[i][j-1]+psi[i][j+1]-
                ➡(double)4*psi[i][j]);
            psi[i][j] = psi_t[i][j] + (double)DT *c_i * lapl;
        }
}
```

Uważny czytelnik na pewno zauważył, że do przybliżenia operatora Laplace’a użyliśmy wzoru pięciopunktowego, wprowadzonego już w rozdziale 3. W omówionej procedurze iteracyjnej zastosowaliśmy również oznaczenie `c_i` dla urojonego  $i$  jako:

```
complex<double> c_i(0,1); // po prostu i
```

Dzięki wprowadzeniu urojonego  $i$  mogliśmy bezpośrednio we wzorze (5.10) wykonać mnożenie<sup>16</sup>  $i \cdot \Delta t \cdot \nabla^2 \Psi_n$ .

Kolejną częścią kroku czasowego jest bezpośrednie wyznaczenie rozkładu funkcji falowej  $\Psi$ , czego dokonujemy za pomocą prostego schematu (5.9), pamiętając o podzieleniu `psi_t` przez 2, przez które zostało pomnożone przed pierwszą iteracją:

```
for(i=1;i<NX-1;i++)
    for(j=1;j<NY-1;j++)
        psi_t[i][j]=psi[i][j] - psi_t[i][j]/(double)2;
```

Ostatni krok przedstawionego algorytmu, którym się zajmujemy, to warunki brzegowe rozwiązania. Dla naszych potrzeb prostych stanów początkowych układu założymy zerowanie funkcji falowej na brzegach, czemu odpowiada prosta pętla zerująca:

```
for(i=0;i<NX;i++)
    psi_t[i][0]=psi_t[i][NY]=complex<double>(0,0);
for(j=0;j<NY;j++)
    psi_t[0][j]=psi_t[NX][j]=complex<double>(0,0);
```

<sup>15</sup> Ilość iteracji rzędu 10 na siatkach rozmiaru  $70 \times 70$  i o kroku czasowym  $DT=0.05$  daje już zadowalające rezultaty.

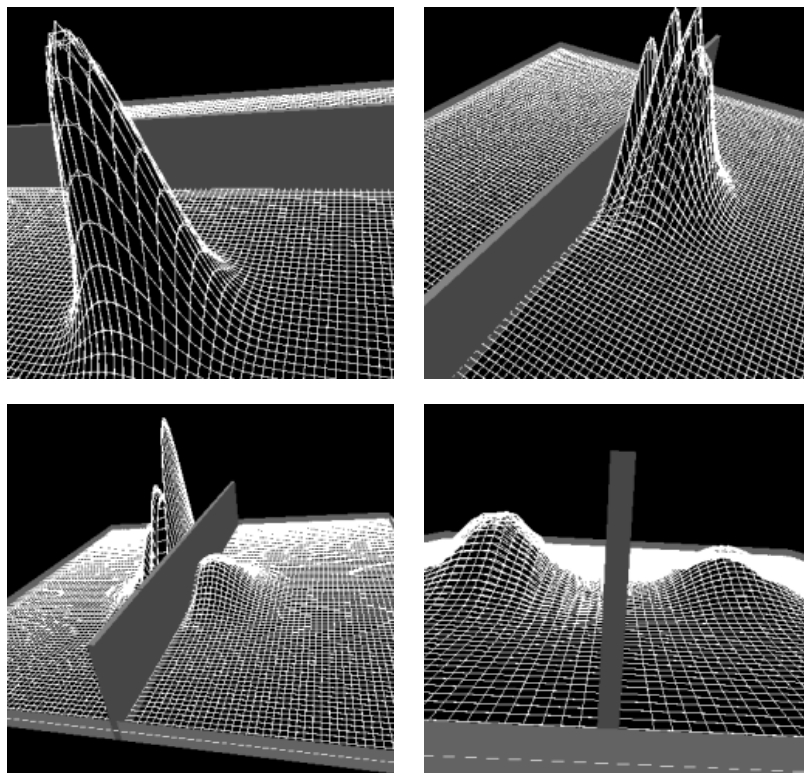
<sup>16</sup> W miejscach takich jak to nie sposób nie zauważyć, jak wspaniałym narzędziem okazuje się C++ i jego obiektowość.

## 5.7. Rezultaty

Przedstawione rezultaty działania programu Schrödinger są zrzutami z ekranu działającej w czasie rzeczywistym animacji. Do wizualizacji wykorzystano stworzone wcześniej na potrzeby programu Waves procedury OpenGL i Fox.

Na rysunku 5.3 przedstawiamy jeden z fundamentalnych efektów kwantowych, czyli przejście przez barierę potencjału.

**Rysunek 5.3.**  
*Przejście paczki  
falowej przez  
barierę potencjału  
(rysunek znajduje  
się na płycie CD)*



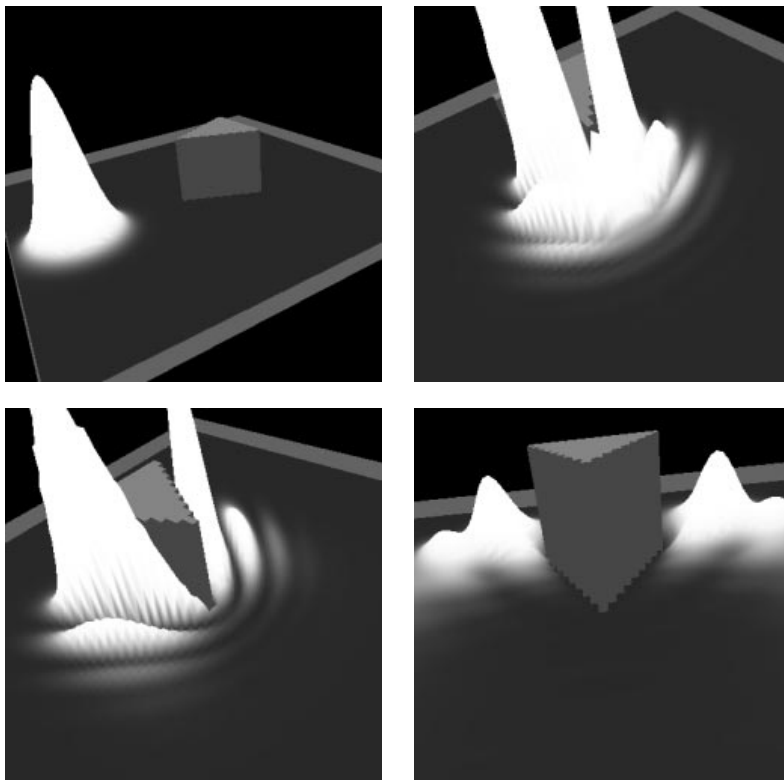
Na przedstawionym rysunku uwidoczniono, że paczka falowa dzieli się na dwie paczki o mniejszej amplitudzie i przeciwnych wektorach falowych. Interesującym efektem jest to, że część paczki falowej, która przeszła przez barierę, jest gładzsza od części odbitej. Sytuacja przedstawiona na rysunku 5.3 nie ma odpowiednika w mechanice klasycznej. Klasycznie bowiem cząstka albo przejdzie przez barierę potencjału (np. ścianę), albo nie przejdzie. Kwantowo rozpatrujemy tylko prawdopodobieństwo przejścia przez barierę. Dlatego niekiedy mówi się, że elektron może być w dwóch miejscach jednocześnie. Niektórzy uważają nawet, że jest w dwóch miejscach jednocześnie.

Na rysunku 5.4 przedstawiamy rozbiecie gaussowskiej paczki falowej na potencjale w kształcie trójkątnego klinu. Ciekawe efekty, które można wyczytać z tych rysunków, to m.in. charakterystyczne spiętrzanie grzbietów fal prawdopodobieństwa w początkowym

etapie symulacji oraz podział paczki na dwie równe części. Na tym rysunku uwidocz-  
niono, że na dalszych etapach symulacji wiemy o rozwiązaniu coraz mniej — funk-  
cja falowa ma teraz przebieg o wiele łagodniejszy, co oznacza, że trudno określić,  
gdzie znajduje się nasz elektron.

**Rysunek 5.4.**

*Rozcięcie  
paczki falowej  
na trójkątnym klinie  
(rysunek znajduje się  
na płycie CD)*



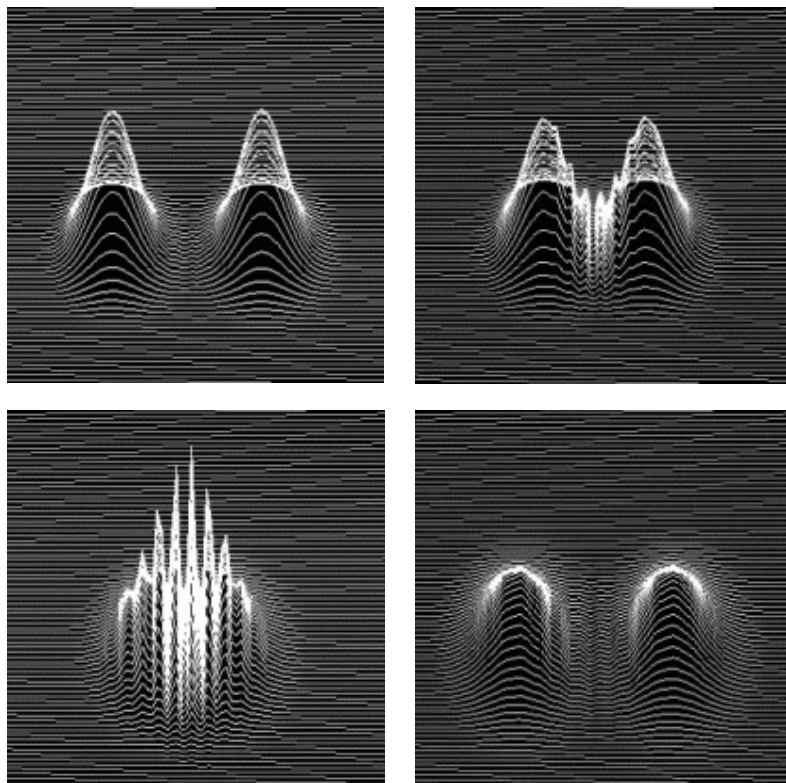
Ostatnim przykładem, który zaprezentujemy, jest przejście wzajemne dwóch paczek falowych przez siebie (rysunek 5.5). Obie paczki falowe mają te same parametry początkowe (amplituda, rozmiary) i przeciwnie skierowane wektory falowe.

## 5.8. Podsumowanie

W tym rozdziale czytelnik zapoznał się ze schematem rozwiązywania numerycznego równania Schrödingera. Zastosowane tu schematy różnicowe zdają się być wystarczające do zaobserwowania ciekawych efektów kwantowych. Istnieje duża dowolność w stosowaniu przedstawionej metody — można próbować dowolnych kształtów i wielkości potencjałów. Stan początkowy nie musi być przedstawiony funkcją Gaussa. Można też stosować hamiltonian inny niż dla cząsteczki swobodnej. Rezultaty uzyskane po wprowadzeniu hamiltonianu oscylatora harmonicznego w dwóch wymiarach do

**Rysunek 5.5.**

*Rozpraszanie  
wzajemne dwóch  
paczek falowych  
(rysunek znajduje się  
na płycie CD)*



przedstawionego algorytmu zdają się potwierdzać, że przedstawiona metoda rozwiązania numerycznego, mimo że prosta, jest dosyć ogólna i ma szerokie zastosowanie, dlatego zachęcamy czytelnika do samodzielnego rozwijania omawianego programu.