

Paweł Syty

# Programowanie metody J-macierzy

Praca dyplomowa wykonana pod kierunkiem  
dr hab. Józefa E. Sienkiewicza, prof. ndzw. PG

Katedra Fizyki Teoretycznej i Metod Matematycznych  
Wydział Fizyki Technicznej i Matematyki Stosowanej  
Politechnika Gdańska

Gdańsk, luty 1999

## Podziękowania

Dziękuję Panu Pawłowi Horodeckiemu za pomoc,  
bez której powstanie tej pracy byłoby niemożliwe.

## SPIS RZECZY

<b>1. Wstęp .....</b>	<b>4</b>
<b>2. Równanie Diraca.....</b>	<b>5</b>
2.1 Cząstka swobodna .....	5
2.2 Cząstka w polu centralnym .....	8
2.3 Funkcje własne hamiltonianu Diraca w polu centralnym .....	15
2.4 Radialne równanie Diraca .....	18
<b>3. Potencjalne rozpraszanie elektronów .....</b>	<b>20</b>
3.1 Obliczanie różniczkowych przekrojów czynnych .....	20
3.2 Metoda przesunięć fazowych .....	24
3.3 Metoda fal parcjalnych.....	26
<b>4. Metoda J-macierzy .....</b>	<b>35</b>
4.1 Nierelatywistyczna metoda J-macierzy.....	35
4.2 Relatywistyczna metoda J-macierzy .....	40
<b>5. Obliczenia numeryczne .....</b>	<b>48</b>
5.1 Przeskalowanie energii.....	48
5.2 Program komputerowy .....	49
5.3 Wzory analityczne do przetestowania programu .....	54
5.3.1 Przypadek nierelatywistyczny .....	54
5.3.2 Przypadek relatywistyczny .....	55
5.4 Metodologia obliczeń.....	56
5.5 Wyniki obliczeń .....	57
5.5.1 Nierelatywistyczna metoda J-macierzy .....	57
5.5.2 Relatywistyczna metoda J-macierzy.....	61
5.5.3 Relatywistyczna metoda J-macierzy w granicy nierelatywistycznej.....	65
<b>6. Podsumowanie .....</b>	<b>66</b>
<b>7. Dodatek.....</b>	<b>67</b>
A. Własności wielomianów Legendre'a oraz harmonik sferycznych.....	67
B. Sferyczne funkcje Bessela, funkcje Riccati-Bessela .....	68
C. Wybrane elementy baz Laguerre'a i Gaussa .....	70
D. Kody źródłowe programów .....	71
D.1 Fortran 77.....	71
D.2 Mathematica 3.0.....	80
<b>8. Bibliografia.....</b>	<b>84</b>

## 1. Wstęp

Metoda J-macierzy, zaproponowana przez Hellera i Yamaniego w 1974 roku [1], [2] i później rozwinięta przez Yamaniego i Fishmana [3], jest przykładem algebraicznej metody w kwantowej teorii rozpraszania potencjalnego. Punktem kluczowym metody jest reprezentacja hamiltonianu w odpowiedniej bazie, co zmienia różniczkowy problem rozpraszania w czysto algebraiczny. Relatywistyczna wersja metody została sformułowana przez P. Horodeckiego w 1998 roku [4] jako naturalne rozszerzenie teorii nierelatywistycznej.

W rozdziale drugim przedstawiono szczegółowy opis elektronu swobodnego oraz elektronu w polu centralnym w oparciu o równanie Diraca. Wyprowadzone zostały również w nim radialne równania Diraca. W rozdziałach trzecim i czwartym zawarte są metody obliczania różniczkowych przekrojów czynnych oraz przesunięć fazowych w rozpraszaniu, między innymi metodą J-macierzy (wersje nierelatywistyczna i relatywistyczna), w oparciu o którą autor wykonał obliczenia numeryczne opisane w rozdziale piątym. Wnioski autor zawarł w rozdziale szóstym.

Rozdziały drugi oraz trzeci oraz uzupełnienia A i B oparte są na notatkach R. Szmytkowskiego [5] i pracy dyplomowej M. Kroćenickiego [6]. Rozdział czwarty oparty jest na pracy P. Horodeckiego [4] oraz częściowo na pracy Yamaniego i Fishmana [3].

## 2. Równanie Diraca<sup>1</sup>

### 2.1 Cząstka swobodna

W celu wyprowadzenia równania Diraca, napiszmy relatywistyczny związek między energią i pędem dla cząstki swobodnej:

$$E = \sqrt{c^2 \mathbf{p}^2 + m^2 c^4}. \quad (2.1.1)$$

Oczywiście  $\mathbf{p}^2 = p_x^2 + p_y^2 + p_z^2$ .

Zastępujemy energię  $E$  oraz składowe pędu  $\mathbf{p}$  odpowiednimi operatorami:

$$E \rightarrow i\hbar \frac{\partial}{\partial t}, \quad p_x \rightarrow -i\hbar \frac{\partial}{\partial x}, \quad p_y \rightarrow -i\hbar \frac{\partial}{\partial y}, \quad p_z \rightarrow -i\hbar \frac{\partial}{\partial z}.$$

Zgodnie z regułami mechaniki kwantowej, powyższe operatory działają na funkcję falową  $\Psi$ :

$$i\hbar \frac{\partial}{\partial t} \Psi = \sqrt{-\hbar^2 c^2 \nabla^2 + m^2 c^4} \Psi.$$

Problem wyciągania pierwiastka z operatora Dirac rozwiązał nie podnosząc obu stron równania (2.1.1) do kwadratu, co prowadziłoby do równania Kleina-Gordona. Skorzystał natomiast z pewnych własności tzw. macierzy Pauliego:

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (2.1.2)$$

Szukany pierwiastek znajdujemy w następującej postaci

$$\sqrt{c^2 \hat{\mathbf{p}}^2 + m^2 c^4} = c\boldsymbol{\alpha} \cdot \hat{\mathbf{p}} + \beta mc^2,$$

gdzie

$$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3), \quad \hat{\mathbf{p}} = (p_x, p_y, p_z).$$

Z porównania stron wynika, że wielkości  $\alpha_k$  i  $\beta$  muszą posiadać następujące własności:

$$\alpha_k \alpha_l + \alpha_l \alpha_k = 2\delta_{kl} I, \quad \alpha_k \beta + \beta \alpha_k = 0, \quad \beta^2 = 1, \quad k, l = 1, 2, 3.$$

Dirac znalazł twory matematyczne spełniające powyższe zależności w postaci macierzy  $4 \times 4$

---

<sup>1</sup> Rozdział oparty jest o nie opublikowane notatki R. Szymtkowskiego [5] oraz pracę M. Krośnickiego [6].

$$\alpha_k = \begin{pmatrix} 0_2 & \sigma_k \\ \sigma_k & 0_2 \end{pmatrix}, \quad \beta = \begin{pmatrix} I_2 & 0_2 \\ 0_2 & -I_2 \end{pmatrix}. \quad (2.1.3)$$

Zależne od czasu równanie Diraca możemy więc zapisać w zwartej postaci

$$i\hbar \frac{\partial}{\partial t} \Psi(\mathbf{r}, t) = \hat{H} \Psi(\mathbf{r}, t), \quad (2.1.4)$$

gdzie

$$\Psi(\mathbf{r}, t) = \begin{pmatrix} \Psi_1(\mathbf{r}, t) \\ \Psi_2(\mathbf{r}, t) \\ \Psi_3(\mathbf{r}, t) \\ \Psi_4(\mathbf{r}, t) \end{pmatrix} \quad (2.1.5)$$

jest funkcją czteroskładnikową (spinorem),  $\hat{H}$  jest hamiltonianem Diraca. Dla cząstki swobodnej ma on wcześniej wprowadzoną postać:

$$\hat{H} = c\boldsymbol{\alpha} \cdot \hat{\mathbf{p}} + \beta mc^2, \quad (2.1.6)$$

gdzie  $\hat{\mathbf{p}} = -i\hbar\nabla$ , natomiast  $\boldsymbol{\alpha}$  jest wektorem określonym wcześniej.

Napiszmy równanie Diraca w postaci

$$\begin{aligned} i\hbar \frac{\partial}{\partial t} \Psi_1(\mathbf{r}, t) + i\hbar c \left( \frac{\partial}{\partial x} \Psi_4(\mathbf{r}, t) - i \frac{\partial}{\partial y} \Psi_4(\mathbf{r}, t) + \frac{\partial}{\partial z} \Psi_3(\mathbf{r}, t) \right) - mc^2 \Psi_1(\mathbf{r}, t) &= 0, \\ i\hbar \frac{\partial}{\partial t} \Psi_2(\mathbf{r}, t) + i\hbar c \left( \frac{\partial}{\partial x} \Psi_3(\mathbf{r}, t) + i \frac{\partial}{\partial y} \Psi_3(\mathbf{r}, t) - \frac{\partial}{\partial z} \Psi_4(\mathbf{r}, t) \right) - mc^2 \Psi_2(\mathbf{r}, t) &= 0, \\ i\hbar \frac{\partial}{\partial t} \Psi_3(\mathbf{r}, t) + i\hbar c \left( \frac{\partial}{\partial x} \Psi_2(\mathbf{r}, t) - i \frac{\partial}{\partial y} \Psi_2(\mathbf{r}, t) + \frac{\partial}{\partial z} \Psi_1(\mathbf{r}, t) \right) + mc^2 \Psi_3(\mathbf{r}, t) &= 0, \\ i\hbar \frac{\partial}{\partial t} \Psi_4(\mathbf{r}, t) + i\hbar c \left( \frac{\partial}{\partial x} \Psi_1(\mathbf{r}, t) + i \frac{\partial}{\partial y} \Psi_1(\mathbf{r}, t) - \frac{\partial}{\partial z} \Psi_2(\mathbf{r}, t) \right) + mc^2 \Psi_4(\mathbf{r}, t) &= 0. \end{aligned} \quad (2.1.7)$$

Zakładamy rozwiązania powyższego równania w postaci fali płaskiej

$$\Psi_\mu(\mathbf{r}, t) = a_\mu e^{i(kz - \omega t)}, \quad \mu = 1, 2, 3, 4.$$

Po podstawieniu rozwiązania, oraz uwzględnieniu, że  $E = \hbar\omega$  i  $p_z = \hbar k$ , otrzymujemy:

$$\begin{array}{rcl}
(E - mc^2)a_1 & -cp_z a_3 & = 0, \\
(E - mc^2)a_2 & +cp_z a_4 & = 0, \\
-cp_z a_1 & + (E + mc^2)a_3 & = 0, \\
cp_z a_2 & + (E + mc^2)a_4 & = 0.
\end{array} \tag{2.1.8}$$

Układ równań (2.1.8) jest układem jednorodnym, ma więc nietrywialne rozwiązania wtedy i tylko wtedy, gdy zeruje się wyznacznik macierzy jego współczynników:

$$\begin{pmatrix} E - mc^2 & 0 & -cp_z & 0 \\ 0 & E - mc^2 & 0 & cp_z \\ -cp_z & 0 & E + mc^2 & 0 \\ 0 & cp_z & 0 & E + mc^2 \end{pmatrix}. \tag{2.1.9}$$

Zachodzi to wówczas, gdy  $E = \pm\sqrt{c^2 \mathbf{p}^2 + m^2 c^4}$ . Spełniony jest zatem, znany ze szczególnej teorii względności, związek pędu i energii.

Ponieważ macierz (2.1.9) jest macierzą rzędu drugiego, to układ równań (2.1.8) posiada tylko dwa liniowo niezależne rozwiązania. Możemy je zapisać w postaci

$$\begin{aligned}
a_1 = 1, \quad a_2 = 0, \quad a_3 = \frac{cp_z}{E + mc^2}, \quad a_4 = 0, \\
a_1 = 0, \quad a_2 = 1, \quad a_3 = 0, \quad a_4 = \frac{-cp_z}{E + mc^2}.
\end{aligned}$$

Rozwiązania te można zapisać w zwartej postaci:

$$\Psi(\mathbf{r}, t) = \begin{pmatrix} 1 \\ 0 \\ \frac{cp_z}{E + mc^2} \\ 0 \end{pmatrix} e^{i(kz - \omega t)} \tag{2.1.10}$$

oraz

$$\Psi(\mathbf{r}, t) = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \frac{-cp_z}{E + mc^2} \end{pmatrix} e^{i(kz - \omega t)}. \tag{2.1.11}$$

Rozwiązanie równania (2.1.7) jest kombinacją liniową powyższych rozwiązań:

$$\Psi(\mathbf{r}, t) = \left[ A \begin{pmatrix} 1 \\ 0 \\ \frac{cp_z}{E + mc^2} \\ 0 \end{pmatrix} + B \begin{pmatrix} 0 \\ 1 \\ 0 \\ \frac{-cp_z}{E + mc^2} \end{pmatrix} \right] e^{i(kz - \omega t)}.$$

## 2.2 Cząstka w polu centralnym

Hamiltonian Diraca dla elektronu w polu centralnym  $V(r)$  ma postać

$$\hat{H} = c\boldsymbol{\alpha} \cdot \hat{\mathbf{p}} + \beta mc^2 + V(r). \quad (2.2.1)$$

Operator całkowitego momentu pędu elektronu  $\mathbf{J}$  jest zdefiniowany jako suma operatorów orbitalnego momentu pędu  $\mathbf{L}$  i spinu  $\mathbf{S}$ , tzn.  $\mathbf{J} = \mathbf{L} + \mathbf{S}$ , gdzie  $\mathbf{L} = -i\hbar\mathbf{r} \times \nabla$ ,  $\mathbf{S} = \frac{1}{2}\hbar\boldsymbol{\sigma}^D$ .

Macierz  $\boldsymbol{\sigma}^D$  jest zdefiniowana następująco

$$\boldsymbol{\sigma}^D = \begin{pmatrix} \boldsymbol{\sigma} & 0 \\ 0 & \boldsymbol{\sigma} \end{pmatrix}. \quad (2.2.2)$$

Zdefiniujmy ponadto następujące operatory:

$$\mathbf{J}^2 = (\mathbf{L} + \mathbf{S})^2, \quad J_z = L_z + S_z.$$

Operator  $L_z$  jest rzutem operatora orbitalnego momentu pędu  $\mathbf{L}$  na wyróżnioną oś  $z$ , natomiast operator  $S_z$  jest rzutem operatora spinu na tę samą oś. Będziemy także posługiwać się związanymi z nimi operatorami  $\boldsymbol{\Sigma}$  i  $\Sigma_z$

$$\begin{aligned} \hbar^2 \boldsymbol{\Sigma}^2 &= \mathbf{J}^2 = \hbar^2 \left( \boldsymbol{\Lambda} + \frac{1}{2} \boldsymbol{\sigma}^D \right)^2, \\ \hbar \Sigma_z &= J_z = \hbar \left( \Lambda_z + \frac{1}{2} \sigma_z^D \right), \end{aligned}$$

gdzie zdefiniowano  $\boldsymbol{\Lambda} = \frac{\mathbf{L}}{\hbar}$  oraz  $\Lambda_z = \frac{L_z}{\hbar}$ .

Napiszmy teraz równanie własne dla operatora składowej całkowitego momentu pędu  $\Sigma_z$  w kierunku  $z$ :

$$\left( \Lambda_z + \frac{1}{2} \sigma_z^D \right) \Psi = m \Psi. \quad (2.2.3)$$



Liczbę  $m$  nazywamy magnetyczną liczbą kwantową<sup>2</sup>.

Przedstawiamy funkcję  $\Psi$  w postaci czteroskładnikowego spinora (2.1.5) oraz korzystamy z postaci operatora  $\Lambda_z$  we współrzędnych sferycznych

$$\Lambda_z = -i \frac{\partial}{\partial \varphi}$$

oraz ze wzorów (2.1.2) i (2.2.2), otrzymując następujące równania:

$$\frac{\partial}{\partial \varphi} \Psi_1(r, \theta, \varphi) = i \left( m - \frac{1}{2} \right) \Psi_1(r, \theta, \varphi), \quad (2.2.4)$$

$$\frac{\partial}{\partial \varphi} \Psi_2(r, \theta, \varphi) = i \left( m + \frac{1}{2} \right) \Psi_2(r, \theta, \varphi),$$

$$\frac{\partial}{\partial \varphi} \Psi_3(r, \theta, \varphi) = i \left( m - \frac{1}{2} \right) \Psi_3(r, \theta, \varphi),$$

$$\frac{\partial}{\partial \varphi} \Psi_4(r, \theta, \varphi) = i \left( m + \frac{1}{2} \right) \Psi_4(r, \theta, \varphi). \quad (2.2.5)$$

Rozwiązania powyższych równań są następujące:

$$\Psi_1(r, \theta, \varphi) = C_1(r, \theta) e^{i \left( m - \frac{1}{2} \right) \varphi},$$

$$\Psi_2(r, \theta, \varphi) = C_2(r, \theta) e^{i \left( m + \frac{1}{2} \right) \varphi},$$

$$\Psi_3(r, \theta, \varphi) = C_3(r, \theta) e^{i \left( m - \frac{1}{2} \right) \varphi},$$

$$\Psi_4(r, \theta, \varphi) = C_4(r, \theta) e^{i \left( m + \frac{1}{2} \right) \varphi}.$$

Aby zachować jednoznaczność problemu własnego (2.2.3) przy obrotach o kąt  $2\pi$ , liczba  $m$  musi być liczbą połówkową, tzn.

$$m = \pm \frac{1}{2}, \pm \frac{3}{2}, \pm \frac{5}{2}, \pm \frac{7}{2}, \dots$$

Wprowadźmy operator

$$\hat{K} = \beta(\mathbf{\Lambda} \cdot \boldsymbol{\sigma}^D + 1).$$

---

<sup>2</sup> Wcześniej symbolem  $m$  oznaczano masę cząstki, jednak, według autora, nie powinno to prowadzić do nieporozumień.

Można wykazać, że operator ten komutuje z hamiltonianem (2.2.1) a także z operatorami  $\mathbf{J}^2$  ( $\Sigma^2$ ). Wynika stąd, że zamiast szukać jednoczesnych funkcji własnych operatorów  $\hat{H}$ ,  $\Sigma^2$ ,  $\Sigma_z$ , możemy szukać jednoczesnych funkcji własnych operatorów  $\hat{H}$ ,  $\hat{K}$ ,  $\Sigma_z$ .

Chcemy znaleźć funkcje własne operatora  $\hat{K}$ . W tym celu przedstawmy funkcję  $\Psi$  w postaci

$$\Psi = \begin{pmatrix} \Psi_A \\ \Psi_B \end{pmatrix}, \quad (2.2.6)$$

gdzie  $\Psi_A$  i  $\Psi_B$  są dwuskładnikowe:

$$\Psi_A = \begin{pmatrix} \Psi_1 \\ \Psi_2 \end{pmatrix}, \quad \Psi_B = \begin{pmatrix} \Psi_3 \\ \Psi_4 \end{pmatrix}. \quad (2.2.7)$$

Równanie na wartości własne operatora  $\hat{K}$  możemy zapisać w postaci

$$\beta(\mathbf{\Lambda} \cdot \boldsymbol{\sigma}^D + 1) \begin{pmatrix} \Psi_A \\ \Psi_B \end{pmatrix} = -\kappa \begin{pmatrix} \Psi_A \\ \Psi_B \end{pmatrix}$$

lub

$$\begin{cases} (\mathbf{\Lambda} \cdot \boldsymbol{\sigma} + 1 + \kappa)\Psi_A = 0 \\ (\mathbf{\Lambda} \cdot \boldsymbol{\sigma} + 1 - \kappa)\Psi_B = 0. \end{cases}$$

Jak widać, w powyższych równaniach  $\Psi_A$  i  $\Psi_B$  są rozseparowane. Używając wzorów (2.1.2) oraz (2.2.7), otrzymujemy

$$\begin{aligned} \begin{pmatrix} \Lambda_z + 1 + \kappa & \Lambda_- \\ \Lambda_+ & -\Lambda_z + 1 + \kappa \end{pmatrix} \begin{pmatrix} \Psi_1 \\ \Psi_2 \end{pmatrix} &= 0, \\ \begin{pmatrix} \Lambda_z + 1 - \kappa & \Lambda_- \\ \Lambda_+ & -\Lambda_z + 1 - \kappa \end{pmatrix} \begin{pmatrix} \Psi_3 \\ \Psi_4 \end{pmatrix} &= 0, \end{aligned} \quad (2.2.8)$$

gdzie zdefiniowano  $\Lambda_{\pm} = \Lambda_x \pm i\Lambda_y$ .

Wykonajmy mnożenie w pierwszym z powyższych równań:

$$\begin{cases} (\Lambda_z + 1 + \kappa)\Psi_1 + \Lambda_- \Psi_2 = 0 \\ (\Lambda_z - 1 - \kappa)\Psi_2 - \Lambda_+ \Psi_1 = 0. \end{cases} \quad (2.2.9)$$

Korzystając z relacji komutacyjnej operatorów  $\Lambda_x$ ,  $\Lambda_y$

$$[\Lambda_x, \Lambda_y] = i\Lambda_z,$$

otrzymujemy dla operatorów  $\Lambda_+$ ,  $\Lambda_-$  następujące związki:

$$\begin{aligned}\Lambda_+\Lambda_- &= \Lambda^2 - \Lambda_z^2 + \Lambda_z, \\ \Lambda_-\Lambda_+ &= \Lambda^2 - \Lambda_z^2 - \Lambda_z.\end{aligned}$$

Działając na pierwsze z równań (2.2.9) operatorem  $\Lambda_+$ , na drugie operatorem  $\Lambda_-$  oraz korzystając z powyższych związków, mamy:

$$\begin{cases} \Lambda_+(\Lambda_z + 1 + \kappa)\Psi_1 + (\Lambda^2 - \Lambda_z^2 + \Lambda_z)\Psi_2 = 0 \\ \Lambda_-(\Lambda_z - 1 - \kappa)\Psi_2 - (\Lambda^2 - \Lambda_z^2 - \Lambda_z)\Psi_1 = 0. \end{cases}$$

W czterospinorze  $\Psi$  założyliśmy, że jest on funkcją własną operatora  $\Sigma_z$  z wartością własną  $m$ . Wykorzystując równania (2.2.4), możemy przepisać powyższe równania w postaci

$$\begin{cases} \Lambda_+\left(m + \frac{1}{2} + \kappa\right)\Psi_1 + \left[\Lambda^2 - \left(m + \frac{1}{2}\right)^2 + \left(m + \frac{1}{2}\right)\right]\Psi_2 = 0 \\ \Lambda_-\left(m - \frac{1}{2} - \kappa\right)\Psi_2 - \left[\Lambda^2 - \left(m - \frac{1}{2}\right)^2 - \left(m - \frac{1}{2}\right)\right]\Psi_1 = 0. \end{cases}$$

Podstawiając wyżej równania (2.2.9), otrzymujemy:

$$\begin{cases} \left(m + \frac{1}{2} + \kappa\right)(\Lambda_z - 1 - \kappa)\Psi_2 + \left[\Lambda^2 - \left(m + \frac{1}{2}\right)^2 + \left(m + \frac{1}{2}\right)\right]\Psi_2 = 0 \\ -\left(m - \frac{1}{2} - \kappa\right)(\Lambda_z + 1 + \kappa)\Psi_1 - \left[\Lambda^2 - \left(m - \frac{1}{2}\right)^2 - \left(m - \frac{1}{2}\right)\right]\Psi_1 = 0. \end{cases}$$

Podstawiając ponownie równania (2.2.4), otrzymamy

$$\begin{cases} \left(m + \frac{1}{2} + \kappa\right)\left(m - \frac{1}{2} - \kappa\right)\Psi_2 + \left[\Lambda^2 - \left(m + \frac{1}{2}\right)^2 + \left(m + \frac{1}{2}\right)\right]\Psi_2 = 0 \\ -\left(m - \frac{1}{2} - \kappa\right)\left(m + \frac{1}{2} + \kappa\right)\Psi_1 - \left[\Lambda^2 - \left(m - \frac{1}{2}\right)^2 - \left(m - \frac{1}{2}\right)\right]\Psi_1 = 0. \end{cases}$$

Po uproszczeniach dostajemy

$$\begin{aligned}\Lambda^2\Psi_1 &= \kappa(\kappa + 1)\Psi_1, \\ \Lambda^2\Psi_2 &= \kappa(\kappa + 1)\Psi_2.\end{aligned}\tag{2.2.10}$$

Powyższe równania są równaniami na wartości własne dla operatora  $\Lambda^2$ . Wiemy, że wartościami własnymi tego operatora są  $l(l+1)$ , a funkcjami własnymi są harmoniki sferyczne, zatem powyższe równania można przepisać w postaci

$$\begin{aligned}\Lambda^2\Psi_1 &= l(l+1)\Psi_1, \\ \Lambda^2\Psi_2 &= l(l+1)\Psi_2.\end{aligned}\tag{2.2.11}$$

Z porównania (2.2.10) i (2.2.11) wynika, że

$$\kappa^2 + \kappa - l(l+1) = 0,\tag{2.2.12}$$

skąd z kolei wynika, że

$$\kappa = l \quad \vee \quad \kappa = -l-1.$$

Ze względu na słuszność równań (2.2.4), musi zachodzić

$$\begin{aligned}\Psi_1 &= R_1(r)Y_l^{m-\frac{1}{2}}(\hat{\mathbf{r}}), \\ \Psi_2 &= R_2(r)Y_l^{m+\frac{1}{2}}(\hat{\mathbf{r}}).\end{aligned}$$

Przepiszmy teraz drugie z równań (2.2.8) w postaci:

$$\begin{cases} (\Lambda_z + 1 - \kappa)\Psi_3 + \Lambda_- \Psi_4 = 0 \\ (\Lambda_z - 1 + \kappa)\Psi_4 - \Lambda_+ \Psi_3 = 0. \end{cases}$$

Porównując te równania z równaniami (2.2.9) zauważamy, że różnią się od siebie tylko znakiem przed  $\kappa$ . Oznaczając rząd harmoniki sferycznej wchodzącej w skład  $\Psi_3, \Psi_4$  przez  $l'$  i wykonując w (2.2.10) i w (2.2.12) zamianę  $\kappa \leftrightarrow -\kappa$ , otrzymujemy dla składowych  $\Psi_3$  i  $\Psi_4$  następujące równania

$$\begin{aligned}\Lambda^2\Psi_3 &= \kappa(\kappa-1)\Psi_3, \\ \Lambda^2\Psi_4 &= \kappa(\kappa-1)\Psi_4,\end{aligned}$$

oraz

$$l'^2 + l' - \kappa(\kappa-1) = 0.$$

Rozwiązania powyższego równania są następujące:

$$l' = \begin{cases} -\kappa \\ \kappa-1. \end{cases}$$

Rozpatrzmy dwa przypadki

$$\kappa = l \Rightarrow l' = \begin{cases} -l \\ l-1, \end{cases}$$

$$\kappa = -l-1 \Rightarrow l' = \begin{cases} l+1 \\ -l-2. \end{cases}$$

Ponieważ  $l'$  jest nieujemne (jest to bowiem rząd harmoniki sferycznej), otrzymujemy:

$$\kappa = \begin{cases} l & \text{dla } \kappa > 0 \\ -l-1 & \text{dla } \kappa < 0, \end{cases}$$

$$l' = \begin{cases} l-1, & l \geq 1, \quad \kappa > 0 \\ l+1, & l \geq 0, \quad \kappa < 0. \end{cases}$$

Ze względu na powyższe zależności oraz wzory (2.2.5) mamy dla  $\kappa > 0$  (czyli  $\kappa = l$ )

$$\Psi_3 = R_3(r)Y_{l-1}^{m-\frac{1}{2}}(\hat{\mathbf{r}}),$$

$$\Psi_4 = R_4(r)Y_{l-1}^{m+\frac{1}{2}}(\hat{\mathbf{r}}),$$

oraz dla  $\kappa < 0$  (czyli  $\kappa = -l-1$ )

$$\Psi_3 = R_3'(r)Y_{l+1}^{m-\frac{1}{2}}(\hat{\mathbf{r}}),$$

$$\Psi_4 = R_4'(r)Y_{l+1}^{m+\frac{1}{2}}(\hat{\mathbf{r}}).$$

Rozważmy teraz równanie na wartości własne dla operatora  $\Sigma^2$ :

$$\Sigma^2 \Psi = j(j+1)\Psi. \quad (2.2.13)$$

Ponieważ

$$\Sigma^2 = \Lambda^2 + \Lambda \cdot \boldsymbol{\sigma}^D + \frac{1}{4}(\boldsymbol{\sigma}^D)^2,$$

gdzie

$$(\boldsymbol{\sigma}^D)^2 = \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{pmatrix},$$

oraz, po skorzystaniu z definicji operatora  $\hat{K}$ , równanie (2.2.13) przyjmuje postać

$$\left( \Lambda^2 + \beta \hat{K} - \frac{1}{4} \right) \Psi = j(j+1)\Psi.$$

Przedstawiając funkcje  $\Psi$  w postaci (2.2.6) otrzymamy:

$$\begin{aligned}\left(\Lambda^2 + \hat{K} - \frac{1}{4}\right)\Psi_A &= j(j+1)\Psi_A, \\ \left(\Lambda^2 - \hat{K} - \frac{1}{4}\right)\Psi_B &= j(j+1)\Psi_B.\end{aligned}$$

Jednakże  $\Psi_A$  jest funkcją własną operatorów  $\Lambda^2$  i  $\hat{K}$  z wartościami własnymi odpowiednio  $l(l+1)$  i  $-\kappa$ , lub równoważnie  $\kappa(\kappa+1)$  i  $-\kappa$ .  $\Psi_B$  jest natomiast funkcją własną operatorów  $\Lambda^2$  i  $\hat{K}$  z wartościami własnymi odpowiednio  $\kappa(\kappa-1)$  i  $+\kappa$ . Oznacza to, że ostatnie równania przyjmują postać

$$\begin{aligned}(\kappa^2 - 1/4)\Psi_A &= j(j+1)\Psi_A, \\ (\kappa^2 - 1/4)\Psi_B &= j(j+1)\Psi_B.\end{aligned}$$

Aby rozwiązania powyższych równań były nietrywialne, musi zachodzić

$$j^2 + j - \left(\kappa^2 - \frac{1}{4}\right) = 0 \Rightarrow j = \begin{cases} -\kappa - \frac{1}{2} \\ \kappa - \frac{1}{2} \end{cases}.$$

Rozpatrujemy dwa możliwe przypadki:  $\kappa > 0$  i  $\kappa < 0$ , przyjmując przy tym, że  $j > 0$ :

$$\begin{aligned}\kappa > 0, \quad \kappa = l &\Rightarrow j = l - 1/2, \\ \kappa < 0, \quad \kappa = -l - 1 &\Rightarrow j = l + 1/2.\end{aligned}$$

Najogólniej związek między liczbami kwantowymi  $\kappa, j, l$  można przedstawić w postaci

$$\kappa = (2j + 1)(l - j).$$

Zbierzmy powyższe rozważania w tabeli:

$\kappa > 0, \quad \kappa = l, \quad j = l - \frac{1}{2}$	$\kappa < 0, \quad \kappa = -l - 1, \quad j = l + \frac{1}{2}$
$\Psi_1 = R_1(r)Y_l^{m-\frac{1}{2}}(\hat{r})$	$\Psi_1 = R'_1(r)Y_l^{m-\frac{1}{2}}(\hat{r})$
$\Psi_2 = R_2(r)Y_l^{m+\frac{1}{2}}(\hat{r})$	$\Psi_2 = R'_2(r)Y_l^{m+\frac{1}{2}}(\hat{r})$
$\Psi_3 = R_3(r)Y_{l-1}^{m-\frac{1}{2}}(\hat{r})$	$\Psi_3 = R'_3(r)Y_{l+1}^{m-\frac{1}{2}}(\hat{r})$
$\Psi_4 = R_4(r)Y_{l-1}^{m+\frac{1}{2}}(\hat{r})$	$\Psi_4 = R'_4(r)Y_{l+1}^{m+\frac{1}{2}}(\hat{r})$

Tabela przedstawia funkcje własne hamiltonianu Diraca dla elektronu w polu centralnym

### 2.3 Funkcje własne hamiltonianu Diraca w polu centralnym

Ponownie rozpatrzmy układ równań (2.2.9), gdzie  $\Psi_1$ ,  $\Psi_2$  są następującej postaci:

$$\Psi_A = \begin{pmatrix} \Psi_1 \\ \Psi_2 \end{pmatrix} = \begin{pmatrix} R_1(r)Y_l^{m-\frac{1}{2}}(\hat{r}) \\ R_2(r)Y_l^{m+\frac{1}{2}}(\hat{r}) \end{pmatrix}.$$

Korzystając ze wzorów (2.2.4), (A.1), możemy (2.2.9) przepisać następująco:

$$\begin{aligned} \left(m - \frac{1}{2} + 1 + \kappa\right) R_1(r)Y_l^{m-\frac{1}{2}}(\hat{r}) + \sqrt{\left(l+m+\frac{1}{2}\right)\left(l-m+\frac{1}{2}\right)} R_2(r)Y_l^{m-\frac{1}{2}}(\hat{r}) &= 0, \\ \sqrt{\left(l-m+\frac{1}{2}\right)\left(l+m+\frac{1}{2}\right)} R_1(r)Y_l^{m+\frac{1}{2}}(\hat{r}) + \left(m + \frac{1}{2} - 1 - \kappa\right) R_2(r)Y_l^{m+\frac{1}{2}}(\hat{r}) &= 0. \end{aligned}$$

Powyższy układ równań musi być spełniony dla dowolnego kierunku wektora  $\hat{r}$ , więc

$$\begin{aligned} \left(m + \kappa + \frac{1}{2}\right) R_1(r) + \sqrt{\left(l+m+\frac{1}{2}\right)\left(l-m+\frac{1}{2}\right)} R_2(r) &= 0, \\ -\sqrt{\left(l-m+\frac{1}{2}\right)\left(l+m+\frac{1}{2}\right)} R_1(r) + \left(m - \kappa - \frac{1}{2}\right) R_2(r) &= 0. \end{aligned} \tag{2.3.1}$$

Aby układ ten posiadał nietrywialne rozwiązania, musi zniknąć jego wyznacznik wiekowy. Dostajemy więc

$$\left(\kappa + \frac{1}{2}\right)^2 = \left(l + \frac{1}{2}\right)^2,$$

skąd otrzymujemy:

$$\kappa_1 = l \quad \vee \quad \kappa_2 = -l - 1.$$

Ze wzoru (2.3.1) wynika, że funkcje  $R_1(r)$  i  $R_2(r)$  różnią się tylko o stały czynnik. W związku z tym możemy napisać, że  $R_1(r)$  i  $R_2(r)$  są proporcjonalne do pewnej funkcji  $\tilde{P}(r)$

$$\begin{aligned} R_1(r) &= A\tilde{P}(r), \\ R_2(r) &= B\tilde{P}(r). \end{aligned}$$

Podstawiając te zależności do (2.3.1), otrzymamy

$$\begin{aligned} \left[ \left( m + \kappa + \frac{1}{2} \right) A + \sqrt{\left( l + m + \frac{1}{2} \right) \left( l - m + \frac{1}{2} \right)} B \right] \tilde{P}(r) &= 0, \\ \left[ -\sqrt{\left( l - m + \frac{1}{2} \right) \left( l + m + \frac{1}{2} \right)} A + \left( m - \kappa - \frac{1}{2} \right) B \right] \tilde{P}(r) &= 0. \end{aligned}$$

Mamy do rozpatrzenia dwa przypadki:  $\kappa = l$  oraz  $\kappa = -l - 1$ .

Najpierw rozpatrzmy przypadek  $\kappa = l$ . Podstawiając tę zależność do (2.3.1) dostajemy:

$$\frac{A}{B} = -\sqrt{\frac{l - m + \frac{1}{2}}{l + m + \frac{1}{2}}}. \quad (2.3.2)$$

Zakładamy, że funkcja  $\tilde{P}(r)$  jest unormowana. Z warunku unormowania funkcji  $\Psi_A$  dostajemy, że  $A^2 + B^2 = 1$ . Wykorzystując związek (2.3.2), dostajemy

$$A = -\sqrt{\frac{l - m + \frac{1}{2}}{2l + 1}}, \quad B = \sqrt{\frac{l + m + \frac{1}{2}}{2l + 1}},$$

skąd wynika, że funkcję falową  $\Psi_A$  dla dodatnich  $\kappa$  można zapisać w postaci

$$\Psi_A^{(+)}(\mathbf{r}) = \tilde{P}(r) \begin{pmatrix} -\sqrt{\frac{l - m + \frac{1}{2}}{2l + 1}} Y_l^{m - \frac{1}{2}}(\hat{\mathbf{r}}) \\ \sqrt{\frac{l + m + \frac{1}{2}}{2l + 1}} Y_l^{m + \frac{1}{2}}(\hat{\mathbf{r}}) \end{pmatrix}.$$

$\Psi_A^{(+)}$  oznacza funkcję falową  $\Psi_A$  dla  $\kappa > 0$ . Często zapisujemy powyższy wzór jako

$$\Psi_A^{(+)}(\mathbf{r}) = \tilde{P}(r) \Omega_{\kappa^+ m}(\hat{\mathbf{r}}), \quad (2.3.3)$$

gdzie

$$\Omega_{\kappa^+ m}(\hat{\mathbf{r}}) = \begin{pmatrix} -\sqrt{\frac{l - m + \frac{1}{2}}{2l + 1}} Y_l^{m - \frac{1}{2}}(\hat{\mathbf{r}}) \\ \sqrt{\frac{l + m + \frac{1}{2}}{2l + 1}} Y_l^{m + \frac{1}{2}}(\hat{\mathbf{r}}) \end{pmatrix}$$

jest spinorem sferycznym dla  $\kappa > 0$ , co symbolizuje indeks  $\kappa^+$ .



Teraz rozpatrzmy przypadek  $\kappa < 0$ , tzn.  $\kappa = -l - 1$ . Podstawiając tę zależność do (2.3.1), dostajemy:

$$\frac{A}{B} = -\sqrt{\frac{l+m+\frac{1}{2}}{l-m+\frac{1}{2}}}.$$

Ponownie, korzystając z warunku unormowania funkcji  $\Psi_A$ , otrzymujemy:

$$A = \sqrt{\frac{l+m+\frac{1}{2}}{2l+1}}, \quad B = \sqrt{\frac{l-m+\frac{1}{2}}{2l+1}}.$$

Zatem funkcję falową możemy zapisać w tym przypadku następująco:

$$\Psi_A^{(-)}(\mathbf{r}) = \tilde{P}(r) \begin{pmatrix} \sqrt{\frac{l+m+\frac{1}{2}}{2l+1}} Y_l^{m-\frac{1}{2}}(\hat{\mathbf{r}}) \\ \sqrt{\frac{l-m+\frac{1}{2}}{2l+1}} Y_l^{m+\frac{1}{2}}(\hat{\mathbf{r}}) \end{pmatrix}.$$

$\Psi_A^{(-)}$  oznacza funkcję falową  $\Psi_A$  dla  $\kappa < 0$ . Zapisujemy powyższy wzór jako

$$\Psi_A^{(-)}(\mathbf{r}) = \tilde{P}(r) \Omega_{\kappa^- m}(\hat{\mathbf{r}}), \quad (2.3.4)$$

gdzie

$$\Omega_{\kappa^- m}(\hat{\mathbf{r}}) = \begin{pmatrix} \sqrt{\frac{l+m+\frac{1}{2}}{2l+1}} Y_l^{m-\frac{1}{2}}(\hat{\mathbf{r}}) \\ \sqrt{\frac{l-m+\frac{1}{2}}{2l+1}} Y_l^{m+\frac{1}{2}}(\hat{\mathbf{r}}) \end{pmatrix}$$

jest spinorem sferycznym dla  $\kappa < 0$ , co symbolizuje indeks  $\kappa^-$ . Ponieważ  $\tilde{P}(r)$  w ogólności zależy od  $\kappa$ , funkcje  $\tilde{P}(r)$  we wzorach (2.3.3), (2.3.4) nie są tymi samymi funkcjami.

Analogiczne obliczenia można przeprowadzić dla  $\Psi_B$ , otrzymując dla  $\kappa > 0$

$$\Psi_B^{(+)}(\mathbf{r}) = i \tilde{Q}(r) \Omega_{-\kappa^+ m}(\hat{\mathbf{r}}), \quad (2.3.5)$$

gdzie

$$\Omega_{-\kappa^+ m}(\hat{\mathbf{r}}) = \begin{pmatrix} \sqrt{\frac{l+m-\frac{1}{2}}{2l-1}} Y_{l-1}^{m-\frac{1}{2}}(\hat{\mathbf{r}}) \\ \sqrt{\frac{l-m-\frac{1}{2}}{2l-1}} Y_{l-1}^{m+\frac{1}{2}}(\hat{\mathbf{r}}) \end{pmatrix}.$$

Indeks  $-\kappa^+$  oznacza, że jest to spinor sferyczny dla składowej  $\Psi_B$  przy  $\kappa > 0$ .

Dla  $\kappa < 0$  otrzymujemy

$$\Psi_B^{(-)}(\mathbf{r}) = i\tilde{Q}(r)\Omega_{-\kappa^- m}(\hat{\mathbf{r}}), \quad (2.3.6)$$

gdzie

$$\Omega_{-\kappa^- m}(\hat{\mathbf{r}}) = \begin{pmatrix} -\sqrt{\frac{l-m+\frac{3}{2}}{2l+3}} Y_{l+1}^{m-\frac{1}{2}}(\hat{\mathbf{r}}) \\ \sqrt{\frac{l+m+\frac{3}{2}}{2l+3}} Y_{l+1}^{m+\frac{1}{2}}(\hat{\mathbf{r}}) \end{pmatrix}.$$

Indeks  $-\kappa^-$  oznacza, że jest to spinor sferyczny dla składowej  $\Psi_B$  przy  $\kappa < 0$ . Jednostka urojona w powyższych wzorach pojawiła się dla ułatwienia dalszych obliczeń.

Zapiszmy teraz radialne funkcje w postaci

$$\begin{aligned} \tilde{P}(r) &= \frac{1}{r} P(r), \\ \tilde{Q}(r) &= \frac{1}{r} Q(r). \end{aligned} \quad (2.3.7)$$

Możemy teraz zebrać poprzednie wyniki i zapisać je w tabeli.

$\kappa > 0, \quad j = l - \frac{1}{2}$	$\kappa < 0, \quad j = l + \frac{1}{2}$
$\Psi^{(+)}(\mathbf{r}) = \frac{1}{r} \begin{pmatrix} P_\kappa(r)\Omega_{\kappa^+ m}(\hat{\mathbf{r}}) \\ iQ_\kappa(r)\Omega_{-\kappa^+ m}(\hat{\mathbf{r}}) \end{pmatrix}$	$\Psi^{(-)}(\mathbf{r}) = \frac{1}{r} \begin{pmatrix} P_\kappa(r)\Omega_{\kappa^- m}(\hat{\mathbf{r}}) \\ iQ_\kappa(r)\Omega_{-\kappa^- m}(\hat{\mathbf{r}}) \end{pmatrix}$

Tabela przedstawia funkcje własne hamiltonianu Diraca w polu centralnym

## 2.4 Radialne równanie Diraca

Napiszmy równanie własne dla hamiltonianu Diraca w polu centralnym (2.2.1)

$$[c\boldsymbol{\alpha} \cdot \hat{\mathbf{p}} + \beta mc^2 + V(r)]\Psi = E\Psi.$$

Korzystając ze wzorów (2.1.3), powyższe równanie możemy wyrazić w następujący sposób:

$$\begin{aligned} c\boldsymbol{\sigma} \cdot \hat{\mathbf{p}}\Psi_B &= [E - mc^2 - V(r)]\Psi_A, \\ c\boldsymbol{\sigma} \cdot \hat{\mathbf{p}}\Psi_A &= [E + mc^2 - V(r)]\Psi_B, \end{aligned}$$

co jest równoważne zapisowi

$$\begin{aligned} \partial_z \Psi_3 + \partial_- \Psi_4 &= \frac{i}{c\hbar} (E - V - mc^2) \Psi_1, \\ \partial_+ \Psi_3 - \partial_z \Psi_4 &= \frac{i}{c\hbar} (E - V - mc^2) \Psi_2, \\ \partial_z \Psi_1 + \partial_- \Psi_2 &= \frac{i}{c\hbar} (E - V + mc^2) \Psi_3, \\ \partial_+ \Psi_1 - \partial_z \Psi_2 &= \frac{i}{c\hbar} (E - V + mc^2) \Psi_4. \end{aligned}$$

Symbole  $\partial_z$ ,  $\partial_+$ ,  $\partial_-$  zdefiniowane są w dodatku A.

Ponownie rozpatrujemy dwa przypadki:  $\kappa = l$  oraz  $\kappa = -l - 1$ .

Dla  $\kappa = l$  ze wzorów (2.3.3), (2.3.5), (2.3.7) oraz korzystając z zależności (A.2), (A.3) dla harmonik sferycznych, mamy

$$\begin{aligned} \frac{d}{dr} P(r) + \frac{l}{r} P(r) &= \frac{E + mc^2 - V(r)}{c\hbar} Q(r) \\ \frac{d}{dr} Q(r) - \frac{l}{r} Q(r) &= -\frac{E - mc^2 - V(r)}{c\hbar} P(r). \end{aligned}$$

Dla  $\kappa = -l - 1$  ze wzorów (2.3.4), (2.3.6), (2.3.7) oraz korzystając z zależności (A.2), (A.3) dla harmonik sferycznych, mamy

$$\begin{aligned} \frac{d}{dr} P(r) - \frac{l+1}{r} P(r) &= \frac{E + mc^2 - V(r)}{c\hbar} Q(r) \\ \frac{d}{dr} Q(r) + \frac{l+1}{r} Q(r) &= -\frac{E - mc^2 - V(r)}{c\hbar} P(r). \end{aligned}$$

Powyższe układy równań można zapisać w bardziej zwartej postaci

$$\begin{aligned} \frac{d}{dr} P(r) + \frac{\kappa}{r} P(r) &= \frac{E + mc^2 - V(r)}{c\hbar} Q(r) \\ \frac{d}{dr} Q(r) - \frac{\kappa}{r} Q(r) &= -\frac{E - mc^2 - V(r)}{c\hbar} P(r), \end{aligned} \tag{2.4.1}$$

lub w postaci macierzowej

$$\begin{pmatrix} mc^2 - E + V(r) & c\hbar\left(-\frac{d}{dr} + \frac{\kappa}{r}\right) \\ c\hbar\left(\frac{d}{dr} + \frac{\kappa}{r}\right) & -mc^2 - E + V(r) \end{pmatrix} \begin{pmatrix} P(r) \\ Q(r) \end{pmatrix} = 0.$$

Przedstawiając energię całkowitą  $E$  jako sumę energii kinetycznej  $\mathcal{E}$  oraz energii spoczynkowej  $mc^2$ , tzn.

$$E = \mathcal{E} + mc^2,$$

powyższe równanie można zapisać w postaci

$$\begin{pmatrix} -\mathcal{E} + V(r) & c\hbar\left(-\frac{d}{dr} + \frac{\kappa}{r}\right) \\ c\hbar\left(\frac{d}{dr} + \frac{\kappa}{r}\right) & -2mc^2 - \mathcal{E} + V(r) \end{pmatrix} \begin{pmatrix} P(r) \\ Q(r) \end{pmatrix} = 0.$$

### 3. Potencjalne rozpraszanie elektronów<sup>3</sup>

#### 3.1 Obliczanie różniczkowych przekrojów czynnych

Będziemy zajmować się relatywistycznym opisem rozpraszania elektronów na potencjałach centralnych zanikających szybciej niż  $r^{-2}$ . Propagację elektronu padającego będziemy opisywać jako propagację fali płaskiej rozchodzącej się w kierunku  $z$ . Asymptotycznie, funkcję opisującą elektron założymy w postaci złożenia fali płaskiej (od elektronu padającego) (por. 2.1) oraz rozchodzącej się fali sferycznej:

$$\Psi(\mathbf{r}) = \Phi(\mathbf{r}) + \Psi_f(\mathbf{r}) \quad \text{dla } r \rightarrow \infty. \quad (3.1.1)$$

$\Phi(\mathbf{r})$  jest falą płaską postaci

$$\Phi(\mathbf{r}) = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} e^{ikz},$$

natomiast  $\Psi_f(\mathbf{r})$  jest falą sferyczną

---

<sup>3</sup> Rozdział oparty jest o nie opublikowane notatki R. Szymtkowskiego [5] oraz pracę M. Krośnickiego [6].

$$\Psi_f(\mathbf{r}) = F(\hat{\mathbf{r}}) \frac{e^{ikr}}{r}.$$

Amplitudę rozpraszania  $F(\hat{\mathbf{r}})$  definiujemy następująco

$$F(\hat{\mathbf{r}}) = \begin{pmatrix} a'_1(\hat{\mathbf{r}}) \\ a'_2(\hat{\mathbf{r}}) \\ a'_3(\hat{\mathbf{r}}) \\ a'_4(\hat{\mathbf{r}}) \end{pmatrix}.$$

Oczywiście  $\hat{\mathbf{r}} = \mathbf{r}/r$  jest wektorem jednostkowym (wersorem) w kierunku  $\mathbf{r}$ . Składowe asymptotycznej postaci funkcji opisującej elektron (3.1.1) można przedstawić w postaci:

$$\Psi_\mu(\mathbf{r}) = a_\mu e^{ikz} + a'_\mu(\theta, \varphi) \frac{e^{ikr}}{r} \quad \text{przy } r \rightarrow \infty, \quad (3.1.2)$$

gdzie  $\mu = 1, 2, 3, 4$ . Pokażemy, że amplitudę  $F(\hat{\mathbf{r}})$  możemy przedstawić w postaci

$$F(\hat{\mathbf{r}}) = \begin{pmatrix} \chi_f(\hat{\mathbf{r}}) \\ \frac{c\boldsymbol{\sigma} \cdot \mathbf{p}_f}{E + mc^2} \chi_f(\hat{\mathbf{r}}) \end{pmatrix}, \quad (3.1.3)$$

gdzie  $\mathbf{p}_f$  jest wektorem pędu cząstki po rozproszeniu, dla którego zachodzi związek  $\mathbf{p}_f = \hbar \mathbf{k}_f$ , gdzie  $\mathbf{k}_f = \hat{\mathbf{r}}k$  jest wektorem falowym cząstki po rozproszeniu. Spinor  $\chi_f(\hat{\mathbf{r}})$  ma postać

$$\chi_f(\hat{\mathbf{r}}) = \begin{pmatrix} a'_1(\hat{\mathbf{r}}) \\ a'_2(\hat{\mathbf{r}}) \end{pmatrix}. \quad (3.1.4)$$

W obszarze, gdzie  $V(r)$  zanika, funkcja  $\Psi$  spełnia równanie dla cząstki swobodnej

$$(c\boldsymbol{\alpha} \cdot \hat{\mathbf{p}} + \beta mc^2 - E)\Psi(\mathbf{r}) = 0. \quad (3.1.5)$$

Podstawiając do powyższego równania funkcję  $\Psi(\mathbf{r})$  w postaci (3.1.1) i korzystając z tego, że funkcja  $\Phi(\mathbf{r})$  spełnia równanie własne dla cząstki swobodnej, otrzymujemy

$$(c\boldsymbol{\alpha} \cdot \hat{\mathbf{p}} + \beta mc^2 - E)F(\hat{\mathbf{r}}) \frac{e^{ikr}}{r} = 0.$$

Obliczmy pomocnicze wyrażenie:

$$\hat{\mathbf{p}} \frac{e^{ikr}}{r} = -i\hbar \frac{\mathbf{r}}{r} \frac{d}{dr} \left( \frac{e^{ikr}}{r} \right) \cong \mathbf{p}_f \frac{e^{ikr}}{r} \quad \text{przy } r \rightarrow \infty.$$

Korzystając z tego, możemy przepisać równanie (3.1.5) w postaci

$$\left( c\boldsymbol{\alpha} \cdot \mathbf{p}_f + \beta mc^2 - E \right) F(\hat{\mathbf{r}}) = 0. \quad (3.1.6)$$

Przedstawiając amplitudę  $F(\hat{\mathbf{r}})$  następująco

$$F(\hat{\mathbf{r}}) = \begin{pmatrix} \chi_f(\hat{\mathbf{r}}) \\ \chi'_f(\hat{\mathbf{r}}) \end{pmatrix}$$

i podstawiając ją do (3.1.6), otrzymujemy

$$\chi'_f(\hat{\mathbf{r}}) = \frac{c\boldsymbol{\sigma} \cdot \mathbf{p}_f}{E + mc^2} \chi_f(\hat{\mathbf{r}}),$$

co kończy dowód wzoru (3.1.3).

Spinor  $\Phi(\mathbf{r})$  opisujący cząstkę swobodną daje się przedstawić w analogicznej postaci

$$\Phi(\mathbf{r}) = \begin{pmatrix} \chi_i(\hat{\mathbf{r}}) \\ \frac{c\boldsymbol{\sigma} \cdot \mathbf{p}_i}{E + mc^2} \chi_i(\hat{\mathbf{r}}) \end{pmatrix} e^{ikz}, \quad (3.1.7)$$

gdzie  $\mathbf{p}_i$  jest wektorem pędu cząstki przed rozproszeniem. Funkcja  $\chi_i$  ma postać

$$\chi_i = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}. \quad (3.1.8)$$

W teorii Diraca gęstość prądu prawdopodobieństwa definiuje się następująco:

$$\mathbf{j} = c\Psi^\dagger \boldsymbol{\alpha} \Psi.$$

Wyprowadzimy teraz wzór na różniczkowy przekrój czynny. Definiuje się go jako stosunek liczby cząstek rozproszonych w jednostce czasu w jednostkowy kąt bryłowy do gęstości strumienia cząstek padających. Obliczmy najpierw liczb cząstek  $dN$  rozproszonych w kąt bryłowy  $d\Omega$  w kierunku  $\hat{\mathbf{n}}_f = \hat{\mathbf{r}}$

$$dN = \mathbf{j}_f \cdot d\mathbf{S} = c\Psi_f^\dagger \boldsymbol{\alpha} \cdot \hat{\mathbf{n}}_f \Psi_f r^2 d\Omega = cF^\dagger(\hat{\mathbf{r}}) \boldsymbol{\alpha} \cdot \hat{\mathbf{n}}_f F(\hat{\mathbf{r}}) d\Omega, \quad (3.1.9)$$

gdzie  $dS$  jest powierzchnią zorientowaną o polu infinitezymalnie małym i ustawioną prostopadle do wektora  $\hat{\mathbf{n}}_f$ , a  $\mathbf{j}_f$  jest gęstością prądu cząstek rozproszonych. Z drugiej strony,  $dN$  jest proporcjonalne do modułu wektora gęstości prądu cząstek padających  $\mathbf{j}_i$  oraz do kąta bryłowego  $d\Omega$ , a współczynnikiem proporcjonalności jest różniczkowy przekrój czynny  $\sigma(\hat{\mathbf{r}})$

$$dN = \sigma(\hat{\mathbf{r}}) \mathbf{j}_i \cdot \hat{\mathbf{n}}_i d\Omega, \quad (3.1.10)$$

gdzie  $\hat{\mathbf{n}}_i$  jest wersorem w kierunku padania. Rozpiszmy następujący iloczyn skalarny

$$\mathbf{j}_i \cdot \hat{\mathbf{n}}_i = c\Phi^\dagger(\hat{\mathbf{r}})\boldsymbol{\alpha} \cdot \hat{\mathbf{n}}_i\Phi(\hat{\mathbf{r}}).$$

Porównując wzory (3.1.9), (3.1.10) i korzystając z ostatniej równości, dostajemy

$$\sigma(\hat{\mathbf{r}}) = \frac{F^\dagger(\hat{\mathbf{r}})\boldsymbol{\alpha} \cdot \hat{\mathbf{n}}_f F(\hat{\mathbf{r}})}{\Phi^\dagger(\hat{\mathbf{r}})\boldsymbol{\alpha} \cdot \hat{\mathbf{n}}_i\Phi(\hat{\mathbf{r}})}. \quad (3.1.11)$$

Korzystając ze wzoru (3.1.3), przekształćmy licznik powyższego wyrażenia:

$$\begin{aligned} F^\dagger(\hat{\mathbf{r}})\boldsymbol{\alpha} \cdot \hat{\mathbf{n}}_f F(\hat{\mathbf{r}}) &= \begin{pmatrix} \chi_f^\dagger & \chi_f^\dagger c \frac{\boldsymbol{\sigma} \cdot \mathbf{p}_f}{E + mc^2} \end{pmatrix} \begin{pmatrix} 0 & \boldsymbol{\sigma} \cdot \hat{\mathbf{n}}_f \\ \boldsymbol{\sigma} \cdot \hat{\mathbf{n}}_f & 0 \end{pmatrix} \begin{pmatrix} \chi_f \\ c \frac{\boldsymbol{\sigma} \cdot \mathbf{p}_f}{E + mc^2} \chi_f \end{pmatrix} = \\ &= \chi_f^\dagger (\boldsymbol{\sigma} \cdot \hat{\mathbf{n}}_f) c \frac{\boldsymbol{\sigma} \cdot \mathbf{p}_f}{E + mc^2} \chi_f + \chi_f^\dagger c \frac{\boldsymbol{\sigma} \cdot \mathbf{p}_f}{E + mc^2} (\boldsymbol{\sigma} \cdot \hat{\mathbf{n}}_f) \chi_f. \end{aligned} \quad (3.1.12)$$

Korzystając z własności  $(\boldsymbol{\sigma} \cdot \mathbf{A})(\boldsymbol{\sigma} \cdot \mathbf{B}) = \mathbf{A} \cdot \mathbf{B} + i\boldsymbol{\sigma} \cdot (\mathbf{A} \times \mathbf{B})$  dla dowolnych, komutujących z  $\boldsymbol{\sigma}$  operatorów  $\mathbf{A}$  i  $\mathbf{B}$ , można udowodnić związek:

$$(\boldsymbol{\sigma} \cdot \hat{\mathbf{n}}_f)(\boldsymbol{\sigma} \cdot \mathbf{p}_f) = (\boldsymbol{\sigma} \cdot \mathbf{p}_f)(\boldsymbol{\sigma} \cdot \hat{\mathbf{n}}_f) = p_f.$$

Wykorzystując go w (3.1.12), otrzymujemy

$$F^\dagger(\hat{\mathbf{r}})\boldsymbol{\alpha} \cdot \hat{\mathbf{n}}_f F(\hat{\mathbf{r}}) = \frac{2cp_f}{E + mc^2} \chi_f^\dagger \chi_f.$$

Analogiczne przekształcenia można przeprowadzić dla mianownika wzoru (3.1.11), otrzymując

$$\Phi^\dagger(\hat{\mathbf{r}})\boldsymbol{\alpha} \cdot \hat{\mathbf{n}}_i\Phi(\hat{\mathbf{r}}) = \frac{2cp_i}{E + mc^2} \chi_i^\dagger \chi_i.$$

Korzystając z faktu, że dla rozpraszania sprężystego zachodzi  $p_i = p_f = p$  i po podstawieniu powyższych obliczeń do (3.1.11), otrzymujemy

$$\sigma(\hat{r}) = \frac{\chi_f^\dagger \chi_f}{\chi_i^\dagger \chi_i}.$$

Wykorzystując związki (3.1.4), (3.1.8), można przepisać wzór na różniczkowy przekrój czynny w postaci

$$\sigma(\theta, \varphi) = \frac{|a_1'|^2 + |a_2'|^2}{|a_1|^2 + |a_2|^2}. \quad (3.1.13)$$

### 3.2 Metoda przesunięć fazowych

Przepiszmy układ równań (2.4.1) w postaci

$$\begin{aligned} \frac{d}{dr} P_\kappa(r) + \frac{\kappa}{r} P_\kappa(r) &= \frac{E_+ - V}{c\hbar} Q_\kappa(r), \\ \frac{d}{dr} Q_\kappa(r) - \frac{\kappa}{r} Q_\kappa(r) &= -\frac{E_- - V}{c\hbar} P_\kappa(r), \end{aligned} \quad (3.2.1)$$

gdzie  $E_+ = E + mc^2$ ,  $E_- = E - mc^2$ . W przypadku cząstki swobodnej (tzn.  $V = 0$ ) powyższy układ równań można przepisać w postaci

$$\frac{d}{dr} \tilde{P}_\kappa(r) + \frac{\kappa}{r} \tilde{P}_\kappa(r) = \frac{E_+}{c\hbar} \tilde{Q}_\kappa(r), \quad (3.2.2)$$

$$\frac{d}{dr} \tilde{Q}_\kappa(r) - \frac{\kappa}{r} \tilde{Q}_\kappa(r) = -\frac{E_-}{c\hbar} \tilde{P}_\kappa(r). \quad (3.2.3)$$

Różniczkując po  $r$  oraz przenosząc wszystkie człony na jedną stronę równania, dostajemy

$$\frac{d^2}{dr^2} \tilde{P}_\kappa(r) - \frac{\kappa}{r^2} \tilde{P}_\kappa(r) + \frac{\kappa}{r} \frac{d}{dr} \tilde{P}_\kappa(r) - \frac{E_+}{c\hbar} \frac{d}{dr} \tilde{Q}_\kappa(r) = 0, \quad (3.2.4)$$

$$\frac{d^2}{dr^2} \tilde{Q}_\kappa(r) + \frac{\kappa}{r^2} \tilde{Q}_\kappa(r) - \frac{\kappa}{r} \frac{d}{dr} \tilde{Q}_\kappa(r) + \frac{E_-}{c\hbar} \frac{d}{dr} \tilde{P}_\kappa(r) = 0. \quad (3.2.5)$$

Z równań (3.2.2), (3.2.3) wyznaczamy odpowiednio  $\frac{d\tilde{Q}_\kappa(r)}{dr}$  oraz  $\frac{d\tilde{P}_\kappa(r)}{dr}$  i podstawiamy odpowiednio do równań (3.2.4), (3.2.5). Po tych operacjach otrzymujemy

$$\frac{d^2}{dr^2} \tilde{P}_\kappa(r) - \frac{\kappa}{r^2} \tilde{P}_\kappa(r) + \frac{\kappa}{r} \left( \frac{d}{dr} \tilde{P}_\kappa(r) - \frac{E_+}{c\hbar} \tilde{Q}_\kappa(r) \right) + \frac{E_+ E_-}{(c\hbar)^2} \tilde{P}_\kappa(r) = 0,$$

$$\frac{d^2}{dr^2} \tilde{Q}_\kappa(r) + \frac{\kappa}{r^2} \tilde{Q}_\kappa(r) - \frac{\kappa}{r} \left( \frac{d}{dr} \tilde{Q}_\kappa(r) + \frac{E_-}{c\hbar} \tilde{P}_\kappa(r) \right) + \frac{E_- E_+}{(c\hbar)^2} \tilde{Q}_\kappa(r) = 0.$$



Wykorzystując w powyższych równaniach odpowiednio równania (3.2.2), (3.2.3), otrzymujemy

$$\begin{aligned} \frac{d^2}{dr^2} \tilde{P}_\kappa(r) - \frac{\kappa(\kappa+1)}{r^2} \tilde{P}_\kappa(r) + k^2 \tilde{P}_\kappa(r) &= 0, \\ \frac{d^2}{dr^2} \tilde{Q}_\kappa(r) - \frac{\kappa(\kappa-1)}{r^2} \tilde{Q}_\kappa(r) + k^2 \tilde{Q}_\kappa(r) &= 0, \end{aligned} \quad (3.2.6)$$

gdzie  $k^2$  oznacza

$$k^2 = k_- k_+ = \frac{E_- E_+}{(c\hbar)^2},$$

gdzie z kolei  $k_- = \frac{E_-}{c\hbar}$ ,  $k_+ = \frac{E_+}{c\hbar}$ .

Równania (3.2.6) są równaniami typu Riccati-Bessela, dzięki czemu  $\tilde{P}_\kappa$  możemy przedstawić jako superpozycję funkcji Riccati-Bessela:

$$\tilde{P}_\kappa(r) = A \hat{j}_l(kr) + B \hat{y}_l(kr). \quad (3.2.7)$$

Podstawiając to równanie do (3.2.2) otrzymujemy

$$\tilde{Q}_\kappa(r) = \sqrt{\frac{k_-}{k_+}} \left\{ A \left[ \hat{j}'_l(kr) + \frac{\kappa}{kr} \hat{j}_l(kr) \right] + B \left[ \hat{y}'_l(kr) + \frac{\kappa}{kr} \hat{y}_l(kr) \right] \right\},$$

gdzie prim oznacza różniczkowanie po  $kr$ .

Korzystając z własności rekurencyjnych dla funkcji Riccati-Bessela (B.1), (B.2) otrzymujemy

$$\tilde{Q}_\kappa(r) = \sqrt{\frac{k_-}{k_+}} \left[ A \hat{j}_{l-1}(kr) + B \hat{y}_{l-1}(kr) \right] \quad \text{dla } \kappa > 0, \quad (3.2.8)$$

$$\tilde{Q}_\kappa(r) = -\sqrt{\frac{k_-}{k_+}} \left[ A \hat{j}_{l+1}(kr) + B \hat{y}_{l+1}(kr) \right] \quad \text{dla } \kappa < 0. \quad (3.2.9)$$

Przy założeniu, że potencjał  $V(r)$  w równaniach (3.2.1) dostatecznie szybko zmierza do zera przy  $r \rightarrow \infty$ , powinno zachodzić

$$\begin{aligned} P_\kappa(r) &\rightarrow \tilde{P}_\kappa(r), \\ Q_\kappa(r) &\rightarrow \tilde{Q}_\kappa(r) \end{aligned}$$

ze współczynnikami zależnymi od postaci potencjału  $V(r)$ . Przepiszmy powyższe równania następująco:

$$P_\kappa(r) \rightarrow C \left[ \hat{j}_l(kr) \cos \delta_\kappa - \hat{y}_l(kr) \sin \delta_\kappa \right] \quad \text{przy } r \rightarrow \infty,$$

$$Q_\kappa(r) \rightarrow \sqrt{\frac{k_-}{k_+}} C [\hat{j}_{l-1}(kr) \cos \delta_\kappa - \hat{y}_{l-1}(kr) \sin \delta_\kappa] \text{ przy } r \rightarrow \infty \text{ dla } \kappa > 0,$$

$$Q_\kappa(r) \rightarrow -\sqrt{\frac{k_-}{k_+}} C [\hat{j}_{l+1}(kr) \cos \delta_\kappa - \hat{y}_{l+1}(kr) \sin \delta_\kappa] \text{ przy } r \rightarrow \infty \text{ dla } \kappa < 0,$$

gdzie  $\delta_\kappa$  jest przesunięciem fazowym. Porównując te warunki z (3.2.7), (3.2.8), (3.2.9), można zauważyć, że współczynniki  $A, B$  powinny być postaci

$$A = C \cos \delta_\kappa,$$

$$B = C \sin \delta_\kappa.$$

Dla bardzo dużych  $r$ , korzystając z asymptotycznej postaci funkcji Riccati-Bessela, funkcje  $P_\kappa(r), Q_\kappa(r)$  możemy zapisać w postaci

$$P_\kappa(r) \xrightarrow{r \rightarrow \infty} C \sin\left(kr - \frac{\pi l}{2} + \delta_\kappa\right)$$

oraz

$$Q_\kappa(r) \xrightarrow{r \rightarrow \infty} \sqrt{\frac{k_-}{k_+}} C \sin\left(kr - \frac{\pi(l-1)}{2} + \delta_\kappa\right) \quad \text{dla } \kappa > 0,$$

$$Q_\kappa(r) \xrightarrow{r \rightarrow \infty} -\sqrt{\frac{k_-}{k_+}} C \sin\left(kr - \frac{\pi(l+1)}{2} + \delta_\kappa\right) \quad \text{dla } \kappa < 0.$$

Powyższe wzory można zapisać w bardziej zwartej postaci

$$P_\kappa(r) \xrightarrow{r \rightarrow \infty} C \sin\left(kr - \frac{\pi l}{2} + \delta_\kappa\right), \quad (3.2.10)$$

$$Q_\kappa(r) \xrightarrow{r \rightarrow \infty} \sqrt{\frac{k_-}{k_+}} C \cos\left(kr - \frac{\pi l}{2} + \delta_\kappa\right).$$

### 3.3 Metoda fal parcjalnych

Na mocy wzorów (3.1.3), (3.1.7) widzimy, że aby otrzymać wszelkie informacje o rozpraszaniu, wystarczy rozpatrywać tylko dużą składową funkcji  $\Psi$ . Duża składowa fali padającej może zostać rozwinięta w bazie sferycznych funkcji Bessela  $j_l(kr)$  i wielomianów Legendre'a  $P_l(\cos\theta)$  w następujący sposób:

$$\Phi_d(\mathbf{r}) = \begin{pmatrix} a_{1/2} \\ a_{-1/2} \end{pmatrix} e^{ikz} = \sum_{l=0}^{\infty} (2l+1) i^l j_l(kr) P_l(\cos\theta) \begin{pmatrix} a_{1/2} \\ a_{-1/2} \end{pmatrix}.$$

Wielomiany Legendre'a można wyrazić przez harmoniki sferyczne wzorem

$$P_l(\cos\theta) = \sqrt{\frac{4\pi}{2l+1}} Y_l^0(\hat{\mathbf{r}}).$$

Podstawmy ostatni wzór do przedostatniego, otrzymując:

$$\Phi_d(\mathbf{r}) = \sum_{l=0}^{\infty} i^l \sqrt{4\pi(2l+1)} j_l(kr) Y_l^0(\hat{\mathbf{r}}) \begin{pmatrix} a_{1/2} \\ a_{-1/2} \end{pmatrix}. \quad (3.3.1)$$

Wprowadźmy dla funkcji opisujących spin elektronu w kierunku osi  $z$  oraz  $-z$  oznaczenia, odpowiednio

$$\chi_{1/2}^{+1/2} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \chi_{1/2}^{-1/2} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Wykorzystując tę notację, wzór (3.3.1) zapisujemy w postaci

$$\Phi_d(\mathbf{r}) = \sum_{l=0}^{\infty} i^l \sqrt{4\pi(2l+1)} j_l(kr) Y_l^0(\hat{\mathbf{r}}) \sum_{m=\mp 1/2} a_m \chi_s^m, \quad (3.3.2)$$

gdzie  $s = 1/2$ . Przepiszmy powyższy wzór w postaci

$$\Phi_d(\mathbf{r}) = \sum_{l=0}^{\infty} i^l \sqrt{4\pi(2l+1)} j_l(kr) \sum_{m=\mp 1/2} a_m Y_l^0(\hat{\mathbf{r}}) \chi_s^m.$$

Iloczyn  $Y_l^0(\hat{\mathbf{r}}) \chi_s^m$  rozwijamy w bazie spinorów sferycznych  $\Omega_{jlm}$  następująco

$$Y_l^0(\hat{\mathbf{r}}) \chi_s^m = \sum_{j=|l-1/2|}^{l+1/2} \langle l s 0 m | j m \rangle \Omega_{jlm}(\hat{\mathbf{r}}),$$

gdzie współczynniki  $\langle l s 0 m | j m \rangle$  są tzw. współczynnikami Clebscha-Gordana, przy czym w powyższym wzorze skorzystaliśmy z faktu, że  $m_l + m_s = m$ , a  $m_l = 0$ . Dla współczynników Clebscha-Gordana stosujemy oznaczenie  $\langle l s m_l m_s | j m \rangle$ , gdzie liczby kwantowe po lewej stronie to odpowiednio

- $l$  – liczba kwantowa orbitalnego momentu pędu,
- $m_l$  – rzut orbitalnego momentu pędu na wyróżnioną oś,
- $s$  – wartość spinu
- $m_s$  – rzut spinu na wyróżnioną oś.

W powyższym rozwinięciu dla każdego  $l \neq 0$  otrzymujemy dwa wyrazy z  $j = l - 1/2$  oraz  $j = l + 1/2$ , natomiast dla  $l = 0$  tylko jeden wyraz z  $j = 1/2$ . Podstawiając to rozwinięcie do (3.3.2) otrzymujemy

$$\Phi_d(\mathbf{r}) = \sum_{j=|l-1/2|}^{j+1/2} \sum_{l=0}^{\infty} i^l \sqrt{4\pi(2l+1)} j_l(kr) \sum_{m=\mp 1/2} a_m \langle l s 0 m | j m \rangle \Omega_{jm}(\hat{\mathbf{r}}). \quad (3.3.3)$$

Zamiast używać pary liczb kwantowych  $j$  i  $l$ , będziemy używać jednej  $\kappa$  (por. rozdział 2.2), gdyż wyznacza ona jednoznacznie zarówno  $j$

$$j = |\kappa| - \frac{1}{2},$$

jak i  $l$

$$l = \begin{cases} \kappa & \text{dla } \kappa > 0 \\ -\kappa - 1 & \text{dla } \kappa < 0. \end{cases}$$

We wzorze (3.3.3) zamiast dwóch sumowań po  $j$  i po  $l$ , możemy napisać jedną sumę po  $\kappa$

$$\Phi_d(\mathbf{r}) = \sum_{\kappa=-\infty}^{\infty} i^l \sqrt{4\pi(2l+1)} j_l(kr) \sum_{m=\mp 1/2} a_m \langle l s 0 m | j m \rangle \Omega_{\kappa m}(\hat{\mathbf{r}}). \quad (3.3.4)$$

Korzystając z symetrii problemu rozproszeniowego (pole centralne) oraz ze sposobu rozwinięcia dla cząstki swobodnej, dużą składową funkcji falowej  $\Psi_d$  dla elektronu rozproszonego można przedstawić w postaci

$$\Psi_d(\mathbf{r}) = \sum_{\kappa=-\infty}^{\infty} i^l \sqrt{4\pi(2l+1)} \frac{P_{\kappa}(r)}{r} \sum_{m=\mp 1/2} a_m \langle l s 0 m | j m \rangle \Omega_{\kappa m}(\hat{\mathbf{r}}). \quad (3.3.5)$$

Daleko od centrum rozproszeniowego różnica funkcji  $\Psi_d - \Phi_d$  powinna opisywać falę sferyczną rozchodzącą się od centrum (por. wzór (3.1.2)).

$$\Psi_d - \Phi_d \rightarrow F_d(\hat{\mathbf{r}}) \frac{e^{i\kappa r}}{r} \quad \text{przy } r \rightarrow \infty. \quad (3.3.6)$$

Ze względu na postać funkcji  $\Psi_d$  oraz  $\Phi_d$ , amplitudę  $F_d(\hat{\mathbf{r}})$  postulujemy w postaci

$$F_d(\hat{\mathbf{r}}) = \sum_{\kappa=-\infty}^{\infty} \sqrt{4\pi(2l+1)} \mathcal{F}_{\kappa} \sum_{m=\mp 1/2} a_m \langle l s 0 m | j m \rangle \Omega_{\kappa m}(\hat{\mathbf{r}}), \quad (3.3.7)$$

gdzie  $\mathcal{F}_{\kappa}$  jest współczynnikiem do wyznaczenia. Podstawiając wzory (3.3.4), (3.3.5) do (3.3.6), otrzymujemy

$$\Psi_d(\mathbf{r}) - \Phi_d(\mathbf{r}) = \sum_{\kappa=-\infty}^{\infty} i^l \sqrt{4\pi(2l+1)} \frac{P_{\kappa}(r) - r j_l(kr)}{r} \sum_{m=\mp 1/2} a_m \langle l s 0 m | j m \rangle \Omega_{\kappa m}(\hat{\mathbf{r}}). \quad (3.3.8)$$

Porównując wzory (3.3.6), (3.3.7), (3.3.8) otrzymujemy

$$i^l \frac{P_\kappa(r) - rj_l(kr)}{r} \rightarrow \mathcal{F}_\kappa \frac{e^{ikr}}{r} \quad \text{przy } r \rightarrow \infty. \quad (3.3.9)$$

Skorzystamy teraz z asymptotycznej postaci radialnej funkcji falowej (3.2.10)

$$P_\kappa(r) \xrightarrow{r \rightarrow \infty} A_\kappa \sin\left(kr - \frac{\pi l}{2} + \delta_\kappa\right)$$

oraz sferycznej funkcji Bessela

$$rj_l(kr) \xrightarrow{r \rightarrow \infty} \frac{1}{k} \sin\left(kr - \frac{\pi l}{2}\right).$$

Podstawiając te wzory do (3.3.9) otrzymamy

$$i^l \frac{P_\kappa(r) - rj_l(kr)}{r} \xrightarrow{r \rightarrow \infty} \frac{i^l}{2ir} \left[ \left( A_\kappa e^{i\delta_\kappa} - \frac{1}{k} \right) e^{i(kr - \pi l/2)} - \left( A_\kappa e^{-i\delta_\kappa} - \frac{1}{k} \right) e^{-i(kr - \pi l/2)} \right]. \quad (3.3.10)$$

Ponieważ zachodzi  $e^{i\pi l/2} = i^l$ , równanie (3.3.10) możemy przepisać w postaci

$$i^l \frac{P_\kappa(r) - rj_l(kr)}{r} \xrightarrow{r \rightarrow \infty} \frac{1}{2i} \left( A_\kappa e^{i\delta_\kappa} - \frac{1}{k} \right) \frac{e^{ikr}}{r} - \frac{(-)^l}{2i} \left( A_\kappa e^{-i\delta_\kappa} - \frac{1}{k} \right) \frac{e^{-ikr}}{r}.$$

Porównując ostatni wzór z (3.3.9) widzimy, że współczynnik stojący przy  $e^{-ikr}$  powinien być równy zeru, otrzymujemy więc

$$A_\kappa = \frac{1}{k} e^{i\delta_\kappa}$$

oraz

$$\mathcal{F}_\kappa = \frac{1}{2i} \left( A_\kappa e^{i\delta_\kappa} - \frac{1}{k} \right).$$

Po podstawieniu do tego wzoru wyrażenia na  $A_\kappa$  otrzymujemy

$$\mathcal{F}_\kappa = \frac{1}{2ik} (e^{2i\delta_\kappa} - 1).$$

Podsumowując, wzór na amplitudę rozpraszania możemy zapisać w postaci

$$F_d(\hat{r}) = \frac{1}{2ik} \sum_{\kappa=-\infty}^{\infty} \sqrt{4\pi(2l+1)} (e^{2i\delta_\kappa} - 1) \sum_{m=\mp 1/2} a_m \langle l s 0 m | j m \rangle \Omega_{\kappa m}(\hat{r}). \quad (3.3.11)$$

Przedstawimy teraz amplitudę rozproszeniową następująco

$$F_d(\hat{\mathbf{r}}) = M(\hat{\mathbf{r}}) \begin{pmatrix} a_{1/2} \\ a_{-1/2} \end{pmatrix}, \quad (3.3.12)$$

gdzie  $M(\hat{\mathbf{r}})$  jest macierzą  $2 \times 2$ . Powyższy wzór przepiszemy w postaci

$$F_d(\hat{\mathbf{r}}) = \sum_{m=\mp 1/2} \sum_{m'=\mp 1/2} a_m M_{m'm}(\hat{\mathbf{r}}) \chi_s^{m'}. \quad (3.3.13)$$

Zauważmy, że spinor sferyczny  $\Omega_{\kappa m}(\hat{\mathbf{r}})$  można rozwinąć następująco

$$\Omega_{\kappa m}(\hat{\mathbf{r}}) = \sum_{m'=\mp 1/2} \langle l s (m-m') m' | j m \rangle Y_l^{m-m'}(\hat{\mathbf{r}}) \chi_s^{m'},$$

gdzie wykorzystano fakt zachodzenia równości  $m = m' + m_l$ . Podstawiając powyższy wzór do (3.3.11), dostaniemy

$$F_d(\hat{\mathbf{r}}) = \frac{1}{2ik} \sum_{k=-\infty}^{\infty} \sqrt{4\pi(2l+1)} (e^{2i\delta_k} - 1) \sum_{m=\mp 1/2} \sum_{m'=\mp 1/2} a_m \langle l s 0 m | j m \rangle \langle l s (m-m') m' | j m \rangle Y_l^{m-m'}(\hat{\mathbf{r}}) \chi_s^{m'}.$$

Zmieniając w powyższym wzorze kolejność sumowania i porównując go ze wzorem (3.3.13) widzimy, że musi zachodzić

$$M_{m'm}(\hat{\mathbf{r}}) = \frac{1}{2ik} \sum_{\kappa=-\infty}^{\infty} \sqrt{4\pi(2l+1)} (e^{2i\delta_\kappa} - 1) \langle l s 0 m | j m \rangle \langle l s (m-m') m' | j m \rangle Y_l^{m-m'}(\hat{\mathbf{r}}). \quad (3.3.14)$$

znak $\kappa$	$j$	$m' = 1/2$	$m' = -1/2$
$\kappa < 0$	$l + \frac{1}{2}$	$\sqrt{\frac{l+m+1/2}{2l+1}}$	$\sqrt{\frac{l-m+1/2}{2l+1}}$
$\kappa > 0$	$l - \frac{1}{2}$	$-\sqrt{\frac{l-m+1/2}{2l+1}}$	$\sqrt{\frac{l+m+1/2}{2l+1}}$

Tabela przedstawia współczynniki Clebscha-Gordana  $\left\langle l \frac{1}{2} (m-m') m' \middle| j m \right\rangle$

Posługując się powyższą tabelą, obliczymy iloczyny współczynników Clebscha-Gordana i w ten sposób znajdziemy jawną postać wyrazów macierzy  $M(\hat{\mathbf{r}})$ . Rozpatrzmy przypadki

1. Gdy  $m' = m = \frac{1}{2}$

- Dla  $\kappa > 0$

$$\left\langle l \frac{1}{2} 0 \frac{1}{2} \middle| j \frac{1}{2} \right\rangle \left\langle l \frac{1}{2} 0 \frac{1}{2} \middle| j \frac{1}{2} \right\rangle = \left( -\sqrt{\frac{l-1/2+1/2}{2l+1}} \right)^2 = \frac{l}{2l+1} = \frac{|\kappa|}{2l+1}.$$

- Dla  $\kappa < 0$

$$\left\langle l \frac{1}{2} 0 \frac{1}{2} \middle| j \frac{1}{2} \right\rangle \left\langle l \frac{1}{2} 0 \frac{1}{2} \middle| j \frac{1}{2} \right\rangle = \left( \sqrt{\frac{l+1/2+1/2}{2l+1}} \right)^2 = \frac{l+1}{2l+1} = \frac{|\kappa|}{2l+1}.$$

Ponieważ zachodzi wzór

$$Y_l^0(\hat{r}) = \sqrt{\frac{2l+1}{4\pi}} P_l(\cos\theta), \quad (3.3.15)$$

wzór (3.3.14) możemy zapisać następująco

$$M_{\frac{1}{2} \frac{1}{2}}(\hat{r}) = \frac{1}{2ik} \sum_{\kappa=-\infty}^{\infty} |\kappa| (e^{2i\delta_\kappa} - 1) P_\kappa(\cos\theta).$$

2. Gdy  $m' = m = -\frac{1}{2}$

- Dla  $\kappa > 0$

$$\left\langle l \frac{1}{2} 0 -\frac{1}{2} \middle| j -\frac{1}{2} \right\rangle \left\langle l \frac{1}{2} 0 -\frac{1}{2} \middle| j -\frac{1}{2} \right\rangle = \left( \sqrt{\frac{l-1/2+1/2}{2l+1}} \right)^2 = \frac{l}{2l+1} = \frac{|\kappa|}{2l+1}.$$

- Dla  $\kappa < 0$

$$\left\langle l \frac{1}{2} 0 -\frac{1}{2} \middle| j -\frac{1}{2} \right\rangle \left\langle l \frac{1}{2} 0 -\frac{1}{2} \middle| j -\frac{1}{2} \right\rangle = \left( \sqrt{\frac{l+1/2+1/2}{2l+1}} \right)^2 = \frac{l+1}{2l+1} = \frac{|\kappa|}{2l+1}.$$

Korzystając z powyższych przekształceń oraz wzoru (3.3.15), wzór (3.3.14) możemy zapisać następująco

$$M_{\frac{1}{2} -\frac{1}{2}}(\hat{r}) = \frac{1}{2ik} \sum_{\kappa=-\infty}^{\infty} |\kappa| (e^{2i\delta_\kappa} - 1) P_\kappa(\cos\theta).$$

3. Gdy  $m' = \frac{1}{2}$ ,  $m = -\frac{1}{2}$

- Dla  $\kappa > 0$

$$\left\langle l \frac{1}{2} 0 -\frac{1}{2} \middle| j -\frac{1}{2} \right\rangle \left\langle l \frac{1}{2} -1 \frac{1}{2} \middle| j -\frac{1}{2} \right\rangle = -\sqrt{\frac{l(l+1)}{(2l+1)^2}} = -\operatorname{sgn} \kappa \sqrt{\frac{l(l+1)}{(2l+1)^2}}.$$

- Dla  $\kappa < 0$

$$\left\langle l \frac{1}{2} \ 0 \ -\frac{1}{2} \middle| j \ -\frac{1}{2} \right\rangle \left\langle l \frac{1}{2} \ -1 \ \frac{1}{2} \middle| j \ -\frac{1}{2} \right\rangle = \sqrt{\frac{l(l+1)}{(2l+1)^2}} = -\operatorname{sgn} \kappa \sqrt{\frac{l(l+1)}{(2l+1)^2}}.$$

Dla funkcji sferycznych zachodzi związek

$$Y_l^{-1}(\hat{\mathbf{r}}) = -Y_l^{1*}(\hat{\mathbf{r}}) = \sqrt{\frac{2l+1}{4\pi l(l+1)}} P_l^1(\cos\theta) e^{-i\varphi}.$$

Podstawiając go oraz obliczone współczynniki Clebscha-Gordana do (3.3.14), otrzymujemy

$$M_{\frac{1}{2} \ -\frac{1}{2}}(\hat{\mathbf{r}}) = -\frac{1}{2ik} \sum_{\kappa=-\infty}^{\infty} \operatorname{sgn} \kappa (e^{2i\delta_\kappa} - 1) P_\kappa^1(\cos\theta) e^{-i\varphi}.$$

Przepiszmy ten wzór w postaci

$$M_{\frac{1}{2} \ -\frac{1}{2}}(\hat{\mathbf{r}}) = -\frac{1}{2ik} \sum_{\kappa=-\infty}^{\infty} \operatorname{sgn} \kappa e^{2i\delta_\kappa} P_\kappa^1(\cos\theta) e^{-i\varphi} + \frac{1}{2ik} \sum_{\kappa=-\infty}^{\infty} \operatorname{sgn} \kappa P_\kappa^1(\cos\theta) e^{-i\varphi}.$$

Po skorzystaniu w powyższym wzorze z relacji  $P_{-l-1}^m(\cos\theta) = P_l^m(\cos\theta)$  zauważamy, że jego drugi człon zeruje się, dzięki czemu dostajemy ostatecznie

$$M_{\frac{1}{2} \ -\frac{1}{2}}(\hat{\mathbf{r}}) = -\frac{1}{2ik} \sum_{\kappa=-\infty}^{\infty} \operatorname{sgn} \kappa e^{2i\delta_\kappa} P_\kappa^1(\cos\theta) e^{-i\varphi}.$$

4. Gdy  $m' = -\frac{1}{2}$ ,  $m = \frac{1}{2}$

- Dla  $\kappa > 0$

$$\left\langle l \frac{1}{2} \ 0 \ \frac{1}{2} \middle| j \ \frac{1}{2} \right\rangle \left\langle l \frac{1}{2} \ 1 \ -\frac{1}{2} \middle| j \ \frac{1}{2} \right\rangle = -\sqrt{\frac{l(l+1)}{(2l+1)^2}} = -\operatorname{sgn} \kappa \sqrt{\frac{l(l+1)}{(2l+1)^2}}.$$

- Dla  $\kappa < 0$

$$\left\langle l \frac{1}{2} \ 0 \ \frac{1}{2} \middle| j \ \frac{1}{2} \right\rangle \left\langle l \frac{1}{2} \ 1 \ -\frac{1}{2} \middle| j \ \frac{1}{2} \right\rangle = \sqrt{\frac{l(l+1)}{(2l+1)^2}} = -\operatorname{sgn} \kappa \sqrt{\frac{l(l+1)}{(2l+1)^2}}.$$

Dla funkcji sferycznych zachodzi związek

$$Y_l^1(\hat{\mathbf{r}}) = -\sqrt{\frac{2l+1}{4\pi l(l+1)}} P_l^1(\cos\theta) e^{i\varphi}.$$

Podstawiając go (oraz obliczone współczynniki Clebscha-Gordana) do (3.3.14) dostajemy



$$M_{-\frac{1}{2} \frac{1}{2}}(\hat{\mathbf{r}}) = \frac{1}{2ik} \sum_{\kappa=-\infty}^{\infty} \operatorname{sgn} \kappa (e^{2i\delta_{\kappa}} - 1) P_{\kappa}^1(\cos\theta) e^{i\varphi}.$$

Przepiszmy powyższy wzór w postaci

$$M_{-\frac{1}{2} \frac{1}{2}}(\hat{\mathbf{r}}) = \frac{1}{2ik} \sum_{\kappa=-\infty}^{\infty} \operatorname{sgn} \kappa e^{2i\delta_{\kappa}} P_{\kappa}^1(\cos\theta) e^{i\varphi} - \frac{1}{2ik} \sum_{\kappa=-\infty}^{\infty} \operatorname{sgn} \kappa P_{\kappa}^1(\cos\theta) e^{i\varphi}.$$

Tutaj również drugi człon zeruje się, dzięki czemu dostajemy ostatecznie

$$M_{-\frac{1}{2} \frac{1}{2}}(\hat{\mathbf{r}}) = \frac{1}{2ik} \sum_{\kappa=-\infty}^{\infty} \operatorname{sgn} \kappa e^{2i\delta_{\kappa}} P_{\kappa}^1(\cos\theta) e^{i\varphi}.$$

Wprowadzając oznaczenia

$$f(\theta) = \frac{1}{2ik} \sum_{\kappa=-\infty}^{\infty} |\kappa| (e^{2i\delta_{\kappa}} - 1) P_{\kappa}(\cos\theta), \quad (3.3.16)$$

$$g(\theta) = \frac{1}{2ik} \sum_{\kappa=-\infty}^{\infty} \operatorname{sgn} \kappa e^{2i\delta_{\kappa}} P_{\kappa}^1(\cos\theta), \quad (3.3.17)$$

macierz  $M(\hat{\mathbf{r}})$  można zapisać następująco

$$M(\hat{\mathbf{r}}) = \begin{pmatrix} f(\theta) & -g(\theta) e^{-i\varphi} \\ g(\theta) e^{i\varphi} & f(\theta) \end{pmatrix}. \quad (3.3.18)$$

Wzory (3.3.16), (3.3.17) można zapisać w postaci

$$f(\theta) = \frac{1}{2ik} \sum_{l=0}^{\infty} [(l+1)(e^{2i\delta_{-l-1}} - 1) + l(e^{2i\delta_l} - 1)] P_l(\cos\theta),$$

$$g(\theta) = \frac{1}{2ik} \sum_{l=1}^{\infty} [e^{2i\delta_l} - e^{2i\delta_{-l-1}}] P_l^1(\cos\theta).$$

Oznaczając przesunięcia fazowe odpowiednio przez  $\delta_l = \delta_l^-$  oraz  $\delta_{-l-1} = \delta_l^+$ , powyższe wzory możemy przepisać jako

$$f(\theta) = \frac{1}{2ik} \sum_{l=0}^{\infty} [(l+1)(e^{2i\delta_l^+} - 1) + l(e^{2i\delta_l^-} - 1)] P_l(\cos\theta),$$

$$g(\theta) = \frac{1}{2ik} \sum_{l=1}^{\infty} [e^{2i\delta_l^-} - e^{2i\delta_l^+}] P_l^1(\cos\theta).$$

Napiżemy teraz jawny wzór na amplitudę rozpraszania. Dla uproszczenia oznaczeń przepiszemy wzór (3.3.12), kładąc  $a_{1/2} = a_1$ ,  $a_{-1/2} = a_2$ .

$$F_d(\hat{r}) = M(\hat{r}) \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}.$$

Powyższy wzór, używając wzoru (3.3.18), możemy zapisać w następujący sposób

$$F_d(\hat{r}) = \begin{pmatrix} f(\theta)a_1 - g(\theta)e^{-i\varphi}a_2 \\ g(\theta)e^{i\varphi}a_1 + f(\theta)a_2 \end{pmatrix}.$$

Dużą składową funkcji falowej dla elektronu rozproszonego możemy zapisać w postaci

$$\Phi_d(\hat{r}) \xrightarrow{r \rightarrow \infty} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} e^{ikr \cos \theta} + \begin{pmatrix} f(\theta)a_1 - g(\theta)e^{-i\varphi}a_2 \\ g(\theta)e^{i\varphi}a_1 + f(\theta)a_2 \end{pmatrix} \frac{e^{ikr}}{r}.$$

Korzystając z (3.1.13), obliczymy dla powyższej funkcji falowej różniczkowy przekrój czynny

$$\sigma(\theta, \varphi) = \frac{|f(\theta)a_1 - g(\theta)e^{-i\varphi}a_2|^2 + |g(\theta)e^{i\varphi}a_1 + f(\theta)a_2|^2}{|a_1|^2 + |a_2|^2}.$$

Po elementarnych przekształceniach otrzymujemy

$$\sigma(\theta, \varphi) = |f(\theta)|^2 + |g(\theta)|^2 + \frac{-a_1 a_2^* e^{i\varphi} + a_1^* a_2 e^{-i\varphi}}{|a_1|^2 + |a_2|^2} [f(\theta)g^*(\theta) - f^*(\theta)g(\theta)].$$

Równanie to, po wprowadzeniu funkcji Shermana

$$S(\theta) = i \frac{f(\theta)g^*(\theta) - f^*(\theta)g(\theta)}{|f(\theta)|^2 + |g(\theta)|^2},$$

możemy przepisać następująco

$$\sigma(\theta, \varphi) = \left[ |f(\theta)|^2 + |g(\theta)|^2 \right] \left[ 1 + S(\theta) \frac{-a_1 a_2^* e^{i\varphi} + a_1^* a_2 e^{-i\varphi}}{i(|a_1|^2 + |a_2|^2)} \right].$$

## 4. Metoda J-macierzy

### 4.1 Nierelatywistyczna metoda J-macierzy

Niech  $\{\phi_n^l\}_{n=0}^{\infty}$  będzie bazą Laguerre'a lub Gaussa (Hermite'a) (patrz dodatek C). Tylko druga z wymienionych baz tworzy bazę ortogonalną, jednak nam wystarczy, aby baza była biortonormalna. Elementy  $\{\bar{\phi}_n^l\}_{n=0}^{\infty}$  są biortonormalne do  $\{\phi_n^l\}_{n=0}^{\infty}$  w odniesieniu do poniższego iloczynu skalarnego:

$$\langle \bar{\phi}_m^l | \phi_n^l \rangle \equiv \int_0^{\infty} \bar{\phi}_m^l(\lambda r) \phi_n^l(\lambda r) dr = \delta_{mn}.$$

Najistotniejszą dla nas własnością powyższych baz jest fakt, że operator radialnej energii kinetycznej

$$H_0 - \frac{\tilde{k}^2}{2} \equiv -\frac{1}{2} \frac{d^2}{dr^2} + \frac{l(l+1)}{2r^2} - \frac{\tilde{k}^2}{2},$$

rozwinęty w dowolnej z nich, daje trójdiagonalną postać (formę Jacobiego):

$$J_{mn} \equiv \left\langle \phi_m^l \left| \left( H_0 - \frac{\tilde{k}^2}{2} \right) \phi_n^l \right. \right\rangle, \quad J_{mn} \neq 0 \text{ tylko dla } m = n, n \pm 1.$$

W powyższych równaniach  $\tilde{k}$  jest liczbą falową zależną od energii  $\mathcal{E}$  i masy  $m$  cząstki bombardującej, tzn.

$$\tilde{k} = \frac{\sqrt{2m\mathcal{E}}}{\hbar}. \quad (4.1.1)$$

Należy podkreślić, że elementy macierzowe  $J_{mn}$  są funkcjami  $\tilde{k}$ , tzn.  $J_{mn} = J_{mn}(\tilde{k})$ .

Regularne rozwiązanie  $S(\tilde{k}, r)$  równania

$$\left( H_0 - \frac{\tilde{k}^2}{2} \right) S(\tilde{k}, r) = 0 \quad (4.1.2)$$

jest wprost proporcjonalne do funkcji Riccati-Bessela i spełnia poniższe zależności:

$$S(\tilde{k}, r) \sim r^{l+1} \text{ dla } r \rightarrow 0 \text{ oraz } S(\tilde{k}, r) \xrightarrow{r \rightarrow \infty} \sin\left(\tilde{k}r - \frac{\pi l}{2}\right).$$

Używając rozwinięcia  $S(\tilde{k}, r)$  w bazie  $\{\phi_n^l\}$ , np.

$$S(\tilde{k}, r) = \sum_{n=0}^{\infty} s_n^l \phi_n^l(\lambda r),$$

równanie (4.1.2) można przepisać w postaci

$$\sum_{n=0}^{\infty} J_{mn} s_n^l = 0. \quad (4.1.3)$$

Używając jawnej postaci elementów macierzowych  $J_{mn}$ , można znaleźć współczynniki rozwinięcia  $s_n^l$ . Procedura ta jest dokładnie opisana w [3]. Znalezione współczynniki są funkcjami  $\tilde{k}$ , tzn.  $s_n^l = s_n^l(\tilde{k})$ .

W celu rozwiązania zagadnienia rozpraszania wprowadzić należy drugą funkcję  $C(\tilde{k}, r)$  typu kosinus. Potrzebujemy jej, aby spełnione były następujące zależności:

$$C(\tilde{k}, r) \sim r^{l+1} \text{ dla } r \rightarrow 0 \text{ oraz } C(\tilde{k}, r) \xrightarrow{r \rightarrow \infty} \cos\left(\tilde{k}r - \frac{\pi l}{2}\right).$$

Nie może nią być drugie rozwiązanie jednorodnego równania (4.1.2), bo rozwiązanie to, proporcjonalne do funkcji Riccati-Neumanna, jest osobliwe w zerze, czego chcemy unikać.

Funkcję  $C(\tilde{k}, r)$  znajdujemy, rozwiązując poniższe, niejednorodne równanie:

$$\left(H_0 - \frac{\tilde{k}^2}{2}\right)C(\tilde{k}, r) = \beta \bar{\phi}_0^l(\lambda r), \text{ gdzie } \beta = -\frac{\tilde{k}}{2s_0^l}.$$

W powyższym równaniu  $s_0^l$  jest pierwszym współczynnikiem rozwinięcia rozwiązania typu sinus. Można teraz napisać, że współczynniki rozwinięcia funkcji  $C(\tilde{k}, r)$  spełniają równanie

$$\sum_{n=0}^{\infty} J_{mn} c_n^l = \beta \bar{\phi}_0^l. \quad (4.1.4)$$

Współczynniki  $c_n^l$  można znaleźć metodą podaną w [3]. Zarówno  $c_n^l$ , jak i  $s_n^l$ , są wypisane w dodatku C. Obliczonych rozwinięć

$$S(\tilde{k}, r) = \sum_{n=0}^{\infty} s_n^l \phi_n^l(\lambda r) \text{ i } C(\tilde{k}, r) = \sum_{n=0}^{\infty} c_n^l \phi_n^l(\lambda r)$$

używamy do znalezienia przybliżonego rozwiązania zagadnienia rozpraszania na potencjale  $V = V(r)$  zanikającym do zera szybciej niż potencjał kulombowski:

$$\left( H_0 + V - \frac{\tilde{k}^2}{2} \right) \Psi_E = 0. \quad (4.1.5)$$

Zastępujemy potencjał  $V$  potencjałem obciętym w następujący sposób:

$$V^N = P_N^\dagger V P_N,$$

gdzie  $P_N$  jest uogólnionym operatorem rzutowania:

$$P_N = \sum_{n=0}^{N-1} |\phi_n^l\rangle \langle \bar{\phi}_n^l|.$$

Nowy, obcięty potencjał może być zapisany w bazie  $\{\phi_n^l\}$  jako macierz  $N \times N$  z elementami macierzowymi określonymi następująco:

$$V_{mn}^N = \langle \phi_m^l | V \phi_n^l \rangle. \quad (4.1.6)$$

Teraz dokładne rozwiązanie nowego zagadnienia

$$\left( H_0 + V^N - \frac{\tilde{k}^2}{2} \right) \Psi_E^N = 0, \quad (4.1.7)$$

może być rozwinięte w bazie  $\{\phi_n^l\}$  w następujący sposób:

$$\Psi_E^N(r) = \sum_{n=0}^{N-1} a_n^l \phi_n^l + S(\tilde{k}, r) + \tilde{t}_N C(\tilde{k}, r),$$

co jest równoważne

$$\Psi_E^N(r) = \sum_{n=0}^{N-1} a_n^l \phi_n^l + \sum_{n=N}^{\infty} (s_n^l + \tilde{t}_N c_n^l) \phi_n^l.$$

Dzięki takiemu rozwinięciu spełnione są warunki brzegowe:

$$\Psi_E^N(r) \xrightarrow{r \rightarrow \infty} \sin\left(\tilde{k}r - \frac{\pi l}{2}\right) + \tilde{t}_N \cos\left(\tilde{k}r - \frac{\pi l}{2}\right).$$

Wielkość  $\tilde{t}_N = \text{tg } \tilde{\delta}_N$  jest przybliżeniem poszukiwanego tangensa przesunięcia fazowego  $\tilde{t} = \text{tg } \tilde{\delta}$  dla dokładnego rozwiązania  $\Psi_E$  równania (4.1.5).

Lewa strona rzutu (4.1.7) na bazę  $\{\phi_n^l\}$  daje nam nieskończenie wiele równań ze względu na  $n$ . Jednakże wszystkie równania dla  $n \geq N+1$  są automatycznie spełnione, ponieważ współczynniki  $s_n^l, c_n^l$  spełniają tę samą rekurencyjną zależność (4.1.3) dla dowolnych  $m > 0$ . Wyraźnie to widać na poniższym, schematycznym przedstawieniu tych równań:

$$\begin{array}{cc} & 0 \qquad \qquad \qquad N-1 \\ 0 & \left( \begin{array}{cccccc|cccc} X & X & X & X & X & X & X & & & & & & & \\ X & X & X & X & X & X & X & & & & & & & & \\ X & X & X & X & X & X & X & & & & & & & & \\ X & X & X & X & X & X & X & & & & & & & & \\ X & X & X & X & X & X & X & & & & & & & & \\ X & X & X & X & X & X & X & & & & & & & & \\ N-1 & X & X & X & X & X & X & X & & & & & & & \\ \hline & & & & & & X & X & X & & & & & & \\ & & & & & & X & X & X & & & & & & \\ & & & & & & & X & X & X & & & & & \\ & & & & & & & & X & X & X & & & & \\ & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & \end{array} \right) \begin{pmatrix} d_0^l \\ d_1^l \\ \vdots \\ \vdots \\ d_{N-1}^l \\ s_N^l + \text{tg} \tilde{\delta}_N c_N^l \\ s_{N+1}^l + \text{tg} \tilde{\delta}_N c_{N+1}^l \\ s_{N+2}^l + \text{tg} \tilde{\delta}_N c_{N+2}^l \\ \vdots \\ \vdots \\ \vdots \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{pmatrix}
\end{array}$$

Dla  $m = 0 \dots N-2$  mamy równania:

$$\sum_{n=0}^{N-1} (J_{mn} + V_{mn}^N) d_n^l = 0,$$

dla  $m = N-1$ :

$$\sum_{n=0}^{N-1} (J_{N-1,n} + V_{N-1,n}^N) d_n^l + J_{N-1,N} (s_N^l + \text{tg} \tilde{\delta}_N c_N^l) = 0,$$

dla  $m = N$ :

$$J_{N,N-1} d_{N-1}^l + J_{N,N} (s_N^l + \text{tg} \tilde{\delta}_N c_N^l) + J_{N,N+1} (s_{N+1}^l + \text{tg} \tilde{\delta}_N c_{N+1}^l) = 0,$$

dla pozostałych  $m > N$ :

$$J_{N+1,N} (s_N^l + \text{tg} \tilde{\delta}_N c_N^l) + J_{N+1,N+1} (s_{N+1}^l + \text{tg} \tilde{\delta}_N c_{N+1}^l) + J_{N+1,N+2} (s_{N+2}^l + \text{tg} \tilde{\delta}_N c_{N+2}^l) \equiv 0.$$

Pozostaje zatem do rozwiązania skończony układ równań z  $N+1$  niewiadomymi:  $\tilde{t}_N, \{a_n^l\}_{n=0}^{N-1}$ . Schematycznie, równania te wyglądają następująco:

$$\left( \begin{array}{cccc|c} (J+V)_{0,0} & \cdots & (J+V)_{0,N-2} & (J+V)_{0,N-1} & 0 \\ (J+V)_{1,0} & \cdots & (J+V)_{1,N-2} & (J+V)_{1,N-1} & 0 \\ \vdots & \cdots & \vdots & \vdots & \vdots \\ (J+V)_{N-2,0} & \cdots & (J+V)_{N-2,N-2} & (J+V)_{N-2,N-1} & 0 \\ (J+V)_{N-1,0} & \cdots & (J+V)_{N-1,N-2} & (J+V)_{N-1,N-1} & J_{N-1,N}c_N^l \\ \hline 0 & \cdots & 0 & J_{N,N-1} & -J_{N,N-1}c_{N-1}^l \end{array} \right) \begin{pmatrix} a_0^l \\ a_1^l \\ \vdots \\ a_{N-2}^l \\ a_{N-1}^l \\ \tilde{t}_N \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -J_{N-1,N}s_N^l \\ J_{N,N-1}s_{N-1}^l \end{pmatrix}$$

Równania te można łatwo rozwiązać ze względu poszukiwany tangens. W szczególności, używając rekurencyjnej zależności dla elementów  $J_{mn}$  można wyliczyć, że

$$\tilde{t}_N = \text{tg} \tilde{\delta}_N = -\frac{s_{N-1}^l + g_{N-1,N-1}(\mathcal{E})J_{N,N-1}s_N^l}{c_{N-1}^l + g_{N-1,N-1}(\mathcal{E})J_{N,N-1}c_N^l}, \quad (4.1.8)$$

gdzie

$$g_{N-1,N-1}(\mathcal{E}) = \frac{\sum_{n=0}^{N-1} \Gamma_{N-1,n}^2}{\mathcal{E}_n - \mathcal{E}}. \quad (4.1.9)$$

Macierz  $\tilde{H}_N$  jest macierzą diagonalizującą zagadnienie

$$\left( \Gamma^\dagger P_N^\dagger \left( H_0 + V - \frac{\tilde{k}^2}{2} \right) P_N \Gamma \right)_{mn} = (\mathcal{E}_n - \mathcal{E}) \delta_{mn}. \quad (4.1.10)$$

Zależna od energii wielkość  $g_{N-1,N-1}(\mathcal{E})$  może być traktowana jako macierzowy element odwrotności obciętego operatora  $P_N^\dagger (H_0 + V^N - \tilde{k}^2 / 2) P_N$ , ograniczonego do  $N$ -wymiarowej przestrzeni, gdzie potencjał nie znika.

Wielkości  $\mathcal{E}_n$  są tzw. wartościami własnymi Harrisa.

## 4.2 Relatywistyczna metoda J-macierzy<sup>4</sup>

Chcemy znaleźć relatywistyczne odpowiedniki funkcji  $S(\tilde{k}, r)$  i  $C(\tilde{k}, r)$  w odpowiedniej bazie, a także wyliczyć relatywistyczne elementy macierzy Jacobiego w tej bazie. W tym celu rozpatrzmy równanie Diraca dla cząstki swobodnej w następującej postaci:

$$\left(\mathcal{H}_0 - \frac{E}{c\hbar}\right)\Psi \equiv \begin{pmatrix} mc^2 - E & -\frac{d}{dr} + \frac{\kappa}{r} \\ c\hbar & -mc^2 - E \\ \frac{d}{dr} + \frac{\kappa}{r} & c\hbar \end{pmatrix} \begin{pmatrix} F(r) \\ G(r) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \quad (4.2.1)$$

$E$  oznacza tutaj energię całkowitą, związaną z energią kinetyczną i spoczynkową zależnością  $E = \bar{\mathcal{E}} + mc^2$ . Niech  $l(\kappa)$  będzie nieujemnym rozwiązaniem równania  $\kappa(\kappa + 1) = l(l + 1)$ , tzn.  $l = \kappa \vee l = -\kappa - 1$  odpowiednio dla dodatnich i ujemnych  $\kappa$  (por. rozdział 2.2). Równanie (4.2.1) ma wtedy dwa niezależne rozwiązania. Pierwsze z nich, regularne w zerze

$$\Psi_{reg}(r) = \begin{pmatrix} F_{reg}(r) \\ G_{reg}(r) \end{pmatrix} \sim \begin{pmatrix} \hat{j}_l(kr) \\ \mp \epsilon \hat{j}_{l\pm 1}(kr) \end{pmatrix}, \quad (4.2.2)$$

jest wyznaczone przez funkcje Riccati-Bessela z następującymi zachowaniami asymptotycznymi:

$$\hat{j}_l(x) \xrightarrow{x \rightarrow 0} \frac{x^{l+1}}{(2l+1)!!} \quad \text{oraz} \quad \hat{j}_l(x) \xrightarrow{x \rightarrow \infty} \sin\left(x - \frac{\pi l}{2}\right).$$

Wielkości  $\epsilon$  oraz  $k$  zdefiniowane są następująco:

$$\epsilon \equiv \sqrt{\frac{E - mc^2}{E + mc^2}}, \quad k \equiv \frac{\sqrt{(E - mc^2)(E + mc^2)}}{c\hbar}.$$

Wielkość  $k$  zbiega się w przypadku nierelatywistycznym ( $c \rightarrow \infty$ ) do nierelatywistycznej liczby  $\tilde{k} = \frac{\sqrt{2m\mathcal{E}}}{\hbar}$ .

Drugie rozwiązanie równania (4.2.1), nieregularne w zerze, jest określone jako

$$\Psi_{irr}(r) = \begin{pmatrix} F_{irr}(r) \\ G_{irr}(r) \end{pmatrix} \sim \begin{pmatrix} -\hat{n}_l(kr) \\ \pm \epsilon \hat{n}_{l\pm 1}(kr) \end{pmatrix}. \quad (4.2.3)$$

Tutaj mamy funkcje Riccati-Neumanna z własnościami:

<sup>4</sup> Rozdział ten jest oparty na pracy P. Horodeckiego [4].



$$\hat{n}_l(x) \xrightarrow{x \rightarrow 0} -\frac{(2l-1)!!}{x^l} \quad \text{oraz} \quad \hat{n}_l(x) \xrightarrow{x \rightarrow \infty} -\cos\left(x - \frac{\pi l}{2}\right).$$

W obydwu rozwiązaniach (4.2.2) i (4.2.3) górne i dolne znaki w małych składowych dotyczą odpowiednio ujemnych i dodatnich  $\kappa$ .

Z powyższych rozważań wynika, że regularne rozwiązanie  $\Psi_{reg}$  jest relatywistycznym odpowiednikiem nierelatywistycznej funkcji  $S(\tilde{k}, r)$ . Oznaczmy rozwiązanie typu sinus  $\Psi_{reg}$  symbolem  $\Psi_S$ . Aby znaleźć macierz Jacobiego, wprowadźmy odpowiednią bazę w przestrzeni Hilberta  $L^2(0, \infty) \otimes C^2$ , w której zdefiniowany jest operator Diraca z równania (4.2.1).

Niech  $\{\phi_n^l(x)\}$  ponownie będzie bazą Laguerre'a lub Gaussa i niech

$$\psi_n^l(\lambda r) = \left( \frac{\kappa}{r} + \frac{d}{dr} \right) \phi_n^l(\lambda r).$$

Wtedy baza zdefiniowana dla naszych potrzeb ma postać:

$$\Phi_{n\kappa}^+(r) \equiv \begin{pmatrix} \phi_n^l(\lambda r) \\ 0 \end{pmatrix}, \quad \Phi_{n\kappa}^-(r) \equiv \begin{pmatrix} 0 \\ \psi_n^l(\lambda r) \end{pmatrix}. \quad (4.2.4)$$

Taki wybór bazy zapewnia, że spełniony jest warunek równowagi kinetycznej.

Elementy biortonormalne do powyższych funkcji mają postać:

$$\bar{\Phi}_{n\kappa}^+(r) \equiv \begin{pmatrix} \bar{\phi}_n^l(\lambda r) \\ 0 \end{pmatrix}, \quad \bar{\Phi}_{n\kappa}^-(r) \equiv \begin{pmatrix} 0 \\ \bar{\psi}_n^l(\lambda r) \end{pmatrix}.$$

Dla naszych celów nie jest konieczna znajomość jawnej postaci funkcji biortonormalnych.

Chcemy teraz znaleźć rozwinięcia rozwiązania typu sinus  $\Psi_S = \Psi_{reg}$  oraz typu kosinus  $\Psi_C$ . Stawiamy następujące żądania:

- (1)  $\Psi_C$  powinno mieć asymptotyczną formę jak  $\Psi_{irr}$ ;
- (2)  $\Psi_C$  powinno wykazywać regularne zachowanie w zerze;
- (3) współczynniki rozwinięcia  $\Psi_C$  powinny spełniać te same równania co współczynniki rozwinięcia  $\Psi_S = \Psi_{reg}$ .

Najpierw rozpatrzmy rozwiązanie

$$\Psi_U(k, r) = \begin{pmatrix} F_U(k, r) \\ G_U(k, r) \end{pmatrix}$$

niejednorodnego równania typu (4.2.1)

$$\left( \mathcal{H}_0 - \frac{E}{c\hbar} \right) \Psi \equiv \Phi_{inh}. \quad (4.2.6)$$

Indeks  $U = S, C$  oznacza odpowiednio rozwiązanie typu sinus i kosinus.

Niejednorodność wybieramy w postaci

$$\Phi_{inh} = \Phi_{0\kappa}^+ = \begin{pmatrix} \Omega_U \bar{\phi}_0^l \\ 0 \end{pmatrix}, \quad \text{gdzie } U = S, C \text{ oraz } \Omega_S = 0, \quad \Omega_C = -\frac{\epsilon}{s_0^l}.$$

Równanie (4.2.6) można zapisać jako

$$\begin{aligned} \left( \frac{\kappa}{r} - \frac{d}{dr} \right) G_U - k\epsilon F_U &= \Omega_U \bar{\psi}_0^l, \\ \left( \frac{\kappa}{r} + \frac{d}{dr} \right) F_U - \frac{k}{\epsilon} G_U &= 0. \end{aligned} \quad (4.2.7)$$

Wprowadźmy teraz relatywistyczny odpowiednik macierzy Jacobiego:

$$\mathcal{J}_{mn}^{ss'} \equiv \left\langle \Phi_{m\kappa}^s \left| \left( \mathcal{H}_0 - \frac{E}{c\hbar} \right) \Phi_{n\kappa}^s \right. \right\rangle, \quad s, s' = \pm, \quad m, n = 0, 1, 2, \dots \quad (4.2.8)$$

Elementy macierzy  $\mathcal{J}$  można wyrazić w bardzo prostej postaci. W tym celu zdefiniujemy macierze  $\mathcal{J}_{mn}$  o wymiarach  $2 \times 2$  jako  $\{\mathcal{J}_{mn}\}_{ss'} \equiv \mathcal{J}_{mn}^{ss'}$ . Nowe macierze otrzymują teraz wyjątkowo prostą formę:

$$\mathcal{J}_{mn} = \begin{pmatrix} -k\epsilon \langle \phi_m^l | \phi_n^l \rangle & \langle \psi_m^l | \psi_n^l \rangle \\ \langle \psi_m^l | \psi_n^l \rangle & -\frac{k}{\epsilon} \langle \psi_m^l | \psi_n^l \rangle \end{pmatrix}. \quad (4.2.9)$$

Występujące w powyższym wzorze całki, wypisane w dodatku C, łączy z nierelatywistycznymi elementami J-macierzy następujący, prosty związek:

$$J_{mn} = \frac{1}{2} \langle \psi_m^l | \psi_n^l \rangle - \frac{\tilde{k}^2}{2} \langle \phi_m^l | \phi_n^l \rangle. \quad (4.2.10)$$

Teraz możemy przewidzieć rozwinięcia obydwu rozwiązań w bazie (4.2.4) w ten sposób, że duże składowe rozwiązań typu sinus i kosinus są dane przez identyczne współczynniki

rozwinąć jak w przypadku nierelatywistycznym, wzięte w punkcie  $k$ , oraz współczynniki małych składowych są odpowiednio przeskalowane:

$$\Psi_U = \sum_{s=\pm} \sum_{n=0}^{\infty} u_{nk}^s \Phi_{nk}^s \equiv \sum_{n=0}^{\infty} u_n^l(k) \begin{pmatrix} \phi_n^l \\ \epsilon \\ -\psi_n^l \\ \kappa \end{pmatrix}, \quad U = S, C, \quad u = s, c. \quad (4.2.11)$$

Łatwo można sprawdzić, że powyższe wyrażenie rzeczywiście jest rozwiązaniem równania (4.2.1). Mianowicie, podstawiając je do tego równania i używając definicji elementów macierzowych (4.2.8), dostajemy nieskończony układ równań:

$$\sum_{s'=\pm} \sum_{n=0}^{\infty} \mathcal{J}_{mn}^{ss'} u_{nk}^{s'} = \Omega_U \bar{\phi}_0^l \delta_{n0} \delta_{s,+}, \quad s = \pm, \quad m = 0, 1, 2, \dots \quad (4.2.12)$$

Dla ustalonej pary  $(m, n)$  drugi element dolnego wiersza macierzy (4.2.9) jest przeskalowany przez  $-\frac{k}{\epsilon}$  i dostajemy od razu, że wszystkie równania (4.2.12) z ujemnym indeksem  $s = "-"$  są spełnione.

Przywołując definicję  $l = l(\kappa)$  po wycalkowaniu przez części dostajemy, że

$$\langle \psi_m^l | \psi_n^l \rangle = \left\langle \left( \frac{\kappa}{r} + \frac{d}{dr} \right) \phi_m^l \left| \left( \frac{\kappa}{r} + \frac{d}{dr} \right) \phi_n^l \right. \right\rangle = \left\langle \phi_m^l \left| \left( -\frac{d^2}{dr^2} + \frac{l(l+1)}{r^2} \right) \phi_n^l \right. \right\rangle.$$

Biorąc pod uwagę kształt górnego wiersza macierzy (4.2.9) oraz tożsamość (4.2.10), natychmiast dostajemy, że równania (4.2.12) z indeksem  $s = "+"$  mają identyczną postać jak układy równań (4.1.3), (4.1.4), wyliczone w punkcie  $k$  zamiast w  $\tilde{k}$ . Pokazaliśmy więc, że rozwinięcia (4.2.11) rzeczywiście są rozwiązaniami równania (4.2.7). Z przypadku nierelatywistycznego widać, że ich duże składowe mają pożądane własności w zerze i nieskończoności. Ponadto, równania (4.2.1) są ze sobą powiązane i zachowanie jednej składowej wyznacza zachowanie drugiej. Stąd obie składowe rozwiązań  $\Psi_S$  i  $\Psi_C$  mają pożądane zachowania asymptotyczne, tzn.  $\Psi_S$  jest regularnym rozwiązaniem,  $\Psi_C$  natomiast zachowuje się jak  $\Psi_{irr}$  w nieskończoności oraz jak  $\Psi_{reg}$  w zerze. Ponieważ niejednorodność wiąże tylko biortonormalny element  $\Psi_{0\kappa}^+$ , to obydwie funkcje spełniają ten sam układ równań.

Rozważmy teraz rozpraszanie na dostatecznie regularnym potencjale  $V = V(r)$  zanikającym w nieskończoności szybciej niż potencjał kulombowski. Szukamy zatem rozwiązania następującego równania:

$$\left( \mathcal{H}_0 + \frac{V}{c\hbar} - \frac{E}{c\hbar} \right) \Psi_E = 0.$$

Rozwiązanie tego równania powinno spełniać następujący warunek asymptotyczny:

$$\Psi_E(k, r) \sim \Psi_S(k, r) + t \Psi_C(k, r)$$

gdzie  $t = \operatorname{tg} \delta$  jest poszukiwanym tangensem przesunięcia fazowego.

W celu rozwinięcia formalizmu relatywistycznych J-macierzy używamy uogólnionego operatora rzutowania

$$\mathcal{P}_N = \sum_{s=\pm} \sum_{n=0}^{\infty} |\Phi_{nk}^s\rangle \langle \bar{\Phi}_{nk}^s|,$$

definiując za jego pomocą obcięty potencjał

$$V^N = \mathcal{P}_N^\dagger \frac{V}{c\hbar} \mathcal{P}_N. \quad (4.2.13)$$

Teraz szukamy dokładnego rozwiązania równania z obciętym potencjałem

$$\left( \mathcal{H}_0 + V^N - \frac{E}{c\hbar} \right) \Psi_E^N(r) = 0. \quad (4.2.14)$$

Należy zauważyć, że dla dowolnej  $\psi \in L^2(0, \infty) \otimes C^2$  funkcja  $V^N \psi$  zanika w nieskończoności szybciej niż  $1/r$ .

W ten sposób, chociaż równanie (4.2.14) nie ma standardowej postaci równania Diraca z tym samym skalarnym potencjałem dla dużej i małej składowej, nadal jego rozwiązanie spełnia asymptotycznie równanie Diraca dla cząstki swobodnej. Zatem rozwiązanie  $\Psi_E^N$  spełnia następujący warunek brzegowy:

$$\Psi_E^N(k, r) \sim \Psi_S(k, r) + t_N \Psi_C(k, r) \quad (4.2.15)$$

gdzie  $t_N = \operatorname{tg} \delta_N$  jest poszukiwanym przybliżeniem tangensa przesunięcia fazowego. Ponieważ

$$V^N \xrightarrow{N \rightarrow \infty} \frac{V}{c\hbar},$$

to spodziewamy się, że dla  $N \rightarrow \infty$   $t_N$  jest zbieżne do właściwej wartości  $t = \operatorname{tg} \delta$ .

Najogólniejszy wzór na  $\Psi_E^N$  ma postać

$$\Psi_E^N = \sum_{s=\pm} \sum_{m=0}^{\infty} d_{mk}^s |\Phi_{mk}^s\rangle.$$

Rozważmy macierzową reprezentację równania (4.2.14). Kładąc powyższe rozwinięcie funkcji  $\Psi_E^N$  w bazie  $\{\Phi_{nk}^{\pm}\}$ , dostajemy nieskończony układ równań:

$$\sum_{s'=\pm} \sum_{n=0}^{\infty} (\mathcal{J} + V^N)_{mn}^{ss'} d_{nk}^{s'} = 0, \quad s = \pm, \quad m = 0, 1, 2, \dots \quad (4.2.16)$$

Współczynniki rozwinięcia  $\{d_{nk}^{\pm}\}$  powinny spełniać następujące warunki:

(1) Dla dużej składowej  $\sum_{n=0}^{\infty} J_{mn}(k) d_{nk}^+ = 0$ ,  $m > N$  (elementy  $J_{mn}$  – nierelatywistyczne);

(2) Dla małej składowej  $d_{nk}^- = \frac{\epsilon}{k} d_{nk}^-$ .

Ponadto nakładamy dodatkowy warunek:

(3)  $\Psi_E^N \sim S(k, r) + t_N C(k, r)$  (patrz warunek (4.2.15)).

Warunki (2) i (3) dają nam następującą postać rozwiązania  $\Psi_E^N$  równania (4.2.14)

$$\Psi_E^N = \sum_{m=0}^{N-1} \begin{pmatrix} d_{mk}^+ \phi_m^l \\ d_{mk}^- \frac{\epsilon}{k} \psi_m^l \end{pmatrix} + \sum_{m=N}^{\infty} \begin{pmatrix} (s_{km}^+ + t_N c_{km}^+) \phi_m^l \\ (s_{km}^- + t_N c_{km}^-) \psi_m^l \end{pmatrix}. \quad (4.2.17)$$

Dodając i odejmując od lewej strony powyższej równości wielkość

$$\sum_{m=0}^{N-1} \begin{pmatrix} (s_{km}^+ + t_N c_{km}^+) \phi_m^l \\ (s_{km}^- + t_N c_{km}^-) \psi_m^l \end{pmatrix},$$

można łatwo zauważyć, że powyższa funkcja spełnia asymptotyczny warunek (4.2.15).

Powróćmy do równania (4.2.16). Schematycznie może być ono przedstawione w następującej postaci:

$$\begin{pmatrix} \cdot & \cdot & \cdot & & & & & & & \cdot & \cdot \\ X & X & X & & 0 & & & & & X & X & X \\ & X & X & X & & & & & & X & X & X \\ & & X & X & X & X & X & & & X & X & X \\ & & & X & X & X & X & X & & X & X & X \\ & & & & X & X & X & X & & & & \\ & & & & & X & X & X & & & & \\ & & & & & & X & X & X & & & \\ & & & & & & & X & X & X & & \\ & & & & & & & & & & 0 & \\ 0 & & & & & & X & X & X & X & X & X \\ & & & & & & & X & X & X & X & X \\ & & & & & & & X & X & X & X & X \\ & & & & & & & X & X & X & X & X \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ X & X & X & & & & & & & X & X & X \\ & & & & & & & & & X & X & X \\ X & X & X & & 0 & & & & & X & X & X \\ \cdot & \cdot & \cdot & & & & & & & \cdot & \cdot & \cdot \end{pmatrix} \begin{pmatrix} \cdot \\ \cdot \\ \cdot \\ s_{N+1,l}^+ + t_N c_{N+1,l}^+ \\ d_{N-1,l}^+ \\ \cdot \\ \cdot \\ d_{1,l}^+ \\ d_{0,l}^+ \\ d_{0,l}^- \\ d_{1,l}^- \\ \cdot \\ \cdot \\ d_{N-1,l}^- \\ \cdot \\ \cdot \\ s_{N+1,l}^- + t_N c_{N+1,l}^- \\ \cdot \\ \cdot \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Z konstrukcji rozwiązania (4.2.17) widzimy, że wszystkie równania dla  $m > N$  są automatycznie spełnione. W ten sposób pozostaje do rozwiązania  $2N+2$  równań z niewiadomymi  $t_N$ ,  $d_{0,k}^+$ ,  $d_{1,k}^+$ , ...,  $d_{N-1,k}^+$ ,  $d_{0,k}^-$ ,  $d_{1,k}^-$ , ...,  $d_{N-1,k}^-$ .

Używając równań (4.2.16), otrzymujemy następującą postać pozostałych równań:

$$\begin{pmatrix} -\mathcal{J}_{N,N-1}^{++} c_{N-1}^+ & \mathcal{J}_{N,N-1}^{++} & 0 & \dots & 0 & \mathcal{J}_{N,N-1}^{+-} & -\mathcal{J}_{N,N-1}^{--} c_{N-1}^- \\ \mathcal{J}_{N-1,N}^{++} c_N^+ & (\mathcal{J} + V^N)_{N-1,N-1}^{++} & (\mathcal{J} + V^N)_{N-1,N-2}^{++} & \dots & \mathcal{J}_{N-1,N-2}^{+-} & \mathcal{J}_{N-1,N-1}^{+-} & \mathcal{J}_{N-1,N}^{+-} c_N^- \\ 0 & (\mathcal{J} + V^N)_{N-2,N-1}^{++} & (\mathcal{J} + V^N)_{N-2,N-2}^{++} & \dots & \mathcal{J}_{N-2,N-2}^{+-} & \mathcal{J}_{N-2,N-1}^{+-} & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ 0 & (\mathcal{J} + V^N)_{0,N-1}^{++} & (\mathcal{J} + V^N)_{0,N-2}^{++} & \dots & (\mathcal{J} + V^N)_{0,N-2}^{+-} & (\mathcal{J} + V^N)_{0,N-1}^{+-} & 0 \\ 0 & (\mathcal{J} + V^N)_{0,N-1}^{+-} & (\mathcal{J} + V^N)_{0,N-2}^{+-} & \dots & (\mathcal{J} + V^N)_{0,N-2}^{--} & (\mathcal{J} + V^N)_{0,N-1}^{--} & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ 0 & \mathcal{J}_{N-2,N-2}^{+-} & \mathcal{J}_{N-2,N-2}^{--} & \dots & (\mathcal{J} + V^N)_{N-2,N-2}^{--} & (\mathcal{J} + V^N)_{N-2,N-1}^{--} & 0 \\ \mathcal{J}_{N-1,N}^{+-} c_N^- & \mathcal{J}_{N-1,N-1}^{+-} & \mathcal{J}_{N-1,N-2}^{+-} & \dots & (\mathcal{J} + V^N)_{N-1,N-2}^{--} & (\mathcal{J} + V^N)_{N-1,N-1}^{--} & \mathcal{J}_{N-1,N}^{--} c_N^- \\ -\mathcal{J}_{N,N-1}^{--} c_{N-1}^- & \mathcal{J}_{N,N-1}^{--} & 0 & \dots & 0 & \mathcal{J}_{N,N-1}^{--} & -\mathcal{J}_{N,N-1}^{--} c_{N-1}^- \end{pmatrix} \begin{pmatrix} t_N \\ d_{N-1}^+ \\ d_{N-2}^+ \\ \vdots \\ d_0^+ \\ d_0^- \\ \vdots \\ d_{N-2}^- \\ d_{N-1}^- \\ t_N \end{pmatrix} = \begin{pmatrix} (\mathcal{J}_{N,N-1}^{++} s_{N-1}^+ + \mathcal{J}_{N,N-1}^{+-} s_{N-1}^-) \\ -\mathcal{J}_{N-1,N}^{++} s_N^+ - \mathcal{J}_{N-1,N}^{+-} s_N^- \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ -\mathcal{J}_{N-1,N}^{+-} s_N^+ - \mathcal{J}_{N-1,N}^{--} s_N^- \\ (\mathcal{J}_{N,N-1}^{+-} s_{N-1}^+ + \mathcal{J}_{N,N-1}^{--} s_{N-1}^-) \end{pmatrix}$$

Mając na uwadze, że wewnętrzna  $2N \times 2N$ -wymiarowa macierz  $(\mathcal{J} + V^N)_{nm}^{ss'}$ ,  $s, s' = \pm$ ,  $m, n = 0, 1, \dots, N-1$  jest hermitowska i rzeczywista, stąd symetryczna, oraz przywołując definicję  $V^N$ , możemy rozwiązać powyższe równania za pomocą pewnej ortogonalnej macierzy  $\Gamma$  (por. przypadek nierelatywistyczny):

$$\left( \Gamma^\dagger \mathcal{P}_N^\dagger \left( \mathcal{H}_0 + \frac{V}{c\hbar} - \frac{E}{c\hbar} \right) \mathcal{P}_N \Gamma \right)_{mn}^{ss'} = \frac{1}{c\hbar} (E_n^s - E) \delta_{nm} \delta_{ss'} \quad (4.2.18)$$

Macierz  $G(E)$  o wymiarze  $2N \times 2N$  z elementami zdefiniowanymi jako

$$G_{mn}^{ss'}(E) = \sum_{p=\pm} \sum_{i=0}^{N-1} c\hbar \frac{\Gamma_{mi}^{sp} \Gamma_{ni}^{s'p}}{E_i^p - E} \quad (4.2.19)$$

jest macierzą odwrotną do macierzowej reprezentacji obciętego operatora

$$\mathcal{P}_N^\dagger \left( \mathcal{H}_0 + \frac{V}{c\hbar} - \frac{E}{c\hbar} \right) \mathcal{P}_N.$$

Macierz ta może być rozumiana jako przybliżenie relatywistycznej funkcji Greena w bazie (4.2.4). Liczby  $E_n^p$  są relatywistycznymi odpowiednikami wartości własnych Harrisa i przedstawiają skończone przybliżenie widma relatywistycznego hamiltonianu  $\mathcal{H}_0 + V / c\hbar$ .

Wprowadźmy  $(2N+2) \times (2N+2)$ -wymiarową macierz  $\tilde{\Gamma}_{(2N+2) \times (2N+2)} = \text{diag}(1, \Gamma_{2N \times 2N}, 1)$  oraz podziałajmy nią na lewe strony powyższych  $2N+2$  równań. Korzystając z faktu, że macierz  $\mathcal{J}_{N,N-1}$  dana wzorem (4.2.9) jest nieosobliwa, układ może być rozwiązany ze względu na  $t_N$ . Używając właściwości elementów macierzy  $\mathcal{J}$ , można wyprowadzić wzór na przybliżony tangens przesunięcia fazowego w postaci bardzo podobnej do nierelatywistycznej:

$$t_N = \text{tg } \delta_N = - \frac{s_{N-1}^l(k) + \frac{2\epsilon}{k} G_{N-1,N-1}^{++}(E) J_{N,N-1}(k) s_N^l(k)}{c_{N-1}^l(k) + \frac{2\epsilon}{k} G_{N-1,N-1}^{++}(E) J_{N,N-1}(k) c_N^l(k)}. \quad (4.2.20)$$

Elementy  $J_{N,N-1}$  w powyższym wzorze są elementami nierelatywistycznymi (por. (4.2.10)). Fakt, że mamy  $2N+2$  równań i tylko  $2N+1$  niewiadomych, skutkuje tym, że mamy drugi, bardzo podobny wzór na  $t_N$ :

$$t_N = \text{tg } \delta_N = - \frac{s_{N-1}^l(k) + 2G_{N-1,N-1}^{++}(E) J_{N,N-1}(k) s_N^l(k)}{c_{N-1}^l(k) + 2G_{N-1,N-1}^{++}(E) J_{N,N-1}(k) c_N^l(k)}. \quad (4.4.21)$$

Wyrażenie  $G_{N-1,N-1}^{++}(E)$  w równaniu (4.2.20) zastąpione zostało przez  $\frac{k}{\epsilon} G_{N-1,N-1}^{++}(E)$ .

Z poprzednich analiz wynika, że oba równania muszą dać tę samą wartość  $t_N$ , z czego wynika, że  $G_{N-1,N-1}^{++}(E) = \frac{k}{\epsilon} G_{N-1,N-1}^{++}(E)$ .

Porównując równanie (4.2.20) z nierelatywistycznym wzorem (4.1.8), można zauważyć, że za wyjątkiem wielkości  $(2\epsilon/k)G_{N-1,N-1}^{++}(E)$ , wszystkie elementy wyrażenia na tangens przesunięcia fazowego mają tę samą postać co w równaniu (4.1.8), są tylko wyliczone dla

relatywistycznej liczby falowej  $k$ . Ponadto, dla dowolnego  $N$  wzór (4.2.20) zbiega się do granicy nierelatywistycznej, gdy prędkość światła  $c$  zdoła do nieskończoności. Rzeczywiście, użyta baza (4.2.4) gwarantuje, że w granicy  $c \rightarrow \infty$  duża składowa spełnia równanie Schrödingera (4.1.7) z liczbą falową  $\tilde{k} = \lim_{c \rightarrow \infty} k$ . Oznacza to, że musi zachodzić

$\lim_{c \rightarrow \infty} t_N = \tilde{t}_N$ . Zatem natychmiast dostajemy, że

$$\lim_{c \rightarrow \infty} (2\epsilon / k) G_{N-1, N-1}^{++}(E) = g_{N-1, N-1}(\mathcal{E}).$$

Ponadto  $G_{N-1, N-1}^{++}(E)$  odgrywa analogiczną rolę jak  $g_{N-1, N-1}(\mathcal{E})$ . Macierze  $G(E)$  oraz  $g(\mathcal{E})$  mogą być rozpatrywane jako skończone przybliżenia funkcji Greena dla odpowiednio relatywistycznego i nierelatywistycznego hamiltonianu z potencjałem  $V$ .

## 5. Obliczenia numeryczne

### 5.1 Przeskalowanie energii

Do obliczeń relatywistycznych autor użył nieco innych postaci odpowiednich wzorów, bardziej odpowiednich z numerycznego punktu widzenia. Mianowicie, zamiast używać całkowitej energii  $E$ , autor posłużył się odpowiednio przeskalowaną energią kinetyczną  $\bar{\mathcal{E}}$ , wskutek czego zmieni się postać liczb  $\epsilon$  i  $k$ , tzn.

$$\epsilon \equiv \sqrt{\frac{\bar{\mathcal{E}}}{\bar{\mathcal{E}} + 2mc^2}}, \quad k \equiv \frac{\sqrt{\bar{\mathcal{E}}(\bar{\mathcal{E}} + 2mc^2)}}{c\hbar}. \quad (5.1.1)$$

Odpowiedniej zmianie ulegają elementy  $\mathcal{J}_{mn}$  (por. wzór (4.2.9)):

$$\mathcal{J}_{mn} = \begin{pmatrix} -\bar{\mathcal{E}} \langle \phi'_m | \phi'_n \rangle & c\hbar \langle \psi'_m | \psi'_n \rangle \\ c\hbar \langle \psi'_m | \psi'_n \rangle & -(\bar{\mathcal{E}} + 2mc^2) \langle \psi'_m | \psi'_n \rangle \end{pmatrix} \quad (5.1.2)$$

oraz obcięty potencjał (wzór (4.2.13))

$$V^N = \mathcal{P}_N^\dagger V \mathcal{P}_N. \quad (5.1.3)$$

Teraz wzór (4.2.18) przyjmuje postać (po dodatkowym przemnożeniu przez  $c\hbar$ ):

$$\left( \bar{\Gamma}^\dagger \mathcal{P}_N^\dagger (\bar{\mathcal{H}}_0 + V - \bar{\mathcal{E}}) \mathcal{P}_N \bar{\Gamma} \right)_{mn}^{ss'} = (\bar{\mathcal{E}}_n^s - \bar{\mathcal{E}}) \delta_{nm} \delta_{ss'}. \quad (5.1.4)$$

Macierz przybliżająca funkcję Greena (4.2.19) będzie postaci



$$\bar{G}_{mn}^{ss'}(\bar{\mathcal{E}}) = \sum_{p=\pm} \sum_{i=0}^{N-1} \frac{\bar{\Gamma}_{mi}^{sp} \bar{\Gamma}_{ni}^{s'p}}{\bar{\mathcal{E}}_i^p - \bar{\mathcal{E}}}. \quad (5.1.5)$$

Wzór na przesunięcie fazowe (4.2.20) będzie wyglądał następująco:

$$t_N = \text{tg} \delta_N = - \frac{s_{N-1}^l(k) + \frac{2\epsilon c \hbar}{k} \bar{G}_{N-1, N-1}^{++}(\bar{\mathcal{E}}) J_{N, N-1}(k) s_N^l(k)}{c_{N-1}^l(k) + \frac{2\epsilon c \hbar}{k} \bar{G}_{N-1, N-1}^{++}(\bar{\mathcal{E}}) J_{N, N-1}(k) c_N^l(k)}. \quad (5.1.6)$$

Ponadto, wszystkie obliczenia prowadzone były w jednostkach atomowych, tzn. przyjęto  $m = \hbar = 1$ ,  $c = 1/\alpha \cong 137,036$ , gdzie  $\alpha$  – stała struktury subtelnej. Przejście z prędkością światła do nieskończoności realizuje się w takim układzie przez położenie  $c = 10^5 \div 10^6$ .

Obliczenia nierelatywistyczne były przeprowadzane bez żadnych dodatkowych operacji, tzn. dla energii kinetycznej elektronu  $\mathcal{E}$ .

## 5.2 Program komputerowy

W celu przetestowania metody J-macierzy autor napisał program komputerowy w języku Fortran 77. Dodatkowo, aby uniknąć pomyłek w programowaniu, obliczenia były prowadzone w programie Mathematica 3.0. Kody źródłowe programów zawarte są w dodatku D.

Program został napisany w podwójnej precyzji (8-bajtowa reprezentacja liczb rzeczywistych) i składa się z szeregu procedur:

- generujących funkcje specjalne i wielomiany różnego typu niezbędne do przeprowadzenia obliczeń (funkcja gamma, funkcje Bessela i Neumanna i ich pochodne, sferyczne funkcje Bessela i Neumanna i ich pochodne, funkcje hipergeometryczne ( ${}_0F_1$ ,  ${}_1F_1$  i  ${}_2F_1$ ), wielomiany Gegenbauera, Laguerre'a, funkcja delta Kroneckera itp.);
- obliczających funkcje bazowe, elementy rozwinięć oraz całki wymienione w dodatku C, używane następnie do sformowania J-macierzy (wzory (4.2.10) i (5.1.2) odpowiednio dla wersji nierelatywistycznej i relatywistycznej);
- obcinających potencjał rozpraszający w odpowiedniej bazie (wzór (4.1.6) i (5.1.3));
- diagonalizujących macierz  $\mathcal{J} + V^N$  (wzory (4.1.10) i (5.1.4)) i wyliczających na tej podstawie macierz przybliżającą funkcję Greena (wzory (4.1.9) i (5.1.5));

- wyliczających przybliżoną wartość tangensa przesunięcia fazowego według wzorów (4.1.8) i (5.1.6);
- wyliczających tangens przesunięcia fazowego według wzorów analitycznych w celu przetestowania programu (patrz rozdział 5.3).

Część procedur generujących funkcje specjalne (a mianowicie procedury generujące funkcje Bessela i Neumanna oraz funkcję gamma) i wykonujących pewne operacje (szukanie wektorów własnych i wartości własnych macierzy oraz precyzyjne całkowanie numeryczne) nie została napisana przez autora, lecz zaczerpnięta z darmowych, ogólnodostępnych bibliotek (dostępnych np. w Internecie pod adresem [www.netlib.org](http://www.netlib.org)) i dostosowana odpowiednio do potrzeb programu. Jednak większość kodu autor napisał samodzielnie, korzystając z pozycji książkowych traktujących o funkcjach specjalnych [7], [8]. Powstała również specjalna wersja programu przeznaczona dla kompilatora *Microsoft Fortran PowerStation* (oraz innych kompilatorów z nim zgodnych, np. *Digital Visual Fortran*), która tym różni się od wersji podstawowej, że tablice i macierze są w niej deklarowane dynamicznie, co nieco uprościło programowanie. Autor planuje w przyszłości przepisać kod programu w języku Fortran 90, w którym dynamiczne deklaracje zmiennych są standardowo wbudowane.

Program wymaga następujących danych wejściowych, podawanych w odpowiednim pliku konfiguracyjnym:

- energia  $\bar{\mathcal{E}}$  (lub  $\mathcal{E}$ ) cząstki bombardującej (patrz rozdział 5.1);
- liczba kwantowa  $l$  i związana z nią  $\kappa$ ;
- parametr skalujący  $\lambda$ ;
- wybrana baza (Laguerre'a lub Gaussa);
- schemat obliczeń (nierelatywistyczny lub relatywistyczny);
- parametr określający, czy do obliczeń relatywistycznych brana jest skończona lub nieskończona wartość prędkości światła (patrz rozważania pod koniec rozdziałów 4.2 oraz 5.1);
- parametr określający rodzaj potencjału rozpraszającego (prostokątna studnia potencjału lub inny, dowolny potencjał w postaci analitycznej);
- parametry sterujące (nazwa pliku wynikowego, początkowa ( $N_{start}$ ) i końcowa ( $N_{end}$ ) wartość  $N$  wykorzystywana przy obcinaniu potencjału);
- parametry prostokątnej studni potencjału (lewa i prawa krawędź, głębokość) – brane pod uwagę tylko wtedy, kiedy potencjał rozpraszający ma kształt studni; w

programie można zdefiniować potencjał o dowolnym kształcie, danym analitycznie.

Przykładowy plik konfiguracyjny:

```

GENERAL PARAMETERS
-----
Energy of projectile          E = 2.5d0
Quantum numbers              l = 1
                              kappa = -2
Scaling parameter            lambda = 1.0d0
Basis set                    basis = gauss
Scheme                       scheme = relativistic
Velocity of light            v_light = finite
Potential type                type = well

STEERING PARAMETERS
-----
File to store results        fout = r_g_1_-2
Initial value of Ntrunc      Nstart = 10
Final value of Ntrunc        Nend = 40

PARAMETERS OF POTENTIAL WELL
-----
Depth                        v0 = -1.5d0
Left bound                   a = 1.0d0
Right bound                  b = 1.5d0

-----
Some notes on parameters
-----

Restriction on kappa: kappa = 1 or kappa = -1 - 1
                        (only if relativistic scheme)

Scaling parameter:        lambda <> 0.5

Basis set:                 basis = { laguerre, gauss }

Scheme:                   scheme = { relativistic, non-relativistic }

Velocity of light:        v_light = { finite, infinite }
                        (doesn't matter if non-relativistic scheme)

Potential type:           type = { well, non-well }
                        All parameters of potential well are ignored
                        if type = non-well

```

Wynikiem działania programu (zapisywanym do pliku) są wartości tangensów przesunięć fazowych  $\text{tg} \delta_N$  ( $\text{tg} \tilde{\delta}_N$  dla wersji nierelatywistycznej) oraz samych przesunięć ( $\delta_N$  i  $\tilde{\delta}_N$ ) dla określonych w pliku konfiguracyjnym naturalnych liczb  $N$ , określających ilość funkcji bazowych w których obcinany jest potencjał. Dla  $N \rightarrow \infty$  wartości  $\text{tg} \delta_N$  są kolejnymi przybliżeniami rzeczywistej wartości  $\text{tg} \delta$  (patrz rozdziały 4.1, 4.2). Dodatkowo zapisywany jest uśredniony wynik dla bieżącej i poprzednich wartości  $N$ . W celu uniknięcia pomyłek, na początku pliku wynikowego przepisywany jest wejściowy plik konfiguracyjny oraz zapisywane są data i czas rozpoczęcia obliczeń (w wersji dla *MS Fortran*) i wynik analityczny (jeżeli potencjał ma kształt studni potencjału).

Przykładowy plik wynikowy (odpowiadający powyższemu plikowi konfiguracyjnemu):

```

# J-MATRIX METHOD
# -----
#
# DATE AND TIME OF COMPUTATION
# -----
# 1999. 2.10    20:30
#
# GENERAL PARAMETERS
# -----
# Energy of projectile          E = 2.5d0
# Quantum numbers              l = 1
#                               kappa = -2
# Scaling parameter            lambda = 1.0d0
# Basis set                     basis = gauss
# Scheme                        scheme = relativistic
# Velocity of light             v_light = finite
# Potential type                 type = well
#
# STEERING PARAMETERS
# -----
# File to store results         fout = r_g_1_-2
# Initial value of Ntrunc       Nstart = 10
# Final value of Ntrunc         Nend = 40
#
# PARAMETERS OF POTENTIAL WELL
# -----
# Depth                         v0 = -1.5d0
# Left bound                    a = 1.0d0
# Right bound                   b = 1.5d0
#
#
# ANALYTICAL RESULT
# -----
# delta    0.567566818088086
#
# NUMERICAL RESULTS: n  tan(delta)  delta  average(delta)
# -----
10  0.613615034010448146  0.550370452536675048  0.550370452536675048
11  0.715270220166890458  0.620901085019879728  0.585635768778277388
12  0.562111675390912224  0.512094422936371751  0.561121986830975472
13  0.663960502723043411  0.586126764709871684  0.56737373181300699581
14  0.663491430095154877  0.585801140164485257  0.571058773073456760
15  0.581521097075351934  0.526721247095865319  0.563669185410524798
16  0.703120510879143623  0.612817198166243582  0.570690330089913211
17  0.606110892548729741  0.544900692325251734  0.567466625369330568
18  0.649294621215873446  0.575879188144596155  0.568401354566582362
19  0.638354200958231255  0.568144753940502345  0.568375694503974294
20  0.639776874892302838  0.569154885402664013  0.568446530040218834
21  0.623038409852989084  0.557187489917602918  0.567508276696667591
22  0.669499070969901355  0.589960936270762071  0.569235404356213226
23  0.606074482960124894  0.544874064444574868  0.567495308648239161
24  0.666398973554741092  0.587817254649145915  0.568850105048299581
25  0.622578796760573927  0.556856330453336268  0.568100494136114298
26  0.636760193849674971  0.567011422449566482  0.568036431095729100
27  0.650197937788655200  0.576514355836384440  0.568507426914654390
28  0.621041228262965372  0.555747491826153794  0.567835851383680645
29  0.658426843508253201  0.582276396201295454  0.568557878624561419
30  0.626864041527826155  0.559938626862196420  0.568147438064448784
31  0.647400851354210660  0.574545880472427162  0.568438276355720529
32  0.635486887681411039  0.566104930922703864  0.568336826554285079
33  0.637666122617901787  0.567655734405599022  0.568308447714756526
34  0.637429363081267075  0.567487398059410841  0.568275605728542632
35  0.638992062566270391  0.568597811745759629  0.568287998267666405
36  0.635399380711809014  0.566042594607637506  0.568204835169146771
37  0.643900350041978653  0.572075283411949798  0.568343065463532637
38  0.633931111738603059  0.564995924357030166  0.568227646804687736
39  0.644145389195793450  0.572248484667905455  0.568361674733461664
40  0.634439217916909270  0.565358290285415954  0.568264791364169963

```

Jak już wspomniano wcześniej, wynikiem działania programu są wartości tangensów przesunięć fazowych  $\text{tg} \delta_N$  dla kolejnych naturalnych liczb  $N$ , określających ilość funkcji

bazowych w których obcinany jest potencjał. Spodziewamy się, że  $\operatorname{tg} \delta_N \xrightarrow{N \rightarrow \infty} \operatorname{tg} \delta$  tzn. że kolejne wartości  $\operatorname{tg} \delta_N$  są coraz lepszymi przybliżeniami rzeczywistej wartości  $\operatorname{tg} \delta$ . Z trzech powodów nie udało się jednak uzyskać wyników dla dużych wartości  $N$ :

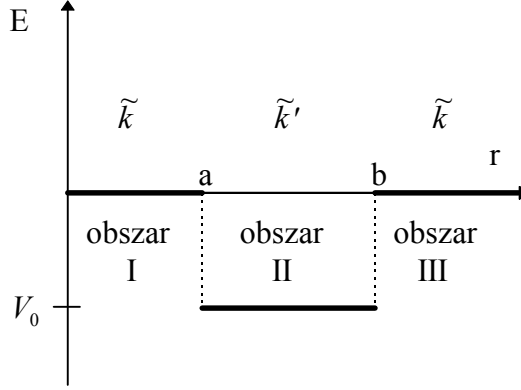
1. Jest trudne (przy użyciu standardowych technik programowania) uzyskanie wyników dla  $N > 171$ . Wynika to stąd, że we wzorach (patrz dodatek C) występuje funkcja  $n!$  oraz funkcja  $\Gamma(n)$ , których wartość dla  $n = N > 171$  przekracza dopuszczalny zakres zmiennej rzeczywistej podwójnej precyzji w Fortranie (około  $1.0E+308$ , w zależności od konkretnej architektury systemu). Dla obliczeń przy użyciu bazy Laguerre'a można częściowo ten problem ominąć, upraszczając wszystkie funkcje  $n!$  z funkcjami gamma dla których argument jest całkowity, co autor uczynił w programie. Dla bazy Gaussa takiego uproszczenia nie można zrobić, bo argumentami funkcji gamma są liczby ułamkowe. Można również posłużyć się logarytmami omawianych funkcji, co może być wykorzystane w przyszłych wersjach programu.
2. Funkcja hipergeometryczna  ${}_2F_1$  jest bardzo trudna do zaimplementowania na podstawie definicji – wiąże się to z poważnymi błędami numerycznymi dla dużych wartości jej parametrów. Autor starał się zminimalizować te błędy poprzez odpowiednie grupowanie jej elementów, jednak w praktyce nie udało się uniknąć ich niepożądanego wpływu na wyniki dla  $N > 50$ . Pozostałe funkcje hipergeometryczne nie sprawiają takich problemów. Dzięki temu obliczenia w bazie Gaussa nie były obciążone omawianym błędem. Być może sposobem obejścia tego problemu byłoby rekurencyjne wyliczanie wartości funkcji.
3. Ze względu na postać wzorów, koszt obliczeniowy programu jest rzędu  $O(\exp N)$ , czyli bardzo wysoki. Wykorzystując opcje profilowania, autor zauważył, że najwięcej czasu zajmuje całkowanie numeryczne, jednak nie ze względu na nieefektywną metodę, ale ze względu na ilość całek do policzenia ( $N^2$  w przypadku nierelatywistycznym oraz  $2N^2$  w relatywistycznym). Ilość tych całek udało się zmniejszyć o połowę (wykorzystano symetryczność macierzy  $V_{mn}$ ), ponadto wykorzystano kształt potencjału użytego do obliczeń (prostokątna studnia, poza którą potencjał się zeruje) do ograniczenia obszaru całkowania. Dzięki tym usprawnieniom udało się znacznie ograniczyć współczynnik proporcjonalności wykładniczej zależności czasu obliczeń od  $N$ . W dalszych pracach nad programem należy dalej zajmować się jego przyspieszeniem, aby móc go w pełni wykorzystać.

Pierwszy i drugi problem nie dotyczy programu Mathematica 3.0, program ten dobrze radzi sobie z bardzo dużymi liczbami, ale okupione jest to nieporównywalnie dłuższym czasem obliczeń, co czyni ten program praktycznie nieprzydatnym dla wartości  $N > 20$ . Obliczenia w Mathematicie mają jednak dużą wartość poglądową, umożliwiają bowiem łatwą obserwację i kontrolę odpowiednich macierzy i wzorów.

## 5.3 Wzory analityczne do przetestowania programu

### 5.3.1 Przypadek nierelatywistyczny

Rozpatrzmy prostokątną studnię potencjału, jak na poniższym rysunku:



Szukamy radialnych części rozwiązań w poszczególnych obszarach w postaci sferycznych funkcji Bessela i Neumanna:

$$\begin{aligned}\Psi_I(r) &= A_1 j_l(\tilde{k}r), \\ \Psi_{II}(r) &= A_2 j_l(\tilde{k}'r) + B_2 n_l(\tilde{k}'r), \\ \Psi_{III}(r) &= A_3 j_l(\tilde{k}r) + B_3 n_l(\tilde{k}r).\end{aligned}$$

W powyższych wzorach  $\tilde{k} = \frac{\sqrt{2m\mathcal{E}}}{\hbar}$ ,  $\tilde{k}' = \frac{\sqrt{2m(\mathcal{E} - V_0)}}{\hbar}$  przy założeniu, że  $\mathcal{E} > V_0$ .

Rozwiązania te należy zszyć, zapewniając ciągłość funkcji i jej pochodnej:

$$\left\{ \begin{array}{l} \Psi_I(a) = \Psi_{II}(a) \\ \frac{d}{dr} \Psi_I(a) = \frac{d}{dr} \Psi_{II}(a) \end{array} \right\}, \quad \left\{ \begin{array}{l} \Psi_{II}(b) = \Psi_{III}(b) \\ \frac{d}{dr} \Psi_{II}(b) = \frac{d}{dr} \Psi_{III}(b) \end{array} \right\}.$$

Tangens przesunięcia fazowego dostajemy w postaci ilorazu:

$$\text{tg } \delta_l = \frac{B_3}{A_3} = \frac{a_3 a_4 (a_9 a_{11} - a_7 a_{13}) + a_2 a_4 (a_8 a_{13} - a_9 a_{12}) + a_1 (a_5 a_9 a_{12} + a_6 a_7 a_{13} - a_6 a_9 a_{11} - a_5 a_8 a_{13})}{a_3 a_4 (a_7 a_{14} - a_{10} a_{11}) + a_2 a_4 (a_{10} a_{12} - a_8 a_{14}) + a_1 (a_5 a_8 a_{14} + a_6 a_{10} a_{11} - a_5 a_{10} a_{12} - a_6 a_7 a_{14})},$$

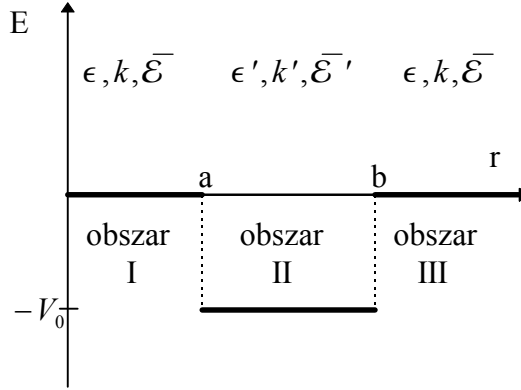
gdzie

$$\begin{aligned}a_1 &= j_l(\tilde{k}a), \quad a_2 = j_l(\tilde{k}'a), \quad a_3 = n_l(\tilde{k}'a), \quad a_4 = j_l'(\tilde{k}a), \quad a_5 = j_l'(\tilde{k}'a), \quad a_6 = n_l'(\tilde{k}'a), \quad a_7 = j_l(\tilde{k}'b), \\ a_8 &= n_l(\tilde{k}'b), \quad a_9 = j_l(\tilde{k}b), \quad a_{10} = n_l(\tilde{k}b), \quad a_{11} = j_l'(\tilde{k}'b), \quad a_{12} = n_l'(\tilde{k}'b), \quad a_{13} = j_l'(\tilde{k}b), \quad a_{14} = n_l'(\tilde{k}b),\end{aligned}$$

a  $j_l'$ ,  $n_l'$  są pochodnymi sferycznych funkcji Bessela i Neumanna.

### 5.3.2 Przypadek relatywistyczny

Rozpatrzmy prostokątną studnię potencjału, jak na poniższym rysunku:



Mamy parametry:  $\bar{\mathcal{E}}'$ ,  $\bar{\mathcal{E}}' \equiv \bar{\mathcal{E}} - V_0$  oraz  $\epsilon(\bar{\mathcal{E}})$ ,  $k(\bar{\mathcal{E}})$  a także  $\epsilon' = \epsilon(\bar{\mathcal{E}}')$  i  $k' = k(\bar{\mathcal{E}}')$ , określone wzorami (5.1.1).

Zdefiniujmy następujące macierze

$$X_r^k = \begin{pmatrix} j_l(kr) & -n_l(kr) \\ \mp \epsilon j_{l\pm 1}(kr) & \pm \epsilon n_{l\pm 1}(kr) \end{pmatrix}, \quad Y_r^k = \begin{pmatrix} \pm \epsilon n_{l\pm 1}(kr) & n_l(kr) \\ \pm \epsilon j_{l\pm 1}(kr) & j_l(kr) \end{pmatrix},$$

gdzie  $j_n(z)$  są sferycznymi funkcjami Bessela pierwszego rodzaju, a  $n_n(z)$  są sferycznymi funkcjami Bessela drugiego rodzaju (sferycznymi funkcjami Neumanna) (patrz dodatek C). W obydwu macierzach górne i dolne znaki dotyczą odpowiednio ujemnych i dodatnich  $\kappa$ .

Radialne części rozwiązań w poszczególnych obszarach możemy zapisać jako

$$\Psi_I(r) = X_r^k \begin{pmatrix} A_1 \\ 0 \end{pmatrix}, \quad \Psi_{II}(r) = X_r^{k'} \begin{pmatrix} A_2 \\ B_2 \end{pmatrix}, \quad \Psi_{III}(r) = X_r^k \begin{pmatrix} A_3 \\ B_3 \end{pmatrix}.$$

Z warunków ciągłości otrzymujemy równania

$$X_a^k \begin{pmatrix} A_1 \\ 0 \end{pmatrix} = X_a^{k'} \begin{pmatrix} A_2 \\ B_2 \end{pmatrix},$$

$$X_b^{k'} \begin{pmatrix} A_2 \\ B_2 \end{pmatrix} = X_b^k \begin{pmatrix} A_3 \\ B_3 \end{pmatrix}.$$

Rozwiązując metodą podstawienia powyższy układ otrzymamy

$$\begin{pmatrix} A_3 \\ B_3 \end{pmatrix} = (X_b^k)^{-1} X_b^{k'} (X_a^{k'})^{-1} X_a^k \begin{pmatrix} A_1 \\ 0 \end{pmatrix}.$$

Przyjmując  $A_1 = 1$  oraz zauważając, że  $(X_r^k)^{-1} = \frac{Y_r^k}{\det X_r^k}$  mamy

$$\begin{pmatrix} A_3 \\ B_3 \end{pmatrix} = \frac{1}{\det X_b^k \det X_b^{k'}} Y_b^k X_b^{k'} Y_a^{k'} X_a^k \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Tangens przesunięcia fazowego znajdujemy w postaci

$$\text{tg } \delta_l = \frac{B_3}{A_3}.$$

## 5.4 Metodologia obliczeń

Autor przeprowadził szereg obliczeń przesunięć fazowych dla różnych energii elektronu, różnych parametrów studni potencjału oraz różnych kombinacji liczb kwantowych, a także dla różnych baz. W tym rozdziale przedstawione są przykładowe wyniki.

Większość obliczeń autor przeprowadził na przypadkowo wybranym potencjale o kształcie prostokątnej studni. Jej głębokość wybrana została na  $V_0 = -1.5\text{au}$ , lewa krawędź na  $a = 1.0\text{au}$ , natomiast prawa na  $b = 1.5\text{au}$ . Energię kinetyczną elektronu padającego ustalono na  $\mathcal{E} = 2.5\text{au}$  (w przypadku relatywistycznym –  $\bar{\mathcal{E}} = 2.5\text{au}$ ). Parametr skalujący  $\lambda$  pozostał jednakowy dla wszystkich obliczeń, jego wartość przyjęto jako 1.0. Taką kombinację parametrów autor będzie dalej nazywał konfiguracją podstawową. Jeżeli nie zaznaczono inaczej, to początkowa wartość  $N$  wynosiła  $N_{start} = 10$ , a końcowa  $N_{end} = 40$ . Ponieważ dla takich wartości  $N$  nie uzyskiwano jeszcze idealnej zbieżności, to wynikiem końcowym była uśredniona wartość przesunięcia fazowego, tzn.

$$\bar{\delta}_N = \frac{1}{N_{end} - N_{start} + 1} \sum_{N=N_{start}}^{N_{end}} \delta_N.$$

Autor przeprowadził obliczenia następującymi metodami:

1. nierelatywistyczną metodą J-macierzy (rozdział 5.5.1),
2. relatywistyczną metodą J-macierzy (rozdział 5.5.2),
3. relatywistyczną metodą J-macierzy w granicy nierelatywistycznej, tzn. dla nieskończonej wartości prędkości światła (rozdział 5.5.3).

W ramach pierwszych dwóch metod, autor przeprowadził następujące testy:

(A) Położenie zerowej głębokości studni potencjału (tzn.  $V_0 = 0$ ). Zgodnie z teorią powinna temu przypadkowi odpowiadać zerowa wartość przesunięcia fazowego. Przeprowadzono testy dla kilku kombinacji liczb kwantowych, a wyniki zebrano w tabelach.



(B) Sprawdzenie, czy i jak numerycznie wyliczone przesunięcia fazowe przy rozpraszaniu na różnych potencjałach w kształcie studni zbiegają się do wartości wyliczonych analitycznie. Obliczenia przeprowadzono dla bazy Gaussa i Laguerre’a, a wyniki zobrazowano na wykresach.

(C) Seria obliczeń dla konfiguracji podstawowej dla różnych kombinacji liczb kwantowych, w obydwu bazach. Wyniki zebrane zostały w tabelach. Zamieszczono przykładowe wykresy dla bazy Gaussa i Laguerre’a.

W ramach trzeciej metody autor zbadał, czy wyniki otrzymywane schematem relatywistycznym, ale przy położeniu nieskończonej wartości prędkości światła, zbiegają się do wyników otrzymanych schematem nierelatywistycznym.

Wszystkie podstawowe obliczenia autor przeprowadził napisanym przez siebie programem w języku Fortran 77, a obliczenia pomocniczo-kontrolne (np. sprawdzenie poprawności wzoru (4.2.21)) w programie Mathematica 3.0.

## 5.5 Wyniki obliczeń<sup>5</sup>

### 5.5.1 Nierelatywistyczna metoda J-macierzy

(A) W poniższej tabeli zamieszczone są wybrane wyniki uzyskane za pomocą programu dla konfiguracji podstawowej, ale z głębokością potencjału  $V_0 = 0$ .

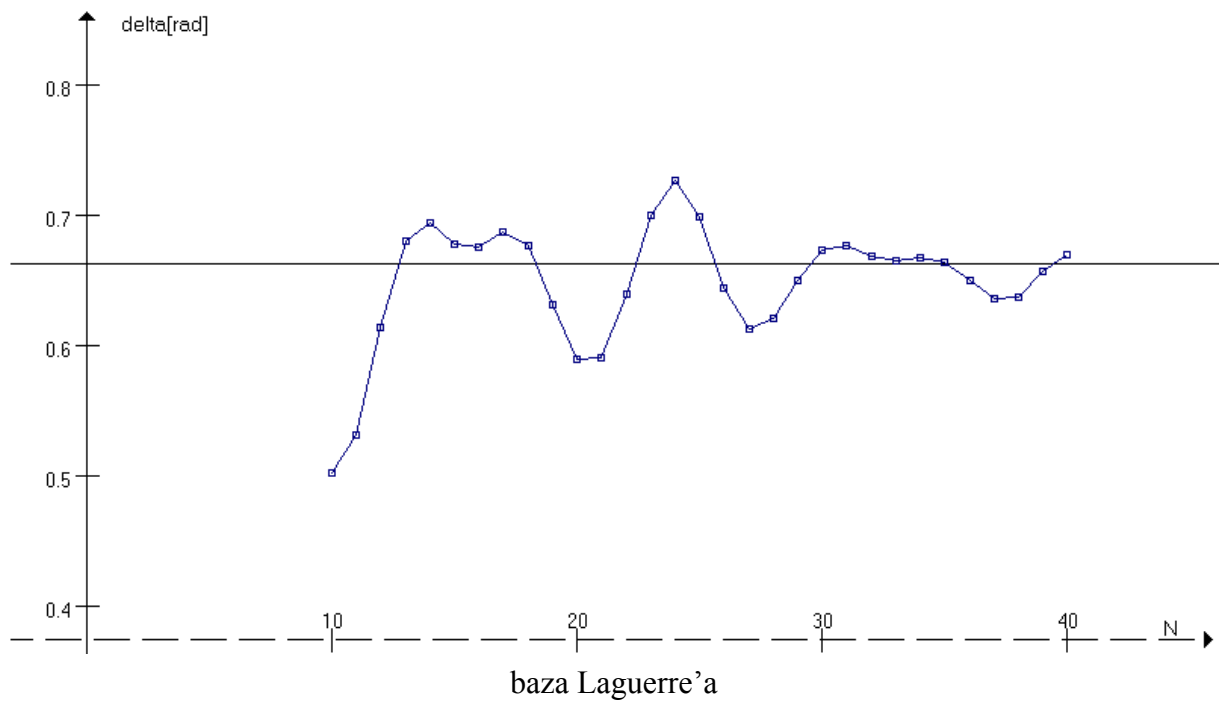
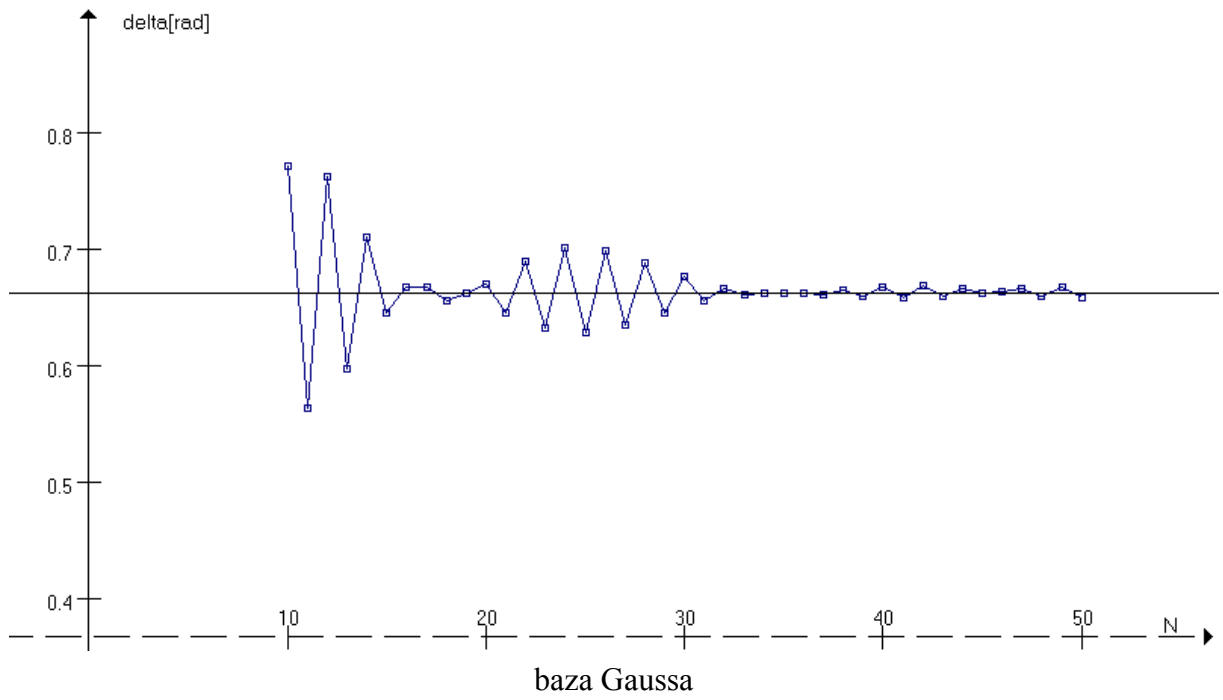
$l$	Baza	$\delta$ [rad] (analitycznie)	$\bar{\delta}_N$ [rad] (numerycznie)
0	gauss	0.0000000000000000E0	3.2920E-12
0	laguerre	0.0000000000000000E0	6.0000E-15
1	gauss	0.0000000000000000E0	9.2710E-12
1	laguerre	0.0000000000000000E0	1.0000E-14
4	gauss	0.0000000000000000E0	1.8309E-11
4	laguerre	0.0000000000000000E0	3.5000E-14

Można uznać, że otrzymane wyniki są zgodne z wynikami analitycznymi.

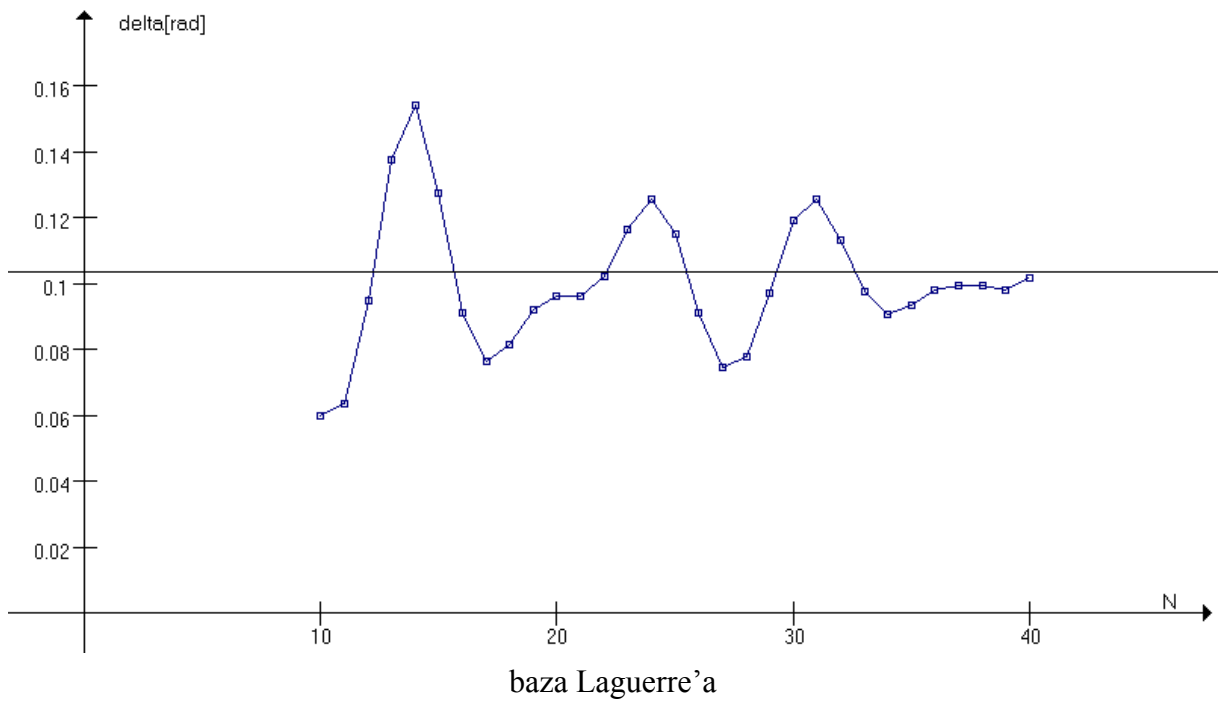
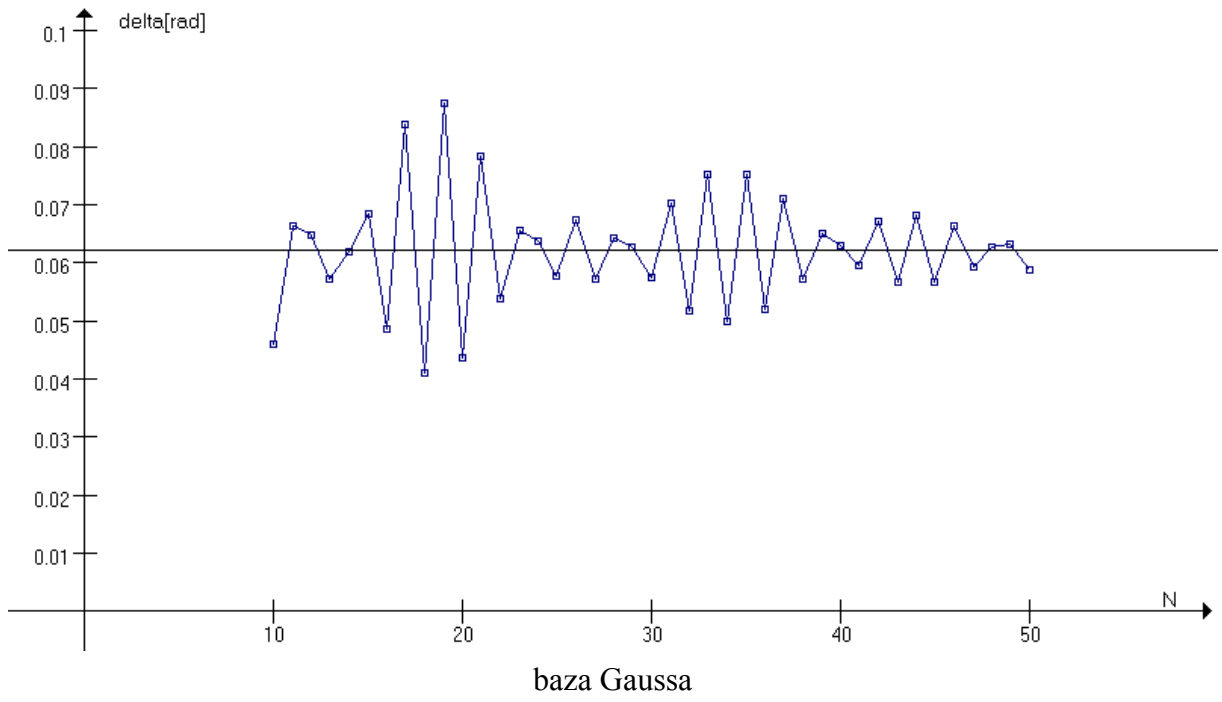
(B) Autor zbadał, czy i w jaki sposób kolejne przybliżenia  $\delta_N$  zbiegają się do wartości wyliczonej analitycznie. Obliczenia przeprowadzono dla przypadkowo wybranych kombinacji parametrów. Ich wyniki przedstawione są na poniższych wykresach. Na osi poziomej zaznaczone są kolejne wartości  $N$ , na osi pionowej  $\delta$ [rad]. Kolejne przybliżenia  $\delta_N$  oznaczone są kwadratami, a wartość analityczna linią prostą ciągłą.

<sup>5</sup> Obliczenia zostały wykonane na komputerze IBM SP/2 w Centrum Informatycznym TASK.

$\mathcal{E} = 0.5\text{au}, l = 0, V_0 = -1.0\text{au}, a = 0.5\text{au}, b = 1.0\text{au}$



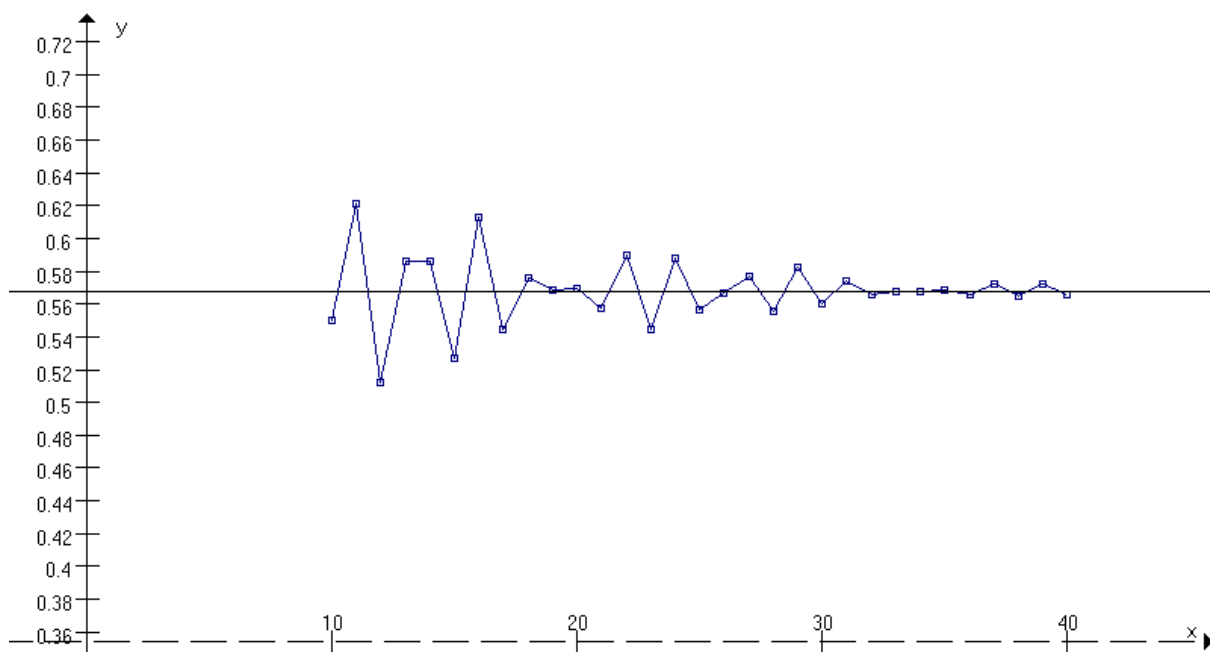
$\mathcal{E} = 0.5 \text{ au}, l = 1, V_0 = -0.5 \text{ au}, a = 1.0 \text{ au}, b = 1.2 \text{ au}$



(C) Systematyczne obliczenia dla konfiguracji podstawowej.

Baza – Gauss:

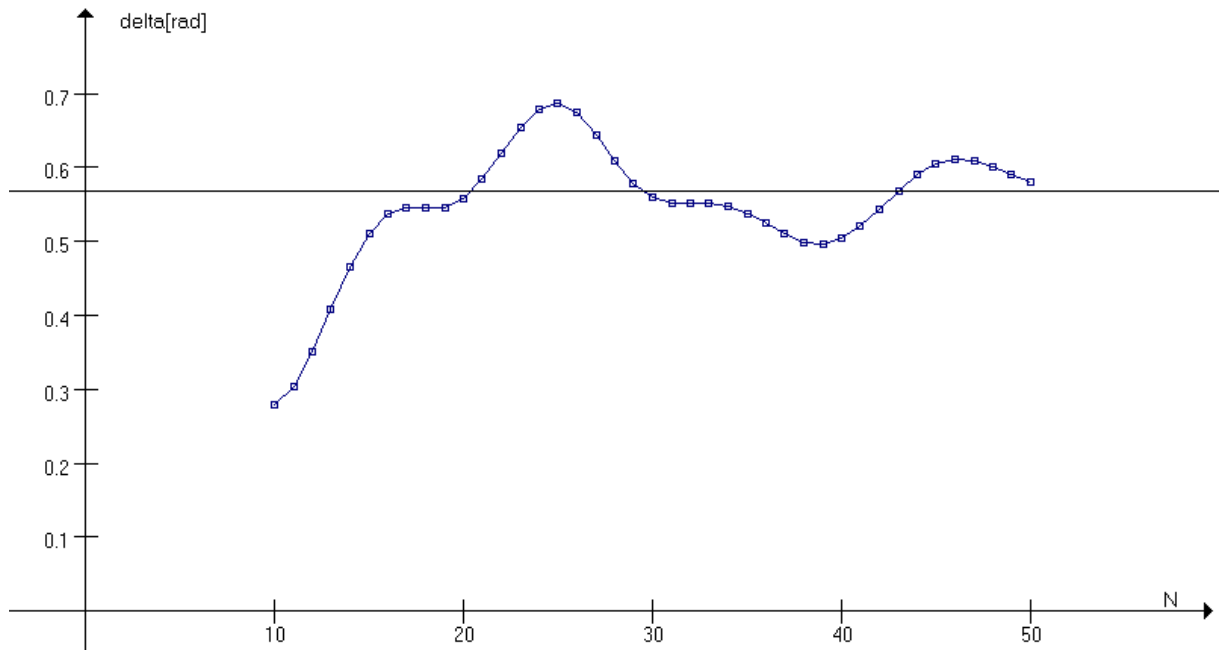
$l$	$\delta[\text{rad}]$ (analitycznie)	$\bar{\delta}_N[\text{rad}]$ (numerycznie)	$\Delta = \bar{\delta}_N - \delta$
0	1.08897176770688E-1	1.09232525867189E-1	3.35E-4
1	5.67545055377277E-1	5.68262677438221E-1	7.18E-4
2	5.99137928636435E-1	5.98604377734328E-1	-5.34E-4
3	1.55084790654800E-1	1.55212847254619E-1	1.28E-4
4	1.82110965544380E-2	1.96517377032165E-2	1.44E-3
5	1.50828438964800E-3	2.89891995385005E-3	1.39E-3
6	9.22636189650000E-5	1.81131895985217E-3	1.72E-3



Wykres dla  $l = 1$

Baza – Laguerre:

$l$	$\delta[\text{rad}]$ (analitycznie)	$\bar{\delta}_N[\text{rad}]$ (numerycznie)	$\Delta = \bar{\delta}_N - \delta$
0	1.08897176770688E-1	1.02765365405441E-1	-6.13E-3
1	5.67545055377277E-1	5.35936853984833E-1	-3.16E-2
2	5.99137928636435E-1	5.93444742996416E-1	-5.69E-3
3	1.55084790654800E-1	1.80803471691081E-1	2.57E-2
4	1.82110965544380E-2	3.20376427905915E-2	1.38E-2
5	1.50828438964800E-3	7.49167626927622E-3	5.98E-3
6	9.22636189650000E-5	2.80219429582033E-3	2.71E-3



Wykres dla  $l = 1$

Okazuje się, że lepszą (szybszą) zbieżnością charakteryzuje się baza Gaussa, chociaż w teście przy zerowej głębokości studni okazała się gorsza od bazy Laguerre'a.

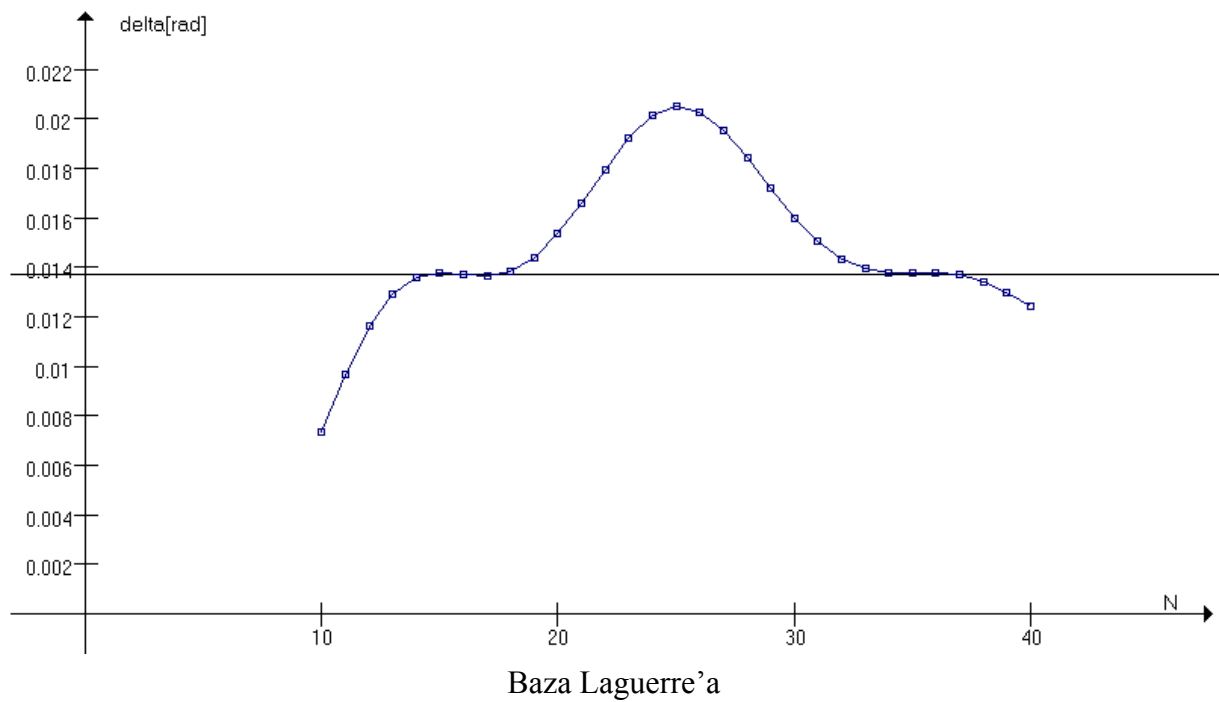
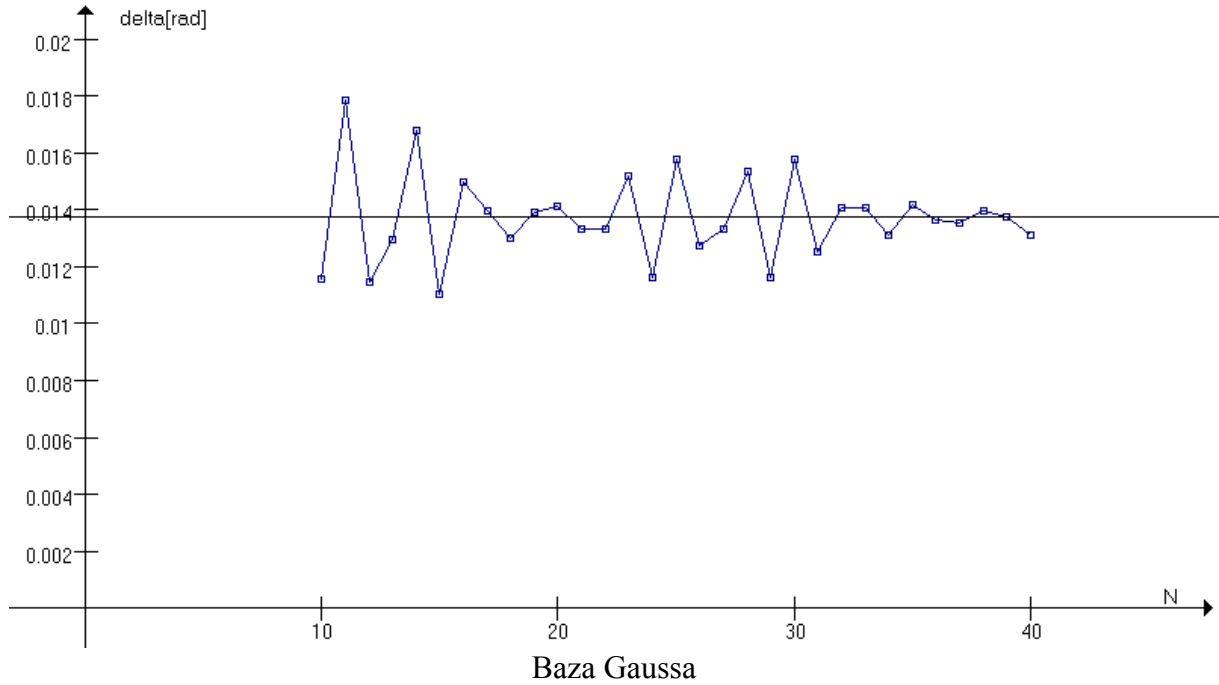
### 5.5.2 Relatywistyczna metoda J-macierzy

(A) Konfiguracja podstawowa,  $V_0 = 0$ .

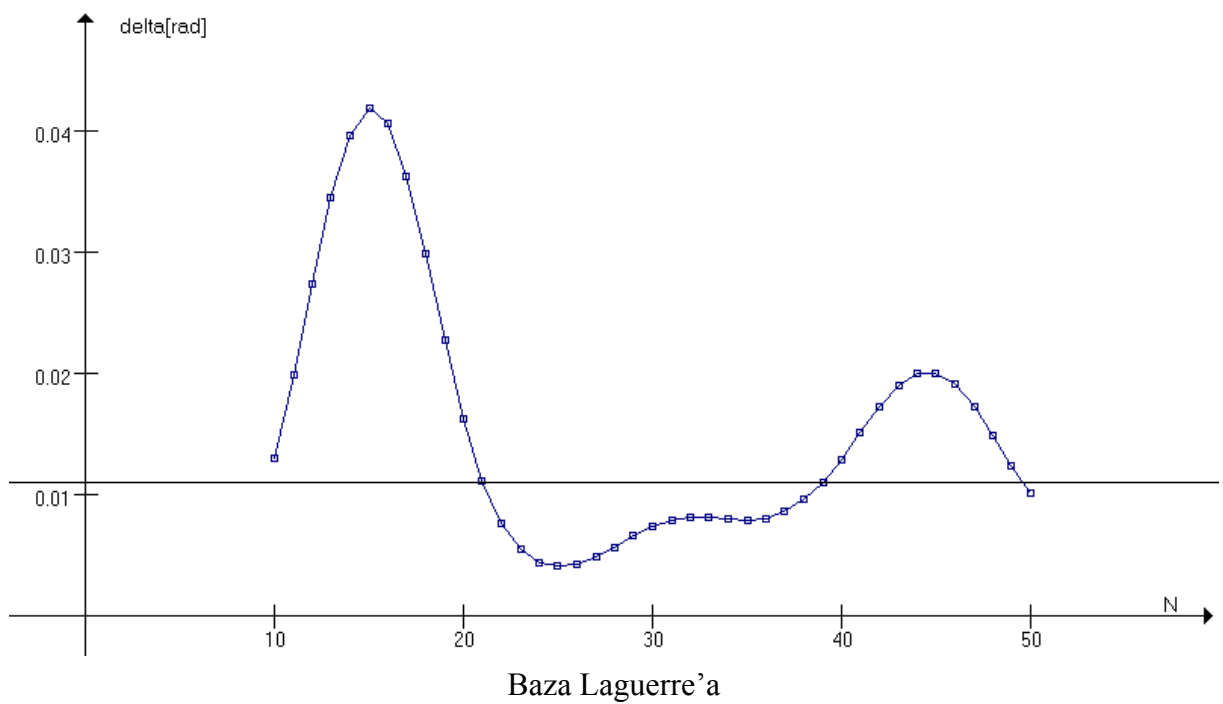
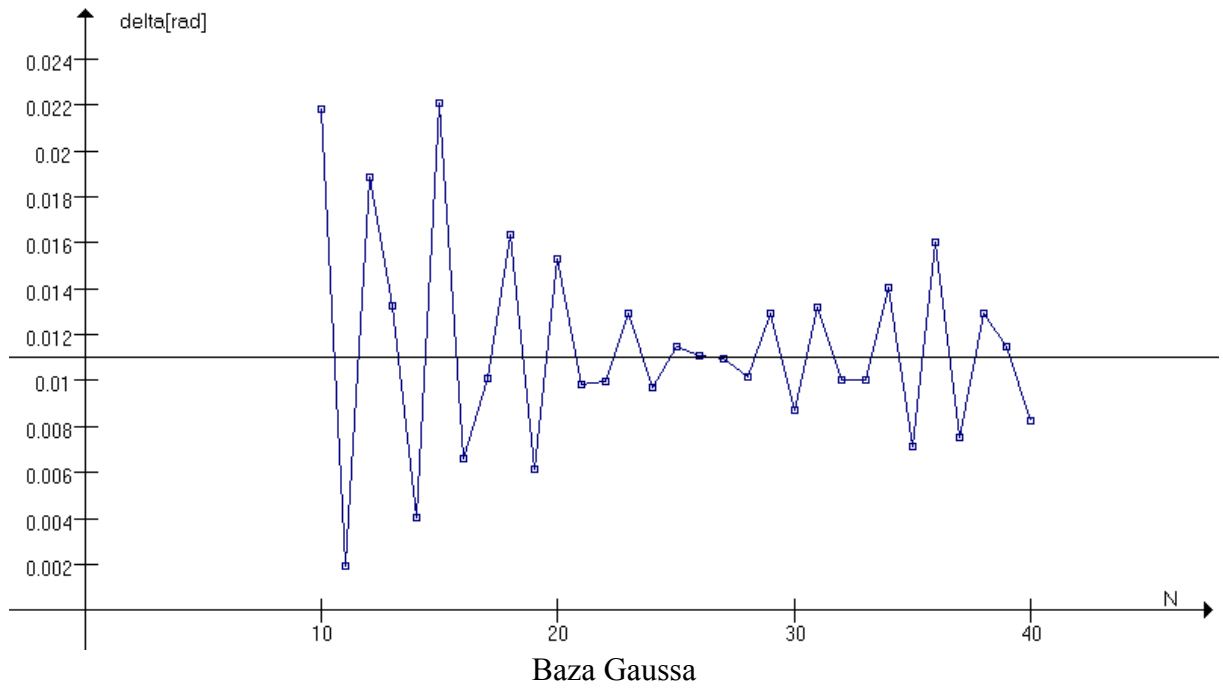
$l$	$\kappa$	Baza	$\delta[\text{rad}]$ (analitycznie)	$\bar{\delta}_N[\text{rad}]$ (numerycznie)
0	-1	gauss	0.0000000000000000E0	5.8720E-12
0	-1	laguerre	0.0000000000000000E0	8.4000E-14
1	1	gauss	0.0000000000000000E0	1.1779E-11
1	1	laguerre	0.0000000000000000E0	3.7600E-13
1	-2	gauss	0.0000000000000000E0	1.1779E-11
1	-2	laguerre	0.0000000000000000E0	3.7600E-13
4	4	gauss	0.0000000000000000E0	9.1226E-11
4	4	laguerre	0.0000000000000000E0	5.8500E-13
4	-5	gauss	0.0000000000000000E0	9.1226E-11
4	-5	laguerre	0.0000000000000000E0	5.8500E-13

(B) Różne studnie potencjałów, wykresy zbieżności.

$$\bar{\mathcal{E}} = 4.5 \text{ au}, l = 2, \kappa = -3, V_0 = -2.5 \text{ au}, a = 1.0 \text{ au}, b = 1.01 \text{ au}$$



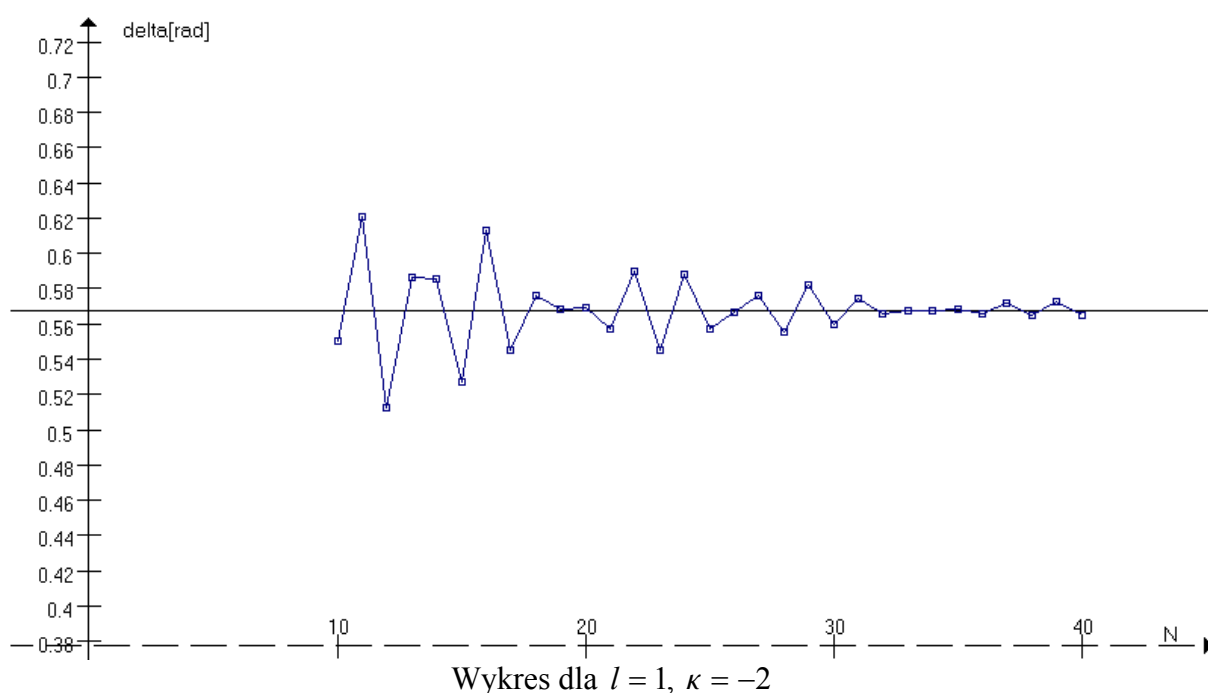
$\bar{\mathcal{E}} = 4.5\text{au}, l = 2, \kappa = -3, V_0 = -2.5\text{au}, a = 0.5\text{au}, b = 0.6\text{au}$



(C) Systematyczne obliczenia dla konfiguracji podstawowej.

Baza – Gauss:

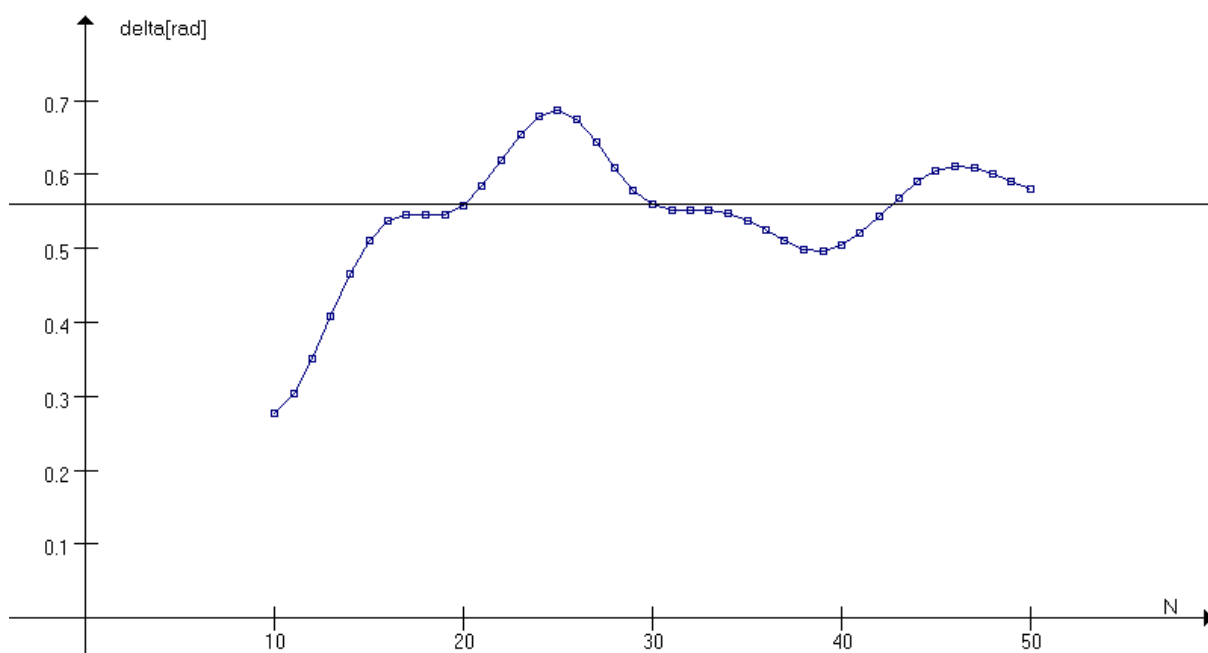
$l$	$\kappa$	$\delta[\text{rad}]$ (analitycznie)	$\bar{\delta}_N[\text{rad}]$ (numerycznie)	$\Delta = \bar{\delta}_N - \delta$
0	-1	1.08927001704795E-1	1.09405833347494E-1	4.79E-4
1	-2	5.67566818088086E-1	5.68264791364169E-1	6.98E-4
1	1	5.67538166974585E-1	5.68254420487773E-1	7.16E-4
2	-3	5.99230480054382E-1	5.98688907626321E-1	-5.42E-4
2	2	5.99258747136310E-1	5.98735081722034E-1	-5.24E-4
3	-4	1.55130447591260E-1	1.55092074555394E-1	-3.84E-5
3	3	1.55175009560722E-1	1.55135931741479E-1	-3.91E-5



Baza – Laguerre:

$l$	$\kappa$	$\delta[\text{rad}]$ (analitycznie)	$\bar{\delta}_N[\text{rad}]$ (numerycznie)	$\Delta = \bar{\delta}_N - \delta$
0	-1	1.08927001704795E-1	1.02796700776405E-1	-6.13E-3
1	-2	5.67566818088086E-1	5.35955153464449E-1	-3.16E-2
1	1	5.67538166974585E-1	5.35925186761733E-1	-3.16E-2
2	-3	5.99230480054382E-1	5.93525830832780E-1	-5.70E-3
2	2	5.99258747136310E-1	5.93545907602457E-1	-5.71E-3
3	-4	1.55130447591260E-1	1.81341288201869E-1	2.62E-2
3	3	1.55175009560722E-1	1.80896168344248E-1	2.57E-2





Wykres dla  $l = 1, \kappa = -2$

Tutaj też baza Gaussa okazała się lepsza, przynajmniej w badanym zakresie  $N$ .

### 5.5.3 Relatywistyczna metoda J-macierzy w granicy nierelatywistycznej

$l$	$\kappa$	Baza	$\bar{\delta}_N$ [rad] numerycznie relatywistycznie	$\bar{\delta}_N$ [rad] numerycznie relatywistycznie granica nierelatywistyczna	$\bar{\delta}_N$ [rad] numerycznie nierelatywistycznie
0	-1	gauss	1.09405833347494E-1	1.09252221868101E-1	1.09232525867189E-1
0	-1	laguerre	1.02796700776405E-1	1.02765475435248E-1	1.02765365405441E-1
1	1	gauss	5.68254420487773E-1	5.68259957438321E-1	5.68262677438221E-1
1	1	laguerre	5.35925186761733E-1	5.35928992784353E-1	5.35936853984833E-1
2	2	gauss	5.98735081722034E-1	5.98609373234608E-1	5.98604377734328E-1
2	2	laguerre	5.93545907602457E-1	5.93499844946110E-1	5.93444742996416E-1

Wyniki dla  $\kappa = l$  i  $\kappa = -l - 1$  okazały się identyczne, dlatego autor nie zamieścił ich w tabeli. Z tabeli można wywnioskować, że wyniki relatywistyczne w granicy nierelatywistycznej zbieżają do wyników nierelatywistycznych, co dowodzi poprawności rozważań pod koniec rozdziału 4.2.

Z wyników obliczeń omówionych w tym rozdziale widać, że obliczenia metodą J-macierzy są dość dokładne dla liczb kwantowych  $l = 0 - 3$ . Autor zbadał, że dla większych  $l$  wyniki numeryczne stają się coraz bardziej niedokładne – w dalszym ciągu są zbieżne do wartości analitycznych, ale ta zbieżność jest coraz wolniejsza. Jest to spowodowane tym, że

bezwzględne wartości przesunięć fazowych są wtedy zwykle coraz mniejsze i nakłada się na nie błąd spowodowany obciążeniem potencjału, dlatego trzeba wykonywać obliczenia w większej bazie. Okazuje się, że nawet dla prostych potencjałów błąd ten ma duży wpływ na wyniki obliczeń i powoduje pogarszanie się zbieżności. Autor podejrzewa, że gdyby udało się przeprowadzić obliczenia dla większych  $N$  (rzędu 100-200), to jego wpływ byłby dużo mniejszy. Ponadto, wykorzystując fakt, że wzór na tangens przesunięcia fazowego jest wzorem analitycznym, można do niego zastosować standardowe poprawki wariacyjne omówione w [1].

## 6. Podsumowanie

Zasadniczym celem pracy było zaprogramowanie metody J-macierzy i praktyczne sprawdzenie, czy stosowanie jej do opisu zderzeń daje dobre rezultaty. W tym celu autor napisał program komputerowy i przeprowadził za jego pomocą obliczenia przesunięć fazowych przy rozpraszaniu na potencjałach o kształcie prostokątnej studni. Wynika z nich, że metoda jest skuteczna, ale dla takich parametrów, dla których przesunięcie fazowe ma dość dużą wartość bezwzględną. Błąd wynikający z przybliżenia potencjału jest wtedy mniejszego rzędu niż wynik i nie wpływa nań znacząco. Okazało się również, że na zbieżność wyników numerycznych ma duży wpływ rodzaj bazy użytej do obliczeń. Generalnie, lepsza okazała się baza Gaussa, chociaż, jak widać z wykresów, można i po bazie Laguerre'a spodziewać się dobrych rezultatów, ale dla większych  $N$ . Ponadto pozostaje do sprawdzenia, w jaki sposób czynnik skalujący  $\lambda$  wpływa na wynik. Trudno będzie jednak wtedy porównywać wyniki z baz Gaussa i Laguerre'a, ze względu na inny wymiar czynnika skalującego w tych bazach.

Po udoskonaleniu programu (przyspieszenie działania, zmniejszenie wpływu błędów wynikających z przybliżania potencjału, zwiększenie stosowalności do kilkukrotnie większych  $N$ ) może on być wykorzystany do relatywistycznych obliczeń na rzeczywistych potencjałach.

## 7. Dodatek

### A. Własności wielomianów Legendre'a oraz harmonik sferycznych

Wielomiany Legendre'a definiujemy następująco:

$$P_l(\cos\theta) = \frac{1}{2^l l!} \frac{d^l}{d(\cos\theta)^l} [(\cos^2\theta - 1)^l].$$

Stowarzyszone funkcje Legendre'a są związane z wielomianami Legendre'a relacją

$$P_l^m(\cos\theta) = (-)^m \sin^m \theta \frac{d^m P_l(\cos\theta)}{d(\cos\theta)^m}.$$

Dla stowarzyszonych funkcji Legendre'a zachodzi wzór

$$P_{-l-1}^m(\cos\theta) = P_l^m(\cos\theta).$$

Harmoniki sferyczne definiujemy następująco:

$$Y_l^m(\hat{\mathbf{r}}) = \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} P_l^m(\cos\theta) e^{im\varphi} \quad \text{dla } m \geq 0,$$

$$Y_l^{-m}(\hat{\mathbf{r}}) = (-)^m Y_l^{m*}(\hat{\mathbf{r}}) \quad \text{dla } m < 0.$$

Dla tak zdefiniowanych harmonik zachodzą poniższe wzory:

$$Y_l^m(-\hat{\mathbf{r}}) = (-)^l Y_l^m(\hat{\mathbf{r}}),$$

$$\Lambda_{\pm} Y_l^m(\hat{\mathbf{r}}) = \sqrt{l(l+1) - m(m \pm 1)} Y_l^{m \pm 1}(\hat{\mathbf{r}}) = \sqrt{(l \mp m)(l \pm m + 1)} Y_l^{m \pm 1}(\hat{\mathbf{r}}), \quad (\text{A.1})$$

gdzie

$$\Lambda_{\pm} = \Lambda_x \pm i \Lambda_y.$$

Przypomnijmy, że  $\Lambda = \frac{\mathbf{L}}{\hbar}$ , przy czym  $\mathbf{L}$  jest operatorem orbitalnego momentu pędu.

Zdefiniujmy następujące symbole różniczkowe

$$\partial_z = \frac{\partial}{\partial z}, \quad \partial_+ = \frac{\partial}{\partial x} + i \frac{\partial}{\partial y}, \quad \partial_- = \frac{\partial}{\partial x} - i \frac{\partial}{\partial y}.$$

Zachodzą wzory

$$\begin{aligned} \partial_z \left[ \frac{1}{r} F(r) Y_l^m(\hat{\mathbf{r}}) \right] &= \sqrt{\frac{(l-m+1)(l+m+1)}{(2l+1)(2l+3)}} \frac{1}{r} \left( \frac{d}{dr} F(r) - \frac{l+1}{r} F(r) \right) Y_{l+1}^m(\hat{\mathbf{r}}) \\ &+ \sqrt{\frac{(l+m)(l-m)}{(2l-1)(2l+1)}} \frac{1}{r} \left( \frac{d}{dr} F(r) + \frac{l}{r} F(r) \right) Y_{l-1}^m(\hat{\mathbf{r}}), \end{aligned} \quad (\text{A.2})$$

$$\begin{aligned} \partial_+ \left[ \frac{1}{r} F(r) Y_l^m(\hat{\mathbf{r}}) \right] &= -\sqrt{\frac{(l+m+1)(l+m+2)}{(2l+1)(2l+3)}} \frac{1}{r} \left( \frac{d}{dr} F(r) - \frac{l+1}{r} F(r) \right) Y_{l+1}^{m+1}(\hat{\mathbf{r}}) \\ &+ \sqrt{\frac{(l-m-1)(l-m)}{(2l-1)(2l+1)}} \frac{1}{r} \left( \frac{d}{dr} F(r) + \frac{l}{r} F(r) \right) Y_{l-1}^{m+1}(\hat{\mathbf{r}}), \end{aligned}$$

$$\begin{aligned} \partial_- \left[ \frac{1}{r} F(r) Y_l^m(\hat{\mathbf{r}}) \right] &= \sqrt{\frac{(l-m+1)(l-m+2)}{(2l+1)(2l+3)}} \frac{1}{r} \left( \frac{d}{dr} F(r) - \frac{l+1}{r} F(r) \right) Y_{l+1}^{m-1}(\hat{\mathbf{r}}) \\ &+ \sqrt{\frac{(l+m-1)(l+m)}{(2l-1)(2l+1)}} \frac{1}{r} \left( \frac{d}{dr} F(r) + \frac{l}{r} F(r) \right) Y_{l-1}^{m-1}(\hat{\mathbf{r}}). \end{aligned} \quad (\text{A.3})$$

## B. Sferyczne funkcje Bessela, funkcje Riccati-Bessela

Rozpatrzmy równanie

$$z^2 \frac{d^2 w}{dz^2} + 2z \frac{dw}{dz} + [z^2 - n(n+1)]w = 0 \quad \text{dla } n = 0, \pm 1, \pm 2, \dots$$

Jego rozwiązaniami są sferyczne funkcje Bessela:

- pierwszego rodzaju:  $j_n(z) = \sqrt{\frac{\pi}{2z}} J_{n+1/2}(z)$ ,
- drugiego rodzaju:  $y_n(z) = \sqrt{\frac{\pi}{2z}} Y_{n+1/2}(z)$ ,
- trzeciego rodzaju:

$$h_n^{(1)}(z) = \sqrt{\frac{\pi}{2z}} H_{n+1/2}^{(1)}(z) = j_n(z) + i y_n(z),$$

$$h_n^{(2)}(z) = \sqrt{\frac{\pi}{2z}} H_{n+1/2}^{(2)}(z) = j_n(z) - i y_n(z).$$

Funkcje Bessela  $J_\mu(z)$ ,  $Y_\mu(z)$  pierwszego i drugiego rodzaju dla rzeczywistego  $\mu$  można zdefiniować następująco

$$J_\mu(z) = \left( \frac{1}{2} z \right)^\mu \sum_{k=0}^{\infty} \frac{(-z^2/4)^k}{k! \Gamma(\mu + k + 1)},$$

$$Y_\mu(z) = \frac{J_\mu(z)\cos(\mu\pi) - J_{-\mu}(z)}{\sin(\mu\pi)}.$$

Funkcje Bessela drugiego rodzaju nazywane są często funkcjami Neumanna, a sferyczne funkcje Bessela drugiego rodzaju – sferycznymi funkcjami Neumanna.

Dla funkcji  $j_n(z)$  zachodzi wzór

$$\begin{aligned} j_n(z) &= f_n(z)\sin z + (-)^{n+1}f_{-n-1}(z)\cos z \\ f_0(z) &= z^{-1}, \quad f_1(z) = z^{-2} \\ f_{n-1}(z) + f_{n+1}(z) &= (2n+1)z^{-1}f_n(z) \quad \text{dla } n = 0, \pm 1, \pm 2, \dots \end{aligned}$$

Dla funkcji  $y_n(z)$  zachodzi wzór

$$y_n(z) = (-)^{n+1}j_{-n-1}(z) \quad \text{dla } n = 0, \pm 1, \pm 2, \dots$$

Funkcje Riccati-Bessela są rozwiązaniami równania

$$z^2 \frac{d^2 w}{dz^2} + [z^2 - n(n+1)]w = 0$$

i wyrażają się przez sferyczne funkcje Bessela w następujący sposób

$$\begin{aligned} \hat{j}_n(z) &= zj_n(z), \\ \hat{y}_n(z) &= zy_n(z) \quad (\text{nazywane często funkcjami Riccati-Neumanna}), \\ \hat{h}_n^{(1)}(z) &= zh_n^{(1)}(z), \quad \hat{h}_n^{(2)}(z) = zh_n^{(2)}(z). \end{aligned}$$

Dla funkcji Riccati-Bessela zachodzą wzory rekurencyjne

$$\frac{d}{dx}\hat{z}_n(x) + \frac{l}{x}\hat{z}_n(x) = \hat{z}_{n-1}(x), \quad (\text{B.1})$$

$$\frac{d}{dx}\hat{z}_n(x) - \frac{l+1}{x}\hat{z}_n(x) = -\hat{z}_{n+1}(x), \quad (\text{B.2})$$

gdzie  $\hat{z}_n(x)$  jest równe  $\hat{j}_n(x)$  lub  $\hat{y}_n(x)$ .

### C. Wybrane elementy baz Laguerre'a i Gaussa

W poniższej tabeli zebrane zostały elementy rozwinięcia rozwiązań typu sinus i kosinus w bazie Laguerre'a i Gaussa oraz same funkcje bazowe.  $L_n^{(\alpha)}$  i  $C_n^{(\alpha)}$  są odpowiednio wielomianami Laguerre'a i Gegenbauera,  ${}_2F_1$  i  ${}_1F_1$  są odpowiednimi funkcjami hipergeometrycznymi,  $\lambda > 0$  jest parametrem skalującym.

Wielkość	Baza Laguerre'a	Baza Gaussa
$\phi_n^l$	$(\lambda r)^{l+1} \exp\left(-\frac{\lambda r}{2}\right) L_n^{(2l+1)}(\lambda r)$	$(\lambda r)^{l+1} \exp\left(-\frac{\lambda^2 r^2}{2}\right) L_n^{(l+1/2)}(\lambda^2 r^2)$
$\bar{\phi}_n^l$	$\frac{n!}{\lambda^2 \Gamma(n+2l+2)} \frac{1}{r} \phi_n^l$	$\frac{2n!}{\lambda^2 \Gamma(n+l+3/2)} \phi_n^l$
$s_n^l$	$\frac{2^l \Gamma(l+1)n!(\sin\theta)^{l+1}}{\Gamma(n+2l+2)} C_n^{(l+1)}(\cos\theta)$  $\cos\theta \equiv \frac{k^2 \lambda^{-2} - 1/4}{k^2 \lambda^{-2} + 1/4}$  $\sin\theta \equiv \frac{k\lambda^{-1}}{k^2 \lambda^{-2} + 1/4}$	$\frac{\sqrt{2\pi}n!(-1)^n}{\Gamma(n+l+3/2)} \exp\left(-\frac{\eta^2}{2}\right) L_n^{(l+1/2)}(\eta^2) \eta^{l+1}$
$c_n^l$	$\frac{-2^l \Gamma(l+1/2)n!}{\sqrt{\pi}\Gamma(n+2l+2)(\sin\theta)^l}$ $\times {}_2F_1\left(-n-2l-1, n+1, \frac{1}{2}-l; \sin^2\left(\frac{\theta}{2}\right)\right)$  $\sin^2(\theta/2) \equiv \frac{\lambda^2}{4k^2 + \lambda^2}$	$\frac{\sqrt{2/\pi}\Gamma(l+1/2)(-1)^n n!}{\Gamma(n+l+3/2)} \exp\left(-\frac{\eta^2}{2}\right) \eta^{-l}$  $\times {}_1F_1\left(-n-l-\frac{1}{2}, \frac{1}{2}-l; \eta^2\right)$  $\eta \equiv \frac{k}{\lambda}$
$\langle \phi_m^l   \phi_n^l \rangle$	$\frac{\Gamma(n+2l+2)}{\lambda n!} \left[ 2(n+l+1)\delta_{mn} - n\delta_{m,n-1} - (n+2l+2)\delta_{m,n+1} \right]$	$\frac{\Gamma(n+2l+3/2)}{2n!} \delta_{mn}$
$\langle \psi_m^l   \psi_n^l \rangle$	$\frac{\Gamma(n+2l+2)}{4n!} \left[ 2(n+l+1)(2\lambda-1)\delta_{mn} + n\delta_{m,n-1} + (n+2l+2)\delta_{m,n+1} \right]$	$\frac{\lambda^2 \Gamma(n+l+3/2)}{2n!} \left[ (2n+l+3/2)\delta_{mn} + n\delta_{m,n-1} + (n+l+3/2)\delta_{m,n+1} \right]$

## D. Kody źródłowe programów

### D.1 Fortran 77

```
PROGRAM Metoda_J_Macierzy
INTEGER n, Nstart, Nend
REAL*8 delta, tan_delta, aver_delta
REAL*8 sum
REAL*8 t, t_analitic
CHARACTER*8 fout, type
CHARACTER*20 fin

COMMON /N/n /steer/fout,Nstart,Nend /potential_type/type

10 FORMAT (A3,I3,A3,F20.15,A4,F20.15,A6,F20.15)
20 FORMAT (I3,3F22.18)
30 FORMAT (I3,A2,F22.18)

* Nazwa pliku z danymi wejsciowymi
fin = 'j-matrix.dat'

* Wczytanie parametrow z pliku
CALL LoadParams(fin)

* Glowny plik wynikowy
OPEN(1,FILE='results'//fout)

* Dodatkowy plik wynikowy (format Comma Separated Variables -
CSV)
OPEN(2,FILE='results'//fout/'.csv')
WRITE(2,*) 'N, delta[rad]'

* Przepisanie uzytych parametrow do pliku wynikowego
CALL SaveParams(fin)

IF (type .EQ. 'well') THEN

* Analityczna wartosc przesunecia fazowego dla studni potencjalu
tan_delta = t_analitic()
delta = atan(tan_delta)

WRITE(*,*)
WRITE(*, '(A,F18.15)') 'Wynik analityczny: d= ', delta
WRITE(1, '(A19,F20.15)') '# ANALITICAL RESULT'
WRITE(1, '(A19)') '# -----'
WRITE(1, '(A7,F20.15)') '# delta=', delta
WRITE(1, '(A1)') '#

ENDIF

WRITE(*,*) 'Wyniki numeryczne:'

WRITE(1, '(A57)')
& '# NUMERICAL RESULTS: n tan(delta) delta average(delta)'
WRITE(1, '(A57)')
& '# -----'

sum = 0.d0

* Glowna petla programu
* -----

DO n = Nstart, Nend

* Obliczenie wartosci tan(delta) i delta dla danego n
tan_delta = t(n)
delta = atan(tan_delta)

* Obliczenie sredniej wartosci przesunecia fazowego
sum = sum + delta
aver_delta = sum / (n-Nstart+1)

* Zapisanie wynikow do plikow i na ekran
WRITE(*,10) 'n=', n, 't=', tan_delta, 'd= ', delta, 'av= ', aver_delta
WRITE(1,20) n, tan_delta, delta, aver_delta
WRITE(2,30) n, ', ', delta

ENDDO

* -----

* Koniec glownej petli programu

CLOSE(1)
CLOSE(2)

END

*****

REAL*8 FUNCTION t(N)

* Funkcja wylicza numeryczna wartosc tangensa przesunecia
fazowego

REAL*8 s, c, g, jm
REAL*8 tmp, gN, jmN
REAL*8 k, Eps, v_light
INTEGER N, n_trunc
CHARACTER*16 scheme, tmp_scheme

COMMON /N/n_trunc /scheme/scheme /k/k /Eps/Eps /v_light/v_light

n_trunc = N
gN = g(N-1,N-1)

* We wzorze na t wystepuja elementy Jm nierelatywistyczne -
* tymczasowo "oszukujemy" procedure liczaca elementy J-macierzy
tmp_scheme = scheme
scheme = 'non-relativistic'
jmN = jm(N,N-1)
scheme = tmp_scheme

IF (scheme .EQ. 'relativistic') THEN
tmp = 2.d0 * Eps * v_light / k
ELSE
tmp = 1.d0
ENDIF

t = - ( s(N-1) + tmp * gN * jmN * s(N) ) /
& ( c(N-1) + tmp * gN * jmN * c(N) )

RETURN
END

*****

REAL*8 FUNCTION t_analitic()

* Funkcja oblicza tangens przesunecia fazowego wg. wzorow
* analitycznych (patrz rozdzial 5.3)

REAL*8 X1(2,2), X2(2,2), Y1(2,2), Y2(2,2), il1(2,2), il2(2,2)
REAL*8 k, kpr, l
REAL*8 Eps, Epspr
REAL*8 v0, a, b
REAL*8 ka, kb, k1a, k1b
REAL*8 a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14
REAL*8 sphbess, sphneum, sphbess_prim, sphneum_prim
INTEGER l
CHARACTER*8 type
CHARACTER*16 scheme

COMMON /k/k /kpr/kpr /potential_type/type /well/v0,a,b
COMMON /scheme/scheme /Eps/Eps /Epspr/Epspr //l

IF (scheme .EQ. 'relativistic') THEN

* Sformowanie odpowiednich macierzy
CALL formY (Y1, b, k, Eps)
CALL formX (X1, b, kpr, Epspr)
CALL formY (Y2, a, kpr, Epspr)
CALL formX (X2, a, k, Eps)

* Wymnozenie macierzy
CALL multiply(2,Y2,X2,il1)
CALL multiply(2,X1,il1,il2)
CALL multiply(2,Y1,il2,il1)
```

```

*      Wynik jako stosunek odpowiednich współczynników
      t_analitic = il1(2,1) / il1(1,1)

ELSE

*      Zmienne pomocnicze
      ka = k * a
      kb = k * b
      k1a = kpr * a
      k1b = kpr * b
      l_ = dble(l)

      a1 = sphbess(l_, ka)
      a2 = sphbess(l_, k1a)
      a3 = sphneum(l_, k1a)
      a4 = sphbess_prim(l_, ka)
      a5 = sphbess_prim(l_, k1a)
      a6 = sphneum_prim(l_, k1a)
      a7 = sphbess(l_, k1b)
      a8 = sphneum(l_, k1b)
      a9 = sphbess(l_, kb)
      a10 = sphneum(l_, kb)
      a11 = sphbess_prim(l_, k1b)
      a12 = sphneum_prim(l_, k1b)
      a13 = sphbess_prim(l_, kb)
      a14 = sphneum_prim(l_, kb)

      t_analitic = ( a3*a4*(a9*a11-a7*a13)+a2*a4*(a8*a13-a9*a12) +
&      a1*(a5*a9*a12+a6*a7*a13-a6*a9*a11-a5*a8*a13) ) /
&      ( a3*a4*(a7*a14-a10*a11)+a2*a4*(a10*a12-a8*a14) +
&      a1*(a5*a8*a14+a6*a10*a11-a5*a10*a12-a6*a7*a14) )

ENDIF

RETURN
END

*****

SUBROUTINE formX(X,r,k,Eps)

REAL*8 X(2,2)
REAL*8 r, k, Eps
REAL*8 kr, l_
REAL*8 sphneum, sphbess
INTEGER l, kappa

COMMON //l /kappa/kappa

kr = k * r
l_ = dble(l)

X(1,1) = sphbess(l_, kr)
X(1,2) = -sphneum(l_, kr)
IF (kappa .GT. 0) THEN
  X(2,1) = Eps * sphbess(l_-1, kr)
  X(2,2) = -Eps * sphneum(l_-1, kr)
ELSE
  X(2,1) = -Eps * sphbess(l_+1, kr)
  X(2,2) = Eps * sphneum(l_+1, kr)
ENDIF

RETURN
END

*****

SUBROUTINE formY(Y,r,k,Eps)

REAL*8 Y(2,2)
REAL*8 r, k, Eps
REAL*8 kr, l_
REAL*8 sphneum, sphbess
INTEGER l, kappa

COMMON //l /kappa/kappa

kr = k * r
l_ = dble(l)

Y(1,2) = sphneum(l_, kr)
Y(2,2) = sphbess(l_, kr)
IF (kappa .GT. 0) THEN

```

```

      Y(1,1) = -Eps * sphneum(l_-1, kr)
      Y(2,1) = -Eps * sphbess(l_-1, kr)
ELSE
      Y(1,1) = Eps * sphneum(l_+1, kr)
      Y(2,1) = Eps * sphbess(l_+1, kr)
ENDIF

RETURN
END

*****

REAL*8 FUNCTION fi(n,r)

*      Funkcja bazowa

REAL*8 lambda, r
REAL*8 lr, lr2
REAL*8 laguerre
INTEGER n, l
CHARACTER*8 basis

COMMON /lambda/lambda /basis/ basis //l

*      Zmienne pomocnicze
      lr = lambda * r
      lr2 = lambda**2 * r**2

IF (basis .EQ. 'laguerre') THEN

*      Baza Laguerre'a
      fi = lr**(l+1) * exp(-lr/2.d0) * laguerre(n,2*l+1.d0,lr)

ELSE

*      Baza Gaussa
      fi = lr**(l+1) * exp(-lr/2.d0) * laguerre(n,l+0.5d0,lr2)

ENDIF

RETURN
END

*****

REAL*8 FUNCTION psi(n,r)

*      Funkcja bazowa

REAL*8 r, lr, lr2, lambda, k
REAL*8 fi, laguerre
INTEGER n, l, kappa
CHARACTER*8 basis

COMMON /lambda/lambda /k/k /basis/basis /kappa/kappa //l

lr = lambda * r

IF (basis .EQ. 'laguerre') THEN

  psi = kappa / r * fi(n,r) + exp(-lr/2.d0) * (
& lambda**(l+1) * (l+1) * r**l * laguerre(n,2*l+1.d0,lr) -
& lr**(l+1) * lambda/2.d0 * laguerre(n,2*l+1.d0,lr) -
& lr**(l+1) * (laguerre(n,2*l+2.d0,lr)-laguerre(n,2*l+1.d0,lr)) )

ELSE

  lr2 = lr * lr
  psi = kappa / r * fi(n,r) + exp(-lr/2.d0) * (
& lambda**(l+1) * (l+1) * r**l * laguerre(n,l+0.5d0,lr2)-
& lr**(l+1) * lambda**2 * r * laguerre(n,l+0.5d0,lr2) -
& lr**(l+1) * (laguerre(n,l+1.5d0,lr2)-laguerre(n,l+0.5d0,lr2)))

ENDIF

RETURN
END

*****

REAL*8 FUNCTION s(n)

*      Funkcja oblicza współczynniki rozwinięcia rozwiązania typu sinus

```



```

INTEGER n, l
REAL*8 lambda, k
REAL*8 gamma, gegen, laguerre, fact, fact1
REAL*8 sint, cost, pi, eta
CHARACTER*8 basis

PARAMETER (pi=3.1415926535897932384626434d0)

COMMON /lambda/lambda /k/k /basis/basis //l

* Zmienne pomocnicze
sint = k / ( k*k/lambda + lambda/4.d0)
cost = (k**2 / lambda**2 - 0.25d0)/(k**2 / lambda**2 + 0.25d0)
eta = k / lambda

IF (basis .EQ. 'laguerre') THEN
  s = 2**l * gamma(l+1.d0) * fact1(n,n+2*l+1)
  & * sint**(l+1) * gegen(n, l+1.d0, cost)
ELSE
  s = sqrt(2.d0*pi) * (fact(n) / gamma(n+l+1.5d0)) * (-1)**n *
  & exp(-eta**2 / 2.d0) * laguerre(n,l+0.5d0,eta**2) * eta**(l+1)
ENDIF

RETURN
END

*****

REAL*8 FUNCTION c(n)

* Funkcja oblicza wspolczynniki rozwinięcia rozwiązania typu kosinus

INTEGER n, l
REAL*8 lambda, k
REAL*8 fact, gamma, hyperg21, hyperg11, fact1
REAL*8 pi, sint, cost, sin2t, eta
CHARACTER*8 basis

COMMON /lambda/lambda /k/k /basis/basis //l

PARAMETER (pi=3.1415926535897932384626434d0)

* Zmienne pomocnicze
sint = k / ( k*k/lambda + lambda/4.d0)
cost = (k**2 * lambda**(-2) - 0.25d0)/(k**2 * lambda**(-2) + 0.25d0)
sin2t = lambda**2 / (4.d0 * k**2 + lambda**2)
eta = k / lambda

IF (basis .EQ. 'laguerre') THEN
  c = (-2)**l * fact1(n,n+2*l+1) / sint**l
  & * hyperg21(-n-2.0d0*-1.d0, n+1.d0, 0.5d0-l, sin2t)
  & / sqrt(pi) * gamma(l+0.5d0)
ELSE
  c = sqrt(2.0/pi) * gamma(l+0.5d0) * (-1)**n * (fact(n) /
  & gamma(n+l+1.5d0)) * exp(-eta**2/2.0) * eta**(-l) *
  & hyperg11(-n-l-0.5d0, 0.5d0-l, eta**2)
ENDIF

RETURN
END

*****

REAL*8 FUNCTION fifi(m,n)

REAL*8 lambda
REAL*8 gamma, k_delta, fact, fact1
INTEGER m, n, l
CHARACTER*8 basis

COMMON /lambda/lambda //l /basis/basis

IF (basis .EQ. 'laguerre') THEN
  fifi = fact1(n+2*l+1,n) / lambda
  & * ( 2.d0*(n+l+1)*k_delta(m,n) - n*k_delta(m,n-1)
  & - (n+2*l+2)*k_delta(m,n+1) )
ELSE
  fifi = gamma(n+l+1.5d0) / 2.d0 / fact(n) * k_delta(m,n)
ENDIF

RETURN
END

*****

REAL*8 FUNCTION psipsi(m,n)

REAL*8 lambda
REAL*8 gamma, k_delta, fact, fact1
INTEGER m, n, l
CHARACTER*8 basis

COMMON /lambda/lambda //l /basis/basis

IF (basis .EQ. 'laguerre') THEN
  psipsi = fact1(n+2*l+1,n) / 4.d0
  & * ( 2.d0*(n+l+1)*(2*l*lambda-1)*k_delta(m,n) +
  & n*k_delta(m,n-1) + (n+2*l+2)*k_delta(m,n+1) )
ELSE
  psipsi = lambda**2 * gamma(n+l+1.5d0) / 2.d0 / fact(n) *
  & ( (2*n+l+1.5)*k_delta(m,n) + n*k_delta(m,n-1) +
  & (n+l+1.5)*k_delta(m,n+1) )
ENDIF

RETURN
END

*****

REAL*8 FUNCTION Jm(i,j)

* Funkcja wylicza elementy J-macierzy
* Funkcja wywołuje właściwe funkcje (obejście braku zmiennych
* dynamicznych w Fortranie 77)

INTEGER i, j, n, dim
REAL*8 Jm1
REAL*8 J1(50,50), J2(50,50)
REAL*8 J3(50,50), J4(50,50), Jmac(100,100)
CHARACTER*16 scheme

COMMON /scheme/scheme /N/n

IF (scheme .EQ. 'relativistic') THEN
  dim = 2 * n
ELSE
  dim = n
ENDIF

Jm = Jm1(i, j, n, dim, J1, J2, J3, J4, Jmac)

RETURN
END

*****

REAL*8 FUNCTION Jm1(i, j, n, dim, J1, J2, J3, J4, Jmac)

INTEGER i, j, m, l
INTEGER dim
CHARACTER*16 scheme
REAL*8 k, v_light, E
REAL*8 psipsi, fifi
REAL*8 J1(n,n), J2(n,n)
REAL*8 J3(n,n), J4(n,n), Jmac(dim,dim)

COMMON /k/k /scheme/scheme /v_light/v_light /E/E

IF (scheme .EQ. 'relativistic') THEN
  DO m=1,n
  DO l=1,n

```

```

      J1(m,l) = -E          * fifi (n-m, n-l)
      J2(m,l) = v_light   * psipsi (n-m, l-1)
      J3(m,l) = -(E + 2 * v_light**2) * psipsi (m-1, l-1)
      ENDDO
      ENDDO

      CALL transpose(n,J2,J4)

      DO m=1,dim
      DO l=1,dim
      IF (m.LE.n .AND. l.LE.n) Jmac(m,l)=J1(m,l)
      IF (m.LE.n .AND. l.GT.n) Jmac(m,l)=J2(m,l-n)
      IF (m.GT.n .AND. l.GT.n) Jmac(m,l)=J3(m-n,l-n)
      IF (m.GT.n .AND. l.LE.n) Jmac(m,l)=J4(m-n,l)
      ENDDO
      ENDDO

      Jm1 = Jmac(i+1,j+1)

      ELSE

      Jm1 = 0.5d0 * psipsi(i,j) - k**2 / 2.d0 * fifi(i,j)

      ENDIF

      RETURN
      END
*****

      REAL*8 FUNCTION g(x,y)
* Funkcja pomocnicza, wywołująca właściwą procedurę g1

      CHARACTER*16 scheme
      INTEGER x, y, dim, n
      REAL*8 A(100,100), A_orig(100,100)
      REAL*8 Eigval(100)
      REAL*8 Gamma(100,100), Gamma_transp(100,100)
      REAL*8 ll(100,100)
      REAL*8 G_(100,100), A_diag(100,100)
      REAL*8 g1

      COMMON /scheme/scheme /N/n

      IF (scheme.EQ.'relativistic') THEN
      dim = 2*n
      ELSE
      dim = n
      ENDIF

      g =
      g1(dim,x,y,A,A_orig,Eigval,Gamma,Gamma_transp,ll,G_,A_diag)

      RETURN
      END
*****

      REAL*8 FUNCTION
      & g1(dim,x,y,A,A_orig,Eigval,Gamma,Gamma_transp,ll,G_,A_diag)
* Funkcja wylicza wartości elementów macierzy aproksymujące
* funkcje Greena

      CHARACTER*16 scheme
      INTEGER x, y, dim
      REAL*8 A(dim,dim), A_orig(dim,dim)
      REAL*8 Eigval(dim)
      REAL*8 Gamma(dim,dim), Gamma_transp(dim,dim)
      REAL*8 ll(dim,dim)
      REAL*8 G_(dim,dim), A_diag(dim,dim)
      REAL*8 Jm, vN
      REAL*8 sum

      COMMON /scheme/scheme

      5 FORMAT (8F15.10)

* Utworzenie macierzy wejsciowej (korzystając z jej symetryczności)
      DO i=1,dim
      DO j=1,i
      A_orig(i,j) = Jm(i-1,j-1) + vN(i-1,j-1)
      A_orig(j,i) = A_orig(i,j)
      ENDDO
      ENDDO

      ENDDO
      ENDDO

* Przepisanie macierzy A_orig do macierzy A
      DO i=1,dim
      DO j=1,dim
      A(i,j) = A_orig(i,j)
      ENDDO
      ENDDO

* Policzenie wartości własnych i wektorów własnych macierzy A
* Z wektorów własnych utworzona zostanie macierz ortogonalna
Gamma,
* użyta następnie do diagonalizacji macierzy A
CALL eigensys(dim,A,Eigval,Gamma)

* Transponowanie macierzy Gamma
CALL transpose(dim,Gamma,Gamma_transp)

* Przepisujemy macierz A_orig do macierzy A
      DO i=1,dim
      DO j=1,dim
      A(i,j) = A_orig(i,j)
      ENDDO
      ENDDO

* Diagonalizacja macierzy A
CALL multiply(dim,A,Gamma,ll)
CALL multiply(dim,Gamma_transp,ll,A_diag)

* Policzenie elementów macierzy G_ = A_inv
      DO k=1,dim
      DO i=1,dim
      sum = 0.d0
      DO j=1,dim
      sum = sum + Gamma(k,j)*Gamma(i,j)/Eigval(j)
      ENDDO
      G_(k,i) = sum
      ENDDO
      ENDDO

      IF (scheme.EQ.'relativistic') THEN
      g1 = G_(1,1)
      ELSE
      g1 = G_(x+1,y+1)
      ENDIF

      RETURN
      END
*****

      REAL*8 FUNCTION v(r)
* Potencjal rozpraszający

      REAL*8 r
      REAL*8 v0, a, b
      CHARACTER*8 type

      COMMON /potential_type/type /well/v0,a,b

      IF (type.EQ.'well') THEN
      Prostokątna studnia potencjału
      IF ( r.GE.a .AND. r.LE.b ) THEN
      v = v0
      ELSE
      v = 0.d0
      ENDIF
      ELSE
* Inny potencjal (przyk³ad)
      v = -1.d0 / r**2
      ENDIF

      RETURN
      END
*****

      REAL*8 FUNCTION vN(i,j)
* Procedura obcina zadany potencjal w wybranej bazie

      INTEGER n_trunc, i, j

```

```

INTEGER i_, j_
REAL*8 integral_inf, integral
REAL*8 v0, a, b
CHARACTER*16 scheme
CHARACTER*8 type

COMMON /mn/i_ j_ /N/n_trunc /scheme/scheme
COMMON /potential_type/type /well/v0,a,b

EXTERNAL fi_v_fi, psi_v_psi

IF (scheme .EQ. 'relativistic') THEN

  IF ( (i .LT. n_trunc) .AND. (j .LT. n_trunc) ) THEN

*   Obszar "++"
    i_ = n_trunc-i-1
    j_ = n_trunc-j-1

*   Jeżeli studnia potencjału, to kalkujemy tylko w jej obszarze
    IF (type .EQ. 'well') THEN
      vN = integral(fi_v_fi, a, b)
    ELSE
      vN = integral_inf(fi_v_fi, 0.d0, 1)
    ENDIF

    ELSE IF ((i .GE. n_trunc) .AND. (j .GE. n_trunc) ) THEN

*   Obszar "--"
    i_ = i-n_trunc
    j_ = j-n_trunc

    IF (type .EQ. 'well') THEN
      vN = integral(psi_v_psi, a, b)
    ELSE
      vN = integral_inf(psi_v_psi,0.d0,1)
    ENDIF

    ELSE

*   Obszar "+-" i "-+"
    vN = 0.d0

    ENDIF

  ELSE

*   Nierelatywistycznie
    i_ = i
    j_ = j

    IF ( (i .LT. n_trunc) .AND. (j .LT. n_trunc) ) THEN

*   Jeżeli studnia potencjału, to kalkujemy tylko w jej obszarze
    IF (type .EQ. 'well') THEN
      vN = integral(fi_v_fi, a, b)
    ELSE
      vN = integral_inf(fi_v_fi, 0.d0, 1)
    ENDIF

    ELSE

      vN = 0.d0

    ENDIF

  ENDIF

RETURN
END

*****

REAL*8 FUNCTION fi_v_fi(r)

*   Funkcja do wycalkowania przez funkcje vN

REAL*8 r
REAL*8 v, fi
INTEGER m, n

COMMON /mn/m,n

fi_v_fi = fi(m,r) * v(r) * fi(n,r)

RETURN
END

*****

REAL*8 FUNCTION psi_v_psi(r)

*   Funkcja do wycalkowania przez funkcje vN

REAL*8 r
REAL*8 v, psi
INTEGER m, n

COMMON /mn/m,n

psi_v_psi = psi(m,r) * v(r) * psi(n,r)

RETURN
END

*****

SUBROUTINE LoadParams(fin)

*   Procedura wczytuje z pliku dane dla programu

CHARACTER*20 fin
CHARACTER*39 dummy
CHARACTER*8 basis, vol, type
CHARACTER*16 scheme
CHARACTER*8 fout
REAL*8 E, k, kpr, lambda
REAL*8 v0, a, b
REAL*8 Eps, Epspr, v_light
INTEGER l, kappa, Nstart, Nend

COMMON /basis/basis /lambda/lambda /k/k /kpr/kpr /l/l
/kappa/kappa
COMMON /scheme/scheme /v_light/v_light /E/E /Eps/Eps
/Epspr/Epspr
COMMON /steer/fout,Nstart,Nend /potential_type/type /well/v0,a,b

OPEN(4,FILE=fin)

READ(4,'(A39)') dummy
READ(4,'(A39)') dummy
READ(4,'(A39,D30.20)') dummy, E
READ(4,'(A39,I2)') dummy, l
READ(4,'(A39,I2)') dummy, kappa
READ(4,'(A39,D30.20)') dummy, lambda
READ(4,'(A39,A8)') dummy, basis
READ(4,'(A39,A15)') dummy, scheme
READ(4,'(A39,A8)') dummy, vol
READ(4,'(A39,A8)') dummy, type
READ(4,'(A39)') dummy
READ(4,'(A39)') dummy
READ(4,'(A39)') dummy
READ(4,'(A39,A8)') dummy, fout
READ(4,'(A39,I3)') dummy, Nstart
READ(4,'(A39,I3)') dummy, Nend
READ(4,'(A39)') dummy
READ(4,'(A39)') dummy
READ(4,'(A39)') dummy
READ(4,'(A39,D30.20)') dummy, v0
READ(4,'(A39,D30.20)') dummy, a
READ(4,'(A39,D30.20)') dummy, b

CLOSE(4)

IF (v0 .GE. E)
& CALL err('setparams: v0 >= E')

IF (Nstart.LT.0 .OR. Nstart.GT.Nend)
& CALL err('setparams: Zle parametry sterujace')

IF (kappa.NE.l .AND. kappa.NE.-l-1 .AND. scheme.EQ.'relativistic')
& CALL err('setparams: Nieskorelowane wartosci liczb kwantowych')

IF (vol .EQ. 'finite') THEN
  v_light = 137.036
ELSE
  v_light = 1.0D6
ENDIF

```

```

ENDIF
IF (scheme .EQ. 'relativistic') THEN
  k = sqrt( E * (E + 2 * v_light**2) ) / v_light
  kpr = sqrt( (E-v0) * ((E-v0) + 2 * v_light**2) ) / v_light
ELSE
  k = sqrt( 2 * E )
  kpr = sqrt( 2 * (E-v0) )
ENDIF

Eps = sqrt( E / (E + 2 * v_light**2) )
Epspr = sqrt( (E-v0) / ((E-v0) + 2 * v_light**2) )

RETURN
END

*****

SUBROUTINE SaveParams(fin)
* Procedura przepisuje do pliku wynikowego uzyte parametry i dane

CHARACTER*60 dummy
CHARACTER*20 fin

WRITE(1,(A17)) '# J-MATRIX METHOD'
WRITE(1,(A17)) '# -----'
WRITE(1,(A1)) '#
OPEN (3,FILE=fin)
DO i=1,22
  READ (3,(A60)) dummy
  WRITE(1,(A2,A60)) '#',dummy
ENDDO
WRITE(1,(A1)) '#
WRITE(1,(A1)) '#
CLOSE(3)

RETURN
END

*****

REAL*8 FUNCTION gegen(n,l,x)
* Funkcja wylicza wielomiany Gegenbauera C=C(cost) dla argumentu
* z przedzialu -1 < x < 1

REAL*8 l, x
REAL*8 fact, fact1, gamma
INTEGER n, m

IF (abs(x) .GT. 1.d0)
& CALL err ('gegen: Argument spoza zakresu ')

gegen = 0.d0

IF (l .EQ. int(l) ) THEN

  DO m=0,n
    gegen = gegen + fact1(int(l+m-1),m) * cos((n-2*m)*acos(x))
& * fact1(int(l+n-m-1),n-m) / gamma(l)**2
  ENDDO

ELSE

  DO m=0,n
    gegen = gegen + gamma(l+m) / fact(m) * cos((n-2*m)*acos(x))
& * gamma(l+n-m) / fact(n-m) / gamma(l)**2
  ENDDO

ENDIF

RETURN
END

*****

SUBROUTINE err(nerr)
* Procedura wyswietla nazwe procedury w ktorej wystapil blad,
* rodzaj bledu oraz wartosci istotnych zmiennych globalnych

CHARACTER*52 nerr
CHARACTER*8 basis, type

CHARACTER*16 scheme
REAL*8 E, k, kpr, lambda, v0, a, b, Eps, Epspr, v_light
INTEGER l, kappa

COMMON /basis/basis /lambda/lambda /k/k /kpr/kpr /l/
/kappa/kappa
COMMON /scheme/scheme /v_light/v_light /E/E /Eps/Eps
/Epspr/Epspr
COMMON /potential_type/type /well/v0,a,b

WRITE(*,*) 'Blad w procedurze ',nerr
WRITE(*,*)
WRITE(*,*) 'Wartosci zmiennych globalnych:'
WRITE(*,*) 'basis = ',basis
WRITE(*,*) 'lambda = ',lambda
WRITE(*,*) 'k = ',k
WRITE(*,*) 'kpr = ',kpr
WRITE(*,*) 'l = ',l
WRITE(*,*) 'kappa = ',kappa
WRITE(*,*) 'scheme = ',scheme
WRITE(*,*) 'v_light = ',v_light
WRITE(*,*) 'E = ',E
WRITE(*,*) 'Eps = ',Eps
WRITE(*,*) 'Epspr = ',Epspr
WRITE(*,*) 'type = ',type
WRITE(*,*) 'v0 = ',v0
WRITE(*,*) 'a = ',a
WRITE(*,*) 'b = ',b

STOP
END

*****

REAL*8 FUNCTION fact(n)
* Funkcja wylicza na podstawie definicji silnie zadanej liczby
* naturalnej n
* Funkcja jest typu rzeczywistego, aby powiekszyc zakres jej
dzialania
* do duzych liczb

INTEGER n, i

IF (n.LT.0)
& CALL err('fact: Ujemny argument funkcji. ')
IF (n.GT.170)
& CALL err('fact: Zbyt duza wartosc argumentu. ')

fact = 1.d0
DO i=1,n
  fact = fact * i
ENDDO

RETURN
END

*****

REAL*8 FUNCTION fact1(m,n)
* Funkcja wylicza na podstawie definicji iloraz silni zadanych
* liczb naturalnych m i n

INTEGER m, n, i

IF (n.LT.0 .OR. m.LT.0)
& CALL err('fact1: Ujemny argument funkcji. ')
IF (abs(n-m).GT.170)
& CALL err('fact1: Zbyt duza wartosc argumentu. ')

IF (m.GT.n) THEN
  fact1 = 1.d0
  DO i=n+1,m
    fact1 = fact1 * i
  ENDDO
ELSE
  fact1 = 1.d0
  DO i=m+1,n
    fact1 = fact1 * i
  ENDDO
  fact1 = 1.d0 / fact1
ENDIF

```

```

RETURN
END
*****

INCLUDE 'lib/dgamma.inc'
*****

REAL*8 FUNCTION gamma(x)
* Funkcja oblicza wartosc funkcji gamma przy wykorzystaniu
* procedury bibliotecznej

REAL*8 x
REAL*8 dgamma

gamma = dgamma(x)

IF (gamma .EQ. 1.D308)
& CALL err('gamma: Niepoprawny argument      ')

RETURN
END
*****

INCLUDE 'lib/rjbesl.inc'
*****

REAL*8 FUNCTION bessel(v,z)
* Funkcja wylicza wartosci funkcji Bessela pierwszego rodzaju

REAL*8 v, z, res(100), v_frac
REAL*8 hyperg01, gamma
INTEGER nerr, v_int

v_int = INT(v)
v_frac = v - v_int

IF (v .LT. 0.) THEN

IF (v .EQ. int(v)) THEN

* Calkowite ujemne wartosci v - biblioteka i wzor redukcyjny
CALL rjbesl(z, -v_frac, -v_int+1, res, nerr)
bessel = (-1)**v * res(-v_int+1)

ELSE

* Wzor bezposredni (moze zawodzic dla duzych z i v)
bessel =
& (z/2.d0)**v / gamma(v+1.d0) * hyperg01(v+1.d0,-0.25d0*z**2)

ENDIF

ELSE

* Dodatnie wartosci v
CALL rjbesl(z, v_frac, v_int+1, res, nerr)
bessel = res(v_int+1)

ENDIF

IF (nerr.LT.0)
& CALL err('bessel: Nieznany blad w funkcji bibliotecznej      ')

RETURN
END
*****

REAL*8 FUNCTION bess_prim(v,z)
* Funkcja wylicza wartosc pochodnej funkcji Bessela

REAL*8 v, z
REAL*8 bessel

bess_prim = 0.5d0 * ( bessel(v-1,z) - bessel(v+1,z) )

RETURN
END

*****

REAL*8 FUNCTION sphbess(v,z)
* Funkcja wylicza wartosc sferycznej funkcji Bessela pierwszego
* rodzaju

REAL*8 v, z
REAL*8 pi
REAL*8 bessel

PARAMETER (pi=3.1415926535897932384626434d0)

IF (z .EQ. 0.d0)
& CALL err('sphbess: Zerowy argument      ')

sphbess = sqrt(pi/2.d0/z) * bessel(v+0.5d0,z)

RETURN
END
*****

REAL*8 FUNCTION sphbess_prim(v,z)
* Funkcja wylicza wartosc pochodnej sferycznej funkcji Bessela

REAL*8 v, z
REAL*8 pi
REAL*8 bessel

PARAMETER (pi=3.1415926535897932384626434d0)

sphbess_prim = sqrt(pi/8.d0) * z**(-1.5d0) * (
& z*bessel(-0.5+v,z)-bessel(0.5+v,z)-z*bessel(1.5+v,z) )

END
*****

INCLUDE 'lib/rybesl.inc'
*****

REAL*8 FUNCTION neumann(v,z)
* Funkcja wylicza wartosc funkcji Neumana( funkcji Bessela
* drugiego rodzaju)

REAL*8 v, z, res(100)
REAL*8 pi, v_frac, bessel
INTEGER v_int, nerr

PARAMETER (pi=3.1415926535897932384626434d0)

v_int = INT(v)
v_frac = v - v_int

IF (v .LT. 0.) THEN

IF (v .EQ. int(v)) THEN

* Calkowite ujemne wartosci v - biblioteka i wzor redukcyjny
CALL rybesl(z, -v_frac, -v_int+1, res, nerr)
neumann = (-1)**v * res(-v_int+1)

ELSE

* Wzor bezposredni
neumann = 1.d0 / sin(v*pi) * (
& bessel(v,z)*cos(v*pi)-bessel(-v,z) )

ENDIF

ELSE

* Dodatnie wartosci v
CALL rybesl(z, v_frac, v_int+1, res, nerr)
neumann = res(v_int+1)

ENDIF

END

```

```

IF (nerr.LT.0)
& CALL err('neumann: Nieznany blad w funkcji bibliotecznej ')
RETURN
END
*****

REAL*8 FUNCTION neum_prim(v,z)
* Funkcja wylicza wartosc pochodnej funkcji Neumanna
REAL*8 v, z
REAL*8 neumann

neum_prim = 0.5d0 * ( neumann(v-1,z) - neumann(v+1,z) )

RETURN
END
*****

REAL*8 FUNCTION sphneum(v,z)
* Funkcja wylicza wartosc sferycznej funkcji Neumanna
REAL*8 v, z
REAL*8 pi
REAL*8 neumann

PARAMETER (pi=3.1415926535897932384626434d0)

IF (z .EQ. 0.d0)
& CALL err('sphneum: Zerowy argument ')

sphneum = sqrt(pi/2.d0/z) * neumann(v+0.5d0,z)

RETURN
END
*****

REAL*8 FUNCTION sphneum_prim(v,z)
* Funkcja wylicza wartosc pochodnej sferycznej funkcji Neumanna
REAL*8 v, z
REAL*8 pi
REAL*8 neumann

PARAMETER (pi=3.1415926535897932384626434d0)

sphneum_prim = sqrt(pi/8.d0) * z**(-1.5d0) * (
& z*neumann(-0.5+v,z)-neumann(0.5+v,z)-z*neumann(1.5+v,z) )

END
*****

REAL*8 FUNCTION hyperg11(a,c,z)
* Funkcja oblicza wartosc funkcji hipergeometrycznej 1F1(a,c;z)
REAL*8 a, c, z
REAL*8 s, d, y

s = 1.d0
y = 1.d0
DO n=1,200
d = n * ( (c+n-1)**2 )
y = (a+n-1) * (y/d)
y = y * (c+n-1)
y = y * z
IF (s .EQ. s+y) GOTO 10
s = s + y
ENDDO

10 hyperg11 = s

RETURN
END
*****

REAL*8 FUNCTION hyperg21(a,b,c,z)
* Funkcja oblicza wartosc funkcji hipergeometrycznej 2F1(a,b,c;z)
REAL*8 a, b, c, z
REAL*8 s, y

s = 1.d0
y = 1.d0
DO i=0,200
y=y*(a+i)/(c+i)*(b+i)*z/(i+1)
IF (s.EQ.s+y) GOTO 10
s=s+y
ENDDO

10 hyperg21 = s

RETURN
END
*****

REAL*8 FUNCTION hyperg01(c,z)
* Funkcja oblicza wartosc funkcji hipergeometrycznej 0F1(a;z)
REAL*8 c, z
REAL*8 s, d, y

s = 1.d0
y = 1.d0
DO n=1,200
d = n * ( (c+n-1)**2 )
y = y/d
y = y * (c+n-1)
y = y * z
IF (s .EQ. s+y) GOTO 10
s = s + y
ENDDO

10 hyperg01 = s

RETURN
END
*****

INCLUDE 'lib/dqag.inc'
*****

REAL*8 FUNCTION integral(fun,a,b)
* Funkcja oblicza calke oznaczona z funkcji <fun> w granicach [a,b]
* 30-61 punktowa metoda Gaussa-Kronoda
REAL*8 abserr, a, b
REAL*8 ans, epsrel, epsabs
REAL*8 work(20000)
INTEGER neval, ierr, last, limit, lenw, key
INTEGER iwork(5000)

EXTERNAL fun

* Przygotowanie zmiennych wymaganych przez procedure
limit = 5000
lenw = limit*4
epsabs = 1.d-18
epsrel = 1.d-18
key = 6

* Wywołanie procedury kalkujacej
CALL dqag(fun, a, b, epsabs, epsrel, key, ans, abserr,
& neval, ierr, limit, lenw, last, iwork, work)

IF (ierr .NE. 0)
& CALL err('integral: Nieznany blad w procedurze dqag ')

integral = ans

RETURN
END
*****

```

```

REAL*8 FUNCTION integral_inf(fun,a,typ)
* Funkcja oblicza calke niewlasciwa z funkcji <fun> w granicach
* [-Inf,Inf], [a,Inf] lub [-Inf,a] w zalezności od argumentu <typ>

REAL*8 abserr, a
REAL*8 ans, epsabs, epsrel
REAL*8 work(20000)
INTEGER typ, neval, ierr, last, limit, lenw
INTEGER iwork(5000)

EXTERNAL fun

* Przygotowanie zmiennych wymaganych przez procedure
limit = 5000
lenw = limit*4
epsabs = 1.d-18
epsrel = 1.d-18

* Wywołanie procedury kalkujacej
CALL dqagi(fun, a, typ, epsabs, epsrel, ans, abserr,
& neval, ierr, limit, lenw, last, iwork, work)

IF ( ierr .NE. 0)
& CALL err('integral_inf: Nieznany blad w procedurze dqagi ')

integral_inf = ans

RETURN
END

*****

REAL*8 FUNCTION k_delta(a,b)
* Funkcja zwraca wartosc funkcji delta Kroneckera dla argumentow
a,b

INTEGER a, b

IF ( a .EQ. b ) THEN
k_delta = 1.d0
ELSE
k_delta = 0.d0
ENDIF

RETURN
END

*****

REAL*8 FUNCTION laguerre(n,a,x)
* Funkcja wylicza wartosci stowarzyszonych wielomianow Laguerre'a

INTEGER n
REAL*8 a, x, n_
REAL*8 gamma, fact, hyperg11, fact1

n_ = dble(n)

IF ( a .EQ. int(a) ) THEN
laguerre = fact1(int(n+a),n)/gamma(a+1)*hyperg11(-n_,a+1,x)
ELSE
laguerre = gamma(n+a+1)/fact(n)/gamma(a+1)*hyperg11(-
n_,a+1,x)
ENDIF

RETURN
END

*****

INCLUDE 'lib/rsm.inc'
*****

SUBROUTINE multiply(n,A,B,M)
* Procedura mnozy dwie macierze kwadratowe A i B o wymiarach nxn
* Wynik => macierz M=A*B (nxn)

INTEGER n, i, j, k
REAL*8 sum
REAL*8 A(n,n), B(n,n), M(n,n)

DO i=1,n
DO k=1,n
sum = 0.d0
DO j=1,n-1
sum = ( A(i,j) * B(j,k) ) + sum
ENDDO
sum = ( A(i,n) * B(n,k) ) + sum
M(i,k) = sum
ENDDO
ENDDO

RETURN
END

*****

SUBROUTINE transpose(n,A,T)
* Procedura transponuje macierz kwadratowa A o wymiarze nxn
* Wynik => macierz T (nxn)

INTEGER n, i, j
REAL*8 A(n,n), T(n,n)

DO i=1,n
DO j=1,n
T(i,j) = A(j,i)
ENDDO
ENDDO

END

*****

SUBROUTINE eigensys(n,A,val,vec)
* Procedura znajduje wektory wlasne i wartosci wlasne rzeczywistej
* i symetrycznej macierzy A o wymiarze nxn metoda ql i odwrotnej
iteracji

REAL*8 A(n,n), val(n), vec(n,n)
REAL*8 fwork(2400)
INTEGER nerr, n, i, j
INTEGER iwork(300)

* Test symetryczności macierzy
DO i=1,n
DO j=1,i
IF (A(i,j) .NE. A(j,i))
& CALL err('eigensys: Niesymetryczna macierz ')
ENDDO
ENDDO

CALL rsm(n, n, A, val, n, vec, fwork, iwork, nerr)

IF (nerr .NE. 0)
& CALL err ('eigensys: Nieokreslony blad w procedurze rsm ')

RETURN
END

```

## D.2 Mathematica 3.0

(\* Metoda J-macierzy - wersja relatywistyczna \*)

(\* ----- \*)

(\* Dane wejsciowe \*)

basis := "laguerre" (\* wybór bazy \*)  
 l := 0 (\* orbitalna liczba kwantowa \*)  
 κ := -1 (\* liczba kwantowa \*)  
 λ := 1.0 (\* parametr skalujacy \*)  
 v<sub>light</sub> := 137.036 (\* predkosc swiatla \*)  
 ε := 2.5 (\* energia kinetyczna elektronu \*)

$$\epsilon := \sqrt{\frac{\epsilon}{\epsilon + 2 v_{\text{light}}^2}}$$

$$k := \frac{\sqrt{\epsilon (\epsilon + 2 v_{\text{light}}^2)}}{v_{\text{light}}} \quad (* \text{liczba falowa} *)$$

N<sub>trunc</sub> := 3 (\* parametr obciecia potencjalu \*)  
 V<sub>0</sub> := -1.5 (\* glebokosc studni potencjalu \*)  
 a := 1.0 (\* lewa krawedz studni potencjalu \*)  
 b := 1.5 (\* prawa krawedz studni potencjalu \*)

(\* Funkcje bazowe \*)

φ[n\_, r\_] := If[basis == "laguerre",

$$(\lambda r)^{l+1} \text{Exp}\left[-\frac{\lambda r}{2}\right] \text{LaguerreL}[n, 2l+1, \lambda r],$$

$$(\lambda r)^{l+1} \text{Exp}\left[-\frac{\lambda^2 r^2}{2}\right] \text{LaguerreL}\left[n, l + \frac{1}{2}, \lambda^2 r^2\right]$$

ψ[n\_, r\_] := If[basis == "laguerre",

$$\frac{\kappa}{r} \phi[n, r] +$$

$$\text{Exp}\left[-\frac{\lambda r}{2}\right] \left( \lambda^{l+1} (l+1) r^l \text{LaguerreL}[n, 2l+1, \lambda r] - (\lambda r)^{l+1} \frac{\lambda}{2} \text{LaguerreL}[n, 2l+1, \lambda r] - \right. \\ \left. (\lambda r)^{l+1} (\text{LaguerreL}[n, 2l+2, \lambda r] - \text{LaguerreL}[n, 2l+1, \lambda r]) \right),$$

$$\frac{\kappa}{r} \phi[n, r] +$$

$$\text{Exp}\left[-\frac{\lambda^2 r^2}{2}\right] \left( \lambda^{l+1} (l+1) r^l \text{LaguerreL}\left[n, l + \frac{1}{2}, \lambda^2 r^2\right] - (\lambda r)^{l+1} \lambda^2 r \text{LaguerreL}\left[n, l + \frac{1}{2}, \lambda^2 r^2\right] - \right. \\ \left. (\lambda r)^{l+1} \left( \text{LaguerreL}\left[n, l + \frac{3}{2}, \lambda^2 r^2\right] - \text{LaguerreL}\left[n, l + \frac{1}{2}, \lambda^2 r^2\right] \right) \right)$$

(\* Współczynniki rozwinięć rozwiązani typu sinus i kosinus \*)

$$\text{sint} := \frac{\frac{\kappa}{\lambda}}{\frac{\kappa^2}{\lambda^2} + \frac{1}{4}}; \quad \text{cost} := \frac{\frac{\kappa^2}{\lambda^2} - \frac{1}{4}}{\frac{\kappa^2}{\lambda^2} + \frac{1}{4}}; \quad \text{sin}2t2 := \frac{\lambda^2}{4\kappa^2 + \lambda^2}; \quad \eta := \frac{\kappa}{\lambda}$$



```

s[n_] := If[basis == "laguerre",
  
$$\frac{2^l \Gamma[1+l] n! \operatorname{sint}^{l+1}}{\Gamma[n+2l+2]} \operatorname{GegenbauerC}[n, l+1, \operatorname{cost}],$$

  
$$\frac{\sqrt{2\pi} n! (-1)^n \operatorname{Exp}\left[-\frac{\eta^2}{2}\right] \eta^{l+1} \operatorname{LaguerreL}\left[n, l+\frac{1}{2}, \eta^2\right]}{\Gamma\left[n+l+\frac{3}{2}\right]}]$$

```

```

c[n_] := If[basis == "laguerre",
  
$$\frac{-2^l \Gamma\left[1+\frac{1}{2}\right] n!}{\sqrt{\pi} \Gamma[n+2l+2] \operatorname{sint}^l} \operatorname{Hypergeometric2F1}\left[-n-2l-1, n+1, \frac{1}{2}-l, \operatorname{sin}2t2\right],$$

  
$$\frac{\sqrt{\frac{2}{\pi}} \Gamma\left[1+\frac{1}{2}\right] (-1)^n n!}{\Gamma\left[n+l+\frac{3}{2}\right]} \operatorname{Exp}\left[-\frac{\eta^2}{2}\right] \eta^{-1} \operatorname{Hypergeometric1F1}\left[-n-l-\frac{1}{2}, \frac{1}{2}-l, \eta^2\right]]]$$

```

(\* Elementy J-macierzy \*)

(\* Funkcja delta Kroneckera \*)

```
δ[a_, b_] := If[a == b, 1.0, 0.0]
```

```
fifi[n_, m_] := If[basis == "laguerre",
```

$$\frac{\Gamma[n+2l+2]}{\lambda n!} (2(n+1+l) \delta[m, n] - n \delta[m, n-1] - (n+2l+2) \delta[m, n+1]),$$

$$\frac{\Gamma\left[n+l+\frac{3}{2}\right]}{2 n!} \delta[m, n]$$

```
psipsi[n_, m_] := If[basis == "laguerre",
```

$$\frac{\Gamma[n+2l+2]}{4 n!} (2(n+1+l) (2\lambda-1) \delta[m, n] + n \delta[m, n-1] + (n+2l+2) \delta[m, n+1]),$$

$$\frac{\lambda^2 \Gamma\left[n+l+\frac{3}{2}\right]}{2 n!} \left( \left(2n+l+\frac{3}{2}\right) \delta[m, n] + n \delta[m, n-1] + \left(n+l+\frac{3}{2}\right) \delta[m, n+1] \right)$$

(\* Nierelatywistyczna J-macierz \*)

```
Jm[n_, m_] :=  $\frac{1}{2}$  psipsi[n, m] -  $\frac{k^2}{2}$  fifi[n, m]
```

```
Jm1[n_, m_] := -ε fifi[Ntrunc - n - 1, Ntrunc - m - 1]
```

```
Jm2[n_, m_] := vlight psipsi[Ntrunc - n - 1, m]
```

```
Jm3[n_, m_] := -(ε + 2 vlight2) psipsi[n, m]
```

```
J1 = Array[Jm1, {Ntrunc, Ntrunc}, 0];
```

```
J2 = Array[Jm2, {Ntrunc, Ntrunc}, 0];
```

```
J3 = Array[Jm3, {Ntrunc, Ntrunc}, 0];
```

```
J4 = Transpose[Array[Jm2, {Ntrunc, Ntrunc}, 0]];
```

```
<< LinearAlgebra`MatrixManipulation`
```

```
Jup = AppendRows[J1, J2];
```

```
Jdown = AppendRows[J4, J3];
```

(\* Relatywistyczna J-macierz \*)

```
J = AppendColumns[Jup, Jdown]; MatrixForm[J]
```

-45.	15.	0.	0.	205.554	616.662
15.	-20.	5.	68.518	274.072	205.554
0.	5.	-5.	68.518	68.518	0.
0.	68.518	68.518	-18780.1	-18780.1	0.
205.554	274.072	68.518	-18780.1	-75120.5	-56340.3
616.662	205.554	0.	0.	-56340.3	-169021.

(\* Potencjal \*)

(\* Potencjal rozpraszajacy \*)

$V[r_] := \text{If}[r > b \ \&\& \ r < a, 0.0, V_0]$

(\* Potencjal obciety \*)

$V_{\text{trunc}}[n_, m_] := \text{If}[n >= N_{\text{trunc}} \ || \ m >= N_{\text{trunc}},$

$\text{If}[n >= N_{\text{trunc}} \ \&\& \ m >= N_{\text{trunc}}, \text{Integrate}[\psi[n - N_{\text{trunc}}, r] V[r] \psi[m - N_{\text{trunc}}, r], \{r, a, b\}], 0],$

$\text{Integrate}[\phi[N_{\text{trunc}} - n - 1, r] V[r] \phi[N_{\text{trunc}} - m - 1, r], \{r, a, b\}]]$

$V_N = \text{Array}[V_{\text{trunc}}, \{2 N_{\text{trunc}}, 2 N_{\text{trunc}}\}, 0]; \text{MatrixForm}[V_N]$

-0.0209252	-0.016689	-0.00648953	0	0	0
-0.016689	-0.187646	-0.245222	0	0	0
-0.00648953	-0.245222	-0.332555	0	0	0
0	0	0	-0.0831388	-0.227583	-0.290511
0	0	0	-0.227583	-0.624689	-0.801628
0	0	0	-0.290511	-0.801628	-1.03905

$A = J + V_N; \text{MatrixForm}[A]$

-45.0209	14.9833	-0.00648953	0.	205.554	616.662
14.9833	-20.1876	4.75478	68.518	274.072	205.554
-0.00648953	4.75478	-5.33256	68.518	68.518	0.
0.	68.518	68.518	-18780.2	-18780.3	-0.290511
205.554	274.072	68.518	-18780.3	-75121.1	-56341.1
616.662	205.554	0.	-0.290511	-56341.1	-169022.

(\* Diagonalizacja macierzy A \*)

$r = \text{Eigenvectors}[A];$

$r^{\dagger} = \text{Transpose}[r];$

$A_{\text{diag}} = r \cdot A \cdot r^{\dagger};$

$g_{\text{m}}[n_, m_] := \text{Sum}[(r^{\dagger}[[n, j]] r^{\dagger}[[m, j]]) / A_{\text{diag}}[[j, j]], \{j, 1, 2 * N_{\text{trunc}}\}]$

$g = \text{Array}[g_{\text{m}}, \{2 N_{\text{trunc}}, 2 N_{\text{trunc}}\}];$

(\* Obliczenie przesunięcia fazowego \*)

$g_N = g[[1, 1]];$

$J_N = \text{Jm}[N_{\text{trunc}}, N_{\text{trunc}} - 1];$

$\text{tmp} = \frac{2 \epsilon v_{\text{light}}}{k};$

(\* Pierwszy wzór \*)

$\tan \delta_N = - \frac{s[N_{\text{trunc}} - 1] + \text{tmp} g_N J_N s[N_{\text{trunc}}]}{c[N_{\text{trunc}} - 1] + \text{tmp} g_N J_N c[N_{\text{trunc}}]};$

$$\delta_N = \text{ArcTan}[\tan\delta_N]$$

0.131831

$$g_N = g[1, 2 N_{\text{trunc}}];$$

$$tmp = 2 v_{\text{light}};$$

(\* Drugi wzór \*)

$$\tan\delta_N = -\frac{s[N_{\text{trunc}} - 1] + tmp g_N J_N s[N_{\text{trunc}}]}{c[N_{\text{trunc}} - 1] + tmp g_N J_N c[N_{\text{trunc}}]}$$

$$\delta_N = \text{ArcTan}[\tan\delta_N]$$

0.131852

(\* Obliczenia za pomocą wzoru analitycznego \*)

(\* Sferyczne funkcje Bessela i Neumanna \*)

$$j[v_, z_] := \sqrt{\frac{\pi}{2z}} \text{BesselJ}[v + \frac{1}{2}, z]$$

$$n[v_, z_] := \sqrt{\frac{\pi}{2z}} \text{BesselY}[v + \frac{1}{2}, z]$$

$$\epsilon_{\text{prim}} := \sqrt{\frac{\epsilon - V_0}{(\epsilon - V_0) + 2 v_{\text{light}}^2}}$$

$$k_{\text{prim}} := \frac{\sqrt{(\epsilon - V_0) ((\epsilon - V_0) + 2 v_{\text{light}}^2)}}{v_{\text{light}}}$$

$$X[r_, k_, \epsilon_] := \text{If}[x < 0, \{ \{j[1, kr], -n[1, kr]\}, \{-\epsilon j[1+1, kr], \epsilon n[1+1, kr]\}\}, \{ \{j[1, kr], -n[1, kr]\}, \{\epsilon j[1-1, kr], -\epsilon n[1-1, kr]\}\}]$$

$$Y[r_, k_, \epsilon_] := \text{If}[x < 0, \{ \{\epsilon n[1+1, kr], n[1, kr]\}, \{\epsilon j[1+1, kr], j[1, kr]\}\}, \{ \{-\epsilon n[1-1, kr], n[1, kr]\}, \{-\epsilon j[1-1, kr], j[1, kr]\}\}]$$

$$Y1 = Y[b, k, \epsilon];$$

$$X1 = X[b, k_{\text{prim}}, \epsilon_{\text{prim}}];$$

$$Y2 = Y[a, k_{\text{prim}}, \epsilon_{\text{prim}}];$$

$$X2 = X[a, k, \epsilon];$$

$$il = Y1 . X1 . Y2 . X2 . \{1, 0\};$$

(\* Obliczenie przesunięcia fazowego \*)

$$\tan\delta_{\text{acc}} = il[[2]] / il[[1]];$$

$$\delta_{\text{acc}} = \text{ArcTan}[\tan\delta_{\text{acc}}]$$

0.108927

## 8. Bibliografia

- [1] E. Heller, H. Yamani, *New  $L^2$  approach to quantum scattering: Theory*, Phys. Rev. A **9** (1974) 1201
- [2] E. Heller, H. Yamani, *J-matrix method: Application to s-wave electron-hydrogen scattering*, Phys. Rev. A **9** (1974) 1209
- [3] H. Yamani, L. Fishman, *J-matrix method: Extensions to arbitrary angular momentum and to Coulomb scattering*, J. Math. Phys. **16** (1975) 410
- [4] P. Horodecki, *The relativistic J-matrix method*, wysłana do druku
- [5] R. Szmytkowski, Nie opublikowane notatki
- [6] M. Krośnicki, *Relatywistyczna teoria rozpraszania potencjalnego*, praca dyplomowa, Politechnika Gdańska, 1998
- [7] H. Bateman, A. Erdély (ed.), *Higher Transcendental Functions*, vol. I, II, McGraw-Hill, New York, 1953
- [8] G. Arfken, *Mathematical Methods For Physicists*, Academic Press, New York, 1970