



Mathematica w nauczaniu fizyki jądrowej

Warszawa, czerwiec 2018

Przedmowa.nb | 2

Książeczka prezentuje wydruk cyklu wykładów wygłoszonych na Wydziale Fizyki UW w semestrze zimowym 2017/18 w ramach przedmiotu “Wykłady monograficzne z fizyki jądrowej”.

Pełna wersja wykładów wraz z dodatkami znajduje się na stronie internetowej autora pod adresem:

<http://www.fuw.edu.pl/~pfitzner/Teaching/Mathematica4Nuclear/Mathematica4Nuclear.html>

Na stronie tej znajdują się wszystkie prezentacje wykładów w formacie .nb, do otwarcia których wymagana jest pełna wersja pakietu *Mathematica*. Wykłady zostały sporządzone przy pomocy wersji *Mathematica 10.4*. Oprócz prezentacji typu SlideShow, dostępne są interaktywne programy w formacie CDF (Computable Data Format), które można otworzyć przy pomocy darmowego programu **CDF Player**.

Na stronie opublikowano też wybrane prace zaliczeniowe studentów.

Informacje wstępne

Założenia, adresaci wykładu, reguły.
Pierwszy kontakt z Mathematicą, źródła i pomoc.
Przykłady Manipulate i format CDF

Punkt wyjścia

- **Możliwości**

Pakiet *Mathematica*, to niezwykle mocne narzędzie obliczeniowe, które ma wszechstronne zastosowania. Pozwala na bardzo złożone obliczenia symboliczne, rachunki numeryczne z dowolnie dużą dokładnością, zaawansowaną analizę i prezentację obrazu, dźwięku, itp., itd. Możliwości pakietu ciągle rosną i wydają się nieskończone. Ogromna liczba użytkowników wymienia doświadczenia na forach internetowych, oferując przykłady, inspiracje i pomoc.

- **Zastosowania**

Pakiet *Mathematica* ma bardzo szeroki zakres zastosowań, od prostych rachunków, do zaawansowanego modelowania i złożonej analizy danych. Można go traktować jako rodzaj Ostatecznego (*Ultimate*) Kalkulatora Programowalnego.

- **Pomoc naukowa i dydaktyczna**

Możliwości pakietu sprawiają, że jest on doskonałym narzędziem wspomagającym pracę naukową i dydaktyczną. Wykład będzie obejmował oba aspekty, ale nacisk kładziemy na ten drugi.

Cel wykładu i jego adresaci

- Dla wszystkich

Poznanie programu *Mathematica* lub pogłębienie wiedzy o nim i o jego możliwościach. Dla początkujących, będzie to raczej intensywny *crash course*. Średnio zaawansowani słuchacze nauczą się być może czegoś nowego. Od słuchaczy zaawansowanych w programie *Mathematica* i/lub w programowaniu, jeśli wytrwają, autor oczekuje sugestii, poprawek i komentarzy.

- Dla młodszych adeptów fizyki jądrowej

Tematy wszystkich wykładów i rozważane przykłady będą zasadniczo dotyczyły fizyki jądra atomowego, opisu jego własności i struktury. Większość z nich powinna być omawiana na wykładach specjalistycznych z fizyki jądrowej. Tu proponujemy spojrzenie na te zagadnienia z innego punktu widzenia i próbę pogłębienia wiedzy o nich.

- Dla starszych, parających się dydaktyką

Wydaje się, że wykorzystanie pakietu *Mathematica* może ułatwić nauczanie fizyki. Szczególnie dotyczy to tych zagadnień, w których prezentuje się tylko wyniki złożonych obliczeń i modeli, które student musi przyjąć “na wiarę”. Możliwość sprawdzenia niektórych obliczeń, nawet znacznie uproszczonych do poziomu zabawki (*toy model*), pomaga zrozumieć istotę podejścia zaawansowanego i wspiera intuicję. Omawiane przykłady z fizyki jądrowej łatwo będzie przenieść na inne działy fizyki (np. r-nie Schrödingera).

W istocie podejmujemy eksperyment dydaktyczny!

Software

Pakiet *Mathematica* jest bardzo kosztowny, ale od niedawna Wydział Fizyki UW zakupił licencję “MATHEMATICA Unlimited Academic Site License”, która umożliwia korzystanie z tego programu pracownikom i studentom Wydziału. Licencja obejmuje też komputery domowe.

- Aby w pełni skorzystać z tego wykładu należy zainstalować pakiet *Mathematica* na swoim komputerze osobistym. Stosowne informacje oraz instrukcje są na stronie Wydziału Fizyki, patrz [tutaj](#).

Najnowsza wersja *Mathematica* ma numer 11. Do przygotowania wykładu i do jego prezentacji wykorzystałem wersję 10. Zalecam jednak instalację najnowszej wersji. W przypadku problemów z odtwarzaniem plików wykładowych proszę o informacje!

Zaliczenie

1. Osoby, które będą chciały zaliczyć ten przedmiot, proszę o wpisanie się na listę z podaniem adresu e-mail.
 2. Warunkiem wstępnym zaliczenia będzie wykonanie jednej (łatwej) pracy domowej. Szczegóły podamy w stosownym czasie.
 3. Końcowe zaliczenie na ocenę odbędzie się na podstawie projektu wykonanego samodzielnie przez studenta, w którym *Mathematica* zostanie wykorzystana do rozwiązania lub ilustracji jakiegoś zjawiska/problemu fizycznego. Tematy przykładowych projektów będą sugerowane w trakcie wykładów, ale inicjatywa studentów będzie mile widziana.
- Wyniki prac domowych oraz najciekawsze projekty zaliczeniowe wejdą w skład końcowych materiałów wykładowych, które będą dostępne w wersji cyfrowej w internecie i zostaną wydrukowane w postaci skryptu.

Strona internetowa

Materiały wykładowe będą na bieżąco umieszczane na stronie internetowej autora, czyli [tutaj](#).

Pełny tekst źródłowy wykładu, w formacie .nb, będzie publikowany po każdym wykładzie. Zaleca się ściągnięcie takiego pliku i samodzielne wykonanie wszystkich instrukcji oraz sprawdzenie wszystkich prezentacji interaktywnych.

W przypadku problemów proszę o informacje!

Literatura i pomoce naukowe

- Wszystko co ważne i potrzebne znajdziemy w internecie!

- Najważniejsze źródło to oczywiście 

A tam bardzo dobre wprowadzenie: [An Elementary Introduction](#), które można czytać on-line.

Polecam, szczególnie początkującym, przerobienie przynajmniej kilku pierwszych rozdziałów.

Alternatywne wprowadzenie dla bardziej zaawansowanych w programowaniu (też on-line): [Fast Introduction for Programmers](#).

Witryna Wolfram'a zawiera oprócz tego całą masę innych informacji i odsyłaczy. Warto w niej poszperać.

- Dla bardziej zaawansowanych: strona [Mathematica Stack Exchange](#).
- Niezła książka, od podstaw języka do zaawansowanych "tricków": Leonid Shifrin, ["Mathematica® programming: an advanced introduction"](#).
- Oczywiście stale korzystamy z pomocy dostępnej w menu: [Help > Wolfram Documentation](#)

Pierwszy kontakt

Omówimy tu tylko wybrane elementy języka Wolframa, żeby się trochę z nim oswoić.

```
x = 5
```

```
5
```

```
x ^ 23
```

```
11 920 928 955 078 125
```

```
y = 2 / 3
```

```
 $\frac{2}{3}$ 
```

```
y * 3 / 5
```

```
 $\frac{2}{5}$ 
```

```
y * 3. / 5
```

```
0.4
```

Wartość numeryczną możemy wyświetlić z dowolną precyzją (jedynym ograniczeniem jest pojemność pamięci komputera)

```
N[1 / 7]
```

```
0.142857
```

```
N[1 / 7, 20]
```

```
0.14285714285714285714
```

```
N[Pi, 769]
```

```

3.14159265358979323846264338327950288419716939937510582097494459230781640628\
620899862803482534211706798214808651328230664709384460955058223172535940812\
848111745028410270193852110555964462294895493038196442881097566593344612847\
564823378678316527120190914564856692346034861045432664821339360726024914127\
372458700660631558817488152092096282925409171536436789259036001133053054882\
046652138414695194151160943305727036575959195309218611738193261179310511854\
807446237996274956735188575272489122793818301194912983367336244065664308602\
139494639522473719070217986094370277053921717629317675238467481846766940513\
200056812714526356082778577134275778960917363717872146844090122495343014654\
958537105079227968925892354201995611212902196086403441815981362977477130996\
0518707211349999998

```

Miejsca od 762 do 767 po przecinku to tzw. punkt Feynmana. To wiem z książki Daniela Tammeta “Urodziłem się pewnego błękitnego dnia”. Tammet nauczył się na pamięć i publicznie wyrecytował pierwsze 22 500 cyfr liczby π , aby zebrać fundusze dla stowarzyszenia na rzecz epilepsji (NSE).

Kilka ważnych detali

Ważna niepisana zasada: wszystkie zmienne zaczynamy małą literą, żeby uniknąć przypadkowej pomyłki z nazwami funkcji wbudowanych.

Nazwy muszą się zaczynać od litery (małej!) i mogą zawierać tylko litery i cyfry. System odróżnia litery małe i duże!

Specjalna rola nawiasów: `[]` - argumenty funkcji, `{}` - lista, w arytmetyce tylko `()`.

Można łączyć wyrażenia w jednym polu Input, oddzielając je średnikiem. Wynik wyrażenia zakończony średnikiem nie będzie wyświetlony.

```
y = x ^ 2 ;
```

```
y = y - 2 ;
y * 2
```

```
46
```

Różnica między '=' a ':='

Znak równości (**Set**) powoduje, że obliczana jest prawa strona i przypisywana do symbolu po lewej. Zawsze, gdy pojawia się ten symbol, ma on tę samą wartość.

```
Clear[x, y]
```

```
x = 55.5
```

```
55.5
```

```
y + x
```

```
55.5 + y
```

```
x = Random[]
```

```
0.521855
```

```
x
```

```
0.521855
```

Znak ':=' (**SetDelayed**) powoduje, że prawa strona jest obliczana za każdym razem od nowa, gdy pojawia się symbol ze strony lewej:

```
y := Random[]
```

```
y
```

```
0.459038
```

```
x
```

```
0.315805
```


Definicja funkcji

```
mojaF[x_] := x^2 + 5
```

```
mojaF[3]
```

14

Zmienna x jest globalna, co może prowadzić do problemów jeśli o niej zapomnimy

```
x
```

0.521855

Dlatego bezpieczniej “chować” zmienne pomocnicze w otoczeniu **Module**

```
mojaF2[x_] := Module[{a},
  a = x^2;
  a + 5
]
```

```
mojaF2[3]
```

14

```
a
```

a

```
x
```

0.521855

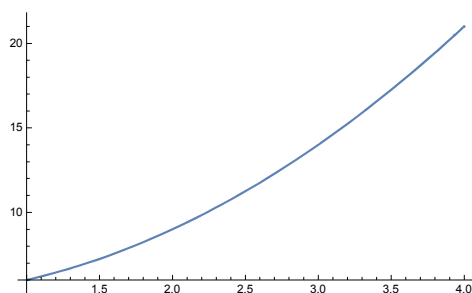
```
Clear[x]
```

```
x
```

x

Wykres funkcji

```
Plot[mojaF2[z], {z, 1, 4}]
```



Wiele funkcji, w tym `Plot`, pozwala na precyzowanie różnych opcji formatujących. Ich definicji, znaczenia i przykładów należy szukać w dokumentacji (Help). Szybką pomoc dostaniemy pisząc:

?? Plot

`Plot[f, {x, xmin, xmax}]` generates a plot of f as a function of x from x_{min} to x_{max} .

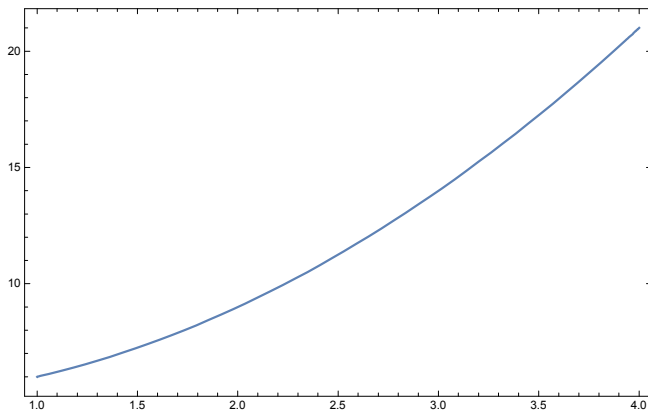
`Plot[{f1, f2, ...}, {x, xmin, xmax}]` plots several functions f_i .

`Plot[... , {x} ∈ reg]` takes the variable x to be in the geometric region reg . >>

`Attributes[Plot] = {HoldAll, Protected, ReadProtected}`

```
Options[Plot] = {AlignmentPoint → Center, AspectRatio →  $\frac{1}{\text{GoldenRatio}}$ , Axes → True, AxesLabel → None,
  AxesOrigin → Automatic, AxesStyle → {}, Background → None, BaselinePosition → Automatic, BaseStyle → {},
  ClippingStyle → None, ColorFunction → Automatic, ColorFunctionScaling → True, ColorOutput → Automatic,
  ContentSelectable → Automatic, CoordinatesToolOptions → Automatic, DisplayFunction → $DisplayFunction,
  Epilog → {}, Evaluated → Automatic, EvaluationMonitor → None, Exclusions → Automatic, ExclusionsStyle → None,
  Filling → None, FillingStyle → Automatic, FormatType → TraditionalForm, Frame → False, FrameLabel → None,
  FrameStyle → {}, FrameTicks → Automatic, FrameTicksStyle → {}, GridLines → None, GridLinesStyle → {},
  ImageMargins → 0., ImagePadding → All, ImageSize → Automatic, ImageSizeRaw → Automatic, LabelStyle → {},
  MaxRecursion → Automatic, Mesh → None, MeshFunctions → {#1 &}, MeshShading → None, MeshStyle → Automatic,
  Method → Automatic, PerformanceGoal → $PerformanceGoal, PlotLabel → None, PlotLabels → None,
  PlotLegends → None, PlotPoints → Automatic, PlotRange → {Full, Automatic}, PlotRangeClipping → True,
  PlotRangePadding → Automatic, PlotRegion → Automatic, PlotStyle → Automatic, PlotTheme → $PlotTheme,
  PreserveImageOptions → Automatic, Prolog → {}, RegionFunction → (True &), RotateLabel → True,
  TargetUnits → Automatic, Ticks → Automatic, TicksStyle → {}, WorkingPrecision → MachinePrecision}
```

```
Plot[mojaF2[z], {z, 1, 4},
  Frame → True,
  ImageSize → 500]
```



Listy

Lista, jako zbiór dowolnych elementów, jest bardzo ważnym obiektem. Będziemy ja spotykać na każdym kroku.

```
l = {1, 4, 6}
```

```
{1, 4, 6}
```

```
Range[10]
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
l2 = Range[7, 10]
```

```
{7, 8, 9, 10}
```

```
l3 = Join[l, l2]
```

```
{1, 4, 6, 7, 8, 9, 10}
```

Większość funkcji w działaniu na listę, działa na każdy element listy (ma atrybut **Listable**):

```
l3^2
```

```
{1, 16, 36, 49, 64, 81, 100}
```

```
2^l3
```

```
{2, 16, 64, 128, 256, 512, 1024}
```

```
13 * 4
```

```
{4, 16, 24, 28, 32, 36, 40}
```

```
Log[13]
```

```
{0, Log[4], Log[6], Log[7], Log[8], Log[9], Log[10]}
```

```
Attributes[Log]
```

```
{Listable, NumericFunction, Protected}
```

Nasza własna funkcja domyślnie nie ma tego atrybutu:

```
Attributes[mojaF2]
```

```
{}
```

Ale i tak dobrze działa na listy, bo składa się z operacji, które są **Listable**.

```
mojaF2[{2, 5, 7}]
```

```
{9, 30, 54}
```

Może się jednak zdarzyć funkcja bez tej własności, np.:

```
? If
```

```
If[condition, t, f] gives t if condition evaluates to True, and f if it evaluates to False.
```

```
If[condition, t, f, u] gives u if condition evaluates to neither True nor False. >>
```

```
mojaF3[x_] := If[x > 0, x * 5., -x / 5.]
```

```
mojaF3[{2, 3, -4}]
```

```
If[{2, 3, -4} > 0, {2, 3, -4} 5., - $\frac{\{2, 3, -4\}}{5.}$ ]
```

co możemy łatwo naprawić:

```
SetAttributes[mojaF3, Listable]
```

```
Attributes[mojaF3]
```

```
{Listable}
```

```
mojaF3[{2, 3, -4}]
```

```
{10., 15., 0.8}
```

```
5 > 7 - 2
```

```
False
```

Często wykreślamy wartości funkcji w zadanych punktach, które tworzą listę

```
tab0 = mojaF2[Range[0, 5, 0.5]]
```

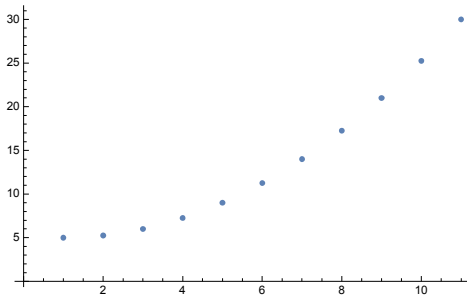
```
{5., 5.25, 6., 7.25, 9., 11.25, 14., 17.25, 21., 25.25, 30.}
```

Inna wersja tego samego, troche bardziej czytelna:

```
tab = Table[mojaF2[x], {x, 0, 5, 0.5}]
```

```
{5., 5.25, 6., 7.25, 9., 11.25, 14., 17.25, 21., 25.25, 30.}
```

```
ListPlot[tab]
```



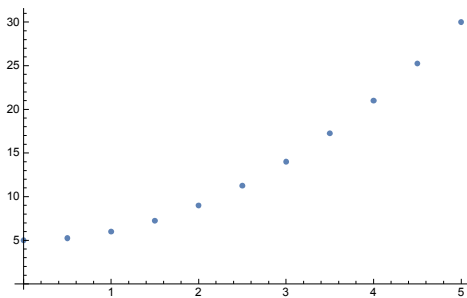
Nie całkiem o to nam chodziło.

Ale lista może być zagnieżdżona!

```
tab2 = Table[{x, mojaF2[x]}, {x, 0, 5, 0.5}]
```

```
{{0., 5.}, {0.5, 5.25}, {1., 6.}, {1.5, 7.25}, {2., 9.}, {2.5, 11.25},  
{3., 14.}, {3.5, 17.25}, {4., 21.}, {4.5, 25.25}, {5., 30.}}
```

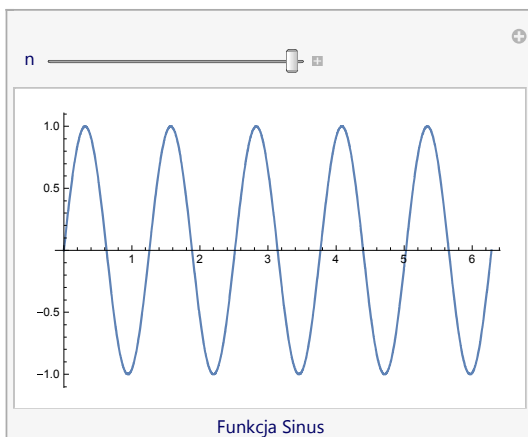
```
ListPlot[tab2]
```



Manipulate

Najważniejsza konstrukcja dla tego cyklu wykładów: funkcja **Manipulate** :

```
Manipulate[
(*Wyrażenie którym manipulujemy*)
Plot[Sin[n*x], {x, 0, 2 Pi}],
(* Kontrolki *)
{n, 1, 5},
(* Opcje formatowania *)
FrameLabel -> "Funkcja Sinus"
]
```



Manipulować można wszystkim:

```
Manipulate[
(*Wyrażenie którym manipulujemy*)
Expand[(a + b) ^ k],
(* Kontrolki *)
{{k, 2}, 1, 10, 1 / 2},
(* Opcje formatowania *)
FrameLabel -> {label1, label 2, "Dwumian Newtona, (a+b)^k", label4}
]
```

Dwumian Newtona, $(a+b)^k$

k

2 label $a^6 + 6 a^5 b + 15 a^4 b^2 + 20 a^3 b^3 + 15 a^2 b^4 + 6 a b^5 + b^6$ label4

label1

Użyteczne przykłady: wybór koloru

Mathematica oferuje rozbudowane metody obsługi koloru i przetwarzania obrazów. Graficzny wygląd obiektów matematycznych był od początku dźwignią reklamową pakietu. Tworząc wykresy, widma, histogramy, itp. możemy korzystać z tej obfitości. W kilku przykładach tworzenia obiektu **Manipulate** omówimy najważniejsze sposoby “kolorowania”.

Zobaczymy tylko czubek "góry lodowej". Zainteresowanych zachęcam do eksplorowania innych możliwości opisanych w dokumentacji.

Kolory nazwane (Named Colors)

Najłatwiej użyć jednego z 13 kolorów zdefiniowanych (nazwanych), które przywołujemy poprzez angielską nazwę.

```
Red
```



```
Blue
```



Dodatkowo można rozjaśnić lub pociemnić kolor funkcjami **Lighter** i **Darker**:

```
Lighter[Red]
```



```
Lighter[Blue]
```

```
Darker[Red]
```



```
Darker[Blue]
```

Stopień rozjaśnienia i przyciemnienia można kontrolować opcjonalnym parametrem α

wartościach między 0 i 1 (wartość domyślna = 1/3):

```
Lighter[Red, 0.9]
```



Ale zamiast sprawdzać ręcznie wszystkie możliwości, możemy zbudować interaktywną manipulację, która pozwoli łatwiej je eksplorować

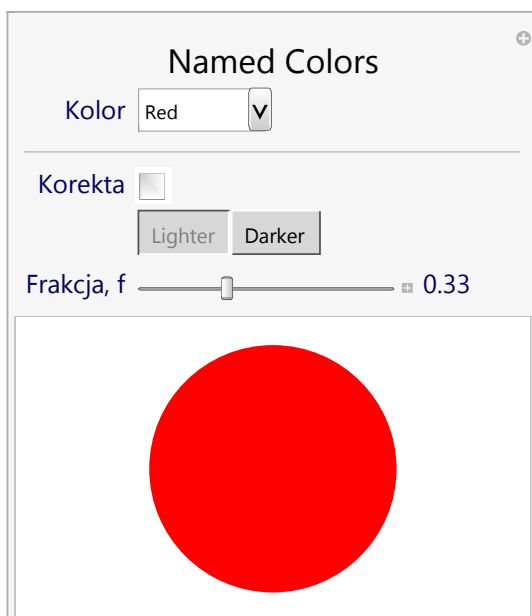
```
Manipulate[
  (*Wyrażenie którym manipulujemy*)
  fcolor = color;
  If[modify,
    If[cor == 1, fcolor = Lighter[fcolor, f],
      If[cor == -1, fcolor = Darker[fcolor, f]], ],
  ];
  Graphics[{fcolor, Disk[{0, 0}]}, ImageSize -> 200],
  (* Kontrolki *)
  {{color, Red, Style["Kolor", 20]},
   {Red -> "Red",
    Green -> "Green",
    Blue -> "Blue",
    Black -> "Black",
    White -> "White",
    Gray -> "Gray",
    Cyan -> "Cyan",
    Magenta -> "Magenta",
```



```

Yellow → "Yellow",
Brown → "Brown",
Orange → "Orange",
Pink → "Pink",
Purple → "Purple"},
ControlType → PopupMenu},
Delimiter,
{{modify, False, Style["Korekta", 20]},
 {False, True}, ControlType → Checkbox},
{{cor, 1, Style["", 18]}, {1 → Style[" Lighter ", Gray, 16],
 -1 → Style[" Darker ", Black, 16]}, Enabled → If[modify, True, False]},
{{f, 0.33, Style["Fracja, f", 20]}, 0, 1, Appearance → "Labeled",
 LabelStyle → 20, Enabled → modify},
(* Opcje i formatowania *)
FrameLabel → {None, None, Style["Named Colors", Black, 26], None},
TrackedSymbols :> {color, modify, cor, f}, Alignment → Center
]

```



Funkcja `RGBColor`

Domyślnym sposobem określania dowolnej barwy jest funkcja `RGBColor`, która ma trzy oczywiste parametry:

```
RGBColor[1, 0, 0]
```



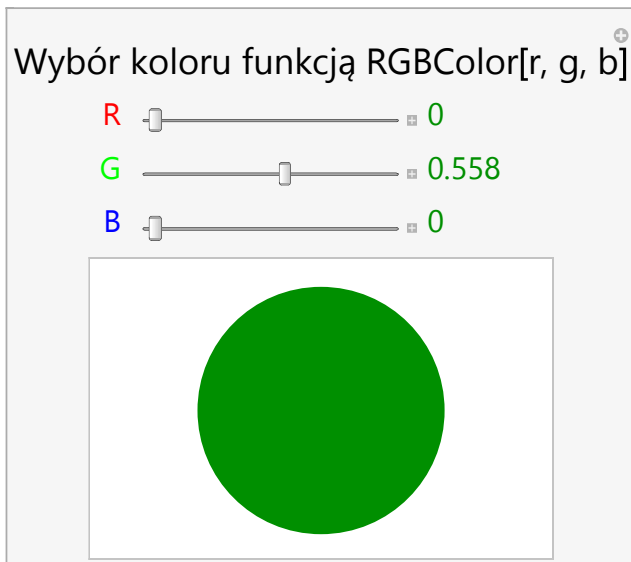
Ale jest jeszcze czwarty, opcjonalny parametr określający przezroczystość (`Opacity`)

```
RGBColor[1, 0, 0, 0.5]
```



Wpływ parametrów `r`, `g`, `b` na końcowy kolor zbadamy też interaktywnie

```
Manipulate[
  (*Wyrażenie którym manipulujemy*)
  Graphics[{RGBColor[r, g, b], Disk[{0, 0}]}, ImageSize -> 200],
  (* Kontrolki *)
  {{r, 0, Style["R ", Red, 24]}, 0, 1, Appearance -> "Labeled",
    LabelStyle -> Directive[RGBColor[r, g, b], 24]},
  {{g, 0, Style["G ", Green, 24]}, 0, 1, Appearance -> "Labeled",
    LabelStyle -> Directive[RGBColor[r, g, b], 24]},
  {{b, 0, Style["B ", Blue, 24]}, 0, 1, Appearance -> "Labeled",
    LabelStyle -> Directive[RGBColor[r, g, b], 24]},
  (* Opcje formatowania *)
  FrameLabel -> {None, None,
    Style["Wybór koloru funkcją RGBColor[r, g, b]", Black, 28], None},
  TrackedSymbols -> {r, g, b}, Alignment -> Center
]
```



Funkcja Hue

Czasem do wyboru koloru wygodna jest funkcja **Hue**, która ma trzy parametry. Ich znaczenie i wartości trudno zapamiętać. Poniższa funkcja ukazuje jak parametry te wpływają na kolor obiektu graficznego.

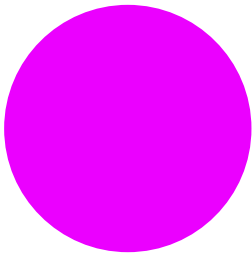
```
Manipulate[
  (*Wyrażenie którym manipulujemy*)
  Graphics[{Hue[h, s, b], Disk[{0, 0}]}, ImageSize -> 200],
  (* Kontrolki *)
  {{h, 0, Style["Hue ", Black, 24]}, 0, 1,
   Appearance -> "Labeled", LabelStyle -> Directive[Hue[h, s, b], 24]},
  {{s, 1, Style["Saturation ", Black, 24]}, 0, 1,
   Appearance -> "Labeled", LabelStyle -> Directive[Hue[h, s, b], 24]},
  {{b, 1, Style["Brightness ", Black, 24]}, 0, 1,
   Appearance -> "Labeled", LabelStyle -> Directive[Hue[h, s, b], 24]},
  (* Opcje formatowania *)
  FrameLabel ->
  {None, None, Style["Wybór koloru funkcją Hue[h, s, b]", Blue, 28], None},
  TrackedSymbols -> {h, s, b}, Alignment -> Center
]
```

Wybór koloru funkcją Hue[h, s, b]

Hue

Saturation

Brightness



Funkcja Blend

Kolory można mieszać, a stopień zmieszania kontrolować parametrem. To bywa bardzo przydatne, gdy chcemy przedstawić graficznie jakąś informację za pomocą koloru. Do mieszania służy funkcja **Blend** :

? Blend

`Blend[{col1, col2}, x]` gives a color obtained by blending a fraction $1 - x$ of color `col1` and x of color `col2`.
`Blend[{col1, col2, col3, ...}, x]` linearly interpolates between colors `col`, as x varies from 0 to 1.
`Blend[{x1, col1}, {x2, col2}, ...], x]` interpolates to give `col`, when $x = x_i$.
`Blend[{col1, col2, ...}, {u1, u2, ...}]` blends all the `col`, using fraction u_i of color `col`.
`Blend[{image1, image2, ...}, ...]` blends pixel values of 2D or 3D images `image`. >>


Sprawdzamy jak to działa, poznając przy okazji **ColorSlider**


```


Manipulate[
  (*Wyrażenie którym manipulujemy*)
  Graphics[{Blend[{c1, c2}, f], Disk[{0, 0}]}, ImageSize → 200],
  (* Kontrolki *)
  {{c1, Black, Style["Kolor 1", 24]}, ColorSlider, ImageSize → {250, 80}},
  {{c2, White, Style["Kolor 2", 24]}, ColorSlider, ImageSize → {250, 80}},
  Delimiter,
  {{f, 0, Style["Frakcja, f", 24]},
   0, 1, Appearance → "Labeled", LabelStyle → 24},
  (* Opcje formatowania *)
  FrameLabel → {None, None,
   Style["Mieszanie (Blend): (1-f)*K1 + f*K2", Black, 28], None},
  TrackedSymbols :> {c1, c2, f}, Alignment → Center
]

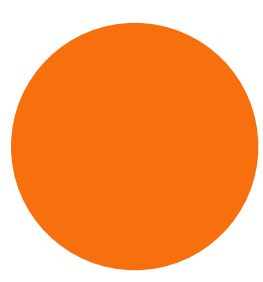
```

Mieszanie (Blend): $(1-f) \cdot K1 + f \cdot K2$

Kolor 1 

Kolor 2 

Frakcja, f  0.518



Cztery w jednym

Wszystkie cztery manipulacje możemy zebrać w jedną korzystając z opcji **TabView**.

Dodatkowo wprowadzamy przycisk "Insert Color", którego kliknięcie powoduje, że w miejscu kursora w poprzednio aktywnym noetobooku wstawia się aktualnie wybrana funkcja koloru.

```
TabView[{
  Style[" NamedColors ", Black, 22] →
  Manipulate[
    (*Wyrażenie którym manipulujemy*)
    fcolor = color;
    If[modify,
      If[cor == 1, fcolor = Lighter[fcolor, f],
        If[cor == -1, fcolor = Darker[fcolor, f]],],
    ];
  Column[{
    Graphics[{fcolor, Disk[{0, 0}]}, ImageSize → 200],
    Button[" Insert Color ",
      NotebookWrite[Notebooks[][[2]], fcolor],
      BaseStyle → {Bold, 22},
      ImageMargins → 5}], Center],
  (* Kontrolki *)
  {{color, Red, Style["Kolor", 20]},
```

```
{Red → "Red",
  Green → "Green",
  Blue → "Blue",
  Black → "Black",
  White → "White",
  Gray → "Gray",
  Cyan → "Cyan",
  Magenta → "Magenta",
  Yellow → "Yellow",
  Brown → "Brown",
  Orange → "Orange",
  Pink → "Pink",
  Purple → "Purple"},
  ControlType → PopupMenu},
  Delimiter,
  {{modify, False, Style["Korekta", 20]},
  {False, True}, ControlType → Checkbox},
  {{cor, 1, Style["", 18]}, {1 → Style[" Lighter ", Gray, 16],
  -1 → Style[" Darker ", Black, 16]}, Enabled → If[modify, True, False]},
  {{f, 0.33, Style["Frakcja, f", 20]}, 0, 1, Appearance → "Labeled",
  LabelStyle → 20, Enabled → modify},
  (* Opcje i formatowania *)
  FrameLabel → {None, None, Column[{Style[" Color, ", "Output", 24],
  Style["Lighter[Color,f], Darker[Color,f]", "Output", 24]}],
  Center, Spacings → 1}, None},
  TrackedSymbols → {color, modify, cor, f}, Alignment → Center
],
```

```

Style[" RGBColor ", Black, 22] →
Manipulate[
  (*Wyrażenie którym manipulujemy*)
  Column[{
    Graphics[{RGBColor[r, g, b], Disk[{0, 0}]}, ImageSize → 200],
    Button[" Insert Color ",
      NotebookWrite[Notebooks[][[2]], RGBColor[r, g, b]],
      BaseStyle → {Bold, 22},
      ImageMargins → 5]}, Center],
  (* Kontrolki *)
  {{r, 1, Style["R ", Red, 22]}, 0, 1, Appearance → "Labeled",
    LabelStyle → Directive[RGBColor[r, g, b], 24]},
  {{g, 0, Style["G ", Green, 22]}, 0, 1, Appearance → "Labeled",
    LabelStyle → Directive[RGBColor[r, g, b], 24]},
  {{b, 0, Style["B ", Blue, 22]}, 0, 1, Appearance → "Labeled",
    LabelStyle → Directive[RGBColor[r, g, b], 24]},
  (* Opcje i formatowania *)
  FrameLabel →
    {None, None, Style["RGBColor[r, g, b]          ", "Output", 24], None},
  TrackedSymbols → {r, g, b},
  ContentSize → {400, 400},
  Alignment → Center
  ],
Style[" Hue ", Black, 22] →
Manipulate[
  (*Wyrażenie którym manipulujemy*)
  Column[{

```

```

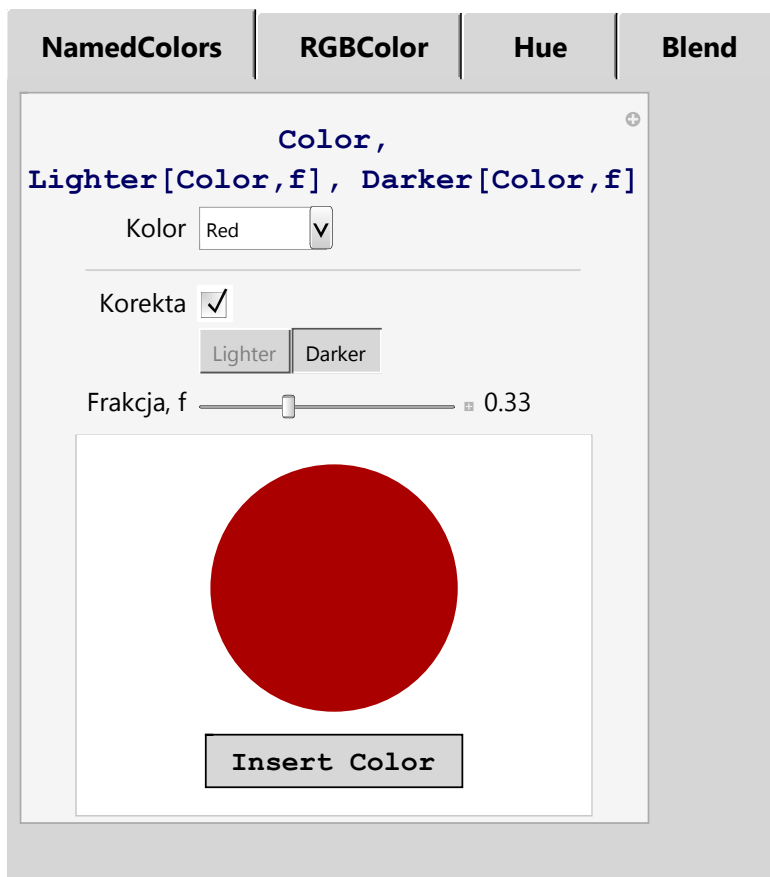
  Graphics[{Hue[h, s, b], Disk[{0, 0}]}, ImageSize → 200],
  Button[" Insert Color ",
    NotebookWrite[Notebooks[][[2]], Hue[h, s, b]],
    BaseStyle → {Bold, 22},
    ImageMargins → 5]}, Center],
  (* Kontrolki *)
  {{h, 0, Style["Hue ", Black, 22]}, 0, 1,
    Appearance → "Labeled", LabelStyle → Directive[Hue[h, s, b], 22]},
  {{s, 1, Style["Saturation ", Black, 22]}, 0, 1, Appearance → "Labeled",
    LabelStyle → Directive[Hue[h, s, b], 22]},
  {{b, 1, Style["Brightness ", Black, 22]}, 0, 1, Appearance → "Labeled",
    LabelStyle → Directive[Hue[h, s, b], 22]},
  (* Opcje i formatowania *)
  FrameLabel →
    {None, None, Style["Hue[h, s, b]          ", "Output", 24], None},
  TrackedSymbols → {h, s, b},
  Alignment → Center
  ],
Style[" Blend ", Black, 22] →
Manipulate[
  (*Wyrażenie którym manipulujemy*)
  Column[{
    Graphics[{Blend[{c1, c2}, f], Disk[{0, 0}]}, ImageSize → 200],
    Button[" Insert Color ",
      NotebookWrite[Notebooks[][[2]], Blend[{c1, c2}, f]],
      BaseStyle → {Bold, 22},

```

```

ImageMargins → 5]], Center],
(* Kontrolki *)
{{c1, Red, Style["Kolor 1", 22]}, ColorSlider, ImageSize → {330, 80}},
{{c2, Blue, Style["Kolor 2", 22]}, ColorSlider, ImageSize → {330, 80}},
Delimiter,
{{f, 0, Style["Frakcja, f", 22]},
 0, 1, Appearance → "Labeled", LabelStyle → 22},
(* Opcje i formatowania *)
FrameLabel → {None, None,
  Style[" Blend[k1,k2,f] = (1-f)*k1 + f*k2", "Output", 24], None},
TrackedSymbols → {c1, c2, f},
Alignment → Center
]
}]

```



Computable Data Format (CDF)

Firma Wolfram stworzyła specjalną platformę, która pozwala kreować interaktywne “manipulacje”, które z kolei dają się odtwarzać bez konieczności posiadania pakietu *Mathematica*. Ta platforma to Computable Data Format (CDF). Plik formatu cdf można oczywiście otwierać i edytować w programie *Mathematica*, ale do samego oglądania (bez możliwości edycji) wystarczy darmowy [Wolfram CDF Player](#).

Jako przykład utworzymy plik cdf zawierający ilustrację funkcji **Blend** (poprzednia manipulacja).

4. Zanaczamy komórki, które chcemy włączyć do pliku.
 5. Wybieramy z Menu: File > CDF Export > Standalone.
 6. Następnie postępujemy wg wskazówek.
- Na specjalnej stronie firmy Wolfram - [Wolfram Demonstration Project](#) - zebrane są tysiące interaktywnych manipulacji z rozmaitych dziedzin. Są one poklasyfikowane wg dziedzin tematycznych, co ułatwia eksplorowanie. Bardzo polecam!

Dwa ładne przykłady dotyczące koloru znalezione na WDP: [NamedColors](#) i [HSBExplorer](#).
Inne przykłady na stronie [autora](#).

Notebook z paskiem graficznym

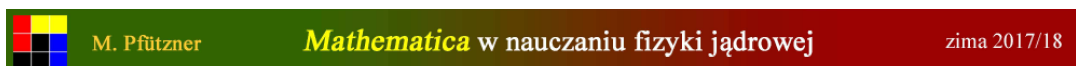
Wszystkie prezentacje wykładowe będą stworzone w programie *Mathematica* i wyświetlane w środowisku SlideShow.

- Aby żaden zastosowany efekt nie został bez wyjaśnienia, na koniec informacja jak stworzyć zeszyt (Notebook) z własnym paskiem graficznym na górze (banner).

Najpierw tworzymy własny obrazek w dowolnym programie graficznym i zapisujemy w formacie graficznym (jpg, png, ...) w stosownym folderze.

Wyszukujemy go i wczytujemy:

```
SetDirectory["d:\\Pfützner\\Teach\\Mathematica4Nuclear\\Style"]
d: \\Pfützner\\Teach\\Mathematica4Nuclear\\Style
FileNames["*.png"]
{Banner2.png, Banner.png}
g1 = Import["Banner2.png"]
```



Następnie tworzymy komórkę “zadokowaną” (*docked cell*) i dopasowujemy kolor tła do

koloru banneru:

```
banner = Cell[BoxData[ToBoxes[g1] /. ImageSize -> {1000, 70}],
  "DockedCell", Background -> Hue[0, 1, 0.6], ImageMargins -> 0,
  CellMargins -> {{0, 0}, {0, 0}}, CellFrameMargins -> {{0, 0}, {0, 0}}];
```

```
banner // CellPrint
```



M. Pfitzner

Mathematica w nauczaniu fizyki jądrowej

zima 2017/18

- Nowy zeszyt z bannerem tworzymy komendą:

```
CreateWindow[DockedCells -> banner];
```

- Możemy też od razu stworzyć plik do pokazu slajdów:

```
nbslide = CreateDocument[];
SetOptions[nbslide, DockedCells -> {FEPrivate`FrontEndResource[
  "FEExpressions", "SlideshowToolbar"], banner}];
```

Kształty Jąder Atomowych

Wizualizacja kształtów używanych do opisu zdeformowanych jąder atomowych

Wprowadzenie

Wiele zjawisk zachodzących w jądrach atomowych sugeruje, że są one zdeformowane. Choć kształt jądra nie jest obserwabłą kwantową, to pojęcie to jest bardzo pomocne w opisie i zrozumieniu budowy jąder.

W zależności od badanego problemu stosuje się różne parametryzacje kształtu. Często stosuje się rozwinięcie w szereg funkcji (harmonik) sferycznych. Tu przyjmijmy tę metodę, ale rozważymy też parametryzację stosowaną w modelu Nilssona, o którym będzie mowa w jednym z późniejszych wykładów.

Rozwinięcie na harmoniki sferyczne

- Opis powierzchni

Promień wodzący opisujący powierzchnię sferoidalnego kształtu:

$$R(\theta, \phi) = c(\alpha) R_0 \left[1 + \sum_{\lambda=2}^{\lambda_{\max}} \sum_{\mu=-\lambda}^{\lambda} a_{\lambda\mu} Y_{\lambda\mu}(\theta, \phi) \right].$$

R_0 to średni promień, $c(\alpha)$ to czynnik, który normalizuje objętość kształtu.

- Harmoniki sferyczne

Funkcje Y to dobrze znane (elektrodynamika, kwanty) harmoniki sferyczne. Ich ogólna definicja:

$$Y_{l,m} = \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} P_{l,m}(\cos\theta) e^{im\phi},$$

gdzie $P_{l,m}(\cos\theta)$ to tzw. stowarzyszone wielomiany Legendre'a.

- **Funkcje kwadrupolowe**

Najczęściej będziemy używać funkcji kwadrupolowych, czyli dla $\lambda=2$:

$$Y_{2,0} = \frac{1}{2} \sqrt{\frac{5}{4\pi}} (3 \cos^2(\theta) - 1)$$

$$Y_{2,-1} = \frac{1}{2} \sqrt{\frac{15}{2\pi}} \sin(\theta) \cos(\theta) e^{-i\phi}, \quad Y_{2,1} = -\frac{1}{2} \sqrt{\frac{15}{2\pi}} \sin(\theta) \cos(\theta) e^{i\phi}$$

$$Y_{2,-2} = \frac{1}{2} \sqrt{\frac{15}{4\pi}} \sin^2(\theta) e^{-2i\phi}, \quad Y_{2,2} = \frac{1}{2} \sqrt{\frac{15}{2\pi}} \sin^2(\theta) e^{2i\phi}$$

- **Promień neurojony**

Ponieważ promień musi być liczbą rzeczywistą, to współczynniki deformacji:

$$a_{\lambda\mu} = (-1)^\mu a_{\lambda,-\mu}.$$

Gdy kształt jest osiowo symetryczny, to znikają wszystkie współczynniki deformacji dla $\mu \neq 0$.

Dla pozostałych współczynników stosuje się wtedy oznaczenie: $a_{\lambda 0} \equiv \beta_\lambda$.

- **Harmoniki w Mathematica**

Mathematica oczywiście "zna" harmoniki sferyczne, więc zawsze można sprawdzić postać konkretnej funkcji:

```
SphericalHarmonicY[2, -1, θ, φ]
```

$$\frac{1}{2} e^{-i\phi} \sqrt{\frac{15}{2\pi}} \cos[\theta] \sin[\theta]$$

```
SphericalHarmonicY[2, 0, θ, φ]
```

$$\frac{1}{4} \sqrt{\frac{5}{\pi}} (-1 + 3 \cos[\theta]^2)$$

Definicje przydatnych funkcji

Aby skrócić pisanie, na użytek tego wykładu definiujemy skrót:

```
Y[l_, m_] := SphericalHarmonicY[l, m,  $\theta$ ,  $\varphi$ ]
```

```
Y[2, 0]
```

$$\frac{1}{4} \sqrt{\frac{5}{\pi}} (-1 + 3 \cos[\theta]^2)$$

Wykres funkcji $r(\theta)$ w 2D:

```
pplot2D[r_] := ParametricPlot[{Sin[ $\theta$ ] r, Cos[ $\theta$ ] r}, { $\theta$ , 0, 2 Pi},
  PlotRange -> {{-2, 2}, {-2, 2}},
  AxesLabel -> {x, z},
  AxesStyle -> Directive[Thick, Black, 18],
  PlotStyle -> {Blue, Thickness[0.008]},
  ImageSize -> 500
]
```

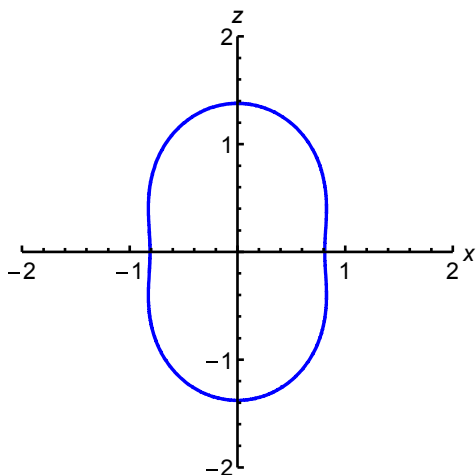
Wykres funkcji $r(\theta, \varphi)$ w 3D:

```
pplot3D[r_] := ParametricPlot3D[{Cos[ $\varphi$ ] Sin[ $\theta$ ] r, Sin[ $\varphi$ ] Sin[ $\theta$ ] r, Cos[ $\theta$ ] r},
  { $\theta$ , 0, Pi}, { $\varphi$ , 0, 2 Pi},
  AxesStyle -> Directive[Thick, Black, 16],
  AxesLabel -> {Style[x, Large, Bold],
    Style[y, Large, Bold],
    Style[z, Large, Bold]},
  PlotPoints -> 20,
  Mesh -> Full,
  ImageSize -> {400, 400}]
```

Kształty osiowo symetryczne w 2 D

Pouczające bywa spojrzenie na dwuwymiarowe projekcje kształtów sferoidalnych, szczególnie przy zachowaniu symetrii osiowej ($m = 0$). Pomijamy po prostu kąt ϕ i rysujemy harmoniki w płaszczyźnie (x, z) . Kształt trójwymiarowy powstaje przez obrót rysunku wokół osi z .

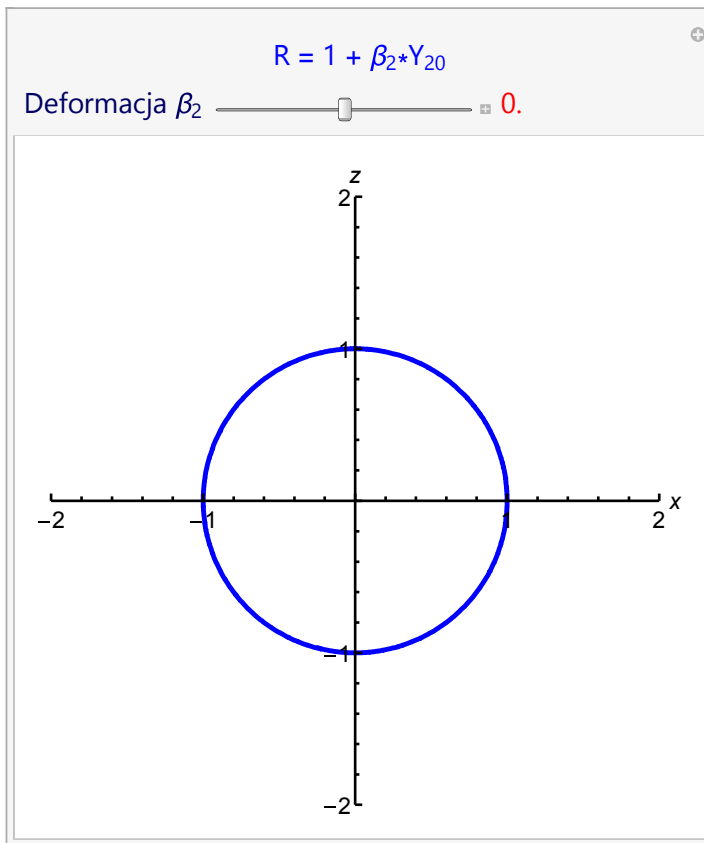
```
pplot2D[1 + 0.6 Y[2, 0]]
```



Manipulacja 1: β_2

Wprowadźmy możliwość interaktywnej zmiany deformacji:

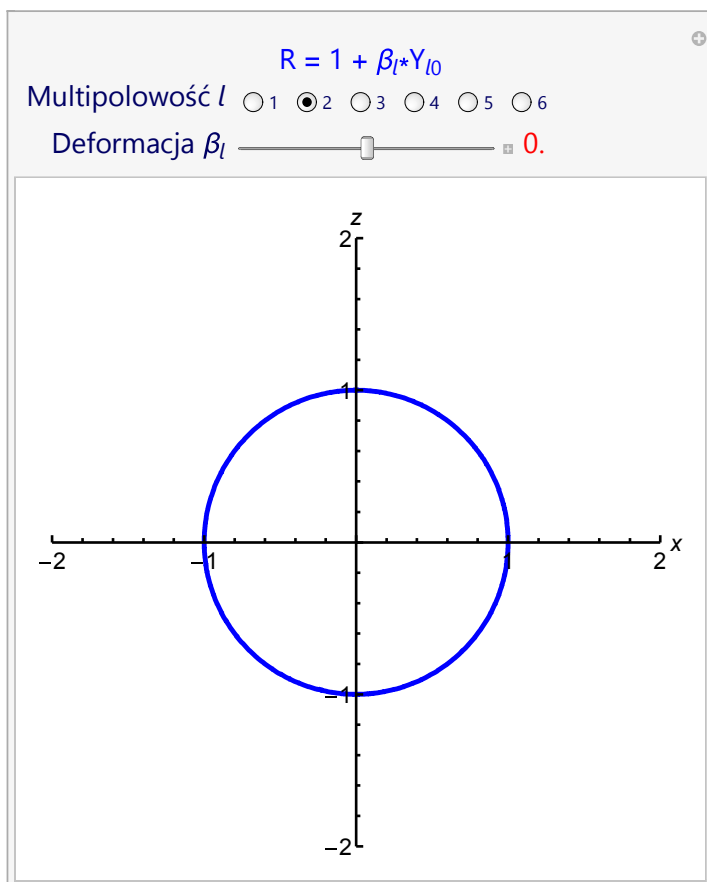
```
Manipulate[
Module[{r},
  r = 1 + beta2 * Y[2, 0];
  pplot2D[r],
  (* Kontrolki *)
  {{beta2, 0, Style["Deformacja  $\beta_2$ ", 22]}, -1.5, 1.5, 0.05,
  Appearance -> "Labeled", LabelStyle -> Directive[Red, 22]},
  FrameLabel -> {None, None, Style["R = 1 +  $\beta_2 * Y_{20}$ ", Blue, 22], None},
  SaveDefinitions -> True
]
```



Manipulacja 2: l, β_l

Dodajemy możliwość wyboru multipolowości:

```
Manipulate[Module[{r},
  r = 1 + betal * Y[mul, 0];
  pplot2D[r],
  (* Kontrolki *)
  {{mul, 2, Style["Multipolowość 1", 22]},
   Range[1, 6], ControlType -> RadioButton},
  {{betal, 0, Style["Deformacja  $\beta_1$ ", 22]}, -1.5, 1.5, 0.05,
   Appearance -> "Labeled", LabelStyle -> Directive[Red, 22]},
  FrameLabel -> {None, None, Style["R = 1 +  $\beta_1 * Y_{l0}$ ", Blue, 22], None},
  SaveDefinitions -> True
]
```



Kształty kwadrupolowe

Pierwszym krokiem w opisie kształtu jąder atomowych jest przyjęcie deformacji kwadrupolowej. Dlatego z tą deformacją mamy najczęściej do czynienia.

W ogólnym przypadku (już nie zakładamy symetrii osiowej!) i w układzie własnym jądra dwa współczynniki całkowicie określają kształt kwadrupolowy.

Przyjęto następującą konwencję:

$$\begin{aligned} a_{20} &= \beta \cos \gamma, \\ a_{22} = a_{2-2} &= \frac{1}{\sqrt{2}} \beta \sin \gamma, \\ a_{21} = a_{2-1} &= 0. \end{aligned}$$

Gdy $\gamma=0$, mamy kształt o symetrii osiowej: $a_{20} = \beta$, $a_{22} = 0$.

- Promień wodzący jako funkcja β i γ :

```
r22 = 1 + beta * Cos[gamma] * Y[2, 0] + beta * Sin[gamma] / Sqrt[2] * (Y[2, 2] + Y[2, -2])
```

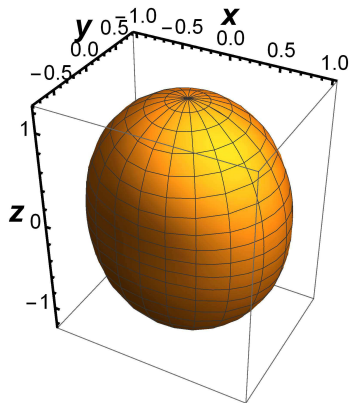

$$1 + \frac{1}{4} \sqrt{\frac{5}{\pi}} \beta \cos[\gamma] (-1 + 3 \cos[\theta]^2) +$$

$$\frac{\beta \sin[\gamma] \left(\frac{1}{4} e^{-2i\varphi} \sqrt{\frac{15}{2\pi}} \sin[\theta]^2 + \frac{1}{4} e^{2i\varphi} \sqrt{\frac{15}{2\pi}} \sin[\theta]^2 \right)}{\sqrt{2}}$$

```
r22 = Simplify[ExpToTrig[r22]]
```

$$\frac{1}{8} \left(8 + \sqrt{\frac{5}{\pi}} \beta \cos[\gamma] (1 + 3 \cos[2\theta]) + 2 \sqrt{\frac{15}{\pi}} \beta \cos[2\varphi] \sin[\gamma] \sin[\theta]^2 \right)$$

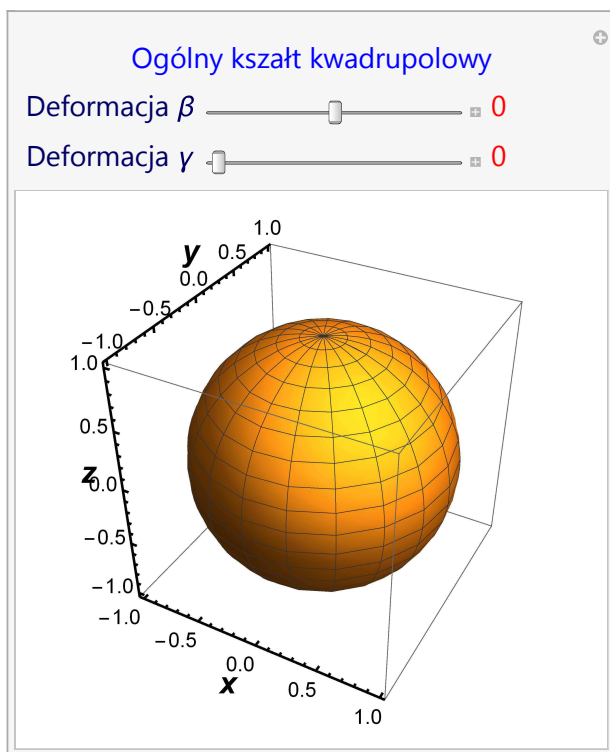
```
rkwad = r22 /. {beta -> 0.4, gamma -> 30 Degree};
pplot3D[rkwad]
```



Manipulacja 3: β , γ

Badamy wpływ parametrów β i γ :

```
Manipulate[Module[{a20, a22, r2},
  a20 = bet Cos[gam Degree];
  a22 = bet Sin[gam Degree] / Sqrt[2];
  r2 = 1 + a20 * Y[2, 0] + a22 (Y[2, 2] + Y[2, -2]);
  pplot3D[r2],
  {{bet, 0, Style["Deformacja  $\beta$ ", 22]}, -0.8, 0.8, 0.05,
  Appearance -> "Labeled", LabelStyle -> Directive[Red, 22]},
  {{gam, 0, Style["Deformacja  $\gamma$ ", 22]}, 0, 60, 5.,
  Appearance -> "Labeled", LabelStyle -> Directive[Red, 22]},
  FrameLabel -> {{None, None},
  {None, Style["Ogólny kształt kwadrupolowy", Blue, 22]}},
  SaveDefinitions -> True
]
```



Manipulacja 4: β , γ , stała objętość

Możemy jeszcze zapewnić stałą objętość.

Objętość sferoidy jest proporcjonalna do iloczynu pól głównych.

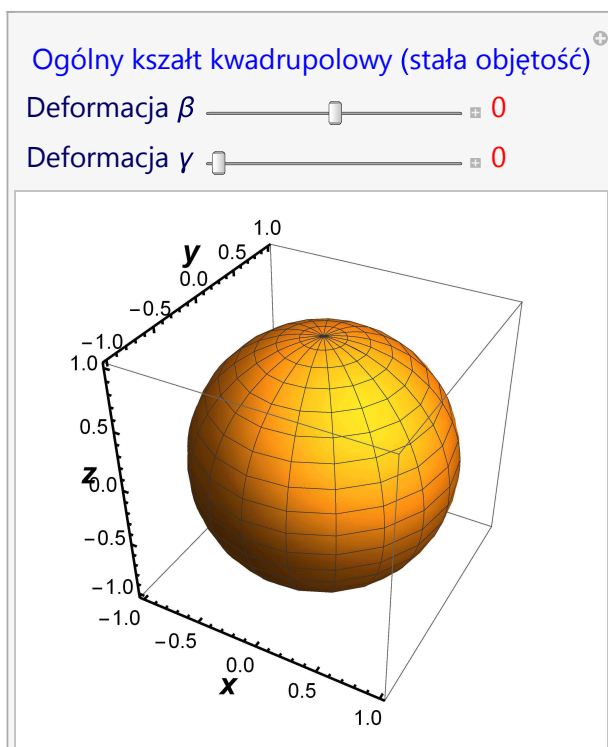
W naszym przypadku:

$$\begin{aligned} \mathbf{R}_x &= \mathbf{R}(\theta = \pi/2, \varphi = 0), \\ \mathbf{R}_y &= \mathbf{R}(\theta = \pi/2, \varphi = \pi/2), \\ \mathbf{R}_z &= \mathbf{R}(\theta = 0, \varphi = 0) \end{aligned}$$

```

Manipulate[Module[{a20, a22, r2, rx, ry, rz},
  a20 = bet Cos[gam Degree];
  a22 = bet Sin[gam Degree] / Sqrt[2];
  r2 = 1 + a20 * Y[2, 0] + a22 (Y[2, 2] + Y[2, -2]);
  rx = r2 /. {θ → Pi / 2, φ → 0};
  ry = r2 /. {θ → Pi / 2, φ → Pi / 2};
  rz = r2 /. {θ → 0, φ → 0};
  r2 = r2 / (rx * ry * rz);
  pplot3D[r2]],
{{bet, 0, Style["Deformacja β", 22]}, -0.8, 0.8, 0.05,
  Appearance -> "Labeled", LabelStyle -> Directive[Red, 22]},
{{gam, 0, Style["Deformacja γ", 22]}, 0, 60, 5.,
  Appearance -> "Labeled", LabelStyle -> Directive[Red, 22]},
FrameLabel -> {{None, None},
  {None, Style["Ogólny kształt kwadrupolowy (stała objętość)", Blue, 22]}},
SaveDefinitions -> True
]

```



Manipulacja 5: interaktywna mapa β, γ

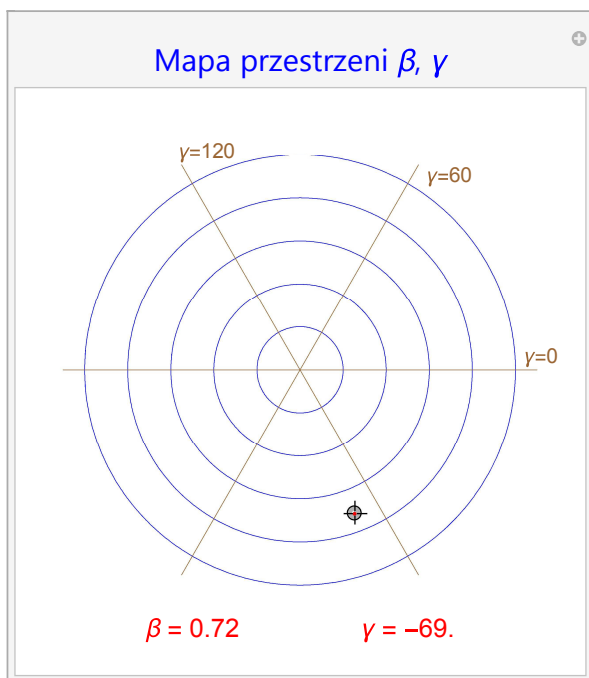
W poprzednich manipulacjach γ zmieniało się od 0 do 60 stopni, bo to wystarcza do opisu wszystkich kształtów. Chcemy teraz teraz to zobaczyć.

W pierwszym kroku tworzymy interaktywną mapę przestrzeni β, γ pozwalającą na wybór tych parametrów.

```
plcircles = PolarPlot[{0.2, 0.4, 0.6, 0.8, 1.0}, {t, 0, 2 Pi},
  PlotStyle -> Directive[Blue, Thin],
  PlotRange -> {{-1.15, 1.15}, {-1.25, 1.15}},
  ImageSize -> 400,
  Axes -> None];
```

```
plline1 = Graphics[{Thin, Brown,
  Line[{{-1.1, 0}, {1.1, 0}}], Text[Style[" $\gamma=0$ ", 16], {1.2, 0.07}, Right]}];
plline2 = Graphics[{Thin, Brown,
  Line[{{-1.1 * Cos[Pi / 3], -1.1 * Sin[Pi / 3]},
  {1.1 * Cos[Pi / 3], 1.1 * Sin[Pi / 3]}]}, Text[Style[" $\gamma=60$ ", 16],
  {1.1 * Cos[Pi / 3] + 0.25, 1.05 * Sin[Pi / 3]}, Right]}];
plline3 = Graphics[{Thin, Brown,
  Line[{{-1.1 * Cos[2 Pi / 3], -1.1 * Sin[2 Pi / 3]},
  {1.1 * Cos[2 Pi / 3], 1.1 * Sin[2 Pi / 3]}]}, Text[Style[" $\gamma=120$ ", 16],
  {1.1 * Cos[2 Pi / 3] + 0.25, 1.18 * Sin[2 Pi / 3]}, Right]}];
```

```
Manipulate[Module[{bet, gam},
  bet = NumberForm[Setting[Dynamic[First[ToPolarCoordinates[pt]]], 2];
  gam =
  NumberForm[Setting[Dynamic[Last[ToPolarCoordinates[pt]] / Degree], 2];
  Grid[{{
    LocatorPane[
      Dynamic[pt],
      Show[{plcircles, plline1, plline2, plline3},
        ImagePadding -> {{0, 0}, {0, 0}},
        Epilog -> {Red, Point[Dynamic[pt]],
          Text[Style[StringForm[" $\beta = \`1\`", bet], Red, 20], {-0.5, -1.2}],
          Text[Style[StringForm[" $\gamma = \`1\`", gam], Red, 20], {0.5, -1.2}]}
        ]
      }},
    ItemSize -> All]],
  {{pt, {0.1, 0}}, None},
  FrameLabel ->
  {{None, None}, {None, Style["Mapa przestrzeni  $\beta, \gamma$ ", Blue, 24]}},
  SaveDefinitions -> True
]$$ 
```



Manipulacja 6: kształty w przestrzeni β i γ

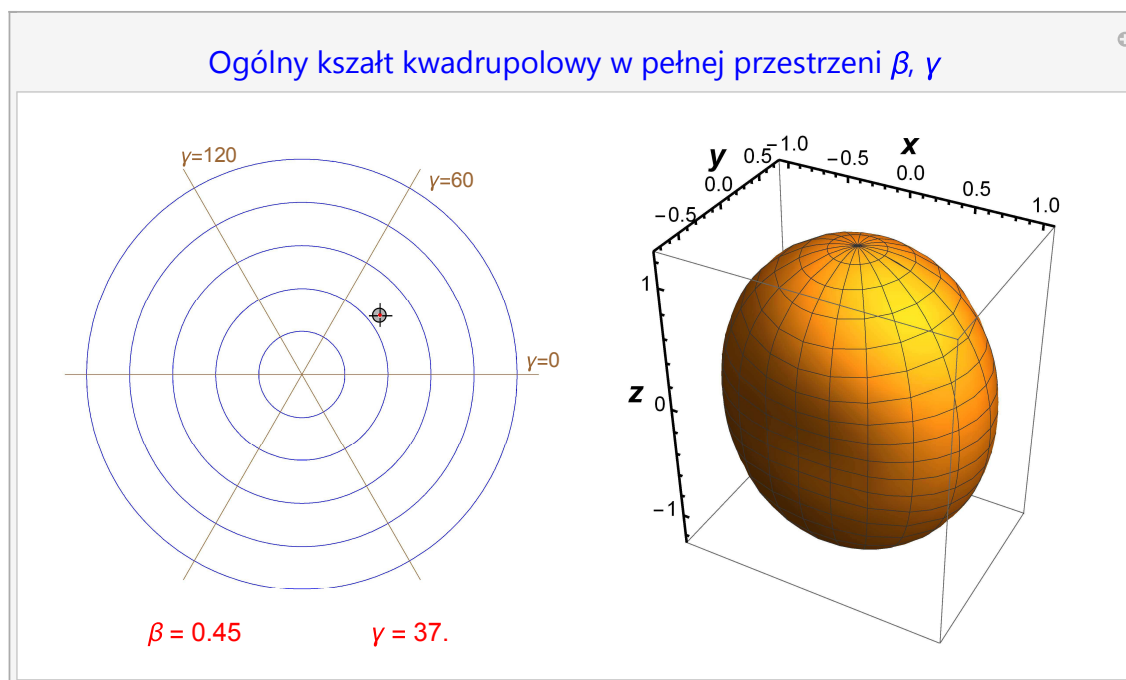
Łączymy teraz manipulację 3 i 5, czyli rysujemy ogólny kształt kwadrupolowy (bez stałej objętości) dla wybranych parametrów β, γ .

```
Manipulate[Module[{bet, gam, a20, a22, r2},
  bet = NumberForm[Setting[Dynamic[First[ToPolarCoordinates[pt]]]], 2];
  gam =
    NumberForm[Setting[Dynamic[Last[ToPolarCoordinates[pt]] / Degree]], 2];
  a20 = Setting[Dynamic[First[ToPolarCoordinates[pt]] *
    Cos[Last[ToPolarCoordinates[pt]]]], 2];
  a22 = Setting[Dynamic[First[ToPolarCoordinates[pt]] *
    Sin[Last[ToPolarCoordinates[pt]] / Sqrt[2]]], 2];
  r2 = 1 + a20 * Y[2, 0] + a22 * (Y[2, 2] + Y[2, -2]);
  Grid[{{
    LocatorPane[
      Dynamic[pt],
      Show[{plcircles, plline1, plline2, plline3},
        ImagePadding -> {{0, 0}, {0, 0}},
        Epilog -> {Red, Point[Dynamic[pt]],
          Text[Style[StringForm[" $\beta =$ `1`", bet], Red, 20], {-0.5, -1.2}],
          Text[Style[StringForm[" $\gamma =$ `1`", gam], Red, 20], {0.5, -1.2}]}
    }]}
  ],
  pplot3D[r2]
```

```

}},
ItemSize → All]],
{{pt, {0.1, 0}}, None},
FrameLabel → {{None, None}, {None, Style[
  "Ogólny kształt kwadrupolowy w pełnej przestrzeni  $\beta, \gamma$ ", Blue, 24]}}},
SaveDefinitions → True
]

```



Deformacja kwadрупolowa w modelu Nilssona

Model Nilssona bazuje na potencjale oscylatora harmonicznego. Deformację wprowadza się tu inaczej, rozważając oscylator harmoniczny anizotropowy.

Wiadomo, że częstość oscylatora jest odwrotnie proporcjonalna do promienia jądra:

$$\hbar\omega \approx \frac{41 \text{ MeV}}{A^{1/3}} \sim \frac{1}{R}.$$

Gdy zmieniamy rozmiar jądra wzdłuż jednej z osi, zmieniamy też częstość drgań w kierunku tej osi:

$$\omega_x = \omega_0 \frac{R_0}{a_x}, \quad \omega_y = \omega_0 \frac{R_0}{a_y}, \quad \omega_z = \omega_0 \frac{R_0}{a_z}.$$

Warunek stałej objętości prowadzi do:

$$\omega_x \omega_y \omega_z = \omega_0^3 \frac{R_0^3}{a_x a_y a_z} = \omega_0^3.$$

Symetria osiowa

W przypadku symetrii osiowej przyjmuje się zwyczajowo, że osią symetrii jest z.

Parametr deformacji w modelu Nilssona, δ :

$$\omega_x = \omega_y = \omega_0 (\delta) \left(1 + \frac{1}{3} \delta\right), \quad \omega_z = \omega_0 (\delta) \left(1 - \frac{2}{3} \delta\right).$$

```
ωxyz = Expand[(1 + δ / 3) * (1 + δ / 3) * (1 - 2 * δ / 3)]
```

$$1 - \frac{\delta^2}{3} - \frac{2 \delta^3}{27}$$

Półosie elipsoidy dane są przez:

$$a_x = a_y = \omega_0 R_0 / \omega_x, \quad a_z = \omega_0 R_0 / \omega_z.$$

W jednostkach R_0 wynoszą więc:

$$\frac{a_x}{R_0} = \frac{a_y}{R_0} = \frac{(1 - \delta^2/3 - 2\delta^3/27)^{1/3}}{1 + \delta/3}, \quad \frac{a_z}{R_0} = \frac{(1 - \delta^2/3 - 2\delta^3/27)^{1/3}}{1 - 2\delta/3}.$$

Możemy porównać parametr deformacji δ z wprowadzonym poprzednio parametrem β .

Promień zapisany przez harmoniki dla symetrii osiowej:

`r22 /. {γ → 0}`

$$\frac{1}{8} \left(8 + \sqrt{\frac{5}{\pi}} \beta (1 + 3 \cos[2\theta]) \right)$$

Długość półosi z dostajemy kładąc $\theta=0$:

`rzharm = r22 /. {γ → 0, θ → 0}`

$$\frac{1}{8} \left(8 + 4 \sqrt{\frac{5}{\pi}} \beta \right)$$

Półoś z Nilssona :

$$rzN = \frac{(1 - \delta^2/3 - 2\delta^3/27)^{1/3}}{1 - 2\delta/3}$$

$$\frac{(1 - \frac{\delta^2}{3} - \frac{2\delta^3}{27})^{1/3}}{1 - \frac{2\delta}{3}}$$

Widać, że w przybliżeniu (dla małych deformacji):

$$r_N \approx 1 + \frac{2}{3} \delta \approx 1 + \frac{1}{2} \sqrt{\frac{5}{\pi}} \beta = r_{\text{harm}},$$

czyli

$$\delta \approx \frac{3}{4} \sqrt{\frac{5}{\pi}} \beta, \text{ lub } \beta \approx \frac{4}{3} \sqrt{\frac{\pi}{5}} \delta.$$

Ale możemy to zrobić lepiej!

Wyznaczamy β jako funkcję δ :

`Solve[rzharm == rzN, β]`

$$\left\{ \left\{ \beta \rightarrow -\frac{2 \sqrt{\frac{\pi}{5}} \left(-3 + 2\delta + \left(- (3 + \delta)^2 (-3 + 2\delta) \right)^{1/3} \right)}{-3 + 2\delta} \right\} \right\}$$

`Flatten[%]`

$$\left\{ \beta \rightarrow -\frac{2 \sqrt{\frac{\pi}{5}} \left(-3 + 2\delta + \left(- (3 + \delta)^2 (-3 + 2\delta) \right)^{1/3} \right)}{-3 + 2\delta} \right\}$$

`beta = Flatten[Solve[rzharm == rzN, β]] [[1, 2]]`

$$-\frac{2 \sqrt{\frac{\pi}{5}} \left(-3 + 2\delta + \left(- (3 + \delta)^2 (-3 + 2\delta) \right)^{1/3} \right)}{-3 + 2\delta}$$

Przedstawiamy wynik jako szereg potęgowy względem δ


```
betaapprox = Series[beta, {delta, 0, 4}]
```

$$\frac{4}{3} \sqrt{\frac{\pi}{5}} \delta + \frac{2}{3} \sqrt{\frac{\pi}{5}} \delta^2 + \frac{32}{81} \sqrt{\frac{\pi}{5}} \delta^3 + \frac{58}{243} \sqrt{\frac{\pi}{5}} \delta^4 + O[\delta]^5$$

Manipulacja 7: elipsoida osiowa Nilssona

Przypominamy długości półosi elipsoidy Nilssona:

$$\frac{a_x}{R_o} = \frac{a_y}{R_o} = \frac{(1 - \delta^2/3 - 2\delta^3/27)^{1/3}}{1 + \delta/3}, \quad \frac{a_z}{R_o} = \frac{(1 - \delta^2/3 - 2\delta^3/27)^{1/3}}{1 - 2\delta/3}.$$

Zbadamy kształt elipsoidy w zależności od δ . Przy okazji wyświetlamy stosunek częstości wzdłuż osi z i osi prostopadłych oraz wartość parametru β odpowiadającą δ .

- Uwaga, teraz zadajemy kształt podając 3 składowe każdego punktu, a nie poprzez promień wodzący, który jest funkcją θ i φ . Dlatego nie możemy skorzystać ze skrótu `Plot3D`.

```
Manipulate[Module[{omega, po, pz, beta},
  omega = (1 - delta^2 / 3 - 2 * delta^3 / 27)^(1 / 3);
  po = omega / (1 + delta / 3);
  pz = omega / (1 - 2 * delta / 3);

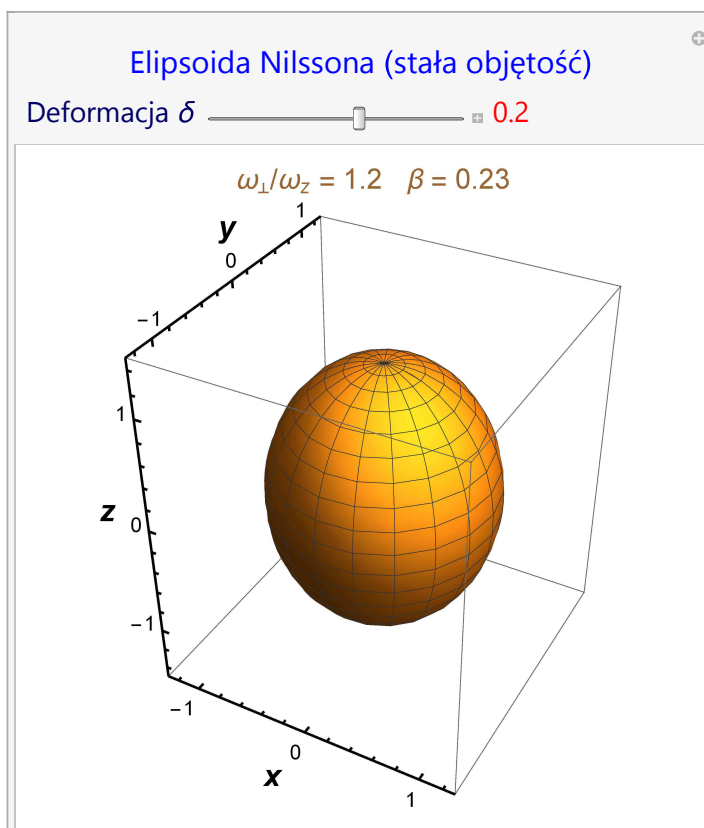
  beta = 4 / 3 * Sqrt[Pi / 5] delta + 2 / 3 * Sqrt[Pi / 5] delta^2;

  ParametricPlot3D[
    {Cos[phi] Sin[theta] * po, Sin[phi] Sin[theta] * po, Cos[theta] * pz}, {theta, 0, Pi}, {phi, 0, 2 Pi},
    PlotRange -> {{-1.3, 1.3}, {-1.3, 1.3}, {-1.5, 1.5}},
    AxesStyle -> Directive[Thick, Black, 16],
    AxesLabel -> {Style[x, Large, Bold],
      Style[y, Large, Bold],
```

```

Style[z, Large, Bold]},
PlotPoints → 20,
Mesh → Full,
ImageSize → {500, 500},
Axes -> True,
FaceGrids → None,
PlotLabel → Row[{
  Style[" $\omega_x/\omega_z =$  ", Brown, FontSize → 22],
  Style[NumberForm[pz / po, 2], Brown, FontSize → 22],
  Style["  $\beta =$  ", Brown, FontSize → 22],
  Style[NumberForm[beta, 2], Brown, FontSize → 22]}] ,
{{delta, 0, Style["Deformacja  $\delta$ ", 22]}, -1, 1, 0.2,
Appearance -> "Labeled", LabelStyle → Directive[Red, 22]},
FrameLabel → {{None, None},
{None, Style["Elipsoida Nilssona (stała objętość)", Blue, 24]}}
]

```



Jądro trójosiowe w modelu Nilssona

W przypadku deformacji trójosiowej, częstości wzdłuż trzech osi zapisuje się:

$$\begin{aligned}\omega_x &= \omega_0(\delta, \gamma) \left(1 - \frac{2}{3} \delta \cos(\gamma + 2\pi/3)\right), \\ \omega_y &= \omega_0(\delta, \gamma) \left(1 - \frac{2}{3} \delta \cos(\gamma - 2\pi/3)\right), \\ \omega_z &= \omega_0(\delta, \gamma) \left(1 - \frac{2}{3} \delta \cos\gamma\right),\end{aligned}$$

gdzie parametr γ opisuje odstępstwo od symetrii osiowej. Jest to ten sam parametr, co używany przy opisie harmonicznym.

Długości pól są teraz, jak poprzednio :

$$\begin{aligned}\frac{a_x}{R_0} &= \frac{(\omega_x \omega_y \omega_z)^{1/3}}{\omega_x}, \\ \frac{a_y}{R_0} &= \frac{(\omega_x \omega_y \omega_z)^{1/3}}{\omega_y}, \\ \frac{a_z}{R_0} &= \frac{(\omega_x \omega_y \omega_z)^{1/3}}{\omega_z}.\end{aligned}$$

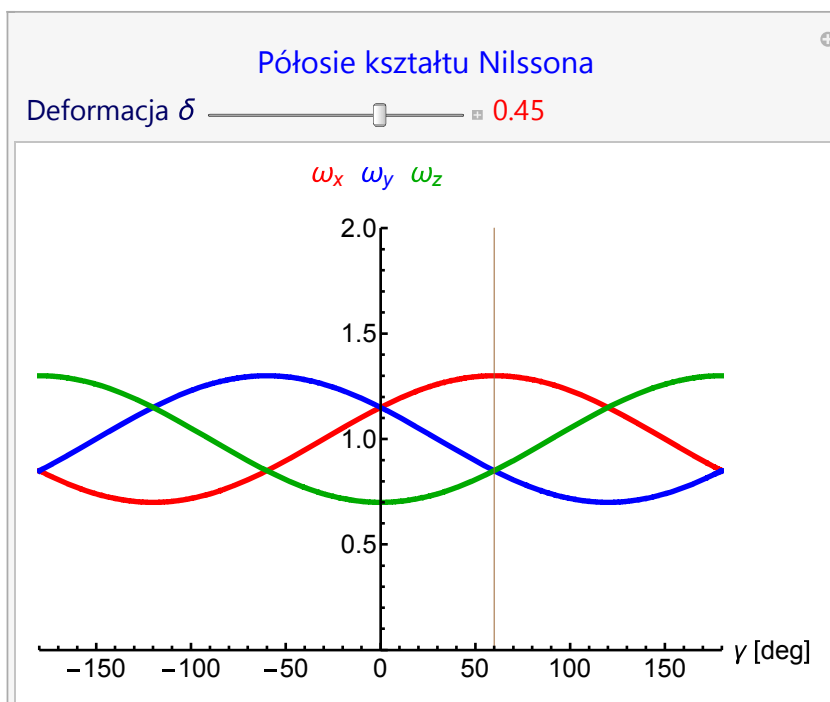
Manipulacja 8: pól w modelu Nilssona

Zobaczmy jak te długości zależą od δ i γ :

```

line60deg = Graphics[{Thin, Brown, Line[{{60, 0}, {60, 2}}]};
Manipulate[Module[{plotN3},
  plotN3 = Plot[{1 - 2 * delta * Cos[x * Pi / 180 + 2 * Pi / 3] / 3,
    1 - 2 * delta * Cos[x * Pi / 180 - 2 * Pi / 3] / 3,
    1 - 2 * delta * Cos[x * Pi / 180] / 3},
  {x, -180, 180},
  PlotRange -> {{-180, 180}, {0, 2}},
  AxesLabel -> {" $\gamma$  [deg]", ""},
  PlotLabel -> Row[{
    Style[" $\omega_x$  ", Red, FontSize -> 22],
    Style["  $\omega_y$  ", Blue, FontSize -> 22],
    Style["  $\omega_z$  ", Darker[Green], FontSize -> 22]
  }],
  AxesStyle -> Directive[Thick, Black, 20],
  PlotStyle -> {{Red, Thickness[0.008]},
    {Blue, Thickness[0.008]},
    {Darker[Green], Thickness[0.008]}},
  ImageSize -> 600];
Show[{plotN3, line60deg}]],
{{delta, 0, Style["Deformacja  $\delta$ ", 22]}, -1.2, 1.2, 0.05,
  Appearance -> "Labeled", LabelStyle -> Directive[Red, 22]},
FrameLabel -> {{None, None},
  {None, Style["Półośie kształtu Nilssona", Blue, 24]}},
SaveDefinitions -> True
]

```

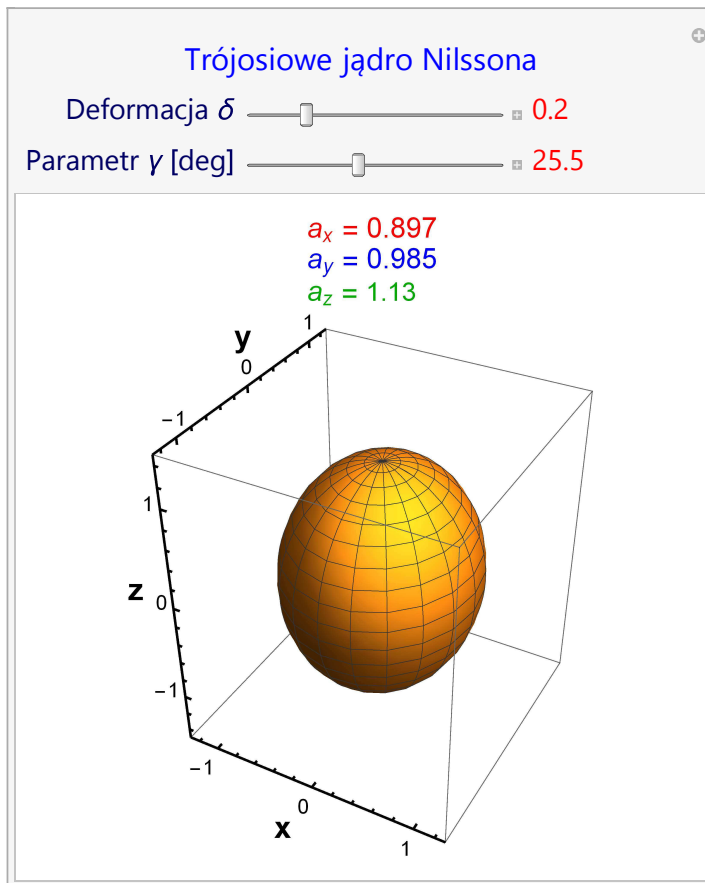


Manipulacja 9: trójosiowe jądro Nilssona

Mając półośie, możemy narysować kształt jądra trójosiowego w modelu Nilssona.

```
Manipulate[Module[{gamma, px, py, pz, V},
  gamma = ang * Pi / 180;
  px = 1 / (1 - 2 * delta * Cos[gamma + 2 * Pi / 3] / 3);
  py = 1 / (1 - 2 * delta * Cos[gamma - 2 * Pi / 3] / 3);
  pz = 1 / (1 - 2 * delta * Cos[gamma] / 3);
  V = (px * py * pz) ^ (1 / 3);
  ParametricPlot3D[{Cos[φ] Sin[θ] * px / V,
    Sin[φ] Sin[θ] * py / V, Cos[θ] * pz / V}, {θ, 0, Pi}, {φ, 0, 2 Pi},
  PlotRange → {{-1.3, 1.3}, {-1.3, 1.3}, {-1.5, 1.5}},
  Axes -> True,
  AxesStyle → Directive[Thick, Black, 16],
  AxesLabel → {Style["x", Large, Bold],
    Style["y", Large, Bold],
    Style["z", Large, Bold]},
  PlotPoints → 20,
  Mesh → Full,
  ImageSize → {500, 500},
  FaceGrids → None,
  PlotLabel → Column[
    {Row[{Style["ax = ", Red, FontSize → 22],
      Style[NumberForm[px / V, 3], Red, FontSize → 22]}],
```

```
  Row[{Style["ay = ", Blue, FontSize → 22],
    Style[NumberForm[py / V, 3], Blue, FontSize → 22]}],
  Row[{Style["az = ", Darker[Green], FontSize → 22],
    Style[NumberForm[pz / V, 3], Darker[Green], FontSize → 20]}] ]],
  ImageSize → 500]],
{{delta, 0, Style["Deformacja δ", 22]}, 0, 1, 0.1,
  Appearance -> "Labeled", LabelStyle → Directive[Red, 22]},
{{ang, 0, Style["Parametr γ [deg]", 22]}, 0, 60, 0.5,
  Appearance -> "Labeled", LabelStyle → Directive[Red, 22]},
FrameLabel → {{None, None},
  {None, Style["Trójosiowe jądro Nilssona", Blue, 24]}},
SaveDefinitions → True
]
```



Deformacje wyższych rzędów

- W fizyce jądrowej duże znaczenie mają też deformacje wyższych rzędów, szczególnie oktapolowe ($l=3$) i heksadekapolowe ($l=4$).
- Jednak rzadko odchodzi się od symetrii osiowej.
- Omówimy jeden interesujący przykład i bardziej rozbudowane narzędzie do wizualizacji kształtów opisanych przez wszystkie multipole od 1 do 4.

Kanciaste jądra?

W 2002 roku w Phys. Rev. Lett. ukazała się praca Jerzego Dudka i in., w której autorzy przewidują możliwość istnienia jąder o kształcie tetrahedralnym (czworościanu foremnego). Praca wzbudziła spore zainteresowanie. Dotychczas nie udało się jednak znaleźć przekonujących świadectw doświadczalnych na poparcie tej hipotezy.

VOLUME 88, NUMBER 25

PHYSICAL REVIEW LETTERS

24 JUNE 2002

Nuclear Tetrahedral Symmetry: Possibly Present throughout the Periodic Table

J. Dudek,¹ A. Goźdz,^{1,2} N. Schunck,¹ and M. Miśkiewicz²¹Institut de Recherches Subatomiques, IN2P3-CNRS/Université Louis Pasteur, F-67037 Strasbourg Cedex 2, France²Katedra Fizyki Teoretycznej, Uniwersytet Marii Curie-Skłodowskiej, PL-20031 Lublin, Poland

(Received 16 November 2001; published 11 June 2002)

→ [J. Dudek et al., Phys. Rev. Lett. 88, 252502 \(2002\)](#)

Kwadratura jądra

- Idea czworościennych jąder odbiła się echem także w prasie codziennej. Pisała o tym np. Gazeta Wyborcza:

Nauka

Kwadratura jądra, czyli polska wizja atomu

Piotr Cieślński | 17 czerwca 2002 | 00:00



Fizycy wyobrażali sobie dotychczas, że jądro atomowe przypomina kształtem kulę, czasami nieco wydłużoną. Tymczasem polscy naukowcy twierdzą, że jądra atomowe mogą być kanciaste! Odkrycie Polaków może zmienić kierunek badań nad superciężkimi pierwiastkami, bo - jak wiadomo - bryła kanciasta" toczy się w zupełnie inną stronę niż kulista...

Manipulacja 10: wizualizacja bogatszych kształtów

Nowe kształty sugerowane w pracy J. Dudka wymagały dodania wyrazów Y_{31} i Y_{32} , zawsze dotąd pomijanych (nieosiowy oktupol).

Na koniec zobaczymy więc jaki wpływ na kształt jądra mają deformacje oktupolowe ($l = 3$) i heksadekapolowe ($l = 4$).

- Wracamy do rozwinięcia na harmoniki sferyczne i rysujemy dowolny kształt dla $l=2,3,4$.

```
Manipulate[
Module[{r2, r3, r4, rn},
  r2 = a20 Y[2, 0] + a22 (Y[2, 2] + Y[2, -2]);
  r3 = a30 Y[3, 0] + a31 (Y[3, 1] - Y[3, -1]) +
    a32 (Y[3, 2] + Y[3, -2]) + a33 (Y[3, 3] - Y[3, -3]);
  r4 = a40 Y[4, 0] + a41 (Y[4, 1] - Y[4, -1]) + a42 (Y[4, 2] + Y[4, -2]) +
    a43 (Y[4, 3] - Y[4, -3]) + a44 (Y[4, 4] + Y[4, -4]);
  rn = 1 + r2 + r3 + r4;
  ParametricPlot3D[{Cos[φ] Sin[θ] rn, Sin[φ] Sin[θ] rn, Cos[θ] rn},
    {θ, 0, Pi}, {φ, 0, 2 Pi}, Axes → False, AspectRatio → Automatic,
    ImageSize → {400, 500}, ViewPoint → {2.6, -4.8, 2.0}],
  {{a20, 0, Style["a20", 20]}, 0, 1, 0.1, Appearance -> "Labeled",
    LabelStyle → Directive[Black, 20]},
  {{a22, 0, Style["a22", 20]}, 0, 1, 0.1, Appearance -> "Labeled",
```

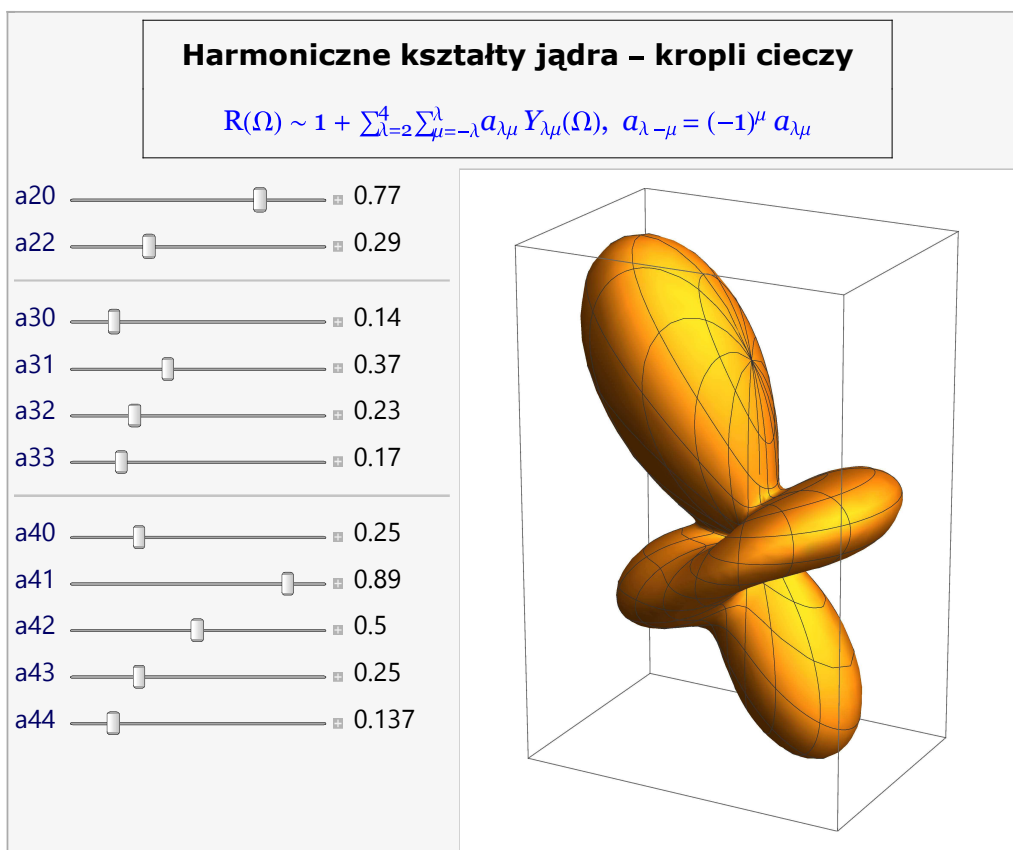
```
  LabelStyle → Directive[Black, 20]},
  Delimiter,
  {{a30, 0, Style["a30", 20]}, 0, 1, 0.1,
    Appearance -> "Labeled", LabelStyle → Directive[Black, 20]},
  {{a31, 0, Style["a31", 20]}, 0, 1, 0.1, Appearance -> "Labeled",
    LabelStyle → Directive[Black, 20]},
  {{a32, 0, Style["a32", 20]}, 0, 1, 0.1, Appearance -> "Labeled",
    LabelStyle → Directive[Black, 20]},
  {{a33, 0, Style["a33", 20]}, 0, 1, 0.1, Appearance -> "Labeled",
    LabelStyle → Directive[Black, 20]},
  Delimiter,
  {{a40, 0, Style["a40", 20]}, 0, 1, 0.1,
    Appearance -> "Labeled", LabelStyle → Directive[Black, 20]},
  {{a41, 0, Style["a41", 20]}, 0, 1, 0.1, Appearance -> "Labeled",
    LabelStyle → Directive[Black, 20]},
  {{a42, 0, Style["a42", 20]}, 0, 1, 0.1, Appearance -> "Labeled",
    LabelStyle → Directive[Black, 20]},
  {{a43, 0, Style["a43", 20]}, 0, 1, 0.1, Appearance -> "Labeled",
    LabelStyle → Directive[Black, 20]},
  {{a44, 0, Style["a44", 20]}, 0, 1, 0.1, Appearance -> "Labeled",
    LabelStyle → Directive[Black, 20]},
  ControlPlacement → Left,
  TrackedSymbols → {a20, a22, a30, a31, a32, a33, a40, a41, a42, a43, a44},
  FrameLabel → {None, None, fulltitle, None},
  AppearanceElements → None,
  SaveDefinitions → True,
```



```

Initialization :> (
  title1 = Graphics[Text[Style["Harmoniczne kształty jądra - kropli cieczy",
    Black, Bold, FontFamily -> "Verdana", 22]]];
  title2 = Graphics[Text[Style[
    "R(Ω) ~ 1 + ∑λ=24 ∑μ=-λλ aλμ Yλμ(Ω), aλ-μ = (-1)μ aλμ",
    Blue, FontFamily -> "Georgia", 22]]];
  fulltitle = GraphicsColumn[{title1, title2}, AspectRatio -> 0.18,
    ImageSize -> 600, Frame -> True];
  Y[l_, m_] := SphericalHarmonicY[l, m, θ, φ]
)
]

```



Własności nuklidów

Korzystamy z funkcji `IsotopeData`

Wstęp

Mathematica oferuje coraz więcej informacji o świecie i zawiera funkcje, które pozwalają do nich dotrzeć, jeśli mamy dostęp do internetu.

Zakres usług Wolfram|Alpha rozwija się bardzo gwałtownie i stan tego, co możemy zrobić szybko się dezaktualizuje.

Dla nas w tym kursie (w dydaktyce i w pracy) bardzo przydatna może być funkcja `IsotopeData`.

Zazwyczaj korzystamy z tej funkcji tak:

```
IsotopeData[{Z, A}, "własność"],  
lub IsotopeData["nazwa", "własność"]
```

Ale są ważne wyjątki. Np. lista wszystkich nuklidów:

```
IsotopeData[] // Short
```

```
{Entity[Isotope, Neutron], hydrogen, deuterium, tritium, hydrogen-4,
hydrogen-5, hydrogen-6, hydrogen-7, helium-2, helium-3, helium-4,
helium-5, helium-6, helium-7, helium-8, helium-9, helium-10, lithium-3,
lithium-4, lithium-5, <<3144>>, ununtrium-287, flerovium-285, flerovium-286,
flerovium-287, flerovium-288, flerovium-289, ununpentium-287, ununpentium-288,
ununpentium-289, ununpentium-290, ununpentium-291, livermorium-289,
livermorium-290, livermorium-291, livermorium-292, ununseptium-291,
ununseptium-292, ununseptium-293, ununseptium-294, ununoctium-293}
```

Lista jest zbyt długa, by wszystko wyświetlić. Zobaczmy ile ich jest:

```
Length[IsotopeData[]]
```

```
3184
```

Podając tylko liczby Z i A dostaniemy nazwę :

```
IsotopeData[{26, 45}]
```

```
iron-45
```

```
IsotopeData["Iron"]
```

```
{iron-45, iron-46, iron-47, iron-48, iron-49, iron-50, iron-51,
iron-52, iron-53, iron-54, iron-55, iron-56, iron-57, iron-58,
iron-59, iron-60, iron-61, iron-62, iron-63, iron-64, iron-65,
iron-66, iron-67, iron-68, iron-69, iron-70, iron-71, iron-72}
```

Podając tylko Z lub symbol chemiczny dostajemy listę izotopów:

```
IsotopeData[26]
```

```
{iron-45, iron-46, iron-47, iron-48, iron-49, iron-50, iron-51,
iron-52, iron-53, iron-54, iron-55, iron-56, iron-57, iron-58,
iron-59, iron-60, iron-61, iron-62, iron-63, iron-64, iron-65,
iron-66, iron-67, iron-68, iron-69, iron-70, iron-71, iron-72}
```

No i wreszcie lista możliwych własności :

```
IsotopeData["Properties"]
```

```
{AtomicMass, AtomicNumber, BindingEnergy, BranchingRatios, DaughterNuclides,  
DecayEnergies, DecayModes, DecayModeSymbols, DecayProducts,  
ExcitedStateEnergies, ExcitedStateHalfLives, ExcitedStateLifetimes,  
ExcitedStateParities, ExcitedStateSpins, ExcitedStateWidths, FullSymbol,  
HalfLife, IsotopeAbundance, Lifetime, MagneticMoment, MassExcess,  
MassNumber, Memberships, Name, NeutronNumber, Parity, QuadrupoleMoment,  
QuantumStatistics, Spin, Stable, StandardName, Symbol, Width}
```

Proste przykłady

```
IsotopeData["He4", "BindingEnergy"]
```

7.073915 MeV

```
IsotopeData["45Fe", "DecayModes"]
```

{TwoProtonEmission, BetaPlusDelayedProtonEmission}

```
IsotopeData["45Fe", "DecayModeSymbols"]
```

{2p, β^+p }

```
IsotopeData["45Fe", "HalfLife"]
```

0.00189 s

```
IsotopeData["45Fe", "BranchingRatios"]
```

{0.75, 0.25}

Lista nuklidów należących do klasy:

```
IsotopeData["TwoProtonEmission"]
```

```
{beryllium-6, boron-6, carbon-8, oxygen-12, neon-16, magnesium-19,
sulfur-26, argon-30, calcium-34, titanium-38, chromium-42,
iron-45, nickel-48, zinc-54, germanium-58, germanium-59}
```

Możemy zobaczyć wszystkie własności nuklidu zawarte w bazie:

```
IsotopeData["45Fe", #] & /@ IsotopeData["Properties"]
```

```
{45.014578 u, 26, 7.318 MeV, {0.75, 0.25},
{chromium-43, chromium-44}, {1134.5 keV, Missing[NotAvailable]},
{TwoProtonEmission, BetaPlusDelayedProtonEmission}, {2p,  $\beta^+p$ },
{{{Proton}, Missing[NotAvailable]}, {{ElectronNeutrino}, 0.4300},
{{Proton, Positron, ElectronNeutrino}, 0.4300}}, {}, {}, {},
 {}, {}, {},  ${}^{45}_{26}\text{Fe}_{19}$ , 0.00189 s, 0., 0.00273 s, Missing[Unknown],
13.579 MeV, 45, {Fermion, Unstable, TwoProtonEmission,
BetaPlusDecay, BetaPlusDelayedProtonEmission}, iron-45, 19,
1, Missing[Unknown], -1,  $\frac{3}{2}$ , False, Iron45,  ${}^{45}\text{Fe}$ ,  $1.67 \times 10^{-19}$  MeV}
```

Taki zapis własności jest bardzo niewygodny, trudno ustalić znaczenie niektórych pozycji. Możemy zrobić to lepiej tworząc tabelę wartości wraz z ich opisem, a potem ładnie ją wyświetlając. Oba kroki zbieramy w jednej procedurze:

```
isoProperties[iso_] := Module[{l},
  l = {IsotopeData[iso, #, "Description"],
    Shallow[IsotopeData[iso, #], {6, 8}] & /@ IsotopeData["Properties"];
  Text[Grid[Prepend[l, {"Property", "Value"}], Frame → All,
    Background → {None, {{{LightBlue, White}}, {1 → LightYellow}}},
    ItemStyle → Directive[FontSize → 18, Bold]]
]
```

```
isoProperties["45Fe"]
```

IsotopeData::notsubprop: "Description" is not a known subproperty of IsotopeData. Use subproperty "Properties" for a list of subproperties. >>

Property	Value
atomic mass	45.014578 u
atomic number	26
binding energy	7.318 MeV
branching ratios	{0.75, 0.25}
daughter nuclides	{chromium-43, chromium-44}
decay modes	{1134.5 keV, Missing[NotAvailable]}
decay products	{TwoProtonEmission, BetaPlusDelayedProtonEmission}
excited state energies	{2p, β^+p }
excited state half-lives	{{{Proton}, Missing[NotAvailable]}, {{ElectronNeutrino}, 0.4300}, {{Proton, Positron, ElectronNeutrino}, 0.4300}}
excited state lifetimes	{}
excited state parities	{}
excited state spins	{}

excited state widths	{}
full symbol	${}^{45}_{26}\text{Fe}_{19}$
physical half-life	0.00189 s
isotope abundance	0.
lifetime	0.00273 s
magnetic dipole moment	Missing[Unknown]
mass excess	13.579 MeV
mass number	45
memberships	{Fermion, Unstable, TwoProtonEmission, BetaPlusDecay, BetaPlusDelayedProtonEmission}
name	iron-45
neutron number	19
parity	1
electric quadrupole moment	Missing[Unknown]
quantum statistics	-1
spin	$\frac{3}{2}$
stable	False
IsotopeData[45Fe, StandardName, Description]	Iron45
symbol	${}^{45}\text{Fe}$
width	1.67×10^{-19} MeV

Moment magnetyczny

Ważną własnością stanów jądrowych jest **magnetyczny moment dipolowy**. Jego wartość jest bardzo czułym testem dla modeli jądrowych. Odegrał np. bardzo ważną rolę w pierwszych pracach o modelu powłokowym.

Model jednocząstkowy

Łatwo obliczyć moment dipolowy pochodzący od jednego nukleonu. Ma on dwie składowe: jedna pochodzi od orbitalnego, druga od spinowego momentu pędu:

$$\vec{\mu} = (g_l \vec{l} + g_s \vec{s}) \mu_N,$$

gdzie g_l i g_s to tzw. czynniki żyromagnetyczne odpowiednio dla części orbitalnej i spinowej, a μ_N to magneton jądrowy (tu naturalna jednostka): $\mu_N = e\hbar/2m_p = 3.152 \cdot 10^{-14}$ MeV/T. Czynniki g są inne dla protonu i neutronu:

$$\begin{aligned} \text{proton} &: g_l = 1, \quad g_s = 5.586 \\ \text{neutron} &: g_l = 0, \quad g_s = -3.826 \end{aligned}$$

Wartość magnetycznego momentu dipolowego definiuje się jako wartość oczekiwaną z-owej składowej wektora $\vec{\mu}$ w stanie o maksymalnym rzucie \vec{j} na oś z:

$$\mu = \langle j, m = j | \mu_z | j, m = j \rangle.$$

Linie Schmidta

Gdy o momencie magnetycznym decyduje pojedynczy nieparzysty nukleon, to w stanie o całkowitym momencie pędu $j=l \pm 1/2$ wartość μ wynosi:

$$\mu_{l \pm 1/2} = j \left[g_l \pm \frac{(g_s - g_l)}{2l + 1} \right] \mu_N.$$

Wstawiając odpowiednie wartości czynników g , dostajemy dla **protonów**:

$$\begin{aligned} \mu_{l+1/2} &= (j + 2.29) \mu_N, \\ \mu_{l-1/2} &= \left(j - 2.29 \frac{j}{j+1} \right) \mu_N, \end{aligned}$$

i dla **neutronów**:

$$\begin{aligned} \mu_{l+1/2} &= -1.91 \mu_N, \\ \mu_{l-1/2} &= 1.91 \frac{j}{j+1} \mu_N. \end{aligned}$$

Powyższe wzory, to tzw. linie Schmidta.

Wartości doświadczalne

Wartość spinu i momentu magnetycznego dla zadanego nuklidu:

```
magMom[iso_] := Module[{spi, mi},
  spi = IsotopeData[iso, "Spin"];
  mi = IsotopeData[iso, "MagneticMoment"];
  {spi, mi}
]
```

```
magMom["37Ca"]
```

```
{ $\frac{3}{2}$ , Missing[Unknown]}
```

Dopiszmy do wyniku doświadczalnego przewidywane wartości Schmidta, pierwsza dla $j=l-1/2$, druga dla $j=l+1/2$:

```
magMomSch[iso_] := Module[{z, n, s, m, s1, s2},
  z = IsotopeData[iso, "AtomicNumber"];
  n = IsotopeData[iso, "NeutronNumber"];
  s = IsotopeData[iso, "Spin"];
  m = IsotopeData[iso, "MagneticMoment"];
  If[OddQ[n + z],
    If[OddQ[z],
      s1 = s - 2.29 * s / (s + 1);
      s2 = s + 2.29,
      s1 = 1.91 * s / (s + 1);
      s2 = -1.91
    ],
    s1 = 0;
    s2 = 0
  ];
  {s, m, s1, s2}
]
```

```
magMomSch["207Pb"]
```

```
{ $\frac{1}{2}$ , 0.592583  $\mu_N$ , 0.636667, -1.91}
```

Zróbmy listę wszystkich danych doświadczalnych, osobno dla nuklidów z nieparzystą liczbą protonów i z nieparzystą liczbą neutronów.

Wybór jąder z nieparzystym protonem: najpierw tworzymy listę nuklidów o parzystym N i

nieparzystym Z , potem sprawdzamy ich moment magnetyczny, na koniec usuwamy przypadki, w których ta wartość nie jest znana (zmierzona):

```
lisMagMomOddZ = Module[{loddZ},
  loddZ = Select[IsotopeData[], (EvenQ[IsotopeData[#, "NeutronNumber"]]) &&
    (OddQ[IsotopeData[#, "AtomicNumber"]]) &];
  Cases[
    magMom[#] & /@ loddZ, Except[[_ , _Missing]]]
];
```

Analogicznie dla nieparzystego neutronu:

```
lisMagMomOddN = Module[{loddN},
  loddN = Select[IsotopeData[], (OddQ[IsotopeData[#, "NeutronNumber"]]) &&
    (EvenQ[IsotopeData[#, "AtomicNumber"]]) &];
  Cases[
    magMom[#] & /@ loddN, Except[[_ , _Missing]]]
];
```

```
Length[lisMagMomOddZ]
```

220

```
Length[lisMagMomOddN]
```

240

Porównanie

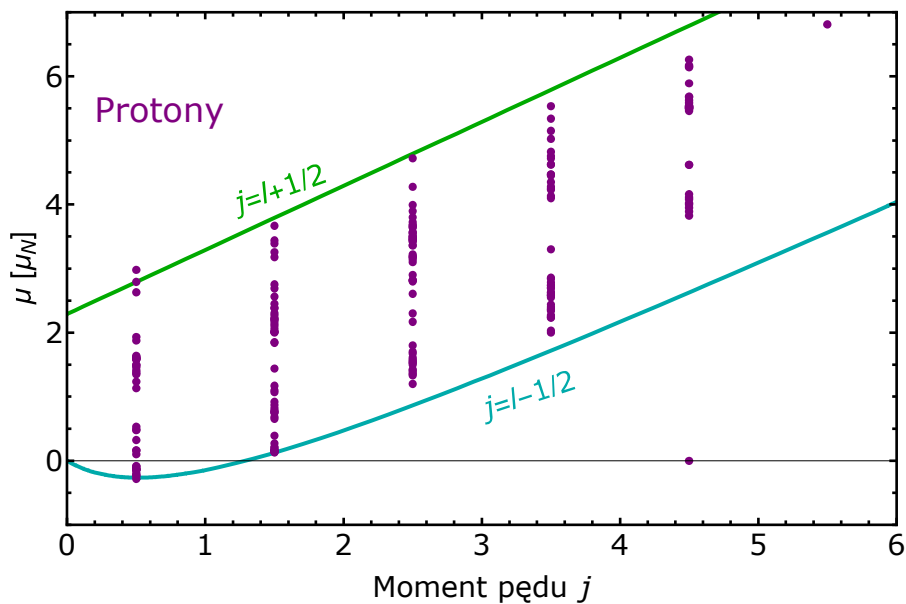
Porównujemy wartości doświadczalne z liniami Schmidta, rysujemy μ w funkcji j .

Najpierw **protony**:

```
gSchProt = Plot[{j + 2.29, j - 2.29 * j / (j + 1)}, {j, 0, 6},
  Frame → True,
  BaseStyle → {Black, FontFamily → "Verdana", FontSize → 22},
  PlotStyle → {{Darker[Green], Thickness[0.005]},
    {Darker[Cyan], Thickness[0.005]}}, PlotRange → {{0, 6}, {-1, 7}},
  ImageSize → 700,
  FrameStyle → Thick,
  FrameLabel → {"Moment pędu j", "μ [μN"]},
  Epilog →
    {Text[Style["Protony", RGBColor[0.5, 0, 0.5], 26], {0.2, 5.5}, Left],
     Text[Style["j=1+1/2", Darker[Green], 22], {1.2, 4}, Left, {1, 0.35}],
     Text[Style["j=1-1/2", Darker[Cyan], 22], {3, 0.8}, Left, {1, 0.33}]}];
```

```
gExpProt = ListPlot[lisMagMomOddZ,
  PlotStyle → {RGBColor[0.5, 0, 0.5], PointSize[0.01]}];
```

Show[gSchProt, gExpProt]

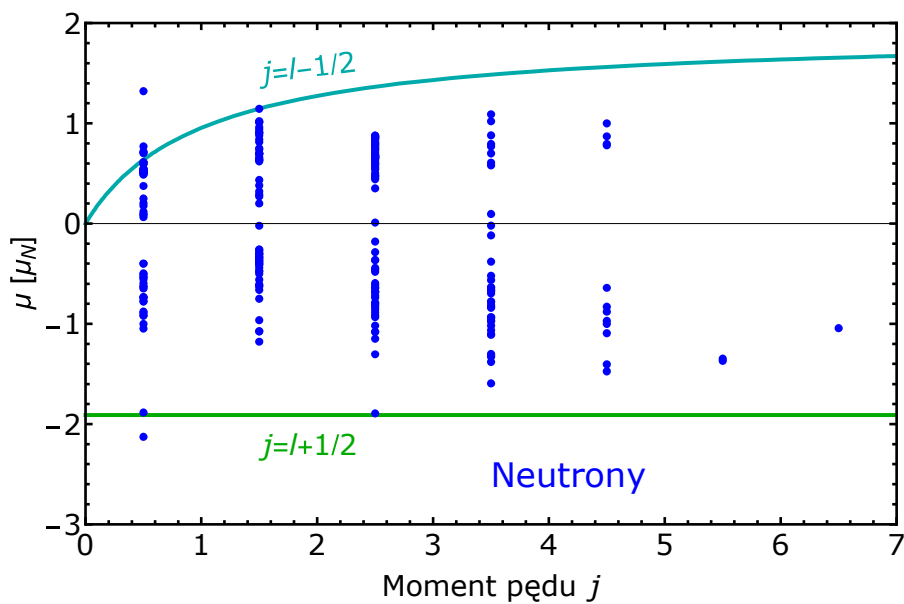


Dla neutronów :

```
gSchNeut = Plot[{-1.91, 1.91 * j / (j + 1)}, {j, 0, 7},
  Frame → True,
  BaseStyle → {Black, FontFamily → "Verdana", FontSize → 22},
  PlotStyle → {{Darker[Green], Thickness[0.005]},
    {Darker[Cyan], Thickness[0.005]}}, PlotRange → {{0, 7}, {-3, 2}},
  ImageSize → 700,
  FrameStyle → Thick,
  FrameLabel → {"Moment pędu  $j$ ", " $\mu$  [ $\mu_N$ ]"},
  Epilog → {Text[Style["Neutrony", Blue, 26], {3.5, -2.5}, Left],
    Text[Style[" $j=l+1/2$ ", Darker[Green], 22], {1.5, -2.2}, Left, {1, 0}],
    Text[Style[" $j=l-1/2$ ", Darker[Cyan], 22], {1.5, 1.5}, Left, {1, 0.12}]}];
```

```
gExpNeut = ListPlot[lisMagMomOddN,
  PlotStyle → {Blue, PointSize[0.01]}];
```

Show [gSchNeut, gExpNeut]



Kwadrupolowy moment elektryczny

Także kwadrupolowy moment elektryczny jest ważną obserwabłą stanów jądrowych. Charakteryzuje on odstępstwo rozkładu ładunku elektrycznego od symetrii sferycznej. Jest więc czułym miernikiem deformacji i zjawisk kolektywnych.

W prostym modelu jednocząstkowym, wartość spektroskopowego momentu kwadrupolowego (czyli mierzonego w LAB) wynosi (w barnach):

$$Q_s^{\text{SP}}(j) = -0.01 \frac{2j-1}{2j+2} \left(N_j + \frac{3}{2}\right) A^{1/3} \text{ b},$$

gdzie j jest spinem stanu, A liczbą masową jądra, a N_j główną liczbą kwantową oscylatora harmonicznego, którym przybliżamy stany jądrowe. Zakłada się tu, że przyczynek do Q pochodzi od jednego, nieparzystego protonu.

Wartości doświadczalne

Wartość elektrycznego momentu kwadrupolowego z bazy Wolframa:

```
eleMom[iso_] := Module[{a, q},
  a = IsotopeData[iso, "MassNumber"];
  q = IsotopeData[iso, "QuadrupoleMoment"];
  {a, q}
]
```

```
eleMom["2H"]
```

```
{2, 0.002860 b}
```

Tworzymy listy wszystkich wartości z tablic, znowu oddzielnie dla nieparzystych protonów i neutronów (nieparzysty neutron nie powinien dawać momentu elektrycznego!):

```
lisEleMomOddZ = Module[{loddZ},
  loddZ = Select[IsotopeData[], (EvenQ[IsotopeData[#, "NeutronNumber"]]) &&
    (OddQ[IsotopeData[#, "AtomicNumber"]]) &];
  Cases[
    eleMom[#] & /@ loddZ, Except[[_ , _Missing]]
  ];
];
```

```
lisEleMomOddN = Module[{loddN},
  loddN = Select[IsotopeData[], (OddQ[IsotopeData[#, "NeutronNumber"]]) &&
    (EvenQ[IsotopeData[#, "AtomicNumber"]]) &];
  Cases[
    eleMom[#] & /@ loddN, Except[[_ , _Missing]]
  ];
];
```

```
Length[lisEleMomOddZ]
```

```
134
```

```
lisEleMomOddZ
```

```
{ {7, -0.0406 b}, {9, 0.0278 b}, {11, 0.04065 b}, {13, 0.037 b}, {17, 0.058 b},
  {21, 0.05 b}, {23, 0.1006 b}, {25, -0.10 b}, {27, -0.06 b}, {29, 0.03 b},
  {27, 0.150 b}, {35, -0.08249 b}, {37, -0.06493 b}, {39, 0.049 b},
  {41, 0.060 b}, {43, -0.26 b}, {45, -0.22 b}, {47, -0.22 b}, {51, -0.052 b},
  {51, 0.42 b}, {55, 0.33 b}, {57, 0.52 b}, {59, 0.404 b}, {63, -0.211 b},
  {65, -0.195 b}, {67, 0.195 b}, {69, 0.168 b}, {71, 0.106 b}, {71, -0.021 b},
  {75, 0.314 b}, {79, 0.331 b}, {81, 0.276 b}, {77, 0.70 b}, {79, -0.098 b},
  {81, 0.398 b}, {83, 0.196 b}, {85, 0.23 b}, {87, 0.127 b}, {91, 0.154 b},
  {93, 0.18 b}, {95, 0.21 b}, {97, 0.581 b}, {93, -0.32 b}, {99, -0.129 b},
  {105, 0.83 b}, {107, 0.807 b}, {109, 0.841 b}, {111, 0.804 b}, {113, 0.799 b},
  {115, 0.86 b}, {117, 0.829 b}, {119, 0.854 b}, {121, 0.814 b}, {123, 0.757 b},
  {125, 0.710 b}, {127, 0.60 b}, {115, -0.36 b}, {117, -0.30 b}, {119, -0.37 b},
  {121, -0.36 b}, {123, -0.49 b}, {125, -0.889 b}, {127, -0.79 b},
```

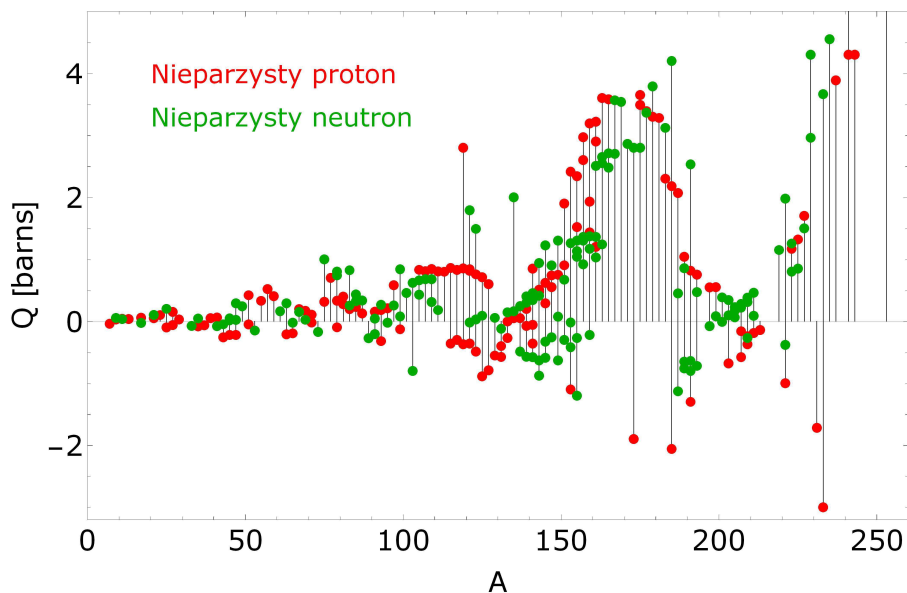
```
{129, -0.553 b}, {131, -0.40 b}, {133, -0.27 b}, {119, 2.8 b},
{121, 0.838 b}, {131, -0.575 b}, {133, -0.00371 b}, {135, 0.050 b},
{137, 0.051 b}, {139, -0.075 b}, {141, -0.36 b}, {143, 0.47 b},
{145, 0.62 b}, {137, 0.24 b}, {139, 0.20 b}, {141, -0.0589 b}, {147, 0.74 b},
{151, 1.9 b}, {141, 0.85 b}, {143, 0.51 b}, {145, 0.29 b}, {147, 0.55 b},
{149, 0.75 b}, {151, 0.903 b}, {153, 2.412 b}, {155, 2.34 b}, {157, 2.6 b},
{159, 1.432 b}, {161, 1.2 b}, {153, -1.1 b}, {155, 1.52 b}, {157, 2.97 b},
{159, 3.19 b}, {161, 3.22 b}, {163, 3.6 b}, {165, 3.58 b}, {159, 1.93 b},
{161, 2.90 b}, {175, 3.49 b}, {177, 3.39 b}, {179, 3.3 b}, {173, -1.9 b},
{175, 3.65 b}, {181, 3.28 b}, {183, 2.3 b}, {185, 2.18 b}, {187, 2.07 b},
{185, -2.06 b}, {189, 1.04 b}, {191, 0.816 b}, {193, 0.751 b}, {191, -1.3 b},
{197, 0.547 b}, {199, 0.55 b}, {203, -0.68 b}, {207, -0.58 b}, {209, -0.37 b},
{207, -0.16 b}, {209, -0.24 b}, {211, -0.19 b}, {213, -0.14 b},
{221, -1.00 b}, {223, 1.17 b}, {225, 1.32 b}, {227, 1.7 b}, {231, -1.72 b},
{233, -3.0 b}, {237, 3.886 b}, {241, 4.3 b}, {243, 4.30 b}, {253, 6.7 b}}
```

```
Length[lisEleMomOddN]
```

149

Wykreślamy wszystkie wartości w funkcji liczby masowej:

```
ListPlot[{lisEleMomOddZ, lisEleMomOddN},
  Frame -> True,
  FrameLabel -> {"A", "Q [barns]"},
  PlotRange -> {{-0.001, 260}, {-3.2, 5}},
  BaseStyle -> {FontFamily -> "Verdana", FontSize -> 22},
  AxesOrigin -> {0, -4},
  ImageSize -> 700,
  PlotStyle -> {{Red, PointSize[0.012]}, {Darker[Green], PointSize[0.012]}},
  Filling -> 0,
  FillingStyle -> Black,
  GridLines -> {{}, {0}},
  Epilog ->
  {Text[Style["Nieparzysty proton", Red, 20], {20, 4}, Left],
   Text[Style["Nieparzysty neutron", Darker[Green], 20], {20, 3.3}, Left]}
]
```



Jądra o nieparzystym N mają podobne wartości Q co jądra o nieparzystym Z !

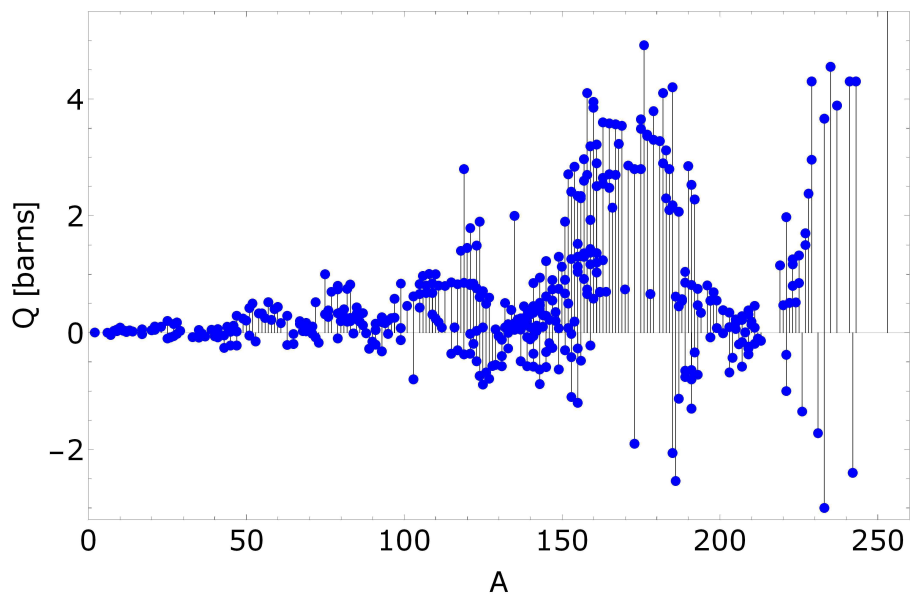
Wybieraliśmy tylko jądra parzysto-nieparzyste. Ale nieparzysto-nieparzyste też mogą mieć moment kwadrupolowy. Zobaczmy więc wszystkie wartości jakie są w bazie:

```
lisEleMom = Cases[
  {IsotopeData[#, "AtomicMass"], IsotopeData[#, "QuadrupoleMoment"]} & /@
  IsotopeData[], Except[[_ , _Missing]]];
```

```
Length[lisEleMom]
```

406

```
ListPlot[lisEleMom,
  Frame -> True,
  FrameLabel -> {"A", "Q [barns]"},
  PlotRange -> {{-0.001, 260}, {-3.2, 5.5}},
  BaseStyle -> {FontFamily -> "Verdana", FontSize -> 22},
  AxesOrigin -> {0, -4},
  ImageSize -> 700,
  PlotStyle -> {Blue, PointSize[0.012]},
  Filling -> 0,
  FillingStyle -> Black,
  GridLines -> {{}, {0}}
]
```



Stany wzbudzone

Wśród informacji w IsotopeData są też dane o stanach wzbudzonych - ich energie, spiny, parzystości i czasy życia (szerokości). Można więc łatwo sporządzać różne systematyki.

Pierwszy stan 2⁺

Chcemy zobaczyć energie wzbudzenia pierwszego stanu 2⁺ w jądrach parzysto-parzystych. Zrobimy to oddzielnie dla każdej liczby atomowej.

Funkcja znajdująca energię pierwszego stanu 2⁺ dla danego nuklidu:

```
fExt2plus[iso_] := Module[{exst, spi, par, spipar, l2, i},
  exst = IsotopeData[iso, "ExcitedStateEnergies"];
  If[Length[exst] > 0,
    spi = IsotopeData[iso, "ExcitedStateSpins"];
    par = IsotopeData[iso, "ExcitedStateParities"];
    spipar = Transpose[{spi, par}];
    l2 = Position[spipar, {2, 1}];
    If[Length[l2] > 0, i = First[l2][[1]]; exst[[i], 0],
    0]
]
```

```
fExt2plus["182Hg"]
```

351.8 keV

Funkcja tworząca listę parzystych izotopów dla danej liczby atomowej:

```
fevA[z_] := Select[IsotopeData[], (IsotopeData[#, "AtomicNumber"] == z) &&
  (EvenQ[IsotopeData[#, "MassNumber"]]) &]
```

```
fevA[50]
```

```
{tin-100, tin-102, tin-104, tin-106, tin-108,
tin-110, tin-112, tin-114, tin-116, tin-118, tin-120, tin-122,
tin-124, tin-126, tin-128, tin-130, tin-132, tin-134, tin-136}
```

Lista par {liczba neutronów, energia pierwszego stanu 2⁺ w MeV} dla izotopów danego pierwiastka:

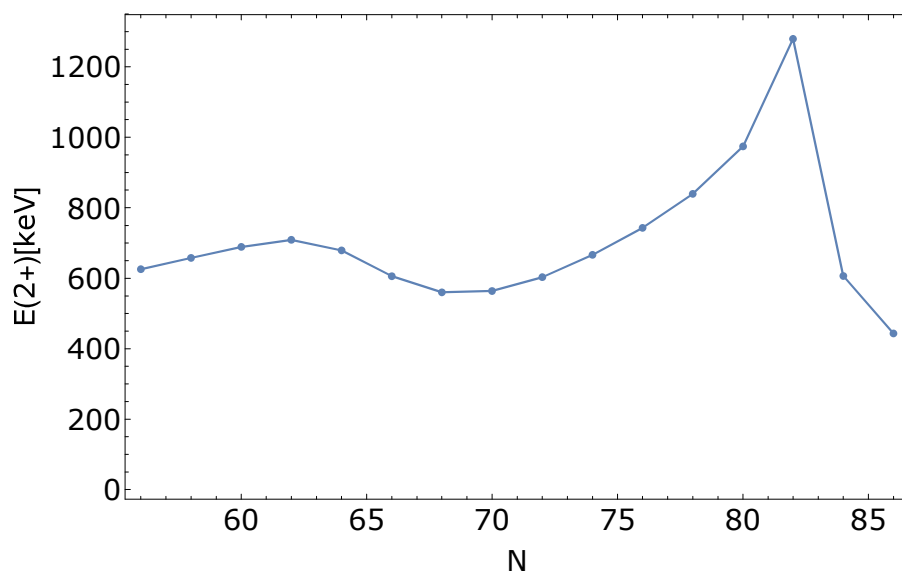
```
fExt2plusZ[z_] := Cases[
  {IsotopeData[#, "NeutronNumber"], fExt2plus[#]} & /@ fevA[z],
  Except[{_ , 0}]]
```

Na przykład pierwsze stany 2⁺ dla izotopów niklu:

```
t2plusNi = fExt2plusZ[52]
```

```
{{56, 625.20 keV}, {58, 657.70 keV}, {60, 689.01 keV}, {62, 708.74 keV},
{64, 678.92 keV}, {66, 605.706 keV}, {68, 560.438 keV}, {70, 564.117 keV},
{72, 602.731 keV}, {74, 666.352 keV}, {76, 743.219 keV}, {78, 839.494 keV},
{80, 974.22 keV}, {82, 1279.11 keV}, {84, 606.64 keV}, {86, 443.1 keV}}
```

```
ListPlot[t2plusNi,  
Joined -> True,  
Mesh -> Full,  
Frame -> True,  
FrameLabel -> {"N", "E(2+) [keV]"},  
PlotRange -> All,  
BaseStyle -> {FontFamily -> "Verdana", FontSize -> 22},  
PlotStyle -> Automatic,  
Filling -> False,  
ImageSize -> 700  
]
```

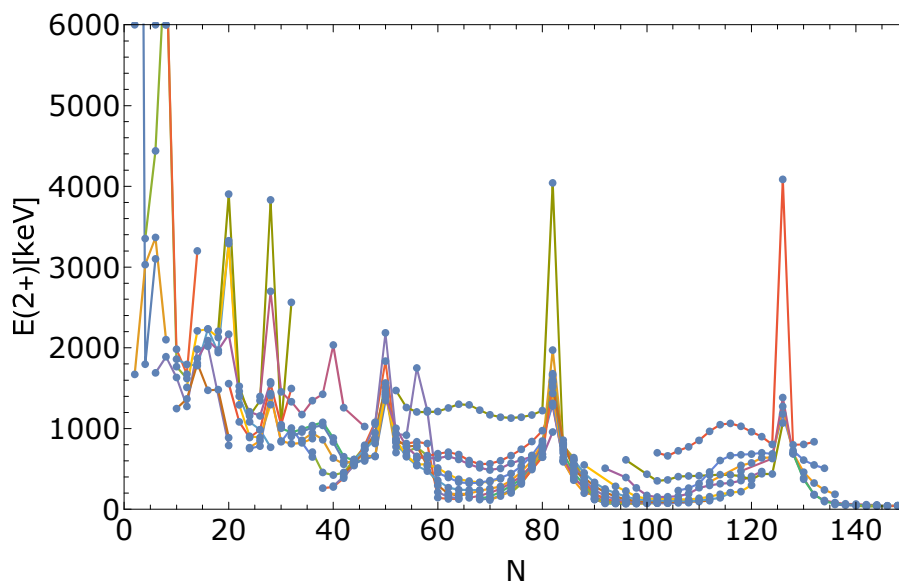


Porównanie energii 2^+ momentem kwadrupolowym

Teraz tworzymy tabelę zawierającą takie listy dla wszystkich parzystych pierwiastków od helu do rutefordu:

```
tab = Table[fExt2plusZ[z], {z, 2, 104, 2}];

g2plus = ListPlot[tab,
  Joined -> True,
  InterpolationOrder -> 1,
  Mesh -> Full,
  Frame -> True,
  FrameLabel -> {"N", "E(2+) [keV]"},
  PlotRange -> {{0, 150}, {0, 6000}},
  BaseStyle -> {FontFamily -> "Verdana", FontSize -> 22},
  PlotStyle -> Automatic,
  Filling -> False,
  ImageSize -> 700
]
```



Możemy teraz porównać systematykę stanów 2^+ i elektrycznego momentu kwadrupolowego.

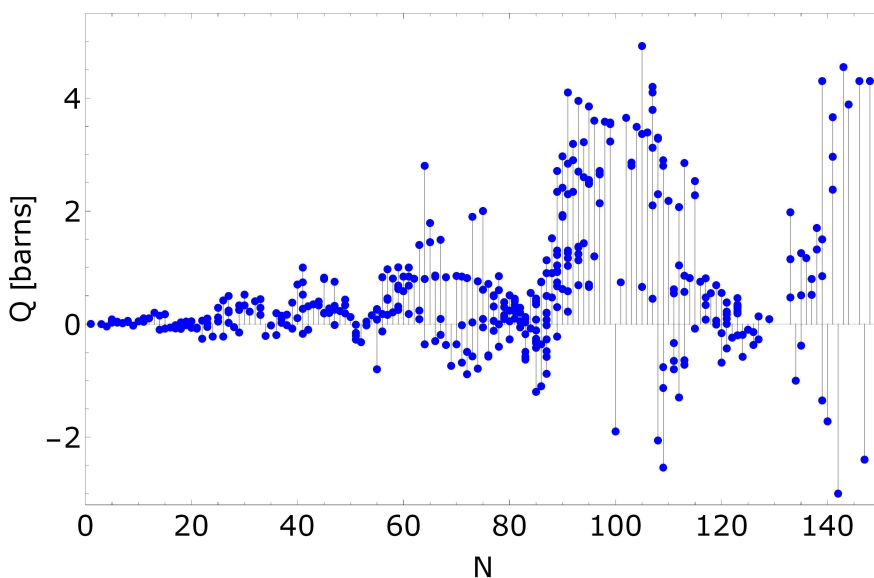
Tylko musimy jeszcze zrobić rysunek wartości Q , ale w funkcji N .

```
lisEleMom2 = Cases [
  {IsotopeData[#, "NeutronNumber"], IsotopeData[#, "QuadrupoleMoment"]} & /@
  IsotopeData[], Except[[_ , _Missing]]];
```

```

gQ = ListPlot[lisEleMom2,
  Frame -> True,
  FrameLabel -> {"N", "Q [barns]"},
  PlotRange -> {{-0.001, 150}, {-3.2, 5.5}},
  BaseStyle -> {FontFamily -> "Verdana", FontSize -> 22},
  AxesOrigin -> {0, -4},
  ImageSize -> 680,
  PlotStyle -> {Blue, PointSize[0.01]},
  Filling -> 0,
  FillingStyle -> Gray,
  GridLines -> {{}, {0}}
]

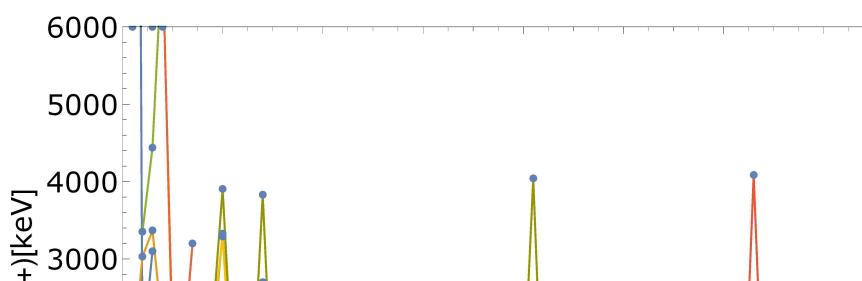
```

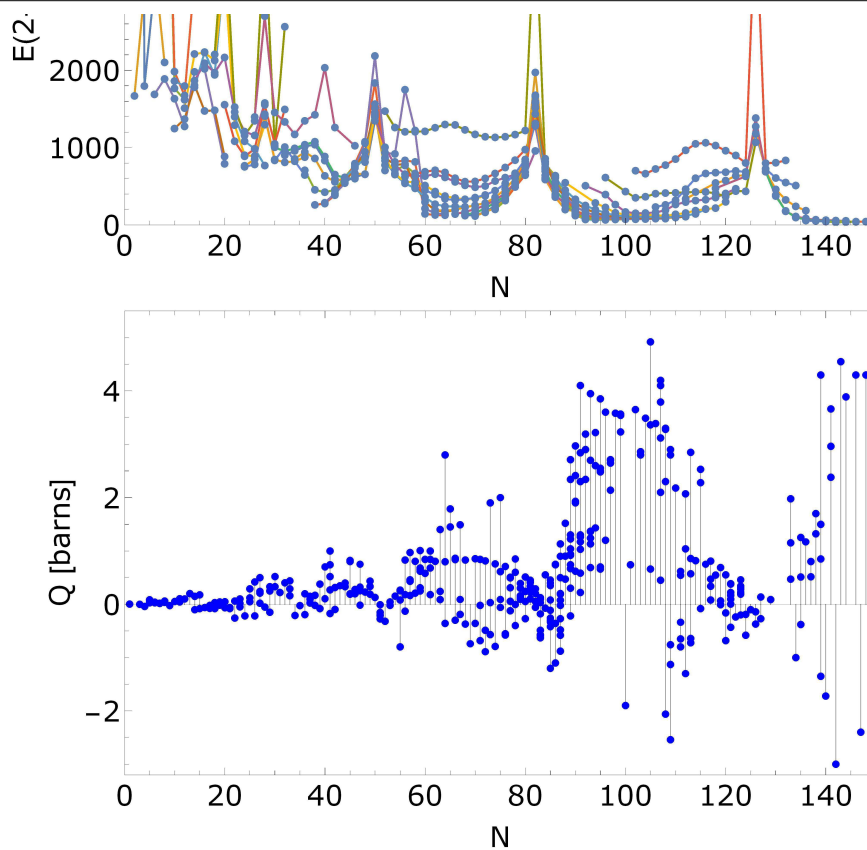


```

GraphicsColumn[{g2plus, gQ}, Spacings -> 0,
  Alignment -> Right, Frame -> All, FrameStyle -> White]

```





Stosunek $R_{4/2}$

Często używanym parametrem w badaniu struktury jądra jest stosunek energii pierwszego stanu 4^+ i stanu 2^+ , $R_{4/2}$. W przypadku widma rotacyjnego stosunek ten wynosi 3.33, zaś dla widma wibracyjnego jest równy 2.

```
fR4to2ratio[iso_] := Module[{exst, spi, par, spipar, l2, l4, i, j},
  exst = IsotopeData[iso, "ExcitedStateEnergies"];
  If[Length[exst] > 0,
    spi = IsotopeData[iso, "ExcitedStateSpins"];
    par = IsotopeData[iso, "ExcitedStateParities"];
    spipar = Transpose[{spi, par}];
    l2 = Position[spipar, {2, 1}];
    l4 = Position[spipar, {4, 1}];
    If[(Length[l2] > 0) && (Length[l4] > 0),
      i = First[l2[[1]]];
      j = First[l4[[1]]];
      exst[[j]] / exst[[i]],
      0],
    0]
]
```

```
fR4to2ratio["110Cd"]
```

2.344994

Tworzymy listę wartości $R_{4/2}$ dla izotopów wybranego pierwiastka, które numerujemy liczbą neutronów:

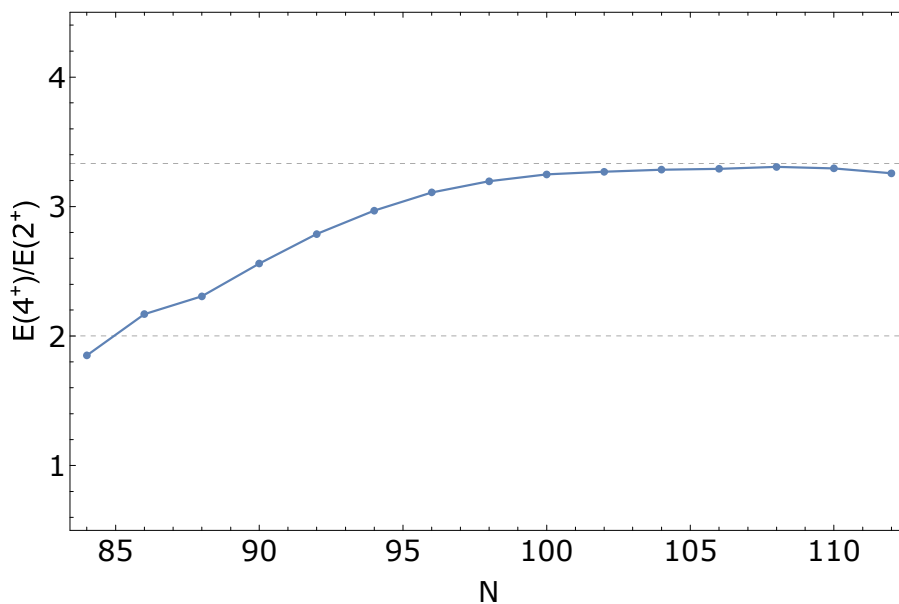
```
fR4to2Z[z_] := Cases[
  {IsotopeData[#, "NeutronNumber"], fR4to2ratio[#]} & /@ fevA[z],
  Except[[_ , 0]]]
```

Na przykład dla hafnu ($Z=72$):

```
tR4to2 = fR4to2Z[72]
```

```
{{84, 1.85}, {86, 2.1692}, {88, 2.3068}, {90, 2.560}, {92, 2.786},
{94, 2.967}, {96, 3.110}, {98, 3.1943}, {100, 3.248}, {102, 3.2685},
{104, 3.2845}, {106, 3.2906}, {108, 3.30652}, {110, 3.295}, {112, 3.257}}
```

```
ListPlot[tR4to2,
  Joined -> True,
  Mesh -> Full,
  Frame -> True,
  FrameLabel -> {"N", "E(4+)/E(2+)"},
  PlotRange -> {All, {0.5, 4.5}},
  GridLines -> {{}, {2, 3.33}},
  GridLinesStyle -> Directive[Gray, Dashed],
  BaseStyle -> {FontFamily -> "Verdana", FontSize -> 22},
  PlotStyle -> Automatic,
  Filling -> False,
  ImageSize -> 700
]
```

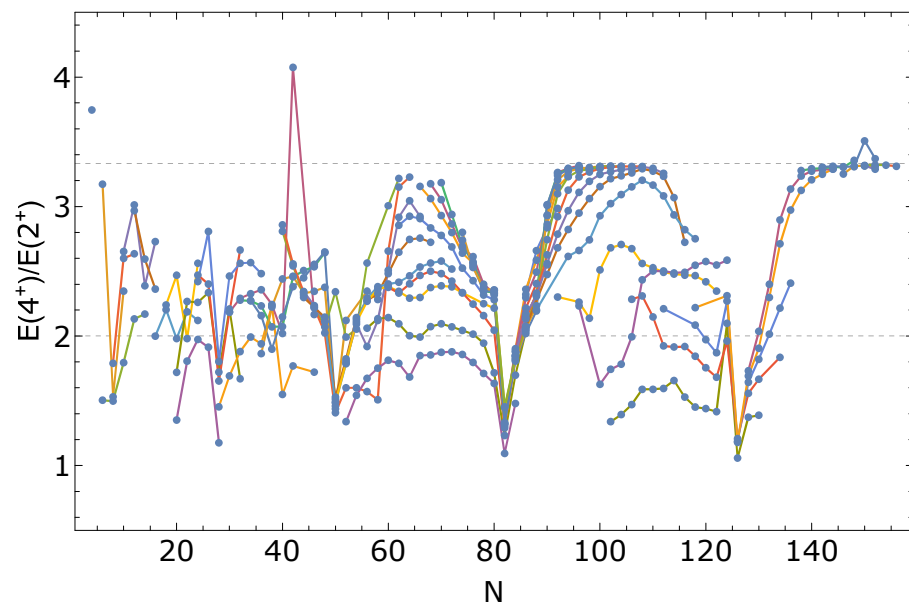


Jak poprzednio tworzymy tabelę z wartościami $R_{4/2}$ dla wszystkich parzystych pierwiastków od helu do rutherfordu:

```
tab = Table[fR4to2Z[z], {z, 2, 104, 2}];
```

```
tab2 = Cases[tab, Except[{}]];
```

```
ListPlot[tab2,
  Joined -> True,
  Mesh -> Full,
  Frame -> True,
  FrameLabel -> {"N", "E(4+)/E(2)"},
  PlotRange -> {All, {0.5, 4.5}},
  GridLines -> {{}, {2, 3.33}},
  GridLinesStyle -> Directive[Gray, Dashed],
  BaseStyle -> {FontFamily -> "Verdana", FontSize -> 22},
  PlotStyle -> Automatic,
  Filling -> False,
  ImageSize -> 700
]
```



Model kroplowy

Porównujemy przewidywania modelu Bethego - Weizsäckera z doświadczeniem, korzystając z **IsotopeData**

Formuła Bethego - Weizsäckera

Ten bardzo prosty, fenomenologiczny model jądra opiera się na obserwacji, że energia wiązania jądra jest w przybliżeniu proporcjonalna do liczby nukleonów. Nasuwa to analogię z kroplą cieczy, co z kolei wskazuje na rolę powierzchni. Trzeba też wziąć pod uwagę, że "ciecz" jądrowa jest naładowana elektrycznie, i że składa się z dwóch rodzajów fermionów. Pozwala to określić masę (energię wiązania) jądra.

Pierwszą wersję tego modelu podał w 1935 roku C. F. v. Weizsäcker. Później ulepszony i do dziś bardzo popularny, nosi nazwę modelu Bethego - Weizsäckera.

Masa nuklidu (obojętnego atomu) o liczbie masowej A i liczbie atomowej Z :

$$M(A, Z) = Z M_H + (A - Z) m_n + B(A, Z) / c^2,$$

gdzie M_H to masa atomu wodoru, m_n to masa neutronu, a $B(A, Z)$ to energia wiązania nuklidu.

$$M_H = 938.783 \text{ MeV} / c^2, \quad m_n = 939.565 \text{ MeV} / c^2$$

Formuła **Bethego - Weizsäckera** na energię wiązania:

$$B(A, Z) = b_V A - b_S A^{2/3} - b_C \frac{Z(Z-1)}{A^{1/3}} - b_{sym} \frac{(A-2Z)^2}{A},$$

Parametryzacja

Często spotykany zestaw parametrów:

```
bVBW = 15.8; (* współczynnik objętościowy [MeV] *)  
bSBW = 18.3; (* współczynnik powierzchniowy [MeV] *)  
bCBW = 0.714; (* współczynnik Coulombowski [MeV] *)  
bsymBW = 23.2; (* współczynnik symetrii [MeV] *)
```

Formuła na energię wiązania:

```
beBW[A_, Z_] :=  
bVBW * A - bSBW * A^(2/3) - bCBW * Z * (Z - 1) / A^(1/3) - bsymBW * (A - 2 * Z)^2 / A
```

I energia wiązania na jeden nukleon:

```
beBWperA[A_, Z_] := beBW[A, Z] / A
```

```
beBWperA[4, 2]
```

4.04683

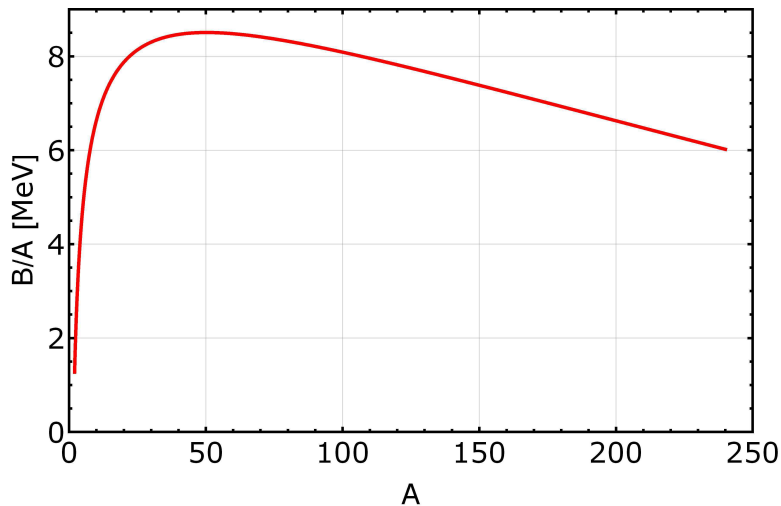
```
beBWperA[58, 28]
```

8.64054

Energia wiązania na nukleon

Energia wiązania na nukleon dla $N = Z$:

```
Plot[beBWperA[x, x / 2], {x, 2, 240},
  Frame → True,
  FrameLabel → {"A", "B/A [MeV]"},
  FrameStyle → Thick,
  ImageSize → 600,
  BaseStyle → {FontFamily → "Verdana", FontSize → 22},
  AxesOrigin → {0, 0},
  GridLines → Automatic,
  PlotStyle → {{Thickness[0.005], Red}},
  PlotRange → {{0, 250}, {0, 9}}
```

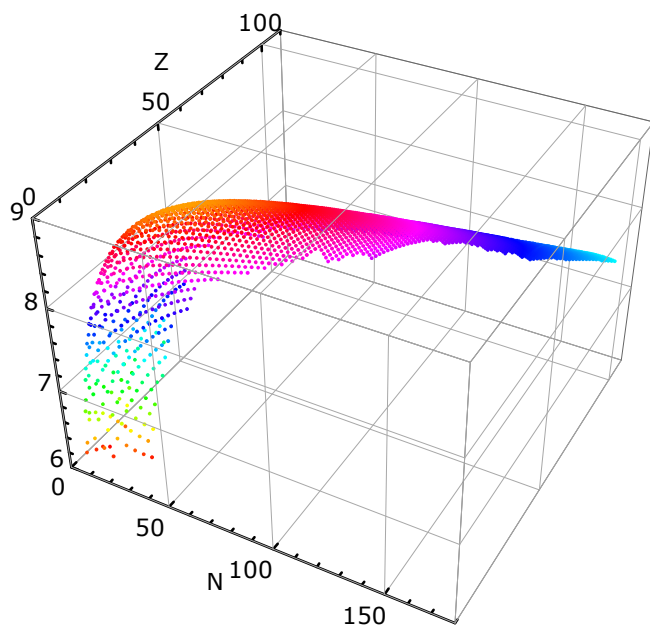


Obraz ogólny - obliczamy energię wiązania dla tych izotopów, które są w bazie **Isotope-**

Data :

```
gbBW = Table[{a - z, z, beBWperA[a, z]}, {z, 2, 110},
  {a, IsotopeData[#, "MassNumber"] & /@ IsotopeData[z]}] // Flatten[#, 1] &;
```

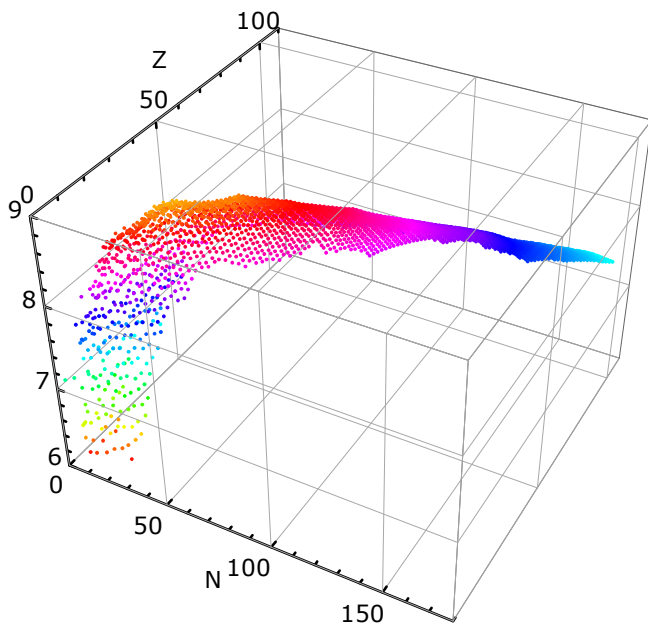
```
pBW = ListPointPlot3D[gbBW, BoxRatios → {1, 1, 0.7},
  BaseStyle → {FontFamily → "Verdana", FontSize → 18},
  AxesLabel → {"N", "Z", ""},
  FaceGrids → All,
  PlotRange → {{0, 180}, {0, 110}, {6, 9}},
  ImageSize → 500,
  AxesStyle → Thick,
  ColorFunction → Function[{x, y, z}, Hue[(z - 6) / 2.5]],
  ColorFunctionScaling → False
]
```



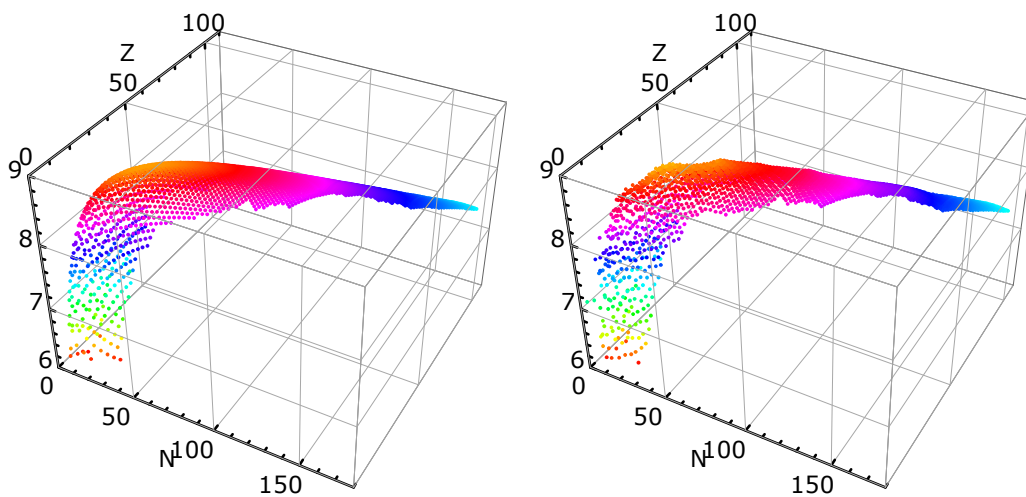
Porównajmy to z wartościami doświadczalnymi

```
gexp = Table[{a - z, z, IsotopeData[{z, a}, "BindingEnergy"]}, {z, 2, 110},
  {a, IsotopeData[#, "MassNumber"] & /@ IsotopeData[z]}] // Flatten[#, 1] &;
```

```
pexp = ListPointPlot3D[gexp, BoxRatios -> {1, 1, 0.7},
  BaseStyle -> {FontFamily -> "Verdana", FontSize -> 18},
  AxesLabel -> {"N", "Z", ""},
  FaceGrids -> All,
  PlotRange -> {{0, 180}, {0, 110}, {6, 9}},
  ImageSize -> 500,
  AxesStyle -> Thick,
  ColorFunction -> Function[{x, y, z}, Hue[(z - 6) / 2.5]],
  ColorFunctionScaling -> False
]
```



```
Show[GraphicsRow[{pBW, pexp}], ImageSize -> 800]
```



Masy nuklidów stabilnych

Wyberzmy najpierw wszystkie nuklidy stabilne, wliczając te bardzo długożyciowe.

```
selectStable = Select[IsotopeData[], (IsotopeData[#, "Stable"] ||
  IsotopeData[#, "HalfLife"] > Quantity[2 * 10^(16), "Seconds"]) &];
```

```
Length[selectStable]
```

283

Lista {N, Z} wszystkich nuklidów stabilnych - w tej formie przyda się też później.

```
stableNZ = {IsotopeData[#, "NeutronNumber"],
  IsotopeData[#, "AtomicNumber"]} & /@ selectStable;
```

Aby jeszcze trochę zawęzić liczbę nuklidów, wybierzmy dla każdego pierwiastka izotop najbardziej rozpowszechniony.

Najpierw funkcja wybierająca najczęściej występujący izotop, a dokładniej liczbę masową tego izotopu :

```
mostAbundIzo[z_] := Module[{iso, abu, maxabu, pos},
  iso = IsotopeData[z];
  abu = IsotopeData[#, "IsotopeAbundance"] & /@ iso;
  maxabu = Max[abu];
  pos = Flatten[Position[abu, maxabu]];
  IsotopeData[iso[[pos]][[1]], "MassNumber"]
]
```

```
mostAbundIzo[92]
```

238

Tworzymy listę liczb atomowych nuklidów stabilnych, bez powtórzeń:

```
stableZ = Union[#[[2]] & /@ stableNZ];
```

```
selectMostAbund = {mostAbundIzo[#, #] & /@ stableZ};
```

Lista wartości zmierzonych $\{A, B_{exp}/A\}$ dla tych nuklidów :

```
stabABperAExp = {#[[1]],
  IsotopeData[{#[[2]], #[[1]]}, "BindingEnergy"]} & /@ selectMostAbund;
```

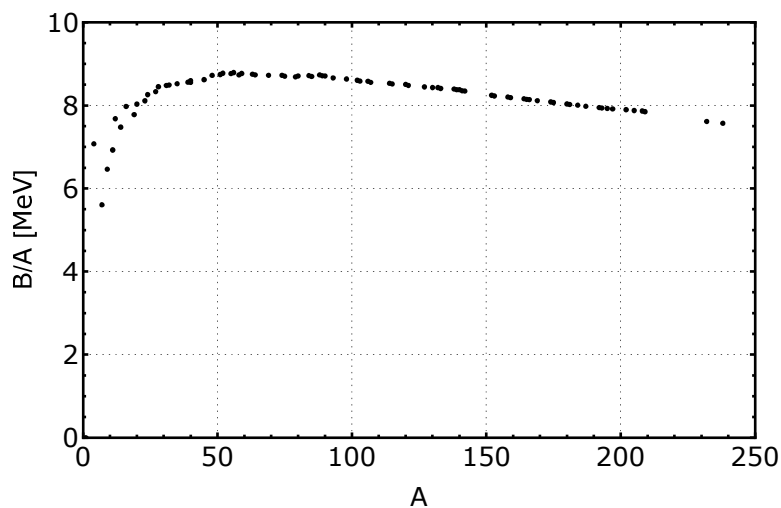
Przwydywanie formuły B-W dla tych nuklidów:

```
stabABperABW = {#[[1]], beBWperA[#[[1]], #[[2]]]} & /@ selectMostAbund;
```

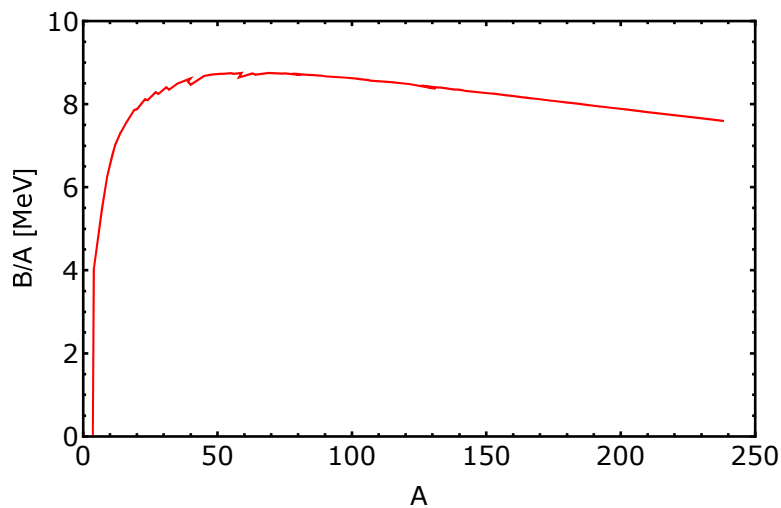
Energia wiązania nuklidów stabilnych

Rysujemy energię wiązania na nukleon dla nuklidów stabilnych.

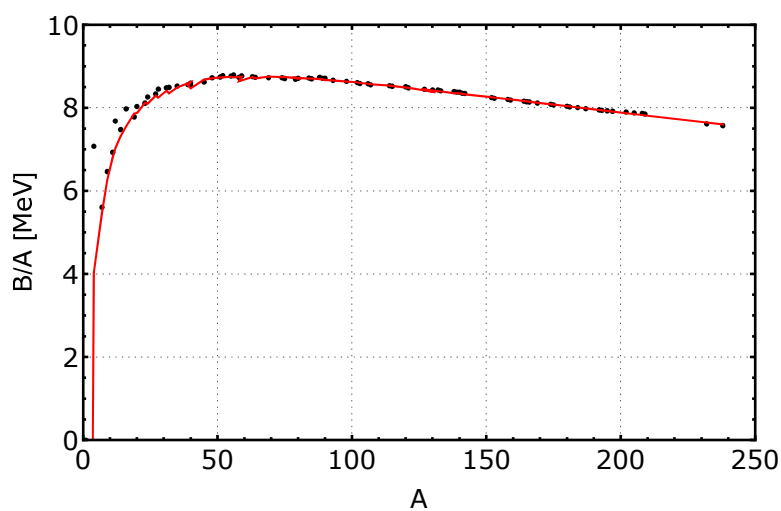
```
gstabBperAExp = ListPlot[stabABperAExp,  
  Frame → True,  
  BaseStyle → {FontFamily → "Verdana", FontSize → 20},  
  FrameLabel → {"A", "B/A [MeV]"},  
  PlotRange → {{0, 250}, {0, 10}},  
  PlotStyle → {PointSize[0.008], Black},  
  FrameStyle → Thick,  
  GridLines → Automatic,  
  GridLinesStyle → Directive[Dotted],  
  ImageSize → 600  
]
```



```
gstabBperABW = ListPlot[stabABperABW,  
  Frame → True,  
  BaseStyle → {FontFamily → "Verdana", FontSize → 20},  
  FrameLabel → {"A", "B/A [MeV]},  
  PlotRange → {{0, 250}, {0, 10}},  
  PlotStyle → {PointSize[0.01], Red},  
  FrameStyle → Thick,  
  ImageSize → 600,  
  Joined → True]
```



```
Show[{gstabBperAExp, gstabBperABW}]
```



Ścieżka stabilności

Model kropłowy pozwala przewidzieć ścieżkę trwałości beta. Dla każdej liczby masowej A trzeba po prostu znaleźć izobar o najmniejszej masie.

Pełna masa nuklidu:

```
mBW[A_, Z_] := Z * 938.783 + (A - Z) * 939.565 - beBW[A, Z]
```

Szukamy minimum tej funkcji względem Z dla ustalonego A . Obliczamy pochodną masy względem Z i badamy kiedy jest ona równa zeru:

```
dmdZ = D[mBW[A, Z], Z]
```

$$-0.782 - \frac{92.8 (A - 2Z)}{A} + \frac{0.714 (-1 + Z)}{A^{1/3}} + \frac{0.714 Z}{A^{1/3}}$$

```
stZ = Solve[dmdZ == 0, Z][[1]]
```

$$\left\{ Z \rightarrow \frac{93.582 + \frac{0.714}{A^{1/3}}}{\frac{185.6}{A} + \frac{1.428}{A^{1/3}}} \right\}$$

```
stableZBW[A_] := \frac{93.58200000000004 + \frac{0.714}{A^{1/3}}}{\frac{185.6}{A} + \frac{1.428}{A^{1/3}}}
```

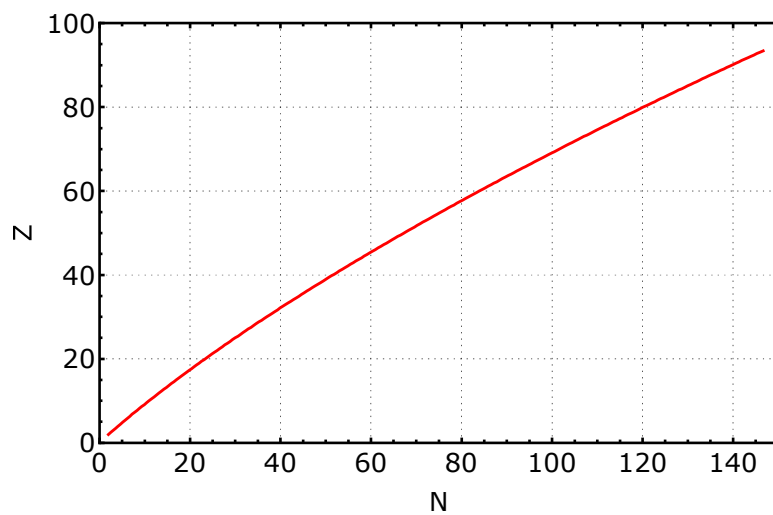
```
stableZBW[208]
```

82.6796

Możemy teraz narysować przewidywaną ścieżkę trwałości na mapie nuklidów i porównać z doświadczeniem!

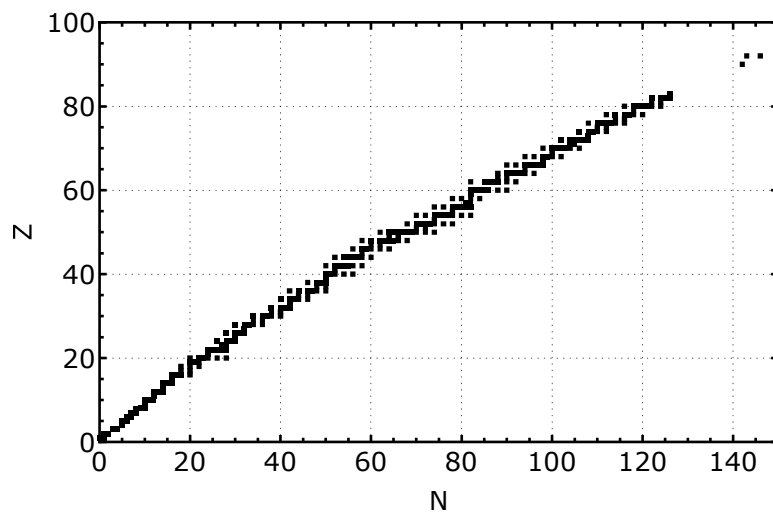
```
tabBW = Table[{a - stableZBW[a], stableZBW[a]}, {a, 4, 240, 1}];
```

```
plotBW = ListPlot[tabBW,
  Frame -> True,
  Joined -> True,
  FrameLabel -> {"N", "Z"},
  BaseStyle -> {FontFamily -> "Verdana", FontSize -> 20},
  AxesOrigin -> {0, 0},
  PlotStyle -> {{Thickness[0.004], Hue[1]}},
  PlotRange -> {{0, 150}, {0, 100}},
  FrameStyle -> Thick,
  GridLines -> Automatic,
  GridLinesStyle -> Directive[Dotted],
  ImageSize -> 600]
```



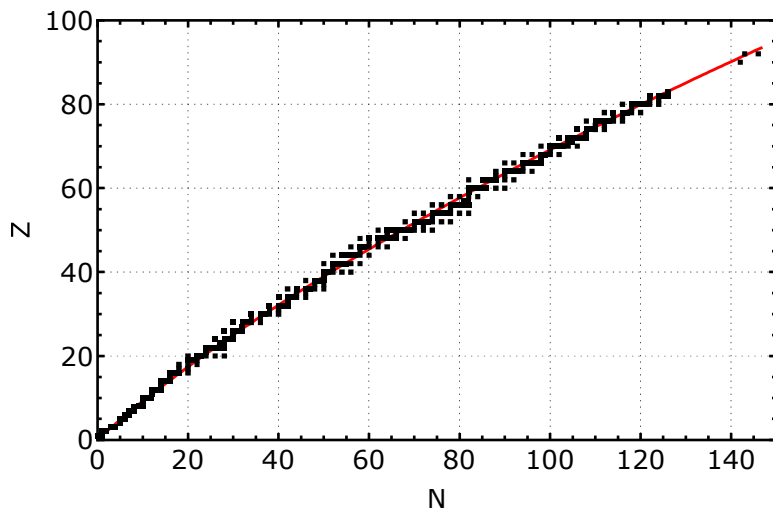
Mamy już listę $\{N,Z\}$ nuklidów stabilnych, zobaczmy je na mapie:

```
plotchart = ListPlot[stableNZ,
  Frame → True,
  FrameLabel → {"N", "Z"},
  BaseStyle → {FontFamily → "Verdana", FontSize → 20},
  AxesOrigin → {0, 0},
  PlotMarkers → {Graphics[{Black, Rectangle[{0, 1}]}], 0.013},
  PlotRange → {{0, 150}, {0, 100}},
  FrameStyle → Thick,
  GridLines → Automatic,
  GridLinesStyle → Directive[Dotted],
  ImageSize → 600]
```



A teraz porównajmy oba wykresy:

```
Show[plotBW, plotchart, ImageSize -> 600]
```



Manipulacja: analiza parametrów modelu

Zobaczymy jeszcze jak poszczególne parametry modelu B-W wpływają na energię wiązania na nukleon. Dla każdej liczby masowej wybieramy nuklid stabilny wg. modelu kropłowego i porównujemy jego zmierzoną energię wiązania z przewidywaniem.

```
Manipulate[
  quality = 0;
  BWdata = Table[{a, be[a, Round[stabZ[a]]]}, {a, 5, 239, 2}];
  (* Do[quality=quality+(expdata[[i,2]]-BWdata[[i,2]])^2,{i,1,118,1}];*)
  quality = Total[(expdata[[All, 2]] - BWdata[[All, 2]])^2];
  qq = Round[quality / 2.49006, 0.001];
  gexp = ListPlot[expdata, PlotStyle -> {PointSize[0.008], Black}];
  gVol = Plot[bV, {x, 1, 240},
    PlotStyle -> {{Thickness[0.005], Hue[0.55]}}];
  gVolS = Plot[bV - bS / x^(1/3), {x, 1, 240},
    PlotStyle -> {{Thickness[0.005], Hue[0.7]}}];
  gVolSC =
    Plot[bV - bS / x^(1/3) - bC * stabZ[x] * (stabZ[x] - 1) / x^(4/3), {x, 1, 240},
      PlotStyle -> {{Thickness[0.005], Hue[0.8]}}];
  gfull = Plot[be[x, stabZ[x]], {x, 1, 240},
    PlotStyle -> {{Thickness[0.006], Hue[1]}}];
  Show[gVol, gVolS, gVolSC, gfull, gexp,
    Frame -> True,
    AxesOrigin -> {0, 0},
```

```

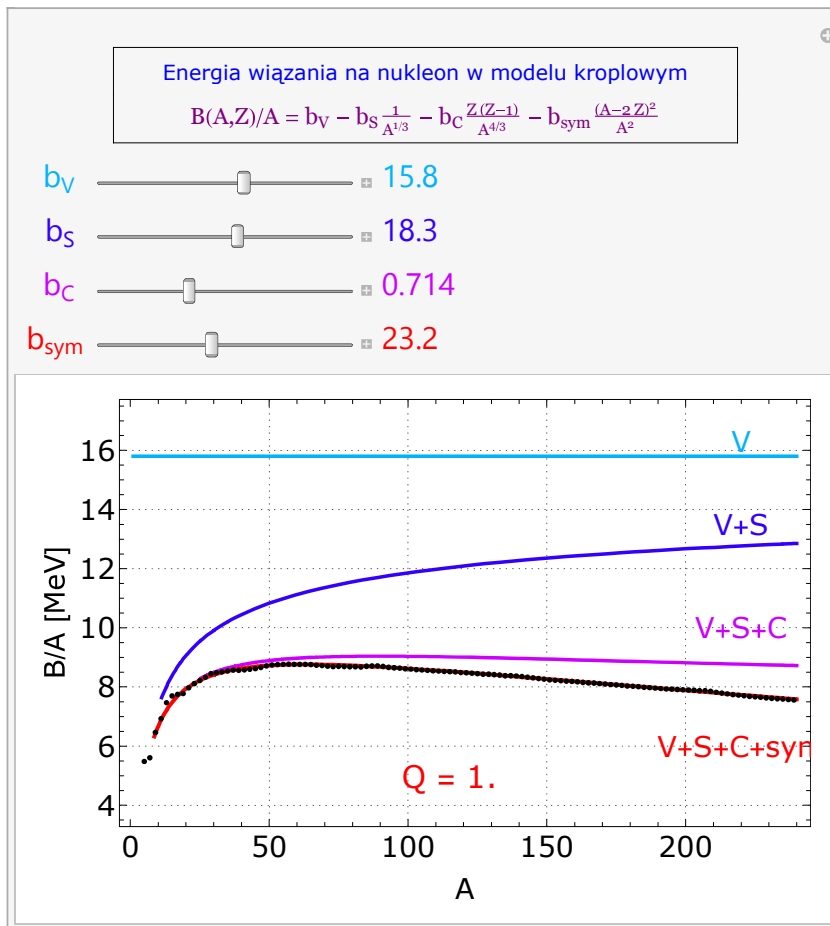
PlotRange -> {{1, 240}, {4, 17}},
GridLines -> Automatic,
GridLinesStyle -> Directive[Dotted],
FrameLabel -> {"A", "B/A [MeV]"},
BaseStyle -> {FontFamily -> "Verdana", FontSize -> 22},
ImageSize -> 600,
Epilog -> {Text[Style[StringForm["Q = `1`", qq], Red, 24], {115, 5}],
  Text[Style["V", Hue[0.55], 22], {220, 16.3}],
  Text[Style["V+S", Hue[0.7], 22], {220, 13.5}],
  Text[Style["V+S+C", Hue[0.8], 22], {220, 10}],
  Text[Style["V+S+C+sym", Hue[1], 22], {220, 6}]
}],
{{bV, 15.8, Style["bv ", Hue[0.55], 24]}, 10, 20, 0.1,
  Appearance -> "Labeled", LabelStyle -> Directive[Hue[0.55], 24]},
{{bS, 18.3, Style["bs ", Hue[0.7], 24]}, 10, 25, 0.1,
  Appearance -> "Labeled", LabelStyle -> Directive[Hue[0.7], 24]},
{{bC, 0.714, Style["bc ", Hue[0.8], 24]}, 0.3,
  1.5, 0.001, Appearance -> "Labeled",
  LabelStyle -> Directive[Hue[0.8], 24]},
{{bsym, 23.2, Style["bsym", Hue[1.0], 24]},
  10, 40, 0.1, Appearance -> "Labeled",
  LabelStyle -> Directive[Hue[1], 24]},
TrackedSymbols -> {bV, bS, bC, bsym},
FrameLabel -> {None, None, fulltit, None},
SaveDefinitions -> True,
Initialization -> {

```

```

be[A_, Z_] :=
  (bV * A - bS * A^(2/3) - bC * Z * (Z - 1) / A^(1/3) - bsym * (A - 2 * Z)^2 / A) / A;
stabZ[A_] := 
$$\frac{93.58200000000004 + \frac{0.714}{A^{1/3}}}{\frac{185.6}{A} + \frac{1.428}{A^{1/3}}}$$
;
expdata = Table[{a, QuantityMagnitude[
  IsotopeData[{Round[stabZ[a]], a}, "BindingEnergy"]]}, {a, 5, 239, 2}];
titleA = Text[Style["Energia wiązania na nukleon w modelu kropowym",
  Blue, FontFamily -> "Verdana", 14]];
titleB = Text[Style["B(A, Z) / A = bv - bs  $\frac{1}{A^{1/3}}$  - bc  $\frac{Z(Z-1)}{A^{4/3}}$  - bsym  $\frac{(A-2Z)^2}{A^2}$ ",
  FontFamily -> "Georgia", Purple, 18]];
fulltit = GraphicsColumn[{titleA, titleB}, AspectRatio -> 0.15,
  ImageSize -> 500, Frame -> True]
]

```



Masy wszystkich nuklidów

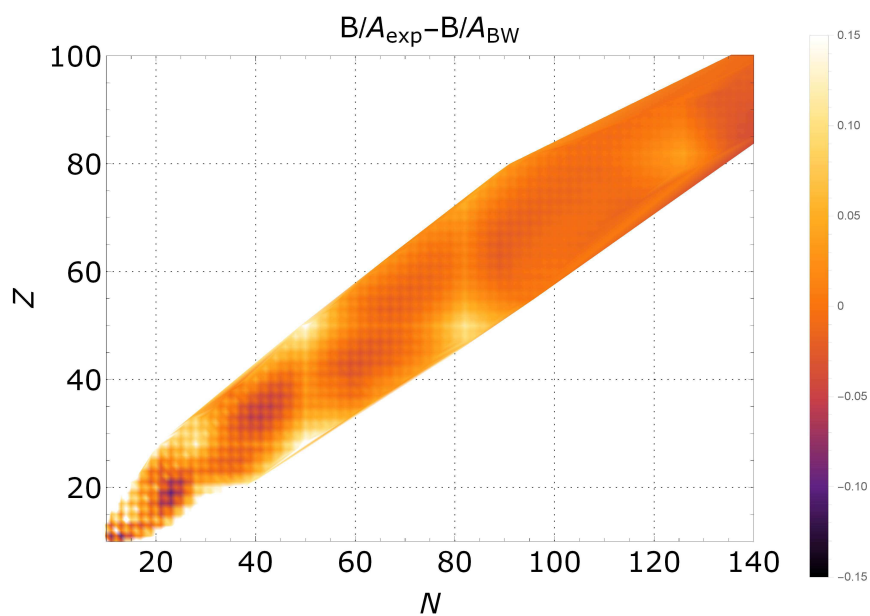
Spoglądamy teraz na całą mapę nuklidów. Żeby sprawdzić model kroplowy obliczamy różnicę między zmierzoną energią wiązania na nukleon a przewidywaniem modelu B-W dla wszystkich nuklidów:

```
diff = Table[{a - z, z,
  QuantityMagnitude[IsotopeData[{z, a}, "BindingEnergy"] - beBWperA[a, z]},
  {z, 1, 110}, {a, IsotopeData[#, "MassNumber"] & /@ IsotopeData[z]}];
```

```
diff[[1]]
```

```
{{0, 1, 25.7}, {1, 1, -0.162997}, {2, 1, 2.29356},
  {3, 1, 2.92863}, {4, 1, 4.59026}, {5, 1, 5.54562}, {6, 1, 6.54331}}
```

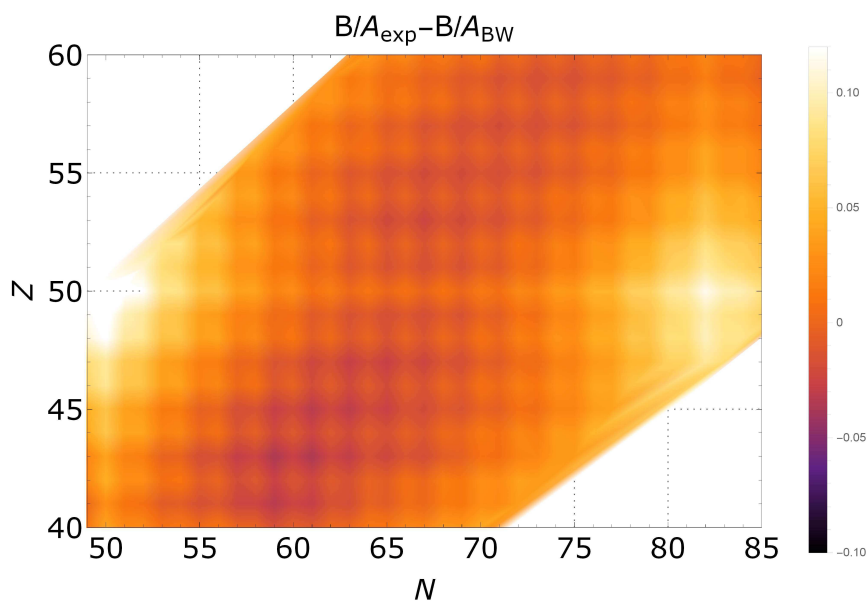
```
gdiff = ListDensityPlot[Flatten[diff, 1],  
  PlotRange -> {{10, 140}, {10, 100}, {-0.15, 0.15}},  
  ColorFunction -> ColorData["SunsetColors"],  
  ColorFunctionScaling -> True,  
  GridLines -> Automatic,  
  GridLinesStyle -> Directive[Dotted],  
  Frame -> True,  
  FrameLabel -> {{ "Z", None }, { "N", "B/Aexp-B/ABW" }},  
  BaseStyle -> {FontFamily -> "Verdana", FontSize -> 22},  
  AspectRatio -> 0.75,  
  PlotLegends -> Placed[Automatic, Right],  
  ImageSize -> 600]
```



Efekt sił pairing

Rozważmy obszar wokół izotopów cyny:

```
gdiff2 = ListDensityPlot[Flatten[diff, 1],
  PlotRange -> {{49, 85}, {40, 60}, {-0.1, 0.12}},
  ColorFunction -> ColorData["SunsetColors"],
  ColorFunctionScaling -> True,
  GridLines -> Automatic,
  GridLinesStyle -> Directive[Dotted],
  Frame -> True,
  FrameLabel -> {{ "Z", None }, { "N", "B/Aexp-B/ABW" }},
  BaseStyle -> {FontFamily -> "Verdana", FontSize -> 22},
  AspectRatio -> 0.7,
  PlotLegends -> Placed[Automatic, Right],
  ImageSize -> 600]
```



Zobaczymy dokładniej różnice między eksperymentem a modelem dla izotopów cyny ($Z=50$) i antymonu ($Z=51$):

```
diffZ50 = {#[[1]], #[[3]]} & /@diff[[50]];
diffZ51 = {#[[1]], #[[3]]} & /@diff[[51]];
```

```

g50 = ListPlot[diffZ50,
  Frame → True,
  Joined → True,
  MeshStyle → Red,
  Mesh → All,
  GridLines → Automatic,
  GridLinesStyle → Directive[Dotted],
  FrameLabel → {"N", "(B/A)exp - (B/A)BW [MeV]"},
  BaseStyle → {FontFamily → "Verdana", FontSize → 20},
  PlotRange → {Automatic, {-0.05, 0.2}},
  PlotStyle → GrayLevel[0.5],
  ImageSize → 600];

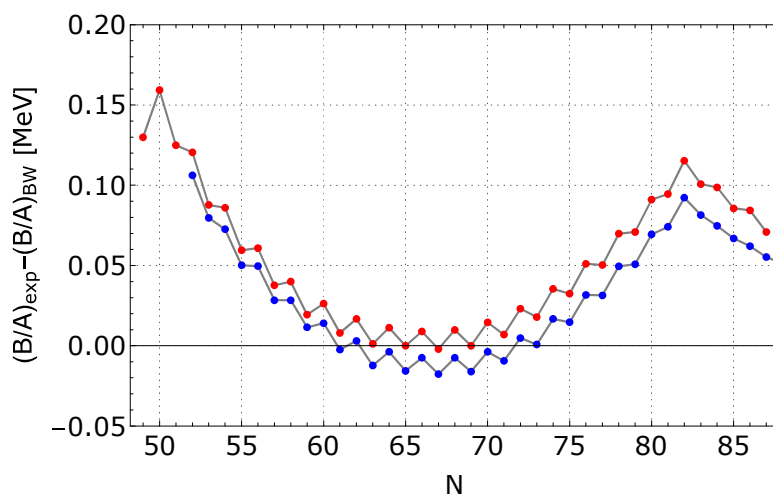
```

```

g51 = ListPlot[diffZ51,
  Frame → True,
  Joined → True,
  MeshStyle → Blue,
  Mesh → All,
  GridLines → Automatic,
  GridLinesStyle → Directive[Dotted],
  FrameLabel → {"N", "(B/A)exp - (B/A)BW [MeV]"},
  BaseStyle → {FontFamily → "Verdana", FontSize → 20},
  PlotRange → {Automatic, {-0.05, 0.2}},
  PlotStyle → GrayLevel[0.5],
  ImageSize → 600];

```

```
Show[g50, g51]
```



Wyznaczamy Δ_n

Dla nieparzystego neutronu:

```
deltaN[n_, z_] := Module[{x0, xm1, xp1, a, d},
  If[EvenQ[n] && EvenQ[z],
    a = n + z;
    x0 = QuantityMagnitude[IsotopeData[{z, a}, "BindingEnergy"]] * a;
    xm1 = QuantityMagnitude[IsotopeData[{z, a - 1}, "BindingEnergy"]] * (a - 1);
    xp1 = QuantityMagnitude[IsotopeData[{z, a + 1}, "BindingEnergy"]] * (a + 1);
    If[NumericQ[x0] && NumericQ[xm1] && NumericQ[xp1],
      d = (2 * x0 - xm1 - xp1) / 2,
      d = 0],
    d = 0];
  d
]
```

```
deltaN[20, 20]
```

3.6402

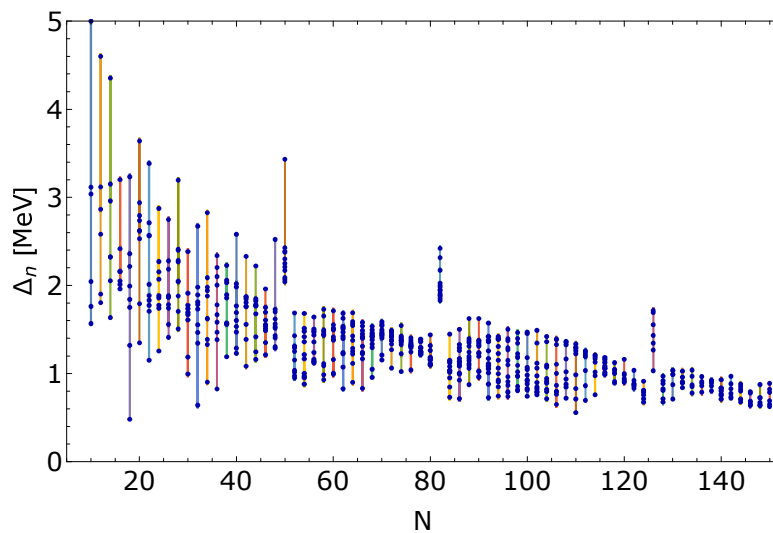
Dla izotonów zadanego N :

```
fdeltaN[nn_] := Module[{lizo, m, lfin},
  lizo = Select[IsotopeData[], (IsotopeData[#, "NeutronNumber"] == nn) &&
    (EvenQ[IsotopeData[#, "AtomicNumber"]]) &];
  lfin = {nn, deltaN[nn, IsotopeData[#, "AtomicNumber"]]} & /@ lizo;
  Select[lfin, #[[2]] > 0 &]
]
```

Zbieramy dane dla wszystkich parzystych N i rysujemy:

```
tN = Table[fdeltaN[n], {n, 10, 150, 2}];
```

```
pldN = ListPlot[tN,
  Joined -> True,
  Mesh -> All,
  MeshStyle -> Directive[Darker[Blue], PointSize[0.007]],
  FrameLabel -> {"N", " $\Delta_n$  [MeV]},
  Frame -> True,
  PlotRange -> {{5, 152}, {0, 5}},
  BaseStyle -> {FontFamily -> "Verdana", FontSize -> 20},
  PlotStyle -> Automatic,
  ImageSize -> 600,
  Filling -> False
]
```

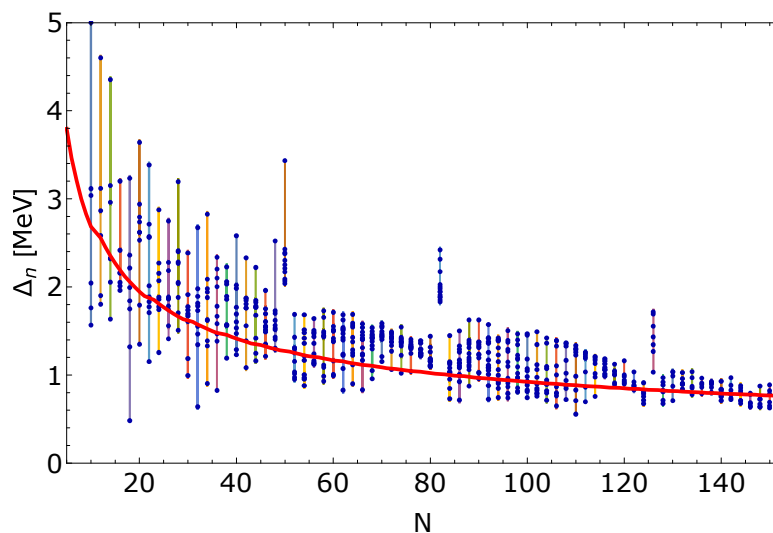


Jak wiadomo, jako wartość Δ przyjmuje się $\Delta = \frac{12}{\sqrt{A}} \text{MeV}$. Zobaczmy jak to się ma do danych doświadczalnych. Wartość Z bierzemy z funkcji `stableZBW[A]` :

```
fdN = Table[{a - Round[stableZBW[a]], 12 / Sqrt[a]}, {a, 10, 250, 2}];
```

```
plfdN = ListPlot[fdN,
  Joined -> True,
  FrameLabel -> {"N", "Δn [MeV]"},
  Frame -> True,
  PlotRange -> {{5, 150}, {0, 5}},
  BaseStyle -> {FontFamily -> "Verdana", FontSize -> 20},
  PlotStyle -> {{Thickness[0.005], Hue[1]}},
  ImageSize -> 600,
  Filling -> False];
```

```
Show[{pldN, plfdN}]
```



Wyznaczamy Δ_p

Powtarzamy te same kroki dla protonów.

Dla nieparzystego protonu:

```
deltaZ[n_, z_] := Module[{x0, xm1, xp1, a, d},
  If[EvenQ[n] && EvenQ[z],
    a = n + z;
    x0 = QuantityMagnitude[IsotopeData[{z, a}, "BindingEnergy"]] * a;
    xm1 =
      QuantityMagnitude[IsotopeData[{z - 1, a - 1}, "BindingEnergy"]] * (a - 1);
    xp1 = QuantityMagnitude[IsotopeData[{z + 1, a + 1}, "BindingEnergy"]] *
      (a + 1);
    If[NumericQ[x0] && NumericQ[xm1] && NumericQ[xp1],
      d = (2 * x0 - xm1 - xp1) / 2,
      d = 0],
    d = 0];
  d
]
```

```
deltaZ[60, 40]
```

2.033

Dla izotopów danego Z:

```
fdeltaZ[zz_] := Module[{lizo, m, lfin},
  lizo = Select[IsotopeData[], (IsotopeData[#, "AtomicNumber"] == zz) &&
    (EvenQ[IsotopeData[#, "NeutronNumber"]]) &];
  lfin = {zz, deltaZ[IsotopeData[#, "NeutronNumber"], zz]} & /@ lizo;
  Select[lfin, #[[2]] > 0 &]
]
```

```
|
```

```
fdeltaZ[30]
```

```
{{30, 2.0}, {30, 1.6}, {30, 2.464}, {30, 1.906},
 {30, 1.885}, {30, 1.828}, {30, 1.684}, {30, 1.626},
 {30, 1.926}, {30, 1.983}, {30, 2.081}, {30, 1.8}, {30, 1.7}}
```

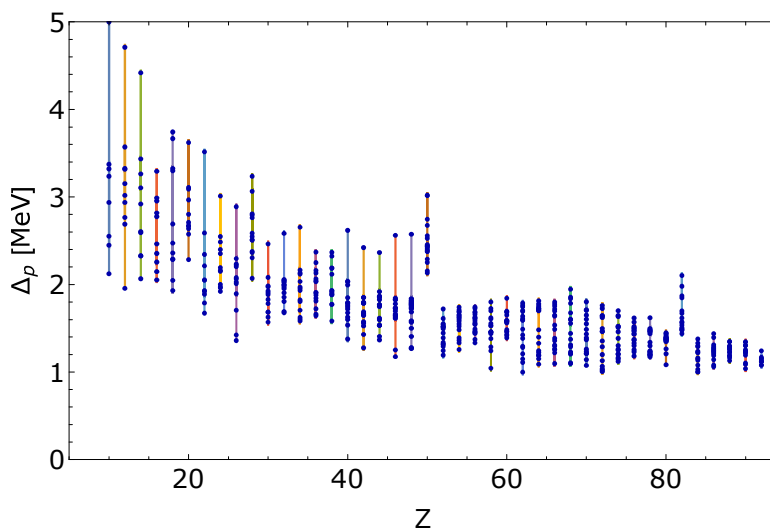
Dla wszystkich Z:

```
tZ = Table[fdeltaZ[z], {z, 10, 92, 2}];
```

```

pldZ = ListPlot[tZ,
  Joined -> True,
  Mesh -> All,
  MeshStyle -> Directive[Darker[Blue], PointSize[0.007]],
  FrameLabel -> {"Z", " $\Delta_p$  [MeV]"},
  Frame -> True,
  PlotRange -> {{5, 94}, {0, 5}},
  BaseStyle -> {FontFamily -> "Verdana", FontSize -> 20},
  PlotStyle -> Automatic,
  ImageSize -> 600,
  Filling -> False
]

```



```

fdZ = Table[{stableZBW[a], 16 / Sqrt[a]}, {a, 10, 240, 2}];

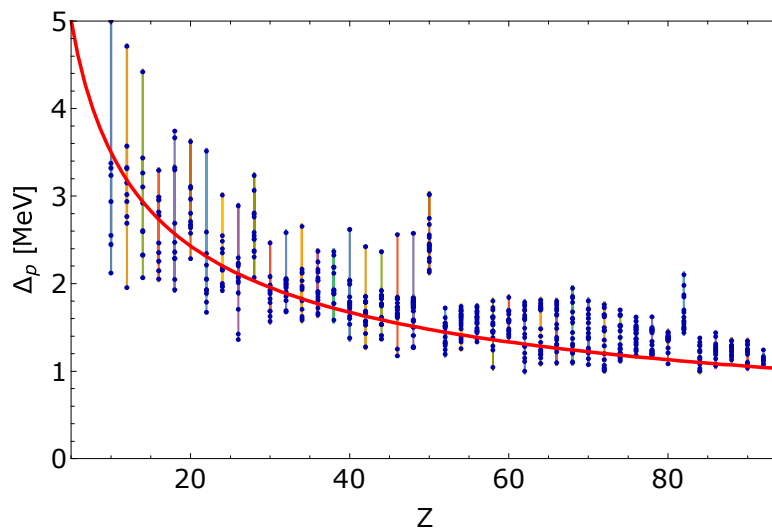
```

```

plfdZ = ListPlot[fdZ,
  Joined -> True,
  FrameLabel -> {"Z", " $\Delta_p$  [MeV]"},
  Frame -> True,
  PlotRange -> {{5, 94}, {0, 5}},
  BaseStyle -> {FontFamily -> "Verdana", FontSize -> 20},
  PlotStyle -> {{Thickness[0.005], Hue[1]}},
  ImageSize -> 600,
  Filling -> False];

```

```
Show[{pldZ, plfdZ}]
```



Pełny model z członem *pairing*

Dodajemy człon opisujący siły *pairing* do modelu B-W.

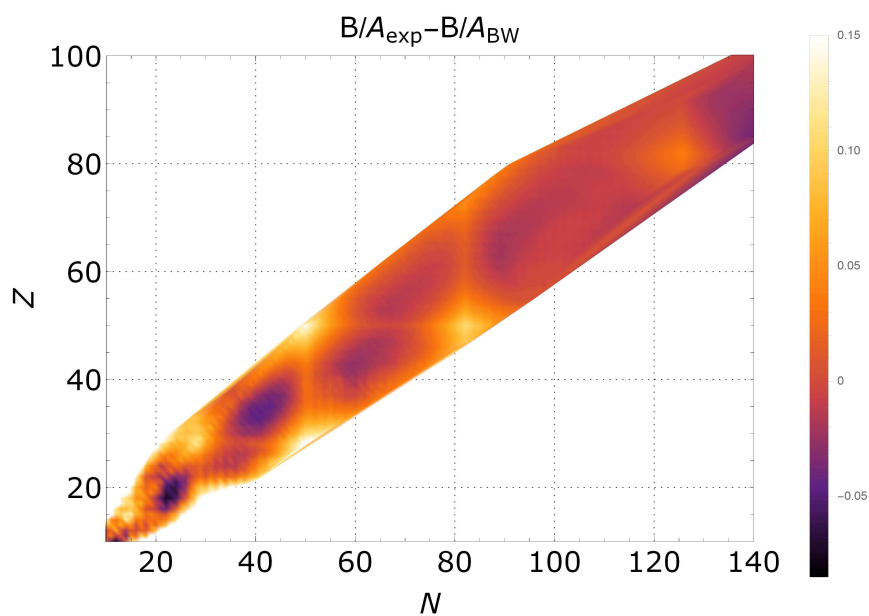
```
bpairBW = 12.0; (* parametr członu pairing [MeV] *)
```

```
beBWpair[A_, Z_] :=  
beBW[A, Z] + If[EvenQ[A], If[EvenQ[Z], 1, -1], 0] * bpairBW / Sqrt[A]
```

Sprawdźmy jak teraz wygląda różnica między doświadczeniem a modelem B-W dla energii wiązania:

```
diffpair =  
Table[{a - z, z, QuantityMagnitude[IsotopeData[{z, a}, "BindingEnergy"]] -  
beBWpair[a, z] / a}, {z, 1, 110},  
{a, IsotopeData[#, "MassNumber"] & /@ IsotopeData[z]};
```

```
gdiffpair = ListDensityPlot[Flatten[diffpair, 1],  
  PlotRange → {{10, 140}, {10, 100}, {-0.15, 0.15}},  
  ColorFunction → ColorData["SunsetColors"],  
  ColorFunctionScaling → True,  
  GridLines → Automatic,  
  GridLinesStyle → Directive[Dotted],  
  Frame → True,  
  FrameLabel → {"Z", None}, {"N", "B/Aexp-B/ABW"},  
  BaseStyle → {FontFamily → "Verdana", FontSize → 22},  
  AspectRatio → 0.75,  
  PlotLegends → Placed[Automatic, Right],  
  ImageSize → 600]
```

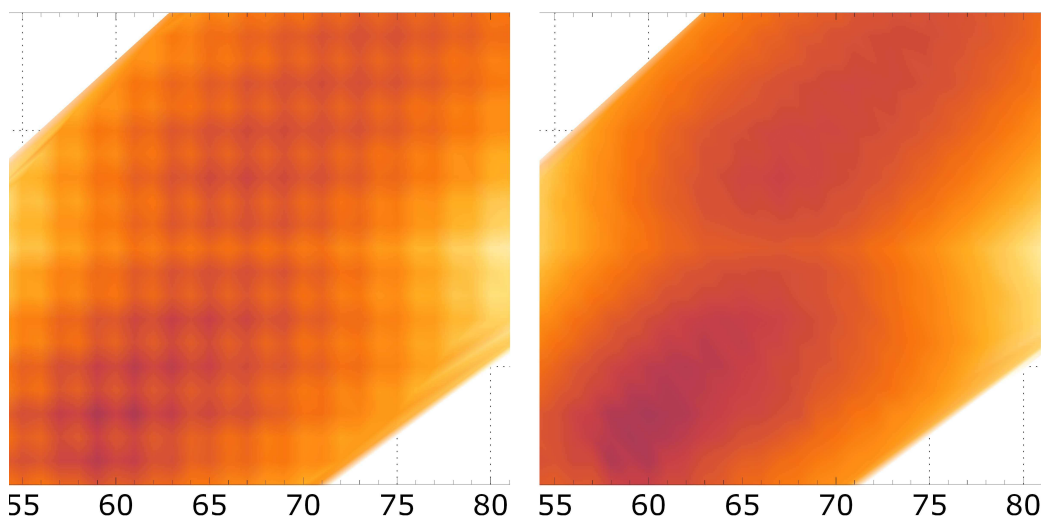


Obszar wokół cyny

Wybieramy jeszcze raz ten sam obszar wokół cyny:

```
gdiff2pair = ListDensityPlot[Flatten[diffpair, 1],
  PlotRange -> {{49, 85}, {40, 60}, {-0.1, 0.12}},
  ColorFunction -> ColorData["SunsetColors"],
  ColorFunctionScaling -> True,
  GridLines -> Automatic,
  GridLinesStyle -> Directive[Dotted],
  Frame -> True,
  FrameLabel -> {{ "Z", None}, {"N", "B/Aexp - (B/ABW+pairing)"}},
  BaseStyle -> {FontFamily -> "Verdana", FontSize -> 22},
  AspectRatio -> 0.7,
  PlotLegends -> Placed[Automatic, Right],
  ImageSize -> 600];
```

```
Show[GraphicsRow[{gdiff2, gdiff2pair}], ImageSize -> 800]
```



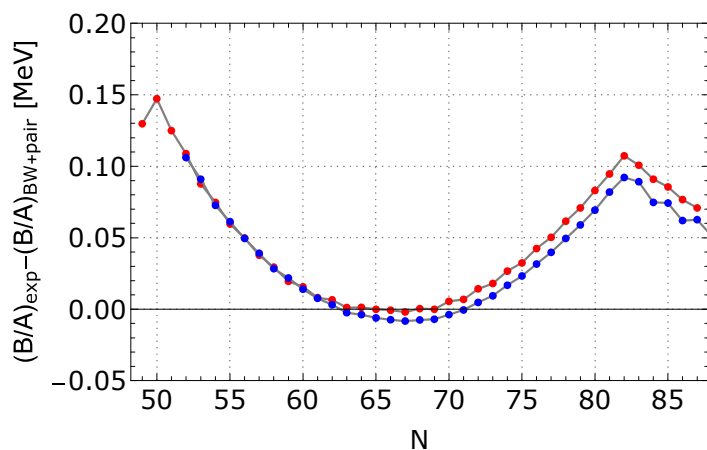
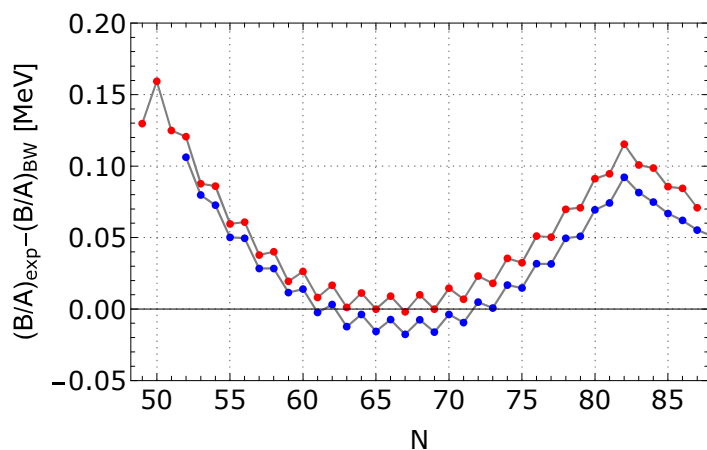
Dla samych izotopów cyny i antymonu :

```
diffZ50pair = {#[[1]], #[[3]]} & /@ diffpair[[50]];
diffZ51pair = {#[[1]], #[[3]]} & /@ diffpair[[51]];
```

```
g50pair = ListPlot[diffZ50pair,  
  Frame → True,  
  Joined → True,  
  MeshStyle → Red,  
  Mesh → All,  
  GridLines → Automatic,  
  GridLinesStyle → Directive[Dotted],  
  FrameLabel → {"N", "(B/A)exp - (B/A)BW+pair [MeV]"},  
  BaseStyle → {FontFamily → "Verdana", FontSize → 20},  
  PlotRange → {Automatic, {-0.05, 0.2}},  
  PlotStyle → GrayLevel[0.5],  
  ImageSize → 600];
```

```
g51pair = ListPlot[diffZ51pair,  
  Frame → True,  
  Joined → True,  
  MeshStyle → Blue,  
  Mesh → All,  
  GridLines → Automatic,  
  GridLinesStyle → Directive[Dotted],  
  FrameLabel → {"N", "(B/A)exp - (B/A)BW+pair [MeV]"},  
  BaseStyle → {FontFamily → "Verdana", FontSize → 20},  
  PlotRange → {Automatic, {-0.05, 0.2}},  
  PlotStyle → GrayLevel[0.5],  
  ImageSize → 600];
```

```
Show[GraphicsColumn[{Show[g50, g51], Show[g50pair, g51pair]}],  
  ImageSize → 600]
```

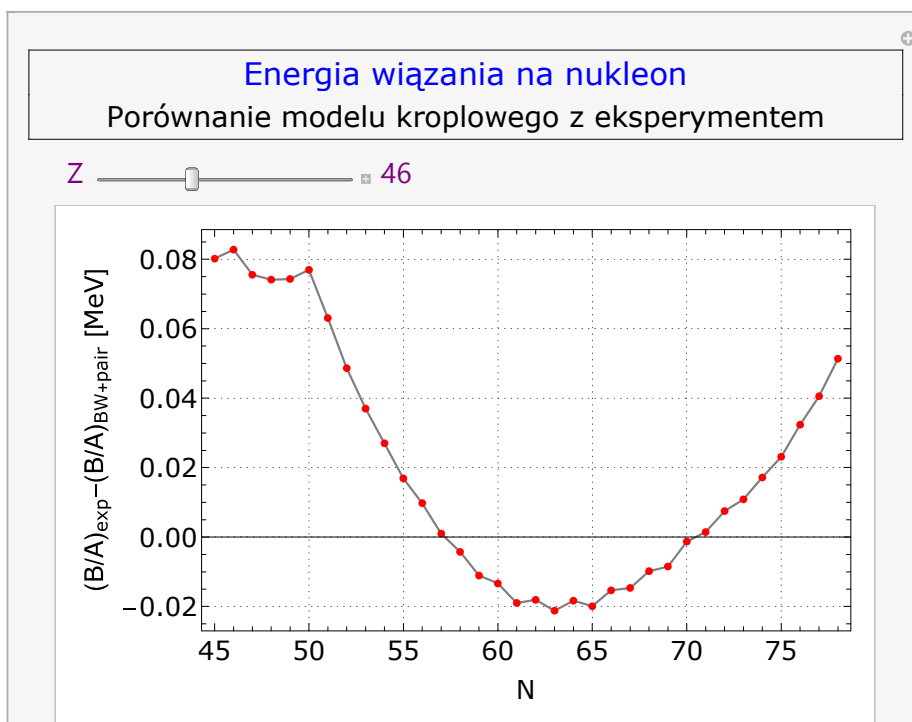
Manipulacja : porównanie dla dowolnego Z

```
Manipulate[
  dataZ = diff[[z]];
  ListPlot[dataZ,
    Frame → True,
    Joined → True,
    MeshStyle → Red,
    Mesh → All,
    GridLines → Automatic,
    GridLinesStyle → Directive[Dotted],
    FrameLabel → {"N", "(B/A)exp - (B/A)BW+pair [MeV]"},
    BaseStyle → {FontFamily → "Verdana", FontSize → 20},
    PlotRange → Automatic,
    PlotStyle → GrayLevel[0.5],
    ImageSize → 600
  ],
  {{z, 50, Style["Z", Purple, 22]}, 10, 110, 1,
    Appearance -> "Labeled", LabelStyle -> Directive[Purple, 22]},
  TrackedSymbols → {z},
  FrameLabel → {None, None, fulltit, None},
  SaveDefinitions → True,
  Initialization ⇒ {
    bepair[a_, z_] :=
      (15.8 * a - 18.3 * a^(2/3) - 0.714 * z * (z - 1) / a^(1/3) - 23.2 * (a - 2 * z)^2 /
```

```

a + If[EvenQ[a], If[EvenQ[z], 1, -1], 0] * 12.0 / Sqrt[a]) / a;
diff = Table[{a - z, QuantityMagnitude[IsotopeData[{z, a},
  "BindingEnergy"]] - bepair[a, z]}, {z, 1, 110},
  {a, IsotopeData[#, "MassNumber"] & /@ IsotopeData[z]}];
titleA = Text[Style["Energia wiązania na nukleon",
  Blue, FontFamily -> "Verdana", 22]];
titleB = Text[Style["Porównanie modelu kropowego z eksperymentem",
  FontFamily -> "Verdana", Black, 20]];
fulltit = GraphicsColumn[{titleA, titleB}, AspectRatio -> 0.10,
  ImageSize -> 700, Frame -> True]];
SynchronousInitialization -> False
]

```



Model powłokowy

Rozwiązujemy numerycznie równanie Schrödingera

Wstęp

Model kropłowy i inne podobne modele makroskopowe, mimo niewątpliwych sukcesów, nie są w stanie opisać wszystkich własności jądra. W szczególności nie da się w nich ująć zjawisk, w których uczestniczą pojedyncze nukleony (efekty jednocząstkowe).

Co więcej, możemy oczekiwać, że wszystkie własności jądra powinny dać się zrozumieć i opisać mikroskopowo, jako wynik oddziaływania między nukleonami. Model powłokowy jest pierwszym ważnym krokiem w tym kierunku.

W przybliżeniu nierelatywistycznym i przyjmując tylko siły dwuciałowe między nukleonami, hamiltonian układu A nukleonów jest:

$$H = T + V = \sum_{i=1}^A \frac{\mathbf{p}_i^2}{2m} + \sum_{i>k=1}^A V_{ik}(\mathbf{r}_i - \mathbf{r}_k)$$

Ponieważ nie znamy dobrze potencjału sił jądrowych, to możemy uprościć problem wprowadzając pole średnie (potencjał jednocząstkowy):

$$H = \sum_{i=1}^A \left[\frac{\mathbf{p}_i^2}{2m} + U_i(\mathbf{r}) \right] + \left(\sum_{i>k=1}^A V_{ik}(\mathbf{r}_i - \mathbf{r}_k) - U_i(\mathbf{r}) \right)$$

Jeśli dobrze dobierzemy jednocząstkowe pole średnie U , to wyraz w prawym nawiasie będzie mały i będzie traktowany jako zaburzenie (oddziaływanie resztkowe). W pierwszym przybliżeniu pomijamy ten wyraz i budujemy jądro z nukleonów, które obsadzają stany

jednocząstkowe w potencjale U .

Istotę i funkcjonowanie modelu powłokowego można poznać przyjmując za U potencjał fenomenologiczny, mniej lub bardziej realistyczny. Takie podejście stosuje się też w wielu prostych zastosowaniach modelu. Lepszym rozwiązaniem jest wyznaczenie pola średniego metodą Hartree-Fock'a, ale to jest temat na inną okazję.

Potencjały fenomenologiczne

W praktyce wykorzystuje się najczęściej dwa potencjały fenomenologiczne: oscylatora harmonicznego i Woods-Saxona.

- Potencjał 3-wym. oscylatora harmonicznego (HO):

$$U_{\text{HO}}(r) = -V_1 + \frac{1}{2} m \omega^2 r^2.$$

Ma rozwiązania analityczne, co sprawia, że odgrywa ważną rolę, jako źródło dogodnej ortonormalnej bazy stanów. Pozwala łatwo wprowadzić deformację (model Nilssona). Często dodaje się człon $\sim l^2$, aby “wyplaszczyć dno” i uczynić go bardziej realistycznym.

- Potencjał Woods-Saxona (WS):

$$U_{\text{WS}}(r) = \frac{-V_0}{\exp\left[\frac{r-R}{a}\right] + 1}.$$

Najbardziej realistyczny z potencjałów, wymaga jednak obliczeń numerycznych.

Wymienione potencjały opisują jedynie główny wkład sił jądrowych. Do tego trzeba dodać odpychanie kulombowskie między protonami, oraz dodatkowy wyraz opisujący jądrowe oddziaływanie spin-orbita.

- Pełny, realistyczny potencjał modelu powłokowego (sferycznie symetryczny):

$$U(r) = U_{\text{WS}}(r) + V_C(r) + V_{\text{LS}}(r) \mathbf{L} \cdot \mathbf{S}.$$

Równanie Schrödingera

Równanie na stany stacjonarne cząstki o masie m w zadanym potencjale U (równanie Schrödingera niezależne od czasu):

$$\frac{-\hbar^2}{2m} \nabla^2 \psi_n(\mathbf{r}) + U(\mathbf{r}) \psi_n(\mathbf{r}) = E_n \psi_n(\mathbf{r}).$$

W przypadku sferycznie symetrycznym można dokonać separacji zmiennych kątowych (θ , φ) od r podstawiając:

$$\psi_n(\mathbf{r}) = R_n(r) \cdot f(\theta, \varphi).$$

Rozwiązaniem części kątowej są oczywiście harmoniki sferyczne: $f(\theta, \varphi) = Y_l^m(\theta, \varphi)$, czyli funkcje własne orbitalnego momentu pędu l . Część radialna funkcji falowej spełnia wtedy równanie:

$$\left[\frac{-\hbar^2}{2m} \nabla_r^2 + \frac{\hbar^2 l(l+1)}{2mr^2} + U(r) \right] R_{nl}(r) = E_{nl} R_{nl}(r),$$

czyli

$$\frac{-\hbar^2}{2m} \frac{1}{r} \frac{d^2}{dr^2} r R_{nl}(r) = \left[E_{nl} - \frac{\hbar^2 l(l+1)}{2mr^2} - U(r) \right] R_{nl}(r).$$

Wprowadzamy nową funkcję $u_{nl}(r) = r R_{nl}(r)$ i porządkujemy:

$$\frac{d^2}{dr^2} u_{nl}(r) = -\frac{2m}{\hbar^2} \left[E_{nl} - \frac{\hbar^2 l(l+1)}{2mr^2} - U(r) \right] u_{nl}(r).$$

To równanie będziemy chcieli rozwiązać numerycznie, żeby uzyskać funkcje radialne i wartości energii stanów jednocząstkowych.

Rozgrzewka: studnia prostokątna

Na początek rozważymy podobne równanie jednowymiarowe, lecz prostsze, żeby zapoznać się z jedną z metod numerycznych.

Rozwiązujemy:

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

z prostokątną jamą potencjału o skończonej głębokości:

$$V(x) = \begin{cases} 0 & \text{dla } -a/2 < x < a/2 \\ V_0 & \text{wszędzie poza tym} \end{cases}$$

Szukamy stanów związanych w tej jamie potencjału, czyli takich że $E < V_0$.

Zadanie to można rozwiązać analitycznie, ale dla nas pełni rolę treningową.

Bardzo ładna ilustracja tego przykładu, ze zbioru Wolfram Demonstration Project, jest [tutaj](#).

Metoda “strzelania” (shooting)

Podejście to jest zainspirowane wykładem Daniela V. Schroedera ([link](#) punkt 7.), na który zwrócił moją uwagę Krzysztof Miernik.

- Warunki brzegowe

Wiemy, że funkcja falowa musi znikać poza studnią. Założymy więc, że jej wartość jest równa zero w pewnym punkcie na lewo od studni, a jej pochodna w tym punkcie jest prawie równa zero. Będziemy szukać (przez zgadywanie, stąd nazwa “shooting”) takiej wartości energii, żeby funkcja zniknęła także po prawej stronie studni.

- Naturalne jednostki

W tego typu problemach należy wybrać rozsądne jednostki, aby algorytm numeryczny miał do czynienia z małymi liczbami. Nie tylko przyspiesza to rachunki, ale czasem je wręcz umożliwia.

Tu naturalną jednostką długości jest a - szerokość studni. Kładziemy też $m/\hbar^2 = 1$.

Jednostką energii będzie wtedy $\hbar^2/(m a^2)$. W tych jednostkach równanie przybiera postać:

$$\frac{d^2}{dx^2} \psi(x) = -2 [E - V(x)] \psi(x).$$

Głębokość studni V_0 jest wolnym parametrem, od którego będą zależeć rozwiązania.

- Wybór V_0

Aby wartości liczbowe miały, mimo wszystko, jakieś odniesienie do jąder atomowych, głębokość studni określamy następująco. Bezwzględną głębokość studni ustalamy na 50

MeV. Jako m bierzemy masę nukleonu, $940 \text{ MeV}/c^2$. Wybieramy teraz “promień” studni, czyli $a/2$, w femtometrach. Ustala to jednostkę energii: $eunit = 197^2 / (940 \cdot a^2) \text{ MeV}$. Wówczas $V_0 = 50 / eunit$.

Manipulacja 1: prostokątna studnia potencjału (1D)

Uwaga: wartość energii oraz jej skok kontrolują przyciski (**Button**). Związane z nimi zmienne są globalne. Trzeba pamiętać, żeby w tej samej sesji unikać powtarzania ich nazw.

```
Manipulate[
Module[
{eunit, v0, v, sol, psi, norm, xb = 2.0,
studnia =
Graphics[{Thick, Darker[Gray],
Line[{{-1.8, 2}, {-0.5, 2}, {-0.5, -2}, {0.5, -2}, {0.5, 2}, {1.8, 2}}]}]
},
eunit = 197.^2 / (940. * (2 * radius)^2);
(* jednostka energii w MeV*)
v0 = 50. / eunit;
(* definicja potencjału *)
v[x_] := If[Abs[x] < 0.5, 0, v0];
(* centralne polecenie: numeryczne rozwiązanie równania S. *)
sol = NDSolve[{psi''[x] == -2 * (energy1 - v[x]) psi[x],
psi[-xb] == 0, psi'[-xb] == 0.0001}, psi, {x, -xb, xb}];
(* normalizacja rozwiązania *)
norm = NIntegrate[(psi[x])^2 /. sol, {x, -xb * 0.6, xb * 0.6}];
(* wykres otrzymanego rozwiązania *)
graph = Plot[psi[x] / Sqrt[norm] /. sol, {x, -xb, xb},
PlotRange -> {{-2, 2}, {-2.2, 2.2}},
Frame -> True,
```

```

FrameLabel → {"x [2R]", "ψ(x)"},
Axes → {True, False},
BaseStyle → {Black, FontFamily → "Verdana", FontSize → 18},
PlotStyle → {Red, Thickness[0.005]},
ImageSize → 500];
Grid[{
  {Row[{Style[" Jednostka energii = ", Black, FontSize → 18],
    Style[NumberForm[eunit, 3],
    Blue, FontFamily → "Verdana", FontSize → 18],
    Style[" MeV ", Black, FontSize → 18]}]},
  Show[graph, studnia]
},
,
{Row[{Style["Głębokość studni = ", Black, FontSize → 18],
  Style[NumberForm[v0, 3], Blue, FontFamily → "Verdana", FontSize → 18],
  Style[" j.e. = 50 MeV", Black, FontSize → 18]}]},
  SpanFromAbove
},
{Graphics[Text[Style[StringForm[" "]]],
  AspectRatio → 0.03, ImageSize → 300],
  SpanFromAbove (* dummy, to make a space *)
},
,
{Button[" Energy Up ",
  energy1 = energy1 + step1,

```

```

  BaseStyle → 18,
  ImageMargins → 5,
  ImageSize → All]
Button[" Step Up ",
  step1 = If[step1 == 10., 10, step1 * 10.],
  BaseStyle → 18,
  ImageMargins → 5,
  ImageSize → All],
  SpanFromAbove
},
,
{Button["Energy Down",
  energy1 = energy1 - step1,
  BaseStyle → 18,
  ImageMargins → 5,
  ImageSize → All]
  Button["Step Down",
  step1 = If[step1 == 0.00001, 0.00001, step1 / 10.],
  BaseStyle → 18,
  ImageMargins → 5,
  ImageSize → All],
  SpanFromAbove
},
,
{Row[{Style[" Krok energii = ", Black, FontSize → 18],
  Style[step1, Blue, FontFamily → "Verdana", FontSize → 18],
  Style[" j.e. ", Black, FontSize → 18]}]},

```



```

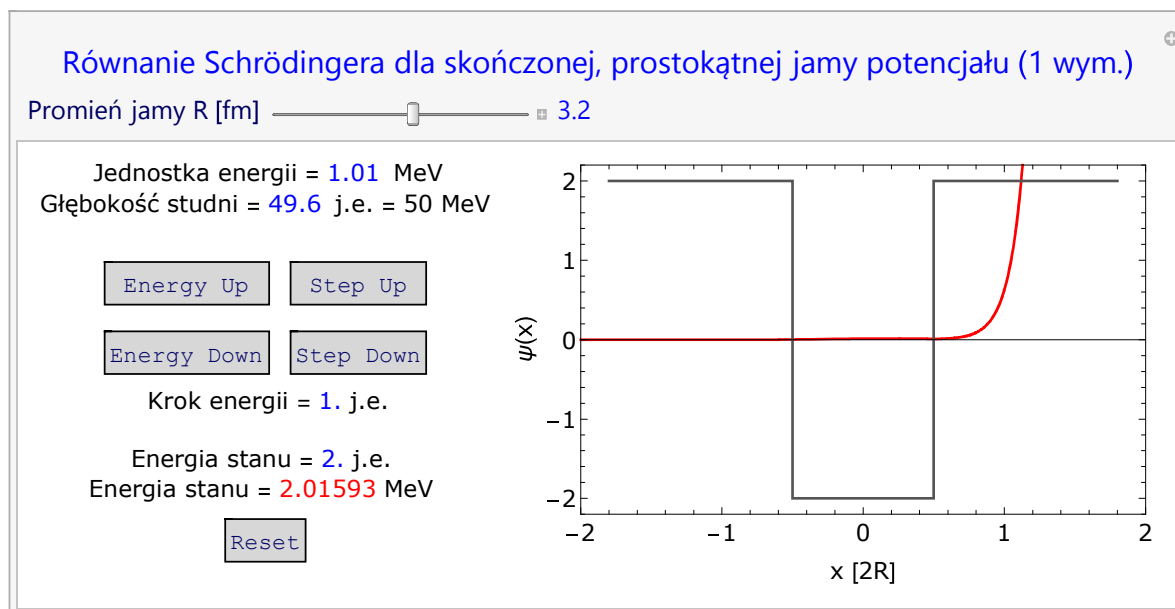
    SpanFromAbove
  },
  {Graphics[Text[Style[StringForm["  "]]],
    AspectRatio → 0.03, ImageSize → 200],
    SpanFromAbove (* dummy, to make a space *)
  }
,
{Row[{Style["Energia stanu = ", Black, FontSize → 18],
  Style[NumberForm[energy1, 6],
    Blue, FontFamily → "Verdana", FontSize → 18],
  Style[" j.e. ", Black, FontSize → 18]}],
  SpanFromAbove
},
{Row[{Style["Energia stanu = ", Black, FontSize → 18],
  Style[NumberForm[energy1 * eunit, 6],
    Red, FontFamily → "Verdana", FontSize → 18],
  Style[" MeV ", Black, FontSize → 18]}],
  SpanFromAbove
},
{Button["Reset",
  energy1 = 2.;
  step1 = 1.,
  BaseStyle → 18,
  ImageMargins → 5,
  ImageSize → All],
  SpanFromAbove
}

```

```

  }
}
, Alignment → Top, ItemSize → Full]
],
(* Kontrolki *)
{{radius, 2.0, Style["Promień jamy R [fm]", 20]}, 1, 5, 0.1,
  Appearance → "Labeled", LabelStyle → Directive[Blue, 20]},
FrameLabel → {None, None,
  Style["Równanie Schrödingera dla skończonej, prostokątnej
  jamy potencjału (1 wym.)", Blue, 24], None},
Initialization → {
  step1 = 1.,
  energy1 = 2.
}
]

```



Potencjał Woodsa - Saxona (WS)

Wracamy do “prawdziwych” jąder atomowych i szukamy funkcji radialnych i energii jednocząstkowych dla realistycznego potencjału Woodsa-Saxona. W tym celu musimy sprecyzować potencjał: $U(r) = U_{WS}(r) + V_C(r) + V_{LS}(r) \mathbf{L} \cdot \mathbf{S}$. Wybieramy parametryzację “klasyczną” pochodzącą z Książki (Bohr, Mottelson) opisaną też w podręczniku J. Suhonena “From Nucleons to Nuclei” (Springer 2007).

- Wyraz główny, $U_{WS}(r)$

$$U_{WS}(r) = \frac{-V_0}{\exp\left[\frac{r-R}{a}\right] + 1},$$

$$V_0 = \left(51 \pm 33 \frac{N-Z}{A}\right) \text{MeV}, \quad R = r_0 A^{1/3}, \quad r_0 = 1.27 \text{ fm}, \quad a = 0.67 \text{ fm}.$$

w powyżym znak “+” odnosi się do protonów, a “-” do neutronów.

- Człon kulombowski, $V_C(r)$

$$V_C(r) = \frac{Ze^2}{4\pi\epsilon_0} \begin{cases} \frac{3-(r/R)^2}{2R} & r \leq R \\ 1/r & r > R \end{cases},$$

Człon ten występuje tylko dla protonów. W obliczeniach korzystamy z tego, że $e^2/(4\pi\epsilon_0) = 1.44 \text{ MeV}\cdot\text{fm}$.

- Spin-orbita, $V_{LS}(r) \mathbf{L} \cdot \mathbf{S}$

$$V_{LS}(r) = v^{(0)} \left(\frac{r_0}{\hbar} \right)^2 \frac{1}{r} \left[\frac{d}{dr} \frac{1}{\exp\left[\frac{r-R}{a}\right] + 1} \right].$$

Szukane funkcje falowe są funkcjami o dobrze określonym całkowitym i orbitalnym momencie pędu, j i l , przy czym $j = l+1/2$ lub $j = l-1/2$ w zależności od ustawienia spinu nukleonu względem orbitalnego momentu pędu. Operator $\mathbf{L} \cdot \mathbf{S}$ działając na taką funkcję daje:

$$\mathbf{L} \cdot \mathbf{S} \psi_{nlj} = \hbar^2 \frac{1}{2} \left[j(j+1) - l(l+1) - \frac{3}{4} \right] \psi_{nlj}.$$

Zbierając, do równania musimy wstawić:

$$V_{LS}(r) \mathbf{L} \cdot \mathbf{S} = v^{(0)} \frac{r_0^2}{2 a r} \frac{\exp\left[\frac{r-R}{a}\right]}{\left(\exp\left[\frac{r-R}{a}\right] + 1\right)^2} \left[j(j+1) - l(l+1) - \frac{3}{4} \right],$$

$$v^{(0)} = 0.44 \cdot V_0.$$

- Człon centryfugalny, V_L :

$$V_L(r) = \frac{\hbar^2 l(l+1)}{2 m r^2}$$

formalnie, nie należy do potencjału, ale zmienia postać równania na funkcje radialne, więc ma znaczenie i warto zobaczyć jego wpływ na potencjał “efektywny”.

W literaturze występuje kilka innych parametryzacji potencjału Woodsa-Saxona. Jedna z nich, dość nowoczesna i chyba bardzo dobra, opisana jest w pracy:

[N. Schwierz, I. Wiedenhöver, and A. Volya, “Parameterization of the Woods-Saxon Potential for Shell-Model Calculations”, arxiv, 2007.](#)

Manipulacja 2 : kształt potencjału WS

```
Manipulate[
Module[
{
(* Wartości parametrów klasycznych *)
V0 = 51.0 (* MeV *),
R0 = 1.27 (* fm *),
a = 0.67 (* fm *),
kappa = 33.0 / 51.,
lambda = 0.44,
(* zmienne lokalne *)
hc = 197.3,
mred, R, RSO, n, j, V1, VN, VC, VL, VLS, Vtot
},
(* definicja potencjału Woodsa-Saxona według SWV *)
mred = 940.0; (* (A-1)/A; *)
R = R0 * A^(1/3);
RSO = R;
n = A - Z;
j = 1 + sign * (1/2);
V1 = V0 * (1 + iso * kappa * (n - Z) / A);
VN[r_] := -V1 / (1 + Exp[(r - R) / a]);
VC[r_] := If[iso == 1, 1, 0] *
If[r < R, 1.44 * Z * (3 - (r/R)^2) / (2 * R), 1.44 * (Z - 1) / r];
```

```

VL[r_] := (hc^2 / (2 * mred)) * l * (1 + 1) / r^2;
VLS[r_] := If[LsOn, -lambda * V1 * R0^2 / (2 * a * r) * Exp[(r - RSO) / a] /
  ((Exp[(r - RSO) / a] + 1)^2) * (j * (j + 1) - 1 * (1 + 1) - 3 / 4), 0];
Vtot[r_] := VN[r] + VC[r] + VL[r] + VLS[r];
(* Plot command *)
gtot = Plot[Vtot[r], {r, 0.2, 21},
  PlotRange -> {{0, 21}, {-60, 50}}, PlotStyle -> {Black, Thickness[0.008]};
gN = Plot[VN[r], {r, 0.2, 21}, PlotRange -> {{0, 21}, {-60, 50}},
  PlotStyle -> {Blue, Thickness[0.005]};
gC = Plot[VC[r], {r, 0.2, 21}, PlotRange -> {{0, 21}, {-60, 50}},
  PlotStyle -> {Red, Thickness[0.005]};
gL = Plot[VL[r], {r, 0.2, 21}, PlotRange -> {{0, 21}, {-60, 50}},
  PlotStyle -> {Magenta, Thickness[0.005]};
gLs = Plot[VLS[r], {r, 0.2, 21}, PlotRange -> {{0, 21}, {-60, 50}},
  PlotStyle -> {Orange, Thickness[0.005]};
Show[gN, gC, gL, gLs, gtot,
  AxesOrigin -> {0.0, 0.0},
  BaseStyle ->
  Directive[FontFamily -> "Helvetica", FontSize -> 16, AxesStyle -> Thick],
  AxesLabel -> {"r [fm]", "Energia [MeV]"},
  AspectRatio -> 0.9, ImageSize -> 500]
],
(* Kontrolki *)
{{Z, 71, Style["Z", 18]}, 2, 100, 1,
  Appearance -> "Labeled", LabelStyle -> Directive[Red, 18]},
(**)
{{A, 151, Style["A", 18]}, Z, 250, 1,

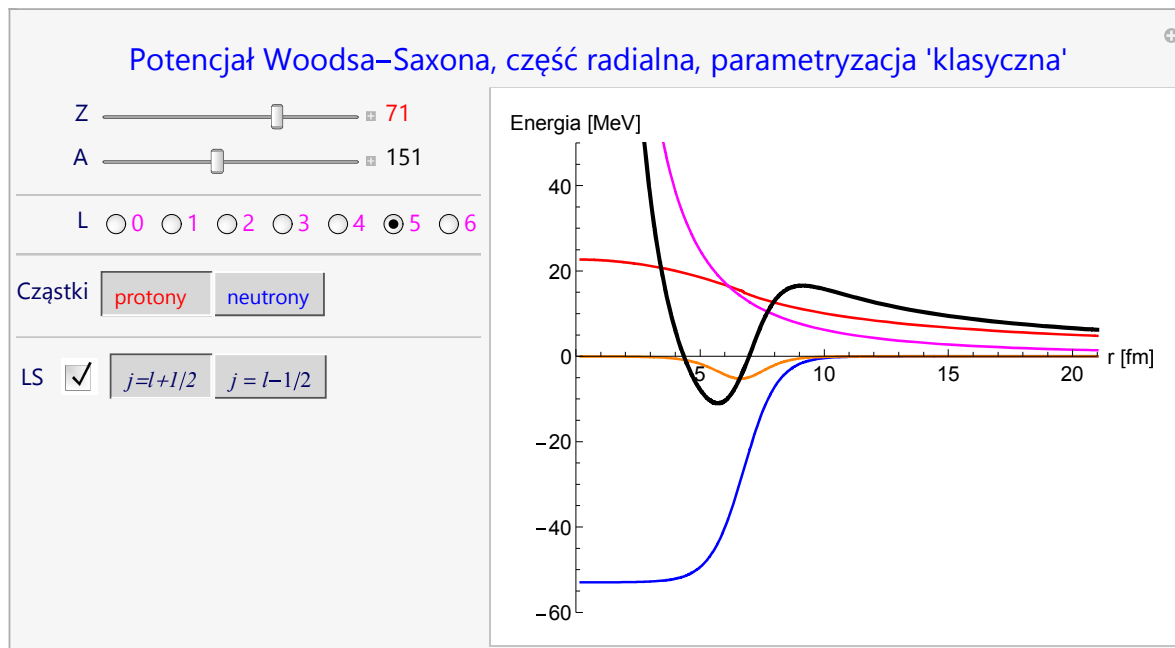
```

```

  Appearance -> "Labeled", LabelStyle -> Directive[Black, 18]},
(**)
Delimiter,
(**)
{{1, 0, Style["L", 18]}, {0, 1, 2, 3, 4, 5, 6}, ControlType -> RadioButtonBar,
  Appearance -> "Labeled", LabelStyle -> Directive[Magenta, 18]},
(**)
Delimiter,
(**)
{{iso, 1, Style["Cząstki", 18]},
  {1 -> Style[" protony ", Red, 16], -1 -> Style[" neutrony ", Blue, 16]}}},
Delimiter,
Row[{Control[{
  {LsOn, False, Style["LS ", 18]}, {True, False}
}],
  Control[{
    {sign, 1, ""},
    {1 -> Style[" j=l+1/2 ", 17],
     -1 -> Style[" j=l-1/2 ", 17]},
    Enabled -> LsOn && l > 0
  ]}],
(**)
ControlPlacement -> Left,
ContinuousAction -> False,
FrameLabel -> {None, None, Style[
  "Potencjał Woodsa-Saxona, część radialna, parametryzacja 'klasyczna'",
  Blue, 24], None}

```

]



Potencjał WS, metoda “strzelania”

Przystępujemy do rozwiązywania radialnej części równania Schrödingera z potencjałem WS. Stosujemy metodę “strzelania”. Przypominamy pełne równanie:

$$\frac{d^2}{dr^2} u_{nlj}(r) = -\frac{2m}{\hbar^2} \left[E_{nlj} - \frac{\hbar^2 l(l+1)}{2mr^2} - U(r) \right] u_{nlj}(r).$$

Podobnie jak poprzednio obieramy “naturalne” jednostki. Narzucającą się jednostką długości jest promień jądra R . Przyjmujemy $\hbar^2/m = 1$, wtedy jednostką energii jest $(\hbar c)^2 / (mc^2 R^2)$. Za mc^2 przyjmujemy 940 MeV (choć powinna być masa zredukowana!).

Manipulacja 3: metoda "strzelania"

Aby ułatwić zapisywanie wyników, tworzymy ich listę. Przycisk **Zapisz** dopisze do niej wartości aktualnych parametrów. Uwaga: Jeśli lista już zawiera jakieś dane, to wykonanie poniższej komendy **skasuje wszystko** co w niej było!

```
listaWynikow = {}

{}

Manipulate[
  Module[{
    (* Wartości parametrów "klasycznych" *)
    V0 = 51.0 (* MeV *),
    R0 = 1.27 (* fm *),
    a0 = 0.67 (* fm *),
    kappa = 33.0 / 51.,
    lambda = 0.44,
    hc = 197.3,
    (* pozostałe zmienne lokalne *)
    A, mred, a, Vtot, norm, psi, g, norm2
  },
  (* definicja potencjału Woodsa-Saxona według SWV *)
  (* przechodzimy do jednostek "naturalnych", R=1 *)
  A = n + Z;
  mred = 940.0; (* (A-1)/A; *)
```

```
R = R0 * A^(1/3); (*jednostka długości w fm*)
eunit = (hc)^2 / (mred * R^2); (* jednostka energii w MeV *)
a = a0 / R;
Vtot[r_] := Module[{V1, VN, VC, VL, VLS},
  V1 = (V0 / eunit) * (1 + iso * kappa * (n - Z) / A);
  VN = -V1 / (1 + Exp[(r - 1) / a]);
  VC =
  If[iso == 1, 1, 0] * (1.44 * Z / (eunit * R)) * If[r < 1, (3 - r^2) / 2, 1.0 / r];
  VL = (1 / 2) * 1 * (1 + 1) / r^2;
  VLS = If[IsOn,
    If[l > 0,
      - (lambda / 2) * (V1 / eunit) *
      (R0 / R)^2 / (r * a) * Exp[(r - 1) / a] / ((Exp[(r - 1) / a] + 1)^2) *
      If[sign > 0, 1, -(1 + 1)],
    0],
  0];
  VN + VC + VL + VLS
];
vmin = First[FindMinimum[{Vtot[x], 0.001 ≤ x ≤ 1}, {x, 0.5}]];
solu = NDSolve[{psi'[x] == -2 * (energy2 - Vtot[x]) * psi[x],
  psi[0.0001] == 0, psi'[0.0001] == 1.0}, psi, {x, 0.0001, 5}];
norm = NIntegrate[(psi[x])^2 /. solu, {x, 0.001, 3}];
g = Plot[psi[x] / Sqrt[norm] /. solu, {x, 0.001, 4},
  PlotRange → {{0, 4}, {-2, 2}},
  Frame → True,
  Axes → {True, False},
```

```

FrameLabel → {"r/R", "r·ψ(r)"},
BaseStyle → {Black, FontFamily → "Verdana", FontSize → 18},
PlotStyle → {Red, Thickness[0.005]},
ImageSize → 600,
Epilog → {{Dashed, Line[{{0, 0}, {4, 0}}]}}];
Row[{
  Column[{
    Row[{Style[" Jednostka długości, R = ", Black, FontSize → 17],

      Style[NumberForm[R, 3], Blue, FontFamily → "Verdana", FontSize → 17],
      Style[" fm", Black, FontSize → 17]}],
    Row[{Style[" Jednostka energii = ", Black, FontSize → 17],
      Style[NumberForm[eunit, 3],
        Blue, FontFamily → "Verdana", FontSize → 17],
      Style[" MeV ", Black, FontSize → 17]}],
    Row[{Style["Głębokość studni = ", Black, FontSize → 17],
      Style[NumberForm[vmin, 5],
        Blue, FontFamily → "Verdana", FontSize → 17]}],
    Row[{Button[" Energy Up ",
      energy2 = energy2 + step2,
      BaseStyle → 17,
      ImageMargins → 5,
      ImageSize → All]
      Button[" Step Up ",
      step2 = If[step2 == 10., 10, step2 * 10.],
      BaseStyle → 17,

```

```

      ImageMargins → 5,
      ImageSize → All]}],
  Row[{Button["Energy Down",
    energy2 = energy2 - step2,
    BaseStyle → 17,
    ImageMargins → 5,
    ImageSize → All]
    Button["Step Down",
    step2 = If[step2 == 0.0000001, 0.0000001, step2 / 10.],
    BaseStyle → 17,
    ImageMargins → 5,
    ImageSize → All]}],
  Row[{Style[" Krok energii = ", Black, FontSize → 17],
    Style[step2, Blue, FontFamily → "Verdana", FontSize → 17],
    Style[" j.e. ", Black, FontSize → 17]}],
  Row[{Style["Energia stanu = ", Black, FontSize → 17],
    Style[NumberForm[energy2, 6],
      Blue, FontFamily → "Verdana", FontSize → 17],
    Style[" j.e. ", Black, FontSize → 18]}],
  Row[{Style["Energia stanu = ", Black, FontSize → 17],
    Style[NumberForm[energy2 * eunit, 6],
      Red, FontFamily → "Verdana", FontSize → 17],
    Style[" MeV ", Black, FontSize → 17]}],
  Row[{Button["Reset",
    energy2 = vmin; step2 = 1.,
    BaseStyle → 17,
    ImageMargins → 5,

```

```

    ImageSize → All],
    Spacer[20],
    Button["Zapisz",
    wavefun =
      Table[First[Flatten[psi[x] /. solu]], {x, 0.01, 3.01, 0.02}];
    norm2 = R / 50. * Sum[wavefun[[i]]^2, {i, 1, Length[wavefun]}];
    wavefun = Table[{(0.01 + i - 1) * R / 50,
      wavefun[[i]] / Sqrt[norm2]}, {i, 1, Length[wavefun]}];
    AppendTo[listaWynikow, {Z, n, n + Z, iso, 1, LsOn, sign,
      eunit, energy2, eunit * energy2, wavefun}],
    BaseStyle → 17,
    ImageMargins → 5,
    ImageSize → All],
    Style[Length[listaWynikow],
      Black, FontFamily → "Verdana", FontSize → 17]
  ]
}, Alignment → Center, Spacings → 1], (* secondary controls *)
Show[g, ImageSize → {500, 350}] (* plot panel *)
}]
], (* End of Module *)
(*Control panel*)
Panel[
  Column[{
    Row[{
      Control[{
        {Z, 54, Style["Z", 18]}, 2, 100, 1,

```

```

    Appearance → "Labeled", LabelStyle → Directive[Red, 18]}],
    Spacer[20],
    Control[{
      {n, 82, Style["N", 18]}, 2, 150, 1,
      Appearance → "Labeled", LabelStyle → Directive[Blue, 18]}],
    Spacer[20],
    Style["A = ", 18], Style[Dynamic[n + Z], 18]
  ]], (* 1st row of controls end *)
  Row[{
    Control[{
      {iso, -1, Style["Cząstki", 18]}, {1 → Style[" protony ", Red, 16],
      -1 → Style[" neutrony ", Blue, 16]}
    }],
    Spacer[40],
    Control[{
      {1, 0, Style["L", 18]},
      {0, 1, 2, 3, 4, 5, 6}, ControlType → RadioButtonBar,
      Appearance → "Labeled", LabelStyle → Directive[Magenta, 18]
    }],
    Spacer[30],
    Control[{
      {LsOn, False, Style["LS ", 18]}, {True, False}
    }],
    Control[{
      {sign, 1, ""},
      Which[
        1 == 0, {1 → Style[" s1/2 ", 17], -1 → Style[" s1/2 ", 17]},

```



```

l = 1, {1 → Style[" p3/2 ", 17], -1 → Style[" p1/2 ", 17]},
l = 2, {1 → Style[" d5/2 ", 17], -1 → Style[" d3/2 ", 17]},
l = 3, {1 → Style[" f7/2 ", 17], -1 → Style[" f5/2 ", 17]},
l = 4, {1 → Style[" g9/2 ", 17], -1 → Style[" g7/2 ", 17]},
l = 5, {1 → Style[" h11/2 ", 17], -1 → Style[" h9/2 ", 17]},
l = 6, {1 → Style[" i13/2 ", 17], -1 → Style[" i11/2 ", 17]}
],
ControlType → Setter,
Enabled → LsOn
}]
}]] (* 2nd row of controls end *)
], Alignment → Center, Spacings → 1]
], (* Control panel end*)
FrameLabel →
{None, None, Style["Równanie Schrödingera dla potencjału Woodsa-Saxona ",
Blue, 24], None},
Initialization → {
step2 = 1.,
energy2 = -20.
}
]

```

Równanie Schrödingera dla potencjału Woodsa-Saxona

Z N A = 136

Cząstki L 0 1 2 3 4 5 6 LS

Jednostka długości, R = 6.53 fm
 Jednostka energii = 0.971 MeV
 Głębokość studni = -45.53

Krok energii = 1. j.e.
 Energia stanu = -20. j.e.
 Energia stanu = -19.4173 MeV

0

```
listaWynikow[[1]]
```

Zapisujemy listę wyników na dysk. Najpierw przechodzimy do właściwego folderu

```
SetDirectory["d:/Pfutzner/Teach/Mathematica4Nuclear/05_ModelPowlokowy"]
```

```
d:\Pfutzner\Teach\Mathematica4Nuclear\05_ModelPowlokowy
```

```
Directory[]
```

```
d:\Pfutzner\Teach\Mathematica4Nuclear\05_ModelPowlokowy
```

i robimy **Export**. Lepiej zachować tę instrukcję wykomentowaną, aby zabezpieczyć się przed niekontrolowanym wykonaniem jej, co nadpisze istniejący plik np. plikiem pustym.

```
Export["Wave_136Xe_p0f72.dat", listaWynikow[[3, 11]], "Table"]
```

```
Wave_136Xe_p0f72.dat
```

Wyniki przykładowe: 136Xe

Klika chwil klikania-strzelania pozwoliło uzyskać energie stanów jednocząstkowych neutronów i protonów dla 136Xe (Z=54, N=82). Aby ujawnić wpływ oddziaływania spin-orbita rachunki wykonano dwa razy: bez tego oddziaływania i z nim.

Wyniki bez członu spin-orbita:

```
neutronsWS = {-38.81, -33.69, -27.62,
              -25.62, -20.76, -17.96, -13.25, -10.10, -9.045};
```

```
protonsWS =
{-35.82, -30.97, -24.96, -22.33, -18.03, -14.44, -10.31, -6.19, -4.71};
```

Wyniki z członem spin orbita :

```
neutronsWSLS = {-38.81, -33.91, -33.27, -28.26, -26.72, -25.65, -22.00,
                -19.20, -18.34, -17.21, -15.25, -10.94, -10.89, -9.088, -8.83};
```

```
protonsWSLS = {-35.82, -31.27, -30.42, -25.80, -23.83, -22.36, -19.61,
                -16.09, -14.91, -13.51, -12.84, -7.38, -7.26, -4.76, -4.58};
```

Dla pełnego porównania dodamy jeszcze energie z potencjału oscylatora harmonicznego (bez członu kulombowskiego i bez oddziaływania spin-orbita). Dane są one wyrażeniem:

$$E_{nl}^{HO} = -V_1 + \left(N + \frac{3}{2}\right) \hbar\omega = -V_1 + \left(2n + l + \frac{3}{2}\right) \hbar\omega, \quad n = 0, 1, 2, 3, \dots$$

Częstość oscylatora, $\hbar\omega$, można oszacować (Blomqvist-Molinari):

$$\hbar\omega = (45 A^{-1/3} + 25 A^{-2/3}) \text{ MeV.}$$

Jako głębokość potencjału bierzemy $V_1 = 50 \text{ MeV}$.

```
A = 136;
homega = 45. * A ^ (-1 / 3) - 25. * A ^ (-2 / 3)
```

7.80518

```
pureHO = Table[-50 + (n + 3 / 2) * 7.805, {n, 0, 4}]
{-38.2925, -30.4875, -22.6825, -14.8775, -7.0725}
```

Rysujemy poziomy:

```
gHO = Graphics[Table[{Black, Thick, Line[{{20, #}, {27, #}}]}] & /@pureHO];
gNWS = Graphics[Table[{Blue, Thick, Line[{{30, #}, {37, #}}]}] & /@neutronsWS];
gNWSLS =
  Graphics[Table[{Red, Thick, Line[{{40, #}, {47, #}}]}] & /@neutronsWSLS];
gPWS = Graphics[Table[{Blue, Thick, Line[{{10, #}, {17, #}}]}] & /@protonsWS];
gPWSLS = Graphics[Table[{Red, Thick, Line[{{0, #}, {7, #}}]}] & /@protonsWSLS];
```

I przedstawiamy na wspólnym rysunku, pracowicie opisując wszystko ręcznie :-)

```
Show[gHO, gNWS, gNWSLS, gPWS, gPWSLS,
  Frame -> {{True, True}, {False, False}},
  FrameLabel -> {None, "Energia [MeV]},
  FrameTicks -> {{All, All}, {None, None}},
  PlotRange -> {{-5, 52}, {-40.5, -1}},
  LabelStyle -> Directive[FontFamily -> "Helvetica", FontSize -> 20],
  AspectRatio -> 0.6,
  PlotLabel -> Style["Stany jednocząstkowe dla 136Xe", Darker[Blue], 24],
  ImageSize -> 1000,
  Epilog -> {
    {Thin, Black, Dashed,
      Line[{{37, neutronsWS[[2]]}, {40, neutronsWSLS[[2]]}}]},
    {Thin, Black, Dashed, Line[{{37, neutronsWS[[2]]},
      {40, neutronsWSLS[[3]]}}]},
    {Thin, Black, Dashed, Line[{{37, neutronsWS[[3]]},
      {40, neutronsWSLS[[4]]}}]},
    {Thin, Black, Dashed, Line[{{37, neutronsWS[[3]]},
      {40, neutronsWSLS[[5]]}}]},
    {Thin, Black, Dashed, Line[{{37, neutronsWS[[5]]},
      {40, neutronsWSLS[[7]]}}]},
    {Thin, Black, Dashed, Line[{{37, neutronsWS[[5]]},
      {40, neutronsWSLS[[8]]}}]},
    {Thin, Black, Dashed, Line[{{37, neutronsWS[[6]]},
      {40, neutronsWSLS[[9]]}}]},
```

```

{Thin, Black, Dashed, Line[{{37, neutronsWS[[6]]},
  {40, neutronsWSLS[[10]]}}]},
{Thin, Black, Dashed, Line[{{37, neutronsWS[[7]]},
  {40, neutronsWSLS[[11]]}}]},
{Thin, Black, Dashed, Line[{{37, neutronsWS[[7]]},
  {40, neutronsWSLS[[13]]}}]},
{Thin, Black, Dashed, Line[{{37, neutronsWS[[8]]},
  {40, neutronsWSLS[[12]]}}]},
{Thin, Black, Dashed, Line[{{37, neutronsWS[[8]]},
  {40, neutronsWSLS[[15]]}}]},
{Thin, Black, Dashed, Line[{{10, protonsWS[[2]]},
  {7, protonsWSLS[[2]]}}]},
{Thin, Black, Dashed, Line[{{10, protonsWS[[2]]},
  {7, protonsWSLS[[3]]}}]},
{Thin, Black, Dashed, Line[{{10, protonsWS[[3]]},
  {7, protonsWSLS[[4]]}}]},
{Thin, Black, Dashed, Line[{{10, protonsWS[[3]]},
  {7, protonsWSLS[[5]]}}]},
{Thin, Black, Dashed, Line[{{10, protonsWS[[5]]},
  {7, protonsWSLS[[7]]}}]},
{Thin, Black, Dashed, Line[{{10, protonsWS[[5]]},
  {7, protonsWSLS[[8]]}}]},
{Thin, Black, Dashed, Line[{{10, protonsWS[[6]]},
  {7, protonsWSLS[[9]]}}]},
{Thin, Black, Dashed, Line[{{10, protonsWS[[6]]},
  {7, protonsWSLS[[10]]}}]},
{Thin, Black, Dashed, Line[{{10, protonsWS[[7]]},

```

```

  {7, protonsWSLS[[11]]}}]},
{Thin, Black, Dashed, Line[{{10, protonsWS[[7]]},
  {7, protonsWSLS[[12]]}}]},
{Thin, Black, Dashed, Line[{{10, protonsWS[[8]]},
  {7, protonsWSLS[[13]]}}]},
{Thin, Black, Dashed, Line[{{10, protonsWS[[8]]},
  {7, protonsWSLS[[14]]}}]},
Text[Style[" protony ", Black, 22], {8, -3}],
Text[Style[" neutrony ", Black, 22], {38, -3}],
Text[Style[" WS+SO ", Darker[Red], 20], {3.5, -40.3}],
Text[Style[" WS ", Darker[Blue], 20], {13.5, -40.3}],
Text[Style[" HO ", Black, 20], {23.5, -40.3}],
Text[Style[" WS ", Darker[Blue], 20], {33.5, -40.3}],
Text[Style[" WS+SO ", Darker[Red], 20], {43.5, -40.3}],
Text[Style[" N ", Black, 17], {19.2, -40}],
Text[Style["0", Black, 17], {19.2, -38.5}],
Text[Style["1", Black, 17], {19.2, -30.5}],
Text[Style["2", Black, 17], {19.2, -22.7}],
Text[Style["3", Black, 17], {19.2, -14.9}],
Text[Style["4", Black, 17], {19.2, -7.01}],
Text[Style["0s", Black, 17], {27, -38.5 + 0.8}],
Text[Style["0p", Black, 17], {27, -30.5 + 0.8}],
Text[Style["1s 0d", Black, 17], {27, -22.7 + 0.8}],
Text[Style["1p 0f", Black, 17], {27, -14.9 + 0.8}],
Text[Style["2s 1d 0g", Black, 17], {27, -7.01 + 0.8}],
Text[Style["0s", Blue, 17], {38, neutronsWS[[1]]}],
Text[Style["0p", Blue, 17], {38, neutronsWS[[2]]}, Background -> White],

```

```

Text[Style["0d", Blue, 17], {38, neutronsWS[[3]]}, Background → White],
Text[Style["1s", Blue, 17], {38, neutronsWS[[4]]}],
Text[Style["0f", Blue, 17], {38, neutronsWS[[5]]}, Background → White],
Text[Style["1p", Blue, 17], {38, neutronsWS[[6]]}, Background → White],
Text[Style["0g", Blue, 17], {38, neutronsWS[[7]]}, Background → White],
Text[Style["1d", Blue, 17], {38, neutronsWS[[8]]}, Background → White],
Text[Style["2s", Blue, 17], {38, neutronsWS[[9]]}],
Text[Style["0s", Blue, 17], {9.2, protonsWS[[1]]}],
Text[Style["0p", Blue, 17], {9.2, protonsWS[[2]]}, Background → White],
Text[Style["0d", Blue, 17], {9.2, protonsWS[[3]]}, Background → White],
Text[Style["1s", Blue, 17], {9.2, protonsWS[[4]]}],
Text[Style["0f", Blue, 17], {9.2, protonsWS[[5]]}, Background → White],
Text[Style["1p", Blue, 17], {9.2, protonsWS[[6]]}, Background → White],
Text[Style["0g", Blue, 17], {9.2, protonsWS[[7]]}, Background → White],
Text[Style["1d", Blue, 17], {9.2, protonsWS[[8]]}, Background → White],
Text[Style["2s", Blue, 17], {9.2, protonsWS[[9]]}],
Text[Style["0s1/2", Red, 17], {48.6, neutronsWSLS[[1]]}],
Text[Style["0p3/2", Red, 17], {48.6, neutronsWSLS[[2]] - 0.2}],
Text[Style["0p1/2", Red, 17], {48.6, neutronsWSLS[[3]] + 0.2}],
Text[Style["0d5/2", Red, 17], {48.6, neutronsWSLS[[4]]}],
Text[Style["0d3/2", Red, 17], {48.6, neutronsWSLS[[5]]}],
Text[Style["1s1/2", Red, 17], {48.6, neutronsWSLS[[6]] + 0.1}],
Text[Style["0f7/2", Red, 17], {48.6, neutronsWSLS[[7]]}],
Text[Style["0f5/2", Red, 17], {48.6, neutronsWSLS[[8]] - 0.3}],
Text[Style["1p3/2", Red, 17], {48.6, neutronsWSLS[[9]] + 0.2}],
Text[Style["1p1/2", Red, 17], {48.6, neutronsWSLS[[10]]}],
Text[Style["0g9/2", Red, 17], {48.6, neutronsWSLS[[11]]}],

```

```

Text[Style["0g7/2", Red, 17], {48.6, neutronsWSLS[[13]] + 0.2}],
Text[Style["1d5/2", Red, 17], {48.6, neutronsWSLS[[12]] - 0.5}],
Text[Style["2s1/2", Red, 17], {48.6, neutronsWSLS[[14]] - 0.2}],
Text[Style["1d3/2", Red, 17], {48.6, neutronsWSLS[[15]] + 0.5}],
Text[Style["0s1/2", Red, 17], {-1.4, protonsWSLS[[1]]}],
Text[Style["0p3/2", Red, 17], {-1.4, protonsWSLS[[2]] - 0.2}],
Text[Style["0p1/2", Red, 17], {-1.4, protonsWSLS[[3]] + 0.2}],
Text[Style["0d5/2", Red, 17], {-1.4, protonsWSLS[[4]]}],
Text[Style["0d3/2", Red, 17], {-1.4, protonsWSLS[[5]]}],
Text[Style["1s1/2", Red, 17], {-1.4, protonsWSLS[[6]] + 0.1}],
Text[Style["0f7/2", Red, 17], {-1.4, protonsWSLS[[7]]}],
Text[Style["0f5/2", Red, 17], {-1.4, protonsWSLS[[8]] - 0.3}],
Text[Style["1p3/2", Red, 17], {-1.4, protonsWSLS[[9]] + 0.2}],
Text[Style["1p1/2", Red, 17], {-1.4, protonsWSLS[[10]]}],
Text[Style["0g9/2", Red, 17], {-1.4, protonsWSLS[[11]]}],
Text[Style["0g7/2", Red, 17], {-1.4, protonsWSLS[[12]] - 0.5}],
Text[Style["1d5/2", Red, 17], {-1.4, protonsWSLS[[13]] + 0.2}],
Text[Style["2s1/2", Red, 17], {-1.4, protonsWSLS[[15]] + 0.5}],
Text[Style["1d3/2", Red, 17], {-1.4, protonsWSLS[[14]] - 0.4}],
Text[Style["2", Black, 20], {43.5, -36.5}],
Text[Style["8", Black, 20], {43.5, -31}],
Text[Style["20", Black, 20], {43.5, -24}],
Text[Style["28", Black, 20], {43.5, -20.8}],
Text[Style["50", Black, 20], {43.5, -13.5}],
Text[Style["2", Black, 20], {3.5, -34}],
Text[Style["8", Black, 20], {3.5, -29}],
Text[Style["20", Black, 20], {3.5, -21.5}],

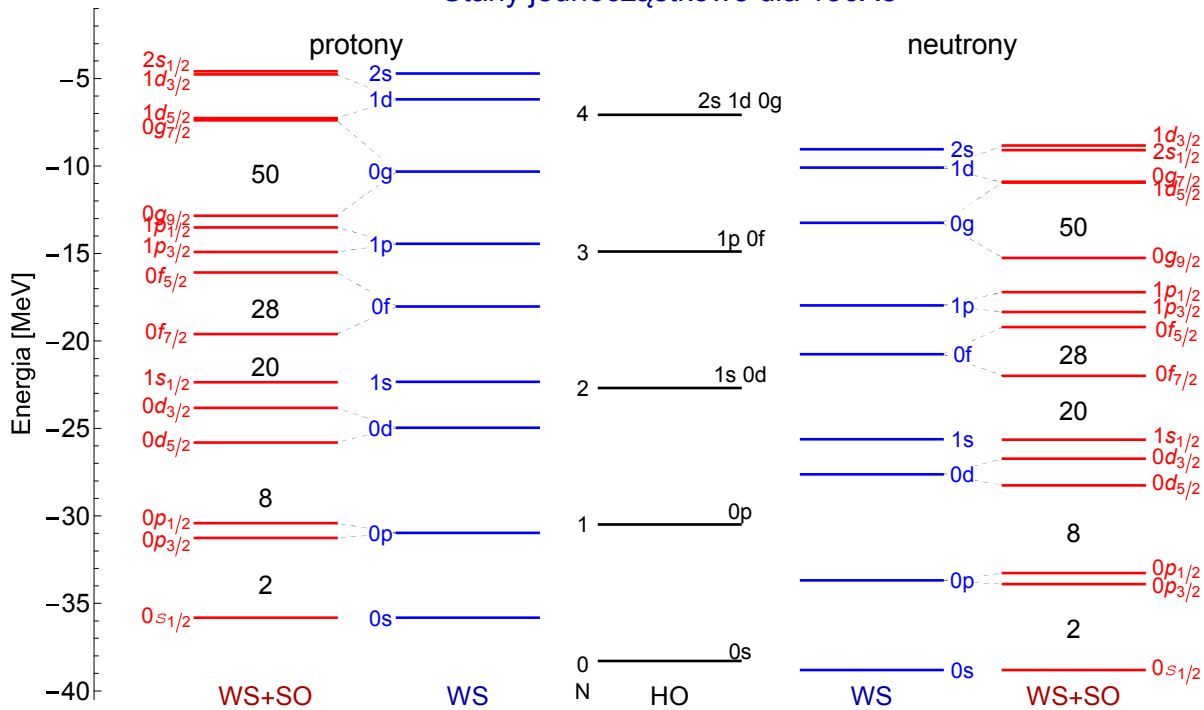
```

```

Text[Style["28", Black, 20], {3.5, -18.2}],
Text[Style["50", Black, 20], {3.5, -10.5}]
}
]

```

Stany jednocząstkowe dla ^{136}Xe



Rozwiązanie w bazie oscylatora harmonicznego

Stany własne i energie jednocząstkowe równania S. z potencjałem WS można znacznie szybciej i wygodniej, choć mniej dokładnie, wyznaczyć inną, bardzo ogólną i ważną metodą.

Przedstawiamy szukane rozwiązania jako kombinację liniową znanych funkcji, które tworzą układ zupełny, ortonormalny. Jako tę bazę wybieramy rozwiązania oscylatora harmonicznego:

$$R_{nlj}(r) = \sum_{\nu} A_{\nu}^{(nlj)} g_{\nu l}(r), \quad \sum_{\nu} [A_{\nu}^{(nlj)}]^2 = 1$$

Tworzymy macierz, której elementy są zadane przez:

$$M_{\nu' \nu} = \langle \nu' | h_{lj} | \nu \rangle \equiv \int_0^{\infty} g_{\nu' l}(r) h_{lj} g_{\nu l}(r) r^2 dr,$$

gdzie h_{lj} jest hamiltonianem, dla którego szukamy rozwiązań. Zgodnie z wcześniejszym, jego postać to:

$$h_{lj} = \frac{-\hbar^2}{2m} \left[\nabla_r^2 - \frac{l(l+1)}{r^2} \right] + U_{WS}(r) + V_C(r) + V_{LS}(r) \cdot \frac{\hbar^2}{2} \left[j(j+1) - l(l+1) - \frac{3}{4} \right].$$

Aby obliczyć elementy macierzowe musimy “zadziałać” h na funkcje g_{nl} . Wypiszmy najpierw r -nie oscylatora harmonicznego, które spełniają te funkcje:

$$\begin{aligned} \left[\frac{-\hbar^2}{2m} \left[\nabla_r^2 - \frac{l(l+1)}{r^2} \right] - V_1 + \frac{1}{2} m \omega^2 r^2 \right] g_{nl}(r) &= E_{nl}^{\text{HO}} g_{nl}(r) \\ &= \left[-V_1 + \left(2n + l + \frac{3}{2} \right) \hbar\omega \right] g_{nl}(r) \end{aligned}$$

Wynik działania pierwszego wyrazu h_{ij} na funkcje g_{nl} możemy “wziąć” z równania oscylatora! W ten sposób unikamy dwukrotnego różniczkowania przy obliczaniu elementów macierzowych. Stosując ten trick dostajemy następujący wzór na elementy macierzy M:

$$\langle \nu' | h_{ij} | \nu \rangle = \int_0^\infty g_{\nu'1}(r) g_{\nu1}(r) \left[\frac{\hbar\omega}{2} \left(4\nu + 2l + 3 - \frac{r^2}{b^2} \right) + U_{\text{WS}}(r) + V_{\text{C}}(r) + V_{\text{LS}}(r) \cdot \frac{\hbar^2}{2} \left[j(j+1) - l(l+1) - \frac{3}{4} \right] \right] r^2 dr,$$

gdzie

$$b^2 = \frac{(\hbar c)^2}{mc^2 \cdot \hbar\omega}, \quad \hbar\omega = (45 A^{-1/3} - 25 A^{-2/3}) \text{ MeV}.$$

Całki te łatwo obliczymy numerycznie. Gdy już utworzymy macierz M, to rozwiązujemy jej problem własny (diagonalizujemy). Wartości własne są szukanymi energiami, a wektory własne to współczynniki rozkładu A_ν , które zadają postać funkcji falowej. Przybliżenie tej metody polega na tym, że musimy zadać skończony wymiar tej macierzy (przy nieskończonym wynik byłby dokładny!), a to jest ograniczone mocą obliczeniową.

Funkcje falowe oscylatora harmonicznego

Musimy jeszcze jawnie wypisać postać funkcji $g_{nl}(r)$. Oto ona:

$$g_{nl}(r) = \sqrt{\frac{2n!}{b^3 \Gamma(n + l + \frac{3}{2})}} \left(\frac{r}{b} \right)^l e^{-r^2/2b^2} L_n^{(l+\frac{1}{2})} (r^2/b^2),$$

gdzie L_n^k to stowarzyszone wielomiany Laguerre’a. *Mathematica* je zna:

```
LaguerreL[1, 1 + 1 / 2, x]
```

$$\frac{3}{2} + 1 - x$$

Zobaczmy jak te funkcje $g_{nl}(r)$ wyglądają.

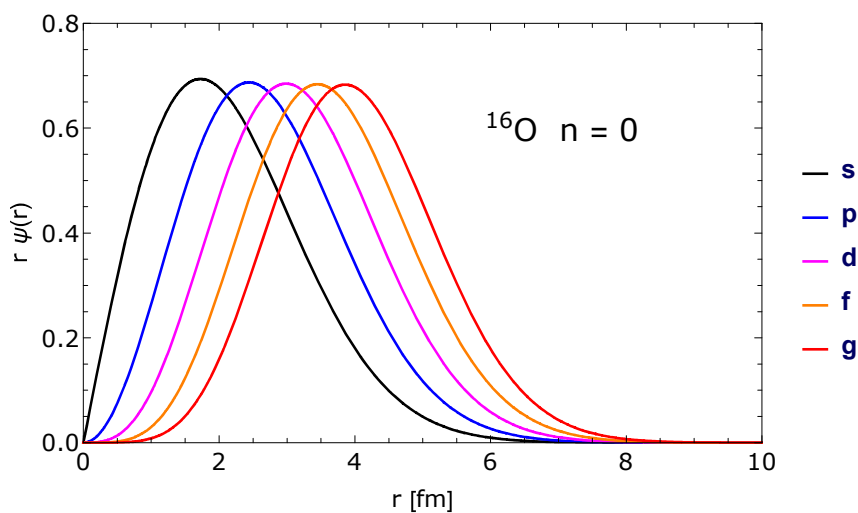
```
gnl[A_, n_, l_, r_] := Module[{b, homega},
  homega = 45. * A^(-1 / 3) - 25. * A^(-2 / 3);
  b = 197.3 / Sqrt[940.0 * homega];
  Sqrt[(2 * n! / (b^3 * Gamma[n + 1 + 3 / 2]))] * (r / b) ^ l *
  Exp[-r^2 / (2 * b^2)] * LaguerreL[n, 1 + 1 / 2, r^2 / b^2]
]
```

Rysujemy zestaw dla $n=0$ i kilku wartości l :

```

Plot[{
  gnl[16, 0, 0, r] * r,
  gnl[16, 0, 1, r] * r,
  gnl[16, 0, 2, r] * r,
  gnl[16, 0, 3, r] * r,
  gnl[16, 0, 4, r] * r
}, {r, 0, 10},
PlotRange -> {{0, 10}, {0, 0.8}},
Frame -> True,
FrameLabel -> {"r [fm]", "r ψ(r)"},
BaseStyle -> {Black, FontFamily -> "Verdana", FontSize -> 18},
PlotStyle -> {
  Directive[Black, Thick],
  Directive[Blue, Thick],
  Directive[Magenta, Thick],
  Directive[Orange, Thick],
  Directive[Red, Thick]
},
PlotLegends -> {"s", "p", "d", "f", "g"},
ImageSize -> 600,
Epilog -> {Text[Style["16O n = 0", Black, 24], {7, 0.6}]}
]

```

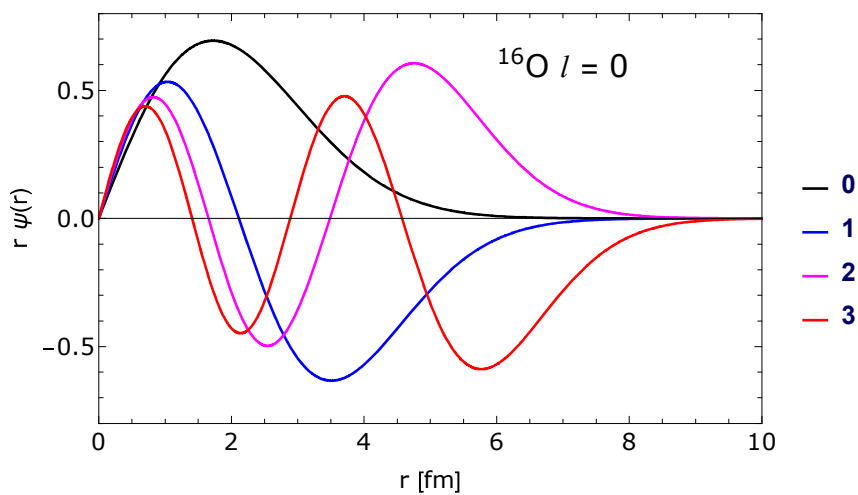


Oraz zestaw dla $l=0$ i kilku wartości n :


```

Plot[{
  gnl[16, 0, 0, r] * r,
  gnl[16, 1, 0, r] * r,
  gnl[16, 2, 0, r] * r,
  gnl[16, 3, 0, r] * r
}, {r, 0, 10},
PlotRange -> {{0, 10}, {-0.8, 0.8}},
Frame -> True,
FrameLabel -> {"r [fm]", "r ψ(r)"},
BaseStyle -> {Black, FontFamily -> "Verdana", FontSize -> 18},
PlotStyle -> {
  Directive[Black, Thick],
  Directive[Blue, Thick],
  Directive[Magenta, Thick],
  Directive[Red, Thick]
},
PlotLegends -> {"0", "1", "2", "3"},
ImageSize -> 600,
Epilog -> {Text[Style["16O l = 0", Black, 24], {7, 0.6}]}
]

```



Powinniśmy też sprawdzić regułę ortonormalności względem n :

$$\int_0^{\infty} g_{n'l}(r) g_{n'l}(r) r^2 dr = \delta_{nn'}$$

```

NIntegrate[gnl[16, 1, 2, x] * gnl[16, 1, 2, x] * x^2, {x, 0, Infinity}]

```

1.

Manipulacja 4: radialne funkcje HO

Dobrze też zobaczyć przebieg funkcji $g_{nl}(r)$ dla dowolnych parametrów, w tym A . Warto też przypomnieć, że metodą “strzelania” wyznaczaliśmy nie ostateczną funkcję radialną, tylko $u_{nlj}(r) = r R_{nlj}(r)$. Dla lepszego porównania i uwidocznienia różnicy rysujemy oba warianty: g_{nl} i $r g_{nl}$.

```
Manipulate[
  Plot[gnl[A, n, l, r] * If[fselect, r, 1], {r, 0, 20},
    PlotRange -> {{0, 20}, All},
    Frame -> True,
    Axes -> {True, False},
    GridLines -> Automatic,
    GridLinesStyle -> Dashed,
    Filling -> Axis,
    FrameLabel -> {"r [fm]", "f(r)"},
    BaseStyle -> {Black, FontFamily -> "Verdana", FontSize -> 20},
    PlotStyle -> {If[fselect, Red, Blue], Thickness[0.005]},
    ImageSize -> 600],
  {{fselect, True, Style["f(r) =", 20]},
   {True -> Style[" r gnl(r) ", Red, 20], False -> Style[" gnl(r) ", Blue, 20]},
  ControlType -> SetterBar, Background -> Gray,
  LabelStyle -> Directive[Magenta, 18]},
  Spacer[20],
  {{A, 40, Style["A", 20]}, 10, 240, 1,
```

```
  Appearance -> "Labeled", LabelStyle -> Directive[Blue, 20]},
  Spacer[30],
  {{n, 0, Style["n", 20]},
   {0 -> Style[" 0 ", 20], 1 -> Style[" 1 ", 20], 2 -> Style[" 2 ", 20],
    3 -> Style[" 3 ", 20], 4 -> Style[" 4 ", 20]}, ControlType -> SetterBar,
   Background -> Orange, LabelStyle -> Directive[Magenta, 18]},
  Spacer[40],
  {{l, 0, Style["l", 20]}, {0, 1, 2, 3, 4, 5, 6}, ControlType -> RadioButtonBar,
   Appearance -> "Labeled", LabelStyle -> Directive[Magenta, 20]},
  SaveDefinitions -> True,
  FrameLabel -> {None, None,
   Style["Radialna część funkcji falowej oscylatora harmonicznego",
    Blue, 24], None}
]
```

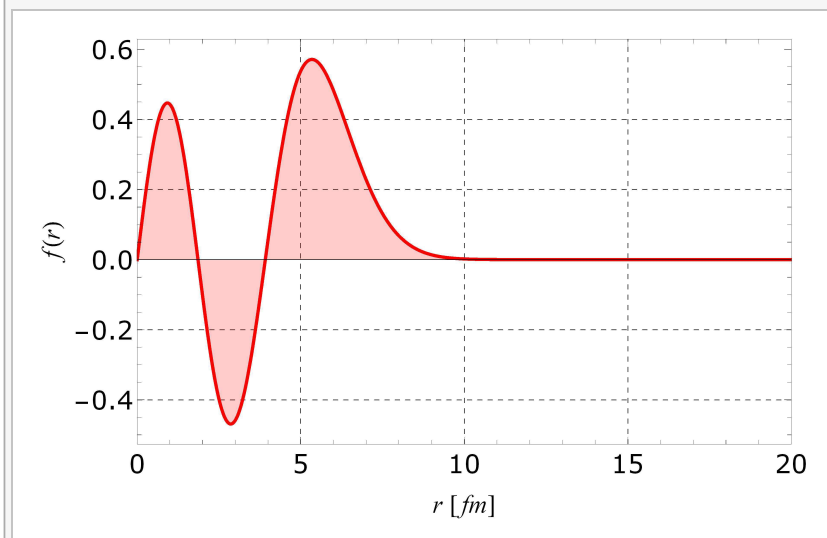
Radialna część funkcji falowej oscylatora harmonicznego

$$f(r) = r g_n(r) \quad g_n(r)$$

A

n

l 0 1 2 3 4 5 6



Rozwinięcie w bazie HO

Budujemy procedurę, która dla zadanych wartości N, Z, j i l oraz rodzaju cząstek (neutrony/protony) oblicza elementy macierzy M (\mathbf{mtx}), a następnie rozwiązuje problem własny.

```
matrixHO[n_, Z_, iso_, j_, l_] := Module[
  (* liczba neutronów, protonów, iso: +1=protony, -1=neutrony *)
  {
    (* Wartości parametrów potencjału WS (Suhonen) *)
    V0 = 51.0 (* MeV *),
    R0 = 1.27 (* fm *),
    (* R0SO = 1.16 *) (* fm *),
    a = 0.67 (* fm *),
    kappa = 33.0 / 51.0,
    lambda = 0.44,
    hc = 197.3,
    A, R, RSO, mred, isocor, homega, b, V1, VWS, hWS, VC, VLS, gn1, mtx
  },
  A = n + Z;
  mred = 940.0 (* (A-1)/A*931.49 *);
  R = R0 * A^(1/3);
  RSO = R (* R0SO*A^(1/3) *);
  (* isocor=If[n==Z,3,If[n>Z,iso*(n-Z+1)+2,iso*(n-Z-1)+2]] *);
  homega = 45. * A^(-1/3) - 25. * A^(-2/3);
```

```

b = hc / Sqrt[mred * homega];
(* potencjał WS bez centryfugi *)
hWS[r_] := Module[{},
  V1 = V0 * (1 + iso * kappa * (n - Z) / A);
  VWS = -V1 / (1 + Exp[(r - R) / a]);
  VC = If[iso == 1, 1, 0] * 1.44 * Z * If[r < R, (3 - (r / R)^2) / (2 * R), 1.0 / r];
  VLS = -lambda * V1 * R0^2 / (2 * a * r) * Exp[(r - RSO) / a] /
    ((Exp[(r - RSO) / a] + 1)^2) * (j * (j + 1) - 1 * (1 + 1) - 3 / 4);
  VWS + VC + VLS
];
(* rozwiązanie dla HO *)
gnl[nu_, r_] := Module[{},
  Sqrt[(2 * nu! / (b^3 * Gamma[nu + 1 + 3 / 2]))] * (r / b)^1 *
  Exp[-r^2 / (2 * b^2)] * LaguerreL[nu, 1 + 1 / 2, r^2 / b^2]
];
(* obliczenie elementu macierzowego *)
mtx = Table[NIntegrate[
  gnl[f, r] * gnl[i, r] * (homega / 2 * (4 * i + 2 * l + 3 - (r / b)^2) + hWS[r]) *
  r^2, {r, 0, Infinity}], {i, 0, 5}, {f, 0, 5}];
Eigensystem[mtx]
]

```

```

sol = matrixHO[82, 54, 1, 7 / 2, 4]
(* Select[Sort[sol[[1]]], # < 2.0 &] *)

```

```

{{63.2818, 45.1972, 30.1082, 17.4722, 12.2519, -7.45325},
 {0.00327987, 0.0644804, -0.0515152, 0.016693, 0.603277, 0.793067},
 {-0.0591491, 0.0670764, -0.0138548, -0.733728, -0.531419, 0.413579},
 {0.0441988, -0.112784, 0.734828, 0.435435, -0.378457, 0.335442},
 {-0.0507807, 0.302891, 0.665152, -0.449288, 0.420132, -0.291343},
 {-0.139817, 0.928709, -0.12036, 0.259637, -0.182919, 0.050931},
 {0.986108, 0.15614, -0.0164084, -0.0499067, -0.0212196, -0.000647033}}

```

```
sol[[1]]
```

```
{44.4801, -26.9865, 26.5569, 13.0764, -9.21279, 3.92101}
```

```
sol[[2, 5]]
```

```
{-0.0960479, -0.992663, 0.0185736, 0.0152077, 0.0694135, 0.000265533}
```

Porównanie metody “strzelania” z diagonalizacją “HO” dla ^{136}Xe : energie stanów

```
neutronsWSLS = {-38.81, -33.91, -33.27, -28.26, -26.72, -25.65, -22.00,
  -19.20, -18.34, -17.21, -15.25, -10.94, -10.89, -9.088, -8.83};
```

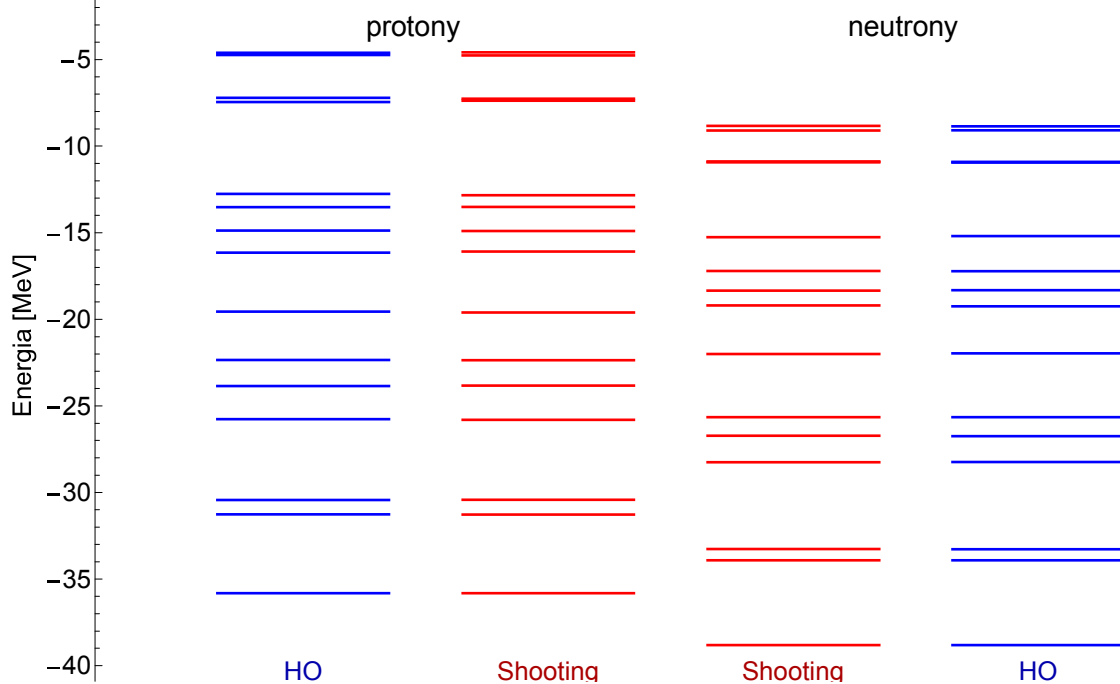
```
neutronsHO = {-38.81, -33.91, -33.28, -28.24, -26.74, -25.65, -21.96,
  -19.24, -18.32, -17.22, -15.19, -10.95, -10.91, -9.08, -8.86};
```

```
protonsWSLS = {-35.82, -31.27, -30.42, -25.80, -23.83, -22.36, -19.61,
  -16.09, -14.91, -13.51, -12.84, -7.38, -7.26, -4.76, -4.58};
```

```
protonsHO = {-35.82, -31.26, -30.43, -25.77, -23.86, -22.35, -19.56,
  -16.14, -14.87, -13.52, -12.76, -7.45, -7.21, -4.74, -4.61};
```

```
g1PHO = Graphics[Table[{Blue, Thick, Line[{{0, #}, {7, #}]}] & /@protonsHO];
g1PWS =
  Graphics[Table[{Red, Thick, Line[{{10, #}, {17, #}]}] & /@protonsWSLS];
g1NWS = Graphics[Table[{Red, Thick, Line[{{20, #}, {27, #}]}] & /@
  neutronsWSLS];
g1NHO = Graphics[Table[{Blue, Thick, Line[{{30, #}, {37, #}]}] & /@
  neutronsHO];
```

```
Show[g1PHO, g1PWS, g1NWS, g1NHO,
  Frame -> {{True, True}, {False, False}},
  FrameLabel -> {None, "Energia [MeV]},
  FrameTicks -> {{All, All}, {None, None}},
  PlotRange -> {{-5, 42}, {-40.9, -1}},
  LabelStyle -> Directive[FontFamily -> "Helvetica", FontSize -> 20],
  AspectRatio -> 0.6,
  PlotLabel -> Style["Stany jednocząstkowe dla  $^{136}\text{Xe}$ ", Darker[Blue], 24],
  ImageSize -> 1000,
  Epilog -> {
    Text[Style[" protony ", Black, 22], {8, -3}],
    Text[Style[" neutrony ", Black, 22], {28, -3}],
    Text[Style[" HO ", Darker[Blue], 20], {3.5, -40.3}],
    Text[Style[" Shooting ", Darker[Red], 20], {13.5, -40.3}],
    Text[Style[" Shooting ", Darker[Red], 20], {23.5, -40.3}],
    Text[Style[" HO ", Darker[Blue], 20], {33.5, -40.3}]
  }
]
```

Stany jednocząstkowe dla ^{136}Xe Porównanie metody “strzelania” z diagonalizacją “HO” dla ^{136}Xe : funkcje falowe

Przycisk "Zapisz" w Manipulacji 3 (slajd 11) rejestrował m.in. postać funkcji falowej w zależności od r w fm i znormalizowanej. Wyznaczając stany dla ^{136}Xe , zapisałem na dysku 3 funkcje falowe. Porównany je teraz z wynikiem drugiej metody.

Wczytujemy dane z dysku, przykładowe funkcje z metody “strzelania”:

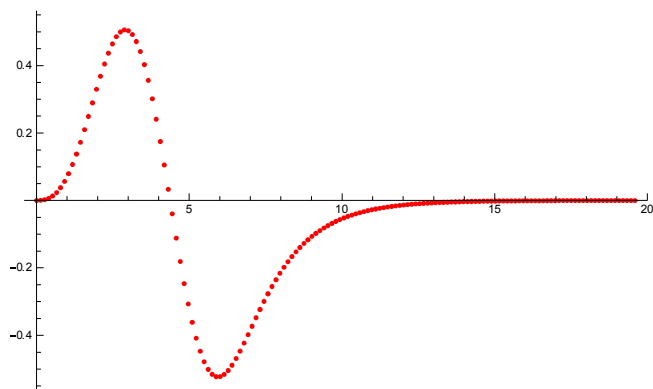
```
SetDirectory["d:/Pfutzner/Teach/Mathematica4Nuclear/05_ModelPowlokowy"]
d:\Pfutzner\Teach\Mathematica4Nuclear\05_ModelPowlokowy

FileNames["Wave*.dat"]
{Wave_136Xe_n1d32.dat, Wave_136Xe_p0f72.dat, Wave_136Xe_p2s12.dat}

n1d32 = Import["Wave_136Xe_n1d32.dat", "Table"] (*  $\nu 1 d_{3/2}$  *);
p0f72 = Import["Wave_136Xe_p0f72.dat", "Table"] (*  $\pi 0 f_{7/2}$  *);
p2s12 = Import["Wave_136Xe_p2s12.dat", "Table"] (*  $\pi 2 s_{1/2}$  *);
```

Wykresy dla stanu $\nu 1 d_{3/2}$

```
gSHOOTn1d32 = ListPlot[n1d32,
  PlotStyle -> {Red, PointSize[0.008]},
  ImageSize -> 500]
```



Jeszcze raz stosujemy metodę HO, tym razem zwracając uwagę na współczynniki rozkładu.

```
sol = matrixHO[82, 54, -1, 3/2, 2]
```

```
{{44.6716, 26.775, -26.7424, 13.2572, -8.86185, 4.06218},
 {0.00633027, -0.0425473, -0.0190819, -0.0333261, -0.565141, -0.822977},
 {-0.0132956, -0.0475436, 0.0242284, -0.64404, -0.615696, 0.450675},
 {-0.993778, 0.0860501, 0.0671777, 0.0185225, -0.00829011, -0.00870762},
 {-0.0465857, 0.0243457, -0.517618, -0.659986, 0.463374, -0.28109},
 {-0.0857532, -0.993731, 0.0105814, 0.00964397, 0.0702576, 0.00183351},
 {-0.051618, -0.0207382, -0.852347, 0.384818, -0.286031, 0.201274}}
```

Stan $1d_{3/2}$ jest na 5-tej pozycji. Współczynniki rozkładu dla niego:

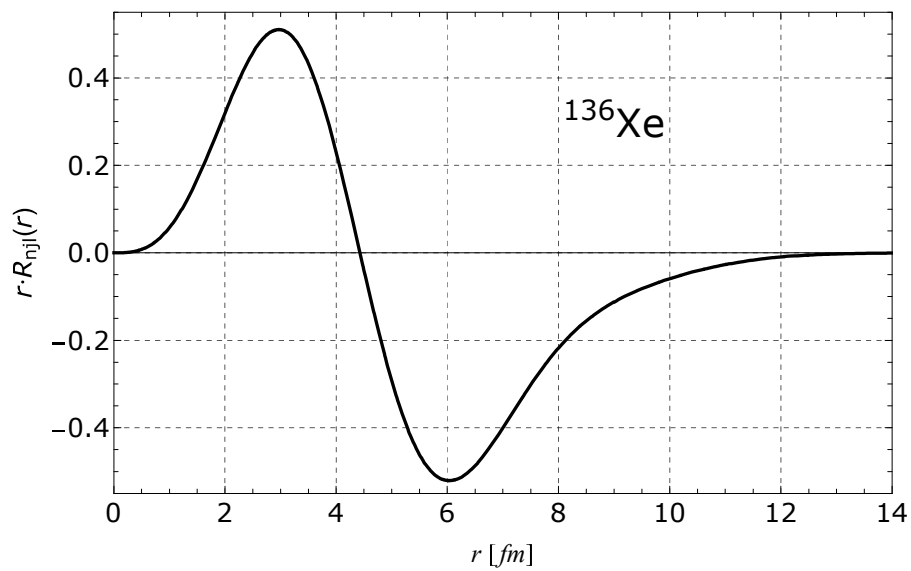
```
coefn1d32 = sol[[2, 5]]
```

```
{-0.0857532, -0.993731, 0.0105814, 0.00964397, 0.0702576, 0.00183351}
```

Obliczamy funkcje składowe (aby je narysować) oraz ich sumę, która reprezentuje funkcję falową. Pamiętajmy, że w metodzie “strzelania” wyznaczaliśmy $u_{nlj}(r) = r R_{nlj}(r)$, podczas gdy metoda HO daje $R_{nlj}(r)$. Mnożymy więc rozwiązania metody HO przez r . Okazuje się, że trzeba jeszcze dodać czynnik -1 (konwencja fazy).

```
tabHOn1d32 = Table[{-gnl[136, n, 2, r] * r * coefn1d32[[n+1]]}, {n, 0, 5}];
solHOn1d32 = Sum[-gnl[136, n, 2, r] * coefn1d32[[n+1]] * r, {n, 0, 5}];
```

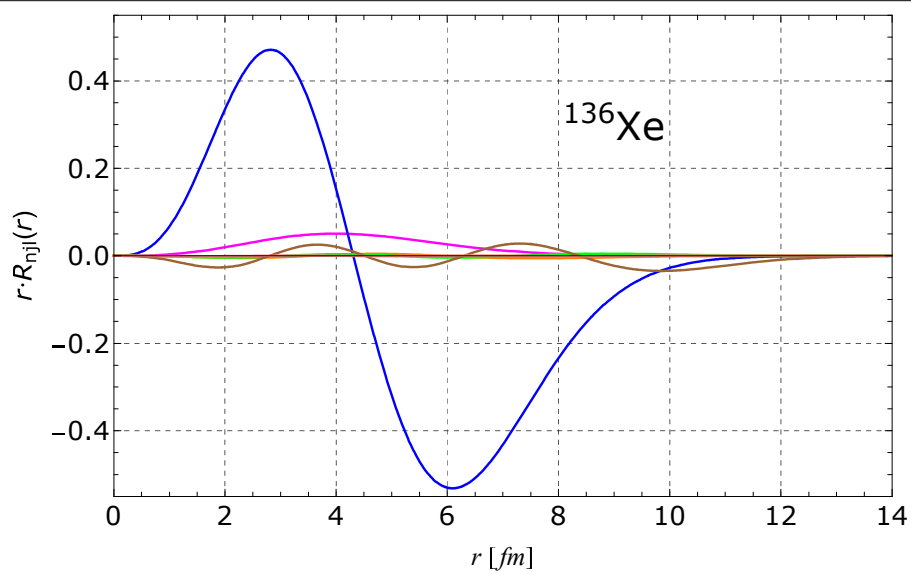
```
gHOn1d32Sum = Plot[solHOn1d32, {r, 0, 20},
  Frame -> True,
  PlotRange -> {{0, 14}, {-0.55, .55}},
  FrameLabel -> {"r [fm]", "r · Rnj1(r)"},
  PlotStyle -> Directive[Black, Thickness[0.004]],
  BaseStyle -> {Black, FontFamily -> "Verdana", FontSize -> 20},
  GridLines -> Automatic,
  GridLinesStyle -> Dashed,
  ImageSize -> 700,
  Epilog -> {Text[Style["136Xe", Black, 30], {9, 0.3}]}
]
```



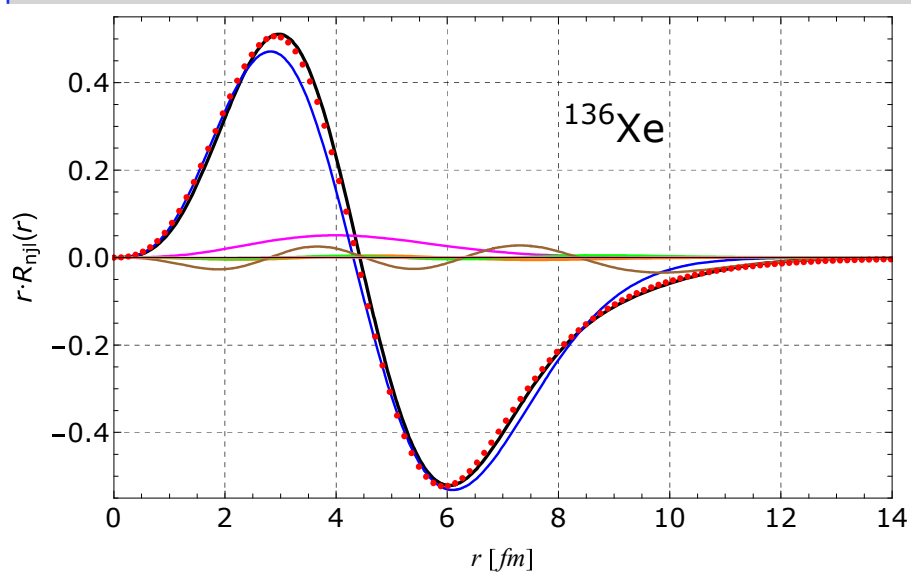
```

gHOn1d32 = Plot[tabHOn1d32, {r, 0, 20},
  Frame -> True,
  PlotRange -> {{0, 14}, {-0.55, .55}},
  FrameLabel -> {"r [fm]", "r·Rnj1(r)"},
  PlotStyle -> {Directive[Magenta, Thickness[0.003]],
    Directive[Blue, Thickness[0.003]],
    Directive[Orange, Thickness[0.003]],
    Directive[Green, Thickness[0.003]],
    Directive[Brown, Thickness[0.003]],
    Directive[Pink, Thickness[0.003]]},
  BaseStyle -> {Black, FontFamily -> "Verdana", FontSize -> 20},
  GridLines -> Automatic,
  GridLinesStyle -> Dashed,
  ImageSize -> 700,
  Epilog -> {Text[Style["136Xe", Black, 30], {9, 0.3}]}
]

```

```
Show[gHOn1d32Sum, gHOn1d32, gSHOOTn1d32]
```



To samo dla $\pi 0 f_{7/2}$

... i dla $\pi 2 s_{1/2}$

Zestawienie funkcji falowych

```

Manipulate[
  TabView[{
    {1, Style["  $\nu$  1d3/2 ", Black, 22] → If[gselect,
      Show[gHOn1d32Sum, gSHOOTn1d32], Show[gHOn1d32Sum, gHOn1d32]}
    },
    {2, Style["  $\pi$  0f7/2 ", Black, 22] → If[gselect,
      Show[gHO0f72Sum, gSHOOTp0f72], Show[gHO0f72Sum, gHO0f72]}
    },
    {3, Style["  $\pi$  2s1/2 ", Black, 22] → If[gselect,
      Show[gHO2s12Sum, gSHOOTp2s12], Show[gHO2s12Sum, gHO2s12]}
    }
  ], Dynamic[t]],
  Delimiter,
  {{gselect, True, Style[" Wyświetl też ", 20]},
  {True → Style[" wyniki Shoting ", Red, 20],
  False → Style[" składowe HO ", Blue, 20]}, ControlType → SetterBar,
  Background → Gray, LabelStyle → Directive[Magenta, 18]},
  {{t, 1}, None},
  SaveDefinitions → True,
  FrameLabel → {None, None, Style[
    "Radialna postać funkcji falowej z metody oscylatora harmonicznego",
    Blue, 25], None}
]

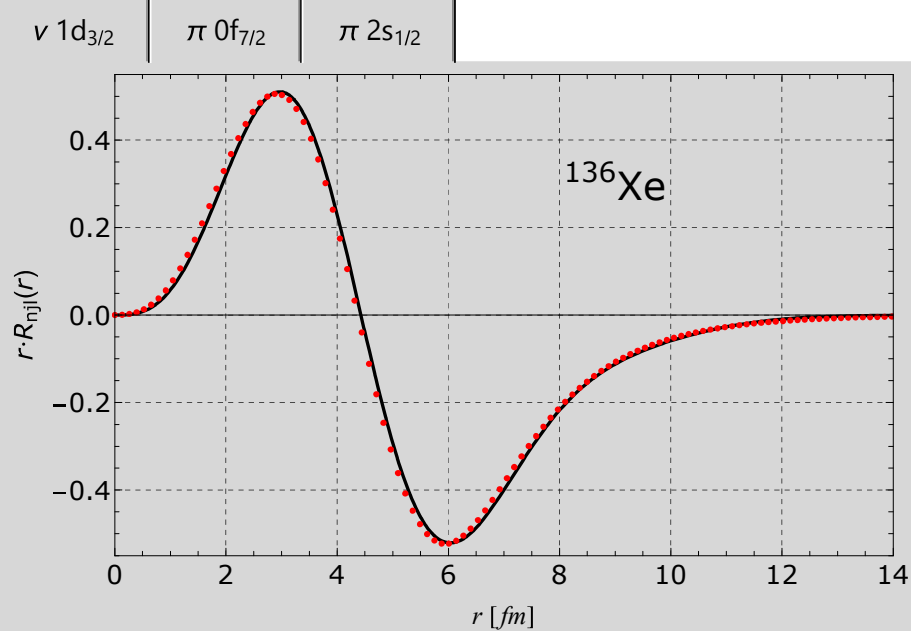
```

Radialna postać funkcji falowej z metody oscylatora harmonicznego

Wyświetl też

wyniki Shoting

składowe HO



Efekty sił reszkowych

Konfiguracje dwóch nukleonów, siły δ , mieszanie stanów

Dwa nukleony poza zamkniętą powłoką

Wyznaczenie stanów jednocząstkowych modelu powłokowego pozwala zrozumieć pojawianie się liczb magicznych, a także interpretować stany podstawowe i nisko leżące stany wzbudzone tylko dla małej liczby jąder leżących blisko jąder (podwójnie)magicznych. Gdy poza zamkniętą powłoką występuje więcej nukleonów, w grę wchodzi oddziaływanie reszkowe, czyli te, które nie są uwzględnione w polu (potencjale) średnim.

W obliczeniach struktury jądra stosuje się kilka metod opisu sił reszkowych. Ważną rolę odgrywają tzw. oddziaływania schematyczne, a wśród nich najprostszym i dogodnym do celów dydaktycznych jest oddziaływanie typu δ . Zakłada ono, że energia oddziaływania między dwoma nukleonami walencyjnymi jest:

$$V_{12}(\delta) = -V_0 \delta (\mathbf{r}_1 - \mathbf{r}_2).$$

Przyjmując oddziaływanie δ zobaczymy jak wpływa to na możliwe stany dwóch nukleonów poza zamkniętą powłoką.

Konfiguracje dwucząstkowe z siłami δ

Zagadnienie to jest dość dokładnie opisane w książce R. Castena "Nuclear Structure from a Simple Perspective" (Oxford 2000). Idziemy tu jego śladami.

Układ dwóch nukleonów na orbitalach j_1, j_2 może utworzyć kilka stanów o różnym całkowitym momencie pędu J . Gdyby nie było oddziaływań resztkowych, wszystkie te stany miałyby tę samą energię (degeneracja) równą $E_0 + \epsilon_1 + \epsilon_2$, gdzie E_0 to energia rdzenia, a ϵ_i to energia stanu jednocząstkowego. Siły resztkowe znoszą tę degenerację. W pierwszym przybliżeniu zmiana energii stanu, względem tej energii niezaburzonej wynosi:

$$\Delta E(j_1, j_2, J) = \langle j_1 j_2, J | -V_0 \delta(\vec{r}_1 - \vec{r}_2) | j_1 j_2, J \rangle.$$

Okazuje się, że te elementy macierzone można przedstawić w postaci:

$$\Delta E(j_1, j_2, J) = -V_0 F_0(n_1 l_1 n_2 l_2) \times A(j_1 j_2 J),$$

gdzie $A(j_1 j_2 J)$ jest całką kątową, która decyduje o względnym położeniu stanów, a F_0 jest całką radialną, która nie zależy od J .

- Takie same nukleony ($n-n$ lub $p-p$)

Mamy wówczas:

$$F_0(n_1 l_1 n_2 l_2) = \frac{1}{4\pi} \int \frac{1}{r^2} R_{n_1 l_1}^2(r) R_{n_2 l_2}^2(r) dr,$$

Dwa takie same nukleony mają izospin $T=1$ (f. symetryczna), co oznacza, że część przestrzenno-spinowa musi być antysymetryczna. Rozważamy 2 przypadki.

- Takie same orbity ($j_1 = j_2$):

Wtedy całkowity moment pędu J może być tylko parzysty, co wynika z symetrii współczynników Clebscha-Gordana. Całka kątowa wynosi wtedy:

$$A(j j J) = \frac{1}{2} \frac{(2j+1)^2}{2J+1} \left\langle j \frac{1}{2}, j - \frac{1}{2}, J0 \right\rangle^2, \quad J \text{ tylko parzyste.}$$

- Różne orbity ($j_1 \neq j_2$):

Wartość J może być parzysta lub nieparzysta, a całka kątowa jest:

$$A(j_1 j_2 J) = \frac{1}{2} \frac{(2j_1+1)(2j_2+1)}{2J+1} \left\langle j_1 \frac{1}{2}, j_2 - \frac{1}{2}, J0 \right\rangle^2 (1 + (-1)^{l_1+l_2+J}).$$

Zauważmy, że zmieniają się energie tylko tych stanów, dla których $l_1 + l_2 + J$ jest parzyste.

- Różne nukleony ($n-p$)

Proton i neutron mogą utworzyć układ $T=1$ lub $T=0$. Oddziaływanie δ działa tylko na symetryczną część przestrzennej funkcji falowej, ale mamy teraz dwie możliwości: $T=1, S=0$ lub $T=0, S=1$. Dla każdej z nich siła oddziaływania może być inna, dlatego przyjmujemy:

$$V_{12}(\delta) = -(V_{T=1} + V_{T=0}) \delta(\vec{r}_1 - \vec{r}_2).$$

Wtedy:

$$\begin{aligned} \Delta E(j_p, j_n, J) &= -F_0(n_p l_p n_n l_n) \times A(j_p j_n J), \\ F_0(n_p l_p n_n l_n) &= \frac{1}{8\pi} \int \frac{1}{r^2} R_{n_p l_p}^2(r) R_{n_n l_n}^2(r) dr, \\ A(j_p j_n J) &= \frac{(2j_p+1)(2j_n+1)}{2J+1} \left\langle j_p \frac{1}{2}, j_n - \frac{1}{2}, J0 \right\rangle^2 \times \\ &\times \left\{ V_{T=1} \frac{1+(-1)^{l_p+l_n+J}}{2} + V_{T=0} \left[\frac{1-(-1)^{l_p+l_n+J}}{2} + \frac{[(2j_p+1)+(-1)^{j_p+j_n+J}(2j_n+1)]^2}{4J(J+1)} \right] \right\} \end{aligned}$$

- **Ogólna formuła**

Powyższe wzory można zgrabnie zapisać przy pomocy jednego wyrażenia, które ma jeszcze i te przewagę, że pozwala obliczać niediagonalne elementy macierzowe (a to zaraz będzie potrzebne). Formuła ta jest wyprowadzona w książce: [P.J. Brussaard, P.W.M. Glaudemans "Shell-model applications in nuclear spectroscopy" \(North-Holland, 1977\)](#). Liczą oni elementy macierzowe dla oddziaływania SDI (Surface Delta Interaction), ale część kątowna jest taka sama jak dla oddziaływania δ . Aby dostać takie same wyniki jak ze wzorów powyżej, należy do wzoru BG wprowadzić dodatkowy czynnik $1/2$.

Element macierzowy dla stanów dwucząstkowych sprzężonych do momentu pędu J i izospinu T ($T=0$, lub $T=1$) dany jest:

$$\langle j_a j_b, JT | -V_T \delta(\vec{r}_1 - \vec{r}_2) | j_c j_d, JT \rangle = -V_T F_T A(j_a, j_b, j_c, j_d, J, T),$$

gdzie funkcję kątowną A można obliczyć przy pomocy procedury:

```
tbmeBG[la_, ja_, lb_, jb_, lc_, jc_, ld_, jd_, J_, T_] :=
Module[{val, dab, dcd},
dab = If[(la == lb) && (ja == jb), 1, 0];
dcd = If[(lc == ld) && (jc == jd), 1, 0];
val = (1 / (2 * J + 1)) *
Sqrt[(2 * ja + 1) * (2 * jb + 1) * (2 * jc + 1) * (2 * jd + 1) / ((1 + dab) * (1 + dcd))] *
((-1) ^ (jb + jd + lb + ld) * ClebschGordan[{jb, -1/2}, {ja, 1/2}, {J, 0}] *
ClebschGordan[{jd, -1/2}, {jc, 1/2}, {J, 0}] *
(1 - (-1) ^ (la + lb + J + T)) - (1 + (-1) ^ T) * ClebschGordan[{jb, 1/2},
{ja, 1/2}, {J, 1}] * ClebschGordan[{jd, 1/2}, {jc, 1/2}, {J, 1}]);
-val / 2.
]
```

Manipulacja 1: konfiguracje dwuczastkowe

Rozważamy jądro, które składa się z parzysto-parzystego (sferycznego) rdzenia i dwóch dodatkowych nukleonów walencyjnych. Zakładamy, że rdzeń ma spin i parzystość 0^+ , a spiny i parzystości stanów jądra wynikają ze sprzężenia obydwu nukleonów walencyjnych. Jako oddziaływanie resztkowe zakładamy siły typu δ . Energie stanów wyrażone są w jednostkach $F_0 V_0$, gdzie F_0 jest wartością całki radialnej (nie zależy od J) a V_0 to stała sprzężenia. W przypadku różnych nukleonów, oddziaływanie może zachodzić w stanach $S=0$ ($T=1$) i $S=1$ ($T=0$), potrzebne są zatem dwie stałe sprzężenia. Względny wkład tych dwóch składowych regulujemy zmieniając stosunek stałych sprzężenia $V_{S=0}/V_{S=1}$.

```
Manipulate[
  l1 = Round[o1];
  l2 = Round[o2];
  par = (-1)^(l1 + l2);
  If[par == 1, charpar = "+", charpar = "-"];
  If[o1 > l1, j1 = l1 + 1/2, j1 = l1 - 1/2];
  If[o2 > l2, j2 = l2 + 1/2, j2 = l2 - 1/2];
  If[same == 1,
    If[o1 == o2,
      lev =
        Table[{-0.5 * (2 * j1 + 1)^2 * ClebschGordan[{j1, 1/2}, {j1, -1/2}, {J, 0}]^2 /
          (2 * J + 1), J}, {J, 0, 2 * j1 - 1, 2}],
      lev = Table[{- (1 + (-1)^(l1 + l2 + J)) / 2 * (2 * j1 + 1) * (2 * j2 + 1) *
        ClebschGordan[{j1, 1/2}, {j2, -1/2}, {J, 0}]^2 / (2 * J + 1),
```

```

  J}, {J, Abs[j1 - j2], j1 + j2, 1}]
],
lev = Table[
  {-(2 * j1 + 1) * (2 * j2 + 1) * ClebschGordan[{j1, 1/2}, {j2, -1/2}, {J, 0}]^2 *
    ((1 + (-1)^(l1 + l2 + J)) / 2 + vs1 * ((1 - (-1)^(l1 + l2 + J)) / 2 +
      (2 * j1 + 1 + (-1)^(j1 + j2 + J) * (2 * j2 + 1))^2 / (4 * (J + 0.000001) *
        (J + 1))) / (2 * J + 1), J}, {J, Abs[j1 - j2], j1 + j2, 1}]
];
imax = Length[lev];
lev1 = Sort[lev];
emin = lev1[[1, 1]];
emax = lev1[[imax, 1]];
demin = (emax - emin) / 20;
graftab = {};
lastE = 10;
sx = 2.1;
For[i = 1, i <= imax, i++,
  jj = lev1[[i, 2]];
  If[EvenQ[jj], cb = 0, cb = 1];
  dsx = 0.2; dpx = 0.05;
  If[jj >= 10, dsx = 0.23; dpx = 0.1];
  e = lev1[[i, 1]];
  If[Abs[e - lastE] < demin, sx = sx + dsx, sx = 2.1];
  lastE = e;
  graftab = Append[graftab,
    {{RGBColor[cb, 0, 1], Thick, Line[{{0, e}, {2, e}}]}},

```

```

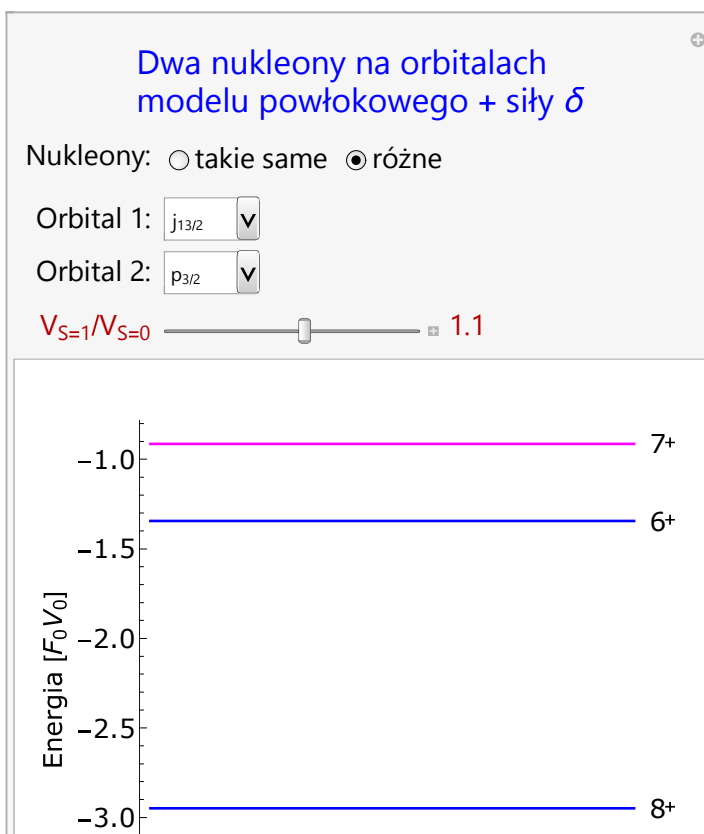
Text[Style[jj, 20], {sx, e}],
Text[Style[charpar, 16], {sx + dpx, e + demin / 5}]
]
];
g0 = Graphics[graftab, AxesOrigin -> {0, 0},
PlotRange -> {emin - demin, emax + demin}];
Show[g0, Frame -> {{True, False}, {False, False}}, FrameLabel ->
{" ", "Energia [ $F_0V_0$ ]", FrameTicks -> {{All, None}, {None, None}},
BaseStyle -> {Black, FontFamily -> "Verdana", FontSize -> 20},
AspectRatio -> Full, PlotLabel -> "", ImageSize -> 500],
Spacer[20],
{{same, 1, Style["Nukleony:", Black, 22]}},
{1 -> Style["takie same", Black, 22], 2 -> Style["różne", Black, 22]},
ControlType -> RadioButton},
Spacer[20],
{{o1, 3.1, Style["Orbital 1:", Black, 22]}},
{0.1 -> "s1/2", 0.9 -> "p1/2", 1.1 -> "p3/2", 1.9 -> "d3/2", 2.1 -> "d5/2",
2.9 -> "f5/2", 3.1 -> "f7/2", 3.9 -> "g7/2", 4.1 -> "g9/2", 4.9 -> "h9/2",
5.1 -> "h11/2", 5.9 -> "i11/2", 6.1 -> "i13/2", 6.9 -> "j13/2", 7.1 -> "j15/2"},
{{o2, 3.1, Style["Orbital 2:", Black, 22]}},
{0.1 -> "s1/2", 0.9 -> "p1/2", 1.1 -> "p3/2", 1.9 -> "d3/2", 2.1 -> "d5/2",
2.9 -> "f5/2", 3.1 -> "f7/2", 3.9 -> "g7/2", 4.1 -> "g9/2", 4.9 -> "h9/2",
5.1 -> "h11/2", 5.9 -> "i11/2", 6.1 -> "i13/2", 6.9 -> "j13/2", 7.1 -> "j15/2"},
{{vs1, 1, Style["VS=1/VS=0", Hue[0, 1, 0.7], 22]}}, 0, 2, 0.1,
Appearance -> "Labeled", LabelStyle -> Directive[Hue[0, 1, 0.7], 22],
Enabled -> If[same == 2, True, False]},
TrackedSymbols -> {same, o1, o2, vs1},

```

```

FrameLabel -> {None, None, Style["Dwa nukleony na orbitalach
modelu powłokowego + siły  $\delta$ ", Blue, 26], None}
]

```





Manipulacja 2: dodajemy wykres energii w funkcji spinu

```

Manipulate[
  l1 = Round[o1];
  l2 = Round[o2];
  par = (-1)^(l1 + l2);
  If[par == 1, charpar = "+", charpar = "-"];
  If[o1 > l1, j1 = l1 + 1/2, j1 = l1 - 1/2];
  If[o2 > l2, j2 = l2 + 1/2, j2 = l2 - 1/2];
  If[same == 1,
    If[o1 == o2,
      lev =
        Table[{-0.5 * (2 * j1 + 1)^2 * ClebschGordan[{j1, 1/2}, {j1, -1/2}, {J, 0}]^2 /
          (2 * J + 1), J}, {J, 0, 2 * j1 - 1, 2}],
      lev = Table[{- (1 + (-1)^(l1 + l2 + J)) / 2 * (2 * j1 + 1) * (2 * j2 + 1) *
        ClebschGordan[{j1, 1/2}, {j2, -1/2}, {J, 0}]^2 / (2 * J + 1),
          J}, {J, Abs[j1 - j2], j1 + j2, 1}]
    ],
    lev = Table[
      {-(2 * j1 + 1) * (2 * j2 + 1) * ClebschGordan[{j1, 1/2}, {j2, -1/2}, {J, 0}]^2 *
        ((1 + (-1)^(l1 + l2 + J)) / 2 + vs1 * ((1 - (-1)^(l1 + l2 + J)) / 2 +
          (2 * j1 + 1 + (-1)^(j1 + j2 + J) * (2 * j2 + 1))^2 / (4 * (J + 0.000001) *
            (J + 1)))) / (2 * J + 1), J}, {J, Abs[j1 - j2], j1 + j2, 1}]
    ];

```



```

imax = Length[lev];
lev1 = Sort[lev];
emin = lev1[[1, 1]];
emax = lev1[[imax, 1]];
demin = (emax - emin) / 20;
graftab = {};
glistEven = {};
glistOdd = {};
oddlevels = False;
lastE = 10;
sx = 2.1;
For[i = 1, i ≤ imax, i++,
  e = lev1[[i, 1]];
  jj = lev1[[i, 2]];
  If[EvenQ[jj],
    cb = 0;
    glistEven = Append[glistEven, {jj, e}],
    cb = 1;
    glistOdd = Append[glistOdd, {jj, e}];
    oddlevels = True];
  dsx = 0.2; dpx = 0.08;
  If[jj ≥ 10, dsx = 0.24; dpx = 0.14];
  If[Abs[e - lastE] < demin, sx = sx + dsx, sx = 2.1];
  lastE = e;
  graftab = Append[graftab,
    {{{RGBColor[cb, 0, 1], Thick, Line[{{0, e}, {2, e}}]}},

```

```

  Text[Style[jj, 20], {sx, e}],
  Text[Style[charpar, 16], {sx + dpx, e + demin / 4}]
]
];
g0 = Graphics[graftab,
  AxesOrigin → {0, emax + demin},
  PlotRange → {emin - demin, emax + demin},
  Frame → {{True, False}, {False, False}},
  FrameLabel → {" ", "Energia [F0V0]"},
  FrameTicks → {{All, None}, {None, None}},
  LabelStyle → Directive[FontFamily → "Verdana", FontSize → 20],
  AspectRatio → 1.1,
  PlotLabel → Style["Poziomy energetyczne", 20]];
If[oddlevels,
  g1 = ListPlot[{glistEven, glistOdd},
    AxesOrigin → {0, emax + demin},
    PlotRange → {emin - demin, emax + demin},
    Frame → {{True, False}, {False, True}},
    FrameTicks → {{All, None}, {None, All}},
    FrameLabel → {"", " "}, {"", "Spin"}},
    PlotStyle → {{Blue, PointSize[0.022]}, {Magenta, PointSize[0.022]}},
    Joined → False, Mesh → Full,
    BaseStyle → {FontFamily → "Verdana", FontSize → 20},
    AspectRatio → 1.1,
    Filling → Axis
  ],
  g1 = ListPlot[glistEven,

```

```

AxesOrigin -> {0, emax + demin},
PlotRange -> {emin - demin, emax + demin},
Frame -> {{True, False}, {False, True}},
FrameTicks -> {{All, None}, {None, All}},
FrameLabel -> {{ "", " " }, { "", "Spin" }},
PlotStyle -> {Blue, PointSize[0.022]},
Joined -> False, Mesh -> Full,
BaseStyle -> {FontFamily -> "Verdana", FontSize -> 20},
AspectRatio -> 1.1
]
];
GraphicsRow[{Show[g0], Show[g1]},
AspectRatio -> 1.2,
Spacings -> Scaled[0.001],
ImageSize -> 800],
Spacer[20],
{{same, 1, Style["Nukleony:", Black, 22]},
{1 -> Style["takie same", Black, 22], 2 -> Style["różne", Black, 22]}},
ControlType -> RadioButton},
Spacer[20],
{{o1, 3.1, Style["Orbital 1:", Black, 22]},
{0.1 -> "s1/2", 0.9 -> "p1/2", 1.1 -> "p3/2", 1.9 -> "d3/2", 2.1 -> "d5/2",
2.9 -> "f5/2", 3.1 -> "f7/2", 3.9 -> "g7/2", 4.1 -> "g9/2", 4.9 -> "h9/2",
5.1 -> "h11/2", 5.9 -> "i11/2", 6.1 -> "i13/2", 6.9 -> "j13/2", 7.1 -> "j15/2"}}},
{{o2, 3.1, Style["Orbital 2:", Black, 22]},
{0.1 -> "s1/2", 0.9 -> "p1/2", 1.1 -> "p3/2", 1.9 -> "d3/2", 2.1 -> "d5/2",
2.9 -> "f5/2", 3.1 -> "f7/2", 3.9 -> "g7/2", 4.1 -> "g9/2", 4.9 -> "h9/2",

```

```

5.1 -> "h11/2", 5.9 -> "i11/2", 6.1 -> "i13/2", 6.9 -> "j13/2", 7.1 -> "j15/2"}}},
{{vs1, 1, Style["VS=1/VS=0", Hue[0, 1, 0.7], 22]}, 0, 2, 0.1,
Appearance -> "Labeled", LabelStyle -> Directive[Hue[0, 1, 0.7], 22],
Enabled -> If[same == 2, True, False]},
TrackedSymbols -> {same, o1, o2, vs1},
FrameLabel -> {None, None,
Style["Dwa nukleony na orbitalach modelu powłokowego + siły δ",
Blue, 26], None}
]

```

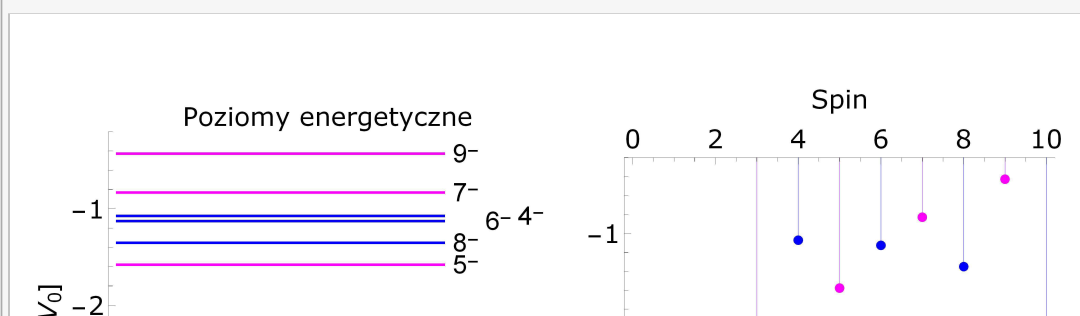
Dwa nukleony na orbitalach modelu powłokowego + siły δ

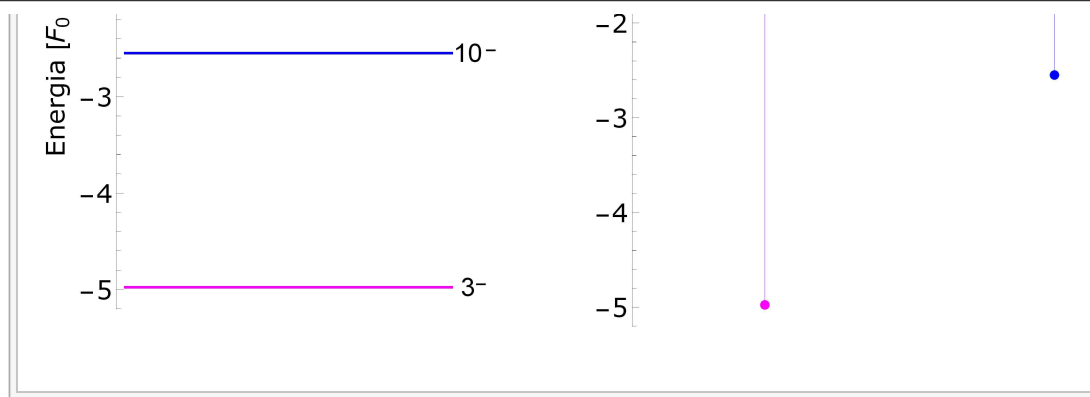
Nukleony: takie same różne

Orbital 1:

Orbital 2:

$V_{S=1}/V_{S=0}$

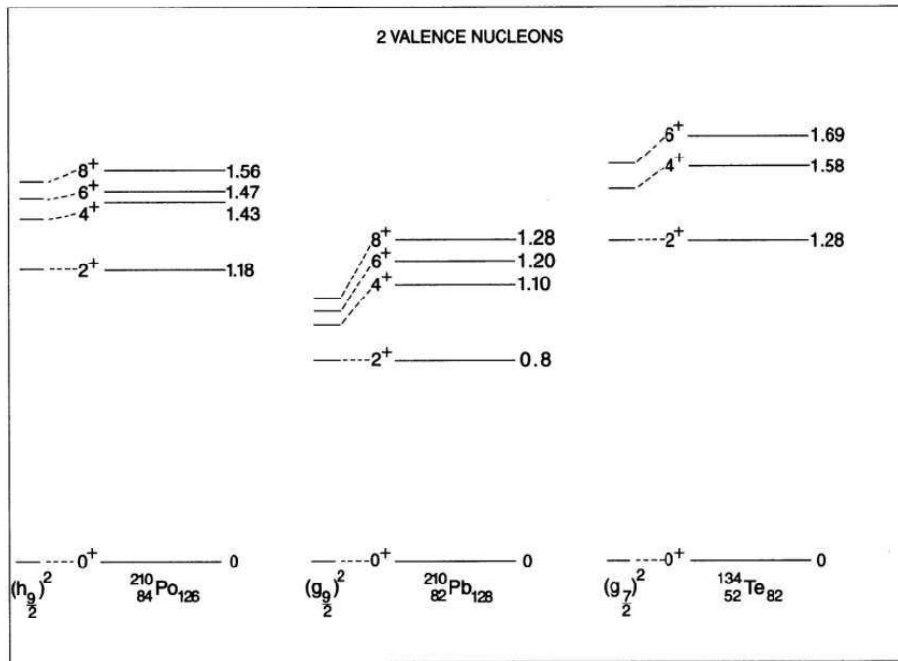




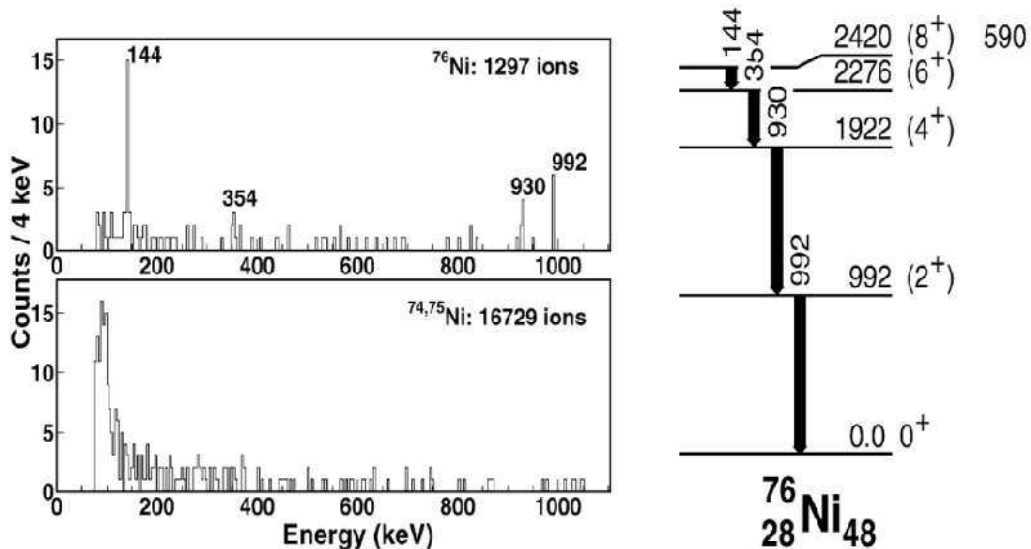
Porównanie z eksperymentem

Kilka rysunków z wykładu "Fizyka jądra atomowego I" (M.P., 2009/10).

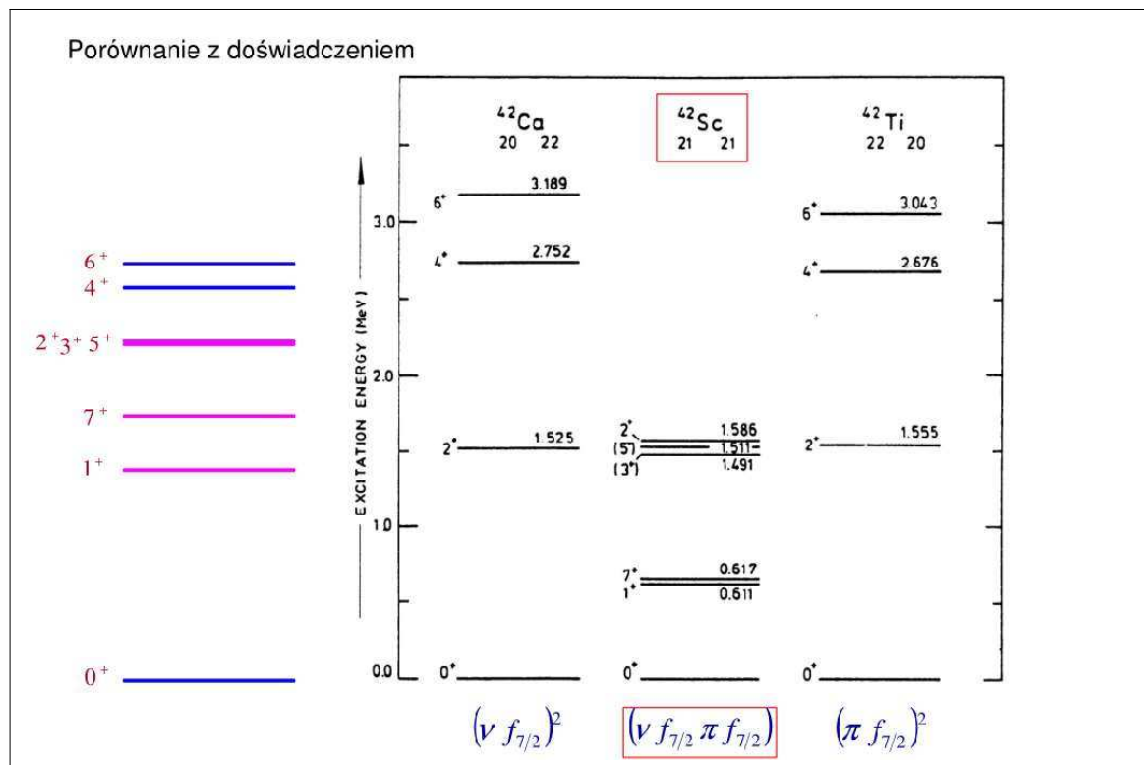
Obserwowane stany typu $(l_j)^2$



Widmo γ świadczące o deekscytacji izomeru 8^+ w ^{76}Ni i wywnioskowany układ stanów wzbudzonych.



Mazzocchi et.al, Phys. Lett. B 622 (2005) 45.



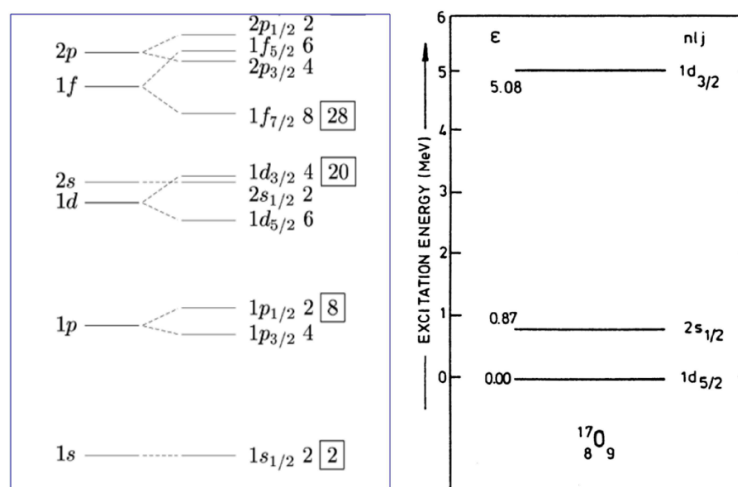
Mieszanie konfiguracji

Badaliśmy powyżej jakie stany można utworzyć z dwóch nukleonów na konkretnych orbitalach. W rzeczywistości para nukleonów może obsadzać wiele orbitali i w szczególności na wielu z nich może tworzyć stany o tym samym momencie pędu i parzystości. Wówczas będzie zachodzić zjawisko mieszania konfiguracji i stany rzeczywiste będą kombinacjami liniowymi składowych pochodzących z różnych orbitali.

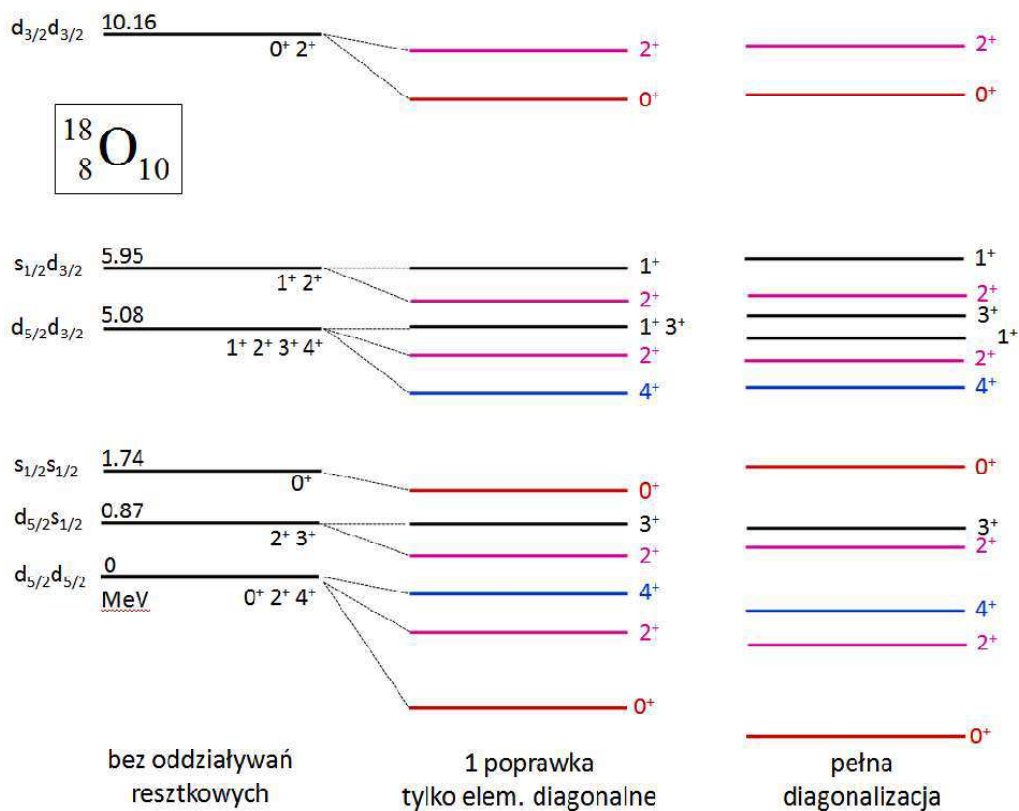
Zilustrujemy to zjawisko na względnie prostym przykładzie stanów w ^{18}O .

¹⁸O i przestrzeń modelowa

Zakładamy, że ¹⁸O można opisać jako podwójnie magiczny rdzeń ¹⁶O plus dwa neutrony walencyjne, które mogą obsadzać orbitale $d_{5/2}$, $s_{1/2}$ i $d_{3/2}$ (tzw. powłoka sd). Energie stanów jednocząstkowych można odczytać ze znanych stanów ¹⁷O.



Możliwe stany dwóch neutronów na powłoce sd :



By obliczyć energie stanów i funkcje falowe z uwzględnieniem mieszania konfiguracji musimy rozwiązać problem własny dla macierzy:

$$H_{ik} = \delta_{ik} E_k^{(0)} + \langle \psi_i^{(0)} | V_{12}(\delta) | \psi_k^{(0)} \rangle,$$

gdzie $E_n^{(0)}$ i $\psi_n^{(0)}$ to energie i funkcje falowe stanów niezaburzonych przez oddziaływanie resztkowe (lewa część rysunku powyżej). Ponieważ mieszać mogą się tylko stany o tym samym spinie i parzystości, można diagonalizować oddzielnie mniejsze macierze, dla każdego J^π .

Stany 0^+

W pełnej przestrzeni powłoki sd są 3 stany 0^+ , które mogą się mieszać.

Stany bazowe (niezaburzone) $|j_1 j_2; J, T = 1\rangle$ i ich energie w MeV, to:

- 1) $|d_{5/2} d_{5/2}; 0\rangle, E_1^{(0)} = 0,$
- 2) $|s_{1/2} s_{1/2}; 0\rangle, E_2^{(0)} = 1.74,$
- 3) $|d_{3/2} d_{3/2}; 0\rangle E_3^{(0)} = 10.16.$

Obliczając elementy macierzowe oddziaływania przyjmujemy (za BG), że we wszystkich przypadkach $V_T F_T = 1 \text{ MeV}$.

Diagonalna część macierzy H :

```
mE0 = DiagonalMatrix[{0, 1.74, 10.16}];
```

```
MatrixForm[mE0]
```

$$\begin{pmatrix} 0. & 0. & 0. \\ 0. & 1.74 & 0. \\ 0. & 0. & 10.16 \end{pmatrix}$$

Elementy oddziaływania:

```
mv11 = -1. * tbmeBG[2, 5/2, 2, 5/2, 2, 5/2, 2, 5/2, 0, 1]
```

```
-3.
```

i dalej podobnie ...

```
mv12 = -1. * tbmeBG[2, 5/2, 2, 5/2, 0, 1/2, 0, 1/2, 0, 1]
```

```
-1.73205
```

```
mv13 = -1. * tbmeBG[2, 5/2, 2, 5/2, 2, 3/2, 2, 3/2, 0, 1]
```

```
-2.44949
```

```
mv22 = -1. * tbmeBG[0, 1/2, 0, 1/2, 0, 1/2, 0, 1/2, 0, 1]
```

```
-1.
```

```
mv23 = -1. * tbmeBG[0, 1/2, 0, 1/2, 2, 3/2, 2, 3/2, 0, 1]
```

```
-1.41421
```

```
mv33 = -1. * tbmeBG[2, 3/2, 2, 3/2, 2, 3/2, 2, 3/2, 0, 1]
```

```
-2.
```

```
mV0 = {{mv11, mv12, mv13}, {mv12, mv22, mv23}, {mv13, mv23, mv33}};
```

```
MatrixForm[mV0]
```

$$\begin{pmatrix} -3. & -1.73205 & -2.44949 \\ -1.73205 & -1. & -1.41421 \\ -2.44949 & -1.41421 & -2. \end{pmatrix}$$

```
MatrixForm[mE0 + mV0]
```

$$\begin{pmatrix} -3. & -1.73205 & -2.44949 \\ -1.73205 & 0.74 & -1.41421 \\ -2.44949 & -1.41421 & 8.16 \end{pmatrix}$$

```
{val0, vec0} = Eigensystem[mE0 + mV0]
```

```
{{8.81042, -4.30779, 1.39737}, {-0.182789, -0.131504, 0.974318},  
{-0.902322, -0.371072, -0.219365}, {0.39039, -0.919245, -0.0508313}}
```


Stany 2^+

W pełnej przestrzeni powłoki sd jest 5 stanów 2^+ , które mogą się mieszać.

Stany bazowe (niezaburzone) $|j_1 j_2; J, T = 1\rangle$ i ich energie w MeV, to:

- 1) $|d_{5/2} d_{5/2}; 2\rangle, E_1^{(0)} = 0,$
- 2) $|d_{5/2} s_{1/2}; 2\rangle, E_2^{(0)} = 0.87,$
- 3) $|d_{5/2} d_{3/2}; 2\rangle, E_2^{(0)} = 5.08,$
- 4) $|s_{1/2} d_{3/2}; 2\rangle, E_2^{(0)} = 5.95,$
- 5) $|d_{3/2} d_{3/2}; 2\rangle, E_3^{(0)} = 10.16.$

Obliczając elementy macierzowe oddziaływania przyjmujemy (za BG), że we wszystkich przypadkach $V_T F_T = 1$ MeV.

Diagonalna część macierzy H :

```
mE2 = DiagonalMatrix[{0, 0.87, 5.08, 5.95, 10.16}];
```

```
MatrixForm[mE2]
```

$$\begin{pmatrix} 0. & 0. & 0. & 0. & 0. \\ 0. & 0.87 & 0. & 0. & 0. \\ 0. & 0. & 5.08 & 0. & 0. \\ 0. & 0. & 0. & 5.95 & 0. \\ 0. & 0. & 0. & 0. & 10.16 \end{pmatrix}$$

```
tbmeBG[0, 1/2, 2, 3/2, 2, 3/2, 2, 3/2, 2, 1]
```

```
-0.565685
```

```
mV2 = {{-0.686, 0.907, -0.485, 0.741, -0.524},
        {0.907, -1.2, 0.641, -0.980, 0.693},
        {-0.485, 0.641, -0.343, 0.524, -0.370},
        {0.741, -0.980, 0.524, -0.8, 0.566},
        {-0.524, 0.693, -0.370, 0.566, -0.4}};
```

```
MatrixForm[mV2]
```

$$\begin{pmatrix} -0.686 & 0.907 & -0.485 & 0.741 & -0.524 \\ 0.907 & -1.2 & 0.641 & -0.98 & 0.693 \\ -0.485 & 0.641 & -0.343 & 0.524 & -0.37 \\ 0.741 & -0.98 & 0.524 & -0.8 & 0.566 \\ -0.524 & 0.693 & -0.37 & 0.566 & -0.4 \end{pmatrix}$$

```
MatrixForm[mE2 + mV2]
```

$$\begin{pmatrix} -0.686 & 0.907 & -0.485 & 0.741 & -0.524 \\ 0.907 & -0.33 & 0.641 & -0.98 & 0.693 \\ -0.485 & 0.641 & 4.737 & 0.524 & -0.37 \\ 0.741 & -0.98 & 0.524 & 5.15 & 0.566 \\ -0.524 & 0.693 & -0.37 & 0.566 & 9.76 \end{pmatrix}$$

```
{val2, vec2} = Eigensystem[mE2 + mV2]
```

```
{{9.8892, 5.55365, 4.64123, -1.84291, 0.389827},
{-0.0355947, 0.0516161, -0.0516508, 0.0965254, 0.99201},
{0.0575859, -0.0903445, 0.477471, 0.870475, -0.0530723},
{0.115703, -0.188208, -0.865642, 0.443106, -0.0742422},
{0.721383, -0.648119, 0.135745, -0.18429, 0.0846068},
{-0.679438, -0.730543, 0.0398873, -0.051489, 0.0207191}}}
```

Stany 4^+

W pełnej przestrzeni powłoki sd są 2 stany 4^+ , które mogą się mieszać.

Stany bazowe (niezaburzone) $|j_1 j_2; J, T = 1\rangle$ i ich energie w MeV, to:

$$\begin{aligned} 1) & |d_{5/2} d_{5/2}; 4\rangle, E_1^{(0)} = 0, \\ 2) & |d_{5/2} d_{3/2}; 4\rangle, E_2^{(0)} = 5.08. \end{aligned}$$

Obliczając elementy macierzowe oddziaływania przyjmujemy (za BG), że we wszystkich przypadkach $V_T F_T = 1$ MeV.

Diagonalna część macierzy H :

```
mE4 = DiagonalMatrix[{0, 5.08}];
```

```
MatrixForm[mE4]
```

$$\begin{pmatrix} 0. & 0. \\ 0. & 5.08 \end{pmatrix}$$

```
mV4 = {{-0.286, -0.571},
{-0.571, -1.143}};
```

```
MatrixForm[mV4]
```

$$\begin{pmatrix} -0.286 & -0.571 \\ -0.571 & -1.143 \end{pmatrix}$$

```
MatrixForm[mE4 + mV4]
```

$$\begin{pmatrix} -0.286 & -0.571 \\ -0.571 & 3.937 \end{pmatrix}$$

```
{val4, vec4} = Eigensystem[mE4 + mV4]
```

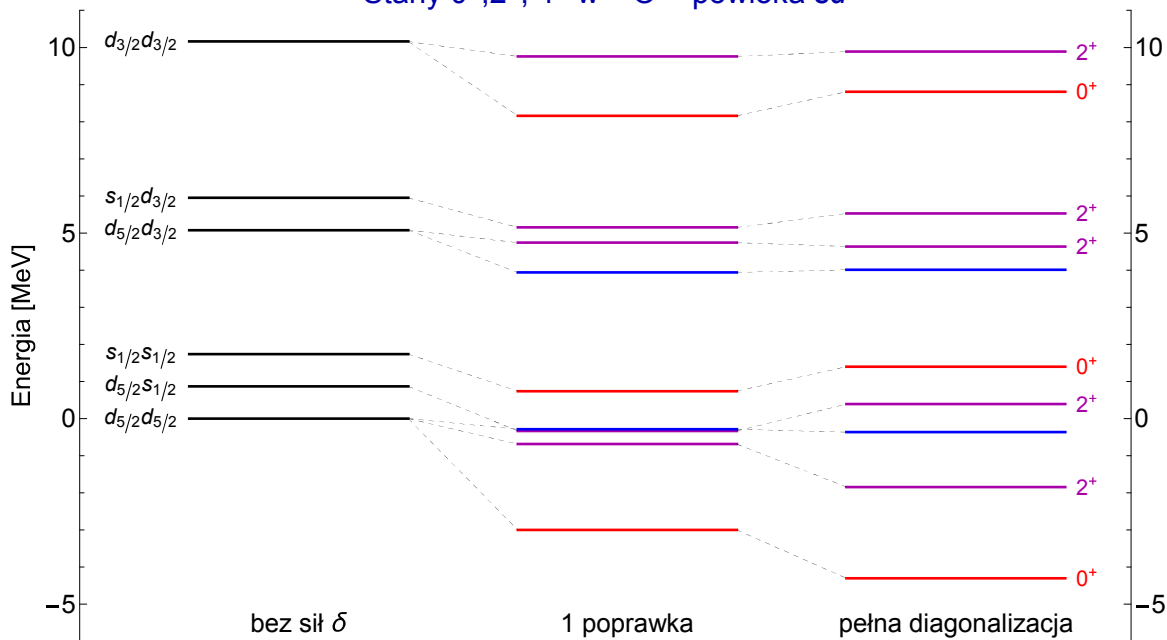
```
{{4.01284, -0.361844}, {{-0.13167, 0.991294}, {-0.991294, -0.13167}}}
```

Stany 0^+ , 2^+ , 4^+ w ^{18}O

Zobaczmy jak siły resztkowe typu δ zmieniają energie stanów 0^+ , 2^+ , 4^+ w ^{18}O i porównajmy te najniższe leżące z doświadczeniem.

Tworzemy listy danych

Rysunek energii przewidywanych:

Stany $0^+, 2^+, 4^+$ w ^{18}O – powłoka sd 

Dane doświadczalne

```
eksp = {0, 1.982, 3.555};
```

Porównanie przewidywań z doświadczeniem

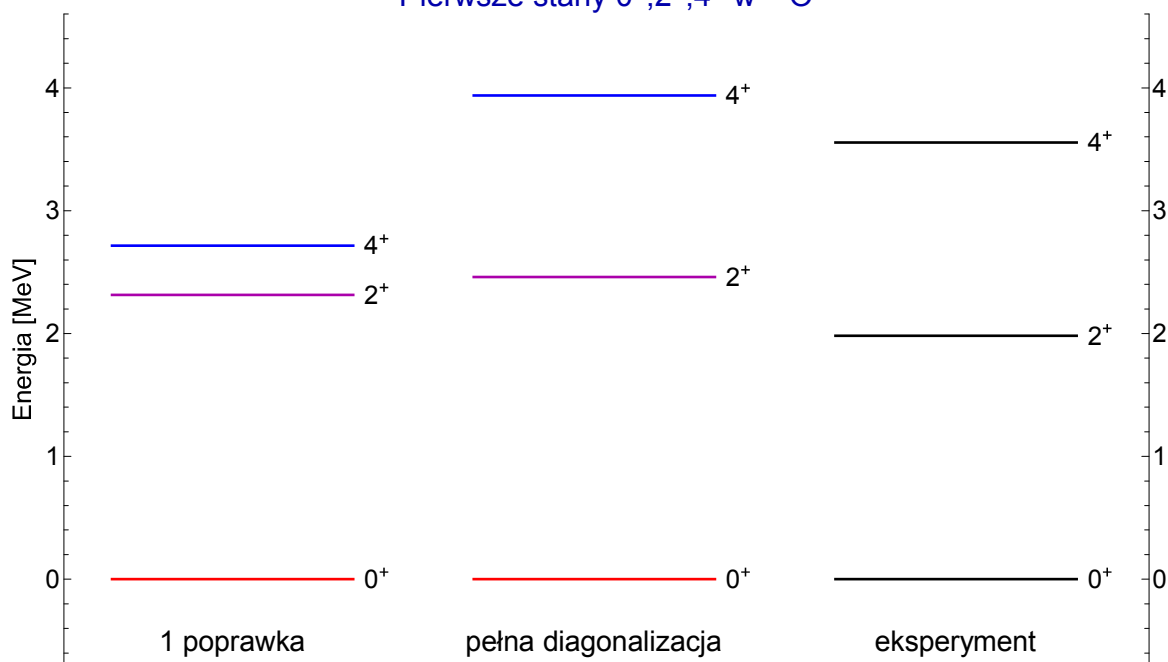
```
Show [
```

```
Graphics[{Red, Thick, Line[{{0, 0}, {10, 0}}]}],
Graphics[{Darker[Magenta], Thick, Line[{{0, dwaplust1[[1]] - zeroplust1[[1]]},
{10, dwaplust1[[1]] - zeroplust1[[1]]}}]}],
Graphics[{Blue, Thick, Line[{{0, czteryplust1[[1]] - zeroplust1[[1]]},
{10, czteryplust1[[1]] - zeroplust1[[1]]}}]}],
Graphics[{Red, Thick, Line[{{15, 0}, {25, 0}}]}],
Graphics[
  {Darker[Magenta], Thick, Line[{{15, dwaplustFull[[1]] - zeroplustFull[[1]]},
  {25, dwaplustFull[[1]] - zeroplustFull[[1]]}}]}],
Graphics[{Blue, Thick, Line[{{15, czteryplustFull[[1]] - zeroplustFull[[1]]},
{25, czteryplustFull[[1]] - zeroplustFull[[1]]}}]}],
Graphics[{Black, Thick, Line[{{30, eksp[[1]]}, {40, eksp[[1]]}}]}],
Graphics[{Black, Thick, Line[{{30, eksp[[2]]}, {40, eksp[[2]]}}]}],
Graphics[{Black, Thick, Line[{{30, eksp[[3]]}, {40, eksp[[3]]}}]}],
Frame -> {{True, True}, {False, False}},
FrameLabel -> {None, "Energia [MeV]},
FrameTicks -> {{All, All}, {None, None}},
PlotRange -> {{-2, 43}, {-0.7, 4.6}},
LabelStyle -> Directive[FontFamily -> "Helvetica", FontSize -> 20],
AspectRatio -> 0.6,
PlotLabel -> Style["Pierwsze stany  $0^+, 2^+, 4^+$  w  $^{18}\text{O}$ ", Darker[Blue], 24],
ImageSize -> 900,
Epilog -> {
  Text[Style[" 1 poprawka ", Black, 22], {5, -0.5}],
  Text[Style[" pełna diagonalizacja ", Black, 22], {20, -0.5}],
  Text[Style[" eksperyment ", Black, 22], {35, -0.5}],
```

```

Text[Style["0+", Black, 20], {11, 0}],
Text[Style["2+", Black, 20], {11, dwaplust1[[1]] - zeroplast1[[1]]}],
Text[Style["4+", Black, 20], {11, czteryplust1[[1]] - zeroplast1[[1]]}],
Text[Style["0+", Black, 20], {26, 0}],
Text[Style["2+", Black, 20], {26, dwaplustFull[[1]] - zeroplastFull[[1]]}],
Text[Style["4+", Black, 20],
  {26, czteryplustFull[[1]] - zeroplastFull[[1]]}],
Text[Style["0+", Black, 20], {41, eksp[[1]]}],
Text[Style["2+", Black, 20], {41, eksp[[2]]}],
Text[Style["4+", Black, 20], {41, eksp[[3]]}],
}]

```

Pierwsze stany $0^+, 2^+, 4^+$ w ^{18}O 

Mieszanie wielu stanów

Zjawisko mieszania stanów przez oddziaływania zaburzające, takie jak resztkowe, występuje nader często w fizyce jądrowej. Na koniec zbadamy pewien szczególny i pouczający przypadek, w którym wszystkie elementy macierzowe oddziaływania zaburzającego mają tę samą wartość.

Macierz, którą chcemy diagonalizować ma więc postać:

$$[H] = \begin{pmatrix} E_1^{(0)} & 0 & \dots & 0 \\ 0 & E_2^{(0)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & E_n^{(0)} \end{pmatrix} - V \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix}$$

Rozważymy model takiej sytuacji, w którym energie niezaburzone są równoodległe, jak w oscylatorze harmonicznym.

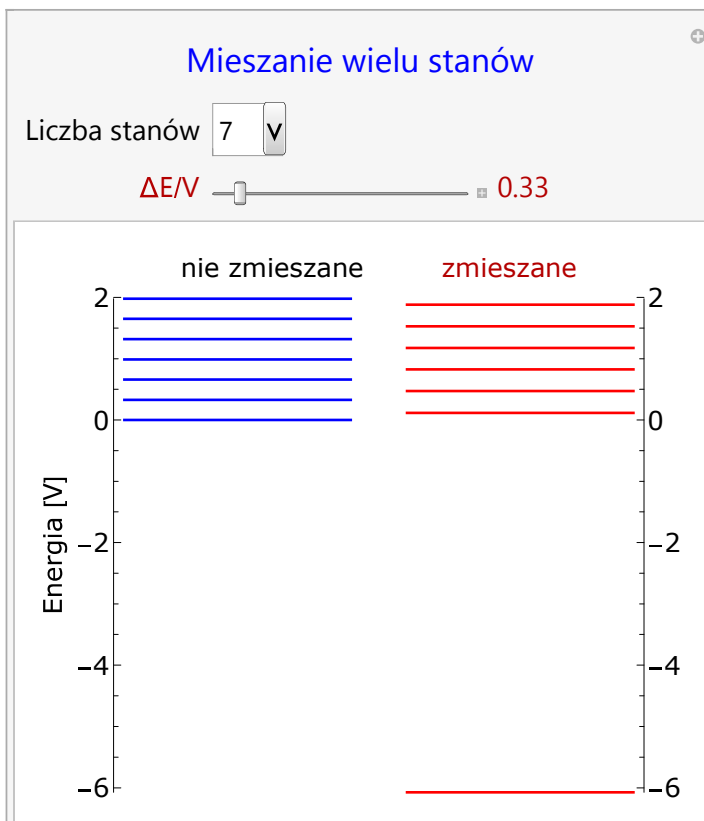
Manipulacja 4: model mieszania wielu stanów

Stany wyjściowe (o liczbie od dwóch do siedmiu), bez oddziaływań szczątkowych mają energie równo odległe od siebie, jak stany oscylatora harmonicznego. Energetyczna odległość między nimi wynosi ΔE . Zakładamy, że wartości elementów macierzowych oddziaływania resztkowego są takie same dla każdej pary stanów (diagonalnych i pozadiagonalnych) i wynoszą $-V$. Wartość V wyznacza skalę energii. Stosunek $\Delta E/V$ może się zmieniać od 0 do 5.

```

Manipulate[
  val0 = r Table[i, {i, 0, n - 1}];
  md = DiagonalMatrix[val0];
  mv = Table[1, {i, n}, {j, n}];
  g0 = Graphics[Table[{{Blue, Thick, Line[{{0.5, #}, {2.5, #}]}}] & /@ val0,
    AxesOrigin -> {0, 0}, AxesLabel -> {None, "Energia [V]"}];
  {val, vec} = Eigensystem[md - mv];
  gmix = Graphics[Table[{{Red, Thick, Line[{{3, #}, {5, #}]}}] & /@ val];
  Show[g0, gmix, Frame -> {{True, True}, {False, False}},
    FrameLabel -> {None, "Energia [V]"},
    FrameTicks -> {{All, All}, {None, None}},
    BaseStyle -> {Black, FontFamily -> "Verdana", FontSize -> 20},
    AspectRatio -> Full, PlotLabel -> Style["nie zmieszane", Black, 20]
    Style["      zmieszane", Hue[0, 1, 0.7], 20],
    ImageSize -> 500],
  Spacer[20],
  {{n, 2, Style["Liczba stanów", Black, 22]},
    {2 -> Style["2", 20], 3 -> Style["3", 20], 4 -> Style["4", 20],
    5 -> Style["5", 20], 6 -> Style["6", 20], 7 -> Style["7", 20]}},
  {{r, 0, Style[" $\Delta E/V$ ", Hue[0, 1, 0.7], 22]}, 0, 5, Appearance -> "Labeled",
    LabelStyle -> Directive[Hue[0, 1, 0.7], 22]},
  FrameLabel -> {None, None, Style["Mieszanie wielu stanów", Blue, 26], None}
]

```



Elementy optyki jonowej

Metoda macierzowa w optyce geometrycznej i prosty model separatora fragmentów

Optyka jonowa

Optyka jonowa obejmuje całość zagadnień związanych z przesyłaniem cząstek z miejsca wytworzenia do miejsca ich wykorzystania. Elementy optyczne występują w każdym laboratorium, gdzie wykorzystuje się wiązki jonów i gdzie trzeba je transportować, ogniskować, separować, itp.

Do opisu biegu jonów w dowolnym układzie jonowo-optycznym powszechnie stosuje się formalizm macierzowy. Metoda ta jest analogiczna do opisu promieni świetlnych w zwykłych układach optycznych, w których można zastosować klasyczną optykę geometryczną.

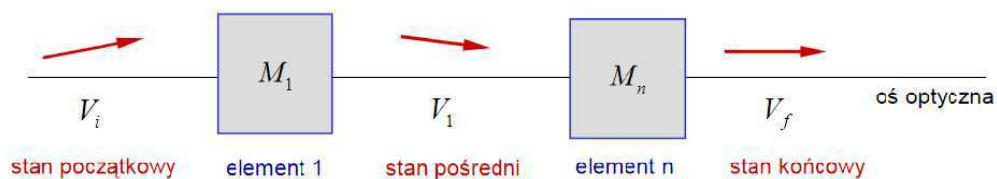
Dokładniejsze omówienie tego tematu można znaleźć w cyklu wykładów pt. [“Fragmentacja pocisków...”](#), który prowadziłem w 2004. Optyce jonowej poświęcony jest rozdział [R4](#). Tutaj powtarzam fragmenty tego wykładu i dodaję kilka interaktywnych ilustracji rozważanych problemów.

Metoda macierzowa

Główne założenia formalizmu macierzowego :

1. W każdym punkcie układu stan cząstki (promienia) opisuje wektor (tablica 1-wym.) jego parametrów zawierający wszystkie istotne zmienne, jak położenie (x,y), nachylenie toru (x', y'), pęd, itp.
2. Każdy element układu optycznego reprezentowany jest przez macierz (tablica 2-wym.), która opisuje wpływ tego elementu na stan cząstki.
3. Macierz układu złożonego z wielu elementów jest iloczynem macierzy odpowiadających tym elementom.
4. Stan cząstki po przejściu przez układ dany jest jako wynik mnożenia wektora opisującego stan początkowy cząstki przez macierz tego układu.

Zasada opisu macierzowego



$$V_i = \begin{pmatrix} x \\ x' \\ \vdots \end{pmatrix}, \quad M_1 = \begin{pmatrix} m_{11} & m_{12} & \cdots \\ m_{21} & m_{22} & \\ \vdots & & \ddots \end{pmatrix}, \quad V_1 = M_1 \times V_i, \quad \dots, \quad V_f = M_n \times V_{n-1}.$$

Działanie układu n elementów optycznych można opisać poprzez jedną macierz M :

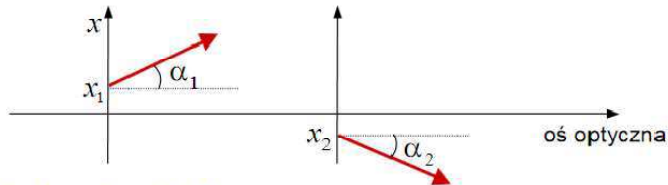
$$V_f = M \times V_i, \quad \text{gdzie} \quad M = M_n \times M_{n-1} \times \dots \times M_1.$$

Przykłady z optyki geometrycznej

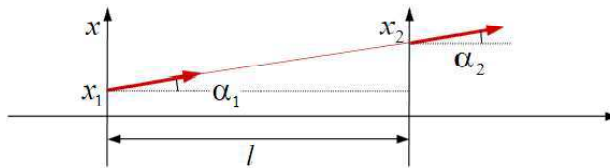
Prosty przykład :

Zastosowanie formalizmu macierzowego w optyce geometrycznej.
Stosujemy przybliżenie 1-go rzędu (liniowe).

- ▶ W każdym punkcie układu opisujemy promień świetlny podając jego odległość od osi optycznej (x) i kąt między jego kierunkiem a osią (α).



- ▶ Obszar dryfu o długości l



Łatwo zobaczyć, że :

$$x_2 = x_1 + \alpha_1 l,$$

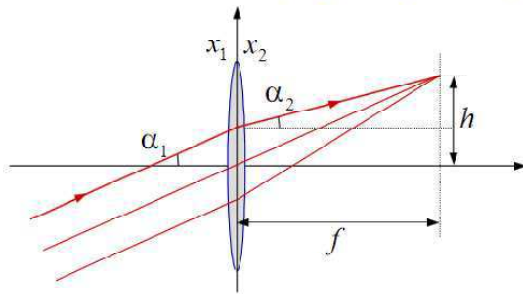
$$\alpha_2 = \alpha_1.$$

$$\begin{aligned}
 x_2 = x_1 + \alpha_1 l, \\
 \alpha_2 = \alpha_1.
 \end{aligned}
 \Rightarrow
 \begin{pmatrix} x_2 \\ \alpha_2 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & l \\ 0 & 1 \end{pmatrix}}_{M_d(l)} \begin{pmatrix} x_1 \\ \alpha_1 \end{pmatrix}.$$

$M_d(l)$ – macierz dla odcinka dryfu długości l

Cienka soczewka

► Cienka soczewka skupiająca o ogniskowej f



Zauważamy, że :

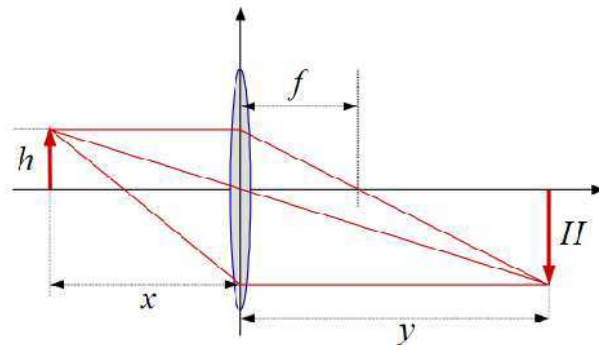
$$x_2 = x_1 \quad \text{– bo soczewka jest cienka}$$

$$\begin{aligned} h &= \alpha_1 f, \\ h &= x_1 + \alpha_2 f, \end{aligned} \quad \Rightarrow \alpha_2 = -\frac{x_1}{f} + \alpha_1.$$

Soczewce o ogniskowej f odpowiada więc macierz : $M_s(f) = \begin{pmatrix} 1 & 0 \\ -1/f & 1 \end{pmatrix}$.

Można łatwo sprawdzić, że macierz dla soczewki rozpraszającej ma taką samą postać, tylko wartość ogniskowej f trzeba wziąć ujemną.

Tworzenie obrazu przez cienką soczewkę możemy zbadać rozważając następujący układ:



Układ optyczny składa się tu z trzech elementów : dryfu o długości x , cienkiej soczewki o ogniskowej f i dryfu o długości y . Macierz odpowiadającą temu układowi konstruujemy przez złożenie elementów składowych :

$$M_{x\hat{y}} = M_d(y) \times M_s(f) \times M_d(x).$$

$$M_{xy} = M_a(y) \times M_s(f) \times M_a(x) = \begin{pmatrix} 1 & y \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -1/f & 1 \end{pmatrix} \begin{pmatrix} 1 & x \\ 0 & 1 \end{pmatrix} =$$

$$-\begin{pmatrix} 1 & y \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & x \\ -1/f & -x/f + 1 \end{pmatrix} = \begin{pmatrix} 1 - y/f & x + y - \frac{xy}{f} \\ -1/f & -x/f + 1 \end{pmatrix}.$$

Warunek utworzenia obrazu oznacza, że wartość x_2 nie może zależeć od kąta α_1 (pęk promieni wychodzących z jednego punktu przedmiotu jest skupiany też w jednym punkcie).

Warunek ten jest spełniony wtedy, gdy $m_{12} = 0$:

$$x + y = \frac{xy}{f} \Rightarrow \frac{1}{f} = \frac{x+y}{xy} = \frac{1}{x} + \frac{1}{y} !$$

Z kolei element m_{11} określa powiększenie układu :

$$\frac{H}{h} = m_{11} = 1 - \frac{y}{f} = 1 - \frac{y}{\frac{xy}{x+y}} = 1 - \frac{x+y}{x} = -\frac{y}{x} !$$

Widzimy, że znając macierz układu możemy łatwo badać jego własności analitycznie. *Math-*

ematica pozwala przedstawić to graficznie, bardziej intuicyjnie.

Manipulacja 1: obrazowanie z jedną soczewką

Ustalamy, że długość ogniskowa soczewki wynosi 1 (naturalna jednostka długości wzdłuż osi optycznej). Przedmiot o wysokości 1 umieszczamy w (zmiennej) odległości X od soczewki i śledzimy tory trzech promieni rozchodzących się z wierzchołka przedmiotu. Nie wiemy z góry gdzie powstanie obraz (nie znamy ostatniej macierzy układu), ale możemy wyznaczyć jego położenie graficznie, jako miejsce gdzie przecinają się dwa promienie. Wtedy wyznaczamy powiększenie, czyli stosunek długości obrazu do przedmiotu.

In[1]=

```
Manipulate[
Module[{
  mlens = {{1, 0}, {-sign, 1}}, (* soczewka o ogniskowej 1 *)
  rays = {{1, 0}, {1, 0.15}, {1, -0.15}},
  lens, object, image, ll, lines, mx, my, yi = 5, r1, r2, hi, im, opis,
  raytrack, findImage
},
mx = {{1, xo}, {0, 1}};
my = {{1, yi}, {0, 1}};
raytrack[r0_] := Module[{}],
  r1 = mx.r0;
  r2 = my.mlens.r1;
  {{-xo, r0[[1]]}, {0, r1[[1]]}, {yi, r2[[1]]}}
];
findImage[rays_] := Module[{ra, rb, sol, x},
  ra = rays[[2]];

```

```

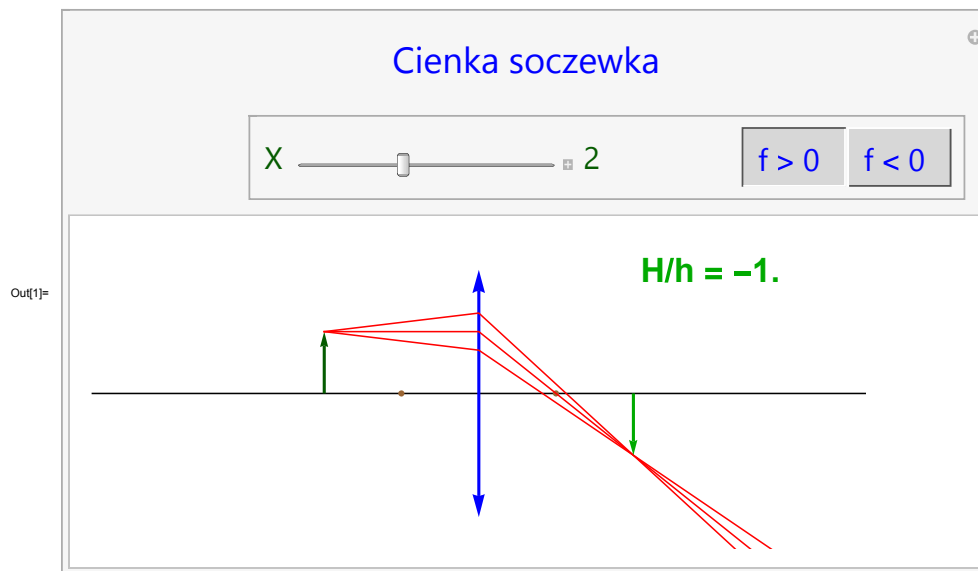
  ra = mlens.mx.ra;
  rb = rays[[3]];
  rb = mlens.mx.rb;
  sol = NSolve[ra[[1]] + ra[[2]] * x == rb[[1]] + rb[[2]] * x, x];
  hi = Infinity;
  If[Length[sol] > 0,
    im = {{x, 0}, {x, ra[[1]] + ra[[2]] * x}} /. sol;
    hi = NumberForm[Flatten[im][[4]], 3];
    image =
      Graphics[{icolor, Thickness[0.004], {Arrowheads[0.02], Arrow[im]}},
      Graphics[Point[{0, 0}]]
    ];
  lens = Graphics[{Blue, Thickness[0.004],
    {Arrowheads[{-sign * 0.03, sign * 0.03}], Arrow[{{0, -2}, {0, 2}]}]}];
  object = Graphics[{ocolor, Thickness[0.004],
    {Arrowheads[0.02], Arrow[{-xo, 0}, {-xo, 1}]}]}];
  ll = raytrack /@ rays;
  lines = Graphics[{Red, Thickness[0.002], Line[ll]}];
  image = findImage[rays];
  opis = Graphics[
    Text[Style[StringForm["H/h = `1`", hi], icolor, Bold, 26], {3, 2}]];
  Show[axis, foci, lens, object, lines, image, opis,
    PlotRange -> {{-5, 5}, {-2.5, 2.5}},
    AspectRatio -> 0.4,
    ImageSize -> 600],
  (* Kontrolki *)
  Panel[

```

```

Row[{
  Control[{{xo, 2, Style["X", ocolor, 24]}, 0, 5,
    Appearance -> "Labeled", LabelStyle -> Directive[ocolor, 24]}],
  Spacer[20],
  Control[{{sign, 1, Style[" ", 10]}, {1 -> Style[" f > 0 ", lcolor, 24],
    -1 -> Style[" f < 0 ", lcolor, 24]}]}],
}],
ImageMargins -> {{130, 0}, {0, 20}}
],
TrackedSymbols -> {xo, sign},
FrameLabel -> {None, None, Style["Cienka soczewka", Blue, 28], None},
Initialization -> {
  axis = Graphics[{Black, Thickness[0.002], Line[{{-5, 0}, {5, 0}}]}],
  lcolor = Blue,
  ocolor = Hue[0.324, 1, 0.356],
  icolor = Darker[Green],
  rcolor = Red,
  foci = Graphics[{Brown, {Disk[{-1, 0], ImageScaled[{0.004, 0.01}]},
    Disk[{1, 0], ImageScaled[{0.004, 0.01]}]}]}]
}]
}
]

```



Manipulacja 2: układ dwóch cienkich soczewek

Ciekawszy jest układ dwóch cienkich soczewek. Rozważamy przypadek soczewek o tej samej wartości ogniskowej (równej 1), ale niekoniecznie o tym samym znaku. Oprócz znaku ogniskowej każdej soczewki możemy zmieniać odległość przedmiotu od pierwszej soczewki, X , oraz odległość między soczewkami, d .

```
Manipulate[
Module[{
  rays = {{1, 0}, {1, 0.15}, {1, -0.15}},
  lens1, foci1, lens2, foci2,
  ml1, ml2, mx, my, md,
  object, image, ll, lines, yi = 8., hi, im, opis,
  raytrack, findImage
},
ml1 = {{1, 0}, {-sign1, 1}};
ml2 = {{1, 0}, {-sign2, 1}};
mx = {{1, xo}, {0, 1}};
md = {{1, d}, {0, 1}};
my = {{1, yi - d}, {0, 1}};
raytrack[r0_] := Module[{r1, r2, r3},
  r1 = mx.r0;
  r2 = md.ml1.r1;
  r3 = my.ml2.r2;
  {{-xo, r0[[1]]}, {0, r1[[1]]}, {d, r2[[1]]}, {yi, r3[[1]]}}
```

```
];
findImage[-rays_] := Module[{ra, rb, sol, m, x},
  m = ml2.md.ml1.mx;
  ra = rays[[3]];
  ra = m.ra;
  rb = rays[[2]];
  rb = m.rb;
  sol = NSolve[ra[[1]] + ra[[2]] * x == rb[[1]] + rb[[2]] * x, x];
  hi = Infinity;
  If[Length[sol] > 0,
    im = {{x + d, 0}, {x + d, rb[[1]] + rb[[2]] * x}} /. sol;
    hi = NumberForm[Flatten[im][[4]], 3];
    image =
      Graphics[{icolor, Thickness[0.004], {Arrowheads[0.02], Arrow[im]}},
      Graphics[Point[{0, 0}]]
  ];
  lens1 = Graphics[{l1color, Thickness[0.004],
    {Arrowheads[{-sign1 * 0.03, sign1 * 0.03}], Arrow[{{0, -2}, {0, 2}]}}];
  foci1 = Graphics[{l1color, {Disk[{-1, 0}, ImageScaled[{0.004, 0.01}]],
    Disk[{1, 0}, ImageScaled[{0.004, 0.01}]]}
  ];
  lens2 = Graphics[{l2color, Thickness[0.004],
    {Arrowheads[{-sign2 * 0.03, sign2 * 0.03}], Arrow[{{d, -2}, {d, 2}]}}];
  foci2 = Graphics[{l2color, {Disk[{-1 + d, 0}, ImageScaled[{0.004, 0.01}]],
    Disk[{1 + d, 0}, ImageScaled[{0.004, 0.01}]]}
  ];
  object = Graphics[{ocolor, Thickness[0.004],
```



```

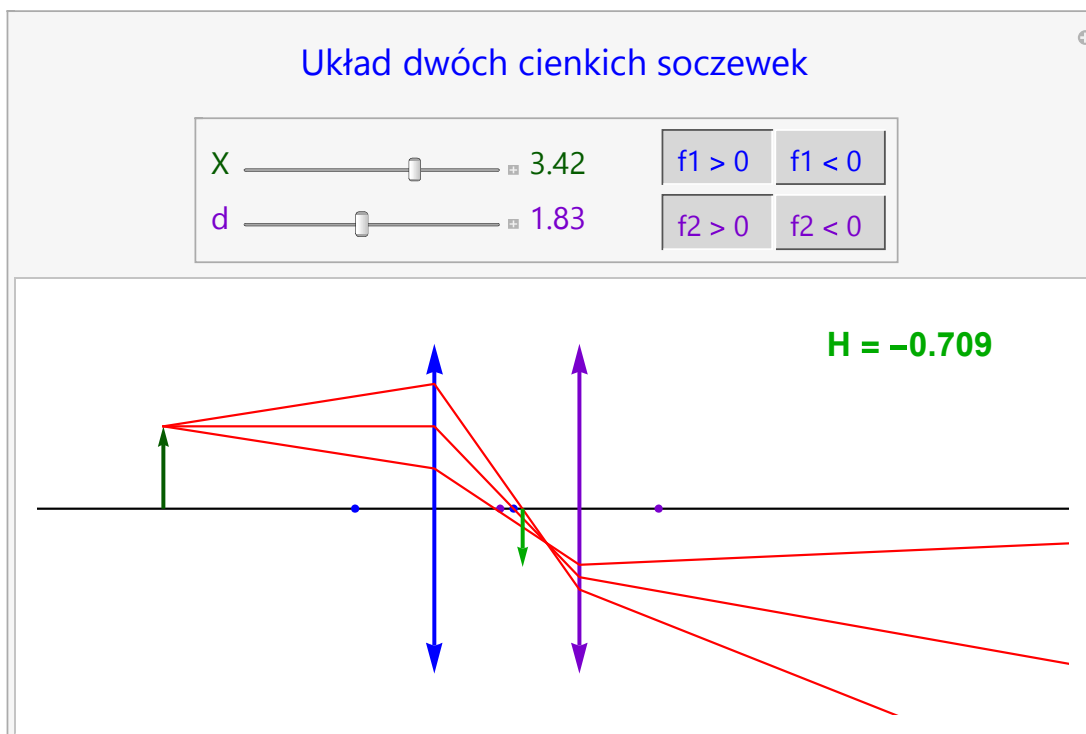
{Arrowheads[0.02], Arrow[{{-x0, 0}, {-x0, 1}}]}}];
ll = raytrack /@ rays;
lines = Graphics[{rcolor, Thickness[0.002], Line[ll]};
image = findImage[rays];
opis = Graphics[
  Text[Style[StringForm["H = `1`", hi], icolor, Bold, 26], {6, 2}]];
Show[axis, lens1, foci1, lens2, foci2, object, lines, image, opis,
  PlotRange -> {{-5, 8}, {-2.5, 2.5}},
  AspectRatio -> 0.4,
  ImageSize -> 800]];
(* Kontrolki *)
Panel[
  Row[{
    Column[{
      Control[{{x0, 2, Style["X", ocolor, 24]}, 0, 5,
        Appearance -> "Labeled", LabelStyle -> Directive[ocolor, 24]}],
      Control[{{d, 2, Style["d", l2color, 24]}, 0, 4,
        Appearance -> "Labeled", LabelStyle -> Directive[l2color, 24]}]
    }],
    Column[{
      Control[{{sign1, 1, Style[" ", 10]}, {1 -> Style[" f1 > 0 ", l1color, 22],
        -1 -> Style[" f1 < 0 ", l1color, 22]}],
      Control[{{sign2, 1, Style[" ", 10]}, {1 -> Style[" f2 > 0 ",
        l2color, 22], -1 -> Style[" f2 < 0 ", l2color, 22]}]
    }]
  ]],
  ImageMargins -> {{130, 0}, {0, 20}}

```

```

],
TrackedSymbols -> {x0, d, sign1, sign2},
FrameLabel ->
  {None, None, Style["Układ dwóch cienkich soczewek", Blue, 28], None},
Initialization -> {
  axis = Graphics[{Black, Thickness[0.002], Line[{{-5, 0}, {8, 0}}]}],
  l1color = Blue,
  l2color = Hue[0.766, 1, 0.8],
  ocolor = Hue[0.324, 1, 0.356],
  icolor = Darker[Green],
  rcolor = Red
}
]

```



Separator magnetyczny

Separatory cząstek są układami jonowo-optycznymi składającymi się z szeregu magnesów dipolowych (zaginanie torów, separacja A/q), kwadrupolowych (ogniskowanie), sekwencyjnych i wyższych rzędów (poprawki), a także innych elementów, takich jak sekcje dryfu (rura próżniowa), kolimatory, itp. Każdy z tych elementów może być opisany macierzą optyczną, a złożenie tych macierzy pozwala badać zachowanie całego układu. Na metodzie tej opiera się np. program LISE++.

W przypadku jonów, wektor zmiennych zawiera zwykle przynajmniej 5 wielkości:

x - położenie (odległość od osi optycznej) w płaszczyźnie dyspersyjnej, zazwyczaj horyzontalnej,

α_x - kąt jaki tworzy z osią x -owa składowa pędu cząstki,

y - położenie w kierunku prostopadłym do x , zazwyczaj wertykalne,

α_y - kąt jaki tworzy z osią składowa y -owa pędu,

$\delta = (B\rho - B\rho_0) / B\rho_0$ - odchylenie sztywności magnetycznej (pędu) cząstki od wartości na osi optycznej. Sztywność magnetyczna to: $B\rho = p/q$.

Czasem dodaje się jeszcze dalsze zmienne, które opisują poprawki wyższego rzędu (nieliniowe).

Pełny układ separatora zazwyczaj składa się z jednej lub więcej **sekcji dipolowych**, które ustawione są tak, że cząstki o określonej wartości $B\rho$ przechodzą wzdłuż osi optycznej.

Sekcja taka składa się z magnesu dipolowego, po obu stronach którego znajdują się układy soczewek kwadrupolowych i korekcyjnych. Sekcja taka ma zazwyczaj własność obrazowania, tzn. tworzy w ognisku końcowym obraz przedmiotu umieszczonego w ognisku początkowym.

Aby zrozumieć działanie i własności sekcji dipolowej, wystarczy rozważyć jej macierz wypadkową, która reprezentuje wszystkie elementy od ogniska początkowego do końcowego.

Model uproszczony, sekcja dipolowa $B\rho$

Analizujemy uproszczony model sekcji dipolowej.

- ▶ Własności optyczne sekcji dipolowej opisuje jedna macierz.
W dalszej dyskusji, dla uproszczenia, pomijamy współrzędne prostopadłe do płaszczyzny dyspersyjnej (czyli y i α_y).

Wektor zmiennych :

$$V = \begin{pmatrix} x \\ \alpha_x \\ \delta \end{pmatrix} = \begin{pmatrix} \text{położenie horyzontalne (w płaszczyźnie dyspersyjnej),} \\ p_x/p_z = dx/dz \text{ – kąt w płaszczyźnie horyzontalnej,} \\ \delta p/p_0 = (p - p_0)/p_0 = (B\rho - B\rho_0)/B\rho_0 \text{ – odchylenie pędu (sztywności) od wartości na osi optycznej.} \end{pmatrix}$$

Warunek obrazowania : położenie w ognisku nie zależy od kąta początkowego

$$M_{B\rho} = \begin{pmatrix} V & 0 & D \\ W & 1/V & D' \\ 0 & 0 & 1 \end{pmatrix}$$

- V – powiększenie liniowe,
- W – powiększenie kątowe,
- D – dyspersja (położenia),
- D' – dyspersja kąta.

Twierdzenia Liouville'a :
objętość przestrzeni fazowej
jest stała $\rightarrow \text{Det}(M) = 1$.

Manipulacja 3: sekcja $B\rho$

W płaszczyźnie ogniska początkowego umieszczamy przedmiot, czyli źródło cząstek. Cząstki wybiegają pod trzema różnymi kątami i z trzema różnymi pędami. Wartość pędu oznaczamy kolorem linii: **czzerwony** - najmniejsza, **niebieski** - największa (analogia do fotonów). W manipulacji możemy zmieniać położenie źródła względem osi, a także wszystkie elementy macierzy i obserwować gdzie dolatują cząstki w płaszczyźnie ogniska końcowego.

```
Manipulate[
Module[{
  mBr, ll, lines, tracks
},
particles = Flatten[Table[{ys, alpha, delta},
  {alpha, {-0.3, 0, 0.3}}, {delta, {-0.5, 0.5, 0}}, 1];
mBr = {{v, 0, d}, {w, 1/v, dprim}, {0, 0, 1}};
source = Graphics[{EdgeForm[Directive[Thick, Red]],
  Yellow, Disk[{-4, ys}, ImageScaled[{0.013, 0.035}]}]}];
raytrack[r0_] := Module[{rf, shift},
  rf = mBr.r0;
  shift = rf[[3]] / 20.;
  AppendTo[lines, {Blend[{Red, Blue}, rf[[3]] + 0.5], Thickness[0.002],
    Line[{{-4, ys + shift}, {-2.7, ys + shift + r0[[2]] * 1.3}}]}];
  AppendTo[lines, {Blend[{Red, Blue}, rf[[3]] + 0.5], Thickness[0.0025],
    Line[{{-1.3, rf[[1]] - rf[[2]] * 1.3}, {0, rf[[1]]}}]}]}];
```

```
];
lines = {};
raytrack /@ particles;
tracks = Graphics[lines];
Show[axis, focus, tracks, source, brho,
  PlotRange -> {{-4.5, 0.5}, {-2, 2}},
  AspectRatio -> 0.4,
  ImageSize -> 800]
],
(* Kontrolki *)
Panel[
Column[{
  Row[{
    Control[{{v, 1, Style["Powiększenie x, V ", Black, 20]},
      -1.5, 1.5, Appearance -> "Labeled",
      LabelStyle -> Directive[Black, 22], Exclusions -> {0}}],
    Control[{{d, 0, Style["Dyspersja x, D ", Black, 20]}, -1, 1,
      Appearance -> "Labeled", LabelStyle -> Directive[Black, 22]}]}],
  Row[{
    Control[{{w, 1, Style["Powiększenie α, W", Black, 20]}, -1, 1,
      Appearance -> "Labeled", LabelStyle -> Directive[Black, 22]}],
    Control[{{dprim, 0, Style["Dyspersja α, D'", Black, 20]}, -1,
      1, Appearance -> "Labeled", LabelStyle -> Directive[Black, 22]}]}],
  Control[{{ys, 0, Style["Położenie źródła, x ", Black, 20]}, -0.5, 0.5,
    Appearance -> "Labeled", LabelStyle -> Directive[Black, 22]}]}];
```

```

    }, Alignment → Center]
  ],
  (* *)
  TrackedSymbols → {v, w, d, dprim, ys},
  FrameLabel → {None, None, Style["Sekcja Bρ", Blue, 28], None},
  Initialization → {
    axis = Graphics[{Black, Thickness[0.002], Line[{{-5, 0}, {5, 0}}]}],
    focus =
      Graphics[{Darker[Green], Thickness[0.004], Line[{{0, -2}, {0, 2}}]}],
    brho = Graphics[{EdgeForm[Directive[Thick, Black]],
      LightGray, Rectangle[{-2.7, -1.5}, {-1.3, 1.5}],
      Text[Style[" Bρ ", Blue, 26], {-2, 0.8}],
      Text[Style[MatrixForm[{{"V", "0", "D"}, {"W", "1/V", "D'"},
        {"0", "0", "1"}]}], Black, 24], {-2.15, -0.3}],
      Text[Style[MatrixForm[{"x", "α", "δ"}], Black, 24], {-1.55, -0.3}]
    ]
  }
]

```

Sekcja Bρ

Powiększenie x, V

Dyspersja x, D

Powiększenie α, W

Dyspersja α, D'

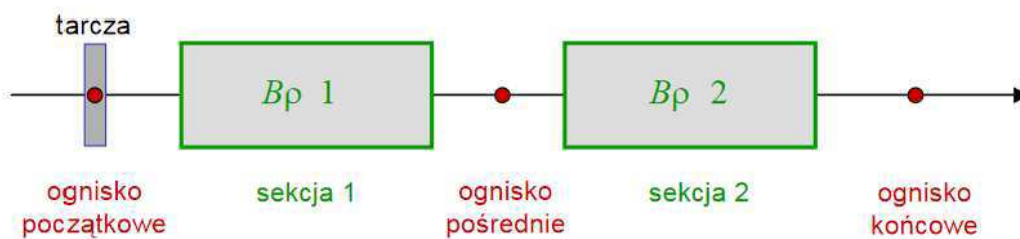
Położenie źródła, x

$$\begin{matrix}
 \text{B}\rho \\
 \begin{pmatrix} V & 0 & D \\ W & 1/V & D' \\ 0 & 0 & 1 \end{pmatrix}
 \begin{pmatrix} x \\ \alpha \\ \delta \end{pmatrix}
 \end{matrix}$$

Dwie sekcje B ρ

Modelem najprostszego separatora fragmentów są dwie sekcje dipolowe. Separator taki służy do selekcji produktów reakcji fragmentacji i jest wykorzystywany do wytwarzania najbardziej egzotycznych nuklidów w laboratoriach takich jak Dubna (Acculinna), GANIL (LISE), GSI Darmstadt (FRS), NSCL/MSU (A1900), czy RIKEN (BigRIPS). Typowo, separator fragmentów jest układem podwójnie achromatycznym, tzn. ogniskuje w jednym punkcie cząstki wybiegające ze źródła z różnymi kątami i różnymi pędami, ale są możliwe inne tryby optyczne.

Najprostszy separator : dwie sekcje dipolowe



Macierz całego układu :

$$\begin{aligned}
 M = M_{B\rho}(2) \times M_{B\rho}(1) &= \begin{pmatrix} V_2 & 0 & D_2 \\ W_2 & 1/V_2 & D'_2 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} V_1 & 0 & D_1 \\ W_1 & 1/V_1 & D'_1 \\ 0 & 0 & 1 \end{pmatrix} \\
 &= \begin{pmatrix} V_1 V_2 & 0 & D_2 + D_1 V_2 \\ \frac{W_1}{V_2} + V_1 W_2 & \frac{1}{V_1 V_2} & \frac{D'_1}{V_2} + D'_2 + D_1 W_2 \\ 0 & 0 & 1 \end{pmatrix}
 \end{aligned}$$

Jeśli $m_{13}=0$, to cały układ będzie **achromatyczny**. Wymaga to by : $D_2 = -D_1 V_2$

W ognisku końcowym tworzy się obraz „przedmiotu” z ogniska początkowego bez zniekształceń związanych z rozkładem pędu.

Zapis macierzowy pozwala badać analitycznie różne warianty optyczne. Trudno jednak w tym zapisie “zobaczyć” co się dzieje z torami cząstek. Następną manipulacją pomaga uzyskać taki “wgląd”.

Manipulacja 4: dwie sekcje $B\rho$

W manipulacji możemy zmieniać położenie źródła, jak poprzednio, i wszystkie elementy macierzowe obydwu sekcji.

```
Manipulate[
Module[{
  mBr1, mBr2, l1, lines, tracks
},
particles = Flatten[Table[{ys, alpha, delta},
  {alpha, {-0.3, 0, 0.3}}, {delta, {-0.5, 0.5, 0}}], 1];
mBr1 = {{v1, 0, d1}, {w1, 1/v1, dprim1}, {0, 0, 1}};
mBr2 = {{v2, 0, d2}, {w2, 1/v2, dprim2}, {0, 0, 1}};
source = Graphics[{EdgeForm[Directive[Thick, Red]],
  Yellow, Disk[{-4, ys}, ImageScaled[{0.013, 0.035}]}]}];
raytrack[r0_] := Module[{rf1, rf2, shift},
  rf1 = mBr1.r0;
  rf2 = mBr2.rf1;
  shift = r0[[3]] / 20.;
AppendTo[lines, {Blend[{Red, Blue}, r0[[3]] + 0.5], Thickness[0.002],
  Line[{{-4, ys + shift}, {-2.7, ys + shift + r0[[2]] * 1.3}}]}];
AppendTo[lines, {Blend[{Red, Blue}, rf1[[3]] + 0.5],
  Thickness[0.0025], Line[
  {{-1.3, rf1[[1]] - rf1[[2]] * 1.3}, {1.3, rf1[[1]] + rf1[[2]] * 1.3}}]}];
AppendTo[lines, {Blend[{Red, Blue}, rf1[[3]] + 0.5], Thickness[0.0025],
```

```

    Line[{{2.7, rf2[[1]] - rf2[[2]] * 1.3}, {4, rf2[[1]]}}]]
];
lines = {};
raytrack /@particles;
tracks = Graphics[lines];
Show[axis, focus1, focus2, tracks, source, brho1, brho2,
  PlotRange -> {{-4.5, 4.5}, {-2, 2}},
  AspectRatio -> 0.4,
  ImageSize -> 800]
],
(* Kontrolki *)
Panel[
  Column[{
    Row[{
      Control[{{v1, 1, Style[" V1 ", Black, 20]}}, -1.5, 1.5, Appearance ->
        "Labeled", LabelStyle -> Directive[Black, 22], Exclusions -> {0}}],
      Control[{{v2, 1, Style[" V2 ", Black, 20]}}, -1.5, 1.5, Appearance ->
        "Labeled", LabelStyle -> Directive[Black, 22], Exclusions -> {0}}]
    ]],
    Row[{
      Control[{{w1, 1, Style[" W1 ", Black, 20]}}, -1, 1,
        Appearance -> "Labeled", LabelStyle -> Directive[Black, 22]}],
      Control[{{w2, 1, Style[" W2 ", Black, 20]}}, -1, 1,
        Appearance -> "Labeled", LabelStyle -> Directive[Black, 22]}]
    ]],
    Row[{
      Control[{{d1, 0, Style[" D1 ", Black, 20]}}, -1, 1,

```

```

  Appearance -> "Labeled", LabelStyle -> Directive[Black, 22]}],
  Control[{{d2, 0, Style[" D2 ", Black, 20]}}, -1, 1,
    Appearance -> "Labeled", LabelStyle -> Directive[Black, 22]}]
  ]],
  Row[{
    Control[{{dprim1, 0, Style[" D1' ", Black, 20]}}, -1, 1,
      Appearance -> "Labeled", LabelStyle -> Directive[Black, 22]}],
    Control[{{dprim2, 0, Style[" D2' ", Black, 20]}}, -1, 1,
      Appearance -> "Labeled", LabelStyle -> Directive[Black, 22]}]
  ]],
  Control[{{ys, 0, Style["Położenie źródła, x ", Black, 20]}}, -0.5, 0.5,
    Appearance -> "Labeled", LabelStyle -> Directive[Black, 22]}]
  ], Alignment -> Center],
  ImageMargins -> {{50, 0}, {0, 10}}
],
(* *)
TrackedSymbols -> {v1, w1, d1, dprim1, v2, w2, d2, dprim2, ys},
FrameLabel -> {None, None, Style["Dwie sekcje Bρ", Blue, 28], None},
Initialization -> {
  axis = Graphics[{Black, Thickness[0.002], Line[{{-5, 0}, {5, 0}}]}],
  focus1 =
    Graphics[{Darker[Green], Thickness[0.004], Line[{{0, -2}, {0, 2}}]}],
  focus2 = Graphics[{Darker[Green], Thickness[0.004],
    Line[{{4, -2}, {4, 2}}]}],
  brho1 = Graphics[{EdgeForm[Directive[Thick, Black]],
    LightGray, Rectangle[{-2.7, -1.5}, {-1.3, 1.5}],
    Text[Style[" Bρ1 ", Blue, 26], {-2, 0}]}]

```



```

    }],
    brho2 = Graphics[{EdgeForm[Directive[Thick, Black]],
        LightGray, Rectangle[{1.3, 1.5}, {2.7, -1.5}],
        Text[Style[" Bρ2 ", Blue, 26], {2, 0}]
    ]}
}
]

```

Dwie sekcje Bρ

V_1 <input type="range" value="1"/> 1.	V_2 <input type="range" value="1"/> 1
W_1 <input type="range" value="0.565"/> 0.565	W_2 <input type="range" value="0.865"/> 0.865
D_1 <input type="range" value="0.37"/> 0.37	D_2 <input type="range" value="-0.36"/> -0.36
D_1' <input type="range" value="-0.525"/> -0.525	D_2' <input type="range" value="0.195"/> 0.195
Położenie źródła, x <input type="range" value="0.074"/> 0.074	


Separator fragmentów z degraderem

Sekcje dipolowe dokonują tylko selekcji jonów względem $B\rho$, a to często nie jest wystarczające. Separator złożony z dwóch sekcji dipolowych pozwala zastosować pewien trick. Pierwsza sekcja rozdziela cząstki o różnych pędach (dyspersja). W ognisku środkowym umieszczamy specjalnie wyprofilowany **degrader**, w którym cząstki o różnych pędach, które wskutek dyspersji mają różne położenia, mogą różnie zmieniać swój pęd. Następnie druga sekcja zbiera cząstki, które przeszły przez degrader i ogniskuje je w ognisku końcowym.

Do układu dodajemy nowy element optyczny: warstwę materiału (degrader).

Warstwa materii na drodze jonów zmienia ich prędkości i w ogólności zakłóca optykę spektrometru. Poprzez specjalnie dobrany kształt degradera można zmniejszać te zakłócenia, a nawet modyfikować własności optyczne separatora.

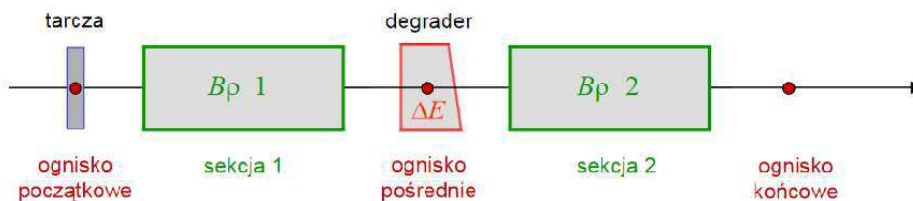
Macierz optyczna dla degradera :



$$\longleftrightarrow M_{\Delta E} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D_K & 0 & V_K \end{pmatrix}$$

D_K – „dyspersja”, czyli jak położeni wpływa na zmianę pędu,
 V_K – „powiększenie”, czyli jak pęd początkowy wpływa na koniec

Wstawiamy degrader pomiędzy dwie sekcje dipolowe i uzyskujemy pełny układ separatora fragmentów:



Macierz całego układu :

$$M = M_{Bp}(2) \times M_{\Delta E} \times M_{Bp}(1) = \begin{pmatrix} V_2 & 0 & D_2 \\ W_2 & 1/V_2 & D'_2 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D_K & 0 & V_K \end{pmatrix} \times \begin{pmatrix} V_1 & 0 & D_1 \\ W_1 & 1/V_1 & D'_1 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} V_1 V_2 + V_1 D_2 D_K & 0 & D_1 D_2 D_K + D_1 V_2 + D_2 V_K \\ \frac{W_1}{V_2} + V_1 D_K D'_2 + V_1 W_2 & \frac{1}{V_1 V_2} & \frac{D'_1}{V_2} + D'_2 V_K + D_1 D_K D'_2 + D_1 W_2 \\ D_K V_1 & 0 & D_1 D_K + V_K \end{pmatrix} \quad \text{Det}[M] = V_K$$

Własność obrazowania jest zachowana, ale własności optyczne zależą od doboru parametrów D_K i V_K . Pojawiają się dzięki temu nowe możliwości.

Cztery najważniejsze ustawienia

Wyróżnia się 4 główne tryby optyczne:

- Tryb monoenergetyczny

Wszystkie jony w ognisku końcowym mają tę samą energię, co oznacza, że $m_{33} = 0$.

Jednak cały układ jest wtedy dyspersyjny, $m_{13} = D_1 V_2$.

$$\begin{pmatrix} V_1 V_2 + V_1 D_2 D_K & 0 & D_1 D_2 D_K + D_1 V_2 + D_2 V_K \\ \frac{W_1}{V_2} + V_1 D_K D'_2 + V_1 W_2 & \frac{1}{V_1 V_2} & \frac{D'_1}{V_2} + D'_2 V_K + D_1 D_K D'_2 + D_1 W_2 \\ D_K V_1 & 0 & D_1 D_K + V_K \end{pmatrix}$$

1. Tryb monoenergetyczny $m_{33} = 0 \Rightarrow D_K = -\frac{1}{D_1} V_K$

- Tryb dopasowanego degradera

Układ bardzo podobny do tego bez degradera. W szczególności, jeśli układ bez degradera był achromatyczny, to taki pozostaje.

$$\begin{pmatrix} V_1 V_2 + V_1 D_2 D_K & 0 & D_1 D_2 D_K + D_1 V_2 + D_2 V_K \\ \frac{W_1}{V_2} + V_1 D_K D'_2 + V_1 W_2 & \frac{1}{V_1 V_2} & \frac{D'_1}{V_2} + D'_2 V_K + D_1 D_K D'_2 + D_1 W_2 \\ D_K V_1 & 0 & D_1 D_K + V_K \end{pmatrix}$$

2. Tryb dopasowanego degradera $m_{33} = 1 \Rightarrow D_K = -\frac{1}{D_1}(V_K - 1)$

• Tryb achromatyczny

Separator staje się achromatyczny nawet jeśli przed włożeniem degradera nie był. Ten tryb jest najczęściej używany w praktyce.

$$\begin{pmatrix} V_1 V_2 + V_1 D_2 D_K & 0 & D_1 D_2 D_K + D_1 V_2 + D_2 V_K \\ \frac{W_1}{V_2} + V_1 D_K D'_2 + V_1 W_2 & \frac{1}{V_1 V_2} & \frac{D'_1}{V_2} + D'_2 V_K + D_1 D_K D'_2 + D_1 W_2 \\ D_K V_1 & 0 & D_1 D_K + V_K \end{pmatrix}$$

3. Tryb achromatyczny $m_{13} = 0 \Rightarrow D_K = -\frac{1}{D_1}\left(V_K + \frac{D_1 V_2}{D_2}\right)$

• Degrader jednorodny

Układ taki bywa nazywany spektrometrem strat energii. Może być achromatyczny, jeśli

wyberzemy $D_2 = -D_1 V_2 / V_K$.

4. Jednorodny degrader $D_K = 0$

Macierz układu przyjmuje wtedy postać :

$$M = \begin{pmatrix} V_1 V_2 & 0 & D_1 V_2 + D_2 V_K \\ \frac{W_1}{V_2} + V_1 W_2 & \frac{1}{V_1 V_2} & \frac{D'_1}{V_2} + D'_2 V_K + D_1 W_2 \\ 0 & 0 & V_K \end{pmatrix}$$

Jest to tryb komplementarny do dopasowanego : pierwsze dwie kolumny są takie, jak dla układu bez degradera.

Manipulacja 5: pełny separator - dwie sekcje $B\rho$ z degraderem

Dla uproszczenia i większej przejrzystości rezygnujemy z manipulowania wszystkimi parametrami i kładziemy:

$$V_1 = 1, \quad W_1 = 1, \quad D_1 = 0, \quad W_2 = 1, \quad D_2 = 0.$$

Dodajemy cztery przyciski, które pozwalają wybrać parametry odpowiadające czterem głównym trybom optycznym.

```
Manipulate[
Module[{
  mBr1, mBr2, mK, l1, lines, tracks,
  v1 = 1, w1 = 1, w2 = 1, dprim1 = 0, dprim2 = 0
},
particles = Flatten[Table[{ys, alpha, delta},
  {alpha, {-0.3, 0, 0.3}}, {delta, {-0.5, 0.5, 0}}, 1];
mBr1 = {{v1, 0, d1}, {w1, 1/v1, dprim1}, {0, 0, 1}};
mK = {{1, 0, 0}, {0, 1, 0}, {dK, 0, vK}};
mBr2 = {{v2, 0, d2}, {w2, 1/v2, dprim2}, {0, 0, 1}};
source = Graphics[{EdgeForm[Directive[Thick, Red]],
  Yellow, Disk[{-4, ys}, ImageScaled[{0.013, 0.035}]}]];
wedge = Graphics[{Opacity[0.5], EdgeForm[Directive[Thick, Black]],
  LightGray, Polygon[{{0, -1.5}, {0, 1.5}, {0.1 * (vK - 1) * (1 - 0.5 * dK),
  1.5}, {0.1 * (vK - 1) * (1 + 0.5 * dK), -1.5}}]}];
raytrack[r0_] := Module[{rf1, rf2, rfK, shift},
  rf1 = mBr1.r0;
```

```
  rfK = mK.rf1;
  rf2 = mBr2.rfK;
  shift = r0[[3]] / 20.;
  AppendTo[lines, {Blend[{Red, Blue}, r0[[3]] + 0.5], Thickness[0.002],
  Line[{{-4, ys + shift}, {-2.7, ys + shift + r0[[2]] * 1.3}}]}];
  AppendTo[lines, {Blend[{Red, Blue}, rf1[[3]] + 0.5], Thickness[0.0025],
  Line[{{-1.3, rf1[[1]] - rf1[[2]] * 1.3}, {0, rf1[[1]]}}]}];
  AppendTo[lines, {Blend[{Red, Blue}, rfK[[3]] + 0.5], Thickness[0.0025],
  Line[{{0, rfK[[1]]}, {1.3, rfK[[1]] + rfK[[2]] * 1.3}}]}];
  AppendTo[lines, {Blend[{Red, Blue}, rf2[[3]] + 0.5], Thickness[0.0025],
  Line[{{2.7, rf2[[1]] - rf2[[2]] * 1.3}, {4, rf2[[1]]}}]}];
];
lines = {};
raytrack /@ particles;
tracks = Graphics[lines];
Show[axis, focus1, focus2, tracks, source, brho1, brho2, wedge,
  PlotRange -> {{-4.5, 4.5}, {-2, 2}},
  AspectRatio -> 0.4,
  ImageSize -> 800]
],
(* Kontrolki *)
Panel[
  Column[{
    Row[{
      Control[{{d1, 0, Style["D1 ", Black, 20]}, -1, 1,
        Appearance -> "Labeled", LabelStyle -> Directive[Black, 22]}],
      Control[{{ys, 0, Style[" x ", Black, 20]}, -0.5, 0.5,
```

```

    Appearance → "Labeled", LabelStyle → Directive[Black, 22]]]],
  Row[{
    Control[{{dK, 0, Style[" DK ", Black, 20]}}, -2, 2, Appearance → "Labeled",
      LabelStyle → Directive[Black, 22], Enabled → (vK > 1)]],
    Control[{{vK, 1, Style[" VK ", Black, 20]}}, 1, 2,
      Appearance → "Labeled", LabelStyle → Directive[Black, 22]]]
  ]],
  Row[{
    Control[{{d2, 0, Style[" D2 ", Black, 20]}}, -1, 1,
      Appearance → "Labeled", LabelStyle → Directive[Black, 22]]],
    Control[{{v2, 1, Style[" V2 ", Black, 20]}}, -1.5, 1.5, Appearance →
      "Labeled", LabelStyle → Directive[Black, 22], Exclusions → {0}]
  ]],
  Spacer[10],
  Row[{
    Button["Monoenergetyczny",
      vK = 1.5; d1 = 1; dK = -1.5,
      BaseStyle → 16, ImageMargins → 5, ImageSize → All],
    Button["Dopasowany",
      vK = 1.5; d1 = 0.5; dK = -1,
      BaseStyle → 16, ImageMargins → 5, ImageSize → All],
    Button["Achromatyczny",
      vK = 1.5; d1 = 1; v2 = 0.5; d2 = 1; dK = -2,
      BaseStyle → 16, ImageMargins → 5, ImageSize → All],
    Button["Homogeniczny",
      dK = 0; d1 = 1; v2 = 0.5; vK = 1.5; d2 = -0.3,
      BaseStyle → 16, ImageMargins → 5, ImageSize → All]
  ]

```

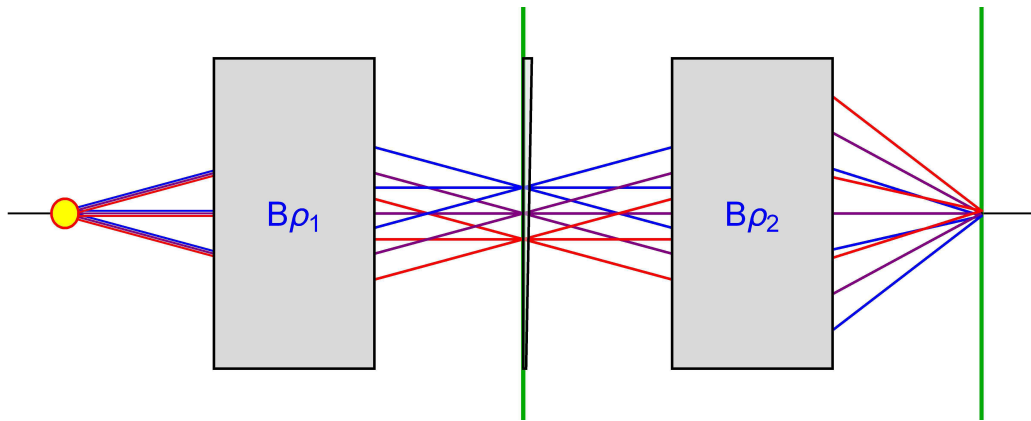
```

  ]], Alignment → Center],
  ImageMargins → {{60, 0}, {0, 10}}
],
(* *)
TrackedSymbols → {d1, dK, d2, ys, vK, v2},
FrameLabel → {None, None, Style[
  "Separator fragmentów: dwie sekcje Bρ z degraderem", Blue, 28], None},
Initialization → {
  axis = Graphics[{Black, Thickness[0.002], Line[{{-5, 0}, {5, 0}}]}],
  focus1 =
    Graphics[{Darker[Green], Thickness[0.004], Line[{{0, -2}, {0, 2}}]}],
  focus2 = Graphics[{Darker[Green], Thickness[0.004],
    Line[{{4, -2}, {4, 2}}]}],
  brho1 = Graphics[{EdgeForm[Directive[Thick, Black]],
    LightGray, Rectangle[{-2.7, -1.5}, {-1.3, 1.5}],
    Text[Style[" Bρ1 ", Blue, 26], {-2, 0}]
  ]],
  brho2 = Graphics[{EdgeForm[Directive[Thick, Black]],
    LightGray, Rectangle[{1.3, 1.5}, {2.7, -1.5}],
    Text[Style[" Bρ2 ", Blue, 26], {2, 0}]
  ]]
}
]

```

Separator fragmentów: dwie sekcje $B\rho$ z degraderem

D_1 x
 D_K V_K
 D_2 V_2



Manipulacja 6: separator z degraderem, wersja rozbudowana

Do poprzedniej wersji dodajemy jeszcze wykres przestrzeni fazowej w ognisku końcowym, czyli zależność $B\rho$ od położenia.

```

Manipulate[
Module[{
  mBr1, mBr2, mK, l1, lines, tracks, phasespace, phaseplot,
  v1 = 1, w1 = 1, w2 = 1, dprim1 = 0, dprim2 = 0
},
  particles = Flatten[Table[{ys, alpha, delta},
    {alpha, {-0.3, 0, 0.3}}, {delta, {-0.5, 0.5, 0}}, 1];
  mBr1 = {{v1, 0, d1}, {w1, 1/v1, dprim1}, {0, 0, 1}};
  mK = {{1, 0, 0}, {0, 1, 0}, {dK, 0, vK}};
  mBr2 = {{v2, 0, d2}, {w2, 1/v2, dprim2}, {0, 0, 1}};
  source = Graphics[{EdgeForm[Directive[Thick, Red]],
    Yellow, Disk[{-4, ys}, ImageScaled[{0.013, 0.035}]]}];
  wedge = Graphics[{Opacity[0.5], EdgeForm[Directive[Thick, Black]],
    LightGray, Polygon[{{0, -1.5}, {0, 1.5}, {0.1 * (vK - 1) * (1 - 0.5 * dK),
      1.5}, {0.1 * (vK - 1) * (1 + 0.5 * dK), -1.5}}]}}];
  raytrack[r0_] := Module[{rf1, rf2, rfK, shift},
    rf1 = mBr1.r0;
    rfK = mK.rf1;
    rf2 = mBr2.rfK;
    shift = r0[[3]] / 20.;
  
```

```

AppendTo[lines, {Blend[{Red, Blue}, r0[[3]] + 0.5], Thickness[0.002],
  Line[{{-4, ys + shift}, {-2.7, ys + shift + r0[[2]] * 1.3}}]};
AppendTo[lines, {Blend[{Red, Blue}, rf1[[3]] + 0.5], Thickness[0.0025],
  Line[{{-1.3, rf1[[1]] - rf1[[2]] * 1.3}, {0, rf1[[1]]}}]};
AppendTo[lines, {Blend[{Red, Blue}, rfK[[3]] + 0.5], Thickness[0.0025],
  Line[{{0, rfK[[1]]}, {1.3, rfK[[1]] + rfK[[2]] * 1.3}}]};
AppendTo[lines, {Blend[{Red, Blue}, rf2[[3]] + 0.5], Thickness[0.0025],
  Line[{{2.7, rf2[[1]] - rf2[[2]] * 1.3}, {4, rf2[[1]]}}]};
AppendTo[phasespace, {rf2[[1]], rf2[[3]]}]
];
lines = {};
phasespace = {};
raytrack /@ particles;
tracks = Graphics[lines];
phaseplot = ListPlot[phasespace,
  Frame → True,
  PlotRange → {{-1, 1}, {-2, 2}},
  BaseStyle → {Black, FontFamily → "Verdana", FontSize → 22},
  PlotStyle → {Red, PointSize[0.05]},
  GridLines → Automatic,
  FrameLabel → {"x-ognisko końcowe", " $\delta = \Delta B \rho / B \delta_0$ "},
  ImageSize → 500
];
TableView[{
  {1, Style["Separator", Darker[Blue], 20] ->
    Show[axis, focus1, focus2, tracks, source, brho1, brho2, wedge,
      PlotRange → {{-4.5, 4.5}, {-2, 2}},

```

```

  AspectRatio → 0.4,
  ImageSize → 800}],
  {2, Style["Przestrzeń fazowa", Darker[Red], 20] → Show[phaseplot]}
}, Dynamic[tab], Alignment → Center]
],
(* Kontrolki *)
Panel[
  Column[{
    Row[{
      Control[{{d1, 0, Style["D1", Black, 20]}, -1, 1,
        Appearance → "Labeled", LabelStyle → Directive[Black, 22]}],
      Control[{{ys, 0, Style[" x0 ", Black, 20]}, -0.5, 0.5,
        Appearance → "Labeled", LabelStyle → Directive[Black, 22]}]
    ]},
    Row[{
      Control[{{dK, 0, Style[" DK ", Black, 20]}, -2, 2, Appearance → "Labeled",
        LabelStyle → Directive[Black, 22], Enabled → (vK > 1)}],
      Control[{{vK, 1, Style[" VK ", Black, 20]}, 1, 2,
        Appearance → "Labeled", LabelStyle → Directive[Black, 22]}]
    ]},
    Row[{
      Control[{{d2, 0, Style[" D2 ", Black, 20]}, -1, 1,
        Appearance → "Labeled", LabelStyle → Directive[Black, 22]}],
      Control[{{v2, 1, Style[" V2 ", Black, 20]}, -1.5, 1.5, Appearance →
        "Labeled", LabelStyle → Directive[Black, 22], Exclusions → {0}}]
    ]},
    Spacer[10],
    Row[{

```



```

    Button["Monoenergetyczny",
      vK = 1.5; d1 = 1; dK = -1.5,
      BaseStyle → 16, ImageMargins → 5, ImageSize → All],
    Button["Dopasowany",
      vK = 1.5; d1 = 0.5; dK = -1,
      BaseStyle → 16, ImageMargins → 5, ImageSize → All],
    Button["Achromatyczny",
      vK = 1.5; d1 = 1; v2 = 0.5; d2 = 1; dK = -2,
      BaseStyle → 16, ImageMargins → 5, ImageSize → All],
    Button["Homogeniczny",
      dK = 0; d1 = 1; v2 = 0.5; vK = 1.5; d2 = -0.3,
      BaseStyle → 16, ImageMargins → 5, ImageSize → All]
  ]]
}, Alignment → Center],
ImageMargins → {{60, 0}, {0, 10}}
],
{{tab, 1}, None},
(* *)
TrackedSymbols ⇒ {d1, dK, d2, ys, vK, v2},
FrameLabel → {None, None, Style[
  "Separator fragmentów: dwie sekcje Bρ z degraderem", Blue, 28], None},
Initialization ⇒ {
  axis = Graphics[{Black, Thickness[0.002], Line[{{-5, 0}, {5, 0}}]}],
  focus1 =
  Graphics[{Darker[Green], Thickness[0.004], Line[{{0, -2}, {0, 2}}]}],
  focus2 = Graphics[{Darker[Green], Thickness[0.004],
  Line[{{4, -2}, {4, 2}}]}],
}
]

```

```

brho1 = Graphics[{EdgeForm[Directive[Thick, Black]],
  LightGray, Rectangle[{-2.7, -1.5}, {-1.3, 1.5}],
  Text[Style[" Bρ1 ", Blue, 26], {-2, 0}]
}],
brho2 = Graphics[{EdgeForm[Directive[Thick, Black]],
  LightGray, Rectangle[{1.3, 1.5}, {2.7, -1.5}],
  Text[Style[" Bρ2 ", Blue, 26], {2, 0}]
}]
}
]

```

Separator fragmentów: dwie sekcje Bρ z degraderem

D_1	<input type="range" value="0.5"/>	x_0	<input type="range" value="0"/>
D_K	<input type="range" value="-1"/>	V_K	<input type="range" value="1.5"/>
D_2	<input type="range" value="0"/>	V_2	<input type="range" value="1"/>

Separator

Przestrzeń fazowa

