

Ohjelma ytimen yksihiukkastilojen ratkaisemiseksi

Pro gradu -tutkielma, 26.8.2019

Tekijä:

JOONA KASURINEN

Ohjaaja:

JOUNI SUHONEN



JYVÄSKYLÄN YLIOPISTO
FYSIKAN LAITOS

© 2019 Joonas Kasurinen

Julkaisu on tekijänoikeussäännösten alainen. Teosta voi lukea ja tulostaa henkilökohtaista käyttöä varten. Käyttö kaupallisiin tarkoituksiin on kielletty. This publication is copyrighted. You may download, display and print it for Your own personal use. Commercial use is prohibited.

Tiivistelmä

Kasurinen, Joonas

Ohjelma ytimen yksihiukkastilojen ratkaisemiseksi

Pro gradu -tutkielma

Fysiikan laitos, Jyväskylän yliopisto, 2019, 29 sivua + 2 liitettä

Esitän Kouran ja Yamadan potentiaalia käyttävän ohjelman ytimen yksihiukkastilojen laskemiseksi. Ratkaisut löydetään diagonalisoimalla Hamiltonin matriisi. Käyn läpi ohjelman taustalla olevan teorian ja numeriikan, minkä jälkeen totean ohjelman toimivuuden vertaamalla sen tuottamia tuloksia Kouran ja Yamadan alkuperäisiin tuloksiin.

Avainsanat: ydinfysiikka, keskeiskenttä, Woods–Saxon, yksihiukkastila

Sisältö

| | |
|---|-----------|
| Tiivistelmä | 3 |
| 1 Johdanto | 7 |
| 2 Teoria | 9 |
| 2.1 Keskeiskenttäapproksimaatio | 9 |
| 2.2 Yksihiukkaspotentiaali | 10 |
| 2.3 Radiaalinen Schrödinger-yhtälö | 11 |
| 2.4 Harmonisen oskillaattorin aaltofunktiot | 13 |
| 3 Numeriikka | 15 |
| 3.1 Numeerisesta integroinnista | 15 |
| 3.2 Simpsonin sääntö | 16 |
| 3.3 Äärettömyys ja nolla | 18 |
| 4 Ohjelman esittely | 21 |
| 5 Tulokset | 23 |
| 5.1 Pb-208 | 23 |
| 5.2 Zr-90 | 24 |
| 5.3 Ca-48 | 25 |
| 5.4 C-14 | 26 |
| 5.5 He-4 | 26 |
| 6 Yhteenveto | 27 |

Lähteet

A Coulomb-termin johto

B Lähdekoodi

1 Johdanto

Koura ja Yamada esittivät artikkelissaan *Single-particle potentials for spherical nuclei* Woods–Saxonin hienonnuksen, jonka tarkoitus oli päteä tavallista Woods–Saxonia laajemmalla nuklidialueella.[1] Artikkelissa esitettyjen tulosten mukaan he myöskin onnistuivat tavoitteessaan.

Olisi siis syytä ottaa potentiaali käyttöön. Analyttisiä ratkaisuja ei tietenkään ole saatavilla, joten on turvauduttava numerikkaan. Ja koska tietokone on tässä hommassa ihmistä paljon lahjakkaampi, tiemme johtaa lopulta ohjelmoinnin pariin.

Esitän ohjelman nukleonien tilojen ratkaisemiseksi. Ohjelma muodostaa Hamiltonin operaattorin matriisiesityksen ja diagonalisoi sen. Totean ohjelman toimivuuden vertaamalla tuloksia Kouran ja Yamadan tuloksiin.

Käyn läpi ohjelman taustalla olevan teorian ja numeeriset menetelmät. Esitän tulokset muutamalle eri ytimelle, minkä jälkeen pohdin näiden avulla ohjelman vahvuuksia ja puutteita, sekä mahdollisia virhelähteitä. Tutkielmaa voi pitää jossain mielessä ohjelman dokumentaationa.

2 Teoria

2.1 Keskeiskenttäapproksimaatio

Näytän ensin lyhyesti, kuinka yksittäisten nukleonien Schrödinger-yhtälöt saadaan keskeiskenttäapproksimaation avulla. Ytimen Hamiltonin operaattori on

$$H = T + V = \sum_{i=1}^A t(\mathbf{r}_i) + \sum_{\substack{i,j=1 \\ i < j}}^A v(\mathbf{r}_i, \mathbf{r}_j). \quad [2] \quad (1)$$

Ensimmäinen summa käy läpi kaikkien nukleonien liike-energiat ja toinen summa käy läpi kaikki kaksihiukkasvuorovaikutukset. Hauskana sivuhuomautuksena voisin mainita, että tarkalleen ottaen ollaan jo nyt siirrytty approksimointiin, koska ydinvoimalla on todettu olevan useamman kappaleen vuorovaikutuksia.[3]

Approksimointi ei kuitenkaan lopu tähän. Seuraavaksi muokataan Hamiltonin operaattori sellaiseen muotoon, josta keskeiskentän idea on helppo ymmärtää. Lisätään ja vähennetään kunkin nukleonin yksihiukkaspotentiaali:

$$H = \left(T + \sum_{i=1}^A v(\mathbf{r}_i) \right) + \left(V - \sum_{i=1}^A v(\mathbf{r}_i) \right) := H_{\text{MF}} + V_{\text{RES}}. \quad (2)$$

Samalla tuli määriteltyä keskeiskentän Hamiltonin operaattori H_{MF} ja residuaali-vuorovaikutus V_{RES} . Keskeiskenttäapproksimaatio on oleellisesti vain toteamus, että $V_{\text{RES}} \approx 0$. Mutta mitä tämä oikeastaan tarkoittaa? Asiaa on syytä pohtia tarkemmin keskeiskenttäapproksimaation idean ymmärtämiseksi.

$$V_{\text{RES}} = V - \sum_{i=1}^A v(\mathbf{r}_i) = 0 \iff V = \sum_{i=1}^A v(\mathbf{r}_i), \quad (3)$$

eli ytimen potentiaali koostuu sen nukleonien yksihiukkaspotentiaalien summasta. Sama toistuu myös keskeiskentän Hamiltonin operaattorissa. Ero on kielellisesti hienovarainen: nukleonit eivät vuorovaikuta keskenään, vaan ne luovat yhdessä kentän, jonka kanssa kukin sitten vuorovaikuttaa. Matemaattisesti tämä näkyy siinä, että muotoa $v(\mathbf{r}_i, \mathbf{r}_j)$ olevat termit ovat hävinneet.

Nyt ytimen Hamiltonin operaattoriksi jää H_{MF} . Se on yksittäisten nukleonien Hamiltonin operaattorien summa:

$$H_{\text{MF}} = T + \sum_{i=1}^A v(\mathbf{r}_i) = \sum_{i=1}^A (t(\mathbf{r}_i) + v(\mathbf{r}_i)) := \sum_{i=1}^A \left(\frac{-\hbar^2}{2m} \nabla_{\mathbf{r}_i}^2 + v(\mathbf{r}_i) \right), \quad (4)$$

joten ytimen Schrödinger-yhtälö separoituu. Näin nukleonit voidaan ratkaista erikseen. Yksittäisen nukleonin Schrödinger-yhtälö on

$$\left(\frac{-\hbar^2}{2m} \nabla^2 + v(\mathbf{r}) \right) \phi(\mathbf{r}) = \varepsilon \phi(\mathbf{r}). \quad (5)$$

2.2 Yksihiukkaspotentiaali

Käytän potentiaalina Kouran ja Yamadan esittämää Woods–Saxonin hienonnusta. Yksityiskohdat (kuten parametrien arvot) voi halutessaan kaivaa heidän artikkelistaan.

Nukleonien tuntema potentiaali koostuu ydinvoiman keskeiskomponentista, Coulomb-termistä ja spin–rata-vuorovaikutuksesta:

$$v(r) = v_{\text{cen}}(r) + v_c(r) + v_{\text{ls}}(r) \mathbf{L} \cdot \mathbf{S}. \quad (6)$$

Ydinvoima on oletettu radiaaliseksi ja, Kouraa ja Yamadaa mukaillen, se on muotoa

$$v_{\text{cen}}(r) = V_0 \frac{1}{\left(1 + \exp\left(\frac{r-R_v}{a_v}\right)\right)^{a_v/\kappa}} \left(1 + V_{\text{dp}} \frac{1}{1 + \exp\left(-\frac{r-R_v}{a_v}\right)}\right). \quad (7)$$

Ytimellä on varaustiheys

$$\rho_C(r) = \frac{\rho_{C0}}{1 + \exp\left(\frac{r-R_C}{a_C}\right)}, \quad (8)$$

josta laskettuna (ks. liite A) Coulomb-termi on (protoneille)

$$v_c(r) = q_e \frac{\rho_{C0}}{\varepsilon_0} \int_r^\infty \frac{1}{r'^2} \int_0^{r'} \frac{r''^2}{1 + \exp\left(\frac{r''-R_C}{a_C}\right)} dr'' dr'. \quad (9)$$

Spin-rata-vuorovaikutus on samaa muotoa kuin tavanomaisessa Woods-Saxonissa:

$$v_{\text{ls}}(r) = V_{\text{ls}0} \frac{1}{2m} \frac{1}{r} \frac{d}{dr} \left(\frac{1}{1 + \exp \frac{r-R_{\text{ls}}}{a_{\text{ls}}}} \right). \quad (10)$$

2.3 Radiaalinen Schrödinger-yhtälö

Seuraavaksi muunnetaan kolmiulotteinen Schrödinger-yhtälö (5) yksiulotteiseksi. Laplace-operaattori on pallokoordinaateissa

$$\nabla^2 = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2}{\partial \varphi^2}. \quad (11)$$

Pyörimismäärän neliö on taas pallokoordinaateissa (ks. vaikkapa Griffiths[4])

$$\mathbf{L}^2 = -\hbar^2 \left(\frac{1}{\sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial}{\partial \theta} \right) + \frac{1}{\sin^2 \theta} \frac{\partial^2}{\partial \varphi^2} \right). \quad (12)$$

Yhtälöitä (11) ja (12) vertaamalla huomataan, että

$$\nabla^2 = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial}{\partial r} \right) - \frac{\mathbf{L}^2/\hbar^2}{r^2} := \nabla_r^2 - \frac{\mathbf{L}^2/\hbar^2}{r^2}. \quad (13)$$

Kun sijoitetaan (6) ja (13) yhtälöön (5), saadaan

$$\left(\frac{-\hbar^2}{2m} \left(\nabla_r^2 - \frac{\mathbf{L}^2/\hbar^2}{r^2} \right) + v_{\text{cen}}(r) + v_c(r) + v_{\text{ls}}(r) \mathbf{L} \cdot \mathbf{S} \right) \phi(\mathbf{r}) = \varepsilon \phi(\mathbf{r}). \quad (14)$$

Oletetaan, että nukleonien pyörimismäärien suuruudet on hyvin määritelty. Tällöin, koska

$$\mathbf{J}^2 := (\mathbf{L} + \mathbf{S})^2 \iff \mathbf{L} \cdot \mathbf{S} = \frac{1}{2} (\mathbf{J}^2 - \mathbf{L}^2 - \mathbf{S}^2), \quad (15)$$

niin yhtälö (14) saadaan muotoon

$$\begin{aligned} & \left(\frac{-\hbar^2}{2m} \left(\nabla_r^2 - \frac{l(l+1)}{r^2} \right) + v_{\text{cen}}(r) + v_c(r) \right. \\ & \left. + \frac{\hbar^2}{2} \left(j(j+1) - l(l+1) - \frac{3}{4} \right) v_{\text{ls}}(r) \right) \phi_{nl\frac{1}{2}jm_j}(\mathbf{r}) = \varepsilon_{nlj} \phi_{nl\frac{1}{2}jm_j}(\mathbf{r}). \end{aligned} \quad (16)$$

Nyt suuntariippuvuutta on jäljellä enää aaltofunktiossa. Tämä voidaan yksinkertaisesti jakaa pois. Tuloksena saadaan nukleonin radiaalinen Schrödinger-yhtälö:

$$\begin{aligned} & \left(\frac{-\hbar^2}{2m} \left(\nabla_r^2 - \frac{l(l+1)}{r^2} \right) + v_{\text{cen}}(r) + v_c(r) \right. \\ & \left. + \frac{\hbar^2}{2} \left(j(j+1) - l(l+1) - \frac{3}{4} \right) v_{\text{ls}}(r) \right) f_{nlj}(r) = \varepsilon_{nlj} f_{nlj}(r), \end{aligned} \quad (17)$$

joka on itse asiassa samaa muotoa tavallisen yksiulotteisen Schrödinger-yhtälön kanssa. Jotta tämä olisi selvemmin nähtävissä, määritellään radiaalinen aaltofunktio $u_{nlj}(r) := r f_{nlj}(r)$. Pienen muokkailun jälkeen yhtälö (17) saadaan näyttämään tältä:

$$\begin{aligned} & \left(\frac{-\hbar^2}{2m} \frac{d^2}{dr^2} + \frac{l(l+1)\hbar^2}{2mr^2} + v_{\text{cen}}(r) + v_c(r) + \frac{\hbar^2}{2} \left(j(j+1) - l(l+1) - \frac{3}{4} \right) v_{\text{ls}}(r) \right) u_{nlj}(r) \\ & = \varepsilon_{nlj} u_{nlj}(r) := h_{nlj}(r) u_{nlj}(r). \end{aligned} \quad (18)$$

2.4 Harmonisen oskillaattorin aaltofunktiot

Ratkaistaan yhtälö (18) ominaisarvo-ongelmana. Muodostetaan siis ensin Hamiltonin operaattorin matriisiesitys jossakin kannassa. Tämän jälkeen energiat ja aaltofunktiot saadaan diagonalisoimalla.

Kantafunktioiksi valitaan harmonisen oskillaattorin radiaaliset aaltofunktiot $q_{nl}(r)$. Ne toteuttavat yhtälön

$$\left(\frac{-\hbar^2}{2m} \frac{d^2}{dr^2} + \frac{l(l+1)\hbar^2}{2mr^2} - V_1 + \frac{1}{2}m\omega^2 r^2 \right) q_{nl}(r) = \epsilon_{nl} q_{nl}(r). \quad (19)$$

Ratkaisut ovat tunnetut ja peräti analyyttisetkin.[5] Energiat ovat

$$\epsilon_{nl} = -V_1 + \left(2n + l + \frac{3}{2} \right) \hbar\omega \quad (20)$$

ja aaltofunktiot

$$q_{nl}(r) = r \sqrt{\frac{2n!}{b^3 \Gamma(n + l + \frac{3}{2})}} \left(\frac{r}{b} \right)^l \exp\left(-\frac{r^2}{2b^2}\right) L_n^{l+\frac{1}{2}}\left(\frac{r^2}{b^2}\right); \quad (21)$$

$$b := \sqrt{\frac{\hbar}{m\omega}} = \frac{\hbar}{\sqrt{m\hbar\omega}}, \quad (22)$$

$$L_n^\alpha(x) = e^x \frac{x^{-\alpha}}{n!} \frac{d^n}{dx^n} (e^{-x} x^{n+\alpha}). \quad (23)$$

Parametriä b kutsutaan oskillaattorin pituudeksi. Jos b valitaan järjellisesti, niin vertailu ydinvoiman harmoniseen malliin saadaan ikään kuin ilmaiseksi. Tehdään siis näin! Hyvä arvio oskillaattorin pituudelle saadaan Blomqvist–Molinari-kaavan avulla, jonka mukaan

$$\hbar\omega = (45A^{-1/3} - 25A^{-2/3})\text{MeV}.[6] \quad (24)$$

Matriisielementti on

$$\begin{aligned}
\langle v' | h_{nlj} | v \rangle &= \int_0^\infty q_{v'l}^*(r) h_{nlj}(r) q_{vl}(r) dr = \int_0^\infty q_{v'l}^*(r) \left(\frac{-\hbar^2}{2m} \frac{d^2}{dr^2} + \frac{l(l+1)\hbar^2}{2mr^2} \right. \\
&\quad \left. + v_{\text{cen}}(r) + v_c(r) + \frac{\hbar^2}{2} \left(j(j+1) - l(l+1) - \frac{3}{4} \right) v_{\text{ls}}(r) \right) q_{vl}(r) dr \\
&= \int_0^\infty q_{v'l}^*(r) \left(\frac{-\hbar^2}{2m} \frac{d^2}{dr^2} + \frac{l(l+1)\hbar^2}{2mr^2} \right) q_{vl}(r) dr \\
&\quad + \int_0^\infty q_{v'l}^*(r) \left(v_{\text{cen}}(r) + v_c(r) + \frac{\hbar^2}{2} \left(j(j+1) - l(l+1) - \frac{3}{4} \right) v_{\text{ls}}(r) \right) q_{vl}(r) dr.
\end{aligned} \tag{25}$$

Seuraavaksi temppuillaan derivoinnit pois ensimmäisen integraalin sisältä. Yhtälöstä (19) nähdään, että

$$\left(\frac{-\hbar^2}{2m} \frac{d^2}{dr^2} + \frac{l(l+1)\hbar^2}{2mr^2} \right) q_{vl}(r) = \left(V_1 - \frac{1}{2} m \omega^2 r^2 + \epsilon_{vl} \right) q_{vl}(r). \tag{26}$$

Sijoitetaan tämä matriisielementin yhtälöön (25) ja sievennällään hiukan. Tuloksena on seuraava kaava, jolla voidaan laskea matriisielementit kohtuullisen pienellä vaivalla:

$$\begin{aligned}
\langle v' | h_{nlj} | v \rangle &= \int_0^\infty q_{v'l}^*(r) q_{vl}(r) \left(\frac{\hbar^2}{2m} \left(\frac{4v+2l+3}{b^2} - \frac{r^2}{b^4} \right) \right. \\
&\quad \left. + v_{\text{cen}}(r) + v_c(r) + \frac{\hbar^2}{2} \left(j(j+1) - l(l+1) - \frac{3}{4} \right) v_{\text{ls}}(r) \right) dr.
\end{aligned} \tag{27}$$

Nyt yksihiukkastilojen ratkaisemiseksi tarvitsee osata vain numeerinen integrointi ja matriisin diagonalisointi. Kummatkin hommat ovat sellaisia, jotka tietokone osaa minua paremmin.

3 Numeriikka

3.1 Numeerisesta integroinnista

Numeerisissa integrointimenetelmissä on usein ideana se, että vaikeasti integroitavaa funktiota arvioidaan jollakin yksinkertaisemmalla funktiolla. Näin voidaan laskea helppo integraali vaikean sijaan. Koska kaikkien luonnossa tavattavien funktioiden käytös yksinkertaistuu pienemmillä mittakaavoilla, arvion tarkkuutta voi parantaa jakamalla integrointiväli pieniin osiin: kun suurennetaan tarpeeksi, todellisuuden hurjimmatkin funktiot saadaan näyttämään suoralta. Matemaatikot sanoisivat kovin vaikeankuuloisesti, että funktio on derivoituva (ilkeimmät saattaisivat jopa käyttää termiä 'differentioituva').

Kun funktio käyttäytyy tarpeeksi kiltisti, on mahdollista jakaa integrointiväli niin pieniin osiin, että funktiota voidaan arvioida tarkasti suoralla kullakin näistä osaväleistä: kun väli on tarpeeksi pieni, funktion derivaatta ei ehdi muuttua juuri ollenkaan, jolloin funktio ei juurikaan poikkea suorasta. Tätä menettelyä kutsutaan puolisuunnikasäännöksi.

Kuitenkin tehokkaampiakin menetelmiä on olemassa. Tämän tutkielman tarkoituksiin sopivin näistä lienee Simpsonin sääntö. Simpsonin säännössä integroitavaa funktiota arvioidaan suorien sijasta paraabelien avulla. Näin ei tarvitse suurentaa niin paljon, että funktion derivaatta näyttää vakiolta: toinen derivaatta riittää! Käytännössä tämä tarkoittaa sitä, että Simpsonin säännöllä saadaan samalla jakotiheydellä parempi tarkkuus.

3.2 Simpsonin sääntö

Johdetaan Simpsonin sääntö yksittäiselle välille. Tehdään tämä kuitenkin fyysikon näkökulmasta, eli ei jäädä pohtimaan sitä, kuinka monessa pisteessä funktio saa olla epäjatkuva, tai että onko integrointiväli suljettu tai avoin tai kenties jotain muuta. Virheen suuruudestakin tarvitsee tietää vain sen verran, että välin kokoa pienentäessä se lähestyy nollaa integraalin arvoa nopeammin.

Olkoon väli $I := [x_1, x_3]$. Simpsonin säännössä kaksi pistettä saadaan välin päätepisteistä ja kolmas jostain välin sisältä. Varas in indeksin 2 välin sisällä olevalle pisteelle, jotta indeksien ja pisteiden järjestys olisi sama.

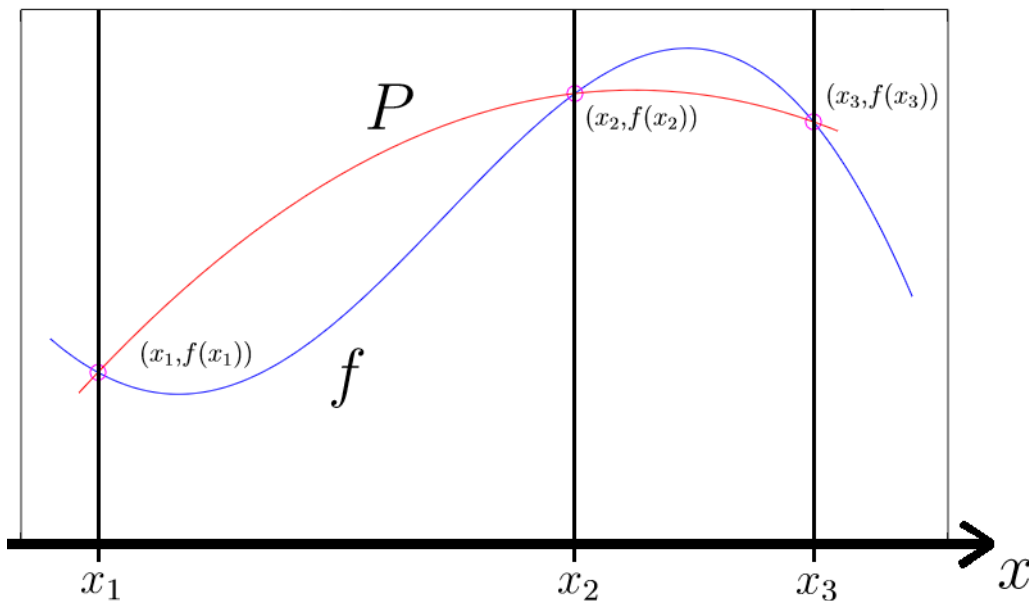
Toisen asteen polynomifunktio, eli y-suuntainen paraabeli, on tietenkin muotoa

$$P(x) = ax^2 + bx + c. \quad (28)$$

Olkoon integroitava funktio nimeltään f . Jos P on tarpeeksi lähellä sitä, niin

$$\int_{x_1}^{x_3} f(x)dx \approx \int_{x_1}^{x_3} P(x)dx = \frac{a}{3}(x_3^3 - x_1^3) + \frac{b}{2}(x_3^2 - x_1^2) + c(x_3 - x_1). \quad (29)$$

Nyt ongelmaksi jää enää vakioiden a , b , ja c asettaminen siten, että P jäljittelee funktiota f tarpeeksi hyvin.



Kuvio 1. Quadraattinen interpolaatio

Vaaditaan, että $f(x)$ ja $y(x)$ ovat yhtäsuuret välin I kummassakin päässä. Koska kiinnitettäviä parametrejä on kolme, tämä ei vielä riitä, vaan tarvitaan vielä yksi piste lisää. Yleensä Simpsonin säännössä piste otetaan täsmälleen välin keskeltä. Itse en kuitenkaan näe tässä yhteydessä mitään syytä rajoittaa menetelmän yleisyyttä, joten kelpuutan minkä tahansa pisteen x_2 , kunhan se sisältyy välille I . Paraabelin P on siis kuljettava pisteiden $(x_1, f(x_1))$, $(x_2, f(x_2))$ ja $(x_3, f(x_3))$ kautta. Tätä vastaa yhtälöryhmä

$$\begin{cases} f(x_1) = ax_1^2 + bx_1 + c \\ f(x_2) = ax_2^2 + bx_2 + c \\ f(x_3) = ax_3^2 + bx_3 + c \end{cases} \quad (30)$$

Ratkaisujen löytäminen on helppoa mutta hiukan työlästä. Jotta ratkaisut saataisiin vähemmän sotkuisen näköisiksi, määritellään erotusosamäärä

$$D(x_j, x_i) = \frac{f(x_j) - f(x_i)}{x_j - x_i}. \quad (31)$$

Yhtälöryhmä (30) voidaan nyt kirjoittaa muodossa

$$\begin{cases} a = \frac{D(x_3, x_2) - D(x_2, x_1)}{x_3 - x_1} \\ b = D(x_2, x_1) - a(x_1 + x_2) \\ c = f(x_1) - ax_1^2 - bx_1 \end{cases} \quad (32)$$

Lasketaan siis ensin a , jolloin saadaan laskettua b , jolloin saadaan laskettua c . Kun nämä sitten heitetään kaavaan (29), saadaan arvio integraalille.

3.3 Äärettömyys ja nolla

Integroitavana on useita sellaisia välejä, joissa esiintyy äärettömyys tai nolla. Äärettömyyden kanssa ongelma on, ettei ääretöntä väliä voi periaatteessa integroida numeerisesti. Nollan kanssa taas on vaarana se, että joihinkin lausekkeisiin saattaa ilmaantua nollalla jako, jolloin laskemisesta ei tietenkään tule mitään.

Nollan käsittelyssä päästään helpolla. Sattuu nimittäin olemaan niin, että tarkasteltavat integrandit lähestyvät nollassa nollaa. Näin ollen tarkkuus pysyy hyvänä, vaikka ei integroitaisi aivan nolnaan asti. Toteutetaan tämä erittäin yksinkertaisella tempulla: kun integrointiväli jaetaan osiin, jätetään ensimmäinen väli pois:

$$\sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} f(x) dx = \int_{x_0}^{x_1} f(x) dx + \sum_{i=1}^{N-1} \int_{x_i}^{x_{i+1}} f(x) dx \approx \sum_{i=1}^{N-1} \int_{x_i}^{x_{i+1}} f(x) dx. \quad (33)$$

Välin $[x_0, x_1]$ pituus on kääntäen verrannollinen jakotiheyteen. Vakiofunktion tapauksessa tarkkuus olisi siis ensimmäistä astetta. Kuitenkin, jos myös itse funktio lähestyy välin pituuden pienentyessä nollaa, tarkkuus on tätä parempi.

Äärettömyyksistä voidaan päästä eroon samantyyppisellä tempulla. Ehtona on kuitenkin se, että funktio lähestyy nollaa riittävän nopeasti. Matriisielementin kaavassa (27) esiintyvä integraali täyttää tämän vaatimuksen oikein hyvin. Harmonisen oskillaattorin aaltofunktiot pienenevät nimittäin Gaussisesti vetäen kaiken muun mukanaan. Näin ollen voidaan valita jokin kyllin suuri katkaisukohta x_M , ja jättää loput integraalista huomioimatta:

$$\int_{x_0}^{\infty} f(x) dx = \int_{x_0}^{x_M} f(x) dx + \int_{x_M}^{\infty} f(x) dx \approx \int_{x_0}^{x_M} f(x) dx. \quad (34)$$

Katkaisukohdan asettamisessa kannattaa palata fysiikkaan. Atomiydin on säteeltään tyypillisesti 2–6 femtometriä.[7] ”Kyllin suuri” on siis tässä tapauksessa kokoluokkaa 10 femtometriä. Käytännössä sopiva katkaisukohta löytyy tarkemmin kokeilemalla.

Myös sähköpotentiaalin lausekkeessa (9) esiintyy äärettömän välin integraali:

$$v_c(r) = q_e \frac{\rho_{C0}}{\varepsilon_0} \int_r^\infty \frac{1}{r'^2} \int_0^{r'} \frac{r''^2}{1 + \exp \frac{r'' - R_C}{a_C}} dr'' dr'. \quad (35)$$

Ensimmäisenä silmään pistää kaksi integraalia sisäkkäin: sisempi integraali määrää funktion, josta ulomman integraalin arvo riippuu. Potentiaalin laskemiseksi täytyy siis laskea lukuisia integraaleja, joista jokaisen laskemiseksi täytyy laskea lukuisia integraaleja.

Löytyy kuitenkin vielä parempi syy pelätä kyseistä integraalia. Integroitava funktio nimittäin lähestyy nollaa varsin hitaasti. Tämän olisi voinut arvata mm. Coulombin laista. Katkaisukohdan on siis oltava hyvin suuri. Jos integroitava väli on pitkä, niin saman jakotiheyden saavuttamiseksi on käytettävä enemmän jakopisteitä. Ja enemmän jakopisteitä tarkoittaa enemmän laskuja.

Integraalista saadaan paljon helpompi, kunhan vain tajuaa sen, että ydin näyttää kaukaa katsottuna pistevaraukselta. Oikeastaan ei tarvitse katsoa kovin kaukaakaan, koska ydin näyttää vähän lähempääkin varatulta pallolta, jolle potentiaalin lauseke on tunnetusti pistevarauksen kanssa sama (kunhan ei olla pallon sisällä vaan ulkopuolella). Käsitellään siis ydintä pistevarauksena hieman ytimen säteen ulkopuolella. Annetaan tälle kohdalle nimeksi vaikkapa R_\odot . Nyt potentiaalin lauseke saadaan numeriikan kannalta paljon helpompaan muotoon:

$$v_c(r) = \begin{cases} q_e \frac{\rho_{C0}}{\varepsilon_0} \int_r^{R_\odot} \frac{1}{r'^2} \int_0^{r'} \frac{r''^2}{1 + \exp \frac{r'' - R_C}{a_C}} dr'' dr' + \frac{q_e^2}{4\pi\varepsilon_0} \frac{Z-1}{R_\odot}, & r < R_\odot \\ \frac{q_e^2}{4\pi\varepsilon_0} \frac{Z-1}{r}, & r \geq R_\odot \end{cases}. \quad (36)$$

R_\odot pitää valita siten, että käytännön kannalta kaikki ytimen varaus jää sen sisäpuolelle. Sen arvolla ei muuten ole niin väliä. Valitaan vaikkapa 16 femtometriä. Tällöin se on reippaasti ytimen sädettä suurempi, mutta ei kuitenkaan turhan suuri.

4 Ohjelman esittely

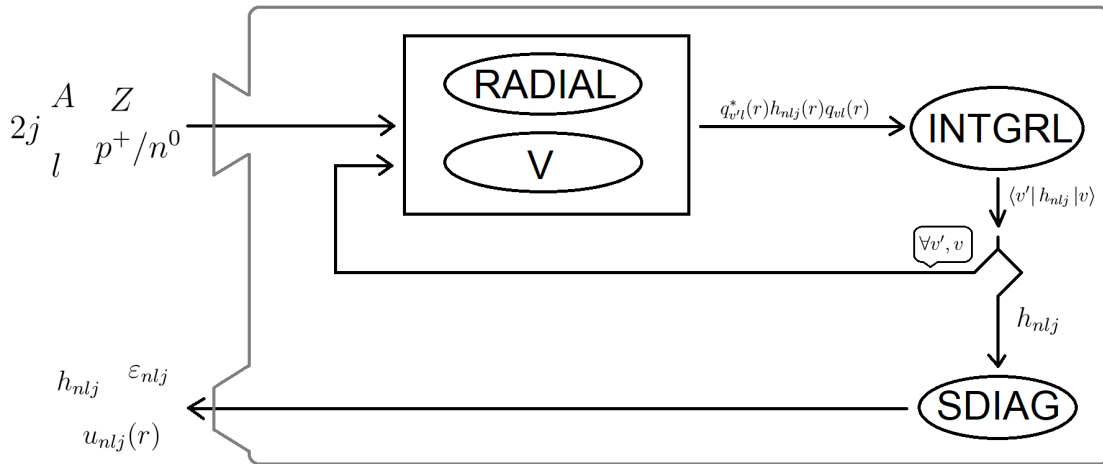
Ohjelma koostuu viidestä palasesta: pääohjelmasta, harmonisen oskillaattorin radiaaliset aaltofunktiot laskevasta osasta, symmetrisen matriisin diagonalisoijasta, integraattorista ja potentiaalista. Diagonalisoija ja radiaaliset funktiot laskeva osa on laitettu pääohjelman kanssa samaan tiedostoon, jottei ohjelman kasaamisessa tarvitsisi kirjoittaa erikseen niin montaa tiedostonimeä.

Ohjelman rakenne on esitetty kuviossa 2. Ensin syötetään A , Z , l ja $2j$, sekä nukleonin tyyppi. Nukleonin tyyppi on joko 0 tai 1: neutronille 0, protonille 1. Syötteen antamisen jälkeen rakennetaan yhtälön (27) integraalin sisällä oleva funktio radiaalinen ja potentiaalinen avulla. Seuraavaksi funktio integroidaan, jolloin saadaan tuloksena matriisielementti. Kun edellä mainitut askeleet on toistettu kaikille eri v' ja v , koko Hamiltonin matriisi on saatu laskettua. Tämän jälkeen matriisi diagonalisoidaan. Lopuksi ruudulle tulostuu Hamiltonin matriisi, aaltofunktiot ja näitä vastaavat energiat.

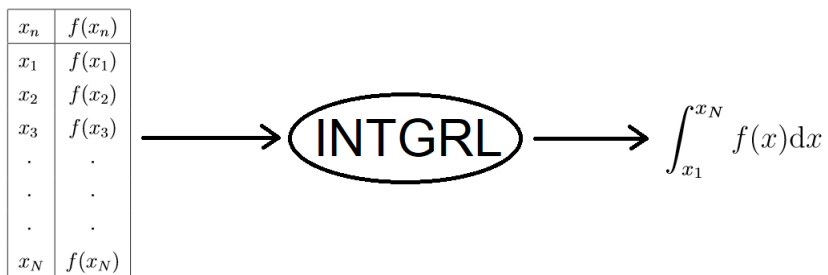
Integraattori ottaa integroitavan funktion vastaan valmiiksi pisteiksi jaettuna, kuten kuviossa 3 on havainnollistettu. Potentiaalia taas käsitellään yksinkertaisesti reaalfunktiona, joka rakentuu potentiaalinen eri termeistä, joista kukin käsitellään myöskin reaalfunktiona. Tämä ei ole läheskään optimaalisin tapa laskunopeuden kannalta, mutta näin voidaan olla sotkematta koodin eri osia keskenään, ja yleiskäyttöisyys säilyy.

Ohjelma on suunniteltu modulaariseksi. Esimerkiksi puolisuunnikasääntöön tai tavalliseen Woods–Saxoniin vaihtamiseksi ei tarvitse muokata koodia mitenkään, vaan integraattorin ja potentiaalinen valinta tehdään ohjelman kasaamisen yhteydessä.

Ohjelman lähdekoodi on sisällytetty tutkielmaan liitteenä B. Mukana on myös puolisuunnikasääntöä käyttävä integraattori ja tavallinen Woods–Saxon.



Kuvio 2. Ohjelman kokonaisrakenne



Kuvio 3. Integraattorin toiminta

5 Tulokset

5.1 Pb-208

Ohjelman antamia tuloksia ytimelle Pb-208 on verrattu Kouran ja Yamadan alkuperäisiin tuloksiin taulukoissa 1 ja 2. Laskemani arvot on merkattu heittomerkillä.

Taulukko 1. Neutronien tiloja ytimessä Pb-208 (MeV)

| | OA' | OB' | HD' | OA | OB | HD | EXP |
|----------|--------|--------|--------|--------|--------|--------|--------|
| $1/2^-$ | -7,27 | -7,28 | -7,23 | -7,28 | -7,29 | -7,24 | -7,37 |
| $5/2^-$ | -7,88 | -7,88 | -7,81 | -7,89 | -7,89 | -7,82 | -7,94 |
| $3/2^-$ | -7,98 | -7,99 | -7,97 | -7,99 | -8,00 | -7,98 | -8,27 |
| $13/2^+$ | -9,16 | -9,14 | -9,23 | -9,17 | -9,15 | -9,24 | -9,00 |
| $7/2^-$ | -9,97 | -9,97 | -9,99 | -9,98 | -9,99 | -10,00 | -9,71 |
| $9/2^-$ | -10,81 | -10,78 | -10,68 | -10,82 | -10,79 | -10,69 | -10,78 |

Taulukko 2. Protonien tiloja ytimessä Pb-208 (MeV)

| | OA' | OB' | HD' | OA | OB | HD | EXP |
|----------|--------|--------|--------|--------|--------|--------|--------|
| $1/2^+$ | -8,13 | -7,94 | -7,94 | -7,92 | -7,76 | -7,77 | -8,01 |
| $3/2^+$ | -8,82 | -8,63 | -8,63 | -8,44 | -8,33 | -8,32 | -8,36 |
| $11/2^-$ | -10,25 | -10,05 | -10,09 | -9,25 | -9,25 | -9,30 | -9,36 |
| $5/2^+$ | -10,21 | -10,03 | -10,03 | -9,92 | -9,79 | -9,79 | -9,69 |
| $7/2^+$ | -12,25 | -12,10 | -12,00 | -11,64 | -11,61 | -11,50 | -11,48 |

Neutronien energiat ovat käytännössä samat. Tulokset näyttäisivät poikkeavan toisistaan aina noin 0,01 MeV:in verran. Poikkeama on pieni, mutta toistuu samana kaikille tiloille. Kyseessä lienee eroavuus käytettyjen vakioiden arvoissa. Epäilen erityisesti nukleonin massaa.

Protonien arvoissa on huomattavasti enemmän heittoa. Tämä johtuu suurimmaksi osaksi siitä, että Kouran ja Yamadan arvoihin on sisällytetty ns. Coulomb-korjaus.

5.2 Zr-90

Vertailu on esitetty neutroneille taulukossa 3 ja protoneille taulukossa 4. Kokeellisesti epävarmat energiat on kirjoitettu kursivilla.

Taulukko 3. Neutronien tiloja ytimestä Zr-90 (MeV)

| | OA' | OB' | HD' | OA | OB | HD | EXP |
|------------------|--------|--------|--------|--------|--------|--------|---------------|
| 9/2 ⁺ | -11,65 | -11,63 | -11,85 | -11,66 | -11,64 | -11,86 | -11,97 |
| 1/2 ⁻ | -13,44 | -13,00 | -13,19 | -13,45 | -13,01 | -13,20 | -12,56 |
| 3/2 ⁻ | -14,95 | -14,48 | -14,78 | -14,95 | -14,49 | -14,79 | <i>-13,06</i> |
| 5/2 ⁻ | -14,84 | -14,82 | -14,73 | -14,85 | -14,83 | -14,74 | -13,42 |

Taulukko 4. Protonien tiloja ytimestä Zr-90 (MeV)

| | OA' | OB' | HD' | OA | OB | HD | EXP |
|------------------|-------|-------|-------|-------|-------|-------|---------------|
| 1/2 ⁻ | -8,09 | -7,76 | -7,86 | -7,93 | -7,69 | -7,78 | -8,36 |
| 3/2 ⁻ | -9,42 | -9,10 | -9,29 | -9,35 | -9,10 | -9,28 | <i>-9,87</i> |
| 5/2 ⁻ | -9,60 | -9,85 | -9,52 | -9,33 | -9,70 | -9,36 | <i>-10,10</i> |

Neutronien energiat ovat keskenään samat, eli siis ohjelma toistaa Kouran ja Yamadan tulokset. Kokeellisiin tuloksiin vertaamalla voi huomata, että energiat ennustetaan 9/2⁺-tilaa lukuun ottamatta aika pahasti pieleen.

Protonien energioissa on Coulomb-korjauksen takia eroa. Energiat ennustetaan jälleen pieleen, mutta tällä kertaa toiseen suuntaan. Seurauksena Coulomb-korjatut arvot ovat tällä kertaa korjaamattomia kauempana todellisista.

5.3 Ca-48

Energiat on esitetty taulukoissa 5 ja 6.

Taulukko 5. Neutronien tiloja ytimestä Ca-48 (MeV)

| | OA' | OB' | HD' | OA | OB | HD | EXP |
|---------|--------|--------|--------|--------|--------|--------|--------|
| $7/2^-$ | -9,23 | -8,97 | -9,33 | -9,24 | -8,98 | -9,34 | -9,95 |
| $3/2^+$ | -14,04 | -13,59 | -13,45 | -14,05 | -13,60 | -13,46 | -12,53 |
| $1/2^+$ | -13,82 | -13,09 | -13,51 | -13,84 | -13,10 | -13,53 | -12,55 |

Taulukko 6. Protonien tiloja ytimestä Ca-48 (MeV)

| | OA' | OB' | HD' | OA | OB | HD | EXP |
|---------|--------|--------|--------|--------|--------|--------|--------|
| $1/2^+$ | -16,54 | -16,38 | -16,43 | -15,75 | -15,61 | -15,65 | -15,81 |
| $3/2^+$ | -16,77 | -17,28 | -16,37 | -16,01 | -16,54 | -15,63 | -16,17 |

Neutronien tilat ovat jälleen kerran Kouran ja Yamadan mukaan noin 0,01 MeV:in verran vahvemmin sidottuja. Arvot poikkeavat merkittävästi kokeellisista. Protonien tilojen kohdalla ilmenee Coulomb-korjauksen tärkeys, sillä laskemani korjaamattomat energiat ovat merkittävästi enemmän pielessä. Poikkeuksena HD-parametrisetillä protonin $3/2^+$ -tila on lähempänä oikeaa ilman Coulomb-korjausta.

5.4 C-14

Tulokset ytimelle C-14 on esitetty taulukoissa 7 ja 8.

Neutroneille ohjelman antamat tulokset yhtenevät Kouran ja Yamadan tuloksiin. Ero kokeellisiin arvoihin on valtava varsinkin neutronin $1/2^-$ -tilan kohdalla. Parametrisetti OB ennustaa Coulomb-korjattuna protonin $3/2^-$ -tilan oikein.

Taulukko 7. Neutronien tiloja ytimessä C-14 (MeV)

| | OA' | OB' | HD' | OA | OB | HD | EXP |
|---------|--------|--------|--------|--------|--------|--------|--------|
| $1/2^-$ | -6,25 | -5,92 | -5,93 | -6,25 | -5,92 | -5,94 | -8,18 |
| $3/2^-$ | -12,86 | -12,38 | -12,97 | -12,86 | -12,38 | -12,97 | -11,86 |

Taulukko 8. Protonin $3/2^-$ -tila ytimessä C-14 (MeV)

| | OA' | OB' | HD' | OA | OB | HD | EXP |
|---------|--------|--------|--------|--------|--------|--------|--------|
| $3/2^-$ | -20,03 | -21,32 | -20,35 | -19,55 | -20,82 | -19,85 | -20,83 |

5.5 He-4

Tulokset ytimelle He-4 on esitetty taulukoissa 9 ja 10.

Ohjelmalla laskettu energia neutronin $1/2^+$ -tilalle poikkeaa merkittävästi Kouran ja Yamadan ilmoittamasta. Tämä on yllättävää. Neutronin tilan kohdalla ero ei nimittäin voi tulla Coulomb-korjauksesta.

Taulukko 9. Neutronin $1/2^+$ -tila ytimessä He-4 (MeV)

| | OA' | OB' | HD' | OA | OB | HD | EXP |
|---------|--------|--------|--------|--------|--------|--------|--------|
| $1/2^+$ | -20,01 | -20,42 | -20,67 | -20,13 | -20,54 | -20,75 | -20,58 |

Taulukko 10. Protonin $1/2^+$ -tila ytimessä He-4 (MeV)

| | OA' | OB' | HD' | OA | OB | HD | EXP |
|---------|--------|--------|--------|--------|--------|--------|--------|
| $1/2^+$ | -19,29 | -19,70 | -19,95 | -19,46 | -19,86 | -20,08 | -19,81 |

6 Yhteenveto

Kouran ja Yamadan tulokset ovat Coulomb-korjauksia vailla varmistettu. Ohjelmalla laskettuna neutronin tilat ovat tyypillisesti noin 0,01 MeV:iä heikommin sidotut, mikä selittyy pienillä eroilla käytetyissä vakioissa.

Protonien kohdalla tuloksia on hieman vaikeampi arvioida. Neutronien tulosten pohjalta lienee turvallista olettaa, että erot protonien energioissa tulevat lähes kokonaan Coulomb-korjauksista. Siten Coulomb-korjaus olisi suuruudeltaan tyypillisesti jotain 0,3 MeV:iä. Sen suuruus näyttäisi kuitenkin vaihtelevan paljon tilasta riippuen.

Tulosten suurin kummajainen oli ytimen He-4 neutronin $1/2^+$ -tila, jossa energiat poikkesivat Kouran ja Yamadan ilmoittamista peräti 0,1 MeV:iä. Koska kyse on neutronista, Coulomb-korjaus ei voi olla syytä. Epäilen vahvasti, että virhe syntyy diagonalisoinnin yhteydessä.

Diagonalisoija olettaa matriisin olevan symmetrinen, mikä ei ole ollenkaan huono oletus silloin, kun diagonalisoitavana on Hamiltonin matriisi. Tässä tapauksessa Hamiltonin matriisi ei kuitenkaan ole tismalleen symmetrinen, kuten yhtälöstä (27) voi päätellä. Yleensä tämä ei ole ongelma, mutta jos oskillaattorin pituus b on pieni, niin matriisi voi olla varsin epäsymmetrinen, ja siten virhe voi olla suuri. Sovellettaessa kevyimpiin ytimiin, suosittelen vaihtamaan diagonalisointimenetelmän johonkin yleisempään.

Lähteet

- [1] H. Koura, M. Yamada: *Nuclear Physics A* 671, 96 (2000)
- [2] Jouni Suhonen: *From Nucleons to Nucleus*, Springer, 2010, s. 40
- [3] H.-W. Hammer, A. Nogga, A. Schwenk: *Rev. Mod. Phys.* 85, 197 (2013)
- [4] David J. Griffiths: *Introduction to Quantum Mechanics*, Pearson, 2005, s. 169
- [5] Jouni Suhonen: *From Nucleons to Nucleus*, Springer, 2010, s. 49
- [6] J. Blomqvist, A. Molinari: *Nuclear Physics A* 106, 545 (1968)
- [7] I. Angeli, K. P. Marinova: Table of experimental nuclear ground state charge radii: An update, *Atomic Data and Nuclear Data Tables* 99, 69 (2013)
- [8] Randall D. Knight, *Physics for Scientists and Engineers*, Pearson, 2014, s. 889
- [9] Randall D. Knight, *Physics for Scientists and Engineers*, Pearson, 2014, s. 946

A Coulomb-termin johto

Ytimellä on varaustiheys

$$\rho_C(r) = \frac{\rho_{C0}}{1 + \exp \frac{r-R_C}{a_C}}. \quad (37)$$

Lasketaan tätä vastaava sähkökenttä ja siitä edelleen potentiaali.

Gaussin laki on integraalimuodossa

$$\Phi = \iint_{\text{pinta}} \mathbf{E} \cdot d\mathbf{A} = \frac{Q}{\varepsilon_0}, [8] \quad (38)$$

missä Q on pinnan sisälle suljettu kokonaisvaraus. Varaustiheys riippuu vain säteestä r , eli se on pallosymmetrinen. Tästä seuraa, että sähkökentän on osoitettava kaikkialla suoraan origosta ulospäin (tai negatiivisen varauksen tapauksessa suoraan sisäänpäin). Pinnaksi järkevin valinta on siis pallo. Tällöin sähkökenttä on kaikkialla pinnalla yhtä suuri ja pinnan normaalin suuntainen. Nyt Gaussin laki yksinkertaistuu seuraavanlaiseksi:

$$\Phi = E\mathcal{A} = E \cdot 4\pi r^2 = \frac{Q}{\varepsilon_0} \iff E = \frac{1}{4\pi\varepsilon_0} \frac{Q}{r^2} = E(r). \quad (39)$$

Pinnan sisällä oleva varaus saadaan integroimalla varaustiheys pallon tilavuuden yli:

$$Q = \iiint_{\text{O}} \rho_C(r') d\mathcal{V}' = 4\pi\rho_{C0} \int_0^r \frac{r'^2}{1 + \exp \frac{r'-R_C}{a_C}} dr' = Q(r). \quad (40)$$

Vakio ρ_{C0} saadaan kiinnitettyä integroimalla koko avaruuden yli. Tällöin pinnan sisään jää koko ytimen varaus (kentässä olevaa protonia lukuun ottamatta):

$$\rho_{C0} = \frac{(Z-1)q_e}{4\pi} \left(\int_0^\infty \frac{r'^2}{1 + \exp \frac{r'-R_C}{a_C}} dr' \right)^{-1}. \quad (41)$$

Sähkökentän suuruus on

$$E(r) = \frac{\rho_{C0}}{\varepsilon_0} \frac{1}{r^2} \int_0^r \frac{r'^2}{1 + \exp \frac{r' - R_C}{a_C}} dr'. \quad (42)$$

Potentiaali pisteessä \mathbf{r} saadaan integroimalla sähkökenttää tästä pisteestä jotakin polkua pitkin nollapotentiaaliin.[9] Toisin sanoen siis

$$V_E(\mathbf{r}) = \int_{\mathbf{r} \rightsquigarrow \perp} \mathbf{E} \cdot d\boldsymbol{\ell}. \quad (43)$$

Potentiaali on valittu menemään nollassi äärettömän kaukana origosta. Käytännöllisin polku lienee pisteestä \mathbf{r} alkava puolisuora, joka osoittaa suoraan origosta poispäin. Näin sähkökenttä on aina polun kanssa samansuuntainen, ja polkuintegraali muuttuu tavalliseksi integraaliksi:

$$V_E(\mathbf{r}) = \frac{\rho_{C0}}{\varepsilon_0} \int_r^\infty \frac{1}{r'^2} \int_0^{r'} \frac{r''^2}{1 + \exp \frac{r'' - R_C}{a_C}} dr'' dr' = V_E(r). \quad (44)$$

Schrödinger-yhtälössä esiintyvä potentiaali on oikeastaan potentiaalienergia eikä potentiaali. Potentiaalienergia saadaan potentiaalista kertomalla hiukkasen varauksella. Neutronille varaus on nolla, joten myös sähköinen potentiaalienergia on nolla. Protonille taas varaus on tietysti tismalleen yksi alkeisvaraus, joten protonin sähköinen potentiaalienergia on

$$v_c(r) = q_e V_E(r) = q_e \frac{\rho_{C0}}{\varepsilon_0} \int_r^\infty \frac{1}{r'^2} \int_0^{r'} \frac{r''^2}{1 + \exp \frac{r'' - R_C}{a_C}} dr'' dr'. \quad (45)$$

B Lähdekoodi

Main.for

```
1      PROGRAM PROG
2      INTEGER, PARAMETER :: K = 180 ! Askelmäärä <<<<<<
3      REAL, PARAMETER :: RMAX = 18 ! Suurin r:n arvo (fm) <<<<<<
4      INTEGER, PARAMETER :: NBAS = 8 ! Kantafunktioiden suurin n <<<<<<
5
6      C -----
7      REAL, PARAMETER :: HBAR = 197.32697 ! Redusoitu Planckin vakio (MeV*fm)
8      REAL, PARAMETER :: MN = 939.5654133 ! Neutronin massa (MeV)
9      REAL, PARAMETER :: MP = 938.2720813 ! Protonin massa (MeV)
10
11     C -----
12     REAL, PARAMETER :: STEP = RMAX/K ! Askeleen pituus
13     REAL :: INTG ! Integraalin arvo
14     REAL :: ZFIX ! Arvio integraalista nollan lähellä
15
16     C -----
17     REAL, DIMENSION(2, K) :: FX ! Integroitava funktio pistejoukkona
18     REAL, DIMENSION(NBAS+1, NBAS+1) :: H ! Hamiltonin operaattorin matriisiesitys
19     REAL, DIMENSION(NBAS+1) :: EVAL ! Energiat säilötään tänne
20     REAL, DIMENSION(NBAS+1, NBAS+1) :: EVEC ! Aaltofunktiot säilötään tänne
21     REAL, DIMENSION(NBAS+1) :: AUX ! Joku ihmevitkutin, jota SDIAG tarvitsee
22
23     C -----
24     INTEGER :: A ! Massaluku [INPUT]
25     INTEGER :: Z ! Järjestysluku [INPUT]
26     INTEGER :: L ! Ratapyörimismäärä l [INPUT]
27     INTEGER :: J2 ! Pyörimismäärä j kertaa kaksi [INPUT]
28     INTEGER :: Q ! 0: neutroni, 1: protoni [INPUT]
29
30     C -----
31     INTEGER :: N ! Neutroniluku
32     REAL :: M ! Nukleonin massa
33     REAL :: HW ! Oskillaattorin kulmataajuus
34     REAL :: B ! Oskillaattorin pituus
35
36     C -----
37     WRITE (*, *) "INPUT: A, Z, L, J2, Q"
38     READ (*, *) A, Z, L, J2, Q
39     WRITE (*, *) ""
40     WRITE (*, *) "MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM"
41     WRITE (*, *) ""
42     WRITE (*, *) "A = ", A
43     WRITE (*, *) "Z = ", Z
44     WRITE (*, *) "L = ", L
45     WRITE (*, *) "J2 = ", J2
46     WRITE (*, *) "Q = ", Q
47     WRITE (*, *) ""
48     WRITE (*, *) "MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM"
49     N = A - Z
50     M = (MN*N + MP*Z)/A ! Nukleonin massaksi keskiarvo
51     HW = 45*A**(-1./3) - 25*A**(-2./3) ! Blomqvist--Molinari (MeV)
52     B = HBAR*(M*HW)**(-1./2)
53
54     C -----
55     WRITE (*, *) ""
56     DO NL = 0, NBAS
57     DO NR = 0, NBAS
58     WRITE (*, *) "NOW CALCULATING...", NL, NR
59     DO I = 1, K ! Luodaan taulukko integrointia varten...
60     FX(1, I) = I*STEP ! |0, RMAX]
61     FX(2, I) = FX(1, I)**2*RADIAL(NL, L, B, FX(1, I))
62     *RADIAL(NR, L, B, FX(1, I))
63     1 * (HBAR**2/(2*M) * ((4*NR+2*L+3)/B**2 - FX(1, I)**2/B**4)
64     2 + V(FX(1, I), A, Z, J2, L, Q))
65     3
66     END DO
67     CALL INTGRL(K, FX, INTG) ! Integroidaan...
68     ZFIX = -0.5*FX(1, 1)**2*(FX(2, 2) - FX(2, 1)) !| Arvioidaan funktiota nollan lähellä
69     / (FX(1, 2) - FX(1, 1)) !| lineaarisella extrapoloinnilla
70     H(NL+1, NR+1) = INTG + ZFIX ! Matriisielementti matriisiin
71
72     END DO
73     END DO
74
75     WRITE (*, *) ""
76     WRITE (*, *) "DONE!"
77     WRITE (*, *) ""
78     WRITE (*, *) "DIAGONALIZING..."
79     CALL SDIAG(NBAS+1, NBAS+1, H, EVAL, EVEC, AUX, 0) ! Diagonalisoidaan...
80
81     C -----
82     WRITE (*, *) ""
83     WRITE (*, *) "DONE!"
84     WRITE (*, *) ""
85     WRITE (*, *) "MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM"
86     WRITE (*, *) ""
87     WRITE (*, *) "HAMILTONIAN"
88     WRITE (*, *) "_____|"
89     WRITE (*, *) ""
90     DO I = 1, NBAS+1
91     WRITE (*, '(*(F8.3)') H(I, :)
92     END DO
93     WRITE (*, *) ""
94     WRITE (*, *) ""
95     WRITE (*, *) "WAVE FUNCTIONS"
96     WRITE (*, *) "_____|"
97     WRITE (*, *) ""
98     DO I = 1, NBAS+1
99     WRITE (*, '(*(F8.3)') EVEC(I, :)
100    END DO
101    WRITE (*, *) ""
102    WRITE (*, *) ""
103    WRITE (*, *) "ENERGIES"
104    WRITE (*, *) "_____|"
105    WRITE (*, *) ""
106    WRITE (*, '(*(F8.3)') EVAL
107    WRITE (*, *) ""
108    WRITE (*, *) ""
109    READ (*, *) ! PAUSE
110
111    END PROGRAM
```

```

100 C
101 C
102 C
103 C
104 REAL FUNCTION RADIAL(N,L,BEE,X)
105 REAL*8 RHO,VEE(0:30,0:12)
106 REAL*8 FACTO,FACT
107 COMMON/FACTI/FACTO(0:40)
108 COMMON/FACTORI/FACT(0:30)
109 DATA PII/3.141592654/
110 *****
111 C READ(5,*)N,L,BEE,X ! n,l,b,r
112 RNU=1/(BEE*BEE)
113 CALL RFACOR
114 CALL RFACFA
115 RHO=DBLE(RNU*X*X) ! here original units
116 CCC RHO=DBLE(X*X) ! here scaled units for integration
117 CALL VEET(RHO,VEE)
118 FACTO1=SNGL(FACTO(L+N))
119 FACTO2=SNGL(FACTO(L))
120 FACTO3=SNGL(FACT(N))
121 APU1=SQRT(2.0*RNU) ! here original units
122 CCC APU1=SQRT(2.0) ! here scaled units for integration
123 APU1=APU1**(2*L+3)
124 APU2=SQRT(2.0/PII)
125 CC PRINT*, 'FACTO1, FACTO2, FACTO3, APU1, APU2: ',
126 CC & FACTO1, FACTO2, FACTO3, APU1, APU2
127 RNORM=FACTO1*APU1*APU2/((2.0**N)*FACTO3)
128 RNORM=SQRT(RNORM)/FACTO2
129 VNL=SNGL(VEE(N,L))
130 RADAPU=1.0
131 IF(L.GT.0)RADAPU=X**L
132 RADIAL=RNORM*EXP(-0.5*RNU*X*X)*RADAPU*VNL ! original
133 CCC RADIAL=RNORM*EXP(-0.5*X*X)*RADAPU*VNL ! scaled
134 C PRINT*, 'RNORM, RADAPU, VNL: ', RNORM, RADAPU, VNL
135 C WRITE(6,100)N,L,X,RADIAL
136 100 FORMAT(/,2X,'n,l,x,R_n1(x): ',11,2X,11,2X,F8.3,5X,1PG15.8)
137 RETURN
138 END
139 C
140 SUBROUTINE VEET(X,VEE)
141 * CALCULATE THE V(N,L,X) FUNCTIONS OF HORIE & SASAKI (RELATED
142 * TO THE ASSOCIATED LAGUERRE POLYNOMIALS) BY USING THE FOLLOWING
143 * RECURSION RELATIONS:
144 * V(N,L-1,X)=V(N-1,L-1,X)-2*X*V(N-1,L,X)/(2*L+1),
145 * V(N,L,X)=(2*L+1)*V(N,L-1,X)+2*N*V(N-1,L,X)/(2*N+2*L+1).
146 * STORE THESE IN TO THE MATRIX VEE(N,L).
147 REAL*8 X,VEE(0:30,0:12)
148 *****
149 DO 100 L=0,12
150 VEE(0,L)=1.00
151 100 CONTINUE
152 N=1
153 200 CONTINUE
154 DO 300 I=1,12
155 VEE(N,L-1)=VEE(N-1,L-1)-2.00*X*VEE(N-1,L)/DFLOAT(2*L+1)
156 VEE(N,L)=(DFLOAT(2*L+1)*VEE(N,L-1)+DFLOAT(2*N)*
157 & VEE(N-1,L))/DFLOAT(2*N+2*L+1)
158 300 CONTINUE
159 N=N+1
160 IF(N.GE.30)GOTO 400
161 GOTO 200
162 400 RETURN
163 END
164 *
165 *****
166 * CALCULATE THE FACTORIALS (2L+1)!! , L=0,...,40 . *
167 *****
168 SUBROUTINE RFACOR
169 IMPLICIT REAL*8(A-H,T-Z)
170 COMMON/FACTI/FACTO(0:40)
171 FACTO(0)=1.00
172 FACTO(1)=3.00
173 DO 100 I=2,40
174 FACTO(I)=DBLE(2*I+1)*FACTO(I-1)
175 100 CONTINUE
176 RETURN
177 END
178 *
179 *****
180 * CALCULATE THE FACTORIALS n! , n=0,...,30 . *
181 *****
182 SUBROUTINE RFACFA
183 IMPLICIT REAL*8(A-H,T-Z)
184 COMMON/FACTORI/FACT(0:30)
185 FACT(0)=1.00
186 FACT(1)=1.00
187 DO 100 I=2,30
188 FACT(I)=DBLE(I)*FACT(I-1)
189 100 CONTINUE
190 RETURN
191 END
192
193
194
195
196
197
198

```

```

199 C=====
200 C
201 C      SUBROUTINE SDIAG(NMAX,N,A,D,X,E,IS)
202 C=====
203 C
204 C
205 C      NMAX    dimension of the matrices
206 C      N       order
207 C      A(i,j)  symmetric matrix to be diagonalized
208 C      D(j)    eigenvalues
209 C      X(i,j)  eigenvector (of components i) for the eigenvalue j
210 C      E(i)    auxiliary field
211 C      IS = 1  eigenvalues are ordered and major component of X is positiv
212 C             0  eigenvalues are not ordered
213 C
214 C-----
215 C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
216 C
217 C      DIMENSION A(NMAX,NMAX),X(NMAX,NMAX),E(N),D(N)
218 C      DATA TOL,EPS/1.E-32,1.E-10/
219 C      IF (N.EQ.1) THEN
220 C          D(1)=A(1,1)
221 C          X(1,1)=1.
222 C          RETURN
223 C      ENDIF
224 C
225 C      DO 10 I=1,N
226 C          DO 10 J=1,I
227 C              10 X(I,J)=A(I,J)
228 C
229 C      CCC  HOUSEHOLDER-REDUKTION
230 C          I=N
231 C          15 IF (I-2) 200,20,20
232 C          20 L=I-2
233 C          F=X(I,I-1)
234 C          G=F
235 C          H=0
236 C          IF (L) 31,31,32
237 C          32 DO 30 K=1,L
238 C          30 H=H+X(I,K)*X(I,K)
239 C          31 S=H+F*F
240 C          IF (S-TOL) 33,34,34
241 C          33 H=0
242 C          GOTO 100
243 C          34 IF (H) 100,100,40
244 C          40 L=L+1
245 C          G=SQRT(S)
246 C          G=DSQRT(S)
247 C          IF (F.GE.0.) G=-G
248 C          H=S-F*G
249 C          HI=1.D0/H
250 C          X(I,I-1)=F-G
251 C          F=0.
252 C          IF (L) 51,51,52
253 C          52 DO 50 J=1,L
254 C          X(J,I)=X(J,I)*HI
255 C          S=0.
256 C          DO 55 K=1,J
257 C          55 S=S+X(J,K)*X(I,K)
258 C          J1=J+1
259 C          IF (L-J1) 57,58,58
260 C          58 DO 59 K=J1,L
261 C          59 S=S+X(K,J)*X(I,K)
262 C          57 E(J)=S*HI
263 C          50 F=F+S*X(J,I)
264 C          51 F=F*HI*.5D0
265 C
266 C
267 C          IF (L) 100,100,62
268 C          62 DO 60 J=1,L
269 C          S=X(I,J)
270 C          E(J)=E(J)-F*S
271 C          F=E(J)
272 C          DO 65 K=1,J
273 C          65 X(J,K)=X(J,K)-S*E(K)-X(I,K)*F
274 C          60 CONTINUE
275 C          100 CONTINUE
276 C          D(I)=H
277 C          E(I-1)=G
278 C          I=I-1
279 C          GOTO 15
280 C      CCC  BEREITSTELLEN DER TRANSFORMATIONSMATRIX
281 C          200 D(1)=0.
282 C          E(N)=0.
283 C          B=0.
284 C          F=0.
285 C          DO 210 I=1,N
286 C          L=I-1
287 C          IF (D(I).EQ.0.) GOTO 221
288 C          IF (L) 221,221,222
289 C          222 DO 220 J=1,L
290 C          S=0.
291 C          DO 225 K=1,L
292 C          225 S=S+X(I,K)*X(K,J)
293 C          DO 226 K=1,L
294 C          226 X(K,J)=X(K,J)-S*X(K,I)
295 C          220 CONTINUE
296 C          221 D(I)=X(I,I)
297 C          X(I,I)=1.
298 C          IF (L) 210,210,232

```

```

298 232 DO 230 J=1,L
299 X(I,J)=0.
300 230 X(J,I)=0.
301 210 CONTINUE
302 CCC DIAGONALISIEREN DER DREIECKSMATRIX
303 DO 300 I=1,N
304 H=EPS*( ABS(D(L))+ ABS(E(L)))
305 IF (H.GT.B) B=H
306 CCC TEST FUER SPLITTING
307 DO 310 J=L,N
308 IF ( ABS(E(J)).LE.B) GOTO 320
309 310 CONTINUE
310 CCC TEST FUER KONVERGENZ
311 320 IF (J.EQ.L) GOTO 300
312 340 F=(D(L+1)-D(L))/(2*E(L))
313 R= DSQRT(P*P+1.D0)
314 PR=P+R
315 IF (P.LT.0.) PR=P-R
316 H=D(L)-E(L)/PR
317 DO 350 I=L,N
318 350 D(I)=D(I)-H
319 F=F+H
320 CCC QR-TRANSFORMATION
321 P=D(J)
322 C=1.D0
323 S=0.
324 I=J
325 360 I=I-1
326 IF (I.LT.L) GOTO 362
327 G=C*E(I)
328 H=C*P
329 IF ( ABS(P)- ABS(E(I))) 363,364,364
330 364 C=E(I)/P
331 R= DSQRT(C*C+1.D0)
332 E(I+1)=S*P*R
333 S=C/R
334 C=1.D0/R
335 GOTO 365
336 363 C=P/E(I)
337 R= DSQRT(C*C+1.D0)
338 E(I+1)=S*E(I)*R
339 S=1.D0/R
340 C=C/R
341 365 P=C*D(I)-S*G
342 D(I+1)=H+S*(C*G+S*D(I))
343 DO 368 K=1,N
344 H=X(K,I+1)
345 X(K,I+1)=X(K,I)*S+H*C
346 368 X(K,I)=X(K,I)*C-H*S
347 GOTO 360
348 362 E(L)=S*P
349 D(L)=C*P
350 IF ( ABS(E(L)).GT.B) GOTO 340
351 CCC KONVERGENZ
352 300 D(L)=D(L)+F
353 C
354 IF (IS.EQ.0) RETURN
355 CCC ORDNERN DER EIGENWERTE
356 DO 400 I=1,N
357 K=I
358 P=D(I)
359 J1=I+1
360 IF (J1-N) 401,401,400
361 401 DO 410 J=J1,N
362 IF (D(J).GE.P) GOTO 410
363 K=J
364 P=D(J)
365 410 CONTINUE
366 420 IF (K.EQ.I) GOTO 400
367 D(K)=D(I)
368 D(I)=P
369 DO 425 J=1,N
370 P=X(J,I)
371 X(J,I)=X(J,K)
372 425 X(J,K)=P
373 400 CONTINUE
374 C
375 C SIGNUM
376 DO 71 K=1,N
377 S=0.
378 DO 72 I=1,N
379 H= ABS(X(I,K))
380 IF (H.GT.S) THEN
381 S=H
382 IM=I
383 ENDIF
384 72 CONTINUE
385 IF (X(IM,K).LT.0.) THEN
386 DO 73 I=1,N
387 73 X(I,K)=-X(I,K)
388 ENDIF
389 71 CONTINUE
390 C
391 C WRITE(*,2000)
392 2000 FORMAT(' *** END SDIAG ***')
393 RETURN
394 C-END-SDIAG
395 END
396

```


kypot.for

```
1   CCC -----
2   CCC | Nuclear Physics A 671 (2000) 96-118 |
3   CCC -----
4
5   CCC -----
6   CCC | HUOM: COULOMB-KORJAUSET LASKETTAVA ERIKSEEN |
7   CCC -----
8
9   REAL FUNCTION V(R, A, Z, J2, L, Q)
10  REAL R
11  INTEGER A
12  INTEGER Z
13  INTEGER J2
14  INTEGER L
15  INTEGER Q
16
17  REAL, PARAMETER :: HBAR = 197.32697 ! Redusoitu Planckin vakio (MeV*fm)
18
19  INTEGER, PARAMETER :: PSET = 1 ! Parametrisetin valinta (1: OA, 2: OB, 3: HD) <<<<<<
20
21  V = VCEN(R, A, Z, Q, PSET)
22  + VC(R, A, Z, Q)
23  + VLS(R, A, Z, Q, PSET)*HBAR**2/2
24  + VLS(R, A, Z, Q, PSET)*HBAR**2/2
25  * (J2/2.* (J2/2.+1) - L*(L+1) - 0.75)
26
27  RETURN
28  END FUNCTION
29
30  C -----
31  REAL FUNCTION VCEN(R, A, Z, Q, PARS) ! Keskeistermi
32  REAL R
33  INTEGER A
34  INTEGER Z
35  INTEGER Q
36  INTEGER PARS
37
38  C -----
39  REAL, PARAMETER :: RM0 = 1.08 ! (<--- fm) r_m0
40  REAL, PARAMETER :: CEXR = 0.005 ! C_exR
41  REAL, PARAMETER :: CEXV = 0.006 ! C_exV
42  REAL, PARAMETER :: RW = 1. ! (<--- fm) r_w
43  REAL :: IPAR
44  REAL :: CW
45
46  C -----
47  REAL :: VV0
48  REAL :: VV1
49  REAL :: VV2
50  REAL :: VV3
51  REAL :: VV4
52  REAL :: VV5
53  REAL :: APHV1
54  REAL :: APHV2
55  REAL :: APHV3
56
57  REAL :: VDP0
58  REAL :: VDP1
59  REAL :: VDP2
60  REAL :: VDP3
61  REAL :: VDP4
62  REAL :: VDP5
63  REAL :: APHVDP1
64  REAL :: APHVDP2
65  REAL :: APHVDP3
66
67  REAL :: D0
68  REAL :: D1
69  REAL :: D2
70  REAL :: D3
71  REAL :: D4
72  REAL :: D5
73  REAL :: APHD1
74  REAL :: APHD2
75  REAL :: APHD3
76
77  REAL :: AA0
78  REAL :: AA1
79  REAL :: AA2
80  REAL :: AA3
81  REAL :: AA4
82  REAL :: AA5
83  REAL :: APHA1
84  REAL :: APHA2
85  REAL :: APHA3
86
87  REAL :: AKAP0
88  REAL :: AKAP1
89  REAL :: AKAP2
90  REAL :: AKAP3
91  REAL :: AKAP4
92  REAL :: AKAP5
93  REAL :: APHKAP1
94  REAL :: APHKAP2
95  REAL :: APHKAP3
96
97  C -----
98  REAL :: V0
99  REAL :: VDP
100 REAL :: RM
101 REAL :: RV
102 REAL :: AV
103 REAL :: AKAPV
104 C -----
```

```

100      INTEGER :: N ! Neutroniluku
101      N = A - Z
102      IF (Q.EQ.0) THEN ! Neutronille parametrit I ja c_w
103          IPAR = 1./A*(N-Z) ! Parametri I neutronille
104          IF (N.GT.Z) THEN ! c_w neutronille
105              CW = -1./A**(2./3)
106          ELSE
107              CW = +1./A**(2./3)
108          ENDIF
109      ENDIF
110      IF (Q.EQ.1) THEN ! Protonille parametrit I ja c_w
111          IPAR = 1./A*(Z-N) ! Parametri I protonille
112          IF (Z.GT.N) THEN ! c_w protonille
113              CW = -1./A**(2./3)
114          ELSE
115              CW = +1./A**(2./3)
116          ENDIF
117      ENDIF
118      C -----
119      IF (PARS.EQ.1) THEN ! OA
120          VV0 = -64.981 ! (<--- MeV) v_0
121          VV1 = 0.8024 ! v_1
122          VV2 = 0.3079 ! v_2
123          VV3 = 1.523 ! v_3
124          VV4 = -1.02 ! v_4
125          VV5 = -1. ! v_5
126          APHV1 = 2. ! alp_v1
127          APHV2 = 0.2 ! alp_v2
128          APHV3 = 0. ! alp_v3
129
130          VDP0 = 0.467 ! v_dp0
131          VDP1 = -2.0167 ! v_dp1
132          VDP2 = -1.1163 ! v_dp2
133          VDP3 = -5.4 ! v_dp3
134          VDP4 = -3. ! v_dp4
135          VDP5 = -0.4 ! v_dp5
136          APHVDP1 = -0.5 ! alp_vdp1
137          APHVDP2 = -0.5 ! alp_vdp2
138          APHVDP3 = 0. ! alp_vdp3
139
140          D0 = 1.5088 ! (<--- fm) d_0
141          D1 = -1.2769 ! (<--- fm) d_1
142          D2 = 11.2 ! (<--- fm) d_2
143          D3 = 11.247 ! (<--- fm) d_3
144          D4 = 6. ! (<--- fm) d_4
145          D5 = -20. ! (<--- fm) d_5
146          APHD1 = 1.5 ! alp_d1
147          APHD2 = 2. ! alp_d2
148          APHD3 = 0.3 ! alp_d3
149
150          AA0 = 1.4449 ! (<--- fm) a_0
151          AA1 = 0.0452 ! a_1
152          AA2 = 7.045 ! a_2
153          AA3 = 1.55 ! a_3
154          AA4 = -0.2 ! a_4
155          AA5 = -15. ! a_5
156          APHA1 = -0.5 ! alp_a1
157          APHA2 = 0. ! alp_a2
158          APHA3 = 2. ! alp_a3
159
160          AKAP0 = 1.508 ! (<--- fm) kap_0
161          AKAP1 = 1.0322 ! kap_1
162          AKAP2 = -0.862 ! kap_2
163          AKAP3 = 1.1763 ! kap_3
164          AKAP4 = -2. ! kap_4
165          AKAP5 = 0.5 ! kap_5
166          APHKAP1 = 2. ! alp_kap1
167          APHKAP2 = 0.1 ! alp_kap2
168          APHKAP3 = -0.5 ! alp_kap3
169      ELSEIF (PARS.EQ.2) THEN ! OB
170          VV0 = -66.546 ! (<--- MeV) v_0
171          VV1 = 0.7668 ! v_1
172          VV2 = 0.02 ! v_2
173          VV3 = 1.76 ! v_3
174          VV4 = -1.0115 ! v_4
175          VV5 = -0.93 ! v_5
176          APHV1 = 2. ! alp_v1
177          APHV2 = 1. ! alp_v2
178          APHV3 = 0.6 ! alp_v3
179
180          VDP0 = 0.5445 ! v_dp0
181          VDP1 = -2.6671 ! v_dp1
182          VDP2 = 0.73 ! v_dp2
183          VDP3 = -4.331 ! v_dp3
184          VDP4 = 2.6917 ! v_dp4
185          VDP5 = 0.5 ! v_dp5
186          APHVDP1 = -0.5 ! alp_vdp1
187          APHVDP2 = -0.4 ! alp_vdp2
188          APHVDP3 = 0. ! alp_vdp3
189
190          D0 = 1.4422 ! (<--- fm) d_0
191          D1 = -0.3909 ! (<--- fm) d_1
192          D2 = 7. ! (<--- fm) d_2
193          D3 = 12.6555 ! (<--- fm) d_3
194          D4 = -0.9991 ! (<--- fm) d_4
195          D5 = -16. ! (<--- fm) d_5
196          APHD1 = 2. ! alp_d1
197          APHD2 = 1.5 ! alp_d2
198          APHD3 = 1. ! alp_d3

```

```

199
200 AA0 = 1.4305 ! (<--- fm) a_0
201 AA1 = 0. ! a_1
202 AA2 = 5.15 ! a_2
203 AA3 = 1.9391 ! a_3
204 AA4 = -0.1 ! a_4
205 AA5 = -14.3 ! a_5
206 APHA1 = -0.5 ! alp_a1
207 APHA2 = 0. ! alp_a2
208 APHA3 = 2. ! alp_a3
209
210 AKAP0 = 1.45 ! (<--- fm) kap_0
211 AKAP1 = 1. ! kap_1
212 AKAP2 = -0.5 ! kap_2
213 AKAP3 = 0.7729 ! kap_3
214 AKAP4 = -2.4 ! kap_4
215 AKAP5 = 0.2 ! kap_5
216 APHKAP1 = 1.6 ! alp_kap1
217 APHKAP2 = 1. ! alp_kap2
218 APHKAP3 = -0.5 ! alp_kap3
219 ELSEIF (PARS.EQ.3) THEN ! HD
220 VV0 = -63.706 ! (<--- MeV) v_0
221 VV1 = 0.7748 ! v_1
222 VV2 = -0.05 ! v_2
223 VV3 = 1.3729 ! v_3
224 VV4 = -0.7464 ! v_4
225 VV5 = 0. ! v_5
226 APHV1 = 1.5 ! alp_v1
227 APHV2 = 0. ! alp_v2
228 APHV3 = 1. ! alp_v3
229
230 VDP0 = 0.5349 ! v_dp0
231 VDP1 = -2.4436 ! v_dp1
232 VDP2 = 0.76 ! v_dp2
233 VDP3 = -4.583 ! v_dp3
234 VDP4 = 1.5088 ! v_dp4
235 VDP5 = 0.5 ! v_dp5
236 APHVDP1 = -0.4 ! alp_vdp1
237 APHVDP2 = -0.4 ! alp_vdp2
238 APHVDP3 = 0. ! alp_vdp3
239
240 D0 = 1.258 ! (<--- fm) d_0
241 D1 = -0.3876 ! (<--- fm) d_1
242 D2 = 6.36 ! (<--- fm) d_2
243 D3 = 11.0095 ! (<--- fm) d_3
244 D4 = -0.942 ! (<--- fm) d_4
245 D5 = -14. ! (<--- fm) d_5
246 APHD1 = 1.73 ! alp_d1
247 APHD2 = 1.5 ! alp_d2
248 APHD3 = 0.7 ! alp_d3
249
250 AA0 = 1.298 ! (<--- fm) a_0
251 AA1 = 0. ! a_1
252 AA2 = 5.3 ! a_2
253 AA3 = 1.3773 ! a_3
254 AA4 = 0.085 ! a_4
255 AA5 = -14. ! a_5
256 APHA1 = -0.5 ! alp_a1
257 APHA2 = 0. ! alp_a2
258 APHA3 = 1.8 ! alp_a3
259
260 AKAP0 = 1.4481 ! (<--- fm) kap_0
261 AKAP1 = 1. ! kap_1
262 AKAP2 = -0.4 ! kap_2
263 AKAP3 = 0.7681 ! kap_3
264 AKAP4 = -2.4 ! kap_4
265 AKAP5 = 0.2 ! kap_5
266 APHKAP1 = 2. ! alp_kap1
267 APHKAP2 = 1. ! alp_kap2
268 APHKAP3 = -0.5 ! alp_kap3
269
270 C - - - - -
271 V0 = VV0*(1-VV1*IPAR-VV2*IPAR**2
272 1 -VV3/(A**(1./3)+APHV1)-VV4*IPAR/(A**(1./3)+APHV2)
273 2 -VV5*IPAR**2/(A**(1./3)+APHV3)-CEXV**2/A**(4./3) ! V_0
274
275 VDP = VDP0*(1-VDP1*IPAR-VDP2*IPAR**2
276 1 -VDP3/(A**(1./3)+APHVDP1)-VDP4*IPAR/(A**(1./3)+APHVDP2)
277 2 -VDP5*IPAR**2/(A**(1./3)+APHVDP3) ! V_dp
278
279 RM = RM0*A**(1./3)*(1+CEXR**2/A**(4./3))/(1+4*CEXR) ! r_m
280
281 RV = RM+RW*CW+D0-D1*IPAR-D2*IPAR**2-D3/(A**(1./3)+APHD1)
282 1 -D4*IPAR/(A**(1./3)+APHD2)-D5*IPAR**2/(A**(1./3)+APHD3) ! R_v
283
284 AV = AA0*EXP(-AA1*IPAR-AA2*IPAR**2-AA3/(A**(1./3)+APHA1)
285 1 -AA4*IPAR/(A**(1./3)+APHA2)-AA5*IPAR**2/(A**(1./3)+APHA3)) ! a_v
286
287 AKAPV = AKAP0*EXP(-AKAP1*IPAR-AKAP2*IPAR**2
288 1 -AKAP3/(A**(1./3)+APHKAP1)-AKAP4*IPAR/(A**(1./3)+APHKAP2)
289 2 -AKAP5*IPAR**2/(A**(1./3)+APHKAP3)) ! kap_v
290
291 C - - - - -
292 VCEN = V0/(1+EXP((R-RV)/AV))** (AV/AKAPV)
293 1 *(1+VDP/(1+EXP(-(R-RV)/AV))) ! V_cen
294
295 RETURN
296 END FUNCTION
297 C - - - - -

```

```

298 REAL FUNCTION VC(R, A, Z, Q) ! Coulomb-termi
299 REAL R
300 INTEGER A
301 INTEGER Z
302 INTEGER Q
303
304 REAL, PARAMETER :: PI = 3.1415926535 ! Pii
305 REAL, PARAMETER :: HBAR = 197.32697 ! Redusoitu Planckin vakio (MeV*fm)
306 REAL, PARAMETER :: FINE = 0.0072973525693 ! Hienorakennevakio
307 REAL, PARAMETER :: RR = 16 ! Kohdellaan pallona tämän säteen jälkeen <<<<<<
308
309 IF (Q.NE.1) THEN !! Jos kyseessä ei ole protoni,
310 VC = 0 !! niin sähkötermi on nolla.
311 RETURN
312 ENDIF
313
314 IF (R.GE.1) THEN !! Jos ollaan selvästi ytimen ulkopuolella,
315 VC = HBAR*FINE*(Z - 1)/R !! niin ydin näyttää palloilta/pisteeltä.
316 RETURN
317 ENDIF
318
319 VC = HBAR*FINE*((Z - 1)/RR + 4*PI*GRAL2(R, A, Z, RR))
320
321 RETURN
322 END FUNCTION
323 C - - - - -
324 REAL FUNCTION GRAL2(R, A, Z, RR) ! Ulompi integraali
325 REAL R
326 INTEGER A
327 INTEGER Z
328 REAL RR
329
330 INTEGER, PARAMETER :: I2N = 65 ! Jakopisteiden lukumäärä <<<<<<
331 REAL :: STEP2 ! Jakoaskel
332 REAL, DIMENSION(2, I2N) :: F2MAT ! Integrandi 2 matriisimuodossa
333 REAL :: I2S ! Integroinnin tulos
334
335 STEP2 = (RR - R)/(I2N - 1)
336
337 DO I = 2, I2N
338 F2MAT(1, I) = R + (I - 1)*STEP2 ! ]R, RR]
339 F2MAT(2, I) = GRAL1(F2MAT(1, I), A, Z, RR)/F2MAT(1, I)**2
340 END DO
341 F2MAT(1, 1) = R !! Lineaarinen extrapolaaatio,
342 F2MAT(2, 1) = 2*F2MAT(2, 2) - F2MAT(2, 3) !! koska R voi olla nolla
343 CALL INTGRL(I2N, F2MAT, I2S)
344 GRAL2 = I2S
345
346 RETURN
347 END FUNCTION
348 C - - - - -
349 REAL FUNCTION GRAL1(R, A, Z, RR) ! Sisempi integraali
350 REAL R
351 INTEGER A
352 INTEGER Z
353 REAL RR
354
355 INTEGER, PARAMETER :: I1N = 65 ! Jakopisteiden lukumäärä <<<<<<
356 REAL :: STEP1 ! Jakoaskel
357 REAL, DIMENSION(2, I1N) :: F1MAT ! Integrandi 1 matriisimuodossa
358 REAL :: I1S ! Integroinnin tulos
359
360 STEP1 = R/(I1N - 1)
361 DO I = 1, I1N ! Luodaan taulukko integrointia varten...
362 F1MAT(1, I) = (I - 1)*STEP1 ! [0, R]
363 F1MAT(2, I) = F1MAT(1, I)**2*RH(F1MAT(1, I), A, Z, RR)
364 END DO
365 CALL INTGRL(I1N, F1MAT, I1S) ! Integroidaan...
366 GRAL1 = I1S
367
368 RETURN
369 END FUNCTION
370 C - - - - -
371 REAL FUNCTION RH(R, A, Z, RR) ! Varausjakauma (jaettuna alkeisvarauksella)
372 REAL R
373 INTEGER A
374 INTEGER Z
375 REAL RR
376
377 REAL, PARAMETER :: PI = 3.1415926535 ! Pii
378 REAL, PARAMETER :: AAC = 0.55 !fm
379 REAL, PARAMETER :: RC0 = 1.07 !fm
380 REAL, PARAMETER :: CEXC = 0.007
381 REAL, PARAMETER :: CC = 0.71 !MeV
382 REAL, PARAMETER :: C1 = 24 !MeV
383 REAL, PARAMETER :: CIS = 0.5
384
385 INTEGER, PARAMETER :: RHN = 65 ! Jakopisteiden lukumäärä <<<<<<
386 REAL :: RHSTEP ! Jakoaskel
387 REAL, DIMENSION(2, RHN) :: RHFMAT ! Integrandi matriisimuodossa
388 REAL :: RHINT ! Integroinnin tulos
389
390 REAL :: AB
391 REAL :: AC
392 REAL :: RC
393 AB = (2*C1/CC*(1.*(A - Z)/Z - 1))**(3./2) ! A_beta
394 AC = (1 - CIS)*A + CIS*AB ! A_C
395 RC = RC0*(1 + CEXC*Z**2/A**(4./3)) ! R_C
396 1 / (1 + 4*CEXC)*AC**(1./3)

```

```

397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495

RHSTEP = RR/(RHN - 1)
DO I = 1,RHN ! Luodaan taulukko integrointia varten...
  RHFMAT(1, I) = (I - 1)*RHSTEP ! [0, RR]
  RHFMAT(2, I) = RHFMAT(1, I)**2
1 / (1 + EXP((RHFMAT(1, I) - RC)/AAC))
END DO
CALL INTGRL(RHN, RHFMAT, RHINT) ! Integroidaan...
RHO = (Z - 1)/(4*PI*RHINT)

RH = RHO/(1 + EXP((R - RC)/AAC))

RETURN
END FUNCTION
C -----
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495

REAL FUNCTION VLS(R, A, Z, Q, PARS) ! Spin--orbit
REAL R
INTEGER A
INTEGER Z
INTEGER Q
INTEGER PARS
C -----
REAL, PARAMETER :: RMO = 1.08 ! (<--- fm) r_m0
REAL, PARAMETER :: CEXR = 0.005 ! C_exR
REAL, PARAMETER :: CEXV = 0.006 ! C_exV
REAL, PARAMETER :: RW = 1. ! (<--- fm) r_w
C -----
REAL :: IPAR
REAL :: CW
C -----
REAL :: VVLS0
REAL :: VVLS1
REAL :: VVLS2
REAL :: VVLS3
REAL :: VVLS4
REAL :: VVLS5
REAL :: APHVLS1
REAL :: APHVLS2
REAL :: APHVLS3
443
REAL :: DLS0
REAL :: DLS1
REAL :: DLS2
REAL :: DLS3
REAL :: DLS4
REAL :: DLS5
REAL :: APHDLS1
REAL :: APHDLS2
REAL :: APHDLS3
453
REAL :: ALS0
REAL :: ALS1
REAL :: ALS2
REAL :: ALS3
REAL :: ALS4
REAL :: ALS5
REAL :: APHALS1
REAL :: APHALS2
REAL :: APHALS3
463
REAL :: VLS0
REAL :: RM
REAL :: RLS
REAL :: ALS
468
REAL :: MN ! Neutronin massa
REAL :: MP ! Protonin massa
REAL :: M ! Nukleonin massa
472
REAL :: DR
REAL :: DDR
475
INTEGER :: N ! Neutroniluku
N = A - Z
477
IF (Q.EQ.0) THEN ! Neutronille parametrin I ja c_w
IPAR = 1./A*(N-2) ! Parametri I neutronille
IF (N.GT.2) THEN ! c_w neutronille
CW = -1./A**(2./3)
ELSE
CW = +1./A**(2./3)
ENDIF
ENDIF
487
IF (Q.EQ.1) THEN ! Protonille parametrin I ja c_w
IPAR = 1./A*(Z-N) ! Parametri I protonille
IF (Z.GT.N) THEN ! c_w protonille
CW = -1./A**(2./3)
ELSE
CW = +1./A**(2./3)
ENDIF
ENDIF
495

```

```

496 C - - - - -
497 IF (PARS.EQ.1) THEN ! OA
498   VVLS0 = 0.872 ! v_ls0
499   VVLS1 = 0.0612 ! v_ls1
500   VVLS2 = -0.5 ! v_ls2
501   VVLS3 = -2.126 ! v_ls3
502   VVLS4 = -2.5 ! v_ls4
503   VVLS5 = 3. ! v_ls5
504   APHVL1 = -0.5 ! alp_vls1
505   APHVL2 = -0.5 ! alp_vls2
506   APHVL3 = 0. ! alp_vls3
507
508   DLS0 = 0.5971 ! (<--- fm) d_ls0
509   DLS1 = -1. ! d_ls1
510   DLS2 = 0. ! d_ls2
511   DLS3 = 1.733 ! d_ls3
512   DLS4 = 0. ! d_ls4
513   DLS5 = 0. ! d_ls5
514   APHDLS1 = 1. ! alp_dls1
515   APHDLS2 = 0. ! alp_dls2
516   APHDLS3 = 0. ! alp_dls3
517
518   ALS0 = 0.5958 ! (<--- fm) a_ls0
519   ALS1 = -0.3688 ! a_ls1
520   ALS2 = 0. ! a_ls2
521   ALS3 = -0.6385 ! a_ls3
522   ALS4 = 1. ! a_ls4
523   ALS5 = 0. ! a_ls5
524   APHALS1 = 1. ! alp_als1
525   APHALS2 = 0. ! alp_als2
526   APHALS3 = 0. ! alp_als3
527 ELSEIF (PARS.EQ.2) THEN ! OB
528   VVLS0 = 0.872 ! v_ls0
529   VVLS1 = 0. ! v_ls1
530   VVLS2 = -0.5 ! v_ls2
531   VVLS3 = -2.126 ! v_ls3
532   VVLS4 = -2.2 ! v_ls4
533   VVLS5 = 3. ! v_ls5
534   APHVL1 = -0.5 ! alp_vls1
535   APHVL2 = -0.5 ! alp_vls2
536   APHVL3 = 0. ! alp_vls3
537
538   DLS0 = 0.6 ! (<--- fm) d_ls0
539   DLS1 = -1. ! d_ls1
540   DLS2 = 0. ! d_ls2
541   DLS3 = 1.75 ! d_ls3
542   DLS4 = 0. ! d_ls4
543   DLS5 = 0. ! d_ls5
544   APHDLS1 = 1. ! alp_dls1
545   APHDLS2 = 0. ! alp_dls2
546   APHDLS3 = 0. ! alp_dls3
547
548   ALS0 = 0.595 ! (<--- fm) a_ls0
549   ALS1 = -0.25 ! a_ls1
550   ALS2 = 0. ! a_ls2
551   ALS3 = -0.65 ! a_ls3
552   ALS4 = 0.3 ! a_ls4
553   ALS5 = 0. ! a_ls5
554   APHALS1 = 1. ! alp_als1
555   APHALS2 = 0. ! alp_als2
556   APHALS3 = 0. ! alp_als3
557 ELSEIF (PARS.EQ.3) THEN ! HD
558   VVLS0 = 0.725 ! v_ls0
559   VVLS1 = -0.3827 ! v_ls1
560   VVLS2 = -0.5 ! v_ls2
561   VVLS3 = -4.0595 ! v_ls3
562   VVLS4 = -1.3961 ! v_ls4
563   VVLS5 = 3. ! v_ls5
564   APHVL1 = -0.25 ! alp_vls1
565   APHVL2 = -0.25 ! alp_vls2
566   APHVL3 = 0. ! alp_vls3
567
568   DLS0 = 0.5519 ! (<--- fm) d_ls0
569   DLS1 = -1.155 ! d_ls1
570   DLS2 = 0.2 ! d_ls2
571   DLS3 = 1.594 ! d_ls3
572   DLS4 = -0.25 ! d_ls4
573   DLS5 = 1. ! d_ls5
574   APHDLS1 = 1. ! alp_dls1
575   APHDLS2 = 0. ! alp_dls2
576   APHDLS3 = 0. ! alp_dls3
577
578   ALS0 = 0.61 ! (<--- fm) a_ls0
579   ALS1 = -0.2 ! a_ls1
580   ALS2 = 0.2 ! a_ls2
581   ALS3 = -0.57 ! a_ls3
582   ALS4 = 0.3 ! a_ls4
583   ALS5 = -1 ! a_ls5
584   APHALS1 = 1. ! alp_als1
585   APHALS2 = 0. ! alp_als2
586   APHALS3 = 0. ! alp_als3
587 ENDIF
588 C - - - - -
589 VLS0 = VVLS0*(1-VVLS1*IPAR-VVLS2*IPAR**2
590 1 -VVLS3/(A**(1./3)+APHVL1)-VVLS4*IPAR/(A**(1./3)+APHVL2)
591 2 -VVLS5*IPAR**2/(A**(1./3)+APHVL3)) ! V_ls0
592
593 RM = RM0*A**(1./3)*(1+CEXR**2/A**(4./3))/(1+4*CEXR) ! r_m
594

```

```

595     RLS = RM+DLS0-DLS1*IPAR-DLS2*IPAR**2
596     1 -DLS3/(A**(1./3)+APHDLS1)-DLS4*IPAR/(A**(1./3)+APHDLS2)
597     2 -DLS5*IPAR**2/(A**(1./3)+APHDLS3) ! R_ls
598
599     ALS = ALS0*EXP(-ALS1*IPAR-ALS2*IPAR**2)
600     1 -ALS3/(A**(1./3)+APHALS1)
601     2 -ALS4*IPAR/(A**(1./3)+APHALS2)
602     3 -ALS5*IPAR**2/(A**(1./3)+APHALS3) ! a_ls
603 C -----
604     MN = 939.5654133 ! (<--- MeV)
605     MP = 938.2720813 ! (<--- MeV)
606     M = (MN*N + MP*Z)/A
607 C -----
608     DR = 1/(1+EXP((R-RLS)/ALS))
609     DDR = -DR**2*EXP((R-RLS)/ALS)/ALS
610 C -----
611     VLS = VLS0/(2*M*R)*DDR
612 C -----
613     RETURN
614     END FUNCTION
615 C -----

```

simpson.for

```
1      SUBROUTINE INTGRL(D, FMAT, S)
2      INTEGER D
3      DIMENSION FMAT(2, D)
4      REAL S
5      REAL :: A
6      REAL :: B
7      REAL :: C
8      S = 0
9      DO I = 1, D-2, 2 ! Simpson
10     CALL QUADR(FMAT(1, I+0), FMAT(2, I+0),
11              1 FMAT(1, I+1), FMAT(2, I+1),
12              2 FMAT(1, I+2), FMAT(2, I+2),
13              3 A, B, C)
14     S = S + A/3*(FMAT(1, I+2)**3 - FMAT(1, I)**3)
15     + B/2*(FMAT(1, I+2)**2 - FMAT(1, I)**2)
16     + C/1*(FMAT(1, I+2)**1 - FMAT(1, I)**1)
17     END DO
18     IF (MOD(D, 2) .EQ. 0) THEN ! Integrointivälän loppuosa
19     CALL QUADR(FMAT(1, D-2), FMAT(2, D-2),
20              1 FMAT(1, D-1), FMAT(2, D-1),
21              2 FMAT(1, D-0), FMAT(2, D-0),
22              3 A, B, C)
23     S = S + A/3*(FMAT(1, D)**3 - FMAT(1, D-1)**3)
24     + B/2*(FMAT(1, D)**2 - FMAT(1, D-1)**2)
25     + C/1*(FMAT(1, D)**1 - FMAT(1, D-1)**1)
26     ENDIF
27     RETURN
28     END SUBROUTINE
29
30     SUBROUTINE QUADR(P11, P12, ! Quadraattinen interpolaatio
31                    P21, P22,
32                    P31, P32,
33                    A, B, C)
34     REAL :: P11 ! x_1
35     REAL :: P12 ! y_1
36     REAL :: P21 ! x_2
37     REAL :: P22 ! y_2
38     REAL :: P31 ! x_3
39     REAL :: P32 ! y_3
40     REAL :: A ! Neljöttermin kerroin
41     REAL :: B ! Lineaaritermin kerroin
42     REAL :: C ! Vakiotermi
43     REAL :: DQ1 ! Pisteiden 1 ja 2 erotusosamäärä
44     REAL :: DQ2 ! Pisteiden 2 ja 3 erotusosamäärä
45     DQ21 = (P22 - P12)
46     / (P21 - P11)
47     DQ32 = (P32 - P22)
48     / (P31 - P21)
49     A = (DQ32 - DQ21)
50     / (P31 - P11)
51     B = DQ21 - A*(P11 + P21)
52     C = P12 - A*P11**2 - B*P11
53     RETURN
54     END SUBROUTINE
```


(wspot.for)

```
1      REAL FUNCTION V(R, A, Z, J2, L, Q)
2      REAL R
3      INTEGER A
4      INTEGER Z
5      INTEGER J2
6      INTEGER L
7      INTEGER Q
8      REAL, PARAMETER :: HBAR = 197.32697 ! Redusoitu Planckin vakio (MeV*fm)
9      V = VCEN(R, A, Z, Q)
10     + VC(R, A, Z, Q)
11     + VLS(R, A, Z, Q)*HBAR**2/2
12     *(J2/2.*(J2/2.+1)-L*(L+1)-0.75)
13
14     RETURN
15     END FUNCTION
16 C -----
17     REAL FUNCTION VCEN(R, A, Z, Q) ! Keskeistermi
18     REAL R
19     INTEGER A
20     INTEGER Z
21     INTEGER Q
22     REAL, PARAMETER :: R0 = 1.27 ! Sädeparametri (fm)
23     REAL, PARAMETER :: SD = 0.67 ! Pinnan diffuusio (fm)
24     INTEGER :: N ! Neutroniluku
25     REAL :: RAD ! Ytimen säde
26     REAL :: V0 ! Potentiaalinen syvyys (MeV)
27 C -----
28     N = A - Z
29     RAD = R0*A**(1./3)
30     V0 = 51 + (2*Q-1)*33*(N-Z)*(1./A)
31 C -----
32     VCEN = -V0/(1+EXP((R-RAD)/SD))
33     RETURN
34     END FUNCTION
35 C -----
36     REAL FUNCTION VC(R, A, Z, Q) ! Coulomb-termi
37     REAL R
38     INTEGER A
39     INTEGER Z
40     INTEGER Q
41     REAL, PARAMETER :: HBAR = 197.32697 ! Redusoitu Planckin vakio (MeV*fm)
42     REAL, PARAMETER :: FINE = 0.0072973525693 ! Hienorakennevakio
43     REAL, PARAMETER :: R0 = 1.27 ! Sädeparametri (fm)
44     REAL :: RAD ! Ytimen säde
45     REAL :: RR
46 C -----
47     RAD = R0*A**(1./3)
48     IF (R.GT.RAD) THEN
49         RR = 1/R
50     ELSE
51         RR = (3-(R/RAD)**2)/(2*RAD)
52     ENDIF
53 C -----
54     VC = HBAR*FINE*Z*RR
55     VC = Q*VC ! Jos kyseessä on neutroni, niin VC = 0.
56     RETURN
57     END FUNCTION
58 C -----
59     REAL FUNCTION VLS(R, A, Z, Q) ! Spin--orbit
60     REAL R
61     INTEGER A
62     INTEGER Z
63     INTEGER Q
64     REAL, PARAMETER :: HBAR = 197.32697 ! Redusoitu Planckin vakio (MeV*fm)
65     REAL, PARAMETER :: R0 = 1.27 ! Sädeparametri (fm)
66     REAL, PARAMETER :: SD = 0.67 ! Pinnan diffuusio (fm)
67     INTEGER :: N ! Neutroniluku
68     REAL :: RAD ! Ytimen säde
69     REAL :: V0 ! Potentiaalinen syvyys (MeV)
70     REAL :: DVWS ! dVWS/dr
71 C -----
72     N = A - Z
73     RAD = R0*A**(1./3)
74     V0 = 51 + (2*Q-1)*33*(N-Z)*(1./A)
75     VCEN = -V0/(1+EXP((R-RAD)/SD))
76     DVWS = 1/(SD*V0)*EXP((R-RAD)/SD)*VCEN**2
77 C -----
78     VLS = -0.44*(R0/HBAR)**2/R*DVWS
79     RETURN
80     END FUNCTION
```

(trapez.for)

```
1  SUBROUTINE INTGRL(D, FMAT, S)
2  INTEGER D
3  DIMENSION FMAT(2, D)
4  REAL S
5
6  REAL :: LEFT ! Vasen summa
7  REAL :: RIGHT ! Oikea summa
8
9  LEFT = 0
10 RIGHT = 0
11 DO I = 1, D-1
12 LEFT = LEFT + (FMAT(1, I+1) - FMAT(1, I))*FMAT(2, I)
13 RIGHT = RIGHT + (FMAT(1, I+1) - FMAT(1, I))*FMAT(2, I+1)
14 END DO
15
16 S = (LEFT + RIGHT)/2
17
18 RETURN
19 END SUBROUTINE
```