

UNIVERSITATEA “BABEȘ-BOLYAI” CLUJ-NAPOCA

FACULTATEA DE FIZICĂ

SPECIALIZAREA FIZICĂ INFORMATICĂ

LUCRARE DE LICENȚĂ

Coordonator științific

Lector dr. Horia Pașca

ABSOLVENT

DENIS-MARIAN HUMĂ

2023

UNIVERSITATEA “BABEȘ-BOLYAI” CLUJ-NAPOCA

FACULTATEA DE FIZICĂ

SPECIALIZAREA FIZICĂ INFORMATICĂ

LUCRARE DE LICENȚĂ

MULTIPLICITATEA NEUTRONILOR ÎN CAZUL FISIUNII INDUSE

Coordonator științific

Lector dr. Horia Pașca

ABSOLVENT

DENIS-MARIAN HUMĂ

2023

Abstract

In this work, a configuration probability of emitting a certain number of neutrons and the mean number of neutrons emitted by elements with atomic numbers between $Z = 28$ and $Z = 66$. There are 4 main chapters in this work as they are to be described. The first chapter is called *Nuclear Model* and contains general information about nuclear fission, a description of the model and detailed steps of the study and the way the work is being prepared.

For the second chapter, it is called *Computational Model and Implementation*, where it's written about the way the physics model into the computational world is implemented and the steps followed in order to create the software for this work, using Jackson's Model. In this work, technologies like HTML, CSS, JavaScript, React.Js and Python are being used

The third chapter is where the results are presented and analysis is being discussed, referring to both the results related to physics and the results related to software and implementation. This is where results regarding probability of emitting a certain number of neutrons and the mean number of neutrons emitted by elements with atomic numbers between $Z = 28$ and $Z = 66$ are being analyzed, showing a good fit to the model.

The fourth chapter is the last one, where there is a short summary presented and the results are being mentioned, followed by further notes and applications.

CUPRINS

	Introducere	4
1.	Modelul Nuclear	6
	1.1 Descrierea procesului de fisiune.....	6
	1.2 Modelul statistic al fisiunii nucleare.....	6
	1.3 Sistemul dinuclear	7
	1.4 Parametrizarea sistemului.....	9
	1.5 Potențialul de interacțiune al sistemului.....	10
	1.6 Energia potențială a sistemului.....	12
	1.7 Minimele energiei potențiale și valori optime ale deformărilor	14
	1.8 Determinarea energiei de excitare	16
2.	Modelul Computațional și Implementare	18
	2.1 Scopul programului	18
	2.2 Determinarea probabilității de a emite x neutroni	18
	2.3 Arhitectura programului	19
	2.4 Structura părții de back-end.....	20
	2.5 Structura bazei de date.....	21
	2.6 Structura API.....	23
	2.7 Structura interfeței grafice	24
3.	Rezultate	29
	3.1 Probabilitatea de a emite neutroni	29
	3.2 Numărul mediu de neutroni emiși per configurație.....	39
	3.3 Prezentarea aplicației și a funcționalităților sale	43
4.	Concluzii	47
	4.1 Descrierea programului realizat	47
	4.2 Rezultate obținute.....	47
	4.3 Concluzii finale	48
	Bibliografie	49

Introducere

În această lucrare se urmărește studiul fisiunii nucleare în modelul statistic dinuclear, alături de calculul potențialului de interacțiune dintre cele două fragmente rezultate, a parametrilor de deformare și a probabilității unei anumite configurații de a emite un număr de neutroni ca urmare a fisiunii nucleare.

Modelul nuclear este realizat folosind sistemul dinuclear, care reprezintă o configurație temporară a două fragmente ce interacționează prin forțe nucleare și electrostatice. Acest sistem este definit de numerele atomice și numerele de masă ale fragmentelor, alături de distanța dintre acestea. Totuși, datorită interacțiunii dintre nucleele rezultate, unde forțele nucleare sunt puternice și de scurtă distanță fac ca polii apropiați să se atragă, iar forțele electrostatice ce sunt de distanță lungă, fac ca polii opuși să se respingă, astfel, cele două fragmente se deformează și mai puternic, luând forma unor elipsoide, deci, sistemul dinuclear este descris adițional și de parametrii de deformare ale celor două fragmente.

Pentru a determina valorile potrivite ale parametrilor de deformare este necesară cunoașterea expresiei energiei potențiale a sistemului, ce este descrisă ca sumă a energiilor fiecărui fragment component, alături de potențialul de interacție dintre acestea. Totuși, pentru a afla energia potențială este nevoie de a determina expresia potențialului de interacție dintre cele două nuclee, acesta fiind o sumă dintre potențialul Coulomb și potențialul de interacțiune nucleară.

Odată aflat potențialul de interacție dintre fragmente, se determină energiile celor două nuclee ca sumă dintre energia Coulomb, energia potențială de suprafață, energia de simetrie și corecția de pături a fiecărui fragment. Cu toate aceste expresii determinate, se poate afla energia potențială totală a sistemului.

Această lucrare urmărește păstrarea unei energii potențiale minime, astfel, observând expresiile componentelor sale, se poate observa că acestea depind de masele atomice și de parametrii de deformare a nucleelor. Pentru a obține valorile potrivite ale maselor atomice se folosește minimizarea, iar pentru aflarea parametrilor de deformare optimi se realizează graficul energiei potențiale în funcție de parametrii de deformare a nucleelor, alegându-se parametrii de deformare corespunzători minimelor de pe grafic.

Energia potențială fiind determinată astfel, se poate afla energia de excitare a sistemului ca diferență dintre energia potențială a nucleului înaintea fisiunii minus energia potențială a sistemului, determinată anterior. Energia potențială înaintea fisiunii este aflată folosind formula semiempirică a maselor.

Folosind modelul lui Jackson se calculează probabilitățile unei configurații de a emite un anumit număr de neutroni în urma fisiunii, aceasta fiind o funcție ce depinde de energia de excitație a sistemului și energiile de legătură a celor două fragmente implicate.

CAPITOLUL 1 – Modelul Nuclear

1.1 Descrierea procesului de fisiune

Fisiunea nucleară reprezintă un proces fizic prin care un nucleu se divide în două sau mai multe fragmente mai mici, ce au suma maselor atomice și suma numerelor atomice egale cu masa atomică, respectiv numărul atomic al nucleului inițial.

În funcție de cantitatea de energie a nucleului inițial, fisiunea este de două tipuri, spontană și indusă. Fisiunea spontană apare pentru nucleele grele și instabile, aceștia având energia internă necesară pentru declanșarea fisiunii. Pentru a induce acest fenomen, nucleul are nevoie să primească energie din exterior, lucru ce se realizează prin lovirea sa cu particule ușoare, care vor crește energia nucleului, dar nu îi vor oferi un moment cinetic considerabil sau particule grele care îi vor oferi un moment cinetic, alături de creșterea energiei.

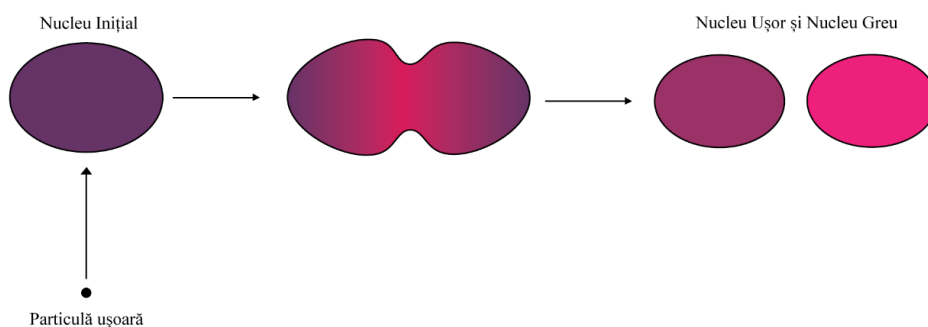


FIG 1.1 – Schemă ce reprezintă procesul de fisiune nucleară indusă de o particulă ușoară, având ca rezultat două fragmente noi (un nucleu ușor și un nucleu greu)

În această lucrare este studiat scenariul în care fisiunea este indusă prin ciocnirea cu o particulă ușoară, așa cum este prezentat în figura 2.1, astfel, nucleul inițial va primi doar energie de excitație, momentul cinetic fiind nul.

Procesul începe prin lovirea nucleului cu particula ușoară, lucru ce face ca energia internă să crească, iar procesul de fisiune nucleară să debuteze. Următoarea etapă a fenomenului este alungirea nucleului și îngustarea acestuia în zona de mijloc, dezvoltându-se o punte între viitoarele fragmente. Fisiunea se încheie cu formarea a două noi nuclee, unul greu și unul ușor.

1.2 Modelul statistic al fisiunii nucleare

Modelul pe care se construiește această lucrare este modelul statistic al fisiunii, ce se bazează pe principiile fizicii statistice și care oferă posibilitatea de a afla probabilitatea unui nucleu de a fisiona. Pentru acest model, se consideră existența unui echilibru statistic la momentul scindării, menținut de efecte non-adiabatică puternice.

În urma procesului de fisiune, două fragmente de fisiune vor fi produse, ele putând fi însoțite de emisii de neutroni și radiații gamma. Folosirea modelului statistic al fisiunii nucleare duce la distribuții statistice pentru masele și energiile fragmentelor rezultate, cât și pentru numărul de neutroni emiși.

Nucleul se poate afla în diferite configurații date de cantitatea de energie de excitare primită, astfel densitatea de stări cuantice va crește rapid odată cu creșterea energiei de excitare. O bună aproximare a temperaturii nucleare este dată de expresia [1]

$$T = \sqrt{\frac{E^*}{a}}$$

(1.1)

unde parametrul a este parametrul densității de nivele. De asemenea, o relație ce descrie foarte bine densitatea de nivel experimentală la energii joase este [1]

$$\rho(E^*) \sim e^{2\sqrt{aE^*}}$$

1.2)

considerând T constant.

1.3 Sistemul dinuclear

Sistemul dinuclear [2] reprezintă o configurație temporară, în care doua nuclee interacționează prin forțe nucleare și electrostatice, ce este folosit în această lucrare pentru descrierea configurației de scindare a unor fragmente de fisiune foarte bine definite ce se află în contact unul cu celălalt.

Acest model a fost folosit inițial pentru studiul coliziunilor profund inelastice ale nucleelor [2], însă a devenit mai târziu un instrument important în studiul fuziunii nucleare, unde a fost utilizat pentru a descrie secțiunea eficace de fuziune a nucleelor grele și foarte grele. Totuși, acest sistem poate fi folosit și în cazul fisiunii nucleare [3] [4], așa cum este cazul acestei lucrări, unde configurația nucleului ce fisionează este considerată a fiind formată din două fragmente separate ce interacționează între ele și care sunt în contact unul cu celălalt, urmărindu-se ultimul proces al fisiunii nucleare [5] [6].

În timpul acestui proces, nucleul ce fisionează va forma un sistem dinuclear în momentul în care cele doua fragmente componente sunt aproape separate (figura 2.1), astfel sistemul va rămâne stabil pentru o perioadă înainte de fisiunea completă. În interiorul DNS se realizează un

transfer de nucleoni între fragmente pentru a duce la o simetrie de masă și sarcină. De asemenea, distanța dintre fragmente se poate schimba, astfel, creșterea acesteia duce la destrămarea sistemului dinuclear.

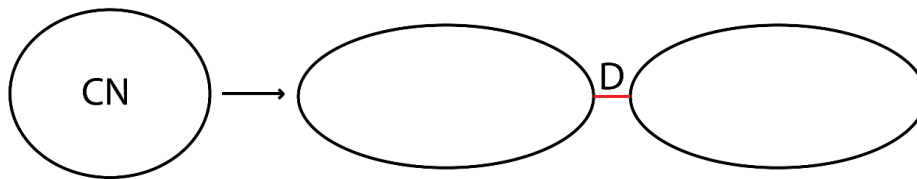


FIG 1.2 – Model al configurației DNS, unde CN este nucleul inițial din care rezultă cele două fragmente, iar D este distanța dintre ele.

Fiecare sistem dinuclear are câteva caracteristici specifice, astfel, ele sunt formate din fragmente ce interacționează puternic între ele, însă își păstrează individualitatea. De asemenea, DNS este o configurație instabilă ce poate fi schimbată prin transfer de nucleoni de la un fragment la celălalt, proces mediat de energia potențială a sistemului [7].

Asupra celor două fragmente vor acționa forțe nucleare și electromagnetice, astfel, forțele nucleare, fiind de rază scurtă și mult mai puternice decât cele electromagnetice, vor face ca suprafețele din zonele [7] apropiate de contact ale fragmentelor să se atragă, iar forțele electromagnetice, ce acționează la o distanță mare, vor face ca suprafețele aflate în extremele exterioare să se respingă, astfel, precum se poate observa atât în figura 2.1 cât și în figura 2.2, fragmentele vor deveni elipse alungite.

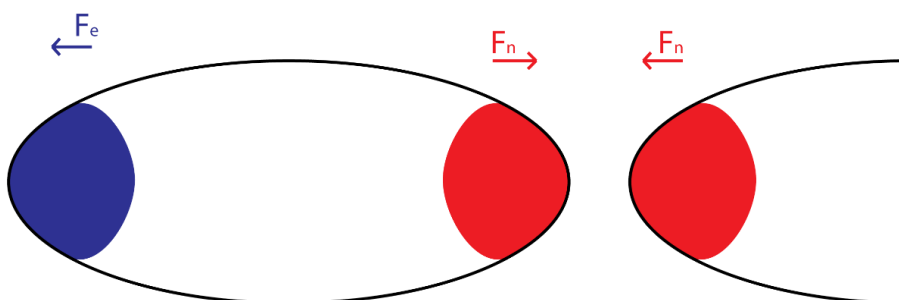


FIG 1.3 – Reprezentare grafică a forțelor ce acționează asupra fragmentelor, unde F_n sunt forțele nucleare, F_e sunt forțele electrostatice, iar culoarea roșie reprezintă atracția și culoarea albastră respingerea.

Rezultatele oferite de către sistemul dinuclear (DNS) asupra caracteristicilor fisiunii nucleare oferă posibilitatea de a analiza emisiile de neutroni în urma acestui proces, lucru ce se urmărește în această lucrare. Un lucru important de menționat, folosirea DNS oferă posibilitatea

de a alege separat masa atomică și numărul atomic pentru fiecare fragment, în mod independent, lucru ce duce la posibilitatea de a studia fenomenul folosind multe configurații de deformare [2].

1.4 Parametrizarea sistemului

Pe baza sistemului descris anterior (DNS) [7], configurația este modelată de două elipsoide deformate, aflate la distanță foarte mică unul de celălalt, unde axa de simetrie a acestora coincide cu dreapta ce unește centrul acestora. Fiecare elipsoid este descris de un număr de masă A , un număr atomic Z și un parametru de deformare β , acestea fiind parametrii geometrici ai sistemului împreună cu distanța d dintre fragmente. În starea aceasta, datorită originii celor două fragmente, numărul de masă A și numărul atomic Z ale nucleului inițial vor fi egale cu suma numerelor de masă, respectiv a numerelor atomice ale fragmentelor [8].

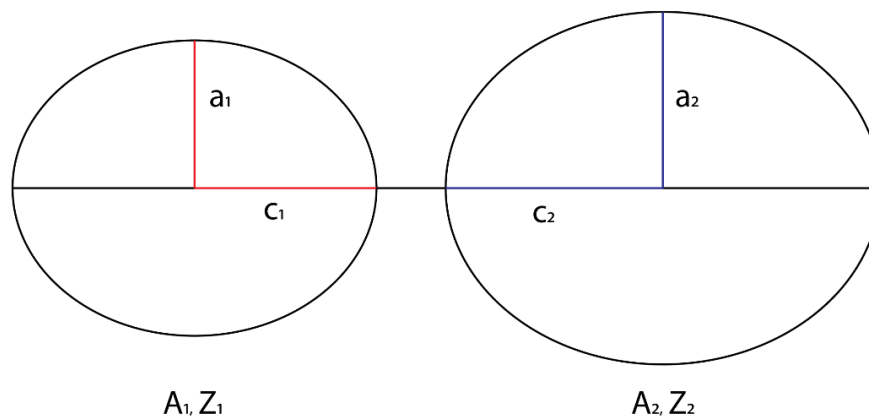


FIG 1.4 – Reprezentare a configurației sistemului privind geometria fragmentelor

Având în vedere deformarea celor două elipsoide, parametrul de deformare β a fost ales după relația $\beta = c/a$, unde c este axa majoră, iar a axa minoră, după cum arată figura 2.3. Lungimile celor două semiaxe vor fi calculate după formulele următoare, luând în calcul conservarea volumului [8]

$$c_i = r_{0i} A_i^{\frac{1}{3}} \beta_i^{\frac{2}{3}} \quad (1.5)$$

$$a_i = r_{0i} A_i^{\frac{1}{3}} \beta_i^{-\frac{1}{3}} \quad (1.6)$$

unde valoarea lui r_0 va fi constantă și egală cu 1.15 fm pentru orice nucleu mai greu decât cel de beriliu, totuși, pentru nuclei ușori, aceasta va varia între 1.03 și 1.12, iar i va lua valori între 1 și 2, reprezentând indicii fragmentelor.

În această parametrizare se alege orientarea pol la pol, adică o configurație coaxială, deoarece schimbarea unghiurilor dintre axa de simetrie și planul sistemului va duce la o creștere a energiei de interacție, astfel, având libertatea de a alege referința acesteia, s-a optat pentru orientarea coaxială ce duce la o energie potențială minimă.

1.5 Potențialul de interacție al sistemului

Pentru a determina expresia energiei de interacție a sistemului, se pornește de la expresia energiei potențiale a unui sistem binar, aceasta fiind formată din energiile fragmentelor, potențialul de interacție dintre cele două componente participante și energia de rotație a sistemului, după cum urmează [9] [10]

$$\begin{aligned} U(\{A_i, Z_i, \beta_i\}, d, L) &= U_1^{MP}(A_1, Z_1, \beta_1) + \delta U_1 \\ &+ U_2^{MP}(A_2, Z_2, \beta_2) + \delta U_2 \\ &+ V_{int}(\{A_i, Z_i, \beta_i\}, d) \end{aligned} \quad (1.7)$$

unde U_i^{MP} este energia fragmentului în modelul nuclear al picăturii, δU_i este corecția de pături, L este momentul cinetic, iar V_{int} este potențialul de interacție.

Potențialul de interacție [10] al acestui sistem este format din două componente, potențialul Coulomb V_C și potențialul nuclear V_N . Pentru a afla expresia potențialului de interacție, se va determina formula pentru cei doi membri ai săi [10].

Expresia potențialului Coulomb va fi definită după cum urmează [7]

$$V_C = 1.44 \frac{q_1 + q_2}{l} [s(e_1) + s(e_2) - 1 + S(e_1, e_2)] \quad (1.8)$$

unde l este distanța dintre centrele celor două fragmente, iar $e_i = \sqrt{\frac{c_i^2 - a_i^2}{l^2}}$, unde c și a sunt cele două semiaxe ale fiecărui elipsoid.

Termenii $s(e_i)$ și $S(e_1, e_2)$ vor fi definiți astfel [7]

$$s(e_i, \theta) = 3 \sum_{n=0}^{\infty} \frac{P_{2n}(\cos\theta)}{(2n+1)(2n+3)} e_i^{2n} \quad (1.9)$$

$$S(e_1, e_2, \theta) = 9 \sum_{j,k=1}^{\infty} \frac{P_{2(j+k)}(\cos\theta)(2j+2k)!}{(2j+1)(2j+3)(2k+1)(2k+3)(2j)!(2k)!} e_1^{2j} e_2^{2k} \quad (1.10)$$

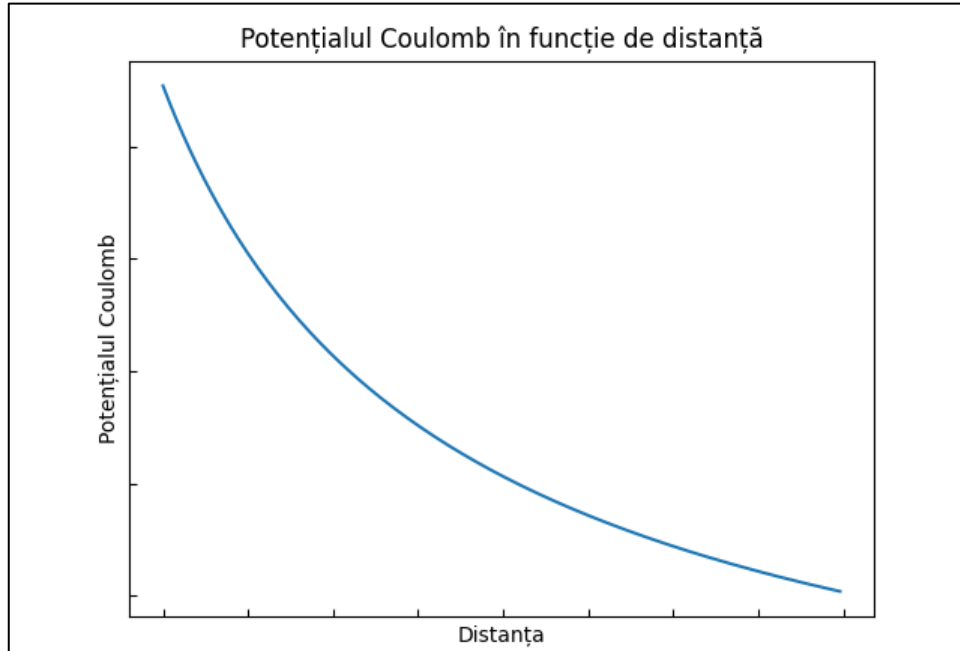


FIG 1.5 – Graficul potențialului Coulomb în funcție de distanță

Totuși, pentru configurația aleasă, cele două fragmente elipsoide sunt coliniare, deci $\vartheta = 0$, ceea ce face ca polinomul prezent în cele două ecuații să fie egal cu 1, deci termenii vor deveni $s(e_i)$ și $S(e_1, e_2)$. [7]

Pentru a determina expresia potențialului nuclear, se va considera că aceasta are expresia de forma următoare [12]

$$V_N(l) = \int \rho_1(r_1)\rho_2(l-r_2)F(r_1-r_2)dr_1dr_2 \quad (1.11)$$

unde ρ_i reprezintă densitatea de nucleoni ale fragmentelor și $F(r_1-r_2)$ interacțiunea efectivă nucleon-nucleon. Această interacțiune dintre nucleoni poartă un rol important și prezintă caracter repulsiv pentru distanțe mici între aceștia. În acest caz, interacțiunea va avea forma uneia de tip Skyrme [12] și expresia va arăta

$$F(r_1-r_2) = C_0 \left[F_{in} \frac{\rho_0(r_1)}{\rho_{00}} + F_{ex} \left(1 - \frac{\rho_0(r_1)}{\rho_{00}} \right) \right] \delta(r_1-r_2) \quad (1.12)$$

iar

$$F_i = (f_i + f'_i \cdot \tau_1 \cdot \tau_2) + (g_i + g'_i \cdot \tau_1 \cdot \tau_2) \sigma_1 \sigma_2 \quad (1.13)$$

unde $\rho_0(r_1)$ este densitatea de materie nucleară în acel punct, ρ_{00} este densitatea de nucleoni în centrul nucleului, σ_i, τ_i reprezintă matricele de spin, respectiv izospin și valorile parametrilor rămași sunt deja cunoscuți din descrierea sistemelor mari de date experimentale din teoria sistemelor Fermi finite [13]: $C_0 = 300 \text{ MeV fm}^3$, $f_{in} = 0.09$, $f_{ex} = -2.59$, $f'_{in} = 0.09$, $f'_{ex} = -2.59$.

Potențialul de interacție va prezenta în partea de început un minim local, în punctul B, așa cum este prezentat în figura 1.6, reprezintă o porțiune de echilibru temporal. În continuare, punctul A de pe grafic reprezintă partea superioară a unei gropi de potențial care, odată trecut, va face curba să continue spre zero.

Atât punctul de minim cât și punctul superior al barierei de potențial joacă un rol important în timpul fisiunii nucleare, fiind ușor de imaginat faptul că cele două fragmente se află în groapa de potențial înaintea fisiunii, ca mai apoi, primind energie de excitare, în cazul în care nu au deja suficientă, vor trece de bariera de potențial și astfel fisiunea va fi declanșată.

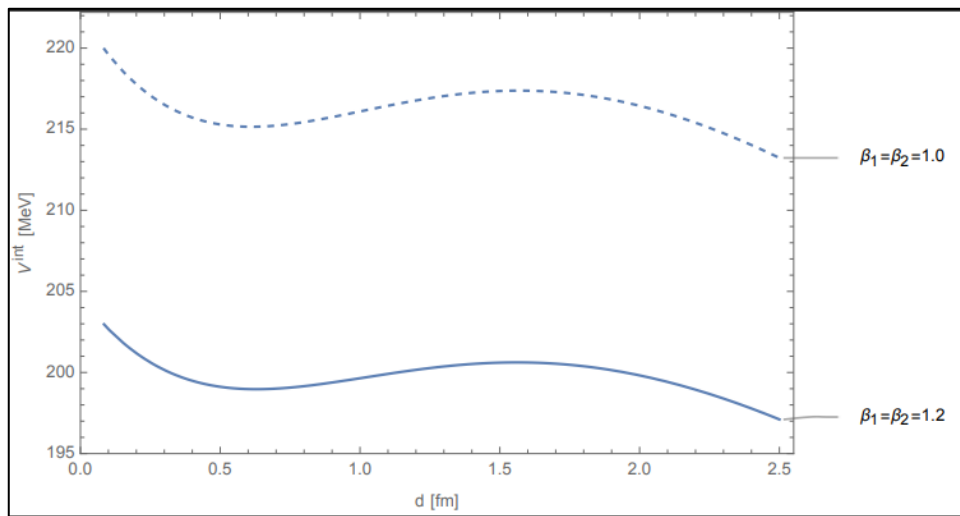


FIG 1.6 – Graficul potențialului de interacție în funcție de distanță pentru fragmente de ^{106}Mo și ^{132}Sn , pentru diferite valori ale parametrilor de deformare

1.6 Energia potențială a sistemului

Anterior a fost prezentată expresia energiei potențiale a sistemului în ecuația (1.7) și se pot observa componentele acesteia. Momentan, singura componentă determinată este potențialul de interacție, iar, așa cum s-a discutat la finalul subcapitolului 1.2, potențialul datorat rotației nu

va fi luat în considerare, încercând obținerea unei energii potențiale minime. În această parte a lucrării se va calcula energia fiecărui fragment în modelul nuclear al picăturii și se va descrie corecția de pături.

Pentru energia fragmentului în modelul nuclear al picăturii, se consideră ca aceasta este formată din trei termeni de energie: energie potențială de suprafață, de simetrie și Coulomb [14]

$$U_i^{MP}(A_i, Z_i, \beta_i) = U_i^C(A_i, Z_i, \beta_i) + U_i^{supr}(A_i, Z_i, \beta_i) + U_i^{sim}(A_i, Z_i) \quad (1.14)$$

Energia Coulomb a fragmentului va fi determinată după modelul energiei Coulomb a unui elipsoid încărcat uniform [14]

$$U_i^C(A_i, Z_i, \beta_i) = \frac{3}{5} \frac{(Z_i e)^2}{R_{0,i}} \frac{\beta_i^{1/3}}{\sqrt{\beta_i^2 - 1}} \ln \left(\beta_i + \sqrt{\beta_i^2 - 1} \right) \quad (1.15)$$

unde $R_{0,i} = 1.2249 A_i^{1/3}$ fm și reprezintă raza fragmentului i atunci când este în formă sferică.

Pentru energia de simetrie, aceasta va avea expresia [14]

$$U_i^{sim}(A_i, Z_i) = 27.612 \frac{(N_i - Z_i)^2}{A_i} \text{ MeV} \quad (1.16)$$

Variația deformării fragmentelor prezintă un rol important în acest model, deoarece oferă posibilitatea de a descoperi energia potențială minimă. În timpul scindării, sistemul rămâne în echilibru statistic, așa cum este prezentat în figura 1.6, unde deformările fragmentelor variază, lucru descris de energia potențială de suprafață. Datorită acestei relații dintre deformare și energia potențială de suprafață, este de dorit ca termenul de energie să fie descris ca funcție dependentă de cele două deformări β_1 și β_2 .

$$U_i^{supr}(A_i, Z_i, \beta_i) = \sigma_{0,i} \left[1 + k_i (\beta_i - \beta_i^{sf})^2 \right] \quad (1.17)$$

unde $\sigma_{0,i} = 0.9517 \left[1 - 1.7826 \left(\frac{N_i - Z_i}{A_i} \right)^2 \right] \text{ MeV}/\text{fm}^2$, β_i^{sf} este deformarea nucleului în stare fundamentală și $k_i = 0.002$.

Modelul nuclear al picăturii este unul macroscopic, ce descrie foarte bine nucleul și proprietățile acestuia, însă pentru energia de legătură, valorile nu respectă un grafic liniar, deși acestea sunt comparative cu valorile experimentale. Pentru a remedia acest lucru, se apelează la modelul nucleului în pături, care este unul microscopic și care, deși nu are valori corecte pentru energia de legătură, respectă forma graficului, astfel, se adaugă corecția de pături pentru a avea cele mai bune valori. [15]

Odată ce expresiile membrilor componenți ai energiei potențiale a sistemului au fost determinate, se poate calcula energia potențială.

1.7 Minimele energiei potențiale și valorile optime ale deformărilor

În subcapitolul anterior a fost determinată expresia energiei potențiale a acestui sistem și se poate observa că aceasta este o funcție ce depinde de masele atomice, de numerele atomice, de parametrii de deformare și de distanța dintre fragmente. Totuși, sistemul aflându-se în acest echilibru statistic, distanța dintre fragmente poate fi considerată constantă. Deoarece numărul atomic a fiecărui fragment va rămâne constant, se poate că energia potențială va fi o funcție ce depinde doar de numerele atomice și de parametrii de deformare ale celor două fragmente.

Modul în care parametrii de deformare și masa atomică influențează forma fragmentelor prin lungimea semiaxelor este definit în ecuațiile (1.5) și (1.6). Pentru a obține energia potențială minimă, în ceea ce privește masele atomice se va recurge la minimizare.

Analizând ecuația (1.15) se poate observa cum U_i^C scade odată cu creșterea parametrilor de deformare, lucru asemănător cu potențialul de interacție V_{int} . Totuși, potențialul de suprafață U_i^{supr} , privind ecuația (1.17) va crește odată cu parametrii de deformare, lucru ce duce la apariția minimelor pentru energia potențială. Astfel, valoarea parametrilor de deformare se va alege în așa fel încât energia potențială calculată să fie minimă.

Pentru a exemplifica modul prin care variația parametrului de deformare ajută la găsirea energiei potențiale minime în starea fundamentală a unuia dintre nuclee, în figura alăturată sunt reprezentate U_i^C și U_i^{supr} în funcție de parametrul de deformare și se poate observa apariția unui minim pe grafic la intersecția celor două curbe

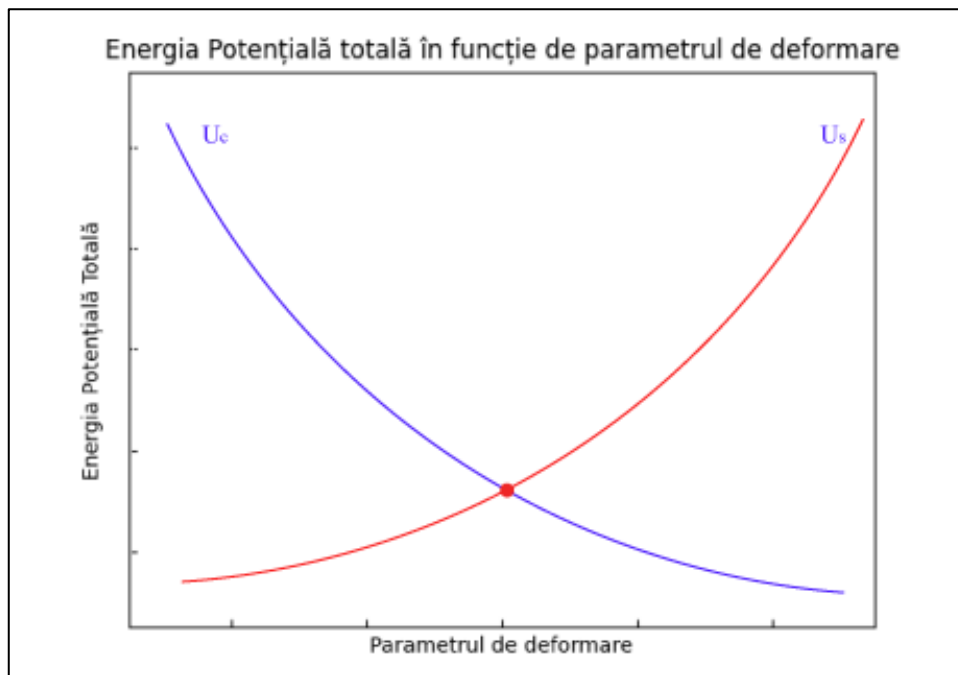


FIG 1.7 – Grafic ce exemplifică existența unui minim al energiei potențiale totale în punctul în care cele două curbe se intersectează

Totuși, sistemul studiat prezintă doi nuclei ce au parametri de deformare diferiți, deci energii potențiale diferite, lucru ce va duce la prezența a mai multor minime locale, așa cum este exemplificat în figura 1.9.

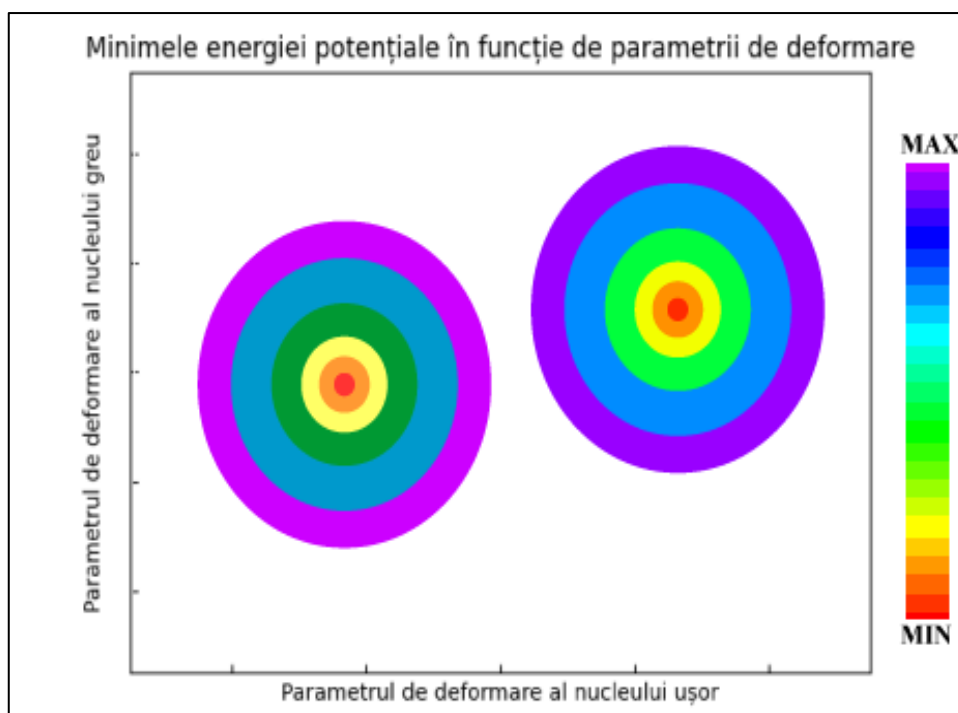


FIG 1.8 – Grafic reprezentativ ce descrie apariția a două minime locale ale energiei potențiale a sistemului, colorate cu roșu pentru valori diferite ale parametrilor de deformare β ale celor două nuclee [16]

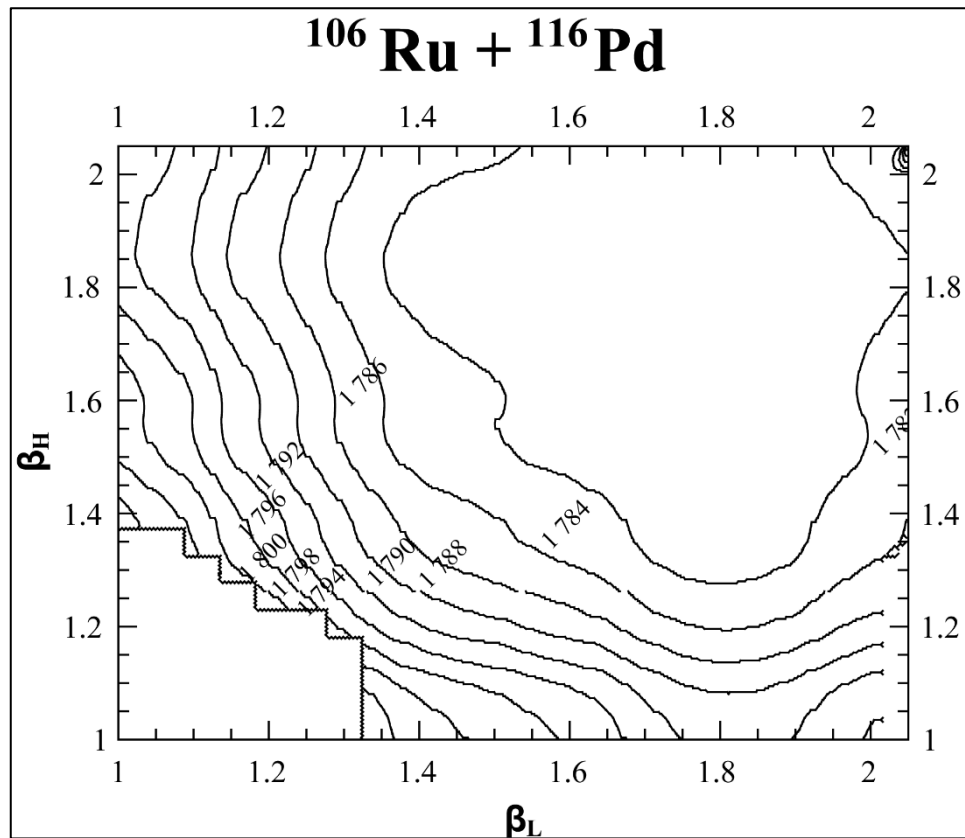


FIG 1.9 – Grafic ce reprezintă apariția minimelor energiei potențiale pentru ^{106}Ru și ^{116}Pd

1.8 Determinarea energiei de excitare

Expresia energiei de excitare a sistemului este foarte importantă pentru această lucrare, ea fiind folosită pentru determinarea emisiei de neutroni după fisiune. Pentru a determina ecuația acesteia se va considera următoarea relație între nucleul inițial și sistemul dinuclear rezultat [14]

$$U = E^* + U_{DNS}$$

(1.18)

unde, folosind formula semi-empirică a masei

$$U = (Zm_p + Nm_n)c^2 - U^{MP}$$

(1.19)

astfel, egalând și simplificând ecuațiile [17]

$$E^* = U_1^{MP} + U_1^{MP} - U^{MP} - V_{int}$$

(1.20)

Din expresia energiei de excitare a sistemului se poate determina expresia energiei de excitare a fiecărui fragment [17] [18]

$$E_i^* = E^* \frac{a_i}{a_L + a_H} \approx E^* \frac{A_i}{A_L + A_H}$$

(1.21)

Însă, datorită deformărilor suferite de fragmente, expresia finală a energiei fiecărui nucleu este [17]

$$E'_i{}^* = E_i^* + E^{Def}$$

(1.22)

CAPITOLUL 3 – Modelul Computațional și Implementare

2.1 Scopul programului

Programul creat pentru această lucrare are ca scop calculul probabilităților ca un sistem să emită un anumit număr de neutroni. De asemenea, acesta oferă o analiză a numărului mediu de neutroni emiși pentru o anumită configurație.

2.2 Determinarea probabilității de emiterie a x neutroni

În urma fisiunii nucleare, cele două fragmente rezultate, despre care s-a discutat în capitolul 2, vor încerca să piardă energie prin emiteria unui număr de nucleoni. Probabilitatea de a emite nucleoni poate fi determinată pentru nuclee grele, astfel, datorită unor bariere Coulomb foarte ridicate, numărul de particule cu sarcină emise va fi neînsemnat comparativ cu cel al neutronilor emiși. În cazul unei energii de excitare mai mare de 100 MeV, fragmentele pot emite un număr considerabil de protoni, însă, pentru acest model, energiile vor fi mai mici de 100 MeV [19].

Pentru energii de excitare de valori egale sau mai mari de 15MeV, probabilitatea de a emite un neutron cu energia ε este proporțională cu $\varepsilon e^{-\frac{\varepsilon}{T}}$, unde T este temperatura nucleară a fragmentului și are expresia $T = aE_{max}^{1/2}$ unde a este o constantă a densității nucleare a nucleului respectiv, fiind de ordinul 0.1 MeV [15].

Luând în considerare aceste aspecte și faptul că emiteria de neutroni se realizează atunci când energia permite nucleului, probabilitatea unui nucleu cu energia de excitare E de a emite exact x neutroni este [19]

$$P(E, x) = P(x) - P(x + 1) = I(\Delta_x, 2x - 3) - I(\Delta_{x+1}, 2x - 1) \quad (2.1)$$

unde $I(\Delta_x, 2x - 3)$ este [19]

$$I(\Delta_x, 2x - 3) = 1 - e^{-\Delta_x} \left[1 + \sum_{k=1}^{2x-3} \frac{(\Delta_x)^k}{k!} \right] \quad (2.2)$$

și [19][26]

$$I(\Delta_{x+1}, 2x - 1) = 1 - e^{-\Delta_{x+1}} \left[1 + \sum_{k=1}^{2x-1} \frac{(\Delta_{x+1})^k}{k!} \right]$$

$$(2.3)$$

fiind funcțiile gamma incomplete ale lui Pearson,

iar $\Delta_x = \frac{E - \sum_1^x B_i}{T_i}$, unde B_i reprezintă energia de legătură a neutronului și $T_i = \left(\frac{E_i^*}{1.5a_i}\right)^{1/2}$, pentru $a_i = \frac{A_i}{12}$ [19].

Probabilitatea ca un fragment să emită exact un neutron, având energia de excitație E_i^* este definită [19] [26]

$$P_{1n}(E_i^*) = \theta(E_i^* - B_1) - \theta(E_i^* - B_1 - B_2)[1 - e^{-\Delta_2}(1 + \Delta_2)]$$

$$(2.4)$$

unde $\theta(E_i^*) = 1$, pentru $E_i^* > 0$ și unde $\theta(E_i^*) = 0$, pentru $E_i^* < 0$.

Probabilitatea ca sistemul să emită exact k neutroni este

$$P(k) = \sum_k P(x) - P(k - x)$$

$$(2.5)$$

2.3 Arhitectura programului

În această lucrare, arhitectura programului a fost concepută pentru a crea o aplicație web scalabilă și modulară, având partea de back-end realizată în Python, o bază de date de tip SQL, un API creat în Python și o interfață grafică realizată cu HTML, CSS, JavaScript și React.js. Tipul acesta de arhitectură asigură aplicației o separare între straturile de prezentare, logica programului și baza de date, reușind o dezvoltare eficientă.

Partea de back-end a unei aplicații este responsabilă de logica acesteia, de comunicarea cu baza de date și de gestionarea cererilor primite de la partea de front-end (interfața grafică). [20] În lucrarea de față, limbajul de programare Python [21] a fost ales pentru a asigura partea de back-end a programului, acesta oferind un mediu de dezvoltare rapid și flexibil, o sintaxă clară, ușor de înțeles și o performanță decentă.

Baza de date are ca scop stocarea datelor primite în urma rulării programului dezvoltat în Python. SQL este limbajul de gestiune a bazelor de date realizate prin intermediul SQL Server

Management Studio, acesta oferind posibilitatea de manipulare a informației folosind ‘queries’ [22].

API este prescurtarea expresiei Application Programming Interface și oferă puncte de intrare și de ieșire standardizate pentru a permite trimiterea sau primirea datelor transmise între interfața grafică și backend, respectiv baza de date. În cazul de față, API-ul a fost realizat în Python.

Interfața grafică reprezintă înfățișarea aplicației, așa cum o vede utilizatorul și este realizată folosind limbaje de programare web standard, HTML, CSS, JavaScript, cât și tehnologii moderne precum framework-ul React.js, dezvoltat pentru dezvoltarea aplicațiilor din suita Meta (Facebook). Limbajul HTML este folosit pentru a structura conținutul paginii, oferind posibilitatea de a crea relații între elementele acesteia. CSS este utilizat pentru a stiliza pagina și a oferi un aspect vizual plăcut. Funcționalitatea aplicației a fost realizată prin limbajul JavaScript, ce este responsabil pentru interacțiunea dinamică cu utilizatorul și framework-ul React.js care facilitează dezvoltarea interfețelor reutilizabile și interactive [23] [24].

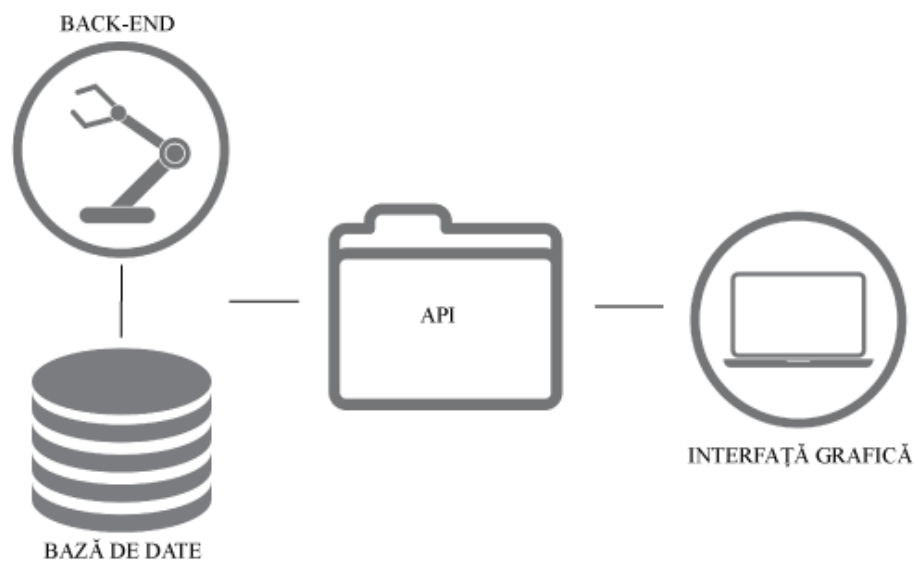


FIG 2.1 – Schemă ce reprezintă arhitectura programului

Schema anterioară a fost creată pentru a exemplifica modul prin care comunică modulele aplicației și pentru a facilita o bună înțelegere a programului.

2.4 Structura părții de back-end

Așa cum s-a discutat în capitolele precedente, partea de back-end este realizată în limbajul de programare Python [21]. În dezvoltarea programului s-au folosit trei librării, *numpy*, *pyodbc* și *numba*. Librăria *numpy* oferă funcții matematice predefinite, ce au o performanță bună, dar și

posibilitatea de a crea matrici într-un mod convenabil și care sunt ușor de manipulat. Pentru a crea conexiunea dintre back-end [20] și baza de date se folosește modulul *pyodbc*, care permite utilizarea funcțiilor *connect*, *cursor* și *execute*. Limbajul de programare Python va rula codul linie cu linie, lucru ce face ca performanța unui program realizat în Python să nu fie la fel de bună ca a unui limbaj de programare ce compilează întreg programul, precum C sau Java, însă, cu ajutorul librăriei *numba*, această problemă a fost remediată, având posibilitatea de a compila funcții în spatele rulării linie cu linie.

Datele introduse în aplicația Python sunt stocate în două fișiere ce sunt în format TSV (Tab Separated Values). Primul fișier va conține trei coloane: masa atomică, numărul atomic și numărul de masă, iar cel de-al doilea fișier va conține valori legate de termeni ai energiei pentru sistemul ales.

În urma rulării aplicației, datele obținute vor fi stocate într-o bază de date SQL, folosind trei tabele diferite. Pentru fiecare rulare a programului, datele stocate anterior în baza de date vor fi șterse și înlocuite de altele noi. Aceasta alegere are ca motivație păstrarea datelor corecte în tabele, fără a se amesteca cu date posibil corupte.

2.5 Structura bazei de date

În această lucrare, baza de date este de tip SQL, având trei tabele, ce vor stoca informațiile primite din back-end. Deoarece toate calculele sunt realizate în Python, iar interfața grafică va comunica cu fiecare tabel individual, s-a ales ca între cele trei tabele să nu existe relații predefinite. Totuși, dacă se dorește vizualizarea împreună a datelor comune existente în tabele, este permisă scrierea unui *query* ce conține sintaxă de tip *JOIN*.

Tabelul *dataInfo2* conține informațiile referitoare la probabilitățile unei configurații definite prin numerele atomice, masa atomică și energiile fragmentelor de a emite un anumit număr de neutroni. Acest tabel este format din optsprezece coloane, așa cum se poate vedea în figura 2.2, având drept cheie primară coloana *indexCount*.

Column	Alias	Table
z1	[Numarul Atomic al Primului Fragment]	dataInfo2
a1	[Masa Atomica a Primului Fragment]	dataInfo2
ex1tot	[Energia Primului Fragment]	dataInfo2
z2	[Numarul Atomic al Celui De-al Doilea Fragment]	dataInfo2
a2	[Masa Atomica a Celui De-al Doilea Fragment]	dataInfo2
ex2tot	[Energia Celui De-al Doilea Fragment]	dataInfo2
prob0	[Probabilitatea de a emite 0 neutroni]	dataInfo2
prob1	[Probabilitatea de a emite 1 neutroni]	dataInfo2
prob2	[Probabilitatea de a emite 2 neutroni]	dataInfo2
prob3	[Probabilitatea de a emite 3 neutroni]	dataInfo2
prob4	[Probabilitatea de a emite 4 neutroni]	dataInfo2
prob5	[Probabilitatea de a emite 5 neutroni]	dataInfo2
prob6	[Probabilitatea de a emite 6 neutroni]	dataInfo2
prob7	[Probabilitatea de a emite 7 neutroni]	dataInfo2
prob8	[Probabilitatea de a emite 8 neutroni]	dataInfo2
prob9	[Probabilitatea de a emite 9 neutroni]	dataInfo2
prob10	[Probabilitatea de a emite 10 neutroni]	dataInfo2
indexCount	[Numarul Randului]	dataInfo2

FIG 2.2 – Structura tabelului *dataInfo2* ce conține probabilitatea de a emite un număr x de neutroni pentru o anumită configurație

Programul realizat în Python este capabil de calculul energiei potențiale minime și a numărului mediu de neutroni emiși într-un act de fisiune, acestea fiind stocate în tabelul *singleInfo*, acesta fiind format din două coloane: *ColumnName* ce reprezintă cheia primară și stochează numele datei și *DataValue* ce stochează valoarea obținută pentru cele două mărimi.

Column	Alias	Table
<i>ColumnName</i>	[Numele datei stocate Umin/NRmediu]	<i>singleInfo</i>
<i>DataValue</i>	[Valoarea marimii stocate]	<i>singleInfo</i>

FIG 2.3 – Structura tabelului *singleInfo* ce stochează energia potențială minimă și numărul mediu de neutroni emiși într-un act de fisiune

Tabelul *averageInfo* este format din două coloane, *Z* ce reprezintă cheia primară, stocând numărul atomic al nucleului și *Value* ce păstrează valoarea numărului mediu de neutroni emiși în urma fisiunii nucleare pentru diferite valori ale numărului atomic.

Column	Alias	Table
Z	[Numarul atomic al nucleului]	averagelInfo
Value	[Numarul de electroni emisi per fisiune]	averagelInfo

FIG 2.4– Structura tabelului *averagelInfo* ce stochează numărul atomic *Z* și valoarea numărului mediu de neutroni emiși

2.6 Structura API

În această lucrare, *API*-ul are ca scop crearea unor puncte de ieșire din baza de date, ce vor fi accesate de către aplicația web ce reprezintă interfața grafică a programului. În general, un *API* permite folosirea metodelor *POST*, *PUT*, *PATCH*, *GET* și *DELETE*, facilitând astfel obținerea, adăugarea și stergerea informațiilor din baza de date, însă, în acest caz particular, doar metoda *POST* este folosită. [25]

Pentru a crea un punct de ieșire sau de intrare, este necesară stocarea acestuia pe un server, care poate fi local sau susținut de un host. În această lucrare s-a folosit un server local, ceea ce face ca punctele *API*-ului să fie accesibile doar clienților ce sunt conectați la rețeaua locală a serverului. Totuși, pentru a accesa aceste date, este necesară cunoașterea *URL*-ului punctului ce se dorește a fi accesat, acesta fiind format dintr-o structură generică *http://localhost:5000/* ce reprezintă numele serverului și *PORT*-ul pe care rulează acesta, alături de o structură particulară pentru fiecare metodă ce este de forma *api/nume*, unde *nume* este locația exactă pe server a punctului de intrare sau de ieșire.

Metoda *POST* oferă posibilitatea de a cere date din partea serverului folosind anumiți parametri, ce sunt folosiți pentru rularea unui *query* în baza de date ce va returna utilizatorului doar valorile cerute pentru o anumită configurație și nu pentru toate configurațiile din baza de date.

În cele ce urmează se va prezenta o porțiune din codul scris pentru *API*, unde s-au folosit librăriile *flask*, *flask-cors* și *pyodbc*, unde primele două oferă acces la crearea serverului și definesc tipul aplicației, iar cea din urmă facilitează accesul cu baza de date.


```

@app.route('/api/data', methods=['POST'])
def get_data():
    z1 = request.json['z1']
    a1 = request.json['a1']
    z2 = request.json['z2']
    a2 = request.json['a2']
    conn = pyodbc.connect(
        'DRIVER={SQL
SERVER};SERVER=localhost;DATABASE=licenta;UID=local;PWD=123456')
    cursor = conn.cursor()
    query = 'SELECT ex1tot, ex2tot, prob0, prob1, prob2, prob3, prob4, prob5, prob6, prob7,
prob8, prob9, prob10 FROM dataInfo2 WHERE Z1 = ? AND A1 = ? AND Z2 = ? AND A2 =
?'
    cursor.execute(query, (z1, a1, z2, a2))
    rows = cursor.fetchall()
    data = []
    column_names = [column[0] for column in cursor.description]
    for row in rows:
        row_dict = {}
        for i in range(len(column_names)):
            row_dict[column_names[i]] = row[i]
        data.append(row_dict)
    cursor.close()
    conn.close()
    return jsonify(data)

```

FIG 2.5– Porțiune de cod Python ce exemplifică modul de creare a unei metode POST care primește ca parametri $z1$, $a1$, $z2$, $a2$, rulează un query în baza de date pentru a obține energiile fragmentelor și probabilitățile de emisie de neutroni, iar apoi le returnează către aplicația web

Acest cod prezintă modul prin care se crează un punct de ieșire prin metoda *POST*, având ca locație *api/data*. Punctul de ieșire preia parametrii aleși de utilizator prin funcția *request.json*, apoi se conectează la baza de date prin *pyodbc.connect()*. Folosind parametrii obținuți, se transmite bazei de date un *query* ce va returna datele cerute pentru configurația definită de acei parametri, urmând ca mai apoi să fie executat prin funcția *cursor.execute()*. În continuare, se pregătesc și se separă datele obținute, ca în final, *API*-ul să returneze informația codată în format *json*.

În componența acestei lucrări există patru metode definite, câte una pentru fiecare tabel din baza de date, iar una auxiliară pentru tabelul *dataInfo2*, cu scopul de a trimite date diferite cu parametri diferiți către partea de front-end.

2.7 Structura interfeței grafice

Interfața grafică a programului, cunoscută drept parte de *front-end*, este o aplicație web, creată cu limbaje de programare specifice, acestea fiind *HTML*, *CSS*, *JavaScript* și *framework*-ul

React.js. Fiecare dintre acestea au roluri importante în dezvoltarea programului, iar în acest subcapitol se va discuta despre funcționalitățile folosite și modul prin care a fost configurată interfața grafică [13].

HTML provine de la HyperText Markup Language și este cel mai folosit instrument de a structura conținutul unei aplicații web. Acesta prezintă o structură simplă, fiind compus din elemente ce descriu și delimitează diferite părți ale paginii. Elementele HTML sunt sub formă de etichete, ele fiind sub formă de pereche, iar în interiorul perechii se va afla conținutul afișat. [19]

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />

    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />

    <title>Titlu Licenta</title>
  </head>
  <body
    style="
      background-color: rgb(40, 56, 69);
      display: grid;
      place-items: center;
      height: 100vh;
    "
  >
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
  </body>
</html>
```

FIG 2.6– Porțiune de cod HTML ce prezintă sintaxa și funcționalitățile acestuia

Fragmentul de cod prezentat în figura 2.6 exemplifică sintaxa și structura unui fișier de tip HTML, acesta fiind delimitat de cele două etichete `<html>`. Prima linie a documentului descrie tipul programului, apoi conținutul este delimitat în două fragmente, *head* și *body*, astfel, în prima parte se găsesc elemente de configurație a paginii, precum titlul și cuplarea cu fișiere de tip

JavaScript, iar în cea de-a doua parte se stilizează eticheta printr-un atribut de tip *style* și se definește elementul *div* ce reprezintă rădăcina proiectului.

CSS provine de la Cascading Style Sheets și este un limbaj ce permite stilizarea elementelor HTML, implicit a paginii web. Acesta poate fi aplicat tuturor elementelor, claselor sau elementelor individuale, astfel se permite stratificarea stilizării.

```
*{
  padding:0;
  margin:0;
}
.menubar{
  display: flex;
  justify-content: center;
  gap: 5vw;
}
button{
  min-width: 15vw;
  min-height: 5vh;
  background-color: rgb(242,149,89);
  margin-bottom: 1vh;
  margin-top: 3vh;
  cursor: pointer;
  border: none;
  border-radius: 1vh;
  transition: 0.75s;
}
button:hover{
  background-color: rgb(228, 169, 130);
  min-width: 16vw;
}
#about{
  text-align: justify;
  padding: 1vh;
}
```

FIG 2.7– Fragment de cod CSS ce prezintă sintaxa și funcționalitățile acestuia

În figura 2.7 se poate observa sintaxa codului CSS și modul prin care se pot stiliza elementele HTML, pornind de la nivel general până la nivel particular. Pentru a influența stilul unui element HTML, sintaxa este sub forma *nume_Nivel {stilizare}*, astfel, în funcție de gradul nivelului, numele acestuia poate avea ca prefix caracterele *.* pentru *clase*, respectiv *#* pentru *id* sau poate fi fără prefix pentru numele etichetei. Dacă se dorește stilizarea tuturor elementelor din document, se poate folosi caracterul *** ca numele nivelului.

Pentru a putea interacționa cu elementele HTML și a controla elementele aplicației, se folosește limbajul de programare JavaScript, acesta fiind unul interpretat și orientat pe obiecte, având ca scop adăugarea de interactivitate și funcționare dinamică. JavaScript poate fi folosit și ca limbaj de back-end, având tehnologii precum *JSON* sau *Ajax*, facilitând crearea de servere și manipularea de date.

```
state = {
  currentPage: 'withoutData',
};

handleStateChange = (page) => {
  this.setState({ currentPage: page });
};

render() {
  const { currentPage } = this.state;

  if (currentPage === 'withData') {
    return <WithDataPage />;
  }

  else if (currentPage === 'home') {
    return <LandingPage />;
  }
}
```

FIG 2.8– Fragment de cod JavaScript ce prezintă sintaxa acestuia și modul prin care se controlează elementele paginii

Exemplul de cod scris în JavaScript din figura 2.8 prezintă modul prin care se definește valoarea atributului `currentPage` a unei clase, urmat de o funcție nedefinită ce gestionează schimbarea valorilor aceluși atribut. Următoarele linii de cod stabilesc conținutul ce va fi afișat utilizatorului în funcție de valoarea atributului definit anterior.

React.Js este un framework JavaScript, folosit pentru construirea unor elemente reutilizabile, ce sunt alcătuite din cod JavaScript și componente HTML [24]. Acesta folosește un model de reprezentare virtual ce conține componentele create și starea acestora, astfel, atunci când starea unui component se va schimba, React.Js va remodela acel element, afișând pe ecran varianta actualizată a paginii. Alături de componentele menționate, ce sunt reprezentate sub formă de obiecte, deci construite pe baza unei clase, React.Js introduce și o serie de funcții prestabilite ce ajută la manipularea stării componentelor.

```

import React, { Component } from 'react';
import './landingPage.css';

class LandingHeader extends Component {
  handleStateChange = (page) => {
    const { onStateChange } = this.props;
    onStateChange(page);
  };

  render() {
    return (
      <React.Fragment>
        <header>
          <div><h1 style={{ color: "white" }}>TITLUL LICENȚEI</h1></div>
          <div className="menubar">
            <button onClick={() => this.handleStateChange('home')}
              className="menuItem"><p>Home</p></button>
            <button onClick={() => this.handleStateChange('withData')}
              className="menuItem"><p>With Experimental Data</p></button>
            <button onClick={() => this.handleStateChange('withoutData')}
              className="menuItem"><p>Without Experimental Data</p></button>
          </div>
        </header>
      </React.Fragment>
    );
  }
}

export default LandingHeader;

```

FIG 2.9– Porțiune de cod scrisă prin React.js, unde se poate observa structura și sintaxa componentei *LandingHeader*

Fragmentul de cod prezentat anterior reprezintă modul prin care se crează o componentă React.js și cum se definesc funcționalitățile acesteia. Primul pas este importarea modului React, respectiv a clasei Component și definirea clasei LandingHeader ce moștenește clasa Component. În interiorul clasei definite, se stabilește modalitatea de gestionare a schimbării parametrului *currentPage*, urmat de funcția *render()* ce va afișa pe ecran elementul părinte *React.Fragment*, ce conține etichetele HTML.

CAPITOLUL 4 – Rezultate

3.1 Probabilitatea de a emite neutroni

Obiectivul principal al acestei lucrări este acela de a afla probabilitatea unei configurații de a emite neutroni, iar în cele ce urmează se vor prezenta rezultatele obținute pentru o parte dintre configurațiile alese.

În tabelul 3.1 au fost stocate date ce reprezintă probabilitatea unei configurații formate dintr-un nucleu de Ni ($Z=28$) și un nucleu de Sm ($Z=62$) de a emite un anumit număr de neutroni în funcție de masele atomice ale fragmentelor și a energiei totale de excitare.

TABEL 3.1 – Probabilități pentru Ni ($Z=28$) și Sm ($Z=62$)

Parametrii configurației			Probabilitatea pentru n neutroni (%)					Total
A ₁	A ₂	E* (MeV)	0	1	2	3	4	
		7.841	99.387	0.6133	0.00	0.00	0.00	
66	156	11.674	61.295	37.973	0.732	0.00	0.00	
		17.163	12.152	62.953	24.096	0.799	0.0001	
		7.137	100.00	0.00	0.00	0.00	0.00	
68	154	15.027	27.656	53.587	18.75	0.007	0.00	100
		19.692	5.216	38.959	51.151	4.622	0.0533	
		6.725	100.00	0.00	0.00	0.00	0.00	
70	152	20.832	3.681	29.265	54.604	11.780	0.6695	
		16.199	20.172	50.054	29.373	0.4	0.0011	

Pentru a oferi o perspectivă mai bună asupra vizualizării datelor din tabelul 3.1 și a analizei acestora, în figurele ce urmează se vor reprezenta grafic probabilitățile calculate sub forma unor histograme. Aceste grafice sunt extrase din aplicația realizată, iar codul este prezentat la finalul subcapitolului.

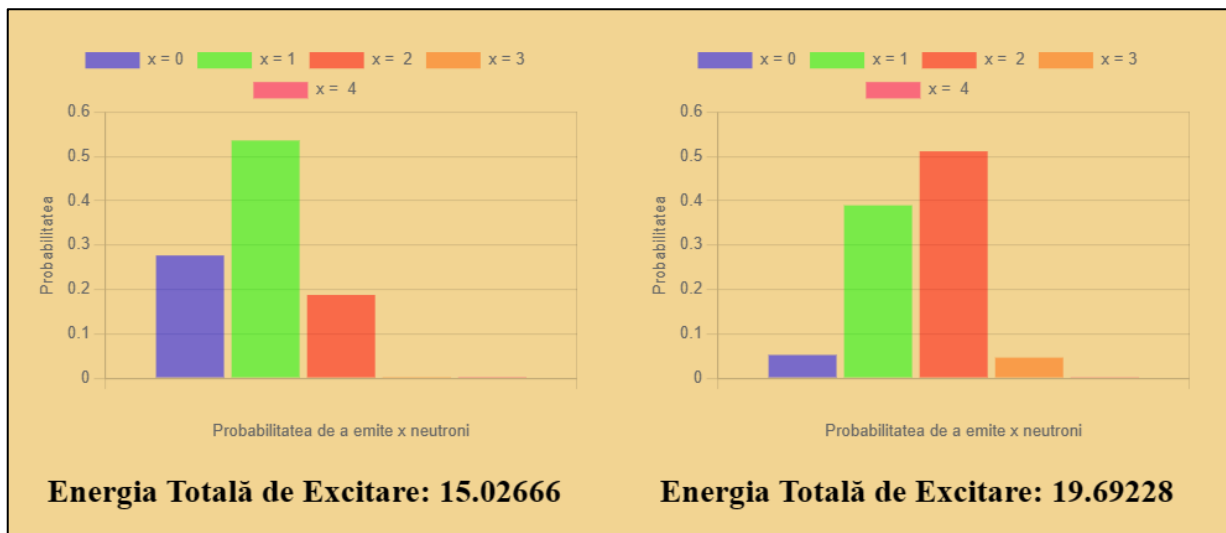


FIG 3.1– Histogramă ce reprezintă grafic probabilitatea de a emite un număr x de neutroni pentru configurația Ni ($Z=28$, $A = 68$), Sm ($Z = 62$, $A = 154$), în stânga pentru energia de excitare egală cu 15.02666 MeV, iar în dreapta pentru energia de excitare egală cu 19.69228 MeV

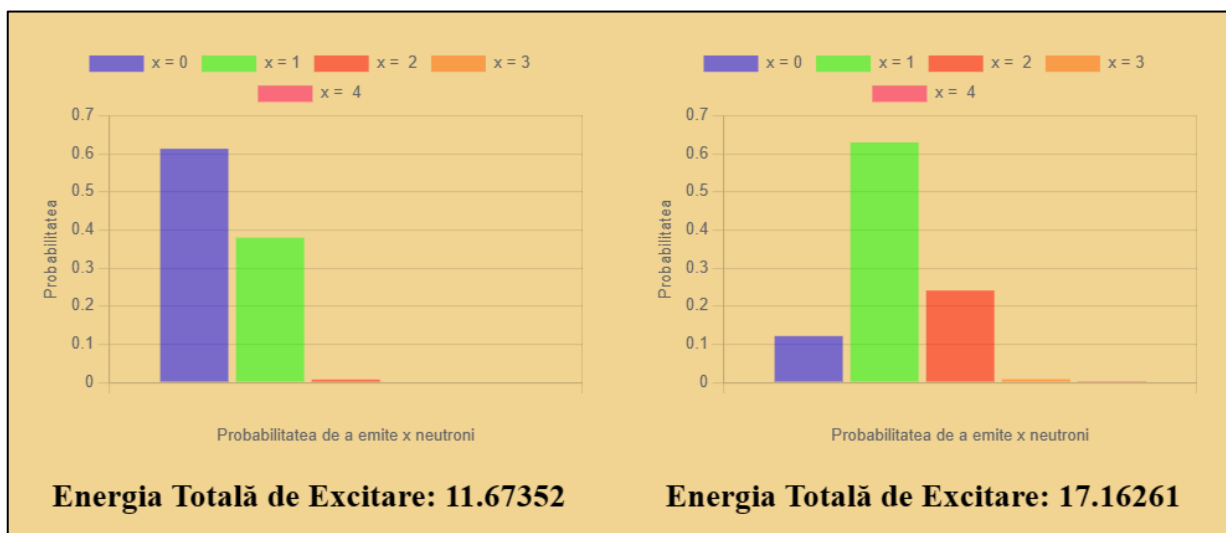


FIG 3.2– Histogramă ce reprezintă grafic probabilitatea de a emite un număr x de neutroni pentru configurația Ni ($Z=28$, $A = 66$), Sm ($Z = 62$, $A = 156$), în stânga pentru energia de excitare egală cu 11.673525 MeV, iar în dreapta pentru energia de excitare egală cu 17.16261 MeV

În tabelul 3.2 au fost stocate date ce reprezintă probabilitatea unei configurații formate dintr-un nucleu de Ge ($Z=32$) și un nucleu de Ce ($Z=58$) de a emite un anumit număr de neutroni în funcție de masele atomice ale fragmentelor și a energiei totale de excitare.

TABEL 3.2 – Probabilități pentru Ge (Z=32) și Ce (Z=58)

Parametrii configurației			Probabilitatea pentru n neutroni (%)						
A ₁	A ₂	E* (MeV)	0	1	2	3	4	5	Total
76	146	17.955	9.838	53.453	32.985	3.723	0.0006	0.00	100
78	144	13.3038	42.810	50.138	7.049	0.0026	0.00	0.00	
		18.1130	8.941	46.852	40.975	3.226	0.0061	0.00	
80	142	24.5930	0.663	11.929	52.085	31.313	4.0091	0.0001	
82	140	12.7955	61.996	35.465	2.538	0.00	0.00	0.00	
84	138	15.0791	15.152	61.115	21.716	2.017	0.00	0.00	

Urmând exemplul de mai sus, în continuare se vor reprezenta grafic probabilitățile de a emite un număr de neutroni pentru configurații ale nucleelor de Ge (Z=32) și Ce (Z=58).

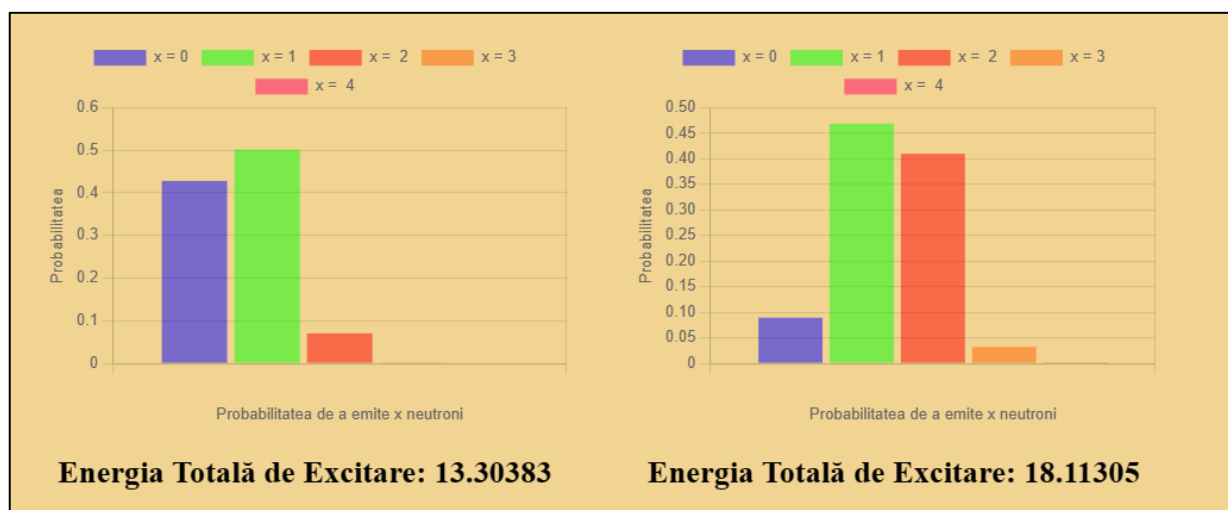


FIG 3.3– Histogramă ce reprezintă grafic probabilitatea de a emite un număr x de neutroni pentru configurația Ge (Z=32, A = 78), Ce (Z = 58, A = 144), în stânga pentru energia de excitare egală cu 13.30383 MeV, iar în dreapta pentru energia de excitare egală cu 18.11305 MeV

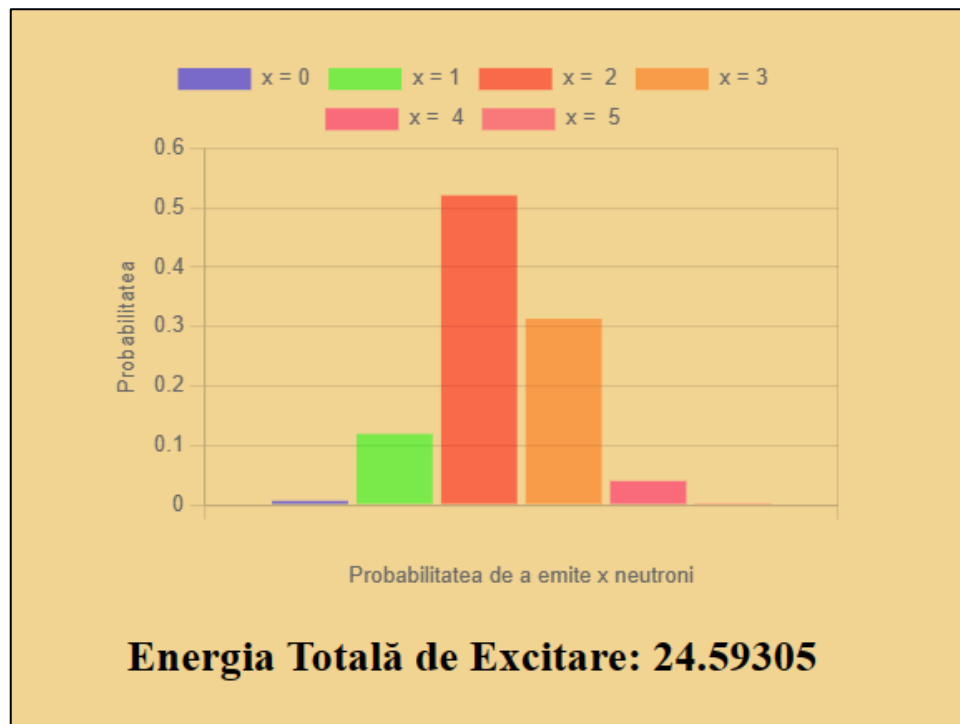


FIG 3.4– Histogramă ce reprezintă grafic probabilitatea de a emite un număr x de neutroni pentru configurația Ge ($Z=32$, $A = 80$), Ce ($Z = 58$, $A = 142$), pentru energia de excitare egală cu 24.59305 MeV

În tabelul 3.3 au fost stocate date ce reprezintă probabilitatea unei configurații formate dintr-un nucleu de Cd ($Z=48$) și un nucleu de Mo ($Z=42$) de a emite un anumit număr de neutroni în funcție de masele atomice ale fragmentelor și a energiei totale de excitare.

TABEL 3.3 – Probabilități pentru Cd ($Z=48$) și Mo ($Z=42$)

Parametrii configurației			Probabilitatea pentru n neutroni (%)							Total
A_1	A_2	E^* (MeV)	0	1	2	3	4	5	6	
112	110	17.0696	11.377	51.888	33.481	3.253	0.00	0.00	0.00	100
		18.1991	7.340	44.485	41.781	6.393	0.0005	0.00	0.00	
116	106	29.3739	0.072	2.991	33.051	47.078	16.647	0.161	0.0001	
118	104	31.7376	0.03	1.64	23.202	47.881	26.833	0.413	0.0016	
120	102	30.8544	0.055	2.482	25.752	49.311	22.102	0.297	0.0007	
122	100	30.7219	0.045	2.467	25.548	52.930	18.648	0.36	0.0004	

Asemenea configurațiilor precedente, se va reprezenta grafic probabilitatea de a emite un număr de electroni pentru o anumită configurație, în funcție de energia de excitare și de masele atomice ale nucleelor.

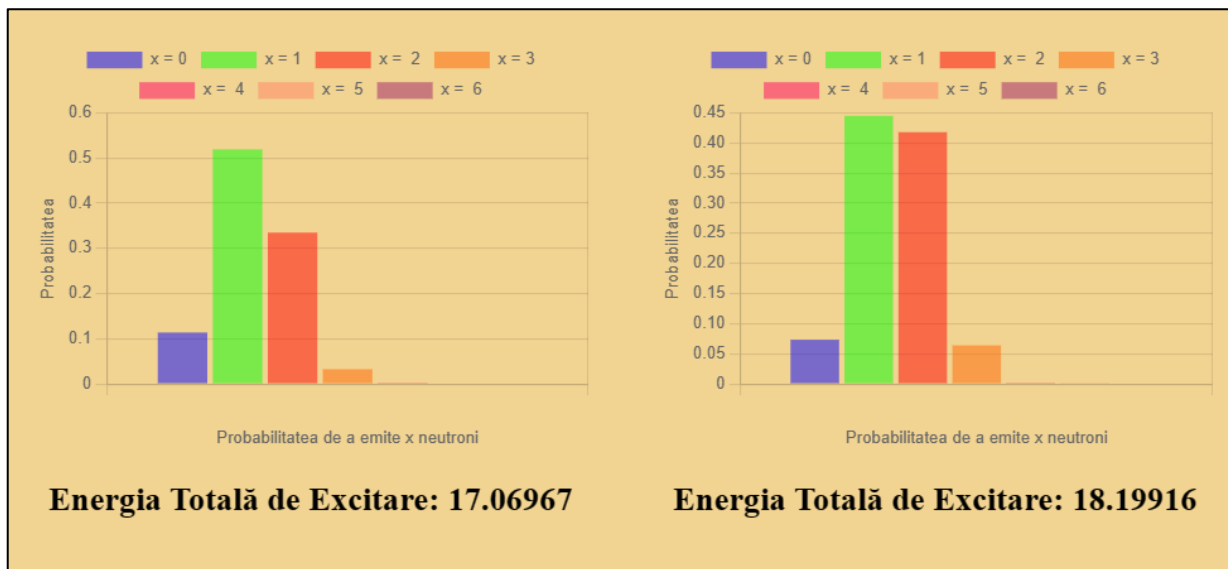


FIG 3.5– Histogramă ce reprezintă grafic probabilitatea de a emite un număr x de neutroni pentru configurația Cd (Z=48, A = 112), Mo (Z = 42, A = 110), în stânga pentru energia de excitare egală cu 17.06967 MeV, iar în dreapta pentru energia de excitare egală cu 18.19916 MeV

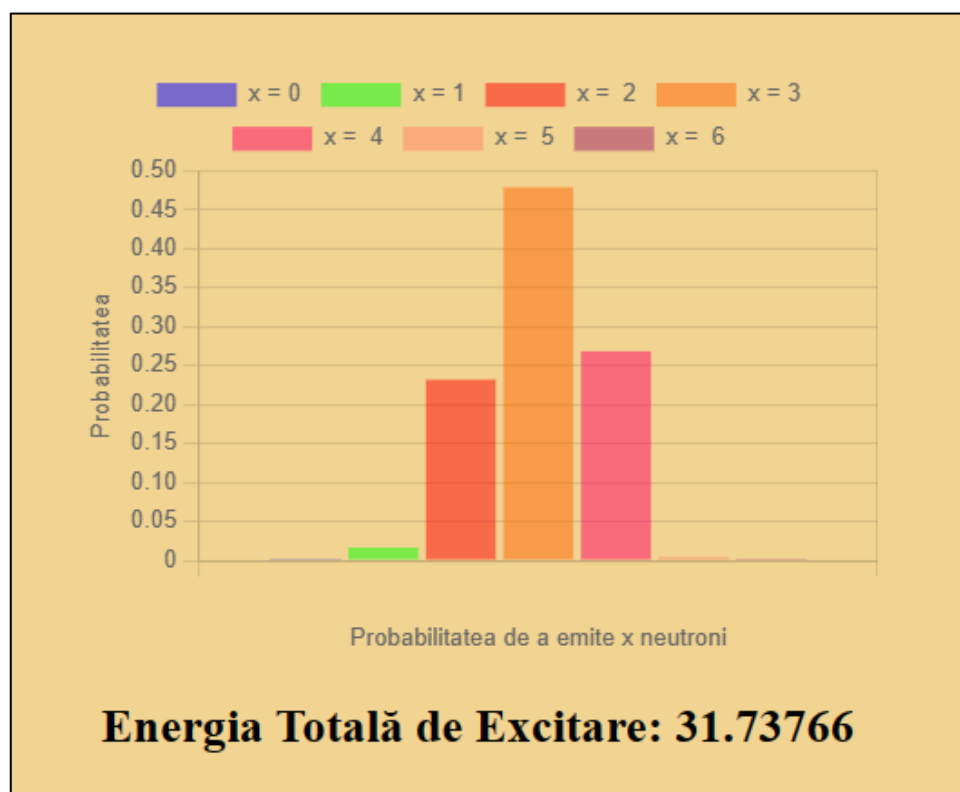


FIG 3.6– Histogramă ce reprezintă grafic probabilitatea de a emite un număr x de neutroni pentru configurația Cd (Z=48, A = 112), Mo (Z = 42, A = 110), pentru energia de excitare egală cu 31.73766 MeV

Ultima configurație ce va fi prezentată în această lucrare este formată dintr-un nucleu de Nd ($Z=60$) și unul de Zn ($Z=30$), având trei rezultate diferite în funcție de energia totală de excitare și masele atomice ale fragmentelor.

În tabelul 3.4 au fost stocate date ce reprezintă probabilitatea unei configurații formate dintr-un nucleu de Nd ($Z=60$) și Zn ($Z=30$) de a emite un anumit număr de neutroni în funcție de masele atomice ale fragmentelor și a energiei totale de excitare.

TABEL 3.4 – Probabilități pentru Nd ($Z=60$) și Zn ($Z=30$)

Parametrii configurației			Probabilitatea pentru n neutroni (%)					
A_1	A_2	E^* (MeV)	0	1	2	3	4	Total
150	72	12.1725	60.12	38.275	1.607	0.00	0.00	100
		19.7525	5.725	42.698	44.357	7.171	0.048	
152	70	16.8144	16.411	62.635	20.303	0.651	0.00	

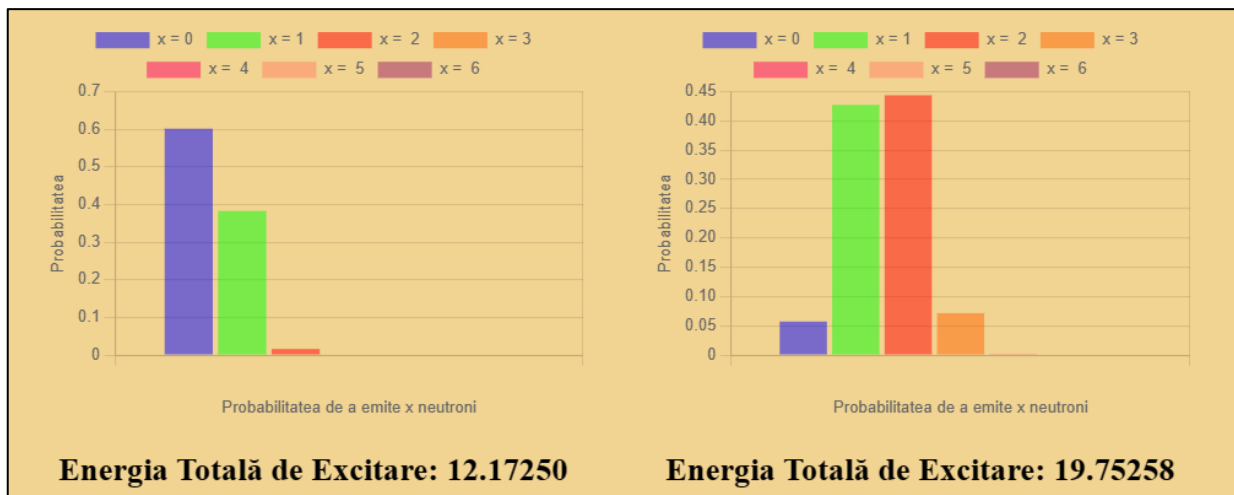


FIG 3.7– Histogramă ce reprezintă grafic probabilitatea de a emite un număr x de neutroni pentru configurația Nd ($Z=60$, $A = 150$), Zn ($Z=30$, $A = 72$), în stânga pentru energia de excitare egală cu 12.1725 MeV, iar în dreapta pentru energia de excitare egală cu 19.75258 MeV

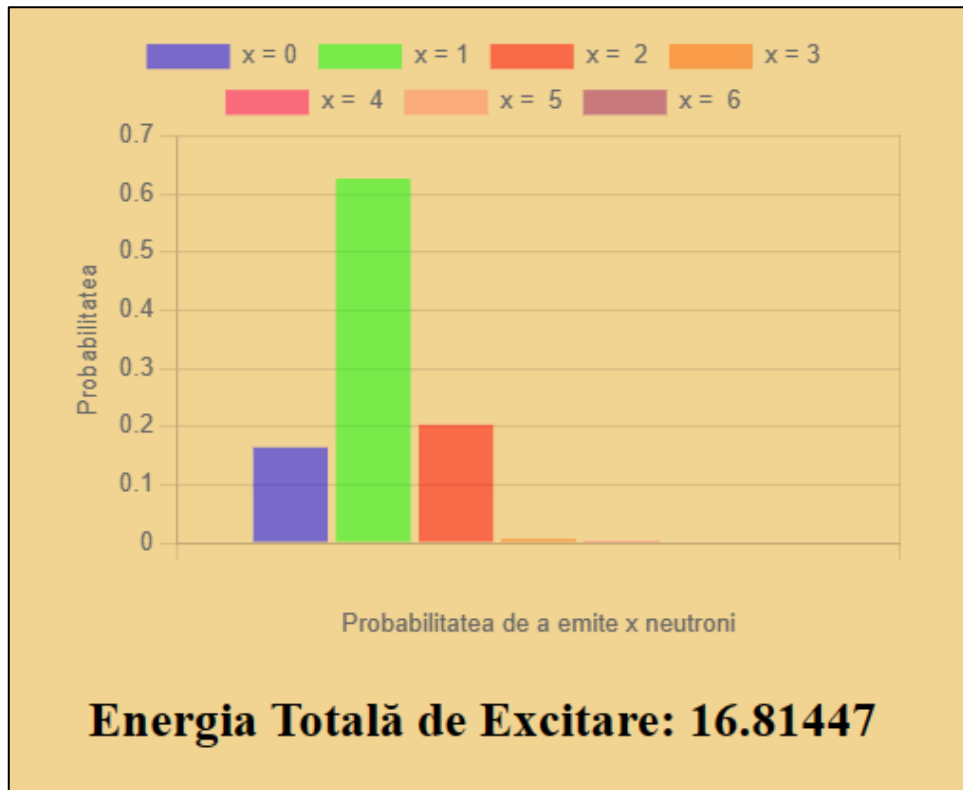


FIG 3.8– Histogramă ce reprezintă grafic probabilitatea de a emite un număr x de neutroni pentru configurația Nd ($Z=60, A = 152$), Zn ($Z=30, A = 70$), pentru energia de excitare egală cu 16.81447 MeV

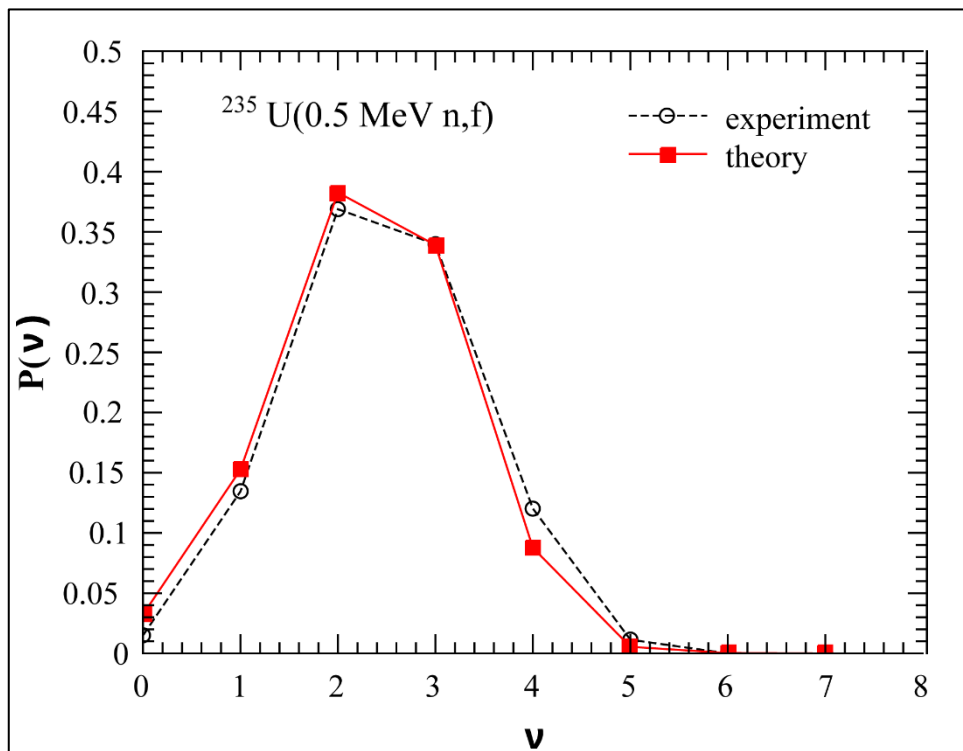


FIG 3.9– Grafic ce compară probabilitatea calculată cu probabilitatea experimentală [26] de emisie a n neutroni pentru ^{235}U

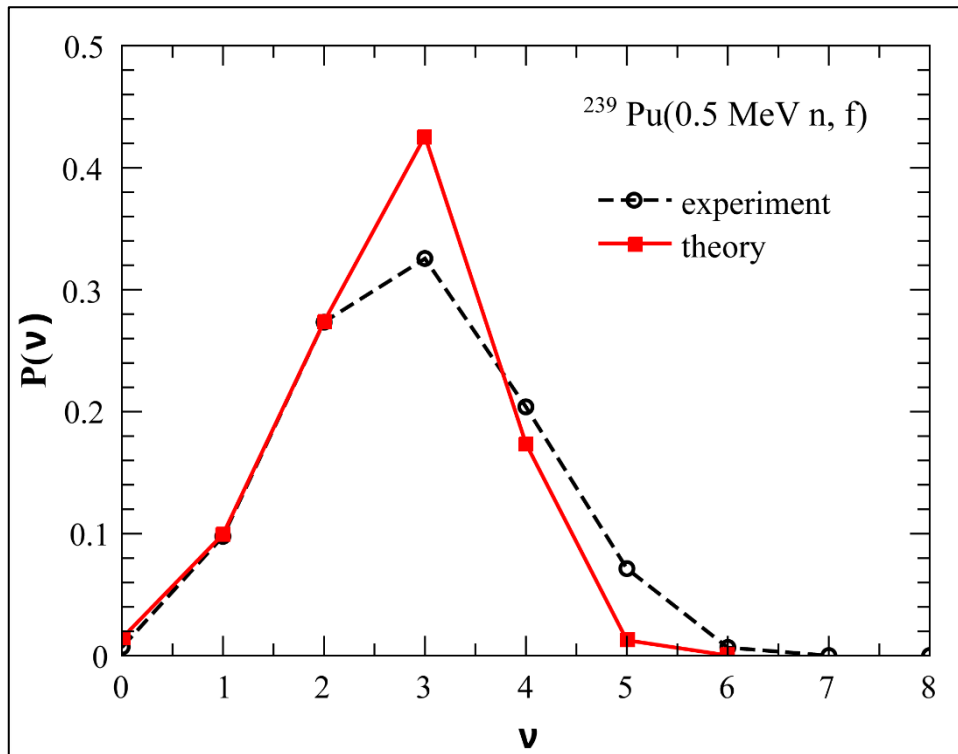


FIG 3.10– Grafic ce compară probabilitatea calculată cu probabilitatea experimentală [26] de emisie a n neutroni pentru ^{239}Pu

Graficele prezentate anterior sunt realizate cu ajutorul librăriei *Chart.js*, o librărie ce oferă posibilitatea de a crea grafice interactive, ușor de implementat și care pot schimba dinamic valorile prezentate odată cu alegerea unei noi configurații.

Pentru o mai bună înțelegere a modului prin care au fost create acestea, în continuare este atașat codul folosit pentru crearea componentei reutilizabile *Histogram* prin intermediul librăriilor *Chart.js* și *React.js*.

```
import React, { useEffect, useRef } from "react";
import Chart from "chart.js/auto";

const Histogram = ({
  ex1tot,
  prob0,
  prob1,
  prob2,
  prob3,
  prob4,
  prob5,
  prob6,
}) => {
  const chartRef = useRef(null);
  useEffect(() => {
    const ex1totArray = [ex1tot];
    const datasets = [
      {
        label: "x = 0",
        data: prob0,
        backgroundColor: "rgba(0, 0, 255, 0.5)",
      },
      {
        label: "x = 1",
        data: prob1,
        backgroundColor: "rgba(0, 255, 0, 0.5)",
      },
      {
        label: "x = 2",
        data: prob2,
        backgroundColor: "rgba(255, 0, 0, 0.5)",
      },
      {
        label: "x = 3",
        data: prob3,
        backgroundColor: "rgba(255, 100, 0, 0.5)",
      },
      {
        label: "x = 4",
        data: prob4,
        backgroundColor: "rgba(255, 0, 100, 0.5)",
      },
      {
        label: "x = 5",
        data: prob5,
        backgroundColor: "rgba(255, 130, 100, 0.5)",
      },
    ];
  });
};
```

```

    },
    {
      label: "x = 6",
      data: prob6,
      backgroundColor: "rgba(155, 30, 100, 0.5)",
    },
  ],
  const drawHistogram = () => {
    const canvas = chartRef.current;
    const context = canvas.getContext("2d");
    new Chart(context, {
      type: "bar",
      data: {
        labels: ex1totArray.map((value, index) => ``),
        datasets: datasets,
      },
      options: {
        responsive: true,
        scales: {
          x: {
            beginAtZero: true,
            title: {
              display: true,
              text: "Probabilitatea de a emite x neutroni",
            },
          },
          y: {
            beginAtZero: true,
            title: {
              display: true,
              text: "Probabilitatea",
            },
          },
        },
      },
    });
    drawHistogram();
  }, [ex1tot, prob0, prob1, prob2, prob3, prob4, prob5, prob6]);

  return <canvas ref={chartRef} width={400} height={300} />;
};

export default Histogram;

```

FIG 3.11– Secțiune de cod prin care se crează componenta reutilizabilă Histogram ce este folosită pentru efectuarea graficelor prezentate anterior

Această secțiune de cod este formată din două părți, primirea valorilor, alături de stocarea lor în șiruri de numere și desenarea propriu-zisă a graficului.

Prima parte a codului stabilește structura componentei Histogram și a parametrilor acesteia, urmând să se definească seturile de date folosite și stocarea valorilor în acestea, iar cea de-a doua parte este inclusă în funcția *drawHistogram*, unde se setează caracteristicile graficului, numele axelor și se crează histograma, urmând să fie exportată în ultimul rând al codului.

3.2 Numărul mediu de neutroni emiși per configurație

În această lucrare s-au calculat probabilitățile de emisie a unui anumit număr de neutroni în urma procesului de fisiune pentru nuclee cu numere atomice cuprinse între $Z = 28$ și $Z = 66$, pentru un total de 19 nuclee, având o energie de excitare de aproximativ 15MeV. Este importantă analiza numărului mediu de neutroni emiși per configurație pentru viitoare aplicații unde se urmărește folosirea de nuclee ce vor emite un număr dorit de neutroni.

În tabelul 3.5 sunt stocate datele referitoare la numărul mediu de neutroni emiși per număr atomic obținute în urma rulării programului realizat pentru această lucrare, rezultatele fiind afișate în ordinea numărului atomic.

TABEL 3.5– Numărul mediu de neutroni emiși per număr atomic

Numărul Atomic Z	Numărul Mediu de Neutroni Emiși
28	1.0936
30	1.5253
32	2.2314
34	2.3854
36	2.8683
38	3.0240
40	2.6320
42	2.9495
44	2.5744
46	2.5681
48	2.9529
50	2.9031
52	3.0553
54	2.9473
56	2.4307
58	2.2867
60	1.5283
62	1.1066
66	0.0012

Din figura 3.13 se poate observa că nucleul cu cel mai mare număr mediu de neutroni emiși este Te ($Z = 52$) având 3.055375 neutroni emiși, urmat de Sr ($Z = 38$) cu 3.024093 neutroni

emiși. În extrema opusă, nucleul cu cel mai mic număr de neutroni emiși este Dy ($Z = 66$), având 0.001279 neutroni emiși per fisiune.

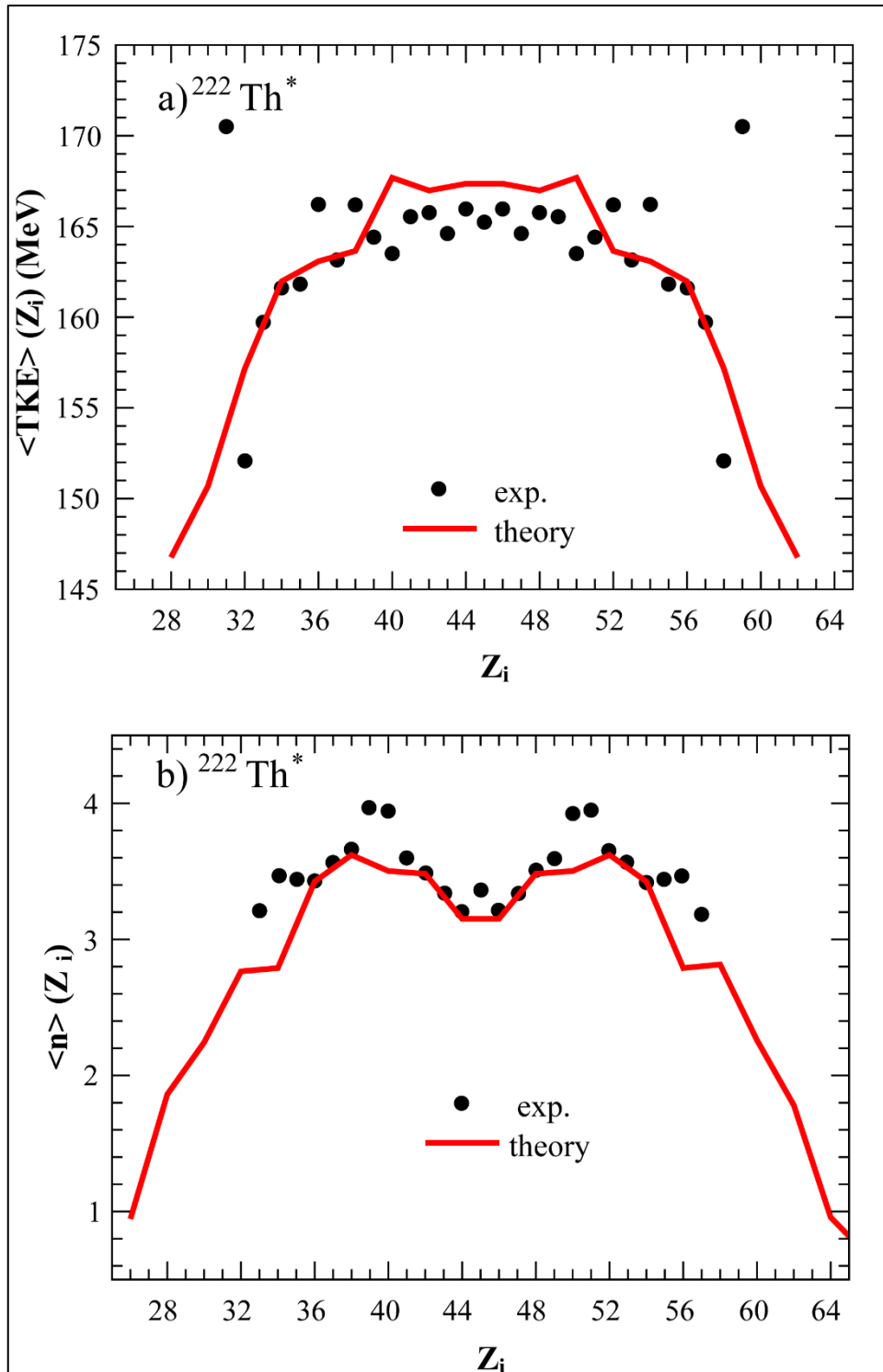


FIG 3.12– Grafic ce compară datele experimentale [27] [28] cu datele calculate referitor la energia cinetică totală, respectiv numărul mediu de neutroni emiși [29] [30] pentru ^{222}Th pentru o energie de excitație de 15MeV

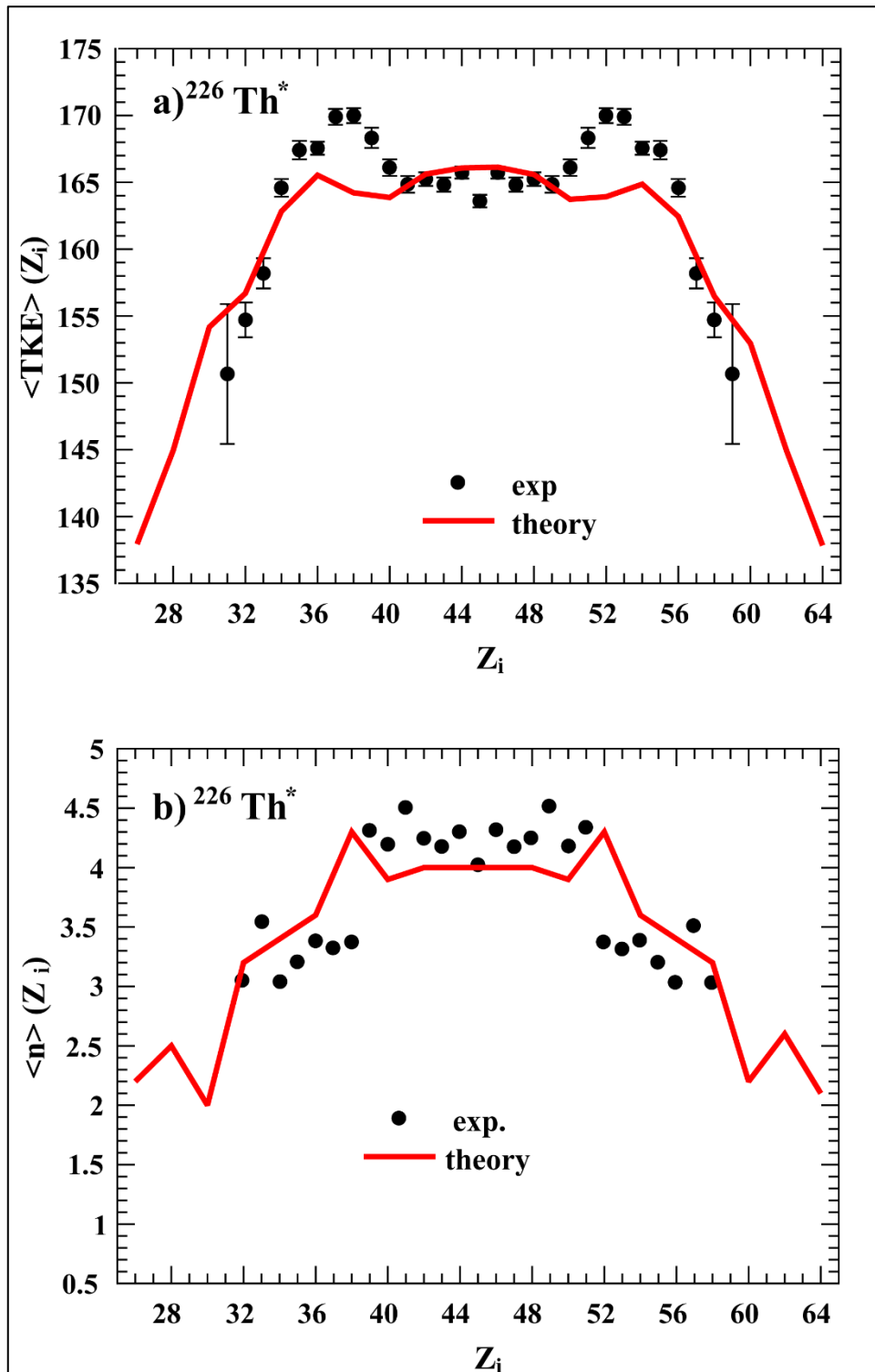


FIG 3.13– Grafic ce compară datele experimentale [27] [28] cu datele calculate referitor la energia cinetică totală, respectiv numărul mediu de neutroni emiși [29][30] pentru ^{226}Th pentru o energie de excitație de 15MeV

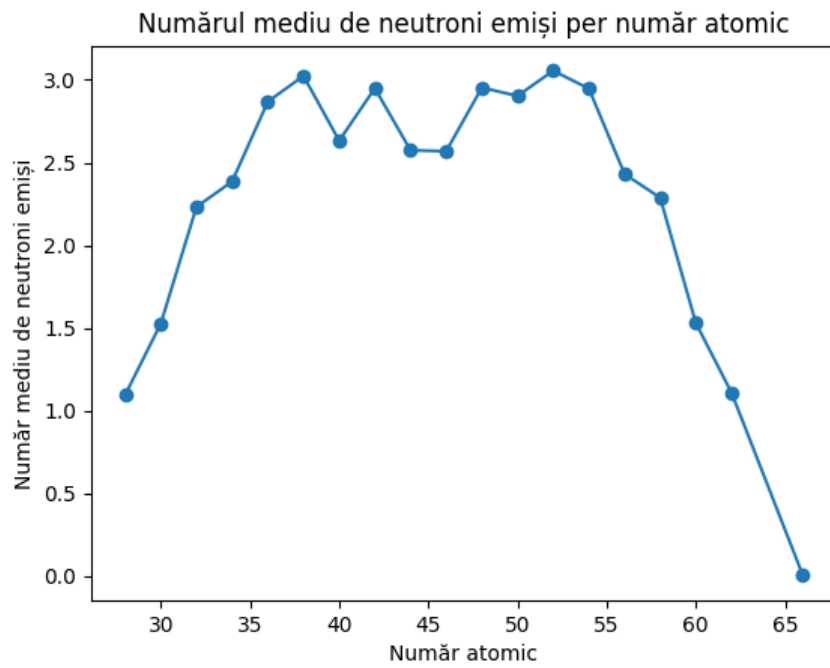


FIG 3.14– Grafic ce reprezintă numărul mediu de neutroni emiși per număr atomic pentru nuclee ce au Z cuprins între 29 și 66

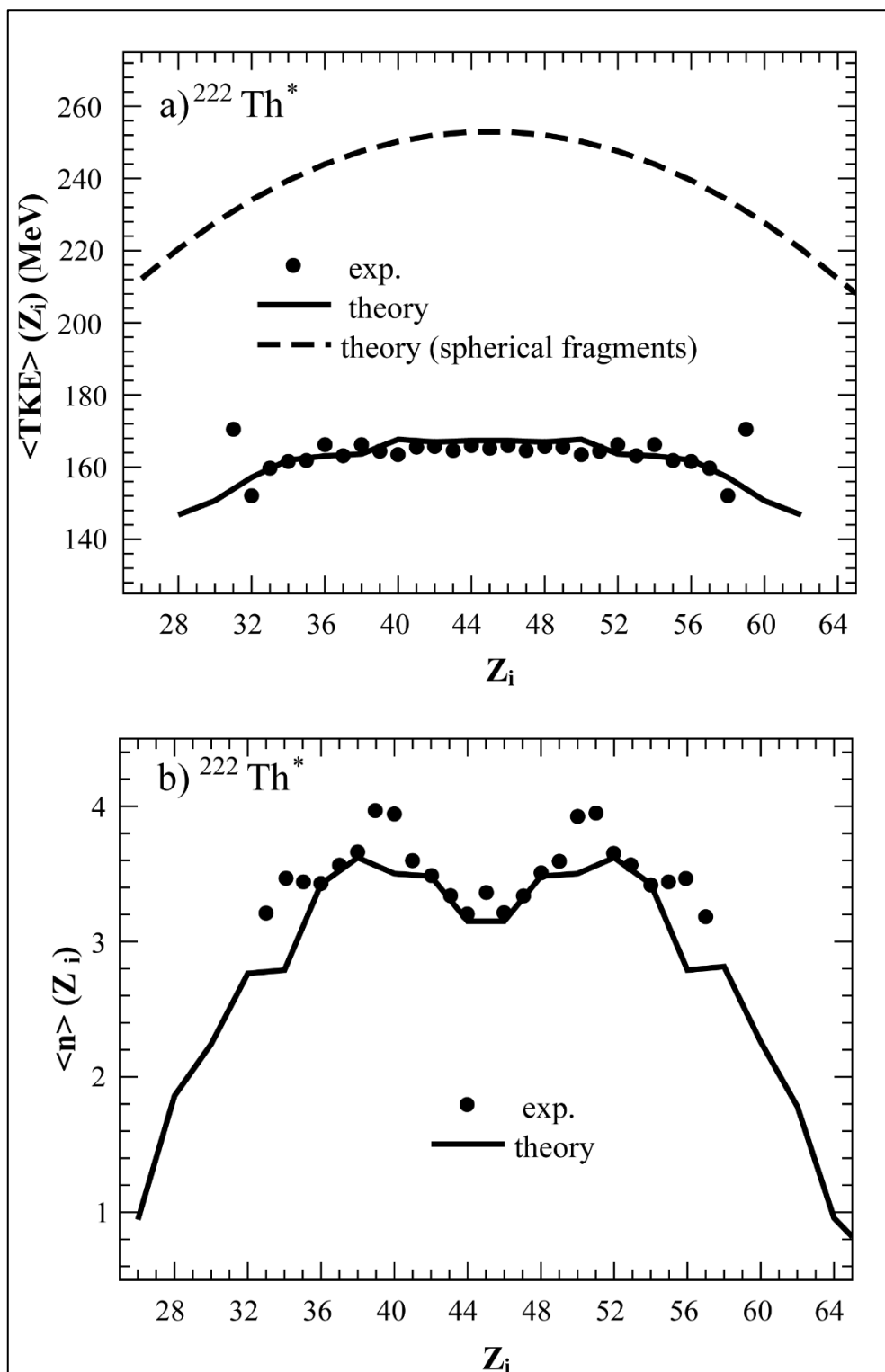


FIG 3.15– Grafice ce prezintă o comparație a rezultatelor[27, 28] pentru nuclee sferice (cu parametrii de deformare = 1) [29][30] cu rezultatele obținute în această lucrare

3.3 Prezentarea aplicației și a funcționalităților sale

Programul realizat pentru această lucrare este specializat în calculul probabilității ca în urma fisiunii nucleare, o anumită configurație să emită un număr n de neutroni, fiind construit pe baza tehnologiilor web *HTML*, *CSS* și *JavaScript*, alături de framework-ul *React.js* pentru partea

vizuală, iar în spatele scenei, aplicația utilizează *Python* și *SQL* pentru gestiunea datelor și efectuarea calculelor complexe, alături de crearea server-ului *API*.

Pentru deschiderea aplicației este necesară folosirea unui *browser web* la alegere și introducerea adresei *localhost:3000*, aplicația rulând pe un server local. Odată ce aplicația este deschisă, se pot distinge trei componente principale: bara de navigare, secțiunea *Despre* și fereastra de subsol.

Bara de navigație este compusă din trei butoane interactive: *Acasă*, *Simularea Probabilității* și *Comparația cu date experimentale*, așa cum se poate observa în figura 3.16.

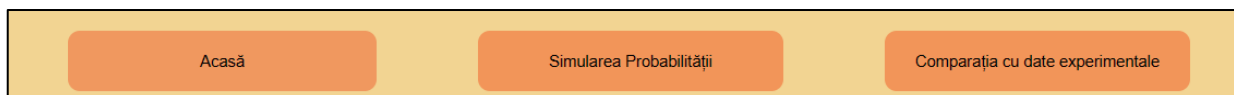


FIG 3.16– Imagine ce reprezintă structura barei de navigare

Aceste butoane au ca scop facilitarea navigării de la o pagină la alta atunci când sunt apăstate, prin schimbarea parametrilor de afișare a aplicației.

Secțiunea *Despre* este reprezentată de un chenar ce conține o scurtă descriere a programului, alături de modul în care este structurat, după cum se poate observa în figura 3.17.

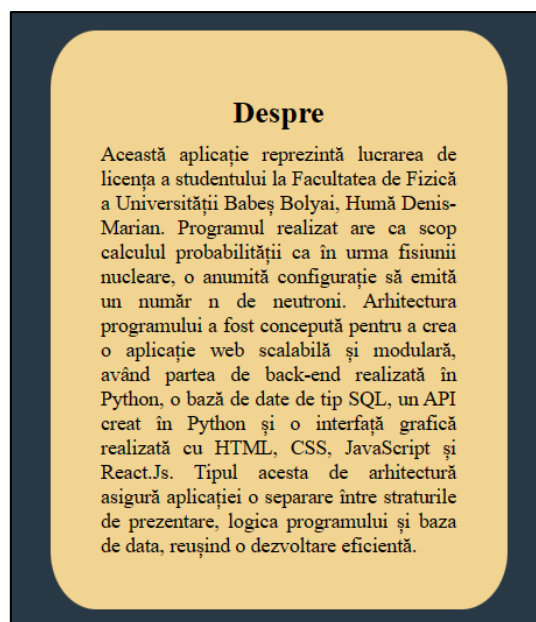


FIG 3.17– Imagine ce prezintă secțiunea *Despre* a paginii de pornire

Fereastra de subsol conține trei obiecte ce pot fi accesate care vor duce spre adrese externe ce conțin informații despre proiect, biografie și o bibliotecă a limbajelor de programare utilizate.



FIG 3.18– Imagine ce prezintă fereastra de subsol a aplicației

Apăsând pe butonul *Simularea Probabilității* din bara de navigare, pe ecran se va afișa un nou chenar ce permite selecția configurațiilor celor două nuclee, așa cum se poate observa în figura 3.19.



FIG 3.19– Imagine care prezintă chenarul ce oferă posibilitatea de selecție a configurațiilor celor două nuclee

Pentru a alege cele două configurații, primul pas este selectarea primei configurații, apăsând pe *Configurația primului nucleu* și alegând din listă una dintre configurațiile existente, urmat de apăsarea butonului *SETEAZĂ PRIMA CONFIGURAȚIE*, lucru ce va trimite spre server opțiunea aleasă și care va aduce înapoi spre aplicație o listă, ce conține configurațiile celui de-al doilea nucleu, ce sunt disponibile pentru configurația primului nucleu ales. Odată ales sistemul dorit, ecranul ar trebui să arate în felul următor



FIG 3.20– Imagine care prezintă modul în care arată chenarul odată ce s-au ales configurațiile celor două nuclee

Apăsând butonul ARATĂ REZULTATE, pe ecran vor fi afișate trei grafice, sub formă de histogramă, ce reprezintă probabilitatea configurației alese de a emite un anumit număr de neutroni în urma procesului de fisiune, iar sub acestea energia totală de excitație a sistemului.

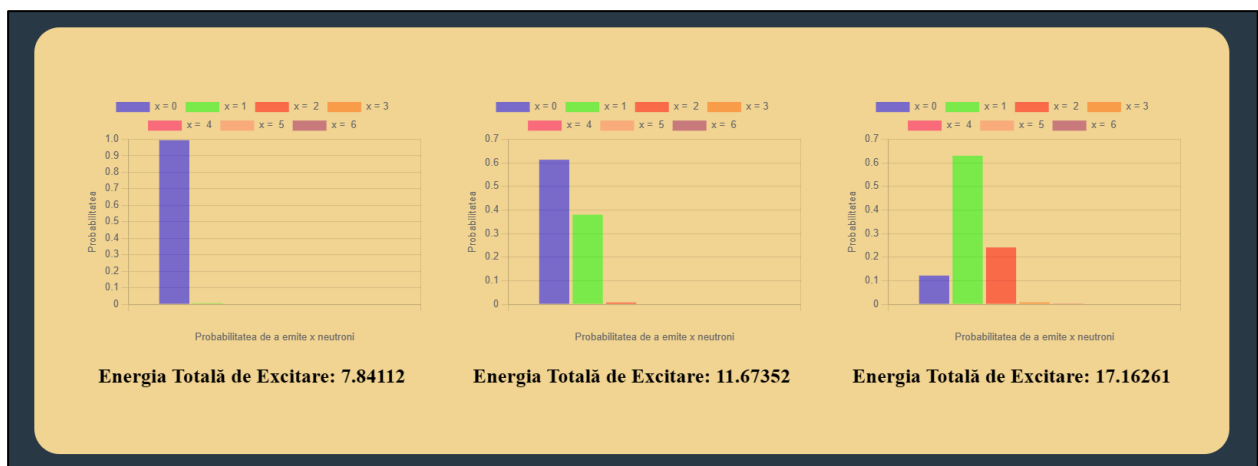


FIG 3.21– Imagine ce prezintă graficele afișate în urma apăsării butonului ARATĂ REZULTATE, acestea reprezentând probabilitatea configurației alese de a emite un anumit număr de neutroni în urma procesului de fisiune

Secțiunea Comparația cu date experimentale poate fi accesată apăsând butonul cu același nume din bara de navigare, unde vor fi prezentate figurile 3.14, respectiv 3.15.

CAPITOLUL 5 – Concluzii

4.1 Descrierea programului realizat

Programul realizat are ca scop calculul probabilităților ca un sistem să emită un anumit număr de neutroni în urma fisiunii nucleare, oferind o analiză a numărului mediu de neutroni emiși pentru o anumită configurație. Această aplicație este una web, scalabilă și modulară, realizată cu ajutorul limbajelor de programare *HTML*, *CSS*, *JavaScript* și a framework-ului *React.js*, având partea de back-end realizată prin *Python* și *SQL*.

Partea de calcul propriu-zisă este realizată în limbajul de programare Python, unde se prelucrează un fișier ce conține parametri legați de termenii de energie a unei liste de sisteme dinucleare, având ca rezultat date legate de parametrii de deformare, probabilitatea de emiteră a unui număr de neutroni calculată până la zece și energia de excitare totală a sistemului.

Aceste date obținute sunt stocate într-o bază de date, în trei tabele, ce comunică cu aplicația web prin intermediul unui API realizat în Python. Interfața grafică a programului primește datele de la server, astfel, se oferă posibilitatea de a alege configurația sistemului dintr-o listă de configurații, ca mai apoi să se realizeze grafice ale probabilităților pentru acea configurație, în funcție de energia de excitare totală.

4.2 Rezultatele obținute

În urma executării programului și a prelucrării rezultatelor, s-au creat o serie de tabele, ce compară datele rezultate, respectiv grafice ce ajută la analizarea acestora. Anterior au fost prezentate o parte dintre rezultatele obținute, fiind analizate datele pentru configurațiile: Ni ($Z = 28$) și Sm ($Z = 62$), Ge ($Z = 32$) și Ce ($Z = 58$), Cd ($Z = 48$) și Mo ($Z = 42$), Nd ($Z = 60$) și Zn ($Z = 30$) pentru diferite valori ale maselor atomice corespunzătoare izotopilor acestor nuclee.

Analizând rezultatele obținute, s-a confirmat dependența probabilității de emiteră a neutronilor de energia de excitare, astfel, configurațiile ce prezentau energii de excitare ridicate aveau probabilități mai mari de a emite un număr mai mare de neutroni și probabilități foarte mici de a emite zero neutroni, pe când configurațiile cu energii mici de excitare aveau probabilități foarte mici de a emite un număr mare de neutroni și probabilități foarte mari de a emite zero neutroni.

S-a observat că pentru o energie de excitare egală, nucleele care emit cei mai mulți neutroni sunt cele de Te ($Z = 52$) și Sr ($Z = 38$), acestea având un număr mediu de neutroni emiși per fisiune egal cu 3.055375, respectiv 3.024093. De asemenea, nucleul cu cel mai mic număr de neutroni emiși este Dy ($Z = 66$), având 0.001279 neutroni emiși per fisiune.

4.3 Concluzii finale

Rezultatele obținute în urma acestei lucrări au fost unele în conformitate cu rezultatele experimentale, având grafice și valori similare. Modelul folosit este unul ideal, aflându-se într-o configurație microcanonică și care poate prezenta mici diferențe față de datele experimentale pentru anumite configurații.

Aplicația dezvoltată poate fi considerată una complexă și care se poate folosi pe viitor în analiza și afișarea datelor.

BIBLIOGRAFIE:

- [1] R.G. Stokstad, *Treatise on Heavy-Ion Science, Vol. 3*, Plenum Press, New York and London, 1985.
- [2] V.V. Volkov, *Deeply Inelastic Nuclear Reactions*, Energoatomizdat, Moscow, 1982.
- [3] G.G. Adamian, N.V. Antonenko, W.Scheid, V.V. Volkov, *Treatments of competition between complete fusion and quasifission in collisions of heavy nuclei*, Nuclear Physics A, **627**, 361-378, 1997.
- [4] G.G. Adamian, N.V. Antonenko, W. Scheid, V.V. Volkov, *Fusion cross sections for superheavy nuclei in the dinuclear system concept*, Nuclear Physics A, **633**, 409-420, 1998.
- [5] G.G. Adamian, N.V. Antonenko, N. Nenoff, W.Scheid, *How to observe hyperdeformed states populated in heavy ion reactions*, Phys. Rev. C, **64**, 2001.
- [6] G.G. Adamian, N.V. Antonenko, R.V. Jolos, Yu.V. Palchikov, W. Scheid, T.M. Shneidman, *Manifestation of cluster effects in the structure of medium mass and heavy nuclei*, Nuclear Physics A, 734, 433-436, 2004.
- [7] G.G. Adamian, N.V. Antonenko, W.Scheid, *Quasifission at extreme sub-barrier energies*, The European Physical Journal A, **48**, 2012.
- [8] Rainer W. Hasse, William D. Myers, *Geometrical Relationships of Macroscopic Nuclear Physics*, Springer-Verlag, Berlin, 1988.
- [9] A.V. Andreev, G.G. Adamian, N.V. Antonenko, S.P. Ivanova, W. Scheid, *Possible explanation of fine structures in mass-energy distribution of fission fragments*, The European Physical Journal A – Hadrons and Nuclei, **22**, 51-60, 2004.
- [10] [30] A.V. Andreev, G.G. Adamian, N.V. Antonenko, S.P. Ivanova, *Bimodality and charge splitting in fission of actinides*, Eur. Phys J A, **26**, 327-332, 2005.
- [11] A.V. Andreev, G.G. Adamian, N.V. Antonenko, *Mass distributions for induced fission of different Hg isotopes*, Phys. Rev. C, **86**, 2012.
- [12] G.G. Adamian, N.V. Antonenko, R.V. Jolos, S.P. Ivanova, O.I. Melnikova, *Effective nucleus-nucleus potential for calculation of potential energy of a dinuclear system*, International Journal of Modern Physics E, **05**, 191-216, 1996.

- [13] A.B. Migdal, *Theory of finite Fermi Systems and Applications to Atomic Nuclei*, Nauka, Moscow, 1982.
- [14] H. Pașca, A.V. Andreev, G.G. Adamian, N.V. Antonenko, Y. Kim, *Energy dependence of mass, charge, isotopic, and charge distribution in neutron-induced fission of ^{235}U and ^{239}Pu* , Physical Review C, **93**, 2016.
- [15] Joachim Maruhn, Walter Greiner, *The asymmetric two center shell model*, Zeitschrift für Physik, **251**, 431-457, 1972.
- [16] H. Pașca, A.V. Andreev, G.G. Adamian, N.V. Antonenko, *Fission withing dinuclear system approach*, International Journal of Modern Physics E, 2023.
- [17] H. Pașca, A.V. Andreev, G.G. Adamian, N.V. Antonenko, *Examination of coexistence of symmetric mass and asymmetric charge distributions of fission fragments*, Phys. Rev. C, , **101**, 2020.
- [18] H. Pașca, A.V. Andreev, G.G. Adamian, N.V. Antonenko, *Simultaneous description of charge, mass, total kinetic energy, and neutron multiplicity distributions in fission of Th and U isotopes*, Phys. Rev. C, **104**, 2021.
- [19] J.D. Jackson, *A SCHEMATIC MODEL FOR (p, xn) CROSS SECTIONS IN HEAVY ELEMENTS*, Canadian Journal of Physics, **34**, 56-87, 1956.
- [20] https://en.wikipedia.org/wiki/Frontend_and_backend (accesat la data 26/6/2023)
- [21] [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)) (accesat la data 26/6/2023)
- [22] <https://en.wikipedia.org/wiki/SQL> (accesat la data 26/6/2023)
- [23] https://en.wikipedia.org/wiki/Web_development (accesat la data 26/6/2023)
- [24] [https://en.wikipedia.org/wiki/React_\(software\)](https://en.wikipedia.org/wiki/React_(software)) (accesat la data 26/6/2023)
- [25] <https://www.techtarget.com/searcharchitecture/tip/The-5-essential-HTTP-methods-in-RESTful-API-development> (accesat la data 26/6/2023)
- [26] A.I. Svirikhin et al., *The emission of prompt neutrons from spontaneous fission of ^{252}No and ^{224}Fm* , The European Physical Journal A, **48**, 2012.
- [27] K.H. Schmidt et al., *Relativistic radioactive beams: A new access to nuclear-fission studies*, Nuclear Physics A, **665**, 221-267, 2000.

[28] K.H. Schmidt, J. Benlliure, A.R. Junghans, *Fission of nuclei far from stability*, Nuclear Physics A, **693**, 169-189, 2001.

[29] A. Chatillon et al., *Experimental study of nuclear fission along the thorium isotopic chain: From asymmetric to symmetric fission*, Phys. Rev. C, **99**, 2019.

[30] A. Chatillon et al, *Evidence for a New Compact Symmetric Fission Mode in Light Thorium Isotopes*, Phys. Rev. Lett., **124**, 2020.

DECLARAȚIE PE PROPRIE RĂSPUNDERE

Subsemnatul, Denis-Marian Humă, declar că Lucrarea de licență intitulată Multiplicitatea neutronilor în cazul fisiunii nucleare induse pe care o voi prezenta în cadrul examenului de finalizare a studiilor la Facultatea de Fizică, din cadrul Universității Babeș-Bolyai, în sesiunea iulie 2023, sub îndrumarea Lect. Dr. Horia Pașca, reprezintă o operă personală. Menționez că nu am plagiat o altă lucrare publicată, prezentată public sau un fișier postat pe Internet. Pentru realizarea lucrării am folosit exclusiv bibliografia prezentată și nu am ascuns nici o alta sursă bibliografică sau fișier electronic pe care să le fi folosit.

Prezenta declarație este parte a lucrării și se anexează la aceasta.

Data

27.06.2023

Semnătură

