

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ



Ә. Бүркітбаев атындағы Өнеркәсіптік автоматтандыру және цифрлау институты

«Роботтық техника және автоматиканың техникалық құралдары» кафедрасы

Даниялов Мадияр Саятұлы

«Әр түрлі ядролардың өзара әрекеттесу өнімдерінің кинетикалық энергияларын
есептеуді автоматтандыру»

ДИПЛОМДЫҚ ЖҰМЫС

5B071600 - Аспап жасау мамандығы

Алматы 2021



Ә. Бүркітбаев атындағы Өнеркәсіптік автоматтандыру және цифрлау институты

«Роботтытехника және автоматиканың техникалық құралдары» кафедрасы

ҚОРҒАУҒА РҰҚСАТ
РТжАТҚ кафедра меңгерушісі
техника ғылым кандидаты

Ожикенов К.А.

«7» маусым 2021 ж.

ДИПЛОМДЫҚ ЖҰМЫС

Тақырыбы: «Әр түрлі ядролардың өзара әрекеттесу өнімдерінің кинетикалық энергияларын есептеуді автоматтандыру»

5B071600 – Аспап жасау мамандығы бойынша

Орындады

Даниялов Мадияр Саятұлы

Физика-математика ғылым
кандидаты, қауымдастық
профессоры

Керимкулов Ж.Қ.

«8» маусым 2021 ж.



SATBAYEV
UNIVERSITY

Ө. Бүркітбаев атындағы Өндірістік автоматтандыру және цифрлау институты

«Робототехники және автоматиканың техникалық құралдары» кафедрасы

5B071600 – Аспап жасау

БЕКІТЕМІН

РТжАТҚ кафедра меңгерушісі
техника ғылым кандидаты

Ожикенов К.А.

«23» қантар 2021 ж.

Дипломдық жобаны орындауға

ТАПСЫРМА

Білім алушыға Даниялов Мадияр Саятұлы

Жобаның тақырыбы: Өр түрлі ядролардың өзара әрекеттесу өнімдерінің кинетикалық энергияларын есептеуді автоматтандыру

Университет ректорының 2020 жылғы «24» қараша №2131-б бұйрығымен бекітілген

Аяқталған жұмысты тапсыру мерзімі «25» мамыр 2021 ж.

Дипломдық жобаның бастапқы мәліметтері: Ядролардың өзара әрекеттесулерінің кинетикалық энергияларын есептеуді “Python” бағдарламасында автоматтандыру

Дипломдық жұмыста әзірленуге жататын мәселелер тізімі:

- а) Ядролық физикаға кіріспе
- б) Ядролық реакция
- в) Шашыраудың импульстік диаграммасы
- г) Python тілінде бағдарламалау
- д) Бағдарламаның графикалық интерфейсі

Графикалық материалдардың тізімі (міндетті түрде қажетті сызбалар көрсетілген): 20 слайд

Ұсынылатын негізгі әдебиеттер: 20 әдебиеттер тізімі

КЕСТЕ
Дипломдық жұмысты (жобаны) дайындау

Бөлімдер қарастырылған сұрақтар тізімі	атауы,	Ғылыми жетекшіге көрсету мерзімі	Ескертулер
Негізгі бөлім		23.01 – 15.02.2021 ж.	Орындады
Есептеу бөлімі		23.01 – 15.02.2021 ж.	Орындады
Бағдармалық бөлім		15.03 – 20.04.2021 ж.	Орындады
Зерттеу бөлімі		20.04 – 05.05.2021 ж.	Орындады
Қорытынды бөлім		05.05 – 15.05.2021 ж.	Орындады

Аяқталған дипломдық жұмыс (жобаға) және оған қатысты
бөлімдерінің кеңесшілері мен қалып бақылаушының
ҚОЛТАҢБАЛАРЫ

Бөлімдер атауы	Ғылыми жетекші, кеңесшілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қолтаңба қайылған мерзімі	Қолы
Қалып бақылаушы	Қабдолдина Әсем Оралханқызы PhD доктор	07.06.2021ж.	

Ғылыми жетекшісі



Керимкулов Ж.Қ.

Тапсырманы орындауға алған білім алушы



Даниялов М.С.

Күні

«8» маусым 2021 ж

АНДАТПА

Дипломдық жұмыс ядролық зерттеулердің маңызды бағыттарының бірі - ядролық реакциялардың кинематикасын есептеуді автоматтандыруға, яғни әр түрлі ядролардың өзара әрекеттесу өнімдерінің кинетикалық энергиясын анықтауға арналған. Кинематикалық талдаудың көмегімен түзілген бөлшектердің спектрлерінің сапалық құрамын жоғары дәлдікпен анықтауға болады. Нақтырақ айтқанда, жұмыс бөлшектердің әр түрлі бұрыштағы кинетикалық энергияларын анықтауды жүзеге асырады. Дипломдық жұмыста серпімді және серпімді емес шашырау, ядролық реакциялар барысында қозғалатын күйде және қоздырылған күйде пайда болатын ядролық процестерді анықтайтын импульс диаграммаларын тұрғызуды автоматтандыратын бағдарлама көрсетілген.

АННОТАЦИЯ

Дипломная работа посвящена автоматизации расчетов одного из важных направлений ядерных исследований – кинематики ядерных реакций, то есть, определения кинетических энергий продуктов взаимодействия различных ядер. При помощи кинематического анализа можно с высокой точностью определять качественный состав спектров образованных частиц, в частности, работа предполагает определение кинетических энергий образованных частиц под разными углами. В дипломной работе представлена программа, автоматизирующая построение импульсных диаграмм, определяющих ядерные процессы, происходящие в возбужденном состоянии при ядерных реакциях.

ABSTRACT

The diploma work is devoted to the automation of calculations of one of the important areas of nuclear research - the kinematics of nuclear reactions, namely, the determination of the kinetic energies of the interaction products of various nuclei products. With the help of kinematic analysis, it is possible to determine with high accuracy the qualitative composition of the spectra of formed particles. The work involves determination of the kinetic energies of the formed particles at different angles. In the project, presented a program which automates the construction of pulse diagrams that determine the nuclear processes occurring in an excited state during nuclear reactions.

МАЗМҰНЫ

КІРІСПЕ	9
1 Технологиялық бөлім	10
1.1 Ядролық физика	10
1.2 Ядролық реакция	10
1.3 Реакцияға түсетін бөлшектер	10
1.4 Үдеткіш. У-150М изохронды циклотрон	11
2 Зерттеу бөлімі	12
2.1 Энергия және импульстің сақталу заңдары	12
2.2 Бөлшектердің ядролардағы шашырау процестерінің кинематикасы	13
2.3 Зертханалық жүйеде шашырау процесінің сипаттамасы	13
2.4 Бөлшектердің серпімді шашырауы	14
2.5 Шашыраудың импульстік диаграммасы	15
3 Бағдарламалау бөлімі	20
3.1 Python тілінде бағдарламалау	20
3.2 Бағдарламаның графикалық интерфейсі	21
3.3 Әр түрлі спектрлерді талдау	24
Қорытынды	
Пайдаланылған әдебиеттер тізімі	
Қосымша А	

КІРІСПЕ

Ядролық зерттеулердің маңыздылығы өте зор. Атомдық және термоядролық энергетика, электронды ядролық қондырғылар, ядролық астрофизика, ядролық медицина, ядролық талдау әдістері - ядролық физиканың қолданбалы бағыттарының аз ғана бөлігі. Осы салалардағы барлық жетістіктер ядролық өзара әрекеттесуді түбегейлі зерттемей мүмкін болмас еді. Қазіргі таңда компьютерлік индустрия, микроэлектроника және бағдарламалық қамтамасыз ету қарқынды дамып келеді. Бұл құралдар халық шаруашылығына және адамның ғылыми қызметіне көбірек енгізілуде. Барлық өлшемдер мен есептеулер толығымен автоматтандырылған. Жұмыстың мақсаты ядролық зерттеулердің маңызды бағыттарының бірі - ядролық реакциялар кинематикасын есептеуді автоматтандыру. Нақтылап айтқанда, Python бағдарламалау тілінің көмгімен өзара әрекеттесетін бөлшектердің кинетикалық энергиясын анықтауды автоматтандыру.

Кинематикалық анализдің көмегімен түзілген бөлшектер спектрлерінің сапалық құрамын жоғары дәлдікпен анықтауға болады. Есептеулерді барынша жылдамдату мақсатында және салыстыру үшін Java бағдарламалау тілі де қолданылды. Үдеткіштегі зерттеу нәтижесінен алынған мәліметтер бағдарламаға енгізілген соң, есептеулер орындалады. Қорытындысы жеке терезеде ашылып, шашырау диаграммасы график түрінде көрсетіледі.

Бағдарламаның графикалық интерфейсі QT-Designer кроссплатформалық ортасында PyQt5 құралдар жиынтығы көмегімен жасалған. PyQt Python тіліндегі ең қуатты және танымал GUI интерфейстерінің кітапханасы болып табылады.

Python және Java тілдерінде оптимизацияланған кодтарды есептеу уақыты салыстырылады. Екі тілде де есептеу уақыты айырмашылығы анықталады.

1 Технологиялық бөлім

1.1 Ядролық физика

Ядролық физика, атом ядросының құрылымын, радиоактивті ыдырау процестерін және ядролық реакциялардың механизмін зерттеуге арналған физика бөлімі. Бұл терминге жалпы мағына бере отырып, қарапайым бөлшектер физикасы ядролық физика деп аталады. Кейде ядролық физиканың салалары технологияның дербес саласы зерттеу салалары ретінде қарастырыла береді, мысалы, үдеткіш технологиясы.

1.2 Ядролық реакция

Ядролық реакция - ядро құрылымын және оның қасиеттерін зерттеудің негізгі әдісі. Алайда, олардың физикадан тыс рөлі де зор: ауыр ядролардың бөліну реакциялары және ең жеңіл ядролардың бірігуі ядролық энергетиканың негізінде жатыр. Ядролық реакциялар нейтрондардың, мезондардың және басқа тұрақсыз бөлшектерді алу көзі ретінде қолданылады. Ядролық реакциялардың көмегімен ғылымның, техниканың және медицинаның барлық салаларында қолданылатын мыңнан астам радиоактивті нуклидтер алынады.

Ядролық реакциялар, элементар бөлшектермен, g-кванттармен немесе бір-бірімен әрекеттесу кезіндегі атом ядроларының түрленуі. Ядролық реакцияны жүзеге асыру үшін бөлшектерді, екі ядроны: ядро мен нуклонды $\sim 10^{-13}$ см қашықтыққа жақындату керек. Түсетін оң зарядталған бөлшектердің энергиясы ядролардың кулондық тосқауылының биіктігінен (жалғыз зарядталған бөлшектер үшін ~ 10 МэВ) үлкен болуы керек. Бұл жағдайда ядролық реакция нысандарды үдетілген бөлшектердің сәулелерімен атқылау (бомбардировка) арқылы жүзеге асырылады. Теріс зарядталған және бейтарап бөлшектер үшін кулондық тосқауыл болмайды, ал ядролық реакциялар түсетін бөлшектердің жылу энергиясымен де жүруі мүмкін.

Ядролық реакциялар $A(a, bcd)B$ түрінде жазылады, мұндағы A - нысана ядро, a - атқылаушы (бомбардирующая) бөлшек, b, c, d - шығарылатын бөлшектер, B - қалдық ядро.

1.3 Реакцияға түсетін бөлшектер

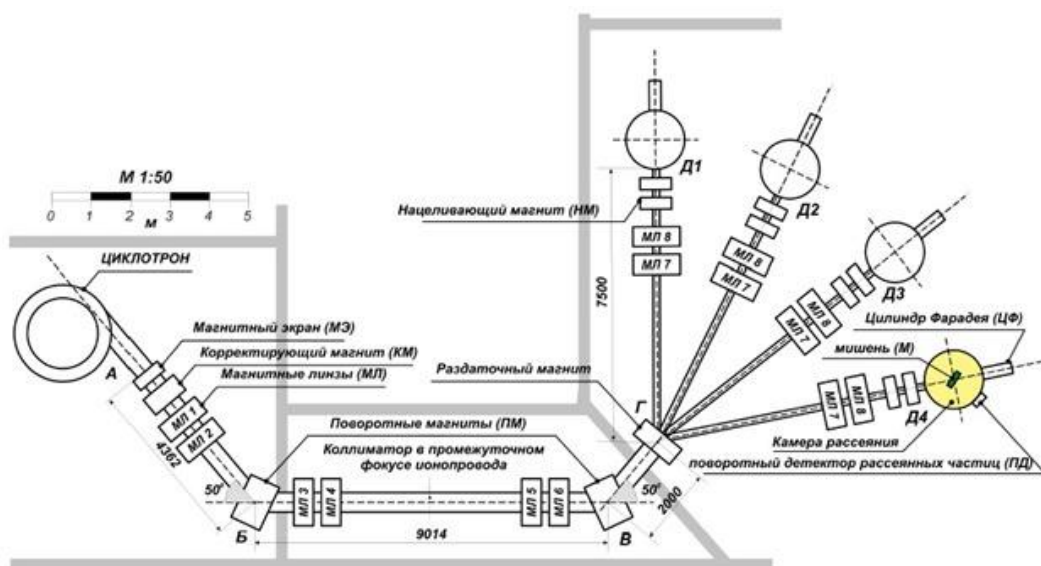
Тритон, сутектің радиоактивті изотопының ядросы - тритий; ${}^3\text{H}$, t арқылы белгіленеді, құрамында 3 нуклон (1 протон және 2 нейтрон) бар. Массасы 3.01646 атомдық масса бірлігі, спин $1/2$ ($\hbar$ бірліктерінде), магниттік момент 2.979 ядролық магнетон. Нуклондардың орташа байланыс энергиясы 2,78 МэВ құрайды.

Дейтрон, массасының саны 2 сутегі - дейтерий - изотопы атомының ядросы. ${}^2\text{H}$, D немесе d деп белгіленеді. Дейтрон 1 протон мен 1 нейтроннан тұрады. Дейтрон массасы 2.014102 атомдық масса бірлігі, нуклонның байланыс

энергиясы 2.22452 ± 0.00010 МэВ, спин 1 ($\hbar$ бірліктерінде), магниттік моменті 0.857411 ± 0.000019 ядролық магнетондар, электрлік квадрупольдік моменті $(2.738 \pm 0.014) 10^{-27} \text{см}^2$.

1.4 Үдеткіш. У-150М изохронды циклотрон

Үдеткіш протондарды 30 МэВ-қа дейін, 2H 25 МэВ-қа дейін, тритондарды 60 МэВ дейін және альфа-бөлшектерді 50 МэВ-қа дейін үдетуге мүмкіндік береді. Қондырғыдағы зарядталған бөлшектер камераның орталық бөлігінде орналасқан көзде тиісті газ(сутегі, дейтерий, гелий-3, гелий-4) берілген кезде пайда болады. Олардың үдеуі бөлшектердің дуандтар арасынан ұшқан сәтінде 1,5м магниттің полюсаралық кеңістігінде болады. Бөлшектер үдеуінің жұмыс параметрлерін белгілеу кезінде ион көзінің жұмыс режиміне, оның жұмыс циклына, ток импульсінің микроқұрылымына, сондай-ақ сәулені нысанға бағыттау сапасына ерекше назар аударылады. Сәуленің кеңістіктік және уақыттық сипаттамаларын оңтайландыру әртүрлі шудың деңгейін, электронды жабдықтың біркелкі емес жүктелуін айтарлықтай төмендетуге мүмкіндік берді. Үдемелі иондардың сәулесін циклотрон камерасынан сәуленің шығуынан 24 м қашықтықта орналасқан шашырау камерасына тасымалдау схемасы көрсетілген.



Сурет 1.1 - Үдемелі иондардың сәулесін циклотрон камерасынан шашырау камерасына тасымалдау схемасы

Оған квадрупольды линзалар жүйесі, екі айналмалы, жайылатын, екі бағыттаушы магнит және коллиматорларлар жүйесі кіреді. Осы қондырғылар нысаналау және түзету элементтерімен бірге $0,4^\circ$ аспайтын бұрыштық саңылауы бар және диаметрі 3 мм зарядталған бөлшектердің сәулесін қамтамасыз етеді.

2 Зерттеу бөлімі

2.1 Энергия және импульстің сақталу заңдары

Шағын мөлшерде (шамамен 10^{-12} см) конденсацияланған заттағы (шамамен 10^{-8} см) атом ядроларының алшақ орналасуы, кіші радиустағы ядролық күштердің және мәні төменгі химиялық байланыстың әсерінен өзара әрекеттесетін екі ядролық бөлшектердің жүйесін жабық (оқшауланған) деп санауға мүмкіндік береді. Оқшауланған жүйеде бөлшектердің толық энергиясы мен импульсі сақталады.

Процесс үшін энергияның сақталу заңын келесі түрде жазуға болады

$$E_1 = E_2 \quad (2.1)$$

немесе

$$E_{01} + T_1 = E_{02} + T_2 \quad (2.2)$$

мұндағы E_{01} және E_0 - тыныштық энергиясы, ал T_1 және T_2 - бастапқы және соңғы бөлшектердің кинетикалық энергиясы:

$$E_{01} = M_A c^2 + M_a c^2, \quad (2.3)$$

$$E_{02} = M_B c^2 + m_b c^2, \quad (2.4)$$

$$T_1 = T_A + T_a, \quad (2.5)$$

$$T_2 = T_B + T_b. \quad (2.6)$$

Жалпы жағдайда $E_{01} \neq E_{02}$. $E_{01} - E_0$ айырмашылығы реакция энергиясы деп аталады және Q әрпімен белгіленеді:

$$Q = E_{01} - E_{02} = T_2 - T_1 \quad (2.7)$$

Егер $Q > 0$ болса, онда реакция тыныштық энергиясын азайту арқылы кинетикалық энергияның шығарылуымен бірге жүреді және экзоэнергетикалық деп аталады. Экзоэнергетикалық реакция атқылаушы бөлшектің кез-келген кинетикалық энергиясында жүруі мүмкін (энергия зарядталған бөлшектің ядроның кулондық кедергісін жеңу үшін жеткілікті болса).

Егер $Q < 0$ болса, онда реакция кинетикалық энергияның төмендеуіне арқылы демалу энергиясының жоғарылауымен бірге жүреді және эндоэнергетикалық деп аталады. Эндоэнергетикалық реакция атқылаушы бөлшектің кинетикалық энергиясы жеткілікті болса ғана орын алады.

$$Q < 0 \quad (2.8)$$

болса,

$$T_1 = |Q| + T_2 > |Q| \quad (2.9)$$

2.2 Бөлшектердің ядролардағы шашырау процестерінің кинематикасы

Үдетілген бомбалаушы бөлшектер нысана бөлшектерге шашырайды, өйткені олардың арасында өзара әрекеттесу күштері бар; құбылыстың көрінісі ең алдымен осы күштердің табиғатына байланысты. T_i - өзара әрекеттесуге дейінгі жүйенің толық кинетикалық энергиясы, ал T_f - өзара әрекеттесуден кейінгі толық кинетикалық энергия болсын.

Егер $T_f = T_i$ болса, онда серпімді шашырау пайда болады деп айтылады; және бомбаланған(шашыраған) бөлшек ауытқып, энергиясын жоғалтады.

Егер $T_f \neq T_i$ болса, онда серпімді емес шашырау пайда болады деп айтылады. Химиялық реакциялар сияқты, $Q = T_f - T_i$ өзара әрекеттесу кезінде энергия тепе-теңдігін есептеуге болады. Егер шашырау процесінде бөлшектердің түрі өзгермесе, онда мұндай шашырау серпімді емес деп аталады.

Берілген анықтамалардан кейін біз m_1 массасы бар үдетілген бөлшектердің серпімді шашырау процесін және m_2 массасы бар нысана бөлшектеріндегі жылдамдықты және өзара әрекеттесуге дейінгі нөлдік бастапқы жылдамдықты қарастырамыз. Біз өзара әрекеттесу сипатына да, оның қарқындылығын сипаттайтын формулаларға да қатысты гипотезалар жасай алмаймыз; бұл жағдайда шашыраңқы бөлшектердің бұрыштық таралуын есептей алмаймыз.

Серпімді шашырау жағдайында нысана бөлшектердің ішкі күйі өзгермейді: бұл механикада қарастырылатын серпімді соқтығысудың әдеттегі процесі. m_1 және m_2 массалары тұрақты (бөлшектердің жылдамдығына тәуелді емес) .

Бақылаушымен байланысты координаттар жүйесі зертханалық жүйе деп аталады. Құрылғылар өзара әрекеттесу нәтижелерін дәл осы жүйеде тіркейді, онда біз «бомбалаушы» (немесе ұшатын) бөлшек және «нысанды бөлшек» туралы айтуға болады.

Масса центрінің жүйесі (МЦ жүйесі) зертханалық жүйеге қатысты біркелкі және түзу қозғалатын бөлшектердің масса центріне ие. Бұл жүйеде бөлшектер массаның центріне бірдей мөлшерде, бірақ қарама-қарсы бағытталған импульстармен жақындайды; жүйенің толық импульсі нөлге тең: $\sum p' = 0$ ("штрих" МЦ жүйесіндегі физикалық шаманы білдіреді). Шашырау процестерін сипаттайтын теңдеулер зертханалық жүйеге қарағанда МЦ жүйесінде әлдеқайда қарапайым, сондықтан көбінесе МЦ жүйесінде барлық есептеулерді жүргізген дұрыс.

2.3 Зертханалық жүйеде шашырау процесінің сипаттамасы

Масса орталығы бомбалаушы(шашыраушы) бөлшекпен бірдей бағытта v_c жылдамдығымен қозғалады:

$$(m_1 + m_2) = m_1 v_c, \quad v_c = v \frac{m_1}{m_1 + m_2} = v \frac{m_0}{m_2} \quad (2.10)$$

Мұнда m_0 – бөлшектердің келтірілген массасы:

$$\frac{1}{m_0} = \frac{1}{m_1} + \frac{1}{m_2}, \quad m_0 = \frac{m_1 m_2}{m_1 + m_2} \quad (2.11)$$

Кванттық механика келтірілген массаның физикалық мағынасын былай нақтылайды: оның мәні массаның жартысы мен соқтығысатын екі бөлшектің ең жеңілінің толық массасы арасында болады.

Өзара әрекеттесу нәтижесінде бомбалаушы(шашыраушы) бөлшек θ бұрышына ауытқиды, ал нысана бөлшек ϕ бұрышында бастапқы шашыраушы бөлшектің импульс бағытына кері қайтарымға ұшырайды. Әсер және кері әсер теңдігі заңына сәйкес жүйенің толық импульсі өзгермейді: масса орталығы бұрынғы бағытта V_c жылдамдығымен қозғалуды жалғастырады.

Энергия мен Импульстің сақталу заңдарын жаза отырып, біз келесі кинематикалық теңдеулерді аламыз:

$$\frac{1}{2} m_1 v^2 = \frac{1}{2} m_1 v_1^2 = \frac{1}{2} m_2 v_2^2, \quad (2.12)$$

$$m_1 v = m_1 v_1 \cos \theta + m_2 v_2 \cos \phi, \quad (2.13)$$

$$m_1 v_1 \sin \theta - m_2 v_2 \sin \phi = 0. \quad (2.14)$$

Бізде үш теңдеу және төрт белгісіз бар: ϕ, θ, v_1, v_2 . Егер олардың біреуін орнатсақ, мысалы θ , онда қалған барлық мәндерді анықтауға болады. θ бұрышына шашыраған бөлшектердің саны және олардың энергетикалық спектрі өзара әрекеттесу түріне байланысты болады. Жоғарыдағы үш теңдеу шашырау бұрыштарының арасындағы қатынастарды береді:

$$\operatorname{tg} \theta = \frac{\sin 2\phi}{(m_1/m_2) - \cos 2\phi} \quad (2.15)$$

2.4 Бөлшектердің серпімді шашырауы

Серпімді шашырау - екі бөлшектің өзара әрекеттесу процесінде жалпы кинетикалық энергиялары сақталып, тек бөлшектер арасында қайта бөлінеді, ал бөлшектердің қозғалыс бағыттары өзгереді.

Ядролық физикада серпімді шашырау пайда болатын күштер ретінде кулон және ядролық күштер (күшті ядролық өзара әрекеттесу), ал соңғы уақыттарда әлсіз күштер де қарастырылады. Шағын энергиялардың зарядталған бөлшектері кулондық шашырауға, үлкен энергиялардың зарядталған бөлшектері және нейтрондар - ядролық шашырауға ұшырайды. Кулондық немесе ядролық шашырау р-соққы параметрімен немесе l орбитальды санымен анықталады. Белгілі бір жылдамдықпен v ұшатын ze зарядталған бөлшегі басқа бір

зарядталған бөлшекке Ze (соққы параметрі кіші $p=p_1$) жақын ұшса шашырау бұрышы ϕ_1 үлкен болады, ал басқа алыс қашықтықта ұшатын зарядталған бөлшектің (соққы параметрі үлкен $p=p_2$) шашырау бұрышы ϕ_2 салыстырмалы түрде кіші болады.

Ұзақ әсер ететін күштер (кулондық немесе гравитациялық) үшін соққы параметрі p өте үлкен болуы мүмкін. Өзара тиімді әрекеттесу үшін қысқа әсер ететін күштер (ядролық) жағдайында p әсер ету параметрі a ядролық күштерінің әсер ету радиусынан аз болуы керек (бөлшек күштердің әсер ету аймағында ұшуы керек). Сонымен қатар, кванттық механикаға сәйкес, p импульсі бар бөлшек үшін

$$p < a \quad (2.16)$$

теңсіздігін қанағаттандыратын барлық мәндер емес,

$$\begin{aligned} p_l &= (\hbar/p) \sqrt{l(l+1)} = \\ &= \lambda \sqrt{l(l+1)} < a, \end{aligned} \quad (2.17)$$

шартымен байланысты кейбір дискретті мәндер жиынтығы ғана болады. Мында, $l = 0, 1, 2, \dots$. Бұл шарт l бөлшегінің қозғалыс моменті үшін классикалық және квантомеханикалық өрнектерді салыстыру нәтижесінде алынады:

$$||| = \rho p = \hbar \sqrt{l(l+1)} \quad (2.18)$$

Бөлшек неғұрлым баяу болса (p импульсі аз болған сайын), l орбитальды саны және соққы параметрлері соғұрлым аз мүмкін мәндерді ($p < a$) қабылдай алатын болады. Мысалы, нейтронның $T < 10$ МэВ энергиясымен протонға шашырауы тек $l = 0$ және $p = 0$ жағдайында ғана орын алуы мүмкін (классикалық механикадағы орталық соққының аналогы). Керісінше, жылдам бөлшектер p үлкен жағдайда (1.1) шартын l және p әр түрлі мәндерінде орындауға болады, бұл жағдайда әр l мәні өз шашыраңқы бөлшектердің бұрыштық таралу заңын анықтайды.

2.5 Шашыраудың импульстік диаграммасы

Егер шашыраңқы бөлшектің қозғалыс бағыты белгілі болса (тәжірибеден немесе соққының алдын ала берілген параметрінен p және күштердің әрекет ету заңынан), онда белгілі жылдамдық мәндері мен атқылаушы бөлшектің қозғалыс бағыты бойынша шашырағаннан кейін екінші бөлшектің жылдамдығы мен қозғалыс бағытын анықтаудың қарапайым геометриялық әдісі бар. Бұл әдіс импульстік диаграмма деп аталады. Төменде релятивистік емес жағдай үшін импульстік диаграмманың құрылысы қарастырылған.

Массасы M_1 және жылдамдығы v болатын бөлшек массасы M_2 болатын қозғалмайтын бөлшекпен соқтығысады делік. Есептеулер кез-келген масса қатынасы үшін жарамды, бірақ сенімділік үшін біз $M_1 < M_2$ деп санаймыз.

$$v_{\text{н.о}} = \frac{M_1}{M_1 + M_2} v \quad (2.19)$$

екі бөлшектің инерция ортасының жылдамдығы. Инерция ортасының жылдамдығындағы M_2 бөлшегінің жылдамдығы:

$$v'_{M_2} = -v_{\text{н.о}} = \frac{-M_1}{M_1 + M_2} v \quad (2.20)$$

және импульсі:

$$p'_{M_2} = -\frac{M_2}{M_1 + M_2} p_{M_1} \quad (2.21)$$

ал M_1 бөлшегінің импульсі:

$$p'_{M_1} = p'_{M_2} = \frac{M_2}{M_1 + M_2} p_{M_1} \quad (2.22)$$

Зертханалық координаттар жүйесінде АВ кесіндісі белгілі бір масштабта шашырауға дейінгі M_1 бөлшегінің бастапқы импульсін көрсетсін делік:

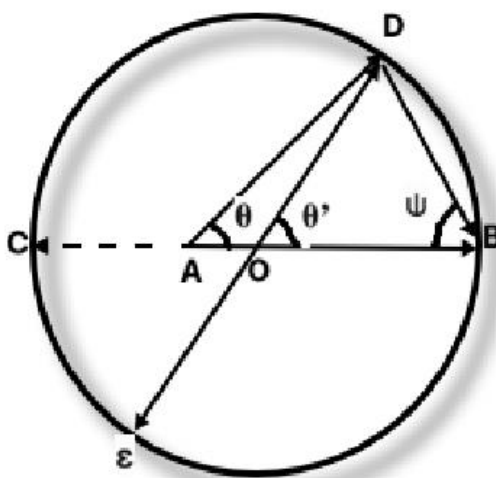
$$p_{M_1} = AB \quad (2.23)$$

Оны $AO:OB = M_1:M_2$ массалар қатынасындағы O нүктесімен бөлеміз.

Онда:

$$OB = \frac{M_2}{M_1 + M_2} p_{M_1} = p'_{M_1}, \quad (2.24)$$

$$\text{ал } p'_{M_2} = -p'_{M_1} = OC.$$



Сурет 2.1 - Импульстік диаграмма тұрғызу

Импульстің сақталу заңына сәйкес жүйедегі екі бөлшектің де импульстері соқтығысқаннан кейін инерция орталықтарының жүйесінде шама жағынан тең және бағыты жағынан қарама-қарсы болуы керек. Сонымен қатар, серпімді соқтығысу үшін бөлшектердің жалпы кинетикалық энергиясы сақталған кезде импульстің абсолютті мәні де өзгермейді.

Осылайша, и.о.ж - дағы бөлшектердің шашырау процесінің сипаттамасы $\vec{p}'_{M_1} = \vec{OB}$ және $\vec{p}'_{M_2} = \vec{OC}$ импульстарының белгілі бір θ' бұрышына бұрылады, шашырағаннан кейін M_1 және M_2 бөлшектердің импульстары $\vec{p}'_{M_1} = \vec{OD}$ және $\vec{p}'_{M_2} = \vec{OE}$ кескіндерімен бейнеленеді (айырмашылық үшін шашыраудан кейінгі бөлшектің импульсі $\vec{p}$ белгісімен белгіленеді).

Зертханалық координаттар жүйесі $\vec{v}$ кері өту үшін и.о.ж-нің з.к.ж-ға қатысты жылдамдығын, $v_{ц.и} = M_1 v / (M_1 + M_2)$, ескеру қажет. Бұл қозғалысқа екі бөлшек те қатысады, сондықтан олар қосымша ауыспалы импульстарға ие болады:

$$(p_{M_1})_{\text{ауыс}} = M_1 v_{ц.и} = \frac{M_1^2}{M_1 + M_2} v = \frac{M_1}{M_1 + M_2} p_{M_1}, \quad (2.25)$$

$$(p_{M_2})_{\text{ауыс}} = M_2 v_{ц.и} = \frac{M_1 M_2}{M_1 + M_2} v = \frac{M_2}{M_1 + M_2} p_{M_1}.$$

$(p_{M_1})_{\text{ауыс}}$ және $(p_{M_2})_{\text{ауыс}}$ импульстарына АО және ОВ кесінділері сәйкес келеді. Осылайша, шашырағаннан кейін и.о.ж-да M_1 бөлшегінің импульсі $\vec{p}'_{M_1} = \vec{OD}$ кесіндісімен, ал оның қосымша импульсі - АО кесіндісімен өлшенеді. OD және АО векторларын қосу арқылы шашыраудан кейінгі $\vec{p}'_{M_1}$ бөлшегінің з.к.ж-дағы импульсін бейнелейтін $\vec{AD} = \vec{OD} + \vec{AO}$ векторын аламыз. Дәл солай $\vec{p}'_{M_2} = \vec{OE}$ және $(p_{M_2})_{\text{ауыс}} = \vec{OB}$ векторларын қосу арқылы шашыраудан кейінгі з.к.ж-дағы бөлшектің импульсін бейнелейтін $\vec{DB} = \vec{OB} - \vec{OD} = \vec{OB} + \vec{OE}$ векторын аламыз. AD және DB векторлары M_1 бөлшегінің $\vec{p}_{M_1} = \vec{AB}$ бастапқы импульсімен бірге векторлық үшбұрышты құрайды, сондықтан

$$\vec{p}_{M_1} = \vec{p}'_{M_1} + \vec{p}'_{M_2} \quad (2.26)$$

Сонымен, шашыраңқы бөлшектер мен қайтару ядросының импульстарын алу үшін келесі амалдарды жасау керек:

1. $M_1:AB = p_{M_1}$ бөлшегінің бастапқы импульсіне тең AB кесіндісін аламыз

2. Оны $AO/OB = M_1/M_2$ массалар қатынасындағы O нүктесімен бөліңіз және O нүктесінен B нүктесі арқылы өтетін шеңберді сызыңыз.

3. А нүктесінен θ шашырау бұрышымен D нүктесіндегі шеңбермен қиылысқанға дейін түзу сызыңыз және D нүктесін B нүктесімен қосыңыз.

4. D нүктесінен өтетін диаметрді құрыңыз.

Сонда AD кесіндісі шашыраңқы бөлшектің импульсін бейнелейді; $\angle BAD = \theta$ — M_1 бөлшегінің шашырау бұрышы; DB кесіндісі — қайтару ядросының импульсі; $\angle DBA = \psi$ — M_2 қайтару ядросының шашырау бұрышы; $\angle DOB = \theta'$ — и.ц.ж-дағы M_1 бөлшегінің шашырау бұрышы; OB және OD и.ц.ж-дағы шашырауға дейінгі және шашыраудан кейінгі M_1 бөлшегінің импульстері; OC және OE — шашырауға дейін және одан кейінгі и.ц.ж-дағы M_2 бөлшегінің импульстары.

Егер θ шашырау бұрышы белгісіз болса және диаграмма оны шашыраңқы бөлшектер мен қайтару ядросы үшін импульстар мәндерімен бірге табу үшін салынса, онда 3 және 4 тармақтарының орындары ауысады. Бұл жағдайда D нүктесі M_1 бөлшегінің и.ц.ж-дағы шашырау бағытын анықтайтын θ' бұрышымен жүргізілген диаметрдің шеңберімен қиылысу нәтижесінде алынады.

1. И.ц.ж-дағы екі бөлшектің толық кинетикалық энергиясы:

$$T' = \frac{M_1 M_2}{M_1 + M_2} \frac{v^2}{2} = \frac{M_2}{M_1 + M_2} T = \frac{\mu v^2}{2} \quad (2.27)$$

Бұл жерде $\mu = M_1 M_2 / (M_1 + M_2)$ — M_1 және M_2 бөлшектерінің келтірілген массасы

2. Екі бөлшектің ауыспалы қозғалысының кинетикалық жнергиясы

$T_{\text{и.ц}} = (M_1 + M_2) v_{\text{и.ц}}^2 / 2$, немее $v_{\text{и.ц}} = M_1 v / (M_1 + M_2)$ болғандықтан

$$T_{\text{и.ц}} = \frac{M_1^2}{M_1 + M_2} \frac{v^2}{2} = \frac{M_1}{M_1 + M_2} T. \quad (2.28)$$

(2.27) және (2.28) теңдеулерін өзара салыстыра отырып $T' + T_{\text{и.ц}} = T$ теңдеуін аламыз, мұндағы T — з.к.ж-дағы M_1 бөлшегінің бастапқы кинетикалық энергиясы ($T = M_1 v^2 / 2$).

3. З.к.ж-дағы соққындан кейінгі M_1 және M_2 бөлшектерінің кинетикалық энергиялары:

$$\begin{aligned} \check{T}_{M_1} &= \frac{M_1^2 + M_2^2 - 2M_1 M_2 \cos 2\psi}{(M_1 + M_2)^2} T & (\check{T}_{M_1})_{\min} &= \left(\frac{M_1 - M_2}{M_1 + M_2} \right)^2 T, \\ & & (\check{T}_{M_1})_{\max} &= T; \end{aligned} \quad (2.29)$$

$$\begin{aligned} \check{T}_{M_2} &= \frac{2M_1 M_2}{(M_1 + M_2)^2} (1 + \cos 2\psi) T & (\check{T}_{M_2})_{\max} &= \left(\frac{4M_1 M_2}{(M_1 + M_2)^2} \right) T, \\ & & (\check{T}_{M_2})_{\min} &= 0. \end{aligned}$$

серпімді соқтығысу үшін сақталу заңына сәйкес

$$\check{T}_{M_1} + \check{T}_{M_2} = T \quad (2.30)$$

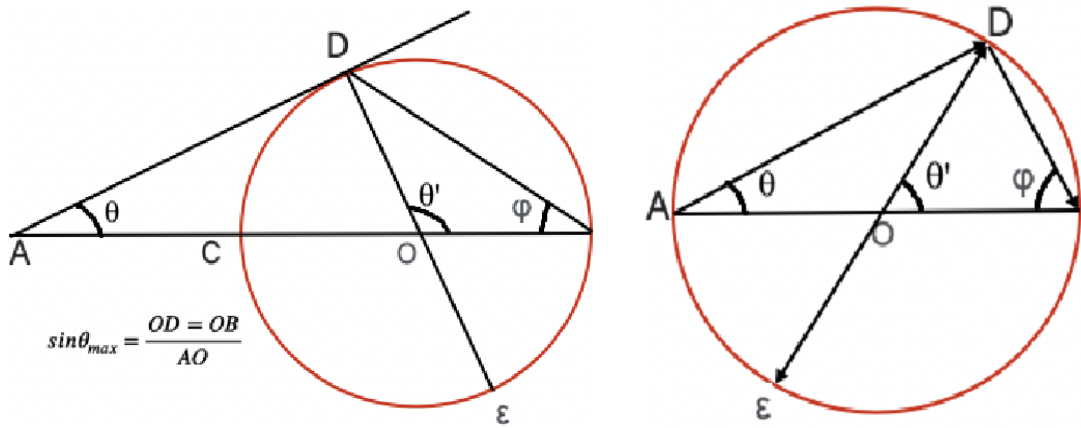
4. З.к.ж—дағы M_1 және M_2 бөлшектерінің шашырау бұрыштары

$$tg\theta = \frac{\sin\theta'}{M_1/M_2 - \cos 2\psi'}.$$

5. $M_1 > M_2$ (2.2-сурет) жағдайына з.к.ж-дағы M_1 бөлшегінің бұрышының ең жоғары мәні тиесілі және

$$\sin\theta_{max} = M_2/M_1 \quad (2.31)$$

шартымен анықталады.



Сурет 2.2 – Импульстік диаграмма

Импульстік диаграмма серпімді соқтығысу үшін әртүрлі тапсырмаларды ыңғайлы және жылдам шешуге мүмкіндік береді. Оның көмегімен массалары белгілі бөлшектердің кез-келген шашырау бұрышы үшін шашыраудан кейінгі олардың импульстері мен энергияларын табуға болады; шашырау бұрыштары θ және ψ және бір бөлшектің массасы белгілі болса, онда екінші бөлшектің массасын табуға болады. $M_1 = M_2$ тең массасы болған жағдайда импульстік диаграмма қарапайым көрініске ие болады. Ондай жағдайда:

$$\begin{aligned} p_{M_1} &= AB; p_{M_2} = 0; p'_{M_1} = p_{M_1}/2 = OB; p'_{M_2} = OA; \\ \mu &= M_1/2 = M_2/2; T'_{M_1} = (p'_{M_1})^2/2M_1 = T/4; \\ T'_{M_2} &= (p'_{M_2})^2/2M_2 = T/4; T'_{M_1} + T'_{M_2} = T/2; \\ \theta' &= 2\theta; \theta + \psi = \pi/2; ctg\theta ctg\psi = 1; \\ \check{p}_{M_1} &= AD = AB\cos\theta = p_{M_1}\cos\theta = p_{M_1}\sin\psi; \\ \check{p}_{M_2} &= DB = AB\cos\psi = p_{M_1}\cos\psi = p_{M_1}\sin\theta; \\ \check{T}_{M_1} &= T_{M_1}\cos^2\theta; \check{T}_{M_2} = T_{M_1}\cos^2\psi = T_{M_1}\sin^2\theta; \\ (\check{T}_{M_1})_{max} &= T_{M_1}; (\check{T}_{M_1})_{min} = 0; (\check{T}_{M_2})_{max} = T_{M_1}; \\ (\check{T}_{M_1})_{min} &= 0. \end{aligned} \quad (2.32)$$

3 Бағдарламалау бөлімі

3.1 Python тілінде бағдарламалау

Бағдарламаның негізгі бөлімі Python бағдарламалау тілінде жазылған. Есептеулерді барынша жылдамдату мақсатында және салыстыру үшін Java бағдарламалау тілі де қолданылды. Алдын ала берілген мәліметтер жеке файлда сақталып, сол файлдан ақпараттарды оқу арқылы есептеулер орындалады. Ол ақпараттарға: реакцияға дейінгі және реакциядан кейінгі бөлшектердің массалары, массалық дефектері мен үдетуге ұшырайтын бөлшектің соққыға дейінгі энергиясы жатады. Бұл ақпараттар кез-келген мәтіндік форматта сақталып, бағдарламада арнайы шақыру командаларымен оқылады. Мысалы:

Соққыға дейінгі бірінші бөлшек массасы(M_1): 2.0141022

Соққыға дейінгі екінші бөлшек массасы(M_2): 11.0093052

Соққыдан кейінгі бірінші бөлшек массасы(M_1'): 4.002603

Соққыдан кейінгі екінші бөлшек массасы(M_2'): 9.0121829

Соққыға дейінгі 1ші бөлшек массалық дефекті: 13.13590

Соққыға дейінгі 2нші бөлшек масса дефектісі: 8.6677000

Соққыдан кейінгі 1ші бөлшек масса дефектісі: 2.4249200

Соққыдан кейінгі 2ші бөлшек масса дефектісі: 11.348100

Соққыға дейінгі бірінші бөлшек энергиясы: 25.00000000

мәліметтер «data.txt»-файлында сақталған. Мәліметтер

with open("data.txt", "r") as f:

readdata = f.read(), командасымен оқылады, және кейінгі операцияларда қолданылады. Клавиатурадан бөлшектің шашырау бұрыштары мен қозу деңгейі беріледі. Есептеулер орындалғаннан соң соққыдан кейінгі кинетикалық энергия шамалары шашырау бұрыштары бойынша жаңа файлға шығарылады. Мысал ретінде соққыға дейінгі бірінші бөлшек ретінде дейтерий, екінші бөлшек ретінде бор-11 алынып, соққыдан кейінгі бірінші бөлшек гелий-2, екінші бөлшек бериллий-9 пайда болды. Бұл жердегі массаның өлшем бірлігі - массаның атомдық бірлігі(м.а.б.). 1м.а.б. = $1,66053 \cdot 10^{-27}$ кг, ^{12}C көміртек нуклидінің 1/12 массасына тең.

Есептеу бұрышы 0-ден 40-қа дейін, қозу деңгейі 2-ге тең деп алып, реакция энергиясы, Q, 6МэВ-қа тең екенін табамыз. Есептеу аяқталған соң нәтижесі «Kinenergy.txt»-файлына сақталады.

Бағдарламаның математикалық бөлімі Python тілінің Math кітапханасы көмегімен орындалды. Айнымалының алдына math.cos, math.sin немесе math.pi амалдарын беру арқылы косинус, синус немесе пи санын ала аламыз. Мысалы: A=90 болсын, math.cos(A) мәні 0-ге тең болады, себебі $\cos 90=0$.

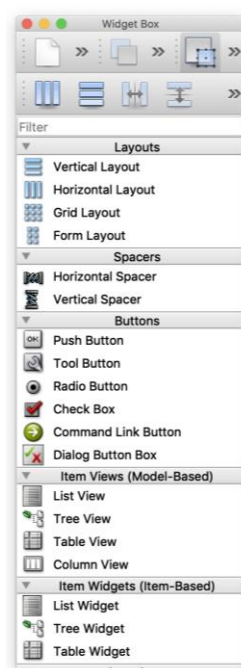
Бағдарламаның графикалық интерфейсі QT-Designer кроссплатформалық ортасында PyQt5 құралдар жиынтығы көмегімен жасалған. PyQt Python тіліндегі ең қуатты және танымал GUI интерфейстерінің кітапханасы болып табылады.

1	Q=6.0 Қозу деңгейі =2.0			
2	Бұрыш	Энергия	Бұрыш	Энергия
3	-----			
4	0	29.42 MeV	21	28.45 MeV
5	1	29.41 MeV	22	28.36 MeV
6	2	29.41 MeV	23	28.26 MeV
7	3	29.4 MeV	24	28.16 MeV
8	4	29.38 MeV	25	28.06 MeV
9	5	29.36 MeV	26	27.95 MeV
10	6	29.33 MeV	27	27.84 MeV
11	7	29.31 MeV	28	27.73 MeV
12	8	29.27 MeV	29	27.62 MeV
13	9	29.23 MeV	30	27.5 MeV
14	10	29.19 MeV	31	27.37 MeV
15	11	29.15 MeV	32	27.25 MeV
16	12	29.09 MeV	33	27.12 MeV
17	13	29.04 MeV	34	26.99 MeV
18	14	28.98 MeV	35	26.85 MeV
19	15	28.92 MeV	36	26.71 MeV
20	16	28.85 MeV	37	26.57 MeV
21	17	28.78 MeV	38	26.43 MeV
22	18	28.7 MeV	39	26.28 MeV
23	19	28.62 MeV	40	26.14 MeV
24	20	28.54 MeV		

Сурет 3.1 - Есептеу нәтижесі

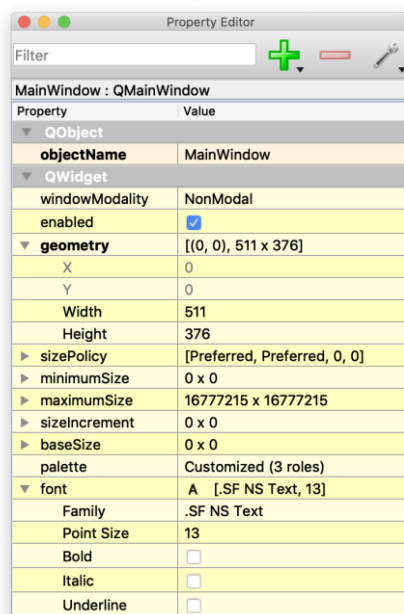
3.2 Бағдарламаның графикалық интерфейсі

3.2 суретте QtDesigner графикалық дизайнерінің интерфейсі Widget box көрсетілген. Ол жерде стандартты жиынтықтар мен басқару элементтері орналасқан. Интерфейс көмегімен вертикалды немесе горизонталды орналастыру шаблондарын, басу батырмасын, түрлі кестелер мен т.б. элементтері орналастыра аламыз.

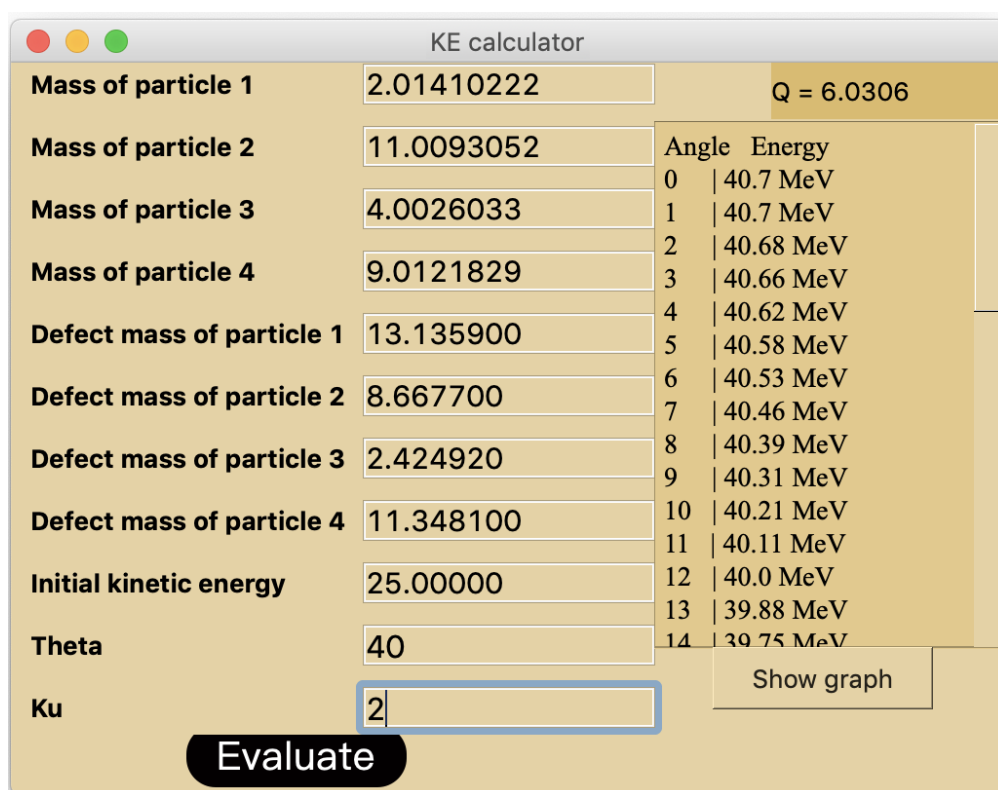


Сурет 3.2 - Widget box терезесі

3.3 суретте Property Editor меншікті редакторы көрсетілген. Редактор көмегімен бағдарламаның дизайнын жасау режимінен шықпай ақ элементтерді өңдеуге болады. Редакторда элементтің атын, терезедегі орналасқан орнын, пішінін, мәтіннің қаріпін(шрифт) өзгертуге болады.



Сурет 3.3 - Property Editor терезесі



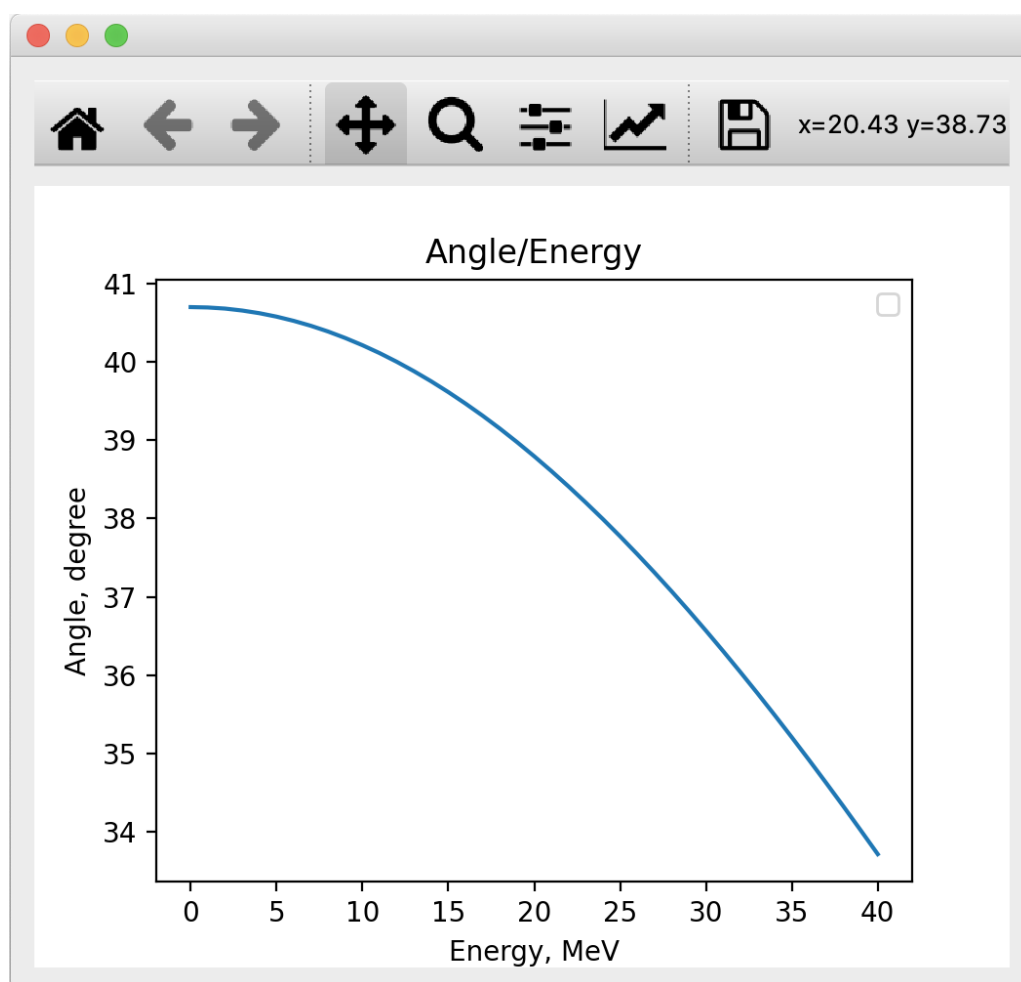
Сурет 3.4 - Бағдарламаның негізгі терезесі

3.4 суретте бағдарламаның негізгі терезі көрсетілген. Бағдарламаның атауы «KE calculator». Терезенің сол жақ бөлігінде кіріс сипаттамалар, яғни,

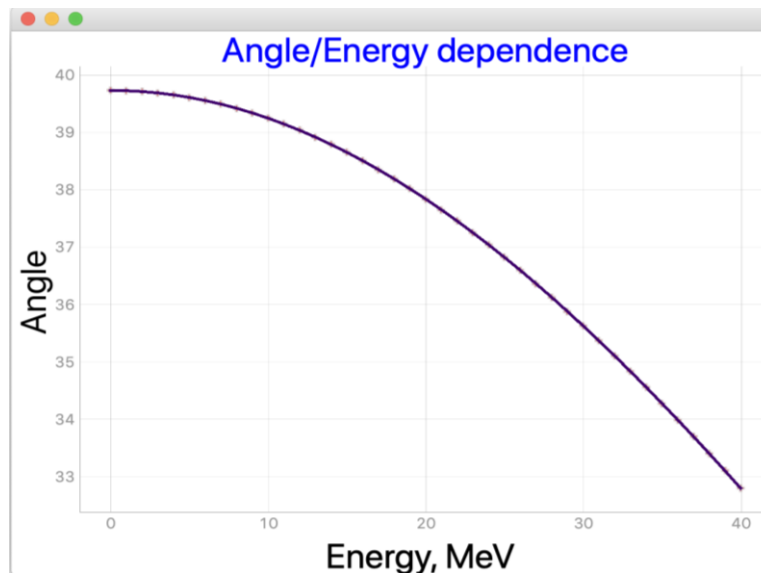
есептеулерге ұшырайтын шамалар жазылады. «Mass of particle» шамаларына бөлшектердің массалары, «Defect mass of particle» шамаларына бөлшектердің дефекту массалары, «Initial kinetic energy» шамасына бөлшектің шашырауға дейінгі кинетикалық энергиясы, «Theta» шамасына есептейтін бұрыштар, «Ku» шамасына қозу деңгейі жазылады. «Evaluate» батырмасын басқаннан кейін терезенің оң жақ бөлігінде реакция энергиясы, Q , және әр бұрышқа тиесілі кинетикалық энергия мәндері көрсетіледі.

3.5 суретте «Matplotlib» Python кітапханасының көмегімен жазылған бұрыш/энергия қатынасындағы график көрсетілген. Бұл графиктің артықшылығы курсор арқылы графиктегі сызықтың белгілі бір нүктесіндегі бұрыштың немесе энергияның нақты мәнін білсе болады. Графикті үлкейту, кішірейту мүмкіншіліктеріне және оны жеке сурет немесе файл ретінде сақтауға мүмкіндік береді.

3.6 суретте «pyqtgraph» Python кітапханасының көмегімен жазылған бұрыш/энергия қатынасындағы график көрсетілген.



Сурет 3.5 - Matplotlib кітапханасының көмегімен тұрғызылған график

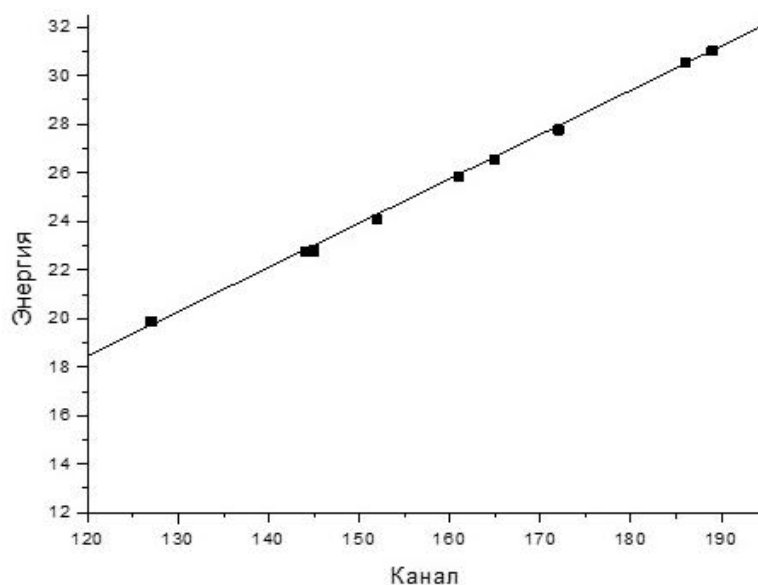


Сурет 3.6 - «pyqtgraph» Pyhon кітапханасының көмегімен жазылған бұрыш/энергия қатынасындағы график

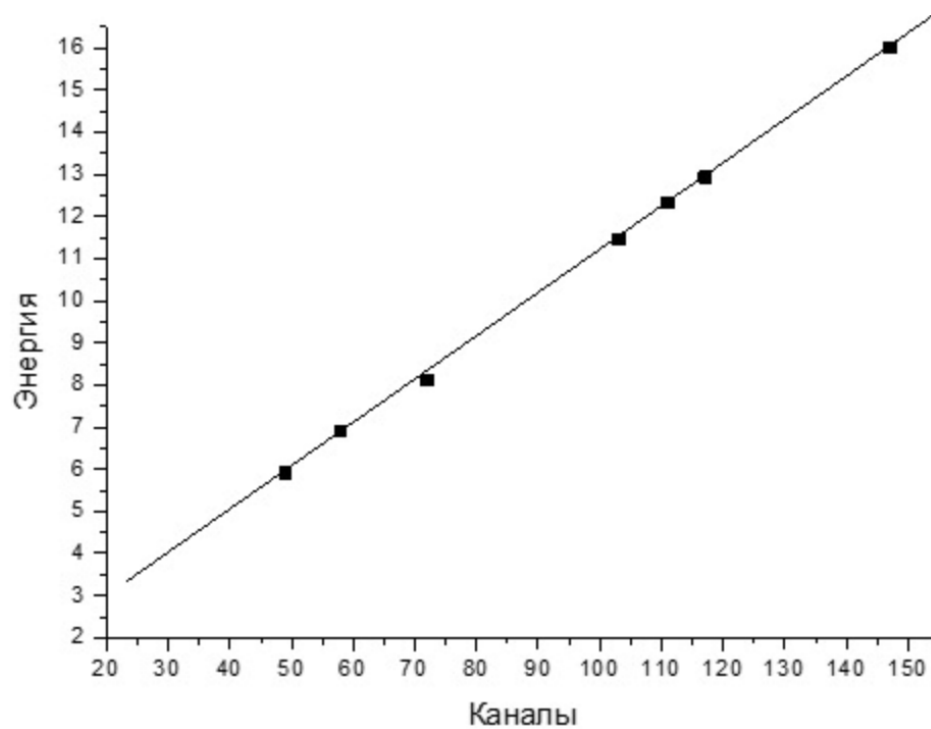
Үдеткіштегі зерттеу нәтижесінен алынған мәліметтер бағдарламаға енгізілген соң, есептеулер орындалады. Қорытындысы жеке терезеде ашылып, шашырау диаграммасы график түрінде көрсетіледі.

3.3 Әр түрлі спектрлерді талдау

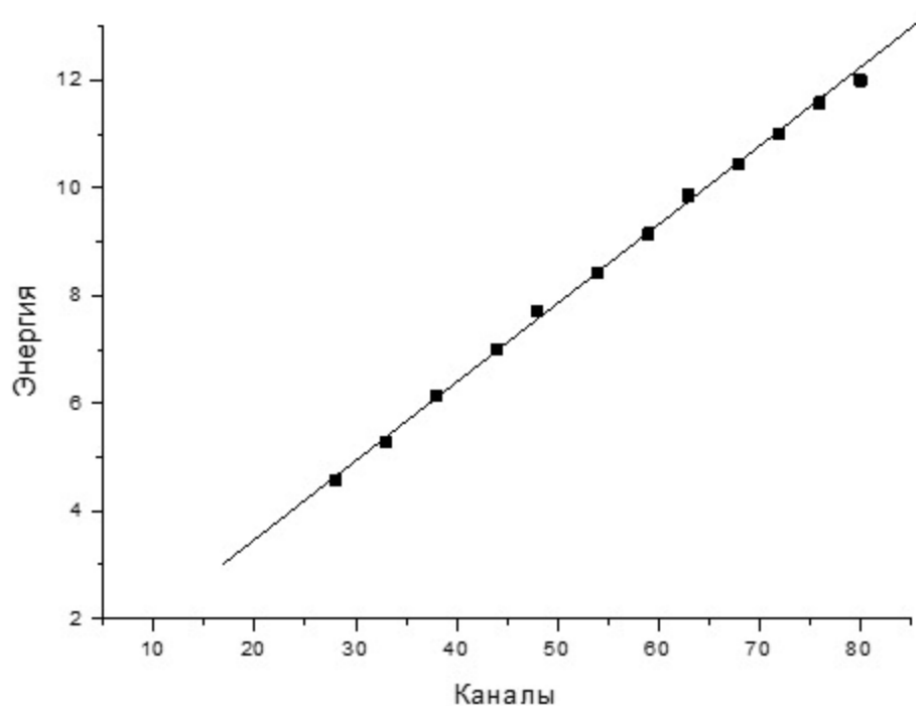
Тәжірибелерде электр сигналдарының амплитудасының ядролық өзара әрекеттесу өнімдерінің энергиясына сызықтық тәуелділігі бар кремний детекторлары қолданылады.



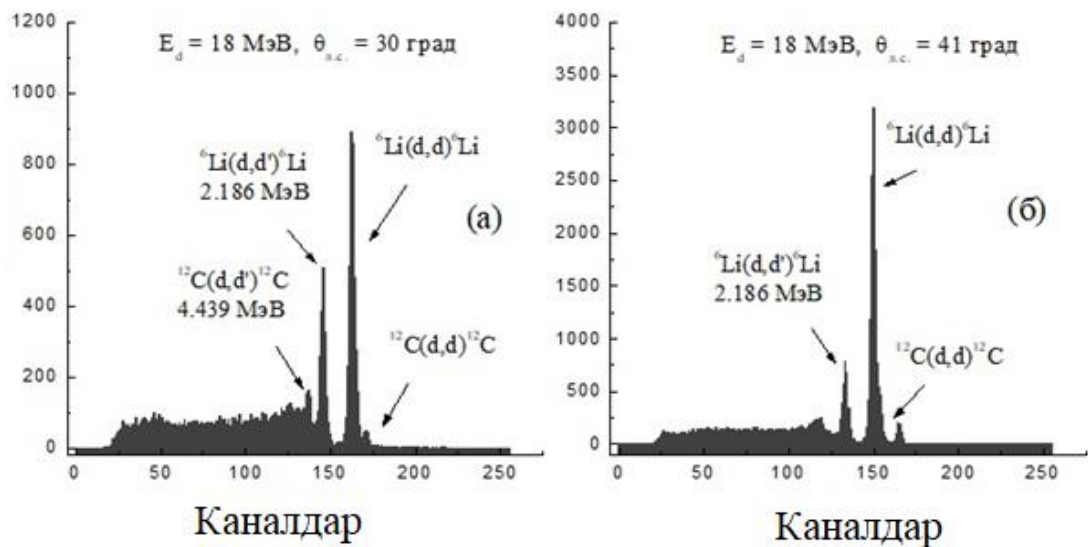
Сурет 3.7 - $E_{\text{He}} = 34 \text{ МэВ}$ энергиясында 16° бұрышпен өлшенген ${}^6\text{Li}({}^3\text{He}, d){}^7\text{Be}$ реакциясының спектрін калибрлеу



Сурет 3.8 - $E_d = 18\text{МэВ}$ энергиямен 44° бұрышпен өлшенген $^{13}\text{C}(d,d)^{13}\text{C}$ реакция спектрін калибрлеу



Сурет 3.9 - ^{14}N иондарының шашырау спектрлерін $E_{14\text{N}} = 1,75\text{ МэВ/нуклон}$ энергиясымен өлшенген ^{14}C ядролармен калибрлеу



Сурет 3.10 - ${}^6\text{Li}$ ядроларға шашыраған дейтерондардың спектрлері $E_d = 18\text{МэВ}$ энергиямен 30 (а) және 41 (б) градус бұрыштарда өлшенді

Бағдарламаның негізгі коды

```
import math
with open("data.txt", "r") as f:
    readdata = f.read()
    ff = [int(float(x)) for x in readdata.split()]
m1=ff[0]
m2=ff[1]
m3=ff[2]
m4=ff[3]
dm1=ff[4]
dm2=ff[5]
dm3=ff[6]
dm4=ff[7]
e1=ff[8]
kteta=int(input('Есептеу бұрышы = '))
ku=float(input('Қозу деңгейі = '))
C = str(input('Есептейміз бе? '))
if C=='y':
    def case1():
        D = str(input('Жаңа файлға шығарамыз ба? '))
        if D == 'y':
            newname = str(input('Жаңа файл атауы: '))
            f1=open(newname,'w')
            q=(dm1+dm2)-(dm3+dm4)-ku
            f1.write('Q='+str(round(q, 6)))
            f1.write(' Қозу деңгейі =' +str(ku)+'\n')
            f1.write('Бұрыш | ')
```

```

f1.write('Энергия\n')
f1.write('_____ \n')
v1=m1*m3*e1
v2=m3+m4
v=v1/(v2*v2)
w=((m4-m1)*e1+m4*q)*v2/v1
theta= 0
while theta <= kteta:
    gr = (math.pi * theta) / 180
    co = math.cos(gr)
    e3 = (co + math.sqrt(co * co + w)) ** 2 * v
    if theta>9:
        f2=f1.write(str(theta)+' | '+str(round(e3, 2))+ ' MeV ' + '\n')
    else:
        f2 = f1.write(str(theta) + ' | ' + str(round(e3, 2)) + ' MeV' + '\n')
    theta = theta + 1

s=str(input('Қайта есептейміз бе? '))
if s=='y':
    case1()
else:
    case1()
    print('Есептеу аяқталды!')
else:
    print('Есептеу аяқталды!')

```

ҚОРЫТЫНДЫ

Python және Java бағдарламалау тілдері негізінде ядролық реакциялар өнімдерінің кинетикалық энергиясын есептеу үшін компьютерлік бағдарлама жасалды.

Кинематикалық анализдің көмегімен түзілген бөлшектер спектрлерінің сапалық құрамын жоғары дәлдікпен анықталды. Есептеулерді барынша жылдамдату мақсатында және салыстыру үшін Java бағдарламалау тілі де қолданылды. Үдеткіштегі зерттеу нәтижесінен алынған мәліметтер бағдарламаға енгізіліп, есептеулер орындалды. Қорытындысы жеке терезеде ашылып, шашырау диаграммасы график түрінде көрсетілді.

Бағдарламаның графикалық интерфейсі QT-Designer кроссплатформалық ортасында PyQt5 құралдар жиынтығы көмегімен бағдарлама құрылды.

Python және Java тілдерінде оптимизацияланған кодтарды есептеу уақыты салыстырылды. Екі тілде де есептеу уақыты аз және жылдамдықта қатты айырмашылық жоқтығы анықталды.

Пайдаланылган әдебиеттер тізімі

1. K.N.Mukhin. Experimental nuclear physics. Vol.1. М.: Изд-во «Лань» 2021.
2. K.N.Mukhin. Experimental nuclear physics. Vol.2. М.: Изд-во «Лань» 2021. 6 с.
3. Блан Д. Ядра, частицы, ядерные реакторы: Пер. с франц. -М.: «Мир», 1989. -336 с., нл.
4. А. М. Балдин, В. И. Гольданский, И. Л. Розенталь. Кинематика ядерных реакции. «Государственное издательство физико-математической литературы», 1959.
5. Dubovichenko S., Burtebayev N., Dzhazairov-Kakhramanov A., Zazulin D., Kerimkulov Zh., Nassurlla M., Omarov Ch., Tkachenko A., Shmygaleva T., Kliczewski S. New measurements and phase shift analysis of $p^{16}\text{O}$ elastic scattering at astrophysical energies // Chinese Physics C. – 2017. – Vol. 41. – No. 1. – P. 014001(10).
6. Burtebayev N., Sakhiyev S.K., Janseitov D.M., Kerimkulov Zh., Alimov D. Investigation of the elastic and inelastic scattering of α -particles from ^{13}C in the energy range 26.6–65 MeV // International Journal of Modern Physics E. – 2016. – Vol. 25. – No. 10. – P. 1650078(12).
7. Burtebayev N., Janseitov D. M., Kerimkulov Zh., Mukhamejanov Y. S., Nassurlla M., Demyanova A. S., Danilov A. N., Ogloblin, A. A., Aimaganbetov A. S. Investigation of exotic states of ^{13}C at low energy // International Journal of Modern Physics E. – 2018. – Vol. 27. – No. 3. – P. 1850025(7)
8. Burtebayev N., Burtebayeva J. T., Duisebayev A., Kerimkulov Zh., K., Nassurlla M., Zholdybayev T., Artemov S. V., Karakhodzhayev A. A., Salikhbayev U. S., Sakuta S. B., Kliczewski S., Piasecki E., Rusek K., Siudak R., Trzcinska A., Wolinska-Cichocka M., Amar A. Mechanizm of the $^7\text{Li}(d,t)^6\text{Li}$ reaction at 25 MeV energy of deuterons, values of spectroscopic factors and asymptotic normalization coefficients for the $\text{Li-7} \rightarrow \text{Li-6} + n$ vertex // Acta Physica Polonica B. – 2015. – Vol. 46. – No. 5. – P. 1037-1054.
9. Nassurlla M., Burtebayev N., Janseitov D.M., Kerimkulov Zh., Alimov D., Morzabayev A.K., Talpakova K., Mukhamejanov Y., Galanina L.I., Demyanova A.S., Danilov A.N., Starastin V. Study of elastic and inelastic scattering of deuterons by ^9Be at energy $E = 14.5 \text{ MeV}$ // Acta Physica Polonica B. – 2020. – Vol.51, - No.3. – P. 751-756
10. Распространенность изотопов в природе и их масса, точная таблица. <https://dpva.ru/Guide/GuideChemistry/PeriodicalMendeleevTable/IsotopiAbundance1/>
11. Centre for photonuclear experiments data. <http://cdfc.sinp.msu.ru/services/ensdfr.html>
12. Реакции. Характеристики вероятности реакций. http://nuclphys.sinp.msu.ru/nucmodl/nml11_1.htm

13. Ускорители заряженных частиц.
<https://www.booksite.ru/fulltext/1/001/008/114/616.htm>
14. Particle accelerator. <https://www.accessscience.com/content/particle-accelerator/490900>
15. Plotting with PyQtGraph. <https://www.mfitzp.com/tutorials/plotting-pyqtgraph/>
16. Kerimkulov Zh., Burtebayev N., Burtebayeva J.T., Duisebayev A., Nassurlla M., Zholdybayev T., Artemov S.V., Karakhodzhayev A.A., Salikhibayev U.S., Sakuta S.B., Kliczewski S., Piasecki E., Rusek K., Siudak R., Trzcińska A., Wolińska-Cichocka M., Amar A. Mechanism of the ${}^7\text{Li}(d,t){}^6\text{Li}$ reaction at 25 MeV energy of deuterons, values of spectroscopic factors and asymptotic normalization coefficients for the ${}^7\text{Li} \rightarrow {}^6\text{Li} + n$ vertex // Acta Physica Polonica B. – 2015. – Vol. 46. – No. 5. – P. 1037-1054
17. Dubovichenko S.B., Kerimkulov Zh.K., Burtebaev N., Zazulin D.M., Amar A. S. A. Astrophysical S-Factor for the Radiative-Capture Reaction $p{}^6\text{Li} \rightarrow {}^7\text{Be}\gamma$ // Physics of Atomic Nuclei. – 2011. – Vol. 74. – No. 7. – P. 984–1000
18. Физика атома и атомного ядра.
<https://physics.ru/courses/op25part2/content/chapter6/section/paragraph8/theory.html#.YKuacileO9a>
19. Г.И. Бабат. Ускорители. Молодая гвардия. 1957 г.
20. В. Е. Бунаков. Кинетические уравнения в теории ядерных реакций. «Физика элементарных частиц и атомного ядра», 1980, ТОМ 11, Вып. 6

Бағдарламаның Python тіліндегі коды

```

import sys
import math
import matplotlib
matplotlib.use('Qt5Agg')
from PyQt5 import QtCore, QtGui, QtWidgets
from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg,
NavigationToolbar2QT as NavigationToolbar
from matplotlib.figure import Figure
from pdata import __init__
class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(511, 376)
        MainWindow.setStyleSheet("background-color: rgb(255, 255, 255);")
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setStyleSheet("\n"
"background-color: rgb(228, 210, 166);")
        self.centralwidget.setObjectName("centralwidget")
        self.Basu = QtWidgets.QPushButton(self.centralwidget)
        self.Basu.setGeometry(QtCore.QRect(90, 340, 113, 32))
        font = QtGui.QFont()
        font.setFamily(".PingFang SC")
        font.setPointSize(20)
        font.setBold(False)
        font.setItalic(False)
        font.setUnderline(False)
        font.setWeight(50)
        font.setStrikeOut(False)
        font.setKerning(True)
        self.Basu.setFont(font)
        self.Basu.setMouseTracking(False)
        self.Basu.setStyleSheet("background-color: rgb(3, 0, 1);\n"
"border-radius: 15px;\n"
"color: white;")
        self.Basu.setObjectName("Basu")
        self.gridLayoutWidget = QtWidgets.QWidget(self.centralwidget)
        self.gridLayoutWidget.setGeometry(QtCore.QRect(10, 0, 321, 345))
        self.gridLayoutWidget.setObjectName("gridLayoutWidget")
        self.gridLayout = QtWidgets.QGridLayout(self.gridLayoutWidget)
        self.gridLayout.setContentsMargins(0, 0, 0, 0)
        self.gridLayout.setObjectName("gridLayout")
        self.label_8 = QtWidgets.QLabel(self.gridLayoutWidget)

```

```
font = QtGui.QFont()
font.setBold(True)
font.setWeight(75)
self.label_8.setFont(font)
self.label_8.setObjectName("label_8")
self.gridLayout.addWidget(self.label_8, 6, 0, 1, 1)
self.e = QtWidgets.QLineEdit(self.gridLayoutWidget)
font = QtGui.QFont()
font.setPointSize(15)
font.setBold(False)
font.setItalic(False)
font.setWeight(50)
self.e.setFont(font)
self.e.setObjectName("e")
self.gridLayout.addWidget(self.e, 8, 1, 1, 1)
self.label_5 = QtWidgets.QLabel(self.gridLayoutWidget)
font = QtGui.QFont()
font.setBold(True)
font.setWeight(75)
self.label_5.setFont(font)
self.label_5.setObjectName("label_5")
self.gridLayout.addWidget(self.label_5, 3, 0, 1, 1)
self.label_6 = QtWidgets.QLabel(self.gridLayoutWidget)
font = QtGui.QFont()
font.setBold(True)
font.setWeight(75)
self.label_6.setFont(font)
self.label_6.setObjectName("label_6")
self.gridLayout.addWidget(self.label_6, 4, 0, 1, 1)
self.label_9 = QtWidgets.QLabel(self.gridLayoutWidget)
font = QtGui.QFont()
font.setBold(True)
font.setWeight(75)
self.label_9.setFont(font)
self.label_9.setObjectName("label_9")
self.gridLayout.addWidget(self.label_9, 7, 0, 1, 1)
self.label_4 = QtWidgets.QLabel(self.gridLayoutWidget)
font = QtGui.QFont()
font.setBold(True)
font.setWeight(75)
self.label_4.setFont(font)
self.label_4.setObjectName("label_4")
```

```
self.gridLayout.addWidget(self.label_4, 2, 0, 1, 1)
self.DM2 = QtWidgets.QLineEdit(self.gridLayoutWidget)
font = QtGui.QFont()
font.setPointSize(15)
self.DM2.setFont(font)
self.DM2.setObjectName("DM2")
self.gridLayout.addWidget(self.DM2, 5, 1, 1, 1)
self.M2 = QtWidgets.QLineEdit(self.gridLayoutWidget)
font = QtGui.QFont()
font.setPointSize(15)
self.M2.setFont(font)
self.M2.setObjectName("M2")
self.gridLayout.addWidget(self.M2, 1, 1, 1, 1)
self.M3 = QtWidgets.QLineEdit(self.gridLayoutWidget)
font = QtGui.QFont()
font.setPointSize(15)
self.M3.setFont(font)
self.M3.setObjectName("M3")
self.gridLayout.addWidget(self.M3, 2, 1, 1, 1)
self.label_3 = QtWidgets.QLabel(self.gridLayoutWidget)
font = QtGui.QFont()
font.setBold(True)
font.setWeight(75)
self.label_3.setFont(font)
self.label_3.setObjectName("label_3")
self.gridLayout.addWidget(self.label_3, 1, 0, 1, 1)
self.label_2 = QtWidgets.QLabel(self.gridLayoutWidget)
font = QtGui.QFont()
font.setPointSize(13)
font.setBold(True)
font.setWeight(75)
self.label_2.setFont(font)
self.label_2.setObjectName("label_2")
self.gridLayout.addWidget(self.label_2, 0, 0, 1, 1)
self.M4 = QtWidgets.QLineEdit(self.gridLayoutWidget)
font = QtGui.QFont()
font.setPointSize(15)
self.M4.setFont(font)
self.M4.setObjectName("M4")
self.gridLayout.addWidget(self.M4, 3, 1, 1, 1)
self.M1 = QtWidgets.QLineEdit(self.gridLayoutWidget)
self.M1.setEnabled(True)
```

```
font = QtGui.QFont()
font.setPointSize(15)
self.M1.setFont(font)
self.M1.setObjectName("M1")
self.gridLayout.addWidget(self.M1, 0, 1, 1, 1)
self.label_10 = QtWidgets.QLabel(self.gridLayoutWidget)
font = QtGui.QFont()
font.setBold(True)
font.setWeight(75)
self.label_10.setFont(font)
self.label_10.setObjectName("label_10")
self.gridLayout.addWidget(self.label_10, 8, 0, 1, 1)
self.label_7 = QtWidgets.QLabel(self.gridLayoutWidget)
font = QtGui.QFont()
font.setBold(True)
font.setWeight(75)
self.label_7.setFont(font)
self.label_7.setObjectName("label_7")
self.gridLayout.addWidget(self.label_7, 5, 0, 1, 1)
self.DM4 = QtWidgets.QLineEdit(self.gridLayoutWidget)
font = QtGui.QFont()
font.setPointSize(15)
self.DM4.setFont(font)
self.DM4.setObjectName("DM4")
self.gridLayout.addWidget(self.DM4, 7, 1, 1, 1)
self.DM3 = QtWidgets.QLineEdit(self.gridLayoutWidget)
font = QtGui.QFont()
font.setPointSize(15)
self.DM3.setFont(font)
self.DM3.setObjectName("DM3")
self.gridLayout.addWidget(self.DM3, 6, 1, 1, 1)
self.DM1 = QtWidgets.QLineEdit(self.gridLayoutWidget)
font = QtGui.QFont()
font.setPointSize(15)
self.DM1.setFont(font)
self.DM1.setObjectName("DM1")
self.gridLayout.addWidget(self.DM1, 4, 1, 1, 1)
self.label_11 = QtWidgets.QLabel(self.gridLayoutWidget)
font = QtGui.QFont()
font.setBold(True)
font.setWeight(75)
self.label_11.setFont(font)
```

```
self.label_11.setObjectName("label_11")
self.gridLayout.addWidget(self.label_11, 9, 0, 1, 1)
self.label_12 = QtWidgets.QLabel(self.gridLayoutWidget)
font = QtGui.QFont()
font.setBold(True)
font.setWeight(75)
self.label_12.setFont(font)
self.label_12.setObjectName("label_12")
self.gridLayout.addWidget(self.label_12, 10, 0, 1, 1)
self.ktet = QtWidgets.QLineEdit(self.gridLayoutWidget)
font = QtGui.QFont()
font.setPointSize(16)
self.ktet.setFont(font)
self.ktet.setText("")
self.ktet.setObjectName("ktet")
self.gridLayout.addWidget(self.ktet, 9, 1, 1, 1)
self.ku = QtWidgets.QLineEdit(self.gridLayoutWidget)
font = QtGui.QFont()
font.setPointSize(16)
self.ku.setFont(font)
self.ku.setText("")
self.ku.setObjectName("ku")
self.gridLayout.addWidget(self.ku, 10, 1, 1, 1)
self.result = QtWidgets.QTextBrowser(self.centralwidget)
self.result.setGeometry(QtCore.QRect(330, 30, 181, 271))
font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(14)
font.setBold(False)
font.setItalic(False)
font.setWeight(50)
self.result.setFont(font)
self.result.setStyleSheet("background-color: rgb(225, 201, 143);\n"
"font: 14pt \"Times New Roman\";")
self.result.setObjectName("result")
self.gra = QtWidgets.QPushButton(self.centralwidget)
self.gra.setGeometry(QtCore.QRect(360, 300, 113, 32))
self.gra.setObjectName("gra")
self.Qwrite = QtWidgets.QLabel(self.centralwidget)
self.Qwrite.setGeometry(QtCore.QRect(390, 0, 121, 29))
self.Qwrite.setStyleSheet("background-color: rgb(216, 187, 115);")
self.Qwrite.setText("")
```

```

self.Qwrite.setObjectName("Qwrite")
font = QtGui.QFont()
font.setPointSize(19)
font.setBold(True)
font.setWeight(75)
MainWindow.setCentralWidget(self.centralwidget)
self.retranslateUi(MainWindow)
QtCore.QMetaObject.connectSlotsByName(MainWindow)
def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow", "KE calculator"))
    self.Basu.setText(_translate("MainWindow", "Evaluate"))
    self.label_8.setText(_translate("MainWindow", "Defect mass of particle 3"))
    self.e.setText(_translate("MainWindow", "25.00000"))
    self.label_5.setText(_translate("MainWindow", "Mass of particle 4"))
    self.label_6.setText(_translate("MainWindow", "Defect mass of particle 1"))
    self.label_9.setText(_translate("MainWindow", "Defect mass of particle 4"))
    self.label_4.setText(_translate("MainWindow", "Mass of particle 3"))
    self.DM2.setText(_translate("MainWindow", "8.667700"))
    self.M2.setText(_translate("MainWindow", "11.0093052"))
    self.M3.setText(_translate("MainWindow", "4.0026033"))
    self.label_3.setText(_translate("MainWindow", "Mass of particle 2"))
    self.label_2.setText(_translate("MainWindow", "Mass of particle 1"))
    self.M4.setText(_translate("MainWindow", "9.0121829"))
    self.M1.setText(_translate("MainWindow", "2.01410222"))
    self.label_10.setText(_translate("MainWindow", "Initial kinetic energy"))
    self.label_7.setText(_translate("MainWindow", "Defect mass of particle 2"))
    self.DM4.setText(_translate("MainWindow", "11.348100"))
    self.DM3.setText(_translate("MainWindow", "2.424920"))
    self.DM1.setText(_translate("MainWindow", "13.135900"))
    self.label_11.setText(_translate("MainWindow", "Theta"))
    self.label_12.setText(_translate("MainWindow", "Ku"))
    self.result.setHtml(_translate("MainWindow", "<!DOCTYPE HTML PUBLIC
\"-//W3C//DTD HTML 4.0//EN\" \"http://www.w3.org/TR/REC-
html40/strict.dtd\">\n"
"<html><head><meta name=\"qrichtext\" content=\"1\" /><style
type=\"text/css\">\n"
"p, li { white-space: pre-wrap; }\n"
"</style></head><body style=\" font-family:'Times New Roman'; font-size:14pt;
font-weight:400; font-style:normal;\">\n"

```

```

"<p style=\"-qt-paragraph-type:empty; margin-top:0px; margin-bottom:0px; margin-
left:0px; margin-right:0px; -qt-block-indent:0; text-indent:0px;\"><br
/></p></body></html>"))
    self.gra.setText(_translate("MainWindow", "Show graph"))
# Create application
app = QtWidgets.QApplication(sys.argv)
MainWindow = QtWidgets.QMainWindow()
ui = Ui_MainWindow()
ui.setupUi(MainWindow)
MainWindow.show()
li_1 = []
li_2 = []
def resultt():
    f1 = open("newname1.txt", 'w')
    f4 = open("new.txt", 'w')
    m1 = float(ui.M1.text())
    m2 = float(ui.M2.text())
    m3 = float(ui.M3.text())
    m4 = float(ui.M4.text())
    dm1 = float(ui.DM1.text())
    dm2 = float(ui.DM2.text())
    dm3 = float(ui.DM3.text())
    dm4 = float(ui.DM4.text())
    e1 = float(ui.e.text())
    kteta = float(ui.ktet.text())
    k = float(ui.ku.text())
    q = (dm1 + dm2) - (dm3 + dm4) - k
    v1 = m1 * m2 * e1
    v2 = m3 + m4
    v = v1 / (v2 * v2)
    w = (((m4 - m1) * e1 + m4 * q) * v2) / v1
    theta = 0
    f3 = f1.write('Angle  Energy \n')
    while theta <= kteta:
        gr = (math.pi * theta) / 180
        co = math.cos(gr)
        e3 = (co + math.sqrt(co * co + w)) ** 2 * v
        if theta<10:
            f2 = (str(theta) + ' | ' + str(round(e3, 2)) + ' MeV ' + ")
            f3 = f1.write(str(theta) + ' | ' + str(round(e3, 2)) + ' MeV ' + '\n')
            li_1.append(theta)
            li_2.append(round(e3, 3))

```

```

else:
    f2 = (str(theta) + ' | ' + str(round(e3, 2)) + ' MeV ' + ")
    f3 = f1.write(str(theta) + ' | ' + str(round(e3, 2)) + ' MeV ' + '\n')
    li_1.append(theta)
    li_2.append(round(e3, 3))
    theta = theta + 1
f4.write(', '.join([str(i) for i in li_1]))
f4.write('\n')
f4.write(', '.join([str(i) for i in li_2]))
f4.close()
def flo():
    dm1 = float(ui.DM1.text())
    dm2 = float(ui.DM2.text())
    dm3 = float(ui.DM3.text())
    dm4 = float(ui.DM4.text())
    k = float(ui.ku.text())
    f = open("newname1.txt", "r+")
    readdata = f.read()
    ui.result.setText(f'{readdata}')
    f.close()
    Q = (dm1 + dm2) - (dm3 + dm4) - k
    ui.Qwrite.setText("Q = " + str(round(Q, 4)))
ui.gra.clicked.connect(__init__)
ui.Basu.clicked.connect(resultt)
ui.Basu.clicked.connect(flo)
sys.exit(app.exec_())

```

Бағдарламаның Java тіліндегі коды

```

package com.company;
import java.io.*;
import java.math.BigDecimal;
import java.math.MathContext;
import java.math.RoundingMode;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
public class Main {
    public static void case1() {

```

```
Scanner scan = new Scanner(System.in);
}
public static void main(String[] args) {
    File myFile = new File("date.txt");
    Scanner myReader = null;
    Scanner scanner = new Scanner(System.in);
    List<BigDecimal> fList = new ArrayList<BigDecimal>();
    try {
        myReader = new Scanner(myFile);
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
    while (myReader.hasNextLine()) {
        String data = myReader.nextLine();
        fList.add(new BigDecimal(data));
    }
    myReader.close();
    System.out.println(fList);

    BigDecimal m1 = fList.get(0);
    BigDecimal m2 = fList.get(1);
    BigDecimal m3 = fList.get(2);
    BigDecimal m4 = fList.get(3);
    BigDecimal dm1 = fList.get(4);
    BigDecimal dm2 = fList.get(5);
    BigDecimal dm3 = fList.get(6);
    BigDecimal dm4 = fList.get(7);
    BigDecimal e1 = fList.get(8);

    System.out.print("Есептеу бұрышы = ");
    int kteta = scanner.nextInt();
    System.out.print("Қозу деңгейі = ");
    BigDecimal ku = scanner.nextBigDecimal();
    System.out.print("Есептейміз бе?");
    char c = scanner.next().charAt(0);

    if (c == 'y') {
        System.out.println("Жаңа файлға шығарамыз ба?");
        char D = scanner.next().charAt(0);
        while (D == 'y') {
            System.out.print("Жаңа файл атауы:");
            String newName = scanner.next();
        }
    }
}
```

```

Path path = Paths.get("data" + newName + ".txt"); //creates Path instance
try {
    Path p = Files.createFile(path);
    BufferedWriter writer = new BufferedWriter(new
FileWriter(p.toString()));
    BigDecimal q = ((dm1.add(dm2)).subtract(dm3.add(dm4))).subtract(ku);
    MathContext m = new MathContext(6);
    writer.write("Q = " + q.round(m));
    writer.write("Қозу деңгейі = " + ku + "\n");
    writer.write("Бұрыш |");
    writer.write("Энергия" + "\n");
    writer.write("_____ " + "\n");

    BigDecimal v1 = m1.multiply(m3).multiply(e1);
    BigDecimal v2 = m3.subtract(m4);
    BigDecimal v = v1.divide(v2.multiply(v2),2
RoundingMode.HALF_UP);
    BigDecimal w =
(((m4.subtract(m1)).multiply(e1).multiply(m4.multiply(q))).multiply(v2)).divide(v1,2
,RoundingMode.HALF_UP);

    int theta = 0;
    while (theta <= kteta) {
        double gr = (Math.PI * theta) / theta;
        double co = Math.cos(gr);
        System.out.println(co);
        double e3 = Math.pow(co + Math.sqrt(new BigDecimal(co *
co).add(w).doubleValue()), v.multiply(BigDecimal.valueOf(2)).doubleValue());
        if (theta > 9) {
            writer.write(theta + " | " + Math.round(e3) + " Mev\n");
        } else {
            writer.write(theta + " | " + Math.round(e3) + " Mev\n");
        }
        theta = theta + 1;
    }

    writer.close();
} catch (IOException e) {
    e.printStackTrace();
}

```

```
    }  
  }  
}
```

Matplotlib кітапханасының көмегімен график тұрғызу коды

```
from PyQt5 import QtCore, QtGui, QtWidgets  
import sys  
from PyQt5 import QtCore, QtGui, QtWidgets  
import matplotlib  
matplotlib.use('Qt5Agg')  
from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg,  
NavigationToolbar2QT as NavigationToolbar  
from matplotlib.figure import Figure  
  
with open("new.txt", "r") as f:  
    readdata = f.read()  
    ff = [str(x) for x in readdata.split()]  
m=[int(s) for s in ff[0].split(',')]  
n=[float(s) for s in ff[1].split(',')]  
print(m)  
print(n)  
  
class MplCanvas(FigureCanvasQTAgg):  
    def __init__(self, parent=None, width=5, height=4, dpi=100):  
        fig = Figure(figsize=(width, height), dpi=dpi)  
        self.axes = fig.add_subplot(111)  
        super(MplCanvas, self).__init__(fig)  
  
class MainWindow(QtWidgets.QMainWindow):  
  
    def __init__(self, *args, **kwargs):  
        super(MainWindow, self).__init__(*args, **kwargs)  
  
        sc = MplCanvas(self, width=5, height=4, dpi=100)  
        sc.axes.plot(m, n)  
  
        # Create toolbar, passing canvas as first parament, parent (self, the  
        MainWindow) as second.  
        toolbar = NavigationToolbar(sc, self)  
        layout = QtWidgets.QVBoxLayout()
```

А қосымшасының жалғасы

```
layout.addWidget(toolbar)
layout.addWidget(sc)
```

```
# Create a placeholder widget to hold our toolbar and canvas.
widget = QtWidgets.QWidget()
widget.setLayout(layout)
self.setCentralWidget(widget)
self.show()
```

```
app = QtWidgets.QApplication(sys.argv)
w = MainWindow()
app.exec_()
```

```
pyqtgraph кітапханасының көмегімен график тұрғызу коды
from PyQt5 import QtWidgets, QtCore
from pyqtgraph import PlotWidget, plot
import pyqtgraph as pg
import sys # We need sys so that we can pass argv to QApplication
import os
with open("new.txt", "r") as f:
    readdata = f.read()
    ff = [str(x) for x in readdata.split()]
```

```
m=[int(s) for s in ff[0].split(',')]
n=[float(s) for s in ff[1].split(',')]
print(m)
print(n)
```

```
class MainWindow1(QtWidgets.QMainWindow):
    def __init__(self, *args, **kwargs):
        super(MainWindow1, self).__init__(*args, **kwargs)
        self.graphWidget = pg.PlotWidget()
        self.setCentralWidget(self.graphWidget)
        self.graphWidget.setBackground('w')
        self.graphWidget.setTitle("<span style='color:blue;font-size:30pt'>Angle/Energy dependence</span>")
        pen = pg.mkPen(color=(60, 0, 106), width=5, style=QtCore.Qt.SolidLine)
        self.graphWidget.plot(m, n, pen=pen, symbol='+', symbolSize=6,
symbolBrush=('r'))
        self.graphWidget.plot(m, n, pen=pen)
        self.graphWidget.showGrid(x=True, y=True)
        styles = {'color': 'b', 'font-size': '30px'}
        self.graphWidget.setLabel('left', 'Angle', **styles)
        self.graphWidget.setLabel('bottom', 'Energy, MeV', **styles)
```

```
def main():  
    app = QtWidgets.QApplication(sys.argv)  
    main = MainWindow1()  
    main.show()  
    sys.exit(app.exec_())  
if __name__ == '__main__':  
    main()
```