

UNIVERSITATEA "BABEȘ-BOLYAI" CLUJ-NAPOCA
FACULTATEA DE FIZICĂ
SPECIALIZAREA FIZICĂ TEHNOLOGICĂ

LUCRARE DE LICENȚĂ

Coordonatori Științifici
Prof. Dr. Ioan Grosu
C.S. I Dr. Liviu-Petru Zârbo

Absolvent
Miruna-Andreea Căpușan

2022

UNIVERSITATEA "BABEȘ-BOLYAI" CLUJ-NAPOCA
FACULTATEA DE FIZICĂ
SPECIALIZAREA FIZICĂ TEHNOLOGICĂ

LUCRARE DE LICENȚĂ

Qubiti bazați pe circuite supraconductoare cu superinductanțe

Coordonatori Științifici

Prof. Dr. Ioan Grosu
C.S. I Dr. Liviu-Petre Zârbo

Absolvent

Miruna-Andreea Căpușan

Cuprins

Introducere	5
1 Calculatoarele și calculul cuantic	6
1.1 Simularea cu ajutorul calculatoarelor cuantice	6
1.2 De la mașinile cuantice Turing la era NISQ	7
1.2.1 Prima demonstrație a supremației cuantice, chip-ul Sycamore	8
1.2.2 Ajungând la trei cifre, chip-ul Eagle	9
2 Qubiți supraconductori	11
2.1 Bitul unui calculator cuantic	11
2.2 Superpoziția cuantică	11
2.3 Registre cuantice	12
2.4 Qubiți supraconductori pentru calculatoarele cuantice	12
3 Partea teoretică	14
3.1 Analiza de circuit	14
3.2 Circuitul LC oscilant	15
3.2.1 Hamiltonianul circuitului LC oscilant	16
3.3 Circuitul qubitului supraconductor cu superinductanță	17
3.3.1 Joncțiunea Josephson	18
3.3.2 Hamiltonianul Fluxonului	18
3.4 Matricea hamiltonianului fluxonului	19
3.4.1 Matricea $\mathbf{a}^\dagger \mathbf{a}$	20
3.4.2 Matricea \mathbf{a}	21
3.4.3 Matricea \mathbf{a}^\dagger	22
4 Partea aplicativă	23
4.1 Jupyter Notebook și Python 3.8	23
4.2 Reprezentarea grafică a funcțiilor de undă	23
4.2.1 Inițializarea constantelor și operatorilor	23
4.2.2 Inițializarea energiilor E_C și E_L și variabilele n_{ZPF} și φ_{ZPF}	24
4.2.3 Descompunerea spectrală a matricii cosinus	25
4.2.4 Construirea matricii Hamiltonianului	26
4.2.5 Funcțiile de undă în funcție de frecvență	27
4.3 Studiul energiilor	32
4.3.1 Exprimarea Hamiltonianului în funcție de energii	32
4.3.2 Forma potențialului qubitului	33
4.3.3 Influența E_J asupra potențialului	35
4.3.4 Nivelele energetice ale sistemului	35
Concluzii	39

Abstract

Quantum computation and quantum information technology are part of a field that has benefited a lot due to a meteoric rise in popularity in recent years. Once developed, a noise-tolerant universal quantum computer will be able to give us access to an inconceivably greater computational power than any existing classical equivalent has given us before. One of the most promising and developed models for quantum bits, or qubits are based on superconducting circuit loops of various architecture.

The purpose of this paper is to study such a superconducting qubit in fluxonic architecture. To accomplish this goal, we calculated the Hamiltonian elements of the superconducting fluxon loop, using the lumped-element model, and built its matrix representation. We ran simulations of the qubit's fluxonic behavior by accessing a server dedicated to quantum simulation within the National Research-Development Institute for Isotopic and Molecular Technologies (INCDTIM). The work is divided in four main chapters.

The historical context and story of quantum computers from the 1980s to the present is presented in the first chapter. There are some discussions about the major events that marked the development of quantum computers, as well as some predictions and perspectives regarding the future of this powerful technology.

Chapter 2 discusses superconducting qubits. A brief introduction to what a qubit is and how it differs from a classical bit is given. The phenomenon of quantum superposition and quantum parallelism is presented as well as the DiVincenzo criteria for building a general quantum computer. The stages of development and the milestones needed to be achieved by a quantum system in order to be able to make error-tolerant quantum calculations are mentioned, as well as how far superconducting qubits are in this developmental stages compared to other types of qubits.

In chapter 3 or "The theoretical part" we presented the theoretical basis of the circuit analysis we did and the simplifying hypotheses we used. In the theoretical model presented, we calculated the Hamiltonian of a simple LC oscillating circuit and the Hamiltonian of the superconducting qubit of interest. Having the analytical form of the hamiltonian, the process used to transform it into a matrix is described: the form of the hamiltonian is obtained in the second quantification model and using the matrix representation of the creation and annihilation operators (obtained by matrix element calculation) the hamiltonian matrix is developed.

Having the Hamiltonian matrix form from chapter 3, in chapter 4, the algorithms used to simulate the wave functions of the system, the potential, and the energy levels of interest are introduced. We presented the construction of the Hamiltonian matrix and the creation of the cosine of a matrix by spectral decomposition. The Josephson energy E_J is discussed and its influence over the energy level spectrum and potential of the qubit. Lastly, the energy level spectra of different realistic parameters sets (that already have physical implementations [42] [43]) are plotted next to each other and a discussion about the influence of the exterior magnetic flux φ over the energy levels is made.

Introducere

Calculatoarele cuantice sau ingineria informației cuantice este un domeniu care a beneficiat de o creștere meteorică de popularitate în ultimele decenii. Odată dezvoltate, calculatoarele cuantice universale vor avea posibilitatea să ne ofere accesul la o putere computațională inconceptibil mai mare decât orice echivalent clasic existent deja. În ultimii 5 ani, lumea este spectatoare la ceea ce a ajuns să fie numită „epoca NISQ” (Noisy Intermediate Scale Quantum) și la competiția tehnologică contra timp dintre giganții industriei de tehnologie spre proiectarea de calculatoare cuantice. Până în prezent, cele mai dezvoltate și promițătoare modele pentru qubit sunt loop-urile supraconductoare de diverse arhitecturi.

Scopul acestei lucrări este studiul unui astfel de qubit supraconductor în arhitectură fluxonică. Pentru îndeplinirea acestui obiectiv, am calculat inițial elementele hamiltonianului sistemului și matricea aferentă acestuia. Am realizat simulări ale comportamentului fluxonic prin acesarea unui server dedicat simulării cuantice al Institutului Național de Cercetare-Dezvoltare pentru Tehnologii Izotopice și Moleculare (INCDTIM). Lucrarea este împărțită în patru capitole principale.

În capitolul 1, Calculatoarele și calculul cuantic, sunt prezentate contextul istoric și povestea calculatoarelor cuantice din anii 1980 până în prezent, dar și câteva prognoze și perspective asupra viitorului, punându-se accent pe momentele majore și decisive din evoluția acestei tehnologii.

Capitolul 2, Qubiți supraconductori, face o scurtă introducere în ceea ce este un qubit și cum se diferențiază acesta de un bit clasic. Este prezentat fenomenul de superpoziție cuantică și de paralelism cuantic. Sunt menționate criteriile DiVincenzo și etapele de dezvoltare a unui sistem cuantic spre a putea face calcule cuantice tolerante la erori, precum și stagiul de dezvoltare în care se află qubiții bazați pe circuite supraconductoare.

În capitolul 3 sau Partea teoretică am prezentat baza teoretică a analizei de circuit și ipotezele simplificatoare pe care le-am folosit. În cadrul modelului teoretic prezentat am calculat hamiltonianul unui circuit simplu LC oscilant și hamiltonianul circuitului supraconductor cu superinductanță. Având forma hamiltonianului, am descris procedeul utilizat pentru transformarea acestuia într-o matrice, folosindu-ne de operatorii de creare și anihilare din cadrul celei de a doua cuantificări și de matricile acestora, prezentând și modalitatea lor de construcție.

Având forma matriceală a hamiltonianului, capitolul 4, Partea aplicativă, descrie algoritmi folosiți pentru simularea funcțiilor de undă ale sistemului, a potențialului și a nivelelor energetice de interes. Am prezentat construcția hamiltonianului matriceal și crearea cosinului unei matrici prin descompunere spectrală. Sunt discutate influența energiei E_J asupra potențialului și sunt reprezentate energiile pentru un set de parametri realiști care au deja implementări fizice.

1 Calculatoarele și calculul cuantic

1.1 Simularea cu ajutorul calculatoarelor cuantice

Richard Feynman a rămas în istorie pentru numeroasele lui contribuții aduse științei și pentru interesele lui intelectuale largi și vaste. Calculul programat, dar și fizica cuantică, se găseau în mulțimea de subiecte care l-au captivat, motiv pentru care, acum aproximativ patruzeci de ani, a propus valorificarea proprietăților intrinsec cuantice ale materiei în scopul construirii unui nou tip de computer mai puternic. Realizarea viziunii lui Feynman este una dintre cele mai mari provocări cu care se confruntă știința și tehnologia în secolul XXI [1].

În cadrul primei ediții a Conferinței despre Fizică și Calcul, organizată de Institutul de Tehnologie din Massachusetts în mai 1981, Feynman este invitat să țină discursul principal despre „Simularea fizicii cu ajutorul calculatoarelor” [2]. Personalități importante din domeniu precum Dr. Freeman Dyson și Dr. John Archibald Wheeler sunt prezenți. Imaginea grupului de participanți se regăsește în Figura 1.1.



Figura 1.1: Participanții primei ediții a Conferinței despre Fizică și Calcul, organizată la Eddicott House MIT, 6-8 Mai, 1981 [3]

Din discursul susținut de Feynman se desprind câteva puncte cheie. În primul rând, Feynman evidențiază rolul inerent al simulării în folosirea fizicii în tehnologie sau alte arii ale științei. O astfel de simulare ar trebui să îndeplinească, după Feynman, trei criterii: exactitatea (calculatorul trebuie să se comporte precum natura, fără aproximații), dimensiunea liniară (numărul de elemente de calculator necesare pentru simularea unui sistem fizic să fie proporțional cu volumul sistemului de simulat) și localitatea (calculatorul nu trebuie să se supună timpilor de propagare a informației). Niciunul din cele trei criterii necesare unei astfel de simulări nu poate fi îndeplinit de un computer clasic.

Singura soluție la problema adusă în fața publicului de către Feynman este atunci un nou tip de computer, intrinsec diferit de cel clasic. Dacă natura nu este clasică, spune Feynman, și ne dorim să construim o simulare a acesteia, ar trebui să o facem pe baza unor calculatoare cuantice.

Este greu de minimalizat impactul pe care computerele cuantice capabile de simulare l-ar avea asupra întregii lumi științifice. Simularea cuantică și-ar găsi aplicații în studiul

multor probleme moderne de fizica stării condensate [4], fizica energiei înalte [5] [6], fizică atomică [7], chimie cuantică [8] și cosmologie [9].

Simularea cuantică nu ar oferi doar perspective noi în legătură cu fenomene deja cunoscute, dar ar putea să conducă la descoperirea unor noi fenomene cuantice. Pe lângă toate acestea, calculatorul și simularea cuantică ar putea deveni o unealtă indispensabilă în testarea noilor teorii și prezicerea comportamentului unor sisteme deja cunoscute în diverse condiții, accesând regimuri fizice care sunt în prezent dincolo de capacitățile experimentale.

Istoric vorbind, putem spune că revoluția calculatoarelor cuantice a început în anul 1980, odată cu articolul publicat de Paul Benioff în care acesta a propus un model cuantic al unei mașini Turing [10]. Tot în acea perioadă, nume importante din fizica teoretică, precum cele ale lui Richard Feynmann și Yuri Manin, încep să-și îndrepte atenția spre calculatoarele cuantice și să publice în acest domeniu. Astfel, cei trei reușesc să pună temeiul unui nou domeniu, construind o fundație pentru toată munca de cercetare care i-a urmat.

1.2 De la mașinile cuantice Turing la era NISQ

La doar patru ani după discursul lui Feynman, în 1985, David Deutsch a propus modelul unui computer cuantic în lucrarea sa fundamentală, „Teoria cuantică, principiul Church-Turing și computerul cuantic universal” [11], formalizând noțiunea de calculator cuantic. Pe lângă acest avans important este ridicată întrebarea dacă computerul cuantic ar putea avea un avantaj față de computerul clasic și la rezolvarea problemelor care nu au legătură cu fizica cuantică.

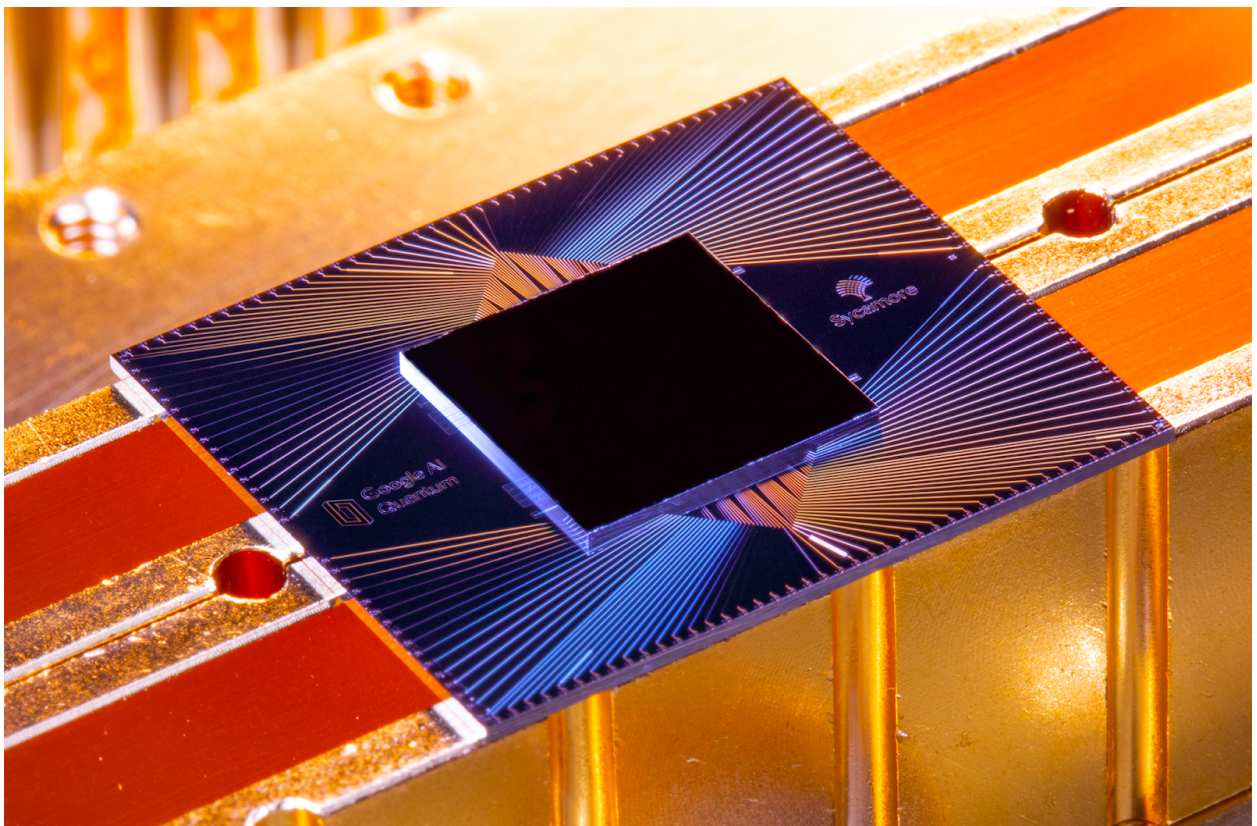


Figura 1.2: Procesorul Sycamore al grupului Google A.I. Quantum [12]

A urmat o perioadă de aproximativ 15 ani, în care majoritatea articolelor din domeniu

se concentrau pe formularea de probleme pe care un computer cuantic le-ar rezolva cu o accelerare exponențială față de cel clasic. Contribuții majore au fost aduse de Ethan Bernstein și Umesh Vazirani în articolul lor „Teoria complexității cuantice” [13] și de Daniel Simon în articolul său „Despre puterea calculului cuantic” [14]. Anii care au urmat acestor articole au văzut mult interes și atenție dedicată algoritmilor cuantici. Interesul pentru acest domeniu a crescut însă exponențial după publicațiile lui Peter Shor în legătură cu factorizarea [15] și reducerea decoerenței în memoriile cuantice [16] și cel al lui Lov Grover despre descoperirea algoritmului său de căutare [17].

În prezent ne aflăm în ceea ce oamenii domeniului calculatoarelor cuantice numesc era NISQ [18]. Acronimul NISQ vine din engleză de la Noisy Intermediate-Scale Quantum și se traduce prin era cuantică zgomotoasă la scară intermediară. Această eră este caracterizată de calculatoare cuantice cu un număr de 50-100 de qubiți, care ar fi capabile să execute sarcini care depășesc capacitățile calculatoarelor clasice din zilele de azi, dar circuitele lor cuantice vor rămâne în continuare limitate din punct de vedere dimensional de zgomot. Era NISQ va produce calculatoare foarte utile care ne vor ajuta să studiem probleme complexe din domeniul fizicii cuantice și nu numai. Însă acestea vor avea multe neajunsuri și limitări, fiind doar un pas spre o tehnologie a calculatoarelor cuantice mai puternică, care ne-ar permite un control mai precis asupra qubiților.

1.2.1 Prima demonstrație a supremației cuantice, chip-ul Sycamore

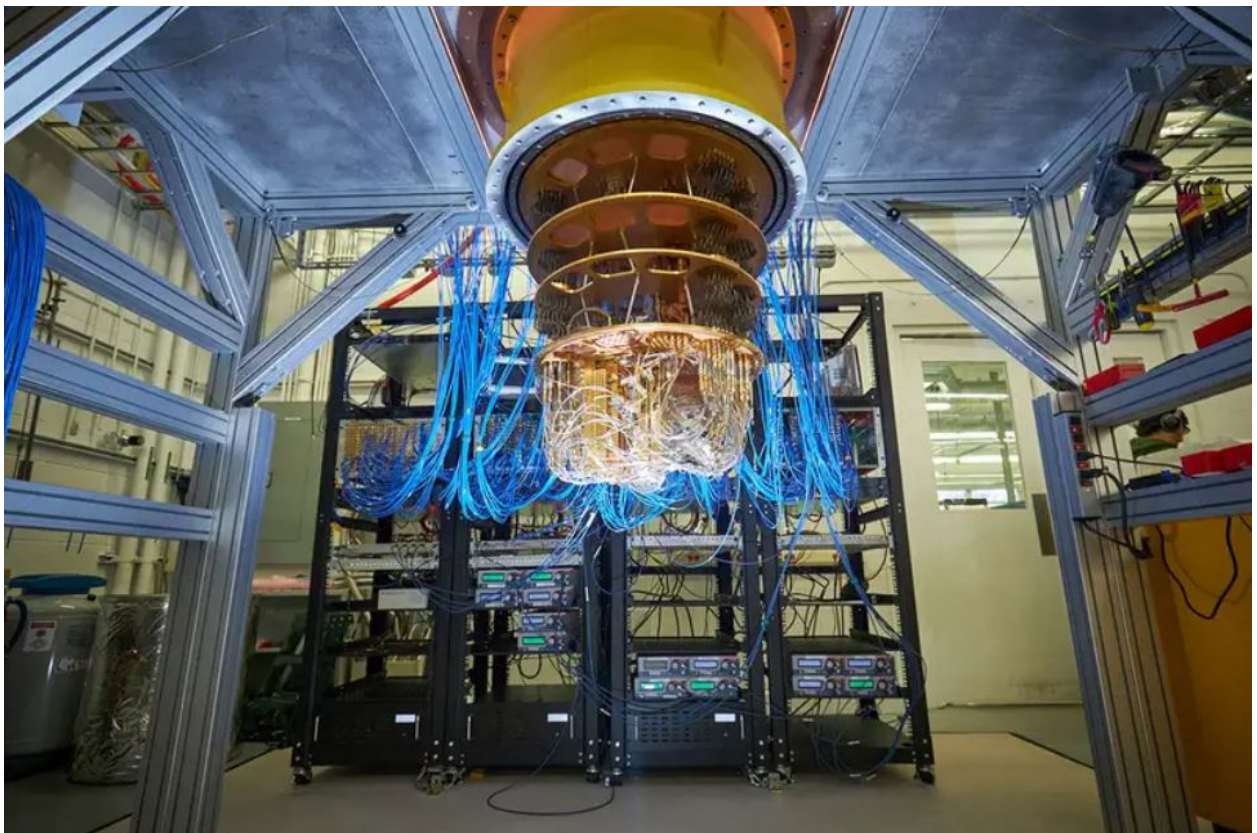


Figura 1.3: Calculatorul cuantic Sycamore construit de grupul A.I. Quantum [19]

Având capacități de calcul superioare supercomputerelor clasice, calculatoarele cuantice din era NISQ au trezit și interesul comunității de business. Așadar, în cei 40 de ani de

la discursul lui Feynman, amploarea aplicațiilor de calculatoare cuantice s-a extins până la cele care vorbesc unui public larg: securitatea informatică, optimizarea de calcul, machine learning etc. Aceste aplicații au atras din ce în ce mai mult atenția gigantilor industriali precum Google, Microsoft, IBM, Intel sau Lockheed Martin.

Printre cele mai semnificative progrese în domeniu se numără rezultatul obținut de Google AI Quantum Group prin calculatorul lor cuantic programabil Sycamore. Folosindu-se de 53 de qubiți supraconductori, grupul de cercetători de la Google a construit chip-ul sub forma unei matrice bidimensionale aranjate pe mai multe straturi. Folosindu-ne de cele mai avansate tehnici clasice, a simula ceea ce face Sycamore în câteva minute ar lua cel puțin câteva zile celui mai puternic supercomputer clasic existent [20]. Din aceste motive, grupul de cercetători de la Google a anunțat în 2019 că a ajuns cu și prin Sycamore la supremația cuantică [21], [22].

Supercomputerele clasice pe care Sycamore le-a depășit ca timp prezintă cerințe imense de spațiu și consum energetic, apropiindu-se de sute de metri pătrați și megawați de putere. Sycamore este de dimensiunea unui chip, răcit de un frigider de diluție. În Figura 1.2 este ilustrat procesorul calculatorului, iar în Figura 1.3 se găsește o imagine a frigiderului de diluție în care funcționează Sycamore.

1.2.2 Ajungând la trei cifre, chip-ul Eagle

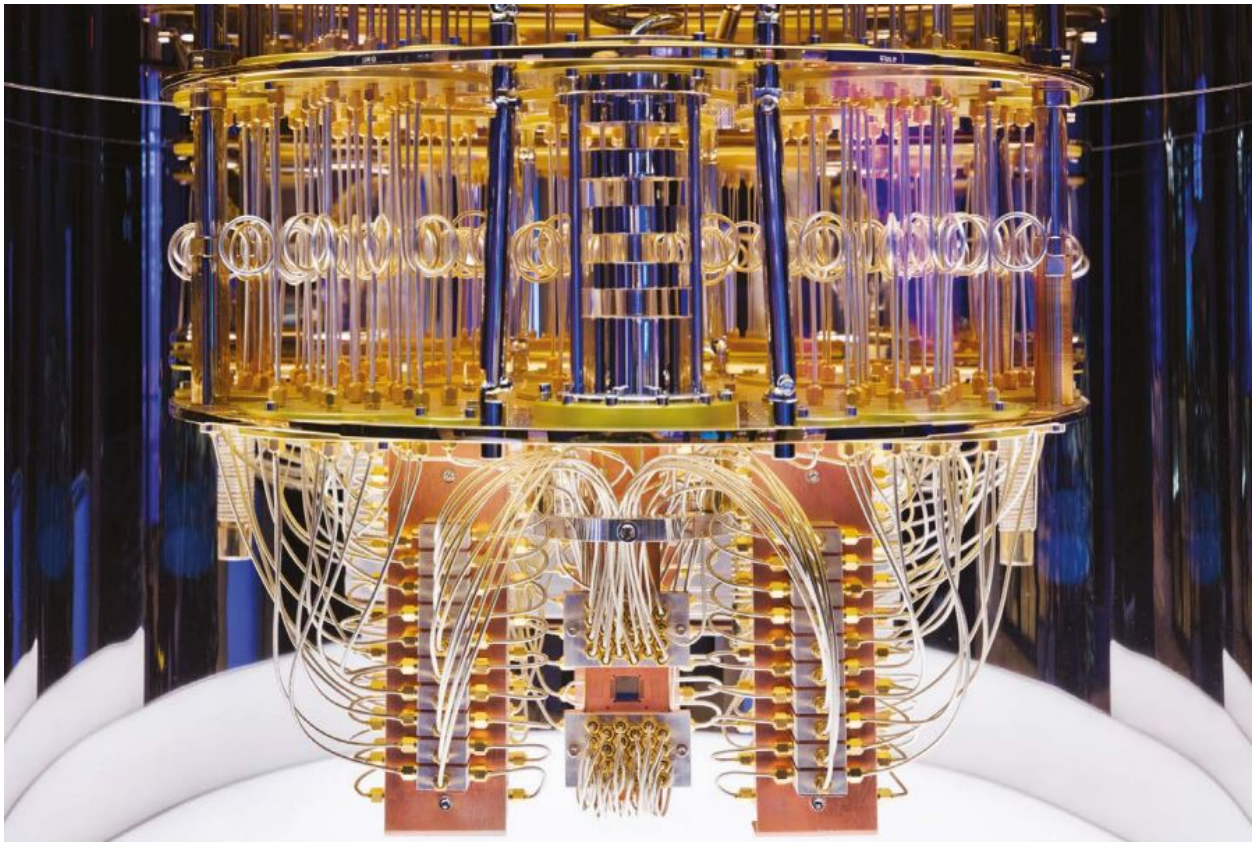


Figura 1.4: Calculatorul cuantic Eagle construit de IBM Quantum [23]

Încă din 2019 laboratoarele IBM își anunțau planurile ambițioase de a ajunge în 2024 la un calculator cuantic capabil să încorporeze 1121 qubiți [24]. De atunci, echipa IBM Quantum

a reușit să-și îndeplinească obiectivele deja propuse prin lansarea în 2019 a calculatorului cuantic cu 27 de qubiți Falcon, în 2020 a calculatorului cuantic cu 65 de qubiți Hummingbird și cel mai recent, în noiembrie 2021, a calculatorului cuantic cu 127 qubiți Eagle [25]. O imagine a materialului grafic care prezintă drumul de dezvoltare al echipei se găsește în Figura 1.5.

Chip-ul Eagle este pentru IBM un pas spre crearea în 2022 a calculatorului cuantic cu 433 qubiți Osprey, succedat la rândul lui de Condor, care ar avea 1121 qubiți. Este primul chip cuantic complet programabil din lume al cărui număr de qubiți a trecut de 100 [23].

Asemenea predecesorilor săi, chip-ul Eagle se bazează pe o arhitectură de qubiți supraconductori de tip transmon. Acești qubiți sunt oscilatori anarmonici, a căror anarmonicitate este datorată prezenței unei joncțiuni Josephson. Această joncțiune este un element hardware care constă într-o barieră plasată între două fire supraconductoare. Joncțiunile Josephson funcționează precum niște inductori neliniari [26].

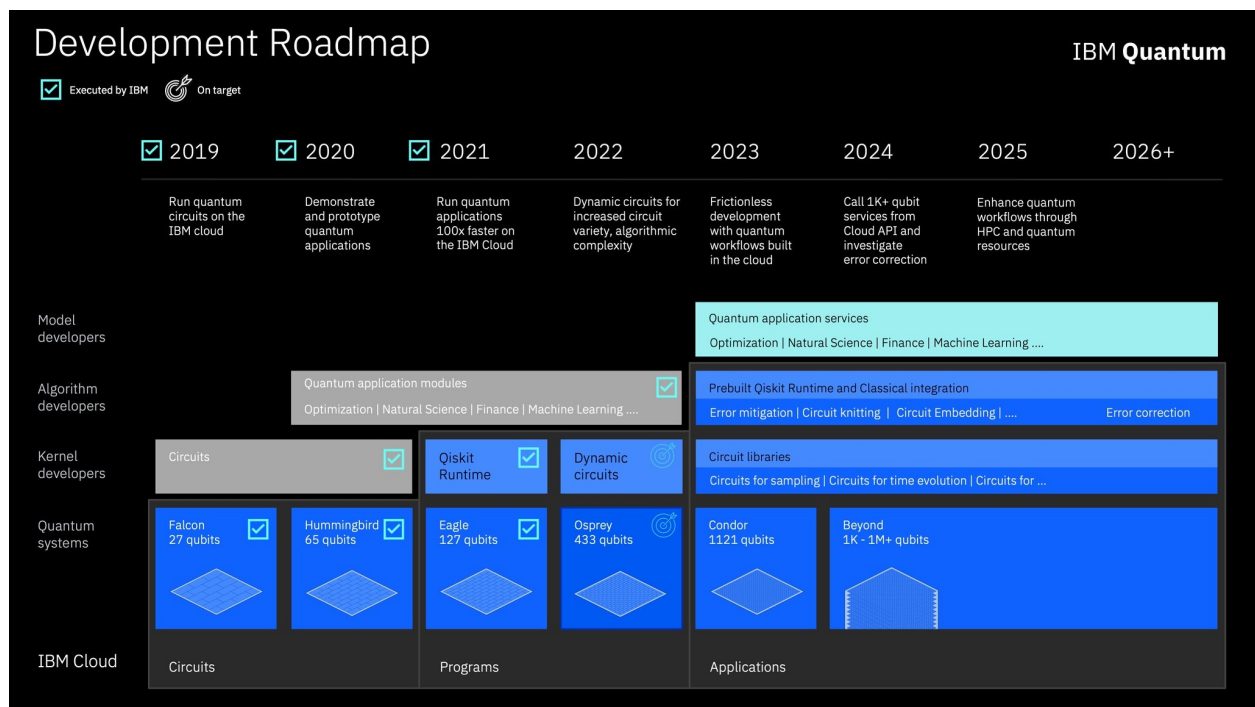


Figura 1.5: Drumul de dezvoltare propus de IBM Quantum din 2019 pana în 2025 [27]

Echipele IBM Quantum se concentrează atât pe scalarea în dimensiune (creșterea numărului de qubiți) a calculatoarelor lor cuantice, cât și pe dezvoltarea calității și vitezei lor. Din acest punct de vedere, echipa a progresat într-atât încât au reușit să măsoare timpi de coerență de peste 400 de microsecunde pe cele mai performante procesoare ale lor [28].

2 Qubiți supraconductori

2.1 Bitul unui calculator cuantic

Qubitul sau bitul cuantic este echivalentul bitului dintr-un calculator clasic, fiind cel mai mic ansamblu purtător de informație. Qubitul poate fi atât în stare fundamentală, cât și în stare excitată, în același timp, precum ilustrează Figura 2.1. Cele două stări logice ale fiecărui qubit trebuie mapate pe stările proprii ale unui sistem fizic cuantic cu două stări.

Un sistem fizic cuantic cu două stări este un sistem care poate exista în orice superpoziție a două stări cuantice care se pot distinge fizic. Cel mai simplu exemplu de sisteme cu două stări este spinul fermionilor. Un qubit de spin se bazează pe un grad de libertate de natură electronică sau nucleară.

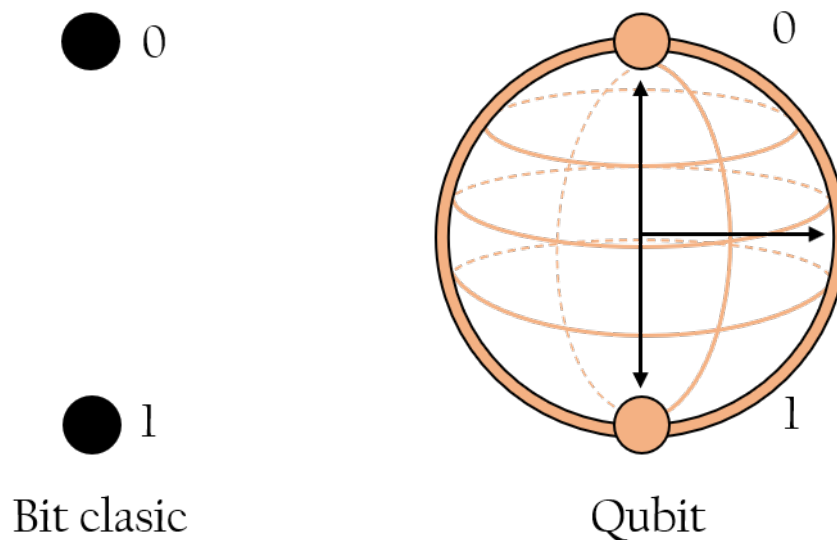


Figura 2.1: Diferența dintre un bit clasic și unul cuantic [29]

Pe lângă qubiții care se bazează pe două nivele de spin se pot găsi mai multe sisteme fizice cuantice cu două stări care să poată fi implementate ca qubiți într-un calculator cuantic. Printre acestea se numără: două stări de polarizare diferite alese pentru un foton, două nivele de energie ale unui electron care orbitează un singur atom etc. [30].

2.2 Superpoziția cuantică

În cazul în care considerăm un sistem cu două stări de bază $|0\rangle$ și $|1\rangle$ ne putem imagina un bit clasic ca un singur atom care se află într-una din cele două stări notate cu $|0\rangle$ și $|1\rangle$. În schimb, unei stări cuantice îi lipsește această delimitare și ea se poate imagina ca o stare continuă între $|0\rangle$ și $|1\rangle$ până la momentul măsurării, când rezultatul va fi doar $|0\rangle$ sau $|1\rangle$. Așadar, starea cuantică a unui qubit poate fi scrisă astfel:

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle, \quad (1)$$

unde α_0 și α_1 sunt amplitudini complexe. Probabilitatea ca qubitul să se afle atunci în starea $|0\rangle$ este $|\alpha_0|^2$, iar pentru starea $|1\rangle$ e $|\alpha_1|^2$, unde

$$|\alpha_0|^2 + |\alpha_1|^2 = 1 \quad (2)$$

Într-un sistem în care avem un qubit avem superpoziție de două stări clasice. Pentru un sistem cu doi qubiți vom avea superpoziție între patru stări clasice, iar pentru trei qubiți sistemul va putea descrie o superpoziție de opt stări clasice. Se poate observa acum natura explozivă a memoriei cuantice, unde, adăugând un singur qubit sistemului, spațiul stărilor disponibile se dublează. Pentru un sistem de n qubiți $|0\rangle, |1\rangle, |2\rangle \dots |n-1\rangle$ spațiul stărilor clasice necesare pentru a descrie superpoziția va avea dimensiunea 2^n datorită superpoziției cuantice.

2.3 Registre cuantice

Starea de natură cuantică a doi sau mai mulți qubiți se poate descrie cel mai ușor dacă aceștia sunt aranjați și tratați împreună, ca un singur registru. Pentru un registru de n qubiți, starea $|\psi\rangle$ a întregului sistem este descrisă de un vector aflat în spațiul produsului 2^n dimensional $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \dots \otimes \mathcal{H}_n$ al spațiilor Hilbert $\mathcal{H}_i (i = 1, \dots, n)$ asociate fiecărui qubit [31].

În general, starea $|\psi\rangle$ a unui registru cuantic care descrie un sistem cu n qubiți poate fi descrisă de:

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \dots + \alpha_{2^n-1} |2^n - 1\rangle$$

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle = \alpha_0 \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \alpha_1 \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \dots + \alpha_{2^n-1} \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \quad (3)$$

Se poate observa la acest pas creșterea exponențială a numărului stărilor de bază (2^n) odată cu adăugarea unui singur qubit la sistem.

Ecuatiile 3 ilustrează un fenomen relevant registrelor cuantice: paralelismul cuantic. Ne putem gândi la acesta ca la capacitatea unui calculator cuantic de a procesa un calcul cu cele 2^n valori în mod simultan. Dacă registrul cuantic se află într-o superpoziție egală a tuturor stărilor ($\alpha_i = 1/\sqrt{2^n}$, pentru oricare i), un calculator cuantic ar putea executa un calcul pentru toate cele 2^n valori, simultan.

2.4 Qubiți supraconductori pentru calculatoarele cuantice

În dezvoltarea unui computer cuantic există mai mulți pași și stagii care se suprapun și se influențează reciproc [32]. În primul rând, sistemul cuantic proiectat spre a deveni un computer cuantic trebuie să poată păstra un qubit în superpoziție suficient timp pentru a putea fi inițializat, manipulat și citit. Al doilea pas este realizarea de algoritmi simpli pe un singur qubit. Pentru ca un sistem să poată finaliza cei doi pași trebuie să îndeplinească primele cinci criterii DiVincenzo, propuse în anul 2000 de fizicianul teoretician de origine americană David P. DiVincenzo [33].

Criteriile DiVincenzo constă în șapte condiții pe care o configurație experimentală trebuie să le îndeplinească cu succes pentru a putea rula algoritmi cuantici. Dintre acestea, primele 5

se referă la calculul cuantic în sine, iar ultimele două vizează comunicarea cuantică. Criteriile necesare pentru implementarea calculului cuantice sunt sumarizate în următoarele cerințe.

1. Sistemul fizic să fie scalabil și să aibă qubiți bine caracterizați;
2. Sistemul să fie capabil să inițializeze starea qubiților;
3. Timpii de coerență să fie mai lungi decât timpii de funcționare ai porților cuantice;
4. Să existe un set „universal” de porți cuantice;
5. Să existe posibilitatea de măsurare a qubitului de interes.

Următoarele etape de dezvoltare ale calculatorului cuantic necesită ceea ce în literatură se numește Quantum Error Corection (QEC) și se traduce drept corectarea de eroare cuantică. Al patrulea pas, după M.H. Devoret și R.J. Schoelkopf, ar fi dezvoltarea unei memorii cuantice capabile să păstreze informația cuantică un timp semnificativ mai lung decât timpii de decoerență. Ultimele două etape spre calculul cuantic tolerant la erori sunt capacitatea de a face toate operațiile de către un singur qubit pe un singur qubit logic și capacitatea de a efectua operații logice între mai mulți qubiți logici (qubiți protejați de mecanisme active de corectare a erorilor).

Diverse sisteme proiectate spre a fi qubiți (precum atomii Rydberg și ionii prinși) au ajuns la diferite etape de proiectare a calculatorului cuantic universal, însă niciunul nu a ajuns atât de departe precum cele bazate pe circuite supraconductoare.

În prezent, qubiții bazați pe circuite supraconductoare cu joncțiuni Josephson sunt cel mai avansat model de qubit care poate fi exploatat spre construcția calculatorului cuantic. Fiind bazați pe circuite electrice la scară micrometrică, aceștia pot fi imprimați pe cipuri în număr mare, conectați unul la altul, manipulați în scopul rulării algoritmilor cuantici și citiți cu precizie.

Operarea pe qubiți supraconductori se bazează pe două fenomene complexe și robuste: superconductivitatea și efectul Josephson. Superconductivitatea sau capacitatea anumitor materiale de a conduce curent electric fără rezistență facilitează apariția efectelor cuantice, precum efectul Josephson. Efectul Josephson, evidențiat în joncțiunea Josephson, este responsabil pentru introducerea neliniarității în circuit fără a introduce disipare sau defazare, fiind crucial în funcționarea circuitului precum un qubit [34].

3 Partea teoretică

3.1 Analiza de circuit

Pentru analiza qubiților bazați pe circuite supraconductoare bazați pe superinductanțe ne interesează să exprimăm relațiile dintre elementele de circuit și modul în care acestea se influențează reciproc.

Am folosit presupunerea simplificatoare că atributele de interes ale circuitului, precum rezistența, capacitatea, inductanța etc., sunt concentrate în componente electrice idealizate: rezistențe, condensatoare, inductoare unite între ele printr-o rețea de fire perfect conductoare. Aceste presupuneri folosite în analiza de rețele poartă numele de modelul elementelor concentrate al circuitelor electronice. Diagrama unui astfel de element este ilustrată în Figura 3.1.

De asemenea, circuitul de interes este supraconductor, neavând rezistență electrică și tot câmpul magnetic fiind expulzat din fire prin efect Meissner. Având un sistem cuantic închis, vom ignora toate pierderile din interacțiunea sistemului cu mediul exterior.

Pentru analiza circuitului qubitului supraconductor ne interesează să separăm toate componentele de circuit și să aflăm tensiunea de pe ele și curentul ce le străbate în funcție de timp $V(t), I(t)$.

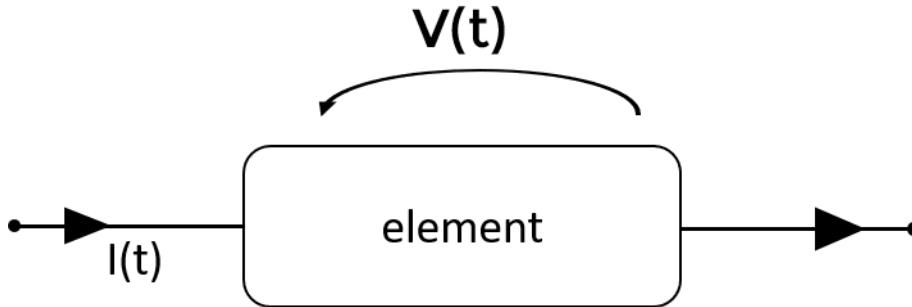


Figura 3.1: Un element de circuit străbătut de o tensiune $V(t)$ prin care trece un curent $I(t)$

Cele două mărimi sunt descrise de ecuațiile:

$$V(t) = \int \mathbf{E}(t) \cdot d\mathbf{l} \quad (4)$$

$$I(t) = \frac{1}{\mu_0} \oint \mathbf{B}(t) \cdot d\mathbf{l} \quad (5)$$

Asupra aceluiași element de circuit ilustrat anterior se pot defini fluxul Φ și încărcătura electrică Q astfel:

$$\Phi(t) = \int_{-\infty}^t V(\tau) d\tau \quad (6)$$

$$\frac{d}{dt} Q(t) = I(t) \quad (7)$$

$$Q(t) = \int_{-\infty}^t I(\tau) d\tau \quad (8)$$

unde s-a presupus că sistemul este în repaus la $\tau = -\infty$ [35].

3.2 Circuitul LC oscilant

Pentru studiul unui circuit LC ne interesează să găsim relațiile de legătură între cele patru mărimi care pot descrie complet comportamentul rezonatorului. Acestea sunt diferența de potențial $V(t)$, intensitatea curentului $I(t)$, sarcina electrică $Q(t)$ și fluxul magnetic $\Phi(t)$.

Reprezentarea grafică a unui astfel de circuit se găsește la Figura 3.2.

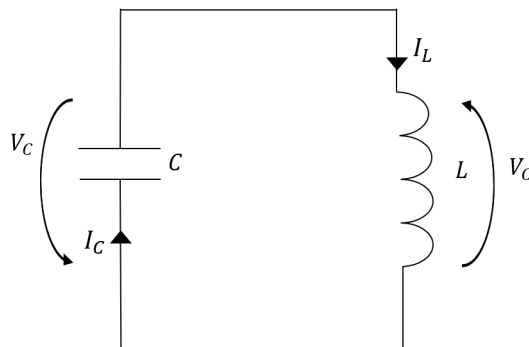


Figura 3.2: Reprezentarea schematică a circuitului simplu LC oscilant alcătuit dintr-un capacitor de capacitanță C legat în serie cu un inductor de inductanță L

Pentru un capacitor general tensiunea care îl străbate $V(t)$ este direct proporțională cu sarcina stocată pe plăcile condensatorului $Q(t)$ printr-un factor de proporționalitate egal cu inversul capacității, $1/C$. Considerând că tensiunea este derivata fluxului, ecuația devine:

$$\dot{\Phi}(t) = V(t) = \frac{1}{C}Q(t) \quad (9)$$

Pentru un inductor general, intensitatea curentului care îl parcurge $I(t)$ este direct proporțională cu fluxul magnetic $\Phi(t)$, având în acest caz factorul de proporționalitate $1/L$, unde L este inductanța caracteristică inductorului.

$$\dot{Q}(t) = I(t) = \frac{1}{L}\Phi(t) \quad (10)$$

Pentru relația dintre intensitate și flux ne vom folosi de ecuația 8 și 9 cu ajutorul cărora vom putea rescrie relațiile:

$$\begin{aligned} Q(t) &= CV(t) = C\dot{\Phi} \\ I(t) &= \dot{Q}(t) = C\ddot{\Phi} \end{aligned} \quad (11)$$

Din combinarea ecuațiilor (10) și (11) și aplicarea legilor lui Kirchhoff asupra circuitului ajungem la o descriere a comportamentului întregului sistem în funcție de fluxul magnetic $\Phi(t)$:

$$\ddot{\Phi}(t) = -\frac{1}{LC}\Phi(t) \quad (12)$$

Se observă din această ecuație că sistemul are un comportament armonic oscilant. Dacă analizăm comportamentul celor două elemente înseriate, observăm că la legarea unui inductor cu un condensator încărcat acesta va începe să se descarce, iar curentul condus prin inductor va genera un câmp magnetic în jurul inductorului. După descărcarea completă a condensatorului, energia stocată în câmpul magnetic al bobinei va induce o tensiune de polarizare inversă, care se va opune schimbării de curent, conducând la reîncărcarea condensatorului descărcat.

Așadar, sarcina va curge înainte și înapoi în circuit iar energia va oscila și ea între condensator și inductor. În circuitele reale există o rezistență internă care va conduce la amortizarea și stingerea oscilațiilor dacă acestea nu sunt întreținute de un circuit extern.

Oscilând între cele două elemente de circuit, energia se transformă în cadrul circuitului LC simplu din energie electrică caracteristică capacitorului C în energie magnetică caracteristică inductorului L . Energia instantanee, dependentă de timp, din fiecare element este derivată din curentul care îl parcurge $I(t)$ și tensiunea de peste acesta $V(t)$ astfel:

$$E(t) = \int_{-\infty}^t V(\tau)I(\tau)d\tau \quad (13)$$

Pentru capacitor, energia descrisă de formula (13) are forma unei energii cinetice:

$$E_{\text{capacitor}} = \frac{1}{2}C\dot{\Phi}^2(t) \quad (14)$$

iar pentru inductor, aceasta seamănă cu o energie potențială:

$$E_{\text{inductor}} = \frac{1}{2L}\Phi^2(t) \quad (15)$$

3.2.1 Hamiltonianul circuitului LC oscilant

Pentru exprimarea hamiltonianului circuitului oscilant LC ne vom folosi de formularea Hamilton-Lagrange din mecanica clasică, unde vom reprezenta elementele de circuit în funcție de una din coordonatele generalizate ale circuitului: sarcină sau flux [36].

Din ecuațiile (14) și (15) putem să construim un Lagrangian pentru sistem, definit ca diferența dintre energia cinetică și energia potențială:

$$\mathcal{L} = E_{\text{cinetică}} - U = E_{\text{capacitor}} - E_{\text{inductor}} = \frac{1}{2}C\dot{\Phi}^2(t) - \frac{1}{2L}\Phi^2(t) \quad (16)$$

Folosind transformarea Legendre putem exprima Hamiltonianul sistemului, cunoscând-i momentul conjugat, care în acest caz este sarcina de pe capacitor.

$$\frac{\partial \mathcal{L}}{\partial \dot{\Phi}} = C\dot{\Phi} = Q \quad (17)$$

Exprimarea clasică a Hamiltonianului circuitului LC are forma:

$$H = Q\dot{\Phi} - \mathcal{L} = \frac{Q^2}{2C} + \frac{\Phi^2}{2L} \quad (18)$$

Pentru exprimarea Hamiltonianului cuantic al circuitului LC avem nevoie de patru mărimi noi: fluxul redus ϕ , sarcina redusă n , energia de încărcare E_C și energia inductivă E_L .

Expresia fluxului redus este:

$$\phi = 2\pi \frac{\Phi}{\Phi_0} \quad (19)$$

unde ϕ_0 este cuanta de flux magnetic supraconductor:

$$\Phi_0 = \frac{h}{2e} \quad (20)$$

Expresia sarcinii reduse este:

$$n = \frac{Q}{2e} \quad (21)$$

Energia de încărcare necesară adăugării fiecărui electron al perechii Cooper pe insulă are forma:

$$E_C = \frac{e^2}{2C} \quad (22)$$

iar energia inductivă este exprimată astfel:

$$E_L = \frac{\Phi_0^2}{4\pi^2 L} \quad (23)$$

Folosindu-ne de mărimile definite mai sus, Hamiltonianul cuantic al circuitului LC are forma:

$$H = 4E_C n^2 + \frac{1}{2} E_L \phi^2 \quad (24)$$

Hamiltonianul circuitului LC cuantic este identic cu cel al oscilatorului armonic cuantic.

3.3 Circuitul qubitului supraconductor cu superinductanță

Circuitul LC clasic este compus doar din elemente care au un răspuns curent-tensiune liniar. Din acest motiv, circuitul armonic LC nu poate fi folosit pe post de qubit, deoarece spațierea energetică dintre stările lui $|0\rangle, |1\rangle, |2\rangle \dots, |n\rangle$ este echidistantă și deci nu va permite adresarea și controlul doar anumitor tranziții de interes precum $|0\rangle \leftrightarrow |1\rangle$.

Așadar, este necesară adăugarea unui element nonliniar circuitului supraconductor. Aceasta a permis crearea de circuite cu nivele energetice neechidistante, care pot fi excitate selectiv de frecvențe radio bine stabilite. Un astfel de element este joncțiunea Josephson, care transformă circuitul într-un adevărat atom artificial.

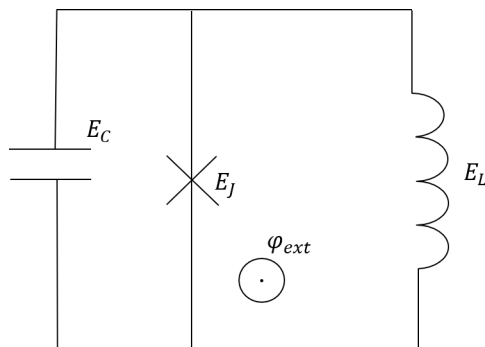


Figura 3.3: Circuitul echivalent al fluxonului alcătuit dintr-o joncțiune Josephson cu energie E_J , un superinductor de energie E_L și un capacitor de energie E_C

3.3.1 Joncțiunea Josephson

Efectul Josephson, observat la interfața dintre două insule supraconductoare separate de un izolator subțire, a fost prima dată teoretizat în 1962 de către Brian David Josephson [37]. Acesta a explicat efectul pe baza tunelării cuantice a perechilor Cooper. Curentul produs sau rata de tunelare, precum și tensiunea dintre cele două insule supraconductoare, depind de diferența de fază dintre funcțiile de undă ale supraconductorilor ϕ [38].

$$I(t) = I_c \sin \phi(t) \quad (25)$$

$$V(t) = \frac{\hbar}{2e} \frac{\partial \phi}{\partial t} \quad (26)$$

În formulele (25) și (26) prezentate mai sus, I_C este o constantă experimentală numită curentul critic al joncțiunii, iar $\hbar/2e$ este cuanta de flux magnetic.

În cazul qubiților bazați pe circuite supraconductoare cu superinductanțe, joncțiunea Josephson este șuntată de un inductor mare cu impedanță caracteristică mare, numit superinductor [39]. Circuitul echivalent acestui qubit este reprezentat în Figura 3.3 [40].

3.3.2 Hamiltonianul Fluxonului

Pentru a obține expresia Hamiltonianului fluxonului vom considera pentru început elementul neliniar, joncțiunea Josephson. Acesta are o energie caracteristică în funcție de flux dată de formula:

$$E(\Phi) = E_J(1 - \cos \varphi) \quad (27)$$

unde φ este faza, proporțională cu fluxul pe joncțiune, iar E_J este scara de energie egală cu $(\hbar I_C)/2e$, unde I_C este curentul critic pe joncțiune.

$$\varphi = 2\pi \frac{\Phi}{\Phi_0} \quad (28)$$

$$E(\Phi) = E_J(1 - \cos 2\pi \frac{\Phi}{\Phi_0}) \quad (29)$$

Elementul inductiv L are o energie caracteristică

$$E_L(\Phi) = \frac{\Phi^2}{2L} \quad (30)$$

Mai departe, am aflat Lagrangianul cu scopul de a-i afla transformata Legendre. Pentru fluxonul reprezentat în Figura 3.3, acesta are forma:

$$\mathcal{L}(\Phi_J, \dot{\Phi}_J) = \frac{C\dot{\Phi}_J^2}{2} - \frac{\Phi_L^2}{2L} - E_J(1 - \cos 2\pi \frac{\Phi_J}{\Phi_0}) \quad (31)$$

Inversa transformatei Legendre transformă Lagrangianul într-o funcție $H(Q_J, \Phi_J)$ care este hamiltonianul căutat:

$$H = Q_J \dot{\Phi}_J - \mathcal{L} \quad (32)$$

unde Q_J este sarcina electrică, care joacă rol de impuls generalizat pentru variabila flux.

$$Q_J = \frac{\partial \mathcal{L}}{\partial \dot{\Phi}_J} = C \dot{\Phi}_J \quad (33)$$

Hamiltonianul poate fi exprimat atunci cu ajutorul expresiei de mai sus în funcție de Q_J și Φ_J astfel:

$$H(Q_J, \Phi_J) = \frac{Q_J^2}{2C} + \frac{\Phi_J^2}{2L} + E_J(1 - \cos 2\pi \frac{\Phi_J}{\Phi_0}) \quad (34)$$

Exprimând acest Hamiltonian în funcție de \hat{n} și $\hat{\varphi}$, unde \hat{n} este definit ca

$$\hat{n} = -i \frac{\partial}{\partial \hat{\varphi}} \quad (35)$$

obținem expresia

$$H(n, \varphi) = \frac{2e^2 n^2}{C} + \frac{\Phi_0^2 \varphi^2}{8\pi^2 L} + E_J(1 - \cos \varphi) \quad (36)$$

Considerând mai departe

$$E_C = \frac{e^2}{2C} \quad (37)$$

$$E_L = \frac{\Phi_0^2}{4\pi^2 L} \quad (38)$$

Hamiltonianul devine

$$H(n, \varphi) = 4E_C n^2 + \frac{1}{2} E_L \varphi^2 + E_J(1 - \cos \varphi) \quad (39)$$

Considerând efectele unui flux exterior indus sistemului φ_{ext} , Hamiltonianul devine:

$$\hat{H}(n, \varphi) = 4E_C \hat{n}^2 + \frac{1}{2} E_L (\hat{\varphi} - \varphi_{ext})^2 - E_J \cos(\hat{\varphi} - \varphi_{ext}) \quad (40)$$

unde, datorită invarianței gabariturii, ne-am lipsit de constanta E_J .

3.4 Matricea hamiltonianului fluxonului

Deoarece ne interesează o formă ușor programabilă a Hamiltonianului circuitului supraconductor, căutăm expresia Hamiltonianului în cea de a doua cuantificare. În cadrul formalismului celei de a doua cuantificări, stările cuantice de interes sunt reprezentate în baza Fock, iar manipularea acestor stări se face prin operatorii de creare \hat{a}^\dagger și anihilare \hat{a} .

După ce ajungem la o formă a Hamiltonianului dependentă doar de variabilele cunoscute și de cele două matrici a și a^\dagger , construcția matricii Hamiltonianului se reduce la câteva operații algebrice de înmulțire și adunare cu matrici.

Introducerea celor doi operatori \hat{a}^\dagger și \hat{a} în formula Hamiltonianului se poate face folosind relațiile:

$$\hat{n} = -i n_{ZPF} (\hat{a} - \hat{a}^\dagger) \quad (41)$$

$$\hat{\varphi} = \varphi_{ZPF} (\hat{a} + \hat{a}^\dagger) \quad (42)$$

Noul nostru Hamiltonian ia forma

$$\hat{H} = -4n_{ZPF}^2 E_C (\hat{a} - \hat{a}^\dagger)^2 + \frac{1}{2} E_L [\varphi_{ZPF}(\hat{a} + \hat{a}^\dagger) - \varphi_{ext}]^2 - E_J \cos[\varphi_{ZPF}(\hat{a} + \hat{a}^\dagger) - \varphi_{ext}] \quad (43)$$

În expresia explicită a Hamiltonianului (43) apar mai mulți termeni compuși din operatorii a și a^\dagger a căror formă matriceală am aflat-o conform algoritmului prezentat mai jos.

3.4.1 Matricea $a^\dagger a$

Pentru a afla elementele matricii $a^\dagger a$ pentru un n aleator vom proceda astfel:

$$\begin{array}{ccccccc} & & |0\rangle & & |1\rangle & & \dots & & |n\rangle \\ \langle 0| & & \langle 0| a^\dagger a |0\rangle & & \langle 0| a^\dagger a |1\rangle & & & & \\ \langle 1| & & \langle 1| a^\dagger a |0\rangle & & \langle 1| a^\dagger a |1\rangle & & & & \\ \vdots & & & & & & \ddots & & \\ \langle n| & & & & & & & & \langle n| a^\dagger a |n\rangle \end{array}$$

Folosindu-ne de efectul operatorilor a și a^\dagger asupra unei stări n

$$\begin{aligned} a |n\rangle &= \sqrt{n} |n-1\rangle \\ a^\dagger |n\rangle &= \sqrt{n+1} |n+1\rangle \end{aligned} \quad (44)$$

putem calcula elementele matricii $a^\dagger a$

$$\begin{aligned} \langle 0| a^\dagger a |0\rangle &= 0 \\ \langle 0| a^\dagger a |1\rangle &= \sqrt{1} \langle 0| a^\dagger |0\rangle = 1 \langle 0|1\rangle = 0 \\ \langle 1| a^\dagger a |0\rangle &= 0 \\ \langle 1| a^\dagger a |1\rangle &= \sqrt{1} \langle 1| a^\dagger |0\rangle = 1 \langle 1|1\rangle = 1 \\ &\dots \\ \langle n| a^\dagger a |n\rangle &= \sqrt{n} \langle n| a^\dagger |n-1\rangle = n \langle n|n\rangle = n \end{aligned} \quad (45)$$

Forma generală a matricii $a^\dagger a$:

$$a^\dagger a = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & & n \end{pmatrix} \quad (46)$$

Pentru $n = 5$ vom avea

$$a^\dagger a = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 4 \end{pmatrix} \quad (47)$$

3.4.2 Matricea a

Aplicăm aceeași tehnică precum în cazul matricii $a^\dagger a$ pentru a afla elementele matricii a .

$$\begin{array}{cccccc}
 & |0\rangle & |1\rangle & |2\rangle & |3\rangle & \dots & |n\rangle \\
 |0\rangle & \langle 0|a|0\rangle & \langle 0|a|1\rangle & \langle 0|a|2\rangle & \langle 0|a|3\rangle & & \\
 |1\rangle & \langle 1|a|0\rangle & \langle 1|a|1\rangle & \langle 1|a|2\rangle & \langle 1|a|3\rangle & & \\
 |2\rangle & \langle 2|a|0\rangle & \langle 2|a|1\rangle & \langle 2|a|2\rangle & \langle 2|a|3\rangle & & \\
 |3\rangle & \langle 3|a|0\rangle & \langle 3|a|1\rangle & \langle 3|a|2\rangle & \langle 3|a|3\rangle & & \\
 \vdots & & & & & \ddots & \\
 |n\rangle & & & & & & \langle n|a|n\rangle
 \end{array}$$

Putem calcula elementele matricii a

$$\begin{aligned}
 \langle 0, 1, 2, \dots, n|a|0\rangle &= 0 \\
 &\vdots \\
 \langle 0|a|1\rangle &= \sqrt{1} \langle 0|0\rangle = \sqrt{1} \\
 \langle 1, 2, 3, \dots, n|a|1\rangle &= 0 \\
 &\vdots \\
 \langle 0|a|2\rangle &= \sqrt{2} \langle 0|1\rangle = 0 \\
 \langle 1|a|2\rangle &= \sqrt{2} \langle 1|1\rangle = \sqrt{2} \\
 \langle 2, 3, 4, \dots, n|a|2\rangle &= 0 \\
 &\vdots \\
 \langle n-1|a|n\rangle &= \sqrt{n} \langle n-1|n-1\rangle = \sqrt{n} \\
 \langle n|a|n\rangle &= \sqrt{n} \langle n|n-1\rangle = 0
 \end{aligned} \tag{48}$$

Forma generală a matricii a :

$$a = \begin{pmatrix} 0 & \sqrt{1} & 0 & 0 & \dots & 0 \\ 0 & 0 & \sqrt{2} & 0 & \dots & 0 \\ 0 & 0 & 0 & \sqrt{3} & \dots & 0 \\ 0 & 0 & 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \sqrt{n-1} \\ 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix} \tag{49}$$

Pentru $n = 5$ vom avea

$$a = \begin{pmatrix} 0 & \sqrt{1} & 0 & 0 & 0 \\ 0 & 0 & \sqrt{2} & 0 & 0 \\ 0 & 0 & 0 & \sqrt{3} & 0 \\ 0 & 0 & 0 & 0 & \sqrt{4} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{50}$$

3.4.3 Matricea a^\dagger

Putem calcula elementele matricii a^\dagger prin transpunerea conjugată a matricii a . Forma generală a matricii a^\dagger va fi

$$a^\dagger = \begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 & \dots \\ \sqrt{1} & 0 & 0 & 0 & \dots & 0 & \dots \\ 0 & \sqrt{2} & 0 & 0 & \dots & 0 & \dots \\ 0 & 0 & \sqrt{3} & 0 & \ddots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \dots & \dots \\ 0 & 0 & 0 & 0 & \sqrt{n-1} & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & \sqrt{n} & 0 \end{pmatrix} \quad (51)$$

Pentru $n = 5$ vom avea

$$a^\dagger = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \sqrt{1} & 0 & 0 & 0 & 0 \\ 0 & \sqrt{2} & 0 & 0 & 0 \\ 0 & 0 & \sqrt{3} & 0 & 0 \\ 0 & 0 & 0 & \sqrt{4} & 0 \end{pmatrix} \quad (52)$$

4 Partea aplicativă

4.1 Jupyter Notebook și Python 3.8

Proiectul Jupyter este un proiect open source dezvoltat de către comunitatea IPython în 2014, folosit pentru vizualizarea datelor științifice într-un mod interactiv și pentru programarea științifică în mai multe limbaje de programare.

Printre produsele dezvoltate de proiectul Jupyter se numără și Jupyter Notebook, JupyterHub și JupyterLab. În redactarea codului suport pentru această lucrare am folosit Jupyter Notebook.

Jupyter Notebook este un mediu de calcul web-based interactiv care crează documente de tip notebook, sau caiet. Un caiet sau notebook este un REPL (Read-Eval-Print Loop) bazat pe browser, alcătuit dintr-o înșiruire de celule ordonate de intrare și ieșire care conțin cod, text, matematică, diagrame sau chiar conținut media.

Limbaajul de programare pe care l-am folosit este Python pentru avantajele pe care acesta le aduce prin faptul că am putut folosi biblioteci standardizate precum NumPy, Matplotlib și SciPy. Acestea s-au dovedit foarte utile în studiul qubiților supraconductori.

Am folosit OpenSSH pentru conectarea la distanță la un server dedicat simulării cuantice al Institutului Național de Cercetare-Dezvoltare pentru Tehnologii Izotopice și Moleculare (INCDTIM).

4.2 Reprezentarea grafică a funcțiilor de undă

Pentru început am reprezentat grafic funcțiile de undă obținute din Hamiltonianul calculat în capitolul anterior (43). Pentru aceasta am folosit inițial reprezentarea matriceală a operatorilor de creare și anihilare descrisă de ecuațiile (49) și (51).

Inițial am folosit un număr maxim ales pentru parametrul dimensional $n_{max} = 5$. Algoritmul folosit este descris în secțiunile de mai jos. Pentru implementarea lui am folosit platforma de calcul interactivă Jupyter Notebook, utilizând limbaajul de programare Python3.

4.2.1 Inițializarea constantelor și operatorilor

Constantele fizice și matematice folosite pe parcursul programului sunt:

$$\hbar, e, E_J, C, L, \varphi_{Ext}, \pi, \phi_0 = \frac{h}{2e}$$

```

1 import scipy.constants as const
2 import math
3
4 hbar = const.hbar
5 electr = const.e
6 EJ = 6.2
7 C = 5.3e-15
8 L = 386e-9
9 phi_ext = 0.0
10 pi = math.pi
11 phi0 = const.h/(const.e*2)
12 nmax = 5

```


Am inițializat apoi matricele operatorilor de creare și anihilare a^\dagger și a , folosindu-ne de relația:

$$\langle m | a | n \rangle = \langle m | \sqrt{n-1} | n-1 \rangle = \sqrt{n} \delta_{m,n-1}$$

```

1 import numpy as np
2 def annihilation_operator(nmax):
3     a = np.zeros((nmax, nmax))
4     for n in range(1, nmax):
5         m = n-1
6         a[m, n] = math.sqrt(n)
7     return a

```

În continuare, operatorul de creare a^\dagger și cel de anihilare a vor fi notați cu **adag** și **a**.

```

1 a = annihilation_operator(5)
2 adag = a.conj().T

```

4.2.2 Inițializarea energiilor E_C și E_L și variabilele n_{ZPF} și φ_{ZPF}

Pentru ușurința de calcul și pentru a putea evidenția ulterior efectul modificării impedenței L și capacității C am exprimat variabilele E_C , E_L , n_{ZPF} și φ_{ZPF} în funcție de acestea.

Cunoscând formulele pentru n_{ZPF} și φ_{ZPF}

$$n_{zpf} = \frac{1}{2e} \sqrt{\frac{\hbar}{2} Z}$$

$$\varphi_{zpf} = 2\pi \frac{\sqrt{\frac{\hbar}{2Z}}}{\phi_0}$$

și luând în considerare

$$Z = \sqrt{\frac{L}{C}}$$

vom exprima

$$n_{zpf} = \frac{1}{2e} \sqrt{\frac{\hbar}{2} \sqrt{\frac{L}{C}}}$$

$$\varphi_{zpf} = 2\pi \frac{\sqrt{\frac{\hbar}{2\sqrt{\frac{L}{C}}}}}{\phi_0}$$

Aceste valori le vom inițializa în cod cu ajutorul funcțiilor

```

1 def Z(L, C):
2     Z = np.sqrt(L/C)
3     return Z
4
5 def n_zpf(L, C):
6     nzpf = 1/(2*electr)*np.sqrt((hbar/2)*np.sqrt(L/C))
7     return nzpf
8

```

```

9     def fi_zpf(L,C):
10         fizpf = 2*pi*np.sqrt(hbar/(2*np.sqrt(L/C)))/phi0
11         return fizpf

```

Energiile E_C și E_L le-am exprimat cu ajutorul formulelor

$$E_C = \frac{e^2}{2C}$$

$$E_L = \frac{\Phi_0^2}{4\pi^2 L}$$

$$\Phi_0 = \frac{h}{2e}$$

pe care le-am definit cu ajutorul funcțiilor

```

1     def fEC(C):
2         EC = electr*electr/(2*C)
3         return EC
4
5     def fEL(L):
6         phi0 = h/(2*electr)
7         EL = phi0*phi0/(4*math.pi*math.pi*L)
8         return EL

```

4.2.3 Descompunerea spectrală a matricii cosinus

În expresia Hamiltonianului (43) apare termenul $\cos[\varphi_{ZPF}(\hat{a} + \hat{a}^\dagger) - \varphi_{ext}]$, a cărui matrice o vom calcula prin descompunere spectrală.

Am inițializat termenul din interiorul cosinusului, numindu-l A .

```

1     A = fi_zpf(L,C)*(a+adag)-phi_ext*np.identity(nmax)

```

Pentru a putea diagonaliza matricea, i-am găsit vectorii și valorile proprii și, cu ajutorul matricii modale, am construit matricea diagonală A_d .

```

1     (wa, va) = np.linalg.eigh(A)
2     Ra = va
3     Ra_dagger = Ra.conj().T
4     A_d = np.matmul(np.matmul(Ra_dagger,A), Ra)

```

Având matricea diagonală, am construit matricile proiector folosindu-ne de formula

$$P_i = |v_i\rangle \langle v_i| \quad (53)$$

Pentru a putea aplica formula (53), am separat matricile vectorilor proprii v_a în 5 vectori individuali, cărora le-am obținut proiectorii P_0, P_1, P_2, P_3, P_4 .

```

1     P = np.empty((5,5), dtype = complex)
2     cosA = np.empty((5,5), dtype = complex)
3
4     vai0 = np.array([[va[0][0]], [va[1][0]], [va[2][0]],
5                     [va[3][0]], [va[4][0]]])
6     vai1 = np.array([[va[0][1]], [va[1][1]], [va[2][1]],
7                     [va[3][1]], [va[4][1]]])
8     vai2 = np.array([[va[0][2]], [va[1][2]], [va[2][2]],
9                     [va[3][2]], [va[4][2]]])

```

```

10 vai3 = np.array([[va[0][3]], [va[1][3]], [va[2][3]],
11                 [va[3][3]], [va[4][3]]])
12 vai4 = np.array([[va[0][4]], [va[1][4]], [va[2][4]],
13                 [va[3][4]], [va[4][4]]])
14
15 P0 = np.matmul(vai0, vai0.conj().T)
16 P1 = np.matmul(vai1, vai1.conj().T)
17 P2 = np.matmul(vai2, vai2.conj().T)
18 P3 = np.matmul(vai3, vai3.conj().T)
19 P4 = np.matmul(vai4, vai4.conj().T)

```

Matricea $\cos A$ va fi construită după formula

$$\cos A = \sum_i \cos(a_i) P_i \quad (54)$$

```

1 cosA=np.cos(wa[0])*P0+np.cos(wa[1])*P1+np.cos(wa[2])*P2
2 +np.cos(wa[3])*P3+np.cos(wa[4])*P4

```

Matricea $\cos[\varphi_{ZPF}(\hat{a} + \hat{a}^\dagger) - \varphi_{ext}]$ va avea valoarea

```

1 import pandas as pd
2 np.set_printoptions(linewidth=np.inf)
3 pd.options.display.float_format = '{:.4f}'.format
4 print(cosA)
5
6 [ 9.999e-01 -4.857e-17 -4.033e-08 -1.110e-16  9.020e-16]
7 [-4.857e-17  9.999e-01  1.387e-16 -6.986e-08  6.938e-16]
8 [-4.034e-08  1.387e-16  9.999e-01  3.331e-16 -9.880e-08]
9 [-1.110e-16 -6.986e-08  3.331e-16  9.999e-01  4.441e-16]
10 [ 9.020e-16  6.938e-16 -9.880e-08  4.4409e-16  9.999e-01]

```

4.2.4 Construirea matricii Hamiltonianului

Având și ultima matrice din formula Hamiltonianului, am defint o funcție care va genera matricea acestuia în funcție de parametrii E_J , L , C și φ_{ext} .

```

1 def const_ham_phiext(EJ,L,C,phiext):
2     t1 = -4*n_zpf(L,C)*n_zpf(L,C)*np.matmul(a-adag,a-adag)*fEC(C)
3     t2 = 1/2*fEL(L)*np.matmul(fi_zpf(L,C)*(a+adag)
4         -phi_ext, fi_zpf(L,C)*(a+adag)-phi_ext)
5     t3 = -EJ*cosA
6     H = t1 + t2 + t3
7     return H

```

Pentru valorile alese $E_J = 6.2GHz$, $L = 386nH$, $C = 5.3e - 6nF$, $\varphi_{ext} = 0.0$, matricea Hamiltonianului are forma:

```

1 H = const_ham_phiext(6.2,386e-9,5.3e-15,0.0)
2 print(H)
3
4 [-6.199e+00  3.011e-16  2.500e-07  6.883e-16 -5.592e-15]
5 [ 3.011e-16 -6.199e+00 -8.604e-16  4.331e-07 -4.302e-15]
6 [ 2.500e-07 -8.604e-16 -6.199e+00 -2.065e-15  6.125e-07]
7 [ 6.883e-16  4.331e-07 -2.065e-15 -6.199e+00 -2.753e-15]
8 [-5.592e-15 -4.302e-15  6.125e-07 -2.753e-15 -6.199e+00]

```

4.2.5 Funcțiile de undă în funcție de frecvență

Pentru a putea descrie comportamentul fluxonului am extras din matricea Hamiltonianului calculată anterior matricea de vectori proprii Ψ , folosindu-ne de o funcție predefinită în librăria NumPy de extragere a vectorilor și valorilor proprii, `linalg.eigh`.

```
1 E, Psi = np.linalg.eigh(H)
```

Am ales să scriem funcțiile de undă ale Hamiltonianului în baza Fock, pentru care am inițializat primii cinci vectori de bază conform definiției (55).

$$Fock_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} Fock_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} Fock_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} Fock_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} Fock_4 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (55)$$

```
1 Fock0 = np.zeros((5,1))
2 Fock0[0,0] = 1
3 Fock1 = np.zeros((5,1))
4 Fock1[1,0] = 1
5 Fock2 = np.zeros((5,1))
6 Fock2[2,0] = 1
7 Fock3 = np.zeros((5,1))
8 Fock3[3,0] = 1
9 Fock4 = np.zeros((5,1))
10 Fock4[4,0] = 1
```

Am separat ulterior matricea Ψ , numită în cod **Psi**, în 5 vectori unidimensionali **psi0**, **psi1**, **psi2**, **psi3**, **psi4**, pe care ulterior i-am folosit la generarea funcțiilor $\Psi_0(\varphi)$, $\Psi_1(\varphi)$, $\Psi_2(\varphi)$, $\Psi_3(\varphi)$, $\Psi_4(\varphi)$.

```
1 psi0 = Fock0*Psi[0,0]+Fock1*Psi[1,0]+Fock2*Psi[2,0]
2       +Fock3*Psi[3,0]+Fock4*Psi[4,0]
3 psi1 = Fock0*Psi[0,1]+Fock1*Psi[1,1]+Fock2*Psi[2,1]
4       +Fock3*Psi[3,1]+Fock4*Psi[4,1]
5 psi2 = Fock0*Psi[0,2]+Fock1*Psi[1,2]+Fock2*Psi[2,2]
6       +Fock3*Psi[3,2]+Fock4*Psi[4,2]
7 psi3 = Fock0*Psi[0,3]+Fock1*Psi[1,3]+Fock2*Psi[2,3]
8       +Fock3*Psi[3,3]+Fock4*Psi[4,3]
9 psi4 = Fock0*Psi[0,4]+Fock1*Psi[1,4]+Fock2*Psi[2,4]
10      +Fock3*Psi[3,4]+Fock4*Psi[4,4]
```

Pentru a afla dependența de φ a funcțiilor de undă Ψ , considerăm descompunerea:

$$\Psi_0(\varphi) = \langle \varphi | \Psi_0 \rangle = \langle \varphi | \alpha_0 | 0 \rangle + \langle \varphi | \alpha_1 | 1 \rangle + \dots + \langle \varphi | \alpha_{n-1} | n-1 \rangle \quad (56)$$

unde

$$\varphi | 0 \rangle = \Phi_0^{HO}(\varphi) \quad (57)$$

Atunci, descompunerea se poate rescrie ca

$$\Psi_0(\varphi) = \alpha_0 \Phi_0^{HO}(\varphi) + \alpha_1 \Phi_1^{HO}(\varphi) + \dots + \alpha_{n-1} \Phi_{n-1}^{HO}(\varphi) \quad (58)$$

Ținând cont că în această formulă:

$$\Phi_n^{HO}(\varphi) = \frac{1}{\sqrt{n!2^n\sqrt{\pi}}} e^{\frac{\varphi^2}{2\varphi_{zpf}}} H_n\left(\frac{\varphi}{\sqrt{2}\varphi_{zpf}}\right) \quad (59)$$

sunt funcțiile de undă ale oscilatorului armonic, pentru a construi aceste funcții de undă ne folosim de **harmonic waven**.

```

1 def harmonic_waven(n, phi, L, C):
2     norm = math.sqrt((2**n)*math.factorial(n)*math.sqrt(math.pi))
3     expon = math.exp(-phi*phi/(2*fi_zpf(L,C)*fi_zpf(L,C)))
4     pol = hermite_poli(phi/(math.sqrt(2)*fi_zpf(L,C)), n)
5     psi = expon*pol/norm
6     return psi

```

Expresile $\Psi_0(\varphi), \Psi_1(\varphi), \Psi_2(\varphi), \Psi_3(\varphi), \Psi_4(\varphi), E(\varphi)$ le-am definit individual cu ajutorul unor funcții de tip:

```

1 def Psi0_phi(phi, nmax, L, C):
2     Psi0fi = psi0.item(0)*harmonic_waven(0, phi, L, C)
3     +psi0.item(1)*harmonic_waven(1, phi, L, C)
4     +psi0.item(2)*harmonic_waven(2, phi, L, C)
5     +psi0.item(3)*harmonic_waven(3, phi, L, C)
6     +psi0.item(4)*harmonic_waven(4, phi, L, C)
7     return (Psi0fi)

```

În gridul pe care l-am folosit la plotare am ales 500 puncte (**ngrid**) și am ales domeniul lui φ astfel:

$$\varphi = [-0.001, 0.001] \quad (60)$$

După această primă inițializare am generat valorile funcțiilor definite anterior peste întreg spațiul.

```

1 ngrid = 1500 #puncte pe axa x
2 phi_min = -0.001
3 phi_max = 0.001 #interval phi
4 d_phi = (phi_max-phi_min)/(ngrid-1) #discretizare
5 phi_grid = np.zeros((ngrid))
6 f_grid = np.zeros((ngrid))
7 g_grid = np.zeros((ngrid))
8 h_grid = np.zeros((ngrid))
9 k_grid = np.zeros((ngrid))
10 l_grid = np.zeros((ngrid))
11 m_grid = np.zeros((ngrid))
12
13 for i in range (ngrid):
14     phi_grid[i] = phi_min+i*d_phi
15     f_grid[i] = Psi0_phi(phi_grid[i], nmax, L, C)
16     g_grid[i] = Psi1_phi(phi_grid[i], nmax, L, C)
17     h_grid[i] = Psi2_phi(phi_grid[i], nmax, L, C)
18     k_grid[i] = Psi3_phi(phi_grid[i], nmax, L, C)
19     l_grid[i] = Psi4_phi(phi_grid[i], nmax, L, C)
20     m_grid[i] = E_phi(phi_grid[i], nmax, L, C)

```

Reprezentarea grafică a acestora am făcut-o pe același grafic:

```

1 import matplotlib.pyplot as plot
2
3 fig, ax1 = plot.subplots(figsize=(14,9))

```

```

4     fig.autofmt_xdate()
5     plot.plot(phi_grid, f_grid)
6     plot.plot(phi_grid, g_grid)
7     plot.plot(phi_grid, h_grid)
8     plot.plot(phi_grid, k_grid)
9     plot.plot(phi_grid, l_grid)
10    plot.plot(phi_grid, m_grid)
11    fig.set_dpi(300)
12
13    ax1.set_xlabel(r"$\varphi$", fontsize=22)
14    ax1.set_ylabel(r" $|\Psi(\varphi)|$", fontsize=22)
15    ax1.set_title(r"\bf{ Func ii de und } $\mathbf{|\Psi(\varphi)|}$ \bf{ }$ \mathbf{n=5}$ ", fontsize=22)
16
17    plot.show()

```

Pentru plotarea probabilităților ne interesează să reprezentăm grafic pătratul funcțiilor de undă.

```

1     fig, ax1 = plot.subplots(figsize=(14,9))
2     fig.autofmt_xdate()
3     plot.plot(phi_grid, f_grid**2)
4     plot.plot(phi_grid, g_grid**2)
5     plot.plot(phi_grid, h_grid**2)
6     plot.plot(phi_grid, k_grid**2)
7     plot.plot(phi_grid, l_grid**2)
8     fig.set_dpi(300)
9
10    ax1.set_xlabel(r"$\varphi$", fontsize=22)
11    ax1.set_ylabel(r" $|\Psi(\varphi)|^2$", fontsize=22)
12    ax1.set_title(r" $\mathbf{|\Psi(\varphi)|^2}$ \bf{ }$ \mathbf{n=5}$ ",
13    fontsize=22)
14    plot.show()

```

Reprezentarea grafică a funcțiilor de undă calculate este ilustrată în Figura 4.1, iar probabilitățile aferente acestora sunt ilustrate în Figura 4.2.

Mai departe, urmând aceiași pași descriși de algoritmul prezentat mai sus, am plotat primele 7 funcții de undă și probabilitățile aferente acestora, însă considerând influența primilor $n_{max} = 20$ termeni din descompunerea descrisă de formula (58). Rezultatele acestor reprezentări se regăsesc în Figura 4.3 și Figura 4.4.

```

1     import scipy.constants as const
2     from numpy.polynomial.hermite import hermval
3     import math
4
5     hbar = const.hbar
6     electr = const.e
7     EJ = 6.5348e-25
8     C = 5.3e-15
9     L = 386e-9
10    phi_ext = 0.0
11    pi = math.pi
12    phi0 = const.h/(const.e*2)
13    nmax = 20

```

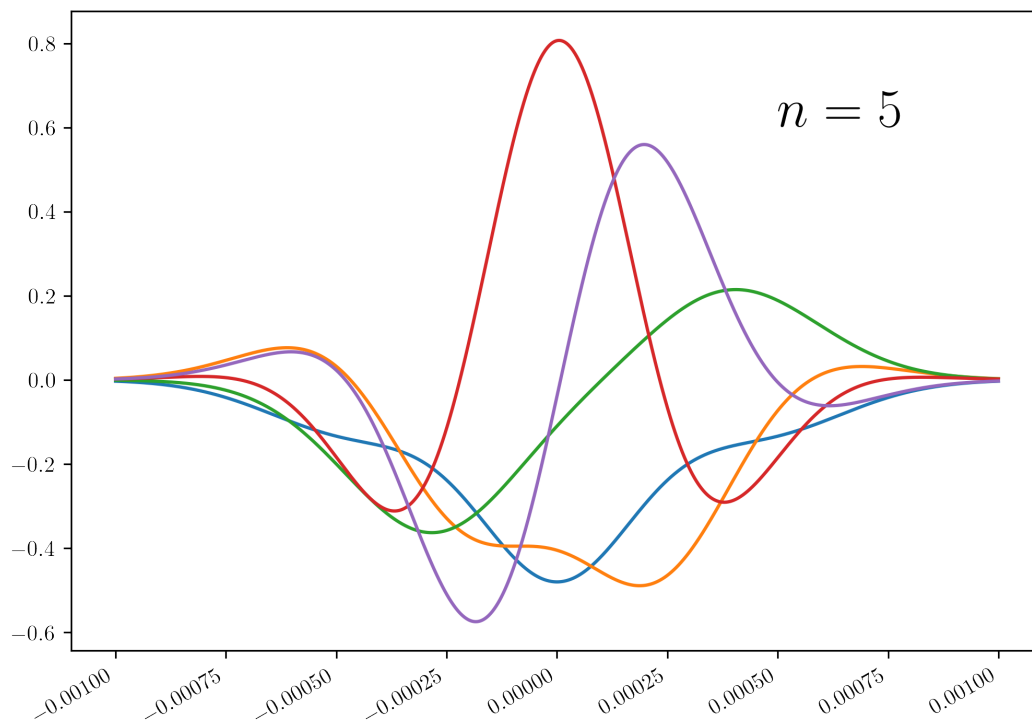


Figura 4.1: Reprezentarea grafică a primelor 5 funcții de undă Ψ în funcție de φ , considerând $n_{max} = 5$

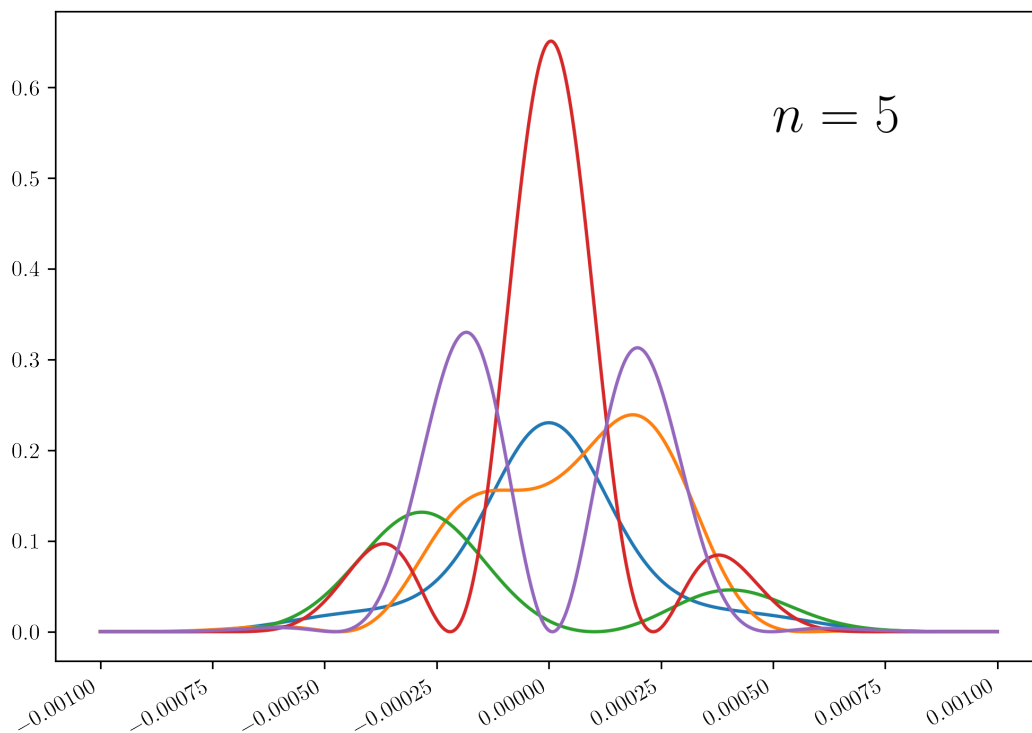


Figura 4.2: Reprezentarea grafică a probabilităților $|\Psi(\varphi)|^2$ primelor 5 funcții de undă

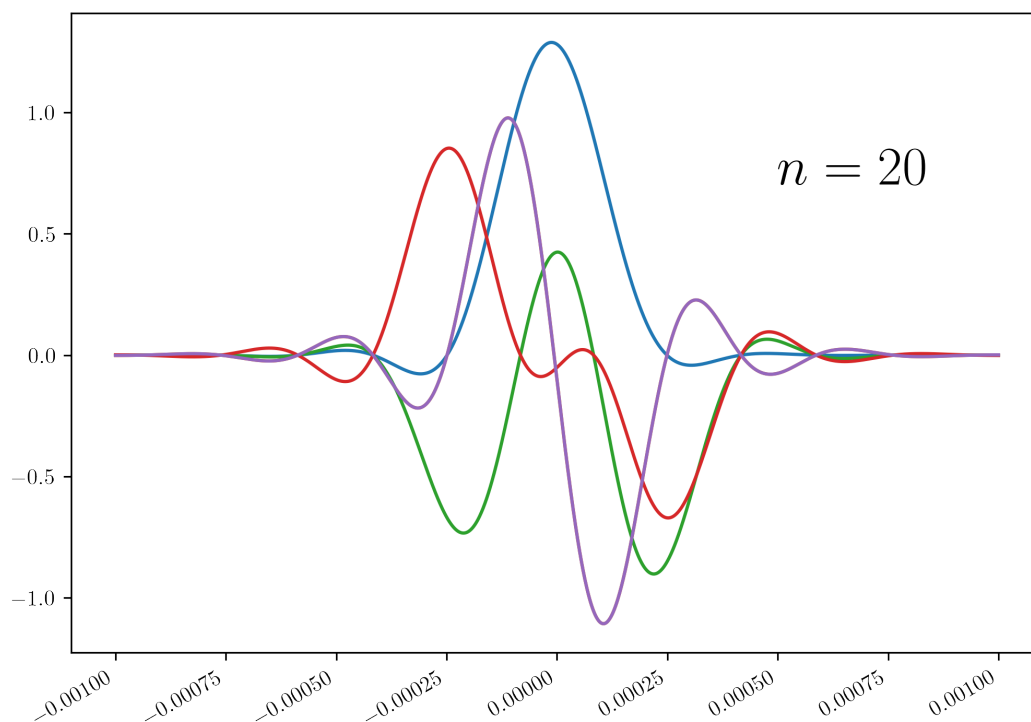


Figura 4.3: Reprezentarea grafică a primelor 7 funcții de undă Ψ în funcție de φ , considerând $n_{max} = 20$

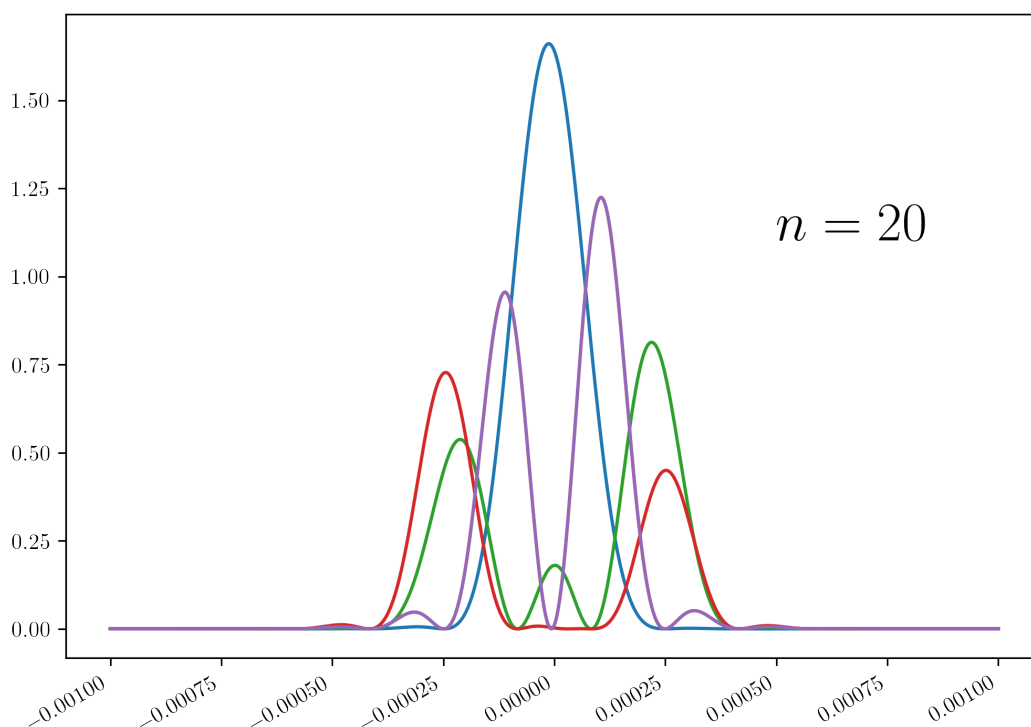


Figura 4.4: Reprezentarea grafică a probabilităților $|\Psi(\varphi)|^2$ primelor 7 funcții de undă cu $n_{max} = 20$

4.3 Studiul energiilor

În această parte a lucrării ne-am propus să studiem nivelele energetice necesare funcționării qubitului supraconductor cu superinductanțe. Pachetele utilizate, precum și constantele de interes, au fost definite. La definirea valorilor pentru capacitate și inductanță am ales valorile $C = 5.3fF$ și $L = 386nH$ din articolul lui Smith et al.(2016) [41]. Iar pentru studiul energiilor, am ales $n_{max} = 20$.

```

1 import scipy.constants as const
2 import math
3 import matplotlib.pyplot as plot
4 import numpy as np
5
6 h = const.h
7 hbar = const.hbar
8 electr = const.e
9 C = 5.3e-15 #din Smith
10 L = 386e-9 #din Smith
11 phi_ext = 0.0
12 pi = math.pi
13 phi0 = const.h/(const.e*2)
14 nmax = 20

```

4.3.1 Exprimarea Hamiltonianului în funcție de energii

Pentru studiul energetic al sistemului alegem să exprimăm hamiltonianul în funcție de termenii de energie E_J , E_L și E_C . Pentru aceasta, definim două funcții **fC** și **fL**, care vor calcula valoare E_C , respectiv E_L , în funcție de valoarea aleasă anterior pentru C și L .

```

1 def fEC(C):
2     EC =electr*electr/(2*C)
3     return EC
4 def fEL(L):
5     EL = phi0*phi0/(4*math.pi*math.pi*L)
6     return EL

```

Pentru a simplifica calculele și reprezentarea grafică ne interesează să schimbăm unitatea de măsură a energiilor din J în GHz .

$$[E]_J = \frac{hc}{\lambda} = h[\nu]_{Hz} \quad (61)$$

```

1 def JtoGHz(x):
2     energy = x/h*1.e-9
3     return energy

```

În expresia Hamiltonianului (43), pe lângă termenii de energie, avem matricile operatorilor a și a^\dagger de dimensiune 20×20 și expresiile lui n_{ZPF} și φ_{ZPF} .

```

1 a = annihilation_operator(nmax)
2 adag = a.conj().T

```

Pentru a afla valorile lui n_{ZPF} și φ_{ZPF} în funcție de energii vom defini o funcție **parametrii**.

```

1 def parametrii(EC, EL):
2     L = phi0*phi0/(4*EL*math.pi*math.pi)

```

```

3     C = electr*electr/(2*EC)
4     Z = math.sqrt(L/C)
5     omega = math.sqrt(L*C)
6     qzpf = math.sqrt((hbar)/(2*Z))
7     nzpf = qzpf/(2*electr)
8     PHIZpf = math.sqrt((hbar*Z)/(2))
9     phizpf = 2*math.pi*PHIZpf/phi0
10
11    return L, C, omega, nzpf, phizpf

```

Am urmat apoi același procedeu de creare a matricii $\cos[\varphi_{ZPF}(\hat{a} + \hat{a}^\dagger) - \varphi_{ext}]$ prin descompunere spectrală, luând în considerare $n_{max} = 20$.

După obținerea expresiei matricii nu a mai rămas decât să definim o funcție \mathbf{H} care să calculeze Hamiltonianul prin simple înmulțiri și adunări.

```

1     def H(EJ,EL,EC,phiext):
2         L, C, omega, nzpf, phizpf = parametrii(EC,EL)
3         t1=-4*nzpf*nzpf*np.matmul(a-adag,a-adag)*EC
4         t2=1/2*EL*np.matmul(phizpf*(a+adag)-phiext,phizpf*(a+adag)-phiext)
5         t3=-EJ*cosA
6         H = t1 + t2 + t3
7     return H

```

4.3.2 Forma potențialului qubitului

Expresia potențialului sistemului circuitului supraconductor este:

$$V(\varphi) = \frac{E_L \varphi^2}{2} - E_J \cos(\varphi - \varphi_{ext}) \quad (62)$$

Pentru acesta, am definit funcția \mathbf{V} :

```

1     def V(EJ,EL,phi,phi_ext):
2         V = EL*phi*phi*1/2-EJ*np.cos(phi-phi_ext)
3     return V

```

După ce am reprezentat matriceal sistemul fluxonului, am putut studia efectul modificării valorii fluxului magnetic exterior φ_{ext} asupra potențialului și funcțiilor proprii ale acestuia. Efectul resimțit de potențial odată cu variația fluxului exterior între cele patru valori $[0, \pi, \frac{\pi}{2}, \frac{3\pi}{2}]$ este ilustrat în Figurile 4.5 și 4.6.

Valoarea fluxului exterior „aranjează” efectul oscilant al potențialului Josephson peste potențialul parabolic al circuitului. Prin variația fluxului magnetic exterior, potențialul sistemului se poate schimba fără mult efort dintr-un potențial de tip groapă $\varphi_{ext} = 0$ la unul simetric cu două gropi la $\varphi_{ext} = \pi$.

În stagiile intermediare celor două forme ale potențialului se observă tendința de a lua forma unei duble gropi. Variația fluxului magnetic exterior este un element esențial în proiectarea de qubiți supraconductori, deoarece acesta dictează atât forma potențialului capabil de confinare a anumitor stări energetice, cât și valorile acestor energii, adică valorile proprii ale matricii sistemului.

Este imperativ atunci în proiectarea unui qubit ca nivelele energetice care prezintă tranziții de interes, precum $|0\rangle \leftrightarrow |1\rangle$, să fie cât mai mult separate energetic de nivelele mai ridicate prin alegerea parametrilor de circuit și fluxului exterior φ_{ext} ideal.

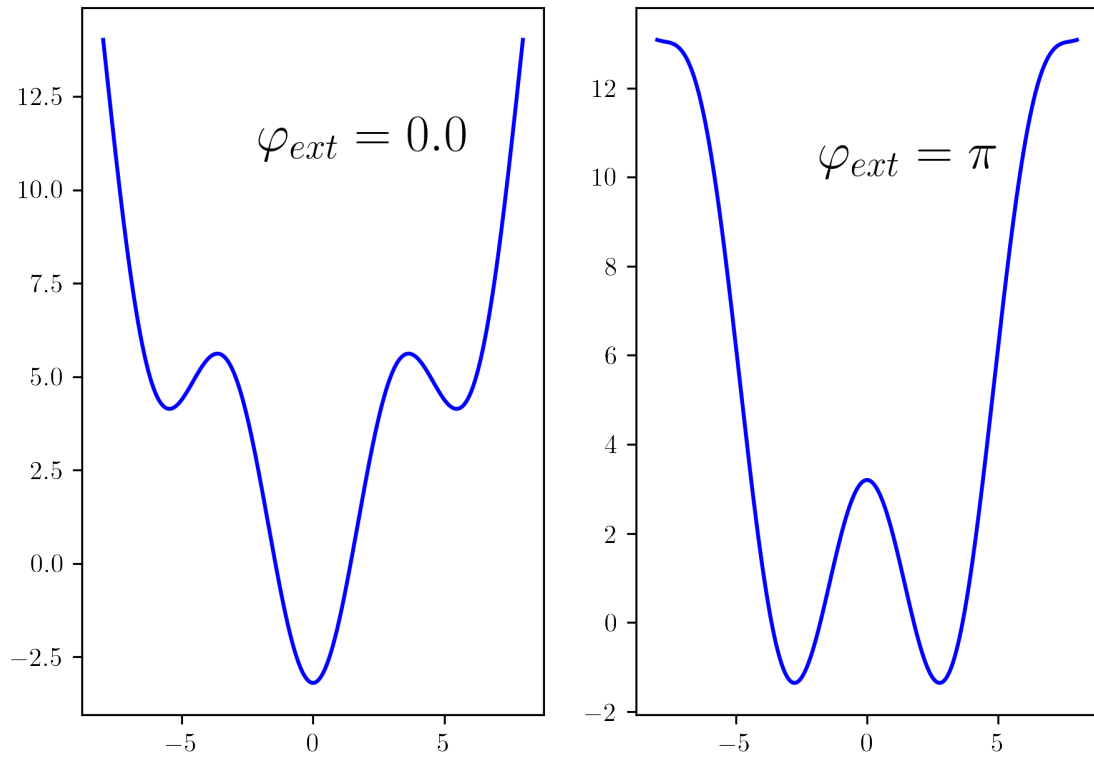


Figura 4.5: Reprezentarea grafică a potențialului $V(\varphi, \varphi_{ext})$ pentru $\varphi_{ext} = 0$ și $\varphi_{ext} = \pi$

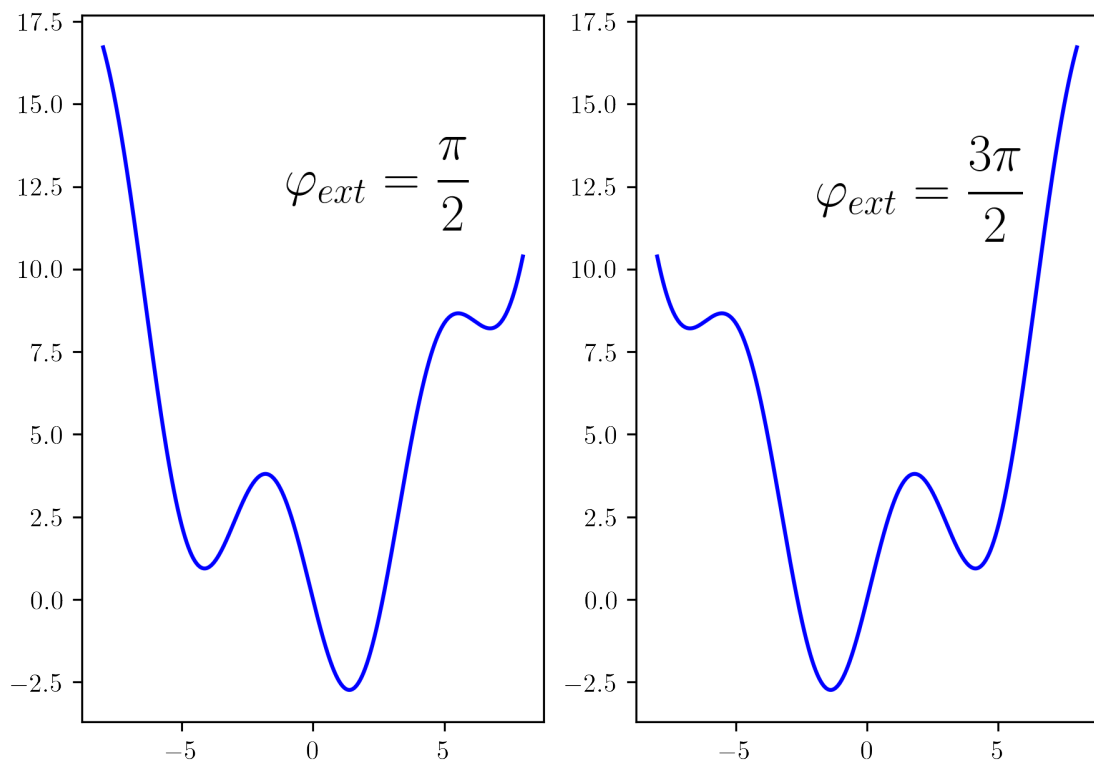


Figura 4.6: Reprezentarea grafică a potențialului $V(\varphi, \varphi_{ext})$ pentru $\varphi_{ext} = \frac{\pi}{2}$ și $\varphi_{ext} = \frac{3\pi}{2}$

4.3.3 Influența E_J asupra potențialului

Prin reprezentarea grafică a potențialului am reușit să studiem influența termenului de energie E_J asupra formei acestuia (Figura 4.7). Odată cu creșterea energiei joncțiunii Josephson se observă o creștere a anarmonicității potențialului, îndepărtându-se din ce în ce mai mult de forma potențialului oscilatorului armonic.

Putem concluziona că parametrul E_J crește rezistența qubitului la zgomot, crescând barierele de potențial pe care particula ar trebui să le „sară” la tunelarea dintr-un minim local în altul.

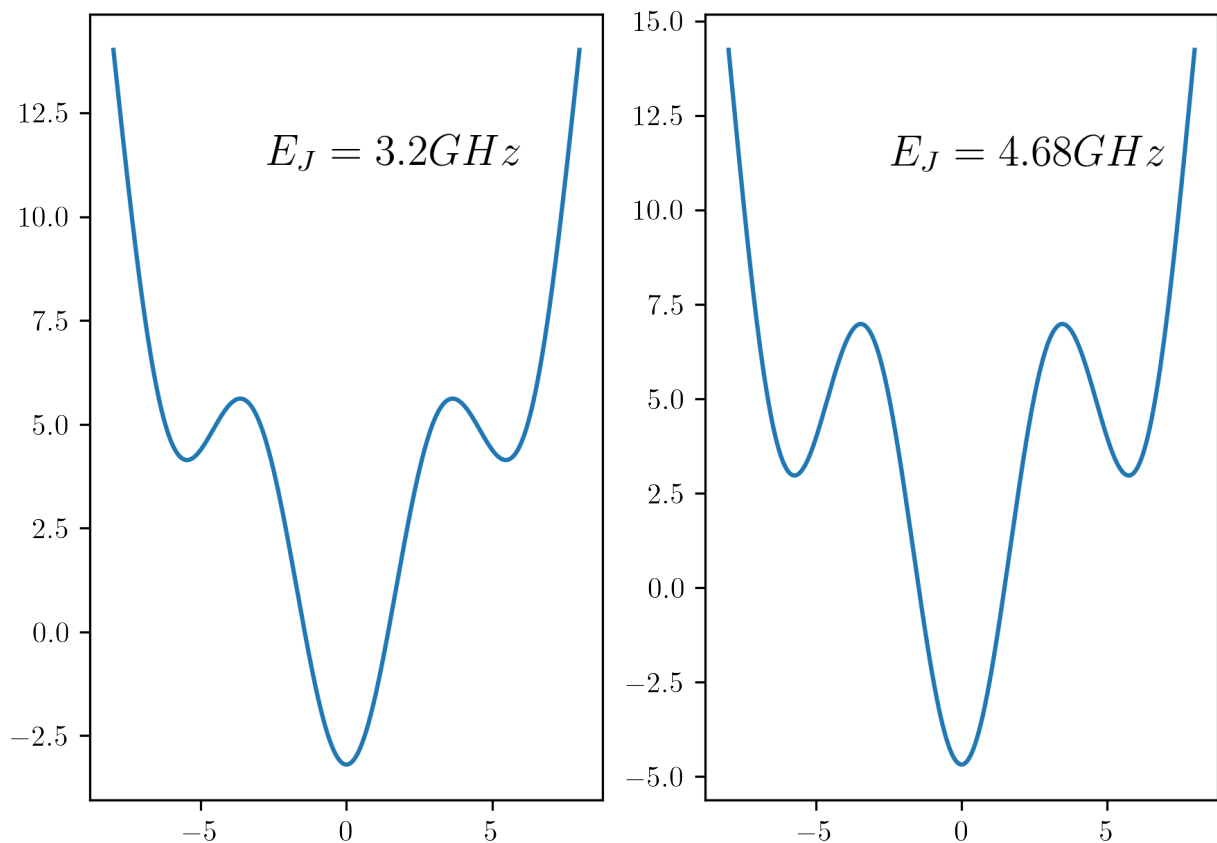


Figura 4.7: Potențialul $V(\varphi)$ reprezentat pentru două valori ale lui E_J

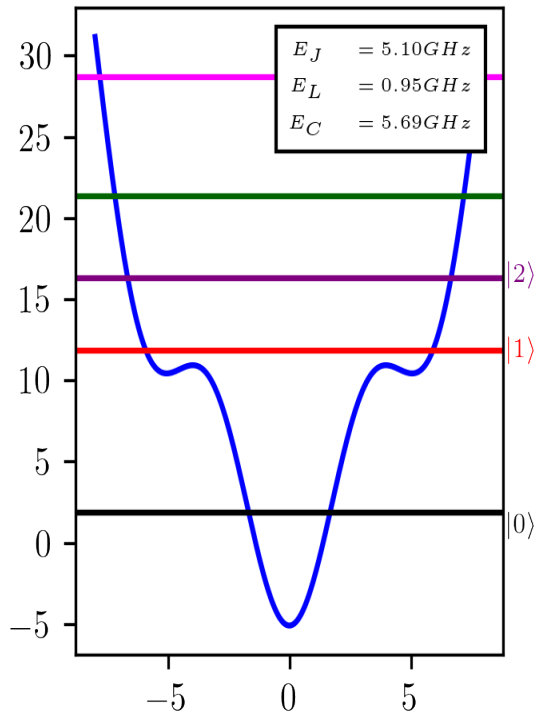
4.3.4 Nivelele energetice ale sistemului

Pentru a reprezenta grafic nivelele energetice ale qubitului supraconductor am definit o funcție **Energie**, care, cu ajutorul `np.linalg.eigh`, va extrage vectorii și valorile proprii din Hamiltonianul calculat mai sus, pentru câteva seturi de parametri ale unor implementări fizice ale fluxoniumului luate din Masluck (2012) [42] și din Gusenkova et al. (2022) [43]. Reprezentarea grafică a potențialului și a nivelelor energetice respective unor configurații de parametri sunt ilustrate mai jos (Figura 4.8).

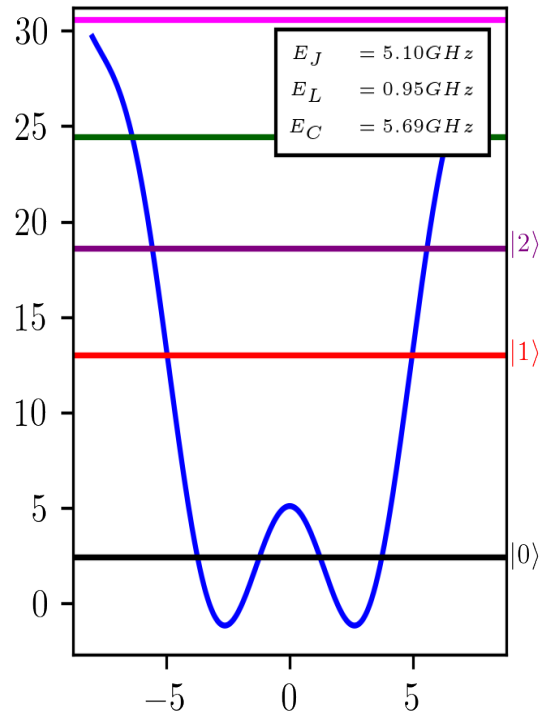
```

1  def Energie(EJ,EL,EC,phiext):
2      fe, ve = np.linalg.eigh(H(EJ,EL,EC,phiext))
3      energy = fe
4      return energy
5  energy = Energie(EJ,EL,EC,phi_ext)

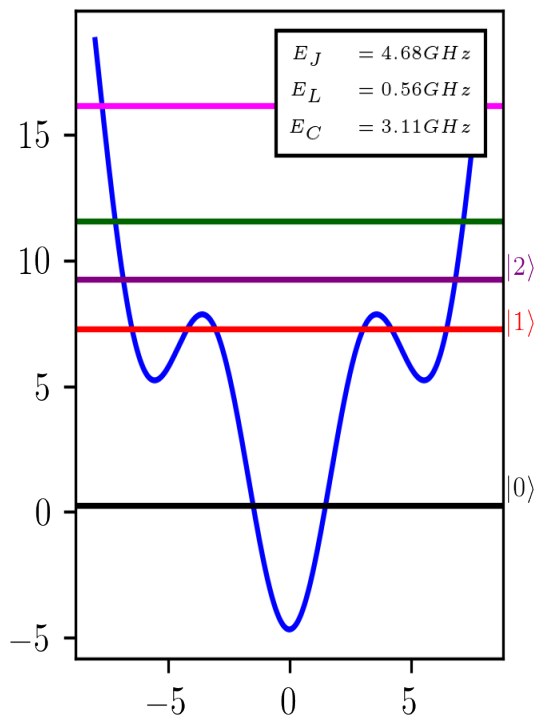
```



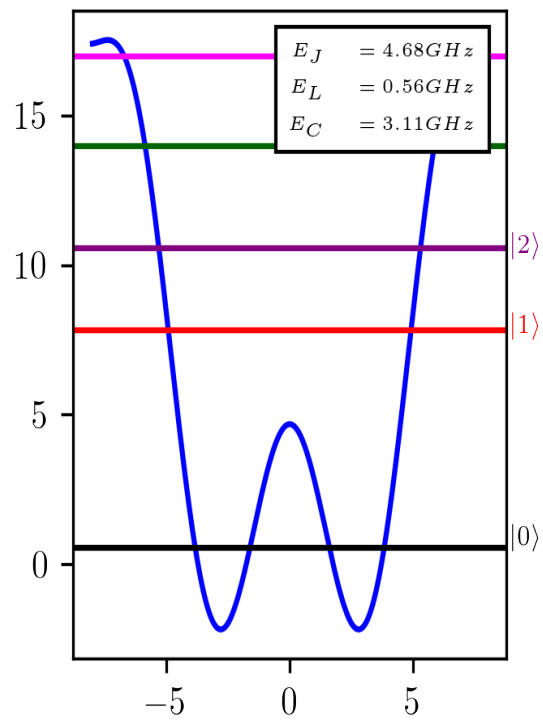
(a) $L = 172 \text{ nH}, C = 3.4 \text{ fF}, \varphi_{ext} = 0.0$



(b) $L = 172 \text{ nH}, C = 3.4 \text{ fF}, \varphi_{ext} = \pi$



(c) $L = 289 \text{ nH}, C = 6.22 \text{ fF}, \varphi_{ext} = 0.0$



(d) $L = 289 \text{ nH}, C = 6.22 \text{ fF}, \varphi_{ext} = \pi$

Figura 4.8

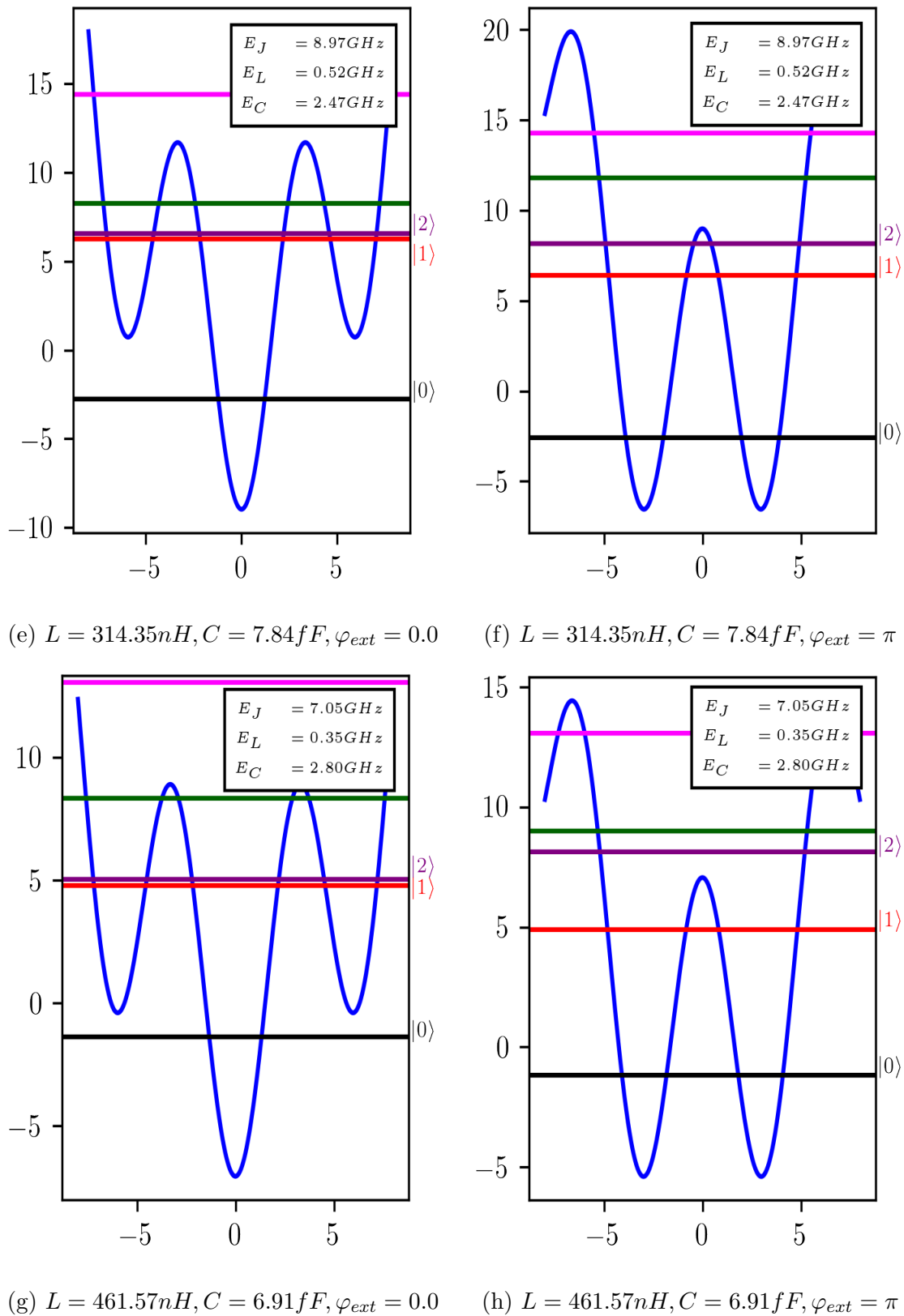


Figura 4.8: Reprezentarea potențialului qubitului supraconductor studiat și a primelor 5 nivele energetice aferente acestuia pentru patru seturi de valori E_J , L , C experimentale. În grafice sunt evidențiate starea fundamentală $|0\rangle$ cu negru, prima stare energetică $|1\rangle$ cu roșu și cea de a doua stare energetică $|2\rangle$ cu mov.

După cum este ilustrat în graficele (a)-(h) din Figura 4.8, parametrii aleși pentru capacitate C , inductanță L și energie Josephson E_J sunt foarte importanți în proiectarea unei arhitecturi de qubit supraconductor.

Se poate observa din grafice că starea fundamentală $|0\rangle$, reprezentată prin linia de culoare neagră, este bine confinată în groapa de potențial (în cazul graficelor (a),(c),(e),(g) pentru $\varphi_{ext} = 0$) sau într-una din cele două gropi de potențial (pentru cazurile (b),(d),(f),(h) în care potențialul are o formă simetrică cu două minime globale datorat fluxului exterior $\varphi_{ext} = \pi$).

Poziționarea pe prima stare energetică $|1\rangle$ prezintă în schimb posibilitatea de tunelare a particulei în toate cele trei gropi de potențial, în cazul $\varphi_{ext} = 0$. Se poate observa din graficele de la punctele (b),(d),(f),(h) că odată cu variația fluxului magnetic exterior putem manipula starea $|1\rangle$ spre o stare care ar permite tranziția $|0\rangle \leftrightarrow |1\rangle$ caracteristică și necesară funcționării circuitului supraconductor precum un qubit.

Urcând spre stările energetice mai înalte, de la $|2\rangle$ în sus, se observă din grafice că acestea nu mai pot fi controlate și manipulate schimbând potențialul prin modificarea fluxului exterior.

În concluzie, prin cunoașterea elementelor care influențează potențialul și implicit și valorile Hamiltonianului se pot proiecta circuite supraconductoare cu superinductanțe capabile să funcționeze optim în regim de qubit.

Concluzii

În concluzie, am realizat în cadrul acestei lucrări un studiu asupra qubiților bazați pe circuite supraconductoare cu superinductanțe.

Am început cu o scurtă introducere istorică asupra calculului cuantic, unde am prezentat contextul istoric și cel actual al calculatoarelor cuantice, prezentarea fiind axată pe momentele cheie din dezvoltarea lor.

Pentru înțelegerea proceselor din spatele funcționării calculatoarelor cuantice, am introdus noțiunea de qubit, echivalentul cuantic al bit-ului clasic. Am prezentat principalele procese pe care se bazează funcționarea unui qubit: superpoziția și paralelismul cuantic, și am discutat succint criteriile necesare pentru implementarea calculului cuantic. În acest context, am prezentat etapele de dezvoltare a unui sistem cuantic spre un calculator cuantic tolerant la erori, și am discutat prezentat etapa în care se află sistemele bazate pe qubiți supraconductori în acel model.

Spre a ne realiza obiectivul de simulare a funcțiilor de undă și a nivelelor energetice, am reușit să modelăm procesele prin construirea matricii hamiltonianului sistemului. Am ajuns la forma analitică a hamiltonianului pornind de la considerente de analiză de circuit, pe care le-am aplicat unui circuit echivalent al fluxonului.

Am reușit să ajung la forma matriceală a hamiltonianului prin scrierea acestuia în formalismul celei de a doua cuantificări, exprimarea operatorilor de creare și anihilare în forma lor matriceală (prin calcularea elementelor de matrice), descompunerea spectrală a elementului oscilant datorat joncțiunii Josephson, și reuniunea tuturor acestor termeni într-o funcție de parametrii E_J , L , C , φ_{ext} specifice qubitului cu superinductanțe.

După ce am găsit relațiile dintre parametrii de circuit și hamiltonianul sistemului, am reușit să reprezentăm pentru mai multe valori realiste, care au fost deja implementate experimental, graficele energetice care cuprind potențialul și primele 5 nivele energetice. Având aceste grafice, am purtat o discuție despre optimizarea funcționării ca qubit a circuitului supraconductor și efectul variației fluxului magnetic exterior, φ_{ext} . Lucrarea de față prezintă deci un algoritm capabil să proiecteze și să simuleze mărimi de interes precum probabilitățile, funcțiile de undă, potențialul energetic și nivelele proprii de energie ale unor qubiți bazați pe circuite supraconductoare cu superinductanțe, pe care l-am folosit pentru evidențierea influenței diferiților parametrii în funcționarea optimă a qubitului.

În concluzie, qubiții bazați pe circuite supraconductoare cu superinductanțe prezintă o arhitectură care ne permite să accesăm țintit nivele energetice de interes precum $|0\rangle \leftrightarrow |1\rangle$ și care ne permite controlul asupra lor prin modificarea fluxului magnetic exterior. Deși încă extrem de zgomotoase și foarte sensibile la factori externi precum temperatura și radiațiile cosmice, qubiții bazați pe circuite supraconductoare sunt cea mai avansată și promițătoare implementare a calculatorului cuantic.

Bibliografie

- [1] John Preskill. Quantum computing 40 years later. *arXiv:2106.10522 [quant-ph]*, June 2021.
- [2] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6):467–488, June 1982.
- [3] <https://mitendicotthouse.org/physics-computation-conference/>(accesat la 02.05.2022).
- [4] I. M. Georgescu, S. Ashhab, and Franco Nori. Quantum Simulation. *Reviews of Modern Physics*, 86(1):153–185, March 2014.
- [5] Bruce M. Boghosian and Washington Taylor. Simulating quantum mechanics on a quantum computer. *Physica D: Nonlinear Phenomena*, 120(1-2):30–42, September 1998.
- [6] A. Bermudez, M. A. Martin-Delgado, and E. Solano. Exact mapping of the $S+1$ Dirac oscillator onto the Jaynes-Cummings model: Ion-trap experimental proposal. *Physical Review A*, 76(4):041801, October 2007.
- [7] J. Q. You and Franco Nori. Atomic physics and quantum optics using superconducting circuits. *Nature*, 474(7353):589–597, June 2011.
- [8] Ivan Kassal, James D. Whitfield, Alejandro Perdomo-Ortiz, Man-Hong Yung, and Alán Aspuru-Guzik. Simulating chemistry using quantum computers. *Annual Review of Physical Chemistry*, 62(1):185–207, May 2011.
- [9] Uwe R. Fischer and Ralf Schützhold. Quantum simulation of cosmic inflation in two-component Bose-Einstein condensates. *Physical Review A*, 70(6):063615, December 2004.
- [10] Paul Benioff. The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. *Journal of Statistical Physics*, 22(5):563–591, May 1980.
- [11] Quantum theory, the Church–Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, July 1985.
- [12] <https://www.nytimes.com/2019/10/30/opinion/google-quantum-computer-sycamore.html> (accesat la 03.05.2022).
- [13] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. In *In Proc. 25th Annual ACM Symposium on Theory of Computing, ACM*, pages 11–20, 1993.
- [14] Daniel R. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997.
- [15] Peter W Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. page 26.
- [16] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, Oct 1995.

- [17] Lov K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, 79:325–328, Jul 1997.
- [18] John Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, aug 2018.
- [19] <https://www.newscientist.com/article/2283945-google-demonstrates-vital-step-towards-large-scale-quantum-computers/>(accesat la 04.05.2022).
- [20] Cupjin Huang, Fang Zhang, Michael Newman, Junjie Cai, Xun Gao, Zhengxiong Tian, Junyin Wu, Haihong Xu, Huanjun Yu, Bo Yuan, Mario Szegedy, Yaoyun Shi, and Jianxin Chen. Classical simulation of quantum supremacy circuits, 2020.
- [21] Aram W. Harrow and Ashley Montanaro. Quantum computational supremacy. *Nature*, 549(7671):203–209, sep 2017.
- [22] John Preskill. Quantum computing and the entanglement frontier, 2012.
- [23] <https://www.nature.com/articles/d41586-021-03476-5/>(accesat la 09.05.2022).
- [24] <https://research.ibm.com/blog/ibm-quantum-roadmap/>(accesat la 09.05.2022).
- [25] <https://research.ibm.com/blog/127-qubit-quantum-processor-eagle/>(accesat la 09.05.2022).
- [26] Jens Koch, Terri M. Yu, Jay Gambetta, A. A. Houck, D. I. Schuster, J. Majer, Alexandre Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf. Charge-insensitive qubit design derived from the cooper pair box. *Phys. Rev. A*, 76:042319, Oct 2007.
- [27] <https://www.hpcwire.com/2021/12/13/ibm-breaks-100-qubit-qpu-barrier-marks-milestones-on-ambitious-roadmap/>(accesat la 09.05.2022).
- [28] <https://research.ibm.com/blog/eagle-quantum-processor-performancefn-2/>(accesat la 09.05.2022).
- [29] Zahid Hussain. Strengths and weaknesses of quantum computing. *International Journal of Scientific and Engineering Research*, 7, 09 2016.
- [30] Farzan Jazaeri, Arnout Beckers, Armin Tajalli, and Jean-Michel Sallese. A Review on Quantum Computing: Qubits, Cryogenic Electronics and Cryogenic MOSFET Physics. *arXiv:1908.02656 [physics, physics:quant-ph]*, August 2019.
- [31] T. Radtke and S. Fritzsche. Simulation of n-qubit quantum systems. I. Quantum registers and quantum gates. *Computer Physics Communications*, 173(1):91–113, 2005.
- [32] M. H. Devoret and R. J. Schoelkopf. Superconducting Circuits for Quantum Information: An Outlook. *Science*, 339(6124):1169–1174, March 2013.
- [33] David P. DiVincenzo. The Physical Implementation of Quantum Computation. *Fortschritte der Physik*, 48(9-11):771–783, 2000.

- [34] Long B. Nguyen, Yen-Hsiang Lin, Aaron Somoroff, Raymond Mencia, Nicholas Grabon, and Vladimir E. Manucharyan. High-coherence fluxonium qubit. *Phys. Rev. X*, 9:041041, Nov 2019.
- [35] S.E. Rasmussen, K.S. Christensen, S.P. Pedersen, L.B. Kristensen, T. Bækkegaard, N.J.S. Loft, and N.T. Zinner. Superconducting Circuit Companion—an Introduction with Worked Examples. *PRX Quantum*, 2(4):040204, December 2021.
- [36] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver. A quantum engineer's guide to superconducting qubits. *Applied Physics Reviews*, 6(2):021318, jun 2019.
- [37] B. D. Josephson. Possible new effects in superconductive tunnelling. *Physics Letters*, 1(7):251–253, July 1962.
- [38] *Physics and Applications of the Josephson Effect*, chapter 12, pages 354–382. John Wiley Sons, Ltd, 1982.
- [39] Lukas Grünhaupt. *Granular aluminium superinductors*. KIT Scientific Publishing, Karlsruhe, Germania, 2019.
- [40] Vladimir E. Manucharyan, Jens Koch, Leonid I. Glazman, and Michel H. Devoret. Fluxonium: Single Cooper-Pair Circuit Free of Charge Offsets. *Science*, 326(5949):113–116, October 2009.
- [41] W. C. Smith, A. Kou, U. Vool, I. M. Pop, L. Frunzio, R. J. Schoelkopf, and M. H. Devoret. Quantization of inductively shunted superconducting circuits. *Physical Review B*, 94(14), oct 2016.
- [42] Nicholas Adam Masluk. *Reducing the Losses of the Fluxonium Artificial Atom*. PhD dissertation, Yale University, 2012.
- [43] Daria Gusenkova, Francesco Valenti, Martin Spiecker, Simon Günzler, Patrick Paluch, Dennis Rieger, Larisa-Milena Pioraş-Țimbolmaş, Liviu P. Zârbo, Nicola Casali, Ivan Colantoni, Angelo Cruciani, Stefano Pirro, Laura Cardani, Alexandru Petrescu, Wolfgang Wernsdorfer, Patrick Winkel, and Ioan M. Pop. Operating in a deep underground facility improves the locking of gradiometric fluxonium qubits at the sweet spots. *Applied Physics Letters*, 120(5):054001, January 2022.

Declarație pe proprie răspundere

Subsemnata, Căpușan Miruna Andreea, declar că Lucrarea de licență pe care o voi prezenta în cadrul examenului de finalizare a studiilor de la Facultatea de Fizică, din cadrul Universității Babeș-Bolyai, în sesiunea 2022, sub îndrumarea Prof. Dr. Ioan Grosu, reprezintă o opera personală. Menționez că nu am plagiat o altă lucrare publicată, prezentată public sau un fișier postat pe Internet. Pentru realizarea lucrării am folosit exclusive bibliografia prezentată și nu am ascuns nici o altă sursă bibliografică sau fișier electronic pe care să le fi folosit la redactarea lucrării.

Data,
22.06.2022

Nume,
Căpușan Miruna Andreea

