

Korrelált hibák a felületi kódon

TDK DOLGOZAT

KINCSES ÁBEL

KONZULENS: DR. ASBÓTH JÁNOS
Egyetemi Docens
BME Elméleti Fizika Tanszék



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
2023

Kivonat

Ahhoz, hogy a kvantumszámítógépek valóban hasznosak lehessenek, le kell küzdeni a kvantumbiteket érő környezeti zajok hatását. Ennek egyik fontos eszköze a kvantum hibajavítás (quantum error correction), aminek legígéretesebb útja az ún. felületi kód (surface code). Itt egy logikai kvantumbitet sok fizikai kvantumbit együttes állapotába kódolnak be, és az állapotot segéd- (ancilla) kvantumbitek ismételt mérésével stabilizálják a környezeti zajokkal szemben. A felületi kód matematikailag és szimulációsan is bizonyítottan jól véd az egyes kvantumbiteken külön-külön ható hibákkal szemben, viszont a korrelált hibákkal szembeni védelem inkább csak szimulációkból ismert. Dolgozatomban a korrelált hibák hatását vizsgáltam a felületi kódra, különböző hibamodellek esetén. A kód működését áramköri szinten vizsgáltam, saját python programmal (a STIM és Pymatching csomagok segítségével). Összehasonlítottam a korreláció nélküli, illetve a korrelált hibákat. Azt találtam, a korreláció ugyan növeli a logikai hiba esélyét, de az ún. hibaküszöb értékét nem változtatja meg - legalábbis az általam vizsgált hibamodellekben. Korrelált hibák esetében a kódtávolság növelése kevésbé bizonyult hatékonynak a logikai hiba esélyének csökkentésében. Eredményeim kiegészítik a felületi kódról alkotott ismereteinket, és értékes tapasztalatokat nyújthatnak a későbbi valós kvantumszámítógépeken való kísérletekhez.

Tartalomjegyzék

1. Bevezetés	3
2. Kvantumos hibajavítás	4
2.1. Bevezetés a kvantumszámítógépekbe	4
2.2. Stabilizátorkód	4
2.3. A felületi kód	5
3. Korrelált hibák a felületi kódon	6
3.1. Hibák keletkezése	6
3.2. Korrelált hibamodell	6
3.3. Korrelált hibák javítása	7
4. Szimulációs módszer	7
4.1. Felületi kódok tesztelése	7
4.2. Felületi kódok tesztelése korrelált hibákkal	8
4.3. A szimuláció programozói megvalósítása	8
5. Eredmények	10
5.1. Korrelált és nem korrelált hibák összehasonlítása	10
5.2. Hibamodellek összehasonlítása	11
5.3. Hibaküszöb vizsgálata	12
6. Összefoglalás	15
7. Köszönetnyilvánítás	16

1. Bevezetés

A kvantumszámítógépek fejlődése az elmúlt években gyorsabb ütemet vett, több technológiai vállalat (IBM, Google) már saját kísérleti kvantumszámítógéppel rendelkezik. Ezen az eszközökön lehetőségessé vált a legalapvetőbb kvantumalgoritmusok tesztelése, és a számítógépek pontosságának meghatározása. Az immáron jelentős gyakorlati tapasztalat azt mutatja, hogy a kvantumállapotok nagyon érzékenyek a környezet hatásaira, és emiatt a kvantumbitek könnyen elveszítik a bennük tárolt információt. A számítógépet alkotó kvantumbitek hibája egy bizonyos mérték felett lehetetlenné teszi a műveletek elvégzését, és a kvantumszámítógép használhatatlan lesz. A hatékony kvantumszámítógépek megvalósítása érdekében ezt a problémát orvosolni kell. A jelenleg legígéretesebb módszer az úgynevezett felületi kód [Fowler et al.(2012)Fowler, Mariantoni, Martinis, and Cleland] megvalósítása, ez a módszer több kvantumbiten tárol el egy kvantumbitnyi információt. Előnye, hogy könnyű megvalósítani, hátránya pedig, hogy sok kvantumbitet igényel.

2. Kvantumos hibajavítás

A kvantumos hibajavítás megértéséhez először a kvantumszámítógép működését ismeretnem, majd rátérek a hibajavító kódok elméletére. A fejezet a felületi kód gyakorlati megvalósításával ér véget.

2.1. Bevezetés a kvantumszámítógépekbe

A kvantumszámítógépeket qubitek alkotják, amik a valóságban kétállapotú kvantumrendszereknek felelnek meg. Egy ilyen rendszer állapotát egy normált, két komplex dimenziós vektor írja le, és az ezen értelmezett fizikailag releváns transzformációk mind unitér operátorok. Több kvantumbit együttes állapota a kvantumbitek terének tenzorszorzataként létrejött vektortérben van, ily módon a bázisállapotok száma a qubitekével exponenciálisan nő. Az átláthatóság kedvéért vezessük be az n qubites rendszer i -edik qubitjén ható $\hat{U}_i$ leképezést a következő módon:

$$\hat{U}_i = \hat{I}^{\otimes(i-1)} \otimes \hat{U} \otimes \hat{I}^{\otimes(n-i)}. \quad (1)$$

Könnyen belátható, hogy ha $i \neq j$ akkor $[U_i, U_j] = 0$, azaz a különböző biteken ható operátorok függetlenek egymástól. A kvantuminformatikában fontos szerepet játszanak a Pauli-mátrixok, amelyet a témában a következő jelöléssel használunk:

$$\hat{X} = \hat{\sigma}_x = \hat{\sigma}_1, \hat{Y} = \hat{\sigma}_y = \hat{\sigma}_2, \hat{Z} = \hat{\sigma}_z = \hat{\sigma}_3. \quad (2)$$

Az ezekből az önadjungált unitér transzformációkból, és az identitásmátrixból (I) tenzorszorzattal generált, úgynevezett Pauli-stringek $\mathbf{P}$ algebrája alkotja a hibajavító kódok alapjait.

A szimulációs módszer miatt fontos még említést tennem a Clifford-csoportról. A Clifford-csoportot azok a $\hat{C}$ operátorok alkotják, amik Pauli-stringet Pauli-stringbe visznek át, azaz

$$\hat{C}\hat{P}\hat{C}^\dagger \in \mathbf{P} \mid \forall \hat{P} \in \mathbf{P}. \quad (3)$$

2.2. Stabilizátorkód

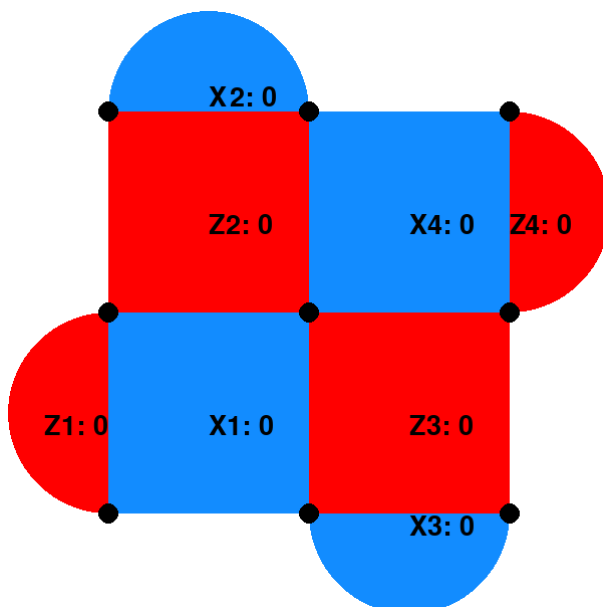
A felületi kód egy stabilizátor kód, amely több kvantumbit együttes állapotában tárol el egy kvantumbitnyi információt [Terhal(2015)]. A fizikai megvalósításban az információt együttesen hordozó, úgynevezett dataqubitek egy síkba rajzolható gráf csúcsain helyezkednek el egymás mellett, és ezek között a hibajavításhoz használt segéd-, azaz ancilla-qubitek találhatók. A rendszer matematikai leírásához a dataqubitek állapotát fogjuk vizsgálni. Ezen az állapoton úgynevezett stabilizáló méréseket fogunk végrehajtani. Egy ilyen stabilizátormérés a rendszert a stabilizátort alkotó Pauli-string valamely sajátállapotába fogja vinni, illetve a mérés eredménye (+1, vagy -1) a sajátállapothoz tartozó sajátérték. Ha megfelelő mennyiségű egymással kommutáló stabilizátort mérünk ki, akkor az azok +1 sajátértékhez tartozó sajátállapotai által alkotott alter pont 2 dimenziós lesz, azaz analóg az egy kvantumbit állapotát leíró térrel.

A hibajavítás úgy valósul meg, hogy ezeket a stabilizátorokat bizonyos időközönként megmérjük, ez alatt az idő alatt a környezet valószínűleg kis mértékben módosította a

dataqubitek állapotát. A stabilizátormérések eredményeivel nyomon tudjuk követni a környezeti behatásokat, hiszen tudjuk, hogy mely operátorok melyik sajátállapotában van rendszer. Az eredmények ismeretében összeállíthatunk egy olyan operátort, ami vissza-viszi az állapotot a kódoló altérbe, és közben nagy valószínűséggel kijavítja az okozott hibát.

2.3. A felületi kód

A felületi kódot síkban elrendezett qubitek alkotják. A kódon ható stabilizátorok szomszédos kvantumbiteket hasonlítanak össze, ez moduláris felépítést, és nagy méretet tesz lehetővé. A hibajavító kódok egy fontos tulajdonsága a kódtávolság, ami azt mutatja meg, hogy minimum hány qubiten kell ahhoz hatni, hogy a kód által tárolt logikai qubit információját a kódoló altérből való kilépés nélkül megváltoztassuk. A legegyszerűbb ilyen $d = 3$ távolságú kód az 1. ábrán látható.



1. ábra. A képen egy saját programból vett $d = 3$ kódtávolságú ún. surface-17 felületi kód ábrája látható.

Az ábrán színesen jelölt négyzetek, vagy félkörök a stabilizátorok elhelyezkedését mutatják. Egy ilyen plakethez tartozó stabilizátor a csúcsain lévő qubitek együttes X, illetve Z paritását méri ki, ahogy az 1. ábrán is látszik.

Az 1-es, 2-es, 4-es, illetve 5-ös qubiteken ható együttes X paritást kimérő S_1 stabilizátor például:

$$\hat{S}_1 = \hat{X}_1 \hat{X}_2 \hat{X}_4 \hat{X}_5. \quad (4)$$

A javítás során, ha pontosan egy qubitnek a négy közül megváltozott az X paritása, akkor a stabilizátor ezt jelzi. Mivel egy qubit általában egynél több operátorban szerepel, a paritás megváltozását nagy valószínűséggel egy másik stabilizátormérés eredménye is jelzi, ezekből pedig kikövetkeztethetjük, hogy mely qubiten történt hiba. Az összes stabilizátor eredményét szindrómának nevezzük.

Minden szindróma esetében meghatározható egy olyan Pauli-string, amely a hibát a legnagyobb valószínűséggel kijavítja. Ennek megkeresése azonban sok időt, és nagy számítási erőforrást igényel, ezért egy egyszerűbb, régóta használt matematikai módszerhez, az operációkutatásból ismert minimális súlyú teljes párosításhoz (minimum weight perfect matching) folyamodunk.

3. Korrelált hibák a felületi kódon

A korrelált hibákkal való foglalkozáshoz érdemes áttekinteni, hogyan is jöhetnek létre. A fizikai megfontolásokat alapul véve meghatározhatunk olyan modelleket, amik segítségével könnyen szimulálni lehet ezeknek hatásait. A következő részekben tehát megvizsgálunk egy lehetséges hibaforrást, ismertetem az általam használt hibamodelleket, és a hibajavítás menetét.

3.1. Hibák keletkezése

A felületi kód analitikailag és szimulációsan is bizonyítottan jól véd az egyes kvantum-biteken egymástól függetlenül fellépő hibákkal szemben. A független azonos eloszlású hibamodellemben minden dataqubiten keletkezhet egy Pauli-hiba P_{fiz} fizikai hiba valószínűséggel, és azt vizsgáljuk, hogy egy kör stabilizátormérés során mekkora eséllyel veszítjük el a tárolt állapotot a fizikai hiba függvényében.

A gyakorlati megvalósítás során a hibákat sokszor nagy energiájú részecskék becsapódása okozza, amely a környezetében felhevíti a kvantumszámítógépet, így a becsapódás körül korreláltan okoz hibát a rendszerben. Szerencsétlen esetben ez a teljes kvantumszámítógép "lefagyását" eredményezi, azaz az összes információ, és számítási eredmény elveszik [AI(2023), McEwen(2022)]. Kutatásom során az ilyen nagyobb területen zavart okozó hibák mentén indultam el, és szimulációs módszerrel vizsgáltam a hatásaikat.

3.2. Korrelált hibamodellel

A korrelált hibák leírásához a Fowler-Martinis cikk [Fowler and Martinis(2014)] hibamodelljét vettem alapul. Az elképzelés lényege, hogy a korrelált hibák mindig egy kvantumbit körül alakulnak ki, és a hibás terület nagyságát egy eloszlás határozza meg. Mindegyik kvantumbit egymástól függetlenül ugyanakkora eséllyel válhat egy ilyen korrelált hiba központjává, így két stabilizátormérés között akár több korrelált hiba is felléphet, ezek területei át is fedhetnek. Annak az esélyét, hogy a központi qubit körül legalább r sugarú körben hiba keletkezett a biteken, egy $P(P_0, r)$ hibafüggvény adja meg, ahol P_0 annak a valószínűsége, hogy a központi qubit állapota sérül. Egy ilyen nagyenergiás részecske becsapódása dekoherenciához vezet, viszont a dataqubitek a mérések során visszatérnek a stabilizátorok sajátterébe, így ezek a hibák helyettesíthetők Pauli-hibákkal.

Két hibamodelt használtam, az exponenciálisat:

$$P(P_0, r) = P_0 C^{-r} = \mathbb{P}(X \text{ hibák } r \text{ sugarú körlapon adott qubit körül}) \quad (5)$$

és a polinomiálisit

$$P(P_0, r) = \begin{cases} \frac{P_0}{10r^C} & , \text{ ha } r > 0 \\ P_0 & , \text{ ha } r = 0. \end{cases} \quad (6)$$

Az egyenletekben a C egy, a korreláció mértékét meghatározó, konstans paraméter. Minél nagyobb C értéke, annál jobban hasonlít a hibaeloszlás a független hibamodellre. Kis C az erősen korrelált eseteket jelenti, így ekkor nagyobb eséllyel szenved egyszerre hibát több kvantumbit.

A Fowler-Martini cikk alapján az r számításánál két normát is használtam, a szokásos euklideszi távolságot generáló $p = 2$ -es normát, illetve az úgynevezett Manhattan-metrikát generáló $p = 1$ -es normát. A $p = 1$ -es norma azért hasznos, mert a felületi kódon négyzetlátszon elhelyezkedő szomszédos qubitek között szorosabb a kapcsolat, így a gyakorlatban a hiba is ilyen irányban terjed gyorsabban. Egy $\mathbf{v}$ vektor p -edik normájának általános képlete:

$$\|\mathbf{v}\|_p = \left(\sum_{i=1}^n |v_i|^p \right)^{\frac{1}{p}}. \quad (7)$$

3.3. Korrelált hibák javítása

A korrelált hibák a szokásos független modellhez képest bonyolultabb eloszlással okoznak szindrómákat. A stabilizátormérérek eredménye alapján ki lehetne következtetni, hogy milyen kvantumbitek körül, és mekkora mértékben alakult ki hiba. Ezt a dekódolási módszert nevezik maximum likelihood decoder-nek. Az egyik lehetséges megvalósítása egy táblázat használata, ami minden lehetséges szindrómához tartalmazza a legvalószínűbb javítást. Ez a hibatábla, és a maximum likelihood decoder algoritmusok általában, exponenciálisan skálázódnak a hibajavító kódokkal, így megvalósításuk nehézkes, lassú és számításigényes.

Az általam használt megoldás az, hogy a minimális súlyú teljes párosítás alkalmazásához megalkotok egy olyan hibagráfot, ami nem veszi figyelembe a korrelációt. A hatást a gráf súlyozásánál úgy kompenzáltam, hogy minden egyes kvantumbitnél figyelembe vettem, hogy saját magán kívül milyen másik qubit körül kialakult korrelációban lehet érintett. Ez a megoldás a maximum likelihood hibajavításnál sokkal kevésbé számításigényes, gyorsabb, és könnyebb megvalósítani.

4. Szimulációs módszer

A szimulációs módszer leírásához először bemutatom, hogyan szokták a felületi kód hibátűrését megállapítani. A szokásos módszerhez viszonyítva ismertetem, hogyan alkalmaztam a korrelált hibamodelleket, majd rátérek a szimuláció programozói megvalósítására.

4.1. Felületi kódok tesztelése

A felületi kód működése közben nagyon sok minden okozhat hibát. A fő hibaforrásokat azonban el lehet különíteni egymástól, így a szimulációba külön modelleket lehet alkalmazni.

Az általam vizsgált zajforrás a környezeti zaj. Mivel a környezeti behatások a kvantumszámítógép pillanatnyi működésétől nem függenek, ezen hibák jellemzően két stabilizátormérés között jelentkeznek. Ennek leírására használatos a mérési körök közötti független Pauli-hiba, én ezt a hibamodellt egészítettem ki korrelációval a fentebb említett módon. A többi hibatípus forrása a kvantumszámítógép belső működéséből adódik.

A második fajtája a hibáknak a kvantumkapuk hibái. A kvantumbiteken ható operátorok ugyanis zajt okozhatnak a rendszerben, mégpedig egyesével, a kvantumkapuban (gate) résztvevő qubiteken. Az előzőekhez hasonlóan a stabilizátorok projekciója miatt célszerű szintén Pauli-hibákkal modellezni ezt a zajforrást is.

A harmadik hibafajta a kiolvasási hiba. Ez azt jelenti, hogy egy qubit kimérésénél a mérés eredménye nem az lesz, mint ami a kvantumbit állapota, ily módon hamis adatot láthatunk. Esetünkben a dataqubitek állapotát kevésbé befolyásolja, hiszen ezeken csak a szimuláció végeztével hajtunk végre méréseket. A hibajavítás szempontjából viszont veszélyes típusnak számít, mert a stabilizátorméréseket befolyásolhatják, hiszen a stabilizáló méréseket fizikailag a segédkvantumbitek kimérésével valósítjuk meg.

A hibával terhelt felületi kódok futtatása során bevett szokás, hogy ezeknek az elkülönített hibamodelleknek ugyanazt a hibaértéket adjuk meg, és így hajtjuk végre a szimulációt. Példának okáért egy kvantumbit ugyanakkora valószínűséggel szenved hibát két stabilizátormérés között, mint egy X kapu alkalmazása esetén. Ennek a közös hibaértéknek a P_{fiz} fizikai hiba a szokásos neve, mert a fizikailag valós kvantumbitek hibarátaját mutatja meg. A szimuláció célja annak a vizsgálata, hogy a felületi kódban tárolt logikai kvantumbit mekkora eséllyel veszíti el az információját, azaz mekkora a logikai hiba.

4.2. Felületi kódok tesztelése korrelált hibákkal

A korrelált hibamodellek alkalmazásához a fent említettek szerint a környezeti zaj hibamodelljét terjesztettem ki. Az alapelv továbbra is az, hogy ezek minden qubiten azonos valószínűséggel okoznak hibát, de itt a zaj a szomszédos bitekre is továbbterjedhet. A hibamodellek paramétereiben ekkor $P_0 = P_{\text{fiz}}$, azaz a fizikai hiba a központi qubit hibájának a valószínűsége. Ez a megközelítés azért célszerű, mert alacsony korreláció mellett a független hibamodellhez hasonló működést fogunk megfigyelni, és a korreláció mértékének növelésével követhetjük az ettől való eltérést. Az eltérésben a legfőbb különbség, hogy a több szomszédal rendelkező belső qubitek kevésbé lesznek megbízhatóak, mint a peremen lévő lényegesen kevesebb közeli kvantumbittel rendelkező társaik, és ez a hibajavító kód méretével, így a kódtávolsággal lesz összefüggésben. A valóság hű szimuláció érdekében a többi hibamodellt nem vettem ki a kódból, viszont nem módosítottam őket. A tapasztalat azt mutatta, hogy a korreláció hatása így is tisztán látszódott a szimulációs eredményeken.

4.3. A szimuláció programozói megvalósítása

Szimulációim [Kincses(2023)] elkészítéséhez a Craig Gidney féle STIM [Gidney(2021)] python package verzióját használtam. Ez egy Clifford szimulátor [Aaronson and Gottesman(2004)], ami Heisenberg-képben [Gottesman(1998)] operátorok sajátállapotaként tárolja el a kvantumállapotokat. A rendszer akkor hatékony, ha a szimuláció kezdetén a kvantumbitek

együttes állapota Pauli-stringek sajátállapota, és csak Clifford-csoportbeli operátorokat használ, ld. (3). egyenlet.

A STIM jól össze van hangolva a pymatching-gel [Higgott(2022)], ami egy olyan csomag (package), ami a minimális súlyú teljes párosítást alkalmazza a hibajavításra. A hibák javításához tehát nem kellett megalkotnom egy gráfot, csak létrehoztam egy olyan alternatív kvantumáramkört, ami a korrelált hibák kivételével minden paraméterében egyezik a javítandó kvantumkörrel. Ennek a javítókörnek a segítségével a sinter python package (<https://pypi.org/project/sinter/>) automatikusan meghatározza a javításhoz használandó gráfot, és elvégzi a szimulációt.

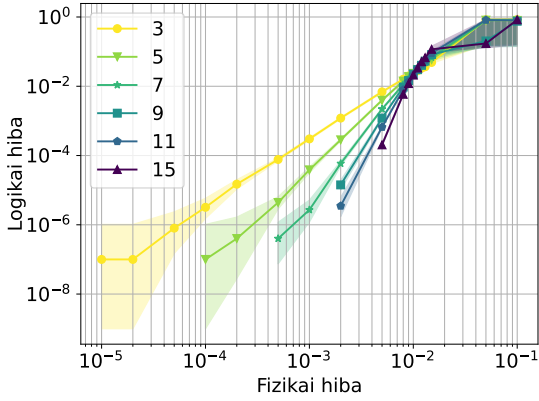
A szimuláció során a felületi kódon 10 kör stabilizáló mérést végeztem, majd összehasonlítottam a rendszer kezdeti, és a végső állapotát. Amennyiben a kettő egyezett, akkor a kísérlet sikeres volt, ellenkező esetben ez egy hiba. A szimulációt 5000 db. hibáig, vagy 10^6 db. lefutott szimulációig folytattam. Ezt a kísérletet végeztem el $d = 3, 5, 7, 9, 11$, illetve 15 kódtávolságú rendszereken, és különböző mértékű fizikai hibák esetében. Az eredmények megjelenítésénél fontos, hogy a logaritmikus skálán nem tudtam megjeleníteni azt az esetet, amikor az összes kísérlet sikeres volt, így amelyik kódtávolsághoz és fizikai hibához nincsen feltüntetve a szimuláció eredménye, ott a szimuláció során 0 lett a logikai hiba relatív gyakorisága.

5. Eredmények

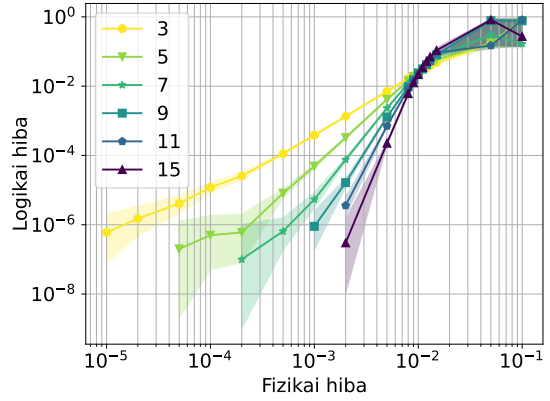
Az eredmények vizsgálatát a korrelált, és a hagyományos (nem korrelált, független azonos eloszlású) hibamodell összehasonlításával kezdtem. Ezután összehasonlítottam egymással a különböző hibamodellek eredményeit, majd külön kitértem a hibaküszöb vizsgálatára.

5.1. Korrelált és nem korrelált hibák összehasonlítása

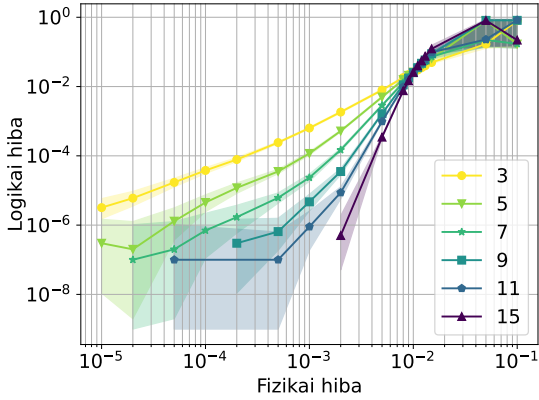
Először az exponenciális hibamodellt (5) $p = 1$ -es norma mellett vizsgáltam, a hibajavításnál pedig a részletesebb, qubit-specifikus hibagráfot alkalmaztam.



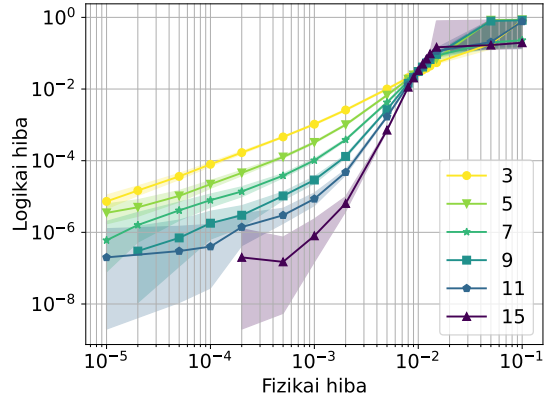
(a) Különböző $d = 3, 5, \dots, 15$ kódtávolságú felületi kódok hibatűrése korreláció nélkül.



(b) Exponenciális hibamodell $p = 1$ -es normával és $C = 40$ -es korrelációval.



(c) Exponenciális hibamodell $p = 1$ -es normával és $C = 10$ -es korrelációval.



(d) Exponenciális hibamodell $p = 1$ -es normával és $C = 5$ -ös korrelációval.

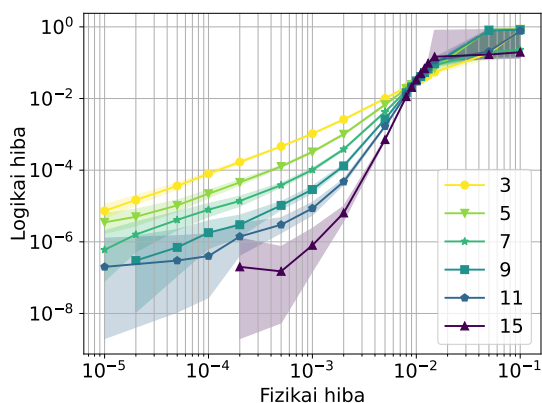
2. ábra

A 2. ábra grafikonjain látható, hogy a hibák közti korreláció növekedésével (a C értékének (5) csökkenésével) a hibajavítás hatékonysága csökken. A küszöb alatti hibarátnál (például 10^{-3} fizikai hiba környékén) a logikai hiba valószínűsége a korreláció növekedésével jelentősen nő. Talán a leglátványosabban ez úgy mutatkozik meg, hogy

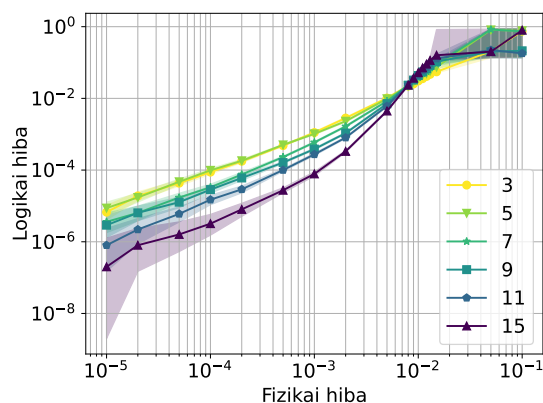
korreláció jelenlétében a nagy kódtávolságnál és alacsony fizikai hiba jelenlétében is előfordultak logikai hibák a szimuláció során – ezért pl. a 2 d) ábrán a legsötétebb, $d = 15$ kódtávolsághoz tartozó görbe még 10^{-3} fizikai hiba esetén is nullától különböző értéket vesz fel, míg a gyengébb korreláció vizsgálata során minden hibát sikeresen kijavított a rendszer.

5.2. Hibamodellek összehasonlítása

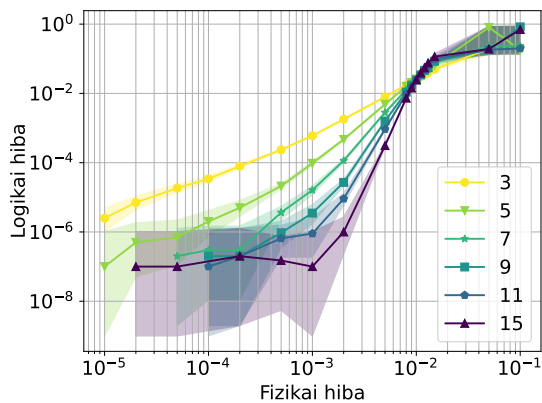
A következőkben erős, $C = 5$ -ös korreláció mellett összehasonlítottam a polinomiális, és az exponenciális hibamodellt. Azért választottam az erős korrelációt, mert így markánsabban látszódik a hibamodellek hatása az eredményeken.



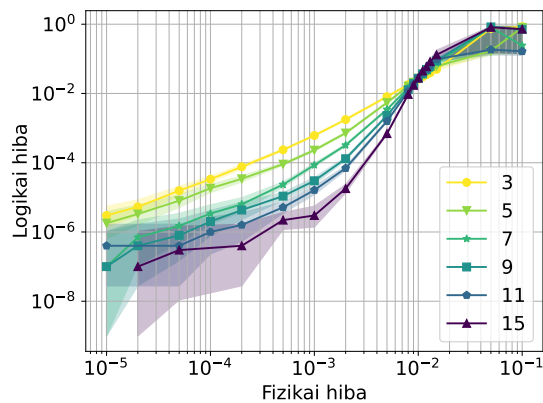
(a) Exponenciális hibamodell $p = 1$ -es normával és $C = 5$ -es korrelációval.



(b) Exponenciális hibamodell $p = 2$ -es normával és $C = 5$ -es korrelációval.



(c) Polinomiális hibamodell $p = 1$ -es normával és $C = 5$ -es korrelációval.



(d) Polinomiális hibamodell $p = 2$ -es normával és $C = 5$ -es korrelációval.

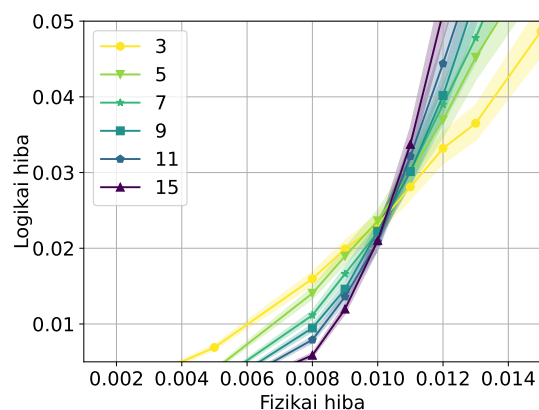
3. ábra

A $p = 2$ -es norma mellett a 3. ábrán jól láthatóan kilaposodtak a nagyobb kódtávolsághoz tartozó görbék. A legnagyobb hatást az exponenciális modell váltotta ki, ott a különböző hibamodellek értékei egy nagyságrenden belül kerültek egymáshoz. Ha ezeket az eredményeket összevetjük a várakozásainkkal, akkor azt tapasztaljuk, hogy a $p = 2$ -es

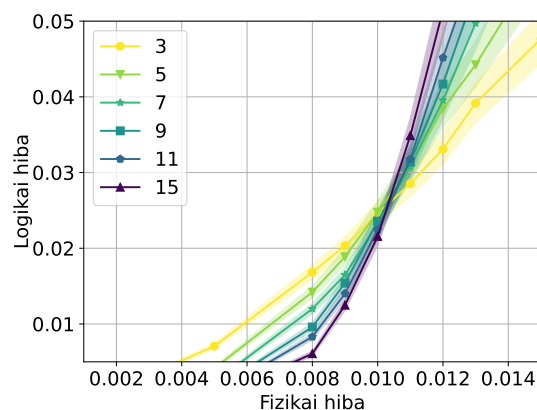
norma valóban nagyobb hatással lesz, hiszen ugyanakkora r sugár esetén több qubitet érint, mint a $p = 1$ -es társa. A hibamodellekkel kapcsolatban viszont azt gondolná az ember, hogy az exponenciális jobban cseng le, mint a polinomiális, ezért a polinomiális modellre lesz rosszabb a hibajavítás hatásfoka. Az eredmény azzal magyarázható, hogy a kódok mérete miatt a kialakult hibák sugarának tartománya kicsi, és ebben a környezetben a polinomiális hibamodell cseng le gyorsabban.

5.3. Hibaküszöb vizsgálata

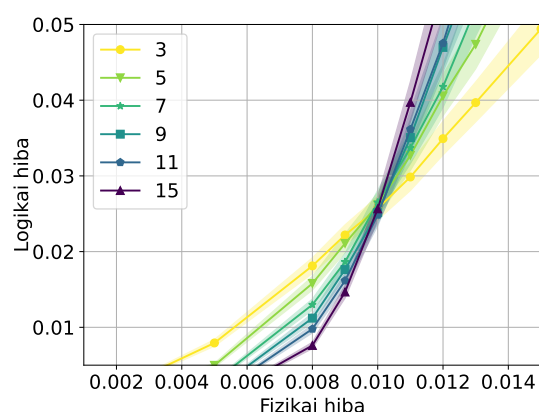
A felületi kódok azon az elven működnek, hogy minél több kvantumbiten tároljuk el az információt, annál kisebb eséllyel veszítjük el azt. Azonban a fizikai hibának van egy olyan értéke, ami felett már egyáltalán nem hatékony a felületi kód, ezt nevezzük hibaküszöbnek. A hibaküszöb nagysága azért fontos, mert ha a kvantumszámítógépen a qubitek hibája kisebb, mint a hibaküszöb, akkor a bitek számának növelésével biztosan javítani tudjuk a számítógép pontosságát.



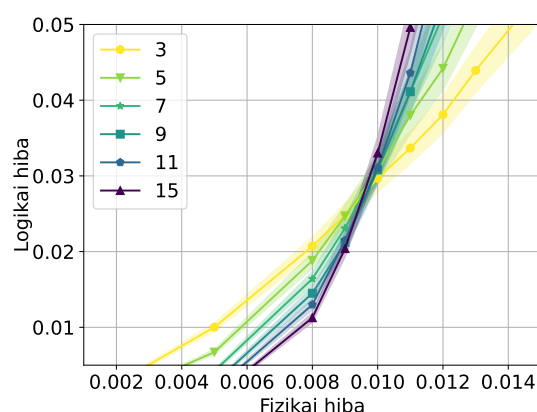
(a) Különböző $d = 3, 5, \dots, 15$ kódtávolságú felületi kódok hibaküszöbke korreláció nélkül.



(b) Exponenciális hibamodell $p = 1$ -es normával és $C = 40$ -es korrelációval.



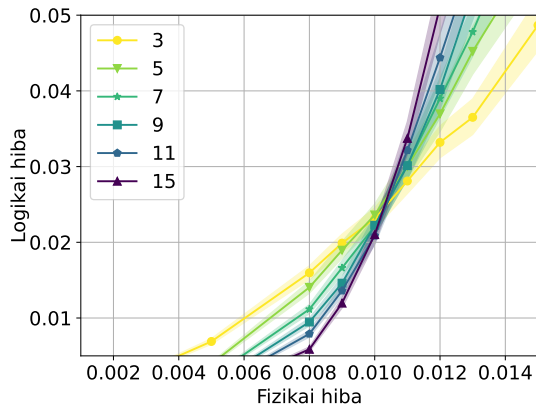
(c) Exponenciális hibamodell $p = 1$ -es normával és $C = 10$ -es korrelációval.



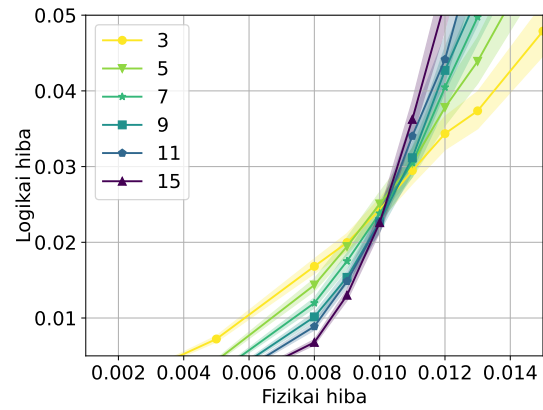
(d) Exponenciális hibamodell $p = 1$ -es normával és $C = 5$ -ös korrelációval.

4. ábra

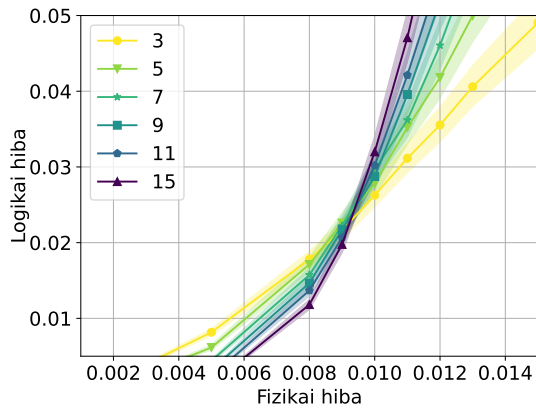
Először az exponenciális hibamodellt vizsgáltam meg, és azt tapasztaltam, hogy a $p = 1$ -es norma mellett csak nagyon kis mértékben változik meg, tulajdonképpen végig $P_k = 0.01$ körül marad. A 4. ábrán láthatóan a hibaküszöb C értékével együtt csökken, viszont ezt csak nagyon kismértékben teszi. A $p = 2$ -es norma használatával az 5. ábrán látható eredmény nagyon hasonló, viszont az erősen korrelált $C = 5$ esetben a hibaküszöb már számottevően $P_k = 0.007$ -re csökkent. Érdekesnek tartom, hogy csak az erősen korrelált esetben csökkent a küszöb, a közepes $C = 10$, és gyenge $C = 40$ esetekben nem észlelhető a változás.



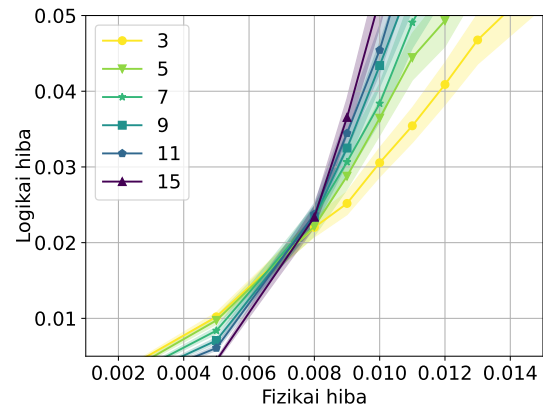
(a) Különböző $d = 3, 5, \dots, 15$ kódtávolságú felületi kódok hibaküszöbe korreláció nélkül.



(b) Exponenciális hibamodell $p = 2$ -es normával és $C = 40$ -es korrelációval.



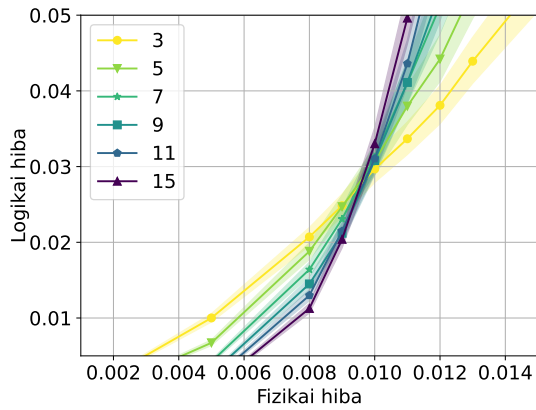
(c) Exponenciális hibamodell $p = 2$ -es normával és $C = 10$ -es korrelációval.



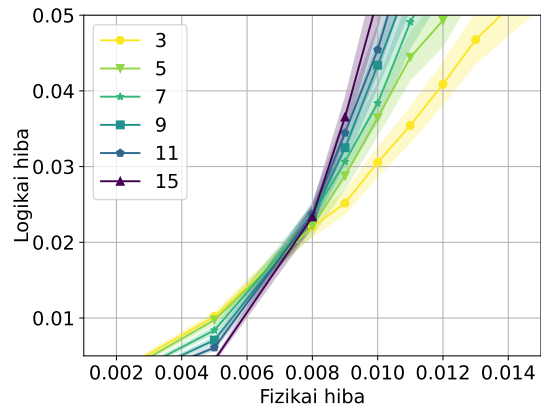
(d) Exponenciális hibamodell $p = 2$ -es normával és $C = 5$ -ös korrelációval.

5. ábra

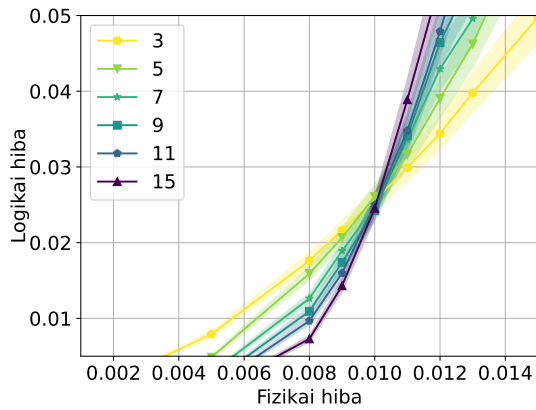
A hibaküszöböket a legerősebben korrelált esetben a 6. ábrán látható módon összehasonlítva megállapíthatjuk, hogy lényeges eltérést a nem korrelált esetben tapasztalt hibaküszöbtől egyedül a $p = 2$ -es norma használatával számolt exponenciális hibamodell produkált. A $C = 5$ -nél kisebb mértékben korrelált kísérleteknél természetesen még inkább közelebb volt a hibaküszöb a nem korrelált viszonyítási alaphoz, vagy egyáltalán nem tért el tőle.



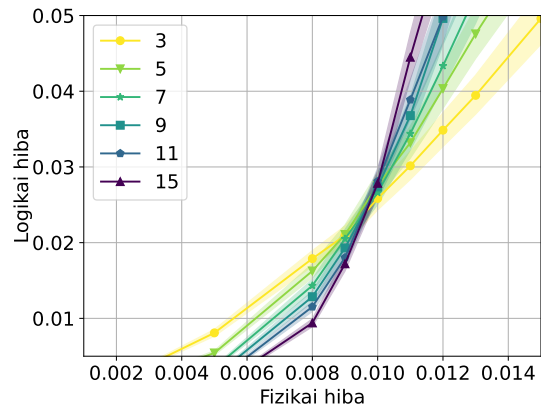
(a) Exponenciális hibamodell $p = 1$ -es normával és $C = 5$ -es korrelációval.



(b) Exponenciális hibamodell $p = 2$ -es normával és $C = 5$ -es korrelációval.



(c) Polinomiális hibamodell $p = 1$ -es normával és $C = 5$ -es korrelációval.



(d) Polinomiális hibamodell $p = 2$ -es normával és $C = 5$ -es korrelációval.

6. ábra

6. Összefoglalás

A dolgozatban röviden bemutattam a kvantumszámítógép matematikai leírását, és a felületi kód működését. Bevezettem a felületi kódok a korrelált hibákat, amit két hibamoddelllel írtam le. Mivel mindkét hibamoddell esetében a távolságok kiszámítására kétféle normát, a $p = 1$, illetve $p = 2$ típusút is használtam, ezért gyakorlatilag 4 különböző korrelált hibamoddellel dolgoztam.

Munkám során megállapítottam, hogy a korrelált hibák kezelhető mértékben növelik a felületi kód hibarátaját. A hibaküszöböt alig változtatják meg, így a küszöb alatti környezetben továbbra is hatékony módszer a kódtávolság növelése a hibaráta csökkentésére. Érdekes módon a hibaküszöbnél néhány nagyságrenddel alacsonyabb tartományban a kódtávolság növelése már kevésbé bizonyult hatékonynak a logikai hiba csökkentésében, mint a nem korrelált hibamoddellek esetében.

A másik tapasztalat, hogy az eddig bevált minimális súlyú teljes párosítás legegyszerű-

rűbb megvalósítása továbbra is jól használható a hibák kijavítására, még akkor is, ha azok korreláltnak jelennek meg a felületi kódon.

Eredményeim biztatóak a felületi kód jövőjére, és gyakorlati alkalmazására nézve, hiszen a korrelált hibákkal szembeni védelem méginkább alkalmassá teszi a valós kvantumszámítógépeken való használatra. Ennek köszönhetően tetszőlegesen pontos logikai kvantumbitek képezhetőek, amik felhasználhatóak egy több ezer kvantumbites számítógépben.

7. Köszönetnyilvánítás

A dolgozat háttérét adó kutatás a Kulturális és Innovációs és Minisztérium ÚNKP-23-6-II-BME-439 kódszámú Új Nemzeti Kiválóság Programjának a Nemzeti Kutatási, Fejlesztési és Innovációs Alapból nyújtott támogatásával, illetve a 2022-2.1.1-NL-2022-00004 számú projekt finanszírozásában valósult meg.

Szeretném megköszönni Dr. Asbóth János témavezetőm munkáját, aki idejét és energiáját nem sajnálva segített a kutatásban, és utat mutatott a dolgozat írásában. Nélküle ez a munka, és a kutatás nem jöhetett volna létre.

Köszönettel tartozom Penc Patriknak a TDK felületen való tájékozódásért, illetve a dolgozat címlapjának megalkotásában nyújtott segítségéért.

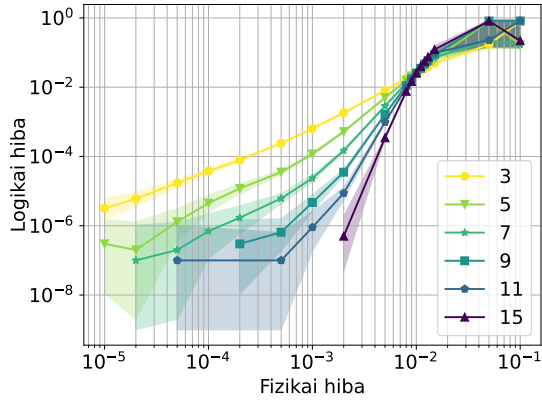
További köszönettel tartozok Márton Áronnak, akitől a STIM használatát elsajátíthattam.

Hivatkozások

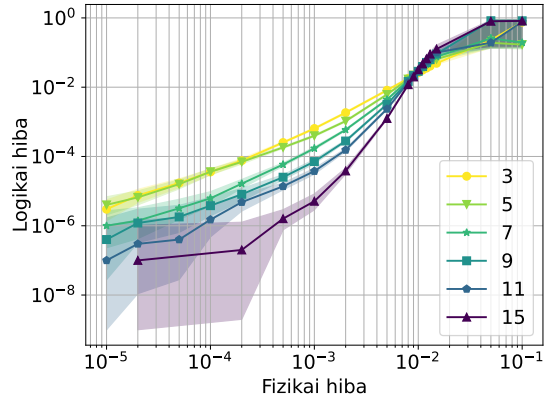
- [Fowler et al.(2012)]Fowler, Mariantoni, Martinis, and Cleland] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86(3), September 2012. doi: 10.1103/physreva.86.032324. URL <https://doi.org/10.1103/physreva.86.032324>.
- [Terhal(2015)] Barbara M. Terhal. Quantum error correction for quantum memories. *Reviews of Modern Physics*, 87(2):307–346, April 2015. doi: 10.1103/revmodphys.87.307. URL <https://doi.org/10.1103/revmodphys.87.307>.
- [AI(2023)] Google Quantum AI. Suppressing quantum errors by scaling a surface code logical qubit. *Nature*, 614(7949):676–681, February 2023. doi: 10.1038/s41586-022-05434-1. URL <https://doi.org/10.1038/s41586-022-05434-1>.
- [McEwen(2022)] Matt McEwen. Resolving high-energy impacts on quantum processors, 2022. URL <https://blog.research.google/2022/01/resolving-high-energy-impacts-on.html?m=1>.
- [Fowler and Martinis(2014)] Austin G. Fowler and John M. Martinis. Quantifying the effects of local many-qubit errors and nonlocal two-qubit errors on the surface code. *Physical Review A*, 89(3), March 2014. doi: 10.1103/physreva.89.032316. URL <https://doi.org/10.1103/physreva.89.032316>.
- [Kincses(2023)] Ábel Kincses. Korrelált hibák a felületi kódon - python kód, November 2023. URL <https://github.com/KincsesABEL/korrelalt-hibak>.
- [Gidney(2021)] Craig Gidney. Stim: a fast stabilizer circuit simulator. *Quantum*, 5:497, July 2021. doi: 10.22331/q-2021-07-06-497. URL <https://doi.org/10.22331/q-2021-07-06-497>.
- [Aaronson and Gottesman(2004)] Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5), November 2004. doi: 10.1103/physreva.70.052328. URL <https://doi.org/10.1103/physreva.70.052328>.
- [Gottesman(1998)] Daniel Gottesman. The heisenberg representation of quantum computers. 1998. doi: 10.48550/ARXIV.QUANT-PH/9807006. URL <https://arxiv.org/abs/quant-ph/9807006>.
- [Higgott(2022)] Oscar Higgott. PyMatching: A python package for decoding quantum codes with minimum-weight perfect matching. *ACM Transactions on Quantum Computing*, 3(3):1–16, June 2022. doi: 10.1145/3505637. URL <https://doi.org/10.1145/3505637>.

Függelék

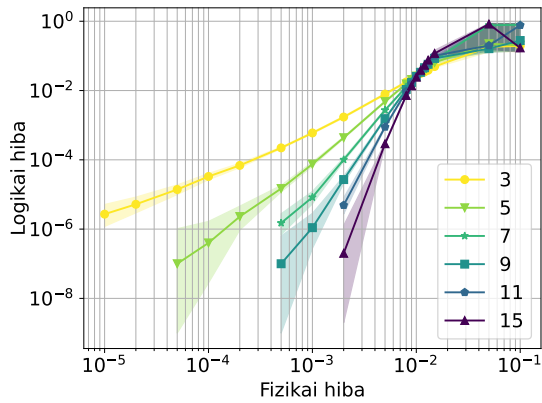
A hibamodellek összehasonlítását az eredményeknél a $C = 5$ -ös legerősebb korrelációnál mutattam be. A 7. ábra $C = 10$ -es, a 8. ábra $C = 40$ -es korrelációnál szemlélteti a felületi kód viselkedését.



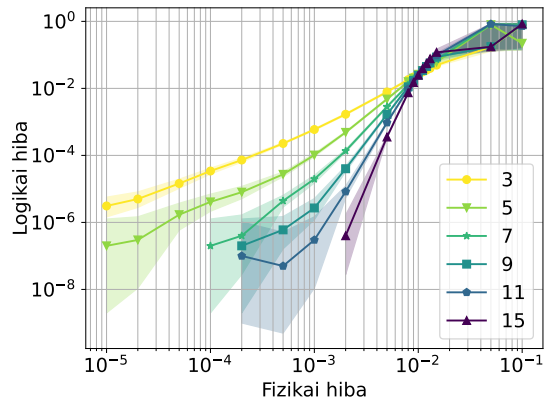
(a) Exponenciális hibamodell $p = 1$ -es normával és $C = 10$ -es korrelációval.



(b) Exponenciális hibamodell $p = 2$ -es normával és $C = 10$ -es korrelációval.

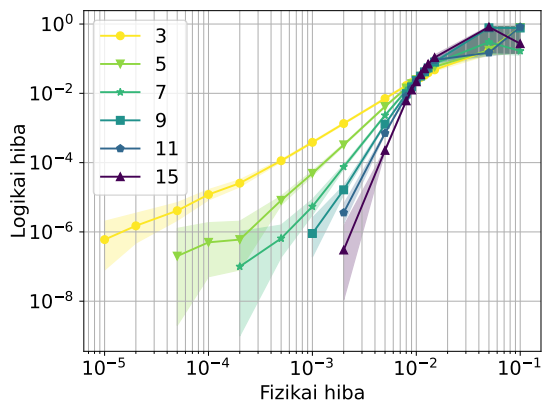


(c) Polinomiális hibamodell $p = 1$ -es normával és $C = 10$ -es korrelációval.

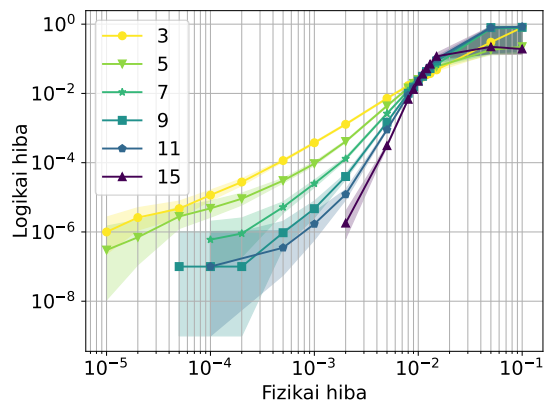


(d) Polinomiális hibamodell $p = 2$ -es normával és $C = 10$ -es korrelációval.

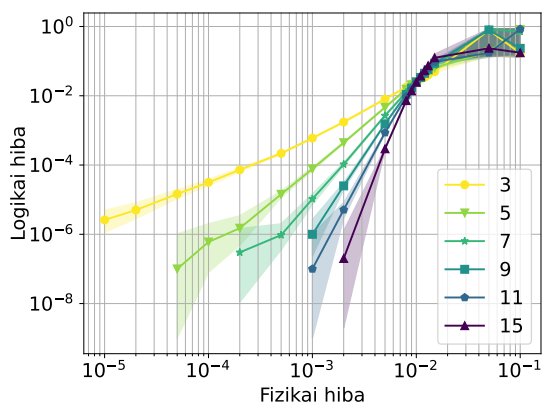
7. ábra



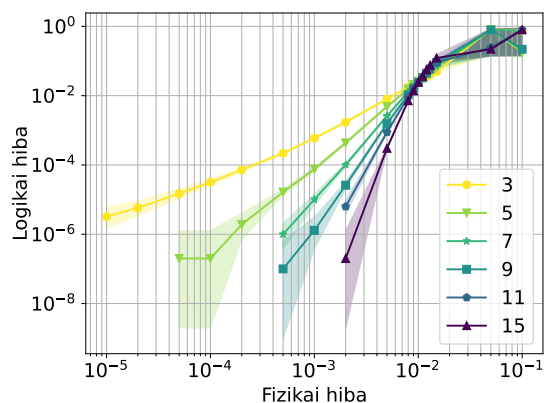
(a) Exponenciális hibamodell $p = 1$ -es normával és $C = 40$ -es korrelációval.



(b) Exponenciális hibamodell $p = 2$ -es normával és $C = 40$ -es korrelációval.



(c) Polinomiális hibamodell $p = 1$ -es normával és $C = 40$ -es korrelációval.



(d) Polinomiális hibamodell $p = 2$ -es normával és $C = 40$ -es korrelációval.

8. ábra