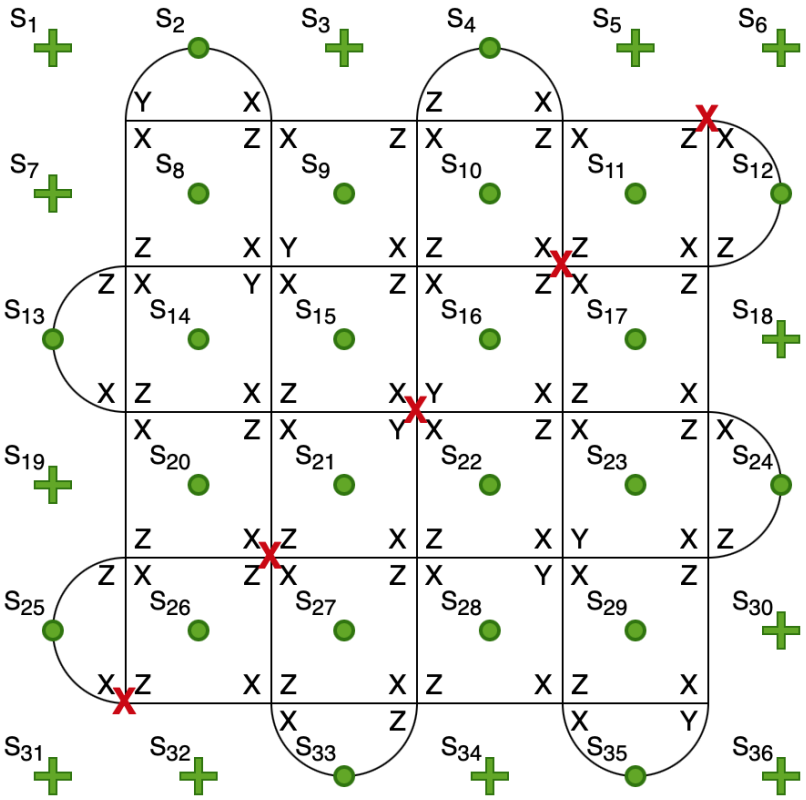
CHALMERS
UNIVERSITY OF TECHNOLOGY



Quantum error correcting Clifford deformed surface codes

A study of analytical and tensor network based solutions for XZZX, XY and XZZXdY qubit error correcting surface codes

Kandidatuppsats vid institutionen för Fysik

Lukas Christersson, Jacob Olsson, Adrian Sandberg, Adam Udén

INSTITUTIONEN FÖR FYSIK
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2024
www.chalmers.se

Kandidatuppsats 2024

Kvantfelskorrigerande Clifforddeformerade ytkoder

En studie av analytiska och tensornätverks baserade lösningar för XZZX, XY och XZZXdY kvantbitsfelkorrigerande ytkoder

LUKAS CHRISTERSSON,
JACOB OLSSON,
ADRIAN SANDBERG,
ADAM UDÉN

CHALMERS
UNIVERSITY OF TECHNOLOGY

Institutionen för Fysik
Chalmers Tekniska Högskola
Göteborg, Sverige 2024

Kvantfelskorrigerande Clifforddeformerade ytkoder

Handledare: Mats Granath, Institutionen för Fysik
Examinator: Jan Swenson, Institutionen för Fysik

Kandidatarbete 2024
Institutionen för Fysik
Chalmers Tekniska Högskola
SE-412 96 Göteborg
Telefon +46 31 772 1000

Omslagsbild: XZZXdY-koden med kodstorlek $d = 5$. Inritat är även alla stabilisatorer samt den logiska X_L-operatorn på högerdiagonalen.

Typsatt i LaTeX

Abstract

Qubits are very sensitive to noise and, if quantum computers are to ever become practically usable, methods to counteract the effect of noise on the qubits are a necessity. In this report the error correcting capability of three different Clifford deformed surface codes, namely XZZX-, XY- and XZZXdY-codes, were examined, both analytically and numerically under phase-biased noise. The latter is a new code not previously described in the literature, to the best of our knowledge.

The analytical solution for the logical error rate P_f subject to bit-flip-, phase-flip- and both bit-flip- and phase-flip-noise with probability p_x, p_z and p_y following a distribution $p_x = p_y = p_z/(2\eta)$ at the specific error rate $p_z = 1 - p$, where all syndromes have the same error rate, was found by mapping the surface codes to a generalized Ising model. The total logical failure rate was calculated to be given by

$$P_f = \frac{3}{4} - \frac{1}{4}e^{-2d_Z \operatorname{artanh}(1/(2\eta))},$$

where $d_Z = d$ for the XZZX-code, $d_Z = d^2$ for the XY-code, and $d_Z = 2d-1$ for the XZZXdY-code, where d is the code-distance.

The logical failure rate corresponding to physical failure rates up to the special point and code distances between 5 and 99 was calculated using a tensor network decoder based on the Python package *qecsim*. The obtained results corresponded well to the analytical solution at $p_z = 1-p$ for all codes. However, at lower physical error rates the numerical results for the XY- and XZZXdY-codes were found to be unreasonable. The reason for this is not known and requires further investigation.

Sammandrag

Kvantbitar är väldigt känsliga för brus och om kvantdatorer någonsin skall blir praktiskt användbara behövs metoder för att motverka brusets effekt. I denna rapport undersöktes felkorrigeringsförmågan hos tre Clifford-deformerade ytkoder, benämnda XZZX, XY och XZZXdY, dels analytiskt och dels numeriskt under fasviktat brus. Den senare är en ny ytkod som, till vår kännedom, ej tidigare beskrivits i litteraturen.

Genom en mappning av ytkoderna på en generaliserad Isingmodell kunde en analytisk lösning för den logiska felsannolikheten P_f vid bit-flip, fas-flip och både bit- och fas-flipbrus med sannolikhet p_x, p_z respektive p_y vid den specifika felsannolikheten $p_x = p_y = p_z/(2\eta)$ med $p_z = 1 - p$, där alla syndrom har samma felsannolikhet, beräknas. Den logiska felsannolikheten räknades ut till

$$P_f = \frac{3}{4} - \frac{1}{4} e^{-2d_Z \operatorname{artanh}(1/(2\eta))},$$

där $d_Z = d$ för XZZX-koden, $d_Z = d^2$ för XY-koden och $d_Z = 2d - 1$ för XZZXdY-koden, där d är koddistansen.

Med hjälp av en tensornätverksavkodare baserad på Python-paket *qecsim* beräknades den logiska felsannolikheten P_f för fysiska felsannolikheter upp till den speciella punkten samt koddistanser från 5 till 99. Resultaten överensstämde väl med de analytiska resultaten i punkten $p_z = 1 - p$, men vid lägre fysiska felsannolikheter gav XY- och XZZXdY-koden orimliga resultat. Anledningen till detta kräver vidare undersökningar.

Innehåll

1 Inledning

Klassiska datorer sparar och hanterar information genom något som kallas för en bit [1]. En bit kan antingen vara en 0:a eller en 1:a och datorn lagrar information som en sträng av dessa. Informationen kan sedan processeras genom att datorn manipulerar dessa bitar med hjälp av olika operationer som ändrar vissa 0:or till 1:or och vice versa. En kvantdator fungerar på liknande sätt i den mån att även den hanterar sin information genom bitar, men till skillnad från den klassiska datorn är dessa kvantbitar inte bundna till att bara vara antingen en 0:a eller en 1:a [2]. Istället har kvantbitarna två tillstånd som utgör beräkningsbasen som ofta kallas för $|0\rangle$ och $|1\rangle$, men kan sen, till skillnad från den klassiska biten, befinna sig i oändligt många tillstånd däremellan som någon superposition av dessa.

Gemensamt för både klassiska bitar och kvantbitar är att de kan falla offer för slumpmässiga bit-fel, där en eller flera bitar oavsiktligt ändras till ett annat tillstånd. Bit-fel förekommer mycket mer frekvent hos kvantbitar än för klassiska bitar eftersom de är väldigt känsliga för brus som enkelt kan störa deras superpositionerade tillstånd [3]. För klassiska bitar kan bit-fel dessutom enkelt motverkas genom att ha flera bitar som lagrar samma information i en så kallad logisk bit [4]. Denna metoden går dock inte att applicera rakt av på kvantbitar eftersom man aldrig kan veta vilket tillstånd en given kvantbit befinner sig i utan att mäta på biten, vilket skulle kollapsa systemet och förstöra kvanttillståndet. Sålunda måste felen i kvantbitar upptäckas och korrigeras utan direkta mätningar eller kunskap om att ett fel har, eller inte har, inträffat.

Ett sätt att hantera detta problemet på är med hjälp av något som kallas för en *ytkod* [5][6]. En ytkod konstrueras av ett gitter med flera kvantbitar som bildar den motsvarande logiska kvantbiten. Paritetsmätningar, en typ av mätningar som inte förstör superpositionen, kan sedan genomföras på kvantbitarna i den logiska kvantbiten. Om eventuella fel har uppstått kommer paritetsmätningarna ge upphov till ett så kallat *syndrom*. Emellertid kan flera olika fel ge upphov till samma syndrom, vilket är problematiskt då de nödvändigtvis inte tillhör samma *ekvivalensklass*, det vill säga att felen inte rättas på samma sätt. Korrigeringen måste således baseras på den mest sannolika ekvivalensklassen för ett givet syndrom.

Flera fysiska implementationer av felkorrektion har redan börjat undersökas [7][8]. Med anledning av detta undersöks även flera olika typer av ytkoder specialiserade på exempelvis en viss typ av brus [9][10]. En sådan typ av ytkod är *Clifford-deformerade ytkoder* (CDSC) [11]; dessa genereras genom att tillämpa så kallade *Hadamard-operatorer* på de ingående kvantbitarna på en ytkod.

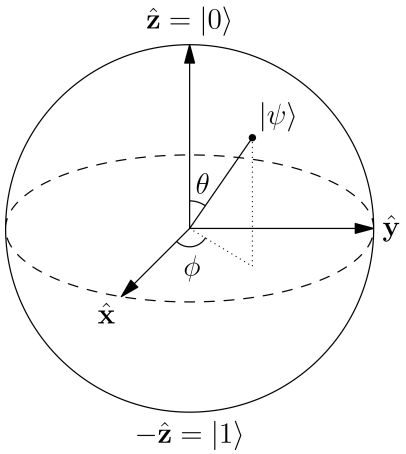
Projektet ämnar således att studera och jämföra felkorrigeringsförmågan hos tre olika CDSCs, kallade XZZX, XY och XZZXdY, specialiserade på fasviktat brus genom att analytiskt och numeriskt beräkna sannolikheten för fel i den logiska kvantbiten som funktion av dess koddistans och jämföra resultaten med varandra. Lösningarna till XZZX- och XY-koden är en reproduktion av lösningarna i en tidigare studie av Y. Xiao, B. Srivastava och M. Granath [2] medan XZZXdY, till vår kännedom, är en ny kod som inte tidigare undersökts. Den analytiska lösningen tas fram med hjälp av en mappning av ytkoderna på en Isingmodell, medan de numeriska beräkningarna utförs med hjälp av en Python-baserad tensornätverksavkodare.

2 Bakgrund

I följande avsnitt kommer den nödvändiga teorin som ligger till grund för projektet att presenteras och förklaras. Först introduceras en kort bakgrund om kvantsystem och kvantbitar samt notation. Därefter introduceras felkorrigering av kvantdatorer och en möjlig lösningsmetod för detta: ytkoder. Vidare presenteras hur en ytkod kan mappas till Isingmodellen för att kunna lösa felsannolikheterna analytiskt. Sist introduceras en tensornätverksbaserad avkodare som används för att numeriskt lösa felsannolikheter på ytkoderna.

2.1 Kvantbitar och Blochsfären

Inom kvantfysiken beskrivs fysiska system med hjälp av linjär algebra i ett flerdimensionellt Hilbertrum [12]. Systemets tillstånd kan beskrivas med en vektor, kallad för en ket, $|\psi\rangle$. I ett N-dimensionellt Hilbertrum ($N \in \mathbb{N}$) kan en ket uttryckas som en N-dimensionell vektor. Skalärprodukten mellan två tillstånd skrivs som $\langle\phi|\psi\rangle$ där $\langle\psi| = |\psi\rangle^\dagger$, alltså $|\psi\rangle$ hermitska konjugat. I en kvantdator används tillstånd på formen $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ där $\alpha, \beta \in \mathbb{C}$ och $\alpha^2 + \beta^2 = 1$ [13]. Tillståndet $|\psi\rangle$ benämns *qubit* eller *kvantbit*. En kvantbit kan i princip vara vilket kvanttillstånd som helst med två distinkta energiegentillstånd och fysiska system som används för att realisera kvantbitar är exempelvis jonfällor [14] eller supraledande kretsar [15]. Tillståndet kan alternativt parametriseras med två vinklar $\theta \in [0, \pi]$ och $\phi \in [0, 2\pi]$, $|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle$ [16]. Denna parametrisering går att identifiera med en enhetssfär, Blochsfären, där tillstånden $|0\rangle$ och $|1\rangle$ utgör polerna, se figur 1. De två grundtillstånden $|0\rangle$ och $|1\rangle$ representeras även ofta genom vektorer, $|0\rangle = \binom{1}{0}$ och $|1\rangle = \binom{0}{1}$, och ett generellt tillstånd $|\psi\rangle$ skrivs som en linjärkombination av dessa.



Figur 1: Visualisering av tillståndet $|\psi\rangle$ genom parametrisering med två vinklar. Ytan som spänns upp kallas för Blochsfären. [17], CC BY-SA 3.0

2.2 Operatorer

Alla fysiska storheter som kan mätas beskrivs inom kvantfysiken med en Hermitsk operator, $\hat{O}$, vilken verkar som en linjärtransformation på ett tillstånd $|\psi\rangle$ [12]. Vid mätning av storheten O som motsvarar operatorn $\hat{O}$ på ett allmänt tillstånd $|\psi\rangle$ kommer mätningens utslag vara ett av egenvärdena till $\hat{O}$ och under mätning kommer tillståndet kollapsa till motsvarande egenvärdes egentillstånd. Hatten över en operator utelämnas ofta när det ändå tydligt framgår av kontexten att det är en operator som menas. För att exakt kunna veta två olika storheter på samma

system samtidigt krävs således att motsvarande operatorer delar egentillstånd. Detta kallas att operatorerna är kompatibla och kan matematiskt undersökas genom operatorernas kommuteringsförhållande. För kompatibla operatorer gäller att $[\hat{A}, \hat{B}] = 0$ medan för inkompatibla operatorer är $[\hat{A}, \hat{B}] \neq 0$ där $[\hat{A}, \hat{B}] \equiv \hat{A}\hat{B} - \hat{B}\hat{A}$. Vanligt förekommande är även anti-kommuteringsoperatorn $\{\hat{A}, \hat{B}\} \equiv \hat{A}\hat{B} + \hat{B}\hat{A}$.

Inom kvantberäkning är vanliga operatorer Paulioperatorerna, kallade $\hat{X}, \hat{Y}, \hat{Z}$, och Hadamardoperatorn $\hat{H} = \frac{1}{\sqrt{2}}(\hat{X} + \hat{Z})$. Eftersom tillstånden $|0\rangle$ och $|1\rangle$ skrivs på vektorform kan operatorerna skrivas som matriser

$$\hat{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \ \hat{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \ \hat{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \ \hat{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Det inses lätt genom att applicera operatorerna på grundtillstånden att:

- $\hat{X}$ motsvarar en bit-flip, alltså $\hat{X}|0\rangle = |1\rangle$ och $\hat{X}|1\rangle = |0\rangle$.
- $\hat{Z}$ motsvarar en fas-flip, $\hat{Z}|0\rangle = |0\rangle$ och $\hat{Z}|1\rangle = -|1\rangle$.
- $\hat{Y}$ motsvarar en kombination av fas-flip och en bit-flip samt lägger till en fas på båda tillstånden, $\hat{Y}|0\rangle = i|1\rangle$ och $\hat{Y}|1\rangle = -i|0\rangle$.
- $\hat{H}$ skapar en superposition av de två grundtillstånden, $\hat{H}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ och $\hat{H}|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

Om tillstånden visualiseras på Blochsfären medför $\hat{X}$ en rotation med π radianer runt x-axeln, $\hat{Z}$ en rotation med π radianer runt z-axeln, $\hat{Y}$ en rotation med π radianer runt y-axeln och $\hat{H}$ en rotation med $\pi/2$ radianer runt y-axeln och π radianer runt x-axeln.

2.3 Felkorrigering, stabilisatorer och generatorer

För att motverka fel i kvantberäkningar till följd av brus från omgivningen kan en första strategi vara att låta flera individuella kvantbitar lagra samma information i så kallade repetitionskoder [13]. Systemet av de individuella kvantbitarna kallas för en logisk kvantbit, och den logiska nollan och ettan kan exempelvis definieras som följande

$$|0\rangle_L = |0\rangle \otimes |0\rangle \otimes |0\rangle \otimes ... = |000...\rangle,$$
$$|1\rangle_L = |1\rangle \otimes |1\rangle \otimes |1\rangle \otimes ... = |111...\rangle.$$

Ett allmänt tillstånd kan följaktligen uttryckas som $|\psi\rangle_L = \alpha|0\rangle_L + \beta|1\rangle_L$. För att finna huruvida ett fel har inträffat hade idealt en mätning av den logiska kvantbitens ingående fysiska kvantbitar genomförts, vilket man exempelvis gör i en klassisk dator [4]; detta går dock inte att göra i en kvantdator eftersom en mätning hade inneburit att superpositionen förstörts. Istället för direkta mätningar av kvantbitarnas tillstånd används därför så kallade paritetsmätningar, vilka inte förändrar systemet utan fel. Betrakta som exempel en logisk kvantbit som består av tre individuella kvantbitar. Om operatorerna $Z_1Z_2 = \hat{Z} \otimes \hat{Z} \otimes \hat{I}$ och $Z_2Z_3 = \hat{I} \otimes \hat{Z} \otimes \hat{Z}$ appliceras på den logiska kvantbiten fås

$$Z_1Z_2\left(\alpha|000\rangle + \beta|111\rangle\right) = \alpha|000\rangle + \beta|111\rangle,$$
$$Z_2Z_3\left(\alpha|000\rangle + \beta|111\rangle\right) = \alpha|000\rangle + \beta|111\rangle,$$

och tillståndet är oförändrat efter mätningen. Antag nu att en $\hat{X}$-operator har verkat på antingen första, andra eller tredje kvantbiten i tillståndet, alltså att *ett* bit-flip fel har inträffat. Detta ger tillstånden och utfallet från mätningar av Z_1Z_2 och Z_2Z_3 som

$$\begin{aligned}
|\psi\rangle_L = \alpha|100\rangle + \beta|011\rangle &\implies Z_1Z_2\text{: -1, } Z_2Z_3\text{: 1,} \\
|\psi\rangle_L = \alpha|010\rangle + \beta|101\rangle &\implies Z_1Z_2\text{: -1, } Z_2Z_3\text{: -1,} \\
|\psi\rangle_L = \alpha|001\rangle + \beta|110\rangle &\implies Z_1Z_2\text{: 1, } Z_2Z_3\text{: -1.}
\end{aligned}$$

Alltså ger en mätning av Z_1Z_2 respektive Z_2Z_3 utfallet $+1$ om första och andra, respektive andra och tredje kvantbiten är samma och -1 om de är olika, därav namnet paritetsmätningar. Vid endast *ett* bit-fel kan alltså felet entydigt bestämmas med dessa mätningar. Om det däremot inträffar fler än ett bit-flip fel så kommer felet inte längre att vara entydigt bestämt. Så länge sannolikheten för ett fel är mindre än sannolikheten för flera fel, så är de mest troliga felen fortfarande de som diskuterats ovan. Således blir målet av kvantfelskorrigering att identifiera och rätta det mest sannolika felet utifrån utfallet av paritetsmätningar, vilka benämns felets syndrom. Viktigt att ta hänsyn till är att en kvantdator, till skillnad från en klassisk dator, inte enbart kan utsättas för bit-flip fel utan även för fas-flip fel, vilka även de behöver korrigeras.

Felkorrigeringen ovan kan formaliseras genom införandet av stabilisatorer, vilka ovanstående paritetsmätningar var ett exempel på. Betrakta ett system bestående av n stycken fysiska kvantbitar som kodar k stycken logiska kvantbitar. Kodrummet $\mathcal{H}_{CS}$ är definierat som ett underrum till systemets hela Hilbertrum $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$ sådant att

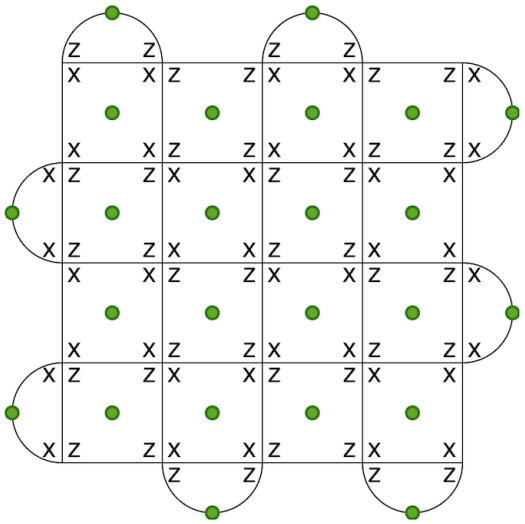
$$\mathcal{H}_{CS} = \{|\psi\rangle \in \mathcal{H} : s|\psi\rangle = |\psi\rangle | \forall s \in \mathcal{S}\},$$

där $\mathcal{S}$ benämns stabilisatorgruppen [13]. Kodrummet definieras alltså som mängden av tillstånd som förblir oförändrat vid applikation av element ur stabilisatorgruppen. Stabilisatorgruppen är i sin tur en undergrupp till Pauligruppen, $\mathcal{P} = \{cf_1 \otimes f_2 \otimes \cdots \otimes f_n | f_i \in \{\hat{I}, \hat{X}, \hat{Y}, \hat{Z}\}, c \in \{\pm 1, \pm i\}\}$, sådan att $\forall s_i, s_j \in \mathcal{S}$ gäller $[s_i, s_j] = 0$. Stabilisatorgruppen beskrivs bäst genom sina generatorer $s_1, ...s_m$, vilka är det minsta antalet element som krävs för att kunna återskapa hela stabilisatorgruppen genom multiplikation av generatorerna. Detta betecknas som $\mathcal{S} = \langle s_1, ..., s_m \rangle$. För att koda k stycken logiska kvantbitar krävs det att stabilisatorgruppen består av $m = n - k$ antal generatorer.

2.4 Ytkoder

Ett sätt att utveckla idéen med repetitionskoder är att arrangera kvantbitarna i ett tvådimensionellt gitter med stabilisatorgeneratorer utplacerade mellan kvantbitarna. Ytkodens storlek, vilket mäts av koddistansen, har betydelse för kodens prestanda. Koddistansen betecknas i denna studie som d och representerar antalet kvantbitar längs en sida av ytkoden. Observera att d måste vara ett udda tal för att ytkoden skall bibehålla alla sina egenskaper vid större koddistanser. Den minsta kodstorleken vi betraktar är $d = 3$. Denna studie kommer att utgå ifrån en standardytkod, kallad XZ-koden [5][6], som visas i figur 2. I varje vertex av koden sitter en kvantbit och i varje ruta sitter en stabilisatorgenerator (illustrerad som en grön punkt), ofta kallad enbart stabilisator. Stabilisatorerna består av fyra operatorer (två för stabilisatorerna vid kanterna) som verkar på varsin individuell kvantbit. Det felfria tillståndet kan egentligen definieras godtyckligt [2], så vi väljer för enkelhetens skull tillståndet då alla stabilisatorer mäter $+1$. På gittret definieras även två logiska operatorer X_L och Z_L, samt deras produkt $Y_L = X_LZ_L$. Dessa kommuterar med alla

stabilisatorer, men anti-kommuterar med varandra. Flera ekvivalenta val av logiska operatorer kan göras då produkten mellan en logisk operator och någon kombination av stabilisatorer även det är en logisk operator.



Figur 2: XZ-koden med X- och Z-stabilisatorer markerade med gröna punkter.

Från XZ-koden kan fler ytkoder genereras genom att applicera operatorer på kvantbitarna. Detta kallas att Clifford-deformera ytkoden [11]. Hadamardoperatorn, $\hat{H}$, och Hadamard-YZ operatorn, $\hat{H}_{YZ} = \hat{H}\sqrt{\hat{Z}}\hat{H}$ kan användas för att deformera XZ-koden. Om vi applicerar dessa operatorer på en kvantbits och stabilisators gemensamma tillstånd så erhålls

$$\hat{H}\left(\hat{Z}|\psi\rangle\right) = \hat{H}\hat{Z}|\psi\rangle = \hat{H}\hat{Z}\hat{H}^{\dagger}\hat{H}|\psi\rangle = \hat{X}\left(\hat{H}|\psi\rangle\right),$$
$$\hat{H}_{YZ}\left(\hat{Z}|\psi\rangle\right) = \hat{H}_{YZ}\hat{Z}|\psi\rangle = \hat{H}_{YZ}\hat{Z}\hat{H}_{YZ}^{\dagger}\hat{H}_{YZ}|\psi\rangle = \hat{Y}\left(\hat{H}_{YZ}|\psi\rangle\right),$$

där vi använt att $\hat{H}^{\dagger}\hat{H} = \hat{I}$ [18]. På samma sätt kan det visas att andra kombinationer av Hadamardoperatorer och stabilisatorer inte förändrar stabilisatorn. Vidare vet vi att Hadamardoperatorn som appliceras direkt på kvantbiten inte påverkar vår lösningsmetod, eftersom vi alltid kan omdefiniera ytkodens felfria tillstånd till att alla stabilisatorer på $\hat{H}|\psi\rangle$ ska mäta $+1$ istället för på $|\psi\rangle$. Vi har alltså att

$$\hat{H}: \quad \hat{Z} \leftrightarrow \hat{X},$$
$$\hat{H}_{YZ}: \quad \hat{Z} \leftrightarrow \hat{Y},$$

medan andra stabilisatorer förblir oförändrade. I denna studie kommer tre Clifford-deformerade ytkoder att undersökas: XZZX, XY och en som vi kallar XZZXdY. XZZX-koden fås genom att applicera $\hat{H}$ på varannan kvantbit i XZ-koden. XY-koden fås genom att applicera $\hat{H}_{YZ}$ på alla kvantbitar i XZ-koden. XZZXdY fås genom att applicera $\hat{H}_{YZ}$ på alla kvantbitar på vänsterdiagonalen i XZZX-koden.

2.4.1 Felkedjor och ekvivalensklasser

Varje individuell kvantbit kan utsättas för tre olika typer av fel: bit-flip, fas-flip eller bit- och fas-flip. Detta motsvaras, som tidigare nämnts i avsnitt 2.2, av att verka med en $\hat{X}$-, $\hat{Z}$- respektive $\hat{Y}$-operator på kvantbiten. Sannolikheten för respektive typ av fel skrivs som p_x, p_z samt p_y.

Identitetsoperatorn $\hat{I}$ motsvarar fallet då inget fel har uppstått och har sannolikheten $1-p$, där $p = p_x + p_y + p_z$. Denna studie följer en tidigare studie av Y. Xiao, B. Srivastava och M. Granath [2] och kommer precis som den endast att studera fasviktat brus med sannolikhetsfördelningen $p_x = p_y = \frac{p_z}{2\eta}$ där $\eta \geq 1/2$. Ytkodens totala fel kallas för en *felkedja* och är tensorprodukten mellan alla individuella kvantbitars "feloperatorer"; $C \in \{\hat{I}, \hat{X}, \hat{Y}, \hat{Z}\}^{\otimes d^2}$. Felkedjorna ger upphov till att vissa av stabilisatorerna mäter -1 istället för $+1$ och som tidigare diskuterats i avsnitt 2.3 är detta syndromet för felkedjan. Felkedjorna har dock inte unika syndrom då flera felkedjor kan ge upphov till samma syndrom. Det går emellertid att dela in felkedjorna i fyra olika ekvivalensklasser $\{\mathcal{I}, \mathcal{X}, \mathcal{Z}, \mathcal{Y}\}$ [2]. Ekvivalensklasserna defineras som följande:

- $C \in \mathcal{I}$: $[X_L, C] = [Z_L, C] = 0$.
- $C \in \mathcal{X}$: $[X_L, C] = \{Z_L, C\} = 0$. Felkedjan C motsvarar en logisk bit-flip.
- $C \in \mathcal{Z}$: $\{X_L, C\} = [Z_L, C] = 0$. Felkedjan C motsvarar en logisk fas-flip.
- $C \in \mathcal{Y}$: $\{X_L, C\} = \{Z_L, C\} = 0$. Felkedjan C motsvarar både en logisk bit- och fas-flip.

Från en felkedja $C \in \mathcal{I}$ kan felkedjor i de andra ekvivalensklasserna fås genom att verka med de motsvarande logiska operatorerna på felkedjan [2].

Vi gör nu två observationer som kommer att vara centrala för den analytiska lösningen.

1. Givet en felkedja C_0 med ett givet syndrom kan en ny felkedja C_1 erhållas som har samma syndrom genom att verka med en stabilisator på kvantbitarna.
2. Felkedjan C_1 kommer att tillhöra samma ekvivalensklass som C_0.

Den första punkten stämmer eftersom kvantbitarnas tillstånd inte förändras när vi verkar med stabilisatorerna och då syndromet direkt bestäms av kvantbitarnas tillstånd så kommer inte heller syndromet att förändras. Den andra punkten stämmer eftersom stabilisatorerna kommuterar med de logiska operatorerna. När vi verkar med en logisk operator förändrar vi den logiska kvantbitens tillstånd. Men eftersom stabilisatorerna varken kan förändra den logiska kvantbitens tillstånd eller den logiska operatorn så kan den heller inte ändra ekvivalensklassen.

När fel på de individuella kvantbitarna uppstår fås alltså ett syndrom. Eftersom syndromet är all information som kan fås av systemet utan att förstöra dess superposition [13] vill vi därför beräkna vilken ekvivalensklass som ett givet syndrom mest sannolikt tillhör, för att sedan kunna åtgärda felet utan att kollapsa superpositionen. Rättningen av en felkedja sker genom applikation av en *korrektionskedja*, vilken korresponderar mot en sekvens av operatorer som används för att motverka felen i systemet [2]. Denna studie kommer identifiera korrektionskedjor med hjälp av en avkodare. Från en felkedja C erhålls dess korrektionskedja C' genom avbildningen $D : C \to C' = D(C)$, där D är själva avkodaren. Då felkedjan i praktiken inte är känd, utan endast dess syndrom, är det syndromet avkodaren appliceras på. Om produkten CC' inte är i ekvivalensklass $\mathcal{I}$ så har felkorrigeringen misslyckats och ett logiskt fel kvarstår.

2.5 Koppling mellan ytkoder och statistisk mekanik

Sannolikheten att ett givet syndrom har uppstått från en av de fyra ekvivalensklasserna betecknas som $P_{\mathcal{I}}$, $P_{\mathcal{X}}$, $P_{\mathcal{Y}}$ och $P_{\mathcal{Z}}$. Vårt mål är att beräkna dessa sannolikheter. För att göra detta kommer vi att utnyttja Isingmodellen, som härstammar från statistisk mekanik och används ofta i det sammanhanget för att beskriva olika ämnens magnetiska egenskaper [19]. Isingmodellen består av ett gitter där det i varje gitterpunkt sitter ett spinn som antingen kan vara upp eller ner. Vidare är varje gitterpunkt även kopplad till sina grannar och påverkas följaktligen av deras tillstånd. Även ytkoden består av ett slags gitter, dock ett som utgörs av stabilisatorer som kan mäta +1 eller -1. Genom att kalla +1 och -1 för "spinn upp" respektive "spinn ner" kan ytkoden mappas till Isingmodellen och således kan de existerande matematiska metoderna för Isingmodellen utnyttjas för att lösa ytkoden. Vi börjar med att beskriva Isingmodellen, för att sedan beskriva hur mappningen går till och slutligen beskriva lösningsmetoden för Isingmodellen.

2.5.1 Isingmodellen

Antag att vi har ett gitter av spinn-variabler s_i som kan anta värdena ± 1 och som är kopplade till varandra genom kopplingskonstanter J_i. Energin i varje koppling fås genom att multiplicera kopplingskonstanten för kopplingen och alla ingående spinnvariabler med varandra. Exempelvis får vi för en koppling med två ingående spinnvariabler s_1 och s_2 och med kopplingskonstant J energin $E = J \cdot s_1 s_2$. Hamiltonianen $\mathcal{H}$ för systemet fås sedan genom att summera alla kopplingar. För ett en-dimensionellt system med kopplingar med endast två ingående spinnvariabler fås alltså

$$\mathcal{H}(\{s\}) = \sum_i J_i s_i s_{i+1}.$$

Med hjälp av Boltzmannfaktorn kan vi sedan översätta detta till en sannolikhet P att systemet har en specifik spinnkonfiguration $\{s\}_0$ [19]

$$P_{\{s\}_0} = \frac{1}{Z} e^{-\mathcal{H}(\{s\}_0)\cdot\beta}, \quad \text{där} \quad Z = \sum_{\{s\}} e^{-\mathcal{H}(\{s\})\cdot\beta}.$$

Här är $\beta = 1/(k_B T)$, där T är temperatur och k_B är Boltzmanns konstant. Z är tillståndssumman och består av en summa över alla möjliga spinnkonfigurationer.

2.5.2 Mappning av ytkod på en generaliserad Isingmodell

Som tidigare nämndes i avsnitt 2.4.1 uppstår ett syndrom från ett fel som i sin tur motsvaras av en felkedja C, där felkedjan kan tillhöra en av fyra ekvivalensklasser. Vi vet också att alla felkedjor som tillhör en given ekvivalensklass kan erhållas genom att verka med elementen ur stabilisatorgruppen på C. Således kan vi skriva sannolikheten att en given felkedja C tillhör en viss ekvivalensklass E enligt

$$P_{E(C)} \sim \sum_S \pi_{S\cdot C},$$

där $\pi_{S\cdot C}$ är sannolikheten för felkedjan som fås när stabilisatorn S verkar på felkedjan C [2]. Summan löper över alla element i stabilisatorgruppen.

Metoden som används för mappningen av felsannolikheten till Isingmodellen är tagen från den tidigare studien av Y. Xiao, B. Srivastava och M. Granath [2] och introducerades först i studien av E. Dennis, A. Kitaev, A. Landahl och J. Preskill [5]. Efter att metoden introducerats kommer

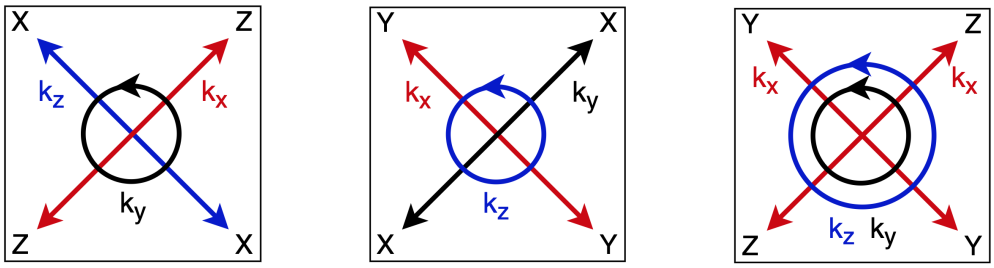
den att illustreras med ett enklare exempel, men ett komplett bevis utelämnas. Metoden lyder som följande:

1. Låt alla stabilisatorer motsvaras av en spinnvariabel s som kan anta värdena ± 1. Placera även ut spinn fixerade till $+1$ på kanterna av ytkoden, där ett spinn hade hamnat vid en större kodstorlek (se exempelvis figur 7a).

2. Introducera kopplingskonstanter $k = J \cdot \beta$. Dessa visas nedan i ekv. 1.

$$\begin{cases} k_x = -\frac{1}{4} \ln \frac{p_x(1-p)}{p_y p_z}, \\ k_y = -\frac{1}{4} \ln \frac{p_y(1-p)}{p_x p_z}, \\ k_z = -\frac{1}{4} \ln \frac{p_z(1-p)}{p_x p_y}. \end{cases} \tag{1}$$

3. Kopplingskonstanterna kopplar samman stabilisatorerna som agerar på samma kvantbit enligt figur 3 nedan. En rak pil innebär en "2-koppling", ks_1s_2, medan en cirkulär pil innebär en "4-koppling", $ks_1s_2s_3s_4$. Hamiltonianen för systemet definieras sedan på samma sätt som i Isingmodellen, fast nu med stabilisatorerna som spinnvariabler och kopplingar och kopplingskonstanter som beskrivet.



Figur 3: Kopplingarna mellan stabilisatorerna över en kvantbit. Dubbelsidiga pilar indikerar en 2-koppling och cirkulära pilar indikerar en 4-koppling.

4. När ett X-fel uppstår på en kvantbit så byts tecknet på kopplingskonstanterna k_y och k_z vid den kvantbiten. För Y- och Z-fel byts tecknet på k_x,k_z respektive k_x,k_y.

5. Att verka på felkedjan med en stabilisator motsvaras av att byta tecken på stabilisatorns motsvarande spinnvariabel.

Med en Hamiltonian som är konstruerad enligt punkterna 1-4 kan vi beräkna sannolikheten av en viss felkedja för ett visst syndrom med hjälp av Boltzmannfaktorn. Vi vill dock ha sannolikheten för *ekvivalensklasserna* givet ett visst syndrom. Som tidigare nämnts kan alla felkedjor med samma syndrom och i samma ekvivalensklass fås genom att verka med stabilisatorerna på felkedjan. Sannolikheten för ekvivalensklassen fås således genom att summera alla Boltzmannfaktorer för alla möjliga sätt att verka med stabilisatorer på en ursprunglig felkedja. Enligt punkt 5 innebär detta att byta tecken på spinnvariablerna, och det framgår därför att sannolikheten för en ekvivalensklass blir tillståndssumman i Isingmodellen.

Som nämnt kommer vi inte visa att denna konstruktion stämmer i allmänhet, utan kommer endast troliggöra det med ett exempel. Som exempel tar vi XZ-koden som sågs i figur 2, men

med godtycklig koddistans d. Börja med en felkedja utan fel. Sannolikheten för denna felkedja är $P = (1-p)^{d^2}$. Antag sedan att fyra X-fel har inträffat. Sannolikheten för denna felkedja är $P' = p_x^4(1-p)^{d^2-4}$. Den relativa felsannolikheten är då $P'/P = p_x^4/(1-p)^4$. Vi vill nu ställa upp Hamiltonianerna för dessa felkedjor enligt metoden ovan. Stabilisatorerna runt varje kvantbit är kopplade till varandra med tre kopplingar k_x, k_z och k_y. I den första felkedjan är alla spinn +1, och vi får $\mathcal{H}\beta = (k_x + k_y + k_z) \cdot d^2$. I den andra kedjan har vi fyra X-fel, vilket byter tecken på k_y och k_z i dessa kopplingar. Vi får därför $\mathcal{H}'\beta = (k_x - k_y - k_z) \cdot 4 + (k_x + k_y + k_z) \cdot (d^2 - 4)$. Den relativa sannolikheten fås nu från Boltzmannfaktorerna

$$\frac{e^{-\mathcal{H}'\beta}}{e^{-\mathcal{H}\beta}} = \exp[-(k_x - k_y - k_z) \cdot 4 - (k_x + k_y + k_z) \cdot (d^2 - 4) + (k_x + k_y + k_z) \cdot d^2] = \exp[8 \cdot (k_z + k_y)]$$
$$= \exp\left[-2\left(\ln\frac{p_y(1-p)}{p_x p_z} + \ln\frac{p_z(1-p)}{p_x p_y}\right)\right] = \left(\frac{p_y(1-p)p_z(1-p)}{p_x p_z p_x p_y}\right)^{-2} = \frac{p_x^4}{(1-p)^4},$$

vilket är den relativa felsannolikheten vi förväntade oss. Observera att samma resultat även kan erhållas genom att verka med en stabilisator, det vill säga byta tecken på det motsvarande spinnet (som ingår i fyra kopplingar). Detta överensstämmer med punkt 5 i metoden ovan.

2.5.3 Lösningsmetod för Isingmodellen

Tillståndssumman i Isingmodellen kommer att lösas med *överföringsmatrismetoden*. Tanken är att skriva om summan som elementen ur någon matrismultiplikation. För att lösa summan kan då matrisoperationer och matrisegenskaper användas. Låt oss skriva tillståndssumman enligt

$$Z = \sum_{\{s\}} e^{-\mathcal{H}(s)\cdot\beta} = \sum_{\{s\}} e^{\sum_i E(\{s\}_i, \{s\}_{i+1})}$$
$$= \sum_{\{s\}} e^{E(\{s\}_1,\{s\}_2)} \cdot e^{E(\{s\}_2,\{s\}_3)} \cdot \ldots \cdot e^{E(\{s\}_{N-1},\{s\}_N)}.$$

Här är $E(\{s\}_i, \{s\}_{i+1})$ någon funktion av en mängd spinn med index i och $i+1$ och en kopplingskonstant k_i. Definiera nu ett matriselement $T_{\{s\}_i\{s\}_{i+1}} = e^{E(\{s\}_i,\{s\}_{i+1})}$. Tillståndssumman får nu följande utseende

$$Z = \sum_{\{s\}} T_{\{s\}_1\{s\}_2} \cdot T_{\{s\}_2\{s\}_3} \cdot \ldots \cdot T_{\{s\}_{N-1}\{s\}_N}.$$

Notera att varje index endast förekommer i två intillliggande faktorer. Det räcker därför att endast undersöka dessa två faktorer när indexet summeras bort. Vi kan nu utnyttja definitionen för elementen av en matrismultiplikation, $\mathbf{C}_{i,k} = \sum_j \mathbf{A}_{i,j}\mathbf{B}_{j,k}$, för att skriva om detta som element ur en matrisprodukt

$$\sum_{\{s\}_{i+1}} T_{\{s\}_i\{s\}_{i+1}} \cdot T_{\{s\}_{i+1}\{s\}_{i+2}} = (\mathbf{T}_i \cdot \mathbf{T}_{i+1})_{\{s\}_i\{s\}_{i+2}}.$$

$\mathbf{T}_j$ är en så kallad *överföringsmatris* som endast beror av kopplingskonstanten k_j. Dessa genereras genom att beräkna $T_{\{s\}_j\{s\}_{j+1}}$ för alla möjliga kombinationer av $\{s\}_j$ och $\{s\}_{j+1}$ och sedan arrangera dessa i en matris. Indexeringen av elementen sker genom att låta +1 och -1 representera 0 respektive 1 och då översätta mängderna $\{s\}_j$ och $\{s\}_{j+1}$ till binära tal som motsvarar rad- respektive kolonnindex.

Om denna process repeteras för alla faktorer i tillståndssumman fås

$$Z = \sum_{\{s\}_1} \sum_{\{s\}_N} \left(\prod_i \mathbf{T}_i\right)_{\{s\}_1,\{s\}_N}.$$

För att lösa tillståndssumman behöver alltså matrisprodukten beräknas. Beroende på randvillkoren $\{s\}_1$ och $\{s\}_N$ kan sedan de relevanta matriselementen plockas ut ur matrisprodukten och summeras.

2.6 Tensornätverksbaserad numerisk avkodare

För att analysera ytkodens prestanda vid godtyckliga fysiska felsannolikheter krävs numeriska metoder. Ett exempel på en numerisk metod är en matrix product states (MPS) baserad avkodare, vilken approximerar en maximum-likelihood avkodare.

2.6.1 Maximum-likelihood avkodare

För att beskriva tillstånd som störs av omgivningen och där det exakta tillståndet $|\psi\rangle$ inte är känt används istället en täthetsmatris ρ [12]. Denna är definierad som $\rho = \sum_k p_k |\psi_k\rangle\langle\psi_k|$ där summan löper över alla tänkbara rena tillstånd $|\psi_k\rangle$ systemet kan vara i och p_k är sannolikheten att systemet befinner sig i det tillståndet. Om vi antar att en ursprunglig täthetsmatris ρ påverkas av en felkedja $C \in \mathcal{P}$, se avsnitt 2.3, med sannolikhet $P(C)$ kan den störda tillståndsmatrisen skrivas som

$$\rho' = \sum_{C \in \mathcal{P}} P(C) C \rho C^\dagger. \tag{2}$$

Det går att visa att det störda tillståndet, givet ett specifikt syndrom s, kan skrivas

$$\begin{aligned}\rho'_s = {} & P(C_I^s) C(s) \rho C(s) \\ & + P(C_X^s) C(s) \hat{X}_L \rho \hat{X}_L C(s) \\ & + P(C_Y^s) C(s) \hat{Y}_L \rho \hat{Y}_L C(s) \\ & + P(C_Z^s) C(s) \hat{Z}_L \rho \hat{Z}_L C(s),\end{aligned}$$

2där $C(s)$ är något $C \in \mathcal{P}$ som ger upphov till det givna syndromet s [20]. Vidare är $C_I^s = C(s)\mathcal{S}$, $C_X^s = C(s)\hat{X}_L\mathcal{S}$, $C_Y^s = C(s)\hat{Y}_L\mathcal{S}$ och $C_Z^s = C(s)\hat{Z}_L\mathcal{S}$ där $C\mathcal{S} \equiv \{Cg : g \in \mathcal{S}\}$ och $\mathcal{S}$ är stabilisatorgruppen. Observera att för $C(s) \in \mathcal{I}$ motsvarar $P(C_I^s)$, $P(C_X^s)$, $P(C_Y^s)$ och $P(C_Z^s)$ sannolikheterna $P_\mathcal{I}$, $P_\mathcal{X}$, $P_\mathcal{Y}$ och $P_\mathcal{Z}$. En maximum-likelihood avkodare beräknar $P(C_I^s)$, $P(C_X^s)$, $P(C_Y^s)$ och $P(C_Z^s)$ och väljer ett element $r(s) \in C_{ML}^s$ som appliceras för att rätta felet där C_{ML}^s är den grupp med högst sannolikhet.

2.6.2 Matrix product states avkodare

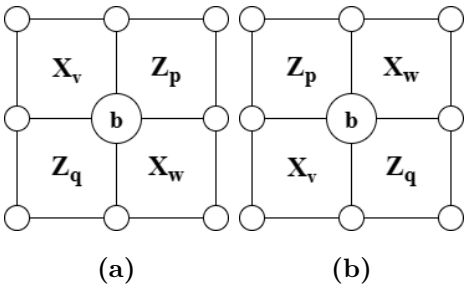
Att exakt beräkna $P(C_I^s)$, $P(C_X^s)$, $P(C_Y^s)$ och $P(C_Z^s)$ är i praktiken omöjligt för större koddistanser och godtyckliga felsannolikheter och approximationer är därför nödvändiga. Detta kan göras på flera sätt, exempelvis genom Metropolis baserad Monte Carlo-sampling [21][22], men i detta arbete används istället en matrix product states baserad avkodare [20] vars beskrivning nedan följer den given av Y. Xiao [16].

Låt C tillhöra någon av C_I^s, C_X^s, C_Y^s, C_Z^s och P_1 vara sannolikhetsfördelningen för felmodellen. Sannolikheten $P(C)$ kan då skrivas som

$$P(C) = \sum_{g \in \mathcal{S}} \prod_b P_1(C_b g_b),$$

där summan löper över alla stabilisatorer i stabilisatorgruppen och produkten över alla kvantbitar på ytkoden där en feloperator C_b har verkat och där g_b betecknar de operatorer från stabilisatorer

som verkat på kvantbiten. Betrakta exempelvis XZ-koden som visas i figur 2. Varje kvantbit, utom de på randen, gränsar till fyra stabilisatorer i två olika konfigurationer, se figur 4, där cirklar betecknar kvantbitar och rutorna stabilisatorer. Stabilisatorerna som verkar på kvantbiten b beror då på fyra variabler X_v, X_w, Z_p och Z_q som kan anta värdena 0 eller 1 beroende på om stabilisatorn har verkat (1) eller inte verkat (0) på kvantbiten. Alltså $g_b(X;Z) = g_b(X_v, X_w; Z_p, Z_q)$ eller $g_b = g_b(i,j,k,l) = X^i X^j Z^k Z^l$ där $i,j,k,l \in \{0,1\}$ för XZ-koden. Kvantbitarna på kanten kan beskrivas på samma sätt, men med två eller tre index istället för fyra.



Figur 4: De två möjliga konfigurationerna för en kvantbit b i XZ-koden. Cirklar betecknar kvantbitar och rutor stabilisatorer.

Definiera $T(X,Z) = \prod_b P_1(C_b g_b(X_v, X_w; Z_p, Z_q))$. Då hela stabilisatorgruppen genereras genom att applicera alla möjliga konfigurationer av stabilisatorer erhålls

$$P(C) = \sum_X \sum_Z T(X,Z).$$

Denna summa kan skrivas som en kontrahering, det vill säga summering, av ett lämpligt tensornätverk bestående av tre olika typer av noder kallade s, h och v. I detta tensornätverk skrivs stabilisatorerna som s-noder och h respektive v noder beskriver kvantbitar för de två olika möjliga konfigurationerna. För att tensornätverket korrekt ska beskriva ytkoden måste alla länkar, det vill säga index, in till varje s-nod vara samma från alla intilliggande h- eller v-noder, eftersom en stabilisator verkar på alla fyra intilliggande kvantbitar. Definiera en binär variabel $\gamma = \gamma(X,Z)$ som är 1 om alla länkar är korrekta och 0 annars. Med detta kan $T(X,Z)$ skrivas som en funktion av γ, vilket ger

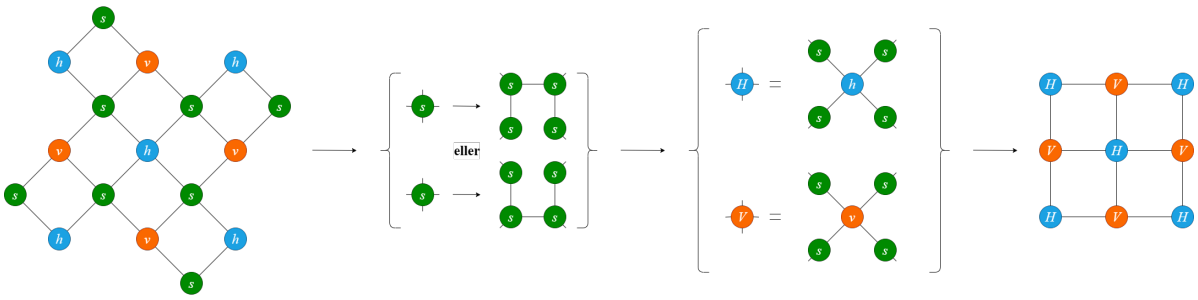
$$\sum_\gamma \prod_{h,v,s} T(\gamma).$$

Summan löper över alla γ och produkten över alla s,- h- och v-noder. Sannolikheten kan alltså skrivas som en kontrahering av ett tensornätverk med tensorelement

$$s(i,j,k,l) = \begin{cases} 1 \text{ om } i=j=k=l, \\ 0 \text{ annars}, \end{cases}$$
$$h(i,j,k,l) = P_1(C_b g_b(j,l,i,k)),$$
$$v(i,j,k,l) = P_1(C_b g_b(i,k,j,l)),$$

där skillanden i indexeringen mellan h- och v-noderna beror på de två olika geometrierna. Tensornätverket som motsvarar en XZ-kod med $d = 3$ visas till vänster i figur 5. Genom att dela upp varje s-nod i fyra stycken s-noder med endast två index var (vilka fortfarande alla blir ett om alla index in är samma och noll annars) sammanbundna i övre eller undre kanten och införa H- och V-noder enligt samma figur erhålls nätverket som visas längst till höger. Länkar ut från

en nod representerar ett av tensorns index och sammanbundna länkar mellan noder representerar en summation av motsvarande delade index. Noder på kanterna beskrivs med endast två eller tre index.



Figur 5: Tensornätverket som motsvarar XZ-koden för $d = 3$ visas till vänster. Genom att dela upp varje s-nod i fyra stycken s-noder med två index var och införa två nya typer av noder H och V erhålls tensornätverket som visas till höger. Länkar ut från en nod betecknar ett av tensorns index, och sammanbunda länkar betecknar en summering av motsvarande delade index.

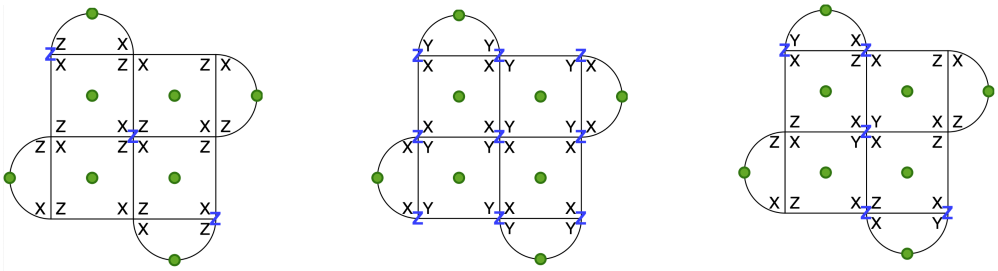
Detta nätverk kan beräknas kolumnvis, vilket emellertid är en krävande beräkning. En approximation kan göras genom att efter varje kolumn göra en Schmidt dekomponering av den erhållna tensorn och endast behålla de $\chi \geq 2$ största Schmidtkoefficienterna, se artikeln av S. Bravyi, M. Suchara och A. Vargo [20] för en ingående beskrivning av approximeringsalgoritmen. Trunkeringsparametern χ kallas för bindningsdimension.

2.6.3 Simulering

Den numeriska simuleringen härstammar från modifikationer till Python paketet *qecsim* [23][24] och dess tillägg *qsdxzzx* [25]. Vid simuleringen tas n_{run} antal felkedjor slumpmässigt från en given sannolikhetsfördelning. Därefter beräknar MPS-avkodaren den mest sannolika korrigeringen C' för varje felkedja enligt algoritmen ovan. Slutligen beräknas antalet gånger CC' kommuterar med både X_L och Z_L, antalet gånger CC' antikommuterar med Z_L och antalet gånger CC' antikommuterar med X_L; alltså antalet lyckade rättningar, antalet bit-flip fel och antalet fas-flip fel. Även antalet gånger CC' inte kommuterade med varken X_L eller Z_L beräknas. Den logiska felsannolikheten beräknas därefter som antalet misslyckade rättningar genom antalet slumpmässiga felkedjor. Simuleringarnas storlekar kräver att de utförs på datakluster tilldelade av Chalmers Centre for Computational Science and Engineering (C3SE), genom Chalmers e-Commons.

3 Analytisk lösning

Med hjälp av mappningen till Isingmodellen kommer tre specifika ytkoder att lösas: XZZX, XY och XZZXdY, som vardera visas med koddistansen $d = 3$ i figur 6a, figur 6b respektive figur 6c. Figurerna visar även de logiska Z_L-operatorerna för varje kod, bestående av individuella Z-operatorer (inritade i blått). Alla tre koder har samma logiska X_L-operator som består av X-operatorer på högerdiagonalen (se exempelvis figur 7a).



(a) XZZX **(b)** XY **(c)** XZZXdY

Figur 6: XZZX-, XY- och XZZXdY-ytkoderna med respektive logiska Z_L-operatorer bestående av individuella Z-operatorer inritade i blått.

I följande avsnitt kommer först en viktig förenkling att introduceras som är nödvändig för att Isingmodellen ska bli analytiskt lösbar. Därefter kommer tillståndssumman och således även ekvivalensklassernas felsannolikheter att lösas för de tre ytkoderna under denna förenkling. Resultaten för XZZX- och XY-koden är reproducerade från Y. Xiao, B. Srivastava och M. Granath, medan metoden för XZZXdY är ny.

3.1 Förenklingar vid en specifik felsannolikhet

Vi kommer att studera en specifik felsannolikhet, $p = p_s = \frac{1+1/\eta}{2+1/\eta}$, där $p_z = 1 - p$ och analytiska lösningar är möjliga. I denna punkt blir $k_x = k_y = 0$ enligt definitionerna av kopplingskonstanterna som visas i ekv. 1. Vid detta p har alla syndrom samma felsannolikhet, vilket innebär att vi endast kommer behöva beräkna sannolikheten för ekvivalensklasserna för *ett* syndrom. Vi ska nu visa detta.

En given ytkod med koddistans d har d^2 kvantbitar och $d^2 - 1$ stabilisatorer. Vi gör nu följande observationer

1. Det finns 2^{d^2-1} syndrom.
2. Det finns 2^{d^2} felkedjor med endast I- och Z-fel.
3. Det finns inga stabilisatorer som endast består av Z-operatorer.
4. Z_L är den enda operatorn bestående av bara I- och Z-operatorer som inte förändrar syndromet när den verkar på en felkedja.
5. Felkedjor med bara I och Z byter tecken på k_x och k_y, som båda är noll vid denna specifika felsannolikhet. Dessa felkedjor ger alltså inget teckenbyte på någon kopplingskonstant.

Punkt 3 inses genom att först direkt observera att inga individuella stabilisatorer med endast Z-operatorer återfinns i någon av de tre ytkoderna. Vi måste dock säkerställa att samma sak gäller för någon produkt av stabilisatorer. De möjliga operatorerna i stabilisatorerna som verkar på kvantbit i är X_i, Y_i och Z_i. Vidare vet vi att $X_iX_j = I$ då $i = j$ och $\neq I$ annars, och på samma sätt för alla operatorer. Vi ser direkt i ytkoderna att för alla kombinationer av produkter av stabilisatorer så finns det ingen som endast innehåller Z-operatorer.

Punkt 4 följer från punkt 3. Eftersom ingen av ytkoderna har någon ren Z-stabilisator så innebär det att X- eller Y-operatorer kommer att innefattas vid varje verkan av en stabilisator, vilket då även kommer förändra dess Z-fel; vi får alltså inte längre en ren Z-felkedja. Av samma anledning går det inte att hitta någon annan Z_L-operator bestående endast av individuella Z-operatorer.

Från dessa observationer erhåller vi följande: givet en felkedja som exklusivt består av I- och Z-fel kan vi få en, och endast en, ytterligare felkedja med samma syndrom av endast I- och Z-fel genom att verka med Z_L. Dessa två kedjor ligger uppenbart i ekvivalensklasserna $\mathcal{I}$ och $\mathcal{Z}$, eftersom $Z_L(C_0 \in \mathcal{I}) = C_{Z_L} \in \mathcal{Z}$ och $Z_L(C_{Z_L} \in \mathcal{Z}) = Z_LZ_L(C_0 \in \mathcal{I}) = C_0 \in \mathcal{I}$. Alla felkedjor med I- och Z-fel utgör $2^{d^2}/2$ antal syndrom, det vill säga det totala antalet syndrom. Alla syndrom kan således representeras av två olika felkedjor bestående av endast I- och Z-fel. Eftersom vi vet att dessa felkedjor inte kräver att kopplingskonstanterna byter tecken kan vi för varje syndrom i ekvivalensklasserna $\mathcal{I}$ och $\mathcal{Z}$ hitta en ursprungskedja utan några teckenbyten. När tillståndssumman kommer att beräknas för $\mathcal{I}$ och $\mathcal{Z}$ kan vi därför alltså utgå från samma Hamiltonian för alla syndrom. Detta innebär att tillståndssumman blir densamma för alla syndrom vilket medför att även sannolikheten för ekvivalensklasserna $\mathcal{I}$ och $\mathcal{Z}$ blir samma för alla syndrom.

Från detta kan vi även dra slutsatser om andra ekvivalensklasser och hur deras sannolikheter relaterar till varandra. Eftersom vi nu vet att alla syndrom kan representeras av en felkedja tillhörande ekvivalensklass $\mathcal{I}$, utan att kräva några teckenbyten, kan vi även erhålla en annan felkedja tillhörande en annan ekvivalensklass genom att agera med motsvarande logiska operator. Antag att vi har en felkedja $C_0 \in \mathcal{I}$ som inte har några teckenbyten, vilket vi nu vet går att hitta för alla syndrom. För resterande ekvivalensklasser fås då följande:

- $\mathcal{Z}$: $Z_LC_0 \in \mathcal{Z}$. Z_L består bara av Z-operatorer, därför sker inga teckenbyten. Felkedjornas Hamiltonianer är således identiska och vi får $P_{\mathcal{Z}} = P_{\mathcal{I}}$.

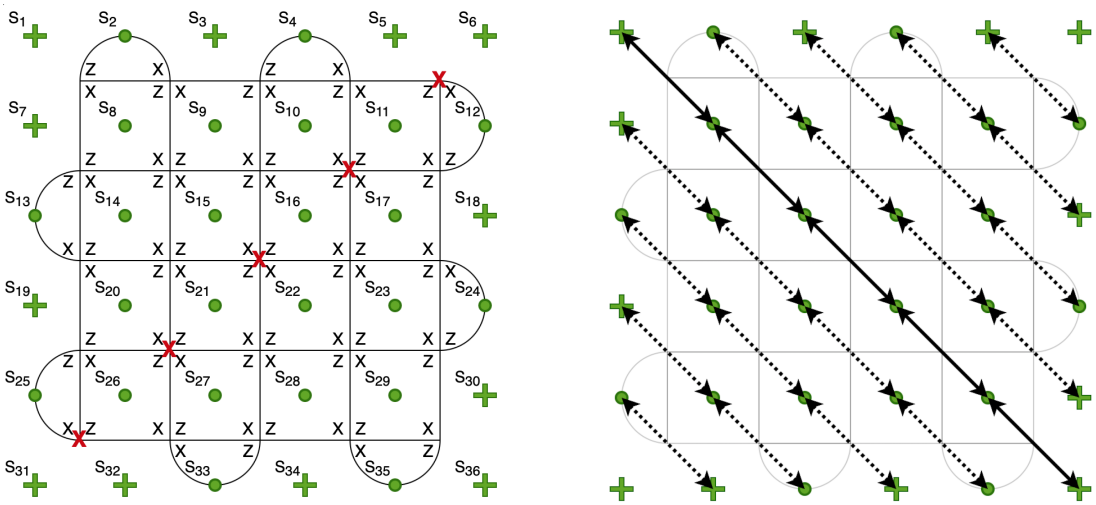
- $\mathcal{X}$: $X_LC_0 \in \mathcal{X}$. X_L består av d stycken X-operatorer, därför sker d teckenbyten. Detta gäller för alla syndrom och därför är $P_{\mathcal{X}}$ samma för alla syndrom.

- $\mathcal{Y}$: $Y_LC_0 = Z_LX_LC_0 \in \mathcal{Y}$. X_L innebär d stycken teckenbyten medan Z_L inte innebär några teckenbyten. Vi får alltså lika många teckenbyten som för $\mathcal{X}$ vilket ger $P_{\mathcal{X}} = P_{\mathcal{Y}}$.

Således räcker det alltså att bara räkna på ekvivalensklasserna $\mathcal{I}$ och $\mathcal{X}$. Vidare inses även att eftersom felsannolikheten är densamma för alla syndrom räcker det att räkna ut felsannolikheten för ett syndrom. För lösningarna av de tre ytkoderna kommer syndromet bestående av att alla stabilisatorer mäter $+1$ att användas. För $\mathcal{I}$ väljer vi en ursprungskedja utan fel. För $\mathcal{X}$ agerar vi bara med X_L på kedjan.

3.2 XZZX

XZZX-koden visas, med X_L-operatorn längs högerdiagonalen, i figur 7a och dess kopplingar ses bredvid i figur 7b. Alla kopplingskedjor som är fästa till ett fixerat spinn i *en* ände är invarianta under teckenbyten i kopplingarna. Detta framgår från att vi kan definiera om det ena spinnet i kopplingarna samt alla efterkommande spinn, och då erhålla samma kedja som innan teckenbytet. Eftersom vi sedan summerar över alla möjliga spinntillstånd så påverkar inte omdefinieringen resultatet. Tillståndssumman med och utan fel kommer därför att få en identisk faktor som tar ut sig själv när den relativa felsannolikheten beräknas. När kedjan är fäst i båda ändar går det inte att omdefiniera det sista spinnet i kedjan då detta är fixerat som +1, och kedjan blir således *inte* invariant under ett teckenbyte. För våra syften räcker det därför att enbart undersöka kedjan som är fäst till ett fixerat spinn i båda ändar. I figur 7b är den relevanta kedjan markerad med heldragna kopplingar, medan icke-relevanta är markerade med streckade kopplingar. Notera att när vi verkar med X_L på vår ursprungskedja så byts tecknet på *en* koppling i den relevanta kedjan.

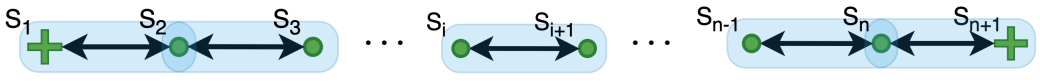


(a) XZZX med X_L-operatorn längs högerdiagonalen.

(b) Kopplingarna mellan dess stabilisatorer.

Figur 7: En d = 5 XZZX-ytkod med X_L-operatorn längs högerdiagonalen samt kopplingarna mellan dess stabilisatorer, där heldragna respektive streckade pilar indikerar relevanta respektive icke-relevanta kopplingar.

För en allmän kodstorlek får vi alltså en kedja med $n + 1$ antal spinn och n antal kopplingar ($n = d$ för XZZX-koden) samt randvillkoret $s_1 = s_{n+1} = +1$. Låt kopplingskonstanten i varje koppling vara $k_i = \pm k_Z$ och döp om spinnen i kedjan som i figur 8. Hamiltonianen för systemet ges av $\mathcal{H}\beta = k_1 s_1 s_2 + k_2 s_2 s_3 + ... + k_n s_n s_{n+1}$ och tillståndssumman av $Z = \sum_{\{s\}} \exp\left[-\mathcal{H}\beta\right]$. För att lösa tillståndssumman kommer överföringsmatrismetoden att användas, och därför definierar vi en enhetscell som vid repetition återskapar hela kedjan. Enhetscellerna visas i blått i figur 8.



Figur 8: Kopplingskedja med fixerade ändar för en XZZX-kod av allmän kodstorlek där enhetcellerna är representerade i blått.

Varje enhetscell kan skrivas som ett matriselement med två index:

$$T_{s_i\ s_{i+1}} = \exp\left[-k_i(s_i s_{i+1})\right].$$

Genom att låta spinn +1 motsvaras av 0 och spinn -1 motsvaras av 1 kan spinnen i indexet översättas som ett binärt tal till ett decimaltal. Det första spinnet motsvarar då matrisens radindex och det andra matrisens kolonnindex. Eftersom rad- och kolonnindexen löper från 0 till 1 fås 4 element, vilket ger en 2x2-matris. Genom att beräkna alla möjliga matriselement och indexera dem på detta vis erhålls överföringsmatrisen

$$\mathbf{T}_i = \begin{bmatrix} T_{++} & T_{+-} \\ T_{-+} & T_{--} \end{bmatrix} = \begin{bmatrix} T_{00} & T_{01} \\ T_{10} & T_{11} \end{bmatrix} = \begin{bmatrix} x_i & x_i^{-1} \\ x_i^{-1} & x_i \end{bmatrix},$$

där $x_i = \exp(-k_i)$. Överföringsmatrisens egenvärden och egenvektorer presenteras i tabell 1. Dessa kommer användas längre fram i lösningsgången.

Tabell 1: Egenvärdena för XZZX-kodens överföringsmatris och deras korresponderande egenvektorer.

Egenvärden	Egenvektorer
$2\cosh(-k_i)$	[1 1]
$2\sinh(-k_i)$	[1 -1]

Tillståndssumman kan nu beräknas enligt

$$Z = \sum_{\{s\}} T_{s_1\ s_2} \cdot T_{s_2\ s_3} \cdot \ldots \cdot T_{s_n\ s_{n+1}} = \sum_{\{s\}} \Big(\prod_i \mathbf{T}_i\Big)_{s_1\ s_{n+1}} = \Big(\prod_i \mathbf{T}_i\Big)_{++}.$$

Notera att tillståndssumman för det periodiska randvillkoret Z_{per} (det vill säga $s_1 = s_{n+1}$), är

$$\sum_{\{s\}} \Big(\prod_i \mathbf{T}_i\Big)_{s_1\ s_1} = \mathrm{Tr}\Big[\Big(\prod_i \mathbf{T}_i\Big)\Big] = \Big(\prod_i \mathbf{T}_i\Big)_{++} + \Big(\prod_i \mathbf{T}_i\Big)_{--}.$$

Eftersom ett teckenbyte på samtliga spinn inte förändrar Hamiltonianen har vi att $\left(\prod_i \mathbf{T}_i\right)_{++} = \left(\prod_i \mathbf{T}_i\right)_{--}$. Alltså har vi att $Z = \frac{1}{2} Z_{\text{per}}$, vilket vi vill utnyttja då Z_{per} är enklare att beräkna än Z.

För att beräkna Z_{per} diagonaliserar vi $\mathbf{T}$. Alltså $\mathbf{T} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1}$, där $\mathbf{P}$ är en matris där kolonnerna är egenvektorerna till $\mathbf{T}$ och $\mathbf{D}$ är en diagonal matris med egenvärdena till $\mathbf{T}$ på diagonalen. Eftersom egenvektorerna är oberoende av k_i så blir även $\mathbf{P}$ oberoende, vilket ger att

$$\begin{aligned} Z_{\text{per}} &= \mathrm{Tr}(\mathbf{P}\mathbf{D}_1\mathbf{P}^{-1}\mathbf{P}\mathbf{D}_2\mathbf{P}^{-1} \cdot \ldots \cdot \mathbf{P}\mathbf{D}_n\mathbf{P}^{-1}) = \mathrm{Tr}\Big[\mathbf{P}\Big(\prod_i \mathbf{D}_i\Big)\mathbf{P}^{-1}\Big] = \mathrm{Tr}\Big[\mathbf{P}\mathbf{P}^{-1}\Big(\prod_i \mathbf{D}_i\Big)\Big] \\ &= \mathrm{Tr}\Big[\prod_i \mathbf{D}_i\Big] = \Big(\prod_i 2\cosh(-k_i)\Big) + \Big(\prod_i 2\sinh(-k_i)\Big). \end{aligned}$$

Utan fel har vi att $k_i = k\ \forall i$ och för X-fel har vi att $k_{n/2} = -k$, medan resterande $k_i = k$. Detta

ger tillståndssumman utan fel ($Z_{\text{utan fel}}$) och tillståndssumman med X-fel ($Z_{\text{med X-fel}}$) som

$$Z_{\text{utan fel}} = \frac{1}{2}[2^n \cosh^n(-k) + 2^n \sinh^n(-k)],$$
$$Z_{\text{med X-fel}} = \frac{1}{2}[2^n \cosh^n(-k) + 2^n \sinh^{n-1}(-k)\sinh(k)] = \frac{1}{2}[2^n \cosh^n(-k) - 2^n \sinh^n(-k)].$$

Slutligen erhålls den relativa felsannolikheten

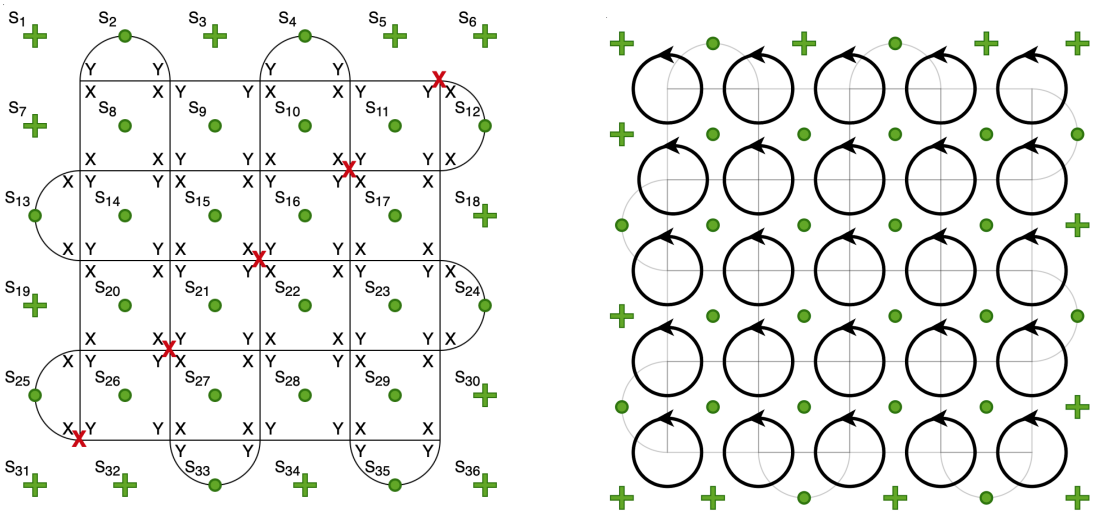
$$\frac{P_{\mathcal{X}}}{P_{\mathcal{I}}} = \frac{Z_{\text{med X-fel}}}{Z_{\text{utan fel}}} = \frac{1 - \tanh^n(-k)}{1 + \tanh^n(-k)}.$$

Med $k = -\frac{1}{2}\ln(2\eta)$ och $n = d$ fås, efter några algebraiska omskrivningar (se appendix A), ekv. 3.

$$\frac{P_{\mathcal{X}}}{P_{\mathcal{I}}} = \tanh[d \cdot \operatorname{artanh}(1/(2\eta))]. \tag{3}$$

3.3 XY

XY-koden visas, med X_L-operatorn längs högerdiagonalen, i figur 9a och dess kopplingar visas i figur 9b.

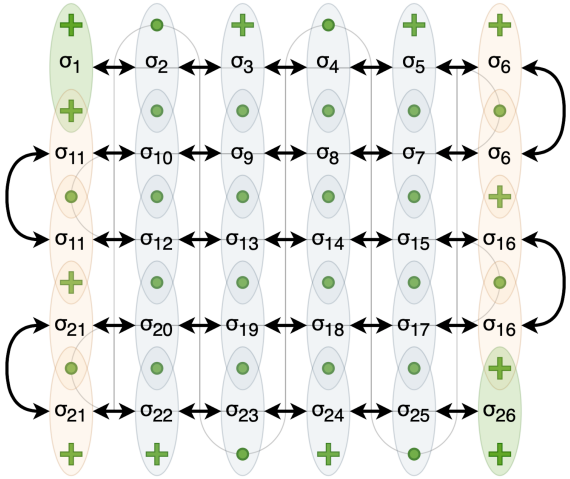


(a) XY med X_L-operatorn längs högerdiagonalen.

(b) Kopplingarna mellan dess stabilisatorer.

Figur 9: En d = 5 XY-ytkod med X_L längs högerdiagonalen samt kopplingarna mellan dess stabilisatorer.

XY-koden har endast kopplingar som inkluderar fyra spinn, det vill säga $J_i s_i s_{i+1} s_{i+d+1} s_{i+d+2}$. För att förenkla problemet utför vi därför variabelbytet $s_i s_{i+d+1} \to \sigma_j$ som visas i figur 10. Notera att spinnen i översta vänstra hörnet och nedersta högra hörnet (färgade gröna) blir +1, och vartannat spinn på randen blir indentiska (färgade gula). Resultatet blir således en en-dimensionell kedja även i detta fall, som visas med svarta pilar i figuren. Kedjan består av $n = d^2$ kopplingar och randvillkoret är $\sigma_1 = \sigma_{n+1} = +1$.



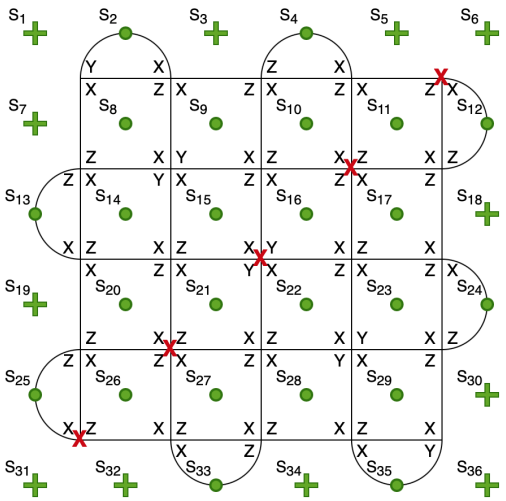
Figur 10: XY-ytkoden efter $s_i s_{i+d+1} \to \sigma_j$ substitutionen med inritad kopplingskedja.

Därefter löses kedjan på samma sätt som för den för XZZX-koden; enda skillnaden är att kedjan nu blir längre än vad den var i fallet för XZZX-koden. För XY-koden fås istället $n = d^2$, vilket resulterar i ekv. 4.

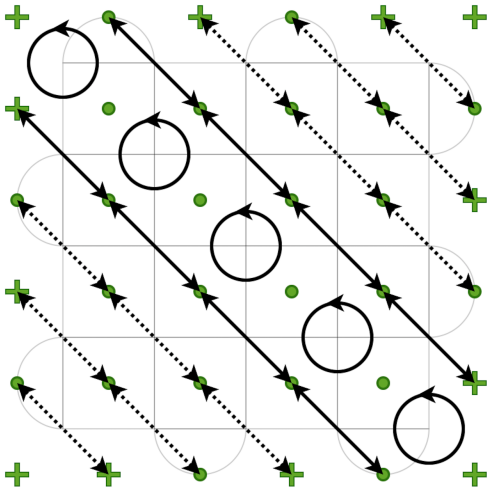
$$\frac{P_{\mathcal{X}}}{P_{\mathcal{I}}} = \tanh\left[d^2 \cdot \operatorname{artanh}(1/(2\eta))\right]. \tag{4}$$

3.4 XZZXdY

XZZXdY-koden är en XZZX-kod med Hadamard-YZ operatorer som verkar på vänsterdiagonalen. I figur 11a visas koden med X_L-operatorn längs högerdiagonalen och i 11b visas dess kopplingar. På samma sätt som för XZZX-koden så kommer kedjorna vars ena ände endast fäster till ett fixerat spinn inte att påverka resultatet. I figur 11b visas de relevanta kopplingarna som heldragna och de icke-relevanta streckade. Notera att X_L endast ger ett relevant teckenbyte på den mittersta 4-kopplingen.



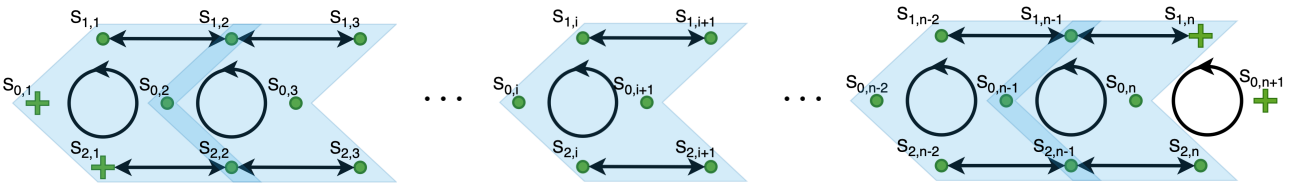
(a) XZZXdY med X_L-operatorn på högerdiagonalen.



(b) Kopplingarna mellan dess stabilisatorer.

Figur 11: En d = 5 XZZXdY-ytkod med X_L-operatorn längs högerdiagonalen samt kopplingarna mellan dess spinn, där heldragna respektive streckade pilar indikerar relevanta respektive icke-relevanta kopplingar.

För en allmän kodstorlek har vi en kedja enligt figur 12, där vi för enkelhets skull har döpt om spinnen. Randvillkoret är $s_{0,1} = s_{2,1} = s_{1,n} = s_{0,n+1} = +1$. Kopplingskonstanterna för 2-kopplingarna genomgår aldrig ett teckenbyte och kallas därför för $k = k_Z$, medan kopplingskonstanterna för 4-kopplingarna kallas för $\kappa_i = \pm k_Z$. Hamiltonianen för kedjan ges som $\mathcal{H}\beta = (ks_{1,1}s_{1,2} + ks_{2,1}s_{2,2} + \kappa_1 s_{0,1}s_{1,1}s_{2,1}s_{0,2}) + (ks_{1,2}s_{1,3} + ks_{2,2}s_{2,3} + \kappa_2 s_{0,2}s_{1,2}s_{2,2}s_{0,3}) + ... + (ks_{1,n-1}s_{1,n} + ks_{2,n-1}s_{2,n} + \kappa_{n-1}s_{0,n-1}s_{1,n-1}s_{2,n-1}s_{0,n}) + (\kappa_n s_{0,n}s_{1,n}s_{2,n}s_{0,n+1})$. För att lösa ytkoden definieras en enhetscell som genom repetition återskapar kedjan. Enhetscellerna visas i blått i figur 12. Notera att det finns $n = d-1$ enhetsceller och att den sista 4-kopplingen och sista spinnet inte inkluderas i någon enhetscell.



Figur 12: Kopplingskedja för en XZZXdY-kod av allmän kodstorlek där enhetcellerna är representerade i blått.

Varje enhetscell kan skrivas som ett matriselement med 6 index.

$$T_{\substack{s_{0,i}\ s_{0,i+1} \\ s_{1,i}\ s_{1,i+1} \\ s_{2,i}\ s_{2,i+1}}} = \exp\left[-k(s_{1,i}s_{1,i+1} + s_{2,i}s_{2,i+1}) - \kappa_i(s_{0,i}s_{1,i}s_{2,i}s_{0,i+1})\right].$$

Genom att låta spinn +1 motsvaras av 0 och spinn -1 motsvaras av 1 kan de två kolonnerna i indexet översättas som ett binärt tal till ett decimaltal. Den första kolonnen motsvarar då

matrisens radindex och den andra kolonnen motsvarar matrisens kolonnindex. Eftersom rad- och kolonnindex löper från 0 till 7 fås 64 element, vilket ger en 8x8 matris. Genom att beräkna alla möjliga matriselement och indexera dem på detta sätt fås överföringsmatrisen

$$\mathbf{T}_i = \begin{bmatrix} x^2y_i & x^2y_i^{-1} & y_i & y_i^{-1} & y_i & y_i^{-1} & x^{-2}y_i & x^{-2}y_i^{-1} \\ x^2y_i^{-1} & x^2y_i & y_i^{-1} & y_i & y_i^{-1} & y_i & x^{-2}y_i^{-1} & x^{-2}y_i \\ y_i^{-1} & y_i & x^2y_i^{-1} & x^2y_i & x^{-2}y_i^{-1} & x^{-2}y_i & y_i^{-1} & y_i \\ y_i & y_i^{-1} & x^2y_i & x^2y_i^{-1} & x^{-2}y_i & x^{-2}y_i^{-1} & y_i & y_i^{-1} \\ y_i^{-1} & y_i & x^{-2}y_i^{-1} & x^{-2}y_i & x^2y_i^{-1} & x^2y_i & y_i^{-1} & y_i \\ y_i & y_i^{-1} & x^{-2}y_i & x^{-2}y_i^{-1} & x^2y_i & x^2y_i^{-1} & y_i & y_i^{-1} \\ x^{-2}y_i & x^{-2}y_i^{-1} & y_i & y_i^{-1} & y_i & y_i^{-1} & x^2y_i & x^2y_i^{-1} \\ x^{-2}y_i^{-1} & x^{-2}y_i & y_i^{-1} & y_i & y_i^{-1} & y_i & x^2y_i^{-1} & x^2y_i \end{bmatrix},$$

där $x = e^{-k}$ och $y_i = e^{-\kappa_i}$. Överföringsmatrisens egenvärden och egenvektorer presenteras i tabell 2. Dessa kommer användas längre fram i lösningsgången.

Tabell 2: Egenvärdena för XZZXdY-kodens överföringsmatris och deras korresponderande egenvektorer.

Egenvärden	Egenvektorer							
$-4\sinh(-2k)\sinh(-\kappa_i)$	[-1	1	$2e^{-2k}$	0	$-2e^{-2k}$	0	-1	1]
$-4\sinh(-2k)\sinh(-\kappa_i)$	[0	0	1	-1	-1	1	0	0]
$4\sinh(-2k)\sinh(-\kappa_i)$	[1	-1	0	0	0	0	-1	1]
$4\sinh(-2k)\sinh(-\kappa_i)$	[$2e^{-2k}$	$-2e^{-2k}$	-1	-1	1	1	0	0]
$8\sinh^2(-k)\cosh(-\kappa_i)$	[1	1	-1	-1	-1	-1	1	1]
$8\cosh^2(-k)\cosh(-\kappa_i)$	[1	1	1	1	1	1	1	1]
$4\sinh(-2k)\cosh(-\kappa_i)$	[-1	-1	0	0	0	0	1	1]
$4\sinh(-2k)\cosh(-\kappa_i)$	[0	0	-1	1	-1	1	0	0]

Tillståndssumman blir

$$\begin{aligned} Z &= \sum_{\{s\}} T_{\substack{s_{0,1}\ s_{0,2} \\ s_{1,1}\ s_{1,2} \\ s_{2,1}\ s_{2,2}}} \cdot T_{\substack{s_{0,2}\ s_{0,3} \\ s_{1,2}\ s_{1,3} \\ s_{2,2}\ s_{2,3}}} \cdot \ldots \cdot T_{\substack{s_{0,n-1}\ s_{0,n} \\ s_{1,n-1}\ s_{1,n} \\ s_{2,n-1}\ s_{2,n}}} \exp\left[-\kappa_n(s_{0,n}s_{1,n}s_{2,n}s_{0,n+1})\right] \\ &= \sum_{\{s\}} \Big(\prod_i \mathbf{T}_i\Big)_{\substack{s_{0,1}\ s_{0,n} \\ s_{1,1}\ s_{1,n} \\ s_{2,1}\ s_{2,n}}} \exp\left[-\kappa_n(s_{0,n}s_{1,n}s_{2,n}s_{0,n+1})\right] = \sum_{\{s\}} \Big(\prod_i \mathbf{T}_i\Big)_{\substack{+\ s_{0,n} \\ s_{1,1}\ + \\ +\ s_{2,n}}} \cdot \exp\left[-\kappa_n(s_{0,n}+s_{2,n}+)\right] \\ &= \Big(\prod_i \mathbf{T}_i\Big)_{\substack{++ \\ ++ \\ ++}} \exp[-\kappa_n] + \Big(\prod_i \mathbf{T}_i\Big)_{\substack{+- \\ ++ \\ ++}} \exp[\kappa_n] + \Big(\prod_i \mathbf{T}_i\Big)_{\substack{++ \\ -+ \\ ++}} \exp[-\kappa_n] + \Big(\prod_i \mathbf{T}_i\Big)_{\substack{++ \\ ++ \\ +-}} \exp[\kappa_n] \\ &+ \Big(\prod_i \mathbf{T}_i\Big)_{\substack{+- \\ -+ \\ ++}} \exp[\kappa_n] + \Big(\prod_i \mathbf{T}_i\Big)_{\substack{+- \\ ++ \\ +-}} \exp[-\kappa_n] + \Big(\prod_i \mathbf{T}_i\Big)_{\substack{++ \\ -+ \\ +-}} \exp[\kappa_n] + \Big(\prod_i \mathbf{T}_i\Big)_{\substack{+- \\ -+ \\ +-}} \exp[-\kappa_n]. \end{aligned}$$

För att förenkla beräkningarna diagonaliseras $\mathbf{T}_i$. Alltså $\mathbf{T}_i = \mathbf{P}\mathbf{D}_i\mathbf{P}^{-1}$, där $\mathbf{P}$ är en matris med egenvektorerna till $\mathbf{T}_i$ som kolonner och $\mathbf{D}_i$ är en diagonal matris med egenvärdena på diagonalen. $\mathbf{T}_i$-matrisen har egenvektorer som är oberoende av κ_i, och är därför blir $\mathbf{P}$-matrisen samma för alla $\mathbf{T}_i$. Detta ger

$$\prod_i \mathbf{T}_i = \mathbf{P}\mathbf{D}_1\mathbf{P}^{-1}\mathbf{P}\mathbf{D}_2\mathbf{P}^{-1}\mathbf{P}\dots\mathbf{P}\mathbf{D}_n\mathbf{P}^{-1} = \mathbf{P}\Big(\prod_i \mathbf{D}_i\Big)\mathbf{P}^{-1}.$$

Låt $\mathbf{D}$ vara den diagonala matrisen då $\kappa_i = k$, och låt $\mathbf{D'}$ vara den diagonala matrisen då $\kappa_i = -k$. Utan X-fel har vi att alla $\kappa_i = k$, och därför även att $\prod_i \mathbf{T}_i = \mathbf{P}(\mathbf{D}^n)\mathbf{P}^{-1}$; med X-fel har vi att $\kappa_{n/2+1} = -k$, medan resterande $\kappa_i = k$. Detta ger $\prod_i \mathbf{T}_i = \mathbf{P}(\mathbf{D}^{n/2}\mathbf{D'}\mathbf{D}^{n/2-1})\mathbf{P}^{-1} = \mathbf{P}(\mathbf{D}^{n-1}\mathbf{D'})\mathbf{P}^{-1}$. Genom att utföra dessa beräkningar med egenvärdena och egenvektorerna ovan erhåller vi tillståndssumman utan, respektive med, fel som

$$Z_{\text{utan fel}} = 8^n \cosh^n(-k)[\cosh^{2n+1}(-k) + \sinh^{2n+1}(-k)],$$
$$Z_{\text{med X-fel}} = 8^n \cosh^n(-k)[\cosh^{2n+1}(-k) - \sinh^{2n+1}(-k)].$$

Dessa beräkningar genomfördes med hjälp av Wolfram Mathematica [26]. Slutligen fås den relativa sannolikheten för X-fel som

$$\frac{P_{\mathcal{X}}}{P_{\mathcal{I}}} = \frac{Z_{\text{med X-fel}}}{Z_{\text{utan fel}}} = \frac{1 - \tanh^{2n+1}(-k)}{1 + \tanh^{2n+1}(-k)}.$$

Med samma omskrivningar som för XZZX-koden och $n = d - 1$ fås ekv. 5.

$$\frac{P_{\mathcal{X}}}{P_{\mathcal{I}}} = \tanh\left[(2d-1)\cdot \operatorname{artanh}(1/(2\eta))\right]. \tag{5}$$

3.5 Logiska felsannolikheter

Utifrån den relativa sannolikheten $P_{\mathcal{X}}/P_{\mathcal{I}}$ beräknas den totala logiska felsannolikheten $P_f = P_{\mathcal{X}} + P_{\mathcal{Y}} + P_{\mathcal{Z}}$, sannolikheten för en icke-exklusiv logisk bit-flip $P_{fZ} = P_{\mathcal{X}} + P_{\mathcal{Y}}$ samt fas-flip $P_{fX} = P_{\mathcal{Z}} + P_{\mathcal{Y}}$ (definierade genom huruvida korrigeringen CC' antikommuterar med Z_L respektive X_L). Vi definierar även en tredje sannolikhet för enbart bit- eller fas-flip, $P_{fY} = P_{\mathcal{X}} + P_{\mathcal{Z}}$. Den totala logiska felsannolikheten kan då uttryckas i termer av dessa sannolikheter enligt $P_f = \frac{1}{2}(P_{fX} + P_{fY} + P_{fZ})$. Genom att definiera en Z-koddistans d_Z med $d_Z = d$ för XZZX-koden, $d_Z = d^2$ för XY-koden och $d_Z = 2d - 1$ för XZZXdY-koden kan den relativa sannolikheten $P_{\mathcal{X}}/P_{\mathcal{I}}$ för de tre koderna skrivas enligt ekv. 6

$$\frac{P_{\mathcal{X}}}{P_{\mathcal{I}}} = \tanh[d_Z \cdot \operatorname{artanh}(1/(2\eta))]. \tag{6}$$

Notera att $1 = P_{\mathcal{I}} + P_{\mathcal{X}} + P_{\mathcal{Y}} + P_{\mathcal{Z}}$, vilket tillåter omskrivningen

$$P_{fZ} = P_{\mathcal{X}} + P_{\mathcal{Y}} = \frac{P_{\mathcal{X}} + P_{\mathcal{Y}}}{P_{\mathcal{I}} + P_{\mathcal{X}} + P_{\mathcal{Y}} + P_{\mathcal{Z}}}.$$

Eftersom $P_{\mathcal{X}} = P_{\mathcal{Y}}$ och $P_{\mathcal{I}} = P_{\mathcal{Z}}$ för alla tre ytkoderna vid $p = p_s$ får vi efter omskrivning, se appendix B,

$$P_{fZ} = \frac{2P_{\mathcal{X}}}{2P_{\mathcal{X}} + 2P_{\mathcal{I}}} = \frac{1}{1 + (P_{\mathcal{X}}/P_{\mathcal{I}})^{-1}} = \frac{1}{2} - \frac{1}{2}e^{-2d_Z \operatorname{artanh}(1/(2\eta))}. \tag{7}$$

Notera även att $1 = P_{\mathcal{I}} + P_{\mathcal{X}} + P_{\mathcal{Y}} + P_{\mathcal{Z}} = 2(P_{\mathcal{Z}} + P_{\mathcal{Y}})$, vilket då ger att $P_{fX} = P_{fY} = \frac{1}{2}$. Utifrån detta kan vi slutligen erhålla den totala logiska felsannolikheten som ekv. 8

$$P_f = \frac{3}{4} - \frac{1}{4}e^{-2d_Z \operatorname{artanh}(1/(2\eta))}. \tag{8}$$

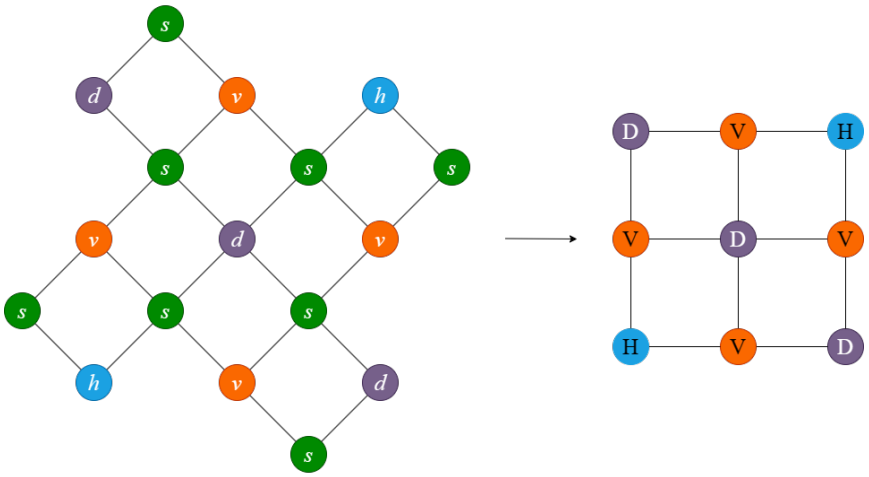
4 Numerisk lösning

I detta avsnitt beskrivs implementeringsmetoderna som användes för numerisk simulering av ytkoderna. Först presenteras den numeriska implementationen av XZZXdY-koden för att därefter presentera XY-koden.

4.1 Implementering av XZZXdY avkodare

Implementeringen av avkodaren för XZZX-koden med H_{YZ} applicerad längs vänsterdiagonalen (se figur 11a), gjordes genom att implementera tre klasser `RotatedPlanarXZZXdYCode`, `RotatedPlanarXZZXdYPauli` och `RotatedPlanarXZZXdYRMPSDecoder` baserade på paketet *qecsim* [23] med tillägget *qsdxzzx* [25]. *Qecsim* är ett paket för att simulera kvantfelskorrigerande ytkoder med flera vanliga typer av ytkoder och felmodeller implementerat. *Qsdxzzx* är ett tillägg som även implementerar XZZX-koden. På grund av likheten mellan XZZX- och XZZXdY-koderna kunde stora delar av de färdigimplementerade klasserna `RotatedPlanarXZCode` och `RotatedPlanarXZPauli` användas utan större modifikationer. Vid implementering av `RotatedPlanarXZZXdYPauli` krävdes däremot modifikationer av stabilisatorerna, där stabilisatorer med Y i nedre vänstra hörnet applicerades längs superdiagonalen och stabilisatorplaketter med Y i övre högra hörnet längs subdiagonalen. Den logiska X operatorn X_L, i `RotatedPlanarXZPauli` definierad som alternerande X och Z operatorer applicerade längs nedre kanten, krävde ingen modifikation då X operatorn applicerad på kvantbiten i nedre högra hörnet inte påverkas av H_{YZ}. Denna logiska operator är ekvivalent med valet av X_L (given i figur 11a) då de två operatorerna kan erhållas från varandra genom multiplikation med stabilisatorer. Vi har valt att beräkna den logiska Z operatorn Z_L med hjälp av den logiska Y operatorn Y_L, då Y_L är definierad mer kodeffektivt än Z_L och lär öka beräkningshastigheten. Y_L består av alternerande Z och X operatorer applicerade längs högra kanten, uppifrån och ner, där operatorn i lägre högra hörnet byts till en Y operator istället för en Z operator. Från de två definierade logiska operatorerna X_L och Y_L erhålls nu Z_L genom multiplikation av X_L och Y_L.

Störst modifikationer krävdes vid implementering av klassen `RotatedPlanarXZZXdYRMPSDecoder` då det i XZZXdY-koden finns en tredje typ av stabilisatoroperatorer runt en kvantbit. Denna konfiguration, med Y snett uppåt höger och snett nedåt vänster och X snett uppåt vänster och snett nedåt höger, se figur 11a, implementeras i form av en fjärde typ av nod i tensornätverket. Noden, $d(i,j,k,l) = P_1(C_b g_d(j,l,i,k))$ där $g_d(i,j,k,l) = X^i X^j Y^k Y^l$, ersätter h-noderna längs diagonalen från övre vänstra hörnet till nedre högra hörnet i tensornätverket, se figur 13. Samma transformation som i figur 5 genomförs med uppdelning av s-noder samt införande av H- och V-noder. D-noden är definierad analogt med H- och V-noderna. För samtliga ändringar, jämför de ursprungliga koderna [25][23] och den nyskapade koden [27].



Figur 13: Tensornätverket som implementeras i `RotatedPlanarXZZXdYRMPSDecoder`. Transformationen sker på samma sätt som i figur 5 med uppdelning av s-noderna samt införande av H-, V- och D-noder där D-noderna är definierade analogt med H- och V-noderna.

4.2 Implementering av XY avkodare

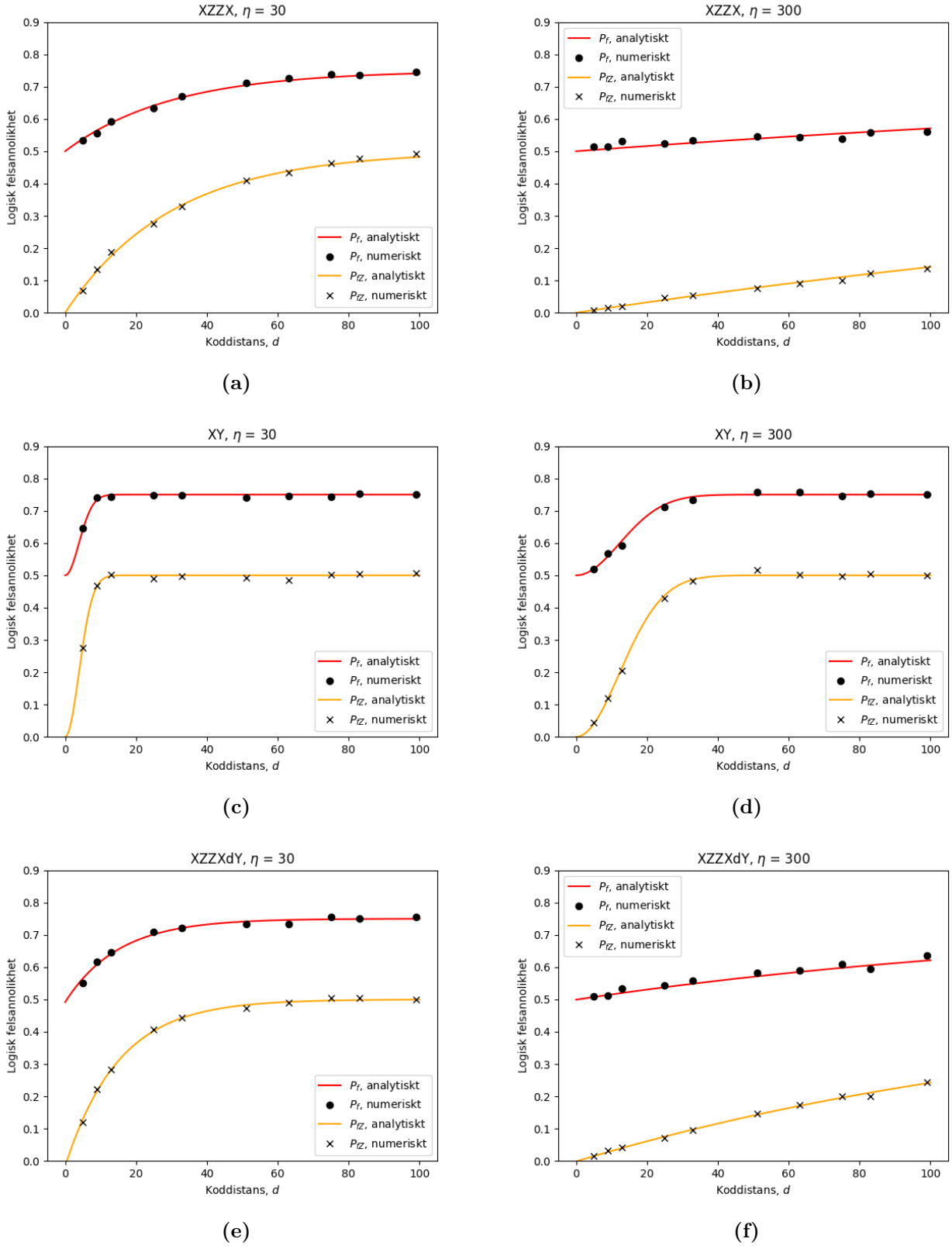
Implementeringen av XY-koden gjordes genom modifikationer till en redan existerande XZ-kod i *qecsim*-paketet [23]. Likt implementeringen av XZZXdY konfigurerades tre klasser: `RotatedPlanarXYCode`, `RotatedPlanarXYPauli` och `RotatedPlanarXYRMPSDecoder`, där var och en ärver klassmetoder från *qecsim*. Vid tillämpningen av `RotatedPlanarXYPauli` från den existerande XZ-modellen behövde endast mindre anpassningar göras till programmets tolkningar av stabilisatorer och Paulimatriser. I `RotatedPlanarXYRMPSDecoder` var det nödvändigt att ändra hur modellen identifierar Y-stabilisatorer, samt hur den genererar så kallad *sample recovery*. Tensornätverket är det som visas i figur 5.

5 Simuleringsresultat

I detta avsnitt presenteras resultaten från de numeriska simuleringarna av XY-, XZZX- och XZZXdY-koden, först i den speciella punkten där en analytisk lösning är möjlig och därefter för fysiska felsannolikheter mindre än den speciella punkten.

5.1 Jämförelse mellan numeriska och analytiska resultat i $p = p_s$

I figur 14 visas den totala logiska felsannolikheten P_f samt sannolikheten för bit-flip fel P_{fZ} i $p = p_s$ för olika koddistanser från $d = 5$ till $d = 99$ för XZZX-koden $\Big($(a) och (b)$\Big)$, XY-koden $\Big($(c) och (d)$\Big)$ samt XZZXdY-koden $\Big($(e) och (f)$\Big)$ vid två olika fasviktning, $\eta = 30$ och $\eta = 300$. De numeriska och analytiska lösningarna observeras stämma väl överens med endast en liten diskrepans vilken beror på det ändliga antalet iterationer som genomfördes i varje punkt samt valet av χ. I figuren användes vid de numeriska simuleringarna $\chi = 10$ och 3000 körningar vid varje koddistans. Överensstämmelsen mellan simulerat och analytiskt resultat troliggör att båda lösningsmetoderna fungerar väl för att analysera ytkoderna i den speciella punkten. Vidare kan det noteras att både den totala logiska felsannolikheten samt sannolikheten för bit-flip fel ökar snabbare med koddistansen för XY-koden och XZZXdY-koden än för XZZX-koden.



(a) **(b)** **(c)** **(d)** **(e)** **(f)**

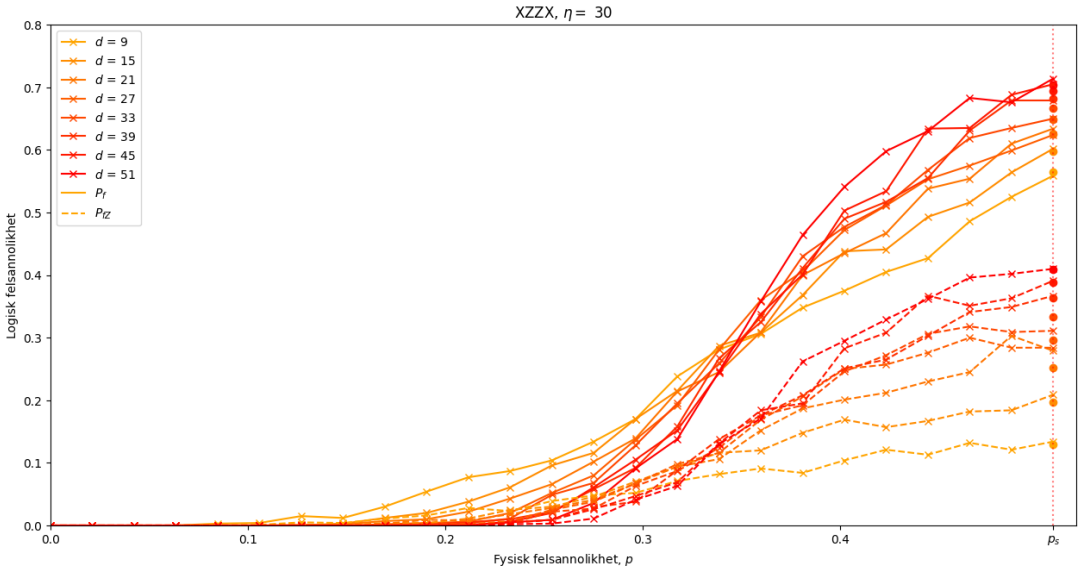
Figur 14: Den totala logiska felsannolikheten P_f samt sannolikheten för bit-flip P_{fZ} som funktion av koddistansen d vid $\eta = 30$ respektive $\eta = 300$ för XZZX-koden $\Big($(a) och (b)$\Big)$, XY-koden $\Big($(c) och (d)$\Big)$ samt XZZXdY-koden $\Big($(e) och (f)$\Big)$ vid $p = p_s$.

5.2 Felsannolikheter för $p \leq p_s$

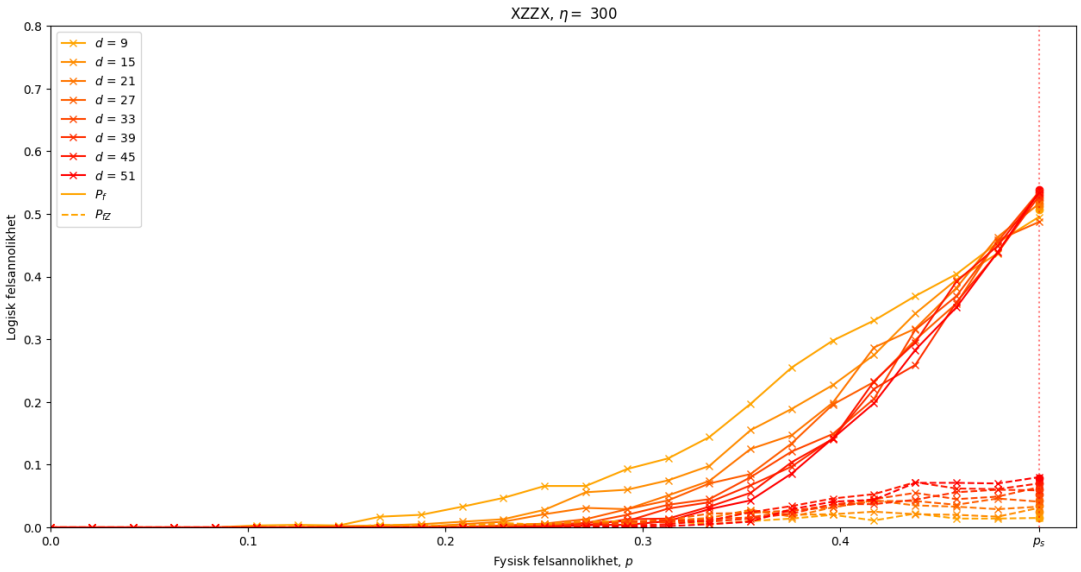
I figur 15, 16 och 17 visas både den logiska felsannolikheten samt sannolikheten för bit-flip fel för olika koddistanser mellan $d = 9$ till $d = 51$ samt för olika fysiska felsannolikheter p mellan 0 och $p_s \gtrsim 0{,}5$ vid $\eta = 30$ i (a) och $\eta = 300$ i (b) för XZZX-, XY- respektive XZZXdY-koden. Runda markörer vid $p = p_s$ markerar de analytiskt framtagna uttrycken.

Figur 15 för XZZX-koden uppvisar ett förväntat beteende där både den totala logiska felsannolikheten och bit-flipfelsannolikheten undertrycks för små fysiska felsannolikheter och där större kodstorlekar undertrycker felen mer än mindre kodstorlekar. Detta sker fram till en viss fysisk felsannolikhet, benämnd tröskelvärde, då större kodstorlekar genererar mer fel än mindre kodstorlekar. Att exakt bestämma tröskelvärdet kräver en noggrannare undersökning, men verkar vara för $0{,}3 < p < 0{,}4$ för $\eta = 30$ och för $0{,}3 < p < p_s$ för $\eta = 300$.

Även XY- och XZZXdY-koden förväntas uppvisa samma generella beteende som XZZX-koden, men i figur 16 och 17 är det tydligt att så inte är fallet. Båda figurerna uppvisar ett likartat beteende med stora logiska felsannolikheter även för små fysiska felsannolikheter och där den totala- och bit-flipfelsannolikheten följer varandra mycket nära. Vidare ger större koddistanser sämre logisk prestanda än små koddistanser för alla fysiska felsannolikheter. Ifall dessa simuleringsresultat hade stämt hade alltså båda koderna varit mycket opassande som kvantfelskorrigerande kretsar; det hade varit bättre inte korrigera alls. XY-koden är dock undersökt sedan tidigare [2] och uppvisade då inte detta beteende. Således dras slutsatsen att det vid de numeriska simuleringarna av XY- och XZZXdY-koden har skett något misstag, varpå figur 16 och 17 inte kan anses tillförlitliga. Det är emellertid värt att notera att simuleringsdatan fortfarande överensstämmer väl med den analytiska lösningen i den speciella punkten $p = p_s$. Vid simuleringarna av figur 15, 16 och 17 användes $\chi = 10$ och 1000 slumpade fel i varje punkt.

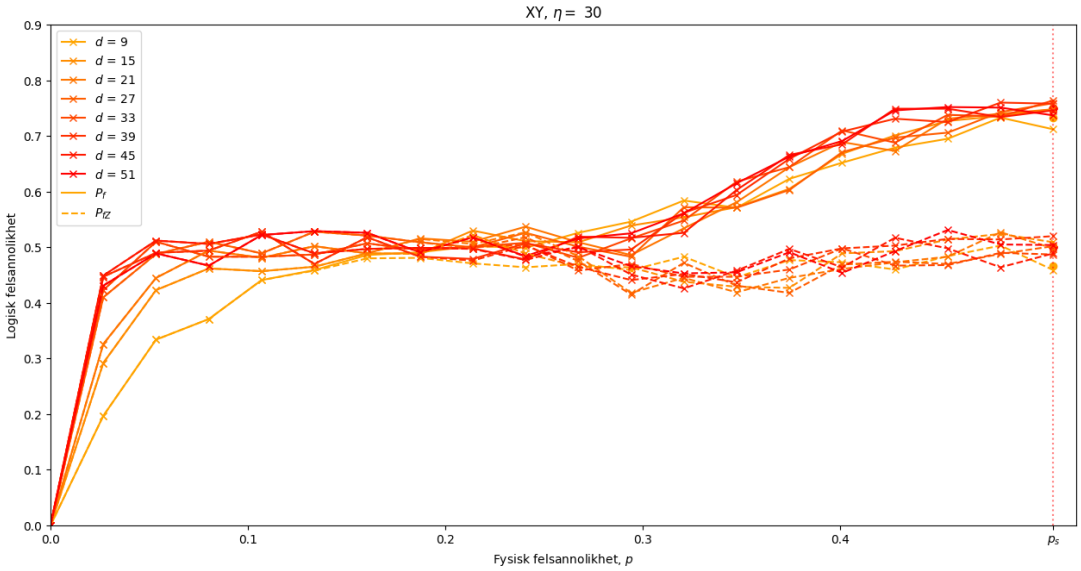


(a)

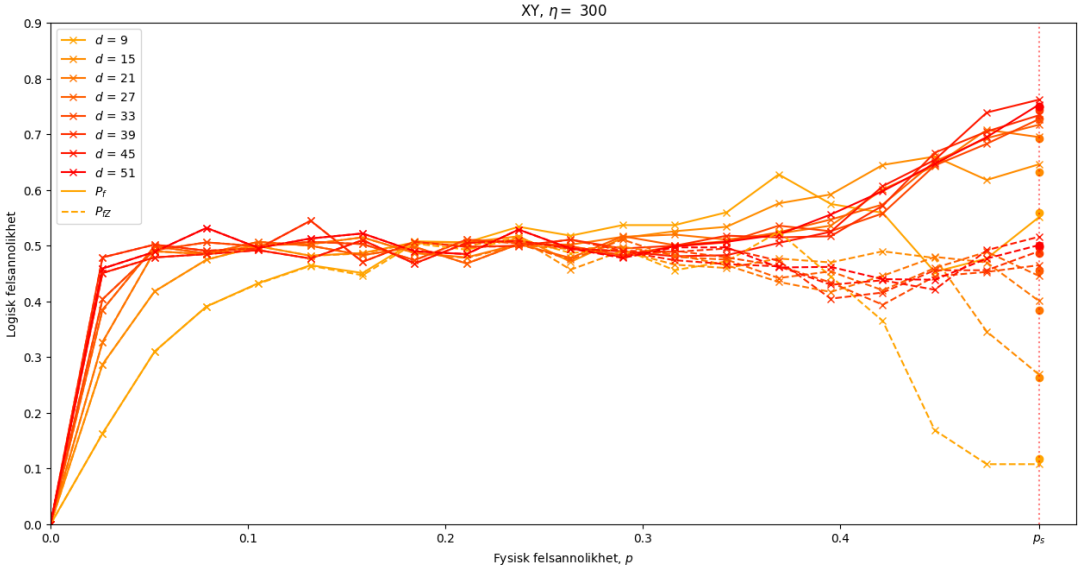


(b)

Figur 15: Den totala logiska felsannolikheten P_f samt felsannolikheten för bit-flip P_{fZ} som funktion av den fysiska felsannolikheten p för olika koddistanser med XZZX-koden med $\eta = 30$, (a), och $\eta = 300$, (b). Runda markörer vid $p = p_s$ visar resultatet från den analytiska lösningen.

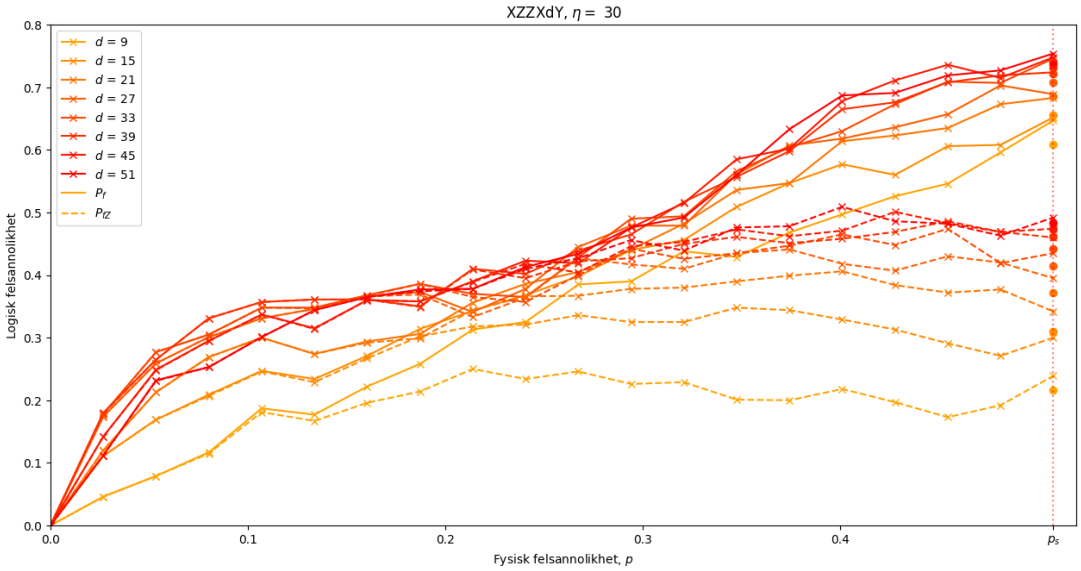


(a)

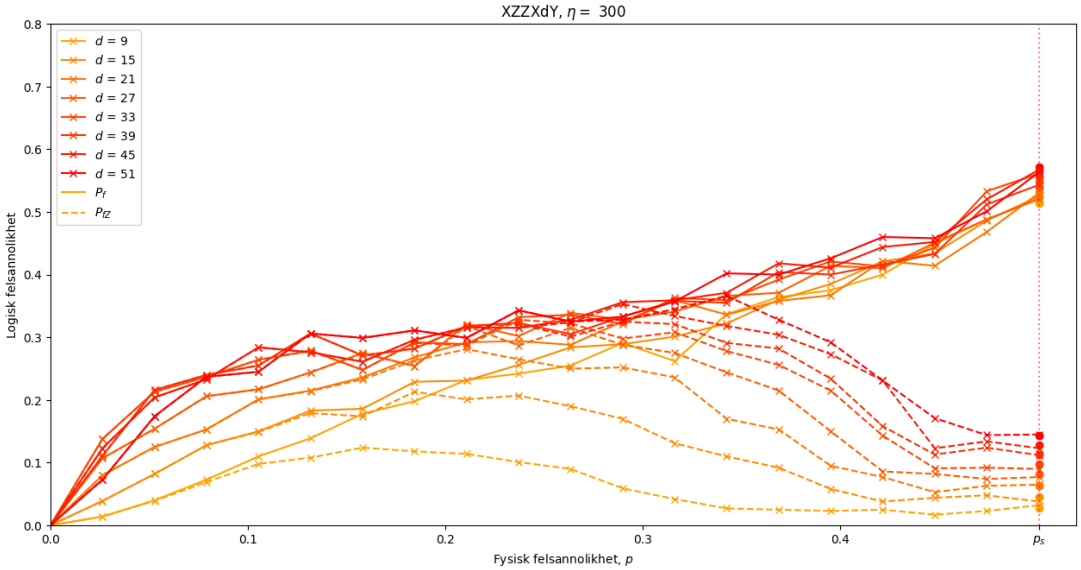


(b)

Figur 16: Den totala logiska felsannolikheten P_f samt felsannolikheten för bit-flip P_{fZ} som funktion av den fysiska felsannolikheten p för olika koddistanser med XY-koden med $\eta = 30$, (a), och $\eta = 300$, (b). Runda markörer vid $p = p_s$ visar resultatet från den analytiska lösningen.



(a)



(b)

Figur 17: Den totala logiska felsannolikheten P_f samt felsannolikheten för bit-flip P_{fZ} som funktion av den fysiska felsannolikheten p för olika koddistanser med XZZXdY-koden med $\eta = 30$, (a), och $\eta = 300$, (b). Runda markörer vid $p = p_s$ visar resultatet från den analytiska lösningen.

6 Diskussion

Som visades i avsnitt 5.2 uppvisade de numeriska simuleringarna för XY- och XZZXdY-koden ett oväntat beteende där större kodstorlekar korrigerade felen sämre än mindre kodstorlekar. Så tros fallet inte vara egentligen, utan båda koder borde ha bra felkorrigeringsförmågor för fasviktat brus. I gränsen $\eta \to \infty$, $p = p_z$ kan den totala logiska felsannolikheten lösas analytiskt för alla felsannolikheter p. I denna gräns förekommer endast fas-flipfel vilket gör att det endast finns två möjliga felkedjor C eller $Z_L C$. ML-avkodaren kommer alltid rätta med den mest sannolika av de två kedjorna, alltså den kortaste av de två. För att rättningen ska bli fel krävs att det applicerats Z-operatorer på fler än hälften av kvantbitarna som utgör Z_L. Varje sådan kedja utgörs av en binomialfördelning varpå den totala logiska felsannolikheten blir summan av sannolikheten för alla felkedjor med fler än hälften Z-applicerade längs Z_L, alltså

$$P_f(p) = \sum_{n=(d_Z+1)/2}^{d_Z} \binom{d_Z}{n} p^n (1-p)^{d_Z-n},$$

där d_Z återigen är det minsta antalet Z-operatorer som krävs för en Z_L operator enbart bestående av Z-operatorer. Vid kraftigt fasviktat brus, likt fallet $\eta = 300$, borde alltså logiska fel undertryckas kraftigare av främst XY-koden då denna har $d_Z = d^2$, men även XZZXdY-koden med $d_Z = 2d - 1$, än XZZX-koden som har $d_Z = d$. Att detta inte alls är fallet visar, som tidigare nämnt, att simuleringarna som visas inte är tillförlitliga.

Anledningen till varför simuleringarna blev felaktiga är inte helt klargjort och det finns flera möjliga orsaker. En möjlig anledning är att implementationerna av XY- och XZZXdY-koderna är felaktiga då dessa, till skillnad från XZZX-koden, inte fanns färdigimplementerad utan implementerades av kandidatarbetesgruppen själv. Att de numeriska simuleringarna överensstämmer mycket väl i punkten $p = p_s$ talar emellertid emot detta. Att endast mycket små modifikationer av redan existerande, och fungerande, kod krävdes för implementering av XY-koden talar också emot denna möjlighet, även om den inte kan avskrivas.

En annan möjlighet är att tensornätverksberäkningarna approximerades i för stor utsträckning genom val av för liten bindningsdimension χ. Vid simuleringarna användes $\chi = 10$. Denna möjlighet troliggörs av att en artikel som undersökt XY-koden funnit att XY-koden var mycket känslig för bindningsdimensionen runt tröskelvärdet [2]. Att både XY- och XZZXdY-koden uppvisar ett oförväntat beteende fram tills fysiska felsannolikheter där tröskelvärdet ungefär förväntas vara för att därefter konvergera väl till de analytiska resultaten vid $p = p_s$, vilket är över tröskelvärdet, tyder på att detta alltså kan vara fallet för båda koderna. På grund av långa simuleringstider, speciellt med större χ, har inte en utförlig analys av bindningsdimensionens påverkan kunnat genomföras i tillräckligt stor utsträckning för att säkert dra slutsatsen att det är bindningsdimensionen som har föranlett de felaktiga simuleringsresultaten och inte något fel i implementeringarna.

Med anledning av detta är en naturlig vidareutveckling av detta arbete att fortsätta undersöka både XY- och XZZXdY-koden med större bindningsdimension χ i syfte att erhålla uppskattningar av deras respektive tröskelvärden. Det därigenom erhållna tröskelvärdet för XY-koden kan sedan jämföras med tidigare uppskattningar [11]. Ökat χ innebär däremot längre simuleringstider varpå optimeringar av koden kan vara nödvändiga. Exempelvis kan koden parallelliseras eller andra implementeringsmöjligheter undersökas. En tänkbar alternativ implementeringsmetod bygger på att istället för att ändra ytkoden och avkodaren, likt genomfördes i detta arbete, justera felsan-

nolikheterna individuellt på kvantbitarna. Att undersöka andra typer av brus än fasviktat brus hade även det kunnat vara intressant.

I ekv. 7 och ekv. 8 visades att alla tre Clifford deformerade ytkoder hade logiska felsannolikheter på samma form men med olika Z-koddistanser d_Z, varpå frågan kan ställas vad den kodspecifika parametern d_Z betyder för ytkoderna. I alla tre fall sammanfaller d_Z med det minsta antalet Z-operatorer som krävs för att definiera en logisk Z-operator enbart bestående av rena Z-operatorer.

Det analytiska sambandet mellan d_Z och antalet Z-operatorer i ytkodernas Z_L-operatorer kan undersökas vidare på fler Clifford deformerade ytkoder. Så länge det går att visa att det endast finns en ren Z-operator, nämligen Z_L, så borde lösningsmetoderna i denna studie gå att applicera till fler CDSC. Svårigheterna ligger då i att hitta en överföringsmatris för ytkoden som har en rimlig storlek. Av intresse är även att visa hurvida sambandet stämmer för en allmän CDSC. Detta hade inneburit att egenskaper om ytkoden hade kunnat dras från direkt observation av ytkoden, närmare bestämt av antalet Z-operatorer i Z_L.

7 Slutsats

Den analytiska lösningen gav resultat som stämmer bra överens med de numeriska beräkningarna. Vi såg även att för alla tre ytkoder så erhölls samma resultat fast med olika värden på variabeln d_Z, något som kan undersökas vidare. Det numeriska resultatet för XZZX-koden ser rimligt ut och stämmer väl överens med tidigare studier [2]. Däremot är resultaten för XY- och XZZXdY-koden då $p < p_s$ orimliga. Detta tros bero på att dessa koder är mer känsliga för värdet på bindningsdimensionen χ. Att öka χ ökar dock beräkningstiden drastiskt och bättre beräkningar hann ej genomföras i denna studie. På grund av de orimliga resultaten kunde de tre ytkodernas prestanda för felkorrigering inte heller jämföras. Detta är något som bör utforskas vidare. Slutligen är det även av intresse att undersöka tröskelvärdet då större ytkoder slutar vara mer effetkiva än små ytkoder [2].

Referenser

[1] C. E. Shannon, "A Mathematical Theory of Communication," *Bell Syst. Tech. J.*, årg. 27, s. 623–656, juli 1948. DOI: 10.1145/584091.584093.

[2] Y. Xiao, B. Srivastava och M. Granath, "Exact results on finite size corrections for surface codes tailored to biased noise," *arXiv preprint*, 2024. DOI: 10.48550/arXiv.2401.04008. eprint: 2401.04008.

[3] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Physical Review A*, årg. 52, nr. 4, R2493–R2496, okt. 1995. DOI: 10.1103/PhysRevA.52.R2493.

[4] R. Hamming, "Error Detecting and Error Correcting Codes," *The Bell System Technical Journal*, årg. 29, nr. 9, s. 147–160, april 1950. DOI: 10.1002/j.1538-7305.1950.tb00463.x.

[5] E. Dennis, A. Kitaev, A. Landahl och J. Preskill, "Topological quantum memory," *Journal of Mathematical Physics*, årg. 43, nr. 9, s. 4452–4505, sept. 2002. DOI: 10.1063/1.1499754.

[6] A. Y. Kitaev, "Fault-tolerant quantum computation by anyons," *Annals of Physics*, årg. 303, nr. 1, s. 2–30, jan. 2003. DOI: 10.1016/S0003-4916(02)00018-0.

[7] Google Quantum AI, "Suppressing quantum errors by scaling a surface code logical qubit," *Nature*, årg. 614, s. 676–681, febr. 2023. DOI: 10.1038/s41586-022-05434-1.

[8] S. Krinner, N. Lacroix, A. Remm m. fl., "Realizing repeated quantum error correction in a distance-three surface code," *Nature*, årg. 605, s. 669–674, maj 2022. DOI: 10.1038/s41586-022-04566-8.

[9] E. Huang, A. Pesah, C. T. Chubb, M. Vasmer och A. Dua, "Tailoring Three-Dimensional Topological Codes for Biased Noise," *PRX Quantum*, årg. 4, nr. 3, s. 030 338, sept. 2023. DOI: 10.1103/prxquantum.4.030338.

[10] B. Srivastava, A. F. Kockum och M. Granath, "The XYZ^2 hexagonal stabilizer code," *Quantum*, årg. 6, s. 698, april 2022. DOI: 10.22331/q-2022-04-27-698.

[11] A. Dua, A. Kubica, L. Jiang, S. Flammia och M. Gullans, "Clifford-deformed Surface Codes," *PRX Quantum*, årg. 5, nr. 1, s. 010 347, mars 2024. DOI: 10.1103/PRXQuantum.5.010347.

[12] D. J. Griffiths och D. F. Schroeter, *Introduction to Quantum Mechanics*, 3. utg. Cambridge, UK: Cambridge University Press, 2018.

[13] D. Gottesman, "Stabilizer Codes and Quantum Error Correction," doktorsavhandling, California Institute of Technology, Pasadena, CA, USA, 1997. DOI: 10.48550/arXiv.quant-ph/9705052.

[14] C. D. Bruzewicz, J. Chiaverini, R. McConell och J. M. Sage, "Trapped-ion quantum computing: Progress and challenges," *Appl. Phys. Rev.*, årg. 6, nr. 2, s. 021 314, juni 2019. DOI: 10.1063/1.5088164.

[15] A. Bengtsson, P. Vikstål, C. Warren m. fl., "Improved Success Probability with Greater Circuit Depth for the Quantum Approximate Optimization Algorithm," *Phys. Rev. Appl.*, årg. 14, nr. 3, s. 034 010, sept. 2020. DOI: 10.1103/PhysRevApplied.14.034010.

[16] Y. Xiao, "Tensor network based decoders for topological stabilizer codes with diverse qubit error rates," examensarb., Dept. of Nanotechnology, Chalmers, Göteborg, Sverige, 2023. URL: http://hdl.handle.net/20.500.12380/306937.

[17] Glosser.ca, *Bloch Sphere*, https://commons.wikimedia.org/wiki/File:Bloch_Sphere.svg, dec. 2012.

[18] D. Gottesman, "The Heisenberg Representation of Quantum Computers," i *XXII International Colloquium on Group Theoretical Methods in Physics*, USA, 1998. DOI: 10.48550/arXiv.quant-ph/9807006. URL: https://doi.org/10.48550/arXiv.quant-ph/9807006.

[19] D. V. Schroeder, *An Introduction to Thermal Physics*. New York, NY, USA: Oxford University Press, 2021.

[20] S. Bravyi, M. Suchara och A. Vargo, "Efficient algorithms for maximum likelihood decoding in the surface code," *Phys. Rev. A*, årg. 90, nr. 3, s. 032 326, sept. 2014. DOI: 10.1103/PhysRevA.90.032326.

[21] A. Hutter, J. R. Wootton och D. Loss, "An efficient Markov chain Monte Carlo algorithm for the surface code," *Physical Review A*, årg. 89, nr. 2, s. 022 326, febr. 2014. DOI: 10.1103/PhysRevA.89.022326.

[22] K. Hammar, A. Orekhov, P. W. Hybelius m. fl., "Error-rate-agnostic decoding of topological stabilizer codes," *Phys. Rev. A*, årg. 105, nr. 4, s. 042 616, april 2022. DOI: 10.1103/PhysRevA.105.042616.

[23] D. K. Tuckett, "Tailoring surface codes: Improvements in quantum error correction with biased noise," (qecsim: https://github.com/qecsim/qecsim), doktorsavhandling, Faculty of Science, University of Sydney, Sydney, Australien, 2020. DOI: 10.25910/x8xw-9077.

[24] D. K. Tuckett, A. S. Darmawan, C. T. Chubb, S. Bravyi, S. D. Bartlett och S. T. Flammia, "Tailoring Surface Codes for Highly Biased Noise," *Phys. Rev. X*, årg. 9, nr. 4, s. 041 031, nov. 2019. DOI: 10.1103/PhysRevX.9.041031.

[25] J. P. B. Ataides, D. K. Tuckett, S. D. Bartlett, S. T. Flammia och B. J. Brown, "The XZZX Surface Code," *Nature Communications*, årg. 12, s. 2172, april 2021, qsdxzzx: https://bitbucket.org/qecsim/qsdxzzx/src/master/. DOI: 10.1038/s41467-021-22274-1.

[26] Wolfram Research, *Mathematica*, version 13, 2022. URL: https://www.wolfram.com/mathematica/.

[27] L. Christersson och A. Udén, *XZZXdY*, https://github.com/Lukasch123/XZZXdY, 2024.

A Omskrivning av $P_{\mathcal{X}}/P_{\mathcal{I}}$ i den analytiska lösningen

Vi har

$$\frac{P_{\mathcal{X}}}{P_{\mathcal{I}}} = \frac{1-\tanh^n(-k)}{1+\tanh^n(-k)} = \frac{1-\left(\frac{1-1/(2\eta)}{1+1/(2\eta)}\right)^n}{1+\left(\frac{1-1/(2\eta)}{1+1/(2\eta)}\right)^n},$$

där den sista likheten gäller för $k = -\frac{1}{2}\ln(2\eta)$. Genom omskrivningen

$$\left(\frac{1-1/(2\eta)}{1+1/(2\eta)}\right)^n = \exp\left(n\ln\left(\frac{1+1/(2\eta)}{1-1/(2\eta)}\right)\right),$$

erhålls sedan

$$\frac{P_{\mathcal{X}}}{P_{\mathcal{I}}} = \frac{1-\exp\left(2n\cdot\frac{1}{2}\ln\left((\frac{1+1/(2\eta)}{1-1/(2\eta)})^{-1}\right)\right)}{1+\exp\left(2n\cdot\frac{1}{2}\ln\left((\frac{1+1/(2\eta)}{1-1/(2\eta)})^{-1}\right)\right)},$$

vilket kan skrivas om med hjälp av identiteterna $\text{artanh(x)} = \frac{1}{2}\ln(\frac{1+\text{x}}{1-\text{x}})$ och $\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$ till

$$\frac{P_{\mathcal{X}}}{P_{\mathcal{I}}} = \tanh\left(n\ \text{artanh}(1/2\eta)\right).$$

B Omskrivning av logisk felsannolikhet

Som motiverades tidigare har vi

$$P_{fZ} = \frac{P_{\mathcal{X}}/P_{\mathcal{I}}}{1+P_{\mathcal{X}}/P_{\mathcal{I}}}.$$

Med $P_{\mathcal{X}}/P_{\mathcal{I}} = \tanh[d_Z\text{artanh}(1/2\eta)]$ får vi

$$P_{fZ} = \frac{\tanh[d_Z\text{artanh}(1/2\eta)]}{1+\tanh[d_Z\text{artanh}(1/2\eta)]}$$

vilket med $\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$ ger

$$P_{fZ} = \left(\frac{\exp(2d_Z\text{artanh}(1/2\eta)-1)}{\exp\left(2d_Z\text{artanh}(1/2\eta)+1\right)}\right)\Bigg/\left(1+\frac{\exp(2d_Z\text{artanh}(1/2\eta)-1)}{\exp(2d_Z\text{artanh}(1/2\eta)+1)}\right).$$

Genom förlängning med $e^{2d_Z\tanh(1/2\eta)}+1$ får vi

$$P_{fZ} = \frac{e^{2d_Z\text{artanh}(1/2\eta)}-1}{2e^{2d_Z\text{artanh}(1/2\eta)}} = \frac{1}{2} - \frac{1}{2}e^{-2d_Z\text{artanh}(1/2\eta)}.$$

Med $P_{fX} = P_{fY} = 1/2$ och $P_f = \frac{1}{2}(P_{fX}+P_{fY}+P_{fZ})$ erhålls direkt

$$P_f = \frac{3}{4} - \frac{1}{2}e^{-2d_Z\text{artanh}(1/2\eta)}.$$