Quantum Error Correction Final Research Project

Cathie So and Zoe Gonzalez Izquierdo

May 11, 2016

1 Problem Statement

We have studied three methods for dealing with decoherence in quantum computation: decoherence free subspace/noiseless subsystem (DFS-NS), dynamical decoupling (DD) and quantum error correcting codes (QEC), as well as a few hybrid/modified versions of those. We can judge their merits on the basis of how high their fidelity is and how low are the resources (time, qubits) that they require; we would of course like to keep the fidelity as high and the resources as low as possible, but unfortunately there's usually a trade-off. In this project we attempt to quantify how those quantities compare for the different methods. To do so, we take a system and bath whose Hamiltonian can be written as:

$$H(t) = H_S(t) + H_B + H_{SB}$$

Where the interaction Hamiltonian H_{SB} has the form:

$$H_{SB} = \left(\sum_{i=1}^{N} \sigma_i^x\right) \otimes B^x + \left(\sum_{i=1}^{N} \sigma_i^y\right) \otimes B^y + \sum_{i=1}^{N} \sigma_i^z \otimes B_i^z = S^x \otimes B^x + S^y \otimes B^y + \sum_{i=1}^{N} \sigma_i^z \otimes B_i^z$$

This means that all qubits are individually coupled to the z component of the bath, while the coupling to the x and y components is global.

To make matters somewhat simpler, we will set:

$$H_B = 0$$

This is an ideal case that does not accurately represent reality, but this constraint shouldn't prevent us from obtaining meaningful results.

We can also take our initial state to be logical zero, that is:

$$|\phi(t=0)\rangle = |00...0\rangle = |\bar{0}\rangle$$

In the following sections, we examine how each of the methods mentioned above fares when trying to keep this initial state from decoherence, as well as the amount of resources employed in the process.

2 Decoherence Free Subspace

The noise is modeled by H_{SB}. Because we can't find a simultaneous eigenstate of S^x, S^y and σ_i^z, there isn't a Hamiltonian DFS which would result in fidelity 1. We must look to collective decoherence instead, and accept that fidelity will be less than one. We also know from our lectures and homework that the rate in this case will be:

$$1 - \frac{3}{2}\frac{log_2 N}{N}$$

For simplicity, let's assume N is even. Then our initial state will be encoded as the DFS logical $|0\rangle$, that is:

$$|\bar{0}\rangle = \bigotimes_{i \text{ odd}} |s\rangle_{i,i+1}$$

Where $|s\rangle_{i,i+1}$ is the singlet state:

$$|s\rangle_{i,i+1} = \frac{1}{\sqrt{2}}(|0_i 1_{i+1}\rangle - |1_i 0_{i+1}\rangle)$$

Our goal is to find out how our state (which is in the code space) evolves. For this, we need to find the evolution operator acting on the code space:

$$U(t)|_{C(S)} = e^{-itH_{SB}}|_{C(S)}$$

$$\begin{aligned} U(t)|_{C(S)} &= e^{-\frac{t^2}{2}\left[\sum_i \sigma_i^z \otimes B_i^z, S^x \otimes B^x + S^y \otimes B^y\right]} e^{-it\sum_i \sigma_i^z \otimes B_i^z} e^{-it(S^x \otimes B^x + S^y \otimes B^y)} + \mathcal{O}(t^3)|_{C(S)} = \\ &= \prod_i e^{-i\frac{t^2}{2}\left(-\sigma_i^y \otimes \left\{B^x, B_i^z\right\} + \sigma_i^x \otimes \left\{B^y, B_i^z\right\}\right)} \end{aligned}$$

Where we have used the BCH equation up to second order for the first line, and then realized that, since we're restricting this operator to the code space, the last exponential acts as just the identity.

To see what this does to our initial state it is easiest to look at its effects on just one pair (e.g. 1, 2 for simplicity). Let $|\phi(t)\rangle$ be the evolved state:

$$\begin{aligned} |\phi(t)\rangle &= e^{-it(\sigma_1^z \otimes B_1^z + \sigma_2^z \otimes B_2^z)} |s\rangle_{1,2} \approx \\ &\approx (\mathbb{1} - it\sigma_1^z \otimes B_1^z - \frac{t^2}{2}\mathbb{1} \otimes (B_1^z)^2)(\mathbb{1} - it\sigma_2^z \otimes B_2^z - \frac{t^2}{2}\mathbb{1} \otimes (B_2^z)^2) |s\rangle_{1,2} \approx \\ &\approx [\mathbb{1} - it\sigma_1^z \otimes B_1^z - it\sigma_2^z \otimes B_2^z - \frac{t^2}{2}\mathbb{1} \otimes ((B_1^z)^2 + (B_2^z)^2)] |s\rangle_{1,2} \approx \\ &\approx |s\rangle_{1,2} - ct\,|t_0\rangle_{1,2} - \frac{t^2}{2}\|B_1^z - B_2^z\|^2 |s\rangle_{1,2} \end{aligned}$$

$|t_0\rangle$ belongs to the triplet state and is thus orthogonal to $|s\rangle$. We have omitted the terms which act trivially when restricted to the code space.

To compute the fidelity:

$$\mathcal{F}(|s\rangle) \approx |\langle s| \, |\phi(t)\rangle \, |$$

But this is for just one pair $|s\rangle_{i,i+1}$. Thus the total fidelity:

$$\mathcal{F} \approx (\mathcal{F}(|s\rangle))^{N/2} \approx 1 - Nt^2 max_i(\left\|B_i^z - B_{i+1}^z\right\|^2)$$

Which, by applying the triangle inequality, we can bound by:

$$\mathcal{F} \approx 1 - Nt^2(\max_i \|B_i^z\|^2)$$

3 Dynamical Decoupling

We will make the assumption that the pulses are ideal, i.e. delta functions, so that H_{SB} only acts between pulses, so the corresponding evolution operator is:

$$f_\tau = e^{-i\tau H_{SB}}$$

We first apply Z to all the qubits to deal with the x and y terms:

$$\overline{Z} f_\tau \overline{Z} = e^{-i\tau(-S^x \otimes B^x - S^y \otimes B^y + \sum_{i=1}^N \sigma_i^z \otimes B_i^z)}$$

Where $\overline{Z} \equiv Z_1 Z_2 ... Z_N$.

So the evolution operator after this becomes:

$$U_1 = \overline{Z} F_\tau \overline{Z} F_\tau = e^{-i2\tau \sum_{i=1}^N \sigma_i^z \otimes B_i^z + \tau^2 \left[S^x \otimes B^x + S^y \otimes B^y, \sum_{i=1}^N \sigma_i^z \otimes B_i^z\right]}$$

We now need to apply X to every qubit sequentially, which will suppress the z terms in H_{SB}:

$$X_1 U_1 X_1 = e^{-i2\tau \sum_{i=1}^N \sigma_i^z \otimes B_i^z + i2\tau \sigma_1^z \otimes B_1^z + \tau^2 \left[S^x \otimes B^x + (S^y - 2\sigma_1^y) \otimes B^y, -\sigma_1^z \otimes B_1^z + \sum_{i=2}^N \sigma_i^z \otimes B_i^z\right]}$$

$$U_2 = X_1 U_1 X_1 U_1 = e^{-i4\tau \sum_{i=3}^N \sigma_i^z \otimes B_i^z + 2\tau^2 \left[S^x \otimes B^x + S^y \otimes B^y, \sum_{i=2}^N \sigma_i^z \otimes B_i^z\right]}$$
$$-2\tau^2 \left[\sigma_1^y \otimes B^y, \sum_{i=2}^N \sigma_i^z \otimes B_i^z - \sigma_1^z \otimes B_1^z\right] - 4\tau^2 \left[\sum_{i=2}^N \sigma_i^z \otimes B_i^z - \sigma_1^z \otimes B_1^z, \sum_{i=1}^N \sigma_i^z \otimes B_i^z\right]$$

$$X_2U_2X_2 = e^{-i4\tau\sum_{i=3}^{N}\sigma_i^z\otimes B_i^z + i4\tau\sigma_2^z\otimes B_2^z - 2\tau^2\left[\sigma_1^y\otimes B^y, -\sum_{i=1}^{2}\sigma_i^z\otimes B_i^z\right]}$$
$$^{+2\tau^2\left[S^x\otimes B^x + (S^y - 2\sigma_2^y)\otimes B^y, \sum_{i=3}^{N}\sigma_i^z\otimes B_i^z - \sigma_2^z\otimes B_2^z\right] - 4\tau^2\left[\sigma_2^y\otimes B^y, -\sigma_2^z\otimes B_2^z\right]} =$$
$$= e^{-i4\tau\sum_{i=3}^{N}\sigma_i^z\otimes B_i^z + i4\tau\sigma_2^z\otimes B_2^z - 2\tau^2\left[\sigma_1^y\otimes B^y, -\sigma_1^z\otimes B_1^z\right] +}$$
$$^{+2\tau^2\left[S^x\otimes B^x + S^y\otimes B^y, \sum_{i=3}^{N}\sigma_i^z\otimes B_i^z\right] - 4\tau^2\left[\sigma_2^y\otimes B^y, -\sigma_2^z\otimes B_2^z\right]}$$

$$U_3 = X_2U_2X_2U_2 = e^{-i8\tau\sum_{i=3}^{N}\sigma_i^z\otimes B_i^z - 4\tau^2\left[\sigma_1^y\otimes B^y, -\sigma_1^z\otimes B_1^z\right] - 4\tau^2\left[\sigma_2^y\otimes B^y, -\sigma_2^z\otimes B_2^z\right]}$$
$$^{+2\tau^2\left[S^x\otimes B^x + S^y\otimes B^y, \sum_{i=3}^{N}\sigma_i^z\otimes B_i^z\right]}$$

This is enough to observe a pattern and deduce that the evolution operator once all the qubits have been acted upon will be:

$$U_{N+1} = e^{2^N\tau^2\sum_{i=1}^{N}\left[\sigma_i^y\otimes B^y, \sigma_i^z\otimes B_i^z\right] + \mathcal{O}(t^3)}$$

Notice that we no longer have a B^x term. This is due to the fact that we pulsed with Z and then with X, but we could also have made B^y disappear instead. We will eventually see that the decision to get rid of one or the other should be taken beforehand depending on which one has a larger norm.

To find U_{N+1} above we applied the pulse sequences one qubit at a time. We thought it would be interesting to compare with the case in which we apply all the X operators at once, which would lower the required resources. U_1 is the same one as before, but then we act on it with $\overline{X} \equiv X_1X_2...X_N$:

$$\overline{X}U_1\overline{X} = e^{2i\tau\sum_{i=1}^{N}\sigma_i^z\otimes B_i^z - \tau^2\left[S^y\otimes B^y, \sum_{i=1}^{N}\sigma_i^z\otimes B_i^z\right]}$$

$$\overline{U} = \overline{X}U_1\overline{X}U_1 = e^{2\tau^2\left[S^y\otimes B^y, \sum_{i=1}^{N}\sigma_i^z\otimes B_i^z\right] + \mathcal{O}(t^3)}$$

Comparing U_{N+1} and $\overline{U}$ we see that the commutator is the same, and the only difference between them is the 2^N factor for U_{N+1} versus the 2 for $\overline{U}$; this means we can use this last method that saves some resources and still get better fidelity, which is very good news.

To find the fidelity we need to evolve the initial state with $\overline{U}$.

$$\rho_S(0) = |\overline{0}\rangle\langle\overline{0}|$$

And we want to compute:

$$\overline{U}\rho_S(0)\otimes\rho_B(0)\overline{U}^\dagger$$

To find our evolved state we can look at one "half" of that expression:

$$
\begin{aligned}
&e^{2\tau^2\left[S^y\otimes B^y,\sum_{i=1}^N\sigma_i^z\otimes B_i^z\right]}\left|0...0\right\rangle = \\
&= e^{2\tau^2\sum_i\left[\sigma_i^y\otimes B^y,\sigma_i^z\otimes B_i^z\right]}\left|0...0\right\rangle = \\
&= \prod_i e^{2\tau^2\left[\sigma_i^y\otimes B^y,\sigma_i^z\otimes B_i^z\right]}\left|0...0\right\rangle = \\
&= \bigotimes_{i=1}^N(e^{2\tau^2\left[\sigma_i^y\otimes B^y,\sigma_i^z\otimes B_i^z\right]}\left|0...0\right\rangle)
\end{aligned}
$$

The only non trivial commutators are those for which σ^y and σ^z act on the same qubit. Those have the form:

$$[\sigma_i^y\otimes B^y,\sigma_i^z\otimes B_i^z]=\sigma_i^y\sigma_i^z\otimes B^yB_i^z-\sigma_i^z\sigma_i^y\otimes B_i^zB^y=i\sigma_i^x\otimes\{B^y,B_i^z\}$$

It is now convenient to write $\left|0\right\rangle$ in the $\left|+\right\rangle,\left|-\right\rangle$ basis instead. Then our previous expression becomes:

$$(\frac{1}{\sqrt{2}})^N\bigotimes_i(e^{i2\tau^2\left\{B^y,B_i^z\right\}}\left|+\right\rangle+e^{-i2\tau^2\left\{B^y,B_i^z\right\}}\left|-\right\rangle)\otimes\sum_\nu\lambda_\nu\left|\nu\right\rangle_B\left\langle\nu\right|$$

Where we have just written $\rho_B(0)$ using its spectral decomposition. Before we can proceed any further we need to make some assumption about the bath operators. We are trying to keep this comparison as general as possible so we'd like to avoid choosing some particular operators. A sufficient assumption that we feel is weak enough to allow for fairly general results is:

$$\{B^y,B_i^z\}=\lambda_\nu\left|\nu\right\rangle$$

We can then take the worst case scenario given this assumption; that every term gives the maximum eigenvalue:

$$\lambda\equiv\max_i\|\{B^y,B_i^z\}\|$$

So we can finally write our evolved state:

$$\left|\psi\right\rangle=\bigotimes_i\frac{e^{i2\tau^2\lambda}\left|+\right\rangle+e^{-i2\tau^2\lambda}\left|-\right\rangle}{\sqrt{2}}$$

This expression already reveals that the fidelity will oscillate. For the actual computation:

$$\mathcal{F} = |\langle\overline{0}| \, |\psi\rangle \, |= |\bigotimes_i \langle 0| \, (\frac{e^{i2\tau^2\lambda} \, |+\rangle + e^{-i2\tau^2\lambda} \, |-\rangle}{\sqrt{2}})|=$$

$$|\bigotimes_i \frac{e^{i2\tau^2\lambda} + e^{-i2\tau^2\lambda}}{2}|= |\bigotimes_i cos(2\tau^2\lambda)|= |cos(2\tau^2\lambda)|^N$$

After expanding, we arrive at:

$$\mathcal{F} = 1 - 2N\tau^4\lambda^2 + \mathcal{O}(t^5)$$

This is good news; the leading term is of order 4, so for small τ the value will be very close to one.

4 Quantum Error Correction

To tackle the QEC method, the first thing we need to do is bridge the gap between the formalism we normally use with this scheme - noise channel given by some completely positive (CP) map

$$\mathcal{N}(\rho) = \sum_i E_i \rho E_i^\dagger$$

where E_i are the error operators - and the problem at hand, defined instead by a Hamiltonian.

The error operators can be written in the form:

$$E_i = \sqrt{p_i} U_i$$

Where U_i are unitary operators and p_i are probabilities, which means that their action produces rotation plus dilation (we know this from the Knill-Laflamme theorem). We need to realize that these will be none other than the Kraus operators associated with our system-bath Hamiltonian H_{SB}:

$$K_{\mu\nu}(t) = \sqrt{\lambda_\nu} \, \langle\mu| \, U(t) \, |\nu\rangle$$

$$U(t) = e^{-itH_{SB}}$$

Using the BCH formula for the exponential we can write:

$$K_{\mu\nu}(t) = \sqrt{\lambda_\nu} \langle mu| e^{-itH_{SB}} |\nu\rangle = \sqrt{\lambda_\nu} \langle \mu| e^{-itS^x\otimes B^x} e^{-itS^y\otimes B^y} e^{-it\sum_{i=1}^N \sigma_i^z\otimes B_i^z} |\nu\rangle + \mathcal{O}(t^2) =$$
$$= \sqrt{\lambda_\nu} \langle mu| e^{-itH_{SB}} |\nu\rangle = \sqrt{\lambda_\nu} \langle \mu| e^{-itS^x\otimes B^x} e^{-itS^y\otimes B^y} \prod_{i=1}^N e^{-it\sigma_i^z\otimes B_i^z} |\nu\rangle + \mathcal{O}(t^2)$$

Introducing identities:

$$K_{\mu\nu}(t) = \sqrt{\lambda_\nu} \sum_{\alpha,\beta_i} \langle \mu| e^{-itS^x\otimes B^x} |\alpha\rangle \langle \alpha| e^{-itS^y\otimes B^y} |\beta_1\rangle \langle \beta_1| e^{-it\sigma_1^z\otimes B_1^z} |\beta_2\rangle \langle \beta_2| \ldots$$
$$\ldots |\beta_N\rangle \langle \beta_N| e^{-it\sigma_N^z\otimes B_N^z} |\nu\rangle + \mathcal{O}(t^2)$$

Expanding the exponentials up to first order:

$$K_{\mu\nu}(t) = \sqrt{\lambda_\nu} \sum_{\alpha,\beta_i} (\delta_{\mu\alpha}\mathbb{1} - itS^x \langle \mu| B^x |\nu\rangle)(\delta_{\alpha\beta_1}\mathbb{1} - itS^y \langle \alpha| B^y |\beta_1\rangle)\cdot$$
$$\cdot \prod_{i=1}^{N-1} (\delta_{\beta_i\beta_{i+1}}\mathbb{1} - it\sigma_i^z \langle \beta_i| B_i^z |\beta_{i+1}\rangle)(\delta_{\beta_N\nu}\mathbb{1} - it\sigma_N^z \langle \beta_N| B_N^z |\nu\rangle) + \mathcal{O}(t^2) =$$
$$= \sqrt{\lambda_\nu} \sum_{\alpha,\beta_i} \delta_{\mu\alpha}\delta_{\alpha\beta_1} \prod_{i=1}^{N-1} \delta_{\beta_i\beta_{i+1}}\mathbb{1} - itS^x \langle \mu| B^x |\alpha\rangle \delta_{\alpha\beta_1} + \sum_{i=1}^{N-1} \delta_{\beta_i\beta_{i+1}} + \delta_{\beta_N\nu} +$$
$$+ S^y \langle \alpha| B^y |\beta_1\rangle (\delta_{\mu\alpha} + \sum_{i=1}^{N-1} \delta_{\beta_i\beta_{i+1}} + \delta_{\beta_N\nu}) + \sum_{i=1}^{N-1} \sigma_i^z \langle \beta_i| B_i^z |\beta_{i+1}\rangle \delta_{\mu\alpha} + \delta_{\alpha\beta_1} + \sum_{i=2}^{N-1} \delta_{\beta_i\beta_{i+1}} + \delta_{\beta_N\nu} +$$
$$+ \sigma_N^z \langle \beta_N| B_N^z |\nu\rangle \delta_{\mu\alpha} + \delta_{\alpha\beta_1} + \sum_{i=1}^{N-1} \delta_{\beta_i\beta_{i+1}}$$

The first term becomes simply $\mathbb{1}$. For the second term:

$$\sum_{\alpha\beta_N} -itS^x \langle \mu| B^x |\alpha\rangle (1 + \sum_{i=1}^{N-1} \sum_{\beta_i\beta_{i+1}} \delta_{\beta_i\beta_{i+1}} + \delta_{\beta_N\nu})$$
$$= \sum_{\alpha} -itS^x \langle \mu| B^x |\alpha\rangle [(N-1) + (N-1)^2 + (N-1)] = \sum_{\alpha} -itS^x \langle \mu| B^x |\alpha\rangle (N^2 - 1)$$

Summing the other terms in a similar fashion we arrive at the expression:

$$K_{\mu\nu}(t) = \sqrt{\lambda_\nu}\mathbb{1} - itN^2(S^x\|B^x\|_{max}+S^y\|B^y\|_{max}+\sum_i \sigma_i^z\|B^z\|_{max})$$

To eventually arrive at the fidelity we first need to compute the effect of the noise channel:

$$\begin{aligned}\mathcal{N}(\rho_S(0),t) \equiv \rho_S(t) = Tr_B[U(t)\rho_S(0)\otimes\rho_B(0)U^\dagger(t)] = \\ = Tr_B[(\mathbb{1} - itH_{SB} - \frac{t^2}{2}H_{SB}^2)\rho_0(\mathbb{1} + itH_{SB} - \frac{t^2}{2}H_{SB}^2)] = \\ = Tr_B(\rho_0 - it[H_{SB},\rho_0] - \frac{t^2}{2}\{H_{SB}^2,\rho_0\} + t^2 H_{SB}\rho_0 H_{SB})\end{aligned}$$

Where we have defined $\rho_0 \equiv \rho_S(0)\otimes\rho_B(0)$.

We can see right away that the first term corresponds to the absence of errors, the second one to a weight 1 error, and the third and fourth ones to weight 2 errors. This means that, to remove the t^2 order term, our code would need to correct at least weight 2 errors.

Let's look at each term separately. The first one is simply:

$$Tr_B(\rho_0) = \rho_S(0)$$

For the last one, ignoring for now the t^2:

$$\begin{aligned}H_{SB}\rho_0 H_{SB} = (S^x\otimes B^x + S^y\otimes B^y + \sum_i \sigma_i^z\otimes B_i^z)\rho_S(0)\otimes\rho_B(0)(S^x\otimes B^x + S^y\otimes B^y + \sum_i \sigma_i^z\otimes B_i^z) = \\ = S^x\rho_S S^x\otimes B^x\rho_B B^x + S^y\rho_S S^y\otimes B^y\rho_B B^y + S^x\rho_S S^y\otimes B^x\rho_B B^y + S^y\rho_S S^x\otimes B^y\rho_B B^x + ...\end{aligned}$$

We then take the partial trace:

$$Tr_B(...) = S^x\rho_S S^x\langle(B^x)^2\rangle_{\rho_B} + S^y\rho_S S^y\langle(B^y)^2\rangle_{\rho_B} + S^x\rho_S S^y\langle B^x B^y\rangle_{\rho_B} + ...$$

We can get an upper bound by using:

$$\langle(B^x)^2\rangle_{\rho_B} \leq \|(B^x)^2\| = \|B^x\|^2$$

$$\langle B^x B^y\rangle_{\rho_B} \leq \|B^x B^y\| = \|B^x\|\|B^y\|$$

Then:

$$Tr_B(H_{SB}\rho_0 H_{SB}) \leq (\|B^x\|S^x+\|B^y\|S^y+\sum_i \sigma_i^z\|B_i^z\|)\rho_S(0)(\|B^x\|S^x+\|B^y\|S^y+\sum_i \sigma_i^z\|B_i^z\|)$$

And we can define the following operator which will be useful later:

$$E \equiv t(\|B^x\|S^x + \|B^y\|S^y + \sum_i \sigma_i^z\|B_i^z\|)$$

By comparing this expression to the standard form of an error operator we can see that $\sqrt{p} = \mathcal{O}(t)$, and thus $p = \mathcal{O}(t^2)$.

Let our density matrix for the system be initialized to:

$$\rho_S(0) = |\overline{0}\rangle\langle\overline{0}|$$

The fidelity is given by:

$$F(\rho_S, \mathcal{N}(\rho_S)) = \sqrt{\langle\overline{0}|\,\mathcal{N}(\rho_S)\,|\overline{0}\rangle}$$

The second and third terms won't contribute to the fidelity. We can see this, for instance, by examining the second term. Once we apply the partial trace to the commutator we obtain:

$$\|B^x\|[S^x, \rho_S] + \|B^y\|[S^y, \rho_S] + \sum_i \|B^z\|[\sigma_i^z, \rho_S]$$

But when we use the expression inside of the square root in the fidelity:

$$\langle\overline{0}|\,[S^x, \rho_S]\,|\overline{0}\rangle = \langle\overline{0}|\,S^x\,|\overline{0}\rangle - \langle\overline{0}|\,S^x\,|\overline{0}\rangle = 0$$

And the same happens for the other terms.

With the results above we can write an expression for the "effective" noise channel:

$$\mathcal{N}_{eff}(\rho_S(0), t) = \rho_S(0) + E\rho_S(0)E^\dagger + \mathcal{O}(t^3)$$

Now recall that the condition for QEC is:

$$PE^2P = \gamma P$$

Where $P \equiv \sum_i |\overline{\psi_i}\rangle\langle\overline{\psi_i}|$ and $|\overline{\psi_i}\rangle$ is in the code space.

This amounts to the following conditions:

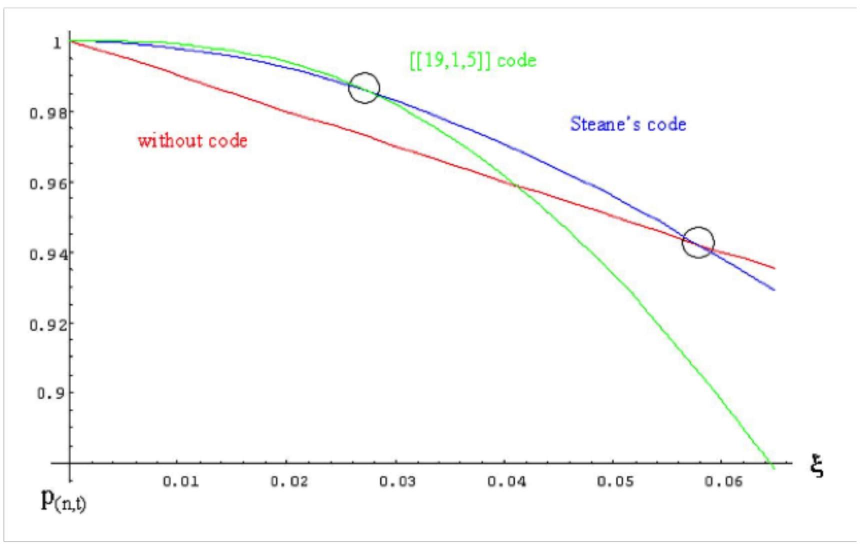
$$\begin{aligned} P\sigma_i^x\sigma_j^x P &= \gamma_{ij}^x P & \qquad P\sigma_i^x\sigma_j^z P &= \gamma_{ij}^{xz} P \\ P\sigma_i^y\sigma_j^y P &= \gamma_{ij}^y P & \qquad P\sigma_i^x\sigma_j^y P &= \gamma_{ij}^{xy} P \\ P\sigma_i^z\sigma_j^z P &= \gamma_{ij}^z P & \qquad P\sigma_i^y\sigma_j^z P &= \gamma_{ij}^{yz} P \end{aligned}$$

$$P\sigma_i^x P = \gamma_i^x P \qquad P\sigma_i^y P = \gamma_i^y P \qquad P\sigma_i^z P = \gamma_i^z P$$

From those we can deduce that $|\overline{\psi_i}\rangle$ needs to be symmetric in x, y and z. Thus $|\overline{\psi_i}\rangle$ could be some kind of "singlet state", or be trivially 0, or possess cyclic property. We're back to the DFS situation. Shor code might work, up to $\mathcal{O}(t^3)$.

We said above that our code would need to correct errors of at least weight 2. Recall $d = 2t + 1$, where t is the weight and d the distance. This means we'd need a QED of $d = 5$, while all the codes we've seen so far have $d = 3$.

Peter Majek, in his 2005 thesis, proposed a [[19,1,5]] code. But, aside from the obvious disadvantage of requiring 19 physical qubits to encode a logical one, it didn't offer much benefit in terms of error probability when compared to Steane's code. Below we can see a graph comparing the probability of the successful survival of a single encoded qubit subjected to single gate as a function of the probability of a single gate failure ξ:



We can see that for very low probability of gate failure this code works slightly better than Steane's, but as soon as ξ increases, the [[19,1,5]] rapidly loses its advantage, quickly becoming worse than no code at all. It could be a worthwhile candidate if the probability of gate failure is small, although it would have to be seen whether its seemingly small advantage over Steane's code justifies the much higher amount of resources.

Still, the creation of other distance 5 codes remains an open and interesting problem.

5 Hybrid method: DFS + DD

We can take the effective evolution operator from DFS:

$$U_{eff}(\tau) = e^{-\frac{\tau^2}{2}\sum_i\left[\sigma_i^x\otimes B^x+\sigma_i^y\otimes B^y,\sigma_i^z\otimes B_i^z\right]}e^{-i\sum_i\sigma_i^z\otimes B_i^z\tau}$$

So the DD evolution operator will be:

$$U' = \overline{X}U_{eff}\overline{X}U_{eff}$$

$$\overline{X}U_{eff}\overline{X} = e^{-\frac{\tau^2}{2}\sum_i\left[\sigma_i^x\otimes B^x-\sigma_i^y\otimes B^y,\sigma_i^z\otimes B_i^z\right]}e^{i\sum_i\sigma_i^z\otimes B_i^z\tau}$$

$$U' = e^{-\frac{\tau^2}{2}\sum_i\left[\sigma_i^x\otimes B^x-\sigma_i^y\otimes B^y,\sigma_i^z\otimes B_i^z\right]}e^{\frac{\tau^2}{2}\left[\sum_i\sigma_i^z\otimes B_i^z,-\sum_j\sigma_j^z\otimes B_j^z\right]}e^{-\frac{\tau^2}{2}\sum_i\left[\sigma_i^x\otimes B^x+\sigma_i^y\otimes B^y,\sigma_i^z\otimes B_i^z\right]}$$

Up to second order in τ:

$$\begin{aligned} U' = e^{-\frac{\tau^2}{2}\sum_i 2\left[\sigma_i^x\otimes B^x,\sigma_i^z\otimes B_i^z\right]-\frac{\tau^2}{2}\sum_{ij}\left[\sigma_i^z\otimes B_i^z,\sigma_j^z\otimes B_j^z\right]} &= \\ = e^{\tau^2\sum_i\sigma_i^y\otimes\left\{B^x,B_i^z\right\}} \end{aligned}$$

Making the same assumptions as for DD, taking:

$$\lambda = \max_i \|\{B^x, B_i^z\}\|$$

We find the fidelity to be:

$$\mathcal{F} = (cos(\tau^2\lambda))^N = 1 - \frac{N\tau^4\lambda^2}{2} + \mathcal{O}(t^5)$$

So there is at least some improvement compared to just one of the methods. Recall that, for DD:

$$\mathcal{F}_{DD} = 1 - 2N\tau^4\lambda^2 + \mathcal{O}(t^5)$$

So we've made the term in τ^4 4 times smaller.

6 Simulations

We thought it would be interesting to perform some numerical simulations that would allow us to better visualize our results. For this we used MATLAB (our codes are included in the appendix). To run the simulations we had to pick a particular set of bath operators. We decided that the bath would have as many qubits as the system, and also the same operators. Then:

$$B^x = S^x$$

$$B^y = S^y$$

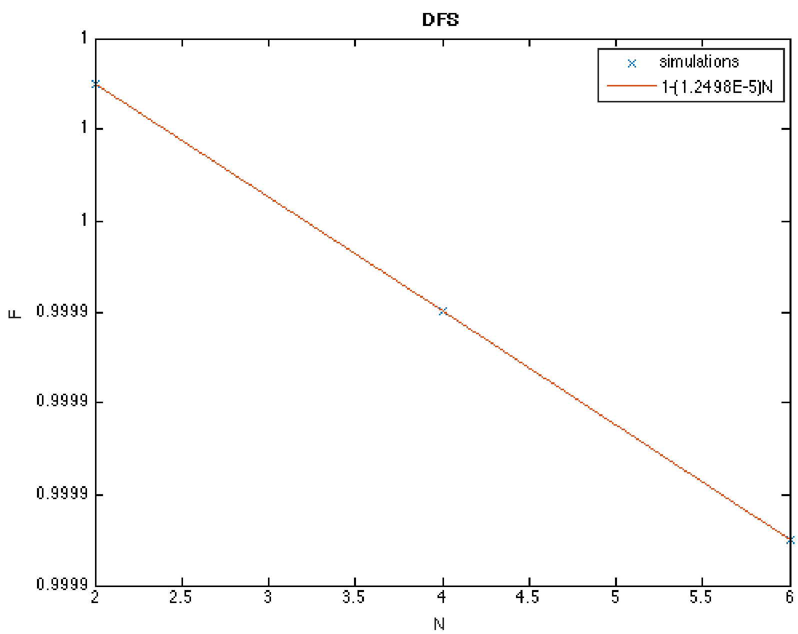
$$B_i^z = \sigma_i^z$$

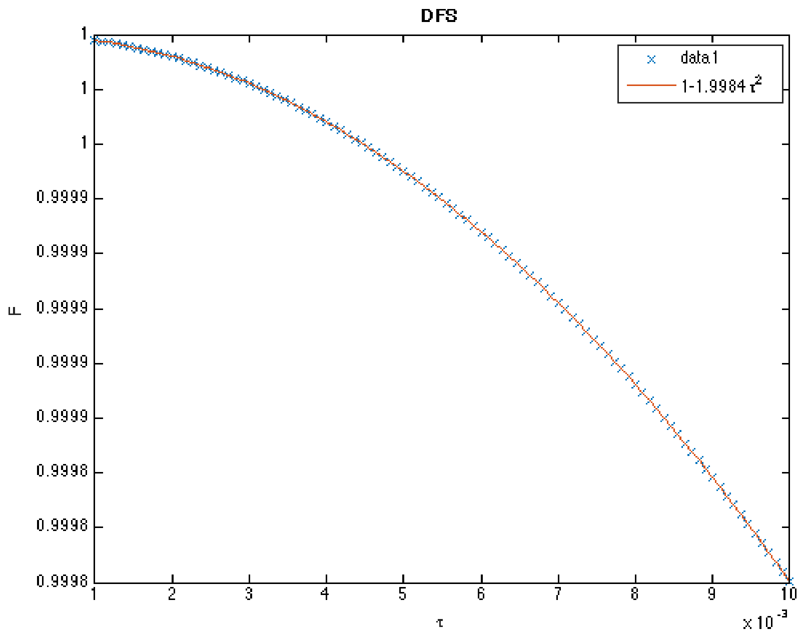
We took the initial state of the bath to be:

$$\rho_B(0) = \frac{\mathbb{1}}{2^N}$$

6.1 DFS

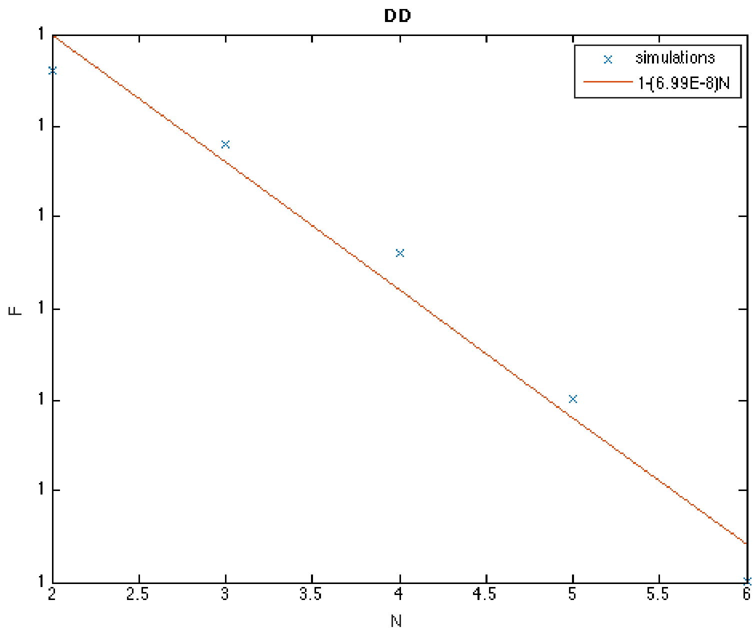
These two graphs show the dependence of the fidelity with the number of qubits and the time, respectively. Unfortunately, our computing power wasn't enough to go beyond 6 qubits, so the behavior of $\mathcal{F}$ as a function of N can't be taken as a valid result for a larger N. The behavior with time is in agreement with what we expected.

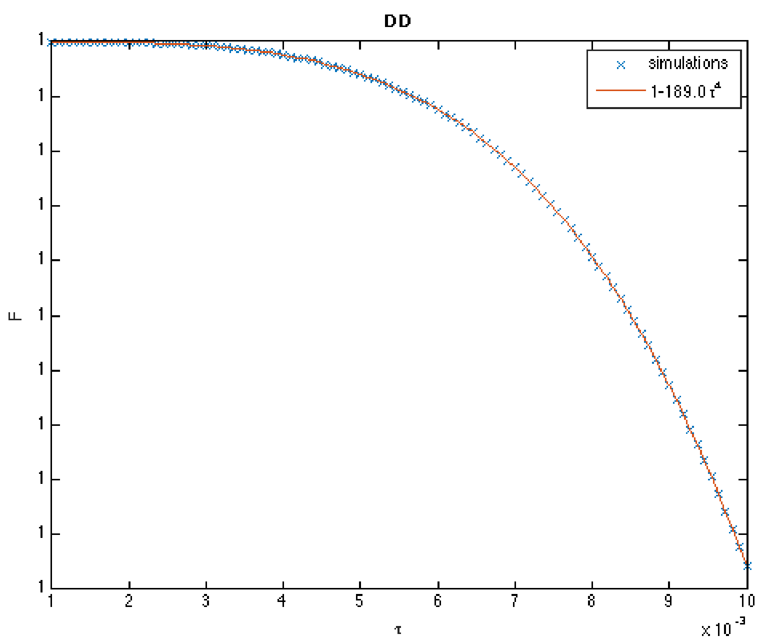




6.2 DD

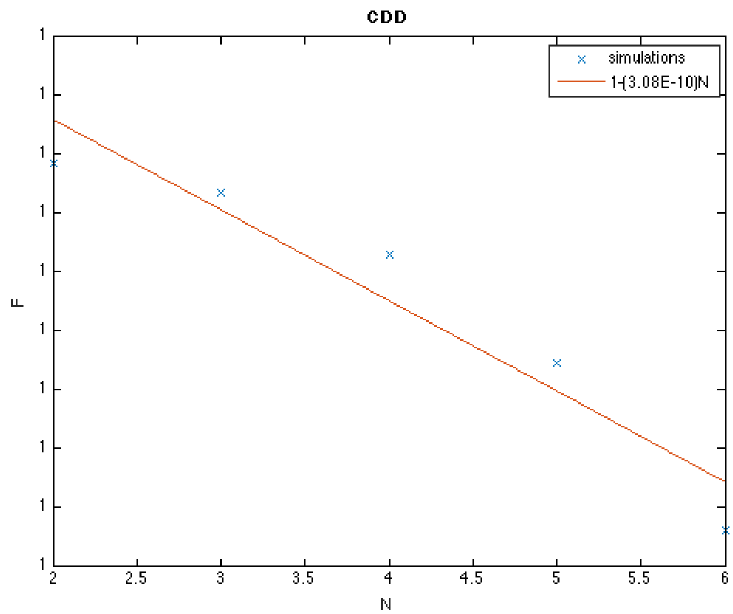
Same two graphs as for DFS. It is already clear that DD is a better method than DFS, and once more the time dependence reasonably fits our expectations.

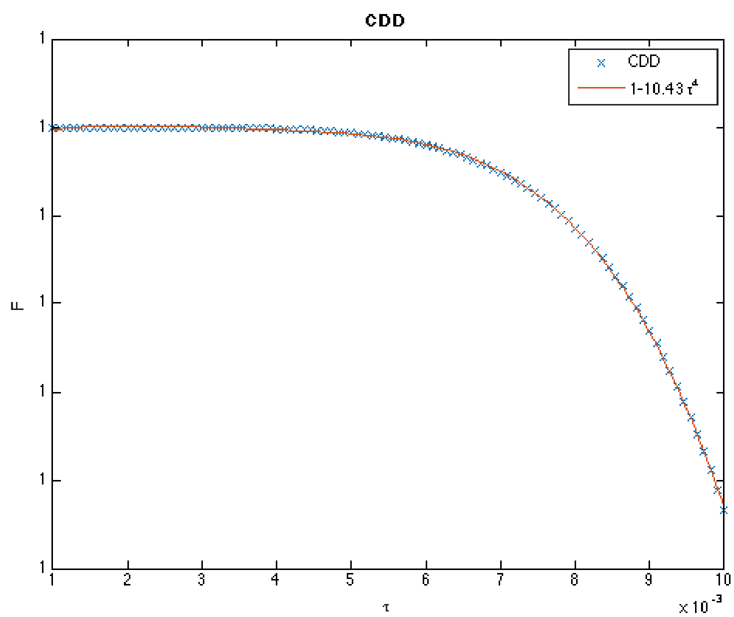

DD
simulations
1-(6.99E-8)N
F
N


DD
simulations
1-189.0 τ⁴
F
τ
x 10⁻³

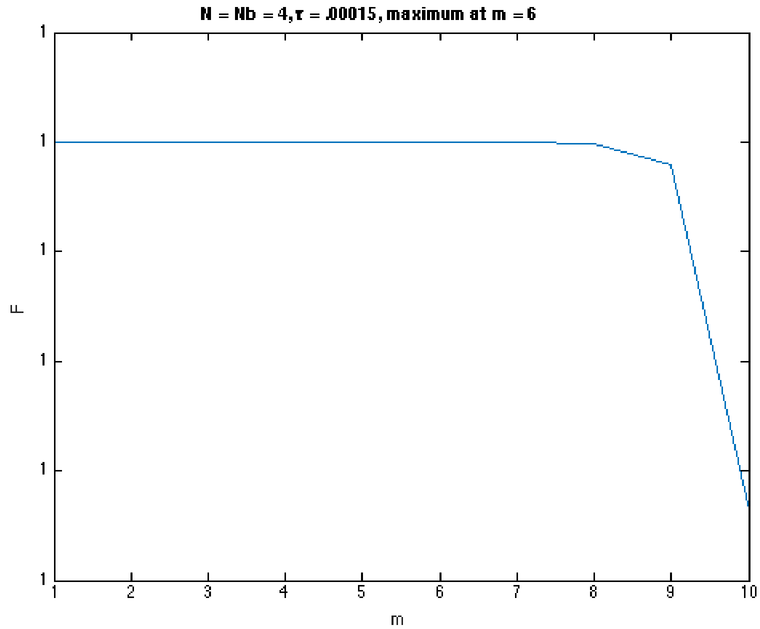
6.3 Concatenated DD

The DD method (our best candidate so far) can be improved by concatenation. The behavior with N improves by 2 orders of magnitude, and the one with time, although we keep the fourth order term, greatly decreases its constant.





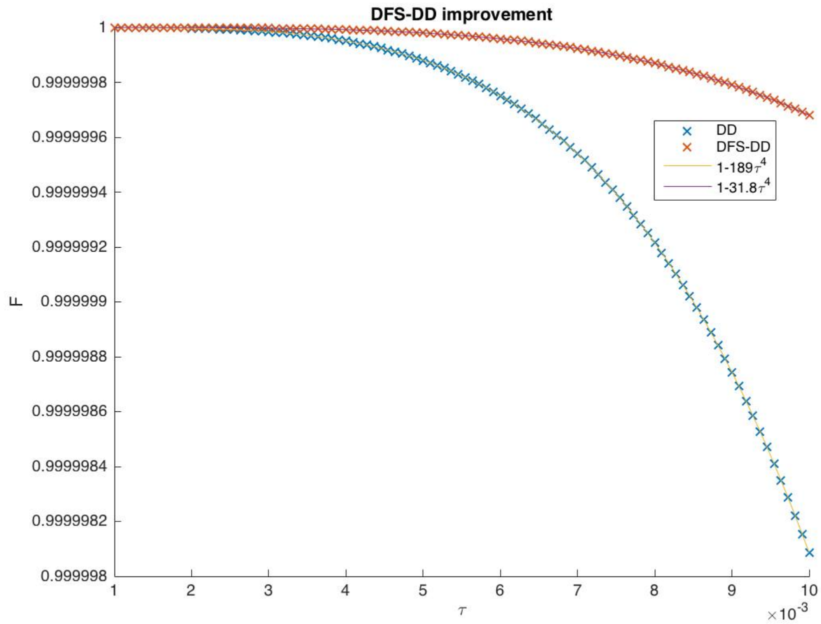
One final interesting graph for CDD is that of the fidelity vs the concatenation level since, as we saw in class, an interesting and encouraging result is that there is usually an optimal level of concatenation and going beyond it doesn't make our results better (and can actually make them worse). We clearly observe this behavior in the following graph, and find that $m_{opt} = 6$.



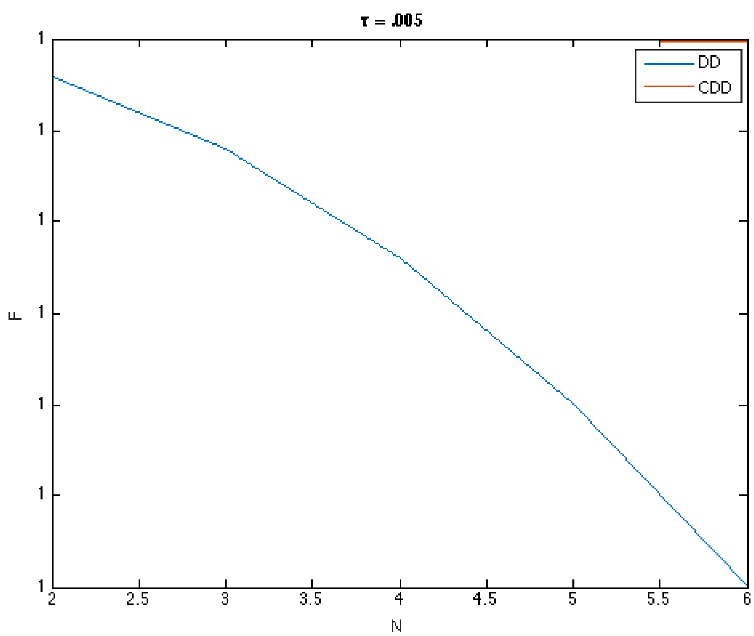
6.4 Comparisons

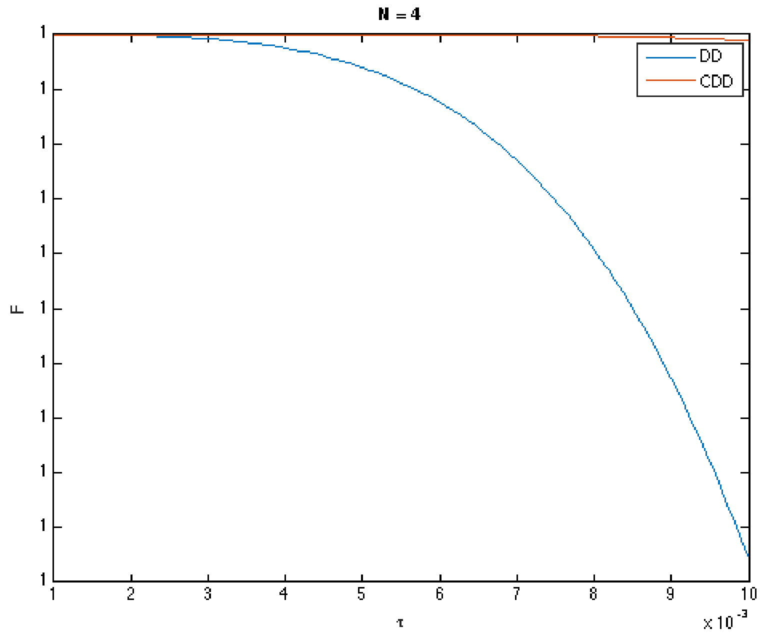
To finish our simulations section, it is interesting to plot some of our results together to better visualize how they compare.

Let's start with the hybrid DFS-DD method along DD (which is the better of those 2 when used separately). We can see that the former increases fidelity as we expected.

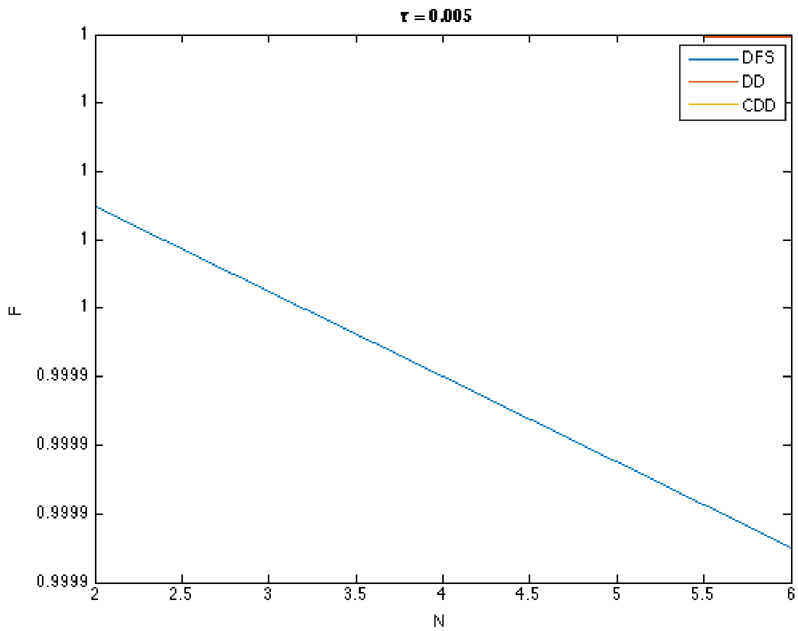


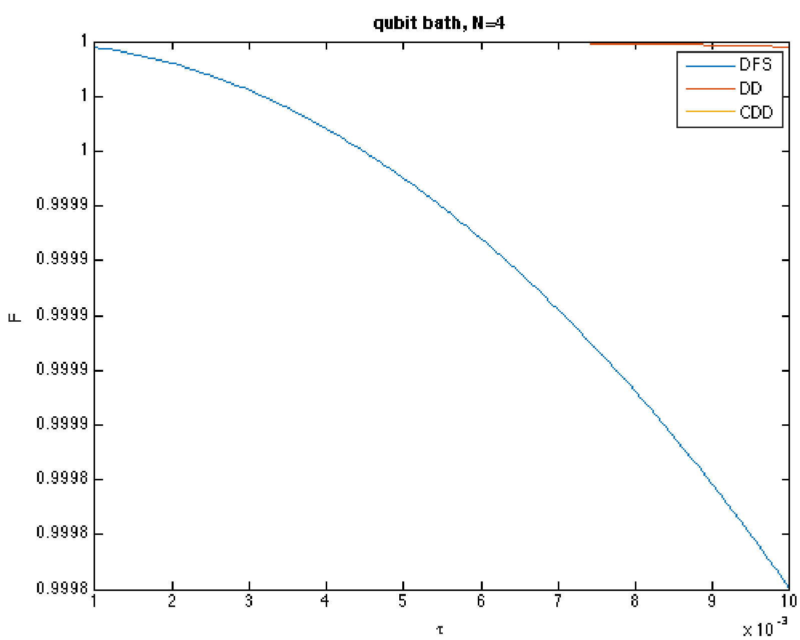
Next let's look at how CDD improves upon CC, both with N dependence and time dependence. (It might be hard to see the CDD line because it stays extremely close to 1).



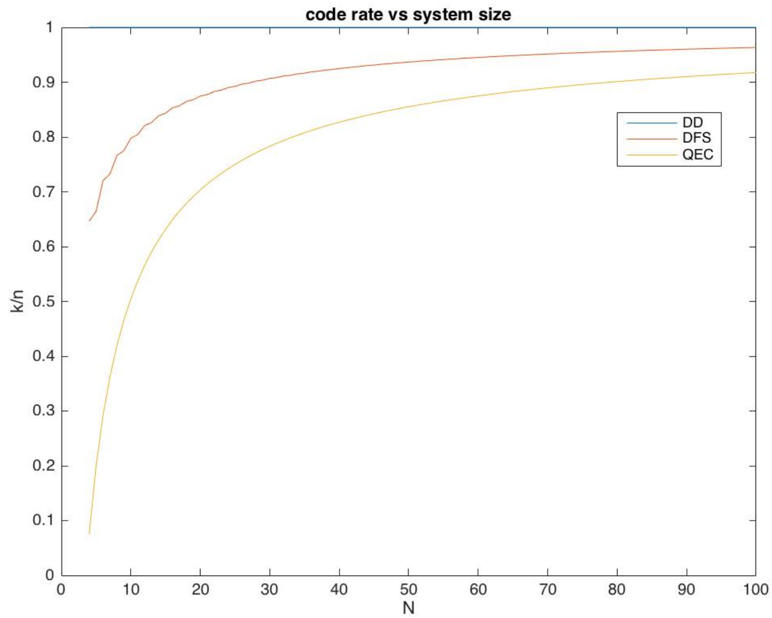


We now plot fidelity vs N and vs time for DFS, DD and CDD. The line corresponding to CDD can't be seen because at this scale it remains at 1, and the one for DD is very close too. Looking at the previous two graphs were it was evident that CDD was much better than DD, it is easy to appreciate how DFS falls very far behind the other two.





Finally we compare the rates for DD, DFS and QEC. We know an advantage of DD is that its rate is always 1. For the other two, it tends to 1 for very large N, but it might take a really high number of qubits, especially for QEC.



7 Summary and conclusions

To finish off, we feel it is useful to provide two tables summarizing our results, where it is easier to visualize how the different methods compare.

Method	$\mathcal{F}(N,\tau)$	Rate	Time resources
DFS	$1-N\tau^2\gamma^2$	$1-\frac{3}{2}\frac{log_2 N}{N}$	τ
DD	$1-2N\tau^4\lambda^2$	1	4τ
QEC	*depends*	$1-\frac{1}{N}\frac{log_2(1+3N)}{N}$	$(n-k+1)\tau$
DFS-DD	$1-\frac{N\tau^4\lambda^2}{2}$	$1-\frac{3}{2}\frac{log_2 N}{N}$	3τ
CDD		1	$4^{(m+1)}\tau$

Where $\gamma \equiv \max_i \|B_i^z\|$.

For the code rates we used information from our lectures, and considered the large N approximation. For the time resources, we assumed initial encoding through unitary gates to be instantaneous, and just counted the number of pulses, etc.

The second table compares simulations and theoretical predictions, where:

$$\tau = 0.005$$
$$N = 4$$
$$\gamma = 1$$
$$\lambda \equiv \max_i \|\{B^y, B_i^z\}\| = 2(N-1) = 6$$

Method	$\mathcal{F}(N,\tau)$	Coeff. of τ^α(sim)	Coeff. of τ^α(the)	Coeff. of N (sim)	Coeff. of N (the)
DFS	$1-\gamma^2 N\tau^2$	1.998	4	$1.25e-5$	$2.5e-5$
DD	$1-2N\tau^4\lambda^2$	189.0	288	$6.99e-8$	$4.5e-8$
DFS-DD	$1-\frac{\lambda^2}{2}N\tau^4$	31.8	72		$1.12e-8$
CDD		10.43		$3.08e-10$	

Although there are some discrepancies in the coefficients, the simulations seem to fit our theoretical predictions relatively well.

We can draw a few conclusions from our study:

- DFS is not a very good method by itself compared to others.
- DD by itself is better than DFS, but it improves when they're combined.
- CDD at the optimal concatenation level is better than any of the previous ones.
- We weren't able to quantify QEC in general terms, as its strength will mostly depend on the particular code that we use. With the current codes that we know about it doesn't seem to be the best candidate, but this could change if better codes with distance of 5 or more are found.

Sources

- Lecture notes
- *Quantum error correcting codes*, diploma thesis by Peter Majek (2005)

Appendix: MATLAB codes

7.1 DFS

```
function [rho_f,F] = DFS(rho_b,Bx,By,Bz,N,Nb,T)
%this encodes N physical qubits into N!/[(N/2)!(N/2+1)!] logical qubits
%requires N even, and only works for initial state being logical zero
%rho_f = system density matrix at time T
%F = fidelity
%T = runtime
%rho_b = bath density matrix at time 0
%Bx,By = Nb*Nb matrix
%Bz(:,:,i) = Bz,i => each an Nb*Nb matrix, i = 1 to N
%N = system size
%Nb = dimension of Hilbert space of bath

%single qubit basis state
u = [1;0];
d = [0;1];

%define singlet
s = (kron(u,d) kron(d,u))/sqrt(2);
v = 1;

%initial system state
for j = 1:N/2
    v = kron(v,s);
end

rho_s = v*v';

%define single qubit Pauli matrices
x = [0 1;1 0];
y =  [0 1i; 1i 0];
z = [1 0; 0  1];

%total initial density matrix
rho = kron(rho_s,rho_b);

Sx = zeros(2^N);
```

```
Sy = zeros(2^N);
Sz = zeros(2^N,2^N,N);
for j = 1:N
    Sz(:,:,j) = zeros(2^N);
end

for j = 1:N
    if j>1
        Id_L = eye(2^(j 1));
    else
        Id_L = 1;
    end
    Id_R = eye(2^(N j));
    Sx = Sx + kron(Id_L,kron(x,Id_R));
    Sy = Sy + kron(Id_L,kron(y,Id_R));
    Sz(:,:,j) = kron(Id_L,kron(z,Id_R));
end

%compute the SB Hamiltonian
H = kron(Sx,Bx) + kron(Sy,By);
for j = 1:N
    H = H + kron(Sz(:,:,j),Bz(:,:,j));
end

%unitary evolution
U = expm( 1i*T*H);

%find final system density matrix
rho_f = U*rho*U';
%take partial trace
rho_f = TrX23(rho_f,2,[2^N Nb]);

%find fidelity
F = sqrt(v'*rho_f*v);
```

7.2 CDD

```
function [rho_f,F,T] = CDD(rho_b,Bx,By,Bz,t,N,Nb,m)
%rho_f = system density matrix at time T
%F = fidelity
```

```
%T = total time (in units of t)
%rho_b = bath density matrix at time 0
%Bx,By = Nb*Nb matrix
%Bz(:,:,i) = Bz,i => each an Nb*Nb matrix, i = 1 to N
%t = tau (pulse interval)
%N = system size
%Nb = dimension of Hilbert space of bath
%m = concatenation level (m = 0 => ordinary DD)

%define single qubit Pauli matrices
x = [0 1;1 0];
y =  [0 1i; 1i 0];
z = [1 0; 0  1];

%initial state = tensor product of all |0> = (1,0) states
v = 1;
for j = 1:N
    v = kron([1;0],v);
end
rho_i = v*v';

%total initial density matrix
rho = kron(rho_i,rho_b);

Sx = zeros(2^N);
Sy = zeros(2^N);
Sz = zeros(2^N,2^N,N);
for j = 1:N
    Sz(:,:,j) = zeros(2^N);
end

for j = 1:N
    if j>1
        Id_L = eye(2^(j 1));
    else
        Id_L = 1;
    end
    Id_R = eye(2^(N j));
    Sx = Sx + kron(Id_L,kron(x,Id_R));
    Sy = Sy + kron(Id_L,kron(y,Id_R));
    Sz(:,:,j) = kron(Id_L,kron(z,Id_R));
```

```
end

%compute the SB Hamiltonian
H = kron(Sx,Bx) + kron(Sy,By);
for j = 1:N
    H = H + kron(Sz(:,:,j),Bz(:,:,j));
end

%free evolution for each pulse interval
U = expm( 1i*t*H);

%collective Z pulses
Zp = eye(2^N);
for j = 1:N
    Zp = Zp*Sz(:,:,j);
end
Zp = kron(Zp,eye(Nb));

%collective X pulses
Xp = eye(2^N);
for j = 1:N
    if j >1
        Id_L = eye(2^(j 1));
    else
        Id_L = 1;
    end
    Id_R = eye(2^(N j));
    Xp = Xp*kron(Id_L,kron(x,Id_R));
end
Xp = kron(Xp,eye(Nb));
T = t;

%apply pulses in each level
for k = 1:m+1
    T = 4*T;
    U = Zp*U*Zp*U;
    U = Xp*U*Xp*U;
end

%find final system density matrix
rho_f = U*rho*U';
```

```
%take partial trace
rho_f = TrX23(rho_f,2,[2^N Nb]);

%find fidelity
F = sqrt(v'*rho_f*v);
```

7.3 DFS-DD

```
function [rho_f,F] = DFSDD(rho_b,Bx,By,Bz,N,Nb,t,m)

%DFS

%single qubit basis state
u = [1;0];
d = [0;1];

%define singlet
s = (kron(u,d) kron(d,u))/sqrt(2);
v = 1;

%initial system state
for j = 1:N/2
    v = kron(v,s);
end

rho_s = v*v';

%define single qubit Pauli matrices
x = [0 1;1 0];
y =  [0 1i; 1i 0];
z = [1 0; 0  1];

%total initial density matrix
rho = kron(rho_s,rho_b);

Sx = zeros(2^N);
Sy = zeros(2^N);
Sz = zeros(2^N,2^N,N);
for j = 1:N
    Sz(:,:,j) = zeros(2^N);
```

```
end

for j = 1:N
    if j>1
        Id_L = eye(2^(j 1));
    else
        Id_L = 1;
    end
    Id_R = eye(2^(N j));
    Sx = Sx + kron(Id_L,kron(x,Id_R));
    Sy = Sy + kron(Id_L,kron(y,Id_R));
    Sz(:,:,j) = kron(Id_L,kron(z,Id_R));
end

%compute the SB Hamiltonian
H = kron(Sx,Bx) + kron(Sy,By);
for j = 1:N
    H = H + kron(Sz(:,:,j),Bz(:,:,j));
end

%unitary evolution
U = expm( 1i*t*H);

%(C)DD
%collective X pulses
Xp = eye(2^N);
for j = 1:N
    if j >1
        Id_L = eye(2^(j 1));
    else
        Id_L = 1;
    end
    Id_R = eye(2^(N j));
    Xp = Xp*kron(Id_L,kron(x,Id_R));
end
Xp = kron(Xp,eye(Nb));

%apply pulses in each concatentation level
for k = 1:m+1
    U = Xp*U*Xp*U;
end
```

```
%find final system density matrix
rho_f = U*rho*U';
%take partial trace
rho_f = TrX23(rho_f,2,[2^N Nb]);

%find fidelity
F = sqrt(v'*rho_f*v);
```