# Superconducting Circuit Quantization: A Mathematical Reference

No physics. Definitions, operators, equations, derivations.

---

## 1. The State Space

**Definition 1.1 (Charge Basis).** Let $\mathcal{H} = \ell^2(\mathbb{Z})$. The orthonormal basis $\{|n\rangle\}_{n\in\mathbb{Z}}$ is the *charge basis*. Each $|n\rangle$ is a standard basis vector. For computation, we truncate to $n \in \{-N, \ldots, N\}$, giving $\mathcal{H}_N \cong \mathbb{C}^{2N+1}$.

**Definition 1.2 (Number Operator).** $\hat{n} : \mathcal{H}_N \to \mathcal{H}_N$ is the diagonal matrix:

$$\hat{n} = \mathrm{diag}(-N, -N+1, \ldots, 0, \ldots, N-1, N)$$

**Definition 1.3 (Shift Operators).** Define $\hat{S}^+, \hat{S}^- : \mathcal{H}_N \to \mathcal{H}_N$ by:

$$\hat{S}^+|n\rangle = |n+1\rangle, \quad \hat{S}^-|n\rangle = |n-1\rangle$$

In matrix form, $\hat{S}^+$ has ones on the superdiagonal, $\hat{S}^- = (\hat{S}^+)^\dagger$ has ones on the subdiagonal. These are truncations of the bilateral shift on $\ell^2(\mathbb{Z})$.

**Definition 1.4 (Phase Operator, approximate).** Define $\hat{\varphi} : \mathcal{H}_N \to \mathcal{H}_N$ by:

$$\hat{\varphi} = \frac{1}{2i}(\hat{S}^+ - \hat{S}^-)$$

This is an antisymmetric tridiagonal matrix with $\pm\frac{1}{2i}$ on the off-diagonals.

*Remark:* This is a finite-difference approximation. On $\ell^2(\mathbb{Z})$, the exact relation is $e^{i\hat{\varphi}} = \hat{S}^+$. Taking $\sin(\hat{\varphi}) = \frac{1}{2i}(e^{i\hat{\varphi}} - e^{-i\hat{\varphi}})$ and identifying $\hat{\varphi} \approx \sin(\hat{\varphi})$ for small phase gives Definition 1.4. The approximation converges as $N \to \infty$ for states localized near $\hat{\varphi} = 0$.

**Proposition 1.5 (Commutation Relation).** On $\ell^2(\mathbb{Z})$:

$$[\hat{\varphi}, \hat{n}] \neq i \cdot I$$

The exact commutation relation $[\hat{\Phi}, \hat{Q}] = i\hbar$ holds for unbounded operators on $L^2(\mathbb{R})$. On the lattice $\ell^2(\mathbb{Z})$, we instead have $e^{i\hat{\varphi}}\hat{n} - \hat{n}e^{i\hat{\varphi}} = e^{i\hat{\varphi}}$, i.e., $\hat{S}^+$ and $\hat{n}$ satisfy $[\hat{n}, \hat{S}^+] = \hat{S}^+$. This is the relation that matters. The "canonical commutation relation" is its infinitesimal shadow.

---

## 2. The Hamiltonian

### 2.1 Parameters

All parameters are real scalars:

| Symbol | Definition | Units |
|---|---|---|
| $E_C = e^2/2C$ | Charging energy. $e = 1.602 \times 10^{-19}$ C, $C$ = total capacitance (Farads). | Joules |
| $E_J$ | Josephson energy. Given as input. | Joules |
| $n_g$ | Offset charge (dimensionless). In your code, $n_g = 0$. | — |
| $\Phi_0 = \hbar/2e$ | Reduced flux quantum. $\hbar = 1.055 \times 10^{-34}$ J·s. | Webers |
| $L$ | Inductance (Henries). | Henries |

### 2.2 Single-Node Hamiltonian (General)

For a single node connected to ground through any combination of capacitors, inductors, and Josephson junctions:

$$\hat{H} = \underbrace{4E_C(\hat{n} - n_g)^2}_{\text{Term A}} + \underbrace{\frac{\Phi_0^2}{2L}\hat{\varphi}^2}_{\text{Term B}} + \underbrace{(-E_J) \cdot \frac{1}{2}(\hat{S}^+ + \hat{S}^-)}_{\text{Term C}}$$

- **Term A** is always present (diagonal matrix).
- **Term B** is present iff there is a linear inductor. Uses $\hat{\varphi}$ from Definition 1.4.
- **Term C** is present iff there is a Josephson junction. Note $\frac{1}{2}(\hat{S}^+ + \hat{S}^-)$ is the matrix representation of $\cos(\hat{\varphi})$ on the lattice, since $\cos(\hat{\varphi}) = \frac{1}{2}(e^{i\hat{\varphi}} + e^{-i\hat{\varphi}})$ and $e^{i\hat{\varphi}} = \hat{S}^+$.

**Special cases:**

| Circuit | Term A | Term B | Term C | Name |
|---|---|---|---|---|
| Capacitor + JJ | ✓ | — | ✓ | Transmon / CPB |
| Capacitor + Inductor | ✓ | ✓ | — | LC oscillator |
| Capacitor + Inductor + JJ | ✓ | ✓ | ✓ | Fluxonium |

### 2.3 Transmon Hamiltonian (your current `_quantize`)

Setting $n_g = 0$ and $L = \infty$ (no inductor):

$$\hat{H} = 4E_C\hat{n}^2 - \frac{E_J}{2}(\hat{S}^+ + \hat{S}^-)$$

In matrix form ($d = 2N + 1$):

$$H_{ij} = \begin{cases} 4E_C \cdot i^2 & \text{if } i = j \text{ (indexing from } -N \text{ to } N) \\ -E_J/2 & \text{if } |i - j| = 1 \\ 0 & \text{otherwise} \end{cases}$$

This is a symmetric tridiagonal matrix. Your code constructs exactly this.

### 2.4 LC Oscillator Hamiltonian

Setting $E_J = 0$:

$$\hat{H} = 4E_C\hat{n}^2 + \frac{\Phi_0^2}{2L}\hat{\varphi}^2$$

where $\hat{\varphi}^2$ is the matrix square of the antisymmetric tridiagonal from Definition 1.4. The exact eigenvalues (in the limit $N \to \infty$) are $E_k = \hbar\omega(k + \frac{1}{2})$ where $\omega = 1/\sqrt{LC}$.

### 2.5 Fluxonium Hamiltonian

All three terms present:

$$\hat{H} = 4E_C\hat{n}^2 + \frac{\Phi_0^2}{2L}\hat{\varphi}^2 - \frac{E_J}{2}(\hat{S}^+ + \hat{S}^-)$$

---

## 3. Diagonalization

**Problem:** Given $\hat{H} \in \mathbb{C}^{d\times d}$ Hermitian, find eigenvalues $\{E_k\}_{k=0}^{d-1}$ (sorted ascending) and unitary $S \in \mathbb{C}^{d\times d}$ such that:

$$S^\dagger \hat{H} S = \text{diag}(E_0, E_1, \ldots, E_{d-1})$$

The columns of $S$ are the eigenvectors: $S = [|E_0\rangle \; |E_1\rangle \; \cdots \; |E_{d-1}\rangle]$.

**Definition 3.1.** The *energy eigenbasis* is $\{|E_k\rangle\}$. Each $|E_k\rangle \in \mathbb{C}^d$ is expressed in the charge basis. The component $(|E_k\rangle)_n = \langle n|E_k\rangle$ is the amplitude of charge state $|n\rangle$ in the $k$-th eigenstate.

**Definition 3.2 (Change of Basis).** For any operator $\hat{A}$ expressed in the charge basis, its representation in the energy eigenbasis is:

$$\hat{A}_{\text{energy}} = S^\dagger \hat{A} S$$

Your code: `self.n_hat_energy = S† @ self.n_hat @ S` where `S = self.states`.

**Definition 3.3 (Derived Quantities).**

- *Transition frequencies:* $f_{jk} = |E_k - E_j|/h$ where $h = 2\pi\hbar$.
- *Qubit frequency:* $f_{01} = (E_1 - E_0)/h$.
- *Anharmonicity:* $\alpha = (E_2 - E_1) - (E_1 - E_0)$. For the transmon, $\alpha < 0$.

---

## 4. Multi-Node Circuits

### 4.1 Graph Structure

**Definition 4.1.** A circuit is a weighted graph $G = (V, E)$ where:

- $V = \{0, 1, \ldots, P-1\}$ are nodes. Node 0 is ground.
- $E$ is a multiset of edges, each labeled with a type (C, L, or JJ) and parameter values.

**Definition 4.2 (Capacitance Matrix).** $\mathbf{C} \in \mathbb{R}^{P \times P}$ is defined by:

$$C_{jk} = \begin{cases} -\sum_{\text{capacitive edges } (j,k)} C_{\text{edge}} & j \neq k \\ -\sum_{m \neq j} C_{jm} & j = k \end{cases}$$

Then delete row 0 and column 0 (ground), yielding $\mathbf{C} \in \mathbb{R}^{(P-1)\times(P-1)}$.

**Definition 4.3 (Inverse Inductance Matrix).** $\mathbf{M}_L^{-1} \in \mathbb{R}^{P \times P}$ is defined by:

$$(M_L^{-1})_{jk} = \begin{cases} -\sum_{\text{inductive edges } (j,k)} 1/L_{\text{edge}} & j \neq k \\ -\sum_{m \neq j} (M_L^{-1})_{jm} & j = k \end{cases}$$

Then delete row 0 and column 0, yielding $\mathbf{M}_L^{-1} \in \mathbb{R}^{(P-1)\times(P-1)}$.

Both matrices are symmetric positive semidefinite (positive definite after symmetry breaking).

### 4.2 Classical Hamiltonian (Multi-Node)

Let $\vec{\Phi} \in \mathbb{R}^{P-1}$ be the vector of node fluxes and $\vec{Q} \in \mathbb{R}^{P-1}$ be the vector of node charges.

$$H(\vec{\Phi}, \vec{Q}) = \frac{1}{2}\vec{Q}^T \mathbf{C}^{-1} \vec{Q} + \frac{1}{2}\vec{\Phi}^T \mathbf{M}_L^{-1} \vec{\Phi} + \sum_{\text{JJ edges}} \left( -E_J \cos\left( \frac{\Phi_j - \Phi_k}{\Phi_0} \right) \right)$$

This is your `get_hamiltonian` method.

### 4.3 Normal Modes

**Proposition 4.4.** The eigenvalue problem $\mathbf{C}^{-1}\mathbf{M}_L^{-1}\vec{v} = \omega^2 \vec{v}$ gives normal mode frequencies $\omega_k$ and mode vectors $\vec{v}_k$.

*Derivation:* The Euler-Lagrange equation for the quadratic part of $H$ gives $\mathbf{C}\ddot{\vec{\Phi}} = -\mathbf{M}_L^{-1}\vec{\Phi}$. Substituting $\vec{\Phi}(t) = \vec{v}e^{i\omega t}$ yields the eigenvalue equation.

### 4.4 Multi-Node Quantization

For $P-1$ nodes, the quantum Hilbert space is $\mathcal{H} = \mathcal{H}_{N_1} \otimes \mathcal{H}_{N_2} \otimes \cdots \otimes \mathcal{H}_{N_{P-1}}$ where each factor is a truncated charge basis. The operators $\hat{n}_j$ and $\hat{S}_j^{\pm}$ act on the $j$-th factor as Identity on all others.

The full quantum Hamiltonian is:

$$\hat{H} = \sum_{j,k} 4(E_C)_{jk}(\hat{n}_j - n_{g,j})(\hat{n}_k - n_{g,k}) + \frac{\Phi_0^2}{2}\sum_{j,k}(M_L^{-1})_{jk}\hat{\varphi}_j\hat{\varphi}_k + \sum_{\text{JJ on } (j,k)} (-E_J)\cos(\hat{\varphi}_j - \hat{\varphi}_k)$$

where $(E_C)_{jk} = \frac{e^2}{2}(\mathbf{C}^{-1})_{jk}$ and $\cos(\hat{\varphi}_j - \hat{\varphi}_k) = \frac{1}{2}(\hat{S}_j^+\hat{S}_k^- + \hat{S}_j^-\hat{S}_k^+)$.

---

## 5. Time Evolution

### 5.1 Setup

Given:

- $\hat{H}_0 \in \mathbb{C}^{d\times d}$: time-independent Hamiltonian, diagonal in the energy eigenbasis. We truncate to the lowest $d$ eigenvalues: $\hat{H}_0 = \text{diag}(E_0, E_1, \ldots, E_{d-1})$.
- $\hat{V} \in \mathbb{C}^{d\times d}$: coupling operator in the energy eigenbasis ($= S^\dagger \hat{n} S$ truncated to $d \times d$).
- $A : \mathbb{R} \to \mathbb{R}$: time-dependent drive amplitude (a scalar function).
- $|\psi_0\rangle \in \mathbb{C}^d$: initial state vector, $\||\psi_0\| = 1$.

The total Hamiltonian at time $t$ is:

$$\hat{H}(t) = \hat{H}_0 + A(t)\hat{V}$$

**5.2 The Schrödinger Equation**

$$i\hbar \frac{d}{dt}|\psi(t)\rangle = \hat{H}(t)|\psi(t)\rangle, \quad |\psi(0)\rangle = |\psi_0\rangle$$

**5.3 Crank-Nicolson Discretization**

Partition $[0, T]$ into steps of size $\Delta t$. At step $i$, let $t_i = i \cdot \Delta t$ and $t_{i+1/2} = t_i + \Delta t/2$.

**Update rule:**

$$\left(I + \frac{i\Delta t}{2\hbar}\hat{H}(t_{i+1/2})\right)|\psi_{i+1}\rangle = \left(I - \frac{i\Delta t}{2\hbar}\hat{H}(t_{i+1/2})\right)|\psi_i\rangle$$

Equivalently: $|\psi_{i+1}\rangle = \mathbf{A}^{-1}\mathbf{B}|\psi_i\rangle$ where:

$$\mathbf{A} = I + \frac{i\Delta t}{2\hbar}\hat{H}(t_{i+1/2}), \quad \mathbf{B} = I - \frac{i\Delta t}{2\hbar}\hat{H}(t_{i+1/2})$$

**Proposition 5.1 (Unitarity).** If $\hat{H}$ is Hermitian, then $\mathbf{A}^{-1}\mathbf{B}$ is unitary.

*Proof:* Let $\mathbf{U} = \mathbf{A}^{-1}\mathbf{B}$. Note $\mathbf{A}^\dagger = I - \frac{i\Delta t}{2\hbar}\hat{H} = \mathbf{B}$. Then:

$$\mathbf{U}^\dagger\mathbf{U} = (\mathbf{A}^{-1}\mathbf{B})^\dagger(\mathbf{A}^{-1}\mathbf{B}) = \mathbf{B}^\dagger(\mathbf{A}^\dagger)^{-1}\mathbf{A}^{-1}\mathbf{B} = \mathbf{B}^\dagger(\mathbf{B})^{-1}\mathbf{A}^{-1}\mathbf{B} = \mathbf{A}^{-1}\mathbf{B}\cdot(\mathbf{A}^{-1}\mathbf{B})^{-1}\cdot I$$

More directly: $\mathbf{B} = \mathbf{A}^\dagger$, so $\mathbf{U} = \mathbf{A}^{-1}\mathbf{A}^\dagger$. Since $\mathbf{A}$ is normal ($\mathbf{A} = I + i\alpha\hat{H}$ for real $\alpha$ and Hermitian $\hat{H}$), write the spectral decomposition $\hat{H} = \sum_k \lambda_k |k\rangle\langle k|$. Then:

$$\mathbf{U} = \sum_k \frac{1 - i\alpha\lambda_k}{1 + i\alpha\lambda_k}|k\rangle\langle k|$$

Each eigenvalue of $\mathbf{U}$ has modulus $\left|\frac{1-i\alpha\lambda_k}{1+i\alpha\lambda_k}\right| = 1$ since $\alpha, \lambda_k \in \mathbb{R}$. $\square$

**Corollary 5.2.** $\|\psi_i\| = 1$ for all $i$. Probabilities are conserved exactly at every step.

**5.4 The SFQ Drive**

The drive amplitude is:

$$A(t) = A_0 \sum_{k=0}^{M-1} \exp\left(-\frac{(t - kT_d)^2}{2\sigma^2}\right)$$

where:

- $T_d = 1/f_d$: pulse spacing. For resonant drive, $f_d = f_{01}$.
- $\sigma$: pulse width (seconds). Typically $\sigma \ll T_d$.
- $A_0 = \epsilon \cdot (E_1 - E_0)$: amplitude, with $\epsilon \ll 1$ (e.g. $\epsilon = 0.01$).
- $M$: total number of pulses.

Each term in the sum is a narrow Gaussian centered at time $kT_d$. The sum is a pulse train.

#### 5.5 Observables

At each timestep $i$, the probability of being in the $k$-th energy eigenstate is:

$$P_k(t_i) = |(\psi_i)_k|^2 = |\langle E_k | \psi_i \rangle|^2$$

since we are working in the energy eigenbasis.

---

## 6. Gate Fidelity

#### 6.1 Propagator

**Definition 6.1.** The *propagator* $U(T) \in \mathbb{C}^{d \times d}$ is the unitary that maps initial states to final states:

$$|\psi(T)\rangle = U(T)|\psi(0)\rangle$$

To compute $U(T)$ numerically: run the Crank-Nicolson scheme $d$ times, once with each standard basis vector $|E_k\rangle$ as the initial condition. Column $k$ of $U(T)$ is the resulting $|\psi(T)\rangle$.

Equivalently:

$$U(T) = \prod_{i=0}^{N_t-1} \mathbf{A}_i^{-1} \mathbf{B}_i$$

where $\mathbf{A}_i, \mathbf{B}_i$ are from step $i$.

#### 6.2 Fidelity

**Definition 6.2.** Given target unitary $U_{\text{target}} \in \mathbb{C}^{d \times d}$ and actual propagator $U_{\text{actual}} \in \mathbb{C}^{d \times d}$, the *average gate fidelity* (restricted to the qubit subspace, $d_q = 2$) is:

$$F = \frac{1}{d_q^2} \left| \mathrm{Tr}\left( U_{\text{target}}^{[q]\dagger} \cdot U_{\text{actual}}^{[q]} \right) \right|^2$$

where $U^{[q]}$ denotes the top-left $2 \times 2$ block (the qubit subspace projection).

**Common targets:**

| Gate | $U_{\text{target}}^{[q]}$ |
|---|---|
| X (NOT) | $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ |
| $R_x(\theta)$ | $\begin{pmatrix} \cos\theta/2 & -i\sin\theta/2 \\ -i\sin\theta/2 & \cos\theta/2 \end{pmatrix}$ |
| Identity | $I_2$ |

### 6.3 Leakage

**Definition 6.3.** The *leakage* of a gate is the probability that a state starting in the qubit subspace $\mathrm{span}\{|E_0\rangle, |E_1\rangle\}$ ends outside it:

$$\mathcal{L} = 1 - \frac{1}{d_q} \sum_{j=0}^{d_q-1} \sum_{k=0}^{d_q-1} |U_{kj}|^2$$

### 6.4 Infidelity Decomposition

The total infidelity $1 - F$ can be decomposed:

$$1 - F = \mathcal{E}_{\text{coherent}} + \mathcal{E}_{\text{leakage}} + \mathcal{E}_{\text{incoherent}}$$

- $\mathcal{E}_{\text{coherent}}$: rotation angle and axis errors within the qubit subspace.
- $\mathcal{E}_{\text{leakage}}$: population leaving the qubit subspace.
- $\mathcal{E}_{\text{incoherent}}$: decoherence effects (absent in your current unitary simulation).

---

## 7. The Optimization Problem

### 7.1 Binary Formulation

An SFQ pulse sequence is a binary vector $\vec{b} \in \{0,1\}^K$ where $K$ is the number of clock ticks. At tick $j$:

$$b_j = \begin{cases} 1 & \text{apply a pulse} \\ 0 & \text{no pulse} \end{cases}$$

The drive amplitude becomes:

$$A(t) = A_0 \sum_{j : b_j = 1} \delta(t - j\tau)$$

where $\tau = 1/f_{\text{clock}}$ is the clock period, and $\delta$ is approximated by a narrow Gaussian.

### 7.2 Objective Function

$$\min_{\vec{b} \in \{0,1\}^K} \quad \mathcal{J}(\vec{b}) = (1 - F(\vec{b})) + \lambda \cdot \mathcal{L}(\vec{b})$$

where $F$ is fidelity (Definition 6.2), $\mathcal{L}$ is leakage (Definition 6.3), and $\lambda > 0$ is a penalty weight.

### 7.3 Continuous Relaxation (for gradient-based methods)

Replace $b_j \in \{0,1\}$ with $p_j \in [0,1]$:

$$A(t) = A_0 \sum_{j=0}^{K-1} p_j \cdot g(t - j\tau)$$

where $g$ is a Gaussian pulse shape. Optimize over $\vec{p} \in [0,1]^K$ using gradient-based methods (e.g. BFGS, L-BFGS-B), then snap to binary: $b_j = \text{round}(p_j)$.

Alternatively, relax the pulse *timings* rather than the amplitudes. Let $t_j \in \mathbb{R}$ be continuous pulse times, optimize with BFGS, then snap to the nearest clock tick: $t_j \to \text{round}(t_j/\tau) \cdot \tau$.

### 7.4 Gradient Computation

The gradient $\nabla_{\vec{p}} \mathcal{J}$ can be computed by:

1. **Finite differences:** $\partial \mathcal{J}/\partial p_j \approx (\mathcal{J}(\vec{p} + \epsilon \vec{e}_j) - \mathcal{J}(\vec{p} - \epsilon \vec{e}_j))/2\epsilon$. Cost: $O(K)$ propagator evaluations.

2. **Adjoint method (GRAPE-style):** Propagate forward to get all intermediate states $|\psi_i\rangle$. Propagate the "co-state" backward from the target. The gradient at step $j$ is a trace involving the forward and backward states. Cost: $O(1)$ propagator evaluations (plus storage). See Section 9.

---

## 8. Frequency Robustness (The Research Problem)

### 8.1 Problem Statement

The Hamiltonian depends on the qubit frequency $\omega_{01} = (E_1 - E_0)/\hbar$, which fluctuates:

$$\hat{H}_0(\delta) = \mathrm{diag}(E_0, E_1 + \hbar\delta, E_2 + \hbar(2\delta + \delta'), \ldots)$$

where $\delta \in \mathbb{R}$ is the frequency detuning (unknown, small) and $\delta'$ accounts for the anharmonicity shift. More precisely, let $\omega_{01}^{\mathrm{nom}}$ be the nominal qubit frequency. The actual frequency is $\omega_{01}^{\mathrm{nom}} + \delta$ where $\delta$ is drawn from some distribution (e.g. $\delta \sim \mathcal{N}(0, \sigma_\delta^2)$).

### 8.2 Robust Objective Function

$$\min_{\vec{b}} \; \mathcal{J}_{\mathrm{robust}}(\vec{b}) = \mathbb{E}_\delta\Big[\mathcal{J}(\vec{b}, \delta)\Big] \approx \frac{1}{M}\sum_{m=1}^{M} \mathcal{J}(\vec{b}, \delta_m)$$

where $\{\delta_m\}$ are sample points from the detuning distribution, and $\mathcal{J}(\vec{b}, \delta)$ is the infidelity + leakage evaluated with detuning $\delta$.

Alternatively, use a worst-case formulation:

$$\min_{\vec{b}} \; \max_{\delta \in [-\Delta, \Delta]} \; \mathcal{J}(\vec{b}, \delta)$$

### 8.3 Fidelity Susceptibility

**Definition 8.1.** The *fidelity susceptibility* is:

$$\chi_F = -\left.\frac{\partial^2 F}{\partial \delta^2}\right|_{\delta = 0}$$

A small $\chi_F$ means the fidelity is flat (robust) near the nominal frequency. This can be estimated numerically:

$$\chi_F \approx \frac{F(\delta = 0) - \frac{1}{2}[F(\delta = \epsilon) + F(\delta = -\epsilon)]}{\epsilon^2}$$

### 8.4 The Research Question

Given: a fixed circuit (transmon with known $E_C, E_J$), an SFQ clock frequency $f_{\text{clock}}$, and a detuning distribution $\delta \sim \mathcal{N}(0, \sigma_\delta^2)$.

Find: the binary sequence $\vec{b}^*$ that minimizes $\mathcal{J}_{\text{robust}}$.

Compare: $\vec{b}^*_{\text{robust}}$ (optimized for robustness) vs. $\vec{b}^*_{\text{nominal}}$ (optimized at $\delta = 0$ only).

Deliverables:

1. $F(\delta)$ curves for both sequences, showing the robust sequence has a flatter profile.
2. The cost of robustness: $F(\vec{b}^*_{\text{robust}}, \delta = 0) \leq F(\vec{b}^*_{\text{nominal}}, \delta = 0)$, i.e., you sacrifice some peak fidelity for stability.
3. Optimal $\sigma_\delta$ threshold: how much frequency uncertainty can you tolerate before fidelity drops below the fault-tolerance threshold ($F > 99.9\%$)?

---

## 9. GRAPE Algorithm (Gradient Ascent Pulse Engineering)

### 9.1 Setup

Discretize time into $K$ steps. At step $j$, the Hamiltonian is:

$$\hat{H}_j = \hat{H}_0 + u_j \hat{V}$$

where $u_j \in \mathbb{R}$ is the control amplitude (in the SFQ case, $u_j = A_0 \cdot p_j$).

The propagator for step $j$ (using Crank-Nicolson) is:

$$U_j = \left(I + \frac{i\Delta t}{2\hbar}\hat{H}_j\right)^{-1}\left(I - \frac{i\Delta t}{2\hbar}\hat{H}_j\right)$$

The total propagator is:

$$U = U_{K-1} U_{K-2} \cdots U_1 U_0$$

### 9.2 Objective

$$\Phi(\vec{u}) = \frac{1}{d_q^2}\left|\mathrm{Tr}(U_{\text{target}}^{[q]\dagger} U^{[q]})\right|^2$$

We want to maximize $\Phi$.

### 9.3 Gradient

Define forward propagators and backward propagators:

$$P_j = U_{j-1} U_{j-2} \cdots U_0 \quad (\text{forward to step } j)$$

$$Q_j = U_{K-1}^\dagger U_{K-2}^\dagger \cdots U_j^\dagger \quad (\text{backward from end to step } j)$$

Then:

$$\frac{\partial \Phi}{\partial u_j} = \frac{2}{d_q^2} \mathrm{Re}\left[ \mathrm{Tr}(U_{\text{target}}^{[q]\dagger} U^{[q]})^* \cdot \mathrm{Tr}\left( Q_{j+1}^{[q]\dagger} \frac{\partial U_j}{\partial u_j} P_j^{[q]} \right) \right]$$

where $\frac{\partial U_j}{\partial u_j}$ can be computed from the Crank-Nicolson formula:

$$\frac{\partial U_j}{\partial u_j} = -\frac{i \Delta t}{2\hbar} \left( I + \frac{i \Delta t}{2\hbar} \hat{H}_j \right)^{-1} \hat{V} \left( U_j + I \right)$$

*Proof sketch:* Differentiate $U_j = \mathbf{A}_j^{-1} \mathbf{B}_j$ with respect to $u_j$, using $\frac{d}{du}\mathbf{A}^{-1} = -\mathbf{A}^{-1}\frac{d\mathbf{A}}{du}\mathbf{A}^{-1}$ and the fact that both $\mathbf{A}_j$ and $\mathbf{B}_j$ depend linearly on $u_j$ through $\hat{H}_j$.

### 9.4 Algorithm

```
Input: initial guess ū, step size α, max iterations
1. Compute all U_j from ū
2. Compute forward propagators P_j
3. Compute backward propagators Q_j
4. For each j: compute ∂Φ/∂u_j
5. Update: ū ← ū + α · ∇Φ
6. Repeat until convergence.
```

For the SFQ binary problem, run GRAPE in the continuous relaxation, then snap to binary.

---

## 10. Open System Extension (Lindblad)

### 10.1 Density Matrix

Replace the state vector $|\psi\rangle \in \mathbb{C}^d$ with the density matrix $\rho \in \mathbb{C}^{d \times d}$, where $\rho$ is positive semidefinite and $\mathrm{Tr}(\rho) = 1$.

For a pure state: $\rho = |\psi\rangle\langle\psi|$. Mixed states (statistical mixtures) have $\mathrm{Tr}(\rho^2) < 1$.

### 10.2 Lindblad Master Equation

$$\frac{d\rho}{dt} = -\frac{i}{\hbar}[\hat{H}(t), \rho] + \sum_k \gamma_k \left( L_k \rho L_k^\dagger - \frac{1}{2}\{L_k^\dagger L_k, \rho\} \right)$$

where $[A, B] = AB - BA, \{A, B\} = AB + BA$, $L_k$ are *Lindblad operators* (matrices), and $\gamma_k > 0$ are decay rates.

### 10.3 Standard Lindblad Operators for a Qubit

| **Process** | $L_k$ | $\gamma_k$ | **Effect** |
|---|---|---|---|
| Relaxation ($T_1$ decay) | $ | E_0\rangle\langle E_1 | $ |
| Pure dephasing ($T_\phi$) | $ | E_1\rangle\langle E_1 | $ |

The total dephasing rate is $1/T_2 = 1/(2T_1) + 1/T_\phi$.

### 10.4 Vectorized Form for Numerics

Reshape $\rho \in \mathbb{C}^{d \times d}$ into $\mathrm{vec}(\rho) \in \mathbb{C}^{d^2}$ by stacking columns. The Lindblad equation becomes:

$$\frac{d}{dt}\mathrm{vec}(\rho) = \mathcal{L}(t) \cdot \mathrm{vec}(\rho)$$

where $\mathcal{L}(t) \in \mathbb{C}^{d^2 \times d^2}$ is the *Liouvillian superoperator*:

$$\mathcal{L} = -\frac{i}{\hbar}(\hat{H} \otimes I - I \otimes \hat{H}^T) + \sum_k \gamma_k \left( L_k \otimes \bar{L}_k - \frac{1}{2}(L_k^\dagger L_k) \otimes I - \frac{1}{2} I \otimes (L_k^\dagger L_k)^T \right)$$

where $\bar{L}_k$ is the element-wise complex conjugate and $\otimes$ is the Kronecker product.

This is a linear ODE in $\mathbb{C}^{d^2}$. Apply Crank-Nicolson (with $\mathcal{L}$ replacing $-i\hat{H}/\hbar$). Note: $\mathcal{L}$ is *not* anti-Hermitian, so the evolution is *not* unitary — norms decrease. This is the mathematical manifestation of

decoherence.

### 10.5 Fidelity for Open Systems

Replace the unitary fidelity with the *average gate fidelity* for quantum channels:

$$F = \frac{d_q \cdot F_{\text{ent}} + 1}{d_q + 1}$$

where $F_{\text{ent}} = \frac{1}{d_q^2} \mathrm{Tr}\left(\chi_{\text{target}}^\dagger \chi_{\text{actual}}\right)$ is the entanglement fidelity, and $\chi$ is the process matrix obtained by applying the channel to one half of a maximally entangled state.

---

## Appendix A: Notation Quick Reference

| Symbol | Meaning |
|---|---|
| $\|n\rangle$ | Charge basis state, $n$ Cooper pairs |
| $\|E_k\rangle$ | Energy eigenstate (k-th, sorted ascending) |
| $\hat{n}$ | Number operator (diagonal in charge basis) |
| $\hat{S}^{\pm}$ | Shift operators on charge basis |
| $\hat{\varphi}$ | Phase operator (antisymmetric tridiagonal) |
| $E_C$ | Charging energy $= e^2/2C$ |
| $E_J$ | Josephson energy |
| $\Phi_0$ | Reduced flux quantum $= \hbar/2e$ |
| $S$ | Eigenvector matrix (columns = eigenstates in charge basis) |
| $F$ | Gate fidelity $\in [0, 1]$ |
| $\mathcal{L}$ | Leakage, or Liouvillian (context-dependent) |
| $\vec{b}$ | Binary SFQ pulse sequence |
| $\delta$ | Frequency detuning |
| $\chi_F$ | Fidelity susceptibility |

## Appendix B: What Your Code Computes (Map)

| Code | Math |
|---|---|
| capacitance_matrix | $\mathbf{C}$ (Definition 4.2) |
| inv_inductance_matrix | $\mathbf{M}_L^{-1}$ (Definition 4.3) |
| inv_capacitance_matrix | $\mathbf{C}^{-1}$ |
| omega_squared | $\mathbf{C}^{-1}\mathbf{M}_L^{-1}$ (Proposition 4.4) |
| _quantize | Constructs $\hat{H}$ (Section 2.3) |
| _diagonalize | Solves $\hat{H} |
| _change_basis | $\hat{n}_{\text{energy}} = S^\dagger \hat{n} S$ (Definition 3.2) |
| crank_nicolson | Section 5.3, with drive from Section 5.4 |
| get_potential_energy | $U(\vec{\Phi})$ from Section 4.2, second and third terms |
| get_hamiltonian | $H(\vec{\Phi}, \vec{Q})$ (Section 4.2) |
| P_0, P_1, P_2 | $P_k(t_i) = |

## Appendix C: What You Need to Implement

For the frequency-robustness project:

1. **Propagator computation** (Definition 6.1): Modify crank_nicolson to evolve $d$ initial states and return $U(T) \in \mathbb{C}^{d \times d}$.
2. **Fidelity function** (Definition 6.2): Implement $F(\vec{b}) = \frac{1}{4}|\mathrm{Tr}(U_{\text{target}}^{[q]\dagger} U_{\text{actual}}^{[q]})|^2$.
3. **Binary pulse representation** (Section 7.1): Replace your Gaussian pulse train with a binary sequence on a clock grid.
4. **Detuning sweep**: For each candidate $\vec{b}$, compute $F(\vec{b}, \delta)$ for $\delta \in \{-\Delta, \ldots, 0, \ldots, \Delta\}$.
5. **Optimization**: Minimize $\mathcal{J}_{\text{robust}}$ (Section 8.2) using scipy.optimize.minimize with the continuous relaxation (Section 7.3).
6. **Comparison**: Plot $F(\delta)$ for robust vs. nominal sequences. Report peak fidelity, fidelity susceptibility, and robustness width.