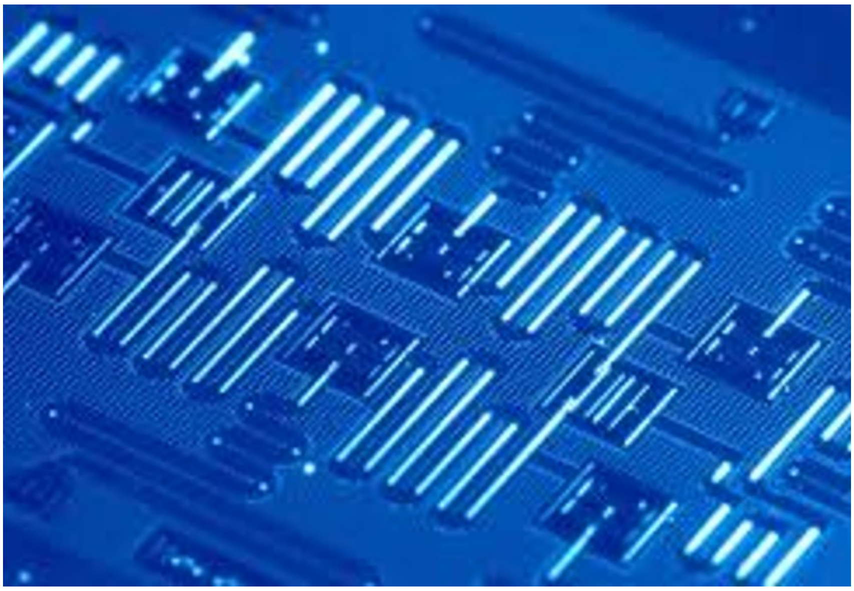
Numerical Analysis of Protected Qubits using Krylov Techniques

Eliot Delachaux[†] **Gabriel Vandersippe**[†] **Alexandru Petrescu**[*] **Emilio Rui**[‡] **Kyrylo Gerashchenko**[§]

†. Mines de Paris - ENS - PSL, {eliot.delachaux, gabriel.vandersippe}@etu.minesparis.psl.eu
∗. Inria, alexandru.petrescu@minesparis.psl.eu
‡. Inria - Alice & Bob, emilio.rui@alice-bob.com
§. Inria, kyrylo.gerashchenko@inria.fr

Keywords: Protected Qubits; Sparse Hamiltonians; Krylov Subspace Methods; Lanczos; Shift-Invert; Comparative Analysis



Source: IBM (via The Quantum Insider)

Abstract: *This report presents the numerical simulation of the KITE, a parity-protected superconducting qubit. Because its multi-mode architecture generates a massive Hilbert space, exact diagonalization becomes computationally intractable for continuous parameter sweeps. To address this, we implement and optimize a matrix-free Shift-Invert iterative algorithm. Benchmarking our method against a reference Lanczos solver demonstrates that while Lanczos remains highly efficient for computing broad lower spectra, the Shift-Invert approach is specifically advantageous for tracking individual eigenstates across external flux variations.*

Contents

1 Introduction

Quantum superconducting circuits offer a wide array of promising prospects for the future of quantum computing. However, these face a major drawback, against which researchers and engineers have been combating since their inception: decoherence. To conceive and design these circuits, the Transmon has been the industry standard for decades [2]; nevertheless, it has relatively poor resistance to the aforementioned hurdle [5]. Hence, more and more people are striving to use qubits that are inherently better at resisting decoherence and decay: the one that we will study here is the KITE (Kinetic Interference coTunneling Element), a protected qubit [4, 6]. Understanding its physical properties is a major challenge as it could lead to important breakthroughs in the field; numerical representation of the problem is thereby a prominent method to analyze this qubit, help understand measured results or perhaps even improve designs before manufacturing them (**section 2.4.2**).

However, a major computational drawback of the KITE circuit is that it operates within a Hilbert space significantly larger than that of a standard transmon (section 2.4). Consequently, computing the energy spectrum via full Exact Diagonalization is computationally prohibitive (section 3.1). Alternative numerical methods are therefore required. Fortunately, analyzing the system's physical properties does not require the entire energy spectrum; only the lowest energy levels are relevant. This allows for partial diagonalization using Krylov subspace solvers (section 3.3), which restrict the eigenvalue search to a much smaller subspace.

The standard approach for this task is the Lanczos algorithm (section 3.4). The aim of this report is to implement an alternative Krylov-based solver in Python: the Shift-Invert method (section 3.5). While Lanczos is efficient at computing a broad lower spectrum, Shift-Invert offers a more targeted approach. Therefore, it represents a potentially advantageous method for tracking a specific target eigenstate. This report details the implementation of this method and explores the tuning of its hyperparameters through numerical benchmarking (section 4). Finally, a comparative analysis between the Lanczos and Shift-Invert algorithms is presented (section 5), highlighting the specific state-tracking scenarios where Shift-Invert proves to be a more suitable choice, before summarizing our findings (section 6).

The initial subsections (section 2.1, section 2.2, and section 2.3) review the foundational quantum mechanical formalism for readers unfamiliar with superconducting circuits. More details can be found in **[1, 3]**. Readers already acquainted with these concepts may proceed directly to the presentation of the KITE architecture (section 2.4). Finally, please note that the theoretical framework presented in section 2 was co-written with G. Vandersippe, whereas the subsequent numerical implementations and benchmarks represent independent work.

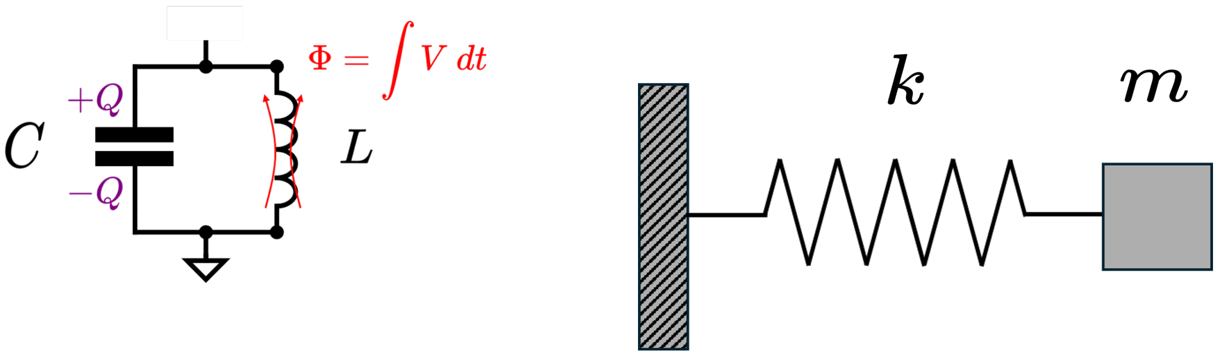
2 Physical Framework

This section establishes the foundational quantum mechanical framework for superconducting circuits. It begins by reviewing the standard harmonic oscillator (section 2.1) and the fundamental requirement of anharmonicity for qubit isolation (section 2.2), followed by the transmon architecture as the standard implementation of non-linearity (section 2.3). Readers already familiar with these concepts may proceed directly to the presentation of the parity-protected KITE circuit (section 2.4). This final subsection emphasizes the massive Hilbert space generated by the KITE architecture, which strictly motivates the numerical methods developed later in this study.

2.1 The Harmonic Oscillator

2.1.1 Harmonic Oscillator Hamiltonian

To introduce the quantum formalism, we start with the simplest possible system. It is the basis for everything else we did: the harmonic oscillator. An electronic harmonic oscillator consists of a capacitor and an inductor connected in parallel (fig. 1a), serving as the exact electrical analog to a mechanical mass-spring system (fig. 1b).



(a) Electronic Harmonic Oscillator

(b) Mechanical Harmonic Oscillator

Figure 1: Analogy between electronic and mechanical harmonic oscillators.

In classical mechanics, the energy of the mass and spring system is given by its Hamiltonian. We write it like this:

$$H = \underbrace{\frac{p^2}{2m}}_{\text{Kinetic}} + \underbrace{\frac{1}{2}kx^2}_{\text{Potential}} \tag{1}$$

We can make a direct analogy for the electronic circuit. Usually, the energy of an LC circuit is written as $H = \frac{1}{2}CV^2 + \frac{1}{2}LI^2$. But here is the problem. If we want to quantize the system, we cannot just use the voltage V and the current I. To apply canonical quantization, our variables must be canonically conjugate. This means they need to act exactly like position and momentum in Hamilton equations. Voltage and current simply do not fit this physical requirement.

Instead, we define the generalized node flux $\Phi = \int V dt$ as our position coordinate. Its corresponding conjugate momentum is the node charge $Q = CV$. By rewriting the energy with these new variables, the Hamiltonian takes the correct canonical form:

$$H = \underbrace{\frac{Q^2}{2C}}_{\text{Kinetic}} + \underbrace{\frac{\Phi^2}{2L}}_{\text{Potential}} \tag{2}$$

To quantize the circuit, we simply promote our classical conjugate variables to quantum operators. Following standard quantum formalism, they must satisfy the fundamental canonical commutation relation:

$$[\hat{\Phi}, \hat{Q}] = i\hbar \tag{3}$$

However, in superconducting circuits, we rarely work with these raw units. It is much more convenient to use dimensionless variables. We introduce the reduced flux $\hat{\varphi}$ and the reduced charge $\hat{n}$. The variable $\hat{n}$ directly represents the number of Cooper pairs (bound electron pairs carrying the lossless supercurrent). These new operators obey a much simpler commutation rule: $[\hat{\varphi}, \hat{n}] = i$.

By applying this change of variables, we rewrite our harmonic oscillator Hamiltonian. We replace the standard capacitance and inductance with the characteristic energy scales of the circuit: the charging energy E_C and the inductive energy E_L. The quantum Hamiltonian becomes:

$$H = 4E_C\hat{n}^2 + \frac{1}{2}E_L\hat{\varphi}^2 \tag{4}$$

2.1.2 From Operators to Matrices

To run numerical simulations, we must represent our quantum operators as matrices, which fundamentally requires choosing an appropriate computational basis.

For the harmonic oscillator, this choice is straightforward. The energy spectrum of such a quantum circuit is strictly discrete, with excitations quantized as individual microwave photons. To mathematically describe this system, we employ the Fock basis (or number basis), which naturally enumerates these photonic excitations. In this formalism, the vector $|0\rangle$ represents the ground state (zero photons), the state $|1\rangle$ represents the first excited state (one photon), and the general state $|n\rangle$ corresponds to an excited state containing exactly n photons.

In this basis, the quantum states are expressed as standard column vectors. The ground state $|0\rangle$ is denoted as $(1, 0, 0, \dots)^T$, and the first excited state $|1\rangle$ is defined as $(0, 1, 0, \dots)^T$.

Since the Fock basis is the eigenbasis of the harmonic oscillator, its Hamiltonian matrix is already perfectly diagonal. The energy levels are equally spaced by $\hbar\omega_r$:

$$H = \hbar\omega_r \begin{pmatrix} 1/2 & 0 & \cdots \\ 0 & 3/2 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \tag{5}$$

This simple example highlights our main issue. For the harmonic oscillator, we know the exact diagonal basis. For complex superconducting circuits, this is never true. We can build their Hamiltonian matrices by choosing a basis, but they will be massive and full of off-diagonal coupling terms. Diagonalizing these matrices numerically is the core motivation of the internship. But before going into the math, there is a fundamental physical concept we need to clarify. What exactly is a qubit?

2.2 Qubits and Anharmonicity

A qubit is essentially a two-level quantum system. We just saw that the harmonic oscillator has an infinite number of energy states. To build a qubit out of it, we could try to restrict the system to just the ground state $|0\rangle$ and the first excited state $|1\rangle$.

However, isolating a two-level system in this manner presents a fundamental physical limitation. The energy spectrum of the harmonic oscillator features strictly equidistant energy levels (fig. 2a), meaning the transition frequency ω_{01} between $|0\rangle$ and $|1\rangle$ is identical to the subsequent transition ω_{12} between $|1\rangle$ and $|2\rangle$.

To control a qubit, we send a coherent microwave pulse at the specific frequency of the transition we want to trigger. If we send a pulse at ω_{01} to flip our state from 0 to 1, the system will inevitably absorb more photons. It will climb up to state $|2\rangle$, then $|3\rangle$, and so on. We simply cannot isolate the first two levels.

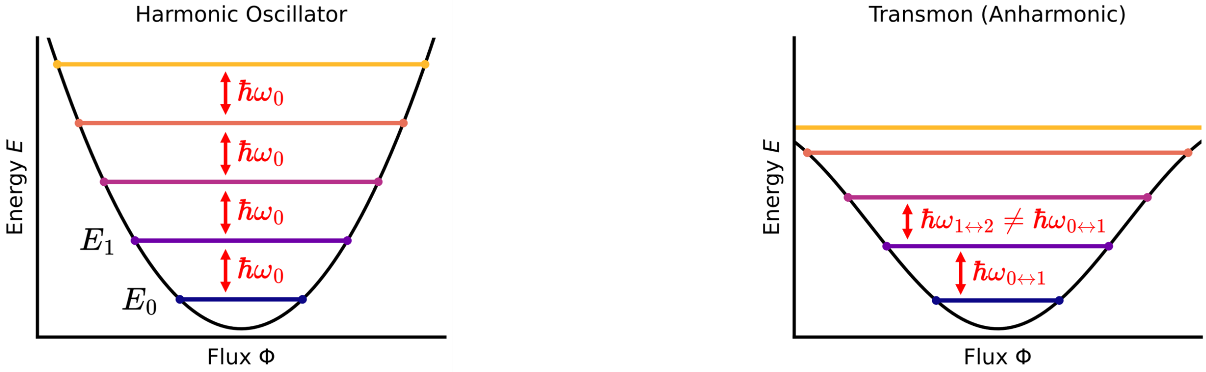
Overcoming this limitation requires introducing anharmonicity, ensuring that the transition frequency ω_{01} differs from ω_{12} (fig. 2b). Consequently, a microwave pulse tuned strictly to ω_{01} selectively drives the fundamental transition while leaving higher energy levels unpopulated.

To get this anharmonic energy spectrum, we must introduce a non-linear electrical component into our circuit. This component is the Josephson junction.

2.3 Introducing Non-Linearity: The Transmon Qubit

2.3.1 The Josephson Junction

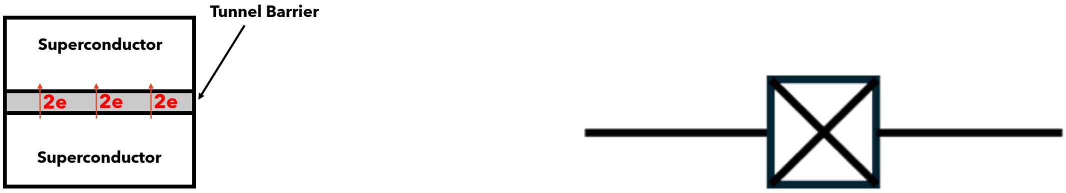
To obtain this anharmonicity, quantum circuits rely heavily on a component: the *Josephson Junction.* It consists of two superconductors (e.g. aluminum) separated by a thin tunneling barrier (e.g. aluminum oxide). It allows pairs of electrons to tunnel through this barrier, creating a nonlinear potential which is the source of the sought anharmonicity.



(a) Harmonic Oscillator energy levels

(b) Transmon energy levels

Figure 2: Comparison of energy level spacings. The harmonic oscillator has equally spaced levels. The Transmon provides the anharmonicity required to isolate a two-level system.



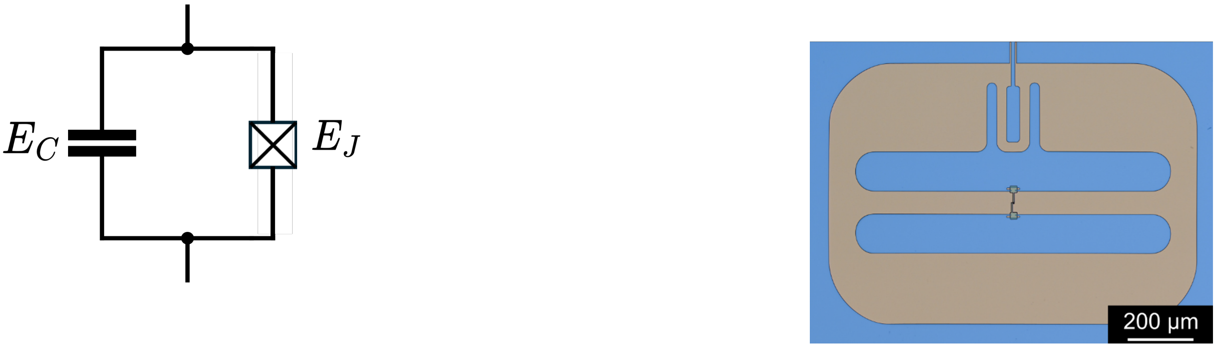
(a) Diagram of a Josephson Junction

(b) Josephson Junction Component

Figure 3: Different representations of the Josephson Junction

2.3.2 The Transmon

Before diving into the complex Kite circuit, let us briefly look at the Transmon [2]. It is the most standard superconducting qubit. As shown in fig. 4a, it simply consists of a Josephson junction in parallel with a large capacitor. A physical implementation is shown in fig. 4b, where the small contact is the Josephson junction and the large pads provide the capacitance.



(a) Diagram of a Transmon circuit

(b) Physical Transmon component

Figure 4: Different representations of the Transmon qubit. The physical layout (b) is sourced from the Houck lab (arXiv:2003.00024).

This specific design gives us the anharmonicity we need. Its Hamiltonian depends on the charging energy E_C and the Josephson energy E_J:

$$H = 4E_C\hat{n}^2 - E_J\cos(\hat{\varphi}) \tag{6}$$

In this basis, the kinetic term $4E_C\hat{n}^2$ is perfectly diagonal. The potential term $-E_J\cos(\hat{\varphi})$ acts as a shift operator. It only populates the first off-diagonals. The resulting matrix is therefore highly sparse and tridiagonal.

By computing the eigenvalues of this matrix numerically, we get the exact energy levels of the circuit. Most importantly, we recover the anharmonic spectrum we were looking for. As shown earlier in fig. 2b, the transition frequency ω_{01} is now strictly different from ω_{12}. We have successfully isolated a two-level system.

The Transmon is therefore a perfectly functional qubit. However, its simple design leaves it highly sensitive to environmental noise. This is why, for the rest of the report, we will focus on a much more complex and protected architecture: the Kite qubit.

2.4 The KITE

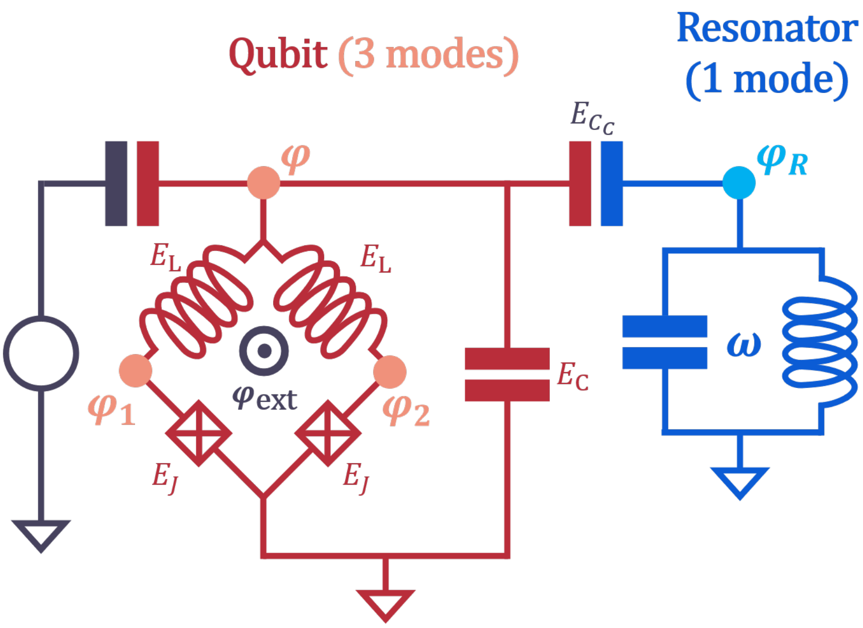


Figure 5: KITE-Readout circuit

2.4.1 The circuit

The KITE (Kinetic Interference coTunneling Element) [4] will be the subject of our study henceforth. On fig. 5, one can see that the system we will be looking into is made of a resonator, coupled to the actual qubit. The former has only one mode, associated with the flux variable φ_R, and the latter has three modes: φ, φ_1 and φ_2. As the main parameter for our study, we will rely heavily on the external flux that passes through the closed loop of the KITE, φ_{ext}, as it is much more convenient to control than changing the other parameters of the system (and its components).

The fact that we have 4 modes means that the system evolves in a truncated Hilbert space of dimension $n_1 * n_2 * n_3 * n_4$, i.e. a product of 4 integers, each corresponding to one of the modes of the system truncated to a n_i-dimensional space.

Instead of working with the aforementioned modes, we will first substitute our variables with new ones:

$$\begin{cases} \varphi \\ \varphi_\Sigma = \frac{1}{2}(\varphi_1 + \varphi_2) \\ \varphi_\Delta = \frac{1}{2}(\varphi_1 - \varphi_2) \\ \varphi_R \end{cases}$$

and the associated charges n, n_Σ, n_Δ and n_R.

2.4.2 Advantages

This circuit has interesting properties, for it has very good intrinsic protection to information loss, especially through decay. Indeed, it is an *Interference-Based cos*(2φ) *Qubit* [6], a class of parity-protected qubits. The core idea is to use the symmetry of the system to obtain a ground state and excited state with different parities, leading to suppressed matrix elements that would otherwise contribute to information loss [6].

2.4.3 Disadvantages

While the Kite circuit offers robust physical protection [5], its precise numerical simulation presents a massive challenge. The local truncation dimensions for the four constituent modes mentioned earlier are 19, 32, 32, and 5, respectively. Consequently, the tensorial product of these individual modes causes the global Hilbert space to explode, yielding a total dimension of $N_{dim} = 97,280$. As we will see next, this immense state space renders standard full-spectrum solvers completely intractable, forcing us to drastically rethink our computational approach.

3 Numerical Methods for Large Sparse Hamiltonians

The core objective of our numerical analysis is to understand how the quantum circuit responds to external control. Mathematically, we are able to construct the exact Hamiltonian matrix of the system for any given external parameter. In our study, this varying control parameter is the external magnetic flux, denoted as φ_{ext}. Our primary goal is therefore to compute and track specific eigenstates and their associated eigenvalues for this parameter-dependent Hamiltonian, $H(\varphi_{ext})$, as the external flux continuously sweeps across a given range.

For basic setups like a single Transmon, the truncated Hilbert space remains small enough that one can easily compute this entire spectral flow. However, as we transition to the complex, multi-mode architecture of the Kite qubit, the sheer scale of the system becomes the primary bottleneck.

To address this computational bottleneck, this section details the transition from full spectral evaluation to partial diagonalization methods. First, we highlight why Exact Diagonalization remains practically intractable for such massive systems, despite the inherent sparsity of the Hamiltonian (section 3.1). To demonstrate how spectral information can be extracted without full diagonalization, we introduce the power iteration method, which relies solely on sparse matrix-vector products (section 3.2). Building on this fundamental concept, we present Krylov subspace techniques (section 3.3) to efficiently target the relevant lower energy spectrum. While the standard Lanczos algorithm (section 3.4) excels at computing eigenvalues within this reduced subspace, it inherently requires calculating all preceding eigenvectors up to the targeted state n. To bypass this redundancy, we finally introduce our core methodology: the Shift-Invert algorithm (section 3.5), an advanced Krylov-based solver designed to surgically target and iteratively track specific eigenstates.

3.1 The Cost of Exact Diagonalization

The fundamental issue with Exact Diagonalization (ED) is that it computes everything unconditionally: it returns the entire spectrum of eigenvalues and their corresponding eigenvectors. While mathematically rigorous, this dense approach comes with a disastrous computational cost that scales as $\mathcal{O}(N_{dim}^3)$ in time and $\mathcal{O}(N_{dim}^2)$ in memory.

With our Kite qubit model reaching matrix sizes of nearly $100,000 \times 100,000$, attempting to run ED is practically impossible. The cubic time scaling ruins any chance of performing a continuous flux sweep within a reasonable timeframe, as the matrix would need to be entirely re-diagonalized at every single φ_{ext} step. Furthermore, merely storing a dense matrix of this magnitude in double precision requires tens of gigabytes of RAM, which easily crashes standard memory limits. Therefore, we had to completely abandon ED for the full Kite system and shift our focus toward sophisticated, highly scalable iterative methods that exploit the extreme sparsity of the quantum Hamiltonian.

3.2 Extracting Eigenspace Information: The Power Iteration

Full diagonalization is computationally expensive because it computes every single eigenspace. This is physically unnecessary. The relevant physics usually happens at the lowest energy levels. We waste massive computational resources calculating highly excited states we do not care about.

To target specific eigenvalues efficiently, we can use simple matrix-vector products. As an introduction, we present the power iteration algorithm. It mathematically demonstrates how we can extract eigenspace information without full diagonalization.

Let H be our Hamiltonian. Assume it has eigenvalues ordered by magnitude $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_N|$, with corresponding eigenvectors $|v_i\rangle$. We start with a random initial state $|x_0\rangle$. We can expand it in the eigenbasis:

$$|x_0\rangle = \sum_{i=1}^{N} c_i |v_i\rangle \tag{7}$$

If we repeatedly apply the matrix H to this state k times, we get:

$$H^k|x_0\rangle = \sum_{i=1}^{N} c_i \lambda_i^k |v_i\rangle = \lambda_1^k \left(c_1|v_1\rangle + \sum_{i=2}^{N} c_i \left(\frac{\lambda_i}{\lambda_1}\right)^k |v_i\rangle \right) \tag{8}$$

As k becomes large, the ratio $(\lambda_i/\lambda_1)^k$ strictly goes to zero. The vector aligns entirely with the dominant eigenvector $|v_1\rangle$. This proves a fundamental concept: repeated matrix-vector multiplications naturally filter out unwanted states and isolate the extreme eigenspace.

3.3 Krylov Subspace Techniques

The power iteration successfully isolates the largest absolute eigenvalue. However, in quantum mechanics, we are interested in the lowest energy values. Furthermore, we need an entire computational subspace, not just a single state. How can we use this matrix-vector product idea to compute more relevant information?

Instead of only keeping the final vector $H^k|x_0\rangle$, we keep the entire history of the sequence. This generates a Krylov subspace. Mathematically, a Krylov subspace of order k, generated by a matrix H and an initial vector v, is the linear span of these repeated products:

$$\mathcal{K}_k(H, v) = \text{span}\{v, Hv, H^2v, \ldots, H^{k-1}v\} \tag{9}$$

We know from the power iteration that the last vectors in this sequence are strongly aligned with the largest absolute eigenvalues. But because we kept the entire history, we can form linear combinations to systematically subtract this dominant information. By doing so, we reveal the next most dominant states. This iterative filtering process naturally extracts the absolute edges of the spectrum, meaning both the highest and the lowest energy states.

Since our quantum Hamiltonians are highly sparse, computing the repeated product Hv is computationally very cheap. This small subspace therefore contains extremely rich information about the extreme eigenvalues, while being computationally easy and fast to build.

3.4 The Lanczos Algorithm

To extract information from the Krylov subspace, we use the Lanczos algorithm. We construct an orthonormal basis $\{|v_1\rangle, |v_2\rangle, \ldots, |v_k\rangle\}$ for this exact subspace by iteratively applying the Hamiltonian.

Normally, Gram-Schmidt orthogonalization requires subtracting projections from all previously computed vectors. However, because H is Hermitian, this simplifies into a three-term recurrence relation. We define the next basis vector using only the two preceding it:

$$\beta_{j+1}|v_{j+1}\rangle = H|v_j\rangle - \alpha_j|v_j\rangle - \beta_j|v_{j-1}\rangle \tag{10}$$

The coefficients α_j and β_j specifically subtract the components of $H|v_j\rangle$ along the previous basis vectors $|v_j\rangle$ and $|v_{j-1}\rangle$ to ensure strict orthogonality. Mathematically, $\alpha_j = \langle v_j|H|v_j\rangle$ forms the diagonal elements of our reduced matrix. The coefficient β_{j+1} acts as the normalization factor to ensure $\langle v_{j+1}|v_{j+1}\rangle = 1$, and populates the sub-diagonals. This strict truncation keeps the memory footprint extremely low.

This process projects the massive $N \times N$ Hamiltonian down to a small $k \times k$ matrix, where $k \ll N$ is the Krylov space size. The recurrence mathematically guarantees this reduced matrix is strictly tridiagonal, populated entirely by the α and β coefficients. Diagonalizing it is virtually instantaneous. Regarding computational complexity, a standard dense matrix-vector product scales as $\mathcal{O}(N^2)$, which would normally cause the overall Lanczos algorithm to scale as $\mathcal{O}(kN^2)$. However, because our quantum Hamiltonians are highly sparse, the matrix-vector product scales almost linearly with N. As a result, the effective complexity drops to approximately $\mathcal{O}(kN)$, making the construction of the Krylov space extremely efficient.

Lanczos is highly efficient for finding the absolute bottom of the spectrum. However, if we want to compute the n-th excited state, Lanczos forces us to redundantly compute all $n-1$ lower states first. This happens because it must build the Krylov space, which inherently converges towards the most extreme eigenvalues first. Furthermore, when sweeping the external magnetic flux, standard Lanczos starts from scratch at each step, wasting the previously computed eigenspace. To bypass this redundancy and target specific states directly, we introduce the Shift-Invert strategy.

3.5 The Shift-Invert Algorithm

The Shift-Invert algorithm operates as a tracking recurrence. To apply it, we need a reliable starting point. We assume that for a given external flux φ_{ext}, we have already computed an eigenpair (for instance with the Lanczos algorithm) $\{\lambda, |\psi\rangle\}$ such that $H(\varphi_{ext})|\psi\rangle = \lambda|\psi\rangle$.

3.5.1 The Core: Spectral Transformation

When we apply a small variation $\delta\varphi$, the new Hamiltonian $H(\varphi_{ext}+\delta\varphi)$ is only slightly perturbed. Thus, the new target eigenstate $|\tilde{\psi}\rangle$ and its eigenvalue $\tilde{\lambda}$ remain close to the previous ones.

To target this specific state, we need a highly accurate guess for its new energy. A robust first-order approximation is the expectation value of the new Hamiltonian evaluated on the previous state:

$$\sigma = \langle\psi|H(\varphi_{ext} + \delta\varphi)|\psi\rangle \tag{11}$$

By subtracting this shift σ from the new Hamiltonian, the operator $(H(\varphi_{ext} + \delta\varphi) - \sigma I)$ will possess an eigenvalue $\delta\lambda = \tilde{\lambda} - \sigma$ that is extremely close to zero.

The mathematical trick is to invert this shifted matrix. The resulting inverse operator shares the exact same eigenvectors as the new Hamiltonian. However, its eigenvalues are transformed into $1/\delta\lambda$. Because $\delta\lambda \approx 0$, this specific eigenvalue becomes overwhelmingly larger than all the others $(1/\delta\lambda \gg 1)$.

By effectively turning the spectrum inside out, the interior state we wish to track instantly becomes the dominant eigenspace. Applying the power iteration to this inverse operator will therefore rapidly isolate and compute the correct new eigenstate $|\tilde{\psi}\rangle$.

The Regularization Shift: In practice, we do not strictly set σ to the exact expectation value. If this initial guess were perfect (meaning $\tilde{\lambda} = \sigma$), the shifted matrix $(H(\varphi_{ext} + \delta\varphi) - \sigma I)$ would become strictly singular. An exactly singular matrix cannot be inverted, which would immediately break the power iteration. Furthermore, the convergence speed heavily depends on the proximity of this guess to the true eigenvalue. To regularize the inversion and control the convergence rate, we introduce an additional tuning parameter called the *shift*, redefining our target energy as:

$$\sigma = \underbrace{\langle\psi|H(\varphi_{ext} + \delta\varphi)|\psi\rangle}_{\text{guess}} + \text{shift} \tag{12}$$

Summary of the Iterative Process: To summarize, our goal is to iteratively compute the sequence of vectors u_n defined by the recurrence relation:

$$u_{n+1} = (H(\varphi_{ext} + \delta\varphi) - \sigma I)^{-1} u_n \tag{13}$$

Because this sequence naturally converges toward the target eigenstate $|\tilde{\psi}\rangle$, this process corresponds exactly to the power iteration method. We initialize this recurrence with $u_0 = |\psi\rangle$. Since our previous state is already an excellent approximation of the new one $(|\psi\rangle \approx |\tilde{\psi}\rangle)$, starting this close to the true eigenvector drastically accelerates the convergence.

In practice, we cannot compute this sequence indefinitely. We truncate the loop after a fixed number of iterations, denoted as N_{power}. Because our initial guess u_0 is highly accurate and the Shift-Invert spectral gap $(1/\delta\lambda)$ is massive, the algorithm converges extremely fast. Typically, a very small N_{power} (often requiring only 2 to 5 iterations) is sufficient to reach the desired precision. We will benchmark this parameter in the next section.

3.5.2 The MINRES Algorithm and Matrix-Free Implementation

A major issue remains with the standard Shift-Invert logic: explicitly computing the inverse matrix $(H(\varphi_{ext} + \delta\varphi) - \sigma I)^{-1}$ is a computational disaster. The inverse of a sparse matrix is generally dense. Storing it would instantly crash the memory.

Instead of computing the inverse, we reformulate the power iteration step $u_{n+1} = (H(\varphi_{ext} + \delta\varphi) - \sigma I)^{-1} u_n$ as a linear system to be solved at each step:

$$(H(\varphi_{ext} + \delta\varphi) - \sigma I) u_{n+1} = u_n \tag{14}$$

To solve this equation efficiently, we use MINRES (Minimum Residual method), a highly optimized Krylov subspace solver from SciPy designed for symmetric linear systems $(Ax = y)$. The Cayley-Hamilton theorem states that the inverse A^{-1} can be expressed as a linear combination of the powers of A. Therefore, the exact solution is mathematically guaranteed to exist within the Krylov subspace, provided the space is large enough. Because MINRES only requires iterative matrix-vector products, it perfectly preserves the sparsity of our Hamiltonian and makes the computational cost of "inversion" negligible.

To formalize this efficiency, we can analyze the overall computational complexity of our matrix-free Shift-Invert approach. A direct matrix inversion would scale disastrously as $\mathcal{O}(N^3)$ and quickly exceed available memory for a massive Hilbert space of dimension N. Instead, the dominant operation within our nested iterative solver is the sparse matrix-vector product $(H - \sigma I)v$, which scales almost linearly as $\mathcal{O}(N)$ due to the high sparsity of the quantum Hamiltonian. If MINRES requires N_{iter} iterations to converge to the target tolerance ϵ, the cost of evaluating one inverse action is $\mathcal{O}(N_{iter}N)$. Multiplying this by the number of outer power iterations (N_{power}), the total time complexity per flux step scales as $\mathcal{O}(N_{power} \cdot N_{iter} \cdot N)$. Because both N_{power} and N_{iter} remain relatively small and tightly bounded across the flux sweep, our method preserves an optimal linear scaling $\mathcal{O}(N)$ while keeping the memory footprint strictly minimal.

To fully optimize the MINRES solver, we actively control two crucial hyperparameters:

- **The Tolerance (ϵ):** The algorithm allows us to define a precision threshold ϵ. The SciPy solver terminates its inner iterations once the residual norm satisfies $||Ax - y|| \leq \epsilon$. We will benchmark the impact of this parameter in the next section, but an order of magnitude of $\epsilon \approx 10^{-10}$ is typically chosen to ensure high-fidelity eigenstates.

- **The Initial Guess (x_0):** To accelerate MINRES convergence, we can provide a starting vector x_0 for the linear system $(H(\varphi_{ext} + \delta\varphi) - \sigma I)x = |\psi\rangle$. A highly efficient guess is $x_0 = -|\psi\rangle/\text{shift}$. We can mathematically justify this by assuming the flux perturbation is small, meaning $H(\varphi_{ext}+\delta\varphi) \approx H(\varphi_{ext})$. Under this approximation, we know exactly that $H(\varphi_{ext})|\psi\rangle = \lambda|\psi\rangle$. Consequently, our target energy $\sigma = \langle\psi|H(\varphi_{ext} + \delta\varphi)|\psi\rangle + \text{shift}$ simplifies to $\sigma \approx \lambda + \text{shift}$. Applying the shifted operator to our guess yields:

$$\begin{aligned}(H(\varphi_{ext} + \delta\varphi) - \sigma I)\left(-\frac{|\psi\rangle}{\text{shift}}\right) &\approx (H(\varphi_{ext}) - (\lambda + \text{shift})I)\left(-\frac{|\psi\rangle}{\text{shift}}\right) \\ &= -\frac{1}{\text{shift}}\left(\underbrace{H(\varphi_{ext})|\psi\rangle - \lambda|\psi\rangle}_{=0} - \text{shift}|\psi\rangle\right) \\ &= |\psi\rangle \end{aligned} \tag{15}$$

  Since the result naturally simplifies to our exact target right-hand side $|\psi\rangle$, this initial guess provides MINRES with an exceptionally close starting point. Crucially, the validity of this entire derivation relies on the $H(\varphi_{ext} + \delta\varphi) \approx H(\varphi_{ext})$ approximation. Therefore, the efficiency of the algorithm is deeply tied to the flux step size: the smaller $\delta\varphi$, the closer the initial guess is to the true solution, which drastically minimizes the number of iterations required for MINRES to converge.

The Trick for Complex Hamiltonians: A final technical issue arises in our Shift-Invert implementation. Unfortunately, the `minres` solver from SciPy does not natively support complex numbers. To bypass this limitation, we exploit the natural isomorphism between C^n and R^{2n}. This implies that a complex vector of size n can be encoded without loss of information into a real vector of size $2n$, and the same principle applies to matrices.

For a given complex vector $x = \text{Re}(x) + i\text{Im}(x) \in C^n$ and a complex matrix $H = \text{Re}(H) + i\text{Im}(H) \in C^{n\times n}$, the transformation to their extended real forms $x^* \in R^{2n}$ and $H^* \in R^{2n\times 2n}$ is explicitly given by the block structure:

$$x^* = \begin{pmatrix}\text{Re}(x) \\ \text{Im}(x)\end{pmatrix}, \quad H^* = \begin{pmatrix}\text{Re}(H) & -\text{Im}(H) \\ \text{Im}(H) & \text{Re}(H)\end{pmatrix} \tag{16}$$

By separating the real and imaginary parts, the complex matrix-vector product Hx is perfectly mapped to the real-valued product H^*x^*. This mathematical trick allows us to use SciPy's highly optimized MINRES solver directly without having to re-implement the entire algorithm from scratch. However, because this approach artificially doubles the dimension of the system, it inherently increases the computational overhead. For maximum efficiency, natively recoding MINRES to support complex arithmetic would be the most optimal long-term solution.

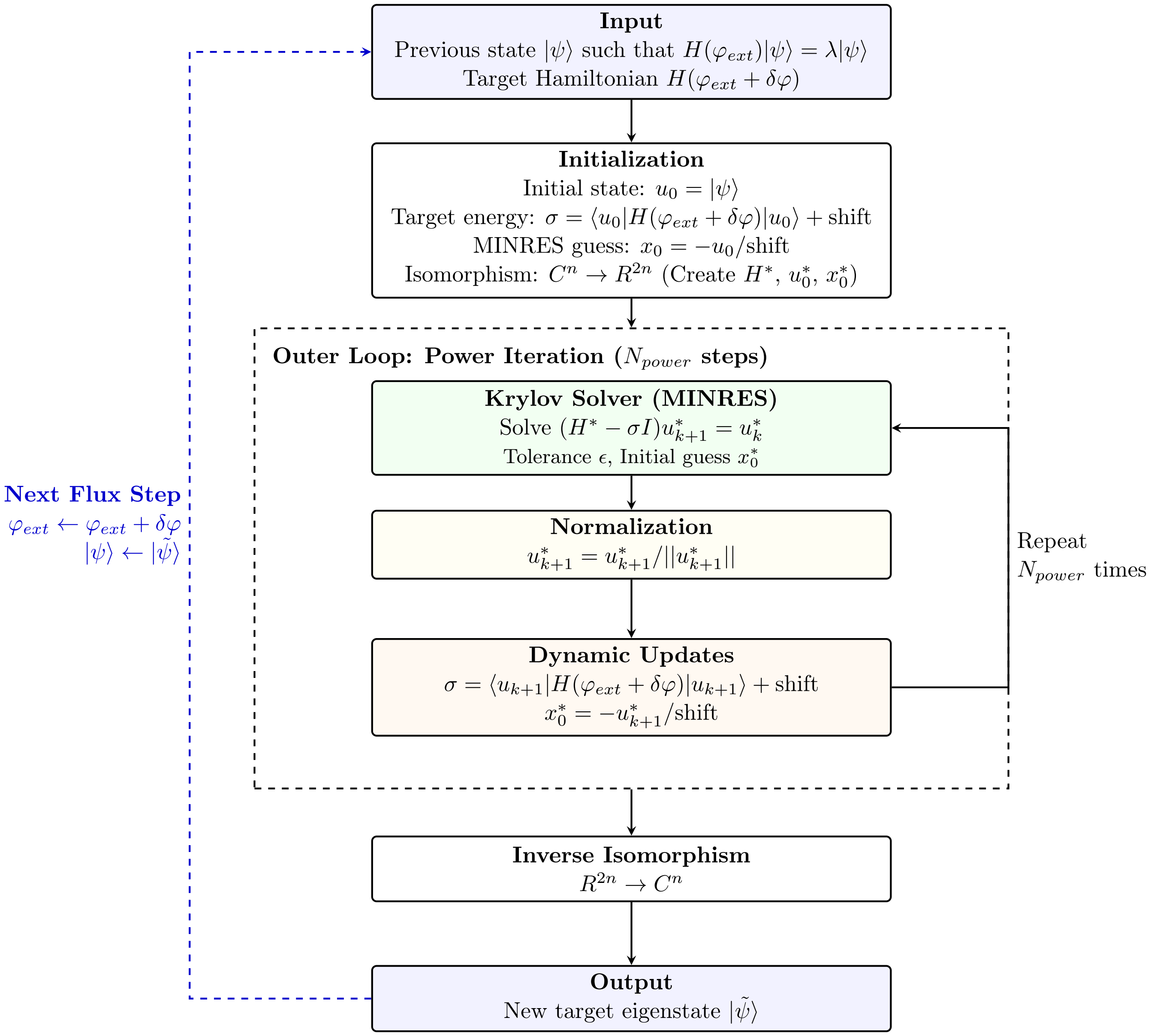

Input
Previous state $|\psi\rangle$ such that $H(\varphi_{ext})|\psi\rangle = \lambda|\psi\rangle$
Target Hamiltonian $H(\varphi_{ext} + \delta\varphi)$
Initialization
Initial state: $u_0 = |\psi\rangle$
Target energy: $\sigma = \langle u_0|H(\varphi_{ext} + \delta\varphi)|u_0\rangle + \text{shift}$
MINRES guess: $x_0 = -u_0/\text{shift}$
Isomorphism: $C^n \to R^{2n}$ (Create H^*, u_0^*, x_0^*)
Outer Loop: Power Iteration (N_{power} steps)
Krylov Solver (MINRES)
Solve $(H^* - \sigma I)u_{k+1}^* = u_k^*$
Tolerance ϵ, Initial guess x_0^*
Normalization
$u_{k+1}^* = u_{k+1}^*/||u_{k+1}^*||$
Dynamic Updates
$\sigma = \langle u_{k+1}|H(\varphi_{ext} + \delta\varphi)|u_{k+1}\rangle + \text{shift}$
$x_0^* = -u_{k+1}^*/\text{shift}$
Repeat
N_{power} times
Next Flux Step
$\varphi_{ext} \leftarrow \varphi_{ext} + \delta\varphi$
$|\psi\rangle \leftarrow |\tilde{\psi}\rangle$
Inverse Isomorphism
$R^{2n} \to C^n$
Output
New target eigenstate $|\tilde{\psi}\rangle$

Figure 6: Flowchart of the matrix-free Shift-Invert tracking algorithm. The core of the method relies on an outer Power Iteration loop, embedding an inner MINRES solver. The entire process is embedded within a global parameter sweep (dashed blue loop) to track the eigenspace across varying external fluxes.

4 Benchmarking shift invert parameters

To ensure the numerical stability, accuracy, and efficiency of our matrix-free Shift-Invert algorithm, it is essential to systematically optimize its core hyperparameters. As established in the theoretical framework, the performance of our solver relies fundamentally on three tunable variables:

- **The regularization shift (`shift`):** which offsets the targeted energy guess to prevent matrix singularities while maintaining a massive spectral gap for the inversion.
- **The MINRES tolerance (ϵ):** which dictates the convergence threshold and the numerical pre-

cision of the inner Krylov solver.

- **The number of power iterations (N_{power}):** which defines how many times the outer Shift-Invert loop is executed to isolate the exact target eigenstate.

To systematically identify the optimal trade-off between computational speed and physical accuracy, this section benchmarks each of these parameters. First, quantitative metrics (variance and overlap) are established to evaluate the solver's performance (section 4.1). Relying on these metrics, the regularization shift is independently tuned to prevent both numerical instability and spectral confusion (section 4.2). Finally, the inner MINRES tolerance and the outer power iterations are concurrently optimized (section 4.3). The resulting optimal configuration serves as the strict foundation for the global comparative analysis against the Lanczos algorithm presented in the subsequent section.

4.1 Metrics for Numerical Accuracy

To benchmark the validity of the computed eigenstate $|\tilde{\psi}\rangle$, we rely on two complementary quantitative metrics: the energy variance and the state overlap.

Energy Variance: The most robust, basis-independent measure of how well a state solves the Schrödinger equation is its energy variance. For a perfectly exact eigenstate of the Hamiltonian H, the variance is strictly zero. For our numerically approximated state $|\tilde{\psi}\rangle$ with an expected energy $E = \langle\tilde{\psi}|H|\tilde{\psi}\rangle$, the variance is defined as:

$$\mathrm{Var}(E) = \langle\tilde{\psi}|H^2|\tilde{\psi}\rangle - \langle\tilde{\psi}|H|\tilde{\psi}\rangle^2 = ||(H - EI)|\tilde{\psi}\rangle||^2 \tag{17}$$

Monitoring the decay of this variance allows us to strictly quantify the absolute error of our solver without requiring a known analytical solution.

State Overlap: While the variance checks if the algorithm successfully converged to a valid mathematical eigenstate, the state overlap ensures we are tracking the *correct* continuous physical branch across the parameter sweep. We define the overlap simply as the absolute value of the inner product between the state at the previous flux step and the newly computed state:

$$\mathcal{O} = |\langle\psi(\varphi_{ext})|\psi(\varphi_{ext} + \delta\varphi)\rangle| \tag{18}$$

Because the state is expected to evolve continuously under a small perturbation $\delta\varphi$, this overlap should remain close to 1.

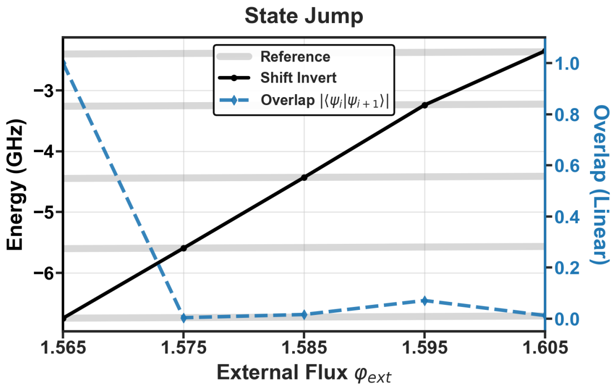


Figure 7: Evolution of the computed eigenenergy (black solid line) and state overlap (blue dashed line) during a parameter sweep of the external flux φ_{ext}. The thick grey horizontal lines denote the exact energy branches of the system.

However, iterative solvers are susceptible to converging to a valid, yet unintended, eigenstate (fig. 7). In this example, the algorithm fails to track the target continuous branch (lowest grey line) and erroneously jumps to higher energy levels at each flux step. This is where the overlap metric becomes indispensable: instead of remaining close to 1 as expected for a continuous physical evolution, the overlap $\mathcal{O}$ (blue dashed line) collapses to near zero, instantly flagging the tracking failure.

This type of instability typically occurs when the regularization shift is poorly tuned, causing the algorithm to lock onto the wrong spectral target. Let us now benchmark this parameter to determine its optimal value and prevent such "state jumps".

4.2 The Shift Parameter

Recall that our iterative algorithm relies on a dynamic target energy defined as $\sigma = \langle u_0|H(\varphi_{ext}+\delta\varphi)|u_0\rangle +$ shift. To systematically benchmark the optimal value for this `shift` parameter, we simulated a highly demanding scenario: tracking the lowest continuous energy branch through a sharp avoided crossing over 20 discrete flux steps (fig. 8). For this test, the other hyperparameters were kept strictly fixed at $N_{power} = 10$ and $\epsilon = 10^{-10}$.

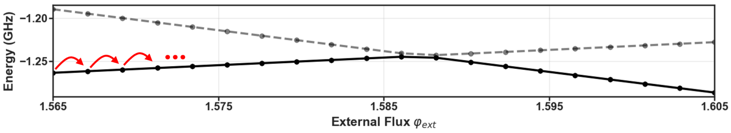


Figure 8: Trajectory of the benchmarked spectral tracking. The algorithm attempts to follow the lower continuous energy branch (solid line) through a strongly avoided crossing as the external flux φ_{ext} varies.

To evaluate the solver's performance, the maximum energy variance, minimum state overlap, and average computational time per flux step were recorded across the trajectory for `shift` values ranging from 10^{-14} to 10^{-1} GHz (fig. 9).

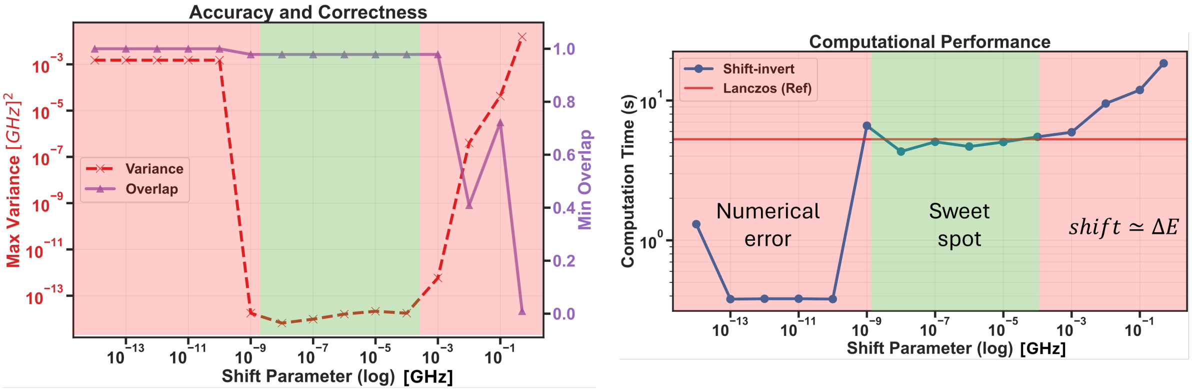


Figure 9: Benchmarking the `shift` parameter. **Left:** Accuracy metrics, showing the maximum variance (red dashed line, log scale) and minimum state overlap (purple solid line, linear scale). **Right:** Computational time per flux step in seconds (blue solid line) compared to the standard Lanczos method (red solid line). Three distinct regimes clearly emerge: numerical instability (red zone, left), the optimal sweet spot (green zone, center), and spectral confusion (red zone, right).

The algorithm exhibits three distinct functional regimes (fig. 9):

1. **The Numerical Error Zone (shift $< 10^{-9}$ GHz):** In this extreme regime, the algorithm appears deceptively fast, but the variance skyrockets to 10^{-3} GHz^2. This occurs because we compute the initial MINRES guess as $x_0 = -u_0/\text{shift}$. Dividing by such a microscopic value produces a massive

initial vector x_0 that triggers severe floating-point arithmetic errors. The numerical evaluation of the initial residual breaks down at the machine precision level, causing MINRES to falsely assume it has converged at the first iteration and abort the process without properly updating the state.

2. **The Spectral Confusion Zone (shift $> 10^{-4}$ GHz):** When the shift becomes arbitrarily large, it approaches the scale of the spectral gap ΔE between neighboring energy levels. The Shift-Invert transformation loses its resolving power, and the Krylov solver struggles to differentiate the target state from its neighbors. Consequently, the computational time increases drastically (exceeding the dense Lanczos reference), the tracking accuracy drops, and the overlap falls to zero, indicating catastrophic "state jumps."

3. **The Sweet Spot ($10^{-9} <$ shift $< 10^{-4}$ GHz):** Between these two boundaries lies a highly stable operational plateau. The variance hits machine-precision zero ($\sim 10^{-14}$ GHz^2), the overlap remains perfectly at 1.0, and the computation time per step is minimized and stable.

Based on this analysis, we select an optimal value of **shift = $\mathbf{5 \times 10^{-8}}$ GHz** for all subsequent computations. We deliberately choose a value closer to the lower numerical boundary rather than centering it in the sweet spot. This is driven by a critical physical consideration: as we probe higher energy manifolds of the Kite qubit, the spectral density increases and the energy gaps (ΔE) inherently shrink. The upper boundary of our sweet spot is therefore physically fragile, whereas the lower boundary is dictated purely by fixed machine precision. Choosing a smaller shift provides a much safer margin to track highly excited states without suffering from spectral confusion.

4.3 Power Iterations (N_{power}) and Tolerance (ϵ)

The remaining two hyperparameters—the inner MINRES solver precision (ϵ) and the number of iterations in the power algorithm (N_{power})—were simultaneously benchmarked to determine the optimal convergence configuration. The avoided-crossing tracking scenario (fig. 8) was utilized, with the regularization shift fixed at 5×10^{-8} GHz.

Performance metrics included the maximum state variance and the maximum energy deviation relative to a dense Lanczos reference ($||E - E_{ref}||_\infty$). The state overlap metric was omitted, as the extremely small energy deviations from reference inherently guarantee tracking of the correct physical branch without state jumps.

Numerical accuracy improves when tightening the tolerance ϵ and increasing N_{power} (fig. 10). Conversely, the computational cost monotonically increases due to the additional matrix-vector operations required by stricter convergence criteria (fig. 11).

However, a clear plateau effect emerges in the accuracy metrics. For $N_{power} \geq 3$, the error stabilizes; performing further power iterations yields negligible improvements while continuing to inflate the computation time. Furthermore, focusing on the tolerance parameter, we observe that for $\epsilon \leq 10^{-9}$, the energy deviation hits a hard floor at approximately 10^{-14} GHz. A closer inspection of this regime reveals that for $\epsilon \leq 10^{-11}$, the precision of the deviation remains strictly identical. As discussed in the previous section, this boundary corresponds exactly to the machine precision limit for double-precision floating-point arithmetic, meaning that enforcing a stricter tolerance provides no tangible physical benefit.

Based on these observations, we can establish the optimal compromise between supreme accuracy and computational efficiency. We select $\epsilon = 10^{-12}$ to operate safely near the machine-precision limit, and we fix $N_{power} = 4$. While $N_{power} = 3$ is theoretically sufficient to reach the plateau, adding one extra iteration provides a necessary safety margin to ensure robust convergence even in highly degenerate or numerically stiff regions of the parameter space.

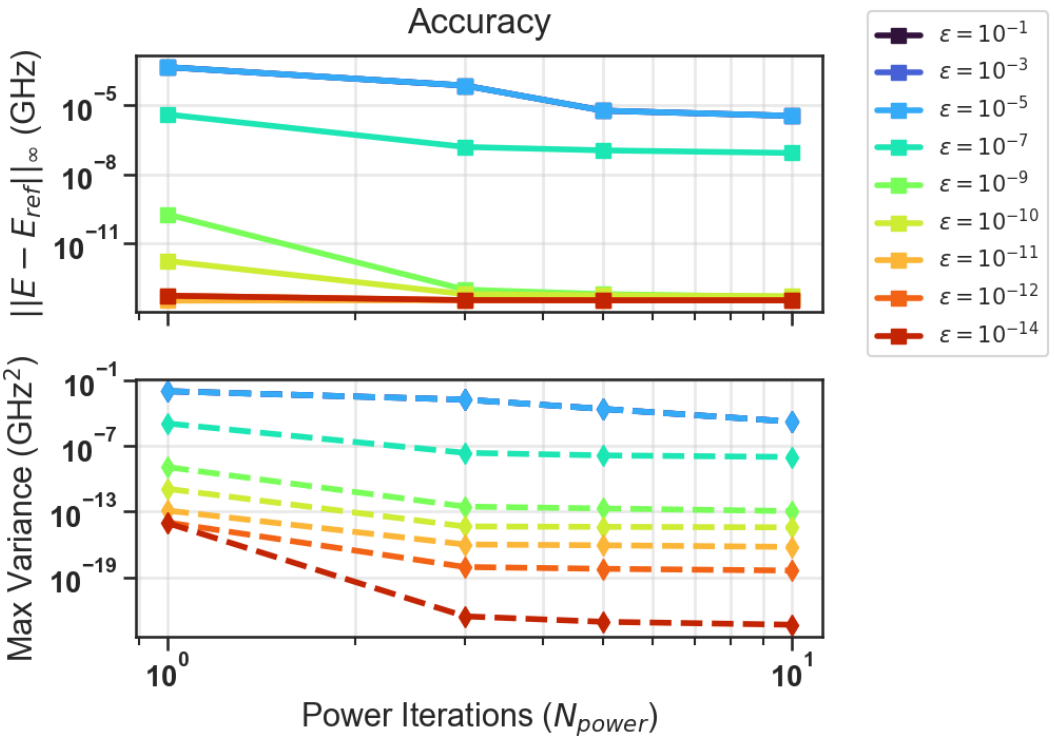


Figure 10: Accuracy benchmark of the Shift-Invert algorithm. **Top:** Maximum energy deviation from the Lanczos reference. **Bottom:** Maximum Hamiltonian variance. Both metrics are evaluated across the trajectory for various MINRES tolerances (ϵ) as a function of the number of power iterations (N_{power}).

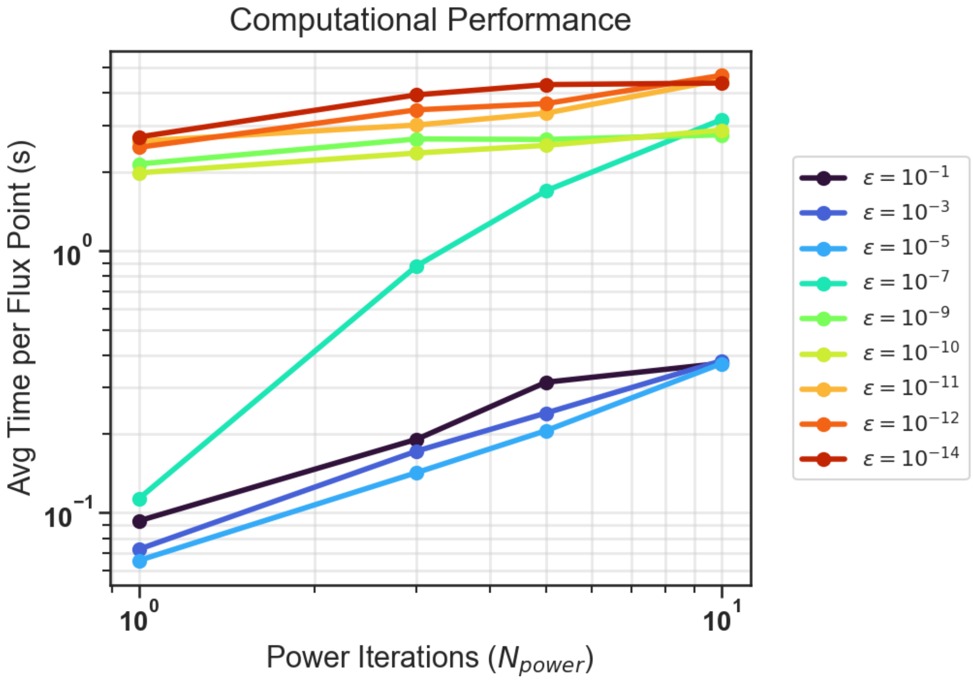


Figure 11: Computational performance benchmark. The average time per flux step is plotted against the number of power iterations (N_{power}) for different MINRES tolerances (ϵ).

Having systematically tuned the regularization `shift`, the tolerance ϵ, and the iterations N_{power}, our matrix-free Shift-Invert solver is now fully optimized. We are finally in a position to benchmark its global performance and scalability against standard exact diagonalization techniques.

5 Lanczos versus Shift-Invert: Final Benchmark

To evaluate the relative performance of our Shift-Invert algorithm against the standard Lanczos method, we devised a straightforward tracking benchmark. For a given target eigenstate of index n, we first computed a reference state at an initial flux φ_{ref} using the Lanczos method. We then attempted to compute the subsequent state at a forward flux step $\varphi_{ref} + \Delta\varphi$ using both algorithms. As detailed below, this benchmark demonstrates that while Lanczos remains a robust baseline, Shift-Invert reduces computation time for small flux steps when tracking a single state. Finally, it is important to acknowledge that the numerical routines used to generate the sparse KITE Hamiltonian, alongside the baseline Lanczos implementation, were part of a codebase provided by our supervisor, K. Gerashchenko.

It is crucial to note a fundamental difference in how these two solvers operate. Our Shift-Invert method targets and extracts a single specific eigenstate near a dynamically updated energy guess. In contrast, the Lanczos algorithm is a broader spectrum iterative method; to retrieve the n-th state, it must explicitly compute all the lower eigenstates in the range $[0, n]$.

Both the target index n and the step size $\Delta\varphi$ were varied to evaluate the solvers. Tracking higher eigenstates is inherently more challenging: the Lanczos solver must resolve a much larger lower subspace (all states from 0 to n), whereas for Shift-Invert, the increasing spectral density complicates target isolation. The methods were compared across computational time, Hamiltonian variance, and state overlap (fig. 12).

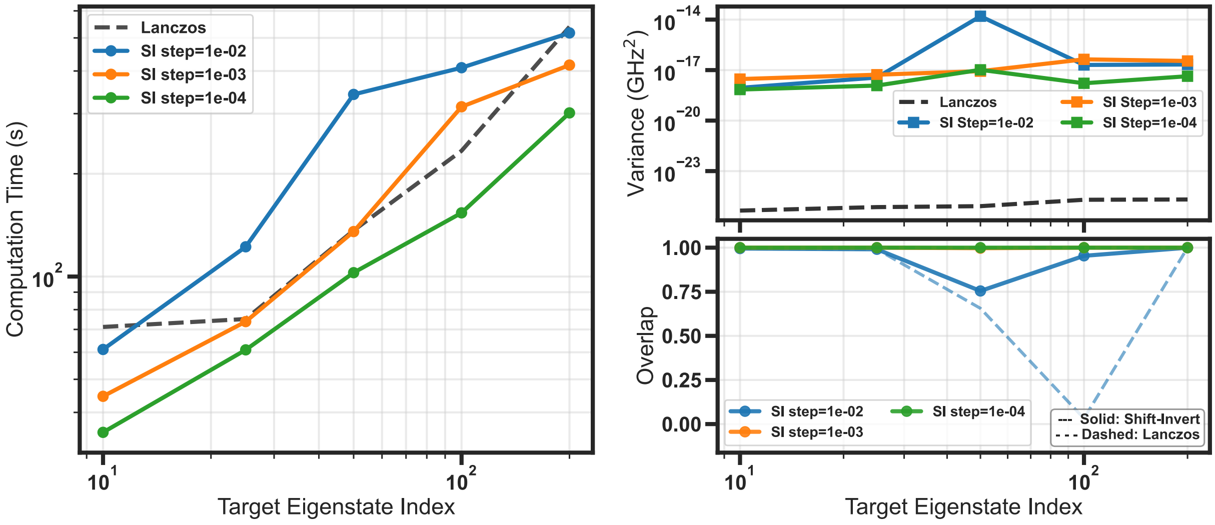


Figure 12: Performance comparison between Lanczos and Shift-Invert. **Left:** Computational time as a function of the eigenstate index n for different flux steps $\Delta\varphi$. **Right:** Tracking accuracy (variance) and state overlap. The solid lines represent the Shift-Invert method, while the dashed lines represent the Lanczos reference.

The computation time increases with the eigenstate index n for both methods due to the rising density of states (fig. 12, Left). However, Shift-Invert's performance is highly sensitive to the step size $\Delta\varphi$: a smaller flux step provides an accurate initial guess x_0, enabling rapid convergence, whereas a larger step yields a poor guess and forces the Krylov solver to perform more iterations. For a small step of $\Delta\varphi = 10^{-4}$, the Shift-Invert algorithm is twice as fast as the standard Lanczos solver.

The Lanczos method, computing the entire $[0, n]$ subspace from scratch, does not utilize previous state information; its computation time is therefore independent of $\Delta\varphi$. Interestingly, we observe a plateau in the Lanczos computation time between $n = 10$ and $n = 25$. This occurs because the computational cost of Lanczos is primarily determined by the size of the constructed Krylov subspace, which typically scales as $\min(N_{dim}, \max(2n + 1, 20))$, where N_{dim} represents the total size of the Hilbert space. For this specific benchmark, based on the system dimensions provided by our supervisor, the total Hilbert space size is $N_{dim} = 97280$. Since the subspace sizes required for the 10th and 25th states are very similar in this regime, the difference in computational cost remains negligible.

Regarding numerical accuracy (fig. 12, Right), while Shift-Invert is strictly less precise than Lanczos, it easily achieves a highly satisfactory variance of around 10^{-18} GHz^2, which is more than sufficient for physical applications.

The most insightful difference between the two methods lies in the state overlap behavior. For small steps ($\Delta\varphi = 10^{-4}$ and 10^{-3}), the curves perfectly overlap for both methods and remain near 1.0, indicating no unphysical state jumps. However, at a larger step of $\Delta\varphi = 10^{-2}$, the overlap for the Lanczos method abruptly collapses to zero at higher indices ($n = 100$), whereas Shift-Invert maintains a high overlap.

This apparent collapse with the Lanczos method is not a computational failure, but rather a fundamental difference in how the two algorithms traverse energy level crossings. Because Lanczos computes the broad lower spectrum ($[0, n]$), it strictly orders the output eigenstates by their ascending eigenvalues (*adiabatic tracking*). During an energy crossing, the physical characteristics of state n and state $n+1$ are exchanged. If the step $\Delta\varphi$ is large enough to jump completely over this crossing, Lanczos blindly returns the new state at the n-th energy slot, which now possesses a completely different physical nature.

In contrast, the Shift-Invert method inherently performs *diabatic tracking*. Because it relies on the previous state as its initial guess, it naturally biases convergence toward the eigenstate with the highest physical resemblance. It smoothly follows the continuous physical branch right through the crossing, regardless of its new eigenvalue ranking. Therefore, the overlap drop observed with Lanczos is strictly a **labelling issue** (an artifact of sorting the basis strictly by energy) and absolutely not a real physical problem or a loss of accuracy.

6 Conclusion

In this study, we investigated the numerical simulation of the KITE, a parity-protected superconducting qubit. After introducing the necessary quantum mechanical formalism, we highlighted a major computational bottleneck: the multi-mode KITE architecture generates a massive Hilbert space, rendering Exact Diagonalization intractable. To bypass this, the standard approach relies on the Lanczos algorithm. However, to avoid redundantly recalculating the lower spectrum during continuous parameter sweeps, we implemented a new iterative tool: a matrix-free Shift-Invert solver.

By systematically benchmarking its core hyperparameters—the regularization shift, the MINRES tolerance, and the number of power iterations—we identified an optimal configuration for numerical stability and speed. Evaluating this solver against Lanczos revealed that while Lanczos remains superior for computing broad spectra from scratch, our Shift-Invert method excels at tracking a single target state. In this specific scenario, it successfully beats Lanczos in computation time and naturally performs diabatic tracking, seamlessly following the correct physical branch through avoided crossings.

Finally, while our numerical benchmarks were successfully executed on a standard personal computer—underscoring the efficiency of our approach—significant optimization avenues remain. Writing a natively complex-valued MINRES solver would eliminate the need for our real-valued isomorphism, dividing the memory footprint by four. Ultimately, deploying this optimized algorithm on supercomputers will be the necessary next step to simulate even larger Hilbert spaces and assist in the design of protected qubits.

Data and Code Availability

The numerical simulations were implemented in Python. We utilized NumPy [8] for core array operations and the SciPy library [7] for sparse matrix manipulations, specifically employing its MINRES and Lanczos (`eigsh`) Krylov solvers. Quantum system properties were verified and visualized using the `scqubits` library [10], while all general data tracing and benchmarking plots were generated with Matplotlib [9]. The complete source code reproducing the numerical methods, benchmarks, and figures presented in this report is openly available on GitHub at `https://github.com/Grucciman/TR-Control.git`.

Acknowledgments

I would like to sincerely thank Emilio Rui and Alexandru Petrescu for the time they dedicated to us. Their guidance through our regular meetings and discussions helped us navigate this highly stimulating project from start to finish.
My thanks also go to Kyrylo Gerashchenko, particularly for the time he spent early in the internship familiarizing us with the quantum mechanics formalism, as well as for providing the reference code for the KITE Hamiltonian.
I am grateful to Zaki Leghtas for introducing this topic during the TR Control presentation and for overseeing our progress throughout the trimester.
Finally, a special thanks to Gabriel Vandersippe, with whom I collaborated closely at the beginning of the internship. Working together made this research experience much less solitary and far more enriching.

References

[1] Blais, *et al., Reviews of Modern Physics* **93 (2)**, 025005 (2021)

[2] Koch, *et al., Phys. Rev. A* **76**, 042319 (2007)

[3] **Introduction to Quantum Electromagnetic Circuits**, U. Vool and M. Devoret, *International Journal of Circuit Theory and Applications* **45**, 897-934 (2017)

[4] Smith, *et al., Phys. Rev. X* **12**, 021002 (2022)

[5] A. Gyenis *et al.*, *Moving beyond the transmon: Noise-protected superconducting quantum circuits*, *PRX Quantum* **2**, 030101 (2021).

[6] S. Messelot, A. Leblanc, J.-S. Tettekpoe, F. Lefloch, Q. Ficheux, J. Renard, and É. Dumur, arXiv:2601.10209 (2026)

[7] P. Virtanen *et al.* (SciPy 1.0 Contributors), *Nature Methods* **17**, 261–272 (2020).

[8] C. R. Harris *et al.*, *Nature* **585**, 357–362 (2020).

[9] J. D. Hunter, *Computing in Science & Engineering* **9(3)**, 90–95 (2007).

[10] P. Groszkowski and J. Koch, *Quantum* **5**, 583 (2021).