

PAPER • OPEN ACCESS

Computer-aided quantization and numerical analysis of superconducting circuits

To cite this article: Sai Pavan Chitta *et al* 2022 *New J. Phys.* **24** 103020

View the [article online](#) for updates and enhancements.

You may also like

- [Quantum information processing with superconducting circuits: a review](#)
G Wendin
- [CircuitQ: an open-source toolbox for superconducting circuits](#)
Philipp Aumann, Tim Menke, William D Oliver et al.
- [A review of design concerns in superconducting quantum circuits](#)
Eli M Levenson-Falk and Sadman Ahmed Shanto



PAPER

Computer-aided quantization and numerical analysis
of superconducting circuitsSai Pavan Chitta^{*} , Tianpu Zhao , Ziwen Huang¹ , Ian Mondragon-Shem
and Jens Koch

Department of Physics and Astronomy, Northwestern University, Evanston, Illinois 60208, United States of America

^{*} Author to whom any correspondence should be addressed.¹ Current address: Superconducting Quantum Materials and Systems Center, Fermi National Accelerator Laboratory (FNAL),
Batavia, IL 60510, USA.E-mail: saichitta2025@u.northwestern.edu**Keywords:** quantum computing, superconducting circuits, circuit quantizationRECEIVED
24 June 2022REVISED
12 September 2022ACCEPTED FOR PUBLICATION
26 September 2022PUBLISHED
15 November 2022Original content from
this work may be used
under the terms of the
[Creative Commons
Attribution 4.0 licence](#).Any further distribution
of this work must
maintain attribution to
the author(s) and the
title of the work, journal
citation and DOI.**Abstract**

The development of new superconducting circuits and the improvement of existing ones rely on the accurate modeling of spectral properties which are key to achieving the needed advances in qubit performance. Systematic circuit analysis at the lumped-element level, starting from a circuit network and culminating in a Hamiltonian appropriately describing the quantum properties of the circuit, is a well-established procedure, yet cumbersome to carry out manually for larger circuits. We present work utilizing symbolic computer algebra and numerical diagonalization routines versatile enough to tackle a variety of circuits. Results from this work are accessible through a newly released module of the `scqubits` package.

1. Introduction

The past two decades have seen dramatic progress in the design and control of quantum devices based on superconducting circuits [1–3]. A central part of this effort is the design of qubits which are robust against decoherence and can be controlled and measured with high fidelity [4–6]. The systematic discovery of new qubits requires exploring the wealth of possible circuits formed by combining inductors, capacitors, and Josephson junctions as the basic circuit elements. This has led to the design of a growing number of qubit types, each with its own advantages [4, 7–12]. The task of exploring new circuit designs, however, can be tedious and become numerically challenging since the Hilbert space dimension needed to describe the low-energy physics of the system tends to grow exponentially with the number of elements in the circuit. In order to make progress in the design of devices of increasing complexity, it is essential to promote tools that facilitate the efficient analysis of spectra and related key properties of superconducting circuits.

Several prescriptions have been proposed to approximate the low-energy Hamiltonian of a superconducting circuit, each depending on the level of detail needed for an accurate description [13–18]. For example, several approaches and software tools model the full geometry and distributed nature of the circuit [13, 15, 18]. This is computationally costly, and often works most readily for circuits with only weak anharmonicity. The most basic, yet quite widely applicable quantization framework is based on the lumped-element description of a circuit [17, 19–23]. Despite the fact that this approach is the simplest one, it nonetheless presents challenges once the number of circuit elements is increased beyond the bare minimum of two (found, for example, in the transmon qubit). This has led to the development of software packages that aim to automate the construction of the Hamiltonian [24–27].

In this work, we present a streamlined computer-aided approach to quantize lumped-element systems. By examining the circuit topology, we show how to classify and systematically detect four different types of degrees of freedom in a user-specified circuit: free, frozen, periodic and extended degrees of freedom. The identification of these variables can help reduce the number of degrees of freedom needed to describe the system and makes manifest the basis that should be used for an efficient representation of the Hamiltonian. This functionality has recently been incorporated into the `scqubits` Python package [28], leveraging the

full array of capabilities already available. We complement our approach with a hierarchical-diagonalization algorithm. This further mitigates the circuit-size limitations governed generally by the size of the Hamiltonian matrix which describes the low-energy spectrum of interest.

For illustration, consider the task of finding the circuit Hamiltonian and numerical eigenenergies of a KITE circuit [29]. Using a simple input file KITE.yaml that encodes the circuit graph and associated circuit-element parameters (see appendix A.1), generating the circuit Hamiltonian and performing numerical diagonalization is readily achieved by a few lines of code:

```
In: import squbits as sqq
kite = sqq.Circuit("KITE.yaml", ext_basis="harmonic")
kite.sym_hamiltonian()

Out: (15.7Q12 + 26.4Q1Q2 - 52.8Q1Q3 + 26.4Q22 - 52.8Q2Q3 + 52.8Q32) + (0.18Φ12 + 0.72Φ1θ1 - 0.36Φ1θ2 +
0.36Φ1θ3 + 0.18Φ22 + 0.72Φ2θ1 - 0.36Φ2θ2 + 1.9θ12 - 1.44θ1θ2 + 1.18θ1θ3 + 0.36θ22 - 0.36θ2θ3 +
0.295θ32 - 5.9 cos(θ2) - 5.9 cos(θ2 + θ3))
```

Once suitable parameters for Hilbert space truncation are chosen, the eigenenergies are obtained simply by

```
In: kite.eigenvals()

Out: array([4.17710271, 6.47610652, 7.24813846, 8.47986653, 8.713839, 9.19210006])
```

The outline of the paper is as follows. In section 2 we set the stage for the conceptual discussion of our approach by reviewing the lumped-element description of superconducting circuits. This includes a simple discussion of constructing the circuit Lagrangian in terms of node variables. In section 3, we show how to systematically identify the four types of degrees of freedom in circuits consisting of capacitances, inductances², and Josephson junctions. In section 4, we describe how the identification of these variables is implemented in the squbits package to efficiently represent the Hamiltonian of a user-provided circuit. Finally, section 5 concludes with a summary and outlook of future developments.

2. Circuit Lagrangian and classification of variables

2.1. Brief review of circuit Lagrangians

The common starting point for the quantization of a superconducting circuit is the construction of a circuit Lagrangian [19–22, 30, 31]. In this paper, we exclusively focus on lumped-element circuits which are composed of capacitances, inductances, and Josephson tunnel junctions. (For more general discussions including additional elements such as gyrators, or extending the discussion beyond the lumped-element regime, consult references [32–35].) Although seemingly restrictive, this small set of circuit elements spans a large family of different circuits, and includes the majority of superconducting qubits studied in the past, and used today. Each circuit element forms a branch in the connected circuit network.

Formulation of the Lagrangian depends on the definition of an appropriate set of *generalized coordinates* (which we also refer to as *variables*). In circuits including Josephson junctions, generalized-flux variables are the most convenient choice. For each branch b , one defines

$$\Phi_b(t) = \int_{t_0}^t d\tau V_b(\tau), \quad (1)$$

where t_0 is a reference time³ and $V_b(t)$ is the voltage across branch b . The energy stored in a branch b is given by the

$$\begin{array}{lll} \text{charging} & : E = \frac{C_b \dot{\Phi}_b^2}{2}, & \text{inductive} : E = \frac{\Phi_b^2}{2L_b}, & \text{junction} : E = -E_{Jb} \cos(2\pi\Phi_b/\Phi_0), \\ \text{energy} & & \text{energy} & \end{array}$$

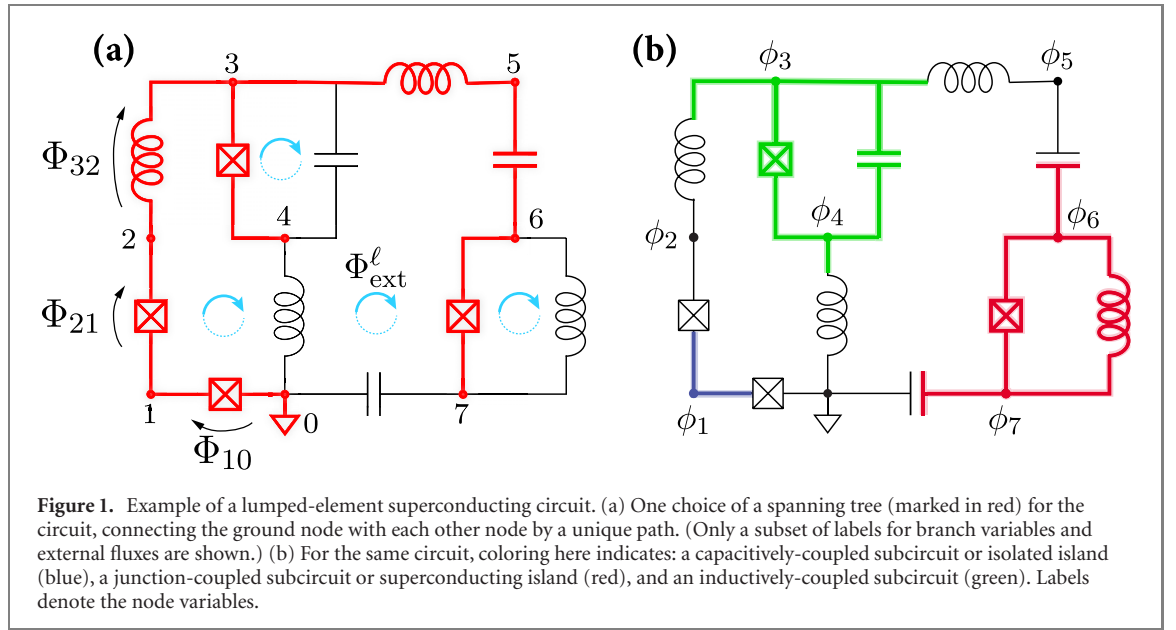
for a capacitance C_b , inductance L_b , and junction with Josephson energy E_{Jb} , respectively. ($\Phi_0 = h/2e$ is the flux quantum.)

The collection of all branch variables $\Phi_b(t)$ is in general not a set of independent variables: for each elementary loop L_ℓ in the circuit, Kirchhoff's loop rule in conjunction with fluxoid quantization [21, 36] imposes a constraint on the corresponding branch fluxes:

$$\Phi_{\text{ext}}^\ell = \sum_{b \in L_\ell} (\pm)_b \Phi_b(t). \quad (2)$$

² Terminology employed here refers to the ideal circuit elements by 'capacitances' and 'inductances', whereas 'capacitor' and 'inductor' are understood as real building blocks that will include parasitic inductance and capacitance.

³ In some references, t_0 is chosen as $-\infty$. To avoid questions about convergence of the integral, we choose t_0 to be finite and denote a time in the 'distant' past when no external fields were present.



Here, Φ_{ext}^ℓ is the external magnetic flux through the ℓ th loop, and the branch sign $(\pm)_b$ is determined according to the directionality of the branch relative to the chosen loop orientation.

The construction of a spanning tree provides a systematic way to determine a set of independent branch variables consistent with the constraints equation (2). A spanning tree T is a subset of connected branches such that every circuit node is connected to ground by one, and only one, path within the spanning tree [21] (see the example in figure 1(a))⁴. Each branch not part of the spanning tree is a closure branch. Iterative inclusion of closure branches, i.e., starting from the spanning tree, adding the first closure branch, then the second, etc, leads to the closing of one circuit loop at a time⁵. Once the constraints have been resolved in this way, the circuit Lagrangian can be expressed in terms of the branch variables $\Phi_{b \in T}$ belonging to the spanning tree, with $\Phi_{b \notin T}$ eliminated by means of equation (2),

$$\mathcal{L} = T - V = \sum_b \left[\frac{1}{2} C_b \dot{\Phi}_b^2 - \frac{1}{2L_b} \Phi_b^2 + E_{J,b} \cos(2\pi \Phi_b / \Phi_0) \right] \Big|_{b \notin T: \text{replace } \Phi_b \text{ via equation (2)}}. \quad (3)$$

Here, notation follows a simple scheme assigning the capacitance C_b to branch $b = (n', n)$ if a capacitance indeed connects nodes n and n' . Otherwise, the term is omitted from the summation. Analogous statements hold for inductance and junction contributions to the potential energy. We note in passing that time-dependent external flux renders the holonomic constraint (2) among the coordinates time-dependent, and hence also yields a constraint among the generalized velocities $\dot{\Phi}_b$ [35, 38]; for simplicity we limit ourselves to the time-independent case here.

Node variables present a useful alternative to branch variables. The node variable ϕ_n at node n corresponds to the generalized flux obtained by adding up branch fluxes along the spanning tree; specifically, the spanning tree defines one path P_n connecting the ground node to node n . Then, the node variable ϕ_n is obtained by summing the branch variables along that path⁶:

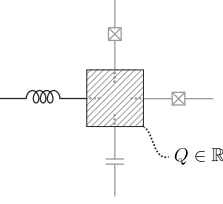
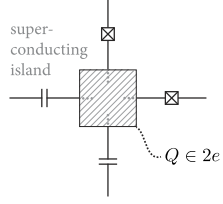
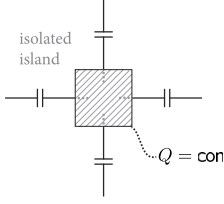
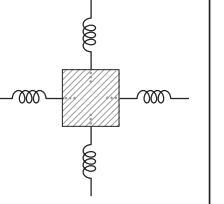
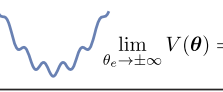
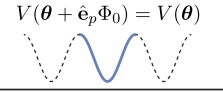
$$\phi_n = \sum_{b \in P_n} \Phi_b, \quad (4)$$

⁴ Spanning trees are not unique. This reflects the fact that the constraints (2) leave open the choice of which branch variable in a loop to eliminate.

⁵ Careful analysis of the circuit in figure 1 illustrates that there are cases in which the inclusion of the next closure branch appears to close multiple loops at a time. However, given the motivating purpose of resolving the loop constraints, it is always the newly closed loop containing exactly one closure branch which determines the relevant external loop flux. A more extensive discussion of this point has been presented in [37].

⁶ Note: while the voltage across branch b is well-defined, one cannot generally fall back to a description in terms of electric-potential differences. In the presence of non-zero external magnetic flux, even if constant in the present, the magnetic field had to be switched on after t_0 and be ramped up to its present value. As a result, the integral in equation (1) necessarily extends over a time interval during which the electric field was not curl-free, and could not be represented as a gradient field of the electric potential.

Table 1. Overview of the different types of variables. Classification is stated readily in the Lagrangian framework: a variable is *extended* if the potential is confining, and *periodic* if the potential is periodic along variable's axis. A variable (or coordinate) absent from the Lagrangian is free, while a generalized velocity absent leads to a *frozen* variable. Each variable category is associated with specific conditions on how the corresponding subcircuit (hatched square) connects to the remaining circuit. The total charge Q on the subcircuit is continuous for extended variables, quantized in units of $2e$ for periodic variables, and conserved in case of a free variable.

extended variable charge continuous	periodic variable charge quantized	free variable charge conserved	frozen variable
			
$Q \in \mathbb{R}$	$Q \in 2e \cdot \mathbb{Z}$	$Q = \text{const.}$	
subcircuit connected to inductances and capacitances	subcircuit connected to junctions and capacitances	subcircuit connected to capacitances only	subcircuit connected to inductances only
potential energy is confining w.r.t. coordinate θ_e	potential energy is periodic w.r.t. coordinate	coordinate absent from Lagrangian	velocity absent from Lagrangian
 $\lim_{\theta_e \rightarrow \pm\infty} V(\theta) = \infty$	 $V(\theta + \hat{e}_p \Phi_0) = V(\theta)$	$\partial_{\theta_e} \mathcal{L}(\dot{\theta}, \theta) = 0$	$\partial_{\dot{\theta}_f} \mathcal{L}(\dot{\theta}, \theta) = 0$

as illustrated in figure 1(a). Branch variables within the spanning tree as well as the generalized flux associated with each closure branch can then be expressed in terms of node variables via

$$\Phi_{b \in T} = \phi_n - \phi_{n'}, \quad \Phi_{b \notin T} = \phi_n - \phi_{n'} + \Phi_{\text{ext}}^\ell. \quad (5)$$

Here n, n' are the nodes connected by the branch $b = (n, n')$, and Φ_{ext}^ℓ is the external flux threading the loop ℓ formed by the closure branch b and the spanning tree. The circuit Lagrangian (2) can then be re-expressed in terms of node variables in the general form

$$\mathcal{L}(\dot{\phi}, \phi) = \frac{1}{2} \dot{\phi}^T \mathbf{C} \dot{\phi} - V(\phi), \quad (6)$$

where $\phi^T = (\phi_1, \dots, \phi_N)$, and the $N \times N$ capacitance matrix \mathbf{C} has non-zero off-diagonal entries $(C)_{nn'} = -C_b$ whenever a capacitance C_b connects nodes $n \neq n'$; diagonal entries are given by $(C)_{nn} = -\sum_{m \neq n} (C)_{nm}$. The potential energy in node variables is obtained via equation (3) by substitution,

$$V(\phi) = \sum_{b=(n,n')} \left[\frac{1}{2L_b} (\phi'_n - \phi_n + \text{cl}_b \Phi_{\text{ext}}^{\ell_b})^2 - E_{J,b} \cos \left(\frac{2\pi}{\Phi_0} [\phi'_n - \phi_n + \text{cl}_b \Phi_{\text{ext}}^{\ell_b}] \right) \right], \quad (7)$$

where cl_b is the indicator for closure branches, i.e. $\text{cl}_b = 1$ for $b \notin T$ and $\text{cl}_b = 0$ otherwise.

In principle, one might now proceed with either the branch- or node-variable Lagrangian, perform a Legendre transformation to obtain the Hamiltonian, and subsequently impose canonical quantization rules. However, it is generally worthwhile (especially where the ultimate aim involves numerical treatment) to consider a further transformation of variables which will achieve two important goals: (1) we may reduce the number of relevant variables, for example by identifying conserved quantities, and (2) we cleanly distinguish variables associated with different boundary conditions upon quantization. A suitable variable transformation can thus help make the quantum description of the circuit both more transparent and numerically efficient.

2.2. Classification of variables

Any variable θ , no matter whether branch, node, or collective variable of neither type, can be classified based on four categories (table 1):

- **Extended:** a variable θ_e is extended if it describes a direction in configuration space in which the potential energy is confining, expressed as $\lim_{\theta_e \rightarrow \pm\infty} V(\theta) = \infty$.
- **Periodic:** a variable θ_p is periodic if it aligns with a direction in configuration space in which the potential energy is periodic, i.e., $V(\theta + \Phi_0 \mathbf{e}_p) = V(\theta)$. Here, \mathbf{e}_p is the unit vector pointing along the axis associated with the θ_p coordinate. We exclude the case of a variable in which the potential energy is constant; this situation is covered by the following category.

- **Free:** a variable θ_c is free⁷ if $\partial_{\theta_c} \mathcal{L}(\dot{\theta}, \theta) = 0$. This means that θ_c does not appear in the potential energy, and the corresponding conjugate charge $Q_c = \partial_{\dot{\theta}_c} \mathcal{L}(\dot{\theta}, \theta)$ is conserved (the subscript c refers to constant charge).
- **Frozen:** a variable θ_f is frozen if $\partial_{\dot{\theta}_f} \mathcal{L}(\dot{\theta}, \theta) = 0$. In this case $\dot{\theta}_f$ is missing from the kinetic energy. This somewhat pathological case only arises when too many parasitic capacitances are neglected in the lumped element model. In the simplest setting, a frozen variable leads to an additional constraint given by $\partial_{\theta_f} \mathcal{L}(\dot{\theta}, \theta) = 0$.

free	periodic	extended
	frozen	

The diagram above shows the relation among these four categories. The *free*, *periodic*, and *extended* categories are mutually exclusive properties of the potential energy. The *frozen* category regards the kinetic energy and as such can occur for each of the former three categories.

This classification holds several advantages when analyzing the Hamiltonian of a circuit. Recognizing free and frozen variables helps reduce the number of degrees of freedom needed to describe the system. Furthermore, knowledge of periodic and extended variables guides a consistent and efficient choice of basis respecting the appropriate boundary conditions. Neither branch nor node variables are generally the suitable choice which recognizes the full number of periodic, free, and frozen degrees of freedom. Failure to find all the variables of each type makes diagonalization of the quantum Hamiltonian usually more difficult, and can potentially lead to wrong conclusions about offset-charge dependence and charge-noise sensitivity of circuit eigenstates.

3. Adaptive variable transformation

In the following, we show how to systematically construct a linear variable transformation which will pinpoint all periodic, free, and frozen degrees of freedom. Against naive instinct, it is helpful to define this transformation by expressing the ‘old’ node variables in terms of the new variables, i.e., $\phi_n = \sum_{m=1}^N z_{nm} \theta_m$, where $\mathbf{Z} = (z_{nm})$ is an $N \times N$ invertible transformation matrix.

We determine the z_{nm} by studying the subcircuits of the system, defined as connected sub-graphs of the circuit. We distinguish three relevant classes of subcircuits: (1) a subcircuit forms an *isolated island* if it is connected to the rest of the circuit exclusively by capacitances; (2) a subcircuit forms a *superconducting island* if it is coupled to the remainder of the circuit by capacitances and at least one junction (but not via any inductances); (3) a subcircuit is purely inductively coupled if it connects to the rest of the circuit only through inductances. Examples of each type of subcircuit are shown in figure 1(b).

3.1. Periodic and extended variables

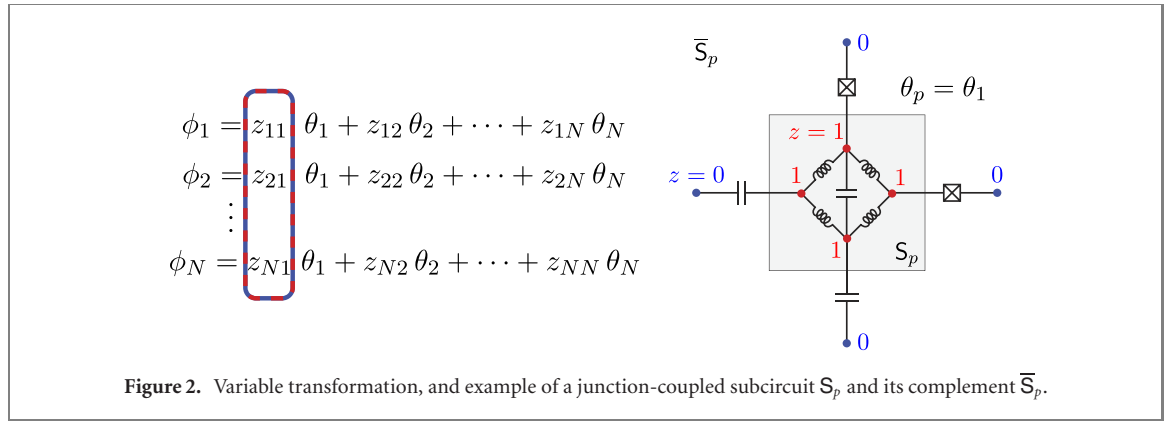
For the analysis of periodic variables, consider a subcircuit \mathbf{S}_p forming a *superconducting island* as in the example of figure 2. There, \mathbf{S}_p is shaded in gray and couples to its circuit complement $\bar{\mathbf{S}}_p$ exclusively through capacitances and junctions. Any inductances present in the circuit are entirely inside \mathbf{S}_p or inside $\bar{\mathbf{S}}_p$, but not at their interface. Given this partitioning, we see there is a simple choice of coefficients z_{n1} that will render the variable θ_1 periodic, namely:

$$z_{n1} = 1 \quad \text{for all } n \in \mathbf{S}_p, \quad z_{n'1} = 0 \quad \text{for all } n' \in \bar{\mathbf{S}}_p. \quad (8)$$

In other words, a change in the new flux variable θ_1 will simultaneously raise the flux of each node variable inside the superconducting island by equal amounts, but keep the flux on all nodes outside the superconducting island unchanged. Since θ_1 clearly produces no flux difference across any inductance, the potential energy must indeed be periodic with respect to θ_1 .

The full set of periodic variables is obtained by identifying all superconducting islands in the circuit. This indicates that the full number of periodic variables, N_p , is equal to the number of *irreducible* superconducting islands \mathbf{S}_p which cannot be decomposed into smaller superconducting islands. Once all periodic variables are assigned, any remaining variable θ_e must now appear in the quadratic potential energy terms generated by the inductances. These terms have the usual confining property $\lim_{\theta_e \rightarrow \pm\infty} V(\theta) = \infty$. As a result, all these degrees of freedom are extended.

⁷ Another common, though potentially misleading, name for this variable type is ‘cyclic’.



3.2. Free variables

Free variables are treated in a manner nearly identical to the periodic case. Suppose $\theta_c = \theta_1$ is one of the free variables. In this case, changes in θ_c cannot produce node-flux differences across any of the inductances or junctions. Again, partitioning into subcircuits facilitates the definition of new variables. For free variables, the subcircuit S_c must now be an *isolated island* connected to the remainder of the circuit \bar{S}_c via capacitances exclusively. Carrying over the definition of z_{n1} to the isolated island case,

$$z_{n1} = 1 \quad \text{for all } n \in S_c, \quad z_{n1} = 0 \quad \text{for all } n \in \bar{S}_c, \quad (9)$$

equips us with a free variable: since flux differences generated by θ_1 are across capacitances only, θ_1 will be absent from all potential energy terms associated with inductances and junctions. As a result, $\partial_{\theta_c} V(\theta) = 0$, so θ_c is indeed free. The full number of free variables, N_c , corresponds to the total number of irreducible isolated islands in the given circuit.

3.3. Frozen variables

As the last category, we discuss the case of frozen variables. For a frozen variable, the relevant subcircuit is of type (3): a subcircuit coupled to the remaining circuit through inductances only. Once such a subcircuit is identified, a frozen variable θ_f is readily defined following the now familiar way: z -coefficients for nodes inside and outside the subcircuit are set to 1 and 0, respectively. As a result, the only flux differences that occur under a change in θ_f are across inductance terms in the Lagrangian. Kinetic energy terms due to capacitances remain unchanged such that $\partial_{\dot{\theta}_f} \mathcal{L} = 0$, and θ_f is, hence, indeed a frozen variable. The full number of frozen variables corresponds to the total number of irreducible, purely inductively coupled subcircuits.

A few remarks about frozen variables are in order. Variables only appear frozen in situations where too many parasitic capacitances have been dropped from the description and the capacitance matrix has become singular. A foolproof, physically accurate but not necessarily efficient way of handling this situation is to re-admit the omitted parasitic capacitances which physically are always present. This is not always desirable as it generally increases the number of degrees of freedom retained in the modeling of the circuit.

Dropping a parasitic capacitance from the charging-energy contributions is akin to setting this capacitance to zero. In the mechanically analogous system, this corresponds to setting the mass of a particle to zero. The $C \rightarrow 0$ limit is generally a singular limit [39] that exhibits subtleties particularly significant when junction capacitances are omitted [32]. We exclude this case here and focus on the more benign scenario in which one of the Euler–Lagrange equations turns into a linear holonomic constraint for $C = 0$, see subsection 3.5.

3.4. Full transformation

As an example, consider the circuit in figure 1(b), which has three subcircuits. The corresponding transformation is

$$\begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & & & & \\ 0 & 0 & 0 & & \vdots & & \\ 0 & 1 & 0 & & & & \\ 0 & 1 & 0 & \dots & z_{nm} & \dots & \\ 0 & 0 & 0 & & & & \\ 1 & 0 & 0 & & \vdots & & \\ 1 & 0 & 0 & & & & \end{pmatrix} \begin{pmatrix} \theta_c \\ \theta_f \\ \theta_p \\ \theta_{e,1} \\ \theta_{e,2} \\ \theta_{e,3} \\ \theta_{e,4} \end{pmatrix}. \quad (10)$$

This transformation identifies a free, a frozen and a periodic variable. The remaining variables are extended. The matrix elements corresponding to the extended variables $\{\theta_{e,i}\}_{i=1}^4$ are chosen such that the transformation is invertible. The Lagrangian in terms of the new variables is

$$\mathcal{L}(\dot{\theta}, \theta) = \frac{1}{2} \dot{\theta}^T \tilde{\mathbf{C}} \dot{\theta} - V(\theta), \quad (11)$$

where $\tilde{\mathbf{C}} = \mathbf{Z}^T \mathbf{C} \mathbf{Z}$.

3.5. Elimination of non-dynamical variables

To obtain the circuit Hamiltonian, we first eliminate the frozen and free variables. Eliminating frozen variables is, in fact, a necessary step, since the capacitance matrix (equation (11)) is singular in the presence of frozen variables.

To carry out the elimination of frozen variables from the Lagrangian, we use that $\partial_{\theta_{f,i}} \mathcal{L} = \frac{d}{dt} \partial_{\dot{\theta}_{f,i}} \mathcal{L} = 0$, where $i = 1, \dots, N_f$ labels the inductively-coupled subcircuits. In general, this set of N_f equations are nonlinear due to the cosine terms in the Lagrangian. As a crucial simplification, we assume that none of the junction capacitances are omitted in the description. In that case, only subcircuits with purely linear-inductive coupling to the remainder of the circuit lead to frozen variables. Thus, the variables $\{\theta_{f,i}\}$ can be solved for using the constraints emerging from the Euler–Lagrange equations. Upon eliminating the $\{\theta_{f,i}\}$ from the Lagrangian, the kinetic energy is expressed in terms of the invertible capacitance matrix $\tilde{\mathbf{C}}_{cpe}$ obtained by removing the rows and columns in $\tilde{\mathbf{C}}$ that were associated with frozen variables. At this point, we perform a Legendre transform to obtain the classical Hamiltonian

$$H = \mathbf{Q}_{cpe}^T \cdot \dot{\theta}_{cpe} - \mathcal{L}(\dot{\theta}_{cpe}, \theta_{pe}),$$

where $\mathbf{Q}_{cpe} = \partial_{\dot{\theta}_{cpe}} \mathcal{L}$ and θ_{cpe} corresponds to free, periodic and extended variables.

For every free variable θ_c , the conjugate charge \mathbf{Q}_c is a conserved quantity. As shown in appendix B, these conserved charges have no impact on the dynamics of the remaining degrees of freedom, and they may hence be omitted. The kinetic energy of the Hamiltonian can thus be written in the form $T = \frac{1}{2} \mathbf{Q}_{pe}^T \tilde{\mathbf{C}}_{pe}^{-1} \mathbf{Q}_{pe}$, where $\tilde{\mathbf{C}}_{pe}^{-1}$ is the matrix block of $\tilde{\mathbf{C}}_{cpe}^{-1}$ that corresponds to the extended and periodic variables only. Similarly, the vector \mathbf{Q}_{pe} only includes the non-conserved charges conjugate to the variables θ_{pe} .

3.6. Quantization

Finally, canonical quantization yields the quantum mechanical Hamiltonian. Quantization is achieved by promoting the coordinate and conjugate momenta to operators $\hat{\theta}_n$ and $\hat{Q}_n = -i\hbar \partial_{\theta_n}$ satisfying canonical commutation relations. The Hamiltonian finally takes the form

$$\hat{H} = \frac{1}{2} (\hat{\mathbf{Q}}_{pe} - \mathbf{Q}_g)^T \tilde{\mathbf{C}}_{pe}^{-1} (\hat{\mathbf{Q}}_{pe} - \mathbf{Q}_g) + V(\hat{\theta}_{pe}), \quad (12)$$

where we have introduced a vector \mathbf{Q}_g representing off-set charges. Since periodic and extended degrees of freedom are identified explicitly, the boundary conditions required for the wavefunction $\Psi(\theta_{pe})$ can be stated transparently:

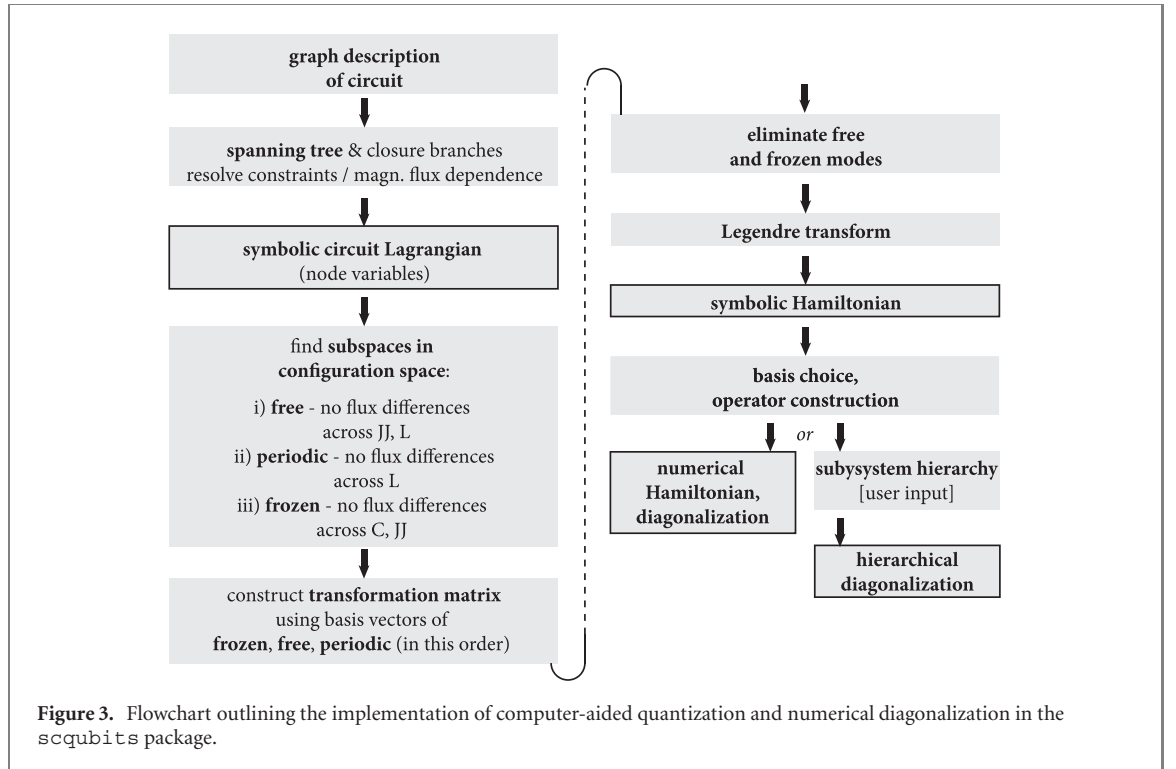
$$\text{Periodic directions: } \Psi(\theta_{pe} + \mathbf{e}_p \Phi_0) = \Psi(\theta_{pe}), \quad \text{extended directions: } \lim_{\theta_e \rightarrow \pm\infty} \Psi(\theta_{pe}) = 0.$$

The basis choice is further simplified by the distinction between periodic and extended variables. Periodic degrees of freedom can be represented in the discrete charge basis, extended variables in the phase basis or an appropriately chosen oscillator basis. The process of constructing the Hamiltonian \hat{H} of a given circuit and obtaining its spectrum has been implemented in the `scqubits` package [28], as we explain in the following.

4. Implementation in `scqubits`

4.1. Variable transformation

The variable transform discussed in section 3 is implemented programmatically in order to automate the quantization of custom superconducting circuits. Several steps described in terms of subcircuits are equivalently formulated in more compact linear-algebra language applied to the vector space of node variables.



The algorithm proceeds by dividing the vector space of node variables into independent subspaces associated with the different variable categories. For instance, the subspace corresponding to periodic variables is spanned by those vectors $\mathbf{z}_p = (z_{1p}, z_{2p}, \dots, z_{Np})$ which are compatible with the condition of vanishing flux difference across all inductances. Similar subspaces are constructed for free and frozen degrees of freedom present in the circuit. The full transformation matrix⁸ is obtained by combining the vectors spanning the above three subspaces, and adding additional linearly independent vectors, if needed, to complete the basis.

Figure 3 summarizes the steps involved in this algorithm which has been implemented as a new module extending the `scqubits` package [28]. The `Circuit` class added for this purpose facilitates the generation and manipulation of symbolic circuit Hamiltonians by leveraging the Python package `SymPy`.

4.2. Numerical diagonalization

To obtain the spectrum of the circuit numerically, its Hamiltonian must be represented as a matrix with respect to a suitable basis. We choose bases $\{|\mu_i\rangle\}$ ($i = 1, 2, \dots$) for all individual degrees of freedom. For the periodic degrees of freedom we employ the discrete charge eigenstates as a convenient (though not necessarily ideal) basis. For extended degrees of freedom, we either apply simple discretization with multi-point stencils turning derivatives into finite differences, or employ the harmonic oscillator basis associated with the potential energy obtained when omitting junction terms.

In the ordinary diagonalization scheme, the basis of the full Hilbert space is chosen as the Kronecker product $|\mu_1\rangle \otimes |\mu_2\rangle \otimes \dots$. For each degree of freedom, individual truncation levels must be monitored to ensure convergence of the procedure. By contrast, hierarchical diagonalization aims to mitigate the fact that the dimension of the Hilbert space grows exponentially with the number of nodes⁹. The `Circuit` class offers an implementation of the hierarchical diagonalization procedure following ideas inspired by Kerman [40]. In the first stage, subsystems are diagonalized individually to obtain low-lying energy eigenstates $\{|E_i\rangle\}$. The basis of the full Hilbert space used in this case is the Kronecker product $|E_1\rangle \otimes |E_2\rangle \otimes \dots$. Under favorable conditions of moderate hybridization among subsystems, this procedure lowers the overall Hilbert space dimension. Once either of the two diagonalization schemes has been chosen, the full functionality of `scqubits` for visualizing spectra and wavefunctions, performing parameter sweeps, etc is available to explore the behavior of the custom circuit.

⁸ Note that the transformation matrix generated can have any real valued coefficients. The only condition is that $z_{ip} - z_{jp} \in \mathbb{Z}$ such that $i \in \mathbf{S}_p$ and $j \notin \mathbf{S}_p$, which ensures that the associated conjugate charge is periodic in a single Cooper pair. In some instances, having non-integer elements in the transformation matrix can help to increase the sparsity of the resultant matrix representation of the Hamiltonian, which can significantly reduce the diagonalization time [17].

⁹ Details of convergence depend not only on the node number, but also critically on the specific parameter regime determined by the circuit parameters.

As a concrete illustration of the hierarchical-diagonalization scheme, consider the fluxonium coupler circuit (see appendix A.2), composed of five nodes and four degrees of freedom. Simulation of the full circuit Hamiltonian generally calls for diagonalization of matrices of size $10^6 \times 10^6$. In suitable parameter regimes of practical interest, the system can be meaningfully divided into weakly coupled sub-systems, and hierarchical diagonalization proves to be a key step that decreases matrix sizes to $10^3 \times 10^3$ and dramatically reduces the required runtime by two orders of magnitude.

A natural question one might ask: what is the maximum size (number of nodes) of a circuit that can be analyzed with the `Circuit` class? Regrettably, there is no simple or general answer to this question. We can easily illustrate that, depending on the specific problem, two nodes can already pose a major challenge, while in other cases hundreds of nodes are straightforward to handle. For the former situation, consider a fluxonium circuit in the regime where the inductive energy is small compared to the charging and Josephson junction energies [41, 42]. Although this circuit only has a *single* degree of freedom, calculating its spectrum can be quite challenging, both when using discretization and harmonic-oscillator basis. By contrast, a circuit made exclusively of inductors and capacitors with *hundreds* of nodes is simple to diagonalize through normal-mode decoupling. These examples illustrate that the challenge of diagonalizing the Hamiltonian of a circuit is not easily quantified by circuit size, but dramatically depends on details including circuit parameters, the circuit anharmonicity, and network topology.

5. Conclusions

We described a computer-aided framework to quantize general lumped-element superconducting circuit systems. We showed how to systematically identify periodic, extended, and non-dynamical (free and frozen) degrees of freedom in a system. Eliminating the non-dynamical degrees of freedom leads to an efficient representation of the Hamiltonian. This framework is implemented programmatically in the new module `circuit` of `scqubits` Python library, which can simulate a user-defined circuit described as a graph. Hierarchical diagonalization was also implemented, which made numerical analysis for larger circuits accessible.

Future work will further extend the scope of the module by incorporating coherence-time calculations for custom circuits, handle the quantization and numerical analysis of circuits in the presence of time-dependent magnetic flux, and utilize the computation of perturbative corrections for hierarchical diagonalization discussed in reference [40]. Additional opportunities for enhancing the performance of our code lie in optimizing the choice of variable transformation which is generally not completely fixed by the protocol reported here. The remaining freedom may offer routes to increasing numerical efficiency, especially if variable transformations can reduce explicit coupling terms [17].

Note added

During preparation of this manuscript, we were made aware of ongoing work in Amir Safavi-Naeini's group with a similar scope. Our two groups did not discuss details of our projects, but agreed to post preprints on the arXiv simultaneously.

Acknowledgments

We thank Andrew Houck, Zlatko Mineev, Michel Devoret, Xinyuan You, Daniel Weiss and Venkat Chandrasekhar for valuable discussions. This material is based upon work in part supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Co-design Center for Quantum Advantage (C2QA) under Contract Number DE-SC0012704, and in part by the Army Research Office under Grant W911NF-1910016. For this work, IMS was funded by the DOE, ZH and SPC were funded by the ARO, and JK was funded by ARO and DOE. During the writing and editing of the manuscript, ZH was funded by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Superconducting Quantum Materials and Systems Center (SQMS) under Contract Number DE-AC02-07CH11359. We thank the Air Force Office of Scientific Research for supporting the `scqubits` package under Grant FA9550-20-1-0271.

Data availability statement

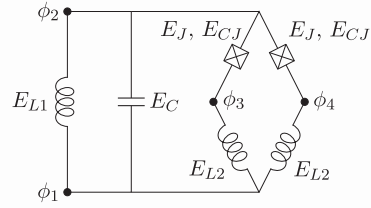
The data that support the findings of this study are available upon reasonable request from the authors.

Appendix A. Examples

A.1. User-defined transformation for the KITE circuit

The following string encodes the circuit graph for the example KITE circuit in section 1. This string can be passed to the method `from_yaml` directly or through a text file.

```
branches:
- [JJ, 2,3, 5.9, 6.6]
- [JJ, 2,4, 5.9, 6.6]
- [L, 1,2, 0.23]
- [C, 1,2, 2.5]
- [L, 1,3, 0.36]
- [L, 1,4, 0.36]
```



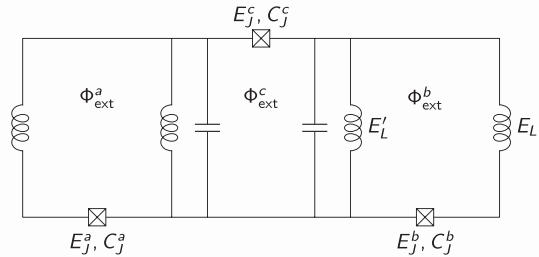
The parameters are (in units of GHz): $E_C = 2.5$, $E_{L1} = 0.23$, $E_{L2} = 0.36$, $E_J = 5.9$, $E_{CJ} = 6.6$. The following snippet sets the truncation for the Hilbert space, chooses a more efficient transformation following reference [29] and also chooses the closure branches.

```
In: trans_mat = np.array([[ -1,  -1,  -1,  1],
                          [ -1,  -1,   3,  1],
                          [ -1,   3,  -1,  1],
                          [  3,  -1,  -1,  1]])*0.25
closure_branches = [kite.branches[-2], kite.branches[-1]]
kite.configure(transformation_matrix=trans_mat, closure_branches=closure_branches)
kite.cutoff_ext_1 = 30
kite.cutoff_ext_2 = 30
kite.cutoff_ext_3 = 30
```

A.2. Hierarchical diagonalization for two fluxonia with a tunable coupler

The following string for the input file `fluxonium-coupler.yaml` encodes the circuit graph for the fluxonium coupler [43, 44]:

```
branches:
- [JJ,5,1,4.6,0.9]
- [L,1,2,0.21]
- [L,5,2,2.0]
- [C,5,2,50]
- [JJ,2,3,3,16.6]
- [L,5,3,2.0]
- [C,5,3,50]
- [L,3,4,0.21]
- [JJ,4,5,5.5,0.9]
```



The circuit can be diagonalized using hierarchical diagonalization by the following lines of code:

```
In: import scqubits as scq

fluxonium_coupler = scq.Circuit("fluxonium-coupler.yaml", ext_basis="harmonic",
                                initiate_sym_calc=False, basis_completion="canonical")

system_hierarchy = [[1], [2], [3], [4]]
closure_branches = [fluxonium_coupler.branches[i] for i in [0, 4, -1]]

fluxonium_coupler.configure(closure_branches=closure_branches, system_hierarchy=
                             system_hierarchy, subsystem_trunc_dims = [6, 6, 6, 6])

fluxonium_coupler.\Phi1 = 0.5 + 0.01768
fluxonium_coupler.\Phi2 = -0.2662
fluxonium_coupler.\Phi3 = -0.5 + 0.01768

fluxonium_coupler.cutoff_ext_1 = 110
fluxonium_coupler.cutoff_ext_2 = 110
fluxonium_coupler.cutoff_ext_3 = 110
fluxonium_coupler.cutoff_ext_4 = 110

eigs = fluxonium_coupler.eigenvals()
eigs - eigs[0]

Out: array([0. , 0.03559404, 0.05819727, 0.09378676, 4.39927874, 4.43488613])
```

We have verified that these eigenvalues agree with results obtained independently by the authors of [43].

Appendix B. Role of conserved charges in the dynamics of superconducting circuits

In this appendix, we show that conserved charges do not impact the physics of the dynamical variables. To simplify the analysis, we will assume that there is a single variable θ_1 that is free. We perform a partial Legendre transform with respect to the free variable to obtain the Routhian function

$$\mathcal{R}(\dot{\theta}_{pe}, \theta_{pe}; Q_1) = Q_1 \dot{\theta}_1 - \mathcal{L}(\dot{\theta}, \theta), \quad (\text{B.1})$$

where $Q_1 = \partial_{\dot{\theta}_1} \mathcal{L}(\dot{\theta}, \theta)$ and θ_{pe} represents the variables excluding θ_1 . The Routhian can be written in the form

$$\mathcal{R}(\dot{\theta}_{pe}, \theta_{pe}; Q_1) = -\mathcal{L}_0(\dot{\theta}_{pe}, \theta_{pe}) + \frac{1}{2(\tilde{\mathbf{C}}_{cpe})_{1,1}} \left(\sum_{i \neq 1} (\tilde{\mathbf{C}}_{cpe})_{i,1} (\dot{\theta}_{pe})_i \right)^2 + \frac{d}{dt} G(Q_1, t), \quad (\text{B.2})$$

where $\mathcal{L}_0(\dot{\theta}_{pe}, \theta_{pe})$ is the Lagrangian of the system without the free degree of freedom, and

$$G(Q_1, t) = -\frac{Q_1}{(\tilde{\mathbf{C}}_{cpe})_{1,1}} \left(\sum_{i \neq 1} (\tilde{\mathbf{C}}_{cpe})_{i,1} (\theta_{pe})_i \right) + \frac{Q_1^2}{2(\tilde{\mathbf{C}}_{cpe})_{1,1}} t. \quad (\text{B.3})$$

The dynamics of the variables θ_{pe} is determined by the Euler–Lagrange equations $\frac{d}{dt} \partial_{\dot{\theta}_{pe}} \mathcal{R} = \partial_{\theta_{pe}} \mathcal{R}$. Since the conserved charge Q_1 enters through $G(Q_1, t)$ as a total time derivative in the Routhian, the value of Q_1 has no impact on the dynamics. As a result, after obtaining the classical Hamiltonian, the charge Q_1 can be set to zero. The generalization to more than one free variable can be carried out straightforwardly.

ORCID iDs

Sai Pavan Chitta  <https://orcid.org/0000-0002-7882-6932>

Tianpu Zhao  <https://orcid.org/0000-0001-8264-9484>

Ziwen Huang  <https://orcid.org/0000-0001-5754-930X>

Ian Mondragon-Shem  <https://orcid.org/0000-0003-1322-7119>

Jens Koch  <https://orcid.org/0000-0002-5047-631X>

References

- [1] Gambetta J M, Chow J M and Steffen M 2017 *npj Quantum Inf.* **3** 2
- [2] Wendin G 2017 *Rep. Prog. Phys.* **80** 106001
- [3] Blais A, Grimsmo A L, Girvin S M and Wallraff A 2021 *Rev. Mod. Phys.* **93** 025005
- [4] Brooks P, Kitaev A and Preskill J 2013 *Phys. Rev. A* **87** 052306
- [5] Gyenis A, Di Paolo A, Koch J, Blais A, Houck A A and Schuster D I 2021 *Phys. Rev. X* **2** 030101
- [6] Zhang H, Chakram S, Roy T, Earnest N, Lu Y, Huang Z, Weiss D K, Koch J and Schuster D I 2021 *Phys. Rev. X* **11** 011010
- [7] Nakamura Y, Pashkin Y A and Tsai J S 1999 *Nature* **398** 786–8
- [8] Orlando T P, Mooij J E, Tian L, van der Wal C H, Levitov L S, Lloyd S and Mazo J J 1999 *Phys. Rev. B* **60** 15398–413
- [9] Koch J et al 2007 *Phys. Rev. A* **76** 042319
- [10] Manucharyan V E, Koch J, Glazman L and Devoret M 2009 *Science* **326** 113–6
- [11] Smith W C, Kou A, Xiao X, Vool U and Devoret M H 2020 *npj Quantum Inf.* **6** 8
- [12] Gyenis A, Mundada P S, Di Paolo A, Hazard T M, You X, Schuster D I, Koch J, Blais A and Houck A A 2021 *Phys. Rev. X* **2** 010339
- [13] Nigg S E, Paik H, Vlastakis B, Kirchmair G, Shankar S, Frunzio L, Devoret M H, Schoelkopf R J and Girvin S M 2012 *Phys. Rev. Lett.* **108** 240502
- [14] Bourassa J, Beaudoin F, Gambetta J M and Blais A 2012 *Phys. Rev. A* **86** 013814
- [15] Solgun F, Abraham D W and DiVincenzo D P 2014 *Phys. Rev. B* **90** 134504
- [16] Smith W C, Kou A, Vool U, Pop I M, Frunzio L, Schoelkopf R J and Devoret M H 2016 *Phys. Rev. B* **94** 144507
- [17] Ding D, Ku H S, Shi Y and Zhao H H 2021 *Phys. Rev. B* **103** 174501
- [18] Mineev Z K, Leghtas Z, Mundhada S O, Christakis L, Pop I M and Devoret M H 2021 *npj Quantum Inf.* **7** 131
- [19] Yurke B and Denker J S 1984 *Phys. Rev. A* **29** 1419–37
- [20] Devoret M H 1997 *Quantum Fluctuations in Electrical Circuits Quantum Fluctuations, Les Houches, Session LXIII* ed S Reynaud, E Giacobino and J Zinn-Justin (France: Edition de Physique)
- [21] Vool U and Devoret M 2017 *Int. J. Circ. Theor. Appl.* **45** 897–934
- [22] Burkard G, Koch R H and DiVincenzo D P 2004 *Phys. Rev. B* **69** 064503
- [23] Burkard G 2005 *Phys. Rev. B* **71** 144511
- [24] Klots A R (2013) Superquantpackage <https://github.com/andreyklots/SuperQuantPackage>
- [25] Parra Rodríguez A 2014 Software tool to analyse superconducting qubits PhD Thesis (Queensland University of Technology)
- [26] Gely M F and Steele G A 2020 *New J. Phys.* **22** 013025
- [27] Aumann P, Menke T, Oliver W D and Lechner W 2021 arXiv:2106.05342
- [28] Groszkowski P and Koch J 2021 *Quantum* **5** 583
- [29] Smith W C, Villiers M, Marquet A, Palomo J, Delbecq M R, Kontos T, Campagne-Ibarcq P, Douçot B and Leghtas Z 2022 *Phys. Rev. X* **12** 021002

- [30] Girvin S M 2011 *Circuit QED: Superconducting Qubits Coupled to Microwave Photons* (Oxford: Oxford University Press)
- [31] Rasmussen S, Christensen K, Pedersen S, Kristensen L, Bækkegaard T, Loft N and Zinner N 2021 *Phys. Rev. X* **2** 040204
- [32] Rymarz M, Bosco S, Ciani A and DiVincenzo D P 2021 *Phys. Rev. X* **11** 011032
- [33] Egusquiza I L and Parra-Rodriguez A 2022 arXiv:2203.06167
- [34] Parra-Rodriguez A 2021 arXiv:2104.09410
- [35] Riwar R P and DiVincenzo D P 2022 *npj Quantum Inf.* **8** 36
- [36] Tinkham M 2004 *Introduction to Superconductivity* (New York: Dover)
- [37] Thomas R E, Rosa A J and Toussaint G J 2016 *The Analysis and Design of Linear Circuits* (New York: Wiley)
- [38] You X, Sauls J A and Koch J 2019 *Phys. Rev. B* **99** 174512
- [39] Bender C M and Orszag S A 1999 *Advanced Mathematical Methods for Scientists and Engineers: I. Asymptotic Methods and Perturbation Theory* (Berlin: Springer)
- [40] Kerman A J 2020 arXiv:2010.14929 [quant-ph]
- [41] Koch J, Manucharyan V E, Devoret M H and Glazman L I 2009 *Phys. Rev. Lett.* **103** 217004
- [42] Pechenezhskiy I V, Mencia R A, Nguyen L B, Lin Y-H and Manucharyan V E 2020 *Nature* **585** 368–71
- [43] Weiss D K, Zhang H, Ding C, Schuster D I and Koch J 2022 High-fidelity entangling gates for fluxonium qubits via flux modulation of a tunable coupler bulletin of the American Physical Society *APS March Meeting* (Chicago) <https://meetings.aps.org/Meeting/MAR22/Session/T41.5> Number 5 in Session T41 (2022)
- [44] Zhang H, Ding C, Weiss D K, Schuster D I and Koch J 2022 Fast control of low-frequency fluxonium qubits bulletin of the American Physical Society *APS March Meeting* (Chicago) <https://meetings.aps.org/Meeting/MAR22/Session/T41.4> Number 4 in Session T41 (2022)