Hessian-based optimization of quantum dynamics under constrained control

Mogens Daalgaard,1 Felix Motzoi,2 Jesper Hassanriis Mohr Jensen,1 and Jacob Sherson1,*

1Department of Physics and Astronomy, Aarhus University, Ny Munkegade 120, 8000 Aarhus C, Denmark
2Forschungszentrum Jülich, Institute of Quantum Control (PGI-8), D-52425 Jülich, Germany

(Dated: May 2019)

Efficient optimization of quantum systems with large state or control spaces is a necessity for reaching fault tolerant thresholds. A standard tool for optimizing simulated quantum dynamics is the gradient-based GRAPE algorithm, which has been successfully applied in a wide range of different branches of quantum physics. In this work, we derive and implement exact 2nd order analytical Hessian of the coherent dynamics and find improvements compared to the approximate 2nd order BFGS standard optimization. We demonstrate performance improvements for both the best and average errors of constrained unitary gate synthesis on a circuit-QED system over a broad range of different gate durations.

I. INTRODUCTION

Nowadays, there exists a wide range of proposed technologies that utilize the potential of quantum mechanics in order to achieve improvements over their classical counterparts. These include quantum variational eigensolvers 11, 12, annealers 13, simulators 14, 15, Boltzmann machines 16, and perhaps most promising, the quantum computer 17. We may reach a point in time where the majority of these quantum-based technologies outperform their classical counterparts.

Reaching this quantum advantage requires, among others, substantial improvements in our ability to control the underlying quantum systems. On the theoretical side, quantum optimal control theory addresses this issue 18–20. Here optimization methods with respect to a chopped random basis (CRAB) 21, 22 and individual pulse amplitudes (Krotov) 23, 24 have been successful especially with respect to unconstrained quantum optimization, where trapping has been shown to occur rarely 25, 26.

One of the most used, and widely successful, algorithms within quantum optimal control theory is the gradient ascent pulse engineering (GRAPE) algorithm 27, 28. The original GRAPE algorithm used first-order approximated gradients in combination with steepest descent 29. Later, significant improvements were obtained when the analytical gradients were calculated and combined with Hessian approximation methods such as BFGS 30–32. GRAPE has been widely successful providing its use in Nuclear Magnetic Resonance 33–35, superconducting qubit circuits 36–38, spin chains 39–41, nitrogen vacancy centers 42, 43, and ultra cold atoms 44, 45. It has also become a standard integrated tool in many numerical packages aimed at quantum physicists 46–50. Further, one could mention the many extensions of GRAPE, which treat filtering 51–53, chopped random basis 54, experiment design 55, and open quantum systems 56. In addition, there exists many hybrid algorithms that combine GRAPE with, e.g. sequential updates 19, and global optimization algorithms 57, 58.

Machine-learning based control 59–64 could also improve the solution exploration of local quantum optimal control methods such as GRAPE, where initial steps towards a hybrid algorithm have already been taken in Ref. 54.

In this paper we show how GRAPE can further be enhanced to significantly increase its ability to reach high fidelity solutions, by incorporating the exact Hessian in the numerical optimization. We present here an efficient calculation of the Hessian and apply both gradient and Hessian-based optimization to unitary gate synthesis for superconducting circuits. A previous calculation of the Hessian made use of the auxiliary matrix method 55, 56. This approach requires the exponentiation of block matrices having three times the size of the Hilbert space, which leads to an unfavorable scaling with the Hilbert space. In this work, we present an new derivation that only requires exponentiation of matrices with the same size as the Hilbert space. In doing so, we show that every component needed to evaluate the gradient can be recycled to evaluate the Hessian and thus that calculating an element of the Hessian is not much more expensive than the gradient. These results enable us to demonstrate improvements with Hessian-based GRAPE in equal wall-time simulations without the explicit need for any code parallelization.

Besides calculating the Hessian, we further seek to benchmark analytical gradient- and Hessian-based numerical optimization within constrained physical settings. Physically realizable pulses are always constrained in amplitude due to either limitations on experimental equipment, but more often imposed to avoid undesirable processes such as leakage, ionization, heating, decoherence, and break down of theoretical models. For many quantum control problems, optimal solutions contain segments that lie on the boundary of the permissible region (see e.g. 57, 58). Under certain assumptions, this has even been proven to more generally occur 59. Therefore,
These include state-to-state transfer for pure quantum states and density matrices in closed quantum systems, as well as state-to-state transfer of density matrices in open quantum systems.

Replicating the target unitary, up to a global phase, is achieved by maximizing the fidelity

\[
F = \left| \frac{1}{\dim} \text{Tr}[U_P U_V^\dagger] \right|^2,
\]

where \( U = U_N U_{N-1} \ldots U_0 \) denotes the final unitary and \( P \) a projection into the subspace of interest, whose dimension is denoted \( \dim \). Equivalently, we seek to minimize the infidelity \( J = 1 - F \), which will serve as our cost function. We express the controls as a vector \( c = (c_{1,1}, c_{2,1}, \ldots, c_{N,1}, c_{1,2}, \ldots, c_{N,M}) \), where the first index denotes the discretized time and the second denotes different controls. Starting from an initial guess \( c_0 \), we seek to make incremental updates such that \( J(c_{n+1}) < J(c_n) \), with \( n \) denoting the iteration number. The incremental update is on the form

\[
c_{n+1} = c_n + \alpha_n p_n.
\]

Here \( p_n \) defines a search direction, while \( \alpha_n \) defines a step size. These incremental updates are illustrated with arrows in Fig. 1. The initial seed could either stem from an analytical ansatz, or based on a random guess to explore the optimization landscape. The latter approach is termed multistarting.

The search direction taken by GRAPE was originally proposed to be steepest descent, \( p_n = -\nabla J(c_n) \), where the gradient was approximated to first order in \( \Delta t \). Steepest descent only uses information about the first derivative and thus suffers from having less information about the curvature of the optimization landscape. Hence, it is often superior to use the second order derivatives i.e. Hessian matrix

\[
H = \begin{bmatrix}
\frac{\partial^2 J}{\partial c_{1,1}^2} & \frac{\partial^2 J}{\partial c_{1,1} \partial c_{1,1}} & \cdots & \frac{\partial^2 J}{\partial c_{1,1} \partial c_{N,M}} \\
\frac{\partial^2 J}{\partial c_{1,1} \partial c_{1,1}} & \frac{\partial^2 J}{\partial c_{2,1}^2} & \cdots & \frac{\partial^2 J}{\partial c_{2,1} \partial c_{N,M}} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial^2 J}{\partial c_{N,M} \partial c_{2,1}} & \frac{\partial^2 J}{\partial c_{N,M} \partial c_{2,1}} & \cdots & \frac{\partial^2 J}{\partial c_{N,M}^2}
\end{bmatrix}
\]

(3)

If one has access to the Hessian, a standard search direction is given by Newtons method \( p_n = -[H(c_n)]^{-1}\nabla J(c_n) \). If the second order derivatives are not available, one can approximate the Hessian \( B \approx H \) via the BFGS Hessian-approximation scheme that gradually builds \( B \) using only the gradient. The iterative BFGS update is of the form

\[
B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{B_k s_k s_k^T B_k^T}{s_k^T B_k s_k},
\]

(4)
where \( y_k = \nabla f(x_{k+1}) - \nabla f(x_k) \) and \( s_k = x_{k+1} - x_k \). Perhaps just as crucially, the derivation and use of an analytically exact gradient of the quantum dynamics used in GRAPE (see below) enables the precision needed for BFGS to outclass its first-order counterpart [18]. It is now the standard within quantum optimal control theory [19, 40, 41].

For optimization, we use a constrained optimization algorithm, Interior-Point [63, 65], implemented in MATLAB’s library fmicon [60]. Interior-Point includes the curvature of the optimization landscape either with the actual Hessian or the Hessian-approximation scheme BFGS. In this work, we found that Interior-Point performed better than other conventional optimization algorithms and therefore formed a better starting point for comparing gradient- and Hessian-based optimization. See Appendix [B] for a comparison with other alternatives to Interior-Point.

In Fig. 1b, we illustrate the difference between these methods for a simple two-dimensional optimization problem (see Appendix [D] for a technical explanation). The problem has a global minimum at \((c_1, c_2) = (0, 0)\). In addition there is also a local one above. We start the optimization near the two solutions and plot how each method moves through the optimization landscape. The two gradient-based approaches converge toward the local solution, while the Hessian-based method manages to avoid this solution and converge towards the global optimum. A possible explanation for this behavior is that Hessian-based optimization generally has more information about the landscape curvature. This can allow it to avoid suboptimal nearby solutions as the one depicted in Fig. 1b.

Moreover, Fig. 1b also illustrates how Hessian-based optimization generally uses fewer function evaluations [67]. If the optimization function is sufficiently simple near an optimal solution \( x^* \) a nearby initial guess (seed) \( x_0 \) will converge as \( ||x_{k+1} - x^*|| \leq C||x_k - x^*||^{q} \) where \( C \geq 0 \) for the optimization methods discussed here, with Hessian being quadratic \((q = 2)\) and BFGS being super-linear \((1 < q < 2)\) [67].

In the following we derive exact analytical expressions for both the gradient and Hessian of Eq. (1). The gradient has previously been calculated, see e.g. Ref. [18, 19], but reproduced here for completeness.

## A. Gradient

The first step is to express the derivative of the fidelity, Eq. (1), in terms of the derivative of the unitary

\[
\frac{\partial F}{\partial c_{j,k}} = \frac{2}{\text{dim}^2} \text{Re} \left[ \text{Tr} \left[ P \frac{\partial U^j}{\partial c_{j,k}} PV^\dagger \right] \text{Tr} \left[ VPU^\dagger P \right] \right].
\]  (5)

It depends on \( c_{j,k} \) and hence the derivative is

\[
\frac{\partial U}{\partial c_{j,k}} = U_N \ldots \frac{\partial U_j}{\partial c_{j,k}} \ldots U_1U_0 = U_{j+1}^R \frac{\partial U_j}{\partial c_{j,k}} U_{j-1}. \]  (6)

where we have defined the left and right unitaries as \( U^j = U_NU_{N-1} \ldots U_j \) and \( U_j^R = U_jU_{j-1} \ldots U_0 \), respectively. The left and right unitaries must be calculated for each time step, but this can be done efficiently since \( U_j^R = U_{j-1}^R U_j \) and \( U_j^R = U_jU_{j-1}^{-1} \). Thus the gradient calculations scales as \( O(N) \) in terms of the number of matrix multiplications rather than the intuitive \( O(N^2) \). This approach exploits that quantum propagation can always be decomposed smoothly into shorter, differential time segments.

In order to evaluate the derivative \( \frac{\partial U}{\partial c_{j,k}} \), we use the following expression for a general \( \eta \)-depend matrix \( X(\eta) \) [68]

\[
\frac{d}{d\eta} e^{X(\eta)} = \int_0^1 e^{\alpha X(\eta)} \frac{dX(\eta)}{d\eta} e^{(1-\alpha)X(\eta)} d\alpha.
\]  (7)

This integral can be solved in the eigenbasis \( \{|n\}\) of the Hamiltonian \( H(t_j) \). A direct calculation reveals

\[
\langle m | \frac{\partial U_j}{\partial c_{j,k}} | n \rangle = \langle m | H_k | n \rangle I(m, n). \]  (8)

Here we have defined

\[
I(m, n) = \begin{cases} 
-\frac{i\Delta \Delta e^{iE_m \Delta t}}{E_m - E_n}, & \text{if } m = n \\
\frac{-e^{-iE_m \Delta t} - e^{-iE_n \Delta t}}{E_m - E_n}, & \text{if } m \neq n,
\end{cases}
\]  (9)

where \( E_n \) denotes the eigenenergy of \(|n\rangle\). Since \( I \) only depends on the Hamiltonian \( H(t_j) \), it can be calculated in advance of evaluating Eq. (8). Note that one can also exploit the symmetry \( I(m, n) = I(n, m) \). After having computed these, one will have to switch back to the original basis.

## B. Hessian

We start by evaluating the derivative of Eq. (5), which reveals

\[
\frac{\partial^2 F}{\partial c_{i,k} \partial c_{j,k}} = \frac{2}{\text{dim}^2} \text{Re} \left[ \text{Tr} \left[ P \frac{\partial U^j}{\partial c_{i,k}} PV^\dagger \right] \text{Tr} \left[ VPU^\dagger P \right] \right] + \text{Tr} \left[ P \frac{\partial^2 U^j}{\partial c_{j,k} \partial c_{j,k}} PV^\dagger \right] \text{Tr} \left[ VPU^\dagger P \right].
\]  (10)

Here the first order derivatives of the unitary that appears in the last term of the above expression have already been calculated when evaluating the gradient.
What remains is hence to calculate the second derivatives of the unitary. Here we distinguish between two cases: when the derivatives are with respect to different time steps \((i \neq j)\) and the same time step \((i = j)\). For different time steps \((j > i)\) the second order derivatives become

\[
\frac{\partial^2 U}{\partial c_{i, k} \partial c_{j, k}} = U_{j+1} L \frac{\partial U}{\partial c_{j, k}} U_{j-1} R (U_{j} R) \frac{\partial U}{\partial c_{i, k}} U_{i-1} R. \tag{11}
\]

Here we have expressed the middle products of unitaries \(U_{j-1} \ldots U_{i+1}\) in terms of the right unitaries we defined when calculating the gradient. Evaluating the second order derivative of the unitary with respect to the same time step is similar to Eq. \([6]\)

\[
\frac{\partial^2 U}{\partial c_{j, k} \partial c_{j, k}} = U_{j+1} L \frac{\partial U}{\partial c_{j, k}} U_{j-1} R. \tag{12}
\]

Eq. (11) and (12) imply that the Hessian may be evaluated efficiently by recycling already calculated quantities. All that is left is to calculate the second order derivative of \(U_j\). This is done by differentiating Eq. (7)

\[
\frac{\partial^2 U_j}{\partial c_{j, k} \partial c_{j, k}} = -i \Delta t \int_0^1 \left( \frac{\partial}{\partial c_{j, k}} e^{\alpha X} \right) H_k e^{(1-\alpha)X} d\alpha
\]

\[
- i \Delta t \int_0^1 e^{\alpha X} H_k \left( \frac{\partial}{\partial c_{j, k}} e^{(1-\alpha)X} \right) d\alpha, \tag{13}
\]

where we have introduced the short-hand notation \(X = -i H \Delta t\). In the last integral above, we make the substitution \(1 - \alpha \rightarrow \alpha\) and then we insert Eq. (7) to obtain

\[
\frac{\partial^2 U_j}{\partial c_{j, k} \partial c_{j, k}} = (-i \Delta t)^2 \left( \int_0^1 d\alpha e^{\beta \alpha X} H_k e^{(1-\beta)X} \right)^2 + \int_0^1 d\alpha e^{(1-\alpha)X} H_k e^{\beta \alpha X} H_k e^{(1-\beta)X}. \tag{14}
\]

Similar to before, we evaluate the elements of the eigenbasis of \(H\). We also insert the identity \(1 = \sum_{n'} |n'\rangle \langle n'|\) to evaluate the middle part i.e. between the two Hamiltonians \(H_k\) and \(H_{k'}\) in the above expression. This reveals

\[
\langle m | \frac{\partial^2 U_j}{\partial c_{j, k} \partial c_{j, k}} | n \rangle = (-i \Delta t)^2 \sum_{n'} \left( H_k^{(m,n')} H_k^{(n',n)} \int_0^1 d\alpha e^{\beta \alpha X} H_k e^{(1-\beta)X} \right) + \int_0^1 d\alpha e^{(1-\alpha)X} H_k e^{\beta \alpha X} H_k e^{(1-\beta)X}, \tag{15}
\]

where we have defined \(H_k^{(m,n')} = \langle m | H_k | n' \rangle\) and \(\lambda_n = -i E_n \Delta t\). The two double integrals in the above expression turn out to be equivalent, which we calculate in Appendix [A]. This allows us to write

\[
\langle m | \frac{\partial^2 U_j}{\partial c_{j, k} \partial c_{j, k}} | n \rangle = \sum_{n'} \left( H_k^{(m,n')} H_k^{(n',n)} + H_k^{(m,n')} H_k^{(n',n)} \right) I(n, n', m). \tag{16}
\]

Here \(I(n, n', m)\) is given by Eq. (A9).
The intuitive scaling for calculating the analytical Hessian would be $O(N^3)$, since one would have to calculate the left, middle, and right sequence of unitaries for each pair of time steps $(i,j)$. However, having the left and right unitaries in advance reduces this to $O(N^2)$ via Eq. (11). Furthermore, Hessian-based optimization generally converges in fewer iterations than gradient-only, especially near the optimum where its convergence is quadratic. Last but not least, the higher accuracy of the calculation may help it altogether avoid local traps that can plague gradient-based QOCT methods. Thus, for a given amount of computational time, we may expect both the best-case and average-case quantum fidelities to be improved through the use of the exact analytical Hessian. In the following sections, we test and benchmark these claims on a standard quantum computational setup.

III. TRANSMON SYSTEM

As a testbed for the Hessian-based optimization, we have chosen two transmon qubits dispersively coupled through a linear microwave resonator activated by a microwave field. This type of setup is currently a frontier within superconducting circuit-based quantum information and could enable the realization of many of the quantum technologies outlined in the introduction. Such a setup has previously been studied for gradient and machine learning based optimal control. Transmons are insensitive to charge noise, but suffer from having relatively low anharmonicity. We therefore include a third level for each transmon qutrit. As a testbed for the Hessian-based optimization, we consider here the control problem outlined in the previous section, where we use piecewise constant pulses that consist of $\Delta t = 2.0 \text{ ns}$ segments, which is within the bandwidth of standard microwave pulse generators. We compare the two approaches, gradient-BFGS vs analytical Hessian-based optimization via the Interior-Point algorithm, over a wide range of different gate durations using multistarting. For each gate duration we use the two approaches to optimize the same 5000 seeds, which we draw uniformly at random. We plot the results in Fig. 2a-c, which respectively shows the best infidelity over the entire time, the mean infidelity per seed, and the mean wall time consumption per seed for each gate duration. The two approaches often find different solutions in the optimization landscape, despite starting from the same initial seed. This stresses the need for selecting the optimization algorithm with care, and also motivates the comparison given in Appendix C.

Fig. 2a, we see that when the gate duration increases above 200 ns the control problem becomes exactly solvable in the sense that the best infidelities become insignificantly low. In literature the minimal amount of time for which the problem becomes exactly solvable is termed the quantum speed limit. Here we see that the Hessian-based optimization is able to come closer to the (unknown) quantum speed limit relative to gradient-only. Even in the low infidelity regime (i.e. below $10^{-8}$), Hessian-based optimization reaches a few orders of magnitude improvement over gradient-only.

Fig. 2b, depicts the mean infidelity for the two algorithms. The two algorithms perform equally well on average at shorter gate durations. In contrast, Hessian-based optimization performs better at longer gate durations, with mean infidelities being 3-4 times lower than gradient-only optimization.
FIG. 3. (Color online) Distribution of infidelities (lower is better) at different gate durations. Blue depicts gradient-only optimization, while orange depicts Hessian-based. Brown depicts the overlap of the two distributions. Gate durations are depicted in the figure.

Fig. 2c depicts the mean wall time consumption for each gate duration. As already elaborated on, the computationally cost of the gradient scales linear $O(N)$ while the Hessian scales quadratically $O(N^2)$. However, Hessian-based optimization generally uses fewer iterations than BFGS. For this reason, Hessian-based optimization actually turns out to be faster than gradient-only at very short gate durations, where the Hessian is cheaper to calculate. For longer gate durations, however, Hessian-based optimization becomes 2-3 times slower than gradient-only.

In summary, Fig. 2 shows that there are two distinct regimes, where the analytical Hessian of the unitary dynamics is preferable to its BFGS-approximation, but for different reasons. For fewer than 50 ns (or 25 control steps), the two methods are able to fairly easily reach the global optimum, but the Hessian-based approach is faster in this task. Above this time, and especially near the quantum speed limit, we see that after an equal computational time, the exact Hessian will produce better gate infidelities, both on average and for the best case.

We plot in Fig. 3 histograms over the infidelity distributions for the same data presented in Fig. 2. Here the gradient-based results are depicted in blue, the Hessian-based with orange, and the mutual overlap with dark brown. At 50 ns in Fig. 3a, we see two almost identical distributions. In contrast, at larger gate durations we see that Hessian-based optimization generally performs better than gradient-only both in terms of the quality and quantity of the best solutions found.

From Figs. 2 and 3 we see that the two approaches, analytical Hessian-based and gradient-only-based optimizations often find different solutions although starting from the same initial seed. We also see that the Hessian-based solutions tend to be better i.e. reach lower infidelity compared to gradient-only, which is clear from the bimodal distributions in Fig. 3. We attribute this to the fact that Hessian-based optimization obtains more information about the curvature of the optimization landscape through the second derivative relative to gradient-only, which only has approximate knowledge of the second derivatives. This can lead gradient-only to slow down, or even be unable to converge at all. It may also cause trapping in local suboptimal solutions as illustrated in Fig. 1b.

Although the Hessian is more expensive to calculate than the gradient, we showed that Hessian-based optimization can be advantageous over gradient-only for even over 100 optimization parameters. However, for a much larger number of parameters, Hessian-based optimization may become too slow to use at one point, even if the final result would be better. Hence to include the Hessian is a question of balancing the quantity and quality of found solutions. When using only a relative few seeds (e.g. due to a large amount of optimization parameters or complicated state spaces that require more exploration), it may be very unlikely to reach near the global optimum for a practical wall time: In this case a gradient-BFGS approach may be expected to produce faster if less reliable results. Even so, most optimization tasks within quantum control have less than 125 optimization parameters, as any more would make experimental implementation and calibration infeasible. Hence, we believe that explicitly incorporating the analytical Hessian of the unitary dynamics is advantageous for typical quantum control.
Finally, one may link the apparent utility of the Hessian to the nature of the control landscape. Indeed, the smoothness of the landscape [7] has earlier been argued to provide a computational advantage for quantum optimization [12, 15, 76]. In the present work, we have explored the case where the cost functional is furthermore strongly constrained. Although such constraints formally remove the possibility of a global convergence [47, 78], in our context the functional smoothness can nonetheless provide a mechanism to greatly speed up convergence alongside a multistarting strategy with analog controls [59].

V. CONCLUSION

In this paper, we have obtained the Hessian of the unitary dynamics as an extension to the widely used gradient-based, quantum optimization algorithm GRAPE. Our calculations revealed that the Hessian may be computed efficiently, with a high level of re-usability of already-calculated quantities obtained when evaluating the gradient. We believe our efficient calculation is advantageous to previous proposals and it allowed us to demonstrate improvements over gradient-only optimization in equal wall-time simulations without any code parallelization. We optimized a CNOT gate on a circuit QED system consisting of two coupled transmon qubits. Here we demonstrated that a CNOT gate is in principle feasible using only a single control on one qubit driven cross-resonantly.

For the numerical optimization, we used an Interior-Point algorithm, with either an analytically-exact Hessian or the Hessian-approximation scheme BFGS that only relies on the gradient. We compared the two approaches, Hessian-based or gradient-only, for a wide range of different gate durations. Since the Hessian contains squared the number of elements of the gradient, it is more expensive to calculate per iteration. However, Hessian-based optimization generally uses fewer iterations to converge. Moreover, the convergence occurs with greater accuracy, which can improve the quality as well as the quantity of good solutions.

We have found that, depending on the number of controls, either the speed or fidelity of the solutions can be improved compared to the gradient. This appears to be generally true, although for very complex spaces where multistarting is not appropriate the gradient may be the only practical choice. Nonetheless, for over 100 controls, we were still able to collect statistics over many seeds. We found that below 25 controls, the Hessian enabled faster convergence towards extrema. For more controls (and for the experimentally most interesting regime near the quantum speed limit), incorporating the Hessian provided a higher percentage of good solutions, often accompanied by bimodal distributions pointing to avoiding local trapping. Thus, best-case error was also seen to be improved.

ACKNOWLEDGEMENTS

The authors would like to thank Jens Jakob Srensen for participating in constructive scientific discussions. This work was funded by the ERC, H2020 grant 639560 (MECTRL), the John Templeton, and Carlsberg Foundations. The numerical results presented in this work were obtained at the Centre for Scientific Computing, Aarhus phys.au.dk/forskning/cscaa.
Appendix A: Hessian calculations

The starting point of these calculations is Eq. (15), which can be written as

\[
\left< m \left| \frac{\partial^2 U_j}{\partial c_{j,k} \partial c_{j,k}} \right| n \right> = \sum_{n'} \left( H_k^{(m,n')} H_k^{(n',m)} I(n,n',m) + H_k^{(m,n')} H_k^{(n',m)} I(m,n,n') \right). \tag{A1}
\]

Here we have defined

\[
I(n,n',m) = (-i \Delta t)^2 e^{\lambda_n} \int_0^1 d\alpha \alpha e^{\alpha(\lambda_{n'} - \lambda_n)} \int_0^1 d\beta e^{\beta(\lambda_m - \lambda_{n'})}.
\tag{A2}
\]

In order to evaluate the above we must consider the the five different scenarios: \( m \neq n \neq n' \), \( m = n = n' \), \( m = n' \neq n \), and \( m = n \neq n' \). We start with the first one which is \( m \neq n \neq n' \). A direct calculation reveals

\[
I(n,n',m) = \frac{1}{E_m - E_{n'}} \left[ e^{-iE_m \Delta t} - e^{-iE_n \Delta t} \right] \left[ e^{-iE_{n'} \Delta t} - e^{-iE_n \Delta t} \right]. \tag{A3}
\]

Note that the above expression is invariant under any permutation of the indices. Similarly, the second one, which is \( m = n = n' \), gives

\[
I(n,n,n) = \frac{(-i \Delta t)^2}{2} e^{-iE_n \Delta t}. \tag{A4}
\]

The third one, which is \( m = n' \neq n \), gives

\[
I(n,m,m) = (-i \Delta t)^2 e^{\lambda_n} \int_0^1 d\alpha \alpha e^{\alpha(\lambda_{m'} - \lambda_n)}. \tag{A5}
\]

The antiderivative of \( xe^{kx} \) is \( \frac{(kx-1)}{k^2}e^{kx} \), which can be used to evaluate the above

\[
I(n,m,m) = \frac{-i \Delta t (E_m - E_n) - 1}{(E_m - E_n)^2} e^{-iE_m \Delta t} + e^{-iE_n \Delta t}. \tag{A6}
\]

The fourth one is \( m \neq n' = n \)

\[
I(n,n,n) = \frac{e^{-iE_m \Delta t} + (-i \Delta t |E_m - E_n| - 1)}{(E_m - E_n)^2} e^{-iE_n \Delta t}. \tag{A7}
\]

And the last one, which is \( m = n \neq n' \)

\[
I(n,n',n) = \frac{(-i \Delta t |E_n - E_{n'}| - 1)}{(E_n - E_{n'})^2} e^{-iE_n \Delta t} + e^{-iE_{n'} \Delta t}. \tag{A8}
\]

We can express the above results by using Eq. (A9)

\[
I(n_1, n_2, n_3) = \begin{cases} 
\frac{1}{E_{n_1} - E_{n_2}} [I(n_3, n_2) - I(n_2, n_1)], & \text{if } n_1 \neq n_2 \neq n_3 \\
\frac{-i \Delta t}{2} I(n_1, n_1), & \text{if } n_1 = n_2 = n_3 \\
\frac{1}{E_{n_1} - E_m} [I(n, m) - I(m, m)], & \begin{cases} 
\text{if } n_1 \neq n_2 = n_3 = m \\
\text{or if } n_2 \neq n_1 = n_3 = m \\
\text{or if } n_3 \neq n_1 = n_2 = m. 
\end{cases}
\end{cases} \tag{A9}
\]

We may evaluate the above integral efficiently by storing the already calculated integrals \( I(m,n) \) from Eq. (9). Also note that above integral is independent of the order of coefficients \( n_1, n_2, n_3 \). This implies that we may calculate the above efficiently and in advance of evaluating the matrix elements of the second derivative. Since \( I(n_1, n_2, n_3) \) is independent of the order of the indices, we may also take the integral out of a parenthesis in Eq. (A1) and hence we obtain Eq. (10).
Appendix B: Benchmarking different optimization algorithms

In the main text we considered synthesizing a CNOT gate using piecewise constant pulses with \( \Delta t = 2.0 \text{ ns} \) using the GRAPE algorithm. GRAPE relies on a numerical optimization algorithm at its backend, for instance in the main text we used Interior-Point that can either be supplied with the gradient and the Hessian-approximation scheme BFGS or the exact Hessian.

To justify this specific choice we benchmark Interior-Point with other conventional optimization algorithms. Another choice, also implemented in MATLABs \texttt{fmincon} \cite{fmincon} for constrained optimization, is the Trust-Region-Reflective algorithm \cite{trust-region-reflective}. Similar to Interior-Point, Trust-Region-Reflective can be supplied with either the gradient and optionally the Hessian. We also benchmark against an unconstrained optimization algorithm quasi-Newton \cite{quasi-Newton} implemented in MATLABs \texttt{fminunc}, where we instead impose constraint-bounds by adding a penalty term to the cost-function. We choose a quadratic penalty function that is zero inside the admissible region and grows quadratically outside \( J_{\text{penalty}} = \sigma (\Omega - \Omega_{\text{min}}/\Omega_{\text{max}})^2 \), where we set the penalty factor to \( \sigma = 10^5 \).

For the comparison we consider the same control problem as in the main text with gate duration \( T = 200 \text{ ns} \). We let each optimization algorithm optimize the same 300 random seeds, which is drawn uniformly at random. The results is plotted in Fig. 4 where we plot the infidelity for each seed as a function of wall time consumption, which we limit to 1000 s. From the figure we see that Interior-Point generally converges faster and at lower infidelity solutions. This justifies our choice of using Interior-Point for the results presented in the main text.

Appendix C: Circuit QED calculations

The following derivation to some extent resembles the one given in Ref. \cite{ciruitQED}. The starting point is the drift Hamiltonian given by Eq. \eqref{eq:17}. Here we eliminate the cavity by going to a frame rotating at \( \omega_r \) via \( R = \exp \left[ -i \omega_r (b_1^\dagger b_1 + b_2^\dagger b_2 + a^\dagger a) \right] \). This gives

\[
H = \sum_{j=1,2} \Delta_j b_j^\dagger b_j + \frac{\delta_j}{2} b_j^\dagger b_j^\dagger b_j b_j - 1 + \sum_{j=1,2} g_j (a b_j^\dagger + a^\dagger b_j),
\]

where \( \Delta_j = \omega_j - \omega_r \). Then we do a Schrieffer-Wolff transformation \cite{Schrieffer-Wolff} using \( S = \sum_{j=1,2} \frac{g_j}{\Delta_j} (a b_j^\dagger - a^\dagger b_j) \) in order to eliminate the cavity. The resulting Hamiltonian, when any constant energy shifts have been removed and the cavity neglected, is
Here we see that the transmon-cavity coupling has been replaced with an effective transmon-transmon coupling where
\[ J = \frac{g J (\Delta_1 - \Delta_2)}{2} \] and \( \tilde{\omega}_j = \omega_j + \frac{g_j^2}{2} \) is now the dressed transmon state. The last step when transforming Eq. (C2) into Eq. (18) is doing a second rotation \( R' = \exp\left[-i \tilde{\omega}_2 (b_1^\dagger b_1 + b_2^\dagger b_2)\right] \) such that the detuning becomes \( \Delta = \tilde{\omega}_2 - \tilde{\omega}_1 \). We also add a direct drive on the first transmon \( H_c(t) = \Omega(t)(b_1^\dagger + b_1) \) see e.g. Ref. 73.

Appendix D: Two-level example

In Fig. 1b, we illustrated three optimization methods based on gradient descent and Interior-Point with either BFGS or Hessian for a two-dimensional optimization problem. Here we have implemented a simple gradient descent and BFGS. We briefly elaborate on what this Figure depicts. We consider a two-level Hamiltonian \( H(t) = \sigma_x + c(t)\sigma_z \), with the goal of synthesizing a \( X \) gate. We limit the control to two steps \( N = 2 \), which reveals an analytical solution at \( T_{QSL} = \pi/2 \) with \( c_1, c_2 = 0 \) and \( c_1, c_2 = 0 \). At \( T = 3T_{QSL} \) the same solution is still optimal, but now several other solutions emergence in the optimization landscape, an effect also studied in Ref. [82]. For instance the figure depicts a suboptimal solution at \( (c_1, c_2) = (0.000, 2.285) \). We start the optimization near the two optima at \( (c_1, c_2) = (-0.915, 2.251) \) and plot the subsequent optimization. The two gradient-based optimization approaches fall into the nearest trap (i.e. suboptimal solution), while the Hessian optimization manages to avoid this solution. We attribute this result to the fact that the Hessian-based optimization has more information about the landscape curvature, which enables it to avoid the local trap and instead reach the optimal solution. The reader should of course keep in mind that this is but one example.

[1] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O’Brien, Nature Communications 5, 4213 (2014).
[2] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, J. M. Brink, J. M. Chow, and J. M. Gambetta, Nature 549, 242 (2017).
[3] M. W. Johnson, M. H. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Kerckhaert, J. Johansson, P. Bunyk, et al., Nature 473, 194 (2011).
[4] A. A. Houck, H. E. Türeci, and J. Koch, Nature Physics 8, 292 (2012).
[5] I. Bloch, J. Dalibard, and S. Nascimbene, Nature Physics 8, 267 (2012).
[6] M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchytskyy, and R. Melko, Physical Review X 8, 021050 (2018).
[7] M. A. Nielsen and I. Chuang, “Quantum computation and quantum information.” (2002).
[8] J. Werschnik and E. Gross, Journal of Physics B: Atomic, Molecular and Optical Physics 40, R175 (2007).
[9] C. Brif, R. Chakrabarti, and H. Rabitz, New Journal of Physics 12, 075008 (2010).
[10] S. J. Glaser, U. Boscain, T. Calarco, C. P. Koch, W. Köckenberger, R. Kosloff, I. Kuprov, B. Luy, S. Schirmer, T. Schulte-Herbrüggen, et al., The European Physical Journal D 69, 279 (2015).
[11] P. Doria, T. Calarco, and S. Montangero, Physical review letters 106, 190501 (2011).
[12] T. Caneva, T. Calarco, and S. Montangero, Physical Review A 84, 022326 (2011).
[13] J. P. Palao and R. Kosloff, Physical review letters 89, 188301 (2002).
[14] J. P. Palao and R. Kosloff, Physical Review A 68, 062308 (2003).
[15] H. A. Rabitz, M. M. Hsieh, and C. M. Rosenthal, Science 303, 1998 (2004).
[16] N. Rauch, M. M. Müller, T. Calarco, and S. Montangero, Physical Review A 92, 062343 (2015).
[17] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, Journal of magnetic resonance 172, 296 (2005).
[18] P. De Fouquieres, S. Schirmer, S. Glaser, and I. Kuprov, Journal of Magnetic Resonance 212, 412 (2011).
[19] S. Machnes, U. Sander, S. J. Glaser, P. de Fouquieres, A. Gruslys, S. Schirmer, and T. Schulte-Herbrüggen, Physical Review A 84, 022305 (2011).
[20] R. Fletcher, Practical methods of optimization (John Wiley & Sons, 2013).
[21] Z. Tošner, S. J. Glaser, N. Khaneja, and N. C. Nielsen, The Journal of Chemical Physics 125, 184502 (2006).
[22] J. Zhang, T.-C. Wei, and R. Laflamme, Physical review letters 107, 010501 (2011).
[23] K. Kolzwar, S. Ehui, T. E. Skinner, S. J. Glaser, and B. Luy, Journal of Magnetic Resonance 225, 142 (2012).
[24] K. R. K. Rao, T. Mahesh, and A. Kumar, Physical Review A 90, 012306 (2014).
[25] J. J. Søgaard, J. S. Nyemann, F. Motzoi, J. Sherson, and T. Vosegaard, arXiv preprint arXiv:1912.05862 (2019).
[26] F. Motzoi, J. M. Gambetta, P. Rebentrost, and F. K. Wilhelm, Physical review letters 103, 110501 (2009).
