Discrete Adjoints for Accurate Numerical Optimization with Application to Quantum Control

N. Anders Petersson[a,*], Fortino M. Garcia[b], Austin E. Copeland[c], Ylva L. Rydin[d], Jonathan L. DuBois[e]

[a] *Center for Applied Scientific Computing, LLNL, Livermore, CA 94550, USA.*
[b] *Department of Applied Mathematics, CU, Boulder, CO 80309, USA*
[c] *Department of Mathematics, SMU, Dallas, TX 75205, USA*
[d] *Department of Information Technology, UU, 751 05 Uppsala, SWEDEN*
[e] *Quantum Simulations Group, LLNL, Livermore, CA 94550, USA*

Abstract

We consider optimizing control functions for realizing logical gates in a closed quantum systems, where the evolution of the state vector is governed by the time dependent Schrödinger equation. This is an optimal control problem where the objective function consists of two parts: the infidelity of the quantum gate and an integral in time over highly energetic states. We write Schrödingers equation as a Hamiltonian system in terms of the real and imaginary parts of the state vector and discretize the system with the Strömer-Verlet scheme, which is a partitioned Runge-Kutta method. We develop a compatible scheme for the adjoint differential equation, which allows the gradient of the discretized objective function to be calculated exactly. This allows all components of the gradient to be calculated at the cost of solving two Schrödinger systems, independently of the number of parameters in the control functions. The control function are expanded in a series of B-spline basis functions with built-in carrier waves. The interior point L-BFGS algorithm from the IPOPT package is used to minimize the objective function, subject to amplitude constraints on the parameter vector. The method is applied to Hamiltonians that model the dynamics of a super-conducting

*Corresponding author

Email addresses: petersson1@llnl.gov (N. Anders Petersson), fortino.garcia@colorado.edu (Fortino M. Garcia), acopeland@mail.smu.edu (Austin E. Copeland), ylva.rydin@it.uu.se (Ylva L. Rydin), dubois9@llnl.gov (Jonathan L. DuBois)

qudit. By calculating the spectrum of the Hessian at the optima we find that many eigenvalues are small and that a handful are negative, indicating that the optimization problem is non-convex.

Keywords: Optimal control, Partitioned Runge-Kutta method, Discrete adjoint, Quantum computing

---

1. Introduction

A key challenge for realizing the potential of quantum computing lies in determining the most efficient and accurate route to controlling the quantum states in a quantum device. This challenge stems from the fact that current quantum computing systems, unlike classical computers, do not have a fixed set of logical gates predetermined in hardware. Instead, the execution of a quantum algorithm is carried out by first devising a set of classical control functions that are then applied to the quantum computing hardware to guide the quantum states through a series of quantum logical operations [13]. Reducing the time required for a quantum gate to be realized is critical for near-term quantum computing because it enables the computation to finish before the quantum state collapses to a classical state, rendering the results meaningless. To mitigate this problem, quantum optimal control techniques have been developed to produce customized control pulses that minimize the execution time for complicated gates that directly map onto a physical system [18].

Optimizing the control functions in a quantum system is an optimal control problem where the objective function contains two main ingredients: the infidelity of the quantum gate and an integral in time over highly energetic states [10]. In this paper we develop a partitioned Runge-Kutta method for discretizing Schrödinger's equation and a compatible time-stepping scheme for the adjoint differential equation. These schemes allow the gradient of the objective function to be calculated exactly. As a result, the control functions can be optimized using exact gradients, which increases the robustness and convergence of the optimization algorithm compared to methods based on approximate gradients, e.g. GRAPE [9, 10, 11].

Logical gates in a closed quantum system can be viewed as linear reversible mappings, $|\boldsymbol{\psi''}\rangle = V|\boldsymbol{\psi'}\rangle$, from an initial state $|\boldsymbol{\psi'}\rangle$ to a final state $|\boldsymbol{\psi''}\rangle$, where the reversibility implies that the mapping V must be unitary, $V^\dagger V = I$. To introduce the quantum control problem, we start by dis-

cussing the case where the unitary transformation is defined in the entire N-dimensional state space, such that V can be represented by a unitary $N \times N$ matrix; a more general case is described in Section 2. In the following, we will replace the ket-notation [13] of the state vector $|\boldsymbol{\psi}\rangle = \psi^{(0)}|0\rangle + \psi^{(1)}|1\rangle + \ldots + \psi^{(N-1)}|N-1\rangle$ by the vector notation $\boldsymbol{\psi} = \psi^{(0)}\boldsymbol{e}_0 + \psi^{(1)}\boldsymbol{e}_1 + \ldots + \psi^{(N-1)}\boldsymbol{e}_{N-1}$, which is more common in the computational mathematics literature[1]. The elements in the state vector are complex probability amplitudes and the squared magnitude of the amplitudes sum to unity, i.e., $\|\boldsymbol{\psi}\|_2^2 = 1$.

To account for all admissible initial data in the Hilbert space $\mathbb{C}^N$, we consider the evolutions from the canonical basis vectors $\boldsymbol{e}_j$, for $j = 0, 1, \ldots, N-1$. The time-dependent control functions are expanded in terms of a finite number of basis functions, such that the control functions are determined by the finite-dimensional parameter vector $\boldsymbol{\alpha} \in \mathbb{R}^D$. This leads to Schrödinger's equation in matrix form for the $N \times N$ complex-valued solution operator matrix $U(t, \boldsymbol{\alpha})$:

$$\frac{\partial U}{\partial t} + iH(t, \boldsymbol{\alpha})U = 0, \quad 0 \le t \le T, \quad U(0, \boldsymbol{\alpha}) = I_N, \quad H^\dagger = H. \tag{1}$$

Here, I_N is the $N \times N$ identity matrix and $H(t, \boldsymbol{\alpha})$ is the Hamiltonian matrix, in which the time-dependence is parameterized by $\boldsymbol{\alpha}$. As a result, the solution operator matrix depends implicitly on $\boldsymbol{\alpha}$ through Schrödinger's equation. Because Schrödinger's equation is linear, the solution for general initial conditions satisfies $\boldsymbol{\psi}(t, \boldsymbol{\alpha}) = U(t, \boldsymbol{\alpha})\boldsymbol{\psi}(0, \boldsymbol{\alpha})$.

The goal of the quantum control problem is to determine the parameter vector $\boldsymbol{\alpha}$ such that the time-dependence in the Hamiltonian matrix leads to a solution of Schrödinger's equation that minimizes the difference between the target gate matrix V and $U(t, \boldsymbol{\alpha})$ at the final time $t = T > 0$. The overlap between the two matrices can be measured by

$$R_V(U_T(\boldsymbol{\alpha})) := \langle U_T(\boldsymbol{\alpha}), V \rangle_F = \mathrm{tr}\Big(U_T^\dagger(\boldsymbol{\alpha})V\Big), \quad U_T(\boldsymbol{\alpha}) := U(T, \boldsymbol{\alpha}). \tag{2}$$

Here, $\langle \cdot\,, \cdot \rangle_F$ denotes the Fröbenius matrix scalar product. Because U_T and V are unitary, $|R_V| \le N$. Note that the magnitude of R_V is independent of the global phase in U_T. In particular, $|R_V(U_T)| = N$ for $U_T = \exp(i\theta)V$,

[1]Here, $\boldsymbol{e}_j$ represents the jth canonical basis vector, in which the jth element is one and all other elements are zero.

where $\theta \in \mathbb{R}$. Based on the overlap between $U_T(\boldsymbol{\alpha})$ and V, their distance can be measured by the gate infidelity,

$$\mathcal{J}_0(U_T(\boldsymbol{\alpha})) = 1 - \frac{1}{N^2} \left| R_V(U_T(\boldsymbol{\alpha})) \right|^2 \geq 0. \tag{3}$$

The quantum control problem is a constrained optimization problem where, in the basic setting, the gate infidelity (3) is minimized under the constraints that the solution operator matrix satisfies Schrödinger's equation (1) and the amplitudes of the control functions (determined by the parameter vector $\boldsymbol{\alpha}$) do not exceed prescribed limits. For a discussion of the solvability of the quantum control problem, see for example Borzi et al. [2].

While not a restriction of our approach, we exemplify our technique on Hamiltonians that model the dynamics of a super-conducting qudit (a qubit with more than two energy levels). We represent the state vector in the energy basis in which the system Hamiltonian matrix is diagonal. To slow down the time scales in the state vector, the problem is transformed to a rotating frame of reference (see Appendix A) in which the Hamiltonian matrix satisfies

$$H(t, \boldsymbol{\alpha}) = H_d + H_c(t, \boldsymbol{\alpha}), \quad H_d = -\frac{\xi_a}{2} a^\dagger a^\dagger a a, \tag{4}$$

$$H_c(t, \boldsymbol{\alpha}) = p(t, \boldsymbol{\alpha})(a + a^\dagger) + iq(t, \boldsymbol{\alpha})(a - a^\dagger). \tag{5}$$

Here, H_d and H_c are called the drift and control Hamiltonians. The terms $p(t, \boldsymbol{\alpha})$ and $q(t, \boldsymbol{\alpha})$ are real-valued control functions of time that depend on the parameter vector $\boldsymbol{\alpha}$, ξ_a is a real constant; a and $a^\dagger$ are the annihilation and creation matrices (see Appendix A).

Several numerical methods for the quantum control problem are based on the GRAPE algorithm [9]. Here the Schrödinger equation is discretized in time using the lowest order Magnus scheme [8], in which the Hamiltonian matrix is evaluated at the midpoint of each time step. A stair-step approximation of the control functions is imposed such that each control function is constant within each time step. With Q control functions, M time steps of size h, the control functions are thus described by M times Q parameters $\alpha_{j,k}$. The time-step propagator in the Magnus method is of the form $\exp(-ih(H_0 + \sum_k \alpha_{j,k} H_k))$. In general, the matrices H_0 and H_k do not commute, leading to an integral expression for the derivative of the propagator with respect to the parameters, which is needed for computing the gradient of the objective function. In the GRAPE method, the integral expression is

approximated by the first term in its Taylor series expansion, leading to an approximate gradient that is polluted by an $\mathcal{O}(h^2)$ error. As the gradient becomes smaller during the minimization, the approximation error will eventually dominate the numerical gradient, which may lead to a slow-down in convergence near an optima. A more accurate way of numerically evaluating the derivative of the time-step propagator can be obtained by retaining more terms in the Taylor series expansion, or by using a matrix commutator expansion [4]. However, these techniques still impose a stair-step approximation of the control functions.

In the GRAPE algorithm, all components of the gradient can be computed by propagating the Schrödinger system forwards in time followed by back-propagating the corresponding adjoint system; the gradient of the gate infidelity with respect to all control function parameters can then be calculated from the state and adjoint variables at each time step. Alternatively, the gradient of the objective function can be calculated by differentiating Schrödinger's equation with respect to each parameter in the control function, leading to a differential equation for each component of the gradient of the state vector. This approach, implemented in the GOAT algorithm [12], allows the gradient of the objective function to be calculated exactly, but requires $(D+1)$ Schrödinger systems to be solved when the control functions depend on D parameters. This makes the method computationally expensive when the number of parameters is large.

The GRAPE algorithm has been generalized to optimize more general objective functions that include a combination of the gate infidelity, integrals penalizing occupation of "forbidden states" (see Section 2) and terms for imposing smoothness and amplitude constraints on the control functions [10]. This approach is based on approximate automatic differentiation for computing the gradient of the objective function. It is approximate because it relies on the same inexact derivative of the time-step propagator as in the GRAPE algorithm. This leads to numerical gradients that are polluted by an $\mathcal{O}(h^2)$ error.

Using a stair-stepped approximation of the control functions often leads to a large number parameters, which may hamper the convergence of the GRAPE algorithm. The total number of parameters can be reduced by instead expanding the control functions in terms of basis functions. The stair-step approximation can then be imposed on each basis function. By using the chain rule, the gradient from the GRAPE algorithm can then be used to calculate the gradient with respect to the coefficients in the basis function

expansion. This approach is implemented in the GRAFS algorithm [11], where the control functions are expanded in terms of Slepian sequences.

Gradient-free optimization methods can also be applied to the quantum optimal control problems. These methods do not rely on the gradient to be evaluated and are therefore significantly easier to implement. However, the convergence of these methods is usually much slower than for gradient-based techniques, unless the number of control parameters is very small. One example of a gradient-free methods for quantum optimal control is the CRAB algorithm [3].

In this paper, we develop a gradient-based method that applies to general parameterizations of the control functions, without imposing stair-stepped approximations. To ensure proper convergence of the optimization algorithm, we develop a "discretize before optimize" approach in which we first discretize the objective function and then derive a compatible time-stepping scheme for the adjoint differential equation, such that the gradient of the discretized objective function can be evaluated exactly.

Our technique is based on formulating Schrödinger's equation as a time-dependent Hamiltonian system in terms of the real and imaginary parts of the state vector. The Hamiltonian structure of the system is preserved by discretizing it with the Strömer-Verlet method, which is a partitioned Runge-Kutta scheme [8]. The final time step in the numerical solution is used for evaluating the gate infidelity, but the discretized objective function can also include integrals over time of the numerical solution (see Section 2). Inspired by modern discretization techniques for optimal control [15, 16, 17], we develop an adjoint partitioned Runge-Kutta scheme for integrating the adjoint differential equation, which allows the gradient of the discretized objective function to be calculated exactly. Similar to the GRAPE method, all components of the gradient can be computed by solving two Schrödinger systems, regardless of the number of parameters in the control functions. The time-history of the solution operator matrix is needed for accumulating the gradient during the reversed time stepping of the adjoint equation. Storing this data can be avoided by exploring the time-reversibility of the Strömer-Verlet scheme. In this case, three Schrödinger systems must be solved to evaluate all components of the gradient.

Many parameterizations of quantum control functions have been proposed in the literature, for example cubic splines [6], Gaussian pulse cascades [5], Fourier expansions [20] and Slepian sequences [11]. The optimization method we develop in this paper applies to any of the aforementioned expansions, but

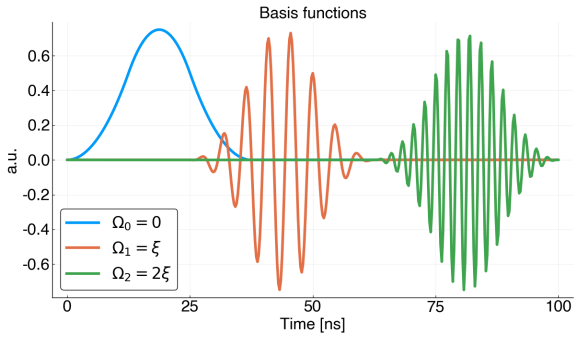

Figure 1: An example of three quadratic B-spline basis functions with carrier wave frequencies $(0, \xi, 2\xi)$.

we choose to exemplify it on control functions that are expanded in terms of B-spline basis functions with carrier waves, see Figure 1. An intuitive motivation of this approach is that transitions between the states in a quantum system are triggered by resonance at frequencies which often can be determined by inspection of the Hamiltonian matrix. By expanding the control functions in terms of a set of B-spline basis functions with carrier waves, we can control the envelope of each carrier wave. Using this approach, we find that a moderate number of basis functions often suffices for realizing highly accurate gates, see Section 5 for details.

The remainder of the paper is organized as follows. In Section 2, we generalize the optimization problem to the case of target gates that are defined in a subspace of the entire state space. In Section 3, we first introduce the real-valued formulation of Schrödinger's equation, followed by a presentation of the symplectic Strömer-Verlet time-stepping method, written as a partitioned Runge-Kutta scheme. To achieve an exact gradient of the discrete objective function, in Section 4 we derive the discrete adjoint time integration method, which turns out to be a related partitioned Runge-Kutta method. The solution of the discrete adjoint equation is used to efficiently calculate all components of the gradient of the discrete objective function. The parameterization of the control functions using B-splines with a carrier wave is presented in Section 5. Section 6 presents numerical verifications and an example of how the proposed technique can be combined with the interior point L-BFGS algorithm [14] from the IPOPT package [19] to realize a multi-level qudit gates. Important properties of the optima are exposed by analyzing

the eigenvalues of the Hessian. Concluding remarks are given in Section 7.

2. Generalized gates

In quantum computing applications it is common to define gate transformations in a subspace of the entire (possibly infinite dimensional) state space, in which the evolution of higher energy states is not relevant for the gate transformation, but if left uncontrolled, may lead to leakage of probability. In the following, let the subspace of interest contain $E > 0$ "essential" states and let $G = N - E \geq 0$ denote the number of "guard" states. The guard states that correspond to the highest energy levels in the model are often called "forbidden" states [10].

In the case of one qudit oscillator, we can always order the elements in the state vector such that they correspond to increasing energy levels. The Schrödinger equation governs the evolution of all energy levels in the state vector, including the guard levels, but the unitary gate transformation is only defined in the subspace of the essential states, which is spanned by the canonical basis vectors $\boldsymbol{e}_j \in \mathbb{R}^N$, for $j = 0, 1, \ldots, E-1$.

Let the state vector $\boldsymbol{\psi}_j(t, \boldsymbol{\alpha}) \in \mathbb{C}^N$ satisfy the Schrödinger equation,

$$\frac{d\boldsymbol{\psi}_j}{dt} + iH(t, \boldsymbol{\alpha})\boldsymbol{\psi}_j = 0, \quad 0 \leq t \leq T, \quad \boldsymbol{\psi}_j(0, \boldsymbol{\alpha}) = \boldsymbol{e}_j, \tag{6}$$

for $j = 0, 1, \ldots, E-1$. The solution operator matrix $U(t, \boldsymbol{\alpha})$ and the target gate matrix V become rectangular with N rows and E columns,

$$U(t, \boldsymbol{\alpha}) = [\boldsymbol{\psi}_0(t, \boldsymbol{\alpha}), \boldsymbol{\psi}_1(t, \boldsymbol{\alpha}), \ldots, \boldsymbol{\psi}_{E-1}(t, \boldsymbol{\alpha})], \quad V = [\boldsymbol{d}_0, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_{E-1}]. \tag{7}$$

We can construct a diagonal matrix P that projects the state vector onto the subspace spanned by the essential states,

$$P = \sum_{j=1}^{E} \boldsymbol{e}_j \boldsymbol{e}_j^\dagger, \quad P^2 = P.$$

The requirement that the unitary transformation is only defined in the subspace of the essential states can be formulated as $PV = V$.

The matrix overlap function $R_V(U_T)$ in (3) generalizes in a straightforward way to unitary gates that are defined in the subspace, resulting in the

gate infidelity function

$$\mathcal{J}_1(U_T(\boldsymbol{\alpha})) = 1 - \frac{1}{E^2}\left|S_V(U_T(\boldsymbol{\alpha}))\right|^2, \quad S_V(U_T(\boldsymbol{\alpha})) = \sum_{j=0}^{E-1} \langle \boldsymbol{\psi}_j(T,\boldsymbol{\alpha}), \boldsymbol{d}_j \rangle_2 , \tag{8}$$

where $\langle \cdot, \cdot \rangle_2$ is the ℓ_2 vector scalar product. The population of the guard states can be measured by the objective function

$$\mathcal{J}_2(U(\cdot,\boldsymbol{\alpha})) = \frac{1}{T}\int_0^T \sum_{j=0}^{E-1} \langle \boldsymbol{\psi}_j(t,\boldsymbol{\alpha}), W\boldsymbol{\psi}_j(t,\boldsymbol{\alpha}) \rangle_2 \, dt. \tag{9}$$

Here, W is a diagonal $N \times N$ positive semi-definite weight matrix. The elements in W are zero for all essential states and are positive for the guard states. The elements are usually increased towards the highest energy levels in the model.

For the quantum control problem with guard states, we formulate the optimization problem as

$$\min_{\boldsymbol{\alpha}} \mathcal{G}(\boldsymbol{\alpha}) := \mathcal{J}_1(U_T(\boldsymbol{\alpha})) + \mathcal{J}_2(U(\cdot,\boldsymbol{\alpha})), \tag{10}$$

$$\frac{\partial U}{\partial t} + iH(t,\boldsymbol{\alpha})U = 0, \quad 0 \le t \le T, \quad U(0,\boldsymbol{\alpha}) = [\boldsymbol{e}_0, \boldsymbol{e}_1, \ldots, \boldsymbol{e}_{E-1}]. \tag{11}$$

$$\alpha_{min} \le \alpha_q \le \alpha_{max}, \quad q = 1, 2, \ldots, D. \tag{12}$$

In the special case of zero guard states, $\mathcal{J}_2(U) = 0$ because $W = 0$. Thus, the above formulation applies to both the cases with and without guard states, i.e., when $G = N - E \ge 0$.

3. Real-valued formulation

A real-valued formulation of Schrödinger's equation (6) is given by

$$\begin{bmatrix} \dot{\boldsymbol{u}} \\ \dot{\boldsymbol{v}} \end{bmatrix} = \begin{bmatrix} S(t) & -K(t) \\ K(t) & S(t) \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{bmatrix} =: \begin{bmatrix} f^u(\boldsymbol{u},\boldsymbol{v},t) \\ f^v(\boldsymbol{u},\boldsymbol{v},t) \end{bmatrix}, \quad \begin{bmatrix} \boldsymbol{u}(0) \\ \boldsymbol{v}(0) \end{bmatrix} = \begin{bmatrix} \boldsymbol{g}^u \\ \boldsymbol{g}^v \end{bmatrix}, \tag{13}$$

where,

$$\boldsymbol{u} = \mathrm{Re}(\boldsymbol{\psi}), \quad \boldsymbol{v} = -\mathrm{Im}(\boldsymbol{\psi}), \quad K = \mathrm{Re}\,(H), \quad S = \mathrm{Im}\,(H),$$

Because the matrix H is Hermitian, $K^T = K$ and $S^T = -S$ (note that the matrix S is unrelated to the matrix overlap function S_V). The real-valued formulation of Schrödinger's equation is a time-dependent Hamiltonian system corresponding to the Hamiltonian functional,

$$\mathcal{H}(\boldsymbol{u}, \boldsymbol{v}, t) = \boldsymbol{u}^T S(t)\boldsymbol{v} + \frac{1}{2}\boldsymbol{u}^T K(t)\boldsymbol{u} + \frac{1}{2}\boldsymbol{v}^T K(t)\boldsymbol{v}. \tag{14}$$

In general, $S(t) \neq 0$, which makes the Hamiltonian system non-separable.

In terms of the real-valued formulation, the columns of the solution operator matrix in (7) satisfies $U = [\boldsymbol{u}_1 - i\boldsymbol{v}_1,\ \boldsymbol{u}_2 - i\boldsymbol{v}_2,\ \ldots, \boldsymbol{u}_E - i\boldsymbol{v}_E]$. Here, $(\boldsymbol{u}_j, \boldsymbol{v}_j)$ satisfy (13) subject to the initial conditions $\boldsymbol{g}_j^u = \boldsymbol{e}_j$ and $\boldsymbol{g}_j^v = \boldsymbol{0}$. The columns in the target gate matrix V correspond to

$$V = [\boldsymbol{d}_1^u - i\boldsymbol{d}_1^v,\ \boldsymbol{d}_2^u - i\boldsymbol{d}_2^v,\ \ldots, \boldsymbol{d}_E^u - i\boldsymbol{d}_E^v], \quad \boldsymbol{d}_j^u = \mathrm{Re}(\boldsymbol{d}_j), \quad \boldsymbol{d}_j^v = -\mathrm{Im}(\boldsymbol{d}_j).$$

Using the real-valued notation, the objective function (10) can be written

$$\begin{aligned}\mathcal{G}(\boldsymbol{\alpha}) = &\left(1 - \frac{1}{E^2}\left|S_V(U_T(\boldsymbol{\alpha}))\right|^2\right) \\ &+ \frac{1}{T}\sum_{j=0}^{E-1}\int_0^T \Big(\langle \boldsymbol{u}_j(t,\boldsymbol{\alpha}), W\boldsymbol{u}_j(t,\boldsymbol{\alpha})\rangle_2 + \langle \boldsymbol{v}_j(t,\boldsymbol{\alpha}), W\boldsymbol{v}_j(t,\boldsymbol{\alpha})\rangle_2\Big)\, dt,\end{aligned} \tag{15}$$

where,

$$\begin{aligned}S_V(U_T) = &\sum_{j=0}^{E-1}\Big(\big\langle \boldsymbol{u}_j(T,\boldsymbol{\alpha}), \boldsymbol{d}_j^u\big\rangle_2 + \big\langle \boldsymbol{v}_j(T,\boldsymbol{\alpha}), \boldsymbol{d}_j^v\big\rangle_2\Big) \\ &+ i\sum_{j=0}^{E-1}\Big(\big\langle \boldsymbol{v}_j(T,\boldsymbol{\alpha}), \boldsymbol{d}_j^u\big\rangle_2 - \big\langle \boldsymbol{u}_j(T,\boldsymbol{\alpha}), \boldsymbol{d}_j^v\big\rangle_2\Big).\end{aligned} \tag{16}$$

3.1. Time integration

Let $t_n = nh$, for $n = 0, 1, \ldots, M$, be a uniform grid in time where $h = T/M$ is the time step. Also let $\boldsymbol{u}^n \approx \boldsymbol{u}(t_n)$ and $\boldsymbol{v}^n \approx \boldsymbol{v}(t_n)$ denote the numerical solution on the grid. We use a partitioned Runge-Kutta (PRK)

scheme [8] to discretize the real-valued formulation of Schrödinger's equation,

$$\boldsymbol{u}^0 = \boldsymbol{g}^u, \qquad \boldsymbol{v}^0 = \boldsymbol{g}^v, \tag{17}$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + h\sum_{i=1}^{s} b_i \boldsymbol{\kappa}^{n,i}, \qquad \boldsymbol{v}^{n+1} = \boldsymbol{v}^n + h\sum_{i=1}^{s} \hat{b}_i \boldsymbol{\ell}^{n,i}, \tag{18}$$

$$\boldsymbol{\kappa}^{n,i} = f^u(\boldsymbol{U}^{n,i}, \boldsymbol{V}^{n,i}, t_n + c_i h), \qquad \boldsymbol{\ell}^{n,i} = f^v(\boldsymbol{U}^{n,i}, \boldsymbol{V}^{n,i}, t_n + \hat{c}_i h), \tag{19}$$

$$\boldsymbol{U}^{n,i} = \boldsymbol{u}^n + h\sum_{j=1}^{s} a_{ij} \boldsymbol{\kappa}^{n,j}, \qquad \boldsymbol{V}^{n,i} = \boldsymbol{v}^n + h\sum_{j=1}^{s} \hat{a}_{ij} \boldsymbol{\ell}^{n,j}. \tag{20}$$

Here, $s \geq 1$ is the number of stages. The stage variables $\boldsymbol{U}^{n,i}$ and $\boldsymbol{V}^{n,i}$ are set in a bold font to indicate that they are unrelated to the solution operator matrix $U(t, \boldsymbol{\alpha})$ and the target gate matrix V.

The Strömer-Verlet scheme is a two-stage PRK method ($s = 2$) that is symplectic, time-reversible and second order accurate [8]. It combines the trapezoidal and the implicit midpoint rules, with Butcher coefficients:

$$a_{11} = a_{12} = 0, \quad a_{21} = a_{22} = \frac{1}{2}, \qquad \hat{a}_{11} = \hat{a}_{21} = \frac{1}{2}, \quad \hat{a}_{12} = \hat{a}_{22} = 0, \tag{21}$$

$$b_1 = b_2 = \frac{1}{2}, \quad c_1 = 0, \quad c_2 = 1, \qquad \hat{b}_1 = \hat{b}_2 = \frac{1}{2}, \quad \hat{c}_1 = \hat{c}_2 = \frac{1}{2}. \tag{22}$$

3.2. Time step restrictions for accuracy and stability

The accuracy in the numerical solution is essentially determined by how well the fastest time scale in the state vector is resolved by the grid in time. The analysis of the time scales in the solution of Schrödinger's equation is most straightforward to perform in the complex-valued formulation (6). The shortest time scale can be estimated by freezing the time-dependent coefficients in the Hamiltonian matrix at some time $t = t_*$ and considering Schrödinger's equation with the time-independent Hamiltonian matrix $H_* = H(t_*)$. The $N \times N$ matrix H_* is Hermitian and can be diagonalized by a unitary transformation,

$$H_* X = X\Gamma, \quad X^\dagger X = I_N, \quad \Gamma = \mathrm{diag}(\gamma_1, \gamma_2, \ldots, \gamma_N),$$

where the eigenvalues γ_k are real. By the change of variables $\widetilde{\boldsymbol{\psi}} = X^\dagger \boldsymbol{\psi}$, the solution of the diagonalized system follows as

$$\widetilde{\psi}_k(t) = e^{-i\gamma_k t}\widetilde{\psi}_k(0),$$

corresponding to the period $\tau_k = 2\pi/|\gamma_k|$. To obtain an accurate numerical solution, the shortest period in the solution must be resolved by at least C_P time steps. This leads to the time step restriction

$$h \leq \frac{\min_k \tau_k}{C_P} = \frac{2\pi}{C_P \max_k |\gamma_k|}.$$

The value of C_P that is needed to obtain a given accuracy in the numerical solution depends on the order of accuracy, the duration of the time integration, as well as the details of the time-stepping scheme. For second order accurate methods such as the Strömer-Verlet and the lowest order Magnus method, acceptable accuracy for engineering applications can often achieved with $C_P \approx 20$. With the Strömer-Verlet method, we note that the time-stepping can become unstable if $C_P \leq 2$, corresponding to a sampling rate below the Nyquist frequency.

Given the parameter vector $\boldsymbol{\alpha}$, we can calculate the largest values of the control functions $p(t)$ and $q(t)$, which then can be used to calculate

$$H_* = H_d + p_\infty(a+a^\dagger) + iq_\infty(a-a^\dagger), \quad p_\infty = \max_t |p(t,\boldsymbol{\alpha})|, \quad q_\infty = \max_t |q(t,\boldsymbol{\alpha})|,$$

where the maximum is evaluated for times $0 \leq t \leq T$. If the optimization imposes amplitude constraints on the parameter vector, $|\boldsymbol{\alpha}|_\infty \leq \alpha_{max}$, those constraints can be used to estimate the time step before the optimization starts. This allows the same time step to be used throughout the iteration and eliminates the need to recalculate the spectrum of H_* when $\boldsymbol{\alpha}$ changes.

3.3. Verifying second order of accuracy

To check the order of convergence in the Strömer-Verlet scheme, we consider two small test problems with analytical solutions. Both problems have a state vector with $N = 2$ components. For comparison, we also evaluate the performance of the lowest order Magnus integrator [8], which often is used for solving Schrödinger's equation. We solve (13) in the time interval $0 \leq t \leq T = 5\pi$, subject to the initial data $\boldsymbol{u}(0) = [1, 0]^T$ and $\boldsymbol{v}(0) = [0, 0]^T$. Also, we set $\omega = 2\pi$. For the first test case, let

$$K_1(t) = \frac{1}{4}(1 - \cos(\omega t)) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad S_1(t) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \tag{23}$$

In this case the analytical solution of (13) satisfies

$$\boldsymbol{u}_1(t) = \begin{pmatrix} \cos(\phi(t)) \\ 0 \end{pmatrix}, \quad \boldsymbol{v}_1(t) = \begin{pmatrix} 0 \\ \sin(\phi(t)) \end{pmatrix}, \quad \phi(t) = \frac{1}{4}\left(t - \frac{1}{\omega}\sin(\omega t)\right).$$

As a second test case, we take

$$K_2(t) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad S_2(t) = \frac{1}{4}(1 - \sin(\omega t)) \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \tag{24}$$

In this case the analytical solution is

$$\boldsymbol{u}_2(t) = \begin{pmatrix} \cos(\theta(t)) \\ -\sin(\theta(t)) \end{pmatrix}, \quad \boldsymbol{v}_2(t) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \theta(t) = \frac{1}{4}\left(t + \frac{1}{\omega}(\cos(\omega t) - 1)\right).$$

For both test cases, we evaluate the L_2-errors as function of the time step at the final time, $T = 5\pi$. Starting with a time step of $h_0 = 0.1$, the time step is successively refined by a factor of $\sqrt{10}$ in each numerical solution, i.e., $h_j = 10^{-j/2}h_0$. The results in Figure 2 indicate that the L_2 norm of the error is reduced by a factor of 10 in each refinement, verifying that both the Strömer-Verlet and Magnus integration scheme are second order accurate in h, as expected. Furthermore, we conclude that the accuracy of the Strömer-Verlet scheme is comparable to the Magnus scheme for both test problems.

4. Discretizing the objective function and its gradient

In this section, we develop a "discretize before optimize" approach in which we first discretize the objective function and then derive a compatible scheme for discretizing the adjoint differential equation, which is used for computing the gradient of the objective function. Because the Strömer-Verlet scheme is a partitioned Runge-Kutta scheme, previous results for autonomous systems show that the compatible discretization of the adjoint equation is another partitioned Runge-Kutta scheme [17]. Here, we generalize the result to the case of time-dependent matrices. We also derive the formula for combining the solutions of the Strömer-Verlet and the adjoint schemes to calculate the gradient of the discretized objective function.

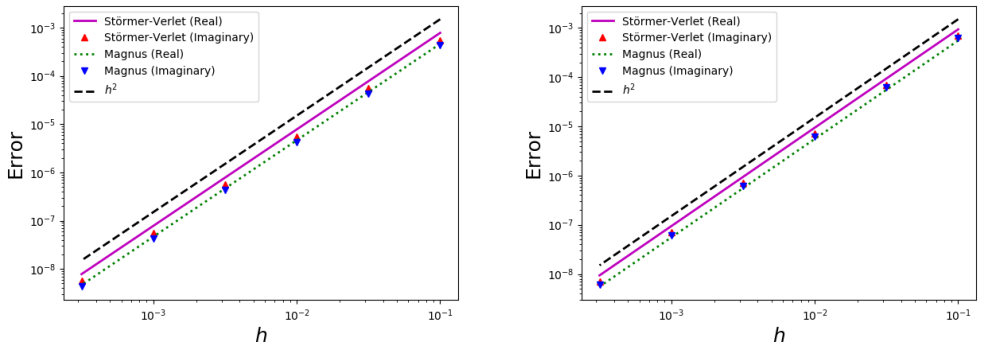

Figure 2: The L_2 errors in the numerical solutions at $t = 5\pi$ with the Strömer-Verlet and the lowest order Magnus time integrators. Second order convergence rate is shown by the dashed black line, indicating that both methods are second order accurate. The first test case (23) is show on the left, with the second test case (24) on the right.

The Strömer-Verlet scheme can be written in terms of the stage variables $(\boldsymbol{U}^{n,i}, \boldsymbol{V}^{n,i})$ by substituting $(\boldsymbol{\kappa}^{n,i}, \boldsymbol{\ell}^{n,i})$ from (19) into (18),

$$\boldsymbol{u}^0 = \boldsymbol{g}^u, \quad \boldsymbol{v}^0 = \boldsymbol{g}^v, \tag{25}$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \frac{h}{2}\left(S_n\boldsymbol{U}^{n,1} + S_{n+1}\boldsymbol{U}^{n,2} - K_n\boldsymbol{V}^{n,1} - K_{n+1}\boldsymbol{V}^{n,2}\right), \tag{26}$$

$$\boldsymbol{v}^{n+1} = \boldsymbol{v}^n + \frac{h}{2}\left(K_{n+1/2}\left(\boldsymbol{U}^{n,1} + \boldsymbol{U}^{n,2}\right) + S_{n+1/2}(\boldsymbol{V}^{n,1} + \boldsymbol{V}^{n,2})\right), \tag{27}$$

and into (20),

$$\boldsymbol{U}^{n,1} = \boldsymbol{u}^n, \tag{28}$$

$$\boldsymbol{U}^{n,2} = \boldsymbol{u}^n + \frac{h}{2}\left(S_n\boldsymbol{U}^{n,1} + S_{n+1}\boldsymbol{U}^{n,2} - K_n\boldsymbol{V}^{n,1} - K_{n+1}\boldsymbol{V}^{n,2}\right), \tag{29}$$

$$\boldsymbol{V}^{n,1} = \boldsymbol{v}^n + \frac{h}{2}\left(K_{n+1/2}\boldsymbol{U}^{n,1} + S_{n+1/2}\boldsymbol{V}^{n,1}\right), \tag{30}$$

$$\boldsymbol{V}^{n,2} = \boldsymbol{v}^n + \frac{h}{2}\left(K_{n+1/2}\boldsymbol{U}^{n,1} + S_{n+1/2}\boldsymbol{V}^{n,1}\right). \tag{31}$$

Here, $S_n = S(t_n)$, $S_{n+1/2} = S(t_n + 0.5h)$, etc. Because $S(t) \neq 0$, the scheme is block implicit. Note that $\boldsymbol{u}^{n+1} = \boldsymbol{U}^{n,2}$ and $\boldsymbol{V}^{n,1} = \boldsymbol{V}^{n,2} = \boldsymbol{v}(t_{n+1/2}) + \mathcal{O}(h^2)$.

The numerical solution at the final time step provides a second order accurate approximation of the continuous solution operator matrix U_T, which

we denote U_{Th}. It is used to approximate the matrix overlap function $S_V(U_T)$ in (16),

$$S_{Vh}(U_{Th}) = \sum_{j=0}^{E-1} \left(\left\langle \boldsymbol{u}_j^M, \boldsymbol{d}_j^u \right\rangle_2 + \left\langle \boldsymbol{v}_j^M, \boldsymbol{d}_j^v \right\rangle_2 \right) + i \sum_{j=0}^{E-1} \left(\left\langle \boldsymbol{v}_j^M, \boldsymbol{d}_j^u \right\rangle_2 - \left\langle \boldsymbol{u}_j^M, \boldsymbol{d}_j^v \right\rangle_2 \right), \tag{32}$$

which is then used as the first part of the discrete objective function,

$$\mathcal{J}_{1h}(U_{Th}) = \left(1 - \frac{1}{E^2} |S_{Vh}(U_{Th})|^2 \right). \tag{33}$$

The integral in the objective function (15) can be discretized to second order accuracy by using the Runge-Kutta stage variables,

$$\mathcal{J}_{2h}(\boldsymbol{U}, \boldsymbol{V}) = \frac{h}{T} \sum_{j=0}^{E-1} \sum_{n=0}^{M-1} \left(\frac{1}{2} \left\langle \boldsymbol{U}_j^{n,1}, W \boldsymbol{U}_j^{n,1} \right\rangle_2 + \frac{1}{2} \left\langle \boldsymbol{U}_j^{n,2}, W \boldsymbol{U}_j^{n,2} \right\rangle_2 + \left\langle \boldsymbol{V}_j^{n,1}, W \boldsymbol{V}_j^{n,1} \right\rangle_2 \right). \tag{34}$$

Based on the above formulas we discretize the objective function (15) according to

$$\mathcal{G}_h(\boldsymbol{\alpha}) = \mathcal{J}_h(U_{Th}^{\alpha}, \boldsymbol{U}^{\alpha}, \boldsymbol{V}^{\alpha}), \quad \mathcal{J}_h(U_{Th}, \boldsymbol{U}, \boldsymbol{V}) := \mathcal{J}_{1h}(U_{Th}) + \mathcal{J}_{2h}(\boldsymbol{U}, \boldsymbol{V}). \tag{35}$$

Here, U_{Th}^{α}, $\boldsymbol{U}^{\alpha}$ and $\boldsymbol{V}^{\alpha}$ represent the time-discrete solution of the Strömer-Verlet scheme for a given parameter vector $\boldsymbol{\alpha}$. We note that $\mathcal{G}_h(\boldsymbol{\alpha})$ can be evaluated by accumulation during the time-stepping of the Strömer-Verlet scheme.

4.1. The discrete adjoint approach

The gradient of the discretized objective function can be derived from first order optimality conditions of the corresponding discrete Lagrangian. This process was first outlined for Runge-Kutta schemes by Hager [7] and later extended to partitioned Runge-Kutta schemes by Ober-Blöbaum [15]. Also see the review paper by Sanz-Serna [17].

In this approach, let $(\boldsymbol{\mu}_j^n, \boldsymbol{\nu}_j^n)$ be the adjoint variables and let $(\boldsymbol{M}_j^{n,i}, \boldsymbol{N}_j^{n,i})$ be Lagrange multipliers. We define the discrete Lagrangian by

$$\begin{aligned}&\mathcal{L}_h(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{M}, \boldsymbol{N}, \boldsymbol{\alpha}) = \\ &\quad \mathcal{J}_h(U_{Th}, \boldsymbol{U}, \boldsymbol{V}) - \sum_{j=0}^{E-1} \left(\left\langle \boldsymbol{u}_j^0 - \boldsymbol{g}_j^u, \boldsymbol{\mu}_j^0 \right\rangle_2 + \left\langle \boldsymbol{v}_j^0 - \boldsymbol{g}_j^v, \boldsymbol{\nu}_j^0 \right\rangle_2 + \sum_{k=1}^{6} T_j^k \right). \quad (36)\end{aligned}$$

The first two terms in the sum enforce the initial conditions (25). The terms T_j^1 and T_j^2 enforce the time-stepping update formulas (26)-(27) in the Strömer-Verlet scheme,

$$T_j^1 = \sum_{n=0}^{M-1} \left\langle \boldsymbol{u}_j^{n+1} - \boldsymbol{u}_j^n - \frac{h}{2} \left(S_n \boldsymbol{U}_j^{n,1} + S_{n+1} \boldsymbol{U}_j^{n,2} - K_n \boldsymbol{V}_j^{n,1} - K_{n+1} \boldsymbol{V}_j^{n,2} \right), \boldsymbol{\mu}_j^{n+1} \right\rangle_2, \quad (37)$$

$$T_j^2 = \sum_{n=0}^{M-1} \left\langle \boldsymbol{v}_j^{n+1} - \boldsymbol{v}_j^n - \frac{h}{2} \left(K_{n+1/2} \left(\boldsymbol{U}_j^{n,1} + \boldsymbol{U}_j^{n,2} \right) + S_{n+1/2} (\boldsymbol{V}_j^{n,1} + \boldsymbol{V}_j^{n,2}) \right), \boldsymbol{\nu}_j^{n+1} \right\rangle_2. \quad (38)$$

Finally, the terms T_j^3 to T_j^6 enforce the relations between the stage variables (28)-(31),

$$T_j^3 = \sum_{n=0}^{M-1} \left\langle \boldsymbol{U}_j^{n,1} - \boldsymbol{u}_j^n, \boldsymbol{M}_j^{n,1} \right\rangle_2, \quad (39)$$

$$T_j^4 = \sum_{n=0}^{M-1} \left\langle \boldsymbol{U}_j^{n,2} - \boldsymbol{u}_j^n - \frac{h}{2} \left(S_n \boldsymbol{U}_j^{n,1} + S_{n+1} \boldsymbol{U}_j^{n,2} - K_n \boldsymbol{V}_j^{n,1} - K_{n+1} \boldsymbol{V}_j^{n,2} \right), \boldsymbol{M}_j^{n,2} \right\rangle_2, \quad (40)$$

$$T_j^5 = \sum_{n=0}^{M-1} \left\langle \boldsymbol{V}_j^{n,1} - \boldsymbol{v}_j^n - \frac{h}{2} \left(K_{n+1/2} \boldsymbol{U}_j^{n,1} + S_{n+1/2} \boldsymbol{V}_j^{n,1} \right), \boldsymbol{N}_j^{n,1} \right\rangle_2, \quad (41)$$

$$T_j^6 = \sum_{n=0}^{M-1} \left\langle \boldsymbol{V}_j^{n,2} - \boldsymbol{v}_j^n - \frac{h}{2} \left(K_{n+1/2} \boldsymbol{U}_j^{n,1} + S_{n+1/2} \boldsymbol{V}_j^{n,1} \right), \boldsymbol{N}_j^{n,2} \right\rangle_2. \quad (42)$$

To derive the discrete adjoint scheme, we note that the discrete La-

grangian (36) has a saddle point if

$$\frac{\partial \mathcal{L}_h}{\partial \boldsymbol{\mu}_j^n} = \frac{\partial \mathcal{L}_h}{\partial \boldsymbol{\nu}_j^n} = \frac{\partial \mathcal{L}_h}{\partial \boldsymbol{N}_j^{n,i}} = \frac{\partial \mathcal{L}_h}{\partial \boldsymbol{M}_j^{n,i}} = 0, \tag{43}$$

$$\frac{\partial \mathcal{L}_h}{\partial \boldsymbol{u}_j^n} = \frac{\partial \mathcal{L}_h}{\partial \boldsymbol{v}_j^n} = \frac{\partial \mathcal{L}_h}{\partial \boldsymbol{U}_j^{n,i}} = \frac{\partial \mathcal{L}_h}{\partial \boldsymbol{V}_j^{n,i}} = 0, \tag{44}$$

for $n = 0, 1, \ldots, M$, $i = 1, 2$ and $j = 0, 1, \ldots, E-1$. Here, the set of conditions in (43) result in the Strömer-Verlet scheme (25)-(31) for evolving $(\boldsymbol{u}_j^n, \boldsymbol{v}_j^n, \boldsymbol{U}_j^{n,i}, \boldsymbol{V}_j^{n,i})$ forwards in time. The set of conditions in (44) result in a time-stepping scheme for evolving the adjoint variables $(\boldsymbol{\mu}_j^n, \boldsymbol{\nu}_j^n)$ backwards in time, as is made precise in the following lemma.

Lemma 1. *Let $\mathcal{L}_h$ be the discrete Lagrangian defined by* (36)*. Furthermore, let $(\boldsymbol{u}_j^n, \boldsymbol{v}_j^n, \boldsymbol{U}_j^{n,i}, \boldsymbol{V}_j^{n,i})$ satisfy the Strömer-Verlet scheme* (25)-(31) *for a given parameter vector $\boldsymbol{\alpha}$. Then, the set of saddle-point conditions* (44) *are satisfied if the Lagrange multipliers $(\boldsymbol{\mu}_j^n, \boldsymbol{\nu}_j^n)$ are calculated according to the reversed time stepping scheme,*

$$\boldsymbol{\mu}_j^M = \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{u}_j^M}, \quad \boldsymbol{\nu}_j^M = \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{v}_j^M}, \tag{45}$$

$$\boldsymbol{\mu}_j^n = \boldsymbol{\mu}_j^{n+1} - \frac{h}{2}\left(\boldsymbol{\kappa}_j^{n,1} + \boldsymbol{\kappa}_j^{n,2}\right), \tag{46}$$

$$\boldsymbol{\nu}_j^n = \boldsymbol{\nu}_j^{n+1} - \frac{h}{2}\left(\boldsymbol{\ell}_j^{n,1} + \boldsymbol{\ell}_j^{n,2}\right), \tag{47}$$

for $n = M-1, M-2, \ldots 0$. Because $S^T = -S$ and $K^T = K$, the slopes satisfy

$$\boldsymbol{\kappa}_j^{n,1} = S_n \boldsymbol{X}_j^n - K_{n+1/2} \boldsymbol{Y}_j^{n,1} - \frac{2}{h} \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{U}_j^{n,1}}, \tag{48}$$

$$\boldsymbol{\kappa}_j^{n,2} = S_{n+1} \boldsymbol{X}_j^n - K_{n+1/2} \boldsymbol{Y}_j^{n,2} - \frac{2}{h} \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{U}_j^{n,2}}, \tag{49}$$

$$\boldsymbol{\ell}_j^{n,1} = K_n \boldsymbol{X}_j^n + S_{n+1/2} \boldsymbol{Y}_j^{n,1} - \frac{2}{h} \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{V}_j^{n,1}}, \tag{50}$$

$$\boldsymbol{\ell}_j^{n,2} = K_{n+1} \boldsymbol{X}_j^n + S_{n+1/2} \boldsymbol{Y}_j^{n,2} - \frac{2}{h} \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{V}_j^{n,2}}, \tag{51}$$

where the stage variables are given by

$$\boldsymbol{X}_j^n = \boldsymbol{\mu}_j^{n+1} - \frac{h}{2}\boldsymbol{\kappa}_j^{n,2}, \tag{52}$$

$$\boldsymbol{Y}_j^{n,2} = \boldsymbol{\nu}_j^{n+1}, \tag{53}$$

$$\boldsymbol{Y}_j^{n,1} = \boldsymbol{\nu}_j^{n+1} - \frac{h}{2}\left(\boldsymbol{\ell}_j^{n,1} + \boldsymbol{\ell}_j^{n,2}\right). \tag{54}$$

The time-stepping scheme (46)-(54) *has the form of a partitioned Runge-Kutta method for solving the adjoint of Schrödinger's equation* (13)*. Note that the adjoint scheme reverses the decomposition in the Strömer-Verlet scheme by using the implicit midpoint rule for* $\boldsymbol{\mu}$ *and the trapezoidal rule for* $\boldsymbol{\nu}$*, with the important exception that the time-dependent matrices* $K(t)$ *and* $S(t)$ *are evaluated at time levels that do* <u>not</u> *follow those rules.*

Proof. The lemma follows after a somewhat tedious but straightforward calculation shown in detail in Appendix C. □

Only the matrices K and S depend explicitly on $\boldsymbol{\alpha}$ in the discrete Lagrangian. When the saddle point conditions (43) and (44) are satisfied, we can therefore calculate the gradient of $\mathcal{G}_h$ by differentiating (36),

$$\frac{\partial \mathcal{G}_h}{\partial \alpha_r} = \frac{\partial \mathcal{L}_h}{\partial \alpha_r}, \quad r = 0, 1, \ldots, E-1.$$

This relation leads to the following Lemma.

Lemma 2. *Let* $\mathcal{L}_h$ *be the discrete Lagrangian defined by* (36)*. Assume that* $(\boldsymbol{u}_j^n, \boldsymbol{v}_j^n, \boldsymbol{U}_j^{n,i}, \boldsymbol{V}_j^{n,i})$ *are calculated according to the Strömer-Verlet scheme for a given parameter vector* $\boldsymbol{\alpha}$*. Furthermore, assume that* $(\boldsymbol{\mu}_j^n, \boldsymbol{\nu}_j^n, \boldsymbol{X}_j^n, \boldsymbol{Y}_j^{n,i})$ *satisfy the adjoint time-stepping scheme in Lemma 1, subject to the terminal conditions*

$$\boldsymbol{\mu}_j^M = -\frac{2}{E^2}\left(Re(S_{Vh})\boldsymbol{d}_j^u - Im(S_{Vh})\boldsymbol{d}_j^v\right), \quad \boldsymbol{\nu}_j^M = -\frac{2}{E^2}\left(Re(S_{Vh})\boldsymbol{d}_j^v + Im(S_{Vh})\boldsymbol{d}_j^u\right),$$

and the forcing functions

$$\frac{\partial \mathcal{J}_h}{\partial \boldsymbol{U}_j^{n,1}} = \frac{h}{T}W\boldsymbol{U}_j^{n,1}, \quad \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{U}_j^{n,2}} = \frac{h}{T}W\boldsymbol{U}_j^{n,2}$$

$$\frac{\partial \mathcal{J}_h}{\partial \boldsymbol{V}_j^{n,1}} = \frac{h}{T}W\boldsymbol{V}_j^{n,1}, \quad \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{V}_j^{n,2}} = 0.$$

Then, the saddle-point conditions (43) *and* (44) *are satisfied and the gradient of the objective function* (35) *is given by*

$$\frac{\partial \mathcal{G}_h}{\partial \alpha_r} = \frac{h}{2} \sum_{j=0}^{E-1} \sum_{n=0}^{M-1} \Big\{ \big\langle S'_n \boldsymbol{U}_j^{n,1} + S'_{n+1} \boldsymbol{U}_j^{n,2} - (K'_n + K'_{n+1}) \boldsymbol{V}_j^{n,1}, \boldsymbol{X}_j^n \big\rangle_2 + \big\langle K'_{n+1/2} \boldsymbol{U}_j^{n,1} + S'_{n+1/2} \boldsymbol{V}_j^{n,1}, \boldsymbol{Y}_j^{n,1} \big\rangle_2 + \big\langle K'_{n+1/2} \boldsymbol{U}_j^{n,2} + S'_{n+1/2} \boldsymbol{V}_j^{n,1}, \boldsymbol{Y}_j^{n,2} \big\rangle_2 \Big\}, \tag{55}$$

where $S'_n = \partial S / \partial \alpha_r(t_n)$, $K'_{n+1/2} = \partial K / \partial \alpha_r(t_{n+1/2})$, *etc.*

Proof. See Appendix D. □

As a result of Lemma 2, all components of the gradient can be calculated from $(\boldsymbol{u}_j^n, \boldsymbol{v}_j^n, \boldsymbol{U}_j^{n,i}, \boldsymbol{V}_j^{n,1})$ and the adjoint variables $(\boldsymbol{\mu}_j^n, \boldsymbol{\nu}_j^n, \boldsymbol{X}_j^n, \boldsymbol{Y}_j^{n,i})$. The first set of variables are obtained from time-stepping the Strömer-Verlet scheme forwards in time, while the second set of variables follow from time-stepping the adjoint scheme backwards in time.

We can avoid storing the time-history of $(\boldsymbol{u}_j^n, \boldsymbol{v}_j^n, \boldsymbol{U}_j^{n,i}, \boldsymbol{V}_j^{n,1})$ by using the time-reversibility of the Strömer-Verlet scheme. However, in order to do so, we must first calculate the terminal conditions $(\boldsymbol{u}_j^M, \boldsymbol{v}_j^M)$ by evolving (25)-(31) forwards in time. The time-stepping can then be reversed and the gradient of the objective function (55) can be accumulated by simultaneously time-stepping the adjoint system (46)-(54) backwards in time.

5. Quadratic B-splines with carrier waves

We expand the control function in terms of B-spline basis functions with carrier waves, see Figure 1, given by

$$p(t, \boldsymbol{\alpha}) = \sum_{\ell=1}^{N_f} \sum_{m=1}^{D_1} \alpha_{m,\ell}^{(1)} B_m(t) \cos(\Omega_\ell t), \quad q(t, \boldsymbol{\alpha}) = \sum_{\ell=1}^{N_f} \sum_{m=1}^{D_1} \alpha_{m,\ell}^{(2)} B_m(t) \cos(\Omega_\ell t). \tag{56}$$

Here, Ω_ℓ is the angular frequency of the ℓth carrier wave, N_f is the number of frequencies, $D_1 \geq 1$ is the number of B-spline basis functions per frequency and $\alpha_{m,\ell}^{(j)}$ are real coefficients. The total number of parameters in (56) is $D = 2N_f D_1$, which equals the size of the parameter vector $\boldsymbol{\alpha}$. The case $\Omega_\ell = 0$ corresponds to a regular B-spline function, see Figure 3. In this

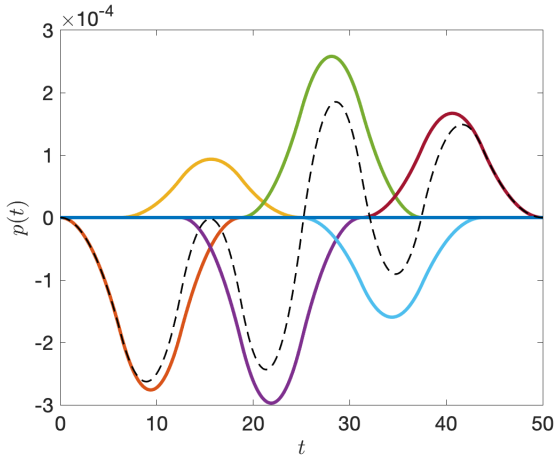

Figure 3: A B-spline control function $p(t, \boldsymbol{\alpha})$ without carrier wave ($\Omega_1 = 0$ and $N_f = 1$). Here, the black dashed line is the control function and the solid colored lines are the individual B-spline basis functions, scaled by $\alpha^1_{m,1}$. In this case, $D_1 = 6$.

paper we use the quadratic B-spline basis, where each basis function is a piecewise quadratic polynomial in time. It is the lowest order B-spline basis that has at least one continuous derivative. We define the basis functions on a uniform grid in time,

$$t_m = (m + 0.5)\delta, \quad m = 1, \ldots, D_1, \quad \delta = \frac{T}{D_1 + 2}. \tag{57}$$

Each basis function $B_m(t)$ is centered around $t = t_m$ and are easily expressed in terms of the scaled time parameter $\tau_m(t) = (t - t_m)/3\delta$,

$$B_m(t) = \widetilde{B}(\tau_m(t)), \quad \widetilde{B}(\tau) = \begin{cases} \frac{9}{8} + \frac{9}{2}\tau + \frac{9}{2}\tau^2, & -\frac{1}{2} \le \tau < -\frac{1}{6}, \\ \frac{3}{4} - 9\tau^2, & -\frac{1}{6} \le \tau < \frac{1}{6}, \\ \frac{9}{8} - \frac{9}{2}\tau + \frac{9}{2}\tau^2, & \frac{1}{6} \le \tau < \frac{1}{2}, \\ 0, & \text{otherwise.} \end{cases} \tag{58}$$

Note that $B_m(t)$ is only non-zero in the interval $t \in [t_m - 1.5\delta, t_m + 1.5\delta]$. Thus, at any fixed time, a control function will only get contributions from at most three B-spline basis functions. This property allows the control functions to be evaluated very efficiently.

The coefficients $\alpha^{(j)}_{m,\ell}$ enter linearly in the expression for the control functions. This property greatly simplifies the calculation of their derivatives. For

example, the derivative of $p(t, \boldsymbol{\alpha})$ with respect to $\alpha^{(1)}_{m,\ell}$ equals $B_m(t)\cos(\Omega_\ell t)$. This function is non-zero in the interval $t \in [t_m - 1.5\delta, t_m + 1.5\delta]$ and vanishes identically outside it. Furthermore, all derivatives of p with respect to $\alpha^2_{m,\ell}$ are identically zero. A similar argument applies to $q(t, \boldsymbol{\alpha})$.

6. Numerical optimization

Our numerical solution of the optimal control problem is based on the general purpose interior-point optimization package IPOPT [19]. This open-source library implements a primal-dual barrier approach for solving large-scale nonlinear programming problems, i.e., it minimizes an objective function subject to inequality (barrier) constraints on the parameter vector. Because the Hessian of the objective function is costly to calculate, we use the L-BFGS algorithm [14] in IPOPT, which only relies on the objective function and its gradient to be evaluated. Inequality constraints that limit the amplitude of the parameter vector $\boldsymbol{\alpha}$ are enforced internally by IPOPT.

The routines for evaluating the objective function and its gradient are implemented in the Julia programming language [1], which provides a convenient interface to IPOPT. Given a parameter vector $\boldsymbol{\alpha}$, the routine for evaluating the objective function solves the Schrödinger equation with the Strömer-Verlet scheme and evaluates $\mathcal{G}_h(\boldsymbol{\alpha})$ by accumulation. The routine for evaluating the gradient first applies the Strömer-Verlet scheme to calculate terminal conditions for the state variables. It then proceeds by accumulating the gradient $\nabla_\alpha \mathcal{G}_h$ by simultaneous reversed time-stepping of the discrete adjoint scheme and the Strömer-Verlet scheme. All problem specific information are encapsulated in these two routines.

6.1. Verifying the discrete adjoint scheme

The discrete objective function can be used to approximate each component of the discrete gradient by a centered finite difference formula,

$$\frac{\partial \mathcal{G}_h(\boldsymbol{\alpha})}{\partial \alpha_k} = \frac{\mathcal{G}_h(\boldsymbol{\alpha} + \varepsilon \boldsymbol{e}_k) - \mathcal{G}_h(\boldsymbol{\alpha} - \varepsilon \boldsymbol{e}_k)}{2\varepsilon} + \mathcal{O}(\varepsilon^2), \quad k = 1, 2, \ldots, D.$$

To test the correctness of the adjoint scheme, we consider the case of a qudit with $N = 6$ energy levels, $E = 4$ essential levels and $G = 2$ guard levels. To define the objective function, we define the target gate matrix V to be a CNOT gate for the essential states, set $T = 100$ ns, $\omega_a = 2\pi \cdot 4.10336$ Grad/s, $\xi_a = 2\pi \cdot 0.2198$ Grad/s and use the time step $h = 2.8833 \cdot 10^{-3}$

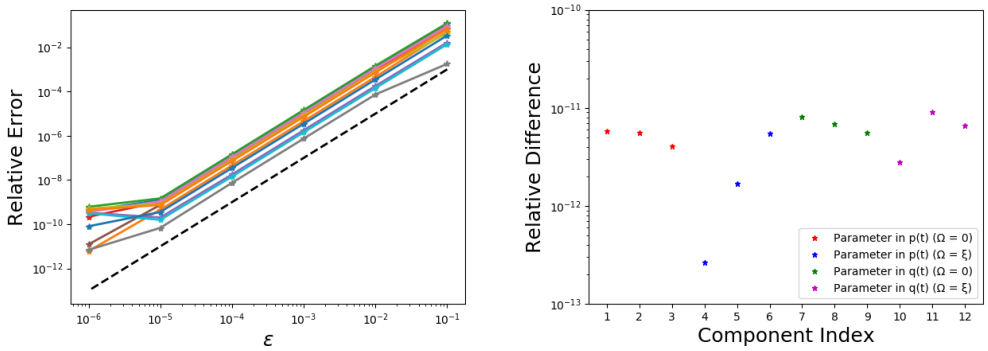

Figure 4: Accuracy of the discrete adjoint gradient. (Left): relative error in a centered finite difference approximation of the gradient compared to the adjoint gradient. The dashed line corresponds to second order accuracy. (Right): relative difference between the adjoint and direct methods for computing the discrete gradient, per component. Both methods agree to within 11-12 digits.

ns. We discourage population of the two guard levels in the state vector by setting $W = \text{diag}[0, 0, 0, 0, 0.2, 2.0]$ in the second part of the objective function, see (9). The control functions are parameterized using B-splines with carrier waves, with $N_f = 2$ carrier frequencies with $\Omega_1 = 0$, $\Omega_2 = \xi_a$ and $D_1 = 3$ basis functions per frequency and control function, for a total of $D = 12$ parameters. We test the gradient on the 12-dimensional parameter vector $\boldsymbol{\alpha} = [-0.05, -0.04, \ldots, 0.05, 0.06]^T$.

A comparison between all components of the adjoint gradient and the centered finite difference approximation is shown in Figure 4. From these results we see that the centered approximation converges towards the adjoint gradient as $\mathcal{O}(\varepsilon^2)$, at least for $\varepsilon \geq 10^{-5}$. The convergence flattens out for $\varepsilon = 10^{-6}$, probably due to round-off errors in the finite difference approximation. As shown in Appendix B, we may also compute the discrete gradient using a direct approach, based on calculating the derivative of the state vector with respect to each component of $\boldsymbol{\alpha}$. The relative difference between the adjoint and direct gradients is displayed in Figure 4, indicating that both approaches yield the same result to within round-off errors. From these results we infer that the discrete adjoint scheme is correctly implemented.

6.2. A CNOT gate on a single qudit with guard levels

To test our methods on a quantum optimal control problem, we consider realizing a CNOT gate on a single qudit with four essential energy levels and two guard levels. The setup is the same as in Section 6.1, unless otherwise noted. We parameterize the two control functions using B-splines with carrier waves and choose the frequencies to be $\Omega_1 = 0$, $\Omega_2 = \xi_a$ and $\Omega_3 = 2\xi_a$. In the rotating frame, these frequencies correspond to transitions between the ground state and the first exited state, the first and second excited states and the second and third excited states. We use two guard levels and penalize population of the fourth and fifth excited states using the weight matrix $W = \text{diag}[0, 0, 0, 0, 0.1, 1.0]$ in $\mathcal{J}_{2h}$, see (34). We use $D_1 = 10$ basis functions per frequency and control function, resulting in a total of $D = 60$ parameters. The amplitudes of the control functions are limited by the constraints

$$\|\boldsymbol{\alpha}\|_\infty := \max_{1\leq r\leq D} |\alpha_r| \leq \alpha_{max}. \tag{59}$$

We estimate the time step using the technique in Section 3.2. To guarantee at least $C_P = 40$ time steps per period with $\alpha_{max} = 0.1$ GHz, we use $M = 8,798$ time steps, corresponding to $h \approx 1.1366 \cdot 10^{-2}$ ns. The L-BFGS algorithm within IPOPT is restarted every 10 iterations.

As initial guess for the elements of the parameter vector, we use a random number generator with a uniform distribution in $[-0.01, 0.01]$. In Figure 5 we present the convergence of the objective function $\mathcal{G}$, decomposed into $\mathcal{J}_{1h}$ and $\mathcal{J}_{2h}$, together with the norm of the gradient $\|\nabla_\alpha \mathcal{G}\|_2$. The unconstrained case ($\alpha_{max} = \infty$) is shown on the left and a constrained case with $\alpha_{max} = 0.05$ is shown on the right. In the unconstrained case, the convergence is very irregular with several spikes in both the objective function and its gradient. The approximate optima has $\|\boldsymbol{\alpha}\|_\infty \approx 0.158$ and the two parts of the objective function are $\mathcal{J}_{1h} \approx 2.37 \cdot 10^{-4}$ and $\mathcal{J}_{2h} \approx 4.97 \cdot 10^{-3}$. The convergence and the robustness of the optimization are greatly improved when the constraint $\alpha_{max} = 0.05$ is imposed on the parameter vector. In this case the objective function remains at approximately the same level from iteration 40 and onwards, the approximate optima has $\|\boldsymbol{\alpha}\|_\infty \approx 0.046$ and the two parts of the objective function are $\mathcal{J}_{1h} \approx 8.89 \cdot 10^{-5}$ and $\mathcal{J}_{2h} \approx 2.26 \cdot 10^{-4}$, corresponding to a trace fidelity greater than 0.9999. Thus, by limiting the amplitude of the parameter vector, the optimizer is able to find a significantly better optima, with smaller amplitude, smaller gate infidelity and smaller population of the guard states. In particular, the "forbidden" state $|5\rangle$ has a population that

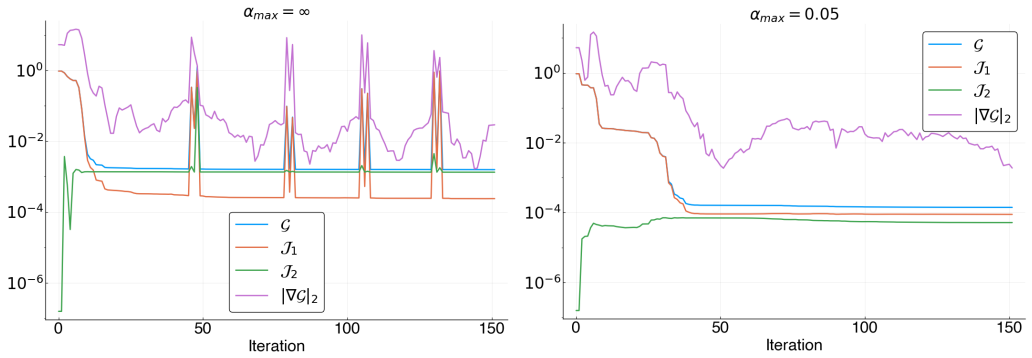

Figure 5: The inequality constraint on the parameter vector has a regularizing effect on the convergence of the iteration. Left: no constraint, $\alpha_{max} = \infty$. Right: enforcing the constraint $\alpha_{max} = 0.05$ (right).

remains below $1.25 \cdot 10^{-6}$ for all times and initial conditions, see Figure 6. The optimized control functions are shown in Figure 7 and the population of the essential states, corresponding to the four initial conditions of the CNOT gate, are presented in Figure 8.

The convergence plots in Figure 5 show that the gradient of the objective function does not tend to zero when the objective function converges. In the constrained case with $\alpha_{max} = 0.05$, the norm of the gradient stops decreasing after 50 iterations and starts oscillating around 10^{-2}. This indicates that the optima is not well defined. To get further clarity into the properties of the optima we study the Hessian of the objective function. Let the optima correspond to the parameter vector $\boldsymbol{\alpha}^*$. Based on the adjoint scheme for calculating the gradient, we can approximate the elements of the Hessian matrix using a centered finite difference approximation,

$$\frac{\partial^2 \mathcal{G}_h(\boldsymbol{\alpha}^*)}{\partial\alpha_j \partial\alpha_k} \approx \frac{1}{2\varepsilon}\left(\frac{\partial \mathcal{G}_h}{\partial\alpha_j}(\boldsymbol{\alpha}^* + \varepsilon \boldsymbol{e}_k) - \frac{\partial \mathcal{G}_h}{\partial\alpha_j}(\boldsymbol{\alpha}^* - \varepsilon \boldsymbol{e}_k)\right) := L_{j,k}, \tag{60}$$

for $j, k = 1, 2, \ldots, D$. To perform this calculation, the gradient must be evaluated for the $2D$ parameter vectors $(\boldsymbol{\alpha}^* \pm \varepsilon \boldsymbol{e}_k)$. Because the objective function and the parameter vector are real-valued, the gradient and the Hessian are also real-valued. Due to the finite difference approximation, the matrix L is only approximately equal to the Hessian. The accuracy in L can be estimated by studying the norm of its asymmetric part, which is zero for the Hessian, see Table 1. Based on this experiment we infer that $\varepsilon = 10^{-5}$ is appropriate to use for approximating the Hessian in (60). To eliminate

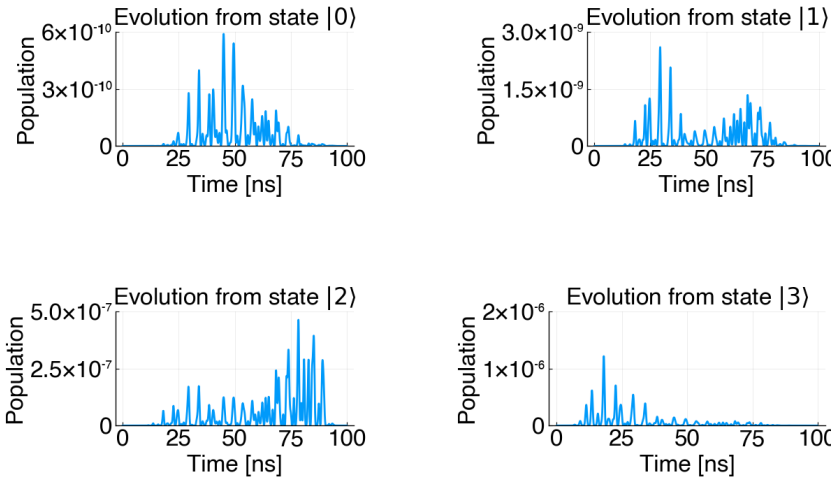

Figure 6: The population of the "forbidden" state $|5\rangle$ as function of time for the four initial conditions of the CNOT gate. Here, $\alpha_{max} = 0.05$.

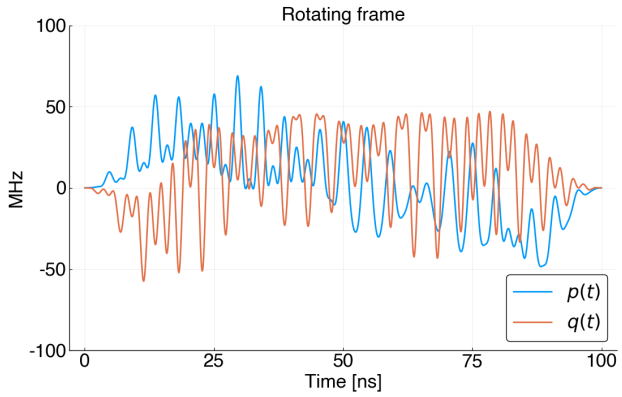

Figure 7: The control functions $p(t)$ (blue) and $q(t)$ (orange) for realizing a CNOT gate with $D_1 = 10$ basis function per carrier wave and three carrier wave frequencies. Here, $\alpha_{max} = 0.05$.

ε	$\|0.5(L + L^T)\|_F$	$\|0.5(L - L^T)\|_F$
10^{-4}	$1.066 \cdot 10^3$	$1.967 \cdot 10^{-6}$
10^{-5}	$1.066 \cdot 10^3$	$7.122 \cdot 10^{-8}$
10^{-6}	$1.066 \cdot 10^3$	$6.096 \cdot 10^{-7}$

Table 1: The Fröbenius norm of the symmetric and asymmetric parts of the approximate Hessian, L, for the case $\alpha_{max} = 0.05$.

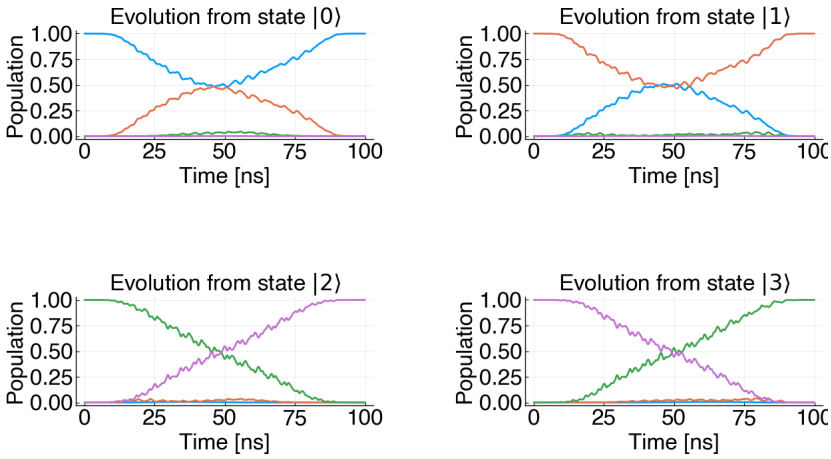

Figure 8: The population of the states $|0\rangle$ (blue), $|1\rangle$ (orange), $|2\rangle$ (green) and $|3\rangle$ (purple), as function of time, for the four initial conditions of the CNOT gate. Here, $\alpha_{max} = 0.05$.

spurious effects from the asymmetry in the L matrix, we study the spectrum of its symmetric part, $L_s = 0.5(L + L^T)$. Because it is real and symmetric, it has a complete set of eigenvectors and all eigenvalues are real. The eigenvalues for three different constraint levels (α_{max}) are shown in Figure 9. Two properties of the spectra are noteworthy. First, for all constraint levels the 15 largest eigenvalues are significantly larger than the rest. Secondly, a handful of eigenvalues are negative, indicating that the quantum optimal control problem is non-convex. The first property indicates that the control functions are essentially described by the 15 eigenvectors associated with the 15 largest eigenvalues. As a result, perturbations along the remaining 45 eigenvectors have a very small effect on the objective function. This hampers the convergence of the iteration, but may be rectified by scaling the parameters. However, the presence of negative eigenvalues means that the constraints are essential for making the optima well-defined. In future work, it would be interesting to explore if the convergence properties can be improved by using constraints based on non-linear functions of the parameter vector.

7. Conclusion

In this paper we have developed numerical methods for optimizing control functions for realizing logical gates in a closed quantum system. Our main contribution is a "discretize before optimize" approach in which we

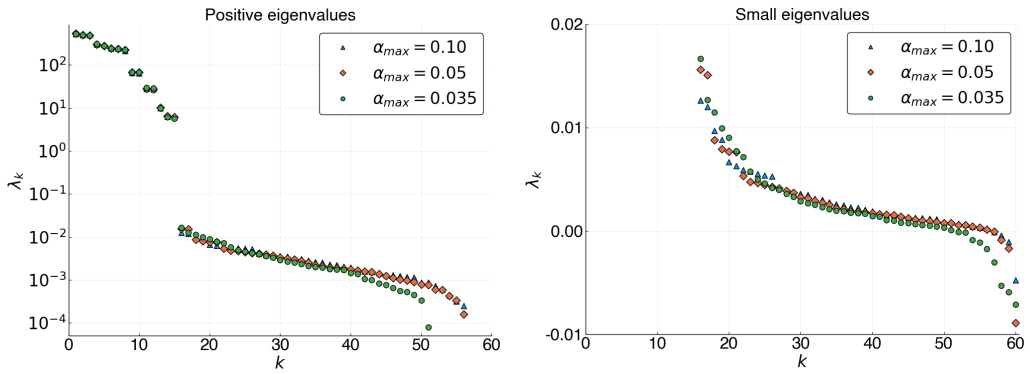

Figure 9: The eigenvalues of the symmetric part of the approximate Hessian, $0.5(L + L^T)$, evaluated at the optima for the constraints $\alpha_{max} = 0.10$ (blue triangles), $\alpha_{max} = 0.05$ (orange diamonds) and $\alpha_{max} = 0.035$ (green circles). The positive eigenvalues are shown on a log-scale on the left and the small eigenvalues are shown on a linear scale on the right.

first discretize the objective function and then derive a compatible scheme for discretizing the adjoint differential equation, which is used for computing the exact gradient of the discretized objective function. We have also introduced a parameterization of the control functions based on B-splines with built in carrier waves, where the smoothness follows from the basis functions. Our numerical solution of the optimal control problem is based on the general purpose interior-point optimization package IPOPT [19], which implements a primal-dual barrier approach for minimizing the objective function subject to amplitude constraints on the parameter vector. We have demonstrated that the amplitude constraints regularize and improve the convergence of the optimization problem. We optimized the control functions for a CNOT gate with two guard states, resulting in a gate trace fidelity greater than 0.9999. Having a moderate number of control parameters enabled us to study the spectrum of the Hessian of the objective function at an optima. A handful of the eigenvalues were found to be small and negative, indicating that the optimal control problem is non-convex. In future work, it would be interesting to study if the control problem can be made convex by modifying the objective function and if the convergence can be improved by imposing non-linear constrains on the parameter vector.

Acknowledgment

We would like to thank Prof. Daniel Appelö for bringing the Strömer-Verlet method to our attention.

This work was supported in part under LLNL laboratory directed research and development project 20-ERD-028 and in part by DOE office of advanced scientific computing research (OASCR) under the Advanced Research in Quantum Computing (ARQC) program, project TEAM.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. This is contribution LLNL-JRNL-800457.

Appendix A. The Hamiltonian in a rotating frame of reference

In the laboratory frame of reference, the Hamiltonian matrix for a single super-conducting qudit is assumed to be of the form

$$H(t) = \omega_a a^\dagger a - \frac{\xi_a}{2} a^\dagger a^\dagger a a + f(t)(a + a^\dagger). \tag{A.1}$$

Here, $\omega_a > 0$ and ξ_a are given real constants and $f(t, \boldsymbol{\alpha})$ is a real-valued function of time that depend on the parameter vector $\boldsymbol{\alpha}$. Furthermore, a is the annihilation matrix,

$$a = \begin{bmatrix} 0 & 1 & & & & \\ & 0 & \sqrt{2} & & & \\ & & \ddots & \ddots & & \\ & & & 0 & \sqrt{N-1} & \\ & & & & 0 & \end{bmatrix},$$

and the creation matrix $a^\dagger$ is its adjoint (conjugate transpose).

To derive the rotating frame transformation, we consider the time-dependent change of variable

$$\boldsymbol{\psi}(t) = R^\dagger(t)\widetilde{\boldsymbol{\psi}}(t), \quad R^\dagger R = I.$$

We have

$$\dot{\boldsymbol{\psi}} = \dot{R}^\dagger \widetilde{\boldsymbol{\psi}} + R^\dagger \dot{\widetilde{\boldsymbol{\psi}}}, \quad H\boldsymbol{\psi} = H R^\dagger \widetilde{\boldsymbol{\psi}}.$$

After some algebra, the Schrödinger equation (6) and the identity $R\dot{R}^\dagger = -\dot{R}R^\dagger$ gives:

$$\dot{\widetilde{\boldsymbol{\psi}}} = -i\tilde{H}(t)\widetilde{\boldsymbol{\psi}}, \quad \tilde{H}(t) = R(t)H(t)R(t)^\dagger + i\dot{R}(t)R(t)^\dagger. \tag{A.2}$$

The rotating frame of reference is introduced by taking the unitary transformation to be

$$R(t) = \exp(i\omega_a t a^\dagger a), \quad a^\dagger a = \begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & 2 & & \\ & & & \ddots & \\ & & & & N-1 \end{bmatrix}, \quad \dot{R}R^\dagger = i\omega_a a^\dagger a. \tag{A.3}$$

From (A.2) and (A.3), the first term in the Hamiltonian (A.1) is canceled by the term $i\dot{R}(t)R(t)^\dagger$. Furthermore, $a^\dagger a^\dagger aa = (a^\dagger a)^2 - a^\dagger a$ and both $a^\dagger a$ and $(a^\dagger a)^2$ commute with $R(t)$. After noting that $Ra^\dagger R^\dagger = e^{i\omega_a t}a^\dagger$, the transformed Hamiltonian can be written

$$\widetilde{H}(t) = -\frac{\xi_a}{2}\left((a^\dagger a)^2 - a^\dagger a\right) + f(t)\left(e^{-i\omega_a t}a + e^{i\omega_a t}a^\dagger\right). \tag{A.4}$$

To slow down the time scales in the control function, we want to absorb the highly oscillatory factors $\exp(\pm i\omega_a t)$ into $f(t)$. Because the control function $f(t)$ is real-valued, this can only be done in an approximate fashion. We make the ansatz,

$$f(t) = 2p_1(t)\cos(\omega_a t) - 2q_1(t)\sin(\omega_a t) = \\ (p_1 + iq_1)\exp(i\omega_a t) + (p_1 - iq_1)\exp(-i\omega_a t), \tag{A.5}$$

where $p_1(t)$ and $q_1(t)$ are real-valued functions. After some algebra, the transformed Hamiltonian (A.4) becomes

$$\widetilde{H}(t) = -\frac{\xi_a}{2}\left((a^\dagger a)^2 - a^\dagger a\right) + p_1\left(a + a^\dagger\right) + iq_1\left(a - a^\dagger\right) \\ + (p_1 - iq_1)\exp(-2i\omega_a t)a + (p_1 + iq_1)\exp(2i\omega_a t)a^\dagger.$$

The rotating frame approximation follows by ignoring the terms that oscillate with twice the frequency, $\exp(\pm 2i\omega_a t)$, resulting in the transformed

Schrödinger system,

$$\dot{\widetilde{\boldsymbol{\psi}}}_j = -i\left(H_d + \widetilde{H}_c(t)\right)\widetilde{\boldsymbol{\psi}}_j, \quad \widetilde{\boldsymbol{\psi}}_j(0) = \boldsymbol{e}_j, \tag{A.6}$$

$$H_d = -\frac{\xi_a}{2}\left(a^\dagger a^\dagger a a\right), \quad \widetilde{H}_c(t) = p_1(t)\left(a + a^\dagger\right) + iq_1(t)\left(a - a^\dagger\right). \tag{A.7}$$

Here, H_d is called the drift Hamiltonian. When $\xi_a \ll \omega_a$, the state vector varies on a significantly slower time scale in the rotating frame than in the laboratory frame.

In the remainder of the paper, the Schrödinger equation is always solved under the rotating frame approximation and we drop the tildes on the state vector and the Hamiltonian matrices.

Appendix B. The direct approach for the discrete gradient

To simplify the notation in this section, we set $\boldsymbol{V}_j^n := \boldsymbol{V}_j^{n,1}$. Denote,

$$\boldsymbol{p}_j^n := \frac{\partial \boldsymbol{u}_j^n}{\partial \alpha_r}, \quad \boldsymbol{q}_j^n := \frac{\partial \boldsymbol{v}_j^n}{\partial \alpha_r}, \quad \boldsymbol{P}_j^{n,i} := \frac{\partial \boldsymbol{U}_j^{n,i}}{\partial \alpha_r}, \quad \boldsymbol{Q}_j^n := \frac{\partial \boldsymbol{V}_j^n}{\partial \alpha_r}.$$

We may calculate a single component of the gradient by first differentiating (35) with respect to α_r,

$$\begin{aligned}\frac{\partial \mathcal{J}_h}{\partial \alpha_r} = &-\frac{2\ \mathrm{Re}(S_{Vh})}{E^2}\sum_{j=0}^{E-1}\left(\left\langle \boldsymbol{p}_j^M, \boldsymbol{d}_j^u\right\rangle_2 + \left\langle \boldsymbol{q}_j^M, \boldsymbol{d}_j^v\right\rangle_2\right) \\ &- \frac{2\ \mathrm{Im}(S_{Vh})}{E^2}\sum_{j=0}^{E-1}\left(\left\langle \boldsymbol{q}_j^M, \boldsymbol{d}_j^u\right\rangle_2 - \left\langle \boldsymbol{p}_j^M, \boldsymbol{d}_j^v\right\rangle_2\right) \\ &+ \frac{2h}{T}\sum_{j=0}^{E-1}\sum_{n=0}^{M-1}\left(\frac{1}{2}\left(\left\langle \boldsymbol{U}_j^{n,1}, W\boldsymbol{P}_j^{n,1}\right\rangle_2 + \left\langle \boldsymbol{U}_j^{n,2}, W\boldsymbol{P}_j^{n,2}\right\rangle_2\right) + \left\langle \boldsymbol{V}_j^n, W\boldsymbol{Q}_j^n\right\rangle_2\right).\end{aligned} \tag{B.1}$$

We can calculate $\boldsymbol{p}^n$, $\boldsymbol{q}^n$, $\boldsymbol{P}^{n,i}$ and $\boldsymbol{Q}^n$ by differentiating the Strömer-Verlet scheme (26)-(27) with respect to α_r,

$$\boldsymbol{p}_j^0 = 0, \quad \boldsymbol{q}_j^0 = 0, \tag{B.2}$$

$$\begin{aligned}\boldsymbol{p}_j^{n+1} = \boldsymbol{p}_j^n + \frac{h}{2}\big(&S_n\boldsymbol{P}_j^{n,1} + S_{n+1}\boldsymbol{P}_j^{n,2} - (K_n + K_{n+1})\boldsymbol{Q}_j^n \\ &+S_n'\boldsymbol{U}_j^{n,1} + S_{n+1}'\boldsymbol{U}_j^{n,2} - (K_n' + K_{n+1}')\boldsymbol{V}_j^n\big),\end{aligned} \tag{B.3}$$

$$\begin{aligned}\boldsymbol{q}_j^{n+1} = \boldsymbol{q}_j^n + \frac{h}{2}\big(&K_{n+1/2}(\boldsymbol{P}_j^{n,1} + \boldsymbol{P}_j^{n,2}) + 2S_{n+1/2}\boldsymbol{Q}_j^n \\ &+K_{n+1/2}'(\boldsymbol{U}_j^{n,1} + \boldsymbol{U}_j^{n,2}) + 2S_{n+1/2}'\boldsymbol{V}_j^n\big),\end{aligned} \tag{B.4}$$

where $S_n' = \partial S_n/\partial\alpha_r$, etc. The stage variables $\boldsymbol{P}^{n,i}$ and $\boldsymbol{Q}^n$ are calculated by differentiating (28)-(30),

$$\boldsymbol{P}_j^{n,1} = \boldsymbol{p}_j^n, \tag{B.5}$$

$$\begin{aligned}\boldsymbol{P}_j^{n,2} = \boldsymbol{p}_j^n + \frac{h}{2}\big(&S_n\boldsymbol{P}_j^{n,1} + S_{n+1}\boldsymbol{P}_j^{n,2} - (K_n + K_{n+1})\boldsymbol{Q}_j^n \\ &+S_n'\boldsymbol{U}_j^{n,1} + S_{n+1}'\boldsymbol{U}_j^{n,2} - (K_n' + K_{n+1}')\boldsymbol{V}_j^n\big),\end{aligned} \tag{B.6}$$

$$\boldsymbol{Q}_j^n = \boldsymbol{q}^n + \frac{h}{2}\left(K_{n+1/2}\boldsymbol{P}_j^{n,1} + S_{n+1/2}\boldsymbol{Q}_j^n + K_{n+1/2}'\boldsymbol{U}_j^{n,1} + S_{n+1/2}'\boldsymbol{V}_j^n\right). \tag{B.7}$$

The above scheme is called the direct approach for computing the gradient. While it provides an accurate way of calculating the discrete gradient, it requires the Strömer-Verlet scheme to be separately applied for each component of the gradient. A significantly more efficient way of computing all components of the discrete gradient follows by instead solving the discrete adjoint equation, see Section 4.1. However, the direct approach serves as an alternative way of calculating the gradient, which is easier to implement and is essential for testing the implementation of the discrete adjoint approach.

Appendix C. Derivation of the discrete adjoint scheme

We seek to determine a scheme for evolving the Lagrange multiplier (adjoint) variables to satisfy the first order optimality conditions (44). In the following, let $\delta_{r,s}$ denote the usual discrete Dirac delta function. Taking the derivative of (36) with respect to $\boldsymbol{u}_j^r$

$$0 = \frac{\partial\mathcal{L}_h}{\partial\boldsymbol{u}_j^r} = \frac{\partial\mathcal{J}_h}{\partial\boldsymbol{u}_j^r} - \left[(\boldsymbol{\mu}_j^n - \boldsymbol{\mu}_j^{n+1})\delta_{r,n} + \boldsymbol{\mu}_j^M\delta_{r,M} - (\boldsymbol{M}_j^{n,1} + \boldsymbol{M}_j^{n,2})\delta_{r,n}\right],$$

which gives the conditions

$$\boldsymbol{\mu}_j^M = \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{u}_j^M}, \quad \boldsymbol{\mu}_j^n - \boldsymbol{\mu}_j^{n+1} = \boldsymbol{M}_j^{n,1} + \boldsymbol{M}_j^{n,2}, \quad n = 0, 1, \ldots, M-1.$$

Similarly, differentiating (36) with respect to $\boldsymbol{v}_j^r$ gives

$$0 = \frac{\partial \mathcal{L}_h}{\partial \boldsymbol{v}_j^r} = \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{v}_j^r} - \left[(\boldsymbol{\nu}_j^n - \boldsymbol{\nu}_j^{n+1})\delta_{r,n} + \boldsymbol{\nu}_j^M \delta_{r,M} - (\boldsymbol{N}_j^{n,1} + \boldsymbol{N}_j^{n,2})\delta_{r,n}\right],$$

which leads to the conditions

$$\boldsymbol{\nu}_j^n - \boldsymbol{\nu}_j^{n+1} = \boldsymbol{N}_j^{n,1} + \boldsymbol{N}_j^{n,2}, \quad \boldsymbol{\nu}_j^M = \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{v}_j^M}.$$

Next we take the derivative of (36) with respect to $\boldsymbol{U}_j^{n,1}$,

$$\begin{aligned}
\frac{\partial \mathcal{L}_h}{\partial \boldsymbol{U}_j^{n,1}} &= \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{U}_j^{n,1}} - \sum_{i=1}^{6} \frac{\partial T_j^i}{\partial \boldsymbol{U}_j^{n,1}} = 0, \\
\frac{\partial T_j^1}{\partial \boldsymbol{U}_j^{n,1}} &= -\frac{h}{2} S_n^T \boldsymbol{\mu}_j^{n+1}, \\
\frac{\partial T_j^2}{\partial \boldsymbol{U}_j^{n,1}} &= -\frac{h}{2} K_{n+1/2}^T \boldsymbol{\nu}_j^{n+1}, \\
\frac{\partial T_j^3}{\partial \boldsymbol{U}_j^{n,1}} &= \boldsymbol{M}_j^{n,1}, \\
\frac{\partial T_j^4}{\partial \boldsymbol{U}_j^{n,1}} &= -\frac{h}{2} S_n^T \boldsymbol{M}_j^{n,2}, \\
\frac{\partial T_j^5}{\partial \boldsymbol{U}_j^{n,1}} &= -\frac{h}{2} K_{n+1/2}^T \boldsymbol{N}_j^{n,1}, \\
\frac{\partial T_j^6}{\partial \boldsymbol{U}_j^{n,1}} &= -\frac{h}{2} K_{n+1/2}^T \boldsymbol{N}_j^{n,2},
\end{aligned}$$

which, using the fact that $S_n^T = -S_n$ and $K_n^T = K_n$, we may write as

$$\boldsymbol{M}_j^{n,1} + \frac{h}{2} S_n \left(\boldsymbol{\mu}_j^{n+1} + \boldsymbol{M}_j^{n,2}\right) - \frac{h}{2} K_{n+1/2} \left(\boldsymbol{\nu}_j^{n+1} + \boldsymbol{N}_j^{n,1} + \boldsymbol{N}_j^{n,2}\right) = \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{U}_j^{n,1}}.$$

Repeating this procedure for the derivative with respect to $\boldsymbol{U}_j^{n,2}$ gives

$$\begin{aligned}
\frac{\partial \mathcal{L}_h}{\partial \boldsymbol{U}_j^{n,2}} &= \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{U}_j^{n,2}} - \sum_{i=1}^{6} \frac{\partial T_j^i}{\partial \boldsymbol{U}_j^{n,2}} = 0, \\
\frac{\partial T_j^1}{\partial \boldsymbol{U}_j^{n,2}} &= -\frac{h}{2} S_{n+1}^T \boldsymbol{\mu}_j^{n+1}, \\
\frac{\partial T_j^2}{\partial \boldsymbol{U}_j^{n,2}} &= -\frac{h}{2} K_{n+1/2}^T \boldsymbol{\nu}_j^{n+1}, \\
\frac{\partial T_j^4}{\partial \boldsymbol{U}_j^{n,2}} &= \boldsymbol{M}_j^{n,2} - \frac{h}{2} S_{n+1}^T \boldsymbol{M}_j^{n,2}, \\
\frac{\partial T_j^3}{\partial \boldsymbol{U}_j^{n,2}} &= \frac{\partial T_j^5}{\partial \boldsymbol{U}_j^{n,2}} = \frac{\partial T_j^6}{\partial \boldsymbol{U}_j^{n,2}} = 0,
\end{aligned}$$

which we may write compactly as

$$\boldsymbol{M}_j^{n,2} + \frac{h}{2} S_{n+1} \left(\boldsymbol{\mu}_j^{n+1} + \boldsymbol{M}_j^{n,2} \right) - \frac{h}{2} K_{n+1/2} \boldsymbol{\nu}_j^{n+1} = \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{U}_j^{n,2}}.$$

Taking the derivative of (36) with respect to $\boldsymbol{V}_j^{n,1}$ gives the set of equations

$$\begin{aligned}
\frac{\partial \mathcal{L}_h}{\partial \boldsymbol{V}_j^{n,1}} &= \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{V}_j^{n,1}} - \sum_{i=1}^{6} \frac{\partial T_j^i}{\partial \boldsymbol{V}_j^{n,1}} = 0, \\
\frac{\partial T_j^1}{\partial \boldsymbol{V}_j^{n,1}} &= \frac{h}{2} K_n^T \boldsymbol{\mu}_j^{n+1}, \\
\frac{\partial T_j^2}{\partial \boldsymbol{V}_j^{n,1}} &= -\frac{h}{2} S_{n+1/2}^T \boldsymbol{\nu}_j^{n+1}, \\
\frac{\partial T_j^3}{\partial \boldsymbol{V}_j^{n,1}} &= 0, \\
\frac{\partial T_j^4}{\partial \boldsymbol{V}_j^{n,1}} &= \frac{h}{2} K_n^T \boldsymbol{M}_j^{n,2}, \\
\frac{\partial T_j^5}{\partial \boldsymbol{V}_j^{n,1}} &= \boldsymbol{N}_j^{n,1} - \frac{h}{2} S_{n+1/2}^T \boldsymbol{N}_j^{n,1}, \\
\frac{\partial T_j^6}{\partial \boldsymbol{V}_j^{n,1}} &= -\frac{h}{2} S_{n+1/2}^T \boldsymbol{N}_j^{n,2},
\end{aligned}$$

which gives the condition

$$\boldsymbol{N}_j^{n,1} + \frac{h}{2} S_{n+1/2} \left(\boldsymbol{\nu}_j^{n+1} + \boldsymbol{N}_j^{n,1} + \boldsymbol{N}_j^{n,2} \right) + \frac{h}{2} K_n \left(\boldsymbol{\mu}_j^{n+1} + \boldsymbol{M}_j^{n,2} \right) = \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{V}_j^{n,1}}.$$

Similarly, taking the derivative with respect to $\boldsymbol{V}_j^{n,2}$ gives

$$\begin{aligned}
\frac{\partial \mathcal{L}_h}{\partial \boldsymbol{V}_j^{n,2}} &= \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{V}_j^{n,2}} - \sum_{i=1}^{6} \frac{\partial T_j^i}{\partial \boldsymbol{V}_j^{n,2}} = 0, \\
\frac{\partial T_j^1}{\partial \boldsymbol{V}_j^{n,2}} &= \frac{h}{2} K_{n+1}^T \boldsymbol{\mu}_j^{n+1}, \\
\frac{\partial T_j^2}{\partial \boldsymbol{V}_j^{n,2}} &= -\frac{h}{2} S_{n+1/2}^T \boldsymbol{\nu}_j^{n+1}, \\
\frac{\partial T_j^4}{\partial \boldsymbol{V}_j^{n,2}} &= \frac{h}{2} K_{n+1}^T \boldsymbol{M}_j^{n,2}, \\
\frac{\partial T_j^6}{\partial \boldsymbol{V}_j^{n,2}} &= \boldsymbol{N}_j^{n,2}, \\
\frac{\partial T_j^3}{\partial \boldsymbol{V}_j^{n,2}} &= \frac{\partial T_j^5}{\partial \boldsymbol{V}_j^{n,2}} = 0,
\end{aligned}$$

giving

$$\boldsymbol{N}_j^{n,2} + \frac{h}{2} S_{n+1/2} \boldsymbol{\nu}_j^{n+1} + \frac{h}{2} K_{n+1} \left(\boldsymbol{\mu}_j^{n+1} + \boldsymbol{M}_j^{n,2} \right) = \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{V}_j^{n,2}}.$$

In summary, the first order optimality conditions (44) are satisfied if the following equations hold:

$$\boldsymbol{\mu}_j^n - \boldsymbol{\mu}_j^{n+1} = \boldsymbol{M}_j^{n,1} + \boldsymbol{M}_j^{n,2}, \quad \boldsymbol{\mu}_j^M = \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{u}_j^M}, \tag{C.1}$$

$$\boldsymbol{\nu}_j^n - \boldsymbol{\nu}_j^{n+1} = \boldsymbol{N}_j^{n,1} + \boldsymbol{N}_j^{n,2}, \quad \boldsymbol{\nu}_j^M = \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{v}_j^M}, \tag{C.2}$$

$$\boldsymbol{M}_j^{n,1} + \frac{h}{2} S_n \left(\boldsymbol{\mu}_j^{n+1} + \boldsymbol{M}_j^{n,2}\right) - \frac{h}{2} K_{n+1/2} \left(\boldsymbol{\nu}_j^{n+1} + \boldsymbol{N}_j^{n,1} + \boldsymbol{N}_j^{n,2}\right) = \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{U}_j^{n,1}}, \tag{C.3}$$

$$\boldsymbol{M}_j^{n,2} + \frac{h}{2} S_{n+1} \left(\boldsymbol{\mu}_j^{n+1} + \boldsymbol{M}_j^{n,2}\right) - \frac{h}{2} K_{n+1/2} \boldsymbol{\nu}_j^{n+1} = \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{U}_j^{n,2}}, \tag{C.4}$$

$$\boldsymbol{N}_j^{n,1} + \frac{h}{2} S_{n+1/2} \left(\boldsymbol{\nu}_j^{n+1} + \boldsymbol{N}_j^{n,1} + \boldsymbol{N}_j^{n,2}\right) + \frac{h}{2} K_n \left(\boldsymbol{\mu}_j^{n+1} + \boldsymbol{M}_j^{n,2}\right) = \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{V}_j^{n,1}}, \tag{C.5}$$

$$\boldsymbol{N}_j^{n,2} + \frac{h}{2} S_{n+1/2} \boldsymbol{\nu}_j^{n+1} + \frac{h}{2} K_{n+1} \left(\boldsymbol{\mu}_j^{n+1} + \boldsymbol{M}_j^{n,2}\right) = \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{V}_j^{n,2}}. \tag{C.6}$$

We now consider the following change of variables

$$\boldsymbol{X}_j^n = \boldsymbol{\mu}_j^{n+1} + \boldsymbol{M}_j^{n,2}, \tag{C.7}$$

$$\boldsymbol{Y}_j^{n,1} = \boldsymbol{\nu}_j^{n+1} + \boldsymbol{N}_j^{n,1} + \boldsymbol{N}_j^{n,2}, \tag{C.8}$$

$$\boldsymbol{Y}_j^{n,2} = \boldsymbol{\nu}_j^{n+1}, \tag{C.9}$$

which, upon substitution into (C.3)-(C.6), gives the set of equations

$$\boldsymbol{M}_j^{n,1} + \frac{h}{2} S_n \boldsymbol{X}_j^n - \frac{h}{2} K_{n+1/2} \boldsymbol{Y}_j^{n,1} = \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{U}_j^{n,1}}, \tag{C.10}$$

$$\boldsymbol{M}_j^{n,2} + \frac{h}{2} S_{n+1} \boldsymbol{X}_j^n - \frac{h}{2} K_{n+1/2} \boldsymbol{Y}_j^{n,2} = \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{U}_j^{n,2}}, \tag{C.11}$$

$$\boldsymbol{N}_j^{n,1} + \frac{h}{2} S_{n+1/2} \boldsymbol{Y}_j^{n,1} + \frac{h}{2} K_n \boldsymbol{X}_j^n = \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{V}_j^{n,1}}, \tag{C.12}$$

$$\boldsymbol{N}_j^{n,2} + \frac{h}{2} S_{n+1/2} \boldsymbol{Y}_j^{n,2} + \frac{h}{2} K_{n+1} \boldsymbol{X}_j^n = \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{V}_j^{n,2}}. \tag{C.13}$$

By adding (C.10)-(C.11),

$$\boldsymbol{M}_j^{n,1} + \boldsymbol{M}_j^{n,2} = -\frac{h}{2}\left[\left(S_n + S_{n+1}\right)\boldsymbol{X}_j^n - K_{n+1/2}\left(\boldsymbol{Y}_j^{n,1} + \boldsymbol{Y}_j^{n,2}\right)\right] + \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{U}_j^{n,1}} + \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{U}_j^{n,2}}. \tag{C.14}$$

Similarly, by adding (C.12)-(C.13),

$$\boldsymbol{N}_j^{n,1} + \boldsymbol{N}_j^{n,2} = -\frac{h}{2}\left[S_{n+1/2}\left(\boldsymbol{Y}_j^{n,1} + \boldsymbol{Y}_j^{n,2}\right) + \left(K_n + K_{n+1}\right)\boldsymbol{X}_j^n\right] + \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{V}_j^{n,1}} + \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{V}_j^{n,2}}. \tag{C.15}$$

Thus, (C.1)-(C.2) can be rewritten as

$$\boldsymbol{\mu}_j^n - \boldsymbol{\mu}_j^{n+1} = -\frac{h}{2}\left[\left(S_n + S_{n+1}\right)\boldsymbol{X}_j^n - K_{n+1/2}\left(\boldsymbol{Y}_j^{n,1} + \boldsymbol{Y}_j^{n,2}\right)\right] + \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{U}_j^{n,1}} + \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{U}_j^{n,2}} \tag{C.16}$$

$$\boldsymbol{\nu}_j^n - \boldsymbol{\nu}_j^{n+1} = -\frac{h}{2}\left[S_{n+1/2}\left(\boldsymbol{Y}_j^{n,1} + \boldsymbol{Y}_j^{n,2}\right) + \left(K_n + K_{n+1}\right)\boldsymbol{X}_j^n\right] + \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{V}_j^{n,1}} + \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{V}_j^{n,2}} \tag{C.17}$$

By combining $\boldsymbol{X}_j^n = \boldsymbol{\mu}_j^{n+1} + \boldsymbol{M}_j^{n,2}$ and (C.11),

$$\boldsymbol{X}_j^n = \boldsymbol{\mu}_j^{n+1} - \frac{h}{2}S_{n+1}\boldsymbol{X}_j^n + \frac{h}{2}K_{n+1/2}\boldsymbol{Y}_j^{n,2} + \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{U}_j^{n,2}}. \tag{C.18}$$

Similarly, by combining $\boldsymbol{Y}_j^{n,1} = \boldsymbol{\nu}_j^{n+1} + \boldsymbol{N}_j^{n,1} + \boldsymbol{N}_j^{n,2}$ and (C.15),

$$\boldsymbol{Y}_j^{n,1} = \boldsymbol{\nu}_j^{n+1} - \frac{h}{2}\left[S_{n+1/2}\left(\boldsymbol{Y}_j^{n,1} + \boldsymbol{Y}_j^{n,2}\right) + \left(K_n + K_{n+1}\right)\boldsymbol{X}_j^n\right] + \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{V}_j^{n,1}} + \frac{\partial \mathcal{J}_h}{\partial \boldsymbol{V}_j^{n,2}}. \tag{C.19}$$

The time-stepping scheme is completed by the relation

$$\boldsymbol{Y}^{n,2} = \boldsymbol{\nu}_j^{n+1}. \tag{C.20}$$

The scheme (C.16)-(C.20) may be written in the form of Lemma 1 by defining the slopes according to (48)-(51). This completes the proof of the lemma.

Appendix D. Computing the gradient of the discrete objective function

Given a solution that satisfies the saddle point conditions of (43) and (44), the gradient of $\mathcal{L}_h(\boldsymbol{\alpha})$ satisfies

$$\frac{d\mathcal{L}_h}{d\alpha_r} = \frac{\partial \mathcal{J}_{1h}}{\partial \alpha_r}(\boldsymbol{u}, \boldsymbol{v}) + \frac{\partial \mathcal{J}_{2h}}{\partial \alpha_r}(\boldsymbol{U}, \boldsymbol{V}), \quad r = 1, 2, \ldots, D.$$

The gradient of $\mathcal{L}_h$ with respect to $\boldsymbol{\alpha}$ only gets a contribution from the terms in T_j^q that involve the matrices K and S. Let $S'_n = \partial S/\partial \alpha_r(t_n)$ and $K'_n = \partial K/\partial \alpha_r(t_n)$. We have,

$$\begin{aligned}
\frac{\partial T_j^1}{\partial \alpha_r} &= -\frac{h}{2}\sum_{n=0}^{M-1} \left\langle S'_n \boldsymbol{U}_j^{n,1} - K'_n \boldsymbol{V}_j^{n,1} + S'_{n+1}\boldsymbol{U}_j^{n,2} - K'_{n+1}\boldsymbol{V}_j^{n,2}, \boldsymbol{\mu}_j^{n+1}\right\rangle_2, \\
\frac{\partial T_j^2}{\partial \alpha_r} &= -\frac{h}{2}\sum_{n=0}^{M-1} \left\langle K'_{n+1/2}\left(\boldsymbol{U}_j^{n,1} + \boldsymbol{U}_j^{n,2}\right) + S'_{n+1/2}\left(\boldsymbol{V}_j^{n,1} + \boldsymbol{V}_j^{n,2}\right), \boldsymbol{\nu}_j^{n+1}\right\rangle_2, \\
\frac{\partial T_j^3}{\partial \alpha_r} &= 0, \\
\frac{\partial T_j^4}{\partial \alpha_r} &= -\frac{h}{2}\sum_{n=0}^{M-1} \left\langle S'_n \boldsymbol{U}_j^{n,1} - K'_n \boldsymbol{V}_j^{n,1} + S'_{n+1}\boldsymbol{U}_j^{n,2} - K'_{n+1}\boldsymbol{V}_j^{n,2}, \boldsymbol{M}_j^{n,2}\right\rangle_2, \\
\frac{\partial T_j^5}{\partial \alpha_r} &= -\frac{h}{2}\sum_{n=0}^{M-1} \left\langle K'_{n+1/2}\boldsymbol{U}_j^{n,1} + S'_{n+1/2}\boldsymbol{V}_j^{n,1}, \boldsymbol{N}_j^{n,1}\right\rangle_2, \\
\frac{\partial T_j^6}{\partial \alpha_r} &= -\frac{h}{2}\sum_{n=0}^{M-1} \left\langle K'_{n+1/2}\boldsymbol{U}_j^{n,1} + S'_{n+1/2}\boldsymbol{V}_j^{n,1}, \boldsymbol{N}_j^{n,2}\right\rangle_2.
\end{aligned}$$

We note that

$$\frac{\partial (T_j^5 + T_j^6)}{\partial \alpha_r} = -\frac{h}{2}\sum_{n=0}^{M-1} \left\langle K'_{n+1/2}\boldsymbol{U}_j^{n,1} + S'_{n+1/2}\boldsymbol{V}_j^{n,1}, \boldsymbol{N}_j^{n,1} + \boldsymbol{N}_j^{n,2}\right\rangle_2.$$

Let $\boldsymbol{X}_j^n$ and $\boldsymbol{Y}_j^{n,i}$ be defined by (C.7)-(C.9). We have,

$$\begin{aligned}
\frac{\partial T_j^4}{\partial \alpha_r} &= -\frac{h}{2}\sum_{n=0}^{M-1} \left\langle S'_n \boldsymbol{U}_j^{n,1} - K'_n \boldsymbol{V}_j^{n,1} + S'_{n+1}\boldsymbol{U}_j^{n,2} - K'_{n+1}\boldsymbol{V}_j^{n,2}, \boldsymbol{X}_j^n - \boldsymbol{\mu}_j^{n+1}\right\rangle_2, \\
\frac{\partial (T_j^5 + T_j^6)}{\partial \alpha_r} &= -\frac{h}{2}\sum_{n=0}^{M-1} \left\langle K'_{n+1/2}\boldsymbol{U}_j^{n,1} + S'_{n+1/2}\boldsymbol{V}_j^{n,1}, \boldsymbol{Y}_j^{n,1} - \boldsymbol{\nu}_j^{n+1}\right\rangle_2.
\end{aligned}$$

Thus,

$$\frac{\partial(T_j^1 + T_j^4)}{\partial \alpha_r} = -\frac{h}{2} \sum_{n=0}^{M-1} \left\langle S'_n \boldsymbol{U}_j^{n,1} - K'_n \boldsymbol{V}_j^{n,1} + S'_{n+1} \boldsymbol{U}_j^{n,2} - K'_{n+1} \boldsymbol{V}_j^{n,2}, \boldsymbol{X}_j^n \right\rangle_2 .$$

Furthermore, from the relation (C.9),

$$\begin{aligned} \frac{\partial(T_j^2 + T_j^5 + T_j^6)}{\partial \alpha_r} = &- \frac{h}{2} \sum_{n=0}^{M-1} \left\langle K'_{n+1/2} \boldsymbol{U}_j^{n,1} + S'_{n+1/2} \boldsymbol{V}_j^{n,1}, \boldsymbol{Y}_j^{n,1} \right\rangle_2 \\ &- \frac{h}{2} \sum_{n=0}^{M-1} \left\langle K'_{n+1/2} \boldsymbol{U}_j^{n,2} + S'_{n+1/2} \boldsymbol{V}_j^{n,2}, \boldsymbol{Y}_j^{n,2} \right\rangle_2 , \end{aligned}$$

We can further simplify the expressions by recognizing that $\boldsymbol{V}^{n,1} = \boldsymbol{V}^{n,2}$. By collecting the terms,

$$\begin{aligned} \frac{\partial \mathcal{L}_h}{\partial \alpha_r} = \frac{h}{2} \sum_{j=0}^{E-1} \sum_{n=0}^{M-1} \Big(&\left\langle S'_n \boldsymbol{U}_j^{n,1} + S'_{n+1} \boldsymbol{U}_j^{n,2} - (K'_n + K'_{n+1}) \boldsymbol{V}_j^{n,1}, \boldsymbol{X}_j^n \right\rangle_2 \\ &+ \left\langle K'_{n+1/2} \boldsymbol{U}_j^{n,1} + S'_{n+1/2} \boldsymbol{V}_j^{n,1}, \boldsymbol{Y}_j^{n,1} \right\rangle_2 \\ &+ \left\langle K'_{n+1/2} \boldsymbol{U}_j^{n,2} + S'_{n+1/2} \boldsymbol{V}_j^{n,1}, \boldsymbol{Y}_j^{n,2} \right\rangle_2 \Big) . \end{aligned}$$

This completes the proof of the lemma.

References

[1] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–89, 2017.

[2] A. Borz, G. Ciarmella, and M. Sprengel. *Formulation and Numerical Solution of Quantum Control Problems*. Computational science and engineering. SIAM, 2017.

[3] T. Caneva, T. Calarco, and S. Montangero. Chopped random-basis quantum optimization. *Physical Review A*, 84(2), Aug 2011.

[4] P. de Fouquieres, S.G. Schirmer, S.J. Glaser, and Ilya Kuprov. Second order gradient ascent pulse engineering. *Journal of Magnetic Resonance*, 212(2):412417, Oct 2011.

[5] L. Emsley and G. Bodenhausen. Gaussian pulse cascades: New analytical functions for rectangular selective inversion and in-phase excitation in NMR. *Chem. Phys.*, 165(6):469–476, 1989.

[6] B. Ewing, S. J. Glaser, and G. P. Drobny. Development and optimization of shaped NMR pulses for the study of coupled spin systems. *Chem. Phys.*, 147:121–129, 1990.

[7] William W. Hager. Runge-Kutta methods in optimal control and the transformed adjoint system. *Numerische Mathematik*, 87(2):247–282, Dec 2000.

[8] E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration*. Number 31 in Springer series in computational mathematics. Springer-Verlag, Heidelberg, 2nd edition, 2006.

[9] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbruggen, and S. Glaser. Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms. *J. Magnetic Resonance*, 172:296–305, 2005.

[10] N. Leung, M. Abdelhafez, Jens Koch, and D. Schuster. Speedup for quantum optimal control from automatic differentiation based on graphics processing units. *Phys. Rev. A*, 95:0432318, 2017.

[11] Dennis Lucarelli. Quantum optimal control via gradient ascent in function space and the time-bandwidth quantum speed limit. *Physical Review A*, 97(6), Jun 2018.

[12] S. Machnes, E. Assmat, D. Tannor, and F. K. Wilhelm. Tunable, flexible, and efficient optimization of control pulses for practical qubits. *Physical Review Letters*, 120(15), Apr 2018.

[13] M. Nielsen and I. Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2000.

[14] J. Nocecdal and S. J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

[15] Sina Ober-Blöbaum. *Discrete mechanics and optimal control*. PhD thesis, University of Paderborn, 2008.

[16] Sina Ober-Blöbaum, Oliver Junge, and Jerrold E. Marsden. Discrete mechanics and optimal control: an analysis. *ESAIM: Control, Optimisation and Calculus of Variations*, 17(2):322–352, 2011.

[17] J. M. Sanz-Serna. Symplectic Runge–Kutta schemes for adjoint equations, automatic differentiation, optimal control, and more. *SIAM Review*, 58(1):3–33, 2016.

[18] Yunong Shi, Nelson Leung, Pranav Gokhale, Zane Rossi, David I. Schuster, Henry Hoffmann, and Frederic T. Chong. Optimized compilation of aggregated instructions for realistic quantum computers. *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems - ASPLOS 19*, 2019.

[19] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, Mar 2006.

[20] D. B. Zax, G. Goelman, and S. Vega. Amplitude-modulated composite pulses. *J. Magn. Reson.*, 80(2):375–382, 1988.