A Generalized Suzuki–Trotter Type Method in Optimal Control of Coupled Schrödinger Equations

Felix Henneke * Manfred Liebmann

January 20, 2015

We propose a generalized Suzuki–Trotter (GST) type method for the solution of an optimal control problem for quantum molecular systems. The control of such systems gives rise to a minimization problem with constraints given by a system of coupled Schrödinger equations. The bottleneck of the corresponding minimization methods is the solution of time-dependent Schrödinger equations. To solve the Schrödinger equations we use the GST framework to obtain an explicit polynomial approximation of the matrix exponential function. The GST method almost exclusively uses the action of the Hamiltonian and is therefor efficient and easy to implement for a variety of quantum systems. Following a first-discretize-then-optimize approach we derive the correct discrete representation of the gradient and the Hessian. The derivatives can naturally be expressed in the GST framework and can therefor be efficiently computed. By recomputing the solutions of the Schrödinger equations instead of saving the whole time evolution, we are able to significantly reduce the memory requirements of the method at the cost of additional computations. This makes first and second order optimization methods viable for large scale problems. In numerical experiments we compare the performance of different first and second optimization methods using the GST method. We observe fast local convergence of second order methods.

1 Introduction

One of the goals of quantum control is to steer a quantum system with an external control from an initial to a target configuration [5]. Applications range from the manipulation

*The first author gratefully acknowledges support from the International Research Training Group IGDK1754, funded by the German Science Foundation (DFG).

of chemical reactions [16] to the experimental realization of quantum computers [4]. In this paper we consider a problem in molecular control, i.e. the design of a laser pulse that changes the configuration of atomic nuclei by inducing transitions in the electronic structure [10]. Mathematically this process can be described by a system of coupled Schrödinger equations. In many cases it is known that there exist controls which accomplish the control goal [21, 2] and the qualitative behavior of those controls is known in some simple limiting cases [3]. Nevertheless, the shape of the pulse for realistic problems is extremely difficult to guess. To overcome this difficulty the quantum control problem typically is reformulated as an optimal quantum control problem, that is equivalent to an optimization problem with PDE constraints [15, 9, 23]. We study two cases of control costs using L^2 and H_0^1 norms [22], which correspond to the penalization of the laser energy and the penalization of strong oscillations of the laser field, respectively.

The bottleneck for the numerical solution of the optimal quantum control problem is the solutions of the time-dependent Schrödinger equation. The evaluation of the functional of the optimal quantum control problem, the associated gradient, and the evaluation of the action of the Hessian require multiple solutions of the time-dependent Schrödinger equation. Therefor a fast solution method for the time-dependent Schrödinger equation is of major importance. The time evolution operator of a quantum system is essentially the exponential of the Hamiltonian of the quantum system. The classical Suzuki–Trotter method approximates an exponential operator with a product ansatz of computationally simple exponential operators [19, 18]. The generalized Suzuki–Trotter (GST) method instead uses a product ansatz of lower order approximations of the exponential operator for the construction of the higher order approximation of the exponential operator [11]. Thus computationally inexpensive linear approximations for the exponential operator can be used as building blocks for the generalized Suzuki–Trotter method. The resulting explicit scheme then generates a polynomial approximation of the exponential operator. In practice only the action of the Hamiltonian is required to construct an approximation for the time evolution operator. Therefore the GST method is easily applicable to a wide variety of Schrödinger operators.

The optimal quantum control problem is typically solved using a gradient based method [15, 8] or self-consistent iterations like the monotonic method [24, 12, 9]. Second order methods like Newtons method are less frequently used [23, 7]. A problem of second order methods is that evaluation of the action of the Hessian involves the solution of inhomogeneous Schrödinger equations. Inhomogeneous Schrödinger equations are typically not of physical relevance and thus most standard solvers do not support this setting. We shall outline a simple approach to tackle this problem.

For our numerical experiments we implemented a first-discretize-then-optimize (DTO) approach using the GST method. It is similar to the calculus for directional derivatives in [13]. We derived expressions for the first and second order derivatives of the time stepping scheme with respect to the control. These expressions fit naturally within the GST framework and thus can be computed efficiently. In particular the inhomogeneous Schrödinger equation for the action of the Hessian can be computed efficiently using the GST method.

In the spirit of a low memory implementation for the functional, gradient, and Hessian

evaluation the algorithms do not store the whole time evolution of the wave function in memory, but instead employ additional solutions of the Schrödinger equation when required. The huge decrease in memory requirements, at the cost of additional solutions of the Schrödinger equation, is a crucial step to make the algorithm practical for large scale problems.

We conducted numerical experiments with different first and second order optimization methods using the functional, gradient, and Hessian evaluation implemented in the GST framework. The problem setting for the optimal quantum control problem are two coupled one-dimensional Schrödinger equations and a cost functional including either an L^2 or H_0^1 norm penalization of the laser pulse. Comparing the DTO with the OTD approach we found that DTO generates better approximations of the derivatives of the cost functional and is therefore more reliable in generating descent directions. This fact also justifies the increased cost of gradient and Hessian evaluations in the DTO approach. For our example we observed fast local convergence of the trust region globalized Newton method. The norm of the gradient at the calculated optimum was several orders of magnitude smaller compared to only first order gradient based methods. However, the gradient based methods were faster in reaching a small neighborhood of a local optimum compared to the trust region globalized Newton method. It might be useful to use a hybrid method to combine the advantages of both methods in future investigations.

For our model problem the choice of the cost term has a major influence on the qualitative behavior of the resulting optimal control function. Comparing the frequency spectrums of the controls for the L^2 and H_0^1 cost terms we see that the controls trigger different physical mechanisms. In the L^2 case the desired oscillating behavior that induces transitions in the electronic structure was visible. For the H_0^1 case the control only contains low frequency components and does not exploit the coupling in the system of Schrödinger equations. This effect might also be sensitive to the choice of the cost parameter. Using a H_0^1 cost term might therefor be unfeasible for the model problem.

The paper is structured as follows. In Section 2 we give a precise formulation of the optimal quantum control problem and derive representations for derivatives and optimality conditions in the continuous setting. The generalized Suzuki–Trotter (GST) method will be introduced and analyzed in Section 3 and we also state the fully discrete problem and carry out the DTO approach. In Section 4 we present the results of the numerical experiments.

2 Optimal Quantum Control Problem

We will first introduce the optimal quantum control problem for a coupled system of Schrödinger equations for the description of molecules and then motivate the typical optimal control formulation as a minimization problem with a cost functional using L^2 or H_0^1 norms for the control penalization. We will then present expressions for the first and second order derivatives of the functional.

2.1 Model Quantum System

The time evolution of a molecule in a laser field can be described by a time-dependent Schrödinger equation. The full molecular Schrödinger equation is very high dimensional, even for small molecules and therefore the equation is approximated in a suitable way. The standard approach is a multilevel Born–Oppenheimer approximation: For each of a small number M of low electronic states, an effective d-dimensional Schrödinger equation is derived for the nuclei, where the electrons create an effective potential, the potential energy surface (PES). The nuclei move on these PES. This leads to a system of coupled Schrödinger equations.

The equations are coupled due to intersections of the potential energy surfaces and the dipole coupling of the time-dependent electric field of the laser. The usual ground state approximation, which corresponds to $M = 1$, is not appropriate in this setting, since excited electronic states might become populated in the presence of the laser interaction. However, we will assume that the field intensity is small enough to allow for a first order approximation of the field–matter coupling, see [5] for a justification and [20] for higher order approximations.

The equation for the nuclei on M PESs then reads

$$i\partial_t\psi(t) = (\hat{T} + \hat{V} + u(t)\hat{\mu})\psi(t), \quad \psi(0) = \psi_0.$$

Here, $\psi(t) = (\psi_1(t), \dots, \psi_M(t))^T$ and $\psi_0 = (\psi_{0,1}, \dots, \psi_{0,M})^T$ are vectors of complex functions describing the wave functions on the different PESs, $\hat{T} = \operatorname{diag}(T, \dots, T)$ is an operator-valued diagonal matrix of kinetic energies, $\hat{V} = (V_{ij})$ is an operator-valued matrix of potentials, $\hat{\mu} = (\mu_{ij})$ is the operator-valued matrix of transition dipole operators of the electric field of the laser and $u(t)$ is the real-valued electric field strength.

In the following we will use the abstract formulation

$$i\partial_t\psi(t) = (H_0 + u(t)H_1)\psi(t), \quad \psi(0) = \psi_0. \tag{1}$$

for this initial value problem. Here ψ and ψ_0 are vectors of complex functions and H_0 and H_1 are self-adjoint matrices of operators in some Hilbert space. The uncontrolled system is governed by the Hamiltonian H_0 and u introduces an additional coupling via the transition operator H_1. In the context of optimal control, (1) is referred to as *state equation*. Solutions of (1) exist under reasonable assumptions on H_0, H_1 and u, see [1, Theorem 2.5].

2.2 Control Problem

A typical goal in molecular quantum control is to steer a system from an initial state ψ_0 into a subspace X, which consists of states with a desired property. Interesting choices for X include subspaces of common electronic structure or localization in space. The corresponding control problem can be formulated as follows: Find a control u in a set of admissible controls $\mathcal{U}$ such that the corresponding solution of (1) satisfies $\psi(T) \in X$ for some $T > 0$. The bilinear structure of this control problem allows for a rich mathematical theory. Unfortunately, in general this problem does not have a solution, see [1] for the

fundamental negative result on controllability in infinite dimensions. Even in cases where it is known that suitable controls exist, the shape of the controls often cannot be derived analytically. The large time behavior of the control system, however, highlights the importance of particular frequencies in the control [3]. One obtains Bohr's frequency law, which states that the frequencies in the control are proportional to energy differences in the quantum system. For finite time horizons the problem is more complicated. We will therefor reformulate the control problem as an optimal control problem. It can be derived as follows. As a first step we fix the final time $T > 0$ and consider the minimization of $\operatorname{dist}(\psi(T), X)$ where ψ and u solve (1). This can equivalently be written as

$$\text{Minimize} \quad \frac{1}{2}\langle \psi(T), \mathcal{O}\psi(T)\rangle \quad \text{s. t. (1)}, \tag{2}$$

where $\mathcal{O} = I - P_X$ and P_X is the orthogonal projection on X. This problem is still not well-posed, by the same argument as above.

There are several ways to guarantee the existence of solutions for problem (2). Among them are constraints on the admissible controls $\mathcal{U}$ like thresholds for the amplitude or restriction to finitely many degrees of freedom. We will instead regularize the problem with a Tikhonov term. This means that we add a term to the cost functional that penalizes the growth of the control in a suitable way. Then the problem reads

$$\text{Minimize} \quad J(\psi, u) = \frac{1}{2}\langle \psi(T), \mathcal{O}\psi(T)\rangle + \frac{\alpha}{2}\|u\|_{\mathcal{U}}^2 \quad \text{s. t. (1)}. \tag{3}$$

Here $\mathcal{U}$ is equipped with a scalar product $\langle \cdot, \cdot \rangle_{\mathcal{U}}$ and $\|u\|_{\mathcal{U}} = \sqrt{\langle u, u\rangle_{\mathcal{U}}}$ is the norm in $\mathcal{U}$, and $\alpha > 0$ is a regularization parameter. The norm in $\mathcal{U}$ should penalize unfavorable behavior of the control u. Problem (3) has solutions under appropriate assumptions, see [9]. Here $\mathcal{O}$ can also be a more general observable.

The observable $\mathcal{O}$ and a time horizon T are typically given by the concrete application. But there is some freedom in the choice of the control space $\mathcal{U}$ and the regularization parameter α. Restrictions on $\mathcal{U}$ might be of physical nature or address issues of the experimental implementation. We will consider the two cases $\mathcal{U} = L^2$ and $\mathcal{U} = H_0^1$. The regularization with an L^2 cost term goes back to the first papers on optimal control for quantum molecular systems [15]. It is used for its simplicity and seeming naturalness. The term $\|u\|_{L^2}^2$ is proportional to the energy of the electric field. Its boundedness is a physical necessity. However, optimal pulses computed for this choice of the cost term seem to suffer from a highly irregular shape and from oscillations which prohibit direct implementations in experiments. The H_0^1 regularization does not suffer from these oscillations since it penalizes them. Additionally, controls in H_0^1 satisfy homogeneous Dirichlet boundary conditions that are suitable for a laser pulse that gets switched on and off. A modified H_0^1 norm is related to the physical work done in the system [7]. An H_0^1 cost term is successfully applied in applications [8]. However, for the problem at hand, it introduces an undesired behavior of the control in the sense that high frequencies are penalized even though they are desirable due to Bohr's law. In the next section we will formally derive the optimality systems for the L^2 and H_0^1 case.

Instead of the constrained optimization problem (3) we can equivalently study the unconstrained problem

$$\text{Minimize} \quad j(u) = J(\psi(u), u), \tag{4}$$

where $\psi(u)$ denotes the solution of (1) for a given control u. In this context, j is called reduced cost functional since the explicit dependence of the cost functional on ψ is dropped. Unconstrained problems are sometimes easier to study from a theoretical point of view. From the numerical point of view, it also reduces the degrees of freedoms of the problem and it is possible to directly apply techniques from unconstrained optimization theory. Note, however, that although equation (1) is linear in the state and linear in the control, the resulting bilinear control problem is non-convex. In the next section we will study derivatives and the optimality system for the reduced problem.

2.3 Derivatives and Optimality Conditions

In this subsection we will give optimality conditions for the problem (4) and provide representations of the gradient and Hessian of the reduced cost functional j. We will see that the different choices of the control space $\mathcal{U}$ will lead to a different qualitative behavior of the local optimizers. The explicit representations of the derivatives of j can be used to set up an optimization method using a first-optimize-then-discretize approach.

Representations for the derivatives of j can formally be derived using a Lagrange ansatz, see [23]. We will only summarize the results. To start we denote by $\varphi(u)$ the unique solution of

$$i\partial_t \varphi(t) = (H_0 + u(t)H_1)\varphi(t), \quad \varphi(T) = \mathcal{O}\psi(T). \tag{5}$$

It can be understood as an equation backwards in time with a terminal value provided by $\psi(u)$. The function φ is called *adjoint state* and (5) is refered to as *adjoint equation*. We will also need the derivatives of the state and adjoint state with respect to the control. One can derive the following formulas for $\psi'(u)(\delta u)$ and $\varphi'(u)(\delta u)$, see [1]. They are the solutions of the inhomogeneous equations

$$i\partial_t \psi' = (H_0 + uH_1)\psi' + \delta u H_1 \psi, \quad \psi'(0) = 0, \tag{6}$$

$$i\partial_t \varphi' = (H_0 + uH_1)\varphi' + \delta u H_1 \varphi, \quad \varphi'(T) = \mathcal{O}\psi'(T), \tag{7}$$

The derivatives of j can now be expressed in terms of ψ, φ, ψ' and φ'.

Proposition 1. *Let $u, \delta u, \tau u \in \mathcal{U}$. The first and second derivative of j are given by*

$$j'(u)(\delta u) = \int_0^T \langle (-iH_1)^* \big(\varphi(u)\big)(t), \big(\psi(u)\big)(t)\rangle_{\mathcal{H}} \delta u(t)\,\mathrm{d}t + \alpha \langle u, \delta u\rangle_{\mathcal{U}} \tag{8}$$

and

$$\begin{aligned} j''(u)(\delta u, \tau u) = \int_0^T & \Big(\langle (-iH_1)^* \big(\varphi'(u)(\delta u)\big)(t), \big(\psi(u)\big)(t)\rangle_{\mathcal{H}} \\ & + \langle (-iH_1)^* \big(\varphi(u)\big)(t), \big(\psi'(u)(\delta u)\big)(t)\rangle_{\mathcal{H}} \Big) \tau u(t)\,\mathrm{d}t \\ & + \alpha \langle \delta u, \tau u\rangle_{\mathcal{U}}. \end{aligned} \tag{9}$$

The form of the gradient and the action of the Hessian depend on the Hilbert space structure of $\mathcal{U}$. Let us first consider the case $\mathcal{U} = L^2(0,T)$. Then (8) implies

$$\nabla j(u) = \alpha u + \langle (-iH_1)^* \varphi(u), \psi(u) \rangle_{\mathcal{H}}, \tag{10}$$

and in the optimum $\bar{u}$ we have

$$\bar{u} = -\frac{1}{\alpha} \langle (-iH_1)^* \varphi(\bar{u}), \psi(\bar{u}) \rangle. \tag{11}$$

For the action of the Hessian we obtain

$$\nabla^2 j(u) \cdot \delta u = \alpha \delta u + \langle (-iH_1)^* \varphi'(u)(\delta u), \psi(u) \rangle_{\mathcal{H}} + \langle (-iH_1)^* \varphi(u), \psi'(u)(\delta u) \rangle_{\mathcal{H}}. \tag{12}$$

For the case $\mathcal{U} = H_0^1(0,T)$ we get

$$\int_0^T \partial_t (\nabla j(u) - \alpha u) \partial_t \delta u \, \mathrm{d}t = \int_0^T \langle (-iH_1)^* \varphi(u), \psi(u) \rangle_{\mathcal{H}} \delta u \, \mathrm{d}t$$

for all $\delta u \in H_0^1$. This means that $z = \nabla j(u) - \alpha u$ is the weak solution of

$$\begin{aligned} -\Delta z &= \langle (-iH_1)^* \varphi(u), \psi(u) \rangle_{\mathcal{H}}, \\ z(0) &= z(T) = 0, \end{aligned} \tag{13}$$

and in the optimum $\bar{u}$ we have

$$\begin{aligned} -\Delta \bar{u} &= -\frac{1}{\alpha} \langle (-iH_1)^* \varphi(\bar{u}), \psi(\bar{u}) \rangle_{\mathcal{H}}, \\ \bar{u}(0) &= \bar{u}(T) = 0. \end{aligned} \tag{14}$$

For the action of the Hessian we obtain

$$\int_0^T \partial_t (\nabla^2 j(u) \cdot \delta u - \alpha \delta u) \partial_t \tau u \, \mathrm{d}t = \int_0^T \begin{array}{c} \langle (-iH_1)^* \varphi'(u)(\delta u), \psi(u) \rangle_{\mathcal{H}} \tau u \\ + \langle (-iH_1)^* \varphi(u), \psi'(u)(h_1) \rangle_{\mathcal{H}} \tau u \, \mathrm{d}t \end{array}$$

for all $\tau u \in \mathcal{U}$. This means that $z = \nabla^2 j(u) \cdot \delta u - \alpha \delta u$ is the weak solution of

$$\begin{aligned} -\Delta z &= \langle (-iH_1)^* \varphi'(u)(\delta u), \psi(u) \rangle_{\mathcal{H}} + \langle (-iH_1)^* \varphi(u), \psi'(u)(\delta u) \rangle_{\mathcal{H}}, \\ z(0) &= z(T) = 0. \end{aligned} \tag{15}$$

Comparing (11) and (14) we see the smoothing effect of the H_0^1 regularization compared to the L^2 regularization. The function $\langle (-iH_1)^* \varphi(\bar{u}), \psi(\bar{u}) \rangle_{\mathcal{H}}$ typically oscillates in time. In (11) we can see that the optimal control for $\mathcal{U} = L^2$ inherits those oscillations. Equation (14) implies that for $\mathcal{U} = H_0^1$ the optimal control is a smoothed version of the oscillating function. In our application where oscillating controls are expected and necessary this can lead to undesirable optimal controls that have large low frequency contributions.

3 Numerical Approach

We discuss the numerical approach for the solution of the optimal quantum control problem. The first part introduces the generalized Suzuki–Trotter (GST) method and in the second part the structural properties of the GST method itself as well as its behavior under differentiation with respect to the control are presented. Then we discretize the optimal control problem and derive representations of the discrete derivatives that are efficiently computable. We close with a discussion of optimization algorithms.

3.1 Generalized Suzuki–Trotter Method

The generalized Suzuki–Trotter method was introduced in [11] and is an extension of the approximation scheme for the exponential operator presented in [19, 18]. The main idea of the method is to approximate the exponential operator in a recursive scheme as a product of lower order approximations in the context of a Banach algebra $\mathfrak{B}$. Throughout this section $\mathfrak{B}$ denotes a C* algebra.

Definition 2. Let $B \in \mathfrak{B}^{\mathbb{C}\times\mathcal{U}^h}$, $u^h \in \mathcal{U}^h$, $z \in \mathbb{C}$ and $m \in \mathbb{N}$ then the generalized Suzuki–Trotter (GST) operator $\mathcal{Q}_m \colon \mathfrak{B}^{\mathbb{C}\times\mathcal{U}^h} \to \mathfrak{B}^{\mathbb{C}\times\mathcal{U}^h}$ is recursively defined by

$$\begin{aligned}\mathcal{Q}_1(B)(z,u^h) &= B(z,u^h),\\ \mathcal{Q}_m(B)(z,u^h) &= \mathcal{Q}_{m-1}(B)(p_m z,u^h)\cdot\mathcal{Q}_{m-1}(B)(\bar{p}_m z,u^h), \quad m>1,\end{aligned}$$

where the coefficients $p_m \in \mathbb{C}$ satisfy the relations

$$p_m + \bar{p}_m = 1, \quad p_m^m + \bar{p}_m^m = 0\ . \tag{16}$$

The equations for the coefficients p_m in Definition 2 can be solved analytically with

$$p_m = \frac{1}{1+e^{-i\pi/m}} = \frac{1}{2} + \frac{i}{2}\tan(\pi/2m)\ . \tag{17}$$

The operator $\mathcal{Q}_m$ can also be defined on $\mathfrak{B}^{\mathbb{C}}$ and then extended pointwise to $\mathfrak{B}^{\mathbb{C}\times\mathcal{U}^h}$ by $\mathcal{Q}_m(B)(z,u^h) = \mathcal{Q}_m(B(\cdot,u^h))(z)$. In practice we will not use the GST method with $z \in \mathbb{C}$ but only for $t \in \mathbb{R}$. However, for the theory it is sometimes useful to allow complex time steps. The GST operator can be written as a product,

$$\mathcal{Q}_m(B)(z,u^h) = \prod_{k=1}^{2^{m-1}} B(\alpha_k z,u^h), \tag{18}$$

where α_k is the $(m-1)$-fold product of the coefficients p_j or $\overline{p_j}$ with $1 < j \le m$. The following result states the fundamental approximation property of the GST method.

Theorem 3. *Let $A \in \mathfrak{B}^{\mathcal{U}^h}$, $B \in \mathfrak{B}^{\mathbb{C}\times\mathcal{U}^h}$, $u^h \in \mathcal{U}^h$, $z \in \mathbb{C}$ and $m \in \mathbb{N}$. If the map $z \mapsto B(z,u^h)$ is analytic and*

$$B(z,u^h) - \exp(zA(u^h)) = o(z)$$

then the map $z \mapsto \mathcal{Q}_m(B)(z, u^h)$ *is analytic and*

$$\mathcal{Q}_m(B)(z, u^h) - \exp(zA(u^h)) = o(z^m).$$

A proof of the theorem is given in [11]. A good choice for the first order approximation operator B is obviously the linear approximation

$$B(z, u^h) := I + zA(u^h). \tag{19}$$

For the linear approximation we obtain an explicit error bound for the approximation of the exponential, as well as a restriction on the norm of the argument for the polynomial approximation.

Proposition 4. *Let* $A \in \mathcal{B}^{\mathcal{U}^h}$, $B \in \mathfrak{B}^{\mathbb{C}\times\mathcal{U}^h}$ *and* $B(z, u^h) := I + zA(u^h)$ *a linear approximation and* $m \in \mathbb{N}$. *If* $s := \frac{3}{2^{m-1}}\|zA(u^h)\| < 1$ *then*

$$\|\mathcal{Q}_m(B)(z, u^h) - \exp(zA(u^h))\| \leq (\exp(2^{m-1}\sum_{k=m+1}^{\infty}\frac{s^k}{k}) - 1)(1+s)^{m-1}.$$

For specific operators A this leads to a restriction of the step size $t \in \mathbb{R}$ of the explicit time stepping scheme as well as on the size of the control u^h. Since $\mathcal{Q}_m$ generates approximations of the exponential operator, it also approximately satisfies the group property. For even m we obtain an enhanced approximation.

Lemma 5. *Under the assumptions of Theorem 3 with* m *even we have*

$$\mathcal{Q}_m(B)(-z, u^h) \cdot \mathcal{Q}_m(B)(z, u^h) - I = o(z^{m+1}).$$

The next lemma states that the equivalence of taking the adjoint and reversing time is preserved by the GST scheme.

Lemma 6. *Let* $B \in \mathfrak{B}^{\mathbb{C}\times\mathcal{U}^h}$, $u^h \in \mathcal{U}^h$ *and* $m \in \mathbb{N}$. *If* B *satisfies the symmetry condition*

$$B(z, u^h)^* = B(-\bar{z}, u^h) \tag{20}$$

for all $z \in \mathbb{C}$, *then*

$$\mathcal{Q}_m(B)(z, u^h)^* = \mathcal{Q}_m(B)(-\bar{z}, u^h).$$

Proof. The result follows by induction over m using Definition 2. □ □

This result is important for the computation of the adjoint state. The adjoint action of the time stepping scheme can be computed by using a negative time step. For the linear approximation (19) the symmetry condition (20) for B is equivalent to the condition $A(u^h)^* = -A(u^h)$ for A. For our application we have $A(u^h) = -iH(u^h)$ for some self-adjoint $H(u^h)$, thus the condition is satisfied. For general approximations the condition on B is stronger than the skew-adjointness of A. In general $\mathcal{Q}_m(B)(t, u^h)$ with $t \in \mathbb{R}$ will not be unitary. However, under the assumptions of Lemma 5 and 6 we obtain

$$\mathcal{Q}_m(B)(t, u^h)^* \cdot \mathcal{Q}_m(B)(t, u^h) - I = o(t^{m+1}) \ .$$

In the context of optimal control we need to differentiate the time stepping scheme with respect to the control. A naive differentiation of the product in (18) leads to the inefficient expression

$$\mathcal{Q}_m'(B)(z,u^h)(\delta u^h) = \sum_{k=1}^{2^{m-1}} \left(\prod_{l=1}^{k-1} B(\alpha_l z, u^h)\right) B'(\alpha_k z, u^h)(\delta u^h) \left(\prod_{l=k+1}^{2^{m-1}} B(\alpha_l z, u^h)\right)$$

with complexity $\mathcal{O}(2^{2m-2})$. Exploiting the multiplicative structure of the algorithm, we will follow a different approach that will give the same complexity $\mathcal{O}(2^{m-1})$ as the original product. The approach can be understood as a modification of the result from [13]. For an analytic function F and square matrices B and δB we have

$$F\begin{pmatrix} B & 0 \\ \delta B & B \end{pmatrix} = \begin{pmatrix} F(B) & 0 \\ F'(B)(\delta B) & F(B) \end{pmatrix}.$$

That is, the directional derivative of F can be obtained by applying the function to a matrix with special structure. For general directional derivatives with respect to a parameter the matrix gets more complicated. We introduce the notation $D^s_{\delta u^h}(B)(z,u^h) = B^{(s)}(z,u^h)(\delta u^h)^s$ for the directional derivative, the space

$$C^r_{\delta u^h} = \{\, B \in \mathfrak{B}^{\mathbb{C}\times\mathcal{U}^h} : \\ z \mapsto D^s_{\delta u^h}(B)(z,u^h) \text{ analytic for each } 0 \le s \le r, u^h \in \mathcal{U}^h \,\}$$

of differentiable functions, and the nilpotent $(r+1)\times(r+1)$-matrix N, given by $N_{s,s+1} = 1$ and zero otherwise.

Definition 7. The operator $\mathcal{D}^{(r)}_{\delta u}\colon C^r_{\delta u^h} \to (\mathfrak{B}^{\otimes r+1})^{\mathbb{C}\times\mathcal{U}^h}$ is defined by

$$\mathcal{D}^{(r)}_{\delta u^h}(B) = \sum_{s=0}^{r} \frac{1}{s!} N^s \otimes D^s_{\delta u} B.$$

This operator generates a matrix of derivatives in the direction δu up to order r. It can also formally be understood as $\mathcal{D}^{(r)}_{\delta u^h}(B) = \exp(N \otimes D_{\delta u})(I \otimes B)$. More explicitely, we have

$$\mathcal{D}^{(r)}_{\delta u^h}(B)(z,u^h) = \begin{pmatrix} B(z,u^h) & 0 & \cdots & \cdots & \cdots & 0 \\ \frac{1}{1!}D^1_{\delta u}B(z,u^h) & B(z,u^h) & 0 & \cdots & \cdots & 0 \\ \frac{1}{2!}D^2_{\delta u}B(z,u^h) & \frac{1}{1!}D^1_{\delta u}B(z,u^h)(\delta u) & B(z,u^h) & 0 & \dots & 0 \\ \vdots & \vdots & & \ddots & \ddots & \vdots \\ \vdots & \vdots & & & \ddots & 0 \\ \frac{1}{r!}D^r_{\delta u}B(z,u^h) & \frac{1}{(r-1)!}D^{r-1}_{\delta u}B(z,u^h) & \cdots & \cdots & \cdots & B(z,u^h) \end{pmatrix}.$$

Alternatively, the operator $\mathcal{D}^{(r)}_{\delta u^h}$ could also be defined on differentiable functions in $\mathfrak{B}^{\mathcal{U}^h}$ first and then extended pointwise to $C^r_{\delta u^h}$ by $\mathcal{D}^{(r)}_{\delta u^h}(B)(z,u^h) = \mathcal{D}^{(r)}_{\delta u^h}(B(z,\cdot))(u^h)$. We obtain the following crucial property for $D^{(r)}_{\delta u^h}$.

Lemma 8. *For $A, B \in C^r_{\delta u^h}$,*

$$\mathcal{D}^{(r)}_{\delta u^h}(A \cdot B) = \mathcal{D}^{(r)}_{\delta u}(A) \cdot \mathcal{D}^{(r)}_{\delta u}(B).$$

Proof. Using the general Leibniz rule we obtain

$$\mathcal{D}^{(r)}_{\delta u^h}(A \cdot B) = \sum_{s=0}^{r} \sum_{k+l=s} N^s \otimes \frac{1}{k!l!} D^k_{\delta u^h} A \cdot D^l_{\delta u^h} B.$$

On the other hand

$$\mathcal{D}^{(r)}_{\delta u^h}(A) \cdot \mathcal{D}^{(r)}_{\delta u^h}(B) = \sum_{k=0}^{r} \sum_{l=0}^{r} N^{k+l} \otimes \frac{1}{k!l!} D^k_{\delta u^h} A \cdot D^l_{\delta u^h} B.$$

Since $N^{k+l} = 0$ for $k + l > r$ the two double sums on the right coincide. □ □

This result can be interpreted as multiplicativity of the differential operator $D^{(r)}_{\delta u^h}$, which is of course not true for the directional derivative itself. The multiplicativity implies the following commutativity relation of differentiation and the GST scheme.

Theorem 9. *Let $\delta u^h \in \mathcal{U}^h$ and $r, m \in \mathbb{N}$. Then*

$$\mathcal{Q}_m \circ \mathcal{D}^{(r)}_{\delta u^h} = \mathcal{D}^{(r)}_{\delta u^h} \circ \mathcal{Q}_m.$$

Proof. We proof the result by induction over m. For $m = 1$ we have $\mathcal{Q}_1(B) = B$ for all $B \in C^r_{\delta u^h}$ and the claim holds. Let the result be true for $m-1$. We will use the preceding lemma and the multiplicative structure of $\mathcal{Q}_m$. Let $z \in \mathbb{C}$, $u^h \in \mathcal{U}^h$ and $B \in C^r_{\delta u^h}$. Using the definition of $\mathcal{Q}_m$, Lemma 8, the induction hypothesis, and the definition of $\mathcal{Q}_m$ again we obtain

$$\begin{aligned}
\mathcal{D}^{(r)}_{\delta u^h}(\mathcal{Q}_m(B))(z,u^h) &= \mathcal{D}^{(r)}_{\delta u^h}\Big(\mathcal{Q}_{m-1}(B)(p_m z,\cdot) \cdot \mathcal{Q}_{m-1}(B)(\overline{p_m} z,\cdot)\Big)(u^h) \\
&= \mathcal{D}^{(r)}_{\delta u^h}\Big(\mathcal{Q}_{m-1}(B)(p_m z,\cdot)\Big)(u^h) \cdot \mathcal{D}^{(r)}_{\delta u^h}\Big(\mathcal{Q}_{m-1}(B)(\overline{p_m} z,\cdot)\Big)(u^h) \\
&= \mathcal{Q}_m(\mathcal{D}^{(r)}_{\delta u^h}(B))(p_m z, \delta u^h) \cdot \mathcal{Q}_m(\mathcal{D}^{(r)}_{\delta u^h}(B))(\overline{p_m} z, \delta u^h) \\
&= \mathcal{Q}_m(\mathcal{D}^{(r)}_{\delta u^h}(B))(z,u^h).
\end{aligned}$$

Since z, u^h and B were arbitrary we obtain $\mathcal{D}^{(r)}_{\delta u^h} \circ \mathcal{Q}_m = \mathcal{Q}_m \circ \mathcal{D}^{(r)}_{\delta u^h}$. □ □

The theorem says that we can obtain a matrix of derivatives of the higher order time stepping operators by applying the GST operator to a matrix of derivatives of the first order time stepping operator. In the case $r = 1$ we get

$$\mathcal{Q}_m \begin{pmatrix} B(u^h) & 0 \\ B'(u^h)(\delta u^h) & B(u^h) \end{pmatrix} = \begin{pmatrix} \mathcal{Q}_m(B)(u^h) & 0 \\ \mathcal{Q}_m(B)'(u^h)(\delta u^h) & \mathcal{Q}_m(B)(u^h) \end{pmatrix}.$$

Second derivatives can be obtained in two ways. For derivatives in the same direction one can use Theorem 9 for $r = 2$. This yields

$$\mathcal{Q}_m \begin{pmatrix} B(u^h) & 0 & 0 \\ B'(u^h)(\delta u^h) & B(u^h) & 0 \\ \frac{1}{2}B''(u^h)(\delta u^h, \delta u^h) & B'(u^h)(\delta u^h) & B(u^h) \end{pmatrix}$$
$$= \begin{pmatrix} \mathcal{Q}_m(B)(u^h) & 0 & 0 \\ \mathcal{Q}_m(B)'(u^h)(\delta u^h) & \mathcal{Q}_m(B)(u^h) & 0 \\ \frac{1}{2}\mathcal{Q}_m(B)''(u^h)(\delta u^h, \delta u^h) & \mathcal{Q}_m(B)'(u^h)(\delta u^h) & \mathcal{Q}_m(B)(u^h) \end{pmatrix}.$$

For second derivatives in different directions one can use

$$\mathcal{Q}_m \circ \mathcal{D}^{(r)}_{\delta u^h} \circ \mathcal{D}^{(r)}_{\tau u^h} = \mathcal{D}^{(r)}_{\delta u^h} \circ \mathcal{D}^{(r)}_{\tau u^h} \circ \mathcal{Q}_m \tag{21}$$

which follows immediately from Theorem 9. We obtain

$$\mathcal{Q}_m \begin{pmatrix} B(u^h) & 0 & 0 & 0 \\ B'(u^h)(\delta u^h) & B(u^h) & 0 & 0 \\ B'(u^h)(\tau u^h) & 0 & B(u^h) & 0 \\ B''(u^h)(\delta u^h, \tau u^h) & B'(u^h)(\tau u^h) & B'(u^h)(\delta u^h) & B(u^h) \end{pmatrix}$$
$$= \begin{pmatrix} \mathcal{Q}_m(B)(u^h) & 0 & 0 & 0 \\ \mathcal{Q}_m(B)'(u^h)(\delta u^h) & \mathcal{Q}_m(B)(u^h) & 0 & 0 \\ \mathcal{Q}_m(B)'(u^h)(\tau u^h) & 0 & \mathcal{Q}_m(B)(u^h) & 0 \\ \mathcal{Q}_m(B)''(u^h)(\delta u^h, \tau u^h) & \mathcal{Q}_m(B)'(u^h)(\tau u^h) & \mathcal{Q}_m(B)'(u^h)(\delta u^h) & \mathcal{Q}_m(B)(u^h) \end{pmatrix}.$$

Using this approach we do not compute the derivatives themselves, but always actions of a lower triangular operator matrix as in

$$\begin{pmatrix} \mathcal{Q}_m(B)(u^h) & 0 \\ \mathcal{Q}_m(B)'(u^h)(\delta u^h) & \mathcal{Q}_m(B)(u_h) \end{pmatrix} \begin{pmatrix} x^h \\ y^h \end{pmatrix}.$$

Often the whole expression is needed, as we will see in Lemma 12. The derivative $\mathcal{Q}_m(B)'(u^h)(\delta u^h)x^h$ together with $\mathcal{Q}_m(B)(u^h)y^h$ can be obtained by computing the right-hand side of

$$\begin{pmatrix} \mathcal{Q}_m(B)(u^h)x^h \\ \mathcal{Q}_m(B)'(u^h)(\delta u^h)y^h \end{pmatrix} = \begin{pmatrix} I & 0 \\ -I & I \end{pmatrix} \begin{pmatrix} \mathcal{Q}_m(B)(u^h) & 0 \\ \mathcal{Q}_m(B)'(u^h)(\delta u^h) & \mathcal{Q}_m(B)(u_h) \end{pmatrix} \begin{pmatrix} x^h \\ y^h \end{pmatrix},$$

see Algorithm 5 and 6 for specific implementations.

3.2 Discrete Optimization Problem

When solving otimal control problems one has to distinguish two approaches. One possibility is a first-optimize-then-discretize (OTD) approach where we discretize the continuous functional as well as the continuous equations for the derivatives. Another possibility is a first-discretize-then-optimize (DTO) approach where we first write down a discrete version of the cost functional and then the correct discrete derivatives for this function. In the subsequent discussion we will carry out a DTO analysis for optimal control problem (4) with the GST method. In order to obtain a discrete optimization problem we need to approximate the functional j by a discrete version j^h. The main ingredient will be the approximation of the state ψ by a discrete state ψ^h. To tackle the time-dependent problem with unbounded operators we proceed in three steps.

First we approximate the state equation (1) by a semi-discrete equation. To this end we approximate the Hilbert space $\mathcal{H}$ by a discrete space $\mathcal{H}^h$, and the control space $\mathcal{U}$ by a discrete control space $\mathcal{U}^h$. We define discrete versions H_0^h and H_1^h of the operators H_0 and H_1. We also approximate the initial state ψ_0 by $\psi_0^h \in \mathcal{H}^h$. The continuous evolution group $S(t,s;u)$ is approximated by the semi-discrete evolution group $S^h(t,s;u^h)$. Examples for this step include approximation with finite differences or finite elements, as well as spectral representations.

As a second step we approximate the action of the evolution group S^h by a product of solutions for time-independend problems. That is we introduce a time grid $(t_n)_{n=0}^N$ and approximate $S^h(t+t_{n-1},t_{n-1};u^h)$ by $\exp(tA_n(u^h))$. Here A_n is obtained by a Magnus expansion for the time-dependent Hamiltonian.

The third step consists of approximating the exponential with the GST method. That is $\exp(tA_n(u^h)) \approx \mathcal{Q}_m(B_n)(t,u^h)$ for some first order approximation B_n of $\exp(tA_n(u^h))$. This step is justified in Section 3.1. The discrete state ψ^h is then given by

$$\psi_n^h(u^h) = \prod_{p=1}^{n} Q_p(u^h)\psi_0^h, \quad n = 1,\dots,N, \tag{22}$$

where

$$Q_p(u^h) = \mathcal{Q}_m(B_p)(t_p - t_{p-1}, u^h)$$

is the discrete time stepping operator. Here and in the following we use the convention $\prod_{i=j}^{k} Q_i = Q_k \cdots Q_j$.

In addition to the discrete state we also need a discrete version $\mathcal{O}^h$ of the observable $\mathcal{O}$. The discrete version of (4) then reads

$$\text{Minimize} \quad j^h(u^h) = \frac{1}{2}\langle \psi_N^h(u^h), \mathcal{O}^h\psi_N^h(u^h)\rangle_{\mathcal{H}^h} + \frac{\alpha}{2}\|u^h\|_{\mathcal{U}^h}^2, \tag{23}$$

where ψ_N^h is given by (22).

Following the OTD approach we discretize equations (1), (5), (6), and (7) to compute approximations of the derivatives of j according to the continuous equations (8) and (9).

For the adjoint equation (5) we obtain

$$\varphi_j^h(u^h) = \left(\prod_{i=j+1}^{N} Q_i(u^h) \right)^* \mathcal{O}^h \psi_N^h(u^h), \quad j = 1, \dots, N, \tag{24}$$

but it is not obvious at all how the inhomogeneous equations (6) and (7) can be discretized. Another problem with a naive OTD approach is that the gradient will not be consistent and in addition the approximated Hessian will in general be not symmetric.

Following the DTO approach we have to derive expressions for the derivatives of j^h. This typically requires more work in setting up the equations, but it is constructive approach in the sense that we know how to correctly discretize (6) and (7). We will see that the correct discretization of the scalar products in (8) lead to additional computations. In Section 4.2 we will numerically compare the results of the two approaches.

We now analyze the derivatives of j^h. To shorten the notation we will sometimes drop the explicit dependence on the control. In paricular we will write ψ_j^h, φ_j^h, $\psi_j^{h\prime}$ and $\varphi_j^{h\prime}$ for $\psi_j^h(u^h)$, $\varphi_j^h(u^h)$, $\psi_j^{h\prime}(u^h)(\delta u^h)$ and $\varphi_j^{h\prime}(u^h)(\delta u^h)$, respectively. The DTO version of Proposition 1 then reads:

Proposition 10. *Let $u^h, \delta u^h, \tau u^h \in \mathcal{U}^h$. The first and second derivative of j^h are given by*

$$j^{h\prime}(u^h)(\delta u^h) = \sum_{n=1}^{N} \langle Q_n'(u^h)(\delta u^h)^* \varphi_n^h, \psi_{n-1}^h \rangle_{\mathcal{H}^h} + \alpha \langle u^h, \delta u^h \rangle_{\mathcal{U}^h} \tag{25}$$

and

$$\begin{aligned} j^{h\prime\prime}(u^h)(\delta u^h, \tau u^h) = \sum_{n=1}^{N} & \langle Q_n'(u^h)(\tau u^h)^* \varphi_n^{h\prime}, \psi_{n-1}^h \rangle_{\mathcal{H}^h} \\ & + \langle Q_n'(u^h)(\tau u^h)^* \varphi_n^h, \psi_{n-1}^{h\prime} \rangle_{\mathcal{H}^h} \\ & + \langle Q_n''(u^h)(\delta u^h, \tau u^h)^* \varphi_n^h, \psi_{n-1}^h \rangle_{\mathcal{H}^h} \\ & + \alpha \langle \delta u^h, \tau u^h \rangle_{\mathcal{U}^h} \end{aligned} \tag{26}$$

Proof. By the product rule applied to (22) we obtain

$$\psi_N^{h\prime}(u^h)(\delta u^h) = \sum_{n=1}^{N} \prod_{p=n+1}^{N} Q_p(u^h) Q_n'(u^h)(\delta u^h) \prod_{p=1}^{n-1} Q_p(u^h) \psi_0^h.$$

Therefore, using (22) and (24),

$$\begin{aligned} j'(u^h)(\delta u^h) &= \langle \mathcal{O}^h \psi_N^h(u^h), \psi_N^{h\prime}(u^h)(\delta u^h) \rangle_{\mathcal{H}^h} + \alpha \langle u^h, \delta u^h \rangle_{\mathcal{U}^h} \\ &= \sum_{n=1}^{N} \langle Q_n'(u^h)(\delta u^h)^* \varphi_n^h(u^h), \psi_{n-1}^h(u^h) \rangle_{\mathcal{H}^h} + \alpha \langle u^h, \delta u^h \rangle_{\mathcal{U}^h}. \end{aligned}$$

Differentiating this expression again and using the symmetry of the derivative in δu^h and τu^h concludes the proof. □ □

Let us now compare the results of the preceding proposition to the approximation of derivatives in an OTD approach. For this bilinear problem the important difference between DTO and OTD is the discretization of the product $\langle(-iH_1)^*\varphi, \psi\rangle$. Naive discretization of (8) results in an expression of the form

$$\sum_{n=1}^{N}(t_n - t_{n-1})\langle(-iH_1)^*\varphi_n, \psi_n\rangle_{\mathcal{H}^h} + \langle u, \delta u\rangle_{\mathcal{U}^h}.$$

This means that $Q_n'^*$ in the DTO approach corresponds to $(t_n - t_{n-1})(-iH_1)^*Q_n^*$ in the OTD approach. For the Hessian we obtain an additional term Q''^* that has no counterpart in the continuous case.

For a fixed basis (h_k) of $\mathcal{U}^h$ we can derive expressions for the coordinates of the gradient and Hessian-vector product. The coordinates Z of an element $u^h \in \mathcal{U}^h$ are determined by the equation $u^h = \sum_k Z_k h_k$. Let the mass matrix M of $\mathcal{U}^h$ with respect to the basis (h_k) be given by $M_{kl} = \langle h_k, h_l\rangle_{\mathcal{U}^h}$. Then the coordinates of the gradient and Hessian-vector product can be obtained from the directional derivatives by solving a linear system of equations.

Corollary 11. *Let $u^h, \delta u^h \in \mathcal{U}^h$, and let Z be the coordinates of $\nabla j^h(u^h) - \alpha u^h$. Then Z solves*

$$MZ = X \tag{27}$$

with

$$X_k = \sum_{n=1}^{N}\langle Q_n'(u^h)(h_k)^*\varphi_n^h, \psi_{n-1}^h\rangle_{\mathcal{H}^h}.$$

Let Z be the coordinates of $\nabla^2 j^h(u^h)(\delta u^h) - \alpha\delta u^h$. Then Z solves

$$MZ = Y \tag{28}$$

with

$$\begin{aligned} Y_k = \sum_{n=1}^{N} \quad & \langle Q_n'(u^h)(h_k)^*\varphi_n^{h\prime}, \psi_{n-1}^h\rangle_{\mathcal{H}^h} \\ & + \langle Q_n'(u^h)(h_k)^*\varphi_n^h, \psi_{n-1}^{h\prime}\rangle_{\mathcal{H}^h} \\ & + \langle Q_n''(u^h)(\delta u^h, h_k)^*\varphi_n^h, \psi_{n-1}^h\rangle_{\mathcal{H}^h}. \end{aligned}$$

Proof. Test (25) with $\delta u^h = h_k$ and (26) with $\tau u^h = h_k$. □ □

For different spaces $\mathcal{U}^h$ we have different mass matrices M. This effects the gradient and Hessian-vector product similar to the effect of $\mathcal{U}$ in Section 2.3. For M, being the finite element mass matrix, equations (27) and (28) result in a discrete version of (10) and (13). Having M equal to the finite element masss matrix in the H_0^1 scalar product, i.e. the stiffness matrix for the Poisson equation, leads to discrete versions of (12) and (15).

The main cost in computing gradients or Hessian actions is in assembling X and Y. This splits into the computation of discrete solutions of PDEs and the action of

the operators Q_j' and Q_j''. For the computation of the derivatives of Q_n we can use Theorem 9. The following lemma addresses the iterative computation of the states and their derivatives.

Lemma 12. *Let $u^h, \delta u^h \in \mathcal{U}^h$. Then ψ_n^h, $\psi_n^{h\prime}$, φ_n^h, $\varphi_n^{h\prime}$ satisfy the forward and backward recursions*

$$\begin{pmatrix} \psi_0^h \\ \psi_0^{h\prime} \end{pmatrix} = \begin{pmatrix} \psi_0^h \\ 0 \end{pmatrix}, \quad \begin{pmatrix} \psi_n^h \\ \psi_n^{h\prime} \end{pmatrix} = \begin{pmatrix} Q_n(u^h) & 0 \\ Q_n'(u^h)(\delta u^h) & Q_n(u^h) \end{pmatrix} \begin{pmatrix} \psi_{n-1}^h \\ \psi_{n-1}^{h\prime} \end{pmatrix}$$

and, respectively,

$$\begin{pmatrix} \varphi_N^h \\ \varphi_N^{h\prime} \end{pmatrix} = \begin{pmatrix} \mathcal{O}^h \psi_N^h \\ \mathcal{O}^h \psi_N^{h\prime} \end{pmatrix}, \quad \begin{pmatrix} \varphi_n^h \\ \varphi_n^{h\prime} \end{pmatrix} = \begin{pmatrix} Q_{n+1}(u^h)^* & 0 \\ Q_{n+1}'(u^h)(\delta u^h)^* & Q_{n+1}(u^h)^* \end{pmatrix} \begin{pmatrix} \varphi_{n+1}^h \\ \varphi_{n+1}^{h\prime} \end{pmatrix}.$$

Proof. The definitions of $\psi_n^h(u^h)$ and $\varphi_n^h(u^h)$ imply

$$\psi_n^h(u^h) = Q_n(u^h)\psi_{n-1}^h(u^h)$$

and

$$\varphi_n^h(u^h) = Q_{n+1}(u^h)^* \varphi_{n+1}^h(u^h).$$

Differentiation of the equations in the direction δu^h yields

$$\psi_n^{h\prime}(u^h)(\delta u^h) = Q_n'(u^h)(\delta u^h)\psi_n^h(u^h) + Q_n(u^h)\psi_{n-1}^{h\prime}(u^h)(\delta u^h)$$

and

$$\varphi_n^{h\prime}(u^h)(\delta u^h) = Q_{n+1}'(u^h)(\delta u^h)^* \varphi_{n+1}^h(u^h) + Q_{n+1}(u^h)^* \varphi_{n+1}^{h\prime}(u^h)(\delta u^h).$$

Additionally we have $\psi_0^{h\prime}(u^h) = 0$, $\varphi_N^h(u^h) = \mathcal{O}^h \psi_N^h(u^h)$ and $\varphi_N^{h\prime}(u^h)(\delta u^h) = \mathcal{O}^h \psi_N^{h\prime}(u^h)(\delta u^h)$. Writing those equations in matrix and vector form gives the first two results. □ □

To reduce the memory requirements [6, 17] we use a trick already used in the beginning of quantum control. We will not save the whole history of quantum states $(\psi_n^h)_n$ and $(\varphi_n^h)_n$ and their derivatives computed in a forward and backward iteration, and thereafter set up the vectors X and Y to compute the gradient and Hessian actions. Instead, we will save only the current iterates and assemble X and Y on the fly during the backwards iteration. This method introduces an additional backwards solve of the forward equation for the state and its derivative and an additional error due to the non-unitarity of the time stepping scheme. The following lemma tells us that we can indeed compute solutions of the (inhomogeneous) forward equations for ψ^h (and $\psi^{h\prime}$) backwards in time with a reasonable error.

Lemma 13. *Let $u^h, \delta u^h \in \mathcal{U}^h$, let B_n satisfy the symmetry condition* (20) *and the additional approximation property*

$$D_{\delta u^h}^1 \big(B_n(z, u^h) - \exp(z A_n(u^h)) \big) = o(z) \ .$$

Then ψ_n^h and $\psi_n^{h\prime}$ satisfy the backward recursion

$$\begin{pmatrix} \psi_n^h \\ \psi_n^{h\prime} \end{pmatrix} = \begin{pmatrix} Q_{n+1}(u^h)^* & 0 \\ Q_{n+1}'(u^h)(\delta u^h)^* & Q_{n+1}(u^h)^* \end{pmatrix} \begin{pmatrix} \psi_{n+1}^h \\ \psi_{n+1}^{h\prime} \end{pmatrix} + o(\Delta t_{n+1}^{m+1}),$$

where $\Delta t_n = t_n - t_{n-1}$.

Proof. We will use the result for the forward recursion from Lemma 12 and combine it with the approximate group property in Lemma 5 and the relation between taking adjoints and inverting time from Lemma 6. We have

$$\begin{pmatrix} \psi^h_{n+1} \\ \psi^{h\prime}_{n+1} \end{pmatrix} = \begin{pmatrix} Q_{n+1} & 0 \\ Q'_{n+1} & Q_{n+1} \end{pmatrix} \begin{pmatrix} \psi^h_n \\ \psi^{h\prime}_n \end{pmatrix}.$$

Therefore we get

$$\begin{pmatrix} Q^*_{n+1} & 0 \\ Q'^*_{n+1} & Q^*_{n+1} \end{pmatrix} \begin{pmatrix} \psi^h_{n+1} \\ \psi^{h\prime}_{n+1} \end{pmatrix} = \begin{pmatrix} Q^*_{n+1} & 0 \\ Q'^*_{n+1} & Q^*_{n+1} \end{pmatrix} \begin{pmatrix} Q_{n+1} & 0 \\ Q'_{n+1} & Q_{n+1} \end{pmatrix} \begin{pmatrix} \psi^h_n \\ \psi^{h\prime}_n \end{pmatrix}. \tag{29}$$

Under the additional assumption on A_{n+1} and B_{n+1} we have

$$\begin{pmatrix} B_{n+1}(\Delta t_{n+1}, u^h) & 0 \\ B'_{n+1}(\Delta t_{n+1}, u^h)(\delta u^h) & B_{n+1}(\Delta t_{n+1}, u^h) \end{pmatrix} \\ = I + \Delta t_{n+1} \begin{pmatrix} A_{n+1}(u^h) & 0 \\ A'_{n+1}(u^h)(\delta u^h) & A_{n+1}(u^h) \end{pmatrix} + o(\Delta t_{n+1}).$$

This implies

$$\begin{pmatrix} B_{n+1}(\Delta t_{n+1}, u^h) & 0 \\ B'_{n+1}(\Delta t_{n+1}, u^h)(\delta u^h) & B_{n+1}(\Delta t_{n+1}, u^h) \end{pmatrix} - \\ \exp \Delta t_{n+1} \begin{pmatrix} A_{n+1}(u^h) & 0 \\ A'_{n+1}(u^h)(\delta u^h) & A_{n+1}(u^h) \end{pmatrix} = o(\Delta t_{n+1}).$$

Thus, we can use Lemma 6 and 5 to obtain

$$\begin{aligned} &\begin{pmatrix} Q^*_{n+1} & 0 \\ Q'^*_{n+1} & Q^*_{n+1} \end{pmatrix} \begin{pmatrix} Q_{n+1} & 0 \\ Q'_{n+1} & Q_{n+1} \end{pmatrix} \\ &\quad = \left[\mathcal{Q}_m \begin{pmatrix} B_{n+1} & 0 \\ B'_{n+1} & B_{n+1} \end{pmatrix} (-\Delta t_{n+1}, u^h) \right] \left[\mathcal{Q}_m \begin{pmatrix} B_{n+1} & 0 \\ B'_{n+1} & B_{n+1} \end{pmatrix} (\Delta t_{n+1}, u^h) \right] \\ &\quad = I + o(\Delta t^{m+1}_{n+1}) \end{aligned}$$

for even m. Plugging this into (29) concludes the proof. □ □

The additional approximation property in the preceding lemma is satisfied for the linear approximation since there we have

$$D^1_{\delta u^h}\big(B_n(z, u^h) - \exp(zA_n(u^h))\big) = zA'_n(u^h)(\delta u^h) - \exp(zA_n(u^h))zA'_n(u^h)(\delta u^h) = o(z)\ .$$

In fact it is satisfied for all polynomial approximations.

The scheme for computations of gradients and Hessian actions are given by Algorithm 1 and 2. The subroutines for the assembly of the right-hand side are in their most general form given by Algorithm 3 and 4. In every step they modify the vectors X and Y, respectively, and evolve the states.

Data: u^h
Result: v^h
$\psi^h \leftarrow \psi_0^h$;
$X \leftarrow 0$;
for $n = 1, \dots, N$ **do**
  $\psi^h \leftarrow Q_n(u^h)\psi^h$;
end
$\varphi^h \leftarrow \mathcal{O}^h\psi^h$;
for $n = N, \dots, 1$ **do**
  ψ^h, φ^h, $X \leftarrow$ assemble_rhs_gradient(u^h, n; ψ^h, φ^h, X);
end
solve for Z: $MZ = X$;
$v^h \leftarrow \sum_{k=1}^K Z_k h_k + \alpha u^h$;

Algorithm 1: Compute the gradient

Data: u^h, δu
Result: w^h
$\begin{pmatrix} \psi^h \\ \psi^{h\prime} \end{pmatrix} \leftarrow \begin{pmatrix} \psi_0^h \\ 0 \end{pmatrix}$;
$Y \leftarrow 0$;
for $n = 1, \dots, N$ **do**
  $\begin{pmatrix} \psi^h \\ \psi^{h\prime} \end{pmatrix} \leftarrow \begin{pmatrix} Q_n(u^h) & 0 \\ Q_n'(u^h)(\delta u^h) & Q_n(u_h) \end{pmatrix} \begin{pmatrix} \psi^h \\ \psi^{h\prime} \end{pmatrix}$;
end
$\begin{pmatrix} \varphi^h \\ \varphi^{h\prime} \end{pmatrix} \leftarrow \begin{pmatrix} \mathcal{O}^h\psi^h \\ \mathcal{O}^h\psi^{h\prime} \end{pmatrix}$;
for $n = N, \dots, 1$ **do**
  ψ^h, $\psi^{h\prime}$, φ^h, $\varphi^{h\prime}$, $Y \leftarrow$ assemble_rhs_hessian(u^h, n; ψ^h, $\psi^{h\prime}$, φ^h, $\varphi^{h\prime}$, Y);
end
solve for Z: $MZ = Y$;
$w^h \leftarrow \sum_{k=1}^K Z_k h_k + \alpha \delta u^h$;

Algorithm 2: Apply the Hessian to a vector δu^h

Data: u^h, n, ψ^h, φ^h, X
Result: ψ^h, φ^h, X
$\psi^h \leftarrow Q_n(u^h)^*\psi^h$;
for $k = 1, \dots, K$ **do**
  $X_k \leftarrow X_k + \langle Q_n'^*(u^h)(h_k)\varphi^h, \psi^h \rangle_{\mathcal{H}^h}$;
end
$\varphi^h \leftarrow Q_n(u^h)^*\varphi^h$;

Algorithm 3: assemble_rhs_gradient for DTO and general controls

Data: u^h, n, ψ^h, $\psi^{h\prime}$, φ^h, $\varphi^{h\prime}$, Y
Result: ψ^h, $\psi^{h\prime}$, φ^h, $\varphi^{h\prime}$, Y
$\begin{pmatrix}\psi^h\\ \psi^{h\prime}\end{pmatrix} \leftarrow \begin{pmatrix} Q_n(u^h)^* & 0 \\ Q_n'(u^h)(\delta u^h)^* & Q_n(u_h)^*\end{pmatrix}\begin{pmatrix}\psi^h\\ \psi^{h\prime}\end{pmatrix}$;
for $k = 1, \dots, K$ **do**
  $Y_k \leftarrow Y_k + \langle Q_n'(u^h)(h_k)^*\varphi^{h\prime}, \psi^h\rangle_{\mathcal{H}^h} + \langle Q_n'(u^h)(h_k)^*\varphi^h, \psi^{h\prime}\rangle_{\mathcal{H}^h}$;
  $\qquad + \langle Q_n''(u^h)(\delta u^h, h_k)^*\varphi^h, \psi^h\rangle_{\mathcal{H}^h}$
end
$\begin{pmatrix}\varphi^h\\ \varphi^{h\prime}\end{pmatrix} \leftarrow \begin{pmatrix} Q_n(u^h)^* & 0 \\ Q_n'(u^h)(\delta u^h)^* & Q_n(u_h)^*\end{pmatrix}\begin{pmatrix}\varphi^h\\ \varphi^{h\prime}\end{pmatrix}$;

Algorithm 4: assemble_rhs_hessian for DTO and general controls

Different control discretizations give rise to different assemble_rhs_gradient and assemble_rhs_hessian routines. For piecewise linear controls and a magnus expansion of order two using the midpoint rule we obtain Algorithm 5 and 6. There the loop over k reduces to a common update of just two entries of X and Y, respectively. Notice how these algorithms compute the derivatives of the time stepping via Theorem 9 and reuse results from the action of the large operator matrices. The second derivatives for the Hessian action are computed using equation (21). In contrast to the DTO approach, for the OTD approach the updates of X and Y in Algorithm 7 and 8 are computed in the same way as the continuous derivatives in Section 2.

Data: u^h, n, ψ^h, φ^h, X
Result: ψ^h, φ^h, X
$\psi^h \leftarrow Q_n(u^h)^*\psi^h$;
$\chi^h \leftarrow \varphi^h$;
$\begin{pmatrix}\varphi^h\\ \chi^h\end{pmatrix} \leftarrow \begin{pmatrix} Q_n(u^h)^* & 0 \\ Q_n'(u^h)(\delta u^h)^* & Q_n(u_h)^*\end{pmatrix}\begin{pmatrix}\varphi^h\\ \chi^h\end{pmatrix}$;
$\chi^h \leftarrow \chi^h - \varphi^h$;
$\delta X \leftarrow \langle \chi^h, \psi^h\rangle_{\mathcal{H}^h}$;
$X_n \leftarrow X_n + \delta X$;
$X_{n-1} \leftarrow X_{n-1} + \delta X$;

Algorithm 5: assemble_rhs_gradient for DTO piecewise linear controls

Comparing the computational complexity of assemble_rhs_gradient we see that the DTO approach has roughly 1.5 times the cost of the OTD approach. We need an application of a 2×2 time stepping matrix instead of just an application of the ordinary time stepping operator. For assemble_rhs_hessian the cost increases roughly by a factor 2. Here we need an additional application of the 4×4 time stepping matrix.

Data: u^h, n, ψ^h, $\psi^{h\prime}$, φ^h, $\varphi^{h\prime}$, Y
Result: ψ^h, $\psi^{h\prime}$, φ^h, $\varphi^{h\prime}$, Y
$\begin{pmatrix}\psi^h\\ \psi^{h\prime}\end{pmatrix} \leftarrow \begin{pmatrix} Q_n(u^h)^* & 0 \\ Q_n'(u^h)(\delta u^h)^* & Q_n(u_h)^*\end{pmatrix}\begin{pmatrix}\psi^h\\ \psi^{h\prime}\end{pmatrix}$;
;
$\chi_2^h \leftarrow \varphi^{h\prime}$;
$\chi_3^h \leftarrow \varphi^{h\prime}$;
$\begin{pmatrix}\varphi^{h\prime}\\ \chi_2^h\end{pmatrix} \leftarrow \begin{pmatrix} Q_n(u^h)^* & 0 \\ Q_n'(u^h)(h_n)^* & Q_n(u_h)^*\end{pmatrix}\begin{pmatrix}\varphi^{h\prime}\\ \chi_2^h\end{pmatrix}$;
$\chi_2^h \leftarrow \chi_2^h - \varphi^{h\prime}$;
$\varphi^{h\prime} \leftarrow \chi_3^h$;
;
$\chi_1^h \leftarrow \varphi^h$;
$\begin{pmatrix}\varphi^h\\ \varphi^{h\prime}\\ \chi_1^h\\ \chi_3^h\end{pmatrix} \leftarrow \begin{pmatrix} Q_n(u^h)^* & 0 & 0 & 0 \\ Q_n'(u^h)(\delta u^h)^* & Q_n(u^h)^* & 0 & 0 \\ Q_n'(u^h)(h_n)^* & 0 & Q_n(u^h)^* & 0 \\ Q_n''(u^h)(\delta u^h, h_n)^* & Q_n(u^h)(h_n)^* & Q_n'(u^h)(\delta u^h)^* & Q_n(u^h)^*\end{pmatrix}\begin{pmatrix}\varphi^h\\ \varphi^{h\prime}\\ \chi_1^h\\ \chi_3^h\end{pmatrix}$;
$\chi_1^h \leftarrow \chi_1^h - \varphi^h$;
$\chi_3^h \leftarrow \chi_3^h - \varphi^{h\prime} - \chi_2^h$;
;
$\delta Y \leftarrow \langle \chi_1^h, \psi^{h\prime}\rangle_{\mathcal{H}^h} + \langle \chi_2^h, \psi^h\rangle_{\mathcal{H}^h} + \langle \chi_3^h, \psi^h\rangle_{\mathcal{H}^h}$;
$Y_n \leftarrow Y_n + \delta Y$;
$Y_{n-1} \leftarrow Y_{n-1} + \delta Y$;

Algorithm 6: assemble_rhs_hessian for DTO and piecewise linear controls

Data: u^h, n, ψ^h, φ^h, X
Result: ψ^h, φ^h, X
$\delta X \leftarrow \langle (-iH_1^h)^*\varphi^h, \psi^h\rangle_{\mathcal{H}^h}$;
$X_n \leftarrow X_n + \delta X$;
$\psi^h \leftarrow Q_n(u^h)^*\psi^h$;
$\varphi^h \leftarrow Q_n(u^h)^*\varphi^h$;

Algorithm 7: assemble_rhs_gradient for OTD and piecewise linear controls

Data: u^h, n, ψ^h, $\psi^{h\prime}$, φ^h, $\varphi^{h\prime}$, Y
Result: ψ^h, $\psi^{h\prime}$, φ^h, $\varphi^{h\prime}$, Y
$\delta Y \leftarrow \langle(-iH_1)^*\varphi^h, \psi^{h\prime}\rangle_{\mathcal{H}^h} + \langle(-iH_1)^*\varphi^{h\prime}, \psi^h\rangle_{\mathcal{H}^h}$;
$Y_n \leftarrow Y_n + \delta Y$;
$\begin{pmatrix}\psi^h \\ \psi^{h\prime}\end{pmatrix} \leftarrow \begin{pmatrix} Q_n(u^h)^* & 0 \\ Q_n'(u^h)(\delta u^h)^* & Q_n(u^h)^*\end{pmatrix}\begin{pmatrix}\psi^h \\ \psi^{h\prime}\end{pmatrix}$;
$\begin{pmatrix}\varphi^h \\ \varphi^{h\prime}\end{pmatrix} \leftarrow \begin{pmatrix} Q_n(u^h)^* & 0 \\ Q_n'(u^h)(\delta u^h)^* & Q_n(u^h)^*\end{pmatrix}\begin{pmatrix}\varphi^h \\ \varphi^{h\prime}\end{pmatrix}$;

Algorithm 8: assemble_rhs_hessian for OTD and piecewise linear controls

3.3 Optimization Method

The optimization problem is often solved with methods based on computing the gradient of the reduced cost functional j (OTD) or its discrete version j^h (DTO). Since our approach makes second derivatives of j^h easily accessible we are able to apply second order methods based on the Hessian like Newton's method. Since our problem is nonconvex we need a globalization strategy to make the Newton method globally convergent. To this end we choose a trust region globalization with a Steihaug conjugate gradient (CG) method to solve the trust region subproblems, see [14]. The main trust region loop is presented in Algorithm 9. The method solves a quadratic model problem in a neighborhood of size r and updates the neighborhood depending on the model fidelity ρ. The main computational cost in the optimization routine comes from the iterative solution of the Newton equation in the trust region subproblems described in Algorithm 10. The algorithm requires the action of the Hessian in each iteration step which results in the solution of several PDEs as described by Algorithm 4 and 6. Since we use a CG method to solve the Newton equation, symmetry of the Hessian is needed. This is in general not the case for the Hessian approximation resulting from the OTD approach. For the DTO approach where the state evolution is stored in memory the Hessian is symmetric. For the memory efficient approximation we use the Hessian is not exactly symmetric with an error controlled by Proposition 12. The only preconditioning we use for the CG method is the implicit preconditioning through the scalar product of $\mathcal{U}^h$.

To understand how the GST method performs in the context of optimal control we test it with different optimization methods. In particular we compare the performance of Newton's method with different gradient based methods. For this comparison we choose the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method, its memory efficient version L-BFGS, both with a strong Wolfe line search, and the Barzilai–Borwein (BB) method with a non-monotone line search. The BFGS method is expected to be the fastest of the three. Due to its large memory requirements it is not applicable to problems with a large number of control parameters. The BB method is supposedly less prone to end up in non-global minima due to its non-monotone line search, which might be useful for nonlinear problems.

Data: u_0^h
Result: $\bar{u}^h$
$u^h \leftarrow u_0^h$;
$\rho \leftarrow 1$;
$r \leftarrow 1$;
$j^h \leftarrow \text{functional}(u^h)$;
$v^h \leftarrow \text{gradient}(u^h)$;
while $\|v^h\| \geq TOL_\nabla$ **do**
  $\delta u^h \leftarrow \text{subproblem}(u^h, v^h, r)$;
  $j^h_{\text{new}} \leftarrow \text{functional}(u^h + \delta u^h)$;
  $w^h \leftarrow \text{hessian}(u^h, \delta u^h)$;
  $\rho \leftarrow (j^h - j^h_{\text{new}})/\langle \delta u^h, -v^h - 0.5 w^h \rangle_{\mathcal{U}^h}$;
  if $\rho < 0.25$ **then**
    $r \leftarrow 0.25r$;
  end
  if $\rho \geq 0$ **then**
    $u^h \leftarrow u^h + \delta u^h$;
    $j^h \leftarrow j^h_{\text{new}}$;
    $v^h \leftarrow \text{gradient}(u^h)$;
    if $\rho \geq 0.8$ **then**
      $r \leftarrow 2r$;
    end
  end
end

Algorithm 9: Trust Region Newton

Data: u^h, v^h, r
Result: δu^h
$p \leftarrow 0$;
$res \leftarrow -v^h$;
$d \leftarrow res$;
while $\|res\| > TOL_{\nabla^2}$ **do**
    $w^h \leftarrow \text{hessian}(u^h, d)$;
    $s \leftarrow \langle d, w^h \rangle_{\mathcal{U}^h}$;
    if $s \leq 0$ **then**
        find τ such that $\|p + \tau d\| = r$;
        $p \leftarrow p + \tau d$;
        break;
    end
    $\alpha \leftarrow \|res\|/s$;
    if $\|p + \alpha d\| > r$ **then**
        find $\tau \geq 0$ such that $\|p + \tau d\| = r$;
        $p \leftarrow p + \tau d$;
        break;
    end
    $p \leftarrow p + \alpha d$;
    $\beta \leftarrow \|res - \alpha w^h\|^2 / |res\|^2$;
    $res \leftarrow res - \alpha w^h$;
    $d \leftarrow res + \beta d$;
end
$\delta u^h \leftarrow p$;

Algorithm 10: Trust Region Newton Subproblem with Steihaug CG

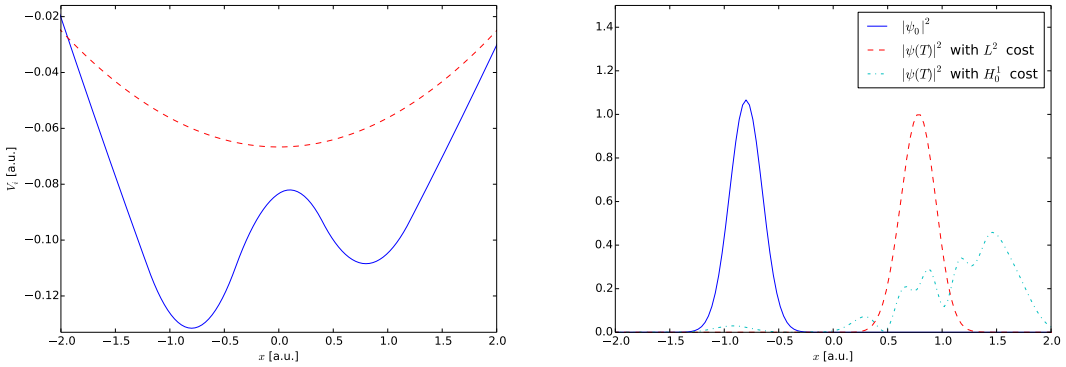

Figure 1: On the left: the two potential energy surfaces. On the right: the initial state (solid) and the final states for regularization with $\mathcal{U} = L^2$ (dashed) and $\mathcal{U} = H_0^1$ (dashdotted).

4 Numerical results

To test the GST method we choose a system of two one-dimensional Schrödinger equations. For this example we analyze the consistency of the derivatives generated by the GST method. We compare the performance of Newton's methods with different gradient based methods. We also study the qualitative behavior of solutions for L^2 and H_0^1 regularization for our model problem.

4.1 Problem Setting and Discretization

As a test example we consider the optimal control of a system of one-dimensional Schrödinger equations on two potential energy surfaces. We consider the model quantum system from Section 2 with $d = 1$ and $M = 2$. The toy problem resembles a 1D version of the problem studied in [10]. We assume the particle mass of a proton and give all quantities in atomic units. The potential energy surfaces are plotted in Figure 1. The control objective is to reach the potential well on the right starting from the potential well on the left. This is modeled by the observation operator $\mathcal{O}$, being the projection on the complement of functions with support on the lower energy surface to the right of the potential barrier. The initial state ψ_0 is given by a Gaussian located in the potential lower well, depicted in Figure 1. The energy differences between the two potential energy surfaces measured at the local minima of the lower surface are around 0.074 and 0.048. Considering Bohr's law we expect the control to contain the two frequencies $\omega_1 \approx 0.074$ and $\omega_2 \approx 0.048$. We chose a time horizon of $T = 3000$. The time horizon was chosen large enough to allow for sufficiently many oscillations with the Bohr frequencies ω_1 and ω_2, and for sufficient movement of the wave packet in space. The spatial domain of each potential energy surface is $[-4, 4]$. For the control space we considered the cases $\mathcal{U} = L^2(0, T)$ and $H_0^1(0, T)$.

We discretized the state space $\mathcal{H}$ by a nodal basis on a uniform grid with 256 points. The discrete drift Hamiltonian H_0^h is given by a simple finite difference stencil and a pointwise multiplication operator at the grid points. The coupling Hamiltonian H_1^h is also a pointwise multiplication operator at the grid points. For the time-discretization scheme we defined a uniform grid $(t_n)_{n=0}^N$ with $N = 2048$ points. We applied linear finite elements on this grid as discrete controls in $\mathcal{U}^h$.

For the first order approximation operator within the GST method we used the linear approximation

$$B_n(z, u^h) = I + zA_n(u^h)$$

where A_n is given by the midpoint rule,

$$A_n(u^h) = -i\left(H_0^h + \frac{u^h(t_{n-1}) + u^h(t_n)}{2} H_1^h\right).$$

Then B_n is an analytical first order approximation of the exponential function and satisfies the symmetry condition $B_n(z, u^h)^* = B_n(-\bar{z}, u^h)$ and the additional assumption $D^1_{\delta u^h}(B_n(z, u^h) - \exp(zA_n(u^h))) = o(z)$ from Lemma 12. Using the GST scheme we increased the order of the approximation of the exponential function to $m = 4$. The grid sizes in time and space have been chosen appropriately to satisfy the stability condition of Proposition 4. Here we found $s = 3 * 2^3 \Delta t \|A_n(u^h)\| \approx 0.3$ to be sufficient for the controls u^h appearing in the optimization method.

The derivatives of the discrete cost functional can then efficiently be computed by the GST method using the calculus in of Section 3. In particular we use Algorithm 1 and 2 to compute the gradient and Hessian, respectively. The subroutines for assembling the right-hand sides in the case of linear finite elements are then given by Algorithm 5 and 6.

As initial guess for the control in the optimization procedure we chose a function containing the frequencies ω_1 and ω_2 that only achieved about 3% of the control goal. The regularization parameter was chosen to be $\alpha = 0.0005$. The optimization method terminates if the norm of the gradient falls below $TOL_\nabla = 10^{-12}$ or the number of iteration exceeds $1.5 * 10^4$. The TRN and (L-)BFGS methods terminate when the functional does not decrease any more for a minimal step size.

4.2 Results

In this section we focus on three comparisons: First we compare the memory efficient discretize-then-optimize approach with its memory inefficient version and an optimize-then-discretize approach. Then we study the performance of different Hessian and gradient based methods where the derivatives are provided by the GST method. Third we compare the qualitative behavior of solutions for L^2 and H_0^1 regularization for our model problem.

First we compare the consistency of the derivatives generated by the memory efficient DTO approach with the derivatives generated by the memory inefficient DTO and the OTD approaches. That means we study how well the approximations of the derivatives match the actual derivatives of the cost functional j. To this end we compare the

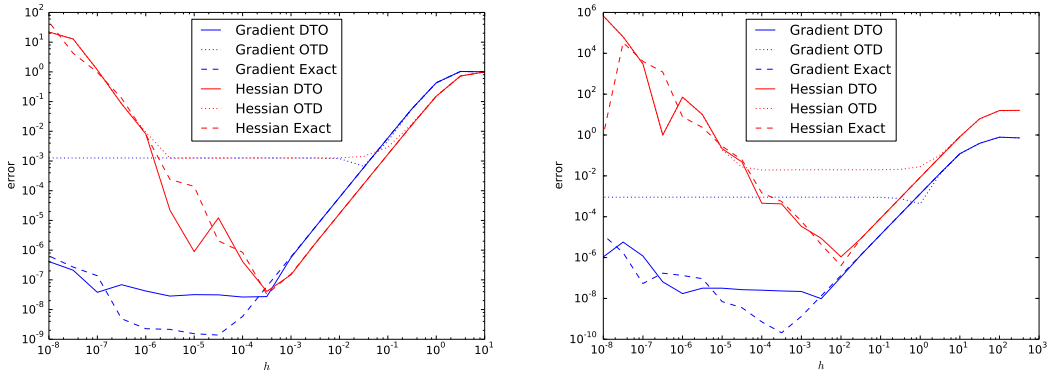

Figure 2: The difference between finite differences and the different approximation of the gradient using the adjoint calculus for $\mathcal{U} = L^2(0,T)$ (left) and $\mathcal{U} = H_0^1(0,T)$ (right).

approximations obtained through the calculus in Section 2 and 3 to finite difference approximations of the derivatives with varying step size h. In Figure 2 we plotted the absolute value of the difference divided by h against h. Comparing finite differences to a good approximation of the derivatives results in a V shape. The right-hand side of the V corresponds to the convergence of the finite differences with decreasing h and the left-hand side results from numerical cancellations for very small h. The gradient and Hessian computed by the exact but memory inefficient DTO approach show this behavior since they are the exact derivatives up to rounding errors. We see that the memory efficient DTO approach leads to an approximation comparable to the exact DTO approach and is much better than the OTD approach. A better approximation of the derivatives can lead to more robustness of the optimization method. With the OTD gradient we often got stuck at suboptimal controls, because the computed gradient direction did not lead to a decrease of the functional. Using an OTD or the memory efficient DTO approach the Hessian will in general be non-symmetric. The conjugate gradient (CG) method we chose to solve the Newton equations relies on the symmetry of the Hessian. We observed that using the non-symmetric matrix from the OTD approach leads to much larger number of CG iterations for the solution of the Newton equation. Although we observed a slowdown by a factor two for the memory efficient approach compared the the memory inefficient DTO approach, the problem was much less pronounced compared to the OTD approach. This justifies using the memory efficient DTO implementation in the context of the GST method.

We will now compare the performance first and second order optimization methods, with derivatives provided by the GST method. In Figure 3 we plotted the norm of the gradient against the iteration step and against the computation time for different optimization methods and the two cases of L^2 and H_0^1 costs. The norm of the gradient is used as a measure of optimality, faster decrease means faster convergence to the optimum. For all methods the gradient norms oscillate which makes it difficult to analyze

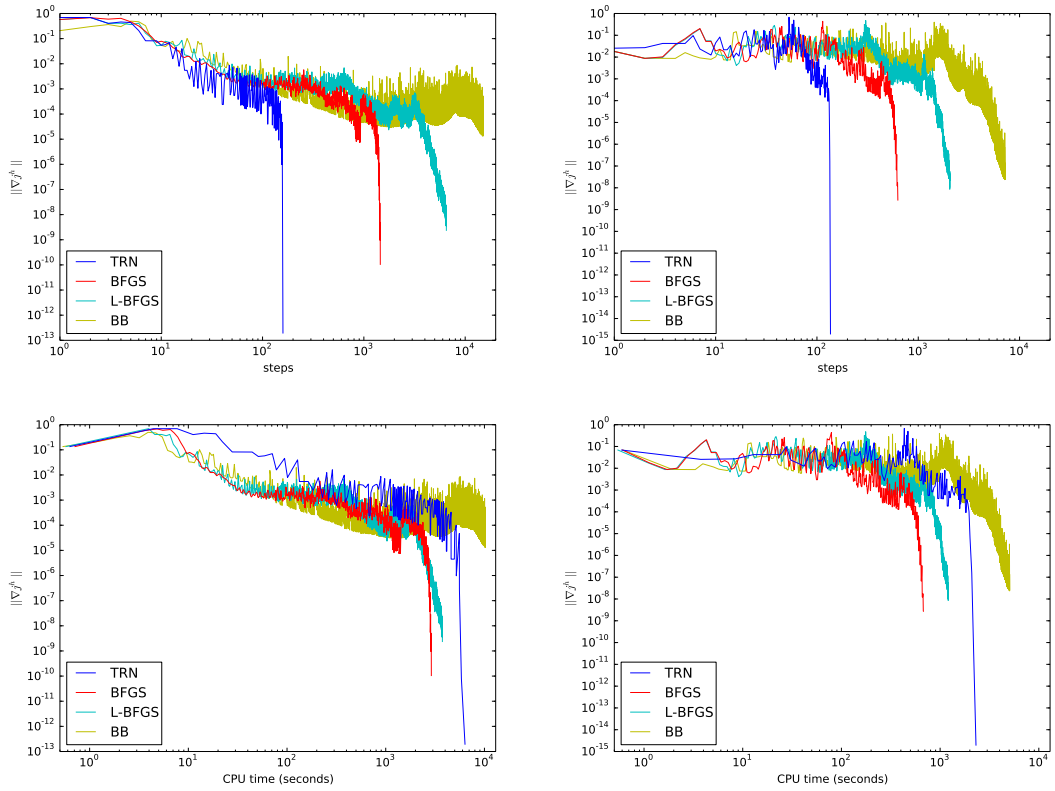

Figure 3: Norm of the Gradient $\|\nabla j^h\|$ against steps (top) and CPU time (bottom) for $\mathcal{U} = L^2(0,T)$ (left) and $\mathcal{U} = H_0^1(0,T)$ (right).

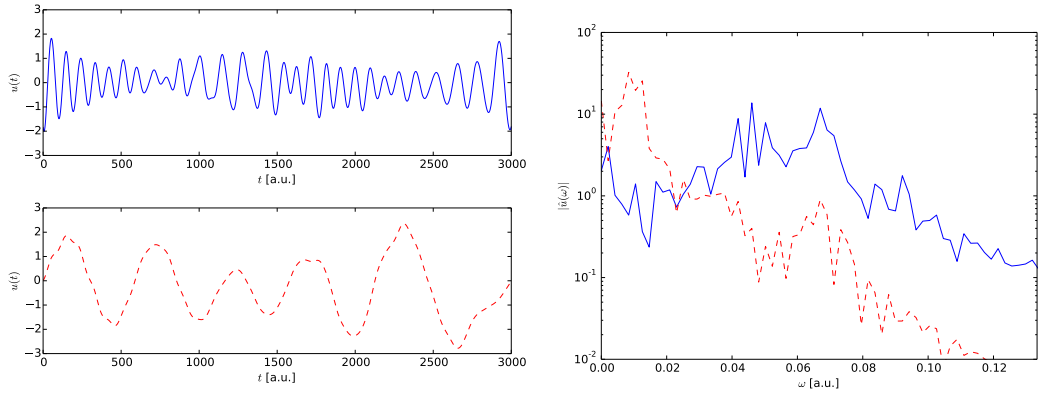

Figure 4: The optimal control fields (left) and their Fourier coefficients (right) for $\mathcal{U} = L^2(0,T)$ (solid) and $\mathcal{U} = H_0^1(0,T)$ (dashed).

the results. We observed that TRN needs an order of magnitude less steps before it terminates compared to the second fastest method BFGS. For large scale problems the comparison to the limited memory version L-BFGS is, however, more appropriate. The L-BFGS method needs about 50 times more steps until termination. The BB method was by far the slowest method. The final gradient norm for TRN was much smaller in our experiments compared to the other methods. The BFGS and L-BFGS method terminated due to lack of decrease in the functional, whereas the BB method reached the maximum number of iterations. Comparing the decrease in the gradient norm with the computation time, the situation changes in favor of the first order methods. This might be due to the fact that it takes many steps in the TRN method to reach the region of fast local convergence. Solving the Newton equations without preconditioning and additionally with a matrix that is not perfectly symmetric might also contribute to the slowdown. Using a hybrid method to combine fast local convergence of Newton's method with the nice global behavior of L-BFGS might be of advantage. The L^2 and the H_0^1 cases behave very similarly with respect to the speed of convergence where the H_0^1 proved to be a bit faster.

In Figure 4 we show solutions of the optimal control problem for the control spaces $\mathcal{U} = L^2(0,T)$ and $\mathcal{U} = H_0^1(0,T)$. The structure of the controls is quite different. Whereas the control for the L^2 case contains major frequency contributions around ω_1 and ω_2, the control for the H_0^1 case contains one main frequency which is much lower. In Figure 1 one can see the two different densities of the final states. In fact a detailed analysis of the state evolution showed that the two controls trigger two different control mechanisms. The L^2 control, containing the Bohr frequencies ω_1 and ω_2, induces a coupling of the two Schrödinger equations which corresponds to transitions in the electronic structure. The H_0^1 control forces the state to oscillate left and right in the lower potential well, like in a resonantly driven oscillator, until it crosses the potential barrier to the upper well. It is not surprising that the H_0^1 control contains less high frequency components compared to the L^2 control since fast oscillations are penalized by the H_0^1 norm. It is interesting that

the choice of the cost term can have a significant influence on the control mechanism. In our application, however, it is desired to induce transition in the electronic structure. Therefor the H_0^1 norm might not be the ideal choice for this kind of application.

5 Conclusion

In summary, we have shown how a generalized Suzuki-Trotter type method can be successfully applied to quantum optimal control problems. First and second order derivatives of the resulting discrete functional can be computed efficiently within the GST framework. This makes higher order methods like Newton's method applicable. In a numerical test example with a system of one-dimensional Schrödinger equations we compared the performance of several optimization methods. The memory efficient implementation of the Schrödinger solver makes the future extension of the method to two and three dimensions feasible. Additionally, we observed that the choice of the cost term in the optimal control formulation can have a significant impact on the optimal control function. Choosing an appropriate cost term for our model system thus deserves further attention.

References

[1] Ball, J.M., Marsden, J.E., Slemrod, M.: Controllability for distributed bilinear systems. SIAM J. Control Optim. **20**(4), 575–597 (1982)

[2] Boscain, U., Caponigro, M., Chambrion, T., Sigalotti, M.: A weak spectral condition for the controllability of the bilinear Schrödinger equation with application to the control of a rotating planar molecule. Comm. Math. Phys. **311**(2), 423–455 (2012)

[3] Chambrion, T.: Periodic excitations of bilinear quantum systems. Automatica **48**(9), 2040 – 2046 (2012)

[4] Cirac, J.I., Zoller, P.: Quantum computations with cold trapped ions. Phys. Rev. Lett. **74**, 4091–4094 (1995)

[5] D'Alessandro, D.: Introduction to quantum control and dynamics. Chapman & Hall/CRC, Boca Raton, FL (2008)

[6] Götschel, S., von Tycowicz, C., Polthier, K., Weiser, M.: Reducing memory requirements in scientific computing and optimal control. In: T. Carraro, M. Geiger, S. Koerkel, R. Rannacher (eds.) Multiple Shooting and Time Domain Decomposition Methods (2014). Accepted for publication on 2014-09-18

[7] Hintermüller, M., Marahrens, D., Markowich, P.A., Sparber, C.: Optimal bilinear control of Gross-Pitaevskii equations. SIAM J. Control Optim. **51**(3), 2509–2543 (2013)

[8] Hohenester, U., Rekdal, P.K., Borzì, A., Schmiedmayer, J.: Optimal quantum control of Bose-Einstein condensates in magnetic microtraps. Phys. Rev. A **75**, 023,602 (2007)

[9] Ito, K., Kunisch, K.: Optimal bilinear control of an abstract Schrödinger equation. SIAM J. Control Optim. **46**(1), 274–287 (2007)

[10] Kosloff, R., Rice, S., Gaspard, P., Tersigni, S., Tannor, D.: Wavepacket dancing: Achieving chemical selectivity by shaping light pulses. Chem. Phys. **139**(1), 201 – 220 (1989)

[11] Liebmann, M.: Ein effizienter Algorithmus zur numerischen Simulation von zeitabhängigen Problemen aus der Quantenmechanik. Diploma thesis, University of Graz, Institute for Mathematics and Scientific Computing (2005)

[12] Maday, Y., Turinici, G.: New formulations of monotonically convergent quantum control algorithms. J. Chem. Phys. **118**(18), 8191–8196 (2003)

[13] Najfeld, I., Havel, T.F.: Derivatives of the matrix exponential and their computation. Adv. Appl. Math. **16**(3), 321 – 375 (1995)

[14] Nocedal, J., Wright, S.J.: Numerical optimization, second edn. Springer, New York (2006)

[15] Peirce, A.P., Dahleh, M.A., Rabitz, H.: Optimal control of quantum-mechanical systems: existence, numerical approximation, and applications. Phys. Rev. A (3) **37**(12), 4950–4964 (1988)

[16] Shapiro, M., Brumer, P.: Quantum Control of Molecular Processes. John Wiley & Sons, Hoboken (2012)

[17] Sternberg, J., Griewank, A.: Reduction of storage requirement by checkpointing for time-dependent optimal control problems in ODEs. In: M. Bcker, G. Corliss, U. Naumann, P. Hovland, B. Norris (eds.) Automatic Differentiation: Applications, Theory, and Implementations, *Lecture Notes in Computational Science and Engineering*, vol. 50, pp. 99–110. Springer Berlin Heidelberg (2006)

[18] Suzuki, M.: Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations. Physics Letters A **146**(6), 319 – 323 (1990). DOI http://dx.doi.org/10.1016/0375-9601(90)90962-N

[19] Suzuki, M.: General theory of fractal path integrals with applications to manybody theories and statistical physics. Journal of Mathematical Physics **32**(2), 400–407 (1991). DOI http://dx.doi.org/10.1063/1.529425

[20] Turinici, G.: Beyond bilinear controllability: applications to quantum control. In: Control of coupled partial differential equations, vol. 155, pp. 293–309. Birkhäuser, Basel (2007)

[21] Turinici, G., Rabitz, H.: Quantum wavefunction controllability. Chem. Phys. **267**(13), 1 – 9 (2001)

[22] von Winckel, G., Borzì, A.: Computational techniques for a quantum control problem with H^1-cost. Inverse Problems **24**(3), 034,007 (2008)

[23] von Winckel, G., Borzì, A., Volkwein, S.: A globalized Newton method for the accurate solution of a dipole quantum control problem. SIAM J. Sci. Comput. **31**(6), 4176–4203 (2009/10)

[24] Zhu, W., Botina, J., Rabitz, H.: Rapidly convergent iteration methods for quantum optimal control of population. J. Chem. Phys. **108**(5), 1953–1963 (1998)