SOLVING BANG-BANG PROBLEMS USING THE IMMERSED INTERFACE METHOD AND INTEGER PROGRAMMING*

RYAN H. VOGT* ‡ §, SARAH STRIKWERDA†

Abstract. In this paper we study numerically solving optimal control problems with bang-bang control functions. We present a formal Lagrangian approach for solving the optimal control problem, and address difficulties encountered when numerically solving the state and adjoint equations by using the immersed interface method. We note that our numerical approach does not approximate the discontinuous control function with smooth functions, instead we solve the true bang-bang optimal control problem. Our approach for solving the optimal control problem uses an adjoint-based gradient. We use the gradient in our first-order trust-region method to generate a local minimizing control. We present detailed numerical results to demonstrate the effectiveness of our method.

Key words. ODE-constrained optimization, PDE-constrained optimization, finite-difference methods, immersed-interface method, bang-bang control, optimal control

AMS subject classifications. 65K10, 49M05, 65M08, 65M12, 90C10

1 Introduction Optimal control problems are frequently studied due to their importance in a wide range of fields including economics, biological systems, and engineering[1, 2]. In these problems, we seek to find a control that minimizes a cost functional subject to a differential equation. These problems can quickly become very complex. Therefore, in many applications a solution that only takes on a few discrete values is sought after. Such a control is called a bang-bang control. The most common bang-bang control is one which takes on the value of 0 or 1. This is called a bang-bang (on-off) control, and it abruptly changes between extreme values. The change between these values is referred to as a "bang". Bang-bang controls have been used to study how to control the population of invasive species [3], regulate chemotherapy [4], and make decisions about stock options [1].

In the current state of the art approaches to solving bang-bang optimal control problems [5, 6, 7], which are a special case of mixed-integer optimal control problems (MIOCPs), several assumptions are made. First, the solution to the different equation constraint is continuous. Second, that when numerically solving the differential equation constraint, the switching points must align with the computational mesh. In this work, we provide a framework to solve bang-bang optimal control problems without assuming regularity in the state, or the computational mesh having to align with the switching points. In this paper we develop finite-difference schemes associated with the bang-bang control problem that are consistent, stable and convergent; using the immersed interface method (IIM). Next we introduce an adjoint-based trust-region method to solve the reduced-space formulation of the bang-bang control problem.

We take a formal Lagrangian approach, assuming a reduced-space model. In other words, we assume that given a control v, there exists a unique state T dependent on v. This is to say that the state-to-control operator $T(v)$ is injective. We note that our method can also readily be applied to full-space formulations as well. In our case of a reduced-space formulation, we derive a gradient for v by solving the strong form of the state and adjoint equation. We address the lack of regularity of the state and

*North Carolina State University, Department of Mathematics, rvogt2@ncsu.edu
†North Carolina State University, Department of Mathematics, slstrikw@ncsu.edu
‡Argonne National Laboratory, Mathematics and Computer Science Division
§Lawrence Livermore National Laboratory, Center for Applied Scientific Computing
*Submitted to the editors April 13, 2021.

adjoint equation, with IIM. We then use this adjoint-based gradient in a trust-region method for binary optimization to find a local minimizing bang-bang control.

To demonstrate our approach for solving bang-bang problems, we focus on numerically solving the following temperature control problem:

$$
\begin{aligned}
\underset{T,\, v}{\text{minimize}} \quad & \frac{1}{2}\int_{\Omega}(T-\hat{T})^2 dt \\
\text{subject to} \quad & \frac{dT}{dt} = -K(T-T_s) + Cw + f \qquad \text{on } \Omega, \\
& \Omega = \bigcup_{i=1}^{N} \Omega_i, \\
& w(t) = \sum_{i=1}^{N} v_i \mathbb{1}_{\Omega_i \setminus \partial\Omega_i}(t) \\
, \quad & v_i \in \{0,1\} \quad \forall i = 1...N, \\
& \Omega_i = [\tau_i, \tau_{i+1}] \quad \forall\, i = 1, ..., N, \\
& \tau_1 = 0, \quad \tau_{N+1} = t_{\text{final}}, \\
& \Omega = [0, t_{\text{final}}], \\
& T(0) = T_0
\end{aligned}
\tag{1.1}
$$

where we assume that $T(t)$ is the temperature of a body at time t, $\hat{T}(t)$ is the target temperature profile, $f \in C(\Omega_i \setminus \partial\Omega_i)$, and $w(t)$ the control function is either "on" or "off" at a time t. We represent the control function $w(t)$ as a linear combination of indicator functions $\mathbb{1}$, and assign the control function value to be v_i on $\Omega_i \setminus \partial\Omega_i$. In this paper we assume that the support of the bang-bang choices, defined by Ω_i, are equal length i.e,

$$
|\Omega_i| = \frac{T}{N}, \quad \forall i = 1...N.
$$

We also assume that $K,\ C,\ t_{\text{final}} > 0$. We note that this assumption does not have to hold for our method to work; it can very well be the case the the intervals lengths can vary, but we choose them to be constant length in this work. We lastly note that we will refer to the objective in this paper as

$$
\mathcal{J} = \frac{1}{2}\int_{\Omega}(T-\hat{T})^2 dt. \tag{1.2}
$$

Even though this paper focuses on the specific model problem (1.1), we explain at each step taken to solve (1.1) how our approach can be tailored to other bang-bang problems.

In this paper we develop a first order, stable convergent method to solve the state and adjoint equations. The method takes advantage of IIM [8]. The IIM was originally developed to solve elliptic partial-differential equations with discontinuous coefficients, also known as interface problems, with a finite-difference approach. The IIM has been successfully applied to generate high order methods for solving equations where discontinuities are present [9, 10, 11]. In our work we use the IIM to solve the adjoint and state equation due to the bang-bang control. We establish a modifed

Euler's method that can solve the state and adjoint equation, while also obtaining second order accuracy, stability, and convergence of the numerical solution.

In practice, analytic solutions to optimal control problems are difficult to obtain. In absence of analytic techniques, numerical methods are a powerful tool to solve optimal control problems. In our example here, the bang-bang control makes it difficult to apply existing numerical techniques to solve ODE's because $T \notin C^1(\Omega)$. One at first glance may look at (2.4) and believe that a simple method, such as Euler's method, may be readily applied to solve the equation numerically. However, Eulers method assumes that $T \in C^2(\Omega)$ for the method to be applicable. This motivates our approach of solving the state equation with IIM.

We note in our framework the effort of numerically solving the bang-bang problem lies in solving the differential equations, and building the gradient. The amount of bangs present in the bang-bang problem, from the perspective of applying our trust-region method, has little impact on the overall computational effort because we solve a sequence of binary knapsack problems which can be solved easily.

2 The Lagrangian and the relaxed gradient To calculate the gradient of the objective function with the respect to the control parameters v, $\frac{\partial \mathcal{J}}{\partial v}$, we use a formal Lagrangian approach. We first relax the condition of $v \in \{0,1\}^N$ to a continuous relaxation: $0 \leq v \leq 1$. We do this so that we have the notion of a continuous gradient. However, because this gradient expression is valid for any $0 \leq v \leq 1$, then it will be valid when evaluated at a $v \in \{0,1\}^N$. We introduce the Lagrangian:

$$\mathcal{L}(T,\lambda,v) = \frac{1}{2}\int_\Omega (T-\hat{T})^2 dt + \int_\Omega \lambda\left(\frac{dT}{dt} + K(T-T_s) - Cw - f\right)dt, \tag{2.1}$$

where λ is the adjoint variable (Lagrangian multiplier).

Next we derive the weak state and weak adjoint equations by calculating the following variational (Gateaux) derivatives:

$$\mathcal{L}_\lambda[\tilde{\lambda}] = \frac{d}{d\epsilon}\Big(\mathcal{L}(T,\lambda+\epsilon\tilde{\lambda},v)\Big)\Big|_{\epsilon=0} = 0 \quad \text{(Weak State Equation)}$$

$$\mathcal{L}_T[\tilde{T}] = \frac{d}{d\epsilon}\Big(\mathcal{L}(T+\epsilon\tilde{T},\lambda,v)\Big)\Big|_{\epsilon=0} = 0 \quad \text{(Weak Adjoint Equation).}$$

The explicit form of these equations are:

$$\int_\Omega \left(\frac{dT}{dt} + K(T-T_s) - Cw - f\right)\tilde{\lambda}dt = 0 \quad \forall \tilde{\lambda} \in H^1(\Omega), \tag{2.2}$$

$$\int_\Omega \left(-\frac{d\lambda}{dt} + K\lambda + T - \hat{T}\right)\tilde{T}dt = 0 \quad \forall \tilde{T} \in H^1(\Omega) \tag{2.3}$$

The state and adjoint solutions are not $C^1(\Omega)$, however from standard ODE theory we see the state and adjoint equations are $C^1(\Omega_i \setminus \partial\Omega_i)$. Therefore, we write the strong form of these equations restricted to the set $\Omega_i \setminus \partial\Omega_i \; \forall i$:

$$\begin{cases} \frac{dT}{dt} = -K(T-T_s) + Cw + f \quad \text{(Restricted Strong State Equation)} \\ T(0) = T_0 \end{cases} \tag{2.4}$$

$$\begin{cases} -\frac{d\lambda}{dt} = -K\lambda - (T-\hat{T}) + g \quad \text{(Restricted Strong Adjoint Equation)} \\ \lambda(T) = 0. \end{cases} \tag{2.5}$$

In the remainder of the paper we refer the restricted state and adjoint equations as just the state and adjoint equations. We note that in the adjoint equation $g = 0$. However, we include $g \in C(\Omega_i \setminus \partial\Omega_i)$ to test our adjoint solver in Section 9 using the method of Manufactured Solution and the IIM.

The i-th component of the gradient with respect to v_i is then:

$$\frac{\partial \mathcal{J}}{\partial v_i} = \frac{\partial \mathcal{L}}{\partial v_i} = -\int_{\Omega_i} C\lambda dt. \tag{2.6}$$

So to obtain a gradient associated with a fixed v, we first solve the strong state equation, then solve the strong adjoint equation. Once we have the adjoint variable, we then integrate a scaled version it over Ω_i to obtain the i-th component of the gradient.

3 The discretization of the restricted state Equation In our optimal control problem, and other bang-bang problems, the IIM is readily applicable to solve the state equation. This is because the presence of the bang-bang control can cause, as seen in (2.4), the state solution to not be $C^1(\Omega)$. However, we can readily apply the IIM to this problem despite the lack of regularity of the state, or its derivatives, and have theoretical guarantees that the numerical solution is correct.

For many numerical methods, such as Euler or Runge-Kutta, there is an assumption of high regularity in state. Indeed in these methods, the error terms will contain a high order derivative term. If the solution to an ODE doesn't have the sufficient regularity to apply these numerical schemes, then one cannot guarantee the consistency of the method. If the consistency of the numerical method cannot be established then there is no hope to establish the convergence of the numerical solution to the true solution. This means one cannot be confident that the numerical method approximates the state solution correctly.

In order to apply IIM to the state equation (2.4), we begin by deriving the jump conditions associated with (2.4). The jump conditions describe how the solution and it's derivatives jump when a bang occurs. The jump conditions are used to create our numerical scheme. In our problem, we introduce N binary variables to model the bang-bang control, so there would be at most $N-1$ bangs (switches). Each one of these bangs will occur at

$$t = \alpha_i = \Omega_i \cap \Omega_{i+1} \quad i = 1, ..., N-1. \tag{3.1}$$

For our state scheme, we focus on developing a first-order method, so we need only to derive the the jump in the state and its first derivative over these bang points. We note if one wishes to apply the IIM to yield a method of order m, then one must derive the jump conditions up to the m-th derivative.

At each bang time $t = \alpha_i$, we define the jump in the state and its first derivative to be:

$$[T]_{t=\alpha_i} = T^+(\alpha_i) - T^-(\alpha_i) \tag{3.2}$$

$$\left[\frac{dT}{dt}\right]_{t=\alpha_i} = \frac{dT}{dt}^+(\alpha_i) - \frac{dT}{dt}^-(\alpha_i), \tag{3.3}$$

where we we denote "+" to represent the solution and its derivative to the right of the bang point and "−" to the left of the bang point. In general we write the k-th order derivative as:

$$\left[\frac{d^kT}{dt^k}\right]_{t=\alpha_i} = \frac{d^kT}{dt^k}^+(\alpha_i) - \frac{d^kT}{dt^k}^-(\alpha_i),$$

where we assume that the zero order derivative is just the state itself.

In many physical examples, even though the derivative is discontinuous (therefore a non-zero jump) the state is continuous so

$$[T]_{t=\alpha_i} = 0.$$

This would be known as a direct IIM because we know the jump in the state explicitly. However, in this work we make no such assumption since for general problems there is no manner to know that the state is continuous across the bang points. Therefore in this work we assume it to be unknown i.e

$$[T]_{t=\alpha_i} = q_i, \tag{3.4}$$

where q_i is to be determined. When the jump conditions are not known explicitly, the method is called the Augmented IIM [11], and q_i for $i = 1, .., N-1$ are called the augmented variables. From (2.4) we see that the jump condition for the derivative of the state is

$$\left[\frac{dT}{dt}\right]_{t=\alpha_i} = -K[T]_{t=\alpha_i} + C[w]_{t=\alpha_i} + [f]_{t=\alpha_i}, \tag{3.5}$$

because we assume K, and T_s to be constant in this work. We then express the jump of the first-derivative in terms of the augmented variables (3.4):

$$\left[\frac{dT}{dt}\right]_{t=\alpha_i} = -Kq_i + C[w]_{t=\alpha_i} + [f]_{t=\alpha_i}.$$

Next we focus on deriving the IIM scheme for our problem. We discretize the interval $[0, t_{\text{final}}]$ into N_t steps with $t_n = n\Delta t$, $n = 0, ...N_t$ and $\Delta t = \frac{t_{\text{final}}}{N_t}$. The idea of the IIM scheme is to use a standard scheme away from the bang points and derive a "corrected" scheme around the bang points to obtain a consistent scheme. In this work we base our scheme around Euler's method. We say that if $t_n \leq \alpha_i \leq t_{n+1}$ for some i then t_n is an irregular grid point. If this is not the case, then the grid point is said to be regular. At regular grid points we apply Euler's method:

$$T^{n+1} = T^n + \Delta t(-K(T^n - T_s) + Cw^n + f^n) + \mathcal{O}(\Delta t). \tag{3.6}$$

In the case t_n is an irregular grid point we use the method of unknown coefficients:

$$c_1^n T^{n+1} + c_2^n T^n - \bar{C}^n(c_1, c_2) = -K(T^n - T_s) + Cw^n + f^n + \mathcal{O}(\Delta t), \tag{3.7}$$

where the goal is to find c_1^n, c_2^n and the correction term, $\bar{C}^n(c_1^n, c_2^n)$, that maintains the same order of error as the standard scheme at the irregular grid points, which in this case is first-order accuracy. Since a grid point in out interval is either regular or irregular, we will obtain the order of accuracy of the scheme globally even though the state nor its derivatives need to be continuous on the interval.

We use the Taylor series to determine c_1, c_2 and the correction term with first order accuracy. We expand the derivative in terms of the "$-$" side about α_i:

$$\frac{dT}{dt}^{-}(\alpha_i) = c_1 T^{n+1} + c_2 T^n = \tag{3.8}$$
$$c_1^n\left(T^+(\alpha_i) + \frac{dT}{dt}^{+}(\alpha_i)(t_{n+1} - \alpha_i) + \mathcal{O}\left((t_{n+1} - \alpha_i)^2\right)\right) +$$

$$c_2^n\bigg(T^-(\alpha_i) + \frac{dT}{dt}^-(\alpha_i)(t_n-\alpha_i) + \mathcal{O}\Big((t_n-\alpha_i)^2\Big)\bigg).$$

Note that we expand t_{n+1} in terms of "+" and t_n in terms of "−" because they are to the right and left of the bang point α_i, respectively.

Next we use the jump conditions (3.2) to write T^+ and $\frac{dT}{dt}^+$ in terms of T^- and $\frac{dT}{dt}^-$ so that we can equate the two sides to find c_1, c_2 and the correction term. When equating the two sides, we find that c_1, c_2 satisfy the following system of equation:

$$\begin{aligned} c_1^n + c_2^n &= 0 \\ c_1^n(t_{n+1}-\alpha_i) + c_2^n(t_n-\alpha_i) &= 1, \end{aligned}$$

and the analytic solution to this system is

$$c_1^n = \frac{1}{\Delta t}, \quad c_2^n = -\frac{1}{\Delta t}. \tag{3.9}$$

The corresponding correction term is then

$$\bar{C}^n = \frac{1}{\Delta t}\bigg(q_i + (-Kq_i + c[w]_{t=\alpha_i} + [f]_{t=\alpha_i})(t_n-\alpha_i)\bigg). \tag{3.10}$$

4 The discretization of the restricted adjoint equation We build on Section 3 to solve the adjoint equation numerically. Like the state equation, we first derive the jump conditions. Due to the fact that the adjoint equation is solved backwards in time, we introduce the notion of "backward" jump in terms of our previous definitions $[\cdot]^{\leftarrow}_{t=\alpha_i} = -[\cdot]_{t=\alpha_i}$ i.e

$$[\lambda]^{\leftarrow}_{t=\alpha_i} = -[\lambda]_{t=\alpha_i} = \lambda^- - \lambda^+ = q_i^\lambda, \tag{4.1}$$

$$\left[\frac{d\lambda}{dt}\right]^{\leftarrow}_{t=\alpha_i} = -\left[\frac{d\lambda}{dt}\right]_{t=\alpha_i} = Kq_i^\lambda - q_i - [g]^{\leftarrow}_{t=\alpha_i}, \tag{4.2}$$

and we introduce a new set of augmented variables q_i^λ that are to be determined. We maintain the same time mesh as the state equation, and move backward instead of forward in time. When moving backwards, from t_n to t_{n-1}, we say t_n is an irregular grid point for the adjoint discretization if $t_{n-1} \le \alpha_i \le t_n$, and we say t_n is a regular grid point otherwise. In the case t_n is a regular grid point, we use a backward finite-difference approximation of the adjoint variable's first derivative:

$$\frac{\lambda^n - \lambda^{n-1}}{\Delta t} = -K\lambda^n - (T^n - \hat{T}^n) + g^n + O(\Delta t).$$

For an irregular grid point, we expand on the $'+'$ side, because we are marching backwards:

$$\frac{d\lambda}{dt}^+(\alpha_i) = c_1^{n,\lambda}(\lambda^+(\alpha_i) + \frac{d\lambda}{dt}^+(\alpha_i)(t_n-\alpha_i)) + c_2^{n,\lambda}(\lambda^-(\alpha_i) + \frac{d\lambda}{dt}^-(\alpha_i)(t_{n-1}-\alpha_i)).$$

Like the state equation, we use the method of manufactured coefficients, and the jump conditions for the adjoint variables (4.1) to find the system of equations that $c_1^{n,\lambda}, c_2^{n,\lambda}$ satisfy which we find to be

$$c_1^{n,\lambda} + c_2^{n,\lambda} = 0$$

$$c_1^{n,\lambda}(t_n - \alpha_i) + c_2^{n,\lambda}(t_{n-1} - \alpha_i) = 1.$$

Solving the systems of equations, we find that

$$c_1^{n,\lambda} = -\frac{1}{\Delta t}, \quad c_2^{n,\lambda} = \frac{1}{\Delta t},$$

with corresponding correction term

$$\bar{C}^{n,\lambda} = \frac{1}{\Delta t}\left(q_i^\lambda + (Kq_i^\lambda - q_i - [g]_{t=\alpha_i}^{\leftarrow})(t_{n-1} - \alpha_i)\right).$$

The scheme at irregular grid points is then

$$\frac{\lambda^n - \lambda^{n-1}}{\Delta t} + \bar{C}^{n,\lambda} = -K\lambda^n - (T^n - \hat{T}^n) + g^n + O(\Delta t)$$

5 Stability, consistency, and convergence of the state and adjoint finite-difference schemes We must discuss stability, consistence, and convergence of numerical schemes, for regular and irregular grid points independently. Because a grid point is either regular or irregular, by showing stability, consistency, and convergence in each case we show the overall scheme is stable, consistent, and convergent.

In practice, the scheme used for the IIM at regular grid points is a standard method, so the stability, consistency, and convergence of the method is already established. What is left to do is to establish the stability, consistency, and convergence for the scheme at irregular grid points.

In practice, we note that consistency of a method is more readily obtainable than stability. We begin by showing that the numerical scheme is consistent for the state scheme.

PROPOSITION 5.1. *The state scheme is consistent with local truncation error of* $\mathcal{O}(\Delta t)$ *at both regular and irregular grid points, therefore the method has global truncation error* $\mathcal{O}(\Delta t)$.

Proof. We begin by showing the scheme is consistent at irregular grid points. We substitute the finite difference weights found in (3.9) and correction term, (3.10), into the numerical scheme for the state equation in (3.7). We note that the local truncation error of the method τ becomes

$$\tau = \frac{1}{\Delta t}\mathcal{O}((t_{n+1} - \alpha_i)^2) - \frac{1}{\Delta t}\mathcal{O}((t_n - \alpha_i)^2).$$

We note that at the irregular grid point t_n, $t_n \le \alpha_i \le t_{n+1}$, and $t_{n+1} - t_n = \Delta t$. Therefore,

$$\tau = \frac{1}{\Delta t}\mathcal{O}((t_{n+1} - \alpha_i)^2) - \frac{1}{\Delta t}\mathcal{O}((t_n - \alpha_i)^2) \le \frac{1}{\Delta t}\mathcal{O}(\Delta t^2) - \frac{1}{\Delta t}\mathcal{O}(\Delta t^2) = \mathcal{O}(\Delta t)$$
$$\implies \tau = \mathcal{O}(\Delta t),$$

which is to say that the method is a first-order method at the irregular grid points. We then observe that

$$\lim_{\Delta t \to 0} \tau = 0,$$

so the method at irregular grid points is consistent as well.

Next we refer to [12] to show that the scheme is consistent at regular grid points. We note that the scheme at regular grid points is the standard forward Euler method. This scheme has a local truncation error of $\mathcal{O}(\Delta t)$, so the scheme is first-order at regular grid points, and moreover is consistent because

$$\lim_{\Delta t \to 0} \mathcal{O}(\Delta t) = 0.$$

We have shown that the scheme for the state equation is first-order accurate and consistent at both the regular and irregular grid points, so the state scheme is consistent and first-order accurate. ∎

We omit a proof that shows the of the adjoint scheme is first-order and consistent because the argument to prove the result is identical to the state equation proof, just applied to the adjoint scheme.

Next, we show the stability of the state and adjoint schemes by referring to the literature. In both cases we write the equations in the form:

$$\frac{dT}{dt} = \Psi_1(T, t),$$
$$\frac{d\lambda}{dt} = \Psi_2(\lambda, T, t).$$

Because Ψ_1 and Ψ_2 are Lipschitz with respect to the variables T, and λ on $\Omega_i \setminus \partial\Omega_i$ for all $i = 1..N$, then a one step method is stable as shown in [12, Theorem 5.7.1, p. 345]. Our methods for the state and adjoint scheme are one step methods, and the correction terms at irregular grid points for both the state and adjoint scheme are bounded constants (which means Ψ_1 and Ψ_2 are Lipschitz for regular and irregular grid points) so then the overall scheme is stable.

Now we show the convergence of the state and adjoint schemes. We again refer to the literature, where if a one step method is consistent and stable, then it is convergent [12, Theorem 5.7.2, p. 348]. Since we have proven the consistency and stability of the method for both regular and irregular grid points we can establish the convergence of the solution at regular and irregular grid points due to this theorem. Therefore, we have established the consistency, stability, and convergence of both the state and adjoint numerical schemes.

We note that in our framework even though the solutions are not continuous on the entire mesh, it is on the pieces, which allows us to establish the consistency, stability and convergence of our method due to our IIM approach. If a schemes consistency, stability, and convergence has been established for differential equations with high regularity, those proofs can traditionally be altered slightly to establish these properties for the irregular grid points. To our knowledge this has not been possible for finite-difference methods applied to bang-bang control problems before this work.

6 The Discrete State and Adjoint Systems

Suppose we are given a control v, then augmented IIM for the state equation can be written as the block matrix system:

$$\begin{bmatrix} \boldsymbol{A_s} & \boldsymbol{B_s} \\ \boldsymbol{C_s} & \boldsymbol{D_s} \end{bmatrix} \begin{bmatrix} \boldsymbol{T} \\ \boldsymbol{q} \end{bmatrix} = \begin{bmatrix} \boldsymbol{S_T} \\ \boldsymbol{S_q} \end{bmatrix}, \tag{6.1}$$

where

$$\boldsymbol{A_s} \in \mathbb{R}^{N_t+1 \times N_t+1}, \qquad \boldsymbol{B_s} \in \mathbb{R}^{N_t+1 \times N-1},$$

$$\boldsymbol{C_s} \in \mathbb{R}^{N-1\times N_t+1}, \qquad \boldsymbol{D_s} \in \mathbb{R}^{N-1\times N-1},$$
$$\boldsymbol{S_T} \in \mathbb{R}^{N_t+1\times 1}, \qquad \boldsymbol{S_q} \in \mathbb{R}^{N-1\times 1},$$
$$\boldsymbol{T} \in \mathbb{R}^{N_t+1\times 1}, \qquad \boldsymbol{T}_i = T(t_{i-1}), \quad i = 1,..,N_t+1,$$
$$\boldsymbol{q} \in \mathbb{R}^{N-1\times 1}, \qquad \boldsymbol{q}_i = q_i, \quad i = 1,..,N-1.$$

In the rows of the upper block of the matrix system we enforce the time step scheme based on if a row corresponds to an irregular or regular grid point as discussed previously. In the lower block rows, we enforce the relationship between the state and state augmented variable for the zero order jump condition.

In a similar manner given $\boldsymbol{T}, \boldsymbol{q}$, and control v, augmented IIM for the adjoint equation can be written as the block matrix system:

$$\begin{bmatrix} \boldsymbol{A_\lambda} & \boldsymbol{B_\lambda} \\ \boldsymbol{C_\lambda} & \boldsymbol{D_\lambda} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{q^\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{S_\lambda} \\ \boldsymbol{S_{q^\lambda}} \end{bmatrix}, \tag{6.2}$$

where

$$\boldsymbol{A_\lambda} \in \mathbb{R}^{N_t+1\times N_t+1}, \qquad \boldsymbol{B_\lambda} \in \mathbb{R}^{N_t+1\times N-1},$$
$$\boldsymbol{C_\lambda} \in \mathbb{R}^{N-1,N_t+1}, \qquad \boldsymbol{D_\lambda} \in \mathbb{R}^{N-1\times N-1},$$
$$\boldsymbol{S_\lambda} \in \mathbb{R}^{N_t+1\times 1}, \qquad \boldsymbol{S_{q^\lambda}} \in \mathbb{R}^{N-1\times 1},$$
$$\boldsymbol{\lambda} \in \mathbb{R}^{N_t+1\times 1} \qquad \boldsymbol{\lambda}_i = \lambda(t_{N_t-i+1}), \quad i = 1,..,N_t+1,$$
$$\boldsymbol{q^\lambda} \in \mathbb{R}^{N-1\times 1} \qquad \boldsymbol{q^\lambda}_i = q_i^\lambda, \quad i = 1,..,N-1.$$

Like in the case of the state equation, we enforce the scheme at regular and irregular grid points in the upper block, and in the lower block enforce the augmented variable on the state.

A consideration of this paper is not to derive appropriate preconditions to apply to the augmented state or augmented adjoint systems. However, constructing such a preconditioner would help speed up the computational time required to solve the systems.

7 Discretization of The Objective Function and Gradient In Section 6, we wrote the discretization schemes to produce a discretized state solution $\boldsymbol{T}$, and discretized adjoint solution $\boldsymbol{\lambda}$, for a fixed control v. In this section we show how we produce an objective function value and gradient for these discretized solutions.

In our optimal control problem the objective function is:

$$\frac{1}{2}\int_\Omega (T-\hat{T})^2 dt.$$

We approximate the objective function with the trapezoid rule:

$$\mathcal{J} = \frac{1}{2}\int_\Omega (T-\hat{T})^2 dt = \int_\Omega \hat{f}(t)dt = \sum_{i=1}^{N_t} \frac{\Delta t}{2}\left(\hat{f}(t_{i-1}) + \hat{f}(t_i)\right) + \mathcal{O}(\Delta t^2), \tag{7.1}$$

where

$$\hat{f}(t) = \frac{1}{2}(T(t)-\hat{T}(t))^2. \tag{7.2}$$

For simplicity of notation, we define the numerical approximation of the objective function to be

$$\hat{J}^{\Delta t} = \sum_{i=1}^{N_t} \frac{\Delta t}{2}\Big(\hat{f}(t_{i-1}) + \hat{f}(t_i)\Big). \tag{7.3}$$

We approximate the $i-$th component of the gradient, associated with the binary variable v_i, using the trapezoid rule:

$$\frac{\partial \mathcal{J}}{\partial v_i} = -\int_{\Omega_i} C\lambda dt = \int_{\Omega_i} \hat{g}(t)dt = \sum_{k=1}^{\hat{N}_t^i} \frac{\Delta t}{2}\Big(\hat{g}(\hat{t}_{k-1}) + \hat{g}(\hat{t}_k)\Big) + \mathcal{O}(\Delta t^2), \tag{7.4}$$

where

$$\hat{g}(t) = -C\lambda(t). \tag{7.5}$$

The notation $\hat{N}_t^i$ corresponds to the number of time steps in the mesh that exist within Ω_i, and $\hat{t}$ corresponds to the specific time steps that exist in Ω_i. Again, for simplicity of notation, we define the numerical approximation of i-th component of the gradient vector to be

$$\Big(g_{\Delta t}\Big)_i = \sum_{k=1}^{N_t^i} \frac{\Delta t}{2}\Big(\hat{g}(\hat{t}_{k-1}) + \hat{g}(\hat{t}_k)\Big) \tag{7.6}$$

8 A trust-region method for bang-bang control In [13], a trust-region method is introduced to solve binary partial-differential equation constrained optimization problems. We note that this method is readily applicable to our problem, and is outlined in Algorithm 8.1. The algorithm can be viewed as solving a sequence of differential equations and binary knapsack problems to yield a local minimizing binary control. A strength of this algorithm is that the computational effort is not bottlenecked by the number of binary variables present in the optimal control problem. This is because the binary variables appear in the trust-region subproblem; the trust-region subproblem is a knapsack problem, so it is computationally inexpensive to solve even for a large amount of binary variables. Instead, the computational effort required to execute the trust-region method is focused on solving the state and adjoint equation, and then building the gradient.

9 Numerical experiments We provide numerical evidence that our approach to solve the bang-bang state equation, using IIM, provides the correct numerical solution. In our simulations, we solve 2.4 with parameters $f = 0$, $C = 3$, $K = 1$, $T_0 = 70$, $T_s = 50$, $t_{\text{final}} = 10, N = 10.$, and choose v in the following manner:

$$\begin{cases} v_i = 1 & i \text{ is odd} \\ v_i = 0 & i \text{ is even,} \end{cases} \tag{9.1}$$

for $i = 1, .., N$. In this instance, we can derive the exact solution to the state equation:

$$T_{\text{true}}(t) = \begin{cases} T_0 e^{-Kt} + T_s - T_s e^{-Kt} + \frac{C}{K} - \frac{C}{K} e^{-Kt} & t \in \Omega_i, i \text{ is odd} \\ T_0 e^{-Kt} + T_s - T_s e^{-Kt} & t \in \Omega_i, i \text{ is even.} \end{cases} \tag{9.2}$$

Algorithm 8.1 Steepest-Descent Trust-Region Algorithm.

Given initial trust-region radius $\Delta_0 = \bar{\Delta} \geq 1$ and initial guess $v^{(0)} \in \{0,1\}^N$
Select an acceptance step parameter $\bar{\rho}$, and set $k \leftarrow 0$
Evaluate the objective function $\hat{J}_{\Delta t}^{(k)} = \hat{J}^{\Delta t}(v^{(k)})$ and the gradient $g_{\Delta t}^{(k)} = \nabla_v \hat{J}^{\Delta t}(v^{(k)})$
while $\Delta_k \geq 1$ **do**
  Solve the trust-region (knapsack) subproblem for $\widehat{v}$:

$$\begin{aligned} \widehat{v} = \underset{v}{\text{minimize}} \quad & {g_{\delta t}^{(k)}}^T \left(v - v^{(k)}\right) + \hat{J}_{\delta t}^{(k)} \\ & \|v - v^{(k)}\|_1 \leq \Delta_k \\ & v \in \{0,1\}^N \end{aligned}$$

  Evaluate the objective $\hat{J}_{\Delta t}(\widehat{v}, T(\widehat{v}))$ by solving state equations with $\widehat{v}$
  Compute the ratio of actual over predicted reduction: $\rho_k = \frac{\hat{J}_{\Delta t}^{(k)} - \hat{J}_{\Delta t}(\widehat{v}, \psi(\widehat{v}))}{-\left(g_{\Delta t}^{(k)}\right)^T \left(\widehat{v} - v^k\right)}$
  if $\rho_k > \bar{\rho}$ **then**
    Accept the step: $v^{(k+1)} = \widehat{v}$, and evaluate the gradient $g_{\Delta t}^{(k+1)} = \hat{J}'(v^{(k+1)})$
    if $\|v^{(k+1)} - v^{(k)}\|_1 = \Delta_k$, **then** increase the trust-region radius $\Delta_{k+1} = 2\Delta_k$;
  else if $\rho_k > 0$ **then**
    Accept the step $v^{(k+1)} = \widehat{v}$, and evaluate the gradient $g_{\Delta t}^{(k+1)} = \hat{J}'(v^{(k+1)})$
    Keep trust-region radius unchanged $\Delta_{k+1} = \Delta_k$
  else
    Reject the step, set $v^{(k+1)} = v^{(k)}$, and copy the gradient $g_{\Delta t}^{(k+1)} = g_{\Delta t}^{(k)}$
    Reduce the trust-region radius $\Delta_{k+1} = \text{floor}\left(\frac{\Delta_k}{2}\right)$
  Set $k \leftarrow k + 1$

We numerically solve 2.4 in several instances containing a different number of time steps i.e $N_t = 2^p$ for $p = 5..11$ using both Eulers method and our IIM approach. We record the maximal of the absolute error between the numerical and true solution in each simulation, and calculate the convergence order as we refine the mesh for both Euler method and our IIM approach; we show these results in Table 1. We note that we calculate the error in the infinity norm, and the convergence γ^* by:

$$\gamma^* = \frac{\log\left(\frac{E^*_{\frac{N_t}{2}}}{E^*_{N_t}}\right)}{\log(2)}.$$

In the IIM approach [11], we expect the average convergence order to reflect the overall convergence of the method. If we average the convergence order for the IIM approach in 1, we see that the order of the method does indeed reflect a convergence order of 1. In comparison, as we observe the convergence order for Euler's method, we see that the convergence order average is not close to 1. Instead, as we refine the mesh the convergence order is approaching 0. This is not surprising, because the theoretical assumption to use Euler's method is $T \in C^2(\Omega)$, which is violated in this case. In the absence of such an assumption, one cannot guarantee the consistency, let alone the convergence of the numerical method. In Figure 1, we visually show the solutions generated from the IIM and Euler approach. We observe that the IIM captures the jumps in the state solution due to the bang-bang control, while Euler's

method does capture these jumps accurately. It is crucial to solve the state equation correctly, and as we have demonstrated, the introduction of bang-bang controls will cause traditional numerical methods for ordinary-differential equations to fail.

On the optimal control front, being unable to solve the state equation correctly undoubtedly produces an incorrect adjoint solution. It should come as no surprise that the gradient, which is calculated with this incorrect state and adjoint solution, also will be incorrect. In Table 2 we demonstrate what occurs when we use the IIM approach to calculate the state and adjoint equation. We first observe that both the state and the adjoint solutions achieve first-order convergence. This is to say that the state solution is first order accurate, but also when we pass this solution to the adjoint equation and use the IIM approach we maintain the first order accuracy. We use the state and adjoint solution to then create a gradient. In the table, we compare the difference of the gradient and the a forward difference approximation of the gradient in the infinity norm. We observe that the gradient also obtains first order convergence, which is expected. Our results demonstrate that not only using our IIM approach solves the state and adjoint equation in the presence of bang-bang control correctly, but also that the gradient produced by using the IIM generated state and adjoint solution is consistent.

Since we have shown that our approach creates a consistent gradient, we now share results for our trust-region method applied to the problem. In our runs, we let $T_0 = 10,\ T_s = 0\ K = 0.1\ C = 2\ f = 0,\ g = 0$ and $t_{\text{final}} = 100$. In our numerical tests, we let $\hat{T}(t) = 5 + 0.5\sin(t)$. We test the trust-region method when the number of bangs are 100, 1000, and 10000. For all simulations we pick $N_t = 20000$ to guarantee a resolved bang-bang control for all the scenarios mentioned.

To execute our trust-region method, we need a feasible starting guess. While a random binary solution would be sufficient, in [13] it was observed that solving the relaxation of the problem ($0 \leq v \leq 1$) and then rounding was produced a minimum with lower objective function value. We choose to use the simple rounding heuristic:

$$[v_{\text{round}}]_i = \begin{cases} 0 & v_i < 1 \\ 1 & v_i \geq 1. \end{cases}$$

In Table 3 we report the computational effort required to solve the relaxation, the number of trust-region iterations required to find a minimizing solution, and the total time required to do the two pieces of the trust region method: solving the state and adjoint equation then constructing the gradient and executing the trust-region subproblem. We observe the amount of time taken to solve the knapsack problem is computationally negligible when compared to the time required to build a gradient. This means that with our trust-region method for binary optimization the bottleneck is the bottleneck of doing classical optimal control; we observe that the binary variables are almost inconsequential. Solving the adjoint and state equation in the most efficient manner possible is out of the scope of this work, but we note that carefully constructed preconditioners for the state and adjoint equation can also drastically decrease the computational time to solve the state and adjoint systems.

In Figure 2, we show the progress of the trust-region methods radius and objective-function value as a function of iteration count. We observe that as we increase the amount of bangs present, we increase the amount of trust-region iterations required to find a minimizing solution. Moreover we reduce the objective-function value quite quickly early in execution of the trust-region method, but then see slow convergence to a local minimum; an artifact of the non-convexity of the objective function.

In Table 4 we report the objective function value and in Figure 3 we show the optimal control, state, and target function for a different amount of bang controls. In the table, we see that we are able to make a significant percent reduction in all cases, and that the percent reduction is observed to be greater when more bangs are present. In the figure, we observe that we are successful at producing controls that create states which model the behavior of the target we are attempting to match. We observe that as we increase the number of bangs, we find an optimal control that makes the state model the target function more accurately. These results indicate that using our IIM approach, combined with an adjoint based gradient and trust-region method, does indeed yield a successful control strategy.

Table 1: Error (E) and convergence (γ) table for Euler (superscript E) method and the IIM (superscript IIM) approach applied to the state equation 2.4.

	IIM		Euler Method	
N_t	$E_{N_t}^{\text{IIM}}$	$\gamma_{N_t}^{\text{IIM}}$	$E_{N_t}^{\text{E}}$	$\gamma_{N_t}^{\text{E}}$
32	3.140		3.140	
64	1.255	1.322	1.517	1.048
128	0.574	1.124	1.431	0.084
256	1.497	-1.382	1.369	0.064
512	0.135	3.467	1.329	0.042
1024	0.067	1.013	1.328	0.001
2048	0.034	0.944	1.323	0.005

Table 2: Error and convergence table for the state, adjoint and gradient .

	State		Adjoint		Gradient	
N_t	$E_{N_t}^{\text{IIM}}$	$\gamma_{N_t}^{\text{IIM}}$	$E_{N_t}^{\text{IIM}}$	$\gamma_{N_t}^{\text{IIM}}$	$E_{N_t}^{\text{IIM}}$	$\gamma_{N_t}^{\text{IIM}}$
32	19.834		44.528		36.177	
64	7.607	1.382	9.945	2.162	19.417	0.897
128	3.375	1.172	4.283	1.215	9.933	0.966
256	1.678	1.008	1.369	1.126	4.919	1.013
512	0.832	1.011	1.329	1.090	2.517	0.966
1024	0.415	1.005	0.448	1.039	1.263	0.994
2048	0.207	1.006	0.223	1.007	0.632	0.998

10 Solving general binary bang-bang problems with our approach In this Section we summarize how the our approach can be applied to more general reduced space binary bang-bang problems. We discuss the three key components in as much generality as possible : building the Lagrangian, building the IIM schemes for the state and adjoint system, and applying the trust-region method.

Construct a Lagrangian.

Suppose we have a bang-bang optimal control problem of the form:

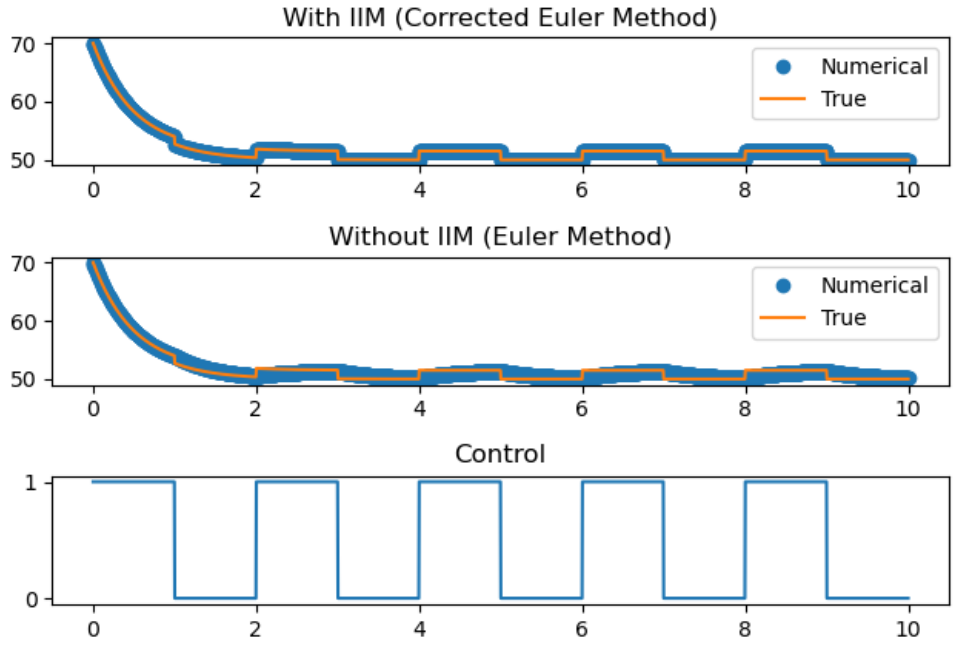
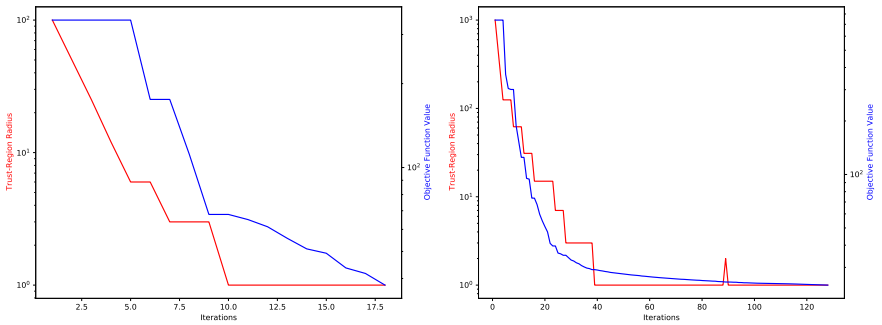


Fig. 1: Comparison of the numerical solution yielded by the IIM and Euler method to the exact solution for a fixed bang-bang control for $N_t = 2048$ from Table

Table 3: For 100, 1000, and 10000 binary variables we record the CPU times(s) for solving the relaxation to initial the trust-region method. In addition we record the CPU times(s) for computing the state, adjoint, and gradient; in addition to the time required to executing the trust-region sub problems (knapsack problems). These times are reflective of the amount of trust-region iterations required to find a minimizing bang-bang control.

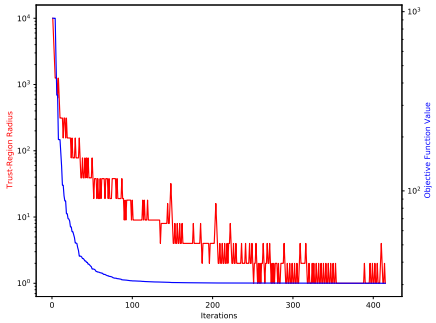
	Solution CPU Time(s)			Trust-Region
Number of Bangs	Relaxation	ODEs/Gradient	MIPs	Iterations
100	7999.89	9372.31	1.31	18
1000	5270.65	72173.45	8.70	128
10000	5637.85	720508.67	218.09	415

$$
\begin{aligned}
\underset{T,\,v}{\text{minimize}} \quad & \mathcal{J}(T(v), v) \\
\text{subject to} \quad & C(T, v) = 0 \quad \text{on } \Omega, \\
& \Omega = \bigcup_{i=1}^{N} \Omega_i, \\
& w(t) = \sum_{i=1}^{N} v_i \mathbb{1}_{\Omega_i \setminus \partial\Omega_i}(t) \\
, \quad & v_i \in \{0, 1\} \quad \forall i = 1...N, \\
& \Omega_i = [\tau_i, \tau_{i+1}] \quad \forall\, i = 1, ..., N, \\
& \tau_1 = 0, \quad \tau_{N+1} = t_{\text{final}}, \\
& \Omega = [0, t_{\text{final}}], \\
& T(0) = T_0.
\end{aligned}
\tag{10.1}
$$



(a) 100 bangs.

(b) 1000 bangs.



(c) 10000 bangs.

Fig. 2: Trust-region method objective function value and trust-region radius evolution for 100,1000, and 10000 binary variables. On the left axis (red) we plot the objective function value as a function of iteration count, and on the right axis (blue) we plot the trust-region radius as a function of iteration count.

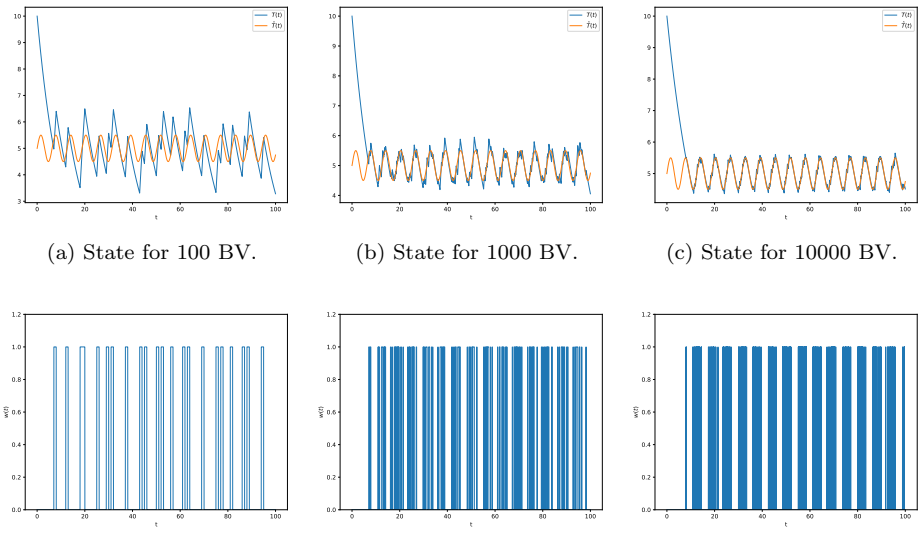
Table 4: The objective function value before (OFVB) at the rounded relaxation (start of trust-region), objective function value after (OFVA) at the termination of trust-region, and the percent-reduction in the objective function value from the start to end of the trust-region method for 100, 1000 and 10000 binary variables.

Number of Bangs	OFVB	OFVA	Percent Reduction (%)
100	337.93	37.93	88.77
1000	739.24	23.84	96.77
10000	918.76	30.50	96.68

Where $C : t \to \mathbb{R}^k$, $k \geq 1$. We define an adjoint variable, $\lambda : t \to \mathbb{R}^k$, and define the Lagrangian

$$\mathcal{L}(T, \lambda, v) = \mathcal{J} + \int_\Omega \lambda^T C dt. \tag{10.2}$$

From the Lagrangian, we derive the restricted state and adjoint equation, and the gradient, as seen in Section 2. We note that if there are additional constraints on the



(a) State for 100 BV. (b) State for 1000 BV. (c) State for 10000 BV.

(d) Control for 100 BV. (e) Control for 1000 BV. (f) Control for 10000 BV.

Fig. 3: Plots of the state $T(t)$, target $\hat{T}(t) = 5 + 0.5\sin(t)$, and bang control $w(t)$ for 100, 1000, and 10000 binary variables (BV).

binary variable, which is not considered in our formulation, these constraints can be ignored in the Lagrangian and instead enforced in the trust-region subproblem.

Derive the augmented IIM Schemes for the state and adjoint system. Next, the state and adjoint problem are discretized using the IIM approach. Typically, we use a standard numerical method (in this paper Euler), as a foundation to construct a "corrected" scheme with the IIM approach. One can view our approach as the standard scheme being applied at regular grid points, while the "corrected" scheme is applied at irregular grid points. If the scheme that is used at the regular grid points has been shown to be consistent, stable, and convergent, then the scheme at the irregular grid points consistency, stability, and convergence proofs need to be only slightly altered. Having a methods that are consistent, stable, and convergent for the state and adjoint equation are critical. Because without the correct state and adjoint computation, it is impossible to have a consistent gradient, let alone to solve the overarching optimal control problem with any numerical optimization approach.

Apply the trust-region method. In our work we first solve the relaxation of our optimal control problem, and then round the relaxation to produce a feasible starting guess to the binary optimal control problem. In our previous work, [13] we have observed that initializing our trust-region method with the rounded relaxation produced controls that yield lower objective function value than we when initialized our initial trust-region method with a random binary guess. Our trust-region methods computational effort is bottle-necked by only the state and adjoint computation, which one would have to perform for any reduced space optimal control problem. However, regardless of the amount of binary variables present in the problem we can solve the trust-region sub problem quickly.

11 Conclusions In this work, we introduced a formal Lagrangian approach for solving bang-bang optimal control problems. We introduced a target optimal control and wrote the optimality conditions, which led to solving the state and adjoint equation using IMM to ensure that we obtain a numerical solutions that are consistent, stable, and convergent. By solving the state and adjoint equation we are able to produce a gradient, and an objective function value that is associated with a fixed bang-bang control. We use the objective function value and gradient to execute our trust-region method. We showed numerical simulations that both demonstrated the importance of not applying standard numerical methods to the state equation, but also that our approach generates bang-bang controls that meets our goals. This work provides a blueprint that readers can use to as a foundation to solve their own bang-bang problems quickly and efficiently.

12 Acknowledgements This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes. This document has been approved to be released for widespread consumption according to order LLNL-JRNL-821389.

REFERENCES

[1] Ping Chen and Sardar MN Islam. *Optimal control models in finance*. Springer, 2005.
[2] U. Ledzewicz and H. Schattler. Optimal control for a bilinear model with recruiting agent in cancer chemotherapy. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, volume 3, pages 2762–2767 Vol.3, 2003.
[3] PDN Srinivasu and BSRV Prasad. Role of quantity of additional food to predators as a control in predator–prey systems with relevance to pest management and biological conservation. *Bulletin of mathematical biology*, 73(10):2249–2276, 2011.
[4] Shuo Wang and Heinz Schättler. Optimal control of a mathematical model for cancer chemotherapy under tumor heterogeneity. *Mathematical Biosciences & Engineering*, 13(6):1223, 2016.
[5] Felix Bestehorn, Christoph Hansknecht, Christian Kirches, and Paul Manns. A switching cost aware rounding method for relaxations of mixed-integer optimal control problems. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 7134–7139. IEEE, 2019.
[6] Marcel Antal, Claudia Pop, Tudor Cioara, Ionut Anghel, Ionut Tamas, and Ioan Salomie. Proactive day-ahead data center operation scheduling for energy efficiency: Solving a miocp using a multi-gene genetic algorithm. In *2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 527–534. IEEE, 2017.
[7] Alberto De Marchi. On the mixed-integer linear-quadratic optimal control with switching cost. *IEEE Control Systems Letters*, 3(4):990–995, 2019.
[8] Randall J Leveque and Zhilin Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis*, 31(4):1019–1044, 1994.
[9] Mark N Linnick and Hermann F Fasel. A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains. *Journal of Computational Physics*,

204(1):157–192, 2005.
[10] Long Lee and Randall J LeVeque. An immersed interface method for incompressible navier–stokes equations. *SIAM Journal on Scientific Computing*, 25(3):832–856, 2003.
[11] Zhilin Li and Ming-Chih Lai. The immersed interface method for the navier–stokes equations with singular forces. *Journal of Computational Physics*, 171(2):822–842, 2001.
[12] Walter Gautschi. *Numerical analysis*. Springer Science & Business Media, 1997.
[13] Ryan H Vogt, Sven Leyffer, and Todd Munson. A mixed-integer PDE-constrained optimization formulation for electromagnetic cloaking. 2020.

Acknowledgements This material was based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through Advanced Computing (SciDAC) program through the FASTMath Institute under Contract DE-AC02-06CH11357 at Argonne National Laboratory. This work was also supported by the U.S. Department of Energy through grant DE-FG02-05ER25694.