

A Computational Framework for the Numerical Solution of Optimal Control Problems Governed by Partial Differential Equations

Alexander M. Davies¹, Miriam E. Dennis², and Anil V. Rao³

Abstract—A computational framework for the solution of optimal control problems with time-dependent partial differential equations (PDEs) is presented. The optimal control problem is transformed from a continuous time and space optimal control problem to a sparse nonlinear programming problem through state parameterization with Lagrange polynomials and discrete controls defined at Legendre-Gauss-Radau (LGR) points. The standard LGR collocation method is coupled with a modified Radau method to produce a collocation point on the typically noncollocated boundary. The newly collocated endpoint allows for a representation of the state derivative and control on the originally noncollocated boundary such that Neumann boundary conditions may be satisfied. Finally, the method developed in this paper is demonstrated on a viscous Burgers' tracking problem and the results are compared to an existing solution.

I. INTRODUCTION

Direct and indirect methods exist for solving optimal control problems with partial differential equations numerically [1]–[7]. While direct methods for the solution of optimal control problems with ordinary differential equations (ODEs) have been well studied [3], [8]–[11], direct methods for the solution of optimal control problems with partial differential equations are less general and thereby less common in the optimal control literature. Moreover, significant challenges arise when the state is a function of both a temporal and spatial variable. Such challenges include difficulties in varying degrees of equation order, handling of boundary conditions, control location, and the computational and mathematical complexity of the additional dependence. In particular, the boundary conditions play an exceptional role in the spatial discretization and many times influence the choice of scheme and inhibit generality of a method. In addition, the location of the control (boundary or distributed) plays a large role in the choice of discretization and solution methodology.

Numerous methods exist to solve partial differential equations numerically, including spectral [12], finite element [13], the method of lines [3], domain decomposition [14], and multigrid methods [15]. Another approach for solving optimal control problems with partial differential equations is by orthogonal collocation. Direct orthogonal collocation methods for the solution of optimal control problems have largely

been applied to systems described by ODEs, and they have been sparingly applied to problems with PDE constraints. In many cases, direct orthogonal collocation methods present challenges when handling Neumann boundary conditions and boundary controls. The primary contribution of this work is a Legendre-Gauss-Radau (LGR) method [16] coupled with a modified Radau scheme [17] for the numerical solution of optimal control problems with time-dependent partial differential equations and Neumann boundary conditions.

The modified method allows for a representation of the derivative on both boundaries with the addition of control on the originally noncollocated boundary. The modified method as introduced by Eide, Hager, and Rao [17] was developed for the solution of optimal control problems with nonsmooth solutions; however, the modified method is used in this paper to gather a representation of the derivative on both boundaries and the control on the originally noncollocated boundary. In doing this, the solution of optimal control problems with time-dependent PDEs and Neumann boundary conditions may be obtained.

The remainder of the paper is presented as follows. The mathematical framework for the solution of optimal control problems with time-dependent PDEs and Neumann boundary conditions by LGR collocation is presented in Section II, and the viscous Burgers' tracking problem presented and solved by Büskens and Griesse [18] is solved by the framework in Section III. Optimal values of the cost are then compared with another existing solution [3].

II. MATHEMATICAL FRAMEWORK

A. LGR and Flipped LGR Collocation

The method presented here employs a direct orthogonal collocation method to transcribe a continuous-time and space optimal control problem into a sparse nonlinear programming problem via state parameterization with Lagrange polynomials and discrete controls defined at the LGR points. In this work, both LGR points and flipped LGR points are used as collocation points for the PDE constraints, boundary conditions, and initial conditions. The LGR points are calculated from the roots of the $P_{N-1}(\tau) + P_N(\tau)$ Legendre polynomial where N is the number of collocation points and are defined on the half-open interval from $[-1, 1)$ [19]. The flipped LGR points are the negative of the LGR points on the half-open interval from $(-1, 1]$. The LGR points can be used to exactly integrate a polynomial, $p(\tau)$, of degree $2N - 2$ by

$$\int_{-1}^1 p(\tau) d\tau = \sum_{i=1}^N w_i p(\tau_i), \quad (1)$$

¹Alexander M. Davies is a Ph.D. student with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611, USA alexanderdavies@ufl.edu

²Miriam E. Dennis is a Research Engineer with the Munitions Directorate, Air Force Research Laboratory, Eglin AFB, FL 32542, USA, miriam.dennis.1@us.af.mil

³Anil V. Rao is a professor with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611, USA, anilvr@ufl.edu

where w_i are the quadrature weights. It has been shown that both LGR and flipped LGR points may essentially be used interchangeably for problems with constraints defined by ordinary differential equations [20]. However, for problems with partial differential equation constraints, the noncollocated point (on the open boundary) becomes important to consider for accurate solutions.

B. Optimal Control with Partial Differential Equations

Without loss of generality, an optimal control problem in Lagrange form is presented as follows. Assume we seek to minimize the arbitrary cost functional, F , given by

$$F = \int_{t_0}^{t_f} \int_{x_0}^{x_f} \mathcal{L}(x, t, \mathbf{y}(x, t), \mathbf{u}(x, t)) dx dt, \quad (2)$$

subject to the parabolic PDE

$$\frac{\partial \mathbf{y}}{\partial t} + a \frac{\partial^2 \mathbf{y}}{\partial x^2} = \mathbf{f}(x, t, \mathbf{y}(x, t), \mathbf{u}(x, t)), \quad (3)$$

with arbitrary boundary and initial conditions

$$\mathbf{b}_1(x_0, t) = \mathbf{g}(x_0, t, \mathbf{y}(x_0, t), \mathbf{u}(x_0, t)), \quad (4)$$

$$\mathbf{b}_2(x_f, t) = \mathbf{h}(x_f, t, \mathbf{y}(x_f, t), \mathbf{u}(x_f, t)), \quad (5)$$

$$\mathbf{c}(x, t_0) = \mathbf{q}(x, t_0, \mathbf{y}(x, t_0), \mathbf{u}(x, t_0)), \quad (6)$$

where t and x are the temporal and spatial variables, respectively, $\mathbf{y}(x, t)$ and $\mathbf{u}(x, t)$ are the state and control, respectively, and a is a constant. For simplicity in the remaining discussion, only parabolic PDEs are addressed, but it should be noted that the methodology is similar for hyperbolic PDE constraints. An affine transformation is used to transform the temporal and spatial domains from $t_0 \leq t \leq t_f$ and $x_0 \leq x \leq x_f$ to $-1 \leq \tau \leq 1$ and $1 \leq \xi \leq 1$, respectively, where τ and ξ are the transformed independent variables given by

$$\tau = 2 \frac{t - t_0}{t_f - t_0} - 1, \quad \xi = 2 \frac{x - x_0}{x_f - x_0} - 1. \quad (7)$$

Using the transformations in (7), it is possible to rewrite the PDE in (3); however, it is useful to expand the PDE in (3) into two PDEs that are first-order in space. This may be done by constructing a new state variable, $\mathbf{v}(x, t)$, and rewriting (3) in an equivalent form as

$$\frac{\partial \mathbf{y}}{\partial t} + a \frac{\partial \mathbf{v}}{\partial x} = \mathbf{f}(x, t, \mathbf{y}(x, t), \mathbf{v}(x, t), \mathbf{u}(x, t)), \quad (8)$$

$$\frac{\partial \mathbf{y}}{\partial x} = \mathbf{v}(x, t). \quad (9)$$

The PDEs in (8)-(9) may then be rewritten using the transformation in (7) as

$$\frac{\partial x}{\partial \xi} \frac{\partial \mathbf{y}}{\partial \tau} + a \frac{\partial t}{\partial \tau} \frac{\partial \mathbf{v}}{\partial \xi} = \frac{\partial x}{\partial \xi} \frac{\partial t}{\partial \tau} \mathbf{f}(\xi, \tau, \mathbf{y}(\xi, \tau), \mathbf{v}(\xi, \tau), \mathbf{u}(\xi, \tau)), \quad (10)$$

$$\frac{\partial \mathbf{y}}{\partial \xi} = \frac{\partial x}{\partial \xi} \mathbf{v}(\xi, \tau), \quad (11)$$

where $\partial t / \partial \tau = (t_f - t_0) / 2$ from (7) and $\partial x / \partial \xi = (x_f - x_0) / 2$ from (7). Suppose that the domain, $\Omega = \{(\xi, \tau) \mid -1 \leq \xi \leq 1; -1 \leq \tau \leq 1\}$, may further be broken into J time

intervals and K space intervals. That is, the time domain, $T = \{\tau \mid -1 = \tau_0 < \tau_1 < \dots < \tau_{J-1} < \tau_J = 1\}$, is transformed from $T_j \in [\tau_{j-1}, \tau_j]$, ($j = 1, \dots, J$), to the new independent variable $r \in [-1, 1]$, and the space domain, $S = \{\xi \mid -1 = \xi_0 < \xi_1 < \dots < \xi_{K-1} < \xi_K = 1\}$, is transformed from $S_k \in [\xi_{k-1}, \xi_k]$, ($k = 1, \dots, K$), to the new independent variable $s \in [-1, 1]$. To perform the J transformations on the temporal variable and K transformations on the spatial variable, a similar transformation as in (7) is employed. The transformations are provided as

$$r = 2(\tau - \tau_{j-1}) / (\tau_j - \tau_{j-1}) - 1, \quad j = 1, \dots, J, \quad \tau \in [\tau_{j-1}, \tau_j], \quad (12)$$

$$s = 2(\xi - \xi_{k-1}) / (\xi_k - \xi_{k-1}) - 1, \quad k = 1, \dots, K, \quad \xi \in [\xi_{k-1}, \xi_k]. \quad (13)$$

The PDEs in (10)-(11) may again be rewritten as

$$\left(\frac{\partial \xi}{\partial s} \right)^{(k)} \frac{\partial x}{\partial \xi} \frac{\partial \mathbf{y}^{(j,k)}}{\partial r} + a \left(\frac{\partial \tau}{\partial r} \right)^{(j)} \frac{\partial t}{\partial \tau} \frac{\partial \mathbf{v}^{(j,k)}}{\partial s} = \quad (14)$$

$$\left(\frac{\partial \xi}{\partial s} \right)^{(k)} \left(\frac{\partial \tau}{\partial r} \right)^{(j)} \frac{\partial x}{\partial \xi} \frac{\partial t}{\partial \tau} \mathbf{f}(s, r, \mathbf{y}^{(j,k)}, \mathbf{v}^{(j,k)}, \mathbf{u}^{(j,k)}), \quad (15)$$

where the dependency of $\mathbf{y}^{(j,k)}$, $\mathbf{v}^{(j,k)}$, and $\mathbf{u}^{(j,k)}$ on the independent variables (s, r) have been excluded for compactness. Note, the values of $(\partial \xi / \partial s)^{(k)}$ and $(\partial \tau / \partial r)^{(j)}$ depend upon the bounds of the intervals $[\xi_{k-1}, \xi_k] \forall k = 1, \dots, K$, and $[\tau_{j-1}, \tau_j] \forall j = 1, \dots, J$, respectively, and are computed by

$$(\partial \xi / \partial s)^{(k)} = (\xi_k - \xi_{k-1}) / 2, \quad k = 1, \dots, K, \quad (16)$$

$$(\partial \tau / \partial r)^{(j)} = (\tau_j - \tau_{j-1}) / 2, \quad j = 1, \dots, J. \quad (17)$$

C. The Legendre-Gauss-Radau Collocation Method

The states in (14)-(15) may be approximated by a basis of Lagrange polynomials by

$$\mathbf{y}^{(j,k)}(s, r) \approx \mathbf{Y}^{(j,k)}(s, r) = \sum_{n=1}^{N_t^{(j)}+1} \sum_{m=1}^{N_x^{(k)}+1} \mathbf{Y}_{(n,m)}^{(j,k)} L_n^{(j)}(r) Q_m^{(k)}(s), \quad (18)$$

$$\mathbf{v}^{(j,k)}(s, r) \approx \mathbf{V}^{(j,k)}(s, r) = \sum_{n=1}^{N_t^{(j)}+1} \sum_{m=1}^{N_x^{(k)}+1} \mathbf{V}_{(n,m)}^{(j,k)} L_n^{(j)}(r) Q_m^{(k)}(s), \quad (19)$$

where $L_n^{(j)}(r)$ and $Q_m^{(k)}(s)$ are defined as

$$L_n^{(j)}(r) = \prod_{i=1, i \neq n}^{N_t^{(j)}+1} \frac{r - r_i^{(j)}}{r_n^{(j)} - r_i^{(j)}}, \quad Q_m^{(k)}(s) = \prod_{l=1, l \neq m}^{N_x^{(k)}+1} \frac{s - s_l^{(k)}}{s_m^{(k)} - s_l^{(k)}}. \quad (20)$$

In (18)-(20), $N_t^{(j)}$ and $N_x^{(k)}$ represent the number of LGR points in time (r) in interval j and in space (s) in interval k , respectively. Note, the number of collocation points in space in a particular space interval must remain consistent across all time intervals, and similarly, the number of collocation

points in time in a particular time interval must remain consistent across all space intervals. Taking the partial derivative of (18) with respect to r and s yields

$$\frac{\partial \mathbf{y}^{(j,k)}}{\partial r} \approx \frac{\partial \mathbf{Y}^{(j,k)}}{\partial r} = \sum_{n=1}^{N_t^{(j)}+1} \sum_{m=1}^{N_x^{(k)}+1} \mathbf{Y}_{(n,m)}^{(j,k)} \frac{\partial L_n^{(j)}(r)}{\partial r} Q_m^{(k)}(s), \quad (21)$$

$$\frac{\partial \mathbf{y}^{(j,k)}}{\partial s} \approx \frac{\partial \mathbf{Y}^{(j,k)}}{\partial s} = \sum_{n=1}^{N_t^{(j)}+1} \sum_{m=1}^{N_x^{(k)}+1} \mathbf{Y}_{(n,m)}^{(j,k)} L_n^{(j)}(r) \frac{\partial Q_m^{(k)}(s)}{\partial s}, \quad (22)$$

and taking the partial derivative of (19) with respect to s yields

$$\frac{\partial \mathbf{v}^{(j,k)}}{\partial s} \approx \frac{\partial \mathbf{V}^{(j,k)}}{\partial s} = \sum_{n=1}^{N_t^{(j)}+1} \sum_{m=1}^{N_x^{(k)}+1} \mathbf{V}_{(n,m)}^{(j,k)} L_n^{(j)}(r) \frac{\partial Q_m^{(k)}(s)}{\partial s}. \quad (23)$$

These approximations may now be substituted into (14)-(15) to define the constraint equations on a finite mesh. Prior to substitution, to simplify the constraint equations in (14)-(15), the state approximations $\mathbf{Y}_{(n,m)}^{(j,k)}$ and $\mathbf{V}_{(n,m)}^{(j,k)}$ can be organized into state vectors with specified indices. For simplicity, in both space and time, the number of collocation points are kept consistent in each interval. In other words, the total number of points in time is given by $N_t J + 1$, and the total number of points in space is given by $N_x K + 1$. As a result, the total number of points in the entire domain is given by $(N_t J + 1) \times (N_x K + 1)$. In this work, indexing was completed by “stacking” in space. Mathematically, this is shown by

$$\bar{\mathbf{Y}} = \begin{bmatrix} \mathbf{Y}_{(:,1)}^{(:,1)} & \dots & \mathbf{Y}_{(:,N_x+1)}^{(:,K)} \end{bmatrix}^T \in \mathbb{R}^{(N_t J + 1)(N_x K + 1) \times 1}. \quad (24)$$

That is, all time components at a particular space point are stacked into the vector, and the process is repeated at each additional space point until the end of the domain. This is similarly done for $\bar{\mathbf{V}}$. A visual depiction of the mesh for the problem presented is given in Fig. 1. In time, the flipped

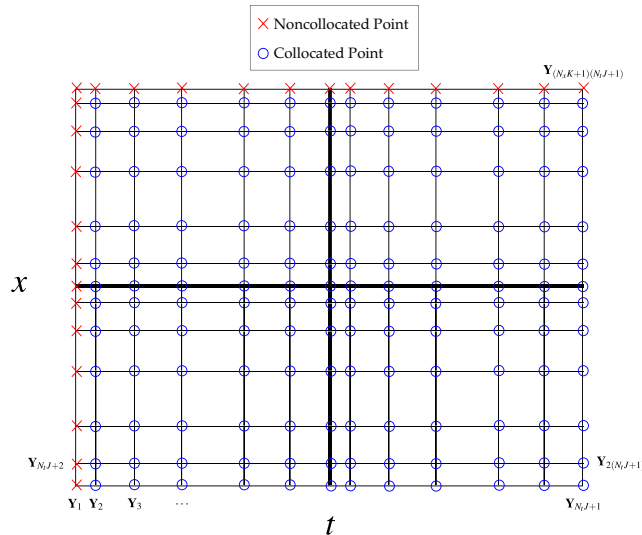


Fig. 1. A visual depiction of the mesh with standard LGR points in space and flipped-LGR points in time with $N_t = 6$, $J = 2$, $N_x = 6$, and $K = 2$. The mesh demonstrates the organization of the points in the state vector.

LGR points are used, and in space, the standard LGR points are used. As points at the interval interfaces exist in both of the connecting intervals, they are treated as the same NLP variable in the optimization (e.g. $\mathbf{Y}_{(:,N_x+1)}^{(:,1)} = \mathbf{Y}_{(:,1)}^{(:,2)}$ at the first mesh interval intersection). Note, (14)-(15) hold for the interior of the domain and the final time boundary. The boundary at $t = t_0$, $x = x_0$, and $x = x_f$ are constrained by an initial condition and boundary conditions, respectively. As a result, these points will be treated separately from the interior.

1) *Collocating the Interior*: The interior of the domain consists of all of the collocated points demonstrated in Fig. 1 with the exception of the boundary at $x = x_0$. For simplicity, the following simplification of the constraint equations is shown for the global case. The multi-interval case is done identically; however, the indexing of the NLP variables makes the explanation unnecessarily complex. In *one mesh interval*, the sums in (21)-(23), can be organized into matrix form to match the state vector organization in (24). As the Lagrange polynomial at a specific collocation point is unity at the collocation point and zero elsewhere, $L_n^{(j)}(r) \in \mathbb{R}^{N_t \times N_t + 1}$ can be expressed as a matrix with unity on the main diagonal and zeros elsewhere. The matrix will be denoted by $\mathbf{L}$. The derivative term, $\partial L_n^{(j)}(r)/\partial r$ can be represented by a dense $N_t \times N_t + 1$ matrix where the value of the i^{th} row and n^{th} column is the value of the derivative of the n^{th} time Lagrange polynomial evaluated at the i^{th} time collocation point. The matrix representation of $\partial L_n^{(j)}(r)/\partial r$ will be denoted as $\mathbf{L}_r \in \mathbb{R}^{N_t \times N_t + 1}$. If the mesh is *only* comprised of one mesh interval, the double sum in (21) can be replaced by the matrix $\mathbf{D}_t$, which has the form

$$\mathbf{D}_t = \begin{bmatrix} \mathbf{L}_r Q_1(s_2) & \mathbf{L}_r Q_2(s_2) & \dots & \mathbf{L}_r Q_{N_x+1}(s_2) \\ \mathbf{L}_r Q_1(s_3) & \mathbf{L}_r Q_2(s_3) & \dots & \mathbf{L}_r Q_{N_x+1}(s_3) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{L}_r Q_1(s_{N_x}) & \mathbf{L}_r Q_2(s_{N_x}) & \dots & \mathbf{L}_r Q_{N_x+1}(s_{N_x}) \end{bmatrix}. \quad (25)$$

Note, due to the isolation property of the Lagrange polynomials, the only non-zero terms in $\mathbf{D}_t$ in (25) occur when the m^{th} spatial polynomial is evaluated at the m^{th} collocation point. These non-zero terms reduce to the matrix $\mathbf{L}_r$.

As the flipped LGR points are used in time, the matrix $\mathbf{L} \in \mathbb{R}^{N_t \times N_t + 1}$ with ones on the main diagonal and zeros elsewhere must be rearranged to account for the noncollocated point. This is done by shifting the column of zeros corresponding to the noncollocated point such that

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots \\ 0 & 0 & \ddots & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1 & 0 & \dots \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \ddots \end{bmatrix}. \quad (26)$$

A matrix, $\mathbf{D}_x$, can be constructed that replaces the double sum in (22). This is given by

$$\mathbf{D}_x = \begin{bmatrix} \mathbf{L} \frac{\partial Q_1(s_2)}{\partial s} & \mathbf{L} \frac{\partial Q_2(s_2)}{\partial s} & \dots & \mathbf{L} \frac{\partial Q_{N_x+1}(s_2)}{\partial s} \\ \mathbf{L} \frac{\partial Q_1(s_3)}{\partial s} & \mathbf{L} \frac{\partial Q_2(s_3)}{\partial s} & \dots & \mathbf{L} \frac{\partial Q_{N_x+1}(s_3)}{\partial s} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{L} \frac{\partial Q_1(s_{N_x})}{\partial s} & \mathbf{L} \frac{\partial Q_2(s_{N_x})}{\partial s} & \dots & \mathbf{L} \frac{\partial Q_{N_x+1}(s_{N_x})}{\partial s} \end{bmatrix}. \quad (27)$$

Sparsity patterns for $\mathbf{D}_t$ and $\mathbf{D}_x$ for the one interval (global Radau) case are given in Fig. 2 and Fig. 3, respectively. Hav-

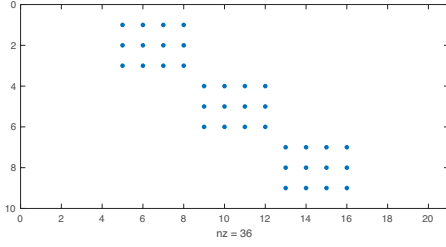


Fig. 2. The sparsity pattern for $\mathbf{D}_t$ with $J = 1$, $K = 1$, $N_t = 3$, and $N_x = 4$.

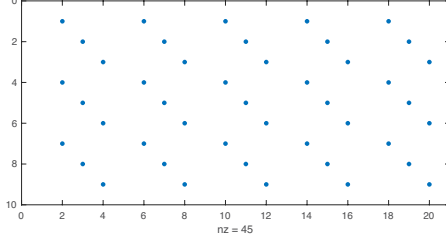


Fig. 3. The sparsity pattern for $\mathbf{D}_x$ with $J = 1$, $K = 1$, $N_t = 3$, and $N_x = 4$.

ing performed the simplification of (21)-(23), the constraint equations in (14)-(15) may be reduced to

$$\frac{\partial \xi}{\partial s} \frac{\partial x}{\partial \xi} \mathbf{D}_t \bar{\mathbf{Y}} + \frac{\partial \tau}{\partial r} \frac{\partial t}{\partial \tau} \mathbf{D}_x \bar{\mathbf{V}} = \frac{\partial \xi}{\partial s} \frac{\partial \tau}{\partial r} \frac{\partial x}{\partial \xi} \frac{\partial t}{\partial \tau} \mathbf{f}(s, r, \mathbf{Y}_{\text{col}}, \mathbf{V}_{\text{col}}, \mathbf{u}), \quad (28)$$

$$\mathbf{D}_x \bar{\mathbf{Y}} = \frac{\partial \xi}{\partial s} \frac{\partial x}{\partial \xi} \mathbf{V}_{\text{col}}, \quad (29)$$

where $\bar{\mathbf{Y}}$ and $\bar{\mathbf{V}}$ are the state vectors and $\mathbf{Y}_{\text{col}}$ and $\mathbf{V}_{\text{col}}$ indicate the portion of the state vector corresponding to the collocated points. For the multi-interval case, the indexing becomes more complicated, as state variables at endpoints in one interval are the same as those at the start of the following interval. Sparsity patterns for $\mathbf{D}_t$ and $\mathbf{D}_x$ for a multi-interval case are given in Fig. 4 and Fig. 5 for reference.

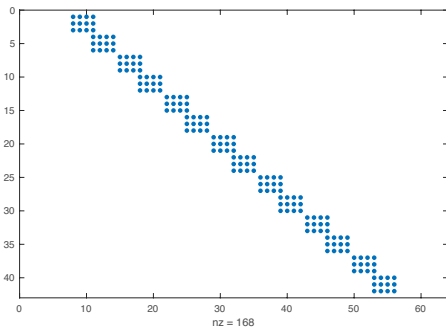


Fig. 4. The sparsity pattern for $\mathbf{D}_t$ with $J = 2$, $K = 2$, $N_t = 3$, and $N_x = 4$.

2) *Handling Initial Conditions:* Typically, the initial condition is a fixed value of the state at each spatial point. Thus, constraining the initial condition is rather straightforward. If a fixed-state initial condition exists, it is advantageous to use the flipped LGR points for the time discretization, as it is not imperative to have collocated points at the

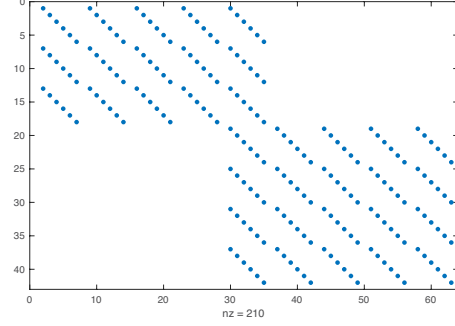


Fig. 5. The sparsity pattern for $\mathbf{D}_x$ with $J = 2$, $K = 2$, $N_t = 3$, and $N_x = 4$.

initial time boundary. The use of the flipped LGR points allows for the final time boundary to be constrained by the dynamics. The only remaining dependence is the connection between the state variable, $\mathbf{y}(t_0, x)$, and its spatial derivative, $\mathbf{v}(t_0, x)$; however, this is easily done by applying (15) at the intermediate points between (x_0, x_f) and constraining the endpoints via the boundary conditions.

3) *Handling Boundary Conditions:* Whereas initial conditions are typically fixed values of the state, a vast array of boundary conditions exist that are dependent on the type of equation and problem being solved. The most problematic set of boundary conditions are those that depend upon the derivative of the state. The LGR method only has a defined representation of the derivative on one of the two boundaries in a particular interval. Thus, for problems when derivatives on both ends are required, the LGR scheme is incapable of satisfying both boundary conditions without modification. To properly modify the LGR scheme to satisfy boundary conditions of this type, a modified LGR scheme [17] is employed. The modified scheme adds an additional collocation point at the endpoint of an interval and an additional control variable to the decision vector. On the first boundary ($x = x_0$), there exists representations for derivatives of the state when the standard LGR points are used. This is given by

$$\mathbf{D}_{x_0} = \left[\mathbf{L} \frac{\partial Q_1(s_1)}{\partial s} \quad \mathbf{L} \frac{\partial Q_2(s_1)}{\partial s} \quad \dots \quad \mathbf{L} \frac{\partial Q_{N_x+1}(s_1)}{\partial s} \right]. \quad (30)$$

On the second boundary ($x = x_f$), the modified equations must be used to gather a representation for the derivative of the state. The equations for the modified scheme are given by

$$\mathbf{D}_{x_f} = \left[\mathbf{L} \frac{\partial Q_1(s_{N_x+1})}{\partial s} \quad \mathbf{L} \frac{\partial Q_2(s_{N_x+1})}{\partial s} \quad \dots \quad \mathbf{L} \frac{\partial Q_{N_x+1}(s_{N_x+1})}{\partial s} \right]. \quad (31)$$

When one space interval is used, the equations in (30)-(31), along with the added control variables at the second boundary, are sufficient to fully define the boundary conditions and consequently the problem. However, in the multi-interval formulation, an additional set of constraints is required. Assume the boundary conditions on the first and second boundary take on the form

$$\partial \mathbf{y}(x_0, t) / \partial x = u_1(t), \quad \partial \mathbf{y}(x_f, t) / \partial x = u_2(t). \quad (32)$$

The lack of complete definition stems from the mesh interval intersection at the end of the first interval. Examining the mesh in Fig. 1, the collocation points at the mesh interval

intersection compute the derivative of the state via the Lagrange polynomials in the second interval. The “forward derivative” references the values of the state at the final boundary to compute its approximation for the derivative. Thus, the information from the second boundary condition may be propagated through the entirety of the domain. However, though the derivative is continuous as defined by the set of Lagrange polynomials, the information from the first boundary condition is not used when computing the value of the derivative at the intersection of the first mesh interval. To connect the first mesh interval accurately to the rest of the solution, the derivative of the state at the mesh interval intersection that references the points in the first interval is additionally required. To do this, the modified constraints in (31) are employed at the intersection of the first mesh interval. The addition of the modified constraints without control variables is only required in the first space interval such that the first boundary condition can be satisfied through an accurate computation of the state derivative at the first mesh interval intersection.

4) *Search Space Modification:* When using the modified method, control variables are to be added to properly modify the search space to arrive at the correct solution. However, if the dynamics equations are not a function of the control, this is not feasible. With the addition of more constraints, the number of degrees of freedom in the problem have been reduced, and as a result, the search space has been improperly reduced. To re-widen the search space, “slack” must be given to the modified constraints. That is, instead of setting the modified constraints equal to zero, the modified constraints are redefined as inequality path constraints between some tolerance $\pm\delta$.

Finding the best value of δ for a specific mesh is non-trivial, and it requires some degree of trial and error depending on the problem and number of collocation points used. If δ is made too large, the search space becomes too large, and as a result, the optimizer may converge to solutions that do not satisfy the parameters of the original problem. However, if δ is made too small, the search space becomes too small, and the optimizer may not converge to a solution. In general, the best procedure for determining a suitable value of δ (say δ^*) is to first define a value of $\delta < \delta^*$. If the optimizer does not converge, gradually increase the value of δ until the optimizer converges to the optimal solution.

The need for additional constraints and “slack” highlights an important feature of Gaussian quadrature and the modified Radau method. In theory, the “forward” and “backward” representations of the derivative of the state at the first mesh interval intersection should be close to (if not) identical; however, when few numbers of collocation points are used, the approximations of the derivative using two distinct sets of points becomes more disparate. As a result, numerical differences in the derivative approximations can present the need for “slack” in the constraint equations. As more collocation points are added, the approximation of the derivative in both intervals improves, and consequently, less “slack” is needed. It was found that the influence of the control on the

first boundary is unable to propagate through to the interior of the domain without a computation of an accurate state derivative that satisfies both the boundary condition and the dynamics at the mesh interval intersection.

III. EXAMPLE

For this work, the viscous Burgers’ tracking problem first presented by Büskens and Griese [18] and resolved by Betts [3] is presented. The problem is to minimize

$$F = \frac{1}{2} \int_0^1 \int_0^1 [y(x,t) - 0.035]^2 dx dt + \frac{\sigma}{2} \int_0^1 [u_1^2(t) + u_2^2(t)] dt, \quad (33)$$

subject to the PDE

$$\frac{\partial y(x,t)}{\partial t} = v \frac{\partial^2 y(x,t)}{\partial x^2} - y(x,t) \frac{\partial y(x,t)}{\partial x}, \quad (34)$$

with the initial and boundary conditions

$$y(x,0) = x^2(1-x)^2, \quad (35)$$

$$\frac{\partial y(0,t)}{\partial x} = u_1(t), \quad \frac{\partial y(1,t)}{\partial x} = u_2(t). \quad (36)$$

The control bounds for $i = 1, 2$ are given by $u_{min} \leq u_i(t) \leq u_{max}$. The problem is defined on the domain $\Omega = \{(x,t) \mid 0 \leq x \leq 1; 0 \leq t \leq 1\}$, and the following parameters complete the definition: $\sigma = 0.01$, $v = 0.1$, $u_{max} = 0.015$, $u_{min} = -0.015$.

The problem is solved using the presented framework on two different meshes. The first mesh contains a relatively small number of collocation points per interval while the second mesh contains a relatively large number of collocation points per interval. The computed solutions for the optimal state, state derivative, and two control inputs are provided in Figs. 6, 7, and 8, respectively, and the solution details are provided in Table I. Note that using a larger number

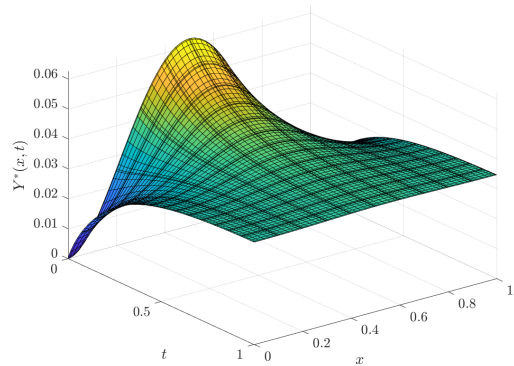


Fig. 6. The optimal state $Y^*(x,t)$ with $J = 14$, $K = 10$, $N_t = 6$, and $N_x = 7$.

TABLE I
SOLUTION DETAILS

	N_t	N_x	J	K	δ^*	F^*
Mesh 1	6	7	14	10	2×10^{-6}	2.896463×10^{-5}
Mesh 2	6	15	14	5	1×10^{-12}	2.896938×10^{-5}
Betts [3]	-	-	-	-	-	2.896682×10^{-5}

of collocation points per interval results in a significant

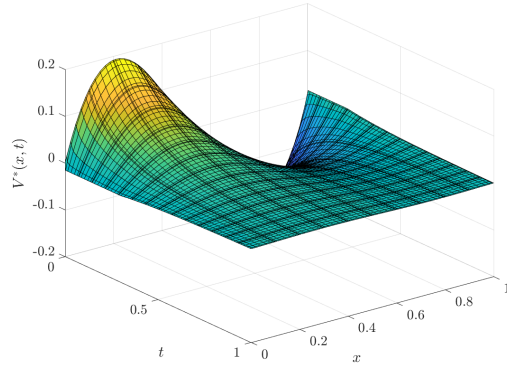


Fig. 7. The optimal state derivative $V^*(x, t)$ with $J = 14$, $K = 10$, $N_t = 6$, and $N_x = 7$.

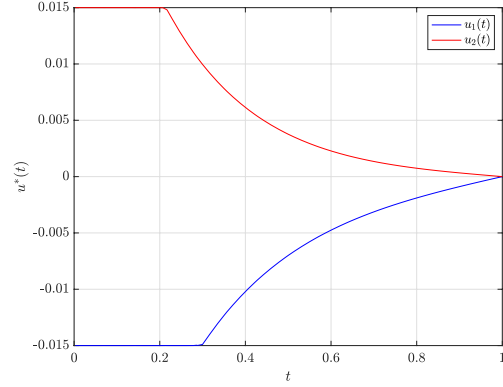


Fig. 8. The optimal controls $u_1(t)$ and $u_2(t)$ with $J = 14$, $K = 10$, $N_t = 6$, and $N_x = 7$.

reduction of the value of δ^* . The solutions found on Mesh 1 and 2 were solved with the open source NLP solver IPOPT [21] using an NLP tolerance of 10^{-12} . The solution on Mesh 1 resulted in an overall NLP error of 5×10^{-13} . The solution on Mesh 2 resulted in an overall NLP error of 9×10^{-13} . The solutions closely match those given by Betts [3].

IV. CONCLUSIONS

A computational framework for the solution of optimal control problems with time-dependent partial differential equations is presented. The standard LGR method is used in association with the modified method presented by Eide, Hager, and Rao [17] to solve optimal control problems with Neumann boundary conditions and boundary controls. The framework is then employed on a viscous Burgers' tracking problem to demonstrate the method in practice. The computed solution matches closely with an existing solution in the literature.

ACKNOWLEDGMENTS

The authors gratefully acknowledge support for this research from the U.S. National Science Foundation under grant CMMI-2031213, the U.S. Office of Naval Research under grant N00014-22-1-2397, and from the U.S. Air Force Research Laboratory under grant FA8651-23-1-0014.

DISCLAIMER

Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of sponsoring agencies.

REFERENCES

- [1] H. Maurer and H. D. Mittelmann, "Optimization techniques for solving elliptic control problems with control and state constraints: Part 1. boundary control," *Computational Optimization and Applications*, vol. 16, pp. 29–55, 2000.
- [2] L. Petzold, J. B. Rosen, P. E. Gill, L. O. Jay, and K. Park, "Numerical optimal control of parabolic pdes using dasopt," in *Large-Scale Optimization with Applications: Part II: Optimal Design and Control*, Springer, 1997, pp. 271–299.
- [3] J. Betts, *Practical Methods for Optimal Control Using Nonlinear Programming, Third Edition* (Advances in Design and Control). Society for Industrial and Applied Mathematics, 2020.
- [4] J. L. Lions, *Optimal control of systems governed by partial differential equations*. Springer, 1971, vol. 170.
- [5] E. Casas, "Optimal control of pde theory and numerical analysis," Ph.D. dissertation, Optimization and Control, 2006.
- [6] A. Manzoni, A. Quarteroni, and S. Salsa, *Optimal control of partial differential equations*. Springer, 2021.
- [7] F. Tröltzsch, *Optimal control of partial differential equations: theory, methods, and applications*. American Mathematical Soc., 2010, vol. 112.
- [8] A. V. Rao, "A survey of numerical methods for optimal control," *Advances in the astronautical Sciences*, vol. 135, no. 1, pp. 497–528, 2009.
- [9] O. Von Stryk, *Numerical solution of optimal control problems by direct collocation*. Springer, 1993.
- [10] D. Garg, M. Patterson, W. Hager, A. Rao, D. R. Benson, and G. T. Huntington, "An overview of three pseudospectral methods for the numerical solution of optimal control problems," 2017.
- [11] J. T. Betts and W. P. Huffman, "Exploiting sparsity in the direct transcription method for optimal control," *Computational Optimization and Applications*, vol. 14, no. 2, pp. 179–201, 1999.
- [12] Y. Chen, N. Yi, and W. Liu, "A legendre–galerkin spectral method for optimal control problems governed by elliptic equations," *SIAM Journal on Numerical Analysis*, vol. 46, no. 5, pp. 2254–2275, 2008.
- [13] R. Becker, H. Kapp, and R. Rannacher, "Adaptive finite element methods for optimal control of partial differential equations: Basic concept," *SIAM Journal on Control and Optimization*, vol. 39, no. 1, pp. 113–132, 2000.
- [14] J.-D. Benamou and B. Desprès, "A domain decomposition method for the helmholtz equation and related optimal control problems," *Journal of Computational Physics*, vol. 136, no. 1, pp. 68–82, 1997.
- [15] A. Borzi, "Multigrid methods for parabolic distributed optimal control problems," *Journal of Computational and Applied Mathematics*, vol. 157, no. 2, pp. 365–382, 2003.
- [16] D. Garg, M. A. Patterson, C. Francolin, *et al.*, "Direct trajectory optimization and costate estimation of finite-horizon and infinite-horizon optimal control problems using a radau pseudospectral method," *Computational Optimization and Applications*, vol. 49, pp. 335–358, 2011.
- [17] J. D. Eide, W. W. Hager, and A. V. Rao, "Modified legendre–gauss–radau collocation method for optimal control problems with nonsmooth solutions," *Journal of Optimization Theory and Applications*, pp. 1–34, 2021.
- [18] C. Büskens and R. Griesse, "Parametric sensitivity analysis of perturbed pde optimal control problems with state and control constraints," *Journal of Optimization Theory and Applications*, vol. 131, pp. 17–35, 2006.
- [19] M. Abramowitz, I. A. Stegun, and R. H. Romer, *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, 1988.
- [20] D. Garg, M. Patterson, W. W. Hager, A. V. Rao, D. A. Benson, and G. T. Huntington, "A unified framework for the numerical solution of optimal control problems using pseudospectral methods," *Automatica*, vol. 46, no. 11, pp. 1843–1851, 2010.
- [21] L. Biegler and V. Zavala, "Large-scale nonlinear programming using ipopt: An integrating framework for enterprise-wide dynamic optimization," *Computers Chemical Engineering*, vol. 33, no. 3, pp. 575–582, 2009.