An Integral Penalty-Barrier Direct Transcription Method for Optimal Control

Martin P. Neuenhofen[1] and Eric C. Kerrigan[2]

Abstract— Some direct transcription methods can fail to converge, e.g. when there are singular arcs. We recently introduced a convergent direct transcription method for optimal control problems, called the penalty-barrier finite element method (PBF). PBF converges under very weak assumptions on the problem instance. PBF avoids the ringing between collocation points, for example, by avoiding collocation entirely. Instead, equality path constraint residuals are forced to zero everywhere by an integral quadratic penalty term. We highlight conceptual differences between collocation- and penalty-type direct transcription methods. Theoretical convergence results for both types of methods are reviewed and compared. Formulas for implementing PBF are presented, with details on the formulation as a nonlinear program (NLP), sparsity and solution. Numerical experiments compare PBF against several collocation methods with regard to robustness, accuracy, sparsity and computational cost. We show that the computational cost, sparsity and construction of the NLP functions are roughly the same as for orthogonal collocation methods of the same degree and mesh. As an advantage, PBF converges in cases where collocation methods fail. PBF also allows one to trade off computational cost, optimality and violation of differential and other equality equations against each other.

I. MOTIVATION

Direct transcription is a popular method for the numerical solution of optimal control, estimation and system identification problems. Many direct transcription methods are based on collocation methods [1]. It is known that the latter can struggle with singular arcs and high-index differential algebraic equalities (DAEs) [1, Chap. 2 & 4.14]; the former arising in the example:

$$\min_{y,u} \quad J = \int_0^{\frac{\pi}{2}} \left(y(t)^2 + \cos(t)u(t) \right) \mathrm{d}t, \qquad (X1)$$

$$\text{s.t.} \quad y(0) = 0, \ \ \dot{y}(t) = \frac{1}{2}y(t)^2 + u(t), \ \ -1.5 \le u(t) \le 1\,,$$

which has the optimal solution $u^\star(t) = \frac{1}{2} - \frac{1.5}{(\cos(t)-2)^2}$.

Quadratic integral penalty methods [2]–[4] are an alternative to collocation methods, where the squared path
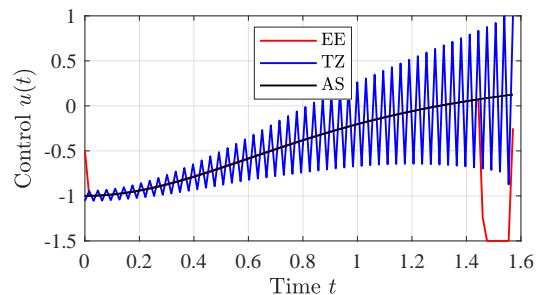

Fig. 1. Solutions to (X1): Explicit Euler (EE), Trapezoidal (TZ), analytic solution (AS).
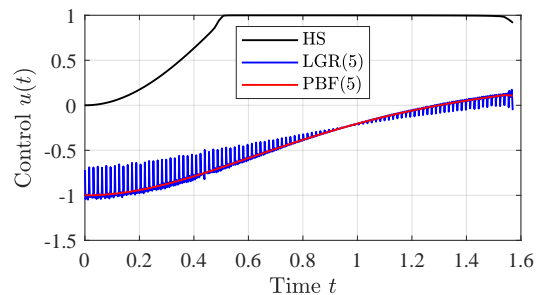

Fig. 2. Solutions to (X1): Hermite-Simpson (HS), Gauss-Legendre-Radau of degree 5 (LGR5), penalty-barrier finite element method of degree 5 (PBF5).

equality constraint residual is integrated and added as a penalty into the objective. In [4] we present such a method, called the penalty-barrier finite element method (PBF), with a rigorous proof of convergence under very mild and easily verifiable assumptions, including convergence for singular arcs and high-index DAEs. The assumptions are discussed in Section III-C.2 below.

Convergence proofs with enforceable assumptions are of high practical relevance. If convergence can only be guaranteed under unverifiable or unenforceable assumptions, then a practitioner cannot determine whether features in numerical solutions are due to optimality or numerical failure.

Figures 1–2 illustrate this dilemma by showing numerical solutions to (X1). They show solutions for the explicit Euler method (EE), trapezoidal method (TZ), Hermite-Simpson collocation (HS), Legendre-Gauss-Radau collocation of degree 5 (LGR5), and PBF of degree 5 (PBF5), each using 100 mesh intervals of equal length. PBF5 converges, while EE diverges for the last section of the time interval. The box constraints have been imposed

[1]Martin P. Neuenhofen is with the Deparment of Electrical & Electronic Engineering, Imperial College London, SW7 2AZ London, UK, e-mail: m.neuenhofen19@imperial.ac.uk www.MartinNeuenhofen.de

[2]Eric C. Kerrigan is with the Deparment of Electrical & Electronic Engineering and Department of Aeronautics, Imperial College London, SW7 2AZ London, UK, e-mail: e.kerrigan@imperial.ac.uk www.imperial.ac.uk/people/e.kerrigan

only to avoid unboundedness of the NLP resulting from the discretization. The solutions from HS and LGR2 (the latter of which was not plotted) also diverge if box constraints are removed. For LGR5 we see the best result among the non-penalty-type methods, however the solution suffers from ringing. All NLPs have been solved until the unscaled KKT residual ∞-norm of $< 10^{-10}$ stagnated.

A. Contributions

In this paper we summarise the presentation in [4] and extend it with practical aspects and new examples.

First, we generalize the problem statement. We extend the objective with a Mayer term and incorporate assumptions on the problem such that the convergence analysis from [4] applies. We extend the box-constraints to both the states and controls. We explain how these constraints can be posed into the NLP via a projection matrix, and where this matrix arises in the optimality conditions.

We then draw a connection between the first-order optimality conditions between the PBF discretization and the first-order optimality conditions of a regularized primal-dual problem. In this context, we visualize and compare sparsity patterns of the derivatives in PBF with the LGR direct collocation method.

We cover further practical aspects: We illustrate the stability condition for quadrature, which is a building block when constructing a convergent implementation of PBF. In the numerical experiments, we demonstrate how PBF minimizes a bias between optimality and feasibility with respect to the choice of the penalty parameter.

B. Outline

Section II reviews the problem statement and notation. Section III compares collocation-type methods with integral penalty-type methods. We highlight key differences in the numerical approaches and review available convergence results. Section IV shows how PBF can be implemented, with emphasis on the formulation of the NLP and the sparsity of derivative matrices. Section V presents numerical experiments, where we compare several collocation methods with PBF and focus on numerical cost, sparsity and (order of) convergence.

II. Numerical Optimal Control

A. Problem Statement

A common reference problem format for optimal control is the Bolza form [5], which can cover most optimal control, estimation and system identification problem instances, including multi-phase problems [6]. Without loss of generality, the initial and end time can be fixed [1]. Using slack variables, constraints can be separated into nonlinear equality constraints and box inequality con-

straints, resulting in the form

$$\min_{x:=(y,u)\in\mathcal{X}} \quad W\big(y(t_0), y(t_F)\big) + \int_\Omega w\big(y(t), u(t), t\big)\,\mathrm{d}t$$
(OCPa)

$$\text{s.t.} \quad 0 = c\big(\dot{y}(t), y(t), u(t), t\big) \quad \text{f.a.e. } t \in \Omega\,,$$
(OCPb)

$$0 = g\big(y(t_0), y(t_F)\big)\,, \quad \text{(OCPc)}$$

$$x_{\mathsf{L}}(t) \le x(t) \le x_{\mathsf{U}}(t) \quad \text{f.a.e. } t \in \Omega\,,$$
(OCPd)

where the constraints have to be satisfied for almost every $t \in \Omega := [t_0, t_F] \subset \mathbb{R}$. Here (OCPd) is optional, or can be specified as only upper/lower bounds or only over subintervals of $\Omega := [t_0, t_F]$ for each component of x.

The trajectory $x := (y, u)$ consists of continuous functions $y : \Omega \to \mathbb{R}^{n_y}$ and (potentially) discontinuous functions $u : \Omega \to \mathbb{R}^{n_u}$. Many works [5], [7], [8] separate (OCPb) into ODEs and algebraic constraints of the form

$$\dot{y}(t) = f_1\big(y(t), u(t), t\big)\,, \quad \text{(OCPb1)}$$
$$0 = f_2\big(y(t), u(t), t\big)\,, \quad \text{(OCPb2)}$$

with functions f_1, f_2, describing the dynamic constraints. This is necessary if the ODE is solved with a different scheme or when the convergence analysis poses different criteria on f_1 and f_2.

In either case, it can be considered an advantage of PBF to be capable of solving (OCPb) directly, i.e. without algebraic transformation into (OCPb1)–(OCPb2), because this allows avoiding divisions by zero and phrasing equations such that their derivatives are smoother, which improves the efficiency of an NLP solver.

B. Notion of Convergence

Numerical methods must minimize

$$J(x) := W\big(y(t_0), y(t_F)\big) + \int_\Omega w\big(y(t), u(t), t\big)\,\mathrm{d}t$$

subject to forcing the total constraint violation

$$r(x) := \int_\Omega \big\|c\big(\dot{y}(t), y(t), u(t), t\big)\big\|_2^2\,\mathrm{d}t + \big\|g\big(y(t_0), y(t_F)\big)\big\|_2^2$$

to zero while obeying (OCPd). To keep their computation time finite, all methods search for an approximation to the optimal x in a finite-dimensional space $\mathcal{X}_{h,p}$, which is a subspace of the solution space $\mathcal{X}$ of continuous functions y and discontinuous functions u; here h is the mesh step size and p is the degree of a method, in the case where the method is parametric. Formal details on the solution spaces are given in [4].

When solving optimization problems, implying (OCP), there are two measures to take into account: feasibility and optimality. One wants to find a weak numerical solution $x_h \in \mathcal{X}_{h,p}$ that solves (OCP) in a *tolerance-accurate* sense. This means, the *optimality gap*

$$g_{\text{opt}} := \max\{0, J(x_h) - J(x^\star)\}\,, \quad (2)$$

where $x^\star$ is a global minimizer of (OCP), and *feasibility residual*

$$r_{\text{feas}} := r(x_h) \tag{3}$$

converge to zero as the mesh step size h decreases. Order-of-convergence results that specify the rate at which g_{opt} and r_{feas} approach zero in terms of h can be derived; see [4].

III. COLLOCATION VS INTEGRAL PENALTY METHODS

Many state-of-the-art methods for the solution of (OCP) are collocation-type methods. We describe their geometric approach below and discuss their principle issues. We then introduce integral penalty methods and explain how they can overcome the convergence problems that collocation methods can exhibit. The section ends with a direct comparison of theoretical convergence results.

A. Collocation-type Methods

These methods determine solutions to y by solving (OCPb1) at a finite number of time instances. The trajectory u must then be found such that (y, u) satisfies (OCPb2). Examples of such methods (see [9] for a survey) are: single- and multiple-shooting methods [10] based on certain classes of Runge-Kutta or other collocation methods for solving the differential equations; multi-step methods, in particular backward-differentiation formula methods; and direct collocation methods, an important class being orthogonal direct collocation methods.

All these methods replace the uncountable set of path constraints (OCPb1)–(OCPb2) by a finite set of constraints, namely

$$\dot{y}(t) = f_1\left(y(t), u(t), t\right) \qquad \forall t \in \tilde{\mathbb{T}}_{h,p},$$
$$0 = f_2\left(y(t), u(t), t\right) \qquad \forall t \in \mathbb{T}_{h,p},$$

where $\tilde{\mathbb{T}}_{h,p}, \mathbb{T}_{h,p}$ are finite subsets of the interval $[t_0, t_F]$. In pure collocation methods $\tilde{\mathbb{T}}_{h,p} = \mathbb{T}_{h,p}$, and these points are called collocation points. In some Runge-Kutta and linear multi-step methods, $\tilde{\mathbb{T}}_{h,p}$ is determined from the step size and Butcher tableau coefficients.

Depending on the method, various finite-difference expressions are used for the derivative $\dot{y}$ on $\tilde{\mathbb{T}}_{h,p}$. Collocation methods (including certain implicit and explicit Runge-Kutta methods), and linear multi-step methods use derivatives of piecewise polynomial interpolants. General Runge-Kutta methods (which are nonlinear) may use nonlinear interpolants.

Since the differential equation is satisfied only at a finite number of points and $\mathcal{X}_{h,p}$ is a strict subset of $\mathcal{X}$, the feasibility residual r_{feas} is non-zero, in general. If the residual is too large, then the dimension of $\mathcal{X}_{h,p}$ is usually increased in correspondence with choosing a set $\mathbb{T}_{h,p}$ with more elements (as in h-methods) and/or increasing the degree of the polynomials in $\mathcal{X}_{h,p}$ (as in p- or hp-methods). It is well-known that if care is not taken

with the choice of $\mathbb{T}_{h,p}$ and the degree of the polynomials in $\mathcal{X}_{h,p}$, then the feasibility residual will not converge, e.g. due to Runge's phenomenon. It is also possible that a solution might not exist, e.g. when there are more constraints than degrees of freedom.

Essentially, problems arise in collocation-type methods because they cannot explicitly take into account what happens in-between collocation points. Problems show up in two ways: On the one hand, convergence analyses rely on very technical assumptions in order for the numerical solution to converge. On the other hand, numerical solutions ring or diverge when these assumptions are violated.

Finally, collocation discretizations yield infeasible NLPs when the OCP is consistently overdetermined. Consider, for example

$$y(0) = 1, \quad \dot{y}(t) = u(t), \quad \mathbf{0} = \begin{bmatrix} \exp(t) - u(t) \\ y(t) - u(t) \end{bmatrix}.$$

This set of constraints has the consistent analytic solution $y^\star(t) = \exp(t)$. However, using any collocation-type method yields $u_h(t) = \exp(t) \ \forall t \in \mathbb{T}_{h,p}$ from the second path constraint. Further, $\dot{y}_h(t) = u_h(t) \ \forall t \in \tilde{\mathbb{T}}_{h,p}$ from the ODE constraint. One would need to construct $\tilde{\mathbb{T}}_{h,p}$ so that $y_h(t) = \exp(t) \ \forall t \in \mathbb{T}_{h,p}$ holds, because this would be required in order for the second path constraint to be satisfied. In the case of pure collocation methods, i.e. $\tilde{\mathbb{T}}_{h,p} = \mathbb{T}_{h,p}$, the discretization is infeasible, because there is no exact linear quadrature rule for the exponential.

B. Integral Penalty-type Methods

Integral penalty-type methods differ fundamentally from collocation-type methods by explicitly considering the violation of constraints everywhere in $[t_0, t_F]$ in the formulation of the optimization problem. This is possible due to the observation that one can replace the uncountable set of constraints (OCPb) with the finite set of constraints

$$\int_T \|c\left(\dot{y}(t), y(t), u(t), t\right)\|_2^2 \mathrm{d}t \leq \varepsilon_T, \ \forall T \in \mathcal{T}_h. \quad \text{(OCPb')}$$

Here $\mathcal{T}_h$ is a finite set of disjoint intervals such that $\bigcup_{T \in \mathcal{T}_h} \mathrm{cl}(T) = [t_0, t_F]$. Clearly, (OCPb') is satisfied with $\varepsilon_T = 0$ only if (OCPb) is satisfied; however, since $\mathcal{X}_{h,p}$ is finite dimensional, usually the best that one can do is to satisfy the above constraint with $\varepsilon_T > 0$. This also acknowledges the fact that one cannot assert feasibility of a discretization when forcing the constraint residual exactly to zero at any particular point.

The approach of integral penalty-type methods is therefore to augment the objective functional with a weighted version of the integrals in (OCPb') and a quadratic penalty on the equality constraints (OCPc), namely r as defined in (3).

The inequality constraints (OCPd) also need careful consideration. Here collocation methods again only enforce these at a finite number of points $\mathbb{T}_{h,p}$. This

can result in overshooting of the numerical solution in-between collocation points.

Our penalty-barrier method in [4] treats (OCPd) by expressing them as positivity constraints, introducing suitable slack variables $s : \Omega \to \mathbb{R}_+^{n_s}$, and then augmenting the objective functional with the *integral* logarithmic barrier function

$$\Gamma(x) := -\sum_{j=1}^{n_s} \int_\Omega \log\big(s(t)\big) \, \mathrm{d}t \qquad (4)$$

to ensure that the inequality constraints are satisfied over the whole interval Ω. Details on the exact reformulation with the slacks and ensuring feasibility in the context of interior points are given in [4].

The resulting method consists of solving the penalty-barrier problem

$$\min_{x \in \mathcal{X}_{h,p}} \ J_{\omega,\tau}(x) = J(x) + \frac{1}{2\omega}r(x) + \tau\Gamma(x) \qquad \text{(PBP)}$$

for suitable penalty- and barrier parameters $\omega, \tau > 0$ with respect to the mesh size $h > 0$ and degree $p \in \mathbb{N}$ of the finite elements.

C. Convergence Results

We briefly compare collocation-type methods to integral penalty-type methods in terms of convergence guarantees. Most important is whether a method converges. Desirably, this should be achieved under mild assumptions. Thereafter, the order of convergence is of importance, so that highly accurate solutions can be obtained with an affordable amount of computing resources.

1) Collocation Methods: A convergence proof for a discretization based on the explicit Euler method is given in [11]. The result makes the following assumptions: (i) functions defining the problem must be locally differentiable with Lipschitz continuous derivatives; (ii) there must be a local solution where the trajectories of the state and free variables are continuously differentiable and continuous, respectively; (iii) a homogeneity condition on active constraints; (iv) surjectivity of linearized equality constraints; and (v) a coercivity assumption.

These conditions are sophisticated, difficult to understand, and very hard to verify and ensure by construction. The conditions ensure that the first order optimality conditions of the non-discretized and of the Euler-discretized optimization problem have a convergent solution. This is why convergence proofs for other collocation schemes make similar assumptions.

Since Euler's method converges only to first order, convergence results based on Euler's method are of limited practical use compared to results for higher-order methods. Convergence of a higher-order method is shown in [12] by making additional assumptions: (vi) functions W, w, c, g must be sufficiently smooth; (vii) the state and co-state trajectories must be sufficiently smooth; (viii) the NLP arising from the discretization must satisfy the Linear Independence Constraint Qualification and Second-Order Sufficient Condition.

2) The Penalty-Barrier Finite Element Method: In contrast to the above results for collocation methods, for PBF as in [4] convergence follows under the following assumptions:

(A.1) (OCP) has a global minimizer $x^\star$.
(A.2) $|w(\dot{y}(t), y(t), u(t), t)|$, $\qquad \|c(\dot{y}(t), y(t), u(t), t)\|_1$, $\|W(y(t_0), y(t_F))\|_1$, $\qquad \|g(y(t_0), y(t_F))\|_1$ $\qquad$ are bounded for all arguments $x \in \mathcal{X}$.
(A.3) W, w, c, g are globally Lipschitz continuous in all arguments except t.
(A.4) The problem (PBP) has a bounded solution $x_{\omega,\tau}^\star$.
(A.5) The solution $x_{\omega,\tau}^\star$ can be approximated by a function $x_h \in \mathcal{X}_{h,p}$ with an error that converges to zero as $h \searrow +0$.

We review the discussion of these assumptions from [4]:

(A.1) is reasonable.
(A.2) can be enforced by construction. To this end, W, w, c, g can be bounded below and/or above, if necessary, with minimum and maximum terms. For example, if $|J(x^\star)| \ll 10^8$ is expected, then we can replace w by a modified function

$$\tilde{w}(\chi, v, t) := \max\Big\{-10^8, \min\big\{w(\chi, v, t), 10^8\big\}\Big\},$$

and W likewise. Similar box bounds can be employed for c, g, since $r(x^\star) = 0 \ll 10^8$.
(A.3) can be enforced. Functions that are not Lipschitz continuous, e.g. the square-root or Heaviside function, can be made so by replacing them with smoothed functions, e.g. via a suitable mollifier. This is a common practice to ensure the derivatives used in a nonlinear optimization algorithm are globally well-defined.
(A.4) can be ensured in practice by using suitable box constraints (OCPd). Note that this assumption also rules out solutions with finite escape times.
(A.5) is reasonable because without it no numerical method based on piecewise polynomial representation of the solution can converge.

With the above, we give a reduced version of [4, Thm 2]:

Theorem 1: Suppose (A.1)–(A.5) hold. Let $\mathcal{X}_{h,p}$ consist of finite elements of fixed degree p on a quasi-uniform mesh. If x_h is a solution to (PBP), then

$$g_{\text{opt}}(x_h), r_{\text{feas}}(x_h) \xrightarrow{h \to 0} 0.$$

In [4, Thm 3] the result is extended to an order-of-convergence result by extending (A.5) with regard to the order of convergence of the approximation error. The result essentially says: If the finite element approximation converges of order $\ell \in (0.5, p]$ then the optimality gap and feasibility residual converge of the order $\ell/2$.

In addition, [4] analyses the convergence when (PBP) is solved inexactly, which occurs when integrals are approximated with numerical quadrature.
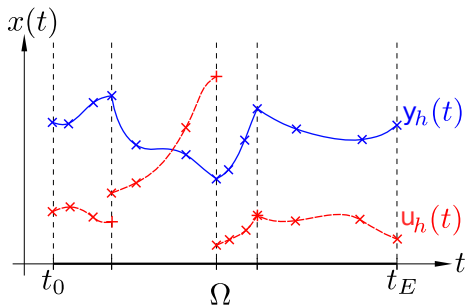
Fig. 3. Continuous and discontinuous piecewise polynomial finite element functions y_h, u_h on a mesh of four intervals.
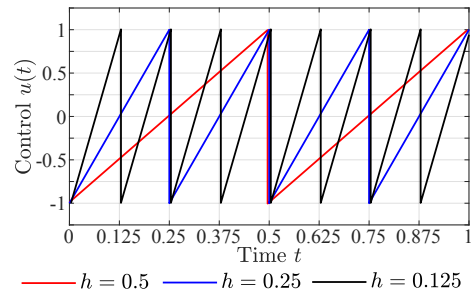

Fig. 4. A piecewise affine function, ranging from -1 to 1 on each mesh interval, which serves as a counter-example to satisfaction of (5) when using the midpoint quadrature rule when $p = 1$.

IV. IMPLEMENTATION

This section outlines how PBF can be implemented. We first discuss suitable finite element bases. For success of the discretization, a stability condition on the quadrature rule is introduced. Using the basis and quadrature, the NLP is stated and discussed in terms of optimality conditions and sparsity of derivative matrices.

A. Finite Element Basis

Independent of a collocation method or integral penalty method, a candidate $x_h \in \mathcal{X}_{h,p}$ is represented using a finite-dimensional solution vector $\mathbf{x} \in \mathbb{R}^n$, where n is the total number of finite element functions of all species of x_h, which is identified with $\mathbf{x}$ by means of a basis. Due to considerations on sparsity in the derivative matrices of the NLP, collocation methods use a nodal basis with abscissae that match the collocation points. For an integral penalty method we may opt for a nodal basis with abscissae that are a subset of the quadrature points used for evaluating the integrals; instead, we used equidistant nodes in favour of conditioning in the experiments with PBF in Section V.

Species of y must be approximated with continuous finite element functions, whereas for u one can use discontinuous elements. Figure 3 shows an example of a nodal basis representation with piecewise polynomial finite element functions of degree $p = 3$.

B. Stability Condition and Quadrature Rule

To minimize (PBP), we approximate the integrals in J and r with quadrature formulas. As a consequence of this, our numerical solutions x_h will not be exact minimizers of $J_{\omega,\tau}$, but of its numerical quadrature approximation $J_{\omega,\tau,h}$. In order to still be able to prove convergence of x_h, the quadrature rule must be chosen to satisfy the following non-trivial requirement [4]:

$$|J_{\omega,\tau}(x_h) - J_{\omega,\tau,h}(x_h)| \leq const \cdot \frac{h^\ell}{\omega} \quad \forall x_h \in \mathcal{X}_{h,p}, \quad (5)$$

where ℓ is the desired order of convergence of x_h. While in general it is not so difficult to devise a convergent quadrature rule, the devil is in the detail that the scheme must converge for *every* element in $\mathcal{X}_{h,p}$.

We illustrate the difficulty with an example. Consider $\Omega = [0, 1]$, and a uniform mesh of size $h \in 1/\mathbb{N}$ of discontinuous piecewise affine finite elements for $u : \Omega \to \mathbb{R}^1$. Consider the path constraint $c(u(t)) := \sin(\pi u(t)) = 0$ and approximate $r(x)$ with the midpoint rule quadrature formula as $r_h(x)$. Then (5) requires that the difference

$$|r_h(x) - r(x)|$$
$$= \left| h \cdot \sum_{j=1}^{1/h} \sin^2(\pi u_h(jh - h/2)) - \int_0^1 \sin^2(\pi u_h(t)) \mathrm{d}t \right|$$

converges to zero as $h \searrow +0$. However, $\mathcal{X}_{h,1}$ contains the element u_h plotted in Figure 4, which yields $r_h(x) = 0$ whereas $r(x) = 1/2$. Usually, the midpoint quadrature rule converges in $\mathcal{O}(h^3)$, but since here the integrand $u_h(t)$ can be chosen in a larger space $\mathcal{X}_{h,1}$, the quadrature rule does not converge as h is decreased. If instead a trapezoidal quadrature rule is used, then (5) is satisfied with order $\ell = 3$.

To satisfy (5) in practice, we suggest the use of Gauss-Legendre quadrature with $q = 2p$ quadrature points per element. In [4] there are further precautions so that, in addition, (4) can be approximated with Gauss-Legendre quadrature. We write $\alpha_j > 0, \tau_j \in \Omega$ for $j = 1, \ldots, N$ for the quadrature weights and abscissae.

Finally, collocation methods must also satisfy (5) as well if they want to ensure a bound on the path constraint residual. However, it is not possible for collocation methods to choose $q > p$, due to over-determination of the NLP.

C. Nonlinear Program

We define the following functions and matrix:

$$F(\mathbf{x}) := W(y_h(t_0), y_h(t_F))$$
$$+ \sum_{j=1}^N \alpha_j w(y_h(\tau_j), u_h(\tau_j)), \quad (6)$$

$$C(\mathbf{x}) := \begin{bmatrix} g(y_h(t_0), y_h(t_F)) \\ \sqrt{\alpha_1} c(\dot{y}_h(\tau_1), y_h(\tau_1), u_h(\tau_1)) \\ \vdots \\ \sqrt{\alpha_N} c(\dot{y}_h(\tau_N), y_h(\tau_N), u_h(\tau_N)) \end{bmatrix} \in \mathbb{R}^{m_\varepsilon}, \quad (7)$$

$$P_s\mathbf{x} := \begin{bmatrix} s(\tau_1) \\ \vdots \\ s(\tau_N) \end{bmatrix} \in \mathbb{R}^{m_{\mathcal{I}}}, \qquad (8)$$

where $m_{\mathcal{E}} := n_g + qNn_c$, $m_{\mathcal{I}} := qNn_s$. The quadrature approximation $J_{\omega,\tau,h}$ of the minimization function $J_{\omega,\tau}$ in (PBP) can then be expressed as

$$\min_{\mathbf{x} \in \mathbb{R}^n} \Phi(\mathbf{x}) := F(\mathbf{x}) + \frac{1}{2\omega}\|C(\mathbf{x})\|_2^2 - \tau\mathbf{g}^{\mathsf{T}}\log(P_s\mathbf{x}),$$
$$(\text{NLP})$$

where $\mathbf{g} \in \mathbb{R}^{m_{\mathcal{I}}}$ is a vector of quadrature weights.

Φ should not be minimized with a general-purpose unconstrained minimization solver. Instead, there are suitable tailored solvers for minimization of this exact function with quadratic penalties and log-barriers [4], [13], [14]. These methods compute vectors $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^{m_{\mathcal{E}}}$, $\mathbf{z} \in \mathbb{R}_+^{m_{\mathcal{I}}}$, such that

$$\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}, \mathbf{y}) - P_s^{\mathsf{T}}\mathbf{z} = \mathbf{0},$$
$$C(\mathbf{x}) + \omega\mathbf{y} = \mathbf{0},$$
$$\operatorname{diag}(P_s\mathbf{x}) \cdot \mathbf{z} - \tau\mathbf{g} = \mathbf{0}.$$

This is equivalent to the first-order necessary optimality conditions for minimizing (NLP), where $\mathcal{L}(\mathbf{x}, \mathbf{y}) := F(\mathbf{x}) - \mathbf{y}^{\mathsf{T}}C(\mathbf{x})$.

D. Derivatives and Sparsity

Second-order NLP solvers need $\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}, \mathbf{y})$, $\nabla_{\mathbf{x},\mathbf{x}}^2\mathcal{L}(\mathbf{x}, \mathbf{y})$ and $\nabla_{\mathbf{x}}C(\mathbf{x})$. We see that (6)–(7) essentially use the same formulas as collocation methods, except that they replace collocation points with quadrature points τ_j, and weight the constraints C by $\sqrt{\alpha_j}$. Hence, the Jacobian $\nabla_{\mathbf{x}}^{\mathsf{T}}C(\mathbf{x}) \in \mathbb{R}^{m_{\mathcal{E}} \times n}$ of PBF has the same sparsity structure as for a collocation method of same degree p, except the Jacobian in PBF has more rows than in LGR; e.g. twice as many rows as a collocation method of the same degree if we choose to use Gauss-Legendre quadrature with $q = 2p$ abscissae per element, as proposed. The observations for $\nabla_{\mathbf{x}}^{\mathsf{T}}C(\mathbf{x})$ can be extended to $\nabla_{\mathbf{x},\mathbf{x}}^2\mathcal{L}(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{n \times n}$: PBF$(p)$ and LGR(p) of the same degree p on the same mesh have the same Hessian dimension and sparsity structure. Section V presents sparsity plots for PBF(5) and LGR(5).

V. Numerical Experiments

We now consider a test problem for which both types of methods converge with success, so that we can compare conditioning, convergence, and rate of convergence to a known analytical solution:

$$\min_{y,u} \quad J = y_2(1), \qquad\qquad\qquad (\text{X2})$$

$$\text{s.t.} \quad y_1(0) = 1, \;\; \dot{y}_1(t) = \frac{u(t)}{2y_1(t)}, \;\; \sqrt{0.4} \le y_1(t),$$

$$y_2(0) = 0, \;\; \dot{y}_2(t) = 4y_1(t)^4 + u(t)^2, \;\; -1 \le u(t).$$

The solution is shown in Figure 5. $u^\star$ is constant outside $t_0 = 1 - \frac{\sqrt{41}}{10} \approx 0.35$ and $t_1 = t_0 + \log 2 - \frac{\log(\sqrt{41}-5)}{2} \approx$
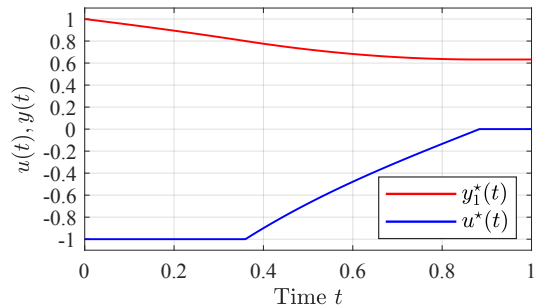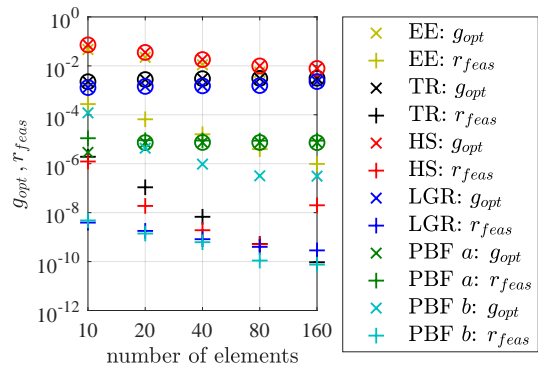

Fig. 5. Analytical solution to (X2).

Fig. 6. Convergence of optimality gap and feasibility residual LGR and PBF use polynomial degree $p = 5$. PBF uses two different values: a) $\omega = 10^{-5}$, b) $\omega = 10^{-10}$.

0.88, between which $u^\star(t) = 0.8\sinh\left(2(t - t_1)\right)$, yielding $J \approx 2.057866062168$.

All methods yield accurate solutions. Figure 6 shows the convergence of the optimality gap and feasibility residual of a respective method. Remarking on the former, we computed $J^\star - J(x_h)$ and encircled the cross when $J(x_h) < J^\star$. Note in the figure that for ≥ 40 elements the most accurate solutions in terms of feasibility are found by PBF with $\omega = 10^{-10}$. Further, we find that the collocation methods significantly underestimate the optimality value for this experiment.

Now we discuss rates of convergence. Convergence of only first order is expected because higher derivatives of $y^\star$ are non-smooth and $u^\star$ has edges. Indeed, r_{feas} converges linearly for all methods. PBF5 with $\omega = 10^{-5}$ stagnates early because it converges to the optimal penalty solution, which for this instance is converged from 20 elements onwards. $g_{\text{opt}}, r_{\text{feas}}$ are then fully determined by ω. The issue is resolved by choosing ω smaller. LGR5 and PBF5 with $\omega = 10^{-10}$ converge similarly, and stagnate at $r_{\text{feas}} \approx 10^{-10}$. Due to the high exponent in the objective, small feasibility errors in the collocation methods amount to significant underestimation of the objective.

Finally, we look into computational cost. Solving the collocation methods with IPOPT and the PBF5 discretization with the interior-point method in [13], the optimization converges in ≈ 20 iterations for any dis-
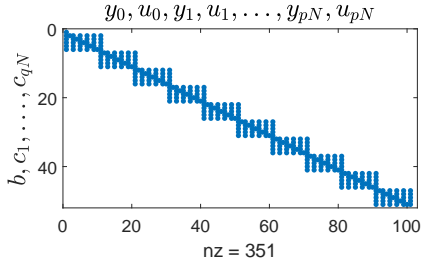
Fig. 7. Sparsity of $\nabla_{\mathbf{x}} C(\mathbf{x})^{\mathsf{T}}$ for LGR5 when $h = \frac{1}{10}$, i.e. $N = 10$. For LGR, notice $q = p - 1$. The discretization does not depend on u_{pN}.
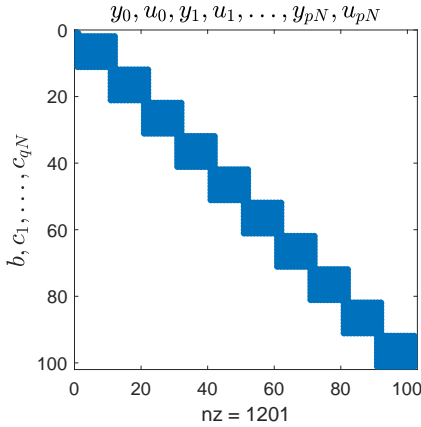

Fig. 8. Sparsity of $\nabla_{\mathbf{x}} C(\mathbf{x})^{\mathsf{T}}$ for PBF5 when $h = \frac{1}{10}$, i.e. $N = 10$, with $q = 2p$.

cretization. Differences in computational cost can arise when one discretization results in much denser or larger problems than others. Here, we compare the sparsity structure of the Jacobian $\nabla_{\mathbf{x}} C(\mathbf{x})^{\mathsf{T}}$ for LGR5 in Figure 7 and PBF5 in Figure 8, each using a mesh size of $h = \frac{1}{10}$. Note that for PBF5, $C(\mathbf{x})$ has a larger dimension $m_{\mathcal{E}}$ than LGR5, thus the Jacobian has more rows and hence more non-zeros. However, critical for computations is the primal Schur complement $\mathbf{\Sigma} = \nabla_{\mathbf{xx}}^2 \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) + \nabla_{\mathbf{x}} C(\mathbf{x})^{\mathsf{T}} \mathbf{D} \nabla_{\mathbf{x}} C(\mathbf{x})$, which is used when solving the KKT system via the reduced form, where $\mathbf{D}$ is a diagonal matrix. $\mathbf{\Sigma}$ is a narrow-banded matrix with dense band of the same bandwidth for LGR5 and PBF5.

With regard to computational cost, it follows from Figure 6 that the ability to choose ω in PBF can be advantageous. In particular, on coarse meshes, one may opt for small feasibility residual by manually decreasing ω, whereas with a collocation method one is stuck with the feasibility residual that one obtains for that particular mesh. The figure shows this: For $\omega = 10^{-10}$, even on the coarsest mesh the PBF method achieves a solution that has a smaller feasibility residual than other methods on the same mesh. For this problem this becomes possible because the path constraint could be satisfied with zero

error by choosing y a polynomial of degree 3 (because here PBF uses $p = 5$).

VI. Conclusions

We compared classical direct discretization methods, i.e. collocation-type methods, with integral penalty-type methods, and in particular with PBF. We highlighted the critical difference, that is, forcing exact constraint satisfaction at a finite number of points for the former, versus forcing approximate constraint satisfaction everywhere for the latter. We also described the consequences of these approaches on the convergence: Collocation-type methods achieve a fixed bias between optimality gap and feasibility residual at a given mesh, whereas with PBF one can trade between the two. Further, collocation methods may fail to converge in several cases where PBF converges.

The convergence has been investigated by reviewing theorems and performing numerical experiments. Both indicate that direct transcription based on integral penalties is competitive in computational cost and accuracy compared to state-of-the-art collocation methods in the general case, but in addition also works reliably in applications where the latter struggle to converge.

The robustness of an optimal control solver is determined by the weakest link in the chain. The discretization is only one half of the numerical method. Future work could therefore be dedicated to the development of tailored NLP solvers to improve reliability and computational efficiency.

References

[1] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, 2nd ed. New York, NY, USA: Cambridge University Press, 2010.

[2] A. V. Balakrishnan, "On a new computing technique in optimal control," *SIAM J. Control*, vol. 6, pp. 149–173, 1968.

[3] W. W. Hager, "Multiplier methods for nonlinear optimal control," *SIAM J. Numer. Anal.*, vol. 27, no. 4, pp. 1061–1080, 1990. [Online]. Available: https://doi.org/10.1137/0727063

[4] M. P. Neuenhofen and E. C. Kerrigan, "Dynamic optimization with convergence guarantees," *arXiv:1810.04059*, 2018.

[5] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *SIAM Rev.*, vol. 59, no. 4, pp. 849–904, 2017. [Online]. Available: https://doi.org/10.1137/16M1062569

[6] M. A. Patterson and A. V. Rao, "GPOPS-II: a Matlab software for solving multiple-phase optimal control problems using hp-adaptive Gaussian quadrature collocation methods and sparse nonlinear programming," *ACM Trans. Math. Software*, vol. 41, no. 1, pp. Art. 1, 37, 2014. [Online]. Available: https://doi.org/10.1145/2558904

[7] W. W. Hager, J. Liu, S. Mohapatra, A. V. Rao, and X.-S. Wang, "Convergence rate for a Gauss collocation method applied to constrained optimal control," *SIAM J. Control Optim.*, vol. 56, no. 2, pp. 1386–1411, 2018. [Online]. Available: https://doi.org/10.1137/16M1096761

[8] B. A. Conway, "A Survey of Methods Available for the Numerical Optimization of Continuous Dynamic Systems," *Journal of Optimization Theory and Applications*, vol. 152, no. 2, pp. 271–306, Feb 2012. [Online]. Available: https://doi.org/10.1007/s10957-011-9918-z

[9] A. V. Rao, "Survey of Numerical Methods for Optimal Control," in *Advances in the Astronautical Sciences*, vol. 135, 2010.

[10] R. Quirynen, S. Gros, B. Houska, and M. Diehl, "Lifted collocation integrators for direct optimal control in acado toolkit," *Mathematical Programming Computation*, vol. 9, no. 4, p. 527–571, 2017.

[11] K. Malanowski, C. Büskens, and H. Maurer, "Convergence of approximations to nonlinear optimal control problems," in *Mathematical programming with data perturbations*, ser. Lecture Notes in Pure and Appl. Math. Dekker, New York, 1998, vol. 195, pp. 253–284.

[12] S. Kameswaran and L. T. Biegler, "Convergence rates for direct transcription of optimal control problems using collocation at Radau points," *Comput. Optim. Appl.*, vol. 41, pp. 81–126, 2008.

[13] A. Forsgren and P. E. Gill, "Primal-dual interior methods for nonconvex nonlinear programming," *SIOPT*, vol. 8, no. 4, pp. 1132–1152, 1998. [Online]. Available: https://doi.org/10.1137/S1052623496305560

[14] Y. Cao, A. Seth, and C. D. Laird, "An augmented Lagrangian interior-point approach for large-scale NLP problems on graphics processing units," *Computers and Chemical Engineering*, vol. 85, pp. 76 – 83, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0098135415003257