SIAM J. OPTIM.
Vol. 24, No. 1, pp. 528–558

SCALABLE NONLINEAR PROGRAMMING VIA EXACT DIFFERENTIABLE PENALTY FUNCTIONS AND TRUST-REGION NEWTON METHODS*

VICTOR M. ZAVALA† AND MIHAI ANITESCU†

Abstract. We present an approach for nonlinear programming based on the direct minimization of an exact differentiable penalty function using trust-region Newton techniques. The approach provides desirable features required for scalability: it can detect and exploit directions of negative curvature, it is superlinearly convergent, and it enables the scalable computation of the Newton step through iterative linear algebra. Moreover, it presents features that are desirable for parametric optimization problems that must be solved in a latency-limited environment, as is the case for model predictive control and mixed-integer nonlinear programming. These features are fast detection of activity, efficient warm starting, and progress on a primal-dual merit function at every iteration. We note that other algorithmic approaches fail to satisfy at least one of these features. We derive general convergence results for our approach and demonstrate its behavior through numerical studies.

Key words. scalable, nonlinear programming, exact differentiable penalty, trust region, Newton, iterative linear algebra

AMS subject classifications. 34B15, 34H05, 49N35, 49N90, 90C06, 90C30, 90C55, 90C59

DOI. 10.1137/120888181

1. Problem definition and previous work.

We consider the general nonlinear programming problem (NLP)

$$\min_x \quad f(x) \tag{1.1a}$$

$$\text{s.t.} \quad h(x) = 0_m \quad (\lambda), \tag{1.1b}$$

$$x \geq 0_n \quad (\mu). \tag{1.1c}$$

Here, the $x \in \mathcal{R}^n$ are primal variables and $\lambda \in \mathcal{R}^m$ and $\mu \in \mathcal{R}^n$ are multipliers for the equality constraints and bounds, respectively. We note that any NLP with general inequality constraints can be transformed into this form. In addition, the properties of the approach presented in this work can be attained for any NLP once it is transformed into this form.

Our work is motivated by the need to compute fast solutions to parametric NLPs such as those arising in model predictive control (MPC). In this area researchers have shown that using one major sequential quadratic programming (SQP) iteration (i.e., computing an inexact solution of the NLP) at each point t along a parameter trajectory, results in an asymptotically stable tracking strategy of the nonsmooth solution manifold [18, 19, 49]. Here we analyze exclusively the NLP (1.1), but the application of NLP algorithms in parametric settings motivates us to seek certain properties. A critical factor in MPC is *latency:* the time before a decision (e.g., control action) is computed. While in [19, 49] the latency is the time it takes to do a

*Received by the editors August 15, 2012; accepted for publication (in revised form) January 24, 2014; published electronically March 27, 2014.
http://www.siam.org/journals/siopt/24-1/88818.html
†Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439 (vzavala@mcs.anl.gov, anitescu@mcs.anl.gov).

quadratic programming (QP) solve, in this work we aim to reduce this even further for cases where the dimension of the underlying NLP is very large.

We have two objectives in designing the algorithms for such problems. First, we seek properties that allow very large NLPs to be solved efficiently. To this end, we seek an algorithm with global convergence properties and the following features:

(i) *superlinear convergence* in the primal-dual space;[1]

(ii) *scalable step computation.* It is matrix-free in that it does not require matrix factorizations, it enables the use of iterative linear algebra, and it does not require storage of derivative matrices;

(iii) *negative curvature.* It detects and exploits directions of negative curvature.

Second, we seek good properties in latency-limited environments, such as MPC, where the problem at parameter t (e.g., data parameterized in time) may need to be solved incompletely because of low-latency requirements before new data come in and the parameter is advanced to $t + \Delta t$. Yet, even in this circumstance it is desirable to show improvement in a way that transitions into good NLP convergence properties, so properties (i), (ii), and (iii) are still relevant. This situation occurs, for example, in MPC when responding to a parametric perturbation Δt, and one restarts far away from the new optimal manifold. The latency concern adds the following properties to our set of requirements:

(iv) *asymptotic monotonicity in minor iterations.* Each minor iteration has small complexity and makes constant progress in a primal-dual function, at least sufficiently close to the solution;

(v) *active set detection and warm-start.* The algorithm enables fast detection of activity, enabling effective warm-starting procedures.

The main algorithmic frameworks found to date can be broadly categorized as interior point (IP), active set SQP, and active set augmented Lagrangian (AL). IP algorithms satisfy (i) [23]. Property (ii) is currently achieved by using incomplete symmetric indefinite factorizations for preconditioning the augmented system. Property (iii) is achieved by detecting its inertia [42] or testing the resulting direction for descent [13] and modifying the Hessian matrix to achieve the correct inertia or other criteria and recalculating the search direction [46, 13]. Another approach consists in using a constraint-projected preconditioned conjugate gradient (PCG) scheme for the augmented system with constraint preconditioners [24] detecting and exploiting directions of negative curvature through trust-region regularization [47]. While this does not require the factorization of the Karush–Kuhn–Tucker (KKT) matrix, it does require the factorization of the constraint preconditioner and thus does not fully satisfy (ii). Property (iv) is currently enforced indirectly by using filter mechanisms or primal merit functions [46, 10, 47]. In order to achieve (i) in this context, however, second-order corrections are needed to avoid the Maratos effect [45]. As for property (v), activity detection is efficient from cold start. For warm starting, reductions in iteration counts are achieved by recovering centrality or crossing over to SQP; but this is cumbersome and computationally costly [27, 48], so IP algorithms do not generally perform well with respect to (v).

SQP algorithms largely mimic the features of IP with respect to (i)–(v) with two exceptions. Emphasis on exact satisfaction of the linearized constraints generally requires a factorization, though perhaps only the one of a constraint preconditioner, so (ii) may not be satisfied. With respect to (v), cold-start active set detection can be slow, requiring at least one minor QP iteration per active variable. Thus, detecting

[1] We use the term *primal-dual function* to define a function that incorporates primal variables and dual variables only for the equality constraints.

multiple changes in activity can be inefficient in latency-limited environments (e.g., might require several factorizations), so (v) is not entirely satisfied [4].

AL frameworks can achieve (ii) and (iii) by using iterative linear algebra on the Hessian of the AL [12, 49]. Property (iv) can be achieved by using the AL function directly as a merit function [11]. Property (v) can be achieved by using gradient projection methods for the bound-constrained AL subproblems [7, 49] although it is still not clear how to initialize the penalty parameter to enable efficient warm starts. Property (i) is satisfied only when second-order multiplier updates or linearly constrained formulations are used [6, 29]; these require major linear algebra operations and thus prevent satisfaction of (ii). An alternative is to use primal-dual AL functions [26] to compute steps on both primal and dual variables simultaneously, but it is still not clear how to ensure (ii), (iii), (iv), and (v) under such a framework.

In this work, we build on an interesting class of exact differentiable penalty functions (EDPFs) that was originally proposed and analyzed by Di Pillo and Grippo for equality-constrained NLPs [16]. These functions are primal-dual AL functions having the distinctive feature that they incorporate a penalty term for the norm of the gradient of the Lagrangian. In addition to the good global convergence properties typical of exact penalty functions, Di Pillo and Grippo's function has powerful properties that can be exploited to construct algorithms that directly minimize it. This satisfies properties (i) and (iii). As for property (ii), the main limitation in designing algorithms based on direct minimization of Di Pillo and Grippo's function is the appearance of complex third-order derivative terms. While these terms can be shown to vanish at a stationary point of the NLP, they greatly complicate the search because they introduce nonconvexity. Consequently, capabilities to detect and exploit directions of negative curvature are essential to enable (ii) and (iii). As for property (v), Di Pillo and Grippo proposed EDPFs to deal with inequality constraints; but these are twice differentiable only in a neighborhood of the solution, thus preventing (iv) and hindering (ii) [38, 17]. A reformulation, central to this work, that maintains the smoothness of the EDPF by enforcing bound constraints explicitly is presented by Bertsekas [6, pp. 229–230]. The function is twice differentiable everywhere, thus enabling (iv) and potentially (ii) and (iii) if capabilities to detect and exploit directions of negative curvature are devised. Moreover, to put the comparison with other methods on an equal footing, an efficient method based on such EDPFs should avoid evaluation of third derivatives without hindering (i) or global convergence.

In this work, we demonstrate that one can construct NLP algorithms based on the minimization of Bertsekas' EDPF capable of achieving all the properties (i)–(v) while avoiding the computation of third derivatives and efficiently handling their nonconvexity. The latter two objectives are accomplished by using a trust-region/gradient-projection approach similar to [31] to solve Bertsekas' bound-constrained EDPF problem [6, pp. 229–230]. Our contributions are the following. (1) We formalize the properties of the bound-constrained EDPF to establish conditions under which first- and second-order conditions for stationary points of the NLP and of the bound-constrained EDPF coincide in the case of inequality constraints. This formalization extends existing results for the case of equality constraints [6, 16]. (2) We derive formulas for the approximate Hessian and the reduced approximate Hessian for the EDPF and discuss their asymptotic convergence properties to the exact counterparts. (3) We demonstrate that superlinear and global convergence can be achieved by using a trust-region Newton framework [31]. (4) We demonstrate that the framework satisfies properties (i)–(v), and we present numerical examples to illustrate this.

The paper is structured as follows. In section 2 we introduce basic notation and properties of the NLP. In section 3 we reformulate the NLP as a bound-constrained minimization problem using the EDPF. In section 4 we present conditions under which stationary points for the EDPF and NLP problems coincide. In section 5 we present the trust-region Newton algorithm and convergence results. Numerical results are presented in section 6. The last section presents concluding remarks and directions of future work.

2. Basic properties and notation. We define the *partial* Lagrange function of the NLP (1.1):

$$\mathcal{L}(x,\lambda) = f(x) + \lambda^T h(x). \tag{2.1}$$

Here, $f : \mathcal{R}^n \to \mathcal{R}$ and $h : \mathcal{R}^n \to \mathcal{R}^m$. The first and second derivatives of the Lagrange function are

$$\nabla_x \mathcal{L}(x,\lambda) = \nabla_x f(x) + \nabla_x h(x)\lambda, \tag{2.2a}$$
$$\nabla_\lambda \mathcal{L}(x,\lambda) = h(x), \tag{2.2b}$$
$$\nabla_{x,x} \mathcal{L}(x,\lambda) = \nabla_{x,x} f(x) + \nabla_x(\nabla_x h(x)\lambda), \tag{2.2c}$$
$$\nabla_{\lambda,x} \mathcal{L} = \nabla_x h(x), \tag{2.2d}$$
$$\nabla_{\lambda,\lambda} \mathcal{L} = 0_{m\times m}. \tag{2.2e}$$

Here, $\nabla_x h(x) \in \mathcal{R}^{n\times m}$ is the Jacobian of the equality constraints and we define the primal-dual vector $w^T := [x^T, \lambda^T]$ with $w \in \mathcal{R}^{n_w}$ and $n_w = n + m$. Using this notation, we can write

$$\nabla_w \mathcal{L}(w) = \begin{bmatrix} \nabla_x \mathcal{L}(x,\lambda) \\ h(x) \end{bmatrix}, \tag{2.3a}$$
$$\nabla_{w,w} \mathcal{L}(w) = \begin{bmatrix} \nabla_{x,x} \mathcal{L}(x,\lambda) & \nabla_x h(x) \\ \nabla_x h(x)^T & 0_{m\times m} \end{bmatrix}. \tag{2.3b}$$

We denote a solution of the NLP (1.1) using the pair x^*, λ^*, or w^*. When convenient, we will also use the dual vector μ^* for the bounds. The KKT conditions for the NLP (1.1) are given by

$$0_n = \nabla_x \mathcal{L}(x^*,\lambda^*) - \mu^*, \tag{2.4a}$$
$$0_m = \nabla_\lambda \mathcal{L}(x^*,\lambda^*), \tag{2.4b}$$
$$0_n \le x^* \perp \mu^* \ge 0_n. \tag{2.4c}$$

In compact form,

$$0_n \le x^* \perp \nabla_x \mathcal{L}(x^*,\lambda^*) \ge 0_n, \tag{2.5a}$$
$$0_m = \nabla_\lambda \mathcal{L}(x^*,\lambda^*). \tag{2.5b}$$

We state the KKT conditions in these two forms to clarify relationships between different reformulations of the NLP (1.1) and their corresponding solutions.

The first expression in (2.5) implies that $X^* \nabla_x \mathcal{L}(x^*,\lambda^*) = 0_n$ with $X^* := \mathrm{diag}(x^*)$. We define the active set $\mathcal{A}(x^*) := \{\, i \in \{1..n\} \mid x^*_{(i)} = 0 \,\}$ and the inactive set $\mathcal{I}(x^*)$, where $\mathcal{I}(x^*) \cup \mathcal{A}(x^*) = \{1..n\}$, $\mathcal{I}(x^*) \cap \mathcal{A}(x^*) = \emptyset$, and $n^*_d = \mathrm{card}(\mathcal{I}(x^*))$. We also define the inactive (free) vector $x^*_d \in \mathcal{R}^{n^*_d}$ and corresponding basis matrix *for the Jacobian of the active bound constraints* (1.1c) as $N_x(x^*) \in \mathcal{R}^{n\times n^*_d}$ such

that x^* can be written as $x^* = N_x(x^*)x_d^*$. In our case, the structure of $N_x(\cdot)$ can be trivially determined from x because the inequality constraints are simple bounds. We also define the corresponding subspace matrix (Jacobian of the active bound constraints) $B_x(x^*) \in \mathcal{R}^{(n-n_d^*)\times n}$, which satisfies $B_x(x^*)x^* = 0_{(n-n_d^*)}$ and $B_x(x^*)N_x(x^*) = 0_{(n-n_d^*)\times n_d^*}$.

Using these definitions, we now state first-order necessary conditions (KKT) and strong second-order sufficient conditions (SSOC) for the particular structure of NLP (1.1). We begin by stating the linear independence constraint qualification (LICQ) and the strict complementarity (SC) condition. LICQ requires the gradients of the constraints functions $h(x)$ and $x_i = 0$ for $i \in \mathcal{A}(x^*)$ to be linearly independent. LICQ implies [35, Theorem 12.1] that there exist unique Lagrange multipliers λ^*, μ^* satisfying the KKT conditions (2.5). SC requires that exactly one of $\mu_{(i)}^*$ and $x_{(i)}$ be zero for $i = 1, \ldots, n$.

If SC and LICQ hold, we can define the critical cone for (1.1) as (see pp. 330 and 337 in [35]),

$$\mathcal{C}(x^*, \mu^*) := \text{Null}\left[\nabla_x h(x^*)^T\right] \cap \mathcal{S}(x^*, \mu^*), \tag{2.6}$$

where

$$\mathcal{S}(x^*, \mu^*) := \left\{ x \in \mathcal{R}^n \,|\, x_{(i)} = 0,\ i \in \mathcal{B}(x^*, \mu^*) \right\}, \tag{2.7a}$$

$$\mathcal{B}(x^*, \mu^*) := \left\{ i \in \mathcal{A}(x^*) \,|\, \mu_{(i)}^* > 0 \right\}. \tag{2.7b}$$

We note that at any point x^*, λ^*, μ^* satisfying (2.4) (or (2.5)), we have that $\nabla_x \mathcal{L}(x^*, \lambda^*) = \mu^*$. Consequently, we can define the SC condition and the critical cone (2.6) in terms of the gradient of the Lagrange function as

$$\mathcal{C}(x^*, \lambda^*) := \text{Null}\left[\nabla_x h(x^*)^T\right] \cap \mathcal{S}(x^*, \lambda^*), \tag{2.8}$$

where

$$\mathcal{S}(x^*, \lambda^*) := \left\{ x \in \mathcal{R}^n \,|\, x_{(i)} = 0,\ i \in \mathcal{B}(x^*, \lambda^*) \right\}, \tag{2.9a}$$

$$\mathcal{B}(x^*, \lambda^*) := \left\{ i \in \mathcal{A}(x^*) \,|\, \nabla_x \mathcal{L}(x^*, \lambda^*)_{(i)} > 0 \right\}. \tag{2.9b}$$

For our definition of SSOC we thus assume that for any x^*, λ^* satisfying (2.5), LICQ and SC hold and that

$$\nu^T \nabla_{x,x} \mathcal{L}(x^*, \lambda^*)\nu > 0, \quad \nu \in \mathcal{C}(x^*, \lambda^*), \quad \nu \neq 0, \tag{2.10}$$

holds. This assumption implies that x^* is a strict local minimizer of the NLP (1.1). This also implies [35, Theorem 12.6] that there exists $\kappa > 0$ such that

$$\nu^T \nabla_{x,x} \mathcal{L}(x^*, \lambda^*)\nu > \kappa \|\nu\|^2, \quad \nu \in \mathcal{C}(x^*, \lambda^*). \tag{2.11}$$

We can state the SSOC by first restricting vector ν to the subspace $\mathcal{S}(x^*, \lambda^*)$. Specifically, we define $\nu := N_x(x^*)\nu_x$, where $\nu_x \in \mathcal{R}^{n_d^*}$ and where the columns of $N_x(x^*)$ span the space $\mathcal{S}(x^*, \lambda^*)$. With this definition, the SSOC can be stated for $\nu_x \in \mathcal{R}^{n_d^*}$ such that $N_x(x^*)\nu_x \neq 0$, $B_x(x^*)N_x(x^*)\nu_x = 0_{(n-n_d)}$, and $\nabla_x h(x^*)TN_x(x^*)\nu_x = 0_m$. This interpretation will become convenient in sections 4 and 5.

Throughout the manuscript, we refer to the following assumptions for the original NLP problem (1.1). We state these assumptions here to avoid any confusion with assumptions made for the EDPF problem described later.

A1. The functions $f(\cdot), h(\cdot)$ are at least two times continuously differentiable (A1a) or at least three times continuously differentiable (A1b).
A2. Any stationary point x^* with associated λ^* satisfies LICQ (A2a) and SC (A2b).
A3. Any stationary point x^* with associated λ^* satisfies the SSOC (2.10).

We note that (A1b) is not a standard assumption in nonlinear programming, but it is standard for EDPFs [6, Proposition 4.16]. This is required in order to guarantee that the EDPF defined in the following section is at least twice continuously differentiable. We also note that (A3), in the way stated, implies (A2a) and (A2b).

3. Exact differentiable penalty formulation. To derive the EDPF used in this work, we first note that one can eliminate the inequality constraints of the NLP (1.1) by using squared-slack variables $z \in \mathcal{R}^n$ [28, 6]. We get

$$\min_{x,z} \quad f(x) \quad \text{s.t.} \quad h(x) = 0_m, \quad x = z^2. \tag{3.1}$$

Here, z^2 denotes a vector composed of the squared elements of z, applied componentwise. The Lagrange function of this problem is given by

$$\bar{\mathcal{L}}(x, \lambda, z, \mu) = \mathcal{L}(x, \lambda) - \mu^T (x - z^2). \tag{3.2}$$

A stationary point $\tilde{x}^*, \tilde{\lambda}^*, \tilde{z}^*, \tilde{\mu}^*$ satisfies the KKT conditions

$$0_n = \nabla_x \bar{\mathcal{L}}(\tilde{x}^*, \tilde{\lambda}^*, \tilde{z}^*, \tilde{\mu}^*) = \nabla_x \mathcal{L}(\tilde{x}^*, \tilde{\lambda}^*) - \tilde{\mu}^*, \tag{3.3a}$$

$$0_n = \nabla_z \bar{\mathcal{L}}(\tilde{x}^*, \tilde{\lambda}^*, \tilde{z}^*, \tilde{\mu}^*) = 2\,\tilde{Z}^* \tilde{\mu}^*, \tag{3.3b}$$

$$0_m = \nabla_\lambda \bar{\mathcal{L}}(\tilde{x}^*, \tilde{\lambda}^*, \tilde{z}^*, \tilde{\mu}^*) = h(\tilde{x}^*), \tag{3.3c}$$

$$0_n = \nabla_\mu \bar{\mathcal{L}}(\tilde{x}^*, \tilde{\lambda}^*, \tilde{z}^*, \tilde{\mu}^*) = \tilde{x}^* - (\tilde{z}^*)^2, \tag{3.3d}$$

where $Z = \operatorname{diag}(z)$. We can see that at a real stationary point satisfying SC, we have $\tilde{z}^*_{(i)} = \sqrt{\tilde{x}^*_{(i)}}$, $i = 1, \ldots, n$, which implies $\tilde{x}^*_{(i)} > 0$, $\tilde{z}^*_{(i)} > 0$ and $\tilde{\mu}^*_{(i)} = 0$ for the inactive variables. Also, if $\tilde{x}^*_{(i)} = 0$ then $\tilde{z}^*_{(i)} = 0$ and $\tilde{\mu}^*_{(i)} > 0$ for the active variables. In other words, (3.3b) and (3.3d) represent the complementarity conditions between the gradient of the Lagrangian and the primal variables (2.5a).

One can show that $\tilde{x}^*, \tilde{\lambda}^*, \tilde{\mu}^*$ is a minimizer for the original NLP (1.1) if and only if $\tilde{x}^*, \tilde{z}^* = \sqrt{\tilde{x}^*}, \tilde{\lambda}^*, \tilde{\mu}^*$ is a minimizer of the reformulated NLP (3.1). This implies that $x^* = \tilde{x}^*, \lambda^* = \tilde{\lambda}^*$, and $\mu^* = \tilde{\mu}^*$. In addition, if the SSOC, LICQ, and SC hold for the original NLP (1.1) at this point (A2), (A3), then these properties are inherited by (3.1) [6, Proposition 1.32]. Note, however, that the reformulated NLP can have spurious stationary points that are not stationary points of the original NLP. These undesired stationary points can be avoided, however, by using algorithms with second-order guarantees. To see this, consider Example 7.2 in [25],

$$\min\ (x_1 - 1)^2 + (x_2 + 1)^2 \tag{3.4a}$$

$$\text{s.t. } x_1 \geq 0. \tag{3.4b}$$

This problem has a unique stationary point at $(1, -1)$. The squared reformulation sets $z_1^2 = x_1$ and $z_2 = x_2$ (by direct elimination of the $x = z^2$ constraints). The reformulated problem thus becomes,

$$\min\ (z_1^2 - 1)^2 + (z_2 + 1)^2. \tag{3.5}$$

The gradient and Hessian of this problem are, respectively,

$$\nabla_z \mathcal{L} = \begin{bmatrix} 4z_1(z_1^2 - 1) \\ 2(z_2 + 1) \end{bmatrix}, \qquad \nabla_{z,z} \mathcal{L} = \begin{bmatrix} 4(3z_1^2 - 1) & 0 \\ 0 & 2 \end{bmatrix}. \tag{3.6}$$

This problem has two stationary points: one at $(1, -1)$, at which the Hessian is positive definite, and one at $(0, -1)$, at which the Hessian is indefinite. Consequently, in order to identify the right stationary point, a solver needs to monitor curvature along the search.

This analysis indicates that we can, in principle, solve the NLP (1.1) indirectly by solving the equality-constrained NLP (3.1). The squared-slacks reformulation has been widely used, particularly in the control community [36, 20, 34], as a mechanism to eliminate inequality constraints. An important caveat of this reformulation, however, arises from a computational point of view because the introduction of squared slacks can lead to numerical instability and poor solver performance [49, 3, 25]. This is related primarily to the introduction of the complementarity condition (3.3b).

While not directly amenable to computation because of numerical instability, Bertsekas noticed that reformulation (3.1) provides a setting to construct EDPFs. He considered the penalty function of Di Pillo and Grippo [38] for the equality-constrained NLP $\min_x f(x)$ s.t. $h(x) = 0$,

$$\bar{P}_{\alpha,\beta}(w) = \mathcal{L}(w) + \frac{1}{2}\alpha h(x)^T h(x) + \frac{1}{2}\beta \nabla_x \mathcal{L}(w)^T \nabla_x \mathcal{L}(w), \tag{3.7}$$

where $\alpha, \beta \in \mathcal{R}_+$ are scalar parameters. We note that the partial Lagrange function of the reformulated problem (3.1) can also be expressed as

$$\mathcal{L}(z^2, \lambda) = f(z^2) + \lambda^T h(z^2). \tag{3.8}$$

This results from the explicit elimination of the constraint $x = z^2$ in (3.1) which yields the problem $\min_z f(z^2)$ *s.t.* $h(z^2) = 0$ because $z = \sqrt{x}$ under the implicit condition that $x \geq 0_n$, required to obtain a real solution. The stationarity condition with respect to z for this problem is

$$\nabla_z \mathcal{L}(z^2, \lambda) = 2Z \left(\nabla_{z^2} f(z^2) + \nabla_{z^2} h(z^2) \lambda \right). \tag{3.9}$$

The EDPF (3.7) of the reformulated problem $\min_z f(z^2)$ *s.t.* $h(z^2) = 0$ is

$$\begin{aligned} P_{\alpha,\beta}(z^2, \lambda) = \mathcal{L}(z^2, \lambda) + \frac{1}{2}\alpha h(z^2)^T h(z^2) \\ + 2\beta \left(\nabla_{z^2} f(z^2) + \nabla_{z^2} h(z^2) \lambda \right)^T ZZ \left(\nabla_{z^2} f(z^2) + \nabla_{z^2} h(z^2) \lambda \right). \end{aligned} \tag{3.10}$$

To avoid numerical instability resulting from direct minimization of (3.10) with respect to z, we apply the reformulation of Bertsekas [6, pp. 229–230]. The reformulation poses (3.10) in terms of x and enforces $x \geq 0_n$ in the minimization. This is done by back substitution of $x = z^2$ in (3.10). The EDPF (3.10) becomes,

$$P_{\alpha,\beta}(x, \lambda) = \mathcal{L}(x, \lambda) + \frac{1}{2}\alpha h(x)^T h(x) + 2\beta \nabla_x \mathcal{L}(x, \lambda)^T X \nabla_x \mathcal{L}(x, \lambda), \tag{3.11}$$

s.t. $x \geq 0_n$. We emphasize that (3.7) is defined for the NLP $\min_x f(x)$ s.t. $h(x) = 0$, (3.10) is defined for problem $\min_z f(z^2)$ *s.t.* $h(z^2) = 0$, and (3.11) is a reformulation of (3.10) in terms of x. From (3.11), we can see that the last term includes the

complementarity condition between the gradient of the Lagrangian and the primal variables. We also note that this penalty function can be seen as Di Pillo and Grippo's function (3.7) with the term in the last inner product scaled by the weighting matrix $4\sqrt{X}\sqrt{X}$. In compact form, the EDPF becomes

$$P_{\alpha,\beta}(x,\lambda) = \mathcal{L}(x,\lambda) + \frac{1}{2}\nabla_w \mathcal{L}(x,\lambda)^T K_{\alpha,\beta}(x) \nabla_w \mathcal{L}(x,\lambda), \tag{3.12}$$

with

$$K_{\alpha,\beta}(x) = \begin{bmatrix} 4\beta X & \\ & \alpha \mathbb{I}_m \end{bmatrix}. \tag{3.13}$$

We can thus, in principle, find a minimizer of the original NLP (1.1) by finding a minimizer of the EDPF problem

$$\min_{x,\lambda} \quad P_{\alpha,\beta}(x,\lambda) \quad \text{s.t.} \quad x \geq 0_n. \tag{3.14}$$

The KKT conditions for a solution $x^*(\alpha,\beta), \lambda^*(\alpha,\beta)$ are

$$0_n \leq x^* \perp \nabla_x P_{\alpha,\beta}(x^*,\lambda^*) \geq 0_n, \tag{3.15a}$$

$$0_m = \nabla_\lambda P_{\alpha,\beta}(x^*,\lambda^*). \tag{3.15b}$$

We define the active set at $w^* = (x^*,\lambda^*)$ as $\mathcal{A}_P(w^*) := \{\, i \in 1..n \mid w^*_{(i)} = 0\}$ and the inactive set $\mathcal{I}_P(w^*)$, where $\mathcal{I}_P(w^*) \cup \mathcal{A}_P(w^*) = \{1..n_w\}$, $\mathcal{I}_P(w^*) \cap \mathcal{A}_P(w^*) = \emptyset$, and $d := \text{card}(\mathcal{I}(w^*))$. Here, d is the number of inactive (free) variables in the primal-dual vector w^*. Note that these definitions imply that the multipliers λ are left free.

The SSOC for the EDPF problem can be stated as

$$\nu^T \nabla_{w,w} P_{\alpha,\beta}(w^*) \nu > 0, \quad \nu \in \mathcal{S}_P(w^*), \quad \nu \neq 0, \tag{3.16}$$

where

$$\mathcal{S}_P(w^*) := \left\{\, w \in \mathcal{R}^{n_w} \,|\, w_{(i)} = 0, \quad i \in \mathcal{B}_P(w^*) \,\right\}, \tag{3.17a}$$

$$\mathcal{B}_P(w^*) := \left\{\, i \in \mathcal{A}_P(w^*) \,|\, \nabla_w P(w^*)_{(i)} > 0 \,\right\}. \tag{3.17b}$$

Several key properties result from the EDPF formulation (3.14). First, the EDPF problem can be solved as an *NLP with box constraints*. As we see in section 5, this enables the use of trust-region Newton techniques and gradient projection [31] in order to obtain global and superlinear convergence and fast identification of activity. Second, the objective of (3.14) is a *natural merit function*. Improvement of the objective value implies progress on the Lagrangian, primal infeasibility, or dual infeasibility. Consequently, similar to Fletcher's AL [22], direct minimization of the EDPF can in principle avoid the Maratos effect. To obtain these desirable features, we first analyze the properties of the derivatives of the EDPF and establish conditions for parameters α, β for which minimizers of (3.14) coincide with those of the NLP (1.1).

4. Properties of exact penalty formulation. Throughout this section we simplify notation as $P := P_{\alpha,\beta}(w)$, $\nabla\mathcal{L} := \nabla\mathcal{L}(w)$, $K := K_{\alpha,\beta}(w)$. We thus write the EDPF (3.12) as

$$P = \mathcal{L} + \frac{1}{2}\nabla\mathcal{L}^T K \nabla\mathcal{L}. \tag{4.1}$$

We also define $\nabla P = \nabla_w \mathcal{P}$, $\nabla^2 P = \nabla_{w,w} P$, and a diagonal matrix $\Gamma \in \mathcal{R}^{n_w \times n_w}$ with entries $\Gamma_{i,i} = 4\beta,\ i = 1, \ldots, n$, and $\Gamma_{i,i} = 0,\ i = n+1, \ldots, n_w$. Using these definitions, we have

$$\begin{aligned} \nabla P &= \nabla \mathcal{L} + \frac{1}{2}\nabla(K\nabla\mathcal{L})\nabla\mathcal{L} + \frac{1}{2}\nabla^2\mathcal{L}K\nabla\mathcal{L} \\ &= \nabla\mathcal{L} + \nabla^2\mathcal{L}K\nabla\mathcal{L} + \frac{1}{2}\Gamma \mathrm{diag}(\nabla\mathcal{L})\nabla\mathcal{L}, \end{aligned} \tag{4.2}$$

where we use the property

$$\nabla(K\nabla\mathcal{L}) = \nabla^2\mathcal{L}K + \Gamma \mathrm{diag}(\nabla\mathcal{L}).$$

For notational convenience, we define the expanded null-space matrix $N := N(w) \in \mathcal{R}^{n_w \times d}$ corresponding to the subspace $\mathcal{S}_P(w)$ with structure,

$$N = \begin{bmatrix} N_x & \\ & \mathbb{I}_m \end{bmatrix}. \tag{4.3}$$

Here, the lower corner is the identity matrix because the multipliers λ are free and $N_x = N_x(x)$ is the null-space matrix defined in section 2.

We now prove that there exist parameters α, β for which, if w^* satisfies the KKT conditions of (3.14), then it satisfies the KKT conditions of the NLP (1.1). We first need the following property.

PROPERTY 1. *At any point w^* satisfying* (A1a) *and the KKT conditions of the NLP* (1.1) *we have that*

$$\begin{aligned} &\nabla_x P(w^*) = \nabla_x \mathcal{L}(w^*) + 2\beta\, diag(\nabla_x \mathcal{L}(w^*))\nabla_x \mathcal{L}(w^*), \\ &\nabla_\lambda P(w^*) = 0_m. \end{aligned}$$

Proof. The result follows because $K_{\alpha,\beta}(w^*)\nabla_w \mathcal{L}(w^*) = 0_{n_w}$, since $4\beta X^* \nabla_x \mathcal{L}(w^*) = 0_n$ and $\nabla_\lambda \mathcal{L}(w^*) = h(x^*) = 0_m$. ∎

We now prove the first-order exactness of P. This is an extension of [6, Proposition 4.15] to the case of inequality constraints.

THEOREM 4.1. *Let $\mathcal{K} \subset \mathcal{R}^n_+ \cap \mathcal{R}^m$ be a compact set such that LICQ* (A2a) *holds for any first component x of $w = (x, \lambda) \in \mathcal{K}$ and* (A1a) *holds. There exist a scalar $\bar{\beta} > 0$ and, for each $\beta \in (0, \bar{\beta}]$, a scalar $\bar{\alpha}(\beta)$ such that for all $\alpha \geq \bar{\alpha}(\beta)$, if $w^* \in \mathcal{K}$ satisfies the KKT conditions of the EDPF problem* (3.14)*, then w^* satisfies the KKT conditions for the NLP* (1.1).

Proof. To simplify notation we write $X = X^*$. From (3.15) we have that a point satisfying the KKT conditions for (3.14) satisfies $0 \leq x^* \perp \nabla_x P(x^*, \lambda^*) \geq 0$, which implies $\sqrt{X}\nabla_x P(x^*, \lambda^*) = 0_n$. We thus have

$$\begin{aligned} 0_n &= \sqrt{X}\nabla_x P(w^*) \\ &= \sqrt{X}\nabla_x \mathcal{L}(w^*) + \alpha\sqrt{X}\nabla_x h(x^*)h(x^*) + 4\beta\sqrt{X}\nabla_{x,x}\mathcal{L}(w^*)\sqrt{X}\sqrt{X}\nabla_x\mathcal{L}(w^*) \\ &\quad + 2\beta\sqrt{X}\mathrm{diag}\left(\nabla_x\mathcal{L}(w^*)\right)\nabla_x\mathcal{L}(w^*), \\ 0_m &= \nabla_\lambda P(w^*) = h(x^*) + 4\beta\nabla_x h(x^*)^T\sqrt{X}\sqrt{X}\nabla_x\mathcal{L}(w^*). \end{aligned}$$

We need to show that these two conditions imply $h(x^*) = 0_m$ and $X\nabla_x\mathcal{L}(w^*) = 0_n$,

the KKT conditions of the NLP. We write the above system as

$$\begin{bmatrix} \mathbb{I}_{n\times n} + 4\beta\sqrt{X}\nabla_{x,x}\mathcal{L}(w^*)\sqrt{X} + 2\beta\mathrm{diag}\left(\nabla_x\mathcal{L}(w^*)\right) & \alpha\sqrt{X}\nabla_x h(x^*) \\ 4\beta\nabla_x h(x^*)^T\sqrt{X} & \mathbb{I}_{m\times m} \end{bmatrix} \cdot \begin{bmatrix} \sqrt{X}\nabla_x\mathcal{L}(w^*) \\ h(x^*) \end{bmatrix} = \begin{bmatrix} 0_n \\ 0_m \end{bmatrix}. \tag{4.4}$$

The result follows if the matrix on the left-hand side is nonsingular. We define

$$G_1 := \mathbb{I}_{n\times n} + 4\beta\sqrt{X}\nabla_{x,x}\mathcal{L}(w^*)\sqrt{X} + 2\beta\mathrm{diag}\left(\nabla_x\mathcal{L}(w^*)\right). \tag{4.5}$$

Let $\bar{\beta} > 0$ be such that, for all $\beta \in (0, \bar{\beta}]$, $G_1 \succ 0$ for all $w^* \in \mathcal{K}$. In this case, G_1 is nonsingular, and we can apply Schur elimination to obtain

$$\sqrt{X}\nabla_x\mathcal{L}(w^*) = -\alpha Q^{-1}\sqrt{X}\nabla_x h(x^*)h(x^*), \tag{4.6}$$

and

$$\begin{aligned} 0_m &= h(x^*) - 4\alpha\,\beta\nabla_x h(x^*)\sqrt{X}Q^{-1}\sqrt{X}\nabla_x h(x^*)h(x^*) \\ &= -Q_2 h(x^*), \\ Q_2 &:= 4\alpha\beta\nabla_x h(x^*)^T\sqrt{X}Q^{-1}\sqrt{X}\nabla_x h(x^*) - \mathbb{I}_{m\times m}. \end{aligned} \tag{4.7}$$

Because we assume that LICQ (A2a) holds for all x^* (the first component of $w^* \in \mathcal{K}$), it follows that $\sqrt{X}\nabla_x h(x^*)$ is full column rank and thus $\nabla_x h(x^*)^T\sqrt{X}Q^{-1} \cdot \sqrt{X}\nabla_x h(x^*) \succ 0$. In turn, for all $\beta \in (0, \bar{\beta}]$ we can choose $\bar{\alpha}(\beta)$ such that for all $\alpha \geq \bar{\alpha}(\beta)$ and $\alpha\beta$ sufficiently large we have that $Q_2 \succ 0$. This implies $h(x^*) = 0$ and $X\nabla_x\mathcal{L}(w^*) = 0_n$. Now we choose $\bar{\beta}$ such that we also have, in addition to $G_1 \succ 0$, that

$$1 + 2\beta\nabla_x\mathcal{L}(w^*)_{(i)} \geq 0 \quad \forall\beta \in (0, \bar{\beta}], \quad i = 1, \ldots, n, \quad \forall w^* \in \mathcal{K}. \tag{4.8}$$

From Property 1 we have $\nabla_x P(w^*)_{(i)} = \nabla_x\mathcal{L}(w^*)_{(i)} + 2\beta(\nabla_x\mathcal{L}(w^*)_{(i)})^2 = \nabla_x\mathcal{L}(w^*)_{(i)} \cdot (1 + 2\beta\nabla_x\mathcal{L}(w^*)_{(i)})$. Because from the optimality conditions of (3.14) we have that $\nabla_x P(w^*)_{(i)} \geq 0$, the previous relationship implies that $\nabla_x\mathcal{L}(w^*)_{(i)} \geq 0$. Combined with $\sqrt{X}\nabla_x\mathcal{L}(w^*) = 0$ and $h(x^*) = 0$, this means precisely that w^* is a KKT point of (1.1). The proof is complete. ∎

Condition $G_1 \succ 0$ indicates that sufficiently small β exists to make the matrix in the upper left corner positive definite. Condition $Q_2 \succ 0$ indicates that for this β, a sufficiently large α exists to make the matrix on the left-hand side positive definite. Consequently, we can see that stationary points of the EDPF coincide with stationary points of the NLP as $\alpha \to +\infty$, $\beta \to 0$, and $\alpha\beta \to +\infty$. Our analysis extends the results of Di Pillo and Grippo [16] for the case in which the exact penalty includes inequalities. Our proof follows along the lines of Proposition 4.15 provided by Bertsekas [6] for equality-constrained problems. We note that the upper left matrix in the inequality case becomes $\mathbb{I}_{n\times n} + 4\beta\sqrt{X}\nabla_{x,x}\mathcal{L}(w^*)\sqrt{X} + 2\beta\mathrm{diag}\left(\nabla_x\mathcal{L}(w^*)\right)$, as opposed to $\mathbb{I}_{n\times n} + \beta\nabla_{x,x}\mathcal{L}(w^*)$ for the equality-constrained case. For the equality-constrained case Bertsekas observed that β can be any nonnegative value if $\nabla_{x,x}\mathcal{L}(w^*)$ is positive definite. In our case, however, even if $\nabla_{x,x}\mathcal{L}(w^*)$ is positive definite, we require β to be sufficiently small to enforce $\nabla_x\mathcal{L}(w^*)_{(i)} \geq 0$ for $x_{(i)} = 0$, which restricts β.

This result also implies that the solution of the EDPF problem identifies the optimal active set of the NLP. Consequently, $\mathcal{A}(x^*) = \mathcal{A}_P(w^*)$ because λ^* is free. This also implies that the upper n_d^* rows of the null-space matrix $N(w^*)$ defined for $\mathcal{S}_P(w^*)$ correspond to the null-space matrix $N_x(x^*)$ defined for $\mathcal{S}(x^*)$. We emphasize that, given a solution w^* for the NLP and corresponding null-space matrix $N_x(x^*)$, we can construct an expanded null-space matrix $N(w^*)$ (4.3). This observation will be needed in the following analysis.

We now connect the SSOC of the NLP to those of the EDPF problem. If $f(\cdot)$ and $h(\cdot)$ are three times continuously differentiable (A1a), the Hessian of the EDPF exists. The Hessian of the EDPF involves a tensor because of the appearance of the gradient of the Lagrangian in the EDPF. To enable notation in the Hessian derivation, we use the fact that

$$\nabla^2 P \cdot u = \nabla(\nabla P^T \cdot u) \tag{4.9}$$

for a constant vector $u \in \mathcal{R}^{n_w}$. We have

$$\nabla P^T \cdot u = \nabla \mathcal{L}^T \cdot u + \nabla \mathcal{L}^T K \nabla^2 \mathcal{L} \cdot u + \frac{1}{2} \nabla \mathcal{L}^T \Gamma \mathrm{diag}(\nabla \mathcal{L}) \cdot u \tag{4.10}$$

and

$$\nabla^2 P \cdot u = \nabla^2 \mathcal{L} \cdot u + \nabla(\nabla \mathcal{L}^T K \nabla^2 \mathcal{L} \cdot u) + \frac{1}{2} \nabla(\nabla \mathcal{L}^T \Gamma \mathrm{diag}(\nabla \mathcal{L}) \cdot u). \tag{4.11}$$

Expanding terms, we obtain

$$\begin{aligned} \nabla(\nabla \mathcal{L}^T K \nabla^2 \mathcal{L} \cdot u) &= \nabla(K \nabla \mathcal{L}) \nabla^2 \mathcal{L} \cdot u + \nabla(\nabla^2 \mathcal{L} \cdot u) K \nabla \mathcal{L} \\ &= \nabla^2 \mathcal{L} K \nabla^2 \mathcal{L} \cdot u + \Gamma \mathrm{diag}(\nabla \mathcal{L}) \nabla^2 \mathcal{L} \cdot u + \nabla(\nabla^2 \mathcal{L} \cdot u) K \nabla \mathcal{L} \end{aligned} \tag{4.12}$$

and

$$\frac{1}{2} \nabla(\nabla \mathcal{L}^T \Gamma \mathrm{diag}(\nabla \mathcal{L}) \cdot u) = \nabla^2 \mathcal{L} \mathrm{diag}(\nabla \mathcal{L}) \Gamma \cdot u. \tag{4.13}$$

Merging terms,

$$\begin{aligned} \nabla^2 P \cdot u = {} & \nabla^2 \mathcal{L} \cdot u + \nabla^2 \mathcal{L} K \nabla^2 \mathcal{L} \cdot u \\ & + \nabla^2 \mathcal{L} \mathrm{diag}(\nabla \mathcal{L}) \Gamma \cdot u + \Gamma \mathrm{diag}(\nabla \mathcal{L}) \nabla^2 \mathcal{L} \cdot u + \nabla(\nabla^2 \mathcal{L} \cdot u) K \nabla \mathcal{L}. \end{aligned} \tag{4.14}$$

From this expression we derive the following properties for the Hessian and reduced Hessian matrices of the EDPF.

PROPERTY 2. *At any point w^* satisfying* (A1b) *and the KKT conditions of the NLP* (1.1) *with corresponding expanded null-space matrix $N := N(w^*)$ we have that*

$$\nabla_w^2 P(w^*) = \nabla^2 \mathcal{L} + \nabla^2 \mathcal{L} K \nabla^2 \mathcal{L} + \nabla^2 \mathcal{L} \mathit{diag}(\nabla \mathcal{L}) \Gamma + \Gamma \mathit{diag}(\nabla \mathcal{L}) \nabla^2 \mathcal{L}, \tag{4.15a}$$

$$N^T \nabla_w^2 P(w^*) N = N^T \nabla^2 \mathcal{L} N + N^T \nabla^2 \mathcal{L} K \nabla^2 \mathcal{L} N. \tag{4.15b}$$

Proof. Equation (4.15a) follows because $K_{\alpha,\beta}(w^*) \nabla_w \mathcal{L}(w^*) = 0_{n_w}$. Equation (4.15b) follows because $N^T \Gamma \mathrm{diag}(\nabla \mathcal{L}) = 0_{n_d \times n}$ and $\Gamma \mathrm{diag}(\nabla \mathcal{L}) N = 0_{n \times n_d}$. ∎

The high-order term $\nabla(\nabla_w^2\mathcal{L}(w^*)\cdot u)K_{\alpha,\beta}(w^*)\nabla_w\mathcal{L}(w^*)$ vanishes at a point w^* satisfying the KKT conditions of the NLP, thus suggesting (4.15a) as a natural Hessian approximation at nearby points, which we define as

(4.16)
$$Q(w) = \nabla^2\mathcal{L}(w) + \nabla^2\mathcal{L}(w)K(w)\nabla^2\mathcal{L}(w) + \nabla^2\mathcal{L}(w)\mathrm{diag}(\nabla\mathcal{L})\Gamma + \Gamma\mathrm{diag}(\nabla\mathcal{L})\nabla^2\mathcal{L}(w).$$

The corresponding reduced Hessian is $N^TQ(w)N$. In section 6 we explain how these properties can be exploited to enable matrix-free implementations.

Another crucial implication resulting from (4.14) and Property 2 is that the Hessian approximation $Q(w)$ satisfies the Dennis–Moré condition [15]:

$$\begin{aligned}(Q(w) - \nabla^2P(w))\cdot u &= \nabla(\nabla^2\mathcal{L}(w)\cdot u)K(w)\nabla\mathcal{L}(w)\\ &\overset{w\to w^*}{=} 0_{n_w}.\end{aligned} \tag{4.17}$$

This follows from the fact that the third derivatives of the Lagrange function are uniformly bounded and that, from (A1b), the term $K(w)\nabla\mathcal{L}(w)$ is continuously differentiable and thus Lipschitz. In addition, it satisfies the condition $K_{\alpha,\beta}(w^*)\nabla_w\mathcal{L}(w^*) = 0_{n_w}$. We thus obtain that

$$\begin{aligned}\nabla(\nabla^2\mathcal{L}(w)\cdot u)K(w)\nabla\mathcal{L}(w) &= O(u)O(||w-w^*||).\\ &\overset{w\to w^*}{=} 0_{n_w}.\end{aligned} \tag{4.18}$$

The result is stronger than the classical Dennis–Moré condition because convergence is independent of the direction u. This result will become important in establishing superlinear convergence results in section 5. We now prove that there exist α, β such that the reduced Hessian matrix of the EDPF is positive definite at w^* if the SSOC are satisfied for the NLP at this point. We also show that there exist α, β for which positive definiteness of the reduced Hessian at a point w^* satisfying the KKT conditions of the EDPF problem (3.14) implies that this is a strict local minimum of the NLP. In this sense and under these conditions, the EDPF is exact. That is, the EDPF has local minimizers where the original NLP (1.1) does, and it will not introduce any additional local minimizers. This is a key advantage over the squared-slacks reformulation (3.1).

THEOREM 4.2. *If w^* satisfies the KKT conditions of the NLP* (1.1) *and the SSOC* (2.10) (A3), *then for every $\beta > 0$ there exists $\bar\alpha(\beta) > 0$ such that for all $\alpha \geq \bar\alpha(\beta)$, w^* satisfies the KKT conditions of the EDPF problem* (3.14), *and the reduced Hessian*

$$H_P = N(w^*)^T\nabla^2P_{\alpha,\beta}(w^*)N(w^*) \tag{4.19}$$

is positive definite. Furthermore, assume that w^* satisfies the KKT conditions of the NLP (1.1) *but not the SSOC* (2.10) *and that the reduced Hessian of the Lagrangian matrix associated with the NLP* (1.1) *has at least one negative eigenvalue. Then, there exists $\bar\beta > 0$ such that for each $\beta \in (0, \bar\beta]$ and $\alpha > 0$, H_P has at least one negative eigenvalue.*

Proof. We define $H = \nabla_{x,x}\mathcal{L}(w^*)$ and $A = \nabla_x h(x^*)^T$. We construct $\nu^TH_P\nu$ using $N = N(w^*)$ constructed from w^* and with $\nu^T = [\nu_x^T\ \nu_\lambda^T]$, where $\nu \in \mathcal{R}^{d^*}$ and $\nu_x \in \mathcal{R}^{n_d^*}$. Because w^* satisfies the SSOC (2.10), we have that $\nu_x^TN_x^THN_x\nu_x > 0$ for all $N_x\nu_x \neq 0$ and $AN_x\nu_x = 0_m$. Using Property 2, we have that $\nu^TH_P\nu$ has the

following simplified form:

$$\nu^T H_P \nu = \begin{bmatrix} \nu_x^T N_x^T & \nu_\lambda^T \end{bmatrix} \begin{bmatrix} H & A^T \\ A & \end{bmatrix} \begin{bmatrix} N_x \nu_x \\ \nu_\lambda \end{bmatrix} + \begin{bmatrix} \nu_x^T N_x^T & \nu_\lambda^T \end{bmatrix} \begin{bmatrix} H & A^T \\ A & \end{bmatrix} \begin{bmatrix} 4\beta X & 0 \\ 0 & \alpha \mathbb{I}_m \end{bmatrix} \begin{bmatrix} H & A^T \\ A & \end{bmatrix} \begin{bmatrix} N_x \nu_x \\ \nu_\lambda \end{bmatrix}. \tag{4.20}$$

Expanding terms, we have

$$\nu^T H_P \nu = \left[\nu_x^T N_x^T H N_x \nu_x + 2\nu_\lambda^T A N_x \nu_x + 4\beta (H N_x \nu_x + A^T \nu_\lambda)^T X (H N_x \nu_x + A^T \nu_\lambda)\right] + \alpha \nu_x^T N_x^T A^T A N_x \nu_x. \tag{4.21}$$

If the SSOC and LICQ (implied by our definition of SSOC) hold, the first term in brackets is positive for all $N_x \nu_x \neq 0$, and we have $\nu_x^T N_x^T A^T A N_x \nu_x = 0$. Under this condition and by Debreu's lemma [23] we have that there exists $\bar{\alpha}(\beta)$ such that for all $\beta \geq 0$ and $\alpha \geq \bar{\alpha}(\beta)$, H_P is positive definite. To establish the second result, let $N_x \nu_x$ satisfy $A N_x \nu_x = 0_m$ but $\nu_x^T N_x^T H N_x \nu_x < 0$ (it does not satisfy the SSOC for the NLP (1.1)). We have

$$\nu^T H_P \nu = \nu_x^T N_x^T H N_x \nu_x + 4\beta \nu_x^T N_x^T H X H N_x < 0$$

for any $\beta < \frac{1}{4}(\nu_x^T N_x^T H N_x \nu_x)/(\nu_x^T N_x^T H X H N_x \nu_x)$, which implies that w^* is not a local minimizer of (3.14). The proof is complete. ∎

This result implies that a point w^* satisfying the SSOC for the NLP (1.1) satisfies the SSOC (3.16) for the EDPF problem (3.14) if α, β are chosen sufficiently large and sufficiently small, respectively. Our proof follows along the lines of Proposition 4.16 in [6], but this is extended to deal with inequality constraints. An important finding resulting from our analysis of the structure of the reduced Hessian (4.15b) is that the introduction of inequalities does not alter the positive curvature of the EDPF. Positive definiteness of the reduced Hessian is also important because this enables the use of PCG to compute truncated Newton steps.

5. Trust-region Newton. The results of the previous section indicate that the reduced Hessian of the EDPF is positive definite close to a solution satisfying SSOC. From the structure of the Hessian (4.14), however, it is clear that negative curvature is likely to occur along the search. Therefore, implementations based on Hessian modifications would be inefficient. We propose to apply a trust-region Newton framework to the EDPF problem (3.14) following the developments of Lin and Moré [31]. In our analysis, we assume that α, β are chosen such that conditions and results of Theorems 4.1 and 4.2 hold. Consequently, we simplify the notation as $P(\cdot) := P_{\alpha,\beta}(\cdot)$. Lin and Moré's convergence analysis relies on the exact Hessian of the objective function. We have extended their results to the particular case of the approximate Hessian (4.15a) and prove that this approximation maintains superlinear convergence.

We write (3.14) as

$$\min_w \quad P(w) \quad \text{s.t.} \quad w \in \Omega, \tag{5.1}$$

where $P(\cdot) := P_{\alpha,\beta}(\cdot)$ and $\Omega := \{w \,|\, w \geq l\} \subseteq \mathcal{R}^{n_w}$, where $l_{(i)} = 0,\ i = 1, \ldots, n$, and $l_{(i)} = -\infty,\ i = n+1, \ldots, n_w$. We also define the projection operator (onto the

bounds) $\text{Proj} : \mathcal{R}^{n_w} \to \Omega$ and the projected gradient as $g_{\text{Proj}}(w)$, which is computed as

$$g_{\text{Proj}}(w)_{(i)} := \begin{cases} \nabla_w P(w)_{(i)} & \text{if} \quad w_{(i)} > 0, \\ \min(\nabla_w P(w)_{(i)}, 0) & \text{if} \quad w_{(i)} = 0, \end{cases} \qquad i = 1, \ldots, n_w. \tag{5.2}$$

For formal definitions, see section 2 in [31]. Lin and Moré use the concept of exposed faces to characterize the active set. We have that the face of the convex set Ω exposed by the negative gradient $-\nabla_w P(w)$ is given by

$$E[-\nabla_w P(w)] = \left\{ w \in \Omega \,|\, w_{(i)} = 0 \quad \text{if} \quad \nabla_w P(w)_{(i)} > 0 \right\}. \tag{5.3}$$

Burke and Moré [9] showed that w^* is a stationary point of (5.1) if and only if $w^* \in E[-\nabla_w P(w^*)]$. We also have that if SC holds, $w \in E[-\nabla_w P(w^*)] \iff \mathcal{A}_P(w^*) \subset \mathcal{A}_P(w)$. Consequently, throughout this section we assume that SC (A2b) holds at any statitionary point w^*. As shown in Theorem 4.2, there exist α, β such that a point w^* satisfying SSOC for the original NLP also satisfies the SSOC for (5.1). In the following analysis, we use the following form of SSOC for (5.1),

$$\exists\, \kappa > 0 \text{ such that } \ \nu^T \nabla^2 P(w^*) \nu \geq \kappa \|\nu\|^2, \quad \nu \in \mathcal{S}_P(w^*)\,. \tag{5.4}$$

The existence of such κ follows from the SSOC (3.16) applied to (5.1) and the fact that $\nu^T \nabla^2 P(w^*) \nu$ is bounded below on the unit sphere intersected with $\mathcal{S}_P(w^*)$. In a trust-region framework, we have at each iteration $w^k \in \Omega$ a radius Δ^k and a quadratic *model* $\Psi^k : \mathcal{R}^{n_w} \to \mathcal{R}$ of the actual reduction $P(w^k + s) - P(w^k)$:

$$\Psi^k(s) = {g^k}^T s + \frac{1}{2} s^T Q^k s. \tag{5.5}$$

Here, Q^k is the approximation of the exact penalty Hessian $\nabla_w^2 P(w^k)$ given in (4.16), $g^k := \nabla_w P(w^k)$ is the gradient, and g^k_{Proj} is the projected gradient. We also define $q^k(w) := \Psi^k(w - w^k)$. Here, we recall that $P(\cdot)$ is twice continuously differentiable if $f(\cdot)$ and $h(\cdot)$ in (1.1) are three times continuously differentiable. We thus make assumption (A1b) thoughout this section to guarantee the existence of g^k and Q^k.

Given a step s^k such that $w^k + s^k \in \Omega$ and $\Psi^k(s^k) < 0$, the trust-region bound is updated according to the reduction ratio

$$\rho^k = \frac{P(w^k + s^k) - P(w^k)}{\Psi^k(s^k)}. \tag{5.6}$$

Because the step is chosen such that $\rho^k > 0$, $\Psi^k(s^k)$ indicates a reduction of the actual function. The iterate is updated by using ρ^k according to

$$w^{k+1} = \begin{cases} w^k + s^k & \text{if} \quad \rho^k > \eta_0, \\ w^k & \text{if} \quad \rho^k \leq \eta_0 \end{cases} \tag{5.7}$$

for $\eta_0 > 0$. The trust-region bound is updated according to the following rule:

$$\Delta^{k+1} \in \begin{cases} \left[\sigma_1 \cdot \min\{\|s^k\|, \Delta^k\}, \sigma_2 \Delta^k\right] & \text{if} \quad \rho^k \leq \eta_1, \\ \left[\sigma_1 \Delta^k, \sigma_3 \Delta^k\right] & \text{if} \quad \rho^k \in (\eta_1, \eta_2), \\ \left[\Delta^k, \sigma_3 \Delta^k\right] & \text{if} \quad \rho^k \geq \eta_2, \end{cases} \tag{5.8}$$

where $0 < \eta_1 < \eta_2 < 1$ and $0 < \sigma_1 < \sigma_2 < 1 < \sigma_3$.

To obtain the search step s^k, we first compute the Cauchy step s_c^k by performing projected line searches on the step size $\kappa_c \in [0,1]$,

$$s_c^k = \text{Proj}(w^k - \kappa_c \nabla_w P(w^k)) - w^k, \tag{5.9}$$

until the following conditions are satisfied:

$$\Psi^k(s_c^k) \leq \mu_0 \nabla_w P(w^k)^T s_c^k, \qquad \|s_c^k\| \leq \mu_1 \Delta^k, \tag{5.10}$$

with $\mu_0, \mu_1 > 0$. These requirements on the Cauchy step can be satisfied with a finite number of evaluations [33]. The projected search also gives the active set guess $\mathcal{A}_P^k := \mathcal{A}_P(w_c^k)$ with $w_c^k = w^k + s_c^k$. We define the inactive set $\mathcal{I}^k = \mathcal{I}(w_c^k)$ such that $\mathcal{I}^k \cup \mathcal{A}^k = \{1..n_w\}$ and $\mathcal{I}^k \cap \mathcal{A}^k = \emptyset$. An important property is that if any sequence w^k converges to a stationary point w^* and w^k lands on $E[-\nabla_w P(w^*)]$, then the Cauchy point $w_k + s_c^k$ remains on $E[-\nabla_w P(w^*)]$ [31, Theorem 3.2].

We now seek a step s^k that improves the Cauchy step in the sense that

$$\Psi^k(s^k) \leq \mu_0 \Psi^k(s_c^k), \quad w^k + s^k \in \Omega. \tag{5.11}$$

Given this step, we check the reduction ratio (5.6), accept or reject the step (5.7), and update the trust-region radius (5.8). This gives a basic trust-region method. Theorem 3.3 in [31] states that if the steps s^k generated by the trust-region method satisfy

$$\mathcal{A}_P(w_c^k) \subset \mathcal{A}_P(w^k + s^k) \tag{5.12}$$

for $k \geq 0$, then if $\{w^k\}$ converges to w^*, there is an index k_0 such that for $k \geq k_0$ we have $w^k \in E[-\nabla_w P(w^*)]$ and $w^k + s^k \in E[-\nabla_w P(w^*)]$. This property is essential to guarantee local convergence.

We refine the Cauchy point w_c^k while satisfying conditions (5.11) and (5.12) in order to preserve global convergence. Specifically, Lin and Moré propose the following procedure. At each *major* iteration k we compute *minor* iterates $w_0^k, \ldots, w_{\ell+1}^k$ with $w_0^k := w_c^k$ and $w^k := w_{\ell+1}^k$. For each minor iterate $j = 0, \ldots, \ell + 1$ we require that

$$w_j^k \in \Omega, \quad \mathcal{A}_P(w_c^k) \subset \mathcal{A}_P(w_j^k), \quad \|w_j^k - w^k\| \leq \mu_1 \Delta^k, \tag{5.13}$$

and

$$q^k(w_{j+1}^k) \leq q^k(w_j^k) + \mu_0 \cdot \min\left\{\nabla_w q^k(w^k)^T (w_{j+1}^k - w_j^k)\ ,\ 0\right\}. \tag{5.14}$$

To compute steps satisfying these conditions, we use the trust-region quadratic program

$$\min_{s^k} \ {g^k}^T s^k + \frac{1}{2} {s^k}^T Q^k s^k \tag{5.15a}$$

$$\text{s.t. } s_{(i)}^k = 0,\ i \in \mathcal{A}_P(w_j^k), \tag{5.15b}$$

$$\|D^k s^k\| \leq \Delta^k, \tag{5.15c}$$

where D^k is a preconditioning matrix for Q^k. Descent directions can be obtained by using Steihaug's PCG approach [43]. At each minor iterate j, we define the null-space

matrix $N_j^k \in \mathcal{R}^{n\times d_j^k}$ consistent with $\mathcal{A}_P(w_j^k)$ and the reduced-space step $s_d^k \in \mathcal{R}^{d_j^k}$ so that $s_j^k = N_j^k s_d^k$. The quadratic program takes the form

$$\min_{s_d^k} g_d^{k^T} s_d^k + \frac{1}{2} s_d^{k^T} Q_d^k s_d^k \tag{5.16a}$$

$$\text{s.t. } \|D^k N_j^k s_d^k\| \le \Delta^k. \tag{5.16b}$$

Here $Q_d^k = N_j^{k^T} Q^k N_j^k$, $g_d^k = N_j^{k^T} g^k$ are the reduced approximate Hessian and reduced gradient, respectively. The PCG search iterates $i > 0$ are terminated by generating step $s_j^k := N_j^k s_{d,i}^k$ if

(C1) a descent step is found,
(C2) the step hits the trust-region radius, or
(C3) negative curvature is detected.

A final step $s_{\ell+1}^k$ is considered successful from a global convergence point of view if it satisfies (5.11). This condition can be enforced by performing projected searches along each step s_j^k,

$$s_j^k \leftarrow \text{Proj}(w^k + \kappa s_j^k) - w^k, \tag{5.17}$$

where $\kappa \in [0,1]$ is the step length, and stopping when (5.13) and (5.14) are satisfied. The procedure terminates with a final step $s^k = s_{\ell+1}^k = w_{\ell+1}^k - w^k$ satisfying (5.11).

To enable superlinear convergence, we require the step $s^k := w_{\ell+1}^k - w^k$ to satisfy

$$\left\|(N_{\ell+1}^k)^T \left[g^k + Q^k s^k\right]\right\| \le \xi^k \|(N_{\ell+1}^k)^T g^k\|, \quad w^k + s^k \in \Omega. \tag{5.18}$$

To satisfy this condition simultaneously with (5.13)–(5.14), Lin and Moré proposed the following approach. Close to the solution, the trust-region constraint is inactive, and directions of negative curvature are not encountered. Consequently, we require that at each minor iterate w_j^k, $j = 0, \ldots, \ell+1$, is obtained by terminating the PCG search at the minimizer of $q^k(\cdot)$. This approach amounts to assuming that $\xi^k = 0$. This gives $s^k = w_{j+1}^k - w^k$. We perform a line search along this step to stay inside Ω, in order to satisfy (5.14), and such that $\mathcal{A}_P(w_{j+1}^k)$ has at least one more active variable than $\mathcal{A}_P(w_j^k)$ such that

$$\mathcal{A}_P(w_j^k) \subset \mathcal{A}_P(w_{j+1}^k). \tag{5.19}$$

This procedure implicitly satisfies (5.13). In the worst case, the procedure terminates with all variables active and for which condition $w^k + s^k \in \Omega$ is satisfied trivially.

The procedure gives the major step s^k. We next compute the reduction (5.6). If the first condition in (5.7) is satisfied, we accept the step $w^{k+1} = w^k + s^k$; otherwise it is rejected. We then update the trust-region radius Δ^{k+1} following (5.8). The trust-region Newton (TRN) algorithm is summarized below.

Trust-Region Newton Algorithm

Assume given algorithmic parameters $\eta_0, \eta_1, \eta_2, \sigma_1, \sigma_2, \mu_0, \mu_1$, and ℓ.

1. **Initialization.** Start with w^0 and Δ^0 at $k = 0$. DO for $k > 0$:
2. **Major Step Test.** If w^k is a stationary point, STOP.
3. **Cauchy Search.** Compute Cauchy step s_k^c and active set $\mathcal{A}_P(w_c^k)$ from (5.9), and perform line search until conditions (5.10) are satisfied. Define step $w_c^k \leftarrow w^k + s_c^k$.

4. **Refinement Search.** Start at $w_0^k := w_c^k$. DO for $j = 0, \ldots, \ell + 1$:
   4.1. *PCG Step.* At $\mathcal{A}_P(w_j^k)$ apply Steihaug's PCG search on (5.16), and terminate with s_d^k if either (C1), (C2), or (C3) holds.
   4.2. *Minor Step Test.* Set $w_{j+1}^k \leftarrow w_j^k + N_j^k s_d^k$. If improvement over Cauchy (5.10) and (5.12) is satisfied, STOP refinement search, set $s^k = w_{j+1}^k - w^k$, and GO TO 5.
   4.3. *Update Minor Step.* Cut step to satisfy (5.13), (5.14), and (5.19), update w_{j+1}^k, set $j \leftarrow j + 1$, and RETURN TO 4.1.
5. **Update Step**. Compute reduction ratio (5.6) update step w^{k+1} according to (5.7), and update the trust-region radius Δ^{k+1} according to (5.8). Set $k \leftarrow k + 1$, and RETURN TO 2.

We now establish convergence results for the trust-region Newton algorithm. We start by establishing global convergence. The following result is an adaptation of the result of Burke, Moré, and Toraldo [8] (see also Theorem 2.1 in [31]) applied to the EDPF problem.

THEOREM 5.1. *Let a sequence of approximate Hessians $\{Q^k\}$ of the quadratic model* (5.5) *be uniformly bounded.* (i) *If w^* is a limit point of the sequence $\{w^k\}$ generated by the trust-region Newton algorithm, then there is a subsequence $\{w^{k_i}\}$ of successful steps that converges to w^* with*

$$\lim_{i \to \infty} \|g_{Proj}(w_c^{k_i})\| = 0. \tag{5.20}$$

In addition, $\{w_c^{k_i}\}$ converges to w^*, and thus w^* is a stationary point for problem (5.1). *Moreover,* (ii) *if the assumptions and conditions of EDPF parameters α, β of Theorems* 4.1 *and* 4.2 *hold, then w^* is also stationary for the NLP* (1.1).

Proof. Result (i) follows from the proof of Theorem 5.5 in [8]. Result (ii) follows from Theorems 4.1 and 4.2 because we have that there exist α, β such that a stationary point w^* of the EDPF problem is also a stationary for the NLP. ∎

For each w^{k_i} we can define a Cauchy point $w_c^{k_i} := \text{Proj}(w^{k_i} + \kappa_c^k s_c^k)$. The result says that there exists a subsequence $\{w^{k_i}\}$ converging to a limit point w^* that satisfies $\lim_{i \to \infty} \|g_{Proj}(w_c^{k_i})\| = 0$. Moreover, the sequence $\{w_c^{k_i}\}$ converges to the limit point w^*. Consequently, we have $\lim_{i \to \infty} \|g_{Proj}(w^*)\| = 0$ and thus the limit point w^* is also a stationary point. Details of this result can be found in [8]. The result thus implies that every limit point of the sequence $\{w^k\}$ is a stationary point of problem (5.1).

From Theorems 4.1 and 4.2 we have that there exist α, β such that $Q(w)$ is bounded if $\nabla_{w,w} \mathcal{L}(w)$ is bounded. The point w^* satisfies the SSOC of the EDPF problem (5.4) if α, β satisfy conditions of Theorems 4.1 and 4.2. In this case we also have that the limit point w^* of the trust-region Newton method is a strict local minimum of the original NLP.

To establish rate of convergence, we first show that if the approximate Hessian $Q(\cdot)$ is used, the limit point of the trust-region radius is bounded away from zero. Our analysis follows that of Lin and Moré [31].

THEOREM 5.2. *Let $\{w^k\}$ be the sequence generated by the trust-region Newton method. Let the assumptions and conditions for the EDPF parameters α, β of Theorems* 4.1 *and* 4.2 *hold. Assume also that $\{w^k\}$ converges to a limit point w^* that satisfies the SSOC of the EDPF* (5.4). *If the minor iterates satisfy* (5.13) *and* (5.14), *then there is an index k_0 such that all steps s^k with $k \geq k_0$ are successful and the trust-region bound Δ^k is bounded away from zero.*

Proof. The proof follows along the lines of the proof of Theorem 5.3 in [31]. We extend this proof by accounting for the Hessian approximation error $\nabla^2 P^k - Q^k$. In the proof we bound $|\rho^k - 1|$, and we show that the bounds converge to zero. This implies that the trust-region update rules eventually accept all steps with Δ^k bounded away from zero. We have that

$$\rho^k - 1 = \frac{P(w^k + s^k) - P(w^k) - \Psi^k(s^k)}{\Psi^k(s^k)} \tag{5.21}$$

and, from Taylor's theorem,

$$P(w^k + s^k) = P(w^k) + \nabla P(w^k)s^k + \frac{1}{2}{s^k}^T \nabla^2 P(w^k + \theta s^k)s^k, \quad \theta \in (0,1), \tag{5.22}$$

so that

$$\begin{aligned} &P(w^k + s^k) - P(w^k) - \Psi^k(s^k) \\ &\quad = \frac{1}{2}{s^k}^T \nabla^2 P(w^k + \theta s^k)s^k - \frac{1}{2}{s^k}^T Q(w^k)s^k \\ &\quad = \frac{1}{2}{s^k}^T \nabla^2 P(w^k + \theta s^k)s^k - \frac{1}{2}{s^k}^T \nabla^2 P(w^k)s^k \\ &\qquad + \frac{1}{2}{s^k}^T \nabla^2 P(w^k)s^k - \frac{1}{2}{s^k}^T Q(w^k)s^k. \end{aligned} \tag{5.23}$$

Taking norms on both sides we obtain,

$$|P(w^k + s) - P(w^k) - \Psi^k(s^k)| = (\sigma_1^k + \sigma_2^k)\|s^k\|^2, \tag{5.24}$$

where the existence of

$$\sigma_1^k = \sup_{\theta \in (0,1)} \left\{ \|\nabla^2 P(w^k + \theta s^k) - \nabla^2 P(w^k)\| \right\} \tag{5.25}$$

follows from Taylor's theorem and $\sigma_2^k = \|\nabla^2 P(w^k) - Q(w^k)\|$. Lemma 5.2 and Theorem 5.3 in [31] show that there exists κ_0 such that $|\rho^k - 1| \leq (\sigma_1^k + \sigma_2^k)/\kappa_0$, so that the result is obtained if $\{\sigma_1^k\}, \{\sigma_2^k\}$ converge to zero.

The sequence $\{\sigma_1^k\}$ converges to zero if $\{s^k\}$ converges to zero because $\{w^k\}$ converges to w^*. Proof of Theorem 5.3 shows that the sequence $\{s^k\}$ converges to zero. Consequently, the sequence $\{\sigma_1^k\}$ converges to zero. As for σ_2^k, we have that for α, β satisfying conditions of Theorems 4.1 and 4.2, the limit point w^* satisfies the KKT conditions of the NLP; and by Property 2 and condition (4.17) we have that $Q(w^k)$ converges to $\nabla^2 P(w^k)$ because $\{w^k\}$ converges to $\{w^*\}$. Consequently, $\{\sigma_2^k\}$ converges to zero. The proof is complete. ∎

We highlight that the convergence result of the Hessian approximation error σ_2^k is stronger than that obtained in a classical quasi-Newton setting because convergence is independent of the step s^k. We also emphasize that α, β need to satisfy conditions of Theorems 4.1 and 4.2 in order to guarantee convergence of $\{\sigma_2^k\}$ because this can be guaranteed only at a point w^* satisfying the KKT conditions of the original NLP.

This result implies that there exists a bound $\mu^* < \mu_1$ such that $\|s^k\| \leq \mu^* \Delta^k$ for all large k. To establish superlinear convergence results, we also require that $\mathcal{A}_P(w^k) = \mathcal{A}_P(w^*)$ for k sufficiently large. We can now state the superlinear convergence result.

THEOREM 5.3. *Let $\{w^k\}$ be the sequence generated by the trust-region Newton method under the assumptions of Theorem* 5.1. *Assume that $\{w^k\}$ converges to a solution w^* that satisfies the SSOC of the penalty problem* (5.4). *If the step satisfies $\|s^k\| \leq \mu^* \Delta^k$, then the sequence $\{w^k\}$ converges Q-superlinearly to w^*.*

Proof. We follow the steps of Theorem 5.4 in [31]. The authors first showed that $\mathcal{A}_P(w^k) = \mathcal{A}_P(w^*)$ for k sufficiently large. We need the following estimate:

$$\begin{aligned}
&\|\mathbb{N}^k \nabla P(w^{k+1})\| \\
&\quad \le \left\|\mathbb{N}^k \left[\nabla P(w^{k+1}) - \nabla P(w^k) - \nabla^2 P(w^k)s^k\right]\right\| + \left\|\mathbb{N}^k \left[\nabla^2 P(w^k)s^k - Q(w^k)s^k\right]\right\| \\
&\qquad + \left\|\mathbb{N}^k \left[\nabla P(w^k) + Q(w^k)s^k\right]\right\| \\
&\quad \le (\epsilon_1^k + \epsilon_2^k)\|s^k\|.
\end{aligned}$$

Here, $\mathbb{N}^k = N^k {N^k}^T$. The first bound on the right-hand side follows from Taylor's theorem, the second bound follows from Property 2 and the Dennis–Moré condition (4.17), and the third term is bounded above by zero because of the convergence condition (5.18) with $\xi^k = 0$. We also have that $\{\epsilon_1^k\}$ converges to zero. From Theorems 4.1 and 4.2 we have that the KKT conditions of the original NLP hold at w^* such that $\{\epsilon_2^k\}$ converges to zero. Theorem 5.3 in [31] shows that there exists $\nu_0 > 0$ such that $\|s^k\| \le \nu_0 \|\mathbb{N}^* \nabla P(w^k)\|$ with $\mathbb{N}^* = N^* {N^*}^T$. With this, we obtain

$$\frac{\|\mathbb{N}^k \nabla P(w^{k+1})\|}{\|\mathbb{N}^* \nabla P(w^k)\|} \le (\epsilon_1^k + \epsilon_2^k)\nu_0 \tag{5.26}$$

and

$$\lim_{k\to\infty} \frac{\|\mathbb{N}^k \nabla P(w^{k+1})\|}{\|\mathbb{N}^* \nabla P(w^k)\|} \le 0. \tag{5.27}$$

Lin and Moré showed that there exists $\nu_1 > 0$ such that

$$\|\mathbb{N}^k P(w^{k+1})\| \ge (\nu_1 - \epsilon_1^k)\|w^{k+1} - w^*\| \tag{5.28}$$

and

$$\|\mathbb{N}^* P(w^k)\| \le \nu_2 \|w^k - w^*\| + \epsilon_1^k \|w^k - w^*\|, \quad \nu_2 = \|\mathbb{N}^* \nabla^2 P(w^*)\mathbb{N}^*\|. \tag{5.29}$$

These estimates use w^* as a fixed point and rely only on $\nabla^2 P(w^*)$, which by Property 2 is equal to $Q(w^*)$. Consequently,

$$(\nu_1 - \epsilon_1^k)\|w^{k+1} - w^*\| \le \nu_2 \|w^k - w^*\| + \epsilon_1^k \|w^k - w^*\|. \tag{5.30}$$

Since $\{\epsilon_1^k\}$ converges to zero, we have

$$\lim_{k\to\infty} \frac{\|w^{k+1} - w^*\|}{\|w^k - w^*\|} \le 0.$$

The proof is complete. ∎

If α, β are appropriately chosen, superlinear convergence can be achieved. This implies the surprising result that ignoring the third-order term (last term in (4.14)) does not destroy superlinear convergence. For the equality-constrained case the result is perhaps less surprising because the gradient of the Lagrangian converges to zero.

6. Summary of algorithm and scalability. In this section we provide the whole set of algorithmic components in order to highlight the main computational steps and discuss scalability issues.

6.1. EDPF-TRN algorithm. Assume given user-provided functions to compute $f(x)$, $c(x)$,$\nabla_x f(x)$,$\nabla_x h(x) \cdot u$, $\nabla_x h(x)^T \cdot u$, and $\nabla_{x,x}\mathcal{L}(x,\lambda) \cdot u$.

EDPF-TRN Algorithm

(A) Start at primal-dual pair $w = [x, \lambda]$. Set α, β and tolerance τ_P.

(B) Call TRN algorithm at $w^0 \leftarrow w$ and Δ^0.

1. **Initialization.** Start at w^0 and Δ^0.
2. **Major Step Test.**
   - 2.1. Compute EDPF $P^k := P_{\alpha,\beta}(w^k)$ from (3.12).
   - 2.2. Compute gradient of EDPF $g^k := \nabla_w P_{\alpha,\beta}(w^k)$ from (4.2) and projected gradient $g_{\mathrm{Proj}}(w^k)$ from (5.2).
   - 2.3. If w^k is a stationary point, set $w^*(\alpha,\beta) \leftarrow w^k$, and STOP.
3. **Cauchy Search.** Compute Cauchy step s_k^c from (5.9). At each line-search step compute approximate Hessian-vector product $Q(w^k)\cdot s_c^k$ using (4.16) and $\Psi^k(s_c^k)$ from (5.5), and STOP when (5.12) is satisfied. Update $w_c^k \leftarrow w^k + s_c^k$.
4. **Refinement Search.** Start at $w_0^k := w_c^k$. DO for $j = 0, \ldots, \ell + 1$:
   - 4.1. *PCG Step.* At $\mathcal{A}_P(w_j^k)$ construct $N \leftarrow N_j^k$, compute $g_d^k = N^T g^k$, and call Steihaug's PCG search for (5.16) with tolerance ϵ:
     - 4.1.1. Starting at $i = 0$ with $r_i = g_d^k$, $d_i = -r_i$, $z_i = 0$, apply preconditioner $y_i = D^{-1} \cdot r_i$. DO for $i > 0$:
     - 4.1.2. Form $\bar{d}_i = N^T d_i$ and $Q(w^k) \cdot \bar{d}_i$ using (4.16).
     - 4.1.3. If (C2) holds: $\bar{d}_i^T (Q(w^k) \cdot d_i) < 0$, find $\tau > 0$ such that $\|s_d^k\| = \Delta^k$ with $s_d^k = z_i + \tau d_i$, and STOP.
     - 4.1.4. Set $\gamma_i \leftarrow r_i^T y_i / \bar{d}_i^T (Q(w^k) \cdot \bar{d}_i)$.
     - 4.1.5. Set $z_{i+1} \leftarrow z_i + \gamma_i d_i$.
     - 4.1.6. If (C3) holds: $\|z_{i+1}\| \geq \Delta^k$, find $\tau > 0$ such that $\|s_d^k\| = \Delta^k$ with $s_d^k = z_i + \tau d_i$, and STOP.
     - 4.1.7. Set $r_{i+1} \leftarrow r_i + \gamma_i N^T (Q(w^k) \cdot \bar{d}_i)$, and apply preconditioner $y_{i+1} = D^{-1} r_{i+1}$.
     - 4.1.8. If (C1) holds: $\|r_{i+1}\| \leq \epsilon$, set $s_k^d = z_{i+1}$, and STOP.
     - 4.1.9. Set $\delta_{i+1} \leftarrow r_{i+1}^T y_{i+1} / r_i^T y_i$ and $d_{i+1} \leftarrow -y_{i+1} + \delta_{i+1} d_i$ and RETURN TO 4.1.2.
   - 4.2. *Minor Step Test.* Set $w_{j+1}^k \leftarrow w_j^k + N_j^k s_d^k$. If improvement over Cauchy test (5.10) and (5.12) are satisfied, STOP refinement search, set $s^k = w_{j+1}^k - w^k$, and GO TO 5.
   - 4.3. *Update Minor Step.* Cut step to satisfy (5.13),(5.14), and (5.19), update w_{j+1}^k, set $j \leftarrow j + 1$, and RETURN TO 4.1.
5. **Update Step.** Compute reduction ratio (5.6), update step w^{k+1} according to (5.7), and update trust-region radius Δ^{k+1} according to (5.8). Set $k \leftarrow k+1$, and RETURN to 2.

(C) Given $w^*(\alpha,\beta)$, check infeasibility $\|w^*(\alpha,\beta) - \mathrm{Proj}(w^*(\alpha,\beta) - \nabla_w \mathcal{L}(w^*(\alpha,\beta)))\| \leq \tau_P$. If satisfactory, STOP. Otherwise, update α, β and RETURN TO B.

Relationship of the EPDF-TRN algorithm to objectives (i)–(v) in section 1. For fixed and appropriate α, β, the EPDF-TRN algorithm does not use third-order derivative information and exhibits global convergence, Theorem 5.1. In addition, it satisfies (i) from Theorem 5.3. By design, since it uses only PCG and projected gradient, it is matrix-free and uses iterative linear algebra, whereas the truncated trust-region algorithm allows it to accommodate negative curvature, so (ii) and (iii) are achieved. Moreover, inertia detection through linear algebra is not necessary, nonconvexity being handled by the truncated trust-region approach. In the limit of the active set being correctly detected and being close to the optimal manifold and

thus the trust-region being inactive, as one does a parameter perturbation for a parametric problem, one PCG iteration will reduce the EPDF for the equality-constrained problem, so monotonic progress will be achieved, and thus (iv) holds. Moreover, inheriting from its bound-constrained active set philosophy, warm start is intrinsically achieved, whereas active set detection can be efficiently done by gradient projection so (v) is achieved.

6.2. Choosing α and β. While we have proved several attractive theoretical features of the EDPF-TRN algorithm, its practical performance strongly relies on the choice of α, β. Compared with typical penalty methods, which need to set only one parameter, we here have to choose two; and even in the equality constrained case the considerations for choice are not trivial [6]. While a complete answer to this question requires signficant additional investigation, we provide here some guidelines to update α, β *outside* the trust-region algorithm by monitoring negative curvature and negative components of the gradient of the Lagrangian and we provide justifications based on the results of Theorems 4.1 and 4.2. Such an outside-the-solver update approach is used in existing AL implementations where the penalty parameter is updated based on the residual of the equality constraints [1, 12]. We emphasize that the convergence analysis required for the case in which α and β are altered dynamically *inside* the trust-region algorithm is challenging, because it changes the merit function itself. One of the consequences is that the active set detection properties of the Cauchy step cannot be guaranteed because the exposed face by the gradient changes with α and β. This can potentially result in cycling. Dynamic parameter updates in an AL setting have been considered only recently for the case of NLPs with equality constraints [14].

The parameters α, β need to be such that the results of Theorems 4.1 and 4.2 hold. The following conditions need to be satisfied:

(D1) $G_1 \succ 0$ (4.5),
(D2) $Q_2 \succ 0$ (4.7),
(D3) $1 + 2\beta\nabla_x \mathcal{L}(w)_{(i)} > 0$, $i = 1, \ldots, n$ (4.8),
(D4) $H_P \succ 0$, (4.19) at a point w^* satisfying the KKT conditions for (1.1) if and only if that point satisfies SSOC (A3).

We sketch an outer loop to update α and β.

Update Algorithm for α, β
Assume given parameters $\mu_\alpha > \mu_\beta > 1$.

(I) Start with $k = 0$, and choose w^0 and $\alpha^0 > 0, \beta^0 > 0$.
(II) Call EDPF-TRN with w^0 (initial guess), α^k, β^k.
(III) First-order test: If w^k is not a stationary point of (1.1), set $\alpha^k \leftarrow \mu_\alpha \alpha^k$, $\beta^k \leftarrow \frac{\beta^k}{\mu_\beta}$, $k = k + 1$, and REPEAT (II)
(IV) Second-order test: If $H_P \succ 0$ but w^k does not satisfy SSOC for (1.1), set $\alpha^k \leftarrow \mu_\alpha \alpha^k$, $\beta^k \leftarrow \frac{\beta^k}{\mu_\beta}$, $k = k + 1$, and REPEAT (II)

We note that with this choice of μ_α and μ_β, $k \to \infty$ implies that $\alpha^k \to \infty$, $\beta^k \to 0$, $\alpha^k \beta^k \to \infty$. If the iterates of w^k remain in a compact set $\mathcal{K}$ as in Theorem 4.1, then for a finite k both optimality tests should be passed. Indeed, for β^k sufficiently small, from (4.5) and, respectively, (4.8), conditions (D1) and, respectively, (D3) must hold for all w^* in $\mathcal{K}$ and thus for all stationary points of (3.14). For β^k sufficiently small, α^k sufficiently large, and $\alpha^k \beta^k$ sufficiently large, (D4) must hold from the discussion around (4.21), and (D2) must hold from (4.7).

The first-order test for w^* on (1.1) is low cost because it verifies only that $h(x^*) = 0$, $X^* \nabla_x \mathcal{L}(w^*) = 0$, and $\nabla_x \mathcal{L}(w^*) \geq 0$. The second-order test requires

the factorization of H_P (which for direct methods can be obtained as part of the minimization of (3.14)) but also determining whether SSOC holds, which requires identifying the signature of the reduced saddle-point matrix by, for example, symmetric indefinite factorization of [35]:

$$\begin{bmatrix} N_x^T H N_x & N_x^T A^T \\ A N_x & \end{bmatrix}. \tag{6.1}$$

If the matrix has exactly m negative eigenvalues and is not singular, SSOC is satisfied. If factorization is not possible and the method is implemented matrix-free, an alternative is to peform a few conjugate gradient (CG) iterations on the AL matrix $N_x^T H N_x + \alpha^k N_x^T A^T A N_x$. If α^k is large enough, then SSOC holds if and only if this matrix is positive definite. Certainly incomplete CG-type tests are inexact (and if (3.14) is solved iteratively, then the inertia of H_P is also likely to be only estimated but not known) and α^k very large may make them ineffective because of ill-conditioning; but there do not seem to be computationally efficient matrix-free ways to assess second-order properties. A projected CG approach is also an alternative [21].

To be practical, such an algorithm for choosing α and β needs to be enhanced in several ways. First, it needs to allow for solving EDPF-TRN inexactly before testing for optimality of (1.1). Second, it needs to find efficient ways to test for second-order optimality conditions, ideally as part as solving the subproblems. Third, adjustments to the parameters should ideally be also done within an iteration, if they can be done in ways to not destroy global convergence, such as in [14]. At this stage, and given our focus on scalability and warm-starting issues, the parameters were chosen empirically for the numerical results presented in section 7.

6.3. Derivatives. We can assemble the gradient and approximate Hessian of the EDPF (4.15a) times a vector using only vector products with the Hessian of the Lagrangian $H = \nabla_{x,x}\mathcal{L}(x,\lambda)$ and the Jacobian of the constraints $A = \nabla_x h(x)^T$. Evaluating the gradient of the EDPF (2.3) requires one Hessian vector product, one Jacobian vector product, and two Jacobian transpose vector products. To see this, we define an arbitrary vector $\nu^T = [\nu_x^T \ \nu_\lambda^T]$, and we note that

$$\nabla^2\mathcal{L}\cdot\nu = \begin{bmatrix} H & A^T \\ A & \end{bmatrix}\begin{bmatrix} \nu_x \\ \nu_\lambda \end{bmatrix} = \begin{bmatrix} H\cdot\nu_x + A^T\cdot\nu_\lambda \\ A\cdot\nu_x \end{bmatrix}. \tag{6.2}$$

Using this relation, we can see from (4.15a) that three *unique* products are needed to form the Hessian approximation of the EDPF. This can be computed with automatic differentiation (AD) packages at a cost that is proportional to the evaluation of the objective function and constraints (i.e., $O(n)$ and $O(m)$, respectively) [35]. We also highlight that coloring needs to be performed only once because the structure of the Hessian of the EDPF does not change along the search. This feature is particularly beneficial when the NLP is solved repetitively, as in parametric problems. The computation of the reduced Hessian can proceed without altering the structure of the Hessian by defining $\nu_x = N\cdot\nu_d$ because the null-space matrix is trivial (i.e., the matrix is formed of single one entries for the inactive elements of x). These observations are important because evaluating the exact Hessian of the EDPF (i.e., by specifying the EDPF directly) requires a recursive application of AD, a requirement resulting from the appearance gradient of the Lagrange function in the definition of the EDPF.

6.4. Step computation. The Cauchy point makes progress in the EDPF (e.g., solution of the NLP) in $O(n_w)$ operations. More important, each PCG iteration

makes progress in $O(n_w)$ operations with the same order for storage requirements. The dominant complexity, as expected, tends to be associated with the application of the preconditioner $D^{-1} \cdot r$ (unless this can be easily factorized).

As revealed by Theorems 4.1 and 4.2, an existing caveat of the EDPF formulation is that a large value of α and a small value of β are typically required to enable identification of stationary points of the original NLP. This requirement increases the spectrum of the approximate Hessian Q and of the reduced Hessian $N^T Q N$. Interestingly, however, the spectrum does not grow as w^* is approached as in IP methods. General preconditioning techniques that can be used in the proposed approach include constraint preconditioning, incomplete Cholesky, and Bunch–Parlett factorization [5, 30, 41] for which parallel implementations exist. The fact that the PCG matrix is positive definite close to solution also opens the possibility of applying general algebraic multigrid preconditioners, which scale as $O(n_w)$ [44]. We also highlight that inertia is detected externally through PCG so that linear algebra solvers do not need to provide inertia.

6.5. Warm start and early termination. The ability to warm start is a key property of the proposed approach. In particular, because it is based on gradient projection, no bound multiplier information and no centrality recovery are required, as in IP methods. In addition, multiple active set changes can be computed at each iteration through the Cauchy search. The ability to determine activity quickly is particularly relevant in applications that require early termination, such as model predictive control, state estimation, and rigid body simulation [20, 49, 2, 32].

7. Numerical studies. In this section we illustrate the behavior of the proposed framework, and we demonstrate its scalability properties.

7.1. Algorithmic behavior. We explain the behavior of the algorithmic framework using the following example:

$$\min \ (x_1 - 1)^2 + (x_2 - 2)^2 + (x_3 - 3)^2 + x_1 x_4 \tag{7.1a}$$

$$\text{s.t.} \ x_1 x_4 + x_1 x_2 + x_3 = 4, \quad (\lambda) \tag{7.1b}$$

$$x_1, x_2, x_3, x_4 \geq 0. \tag{7.1c}$$

The solution of this problem is $x^* = [1.62\ 1.62\ 1.38\ 0]$, $\lambda^* = -7.64$, and $f(x^*) = 0.91$. The gradient of the Lagrangian is given by

$$\nabla_x \mathcal{L}(x, \lambda) = \begin{bmatrix} 2(x_1 - 1) + x_4 + \lambda x_2 + \lambda x_4 \\ 2(x_2 - 2) + \lambda x_1 \\ 2(x_3 - 3) + \lambda \\ x_1 + \lambda x_1 \end{bmatrix}. \tag{7.2}$$

We can see that nonlinear terms exist in λ and x_1, x_2, and x_4. These induce third-derivative terms in the Hessian of the EDPF. To solve this problem, we set $\alpha = 1\text{e}+2$ and $\beta = 1\text{e}-3$ and use a tolerance for the projected gradient of 1e−5. We initialize the problem at $x^0 = [1\ 1\ 1\ 1]$ and $\lambda^0 = 1$. We summarize the convergence history of the trust-region Newton algorithm in Table 1. Here, we define $H^k := \nabla^2 P(w^k)$, $H_d^k := N^T H^k N$, $Q_d^k := N^T Q^k N$, and $\underline{\lambda}(W)$ is the minimum eigenvalue of a given matrix W. We make the following observations.

- The step in $k = 1$ is obtained from a direction of negative curvature and is accepted, because it leads to $\rho^k > 1$. The trust-region radius Δ^k is increased. We can also observe a large error in the Hessian approximation.

TABLE 1
Convergence history for example problem.

k	P^k	g^k_{Proj}	ρ^k	$\|s^k\|$	$\|\Delta^k\|$	$\|Q^k - H^k\|$	$\underline{\lambda}(Q^k_d)$	$\underline{\lambda}(H^k_d)$	$\text{card}(\mathcal{A}^k_P)$
0	25.150	2.0e+2							0
1	3.449	5.9e+1	+3.26	2.5e−1	261.9	2.0e+2	−2.48	−22.67	0
2	3.449	5.9e+1	−0.70	0.0e+0	523.9	5.8e+1	−2.48	−22.67	0
3	3.449	5.9e+1	−0.62	0.0e+0	131.0	5.8e+1	−2.48	−22.67	0
4	3.449	5.9e+1	−0.33	0.0e+0	32.0	5.8e+1	−2.48	−22.67	0
5	3.449	5.9e+1	−0.28	0.0e+0	8.0	5.8e+1	−2.48	−22.67	0
6	1.533	2.5e+1	+0.37	2.0e+0	2.0	5.8e+1	−2.48	−22.67	0
7	0.945	1.6e+0	+0.52	1.9e−1	2.0	2.9e+1	+0.15	−0.39	0
8	0.944	4.9e−1	+0.48	2.6e−3	4.0	1.9e+0	+0.19	+0.37	0
9	0.943	4.5e−1	+0.93	1.4e−3	4.0	4.0e−1	+0.19	+0.25	0
10	0.909	2.3e−1	+0.94	1.8e−1	8.0	3.4e−1	+0.40	+0.40	1
11	0.908	1.7e−6	+0.99	8.7e−3	16.0	3.1e−6	+0.38	+0.38	1

- The step in $k = 2$ is rejected because $\rho^k < 0$. The trust-region radius is decreased. The same behavior is observed until $k = 6$, when Δ^k is sufficiently small to make $\rho^k > 0$. The step at this iteration results from a direction of negative curvature and is accepted but Δ^k is kept constant because ρ^k is not large enough.
- At iterations $k = 7, 8, 9$ the algorithm makes progress, and Δ^k keeps increasing. At $k = 7$ we note that the minimum eigenvalue of the approximate reduced Hessian is positive while that of the exact is negative. This illustrates how the third-derivative term can introduce negative curvature.
- At iterations $k = 10, 11$ the Cauchy step detects the active variable and convergence is attained. The error of the approximate Hessian converges to zero, and ρ^k converges to one. The reduced Hessian is positive definite.

7.2. Scalability. To demonstrate scalability, we consider the following continuous-time optimal control problem (OCP):

$$\text{(7.3a)} \qquad \min \int_0^T \left(\alpha_c \cdot (c(\tau) - \bar{c})^2 + \alpha_t \cdot (t(\tau) - \bar{t})^2 + \alpha_u \cdot (u(t) - \bar{u})^2\right) d\tau$$

$$\text{(7.3b)} \qquad \text{s.t. } \dot{c}(\tau) = \frac{1 - c(\tau)}{\theta} - p_k \cdot \exp\left(-\frac{p_E}{t(\tau)}\right) \cdot c(t),$$

$$\text{(7.3c)} \qquad \dot{t}(\tau) = \frac{t_f - t(\tau)}{\theta} + p_k \cdot \exp\left(-\frac{p_E}{t(\tau)}\right) \cdot c(\tau) - p_\alpha \cdot u(\tau) \cdot (t(\tau) - t_c),$$

$$\text{(7.3d)} \qquad c(\tau), t(\tau), u(\tau) \geq 0, \quad \tau \in [0, T],$$

$$\text{(7.3e)} \qquad c(0) = c(\tau_{sys}), \quad t(0) = t(\tau_{sys}).$$

The system is an unstable chemical reactor. The internal time is given by τ covering the prediction horizon $[0, T]$. The system states are concentration of reactant $c(\cdot)$ and temperature of reacting mixture $t(\cdot)$. The control is the cooling water flow $u(\cdot)$. The real time of the system is τ_{sys} and the system states at this time are $c(\tau_{sys}), t(\tau_{sys})$. Symbols $\alpha_c, \alpha_t, \alpha_u, p_\alpha, p_E, t_f, t_c, p_k$ are model parameters and can be found in [49]. The objective function is of Bolza type with the desired end points $\bar{c}, \bar{t}, \bar{u}$. We transform this problem into an NLP by direct transcription with implicit Euler discretization. We use a mesh with N points each of length $\Delta\tau$. To scale the problem, we increase the number of time steps in the horizon N. The resulting dimensions of the

TABLE 2
Dimensions of discretized OCP.

N	n	m	n_w	nnz($\nabla^2\mathcal{L}$)	nnz(Q)	%dens($\nabla^2\mathcal{L}$)	%dens(Q)
500	1,500	1,000	2,500	10,486	26,492	2.0e−1	4.0e−1
1,000	3,000	2,000	5,000	20,996	52,972	8.4e−2	2.0e−1
5,000	15,000	10,000	25,000	104,996	264,972	1.6e−2	4.0e−2
10,000	30,000	20,000	50,000	209,996	529,972	8.3e−3	2.1e−2

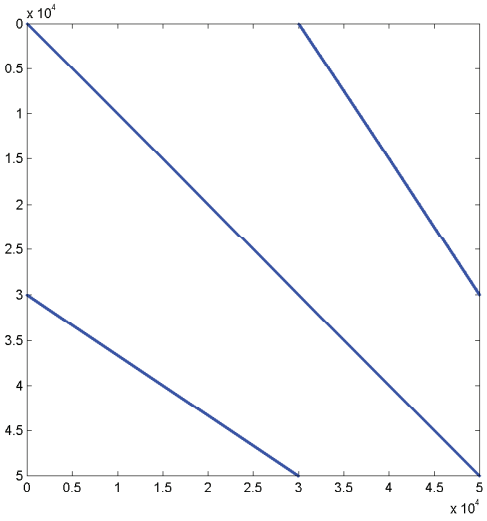

FIG. 1. *Sparsity structure of approximate Hessian Q for NLP with $n_w = 5{,}000$.*

NLPs are presented in Table 2. Here, we present the density in percentage for the Hessian of the Lagrange function %dens($\nabla^2\mathcal{L}$) and for the approximate Hessian of the EDPF %dens(Q). From these numbers we observe that the Hessian is very sparse, which is typical in direct transcription approaches for optimal control [50]. We also observe that even if the number of nonzeros in the EDPF Hessian is larger by a factor of 2.5 compared with the Hessian of the Lagrangian (augmented system), sparsity is preserved. The sparsity structure for a problem with effective dimension $n_w = 5{,}000$ is presented in Figure 1, from where we can also appreciate a banded structure also typical in OCPs [39].

In this study, we test for scalability of the two dominant computational steps. The first step is the incomplete sparse Cholesky factorization of the approximate Hessian matrix to generate the preconditioner D for PCG. In our experiments we set the drop tolerance to 1e−4. We compare this with the case in which a full sparse Cholesky factorization of the Hessian is performed. The second dominant step is the PCG search, which involves recursive backsolves with the factors of the preconditioner. In the case of full Cholesky factorization a single PCG iteration (backsolve) is needed. All computations are performed in MATLAB.

To solve these problems, we used $\alpha = 1\text{e}+6$ and $\beta = 1\text{e}{-1}$, and we initialize the problems at a perturbed point from the optimal solution generated. This perturbation gives an initial error of the projected gradient of $O(10^3)$. We use a tolerance for the projected gradient of 1e−5. All problems are solved in four iterations.

Scalability results as a function of the effective dimension n_w are reported in Table 3. Here, we compare the average number of PCG iterations (it_{cg}) per Newton

TABLE 3
Average computational times for OCP per Newton iteration.

n_w	it_{pcg}	$\theta_{ichol,pcg}$	$\theta_{ichol,fact}$	$\theta_{ichol,tot}$	$\theta_{chol,pcg}$	$\theta_{chol,fact}$	$\theta_{chol,tot}$
1,250	17	8.5e−2	3.1e−2	1.1e−1	2.7e−2	3.3e−2	6.0e−2
2,500	24	4.9e−1	1.3e−1	6.2e−1	1.1e−1	1.5e−1	2.6e−1
5,000	29	1.7e+0	4.4e−1	2.2e+0	5.7e−1	8.5e−1	1.4e+0
12,500	31	9.0e+0	1.8e+0	1.1e+1	3.8e+0	8.4e+0	1.2e+1
25,000	31	1.8e+1	5.5e+0	2.4e+1	2.5e+1	5.4e+1	7.8e+1
50,000	31	3.7e+1	1.8e+1	5.5e+1	-	-	-
125,000	31	9.4e+1	1.1e+2	2.0e+2	-	-	-
250,000	31	1.9e+2	4.9e+2	6.8e+2	-	-	-

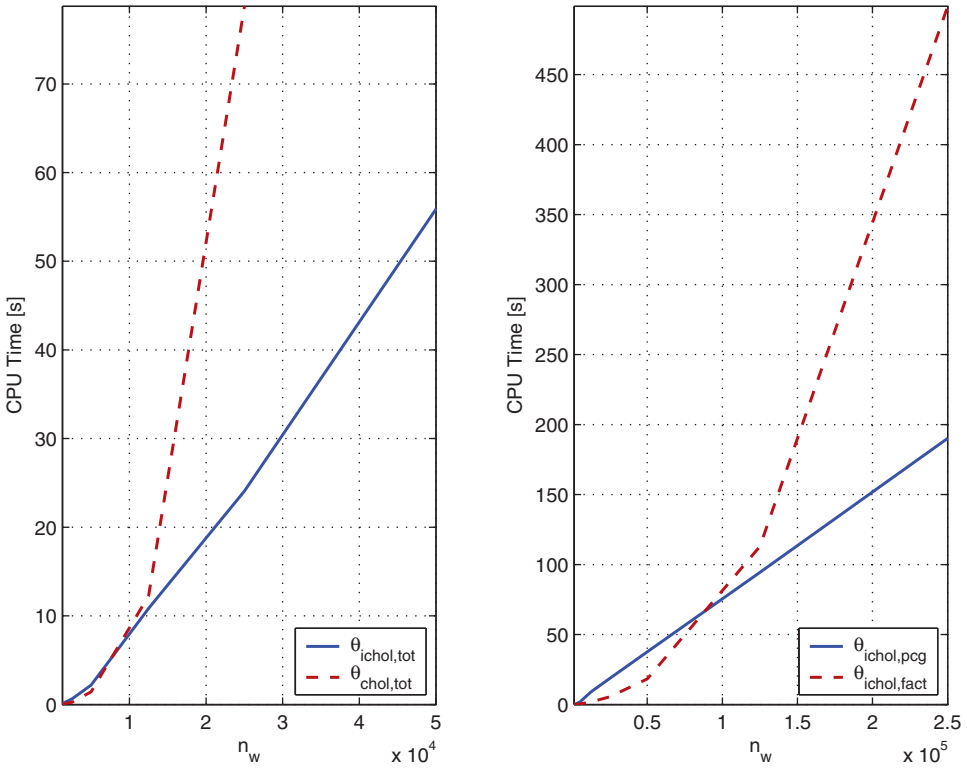

FIG. 2. *Average total time per iteration with Cholesky and incomplete Cholesky (left). Average PCG and incomplete Cholesky times per iteration (right).*

iteration, average time of PCG procedure ($\theta_{ichol,pcg}$) per Newton iteration, average time of incomplete factorization ($\theta_{ichol,fact}$), and average total time per iteration ($\theta_{ichol,tot} = \theta_{ichol,pcg} + \theta_{ichol,fact}$). The same quantities are presented for the full Cholesky case and are denoted as $\theta_{chol,pcg}$, $\theta_{chol,fact}$, and $\theta_{chol,tot}$, respectively. The times are illustrated graphically in Figure 2. We make the following observations.

- The number of PCG iterations increases linearly with n_w and settles. This "settling" is a particular property of OCPs with Bolza objective because the end point $\bar{c}, \bar{t}, \bar{u}$ is reached after a given number of time steps N and the Krylov subspace remains constant.
- The total times for full Cholesky are not competitive. This result is clearly seen in the left panel of Figure 2. This illustrates the additional flexibil-

TABLE 4
Active set identification histories for NLP with $n_w = 2{,}500$.

	Case 1				Case 2			
k	P^k	g^k_{Proj}	$\mathcal{A}_P(w^k)$	n^k_{PCG}	P^k	g^k_{Proj}	$\mathcal{A}_P(w^k)$	n_{PCG}
0	4.05e+3	4.52e+3	44	-	1.21e+4	2.43e+5	173	-
1	1.14e+2	4.70e+3	44	41	4.96e+2	5.76e+4	0	132
2	1.83e+1	3.72e+3	119	32	9.48e+1	1.86e+3	0	45
3	1.83e+1	1.55e+2	170	27	5.57e+0	3.27e+4	26	37
4	1.83e+1	5.59e−6	173	17	3.98e+0	1.11e+3	43	26
5	-	-	-		3.98e+0	8.50e−6	44	13

ity gained by enabling relaxation in the incomplete Cholesky factorization procedure.

- From the full Cholesky times we can see that, as expected, the complexity of performing a single backsolve $\theta_{chol,pcg}$ is similar to that performing the factorization $\theta_{chol,fact}$ particularly because of fill-in effects.
- From the incomplete Cholesky times and the number of PCG iterations we can see that the time per backsolve can be dramatically reduced. This is particularly evident in the very large problems where the number of PCG iterations is constant.
- The number of backsolves needed makes PCG the dominant expense below the $n_w = 100{,}000$ threshold. Factorization becomes dominant above the threshold. This is illustrated in the right panel of Figure 2.

To demonstrate the activity detection properties of the framework, we generate an NLP with $n_w = 2{,}500$. We consider two cases. In the first case we start the search at a point with $\mathcal{A}_P(w^0) = 44$ and with optimal solution $\mathcal{A}_P(w^*) = 173$. In the second case we reverse the process to generate an initial point with $\mathcal{A}_P(w^0) = 173$ and $\mathcal{A}_P(w^*) = 44$. The convergence history is summarized in Table 4. As can be seen, the Cauchy step can make large adjustments in the active set at each iteration. Moreover, the number of PCG iterations remains fairly stable and in fact reduces as the solution is approached (as the third-order term vanishes and the active set settles). This is a key advantage over IP methods where preconditioner performance degrades as the solution is approached [37].

7.3. Warm starting and early termination. One of the crucial properties of the framework is that it enables warm starting and early termination. To demonstrate these capabilities, we consider a receding-horizon OCP framework. This approach is also called MPC and is typically used to avoid solving problems with infinite horizons. The OCP (7.3) is solved recursively as a parametric problem, and the time horizon is shifted by a factor $\Delta\tau$ as $\tau_{sys} \leftarrow \tau_{sys} + \Delta\tau$. The initial states are updated by using the time-evolving $c(\tau_{sys}), t(\tau_{sys})$ system states as $c(0) \leftarrow c(\tau_{sys})$, $t(0) \leftarrow t(\tau_{sys})$. This update is done until the system converges $c(\tau_{sys}) \to \bar{c}$ and $t(\tau_{sys}) \to \bar{t}$. The procedure generates a continuous-time manifold $c^*(\tau), t^*(\tau), u^*(\tau),\ \tau \in [0, \tau_{sys}]$, where τ_{sys}.

The main obstacle preventing the use of MPC is the computational latency of the OCP solution. We have recently shown that one inexact QP solution (inexact Newton iteration in case of no inequalities) for the NLP can be sufficient to track the optimal manifold stably and to ultimately steer the system to the end point [49]. In the case described here, the use of iterative linear algebra provides significantly more flexibility than does direct linear algebra for early termination. This flexibility, combined with fast active set identification, enables the use of MPC in a much wider range of applications. Recently, we proposed an AL NLP reformulation and a pro-

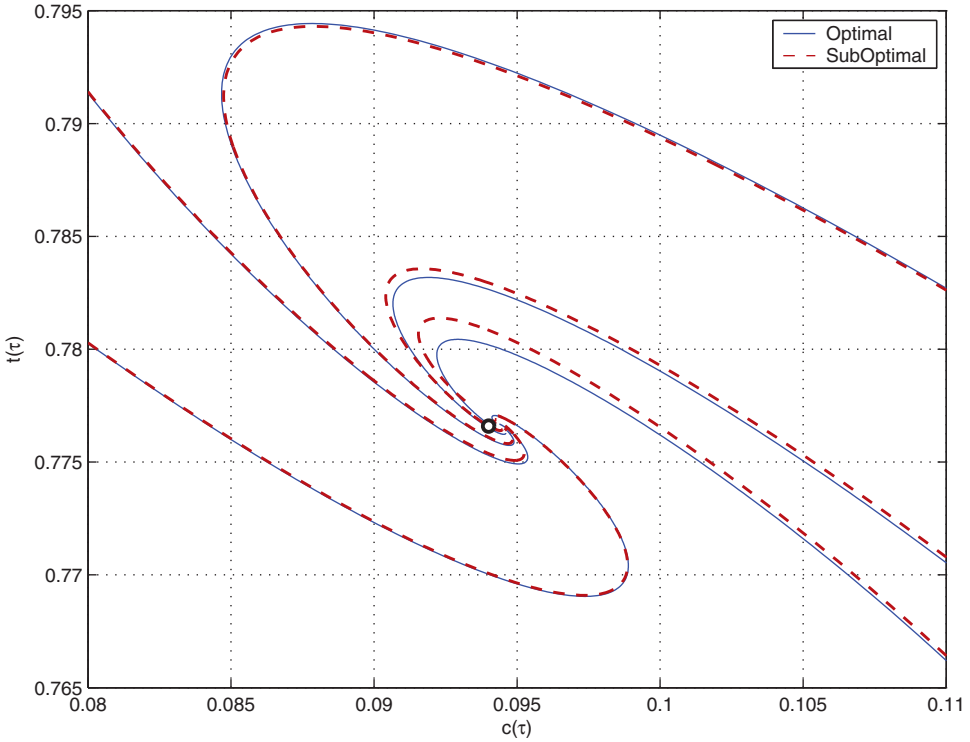

FIG. 3. *Phase plane manifolds with tight tolerance and with early termination.*

jected Gauss–Seidel scheme to solve the QP inexactly by terminating after a finite number of iterations [49]. The approach, however, is limited in scalability because of the difficulty in preconditioning projected Gauss–Seidel, and it requires estimates of the multipliers. The EDPF approach proposed here overcomes these limitations.

To illustrate the flexibility gained with the ability to warm start and terminate early, we performed MPC computations for a problem with $N = 100$. We computed optimal manifolds from different starting points by solving the NLPs with a tolerance of 1e−5. This process required, on average, 4 Newton iterations and 40 PCG iterations per Newton iteration. For comparison, we consider a suboptimal strategy in which we terminate after 2 Newton iterations and 20 PCG iterations per Newton iteration. This *reduced the latency by a factor of* 4. The optimal and suboptimal manifolds are presented in Figure 3. As can be seen, errors are present but are stable and the trajectories converge in all cases to the desired end point.

8. Conclusions and future work. We have presented a general approach for nonlinear programming based on the direct minimization of an exact differentiable penalty function using a trust-region Newton approach. We demonstrate that the framework is scalable in the sense that (i) it is superlinearly convergent, (ii) it is matrix-free, (iii) it exploits directions of negative curvature, (iv) it makes direct progress on the merit function in its minor iterations, and (v) it enables efficient detection of activity through gradient projection and enables the use of activity information to warm start.

As part of future work, we will develop a more general implementation of the framework allowing for automatic adjustment of the penalty function parameters.

In addition, we will consider more general penalty functions requiring only the α parameter. This function has the form [6]

$$P_\alpha(w) = \mathcal{L}(w) + \frac{1}{2}\alpha h(x)^T h(x) + 2\,\nabla_x \mathcal{L}(w)^T \sqrt{X} M(x) M(x)^T \sqrt{X} \nabla_x \mathcal{L}(w), \tag{8.1}$$

where $M(x) \in \mathcal{R}^{n\times p}$, $m \leq p \leq n$ is a matrix that makes $M(x)\nabla_x h(x)^T$ nonsingular. A typical choice is $M(x) = \nabla_x h(x)$ with $p = m$ for which $\nabla_x h(x)^T$ is required to be full rank. The use of this function will enable more algorithmic flexibility and will limit the spectrum of the Hessian matrix. However, derivative computations are more complicated, and efficient ways for computing them using Hessian-vector and Jacobian-vector products need to be investigated. These extensions will require major modifications to existing convergence results because a more intrusive implementation of the trust-region Newton method and different step acceptance criteria are needed. We will also investigate different approaches to efficiently solve the trust-region QP [40]. Moreover, we will investigate efficient preconditioning approaches including the multilevel type for the resulting linear systems, which is somewhat facilitated by the fact that we do not need to detect inertia as is the case for other NLP frameworks. All these additions will enable us to benchmark against existing NLP solvers.

Acknowledgment. Mihai Anitescu thanks Stephen Wright for pointing him to some key exact penalty function issues during the former's Wilkinson fellowship.

REFERENCES

[1] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt, *On augmented Lagrangian methods with general lower-level constraints*, SIAM J. Optim., 18 (2007), pp. 1286–1309.

[2] M. Anitescu and F. A. Potra, *Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems*, Nonlinear Dynam., 14 (1997), pp. 231–247.

[3] P. Armand and D. Orban, *The squared slacks transformation in nonlinear programming*, SQU J. Sci., 17 (2012), pp. 22–29.

[4] R. A. Bartlett, A. Wächter, and L. T. Biegler, *Active set vs. interior point strategies for model predictive control*, in Proceedings of the 2000 American Control Conference, IEEE, Piscataway, NJ, 2000, pp. 4229–4233.

[5] M. Benzi, *Preconditioning techniques for large linear systems: A survey*, J. Comput. Phys., 182 (2002), pp. 418–477.

[6] D. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 1982.

[7] E. G. Birgin, R. A. Castillo, and J. M. Martinez, *Numerical comparison of augmented Lagrangian algorithms for nonconvex problems*, Comput. Optim. Appl., 31 (2005), pp. 31–55.

[8] J. Burke, J. Moré, and G. Toraldo, *Convergence properties of trust region methods for linear and convex constraints*, Math. Program., 47 (1990), pp. 305–336.

[9] J. V. Burke and J. J. Moré, *On the identification of active constraints*, SIAM J. Numer. Anal., 25 (1988), pp. 1197–1211.

[10] R. H. Byrd, J. Ch. Gilbert, and J. Nocedal, *A trust-region method based on interior-point techniques for nonlinear programming*, Math. Program., 89 (2000), pp. 149–185.

[11] A. R. Conn, N. I. M. Gould, and P. Toint, *A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds*, SIAM J. Numer. Anal., 28 (1991), pp. 545–572.

[12] A. Conn, N. Gould, and P. Toint, *Numerical experiments with the LANCELOT package (release a) for large-scale nonlinear optimization*, Math. Program., 73 (1996), pp. 73–110.

[13] F. E. Curtis, O. Schenk, and A. Wächter, *An interior-point algorithm for large-scale nonlinear optimization with inexact step computations*, SIAM J. Sci. Comput., 32 (2010), pp. 3447–3475.

[14] F. E Curtis, H. Jiang, and D. P. Robinson, *Adaptive augmented Lagrangian methods for large-scale equality constrained optimization*, submitted.
[15] J. E. Dennis and J. J. Moré, *A characterization of superlinear convergence and its application to quasi-Newton methods*, Math. Comp, 28 (1974), pp. 549–560.
[16] G. Di Pillo and L. Grippo, *A new class of augmented Lagrangians in nonlinear programming*, SIAM J. Control Optim., 17 (1979), pp. 618–628.
[17] G. Di Pillo and L. Grippo, *Exact penalty functions in constrained optimization*, SIAM J. Control Optim., 27 (1989), pp. 1333–1360.
[18] M. Diehl, *Real-Time Optimization of Large-Scale Nonlinear Processes*, Ph.D. thesis, University of Heidelberg, Heidelberg, Germany, 2001.
[19] M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, *Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations*, J. Process Control, 12 (2002), pp. 577–585.
[20] M. Diehl, H. J. Ferreau, and N. Haverbeke, *Efficient numerical methods for nonlinear MPC and moving horizon estimation*, in Nonlinear Model Predictive Control, Springer, Berlin, 2009, pp. 391–417.
[21] H. S. Dollar, N. I. M. Gould, W. H. A. Schilders, and A. J. Wathen, *Implicit-factorization preconditioning and iterative solvers for regularized saddle-point systems*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 170–189.
[22] R. Fletcher, *Practical Methods of Optimization*, Wiley, Chichester, England, 1987.
[23] A. Forsgren, P. E. Gill, and M. H. Wright, *Interior methods for nonlinear optimization*, SIAM Rev., 44 (2002), pp. 525–597.
[24] A. Forsgren, P.E. Gill, and J.D. Griffin, *Iterative solution of augmented systems arising in interior methods*, SIAM J. Optim., 18 (2008), pp. 666–690.
[25] P. Gill, W. Murray, and M. Wright, *Practical Optimization*, Academic Press, London, 1981.
[26] P. E. Gill and D. Robinson, *A primal-dual augmented Lagrangian*, Comput. Optim. Appl., 51 (2012), pp. 1–25.
[27] J. Gondzio and A. Grothey, *Reoptimization with the primal-dual interior point method*, SIAM J. Optim., 13 (2002), pp. 842–864.
[28] S. P. Han and O. L. Mangasarian, *Exact penalty functions in nonlinear programming*, Math. Program., 17 (1979), pp. 251–269.
[29] A. Izmailov and M. Solodov, *Inexact Josephy-Newton framework for generalized equations and its applications to local analysis of Newtonian methods for constrained optimization*, Comput. Optim. Appl., 46 (2010), pp. 347–368.
[30] C.-J. Lin and J. J. Moré, *Incomplete Cholesky factorizations with limited memory*, SIAM J. Sci. Comput., 21 (1999), pp. 24–45.
[31] C.-J. Lin and J. J. Moré, *Newton's method for large bound-constrained optimization problems*, SIAM J. Optim., 9 (1999), pp. 1100–1127.
[32] J. L. Morales, J. Nocedal, and M. Smelyanskiy, *An algorithm for the fast solution of symmetric linear complementarity problems*, Numer. Math., 11 (2008), pp. 251–266.
[33] J. J. Moré, *Trust regions and projected gradients*, in System Modelling and Optimization, Springer, Berlin, 1988, pp. 1–13.
[34] S. G. Nash, *Sumt (Revisited)*, Oper. Res., 46 (1998), pp. 763–775.
[35] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, New York, 2006.
[36] T. Ohtsuka, *A continuation/GMRES method for fast computation of non-linear receding horizon control*, Automatica J. IFAC, 40 (2004), pp. 563–574.
[37] C. Petra and M. Anitescu, *A preconditioning technique for Schur complement systems arising in stochastic optimization*, Comput. Optim. Appl., 52 (2012), pp. 315–344.
[38] G. Di Pillo and L. Grippo, *A continuously differentiable exact penalty function for nonlinear programming problems with inequality constraints*, SIAM J. Control Optim., 23 (1985), pp. 72–84.
[39] C. V. Rao, S. J. Wright, and J. B. Rawlings, *Application of interior-point methods to model predictive control*, J. Optim. Theory Appl, 99 (1998), pp. 723–757.
[40] M. Rojas, S. A. Santos, and D. C. Sorensen, *A new matrix-free algorithm for the large-scale trust-region subproblem*, SIAM J. Optim., 11 (2001), pp. 611–646.
[41] O. Schenk, A. Wächter, and M. Hagemann, *Matching-based preprocessing algorithms to the solution of saddle-point problems in large-scale nonconvex interior-point optimization*, J. Comput. Opt. Appl., 36 (2007), pp. 321–341.
[42] O. Schenk, A. Wächter, and M. Weiser, *Inertia-revealing preconditioning for large-scale nonconvex constrained optimization*, SIAM J. Sci. Comput., 31 (2009), pp. 939–960.
[43] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.

[44] K. Stueben, *A review of algebraic multigrid*, J. Comput. Appl. Math., 128 (2001), pp. 281–309.
[45] A. Wächter and L. T. Biegler, *Line search filter methods for nonlinear programming: Motivation and global convergence*, SIAM J. Optim., 16 (2005), pp. 1–31.
[46] A. Wächter and L. T. Biegler, *On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming*, Math. Program., 106 (2006), pp. 25–57.
[47] R. A. Waltz, J. L. Morales, J. Nocedal, and D. Orban, *An interior algorithm for nonlinear optimization that combines line search and trust region steps*, Math. Program., 107 (2006), pp. 391–408.
[48] V. M. Zavala, *Computational Strategies for the Operation of Large-Scale Chemical Processes*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, 2008.
[49] V. M. Zavala and M. Anitescu, *Real-time nonlinear optimization as a generalized equation*, SIAM J. Control Optim., 48 (2010), pp. 5444–5467.
[50] V. M. Zavala and L. T. Biegler, *Nonlinear programming strategies for state estimation and model predictive control*, in Nonlinear Model Predictive Control, Lecture Notes in Control and Inform. Sci. 384, Springer, Berlin, 2009, pp. 419–432.