SIAM J. SCI. COMPUT.
Vol. 37, No. 4, pp. A1801–A1824

A FLEXIBLE ITERATIVE SOLVER FOR NONCONVEX, EQUALITY-CONSTRAINED QUADRATIC SUBPROBLEMS*

JASON E. HICKEN† AND ALP DENER†

Abstract. We present an iterative primal-dual solver for nonconvex equality-constrained quadratic optimization subproblems. The solver constructs the primal and dual trial steps from the subspace generated by the generalized Arnoldi procedure used in flexible GMRES (FGMRES). This permits the use of a wide range of preconditioners for the primal-dual system. In contrast with FGMRES, the proposed method selects the subspace solution that minimizes a quadratic penalty function over a trust region. Analysis of the method indicates the potential for fast asymptotic convergence near the solution, which is corroborated by numerical experiments. The results also demonstrate the effectiveness and efficiency of the method in the presence of nonconvexity. Overall, the iterative solver is a promising matrix-free linear algebra kernel for inexact-Newton optimization algorithms and is well-suited to partial differential equation–constrained optimization.

Key words. PDE-constrained optimization, Krylov subspace methods, matrix-free optimization, nonconvex programming, flexible preconditioning, inexact Newton

AMS subject classifications. 49M37, 65K05, 65F10, 90C06, 90C30, 90C55

DOI. 10.1137/140994496

1. Introduction.

Consider nonlinear equality-constrained optimization problems of the form

$$\min_x \quad f(x), \qquad \text{s.t.} \quad c(x) = 0, \tag{1.1}$$

where $x \in \mathbb{R}^n$ is the variable, $f : \mathbb{R}^n \to \mathbb{R}$ is a C^2-continuous objective, and $c : \mathbb{R}^n \to \mathbb{R}^m$ is a C^2-continuous constraint. Many optimization frameworks for solving (1.1) rely on the solution of quadratic optimization (QO) subproblems, for example,

$$\min_p \quad g^T p + \frac{1}{2} p^T \mathsf{W} p, \qquad \text{s.t.} \quad \mathsf{A}p + c = 0, \tag{1.2}$$

where $p \in \mathbb{R}^n$ is a candidate step, A is the Jacobian of the constraints, and g and W are, respectively, the gradient and Hessian of the Lagrangian of (1.1). This paper describes an efficient matrix-free algorithm for obtaining inexact solutions to (1.2).

Numerous algorithms have been proposed to solve the generic problems (1.1) and (1.2); see, for example, the monographs [20] and [5] and the references therein. Nevertheless, there remain applications for which conventional algorithms are not suitable for these problems. One such application is partial differential equation (PDE)-constrained optimization.

The present work is motivated by *reduced-space*, or "black-box," PDE-constrained optimization. In the reduced-space formulation, each evaluation of f and c entails solving a (discretized) PDE for the state variables. Similarly, evaluating derivatives of f and c requires the solution of additional PDEs. In contrast, *full-space* approaches

*Submitted to the journal's Methods and Algorithms for Scientific Computing section November 4, 2014; accepted for publication (in revised form) April 30, 2015; published electronically July 21, 2015. This work was supported by the National Science Foundation through grant 1332819.
http://www.siam.org/journals/sisc/37-4/99449.html

†Mechanical, Aerospace, and Nuclear Department, Rensselaer Polytechnic Institute, Troy, NY 12180 (hickej2@rpi.edu, denera@rpi.edu).

absorb the state variables into x and include the PDE as an additional constraint. This avoids the need to solve the state and adjoint equations at each optimization iteration, but increases the size of the problem. A more detailed discussion of the relative merits of the reduced- and full-space formulations can be found in [3] and [1].

Although our motivation is reduced-space PDE-constrained optimization, both reduced- and full-space formulations present similar challenges for conventional matrix-based optimization algorithms. To appreciate these challenges, consider the quadratic subproblem (1.2). In reduced-space formulations, obtaining a row of W and A requires the solution of a linearized PDE. Thus, the CPU time needed to form W and A can be prohibitive, even for modest values of n and m where conventional matrix-based algorithms would be ideal. In full-space formulations, forming W and A may be possible, but their size makes factorizations impractical.

To avoid the difficulties presented by forming or factoring W and A, researchers have proposed matrix-free optimization algorithms. An important issue faced by these algorithms is how to deal with the possibility that W is nonconvex in the null space of A. Most matrix-based algorithms address nonconvexity by factoring the Jacobian to determine a basis for its null space; however, factoring A is not an option for matrix-free algorithms, so researchers have investigated alternative ways to globalize these algorithms.

Recently, Heinkenschloss and Ridzal [14] proposed and analyzed a matrix-free trust-region algorithm that accommodates several forms of inexactness, including inexactness in the null-space basis. Their target application was full-space PDE-constrained optimization, but their algorithm is equally valid for reduced-space formulations that include (non-PDE) constraints. However, their composite-step approach relies on the (inexact) solution of augmented systems, which require additional linearized and adjoint PDE solutions that we would prefer to avoid in reduced-space PDE-constrained optimization.

In [4], Byrd, Curtis, and Nocedal propose a matrix-free framework for solving (1.1) based on an inexact-Newton method with termination tests tailored to the primal-dual subproblems. Their framework accommodates standard iterative methods and handles nonconvexity by modifying the Hessian when certain conditions are met. Modifying the Hessian usually requires a restart of the iterative method with an accompanying loss of valuable information regarding the Hessian and Jacobian. Consequently, restarting is also something we would like to avoid.

In addition to globalization, preconditioning is an important issue that matrix-free methods must address. We do not propose specific preconditioners in the present work, but we anticipate the need for nested iterative methods in our applications; see, for example, [8]. In general, such preconditioners are nonstationary and require so-called flexible iterative methods. To the best of our knowledge, flexible *and* globalized Krylov-based methods for (1.2) have not previously been discussed in the literature.

In summary, we would like an iterative solver for (1.2) that has the following properties.

1. The solver should yield an approximate solution using matrix-free operations, e.g., matrix-vector products and preconditioning operations.
2. The solver should handle nonconvexity without compromising superlinear asymptotic convergence when used in an inexact-Newton framework.
3. The solver should permit nonstationary preconditioners, i.e., it should be flexible in the sense that the preconditioner can change from one inner iteration to the next.

In this paper we describe how the Flexible Generalized Minimal RESidual (FGMRES) method [23] can be adapted to satisfy the above properties.

After presenting some additional notation in section 1.1, we provide a review of FGMRES and its underlying flexible Arnoldi algorithm in section 2. Section 3 describes the proposed algorithm and the rationale for aspects of its design. This is followed by a convergence analysis of the method in section 4. Results of numerical experiments are provided in section 5 and conclusions can be found in section 6.

1.1. Assumptions, definitions, and notation. The Lagrangian for (1.1) is given by

$$\mathcal{L}(x,\lambda) = f(x) + \lambda^T c(x),$$

where $\lambda \in \mathbb{R}^m$ denotes the Lagrange multipliers. The gradient and Hessian of the Lagrangian and the constraint Jacobian are defined by

$$\begin{aligned} g(x,\lambda) &\equiv \nabla_x \mathcal{L}(x,\lambda)^T, \\ \mathsf{W}(x,\lambda) &\equiv \nabla^2_{xx} \mathcal{L}(x,\lambda), \\ \mathsf{A}(x) &\equiv \nabla_x c(x), \end{aligned}$$

respectively. All vectors are column vectors, and, for the dimensions of the Jacobian, we follow the convention that $\mathsf{A} \in \mathbb{R}^{m\times n}$.

The first-order optimality conditions for (1.1) are

$$\begin{bmatrix} g(x,\lambda) \\ c(x) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{1.3}$$

When Newton's method is applied to (1.3), the linear primal-dual subproblems take the form

$$\begin{bmatrix} \mathsf{W} & \mathsf{A}^T \\ \mathsf{A} & 0 \end{bmatrix} \begin{bmatrix} p \\ d \end{bmatrix} = \begin{bmatrix} -g \\ -c \end{bmatrix}, \tag{1.4}$$

where g, c, W, and A are evaluated at the current (outer) solution and $p \in \mathbb{R}^n$ and $d \in \mathbb{R}^m$ denote the primal and dual Newton updates, respectively. When W is convex in the null space of A, the solution to (1.4) is also the solution to the QO problem (1.2).

We will sometimes write the primal-dual subproblem (1.4) in the more compact form

$$\mathsf{K}s = b, \tag{1.5}$$

where $\mathsf{K} \in \mathbb{R}^{(n+m)\times(n+m)}$ denotes the primal-dual, or KKT, matrix, and $b \in \mathbb{R}^{(n+m)}$ denotes the right-hand side of (1.4). In this work we assume that W is nonsingular and A has full rank; consequently, K is also nonsingular.

It will often be necessary to decompose primal-dual vectors and matrices into their constituent parts. For this purpose, a superscript p will be used to denote the primal part of the vector and a superscript d will denote the dual part. For example, the primal and dual subvectors of $v \in \mathbb{R}^{(n+m)}$ are, in MATLAB notation,

$$v^p \equiv v_{1:n}, \qquad \text{and} \qquad v^d \equiv v_{n+1:n+m},$$

respectively. Similarly, for a matrix $\mathsf{V}_j \in \mathbb{R}^{(n+m)\times j}$ we have

$$\mathsf{V}^p \equiv \mathsf{V}_{1:n,1:j}, \qquad \text{and} \qquad \mathsf{V}^d \equiv \mathsf{V}_{n+1,n+m,1:j}.$$

Unless stated otherwise, subscripts are used to indicate vectors or matrices at a particular (inner) iteration of the solver for (1.2). For example, z_j denotes the solution subspace vector from the jth iteration. At iteration j, the trial step for the quadratic subproblem is denoted by $s_j \in \mathbb{R}^{(n+m)}$ and its primal and dual subvectors are $p_j \in \mathbb{R}^n$ and $d_j \in \mathbb{R}^m$, respectively:

$$s_j = \begin{bmatrix} s_j^p \\ s_j^d \end{bmatrix} \equiv \begin{bmatrix} p_j \\ d_j \end{bmatrix}.$$

Based on this trial step, the residual of the primal-dual system is defined as

$$r_j \equiv b - \mathsf{K} s_j,$$

and it is clear that the residual's primal and dual subvectors are given by

$$\begin{aligned} r_j^p &\equiv -g - \mathsf{W} p_j - \mathsf{A}^T d_j, \\ r_j^d &\equiv -c - \mathsf{A} p_j. \end{aligned}$$

Let $\mathcal{M}_j : \mathbb{R}^{n+m} \to \mathbb{R}^{n+m}$ denote the preconditioner at iteration j. The preconditioner is represented as a general operator, rather than a matrix, because we want to permit the use of nonstationary iterative methods for the preconditioning operation.

2. Flexible Arnoldi and FGMRES. As mentioned in the introduction, the proposed algorithm builds on FGMRES [23]. Therefore, it is prudent to review FGMRES and its underlying (flexible) Arnoldi procedure in the context of the primal-dual system (1.4).

Suppose we have a set of linearly independent vectors, $\{z_i\}_{i=1}^j$; how we generate the z_i will be described below. FGMRES selects the trial solution s_j from the subspace[1] $\text{span}\{z_1, z_2, \ldots, z_j\}$ that minimizes the 2-norm of the residual r_j. In other words, if $\mathsf{Z}_j = \begin{bmatrix} z_1 z_2 \cdots z_j \end{bmatrix}$, then

$$s_j = \mathsf{Z}_j y_j, \qquad \text{where} \qquad y_j = \underset{y \in \mathbb{R}^j}{\text{argmin}} \, \|b - \mathsf{K} s_j\|.$$

To generate the subspace vectors $\{z_i\}_{i=1}^j$, FGMRES uses a generalized version of Arnoldi's procedure. Let $v_1 = b/\beta$, where $\beta = \|b\|$. At iteration j of the procedure, the preconditioner is applied to v_j to generate z_j. Subsequently, the matrix-vector product $\mathsf{K} z_j$ is evaluated and orthonormalized with respect to $\{v_i\}_{i=1}^j$ to obtain v_{j+1}. This yields the following relationship between the v_i and the z_i [23]:

$$h_{j+1,j} v_{j+1} = \mathsf{K} z_j - \sum_{i=1}^{j} h_{i,j} v_i,$$

where

$$h_{i,j} \equiv v_i^T (\mathsf{K} z_j), \quad i = 1, \ldots, j+1.$$

Introducing the matrix $\mathsf{V}_{j+1} = \begin{bmatrix} v_1 v_2 \cdots v_{j+1} \end{bmatrix}$, the above relationship can be written concisely as

$$\mathsf{K} \mathsf{Z}_j = \mathsf{V}_{j+1} \bar{\mathsf{H}}_j, \tag{2.1}$$

[1] More generally, FGMRES seeks a solution in the affine subspace $s_0 + \text{span}\{z_1, z_2, \ldots, z_j\}$.

where $\bar{\mathsf{H}}_j \in \mathbb{R}^{(j+1)\times j}$ denotes the upper Hessenberg matrix whose nonzero entries are $h_{i,j}$. Expanding (2.1) into its primal and dual parts, we find

$$\mathsf{W}\mathsf{Z}_j^p + \mathsf{A}^T\mathsf{Z}_j^d = \mathsf{V}_{j+1}^p\bar{\mathsf{H}}_j, \tag{2.2}$$

$$\mathsf{A}\mathsf{Z}_j^p = \mathsf{V}_{j+1}^d\bar{\mathsf{H}}_j. \tag{2.3}$$

As with GMRES, the residual of FGMRES at iteration j can be computed inexpensively, i.e., without matrix-vector products, using (2.1):

$$\begin{aligned} r_j = b - \mathsf{K}s_j &= b - \mathsf{K}\mathsf{Z}_j y_j \\ &= r_0 - \mathsf{V}_{j+1}\bar{\mathsf{H}}_j y_j \\ &= \mathsf{V}_{j+1}\left(\beta e_1 - \bar{\mathsf{H}}_j y_j\right), \end{aligned}$$

where e_1 is the first column of the $(j+1)\times(j+1)$ identity matrix. In addition, the orthogonality of the $\{v_i\}_{i=1}^{j+1}$ implies that the norm of the residual is the norm of the reduced-space residual. Thus,

$$\|r_j\| = \|\beta e_1 - \bar{\mathsf{H}}_j y_j\|,$$

and

$$y_j = \operatorname*{argmin}_{y\in\mathbb{R}^j} \|\beta e_1 - \bar{\mathsf{H}}_j y\|. \tag{2.4}$$

One disadvantage of FGMRES is that it requires us to store both V_{j+1} and Z_j. These vectors must be saved, because FGMRES cannot exploit the symmetry of K to develop short-term recurrences between the v_i and z_i. This is the price that must be paid to permit arbitrary preconditioners. In many reduced-space PDE-constrained optimization problems, the primal and dual dimensions are significantly smaller than the state variable dimension, and the cost of storing V_{j+1} and Z_j is relatively modest; however, the memory requirements may pose an issue in applications where n or m is comparable to the dimension of the state space.

For completeness, FGMRES is listed in Algorithm 1. The parameter η in the algorithm defines a commonly used relative-tolerance convergence criterion, although other convergence criteria are possible.

3. Flexible iterative solver for nonconvex problems. FGMRES is suitable for solving (1.2) when W is convex in the null space of A, but, in general, it will converge to a stationary point that is not necessarily the minimum. In this section we propose a novel solver that extends FGMRES to be able to handle nonconvex problems. The basic idea is to use the same solution subspace, Z_j, but to modify the subspace problem (2.4) that defines y_j.

3.1. Penalty-based subspace problem for the primal step. We seek an approximate solution to the quadratic subproblem (1.2) by minimizing a quadratic penalty function. Moreover, we require this approximate solution to lie in the subspace generated by the flexible Arnoldi procedure. In other words, we solve

$$\min_{p\in\operatorname{span}\{Z_j^p\}} Q(p,\mu) \equiv g^T p + \frac{1}{2}p^T\mathsf{W}p + \frac{\mu}{2}\left(\mathsf{A}p + c\right)^T\left(\mathsf{A}p + c\right), \tag{3.1}$$

where $\mu \geq 0$ is the penalty parameter.

Algorithm 1: FGMRES.

Data: b, $\eta \in (0,1)$
Result: s, the inexact solution satisfying $\|\mathsf{K}s + b\| \leq \eta\|b\|$

compute $r_0 = b$, $\beta = \|r_0\|$, and $v_1 = r_0/\beta$
for $j = 1, 2, 3, \ldots$ **do**
  $z_j \leftarrow \mathcal{M}_j(v_j)$
  $v_{j+1} \leftarrow \mathsf{K}z_j$
  for $i = 1, \ldots, j$ **do** (Modified Gram–Schmidt)
    $h_{i,j} = v_{j+1}^T v_j$
    $v_{j+1} \leftarrow v_{j+1} - h_{i,j} v_j$
  end
  compute $h_{j+1,j} = \|v_{j+1}\|$, and $v_{j+1} \leftarrow v_{j+1}/h_{j+1,j}$
  if $\|r_i\| \leq \eta\beta$ **then** (check convergence)
    compute $y_j = \operatorname{argmin}_y \|\beta e_1 + \mathsf{H}y\|$
    $s \leftarrow \mathsf{Z}_j y_j$
    return
  end
end

The choice of (3.1) may be troubling to some readers, because quadratic penalty methods are known to suffer from ill-conditioning as $\mu \to \infty$ [20]. Recall, however, that the unbounded growth of the penalty parameter is only necessary if we require an *exact solution* to (1.2). In an inexact-Newton framework, where the quadratic subproblem is solved approximately, nonlinear convergence is typically achieved using modest values of μ.

A key feature of the penalty problem (3.1) is that it can be solved matrix-free using information already available from the Arnoldi relation (2.1), as we now show. First, consider the terms that make up the Hessian of $Q(p,\mu)$. Let W_Z denote the Hessian of the Lagrangian in the primal subspace $\operatorname{span}(\mathsf{Z}_j^p)$. An expression for W_Z follows immediately from the Arnoldi relation (2.1) (see also (2.2) and (2.3)):

$$\mathsf{W}_\mathsf{Z} \equiv \left(\mathsf{Z}_j^p\right)^T \mathsf{W}\mathsf{Z}_j^p = \mathsf{Z}_j^T \mathsf{V}_{j+1}\bar{\mathsf{H}}_j - \left(\mathsf{Z}_j^d\right)^T \mathsf{V}_{j+1}^d \bar{\mathsf{H}}_j - \bar{\mathsf{H}}_j^T \left(\mathsf{V}_{j+1}^d\right)^T \mathsf{Z}_j^d. \tag{3.2}$$

Note that the right-hand side of this identity involves only inner products between Z_j and V_{j+1}, and small matrix-matrix products. In particular, no Hessian-vector or Jacobian-vector products are required.

In addition to W_Z, the Hessian of $Q(p,\mu)$ has a contribution due to $\mu\mathsf{A}^T\mathsf{A}$. This term can also be simplified in the subspace of the solution using (2.3):

$$\left(\mathsf{A}^T\mathsf{A}\right)_\mathsf{Z} \equiv \left(\mathsf{Z}_j^p\right)^T \mathsf{A}^T\mathsf{A}\mathsf{Z}_j^p = \bar{\mathsf{H}}_j^T \left(\mathsf{V}_{j+1}^d\right)^T \mathsf{V}_{j+1}^d \bar{\mathsf{H}}_j. \tag{3.3}$$

As with W_Z, this small matrix can be evaluated using a few inner products and small matrix-matrix products.

Finally, the terms that make up the gradient of $Q(p,\mu)$ can also be evaluated inexpensively. If we use g_Z and $\left(\mathsf{A}^T c\right)_\mathsf{Z}$ to denote, respectively, the product of g and $\mathsf{A}^T c$ with $(\mathsf{Z}_j^p)^T$, then

$$g_{\mathsf{Z}} \equiv \left(\mathsf{Z}_j^p\right)^T g = -\left(\mathsf{Z}_j^p\right)^T (\beta v_1^p), \quad \text{and} \tag{3.4}$$

$$\left(\mathsf{A}^T c\right)_{\mathsf{Z}} \equiv \left(\mathsf{Z}_j^p\right)^T \mathsf{A}^T c = -\bar{\mathsf{H}}_j^T \left(\mathsf{V}_{j+1}^d\right)^T (\beta v_1^d). \tag{3.5}$$

Putting the pieces together, the solution to the penalty problem (3.1) is equivalent to $p_j = \mathsf{Z}_j^p y_j^p$ where

$$\begin{aligned} y_j^p &= \underset{y\in\mathbb{R}^j}{\operatorname{argmin}} \quad Q(\mathsf{Z}_j^p y, \mu) \\ &= \underset{y\in\mathbb{R}^j}{\operatorname{argmin}} \quad \left[g_{\mathsf{Z}} + \mu\left(\mathsf{A}^T c\right)_{\mathsf{Z}}\right]^T y + \frac{1}{2} y^T \left[\mathsf{W}_{\mathsf{Z}} + \mu\left(\mathsf{A}^T \mathsf{A}\right)_{\mathsf{Z}}\right] y. \end{aligned} \tag{3.6}$$

Note that we have dropped constant terms from $Q(p, \mu)$, which do not impact the solution.

If W is nonconvex in the null space of A, then the solution to the target QO (1.2) is unbounded. Nonconvexity also manifests itself in the solution of the subspace penalty problem (3.6). While the probability that one of the z_i is *exactly* in the null space of A is low, the subspace vectors can, in practice, become close to the null space. When this happens $\|y_j^p\|$ and, consequently, $\|p_j\|$ can become large.

To prevent large stepsizes, the proposed algorithm accommodates nonconvexity by adding the trust-region constraint $\|p_j\| \le \Delta$ to (3.1). Thus, the actual subproblem that defines y_j^p is

$$\begin{aligned} \min_{y\in\mathbb{R}^j} \quad & \left[g_{\mathsf{Z}} + \mu\left(\mathsf{A}^T c\right)_{\mathsf{Z}}\right]^T y + \frac{1}{2} y^T \left[\mathsf{W}_{\mathsf{Z}} + \mu\left(\mathsf{A}^T \mathsf{A}\right)_{\mathsf{Z}}\right] y \\ & \text{s.t.} \quad \|\mathsf{Z}_j^p y\| \le \Delta. \end{aligned} \tag{3.7}$$

In practice, we solve this (small) subproblem by applying the Moré and Sorensen algorithm [19].

3.2. Subspace problem for the dual step. The subproblem (3.7) determines the primal step p_j only; an alternative subproblem is needed for the dual step d_j. Finding a suitable dual-step subproblem proved to be challenging. We considered several approaches.

- Assume that $\lim_{j\to\infty} \mu = \infty$ and that the primal problem converges. Then the asymptotic result [20, p. 503] $\lim_{j\to\infty} \mu\left(\mathsf{A}p_j + c\right) = d$ suggests the inexact dual step

$$d_j = \mu\left(\mathsf{A}p_j + c\right);$$

  however, not surprisingly, this dual step does not perform well for inexact solutions and is sensitive to the growth schedule adopted for μ. Moreover, it is not appropriate when the trust-region constraint is active.
- Ideally, we would like d_j such that

$$d_j = \underset{d}{\operatorname{argmin}} \|g + \mathsf{W}p_j + \mathsf{A}^T d\|.$$

  Unfortunately, if we choose d_j of the form $\mathsf{Z}_j^d y_j^d$, the residual $g + \mathsf{W}p_j + \mathsf{A}^T d_j$ is not computable based on the Arnoldi relation. This is because the primal and dual subspace solutions, y_j^p and y_j^d, are different, in general.

One can approximate the multiplier residual by projecting it onto Z_j^p and choosing $d_j = \mathsf{V}_{j+1}^d y_j$. Setting the resulting subspace residual to zero leads to the subproblem

$$\mathsf{Z}_j^T \left(g + \mathsf{W}p_j + \mathsf{A}^T d_j\right) = g_\mathsf{Z} + \mathsf{W}_\mathsf{Z} y_j + \left[\bar{\mathsf{H}}_j^T \left(\mathsf{V}_{j+1}^d\right)^T \mathsf{V}_{j+1}^d\right] y_j^d = 0,$$

where $y_j^d \in \mathbb{R}^{j+1}$ in this case. Perversely, this is an underdetermined subproblem trying to mimic an overdetermined subproblem. We investigated several solution strategies, including pseudoinverses based on dropping singular values below a threshold, but the convergence of the resulting methods was generally erratic and slow.

- Solving the primal subproblem (3.7) produces the Lagrange multiplier λ, corresponding to the trust-region constraint $\|p_j\| \leq \Delta$. This can then be used as a regularization parameter for the Hessian in the primal-dual problem:

$$\begin{bmatrix} (\lambda \mathsf{I} + \mathsf{W}) & \mathsf{A}^T \\ \mathsf{A} & 0 \end{bmatrix} \begin{bmatrix} p \\ d \end{bmatrix} = \begin{bmatrix} -g \\ -c \end{bmatrix}.$$

  Setting the primal-dual solution to $s_j = \mathsf{Z}_j y_j$, projecting the residual onto V_j, and setting the result to zero, yields the following subproblem:

$$\left[\lambda \left(\mathsf{V}_j^p\right)^T \mathsf{Z}_j^p + \mathsf{H}_j\right] y_j = \beta e_1,$$

  where $\mathsf{H}_j \in \mathbb{R}^{j \times j}$ is equal to the first j rows of $\bar{\mathsf{H}}_j$. In general, setting the dual step to $\mathsf{Z}_j^d y_j$ resulted in disappointing performance.

Ultimately, the FGMRES solution was adopted for the dual step. That is, we set $d_j = \mathsf{Z}_j^d y_j^d$ with y_j^d given by (2.4). This choice produces excellent asymptotic convergence. A disadvantage of using the FGMRES dual solution is the risk that it may "pull" the primal step toward stationary points that are not local minima. We have found that this risk can be alleviated through the outer globalization method and the update used for μ.

Finally, we emphasize that our investigations of the dual-step methods were undertaken in the context of a particular trust-region framework (described below). Therefore, it is possible that further analysis and experimentation may lead to an effective dual-step approach based on the alternative strategies described above.

3.3. Summary of the FLECS iterative solver. Algorithm 2 lists the proposed iterative method, which we refer to as the FLexible Equality-Constrained Subproblem solver, or FLECS for short. The primary difference between FLECS and FGMRES can be found on line 11, where the subspace problems are solved. The calculation of the subspace steps y_j^p and y_j^d is listed separately in Algorithm 3. Note that the subscript j is omitted from y^p and y^d in Algorithm 3, because it is implicit in the size of the input matrices.

Some remarks on stopping criteria are necessary. One possible criterion is that the norm of the penalty's gradient be reduced below some threshold, i.e.,

$$\|(\mathsf{W} + \mu \mathsf{A}^T \mathsf{A})p + g + \mu \mathsf{A}^T c\| \leq \eta.$$

However, computing this norm requires explicit evaluation of the penalty gradient, which, in turn, requires at least two linear PDE solutions in our applications. Moreover, the gradient norm is not a suitable criterion when the trust-radius constraint is active.

Algorithm 2: FLECS.

Data: b, $\eta \in (0,1)$, $\mu \geq 0$ and $\Delta > 0$.
Result: s, an inexact solution to (1.2)

compute $r_0 = b$, $\beta = \|r_0\|$, and $v_1 = r_0/\beta$
compute initial norms, $\omega_0 = \|r_0^p\|$ and $\gamma_0 = \|r_0^d\|$
for $j = 1, 2, 3, \ldots$ **do**
  $z_j \leftarrow \mathcal{M}_j(v_j)$
  $v_{j+1} \leftarrow \mathsf{K} z_j$
  for $i = 1, \ldots, j$ **do** (Modified Gram–Schmidt)
    $h_{i,j} = v_{j+1}^T v_j$
    $v_{j+1} \leftarrow v_{j+1} - h_{i,j} v_j$
  end
  compute $h_{j+1,j} = \|v_{j+1}\|$, and $v_{j+1} \leftarrow v_{j+1}/h_{j+1,j}$
  solve for y_j^p and y_j^d using Algorithm 3
  compute FGMRES primal and dual residual norms:

$$\omega_j = \|V_{j+1}^p \left(\beta e_1 - \bar{\mathsf{H}}_j y_j^d\right)\|,$$
$$\gamma_j = \|V_{j+1}^d \left(\beta e_1 - \bar{\mathsf{H}}_j y_j^d\right)\|.$$

  if $\omega_j \leq \eta\omega_0$ *and* $\gamma_j \leq \eta\gamma_0$ **then** (check subspace quality)
    compute primal step: $p_j \leftarrow \mathsf{Z}_j^p y_j^p$
    compute dual step: $d_j \leftarrow \mathsf{Z}_j^d y_j^d$
    return
  end
end

Algorithm 3: FLECS subspace solution.

Data: $\Delta > 0$, $\mu \geq 0$, β, and the matrices

$$\begin{aligned} \bar{\mathsf{H}} &\in \mathbb{R}^{(j+1)\times j}, & \mathsf{V}^T\mathsf{Z} &\in \mathbb{R}^{(j+1)\times j}, \\ \left(\mathsf{V}^p\right)^T \mathsf{Z}^p &\in \mathbb{R}^{(j+1)\times j}, & \left(\mathsf{V}^d\right)^T \mathsf{Z}^d &\in \mathbb{R}^{(j+1)\times j}, \\ \left(\mathsf{Z}^p\right)^T \mathsf{Z}^p &\in \mathbb{R}^{j\times j}, & \left(\mathsf{V}^d\right)^T \mathsf{V}^d &\in \mathbb{R}^{(j+1)\times(j+1)}. \end{aligned}$$

Result: The primal and dual subspace solutions, y^p and y^d, resp.

compute W_Z, $\left(\mathsf{A}^T\mathsf{A}\right)_\mathsf{Z}$, g_Z and $\left(\mathsf{A}^T c\right)_\mathsf{Z}$ using (3.2), (3.3), (3.4), and (3.5), respectively.
solve the trust-region primal subproblem for y^p:

$$\min_{y\in\mathbb{R}^j} \quad \left[g_\mathsf{Z} + \mu\left(\mathsf{A}^T c\right)_\mathsf{Z}\right]^T y + \frac{1}{2} y^T \left[\mathsf{W}_\mathsf{Z} + \mu\left(\mathsf{A}^T\mathsf{A}\right)_\mathsf{Z}\right] y, \qquad \text{s.t.} \qquad \|Z^p y\| \leq \Delta.$$

solve the dual subproblem $y^d = \operatorname{argmin}_{y\in\mathbb{R}^j} \|\beta e_1 - \bar{\mathsf{H}}_j y\|$.

TABLE 1
Comparison of the computational cost and memory requirements of FGMRES and FLECS, where J denotes the number of iterations.

		FGMRES	**FLECS**
Cost	**dots** (dim. n)	$\frac{1}{2}J(J+1)+2$	$\frac{5}{2}J(J+1)+2$
	dots (dim. m)	$\frac{1}{2}J(J+1)+2$	$\frac{(5J+2)(J+1)}{2}+2$
	axpbys	$\frac{1}{2}J(J+3)+2$	$\frac{1}{2}J(J+3)+2$
	matvecs	J	J
Memory		$(2J+1)(m+n)$	$(2J+2)(m+n)$

For this work, we use the (normalized) FGMRES primal and dual residual norms to determine when to stop. These do not measure convergence of the FLECS penalty problem per se, but they do indicate the quality of the subspace vectors. In particular, if FGMRES feasibility is sufficiently small, we can always find a μ large enough to ensure that the FLECS step is at least as feasible. This claim is supported by the theory presented in the next section.

For completeness, Table 1 compares the computational cost and memory requirements of FGMRES and FLECS for the same number of iterations, J. In the table, dots denotes inner products, axpbys denotes scalar-vector-plus-vector operations, and matvecs denotes matrix-vector products. The number of preconditioning operations is equal to the number of matvecs, so it is not listed. Inner products are separated into primal and dual parts, since FLECS requires slightly more dual products.

From Table 1, we see that the two algorithms have almost identical cost and memory requirements, except that FLECS requires approximately five times the number of inner products; however, for the target applications the dominant costs are the matrix-vector and preconditioning operations, which are the same for both algorithms.

4. Analysis. If the Hessian is convex in the null space of the Jacobian and FGMRES converges, what can we say about the inexact FLECS solution? To answer this question we need the following lemma regarding the Hessian of the penalty function.

LEMMA 1. *If W is positive definite in the null space of A, then there exists $\mu^\star$ such that $\forall \mu > \mu^\star$, $\mathsf{W} + \mu \mathsf{A}^T\mathsf{A}$ is positive definite.*

The proof of this lemma can be found in Appendix A.

Our first main result concerns the convergence of FLECS in the convex case with no trust-radius constraint, i.e., $\Delta = \infty$ in (3.7). In the following, $p_j = \mathsf{Z}_j^p y_j$ and $p_j^F = \mathsf{Z}_j^p y_j^F$ denote the FLECS and FGMRES primal steps at iteration j, respectively. Similarly, we use $d_j = d_j^F$ to denote the dual step.

THEOREM 2. *Assume that W is positive definite in the null space of A. Furthermore, assume that the flexible Arnoldi method does not break down for any $j \leq n$, and for any $\delta > 0$ there exists a k such that the FGMRES residual norm satisfies $\|r_j\| < \delta$ for all $j \geq k$. Then, for any $\tau \geq 0$, there exists a $\mu \geq 0$ and iteration j such that the FLECS solution satisfies*

$$\|\mathsf{A}p_j + c\| \leq \tau,$$
$$\textit{and} \qquad \|\mathsf{W}p_j + g + \mathsf{A}^T d_j\| \leq \tau.$$

Proof. We will show that the FLECS and FGMRES primal steps can be brought arbitrarily close to one another by choosing sufficiently large j and μ. To do this, we make use of the inequality

$$\|p_j - p_j^F\| \le \|p_j - p(\mu)\| + \|p(\mu) - p\| + \|p - p_j^F\|, \tag{4.1}$$

where $p(\mu) = \operatorname{argmin}_p Q(p, \mu)$ is the optimum of the quadratic penalty over all $\mathbb{R}^n$, and p is the solution to the equality-constrained QO (1.2). We will show that each of these norms can be made arbitrarily small under the assumptions.

The last norm on the right, $\|p - p_j^F\|$, can be bounded using the identity $\mathsf{K}(p - p_j^F) = r_j^p$ and our earlier assumption that the primal-dual matrix is invertible:

$$\|p - p_j^F\| = \| \left(\mathsf{K}^{-1} r_j\right)^p \| \le \|\mathsf{K}^{-1}\| \|r_j\|.$$

The FGMRES residual can be made arbitrarily small by assumption; therefore, $\forall \epsilon > 0$, there exists an iteration k such that $\|p - p_j^F\| \le \epsilon$ whenever $j > k$.

The second term on the right of (4.1) is the difference between the solutions of the quadratic-penalty minimization and the QO problem. It is known (see, for example, [20, p. 502]) that this difference tends to zero as $\mu \to \infty$. Consequently, $\forall \epsilon > 0$, there exists $\mu > 0$ such that

$$\|p(\mu) - p\| \le \epsilon.$$

The first term, $\|p_j - p(\mu)\|$, is the error in the FLECS solution to $\min_p Q(p, \mu)$. Observe that the FLECS algorithm finds the solution $p_j = \mathsf{Z}_j y$ that ensures $\nabla_p Q$ is orthogonal to $\operatorname{span}\{\mathsf{Z}_j\}$ (i.e., the Galerkin solution):

$$\mathsf{Z}_j^T \nabla_p Q = \mathsf{Z}_j^T \left[\left(\mathsf{W} + \mu \mathsf{A}^T \mathsf{A}\right) \mathsf{Z}_j y + g + \mu \mathsf{A}^T c \right] = 0.$$

If we ensure $\mu > \mu^\star$, we have that $\mathsf{W} + \mu \mathsf{A}^T \mathsf{A}$ is positive definite by Lemma 1. Consequently, we can make use of the optimality of orthogonal projection to conclude that the FLECS error is minimized in the energy norm defined by $\mathsf{W} + \mu \mathsf{A}^T \mathsf{A}$ [24, p. 113]:

$$\begin{aligned} \|p_j - p(\mu)\|^2_{\mathsf{W}+\mu\mathsf{A}^T\mathsf{A}} &= (p_j - p(\mu))^T \left(\mathsf{W} + \mu \mathsf{A}^T \mathsf{A}\right) (p_j - p(\mu)) \\ &= \min_{p \in \operatorname{span}\{\mathsf{Z}_j\}} (p - p(\mu))^T \left(\mathsf{W} + \mu \mathsf{A}^T \mathsf{A}\right) (p - p(\mu)). \end{aligned}$$

The energy-norm error is nonincreasing in j, since the solution space is expanding. In particular, the energy-norm error vanishes once Z_j spans all of $\mathbb{R}^n$, which it must do eventually since the flexible Arnoldi method does not break down by assumption. So, for any $\epsilon > 0$, there exists an iteration k such that

$$\|p_j - p(\mu)\| \le \epsilon, \qquad \forall j > k,$$

where we have used the equivalence of finite-dimensional norms.

Thus, we have shown that each of the terms on the right of inequality (4.1) can be made arbitrarily small provided μ and j are sufficiently large.

Now, the constraint residual based on the FLECS solution satisfies

$$\begin{aligned} \|\mathsf{A} p_j + c\| &\le \left\|\mathsf{A} p_j^F + c\right\| + \|\mathsf{A}(p_j - p_j^F\| \\ &\le \|r_j^d\| + \|\mathsf{A}\| \|p_j - p_j^F\|. \end{aligned}$$

Both terms on the right of this inequality can be made arbitrarily small: by assumption, the FGMRES dual residual, $\|r_j^d\|$, can be made arbitrarily small by choosing j sufficiently large; and, as we showed above, the difference $\|p_j - p_j^F\|$ can be made

arbitrarily small by choosing both μ and j sufficiently large. The result then follows, because $\|\mathsf{A}\|$ is bounded.

Similarly, the norm of the gradient of the Lagrangian evaluated at the FLECS solution satisfies (recall $d_j = d_j^F$)

$$\begin{aligned} \|\mathsf{W}p_j + g + \mathsf{A}^T d_j\| &\leq \|\mathsf{W}p_j^F + g + \mathsf{A}^T d_j^F\| + \|\mathsf{W}(p_j - p_j^F\| \\ &\leq \|r_j^p\| + \|\mathsf{W}\|\|p_j - p_j^F\|. \end{aligned} \tag{4.2}$$

As before, we can choose the iteration index and penalty parameter sufficiently large to make the FGMRES residual and the difference $\|p_j - p_j^F\|$ arbitrarily small. Thus, since $\|\mathsf{W}\|$ is bounded, the norm of the gradient can be made less than $\tau > 0$. ∎

Next, we consider the case where W is not necessarily positive definite in the null space of A.

THEOREM 3. *Assume that the primal-dual matrix K is nonsingular. Furthermore, for any $\epsilon > 0$, we assume that there exists an iteration k such that the FGMRES residual satisfies $\|r_j\| < \epsilon$ for all $j \geq k$. If FLECS uses a finite trust radius $\Delta > 0$, then, for any $\tau \geq 0$, there exists a $\mu \geq 0$ and iteration j such that the FLECS solution satisfies*

$$\|\mathsf{A}p_j + c\| \leq \tau.$$

Proof. By definition of the FLECS solution, we have that $Q(p_j, \mu) \leq Q(p_j^F, \mu)$, or, expanding and rearranging this inequality,

$$\|\mathsf{A}p_j + c\|^2 \leq \frac{2}{\mu}\left[g^T p_j^F + \frac{1}{2}\left(p_j^F\right)^T \mathsf{W}p_j^F - g^T p_j - \frac{1}{2}p_j^T \mathsf{W}p_j\right] + \left\|\mathsf{A}p_j^F + c\right\|^2. \tag{4.3}$$

As in the convex case, we can make the second term on the right of (4.3) arbitrarily small by choosing j sufficiently large. If we can show that the term multiplied by $2/\mu$ is bounded independent of j and μ, then the result will follow by making μ sufficiently large.

Now, the quadratic objective evaluated at the FGMRES solution is bounded above because

$$g^T p_j^F + \frac{1}{2}\left(p_j^F\right)^T \mathsf{W}p_j^F \leq \|g\|\|p_j^F\| + \frac{\lambda_{\max}}{2}\|p_j^F\|^2,$$

where $\lambda_{\max}$ is the maximum eigenvalue of W. The expression on the right is bounded above, because the FGMRES solution is bounded; the residual is bounded and K is assumed nonsingular.

To bound the quadratic objective evaluated at the FLECS solution, we note that

$$-g^T p_j - \frac{1}{2}\left(p_j\right)^T \mathsf{W}p_j \leq \|g\|\|p_j\| - \frac{\lambda_{\min}}{2}\|p_j\|^2,$$

where $\lambda_{\min}$ is the minimum eigenvalue of W. In this case, the right-hand side is bounded above due to the trust-region constraint $\|p_j\| \leq \Delta$.

In summary, the term in brackets on the right of (4.3) is bounded above, so we can choose μ and j sufficiently large to satisfy

$$\|\mathsf{A}p_j + c\|^2 \leq \tau^2$$

for all $\tau > 0$, as desired. ∎

4.1. Discussion. Theorem 2 suggests that the FLECS steps will yield fast asymptotic convergence near the solution of (1.1), where W is positive definite in the null space of the constraints. Unfortunately, the theorem does not tell us if FLECS will require significantly more iterations than FGMRES to achieve comparable convergence. Numerical experiments indicate that the number of iterations is similar, and the results presented below demonstrate that FLECS does indeed produce excellent asymptotic performance.

In both the convex and nonconvex situations, the theory suggests that we increase μ dynamically within FLECS. The risk with this approach, especially during the early nonlinear iterations, is that the steps can be drawn toward stationary points. Our experience is that gradually increasing μ outside of FLECS is more robust and effective. As long as μ increases in the outer iterations, it will eventually be large enough to produce the desired asymptotic performance.

Actually, it is not clear that μ needs to increase indefinitely, and we have often obtained excellent results with a fixed (relatively small) value of μ. This may reflect the close relationship between FLECS and augmented Lagrangian methods: the quadratic penalty used in FLECS is a quadratic approximation to the augmented Lagrangian for (1.1) with the term $\mu c^T \nabla^2_{xx} c$ missing from the augmented Lagrangian Hessian. Thus, since augmented Lagrangian methods can converge for finite penalty values, we might expect similar behavior from FLECS.

5. Numerical experiments. In this section we present the results of numerical experiments, including synthetic QOs and a nonlinear PDE-constrained optimization. In all cases, the primal and dual dimensions are small, on the order of 100. We emphasize that such sizes are typical of reduced-space PDE-constrained optimization problems in engineering applications and do not diminish the usefulness of matrix-free algorithms; recall that forming each row of the Hessian and Jacobian requires a PDE solution in the reduced space. The application of FLECS to a large-scale aerodynamic shape optimization problem is presented in [9].

5.1. Synthetic QOs. Our first numerical experiments exercise the FLECS algorithm on a set of synthetic QOs problems. Synthetic QO are used to verify that FLECS performs well relative to FGMRES when the problem is convex; this is important to ensure good asymptotic performance. Synthetic QOs are also used to assess the performance of FLECS on nonconvex problems; since FGMRES fails on nonconvex problems, we draw comparisons with a composite-step approach.

The optimization algorithms are applied to the synthetic QOs without preconditioning, so the condition number of K is kept modest to reflect values observed in preconditioned systems. That said, the condition number does not have a significant impact on the presented results, since the performance of FLECS is measured relative to other unpreconditioned solvers.

The synthetic QOs were generated using the following procedure.

1. The number of variables was drawn from a (pseudo)random uniform distribution such that $n \in [10, 100]$. Similarly, the number of constraints was drawn such that $m \in [1, n-1]$.
2. A set of Hessian eigenvalues $\{\lambda_i\}_{i=1}^n$ was generated from a continuous uniform distribution such that $\max_i \lambda_i = 1$ and $\min_i \lambda_i = 1/\kappa$, where $\kappa = 10^4$ was chosen as the condition number for W. Note that the condition number of K is larger than W, typically by an order of magnitude, in the tests.
3. The sign of the first m eigenvalues was assigned randomly. To investigate performance on convex problems, the last $n - m$ eigenvalues were left

positive (henceforth referred to as the convex case). For nonconvex performance investigations, at least one of the last $n-m$ eigenvalues was forced to be negative (the nonconvex case).

4. A set of n eigenvectors E was generated by orthonormalizing a random $n \times n$ matrix (entries drawn from a continuous uniform distribution), and the Hessian was then constructed as

$$\mathsf{W} \equiv \mathsf{E}\Lambda\mathsf{E}^T,$$

where $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$.

5. The constraint Jacobian was constructed by generating a random $m \times m$ matrix R and multiplying this from the right by the first m eigenvectors transposed:

$$\mathsf{A} \equiv \mathsf{R}\mathsf{E}^T_{:,1:m}.$$

6. The gradient was defined as $g = \mathsf{W}\hat{p}$, where $\hat{p}$ was constructed with random entries and normalized to unit length.
7. The vector c was defined as $-\mathsf{A}p_\perp$, where $p_\perp$ is a vector that is perpendicular to the null space and has a (random) length less than $1/2$. This choice for c ensures that a trust-radius constraint $\|p\| \leq 1$ is consistent with the constraints.

The MATLAB scripts that generate the QO problems, as well as functions that implement the algorithms, can be found in the public git repository [15].

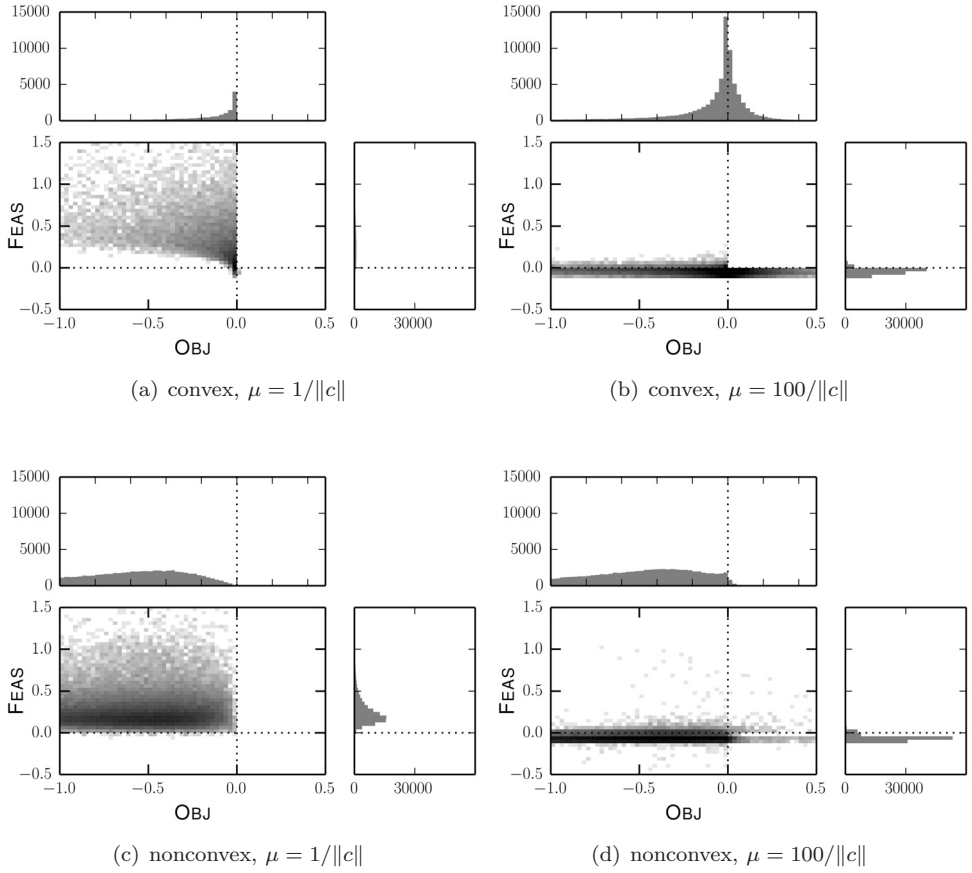
5.1.1. Convex Hessian in the null space. For the convex case, FLECS was compared against FGMRES on 10^5 unique samples. Both solvers used a relative tolerance of $\eta = 0.1$ for the primal and dual residual norms; in fact, both solvers exit after the same number of iterations, because FLECS uses the FGMRES residual norms for its convergence criterion. The trust radius used by FLECS was set to 100 times the length of the FGMRES solution, to avoid activating the trust-region constraint. Results were gathered for two values of the penalty parameter, $\mu = 1/\|c\|$ and $\mu = 100/\|c\|$, to investigate its impact on the FLECS solution.

Figures 1(a) and 1(b) plot the convex-case results for the two values of μ. The figures include a two-dimensional histogram of normalized feasibility versus the normalized objective, which are defined as

$$\textsc{Feas}(p_j, p_{\mathrm{ref}}) = \frac{\|\mathsf{A}p_j + c\| - \|\mathsf{A}p_{\mathrm{ref}} + c\|}{\|c\| - \|\mathsf{A}p_{\mathrm{ref}} + c\|}, \quad \text{and} \tag{5.1}$$

$$\textsc{Obj}(p_j, p_{\mathrm{ref}}) = \frac{\left(g^T p_j + \frac{1}{2}p_j^T \mathsf{W} p_j\right) - \left(g^T p_{\mathrm{ref}} + \frac{1}{2}p_{\mathrm{ref}}^T \mathsf{W} p_{\mathrm{ref}}\right)}{\left|g^T p_{\mathrm{ref}} + \frac{1}{2}p_{\mathrm{ref}}^T \mathsf{W} p_{\mathrm{ref}}\right|}, \tag{5.2}$$

respectively, where p_j is the FLECS solution and p_{ref} is the reference solution (the FGMRES solution here). The shade of each small box in the two-dimensional histograms indicates the relative number of results that fall in that box's range of objective and feasibility measures. The shade can be regarded as the value of a probability density function. The figures also include one-dimensional histograms for the normalized objective (upper histogram) and normalized feasibility (histogram to the right of the two-dimensional histogram).



(a) convex, $\mu = 1/\|c\|$

(b) convex, $\mu = 100/\|c\|$

(c) nonconvex, $\mu = 1/\|c\|$

(d) nonconvex, $\mu = 100/\|c\|$

FIG. 1. *Histograms of the normalized objective and normalized feasibility for the synthetic QOs.*

The dotted lines in the histograms separate the FLECS results that improve upon the FGMRES results from those that perform worse. For example, any FLECS result with a normalized objective less than zero has performed better than FGMRES with respect to this criterion. The case is similar for the normalized feasibility. Ideally, we want results that improve in both criteria, i.e., the results in the lower-left quadrant of the two-dimensional histogram. Table 2 summarizes the percentage of cases that fall in each of the four quadrants.

The results demonstrate, as one would expect, that FLECS does not improve upon the feasibility of the FGMRES solution if μ is not sufficiently large. Indeed, for $\mu = 1/\|c\|$, 83.2% of FLECS solutions have FEAS > 1, indicating a loss of feasibility relative to the initial guess $p = 0$. However, the number of FLECS solutions that have both OBJ and FEAS greater than zero is less than 0.3% for both values of μ.

For $\mu = 100/\|c\|$, FLECS produces a better solution than FGMRES more than 50% of the time. Moreover, for this value of μ, FLECS improves on the feasibility in approximately 83% of the cases. These results suggest that the asymptotic performance of an inexact-Newton solver based on FLECS will be comparable to a solver based on FGMRES, provided μ is sufficiently large. Evidence for this is presented for a PDE-constrained optimization example below.

TABLE 2
Percentage of QO results in each quadrant of the quality measures.

	OBJ > 0		OBJ ≤ 0	
	FEAS > 0	FEAS ≤ 0	FEAS > 0	FEAS ≤ 0
convex, $\mu = 1/\|c\|$	0.28%	0.01%	99.04%	0.67%
convex, $\mu = 100/\|c\|$	0.20%	32.38%	16.44%	50.99%
nonconvex, $\mu = 1/\|c\|$	0.02%	0.0%	99.82%	0.16%
nonconvex, $\mu = 100/\|c\|$	0.44%	2.10%	7.07%	90.40%

5.1.2. Nonconvex Hessian in the null space. For nonconvex QOs, FLECS was compared against a composite-step trust-region algorithm [22], which we briefly describe (the algorithm is provided in the repository [15]). The quasi-normal step, $p_\perp$, is found by inexactly solving for the minimum-norm solution that satisfies the constraint. In particular, the normal step satisfies

$$\left\| \begin{bmatrix} \mathsf{I} & \mathsf{A}^T \\ \mathsf{A} & 0 \end{bmatrix} \begin{bmatrix} p_\perp \\ d_\perp \end{bmatrix} + \begin{bmatrix} 0 \\ c \end{bmatrix} \right\| \leq \eta \|c\|,$$

where the relative tolerance is set to $\eta = 0.1$. We use GMRES, which is mathematically equivalent to MINRES in this case, to solve for the normal step. The length of $p_\perp$ is shortened to $\Delta_\perp = 0.8\Delta$ if it exceeds this trust radius.

Once the normal step is computed, the tangential step, $p_\parallel$, is found using the projected conjugate gradient method [20, p. 461] with the Steihaug–Toint modification [26, 25]. The preconditioner used in CG inexactly projects a given $v \in \mathbb{R}^n$ onto the null space of A. It achieves this (inexact) projection by using GMRES to compute a $v_\parallel$ that satisfies

$$\left\| \begin{bmatrix} \mathsf{I} & \mathsf{A}^T \\ \mathsf{A} & 0 \end{bmatrix} \begin{bmatrix} v_\parallel \\ d_\parallel \end{bmatrix} - \begin{bmatrix} v \\ 0 \end{bmatrix} \right\| \leq \eta_\parallel \|c\|.$$

The tolerance for the projection is $\eta_\parallel = 0.001$, which is two orders smaller than the tolerance $\eta = 0.1$ used in the outer CG algorithm; the smaller projection tolerance is chosen to reduce issues related to loss of conjugacy between the subspace vectors generated by CG [12].

For each QO, the total number of augmented matrix-vector and Hessian-vector products required by the composite-step algorithm was recorded. This number was then used as the iteration upper bound for FLECS to reach the relative tolerance of $\eta = 10^{-10}$; thus, the FLECS solution will (almost) always use the same number of products. Note that the primal-dual, augmented matrix and Hessian products are considered equal in terms of computational cost, because in the context of reduced-space PDE-constrained problems, each of these products requires two linearized PDE solutions.

As in the convex case, 10^5 random QOs were generated and FLECS solutions were obtained using both $\mu = 1/\|c\|$ and $\mu = 100/\|c\|$. In all cases the trust radius was $\Delta = 1$. The nonconvex results are shown in Figures 1(c) and 1(d). The normalized feasibility and objective are defined by (5.1) and (5.2), with the composite-step solution as the reference. The percentage of solutions in each quadrant of the two-dimensional histogram is again listed in Table 2.

In some respects, the nonconvex results are similar to the convex results. For instance, we observe the expected tradeoff between OBJ and FEAS as μ increases.

A notable difference from the convex results is the increased range of objective values obtained, reflected in the spread of the OBJ histograms. Another difference is that for $\mu = 1/\|c\|$ almost all FLECS solutions (98%) improve over the initial feasibility while simultaneously obtaining a lower objective than the composite-step approach. Finally, for $\mu = 100/\|c\|$, 90% of the FLECS solutions are superior to the composite-step solutions.

5.2. PDE-constrained optimization example.

5.2.1. Optimization algorithms. To assess its performance on nonlinear PDE-constrained optimization problems, FLECS was incorporated into a trust-region sequential quadratic optimization (SQO) algorithm with a filter-based globalization [11]; see Algorithm 4. This algorithm is simple, and we acknowledge that further analysis and enhancements would be necessary to achieve a production-level optimization library.

Several aspects of Algorithm 4 need clarification. In the algorithm and in the following, the subscript k on functions and their derivatives indicates evaluation at x_k.

- The penalty parameter is updated in line 6 according to the rule

$$\mu \leftarrow \max(\mu, \mu_0 \|c_0\|/\|c_k\|).$$

  This ensures that the penalty will eventually become sufficiently large to meet the requirements of the theory.
- In line 7, the tolerance η, used for the stopping criteria in FLECS, is computed based on a standard inexact-Newton update [7], with a lower bound of 0.001.
- If the trail step is dominated by the filter on the first filter iteration, a second-order correction is computed. This correction is based on inexactly solving the augmented system

$$\begin{bmatrix} \mathsf{I} & \mathsf{A}^T \\ \mathsf{A} & 0 \end{bmatrix} \begin{bmatrix} p_c \\ d_c \end{bmatrix} = - \begin{bmatrix} 0 \\ c(x_k + p_k) \end{bmatrix},$$

  where the correction is of the form $p_c = \mathsf{Z}_j^p y_c$, where Z_j^p is the subspace generated during the first call to FLECS in iteration k. To find y_c, we perform an oblique projection of the residual onto $[(\mathsf{Z}_j^p)^T (\mathsf{V}_j^d)^T]$. This leads to the subspace problem listed in line 16.
- The FLECS subspace is also recycled each time a new trail step is computed within iteration k; see line 23. This amounts to resolving the subproblem (3.7) using an updated trust radius, and no further iterations are taken within FLECS. Thus, each additional call to FLECS within the filter loop is significantly less expensive, assuming the matrix-vector and preconditioning operations are the most time-intensive tasks.

The performance of the FLECS-based algorithm is measured against the composite-step algorithm described by Algorithm 5 in Appendix B. The quasi-normal and quasi-tangential steps are computed in the same way as they were for the nonconvex QO tests, with the following differences and enhancements.

- As for the FLECS-based algorithm, the forcing parameter η is computed dynamically.
- When appropriate, a second-order correction is found by solving the appropriate augmented system.

- If a step is dominated by the filter, and the second-order correction fails, then the quasi-normal step is shrunk, if necessary (it is not recomputed), and a new quasi-tangential step is computed using the updated trust radius within projected CG. CG typically uses short-term recursions, so recycling the Krylov subspace is not normally possible here; however, some authors have investigated CG variants that save the subspace vectors [2, 21, 14]. Such an approach might offer a performance improvement for the composite-step algorithm, but this was not investigated here.

The nonlinear optimization algorithms were implemented in the open-source Kona library [17].

5.2.2. Multidisciplinary design problem. The nonlinear test case is an inverse design of an elastic nozzle subject to a quasi-one-dimensional inviscid flow. This problem was developed to investigate the issues that arise during the aero-structural optimization of an aircraft wing. The flow is modeled using the quasi-one-dimensional Euler equations, and the nozzle structure is modeled using Euler–Bernoulli beam theory. This yields a multidisciplinary design optimization (MDO) problem, since the flow pressure, which is a function of the nozzle area, deflects the nozzle structure, which in turn modifies the flow. Further details of this problem can be found in [8].

The control variables for the nozzle design problem are coordinates of B-spline control points that define the (static) nozzle area, viz.,

$$A(y) = 2.0\mathcal{N}_0^{(4)}(y) + 1.75\mathcal{N}_{n'+1}^{(4)} + \sum_{i=1}^{n'} x_i \mathcal{N}_i^{(4)}(y), \tag{5.3}$$

where y denotes the location along the nozzle, $\mathcal{N}_i^{(4)}$ is the ith fourth-order (cubic) B-spline basis function, and x_i is the ith component of the design variable. Uniform open knot vectors are used; consequently, the above parameterization fixes the nozzle inlet and outlet areas at $A(0) = 2.0$ and $A(1) = 1.75$, respectively.

The objective function is defined as

$$f(x) = \frac{1}{2}\sum_{j=0}^{N} w_j (p_j(x) - p_j^{\text{targ}})^2,$$

where w_j is a quadrature weight (see [16]), and $p_j(x)$ and p_j^{targ} are the pressure and target pressure at the jth spatial mesh node, respectively. We use uniformly spaced nodes with $N = 60$. Note that pressure is an implicit function of the design variable x via the flow's dependence on the deformed shape of the nozzle.

The target pressure, p_j^{targ}, is based on the solution of the fluid-structure problem for which the undeformed nozzle area, $A(y)$, is the unique cubic that satisfies $A(0.5) = 1.5$ and $A'(0.5) = 0.0$; the inlet and outlet areas fix the remaining two degrees of freedom for the cubic. The static and deformed nozzle areas are shown in Figure 2, together with the resulting target pressure.

We use a particular MDO problem formulation called individual discipline feasible (IDF) [13, 6]. The IDF formulation decouples the PDE solvers by introducing additional optimization variables. In the present example, pressure and nozzle-area coupling variables are used to perform this role; we will use $\bar{p}_j$ and $\bar{A}_j$ to denote the value of these variables at node j. The $\bar{p}_j$ pressures are supplied to the structural solver, and the $\bar{A}_j$ areas are given to the flow solver. The two PDEs can then be solved in a decoupled manner, which facilitates a modular approach to the problem.

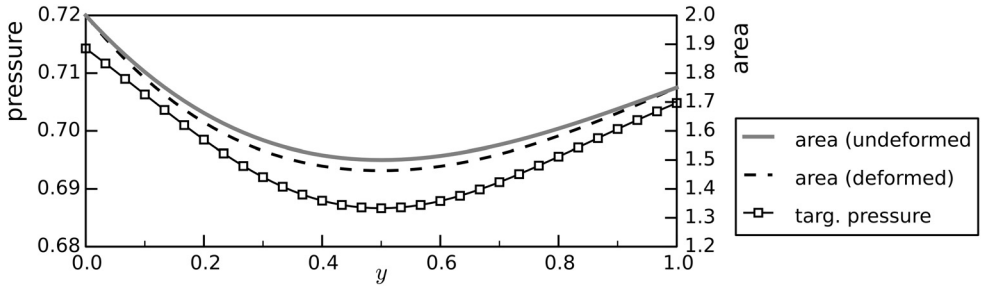


FIG. 2. *The static-nozzle area and deformed-nozzle area for the MDO design problem. The corresponding target pressure distribution is also shown, with every other node masked.*

TABLE 3
Parameter values adopted for the nozzle inverse-design problem.

parameter	value	parameter	value
τ^p	10^{-6}	η_0	0.5
τ^d	10^{-6}	max_iter	100
Δ_0	1	max_filter_iter	10
$\Delta_{\max}$	2	μ_0 (FLECS only)	0.01

The coupling variables must eventually match the true values they purport to represent. Consequently, the MDO problem includes the following state-based constraints:

$$c_{p,j} = \bar{p}_j - p_j(x) = 0, \qquad \text{and} \qquad c_{A,j} = \bar{A}_j - A_j(x) = 0$$

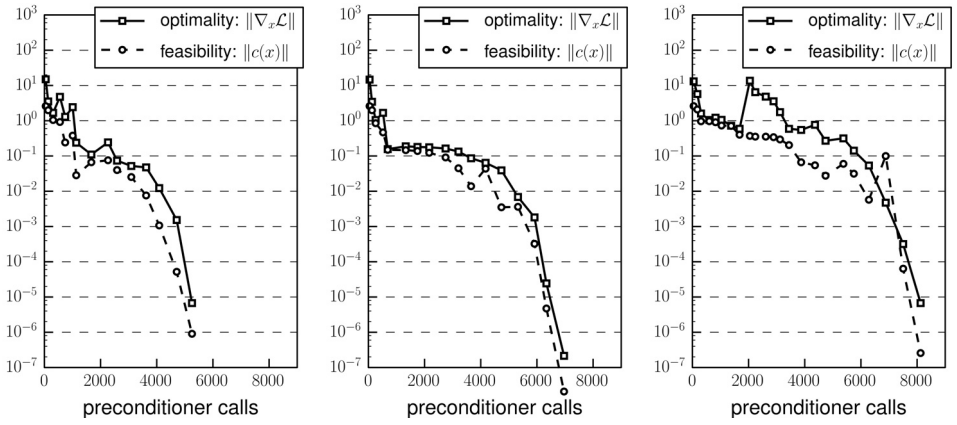
for all $j \in 0, 1, 2, \ldots, N$.

In summary, the optimization variables, x, include the B-spline coefficients and coupling variables, $\bar{p}_j$ and $\bar{A}_j$. We consider $n' = 20$ B-spline coefficients, and a computational grid with $N = 60$ intervals (61 nodes). Consequently, there are $n = 20 + 2(61) = 142$ optimization variables and $m = 2(61) = 122$ constraints.

The preconditioner used in FLECS is the IDF preconditioner presented in [8]. We omit the details of this preconditioner, but provide a brief qualitative description. For a given vector $z \in \mathbb{R}^n$, the subvector corresponding to the B-spline coefficients can be used to uniquely define the subvectors corresponding to $\bar{p}_j$ and $\bar{A}_j$; this is because the B-spline coefficients define the undeformed nozzle, which is sufficient to implicitly define the remaining variables via the PDEs. By linearizing this relationship, we can develop a projection-type preconditioner. An important aspect of this preconditioner is that it involves nested iterative methods.

The IDF preconditioner used by FLECS is also adopted as the preconditioner for augmented systems in the composite-step algorithm. In addition, the IDF preconditioner acts as the projection within the CG method for the tangential steps; trial and error was used to find a tolerance for the nested method that balanced loss of conjugacy with the expense of oversolving.

To characterize the FLECS-based algorithm on the MDO problem, 200 problems were solved, each with randomly generated initial guesses. These initial guesses were obtained by computing B-spline coefficients that yield a linear variation in nozzle area between the inlet and outlet, and then by perturbing these variables by uniform (pseudo) random variables drawn from $\mathcal{U}(0.75, 1.25)$. The initial coupling area and pressures were set based on this randomly perturbed static (i.e., undeformed) nozzle.



(a) sample 43 (b) sample 179 (c) sample 190

FIG. 3. *FLECS-based algorithm convergence histories for a random set of samples from the MDO inverse-design results.*

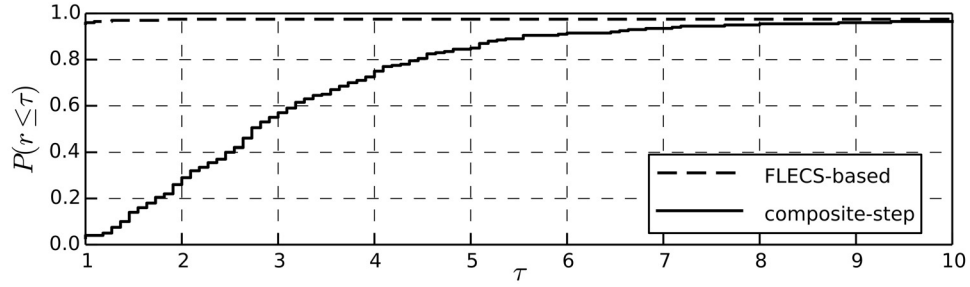


FIG. 4. *Performance profile for the FLECS-based and composite-step algorithms applied to the MDO inverse-design problem using random initial guesses.*

The parameter values adopted for the MDO problem are listed in Table 3. FLECS and the iterative solvers used in the composite-step algorithm were limited to 15 iterations. Samples that failed to converge in 100 nonlinear iterations were considered unsuccessful.

Figure 3 plots the convergence histories of the FLECS-based algorithm from three randomly selected samples. These results provide evidence that the FLECS trial steps can yield superlinear asymptotic convergence.

Figure 4 shows the performance profiles [10] for the FLECS-based and composite-step algorithms. For the MDO problem, the probability is 0.95 that the FLECS-based method will be the fastest choice. Moreover, in three out of four cases, the FLECS-based algorithm is faster by a factor of 2 or more. The composite-step algorithm is the fastest algorithm for 3% of the samples. The probability that the FLECS-based algorithm will fail is 0.025, while the probability that the composite-step algorithm will fail is 0.035.

6. Conclusions. We have presented the FLECS algorithm, an iterative solver for nonconvex quadratic optimization subproblems with equality constraints. The FLECS primal and dual trial steps are constructed using the same (flexible) Arnoldi procedure as used in FGMRES. Unlike FGMRES, which minimizes some combination

of the norms of the first-order optimality conditions, FLECS minimizes a quadratic penalty function over a trust radius. This helps to discourage trial steps from converging to stationary points that are not local minima.

In addition to handling nonconvex objectives, the FLECS algorithm has the following properties.

- FLECS requires only KKT-matrix-vector products and preconditioning operations, so it can be implemented matrix-free.
- FLECS exhibits excellent asymptotic convergence near the solution when used as the linear-algebra kernel in an inexact-Newton algorithm.
- FLECS can accommodate nonstationary preconditioners; this permits, for example, dynamic modification of the preconditioner, and the use of nested (inexact) iterative methods to precondition the KKT system.

The price paid for the flexibility in preconditioning is the need to store additional subspace vectors.

There are several questions to address in future work. For example, what is the best way to incorporate FLECS into the outer nonlinear algorithm? Related to this question, is there a better, and less ad hoc, method of computing the penalty parameter μ? Finally, how should FLECS be adapted for problems with inequality constraints?

Appendix A. Auxiliary proofs.

LEMMA 4. *If* W *is positive definite in the null space of* A*, then there exists* $\mu^\star$ *such that* $\forall \mu > \mu^\star$, $\mathsf{W} + \mu \mathsf{A}^T\mathsf{A}$ *is positive definite*

Proof. If W is positive definite then $\mathsf{W} + \mu\mathsf{A}^T\mathsf{A}$ will be positive definite for any $\mu > \mu^\star = 0$. Moreover, if $x \neq 0$ and $\mathsf{A}x = 0$, then $x^T\mathsf{W}x > 0$ by assumption. Thus, we need only consider the case when W has at least one nonpositive eigenvalue and $\mathsf{A}x \neq 0$. Define

$$\mu(x) = \frac{-x^T\mathsf{W}x}{x^T\mathsf{A}^T\mathsf{A}x}, \qquad \forall x \in \{z \in \mathbb{R}^n | \mathsf{A}z \neq 0\}.$$

We will show that $\mu(x)$ is bounded above, and then use this bound to define $\mu^\star$. Note that it is sufficient to consider vectors x of unit length, since the length of x can be factored out of $\mu(x)$.

The numerator of $\mu(x)$ is bounded by the negative of the minimum eigenvalue, since the field of values of the Rayleigh quotient of W is equal to the convex hull of its spectrum; that is,

$$-x^T\mathsf{W}x \leq -\lambda_{\min} \qquad \text{when} \qquad \|x\| = 1.$$

Thus, to show that $\mu(x)$ is unbounded above, we must show that the denominator can be made arbitrarily small while keeping $-x^T\mathsf{W}x > 0$. In particular, for any $\epsilon > 0$ we would need x such that

$$\|\mathsf{A}x\| \leq \epsilon \qquad \text{and} \qquad x^T\mathsf{W}x < 0.$$

However, this requirement is contradicted by the assumption that W is positive definite in the null space of A and by the continuity of the quadratic form $x^T\mathsf{W}x$. Therefore, $\mu(x)$ is bounded above and we take

$$\mu^\star = \sup \mu(x).$$

To summarize, when W has at least one nonpositive eigenvalue and $\mathsf{A}x \neq 0$, we have, for $\mu > \mu^\star$,

$$\begin{aligned} x^T\left(\mathsf{W} + \mu \mathsf{A}^T\mathsf{A}\right) x &\geq x^T\mathsf{W}x + \mu^\star x^T\mathsf{A}^T\mathsf{A}x \\ &= x^T\mathsf{A}^T\mathsf{A}x\left(-\mu(x) + \mu^\star\right) > 0. \end{aligned}$$

Thus, $\mathsf{W} + \mu\mathsf{A}^T\mathsf{A}$ is positive definite for this choice of $\mu^\star$. ∎

Appendix B. Nonlinear optimization algorithms.

Algorithm 4: FLECS-based SQO algorithm with filter globalization.

Data: x_0, μ_0, η_0, Δ_0, $\Delta_{\max}$, τ^p, τ^d
Result: optimal solution x

$\mu = \mu_0$, $\Delta = \Delta_0$
for $k = 0, 1, 2, \ldots,$ *max_iter* **do**
  if $\|g_k\| \leq \tau^p\|g_0\|$ *and* $\|c_k\| \leq \tau^d\|c_0\|$ **then** (check convergence)
    set $x = x_k$ and return
  end
  $\mu \leftarrow \max(\mu, \mu_0\|c(x_0)\|/\|c(x_k)\|)$
  $\eta \leftarrow \max\left[0.001, \eta_0 \min\left(1.0, \sqrt{\|g_k\|^2 + \|c_k\|^2}/\sqrt{g_0\|^2 + \|c_0\|^2}\right)\right]$
  solve for primal-dual trial step s_k using FLECS, Algorithm 2, with tolerance η, penalty μ, and trust radius Δ
  for $i = 0, 1, 2, \ldots,$ *max_filter_iter* **do**
    if $[f(x_k + p_k), c(x_k + p_k)]$ *is not dominated by filter* **then**
      if $i = 0$ *and* $\|p_k\| = \Delta$ **then** $\Delta \leftarrow \min(2\Delta, \Delta_{\max})$
      set filter_success = true
      exit loop
    end
    if $i = 0$ **then** (try a second-order correction)
      compute the correction $p_c = \mathsf{Z}^p y_c$, where y_c is the solution of

$$\left[\left(\mathsf{Z}_j^p\right)^T\mathsf{Z}_j^p + \left(\mathsf{Z}_j^d\right)^T\mathsf{Z}_{j+1}^d\bar{\mathsf{H}}_j + \bar{\mathsf{H}}_j^T\left(\mathsf{Z}_{j+1}^d\right)^T\mathsf{Z}_j^d\right] y_c = \left(\mathsf{Z}_j^d\right)^T c(x_k + p_k)$$

      if $[f(x_k + p_k + p_c), c(x_k + p_k + p_c)]$ *is not dominated by filter* **then**
        set filter_success = true and $p_k \leftarrow p_k + p_c$
        exit loop
      end
    end
    $\Delta \leftarrow \frac{1}{4}\Delta$
    re-solve for primal-dual trial step s_k using FLECS, Algorithm 2, with penalty μ and trust radius Δ
  end
  if *filter_success* **then**
    $x_{k+1} = x_k + p_k$, $\lambda_{k+1} = \lambda_k + d_k$
  else
    $x_{k+1} = x_k$, $\lambda_{k+1} = \lambda_k$
  end
end

Algorithm 5: Composite-step SQO algorithm with filter globalization.

Data: x_0, η_0, Δ_0, $\Delta_{\max}$, τ^p, τ^d
Result: optimal solution x

$\Delta = \Delta_0$
for $k = 0, 1, 2, \ldots,$ *max_iter* **do**
  if $\|g_k\| \le \tau^p \|g_0\|$ *and* $\|c_k\| \le \tau^d \|c_0\|$ **then** (check convergence)
    set $x = x_k$ and return
  end
  $\eta \leftarrow \max\left[0.001, \eta_0 \min\left(1.0, \sqrt{\|g_k\|^2 + \|c_k\|^2}/\sqrt{\|g_0\|^2 + \|c_0\|^2}\right)\right]$
  compute quasi-normal step, $p_\perp$, by inexactly solving augmented system with right-hand side $[0, -c_k^T]^T$ to a tolerance η
  if $\|p_\perp\| \ge 0.8\Delta$ **then** $p_\perp \leftarrow \frac{0.8\Delta}{\|p_\perp\|} p_\perp$
  compute quasi-tangential step, $p_\|$, using projection CG with a tolerance η and trust radius $\Delta_\| = \sqrt{\Delta^2 - p_\perp^T p_\perp}$
  set $p_k = p_\perp + p_\|$
  for $i = 0, 1, 2, \ldots,$ *max_filter_iter* **do**
    if $[f(x_k + p_k), c(x_k + p_k)]$ *is not dominated by filter* **then**
      if $i = 0$ *and* $\|p_k\| = \Delta$ **then** $\Delta \leftarrow \min(2\Delta, \Delta_{\max})$
      set filter_success = true
      exit loop
    end
    if $i = 0$ **then** (try a second-order correction)
      compute the correction p_c, by inexactly solving augmented system with right-hand side $[0, -c(x_k + p_k)^T]^T$ to a tolerance η
      if $[f(x_k + p_k + p_c), c(x_k + p_k + p_c)]$ *is not dominated by filter* **then**
        set filter_success = true and $p_k \leftarrow p_k + p_c$
        exit loop
      end
    end
    $\Delta \leftarrow \frac{1}{4}\Delta$
    if $\|p_\perp\| \ge 0.8\Delta$ **then** $p_\perp \leftarrow \frac{0.8\Delta}{\|p_\perp\|} p_\perp$
    compute quasi-tangential step, $p_\|$, using projection CG with a tolerance η and trust radius $\Delta_\| = \sqrt{\Delta^2 - p_\perp^T p_\perp}$
    set $p_k = p_\perp + p_\|$
  end
  if *filter_success* **then**
    $x_{k+1} = x_k + p_k$
    compute multiplier step, d_k, by solving augmented system with right-hand side $[-g_k^T, 0]^T$ to a tolerance η
    $\lambda_{k+1} = \lambda_k + d_k$
  else
    $x_{k+1} = x_k$, $\lambda_{k+1} = \lambda_k$
  end
end

Acknowledgments. The authors thank the anonymous reviewers for their feedback, which helped improve the paper. All figures were produced using Matplotlib [18].

REFERENCES

[1] V. Akçelik, G. Biros, O. Ghattas, J. Hill, D. Keyes, and B. van Bloemen Waanders, *Parallel algorithms for PDE-constrained optimization*, in Parallel Processing for Scientific Computing, M. A. Heroux, P. Raghavan, and H. D. Simon, eds., SIAM, Philadelphia, 2006, pp. 291–322.

[2] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, Cambridge, UK, 1996.

[3] A. Borzì and V. Schulz, *Computational Optimization of Systems Governed by Partial Differential Equations*, SIAM, Philadelphia, 2011.

[4] R. H. Byrd, F. E. Curtis, and J. Nocedal, *An inexact Newton method for nonconvex equality constrained optimization*, Mathematical Programming, 122 (2010), pp. 273–299.

[5] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust Region Methods*, SIAM, Philadelphia, 2000.

[6] E. J. Cramer, J. E. Dennis, Jr., P. D. Frank, R. M. Lewis, and G. R. Shubin, *Problem formulation for multidisciplinary optimization*, SIAM Journal on Optimization, 4 (1994), pp. 754–776.

[7] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM Journal on Numerical Analysis, 19 (1982), pp. 400–408.

[8] A. Dener and J. E. Hicken, *Revisiting individual discipline feasible using matrix-free inexact-Newton-Krylov*, in Proceedings of the 10th AIAA Multidisciplinary Design Optimization Conference, American Institute of Aeronautics and Astronautics, 2014, AIAA 2014–0110.

[9] A. Dener, G. K. W. Kenway, Z. Lyu, J. E. Hicken, and J. R. R. A. Martins, *Comparison of inexact- and quasi-Newton algorithms for aerodynamic shape optimization*, in Proceedings of the AIAA SciTech Conference, 2015, AIAA 2015-1945.

[10] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Mathematical Programming, 91 (2002), pp. 201–213.

[11] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Mathematical Programming, 91 (2002), pp. 239–269.

[12] G. H. Golub and Q. Ye, *Inexact preconditioned conjugate gradient method with inner-outer iteration*, SIAM Journal on Scientific Computing, 21 (1999), pp. 1305–1320.

[13] R. T. Haftka, J. Sobieszczanski-Sobieski, and S. L. Padula, *On options for interdisciplinary analysis and design optimization*, Structural and Multidisciplinary Optimization, 4 (1992), pp. 65–74.

[14] M. Heinkenschloss and D. Ridzal, *A matrix-free trust-region SQP method for equality constrained optimization*, SIAM Journal on Optimization, 24 (2014), pp. 1507–1541.

[15] J. E. Hicken, *FLECS: MATLAB Implementation of the FLexible Equality-Constrained Subproblem Solver*, https://bitbucket.org/odl/flecs (8 October 2014).

[16] J. E. Hicken, *Inexact Hessian-vector products in reduced-space differential-equation constrained optimization*, Optimization and Engineering, 15 (2014), pp. 575–608.

[17] J. E. Hicken, *Kona: A Software Library for De-Constrained Optimization*, https://bitbucket.org/odl/kona (2 October 2014).

[18] J. D. Hunter, *Matplotlib: A 2D graphics environment*, Computing in Science & Engineering, 9 (2007), pp. 90–95.

[19] J. J. Moré and D. C. Sorensen, *Computing a Trust Region Step*, SIAM Journal on Scientific and Statistical Computing, 4 (1983), pp. 553–572.

[20] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., Springer-Verlag, Berlin, Germany, 2006.

[21] Y. Notay, *Flexible Conjugate Gradients*, SIAM Journal on Scientific Computing, 22 (2000), pp. 1444–1460.

[22] E. O. Omojokun, *Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints*, Ph.D. thesis, University of Colorado at Boulder, Boulder, CO, 1989.

[23] Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM Journal on Scientific and Statistical Computing, 14 (1993), pp. 461–469.

[24] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.

[25] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM Journal on Numerical Analysis, 20 (1983), pp. 626–637.

[26] P. L. Toint, *Towards an efficient sparsity exploiting Newton method for minimization*, in Sparse Matrices and their Uses, I. S. Duff, ed., Academic Press, New York, 1981, pp. 57–88.