

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

JASON E. HICKEN\*<sup>‡</sup> AND ALP DENER<sup>†‡</sup>

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

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

**1. Introduction.** Consider nonlinear equality-constrained optimization problems of the form

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

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

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

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 [18] and [4] 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 absorb the state variables into  $x$  and include the PDE as an additional constraint. This simplifies the evaluation of derivatives, but increases the size of the problem. A more

---

\*This work was supported by the National Science Foundation through grant number 1332819

\*Assistant Professor ([hickej2@rpi.edu](mailto:hickej2@rpi.edu))

<sup>†</sup>Graduate Student

<sup>‡</sup>Mechanical, Aerospace, and Nuclear Department, Rensselaer Polytechnic Institute, 110 8<sup>th</sup> Street, Troy, New York, 12180

detailed discussion of the relative merits of the reduced- and full-space formulations can be found in [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 each 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,  $A$ , to determine a basis for its null-space. Factoring the Jacobian is not an option for matrix-free algorithms, so researchers have investigated alternative ways to globalize these algorithms.

Recently, Heinkenschloss and Ridzal [12] 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 solves that we would prefer to avoid in reduced-space PDE-constrained optimization.

In [3], 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, so the computational time preceding this restart is essentially wasted unless the subspace is recycled. 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, [7]. In general, such preconditioners are nonstationary and require so-called flexible iterative methods. To the best of our knowledge, flexible *and* globalized iterative 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 steps produced by the solver should yield 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 [21] 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

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

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

The first-order optimality conditions for (1.1) are

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

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

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

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 quadratic optimization problem (1.2).

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

$$Ks = b \quad (1.5)$$

where  $K \in \mathbb{R}^{(n+m) \times (n+m)}$  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 non-singular and  $A$  has full rank; consequently,  $K$  is also non-singular.

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,

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

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

$$V^p \equiv V_{1:n,1:j}, \quad \text{and} \quad V^d \equiv V_{n+1:n+m,1:j}.$$

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  $j^{\text{th}}$  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

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

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

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

Let  $\mathcal{M}_j : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^{n+m}$  denote the preconditioner at iteration  $j$ . The preconditioner is represented as a general operator, rather than a matrix, because we would like to permit the use of nonstationary iterative methods for the preconditioning operation. For example, the preconditioner might approximate the action of the inverse of  $\mathbf{K}$ , that is, for an arbitrary vector  $u \in \mathbb{R}^{n+m}$ , the vector

$$\mathcal{M}_j(u) - \mathbf{K}^{-1}u$$

should be small in some sense.

**2. Flexible Arnoldi and FGMRES.** As mentioned in the introduction, the proposed algorithm builds off of FGMRES [21]. 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<sup>1</sup>  $\text{span}\{z_1, z_2, \dots, z_j\}$  that minimizes the 2-norm of the residual  $r_j$ . In other words, if  $Z_j = [z_1 z_2 \cdots z_j]$ , then

$$s_j = Z_j y_j, \quad \text{where} \quad y_j = \underset{y \in \mathbb{R}^j}{\text{argmin}} \|b - \mathbf{K}s_j\|.$$

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  $\mathbf{K}z_j$  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  $z_i$  [21]:

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

<sup>1</sup>More generally, FGMRES seeks a solution in the affine subspace  $s_0 + \text{span}\{z_1, z_2, \dots, z_j\}$ .

where

$$h_{i,j} \equiv v_i^T(Kz_j), \quad i = 1, \dots, j+1.$$

Introducing the matrix  $V_{j+1} = [v_1 v_2 \cdots v_{j+1}]$ , the above relationship can be written concisely as

$$KZ_j = V_{j+1}\bar{H}_j, \quad (2.1)$$

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

$$WZ_j^p + A^T Z_j^d = V_{j+1}^p \bar{H}_j, \quad (2.2)$$

$$AZ_j^p = V_{j+1}^d \bar{H}_j. \quad (2.3)$$

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

$$\begin{aligned} r_j &= b - Ks_j = b - KZ_j y_j \\ &= r_0 - V_{j+1} \bar{H}_j y_j \\ &= V_{j+1} (\beta e_1 - \bar{H}_j y_j), \end{aligned}$$

where  $e_1$  is the first column of the  $(j+1) \times (j+1)$  identity. 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,

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

and

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

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 some applications.

For completeness, FGMRES is listed in Algorithm 1. The parameter  $\eta$  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 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  $y_j$ .

**Algorithm 1:** FGMRES.

---

**Data:**  $b, \eta \in (0, 1)$   
**Result:**  $s$ , the inexact solution satisfying  $\|Ks + b\| \leq \eta\|b\|$

- 1 compute  $r_0 = b$ ,  $\beta = \|r_0\|$ , and  $v_1 = r_0/\beta$
- 2 **for**  $j = 1, 2, 3, \dots$  **do**
- 3      $z_j \leftarrow \mathcal{M}_j(v_j)$
- 4      $v_{j+1} \leftarrow Kz_j$
- 5     **for**  $i = 1, \dots, j$  **do**     (Modified Gram-Schmidt)
- 6          $h_{i,j} = v_{j+1}^T v_j$
- 7          $v_{j+1} \leftarrow v_{j+1} - h_{i,j}v_j$
- 8     **end**
- 9     compute  $h_{j+1,j} = \|v_{j+1}\|$ , and  $v_{j+1} \leftarrow v_{j+1}/h_{j+1,j}$
- 10    **if**  $\|r_j\| \leq \eta\beta$  **then**     (check convergence)
- 11        compute  $y_j = \operatorname{argmin}_y \|\beta e_1 + Hy\|$
- 12         $s \leftarrow Z_j y_j$
- 13        **return**
- 14    **end**
- 15 **end**

---

**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

$$\min_{p \in \operatorname{span}\{Z_j^p\}} Q(p, \mu) \equiv g^T p + \frac{1}{2} p^T W p + \frac{\mu}{2} (Ap + c)^T (Ap + c), \quad (3.1)$$

where  $\mu \geq 0$  is the penalty parameter. A key feature of this penalty problem 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  $\operatorname{span}(Z_j^p)$ . An expression for  $W_Z$  follows immediately from the Arnoldi relation (2.1) (see also (2.2) and (2.3)):

$$W_Z \equiv (Z_j^p)^T W_Z^p = Z_j^T V_{j+1} \bar{H}_j - (Z_j^d)^T V_{j+1}^d \bar{H}_j - \bar{H}_j^T (V_{j+1}^d)^T Z_j^d. \quad (3.2)$$

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  $\mu A^T A$ . This term can also be simplified in the subspace of the solution using (2.3):

$$(A^T A)_Z \equiv (Z_j^p)^T A^T A Z_j^p = \bar{H}_j^T (V_{j+1}^d)^T V_{j+1}^d \bar{H}_j. \quad (3.3)$$

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  $(A^T c)_Z$  to denote, respectively, the product of  $g$  and

$\mathbf{A}^T c$  with  $(\mathbf{Z}_j^p)^T$ , then

$$gz \equiv (\mathbf{Z}_j^p)^T g = -(\mathbf{Z}_j^p)^T (\beta v_1^p), \quad (3.4)$$

$$\text{and } (\mathbf{A}^T c)_z \equiv (\mathbf{Z}_j^p)^T \mathbf{A}^T c = -\bar{\mathbf{H}}_j^T (\mathbf{V}_{j+1}^d)^T (\beta v_1^d). \quad (3.5)$$

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

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

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

If  $\mathbf{W}$  is nonconvex in the null space of  $\mathbf{A}$ , then the solution to the target QO (1.2) is unbounded below. This nonconvexity also manifests itself in the solution of the subspace penalty problem (3.6). While the probability is low that one of the  $z_i$  is *exactly* in the null space of  $\mathbf{A}$ , 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.

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

$$\begin{aligned} \min_{y \in \mathbb{R}^j} \quad & [gz + \mu (\mathbf{A}^T c)_z]^T y + \frac{1}{2} y^T [\mathbf{W}_z + \mu (\mathbf{A}^T \mathbf{A})_z] y \\ \text{s.t.} \quad & \|\mathbf{Z}_j^p y\| \leq \Delta. \end{aligned} \quad (3.7)$$

In practice, we solve this (small) subproblem by applying the Moré and Sorensen algorithm [17].

**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 very challenging. We considered several approaches.

- Assume that  $\lim_{j \rightarrow \infty} \mu = \infty$  and that the primal problem converges. Then the asymptotic result ([18], pg. 503)  $\lim_{j \rightarrow \infty} \mu (\mathbf{A} p_j + c) = d$  suggests the inexact dual step

$$d_j = \mu (\mathbf{A} p_j + c);$$

however, not surprisingly, this dual step does not perform well for inexact solutions and is sensitive to the growth schedule adopted for  $\mu$ . Moreover, it is not appropriate when the trust-region constraint is active.

- Ideally, we would like  $d_j$  such that

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

Unfortunately, if we choose  $d_j$  of the form  $\mathbf{Z}_j^d y_j^d$ , the residual  $g + \mathbf{W} p_j + \mathbf{A}^T d$  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  $d_j = V_{j+1}^d y_j$ . Setting the resulting subspace residual to zero leads to the subproblem

$$Z_j^T (g + Wp_j + A^T d_j) = g_Z + W_Z y_j + \left[ \bar{H}_j^T (V_{j+1}^d)^T V_{j+1}^d \right] y_j^d = 0,$$

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 pseudo-inverses 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,  $\lambda$ , 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:

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

Setting the primal-dual solution to  $s_j = Z_j y_j$ , projecting the residual onto  $V_j$ , and setting the result to zero, yields the following subproblem:

$$\left[ \lambda (V_j^p)^T Z_j^p + H_j \right] y_j = \beta e_1,$$

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

Ultimately, the FGMRES solution was adopted for the dual step. That is, we set  $d_j = Z_j^d y_j^d$  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, especially when  $\mu$  is large. We have found that this risk can be ameliorated through the outer globalization method and the update used for  $\mu$ .

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.

$$\|(W + \mu A^T A)p + g + \mu A^T c\| \leq \eta;$$

However, computing this norm requires explicit evaluation of the penalty gradient, which, in turn, requires at least two linear PDE solves 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)

---

```

1 compute  $r_0 = b, \beta = \|r_0\|$ , and  $v_1 = r_0/\beta$ 
2 compute initial norms,  $\omega_0 = \|r_0^p\|$  and  $\gamma_0 = \|r_0^d\|$ 
3 for  $j = 1, 2, 3, \dots$  do
4    $z_j \leftarrow \mathcal{M}_j(v_j)$ 
5    $v_{j+1} \leftarrow \mathbf{K}z_j$ 
6   for  $i = 1, \dots, j$  do   (Modified Gram-Schmidt)
7      $h_{i,j} = v_{j+1}^T v_j$ 
8      $v_{j+1} \leftarrow v_{j+1} - h_{i,j}v_j$ 
9   end
10  compute  $h_{j+1,j} = \|v_{j+1}\|$ , and  $v_{j+1} \leftarrow v_{j+1}/h_{j+1,j}$ 
11  solve for  $y_j^p$  and  $y_j^d$  using Algorithm 3
12  compute FGMRES primal- and dual-residual norms:

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

13  if  $\omega_j \leq \eta\omega_0$  and  $\gamma_j \leq \eta\gamma_0$  then   (check subspace quality)
14    compute primal step:  $p_j \leftarrow Z_j^p y_j^p$ 
15    compute dual step:  $d_j \leftarrow Z_j^d y_j^d$ 
16    return
17  end

```

---

**Algorithm 3: FLECS subspace solution.****Data:**  $\Delta > 0, \mu \geq 0, \beta$ , and the matrices

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

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

- 1 compute  $\mathbf{W}_Z, (\mathbf{A}^T \mathbf{A})_Z, g_Z$  and  $(\mathbf{A}^T c)_Z$  using (3.2), (3.3), (3.4), and (3.5), respectively.
- 2 solve the trust-region primal subproblem for  $y^p$ :

$$\min_{y \in \mathbb{R}^j} [g_Z + \mu (\mathbf{A}^T c)_Z]^T y + \frac{1}{2} y^T [\mathbf{W}_Z + \mu (\mathbf{A}^T \mathbf{A})_Z] y, \quad \text{s.t.} \quad \|Z^p y\| \leq \Delta.$$

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

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  $\mu$  large enough to ensure the FLECS' step is at least as feasible. This claim is supported by the theory presented in the next section.

**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 penalty function's Hessian.

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

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

$$\mu(x) = \frac{-x^T W x}{x^T A^T A x}, \quad \forall x \in \{z \in \mathbb{R}^n \mid Az \neq 0\}$$

We will show that  $\mu(x)$  is bounded above, and then use this bound to define  $\mu^*$ . 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,

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

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

$$\|Ax\| \leq \epsilon \quad \text{and} \quad x^T W x < 0;$$

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  $x^T W x$ . Therefore,  $\mu(x)$  is bounded above and we take

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

To summarize, when  $W$  has at least one nonpositive eigenvalue and  $Ax \neq 0$ , we have, for  $\mu > \mu^*$ ,

$$\begin{aligned} x^T (W + \mu A^T A) x &\geq x^T W x + \mu^* x^T A^T A x \\ &= \frac{1}{x^T A^T A x} (-\mu(x) + \mu^*) > 0. \end{aligned}$$

Thus  $W + \mu A^T A$  is positive definite for this choice of  $\mu^*$ .  $\square$

With the lemma established, we turn to our first result, which concerns the convergence of FLECS in the convex case with no trust-radius constraint, i.e.  $\Delta = \infty$  in (3.7). In the following,  $p_j = Z_j^p y_j$  and  $p_j^F = Z_j^p y_j^F$  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*

$$\begin{aligned} \|Ap_j + c\| &\leq \tau, \\ \text{and} \quad \|Wp_j + g + A^T d_j\| &\leq \tau. \end{aligned}$$

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

$$\|p_j - p_j^F\| \leq \|p_j - p(\mu)\| + \|p(\mu) - p\| + \|p - p_j^F\|, \quad (4.1)$$

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  $K(p - p_j^F) = r_j^p$  and our earlier assumption that the primal-dual matrix is invertible:

$$\|p - p_j^F\| = \|(K^{-1}r_j^p)^p\| \leq \|K^{-1}\| \|r_j^p\|.$$

The FGMRES residual can be made arbitrarily small, by assumption; therefore,  $\forall \epsilon > 0$ , there exists an iteration  $k$  such that  $\|p - p_j^F\| \leq \epsilon$  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 quadratic optimization problem. It is known (see, for example, [18], pg 502) that this difference tends to zero as  $\mu \rightarrow \infty$ . Consequently,  $\forall \epsilon > 0$ , there exists  $\mu > 0$  such that

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

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  $p_j = Z_j y$  that ensures  $\nabla_p Q$  is orthogonal to  $\operatorname{span}\{Z_j\}$  (i.e. the Galerkin solution):

$$Z_j^T \nabla_p Q = Z_j^T [(W + \mu A^T A) Z_j y + g + \mu A^T c] = 0.$$

If we ensure  $\mu > \mu^*$ , we have that  $W + \mu A^T A$  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  $W + \mu A^T A$  ([22], page 113):

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

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

eventually since the flexible Arnoldi method does not breakdown by assumption. So, for any  $\epsilon > 0$ , there exists an iteration  $k$  such that

$$\|p_j - p(\mu)\| \leq \epsilon, \quad \forall j > k,$$

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

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

Now, the constraint residual based on the FLECS solution satisfies

$$\begin{aligned} \|Ap_j + c\| &\leq \|Ap_j^F + c\| + \|A(p_j - p_j^F)\| \\ &\leq \|r_j^d\| + \|A\| \|p_j - p_j^F\|. \end{aligned}$$

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  $\mu$  and  $j$  sufficiently large. The result follows, because  $\|A\|$  is bounded.

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

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

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  $\|W\|$  is bounded, the norm of the gradient can be made less than  $\tau > 0$ .  $\square$

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*

$$\|Ap_j + c\| \leq \tau.$$

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

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

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  $\mu$ , then the result will follow by making  $\mu$  sufficiently large.

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

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

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

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

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  $\mu$  and  $j$  sufficiently large to satisfy

$$\|A p_j + c\|^2 \leq \tau^2,$$

for all  $\tau > 0$ , as desired.  $\square$

**4.1. Discussion.** Theorem 2 suggests that 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 non-convex situations, the theory suggests that we increase  $\mu$  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  $\mu$  outside of FLECS is more robust and effective. As long as  $\mu$  increases in the outer iterations, it will eventually be large enough to produce the desired asymptotic performance.

Actually, it is not clear that  $\mu$  needs to increase indefinitely, and we have often obtained excellent results with a fixed (relatively small) value of  $\mu$ . 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  $\mu c^T \nabla_{xx}^2 c$  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.

**5.1. Synthetic Quadratic Optimizations.** Our first numerical experiments exercise the FLECS algorithm on a set of synthetic quadratic optimization problems. Each problem was 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  $\kappa = 100$  was chosen as the condition number for  $W$ .
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 non-convex 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

$$W \equiv E\Lambda E^T$$

where  $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ .

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:

$$A \equiv RE_{:,1:m}^T.$$

6. The gradient was defined as  $g = W\hat{p}$ , where  $\hat{p}$  was constructed with random entries and normalized to unit length.
7. The vector  $c$  was defined as  $-Ap_{\perp}$ , 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 a Bitbucket repository [13].

**5.1.1. Convex 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 5.1(a) and 5.1(b) plot the convex-case results for the two values of  $\mu$ . The figures include a 2-dimensional histogram of normalized feasibility versus normalized objective, which are defined as

$$\text{FEAS}(p_j, p_{\text{ref}}) = \frac{\|Ap_j + c\| - \|Ap_{\text{ref}} + c\|}{\|c\| - \|Ap_{\text{ref}} + c\|}, \quad (5.1)$$

$$\text{and } \text{OBJ}(p_j, p_{\text{ref}}) = \frac{(g^T p_j + \frac{1}{2}p_j^T W p_j) - (g^T p_{\text{ref}} + \frac{1}{2}p_{\text{ref}}^T W p_{\text{ref}})}{|g^T p_{\text{ref}} + \frac{1}{2}p_{\text{ref}}^T W p_{\text{ref}}|}, \quad (5.2)$$

respectively, where  $p_j$  is the FLECS solution and  $p_{\text{ref}}$  is the reference solution (the FGMRES solution here). The shade of each small box in the 2-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 1-dimensional histograms for the normalized objective (upper histogram) and normalized feasibility (histogram to the right of the 2-dimensional histogram).

The dotted lines in the histograms separate FLECS results that improve upon the FGMRES results from those that do worse. For example, any FLECS result with a normalized objective less than zero has performed better than FGMRES with respect to this criterion. Similarly for the normalized feasibility. Ideally, we want results that improve in both criterion, i.e. the results in the lower-left quadrant of the

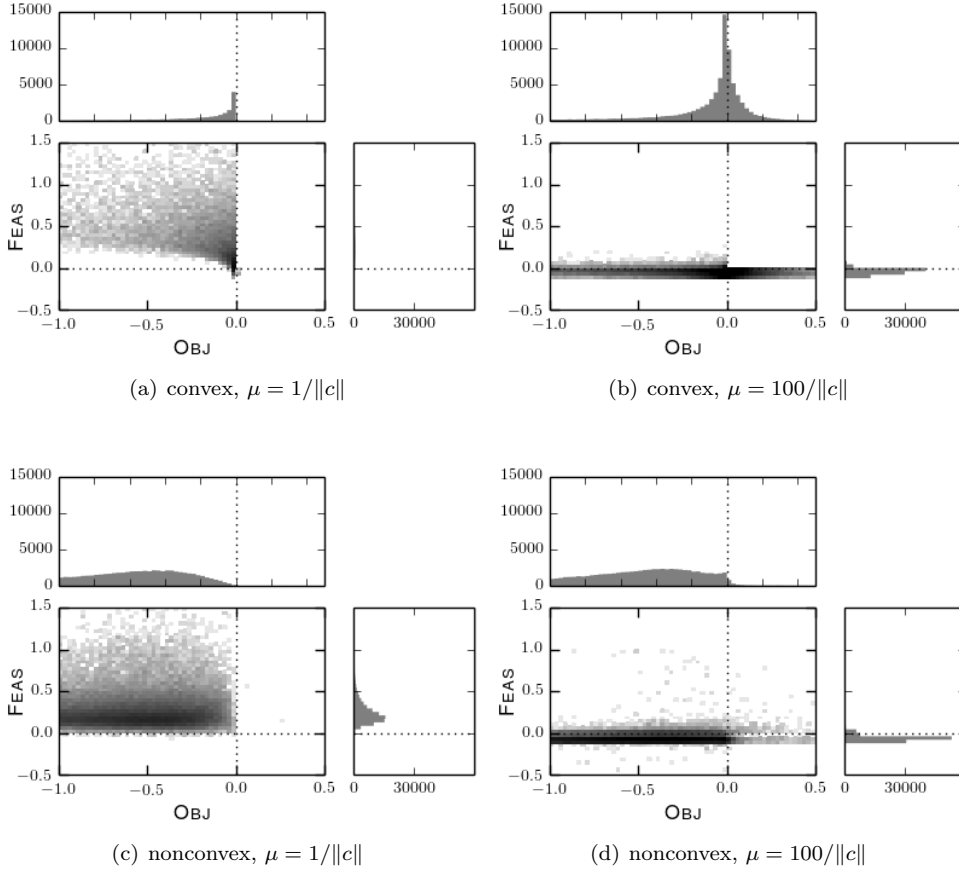


FIG. 5.1. Histograms of normalized objective and normalized feasibility for the synthetic QOs.

2-dimensional histogram. Table 5.1 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  $\mu$  is not sufficiently large. Indeed, for  $\mu = 1/\|c\|$ , 83.3% of FLECS solutions have  $\text{FEAS} > 1$ , 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 0.25% for both values of  $\mu$ .

For  $\mu = 100/\|c\|$ , FLECS produces a better solution than FGMRES more than 50% of the time. Moreover, for this value of  $\mu$ , 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  $\mu$  is sufficiently large. Evidence for this is presented for a PDE-constrained optimization example presented later.

**5.1.2. Nonconvex in the Null Space.** For nonconvex QOs, FLECS was compared against a composite-step trust-region algorithm [20], which we briefly describe (the algorithm is provided in the Matlab repository). The quasi-normal step,  $p_{\perp}$  is found by inexactly solving for the minimum-norm solution that satisfies the constraint.

TABLE 5.1  
*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\ $	0.22%	0.02%	99.12%	0.65%
convex, $\mu = 100/\ c\ $	0.12%	31.73%	16.55%	51.60%
nonconvex, $\mu = 1/\ c\ $	0.03%	0.0%	99.83%	0.14%
nonconvex, $\mu = 100/\ c\ $	0.55%	1.99%	7.05%	90.42%

In particular, the normal step satisfies

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

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 projected conjugate gradient ([18], pg 461) with the Steihaug-Toint modification [24, 23]. The preconditioner used in CG inexactly projects a given  $v \in \mathbb{R}^n$  onto the null space of  $\mathbf{A}$ . It achieves this (inexact) projection by using GMRES to compute a  $v_{\parallel}$  that satisfies

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

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 [10].

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 2 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 5.1(c) and 5.1(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 2-dimensional histogram is again listed in Table 5.1.

In some respects, the nonconvex results are similar to the convex results. For instance, the expected tradeoff between OBJ and FEAS as  $\mu$  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's solutions are superior to the composite-step solutions.

**5.2. Nonlinear Tests.** To assess its performance on nonlinear problems, FLECS was incorporated into a trust-region sequential-quadratic optimization (SQO) algorithm with a filter-based globalization [9]; see Algorithm 4. The FLECS-based algorithm is, in many respects, 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 the following, the subscript  $k$  on functions and their derivatives indicate 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  $\eta$ , used for the stopping criteria in FLECS, is computed based on a standard inexact-Newton update [6], 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

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

where the correction is of the form  $p_c = Z_j^p y_c$ , 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  $[(Z_j^p)^T (V_j^d)^T]$ . This leads to the subspace problem listed on 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, also in the Appendix. 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.

- Like the FLECS-based algorithm, the forcing parameter  $\eta$  is computed dynamically.
- The projection preconditioner for Steihaug-Toint CG varies depending on the case. For the constrained nozzle inverse design, we use the augmented-system with  $\eta_{\parallel} = 0.01$ . For the MDO problem, a description of the modified projection preconditioner is delayed until Section 5.4.
- 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, 19, 12]. 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 [15].

**5.3. Constrained Nozzle Inverse Design.** The first nonlinear test case is an inverse design of a quasi-one-dimensional nozzle flow with a single state-based constraint. The inviscid flow in the nozzle is modeled using the quasi-one-dimensional Euler equations. Further details regarding this flow and its discretization are provided in reference [14].

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

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

where  $y$  denotes the location along the nozzle,  $\mathcal{N}_i^{(4)}$  is the  $i^{\text{th}}$  fourth-order (cubic) B-spline basis function, and  $x_i$  is the  $i^{\text{th}}$  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. We consider  $n = 40$  design variables for the nozzle problem.

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 [14]), and  $p_j(x)$  and  $p_j^{\text{targ}}$  are the pressure and target pressure at the  $j^{\text{th}}$  spatial mesh node, respectively. For the nozzle inverse design, we use uniformly spaced nodes with  $N = 200$ . Note that the pressure is an implicit function of the variable  $x$ : the pressure is computed from the PDE state solution, which is determined by the nozzle area.

The target pressure is the pressure that results when  $A(y)$  is the unique cubic nozzle 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 target pressure is illustrated in Figure 5.2.

A single constraint is imposed on the pressure at the midpoint node ( $j = 100$ ), namely

$$c(x) = p_{100}(x) - p_{\text{mid}} = 0.$$

The pressure at the midpoint is required to equal the average of the (precomputed) inlet and outlet pressures:  $p_{\text{mid}} = \frac{1}{2}(p(0) + p(1))$ .

If the constraint were absent, the optimal nozzle area would produce  $p_j = p_j^{\text{targ}}$ ; however, since  $p_{\text{mid}}$  is not equal to  $p_{100}^{\text{targ}}$ , the optimal solution is not obtained by matching the target pressure. Figure 5.2 plots the pressure obtained by the optimal solution. Satisfying the pointwise constraint on the pressure causes oscillations about the target pressure that damp out as we move away from the midpoint.

TABLE 5.2

*Parameter values adopted for the nozzle inverse design and MDO problems.*

parameter	Nonlinear Problem	
	inverse	MDO
$\tau^p$	$10^{-5}$	$10^{-6}$
$\tau^d$	$10^{-6}$	$10^{-6}$
$\Delta_0$	1	1
$\Delta_{\max}$	2	2
$\eta_0$	0.5	0.5
max_iter	100	100
max_filter_iter	10	10
$\mu_0$ (FLECS only)	0.01	0.01

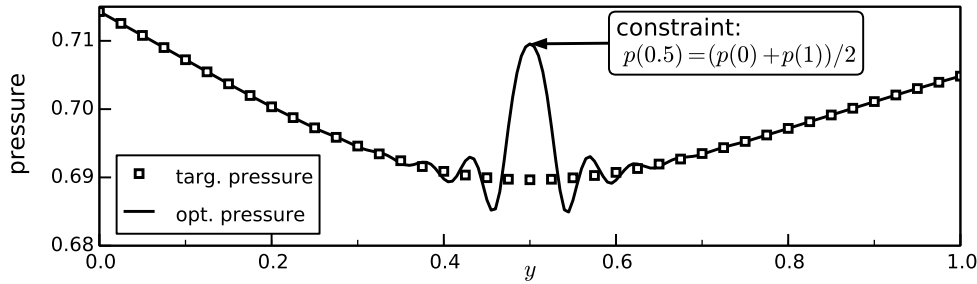


FIG. 5.2. Target pressure and optimized pressure for the nozzle inverse-design problem. The location and value of the pressure constraint is also indicated.

The FLECS-based SQO algorithm was applied to the nozzle problem using 1000 distinct initial guesses for the nozzle area. The initial guesses were obtained by computing b-spline coefficients that yield a linear variation in nozzle area between the inlet and outlet, and then perturbing these variables by uniform (pseudo) random variables drawn from  $\mathcal{U}(0.75, 1.25)$ . The same 1000 initial guesses were also supplied to the composite-step algorithm, and Figure 5.3 plots the performance profile [8] for the two algorithms over this set of guesses. The results are based on parameters listed in Table 5.2. FLECS and the iterative solvers used in the composite-step algorithm were limited to 15 iterations.

On this set of problems, the probability is 0.962 that the FLECS-based SQO is the faster of the two algorithms. The probability that the composite-step method is the fastest is 0.038. In addition, the profile shows that the FLECS-based method is more than 2 times faster than the composite-step algorithm in over 50 percent of cases. The probability that the FLECS-based algorithm fails is 0.02, while the probability that the composite-step algorithm fails is 0.026.

**5.4. Multidisciplinary Design Problem.** The second nonlinear problem is the quasi-1d flow in a linear elastic nozzle. The flow is once again modeled using the quasi-one-dimensional Euler equations; however the nozzle structure is now 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, in turn modifying the flow. Further details of this problem can be found in [7].

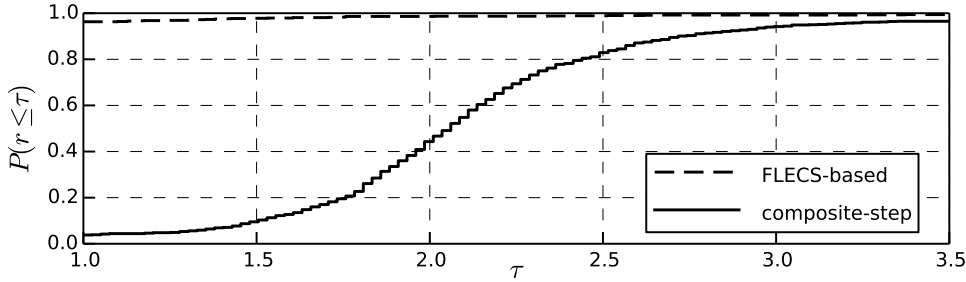


FIG. 5.3. Performance profile for the FLECS-based and composite-step algorithms applied to the nozzle inverse-design problem using random initial guesses.

As with the nozzle inverse design, the control variables are the b-spline coefficients that determine the shape of the nozzle; however, in the present case, these coefficients dictate the static nozzle shape only.

We use a particular MDO problem formulation called individual discipline feasible (IDF) [11, 5]. 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.

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

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

for all  $j \in 0, 1, 2, \dots, N$ . The objective function is the same used in the previous nozzle design problem.

As in the nozzle-design problem,  $x$  denotes the optimization variables, but these now include the  $\bar{p}_j$  and  $\bar{A}_j$  in addition to the b-spline coefficients. We consider 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 [7]. 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 solves.

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 solves that balanced loss of conjugacy with the expensive of oversolving.

To characterize the FLECS-based algorithm on the MDO problem, 200 problems were solved, each with randomly generated initial guesses. The b-spline coefficient

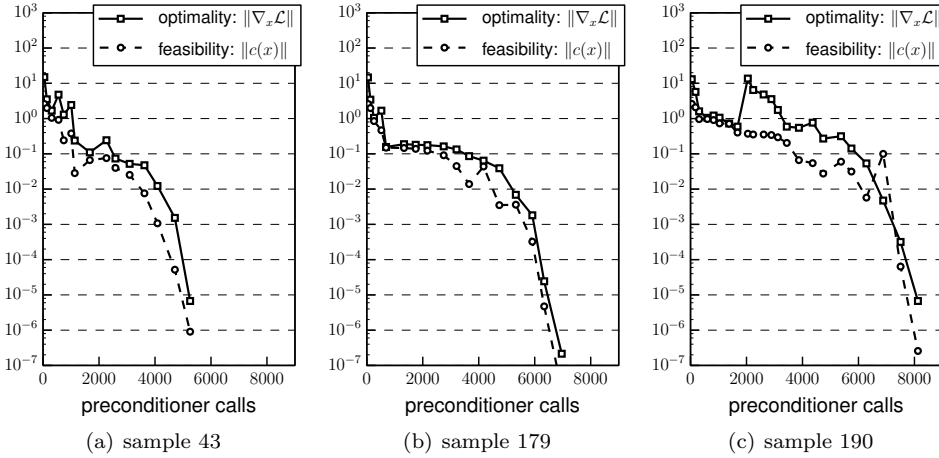


FIG. 5.4. FLECS-based algorithm convergence histories for a random set of samples from the MDO inverse-design results.

values were set using the same procedure adopted for the nozzle inverse design; that is, baseline coefficients producing a linear nozzle-area variation that are perturbed using  $\mathcal{U}(0.75, 1.25)$ . The initial coupling area and pressures were set based on this randomly perturbed, static (i.e. undeformed) nozzle.

The parameter values adopted for the MDO problem were mostly unchanged from the nozzle inverse-design problem; see Table 5.2. As with the nozzle problem, FLECS and the iterative solvers used in the composite-step algorithm were limited to 15 iterations, and samples that failed to converge in 100 nonlinear iterations were considered unsuccessful.

Figure 5.4 plots the convergence histories of the FLECS-based algorithm from 3 randomly selected samples. These results provide evidence that FLECS' trial steps can yield superlinear asymptotic convergence.

Figure 5.5 shows the performance profiles 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 3 out of 4 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 of failure is slightly higher on the MDO problem for both methods: 0.025 for the FLECS-based algorithm and 0.035 for the composite-step algorithm.

**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 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 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.

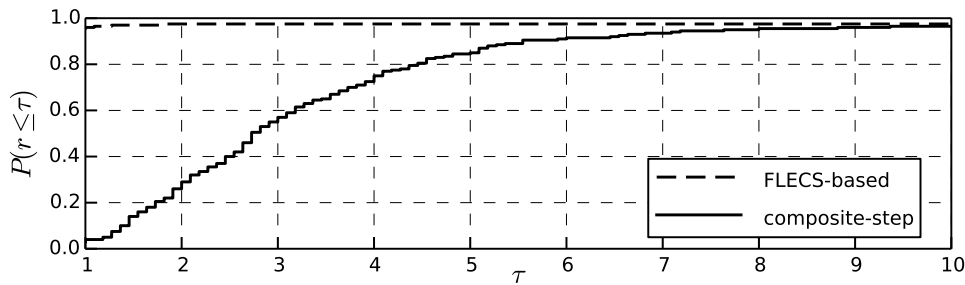


FIG. 5.5. Performance profile for the FLECS-based and composite-step algorithms applied to the MDO inverse-design problem using random initial guesses.

- 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  $\mu$ ? Finally, how should FLECS be adapted for problems with inequality constraints?

**Acknowledgments.** This work was supported by the National Science Foundation through grant number 1332819. The authors gratefully acknowledge this support. All figures were produced using Matplotlib [16].

## Appendix A. Nonlinear Optimization Algorithms.

### REFERENCES

- [1] V. AKÇELİK, 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., Society for Industrial and Applied Mathematics, Jan. 2006, ch. 16, pp. 291–322.
- [2] O. AXELSSON, *Iterative solution methods*, Cambridge University Press, 1996.
- [3] 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.
- [4] A. R. CONN, N. I. M. GOULD, AND P. L. TOINT, *Trust Region Methods*, Society for Industrial and Applied Mathematics, Jan. 2000.
- [5] E. J. CRAMER, J. E. DENNIS, P. D. FRANK, R. M. LEWIS, AND G. R. SHUBIN, *Problem Formulation for Multidisciplinary Optimization*, *SIAM Journal on Optimization*, 4 (1994), pp. 754–776.
- [6] R. S. DEMBO, S. C. EISENSTAT, AND T. STEIHAUG, *Inexact Newton methods*, *SIAM Journal on Numerical Analysis*, 19 (1982), pp. 400–408.
- [7] A. DENER AND J. E. HICKEN, *Revisiting Individual Discipline Feasible using matrix-free Inexact-Newton-Krylov*, in 10th AIAA Multidisciplinary Design Optimization Conference, American Institute of Aeronautics and Astronautics, Jan. 2014. AIAA 2014–0110.
- [8] E. D. DOLAN AND J. J. MORÉ, *Benchmarking optimization software with performance profiles*, *Mathematical Programming*, 91 (2002), pp. 201–213.
- [9] R. FLETCHER AND S. LEYFFER, *Nonlinear programming without a penalty function*, *Mathematical Programming*, 91 (2002), pp. 239–269.

---

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

---

**Data:**  $x_0, \mu_0, \eta_0, \Delta_0, \Delta_{\max}, \tau^p, \tau^d$ **Result:** optimal solution  $x$ 

```

1   $\mu = \mu_0, \Delta = \Delta_0$ 
2  for  $k = 0, 1, 2, \dots, \text{max\_iter}$  do
3      if  $\|g_k\| \leq \tau^p \|g_0\|$  and  $\|c_k\| \leq \tau^d \|c_0\|$  then    (check convergence)
4          | set  $x = x_k$  and return
5      end
6       $\mu \leftarrow \max(\mu, \mu_0 \|c(x_0)\| / \|c(x_k)\|)$ 
7       $\eta \leftarrow \max \left[ 0.001, \eta_0 \min \left( 1.0, \sqrt{\|g_k\|^2 + \|c_k\|^2} / \sqrt{\|g_0\|^2 + \|c_0\|^2} \right) \right]$ 
8      solve for primal-dual trial step  $s_k$  using FLECS, Algorithm 2, with
      tolerance  $\eta$ , penalty  $\mu$ , and trust-radius  $\Delta$ 
9      for  $i = 0, 1, 2, \dots, \text{max\_filter\_iter}$  do
10         | if  $[f(x_k + p_k), c(x_k + p_k)]$  is not dominated by filter then
11             | if  $i = 0$  then  $\Delta \leftarrow \min(2\Delta, \Delta_{\max})$ 
12             | set filter_success = true
13             | exit loop
14         | end
15         | if  $i = 0$  then    (try a second-order correction)
16             | compute the correction  $p_c = Z^p y_c$ , where  $y_c$  is the solution of
17             |
18             | 
$$\left[ (Z_j^p)^T Z_j^p + (Z_j^d)^T Z_{j+1}^d \bar{H}_j + \bar{H}_j^T (Z_{j+1}^d)^T Z_j^d \right] y_c = (Z_j^d)^T c(x_k + p_k)$$

19             |
20             | if  $[f(x_k + p_k + p_c), c(x_k + p_k + p_c)]$  is not dominated by filter then
21                 | set filter_success = true
22                 | exit loop
23             | end
24         | end
25         |  $\Delta \leftarrow \frac{1}{4} \Delta$ 
26         | re-solve for primal-dual trial step  $s_k$  using FLECS, Algorithm 2, with
27         | penalty  $\mu$  and trust-radius  $\Delta$ 
28     end
29     if filter_success then
30         |  $x_{k+1} = x_k + p_k, \lambda_{k+1} = \lambda_k + d_k$ 
31     else
32         |  $x_{k+1} = x_k, \lambda_{k+1} = \lambda_k$ 
33     end
34 end

```

---

- [10] 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.
- [11] 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. 10.1007/BF01759919.
- [12] 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.

---

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

---

**Data:**  $x_0, \eta_0, \Delta_0, \Delta_{\max}, \tau^p, \tau^d$ 
**Result:** optimal solution  $x$ 

```

1  $\Delta = \Delta_0$ 
2 for  $k = 0, 1, 2, \dots, \text{max\_iter}$  do
3   if  $\|g_k\| \leq \tau^p \|g_0\|$  and  $\|c_k\| \leq \tau^d \|c_0\|$  then    (check convergence)
4     | set  $x = x_k$  and return
5   end
6    $\eta \leftarrow \max \left[ 0.001, \eta_0 \min \left( 1.0, \sqrt{\|g_k\|^2 + \|c_k\|^2} / \sqrt{\|g_0\|^2 + \|c_0\|^2} \right) \right]$ 
7   compute quasi-normal step,  $p_\perp$ , by inexactly solving augmented system
   with right-hand side  $[0, -c_k^T]^T$  to a tolerance  $\eta$ 
8   if  $\|p_\perp\| \geq 0.8\Delta$  then  $p_\perp \leftarrow \frac{0.8\Delta}{\|p_\perp\|} p_\perp$ 
9   compute quasi-tangential step,  $p_\parallel$ , using projection CG with a tolerance  $\eta$ 
   and trust radius  $\Delta_\parallel = \sqrt{\Delta^2 - p_\perp^T p_\perp}$ 
10  set  $p_k = p_\perp + p_\parallel$ 
11  for  $i = 0, 1, 2, \dots, \text{max\_filter\_iter}$  do
12    | if  $[f(x_k + p_k), c(x_k + p_k)]$  is not dominated by filter then
13      |   if  $i = 0$  then  $\Delta \leftarrow \min(2\Delta, \Delta_{\max})$ 
14      |   set filter_success = true
15      |   exit loop
16    | end
17    | if  $i = 0$  then    (try a second-order correction)
18      |   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  $\eta$ 
19      |   if  $[f(x_k + p_k + p_c), c(x_k + p_k + p_c)]$  is not dominated by filter then
20      |   | set filter_success = true and  $p_k \leftarrow p_k + p_c$ 
21      |   | exit loop
22      | end
23    | end
24    |  $\Delta \leftarrow \frac{1}{4}\Delta$ 
25    | if  $\|p_\perp\| \geq 0.8\Delta$  then  $p_\perp \leftarrow \frac{0.8\Delta}{\|p_\perp\|} p_\perp$ 
26    | compute quasi-tangential step,  $p_\parallel$ , using projection CG with a
   tolerance  $\eta$  and trust radius  $\Delta_\parallel = \sqrt{\Delta^2 - p_\perp^T p_\perp}$ 
27    | set  $p_k = p_\perp + p_\parallel$ 
28  end
29  if filter_success then
30    |  $x_{k+1} = x_k + p_k$ 
31    | compute multiplier step,  $d_k$ , by solving augmented system with
   right-hand side  $[-g_k^T, 0]^T$  to a tolerance  $\eta$ 
32    |  $\lambda_{k+1} = \lambda_k + d_k$ 
33  else
34    |  $x_{k+1} = x_k, \lambda_{k+1} = \lambda_k$ 
35  end
36 end

```

---



- [13] J. E. HICKEN, *FLECS: Matlab implementation of the FLExible Equality-Constrained Subproblem solver*. <https://bitbucket.org/odl/flecs>, 2014. accessed: 10/8/2014.
- [14] ———, *Inexact Hessian-vector products in reduced-space differential-equation constrained optimization*, Optimization and Engineering, (2014), pp. 1–34.
- [15] ———, *Kona: a software library for DE-constrained optimization*. <https://bitbucket.org/odl/kona>, 2014. accessed: 07/14/2014.
- [16] J. D. HUNTER, *Matplotlib: A 2D graphics environment*, Computing In Science & Engineering, 9 (2007), pp. 90–95.
- [17] J. J. MORÉ AND D. C. SORENSEN, *Computing a Trust Region Step*, SIAM Journal on Scientific and Statistical Computing, 4 (1983), pp. 553–572.
- [18] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, Springer–Verlag, Berlin, Germany, second ed., 2006.
- [19] Y. NOTAY, *Flexible Conjugate Gradients*, SIAM Journal on Scientific Computing, 22 (2000), pp. 1444–1460.
- [20] E. O. OMOJOKUN, *Trust region algorithms for optimization with nonlinear equality and inequality constraints*, PhD thesis, University of Colorado at Boulder, 1989.
- [21] Y. SAAD, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM Journal on Scientific and Statistical Computing, 14 (1993), pp. 461–469.
- [22] ———, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, PA, second ed., 2003.
- [23] T. STEihaug, *The Conjugate Gradient Method and Trust Regions in Large Scale Optimization*, SIAM Journal on Numerical Analysis, 20 (1983), pp. 626–637.
- [24] P. L. TOINT, *Towards an efficient sparsity exploiting Newton method for minimization*, 1981, pp. 57–88.