SIAM J. OPTIM.
Vol. 16, No. 3, pp. 798–825

NONLINEAR INEXACT UZAWA ALGORITHMS FOR LINEAR AND NONLINEAR SADDLE-POINT PROBLEMS[*]

QIYA HU[†] AND JUN ZOU[‡]

Abstract. This paper proposes some nonlinear Uzawa methods for solving linear and nonlinear saddle-point problems. A nonlinear inexact Uzawa algorithm is first introduced for linear saddle-point problems. Two different PCG techniques are allowed in the inner and outer iterations of the algorithm. This algorithm is then extended for a class of nonlinear saddle-point problems arising from some convex optimization problems with linear constraints. For this extension, some PCG method used in the inner iteration needs to be carefully constructed so that it converges in a certain energy norm instead of the usual l^2-norm. It is shown that the new algorithm converges under some practical conditions and there is no need for any a priori estimates on the minimal and maximal eigenvalues of the two local preconditioned systems involved. The two new methods perform more efficiently than the existing methods in the cases where no good preconditioners are available for the Schur complements.

Key words. linear and nonlinear saddle-point problems, inexact Uzawa algorithm, preconditioning

AMS subject classifications. 65F10, 65N22

DOI. 10.1137/S1052623403428683

1. Introduction.

This paper is mainly concerned with the construction of efficient nonlinear inexact Uzawa algorithms for solving the nonlinear saddle-point system

$$\left\{\begin{array}{rcl} F(x) + B\,y &=& f, \\ B^t\,x &=& g, \end{array}\right. \tag{1.1}$$

where B is an $n \times m$ matrix with full column rank ($m \le n$), and $F : R^n \to R^n$ is a nonlinear vector-valued function, not necessarily differentiable.

The nonlinear saddle-point system of form (1.1) arises frequently in augmented Lagrangian formulations of inverse problems [16], electromagnetic Maxwell equations [13], [15], and nonlinear optimizations, for example, of the form (cf. [12], [22], [41])

$$\left\{\begin{array}{l} \min\limits_{x\in R^n} \{J(x) - (f,x)\} \\ \text{s.t.} \quad B^t x = g, \end{array}\right. \tag{1.2}$$

where $J(x)$ is the function satisfying $\nabla J(x) = F(x)$.

When $F(x)$ is linear, for example, $F(x) = Ax$ with A being an $n \times n$ symmetric positive definite matrix, system (1.1) reduces to the well-known (linear) saddle-point problem

$$\left\{\begin{array}{rcl} A\,x + B\,y &=& f, \\ B^t\,x &=& g. \end{array}\right. \tag{1.3}$$

[*]Received by the editors May 24, 2003; accepted for publication (in revised form) June 28, 2005; published electronically January 12, 2006.

http://www.siam.org/journals/siopt/16-3/42868.html

[†]Institute of Computational Mathematics and Scientific Engineering Computing, Academy of Mathematics and System Sciences, The Chinese Academy of Sciences, Beijing 100080, China (hqy@lsec.cc.ac.cn). The work of this author was supported by National Natural Science Foundation of China grant 10371129, National Basic Research Program of China grant 2005CB321702, and the Key Project of Natural Science Foundation of China G10531080.

[‡]Department of Mathematics, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong (zou@math.cuhk.edu.hk). The work of this author was substantially supported by Hong Kong RGC grants (Projects 403403 and 404105).

As we shall see, the Schur complement matrix

$$K = B^t A^{-1} B \tag{1.4}$$

associated with system (1.3), and its preconditioners, play an essential role in solving the saddle-point problem.

During the past decade, there has been a growing interest in preconditioned iterative methods for solving the indefinite saddle-point system of equations like (1.3); see [4], [10], [11], [18], [20], [37], [39]. The standard Uzawa-type method [2], [3] and the minimal residual (MINRES) method are the most popular iterative methods for solving (1.3). Let $\hat{A}$ and $\hat{K}$ be two positive definite matrices, which are assumed to be the preconditioners for the matrices A and K, respectively. Then it is known that the convergence rates of both the standard Uzawa-type method and the MINRES method depend on the condition numbers $\text{cond}(\hat{A}^{-1}A)$ and $\text{cond}(\hat{K}^{-1}K)$ of the two local preconditioned systems, and they are much less efficient when one of the two condition numbers is relatively larger than the other. To effectively deal with the case where $\text{cond}(\hat{A}^{-1}A)$ is relatively larger than $\text{con}(\hat{K}^{-1}K)$, a nonlinear inexact Uzawa algorithm was proposed in [10], in which the inner iteration uses a (nonlinear) iterative method to replace the action of A^{-1}. Recently we have introduced two new algorithms (Algorithms 3.1 and 4.1 in [29]) to improve the existing algorithms and convergence results. These two algorithms were, respectively, designed to effectively treat two different cases: (a) $\text{cond}(\hat{A}^{-1}A) \gg \text{cond}(\hat{K}^{-1}K)$; (b) $\text{cond}(\hat{K}^{-1}K) \gg \text{cond}(\hat{A}^{-1}A)$. It was shown that Algorithm 3.1 in [29] is efficient for case (a), but Algorithm 4.1 there may not always be efficient for case (b), as there is one condition (see (4.2) in [29]) which may not be easily guaranteed in applications.

The purpose of this paper is twofold. To better understand the difference between linear and nonlinear saddle-point problems, we first propose some improved version of Algorithm 4.1 in [29] for solving linear system (1.3). As we shall see, the new algorithm is always convergent without any assumptions on the spectra of the preconditioned systems $\widehat{K}^{-1}K$ and $\widehat{A}^{-1}A$. This seems to be an important advantage of the new algorithm over the existing iterative methods for saddle-point problems. Then we extend this improved algorithm to effectively solve nonlinear saddle-point problems of form (1.1), which is assumed to arise from some convex minimization problems with linear constraints.

To our knowledge, there have been very few investigations into the rate of convergence for preconditioned iterative methods for nonlinear saddle-point systems. Al-Baali and Fletcher studied in [1] rates of convergence of preconditioned nonlinear conjugate gradient (CG) methods for unconstrained optimizations, when the preconditioning matrix is taken to be the exact Hessian matrix at each iteration. In [14], Chen gave a deep analysis on rates of convergence of inexact Uzawa methods for nonlinear saddle-point systems, when the exact Hessian matrix at each iteration is used in the preconditioner for the Schur complement. Most existing analyses are carried out in the standard l^2-norm, which may not be so natural and accurate for many problems from applications.

In this paper, we shall make an effort to study convergence rates of inexact Uzawa algorithms in the *energy-norm* when the exact Hessian matrix in the Schur complement is replaced by some *inexact preconditioner* at each iteration. Due to the nonlinearity of the saddle-point system, the conditioning of the preconditioned Schur complement may become much worse than that of the preconditioned Hessian matrix at each iteration. In this case, a special nonlinear preconditioning process is first introduced to improve the conditioning of the preconditioned Schur complement. Further, a preconditioned nonlinear CG method is introduced in the inner iteration for the nonlinear system related to $F(\cdot)$ at each iteration. To ensure the convergence of the global inexact Uzawa algorithm, the preconditioned nonlinear CG method needs to be carefully constructed so that it converges in a certain energy norm instead of the usual l^2-norm. As we shall see, the new algorithm is always convergent without any assumptions on the spectra of the preconditioned Schur complement systems and Hessian matrices. More important, all the tolerance parameters involved in the inner iterations of the inexact Uzawa algorithm can be taken to be some fixed constants independent of the iterations, for example, 1/2 or 1/3. This appears to be an important advantage of the new algorithm over the existing ones.

Although quite different from what we are doing here, we mention another interesting and popular approach widely used in the optimization community. This approach intends to solve a nonlinear equality-constrained minimization problem by sequential quadratic programming in which successively quadratic subproblems are solved. Each quadratic subproblem amounts to solving a linear Karush–Kuhn–Tucker (KKT) saddle-point system. Global preconditioners for the resulting KKT coefficient matrices have been widely studied, and maintain the block structures of the original KKT matrices; see [5], [6], [7], [23], [32], and the references therein.

The rest of this paper is organized as follows. First, in section 2 we propose an improved variant of Algorithm 4.1 studied in [29] for linear saddle-point problems. The algorithm is then extended for nonlinear saddle-point problems in section 3, and its rate of convergence is also analyzed under some weak smoothness assumptions on the nonlinear functions $F(\cdot)$. Finally, in section 4 we apply two new algorithms proposed in sections 2 and 3 to solve an algebraic system of nonlinear saddle-point problem and a linear saddle-point problem arising from the domain decomposition method with Lagrange multiplier.

2. Nonlinear inexact Uzawa algorithms for linear saddle-point problems.

In this section, we shall propose an improved variant of Algorithm 4.1 from [29] for solving the linear saddle-point problem (1.3) and study its convergence. This improved algorithm will be extended in section 3 to solve the nonlinear saddle-point system (1.1). To do so, we need to introduce some notation. R^l will mean the usual l-dimensional Euclidean space. For any $l \times l$ positive definite matrix G, $\|x\|_G$ will represent the G-induced norm, namely $\|x\|_G = (Gx,\ x)^{1/2}$ for all $x \in R^l$. To describe the nonlinear inexact Uzawa algorithm, we introduce a nonlinear mapping $\Psi_A : R^n \to R^n$ such that for any given $\xi \in R^n$, $\Psi_A(\xi)$ is an "approximation" to the solution φ of the linear system

$$A\varphi = \xi. \tag{2.1}$$

The following assumption was often made on the accuracy of the approximation (e.g., see (4.2) in [10]):

$$\|\Psi_A(\xi) - A^{-1}\xi\|_A \le \delta \, \|A^{-1}\xi\|_A \quad \forall \xi \in R^n \tag{2.2}$$

for some $\delta \in (0, 1)$. Assumption (2.2) is natural and can be satisfied, for example, by the approximate inverse generated by the preconditioned conjugate gradient (PCG) iteration or by one sweep of a multigrid method with conjugate gradient smoothing [10].

We first recall an algorithm from [29] for solving the linear saddle-point problem (1.3).

Algorithm 2.1 (nonlinear inexact Uzawa-steepest descent). *Given* $\{x_0, y_0\} \in R^n \times R^m$, *the sequence of pairs* $\{x_i, y_i\} \in R^n \times R^m$ *is defined for* $i = 1, 2, \dots,$ *by the following.*

Step 1. *Compute* $f_i = f - (Ax_i + By_i)$ *and* $\Psi_A(f_i)$*; update*

$$x_{i+1} = x_i + \Psi_A(f_i). \tag{2.3}$$

Step 2. *Compute* $g_i = B^t x_{i+1} - g$ *and* $d_i = \hat{K}^{-1} g_i$*. Then compute the relaxation parameter*

$$\tau_i = \begin{cases} \frac{(g_i, d_i)}{(\Psi_A(Bd_i), Bd_i)} & \textit{for} \quad g_i \neq 0; \\ 1 & \textit{for} \quad g_i = 0. \end{cases} \tag{2.4}$$

Update

$$y_{i+1} = y_i + \frac{1}{2}\, \tau_i \, d_i. \tag{2.5}$$

To study the convergence of Algorithm 2.1, we assume Ψ_A satisfies

$$\|A^{-1} f_i - \Psi_A(f_i)\|_A \leq \delta_f \, \|A^{-1} f_i\|_A, \tag{2.6}$$

$$\|A^{-1} Bd_i - \Psi_A(Bd_i)\|_A \leq \delta_d \, \|A^{-1} Bd_i\|_A \tag{2.7}$$

for two positive constants $\delta_f < 1$ and $\delta_d < 1$. We remark that one can simply take δ_f and δ_d to be the constant δ in (2.2). But the introduction of these two different constants enables us to see how the rate of convergence depends more explicitly on the accuracies of the nonlinear inner iterations in (2.3) and (2.5).

To measure the convergence rate of Algorithm 2.1 more accurately, an appropriate norm is very crucial. For each element v from the product space $R^n \times R^m$, we will write it as $v = \{v_1, v_2\}$, where $v_1 \in R^n$ and $v_2 \in R^m$. Then, as we did in [28], [29], we shall use the norm

$$|||v||| = (\|v_1\|_{A^{-1}}^2 + \|v_2\|_K^2)^{\frac{1}{2}} \quad \forall\, v = \{v_1, v_2\} \in R^n \times R^m. \tag{2.8}$$

Finally, we introduce three error vectors $e_i^x \in R^n$, $e_i^y \in R^m$, and $E_i \in R^n \times R^m$:

$$e_i^x = x - x_i, \quad e_i^y = y - y_i, \quad E_i = \{\sqrt{\delta} f_i,\, e_i^y\}, \quad i = 0, 1, 2, \dots,$$

and two parameters

$$\hat{\kappa} = \mathrm{cond}(\hat{K}^{-1} K), \quad \hat{\beta} = \sqrt{1 - \frac{4\hat{\kappa}(1 - 2\delta_d)}{(1 + \hat{\kappa})^2 (1 - \delta_d)^2}}; \tag{2.9}$$

then we have the following estimates on the rate of convergence of Algorithm 2.1 in [29].

Lemma 2.1. *Assume that* (2.6) *and* (2.7) *are satisfied with the parameters* $\delta_f < \frac{1}{3}$ *and* $\delta_d < \frac{1}{2}$*; then Algorithm* 2.1 *converges. Moreover, the following estimate holds:*

$$|||E_{i+1}||| \leq \hat{\rho}\, |||E_i|||, \quad i = 0, 1, 2, \dots. \tag{2.10}$$

Also, (2.10) *implies for* $i = 1, 2, 3, \dots$ *that*

$$\|e_i^x\|_A \leq (\sqrt{1 + 4\delta_f} + \hat{\rho}) \hat{\rho}^{i-1} |||E_0|||, \quad \|e_i^y\|_K \leq \hat{\rho}^i \, |||E_0|||, \tag{2.11}$$

where the rate of convergence $\hat{\rho} < 1$ *and can be estimated by*

$$\hat{\rho} = \begin{cases} \sqrt{\delta_f + \delta_f^2} + \delta_f & \text{for} \quad 0 < \frac{1+\hat{\beta}}{2} \le \frac{4\delta_f}{1+\delta_f}; \\ 1 - \frac{1}{4}(1-\hat{\beta})(1+\delta_f) & \text{for} \quad \frac{4\delta_f}{1+\delta_f} < \frac{1+\hat{\beta}}{2} < 1. \end{cases} \tag{2.12}$$

Remark 2.1. Algorithm 2.1 converges when a general preconditioner $\hat{K}$ is used for the Schur complement system $K = B^t A^{-1} B$ and a general nonlinear iteration Ψ_A is used for solving $A\varphi = \xi$ involved in the inner iteration. However, the steepest descent method converges with a reasonable rate only when a good preconditioner is available for the Schur complement system, namely, $\hat{\kappa} = \text{cond}(\hat{K}^{-1}K)$ is not large. This is the case when the saddle-point problem arises, for example, from the Stokes problem [39]. Without such a good preconditioner the method may converge with a very slow rate. Particularly, Algorithm 2.1 may be much less effective when $\text{cond}(\hat{K}^{-1}K) \gg \text{cond}(\hat{A}^{-1}A)$.

Another algorithm (Algorithm 4.1) was proposed in [29] that combines the nonlinear inexact Uzawa algorithm with the CG method, in an effort to accelerate the nonlinear inexact Uzawa algorithm when $\text{cond}(\hat{K}^{-1}K) \gg \text{cond}(\hat{A}^{-1}A)$. This is the case when the saddle-point problems arise from the domain decomposition method with Lagrange multiplier [27], [34], or from the Lagrange multiplier formulations for optimization problems [25] and the parameter identification [16], [33]. But the algorithm still does not seem satisfactory, as its convergence can be guaranteed only under some restriction; see (4.2) in [29]. Next, we propose an improved variant of Algorithm 4.1 in [29].

Let $H = B^t \hat{A}^{-1} B$. Consider the equation

$$H\psi = g_i, \tag{2.13}$$

where $g_i = B^t x_{i+1} - g$ comes from Algorithm 2.1. We apply the PCG method with the preconditioner $\hat{K}$ to solve system (2.13) and let $\Psi_H(g_i)$ be the approximation generated by this iteration. Assume that the approximation satisfies

$$\|\Psi_H(g_i) - H^{-1}g_i\|_H \le \delta_g \, \|H^{-1}g_i\|_H \tag{2.14}$$

for some $\delta_g \in (0,1)$. For the approximation $d_i = \Psi_H(g_i)$, we introduce a relaxation parameter $\overline{\tau}_i$ such that the error

$$\|\overline{\tau}_i \, d_i - K^{-1}g_i\|_K^2$$

is minimized. If $d_i \ne 0$, the direct calculation gives

$$\overline{\tau}_i = \frac{(g_i, d_i)}{(Kd_i, d_i)} = \frac{(g_i, d_i)}{(A^{-1}Bd_i, Bd_i)}.$$

But the action of A^{-1} is usually very expensive, and thus will be replaced by the action of Ψ_A:

$$\tau_i = \frac{(g_i, d_i)}{(\Psi_A(Bd_i), Bd_i)} \approx \overline{\tau}_i. \tag{2.15}$$

With this parameter τ_i, we propose the following new algorithm.

Algorithm 2.2 (nonlinear inexact Uzawa with mixed iteration). *Given* $\{x_0, y_0\} \in R^n \times R^m$*, the sequence of pairs* $\{x_i, y_i\} \in R^n \times R^m$ *is defined for* $i = 1, 2, \ldots,$ *as follows.*

Step 1. *Compute* $f_i = f - (Ax_i + By_i)$ *and* $\Psi_A(f_i)$*; update*

$$x_{i+1} = x_i + \Psi_A(f_i). \tag{2.16}$$

Step 2. *Compute* $g_i = B^t x_{i+1} - g$ *and* $d_i = \Psi_H(g_i)$*. Then compute the parameter* τ_i*:*

$$\tau_i = \begin{cases} \frac{(g_i, d_i)}{(\Psi_A(Bd_i), Bd_i)} & \textit{if} \quad d_i \neq 0; \\ 1 & \textit{if} \quad d_i = 0 \end{cases} \tag{2.17}$$

and update

$$y_{i+1} = y_i + \frac{1}{2}\tau_i\, d_i. \tag{2.18}$$

Remark 2.2. Clearly when both f_i and g_i vanish, the vectors x_i and y_i are the exact solution of (1.3). Thus Algorithm 2.2 terminates.

Next we shall analyze the convergence of Algorithm 2.2. Let $\kappa^* = \text{cond}(\hat{A}^{-1}A)$ and $\kappa = \kappa^*(1+\delta_g)/(1-\delta_g)$. It follows from (2.14) that there is a symmetric and positive definite matrix $\hat{Q}_i$ (see Lemma 9 in [4]) such that $\hat{Q}_i^{-1}g_i = \Psi_H(g_i)$ and all eigenvalues of the matrix $\hat{Q}_i^{-1}H$ are in the interval $[1-\delta_g, 1+\delta_g]$. Using this property, one can directly check that

$$\text{cond}(\hat{Q}_i^{-1}K) \le \kappa = \frac{1+\delta_g}{1-\delta_g}\,\text{cond}(\hat{A}^{-1}A).$$

This relation tells us the *actual effect* of introducing of approximation Ψ_H: when $\text{cond}(\hat{K}^{-1}K) \gg \text{cond}(\hat{A}^{-1}A)$, the effect of $\Psi_H(g_i)$ $(= \hat{Q}_i^{-1}g_i)$ amounts to generating a new preconditioner $\hat{Q}_i$ such that $\text{cond}(\hat{Q}_i^{-1}K)$ is much more improved than $\text{cond}(\hat{K}^{-1}K)$ and has about the same magnitude as $\text{cond}(\hat{A}^{-1}A)$, e.g., less than three times $\text{cond}(\hat{A}^{-1}A)$ when we take $\delta_g = \frac{1}{2}$.

Let δ_f and δ_d be two parameters in (2.6) and (2.7), respectively, with f_i and d_i given in Algorithm 2.2, and define

$$\beta = \sqrt{1 - \frac{4\kappa(1-2\delta_d)}{(1+\kappa)^2(1-\delta_d)^2}}; \tag{2.19}$$

then Algorithm 2.2 can be viewed as a variant of Algorithm 2.1 with $\hat{K}$ replaced by $\hat{Q}_i$. The following theorem follows from Lemma 2.1.

THEOREM 2.2. *Assume that* (2.6) *and* (2.7) *are satisfied with* $\delta_f < \frac{1}{3}$ *and* $\delta_d < \frac{1}{2}$*; then Algorithm* 2.2 *converges. Moreover, the following estimate holds:*

$$|||E_{i+1}||| \le \rho\, |||E_i|||, \quad i = 0, 1 \ldots, \tag{2.20}$$

which implies for $i = 1, 2, \ldots$ *that*

$$\|e_i^x\|_A \le (\sqrt{1+4\delta_f} + \rho)\rho^{i-1}|||E_0|||, \quad \|e_i^y\|_K \le \rho^i |||E_0|||, \tag{2.21}$$

where the rate of convergence $\rho(< 1)$ *can be estimated by*

$$\rho = \begin{cases} \sqrt{\delta_f + \delta_f^2} + \delta_f & \text{for} \quad 0 < \frac{1+\beta}{2} \le \frac{4\delta_f}{1+\delta_f}; \\ 1 - \frac{1}{4}(1-\beta)(1+\delta_f) & \text{for} \quad \frac{4\delta_f}{1+\delta_f} < \frac{1+\beta}{2} < 1. \end{cases} \tag{2.22}$$

Remark 2.3. We see from Theorem 2.2 that the convergence of Algorithm 2.2 is independent of the spectrum of the preconditioned Schur complement $\hat{K}^{-1}K$, and the convergence rate of this new algorithm depends only on the condition number κ^*, not on $\operatorname{cond}(\hat{K}^{-1}K)$. In contrast to Algorithm 2.1, Algorithm 2.2 should be very efficient for the case when $\operatorname{cond}(\hat{K}^{-1}K) \gg \operatorname{cond}(\hat{A}^{-1}A)$. This seems to be an important advantage of the new algorithm over the existing iterative methods for saddle-point problems. The coefficient 1/2 in (2.18) is obtained by the worst case $\delta_g \to 1^-$ (refer to [29]). In applications, the parameter δ_g is much less than 1, so we can choose a larger parameter than 1/2 in (2.18), e.g., 7/10.

3. Nonlinear inexact Uzawa algorithms for nonlinear saddle-point problems.

In this section, we discuss how to effectively extend the new Algorithm 2.2 proposed in section 2 for the linear saddle-point problem (1.3) to solve the nonlinear saddle-point system (1.1), which is assumed to arise from some convex minimization problems with linear constraints, e.g., of the form

$$\begin{cases} \min\limits_{x\in R^n} \{J(x) - (f,x)\} \\ \text{ s.t. } \ B^t x = g, \end{cases} \tag{3.1}$$

where $J(x)$ is the function satisfying $\nabla J(x) = F(x)$.

3.1. Notation and assumptions.

We start with a few smoothness descriptions on the nonlinear mapping $F : R^n \to R^n$ in (1.1) and recall some existing results from [18] and [36], which will be used in the subsequent analysis.

As standard assumptions for nonlinear systems (cf. [1], [14]), we assume that F is Lipschitzian and strongly monotone with modulus μ, i.e.,

$$(F(\xi) - F(\eta), \xi - \eta) \ge \mu \, \|\xi - \eta\|^2 \quad \forall \, \xi, \eta \in R^n. \tag{3.2}$$

By Rademacher's theorem [18], the Lipschitzian property of F implies that F is differentiable almost everywhere. Let D_F be the set of points where F is differentiable, and let $\nabla F(\xi)$ be the gradient of F at $\xi \in D_F$. Then at any point $x \in R^n$, we introduce a set $\partial_s F(x)$:

$$\partial_s F(x) = \left\{ \lim_{\substack{\xi \to x \\ \xi \in D_F}} \nabla F(\xi) \right\}.$$

With this set, we can define a generalized Jacobian of F at x in the sense of Clarke [18] by

$$\partial F(x) = \operatorname{co} \partial_s F(x),$$

where $\operatorname{co} \partial_s F(x)$ is the convex hull of the set.

It is known (cf. [18]) that if F is locally Lipschitzian, then the following generalized mean-value theorem holds: for any $\xi,\ \eta \in R^n$,

$$F(\xi) - F(\eta) \in \operatorname{co} \partial F(\overline{\xi\eta})(\xi - \eta), \tag{3.3}$$

where $\overline{\xi\eta}$ is the line segment between ξ and η, and $\operatorname{co} \partial F(\overline{\xi\eta}) = \operatorname{co}\{V \in \partial F(\zeta),\ \zeta \in \overline{\xi\eta}\}$.

A nice consequence (cf. [36]) of the strong monotone property (3.2) is that all matrices from $\partial F(\eta)$ for any $\eta \in R^n$ are positive definite, and the following holds for any $V \in \partial F(\eta)$:

$$(V\xi,\, \xi) \ge \mu \, (\xi,\, \xi) \quad \forall \, \xi \in R^n. \tag{3.4}$$

As in [14], we do not assume F is differentiable everywhere but that it is semismooth on R^n in the sense that for any $\xi \in R^n$ there is a positive definite matrix A_ξ such that

$$\lim_{\substack{\alpha\to 0\\ A_\xi \in \partial_s F(\xi+\alpha)}} \frac{\|F(\xi+\alpha)-F(\xi)-A_\xi\alpha\|}{\|\alpha\|} = 0. \tag{3.5}$$

Nonlinear saddle-point problems with nondifferentiable but semismooth vector-valued functions F arise from some convex optimizations and numerical solutions of certain nonlinear partial differential equations; we refer to [14] for such examples.

3.2. Properties of F in terms of its generalized Jacobian. Note that all the descriptions in section 3.1 of the smoothness of the nonlinear mapping $F : R^n \to R^n$ are in terms of the l^2-norm. As we will see later, it is more accurate to interpret these smoothness properties in terms of the so-called energy-norm, that is, the induced norm by the generalized Jacobian of F, especially by the generalized Jacobian A_x of F at x, where $\{x, y\} \in R^n \times R^m$ is the exact solution of system (1.1). This will be the task of this section. For the sake of simplicity, we shall write A_x as A below.

First, directly from (3.4) and (3.5), we know that if F is semismooth on R^n, then for any $\xi \in R^n$ there is a positive definite matrix A_ξ such that

$$\lim_{\substack{\alpha\to 0\\ A_\xi \in \partial_s F(\xi+\alpha)}} \frac{\|F(\xi+\alpha)-F(\xi)-A_\xi\alpha\|_{A^{-1}}}{\|\alpha\|_A} = 0. \tag{3.6}$$

Next, by the strictly monotone and Lipschitzian property of F we immediately know there are two positive constants c_0 and C_0, which will be frequently needed later, such that

$$c_0\,\|\xi-\eta\|_A^2 \le (F(\xi)-F(\eta), \xi-\eta) \quad \forall\,\xi,\ \eta \in R^n, \tag{3.7}$$

$$\|F(\xi)-F(\eta)\|_{A^{-1}}^2 \le C_0\,\|\xi-\eta\|_A^2 \quad \forall\,\xi,\ \eta \in R^n. \tag{3.8}$$

The use of constants c_0 and C_0 is more reasonable than the use of the corresponding constants in the sense of the l^2-norm. For instance, when $F(x)$ is linear, say $F(x) = Ax$, then $c_0 = C_0 = 1$, but the corresponding constants in the sense of the l^2-norm depend on the smallest and largest eigenvalues of A.

Starting now, we shall often use $S_V(\xi, r)$ to denote a ball in R^n which is centered at point ξ, with radius r measured in the $\|\cdot\|_V$-norm. The next lemma gives two further properties of the nonlinear vector-valued function F.

LEMMA 3.1. *There are two positive constants $c_1 \le 1$ and $C_1 \ge 1$ depending only on constants c_0 and C_0 such that*

$$(F(\zeta)-F(\xi), \zeta-\xi) \le C_1(A_\eta(\zeta-\xi), \zeta-\xi) \quad \forall\zeta,\ \xi,\ \eta \in R^n, \tag{3.9}$$

$$\|F(\zeta)-F(\xi)\|_{A_\eta^{-1}} \ge c_1\,\|\zeta-\xi\|_{A_\eta} \qquad \forall\zeta,\ \xi,\ \eta \in R^n. \tag{3.10}$$

Proof. We first consider (3.9). It follows from (3.8) that for any $\zeta, \xi \in R^n$,

$$(F(\zeta)-F(\xi), \zeta-\xi) \le \|F(\zeta)-F(\xi)\|_{A^{-1}}\,\|\zeta-\xi\|_A \le \sqrt{C_0}\|\zeta-\xi\|_A^2. \tag{3.11}$$

But for any $\eta \in R^n$, by (3.6) there exists a positive number $\bar r = \bar r(\eta)$ such that

$$\|F(\eta+\alpha)-F(\eta)-A_\eta\alpha\|_{A^{-1}} \le \frac{c_0}{2}\|\alpha\|_A \quad \forall\ \alpha \in S_A(0, \bar r). \tag{3.12}$$

This, together with (3.7), leads to

$$\begin{aligned}\|\alpha\|_A^2 \le \frac{1}{c_0}(F(\eta+\alpha)-F(\eta),\alpha) &= \frac{1}{c_0}\{(F(\eta+\alpha)-F(\eta)-A_\eta\alpha,\alpha)+(A_\eta\alpha,\alpha)\}\\ &\le \frac{1}{c_0}\{\|F(\eta+\alpha)-F(\eta)-A_\eta\alpha\|_{A^{-1}}\,\|\alpha\|_A+(A_\eta\alpha,\alpha)\}\\ &\le \frac{1}{2}\|\alpha\|_A^2+\frac{1}{c_0}(A_\eta\alpha,\alpha).\end{aligned}$$

So we have

$$\|\alpha\|_A^2 \le \frac{2}{c_0}(A_\eta\alpha,\alpha) \quad \forall\alpha\in S_A(0,\overline{r}). \tag{3.13}$$

Noting that inequality (3.13) is invariant with respect to any constant scaling of α, (3.9) follows readily from (3.11) and (3.13) with $C_1 = 2\sqrt{C_0}/c_0$, or $C_1 = 1$ if $\sqrt{C_0}/c_0 < 1/2$.

Now we consider (3.10). By (3.12) and (3.8), we derive

$$\begin{aligned}(A_\eta\alpha,\alpha) &\le \|A_\eta\alpha\|_{A^{-1}}\,\|\alpha\|_A\\ &\le (\|A_\eta\alpha-F(\eta+\alpha)+F(\eta)\|_{A^{-1}}+\|F(\eta+\alpha)-F(\eta)\|_{A^{-1}})\,\|\alpha\|_A\\ &\le \left(\frac{c_0}{2}+\sqrt{C_0}\right)\|\alpha\|_A^2 \quad \forall\alpha\in S_A(0,\overline{r}),\end{aligned} \tag{3.14}$$

which is invariant up to any constant scaling of α, while by (3.7) we have

$$\|\alpha\|_A^2 \le \frac{1}{c_0}(F(\eta+\alpha)-F(\eta),\alpha) \le \frac{1}{c_0}\|F(\eta+\alpha)-F(\eta)\|_{A_\eta^{-1}}\,\|\alpha\|_{A_\eta}. \tag{3.15}$$

Then (3.10) follows immediately from (3.14) and (3.15), with $c_1 = 2c_0/(c_0+2\sqrt{C_0})$ or $c_1 = 1$ if $\sqrt{C_0}/c_0 < 1/2$. $\square$

We end this section by assuming some sort of Lipschitzian property on the generalized Jacobian of F: there exists a positive constant L such that for any two vectors $\xi,\eta\in R^n$,

$$\|A^{-\frac{1}{2}}(V_\xi-V_\eta)A^{-\frac{1}{2}}\| \le L\,\|\xi-\eta\|_A \quad \forall V_\xi\in\partial F(\xi),\ V_\eta\in\partial F(\eta). \tag{3.16}$$

3.3. Algorithms and their convergence. We are now going to extend the new Algorithm 2.2 in section 2 for the linear saddle-point problem (1.3) to solve the nonlinear saddle-point problem (1.1). As we have observed, an essential improvement of this new algorithm over the existing ones (cf. [10]) lies in the fact that its convergence is guaranteed and its rate of convergence can be estimated by assuming only constant upper bounds for the error reduction factors ($\delta_f < 1/3$ and $\delta_d < 1/2$) in the nonlinear inner iterations. These conditions may be easily satisfied, for example, by the approximate inverse generated by the PCG iteration with the preconditioner $\hat{A}$. Such loose requirements come as the consequence of the choice of a particular norm $|||\cdot|||$; see Remark 2.1 in [29]. In order to preserve this good feature in the current nonlinear saddle-point system, one should use a norm similar to $|||\cdot|||$ involving the matrix $A = A_x$. But unlike the linear saddle-point problem, the matrix A_x is not available now since it involves the x-component of the exact solution $\{x, y\}$ to system (1.1). This fact brings in one of the major difficulties of nonlinear systems and can be regarded as the main distinction between the linear and nonlinear saddle-point problems.

Now, we discuss how to extend Algorithm 2.2 of section 2 to solve the nonlinear saddle-point problem (1.1). Let $\{x_i, y_i\} \in R^n \times R^m$ be the ith iterate, and set

$$f_i = f - F(x_i) - By_i.$$

Let x_{i+1} be an approximate solution of the nonlinear equation

$$F(\xi) = f - By_i \tag{3.17}$$

such that the residual $\varepsilon_i = F(x_{i+1}) - (f - By_i)$ satisfies

$$\|\varepsilon_i\|_{A^{-1}} \leq \delta_0 \, \|f_i\|_{A^{-1}} \tag{3.18}$$

with $0 \leq \delta_0 < 1$. In general, the approximation x_{i+1} can be obtained by some iterative method with x_i as a natural initial guess. This will be discussed in detail in section 3.4.

Let $A_i = A_{x_i}$ be a positive definite matrix as defined in (3.5), and let $\hat{A}_i$ be a (positive definite) preconditioner of A_i; then we obtain an exact Schur complement at x_i and its approximation:

$$K_i = B^t A_i^{-1} B, \quad H_i = B^t \hat{A}_i^{-1} B.$$

Let $\hat{K}_i$ be a preconditioner for H_i, and set $g_i = B^t x_{i+1} - g$. Similarly to the introduction of the mapping Ψ_H in (2.14), we define a nonlinear mapping $\Psi_{H_i} : R^m \to R^m$ such that

$$\|\Psi_{H_i}(g_i) - H_i^{-1} g_i\|_{H_i} \leq \delta_g \, \|H_i^{-1} g_i\|_{H_i} \tag{3.19}$$

for some $\delta_g \in (0, 1)$. Let $d_i = \Psi_{H_i}(g_i)$; then we introduce a nonlinear solver $\Psi_{A_{i+1}} : R^n \to R^n$ satisfying

$$\|\Psi_{A_{i+1}}(Bd_i) - A_{i+1}^{-1} Bd_i\|_{A_{i+1}} \leq \gamma \, \|A_{i+1}^{-1} Bd_i\|_{A_{i+1}} \tag{3.20}$$

for some $\gamma \in [0, 1)$. As we observed in the linear saddle-point case, a relaxation parameter $\overline{\tau}_i$ (see (2.15)) is important to ensure the convergence of our new algorithm:

$$\overline{\tau}_i = \frac{(g_i, \, d_i)}{(\Psi_A(Bd_i), Bd_i)} \quad \text{for} \quad d_i \neq 0.$$

Unfortunately, the matrix $A = A_x$ is no longer available for the current nonlinear problem (1.1). One alternative is to use the approximation A_{i+1} of A; this leads to the following new choice of the relaxation parameter τ_i:

$$\tau_i = \frac{(g_i, \, d_i)}{(\Psi_{A_{i+1}}(Bd_i), Bd_i)} \quad \text{for} \quad d_i \neq 0. \tag{3.21}$$

With this parameter and the motivation of Algorithm 2.2 for linear saddle-point problems, we propose the following algorithm for solving the nonlinear saddle-point system (1.1).

Algorithm 3.1 (nonlinear inexact Uzawa with mixed iteration). *Given* $\{x_0, y_0\} \in R^n \times R^m$*, the sequence* $\{x_i, y_i\} \in R^n \times R^m$ *is defined for* $i = 1, 2, \ldots$ *as follows:*

Step 1. *Compute* x_{i+1} *such that*

$$F(x_{i+1}) = f - By_i + \varepsilon_i.$$

Step 2. *Compute* $g_i = B^t x_{i+1} - g$ *and* $d_i = \Psi_{H_i}(g_i)$. *Then compute the parameter* τ_i*:*

$$\tau_i = \begin{cases} \frac{(g_i,\, d_i)}{(\Psi_{A_{i+1}}(Bd_i),\, Bd_i)} & \text{if} \quad d_i \neq 0; \\ 1 & \text{if} \quad d_i = 0 \end{cases}$$

and update

$$y_{i+1} = y_i + \frac{1}{2}\tau_i\, d_i. \tag{3.22}$$

To understand and more accurately describe the convergence of this new algorithm, we need to introduce a few more parameters. First, by (3.6) we know that for any parameter $\omega \in (0,\ 1)$, there is a positive number r_ω such that

$$\|F(x+\alpha) - F(x) - Ax\|_{A^{-1}} \leq \omega\, \|\alpha\|_A \quad \forall \alpha \in S_A(0, r_\omega). \tag{3.23}$$

Then we introduce a constant δ_d that is the minimal positive number satisfying

$$\|\Psi_{A_{i+1}}(Bd_i) - A^{-1}Bd_i\|_A \leq \delta_d\, \|A^{-1}Bd_i\|_A. \tag{3.24}$$

Now set $r_\gamma = (1-2\gamma)/(6L)$ for any positive parameter $\gamma < \frac{1}{2}$, and

$$\kappa_i = \text{cond}(\hat{A}_i^{-1}A_i), \quad \kappa_0 = \frac{1+\delta_g}{1-\delta_g}\max_i \kappa_i, \quad \beta_0 = \sqrt{1 - \frac{4\kappa_0(1-2\delta_d)}{(1+\kappa_0)^2(1-\delta_d)^2}}, \tag{3.25}$$

we will see $\delta_d < \frac{1}{2}$ when $x_{i+1} \in S_A(x, r_\gamma)$ (Lemma 3.7), and hence we have $0 < \beta_0 < 1$.

The following few parameters will be used to describe the convergence region and convergence rate of Algorithm 3.1:

$$\rho_0 = \begin{cases} \sqrt{\delta_0 + \delta_0^2} + \delta_0 & \text{for} \quad 0 < \frac{1+\beta_0}{2} \leq \frac{4\delta_0}{1+\delta_0}, \\ 1 - \frac{1}{4}(1-\beta_0)(1+\delta_0) & \text{for} \quad \frac{4\delta_0}{1+\delta_0} < \frac{1+\beta_0}{2} < 1 \end{cases} \tag{3.26}$$

and

$$\rho_\omega = \frac{\omega\sqrt{1+\delta_0}(1+\sqrt{\delta_0})}{c_0}, \quad \omega_0 = \frac{c_0(1-\rho_0)}{\sqrt{1+\delta_0}(1+\sqrt{\delta_0})}, \quad r_\omega^* = \frac{c_0}{(1+\sqrt{\delta_0})}\min\{r_\omega,\ r_\gamma\}. \tag{3.27}$$

Our final preparation is to choose an appropriate norm in which the convergence can be ensured and well measured. As for the linear saddle-point problem, we define $|||\cdot|||$ to be the same as in (2.8), but with $A = A_x$ and $K = B^tA^{-1}B$ here.

Now we are ready to state our main results of this section about the convergence of Algorithm 3.1, whose proof will be provided in section 3.5.

THEOREM 3.2. *Let* $\omega_0 < 1$ *be a fixed positive constant,* $\omega \in (0, \omega_0)$ *be a given parameter, and* r_ω *be the maximal positive number such that estimate* (3.23) *is satisfied. Assume that* (3.20) *holds with* $\gamma < \frac{1}{2}$ *and that the initial guess* $\{x_0,\ y_0\}$ *satisfies* $|||E_0||| \leq r_\omega^*$. *If the tolerance* ε_i *in Step* 1 *satisfies* (3.18) *with* $\delta_0 < \frac{1}{3}$, *then Algorithm* 3.1 *converges, and the rate of convergence can be estimated by*

$$|||E_{i+1}||| \leq \rho_0^*\, |||E_i|||, \quad i = 0, 1, 2, \ldots, \tag{3.28}$$

where $\rho_0^* = \rho_0 + \rho_\omega < 1$, *and*

$$E_i = \{\sqrt{\delta_0}f_i, e_i^y\}, \quad f_i = f - F(x_i) - By_i, \quad e_i^y = y - y_i.$$

Remark 3.1. From Theorem 3.2 we see that the preconditioner $\hat{K}_i$ for H_i can be chosen in Algorithm 3.1 without any particular restriction, and the approximation accuracy parameters δ_0 and δ_d are independent of any other parameter, unlike in most existing algorithms. The rate of convergence ρ_0^* depends only on δ_d and β_0, and does not depend directly on $\text{cond}(\hat{K}_i^{-1}K)$; therefore no proper scalings of the preconditioner $\hat{K}_i$ are required as in the existing inexact Uzawa algorithms.

Remark 3.2. In Theorem 3.2, the initial guess $\{x_0, y_0\}$ is required to lie within a small neighborhood of $\{x, y\}$. Care must be taken for the choice of such initial guesses. In applications, the initial guess may be obtained using a globally convergent algorithm for (1.1) with one or two iterations, for example, the simple perturbation method as described below.

Given a small positive number μ, approximate (1.1) by the perturbed system

$$F(x) + By = f, \quad B^t x - \mu\, y = g. \tag{3.29}$$

Expressing y in terms of x from the second equation and then substituting it into the first equation, we obtain

$$F(x) + \frac{1}{\mu} BB^t x = f + \frac{1}{\mu} Bg. \tag{3.30}$$

One can solve this nonlinear equation using some classical iterative methods, for instance, the steepest descent method, which is known to have slow convergence but usually converges very fast at the first few iterations. Once an approximation of x is available, the approximation of y can be obtained directly from the second equation in (3.29). As this process is used to generate only an initial guess, the perturbation parameter μ need not be too small, e.g., one may take $\mu = 0.1$.

3.4. Solution of system (3.17). One major task in Algorithm 3.1 is to find some effective way to compute x_{i+1} such that the tolerance requirement (3.18) is satisfied with $\delta_0 < \frac{1}{3}$. In this subsection we will propose an iterative algorithm for computing x_{i+1} which meets the requirement.

Let x_i^* be the exact solution of (3.17); then we have $f - By_i = F(x_i^*)$, and (3.18) can be written as

$$\|F(x_{i+1}) - F(x_i^*)\|_{A^{-1}} \le \delta_0 \|F(x_i) - F(x_i^*)\|_{A^{-1}}. \tag{3.31}$$

When nonlinear equation (3.17) is solved by an iterative method with the initial guess x_i, condition (3.31) should come from the convergence results in the underlying norm. Unfortunately, this conclusion is not straightforward here since the convergences of *most iterative methods* for nonlinear equations are analyzed in *the l^2-norm* (cf. [30], [31], [36]), not in the "energy-norm" as required in (3.31).

Let G_i be a functional defined by

$$G_i(\xi) = J(\xi) + (By_i, \xi) - (f, \xi),$$

where $J(x)$ is the functional in (3.1) satisfying $\nabla J(x) = F(x)$. Then (3.17) amounts to the following minimization problem: Find $x_i^* \in R^n$ such that

$$G_i(x_i^*) = \min_{\xi \in R^n} G_i(\xi). \tag{3.32}$$

Next, we propose a PCG-type method to solve this minimization problem.

Algorithm 3.2 (PCG-type method for solving (3.32)). *Set*

$$x_i^{(0)} = x_i, \quad p_i^{(0)} = -\hat{A}_i^{-1}\nabla G_i(x_i),$$

then generate a sequence $\{x_i^{(k)}\}_{k=1}^{\infty}$ *as follows.*

Step 1. *Compute the parameter* $\tau_i^{(k-1)}$ *such that*

$$G_i(x_i^{(k-1)} + \tau_i^{(k-1)} p_i^{(k-1)}) = \min_{\tau} G_i(x_i^{(k-1)} + \tau\, p_i^{(k-1)}). \tag{3.33}$$

Update

$$x_i^{(k)} = x_i^{(k-1)} + \tau_i^{(k-1)} p_i^{(k-1)}. \tag{3.34}$$

Step 2. *Compute*

$$\theta_i^{(k)} = -(\hat{A}_i^{-1}\nabla G_i(x_i^{(k)}),\, A_{x_i^{(k)}} p_i^{(k-1)})/(A_{x_i^{(k)}} p_i^{(k-1)},\ p_i^{(k-1)}). \tag{3.35}$$

Compute

$$p_i^{(k)} = -\hat{A}_i^{-1}\nabla G_i(x_i^{(k)}) - \theta_i^{(k)} p_i^{(k-1)}. \tag{3.36}$$

Before analyzing the convergence of Algorithm 3.2, let us introduce a few useful constants. For the sake of simplicity, we shall write $A_i^{(k)} = A_{x_i^{(k)}}$ below. First, we can easily see the existence of the two constants $C_i^{(k)}$ and $c_i^{(k)}$ from Lemma 3.1 and by noting the relation

$$\nabla G_i(\xi + \alpha) - \nabla G_i(\xi) = \nabla J(\xi + \alpha) - \nabla J(\xi) = F(\xi + \alpha) - F(\xi).$$

The first constant $C_i^{(k)} \geq 1$ is the smallest positive number satisfying

$$(\nabla G_i(\xi + \alpha) - \nabla G_i(\xi), \alpha) \leq C_i^{(k)} (A_i^{(k)}\alpha, \alpha) \tag{3.37}$$

for $\xi = x_i^{(k)}$ and $\alpha = s\, p_i^{(k)}$ with all $s > 0$, also for $\xi = x_i^*$ and $\alpha = t(x_i^{(k)} - x_i^*)$ with all $t \in (0,1)$; the second constant $c_i^{(k)} \leq 1$ is the largest positive number satisfying

$$c_i^{(k)} \|x_i^{(k)} - x_i^*\|_{A_i^{(k)}} \leq \|\nabla G_i(x_i^{(k)}) - \nabla G_i(x_i^*)\|_{(A_i^{(k)})^{-1}}\,. \tag{3.38}$$

It is obvious that for a less accurate estimate one may simply take the above two constants $C_i^{(k)}$ and $c_i^{(k)}$ to be the constants C_1 and c_1 from (3.9) and (3.10), respectively.

Now define

$$\kappa_i^{(k)} = \mathrm{cond}(\hat{A}_i^{-\frac{1}{2}} A_i^{(k)} \hat{A}_i^{-\frac{1}{2}}), \quad \rho_i^{(k)} = 1 - \left(\frac{c_i^{(k)}}{C_i^{(k)}}\right)^2 \frac{4\,\kappa_i^{(k)}}{(\kappa_i^{(k)}+1)^2}.$$

Clearly, we see that $\rho_i^{(k)}$ lies in the range $0 < \rho_i^{(k)} < 1$. The following estimate holds on the rate of convergence with Algorithm 3.2.

Lemma 3.3. *Let* x_i^* *be the exact solution of* (3.32)*, and let the sequence* $\{x_i^{(k)}\}_{k=1}^{\infty}$ *be generated by Algorithm* 3.2*. Then the following estimate holds:*

$$G_i(x_i^{(k+1)}) - G_i(x_i^*) \leq \rho_i^{(k)}\,(G_i(x_i^{(k)}) - G_i(x_i^*)), \quad k = 0, 1 \ldots. \tag{3.39}$$

If we set $x_{i+1} = x_i^{(k_0)}$ *for some positive integer* k_0*, then*

$$G_i(x_{i+1}) - G_i(x_i^*) \le \left(\prod_{k=1}^{k_0} \rho_i^{(k-1)} \right) (G_i(x_i^{(k)}) - G_i(x_i^*)). \tag{3.40}$$

Proof. Using the generalized mean-value theorem and (3.34), we have for any $\tau > 0$,

$$\begin{aligned} &G_i(x_i^{(k)} + \tau p_i^{(k)}) - G_i(x_i^{(k)}) \\ &= \int_0^1 \Big(\nabla G_i(x_i^{(k)} + t\,\tau p_i^{(k)}),\, \tau p_i^{(k)}\Big) dt \\ &= \tau \Big(\nabla G_i(x_i^{(k)}),\, p_i^{(k)}\Big) + \int_0^1 \Big(\nabla G_i(x_i^{(k)} + t\,\tau p_i^{(k)}) - \nabla G_i(x_i^{(k)}),\, \tau p_i^{(k)}\Big) dt. \end{aligned} \tag{3.41}$$

But it follows from (3.37) with $\xi = x_i^{(k)}$ and $\alpha = t\,\tau p_i^{(k)}$ that

$$\Big(\nabla G_i(x_i^{(k)} + t\,\tau p_i^{(k)}) - \nabla G_i(x_i^{(k)}),\, \tau p_i^{(k)}\Big) \le C_i^{(k)} t\, \tau^2 (A_i^{(k)} p_i^{(k)},\, p_i^{(k)}).$$

Plugging this into (3.41) yields

$$G_i(x_i^{(k)} + \tau p_i^{(k)}) - G_i(x_i^{(k)}) \le \tau (\nabla G_i(x_i^{(k)}),\ p_i^{(k)}) + \frac{1}{2} C_i^{(k)} \tau^2 (A_i^{(k)} p_i^{(k)},\ p_i^{(k)}). \tag{3.42}$$

Noting that the parameter $\tau_i^{(k)}$ defining $x_i^{(k+1)}$ satisfies (3.33), we obtain

$$G_i(x_i^{(k+1)}) - G_i(x_i^{(k)}) \le - \frac{(\nabla G_i(x_i^{(k)}),\ p_i^{(k)})^2}{2C_i^{(k)} (A_i^{(k)} p_i^{(k)},\ p_i^{(k)})} \tag{3.43}$$

if we take in (3.42) that

$$\tau = - \frac{(\nabla G_i(x_i^{(k)}),\ p_i^{(k)})}{C_i^{(k)} (A_i^{(k)} p_i^{(k)},\ p_i^{(k)})}.$$

To further estimate the fraction in (3.43), we first know from (3.33) that

$$(\nabla G_i(x_i^{(k)}),\ p_i^{(k-1)}) = 0. \tag{3.44}$$

Using this, and making the scalar product of both sides of (3.36) with $\nabla G_i(x_i^{(k)})$, we derive

$$(\nabla G_i(x_i^{(k)}),\ p_i^{(k)}) = -\|\hat{A}_i^{-\frac{1}{2}} \nabla G_i(x_i^{(k)})\|^2. \tag{3.45}$$

On the other hand, by direct computing using (3.36) and (3.35) we get

$$\begin{aligned} (A_i^{(k)} p_i^{(k)},\ p_i^{(k)}) &= \|\hat{A}_i^{-1} \nabla G_i(x_i^{(k)})\|^2_{A_i^{(k)}} - \frac{(\hat{A}_i^{-1} \nabla G_i(x_i^{(k)}),\ A_i^{(k)} p_i^{(k-1)})^2}{\|p_i^{(k-1)}\|^2_{A_i^{(k)}}} \\ &\le \|\hat{A}_i^{-1} \nabla G_i(x_i^{(k)})\|^2_{A_i^{(k)}}. \end{aligned} \tag{3.46}$$

Now it follows from (3.45), (3.46), (3.43), and a well-known matrix-eigenvalue inequality (see (3.1) in [29]) that

$$G_i(x_i^{(k+1)}) - G_i(x_i^{(k)}) \le -\frac{\|\hat{A}_i^{-\frac{1}{2}}\nabla G_i(x_i^{(k)})\|^4}{2C_i^{(k)}\|\hat{A}_i^{-1}\nabla G_i(x_i^{(k)})\|^2_{A_i^{(k)}}}$$

$$\le -\frac{1}{2C_i^{(k)}}\,\frac{4\kappa_i^{(k)}}{(\kappa_i^{(k)}+1)^2}\|\nabla G_i(x_i^{(k)})\|^2_{(A_i^{(k)})^{-1}}. \tag{3.47}$$

But noting $\nabla G_i(x_i^*) = 0$, we deduce from (3.38) that

$$\|\nabla G_i(x_i^{(k)})\|^2_{(A_i^{(k)})^{-1}} = \|\nabla G_i(x_i^{(k)}) - \nabla G_i(x_i^*)\|^2_{(A_i^{(k)})^{-1}} \ge (c_i^{(k)})^2\,\|x_i^{(k)} - x_i^*\|^2_{A_i^{(k)}}.$$

This, along with (3.47), implies

$$G_i(x_i^{(k+1)}) - G_i(x_i^{(k)}) \le -\frac{(c_i^{(k)})^2}{2C_i^{(k)}}\,\frac{4\kappa_i^{(k)}}{(\kappa_i^{(k)}+1)^2}\|x_i^{(k)} - x_i^*\|^2_{A_i}. \tag{3.48}$$

Furthermore, by the generalized mean-value theorem and the fact that $\nabla G_i(x_i^*) = 0$ again, we can write

$$G_i(x_i^{(k)}) - G_i(x_i^*) = \int_0^1 \Big(\nabla G_i(x_i^* + t(x_i^{(k)} - x_i^*)) - \nabla G_i(x_i^*),\, x_i^{(k)} - x_i^*\Big)dt.$$

Taking $\xi = x_i^*$ and $\alpha = t\,(x_i^{(k)} - x_i^*)$ in (3.37), we come to

$$G_i(x_i^{(k)}) - G_i(x_i^*) \le \frac{C_i^{(k)}}{2}\|x_i^{(k)} - x_i^*\|^2_{A_i^{(k)}}.$$

Combining this with (3.48) leads to

$$G_i(x_i^{(k+1)}) - G_i(x_i^{(k)}) \le -\left(\frac{c_i^{(k)}}{C_i^{(k)}}\right)^2\frac{4\kappa_i^{(k)}}{(\kappa_i^{(k)}+1)^2}(G(x_i^{(k)}) - G(x_i^*)).$$

Therefore

$$G_i(x_i^{(k+1)}) - G_i(x_i^*) \le \left(1 - \left(\frac{c_i^{(k)}}{C_i^{(k)}}\right)^2\frac{4\kappa_i^{(k)}}{(\kappa_i^{(k)}+1)^2}\right)(G(x_i^{(k)}) - G(x_i^*)),$$

which proves the desired result. □

Remark 3.3. Algorithm 3.2 is always convergent, so we have $x_i^{(k)} \to x_i^*$ and $p_i^{(k)} \to 0$ as $k \to +\infty$. Then by (3.6) one can verify that $C_i^{(k)} \to 1$ and $c_i^{(k)} \to 1$ as $k \to +\infty$. This implies

$$\lim_{k\to\infty}\rho_i^{(k)} = 1 - \frac{4\kappa_i}{(\kappa_i+1)^2} = \Big(\frac{1-\kappa_i}{1+\kappa_i}\Big)^2 \quad \text{with} \quad \kappa_i = \mathrm{cond}(\hat{A}_i^{-1}A_i).$$

Remark 3.4. The second inequality in (3.46) cannot become an equality except that

$$(\hat{A}_i^{-1}\nabla G_i(x_i^{(k)}),\, A_i^{(k)}p_i^{(k-1)}) = 0.$$

But this orthogonality does not hold, since we have only $(\nabla G_i(x_i^{(k)}),\, p_i^{(k-1)}) = 0$ but $\hat{A}_i \neq A_i^{(k)}$ in general. So (3.46) should be a strict inequality, and in turn the actual rate of convergence of Algorithm 3.2 is faster than that described by (3.39), i.e., the convergence rate of the steepest descent method (cf. [35]). Also, we remark that the exact line search is assumed in Algorithm 3.2. For the cases with inexact line searches, the relaxation parameter $\theta_i^{(k)}$ needs to be corrected somehow, and the discussion is much more technical (cf. [1], [19], [40]).

With the convergence rate estimate given by Lemma 3.3, the next lemma discusses how to meet the tolerance condition (3.18).

LEMMA 3.4. *For a given pair $\{x_i, y_i\}$ in $R^n \times R^m$, let $\{x_i^{(k)}\}_{k=1}^{\infty}$ be a sequence generated by Algorithm* 3.2 *for the minimization problem* (3.32). *Then for any $\delta_0 \in (0,1)$, there exists an integer k_0 depending on δ_0 such that with $x_{i+1} = x_i^{(k_0)}$, the residual $\varepsilon_i = F(x_{i+1}) - (f - By_i)$ satisfies the tolerance condition* (3.18).

Proof. Let x_i^* be the minimizer in (3.32) and the solution to (3.17). It follows from (3.8) that

$$\|\varepsilon_i\|_{A^{-1}}^2 = \|F(x_{i+1}) - F(x_i^*)\|_{A^{-1}}^2 \le C_0 \|x_{i+1} - x_i^*\|_A^2. \tag{3.49}$$

By the mean-value theorem and the fact that $\nabla G_i(x_i^*) = 0$, we can write

$$\begin{aligned} G_i(x_{i+1}) - G_i(x_i^*) &= \int_0^1 (\nabla G_i(x_i^* + t(x_{i+1} - x_i^*)),\, x_{i+1} - x_i^*)dt \\ &= \int_0^1 (\nabla G_i(x_i^* + t(x_{i+1} - x_i^*)) - \nabla G_i(x_i^*),\, x_{i+1} - x_i^*)dt \\ &= \int_0^1 (F(x_i^* + t(x_{i+1} - x_i^*)) - F(x_i^*),\, x_{i+1} - x_i^*)dt. \end{aligned}$$

This, along with (3.7), leads to

$$G_i(x_{i+1}) - G_i(x_i^*) \ge c_0 \int_0^1 t\, \|x_{i+1} - x_i^*\|_A^2 dt = \frac{c_0}{2} \|x_{i+1} - x_i^*\|_A^2;$$

combining it with (3.49) and (3.39), we have

$$\|\varepsilon_i\|_{A^{-1}}^2 \le \frac{2C_0}{c_0}(G_i(x_{i+1}) - G_i(x_i^*)) \le \frac{2C_0\, \hat{\delta}_i}{c_0}(G_i(x_i) - G_i(x_i^*)) \tag{3.50}$$

with $\hat{\delta}_i = \prod_{k=1}^{k_0} \rho_i^{(k-1)}$. On the other hand, using the mean-value theorem and (3.8), we see

$$\begin{aligned} G_i(x_i) - G_i(x_i^*) &= \int_0^1 (\nabla G_i(x_i^* + t(x_i - x_i^*)),\, x_i - x_i^*)dt \\ &= \int_0^1 (F(x_i^* + t(x_i - x_i^*)) - F(x_i^*),\, x_i - x_i^*)dt \\ &\le \frac{C_0}{2} \|x_i - x_i^*\|_A^2. \end{aligned} \tag{3.51}$$

But it follows from (3.7) that

$$\|x_i - x_i^*\|_A^2 \le \frac{1}{c_0}(F(x_i) - F(x_i^*), x_i - x_i^*) \le \frac{1}{c_0} \|F(x_i) - F(x_i^*)\|_{A^{-1}}\, \|x_i - x_i^*\|_A,$$

which implies that

$$\|x_i - x_i^*\|_A \le \frac{1}{c_0}\|F(x_i) - F(x_i^*)\|_{A^{-1}}.$$

This combined with (3.51) gives

$$G_i(x_i) - G_i(x_i^*) \le \frac{C_0}{2c_0^2}\|F(x_i) - F(x_i^*)\|^2_{A^{-1}}.$$

Now we obtain from (3.50) that

$$\|\varepsilon_i\|^2_{A^{-1}} \le \frac{\hat{\delta}_i C_0^2}{c_0^3}\|F(x_i) - F(x_i^*)\|^2_{A^{-1}} = \frac{\hat{\delta}_i C_0^2}{c_0^3}\|f_i\|^2_{A^{-1}},$$

which leads to the satisfaction of (3.18) when k_0 is chosen such that $\hat{\delta}_i \le \frac{c_0^3\delta_0^2}{C_0^2}$. ∎

3.5. An analysis on the convergence of Algorithm 3.1. We are now ready to study the convergence of Algorithm 3.1 and show Theorem 3.2. For this purpose, we need a few auxiliary lemmas. We remark that all notation below will be the same as in subsection 3.3. But for the sake of convenience, let us recall some frequently used notation here: $\{x, y\}$ is the exact solution to (1.1), A_ξ is a positive definite matrix defined by (3.6) at any given point $\xi \in R^n$, but with A_x simply denoted as A and A_{x_i} as A_i. The following are some error, or residual, vector quantities:

$$E_i = \{\sqrt{\delta_0} f_i, e_i^y\}, \quad f_i = f - F(x_i) - By_i, \quad e_i^x = x - x_i, \quad e_i^y = y - y_i.$$

The first lemma below gives conditions on the current approximation pair $\{x_i, y_i\}$ to ensure x_{i+1} lies in a specified neighborhood of x.

LEMMA 3.5. *Let r_ω be a fixed positive number, and let the pair $\{x_i, y_i\}$ be given such that $|||E_i||| \le \frac{c_0}{(1+\sqrt{\delta_0})} r_\omega$. If $x_{i+1} \in R^n$ is generated such that condition* (3.18) *holds for the residual $\varepsilon_i = F(x_{i+1}) - (f - By_i)$, then $x_{i+1} \in S_A(x, r_\omega)$.*

Proof. We know from the first equation of (1.1) that $f = F(x) + By$; hence

$$F(x_{i+1}) - F(x) = B(y - y_i) + \varepsilon_i. \tag{3.52}$$

But it follows from (3.7) that

$$\begin{aligned} c_0\|x_{i+1} - x\|_A^2 &\le (F(x_{i+1}) - F(x), x_{i+1} - x) \\ &= (B(y - y_i) + \varepsilon_i, x_{i+1} - x) \\ &\le \|B(y - y_i) + \varepsilon_i\|_{A^{-1}}\, \|x_{i+1} - x\|_A, \end{aligned}$$

which implies

$$\|x_{i+1} - x\|_A \le \frac{1}{c_0}\|B(y - y_i) + \varepsilon_i\|_{A^{-1}} \le \frac{1}{c_0}(\|B(y - y_i)\|_{A^{-1}} + \|\varepsilon_i\|_{A^{-1}}).$$

This, along with (3.18), leads to

$$\begin{aligned} \|x_{i+1} - x\|_A &\le \frac{1}{c_0}(\|B(y - y_i)\|_{A^{-1}} + \delta_0\|f_i\|_{A^{-1}}) \\ &= \frac{1}{c_0}(\|e_i^y\|_K + \sqrt{\delta_0}\|\sqrt{\delta_0} f_i\|_{A^{-1}}) \\ &\le \frac{(1 + \sqrt{\delta_0})}{c_0}|||E_i|||, \end{aligned} \tag{3.53}$$

which proves the desired result. ∎

The next lemma demonstrates that the matrix $A = A_x$ will be close to A_{i+1} as long as x_{i+1} stays close to x.

LEMMA 3.6. *For any given positive $\varepsilon < 1$, and any $x_{i+1} \in S_A(x, \varepsilon/(2L))$, we have*

$$\|(A - A_{i+1})\alpha\|_{A^{-1}} \le \varepsilon \, \|\alpha\|_A \quad \forall \alpha \in R^n. \tag{3.54}$$

So all the eigenvalues of the matrix $A^{-1}A_{i+1}$ lie in the interval $[1-\varepsilon, 1+\varepsilon]$, and

$$\|\alpha\|_A \le \frac{\|\alpha\|_{A_{i+1}}}{\sqrt{1-\varepsilon}} \quad \forall \alpha \in R^n. \tag{3.55}$$

Proof. Clearly (3.54) is invariant with respect to any constant scaling of α. Therefore it suffices to show that there is a number $\alpha_0 > 0$ such that (3.54) holds for all $\alpha \in R^n$ satisfying $\|\alpha\|_A \le \alpha_0$. To show this, we rewrite $(A - A_{i+1})\alpha$ as

$$\begin{aligned}(A - A_{i+1})\alpha = {} & [A\alpha - (F(x+\alpha) - F(x))] + [F(x_{i+1}+\alpha) - F(x_{i+1}) - A_{i+1}\alpha] \\ & + [F(x+\alpha) - F(x) - (F(x_{i+1}+\alpha) - F(x_{i+1}))],\end{aligned}$$

then by the triangle inequality we have

$$\begin{aligned}\|(A - A_{i+1})\alpha\|_{A^{-1}} \le {} & \|A\alpha - (F(x+\alpha) - F(x))\|_{A^{-1}} \\ & + \|F(x_{i+1}+\alpha) - F(x_{i+1}) - A_{i+1}\alpha\|_{A^{-1}} \\ & + \|F(x+\alpha) - F(x) - (F(x_{i+1}+\alpha) - F(x_{i+1}))\|_{A^{-1}}.\end{aligned} \tag{3.56}$$

But for any given positive $\varepsilon < 1$, we know from (3.6) that there is a positive number α_0 such that the following two estimates hold for any $\alpha \in R^n$ satisfying $\|\alpha\|_A \le \alpha_0$,

$$\|A\alpha - (F(x+\alpha) - F(x))\|_{A^{-1}} \le \frac{\varepsilon}{4} \|\alpha\|_A, \tag{3.57}$$

$$\|F(x_{i+1}+\alpha) - F(x_{i+1}) - A_{i+1}\alpha\|_{A^{-1}} \le \frac{\varepsilon}{4} \|\alpha\|_A. \tag{3.58}$$

Let $G(\xi) = F(\xi+\alpha) - F(\xi)$. Clearly, the Lipschitzian property of F implies the same property for G. Thus by (3.3) there is a matrix $V \in \operatorname{co} \partial G(\overline{x_{i+1}x})$ such that

$$F(x+\alpha) - F(x) - (F(x_{i+1}+\alpha) - F(x_{i+1})) = G(x) - G(x_{i+1}) = V(x - x_{i+1});$$

this gives

$$\|F(x+\alpha) - F(x) - (F(x_{i+1}+\alpha) - F(x_{i+1}))\|_{A^{-1}} \le \|A^{-\frac{1}{2}} V A^{-\frac{1}{2}}\| \, \|x - x_{i+1}\|_A. \tag{3.59}$$

As $G(\xi) = F(\xi+\alpha) - F(\xi)$, we have

$$\operatorname{co} \partial G(\overline{x_{i+1}x}) = \operatorname{co} \partial F(\alpha + \overline{x_{i+1}x}) - \operatorname{co} \partial F(\overline{x_{i+1}x}),$$

so it follows from (3.16) that

$$\|A^{-\frac{1}{2}} V A^{-\frac{1}{2}}\| \le L \, \|\alpha\|_A.$$

This with (3.59) yields

$$\|F(x+\alpha) - F(x) - (F(x_{i+1}+\alpha) - F(x_{i+1}))\|_{A^{-1}} \le L \|x - x_{i+1}\|_A \, \|\alpha\|_A.$$

Thus for any $x_{i+1} \in S_A(x, \varepsilon/(2L))$, we have

$$\|F(x+\alpha) - F(x) - (F(x_{i+1}+\alpha) - F(x_{i+1}))\|_{A^{-1}} \leq \frac{\varepsilon}{2}\,\|\alpha\|_A;$$

this, along with (3.57) and (3.58), proves (3.54).

Now by writing (3.54) as

$$\|(I - A^{-1}A_{i+1})\alpha\|_A \leq \varepsilon\,\|\alpha\|_A,$$

we see that the eigenvalues of $A^{-1}A_{i+1}$ must lie in the interval $[1-\varepsilon, 1+\varepsilon]$. This fact further implies

$$(A(A^{-1}A_{i+1})\alpha, \alpha) \geq (1-\varepsilon)(A\alpha, \alpha) \quad \forall\, \alpha \in R^n;$$

therefore,

$$\begin{aligned}(A\alpha,\alpha) &= (A_{i+1}\alpha,\alpha) + (A\alpha,\alpha) - (A(A^{-1}A_{i+1})\alpha,\alpha)\\ &\leq (A_{i+1}\alpha,\alpha) + \varepsilon\,(A\alpha,\alpha),\end{aligned}$$

which leads to estimate (3.55). ∎

Remark 3.5. We see from the proof of Lemma 3.6 that estimate (3.54) is a direct consequence of (3.5) or (3.6). If F is smooth, then inequality (3.54) amounts to the condition that the gradient of F is Lipschitzian.

LEMMA 3.7. *For a given parameter $\gamma < \frac{1}{2}$, assume $x_{i+1} \in S_A(x, \varepsilon/(2L))$ with $\varepsilon < (1-2\gamma)/3$, and* (3.20) *is satisfied. Then* (3.24) *holds with $\delta_d < \frac{1}{2}$.*

Proof. For simplicity, we write $b = Bd_i$ and $\Psi(b) = \Psi_{A_{i+1}}(Bd_i)$. Then it suffices to verify

$$\|\Psi(b) - A^{-1}b\|_A \leq \frac{1}{2}\|A^{-1}b\|_A \tag{3.60}$$

under the condition

$$\|\Psi(b) - A_{i+1}^{-1}b\|_{A_{i+1}} \leq \gamma\,\|A_{i+1}^{-1}b\|_{A_{i+1}}. \tag{3.61}$$

To see this, first by the triangle inequality,

$$\|\Psi(b) - A^{-1}b\|_A \leq \|\Psi(b) - A_{i+1}^{-1}b\|_A + \|A_{i+1}^{-1}b - A^{-1}b\|_A. \tag{3.62}$$

Using (3.55) and (3.61) above we derive

$$\|\Psi(b) - A_{i+1}^{-1}b\|_A \leq \frac{\|\Psi(b) - A_{i+1}^{-1}b\|_{A_{i+1}}}{\sqrt{1-\varepsilon}} \leq \frac{\gamma}{\sqrt{1-\varepsilon}}\|A_{i+1}^{-1}b\|_{A_{i+1}}. \tag{3.63}$$

On the other hand, it follows from (3.54) and (3.55) that

$$\|A_{i+1}^{-1}b - A^{-1}b\|_A = \|(A - A_{i+1})A_{i+1}^{-1}b\|_{A^{-1}} \leq \varepsilon\,\|A_{i+1}^{-1}b\|_A \leq \frac{\varepsilon}{\sqrt{1-\varepsilon}}\|A_{i+1}^{-1}b\|_{A_{i+1}}.$$

Substituting this and (3.63) into (3.62), and using Lemma 3.6 again, leads to

$$\begin{aligned}\|\Psi(b) - A^{-1}b\|_A &\leq \frac{\gamma+\varepsilon}{\sqrt{1-\varepsilon}}\|A_{i+1}^{-1}b\|_{A_{i+1}} = \frac{\gamma+\varepsilon}{\sqrt{1-\varepsilon}}\|b\|_{A_{i+1}^{-1}}\\ &\leq \frac{\gamma+\varepsilon}{1-\varepsilon}\|b\|_{A^{-1}} = \frac{\gamma+\varepsilon}{1-\varepsilon}\|A^{-1}b\|_A.\end{aligned} \tag{3.64}$$

This proves (3.60) by noting $\frac{\gamma+\varepsilon}{1-\varepsilon} < \frac{1}{2}$ when $\varepsilon < \frac{1-2\gamma}{3}$. ∎

Recalling that g_i, τ_i, and $\Psi_{H_i}(g_i)$ are the quantities used in Algorithm 3.1, that the parameter β_0 is defined in (3.25), and that $K = B^tA^{-1}B$ is the exact Schur complement with $A = A_x$, using (3.24) ensured by Lemma 3.7 we can derive the following lemma, which is basically the same as Lemma 3.1 in [29] (but with different notation).

LEMMA 3.8. *For a given parameter $\gamma < \frac{1}{2}$, assume $x_{i+1} \in S_A(x, \varepsilon/(2L))$ with $\varepsilon < (1-2\gamma)/3$, and* (3.20) *is satisfied. Then there is a symmetric and positive definite matrix Q_i such that*

(i) $Q_i^{-1}g_i = \frac{1}{2}\,\tau_i\,\Psi_{H_i}(g_i)$;

(ii) *all eigenvalues of the matrix $Q_i^{-1}K$ are in the interval* $[(1-\beta_0)/2, 1]$.

The following lemma is basically Lemma 3.5 in [28], with slight modifications.

LEMMA 3.9. *Let N be an $n \times n$ symmetric and positive semidefinite matrix, and let $\mathcal{F}(N)$ be a block matrix given by*

$$\mathcal{F}(N) = \begin{pmatrix} -\delta_0(I+N) & -\sqrt{\delta_0}N \\ -\sqrt{\delta_0}N & (I-N) \end{pmatrix}.$$

If all positive eigenvalues of N lie in the interval $[1-\frac{1+\beta_0}{2}, 1]$, then we have $\|\mathcal{F}(N)\| \le \rho_0$, where $\rho_0 < 1$ is defined in (3.26).

Proof of Theorem 3.2. With the previous technical preparations, we are now ready to demonstrate Theorem 3.1. First, we recall that ρ_0, ρ_ω, and r_ω^* are three parameters defined in (3.26) and (3.27). Then for Theorem 3.2 it suffices to prove that $\rho_0^* = \rho_0 + \rho_\omega < 1$, and the following relations hold for $i = 0, 1, 2, \ldots$:

$$|||E_{i+1}||| \le \rho_0^*|||E_i||| < |||E_i||| \le r_\omega^*. \tag{3.65}$$

We shall achieve this by induction. We start with the verification of this for $i = 0$. To do so, we shall first derive an error propagation equation. We know from (3.52) ($i = 0$) that

$$F(x_1) - F(x) = Be_0^y + \varepsilon_0. \tag{3.66}$$

But by the assumption of Theorem 3.2 on the initial guess $\{x_0, y_0\}$ and the definition of r_ω^*, we know $|||E_0||| \le r_\omega^* = c_0\hat{r}_\omega/(1+\sqrt{\delta_0})$ with $\hat{r}_\omega = \min\{r_\omega, r_\gamma\}$, so $x_1 \in S_A(x, \hat{r}_\omega)$ by Lemma 3.5, which implies $x_1 \in S_A(x, r_\gamma) \cap S_A(x, r_\omega)$. This enables us to apply Lemma 3.8(i) and Algorithm 3.1 to write

$$y_1 = y_0 + Q_0^{-1}B^t(x_1 - x). \tag{3.67}$$

With (3.66), we can further deduce

$$\begin{aligned} &A^{\frac{1}{2}}(x_1 - x) \\ &\quad = A^{-\frac{1}{2}}(F(x_1) - F(x)) - A^{-\frac{1}{2}}[F(x_1) - F(x) - A(x_1 - x)] \\ &\quad = A^{-\frac{1}{2}}(Be_0^y + \varepsilon_0) - \varphi_1, \end{aligned} \tag{3.68}$$

where $\varphi_1 = A^{-\frac{1}{2}}[F(x_1) - F(x) - A(x_1 - x)]$. Setting $N_0 = A^{-\frac{1}{2}}BQ_0^{-1}B^tA^{-\frac{1}{2}}$, we obtain from (3.67) and (3.68) that

$$\begin{aligned} A^{-\frac{1}{2}}Be_1^y &= A^{-\frac{1}{2}}Be_0^y - N_0[A^{-\frac{1}{2}}(Be_0^y + \varepsilon_0) - \varphi_1] \\ &= (I - N_0)A^{-\frac{1}{2}}Be_0^y - N_0A^{-\frac{1}{2}}\varepsilon_0 + N_0\varphi_1. \end{aligned} \tag{3.69}$$

Multiplying (3.66) by $A^{-\frac{1}{2}}$, we have

$$A^{-\frac{1}{2}}(F(x) - F(x_1)) = -A^{-\frac{1}{2}}Be_0^y - A^{-\frac{1}{2}}\varepsilon_0. \tag{3.70}$$

Then using the fact that

$$f_1 = f - F(x_1) - By_1 = F(x) - F(x_1) + Be_1^y,$$

we derive from (3.69) and (3.70) that

$$A^{-\frac{1}{2}}f_1 = -(I + N_0)A^{-\frac{1}{2}}\varepsilon_0 - N_0A^{-\frac{1}{2}}Be_0^y + N_0\varphi_1. \tag{3.71}$$

Now by defining for $k = 0, 1, 2, \dots,$

$$E_k^y = A^{-\frac{1}{2}}Be_k^y, \quad E_k^{xy} = \sqrt{\delta_0}A^{-\frac{1}{2}}f_k, \quad e^{\varepsilon_k} = \sqrt{\delta_0}^{-1}A^{-\frac{1}{2}}\varepsilon_k,$$

we come to the following propagation equation using (3.69) and multiplying (3.71) by $\sqrt{\delta_0}$:

$$\begin{pmatrix} E_1^{xy} \\ E_1^y \end{pmatrix} = \mathcal{F}(N_0)\begin{pmatrix} e^{\varepsilon_0} \\ E_0^y \end{pmatrix} + \begin{pmatrix} \sqrt{\delta_0}N_0\,\varphi_1 \\ N_0\,\varphi_1 \end{pmatrix}. \tag{3.72}$$

We know from Lemma 3.8(ii) that all the positive eigenvalues of the matrix N_0 lie in the interval $[1 - \frac{1+\beta_0}{2}, 1]$; thus $\|\mathcal{F}(N_0)\| \le \rho_0$ by Lemma 3.9. Then with the assumption of Theorem 3.2 on the tolerance ε_i for $i = 0, 1, 2, \dots,$ we know (3.18) is satisfied, leading to

$$\|e^{\varepsilon_0}\| \le \|\sqrt{\delta_0}A^{-\frac{1}{2}}f_0\| = \|E_0^{xy}\|.$$

By this, with the definition of the norm $|||\cdot|||$, we see

$$\|e^{\varepsilon_0}\|^2 + \|E_0^y\|^2 \le |||E_0|||^2.$$

Using this and the bound $\|\mathcal{F}(N_0)\| \le \rho_0$, we derive from (3.72) that

$$|||E_1||| \le \rho_0\,|||E_0||| + \sqrt{1+\delta_0}\,\|\varphi_1\|. \tag{3.73}$$

Noting $x_1 \in S_A(x, r_\omega)$, it follows from (3.23), (3.53), and the definition of ρ_ω in (3.27) that

$$\begin{aligned}\sqrt{1+\delta_0}\,\|\varphi_1\| &= \sqrt{1+\delta_0}\,\|F(x_1) - F(x) - A(x_1 - x)\|_{A^{-1}} \\ &\le \omega\sqrt{1+\delta_0}\,\|x_1 - x\|_A \\ &\le \frac{\omega\sqrt{1+\delta_0}(1+\sqrt{\delta_0})}{c_0}|||E_0||| = \rho_\omega|||E_0|||.\end{aligned}$$

Then we know from (3.73) that

$$|||E_1||| \le (\rho_0 + \rho_\omega)|||E_0||| = \rho_0^*|||E_0|||.$$

Noting the fact that ω is taken from the range $(0, \omega_0)$, we have

$$\rho_\omega = \frac{\omega\sqrt{1+\delta_0}(1+\sqrt{\delta_0})}{c_0} < \frac{\omega_0\sqrt{1+\delta_0}(1+\sqrt{\delta_0})}{c_0}.$$

This, with the definition of ω_0 in (3.27), shows $\rho_\omega < 1 - \rho_0$, so $\rho_0^* = \rho_\omega + \rho_0 < 1$, and

$$|||E_1||| \le \rho_0^* |||E_0||| < |||E_0||| \le r_\omega^*,$$

which verifies (3.65) for $i = 0$.

Now we assume (3.65) holds for $i = k-1$ with any integer $k > 1$; then in exactly the same manner as for deriving the error propagation equation (3.72), we have

$$\begin{pmatrix} E_{k+1}^{xy} \\ E_{k+1}^{y} \end{pmatrix} = \mathcal{F}(N_k) \begin{pmatrix} e^{\varepsilon_k} \\ E_k^y \end{pmatrix} + \begin{pmatrix} \sqrt{\delta_0} N_k\, \varphi_{k+1} \\ N_k\, \varphi_{k+1} \end{pmatrix} \tag{3.74}$$

where $\varphi_{k+1} = A^{-\frac{1}{2}}[F(x_{k+1}) - F(x) - A(x_{k+1} - x)]$ and $N_k = A^{-\frac{1}{2}} B Q_k^{-1} B^t A^{-\frac{1}{2}}$. With this relation, one can follow exactly the same proof as for $i = 0$ above to verify that (3.65) holds for $i = k$. This completes the proof of (3.65) by induction. $\square$

4. Numerical experiments. In this section, we shall apply two new algorithms proposed in sections 2 and 3, Algorithms 2.2 and 3.1, and some other existing algorithms, to solve a linear saddle-point problem arising from a domain decomposition method with a Lagrange multiplier and a nonlinear saddle-point problem.

4.1. A linear saddle-point problem arising from a domain decomposition method with a Lagrange multiplier. Domain decomposition methods with Lagrange multipliers have become popular in solving second order elliptic problems; see, for example, [8], [27], [34], and the references therein. This method allows nonmatching grids to be used in different subdomains, with Lagrange multipliers introduced to preserve necessary interface continuities between local solutions from neighboring subdomains. A domain decomposition method with a Lagrange multiplier results in a saddle-point system with respect to the primal variable and the Lagrange multiplier. Two different approaches are often used to solve the resulting saddle-point system: the first one eliminates the primal variable in the system and forms an interface equation for the multiplier, then solves the interface equation by a PCG method [26]; the second directly solves the saddle-point system by some preconditioned iterative method [27], [34]. We shall compare the efficiency of these two different approaches.

Consider the model elliptic problem

$$-\nabla \cdot (a \nabla u) = f \quad \text{in} \quad \Omega; \qquad u = g \quad \text{on} \quad \partial\Omega, \tag{4.1}$$

where Ω is a three-dimensional rectangular domain $\Omega = [0,2] \times [0,1]^2$. We decompose Ω into two subdomains Ω_1 and Ω_2: $\Omega_1 = [0,1]^3$, $\Omega_2 = [1,2] \times [0,1]^2$, and then triangulate each subdomain Ω_k $(k = 1,2)$ into smaller cubic elements, each with edges of equal length h_k. We remark that the two triangulations in Ω_1 and Ω_2, denoted $\mathcal{T}^{h_1}$ and $\mathcal{T}^{h_2}$, respectively, are not required to match on the interface $\Gamma = \overline{\Omega}_1 \cap \overline{\Omega}_2$. By $\mathcal{N}_{h_k}$ we denote the set of vertices of all elements in the triangulation of Ω_k, and $\Gamma_{h_k} = \Gamma \cap \mathcal{N}_{h_k}$ for $k = 1,2$.

On each Ω_k, we define $V^h(\Omega_k) \subset H^1(\Omega_k)$ to be the standard Q_1 finite element space [17], [24], associated with the triangulation $\mathcal{T}^{h_k}$, and

$$V_g^h(\Omega_k) = \Big\{ v \in V^h(\Omega_k); \quad v(x_i) = g(x_i) \quad \forall\, x_i \in \mathcal{N}_{h_k} \cap (\partial\Omega_k \backslash \Gamma) \Big\},$$

$$V_g^h(\Omega) = \Big\{ v = \{v_1,\, v_2\} \in V_g^h(\Omega_1) \times V_g^h(\Omega_2); \quad v_1(x_i) = v_2(x_i) \quad \forall\, x_i \in \Gamma_{h_1} \Big\}.$$

Now the finite element approximation of the elliptic problem (4.1) can be formulated as follows: Find $\{u_{h_1}, u_{h_2}\} \in V_g^h(\Omega)$ such that

$$\sum_{k=1}^{2}(a\nabla u_{h_k}, \nabla v_k)_{\Omega_k} = \sum_{k=1}^{2}(f, v_k)_{\Omega_k} \quad \forall\, v = \{v_1, v_2\} \in V_0^h(\Omega). \tag{4.2}$$

By introducing a discrete Lagrange multiplier χ to remove the constraints on the interface as required in the finite element space $V_g^h(\Omega)$, system (4.2) can be written as the algebraic saddle-point system [27]

$$\begin{pmatrix} A_1 & 0 & B_1 \\ 0 & A_2 & B_2 \\ B_1^t & B_2^t & 0 \end{pmatrix} \begin{pmatrix} U_1 \\ U_2 \\ \chi \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ d \end{pmatrix}, \tag{4.3}$$

where A_1 and A_2 are the stiffness matrices associated with the bilinear form $(a\nabla\cdot, \nabla\cdot)_{\Omega_k}$ under the nodal basis of $V_0^{h_k}(\Omega_k)$, and U_k corresponds to the nodal values of u_{h_k} in $\Omega_k \cup \Gamma_{h_k}$.

By eliminating the variables U_1 and U_2 in (4.3), we obtain the interface equation [26]

$$K\chi = b \tag{4.4}$$

with

$$K = \sum_{k=1}^{2} B_k^t A_k^{-1} B_k, \quad b = \sum_{k=1}^{2} B_k^t A_k^{-1} b_k - d.$$

We note that the Schur complement K is a dense matrix, possibly of large size. The direct solver for system (4.4) is very expensive. Instead, we will consider the following three iterative methods for solving (4.3).

M1. Solve the interface equation (4.4) by the CG method. Recall that we are mainly interested in the case where no good preconditioners are available for the Schur complement, so CG is used here instead of PCG, although some effective preconditioners are available for the current Schur complement; see, e.g., [9], [21]. In fact, this main interest prompts us to choose the worst preconditioner, the identity, for the Schur complement K involved in all three algorithms we shall test.

M2. Solve the saddle-point system (4.3) directly by the well-known preconditioned MINRES method [39]. We will take an (algebraic) multigrid preconditioner (cf. [42]) to be the preconditioner $\hat{A}$ for the first 2×2 block matrix in the coefficient matrix of (4.3) and the identity matrix to be the preconditioner $\hat{K}$ for the Schur complement K.

M3. Solve the saddle-point system (4.3) directly by Algorithm 2.2 of section 2. The preconditioners $\hat{A}$ and $\hat{K}$ are taken to be the same as in **M2**. The approximation $\Psi_A(\phi)$ is taken to be $\hat{A}^{-1}\phi$ for any ϕ, while the approximation $\Psi_H(g_i)$ is generated by two PCG iterations for solving (2.13). (Note that PCG is the same as CG here since $\hat{K}$ is taken to be the identity.)

Table 4.1 shows the numerical results with M1, M2, and M3, where the coefficient $a(x, y, z)$ in (4.1) is taken to be $a(x, y, z) = 1+xyz$, and functions f and g are taken so that the exact solution of (4.1) is $u(x, y, z) = (x+y+z)^{\frac{1}{5}}$. Considering the singularity

TABLE 4.1
Number of iterations and CPU time (in seconds).

h_1	M1		M2		M3 ($\theta = \frac{1}{2}$)		M3 ($\theta = 0.7$)	
	iter	CPU	iter	CPU	iter	CPU	iter	CPU
1/8	10	0.14	20	0.53	10	0.52	8	0.43
1/16	15	16.6	33	8.5	11	6.6	10	6.0
1/32	21	1956.9	58	158.5	13	89.9	12	86.3

of the solution at the origin, we take $h_1 = h_2/2$. In all the experiments, the initial guesses for the three iterative methods are taken to be zero, and the outer iterations terminate when the relative error of the residual reaches 10^{-5}. We mentioned in Remark 2.3 that the coefficient 1/2 in (2.18) can be replaced by a larger number than 1/2, which is now tested in Table 4.1 with two different choices of this coefficient, denoted by a parameter θ.

We remark that the actions of the Schur complement K involve the actions of the local solvers A_k^{-1}. These actions must be very accurate, since our target is to solve system (4.3) or (4.4). One may use both direct solvers or iterative solvers to realize the actions of these local solvers. But if iterative solvers are used, the stopping criterion should be up to a very high accuracy. So we have chosen in our experiments to realize these local solvers by the standard direct solver, a banded sparse version of the Gauss elimination.

From Table 4.1 we see that the new method (M3), Algorithm 2.2, outperforms M1 and M2 essentially, and this appears to be more evident when the discrete system becomes larger.

4.2. An algebraic nonlinear saddle-point problem. We now consider an algebraic nonlinear saddle-point problem and compare the convergence of the new Algorithm 3.1 of section 3 and the well-known augmented-Uzawa method, which has been widely used for nonlinear saddle-point systems.

Let I_m be the $m \times m$ identity matrix, and let T_m an $m \times m$ matrix with entries given by

$$t_{ij} = \begin{cases} 1 & \text{if } |i-j| = 1; \\ 0 & \text{otherwise.} \end{cases}$$

For $n = 2m$, we define an $n \times n$ symmetric positive definite matrix M and an $n \times m$ matrix B with full rank as follows:

$$M = \begin{pmatrix} \frac{5}{2}I_m - \frac{1}{4}T_m & -I_m \\ -I_m & \frac{5}{2}I_m - \frac{1}{4}T_m \end{pmatrix}, \quad B = (0, \quad 2I_m - T_m)^t.$$

The smallest and largest eigenvalues of M are given by [14]

$$\lambda_1 = 4\sin^2 \frac{m}{2(n+m)}\pi + \sin^2 \frac{1}{2(m+1)}\pi = 1 + \sin^2 \frac{1}{2(m+1)}\pi,$$
$$\lambda_n = 4\sin^2 \frac{n}{2(n+m)}\pi + \sin^2 \frac{m}{2(m+1)}\pi = 3 + \sin^2 \frac{m}{2(m+1)}\pi.$$

Now we define the nonlinear mapping F as

$$(4.5)\ F(\xi) = M\xi + \frac{1}{5}\left(\frac{\xi_1}{1+\xi_1^2}, \frac{\xi_2}{1+\xi_2^2}, \dots, \frac{\xi_n}{1+\xi_n^2}\right)^t \quad \forall\, \xi = (\xi_1\ \xi_2 \cdots \xi_n)^t \in R^n.$$

One can verify that F is strongly monotone and Lipschitz continuous. Moreover, we have

$$\nabla F(\xi) = M + \frac{1}{5}\text{diag}\left(\frac{1-\xi_1^2}{(1+\xi_1^2)^2}, \frac{1-\xi_2^2}{(1+\xi_2^2)^2}, \dots, \frac{1-\xi_n^2}{(1+\xi_n^2)^2}\right),$$

which implies

$$\frac{4}{5}\|\zeta\|^2 \le (\nabla F(\xi)\zeta, \zeta) \le \frac{21}{5}\|\zeta\|^2 \quad \forall \zeta,\ \xi \in R^n. \tag{4.6}$$

Let $A_i = \nabla F(x_i)$. If we choose $\hat{A}_i = I_n$, we have $\text{cond}(\hat{A}_i^{-1}A_i) \le 21/4$. Noting that the Schur complement $K_i = B^t A_i^{-1} B$ is a dense matrix without any special structure, it is difficult to find a reasonable preconditioner $\hat{K}_i$ for K_i. Therefore we will take the worst preconditioner $\hat{K}_i = I_m$.

The functional $J(\xi)$ satisfying $\nabla J(\xi) = F(\xi)$ can be written as

$$J(\xi) = \frac{1}{2}(M\xi, \xi) + \frac{1}{10}\sum_{l=1}^{n} \ln(1+\xi_l^2). \tag{4.7}$$

One can see that $J(\xi)$ is uniformly convex. This enables us to realize the nonlinear inner iteration in Algorithm 3.1 for finding x_{i+1} by Algorithm 3.2.

The right-hand side functions f and g in (1.1) are generated using system (1.1) when the exact solution is taken to be

$$x = (1, 1, \dots, 1)^t, \quad y = \left(1, \frac{1}{2}, \dots, \frac{1}{m}\right)^t.$$

We will compare the new nonlinear inexact Uzawa algorithm (Algorithm 3.1) with the well-known augmented-Uzawa method (see [38, pp. 234–244]). The augmented-Uzawa method converges, provided that the inner iteration involved is accurate enough, and the augmented parameter r is taken to be sufficiently large (or the initial guess is very close to the exact solution).

The initial guess x_0 for both Algorithm 3.1 and the augmented-Uzawa method will be taken to be the approximation generated by three steps of the steepest descent method for solving the perturbation system (3.30) with $\mu = 1/10$ and the zero initial guess. With x_0 available, y_0 is then determined from the second equation of (3.29). The iterations of Algorithm 3.1 and the augmented-Uzawa method terminate when

$$\varepsilon = \left\{\frac{\|f - F(x_i) - By_i\|^2 + \|g - B^t x_i\|^2}{\|f\|^2 + \|g\|^2}\right\}^{\frac{1}{2}} \le 10^{-5}. \tag{4.8}$$

We first apply Algorithm 3.1 for the nonlinear saddle-point problem (1.1) with the data described as above. There are three stopping parameters δ_0, δ_g, and γ involved in the inner iterations. For the convenience of numerical tests, we will replace the norms used in (3.18), (3.19), and (3.20) by the l^2-norms of the relative errors of the residuals, and take $\delta_0 = 1/4$ and $\gamma = 1/4$, but a few different choices for δ_g. The convergence results of Algorithm 3.1 are summarized in Table 4.2.

We then apply the augmented-Uzawa method for the nonlinear saddle-point problem (1.1) with the data described as above. Each nonlinear inner iteration involved in the augmented-Uzawa algorithm is realized by 30 or 40 iteration of Algorithm 3.2—a

Table 4.2
Number of iterations and CPU time (in seconds) with Algorithm 3.1.

$(n,\ m)$	(100, 50)		(200, 100)		(400, 200)		(800, 400)	
	iter	CPU	iter	CPU	iter	CPU	iter	CPU
$\delta_g = 1/4$	32	2.1	30	9.5	30	104.9	28	676.6
$\delta_g = 1/6$	29	2.2	31	14.4	29	87.3	27	635.6
$\delta_g = 1/8$	29	1.8	28	11.0	28	77.4	27	633.8

Table 4.3
*Number of iterations and CPU time with augmented-Uzawa algorithm (*30 *inner iterations).*

$(n,\ m)$	(100, 50)		(200, 100)		(400, 200)		(800, 400)	
	iter	CPU	iter	CPU	iter	CPU	iter	CPU
$r = 10$	255	85.5	180	267.8	139	987.7	106	7375.1
$r = 50$	62	20.8	49	73.1	40	284.9	31	2160.8
$r = 100$	64	21.6	61	90.8	51	361.8	40	2786.6

Table 4.4
*Number of iterations and CPU time with augmented-Uzawa algorithm (*40 *inner iteration).*

$(n,\ m)$	(100, 50)		(200, 100)		(400, 200)		(800, 400)	
	iter	CPU	iter	CPU	iter	CPU	iter	CPU
$r = 10$	255	115.5	179	388.1	138	1311.1	106	10553.7
$r = 50$	53	24.1	38	82.6	28	266.7	22	2195.9
$r = 100$	44	19.9	41	89.0	30	285.6	24	2406.2

Table 4.5
Number of iterations and CPU time with augmented-Uzawa algorithm ($\tilde{\varepsilon} \leq 10^{-2}$).

$(n,\ m)$	(100, 50)		(200, 100)		(400, 200)		(800, 400)	
	iter	CPU	iter	CPU	iter	CPU	iter	CPU
$r = 10$	255	97.0	179	354.8	138	1342.1	106	10269.6
$r = 50$	52	62.5	37	198.7	28	880.8	22	6249.0
$r = 100$	27	34.8	18	114.7	13	331.9	10	2356.4

preconditioned CG-type method. The numerical results are summarized in Table 4.3 (30 inner iterations) and Table 4.4 (40 inner iterations).

In order to better understand the effect of the inner iterations, we have also implemented stopping the inner iterations by the standard stopping criterion, which is set to stop the inner iterations when the relative residual is less than $\tilde{\varepsilon}$. The numerical results are summarized in Table 4.5.

Tables 4.3–4.5 indicate that both the convergence rate and the CPU time of the augmented-Uzawa algorithm depend strongly on the augmented parameter r. But there is still no theory about the selection of a reasonable or optimal parameter r.

One can see from Tables 4.2–4.5 that Algorithm 3.1 is evidently more efficient than the augmented-Uzawa method (even if the optimal parameter r may be found). For the augmented-Uzawa method, one can clearly see that it converges faster with more inner iterations, which, however, does not necessarily lead to less CPU times; see the figures in Table 4.3 and 4.4 with $r = 50$. When the inner iterations are set to be too accurate, the total CPU time may be much longer; compare the figures in Tables 4.4 and 4.5 with $r = 50$. But how to set the stopping criterion for the inner iterations of the augmented-Uzawa method is rather difficult and often quite problem dependent.

Acknowledgments. The authors wish to thank two anonymous referees for many constructive comments which led to great improvement in the results and the presentation of the paper.

REFERENCES

[1] M. Al-Baali and R. Fletcher, *On the order of convergence of preconditioned nonlinear conjugate gradient methods*, SIAM J. Sci. Comput., 17 (1996), pp. 658–665.

[2] K. Arrow, L. Hurwicz, and H. Uzawa, *Studies in Linear and Nonlinear Programming*, Stanford University Press, Stanford, CA, 1958.

[3] O. Axelsson, *Numerical algorithms for indefinite problems*, in Elliptic Problem Solvers, Academic Press, New York, 1984, pp. 219–232.

[4] R. Bank, B. Welfert, and H. Yserentant, *A class of iterative methods for solving saddle point problems*, Numer. Math., 56 (1990), pp. 645–666.

[5] A. Battermann and M. Heinkenschloss, *Preconditioners for Karush–Kuhn–Tucker matrices arising in the optimal control of distributed systems*, in Control and Estimation of Distributed Parameter Systems (Vorau, 1996), W. Desch, F. Kappel, and K. Kunisch, eds., Birkhäuser, Basel, 1998, pp. 15–32.

[6] A. Battermann and E. W. Sachs, *Block preconditioner for KKT systems in PDE-governed optimal control problems*, in Fast Solution of Discretized Optimization Problems, R. H. Hoppe, K.-H. Hoffmann, and V. Schulz, eds., Birkhäuser, Basel, 2001, pp. 1–18.

[7] A. Battermann and E. W. Sachs, *An Indefinite Preconditioner for KKT Systems Arising in Optimal Control Problems*, Technical Report, University Trier, Trier, Germany, 2004.

[8] F. Belgacem, *The mortar finite element method with Lagrange multipliers*, Numer. Math., 84 (1999), pp. 173–197.

[9] M. Bhardwaj, D. Day, C. Farhat, M. Lesoinne, K. Pierson, and D. Rixen, *Application of the FETI method to ASCI problems: Scalability results on one thousand processors and discussion of highly heterogeneous problems*, Internat. J. Numer. Methods Engrg., 47 (2000), pp. 513–535.

[10] J. H. Bramble, J. E. Pasciak, and A. T. Vassilev, *Analysis of the inexact Uzawa algorithm for saddle point problems*, SIAM J. Numer. Anal., 34 (1997), pp. 1072–1092.

[11] J. Bramble and J. Pasciak, *A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems*, Math. Comp., 50 (1988), pp. 1–18.

[12] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer-Verlag, New York, 1991.

[13] K. H. Chan, K. Zhang, J. Zou, and G. Schubert, *A non-linear, 3-D spherical α^2 dynamo using a finite element method*, Phys. Earth Planetary Int., 128 (2001), pp. 35–50.

[14] X. Chen, *Global and superlinear convergence of inexact Uzawa methods for saddle point problems with nondifferentiable mappings*, SIAM J. Numer. Anal., 35 (1998), pp. 1130–1148.

[15] Z. Chen, Q. Du, and J. Zou, *Finite element methods with matching and nonmatching meshes for Maxwell equations with discontinuous coefficients*, SIAM J. Numer. Anal., 37 (2000), pp. 1542–1570.

[16] Z. Chen and J. Zou, *An augmented Lagrangian method for identifying discontinuous parameters in elliptic systems*, SIAM J. Control Optim., 37 (1999), pp. 892–910.

[17] P. Ciarlet, *Basic error estimates for elliptic problems*, in Handbook of Numerical Analysis, Volume II, P. Ciarlet and J.-L. Lions, eds., North-Holland, Amsterdam, 1991, pp. 17–351.

[18] F. H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley, New York, 1983.

[19] Y. Dai and Y. Yuan, *Nonlinear Conjugate Gradient Methods*, Shanghai Scientific and Technical Publishers, China, 2000 (in Chinese).

[20] H. C. Elman and G. H. Golub, *Inexact and preconditioned Uzawa algorithms for saddle point problems*, SIAM J. Numer. Anal., 31 (1994), pp. 1645–1661.

[21] C. Farhat and F.-X. Roux, *Implicit parallel processing in structural mechanics*, Comput. Mech. Adv., 2 (1994), pp. 1–124.

[22] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., John Wiley, Chichester, 1987.

[23] P. E. Gill, W. Murray, D. B. Ponceleón, and M. A. Saunders, *Preconditioners for indefinite systems arising in optimization*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 292–311.

[24] V. Girault and P.-A. Raviart, *Finite Element Methods for Navier–Stokes Equations*, Springer-Verlag, Berlin, 1986.

[25] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, SIAM, Philadelphia, 1989.

[26] Q. Hu, G. Liang, and J. Liu, *A preconditioner for three dimensional domain decomposition methods with Lagrange multipliers*, J. Syst. Sci. Complex., 16 (2003), pp. 513–526.

[27] Q. Hu, Z. Shi, and D. Yu, *Efficient solvers for saddle-point problems arising from domain decompositions with Lagrange multipliers*, SIAM J. Numer. Anal., 42 (2004), pp. 905–933.

[28] Q. Hu and J. Zou, *An iterative method with variable relaxation parameters for saddle-point problems*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 317–338.

[29] Q. Hu and J. Zou, *Two new variants of nonlinear inexact Uzawa algorithms for saddle-point problems*, Numer. Math., 93 (2002), pp. 333–359.

[30] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, 1995.

[31] C. T. Kelley, *Iterative Methods for Optimization*, SIAM, Philadelphia, 1999.

[32] C. Keller, N. I. M. Gould, and A. J. Wathen, *Constraint preconditioning for indefinite linear systems*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1300–1317.

[33] Y. L. Keung and J. Zou, *An efficient linear solver for nonlinear parameter identification problems*, SIAM J. Sci. Comput., 22 (2000), pp. 1511–1526.

[34] A. Klawonn and O. B. Widlund, *A domain decomposition method with Lagrange multiplier and inexact solvers for linear elasticity*, SIAM J. Sci. Comput., 22 (2000), pp. 1199–1219.

[35] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, 1999.

[36] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Math. Programming, 58 (1993), pp. 353–368.

[37] W. Queck, *The convergence factor of preconditioned algorithms of the Arrow–Hurwicz type*, SIAM J. Numer. Anal., 26 (1989), pp. 1016–1030.

[38] B. T. Polyak, *Introduction to Optimization*, Optimization Software, New York, 1987.

[39] T. Rusten and R. Winther, *A preconditioned iterative method for saddlepoint problems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 887–904.

[40] J. Stoer and Y. Yuan, *A subspace study on conjugate algorithms*, Z. Angew. Math. Mech., 75 (1995), pp. 69–77.

[41] P. Tseng, *Applications of a splitting algorithm to decomposition in convex programming and variational inequalities*, SIAM J. Control Optim., 29 (1991), pp. 119–138.

[42] J. M. P. Vanek and M. Brezina, *Convergence of algebraic multigrid based on smoothed aggregation*, Numer. Math., 88 (2001), pp. 559–579.