Numer. Math. (2002) 93: 333–359
Digital Object Identifier (DOI) 10.1007/s002110100386

Numerische
Mathematik

Two new variants of nonlinear inexact Uzawa algorithms for saddle-point problems

Qiya Hu[1,⋆], **Jun Zou**[2,⋆⋆]

[1] Institute of Computational Mathematics and Scientific Engineering Computing,
Chinese Academy of Sciences, Beijing 100080, People's Republic of China;
e-mail: hqy@lsec.cc.ac.cn

[2] Department of Mathematics, The Chinese University of Hong Kong, Shatin, N.T.,
Hong Kong, People's Republic of China; e-mail: zou@math.cuhk.edu.hk

Received December 8, 1999 / Revised version received September 8, 2001 /
Published online March 8, 2002 – © Springer-Verlag 2002

Summary. In this paper, we consider some nonlinear inexact Uzawa methods for iteratively solving linear saddle-point problems. By means of a new technique, we first give an essential improvement on the convergence results of Bramble-Paschiak-Vassilev for a known nonlinear inexact Uzawa algorithm. Then we propose two new algorithms, which can be viewed as a combination of the known nonlinear inexact Uzawa method with the classical steepest descent method and conjugate gradient method respectively. The two new algorithms converge under very practical conditions and do not require any apriori estimates on the minimal and maximal eigenvalues of the preconditioned systems involved, including the preconditioned Schur complement. Numerical results of the algorithms applied for the Stokes problem and a purely linear system of algebraic equations are presented to show the efficiency of the algorithms.

Mathematics Subject Classification: 65F10, 65N20

1 Introduction

The purpose of this paper is to study some nonlinear inexact Uzawa algorithms for solving the following system of linear equations

[⋆] The work of this author was partially supported by a grant from The Institute of Mathematical Sciences, CUHK

[⋆⋆] The work of this author was partially supported by Hong Kong RGC Grants CUHK 4292/00P and CUHK 4244/01P

$$\begin{pmatrix} A & B \\ B^{\mathrm{t}} & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \tag{1.1}$$

where A is a symmetric positive definite $n \times n$ matrix, and B is an $n \times m$ matrix ($m \leq n$). We assume that the $(n+m) \times (n+m)$ coefficient matrix

$$M = \begin{pmatrix} A & B \\ B^{\mathrm{t}} & 0 \end{pmatrix}$$

is nonsingular, which is equivalent to the positive definiteness of the Schur complement matrix

$$C = B^{\mathrm{t}} A^{-1} B \,. \tag{1.2}$$

Linear systems such as (1.1) are called saddle-point problems, which may arise from finite element discretizations of Stokes equations and Maxwell equations [7,11,15], mixed finite element formulations for second order elliptic problems [2,7] or from Lagrange multiplier methods for optimization problems [1,16], for the parameter identification and domain decomposition problems [12,17] and [19].

In recent years, there is a rapidly increasing literature which is concerned with preconditioned iterative methods for solving the indefinite system of equations like (1.1), see [3,5,6,8,14,20–22]. In particular, inexact Uzawa-type algorithms have attracted wide attention, see [3,5,8,14,21] and the references therein. The common nice properties of these Uzawa-type algorithms are that they have minimal memory requirements, and are easy to implement.

Let $\hat{A}$ and $\hat{C}$ be two positive definite matrices, which are assumed to be the preconditioners of the matrices A and $C = B^{\mathrm{t}} A^{-1} B$ respectively. And let R^l denote the usual l-dimensional Euclidean space. For any $l \times l$ positive definite matrix G, we use $\|x\|_G$ to denote the G-induced norm, namely $\|x\|_G = (Gx,\ x)^{1/2}$ for all $x \in R^l$. Then the standard inexact Uzawa algorithm can be described as follows (cf. [5] and [14]):

Algorithm 1.1 *(inexact Uzawa).*
Given $x_0 \in R^n$ and $y_0 \in R^m$, the sequence $\{(x_i, y_i)\} \subset R^n \times R^m$ is defined for $i = 1, 2, \cdots$, by

$$x_{i+1} = x_i + \hat{A}^{-1}[f - (Ax_i + By_i)], \tag{1.3}$$

$$y_{i+1} = y_i + \hat{C}^{-1}(B^{\mathrm{t}} x_{i+1} - g). \tag{1.4}$$

There are some earlier versions of the above algorithm, e.g., see [3] and [21]. The existing convergence results indicate that these algorithms are convergent by assuming some good knowledge of the spectrum of the preconditioned matrices $\hat{A}^{-1}A$ and $\hat{C}^{-1}C$ or under some proper scalings of the preconditioners $\hat{A}$ and $\hat{C}$. This ”preprocessing” may not be easy to achieve in some applications.

To avoid the estimate on the minimal generalized eigenvalues of $\hat{A}$ with respect to A, a practical way is to replace the inner iteration (1.3) of the standard inexact Uzawa algorithm by some nonlinear iteration, for example, the PCG iteration. The detailed convergence properties of the resulting algorithm can be found in [5] and [14]. However, some improvements on the convergence results achieved so far seem necessary (see the numerical examples given in [5] and Sect. 2 below). Moreover, a good knowlwdge on the spectrum of the preconditioned Schur complement is needed for a proper scaling of the preconditioner to guarantee the convergence of the global inexact Uzawa iteration, see Sect. 2 for the details.

To avoid the estimate of the generalized eigenvalues of $\hat{C}$ with respect to $B^{\mathrm{t}}\hat{A}^{-1}B$, the Uzawa-type algorithm proposed in [3] introduced a PCG algorithm as an inner iteration of (1.4). It was proved that the Uzawa-type algorithm converges under the assumptions that each inner iteration teminates with a certain accuracy and the basic iteration for the preconditioned system $\hat{A}^{-1}A$ is convergent.

The preconditioned minimal residual method studied in [20] and [22] is always convergent, but needs some good knowledge of the smallest eigenvalues of the matrices $\hat{A}^{-1}A$ and $\hat{C}^{-1}(B^{\mathrm{t}}\hat{A}^{-1}B)$ to achieve a practical convergence rate. Without a good apriori estimate on the smallest eigenvalues of these two preconditioned matrices, the condition number of the (global) preconditioned system may be still very large even if the condition numbers of the matrices $\hat{A}^{-1}A$ and $\hat{C}^{-1}(B^{\mathrm{t}}\hat{A}^{-1}B)$ are small (cf. [22]). In this case, the convergence of this iterative method may be slow (see Sect. 4).

The purpose of this paper is twofold. By means of a new technique, we first give an essential improvement on the convergence results of Bramble-Paschiak-Vassilev for a known nonlinear inexact Uzawa algorithm. Then we propose two new algorithms, which can be viewed as a combination of the known nonlinear inexact Uzawa method with the classical steepest descent method and conjugate gradient method respectively. We will show our algorithms are always convergent without any assumptions on the spectra of the preconditioners $\widehat{C}$ and $\widehat{A}$.

The outline of the remainder of the paper is as follows. We give a new convergence result for a known nonlinear inexact Uzawa algorithm in Sect. 2, and present two new algorithms and analyse their convergence rates in Sects. 3 and 4. Finally, we apply the proposed new algorithms for solving the Stokes problem and a linear system of purely algebraic equations in Sect. 5.

2 An improved convergence result for the known nonlinear inexact Uzawa algorithm

The main purpose of this section is to present an essential improvement on the convergence results of Bramble-Paschiak-Vassilev for a nonlinear inexact Uzawa algorithm. In this algorithm, the computation of the action of the inverse A^{-1} in (1.3) is replaced with that of an approximation to A^{-1} which results from applying a nonlinear iterative process for inverting A. Two examples of such approximations are from defining the approximate inverse by a PCG iteration or the operator which results from the application of a multigrid cycling algorithm with a nonlinear smoother (cf. [5]).

We now describe the nonlinear inexact Uzawa algorithm, in which the nonlinear approximate inverse is defined by a map $\Psi : R^n \to R^n$. For $\phi \in R^n$, $\Psi(\phi)$ is an "approximation" to the solution ξ of

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

The following assumption was made in [5] on the accuracy of the approximation:

$$\|\Psi(\phi) - A^{-1}\phi\|_A \leq \delta \, \|A^{-1}\phi\|_A, \quad \forall \phi \in R^n \tag{2.2}$$

for some $\delta \in (0,1)$. This is a natural assumption and can be satisfied by the approximate inverse associated with the PCG iteration and the approximate inverse defined by one sweep of a multigrid algorithm with conjugate gradient smoothing (cf. [5]).

The nonlinear inexact Uzawa algorithm is defined as follows:

Algorithm 2.1 *(nonlinear inexact Uzawa).*
Given $x_0 \in R^n$ and $y_0 \in R^m$, the sequence $\{(x_i, y_i)\} \subset R^n \times R^m$ is defined for $i = 1, 2, \cdots$, by

$$x_{i+1} = x_i + \Psi(f - (Ax_i + By_i)), \tag{2.3}$$

$$y_{i+1} = y_i + \hat{C}^{-1}(B^{\mathrm{t}} x_{i+1} - g). \tag{2.4}$$

Under the assumption (2.2) and

$$(1-\gamma)(\hat{C}y, y) \leq (Cy, y) \leq (\hat{C}y, y), \quad \forall y \in R^m\,, \tag{2.5}$$

it was proved in [5] that Algorithm 2.1 converges if

$$\delta < \frac{1-\gamma}{3-\gamma}\,. \tag{2.6}$$

And in this case the following estimate of the convergence rate holds:

$$\frac{\delta}{1+\delta}\|e_i^x\|_A^2 + \|e_i^y\|_{\hat{C}}^2 \leq \rho^{2i}\left(\frac{\delta}{1+\delta}\|e_0^x\|_A^2 + \|e_0^y\|_{\hat{C}}^2\right)$$

and

$$\|\mathrm{e}_i^x\|_A^2 \le (1+\delta)\,(1+2\delta)\,\rho^{2i-2}\left(\frac{\delta}{1+\delta}\|e_0^x\|_A^2 + \|e_0^y\|_{\hat{C}}^2\right),$$

where e_i^x and e_i^y are the error functions

$$\mathrm{e}_i^x = x - x_i, \quad \mathrm{e}_i^y = y - y_i$$

and

$$\rho = \frac{2\delta + \gamma + \sqrt{(2\delta+\gamma)^2 + 4\delta(1-\gamma)}}{2}.$$

Generally speaking, the convergence condition (2.6) is somehow not very satisfactory as it depends on the “preconditioning parameter” γ. The condition says that the evaluation of $A^{-1}f_i$ for some f_i required in (2.3) by the nonlinear iteration Ψ should be more accurate if the preconditioning of the Schur complement becomes worse (i.e., γ is close to 1). This seems not so reasonable. Indeed we find surprisingly that such a requirement on δ can be made completely independent of γ, namely the same results still hold when (2.6) is replaced by $\delta < 1/3$ as shown below.

To demonstrate such an essentially improved result, we will first establish the convergence rate for the residual

$$f_i = f - (Ax_i + By_i) = A\,e_i^x + B\,e_i^y,$$

instead of the traditionally used error vector e_i^x as was done in [5]. This is an important observation which leads to our improved results. To do so, we define a new norm $|||\cdot|||$:

$$|||v||| = (\|v_1\|^2 + \|v_2\|_C^2)^{\frac{1}{2}}, \;\; v = (v_1, v_2) \in R^n \times R^m.$$

With this norm we can show

Theorem 2.1 *Under the assumptions (2.2) and (2.5), for $\delta < \frac{1}{3}$ we have*

$$|||E_{i+1}||| \le \overline{\rho}\,|||E_i|||, \quad E_i = (\sqrt{\delta}A^{-\frac{1}{2}}f_i, \mathrm{e}_i^y) \in R^n \times R^m. \tag{2.7}$$

The convergence rate $\overline{\rho}$ (< 1) can be estimated by

$$\overline{\rho} \le \begin{cases} \sqrt{\delta + \delta^2} + \delta & \text{for} \quad 0 \le \gamma \le \frac{4\delta}{1+\delta}, \\ \frac{\gamma(1+\delta) - 2\delta + \sqrt{[\gamma(1+\delta)-2\delta]^2 + 4\delta}}{2} & \text{for} \quad \frac{4\delta}{1+\delta} < \gamma < 1 \end{cases} \tag{2.8}$$

$$\le \begin{cases} \sqrt{\delta + \delta^2} + \delta & \text{for} \quad 0 \le \gamma \le \frac{4\delta}{1+\delta}, \\ 1 - \frac{1}{2}(1-\gamma)(1+\delta) & \text{for} \quad \frac{4\delta}{1+\delta} < \gamma < 1. \end{cases} \tag{2.9}$$

Proof. The inequality (2.8) was discovered and proved independently by Cheng [10] and the authors [18]. However, for reader's convenience, we outline the main idea of the proof.

Let

$$B^{\mathrm{t}} A^{-\frac{1}{2}} = U(\Sigma_0\, 0)V^{\mathrm{t}}$$

be the singular value decomposition of the matrix $B^{\mathrm{t}} A^{-\frac{1}{2}}$. Define

$$Q_0 = \Sigma_0 U^{\mathrm{t}} \hat{C}^{-1} U \Sigma_0$$

and

$$F = \begin{pmatrix} -\delta(I+Q_0) & \sqrt{\delta}Q_0 \\ \sqrt{\delta}Q_0 & (I-Q_0) \end{pmatrix}.$$

It can be verified directly that

$$\|F\| \le \max_{\beta\in\sigma(Q_0)} \frac{|1-\delta-\beta(1+\delta)| + \sqrt{|1-\delta-\beta(1+\delta)|^2+4\delta}}{2},$$

(2.10)

where $\sigma(Q_0)$ is the spectrum of the matrix Q_0. One can check that the matrix Q_0 has the same eigenvalues as the matrix $\hat{C}^{-1}C$. Thus we have $\sigma(Q_0) \subset [1-\gamma, 1]$ by (2.5). Then we can show $\overline{\rho} \le \|F\|$, and (2.8) follows from (2.10).

We next show (2.9). When $\gamma > 4\delta/(1+\delta)$, we have

$$\gamma(1+\delta) - 2\delta > 2\delta > 0.$$

This with the condition that $\gamma < 1$ leads to

$$\begin{aligned} &\gamma(1+\delta) - 2\delta + \sqrt{[\gamma(1+\delta)-2\delta]^2 + 4\delta} \\ &\quad < \gamma(1+\delta) - 2\delta + \sqrt{(1-\delta)^2+4\delta} \\ &\quad = \gamma(1+\delta) - 2\delta + 1 - \delta \\ &\quad = 2 + (\gamma-1)(1+\delta). \end{aligned}$$

(2.9) follows then from this relation and (2.8). In particular, when $\delta < \frac{1}{3}$ we have $\overline{\rho} < 1$. □

Using Theorem 2.1, we can estimate the convergence rates of the errors e_i^x and e_i^y:

Theorem 2.2 *Under the assumptions (2.2) and (2.5), Algorithm 2.1 converges for any $\delta < \frac{1}{3}$. Moreover, we have*

$$\|\mathrm{e}_i^x\|_A \le (\sqrt{1+4\delta} + \overline{\rho})\overline{\rho}^{\mathrm{i}-1}|||E_0|||, \quad i = 1, 2, \cdots \tag{2.11}$$

and

$$\|\mathrm{e}_i^y\|_C \le \overline{\rho}^i|||E_0|||, \quad i = 1, 2, \cdots. \tag{2.12}$$

Here the rate of convergence $\overline{\rho}$ has the following bounds:

$$\overline{\rho} \le \begin{cases} \rho - \frac{1}{2}\gamma - \frac{1}{6}\gamma^2 & \text{for} \quad 0 \le \gamma \le \frac{4\delta}{1+\delta}, \\ \rho - \frac{3}{2}(\delta + \delta^2) & \text{for} \quad \frac{4\delta}{1+\delta} < \gamma < 1. \end{cases} \tag{2.13}$$

Proof. The estimate (2.12) is a direct consequence of (2.7). We next prove (2.11). To do so, we start with the derivation of a useful expression for $A^{-1/2}f_i$. Clearly, we can write

$$f_i = Ae_i^x + B\mathrm{e}_i^y, \quad B^{\mathrm{t}}x_{i+1} - g = -B^{\mathrm{t}}\mathrm{e}_{i+1}^x,$$

this with (2.3) and (2.4) leads to

$$A^{\frac{1}{2}}\mathrm{e}_{i+1}^x = A^{\frac{1}{2}}(\mathrm{e}_i^x - \Psi(f_i)) = (A^{-\frac{1}{2}}f_i - A^{\frac{1}{2}}\Psi(f_i)) - A^{-\frac{1}{2}}Be_i^y$$

and

$$\begin{aligned} A^{-\frac{1}{2}}B\mathrm{e}_{i+1}^y &= A^{-\frac{1}{2}}B(\mathrm{e}_i^y + \hat{C}^{-1}B^{\mathrm{t}}\mathrm{e}_{i+1}^x) \\ &= A^{-\frac{1}{2}}B\Big[\mathrm{e}_i^y + \hat{C}^{-1}B^{\mathrm{t}}A^{-\frac{1}{2}} \\ &\quad \times\Big((A^{-\frac{1}{2}}f_i - A^{\frac{1}{2}}\Psi(f_i)) - A^{-\frac{1}{2}}Be_i^y\Big)\Big] \\ &= A^{-\frac{1}{2}}B\hat{C}^{-1}B^{\mathrm{t}}A^{-\frac{1}{2}}(A^{-\frac{1}{2}}f_i - A^{\frac{1}{2}}\Psi(f_i)) \\ &\quad +(I - A^{-\frac{1}{2}}B\hat{C}^{-1}B^{\mathrm{t}}A^{-\frac{1}{2}})A^{-\frac{1}{2}}Be_i^y. \end{aligned} \tag{2.14}$$

So we have

$$\begin{aligned} A^{-\frac{1}{2}}f_{i+1} &= A^{\frac{1}{2}}\mathrm{e}_{i+1}^x + A^{-\frac{1}{2}}B\mathrm{e}_{i+1}^y \\ &= (I + A^{-\frac{1}{2}}B\hat{C}^{-1}B^{\mathrm{t}}A^{-\frac{1}{2}})(A^{-\frac{1}{2}}f_i - A^{\frac{1}{2}}\Psi(f_i)) \\ &\quad - (A^{-\frac{1}{2}}B\hat{C}^{-1}B^{\mathrm{t}}A^{-\frac{1}{2}})A^{-\frac{1}{2}}Be_i^y. \end{aligned} \tag{2.15}$$

Because of (2.2), for any given natural number i one can construct a symmetric and positive definite $n \times n$ matrix Q_{Ai} such that

$$Q_{Ai}\Psi(f_i) = f_i$$

and

$$\left\| I - Q_{Ai}^{-\frac{1}{2}} A Q_{Ai}^{-\frac{1}{2}} \right\| \leq \delta\,,$$

see, for example, Lemma 9 of [3] for such a construction of Q_{Ai}. Thus we have

$$\Psi(f_i) = Q_{Ai}^{-1} f_i \tag{2.16}$$

and

$$\left\| I - A^{\frac{1}{2}} Q_{Ai}^{-1} A^{\frac{1}{2}} \right\| \leq \delta. \tag{2.17}$$

By means of (2.14)-(2.16), we can further write

$$\begin{aligned} A^{-\frac{1}{2}} f_{i+1} = (I + A^{-\frac{1}{2}} B \hat{C}^{-1} B^{\mathrm{t}} A^{-\frac{1}{2}})(I - A^{\frac{1}{2}} Q_{Ai}^{-1} A^{\frac{1}{2}}) A^{-\frac{1}{2}} f_i \\ - (A^{-\frac{1}{2}} B \hat{C}^{-1} B^{\mathrm{t}} A^{-\frac{1}{2}}) A^{-\frac{1}{2}} B e_i^y\,, \end{aligned}$$

or equivalently write (replacing i by $i-1$)

$$A^{-\frac{1}{2}} f_i = (I + \widetilde{Q}_2) \widetilde{Q}_{1i} A^{-\frac{1}{2}} f_{i-1} - \widetilde{Q}_2 A^{-\frac{1}{2}} B e_{i-1}^y$$

where

$$\widetilde{Q}_{1i} = I - A^{\frac{1}{2}} Q_{Ai}^{-1} A^{\frac{1}{2}} \quad \text{and} \quad \widetilde{Q}_2 = A^{-\frac{1}{2}} B \hat{C}^{-1} B^{\mathrm{t}} A^{-\frac{1}{2}}.$$

Applying the arithmetic-geometric mean inequality yields

$$\begin{aligned} \|A^{-\frac{1}{2}} f_i\|^2 \leq (1+\eta) \|(I + \widetilde{Q}_2) \widetilde{Q}_{1i} A^{-\frac{1}{2}} f_{i-1}\|^2 \\ +(1+\eta^{-1}) \|\widetilde{Q}_2 A^{-\frac{1}{2}} B e_{i-1}^y\|^2, \end{aligned} \tag{2.18}$$

where η is any positive number. One can easily see that the matrix $\widetilde{Q}_2$ has the same positive eigenvalues as the matrix $\hat{C}^{-1}C$. Hence, the assumption (2.6) implies that the eigenvalues of $\widetilde{Q}_2$ are in the interval $[0, 1]$. Namely, we have

$$\|\widetilde{Q}_2\| \leq 1, \quad \|I + \widetilde{Q}_2\| \leq 2,$$

this, with (2.18) and (2.17), leads to

$$\begin{aligned} \|A^{-\frac{1}{2}} f_i\|^2 &\leq (1+\eta) 4\delta^2 \|A^{-\frac{1}{2}} f_{i-1}\|^2 + (1+\eta^{-1}) \|A^{-\frac{1}{2}} B \mathrm{e}_{i-1}^y\|^2 \\ &= 4\delta(1+\eta) \|\sqrt{\delta} A^{-\frac{1}{2}} f_{i-1}\|^2 + (1+\eta^{-1}) \|\mathrm{e}_{i-1}^y\|_C^2. \end{aligned}$$

Taking $\eta = 1/(4\delta)$ above and using (2.7) we obtain

$$\|A^{-\frac{1}{2}} f_i\| \leq \sqrt{1+4\delta}\,\overline{\rho}^{\mathrm{i}-1} |||E_0|||.$$

Now using the relation

$$A^{\frac{1}{2}} \mathrm{e}_i^x = A^{-\frac{1}{2}} f_i - A^{-\frac{1}{2}} B \mathrm{e}_i^y,$$

(2.11) follows immediately from (2.7) and the triangle inequality.

It remains to show (2.13). Let us first consider $0 < \gamma \leq 4\delta/(1+\delta))$. We have

$$\begin{aligned} \rho - \overline{\rho} &= \delta + \frac{\gamma}{2} + \frac{\sqrt{4\delta^2 + \gamma^2 + 4\delta}}{2} - (\sqrt{\delta^2 + \delta} + \delta) \\ &= \frac{\gamma}{2} + \frac{\sqrt{4(\delta^2 + \delta) + \gamma^2} - \sqrt{4(\delta^2 + \delta)}}{2} \\ &= \frac{\gamma}{2} + \frac{\gamma^2}{2(\sqrt{4(\delta^2 + \delta) + \gamma^2} + \sqrt{4(\delta^2 + \delta)})}. \end{aligned} \tag{2.19}$$

Using $\delta < 1/3$ and $\gamma < 1$, we know

$$\sqrt{4(\delta^2 + \delta) + \gamma} + \sqrt{4(\delta^2 + \delta)} < 3.$$

This along with (2.19) yields

$$\rho - \overline{\rho} > \frac{\gamma}{2} + \frac{\gamma^2}{6}. \tag{2.20}$$

Now consider the remaining case: $4\delta/(1+\delta) < \gamma < 1$. We have

$$\begin{aligned} \rho - \overline{\rho} &= \delta + \frac{\gamma}{2} + \frac{\sqrt{(2\delta)^2 + \gamma^2 + 4\delta}}{2} \\ &\quad - \left\{ \frac{\gamma}{2} - \delta(1 - \frac{\gamma}{2} + \frac{\sqrt{[\gamma(1+\delta) - 2\delta]^2 + 4\delta}}{2} \right\} \\ &= \left(2 - \frac{\gamma}{2}\right)\delta + \frac{(2\delta)^2 + \gamma^2 - [\gamma(1+\delta) - 2\delta]^2}{2(\sqrt{(2\delta)^2 + \gamma^2 + 4\delta} + \sqrt{[\gamma(1+\delta) - 2\delta]^2 + 4\delta})}. \end{aligned} \tag{2.21}$$

Let

$$f(\gamma) = (2\delta)^2 + \gamma^2 - [\gamma(1+\delta) - 2\delta]^2,$$

it is easy to see that

$$f(\gamma) = \gamma\delta[4(1+\delta) - \gamma(2+\delta)]$$

and

$$f'(\gamma) = 2\delta^2 > 0.$$

Thus by the monotonicity of f,

$$f(\gamma) > f\left(\frac{4\delta}{1+\delta}\right) = \frac{16\delta^2}{(1+\delta)^2}. \tag{2.22}$$

Moreover, using $\delta < 1/3$ and $\gamma < 1$ we have

$$\sqrt{(2\delta)^2 + \gamma^2 + 4\delta} + \sqrt{[\gamma(1+\delta) - 2\delta]^2 + 4\delta} < 3.$$

This combining with (2.21)-(2.22) leads to

$$\rho - \overline{\rho} > \frac{3}{2}\delta + \frac{16\delta^2}{6(1+\delta)^2} > \frac{3}{2}(\delta + \delta^2),$$

which completes the proof of (2.13). □

Remark 2.1 The key points in the proof of Theorem 2.1 is the replacement of the error e_i^x by the residual f_i and the use of the norm $|||\cdot|||$. The $n \times n$ matrix Q_{Ai} used in the proof can be viewed as a preconditioner of the matrix A (which is just the preconditioner $\hat{A}$ in [3] and the preconditioner Q_A in [5]). However, this preconditioner changes with the iteration number i. This makes the norm used in [3] (involving $\|\cdot\|_{\hat{A}}^2$) and the norm used in the proof of Theorem 3.1 of [5] (involving $\|\cdot\|_{Q_A - A}^2$) not applicable here for the analysis of the convergence of Algorithm 2.1, because the local contraction of errors with a changing norm can not guarantee the global convergence of the algorithm (refer to Theorem 4.2, [5]).

Remark 2.2 The numerical results shown in [5] indicated that Algorithm 2.1 may still converge even if the condition (2.6) is not satisfied. It was seen as a surprising phenomenon (see [5]). Now we would like to illustrate a bit about this phenomenon. It can be verified directly by (2.10) that $\|F\| < 1$ is equivalent to

$$|1 - \delta - \beta(1+\delta)| < 1 - \delta.$$

Thus, $\|F\| < 1$ if and only if

$$\beta < \frac{2(1-\delta)}{1+\delta}, \quad \forall \beta \in \sigma(Q_0). \tag{2.23}$$

Under the assumption (2.5), we can only obtain $\beta_{\max} \leq 1$. Here $\beta_{\max}$ denotes the maximal eigenvalue of the matrix Q_0. Solving the inequality

$1 < \frac{2(1-\delta)}{1+\delta}$, we get $\delta < \frac{1}{3}$, which means that the covergence condition $\delta < \frac{1}{3}$ is sharp theoretically when $\beta_{\max} = 1$. However, if the preconditioner $\hat{C}$ is not properly scaled $\beta_{\max}$ may be bounded far away from one (namely, close to zero). In this case (e.g., see the example considerd in [5]), the inequality (2.23) is also valid even if δ is close to one. But the convergence can be very slow (cf. [5]) because the assumption (2.5) holds only with the γ close to one (note that $0 < 1 - \gamma \le \beta_{\max}$). These facts indicate that the afore-mentioned phenomenon is not too surprising.

3 New algorithm I: the nonlinear inexact Uzawa combined with the steepest descent method

In general, the crucial assumption (2.5) is not so easy to fulfill as it requires a good estimate on the maximum eigenvalue of the preconditioned Schur complement $\hat{C}^{-1}C$ in order to have the desired proper scaling of $\hat{C}$. Without such a good estimate this assumption is not met or it is satisfied only with some γ very close 1, which may cause the divergence of the algorithm or results in very slow convergence of the algorithm. Obtaining such a good estimate of the eigenvalues is not very convenient or practical. In this and next sections, we propose two new algorithms to avoid such estimates.

Our first idea is to introduce a relaxation parameter $\overline{\tau}_i$ in (2.4) such that the norm

$$\|\overline{\tau}_i \hat{C}^{-1} g_i - C^{-1} g_i\|_C^2$$

is minimized, where $g_i = B^{\mathrm{t}} x_{i+1} - g$. If $g_i \neq 0$, the direct calculation gives

$$\overline{\tau} = \frac{(g_i, \hat{C}^{-1} g_i)}{\|\hat{C}^{-1} g_i\|_C^2} = \frac{(g_i, \hat{C}^{-1} g_i)}{(A^{-1} B \hat{C}^{-1} g_i, B \hat{C}^{-1} g_i)}.$$

Since the action of A^{-1} is not allowed, $A^{-1} B \hat{C}^{-1} g_i$ will be then replaced by $\Psi(B \hat{C}^{-1} g_i)$, which results in the following new algorithm:

Algorithm 3.1 *(Nonlinear inexact Uzawa-steepest descent).*
Given $x_0 \in R^n$ and $y_0 \in R^m$, the sequence $\{(x_i, y_i)\} \subset R^n \times R^m$ is defined for $i = 1, 2, \cdots$, by

Step 1. Compute $f_i = f - (Ax_i + By_i)$ and $\Psi(f_i)$, update

$$x_{i+1} = x_i + \Psi(f_i). \tag{3.1}$$

Step 2. Compute $g_i = B^{\mathrm{t}} x_{i+1} - g,\ d_i = \hat{C}^{-1} g_i$ and

$$\tau_i = \begin{cases} \frac{1}{2} \frac{(g_i, d_i)}{(\Psi(Bd_i), Bd_i)} & \text{for} \quad g_i \neq 0; \\ 1 & \text{for} \quad g_i = 0. \end{cases} \tag{3.2}$$

Update

$$y_{i+1} = y_i + \tau_i \, d_i. \tag{3.3}$$

Remark 3.1 It is clear that when both f_i and g_i vanish, then the vectors x_i and y_i are the exact solution of the equation (1.1), and thus Algorithm 3.1 terminates.

For the sake of convenience, we introduce two parameters δ and δ_0 which are the minimal positive numbers satisfying

$$\|A^{-1}f_i - \Psi(f_i)\|_A \leq \delta \|A^{-1}f_i\|_A \tag{3.4}$$

$$\|A^{-1}Bd_i - \Psi(Bd_i)\|_A \leq \delta_0 \|A^{-1}Bd_i\|_A. \tag{3.5}$$

Remark 3.2 Note that the assumptions (3.4) and (3.5) are in fact consequences of the assumption (2.2). So we can simply assume the condition (2.2) instead of (3.4)-(3.5) and all the results in this and next sections hold. We choose to formulate the above two conditions here so that we can see how the convergence rate of Algorithm 3.1 depends more explicitly on the accuracies of the two nonlinear inner iterations in both (3.1) and (3.3). The evaluations of $A^{-1}f_i$ and $A^{-1}Bd_i$ by the nonlinear iteration Ψ can be different.

Let

$$\kappa = \mathrm{cond}(\hat{C}^{-1}C), \quad \beta = \sqrt{1 - \frac{4\kappa(1-2\delta_0)}{(1+\kappa)^2(1-\delta_0)^2}}, \tag{3.6}$$

then we have

Theorem 3.1 *Assume that the approximation parameters δ and δ_0 satisfy $\delta < \frac{1}{3}$ and $\delta_0 < \frac{1}{2}$, then Algorithm 3.1 converges. Moreover, we have*

$$\|\mathrm{e}_i^x\|_A \leq (\sqrt{1+4\delta} + \hat{\rho})\hat{\rho}^{\mathrm{i}-1}|||E_0|||, \quad i = 1, 2, \cdots \tag{3.7}$$

and

$$\|\mathrm{e}_i^y\|_C \leq \hat{\rho}^i |||E_0|||, \quad i = 1, 2, \cdots. \tag{3.8}$$

Here the rate of convergence $\hat{\rho}$ (< 1) can be estimated by

$$\hat{\rho} = \begin{cases} \sqrt{\delta + \delta^2} + \delta & \text{for} \quad 0 < \frac{1+\beta}{2} \leq \frac{4\delta}{1+\delta}; \\ 1 - \frac{1}{4}(1-\beta)(1+\delta) & \text{for} \quad \frac{4\delta}{1+\delta} < \frac{1+\beta}{2} < 1. \end{cases} \tag{3.9}$$

Remark 3.3 From Theorem 3.1 we know that the convergence of Algorithm 3.1 is independent of the spectrum of the preconditioned Schur complement $\hat{C}^{-1}C$, and the convergence rate of this new algorithm depends only on the condition number κ, not necessary to have a proper scaling of the preconditioner $\hat{C}$ and a good apriori estimate on the maximum and minimum eigenvalues of $\hat{C}^{-1}C$. This seems to be an important advantage of the new algorithm over the existing nonlinear inexact Uzawa algorithms. The assumptions of Theorem 3.1 are natural and can be satisfied by the approximate inverse associated with the PCG iteration and the approximate inverse defined by one sweep of a multigrid algorithm with a suitable smoother (cf. [3,5,4] and [23]).

For the proof of Theorem 3.1 we need the following lemma:

Lemma 3.1 *Assume that the condition (3.5) holds with* $\delta_0 < \frac{1}{2}$. *Then for any given natural number* i *there is a symmetric and positive definite* $m \times m$ *matrix* Q_{Bi} *such that*

(i) $$Q_{Bi}^{-1}g_i = \tau_i \hat{C}^{-1} g_i.$$
(ii) All eigenvalues of the matrix $Q_{Bi}^{-1}C$ *are in the interval* $[\frac{1-\beta}{2}, 1]$.

Proof of Theorem 3.1. With Lemma 3.1, we can view the matrix Q_{Bi} as the preconditioner $\hat{C}$ for the Schur complement C, which satisfies the condition (2.5) with $\gamma = \frac{1-\beta}{2}$. Thus Theorem 3.1 is a direct consequence of Theorems 2.1-2.2. Here we have to make use of the important fact that the norms adopted in Theorems 2.1-2.2 are independent of the preconditioner $\hat{C}$, in comparison with [5]. □

Proof of Lemma 3.1. It is easy to see that

$$(\Psi(B\hat{C}^{-1}g_i), B\hat{C}^{-1}g_i) = \|\hat{C}^{-1}g_i\|_C^2 + (\Psi(B\hat{C}^{-1}g_i) \\ -A^{-1}(B\hat{C}^{-1}g_i), B\hat{C}^{-1}g_i). \tag{3.10}$$

Using the Cauchy's inequality and (3.5), yields

$$\begin{aligned}
&\left|(\Psi(B\hat{C}^{-1}g_i) - A^{-1}(B\hat{C}^{-1}g_i), B\hat{C}^{-1}g_i)\right| \\
&\quad \le \|\Psi(B\hat{C}^{-1}g_i) - A^{-1}(B\hat{C}^{-1}g_i)\|_A \, \|B\hat{C}^{-1}g_i)\|_{A^{-1}} \\
&\quad \le \delta_0 \|A^{-1}(B\hat{C}^{-1}g_i)\|_A \, \|B\hat{C}^{-1}g_i)\|_{A^{-1}} \\
&\quad = \delta_0 \|\hat{C}^{-1}g_i\|_C^2.
\end{aligned}$$

Substituting the above inequality into (3.10) leads to

$$(1-\delta_0)\|\hat{C}^{-1}g_i\|_C^2 \le (\Psi(B\hat{C}^{-1}g_i), B\hat{C}^{-1}g_i) \le (1+\delta_0)\|\hat{C}^{-1}g_i\|_C^2.$$

Thus,

$$\frac{(g_i, \hat{C}^{-1} g_i)}{(1+\delta_0)^2 \|\hat{C}^{-1} g_i\|_C^2} \le \widetilde{\tau}_i \le \frac{(g_i, \hat{C}^{-1} g_i)}{(1-\delta_0)^2 \|\hat{C}^{-1} g_i\|_C^2}, \tag{3.11}$$

where the parameter $\tilde{\tau}_i$ is defined by

$$\widetilde{\tau}_i = \frac{(g_i, \hat{C}^{-1} g_i)}{\|\Psi(B\hat{C}^{-1} g_i)\|_A^2}.$$

For $\tau > 0$, let

$$F(\tau) = \|\tau \hat{C}^{-1} g_i - C^{-1} g_i\|_C^2.$$

Clearly,

$$F(\tau) = \|C^{-1} g_i\|_C^2 - 2\tau (g_i, \hat{C}^{-1} g_i) + \tau^2 \|\hat{C}^{-1} g_i\|_C^2. \tag{3.12}$$

It is easy to verify that $F(\tau)$ is an increasing function for

$$\tau \ge \frac{(g_i, \hat{C}^{-1} g_i)}{\|\hat{C}^{-1} g_i\|_C^2},$$

and is a decreasing function for

$$\tau \le \frac{(g_i, \hat{C}^{-1} g_i)}{\|\hat{C}^{-1} g_i\|_C^2}.$$

Using this property and (3.11), we know

$$F(\widetilde{\tau}_i) \le \max\left\{ F\left(\frac{\tau_{0i}}{1+\delta_0}\right),\ F\left(\frac{\tau_{0i}}{1-\delta_0}\right) \right\} \tag{3.13}$$

where

$$\tau_{0i} = \frac{(g_i, \hat{C}^{-1} g_i)}{\|\hat{C}^{-1} g_i\|_C^2}.$$

On the other hand, it follows from (3.12) that

$$\begin{aligned} F\left(\frac{\tau_{0i}}{1+\delta_0}\right) &= \|C^{-1} g_i\|_C^2 - \frac{2(1+\delta_0)-1}{(1+\delta_0)^2}\, \frac{(g_i, \hat{C}^{-1} g_i)^2}{\|\hat{C}^{-1} g_i\|_C^2} \\ &= \left\{ 1 - \frac{2(1+\delta_0)-1}{(1+\delta_0)^2}\, \frac{(g_i, \hat{C}^{-1} g_i)^2}{\|\hat{C}^{-1} g_i\|_C^2\, \|C^{-1} g_i\|_C^2} \right\} \|C^{-1} g_i\|_C^2. \end{aligned}$$

Let λ_{min} and λ_{max} denote respectively the minimal and maximal eigenvalues of the matrix $\hat{C}^{-\frac{1}{2}}C\hat{C}^{-\frac{1}{2}}$. Then using the well-known matrix-eigenvalue inequality we obtain

$$\begin{aligned}&\frac{(g_i,\hat{C}^{-1}g_i)^2}{\|\hat{C}^{-1}g_i\|_C^2\,\|C^{-1}g_i\|_C^2}\\&=\frac{(\hat{C}^{-\frac{1}{2}}g_i,\hat{C}^{-\frac{1}{2}}g_i)^2}{(\hat{C}^{-\frac{1}{2}}C\hat{C}^{-\frac{1}{2}}(\hat{C}^{-\frac{1}{2}}g_i),\hat{C}^{-\frac{1}{2}}g_i)\,((\hat{C}^{-\frac{1}{2}}C\hat{C}^{-\frac{1}{2}})^{-1}(\hat{C}^{-\frac{1}{2}}g_i),\hat{C}^{-\frac{1}{2}}g_i)}\\&\geq\frac{4\lambda_{min}\,\lambda_{max}}{(\lambda_{min}+\lambda_{max})^2}=\frac{4\kappa}{(1+\kappa)^2}.\end{aligned}$$

Hence

$$F\left(\frac{\tau_{0i}}{1+\delta_0}\right)\leq\left\{1-\frac{2(1+\delta_0)-1}{(1+\delta_0)^2}\,\frac{4\kappa}{(1+\kappa)^2}\right\}\|C^{-1}g_i\|_C^2. \tag{3.14}$$

Similarly we can show (note that $\delta_0<\frac{1}{2}$)

$$F\left(\frac{\tau_{0i}}{1-\delta_0}\right)\leq\left\{1-\frac{2(1-\delta_0)-1}{(1-\delta_0)^2}\,\frac{4\kappa}{(1+\kappa)^2}\right\}\|C^{-1}g_i\|_C^2. \tag{3.15}$$

Now by the direct calculation, we have

$$\begin{aligned}&\frac{2(1+\delta_0)-1}{(1+\delta_0)^2}-\frac{2(1-\delta_0)-1}{(1-\delta_0)^2}\\&\quad=\left(\frac{1}{1-\delta_0}-\frac{1}{1+\delta_0}\right)\left(\frac{1}{1-\delta_0}+\frac{1}{1+\delta_0}-2\right)\\&\quad=\left(\frac{1}{1-\delta_0}-\frac{1}{1+\delta_0}\right)\frac{2\delta_0^2}{1-\delta_0^2}>0,\end{aligned}$$

namely,

$$\frac{2(1-\delta_0)-1}{(1-\delta_0)^2}<\frac{2(1+\delta_0)-1}{(1+\delta_0)^2},$$

which, together with (3.13)-(3.15), yields

$$\begin{aligned}F(\widetilde{\tau}_i)&\leq\left\{1-\frac{2(1-\delta_0)-1}{(1-\delta_0)^2}\,\frac{4\kappa}{(1+\kappa)^2}\right\}\|C^{-1}g_i\|_C^2\\&=\beta^2\|C^{-1}g_i\|_C^2.\end{aligned}$$

Summarizing the above, we have proved

$$\|\widetilde{\tau}_i\hat{C}^{-1}g_i-C^{-1}g_i\|_C\leq\beta\|C^{-1}g_i\|_C.$$

It is clear that $\beta < 1$ when $\delta_0 < \frac{1}{2}$. Hence by Lemma 9 of [3] we know that there is a symmetric and positive definite $m \times m$ matrix $\hat{Q}_{Bi}$ such that

$$\hat{Q}_{Bi}^{-1} g_i = \widetilde{\tau}_i \hat{C}^{-1} g_i$$

and

$$\left\| I - \hat{Q}_{Bi}^{-\frac{1}{2}} C \hat{Q}_{Bi}^{-\frac{1}{2}} \right\| \leq \beta.$$

Now $Q_{Bi} = 2\hat{Q}_{Bi}$ is the desired matrix, since the matrix $Q_{Bi}^{-1} C$ has the same eigenvalues as the matrix $Q_{Bi}^{-\frac{1}{2}} C Q_{Bi}^{-\frac{1}{2}}$. □

4 New algorithm II: the nonlinear inexact Uzawa combined with the conjugate gradient method

In the last section, we proposed a nonlinear inexact Uzawa combined with the steepest descent method (Algorithm 3.1), which converges when a general preconditioner is used for the Schur complement system and a general nonlinear iteration is used for solving $Ax = b$ 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. This is the case when the saddle-point problem arises, for example, from the Stokes problem, mixed finite element formulations of second order elliptic problems [22], and domain decomposition methods with Lagrange multipliers [17]. Without such a good preconditioner the method may converge with a slow rate. This fact can be observed from the numerical experiments shown in Sect. 5.

In this section, we propose another new algorithm which combines the nonlinear inexact Uzawa algorithm with the conjugate gradient (CG) method, in an effort to accelerate the nonlinear inexact Uzawa algorithm when a good preconditioner is not available for the Schur complement system. This is the case when the saddle-point problems arise from the Lagrange multiplier formulations for optimization problems [16] and the parameter identification [12] [19].

In the following, we will first derive a new nonlinear inexact Uzawa algorithm and combine it with the CG method, then analyse the convergence of the new algorithm and compare its convergence rate with that of Algorithm 3.1, which uses the steepest descent method.

For a given sequence $\{f_i\} \subset R^n$, let $\{Q_{Ai}\}$ be the sequence of symmetric and positive definite $n \times n$ matrices satisfying (2.16) and (2.17), and set $H_i = B^{\mathrm{t}} Q_{Ai}^{-1} B$. Clearly, H_i is also a symmetric and positive definite $m \times m$

matrix. For a given sequence $\{g_i\} \subset R^m$, consider the linear system (with the index i fixed)

$$H_i z_i = g_i. \tag{4.1}$$

We intend to solve (4.1) for each fixed index i by the PCG iteration with the preconditioner $\hat{C}$. This process is to find a sequence of approximate solutions $\{\hat{z}_i^j\}$ as follows:

$$\begin{array}{ll}
\text{initialize :} & \hat{z}_i^0 = 0, \quad \hat{r}_i^0 = g_i, \quad \hat{p}_i^0 = \hat{C}^{-1} g_i, \quad \hat{h}_i^0 = 0; \\
\text{iterate for } j = 1, 2, \cdots : & \\
& \hat{z}_i^j = \hat{z}_i^{j-1} + \hat{\tau}_i^j \hat{p}_i^{j-1}, \quad \hat{\tau}_i^j = \frac{(\hat{r}_i^{j-1}, \hat{p}_i^{j-1})}{(H_i \hat{p}_i^{j-1}, \hat{p}_i^{j-1})}, \\
& \hat{h}_i^j = \hat{h}_i^{j-1} + \hat{\tau}_i^j H_i \hat{p}_i^{j-1}, \\
& \hat{r}_i^j = g_i - \hat{h}_i^j, \\
& \hat{p}_i^j = \hat{C}^{-1} \hat{r}_i^j - \hat{\theta}_i^j \hat{p}_i^{j-1}, \quad \hat{\theta}_i^j = \frac{(\hat{C}^{-1} \hat{r}_i^j, H_i \hat{p}_i^{j-1})}{(H_i \hat{p}_i^{j-1}, \hat{p}_i^{j-1})}.
\end{array}$$

Note that one has the relation $\hat{h}_i^j = H_i \, \hat{z}_i^j$ between the variables $\hat{h}_i^j$ and $\hat{z}_i^j$ above, and each action of H_i involves the action of Q_{Ai}^{-1}. In general it is difficult to obtain the explicit form of the matrix Q_{Ai}. Thus, it is important and practical to use the inexact solver $\Psi(B\hat{p}_i^{j-1})$ to replace $Q_{Ai}^{-1} B\hat{p}_i^{j-1}$. The resulting iteration can be formulated as follows:

$$\begin{array}{ll}
\text{initialize :} & z_i^0 = 0, \quad r_i^0 = g_i, \quad p_i^0 = \hat{C}^{-1} g_i, \quad h_i^0 = 0; \\
\text{iterate for } j = 1, 2, \cdots : & \\
& z_i^j = z_i^{j-1} + \tau_i^j p_i^{j-1}, \quad \tau_i^j = \frac{(r_i^{j-1}, p_i^{j-1})}{(\Psi(Bp_i^{j-1}), Bp_i^{j-1})}, \\
& h_i^j = h_i^{j-1} + \tau_i^j B^{\mathrm{t}} \Psi(Bp_i^{j-1}), \\
& r_i^j = g_i - h_i^j, \\
& p_i^j = \hat{C}^{-1} r_i^j - \theta_i^j p_i^{j-1}, \; \theta_i^j = \frac{(B\hat{C}^{-1} r_i^j, \Psi(Bp_i^{j-1}))}{(\Psi(Bp_i^{j-1}), Bp_i^{j-1})}.
\end{array}$$

For convenience, we use $\Phi_k(g_i)$ to denote the kth iterate z_i^k generated from the above process. Φ_k defines a (nonlinear) mapping from R^m to R^m, and $\Phi_k(g_i)$ can be regarded as an approximate solution of the system (4.1). Note that we may not have $\Psi(Bb) = Q_{Ai}^{-1} Bb$ for a vector $b \in R^m$ in general, so $\Phi_k(g_i)$ is not the true approximation generated by the k steps of PCG iteration for (4.1).

We assume that there is a positive number $\beta_k < 1$ such that

$$\|\Phi_k(g_i) - z_i\|_{H_i} = \|\Phi_k(g_i) - H_i^{-1} g_i\|_{H_i} \le \beta_k \|H_i^{-1} g_i\|_{H_i}. \tag{4.2}$$

Then by Lemma 9 of [3], there is a symmetric and positive definite $m \times m$ matrix $\hat{Q}_{ki}$, such that $\hat{Q}_{ki}^{-1} g_i = \Phi_k(g_i)$ and

$$\left\| I - \hat{Q}_{ki}^{-\frac{1}{2}} H_i \hat{Q}_{ki}^{-\frac{1}{2}} \right\| \le \beta_k,$$

which implies

$$\frac{1}{1+\beta_k}(H_i w, w) \le (\hat{Q}_{ki} w, w) \le \frac{1}{1-\beta_k}(H_i w, w) \quad \forall\, w \in R^m. \tag{4.3}$$

On the other hand, it follows from (2.17) that

$$(1-\delta)(A^{-1}w, w) \le (Q_{Ai}^{-1} w, w) \le (1+\delta)(A^{-1}w, w) \quad \forall\, w \in R^m.$$

This, together with (4.3), leads to

$$\frac{1-\delta}{1+\beta_k}(C\, w, w) \le (\hat{Q}_{ki} w, w) \le \frac{1+\delta}{1-\beta_k}(C\, w, w). \tag{4.4}$$

Define $Q_{ki} = \frac{2}{1-\delta}\hat{Q}_{ki}$. It is easy to see that

$$Q_{ki}^{-1} g_i = \frac{1-\delta}{2}\Phi_k(g_i), \tag{4.5}$$

and

$$\begin{aligned} \frac{(1-\delta)(1-\beta_k)}{2(1+\delta)}(Q_{ki} w, w) &\le (C\, w, w) \\ &\le (Q_{ki} w, w) \quad \forall\, w \in R^m \end{aligned} \tag{4.6}$$

by using (4.4) and the fact that $\beta_k < 1$.

Based on (4.5) and Algorithm 2.1, we propose the following new algorithm:

Algorithm 4.1 *(Nonlinear inexact Uzawa–PCG method).*
Given $x_0 \in R^n$ and $y_0 \in R^m$, the sequence $\{(x_i, y_i)\} \subset R^n \times R^m$ is defined for $i = 1, 2, \cdots$, by

Step 1. Compute $f_i = f - (Ax_i + By_i)$ and $\Psi(f_i)$, update

$$x_{i+1} = x_i + Q_{Ai}^{-1} f_i = x_i + \Psi(f_i). \tag{4.7}$$

Step 2. Compute $g_i = B^{\mathrm{t}} x_{i+1} - g$, and $d_{ki} = \Phi_k(g_i)$, update

$$y_{i+1} = y_i + Q_{ki}^{-1}\, g_i = y_i + \frac{1-\delta}{2}\, d_{ki}. \tag{4.8}$$

The inequality (4.6) indicates that the matrix Q_{ki} satisfies the inequality (2.5) with $\gamma_k = \frac{1+3\delta+\beta_k(1-\delta)}{2(1+\delta)}$. Thus by Theorem 2.2 we obtain

Theorem 4.1 *Assume that (2.2) holds for $\delta < \frac{1}{3}$, and (4.2) holds for $\beta_k < 1$. Then Algorithm 4.1 converges. Moreover, we have*

$$\|\mathrm{e}_i^x\|_A \le (\sqrt{1+4\delta} + \hat\rho_k)(\hat\rho_k)^{\mathrm{i}-1}|||E_0|||, \quad i = 1, 2, \cdots \tag{4.9}$$

and

$$\|\mathrm{e}_i^y\|_C \le (\hat\rho_k)^i|||E_0|||, \quad i = 1, 2, \cdots , \tag{4.10}$$

and the convergence rate $\hat\rho_k$ (< 1) can be estimated by

$$\hat\rho_k \le \begin{cases} \sqrt{\delta+\delta^2} + \delta & \text{for} \quad 0 \le \beta_k \le \frac{5\delta-1}{1-\delta}, \\ 1 - \frac{1}{4}(1-\beta_k)(1-\delta) & \text{for} \quad \frac{5\delta-1}{1-\delta} < \beta_k < 1. \end{cases} \tag{4.11}$$

Remark 4.1 One can easily see from (4.5)-(4.6) that Theorem 4.1 also holds if the factor $(1-\delta)/2$ in (4.8) is replaced by any number $\alpha \in (0, 1/3)$. In fact, the numerical results in Sect. 5 indicate that Algorithm 4.1 still converges when the factor $(1-\delta)/2$ is replaced by $\frac{1}{2}$. So Algorithm 3.1 is a special case of Algorithm 4.1 with $k = 1$.

As noted in Remark 3.3, the first assumption of Theorem 4.1 holds when the mapping $\varPsi$ is properly defined. However, it is difficult to verify the condition (4.2) in general. The following proposition gives a condition which guarantees (4.2) when all H_i's are taken to be the same:

Proposition 4.1 *Assume that there is a symmetric and positive definite $n \times n$ matrix Q_A, such that $Q_A^{-1}\phi = \varPsi(\phi)$ for* **any** *$\phi \in R^n$. Define $H = B^{\mathrm{t}}Q_A^{-1}B$. Then the following inequality holds for some $\beta_k < 1$:*

$$\|\varPhi_k(g_i) - H^{-1}g_i\|_H \le \beta_k\|H^{-1}g_i\|_H. \tag{4.12}$$

Proof. Under the assumption of Proposition 4.1, $\varPhi_k(g_i)$ is just the approximate solution generated by the k steps of PCG iteration for the linear system

$$H\, z_i = g_i. \tag{4.13}$$

Thus, the parameter β_k satisfying the relation (4.12) can be estimated by (cf. [2])

$$\beta_k = \min\left\{2\left(\frac{\sqrt{\kappa^*}-1}{\sqrt{\kappa^*}+1}\right)^k, \frac{\kappa^*-1}{\kappa^*+1}\right\} < 1$$

with κ^* =cond$(\hat C^{-1}H)$. □

Remark 4.2 The assumption in Proposition 4.1 is satisfied in some applications. For example, it is true when the mapping Ψ is defined by the preconditioned Richardson iteration, which includes the standard multigrid sweeps.

To see this, for any $b \in R^n$, recall the preconditioned Richardson iteration for solving the system

$$A\,x = b \tag{4.14}$$

with the preconditioner $\hat{A}$ is

$$x_{i+1} = x_i + \hat{A}^{-1}(b - A\,x), \quad i = 1, 2, \cdots .$$

If the eigenvalues of $\hat{A}^{-1}A$ is in the range $0 < \lambda < 2$, the iteration converges. Taking $x_0 = 0$, we can easily see

$$\begin{aligned} x_{i+1} &= x - (I - \hat{A}^{-1}A)^{\mathrm{i+1}}x \\ &= \sum_{j=1}^{\mathrm{i+1}} (-1)^{j-1} C_{i+1}^{j} (\hat{A}^{-1}A)^{j} x \\ &= \sum_{j=1}^{\mathrm{i+1}} (-1)^{j-1} C_{i+1}^{j} (\hat{A}^{-1}A)^{j-1} \hat{A}^{-1}\, b\,. \end{aligned}$$

This suggests us to define Q_A^{-1} as

$$Q_A^{-1} = \sum_{j=1}^{\mathrm{i+1}} (-1)^{j-1} C_{i+1}^{j} (\hat{A}^{-1}A)^{j-1} \hat{A}^{-1} = \Big(I - (I - \hat{A}^{-1}A)^{\mathrm{i+1}}\Big) A^{-1}\,.$$

We can easily verify that the matrix Q_A is symmetric and positive definite, and satisfies

$$Q_A^{-1} b = x_{i+1} = \Psi(b) \quad \forall\, b \in R^n.$$

Note that if A is the stiffness matrix which arises from the finite element discretization of self-adjoint second order elliptic problems, then one can choose $\hat{A}^{-1}$ to be the operator corresponding to one V-cycle sweep of multigrid method [5] or the hierarchical basis multigrid method [3].

In the following we make some comparison between the convergence rates of Algorithm 3.1 and Algorithm 4.1.

Proposition 4.2 *Assume that the parameters δ, κ and β_k in (2.2), (3.6) and (4.2) satisfy*

$$\delta < \frac{1}{5}, \quad \kappa = \operatorname{cond}(\hat{C}^{-1}C) > 2, \quad \lim_{k\to\infty} \beta_k = 0$$

respectively. Then Algorithm 4.1 converges faster than Algorithm 3.1 when k is sufficiently large. But this may not be true for general $k \geq 2$.

Proof. We first show that the parameter β in (3.6) is larger than $1/3$ when $\kappa > 2$. It is easy to see that

$$\frac{1-2\delta_0}{(1-\delta_0)^2} \leq 1,$$

and so

$$\beta \geq \sqrt{1 - \frac{4\kappa}{(1+\kappa)^2}} = \frac{\kappa - 1}{\kappa + 1} > \frac{1}{3}$$

when $\kappa > 2$.

Now let $\hat{\rho}$ and $\hat{\rho}_k$ be the convergence rates of Algorithm 3.1 and Algorithm 4.1 (see Theorems 3.1 and 4.1). Since $\beta > \frac{1}{3}$ and $\delta < \frac{1}{5}$, it is easy to see

$$\frac{4\delta}{1+\delta} < \frac{2}{3} < \frac{1+\beta}{2} < 1 \text{ and } \frac{5\delta - 1}{1-\delta} < 0 \leq \beta_k.$$

Then by (3.9) and (4.11) we have

$$\hat{\rho} = 1 - \frac{1}{4}(1-\beta)(1+\delta) > 1 - \frac{3}{10}(1-\beta)$$

and

$$\hat{\rho}_k = 1 - \frac{1}{4}(1-\beta_k)(1-\delta) < 1 - \frac{1}{5}(1-\beta_k).$$

Therefore one has $\hat{\rho}_k < \hat{\rho}$ if

$$\frac{1}{5}(1-\beta_k) > \frac{3}{10}(1-\beta),$$

that is,

$$\beta_k < \frac{3}{2}\beta - \frac{1}{2}. \tag{4.15}$$

It is clear that when $\beta > \frac{1}{3}$ the inequality (4.15) can be satisfied for large k.

It remains to show the second part of Proposition 4.2. Form Theorem 2.1 and the proof of Lemma 3.1, we know that the convergence rate of Algorithm 4.1 is determined by the parameter $\hat{\beta}_k$ satisfying

$$\|\Phi_k(g_i) - C^{-1}g_i\|_C \leq \hat{\beta}_k \|C^{-1}g_i\|_C, \quad g_i \in R^m,$$

more accurately, it is determined by the value $\|\Phi_k(g_i) - C^{-1}g_i\|_C$, i.e., the error at the k-th iteration. Since the results generated by Algorithm 4.1 with $k = 1$ are the same as the ones generated by Algorithm 3.1, for our purpose it suffices to construct a counterexample to show that the following inequality does not hold:

$$\|\Phi_2(g_i) - C^{-1}g_i\|_C < \|\Phi_1(g_i) - C^{-1}g_i\|_C. \tag{4.16}$$

In fact, we can show that (4.16) is not true when $\hat{C} = C = I$. By the definition of Algorithm 4.1, we have

$$\Phi_2(g_i) = z_i^2 = z_i^1 + \tau_i^2\, p_i^1,$$

thus

$$\begin{aligned}\|\Phi_2(g_i) - C^{-1}g_i\|_C^2 &= \|z_i^2 - g_i\|^2 \\ &= \|z_i^1 - g_i\|^2 + 2\tau_i^2\,(z_i^1 - g_i, p_i^1) + (\tau_i^2)^2\,\|p_i^1\|^2. \end{aligned} \tag{4.17}$$

By the basic property of the CG method, we know $(r_i^1, p_i^0) = 0$, which implies

$$(r_i^1, p_i^1) = (r_i^1, r_i^1 - \theta_i^1 p_i^0) > 0,$$

and so $\tau_i^2 > 0$ by the definition of τ_i^2. Now the desired conclusion follows if we can find a matrix H such that $(z_i^1 - g_i, p_i^1) > 0$. Indeed, if this is true, then it follows from (4.17) that

$$\|\Phi_2(g_i) - C^{-1}g_i\|_C^2 > \|z_i^1 - g_i\|^2 = \|\Phi_1(g_i) - C^{-1}g_i\|_C^2,$$

which negates (4.16). It remains to find such a matrix H. By the definition of Algorithm 4.1, $z_i^1 = \tau_i^1\, g_i$ and

$$p_i^1 = r_i^1 - \theta_i^1 p_i^0 = g_i - \tau_i^1 H g_i - \theta_i^1 g_i.$$

Therefore we have

$$\begin{aligned}(z_i^1 - g_i, p_i^1) &= (\tau_i^1 - 1)(g_i, (1 - \theta_i^1)g_i - \tau_i^1 H g_i) \\ &= (\tau_i^1 - 1)\Big\{(1 - \theta_i^1)\|g_i\|^2 - \tau_i^1\|g_i\|_H^2\Big\} \\ &= \theta_i^1(1 - \tau_i^1)\|g_i\|^2, \end{aligned} \tag{4.18}$$

where we have used the fact that $\tau_i^1 = \|g_i\|^2/\|g_i\|_H^2$. By the definitions of θ_i^1, it is easy to verify that

$$\theta_i^1(1 - \tau_i^1) = \frac{(\|g_i\|^2\,\|Hg_i\|^2 - \|g_i\|_H^4)(\|g_i\|^2 - \|g_i\|_H^2)}{\|g_i\|_H^6}. \tag{4.19}$$

By the Cauchy-Schwarz inequality,

$$\|g_i\|^2 \, \|Hg_i\|^2 - \|g_i\|_H^4 > 0$$

provided that $H \neq c\,I$ (c is a constant). Without loss of generality, we assume that the first component of g_i is not zero, then define

$$H = \text{diag}(\frac{3}{4}\ 1\ 1\ \cdots\ 1).$$

It is clear that

$$\|g_i\|^2 - \|g_i\|_H^2 > 0,$$

so we have from (4.19) that $\theta_i^1(1 - \tau_i^1) > 0$. This, together with (4.18), leads to

$$(z_i^1 - g_i, p_i^1) > 0. \qquad \square$$

5 Numerical experiments

In this section, we apply our newly proposed Algorithm 3.1 and Algorithm 4.1, and some other existing algorithms, to solve the two-dimensional Stokes problem and another system of purely algebraic equations. Let Ω be the unit square domain in R^2, and $L_0^2(\Omega)$ be the set of all square integrable functions over Ω with zero averages. Let $H^1(\Omega)$ be the usual Sobolev space of order one on Ω and $H_0^1(\Omega)$ be a space consisting of those functions in $H^1(\Omega)$ whose traces vanish on the boundary of Ω.

Our first example is the generalized Stokes problem whose variational formulation reads as follows:

$$(5.1) \qquad \begin{cases} (\mu\nabla u, \nabla v) - (p, \nabla\cdot v) = (f, v), & \forall v \in H_0^1(\Omega)^2, \\ (q, \nabla\cdot u) = g, \quad \forall q \in L_0^2(\Omega), \end{cases}$$

where $f \in L^2(\Omega)$ and $\mu \in L^\infty(\Omega)$ with $\mu(x) \geq c > 0$ almost everywhere in Ω.

We use one of the well-known conforming Taylor-Hood elements, which has been widely used in engineering, to solve the system (5.1). For any positive integer N, we divide Ω into $N \times N$ subsquares, and set $h = 1/N$. Such a triangulation of Ω, with its elements being all squares, is denoted as $\mathcal{T}^h$. Let $X_h \subset H_0^1(\Omega)$ and $M_h \subset H^1(\Omega) \cap L_0^2(\Omega)$ be the usual continuous Q_2 and Q_1 finite element spaces defined on $\mathcal{T}^h$ respectively (cf. [7,13]). The total number of unknowns for this finite element is $n + m = 2(2N -$

Table 1. Number of iterations (Algorithm 3.1)

	δ=0.1			δ=0.2		
N	8	16	32	8	16	32
PCG	21	21	22	26	25	24
CG	22	24	24	31	31	30

$1)^2 + [(N+1)^2 - 1]$. The finite element approximation of the above Stokes system can be formulated as follows:

$$(\mu(x)\nabla u_h, \nabla v) - (p_h, \nabla\cdot v) = (f, v), \quad \forall\, v \in X_h^2, \tag{5.2}$$

$$(q, \nabla\cdot u_h) = 0, \quad \forall\, q \in M_h. \tag{5.3}$$

It is known that the *inf-sup* condition is satisfied by the pair (X_h^2, M_h) (see [7]), thus the Schur complement matrix $B^{\mathrm{t}}A^{-1}B$ associated with the system (5.2)-(5.3) has a condition number independent of h. As in [6] and [22], we take the variable coefficient μ to be $\mu = 1 + x_1x_2 + x_1^2 - x_2^2/2$, and know that the corresponding matrix A is block diagonal with two copies of the stiffness matrix associated with the bilinear form $(\mu(x)\nabla\cdot\,, \nabla\cdot\,)$ on the diagonal. Therefore, if $\mu = 1$, it can be solved by the fast Poisson solver. It is natural to choose this fast solver $\hat{A}$ as the preconditioner of A. It is clear that

$$0.5\,(\hat{A}z, z) \le (Az, z) \le 2.5\,(\hat{A}z, z), \tag{5.4}$$

thus the matrix $\hat{A}^{-1}A$ is well-conditioned, and so is the matrix $B^{\mathrm{t}}\hat{A}^{-1}B$. In this case, we can choose for Algorithm 3.1 $\hat{C} = I$ and Ψ defined by the CG method (without preconditioning) or the PCG method with the preconditioner $\hat{A}$ for A. For all the numerical results shown below, the inner nonlinear iteration is terminated when Ψ satisfies

$$\|\phi - A\Psi(\phi)\|/\|\phi\| \le \delta$$

for a given vector $\phi \in R^m$. And the outer iteration of Algorithm 3.1 terminates when

$$\varepsilon = \|Mv_i - b\|/\|Mv_0 - b\| \le 1.0 \times 10^{-5} \tag{5.5}$$

with $v_i = (x_i\, y_i)^{\mathrm{t}}$, $v_0 = (x_0\, y_0)^{\mathrm{t}}$ and $b = (f\, g)^{\mathrm{t}}$. The number of iterations of Algorithm 3.1 is listed in Table 1, where we can see the rate of convergence independent of mesh size h.

In comparison with Algorithm 3.1, we tried to solve the Stokes system also by Algorithm 4.1 with the same preconditioners as above. The number of iterations for different k is listed in Table $1'$.

Table 1′. Number of iterations (Algorithm 4.1) PCG inner iteration, $\delta = 0.1$

N	k=2	k=5	k=10	k=20
8	102	60	30	16
16	19	28	45	28
32	19	18	27	20

We can see from Table 1′ that Algorithm 3.1 outperforms Algorithm 4.1 in this example, and Algorithm 3.1 shows a more stable and consistent convergence history than Algorithm 4.1. The results are reasonable as Algorithm 3.1 should perform better than Algorithm 4.1 when a good preconditioner $\hat{C}$ is available for the Schur complement C. Indeed, the preconditioner $h^2\hat{C} = h^2 I$ is spectrally equivalent to C (and the two algorithms are invariant under constant scalings).

It can be verified by (5.4) that $\hat{C} = 2I$ satisfies the assumption (2.5) (cf. [6]). Thus we can also apply Algorithm 2.1 to solve the system (5.2)-(5.3). However, the resulting convergence is very slow for this example because the parameter γ defined by (2.5) is close to one. For example, 129 iterations are needed to reach the same tolerance as (5.5) when $N = 8$ and Ψ is defined by the CG iteration with $\delta = 0.1$.

We can also use the diagonal preconditioner with diagonal block $\hat{A}$ and $\hat{C} = h^2 I$ (spectrally equivalent to C) to solve the system (5.2)-(5.3) by the minimum residual method. Unfortunately, its convergence seems slow. For example, 65 iterations are needed to reach the same tolerance when $N = 16$.

Our second example is a system of purely algebraic equations. We define the matrices $A = (a_{ij})_{n\times n}$ and $B = (b_{ij})_{n\times m}$ $(n \geq m)$ in (1.1) as follows:

$$a_{ij} = \begin{cases} i+1, \ i = j, \\ 1, \quad |i-j| = 1, \\ 0, \quad \text{otherwise}; \end{cases} \qquad b_{ij} = \begin{cases} 15j, \quad i = j+n-m, \\ 0, \quad \text{otherwise}. \end{cases}$$

The preconditioners $\hat{A} = (\overline{a}_{ij})_{n\times n}$ and $\hat{C} = (c_{ij})_{n\times m}$ are defined by

$$\overline{a}_{ij} = \begin{cases} i, \quad i = j, \\ 0, \quad \text{otherwise}; \end{cases} \qquad c_{ij} = \begin{cases} i^2+3, \quad i = j, \\ 0, \ \text{otherwise}. \end{cases}$$

The right-hand side vectors f and g are defined such that the exact solutions x and y are the vectors with all components being 1.

We solve the corresponding system (1.1) by Algorithm 3.1 with the preconditioner $\hat{C}$ and Ψ defined by N_c iterations of the CG method (without preconditioner) and the PCG method with the preconditioner $\hat{A}$. Algorithm 3.1 terminates when the error $\varepsilon \leq 1.0 \times 10^{-4}$, the resulting number of iterations is given in Table 2.

To compare Algorithm 3.1 with Algorithm 4.1 for this example, we solved the considered system again by Alg. 4.1. The numerical experiments

Table 2. Numbers of iteration (Algorithm 3.1)

	$n = 200, m = 150$	$n = 400, m = 300$	$n = 800, m = 600$
PCG($N_c = 2$)	18	18	19
CG($N_c = 6$)	18	19	20

Table 3. Number of iterations PCG(N_c=2) inner iteration, $\hat{C} = I$

(n,m)	Algorithm 3.1	Algorithm 4.1			
	k=1	k=2	k=5	k=10	k=20
(200,150)	297	138	93	47	20
(400,300)	254	163	94	40	23
(800,600)	364	147	78	38	21

Table 4. Number of iterations CG($N_c = 6$) inner iteration, $\hat{C} = I$

(n,m)	Algorithm 3.1	Algorithm 4.1			
	k=1	k=2	k=5	k=10	k=20
(200,150)	318	179	82	39	21
(400,300)	387	166	86	43	23
(800,600)	362	166	81	50	26

indicate that both algorithms have almost the same number of iterations with the preconditioner $\hat{C}$ and Ψ defined as above, even when the factor $\frac{1-\delta}{2}$ in (4.8) is replaced by $\frac{1}{2}$. So the detailed number of iterations is omitted here. To further compare the performance of both algorithms in the case with a very bad preconditioner $\hat{C}$, we take $\hat{C} = I$, the worst preconditioner for C, namely without any preconditioning. Tables 3 and 4 below listed the number of iterations in this case.

Tables 3 and 4 above indicate that Algorithm 4.1 is more efficient in the case with a very bad preconditioner $\hat{C}$ for the Schur complement C.

We also tried to use Algorithm 2.1 to solve the considered system, but the numerical experiments indicate that this algorithm is divergent for this example. In fact, the assumption (2.5) is not satisfied with the given preconditioner $\hat{C}$. Also note that it is not easy to achieve a good estimate on the minimal generalized eigenvalue of the preconditioner $\hat{C}$ with respect to the matrix $B^{\mathrm{t}}A^{-1}B$ in this case (except that the action of A^{-1} is allowed), then there seems no direct way to solve the considered system using Algorithm 2.1.

Acknowledgements. The authors wish to thank the anonymous referees for many constructive comments which lead to the great improvement of the paper.

References

1. K. Arrow, L. Hurwicz, H. Uzawa: Studies in linear and nonlinear programming. Stanford University Press, Stanford, CA (1958)
2. O. Axelsson: Numerical algorithms for indefinite problems. in Elliptic Problem Solvers, Academic Press, New York, 219–232 (1984)
3. R. Bank, B. Welfert, H. Yserentant: A class of iterative methods for solving saddle point problems. Numer. Math. **56**, 645–666 (1990)
4. D. Braess: The convergence rate of a multigrid method with Gauss- Seidel relaxation for the Poission equation. In Multigrid Methods (W. Hackbusch, U. Trottenberg, Eds.), 369–386, 1982, Berlin: Springer-Verlag
5. J. Bramble, J. Pasciak, A. Vassilev: Analysis of the inexact Uzawa algorithm for saddle-point problems. SIAM J. Numer. Anal. **34**, 1072–1092 (1997)
6. J. Bramble, J. Pasciak: A preconditioning technique for indefinite systems resulting from mixied approximations of elliptic problems. Math. Comp. **50**, 1–18 (1988)
7. F. Brezzi, M. Fortin: Mixed and Hybrid Finite Element Methods. Springer-Verlag, New York, 1991
8. X. Chen: On preconditioned Uzawa methods and SOR methods for saddle-point problems. J. Comput. Appl. Math. **100**, 207–224 (1998)
9. X. Chen: Global and superlinear convergence of inexact Uzawa methods for saddle-point problems with nondifferentiable mappings. SIAM J. Numer. Anal. **35**, 1130–1148 (1998)
10. X.-L. Cheng: On the nonlinear inexact Uzawa algorithm for saddle-point problems. SIAM J. Numer. Anal. **37**, 1930–1934 (2000)
11. Z. Chen, Q. Du, J. Zou: Finite element methods with matching and non-matching meshes for Maxwell equations with discontinuous coefficients. SIAM J. Numer. Anal. **37**, 1542–1570 (2000)
12. Z. Chen, J. Zou: An augmented Lagrangian method for identifying discontinuous parameters in elliptic systems. SIAM J. Control Optim. **37**, 892–910 (1999)
13. P. Ciarlet: Basic error estimates for elliptic problems. In Handbook of Numerical Analysis, Volume II, P. Ciarlet, J.-L. Lions, eds., North Holland, (1991) 17–352
14. H. Elman, G. Golub: Inexact and preconditioned Uzawa algorithms for saddle point problems. SIAM J. Numer. Anal. **31**, 1645–1661 (1994)
15. V. Girault, P.-A. Raviart: Finite Element Methods for Navier–Stokes Equations. Springer–Verlag, Berlin, 1986
16. R. Glowinski, P. Le Tallec: Augmented Lagrangian and operator-splitting methods in nonlinear mechanics. SIAM, Philadelphia, 1989
17. Q. Hu, G. Liang, J. Liu: Construction of a preconditioner for domain decomposition methods with polynomial Lagrangian multipliers. J. Comp. Math. **19**(2001), 213–224
18. Q. Hu, J. Zou, On nonlinear ineaxct Uzawa algorithms for saddle-point problems: Research Report MATH-1999-31(199). Department of Mathematics, The Chinese University of Hong Kong
19. Y. Keung, J. Zou: An efficient linear solver for nonlinear parameter identification problems. SIAM J. Sci. Comput. **22**, 1511–1526 (2000)
20. A. Klawonn: An optimal preconditioner for a class of saddle point problems with a penalty term. SIAM J. Sci. Comput. **19**, 540–552 (1998)
21. W. Queck: The convergence factor of preconditioned algorithms of the Arrow-Hurwicz type. SIAM J. Numer. Anal. **26**, 1016–1030 (1989)
22. T. Rusten, R. Winther: A preconditioned iterative method for saddlepoint problems. SIAM J. Matrix Anal. Appl. **13**, 887–904 (1992)
23. R. Verfürth: The contraction number of a multigrid method. Z. angew. Math. Mech. **63**, T386–T387 (1983)