A stable algorithm is presented to solve a nonsingular bordered system of the form

$$\begin{pmatrix} A & B \\ C^T & D \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}$$

where B and C are n by m matrices and the n by n matrix A could be nearly singular with at most μ small singular values. The algorithm needs only a solver for A and the solution to an $m+\mu$ by $m+\mu$ dense linear system. It is, thus, well suited for problems for which A has easily exploitable structures and $m+\mu \ll n$, such as in continuation methods, bifurcation problems and constrained optimization.

Generalized Deflated Block-Elimination

Tony F. Chan and Diana C. Resasco †

Research Report YALEU/DCS/RR-337
February 1985

The authors were supported in part by the Department of Energy under contract DE-AC02-81ER10996, by the Army Research Office under contract DAAG-83-0177, and by a BID-CONICET fellowship from Argentina.

Keywords: *bordered systems, block-elimination, nearly singular matrices, deflated solution.*

†Dept. of Computer Science, Yale Univ., Box 2158, Yale Station, New Haven, Conn. 06520.

1. Introduction

We are interested in solving a system of the form:

$$M\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} A & B \\ C^T & D \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \tag{1.1}$$

where the n by n matrix A may be singular or nearly singular with at most μ small singular values, but the n by m matrices B and C and the m by m matrix D are such that M is nonsingular and well conditioned. We shall show later that this implies that m must not be less than μ. Systems such as (1.1) arise, for example, in continuation methods, bifurcation problems [3] and constrained optimization [2, 7], where usually $m+\mu \ll n$. The case $m = 1, \mu = 1$ is rather common, especially in continuation methods. However, in applications like computation of singular points by augmented systems [3, 8, 9], m is often larger than 1 and in constrained optimization, the nullity of A can also be larger than 1.

In many applications, A possesses certain properties, such as sparseness or the existence of a special solver, which can be exploited. In these situations, we want an algorithm to solve (1.1) that involves primarily using solvers for A. The following block-elimination algorithm has this property:

ALGORITHM BE.

Step 1: Find the n by m matrix V and the n-vector w that solve the n by n systems:

$$AV = B \quad \text{and} \quad Aw = f \tag{1.2}$$

Step 2: Compute the Schur complement of A in M:

$$S = D - C^T V$$

and solve the m by m system

$$Sy = g - C^T w \tag{1.3}$$

Step 3: Compute

$$x = w - Vy \tag{1.4}$$

This algorithm is well defined if both A and M are nonsingular, since in that case, it is easy to show that S is nonsingular (see section 2). It, however, can be numerically unstable when A is nearly singular and it can produce completely inaccurate solutions (x, y) in those situations [3, 6].

For the case $m = 1$ and $\mu \leq 1$, Algorithm BE can be rendered stable by employing ***implicit deflation*** techniques [6] to give the Deflated Block-Elimination (DBE) Algorithm which exploits structures in A while giving accurate solutions. In this paper, we consider generalizations of Algorithm DBE to the case $m > 1$ and $\mu > 1$. In section 3 we review the basic deflation techniques developed in [4, 11] and extend them to the case of higher dimensional null space. These techniques are used to compute accurate representations for the solutions V and w in (1.2). In section 4 we review Algorithm DBE for the case $m = 1$, $\mu = 1$, which is then used in section 5 to motivate the generalization to m and μ greater than one.

We mention a class of related methods that are also based on Algorithm BE [10, 8, 9]. Instead of employing implicit deflation techniques, they rely on computing an LU factorization of A with a small n-th pivot, which can be computed, for example, by the algorithm described in [5]. However, even when A is nearly singular, the usual pivoting strategies (e.g. partial and complete pivoting) are not guaranteed to produce any small pivot and the row and column permutations needed to produce such a factorization may not preserve the sparsity of the LU-factors. Moreover, the deflation techniques used here can be extended to iterative methods, such as multigrid [1]. For a survey of other methods for solving (1.1) see [3].

We shall use only the 2-norm in this paper and P_u with $\|u\| = 1$ will denote the orthogonal projector $I - uu^T$.

2. Nonsingularity of M

In this section we will give necessary and sufficient conditions on A, B, C^T and D for M to be nonsingular. These conditions are derived from a block factorization of M based on the singular value decomposition of A, and will later turn out to be useful in establishing the stability of our algorithm. We shall first introduce some notation and definitions.

Definition 2.1. The singular value decomposition (SVD) of A is denoted by

$$A = U\Sigma V^T.$$

Since A has at most μ small singular values, we shall partition Σ as follows:

$$\Sigma = \begin{pmatrix} \Sigma_\mu & 0 \\ 0 & \Delta \end{pmatrix} \tag{2.1}$$

where

$$\Sigma_\mu = \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_{n-\mu} \end{pmatrix} \tag{2.2}$$

and

$$\Delta = \begin{pmatrix} \delta_1 & & \\ & \ddots & \\ & & \delta_\mu \end{pmatrix} \tag{2.3}$$

with $\sigma_1 \geq \ldots \sigma_{n-\mu} \geq \delta_1 \geq \ldots \delta_\mu \geq 0$. We shall use Φ and Ψ to denote the last μ columns of V and U, i.e. Φ and Ψ contain the right and left singular vectors corresponding to the μ smallest singular values of A. The matrices U_μ and V_μ will denote the first $n - \mu$ columns of U and V, respectively. Thus, the SVD of A can be written as:

$$A = \begin{pmatrix} U_\mu & \Psi \end{pmatrix} \begin{pmatrix} \Sigma_\mu & 0 \\ 0 & \Delta \end{pmatrix} \begin{pmatrix} V_\mu^T \\ \Phi^T \end{pmatrix} \tag{2.4}$$

The following theorem gives conditions for the nonsingularity of M.

Theorem 2.1. *Let the SVD of A be given by (2.4), where $\sigma_{n-\mu} > 0$. Then, M is nonsingular if and only if the matrix*

$$E = \begin{pmatrix} \Delta & \Psi^T B \\ C^T\Phi & D - C^T V_\mu \Sigma_\mu^{-1} U_\mu^T B \end{pmatrix}$$

is nonsingular.

Proof. Based on (2.4), define the $\mu + m$ by $n - \mu$ matrix Q as:

$$Q = \begin{pmatrix} 0 \\ C^T V_\mu \Sigma_\mu^{-1} \end{pmatrix},$$

where 0 denotes μ zero rows, and the $n-\mu$ by $\mu+m$ matrix H as:

$$H = (0 \quad U_\mu^T B).$$

It can be easily verified that the matrix M can be factored as follows

$$M = \begin{pmatrix} U & 0 \\ 0 & I_m \end{pmatrix} LR \begin{pmatrix} V^T & 0 \\ 0 & I_m \end{pmatrix}, \tag{2.5}$$

where

$$L = \begin{pmatrix} I_{n-\mu} & 0 \\ Q & I_{\mu+m} \end{pmatrix} \tag{2.6}$$

and

$$R = \begin{pmatrix} \Sigma_\mu & H \\ 0 & E \end{pmatrix}. \tag{2.7}$$

The matrix L is clearly nonsingular, therefore it is easy to see that M is nonsingular if and only if R is nonsingular. Since Σ_μ is diagonal with non-zero diagonal entries, R is nonsingular if and only if E is nonsingular.

∎

Corollary 2.1. *(a) Assume that A is nonsingular and let the m by m matrix S be the Schur complement:*

$$S \equiv D - C^T A^{-1} B \tag{2.8}$$

then M is nonsingular if and only if S is nonsingular.

(b) If A is singular with $\dim N(A) = \mu$, i.e. $\Delta \equiv 0$ and $\sigma_{n-\mu} \neq 0$, then M is nonsingular if and only if $m \geq \mu$ and the matrix

$$\begin{pmatrix} 0 & \Psi^T B \\ C^T \Phi & D - C^T V_\mu \Sigma_\mu^{-1} U_\mu^T B \end{pmatrix}$$

is nonsingular.

∎

3. Deflated Decomposition

Consider the system

$$Az = p \tag{3.1}$$

where the SVD of A is given by (2.4). If A is nonsingular, by applying the inverse of the expression (2.4), the solution z can always be represented as:

$$z = A^{-1} p = z_d + \Phi \Delta^{-1} \Psi^T p \tag{3.2}$$

where

$$z_d \equiv V_\mu \Sigma_\mu^{-1} U_\mu^T p \tag{3.3}$$

We call (3.2) the *deflated decomposition* of z. When μ is such that $\sigma_{n-\mu} \gg 0$, (3.2) can be interpreted as a decomposition of the solution into a deflated part, z_d, and a part spanned by approximate null vectors of A, since Φ contains the singular vectors corresponding to the μ smallest singular values of A. When A is singular, z_d is still well defined and $\|z_d\|$ remains bounded. We call z_d the *deflated* solution of (3.1). We will prove that z_d is the solution to a nearby singular but consistent system derived from (3.1). This new definition of z_d gives the basis for an algorithm to compute z_d that does not require explicitly computing the SVD of A.

Theorem 3.1. *The deflated solution z_d of (3.1) is the unique solution to the following system:*

$$P_\Psi A z_d = P_\Psi p \\ P_\Phi z_d = z_d. \tag{3.4}$$

Proof. Since U and V are orthogonal, it can be easily proved that

$$P_\Phi = V_1 V_1^T \tag{3.5}$$

and

$$P_\Psi = U_1 U_1^T. \tag{3.6}$$

By substituting (3.3) into (3.4), we can prove that z_d is a solution to (3.4). On the other hand, if u is a solution to (3.4), we have

$$U_1 U_1^T A u = U_1 U_1^T p$$

and by multiplying by $V_1 \Sigma_1^{-1} U_1^T$ we get $P_\Phi u = z_d$. Since $u = P_\Phi u$, we have $u = z_d$, thus proving uniqueness.

∎

The following algorithm [6, 11] for computing z_d is based on Theorem 3.1.

ALGORITHM DEFLATE

Step 1. Compute $\hat{p} = P_\Psi p$
Step 2. Solve $Ad = \hat{p}$
Step 3. Compute $z_d = P_\Phi d$.

Because of the deflation performed in Step 1, the size of the vector d computed in Step 2 is kept small and therefore Algorithm Deflate is stable even when A is nearly singular [4]. Step 3 is not essential. Its function is to purge z_d of any component in the Φ direction. Note that this algorithm only requires a solver for A and does not require computing the SVD of A. Moreover, either direct or iterative methods such as multigrid [1] can be used.

The following lemma will be needed later.

Lemma 3.1. *z_d is also a solution to*

$$A z_d = P_\Psi p$$

Proof. Follows from the definition (3.3) and (3.6).

∎

4. Deflated Block-Elimination

The Algorithm DBE was presented in [6] as a method for solving the system (1.1) when $\mu = 1$ and $m = 1$, i.e. B and C are vectors in R^n, D is a scalar, and the matrix A can be singular with nullity ≤ 1. The proposed method applies Algorithm BE combined with deflation techniques to find the deflated decomposition for the solution to the systems with the matrix A in (1.2) and it is made stable by avoiding the division by Δ. By replacing V and w by their deflated decompositions

$$w = w_d + \frac{\Psi^T f}{\Delta} \Phi$$

$$V = V_d + \frac{\Psi^T B}{\Delta}\Phi$$

in (1.3) and (1.4) we can derive that the solution to (1.1) is given by:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} w_d \\ 0 \end{pmatrix} + \begin{pmatrix} -V_d \\ I \end{pmatrix} \beta + \begin{pmatrix} \Phi \\ 0 \end{pmatrix} \alpha \tag{4.1}$$

where

$$\beta = D_1^{-1}\left[(C^T\Phi)\Psi^T f - \Delta(g - C^T w_d)\right] \tag{4.2}$$

and

$$\alpha = D_1^{-1}\left[(g - C^T w_d)\Psi^T B - (D - C^T V_d)\Psi^T f\right], \tag{4.3}$$

with

$$D_1 \equiv (C^T\Phi)\Psi^T B - \Delta(D - C^T V_d), \tag{4.4}$$

Note that D_1 is exactly the determinant of the matrix E in Theorem 2.1 and therefore is nonzero if M is nonsingular. The algorithm is proven stable in [6], where a backward error bound is derived.

5. Generalizing Deflated Block Elimination

In this section we consider the generalization of Algorithm DBE to $m \geq 1$ and $\mu \geq 1$. Let us first restrict our attention to the case $\mu = 1$. It might first appear that equations (4.1) – (4.4) generalize directly, with scalar divisions by D_1 replaced by matrix inversion. However, the m by m matrix D_1 (4.4) tends to a rank-one matrix when A tends to being singular, i.e. as Δ tends to zero, and this would produce very inaccurate results if the inverse of that matrix were to be applied directly in (4.2) and (4.3).

Since (4.4) is a rank-one modification of the matrix

$$S_d \equiv D - C^T V_d, \tag{5.1}$$

if S_d is nonsingular, the Sherman-Morrison formula could be applied to express the inverse of D_1 in terms of S_d^{-1} in (4.2) and (4.3), in the process cancelling out the singularity. Unfortunately, this method may fail because S_d could be singular. For example, let $n = m = 2$ and

$$M = \left[\begin{array}{cc|cc} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ \hline 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right], \tag{5.2}$$

then $\mathcal{K}(M) \approx 5$ and

$$S_d = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix},$$

which is clearly singular.

Another alternative could be considered, which consists of applying Algorithm DBE in a recursive way as follows. Consider a new splitting of the matrix M:

$$M = \begin{pmatrix} \tilde{A} & \tilde{B} \\ \tilde{C}^T & \tilde{D} \end{pmatrix}, \tag{5.3}$$

where $\tilde{A}$ is the matrix A augmented with the first column of B, the first row of C^T, and the (1,1)-th element of D, i.e. :

$$\tilde{A} = \begin{pmatrix} A & b_1 \\ c_1^T & d_{11} \end{pmatrix}.$$

If $\tilde{A}$ is non-singular, then Algorithm BE can be applied to solve (5.3). When a system with $\tilde{A}$ needs to be solved in Algorithm BE, Algorithm DBE is applied. In this way we have reduced the problem to the case $m = 1$, solved in section 3. However, this method fails if $\tilde{A}$ is singular as the following example shows.

Let $n = m = 2$ and

$$M = \left[\begin{array}{cc|cc} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{array}\right], \tag{5.4}$$

then $\mathcal{K}(M) \approx 5$ and A is singular. Since

$$\tilde{A} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

is also singular, the recursive algorithm just described cannot be applied.

We next derive a generalization to Algorithm DBE that works and is stable as long as M is nonsingular and well-conditioned.

Analogous to (4.1), we look for solutions to (1.1) of the form:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} w_d \\ 0 \end{pmatrix} + \begin{pmatrix} -V_d \\ I \end{pmatrix} \beta + \begin{pmatrix} \Phi \\ 0 \end{pmatrix} \alpha \tag{5.5}$$

with $\alpha \in R^{\mu}$ and $\beta \in R^m$, and V_d and w_d are the deflated solutions (3.3) to the systems $AV = B$ and $Aw = f$. By substituting (5.5) in (1.1), and using the relationships

$$A\Phi = \Psi\Delta,$$

$$AV_d = P_\Psi B, \quad Aw_d = P_\Psi f$$

(the last two from Lemma 3.1), it can be easily shown that (5.5) is a solution to (1.1) if the vectors α and β solve the following system:

$$E\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \Psi^T f \\ g - C^T w_d \end{pmatrix}, \tag{5.6}$$

where the $m + \mu$ by $m + \mu$ matrix E is given by

$$E \equiv \begin{pmatrix} \Delta & \Psi^T B \\ C^T\Phi & D - C^T V_d \end{pmatrix}. \tag{5.7}$$

From Theorem 2.1, E is nonsingular, and therefore (5.6) has a unique solution. Note that for the case $m = \mu = 1$, the solution (4.1) is of the form (5.5), where the 2 by 2 system (5.6) has been solved directly giving (4.2) and (4.3). Note that in the case $m + \mu > 2$ some form of pivoting should be used when solving (5.6). The expressions (5.5) and (5.6) can also be derived from the factorization (2.5) of M, with the solution obtained by simple backsubstitution. In this process, the expressions for the deflated solutions w_d and V_d corresponding to (3.3) naturally arise and can thus be computed by Algorithm Deflate, instead of using the SVD of A. This approach of deriving an algorithm for (1.1) through a factorization of M is similar in spirit to an algorithm derived in [8, 9], where an LU-factorization of M is derived from an LU-factorization of A with a small pivot [5]. Our approach can be viewed as a generalization of the algorithm in [8, 9], where instead of computing the LU-factorization of A, implicit deflation techniques are employed to fully exploit structures in A. For large and sparse problems, such as those arising from discretizations of partial differential equations, our approach should be more efficient. Moreover, it is more general because the deflated solutions can be computed by methods other than Gaussian Elimination.

We next show that E is well conditioned as long as M is well conditioned. Based on the factorization of M given in the proof of Theorem 2.1, an upper bound on the condition number of E can be derived in terms of the condition number of M.

Theorem 5.1. *The condition number of E is bounded by*

$$\mathcal{K}(E) \leq \mathcal{K}(M)\left(1+\frac{\|C\|}{\sigma_{n-\mu}}\right)^2$$

Proof. Consider the expression (2.5). For any lower triangular matrix of the form (2.6), it is easy to prove that

$$\|L\| \leq 1+\|Q\|$$

and similarly,

$$\|L^{-1}\| \leq 1+\|Q\|.$$

Therefore

$$\mathcal{K}(L) \leq (1+\|Q\|)^2 \leq \left(1+\frac{\|C\|}{\sigma_{n-\mu}}\right)^2 \tag{5.8}$$

Since

$$\|E\| \leq \|R\| \leq \|M\|\cdot\|L^{-1}\|$$

and

$$\|E^{-1}\| \leq \|R^{-1}\| \leq \|M^{-1}\|\cdot\|L\|$$

we have

$$\mathcal{K}(E) \leq \mathcal{K}(M)\mathcal{K}(L)$$

By applying (5.8) the proof is complete.

∎

The bound in Theorem 5.1 is independent of the size of $\|\Delta\|$ and thus E is well conditioned if M is, *regardless of whether A is nearly singular or not.* Observe also that there is a freedom in the choice of μ, in the sense that, as long as $\sigma_{n-\mu} \gg 0$, the algorithm is stable. The larger we choose μ, the smaller the bound on $\mathcal{K}(E)$ will be, but on the other hand, it would require more work in computing the singular vectors Φ and Ψ.

6. Error Analysis and practical considerations

Here we present a backward error bound for the residual

$$r_M \equiv M\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} f \\ g \end{pmatrix}.$$

If the matrix M is well conditioned and r_M is small, then the error in the solution is small. On the other hand, if M is ill conditioned, a small residual is all one can hope for.

We shall use tildes to denote computed quantities.

Theorem 6.1. *For any $\tilde{V}_d, \tilde{w}_d$, $\tilde{\Phi}, \tilde{\Psi}$, $\tilde{\Delta}$, $\tilde{E}$, $\tilde{\alpha}$ and $\tilde{\beta}$ that satisfy*

$$A\tilde{\Phi} - \tilde{\Psi}\tilde{\Delta} = R_\Delta, \tag{6.1}$$

$$A\tilde{V}_d - P_{\tilde{\Psi}}B = R_V, \tag{6.2}$$

$$A\tilde{w}_d - P_{\tilde{\Psi}}f = r_w \tag{6.3}$$

and

$$\tilde{E}\begin{pmatrix}\tilde{\alpha}\\ \tilde{\beta}\end{pmatrix} - \begin{pmatrix}\tilde{\Psi}^T f\\ g - C^T\tilde{w}_d\end{pmatrix} \equiv r_E = \begin{pmatrix}r_E^1\\ \\ r_E^2\end{pmatrix}, \tag{6.4}$$

the solution $\tilde{x}, \tilde{y}$ computed by (5.5) and (5.6) satisfies

$$r_M \equiv M\begin{pmatrix}\tilde{x}\\ \tilde{y}\end{pmatrix} - \begin{pmatrix}f\\ g\end{pmatrix} = \begin{pmatrix}r_w - R_V\tilde{\beta} + R_\Delta\tilde{\alpha}\\ 0\end{pmatrix} + \begin{pmatrix}\tilde{\Psi}r_E^2\\ \\ r_E^1\end{pmatrix} \tag{6.5}$$

This theorem shows that the stability and accuracy of this algorithm depend on keeping the residuals R_Δ, R_V, r_w and r_E small and the solution $\tilde{\alpha}, \tilde{\beta}$ to (5.6) bounded. In the rest of this section we will look at each residual individually.

In order to analyze R_Δ, we have to look at algorithms for finding the singular vectors Φ and Ψ. The following is a generalization of the inverse iteration algorithm presented in [4, 11] for higher dimension. At every iteration, the algorithm gives $\tilde{\Phi}, \tilde{\Psi}$ and $\tilde{\Delta}$ such that R_Δ is small.

```
ALGORITHM SII (Subspace inverse iteration)
Given $\Phi \in R^{n\times\mu}$ such that $\Phi^T\Phi = I$, repeat until convergence:
1-  Solve $A^T\Psi = \Phi$
2-  Compute the QR-factors $\Psi = QR$
3-  Set $\Psi = Q$
4-  Solve $A\Phi = \Psi$
5-  Compute the QR-factors $\Phi = QR$
6-  Set $\Phi = Q, \quad \Delta = R^{-1}$.
```

In general, when A is nearly singular, one or two iterations should be sufficient to get convergence. However, when A is not nearly singular or $\mu > \dim N(A)$, Algorithm SII may converge slowly. Since only a few iterations will be performed, Φ, Ψ and Δ may not be computed accurately. Although the computed $\tilde{\Delta}$ may not necessarily be diagonal and the relationship $A^T\tilde{\Psi} \approx \tilde{\Phi}\tilde{\Delta}$ may not hold, the computed $\tilde{\Phi}$ and $\tilde{\Psi}$ are still orthogonal and

$$A\tilde{\Phi} \approx \tilde{\Psi}\tilde{\Delta}, \tag{6.6}$$

i.e., R_Δ is small.

In Section 3 we presented Algorithm DEFLATE for computing the deflated solutions V_d and w_d. In order to keep the residuals R_V and r_w small, Step 3 is not necessary. In fact, with that step, small residuals cannot be guaranteed if the singular vectors Φ, Ψ and the matrix Δ of singular values are not accurate. As noted before, such situations can occur when SII terminates prematurely. On the other hand, it is clear that if Step 3 in Algorithm Deflate is skipped, the residuals R_V and r_w will be small, even when Φ, Ψ and Δ are not accurate, as long as $\tilde{V}_d$ and $\tilde{w}_d$ remain bounded. The computed deflated solutions will have some components in the directions of Φ, but they will be bounded. For this reason, we recommend that Step 3 be skipped in Algorithm DEFLATE, which also saves a few inner products.

Finally, let us look at r_E. In exact arithmetic, Theorem 5.1 shows that E is well conditioned, therefore the solution to the system (5.6) should remain bounded and the residual r_E should be small. However, it was mentioned above that Algorithms SII and DEFLATE without the correction Step 3 might yield answers that keep the corresponding residuals small but at the cost of changing the entries of E in a nontrivial way. If the computed matrix $\tilde{E}$ turns out to be well conditioned, then $\tilde{\alpha}$ and $\tilde{\beta}$ are bounded and r_E is small. When Ψ and Φ are computed exactly and Algorithm DEFLATE without Step 3 is applied, the computed $\tilde{V}_d$ will have a component in each direction of Φ that will be proportional to the ratio between the machine precision ϵ_M and the corresponding singular value, i.e. $\tilde{V}_d$ satisfies

$$\tilde{V}_d = V_d + \Phi u, \qquad (6.7)$$

where $\|\Delta u\| = \mathcal{O}(\epsilon_M)$. The computed $\tilde{E}$ can thus be written as

$$\tilde{E} = E\begin{pmatrix} I & -u \\ 0 & I \end{pmatrix} + \mathcal{O}(\epsilon_M),$$

therefore,

$$\mathcal{K}(\tilde{E}) \approx \mathcal{K}(E)\,(1 + \|u\|)^2 .$$

If $\|\Delta^{-1}\| \leq \epsilon_M^{-1}$, then u is bounded and, by Theorem 5.1, $\tilde{E}$ is well conditioned when M is well conditioned.

We summarize our results, giving the outline for the algorithm and a rough estimate of work and storage.

ALGORITHM GDBE (Generalized Deflated Block-Elimination)

Step 1 -Compute the n by μ matrices of singular vectors Φ and Ψ and the μ by μ matrix Δ of singular values, e.g. by two iterations of Algorithm SII.

Step 2 -Compute the deflated solutions V_d and w_d as follows:

- Compute $\Psi^T B$ and $\Psi^T f$.
- Compute $P_\Psi B = B - \Psi(\Psi^T B)$ and $P_\Psi f = f - \Psi(\Psi^T f)$.
- Solve

$$AV_d = P_\Psi B$$

and

$$Aw_d = P_\Psi f.$$

Step 3- Form the $m+\mu$ by $m+\mu$ matrix

$$E = \begin{pmatrix} \Delta & \Psi^T B \\ C^T\Phi & D - C^T V_d \end{pmatrix}.$$

Step 4- Solve the dense system

$$E\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \Psi^T f \\ g - C^T w_d \end{pmatrix}$$

Step 5- Compute the solution:

$$x = w_d - V_d\beta + \Phi\alpha$$
$$y = \beta$$

The algorithm requires the factorization of the matrix A, $m+1$ back-solves for V_d and w_d and 2μ back-solves per inverse iteration for computing Ψ and Φ, plus lower order terms that include the solution to a dense $m+\mu$ by $m+\mu$ system and a few inner products. It is very efficient for multiple right hand sides, since only backsolves with the factored matrices A and E are necessary.

In addition to A, B, C^T, D and the right hand side, the algorithm requires $2n\mu$ storage for Ψ and Φ, plus $\mathcal{O}(m\mu)$ extra storage for solving the E system.

7. Numerical Results

We performed some numerical tests with matrices of the form

$$A = (I - 2uu^T)Diag(n-1,\ldots,1,\delta)(I - 2vv^T) \tag{7.1}$$

where u and v are random vectors of norm 1 and $\delta = 10^{-i}$ with i varying from 1 to 8. The matrices B, C and D are chosen randomly but such that M is well conditioned. The computations were performed on a VAX-780, with a 27-bit mantissa. LINPACK routines were used to solve linear systems and compute the QR factorizations for the inverse iteration. Algorithm GDBE was tested and compared with Gaussian Elimination on the matrix M. Two iterations of Algorithm SII were used to estimate Φ, Ψ and Δ. It was also verified that, when step 3 of Algorithm DEFLATE is included, inaccurate results can be obtained if the singular vectors are not computed accurately.

In fig .1 , the relative error in the solution is plotted versus i for the case $m = 2$ and $\mu = 1$. While the Block-Elimination Algorithm gives an error that increases as the matrix A becomes more singular, Algorithm GDBE stays stable, giving an error that is comparable to applying regular Gaussian Elimination on the matrix M.

In fig .2, the same example is considered. Here the dimension of the null space of A was overestimated to $\mu = 2$, so that

$$\Delta = \begin{pmatrix} 1 & 0 \\ 0 & \delta \end{pmatrix}.$$

In this case, Algorithm SII does not converge to the exact singular vectors corresponding to the singular value 1. Nevertheless, Algorithm GDBE is still stable because the residual (6.1) remains small. This confirms that the estimation of the exact value of dim $N(A)$ is not critical for the success of Algorithm GDBE, as long as $\mu \geq$ dim $N(A)$. Finally, if Step 3 in Algorithm Deflate is not skipped, this makes (6.1) large, giving completely inaccurate results.

Acknowledgments. The authors would like to thank Prof. Allan Jepson, of the University of Toronto, for pointing out the relationship of his work [8] to the approach taken in [6] which greatly improves this paper.

References

[1] R.E. Bank and T.F. Chan, *PLTMGC: A Multigrid-Continuation Program for Parameterized Nonlinear Elliptic Systems,* Technical Report 261, Dept. of Computer Science, Yale Univeristy, 1983. Submitted to Siam J. Sci. Stat. Comp..

[2] T.F. Chan, *An Efficient Modular Algorithm for Coupled Nonlinear Systems,* Technical Report YALEU/DCS/RR-328, Yale Computer Science Department, 1984.

[3] ———, Techniques for Large Sparse Systems Arising from Continuation Methods, T. Kupper, H. Mittelmann and H. Weber eds., *Numerical Methods for Bifurcation Problems,* International Series of Numerical Math., Vol. 70, Birkhauser Verlag, Basel, 1984, pp. 116–128.

[4] ———, *Deflated Decomposition of Solutions of Nearly Singular Systems,* Siam J. Numer. Anal., 1984, 21/4 August (1984), pp. 738–754.

[5] ———, *On the Existence and Computation of LU-factorizations with Small Pivots,* Math. Comp., 42/166 April (1984).

[6] ———, *Deflation Techniques and Block-Elimination Algorithms for Solving Bordered Singular Systems,* Siam J. Sci. Stat. Comp., 5/1 March (1984).

[7] P.E. Gill, W. Murray and M. Wright, *Practical Optimization,* Academic Press, New York, 1981.

[8] A. Jepson and A. Spence, Singular Points and Their Computations, T. Kupper, H. Mittelmann and H. Weber eds., *Numerical Methods for Bifurcation Problems,* International Series of Numerical Math., Vol. 70, Birkhauser Verlag, Basel, 1984, pp. 195–209.

[9] ———, *Folds in Solutions of two-parameter Systems, Part I,* SIAM J. of Numer. Anal., (1985). to appear.

[10] H.B. Keller, *The Bordering Algorithm and Path Following Near Singular Points of Higher Nullity,* SIAM J. Sci. and Stat. Comp., 4/4 (1983).

[11] G.W. Stewart, *On the Implicit Deflation of Nearly Singular Systems of Linear Equations,* SIAM J. Sci. Stat. Comp., 2/2 (1981), pp. 136–140.

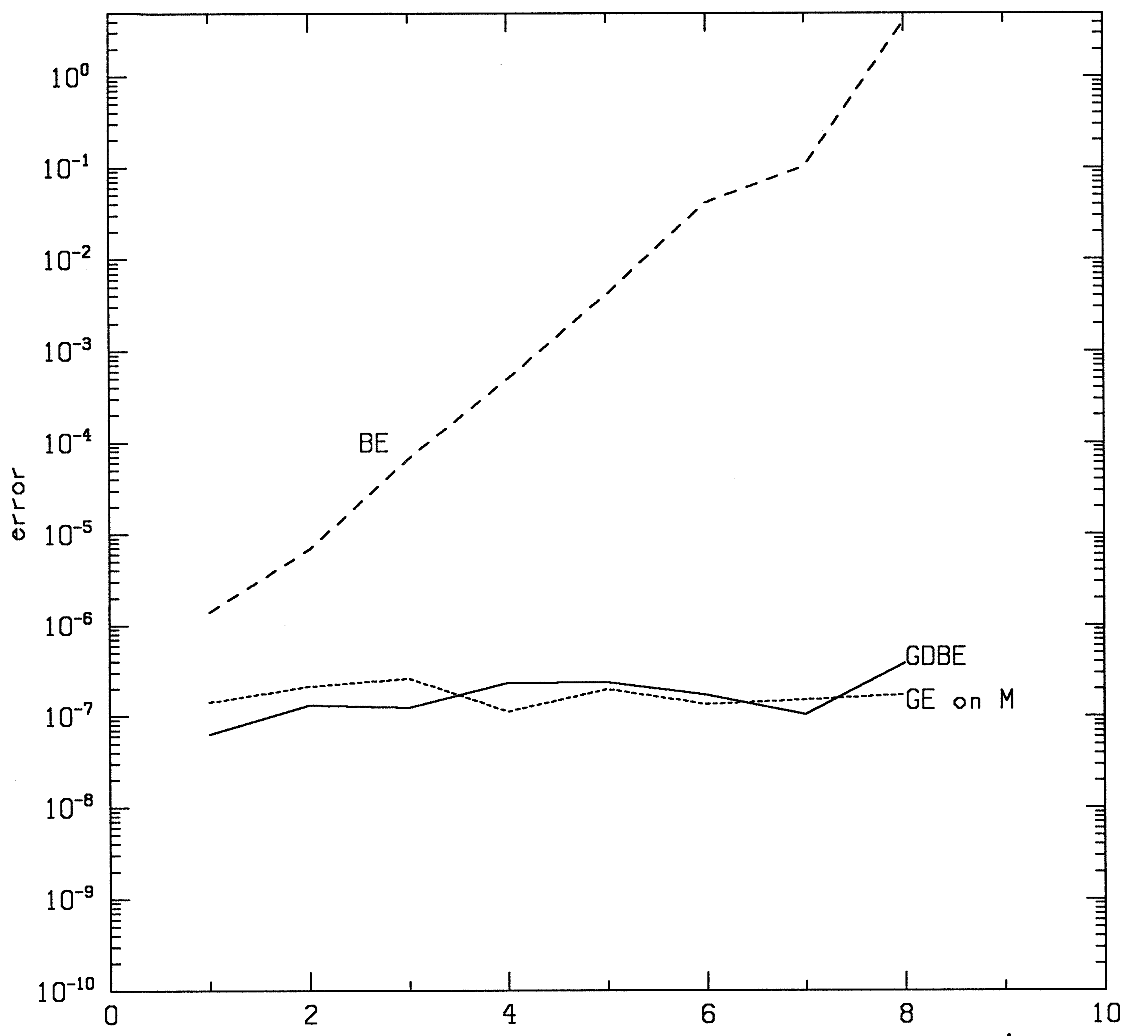

Fig.1 : Relative Error in BE and GDBE. $\sigma = 10^{-i}$

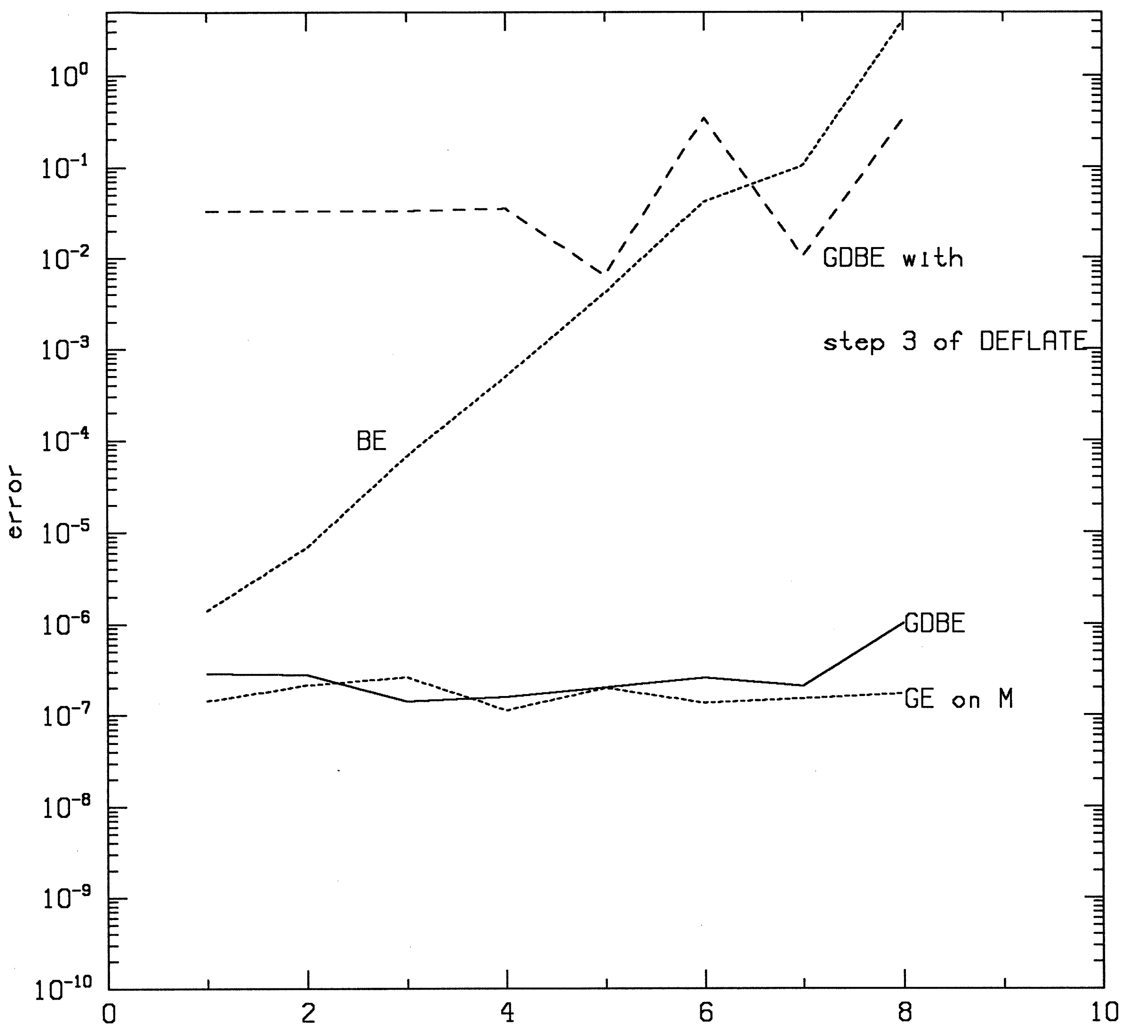

Fig.2 : Relative Error in BE and GDBE. $\sigma_i = 10^{-i}$