DEFLATION TECHNIQUES AND BLOCK-ELIMINATION ALGORITHMS FOR SOLVING BORDERED SINGULAR SYSTEMS

Tony F. Chan

Technical Report # 226
March 10, 1982

Computer Science Department, Yale University, Yale Station, New Haven, CT 06520.
This work was supported by the Department of Energy under grant DE-AC02-81ER10996.

ABSTRACT

In numerical continuation methods for solving nonlinear systems, one often has to solve linear systems with matrices of the following form:

$$M = \begin{bmatrix} A & b \\ c^T & d \end{bmatrix}$$

where A may become singular but M is well-conditioned and therefore direct Gaussian Elimination on M with some form of pivoting is stable. However, if A has special structures (e.g. sparseness, special data structure) or if a special solver for A is available, then an often used method for solving such systems is the block-elimination (BE) algorithm which involves solving two systems with A for each system with M. In this paper, we shall show that the BE algorithm may be inaccurate when A is nearly singular. We then propose a stable variant of the BE algorithm which employs deflation techniques when solving the two systems with A. The deflation techniques can be viewed as working in coordinate systems orthogonal to the approximate null vectors of A, enabling an accurate representation of the solution to be computed. The extra work amounts to a few more backsolves with A. Numerical results will be presented.

1. Introduction

In numerical continuation methods for solving nonlinear eigenvalue problems [11, 12, 13, 14] and in homotopy continuation methods for solving general nonlinear systems [2, 8], the central computational problem often reduces to solving linear systems of the form:

$$M \begin{vmatrix} x \\ y \end{vmatrix} = \begin{vmatrix} A & b \\ c^T & d \end{vmatrix} \begin{vmatrix} x \\ y \end{vmatrix} = \begin{vmatrix} f \\ g \end{vmatrix} \tag{1}$$

where the n by n matrix A may become singular[2] at certain points on the solution paths but the vectors b and c are chosen so that M remains nonsingular and well-conditioned. The following lemma gives necessary and sufficient conditions for M to be nonsingular.

Lemma 1: (1) If A is nonsingular, then M is nonsingular iff:

$$d - c^T A^{-1} b \neq 0. \tag{2}$$

(2) If A is singular and has a one dimensional null space represented by a left null vector ξ and a right null vector ϕ, then M is nonsingular iff

$$\xi^T b \neq 0, \tag{3}$$

and

$$c^T \phi \neq 0. \tag{4}$$

Proof: Straight-forward. A more general version can be found

---

[2] We shall assume the nullity of A to be one, which is the most common case in applications. The algorithms generalize easily to higher dimensional null spaces but we shall not discuss that here.

in [11]. The version given above is more suitable for our discussion.

Since M is assumed to be well-conditioned, the use of Gaussian Elimination on M with some form of pivoting is guaranteed to be stable. However, this approach is only suitable when n is small or when A is dense since the whole matrix M has to be stored to allow for fill-ins. When A is large but has special structures (e.g. sparseness, band or profile structures) or when a special solver is available for A (e.g. fast elliptic solvers, sparse matrix solvers, band or profile solvers), or when a solver for A is needed for some other purposes anyway, it is natural to consider algorithms for solving systems with M which only involve solving systems with A. The following block elimination algorithm has this desirable property:

Algorithm BE: [4, 11]

(1) Solve $A v = b,$ (5)

$A w = f.$ (6)

(2) Compute $y = (g - c^T w) / (d - c^T v).$ (7)

(3) Compute $x = w - y v.$ (8)

The work consists mainly of one factorization of A and two backsolve with the LU factors of A. If there are many right hand sides with the same matrix M, then the factorization and the vector v can be computed once, and the work reduces to only one backsolve for each right hand side, which makes the BE algorithm extremely attractive in such cases. These situations arise, for example, in continuation methods where Chord-Newton type methods are used.

Algorithm BE is well-defined if A and M are nonsingular because the denominator in (7) is nonzero by Lemma 1. However, in Section 2, we show that

Algorithm BE maybe unstable numerically when A is nearly singular and can produce completely inaccurate solutions (x, y) in those situations. The main source of instability is in Step (1) of Algorithm BE where the vectors v and w are computed inaccurately when A is nearly singular. In Section 3, we review implicit deflation techniques developed in [3, 16] which can be used to compute accurate representations for the solutions v and w. These deflation techniques can be viewed as working in subspaces orthogonal to approximate null vectors of A and are implicit in the sense that they only involve solving systems with A. In Section 4, we show how to use these deflated decompositions of v and w to obtain a stable variant of the BE algorithm. Further, we show that the new algorithm can be used to obtain a stable deflated decomposition of the solution (x, y) when M itself is nearly singular, for example, in applications to continuation around bifurcation points [1, 11, 13, 14]. We present a backward error analysis in Section 5 that shows that the stability of the new algorithm is independent of the singularity of A. This means that in practice only one technique is needed to solve with the matrix M independent of whether A is singular or not. Numerical tests demonstrating the accuracy and stability of the new algorithm will be presented in Section 6.

Rheinboldt [15] has considered an interesting related algorithm for solving the system (1). In his applications, the vector c is always equal to a unit vector, d is always equal to zero but the vector b is general. He considered splittings of M of the form $M = M_0 + uz^T$ where u and z are chosen so that M_0 is nonsingular and it is easy to obtain a factorization for M_0 (actually for a reduced matrix of smaller dimension than M_0). The solutions (x, y) can then obtained through the use of the Sherman-Morrison formula by two backsolves with M_0. However, his approach requires explicitly working with the storage

structure of A in order to obtain the factorization of the reduced matrix whereas our approach is completely implicit in that it only requires the ability to solve systems with A. Moreover, our approach works for general b, c and d as long they satisfy the conditions in Lemma 1 so that M is nonsingular. However, it should be pointed out that Rheinboldt's algorithm can be generalized to handle this more general case. A rank two modification is required and one more backsolve is involved.

2. Stability of Algorithm BE

In this section, we show that Algorithm BE may produce inaccurate solutions when A is nearly singular even though M is well-conditioned.

Lemma 2: If we use vectors $\tilde{v}$ and $\tilde{w}$ satisfying $A\tilde{v} - b = r_1$ and $A\tilde{w} - f = r_2$ in Steps (2) and (3) of Algorithm BE, then the solutions $(\tilde{x}, \tilde{y})$ satisfy:

$$M \begin{vmatrix} \tilde{x} \\ \tilde{y} \end{vmatrix} = \begin{vmatrix} r_2 - \tilde{y} r_1 \\ 0 \end{vmatrix} . \tag{9}$$

Proof: Straight-forward.

In other words, the computed solutions $(\tilde{x}, \tilde{y})$ always satisfy the last equation of (1) independent of their accuracy whereas the residual for the first n equations in (1) depends on the accuracy of v and w.

Next we show that when A is nearly singular, r_1 and r_2 are generally large. From (3), we see that $b \notin$ Range(A) and therefore the computed $\tilde{v}$ will be large when A is nearly singular. Similarly, $\tilde{w}$ will also be large unless $f \in$ Range(A). From standard round-off error analysis [6], it follows that the residuals r_1 and r_2 will be large. Moreover, these residuals will not cancel out in (9).

Specifically, consider solving the system $Az = p$ where A is nearly singular but p is not consistent with A. Let $\{\sigma_1,.....,\sigma_n\}$ be the singular values of A arranged in descending magnitude, $\{u_1,...,u_n\}$ be the corresponding left singular vectors and $\{v_1,...,v_n\}$ be the corresponding right singular vectors. Then the solution z can be written as:

$$z = \sum_{i=1}^{n} c_i \, v_i, \tag{10}$$

where

$$c_i = (u_i^T p \; / \; \sigma_i). \tag{11}$$

Since p is not consistent with A, $u_n^T p$ is not small and therefore the last term in the sum in (10) will be large if σ_n is small. In finite precision arithmetic, this last term will dominate the rest of the sum and the <u>computed</u> $\tilde{z}$ will have an expansion similar to (10) but where the first (n-1) coefficients $\tilde{c}_i$ are inaccurate. Since the residual for $\tilde{z}$ can be written as

$$r(\tilde{z}) = (A\tilde{z} - p) = \sum_{i=1}^{n} (c_i - \tilde{c}_i) \, u_i, \tag{12}$$

we see that the residual corresponding to the first (n-1) terms in (12) will be large. Furthermore, this part of the residual depends on the particular values of the coefficients c_i's. Therefore, we cannot expect this part of the residuals r_1 and r_2 in (9) to cancel out. It follows from Lemma 2 that the computed solutions $(\tilde{x}, \tilde{y})$ will give large residuals for the system (1). Since M is assumed to be well-conditioned, it follows that the errors in $(\tilde{x}, \tilde{y})$ will generally be large. As we shall see in Section 6, this actually happens numerically.

3. Deflation

The implicit deflation techniques that we are going to discuss here were developed in [3, 16]. They can be viewed as techniques for separating the subspaces corresponding to σ_n from its orthogonal complement, and only need the ability to solve systems with A. Explicit deflation techniques [3, 10, 9], which achieve the same goal by working with parts of the LU factorization of A explicitly, can also be used when they are applicable.

We consider computing a <u>deflated decomposition</u> of the solution z of the system $Az = p$ of the form

$$z = z_D + (c_p / \delta)\, \phi, \tag{13}$$

where z_d is the <u>unique</u> solution to the following system:

$$A_S\, z_D \equiv S\, A\, z_D = R\, p, \tag{14}$$

$$N\, z_D = z_D, \tag{15}$$

where A_S is a singular matrix 'close' to A with a left null vector ξ and a right null vector ϕ. The matrices R and N are defined in terms of ξ and ϕ and are chosen so that the system (14) and (15) is consistent and has a unique solution that remains bounded as A tends to being exactly singular. The coefficient c_p is a scalar that depends on p and the scalar δ tends to zero as A tends to being singular. In [3], we considered two different classes of such deflated decompositions, one corresponding to choosing A_S to be the nearest singular matrix to A in the Frobenius norm, and the other corresponding to choosing A_S to be a singular matrix obtained by perturbing some elements of A by amounts bounded by the smallest pivot, say ε, in a LU-factorization of A. For the LU-based approach to be successful, we need to make the assumption that $\varepsilon = O(\sigma_n)$, which is definitely not valid in general but which we showed empirically

here and in [3] and theoretically in [5] to be valid in practice.

Before we specify ξ, ϕ, S, R, N and δ defining the deflated solutions, we need some definitions.

Definition 3: (a) Let $P_u = I - u\,u^T$ be the orthogonal projector with respect to a given vector u with $||u|| = 1$.[3]

(b) For any vector u with $u_j \neq 0$ and $1 \leq j \leq n$, define

$$E_u^j = I - u\,e_j^T / u_j.$$

(c) Let ξ_{SV}, with norm equal to one, be an approximation to the left singular vector corresponding to the smallest singular vector σ_n of A. Define $\phi_{SV} = \sigma A^{-1}\xi_{SV}$, where $\sigma = 1 / ||A^{-1}\xi_{SV}||$.

(d) $$\xi_{LU} = \alpha A^{-T} e_k, \tag{16}$$

where $\alpha = 1 / || A^{-T} e_k ||$, (17)

and $1 \leq k \leq n$ is the index of the smallest pivot in the LU-factorization of A.

(e) $$\phi_P = \gamma A^{-1} \xi_{LU}, \tag{18}$$

where $\gamma = 1 / || A^{-1} \xi_{LU} ||$. (19)

(f) $$\phi_E = \beta A^{-1} e_j, \tag{20}$$

where $\beta = 1 / || A^{-1} e_j ||$, (21)

---

[3] All norms used are the Euclidean norm.

and j is chosen so that $|(\xi_{LU})_j| = O(1)$.[4]

The computation of ξ_{LU}, ϕ_E and ϕ_P costs one back-substitution each. For ξ_{SV}, a variant of the inverse power method can be used, which is fast when the smallest singular value of A is well-isolated. When such is not the case, the inverse power method may have difficulty in convergence.

In [3], we discussed eight deflated decompositions based on the LU-factorization and three based on the Singular Value Decomposition. We shall only use a subset of these here since there exist simple relationships among the deflated solutions and the ones used here are representative and can be computed most efficiently. The algorithms developed here can be applied to the other deflated decompositions as well.

In Table 3-1, we give the expressions for the corresponding ξ, ϕ, S, R, N and δ. It was shown in [3] that for the z_S deflation, the coefficient c_p reduces to $\xi_{SV}^T p$ if ξ_{SV} is exactly equal to the left singular vector corresponding to σ_n. The more general form in the table has to be used when the inverse iteration fails to converge or converges to the wrong singular value. We also gave the following algorithm, derived from one first proposed by Stewart [16], for stably computing the deflated solutions:

Algorithm IIA:

---

[4] We shall use the notation $(u)_k$ to denote the k-th component of the vector u.

Table 3-1: Deflated Decompositions

z_D	ξ	ϕ	$S = R$	N	δ	c_p
z_S	ξ_{SV}	ϕ_{SV}	$P_{\xi_{SV}}$	$P_{\phi_{SV}}$	σ	$\xi_{SV}^T(p - Az_S)$
z_E	ξ_{LU}	ϕ_E	$(E^j_{\xi_{LU}})^T$	$E^k_{\phi_E}$	$\beta(\xi_{LU})_j$	$\xi_{LU}^T p$
z_P	ξ_{LU}	ϕ_P	$P_{\xi_{LU}}$	$E^k_{\phi_P}$	γ	$\xi_{LU}^T p$

Start with z such that N z =‹ z.

Loop

(1) Form r =‹ R p - S A z.

(2) Form d =‹ A^{-1} r.

(3) z <- z + N d.

In [3], we analyzed the convergence and stability of Algorithm IIA, and showed that for z_S, z_E and z_P, no iteration is necessary. We then proposed the following non-iterative algorithm for computing z_D:

Algorithm NIA:

(1) r =‹ R p.

(2) d =‹ A^{-1} r.

(3) z =‹ N d.

Since the vector r in Step (1) of both Algorithm IIA and Algorithm NIA are consistent with A_S which is 'close' to A, the first (n-1) coefficients in the singular vector expansion of z_D will not be dominated by the last coefficient, and therefore they can be computed accurately. Therefore, the deflated decomposition (13) can be viewed as an accurate representation of the solution z. It was shown in [3] that the scalar σ (Definition 3, part c) tends to the smallest singular value σ_n as A tends to being singular and that α, β and γ are $O(\varepsilon)$. Thus we see from Table 3-1 that δ goes to zero as A becomes singular. If we were to compute z in (13) directly, then the last term will dominate the first and the accuracy in z_D will be lost.

4. Deflated Block Elimination

Recall that the reason Algorithm BE becomes unstable when A is nearly

singular is that the computed $\tilde{v}$ and $\tilde{w}$ have large relative errors in the subspace orthogonal to ϕ. The deflation techniques discussed in Section 3 overcome this problem by computing z_D with low relative errors in the same subspace. In this section, we show how to use the deflation techiques to obtain a stable variant of Algorithm BE.

The main idea is to compute the deflated decompositions of v and w instead of computing them directly from (5) and (6). By using Algorithm NIA (or Algorithm IIA if necessary), we can obtain the following deflated decompositions for v and w:

$$v = v_D + (c_b / \delta)\,\phi, \tag{22}$$

amd

$$w = w_D + (c_f / \delta)\,\phi. \tag{23}$$

When A is nearly singular, one wants to avoid actually carrying out the divisions by δ and the additions in the above formulas because the second terms will be large and will overwhelm the first terms, causing a loss of accuracy. It turns out that it is not difficult to derive a stable variant of Algorithm BE that uses the representations of v and w in (22) and (23) but which does not involve adding large vectors to the accurate deflated solutions v_D and w_D.

Lemma 4: If v and w are represented by (22) and (23), then the solutions (x , y) of (1) can be expressed as:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} w_D \\ 0 \end{bmatrix} + \frac{1}{D} \begin{bmatrix} h_3\,\phi - h_4\,v_D \\ h_4 \end{bmatrix}, \tag{24}$$

where

$$h_1 = g - c^T w_D,$$

$$h_2 = d - c^T v_D,$$

$$h_3 = h_1 c_b - h_2 c_f,$$

$$h_4 = (c^T\phi)\, c_f - \delta h_1,$$
$$D = (c^T\phi)\, c_b - \delta h_2.$$

Moreover, D is nonzero if M is nonsingular.

Proof: The proof that (x, y) given by (24) satisfy (1) can be derived by direct substitution. We shall only show that D is nonzero if M is nonsingular. When A is singular, $\delta = 0$ and from Lemma 1, both $c^T\phi$ and $\xi^T b$ are nonzero. For the z_E and z_P deflations, $c_b = \xi^T b$. For the z_S deflation, $c_b = \xi^T(b - Az_S) = \xi^T b$ because ξ is then a left null vector of A. Therefore, for all three deflations, we have $D = (c^T\phi)(\xi^T b) \neq 0$. When A is not singular ($\delta \neq 0$), it can be easily shown from (22) that $D = \delta(d - c^T A^{-1} b)$ and therefore by Lemma 1, $D \neq 0$.

We shall call the algorithm represented by (24) Algorithm DBE.

The expressions defining h_1, h_2, h_3 and h_4 are all stable formulas in the sense that no large vectors are involved. Note also from (24) that the vectors v_D and w_D generally have as much weight in the solutions (x, y) as the vector ϕ, and therefore the accuracy of (x, y) depends directly on the accuracy of v_D and w_D. Moreover, one can also see from the formulas involved that when δ is small (i.e. when A is nearly singular), it is enough to control the <u>absolute</u> error in δ. This is important because in general we cannot hope to be able to do better than this in computing δ. Furthermore, the stability of Algorithm DBE depends only on the singularity of M in the sense that $|D|$ is as small as M is singular, and is independent of the singularity of A. This means that in practice only one algorithm is needed for dealing with solving (1). We shall prove all the above assertions rigorously in the next section.

There is a reason why we use the particular form (24) for expressing the solutions (x, y). The reason is that as M itself tends to being singular, D tends to zero, and (24) automatically becomes a deflated decomposition of (x, y). Moreover, the vector multiplying (1 / D) in (24) is then a null vector of M. In practice, one can monitor the size of D and avoid performing the division by D and the addition to the first vector when |D| becomes too small. The form (24) will then remain an accurate representation of the solutions (x, y) and can be used in further computations just like what we have done here for the deflated decompositions of v and w. Such situations arise, for example, in applying continuation methods around bifurcation points [11], where A is singular but $\xi^T b = 0$ and therefore M is also singular.

Compared to Algorithm BE, the extra overhead involved in Algorithm DBE, with Algorithm NIA for computing the deflated solutions, amounts to a few more backsolves for computing the two null vectors and storage for them. For the SVD-based deflated decompositions, the number of extra backsolves depends on the convergence of the inverse iteration. If the inverse iteration fails to converge, then an extra copy of A has to be stored for computing c_p. For the LU-based deflated decompositions, only two more backsolves are needed and the matrix A does not have to be stored. When there are multiple right hand sides for the M-system, the null vectors and the deflated decomposition for v have to computed only once and the cost per extra right hand side is then no more than that of Algorithm BE.

5. Error Analysis

We prove in this section that Algorithm DBE is stable by exhibiting a backward round-off error bound for (x, y). We shall show that the (x, y)

produced by Algorithm DBE give small residuals to the system (1) if the computed solutions are not large. If M is well-conditioned, then it follows that the errors in (x, y) are also small. If M is ill-conditioned, a small residual is all we can hope for.

We shall use '~' to denote computed quantities. To simplify the error analysis, we shall make the following assumption:

Assumption 1: We shall assume that the only source of errors in Algorithm DBE is in solving systems with A (i.e. errors in $\tilde{v}_D$, $\tilde{w}_D$, $\tilde{\xi}$ and $\tilde{\Phi}$) and that no round-off errors are made in carrying out the operations in (24).

This is a reasonable assumption because (24) represents a stable algorithm. The actual round-off errors made in (24) can be bounded by a small constant times the machine precision times quantities like $\tilde{v}_D$, $\tilde{w}_D$, $\tilde{\delta}$, b, c, d, f, g. These can all be absorbed into our final bounds.

The following lemma shows that the residuals for (1) depends on the accuracy of the computed $\tilde{v}_D$, $\tilde{w}_D$, $\tilde{\xi}$, $\tilde{\Phi}$ and $\tilde{\delta}$.

Lemma 5: If the computed $\tilde{v}_D$, $\tilde{w}_D$, $\tilde{\xi}$, $\tilde{\Phi}$ and $\tilde{\delta}$ used in Algorithm DBE satisfy

$$SA\tilde{v}_D - Rb = r_b,$$

$$SA\tilde{w}_D - Rf = r_f,$$

$$A\tilde{\Phi} - \tilde{\delta}\tilde{\xi} = r_\xi,$$

then the computed solutions $(\tilde{x}, \tilde{y})$ satisfy:

$$M \begin{vmatrix} \tilde{x} \\ \tilde{y} \end{vmatrix} = \begin{vmatrix} r_f - \tilde{y} r_b + (\tilde{h}_3/\tilde{D})\, r_\xi \\ 0 \end{vmatrix}. \tag{25}$$

Proof: The proof is rather straight-forward albeit a bit tedious and follows from a direct substitution of (24) into the expression for the residuals.

The actual residuals r_b, r_f and r_ξ depend on the way $\tilde{v}_D$, $\tilde{w}_D$ and $\tilde{\Phi}$ are computed. Next, we consider the use of Gaussian Elimination with some form of pivoting.

Lemma 6: If we use Gaussian Elimination with some form of pivoting for solving systems with A, then the residual r_ξ satisfy:

$$r_\xi \leq k(n)\varepsilon_M ||A||, \tag{26}$$

where ε_M is the machine precision and k(n) is a polynomial in n that depends on the form of pivoting used. Further, if Algorithm NIA <u>without Step (3)</u> is used to compute $\tilde{v}_D$ and $\tilde{w}_D$, then

$$r_b \leq k(n)\varepsilon_M ||\tilde{v}_D|| ||A||, \tag{27}$$

$$r_f \leq k(n)\varepsilon_M ||\tilde{w}_D|| ||A||. \tag{28}$$

Proof: The standard backward error bounds for Gaussian Elimination with pivoting (e.g. [6], p.181) for solving a general linear system Az = p has the form

$$||p - A\tilde{z}|| \leq k(n)\varepsilon_M ||\tilde{z}|| ||A||, \tag{29}$$

where k(n) is a slowly growing polynomial that depends on the pivoting strategy. The bound for r_ξ follows directly from this. For the other two bounds, observe that the vector d produced at Step (2) Algorithm NIA satisfies Ad = Rb + B, where the vector B satisfies a bound similar to (29). Therefore, r_d = SAd - Rb = S(Rb + B) - Rb = (S - I)Rb + SB. It

can easily be shown from Table 3-1 that $(S - I)R = 0$ for all the choices of S and R. For $S = P_{\xi_{SV}}$ or $P_{\xi_{LU}}$, $||SB|| \leq ||B||$. For $S = (E^j_{\xi_{LU}})^T$, $||SB|| \leq (1 + 1/(\xi_{LU})_j)\ ||B||$. Since $(\xi_{LU})_j$ is chosen to be $O(1)$, we have $||SB|| \leq C\ ||B||$ where C is a small constant. The bounds for r_b and r_f follows since C can be absorbed into k(n).

Notice that we have used Algorithm NIA without Step (3) in order to obtain the above bounds. If Step (3) of Algorithm NIA is used, then we can show, by using the bound (26), that $SANd - SAd = B_1$, where $||B_1|| \leq 2\varepsilon_M k(n)||A||||d||$. It follows that $||r_z|| = ||(SANd - SAd) + (SAd - Rb)|| \leq \varepsilon_M k(n)||A||||d||$, where again the constants are absorbed into k(n). However, we cannot in general obtain a bound in terms of $||z||$ rather than $||d||$. If ξ is not close to the left null vector of A in the z_S deflation, or if $\varepsilon \gg O(\sigma_n)$ in the LU-based deflation, then $||z||$ can be much smaller than $||d||$. However, we believe that in practice the bounds for r_b and r_f will still be satisfied if either Algorithm NIA or Algorithm IIA is used for computing $\tilde{v}_D$ and $\tilde{w}_D$.

Using Lemmas 5 and 6, we obtain our main result:

Theorem 7: If the computed quantities $\tilde{v}_D$, $\tilde{w}_D$, $\tilde{\phi}$, $\tilde{\xi}$ and $\tilde{\delta}$ satisfy the bounds in Lemma 6, and no further round-off errors are made in Algorithm DBE in the sense of Assumption 1, then the computed solutions $(\tilde{x}, \tilde{y})$ satisfy:

$$||A\tilde{x} + \tilde{y}b - f|| \leq \{2(||\tilde{w}_D|| + |\tilde{y}|||\tilde{v}_D||) + ||\tilde{x}||\}\ k(n)||A||\varepsilon_M.$$

$$||c^T\tilde{x} + \tilde{y}d - g|| = 0.$$

Proof: The proof follows directly from Lemmas 5 and 6, and by observing that, from (24), $|\tilde{h}_3/\tilde{D}| \leq ||\tilde{x}|| + ||\tilde{w}_D|| + |\tilde{y}|\ ||\tilde{v}_D||$.

From Theorem 7, we see that the key to the success of Algorithm DBE is to control the size of $\tilde{v}_D$, $\tilde{w}_D$, $\tilde{y}$ and $\tilde{x}$. When A is not nearly singular, this is no problem. When A is nearly singular, Algorithm DBE achieves this by the deflation techniques. Thus, the stability of Algorithm DBE is <u>independent</u> of the singularity of A. In practice, this means that, with only a little overhead, the same algorithm can be used to solve systems with M accurately independent of whether A is nearly singular or not.

6. Numerical Results

We performed some numerical tests to verify the accuracy and stability of Algorithm DBE with the various deflation techniques. We considered two classes of matrices for A:

A_1 : $(I-2uu^T)$ Diag$(\sigma_n, n-1, n-2, \ldots\ldots, 1)$ $(I-2vv^T)$ where u and v are chosen randomly and scaled to have norm 1, and σ_n varies from 1 to 10^{-8}.

A_2 : $T - \lambda_{min}(T)I - \sigma_n I$ where T = Tridiagonal(1,-2,1) and σ_n again varies from 1 to 10^{-8}.

Note that σ_n is equal to the smallest singular value of A_1 and A_2. For A_1, the smallest singular value has multiplicity 2 when $\sigma_n = 1$. The dimension n of A is chosen to be 19, so that the dimension of M is 20. The vectors b and c are chosen randomly in (0,1) and d is set to 1. The solutions (x, y) are also generated randomly and the corresponding right hand sides (f, g) are then computed by multiplying (x, y) by M.

In the inverse iteration for determining the approximate singular vector ξ_{SV}, we start with the vector with all components equal to 1 and always take 5

iterations. When A is highly singular, one or two iterations is enough for full accuracy. However, we had some convergence difficulties when A does not have a well isolated small singular value.

For the deflation, we use Algorithm NIA without the correction step (3). This is the case covered by the error bounds in Section 5. We have also performed the same tests using Algorithms NIA and IIA with qualitatively the same results. Moreover, Algorithm IIA always converged in one iteration. For comparison, we will also use Algorithm BE without deflation and direct Gaussian Elimination (GE) on M itself. All LU-factorizations are performed by the routine SGECO of LINPACK [7] which uses the partial pivoting strategy. The computations were performed on a DEC-20 with 27 bits mantissas corresponding to a machine precision of about $.4 \times 10^{-8}$.

The first set of tests is to see how ε varies with σ_n. The computed ε, its position k, the computed σ and the reciprocal of the estimated condition number of M (the parameter RCOND in routine SGECO) are given in Table 6-1 for A_1 and A_2. We see that, at least for these two classes of matrices, ε is indeed roughly $O(\sigma_n)$. Moreover, the smallest pivot always appears at the (n,n)-th position. Note also that, when σ_n is well-isolated, the computed σ is rather accurate and has low absolute error. However, when the smallest singular value is not well isolated, the inverse iteration is not successful at all. This is especially true for A_2 because its lowest eigenvalues are rather close to each other. Since this only occurs when A is well-conditioned, the accuracy of Algorithm DBE with the SVD-based deflations is not affected.

The relative residuals of the M equation as computed by the various algorithms are displayed in Figures 6-1 and 6-2. The relative errors are

Table 6-1: Table of ε, k and RCOND as a Function of $\sigma_n = 10^{-I}$

A_1

I	computed σ	ε	k	RCOND	
0	0.1000007E+01	0.1183648E+01	19	0.1065610E-01	*
1	0.9999996E-01	0.9543458E+00	19	0.1099506E-01	
2	0.1000000E-01	-0.6528494E-01	19	0.1687405E-02	
3	0.1000016E-02	0.7956855E-01	19	0.1218959E-03	
4	0.9999102E-04	-0.1555381E-02	19	0.2518016E-02	
5	0.1001859E-04	-0.9973533E-04	19	0.1758764E-02	
6	0.9929115E-06	0.4390627E-04	19	0.6221786E-02	
7	0.9942711E-07	-0.7852563E-04	19	0.2147673E-02	
8	0.1375734E-07	0.8866191E-06	19	0.6173249E-03	

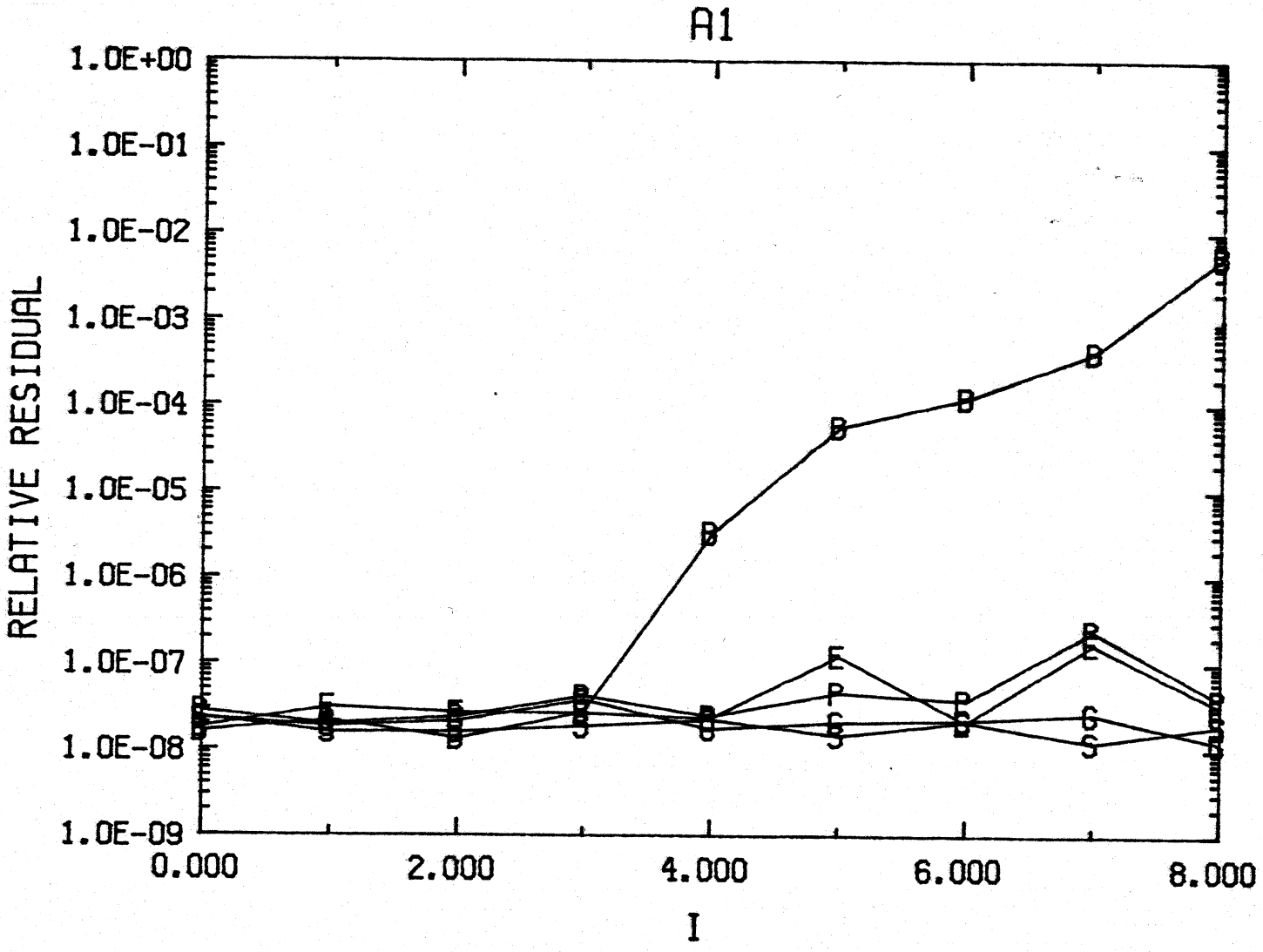
A_2

I	computed σ	ε	k	RCOND	
0	0.6739568E-01	-0.7148705E+00	19	0.9330396E-02	**
1	0.9806986E-01	0.7308936E+00	19	0.4726424E-02	***
2	0.1000000E-01	0.5541958E+00	19	0.1737031E-03	
3	0.1000000E-02	0.6328177E-01	19	0.3318758E-03	
4	0.1000006E-03	0.6386045E-02	19	0.3319462E-02	
5	0.9999954E-05	0.6391779E-03	19	0.1049820E-02	
6	0.9980513E-06	0.6379932E-04	19	0.2220898E-03	
7	0.1047228E-06	0.6694347E-05	19	0.5903182E-02	
8	0.1462737E-07	0.9350479E-06	19	0.7973481E-03	

* Singular Vectors had not converged after 5 iterations.
** Singular Value converged to the wrong value.
*** Inverse Iteration had not converged after 5 iterations.

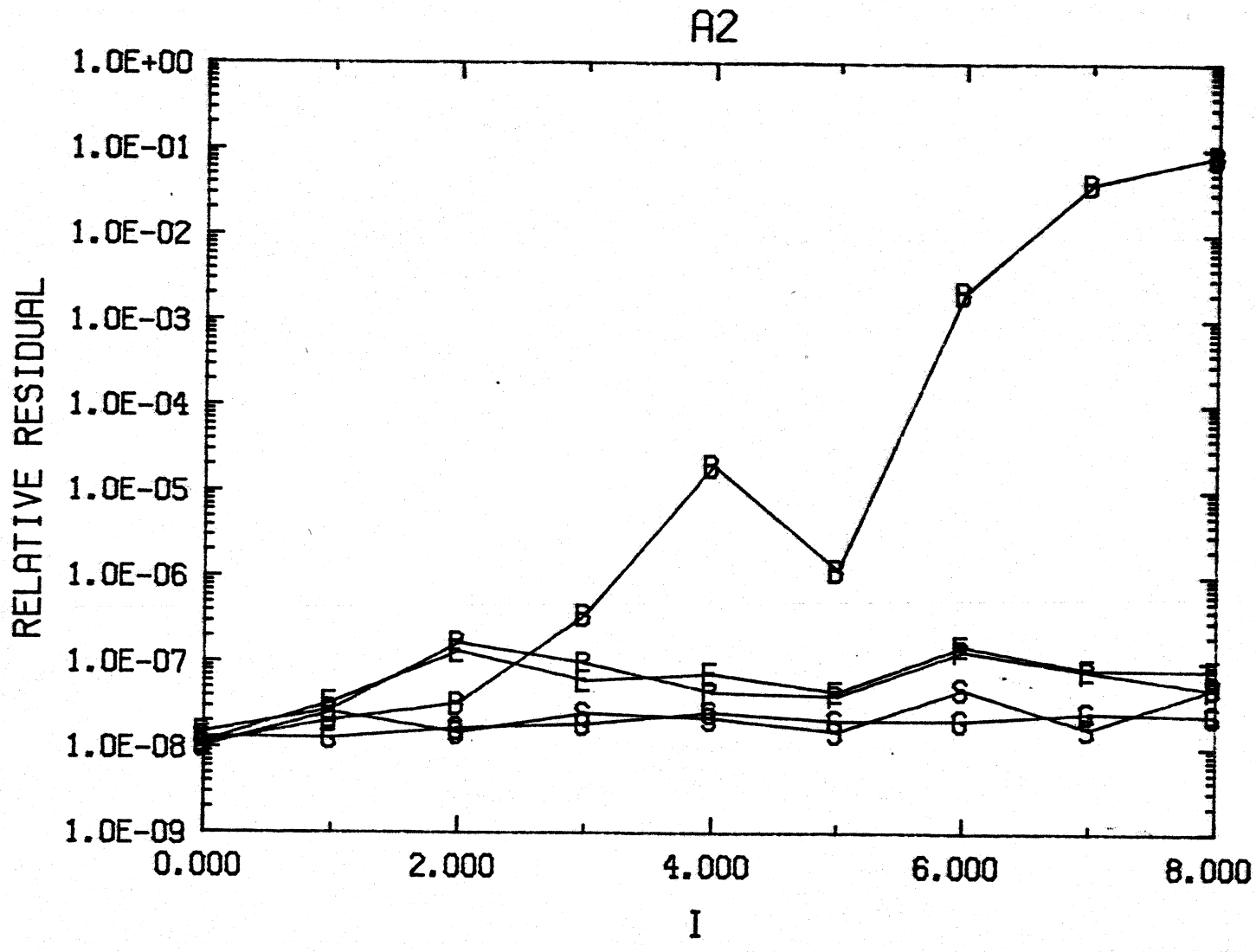
displayed in Figures 6-3 and 6-4. It is seen that Algorithm DBE with any of the three deflation techniques achieve about the same accuracy as that of GE on M, whereas Algorithm BE with no deflation loses accuracy as A tends to being singular. Note also that we do not try to control the singularity of M itself and it can become rather ill-conditioned. However, by the backward error bounds in Section 5, the relative residuals to the M equation should still be small in these situations because the solutions (x, y) are small by construction. The relative errors, however, will be large if M is ill-conditioned. This is reflected in the figures too.

Figure 6-1: Relative Residual as a Function of $\sigma_n = 10^{-I}$ for A_1



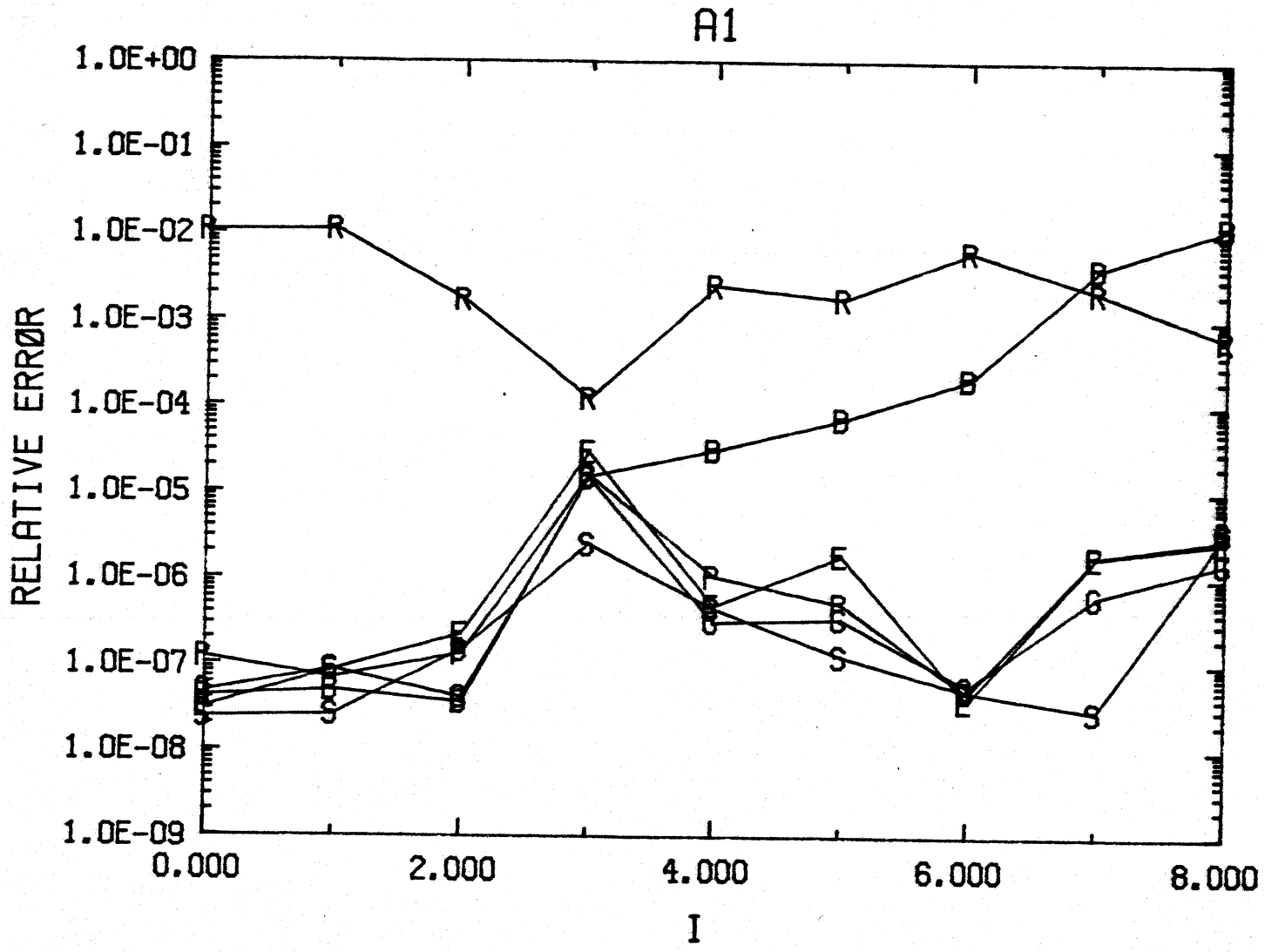
G G.E. ØN M
B B.E. WITH NØ DEFLATIØN
E B.E. WITH E-DEFLATIØN
P B.E. WITH P-DEFLATIØN
S B.E. WITH S-DEFLATIØN

Figure 6-2: Relative Residual as a Function of $\sigma_n = 10^{-I}$ for A_2



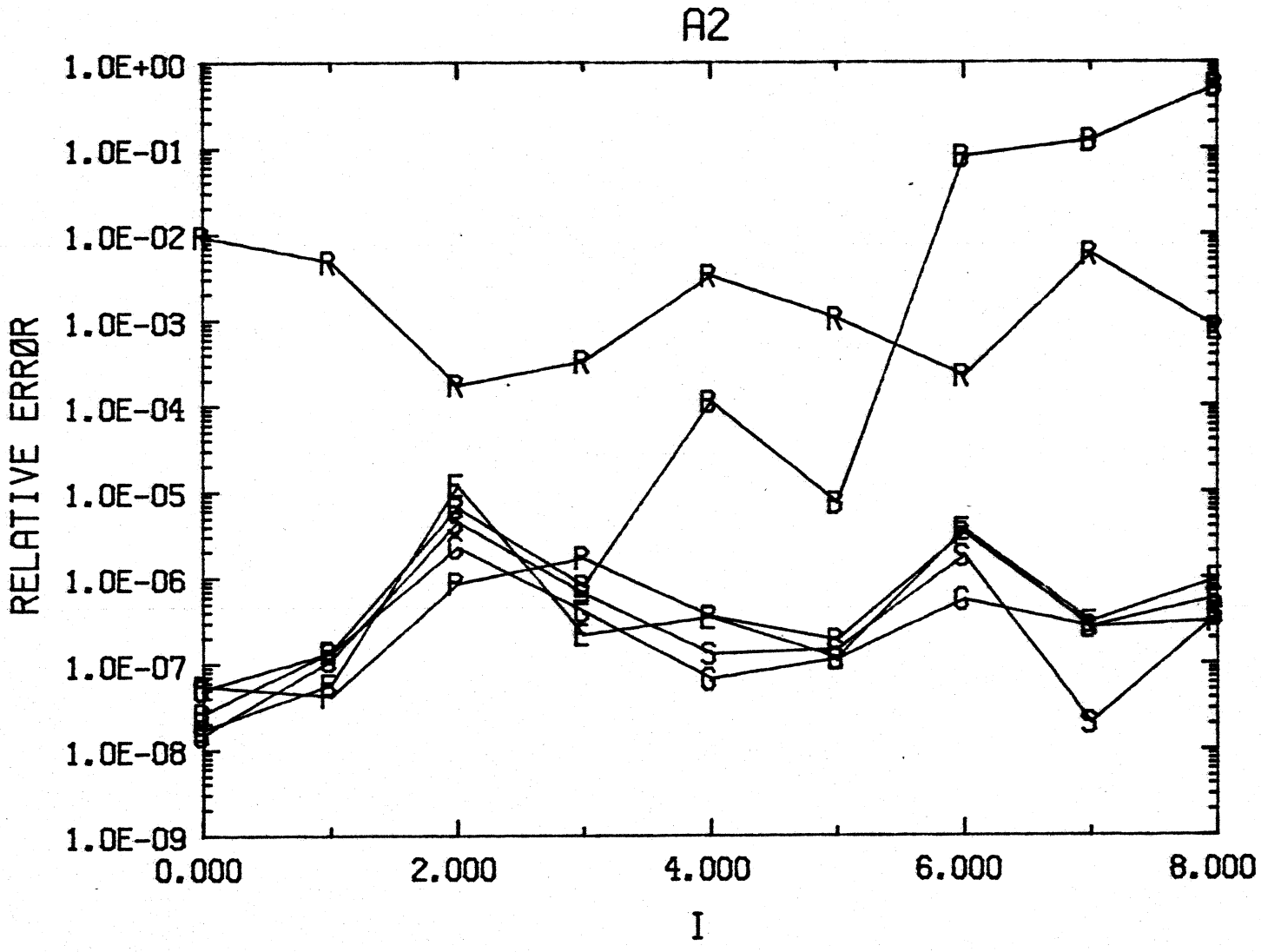
G G.E. ØN M
B B.E. WITH NØ DEFLATIØN
E B.E. WITH E-DEFLATIØN
P B.E. WITH P-DEFLATIØN
S B.E. WITH S-DEFLATIØN

Figure 6-3: Relative Error as a Function of $\sigma_n = 10^{-I}$ for A_1



G G.E. ØN M
B B.E. WITH NØ DEFLATIØN
E B.E. WITH E-DEFLATIØN
P B.E. WITH P-DEFLATIØN
S B.E. WITH S-DEFLATIØN
R RCØND ØF M (LINPACK)

Figure 6-4: Relative Error as a Function of $\sigma_n = 10^{-I}$ for A_2



G G.E. ØN M
B B.E. WITH NØ DEFLATIØN
E B.E. WITH E-DEFLATIØN
P B.E. WITH P-DEFLATIØN
S B.E. WITH S-DEFLATIØN
R RCØND ØF M (LINPACK)

REFERENCES

[1]
J.P. Abbott, An Efficient Algorithm for the Determination of Certain Bifurcation Points, Journal of Computational and Applied Mathematics, 4 (1978), pp. 19 - 27.

[2]
E. Allgower and K. Georg, Simplicial and Continuation Methods for Approximating Fixed Points and Solutions to Systems of Equations, SIAM Review, 22 (1980), pp. 28 - 85.

[3]
T.F. Chan, Deflated Decomposition of Solutions of Nearly Singular Systems, Tech. Rep.225, Yale Computer Science Department, New Haven, CT06520, 1982.

[4]
T.F. Chan and H.B. Keller, Arclength Continuation and Multi-Grid Techniques for Nonlinear Eigenvalue Problems, to appear in SIAM J. Sci. Stat. Comp., June, 1982.

[5]
T.F. Chan, On the Existence of Small Pivots in the LU-factorization of a matrix, in preparation.

[6]
G. Dahlquist and A. Bjorck, Numerical Methods, Prentice-Hall, Englewood Cliffs, N.J., 1974.

[7]
J.J. Dongarra, J.R. Bunch, C.B. Moler and G.W. Stewart, LINPACK User's Guide, SIAM, Philadelphia, 1979.

[8]
C.B. Garcia and W.I. Zangwill, Pathways to Solutions, Fixed Points and Equilibria, Prentice-Hall, Englewood Cliffs, N.J., 1981.

[9]
H.B. Keller, Singular Systems, Inverse Iteration and Least Squares, unpublished manuscript, Applied Mathematics Department, Caltech, Pasadena, California, 1978.

[10]
H.B. Keller, Numerical Continuation Methods, Short Course Lecture Notes, National Bureau of Standards, Center for Applied Mathematics, 1981.

[11]
H.B.Keller, Numerical Solution of Bifurcation and Nonlinear Eigenvalue Problems, Applications of Bifurcation Theory, P. Rabinowitz, ed., Academic Press, New York, 1977; pp. 359-384.

[12]

H.D. Mittelmann and H. Weber, Numerical Methods for Bifurcation Problems - A Survey and Classification, Bifurcation Problems and their Numerical Solution, Workshop on Bifurcation Problems and their Numerical Solution, January 15-17; Dortmund, 1980, pp. 1-45.

[13]
W.C. Rheinboldt, Numerical Methods for a Class of Finite Dimensional Bifurcation Problems, SIAM J. of Numer. Anal., 15 (1978), pp. 1-11.

[14]
W.C. Rheinboldt, Solution Fields of Nonlinear Equations and Continuation Methods, SIAM J. Numer. Anal., 17 (1980), pp. 221-237.

[15]
W.C. Rheinboldt, Numerical Analysis of Continuation Methods for Nonlinear Structural Problems, Computers and Structures, 13 (1981), pp. 103-113.

[16]
G.W. Stewart, On the Implicit Deflation of Nearly Singular Systems of Linear Equations, SIAM J. Sci. Stat. Comp., 2 (1981), pp. 136-140.