Validated Bounds on Basis Vectors for the Null Space of a Full Rank Rectangular Matrix

R. Baker Kearfott

Department of Mathematics, University of Louisiana, Box 4-1010, Lafayette, Louisiana 70504-1010, USA

Abstract

An orthogonal basis for the null space of a rectangular m by n matrix, with $m < n$, is required in various contexts, and numerous well-known techniques, such as QR factorizations or singular value decompositions, are effective at obtaining numerical approximations to such a basis. However, validated bounds on the components of each of these null space basis vectors are sometimes required. In this note, we present a simple method for reliably computing such bounds, given an approximation to the null space. We have implemented the method, illustrating its practicality.

Key words: numerical linear algebra, validated computations, null space, interval Newton method
1991 MSC: 15-04, 15A18, 65G20

1 Introduction

Consider a matrix $A \in \mathbb{R}^{m\times n}$ with $m < n$, assume A has full rank (that is, assume that the rank of A is m) and consider an orthonormal basis $Z = [Z^{(1)}, Z^{(2)}, \ldots Z^{(n-m)}]$ for the null space of A, that is

$$AZ = 0, \quad A \in \mathbb{R}^{m\times n}, \quad Z \in \mathbb{R}^{n\times(m-n)}, \quad Z^T Z = I, \quad m \le n \tag{1}$$

One use for such an orthonormal basis is in determining the nature of a critical point of a constrained optimization problem. See, for example, the sufficient conditions O1 to O4 on [1, p. 82] for a critical point of a constrained problem to

Email address: `rbk@louisiana.edu` (R. Baker Kearfott).
URL: `http://interval.louisiana.edu/kearfott.html` (R. Baker Kearfott).

correspond to a local minimum; the fourth condition is that $Z^T HZ$ be positive definite, where H is the second-derivative matrix of the Lagrangian function with respect to the primal variables and where the rows of Z correspond to the gradients of the active constraints, evaluated at the critical points. In validated global optimization (where roundoff is taken into account in such a way that the computations are mathematically rigorous), it is not sufficient to use approximations to Z and H, but enclosures that rigorously encompass roundoff and algorithmic approximation errors are needed.

Here, we give a simple but effective way of obtaining tight but rigorous enclosures for the matrix $Z \in \mathbb{R}^{n\times(n-m)}$, given an approximation to Z.

2 The Method

The method is simply as follows:

(1) Find an approximate basis $\check{Z}$ satisfying (1) using a traditional method, such as a QR factorization or singular value decomposition.
(2) Construct a small box $\boldsymbol{Z}$ around $\check{Z}$.
(3) Simply apply an interval Newton method to the system

$$AZ = 0, \qquad Z^T Z = I$$

to prove existence and uniqueness of the solution to (1) within $\boldsymbol{Z}$.
(4) Optionally, iterate the interval Newton method to obtain narrow bounds $\boldsymbol{Z}^*$.

Interval Newton methods are fundamental in validated computing based on interval arithmetic, and are found in most monographs on interval analysis, such as [2], [3], [4], or [5].

3 Discussion

The system of equations to which the interval Newton method is applied is a square system with $n(n-m)$ variables and equations. If we order the variables in column major format, then the linearized system for the interval Newton method is of the form $\boldsymbol{F}'(\boldsymbol{Z})V = -F(\check{Z})$, where $V_{i,j} = Z_{i,j} - \check{Z}_{i,j}$, that is,

$$\left(\begin{array}{cccc} A & 0 & \cdots & 0 \\ 0 & A & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & A \\ \hline 2Z_{:,1}^T & 0_{1\times n} & \cdots & 0_{1\times n} \\ 0_{1\times n} & Z_{:,1}^T & \cdots & 0_{1\times n} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{1\times n} & 0_{1\times n} & \cdots & Z_{:,1}^T \\ \hline \vdots & \vdots & \vdots & \vdots \\ \hline Z_{:,n-m}^T & 0_{1\times n} & \cdots & 0_{1\times n} \\ 0_{1\times n} & Z_{:,n-m}^T & \cdots & 0_{1\times n} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{1\times n} & 0_{1\times n} & \cdots & 2Z_{:,n-m}^T \end{array}\right) \left(\begin{array}{c} V_{1,1} \\ \vdots \\ V_{n,1} \\ \hline V_{1,2} \\ \vdots \\ V_{n,2} \\ \hline \vdots \\ \hline V_{1,n-m} \\ \vdots \\ V_{n,n-m} \end{array}\right) = - \left(\begin{array}{c} A\check{Z}_{:,1} \\ \vdots \\ A\check{Z}_{:,n-m} \\ \hline \check{Z}_{:,1}^T\check{Z}_{:,1} - 1 \\ \check{Z}_{:,1}^T\check{Z}_{:,2} \\ \vdots \\ \check{Z}_{:,1}^T\check{Z}_{:,n-m} \\ \hline \vdots \\ \hline \check{Z}_{:,n-m}^T\check{Z}_{:,n-m} \\ \check{Z}_{:,1}^T\check{Z}_{:,2} \\ \vdots \\ \check{Z}_{:,n-m}^T\check{Z}_{:,n-m} - 1 \end{array}\right). \quad (2)$$

There are $n - m$ block columns in $\boldsymbol{F}'(\boldsymbol{Z})$, and each of the $n - m$ row blocks below the block $m(n-m) \times n(n-m)$ diagonal matrix of copies of A contains $n - m$ rows.

If Z is a point vector, then $F'(Z)$ is singular (and the traditional interval Newton validation will fail) if A is not of full rank. Conversely, if A is of full rank and Z satisfies (1), then $F'(Z)$ is nonsingular; this is because the vectors vectors $Z_{:,i}^T$, $1 \le i \le n - m$ must be orthogonal to the row space of A, and hence are linearly independent of the rows of A.

Although the matrix $\boldsymbol{F}'(\boldsymbol{Z})$ is of order $n(n - m)$, it has only $(n - m)(mn) + (n - m)^2 n = n^2(n - m)$ nonzeros, for a fill-in ratio of $1/(n - m)$.

4 Experimental Results

We implemented the validation using the interval arithmetic support within GlobSol [6], on a dual 3.2GHz Pentium-4 based machine with 2 gigabytes of memory, running Microsoft Windows XP, and using the Compaq Visual Fortran compiler version 6.6, with optimization level 0. We used the LINPACK [7] routine DSVDC to compute the approximate null space, and the interval

Gauss–Seidel method, with inverse midpoint preconditioner (see, e.g. [2] or [5]), for the interval Newton method; we did not take advantage of possible sparsity.

We tested the procedure with various runs of randomly generated matrices. In these tests, we first specified the number of matrices N to be generated, and the maximum M, $m \le M$, $n \le N$, that m and n could be. We then used the Fortran intrinsic uniform pseudo-random number generator to generate m and n. We then generated the matrices themselves, containing pseudo-uniformly distributed entries between -1 and 1. The results appear in Table 1. *Remarkably, the code never failed to validate the null space vectors* that DSVDC computed, although a significant amount of time was required for the larger systems [1]. The other quantities in Table 1 are: $D_{\min}$, the minimum dimension of the system (2) for the run; $D_{\max}$, the maximum dimension of the system (2) for the run; and T_{tot}, the total processor time in seconds for the run, rounded to the nearest second or four digits.

Table 1
Runs with randomly generated matrices (See text.)

N	M	$D_{\min}$	$D_{\max}$	T_{tot}
10	25	16	456	32
100	25	3	504	262
10	50	8	1,560	853
100	50	8	2,205	14,680
10	100	388	6,090	62,940

5 Conclusions

We have proposed a simple method for producing validated bounds on the null space of a rectangular, full-rank matrix. Numerical experiments demonstrate that the method is highly reliable. The reported times are implementation-dependent, and can be considerably improved.

[1] Besides line-by-line study, we carefully tested the correctness of the code by stepping through it in an interactive debugger, using specific test cases. The code is available from the author

References

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

[2] A. Neumaier, Interval Methods for Systems of Equations, Cambridge University Press, Cambridge, England, 1990.

[3] E. R. Hansen, Global Optimization Using Interval Analysis, Marcel Dekker, Inc., New York, 1992.

[4] E. Hansen, G. W. Walster, Global Optimization Using Interval Analysis, Marcel Dekker, Inc., New York, 2003.

[5] R. B. Kearfott, Rigorous Global Search: Continuous Problems, Kluwer, Dordrecht, Netherlands, 1996.

[6] R. B. Kearfott, Globsol: History, composition, and advice on use, in: Global Optimization and Constraint Satisfaction, Lecture Notes in Computer Science, Springer-Verlag, New York, 2003, pp. 17–31.

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