ON A CONJECTURE OF J.C. BUTCHER AND H. PODHAISKY

I.D. Coope

Department of Mathematics and Statistics
University of Canterbury
Private Bag 4800
Christchurch, New Zealand

Report Number: UCDMS2006/1 FEBRUARY 2006

On a Conjecture of J.C. Butcher and H. Podhaisky

I. D. Coope*

28 February, 2006

Abstract

Given an $n \times n$ orthogonal matrix Q, there exists a diagonal matrix D with each diagonal entry chosen from $\{-1, 1\}$, such that $QD + I$ is non-singular and such that if

$$S = (QD - I)(QD + I)^{-1},$$

then the skew matrix S has every element in the interval $[-1, 1]$.

We prove that such a D exists and show that it can be computed efficiently and reliably.

Contents

*Dept. Mathematics & Statistics, University of Canterbury, Christchurch, NEW ZEALAND. mailto:ian.coope@canterbury.ac.nz

1 Introduction

In this paper we resolve a conjecture observed by J.C. Butcher and H. Podhaisky and stated in [2]. Because the conjecture is true we state it formally as a theorem.

Theorem 1.1 *Given an $n \times n$ orthogonal matrix Q, there exists a matrix $D = \mathrm{diag}(d_{11}, d_{22}, d_{33}, \ldots, d_{nn})$ with each $d_{ii} \in \{-1, 1\}$, such that $QD + I$ is non-singular and such that if*

$$S = (QD - I)(QD + I)^{-1}, \tag{1}$$

then the skew matrix S has every element in $[-1, 1]$.

The method of proof is algorithmic. We present an algorithm for computing D and S which is iterative, well-defined, and guaranteed to terminate in a finite number of iterations. Starting from an initial guess, D_1, chosen so that $QD_1 + I$ is invertible, a sequence of diagonal matrices, $\{D_k,\ k = 1, 2, ..., m\}$, is generated with each diagonal entry $(D_k)_{(i,i)} \in \{-1, 1\}$ such that $(QD_k + I)$ is invertible and $S_m = (QD_m - I)(QD_m + I)^{-1}$ satisfies $\|S_m\| \leq 1$, where $\|\cdot\|$ denotes the (inconsistent) Hölder matrix norm

$$\|S\| = \max_{1 \leq i,j \leq n} |S_{(i,j)}|,$$

and where $A_{(i,j)}$ denotes the (i, j) entry of the matrix A. (Sometimes a_{ij} is used to denote the (i, j) entry of A but we need to refer to entries of inverses too e.g. $(A^{-1})_{(i,j)}$ as in [1]).

We let $\{\sigma_k = \|S_k\|, k = 1, 2, \ldots, m\}$ denote the sequence of matrix norms, where $S_k = (QD_k - I)(QD_k + I)^{-1}$ and we stress that there is no requirement that the sequence $\{\sigma_k\}_1^m$ be decreasing. Nevertheless, it is shown in Section 3 that for the recommended choice of $D_k, k = 1, 2, \ldots,$ a suitable D must be found in finitely many iterations. In practice very few iterations are required because the recommended initial guess rules out almost all of the (at most) 2^{n-1} allowable choices for D and for $n \leq 3$ no iterations are required.

The paper is organised as follows. In Section 2 we assemble some preliminary results that are helpful in understanding and analysing the problem. Most of these are standard results or special cases of standard results that can be found in most modern texts on linear algebra. In Section 3 the algorithm is presented and analysed and it is proved that the required D can always be calculated. In Section 4 we discuss some computational issues which support the claim that the recommended algorithm is efficient and numerically stable.

2 Matrix preliminaries

In this section we present some useful results that are needed in the sequel. First, we observe that alternative forms to (1) for representing S are:

$$\begin{aligned} S &= (QD+I)^{-1}(QD-I) && (2)\\ &= I - 2(QD+I)^{-1} && (3)\\ &= I - 2D(Q+D)^{-1}. && (4) \end{aligned}$$

Because Q and D are orthogonal it follows that QD is also orthogonal. Now if Q_1 and Q_2 are orthogonal matrices of the same order with $\det Q_1 = -\det Q_2$ then

$$\begin{aligned} \det(Q_1+Q_2) &= -\det Q_1 \det(Q_1^T + Q_2^T)\det Q_2\\ &= -\det Q_1(Q_1^T+Q_2^T)Q_2\\ &= -\det(Q_2+Q_1)\\ &= 0. \end{aligned} \quad (5)$$

Therefore, a necessary condition for $QD+I$ (and $Q+D$) to be invertible is

$$\det(Q) = \det(D). \quad (6)$$

In other words there are 2^n possible choices for the diagonal entries of D but at least half of these correspond to singular matrices. If Q is the identity matrix then there is only one choice for D but it is easy to construct examples where 2^{n-1} candidates for D make $Q+D$ invertible. In this latter case numerical experience suggests that, often, only one of these 2^{n-1} candidates satisfies the requirements of Theorem 1.1. Therefore, trial and error will not generally find a suitable D. In contrast the algorithm to be described in Section 3 has no difficulty finding D when $n = 1000$, (which is about the limit of the "comfort zone" of the author's present computer).

We need two more results relating to rank-2 corrections to the identity matrix. If $x, y, u, v \in \mathbf{R}^n$ and $x^Ty = -1 = u^Tv$ then

$$\det\left[I + xy^T + uv^T\right] = -(v^Tx)(y^Tu). \quad (7)$$

If, in addition, v^Tx and y^Tu are nonzero then

$$\left[I + xy^T + uv^T\right]^{-1} = I + \frac{xv^T}{v^Tx} + \frac{uy^T}{y^Tu}. \quad (8)$$

These are easily verified special cases of more general results on rank-2 corrections (see for example, [1]).

Now we can describe and analyse the algorithm.

3 The algorithm

In this section we develop an algorithm for finding a suitable D and corresponding S. First we find a diagonal matrix, D_1, such that A_1 is invertible, where A_k is the matrix

$$A_k = QD_k + 1, \quad k = 1, 2, \ldots, m.$$

The sequence $\{D_k\}_1^m$ is generated so that each A_k is invertible and the sequence $\{\det A_k\}_1^m$ is strictly increasing, provided that $\|S_k\| > 1$ where

$$S_k = I - 2A_k^{-1}, \quad k = 1, 2, \ldots, m.$$

The sequence terminates when $k = m$ is such that $\|S_m\| \leq 1$. Equations (5,6) show that if $Q + D$ is invertible then changing the sign of just one of the diagonal entries of D will result in a singular matrix. Therefore, we must change at least two (and always an even number) of the diagonal entries of D in the search for S. This observation is the basis of the following algorithm.

Algorithm 3.1

1. Given D_1 and $A_1 = QD_1 + I$ with $\det A_1 > 0$, set $k = 1$.
2. Calculate $S_k = I - 2A_k^{-1}$, and $\sigma_k = \|S_k\|$.
3. If $\sigma_k \leq 1$ set $m = k, S = S_k$ and terminate.
4. Otherwise, find a pair of indices (p, q) such that $\sigma_k = S_{(p,q)}$.
5. Reverse the signs of the (p, p) and (q, q) entries of D_k and call the resulting matrix D_{k+1}. Calculate $A_{k+1} = QD_{k+1} + I$.
6. Set $k := k + 1$ and go to Step 2.

We now show that it is always possible to choose D_1 so that $\det A_1 > 0$ and that, if $m > 1$, then $\det A_{k+1} > \det A_k, \quad k = 1, 2, \ldots, m - 1$, which is enough to show that Algorithm 3.1 is well-defined.

There are many ways to choose a suitable D_1. First consider applying Gaussian elimination, without row or column interchanges, to the matrix $Q + D$ but defer the choice of $D_{(i,i)}$ until the ith stage of elimination so that $D_{(i,i)}$ can be chosen to have the same sign as the ith pivot entry (before adding $D_{(i,i)}$). If the pivot entry is zero then either sign may be chosen for $D_{(i,i)}$. In this way, we produce an upper triangular matrix, U say, whose diagonal entries satisfy $|U_{(i,i)}| \geq 1$. This choice of D satisfies $\det(Q + D) =$

$\prod_{i=1}^{n} U_{(i,i)}$. In fact it is easy to show that $|U_{(n,n)}| = 2$, so we have the stronger result that, with this strategy for choosing D_1, we get $\det(QD_1+I) = \det(Q + D_1)\det D_1 \geq 2$. A variation on this choice which turns out to be highly advantageous is to incorporate diagonal pivoting keeping track of the row/column interchanges in an attempt to make each $|U_{(i,i)}|$ as large as possible. Therefore, it is always easy to find a suitable D_1 and we observe that this strategy is also helpful in calculating A_1^{-1}.

Lemma 3.1 *Let the $n \times n$ matrix $A = QD + I$ be invertible where Q is orthogonal and D is diagonal with $D_{(i,i)} \in \{-1, 1\}, \quad i = 1, 2, ..., n$. Let $\tilde{D}$ be the diagonal matrix with entries $\tilde{D}_{(i,i)} = -D_{(i,i)}, i = p, q, \quad \tilde{D}_{(i,i)} = D_{(i,i)}, i \neq p, q$. If $\tilde{A} = Q\tilde{D} + I$ then*

$$\tilde{A} = A + 2(e_p - Ae_p)e_p^T + 2(e_q - Ae_q)e_q^T, \tag{9}$$

and

$$\det \tilde{A} = 4\left(\left(A^{-1}\right)_{(p,q)}\right)^2 \det A. \tag{10}$$

Proof: Let e_p, e_q denote, respectively, the pth, qth column of the $n \times n$ identity matrix and let $d_{pp} = D_{(p,p)}, \quad d_{qq} = D_{(q,q)}$. Then the pth column of A is

$$Ae_p = (QD + I)e_p = e_p + Qe_p d_{pp}.$$

Similarly, the pth column of $\tilde{A}$ is

$$\tilde{A}e_p = (Q\tilde{D} + I)e_p = e_p - Qe_p d_{pp}.$$

Eliminating the term $Qe_p d_{pp}$ from these two equations gives

$$\tilde{A}e_p = Ae_p + 2(e_p - Ae_p), \tag{11}$$

where we have deliberately written $\tilde{A}e_p$ as a correction to Ae_p. Equation (9) then follows because there is a corresponding equation for $\tilde{A}e_q$. This enables the determinant of $\tilde{A}$ to be calculated efficiently in terms of $\det A$.

$$\begin{aligned} \det \tilde{A} &= \det\left[A + 2(e_p - Ae_p)e_p^T + 2(e_q - Ae_q)e_q^T\right], \\ &= \det A \det\left[I + 2(A^{-1}e_p - e_p)e_p^T + 2(A^{-1}e_q - e_q)e_q^T\right]. \end{aligned} \tag{12}$$

The bracketed terms in equation (12) have the form $I + xy^T + uv^T$ with $y^T x = -1 = v^T u$ (because equation (2) shows that $e_i^T A^{-1} e_i = \frac{1}{2}, \quad i = 1, 2, \ldots, n$), so formula (7) can be applied.

$$\begin{aligned} \det \tilde{A} &= -4\left(e_q^T(A^{-1}e_p - e_p)\right)\left(e_p^T(A^{-1}e_q - e_q)\right)\det A, \\ &= -4\left(e_q^T A^{-1} e_p\right)\left(e_p^T A^{-1} e_q\right)\det A, \end{aligned} \tag{13}$$

and equation (10) follows because equation (2) shows that the matrix A^{-1} is quasi–skew.

It is now easy to see why Algorithm 3.1 works. Equation (2) shows that $S_{(i,j)} = -2\,(A^{-1})_{(i,j)}, i \neq j$, so that at each iteration where $\sigma_k > 1$, Step 4 and Step 5 force a strict increase in $\det A_k$ because

$$\det A_{k+1} = \sigma_k^2 \det A_k.$$

There are only finitely many different choices for D and it is not possible to return to a choice already considered so the iterations must terminate with $\|S\| \leq 1$. Therefore, we have proved Theorem 1.1.

4 Computational aspects

In practice it is not necessary to use the matrices A_k in Algorithm 3.1. Equation (9) shows that A_{k+1} is a rank-2 correction to A_k. Therefore, once an initial D_1 and S_1 have been calculated it is possible to calculate S_{k+1} from S_k by a rank-2 update. In view of the skew–symmetry of S this must have the form

$$S_{k+1} = S_k + b_k a_k^T - a_k b_k^T. \tag{14}$$

Letting $\tilde{S} = I - 2\tilde{A}^{-1}$ and dropping iteration subscripts we write this as

$$\tilde{S} = S + \alpha(ba^T - ab^T), \tag{15}$$

where the scalar α (which can be incorporated into a or b) has been introduced for convenience. The vectors $a, b \in \mathbf{R}^n$ can then be determined by employing formula (8) to obtain $\tilde{A}^{-1}$ in terms of A^{-1} and then using equation (2) to replace A^{-1} by $(I - S)/2$. We find that

$$a = (I + S)e_p, \quad b = (I + S)e_q, \quad \alpha = 1/S_{(p,q)}, \tag{16}$$

so that a and b can be calculated very easily from S. Therefore, the following alternative to Algorithm 3.1 is recommended.

Algorithm 4.1

1. Given D_1 and $S_1 = I - 2(QD_1 + I)^{-1}$, set $k = 1$.
2. Calculate $\sigma_k = \|S_k\|$.
3. If $\sigma_k \leq 1$ set $m = k, S = S_k$ and terminate.
4. Otherwise, find a pair of indices (p, q) such that $\sigma_k = S_{(p,q)}$.

5. Reverse the signs of the (p,p) and (q,q) entries of D_k and call the resulting matrix D_{k+1}.

6. Update S_k using formula (14).

7. Set $k := k+1$ and go to Step 2.

Each iteration of Algorithm 4.1 can be computed in $\mathcal{O}(n^2)$ floating point operations. The search for σ_k in Step 2 has a computational complexity of $\mathcal{O}(n^2)$ which supports the claim that the recommended approach is efficient.

Next we consider the calculation of D_1. If Gaussian elimination with (or without) diagonal pivoting is used with $(D_1)_{(i,i)} \in \{-1,1\}$ chosen dynamically as the elimination proceeds, then it is equivalent to applying Gaussian elimination with complete (i.e. full row and column) pivoting *a posteriori* to the matrix QD_1+I. It is well known that Gaussian elimination with complete pivoting is extremely stable numerically. In order to explain this equivalence, consider the effect of applying the first step of Gaussian elimination without any pivoting. Partition Q (in an obvious notation) as

$$\begin{bmatrix} q_{11} & u^T \\ v & Q_1 \end{bmatrix},$$

so that $q_{11}^2 + u^T u = 1$ and $Q_1^T Q_1 = I_{n-1}$. Then add $d_{11} = \pm 1$ to the $(1,1)$ entry. Applying one step of Gaussian elimination to the resulting matrix, we calculate a vector of multipliers $\ell = v/(q_{11}+d_{11})$ and use it to introduce zeros in the first column to get the matrix

$$\begin{bmatrix} q_{11}+d_{11} & u^T \\ \mathbf{0} & Q_2 \end{bmatrix}, \tag{17}$$

where $Q_2 = Q_1 - \ell u^T$. It is elementary to verify that the submatrix Q_2 is orthogonal irrespective of the choice of sign for d_{11}. Choosing the sign of d_{11} to match the sign of q_{11} ensures that the pivot element $U_{(1,1)} = q_{11}+d_{11}$ is the largest element in the matrix which is what complete pivoting does. Then the same argument can be applied to the submatrix $Q_2 \in \mathbf{R}^{n-1\times n-1}$ as we dynamically convert $Q+D_1$ to the upper triangular matrix U. Eventually we arrive at $Q_n \in \mathbf{R}^{1\times 1}$ which must also be orthogonal. Therefore, the choice of d_{nn} has to make $|U_{(n,n)}| = 2$ in order to avoid a singular upper triangular matrix U.

If diagonal pivoting is used we are essentially computing the factorisation $P(Q+D)P^T = LU$, where P is a permutation matrix. With or without pivoting the procedure for choosing d_{ii} guarantees that each off-diagonal entry of L and U is less than 1 in modulus and each diagonal entry of U

has modulus no less than 1. (If $q_{ii} = 0$ at the ith stage then either the choice $d_{ii} = +1$ or $d_{ii} = -1$ can be made but to avoid ambiguity we choose $d_{ii} = +1$.)

Diagonal pivoting has the advantage that no iterations are required of Algorithm 3.1 when $n \leq 3$ and numerical experience shows that it usually causes fewer iterations for $n > 3$. For example, in over 100,000 trials with random orthogonal matrices of order $n = 200$, no more than 14 iterations were required when diagonal pivoting was used to calculate D_1, and the average was less than 2.3 iterations; 11.5% required no iterations. With no pivoting 29 iterations were needed in one case and the average was greater than 8.3 iterations (which is still good bearing in mind that there are potentially 2^{199} candidates for D); only one of the trials required no iterations in this case.

There is another way to calculate D_1 based on rank-one matrix updating which is extremely convenient and easy to code. The matrix $Q + D = Q + \sum_1^n d_{jj} e_j e_j^T$, so $(Q + D)^{-1}$ can be calculated by applying n rank-one updates sequentially starting from $Q^T = Q^{-1}$. The diagonal pivoting procedure is easily incorporated since it corresponds to changing the order of the updates to take account of the pivoting. The numerical stability of this approach is expected to be inferior to the Gaussian elimination approach described above and the computational cost is about the same. Limited numerical trials suggest that good accuracy can still be achieved but no error analysis is available at present. Yet another way would be to update QR factors at about twice the computational cost. Therefore, there are many ways to calculate a suitable D_1.

5 Concluding remarks

The major result of this article is to answer affirmatively the conjecture raised in [2]. We have shown further that a suitable D and corresponding S can be computed reliably and efficiently in about the time it takes to invert a general $n \times n$ matrix. A surprising feature (at least to the author) of Algorithm 3.1 is that the sequence $\{\sigma_k\}$ need not be monotonic. In one trial with $n = 7$, a matrix Q was found which required 3 iterations, producing the sequence $\{\sigma_k\} = \{1.0126, 1.1216, 1.0981, 0.9107\}$; each iteration apart from the last was worse than the initial approximation!

It is clear from Lemma 3.1 that any D which maximizes $\det(QD + I)$ must yield an S with $\|S\| \leq 1$ but $\det(QD + I)$ need not be maximized (when $n > 3$) in order to produce an S satisfying $\|S\| \leq 1$. For example, the

matrix $Q \in \mathbf{R}^{4\times 4}$,

$$Q = \begin{bmatrix} -0.1650 & 0.6095 & -0.2451 & 0.7357 \\ 0.5384 & 0.1217 & 0.7849 & 0.2814 \\ -0.1270 & -0.7795 & -0.0119 & 0.6133 \\ 0.8166 & -0.0784 & -0.5689 & 0.0585 \end{bmatrix},$$

has two solutions for D namely,

$$\begin{aligned} D_1 &= \mathrm{diag}(1,-1,-1,-1), \quad \|S_1\| = 0.9022, \quad \det A_1 = 3.3090, \\ D_2 &= \mathrm{diag}(-1,1,1,1), \quad \|S_2\| = 0.7617, \quad \det A_2 = 4.6421. \end{aligned}$$

When $n = 3$, if $\|S\| < 1$ then the D which produces it is unique. To see why, observe that any alternative D must have the same determinant so two signs must be reversed, say d_{pp} and d_{qq}. The effect of this on S is is to produce $\tilde{S}$ with

$$\tilde{S}_{(p,q)} = \frac{-1}{S_{(p,q)}}.$$

Therefore, when $\|S\| < 1$ every allowable alternative $\tilde{D}$ results in $\|\tilde{S}\| > 1$. The argument above does not extend to $n > 3$. In the $n = 4$ case, for example, if $\|S\| < 1$ for a given D we have to consider the extra possibility of making 4 sign changes. The example above then shows that sometimes it is possible to find an alternative. Clearly this extends to $n > 4$ since the 4×4 case can be embedded in the higher order cases.

Finally, we remark that the cases $n \leq 3$ are very special since consideration of the eigenvalues of QD for invertible $QD + I$ provides a simple affine relationship between the trace of QD and $\det(QD + I)$,

$$\begin{aligned} \det(QD + I) &= \mathrm{tr}(QD) + 1, \quad n = 1, \\ &= \mathrm{tr}(QD) + 2, \quad n = 2, \\ &= 2\,\mathrm{tr}(QD) + 2 \quad n = 3. \end{aligned}$$

This enables these special cases to be analysed easily. In particular, it explains why the recommended diagonal pivoting procedure works.

Acknowledgements: The author is most grateful to P.A. Sprules and J.C. Butcher for many valuable comments and suggestions on earlier versions of this article.

References

[1] D.S. Bernstein. *Matrix Mathematics.* Princeton University Press, Princeton, N.J., 2005.

[2] J.C. Butcher. Mathematical Miniature 29. *Newsletter of the New Zealand Mathematical Society*, 95:39, 2005.