A Schur Method for the Square Root of a Matrix

Åke Björck
Department of Mathematics
University of Linköping
S-581 83 Linköping, Sweden

and

Sven Hammarling
Numerical Algorithms Group Limited
NAG Central Office
Mayfield House
256 Banbury Road
Oxford OX2 7DE, England

Dedicated to Professor A. M. Ostrowski on his ninetieth birthday.

Submitted by Walter Gautschi

ABSTRACT

A fast and stable method for computing the square root X of a given matrix A ($X^2 = A$) is developed. The method is based on the Schur factorization $A = QSQ^H$ and uses a fast recursion to compute the upper triangular square root of S. It is shown that if $\alpha = \|X\|^2/\|A\|$ is not large, then the computed square root is the exact square root of a matrix close to A. The method is extended for computing the cube root of A. Matrices exist for which the square root computed by the Schur method is ill conditioned, but which nonetheless have well-conditioned square roots. An optimization approach is suggested for computing the well-conditioned square roots in these cases.

1. INTRODUCTION

Given an n by n matrix A, we propose a method for finding an n by n matrix X such that

$$X^2 = A. \tag{1.1}$$

© Elsevier Science Publishing Co., Inc., 1983
52 Vanderbilt Ave., New York, NY 10017 0024-3795/83/$3.00

X is called a square root of A and is denoted as

$$X = A^{1/2}.$$

Unlike the square root of a scalar, the square root of a matrix may not exist. For example, it is easy to verify that the matrix

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{1.2}$$

has no square root.

If A has at least $n - 1$ nonzero eigenvalues, then A always has a square root; otherwise the existence of a square root depends upon the structure of the elementary divisors of A corresponding to the zero eigenvalues. (See for example [6], [15], [4], [1].) To indicate that existence is not a straightforward matter we note that, although the matrix of Equation (1.2) does not have a square root, the matrix

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{1.3}$$

does have a square root, one such square root being

$$X = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Note that in this case X is not representable in the form of a polynomial in A [6, p. 239].

A number of methods have been proposed for computing the square root of a matrix, and these are usually based upon Newton's method, either directly or via the sign function (for example Beavers and Denman [3], Hoskins and Walton [10], Alefeld and Schneider [1]). We first discuss Newton's method, then propose a method for finding a square root based upon the Schur factorization together with a fast recursion method for finding a square root of an upper triangular matrix. Finally we discuss some numerical difficulties and suggest a method for finding a square root in such difficult cases.

2. NEWTON'S METHOD

If X_r is an approximation to X and we put

$$X = X_r + E, \tag{2.1}$$

then substituting in Equation (1.1) and ignoring the term in E^2 gives

$$X_r E + E X_r = A - X_r^2,$$

which gives the Newton iteration

$$X_r E_r + E_r X_r = A - X_r^2, \quad X_{r+1} = X_r + E_r, \qquad r = 0, 1, \dots. \tag{2.2}$$

Several proposed methods start with an initial approximation X_0 which commutes with A, and instead basically use the iteration

$$E_r X_r = \tfrac{1}{2}\left(A - X_r^2\right), \quad X_{r+1} = X_r + E_r, \qquad r = 0, 1, \dots. \tag{2.3}$$

It is easily shown that X_r will then also commute with A for $r \geqslant 1$. However, because of roundoff errors this will not be true for the computed X_r. When $A^{1/2}$ is ill conditioned the iteration (2.3) is therefore less stable than (2.2). Since (2.2) also allows the use of more general initial approximations, it seems to be the more useful extension of Newton's method.

The iteration (2.2) is a special case of the Newton iteration given by Davis [5] for the more general quadratic problem

$$CX^2 + BX - A = 0,$$

and the analysis given by Davis can be applied to (2.2). The equation in (2.2) is a special case of the Sylvester equation

$$GX + XH = C, \tag{2.4}$$

and effective methods for the solution of this equation have been given by Bartels and Stewart [2] and by Golub, Nash, and Van Loan [8]. Equation (2.4) has a unique solution if and only if G and $-H$ have no eigenvalues in common, so that Equation (2.2) has a unique solution if and only if $\beta_i \neq -\beta_j$ for all i and j, where β_i is an eigenvalue of X_r.

Notice that if A and X_r are both upper triangular, then X_{r+1} will also be upper triangular. This suggests that we might first find the Schur factorization of A, given by (Wilkinson [17]):

$$A = QSQ^H, \tag{2.5}$$

where Q is unitary and S is upper triangular, and then use Newton's method to obtain an upper triangular matrix U such that

$$U^2 = S.$$

We can then take

$$X = QUQ^H. \tag{2.6}$$

A suitable initial approximation for U is

$$U_0 = \text{diag}\left(s_{ii}^{1/2}\right),$$

where s_{ii} is the ith diagonal element of S and hence of course an eigenvalue of A. The unexpectedly rapid convergence of Newton's method for this case led us to discover the simple method of the next section.

We should note at this point that an upper triangular matrix may not have an upper triangular square root, but may nevertheless still have a square root. The matrix of Equation (1.3) is such an example. We shall return to this point in Section 6.

3. FINDING A SQUARE ROOT OF AN UPPER TRIANGULAR MATRIX

Let S be an upper triangular matrix with at most one zero diagonal element, and let U be an upper triangular matrix such that

$$U^2 = S. \tag{3.1}$$

For a general analytic function f, Parlett [13] has given a recurrence for the elements of $f(S)$ based on the fact that if $f(S)$ is well defined, then it commutes with S. This permits the buildup of U from its diagonal provided that $u_{ii} \neq u_{jj}$, $i \neq j$ for all i, j. Whenever two or more diagonal elements of U are close, this recurrence has to be modified in a nontrivial way. For finding the square root of S we derive here a recurrence based instead on the relation (3.1).

Comparing coefficients in Equation (3.1) gives

$$s_{ij} = \sum_{k=i}^{j} u_{ik} u_{kj}, \qquad i \leqslant j, \tag{3.2}$$

and hence

$$u_{ii} = s_{ii}^{1/2}, \qquad i = 1, 2, \ldots, n, \tag{3.3}$$

and

$$u_{ij} = \frac{s_{ij} - \sum_{k=i+1}^{j-1} u_{ik} u_{kj}}{u_{ii} + u_{jj}}, \qquad i < j. \tag{3.4}$$

We can therefore compute the elements of U, one superdiagonal at a time, starting with the leading diagonal and moving upwards. When computing the rth diagonal Equation (3.4) only involves previously computed elements of U on the right hand side.

If, whenever $s_{ii} = s_{jj}$, we choose $u_{ii} = u_{jj}$, then since S has at most one zero diagonal element, we are assured that

$$u_{ii} + u_{jj} \neq 0$$

and hence u_{ij} is defined.

If S has more than one zero diagonal element, then an upper triangular matrix U satisfying Equation (3.1) may still exist. This can readily be determined if the Schur factorization of Equation (2.5) is chosen so that the diagonal elements of S are in descending order of absolute magnitude. Such an algorithm for the real case is given by Stewart [14]. In this case if S is singular, then for some $r < n$

$$s_{r+1,r+1} = s_{r+2,r+2} = \cdots = s_{nn} = 0,$$

and U will exist if we also have

$$s_{ij} = 0, \qquad i = r+1, r+2, \ldots, n, \quad j > i. \tag{3.5}$$

In particular we can choose

$$u_{ij}=0, \qquad i=r+1,r+2,\dots,n, \quad j\geqslant i,$$

and then the remaining elements of U can be computed by Equations (3.3) and (3.4). If Equation (3.5) does not hold, then it is readily seen from Equation (3.2) that U does not exist.

We note that if A is Hermitian, then the Schur factorization (2.5) becomes the classical spectral factorization with S diagonal. The matrix U can therefore also be chosen to be diagonal with diagonal elements given by Equation (3.3). Such a square root always exists.

4. STABILITY OF THE METHOD

If we now let U be the *computed* upper triangular matrix given by equations (3.3) and (3.4) and we define the matrix E as

$$U^2=S+E,$$

then a straightforward error analysis applied to Equations (3.3) and (3.4) in the style of Wilkinson [16], as for example for the LU method, shows that

$$|e_{ij}|\leqslant\left(|s_{ij}|+\sum_{k=i}^{j}|u_{ik}u_{kj}|\right)\delta, \qquad i\leqslant j,$$

where δ is of the order of n times the relative machine accuracy, and hence

$$\|E\|\leqslant\left(\|S\|+\|U\|^2\right)\varepsilon, \tag{4.1}$$

where for the 1, ∞, and Frobenius norms $\varepsilon=\delta$, and for the 2 norm $\varepsilon=n^{1/2}\delta$. Let us define α to be such that

$$\|A^{1/2}\|^2=\alpha\|A\|. \tag{4.2}$$

We can think of α as a condition number for the square root. If α is large, then we shall see that the problem of determining $A^{1/2}$ is inherently ill conditioned. Note that we always have $\alpha \geqslant 1$, and that

$$\kappa = \kappa(A^{1/2}) = \|A^{-1/2}\|\|A^{1/2}\| \geqslant \alpha, \tag{4.3}$$

so that when α is large $A^{1/2}$ is necessarily ill conditioned in the usual sense.

The inequality (4.3) is easily derived from the identity $A^{1/2} = A^{-1/2}A$. By taking norms we get

$$\|A^{1/2}\| \leqslant \|A^{-1/2}\|\|A\| = \kappa\|A\|\|A^{1/2}\|^{-1},$$

and multiplying by $\|A^{1/2}\|$, it follows that

$$\frac{\|A^{1/2}\|^2}{\|A\|} = \alpha \leqslant \kappa.$$

Since unitary transformations preserve the 2 and Frobenius norms, Equations (2.5) and (2.6) give that

$$\|U\|^2 = \alpha\|S\| \tag{4.4}$$

and we can expect Equation (4.4) to be a very good approximation even when U and S are the computed matrices. Thus from (4.1) and (4.4) we get for the 2 and Frobenius norms

$$\|E\| \leqslant (1+\alpha)\|S\|\varepsilon \tag{4.5}$$

and $\|E\|$ will be small relative to $\|S\|$ if α is not large relative to unity. In particular if A is Hermitian, so that S and U are diagonal, then

$$\|U\|_2^2 = \|S\|_2, \qquad S = S^H,$$

and $\alpha = 1$, so that the Schur method applied to a Hermitian matrix is numerically stable.

It is important to realize that there exist matrices for which α can be large, and we strongly recommend that α be returned by any routine implementing this method.

A result of the form of Equation (4.5) is more or less inevitable. If we let X be the exact square root of A and let $\tilde{X}$ be the matrix X rounded to working accuracy so that

$$\tilde{X} = X + F, \qquad \|F\| \leqslant u\|X\|,$$

where u is the unit rounding error, then

$$\tilde{X}^2 = A + XF + FX + F^2,$$

and if we put

$$E = XF + FX + F^2,$$

we have that

$$\tilde{X}^2 = A + E,$$

where, using Equation (4.2),

$$\|E\| \leqslant (2+u)u\|X\|^2 = (2+u)u\alpha\|A\|.$$

Once again $\|E\|$ will be small relative to $\|A\|$ if α is not large relative to unity.

Even if the square root obtained by the Schur method is ill conditioned, it may be that A nevertheless has a well-conditioned square root. We pursue this point in Section 6.

5. THE CUBE ROOT OF A MATRIX

The Schur method can also be used to find cube roots of matrices. Let S again be the upper triangular matrix in the factorization (2.5), but here let U be an upper triangular matrix such that

$$U^3 = S, \tag{5.1}$$

We can again find U by comparing coefficients in Equation (5.1). If we let R be the upper triangular matrix given by

$$R = U^2$$

and put

$$t_{ij} = \sum_{k=i+1}^{j-1} u_{ik}u_{kj}, \qquad i < j-1,$$

then

$$s_{ij}=\sum_{k=i}^{j} u_{ik}r_{kj}=u_{ii}r_{ij}+u_{ij}r_{jj}+\sum_{k=i+1}^{j-1} u_{ik}r_{kj}$$

and

$$r_{ij}=\sum_{k=i}^{j} u_{ik}u_{kj}=u_{ij}(u_{ii}+u_{jj})+t_{ij},$$

which leads to the equations

$$u_{ii}=s_{ii}^{1/3}, \qquad r_{ii}=u_{ii}^2, \tag{5.2}$$

$$u_{ij}=\frac{s_{ij}-u_{ii}t_{ij}-\sum_{k=i+1}^{j-1} u_{ik}r_{kj}}{r_{ii}+u_{ii}u_{jj}+r_{jj}}, \qquad i<j, \tag{5.3}$$

$$r_{ij}=u_{ij}(u_{ii}+u_{jj})+t_{ij}, \qquad i<j. \tag{5.4}$$

These equations allow us to compute the elements of U and R one subdiagonal at a time in a similar manner to the computation of the square root. Having found U, we again determine X from Equation (2.6), but here of course X is a cube root of A, so that

$$X^3=A.$$

With some algebraic effort this technique can clearly be extended to other roots.

6. FINDING THE OPTIMALLY CONDITIONED SQUARE ROOT

Consider first the matrix

$$A=\begin{bmatrix}\varepsilon & 1\\ 0 & \varepsilon\end{bmatrix}, \qquad \varepsilon>0.$$

The only square roots of this matrix are

$$X=\begin{bmatrix}\varepsilon^{1/2} & 1/(2\varepsilon^{1/2})\\ 0 & \varepsilon^{1/2}\end{bmatrix} \quad\text{and}\quad X=\begin{bmatrix}-\varepsilon^{1/2} & -1/(2\varepsilon^{1/2})\\ 0 & -\varepsilon^{1/2}\end{bmatrix},$$

so that when ε is small, α is large and the problem of determining *any* square root of A is ill-conditioned. Of course, as $\varepsilon \to 0$ the matrix A tends to the matrix of Equation (1.2), for which no square root exists.

However, if we consider the matrix

$$A=\begin{bmatrix}\varepsilon & 1 & 0\\ 0 & \varepsilon & 0\\ 0 & 0 & \varepsilon\end{bmatrix}, \qquad \varepsilon > 0,$$

then all the upper triangular square roots of A have

$$x_{12} = \pm 1/(2\varepsilon^{1/2}),$$

so that when ε is small, these square roots are ill conditioned, but A also has well-conditioned square roots such as

$$X=\begin{bmatrix}\varepsilon^{1/2} & 0 & 1\\ 0 & \varepsilon^{1/2} & 0\\ 0 & 1 & -\varepsilon^{1/2}\end{bmatrix},$$

for which α is of order unity.

The problem of determining such square roots seems to be far from trivial, but one possible approach is to solve the optimization problem given by

$$\begin{aligned}&\text{minimize} && f(X)=\sum_{j=1}^{n}\sum_{i=1}^{n}|x_{ij}|^2\\ &\text{subject to} && X^2-A=0,\end{aligned} \tag{6.1}$$

which is a nonlinearly constrained least squares problem with n^2 variables and n^2 constraints. We have successfully used this approach to solve a number of small problems using the NPL optimization library [12].

Because there are n^2 variables, when n is not small the use of standard optimization software is expensive in terms of time and/or storage, but we propose such an approach only when the solution given by the Schur method is unacceptable. If a large number of such problems are to be solved then the construction of a routine to take advantage of the special form of the Jacobian of the constraints may be warranted.

Let $C = C(X)$ be the constraint matrix given by

$$C(X) = X^2 - A, \tag{6.2}$$

and use the notation

$$\mathbf{c} = \operatorname{vec}(C) = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}, \quad \text{where} \quad c_j = \begin{bmatrix} c_{1j} \\ c_{2j} \\ \vdots \\ c_{nj} \end{bmatrix}.$$

Also let G be the Jacobian matrix of $\mathbf{c}$, so that G is the n^2 by n^2 matrix such that

$$g_{(k-1)n+i,(m-1)n+j} = \frac{\partial c_{ki}}{\partial x_{jm}}, \qquad k, m, i, j = 1, 2, \ldots, n.$$

By differentiating in Equation (6.2) we find that

$$G = I \otimes X + X^T \otimes I,$$

where $\otimes$ denotes the Kronecker product (see for example [9]), so that if the n^2 element vectors $\mathbf{h} = \operatorname{vec}(H)$ and $\mathbf{y} = \operatorname{vec}(Y)$ are related by

$$\mathbf{y} = G\mathbf{h} \tag{6.3}$$

then

$$Y = XH + HX.$$

Thus if $\mathbf{y}$ is given and we wish to solve the equations (6.3) for $\mathbf{h}$, we again have a special case of Sylvester's equation to solve. The equations (6.3) need to be solved if, for example, we are using an augmented Lagrangian method [7] to solve the problem (6.1).

We note that the matrix G is singular at the solution if $\beta_i = -\beta_j$ for any i and j, where β_i is an eigenvalue of the solution matrix X, and this can cause difficulties when solving the problem (6.1). If this is the case, an alternative possibility is to use a penalty function approach and solve an unconstrained problem such as

$$\text{minimize} \quad f(X) = \sum_{j=1}^{n} \sum_{i=1}^{n} |x_{ij}|^2 + \rho \|X^2 - A\|_F^2, \qquad \rho > 0, \tag{6.4}$$

where ρ is the penalty parameter. Experience is needed here to assess suitable choices of ρ (see for example [7] for a discussion on penalty functions), but we were able to find a square root for the matrix A of the example (1.3) by solving the problem (6.4). Note that any square root of this matrix will have all its eigenvalues zero, and therefore G will be singular.

It is not necessary to apply the method outlined in this section to the full matrix A. Assume that the Schur factorization of A has been computed with diagonal elements of S ordered in descending order of magnitude. Partition S and $U = S^{1/2}$ as

$$S = \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix} \}k, \qquad U = \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} \}k,$$

where k is chosen as large as possible but such that S_{11} has an upper triangular, well-conditioned square root. We need then apply the method of this section only to compute the square root U_{22} of S_{22}. Now

$$\begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix}^2 = \begin{bmatrix} U_{11}^2 & U_{11}U_{12} + U_{12}U_{22} \\ 0 & U_{22}^2 \end{bmatrix},$$

so that we can finally solve for U_{12} from the Sylvester equation,

$$U_{11}U_{12} + U_{12}U_{22} = S_{12}.$$

It should be noted that, even with this sort of optimization approach, a satisfactory square root may still be elusive because the problem (6.1) may have a number of local minima, so that we can still converge to a square root with a large α even when better square roots exist. Of course with suitable starting approximations it may also be possible to coax Newton's method to converge to a satisfactory square root. There are plenty of open questions here related to the existence of well-conditioned square roots.

7. CONCLUSION

We have presented a method for computing a square root of a matrix based upon the Schur factorization of the matrix. The Schur factorization can be computed by numerically stable techniques [17] and requires $O(n^3)$ operations, this being the dominant part of the square root computation. Thus this method is efficient compared to methods based upon Newton's method.

It is important to monitor growth in the size of the norm of the square root relative to $\|A\|^{1/2}$. Unacceptable growth is not likely to be frequent in practice, but when it occurs the discussion in Section 6 presents an alternative strategy.

We gratefully acknowledge the help and advice of Susan Hodson and Helen Jones in the use of the NPL *optimization library. We also thank Michael Saunders who ran some examples using* MINOS [11].

REFERENCES

1 G. Alefeld and N. Schneider, On square roots of M-matrices, *Linear Algebra Appl.* 42:119–132 (1982).
2 R. H. Bartels and G. W. Stewart, Solution of the matrix equation $AX + XB = C$, *Comm. ACM* 15:820–826 (1972).
3 A. N. Beavers and E. D. Denman, The matrix sign function and computations in systems, *Appl. Math. Comput.* 2:63–94 (1976).
4 G. W. Cross and P. Lancaster, Square roots of complex matrices, *Linear and Multilinear Algebra* 1:289–293 (1974).
5 G. J. Davis, Numerical solution of a quadratic matrix equation, *SIAM J. Sci. Statist. Comput.* 2:164–175 (1981).
6 F. R. Gantmacher, *The Theory of Matrices*, Vol. 1, Chelsea, New York, 1959.
7 P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic, London, 1981.
8 G. H. Golub, S. Nash, and C. Van Loan, A Hessenburg-Schur method for the problem $AX + XB = C$, *IEEE Trans. Automat. Control* AC-24:909–913 (1979).
9 A. Graham, *Kronecker Products and Matrix Calculus with Applications*, Ellis Horwood, Chichester, 1981.
10 W. D. Hoskins and D. J. Walton, A faster method of computing the square root of a matrix, *IEEE Trans. Automat. Control* AC-23:494–495 (1978).
11 B. A. Murtagh and M. A. Saunders, A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints, *Mathematical Programming Stud.* 16:84–117 (1982).
12 NPL, *A Brief Guide to the NPL Numerical Optimization Software Library*, National Physical Laboratory, Teddington, Middlesex, TW11 0LW, U.K., 1980.

13 B. N. Parlett, A recurrence among the elements of functions of triangular matrices, *Linear Algebra Appl.* 14:117–121 (1976).

14 G. W. Stewart, HQR3 and EXCHNG: Fortran subroutines for calculating and ordering the eigenvalues of a real upper Hessenburg matrix, *ACM Trans. Math. Software* 2:275–280 (1976).

15 J. H. M. Wedderburn, *Lectures on Matrices*, Dover, New York, 1964.

16 J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Notes on Applied Science No. 32, HMSO, London, 1963.

17 J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford U.P., London, 1965.

Received 16 August 1982; revised 20 September 1982