COMPUTATION OF FUNCTIONS OF TRIANGULAR MATRICES

by

B.N. Parlett

Memorandum No. ERL-M481

November 1974

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

COMPUTATION OF FUNCTIONS OF TRIANGULAR MATRICES†

B.N. Parlett

Department of Mathematics
and
Computer Science Division
Department of Electrical Engineering
and Computer Sciences
Electronics Research Laboratory
University of California, Berkeley

November 1974

Abstract. A simple relation exists among the elements of $\phi(T)$ when ϕ is an analytic function and T is triangular. This permits the rapid build up of $\phi(T)$ from its diagonal. An analogous relation holds for block triangular matrices. This permits the formation in real arithmetic of real functions of real matrices with complex eigenvalues. The confluent case is included. Algorithms are given and some numerical examples.

Keywords. Function of a Matrix, Triangular Matrix

†The author gratefully acknowledges partial support from Office of Naval Research Contract N00014-69-A-0200-1017.

1. Introduction

One of the most spectacular applications of the Jordan Canonical form in the realm of matrix theory is the simple expression it yields for a function of an arbitrary square matrix B, say. Let J be the (upper) Jordan canonical form of B so that

$$B = XJX^{-1} \tag{1}$$

where J is a direct sum of blocks of the typical form

$$J_4(\lambda) \equiv \begin{bmatrix} \lambda & 1 & & \\ & \lambda & 1 & \\ & & \lambda & 1 \\ & & & \lambda \end{bmatrix} . \tag{2}$$

See [3] for more details. For any analytic function ϕ which is regular in an open region containing B's spectrum

$$\phi(B) = X\phi(J)X^{-1} \tag{3}$$

where $\phi(J)$ is a direct sum of blocks of the typical form

$$\phi(J_4(\lambda)) = \begin{bmatrix} \phi(\lambda) & \phi'(\lambda) & \phi''(\lambda)/2 & \phi'''(\lambda)/6 \\ & \phi(\lambda) & \phi'(\lambda) & \phi''(\lambda)/2 \\ & & \phi(\lambda) & \phi'(\lambda) \\ & & & \phi(\lambda) \end{bmatrix} . \tag{4}$$

Consequently, when X, J, ϕ and its derivatives are all given, formula (3) specifies $\phi(B)$ in a way that is useful both in theory and for computation.

Things are not so simple in the usual case when only B and ϕ are given. The fact is that the computation of X and J from B is not a routine matter. The difficulty here stems from the fact that J is not a continuous function of B's elements when B has multiple eigenvalues. In fact near any (defective) matrix with a multiple eigenvalue there are other matrices whose eigenvalues differ in their leading decimal digits! Consequently it is not clear what approximation to J a computer program should deliver. Recent results, see [1], suggest that the bigger the size of the Jordan blocks the more robust will be the approximation. This being the case it is attractive to consider ways of computing $\phi(B)$ without finding J and X.

We recall that $\phi(B) = \phi_I(B)$ where ϕ_I is the polynomial which interpolates ϕ at B's eigenvalues (counting multiplicities). See [3].

The Lagrangian form of the interpolating polynomial gives

$$\phi(B) = \sum_{k=1}^{n} \prod_{\substack{j=1 \\ j\neq k}}^{n} \left(\frac{B-\lambda_j}{\lambda_k-\lambda_j}\right) \phi(\lambda_k) \tag{5}$$

where $\{\lambda_i,\ i=1,\ldots,n\}$ is the spectrum of B. More economical in terms of matrix multiplications is Newton's form of the interpolating polynomial

$$\phi(B) = \phi(\lambda_1)I + \sum_{k=2}^{n} \prod_{i=1}^{k-1} (B-\lambda_i)\phi[\lambda_1,\ldots,\lambda_k] \tag{6}$$

where $\phi[\lambda_1,\ldots,\lambda_k]$ is the divided difference of order k-1 which uses data points $\lambda_1,\ldots,\lambda_k$. Expression (6) requires (n-1) matrix multiplications and one extra matrix array for temporary storage of the partial products. Techniques based on interpolation require $0(n^4)$ basic arithmetic operations and extra storage beyond that needed for B and $\phi(B)$.

For any similarity transformation $B = GCG^{-1}$ we have

$$\phi(B) = G\phi(C)G^{-1} \tag{7}$$

and this suggests a compromise between the use of the Jordan decomposition (3) and the interpolation formulae (5) and (6). There are available today standard programs for effecting the Schur decomposition

$$B = PTP^* \tag{8}$$

where P is unitary $(P^* = P^{-1})$ and T is upper triangular. Between $10n^3$ and $15n^3$ multiplications suffice for this robust factorization. See [0, pp. 16,25].

The usefulness of (8) for the present purpose depends on the cost of forming $\phi(T)$. In Section 3 we present a simple relation among the elements of $\phi(T)$ which permits us to build up $\phi(T)$, one superdiagonal after another, going away from the main diagonal. The operation count is only $n^3/3$ and no extra storage is required. The other sections present the relation and expound the method in greater detail, covering the treatment of multiple eigenvalues.

2. A Matrix Function as a Contour Integral

Let B be a square matrix with real or complex elements and let $\phi(z)$ be an analytic function which is regular inside and on some contour Γ which contains T's spectrum. One of the most elegant definitions of the matrix $\phi(B)$ is attributed to E. Cartan; see [1, p. 44] or [3] for more details.

$$\phi(B) = \frac{1}{2\pi i}\oint_{\Gamma} \phi(z)(z-B)^{-1}dz \qquad (1)$$

where the integral is taken counterclockwise along Γ. This formula is very useful in theoretical work.

As an application of (1) we prove the following well known result.

> Lemma 1. Let T be upper triangular. Then $f_{rs} \equiv \phi(T)_{rs}$ is completely determined by ϕ and those elements t_{ij} of T satisfying $r \leq i \leq j \leq s$.

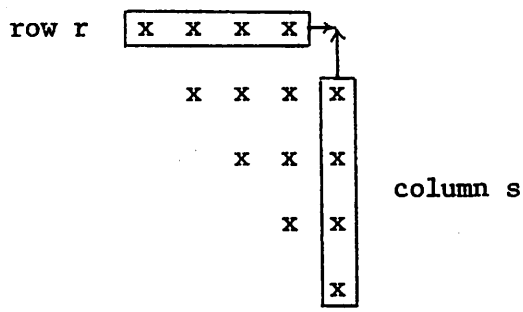
Proof. The (r,s) element, u_{rs}, of $U \equiv (z-T)^{-1}$ is completely determined by z and the elements of T specified above. This can be seen by back-solving $(z-T)u_s = e_s$ for u_s, the s-th column of U with e_s denoting the s-th column of the identity I.

By (1) the (r,s) element of $F = \phi(T)$ must inherit the same dependence as u_{rs} since

$$f_{rs} = \frac{1}{2\pi i}\oint_{\Gamma} \phi(z)u_{rs}(z)dz \quad . \qquad \square$$

3. A Relation Among the Elements of $\phi(T)$

Let T be an upper triangular matrix with real or complex elements and let $\phi(z)$ be an analytic function which is regular inside and on some contour Γ which contains T's spectrum. Then the matrix $F = \phi(T)$ is well defined and also upper triangular. Although a typical element f_{rs}, $r < s$, of F is a complicated function of ϕ and T it turns out that it is simply related to the elements in its row and column which are closer to the main diagonal, as the figure shows.



The result is a simple consequence of the following well known result.

> <u>Lemma 2</u>. Any square matrix B comutes with $\phi(B)$ for any scalar function ϕ for which $\phi(B)$ is well defined.

<u>Proof</u>. The integral definition (Section 2) of the function $z\phi(z)$ evaluated at B yields

$$B\phi(B) = \frac{1}{2\pi i} \oint_{\Gamma} z\phi(z)(z-B)^{-1}dz = \phi(B)B$$

since $z\phi(z) = \phi(z)z$. $\square$

> <u>Theorem 1</u>. Let $F \equiv \phi(T)$ be an analytic function of an upper triangular matrix T. For $r < s$,
>
> $$(t_{rr}-t_{ss})f_{rs} = (f_{rr}-f_{ss})t_{rs} + \sum_{k=1}^{s-r-1} (f_{r,r+k}t_{r+k,s} - t_{r,s-k}f_{s-k,s}) \quad .$$

<u>Proof</u>. By Lemma 1 F is upper triangular. By Lemma 2

$$FT - TF = 0 \quad .$$

On writing out the (r,s) element of the left hand side and rearranging terms the result is obtained. □

The formation of an $n \times n$ matrix $\phi(T)$ from its diagonal elements $f_{jj} = \phi(t_{jj})$, $j = 1,\ldots,n$ requires $(n-2)(n-1)(2n-3)/6 + (n-1)^2$ multiplications.

Theorem 1 is of no use when $t_{rr} = t_{ss}$ and for completeness we describe in the next two sections how to form $\phi(T)$ in the confluent case on the assumption that all necessary derivatives of ϕ can be evaluated. In practice, however, the cases to consider are those in which the diagonal elements are close and only ϕ is available. We will not pursue this topic here.

<u>Corollary</u>. Let $T = (T_{ij})$ be block upper triangular. Then $F = \phi(T)$ will have the same block structure and, for $r < s$,

$$T_{rr}F_{rs} - F_{rs}T_{ss} = \sum_{k=0}^{s-r-1} (F_{r,r+k}T_{r+k,s} - T_{r,s-k}F_{s-k,s}) \quad .$$

<u>Proof</u>. Equate the (r,s) block of $TF - FT$ to zero. □

This corollary gives a set of linear equations for the elements of F_{rs} in terms of T_{rr}, T_{ss} and the right hand side R_{rs}. A solution exists provided that T_{rr} and T_{ss} have no eigenvalues in common. Equations of the form $AX - XB = C$ have been studied quite extensively and special methods for solving them have been devised. However if T_{rr}

and T_{ss} are at most 2×2 then it is probably best to use an ordinary linear equations solver which uses pivotal interchanges. In the 1×2 and 2×1 case the solution should be written out explicitly.

This corollary is useful because it shows how real functions of real matrices with complex eigenvalues can be evaluated in real arithmetic. First the given matrix is reduced to block triangular form T by the QR Algorithm [0, p.25]. The diagonal blocks are either 1×1 or 2×2 and the result of the corollary can be readily applied to T.

Another use of the corollary is in dealing with those cases in which T has exactly multiple eigenvalues. These can be grouped into blocks and the diagonal blocks of F can be evaluated by the algorithm described in the next section.

4. The Confluent Case

Suppose that $t_{rr} = t_{ss}$ and, further, that $t_{ii} = \lambda$ for $r \leq i \leq s$. This stronger condition can be enforced by taking the trouble to put all equal eigenvalues into adjacent positions down T's diagonal. There exist plane rotations which exchange adjacent diagonal elements and preserve triangularity.

By Lemma 1 it suffices to consider the $(1,n)$ element in the case when all the eigenvalues are identical and

$$T = \lambda + N$$

where N is strictly upper triangular and therefore nilpotent of index n, i.e. $N^n = 0$, $N^{n-1} \neq 0$.

> <u>Theorem 2</u>. Let $T = \lambda + N$ and suppose that ϕ has $(n-1)$ continuous derivatives at λ. Then
>
> $$[\phi(T)]_{1n} = \sum_{j=1}^{n-1} (N^j)_{1n} \phi^{(j)}(\lambda)/j! \quad ,$$
>
> where
>
> $$(N^j)_{1n} = \tau_j^{1n} = \sum_{\sigma} t_{\sigma_0\sigma_1} t_{\sigma_1\sigma_2} \cdots t_{\sigma_{j-1}\sigma_j}$$
>
> and σ ranges over the set $S_{1n}(j)$ of all strictly increasing sequences of integers with $\sigma_0 = 1$, $\sigma_j = n$.

<u>Proof</u>.

$$z - T = (z-\lambda)[1 - (z-\lambda)^{-1}N]$$

$$(z-T)^{-1} = [1 - (z-\lambda)^{-1}N]^{-1}/(z-\lambda)$$

$$= \sum_{j=0}^{n-1} N^j/(z-\lambda)^{j+1} \quad .$$

Recall [1, p. 45] that

$$\phi^{(j)}(\lambda)/j! = \frac{1}{2\pi i}\oint_{\Gamma}\phi(z)dz/(z-\lambda)^{j+1} \quad .$$

Multiply $(z-T)^{-1}$ by $\phi(z)/2\pi i$ and integrate round Γ to obtain

$$\begin{aligned}\phi(T) &= \frac{1}{2\pi i}\oint_{\Gamma}\phi(z)(T-z)^{-1}dz \\ &= \sum_{j=0}^{n-1} N^j\phi^{(j)}(z)/j! \quad .\end{aligned}$$

From the very definition of matrix multiplication restricted to strictly upper triangular matrices

$$(N^j)_{1n} \equiv \tau_j^{1n} = \sum_{\sigma} t_{\sigma_0\sigma_1}\cdots t_{\sigma_{j-1}\sigma_j} \quad .$$

Note that the term $j = 0$ contributes nothing to the $(1,n)$ element of $\phi(T)$ and so may be omitted. In other words $\tau_0^{1n} = 0$, by convention.

□

For Theorem 2 to be useful it must be possible to compute $\phi^{(j)}(\lambda)$ and we assume that such facilities are available. We observe that Theorem 2 is the special case of Newton's interpolating formula when all the λ_i coincide.

If an extra array of storage is available, beyond that required for T and $F = \phi(T)$, then there is a straightforward way to compute F. Let N be the array which will contain the successive N^j. The following algorithm is self explanatory.

Step 1. $N \leftarrow I$, $F \leftarrow 0$, $\phi(\lambda)I$.

Step 2. For $j = 1,\ldots,n-1$

$$N \leftarrow N(T-\lambda)$$

$$F \leftarrow F + N\phi^{(j)}(\lambda)/j!$$

The relevant part of N shrinks by one diagonal at each step.

Cost. At step j the last $n-j+1$ rows of N are zero. The step $N \leftarrow N(T-\lambda)$ requires $\binom{n-j+2}{3}$ multiplications. Summing this, for $j = 2,\ldots,n-2$, yields a total of $\binom{n+1}{4}$ multiplications for all the N's. This is $O(n^4)$ but the leading coefficient is $1/24$. The second steps, $F \leftarrow F + \cdots$, require a total of $\binom{n}{3}$ multiplications, a lower order term.

In the next section we describe a slightly more complicated algorithm which costs no more and requires no extra storage. Essentially it executes the above algorithm on one row of F and N at a time.

5. The τ Array

In this section we describe what amounts to a programming device for building up the coefficients $\tau_j^{r,s}$ in a manner that is economical in both storage and in arithmetic operations. We leave the special case $r = 1$, $s = n$ and observe that $\tau_j^{r,s}$ is a sum of products of t's and we can group together those products with the same last term $t_{\sigma_{j-1}\sigma_j}$. Formally

$$\begin{aligned}\tau_j^{r,s} &\equiv \sum_{\sigma\,\varepsilon\, S_j(r,s)} t_{\sigma_0\sigma_1} t_{\sigma_1\sigma_2} t_{\sigma_2\sigma_3} \cdots t_{\sigma_{j-1}\sigma_j}\,, \qquad \sigma_0 = r,\quad \sigma_j = s\,, \\ &= \sum_{k=r+1}^{s-1} \tau_{j-1}^{r,k}\, t_{ks}\,. \end{aligned} \tag{1}$$

In order to build up $\tau_j^{r,s}$ for $j = 1,\ldots,s-r$, using (1), it will be necessary to form and hold $\tau_m^{r,k}$ for $m = 1,\ldots,k-r$; $k = r+1,\ldots,s-1$. In practice these τ values can all be stored in that portion of the array $F = \phi(T)$ which is subtended by the (r,s) element.

$$\begin{array}{ccccccc}
x & \tau_1^{r,r+1} & \tau_1^{r,r+2} & \tau_1^{r,r+3} & \cdot & \cdot & \tau_1^{r,s}\\
 & x & \tau_2^{r,r+2} & \tau_2^{r,r+3} & \cdot & \cdot & \tau_2^{r,s}\\
 & & x & \tau_3^{r,r+3} & \cdot & \cdot & \tau_3^{r,s}\\
 & & & x & \cdot & \cdot & \tau_4^{r,s}
\end{array}$$

The array is built column by column as indicated above, the first row is given by $\tau_1^{r,k} = t_{r,k}$, $r < k \leq s$.

<u>Cost</u>. For a confluent block with m rows, $m \geq 2$:

$\binom{m-1}{2}$ operations for column m of τ for row 1.

$\binom{m}{3}$ operations for the whole τ array for row 1.

$\binom{m-1}{2}$ additional operations for the formation of row 1.

$\binom{m+1}{4} + \binom{m}{3}$ operations for the whole confluent block.

6. <u>Functions of Matrix Pencils</u>

An initial value problem which arises in many applications has the form

$$B\dot{u}(t) = Au(t) , \qquad u(0) = u_o$$

where A and B are constant. When B is invertible the solution is, formally,

$$u(t) = \exp(B^{-1}At)u_o \quad .$$

When working with floating point arithmetic of fixed precision the explicit inversion of B may sometimes provoke unnecessarily large roundoff errors.

The next theorem shows how $\exp(B^{-1}At)$ can be computed without inverting B. The storage requirements of this phase are doubled but the "extra" arithmetic operations are saved by avoiding the formation of $B^{-1}At$. The method is not confined to the exponential function.

The first step is to reduce A and B to upper triangular form by unitary transformations. This is accomplished by the QZ algorithm [4].

<u>Theorem 3</u>. Let A and B^{-1} be upper triangular and let ϕ be a scalar function for which $F = \phi(B^{-1}A)$ and $\tilde{F} = \phi(AB^{-1})$ are well defined. The elements of the upper triangular matrices F and $\tilde{F}$ are given by

$$f_{jj} = \tilde{f}_{jj} = \phi(a_{jj}/b_{jj}) \ , \qquad j = 1,\ldots,n \quad ,$$

and, for $r < s$,

$$\begin{aligned} a_{rr}f_{rs} - \tilde{f}_{rs}a_{ss} &= (\tilde{f}_{rr}-f_{ss})a_{rs} \\ &\quad + \sum_{k=1}^{s-r-1} (\tilde{f}_{r,r+k}a_{r+k,s} - a_{r,s-k}f_{s-k,s}) \\ b_{rr}f_{rs} - \tilde{f}_{rs}b_{ss} &= (\tilde{f}_{rr}-f_{ss})b_{rs} \\ &\quad + \sum_{k=1}^{s-r-1} (\tilde{f}_{r,r+k}b_{r+k,s} - b_{r,s-k}f_{s-k,s}) \quad . \end{aligned}$$

<u>Proof</u>. Since $AB^{-1} = B(B^{-1}A)B^{-1}$, $\tilde{F} = BFB^{-1}$. By Lemma 2,

$$B^{-1}AF = FB^{-1}A \ , \qquad AB^{-1}\tilde{F} = \tilde{F}AB^{-1} \quad .$$

Thus

$$AF - \tilde{F}A = 0 \ , \qquad BF - \tilde{F}B = 0 \quad . \tag{1}$$

Moreover F and $\tilde{F}$ are upper triangular (by Lemma 1). On equating the (r,s) element on each side of (1) the given relations are obtained. □

By tolerating the storage of the auxiliary matrix $\tilde{F}$ the pair $F, \tilde{F}$ can be built up together from the diagonal outwards, <u>without ever inverting</u> B. When the two linear equations do not have a unique solution, i.e. when

$$a_{rr}b_{ss} - a_{ss}b_{rr} = \det\begin{pmatrix} a_{rr} & a_{ss} \\ b_{rr} & b_{ss} \end{pmatrix} = 0 \quad ,$$

then the confluent form, involving derivatives of ϕ, must be used.

7. Programs and Examples[†]

```
      SUBROUTINE FUNUPPD(R,S,T,F,MM,PHI)
      DIMENSION T(MM,MM), F(MM,MM)
      INTEGER R,S
C  THE FUNCTION (SUBPROGRAM) PHI OF THE BLOCK IN ROWS R THROUGH S OF
C        THE MM X MM  UPPER TRIANGULAR MATRIX T IS STORED IN F
C  THE DIAGONAL ELEMENTS ARE ASSUMED TO BE DISTINCT.  IF THIS CONDITION
C        IS VIOLATED EXECUTION WILL BE INTERRUPTED AND MM WILL BE SET TO
C        -MM AS A FLAG
C
C  INSERT FUNCTION VALUES ON THE DIAGONAL
      DO 10 I=R,S
10    F(I,I) = PHI(T(I,I))
C  PROCESS THE KTH SUPERDIAGONAL
      N = S-R+1
      NN= N-1
      IF (NN.EQ.0) RETURN
      DO 13 K=1,NN
      LL = R - 1 + N - K
         DO 12 I=R,LL
         DIFF = T(I,I)-T(I+K,I+K)
         IF (ABS(DIFF) .EQ. 0.0) GOTO 14
         G= T(I,I+K) * (F(I,I) - F(I+K,I+K))
         KK = K-1
         IF (KK.EQ.0) GOTO 12
            DO 11 M=1,KK
11          G = G + (F(I,I+M)*T(I+M,I+K) - T(I,I+K-M)*F(I+K-M,I+K))
12       F(I,I+K) = G/DIFF
13    CONTINUE
      RETURN
14    MM = -MM
      RETURN
      END
```

[†] The author wishes to thank Mr. Ron Feldman for programming assistance.

```
      SUBROUTINE FUNUPPC(R,S,T,F,MM,PHI)
      DIMENSION T(MM,MM),F(MM,MM)
      INTEGER R,RR,S,SS
C  CONFLUENT CASE - EQUAL DIAGONAL ELEMENTS
C  THE FUNCTION (SUBPROGRAM) PHI OF THE BLOCK IN ROWS R THROUGH S OF
C        THE MM X MM UPPER TRIANGULAR MATRIX T IS STORED IN F
C PHI(M,X) IS THE MTH DERIVATIVE OF PHI(X), THE GIVEN FUNCTION
C
      FAC = 1
      F(R,R) = PHI(0,T(R,R))
      IF (R.EQ.S) RETURN
      RR = R+1
      DO 10 K=RR,S
      FAC = FAC * (K-R)
10    F(K,K) = PHI(K-R,T(R,R))/FAC
C   THE DIAGONAL IS A CONVENIENT STORE FOR DERIVATIVES
C
C   GENERATE TAU-SIGMAS
      SS = S-1
      DO 16 I=R,SS
      II = I+1
         DO 13 N=II,S
         F(I,N) = T(I,N)
         NN = N-1
         IF (NN.LT.II) GOTO 13
            DO 12 K=II,NN
            G=0
               DO 11 M=K,NN
               L = K-1
11             G = G+ F(L,M)*T(M,N)
               F(K,N) = G
12          CONTINUE
13       CONTINUE
C  F NOW CONTAINS THE CORRECT COEFFICIENTS
         DO 15 N=II,S
         G =0
         NN=N-1
            DO 14 K=I,NN
            M = K-I+R+1
14          G = G + F(K,N)*F(M,M)
         F(I,N) = G
15       CONTINUE
16    CONTINUE
C  RESTORE DIAGONAL
      RR=R+1
      DO 17 K=RR,S
17    F(K,K) = F(R,R)
      RETURN
      END
```

Example 1. Distinct Eigenvalues

```
R =   2          S=   4
PHI = X*X + 3.0*X + 2.0
```

UPPER TRIANGULAR MATRIX T

3.00	-2.00	0.	1.00	-2.00
0.	2.00	4.00	3.00	2.00
0.	0.	1.00	5.00	1.00
0.	0.	0.	-4.00	1.00
0.	0.	0.	0.	2.00

FUNCTION MATRIX F

0.	0.	0.	0.	0.
0.	12.00	24.00	23.00	0.
0.	0.	6.00	0.	0.
0.	0.	0.	6.00	0.
0.	0.	0.	0.	0.

Example 2. Confluent Case

```
R =   2          S=   4
PHI = X*X + 2.0*X + 2.0
```

UPPER TRIANGULAR MATRIX T

3.00	-2.00	0.	1.00	-2.00
0.	2.00	4.00	3.00	-4.00
0.	0.	2.00	5.00	1.00
0.	0.	0.	2.00	1.00
0.	0.	0.	0.	1.00

FUNCTION MATRIX F

0.	0.	0.	0.	0.
0.	10.00	24.00	38.00	0.
0.	0.	10.00	30.00	0.
0.	0.	0.	10.00	0.
0.	0.	0.	0.	0.

References

[0] EISPACK Guide, Lectures Notes in Computer Science No. 6, Springer-Verlag (1974).

[1] W. Kahan, "Conserving Confluence Curbs Ill Condition," Technical Report No. 6, Computer Science Department, University of California, Berkeley (1972).

[2] T. Kato, Perturbation Theory for Linear Operators, Springer-Verlag (1966).

[3] C.C. MacDuffee, The Theory of Matrices, Chelsea Publishing Co. (1956).

[4] C.B. Moler and G.W. Stewart, "An Algorithm for Generalized Matrix Eigenvalue Problems," SIAM Journal of Numerical Analysis 10 (1973), 241-256.

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Computer Science Division University of California Berkeley, California 94720	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

COMPUTATION OF FUNCTIONS OF TRIANGULAR MATRICES

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Scientific Final

5. AUTHOR(S) (First name, middle initial, last name)

B.N. Parlett

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
November 1974	18	5

8a. CONTRACT OR GRANT NO. ONR-N00014-69-A-0200-1017	9a. ORIGINATOR'S REPORT NUMBER(S) Electronics Research Laboratory Memorandum M-481
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Mathematics Branch Office of Naval Research Washington, D.C. 20360

13. ABSTRACT

A simple relation exists among the elements of $\phi(T)$ when ϕ is an analytic function and T is triangular. This permits the rapid build up of $\phi(T)$ from its diagonal. An analogous relation holds for block triangular matrices. This permits the formation in real arithmetic of real functions of real matrices with complex eigenvalues. The confluent case is included. Algorithms are given and some numerical examples.

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
function of a matrix						
triangular matrix						

Unclassified

Security Classification