Grumman Research Department Memorandum RM-386

N68-13614

PARAMETERIZATION OF THE SET OF POSITIVE-DEFINITE MATRICES AND AN ALGORITHM FOR ITS GENERATION

by

D. Rothschild

and

G. R. Geiss

Systems Research Section

November 1967

This work was performed under Contract NAS 2-4063 from NASA Ames Research Center and Contract NAS 8-20306 from NASA George C. Marshall Space Flight Center.

Approved by: Charles E. Mack, Jr.
Charles E. Mack, Jr.
Director of Research

ABSTRACT

An algorithm is presented for generating the set of $n \times n$ positive-definite symmetric matrices, based on the choice of $\frac{n(n+1)}{2}$ arbitrary parameters.

This method was developed as a necessary step in the development of an approach to the problem of estimating the domain of attraction of an equilibrium solution to a system of nonlinear autonomous differential equations. The technique however, is general, and may be applied to any problem requiring this type of matrix.

TABLE OF CONTENTS

Item	Page
Introduction	1
Development of the Numerical Algorithm	2
Computer Program	8
References	9
Appendix - Listing of Computer Program	10

INTRODUCTION

In conjunction with research being done on the formulation of numerical techniques for estimating the domain of attraction of an equilibrium solution to a nonlinear dynamical system, the need arose for an efficient algorithm to generate a set of $n \times n$ positive-definite symmetric matrices. These matrices can be used as candidate "Q" matrices which in turn give rise to associated "P" matrices through the solution of the Liapunov equation, $A^T P + PA = - Q$. Having parameterized the Q set, one can proceed in an orderly fashion in pursuit of an "optimal" quadratic form Liapunov function to resolve the domain of stability problem (see Ref. 1).

Another problem area for which this matrix generation technique has potential application is in the class of optimization problems where the so-called conjugate direction methods are employed (Refs. 2 and 3). In many of these methods, the initial arbitrary or random choice of a positive-definite symmetric matrix is required.

This memorandum discusses a procedure and associated computer program for satisfying the above requirements.

DEVELOPMENT OF THE NUMERICAL ALGORITHM

The generation of the set of positive-definite $n \times n$ symmetric matrices can be carried out by resorting to the brute force approach of forming an arbitrary $n \times n$ symmetric matrix and then applying the determinantal test (Ref. 4) for positive-definiteness. The arbitrary choice of $\frac{n(n+1)}{2}$ matrix elements followed by the evaluation of the determinants of the n-principal minors would be necessary. It would be desirable to generate these matrices by a procedure that guarantees all to be positive-definite, and in addition, that the entire set of positive-definite matrices be spanned.

It is well known (Ref. 4) that all real symmetric matrices are orthogonally similar to a diagonal matrix, and that all positive-definite (pd) matrices are then orthogonally similar to a diagonal matrix with positive-diagonal elements; i.e., let Q be pd, then

$$Q = S^T \Lambda S \ , \tag{1}$$

where

$$\begin{aligned} \Lambda &= \operatorname{diag}\left\{\lambda_1, \lambda_2, \ldots, \lambda_n\right\} \\ \lambda_i &> 0 \quad , \quad i = 1, 2, \ldots, n \\ S^T S &= I \ . \end{aligned} \tag{2}$$

Thus, the parameterization of all pd matrices, Q, is reduced to the parameterization of the group of orthogonal matrices, S.

In Ref. 5, Murnaghan proves that the parameterization of the group of $n \times n$ unitary matrices U is accomplished by the factorization

$$U = D \left[\prod_{k=1}^{n-1} U_{n-k} \right] = D \times U_{n-1} \times \ldots \times U_1 , \tag{3}$$

where

$$D = \mathrm{diag} \left\{ e^{i\delta_1}, e^{i\delta_2}, \ldots, e^{i\delta_{n-1}}, e^{i\varphi_n} \right\} , \tag{4}$$

$$U_k = \left[\prod_{\ell=k+1}^{n-1} U_{k\ell}(\theta_\mu, \sigma_\rho) \right] \left[U_{kn}(\varphi_k, \sigma_\gamma) \right] , \tag{5}$$

$$\gamma = \frac{(2n - k)(k - 1)}{2} + 1 ,$$

$$\rho = \frac{(2n - k)(k - 1)}{2} + 1 + n - \ell ,$$

$$\mu = \frac{(2n - k - 2)(k - 1)}{2} + (n - \ell) ,$$

$$U_{k\ell}(\theta, \sigma) = (u_{ij}) : \begin{cases} u_{ii} = 1, \ i \neq k, \ \ell \\ u_{kk} = \cos\theta \\ u_{\ell\ell} = \cos\theta \\ \\ u_{ij} = 0 \ , \ i \neq j \ , \ i, j \neq k, \ \ell \\ u_{k\ell} = - e^{-i\sigma} \sin\theta \\ u_{\ell k} = + e^{+i\sigma} \sin\theta \ , \end{cases} \tag{6}$$

$$-\pi \leq \phi < \pi \ , \quad -\frac{\pi}{2} \leq \sigma \leq \frac{\pi}{2} \ , \quad -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2} \ , \quad -\frac{\pi}{2} \leq \delta \leq \frac{\pi}{2} \ .$$

The factorization of the group of orthogonal matrices is immediately obtained by requiring U to be real; i.e., $\delta = \sigma = 0$, $\varphi_n = \pm\pi$, $-\pi \leq \varphi_k < \pi$, $k \neq n$, and $-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$. In particular,

$$S = D_1 \left[\prod_{k=1}^{n-1} S_{n-k} \right] , \tag{7}$$

$$D_1 = \mathrm{diag} \left\{ 1, \ \ldots, \ 1, \ \pm 1 \right\} , \tag{8}$$

$$S_k = \left[\prod_{\ell=k+1}^{n-1} S_{k\ell}(\theta_\mu) \right] \left[S_{kn}(\varphi_k) \right] \quad , \quad S_{k\ell}(\theta_\mu) = U_{k\ell}(\theta_\mu, 0)$$

$$\mu = \frac{(2n - k - 2)(k - 1)}{2} + n - \ell \ . \tag{9}$$

This factorization contains $\frac{(n-1)(n-2)}{2}$ thetas and n phis, or a total of $\frac{n(n-1)}{2} + 1$ parameters. The n lambdas in Eq. (2) raise the number of parameters to $\frac{n(n+1)}{2} + 1$, or one more than required. Thus, if we restrict S to be a rotation matrix (i.e., choose $\varphi_n = 0$), the number of parameters will be $\frac{n(n+1)}{2}$, the number required to represent an arbitrary symmetric matrix. The choice $\varphi_n = 0$ is intuitively motivated by the consideration that we wish to rotate and scale the ellipsoid associated with the quadratic form formed from the pd matrix and do not want to reflect coordinates or change the handedness of the coordinate system.

The factorization of a pd matrix of dimension three is thus given by

$$P = S^T \Lambda S \ , \tag{10}$$

where

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \ , \tag{11}$$

$$\lambda_1, \lambda_2, \lambda_3 > 0 \ ,$$

and

$$S = S_2 S_1 = S_{23}(\varphi_2) S_{12}(\theta_1) S_{13}(\varphi_1) \ , \tag{12}$$

$$S_{23} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c\varphi_2 & -s\varphi_2 \\ 0 & s\varphi_2 & c\varphi_2 \end{pmatrix} ,$$

$$S_{12} = \begin{pmatrix} c\theta_1 & -s\theta_1 & 0 \\ s\theta_1 & c\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} , \tag{13}$$

$$S_{13} = \begin{pmatrix} c\varphi_1 & 0 & -s\varphi_1 \\ 0 & 1 & 0 \\ s\varphi_1 & 0 & c\varphi_1 \end{pmatrix} ,$$

$$-\pi \leq \varphi_1 < \pi , \; -\pi \leq \varphi_2 < \pi , \; -\frac{\pi}{2} \leq \theta_1 \leq \frac{\pi}{2} ,$$

$$c\varphi_1 = \cos \varphi_1 , \; s\varphi_1 = \sin \varphi_1 .$$

Thus, it is clear that by using this representation under the restrictions

$$\begin{aligned} &\lambda_i > 0 , && i = 1, 2, \ldots, n \\ &-\pi \leq \varphi_i < \pi , && i = 1, 2, \ldots, n - 1 \\ &-\frac{\pi}{2} \leq \theta_i \leq \frac{\pi}{2} , && i = 1, 2, \ldots, \frac{(n-1)(n-2)}{2} , \end{aligned} \tag{14}$$

the candidate Q matrices are guaranteed to be positive-definite. The constraints, Eq. (14), can be removed by defining:

$$\lambda_i = e^{\eta_i} \quad , \qquad \eta_i \text{ real}$$
$$\theta_i = \frac{1}{2}(-\pi + \theta_i' \bmod 2\pi) \quad , \qquad \theta_i' \text{ real} \tag{15}$$

and recognizing that since the φ_i's only appear as arguments of trigometric functions, they can be arbitrary real numbers.

COMPUTER PROGRAM

A computer program which implements the preceding numerical algorithm has been developed and is being used in conjunction with the problem of finding an "optimal" quadratic Liapunov function for a 9th order quasi-linear differential equation. The dimension of the problem, however, is read as data and is arbitrary.

To use the program, one must specify the parameters: η_1; η_2; ... η_n; θ_1'; θ_2'; ... θ_k'; φ_1; φ_2; ... φ_{n-1}, where the restrictions of Eq. (15) hold. The names given to these arrays are:

η array $\equiv$ XLAM (I)

θ' array $\equiv$ THETA (I)

φ array $\equiv$ PHI (I) .

If it is desired to print out the orthogonal matrix, S, which is used in Eq. (11), the control variable PP is set equal to 1.0, otherwise it is set equal to zero. This matrix [S] is called B(I, J) in the program. The final required output of the program is Q(I, J), an n x n positive-definite symmetric matrix. Enclosed in the appendix is the complete computer program listing.

REFERENCES

1. Geiss, G. R. and Abbate, J. V., Study On Determining Stability Domains for Nonlinear Dynamical Systems, Grumman Research Department Report RE-282, February 1967.

2. Kelley, H. J. and Myers, G. E., Conjugate Direction Methods for Parameter Optimization, presented at 18th Congress of the International Astronautical Federation, Belgrade, Yugoslavia, 1967.

3. Kopp, R. E., McGill, R., Moyer, H. G., and Pinkham, G., "Several Trajectory Optimization Techniques," in Computing Methods in Optimization Problems, A. V. Balakrishnan and L. W. Neustadt, Editors, Academic Press, New York, N.Y., 1964.

4. Gantmacher, F. R., The Theory of Matrices, Vol. I, Chelsea Publishing Co., New York, 1959.

5. Murnaghan, F. D., Lectures on Applied Mathematics, Vol. III: The Unitary and Rotation Groups, Spartan Books, Washington, D.C., 1962.

APPENDIX

LISTING OF COMPUTER PROGRAM

```
COMPILER OPTIONS - NAME= $MAIN,OPT=00,LINECNT=50,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID

                C     GENERATION OF POSITIVE DEFINITE Q MATRIX
                C
ISN 0002              DIMENSION   THETA(28),PHI(8),XLAM(9),ZTHETA(28),A(20,20),
                     1            BA(20,20),SS(20,20,20),C(20,20),Z(20,20,20),Q(20,20),
                     2            B(20,20),S(20,20),AA(20,20),G(20,20),QQ(20,20),
                     3            TTHETA(28),DA(45)
ISN 0003           32 READ(5,1)(THETA(I),I=1,28),(PHI(I),I=1,8),(XLAM(I),I=1,9),PP
ISN 0004            1 FORMAT(6E12.4)
ISN 0005           33 READ(5,5)IDEM
ISN 0006            5 FORMAT(I3)
ISN 0007              WRITE(6,3)THETA,PHI,XLAM,PP
ISN 0008            3 FORMAT(23H DATA-THETA,PHI,XLAM,PP/(6E15.7))
ISN 0009              WRITE(6,4)IDEM
ISN 0010            4 FORMAT(11H DIMENSION=,I3)
ISN 0011              N=IDEM
ISN 0012              NN=(N-1)*(N-2)/2
ISN 0013              DO 6I=1,NN
ISN 0014              BAD=THETA(I)
ISN 0015            6 TTHETA(I)=AMOD(BAD,1.5708)
ISN 0016              DO 27 I=1,N
ISN 0017           27 XLAM(I)= EXP(XLAM(I))
                C     WE HAVE NOW INDEXED THETA.
                C     NOW WANT CONTINUED PRODUCT OF SS(I,J,L) FOR L=K&1,N .
                C     FOR EACH K=1,N-1 OBTAIN Z(K,I,J).
ISN 0018              NNI=N-1
ISN 0019           69 DO 20 K=1,NNI
                C
ISN 0020              DO 8 I=1,N
ISN 0021              DO 8 J=1,N
ISN 0022            8 BA(I,J)=0.0
ISN 0023              DO 99 I=1,N
ISN 0024           99 BA(I,I)=1.0
                C
ISN 0025              KK=K&1
ISN 0026              DO 10 L=KK,N
                C
ISN 0027              DO 15 I=1,N
ISN 0028              DO 15 J=1,N
ISN 0029           15 SS(I,J,L)=0.0
ISN 0030              DO 98 I=1,N
ISN 0031           98 SS(I,I,L)=1.0
                C     WE DEVELOP SS(I,J,L) AS FUNCTION THETA(L,K,N) FOR L L.T. N
                C     AND SS(I,J,L) FUNCTION OF PHI(K) FOR L=N
ISN 0032              IF(L-N)25,23,23
ISN 0033           25 M=((2*N -K-2)*(K-1)/2)&N-L
ISN 0034              SS(K,K,L)=COS(TTHETA(M))
ISN 0035              SS(L,L,L)=COS(TTHETA(M))
ISN 0036              SS(K,L,L)=-SIN(TTHETA(M))
ISN 0037              SS(L,K,L)=SIN(TTHETA(M))
```

```
ISN 0038          GO TO 35
ISN 0039       23 SS(K,K,L)=COS(PHI(K))
ISN 0040          SS(L,L,L)=COS(PHI(K))
ISN 0041          SS(K,L,L)=-SIN(PHI(K))
ISN 0042          SS(L,K,L)=SIN(PHI(K))
            C
ISN 0043       35 DO 70 I=1,N
ISN 0044          DO 70 J=1,N
ISN 0045       70 C(I,J)=0.0
            C
ISN 0046          DO 50 M=1,N
ISN 0047          DO 50 J=1,N
ISN 0048          DO 50 I=1,N
ISN 0049       50 C(M,J)=BA(M,I)*SS(I,J,L) &C(M,J)
ISN 0050          DO110 I=1,N
ISN 0051          DO110 J=1,N
ISN 0052      110 BA(I,J)=C(I,J)
ISN 0053       10 CONTINUE
ISN 0054          DO 20 I=1,N
ISN 0055          DO 20 J=1,N
ISN 0056       20 Z(K,I,J)=BA(I,J)
            C
ISN 0057          DO 7 I=1,N
ISN 0058          DO 7 J=1,N
ISN 0059        7 B(I,J)=0.0
ISN 0060          DO 16 I=1,N
ISN 0061       16 B(I,I)=1.0
            C
ISN 0062          DO 40 K=1,NNI
            C
ISN 0063          DO 75 I=1,N
ISN 0064          DO 75 J=1,N
ISN 0065       75 S(I,J)=0.0
            C
ISN 0066          DO 55 M=1,N
ISN 0067          DO 55 J=1,N
ISN 0068          DO 55 I=1,N
ISN 0069       55 S(M,J)=Z(K,M,I)*B(I,J)&S(M,J)
ISN 0070          DO 40 I=1,N
ISN 0071          DO 40 J=1,N
ISN 0072       40 B(I,J)=S(I,J)
            C
            C     B(I,J) IS CONTINUED PRODUCT OF Z(K,I,J) FROM K=1 TO N-1
            C
ISN 0073          IF(PP)41,41,19
ISN 0074       19 WRITE(6,18)((B(I,J),I=1,N),J=1,N)
ISN 0075       18 FORMAT(7H B(I,J)/(6E15.7))
ISN 0076       41 DO 78 I=1,N
ISN 0077          DO 78 J=1,N
ISN 0078       78 AA(I,J)=B(J,I)
```

```
                C
                C     AA(I,J) IS TRANSPOSE B(I,J)
                C
ISN 0079              DO 82 I=1,N
ISN 0080              DO 82 J=1,N
ISN 0081           82 G(I,J)=0.0
ISN 0082              DO 85 I=1,N
ISN 0083           85 G(I,I)=XLAM(I)
                C     G(I,J) IS THE LAMDA MATRIX
                C
ISN 0084              DO 86 I=1,N
ISN 0085              DO 86 J=1,N
ISN 0086           86 QQ(I,J)=0.0
ISN 0087              DO 88 I=1,N
ISN 0088              DO 88 J=1,N
ISN 0089              DO 88 M=1,N
ISN 0090           88 QQ(I,J)=G(I,M)*B(M,J)&QQ(I,J)
                C
                C     QQ(I,J)=LAMDA MATRIX * B(I,J)
                C
ISN 0091              DO 90 I=1,N
ISN 0092              DO 90 J=1,N
ISN 0093           90 Q(I,J)=0.0
ISN 0094              DO 95 I=1,N
ISN 0095              DO 95 J=1,N
ISN 0096              DO 95 M=1,N
ISN 0097           95 Q(I,J)=AA(I,M)*QQ(M,J)&Q(I,J)
ISN 0098              WRITE(6,93)((Q(I,J),I=1,N),J=1,N)
ISN 0099           93 FORMAT(1H /1X ,7H Q(I,J)/(1X,6E15.7))
ISN 0100              GO TO 32
ISN 0101              END
```