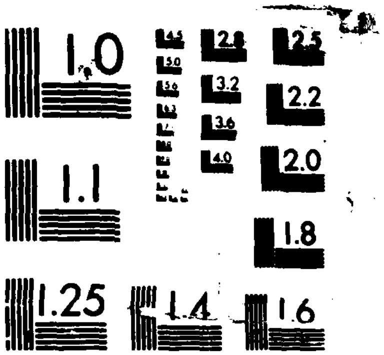
AD-A181 948   A COMPUTER BENCHMARK PROGRAM FOR HILBERT MATRIX   1/1
INVERSION USING IMSL ROUTINES(U) DEFENCE RESEARCH
ESTABLISHMENT ATLANTIC DARTMOUTH (NOVA SCOTIA)
UNCLASSIFIED   J A THERIAULT MAR 87 DREA-TC-87/305   F/G 12/5   NL



MICROCOPY RESOLUTION TEST CHART

3

UNLIMITED DISTRIBUTION

National Defence
Research and Development Branch

Défense Nationale
Bureau de Recherche et Développment

TECHNICAL COMMUNICATION 87/305
March 1987

A COMPUTER BENCHMARK PROGRAM FOR HILBERT MATRIX INVERSION USING IMSL ROUTINES

James A. Theriault

Defence
Research
Establishment
Atlantic

Centre de
Recherches pour la
Défense
Atlantique

Canada

PLEASE ADDRESS FURTHER ENQUIRES TO:

THE CHIEF,
DEFENCE RESEARCH ESTABLISHMENT ATLANTIC,
P.O.BOX 1012, DARTMOUTH, N.S., B2Y 3Z7
CANADA.

DEFENCE RESEARCH ESTABLISHMENT ATLANTIC

9 GROVE STREET

P.O. BOX 1012
DARTMOUTH, N.S.
B2Y 3Z7

TELEPHONE
(902) 426-3100

CENTRE DE RECHERCHES POUR LA DÉFENSE ATLANTIQUE

9 GROVE STREET

C.P. 1012
DARTMOUTH, N.É.
B2Y 3Z7

UNLIMITED DISTRIBUTION

National Defence
Research and
Development Branch

Défense nationale
Bureau de recherche
et développement

A COMPUTER BENCHMARK PROGRAM FOR HILBERT MATRIX INVERSION USING IMSL ROUTINES

James A. Theriault

March 1987

Approved by A. Mohammed H/APPLIED MATHEMATICS SECTION

DISTRIBUTION APPROVED BY

D/UAD

TECHNICAL COMMUNICATION 87/305

Defence
Research
Establishment
Atlantic

Centre de
Recherches pour la
Défense
Atlantique

Canada

Abstract

In considering potential replacements for computer facilities, such as that for the DREA DECSYSTEM 20/60, comparison standards must be specified. In the comparison standards, two notable characteristics that should be specified are the speed and accuracy of numerical computations. HILBRT is a benchmarking program which may be used for this purpose. It is based on the inversion of finite segments of the infinite Hilbert matrix. The theory, program, and example results are presented.

Résumé

L'examen des systèmes de replacement potentiels pour les systèmes informatiques comme, par exemple, le DECSYSTEM 20/60 du CRDA, exige l'établissement de critères de comparaison. Il est nécessaire de préciser deux caractéristiques importantes dans ces critères de comparaison, c'est-à-dire la vitesse et l'exactitude des calculs numériques. Le programme d'évaluation des performances HLBERT peut servir à cette fin. Il est fondé sur l'inversion de segments finis de la matrice de Hilbert infinie. Le texte expose les principes théoriques, décrit le logiciel et donne un exemple des résultats obtenus.

Contents

1 Introduction

In order to make an objective comparison between computers, one must consider many features. One may consider such things as user friendliness, software availability, hardware availability, speed, and accuracy. This technical communication describes one method of measuring and comparing the features of computer speed and accuracy. The method is based on the inversion of finite segments of the Hilbert matrix.

The Hilbert matrix is a classic matrix in numerical analysis. As the finite segment size increases, the segments become increasingly ill-conditioned. HILBRT is a benchmark program written in ANSI Fortran '77 which inverts finite segments oí the infinite Hilbert matrix using IMSL Version 9.2[IMSL, 1984], then computes accuracy measurements of the solution. Two measures are used. The first is to compute the sup-norm of the difference between the identity matrix and the product of the Hilbert matrix and its IMSL approximated inverse. The second measure is the sup-norm of the difference between the IMSL computed inverse and a computed closed form solution. The results are then written on the file "HIL.DAT." Output consists of the matrix order and both sup-norms for each matrix. These are calculated for matrices starting with order one and increasing by one each time. This is repeated until the IMSL subroutine is unable to find the solution or an order of thirty is reached. Calculations for those matrices of orders one through four are repeated 12000 times. The CPU time required for the repetitions is calculated and written on the output file to allow comparison of systems. The total CPU time required is also included.

In the technical specifications for the replacement of DREA's DECSYSTEM 20/60, the IMSL library is included as an essential requirement so that the use of this library in the benchmark program HILBRT is justified. In essence, the performance characteristics of speed and accuracy depend on the performance of the subroutines in the IMSL library; and, in turn, the performance of the subroutines depends on the computer wordlength and the floating-point format being used. As a result, this benchmark measures speed and accuracy in an IMSL environment.

2 Operational Section

2.1 Identification

PROGRAM HILBRT

A Fortran program to measure computer speed and accuracy. It will be one of many benchmark programs used to assist in the selection of a new DREA central computing facility.

Author: James A. Theriault

2.2 Hardware/Software Environment

HILBRT requires the following software:

- ANSI Fortran '77

- Fortran subroutine LINV2F from IMSL Version 9.2

- Fortran subroutine LEQT2F from IMSL Version 9.2

- Fortran subroutine CPUTIM from DEC 20/60 library. This may be replaced by similar routines on other computers.

HILBRT may be run in batch mode. There is no user input. All output is directed to a named disk file, "HIL.DAT."

2.3 Purpose

The purpose of HILBRT is to benchmark computers in an IMSL environment for the DEC 20/60 replacement. It does so by inverting finite segments of the infinite Hilbert matrix.

2.4 Outline of Working

The finite segment of order n of the Hilbert matrix is given by [Gregory et al, 1969]:

$$A_n = [a_{i,j}^{(n)}] \tag{2.1}$$

where

$$a_{i,j}^{(n)} = \frac{1}{i+j-1}. \tag{2.2}$$

Its inverse may be given in a closed form as shown in section 3.1. The inverse may also be found using various numerical techniques such as the Crout algorithm for Gaussian elimination utilized by the IMSL routine, LEQT2F.

HILBRT uses both methods to compute the inverse, then compares the results for accuracy.

2.5 Input

The program requires no user input.

2.6 Output

The following is the output file, 'HIL.DAT,' from the DREA DEC 20/60.

```
ORDER   NORM OF A*D-I  NORM OF D-B
  1      .00000E+00     .00000E+00
  2      .00000E+00     .23842E-06
  3      .47684E-06     .85831E-04
  4      .76294E-05     .13245E-01
  5      .24414E-03    51.516
  6      .78125E-02    31173.
  7      .18750         .17929E+08

MATRIX SIZE    8 UNABLE TO BE INVERTED.   IMSL ERROR # 129

TIME AFTER   12000 REPETITIONS OF THE FIRST FOUR ORDERS = 4.984
TOTAL CPU TIME = 4.991  MINUTES
```

A refers to the finite segment of the Hilbert matrix. I refers to the identity matrix. D and B refer to the IMSL calculated approximation and the closed form approximation of the inverse respectively.

2.7 Operation

The sequence of monitor instructions for executing HILBRT on the DREA DEC 20/60 is given by

```
@COMPILE DREA:<MAG.THERIAULT>HILBRT.FOR
@LOAD DREA:<MAG.THERIAULT>HILBRT, <LIB>DECLIB/LIB, <LIB>IMSL/LIB
@SAVE
@RUN HILBRT
```

A similar sequence of commands should be available on all benchmarked computers. Care should be taken to ensure that the calculated CPU time does not include time for compile and load operations. *Compiler optimization must not be used.*

HILBRT requires no further user input.

2.8 Restrictions & Limitations

A maximum size for the finite segments of the Hilbert matrix is 30 by 30. This may be changed by modifying the value of NMAX in the parameter statement of the program. It is assumed that the computer can handle Hilbert matrix segments up to order $n = 4$.

2.9 Subroutines Called

The following subroutines and functions are called by HILBRT.

CMPINV - Subroutine to compute inverse of Hilbert matrix and appropriate norms. It has been supplied with the program.

PERM - Real function which computes a permutation. It has been supplied with the program.

CPUTIM - A DEC 20/60 dependent subroutine to return the CPU time in milliseconds since the start of execution. Systems to be compared should have a similar subroutine.

LINV2F - IMSL version 9.2 subroutine to invert matrix. Calls IMSL subroutine LEQT2F to perform inversion.

2.10 Memory Requirements

The following is a list of the files involved and their sizes on the DREA DEC 20. HIL.DAT is the output file. HILBRT.FOR is the Fortran source for HILBRT. HILBRT.REL and HILBRT.EXE are the relocatable and the saved executable versions of HILBRT. IMSL.REL is the IMSL library.

FILE	Characters(7 bit)
DREA:<MAG.THERIAULT>HIL.DAT	457
DREA:<MAG.THERIAULT>HILBRT.FOR	2958

FILE	Words(36 bit)
DREA:<MAG.THERIAULT>HILBRT.EXE	8704
DREA:<MAG.THERIAULT>HILBRT.REL	1427
PS:<LIB>IMSL.REL	400267

2.11 Execution Times

HILBRT requires approximately five minutes of CPU time on the DEC 20/60 to complete an execution. Subroutine CPUTIM is accurate within 10%.

3 Technical Section

3.1 Details of Working

The finite segment of order n of the infinite Hilbert matrix is given by [Gregory et al, 1969]:

$$A_n = [a_{i,j}^{(n)}] \tag{3.1}$$

where

$$a_{i,j}^{(n)} = \frac{1}{i+j-1} \tag{3.2}$$

for $1 \le i \le n$ and $1 \le j \le n$.

Its inverse may be found numerically. It is also available in the following closed form [Gregory et al, 1969].

$$\begin{aligned} B_n &= A_n^{-1} && (3.3) \\ &= [b_{i,j}^{(n)}], && (3.4) \end{aligned}$$

where

$$b_{i,j}^{(n)} = (-1)^{i+j} \frac{(n+i-1)!(n+j-1)!}{(i+j-1)\left((i-1)!(j-1)!\right)^2 (n-i)!(n-j)!}. \tag{3.5}$$

This can be computed for successive values of n by using the recursion,

$$b_{i,j}^{(n+1)} = b_{i,j}^{(n)} \left(\frac{n+i}{n-i+1}\right)\left(\frac{n+j}{n-j+1}\right) \tag{3.6}$$

for $1 \le i \le n$ and $1 \le j \le n$. Additional elements, $b_{i,j}^{(n+1)}$ may be computed as follows:

$$\begin{aligned} b_{n+1,j}^{(n+1)} &= b_{j,n+1}^{(n+1)} && (3.7) \\ &= (-1)^{n+j+1}\left(\frac{(2n+1)!/n!}{(j-1)!(n-j+1)!}\right)\left(\frac{(n+j-1)!/n!}{(j-1)!}\right) && (3.8) \end{aligned}$$

for $1 \le j \le n+1$.

HILBRT uses this and the IMSL subroutine LEQT2F to compute the inverse. The two methods are compared and the results of the comparison are returned as output.

Define the sup-norm, $||C||_\infty = \sup_{i,j}|c_{i,j}| = \max_{i,j}|c_{i,j}|$. Let A_n be defined to be a finite segment of the Hilbert matrix as above. Let D_n be the computed approximation of the inverse using LEQT2F and B_n be the inverse computed using the above closed form.

The first error measure is the sup-norm of the difference between the identity matrix and the product of the Hilbert matrix A_n and its IMSL approximated inverse D_n:

$$\begin{aligned} \epsilon_1^{(n)} &= \|A_n D_n - I_n\|_\infty \tag{3.9}\\ &= \|E\|_\infty \tag{3.10}\\ &= \max_{i,j} |e_{i,j}| \tag{3.11} \end{aligned}$$

where I_n is the identity matrix of order n.

The second error is the sup-norm of the difference between the IMSL approximated inverse D_n and the closed form inverse B_n:

$$\begin{aligned} \epsilon_2^{(n)} &= \|D_n - B_n\|_\infty \tag{3.12}\\ &= \|U\|_\infty \tag{3.13}\\ &= \max_{i,j} |u_{i,j}|. \tag{3.14} \end{aligned}$$

The computation of the inverses and comparisons are repeated 12000 times for matrices of orders one through four. After this the computations continue for increasing n until the IMSL subroutine is unable to compute the inverse or until a maximum order of thirty is reached.

3.2 Flowchart

Figure 3.1 show a flowchart of the program HILBRT.

A DEC 20/60 dependent subroutine, CPUTIM, is utilized to return the CPU time in milliseconds. CPUTIM is called at the start of HILBRT to return the CPU time since the start of the job. It is called again after the first four matrices have been inverted 12000 times to return the time after execution. It is called a final time after all calculations are complete in order to compute the total CPU time. The elapsed times are then calculated in minutes and written on the output file.

A subroutine similar to CPUTIM is usually available on most computers.

3.3 Important Variables

`N`	matrix order
`M`	matrix order (= `N`)
`NMAX`	maximum matrix order
`A(NMAX,NMAX)`	finite segment of Hilbert matrix
`B(NMAX,NMAX)`	inverse of A computed from closed form solution
`D(NMAX,NMAX)`	inverse of A computed by LEQT2F
`NORMI`	$\|\|$ `A D - I` $\|\|_\infty$

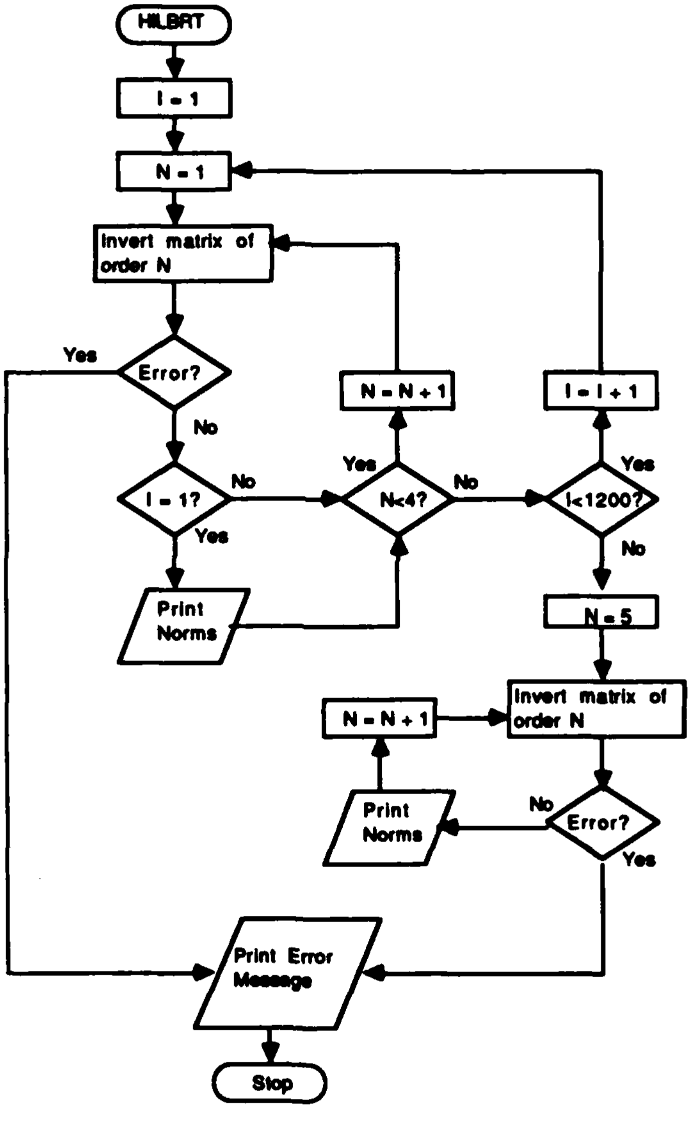


Figure 3.1: Flowchart of HILBRT Program

NORMD	$\|D - B\|_\infty$
ITIM	time at start of job (in msec)
ITIM1	time at end of job (in msec)
ITIM2	time after 12000 repetitions of first 4 matrices (in msec)

4 Summary

HILBRT is a program for benchmarking computers. It has been written in ANSI Fortran '77 with the addition of a call to a DEC 20/60 library subroutine to determine the CPU time. It also requires a number of IMSL Version 9.2 subroutines.

The program creates and inverts finite segments of the infinite Hilbert matrix of orders one through to the largest possible on the particular computer. The program terminates when an IMSL error is detected or when order 30 is completed. The errors in approximating the inverses are written on the disk file "HIL.DAT" along with the time required.

Bibliography

[Gregory et al, 1969] Robert T. Gregory, David L. Karney, *A Collection of Matrices for Testing Computational Algorithms,* John Wiley & sons, Inc., 1969.

[IMSL, 1984] *User's Manual, IMSL Library,* Vol. 2, 1984; International Mathematical & Statistical Libraries, Inc., Houston, Texas.

Appendix A Program Listing

```
      PROGRAM HILBRT
C
C     INVERT FINITE SEGMENT OF INFINITE HILBERT MATRIX
C
C     LAST REVISION: DECEMBER 1, 1986
C     AUTHOR:        J. A. THERIAULT
C
      PARAMETER NMAX=30,IRPT = 12000
      REAL B(NMAX,NMAX),A(NMAX,NMAX),D(NMAX,NMAX),NORMI,NORMD
C     CALL TO DEC20 DEPENDENT SUBROUTINE  (ITIM = TIME FROM START IN MSEC)
      CALL CPUTIM(ITIM)
C
      OPEN(UNIT = 22, FILE = 'HIL.DAT')
      WRITE(22,100)
      DO 8 II = 1,IRPT
         B(1,1) = 1.
         DO 8 N = 1,4
            M = N
            CALL CMPINV(A,B,D,M,NMAX,NORMI,NORMD,IER)
            IF (II.EQ.1) WRITE(22,102) N,NORMI,NORMD
            IF (IER.NE.0) GOTO 9
8     CONTINUE
9     IF (IER.NE.0) THEN
         WRITE(22,105) I
         STOP
      ENDIF
C     CALL TO DEC20 DEPENDENT SUBROUTINE (ITIM2 = TOTAL ELAPSED TIME IN MSEC)
      CALL CPUTIM(ITIM2)
C
      N = 4
11    N = N+1
         M = N
         CALL CMPINV(A,B,D,M,NMAX,NORMI,NORMD,IER)
         IF (IER.NE.0) GOTO 200
         WRITE(22,102) N,NORMI, NORMD
      IF (N.LE.NMAX) GOTO 11
```

```
C
200   WRITE(*,104) N,IER
      WRITE(22,104) N,IER
C     CALL TO DEC20 DEPENDENT SUBROUTINE (ITIM1 = TOTAL ELAPSED TIME IN MSEC)
      CALL CPUTIM(ITIM1)
C
      WRITE(22,106) IRPT,(ITIM2-ITIM)/60000.
      WRITE(22,103) (ITIM1-ITIM)/60000.
100   FORMAT(/ ' ORDER   NORM OF A*D-I  NORM OF D-B')
102   FORMAT (I4,6X,G10.5,4X,G10.5)
103   FORMAT(' TOTAL CPU TIME =',F, '  MINUTES')
104   FORMAT (/' MATRIX SIZE ',I4,' UNABLE TO BE INVERTED.',
     /'    IMSL ERROR #', I4,/)
105   FORMAT(' ERROR WITH ORDER LESS THAN 4', I)
106   FORMAT(' TIME AFTER ',I7,' REPETITIONS OF THE FIRST',
     /' FOUR ORDERS =',F)
      STOP
      END
C
      SUBROUTINE CMPINV(A,B,D,N,NMAX,NORMI,NORMD,IER)
      REAL B(NMAX,NMAX),A(NMAX,NMAX),D(NMAX,NMAX),NORMI,NORMD
      REAL WORK(1000)
C
      NORMI = 0.
      NORMD = 0.
      DO 2 I = 1,N-1
         A(I,N) = 1./(I+N-1.)
2        A(N,I) = A(I,N)
      A(N,N) = 1./(2*N-1.)
      IDGT = 0
      CALL LINV2F(A,N,NMAX,D,IDGT,WORK,IER)
      IF (IER.NE.0) GOTO 8
      DO 4 I = 1,N
         DO 4 J = 1,N
            C = 0.
            DO 3 K = 1,N
3              C = C + A(I,K)*D(K,J)
            IF ((I.NE.J).AND.(ABS(C).GT.NORMI)) NORMI = ABS(C)
            IF ((I.EQ.J).AND.(ABS(C-1.).GT.NORMI)) NORMI=ABS(C-1.)
            IF (ABS(D(I,J)-B(I,J)).GT.NORMD) NORMD=ABS(D(I,J)-B(I,J))
4     CONTINUE
C
      DO 5 I = 1,N
         DO 5 J=1,N
```

```
5          B(I,J) = B(I,J)*((N+J)/(N+1.-J))*((N+I)/(N+1.-I))
      DO 6 J = 1,N+1
         TMP = PERM(J-1,1)
         B(N+1,J) = (-1)**(N+J-1)*(PERM(2*N+1,N)/TMP/PERM(N+1-J,1))
     &                     *(PERM(N+J-1,N)/TMP)
6        B(J,N+1) = B(N+1,J)
8     CONTINUE
C
      RETURN
      END
C
      REAL FUNCTION PERM(M,N)
C     COMPUTE M!/N! WHERE M > N
      F = 1
      DO 10 I = N+1,M
10       F = F * I
      PERM = F
      RETURN
      END
```

UNLIMITED DISTRIBUTION

AD-A181948

Unclassified
Security Classification

DOCUMENT CONTROL DATA – R & D

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

1 ORIGINATING ACTIVITY Defence Research Establishment Atlantic	2a. DOCUMENT SECURITY CLASSIFICATION Unclassified 2b GROUP

3. DOCUMENT TITLE

A COMPUTER BENCHMARK PROGRAM FOR HILBERT MATRIX INVERSION USING IMSL ROUTINES

4 DESCRIPTIVE NOTES (Type of report and inclusive dates) Technical Communication

5. AUTHOR(S) (Last name, first name, middle initial)

Theriault, James A.

6 DOCUMENT DATE March 1987	7a. TOTAL NO. OF PAGES 19	7b. NO. OF REFS 2

8a. PROJECT OR GRANT NO.	9a. ORIGINATOR'S DOCUMENT NUMBER(S) DREA Technical Communication 87/305
8b. CONTRACT NO.	9b. OTHER DOCUMENT NO.(S) (Any other numbers that may be assigned this document)

10. DISTRIBUTION STATEMENT

Unlimited Distribution

11. SUPPLEMENTARY NOTES	12. SPONSORING ACTIVITY

13. ABSTRACT

In considering potential replacements for computer facilities, such as that for the DREA DECSYSTEM 20/60, comparison standards must be specified. In the comparison standards, two notable characteristics that should be specified are the speed and accuracy of numerical computations. HILBRT is a benchmarking program which may be used for this purpose. It is based on the inversion of finite segments of the infinite Hilbert matrix. The theory, program, and example results are presented.

DSIS
78-070

Unclassified
Security Classification

KEY WORDS

MATRICES (MATHEMATICS)

COMPUTER PROGRAMS

NUMERICAL ANALYSIS

COMPUTER BENCHMARK

INSTRUCTIONS

1. ORIGINATING ACTIVITY Enter the name and address of the organization issuing the document.

2a. DOCUMENT SECURITY CLASSIFICATION: Enter the overall security classification of the document including special warning terms whenever applicable.

2b. GROUP Enter security reclassification group number. The three groups are defined in Appendix 'M' of the DRB Security Regulations.

3. DOCUMENT TITLE: Enter the complete document title in all capital letters. Titles in all cases should be unclassified. If a sufficiently descriptive title cannot be selected without classification, show title classification with the usual one-capital-letter abbreviation in parentheses immediately following the title.

4. DESCRIPTIVE NOTES: Enter the category of document, e.g. technical report, technical note or technical letter. If appropriate, enter the type of document, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the document. Enter last name, first name, middle initial. If military, show rank. The name of the principal author is an absolute minimum requirement.

6. DOCUMENT DATE: Enter the date (month, year) of Establishment approval for publication of the document.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the document.

8a. PROJECT OR GRANT NUMBER: If appropriate, enter the applicable research and development project or grant number under which the document was written.

8b. CONTRACT NUMBER If appropriate, enter the applicable number under which the document was written.

9a. ORIGINATOR'S DOCUMENT NUMBER(S) Enter the official document number by which the document will be identified and controlled by the originating activity. This number must be unique to this document.

9b. OTHER DOCUMENT NUMBER(S): If the document has been assigned any other document numbers (either by the originator or by the sponsor), also enter this number(s).

10. DISTRIBUTION STATEMENT: Enter any limitations on further dissemination of the document, other than those imposed by security classification, using standard statements such as:

    (1) "Qualified requesters may obtain copies of this document from their defence documentation center."

    (2) "Announcement and dissemination of this document is not authorized without prior approval from originating activity."

11. SUPPLEMENTARY NOTES Use for additional explanatory notes.

12. SPONSORING ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document, even though it may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall end with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (TS), (S), (C), (R), or (U).

    The length of the abstract should be limited to 20 single-spaced standard typewritten lines; 7½ inches long.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a document and could be helpful in cataloging the document. Key words should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context.