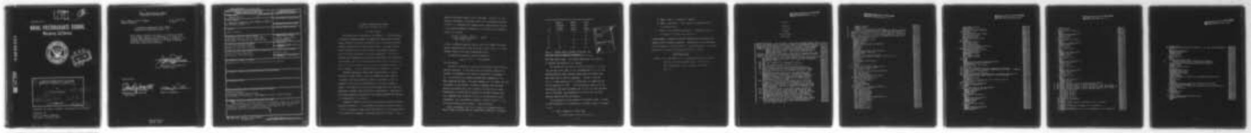
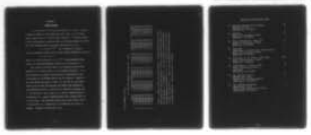
AD-A059 597 NAVAL POSTGRADUATE SCHOOL MONTEREY CALIF F/G 12/1
A COMPUTER SUBROUTINE FOR STRESS ANALYSIS OF ROTATING DISKS. II--ETC(U)
AUG 78 J E BROCK
UNCLASSIFIED NPS-69-78-014 NL

| OF |
AD
A059597





END
DATE
FILMED
-12-78
DDC

NPS-69-78-014

NAVAL POSTGRADUATE SCHOOL

Monterey, California

(6) A COMPUTER SUBROUTINE FOR STRESS ANALYSIS OF ROTATING DISKS - II

by

(10) John E. Brock

(16) RR00001

(14) NPS-69-78-014

(12) 15p.

(11) August 1978

(17) RR0000101

Approved for public release; distribution unlimited.

Prepared for:

Chief of Naval Research
Arlington, Virginia 22217

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral Tyler Dedman
Superintendent

J. R. Borsting
Provost

A COMPUTER SUBROUTINE FOR STRESS ANALYSIS OF ROTATING DISKS - II

This report corrects errors in a previous report on the same subject and presents a listing of a revised and improved digital computer program for finding stress distribution in a thin rotating disk with nonuniform heating.

John E. Brock
John E. Brock
Professor of Mechanical Engineering

Approved by:

Paul J. Marto
Paul J. Marto, Chairman
Mechanical Engineering Department

William M. Tolles
W. M. Tolles
Dean of Research

NPS-69-78-014
August 1978

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS-69-78-014	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A COMPUTER SUBROUTINE FOR STRESS ANALYSIS OF ROTATING DISKS - II.		5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) John E. Brock		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Professor John E. Brock (Code 69Bc) Department of Mechanical Engineering		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61152N, RR000-01-01 N0001478WR80023
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE August 1978 13. NUMBER OF PAGES 12
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Chief of Naval Research Arlington, Virginia 22217		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Stress analysis
Rotating disks, Heated Disks
Axisymmetric Elasticity, Axisymmetric Disks, Elastic Disks

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report corrects errors in a previous report on the same subject and presents a listing of a revised and improved digital computer program for finding stress distribution in a thin rotating disk with nonuniform heating.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

78 10 06 021

A Computer Subroutine for Stress Analysis of Rotating Disks - II.

by John E. Brock

Based upon theory developed by the writer, R. E. Brown developed a successful computer program for analysis of radial and circumferential stresses in rotating axisymmetric disks of variable thickness having an axisymmetrical thermal strain field. The writer revised Brown's program so as to invoke a group of ancillary subroutines which have been found useful in another application. In doing so, however, much unnecessary and confusing normalization was introduced. In particular, one of the normalizations would cause the analysis to fail in the quite common case of a disk with no radial loading at its outer boundary. All this material appears as Reference 1, hereof.

Referees evaluating a paper based upon Reference 1, called attention to these faults so that the program has been rewritten. A listing of the main subroutine, RODISK, as revised, as well as listings of the ancillary subroutines may be found in Appendix A hereof. The reader will note that other changes have also been made resulting in somewhat more flexibility of application. Employment of the revised program is described in the textual material which appears at the beginning of the listing.

Appendix B contains a revision of the second illustrative example problem of Reference 1. This problem was solved for various values of $M = N-1$, the number of equal subdivisions into which the annular radius $b-a$ is divided for purposes of numerical analysis by RODISK. Also, a number of different values of KP(3) were used. If KP(3) > 0, its value is the number of iterations which will be performed by RODISK. If KP(3) < 0, iteration will continue until three successive values of the unknown parameter B determined in the course of the analysis, satisfy the relation

$$\frac{|B_1-B_2| + |B_2-B_3| + |B_3-B_1|}{|B_1| + |B_2| + |B_3|} < 10^{KP(3)}$$

We also determined execution time by use of the library subroutine IXCLOK, executing under CP-cms on the IBM 360/67 at the W. R. Church Computer Center at the Naval Postgraduate School.

We found that execution time per iteration is

$$t_{iter} = 1.2\ M + 5 \quad \text{(milliseconds)}$$

for any problem.

Accuracy was evaluated by dealing with problems having available analytic solutions. It was found that the principal limitation on accuracy is determined by the choice of subdivions, the integer M = N-1, so that there is a certain inherent error regardless of how many iterations are made. This error depends on M, of course, and upon the details of the problem. The error is greatest near the inner radius of an annular disk, and is large if the ratio a/b is small. Fortuitously, the error may be smaller for an early iteration than for a somewhat later iteration but this is not practically useful information. For the problem of Appendix B hereof, with a/b = .165, we find the results given in Table 1, (see next page).

Thus, for example, with M = 20, there is an inherent error of about 1% and the results are not significantly improved by iterating

M	approx. limiting % error	approx. iters. req'd.	total time, secs.
5	16	5	.055
10	5	7	.12
20	1	11	.32
40	.1	17	.90
100	.01	25	3.1

Table 1. Percent error, required iterations, and execution time for problem of Appendix B.

more than eleven times. With eleven iterations, the solution is returned from RODISK in 0.32 seconds.

The significant conclusion is that the execution is so fast that one may as well take M = 100 (corresponding to N = 101, the maximum available under present dimensioning) and iterate many more times than is strictly necessary. Taking N = 101 and KP(3) = -8 gave execution in 3.7 seconds with 31 iterations and with an accuracy of 0.004% (In the problem at hand, $\sigma_r(a)$ was specified as zero and the program gets -1.14E-11 so that the error here is "infinite". Our evaluation of 0.004% is for the first position rather than for the zeroeth.)

This concludes the text proper of the present report. However, we take advantage of this opportunity to correct errors in Reference 1, viz.:

(1) Page 3, equation 12 should read

$$m = \pm\sqrt{(n^2-4\nu n+4)} = \pm\sqrt{[(n-2)^2+4(1-\nu)n]}$$

(2) Page 6, line 2. In place of T read αT.

(3) Page 6, equation 33. Lower limit of integration should be a rather than 0.

(4) Page 7, line following equation 40. Reference should be to equation 37 rather than equation 38.

Acknowledgment is gratefully made for assistance by the Naval Postgraduate School Research Foundation. Appreciation is also expressed to the referees of the ASME Journal of Applied Mechanics for directing attention to the flaws in the earlier version of RODISK.

REFERENCE

1. Brock, J. E., and Brown, R. E., A computer subroutine for stress analysis of rotating, heated disks. NPS-69-78-012, Naval Postgraduate School, Monterey, California, May 1978

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

Appendix A

Listing of
Subroutine
RODISK
and ancillary
subroutines

```
C SUBROUTINE RODISK.  JOHN E. BROCK, 1 MAY 1978, REVISED 1 AUGUST 1978. ROD00010
C THIS IS A SUBROUTINE FOR DETERMINING RADIAL AND CIRCUMFERENTIAL STRESSROD00020
C IN AN AXISYMMETRIC THIN ELASTIC DISK HAVING AN AXISYMMETRIC THERMAL   ROD00030
C STRAIN FIELD AND ROTATING AT ANGULAR VELOCITY OMEGA (RADIANS/SECOND)  ROD00040
C ABOUT THE AXIS OF SYMMETRY.  TWO TYPES OF PROBLEM MAY BE TREATED:     ROD00050
C   TYPE 1: ANNULAR DISK OF INSIDE RADIUS ARAD AND OUTSIDE RADIUS       ROD00060
C           BRAD.  THE RADIAL STRESS IS SRA AT THE INNER RADIUS AND     ROD00070
C           SRB AT THE OUTER RADIUS.  THE INSIDE RADIUS MUST BE         ROD00080
C           GREATER THAN ZERO                                           ROD00090
C   TYPE 2: SOLID DISK HAVING RADIAL STRESS SRB AT OUTSIDE RADIUS BRAD. ROD00100
C THE USER MUST PROVIDE A MAIN PROGRAM WHICH CALLS SUBROUTINE RODISK    ROD00110
C AFTER IT HAS SUPPLIED THE FOLLOWING INFORMATION.                      ROD00120
C   (1) N, INTEGER. (N-1) IS THE NUMBER OF EQUAL SUBDIVISIONS INTO WHICHROD00130
C       THE ANNULAR RADIUS (BRAD MINUS ARAD) IS DIVIDED FOR COMPU-      ROD00140
C       TATIONAL PURPOSES. THE PRESENT DIMENSIONING CAN ACCOMMODATE N   ROD00150
C       NOT GREATER THAN 101.                                           ROD00160
C   (2) BRAD                                                            ROD00170
C   (3) ARAD (NOT NECESSARY FOR PROBLEMS OF TYPE 2.)                    ROD00180
C   (4) SRB                                                             ROD00190
C   (5) SRA (NOT NECESSARY FOR PROBLEMS OF TYPE 2.)                     ROD00200
C   (6) POIS, POISSON'S RATIO                                           ROD00210
C   (7) KP(1)=1,2.  INTEGER TO DENOTE PROBLEM OF TYPE 1,2.              ROD00220
C   (8) KP(2), INTEGER TO PROVIDE FOR SKIPPING WHILE PRINTING           ROD00230
C       OUTPUT.  FOR EXAMPLE, IF N=101 AND KP(2)=5, ONLY EVERY          ROD00240
C       FIFTH SET OF VALUES WILL BE PRINTED: 1ST ,6TH,...., 96TH,       ROD00250
C       AND 101ST.                                                      ROD00260
C   (9) KP(3), INTEGER SPECIFYING NUMBER OF ITERATIONS TO BE            ROD00270
C       PERFORMED.  USUALLY KP(3)=10 IS SUFFICIENT FOR ENGIN-           ROD00280
C       EERING ACCURACY.  ALTERNATELY, IF KP(3) IS A NEGATIVE           ROD00290
C       INTEGER, ITERATION WILL CONTINUE UNTIL THREE SUCCESSIVE         ROD00300
C       VALUES OF A PARAMETER E, DETERMINED INTERNALLY, ARE             ROD00310
C       SUFFICIENTLY CLOSE AS COMPARED TO AN EPSILON EQUAL TO           ROD00320
C       TEN RAISED TO THE KP(3) POWER.                                  ROD00330
C  (10) KP(4). IF KP(4)=0, ONLY FINAL ANSWERS WILL BE PRINTED.          ROD00340
C       IF KP(4)=1, A SEQUENCE OF ITERANT VALUES OF E WILL BE           ROD00350
C       PRINTED TO INDICATE DEGREE OF CONVERGENCE.  IF KP(4)>1,         ROD00360
C       THERE WILL BE NO PRINTING AT ALL WITHIN RODISK, BUT UPON        ROD00370
C       RETURN KP(5) WILL CONTAIN THE NUMBER OF ITERATIONS WHICH        ROD00380
C       WERE PERFORMED SO THAT KP(5) MUST BE RESET BEFORE RODISK        ROD00390
C       IS CALLED AGAIN.                                                ROD00400
C  (11) KP(5).  KP(5)=0 CAUSES MILNE CUBIC SPLINE INTEGRATION           ROD00410
C       TO BE USED. OTHERWISE TRAPEZOIDAL INTEGRATION IS USED.          ROD00420
C  (12) VECTOR X(1,J), J=1,2,...,N, CONTAINING VALUES OF DISK           ROD00430
C       THICKNESS AT EQUALLY SPACED RADII FROM INSIDE TO OUTSIDE.       ROD00440
C  (13) VECTOR X(2,J) CONTAINS VALUES OF GAMMA TIMES OMEGA              ROD00450
C       SQUARED WHERE GAMMA IS (MASS) DENSITY OF THE MATERIAL.          ROD00460
C       FOR MOST PROBLEMS GAMMA DOES NOT VARY WITH RADIUS AND           ROD00470
C       ALL ELEMENTS OF THE VECTOR WILL BE THE SAME.                    ROD00480
C  (14) VECTOR X(3,J) CONTAINS VALUES OF (EE)(ALPHA)(TEE) WHERE         ROD00490
C       EE IS YOUNG'S MODULUS, ALPHA IS THE COEFFICIENT OF LINEAR       ROD00500
C       THERMAL EXPANSION, AND TEE IS THE TEMPERATURE CHANGE.           ROD00510
C             THE MAIN PROGRAM MUST CONTAIN THE STATEMENTS:             ROD00520
C       IMPLICIT REAL*8 (A-H,O-Z)                                       ROD00530
```

```
C           REAL*8 X(20,101)
C           INTEGER KP(5)
C           COMMON X,N,KP
C           COMMON /ONE/ARAD,BRAD,SRA,SRB,POIS
C             FOLLOWING SUBROUTINE RODISK THERE ARE SEVERAL ANCILLARY
C     SUBROUTINES WHICH PERFORM VARIOUS OPERATIONS ON THE VECTORS
C     X(I,J).  THE PURPOSE OF EACH IS OBVIOUS FROM THE LISTING.
C     THEY MAY BE EMPLOYED IN THE USER'S MAIN PROGRAM.  SUBROUTINE
C     DUPV, WHICH DUPLICATES A VECTOR, AND SUBROUTINE PRIV, WHICH
C     PRINTS A VECTOR, ARE NOT CALLED BY RODISK BUT MAY BE USEFULL
C     IN THE USER'S MAIN PROGRAM.
      SUBROUTINE RODISK
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 X(20,101)
      INTEGER KP(5)
      COMMON X,N,KP
      COMMON /ONE/ARAD,BRAD,SRA,SRB,POIS
      ONE=1.D+0
      ZERO=0.D+0
      B1=1.D+8
      B2=1.D+10
      B3=1.D+12
      IF(KP(3).LT.0) EPS=(1.D+1)**(KP(3))
      IF(KP(1).EQ.2) ARAD=ZERO
      BMA=BRAD-ARAD
      ENM=N-1
      IF(KP(4).LE.1) WRITE(6,2) KP(1)
    2 FORMAT(//,10X,'RODISK PROBLEM OF TYPE ',I1,'.')
      DO 5 I=1,N
      EIM=I-1
      Y=EIM/ENM
      X(4,I)=ARAD+(BRAD-ARAD)*Y
      X(5,I)=Y
    5 X(6,I)=Y
      ITER=1
      IF(KP(1).EQ.2) GO TO 100
C THE PROBLEM IS OF TYPE 1: ANNULAR DISK
      C1=(2.D+0+POIS)*(BRAD-ARAD)
      CALL INTV(1,7,BMA)
      C2=X(7,N)
      C5=X(3,1)-X(3,N)+(ONE-POIS)*(SRA-SRB)
      CALL MULV(1,2,8)
      CALL MULV(4,8,9)
      CALL INTV(9,10,BMA)
      C6=X(10,N)+X(1,N)*SRB-X(1,1)*SRA
   20 CALL INTV(6,11,BMA)
      C3=BRAD+(ONE+POIS)*X(11,N)
      CALL MULV(1,6,12)
      CALL INTV(12,13,BMA)
      C4=X(13,N)
      D=C1*C4-C2*C3
      A=(C5*C4-C6*C3)/D
      B=(C1*C6-C2*C5)/D
   30 CONTINUE
      IF(KP(4).EQ.1) WRITE(6,7) ITER,A,B
    7 FORMAT(5X,I10,1P2E20.5)
      CALL MULS(7,14,A)
      CALL MULS(13,15,B)
      CALL ADDV(14,15,15)
      CALL SUBV(15,10,16)
      S=SRB*X(1,N)-X(16,N)
      CALL ADDS(16,16,S)
      CALL DIVV(16,1,16)
      ZA=X(3,1)+A*ARAD+(ONE-POIS)*X(16,1)
      CALL MULS(11,17,B)
      CALL SUBS(4,18,ARAD)
      CALL MULS(18,18,A)
      CALL ADDV(17,18,17)
      S=-(ONE+POIS)
      CALL MULS(17,17,S)
      CALL ADDS(17,17,ZA)
      S=ONE-POIS
      CALL MULS(16,18,S)
      CALL SUBV(17,3,17)
```

```
      CALL SUBV(17,18,17)                                               RODO1280
      ITER=ITER+1                                                       RODO1290
      IF(KP(3).LT.0) GO TO 215                                          RODO1300
      IF(ITER.GT.KP(3)) GO TO 200                                       RODO1310
   40 CONTINUE                                                          RODO1320
      CALL DUPV(4,14)                                                   RODO1330
      IF(KP(1).EQ.2) X(14,1)=ONE                                        RODO1340
      CALL DIVV(17,14,18)                                               RODO1350
      CALL SUBS(18,18,A)                                                RODO1360
      CALL DIVS(18,6,B)                                                 RODO1370
      IF(KP(1).EQ.2) GO TO 150                                          RODO1380
      GO TO 20                                                          RODO1390
  100 A=ZERO                                                            RODO1400
      C=(ONE-POIS)/X(1,1)                                               RODO1410
      CALL MULV(1,2,7)                                                  RODO1420
      CALL MULV(4,7,8)                                                  RODO1430
      CALL INTV(8,10,BMA)                                               RODO1440
      SUM=X(10,N)+SRB*(X(1,N)-X(1,1))                                   RODO1450
      SUM=C*SUM+X(3,1)-X(3,N)                                           RODO1460
  150 CALL INTV(6,11,BMA)                                               RODO1470
      DEN=BRAC+(ONE+POIS)*X(11,N)                                       RODO1480
      CALL MULV(1,6,11)                                                 RODO1490
      CALL INTV(11,13,BMA)                                              RODO1500
      DEN=DEN+C*X(13,N)                                                 RODO1510
      B=SUM/DEN                                                         RODO1520
      CALL INTV(1,7,BMA)                                                RODO1530
      CALL INTV(6,11,BMA)                                               RODO1540
      GO TO 30                                                          RODO1550
  215 B3=B2                                                             RODO1560
      B2=B1                                                             RODO1570
      B1=B                                                              RODO1580
      ZUM=DABS(B1-B2)+DABS(B2-B3)+DABS(B3-B1)                           RODO1590
      DIV=DABS(B1)+DABS(B2)+DABS(B3)                                    RODO1600
      CRIT=ZUM/DIV                                                      RODO1610
      IF(CRIT.LT.EPS) GO TO 200                                         RODO1620
      GO TO 40                                                          RODO1630
  200 CALL ADDV(17,16,19)                                               RODO1640
      IF(KP(3).LT.0.AND.KP(4).EQ.0) WRITE(6,201) ITER,EPS               RODO1650
  201 FORMAT(//,20X,I8,' ITERATIONS REQUIRED WITH EPSILON = ',1PE8.1)   RODO1660
      IF(KP(4).LE.1) WRITE(6,204)                                       RODO1670
  204 FORMAT(//)                                                        RODO1680
      IF(KP(4).LE.1) WRITE(6,205)                                       RODO1690
  205 FORMAT(23X,'RADIUS',10X,'THICKNESS',5X,'GAMMA OMEGA SQ',7X,       RODO1700
     1'EE ALPHA TEE',7X,'SIGMA RADIAL',6X,'SIGMA CIRCUMF')              RODO1710
      NSKIP=KP(2)                                                       RODO1720
      DO 210 I=1,N,NSKIP                                                RODO1730
      J=I/NSKIP                                                         RODO1740
      IF(KP(4).LE.1) WRITE(6,211)J,X(4,I),X(1,I),X(2,I),                RODO1750
     1X(3,I),X(16,I),X(19,I)                                            RODO1760
  210 CONTINUE                                                          RODO1770
      IF(KP(4).GT.1) KP(5)=ITER                                         RODO1780
  211 FORMAT(I10,1P6E19.5)                                              RODO1790
      RETURN                                                            RODO1800
      END                                                               RODO1810
C   THIS IS THE START OF THE ANCILLARIES                                RODO1820
      SUBROUTINE ADDV(N1,N2,N3)                                         RODO1830
      REAL*8 X(20,101),S                                                RODO1840
      INTEGER KP(5)                                                     RODO1850
      COMMON X,N,KP                                                     RODO1860
      DO 1 I=1,N                                                        RODO1870
    1 X(N3,I)=X(N1,I)+X(N2,I)                                           RODO1880
      RETURN                                                            RODO1890
      END                                                               RODO1900
      SUBROUTINE SUBV(N1,N2,N3)                                         RODO1910
      REAL*8 X(20,101),S                                                RODO1920
      INTEGER KP(5)                                                     RODO1930
      COMMON X,N,KP                                                     RODO1940
      DO 1 I=1,N                                                        RODO1950
    1 X(N3,I)=X(N1,I)-X(N2,I)                                           RODO1960
      RETURN                                                            RODO1970
      END                                                               RODO1980
      SUBROUTINE MULV(N1,N2,N3)                                         RODO1990
      REAL*8 X(20,101),S                                                RODO2000
      INTEGER KP(5)                                                     RODO2010
```

```
      COMMON X,N,KP
      DO 1 I=1,N
    1 X(N3,I)=X(N1,I)*X(N2,I)
      RETURN
      END
      SUBROUTINE DIVV(N1,N2,N3)
      REAL*8 X(20,101),S
      INTEGER KP(5)
      COMMON X,N,KP
      DO 1 I=1,N
    1 X(N3,I)=X(N1,I)/X(N2,I)
      RETURN
      END
      SUBROUTINE ADDS(N1,N2,S)
      REAL*8 X(20,101),S
      INTEGER KP(5)
      COMMON X,N,KP
      DO 1 I=1,N
    1 X(N2,I)=X(N1,I)+S
      RETURN
      END
      SUBROUTINE SUBS(N1,N2,S)
      REAL*8 X(20,101),S
      INTEGER KP(5)
      COMMON X,N,KP
      DO 1 I=1,N
    1 X(N2,I)=X(N1,I)-S
      RETURN
      END
      SUBROUTINE MULS(N1,N2,S)
      REAL*8 X(20,101),S
      INTEGER KP(5)
      COMMON X,N,KP
      DO 1 I=1,N
    1 X(N2,I)=X(N1,I)*S
      RETURN
      END
      SUBROUTINE DIVS(N1,N2,S)
      REAL*8 X(20,101),S
      INTEGER KP(5)
      COMMON X,N,KP
      DO 1 I=1,N
    1 X(N2,I)=X(N1,I)/S
      RETURN
      END
      SUBROUTINE PRIV(N1,I,J)
      REAL*8 X(20,101),S
      INTEGER KP(5)
      COMMON X,N,KP
C   IF SECOND ARGUMENT EQUALS 0. GO DIRECTLY TO RETURN
C   IF SECOND ARGUMENT EQUALS 1.PRINT THE VECTOR.
C   IF SECOND ARGUMENT EQUALS 2. PRINT THE IDENTITY AND THE VECTOR.
C   IF SECOND ARGUMENT EQUALS 3. PRINT THE VECTOR NUMBER AND THE VECTOR.
C   IF SECOND ARGUMENT EQUALS 4. PRINT NUMBER, IDENTITY, AND VECTOR.
C   IF SECOND ARGUMENT EQUALS 5. PRINT IDENTITY ONLY.
      IF(I.EQ.0) GO TO 10
      IF(I.EQ.1) GO TO 1
      IF(I.EQ.2) GO TO 2
      IF(I.EQ.3) GO TO 3
      IF(I.EQ.4) GO TO 4
      IF(I.EQ.5) GO TO 5
    1 DO 8 K=1,N
    8 WRITE(6,9) K,X(N1,K)
    9 FORMAT(30X,I5,1PE20.5)
   10 RETURN
    2 WRITE(6,21) J
   21 FORMAT(//,30X,'VECTOR WITH IDENTITY ',I5,' FOLLOWS:')
      GO TO 1
    3 WRITE(6,31) N1
   31 FORMAT(//,30X,'VECTOR NUMBER ',I5,' FOLLOWS:')
      GO TO 1
    4 WRITE(6,41) N1,J
   41 FORMAT(//,30X,'VECTOR X',I2,' WITH IDENTITY ',I5,' FOLLOWS:')
      GO TO 1
```

```
    5 WRITE(6,51) J
   51 FORMAT(//,30X,'VECTOR WITH IDENTITY ',I5,' HAS BEEN GENERATED.')
      GO TO 10
      END
      SUBROUTINE DUPV(N1,N2)
      REAL*8 X(20,101),S
      INTEGER KP(5)
      COMMON X,N,KP
      DO 1 I=1,N
    1 X(N2,I)=X(N1,I)
      RETURN
      END
      SUBROUTINE INTV(N1,N2,S)
C         KP(5)=0 CAUSES MILNE INTEGRATION TO BE USED.
C         OTHERWISE TRAPEZOIDAL INTEGRATION IS USED.
      REAL*8 X(20,101),S,ADD,NINO,NTNO,FIVO,THTO,EN,R,P
      INTEGER KP(5)
      COMMON X,N,KP
      IF(KP(5).NE.0) GO TO 10
      EN=N-1
      EN=1.D+0/EN
      NINO=EN*9.D+0/2.4D+1
      NTNO=EN*1.9D+1/2.4D+1
      FIVO=EN*5.D+0/2.4D+1
      THTO=EN*1.3D+1/2.4D+1
      R=EN/2.4D+1
      X(N2,1)=0.D+0
      X(N2,2)=NINO*X(N1,1)+NTNO*X(N1,2)-FIVO*X(N1,3)+X(N1,4)*R
      NM3=N-3
      DO 1 K=1,NM3
      KP1=K+1
      KP2=K+2
      KP3=K+3
      ADD=THTO*(X(N1,KP1)+X(N1,KP2))-R*(X(N1,K)+X(N1,KP3))
    1 X(N2,KP2)=X(N2,KP1)+ADD
      X(N2,N)=X(N2,N-1)+NINO*X(N1,N)+NTNO*X(N1,N-1)-FIVO*X(N1,N-2)
     1+X(N1,N-3)*R
      CALL MULS(N2,N2,S)
      RETURN
   10 CONTINUE
      X(N2,1)=0.D+0
      P=2*(N-1)
      DO 11 I=2,N
      J=I-1
   11 X(N2,I)=X(N2,J)+X(N1,I)/P+X(N1,J)/P
      CALL MULS(N2,N2,S)
      RETURN
      END
```

Appendix B

Sample Problem

A disk rotating at 7200 rpm and composed of a metal having a a specific weight of 0.283 pounds per cubic inch, is 0.85 inches inside diameter and 5.15 inches outside diameter. The radial stress at the inside radius is zero and that at the outside radius is 22,000 psi. The thickness varies with radius according to the law

$$t = 0.1493\ r^{-0.42} \qquad \text{(all dimensions in inches)}$$

and the temperature change (from the zero stress condition) is given by

$$T = 60 - 1.6\ r^2.$$

Take $E = 29{,}000{,}000$ psi and $\alpha = 6.7\ 10^{-6}$ /°F and determine radial stress (σ_r) and circumferential stress (σ_θ) as functions of r.

This problem illustrates most of the capabilities of RODISK. Because of the special nature of the thickness variation, i.e, a power relation, an analytic solution may be established so that the accuracy of the RODISK solution may be evaluated. Results of such evaluations are given in Table 1 of the body of this report. There it may be seen that accuracy far better than engineering considerations require or justify may be obtained by taking, say, N = 101 and KP(3) = 25, so that in 3.1 seconds RODISK returns to the calling (i.e., input) program results with a maximum error of 0.01 % or less. The tabulation which follows shows output with N = 101 and KP(3) = -6, resulting in 27 iterations and taking 3.3 seconds. Accuracy is better than .006%.

```
RODISK PROBLEM OF TYPE 1

           27 ITERATIONS REQUIRED WITH EPSILON = 1.0D-6

        RADIUS       THICKNESS GAMMA OMEGA SQ   EE ALPHA TEE   SIGMA RADIAL  SIGMA CIRCUMF
 0    8.50000D-01    1.59847D 00    4.16880D 02   1.14334D 04  -2.27592D-12    3.26495D 04
 1    1.28000D 00    1.34596D 00    4.16880D 02   1.11487D 04   9.84878D 03    2.40489D 04
 2    1.71000D 00    1.19179D 00    4.16880D 02   1.07490D 04   1.41231D 04    2.17416D 04
 3    2.14000D 00    1.08464D 00    4.16880D 02   1.02343D 04   1.65779D 04    2.12388D 04
 4    2.57000D 00    1.00436D 00    4.16880D 02   9.60467D 03   1.82055D 04    2.14332D 04
 5    3.00000D 00    9.41172D-01    4.16880D 02   8.86008D 03   1.93710D 04    2.19515D 04
 6    3.43000D 00    8.89685D-01    4.16880D 02   8.00053D 03   2.02394D 04    2.26427D 04
 7    3.86000D 00    8.46629D-01    4.16880D 02   7.02601D 03   2.08959D 04    2.34382D 04
 8    4.29000D 00    8.09893D-01    4.16990D 02   5.93653D 03   2.13897D 04    2.43040D 04
 9    4.72000D 00    7.78044D-01    4.16880D 02   4.73209D 03   2.17511D 04    2.52226D 04
10    5.15000D 00    7.50068D-01    4.16880D 02   3.41269D 03   2.20000D 04    2.61848D 04
```

Figure 1. Typical output (for sample problem). The main program supplied information about inner and outer radii and the rdial stressese thereat, angular velocity and density, and "EE ALPHA TEE." These data reappear in the output above. The main program also supplied N = 101, ν = 0.3, KP(1) = 1, KP(2) = 10, KP(3) = -6, KP(4) = 0, and KP(5) = 0. Then it called subroutine RODISK which produced the output shown here. Execution time was 3.3 seconds.

INITIAL DISTRIBUTION LIST

1. Defense Documentation Center 2
Cameron Station
Alexandria, VA 22314

2. Library 2
Naval Postgraduate School
Monterey, CA 93940

3. Dean of Research, Code 012 1
Naval Postgraduate School
Monterey, CA 93940

4. Chairman 1
Department of Mechanical Engineering
Naval Postgraduate School
Monterey, CA 93940

5. Professor J. E. Brock, Code 69Bc 15
Naval Postgraduate School
Monterey, CA 93940

6. LCDR Robert E. Brown, USN 2
Naval Amphibious School
Coronado
San Diego, CA 92155

7. Mr. Charles Miller 1
NAVSEA Code 0331
Naval Ship Systems Command
Washington, DC 20362

8. Mr. Richard Carleton 1
NAVSEC Code 6146
Naval Ship Engineering Center
Washington, DC 20362