NASA Technical Memorandum 107741

SCBUCKLE Users Manual

Buckling Analysis Program for Simply Supported and Clamped Panels

Juan R. Cruz

March 1993

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-0001

Introduction

The program SCBUCKLE calculates the buckling loads and mode shapes of cylindrically curved, rectangular panels (see fig. 1). The panel is assumed to have no imperfections. SCBUCKLE is capable of analyzing specially orthotropic symmetric panels (i.e., $A_{16} = A_{26} = 0.0$, $D_{16} = D_{26} = 0.0$, $B_{ij} = 0.0$). The analysis includes first-order transverse shear theory, and is thus capable of modeling sandwich panels. The analysis supports two types of boundary conditions: either simply supported or clamped on all four edges. The panel can be subjected to linearly varying normal loads N_x and N_y in addition to a constant shear load N_{xy}. The applied loads can be divided into two parts: a preload component, and a variable (eigenvalue-dependent) component. The analysis is based on the modified Donnell's equations for shallow shells (see ref. 1). The governing equations are solved by Galerkin's method. More details on the present analysis, including the limitations of shallow shell theory and important assumptions about the inplane boundary conditions, are provided in reference 2.

Symbols

a	panel length (see fig. 1)
$a_{m\,n}$	amplitude coefficients of normal displacement series (see eqs. 8 and 9)
A_{11}, A_{12}, A_{22}, A_{66}	orthotropic inplane stiffnesses of panel
A_{16}, A_{26}	anisotropic inplane stiffnesses of panel
b	panel width (see fig. 1)
B_{ij}	bending-stretching coupling matrix of panel (i, j = 1, 2, 6)
c_x, c_y, c_{xy}	constant N_x, N_y, and N_{xy} components, respectively

c_x^0, c_y^0, c_{xy}^0	constant N_x, N_y, and N_{xy} preload components, respectively
D_{Qx}, D_{Qy}	transverse shear rigidities in the x- and y-directions, respectively (see eqs. 1-4)
D_{11}, D_{12}, D_{22}, D_{66}	orthotropic bending stiffnesses of panel
D_{16}, D_{26}	anisotropic bending stiffnesses of panel
G_{xz}, G_{yz}	transverse shear stiffnesses in the x- and y-directions, respectively
k_x, k_y	linear N_x and N_y components, respectively
k_x^0, k_y^0	linear N_x and N_y preload components, respectively
m, n	indices in assumed series solutions (see eqs. 8 and 9)
m_0, n_0	number of terms in the x- and y-series, respectively
N_x, N_y, N_{xy}	longitudinal, transverse, and shearing stress resultants, respectively (see fig. 2)
r	panel radius (see fig. 1)
t_c	core thickness (see fig. 1)
t_f	panel or facesheet thickness (see fig. 1)
x, y	coordinate system (see fig. 1)
w	buckling mode normal displacement (see eqs. 8 and 9)
λ	buckling eigenvalue (see eqs. 5, 6, and 7)

Note: Symbols in Courier font represent program variables. These variables are often a direct representation of physical or mathematical quantities (e.g., A11 represents A_{11}, Gxz represents G_{xz}, m0 represents m_0, etc.).

Program Description

The program is written in FORTRAN. The program consists of a main procedure and several subroutines, and was developed on a CONVEX C240 computer under the UNIX operating system. The subroutine SYMGEP, which solves the symmetric eigenvalue problem, was obtained from the NASA Langley Research Center Computer Applications Branch. SYMGEP has its own documentation (see reference 3). Appendix A of this document includes the appropriate pages of reference 3.

The program is capable of taking advantage of vectorizing compilers. By using the *-O2* option (i.e., local scalar optimization, global scalar optimization, and vectorization) in the CONVEX FORTRAN *fc* compiler, an almost tenfold reduction in CPU execution time was achieved.

Input

The program reads its input from the file *inpt.dat*. There are 31 lines in the file *inpt.dat*, with one entry per line. The program reads numerical data in free format. Dimensional variables may use any set of consistent units. However, all dimensional variables must use the same set of units. Appendix B contains an example set of input and output.

A typical input file is:

```
Test Case 42                        ! title (32 characters maximum)
10.0            ! a     - panel length [Length]
30.0            ! b     - panel width [Length]
60.0            ! r     - panel radius [Length]
0.32            ! tf    - panel or facesheet thickness [Length]
0.0             ! tc    - core thickness [Length]
```

```
0.245888E+07    ! A11   - orthotropic inplane stiffness [Force/Length]
0.747557E+06    ! A12   - orthotropic inplane stiffness [Force/Length]
0.245888E+07    ! A22   - orthotropic inplane stiffness [Force/Length]
0.855662E+06    ! A66   - orthotropic inplane stiffness [Force/Length]
0.209825E+05    ! D11   - orthotropic bending stiffness [Force-Length]
0.637916E+04    ! D12   - orthotropic bending stiffness [Force-Length]
0.209825E+05    ! D22   - orthotropic bending stiffness [Force-Length]
0.730165E+04    ! D66   - orthotropic bending stiffness [Force-Length]
0.5E+06         ! Gxz   - transverse shear stiffness [Force/Length^2]
0.5E+06         ! Gyz   - transverse shear stiffness [Force/Length^2]
0.0             ! kx0   - linear Nx preload component [Force/Length^2]
0.0             ! cx0   - constant Nx preload component [Force/Length]
0.0             ! ky0   - linear Ny preload component [Force/Length^2]
0.0             ! cy0   - constant Ny preload component [Force/Length]
0.0             ! cxy0  - constant Nxy preload component [Force/Length]
6.67E-02        ! kx    - linear Nx component [Force/Length^2]
-1.0            ! cx    - constant Nx component [Force/Length]
0.0             ! ky    - linear Ny component [Force/Length^2]
0.0             ! cy    - constant Ny component [Force/Length]
1.0             ! cxy   - constant Nxy component [Force/Length]
10              ! m0    - number of terms in x series (integer)
10              ! n0    - number of terms in y series (integer)
1               ! isc   - simply supported/clamped flag (integer)
1               ! iv    - eigensolver flag (integer)
1               ! nmodp - # of eigenvectors written to output file (integer)
```

Comments to the right of the exclamation mark are not necessary for input; they are merely used for the purpose of documentation. The program will ignore the exclamation marks and comments when reading the input file. The quantities in square brackets indicate the dimensions of the variable. In the above example, forces are in units of pounds and lengths in units of inches. A brief explanation of the input variables follows:

`title` - Title line for the input file. Any string of up to 32 characters in length is admissible. This input line must be present even if it is blank.

`a, b` - Panel length and width, respectively. See figure 1.

`r` - Panel radius. See figure 1. For a flat plate analysis `r` can be set to some very large number.

`tf` - Panel or facesheet thickness. See figure 1. If the panel is of monolithic construction, `tf` is the total panel thickness. If the panel is of sandwich construction, `tf` is the thickness of a single facesheet; both facesheets are assumed to have the same thickness. Note that `tf` has

two interpretations. The variable `tc` (see next entry) determines the interpretation of `tf`.

`tc` - Core thickness. See figure 1. If the panel is of monolithic construction, set `tc = 0.0`. Then `tf` is interpreted as the total panel thickness. If `tc` $\neq$ `0.0`, the panel is assumed to be of sandwich construction and `tf` is interpreted as the facesheet thickness (see previous entry).

Note: The importance of `tf` and `tc` comes about in the calculation of the transverse shear rigidities D_{Qx} and D_{Qy}. If the panel is of monolithic construction, D_{Qx} and D_{Qy} are calculated by the formulas

$$D_{Qx} = \frac{5}{6} G_{xz} t_f \tag{1}$$

and

$$D_{Qy} = \frac{5}{6} G_{yz} t_f \tag{2}$$

If the panel is of sandwich construction, D_{Qx} and D_{Qy} are calculated by the formulas

$$D_{Qx} = G_{xz} \frac{(t_f + t_c)^2}{t_c} \tag{3}$$

and

$$D_{Qy} = G_{yz} \frac{(t_f + t_c)^2}{t_c} \tag{4}$$

`A11, A12, A22, A66` - Orthotropic inplane stiffnesses of panel. The panel is assumed to be specially orthotropic and thus, $A_{16} = A_{26} = 0.0$.

`D11, D12, D22, D66` - Orthotropic bending stiffnesses of panel. The panel is assumed to be specially orthotropic and thus, $D_{16} = D_{26} = 0.0$.

Gxz, Gyz - Transverse shear stiffnesses in the x- and y-directions, respectively. If the panel is of monolithic construction, these quantities are the transverse shear stiffnesses of the panel. If the panel is of sandwich construction, these quantities are the transverse shear stiffnesses of the core material. If no transverse shear deformation is desired in the analysis, Gxz and Gyz should be set equal to some large number. In this case setting Gxz and Gyz to be two or three orders of magnitude greater than their actual values is usually sufficient.

kx0, cx0, ky0, cy0, cxy0, and kx, cx, ky, cy, cxy - Preload and load components. The stress resultants N_x, N_y, and N_{xy} define the loading on the panel. The longitudinal stress resultant N_x is assumed to vary linearly with y and is given by:

$$N_x = k_x^0 y + c_x^0 + \lambda(k_x y + c_x) \tag{5}$$

The transverse stress resultant N_y is assumed to vary linearly with x and is given by:

$$N_y = k_y^0 x + c_y^0 + \lambda(k_y x + c_y) \tag{6}$$

The shear stress resultant N_{xy} is assumed to be constant with respect to x and y and is given by:

$$N_{xy} = c_{xy}^0 + \lambda c_{xy} \tag{7}$$

The stress resultants N_x and N_y are positive in tension, while N_{xy} is positive in its usual sense (see fig. 2). The program calculates the parameter λ, which is the eigenvalue. Once λ is known, the buckling stress resultants can be calculated from equations 5, 6, and 7.

The variables kx0, cx0, ky0, cy0, cxy0, kx, cx, ky, cy, and cxy allow the user to define the loading on the panel. These variables are intended to be defined such that the loads on the panel have the desired

sign assuming a positive value of the eigenvalue, λ. Variables with a zero (e.g., kx0, cx0, etc.) are the preload components. As can be seen from equations 5, 6, and 7, the loads defined by these preload components are not affected by the eigenvalue. The loads defined by the load components, kx, cx, ky, cy, and cxy, are affected by the eigenvalue. Although all of the preload components may be set equal to zero, at least one of the load components must be nonzero.

The usage of these load components is illustrated by the following two examples. The first example is a uniform compressive N_x load. The user sets cx = -1.0, and all other load and preload components equal to zero. The program will return a positive value of λ as its lowest eigenvalue. The compressive buckling load, N_x, can then be calculated from equation 5. This buckling N_x will be negative. The second example is a N_x load with a fixed gradient with respect to y but an unknown constant component. The user sets kx0 equal to some nonzero value, cx = -1.0, and all other load and preload components equal to zero. Again, once the program returns a value for λ, the buckling load, N_x, can be calculated from equation 5.

The variables kx and ky allow the user to define linearly varying portions of the loads. However, the gradients of N_x and N_y with respect to y and x respectively may not be defined *a priori* if kx and/or ky are not equal to zero. These gradients also depend on λ, which is a result of the program calculations, and thus not known in advance. If the gradients of N_x and N_y are to be specified *a priori*, it should be done through the variables kx0 and ky0.

Care should be exercised in choosing values for the preload components. If these components are sufficient by themselves to buckle the panel, the program will be unable to return a value for λ. See the Output section for more information.

m0, n0 - Number of terms in the assumed series in the x- and y-directions, respectively. Although m0 and n0 are two independent entries, the current implementation of the program requires them to

have the same value. Setting `m0 = n0 = 10` will usually give acceptable results. The current implementation of the program requires `m0` and `n0` to have a value that is no greater than 20 due to the sizing of the arrays. These arrays can be resized as discussed in Appendix C.

`isc` - Flag to specify boundary conditions. For simply supported boundary conditions set `isc = 1`. For clamped boundary conditions set `isc = 2`.

Note: Due to the limitations of shallow shell analysis, the present clamped boundary solution is accurate only for very small values of b/r. As a result, this version of SCBUCKLE should not be used for curved panels with clamped boundary conditions.

`iv` - Flag to specify whether or not to calculate the eigenvectors. If `iv = 0`, no eigenvectors are calculated. If `iv = 1`, all eigenvectors are calculated.

`nmodp` - Number of eigenvectors to be written into the output file, starting with the eigenvector of the first eigenvalue. The program reads this line, but ignores it if `iv = 0`. Care should be used in selecting `nmodp` since large values result in voluminous output. The number of output lines generated by the program is greater than the product `nmodp*m0*n0`.

Output

The program writes its output to the file *outp.dat*. The program first echoes the input onto the output file to provide a mechanism for error checking. Next, all eigenvalues are written to the output file. There are `m0*n0` eigenvalues. All positive eigenvalues are printed in ascending order followed by the negative eigenvalues also in ascending order in their signed sense (i.e., not in their absolute value sense). If the loading is specified as described previously, the user will usually be interested in the first eigenvalue. If the program detects that any eigenvalue is less than the first one in the absolute value sense, a

warning message (including the mode and value of this lower eigenvalue) will appear before the eigenvalue listing. Finally, if `iv = 1` and `nmodp > 0`, the eigenvectors of the first `nmodp` eigenvalues are written to the output file. These eigenvectors are the coefficients, a_{mn}, of the normal displacement series. For simply supported boundary conditions this series is:

$$w = \sum_{m=1}^{m_0} \sum_{n=1}^{n_0} a_{mn} \sin\left(\frac{m \pi x}{a}\right) \sin\left(\frac{n \pi y}{b}\right) \tag{8}$$

For clamped boundary conditions this series is:

$$w = \sum_{m=1}^{m_0} \sum_{n=1}^{n_0} a_{mn}\left[\cos\left(\frac{(m - 1)\pi x}{a}\right) - \cos\left(\frac{(m + 1)\pi x}{a}\right)\right] \cdot\left[\cos\left(\frac{(n - 1)\pi y}{b}\right) - \cos\left(\frac{(n + 1)\pi y}{b}\right)\right] \tag{9}$$

If the preload components are sufficient by themselves to buckle the panel, a warning message will be printed stating that one of the matrices is not positive definite. In this case no eigenvalues or eigenvectors are printed and execution is terminated.

Program Execution

The program is intended to be executed in the batch or background modes. Care should be taken in selecting `m0` and `n0`; making these quantities greater than 10 to 12 will greatly increase the execution time of the program. Because of the added complexity in the calculations, the solution for clamped boundary conditions will take from five to eight times longer than an equivalent solution for simply supported boundary conditions.

Appendix A: SYMGEP Documentation

MATHEMATICAL AND STATISTICAL SOFTWARE AT LANGLEY--PART I

DATE 03/01/87 REPLACES COPY DATED 10/01/77 SECTION F2.7

TITLE Subroutine SYMGEP PAGE 1 OF 7

LANGUAGE: FORTRAN

PURPOSE: To compute all the eigenvalues and eigenvectors of the symmetric generalized eigenvalue problem $Ax = \lambda Bx$ where A is symmetric and B is symmetric, positive definite by the Cholesky-Wilkinson algorithm.

USE: CALL SYMGEP(MAX,N,A,B,E,IV,WK,IERR)

MAX An input integer specifying the first dimension of arrays A and B as stated in the dimension statement of the calling program.

N An input integer specifying the order of A and B where $1 \leq N \leq MAX$.

A An input/output two-dimensional real array.

Input A contains the symmetric input A matrix (only the full upper triangle of A need be supplied).

Output A contains the eigenvectors normalized to unit length if IV = 1. If IV = 0, A is destroyed. The A array is dimensioned with variable dimensions in the subroutine. Therefore, A must be dimensioned in the calling program with first dimension MAX and second dimension at least N.

B An input two-dimensional real array containing the symmetric, positive definite input B matrix. Only the full upper triangle need be supplied and only this part of the B array is not destroyed during the computations. The array B is dimensioned with variable dimensions in the subroutine. Therefore, B must be dimensioned in the calling program with first dimension MAX and second dimension at least N.

E — An output one-dimensional real array containing the eigenvalues in ascending order. The array E is dimensioned with variable dimensions in the subroutine. Therefore, E must be dimensioned in the calling program by at least N.

IV — An input integer eigenvector option parameter.
= 0 Compute all eigenvalues and no eigenvectors.
= 1 Compute all eigenvalues and eigenvectors.

WK — An array for working storage. The array WK may be a multi-dimensional array of any type. The only requirement on WK is that it must occupy at least 2N locations.

IERR — An output integer error code
= 0 Normal return.
= J J^{th} eigenvalue has not been determined after 30 iterations.
= 7N+1 B is detected to be non-positive definite.

Upon return, the calling program should test this parameter.

OUTPUT INFORMATION: The eigenvalues are stored in E by ascending order. The eigenvector associated with the i^{th} eigenvalue is found in the i^{th} column of A. If the error return J is made, eigenvalues and eigenvectors are correct (but eigenvalues may be unordered) for indices 1,2,...,IERR-1. If the error return 7N+1 is made, then no eigenvalues or eigenvectors are computed.

RESTRICTIONS: None

METHOD: The Cholesky decomposition of B into LL^T where L is lower triangular is first performed. The composition $L^{-1}AL^{-T}$ is then performed, resulting in a symmetric matrix with the same eigenvalues as the original matrix system. The eigenvalues and eigenvectors of this symmetric matrix are found by the explicit QL algorithm. These eigenvectors are then transformed into

the eigenvectors of the original matrix system. The eigenvectors are normalized such that $x^TBx = 1$.

ACCURACY: This subroutine is numerically stable; i.e., each computed eigenvalue and its corresponding eigenvector are exact for a matrix problem very close to the original problem. The accuracy of the eigenvalues is dependent upon the condition of B with respect to inversion. If B is well-conditioned, then the largest eigenvalue is computed to approximately 12 significant figures and the smaller eigenvalues will suffer from absolute errors which are no larger. The accuracy of the eigenvectors is dependent on their inherent sensitivity.

REFERENCES:

1. Martin, R. S.; and Wilkinson, J. H.: Reduction of the Symmetric Eigenproblem, $Ax = \lambda Bx$ and Related Problems to Standard Form. Numer. Math., Bd. 11, 1968, pp. 99-110.

2. Martin, R. S.; Reinsch, C.; and Wilkinson, J. H.: Householder's Tridiagonalization of a Symmetric Matrix. Numer. Math. Bd. 11, 1968, pp. 181-195.

3. Bowdler, H.; Martin, R.S.; Reinsch, C.; and Wilkinson, J. H.: The QR and QL Algorithms for Symmetric Matrices. Numer. Math., Bd. 11, 1968, pp. 293-306.

All the references are reprinted in Handbook for Automatic Computations, Volume II, Linear Algebra by J. H. Wilkinson and C. Reinsch, Springer-Verlag, 1971.

STORAGE: 2623_8

SUBPROGRAMS USED:

SYMQL (F2.1)	73_8
QXZ138	426_8
QXZ139	351_8
QXZ140	304_8
QXZ155	277_8
QXZ156	110_8

	QXZ168	264_8
	QXZ262	17_8
	QXZ263	31_8

FORTRAN FUNCTIONS: ABS, AMAX1, AMIN1, IABS, SIGN, SQRT

OTHER CODING INFORMATION: If B is ill-conditioned with respect to inversion, subroutine CSQZ (F2.8) should be used for more accurate results.

EXAMPLE

If a user wanted to compute all the eigenvalues and eigenvectors of a 5 by 5 symmetric generalized eigenproblem, then the user must store at least the full upper triangle of the A and B matrices in two arrays. Let A be stored in an array denoted A and dimensioned 5 by 5 and let B be stored in an array denoted B and dimensioned 5 by 5. (Note that A and B must have the same first dimension). Then the elements of the A matrix, denoted a_{ij}, and of the B matrix, denoted b_{ij}, are stored as follows:

A

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ x & a_{22} & a_{23} & a_{24} & a_{25} \\ x & x & a_{33} & a_{34} & a_{35} \\ x & x & x & a_{44} & a_{45} \\ x & x & x & x & a_{55} \end{bmatrix}$$

B

$$\begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ x & b_{22} & b_{23} & b_{24} & b_{25} \\ x & x & b_{33} & b_{34} & b_{35} \\ x & x & x & b_{44} & b_{45} \\ x & x & x & x & b_{55} \end{bmatrix}$$

where x denotes elements which are ignored by the subroutine. Thus, the full A and B matrices may be stored in A and B respectively, but only the full upper triangle will be used as input. Let the array in which the computed eigenvalues are to be stored be denoted by E and dimensioned by 5. (E must be dimensioned by at least 5). Also, suppose that there exists an array

denoted W which is dimensioned 3 by 2 by 2 and whose contents may be destroyed. Since W occupies 12 locations and 10 locations are needed for a work array, W may be used for this array. Then, a FORTRAN program performing this task could be as follows:

```
      PROGRAM TTTSYMG(OUTPUT,TAPE6=OUTPUT)
C
C   THIS ROUTINE COMPUTES THE COMPLETE EIGENVECTOR
C   SYSTEM OF THE SYMMETRIC GENERALIZED EIGENVALUE
C   PROBLEM BY THE CHOLESKY-WILKINSON ALGORITHM.
C
      DIMENSION A(5,5),E(5),B(5,5),W(3,2,2)
      DATA ((A(I,J),I=1,5),J=1,5)/10.,2.,3.,1.,1.,2.,
     +     12.,1.,2.,1.,3.,1.,11.,1.,-1.,1.,2.,1.,9.,1.,
     +     1.,1.,-1.,1.,15./
      DATA ((B(I,J),I=1,5),J=1,5)/12.,1.,-1.,2.,1.,1.,
     +     14.,1.,-1.,1.,-1.,1.,16.,-1.,1.,2.,-1.,-1.,12.,-1.,
     +     1.,1.,1.,-1.,11./
      MAX = 5
      N = 5
      IV = 1
      WRITE (6,896)
      WRITE (6,899) ((A(I,J),J=1,5),I=1,5)
      WRITE (6,898)
      WRITE (6,899) ((B(I,J),J=1,5),I=1,5)
      CALL SYMGEP(MAX,N,A,B,E,IV,W,IERR)
      IF (IERR.EQ.0)GO TO 100
      WRITE (6,900) IERR
  896 FORMAT(/,1X,37HTHE ORIGINAL SYMMETRIC REAL MATRIX A:)
  898 FORMAT(/,1X,37HTHE ORIGINAL SYMMETRIC REAL MATRIX B:)
  899 FORMAT(/,5(5X,F5.1))
      STOP
  100 CONTINUE
      WRITE (6,903)
      WRITE (6,904) (E(K),K=1,5)
      WRITE (6,905)
      WRITE (6,906) ((A(I,J),I=1,5),J=1,5)
      WRITE (6,900) IERR
  900 FORMAT (/,1X,17HTHE ERROR CODE IS,1X,I2)
  903 FORMAT(/,1X,20HTHE EIGENVALUES ARE:)
  904 FORMAT(/,1X,E20.13)
  905 FORMAT(/,1X,28HTHE NORMALIZED EIGENVECTORS:)
  906 FORMAT(/,1X,5(1X,E20.13,/))
      STOP
      END
```

If the j^{th} eigenvalue is denoted by e_j and v_{ij} denotes the i^{th} component of the eigenvector associated with the j^{th} eigenvalue, then the E and A arrays contain the following elements upon a normal return:

$$E = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \end{bmatrix} \qquad A = \begin{bmatrix} v_{11} & v_{12} & v_{13} & v_{14} & v_{15} \\ v_{21} & v_{22} & v_{23} & v_{24} & v_{25} \\ v_{31} & v_{32} & v_{33} & v_{34} & v_{35} \\ v_{41} & v_{42} & v_{43} & v_{44} & v_{45} \\ v_{51} & v_{52} & v_{53} & v_{54} & v_{55} \end{bmatrix}$$

This is shown by the following output from program TITSYMG

```
THE ORIGINAL SYMMETRIC REAL MATRIX A:

    10.0      2.0      3.0      1.0      1.0
     2.0     12.0      1.0      2.0      1.0
     3.0      1.0     11.0      1.0     -1.0
     1.0      2.0      1.0      9.0      1.0
     1.0      1.0     -1.0      1.0     15.0

THE ORIGINAL SYMMETRIC REAL MATRIX B:

    12.0      1.0     -1.0      2.0      1.0
     1.0     14.0      1.0     -1.0      1.0
    -1.0      1.0     16.0     -1.0      1.0
     2.0     -1.0     -1.0     12.0     -1.0
     1.0      1.0      1.0     -1.0     11.0

THE EIGENVALUES ARE:

  .4327872110170E+00

  .6636627483923E+00

  .9438590046684E+00
```

.1109284540017E+01

.1492353232543E+01

THE NORMALIZED EIGENVECTORS:

.5471538850361E+00
-.2491827216828E+00
-.6419246001273E+00
.4450135642406E+00
-.1686011052691E+00

.2972241326849E+00
.5489568898292E+00
-.4251321104336E+00
-.6552891292303E+00
.1276690446057E-01

-.6238090551130E+00
.5173446408172E+00
-.2435203298675E+00
.4473128625251E+00
-.2895269108842E+00

.5333052571006E+00
.5348374066823E+00
.4543889718684E+00
.4714134667487E+00
.2888696547088E-01

-.2394728063543E+00
.5360233565384E-01
-.2089937021252E+00
.2697610113658E+00
.9073754426081E+00

THE ERROR CODE IS 0

SOURCE: NASA, LaRC, Computer Applications Branch.

QUESTIONS ON THE USE OF THIS PROGRAM SHOULD BE DIRECTED TO THE ACD USER CONSULTATION OFFICE, EXT. 3548.

Appendix B: Example Input and Output

inpt.dat

```
Test Case 42                        ! title (32 characters maximum)
10.0            ! a     - panel length [Length]
30.0            ! b     - panel width [Length]
60.0            ! r     - panel radius [Length]
0.32            ! tf    - panel or facesheet thickness [Length]
0.0             ! tc    - core thickness [Length]
0.245888E+07    ! A11   - orthotropic inplane stiffness [Force/Length]
0.747557E+06    ! A12   - orthotropic inplane stiffness [Force/Length]
0.245888E+07    ! A22   - orthotropic inplane stiffness [Force/Length]
0.855662E+06    ! A66   - orthotropic inplane stiffness [Force/Length]
0.209825E+05    ! D11   - orthotropic bending stiffness [Force-Length]
0.637916E+04    ! D12   - orthotropic bending stiffness [Force-Length]
0.209825E+05    ! D22   - orthotropic bending stiffness [Force-Length]
0.730165E+04    ! D66   - orthotropic bending stiffness [Force-Length]
0.5E+06         ! Gxz   - transverse shear stiffness [Force/Length^2]
0.5E+06         ! Gyz   - transverse shear stiffness [Force/Length^2]
0.0             ! kx0   - linear Nx preload component [Force/Length^2]
0.0             ! cx0   - constant Nx preload component [Force/Length]
0.0             ! ky0   - linear Ny preload component [Force/Length^2]
0.0             ! cy0   - constant Ny preload component [Force/Length]
0.0             ! cxy0  - constant Nxy preload component [Force/Length]
6.67E-02        ! kx    - linear Nx component [Force/Length^2]
-1.0            ! cx    - constant Nx component [Force/Length]
0.0             ! ky    - linear Ny component [Force/Length^2]
0.0             ! cy    - constant Ny component [Force/Length]
1.0             ! cxy   - constant Nxy component [Force/Length]
10              ! m0    - number of terms in x series (integer)
10              ! n0    - number of terms in y series (integer)
1               ! isc   - simply supported/clamped flag (integer)
1               ! iv    - eigensolver flag (integer)
1               ! nmodp - # of eigenvectors written to output file (integer)
```

outp.dat

```
++++++++++
+SCBUCKLE+
++++++++++

Case:  Test Case 42

*** Panel Definition ***
  a  =  0.100000E+02 [Length]
  b  =  0.300000E+02 [Length]
  r  =  0.600000E+02 [Length]
  tf =  0.320000E+00 [Length]
  tc =  0.000000E+00 [Length]

*** Load Definition ***
  kx0  =  0.000000E+00 [Force/Length^2]
  cx0  =  0.000000E+00 [Force/Length]
  ky0  =  0.000000E+00 [Force/Length^2]
  cy0  =  0.000000E+00 [Force/Length]
  cxy0 =  0.000000E+00 [Force/Length]
  kx   =  0.667000E-01 [Force/Length^2]
```

```
  cx    = -0.100000E+01 [Force/Length]
  ky    =  0.000000E+00 [Force/Length^2]
  cy    =  0.000000E+00 [Force/Length]
  cxy   =  0.100000E+01 [Force/Length]

*** Inplane Constants ***
  A11   =  0.245888E+07 [Force/Length]
  A12   =  0.747557E+06 [Force/Length]
  A22   =  0.245888E+07 [Force/Length]
  A66   =  0.855662E+06 [Force/Length]

*** Bending Constants ***
  D11 =  0.209825E+05 [Force-Length]
  D12 =  0.637916E+04 [Force-Length]
  D22 =  0.209825E+05 [Force-Length]
  D66 =  0.730165E+04 [Force-Length]

*** Transverse Shear Constants ***
  Gxz =  0.500000E+06 [Force/Length^2]
  Gyz =  0.500000E+06 [Force/Length^2]

*** Program Execution and Output ***
  m0     =  10
  n0     =  10
  isc    = 1
  iv     = 1
  nmodp =   1



+++++++++++++++
+++ Results +++
+++++++++++++++

--- Warning
--- The first eigenvalue does not have the lowest absolute value
--- Mode 100,      lambda = -0.907928E+04


*** Eigenvalues ***

Mode      lambda

   1       0.908700E+04
   2       0.103833E+05
   3       0.154261E+05
   4       0.159696E+05
   5       0.207969E+05
   6       0.224853E+05
   7       0.233828E+05
   8       0.262262E+05
   9       0.298956E+05
  10       0.310562E+05
  11       0.328190E+05
  12       0.356992E+05
  13       0.402612E+05
  14       0.441724E+05
  15       0.460230E+05
  16       0.483892E+05
  17       0.541446E+05
  18       0.589957E+05
  19       0.623080E+05
  20       0.632010E+05
  21       0.674648E+05
```

22	0.690096E+05
23	0.733284E+05
24	0.757669E+05
25	0.789729E+05
26	0.825614E+05
27	0.861792E+05
28	0.883095E+05
29	0.913011E+05
30	0.934032E+05
31	0.107579E+06
32	0.115188E+06
33	0.126603E+06
34	0.130881E+06
35	0.144769E+06
36	0.151668E+06
37	0.161446E+06
38	0.168785E+06
39	0.200883E+06
40	0.215595E+06
41	0.226502E+06
42	0.228292E+06
43	0.275350E+06
44	0.284341E+06
45	0.448697E+06
46	0.467881E+06
47	0.659725E+06
48	0.698243E+06
49	0.817984E+06
50	0.861281E+06
51	-0.852285E+06
52	-0.809542E+06
53	-0.691439E+06
54	-0.653236E+06
55	-0.463890E+06
56	-0.444889E+06
57	-0.283333E+06
58	-0.274285E+06
59	-0.227505E+06
60	-0.225592E+06
61	-0.214273E+06
62	-0.199933E+06
63	-0.168427E+06
64	-0.161072E+06
65	-0.151252E+06
66	-0.144397E+06
67	-0.130631E+06
68	-0.126358E+06
69	-0.115011E+06
70	-0.107418E+06
71	-0.932592E+05
72	-0.911722E+05
73	-0.881734E+05
74	-0.860516E+05
75	-0.824105E+05
76	-0.788320E+05
77	-0.756697E+05
78	-0.731393E+05
79	-0.688746E+05
80	-0.673460E+05
81	-0.631025E+05
82	-0.622398E+05
83	-0.589280E+05
84	-0.540839E+05
85	-0.483303E+05

86	-0.459742E+05
87	-0.441282E+05
88	-0.402225E+05
89	-0.356657E+05
90	-0.327877E+05
91	-0.310288E+05
92	-0.298654E+05
93	-0.262048E+05
94	-0.233618E+05
95	-0.224666E+05
96	-0.207773E+05
97	-0.159561E+05
98	-0.154127E+05
99	-0.103753E+05
100	-0.907928E+04

*** Eigenvectors ***

Mode 1

m	n	Amn [Length]
1	1	-0.286601E+00
1	2	-0.652203E+00
1	3	-0.614955E+00
1	4	-0.187865E+00
1	5	0.249868E-01
1	6	0.361405E-01
1	7	0.201525E-01
1	8	0.104224E-01
1	9	0.591865E-02
1	10	0.363156E-02
2	1	0.108738E+00
2	2	0.857110E-01
2	3	-0.946775E-01
2	4	-0.178772E+00
2	5	-0.108622E+00
2	6	-0.446994E-01
2	7	-0.179662E-01
2	8	-0.847606E-02
2	9	-0.447371E-02
2	10	-0.224728E-02
3	1	0.441720E-02
3	2	0.233624E-01
3	3	0.327385E-01
3	4	0.993520E-02
3	5	-0.108144E-01
3	6	-0.128838E-01
3	7	-0.926790E-02
3	8	-0.593910E-02
3	9	-0.394769E-02
3	10	-0.255752E-02
4	1	0.254005E-02
4	2	0.211561E-02
4	3	-0.222684E-02
4	4	-0.401106E-02
4	5	-0.247707E-02
4	6	-0.166062E-02
4	7	-0.135056E-02
4	8	-0.117728E-02
4	9	-0.935713E-03
4	10	-0.789628E-03
5	1	0.132101E-03

5	2	0.150827E-02
5	3	0.243229E-02
5	4	0.728031E-03
5	5	-0.100057E-02
5	6	-0.124696E-02
5	7	-0.101230E-02
5	8	-0.768483E-03
5	9	-0.602899E-03
5	10	-0.485602E-03
6	1	0.417151E-03
6	2	0.377364E-03
6	3	-0.384930E-03
6	4	-0.751984E-03
6	5	-0.477391E-03
6	6	-0.317430E-03
6	7	-0.261496E-03
6	8	-0.247351E-03
6	9	-0.213723E-03
6	10	-0.207705E-03
7	1	0.184008E-04
7	2	0.316782E-03
7	3	0.535752E-03
7	4	0.160534E-03
7	5	-0.236204E-03
7	6	-0.298661E-03
7	7	-0.250287E-03
7	8	-0.199534E-03
7	9	-0.164887E-03
7	10	-0.144380E-03
8	1	0.126053E-03
8	2	0.117965E-03
8	3	-0.118395E-03
8	4	-0.239260E-03
8	5	-0.153487E-03
8	6	-0.101536E-03
8	7	-0.835194E-04
8	8	-0.811746E-04
8	9	-0.720956E-04
8	10	-0.742674E-04
9	1	0.503171E-05
9	2	0.109984E-03
9	3	0.189801E-03
9	4	0.569054E-04
9	5	-0.860375E-04
9	6	-0.109373E-03
9	7	-0.927951E-04
9	8	-0.754565E-04
9	9	-0.637054E-04
9	10	-0.579467E-04
10	1	0.525962E-04
10	2	0.498495E-04
10	3	-0.494767E-04
10	4	-0.100961E-03
10	5	-0.646068E-04
10	6	-0.428839E-04
10	7	-0.355222E-04
10	8	-0.353179E-04
10	9	-0.318578E-04
10	10	-0.343392E-04

Appendix C: Array Resizing

The user may wish to resize the arrays in the program for use on smaller or larger computers. The arrays in the main program can be resized by changing the values of mmax, nmax, and mnmax. Although mnmax and nmax are independent quantities, they must have the same numerical value. The quantity mnmax must be set equal to the product mmax*nmax. If these quantities are changed, the arrays in the subroutines SSBUCK and CCBUCK must also be resized to agree with the new values of mmax, nmax, and mnmax. The arrays in these subroutines must be resized as follows:

Subroutine SSBUCK array resizing

Lmn	->	(mmax,nmax)
P1 through P13	->	(mmax,nmax)
Q1, Q2, and Q4	->	(mmax,nmax)
d1 and g1	->	(mmax,mmax)
d2 and g2	->	(nmax,nmax)
bb	->	(mnmax,mnmax)
wk	->	(2,mnmax)

Subroutine CCBUCK array resizing

R1 through R4	->	(mmax,nmax)
S1 through S4	->	(mmax,nmax)
P	->	(48,mmax,nmax)
Q1 through Q4	->	(mmax,nmax)
Q9 through Q12	->	(mmax,nmax)
d1 through d4	->	(mmax+2,nmax+2)
V1 through V10	->	(mnmax,mnmax)
temp1 through temp19	->	(mnmax,mnmax)
bb	->	(mnmax,mnmax)
wk	->	(2,mnmax)

where the numeric values of mmax, nmax, and mnmax are used.

References

1. Stein, Manuel; and Mayers, J.: *A Small-Deflection Theory for Curved Sandwich Plates.* NACA Report 1008, 1951.

2. Cruz, Juan R.: Buckling Analysis of Curved Composite Sandwich Panels Subjected to Inplane Loadings. *Third NASA Advanced Composites Technology Conference-Volume I, Part 2,* NASA CP 3178, June 1992, pp. 919-932.

3. Anon.: *Mathematical and Statistical Software at Langley.* Central Scientific Computing Complex Document N2-3b, March 1987.

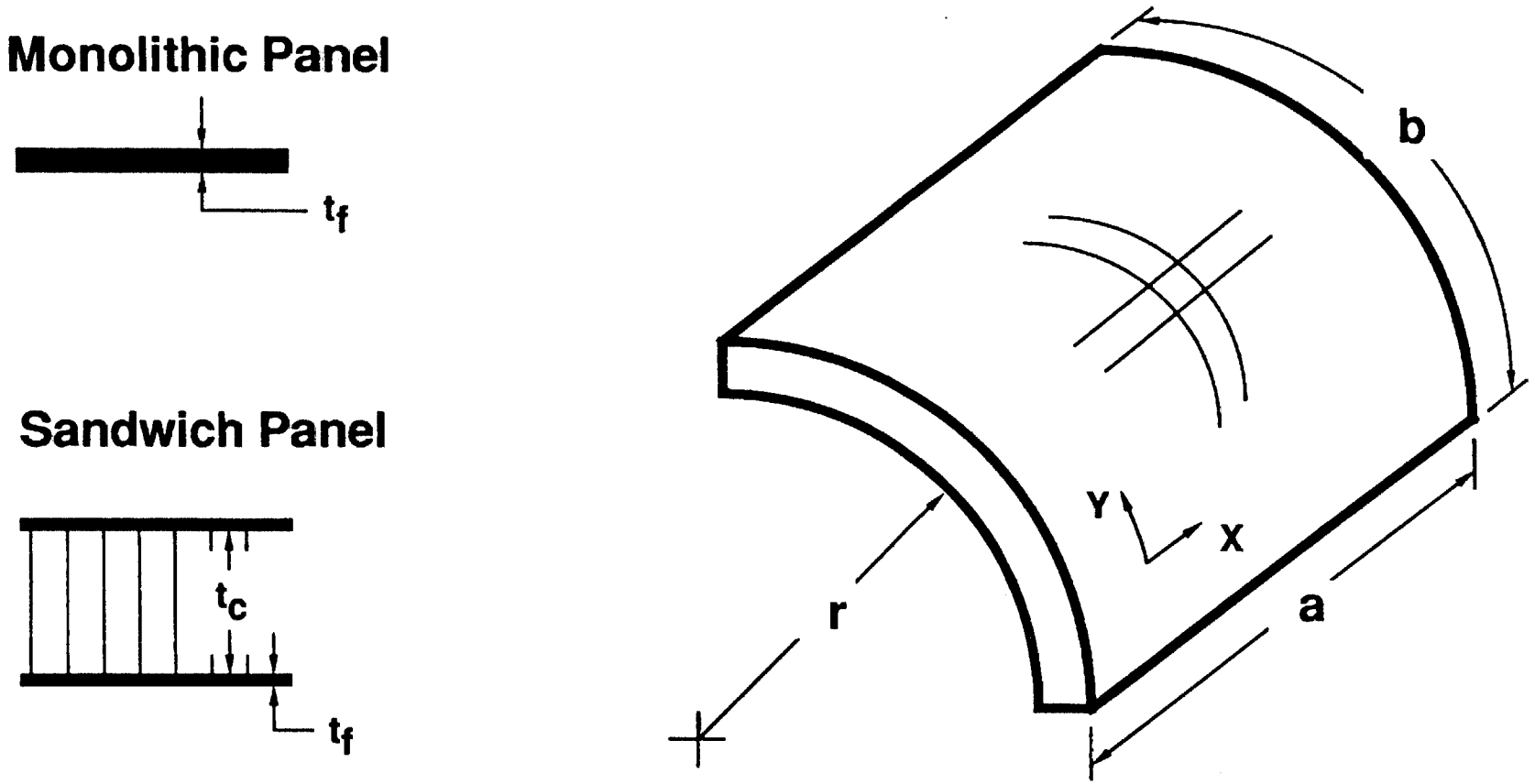

Figure 1. Panel Geometry.

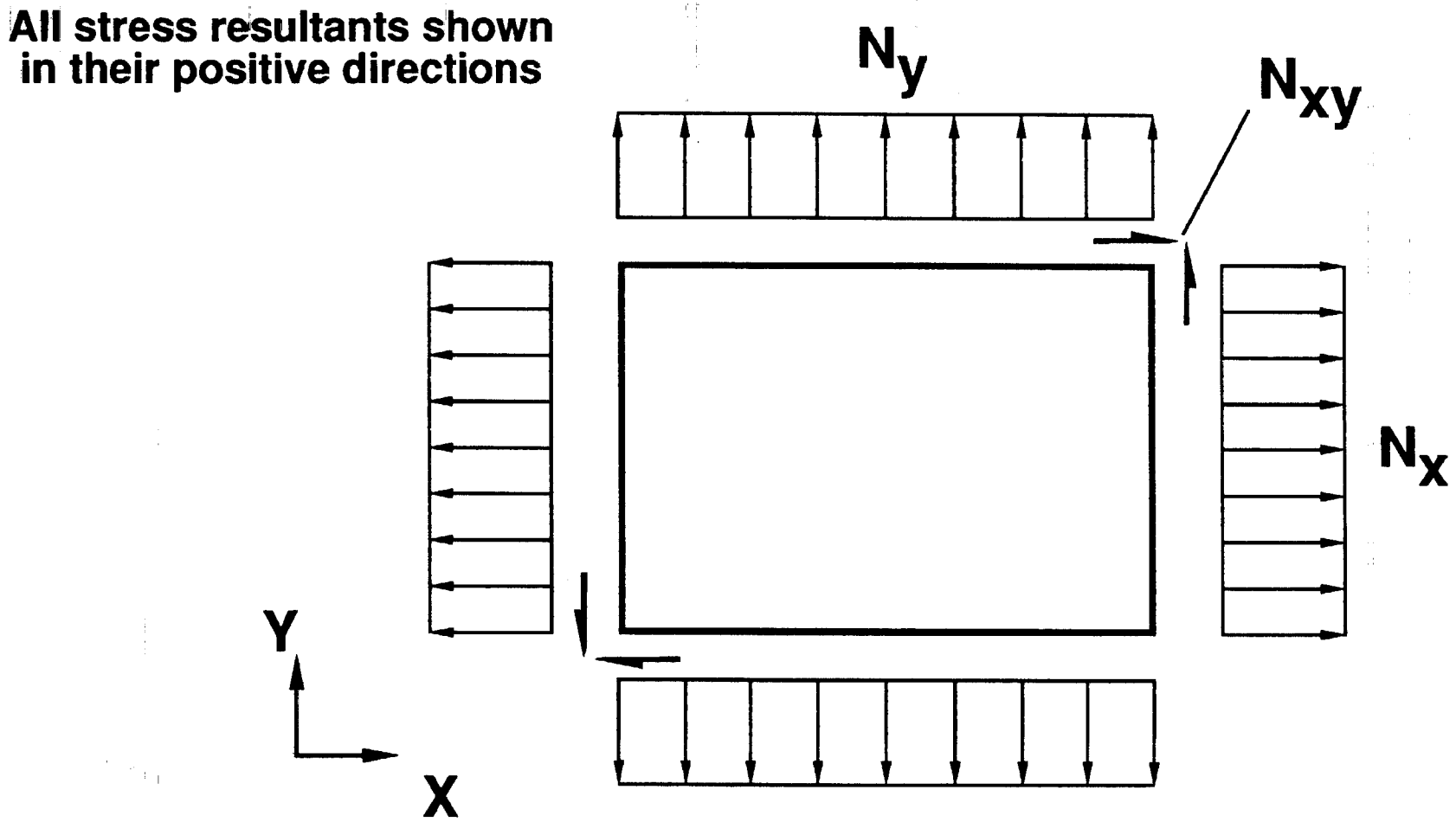

Figure 2. Stress Resultant Sign Convention.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1993	Technical Memorandum

4. TITLE AND SUBTITLE
SCBUCKLE Users Manual
Buckling Analysis Program for Simply Supported and Clamped Panels

5. FUNDING NUMBERS
WU 505-63-50-08

6. AUTHOR(S)
Juan R. Cruz

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23681-0001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA TM-107741

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited

Subject Category 39

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The program SCBUCKLE calculates the buckling loads and mode shapes of cylindrically curved, rectangular panels. The panel is assumed to have no imperfections. SCBUCKLE is capable of analyzing specially orthotropic symmetric panels (i.e., $A_{16} = A_{26} = 0.0$, $D_{16} = D_{26} = 0.0$, $B_{ij} = 0.0$). The analysis includes first-order transverse shear theory, and is thus capable of modeling sandwich panels. The analysis supports two types of boundary conditions; either simply supported or clamped on all four edges. The panel can be subjected to linearly varying normal loads N_x and N_y in addition to a constant shear load N_{xy}. The applied loads can be divided into two parts: a preload component, and a variable (eigenvalue-dependent) component. The analysis is based on the modified Donnell's equations for shallow shells. The governing equations are solved by Galerkin's method.

14. SUBJECT TERMS	15. NUMBER OF PAGES
composites, panel buckling, sandwich structures, shell buckling	26
	16. PRICE CODE: A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	