Int. J. Solids Structures, 1965, Vol. 1, pp. 113 to 138. Pergamon Press Ltd. Printed in Great Britain

COMPUTER ALGORITHMS FOR SOLVING NON-LINEAR PROBLEMS

W. A. MADSEN*, L. B. SMITH† and N. J. HOFF‡

Department of Aeronautics and Astronautics, Stanford University, Stanford, California

Abstract—Many problems in applied mechanics defy exact solution because of the non-linear nature of the descriptive differential equations. In such cases, a solution is often obtained through assumption of appropriate trigonometric or power series, and expansion and collection of coefficients of like trigonometric terms or powers of the variable. The resulting set of non-linear algebraic equations is then used to obtain the solution.

An alternative solution to many problems of this nature consists of developing a potential expression and of minimizing it with respect to the coefficients of the series.

In this paper, algorithms are presented for deriving equations with the aid of a digital computer. The equations are then stored by means of an integer representation. A Newton–Raphson algorithm for minimizing the integer-form potential is also presented. The method is illustrated by re-deriving the solution for the post-buckling behavior of thin-walled circular cylindrical shells under axial compression.

The time required for the computations is short; with a Burroughs B5000 computer the total potential expression was derived in approximately 2 min; and the major stable portion of the load-shortening curve was found in 10 min.

NOTATION

Physical

A	radial displacement expansion coefficient
E	Young's modulus
F	stress function
L	cylindrical shell length
R	cylindrical shell radius
W	energy
n	number of waves around shell circumference
t	cylindrical shell wall thickness
u, v, w	midsurface displacements in the axial, circumferential and inward radial directions respectively
x, y	coordinates in the axial and circumferential directions
$\varepsilon_x, \varepsilon_y, \varepsilon_{xy}$	strains in the shell wall midsurface
ε	end shortening per unit length
η	$n^2(t/R)$
λ_x	half wave length in the axial direction
λ_y	half wave length in the circumferential direction
μ	λ_y/λ_x
ν	Poisson's ratio
σ	axial compressive stress
$\sigma_x, \sigma_y, \sigma_{xy}$	membrane stresses in the shell-wall midsurface

Subscripts

i, j, k	running indices
m, n	subscripts of stress function expansion coefficients
$(\bar{\ })$	non-dimensionalization of w with respect to t; x and y with respect to λ_x and λ_y respectively
e	membrane quantity
b	bending quantity
σ	external load quantity
p	non-dimensional harmonic portion of stress function expansion

Computer

ARG	array containing arguments of a trigonometric product which is to be expanded
TRIG	array containing the types of trigonometric terms, designated by code numbers, present in a trigonometric product which is to be expanded

* Research Assistant. † Programmer Analyst. ‡ Head of Department.

BOO	integer test quantity giving the sign of an exponent
C	array made up of elements which are to be kept in a trigonometric product expansion
D	array of coefficients of an expanded trigonometric product with a 1–1 correspondence to the rows of C
DIM	array giving width of SCF array and depth of C array
EBAR	$\varepsilon R/t$
ETAPWR	a code number for the sign of the exponent of η
NETA	the magnitude of the exponent of η
II	a temporary storage array for deflection coefficient subscript numbers
K	the degree of a product
N	the depth of the C array
NEBAR	exponent of EBAR
NMU	exponent of μ
PROD1	coefficient of term in the potential exclusive of the factor contained in D
S	a temporary storage array telling which assumed trigonometric series are to be multiplied together
SCF	an array containing coefficients of the assumed trigonometric series
SFN	an array containing arguments and types of trigonometric terms of the assumed trigonometric series
SW	a temporary storage array in which algebraic terms are stored for searching before final storage in TERM
TCF	numerical coefficient in the equation being expanded by the method
TERM′	an array representing the algebraic portions of terms in a potential or differential equation through the use of integers
TERM	an array representing the algebraic portions of terms in a potential or differential equation through the use of packed integer words
TERMC	an array containing numerical coefficients of TERM′ or TERM array elements
ITER	an array specifying the number of terms in the coefficient of each trigonometric-expansion-element
JLIM	number of terms in the final potential expression

BASIC CONSIDERATIONS

THERE are many problems in applied mechanics whose solution cannot be obtained explicitly because of non-linearities in the governing differential equations. An approximate solution, however, can often be obtained by assuming a suitable truncated trigonometric or power series. Substitution of the assumed series, expansion and subsequent collection of the coefficients of like trigonometric terms or powers lead to a set of non-linear algebraic equations. For all but the most simple equations, however, this procedure is conducive to error, and requires diligent checking to insure correct results. In fact, if the system becomes too extensive, obtaining the solution is physically almost impossible.

A situation similar to the above exists when a potential energy function is to be minimized, as in the total potential energy approach in structural theory. Here, one generally substitutes deflections in the form of trigonometric series into a potential energy expression. The resulting algebraic expression is then minimized. Again, the difficulty of solution increases rapidly as more terms are included in the series.

To obviate these difficulties, a method was obtained and a representation invented for computer development and simplified computer storage of the equations derived from collecting coefficients of like trigonometric functions. The following example makes clear the situation considered, and also serves to illustrate the roles played by three of the computer algorithms.

Consider the hypothetical equation

$$w_x^2 + u_x = F(x, y), \tag{1}$$

where

$$w = A_1 \cos(\pi x)\cos(\pi y) + A_2 \cos(2\pi x) + A_3 \tag{2a}$$

and

$$u = C_1 \sin(\pi x)\cos(\pi y) + C_2 \sin(2\pi x)\cos(2\pi y) + C_3 \sin(2\pi x) + C_4 \sin(4\pi x) + C_5 \sin(3\pi x)\cos(\pi y). \tag{2b}$$

$F(x, y)$, a known function, is given by

$$F(x, y) = F_1 \cos(\pi x)\cos(\pi y) + F_2 \cos(2\pi x)\cos(2\pi y) + F_3 \cos(3\pi x)\cos(\pi y) + F_4 \cos(2\pi x) + F_5 \cos(4\pi x) + F_6 \cos(2\pi y) + F_7. \tag{2c}$$

Substitution of (2a) into (1) gives

$$[-\pi A_1 \sin(\pi x)\cos(\pi y) - 2\pi A_2 \sin(2\pi x)]^2 + u_x = F(x, y). \tag{3a}$$

When working by hand the first step is expansion of products of series. On the computer this is done through a procedure (i.e. an algorithm—a block of code complete in itself) called SERIESMULT. Application of SERIESMULT to the above expression and substitution in equation (3a) yield

$$\pi^2 A_1^2 \sin^2(\pi x)\cos^2(\pi y) + 4\pi^2 A_1 A_2 \sin(\pi x)\sin(2\pi x)\cos(\pi y) + 4\pi^2 A_2^2 \sin^2(2\pi x) + u_x = F(x, y). \tag{3b}$$

Next, each trigonometric factor has to be expanded in a double trigonometric series and the coefficients of like trigonometric functions have to be collected. This is accomplished through the application of the procedure TRIGSPAND to the series. The above expression becomes

$$\pi^2 A_1^2[\tfrac{1}{4} - (\tfrac{1}{4})\cos(2\pi x) + (\tfrac{1}{4})\cos(2\pi y) - (\tfrac{1}{4})\cos(2\pi x)\cos(2\pi y)] + 4\pi^2 A_1 A_2[(\tfrac{1}{2})\cos(\pi x)\cos(\pi y) - (\tfrac{1}{2})\cos(3\pi x)\cos(\pi y)] + 4\pi^2 A_2^2[\tfrac{1}{2} - (\tfrac{1}{2})\cos(4\pi x)] + u_x = F. \tag{3c}$$

The last step is to collect terms which are coefficients of like trigonometric functions. Of course the computer must know which terms combine, and the procedure given for this discernment is called SEARCHNSTORE. Its application yields

$$[(\pi^2 A_1^2/4) + 2\pi^2 A_2^2 - F_7] + [(-\pi^2 A_1^2/4) + 2\pi C_3 - F_4]\cos(2\pi x) + [(\pi^2 A_1^2/4) - F_6]\cos(2\pi y) + [(-\pi^2 A_1^2/4) + 2\pi C_2 - F_2]\cos(2\pi x)\cos(2\pi y) + (2\pi^2 A_1 A_2 + \pi C_1 - F_1)\cos(\pi x)\cos(\pi y) + (-2\pi^2 A_1 A_2 + 3\pi C_5 - F_3)\cos(3\pi x)\cos(\pi y) + (-2\pi^2 A_2^2 + 4\pi C_4 - F_5)\cos(4\pi x) = 0. \tag{3}$$

If each coefficient of (3) is set equal to zero, a set of equations is obtained for the A_i and C_i in terms of F_i. Of course, in this example, an arbitrary function of y can be added to both u and w.

In actual use the procedures SERIESMULT, TRIGSPAND and SEARCHNSTORE are applied to a typical equation one term at a time. The procedural flow chart shown in Fig. 1 illustrates this. For example, from the above sample problem the term w_x^2 can be considered first. This term is expanded in orderly fashion in SERIESMULT. To each member of this expanded term as it appears TRIGSPAND, then SEARCHNSTORE are applied, and each member is treated in this manner until the term is exhausted. When this occurs, the next term in the equation (u_x, in the example above) is considered and the loop is repeated. Of course, for a linear term such as u_x the process is short. The operations are continued in this manner until every term in the equation has been multiplied, expanded and stored. In the present problem the resulting equation, in the form of data, was then punched onto cards. For convenience, or in case of insufficient machine storage capacity, storage on magnetic tape should present no difficulty.

Algorithms [1] which do somewhat the same thing as SERIESMULT and TRIGSPAND are available. In [1] a method is developed for adding, subtracting and multiplying two trigonometric series if the coefficients are known constants. Thus these algorithms differ from those developed here, since in the present case unknown coefficients are considered and provision is made for terms as high in degree as 8 (i.e. eight truncated trigonometric double series multiplied together).

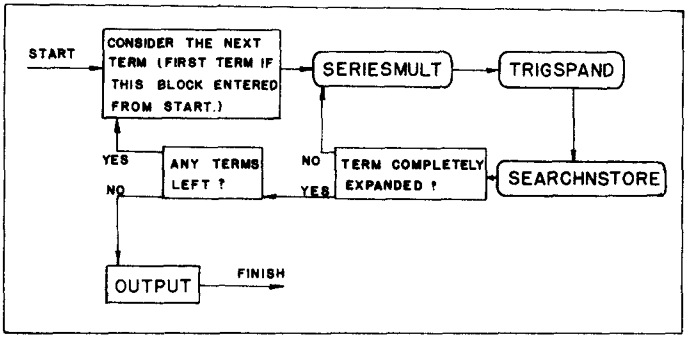


FIG. 1. Procedural flow chart-equation development.

A Newton–Raphson procedure, NEWTNRAPH, was written to solve the equations as derived and stored by the above algorithms.

Some specialization in the derivation of the equations, as well as in the Newton–Raphson process, was required since a parameter μ occurs in non-polynomial form. Most problems have their own peculiarities and the procedures will undoubtedly have to be modified slightly to incorporate them.

The equations for the example problem are relatively simple to derive by hand. However, even for this simple problem machine storage and solution is difficult without the simplified representation introduced. For expressions ten times as long the advantages are much more apparent.

This method can be applied to any problem of the type outlined above. Limitations are imposed only by the degree of the equations, storage requirements and available computer time. The limitation on the degree of the equations is artificial; expanded programs can easily be developed. Programs written in ALGOL as employed for the Burroughs B5000 [2, 3] can be found in [4].

INTEGER REPRESENTATION

Algebraic equations

Integer representation of algebraic terms in the coefficients of trigonometric functions is the heart of the present computer program. The desirability of storing large amounts of information in the computer core, rather than on tape, gave birth to the requirement for representational compactness. The basic idea is as follows.

Consider a polynomial expression in x,

$$c_1x^{\alpha_1}+c_2x^{\alpha_2}+\cdots+c_nx^{\alpha_n}=0. \tag{4}$$

Instead of repeating x in each of the n terms, the expression could simply be written

$$c_1(\ )^{\alpha_1}+c_2(\ )^{\alpha_2}+\cdots c_n(\ )^{\alpha_n}=0, \tag{5}$$

where the presence of x is understood. To record this equation only the coefficients and exponents need be stored, which is of course easily done in a matrix (an array, in computer phraseology) of size $n\times 2$. Thus

$$\text{TERM}'=\begin{pmatrix}\alpha_1 & c_1\\ \alpha_2 & c_2\\ & \vdots\\ \alpha_n & c_n\end{pmatrix}. \tag{6a}$$

In fact, if $\alpha_i=i$, the equation could be stored in a 1-dimensional array, but this is not assumed. To find the kth term in the polynomial (4) only TERM'$[k,1]$ and TERM'$[k,2]$ need be known.

Consider next a more complicated polynomial expression, constructed of various combinations of $A_1\ldots A_6$. For example,

$$c_1A_5A_6+c_2A_2^4A_4^4+c_3A_1A_5^2+\cdots+c_nA_1^2A_3A_6=0. \tag{7}$$

This may be stored in an array $n\times 7$ as

$$\begin{matrix} & (A_1) & (A_2) & (A_3) & (A_4) & (A_5) & (A_6) & (\text{coeff.})\end{matrix}$$

$$\text{TERM}'=\begin{pmatrix}0 & 0 & 0 & 0 & 1 & 1 & c_1\\ 0 & 4 & 0 & 4 & 0 & 0 & c_2\\ 1 & 0 & 0 & 0 & 2 & 0 & c_3\\ & & & \vdots & & & \\ 2 & 0 & 1 & 0 & 0 & 1 & c_n\end{pmatrix} \tag{8a}$$

To extract a term from this equation, say the kth, one writes

$$T_k=\text{TERM}'[k,7]A_1^{\text{TERM}'[k,1]}A_2^{\text{TERM}'[k,2]}\ldots A_6^{\text{TERM}'[k,6]}. \tag{9}$$

Thus for $k=2$,

$$T_2=c_2A_1^0A_2^4A_3^0A_4^4A_5^0A_6^0=c_2A_2^4A_4^4. \tag{10}$$

An even greater simplification can be made if the exponents are positive integers less than ten. Then (7) may be written

$$\text{TERM} = \begin{pmatrix} 000011 \\ 040400 \\ 100020 \\ \vdots \\ 201001 \end{pmatrix} \quad \text{and} \quad \text{TERMC} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{pmatrix}, \tag{8}$$

where the additional array of numerical coefficients was introduced because computer arrays will not accept both integer and non-integer real numbers without the hazard of round-off error during computation (i.e. 1 may become 0·999 . . .). This simplification is obviously necessary for large systems where storage may become a problem. In most digital computers integers of up to 10-digit length may be used—if more than ten unknowns are present, an additional column may be inserted in TERM for each additional ten unknowns.

If in the expanded equation more than one trigonometric-expansion-element is to be kept, and (7) is the coefficient of one of these, then another dimension is added to the TERM and TERMC arrays (8) to record this fact.

So far only integer representation of terms with positive exponents has been considered. There need be no such limitation. When the possibility of a negative exponent exists, two digits are used to record (1) the magnitude of the exponent and (2) its sign. Here the convention is adopted that

$$1 \rightarrow \text{positive exponent and } 0 \rightarrow \text{negative exponent.} \tag{11}$$

Consider two additional variables, μ and η (for this discussion they may be any unknowns). Assume μ always appears to some positive power, while η may have a negative exponent. Then if one wishes to represent

$$c_1 \mu \frac{1}{\eta} A_5 A_6 + c_2 \eta A_2^4 A_4^4 = 0, \tag{12}$$

provision must be made for η to appear in the denominator and TERM′ becomes

$$\begin{array}{l} \quad\;\; (A_1)\;(A_2)\;(A_3)\;(A_4)\;(A_5)\;(A_6)\;(\quad)\;(\text{NMU})\;(\text{NETA})\;(\text{ETAPWR})\;(\text{coeff.}) \\ \text{TERM}' = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & c_1 \\ 0 & 4 & 0 & 4 & 0 & 0 & 0 & 0 & 1 & 1 & c_2 \end{pmatrix} \end{array} \tag{13a}$$

while

$$\text{TERM} = \begin{pmatrix} 0000110110 \\ 0404000011 \end{pmatrix} \quad \text{and} \quad \text{TERMC} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \tag{13}$$

where NMU and NETA are magnitudes of the powers of μ and η and ETAPWR is defined according to (11).

The above machinery is sufficient for many problems. It is necessary but not sufficient for the problem discussed in the Application Section.

There, quantities of the type $(m^2\mu^2+n^2)^2$, where m and n are integers less than or equal to 9, may occur in either the numerator or denominator of a term. A test quantity as defined by (11) must therefore be stored, as well as m and n. In addition, a parameter $\varepsilon R/t$ (defined in the notation) raised to a small positive integer power, NEBAR, appears in the analysis. Then

$$c_1\mu\frac{1}{\eta}(2^2\mu^2+4^2)^2A_5A_6+c_2\eta\frac{1}{(2^2\mu^2+3^2)^2}(\varepsilon R/t)A_2^4A_4^4 \tag{14}$$

is written in TERM′ as

$$\begin{array}{l} \qquad\quad (A_1)\,(A_2)\,(A_3)\,(A_4)\,(A_5)\,(A_6)\,(\;)\,(\text{NMU})\,(\text{NETA})\,(\text{ETAPWR}) \\ \text{TERM}' = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 4 & 0 & 4 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}$$

$$\begin{array}{ccccc} (\text{BOO}) & (m) & (n) & (\qquad) & (\text{NEBAR}) & (\text{coeff.}) \\ 1 & 2 & 4 & 0\ldots0 & 0 & c_1 \\ 0 & 2 & 3 & 0\ldots0 & 1 & c_2 \end{array} \tag{15a}$$

which leads to a TERM and TERMC written as

$$\text{TERM} = \begin{pmatrix} 0000110110 & 1240000000 \\ 0404000011 & 0230000001 \end{pmatrix} \qquad \text{TERMC} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}. \tag{15}$$

To dissect a term for use an integer division routine (dividing by powers of ten) is applied. Thereby the coefficient exponents and other pieces of information are found one by one.

Trigonometric series

A simple method of integer representation of trigonometric series is needed while equations as given above are being derived. Trigonometric function arguments are represented by simple integers, while the code

$$1 \rightarrow \cos, \qquad 0 \rightarrow \sin \tag{16}$$

is used to denote the trigonometric functions.

In case (iii) of the example in the Application Section the radial displacement is given by

$$\bar{w} = (w/t) = A_1\cos(\pi\bar{x})\cos(\pi\bar{y})+A_2\cos(2\pi\bar{x})\cos(2\pi\bar{y})+A_3\cos(3\pi\bar{x})\cos(3\pi\bar{y})$$
$$+A_4\cos(2\pi\bar{x})+A_5\cos(4\pi\bar{x})+A_6\cos(6\pi\bar{x})+A_7$$

where the subscripts are renumbered for simplicity.

SFN and SCF arrays, each with 6 rows, are defined. The rows contain information on $\bar{w}$, $\bar{w}_{\bar{x}}$, $\bar{w}_{\bar{y}}$, $\bar{w}_{\bar{x}\bar{x}}$, $\bar{w}_{\bar{x}\bar{y}}$ and $\bar{w}_{\bar{y}\bar{y}}$, in order. The 7 columns of SCF contain the numerical factors of the terms of the series. Columns 1 and 2 of SFN show the types of trigonometric series we are dealing with, $\Sigma A_{ij}\sin(i\pi\bar{x})\sin(j\pi\bar{y})$, $\Sigma A_{ij}\sin(i\pi\bar{x})\cos(j\pi\bar{y})$, $\Sigma A_{ij}\cos(i\pi\bar{x})\sin(j\pi\bar{y})$

or $\Sigma A_{ij}\cos(i\pi\bar{x})\cos(j\pi\bar{y})$, by an application of the convention of (16). If an unknown consists of two or more types, for example $w = \Sigma A_{ij}\sin(i\pi\bar{x})\sin(j\pi\bar{y})+\Sigma B_{ij}\cos(i\pi\bar{x})\sin(j\pi\bar{y})$ it can be written as the sum of two or more new unknowns, each homogeneous in a trigonometric type. Thus $w = w_1+w_2$, where $w_1 = \Sigma A_{ij}\sin(i\pi\bar{x})\sin(j\pi\bar{y})$ and $w_2 = \Sigma B_{ij}\cos(i\pi\bar{x})\sin(j\pi\bar{y})$. Columns 3–9 and 10–16 of SFN give the trigonometric arguments for $\bar{x}$ and $\bar{y}$ terms, respectively, while the number of the last term in the series whose coefficient is non-zero is stored in column 17. Thus

$$\begin{array}{cccc} \left(\begin{array}{c}\text{Trig.}\\ \text{types}\\ \bar{x}\ \bar{y}\end{array}\right) & (\bar{x}\text{-function arguments}) & (\bar{y}\text{-function arguments}) & \left(\begin{array}{c}\text{Last}\\ \text{non-}\\ \text{zero}\\ \text{coeff.}\end{array}\right) \end{array}$$

$$\mathrm{SFN} = \begin{pmatrix} 1\ 1 & 1\ 2\ 3\ 2\ 4\ 6\ 0 & 1\ 2\ 3\ 0\ 0\ 0\ 0 & 7 \\ 0\ 1 & 1\ 2\ 3\ 2\ 4\ 6\ 0 & 1\ 2\ 3\ 0\ 0\ 0\ 0 & 6 \\ 1\ 0 & 1\ 2\ 3\ 2\ 4\ 6\ 0 & 1\ 2\ 3\ 0\ 0\ 0\ 0 & 3 \\ 1\ 1 & 1\ 2\ 3\ 2\ 4\ 6\ 0 & 1\ 2\ 3\ 0\ 0\ 0\ 0 & 6 \\ 0\ 0 & 1\ 2\ 3\ 2\ 4\ 6\ 0 & 1\ 2\ 3\ 0\ 0\ 0\ 0 & 3 \\ 1\ 1 & 1\ 2\ 3\ 2\ 4\ 6\ 0 & 1\ 2\ 3\ 0\ 0\ 0\ 0 & 3 \end{pmatrix} \begin{array}{l} (\bar{w}) \\ (\bar{w}_{\bar{x}}) \\ (\bar{w}_{\bar{y}}) \\ (\bar{w}_{\bar{x}\bar{x}}) \\ (\bar{w}_{\bar{x}\bar{y}}) \\ (\bar{w}_{\bar{y}\bar{y}}) \end{array} \tag{17}$$

and

$$\text{(Coefficients)}$$

$$\mathrm{SCF} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -\pi & -2\pi & -3\pi & -2\pi & -4\pi & -6\pi & 0 \\ -\pi & -2\pi & -3\pi & 0 & 0 & 0 & 0 \\ -\pi^2 & -4\pi^2 & -9\pi^2 & -4\pi^2 & -16\pi^2 & -36\pi^2 & 0 \\ +\pi^2 & +4\pi^2 & +9\pi^2 & 0 & 0 & 0 & 0 \\ -\pi^2 & -4\pi^2 & -9\pi^2 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{array}{l} (\bar{w}) \\ (\bar{w}_{\bar{x}}) \\ (\bar{w}_{\bar{x}}) \\ (\bar{w}_{\bar{x}\bar{x}}) \\ (\bar{w}_{\bar{x}\bar{y}}) \\ (\bar{w}_{\bar{y}\bar{y}}). \end{array} \tag{18}$$

PROCEDURE TRIGSPAND

TRIGSPAND was developed to give the Fourier expansion of a product of trigonometric terms of the type

$$\left\{\left[\begin{pmatrix}\sin\\ \cos\end{pmatrix}(n_1\pi x)\right]\left[\begin{pmatrix}\sin\\ \cos\end{pmatrix}(n_2\pi x)\right]\cdots\left[\begin{pmatrix}\sin\\ \cos\end{pmatrix}(n_8\pi x)\right]\right\} \times\left\{\left[\begin{pmatrix}\sin\\ \cos\end{pmatrix}(m_1\pi y)\right]\left[\begin{pmatrix}\sin\\ \cos\end{pmatrix}(m_2\pi y)\right]\cdots\left[\begin{pmatrix}\sin\\ \cos\end{pmatrix}(m_8\pi y)\right]\right\} \tag{19}$$

where there are up to eight terms in each of x and y. In doing so, repeated use is made of identities such as

$$\cos(ax)\cos(bx) = [\cos(a+b)x+\cos(a-b)x]/2. \tag{20}$$

Expression (19) includes, of course, the single Fourier series when either the function of x or that of y is a constant.

TRIGSPAND furnishes a complete expansion of the product. However, at times only a portion of this expansion is required. This is notably true when in an expansion of the total potential integrand prior to integration only the constant term need be kept because the contributions of the higher harmonics to the integral vanish. The trigonometric terms to be kept in the expansion are specified by the array C, which is an input array to TRIGSPAND.

The other input parameters are K, the number of terms in the product ($K \leq 8$), N, the number of rows in C and the arrays TRIG and ARG. Elements 1 through K and 9 through $8+K$ of TRIG are 0 or 1, with

$$1 \rightarrow \cos \quad \text{and} \quad 0 \rightarrow \sin. \tag{21}$$

Corresponding elements of the array ARG are the input arguments. C is a two-dimensional array. In each row we store the two types of trigonometric terms (0 or 1), and two arguments. Each row of C corresponds to a term which must be kept should it occur in the expansion.

Setting C[1, 2] = 2 is a signal to the procedure that a Fourier expansion of a function of a single variable is under consideration.

D, the output array, gives the magnitude of the coefficients of each term of type C which is represented in the Fourier series.

As an example of input and output, suppose the product in question is

$$\cos(\pi x)\cos(3\pi x)\sin(\pi y)\cos(3\pi y), \tag{22a}$$

then $K = 2$,

$$\begin{array}{rll} & (x\text{-function}) & (y\text{-function}) \\ \text{TRIG} = (1 & 1\ 0\ 0\ 0\ 0\ 0\ 0 & 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0) \\ \text{ARG} = (1 & 3\ 0\ 0\ 0\ 0\ 0\ 0 & 1\ 3\ 0\ 0\ 0\ 0\ 0\ 0). \end{array} \tag{22}$$

If the terms to be kept in the Fourier series are

$$\begin{array}{c} \sin(2\pi x)\sin(4\pi y), \\ \cos(4\pi x)\sin(2\pi y), \\ \sin(\pi x)\cos(3\pi y) \\ \cos(2\pi x)\sin(2\pi y), \end{array} \tag{23a}$$

and

then $N = 4$ and

$$\begin{array}{cc} \begin{pmatrix}\text{Trig.}\\ \text{types}\\ x\ \ y\end{pmatrix} & \begin{pmatrix}\text{Arguments}\\ x\ \ y\end{pmatrix} \end{array}$$

$$C = \begin{pmatrix} 0 & 0 & 2 & 4 \\ 1 & 0 & 4 & 2 \\ 0 & 1 & 1 & 3 \\ 1 & 0 & 2 & 2 \end{pmatrix}. \tag{23}$$

The output is

$$-(\tfrac{1}{4})\cos(4\pi x)\sin(2\pi y)-(\tfrac{1}{4})\cos(2\pi x)\sin(2\pi y), \tag{24a}$$

and therefore D is given by

$$D = \begin{pmatrix} 0\cdot 0 \\ -0\cdot 25 \\ 0\cdot 0 \\ -0\cdot 25 \end{pmatrix}. \tag{24}$$

PROCEDURE SEARCHNSTORE

Construction of each term in either a potential expression or a given trigonometric-expansion-element coefficient takes place in SEARCHNSTORE, as explained in the Section on Integer Representation. Input to SEARCHNSTORE for this construction in the numerical example treated under the Section Application is obtained from SERIESMULT. The trigonometric-expansion-elements which a given term (e.g. A_1^2) multiplies are also input to SEARCHNSTORE; they are found by TRIGSPAND.

Next, a search is performed in the potential expression or in the coefficients of eligible trigonometric-expansion-element coefficients (as deduced by TRIGSPAND) for terms containing like unknowns and parameters. If such a term is already present in the trigonometric-expansion-element coefficient, the numerical multiplier of the new term is added to that of the old one; otherwise, the new term is added to the equation or trigonometric-expansion-element coefficient. The algebraic portion of each term is stored in an array TERM. If more than ten integer bits of information are required, TERM becomes 3-dimensional, as in our sample problem. Numerical multipliers of each term are stored in an array TERMC, with a one-to-one correspondence to the first two subscripts of TERM. TERM and TERMC then are both input and output of SEARCHNSTORE.

The II array tells which terms of the assumed trigonometric expansion are present (that is, the A_js in the Section Integer Representation), while from S the degree of the product, K, is determined.

PROD1 is the numerical multiplier of a term under consideration except for the factors to be found in the D array (e.g. in equation (3c) PROD1 for the A_1^2 term is π^2). Thus multiplication of the D array by PROD1 yields the numerical multipliers to be added to TERMC. D is found by TRIGSPAND; and PROD1 by SERIESMULT.

NMU, NETA, ETAPWR and NEBAR are integers, defined in the Section on Integer Representation. These quantities are stored with the elements of II to form a member of TERM.

SEARCHNSTORE can be illustrated through a simple example. Suppose that

$$II = (3\ 6\ 0\ 0\ 0\ 0\ 0\ 0),$$

$$NMU = 2, \quad NETA = 1, \quad ETAPWR = 0 \quad \text{and} \quad NEBAR = 0. \tag{25}$$

Also suppose that D is as found from the simple case presented in the section on TRIGSPAND,

$$D = (0, -0\cdot 25, \ 0, -0\cdot 25), \tag{24}$$

with PROD1 = 3·0. The unknowns are stored as in Integer Representation. During construction and searching a temporary storage array SW is defined so that

$$\text{SW}[1] = 0010010210 \quad \text{and} \quad \text{SW}[2] = 0000000000 \tag{26}$$

are the integers comprising the algebraic portion of the term. (The second integer was used because this is only the storage scheme for compatibility; additional multiplying factors, such as $(m^2\mu^2+n^2)^2$, arising in the calculation of the potential will be stored later.) This algebraic portion will be entered in some

$$\text{TERM}[2, I_2, 1], \qquad \text{TERM}[2, I_2, 2], \tag{27a}$$

as well as

$$\text{TERM}[4, I_4, 1], \qquad \text{TERM}[4, I_4, 2]. \tag{27b}$$

The coefficients obtained from multiplying PROD1 by the D array are −0·75 and −0·75; they are added to TERMC[2, I_2] and TERMC[4, I_4] respectively. I_2 and I_4 depend on the algebraic terms already entered. ITER is an array containing the number of terms in the potential or in each trigonometric-expansion-element coefficient. I_2 must be less than or equal to ITER[2], while I_4 must be less than or equal to ITER[4]. If algebraic terms like SW[1] and SW[2] have not been previously entered in TERM, a new entry is made and the corresponding values of ITER are increased by one.

PROCEDURE FCALC

It often happens that many terms in the equation to be expanded have factors of second or higher degree in either an unknown function or derivative. FCALC, called by SERIESMULT, is a time-saver in expanding such terms.

For unknowns containing only two elements in their series representation a call on FCALC yields the binomial coefficients $\binom{i}{j}$. For more than two elements equivalent coefficients are found. Thus duplication of calls on TRIGSPAND and SEARCHNSTORE is omitted. This procedure is especially useful on third and fourth degree terms. For example, for a 3-element series,

$$(a+b+c)^3 = a^3+b^3+c^3+3a^2b+3a^2c+3ab^2+3ac^2+6abc+3b^2c+3bc^2. \tag{28}$$

Thus only ten calls on TRIGSPAND and SEARCHNSTORE are required, rather than twenty-seven; the latter number of calls would be necessary in a direct expansion where such terms as abc and acb would be treated separately. Thus a time saving of approximately 60 per cent is realized.

PROCEDURE SERIESMULT

SERIESMULT is the principal procedure. It furnishes information to TRIGSPAND through K, N, ARG, TRIG and C, and then to SEARCHNSTORE in the form of II, S, D, PROD1, ITER, TERM, TERMC, NMU, NETA, ETAPWR and NEBAR. SERIESMULT is the procedure actually 'called' in an application of the method. It in turn calls TRIGSPAND, SEARCHNSTORE and FCALC.

A call on SERIESMULT means that the assumed trigonometric series for the unknowns are substituted into one term of the differential equation or potential expression, and the result processed by TRIGSPAND and SEARCHNSTORE as shown in Basic Considerations. Thus in the introductory example one call on SERIESMULT is equivalent to expanding, searching and storing either w_x^2, u_x or F. Terms up to the eighth degree may be processed by SERIESMULT as given in detail in the program of Appendix B1.

Input quantities to SERIESMULT are: (TERM, TERMC, NMU, NETA, ETAPWR, NEBAR, SFN, SCF, C, ITER, TCF, S, DIM). TERM, TERMC, NMU, NETA, ETAPWR, NEBAR, SFN and SCF are explained in Integer Representation, C in TRIGSPAND and ITER in SEARCHNSTORE.

TCF is the numerical coefficient of the term under consideration. The array S contains the numbers of the terms (rows) of the companion arrays SFN and SCF which are to be multiplied together. The ninth position of S is the degree of the term.

DIM gives the width of SFN and the depth of the C array.

PROCEDURE NEWTNRAPH

The Newton–Raphson iterative technique was chosen to solve the non-linear algebraic equations derived in the preceding sections. To find the solutions A_i of the set of equations

$$F_j(A_i) = 0, \qquad j = 1, 2, \ldots, n, \tag{29}$$

the iterative process

$$\{A_j\}^{(k+1)} = \{A_j\}^{(k)} - (\partial F_j/\partial A_i)^{-1}\{F_i\} \tag{30}$$

is used. This converges if a solution A_i exists, $(\partial F_j/\partial A_i)$ is not singular at the solution, and the $A_i^{(0)}$ are 'near enough' to the A_i as shown in [5]. To minimize a function U, F_j is set equal to $\partial U/\partial A_j$ in (30).

NEWTNRAPH as included in this report is specialized to the case of the minimization of a function. The differentiation scheme used in NEWTNRAPH could also be employed in solving a set of simultaneous nonlinear algebraic equations.

Arguments are the arrays TERM, TERMC and SOLN, the integer JLIM and the parameter EBAR (actually $\varepsilon R/t$). SOLN is the solution vector, JLIM the number of terms in the total potential energy and EBAR is the end shortening of the cylindrical shell in the numerical example treated in the Section entitled Application.

Each call on NEWTNRAPH is equivalent to one application of the formula (30) to the system, with of course $F_j = \partial U/\partial A_j$. For the example problem (eight unknowns) the solution at a given EBAR was used as a set of starting values for finding a solution at EBAR + 0·4. For this increment the number of iterations required to find a solution of satisfactory accuracy was usually five to eight.

APPLICATION

The method described was used to find the solution for the post-buckling behavior of thin-walled circular cylindrical shells under axial compression, [6–8]. The shell geometry is shown in Fig. 2.

The three cases treated are distinguished by the number and types of terms taken in the w deflection expansion. Given w as shown in (A-15), the coefficients retained are

$$\begin{aligned} \text{case}\quad &(\text{i}): A_{00}, A_{11}, A_{20}, A_{02};\\ &(\text{ii}): A_{00}, A_{11}, A_{22}, A_{20}, A_{40};\\ &(\text{iii}): A_{00}, A_{11}, A_{22}, A_{33}, A_{20}, A_{40}, A_{60}. \end{aligned} \tag{31a, b, c}$$

Case (i) was solved first by Kempner, [7], while (ii) and (iii) were treated by Almroth, [8]. As an example, case (iii) is discussed here in some detail. For convenience one may write

$$w = t(A_1 \cos(\pi x)\cos(\pi y) + A_2 \cos(2\pi x)\cos(2\pi y) + A_3 \cos(3\pi x)\cos(3\pi y) + A_4 \cos(2\pi x) + A_5 \cos(4\pi x) + A_6 \cos(6\pi x) + A_7) \quad (32)$$

The equations governing the post-buckling behavior are given in Appendix A.

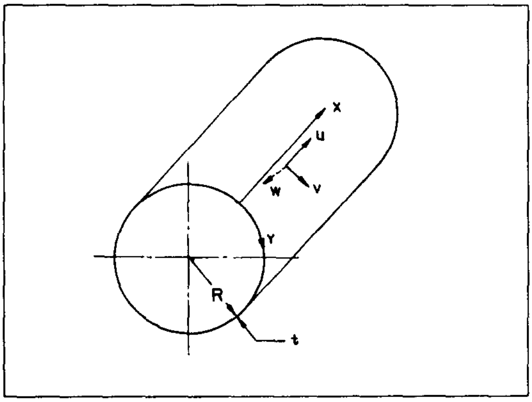


FIG. 2. Circular cylindrical shell geometry.

The method of this paper may be applied twice to this problem. First a total potential expression is found by a direct substitution of (A-15) and (A-16) into (A-17). Automated derivation, though possible, was omitted because orthogonality permitted derivation by hand without difficulty. The computer method was employed, however, to find the coefficients F_i of the stress function from the A_j through use of (A-18). The form of F, and therefore the composition of the C Array, was found by a preliminary multiplication of $\bar{w}_{\bar{x}\bar{x}}$ by $\bar{w}_{\bar{y}\bar{y}}$ and for case (iii) was assumed as

$$\begin{aligned} F_p = {} & F_1 \cos(\pi\bar{x})\cos(\pi\bar{y}) + F_2 \cos(2\pi\bar{x})\cos(2\pi\bar{y}) + F_3 \cos(3\pi\bar{x})\cos(3\pi\bar{y}) \\ & + F_4 \cos(4\pi\bar{x})\cos(4\pi\bar{y}) + F_5 \cos(5\pi\bar{x})\cos(5\pi\bar{y}) + F_6 \cos(6\pi\bar{x})\cos(6\pi\bar{y}) + F_7 \cos(\pi\bar{x})\cos(3\pi\bar{y}) \\ & + F_8 \cos(3\pi\bar{x})\cos(\pi\bar{y}) + F_9 \cos(\pi\bar{x})\cos(5\pi\bar{y}) + F_{10} \cos(2\pi\bar{x})\cos(4\pi\bar{y}) + F_{11} \cos(4\pi\bar{x})\cos(2\pi\bar{y}) \\ & + F_{12} \cos(5\pi\bar{x})\cos(\pi\bar{y}) + F_{13} \cos(5\pi\bar{x})\cos(3\pi\bar{y}) + F_{14} \cos(6\pi\bar{x})\cos(2\pi\bar{y}) + F_{15} \cos(7\pi\bar{x})\cos(\pi\bar{y}) \\ & + F_{16} \cos(7\pi\bar{x})\cos(3\pi\bar{y}) + F_{17} \cos(8\pi\bar{x})\cos(2\pi\bar{y}) + F_{18} \cos(9\pi\bar{x})\cos(3\pi\bar{y}) + F_{19} \cos(2\pi\bar{x}) \\ & + F_{20} \cos(4\pi\bar{x}) + F_{21} \cos(6\pi\bar{x}) + F_{22} \cos(2\pi\bar{y}) + F_{23} \cos(4\pi\bar{y}) + F_{24} \cos(6\pi\bar{y}) \end{aligned} \quad (33)$$

Finally, synthesization of these results was required to derive an equation completely in terms of the A_i, μ, η, and $(\varepsilon R/t)$. Integer representations of the various equations as well as the programs used, may be found in Appendix B of [4].

Post-buckling curves for the three cases were obtained quite simply once starting values were found. Results, which agree with those of [7], [8] appear in Figs. 3–7.

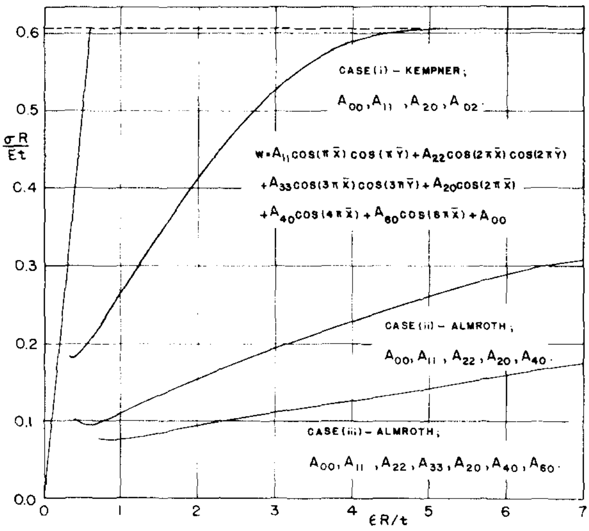


FIG. 3. Axial compression vs. end shortening.

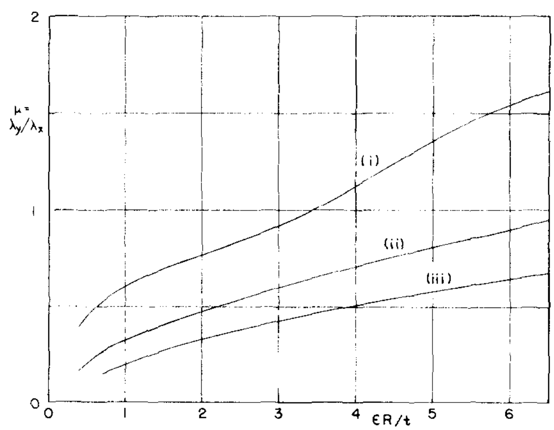


FIG. 4. Wave length ratio vs. end shortening.

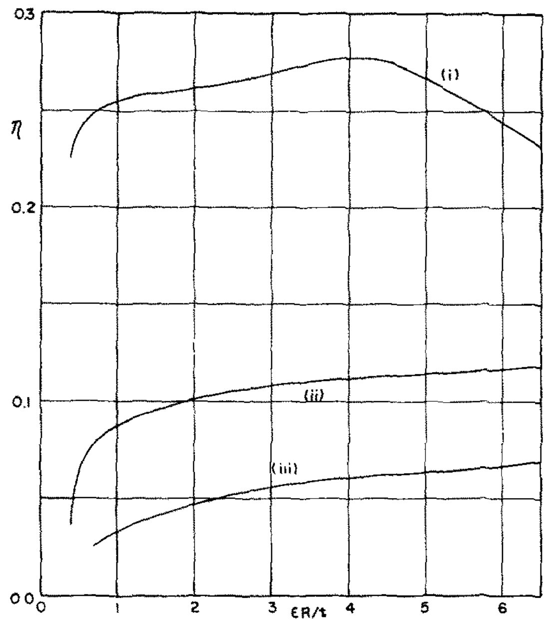


FIG. 5. $\eta = n^2(t/R)$ vs. end shortening.

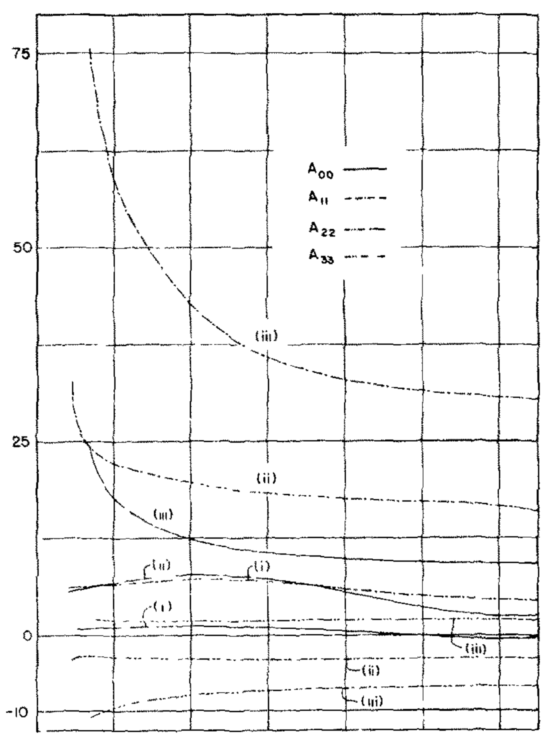


FIG. 6. w coefficients A_{00}, A_{11}, A_{22}, A_{33}, vs. end shortening.

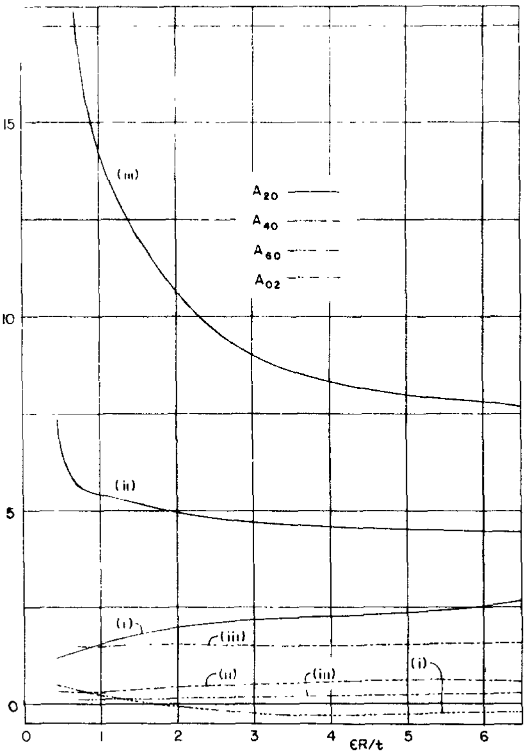


FIG. 7. w coefficients A_{20}, A_{40}, A_{60}, A_{02} vs. end shortening.

Acknowledgements—The work here reported was sponsored by the U.S. Air Force Office of Scientific Research under contract No. AF 49(638)1276.

REFERENCES

[1] M. L. CHARNOW, *Fourier Series Operating Package*, NACA Report No. TN D-1078 (1961).
[2] D. D. MCCRACKEN, *A Guide to Algol Programming*, John Wiley, New York (1962).
[3] *Extended Algol Reference Manual for the Burroughs B5000* (5000–21012), Burroughs Corporation, Detroit, Michigan (1962).
[4] W. A. MADSEN, L. B. SMITH and N. J. HOFF, *Automated Derivation and Integer Representation of Total Potential and Differential Equation Expansions for Assumed Trigonometric Series with an Application to the Post-Buckling Behavior of Circular Cylindrical Shells*, Department of Aeronautics and Astronautics, Stanford University, SUDAER No. 201 (1964).
[5] P. HENRICI, *Lecture Notes on Elementary Numerical Analysis*, John Wiley, New York (1962).
[6] TH. VON KARMAN and H. S. TSIEN, *J. aero. Sci.* **8**, 303 (1941).
[7] J. KEMPNER, *J. aero. Sci.* **17**, 329 (1954).
[8] B. O. ALMROTH, *AIAA J.* **1**, 630 (1963).

APPENDIX A

Equations for determination of the post-buckling behavior of thin-walled circular cylindrical shells

Basic equations

The equations used in this solution are the same as those used by earlier investigators, [6–8].

The strain–displacement relations, valid for deflections of order t and small rotations, are

$$\varepsilon_x = u_x + w_x^2/2, \tag{A-1}$$

$$\varepsilon_y = v_y + w_y^2/2 - w/R \tag{A-2}$$

$$\varepsilon_{xy} = u_y + v_x + w_x w_y. \tag{A-3}$$

Linear elastic behavior is assumed and Hooke's Law is written as

$$\sigma_x = E(\varepsilon_x + \nu\varepsilon_y)/(1-\nu^2), \tag{A-4}$$

$$\sigma_y = E(\varepsilon_y + \nu\varepsilon_x)/(1-\nu^2) \tag{A-5}$$

$$\sigma_{xy} = E\varepsilon_{xy}/2(1+\nu). \tag{A-6}$$

The membrane and bending strain energies are given respectively by

$$W_e = \frac{t}{2E}\int_0^L\int_0^{2\pi R}[(\sigma_x+\sigma_y)^2 - 2(1+\nu)(\sigma_x\sigma_y - \sigma_{xy}^2)]\,\mathrm{d}x\,\mathrm{d}y \tag{A-7}$$

and

$$W_b = \frac{Et^3}{24(1-\nu^2)}\int_0^L\int_0^{2\pi R}[(w_{xx}+w_{yy})^2 - 2(1-\nu)(w_{xx}w_{yy} - w_{xy}^2)]\,\mathrm{d}x\,\mathrm{d}y. \tag{A-8}$$

In the bending energy expression the assumption is made that the curvatures and twist are given to a sufficient degree of accuracy by the second derivatives of w. The potential of the axial load is

$$W_\sigma = -t\int_0^{2\pi R}(\sigma_x)_{x=L}\,\mathrm{d}y\int_0^L u_x\,\mathrm{d}x. \tag{A-9}$$

Variation of the total potential energy $W_e + W_b + W_\sigma$ with respect to the three displacements u, v and w yields three equations of equilibrium, given by

$$\sigma_{x,x} + \sigma_{xy,y} = 0, \tag{A-10}$$

$$\sigma_{y,y} + \sigma_{xy,x} = 0 \tag{A-11}$$

and

$$\frac{D}{t}\nabla^4 w = \frac{\partial}{\partial x}(\sigma_x w_x) + \frac{\partial}{\partial y}(\sigma_{xy} w_x) + \frac{\partial}{\partial x}(\sigma_{xy} w_y) + \frac{\partial}{\partial y}(\sigma_y w_y) + \frac{\sigma_y}{R}. \tag{A-12}$$

The usual Airy stress function is next introduced, with

$$\sigma_x = F_{yy}, \qquad \sigma_y = F_{xx}, \qquad \sigma_{xy} = -F_{xy}. \tag{A-13}$$

Thus equations (A-10) and (A-11) are satisfied implicitly, while the direct method of the calculus of variations is used to solve (A-12). The total potential must first be expressed in terms of radial displacements. To do this, F is found as a function of w. From a combination of (A-4)–(A-6), (A-10), (A-11) and (A-13) the compatibility relation

$$\nabla^4 F = E(w_{xy}^2 - w_{xx}w_{yy} - w_{xx}/R) \tag{A-14}$$

follows. Thus an assumed deflection pattern is substituted into (A-14), the result for F, as well as w, is substituted into the sum of (A-7), (A-8) and (A-9), and the resulting expression is minimized with respect to the w deflection coefficients.

Non-dimensionalization

The radial displacement is written as a series for w of the form

$$w = t\sum A_{ij}\cos(i\pi x/\lambda x)\cos(j\pi y/\lambda y) \tag{A-15}$$

while the stress function is given by

$$F = Et^2\sum F_{mn}\cos(m\pi x/\lambda x)\cos(n\pi y/\lambda y) - \sigma y^2/2 = Et^2F_p - \sigma y^2/2. \tag{A-16}$$

Enough terms are kept in the F series so that F is a particular solution of the compatibility equation (A-14).

Non-dimensionalization of the total potential energy with respect to Et^3L/π^3R, and integration over a unit area, yield

$$\begin{aligned}\pi^3R(W_e+W_b+W_\sigma)/Et^3L = \int_0^1\int_0^1\Bigg\{\left(\eta F_{p\bar{y}\bar{y}}+\eta\mu^2F_{p\bar{x}\bar{x}}-\pi^2\frac{\sigma R}{Et}\right)^2 \\ -2(1+\nu)\left[\left(\eta F_{p\bar{y}\bar{y}}-\pi^2\frac{\sigma R}{Et}\right)(\eta\mu^2F_{p\bar{x}\bar{x}})-(\eta\mu F_{p\bar{x}\bar{y}})^2\right]\Bigg\}\mathrm{d}\bar{x}\,\mathrm{d}\bar{y}+\frac{1}{12(1-\nu^2)} \\ \times\int_0^1\int_0^1\{(\eta\mu^2\bar{w}_{\bar{x}\bar{x}}+\eta\bar{w}_{\bar{y}\bar{y}})^2-2(1-\nu)(\mu^2\eta^2\bar{w}_{\bar{x}\bar{x}}\bar{w}_{\bar{y}\bar{y}}-\mu^2\eta^2\bar{w}_{\bar{x}\bar{y}}^2)\}\,\mathrm{d}\bar{x}\,\mathrm{d}\bar{y}-2\pi^4\left(\frac{\sigma R}{Et}\right)\left(\frac{\varepsilon R}{t}\right)\end{aligned} \tag{A-17}$$

where $\bar{x}$ and $\bar{y}$ indicate coordinates non-dimensionalized with respect to λ_x and λ_y.

Similarly, the compatibility equation (A-14) when non-dimensionalized becomes

$$\left(\mu^2\frac{\partial^2}{\partial\bar{x}^2}+\frac{\partial^2}{\partial\bar{y}^2}\right)^2F_p = \mu^2\left(\bar{w}_{\bar{x}\bar{y}}^2-\bar{w}_{\bar{x}\bar{x}}\bar{w}_{\bar{y}\bar{y}}-\frac{\pi^2}{\eta}\bar{w}_{\bar{x}\bar{x}}\right) \tag{A-18}$$

End shortening

A relationship between $\varepsilon R/t$ and $\sigma R/Et$ can be found from the definition of end shortening. Thus the number of unknowns in the problem may be reduced by one. We have

$$\varepsilon = -(1/L)\int_0^L u_x\,\mathrm{d}x \tag{A-19}$$

which becomes through use of (A-1), (A-4), (A-5) and (A-13)

$$\varepsilon = -(1/L)\int_0^L [(F_{yy} - \nu F_{xx})/E - w_x^2/2]\,dx. \tag{A-20}$$

Using expressions (A-15) and (A-16) for w and F, integrating and evaluating the resulting expression at $y = 0$, one obtains

$$(\varepsilon R/t) = (\sigma R/Et) + \tfrac{1}{8}\eta\mu^2\Big(\sum_{j\neq 0} i^2 A_{ij}^2 + 2\sum i^2 A_{i0}^2\Big). \tag{A-21}$$

Circumferential-displacement continuity

A_{00} may be found from the condition of continuity of v in the circumferential direction. The continuity requirement on v demands that the constant part of v_y be zero. From (A-2), (A-4), (A-5) and (A-13) one obtains

$$v_y = (F_{xx} - \nu F_{yy})/E + w/R - w_y^2/2. \tag{A-22}$$

Substitution of (A-15) and (A-16) into (A-22) and setting to zero the constant terms give

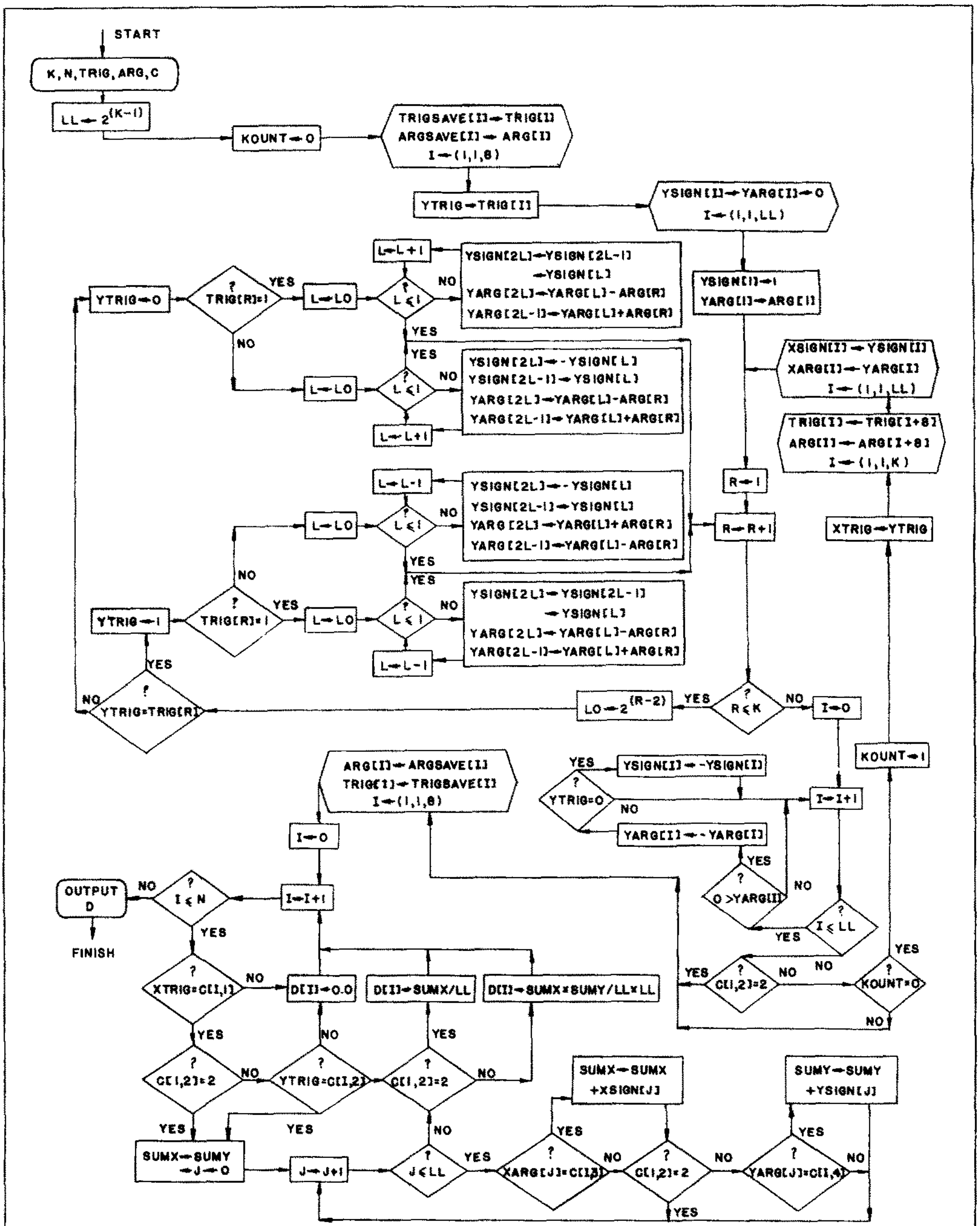
$$A_{00} = \tfrac{1}{8}\eta\Big(2\sum j^2 A_{0j}^2 + \sum_{i\neq 0} j^2 A_{ij}^2\Big) - \nu(\sigma R/Et). \tag{A-23}$$

Assumed deflection pattern

The post-buckling behavior was studied for three cases, as shown in (31a, b, c).

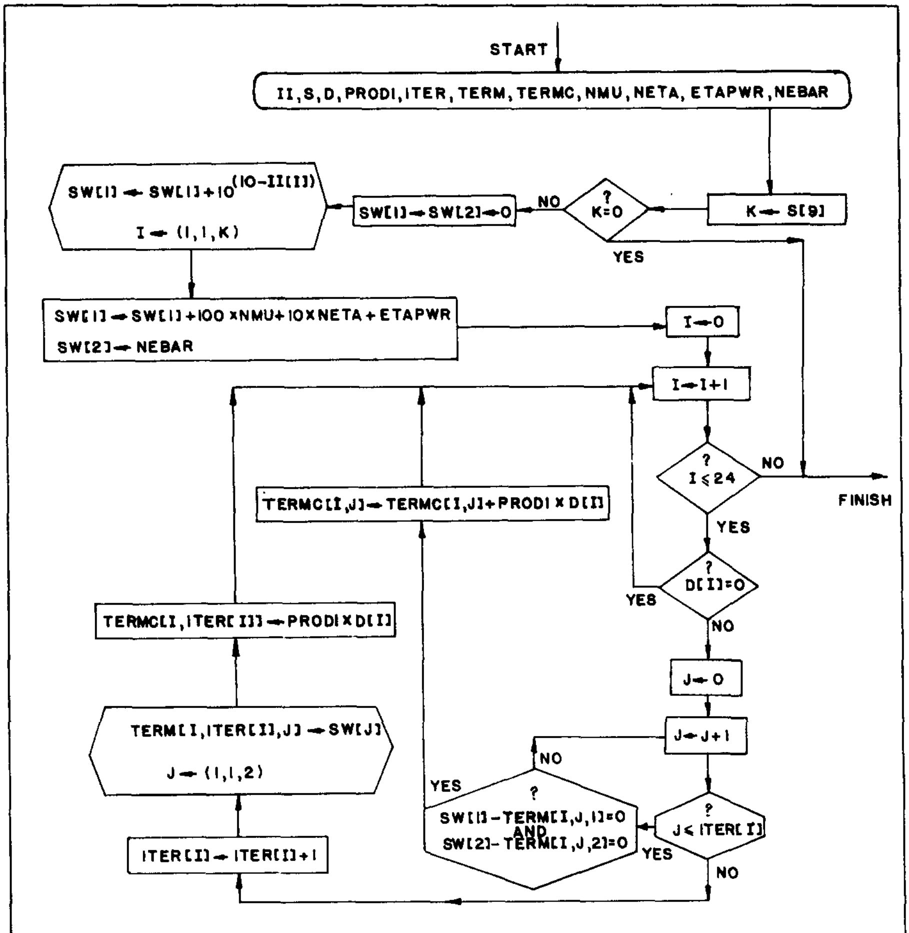
The equation of compatibility and the total potential energy in integer form may be found in Appendix B of [4].

APPENDIX B



APPENDIX BI— TRIGSPAND

FIG. 8



APPENDIX B2-SEARCHNSTORE

FIG. 9

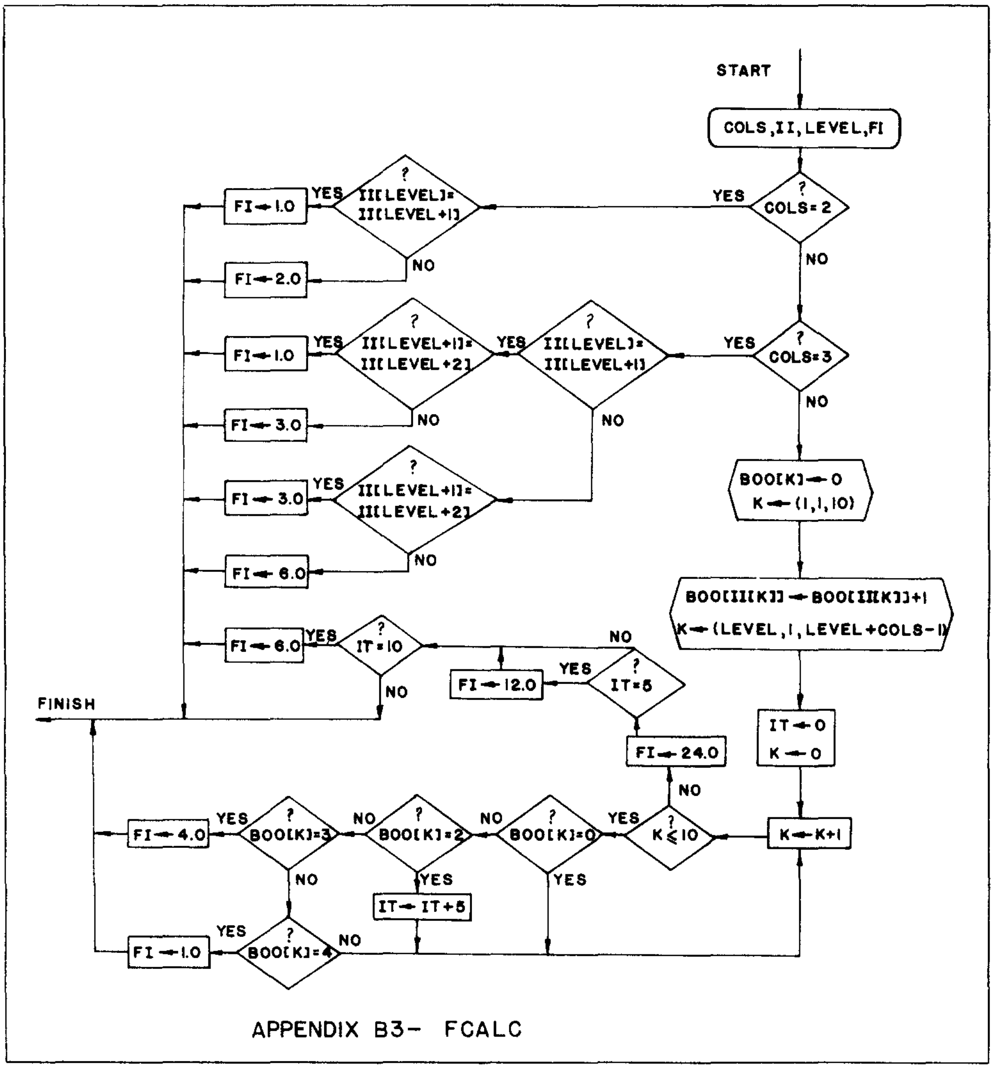

START
COLS, II, LEVEL, FI
? COLS = 2
YES
NO
? II[LEVEL] = II[LEVEL+1]
YES
NO
FI ← 1.0
FI ← 2.0
? COLS = 3
YES
NO
? II[LEVEL] = II[LEVEL+1]
? II[LEVEL+1] = II[LEVEL+2]
FI ← 1.0
FI ← 3.0
? II[LEVEL+1] = II[LEVEL+2]
FI ← 3.0
FI ← 6.0
BOO[K] ← 0
K ← (1,1,10)
BOO[II[K]] ← BOO[II[K]]+1
K ← (LEVEL, 1, LEVEL+COLS−1)
IT ← 0
K ← 0
K ← K+1
? K ≤ 10
FI ← 24.0
? IT = 5
FI ← 12.0
? IT = 10
FI ← 6.0
FINISH
? BOO[K] = 0
? BOO[K] = 2
IT ← IT+5
? BOO[K] = 3
FI ← 4.0
? BOO[K] = 4
FI ← 1.0
APPENDIX B3 – FCALC

FIG. 10

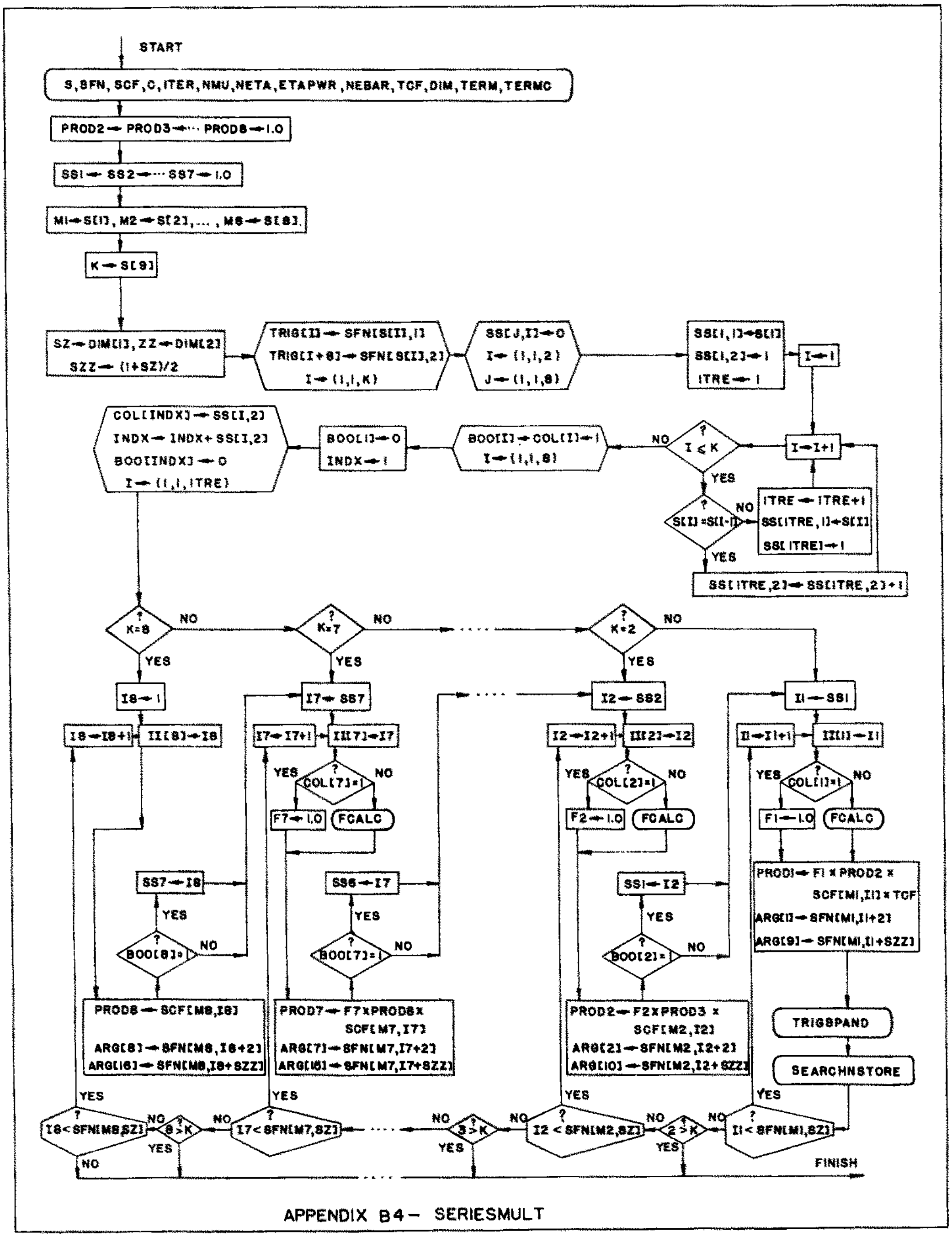
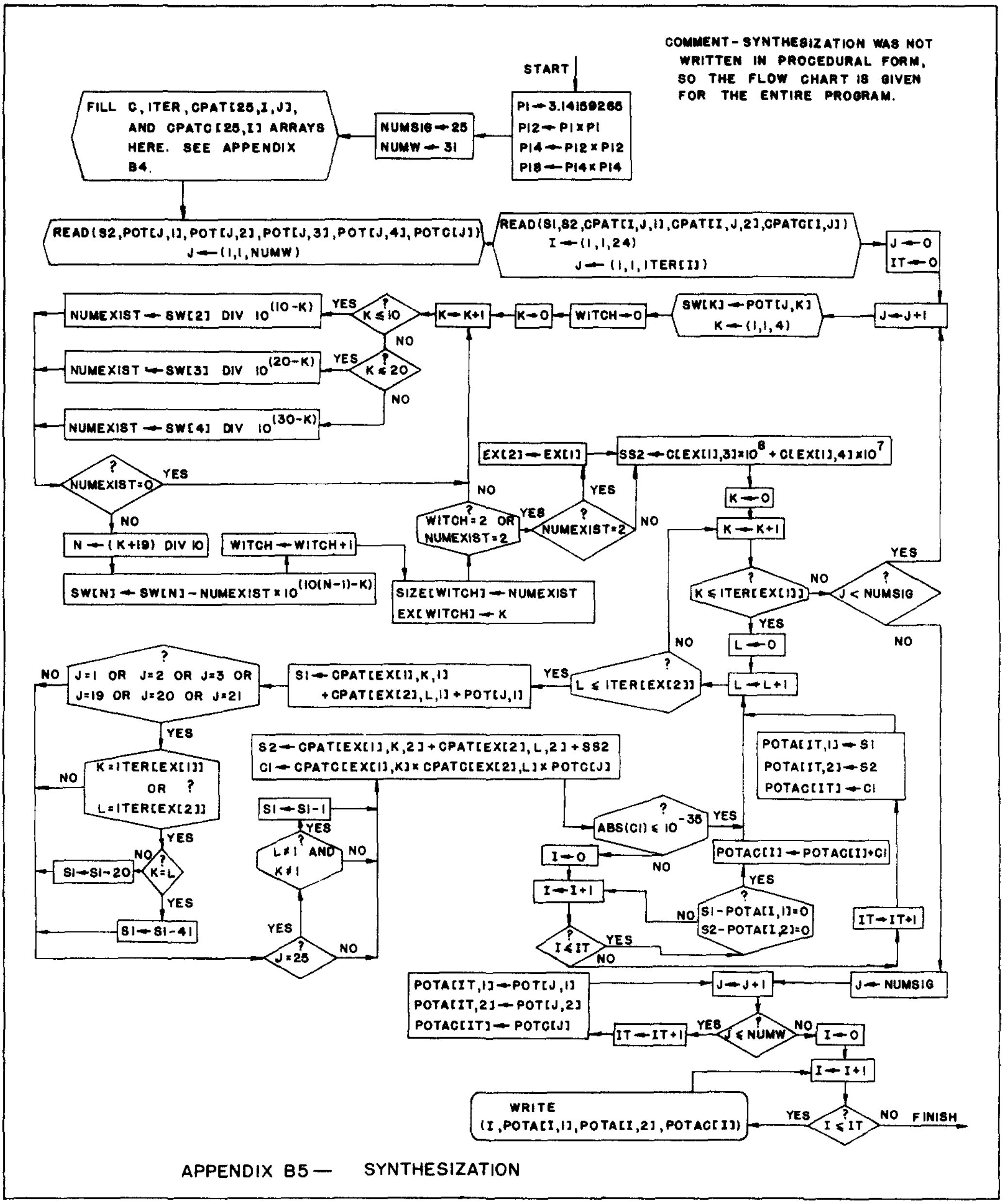
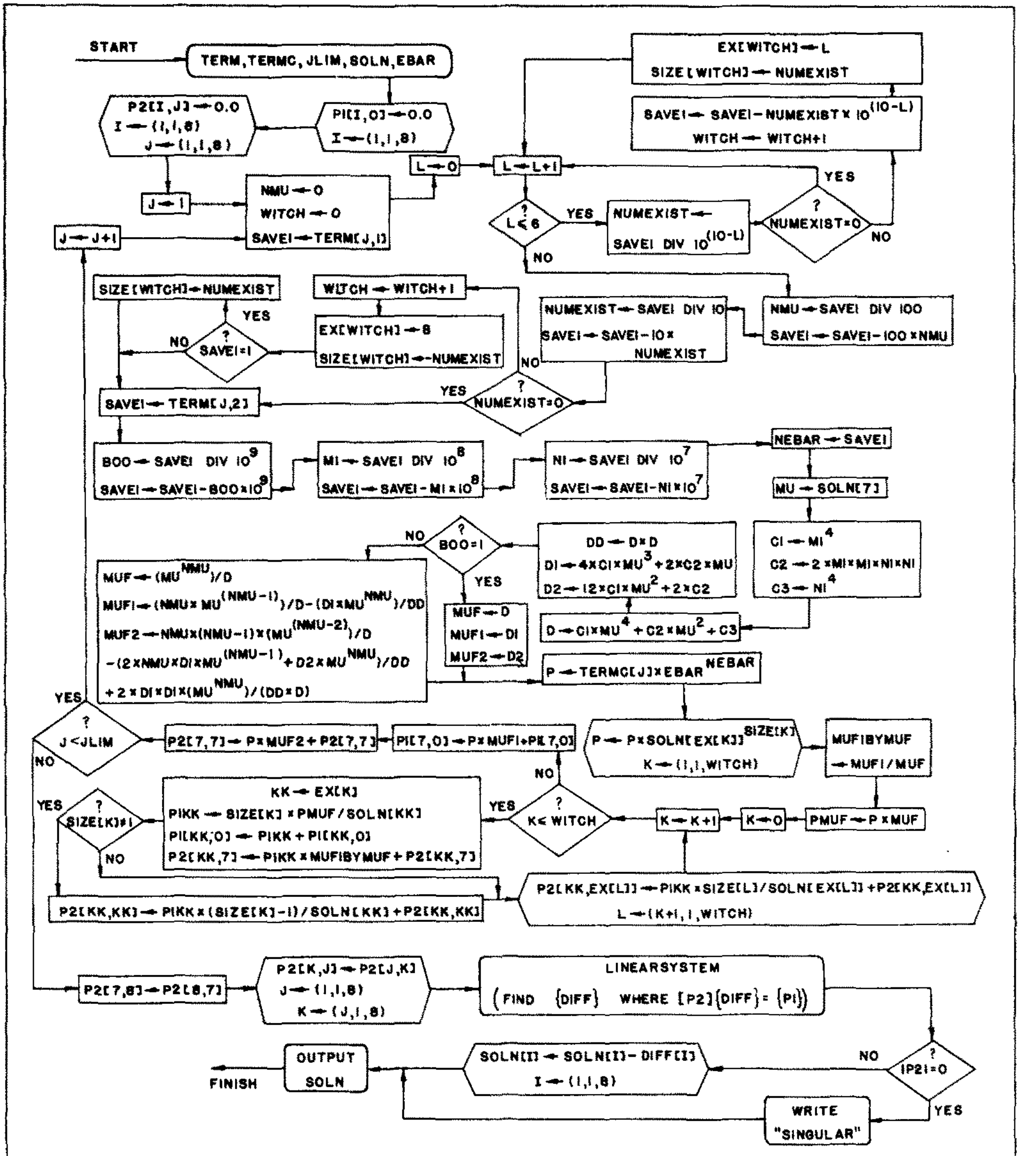

START
S, SFN, SCF, C, ITER, NMU, NETA, ETAPWR, NEBAR, TCF, DIM, TERM, TERMC
PROD2 ← PROD3 ←··· PROD8 ← 1.0
SS1 ← SS2 ←··· SS7 ← 1.0
M1 ← S[1], M2 ← S[2], ... , M8 ← S[8]
K ← S[9]
SZ ← DIM[1], ZZ ← DIM[2]
SZZ ← (1+SZ)/2
TRIG[I] ← SFN[S[I],1]
TRIG[I+8] ← SFN[S[I],2]
I ← (1,1,K)
SS[J,I] ← 0
I ← (1,1,2)
J ← (1,1,8)
SS[1,1] ← S[1]
SS[1,2] ← 1
ITRE ← 1
I ← 1
COL[INDX] ← SS[I,2]
INDX ← INDX + SS[I,2]
BOO[INDX] ← 0
I ← (1,1,ITRE)
BOO[I] ← 0
INDX ← 1
BOO[I] ← COL[I] ← 1
I ← (1,1,8)
I ≤ K
I ← I+1
S[I] = S[I-1]
ITRE ← ITRE+1
SS[ITRE,1] ← S[I]
SS[ITRE] ← 1
SS[ITRE,2] ← SS[ITRE,2]+1
K=8
K=7
K=2
I8 ← 1
I8 ← I8+1
II[8] ← I8
I7 ← SS7
I7 ← I7+1
II[7] ← I7
COL[7]=1
F7 ← 1.0
FCALC
I2 ← SS2
I2 ← I2+1
II[2] ← I2
COL[2]=1
F2 ← 1.0
I1 ← SS1
I1 ← I1+1
II[1] ← I1
COL[1]=1
F1 ← 1.0
SS7 ← I8
BOO[8]=1
SS6 ← I7
BOO[7]=1
SS1 ← I2
BOO[2]=1
PROD1 ← F1 × PROD2 × SCF[M1,I1] × TCF
ARG[1] ← SFN[M1,I1+2]
ARG[9] ← SFN[M1,I1+SZZ]
PROD8 ← SCF[M8,I8]
ARG[8] ← SFN[M8,I8+2]
ARG[16] ← SFN[M8,I8+SZZ]
PROD7 ← F7 × PROD8 × SCF[M7,I7]
ARG[7] ← SFN[M7,I7+2]
ARG[15] ← SFN[M7,I7+SZZ]
PROD2 ← F2 × PROD3 × SCF[M2,I2]
ARG[2] ← SFN[M2,I2+2]
ARG[10] ← SFN[M2,I2+SZZ]
TRIGSPAND
SEARCHNSTORE
I8 < SFN[M8,SZ]
8 > K
I7 < SFN[M7,SZ]
3 > K
I2 < SFN[M2,SZ]
2 > K
I1 < SFN[M1,SZ]
YES
NO
FINISH
APPENDIX B4 – SERIESMULT

FIG. 11



APPENDIX B5 — SYNTHESIZATION

FIG. 12



APPENDIX B6 — NEWTNRAPH

FIG. 13

(*Received* 20 *July* 1964)

Resumé—En mécanique appliquée, beaucoup de problèmes ne peuvent être résolus d'une manière exacte parcequ'ils sont définis par des équations differentielles non-linéaires. Dans ces cas on peut souvent obtenir une solution par une méthode où les inconnues sont représentées par séries trigonométriques, ou polynomiales dont les coefficients sont plus tard égalés. Ensuite on procède à la résolution d'équations non-linéaires algébriques.

Par un autre procédé on écrit une expression pour l'énergie potentielle du système qui est ensuite minimisée par rapport aux coefficients des séries.

Dans cet article des algorithmes sont présentés, conduisant à l'obtention d'équations à l'aide d'un calculateur digital. Ces équations sont enregistrées au moyen d'une représentation par nombres entiers. L'algorithme de Newton–Raphson est expliqué pour minimiser l'énergie potentielle représentée par des nombres entiers.

La méthode est appliquée à la résolution des grandes déformations après flambage de coques minces cylindriques circulaires soumises à une compression axiale.

Le temps nécessaire pour le calcul est court; grâce à l'aide d'un calculateur Burroughs B5000 l'expression représentant l'énergie potentielle étant déduite en deux minutes environ, et la partie principale de la courbe charge—raccourcissement étant obtenue en dix minutes.

Абстракт—Многие проблемы прикладной механики не поддаются точному решению из-за нелинейного характера описательных дифференциальных уравнений. В таких случаях решение нередко получается путем принятия соответствующих тригонометрических или степенных рядов, и разложения и собирания коэффициентов подобных тригонометрических членов или степеней переменной величины. Затем набор нелинейных алгебраических уравнений, полученный таким образом, употребляется для получения решения.

Другое решение многих проблем такого характера состоит из разработки потенциального выражения и его минимизирования по отнтшению к коэффициентам ряда.

В этой работе представлены алгорифмы для выведения уравнений при помощи цифровой вычислительной машины. Уравнения тогда запоминаются через целочисленное представление. Алгорифм Ньютона–Рафсона для минимизирования потенциала целое число-форма тоже представлен. Метод иллюстрирован выведением снова решения для поведения после продольного изгиба тонко-стенных кругло-цилиндрических оболочек под осевым сжатием.

Расчет занимает немного времени; употребляя вычислитель Борроус B5000 все потенциальное выражение было выведено приблизительно в две минуты; а главная устойчивая часть кривой нагрузка-сокращение была найдена в десять минут.