Computers & Structures Vol. 46, No. 2, pp. 331–338, 1993
Printed in Great Britain.

0045-7949/93 $6.00 + 0.00

A PRACTICAL AND EFFICIENT VECTORIAL APPROACH TO CONSIDER LOADS ON GUYED-TOWERS' CABLE ELEMENTS

M. BRUNEAU†

Civil Engineering Department, 161 Louis Pasteur, University of Ottawa, Ottawa, Ontario K1N 6N5, Canada

(*Received* 8 *October* 1991)

Abstract—An elegant vectorial approach to consider arbitrary combinations of wind/ice/self-weight loadings on cables is formulated from a synthesis of the existing knowledge. Both the theory and practical considerations are discussed. The simplicity of the method and its implementation are respectively illustrated by an example and pertinent excerpts of source code taken from a commercially available program.

1. INTRODUCTION

Guyed-towers, although visually of a great simplicity, are rather complex to analyse. While crude analytical assumptions led to adequate design in earlier days [1], it was quickly realized that the use of computers could not only accelerate the manual analyses but also lead to more reliable designs by enabling the engineer to consider more complex and realistic models of structural behaviour. Nowadays, numerous computer programs, 'tailor-made' to expediently analyse these structures, are available to practising engineers and manual calculations have effectively disappeared, leaving more time to address design-related issues and other aspects of the problem. The majority of these programs model the mast as a beam supported by elastic springs whose stiffnesses are adjusted to account for the cables' non-linearity through an iteration process.

There has been, however, in recent years, a desire in the industry to enhance the analytical capabilities of the existing programs to be able to analyse guyed-structures of any configuration in a fully integrated geometrically non-linear framework, thus allowing to overcome some limitations of the existing software and minimize the need for engineering simplifications and/or manual post-processing. Researchers have demonstrated the viability of large-displacements formulations [2–4], and some of the on-going efforts invested to implement the aforementioned capabilities within a production environment have already been reported [5, 6]. The author, being involved in such development work, has had the opportunity to review source code from a number of guyed-tower analysis programs, including some of a recent vintage, and has, in many instances, found that many structural engineering and programming aspects still follow obsolete, inefficient or too restrictive practices. The method used to consider arbitrary combinations of wind/ice/self-weight loadings on guys is one such case. Non-intuitive geometric conventions and 'congested' trigonometric operations are widespread in existing code. This paper proposes a more elegant vectorial approach to this problem that simultaneously minimizes in-core storage requirements.

The introduction of vectorial algebra in structural engineering programs is far from new. Among earlier examples, Wilson [7] introduced vectorial notions in a general-purpose finite-element program for the identification of beams' weak and strong axes of bending, thus greatly simplifying the cumbersome previous practice. Similarly, some of the concepts of a vectorial approach for loads on guys have been formulated previously [8]; however, a comprehensive explanation of this method and how it can be efficiently implemented is not available. This may explain why it remains unknown to many engineers currently involved in developing the next generation of guyed-towers analysis software. This paper hopes to remediate this situation; both a theory for the aforementioned vectorial strategy as well as practical considerations are discussed. Finally, the simplicity of the method and its implementation are respectively illustrated by an example and pertinent excerpts of source code taken from a commercially available program [5].

2. DESCRIPTION OF VECTORIAL APPROACH TO LOADS ON CABLES

The method is applicable to any guy arbitrarily oriented in a 3-D cartesian space and exposed to

† Special Consultant for Morrison Hershfield Ltd, 1980 Merivale Road, Nepean, Ontario K2G 1G4, Canada.

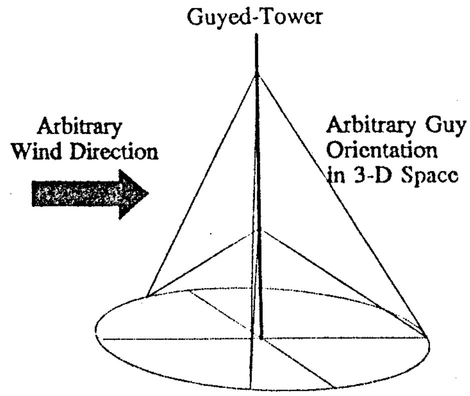

Fig. 1. Arbitrary wind and cable directions in a 3-D cartesian space.

wind, as shown in Fig. 1. Traditionally, the wind-input to guyed-tower analysis programs is expressed as a function of azimuth angles form a north direction, but conversion by a pre-processor program to a cartesian system is a trivial task. It is also normally assumed in guyed-tower analysis that only the component of wind normal to the cable loads it, the drag coefficient for the wind component parallel to the guy being virtually zero. A further simplification is obtained by considering the acting wind component perpendicular to the *chord* between the guy's attachment points, as opposed to the actual cable profile; this is deemed sufficiently accurate in practice as cables are usually pre-tensioned to high values, thus producing little sagging. Nonetheless, if cables with excessively large sag were to be analysed, improvements in accuracy could be obtained simply by subdividing each cable into many shorter cable-elements for which the sum of the chords would closely approximate the actual sagging profile; for actual guyed-towers, this refinement is rarely necessary.

Initially, the magnitude and orientation of only two vectors are known: that of the blowing wind (W) and the one defined by connecting ends 'i' and 'j' (i.e. the chord) of the cable under consideration (C). In the plane defined by these two vectors, the orientation of the vector perpendicular to that chord is simply obtained by

$$(C \times W) \times C = W_P \tag{1}$$

as illustrated in Fig. 2.

The magnitude of that wind force perpendicular to the cable can be obtained from the well-known expression for the wind-induced drag force on a structural element

$$F_{\text{drag}} = 0.5 C_D \rho V^2 A, \tag{2}$$

where C_D is the drag coefficient, ρ is the air density, V the wind velocity, and A the structure's projection in the plane perpendicular to the wind direction. However, some design standards use slightly modified forms of this equation. For example, the Canadian standard CAN-CSA-S37-M86 'Antennas, Towers and Antenna-supporting Structures' recommends

$$F_{\text{drag}} = 0.5 p C_D A, \tag{3}$$

where

$$p = q C_e C_g C_a,$$

where

$$q\ (\text{kPa}) = 50 \times 10^{-6} V^2\ (\text{km/hr}), \tag{4}$$

C_e, C_g and C_a are coefficients taking into account elevation, gust and acceleration effects, respectively, and q is the reference velocity pressure normally obtained from probabilistic contour maps or site-specifically from specialized public agencies [9].

The magnitude of the wind drag force acting perpendicular to the guy's chord can then be obtained by substituting, in eqns (3), the drag coefficient C_D for a normal round cross-section, and correspondingly the component of wind-velocity perpendicular to the guy (V_P), i.e.

$$V_P^2 = (V \sin \alpha)^2 = V^2 \sin^2 \alpha = V^2(1 - \cos^2 \alpha), \tag{5}$$

where the value of $\cos(\alpha)$ can be calculated by the dot-product of unit vectors parallel to W and C, respectively. While the rigour of this derivation can be debated, experimental results have demonstrated this to be a valid and realistic model for guy cables [10].

The resulting magnitude and orientation of the total load-vector applied to the cable (V_{TOT}) is obtained by vectorial addition of the gravity load-vectors (ice and self-weight) and the above normal wind load one; this in turn dictates the direction of the cable sag and the plane within which lie all resulting tension end-forces.

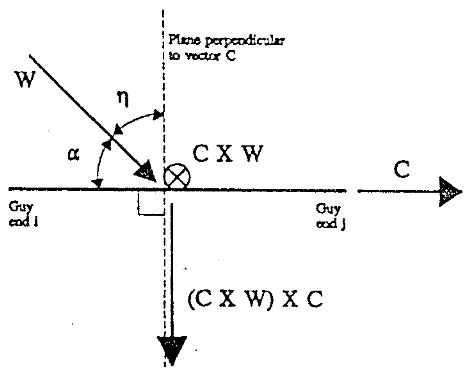

Fig. 2. Cross-product operations to obtain vector of wind perpendicular to cable.

Effectively, the analysis of the cable element can now proceed successfully in the new plane defined using any of the numerous 2-D cable element formulations available ([8, 11, 12], to name a few). For example, one such formulation defines the local y-axis (y) as being parallel and opposed in direction to the calculated V_{TOT}. In that case, the other two local axes (normalized vectors x and z) would be obtained by cross products while the vertical (l) and horizontal (h) local projections of the guy would be calculated by dot-products, i.e.

$$y = -V_{TOT} \tag{6}$$

$$z = C \times y \tag{7}$$

$$x = y \times z \tag{8}$$

$$l = x \cdot C \tag{9}$$

$$h = y \cdot C. \tag{10}$$

Furthermore, using the previously defined vectorial information, a transformation matrix can be assembled to obtain the cable stiffness matrix and end-forces in the global coordinate system from those calculated in the element's local coordinate system. Typically

$$F_{global} = TF_{local} \tag{11}$$

$$K_{global} = TK_{local}T^T, \tag{12}$$

where T is the transformation matrix, F and K are the vector of end-forces and stiffness matrix, respectively, in the local or global coordinate system as subscripted. The specifics of that transformation matrix are presented in the following section.

3. EFFICIENT IN-CORE MEMORY UTILIZATION AND SOURCE CODE

Clearly, the use of a cross-product subroutine will minimize and make more legible the coding. One such subroutine (CROSS) is presented in Appendix A along with another subroutine (VECTOR) that returns the normalized vector and distance between two points; these are slightly adapted from a public-domain structural analysis program [7]. It is noteworthy that the vectorial operations mostly require additions and multiplications, operations quickly performed by computers as compared to trigonometric functions.

Within the main program, a single 4 × 4 matrix (VC) is defined in which intermediate and final vectors resulting from the above cross- and dot-products are stored column-wise; as no other temporary matrices are needed, memory usage is optimized. The VC matrix is structured such that, for any vector, the entries on the first three rows are the normalized X, Y and Z cartesian coordinates of that vector, while length is kept on the fourth row. Both the VECTOR and CROSS subroutines are designed to comply with this convention. Furthermore, the deliberate column-wise storage of vectors in VC is also intended to simplify interaction with VECTOR and CROSS; while the latter are designed to operate on vectors, the address VC(1, k) of the vector in column k, in accordance with FORTRAN conventions for reading matrices (i.e. column-wise), is sufficient information to extract that vector from VC and pass it to the subroutine, eliminating the need for intermediate manipulation of vectors.

The calculation of the effective resulting total uniformly distributed load on the cable and, consequently, the definition of the local coordinate system can be contained within a single subroutine. One such routine, CLOAD, is proposed in Appendix B; many comments describing the procedures are already embedded in the code. Yet, a few practical aspects deserve some additional consideration and are discussed in the following.

	Col.1	Col.2	Col.3	Col.4
Row 1	W			C
Row 2	Wind	-	-	Chord
Row 3	Vector			Vector
Row 4				

(a)

	Col.1	Col.2	Col.3	Col.4
Row 1	W	W_P	C × W	C
Row 2	Wind	⊥Wind	Interim	Chord
Row 3	Vector	Vector		Vector
Row 4				

(b)

	Col.1	Col.2	Col.3	Col.4
Row 1	W	V_{TOT}	C × W	C
Row 2	Wind	Total	Interim	Chord
Row 3	Vector	Load		Vector
Row 4		Vector		

(c)

	Col.1	Col.2	Col.3	Col.4
Row 1	x =	y =	z =	C
Row 2	y × z	$-V_{TOT}$	C × y	Chord
Row 3	local	local	local	Vector
Row 4	axis	axis	axis	

(d)

Fig. 3. Evolutionary storage disposition within the matrix VC.

At the onset of this routine, the chord-vector (C) has been calculated by the VECTOR subroutine from the end coordinates of the guy, and the normalized self-weight vector is also available (SW). The status of matrix VC following the normalization of the user-inputted wind vector is shown in Fig. 3(a). The vector of the wind load perpendicular to the cable, as per the equations previously defined, is then calculated and stored as shown in Fig. 3(b). The magnitude of that vector is most conveniently determined separately; the cable's diameter, as incremented by the thickness of ice coating specified for the load case under consideration, must be used for drag force calculations. Further to the vectorial addition of the ice/self-weight and wind vectors (Fig. 3c), the local coordinate system can be defined (Fig. 3d) as well as the horizontal and vertical projections of the guy cable for that load case (stored separately).

```
                  DO 500  I=1,3
                    GRS(I)    = VC(I,1) * RS(1) + VC(I,2) * RS(2)
                    GRS(I+3)  = VC(I,1) * RS(3) + VC(I,2) * RS(4)
          500     CONTINUE
```

Once the local coordinate system has been established and stored within VC, the aforementioned transformation matrix T is directly obtained from a 3×3 subset of the VC matrix. Explicitly

$$T = \begin{bmatrix} T_S & 0 \\ 0 & T_S \end{bmatrix},$$

where

$$T_S = \begin{bmatrix} x_X & y_X & z_X \\ x_Y & y_Y & z_Y \\ x_Z & y_Z & z_Z \end{bmatrix}, \tag{13}$$

where x, y and z are the unit vectors of that local coordinate system expressed in the X, Y and Z global cartesian coordinate system. The matrix T_S need not be assembled as it is already contained within VC. Furthermore, recognizing the absence of end-forces in the local z-direction, only a 3×2 subset of T_S is needed as shown below. As the vector of end-forces at ends i and j of the guy are usually returned from the cable-element analysis subroutine in the following form, or variation thereof

$$RS = \begin{bmatrix} RS(1) = F_{xi} \\ RS(2) = F_{yi} \\ RS(3) = F_{xj} \\ RS(4) = F_{yj} \\ RS(5) = T_i \\ RS(6) = T_i \\ RS(6) = T_j \end{bmatrix} \tag{14}$$

the transformation of eqn (11) can directly be accomplished by

Similarly, realizing that a satisfactory 4×4 tangent stiffness matrix in the local coordinate system can be constructed from the 2×2 module (K_1) by

$$K_{\text{local}} = \begin{bmatrix} K_1 & -K_1 \\ -K_1 & K_1 \end{bmatrix} \tag{15}$$

the complete 6×6 stiffness matrix can be easily constructed by breaking down the transformation process into smaller sub-matrix operations, and using only the necessary 3×2 subset of T_S

$$\begin{aligned} K_{\text{global}} &= T K_{\text{local}} T^T \\ &= \begin{bmatrix} T_S & 0 \\ 0 & T_S \end{bmatrix} \begin{bmatrix} K_1 & -K_1 \\ -K_1 & K_1 \end{bmatrix} \begin{bmatrix} T_S^T & 0 \\ 0 & T_S^T \end{bmatrix} \\ &= \begin{bmatrix} T_S K_1 T_S^T & -T_S K_1 T_S^T \\ -T_S K_1 T_S^T & T_S K_1 T_S^T \end{bmatrix}. \end{aligned} \tag{16}$$

Therefore, further to the calculation of that 2×2 stiffness submatrix (normally by another subroutine), the following code can rapidly perform the necessary transformation to obtain the 6×6 stiffness matrix in the global coordinate system

```
                  DO 800 I=1,2
                    DO 800 J=1,3
          800         ST(I,J)=S(I,1)*VC(J,1) + S(I,2)*VC(J,2)
      C
                  DO 850 I=1,3
                    DO 850 J=1,3
                      S(I,J)     = VC(I,1)*ST(1,J)+VC(I,2)*ST(2,J)
                      S(J,I)     = S(I,J)
                      S(I+3,J+3) = S(I,J)
                      S(J+3,I+3) = S(I,J)
                      S(I+3,J)   = -S(I,J)
                      S(J,I+3)   = -S(I,J)
                      S(I,J+3)   = -S(I,J)
                      S(J+3,I)   = -S(I,J)
          850     CONTINUE
```

	Col.1	Col.2	Col.3	Col.4
Row 1	1			.3536
Row 2	0	-	-	.7071
Row 3	0			.6124
Row 4	1			14.14

(a)

	Col.1	Col.2	Col.3	Col.4
Row 1	1	.9354	0	.3536
Row 2	0	-.2673	.6547	.7071
Row 3	0	-.2314	-.7560	.6123
Row 4	1	1*	.9354	14.14

* Magnitude calculated seperately

(b)

	Col.1	Col.2	Col.3	Col.4
Row 1	1	.9354	0	.3536
Row 2	0	-.2673	.6547	.7071
Row 3	0	-.2314	-.7560	.6123
Row 4	1	4.20	.9354	14.14

(c)

	Col.1	Col.2	Col.3	Col.4
Row 1	.3536	-.9354	0	.3536
Row 2	.7071	.2673	-.6547	.7071
Row 3	.6124	.2314	.7559	.6123
Row 4	1	1	1	14.14

(d)

Fig. 4. Values in VC for a numerical example.

It is noteworthy that the submatrix T_S need not be physically transposed; inverted addressing performs the equivalent task. Finally, the simple procedures described herein are to be repeated for each cable present in the structure under consideration, as well as for all load cases since the definition of the local coordinate system, here, is loading-dependent.

4. EXAMPLE

Results from a short numerical example are presented to illustrate the above concepts, while providing an opportunity for developers to verify the accuracy of their results. Here, the ends i and j of a cable are arbitrarily located at (0, 0, 0) and (5, 10, 8.66) ft, respectively, in (X, Y, Z) global cartesian coordinates. The wind acting on the cable is blowing in the (1, 0, 0) direction. The unstressed length of that cable is 14.5 ft. Without loss of generality, and to keep the problem simple, ice and gravity loads are neglected in this example. The cable diameter is 0.1 ft, and the value of its area times the modulus of elasticity is assumed to 3,000,000 lb/ft^2. ... drag and gust coefficients are taken as 1.2 and ... respectively, all other coefficients being 1.0. The design reference velocity pressure is selected to be 20 lb/ft^2.

The numerical values corresponding to the vectors presented in Fig. 3(a–d) are correspondingly tabulated in Fig. 4(a–d).

The design pressure p is 40 psf, while the design total uniformly distributed load perpendicular to the chord becomes 4.2 lb/ft. The horizontal and vertical projections of the guy in the local coordinate system are 14.14 and 0.0 ft, respectively, the latter being null as logically anticipated in the absence of gravity-loads. End-forces in local and global coordinates are respectively

$$RS = \begin{bmatrix} RS(1) = -76.08 \\ RS(2) = 30.44 \\ RS(3) = 76.08 \\ RS(4) = 30.44 \\ RS(5) = 81.94 \\ RS(6) = 81.94 \end{bmatrix},$$

$$GRS = \begin{bmatrix} GRS(1) = -55.38 \\ GRS(2) = -45.66 \\ GRS(3) = -39.54 \\ GRS(4) = -1.58 \\ GRS(5) = 61.93 \\ GRS(6) = 53.64 \end{bmatrix}. \quad (17)$$

Finally, the numerical values for both the 2×2 stiffness submatrix K_1 in local coordinates and the global 3×3 $T_S K_1 T_S^T$ stiffness submatrix in global coordinates are

$$K_1 = \begin{bmatrix} 111.9 & 0.005114 \\ 0.005114 & 5.65 \end{bmatrix},$$

$$T_S K_1 T_S^T = \begin{bmatrix} 18.93 & 26.55 & 23.00 \\ 26.55 & 56.33 & 48.79 \\ 23.00 & 48.79 & 42.25 \end{bmatrix}. \quad (18)$$

It is noteworthy that a large sag occurs in this example, and that much accuracy is to be gained by modelling the cable by smaller subelements.

5. CONCLUSIONS

A practical and efficient vectorial approach to consider loads on guyed-towers' cable elements has been formulated from a synthesis of the existing knowledge. Sufficiently detailed practical information has been provided to facilitate the integration of the proposed method into newer guyed-towers analysis programs.

Acknowledgement—The samples of source code have been taken from a commercial Tower Analysis Program (TAP) whose structural engineering analysis modules have been developed by the author. However, TAP remains a property of Morrison Hershfield Ltd, Toronto, Canada, and the author is grateful to the later for permission to release part of source code for publication purposes. The findings and conclusions of this paper are, however, those of the author alone.

REFERENCES

1. E. Cohen, Design of multi-level guyed towers: structural analysis. *J. Struct. Div., ASCE* **83**, 1356-1–1356-29 (1957).
2. A. H. Peyrot, *SAPS User's Manual—Version 88.* Civil Engineering Department, University of Wisconsin, Madison (1988).
3. N. V. Raman, G. V. Surya Kuman and V. V. Sreedhara Rao, Large displacement analysis of guyed towers. *Comput. Struct.* **28**, 93–104 (1988).
4. S. G. Ekhande and K. S. Madugula, Geometric non-linear analysis of three-dimensional guyed towers. *Comput. Struct.* **29**, 801–806 (1988).
5. M. Bruneau, T.A.P. (Tower Analysis Program)—a non-geometrically non-linear stiffness analysis program for the analysis of guyed towers. Documentation, Underlying Theory and User's Guide, Morrison Hershfield Ltd report (1989).
6. M. G. Nielsen, Simplified dynamic analysis of guyed masts. *IASS 15th Meeting of the Working Group on Masts and Towers*, Stockholm (1991).
7. E. L. Wilson, CAL 86—computer assisted learning of structural analysis and the CAL/SAP development system. Structural Engineering, Mechanics and Materials Report SEMM-86-05, Civil Engineering Department, University of California, Berkeley (1986).
8. R. I. S. Miller, Analysis of cable structures by Newton's method. Thesis submitted in partial fulfilment of the requirements for the degree of Master of Applied Science, Department of Civil Engineering, University of British Columbia (1971).
9. M. H. Magued, M. Bruneau and R. B. Dryburgh, Evolution of design standards and recorded failures of guyed towers in Canada. *Canadian J. Civil Engng* **16**, 725–732 (1989).
10. M. J. Casarella and M. Parsons, Cable systems under hydrodynamic loadings. *MTS Jnl* **4**, 27–44 (1971).
11. M. Irvine, *Cable Structures*. MIT Press, Cambridge, MA (1981).
12. A. H. Peyrot and A. M. Goulois, Analysis of cable structures. *Comput. Struct.* **10**, 805–813 (1979).

APPENDIX A

```
      SUBROUTINE CROSS(A,B,C)
C---------------------------------------------------------------------
C     THIS ROUTINE PERFORM THE VECTOR CROSS PRODUCT    A X B = C
C---------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(4),B(4),C(4)
C--------------------------------------------------
      X = A(2)*B(3) - A(3)*B(2)
      Y = A(3)*B(1) - A(1)*B(3)
      Z = A(1)*B(2) - A(2)*B(1)
      C(4) = DSQRT( X*X + Y*Y + Z*Z )
      IF(C(4).GT.0.0) GO TO 100
      WRITE (*,2000)
      STOP
C
C----RETURN THE AXIAL COMPONENTS OF A UNIT LENGTH VECTOR
C      AND THE RESULTING LENGTH FROM THE CROSS PRODUCT IN C(4)
C
  100 C(3) = Z/C(4)
      C(2) = Y/C(4)
      C(1) = X/C(4)
      RETURN
C
 2000 FORMAT (' ** COINCIDENT VECTORS FOR CROSS PRODUCT - ERROR **'/)
      END
C---------------------------------------------------------------------
C---------------------------------------------------------------------
C
      SUBROUTINE VECTOR (V,XI,YI,ZI,XJ,YJ,ZJ)
C---------------------------------------------------------------------
C     THIS ROUTINE CALCULATE THE X, Y AND Z LENGTH PROJECTION
C       OF VECTOR V, AND RETURN UNIT COORDINATES IN V(1) TO V(3)
C       AND THE ACTUAL LENGTH IN V(4)
C---------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION V(4)
C
      X = XJ - XI
      Y = YJ - YI
      Z = ZJ - ZI
      XX  = X*X + Y*Y + Z*Z
C
```

```
      IF(XX.GT.0.0) GO TO 100
      WRITE (*,2000)
      STOP
C
  100 V(4) = DSQRT(XX)
      V(3) = Z/V(4)
      V(2) = Y/V(4)
      V(1) = X/V(4)
C
      RETURN
 2000 FORMAT (' ZERO LENGTH VECTOR - ERROR')
      END
C----------------------------------------------------------------------
```

APPENDIX B

```
C----------------------------------------------------------------------
      SUBROUTINE CLOAD (VC,SW,FLOAD,NLCCUR,CD,CE,CG,CA,DIA,WTOT,DHORIZ,
     .                  DVERT,PROICE,NUMLCT)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION VC(4,4),SW(4),FLOAD(12,NUMLCT)
C----------------------------------------------------------------------
C -- NLCCUR = current load case
C    FLOAD  = matrix of load-case information
C    Q      = reference velocity pressure
C    VWX    = X component of wind direction
C    DELT   = change in temperature
C    THKICE = radial ice thickness
C    DENICE = ice density
C    VC     = temporary vector storage area and transformation matrix
C    PROICE = percentage of ice thickness on that particular cable
C    CD, CE, CG, CA = drag, exposure, gust, acceleration coefficients
C    WTOT   = total load
C    DHORIZ = Horizontal projection of cable in local coordinates
C    DVERT  = Vertical projection of cable in local coordinates
C----------------------------------------------------------------------
C  -- THIS SUBROUTINE DEFINES LOCAL COORDINATE SYSTEM OF CABLE
C     FOR A SPECIFIED LOAD CONDITION.  TOTAL LOAD ON CABLE ALSO CALC.
C     VC(I,1), VC(I,2), AND VC(I,3) USED TO DEFINE THE TEMP. VECT.
C     AT THE END OF THIS ROUTINE:
C     X IN VC(I,1), Y IN VC(I,2), AND Z IN VC(I,3) (C STILL IN VC(I,4))
C     SELF-WEIGHT LOCAL HORIZ. AND VERT. PROJECTION DEFINED
C     BY DOT PRODUCT OF C (ACTUAL LENGTH) OVER X AND Y (UNIT LENGTHS)
C
      ZERO    = 0.0D0
      PI      = 3.14159265
      Q       = FLOAD(1,NLCCUR)
      VWX     = FLOAD(2,NLCCUR)
      VWY     = FLOAD(3,NLCCUR)
      VWZ     = FLOAD(4,NLCCUR)
      DELT    = FLOAD(5,NLCCUR)
      THKICE  = FLOAD(6,NLCCUR)
      DENICE  = FLOAD(7,NLCCUR)
C
C---ADDITIONAL OPERATIONS TO BE EXECUTED AT FIRST CYCLE ONLY
C   CAN BE INSERTED HERE, DEPENDING ON THE PROGRAM STRUCTURE
C
C---- SELF WEIGHT VECTOR SW(4) PROVIDED BY OTHER ROUTINE
C
      DEFF = DIA + (2.*THKICE) * PROICE
C
C----GET WIND DIRECTION UNIT VECTOR VC(1,1)
C----IF NO WIND REFERENCE PRESSURE SPECIFIED, BYPASS FOLLOWING
C
      IF(Q.EQ.0.0) THEN
        WWIND = 0.0
        VC(1,2) = 0.0
        VC(2,2) = 0.0
        VC(3,2) = 0.0
        GO TO 100
      ENDIF
      CALL VECTOR(VC(1,1),ZERO,ZERO,ZERO,VWX,VWY,VWZ)
```

```
C
C---- CALCULATE PRESSURE AS PER CAN3-S37-M86
C
      PRESUR = Q*CE*CG*CA
C
C----GET ACTING DIRECTION OF WIND PRESSURE : (C X W) IN V(I,3)
C                                     AND: (C X W) X C IN V(I,2)
C
      CALL CROSS (VC(1,4),VC(1,1),VC(1,3))
      CALL CROSS (VC(1,3),VC(1,4),VC(1,2))
C
C----DIRECTION OF PERPENDICULAR WIND UDL IS IN VC(I,2)
C    CHORD DIRECTION VECTOR IS IN VC(I,4)
C    ORIGINAL WIND DIRECTION IS IN VC(I,1)
C    WEFF TO CONTAIN SIN*SIN OF ANGLE BETWEEN CHORD AND WIND DIR.
C
      COSANG = (VC(1,1)*VC(1,4) + VC(2,1)*VC(2,4) + VC(3,1)*VC(3,4))
      WEFF = 1. - COSANG*COSANG
C
C----DIAMETER OF CABLE MUST INCLUDE % OF ICE THICKNESS
C
      WWIND = PRESUR * DEFF * CD * WEFF
C
C----GRAVITY VECTORIAL EFFECT MUST BE CONSIDERED (SELF WEIGHT + ICE)
C
  100 CONTINUE
      WICE = (DENICE * (DEFF*DEFF - DIA*DIA) * PI) / 4.
      WGRAV = WICE + SW(4)
C
      VCX = WGRAV * SW(1) + WWIND * VC(1,2)
      VCY = WGRAV * SW(2) + WWIND * VC(2,2)
      VCZ = WGRAV * SW(3) + WWIND * VC(3,2)
C
      CALL VECTOR(VC(1,2),ZERO,ZERO,ZERO,VCX,VCY,VCZ)
C
C----RESULTING UDL ON CABLE IN WTOT, NORMALIZED IN VC(I,2)
C     SET LOCAL AXES AS Y = -W DIRECTION, OTHERS BY CROSS PRODUCT
C       Z AXIS = C X Y     AND     X AXIS = Y X Z
C
      WTOT = VC(4,2)
      VC(1,2) = -VC(1,2)
      VC(2,2) = -VC(2,2)
      VC(3,2) = -VC(3,2)
      VC(4,2) = 1.0
C
      CALL CROSS (VC(1,4),VC(1,2),VC(1,3))
      CALL CROSS (VC(1,2),VC(1,3),VC(1,1))
C
C----NOW X AXIS IN VC(I,1), Y IN VC(I,2), AND Z IN VC(I,3)
C----DEFINE LOCAL HORIZ. AND VERT. PROJECTION (DHORIZ AND DVERT)
C     BY DOT PRODUCT OF C (ACTUAL LENGTH) OVER X AND Y (UNIT LENGTHS)
C
      DHORIZ = VC(4,4)*(VC(1,1)*VC(1,4)+VC(2,1)*VC(2,4)+VC(3,1)*VC(3,4))
      DVERT  = VC(4,4)*(VC(1,2)*VC(1,4)+VC(2,2)*VC(2,4)+VC(3,2)*VC(3,4))
C
      RETURN
      END
C----------------------------------------------------------------------
C----------------------------------------------------------------------
```