". . . although the derivation of the equation assumes a free-ended shaft, it is applicable to other support configurations."

A Computer Program to Determine the Lateral Critical Speeds of Flexible Rotors

by

Alexander P. Paran, S.M. and Willie C. Si*

ABSTRACT

This program determines by numerical methods the lateral critical speeds of a rotating shaft of circular cross section and of uniform density. The core of this program uses the method developed independently by Prohl and Myklestad as adapted by J.W. Lund. There is no limit to the number of critical speeds that the program can find, the only requirement being that the shaft in consideration be divided into a sufficient number of sections. The program can handle either English units or the SI and the operator has a choice of entering some values in units of weight or units of mass. Another feature of the program is the ability to plot the mode shapes of the rotor at the different critical speeds.

INTRODUCTION

An important part in the design of rotors is the calculation of its critical speeds. The critical speed of a rotating shaft is the speed at which the shaft starts to vibrate violently in a transverse direction. If this condition is allowed to persist, the amplitude of the vibration will build up to such a magnitude that rupture of the shaft may occur.

Classical mathematical methods for solving for the critical speeds have been formulated for beams of simple geometry (e.g., a simply supported homogenous beam of uniform cross-section). However, these methods become very tedious for more complex geometries and more so if the shaft has redundant supports. Consequently, alternate methods have been developed.

The Holzer method, which was originally devised for torsional vibrations, is a tabular method for the analysis of multi-mass lumped-parameter systems. In this method, a trial critical speed must be assumed, and after working across the shaft, a residual function must be determined. A remainder curve of this function may be plotted against the assumed speed to locate the speed where the function equals zero. Being a trial and error method, some of the difficulties of this scheme are in estimating the initial trial value of the critical speed and in selecting a second trial value if the initial trial value fails to satisfy the governing equations. Furthermore, the repetitive calculations become laborious when the shaft is divided into four or more stations.

* Department of Mechanical Engineering, University of the Philippines, Diliman, Quezon City.

The Prohl-Myklestad method is similar to the Holzer method. However, this procedure is more involved than the Holzer method, as the assumed critical speed must satisfy the four boundary conditions of bending moment, shear, slope, and deflection. For any assumed speed, the governing equations can be solved to satisfy three of the four boundary conditions. By plotting the fourth boundary condition against speed, the critical speed will occur when this remainder equals zero.

Both these methods can be programmed into a digital computer and, to be accurate, should be carried to at least five significant figures.

J.W. Lund, using the Prohl-Myklestad method, developed a procedure which makes the successive trial values of the critical speed converge to a true value. This is the method used in the computer program presented in this article.

NOMENCLATURE

x	coordinate along the axis of the shaft
y	radial shaft displacement in the x-y plane
θ	angular shaft displacement in the x-y plane
E	Young's modulus of elasticity
I	cross-sectional transverse moment of inertia of the shaft
L	length
F	force
V	shear force
M	bending moment
K	support stiffness coefficient
[A]	2 x 2 matrix of residual bending moments and shear forces
[B]	2 x 2 matrix of the derivatives of the residual bending moments and shear forces
β	determinant of [A], residual determinant
β_k	the determinant of the matrix [A] where the elements in column k have been replaced by the corresponding elements in column k of the matrix [B]
J	number of critical speeds found

Subscripts:

y	y-direction in the x-y plane
θ	slope; rotation in the x-y plane

n	rotor station number
N	number of the last rotor station
j, k	indices

ANALYSIS

As in the conventional Prohl-Myklestad method, the rotor is represented in the calculations by a series of lumped masses, called stations, which are connected by uniform shaft sections. Stations are provided at the two free ends of the rotor, at the bearing centerlines and at places where heavy components are mounted on the shaft such as wheels, impellers, or thrust collars. The shaft section between two stations is assumed to be uniform and its mass is lumped at the ends of the section at the stations. Since this method is essentially a finite element problem, there must be a sufficient number of stations to represent adequately the highest mode in the frequency range of interest.

Figure 1 schematically shows a divided portion of a rotor. The convention for the shear forces, bending moments, slopes, and deflections are shown. From mechanics and beam theory, the different quantities in Figure 1 are found to be:

$$F_{yn} = (-K_{yn})y_n \qquad (1)$$

$$M_{\theta n} = (-K_{\theta n})\theta_n \qquad (2)$$

$$V_n' = V_n + w^2 m_n y_n + F_{yn} \qquad (3)$$

$$M_n' = M_n - M_{\theta n} \qquad (4)$$

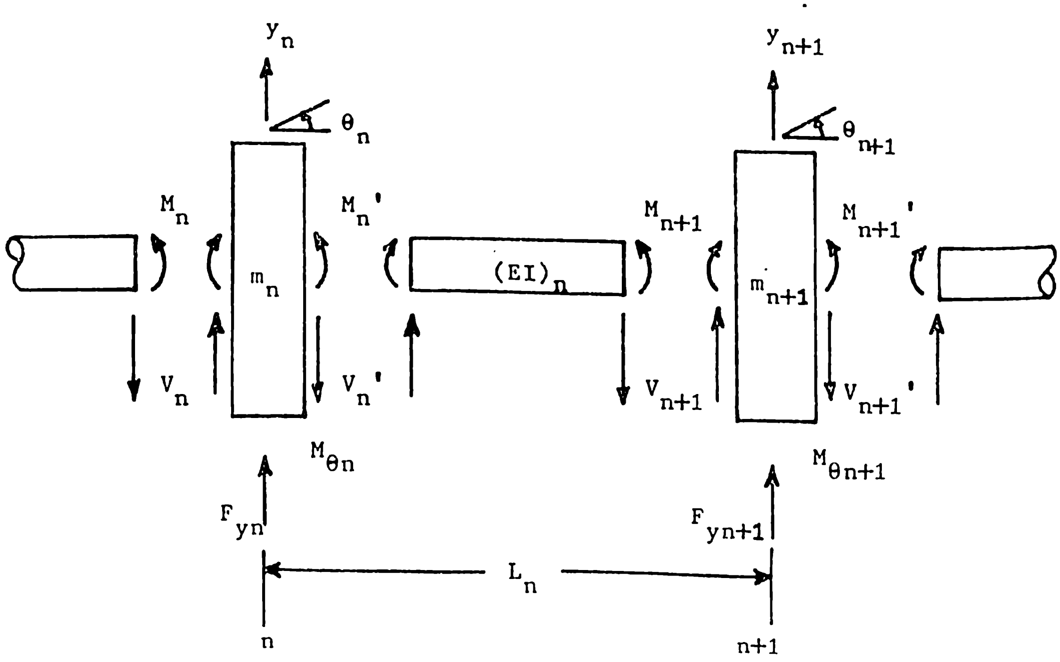


Figure 1. Sign Convention for Radial Displacement, Angular Displacement, Bending Moment, and Shear Force

where w = shaft angular velocity in rad/sec

K_{yn} = lateral stiffness coefficient

$K_{\theta n}$ = rotational stiffness coefficient

Substituting (1) and (2) into (3) and (4), we have:

$$V_n' = V_n + (w^2 m_n - K_{yn}) y_n \tag{5}$$

$$M_n' = M_n + K_{\theta n}\theta_n \tag{6}$$

For the (n+1)th station,

$$V_{n+1} = V_n' \tag{7}$$

$$M_{n+1} = L_n V_n' + M_n' \tag{8}$$

The corresponding slope and deflection at the (n+1)th station are:

$$\theta_{n+1} = bV_n' + aM_n' + \theta_n \tag{9}$$

$$y_{n+1} = cV_n' + bM_n' + L_n\theta_n + y_n \tag{10}$$

where $a = (L/EI)_n$

$b = (L^2/EI)_n/2$

$c = (L^3/EI)_n/6$

Equations (5)-(10) are used to calculate the shear, moment, slope and deflection across the rotor until the last station N is reached. Since the equations derived were that of a rotor whose two ends were assumed to be free, the boundary conditions are:

$$M_1 = V_1 = 0 \tag{11}$$

$$M_n' = V_n' = 0 \tag{12}$$

The two other quantities, y_N and θ_N, remain indeterminate in the calculation and thus arbitrary values must be assigned to them. With an assumed value of w, a total of two calculations are performed: In the first calculation, we assign $y_1 = 1$ and $\theta_1 = 0$. In the second calculation, $y_1 = 0$ and $\theta_1 = 1$. The results of these calculations are then put into a matrix

$$[A] = \begin{bmatrix} V_N'\Big|_{y_1=1,\ \theta_1=0} & V_N'\Big|_{y_1=0,\ \theta_1=1} \\ M_N'\Big|_{y_1=1,\ \theta_1=0} & M_N'\Big|_{y_1=0,\ \theta_1=1} \end{bmatrix} \tag{13}$$

where [A] is the matrix of residuals.

In matrix form, the boundary conditions of equation (12) is written as:

$$\begin{bmatrix} M_N' \\ \\ V_N' \end{bmatrix} = [A] \begin{bmatrix} y_1 \\ \\ \theta_1 \end{bmatrix} \qquad (14)$$

For M_N' and V_N' to be equal to zero, equation (14) must be equal to zero. The values of w for which this is satisfied and where the solution is not trivial, are those values which make the determinant of matrix [A] equal to zero. That is,

$$\beta \quad = \quad \det [A] = 0 \qquad (15)$$

For some value $w = w_o$, the corresponding computed determinant is $\beta = \beta_o$. Lund makes use of Taylor's series expansion to predict succeeding values of w. A first order expansion yields

$$\beta \quad \cong \quad \beta_o + (w - w_o)(d\beta/dw)_o = 0 \qquad (16)$$

To evaluate $d\beta/dw$, we first take the derivatives of equations (5) to (10) with respect to the angular velocity w to obtain the following set of equations:

$$dV_n' \quad = \quad dV_n + (-K_{yn} + w^2 m_n)dy_n + 2wm_n y_n \qquad (17)$$

$$dM_n' \quad = \quad dM_n + (K_{\theta n})d\theta_n \qquad (18)$$

$$dV_{n+1} \quad = \quad dV_n' \qquad (19)$$

$$dM_{n+1} \quad = \quad dM_n' + L_n dV_n' \qquad (20)$$

$$dy_{n+1} \quad = \quad dy_n + L_n d\theta_n + BdM_n' + CdV_n' \qquad (21)$$

$$d\theta_{n+1} \quad = \quad d\theta_n + AdM_n' + BdV_n' \qquad (22)$$

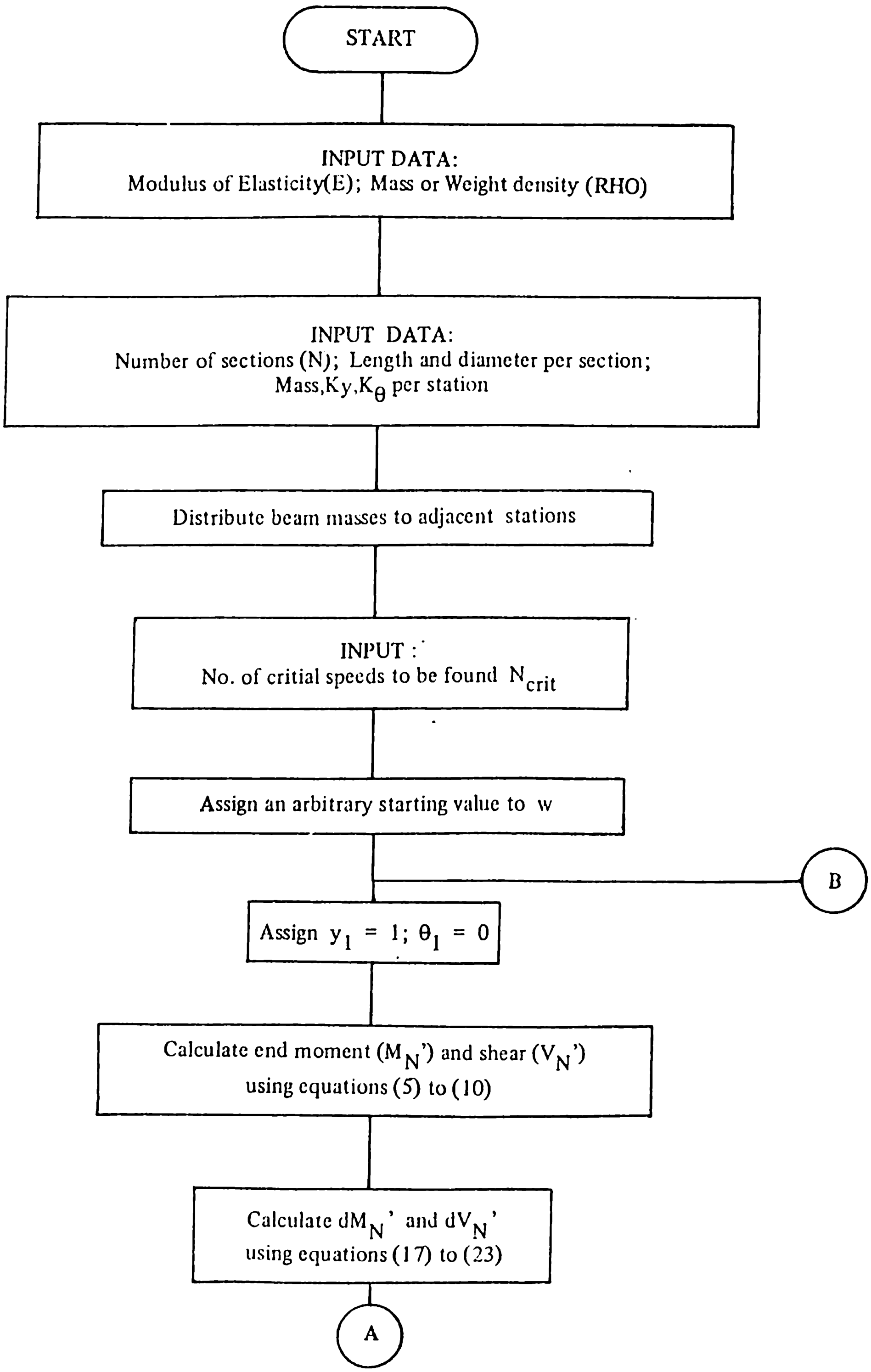
With the starting values of equations (5)-(10) being independent of the value of w, the starting values of the derivatives are zero

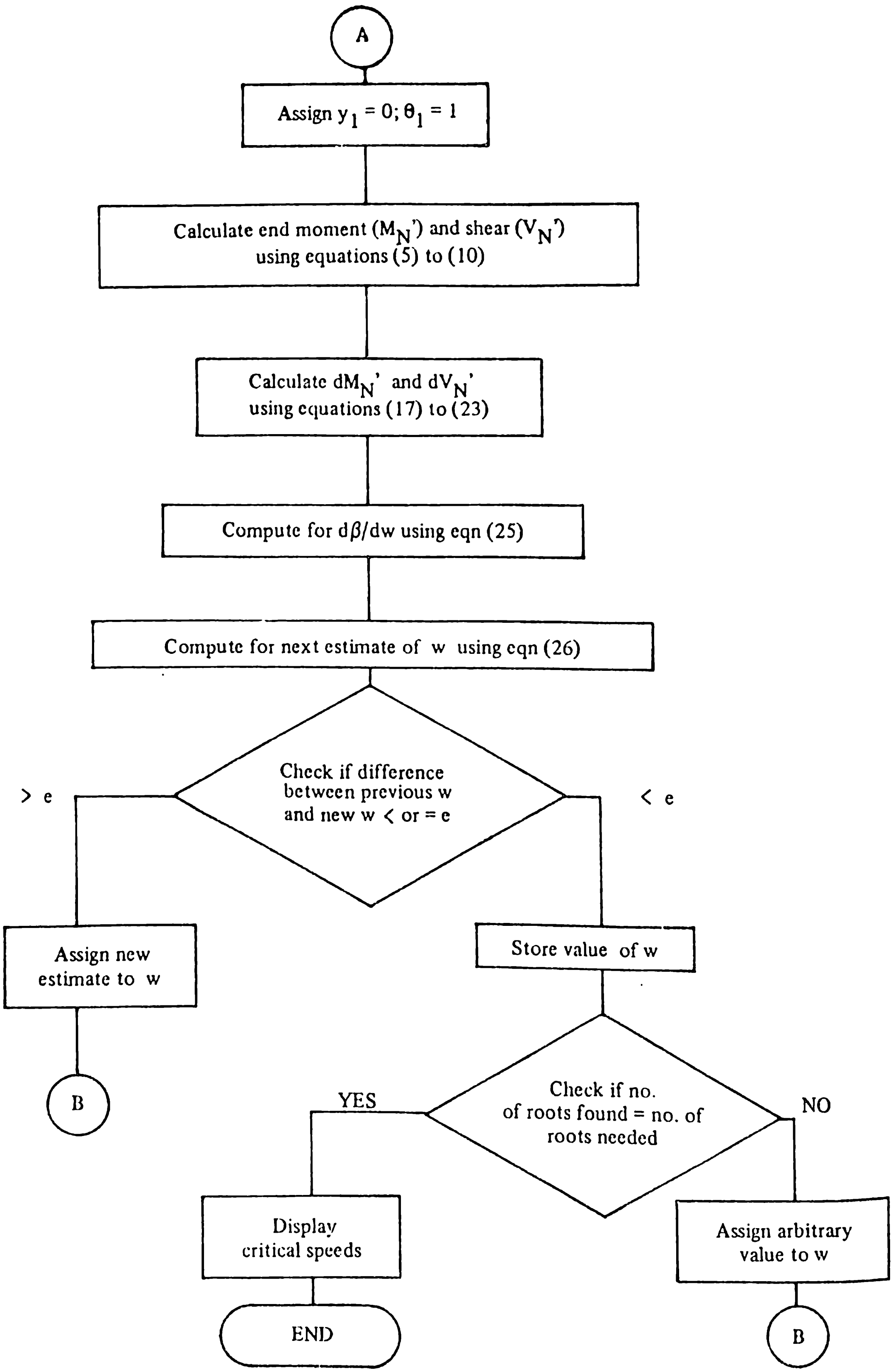
$$dy_1 = d\theta_1 = dV_1 = dM_1 = 0 \qquad (23)$$

Equations (17)-(23) are then evaluated in a similar manner as equations (5)-(10) to obtain a matrix of the form:

$$[B] = \begin{bmatrix} dV_N'\Big|_{y_1 = 1,\ \theta_1 = 0} & dV_N'\Big|_{y_1 = 0,\ \theta_1 = 1} \\ \\ dM_N'\Big|_{y_1 = 1,\ \theta_1 = 0} & dM_N'\Big|_{y_1 = 0,\ \theta_1 = 1} \end{bmatrix} \qquad (24)$$

Flowchart for Calculating the Lateral Critical Speeds of Flexible Rotors




A
Assign y1 = 0; θ1 = 1
Calculate end moment (MN') and shear (VN')
using equations (5) to (10)
Calculate dMN' and dVN'
using equations (17) to (23)
Compute for dβ/dw using eqn (25)
Compute for next estimate of w using eqn (26)
Check if difference
between previous w
and new w < or = e
> e
< e
Assign new
estimate to w
B
Store value of w
Check if no.
of roots found = no. of
roots needed
YES
NO
Display
critical speeds
END
Assign arbitrary
value to w
B

$d\beta/dw$ is then evaluated as

$$d\beta/dw = \sum_{K=1}^{2} \beta_k \qquad (25)$$

β_k is the determinant of the matrix [A] whose elements in column k have been replaced by the corresponding elements in column k of matrix [B]. Equation (16) may now be solved to yield a new estimate of w

$$w = w_o - [\beta_o/(d\beta/dw)] \qquad (26)$$

To prevent the solution from converging toward an already obtained root, Lund modified equation (26) to

$$w = w_o - \beta_o\left[(d\beta/dw)_o - \beta_0 \sum_{j=1}^{J} 1/(w_o - w_j)\right]^{-1} \qquad (27)$$

where J = the total number of roots found (or number of critical speeds found).

Starting with some estimated value of w, equation (27) is used repeatedly until the difference between two successive values becomes sufficiently small or less than a prescribed tolerance value.

It is interesting to note that although the derivation of the equations assumes a free-ended shaft, it is applicable to other support configurations. The stiffness constants K_{y_n} and K_{θ_n} can account for different shaft configurations. Thus, using only one method of solution, we can extend it to solve not only pin ended or cantilevered shafts but even rotors with intermediate supports.

SAMPLE PROBLEMS

Two sample problems will now be presented to illustrate the techniques of critical speed determination just discussed. The first problem involves an unloaded shaft of uniform cross-section and simply supported at the ends. The second problem shows a stepped shaft with intermediate loads and supported at the ends.

The method is programmed into an IBM Personal Computer and a complete listing is presented. The operator can enter the input values either in the English System of units or in the SI. Table 1 shows the units of the input quantities for both systems.

Table 1. Units of Input Quantities

Quantity	Units in English	Units in SI
E	lb/in^2	N/m^2
Weight Density	lb/in^3	N/m^3
Mass Density	$slug/in^3$	kg/m^3
Weight of Disk	lb	N
Length	in	m
Diameter	in	m

The program will also prompt for the desired number of critical speeds and will display all the results in rad/sec, cps, and rpm.

Sample Problem 1.

A circular steel shaft with uniform cross section is simply supported at the ends. The shaft is four feet long and has a diameter of 1/8 of an inch. Find the fundamental, second, and third critical speeds.

The shaft is arbitrarily subdivided into 10 sections and 11(n+1) stations as shown in Figure 2. The corresponding lengths and diameters of each section are then entered. The loads on each station as well as the corresponding displacement and rotational stiffness constants are also entered. In this problem, the loads and rotational stiffness constants are zero. For the displacement stiffness constants, we enter a fairly large value say 10 x 10^6 at the supports and zero at the intermediate stations. The results are shown below:

Critical Speeds:

Mode	rad/sec	cps	rpm
1	27.03	4.30	258.12
2	108.11	17.20	1,032.39
3	243.11	38.69	2,321.58

Sample Problem 2.

Figure 3(a) shows a steel shaft with a 20 lb gear and subjected to the various intermediate loads. The shaft diameter also varies along the length and we are to determine the first three critical speeds of the shaft.

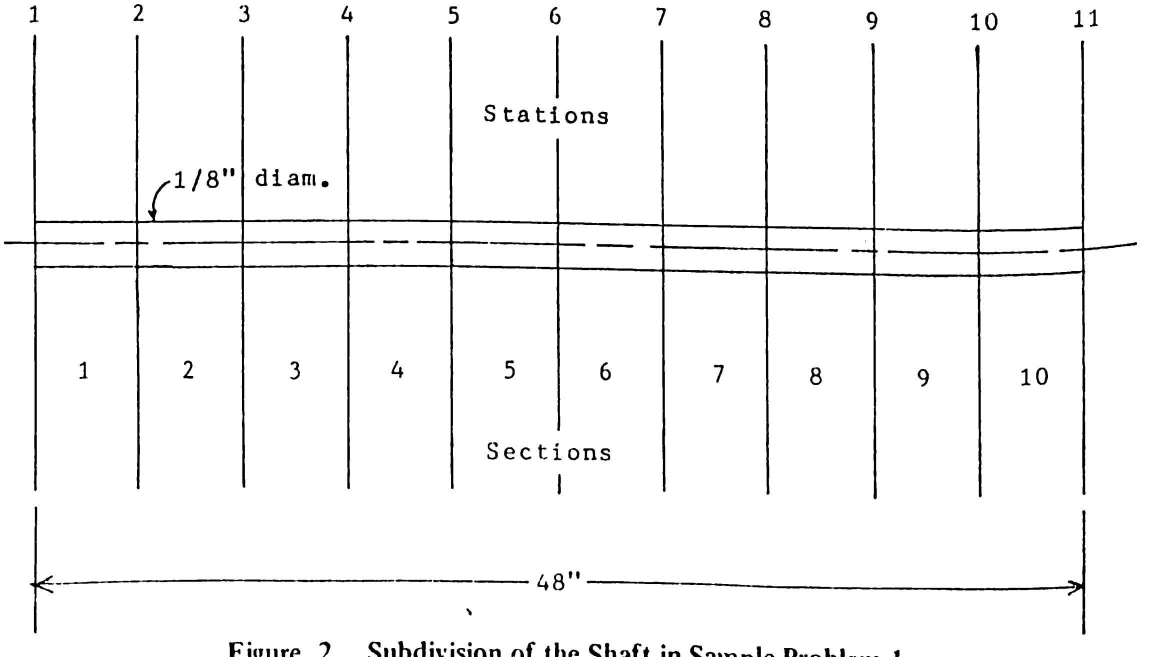


Figure 2. Subdivision of the Shaft in Sample Problem 1

The shaft is divided into nine sections and ten stations as shown in Figure 3(b). Each section length and diameter are fed into the computer as well as all the information required at every station. Note here that the shaft was divided in such a manner so that each section is of uniform diameter. The results are shown below.

Critical Speeds:

Mode	rad/sec	cps	rpm
1	704.37	112.10	6,726.23
2	3,410.14	542.74	32,564.42
3	6,725.40	1,070.38	64,222.80

These two problems can be solved with relative ease using the method just described on a digital computer. The reader should refer to pp. 551-554 of reference (4), where the Rayleigh Method is used to solve sample problem 2, in order to appreciate the usefulness of this method as a vital tool in rotor design.

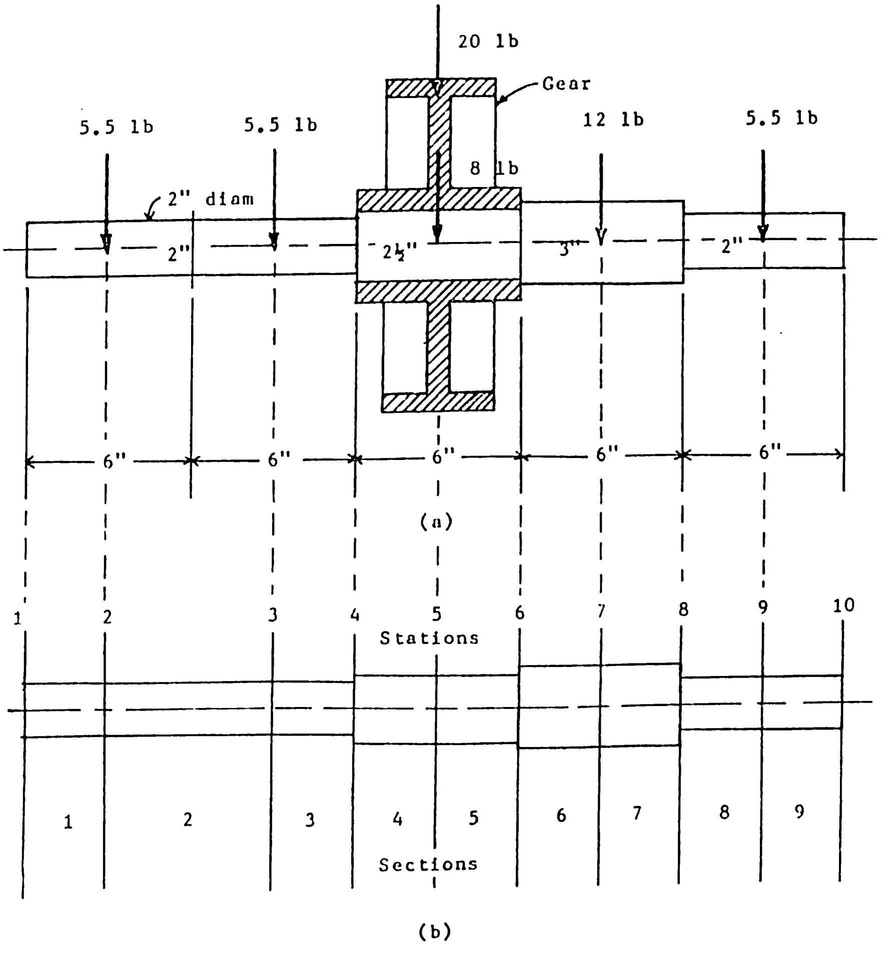


Figure 3. (a) Shaft of Sample Problem 2. (b) Subdivided Shaft Showing the Different Sections and Stations

REFERENCES

1. Church, A.H., *Mechanical Vibrations,* John Wiley, 1963.

2. Den Hartog, J.P., *Mechanical Vibrations,* McGraw Hill, 1956.

3. Lund, J.W., "Stability and Damped Critical Speeds of a Flexible Rotor in Fluid-Film Bearings", J.Eng.Ind., ASME Trans., no. 73, pp.509-517, 1974.

4. Mabie, H.H. and Ocvirk, F.W., *Mechanisms and Dynamics of Machinery,* John Wiley, 1963.

5. Tse, F.S., Morse, I.E., Hinkle, R.T., *Mechanical Vibrations,* Allyn and Bacon, 1963.

PROGRAM LISTING

```
10 '
20 '     Myklestad-Prohl Method
30 '
40 '     Use to determine the critical speeds
50 '     of a solid circular shaft of uniform density
60 '
70 CLS : KEY OFF
80 OPTION BASE 1
90 DEFINT I-K, N, O
100 DEFDBL A-H, L, M, P-Z
110 PI = 4*ATN(1#)
120 '
130 EPSILON = .000001# : NROOTS = 0#
140 '
150 LOCATE 1,25 : PRINT "The Myklestad-Prohl Method"
160 LOCATE 5,25 : PRINT "This program neglects the effects of"
170 LOCATE 6,25 : PRINT "damping and the cross coupling of"
180 LOCATE 7,25 : PRINT "stiffness coefficients."
190 LOCATE 10,25 : PRINT "by: Alexander Paran"
200 LOCATE 11,35 : PRINT "&"
210 LOCATE 12,30 : PRINT "Willie Si"
220 LOCATE 15,25 : INPUT "Press <RETURN> to continue:   "; N
230 CLS
240 LOCATE 7,25 : PRINT "What system of units will you be using?"
250 LOCATE 10,25 : PRINT "[1] English"
260 LOCATE 11,25 : PRINT "[2] SI"
270 LOCATE 14,25 : INPUT "Enter the no. of your choice:   "; RESP1
280 IF FIX(RESP1) < OR FIX(RESP1) > 2 THEN GOTO 270
290 IF FIX(RESP1) = 1 THEN G = 386# : GOTO 310
300 IF FIX(RESP1) = 2 THEN G = 1#
310 LOCATE 16,10 : INPUT "Will you be using units of [1] Weight or [2]
    Mass; RESP2
320 IF FIX(RESP) < 1 OR FIX(RESP2) > 2 GOTO 310
330 IF FIX(RESP2) = 1 THEN A$ = "Weight" : IF FIX(RESP1) = 2 THEN G
    = 9.81#
340 IF FIX(RESP2) = 2 THEN A$ = "  Mass" : IF FIX(RESP1) = 1 THEN G
    = 1#
350 '
360 CLS
370 INPUT "Modulus of Elasticity = "; E
380 PRINT A$; " Density = "; : INPUT RHO
390 RHO = RHO/G
400 INPUT "No. of Sections = "; N
410 '
420 DIM A(2,2), B(2,2), C11(N+1), C12(N+1), C21(N+1), C22(N+1)
430 DIM D(N), EI(N), L(N), LI(N), LEI(N), L2EI2(N), L3EI6(N), M(N+1)
440 DIM V(N+1,2), MO(N+1,2), S(N+1,2), Y(N+1,), YI(N+1,2)
450 '
460 CLS : PRINT "------------------------------------------------"
470 FOR I = 1 TO N
480       PRINT "Section"; I
490       PRINT
500       INPUT "      Length = "; L(I)
```

```
510        INPUT "      Diameter = "; D(I)
520        EI(I) = E*PI*D(I)^4#/64#
530        PRINT "------------------------------------------------"
540 NEXT I
550 M(1) = RHO*PI*L(1)*D(1)^2#/8#
560 FOR I = 2 TO N
570        M(I) = RHO*PI*(L(I-1)*D(I-1)^2# + L(I)*D(I)^2#)/8#
580 NEXT I
590 M(N+1) = RHO*PI*L(N)*D(N)^2#/8#
600 CLS : PRINT "------------------------------------------------"
610 FOR I = 1 TO N+1
620        PRINT "Station "; I
630        PRINT
640        PRINT "       "; A$; : INPUT " of Disk = "; DISK
650        PRINT
660        M(I) = M(I) + DISK/G
670        PRINT " Support stiffness:  ";
680        INPUT "Ky = "; C11(I)
690        INPUT "            Ko = "; C22(I)
700        PRINT "------------------------------------------------"
710 NEXT I
720 '
730 FOR I = 1 TO N
740        LEI(I) = L(I)/EI(I)
750        L2EI2(I) = L(I)^2#/(2#*EI(I))
760        L3EI6(I) = L(I)^3#/(6#*EI(I))
770 NEXT I
780 '
790 CLS
800  PRINT "Sec."; TAB(10) "Length"; TAB(30) "Dia."; TAB(50) "EI";
   TAB(70) "Mass"
810 PRINT
820 FOR I = 1 TO N
830        PRINT TAB(1) I;
840        PRINT TAB(5) USING "##.########^^^^"; L(I);
850        PRINT TAB(25) USING "##.########^^^^"; D(I);
860        PRINT TAB(45) USING "##.########^^^^"; EI(I);
870        PRINT TAB(65) USING "##.########^^^^"; M(I)
880 NEXT I
890 PRINT TAB(1) N+1;
900 PRINT TAB(65) USING "##.########^^^^"; M(N+1)
910 '
920 PRINT : PRINT
930 INPUT "How many Critical Speeds do you want to determine"; NCRIT
940 IF NCRIT = 0 THEN CLS : GOTO 2190
950 W = .1#    'Initial value assigned to omega (rad/sec)
960 '
970 DIM W(NCRIT)
980 '
990 '      Assign arbitrary values to the slope and the deflection
1000 '     and compute for the matrix of residuals at the last station.
1010 '
1020 Y = 1# : S = 0# : GOSUB 2400
1030 A(1,1) = VP : B(1,1) = DVP
1040 A(2,1) = MP : B(2,1) = DMP
1050 '
1060 Y = 0# : S = 1# : GOSUB 2400
```

```
1070 A(1,2) = VP : B(1,2) = DVP
1080 A(2,2) = MP : B(2,2) = DMP
1090 '
1100 '    Iterate for the natural frequencies
1110 '
1120 DET = A(1,1)*A(2,2) - A(2,1)*A(1,2)
1130 DET1 = B(1,1)*A(2,2) - B(2,1)*A(1,2)
1140 DET2 = A(1,1)*B(2,2) - A(2,1)*B(1,2)
1150 '
1160 DWDET = DET1 + DET2
1170 '
1180 SIGMA = 0#
1190 FOR I = 1 TO NROOTS
1200      SIGMA = SIGMA + 1#/(W - W(I))
1210 NEXT I
1220 SIGMA = SIGMA + 1#/W
1230 '
1240 WP = W - DET/(DWDET - DET*SIGMA)
1250 DELTA = ABS((WP - W)/W)
1260 LOCATE 25,1 : PRINT USING "##.####^^^^"; W;
1270 PRINT TAB(20) USING "##.####^^^^"; DELTA; : PRINT TAB(40)
     NROOTS
1280 IF DELTA <= EPSILON THEN GOTO 1290 ELSE GOTO 1310
1290 NROOTS = NROOTS + 1 : W(NOOTS) = W
1300 W = W(NROOTS)*1.1# : GOTO 1320     'Assign new estimate to
     omega
1310 W = WP
1320 IF NROOTS < NCRIT THEN GOTO 1020
1330 '
1340 CLS
1350 PRINT "Critical Speeds:" : PRINT
1360 PRINT "Mode"; TAB(20) "rad/s"; TAB(40) "cps"; TAB(60) "rpm"
1370 PRINT
1380 FOR I = 1 TO NROOTS
1390      PRINT TAB(1) I;
1400      PRINT TAB(15) USING "##.########^^^^"; W(I);
1410      PRINT TAB(35) USING "##.########^^^^"; W(I)/(2#*PI)
1420      PRINT TAB(55) USING "##.########^^^^"; 60#*W(I)/(2#*PI)
1430 NEXT I
1440 PRINT
1450 '
1460 '    Draw the modal shape
1470 '
1480 INPUT "Press <RETURN> to continue:  "; J
1490 CLS
1500 LOCATE 1,25 : PRINT "Draw Mode Shape."
1510 LOCATE 4,20 : PRINT "What are your left-end and right-end
     supports?"
1520 LOCATE 6,25 : PRINT "[1] fixed-fixed"
1530 LOCATE 7,25 : PRINT "[2] pin-fixed"
1540 LOCATE 8,25 : PRINT "[3] free-fixed"
1550 LOCATE 9,25 : PRINT "[4] pin-pin"
1560 LOCATE 11,20 : INPUT "Enter the no. of your choice:  "; RESP1
1570 IF FIX(RESP1) <1 OR FIX(RESP1) > 4 THEN GOTO 1560
1580 J = FIX(RESP1)
1590 '
```

```
1600 SCREEN 2 : CLS : WINDOW (0,-2)-(1,2) : LOCATE 12,35 : PRINT
     "Please Wait."
1610 FOR O = 1 TO NCRIT
1620      W = W(O)
1630      ON J GOTO 1650, 1700, 1750, 1800
1640 '
1650      S(1,1) = 0# : S(1,2) = 0# : Y(1,1) =0# : Y(1,2) = 0#
1660      K = 1 : V(1,K) = 1# : MO(1,K) = 0# : GOSUB 2230
1670      K = 2 : V(1,K) = 0# : MO(1,K) = 1# : GOSUB 2230
1680      GOTO 1840
1690 '
1700      Y(1,1) = 0# : Y(1,2) = 0# : MO(1,1) = 0# : MO(1,2) = 0#
1710      K = 1 : V(1,K) = 1# : S(1,K) = 0# : GOSUB 2230
1720      K = 2 : V(1,K) = 0# : S(1,K) = 1# : GOSUB 2230
1730      GOTO 1840
1740 '
1750      V(1,1) = 0# : V(1,2) = 0# : MO(1,1) = 0# : MO(1,2) = 0#
1760      K = 1 : S(1,K) = 1# : Y(1,K) = 0# : GOSUB 2230
1770      K = 2 : S(1,K) = 0# : Y(1,K) = 1# : GOSUB 2230
1780      GOTO 1840
1790 '
1800      Y(1,1) = 0# : Y(1,2) = 0# : MO(1,1) = 0# : MO(1,2) = 0#
1810      K = 1 : V(1,K) = 1# : S(1,K) = 0# : GOSUB 2230
1820      K = 2 : V(1,K) = 0# : S(1,K) = 1# : GOSUB 2230
1830 '
1840      FOR I = 1 TO N+1
1850           YI(I) = Y(I,2) - Y(I,1)*Y(N+1,2)/Y(N+1,1)
1860      NEXT I
1870 '
1880      L = 0#
1890      FOR I = 1 TO N
1900           L = L + L(I)
1910      NEXT I
1920 '
1930      FOR I = 1 TO N
1940           LI(I) = L(I)/L
1950      NEXT I
1960 '
1970      YMAX = 0#
1980      FOR I = 1 TO N+1
1990           IF ABS(YI(I)) > ABS(YMAX) THEN YMAX = YI(I)
2000      NEXT I
2010 '
2020      FOR I = 1 TO N+1
2030           YI(I) = YI(I)/YMAX
2040      NEXT I
2050 '
2060      CLS
2070      LOCATE 1,35 : PRINT "Mode "; O
2080      LINE (0,2)-(0,-2) : LINE (0,0)-(1,0)
2090      L = 0#
2100      FOR I = 1 TO N
2110           LINE (L,YI(I))-(L+LI(I),YI(I+1))
2120           L = L + LI(I)
2130      NEXT I
2140      LOCATE 23,40 : INPUT "Press <RETURN> to continue: ";
          RESP1
```

```
2150      LOCATE 24,35 : INPUT "Please wait.";
2160 NEXT O
2170 '
2180 CLS : SCREEN 0,0,0 : CLS
2190 END          '******* End of Program *******
2200 '
2210 '    Subroutine to get the deflection at each station
2220 '
2230 FOR I = 1 TO N+1
2240 '
2250      VP = V(I,K) + (W^2#*M(I) - C11(I))*Y(I,K)
2260      MP = MO(I,K) + C22(I)*S(I,K)
2270 '
2280      IF I > N THEN GOTO 2350
2290 '
2300      V(I+1,K) = VP
2310      MO(I+1,K) = L(I)*VP + MP
2320      S(I+1,K) = L2EI2(I)*VP + LEI(I)*MP + S(I,K)
2330      Y(I+1,K) = L3EI6(I)*VP + L2EI2(I)*MP + L(I)*S(I,K) + Y(I,K)
2340 '
2350 NEXT I
2360 RETURN
2370 '
2380 '    Iteration Subroutine
2390 '
2400 V = 0# : M = 0# : DV = 0# : DM = 0# : DS = 0# : DY = 0#
2410 '
2420 FOR I = 1 TO N+1
2430 '
2440      VP = V + (W^2#*M(I) - C11(I))*Y
2450      MP = M + C22(I)*S
2460 '
2470      DVP = DV + (W^2#*M(I) - C11(I))*DY + 2#*W*M(I)*Y
2480      DMP = DM + C22(I)*DS
2490 '
2500      IF I > N THEN GOTO 2660
2510 '
2520      VN = VP
2530      MN = L(I)*VP + MP
2540      SN = L2EI2(I)*VP + LEI(I)*MP + S
2550      YN = L3EI6(I)*VP + L2EI2(I)*MP + L(I)*S + Y
2560 '
2570      DVN = DVP
2580      DMN = L(I)*DVP + DMP
2590      DSN = L2EI2(I)*DVP + LEI(I)*DMP + DS
2600      DYN = L3EI6(I)*DVP + L2EI2(I)*DMP + L(I)*DS + DY
2610 '
2620      V = VN : M = MN : S = SN : Y = YN
2630 '
2640      DV = DVN : DM = DMN : DS = DSN : DY = DYN
2650 '
2660 NEXT I
2670 RETURN
```