ANCHOR-LAST DEPLOYMENT PROCEDURE FOR MOORING

Robert W. Thresher, et al

Oregon State University

Prepared for:

Office of Naval Research

June 1973

Department of

OCEANOGRAPHY

SCHOOL OF SCIENCE

OREGON STATE UNIVERSITY

Anchor-last Deployment Procedure for Mooring

by

Robert W. Thresher
and
John H. Nath

Office of Naval Research
Contract N00014-67-A-0369-0007
Project NR 083-102
and
Sea Grant Program

Reference 73-5
June 1973

98
R

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*

School of Oceanography
Oregon State University
Corvallis, Oregon 97331

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

ANCHOR-LAST DEPLOYMENT PROCEDURE FOR MOORING

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Engineering Research Report (1 July 1971 to 30 June 1973

5. AUTHOR(S) *(First name, middle initial, last name)*

Robert W. Thresher and John H. Nath

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1973	92	10

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
N00014-67-A-0369-0007	Oceanography Reference 73-5
b. PROJECT NO. NR 083-102	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
c.	
d.	

10. DISTRIBUTION STATEMENT

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Office of Naval Research Ocean Science and Technology Division Arlington, Virginia 22217

13. ABSTRACT

The anchor-last mooring procedure is investigated in order to determine the transient forces in the mooring line and the velocities of the anchor. Transient forces were determined and the results showed that no severe snap loads occurred for the cases investigated. In addition, it was found that the vertical velocity of the anchor can be small as it approaches impact with the floor of the ocean.

Both extensible (nylon and dacron) and inextensible (steel wire rope) lines were investigated. Lumped mass numerical models were developed for both cases. For the extensible line case the equations of motion were determined for each mass from Newton's Second Law, and they were integrated using a second order predictor-corrector integration technique. Hamiltonian techniques were utilized to determine the equations of motion for the inextensible line. The predictions from the numerical models show the line tensions and positions as a function of time.

I

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Numerical Models for Lines						
Anchor-Last Mooring Procedure						
Anchoring						
Lumped Mass Models						

School of Oceanography
and
Department of Mechanical Engineering
Oregon State University
Corvallis, Oregon 97331

ANCHOR-LAST DEPLOYMENT PROCEDURE FOR MOORING

A Progress Report
for the
Ocean Science and Technology Division
U.S. Office of Naval Research
Contract N00014-67-A-0369-0007
Project NR083-102

by

Robert W. Thresher
Assistant Professor of Mechanical Engineering

and

John H. Nath
Professor of Mechanical Engineering/Oceanography

Oceanography Reference 73-5

John V. Byrne
Dean
School of Oceanography

June 1973

ACKNOWLEDGMENTS

This research was supported by the United States Office of Naval Research through contract N00014-67-A-0369-0007 under project NR083-102 with Oregon State University, and by the Sea Grant Program. Part of the computer work was supported by the Oregon State University Computer Center.

TABLE OF CONTENTS

LIST OF FIGURES

Figures

ABSTRACT

The anchor-last mooring procedure is investigated in order to determine the transient forces in the mooring line and the velocities of the anchor. Transient forces were determined and the results showed that no severe snap loads occurred for the cases investigated. In addition, it was found that the vertical velocity of the anchor can be small as it approaches impact with the floor of the ocean.

Both extensible (nylon and dacron) and inextensible (steel wire rope) lines were investigated. Lumped mass numerical models were developed for both cases. For the extensible line case the equations of motion were determined for each mass from Newton's Second Law, and they were integrated using a second order predictor-corrector integration technique. Hamiltonian techniques were utilized to determine the equations of motion for the inextensible line. The predictions from the numerical models show the line tensions and positions as a function of time.

INTRODUCTION

Oceanographic research buoys moored in deep water are often deployed by the anchor-last deployment procedure. The general sequence of events in this procedures is: the buoy is deployed from the ship with the mooring line attached and distributed in some manner on the ship; while the buoy drifts away from the ship the line is paid out until only the anchor, which is secured to the lower end of the mooring line, remains on board; finally the anchor is cast overboard. At sometime during the descent of the anchor it is possible for large transient forces to be exerted on the line, which can be destructive to the line or to attached conductors and instruments. One purpose of this research is to examine the possibilities of such large forces for line scopes greater than one. The line scope is the unstressed length of the line divided by the water depth.

The Woods Hole Oceanographic Institution has conducted a series of field measurements of the launching transient in mooring lines where the anchor-last deployment procedure was utilized (1). However, the study was restricted to the consideration of moorings where the scope was less than 1.0. It was found that a fairly steady increase in line tension occurred after the anchor drop and during the free-fall stage. Then a pendulum action followed where the final line tension was equal to the submerged weight of the anchor. During the pendulum action, the line (nylon) was stretching until the anchor reached bottom. After the anchor reached bottom a reduction in line tension occurred.

A similar problem was investigated by Froidevaux and Scholten (4) with a numerical model which considered the line to be a discrete number of lumped masses connected by weightless line segments. However, when the elongation properties of the line were included, a prohibitive amount of computer time resulted and the

investigation only calculated the first and last few seconds of fall. A short system was considered and results were extrapolated to apply to the 6500 Oceanic Telescope. One scope which was greater than 1.0 was investigated but the distance from the buoy to the anchor at launch time was not determined, nor the influence of scope and line type on line tensions. The conclusion from the study was that there may be an overstress shortly after the drop and that a severe transient may occur when the anchor made impact with the ocean floor. Therefore, the anchor-first mooring procedure was recommended. During this investigation it was found that large transient forces shortly after anchor deployment can be an artifact of the numerical program.

The Electronics Division of General Dynamics has employed the anchor-last deployment procedure for several moorings of the forty-feet diameter oceanographic buoy for the Office of Naval Research. It has been noticed that some conductors in the upper portion of the line have experienced large stresses during the deployment of some moorings but not for others. Thus it was suspected that large transient forces occurred during the anchor deployment either concentrated at the buoy,or propagating up the line to the buoy. This investigation predicts that large snap loads should not occur for such moorings.

Goeller (5) investigated snap loads in steel cables but the study was restricted to the consideration of straight lines with a scope equal to 1.0 where the mooring was already in place and the upper end was excited with a sinusoidal motion.

A review of the current techniques for the dynamic analysis of morring line systems was presented by Casarella and Parsons (3).

The first technique to be employed during this study was like the one used by Nath (7, 8) where the line is considered to be a continuum and the partial

differential equations of motion are solved by the method of characteristics. Later the analysis technique was changed to a lumped mass approach that was very similar to the analysis made by Wang (10) except that the equations developed were in a different coordinate reference frame and the integration technique was different.

NUMERICAL MODELS FOR EXTENSIBLE LINES

The numerical models presented here pertain specifically to the anchor-last mooring procedure. The basic equations and procedures for the method of characteristics solution were presented by Nath (7, 8) and they will not be reproduced here. However, certain modifications to the earlier program will be presented which shows the attempts to utilize the method of characteristics for the anchor-last procedure. The lumped mass model for the extensible mooring line will be presented in detail. It is similar to the approach taken by Froidevaux and Scholten (4). The lumped mass model for the inextensible case will not be presented in detail because it is available in the work by Rupe (9).

Method of Characteristics

The computer program presented in (7) was modified to accept the new boundary conditions of the anchor-drop problem. The first runs produced results which gave unrealistic mooring line shapes. Attempts to improve on the numerical procedure met with only limited success. The problem was caused by the extremely tight curvature of the line in the first few seconds of fall. The calculations of line positions did not correspond to the line velocities.

The solution technique was to solve for the normal velocity, VN, the tangential velocity VT and the local angle, θ, made by the tangent to the line and the horizon.

For simplicity and as a first approximation the buoy attachment point was assumed to be fixed. At first it was assumed that the mooring line was composed of a series of straight lines between calculation points. Then the coordinates of points along the line could be calculated using:

$$X_i = X_{i-1} + \Delta S \cos \theta_i$$

$$Z_i = Z_{i-1} + \Delta S \sin \theta_i \qquad (1)$$

where ΔS is the distance between calculation points. It is easy to visualize why this technique would give unreasonable results if the mooring line were to be given a high curvature or "kink", which is exactly what takes place during the early stages of anchor drop.

In an attempt to eliminate this problem the following more accurate approximation was used to calculate the mooring line position:

$$\frac{dx_i}{ds} = \cos \theta_i$$

$$\frac{dz_i}{ds} = \sin \theta_i \qquad (2)$$

Since the θ_i's are known at constant intervals along the line, then all that is required is to integrate the above equations along the line from the attachment point to the anchor. For example, the trapazoidal integration formula gives

$$X_i = X_{i-1} + \frac{\Delta S}{2} (\cos \theta_i + \cos \theta_{i+1})$$

$$Z_i = Z_{i-1} + \frac{\Delta S}{2} (\sin \theta_i + \sin \theta_{i+1}) \qquad (3)$$

which is equivalent to assuming that the line shape is quadratic between calculation points. Both the above formula and Simpson's rule were used, however, the curvature of the line is so steep that even these higher order approximations did not eliminate the problem. Therefore, the decision was made to proceed with the lumped mass model of the following section.

Lumped Mass Model

A mooring line can be modeled as a group of discrete masses interconnected by springs as illustrated in Fig. 1. The various distributed forces which act on a mooring line are gravity, hydrodynamic viscous drag and acceleration, and the line tension, which is influenced by the internal, or structural, damping and the stress-strain relationship. In the lumped mass model these forces are discretized and it is assumed that they act at each mass. Figure 2 shows the various forces acting on a typical mass. The magnitude of the various forces were calculated in the following manner:

Spring Force. From closed solution studies of taut line mooring dynamics it has been found that the external hydrodynamic forces are considerably larger than the internal damping force. For this study in particular, where periodic oscillations are not a factor, the internal damping force was negligible. Therefore it was not included into the equations of motion of the masses.

Hooke's Law was taken in the general form

$$\sigma_i = f(\varepsilon_i) \tag{4}$$

calculating the strain, using the straight line distance between masses, then

$$\varepsilon_i = \frac{\Delta \ell_i}{\ell_i} = \frac{\left[(x_{i+1} - x_i)^2 + (z_{i+1} - z_i)^2\right]^{1/2} - \ell_i}{\ell_i} \tag{5}$$

where ℓ_i is the unstressed distance between masses and X_i and Z_i are the coordinates of the mass M_i. This gives the spring force SF_i as:

$$SF_i = \text{Area} \cdot f\left\{\frac{\left[(X_{i+1} - X_i)^2 + (Z_{i+1} - Z_i)^2 - \ell_i\right]}{\ell_i}\right\} .$$

The net spring force acting on the mass M_i is given by

$$\begin{Bmatrix}\text{spring force} \\ \text{x direction}\end{Bmatrix} = SFX_i = SF_i \cos(\theta_i) - SF_{i-1} \cos(\theta_{i-1})$$

$$\begin{Bmatrix}\text{spring force} \\ \text{z direction}\end{Bmatrix} = SFZ_i = SF_i \sin(\theta_i) - SF_{i-1} \sin(\theta_{i-1}) . \qquad (7)$$

The Hooke's Law relationship was assumed to be in the form

$$\sigma_i = C_1 (\varepsilon_i)^{C_2}$$

where C_1 and C_2 are constants. The values used for the various lines are given in the appendix.

Hydrodynamic Drag Forces. Using the quadratic drag law, the tangential and normal drag forces on the mass M_i are given by

$$\left\{\text{Tangential Drag}\right\} = DFT_i = \frac{CDT \cdot \rho \cdot AT}{2} (VT_i)|VT_i|$$

$$\left\{\text{Normal Drag}\right\} = DFN_i = \frac{CDN \cdot \rho \cdot AN}{2} (VN_i)|VN_i| \qquad (8)$$

where tangent and normal are referrenced to the angle θAV_i as illustrated in Figure 2, and

CDT = Tangential Drag Coefficient

CDN = Normal Drag Coefficient

VT_i = Tangential Velocity of Mass M_i

VN_i = Normal Velocity of Mass M_i

ρ = Fluid Density

AT = Tangential Area = $\pi D\ell_i$

AN = Normal Area = $D\ell_i$

Defining VCX and VCZ to be the local velocities due to currents and waves, then the tangential and normal velocities of mass M_i are given by

$$VT_i = (\dot{x}_i - VCX) \cos(\theta AV_i) + (\dot{z}_i - VCZ) \sin(\theta AV_i)$$

$$VN_i = -(\dot{x}_i - VCX) \sin(\theta AV_i) + (\dot{z}_i - VCZ) \cos(\theta AV_i) \qquad (9)$$

Added Mass Forces. The added mass forces due to the acceleration of the line masses through the still fluid was considered in the usual way. That is, the excitation force accelerates the line and the fluid medium around the line. The pressure distribution on the line from the accelerating fluid is characterized by the displaced mass of fluid times a coefficient called the added mass coefficient and the product is called the added mass. The sum of the line mass and the added mass coefficient for the line in the direction normal to the line was taken equal to that for a smooth cylinder (0.5) and in the tangential direction it was assumed to be zero.

Governing Equations for the Line. Substituting the above forces into Newton's second law and including the hydrodynamic added mass effect with the actual line mass (the sum is the virtual mass)

$$\sum F_{tangential} = (M_i + CIT \cdot \rho \cdot Vol_i)\ ACT_i$$

$$\sum F_{normal} = (M_i + CIN \cdot \rho \cdot Vol_i)\ ACN_i \qquad (10)$$

where

CIT = Added Mass Coefficient Tangential

CIN = Added Mass Coefficient Normal

ρ = Fluid density

Vol = Displaced Line Volume = $\frac{\pi D^2}{4} \ell_i$

ACN = Normal Acceleration of the Mass M_i

ACT = Tangential Acceleration of the Mass M_i

Then the accelerations are given by

$$ACT_i = \frac{(SFX_i)\cos(\theta AV_i) + (SFZ_i - WT_i)\ \sin\ (\theta AV_i)\ + DFT_i}{(M_i + CIT \cdot \rho \cdot Vol_i)}$$

$$ACN_i = \frac{-(SFX_i)\ \sin\ (\theta AV_i) + (SFZ_i - WT_i)\ \cos\ (\theta AV_i) + DFN_i}{(M_i + CIN \cdot \rho \cdot Vol_i)} . \qquad (11)$$

In the x, z coordinate system the accelerations are given by

$$\ddot{x}_i = ACT_i\ \cos\ (\theta AV_i) - ACN_i\ \sin\ (\theta AV_i)$$

$$\ddot{z}_i = ACT_i\ \sin\ (\theta AV_i) + ACN_i\ \cos\ (\theta AV_i) . \qquad (12)$$

Governing Equations for the Anchor. Figure 3 shows a free body diagram of the anchor from which the following governing equations can be obtained

$$\ddot{x} = \frac{T\cos(\beta) - 1/2\ CD\cdot A\cdot\rho\cdot\dot{x}\cdot V}{(MA + CI\cdot\rho\cdot Vol)}$$

$$\ddot{z} = \frac{T\sin(\beta) - W - 1/2\ CD\cdot A\cdot\rho\cdot\dot{z}\cdot V}{(MA + CI\cdot\rho\cdot Vol)} \qquad (13)$$

where CD = the anchor drag coefficient

T = line tension at the anchor

CI = the anchor added mass coefficient, constant in all directions

A = Area = $\pi D_a^2/4$.

For simplification, the water velocities from current and waves have been ignored. They can be easily added if necessary. The anchor was assumed to be a sphere, with a high drag coefficient to recognize the fact it is not actually a sphere.

Numerical Integration. Numerical integration of Eqs. (12) and (13) was accomplished with the following second order predictor-corrector set (6) .

Velocity Predictor:

Let

$$\ddot{x}_i(t) = f(x_i(t),\ z_i(t),\ \dot{x}_i(t),\ \dot{z}_i(t)\ )$$

and $$\ddot{z}_i(t) = g(x_i(t),\ z_i(t),\ \dot{x}_i(t),\ \dot{z}_i(t)\ ) \qquad (14)$$

where the functions f and g are equations (12) above. The velocity predictors are then

$$\dot{x}_i^p(t) = \dot{x}_i(t-\Delta t) + 2\Delta t\ \ddot{x}_i(t)$$

and $$z_i^P(t) = \dot{z}_i(t-\Delta t) + 2\Delta t\, \ddot{z}_i(t). \tag{15}$$

Displacement Predictor: Similarly the predictors for displacements are

$$x_i^P(t) = x_i(t) + \frac{\Delta t}{2}\left(\dot{x}_i(t) + \dot{x}_i^P(t+\Delta t)\right)$$

and $$z_i^P(t) = z_i(t) + \frac{\Delta t}{2}\left(\dot{z}_i(t) + \dot{z}_i^P(t+\Delta t)\right). \tag{16}$$

Velocity Corrector: The corrector equations are for velocities

$$\ddot{x}_i^P(t+\Delta t) = f(x_i^P(t+\Delta t),\ z_i^P(t+\Delta t),\ \dot{x}_i^P(t+\Delta t),\ \dot{z}_i^P(t+\Delta t))$$

$$\ddot{z}_i^P(t+\Delta t) = g(x_i^P(t+\Delta t),\ z_i^P(t+\Delta t),\ \dot{x}_i^P(t+\Delta t),\ \dot{z}_i^P(t+\Delta t)) \tag{17}$$

$$\dot{x}_i^C(t+\Delta t) = \dot{x}_i(t) + \frac{\Delta t}{2}\left(\ddot{x}_i(t) + \ddot{x}_i^P(t+\Delta t)\right)$$

and $$\dot{z}^C(t+\Delta t) = \ddot{z}_i(t) + \frac{\Delta t}{2}\left(\ddot{z}_i(t) + \ddot{z}_i^P(t+\Delta t)\right). \tag{18}$$

Displacement Corrector: Likewise, the corrector equations for displacements are

$$x_i^c(t+\Delta t) = x_i(t) + \frac{\Delta t}{2}\left(\dot{x}_i(t) + \dot{x}_i^c(t+\Delta t)\right)$$

and $$z_i^c(t+\Delta t) = z_i(t) + \frac{\Delta t}{2}\left(\dot{z}_i(t) + \dot{z}_i^c(t+\Delta t)\right). \tag{19}$$

The corrector values are then considered to be the actual position and velocity. The appropriate error at each step of the integration for each mass M

can be calculated (6) using the following formula

$$\varepsilon(\dot{x}_i) = (\dot{x}_i^c(t) - \dot{x}_i^p(t))/5.$$

$$\varepsilon(\dot{z}_i) = (\dot{z}_i^c(t) - \dot{z}^p(t))/5. \tag{20}$$

These errors were monitored at all time during the integration and when errors exceeded a specified limit, the time step was cut in half and the integration continued; conversely, if they fell below some minimum limit the time step was doubled.

From the form of equations (14) through (19) it is evident that both position and velocity must be known for the time $(t - \Delta t)$ and (t). However, when starting a solution or just after the time step has been changed the position and velocity will only be known for the last time step, and therefore some other equations must be used to generate this required data. In this particular case the Runge-Kutta third order single step integration formulas were used.

Runge-Kutta: Equations (14) can be integrated to give

$$\dot{x}_i(t+\Delta t) = \dot{x}_i(t) + (\ell_1 + 4\ell_2 + \ell_3)/6 \tag{21}$$

$$\dot{z}_i(t+\Delta t) = \dot{z}_i(t) + (k_1 + 4k_2 + k_3)/6$$

and

$$x_i(t+\Delta t) = x_i(t) + (n_1 + 4n_2 + n_3)/6$$

$$z_i(t+\Delta t) = z_i(t) + m_1 + 4m_2 + m_3)/6 \qquad (22)$$

where

$$\ell_1 = \Delta t\ f(x_i(t),\ z_i(t),\ \dot{x}_i(t),\ \dot{z}_i(t)\ )$$

$$\ell_2 = \Delta t\ f(x_i(t) + n_1/2,\ Z_i(t) + m_1/2,\ \dot{x}_i(t) + \ell_1/2,\ \dot{z}_i(t) + k_1/2)$$

$$\ell_3 = \Delta t\ f(x_i(t) - n_2 + 2n_2,\ z_i(t) - m_1 + 2m_2,\ \dot{x}_i(t) - \ell_1 + 2\ell_2,$$

$$\dot{z}_i(t) - k_1 + 2k_2) \quad ; \qquad (23)$$

$$k_1 = \Delta t\ g(x_i(t),\ z_i(t),\ \dot{x}_i(t),\ \dot{z}_i(t))$$

$$k_2 = \Delta t\ g(x_i(t) + n_1/2,\ z_i(t) + m_1/2,\ \dot{x}_i(t) + \ell_1/2,\ \dot{z}_i(t) + k_1/2)$$

$$k_3 = \Delta t\ g(x_i(t) - n_2 + 2n_2,\ z_i(t) - m_1 + 2m_2,\ \dot{x}_i(t) - \ell_1 + 2\ell_2,$$

$$\dot{z}_i(t) - k_1 + 2k_2) \quad ; \qquad (24)$$

$$n_1 = \Delta t\ \dot{x}_i(t)$$

$$n_2 = \Delta t\ (\dot{x}_i(t) + 1/2\ \ell_1)$$

$$n_3 = \Delta t(\dot{x}_i(t) - \ell_1 + 2\ell_2) \quad ; \tag{25}$$

and

$$m_1 = \Delta t\ \dot{z}_i(t)$$

$$m_2 = \Delta t(\dot{z}_i(t) + 1/2\ k_1)$$

$$m_3 = \Delta t(\dot{z}_i(t) - k_1 + 2k_2) \quad . \tag{26}$$

Using the predictor-corrector set and the single step Runge-Kutta equation for the first time step the governing equation for the lumped mass mooring line and the anchor can be numerically integrated. In this routine, the error can be automatically controlled by increasing or decreasing the time step Δt through Eqs. (20). The general outline of the calculation scheme is illustrated in Fig. 4.

It should be noted that higher ordered predictor-corrector schemes exist which increase the precision of the integration. However, the accuracy of the technique used may be well within the accuracy of the entire computations. Until data from physical models becomes available to prove otherwise it is felt that this technique gives adequate results.

A tentative model for the inextensible case has been developed jointly with the Oregon State University Sea Grant Program (9). Hamiltonian techniques were utilized to develop the equations of motion. The explicit detailing of the numerical procedures will not be presented here as for the inextensible line case because this has been accomplished in Reference (9).

The model assumed the line to be perfectly inextensible. Thus the motion of each mass is more directly influenced by the motion of each other mass than for the extensible line case. A listing of the program is presented in the Appendix B.

Prior to the production runs several preliminary tests were made. The first of these showed large dynamic loading in the segment of the line near the anchor at about 20 seconds after the anchor was released from the "standard catenary configuration" of Fig. 5a. This result was reported in the 1972 report to ONR together with results where the anchor was released from the "goose neck" catenary configuration of Fig. 5b. The second configuration showed much lower dynamic loading. Since that time the segment lengths were made considerably smaller and the control on the time step was improved by using the integration method described in this report. It is now evident that the high dynamic loading reported earlier was an artifact of the first lumped mass numerical program. The results presented in the following section show only a small amount of transient loading during the early stages of anchor deployment.

After this experience, each test case to be run was divided into several time segments and the computer results were examined at the end of each time segment. In this manner, links in a critical area along the line could be subdivided while links in areas where the line tensions were more constant could be consolidated. After these adjustments, the computer continued the solution from that time until the end of the next time segment where the output would again be examined and adjustments made in the link lengths. In this way, accurate results could be obtained without using an excessive number of links and computer time could be conserved.

Nylon Line

Figures 6 through 13 present the results for the 2.5" diameter nylon mooring line for both the 15° catenary and the 60° catenary initial condition. Figures

12 and 13 show results for the "goose neck" initial position. Figure 6 shows the line shape for various times during the drop for the 15° catenary. Figure 7 shows the tension contours for the 15° catenary as a function of position and time. The most notable feature of these results is that the line has essentially constant tension during most of the drop, and there is no snap load during the fall or after the anchor touches bottom. Figures 8 and 9 present the same plots for the 60° catenary. Although the line tension is not constant the tensions vary slowly with time and position due to the high hydrodynamic resistance forces acting on the line.

Figure 10 compares the tension for both catenaries at the anchor link for the first 140 seconds of drop. Figure 11 compares the vertical anchor velocities during the drop. It is interesting to note that the terminal velocity for the 12,000 lb. anchor in free fall (unrestrained by a mooring line and where drag acts only on the anchor) is about 50 ft/sec. When the anchor is attached to a mooring line, the velocity approaches the terminal free fall velocity during the first few seconds of fall and then reduces considerably due to the drag on the mooring line. Figure 11 also shows the horizontal anchor velocities. The maximum tension during the drop was approximately equal to the anchor weight.

In general, the long run times for these solutions (see Appendix E) were a result of the high curvature in the line near the anchor. When the anchor is released it drops straight down which causes a "kink" in the line just behind the anchor. Early during the drop the line motion tends to be mostly tangential especially near the anchor, which requires that the line on the buoy side of the "kink" must move around the "kink" at relatively large velocities. For the lumped mass model, this means that a lump on the buoy side of the "kink" is

accelerated toward the "kink" and obtains a relatively large horizontal velocity. As this lump moves around the "kink" its velocity must very rapidly change from nearly horizontal to vertical. Because of this large acceleration the time step is drastically reduced as a mass moves around the "kink".

The problem is compounded if the links are relatively large. As a large heavy link is accelerated toward the "kink" it obtains a considerable amount of momentum and it has a tendency to over-shoot the "kink" which causes jumps in the line tension. In one particular case the horizontal anchor velocity changed directions as a relatively large line mass overshot the "kink". This result is shown in Figure 11 where the horizontal anchor velocity becomes positive for a short time during the early portion of the drop.

Comparison of the output for the "standard catenary configuration" and the "goose neck catenary configuration" of Figure 5 showed about the same results provided sufficiently small link lengths were used. Figures 12 and 13 show the results for the "goose neck" catenary during the first 70 seconds of the drop. The comparison of anchor link tensions for the "goose neck catenary" and the "standard catenary" shown in Figure 13 indicates very similar results except that at any given time the "standard catenary configuration" has about 1000 lb. more tension. Since the"standard catenary configuration" seemed to be a more realistic starting configuration, and because the results are very similar, no further runs were made using the "goose neck" catenary.

Dacron Line

Figure 14 through 19 present the results for a 1.0 "diameter dacron mooring line for both the 15° catenary and 60° catenary initial conditions. Figure 14 shows the line shape for various times during the drop for the 15° catenary.

Figure 15 shows the tension contours for the 15° catenary as a function of position along the line and time. The computer runs for this case were terminated after 140 seconds of drop because the results in general were not greatly different from the 2.5" diameter nylong line presented in Figs. 6 and 7. There was no snap and the tensions seemed to be fairly constant and approximately equal to the anchor weight. Figures 16 and 17 present the same plots for the 60° catenary initial condition. For this case the computer runs were terminated after 100 seconds of drop because of the general similarity to the results obtained for the nylon line.

Figure 18 compares the tension in the anchor link for both catenary initial conditions during the first 140 seconds of drop. Figure 19 compares the anchor velocities during the drop and here again these results are very similar to the results for the nylon line.

Steel Wire Rope

At the time of this writing a computer program to model inextensible steel or chain mooring lines has just been completed. A few preliminary runs have been made for the anchor drop problem. The results of one of these runs is presented in Figs. 20 and 21. Figure 20 shows the line shape for various times during the drop for the 15° catenary initial configuration of a fictitious equivalent 2.5" diameter steel cable with a density of 350 lb/ft^3. This diameter was selected initially to compare with the 2.5" diameter nylon line results. Subsequent runs will use more realistic diameters.

Although the program was only run for about the first 85 seconds of fall, the results are considerably different from the previous results as would be

expected due to the large line weight and the inextensibility conditions. The results show that the line falls almost as fast as the anchor, which is expected. Figure 21 shows the tension contours in the line and the results are markedly different with the maximum tensions occurring at the attachment point. Although these results are preliminary, in that the link lengths should be reduced and a more realistic case should be run, they do indicate general tendencies. The drop was run for only 85 seconds due a limited computer budget.

CONCLUSIONS AND DISCUSSION

1. The lumped mass model together with an appropriate integration scheme such as the predictor-corrector method is a useful tool for the dynamic analysis of extensible mooring lines. It is easily adapted to almost any line configuration and can readily handle non-linearities.
2. The results obtained for the anchor drop problem show no dynamic line snap for the configurations that were considered. It appears that the high drag of the long mooring line quickly disipates the kinetic energy of the anchor, which slows the anchor to a velocity well below its free fall terminal velocity of 50 ft/sec.
3. The elastic properties of the line do not seem to play a great role in the line motion during anchor drop. The nylon and the dacron line showed similar position time histories.
4. Mooring line diameter did not seem to play an important role in the mooring line motion. Both the 2.5" line and the 1.0" line showed about the same position time histories.
5. For an anchor drop from the 15° catenary as modelled here, the line tension is essentially constant with respect to line position.
6. For an initial condition of a 60° catenary the tension increases more slowly than for the 15° catenary.
7. At no time, for any run, was the line tension greater than the anchor weight for the extensible mooring lines.
8. Time-velocity histories of the anchor show that the anchor settles onto the bottom at a vertical velocity of about 5 ft/sec for the 2.5" nylon rope.

9. Preliminary runs made for an inextensible steel line show a much different behavior. At least for the limited runs made, no snaps loads were observed. Further runs must be made to verify this result.

10. It is postulated that the high loads during buoy mooring implantment that have been experienced at sea for scopes greater than one were not caused by shock, or snap, loads in the line. Possibly, during the time of anchor fall, the line was relatively taut and the dynamic loads from waves acting on the buoy created the high tension forces close to the buoy.

1. Berteaux, H. O. and Walden, R. G., "Analysis and Experimental Evaluation of Single Point Moored Buoy Systems", W.H.O.I. Reference No. 69-36.

2. Bulirsch, R. and Stoer, J., "Numerical Treatment of Ordinary Differential Equations by Extrapolation Methods", Numerische Mathematik 8, 1-13 (1966).

3. Casarella, M. J. and Parsons, M.,"A Survey of Investigations on the Motion of Cable Systems under Hydrodynamic Loading", Marine Tech. Soc. J. Vol. 24, No. 4, pp 27-44 (Jul-Aug 1970).

4. Froidevaux, M. R. and Scholten, R. A., "Calculation of the Gravity Fall Motion of a Mooring System", M.I.T. Instrumentation Laboratory Report. E-2319, August, 1968.

5. Goeller, J. E. "A Theoretical and Experimental Investigation of Snap Loads in Stranded Steel Cables", U. S. Naval Ordinance Laboratory Report NOLTR 69-215, November 26, 1969.

6. Lapidus, L. and Seinfeld, J. H., "Numerical Solution of Ordinary Differential Equations", Academic Press 1971.

7. Nath, J. H., "Dynamics of Single Point Ocean Moorings of a Buoy--A Numerical Model for Solution by Computer", Oregon State University, Dept. of Oceanography, ONR Progress Report, Ref. 69-10. July, 1969.

8. Nath, J. H., "Analysis of Deep Water Single Point Moorings", Colorado State University, Technical Report CER 70-71 JHN 4, August, 1970.

9. Rupe, R., "An Inextensible Lumped Mass Dynamical Mooring Line Model", M.S. Thesis, Oregon State University. To be completed, June 1973.

10. Wang, T. H., "A Two-degree-of-freedom Model for the Two-dimensional Dynamic Motion of Suspended Extensible Cable Systems". Naval Ship Research and Development Center, Bethesda MD. Report 3663, October 1971.

FIGURES

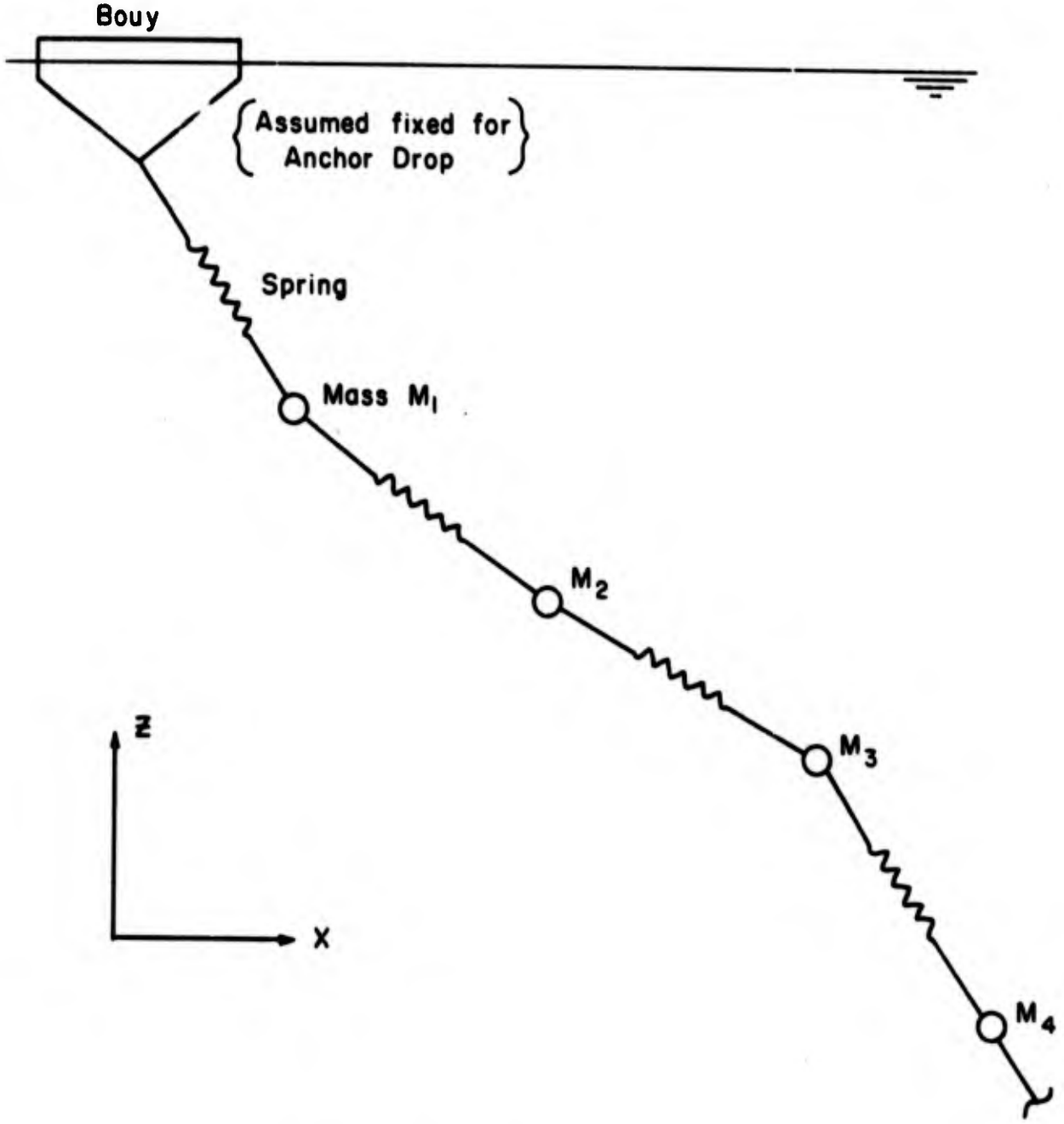
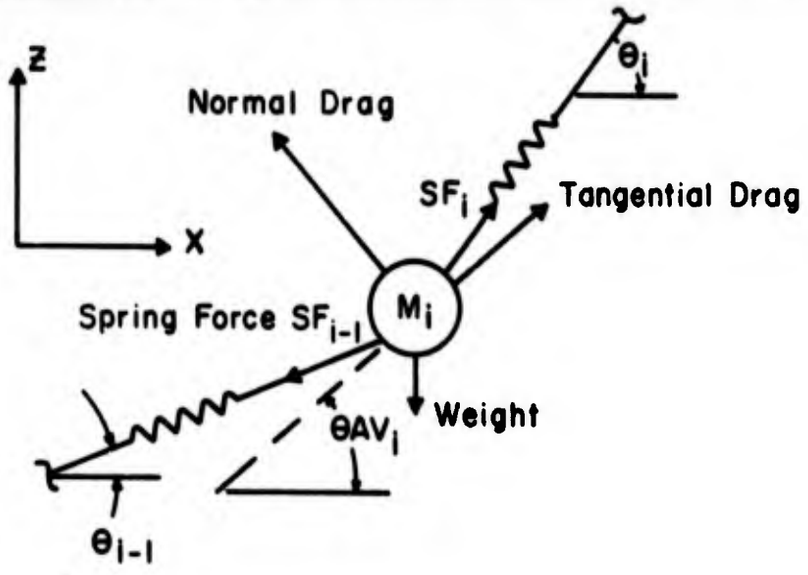


Figure 1. The Lumped Mass Model



$$\theta AV_i = (\theta_{i-1} + \theta_i)/2$$

Figure 2. Free Body Diagram of a Typical Mass

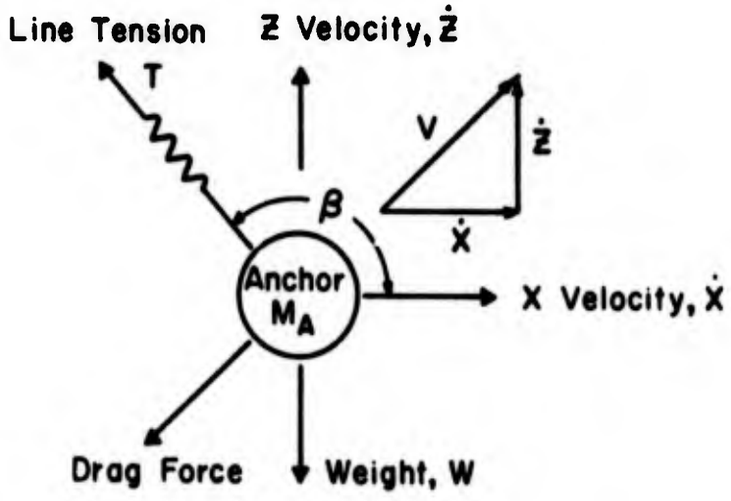


Figure 3. Free Body Diagram of the Anchor

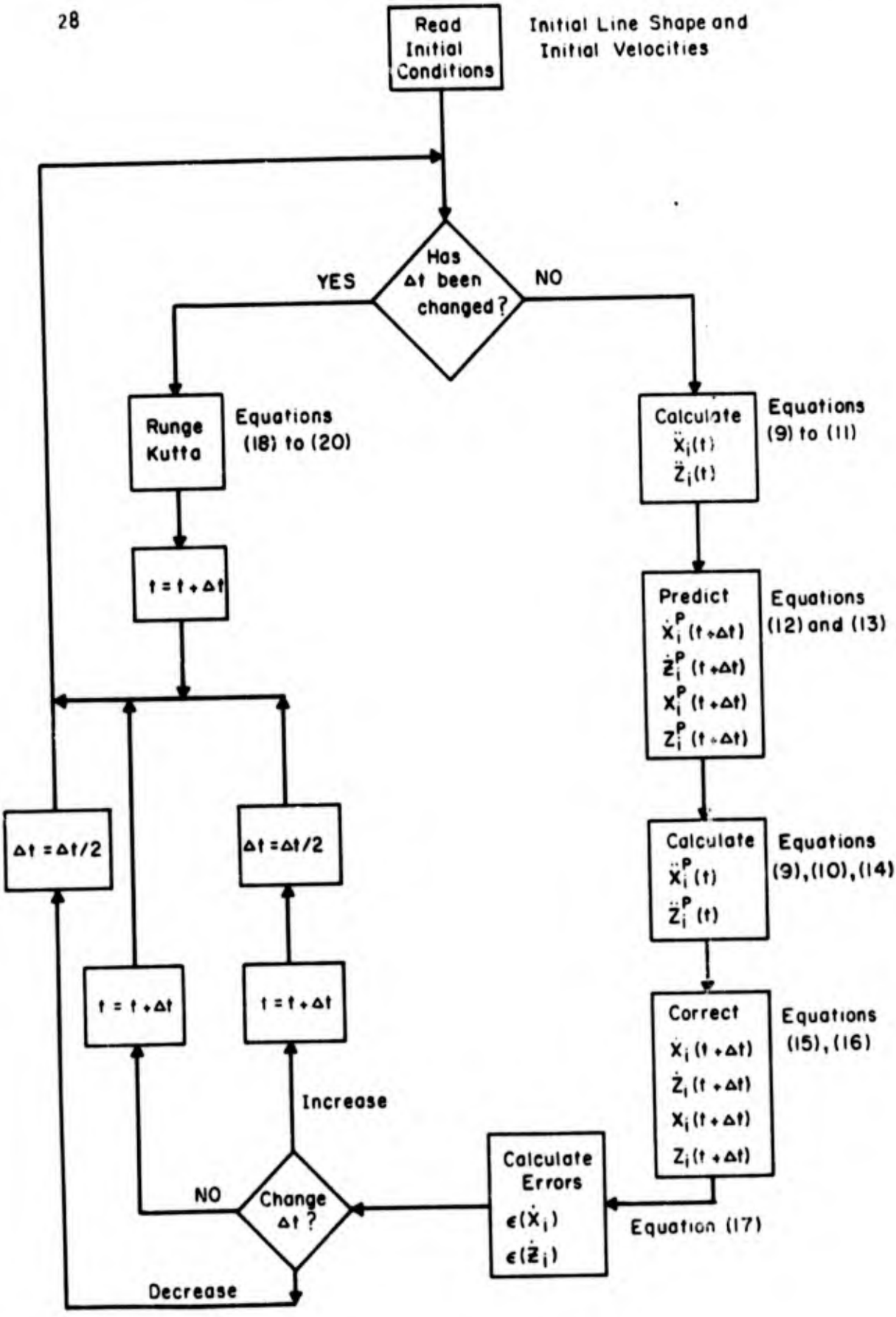


Figure 4. Computation Flow Chart

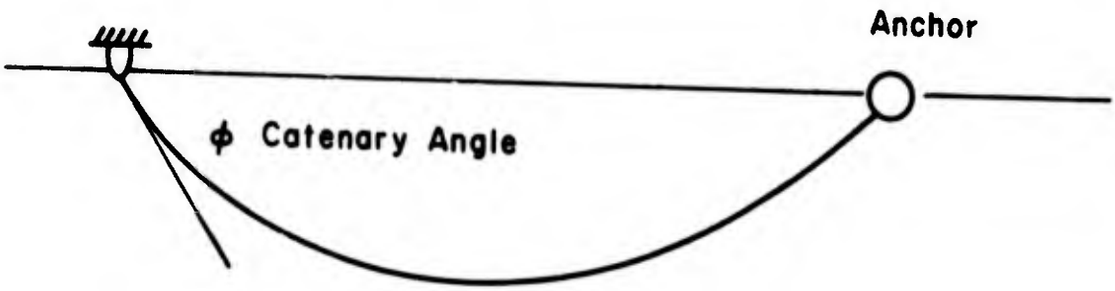


Figure 5a. Standard Catenary Configuration

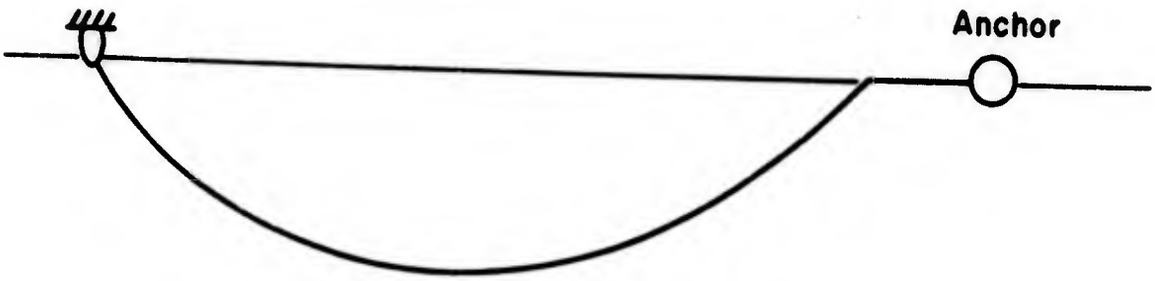


Figure 5b. Goose Neck Catenary Configuration

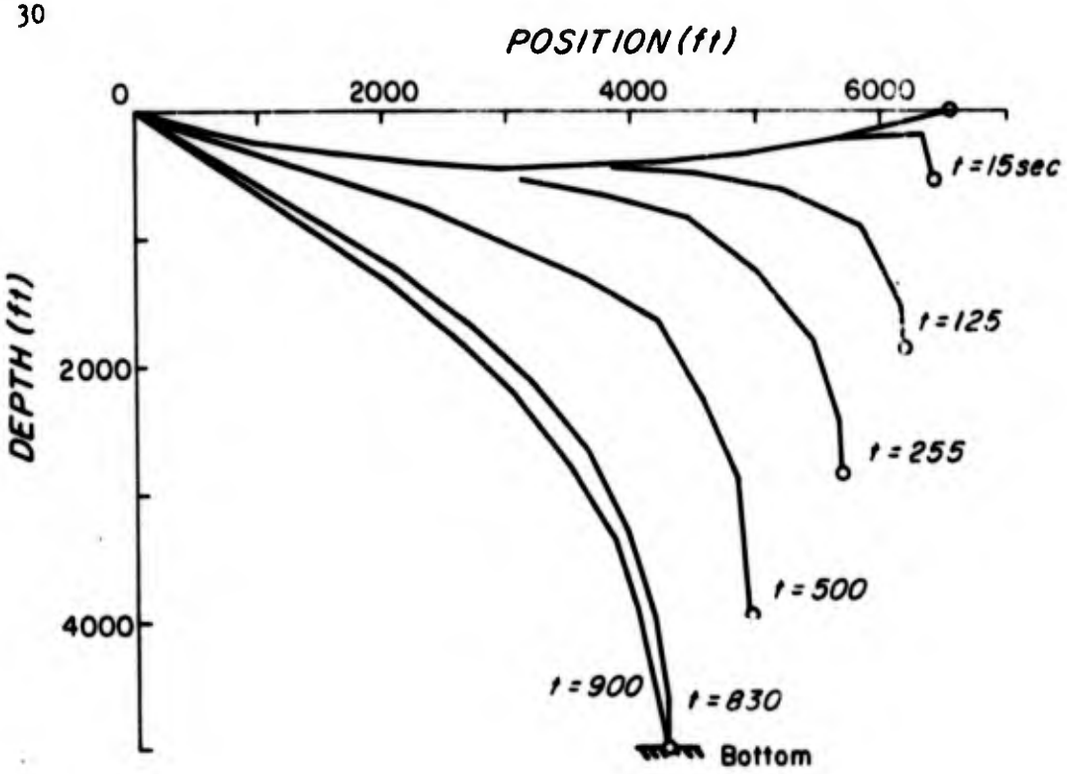


Figure 6. Nylon Line 2.5" Dia (NVR05)

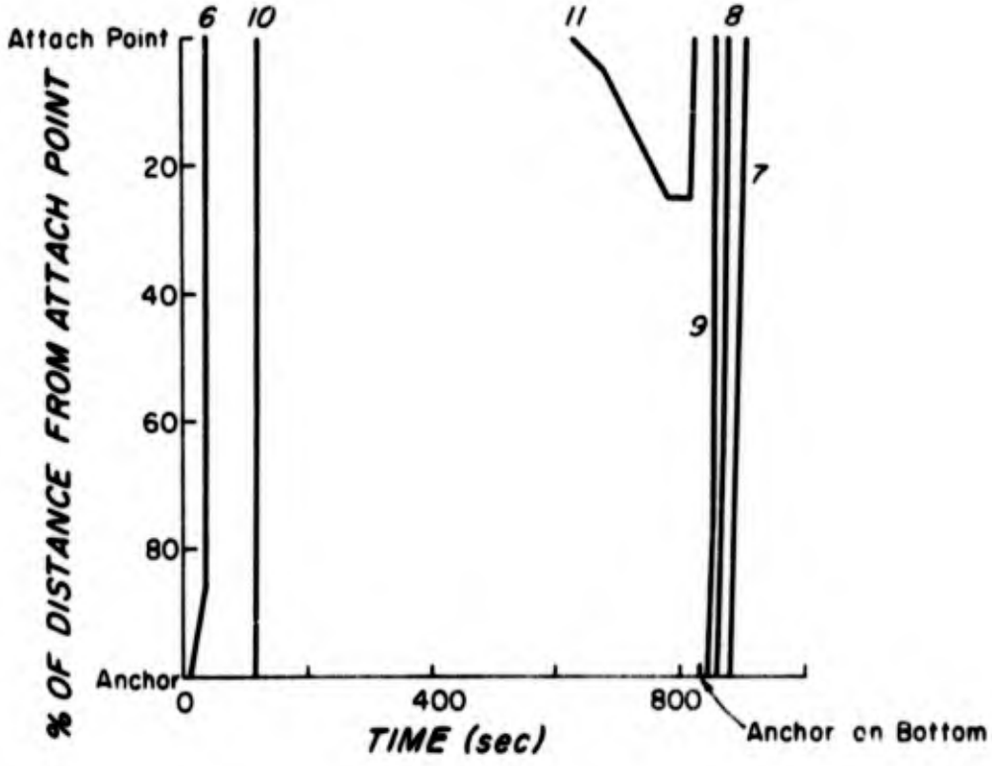


Figure 7. Tension Contours % EIP
15^{o} Catenary Nylon Line 2.5" Dia (NVR05)

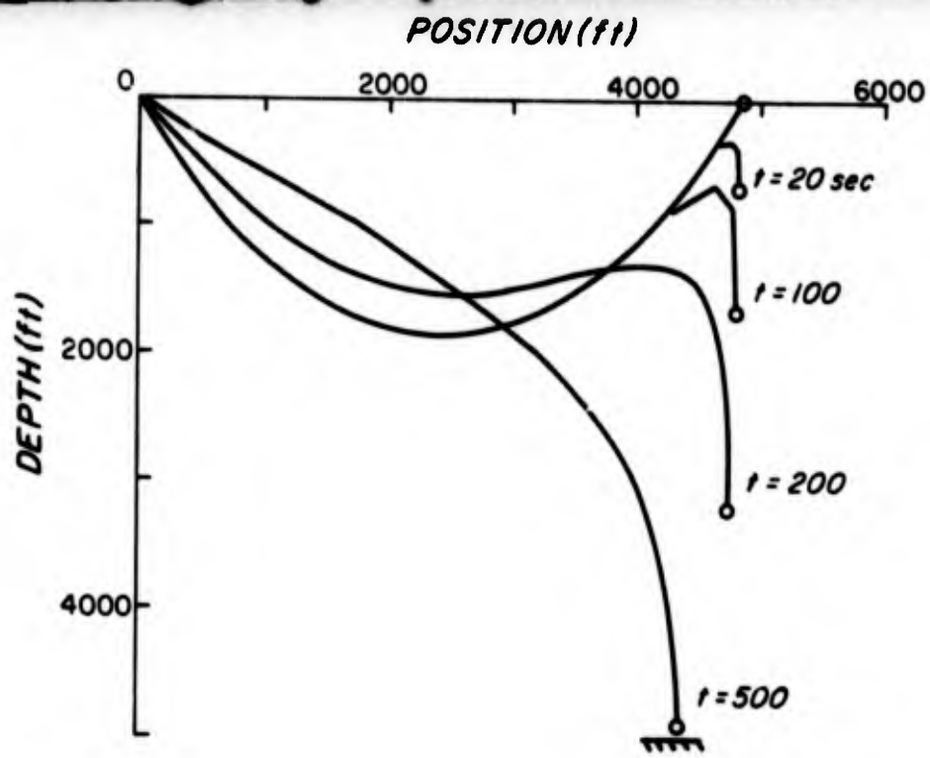


Figure 8. 60° Catenary Nylon Line 2.5" Dia. (NVR05)

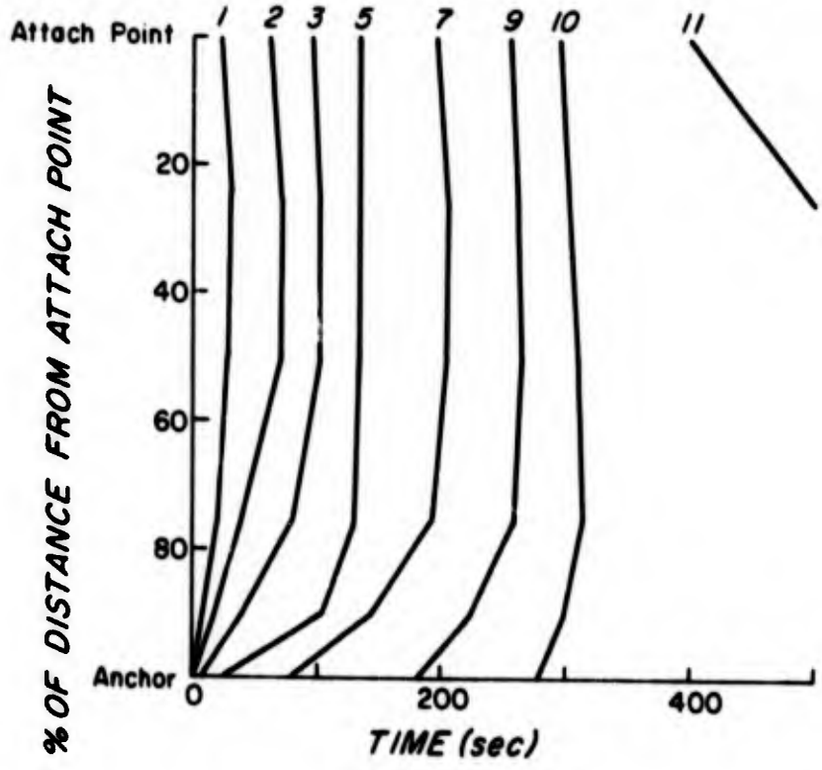


Figure 9. Tension contours KIP
60° Catenary Nylon Line 2.5" Dia (NVR05)

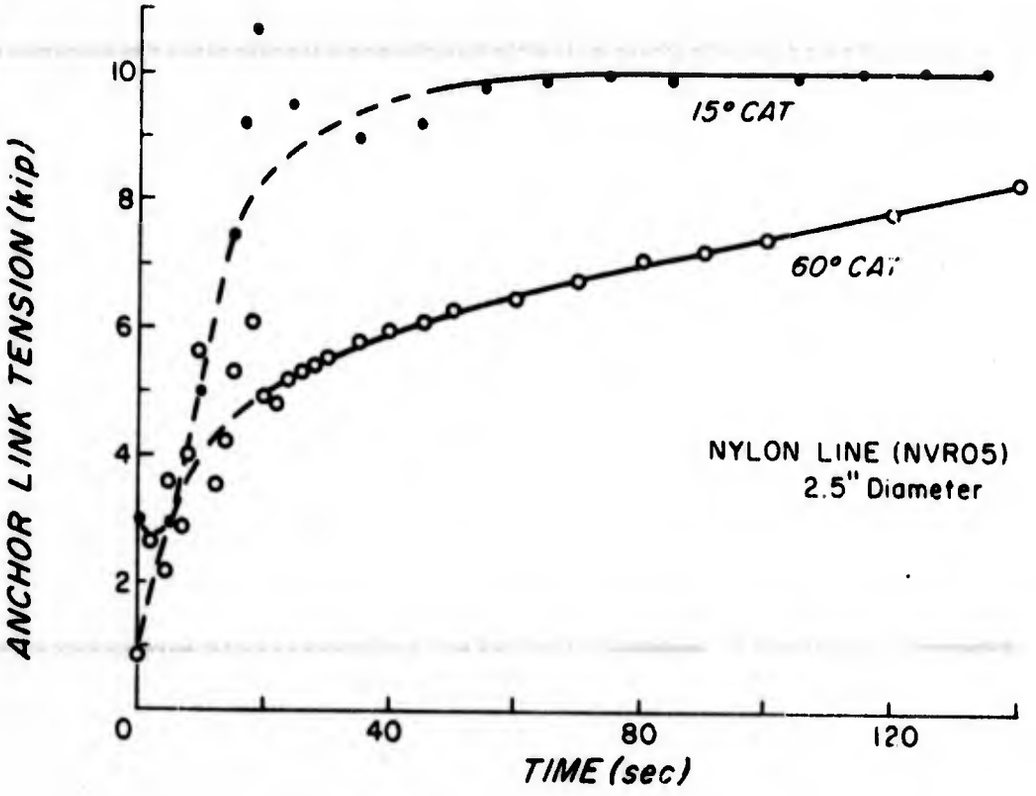


Figure 10. Anchor Link Tension for Nylon Line

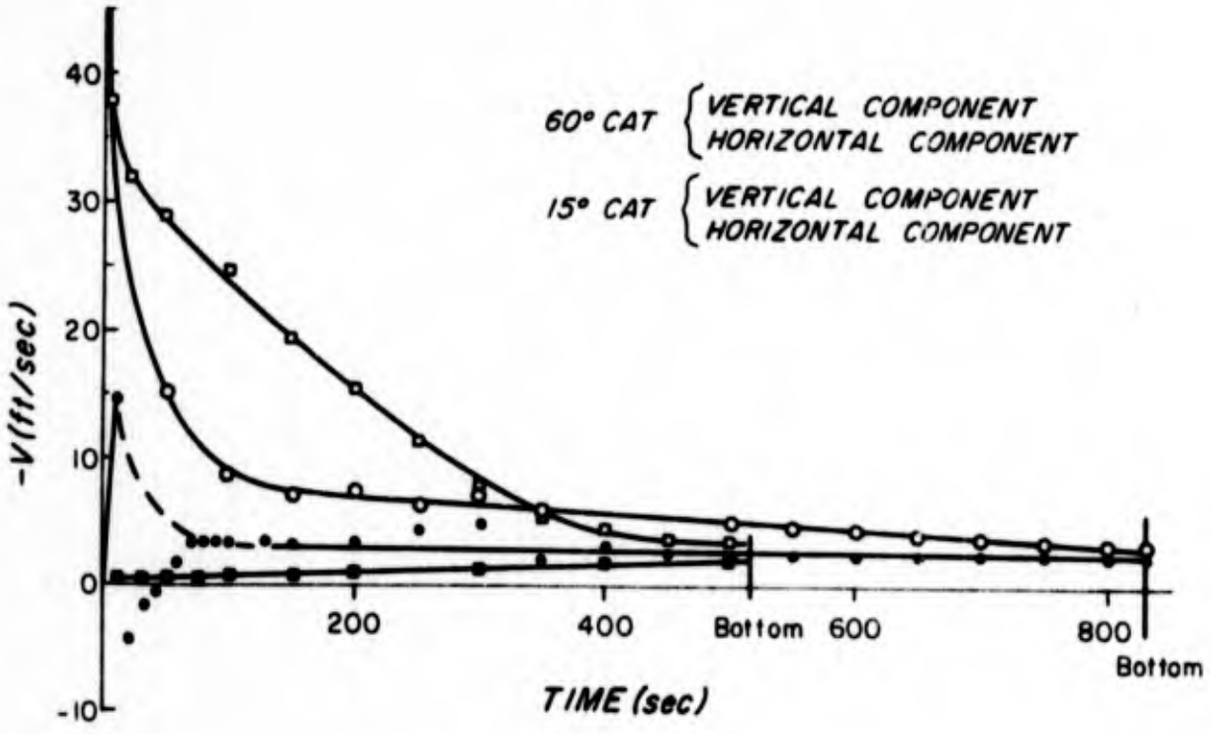


Figure 11. Components of Anchor Velocity
Nylon Line 2.5" Dia. (NVR05)

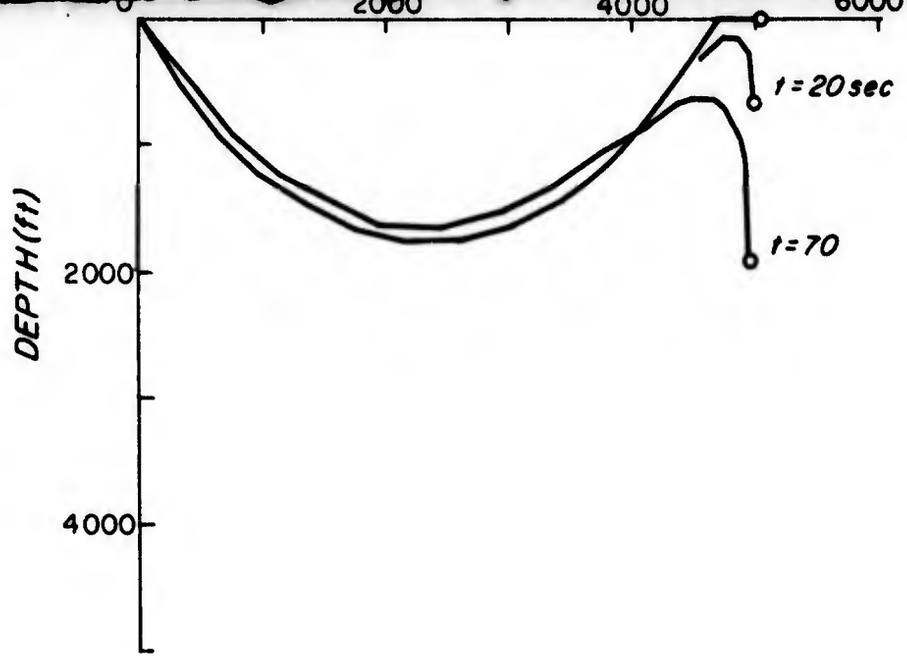


Figure 12. 60° "Goose Neck" Catenary, Nylon Line, 2.5" Dia. (NVR06)

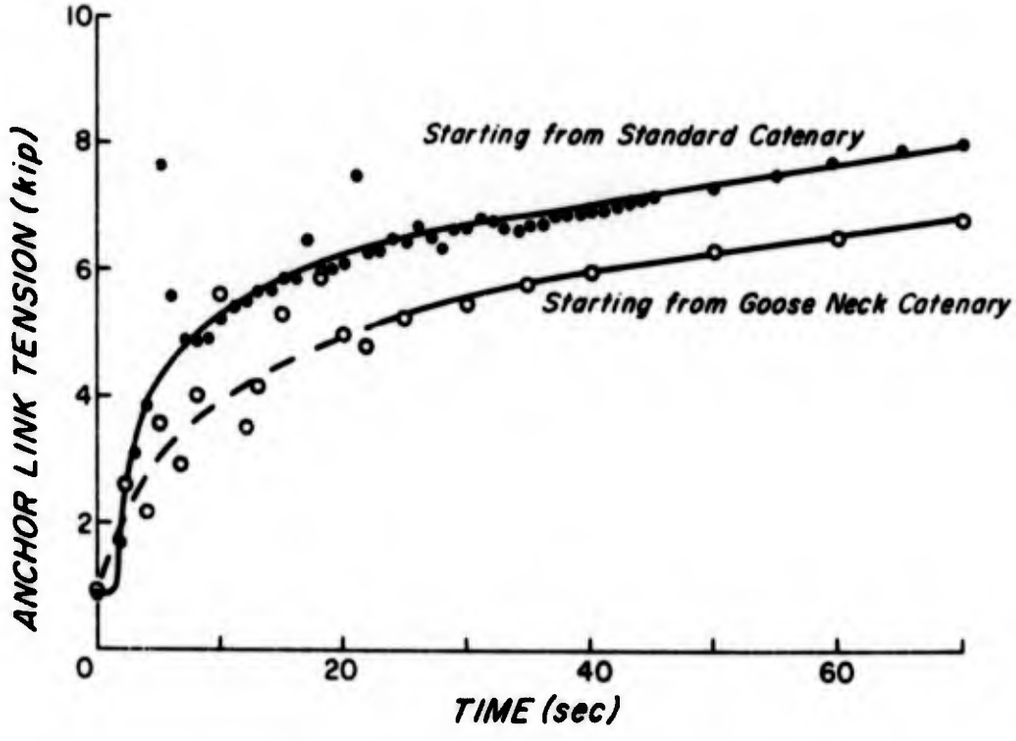


Figure 13. Anchor Link Tension for Nylon Lines Starting from 60° Standard Catenary and 60° "Goose Neck Catenary".

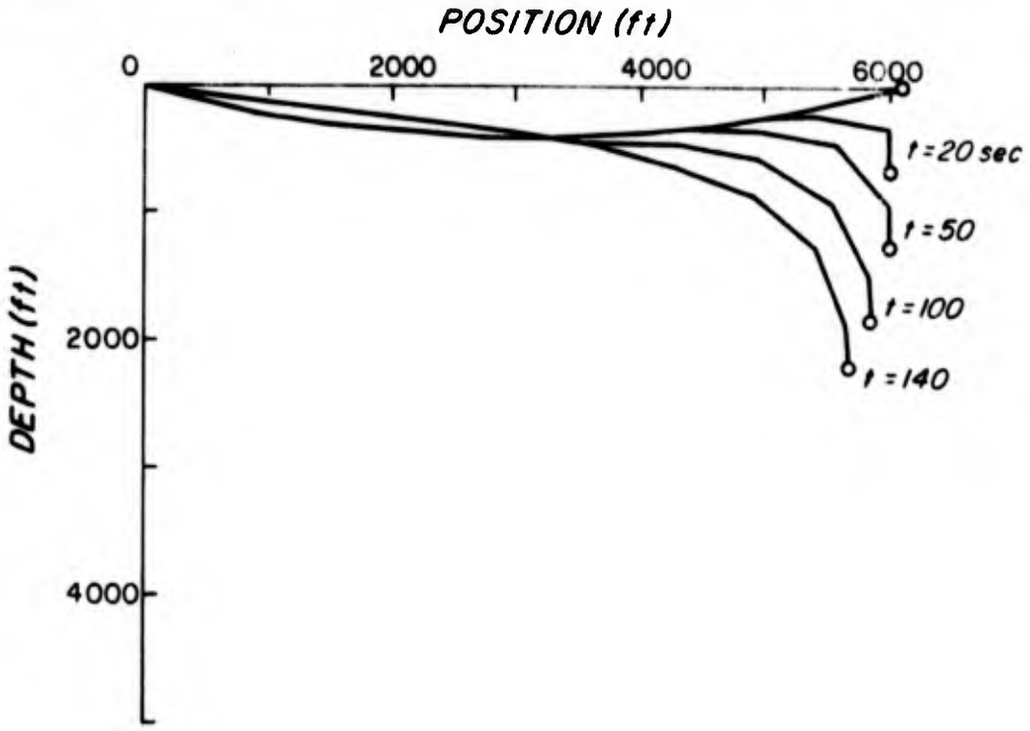


Figure 14. 15° Cat Dacron Line (VR08)
1.0" Dia.

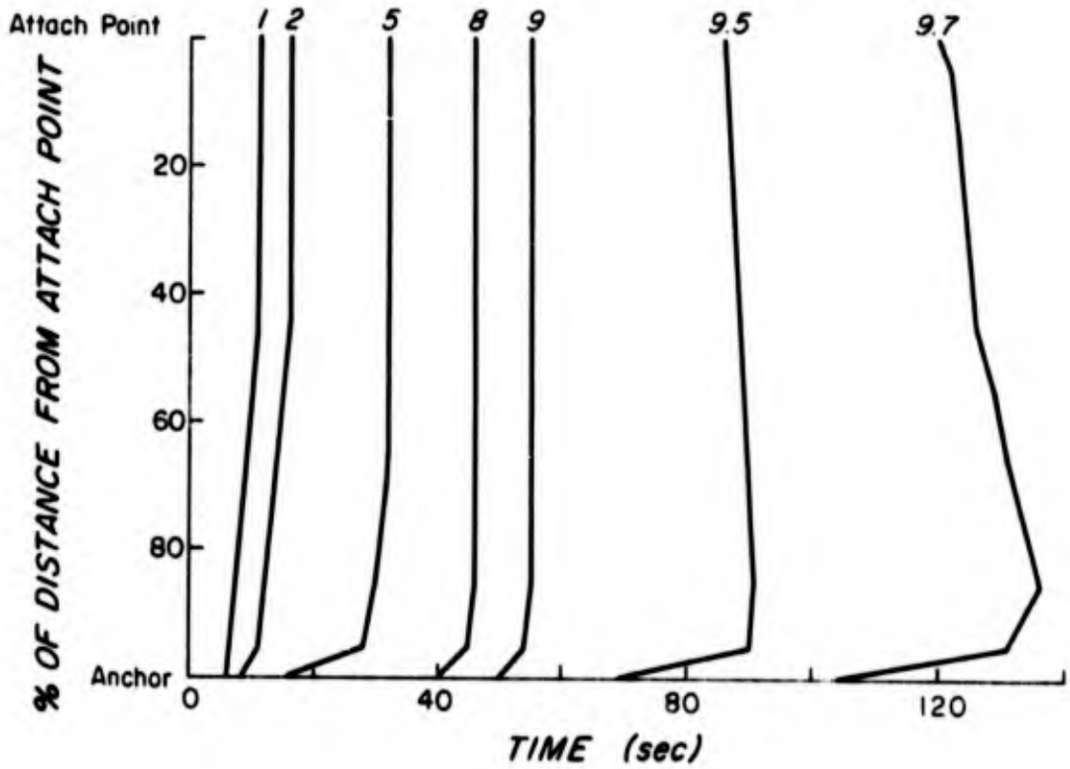


Figure 15. Tension Contours @ KIP
15° Catenary Dacron Line 1.0" Dia. (NVR08)

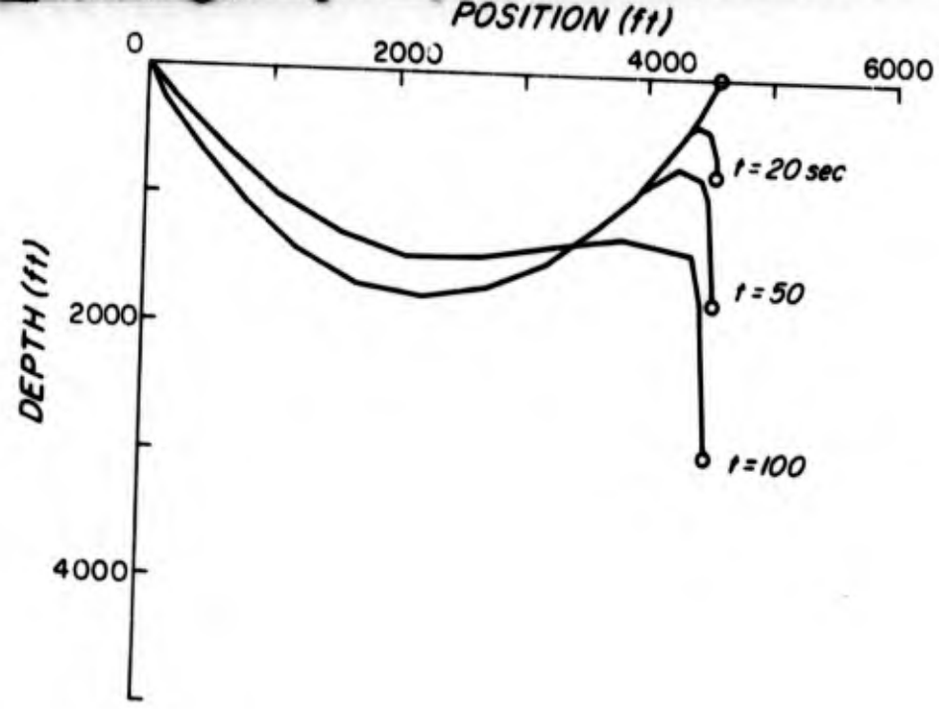


Figure 16. 60° Catenary Dacron Line 1.0" Dia. (NVR08)

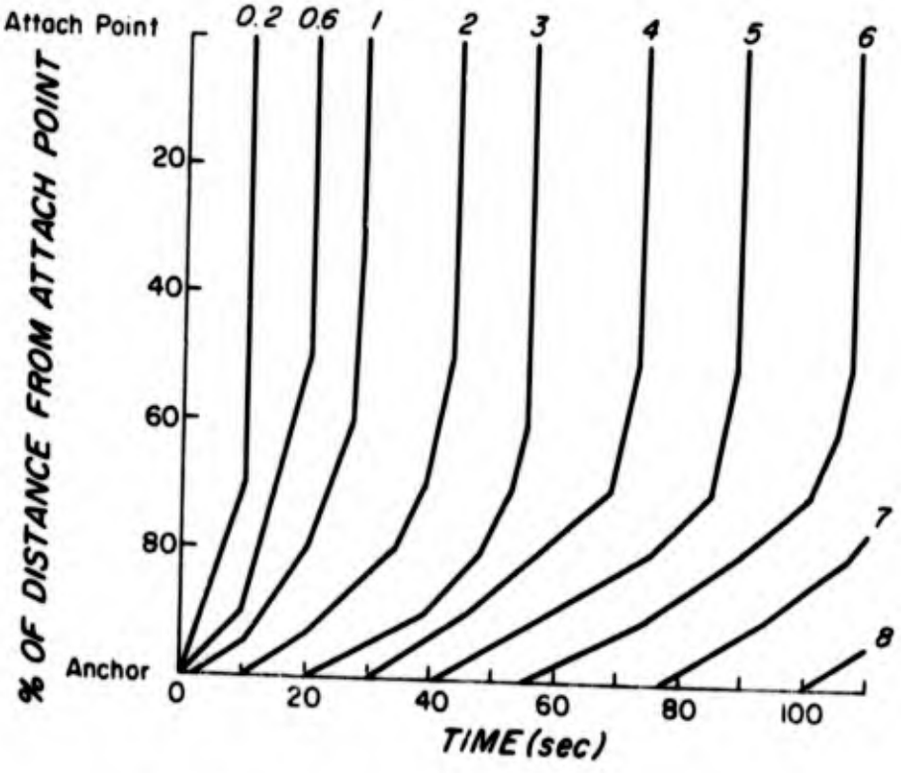


Figure 17. Tension Contours ~ KIP
60° Catenary Dacron Line 1.0" Dia. (NVR08)

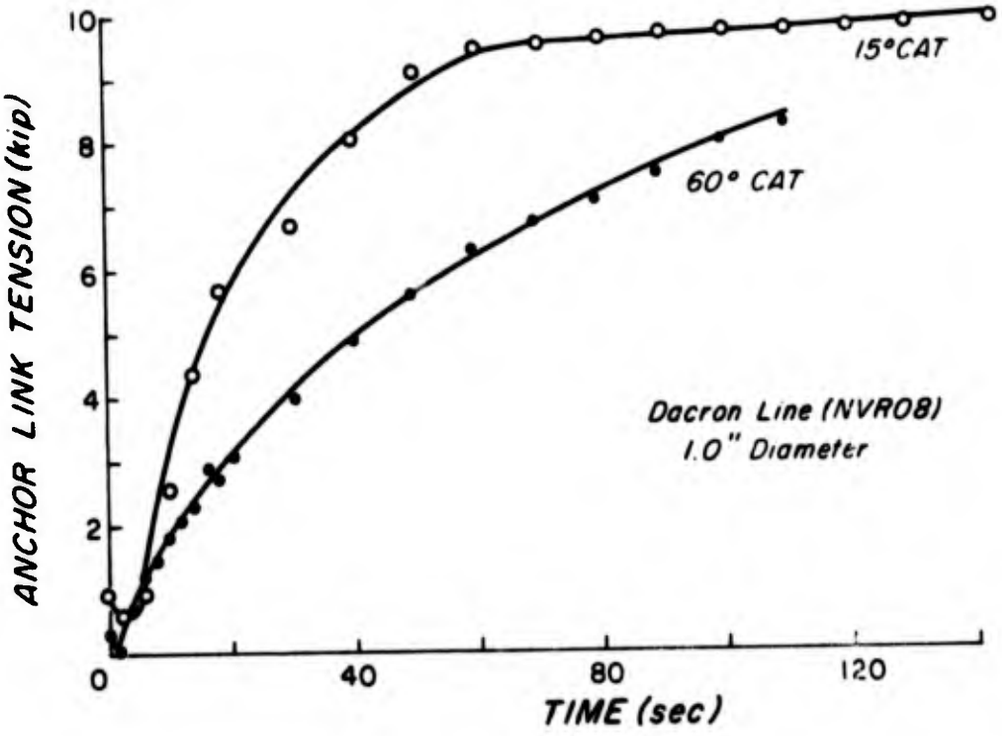


Figure 18. Anchor Link Tension for Dacron Line

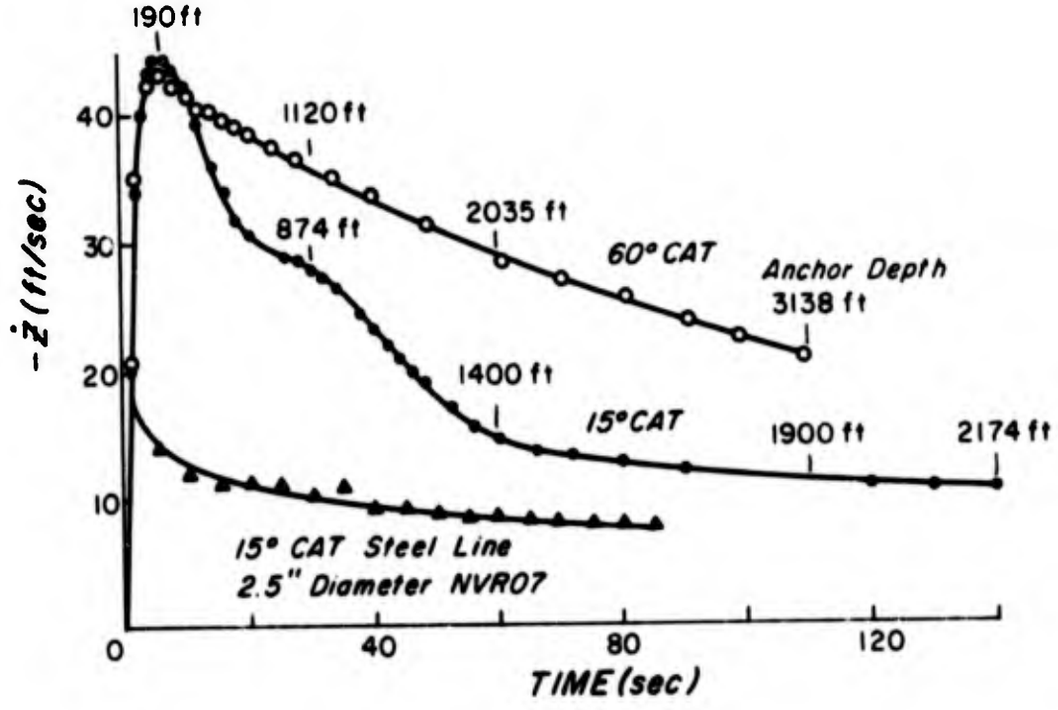


Figure 19. Vertical Component of Anchor Velocity
Dacron Line 1.0" Dia. (NVR08)

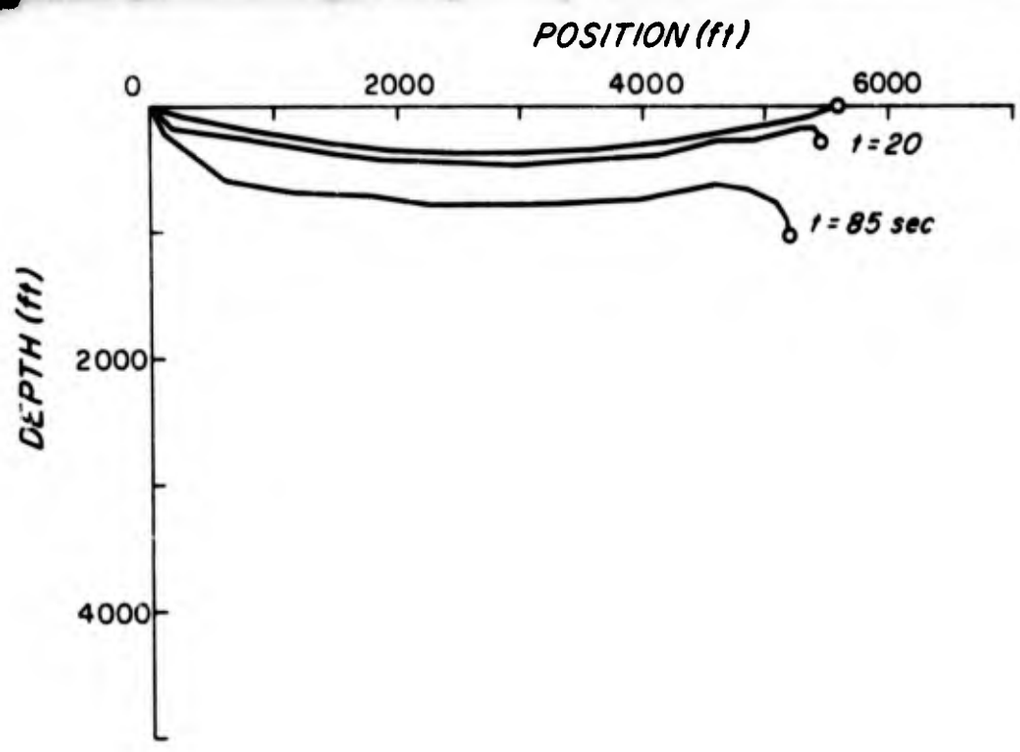


Figure 20. 15° Catenary Steel Line 2.5" Dia. (NVR07)

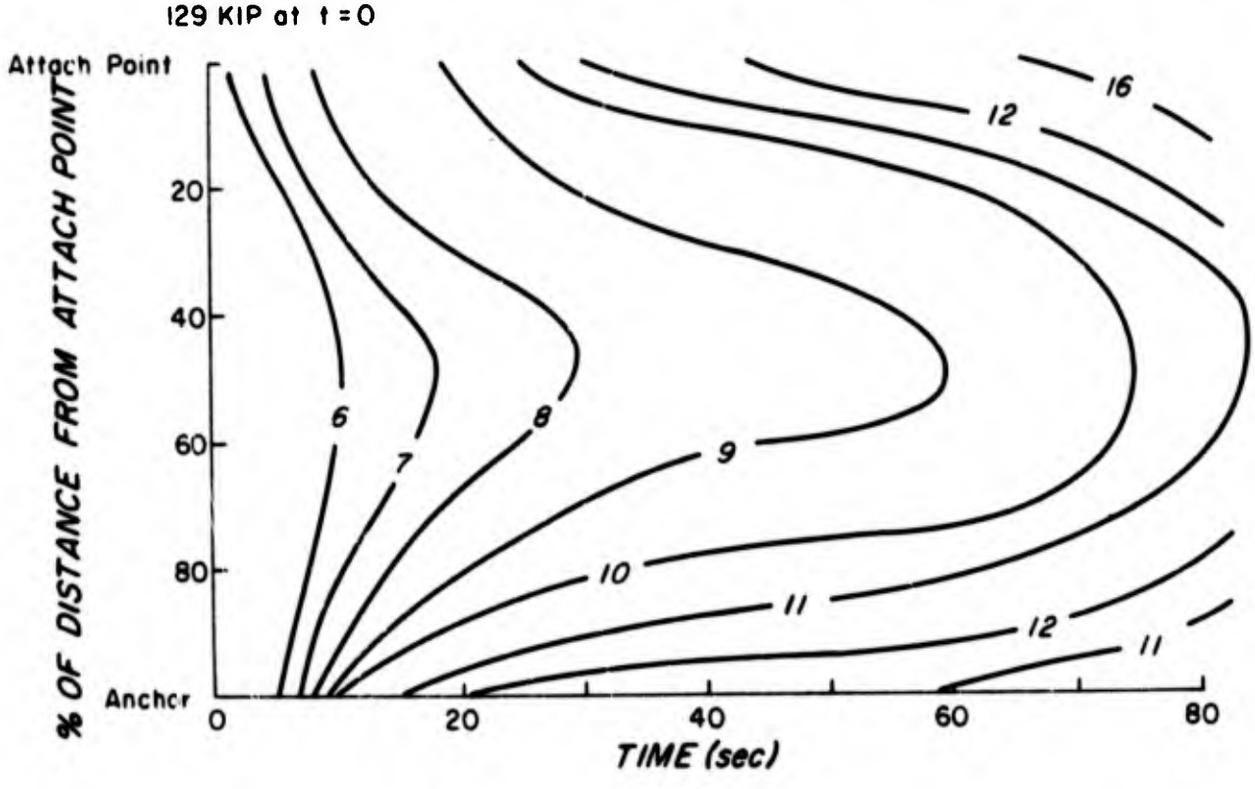


Figure 21. Tension Contours KIP
15° Catenary Steel Line 2.5" Dia. (NVR07)

Appendix A

Listing for Extensible Line

```
      PROGRAM LUMP
      COMMON/TIME/DELTAT,TIME,TIMEMAX
      COMMON/OTPT/PRINTIM,PRINTIV
      COMMON/PNCH/PNCHCOD,READCOD
      COMMON/FLAG/IFLAG,M2,JFLAG
      IFLAG=0
      JFLAG=1
      CALL INPUT
      TIME=0.
      PRINTIM=PRINTIV-.0001
      CALL CARLOIC
      CALL OUTPUT
      CALL LENGTH
  100 IF(TIME.GE.TIMEMAX) GO TO 200
      CALL FND1
      CALL WUNITE
      TIME=TIME+DELTAT
      IF(TIME.GE.PRINTIM)110,20
  110 CALL OUTPUT
      PRINTIM=PRINTIM+PRINTIV
   20 CONTINUE
      GO TO 100
  200 CONTINUE
      IF(PNCHCOD.EQ.1)CALL PUNCHER
      CALL LENGTH
      STOP
      END
```

```
      SUBROUTINE INPUT
      COMMON/CONST/PI,GC
      COMMON/ACCEL/S(101),AREA,D,WATRDEN,DENLINE,T(101),D
      COMMON/STRAS/SCON,SEXP
      COMMON/TIME/DELTAT,TIME,TIMEMAX
      COMMON/FAIL/FAILS
      COMMON/OTPT/PRINTIM,PRINTIV
      COMMON/INPT/DP,NMASS,N1,STOTAL,SFINAL
      COMMON/ERRORS/ERRMAX,ERRMIN
      COMMON/PUNCH/PNCHCOD,READCOD
      COMMON/SAILF/THET
      COMMON/FLAG/IFLAG,N2,JFLAG
      PRINT 101
  101 FORMAT(1H1,50X,32HDISCRETE MASS MOORING LINE MODEL)
      READ 102,STOTAL,DENLINE,D,G,NMASS
  102 FORMAT(3F10.1,F10.3,I10)
      READ 103,DP,WATRDEN
      READ 107,FAILS
  107 FORMAT(F10.0)
  103 FORMAT(2F10.0)
      READ 401,THET
      READ 32,SCON,SEXP
   32 FORMAT(E10.3,F10.3)
      READ 103,TIMEMAX,DELTAT
      READ 103,ERRMAX,ERRMIN
      READ 401,PRINTIV
      READ 401,PNCHCOD
      READ 401,READCOD
  401 FORMAT(F10.0)
      PI=3.1415927
      GC=32.17
      AREA=PI/4.*(D/12.)**2
      D=D/12.
      PRINT 104,STOTAL,DENLINE,D,G,NMASS
  104 FORMAT(///14HLINE LENGTH =F6.1,4H FT./
```

```
     1 15HLINE DENSITY =F7.3,10H LB./FT**3/
     2 21HLINE DIAMETER =F5.3,4H FT. /
     3 15HDAMPING CONSTANT =E10.3,13H LB SEC/FT**2/
     4 14H NO. OF MASSES =I3)
       N1=NMASS+1
       N2=N+1
       PRINT 105,DP,WATDEN,TIMEMAX,DELTAT
       PRINT 412,ERRMAX,ERRMIN
       PRINT 406,THET
   406 FORMAT(28HCATENARY ANGLE AT SURFACE =F5.2,5H DEG.)
       THET=THET*PI/180.
   412 FORMAT(22HVELOCITY ERROR BOUNDS,F8.4,3H TO,F8.4)
   105 FORMAT( 16HWATER DEPTH =F9.2,3H FT /
     1 16HWATER DENSITY =F6.2,9H LB/FT-CU /
     2 11HMAX TIME =F5.2,4H SEC /
     3 12HTIME STEP =F5.3,4H SEC )
       PRINT 108,PRINTIV
   108 FORMAT(13HOPRINT EVERY F5.3,4H SEC)
       PRINT 109,FAILS
   109 FORMAT(20HLINE FAILURE LOAD  =  F11.1,4H LBS)
       PRINT 403,PUNCOD
   403 FORMAT(13H PUNCH CODE =F5.1,5X,56HPUNCH CODE = 1. PUNCHS ALL POS A
     1ND VEL AT TIME = TIMEMAX    )
       PRINT 404, READCOD
   404 FORMAT(13HO READ CODE =F5.1,5X,57H READ CODE = 1.  READS ALL POS A
     1ND VEL FOR TIME = TIMEMAX    )
       PRINT 405,SCON,SEXP
   405 FORMAT(//12X,34HST-VSS-STRAIN RELATION   STRESS = ,E10.3,9H*STRAIN
     1**,F5.1)
C
C      SEGMENT LENGTHS
C
       SL=STOTAL/NMASS
       S(1)=SL/2.
       DO 21 I=2,NMASS
       S(I)=SL
    21 CONTINUE
       S(NMASS+1)=SL/2.
       RETURN
       END
```

```
      SUBROUTINE NUMINTG
      COMMON/TIME/DELTAT,TIME,TIMEMAX
      COMMON/NUMI/XL(101),ZL(101),XD(101),ZD(101),XDD(101),ZDD(101),
     1 XLN(101),ZLN(101),XDN(101),ZDN(101),ERRORX(101),ERRORZ(101)
      COMMON/INPT/DP,WMASS,N1,STOTAL,SFINAL
      COMMON/ACCEL/S(101),AREA,Q,WATRDEN,DENLINE,T(101),D
      COMMON/FAIL/FAILS
      COMMON/ERRORS/ERRMAX,ERRMIN
      COMMON/JUMP/PRINTIM,PRINTIV
      COMMON/OLD/XLOLD(101),ZLOLD(101),XDOLD(101),ZDOLD(101),XDDOLD(101)
     1,ZDDOLD(101)
      COMMON/FLAG/IFLAG,N2,JFLAG
      DIMENSION XDDN(101),ZDDN(101)
      DIMENSION AKX1(101),AKZ1(101),AKX2(101),AKZ2(101),AKX3(101),AKZ3(1
     101),QX1(101),QZ1(101),QX2(101),QZ2(101),QX3(101),QZ3(101)

C
C     DEFINE VELOCITY AND POSITION AT TIME T+DELTAT
C
    1 CONTINUE
      IF(IFLAG.EQ.0) GO TO 1000
      DO 100 I=2,N2
      IF(I.EQ.N2) GO TO 73
      CALL ACCEL(I,XL(I+1),XL(I),XL(I-1),ZL(I+1),ZL(I),ZL(I-1),XD(I+1),
     1 XD(I),XD(I-1),ZD(I+1),ZD(I),ZD(I-1),XDD(I),ZDD(I))
      GO TO 74
   73 CALL ENDP(XL(N2),XL(N1),ZL(N2),ZL(N1),XD(N2),ZD(N2),S(N1),AREA,
     1 XDD(N2),ZDD(N2))
      IF(I.EQ.N2.AND.ZL(N2).LT.-DP)XDOLD(I)=0.
      IF(I.EQ.N2.AND.ZL(N2).LT.-DP)ZDOLD(I)=0.
   74 CONTINUE
C
      XDN(I)=XDOLD(I)+DELTAT*2.*XDD(I)
      ZDN(I)=ZDOLD(I)+DELTAT*2.*ZDD(I)

C
      XLN(I)=XL(I)+DELTAT*(XD(I)+XDN(I))/2.
      ZLN(I)=ZL(I)+DELTAT*(ZD(I)+ZDN(I))/2.
```

```
C
      ERRORX(I)=-XDN(I)
      ERRORZ(I)=-ZDN(I)
  100 CONTINUE
C     CORRECTOR
C
      XM=XLN(1) $ZM=ZLN(1) $XDM=XDN(1) $ZDM=ZDN(1)
      DO 200 I=2,N2
      IF(I.EQ.N2) GO TO 75
      CALL MODEL(I,XLN(I+1),XLN(I),XM        ,ZLN(I+1),ZLN(I),ZM        ,
     1 XDN(I+1),XDN(I),XDM        ,ZDN(I+1),ZDN(I),ZDM        ,DDX,DDZ)
      GO TO 76
   75 CALL ENDE(XLN(N2),XLN(N1),ZLN(N2),ZLN(N1),XDN(N2),ZDN(N2),S(N1),
     1 AREA,DDX,DDZ)
   76 CONTINUE
C
      XM=XLN(I) $ZM=ZLN(I) $XDM=XDN(I) $ZDM=ZDN(I)
      XDN(I)=XD(I)+DELTAT*(XDD(I)+DDX)/2.
      ZDN(I)=ZD(I)+DELTAT*(ZDD(I)+DDZ)/2.
C
      XLN(I)=XL(I)+DELTAT*(XD(I)+XDN(I))/2.
      ZLN(I)=ZL(I)+DELTAT*(ZD(I)+ZDN(I))/2.
C
      ERRORX(I)=(ERRORX(I)+XDN(I))/(5.*XDN(I))
      ERRORZ(I)=(ERRORZ(I)+ZDN(I))/(5.*ZDN(I))
      IF(I.EQ.N2.AND.ZL(N2).LT.-DF)ERRORX(N2)=0.
      IF(I.EQ.N2.AND.ZL(N2).LT.-DF)ERRORZ(N2)=0.
      ERRX=ERRMAX
      ERRZ=ERRMAX
      IF(ABS(XDN(I)).LT..1) ERRX=10.
      IF(ABS(XDN(I)).LT.1.)ERRX=1.
      IF(ABS(ERRORX(I)).GT.ERRX  .OR.ABS(ERRORZ(I)).GT.ERRZ  )140,200
  140 DELTAT=DELTAT/2.
      IFLAG=1
      PRINT 900,DELTAT,TIME,I,ERRORX(I),ERRORZ(I)
```

```
      FORMAT(13H0DELTAT REDUCE) TO F7.4,3H AT,F8.3,4H SEC,3X,I5,2(E10.3,
     1 2X ))
       IF(DELTAT.LT..0001)155,156
  155  PRINT 939
  939  FORMAT(23H. ERROR DELTAT LT .0001 )
       STOP
  156  GO TO 1
  350  CONTINUE
       NOUNTER=
       DO 351 I=1,N2
       XLOLD(I)=XL(I)
       ZLOLD(I)=ZL(I)
       XDOLD(I)=XD(I)
       ZDOLD(I)=ZD(I)
       XL(I)=XLN(I)
       ZL(I)=ZLN(I)
       XD(I)=XDN(I)
       ZD(I)=ZDN(I)
       IF(ABS(ERRORX(I)).LT.ERRMIN.AND.ABS(ERRORZ(I)).LT.ERRMIN)NOUNTER
     1 =NOUNTER+1
  351  CONTINUE
       IF(NOUNTER.EQ.N2)352,360
  352  DELTAT=DELTAT*2.
       IFLAG=0
       DO 407 I=1,N2
       XLOLD(I)=XLN(I)
       ZLOLD(I)=ZLN(I)
       XDOLD(I)=XDN(I)
  407  ZDOLD(I)=ZDN(I)
       PRINT 910,DELTAT,TIME
  910  FORMAT(21H0DELTAT INCREASED TO F7.4,3H AT,F8.3,4H SEC)
  360  CONTINUE
       GO TO 2000
 1000  CONTINUE
```

```
C
C      RUNGE-KUTTA (THIRD ORDER)
C
       DO 1100 I=1,N2
       IF(T.EQ.0.0) GO TO 1105
       CALL ENDPT(I,XLOLD(I+1),XLOLD(I),XLOLD(I-1),ZLOLD(I+1),ZLOLD(I),ZL
      1OLD(I-1),XDOLD(I+1),XDOLD(I),XDOLD(I-1),ZDOLD(I+1),ZDOLD(I),ZDOLD(
      1I-1),XDOLD(I),ZDOLD(I))
       GO TO 1100
 1105  CALL ENDPT(XLOLD(N2),XLOLD(N1),ZLOLD(N2),ZLOLD(N1),XDOLD(N2),ZDOLD(
      1N2),S(1),ERLN,XDOLD(N2),ZDOLD(N2))
 1100  CONTINUE
       AKX1(I)=DELTAT*XDOLD(I)
       AKZ1(I)=DELTAT*ZDOLD(I)
       AX1(I)=DELTAT*XDOLD(I)
       AZ1(I)=DELTAT*ZDOLD(I)
       XLN(I)=XLOLD(I)+AKX1(I)/2.
       ZLN(I)=ZLOLD(I)+AKZ1(I)/2.
       XLN1(I)=XLOLD(I)+AX1(I)/2.
       ZLN1(I)=ZLOLD(I)+AZ1(I)/2.
 1101  CONTINUE
       DO 1200 I=2,N2
       IF(T.EQ.0.0) GO TO 1205
       CALL ENDPT(I,XLN(I+1),XLN(I),XLN(I-1),ZLN(I+1),ZLN(I),ZLN(I-1),XDN
      1(I+1),XDN(I),XDN(I-1),ZDN(I+1),ZDN(I),ZDN(I-1),XDDN(I),ZDDN(I))
       GO TO 1200
 1205  CALL ENDPT(XLN(N2),XLN(N1),ZLN(N2),ZLN(N1),XDN(N2),ZDN(N2),S(N1),
      1ERLN,XDDN(N2),ZDDN(N2))
 1200  CONTINUE
```

```
      AKX2(I)=DELTAT*XDDN(I)
      AKZ2(I)=DELTAT*ZDDN(I)
      QX2(I)=DELTAT*XDN(I)
      QZ2(I)=DELTAT*ZDN(I)
 1200 CONTINUE
      DO 1300 I=2,N2
      XDN(I)=XDOLD(I)-AKX1(I)+2.*AKX2(I)
      ZDN(I)=ZDOLD(I)-AKZ1(I)+2.*AKZ2(I)
      XLN(I)=XLOLD(I)-QX1(I)+2.*QX2(I)
      ZLN(I)=ZLOLD(I)-QZ1(I)+2.*QZ2(I)
 1300 CONTINUE
      DO 1400 I=2,N2
      IF(I.LE.N2) GO TO 1305
      CALL ADDEL(I,XLN(I+1),XLN(I),XLN(I-1),ZLN(I+1),ZLN(I),ZLN(I-1),XDN
     1(I+1),XDN(I),XDN(I-1),ZDN(I+1),ZDN(I),ZDN(I-1),XDDN(I),ZDDN(I))
      GO TO 1306
 1305 CALL ENDE(XLN(N2),XLN(N1),ZLN(N2),ZLN(N1),XDN(N2),ZDN(N2),S(N1),
     1 AREA,XDDN(N2),ZDDN(N2))
 1306 CONTINUE
      AKX3(I)=DELTAT*XDDN(I)
      AKZ3(I)=DELTAT*ZDDN(I)
      QX3(I)=DELTAT*XDN(I)
      QZ3(I)=DELTAT*ZDN(I)
 1400 CONTINUE
      DO 1500 I=2,N2
      XD(I)=XDOLD(I)+(AKX1(I)+4.*AKX2(I)+AKX3(I))/6.0
      ZD(I)=ZDOLD(I)+(AKZ1(I)+4.*AKZ2(I)+AKZ3(I))/6.0
      XL(I)=XLOLD(I)+(QX1(I)+4.*QX2(I)+QX3(I))/6.0
      ZL(I)=ZLOLD(I)+(QZ1(I)+QZ2(I)*4.+QZ3(I))/6.0
 1500 CONTINUE
      IFLAG=1
 2000 CONTINUE
      DO 390 I=2,N1
      CALL TENSION(I,XL(I+1),XL(I),XL(I-1),ZL(I+1),ZL(I),ZL(I-1))
  390 CONTINUE
      DO 400 I=1,N1
      IF(T(I).GT.FAILS)410,400
  410 CALL OUTPUT
      SUMY=RUPT(1.)
  400 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PUNCHER
      COMMON/TIME/DELTAT,TIME,TIMEMAX
      COMMON/ACCEL/S(101),ARLA,Q,WATRDEN,DENLINE,T(101),D
      COMMON/INPT/DP,NMAPS,NL,STOTAL,SFINAL
      COMMON/WORM/XL(101),ZL(101),XO(101),ZO(101),XDO(101),ZDO(101),
     1 XLM(101),ZLM(101),XDN(101),ZDN(101),ERRORX(101),ERRORZ(101)
      NZERO=2
      PUNCH 100,DELTAT,TIME
      DO 20  I=1,NZ
      PUNCH 101,XO(I),ZL(I),XO(I),ZO(I)
   20 CONTINUE
  100 FORMAT(2F10.4)
  101 FORMAT(2F10.2,2F10.5)
      RETURN
      END
```

```
      SUBROUTINE ACCEL(I,XP,X,XM,ZP,Z,ZM,XDP,XD,XDM,ZDP,ZD,ZDM,XDD,ZDD
     1)
      COMMON/ACCEL/S(101),AREA,Q,WATDEN,DENLINE,T(101),D
      COMMON/INPT/DP,NMASS,NI,STOTAL,SFINAL
      COMMON/CONST/PI,GC
      COMMON/TIME/DELTAT,TIME,TIMEMAX
      COMMON/OTPT/PRINTIM,PRINTIV
      THETA=ATAN2((ZP-Z),(XP-X))
      THETAM=ATAN2((Z-ZM),(X-XM))
C
C     SPRING FORCE ON MASS I
C
      EPM=SQRT((X-XM)**2+(Z-ZM)**2)/S(I-1)-1.
      EP=SQRT((XP-X)**2+(ZP-Z)**2)/S(I)-1.
      SFORCEX=AREA*(STRASS(EP)*COS(THETA)-STRASS(EPM)*COS(THETAM))
      SFORCEZ=AREA*(STRASS(EP)*SIN(THETA)-STRASS(EPM)*SIN(THETAM))
C
C     WATER DAMPING FORCES ON MASS I
      THAV=(THETA+THETAM)/2.
C
C     VX AND VZ ARE LOCAL VELOCITIES
```

```
C
      VX=0.
      VZ=0.
      VTANG=(XD-VX)*COS(THAV)+(ZD-VZ)*SIN(THAV)
      VNORM=-(XD-VX)*SIN(THAV)+(ZD-VZ)*COS(THAV)
      SL=(S(I)+S(I-1))/2.
      IF(I.EQ.2)SL=S(I-1)+S(I)/2.
      IF(I.EQ.N1)SL=S(I-1)/2.+S(I)
      ARET=PI*D*SL
      AREN=D*SL
      CT=[illegible]
      CN=[illegible]
      DRAGT=-CT*WATRDEN*ARET*VTANG*ABS(VTANG)/(2.*GC)
      DRAGN=-CN*WATRDEN*AREN*VNORM*ABS(VNORM)/(2.*GC)
C
C     VIRTUAL MASS CORRECTION
C
      SMASS=SL*DENLINE*AREA/GC
      AMASS=WATRDEN*AREA*SL/GC
      CIT=0.
      CIN=1.
      WT=(DENLINE-WATRDEN)*SL*AREA
      ACCTANG=((SFORCEX         )*COS(THAV)+(SFORCEZ         -WT)*
     1SIN(THAV)+DRAGT)/(SMASS+CIT*AMASS)
      ACCNORM=(-(SFORCEX         )*SIN(THAV)+(SFORCEZ         -WT)*
     1COS(THAV)+DRAGN)/(SMASS+CIN*AMASS)
      XDD=ACCTANG*COS(THAV)-ACCNORM*SIN(THAV)
      ZDD=ACCTANG*SIN(THAV)+ACCNORM*COS(THAV)
      RETURN
      END
```

```
      SUBROUTINE ENDO2(X,XM,Z,ZM,XD,ZD,S,AR,XDD,ZDD)
      COMMON/FLAG/IFLAG,PI,JFLAG
      COMMON/INPT/DB
      REAL I
      IF(JFLAG.EQ.1) GO TO 95
      BWEIGHT=120000.
      VOL=24.5
      CD=.4
      CL=1.
      AREA=12.5
      WDEN=54.
      RHO=2.
      G=32.2
      WB=BWEIGHT-WDEN*VOL
      PI=3.14159265
      M=BWEIGHT/G+CL*VOL*RHO
      A=CD*AREA*RHO/(2.*M)
      JFLAG=1
   95 CONTINUE
      IF(Z.LT.-DB)50,50
   50 XD=ZD=XDD=ZDD=0.
      RETURN
   51 EP=SQRT((X-XM)**2+(Z-ZM)**2)/S-1.
      T=AR*STR*ABS(EP)
      THETA=ATAN2(Z-ZM,X-XM)+PI
      FX=T*COS(THETA)
      FZ=T*SIN(THETA)-WB
      V=SQRT(XD**2+ZD**2)
      XDD=FX/M-A*XD*V
      ZDD=FZ/M-A*ZD*V
      RETURN
      END
```

```
      FUNCTION STRASS(X)
      COMMON/STRAS/SCON,SEXP
C           GIVEN STRAIN X,  FIND STRESS
      IF(X.LT.0.)10,20
   10 STRASS=0.
      RETURN
   20 CONTINUE
      STRASS=SCON*X**SEXP
      RETURN
      ENTRY STRAIN
      STRAIN=(X/SCON)**(1./SEXP)
      RETURN
      ENTRY PUNT
      PRINT  6
    6 FORMAT(1X,  27H LINE HAS BEEN RUPTURED)
      STOP
      END
```

```
      SUBROUTINE END1
      COMMON/TIME/DELTAT,TIME,TIMEMAX
      COMMON/NUMI/XL(101),ZL(101),XD(101),ZD(101),XDD(101),ZDD(101),
     1 XLN(101),ZLN(101),XDN(101),ZDN(101),ERRORX(101),ERRORZ(101)
      COMMON/OLD/XLOLD(101),ZLOLD(101),XDOLD(101),ZDOLD(101),XDDOLD(101)
     1,ZDDOLD(101)
C     SET BOUNDARY CONDITIONS AT END 1
C
      XLN(1)=0.   ZLN(1)=0.   XDN(1)=0.   ZDN(1)=0.
      XLOLD(1)=XDOLD(1)=ZLOLD(1)=ZDOLD(1)=0.0
      ERRORX(1)=0.   ERRORZ(1)=0.
      RETURN
      END
      SUBROUTINE LENGTH
      COMMON/NUMI/XL(101),ZL(101),XD(101),ZD(101),XDD(101),ZDD(101),
     1 XLN(101),ZLN(101),XDN(101),ZDN(101),ERRORX(101),ERRORZ(101)
      COMMON/INPT/DP,WMASS,N1,STOTAL,SFINAL
C
C     STRAINED LENGTH OF CABLE
C
      SFINAL=0.
      N2=N1+1
      DO 110 I=2,N2
      SFINAL =SFINAL+SQRT((XL(I)-XL(I-1))**2+(ZL(I)-ZL(I-1))**2)
  110 CONTINUE
      PRINT 111,SFINAL
  111 FORMAT(//1H0,23X,13HLINE LENGTH =,F8.1)
      RETURN
      END
```

```
      SUBROUTINE OUTPUT
      COMMON/TIME/DELTAT,TIME,TIMEMAX
      COMMON/NUMI/XL(101),ZL(101),XD(101),ZD(101),XDD(101),ZDD(101),
     1 XLN(101),ZLN(101),XDN(101),ZDN(101),ERRORX(101),ERRORZ(101)
      COMMON/ACCEL/S(101),AREA,Q,WATRDEN,DENLINE,T(101)
      COMMON/INPT/DP,NMASS,N1,STOTAL,SFINAL
      PRINT 100 ,TIME
  100 FORMAT(1H1,//,7H0TIME =F6.2)
      PRINT 101
  101 FORMAT(////1H0,5X,6H LINK ,6H MASS ,5X,13H COORDINATES ,10X,7HTENS
     1ION,10X,10HVELOCITIES,14X,6HERRORS)
      PRINT 102
  102 FORMAT(1H ,7X,3HNO.,3X,3HNO.,9X,1HX,9X,1HZ,29X,1HX,10X,1HZ,9X,
     1 4HXDOT, 4X,4HZDOT)
      PRINT 103,(I,XL(I),ZL(I),XD(I),ZD(I),ERRORX(I),ERRORZ(I),I,T(I),
     1 I=1,N1)
  103 FORMAT(13X,I3,2X,2(1X,F8.2),17X,2(1X,F10.4),2(2X,E10.3)/7X,I3,31X,
     1 E11.4)
      N2=NMASS+2
      PRINT 103,N2,XL(N2),ZL(N2),XD(N2),ZD(N2),ERRORX(N2),ERRORZ(N2)
      RETURN
      END
```

```
      SUBROUTINE CABLEIC
      COMMON/SANGLE/THET
      COMMON/OUTPT/PRINTIM,PRINTIV
      COMMON/FLOW/FLOWLOC,READCOD
      COMMON/TIME/DELTAT,TIME,TIMEMAX
      COMMON/CONST/PI,GC
      COMMON/ACCEL/S(101),AREA,Q,WATRDEN,DENLINE,T(101),D
      COMMON/INPT/DF,AMASS,N1,STOTAL,SFINAL
      COMMON/NUMI/XL(101),ZL(101),XD(101),ZD(101),XDD(101),ZDD(101),
     1 XLN(101),ZLN(101),XDN(101),ZDN(101),ERRORX(101),ERRORZ(101)
      COMMON/OLD/XLOLD(101),ZLOLD(101),XDOLD(101),ZDOLD(101),XDDOLD(101)
     1,ZDDOLD(101)
      N2=N1+1
      IF(READCOD.EQ.1.) GO TO 500
      WL=(DENLINE-WATRDEN)*AREA
      TANG=SIN(THET)/COS(THET)
      SL=STOTAL
      DO 2 L=1,3
      TO=WL*SL     /(2.*TANG)
      FL=TO*2./WL*SINHI(TANG)
      SL=0.
      DO 200 J=1,N2
      IF(J.NE.1)SL=SL+S(J-1)
      XL(J)=FL/2.+TO/WL*SINHI(SL*WL/TO-SINH(WL*FL/(2.*TO)))
      ZL(J)=TO/WL*(COSH(WL/TO*(XL(J)-FL/2.))-COSH(WL*FL/(2.*TO)))
      T2=TO*COSH(WL/TO*(XL(J)-FL/2.))
      ERR=STRAIN(TO/AREA)
      IF(J.LE.1)GO TO 50
      T(J-1)=(T1+T2)/2.
      DELTAL=(ERR1+ERR2)*S(J-1)/2.
      ANG=ATAN2(ZL(J)-ZL(J-1),XL(J)-XL(J-1))
      XL(J)=XL(J)+DELTAL*COS(ANG)
      ZL(J)=ZL(J)+DELTAL*SIN(ANG)
      SL=SL+DELTAL
   50 CONTINUE
```

```
      T1=T2
      DP1=DP2
      ZDOLD(J)=0.
      XDOLD(J)=0.
      ZD(J)=0.
      XD(J)=0.
      XLN(J)=XL(J)   $ZLN(J)=ZL(J)
      XLOLD(J)=XL(J)
      ZLOLD(J)=ZL(J)
      XDN(J)=0.   $ZDN(J)=0.
      ERRORX(J)=0.   $ERRORZ(J)=0.
  230 CONTINUE
  202 CONTINUE
      RETURN
  500 CONTINUE
      READ 501,DELTAT,TIME
      PRINTIM=TIME+PRINTIV
      DO 550 I=1,N2
      READ 601,XL(I),ZL(I),XD(I),ZD(I)
      T(I)=T.
      XLN(I)=XL(I)   $ZLN(I)=ZL(I)
      XDN(I)=XD(I)   $ZDN(I)=ZD(I)
      XLOLD(I)=XL(I)
      ZLOLD(I)=ZL(I)
      XDOLD(I)=XD(I)
      ZDOLD(I)=ZD(I)
      ERRORX(I)=0.   $ERRORZ(I)=0.
  550 CONTINUE
      READ 502,TIMEMAX
  502 FORMAT(2F10.3)
  601 FORMAT(4F10.4,2E12.5)
      RETURN
      END
```

```
      FUNCTION ASINH(P)
      ASINH=ALOG(P+SQRT(P**2+1.))
      RETURN
      END
      FUNCTION AVP(X)
      COMMON XI,YM
      AVP=.5*(XI(X)-XI(-X))
      RETURN
      ENTRY AVM
      AVM=.5*(XI(X)+XI(-X))
      RETURN
      END
```

```
      SUBROUTINE TISSIN(I,XP,X,XM,ZP,Z,ZM)
      COMMON/ACTEL/D(131),AREA,Q,WATDEN,DENLIN,T(131),D
      S(I)=T((XP-X)**2+(ZP-Z)**2)/S(I)-1.
      IF(I.NE.2) GO TO 1
      S(I-1)=T((XM-X)**2+(ZM-Z)**2)/S(I-1)-1.
      T(I-1)=DTRISS(S(I-1))*AREA
    1 CONTINUE
      T(I)=DTRISS(S(I))*AREA
      RETURN
      END
```

Appendix B

Listing for Inextensible Line

```
      PROGRAM CABLE1
      DIMENSION THETA(30),TO(30),THETA2(30),TO2(30),TOO(30),TOO2(30),TO1
     C(30),THETA1(30)
      DIMENSION TOO1(30)
      DIMENSION VARBOP(30), PERROR(30)
      COMMON/ANCHMASS/ANCHOR
      COMMON/POS/X(30),Y(30)/TEN/TEN(30)/VEL/VX(30),VY(30)/BS/XO,YO,XO,Y
     CO,XO,YOO,DENSITY,YC
      COMMON/DRAG2/DRAG(30)/G/GRAV(30)
      COMMON/T/TIME,DELTAT,DELTATN
      XO=YO=XO=YO=XOO=YOO=0.
      READ 5,TIMEOUT,TPRINT,CI,CN,CT,KORRECT
    5 FORMAT(F6.2,1X,F5.2,1X,F5.4,1X,F4.2,1X,F5.4,1X,I2)
      READ 6,DELTAT,EUP,ELOW
    6 FORMAT(F6.4,1X,F6.4,1X,F11.9)
      CALL INPUT(N,DIA,DENSITY,XC,YC,TOTAL,ANCHOR)
      CALL ADDMASS(N,DIA,CI)
      CALL TN(N,DIA,CN,CT,ANCHOR)
      PRINT 11,DIA,DENSITY,TOTAL,ANCHOR,XC,YC,N
   11 FORMAT(///,11H LINE DIA= ,F6.3,/,15H LINE DENSITY= ,F5.1,/,14H LIN
     CE LENGTH= ,F6.0,/,14H ANCHOR MASS= ,F7.0,/,7H SPAN= ,F5.0,/,14H WA
     CER DEPTH= ,F5.0,/,18H NO. OF SEGMENTS= ,I2)
      PRINT 15,TIMEOUT,TPRINT,CI,CN,CT,KORRECT
   15 FORMAT(10H TIMEOUT= ,F6.2,/,18H PRINT INCREMENT= ,F5.2,/,18H ADDED
     C MASS COEF= ,F4.2,/,17H DRAG COEF,/,5X,8HNORMAL= ,F4.2,/,5X,12HTAN
     CGENTIAL= ,F5.4,/,21H, NO OF CORRECTIONS= ,I2)
      PRINT 18,EUP,ELOW
   18 FORMAT(19H ERROR LIMITS,/,5X,7HUPPER= ,F7.4,/,5X,7HLOWER= ,F12.9,/
     C////)
      CALL INITIAL(N,XC,THETA1,TO1)
      CALL ANGLES(N,THETA1)
      CALL COORD(N,XO,YO,XO,YO,TO1)
      PRINT 20
```

```
2     FORMAT(32X,14HINITIAL CONDITIONS,///,16X,10HLINK  MASS,8X,1HX,11X,
     C1HY,4X,7HTENSION,7),5HTHETA,8X,9HTHETA DOT,//)
      A=180./3.1459
      DO 4 I=1,N
      THETA(I)=THETA1(I)*A
      TD2(I)=TD1(I)*A
      PRINT 3,I,THETA2(I),TD2(I),I,X(I),Y(I)
3     FORMAT(11X,I2,5X,F7.2,3X,F8.2,/,19X,I2,5X,F7.2,5X,F7.2)
4     CONTINUE
      PRINT 5,DELTAT


5     FORMAT(/////,10H DELTA T= ,F8.4,/////)
      TIME=TOM+DT
      DELTAT=DF_TAT
      JFLAG=0
      KFLAG=0
      NFLAG=0
      KOMPOK=0
60    IF(KFLAG.EQ.1) GO TO 62
      IF(Y(N).LT.Y0) GO TO 62
      NFLAG=1
      GO TO 1FF
63    CALL BOTTOM(N,TD,TIME)
      KFLAG=1
      NFLAG=0
      GO TO 1FF
62    IF(JFLAG.EQ.0) GO TO 70
      CALL CALL(N,THETA,TD)
      CALL SOLVE(N,TDD,KUKRFOT)
      CALL PREDICT(N,DELTAT,TD1,THETA,TD,TDD,THETA2,TD2)
      DO 61 I=1,N
      THETA1(I)=THETA(I)
```

```
      TD1(I)=TD(I)
      TDD1(I)=TDD(I)
61    CONTINUE
      CALL CALL(N,THETA2,TD2)
      CALL SOLVE(N,TDD2,KORRECT)
      CALL CORRECT(N,DELTAT,THETA1,TD1,TDD1,TDD2,THETA,TD)
      DO 65 I=1,N
      VERROR(I)=(TD2(I)-TD(I))/(5.*TD(I))
      PERROR(I)=(THETA2(I)-THETA(I))/(THETA(I)*5.)
65    CONTINUE
      DO 112 I=1,N
      IF(ABS(VERROR(I)).LE.EUP) GO TO 112
      IF(DELTAT.GE.0.0001) GO TO 105
      PRINT 106,DELTAT,TIME,I
106   FORMAT(17H DELTA T HELD TO ,F9.7,10H AT TIME= ,F11.7,17H BECAUSE O
     1F LINK ,I2)
      GO TO 113
104   DELTATN=DELTAT/2.
      TIME=TIME-DELTAT
      TCHECK=TCHECK-DELTAT
      DELTAT=DELTATN
      PRINT 103,DELTAT,TIME
103   FORMAT(22H DELTA T DECREASED TO ,F9.7,10H AT TIME= ,F11.7)
      KCHECK=KCHECK+1
      IF(KCHECK.GE.5) CALL EXIT
      DELAT=
      GO TO 61
112   CONTINUE
      DO 111 I=1,N
      IF(ABS(VERROR(I)).GE.ELOW) GO TO 11
      DELTATN=DELTAT*2.
      PRINT 80,DELTATN,TIME
80    FORMAT(22H DELTA T INCREASED TO ,F9.7,10H AT TIME= ,F11.7)
```

```
      KCHECK=KCHECK+1
      IF(KCHECK.GE.8) CALL EXIT
      JFLAG=0
      GO TO 115
  110 CONTINUE
      GO TO 115
   70 CONTINUE
      CALL PR(N,DELTAT,THETA1,TD1,TDD,THETA,TD,KORRECT)
      JFLAG=1



  115 TIME=TIME+DELTAT
      TCHECK=TCHECK+DELTAT
      DELTAT=DELTATA
      KCHECK=0
      IF (TCHECK.GE.TPRINT) GO TO 130
      GO TO 60
  130 TCHECK=TCHECK-TPRINT
  135 PRINT 140,TIME
  140 FORMAT(////,7H TIME= ,F11.7,////,15X,11HLINK  MASS,6X,11HCOORDINAT
     ES,11X,8HVELOCITY,7X,12HTHETA  THETA,7X,7HTENSION,13X,5HERROR,//,2
     07X,1HX,8X,1HY,8X,1HX,7X,1HY,15X,3HDOT,23X,5HTHETA,4X,9HTHETA DOT,/
     0/)
      CALL ANGLES(N,THETA)
      CALL COORD(N,XC,YC,XD,YD,TD)
      CALL TENSION(N,YDD,YDD,THETA,TD,TDD,DENSITY,XC,YD,YD)
      A=180./3.14159
      DO 150 I=1,N
      THETAC(I)=THETA(I)*A
      TDD(I)=TD(I)*A
```

```
      PRINT 150,I,THETA2(I),T02(I),TEN(I),PERROR(I),VERROR(I),I,X(I),Y(I
     C),VX(I),VY(I)
150   FORMAT(11X,I2,4X,F6.2,2X,F6.2,4X,F11.4,4X,F8.5,3X,F11.8,/,17X,I2,
     *4X,F7.2,2X,F7.2,4X,F6.2,2X,F6.2,/)
160   CONTINUE
      PRINT 155
155   FORMAT(/////)
      IF(NFLAG.EQ.1) GO TO 65
      IF(TIME.GE.TIMECUT) GO TO 170
      GO TO 60
170   CONTINUE
      CALL EXIT
      END
```

```
      SUBROUTINE INPUT(N,DIA,DENSITY,X,Y,TOTAL,ANCHOR)
      REAL L,M
      COMMON/LENGTH/L(30)/MASS/M(30)/CURRENT/VC(30)/TOTMAS/TM(30)
      READ 10,   N,ANCHOR,DIA,DENSITY,TOTAL,X0,Y0
10    FORMAT(I2,1X,F7.1,1X,F4.1,1X,F5.1,1X,F5.1,1X,F5.0,1X,F5.0)
      DENSITY=DENSITY*3.14159*(DIA/24.)**2
      DO 20 I=1,N
      READ 30,   L(I),VC(I)
20    FORMAT(F5.1,1X,F5.0)
30    CONTINUE
      M(1)=.5*(L(1)+.5*L(2))
      N1=N-1
      DO 75 I=2,N1
      M(I)=.5*(.5*(L(I)+L(I+1))
75    CONTINUE
      M(N)=.5*.5*L(N)+ANCHOR
      TM(1)=M(N)
      DO 40 I=2,N
      K=N+1-I
      TM(K)=TM(K+1)+M(K)
40    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ADDMASS(N,DIA,CI)
      REAL L
      COMMON/LENGTH/L(30)/ADMAS/AM(30)
      A=CI*64.*3.14159*(DIA/24.)**2
      DO 10 I=1,N
      AM(I)=A*L(I)
   10 CONTINUE
      RETURN

      END
```

```
      SUBROUTINE [illegible](N,DIA,CN,CT,ANCHOR)
      REAL L
      COMMON/[illegible]/L(30)/COEF/DT(30),DN(30)
      COMMON/[illegible]/DA
      WDEN=64.
      B=CT*WDEN*3.14159*DIA/12.
      A=CN*WDEN*DIA/12.
      DO 10 I=1,N
      DN(I)=A*L(I)
      DT(I)=B*L(I)
   10 CONTINUE
      DEN1=.284*12.**3
      VANCHOR=ANCHOR/DEN1
      RANCHOR=(3.*VANCHOR/(4.*3.14159265))**(1./3.)
      AANCHOR=3.14159265*RANCHOR**2
      DA=.5*WDEN*AANCHOR/2.
      RETURN
      END
```

```
      SUBROUTINE INITIAL(N,XO,THETA,TO)
      DIMENSION THETA(30),TO(30),TX(30),TY(30)
      REAL L,M
      COMMON/LENGTH/L(30)/MASS/M(30)/TEN/TEN(30)
      TMASS=0.0
      N1=N-1
      DO 10 I=1,N1
      TMASS=TMASS+M(I)
      TO(I)=0.0
10    CONTINUE
      TO(N)=0.00
      TY(N)=.5*TMASS
      TX(N)=TY(N)
      GAMMA=.2*TMASS
      SAVE=1000000.
      TXS=TX(N)
      TYS=TY(N)
20    TY(1)=TY(N)
      TY(1)=TMASS-TY(N)
      TEN(1)=SQRT(TX(1)**2+TY(1)**2)
      A=TY(1)/TX(1)
      THETA(1)=ATAN(A)
      EXO=L(1)*COS(THETA(1))
      EYO=L(1)*SIN(THETA(1))
      DO 30 I=2,N
      TY(I)=TY(I-1)-M(I-1)
      TX(I)=TX(I-1)
      TEN(I)=SQRT(TY(I)**2+TX(I)**2)
      A=TY(I)/TX(I)
      THETA(I)=ATAN(A)
      EXO=EXO+L(I)*COS(THETA(I))
      EYO=EYO+L(I)*SIN(THETA(I))
30    CONTINUE
```

```
      R=SQRT((X(
      IF(R.GT.SAVE) GO TO 4
      GAMMA=ATAN2(
      GO TO 6
    4 IF(R.EQ.0.0) GO TO 5
      SAVE=R
      EXSAVE=EX(
      EYSAVE=EY(
      TXS=TX(N)
      TYS=-TY(N)
      SAVE=SAVE
    5 TX(N)=TXS+GAMMA*(X(
      TY(N)=TYS+GAMMA*EY(S/SAVE
      GO TO 21
    6 CONTINUE
      A=3.14159265*2.
      DO 7 I=1,N
      IF(THETA(I).GT.3.14159265) THETA(I)=THETA(I)-A
    7 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CALL(N,THETA,TD)
      COMMON/BS/XO,YO,XD,YD,XDD,YDD,DENSITY,YC
      DIMENSION THETA(30),TD(30)
      CALL ANGLES(N,THETA)
      CALL COORD(N,XO,YO,XD,YD,TD)
      CALL MASS(N,YO)
      CALL DRAG1(N)
      CALL XYMASS1(N,XDD,YDD,YD)
      CALL GRAVITY(N,YO,DENSITY)
      CALL DRAG2(N,THETA)
      CALL SETUP(N,TD)
      RETURN
      END
```

```
      SUBROUTINE ANGLES(P,RTHETA)
      DIMENSION RTHETA(30)
      COMMON/ANG/S1(30),C1(30)/DIFFANG/S2(30,30),C2(30,30)
      DO 1 I=1,P
      S1(I)=SIN(RTHETA(I))
      C1(I)=COS(RTHETA(I))
      DO 1 J=1,P
      DIFF=RTHETA(I)-RTHETA(J)
      S2(I,J)=SIN(DIFF)
      C2(I,J)=COS(DIFF)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE COORD(N,XO,YO,XO,YO,TO)
      DIMENSION TO(30)
      REAL L
      COMMON/LENGTH/L(30)/ANG/S1(30),C1(30)/POS/X(30),Y(30)/VEL/VX(30),V
     CY(30)
      X(1)=L(1)*C1(1)+XO
      Y(1)=L(1)*S1(1)+YO
      VX(1)=XO-L(1)*TO(1)*S1(1)
      VY(1)=YO+L(1)*TO(1)*C1(1)
      DO 10 I=2,N
      X(I)=X(I-1)+L(I)*C1(I)
      Y(I)=Y(I-1)+L(I)*S1(I)
      VX(I)=VX(I-1)-L(I)*TO(I)*S1(I)
      VY(I)=VY(I-1)+L(I)*TO(I)*C1(I)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MASS(N,M)
      REAL M,L,MASS1,MASS2
      COMMON/TOTMAS/T1(30)/ADMAS/A1(30)/DIFFANG/S2(30,30),C2(30,30)/LENG
     1TH/L(30)/MASS/M(30)/NUMMAS/MASS1(30,30),MASS2(30,30)
      DO 60 I=1,N
      DO 60 J=1,N
      K=I
      SIGMA=0.
      IF(I.GT.J) GO TO 10
      SIGMA=1.
      K=J

10    CONTINUE
      MASS1(I,J)=T1(K)*C2(I,J)-.5*SIGMA*A1(J)*C2(I,J)
      MASS2(I,J)=-T1(K)*S2(I,J)
      TOTALM1=0.
      TOTALM2=0.
      DO 20 L1=K,N
      TOTALM1=TOTALM1+A1(L1)*C2(I,L1)*C2(J,L1)
20    CONTINUE
      IF(J.EQ.N) GO TO 40
      K1=J+1
      IF(I.GT.J) K1=I
      DO 30 L1=K1,N
      TOTALM2=TOTALM2+A1(L1)*C2(I,L1)*S2(J,L1)
30    CONTINUE
40    MASS1(I,J)=MASS1(I,J)+TOTALM1
      MASS2(I,J)=MASS2(I,J)+TOTALM2
      MASS1(I,J)=MASS1(I,J)*L(J)
60    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DRAG1(N)
      DIMENSION VN(30),VT(30)
      COMMON/VEL/VX(30),VY(30)/ANG/S1(30),C1(30)/DIFFANG/S2(30,30),C2(30
     C,30)/COEF/DT(30),DN(30)/CURRENT/VC(30)/DRAG2/DRAG(30)
      VN(1)=.5*((VC(1)-VX(1))*S1(1)+VY(1)*C1(1))
      VT(1)=0.00
      DO 10 I=2,N
      J=I-1
      VN(I)=.5*(VC(I)+VC(J)-VX(I)-VX(J))*S1(I)+.5*(VY(I)+VY(J))*C1(I)
      VT(I)=.5*(VC(I)+VC(J)-VX(I)-VX(J))*C1(I)-.5*(VY(I)+VY(J))*S1(I)
10    CONTINUE
      DO 30 I=1,N
      DRAG(I)=0.00
      DO 20 J=I,N
      DRAG(I)=DRAG(I)-DN(J)*VN(J)*ABS(VN(J))*C2(I,J)-DT(J)*VT(J)*ABS(VT(
     CJ))*C2(I,J)
20    CONTINUE
30    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE XYMASS1(N,XDD,YDD,YL)
      REAL M,L
      COMMON/TOTMAS/TM(30)/ADMAS/AM(30)/LENGTH/L(30)/MASS/M(30)/XY/XYMAS
     1S(30)/ANG/S1(30),C1(30)/DIFFANG/S2(30,30),C2(30,30)
      DO 21 I=1,N
      XYMASS(I)=0.0
      DO 10 J=1,N
      XYMASS(I)=XYMASS(I)+AM(J)*S1(J)*C2(I,J)*XDD-AM(J)*C1(J)*C2(I,J)*YD
     1D
   10 CONTINUE
      XYMASS(I)=XYMASS(I)+1.*(XDD*S1(I)-YDD*C1(I))*TM(I)
   20 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GRAVITY(N,Y,,DENSITY)
      REAL M
      COMMON/MASS/M(30)
      COMMON/ANG/S1(30),C1(30)/TOTMAS/TM(30)/POS/X(30),Y(30)/G/GRAV(30)
      WDEN=64.
      G=32.2*(DENSITY-WDEN)/DENSITY
      DO 10 I=1,N
      GRAV(I)=G*TM(I)*C1(I)
10    CONTINUE


      IF(M(N).LT.100.00.) GO TO 60
      DO 20 I=1,N
      GRAV(I)=GRAV(I)-M(I)*C1(I)*G
20    CONTINUE
      N1=N-1
      DO 50 I=1,N1
      IF(Y(I).LT.Y( ) GO TO 50
      DO 40 J=1,I
      GRAV(J)=GRAV(J)-M(I)*C1(J)*G
40    CONTINUE
50    CONTINUE
60    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DRAG2A(N,THETA)
      DIMENSION THETA(3)
      COMMON/ACOEF/D2,RL/VX(3),VY(3)/ANG/ADRAG(3)
      IF(VX(N).NE.0.0) GO TO 20
      PHI=1.5707963
      GO TO 30
20    R=VY(N)/ABS(VX(N))
      PHI=ATAN(R)
      IF(VX(N).LT.0.0)PHI=3.1415926-PHI
30    FDRAG=D2*(VX(N)**2+VY(N)**2)
      DO 40 I=1,3
      ADRAG(I)=FDRAG*SIN(THETA(I)-PHI)
40    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SETUP(N,TD)
      REAL MASS1,MASS2
      DIMENSION TD(30),TDSQ(30)
      COMMON/G/GRAV(30)/NUMMAS/MASS1(30,30),MASS2(30,30)/XY/XYMASS(30)/D
     1RAG2/DRAG(30)/EQUAL/EQUALS(30)
      COMMON/ANG/ADRAG(30)
      DO 10 I=1,N
      TDSQ(I)=TD(I)**2
10    CONTINUE
      DO 20 I=1,N
      EQUALS(I)=0.0
      DO 21 J=1,N
      EQUALS(I)=EQUALS(I)+MASS2(I,J)*TDSQ(J)
21    CONTINUE
      EQUALS(I)=EQUALS(I)+ADRAG(I)
      EQUALS(I)=EQUALS(I)+GRAV(I)+XYMASS(I)+DRAG(I)
20    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SOLVE(N,TOO,KORRECT)
      DIMENSION DELTA(3 ),TOO(3 )
      REAL MASS1,MASS2,L
      COMMON/NUMAS/MASS1(3 ,3 ),MASS2( ,3 )/EQUAL/EQUALS(3 )/RESID/R(3
     00)/W/WORK(3 ,31)/LENGTH/L(3 )
      DO 11 I=1,N
      WORK(I,N+1)=EQUALS(I)
      DO 11 J=1,N
      WORK(I,J)=MASS1(I,J)
10    CONTINUE
      CALL GAUSS(N,TOO)
      IF(KORRECT.EQ.0) GO TO 5.
      DO 51 K=1,KORRECT
      DO 21 I=1,N
      R(I)=-EQUALS(I)

      DO 21 J=1,N
      R(I)=R(I)+MASS1(I,J)*TOO(J)
20    CONTINUE
      DO 31 I=1,N
      WORK(I,N+1)=R(I)
      DO 30 J=1,N
      WORK(I,J)=MASS1(I,J)
30    CONTINUE
      CALL GAUSS(N,DELTA)
      DO 41 I=1,N
      TOO(I)=TOO(I)-DELTA(I)
40    CONTINUE
50    CONTINUE
      DO 61 I=1,N
61    TOO(I)=TOO(I)/L(I)
      RETURN
      END
```

```
      SUBROUTINE GAUSS(N,X)
      DIMENSION X(30)
      COMMON/W/WORK(30,31)
      N1=N+1
      DO 10 I=1,N
      WORKII=WORK(I,I)
      IF(WORKII.EQ.0.00) GO TO 8
      GO TO 5
    8 PRINT 9
    9 FORMAT(53H ABNORMAL HALT CAUSED BY DIVIDING BY ZERO IN GAUSS ROUTI
     ONE)
      CALL EXIT
    5 DO 10 J=1,N1
      WORK(I,J)=WORK(I,J)/WORKII
   10 CONTINUE
      DO 30 J=1,N
      WORKIJ=WORK(J,I)
      DO 20 K=1,N1
      IF(J.EQ.I) GO TO 20
      WORK(J,K)=WORK(J,K)-WORK(I,K)*WORKIJ
   20 CONTINUE
   30 CONTINUE
      DO 40 I=1,N
      X(I)=WORK(I,N+1)
   40 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE RK(N,DELTAT,THETA1,TD1,TDD,THETA,TD,KORRECT)
      DIMENSION THETA1(30),TD1(30),TDD1(30),THETA(30),TD(30),TDD(30),THE
     1TA2(30),TD2(30),TDD2(30),AK1(30),AK2(30),AK3(30),Q1(30),Q2(30),Q3(
     230)
      CALL CALL(N,THETA1,TD1)
      CALL SOLVE(N,TDD1,KORRECT)
      DO 10 I=1,N
      AK1(I)=DELTAT*TDD1(I)
      Q1(I)=DELTAT*TD1(I)
      TD2(I)=TD1(I)+AK1(I)/2.
      THETA2(I)=THETA1(I)+Q1(I)/2.
10    CONTINUE
      CALL CALL(N,THETA2,TD2)
      CALL SOLVE(N,TDD2,KORRECT)
      DO 20 I=1,N
      AK2(I)=DELTAT*TDD2(I)
      Q2(I)=DELTAT*TD2(I)
      TD(I)=TD1(I)-AK1(I)+2.*AK2(I)
      THETA(I)=THETA1(I)-Q1(I)+2.*Q2(I)



20    CONTINUE
      CALL CALL(N,THETA2,TD2)
      CALL SOLVE(N,TDD2,KORRECT)
      DO 30 I=1,N
      AK3(I)=DELTAT*TDD2(I)
      Q3(I)=DELTAT*TD2(I)
      TD(I)=TD1(I)+(AK1(I)+4.*AK2(I)+AK3(I))/6.
      THETA(I)=THETA1(I)+(Q1(I)+4.*Q2(I)+Q3(I))/6.
30    CONTINUE
      CALL CALL(N,THETA,TD)
      CALL SOLVE(N,TDD,KORRECT)
      RETURN
      END
```

```
      SUBROUTINE PREDICT(N,DELTAT,TD1,THETA,TD,TDD,THETA2,TD2)
      DIMENSION TD1(30),THETA(30),TD(30),TDD(30),THETA2(30),TD2(30)
      DO 10 I=1,N
      TD2(I)=TD1(I)+2.*DELTAT*TDD(I)
      THETA2(I)=THETA(I)+DELTAT*(TD(I)+TD2(I))/2.
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CORRECT(N,DELTAT,THETA1,TD1,TDD1,TDD2,THETA,TD)
      DIMENSION THETA1(30),TD1(30),TDD1(30),TDD2(30),THETA(30),TD(30)
      DO 10 I=1,N
      TD(I)=TD1(I)+DELTAT*(TDD1(I)+TDD2(I))/2.
      THETA(I)=THETA1(I)+DELTAT*(TD1(I)+TD(I))/2.
10    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TENSION(N,XDD,YDD,THETA,TD,TDD,DENSITY,XD,YD,YD)
      REAL L,M
      DIMENSION THETA(30),TD(30),TDD(30),AT(30),VN(30),VT(30)
      COMMON/L.NGTH/L(30)/MASS/M(30)/COEF/DT(30),DN(30)/VEL/VX(30),VY(30
     C)/ANG/S1(30),C1(30)/DIFFANG/S2(30,30),C2(30,30)/TEN/T(30)
      COMMON/CURRENT/VC(30)/POS/X(30),Y(30)
      COMMON/ACOEF/DA
      AT(1)=-XDD*C1(1)-YDD*S1(1)+L(1)*TD(1)**2
      DO 11 I=2,N
      I1=I-1
      AT(I)=-XDD*C1(I)-YDD*S1(I)+L(I)*TD(I)**2
      DO 11 J=1,I1
      AT(I)=AT(I)+L(J)*(TDD(J)*S2(J,I)+C2(J,I)*TD(J)**2)
10    CONTINUE
      VN(1)=.5*L(1)*TD(1)-XD*S1(1)+YD*C1(1)+VC(1)*S1(1)
      VT(1)=-XD*C1(1)-YD*S1(1)+VC(1)*C1(1)
      DO 20 I=2,N
      I1=I-1
      VN(I)=.5*L(I)*TD(I)-VX(I1)*S1(I)+VY(I1)*C1(I)+VC(I)*S1(I)
      VT(I)=-VX(I1)*C1(I)-VY(I1)*S1(I)+VC(I)*C1(I)
20    CONTINUE
      WDEN=64.
      G=32.2*(DENSITY-WDEN)/DENSITY
      IF(VX(N).NE.0.0) GO TO 3
      PHI=3.14159265/2.
      GO TO 2
3     A=VY(N)/ABS(VX(N))
      IF(VX(N).LT.0.0) GO TO 1
      PHI=ATAN(A)
      GO TO 2
1     A=ABS(A)
      PHI=3.141592e5-ATAN(A)
2     CONTINUE
      T(N)=M(N)*AT(N)+M(N)*G*S1(N)+.5*DT(N)*VT(N)*ABS(VT(N))+DA*(VX(N)**
```

```
     [illegible]+VY(N)**2)*COS(PHI-THETA(N))
      IF(M(N).G [illegible].) T(N)=T(N)-M(N)*G*S1(N)-M(N)*AT(N)
      DO 30 I=2,N
      K=N+1-I
      K1=K+1
      T(K)=M(K)*AT(K)+T(K1)*C1(K,K1)+M(K)*G*S1(K)+.5*(DT(K)*VT(K)*ABS(VT
     1(K))+DT(K1)*C1(K,K1)*VT(K1)*ABS(VT(K1))-DN(K1)*S2(K,K1)*VN(K1)*ABS
     2(VN(K1)))
      IF(M(K).[illegible]) T(K)=T(K)-M(K)*G*S1(K)
30    CONTINUE
      DO 40 I=1,N
      T(I)=T(I)/[illegible]
40    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BOTTOM(I,TC,TIME)
      REAL L,M
      DIMENSION TC(30)
      COMMON/LENGTH/L(30)/ANG/S1(30),C1(30)/DIFFANG/S2(30,30),C2(30,30)/
     1MASS/M(30)/TOTMAS/TM(30)
      N1=N-1
      N2=N-2
      A=B=0.00
      DO 10 I=1,N2
      C=L(I)*TC(I)
      A=A+C*S2(I,N)
      B=B+C*S2(I,N1)
   10 CONTINUE
      TC(N)=A/(L(N)*S2(N,N1))
      TC(N1)=B/(L(N1)*S2(N1,N))
      PRINT 20,TIME
   20 FORMAT(////,20H STOP CALLED AT TIME= ,F11.7)
      M(I)=M(I)+1.0
      DO 30 I=1,N
      TM(I)=TM(I)+1.0
   30 CONTINUE
      RETURN
      END
```

Appendix C

Program Sequence

NVR01 - The standard method of characteristic program reference [5] debugged for the Bonneville Power Administration Computer.

NVR02 - The first modification of NVR01 to allow the boundary conditions for the anchor-drop problem.

NVR03 - An improvement of NVR02 which used equations (3) of the text for calculating line coordinates.

NVR04 - An improved version of NVR02 using a Simpson's integration formula to integrate equations (2) of the text.

NVR05 - The first lumped mass program using a self starting predictor-corrector for the "Standard catenary configuration" of a Fig. G.

NVR06 - The same program as NVR05 except the initial condition is the "Goose neck configuration" of Figure E.

NVR07 - The inextensible line lumped mass model of reference [9].

NVR08 - The final version of the lumped mass model using the second order predictor-corrector of Section II-3.0.

Mooring Line

Nylon - 2.5" Dia.

Density = 71 lb./ft^3

$C_N = 1.4$

$C_T = (.008)(1.4)$

$CI_t = 0.$

$CI_N = .5$

Length = 5660 ft. unstressed

Stress = $C_1(\text{strain})^{C_2}$

$C_1 = 1.67 \times 10^8$ PSF

$C_2 = 4.37$

Dacron - 1.0" Dia.

Density - 1.0" Dia.

$C_N = 1.4$

$C_T = (.008)(1.4)$

$CI_t = 0.$

$CI_N = .5$

Stress = $C_1(\text{strain})^{C_2}$

$C_1 = 1.2 \times 10^9$

$C_2 = 3.7$

length 5660 ft. unstressed

Steel - 2.5" Dia.

Density = 350 lb/ft^3

C_N = 1.3

C_T = (.008)(1.3)

CI_t = 0.

CI_N = .5

Inextensible

Anchor

Weight = 12000 lb.

Vol. = 24.5 ft^3

Frontal Area = 12.6 ft^2

C_D = .4

C_I = 1.0

Length = 5660 ft.

Water

Density = 64 lb/ft^3

Depth = 5000 ft.

Computer run times for computations made on the Bonneville Power Administration CDC-6400.

Runs made for nylon lines

1) 15° catenary

a) Early portion of drop $t = \frac{.14 \text{ computer sec}}{\text{Real Sec - Mass}}$

b) Late portion of drop $t = \frac{.1 \text{ Computer Sec}}{\text{Real Sec - Mass}}$

2) 60° catenary

a) Early portion of drop $t = \frac{.28 \text{ Computer Sec}}{\text{Real Sec - Mass}}$

b) Late portion of drop $t = \frac{.21 \text{ Computer Sec}}{\text{Real Sec - Mass}}$

Runs made for dacron lines

1) 15° catenary

Early portion of drop $t = \frac{.84 \text{ Computer Sec}}{\text{Real Sec - Mass}}$

2) 60° Catenary

Early portion of drop $t = \frac{1.4 \text{ Computer Sec}}{\text{Real Sec - Mass}}$

List of Symbols

A	Frontal Area of Anchor
ACN	Normal Acceleration
ACT	Tangential Acceleration
C_n	Constants
CD	Drag Coefficients
CDN	Normal Drag Coefficient
CDT	Tangential Drag Coefficient of the Line
CI	Added Mass Coefficient for Anchor
CIN	Added Mass Coefficient Normal for Line
CIT	Added Mass Coefficient Tangential for Line
D	Line Diameter
Da	Anchor Diameter
DFN	Drag Force in the Normal Direction
DFT	Drag Force in the Tangential Direction
M_i	The i^{th} mass representing the Line
MA	Anchor Mass
SF	Spring Force
SFX	Spring Force in the X direction
SFZ	Spring Force in the Z direction
T	Line Tension at Anchor
VCX	Local Water Velocity in X Direction
VCY	Local Water Velocity in Y Direction

VOL	Anchor Displaced Volume
VCN	Line Velocity Normal
VCT	Line Velocity Tangential
W	Anchor Weight
WT_o	Weight of the i^{th} Mass
ℓ_j	Line Segment Length
k_1 to k_4	Runge Kutta Parameters
ℓ_1 to ℓ_3	Runge Kutta Parameters
m_1 to m_4	Runge Kutta Parameters
n_1 to n_4	Runge Kutta Parameters
x	Horizontal Coordinate
z	Vertical Coordinate
S	Arc Distance Along the Line
ΔS	Equal Distance Taken between Calculation Point
Δt	Time Increment
β	Angle from the Horizontal to the Anchor Line Link
ϵ_i	Line Strain at the i^{th} link
θ	Angle Between the Horizontal and the Line Tangent
θ_{AV}	The Average Between Two Adjacent Links
ρ	Mass Density of the Line
ρa	Mass Density of the Anchor
σ	Stress in the Line
ε	Error in Velocity Computation