RIGID LINK AND CATENARY SKYLINE ANALYSIS
PROGRAMS FOR MICROCOMPUTERS
by
David E. Kendrick

A PAPER
Submitted to
Oregon State University

in partial fulfillment of the
requirements for the degree of
Master of Forestry
Completed --- --, 1992
Commencement June 1992
AN ABSTRACT OF THE PAPER OF


Title: RIGID LINK AND CATENARY SKYLINE ANALYSIS PROGRAMS FOR MICROCOMPUTERS

Abstract Approved: Dr. John Sessions

Skyline logging is a commonly used method of harvesting logs, particularly on steep slopes. Logging systems planners frequently solve two kinds of skyline analysis problems. The first is the maximum payload problem and the second is the standing skyline load path problem. Computer assisted skyline analysis techniques have been available for over two decades. As microcomputers have become less expensive and more powerful, computer assisted skyline analysis has come within the reach of most logging systems planners.

The objective of this paper is to present two skyline analysis programs, one using a rigid link model and the other using a catenary model to solve maximum payload problems for live, running, and standing skyline programs and load path problems for standing skylines. The standing skyline analysis can include either single span or multispan systems.
A live skyline lift routine is included. Both programs can model a carriage that is clamped to the skyline or a carriage that is not clamped to the skyline. Both programs are written in Microsoft QuickBASIC Version 4.5.
# TABLE OF CONTENTS

1. **INTRODUCTION** .................. 1
   1.1 Skyline Analysis .................. 1
      1.1.1 Types of Skyline Systems ....... 1
      1.1.2 Skyline Analysis Problems ...... 2
      1.1.3 Skyline Analysis Using Microcomputers ... 2
   1.2 Objectives ........................ 3
      1.2.1 Skyline Systems Analyzed ...... 4
      1.2.2 Additional Features and Options ... 5
   1.3 Scope ................................ 5
   1.4 Literature Review .................. 5
      1.4.1 Computer Models ................ 7
      1.4.2 Mathematical Formulations ...... 7

2. **SOLUTION MODELS** ................ 7
   2.1 Format ................................ 8
   2.2 Description of Rigid Link and Catenary Models ... 8
      2.2.1 Catenary Model .................. 9
      2.2.2 Rigid Link Model ................ 9

3. **SKYLINE SYSTEMS MODELED** .......... 10
   3.1 Assumptions ........................ 10
   3.2 User Defined Variables ............. 11
   3.3 Four Segment Analysis Model ....... 11
   3.4 Standing Skyline .................... 12
      3.4.1 Description of Standing Skyline ... 12
      3.4.2 Standing Skyline Line Length ...... 13
      3.4.3 Standing Skyline Load Path ......... 14
   3.5 Live Skyline ........................ 15
      3.5.1 Description of Live Skyline ....... 15
      3.5.2 Live Skyline with Constant Clearance .... 15
      3.5.3 Live Skyline with Lifts .......... 15
      3.5.4 Live Skyline Lift Routine Example ... 16
      3.5.5 Live Skyline with Clamping Carriage .... 18
      3.5.6 Clamping Carriage with Slack Mainline ... 19
      3.5.7 Clamping Carriage with Active Mainline .. 20
   3.6 Multispan Skyline ................... 20
   3.7 Running Skyline ........................ 21
      3.7.1 Description of Running Skyline ...... 21
      3.7.2 Running Skyline Limited by ....... 22
      3.7.3 Calculation of Torque-Limited Tension .......... 22

4. **OPTIONAL FEATURES** .............. 23
   4.1 Skyline Adjustment .................. 23
   4.2 Unloaded Skyline Tension and Line Length ... 24

5. **REFERENCES** ....................... 25

6. **APPENDIX A--LIST OF EQUATIONS** .... 28
<table>
<thead>
<tr>
<th>Appendix</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>APPENDIX B--PARTIAL SUSPENSION</td>
<td>39</td>
</tr>
<tr>
<td>8</td>
<td>APPENDIX C--STANDING SKYLINE LOAD PATH</td>
<td>44</td>
</tr>
<tr>
<td>9</td>
<td>APPENDIX D--LIST OF VARIABLES</td>
<td>48</td>
</tr>
<tr>
<td>10</td>
<td>APPENDIX E--PROGRAM LISTING</td>
<td>59</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

1.1 Skyline Analysis

Skyline yarding is a commonly used logging method, particularly on steeper slopes. The flexibility of skyline yarding and the ability to suspend one or both ends of the payload make it well suited over a broad range of conditions, both economically and environmentally. Skyline systems can be divided into three categories: standing skyline, live skyline, and running skyline. All three types are characterized by having a skyline supported at both ends with the payload suspended beneath a carriage that moves along the skyline.

1.1.1 Types of Skyline Systems

The three types of skyline yarding systems and many of the variations of those systems are described in detail in the Logging Systems Guide (1979). The distinguishing features of the three systems are described briefly below.

A standing skyline is fixed at both ends so that the unstretched skyline length does not change during the yarding cycle. Standing skylines can be further divided in single span or multispan systems. Multispan systems have one or more intermediate supports between the headspar and the tailspar. The distinguishing characteristic between a live skyline and
a standing skyline is the ability to raise, and sometimes lower, the live skyline during a given yarding cycle. The ability to raise or lower the skyline, and thereby change the unstretched skyline length, allows the operator to increase the payload capacity along the corridor. The "skyline" of a running skyline system is the haulback line which constantly moves as the carriage is brought to the landing. The ability to constantly change the length of the haulback line as the carriage is brought in allows the greatest opportunity to optimize the payload capacity over an entire skyline corridor.

### 1.1.2 Skyline Analysis Problems
In designing skyline logging systems that are economically efficient and that achieve environmental protection goals, the logging systems planner often needs to solve problems of two types. The first, and probably the most common, is the "maximum payload" problem. As its name suggests, a maximum payload analysis calculates the maximum payload that can be brought to the landing given the ground profile, log to ground geometry, user defined payload clearances, and the physical limitations of the yarder and its operating lines. A "load path" analysis differs from the maximum payload analysis in that the payload is known and the analyst calculates the payload clearances and the resulting skyline tensions as the payload is brought to the landing.
1.1.3 Skyline Analysis Using Microcomputers

Both types of skyline analysis problems require calculating payloads or line tensions at each terrain point, often as many as 40 or 50, along a skyline profile. The repetitive nature of the calculations makes computerized solutions a cost efficient method of analyzing skyline systems. The ability to rapidly evaluate a variety of yarding system configurations encourages logging systems planners to test a range of alternatives to improve economic efficiency and environmental protection. As microcomputers have become increasingly available and less expensive, computer analysis of skyline problems has come within the reach of most logging systems planners in both the public and private sectors.

1.2 Objectives

A microcomputer based skyline analysis program in common use is LOGGER PC (Balcom, 1988). A joint effort by Oregon State University and the USDA Forest Service is currently underway to produce a "state of the art" successor to LOGGER PC. The primary objective of this paper is to provide a computer program containing the mathematical formulations that can be used to solve skyline analysis problems in that successor to LOGGER PC.

LOGGER PC was adapted from an earlier skyline analysis program written for the HP 9845 and HP 9000 series of
computers (Nickerson 1980). The code for LOGGER PC evolved over a number of years with contributions by a number of authors. Because of the difficulty in understanding the computer code that has come from a number of sources and the need to add new features, it was decided to simply start over rather than try to adapt the existing code to the new program.

At the time this project was started, the decision of what programming language to use had not been made. To facilitate conversion to higher programming languages, this program was written in Microsoft QuickBASIC for DOS systems. This program was also written with the knowledge that it would be taken by someone else and incorporated into a larger software package. While it is always difficult to understand a computer code written by a different programmer, efforts were made to make this program relatively easy to read and understand. In addition to user notes throughout the program, variable and subroutine names are intended to be descriptive. For example, the variable used in the program to represent the calculated payload is "payloadt" and a subroutine that adjusts the skyline tension is "SkylineAdjustment".

1.2.1 Skyline Systems Analyzed

This program will solve two-dimensional live, standing, and running skyline analysis problems. Live and standing
Skyline systems can be solved either with or without a haulback line. Both single span and multispans standing skylines can be analyzed. The maximum payload problem can be solved for any of the above systems. The load path analysis is available only for single span standing skylines. All systems can be analyzed in either uphill or downhill yarding configurations and all can be analyzed for either full suspension or front end suspension of the payload. An option is included for the user to specify a maximum torque for the haulback drum that can be supplied by the yarder for analyzing running skyline payloads. Both clamping carriages and nonclamping carriages can be modeled.

1.2.2 Additional Features and Options

1. A routine was included to more accurately model live skyline systems than is available in either LOGGER PC or other skyline analysis programs currently available.

2. Both catenary and rigid link were written. Only the program listing for the catenary model is included in Appendix E. The structure of the two models is very similar and, in most cases, rigid link approximations are used as first estimates in the catenary model.

3. The user has the option of calculating the unloaded length and tension of standing skylines.

4. Partial suspension analysis can include the diameter of the payload.
1.3 Scope

1. This program presents a mathematical model only. No field measurements were taken.

2. All systems are assumed to be in two-dimensional static equilibrium.

3. Only skyline systems are analyzed. Other cable systems such as high lead and jazunier yarding are not included.

4. Only catenary and rigid link models are included.

5. Systems with more than four line segments cannot be analyzed with this program. An exception to this is a system with a slackpulling line. The weight of the slackpulling line is added to that of the mainline and the weight of the slackpulling line is assumed to be zero.

6. With the exception of friction between a dragging log and the ground, all systems are assumed to be frictionless.

7. This program includes only the mathematical formulations needed to solve skyline analysis problems. User inputs, output screens, and other user interface features are beyond the scope of this project. Output screens include in the program listing are included only to facilitate programming and debugging.

8. Payloads based on the ability of the yarder to provide tension in the haulback line are based on a user-defined maximum haulback torque. Modeling of the
yarder's drive train is beyond the scope of this project. This program focuses primarily on the payload limited by the allowable tensions in the working lines.

9. To the extent possible, existing mathematical formulations developed by other authors were used. In some cases, those formulations were adapted to fit situations not found elsewhere in the literature.

1.4 Literature Review

1.4.1 Computer Models

Computer and programmable calculator methods of analyzing skyline systems have been available for over two decades (Perkins, et. al., 1969). The earliest of these solved payload analysis problems only for fully suspended payloads or used a percentage of the fully suspended payload to approximate the partially suspended payload. Later programs calculated partially suspended payloads based on the log-to-ground geometry (Tobey, 1980, Falk, 1981, Chung, 1987, Twito, et. al., 1988). Several programs include analysis routines for multispans standing skyline systems (Chung, 1987, Balcom, 1988, Durston, 1989). LOGGER PC is written using rigid link assumptions for the cable segments.

1.4.2 Mathematical Formulations

The mathematical formulations for modelling skyline systems were developed early in this century (Anderson,

2. SOLUTION MODELS

2.1 Format

This paper will focus on the assumptions that went into making the models rather than on the mathematical formulations. As stated earlier, the formulations used here were taken from the work of other authors. Complete descriptions of their formulations can be found in their referenced publications. The formulations themselves will be discussed only where needed to clarify combinations or adaptations of the original formulations. Mathematical formulations used in this program are discussed further in Appendix A. Partial suspension payloads are discussed in Appendix B.

The format of this paper will be to first discuss the rigid link and catenary models. Next, it will describe some of the basic assumptions and features available for all skyline systems analyzed. The different skyline systems (standing, live, multispans, and running skylines) and how they are modeled will then be discussed. Finally, it will describe optional routines that are available.
2.2 Description of Rigid Link and Catenary Models

Skyline systems can be modelled in several different ways. The models differ primarily in their assumptions regarding the forces acting on an individual cable segment, including the cable's own weight. Among the models used are: weightless line, rigid link, parabolic, and catenary, as well as hybrid models combining two or more models. Both the catenary and rigid link models are used in this paper. Following is a brief discussion of the assumptions, advantages and disadvantages of the catenary and rigid link models. Miller (1990) describes and compares the two models in greater detail.

2.2.1 Catenary Model

The catenary model assumes that an individual segment acts as a cable loaded by its own weight. That assumption means that the length of the cable segment is the length of a catenary arc and, unless both ends of the segment are at the same elevation, the resultant force of the weight of the segment will act at a point other than the segment's midpoint. While the catenary model more accurately describes the true nature of the skyline cables, solution of the catenary equations requires iterative methods and takes longer to compute than does the rigid link model.
2.2.2 Rigid Link Model

The rigid link model assumes that the weight of the individual segments can be approximated by the straight line distance between the ends of the segments multiplied by the unit weight of the cable. The resultant force of the weight of the segment acts at the segment's midpoint. Those simplifying assumptions allow a direct solution of the payload that is faster computationally than the catenary model. The rigid link model is, in theory, not as accurate as the catenary model. However, for cables loaded to approximately one-third breaking strength, the difference in payload and tensions compared to the catenary solutions are negligible.

3. SKYLINE SYSTEMS MODELED

The following sections will describe the features and options included for standing, live, multispan, and running skyline analysis. Aside from the differences discussed in the previous section, the assumptions made are identical for both the rigid link and catenary models.

3.1 Assumptions

All routines in this program are designed to model either uphill or downhill yarding and either full or partial suspension for the payload. The type of suspension must be specified by the user and can vary from one terrain point to
another. For example, full suspension could be specified over a stream with partial suspension allowed for the remainder of the profile.

The program needs to receive ground profile data as $x$ and $y$ coordinates with terrain points numbered consecutively. Whether the terrain points are numbered from the headspar to the tailspar or from the tailspar to the headspar is not important. Profile coordinates are accepted either way with no need to reverse the profile.

The models do not consider friction except for the friction between the log and the ground in the partial suspension analysis.

3.2 User Defined Variables

In general, the user must provide all data describing the yarder, carriage, ground profile, payload dimensions, and the locations of the headspar, tailspar, and intermediate supports. A complete list of the variables the user must provide is included in Appendix E. In cases where the typical user is unlikely to know or have ready access to the value of a variable or where the value probably will not change much from one problem to the next, default values could be included. Examples where that might be the case include the log-to-ground coefficient of friction or the dimensions of the haulback drum for a running skyline.
FIGURE 1. Model of Four Segment Skyline Harvesting System
Set $T_3 = 1'3$ allowable?

Adjust $T_1$; Solve for $H_1, V_1, T_2, H_2, V_2, T_3, T_4$.

Yes

Set $T_3 = 0$; Solve for $H_3, V_3, T_4$.

No

$sT_4 > 0$?

Adjust $T_1$; Calculate $V_1, T_2, H_2, V_2, H_3, V_3, T_3, T_4$.

$sT_4 > 0$?

No

$sT_4 > 0$?

Yes

$sT_3 > 0$?

Yes

$sT_3 > 0$?

No

$sT_3 = 0$?

No

$sT_3 = 0$?

Yes

$sT_3 > 0$?

Yes

$sT_3 = 0$?

No

$sT_3 = 0$?

Yes

$sT_3 = 0$?

No

$sT_3 = 0$?

Yes

$sT_3 = 0$?

No

$sT_3 = 0$?

Yes

$sT_3 = 0$?

No

$sT_3 = 0$?

Yes

$sT_3 = 0$?

No

$sT_3 = 0$?

Yes

$sT_3 = 0$?

No

$sT_3 = 0$?

FIGURE 2. Skyline Payload Analysis Decision Tree
3.3 **Four Segment Analysis Model**

These programs solve skyline analysis problems the multiple segment analysis technique described by Sessions (1990). In this case, all skyline systems are analyzed as four segment systems. For three segment live and standing skyline, the fourth segment is included, but its tension and unit weight are set equal to zero. Figure 1 shows a representation of a four segment system and the nomenclature used in this report. Segments 1 and 2 are the skyline for a live or standing skyline and the haulback for a running skyline. Segment 3 is the mainline, and Segment 4 is the haulback line.

In the case of the running skyline, this program assumes that the payload is limited by the minimum of the tensioning capability of the yarder or by the allowable tensions in the lines themselves. For all other systems, the model assumes that the payload is limited by the allowable line tensions.

In solving skyline analysis problems, this program always begins by assuming the limiting line is the haulback line for a running skyline or the skyline for any other system (skyline limited case). If the tension of any other line exceeds its allowable tension when the skyline limited case is calculated, either the mainline or the haulback line is then considered limiting and the problem is recalculated. Figure 2 shows a decision tree illustrating how the program
branches in calculating payload and determining which line is limiting.

3.4 Standing Skyline

3.4.1 Description of Standing Skyline

As used in this program, a standing skyline has the skyline anchored at both ends. Once the skyline is set up, it is neither raised nor lowered on any given corridor. One application of the standing skyline is to anchor the skyline at the tail spar and spool it off a drum on a line horse on the other end. In that case, the skyline can be raised or lowered as the corridor is logged, but not on any given yarding cycle. This application is more accurately modeled in the live skyline lift routine discussed below in Section 3.5.3.

The standing skyline analysis can be run either with or without a haulback line. If a haulback line is included, the user must specify a haulback tension that is applied by the yarder operator. The applied haulback tension is assumed in the program to be the maximum haulback tension anywhere in the system—either at the head spar or the tail spar for a single span skyline. This assumption is not completely consistent since the tension is actually set by the yarder operator as a torque at the haulback drum, but it prevents the value of the applied tension from causing the haulback tension to exceed its allowable tension somewhere.
else in the system. This is especially likely in downhill yarding where the tension at the tailspar is higher than the tension at the headspar and the haulback drum.

3.4.2 Standing Skyline Line Length

For any given corridor, the unstretched line length remains unchanged at each terrain point. The program calculates the unstretched line length based on the specified minimum payload clearances. First, the stretched line length that will give the minimum clearance is calculated for each terrain point. The unstretched line length for each point is then calculated using the elastic strain formula

\[ S_{pr} = S_0 \left(1 + \frac{T}{AE}\right) \]  

Where \( A \) is the metallic cross-sectional area of the skyline wire rope, \( E \) is the modulus of elasticity of the wire rope, \( T \) is the average tension in the skyline, and \( S \) is the line length. The program assumes that the ratio of \( AE \) divided by the unit weight of the skyline is 3,500,000.

The shortest unstretched line length for any terrain point is the unstretched line length for the entire profile. The stretched line length is calculated using the elastic strain formula and will vary from point to point depending
upon the average skyline tension. Once the stretched line length is known, the carriage clearance, payload clearance, and skyline geometry can be calculated.

3.4.3 Standing Skyline Load Path

In addition to knowing the maximum payload that a system can yard to the landing, it is sometimes useful to know what line tensions are developed when a payload of known weight is yarded up a corridor. Examples of when this type of analysis might be used include when mainline tensions are required for estimating line speeds or when actual carriage or payload clearances are required. That kind of analysis is known as a load path analysis. The load path problem is one in which the user specifies the payload and the unstretched line length and solves for the line tensions and clearances. This program will solve the load path problem for a standing skyline with or without a haulback line.

Since neither the skyline tension nor the stretched line length is known for a load path analysis, a direct solution is not possible. Kendrick and Sessions (1991) describe the iterative method used in this program for solving for the clearances and tensions. That method is described in greater detail in Appendix C.
3.5 Live Skyline

3.5.1 Description of Live Skyline

A live skyline is similar to a standing skyline except that the length of the unstretched skyline can be changed during carriage inhaul. A live skyline can be a three segment system with a skyline and a mainline (shotgun or flyer) or a haulback line can be added (slackline).

3.5.2 Live Skyline with Constant Clearance

Two separate options, based on different assumptions of skyline length, are available for the live skyline. The first model assumes a constant skyline or front end log clearance at each terrain point. In that model, the user specifies a clearance and the model calculates the resulting line length at each terrain point.

The assumption of a constantly changing unstretched skyline length does not accurately depict how a live skyline operates. Payload estimates obtained with this option should be considered the most optimistic since it assumes the maximum deflection possible at each terrain point.

3.5.3 Live Skyline with Lifts

The live skyline lift routine raises the skyline at the log pick up point and at user specified terrain points (lift points) between the log pick up point and the landing. The
The lift routine determines the unstretched line length and, therefore, the height of the skyline above the ground, for each lift point. The lift routine calculates the unstretched line length required to achieve the minimum required payload clearance at each terrain point and then selects the shortest unstretched line length from the log pick up point up to, but not including, the next lift point. If there are no other lift points between the pick up point and the landing, the lift routine selects the shortest unstretched line length from the pick up point to the landing.

The skyline can be raised, but cannot be lowered. If the unstretched line length at some terrain point (Pt. A) between the pick up point and the outer yarding limit is shorter than at any point from the pick up point to the landing, the line length for Pt. A will be used for all terrain points from the pick up point to the landing since a longer line length would require lowering the skyline. One result of this algorithm is that the payload clearance will always be greater than the minimum allowable clearance if the shortest unstretched line length for the entire profile is between the pick up point and the outer yarding limit.

3.5.4 Live Skyline Lift Routine Example

To illustrate how the lift routine selects the unstretched line length, consider the following table. The
required line length is the unstretched line length that would give the user specified minimum payload clearance. The selected line length is the unstretched length the routine would select when it analyzes the payload for a given terrain point. Assume the head spar is at terrain point 1 and the tail spar is at some point beyond terrain point 10. Terrain point 10 is the log pick up point.

<table>
<thead>
<tr>
<th>Terrain Point</th>
<th>Lift Point</th>
<th>Line Length Required</th>
<th>Line Length Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Yes</td>
<td>1000</td>
<td>998</td>
</tr>
<tr>
<td>9</td>
<td>No</td>
<td>998</td>
<td>998</td>
</tr>
<tr>
<td>8</td>
<td>No</td>
<td>1001</td>
<td>998</td>
</tr>
<tr>
<td>7</td>
<td>No</td>
<td>1003</td>
<td>998</td>
</tr>
<tr>
<td>6</td>
<td>Yes</td>
<td>999</td>
<td>998</td>
</tr>
<tr>
<td>5</td>
<td>No</td>
<td>1000</td>
<td>998</td>
</tr>
<tr>
<td>4</td>
<td>No</td>
<td>1002</td>
<td>998</td>
</tr>
<tr>
<td>3</td>
<td>Yes</td>
<td>996</td>
<td>995</td>
</tr>
<tr>
<td>2</td>
<td>No</td>
<td>995</td>
<td>995</td>
</tr>
</tbody>
</table>

The skyline is first raised at terrain point 10 since that is the pick up point. The critical unstretched line length is 998 feet at terrain point 9. The lift routine selects 998 feet as the unstretched line length for each terrain point from the pick up point until it reaches the next lift point at terrain point 6. At terrain point 6, the lift routine looks at all terrain points from 6 to the next lift point at terrain point 3. The shortest line length in that interval is 999 feet at terrain point 6. However, selecting a line length of 999 feet would require that the line be lowered. Therefore the line length remains at 998 feet. The final decision point comes at terrain point 3.
The shortest unstretched line length from terrain point 3 to the landing is 995 feet at terrain point 2. Since selecting 995 feet as the line length would require lifting rather than lowering the skyline, the lift routine selects 995 feet as the unstretched line length for terrain points 3 and 2.

This live skyline with lifts may provide more reasonable estimates of live skyline payloads because of the way it models live skyline operations. It gives estimates of the payload that can be supported at a given point as well as the payload that can be yarded back to the landing since it calculates the payload at each terrain point for each line length it uses.

The number of lifts that the user can specify is limited by the number of terrain points on the profile. However, the program will only "lift" the skyline if it finds that the line can be shortened without later having to lengthen it.

3.5.6 Live Skyline with Clamping Carriage

Up to this point, all analyses discussed have assumed a nonclamping carriage. At times, the user will want to know the payload using a clamping carriage. A clamping carriage is one that is held stationary on the skyline by a clamp or stop. This option is available in the program only for a live or standing skyline with no haulback line. It is assumed that the haulback line can be used to hold the
carriage stationary for systems equipped with a haulback line. There are many types of clamping carriages available. This program will model two commonly used types.

3.5.7 Clamping Carriage with Slack Mainline

The first type of clamping carriage is one where the mainline is slack and does not assist in supporting the payload as it is brought in to the carriage. This is generally the case with radio controlled carriages. The second type of clamping carriage assumes that the mainline passes through the carriage and functions as the tagline supporting the payload. The clamping carriage payload cannot exceed the nonclamping carriage payload since all carriages must be unclamped in order to bring the payload to the landing.

When the mainline is slack, the payload is suspended by the two skyline segments (and the ground if the payload is not fully suspended). The allowable tension in either segment of the skyline can limit the payload. The program first assumes that the segment between the head spar and the carriage is the limiting line and then calculates the tensions in the other segment. If the second segment's tension exceeds its allowable, the program then recalculates all tensions and the payload with the second segment at its allowable. Both the weight and the tension of the mainline are assumed equal to zero. The height of both the payload
and the carriage are assumed to be the same as for the nonclamping carriage.

3.5.8 Clamping Carriage with Active Mainline

To facilitate lateral yarding with a two drum yarder, some carriages are designed to be used with an "active" mainline. That is, the mainline can be pulled through the carriage once the carriage is clamped to the skyline. The mainline then acts in a role analogous to the dropline in a slackpulling carriage. An active mainline supports the entire weight of the payload if the payload is fully suspended. Therefore the payload cannot exceed the allowable mainline tension.

When the mainline is active, the factor limiting payload is normally either the mainline tension at the carriage, since it supports the payload, or by the payload once the carriage is unclamped. The program analyzes the clamping carriage, active mainline case by first running an analysis with the assumption of a nonclamping carriage. It then checks the tagline tension and compares it to the allowable mainline tension adjusted to the elevation of the carriage. If the tagline tension exceeds the allowable mainline tension, the skyline tension is adjusted until the tagline tension equals the allowable mainline tension. If the tagline tension is less than or equal to the allowable
mainline tension, the clamping carriage payload is assumed to be equal to the nonclamping carriage payload.

3.6 Multispan Skyline

A multispan skyline is analyzed as a standing skyline with one or more intermediate supports between the head spar and the tail spar. The carriage must pass over a jack at the intermediate support. The user must specify the locations of the intermediate supports and the height of each support jack. The number of intermediate supports is limited only by the number of terrain points on the profile.

The multispan analysis algorithm is essentially the same as that for a standing skyline. It can be run either with or without a haulback line. The unstretched line length calculated in the program is the length for the entire system, both loaded and unloaded spans. The program will locates the point of maximum skyline tension at either the head spar, the tail spar, or an intermediate support jack.

Friction between the carriage and the intermediate support jacks is ignored. This program does not provide a test of whether the carriage will pass over the intermediate support jacks.
3.7 Running Skyline

3.7.1 Description of Running Skyline

A running skyline has two working lines, the mainline and the haulback line. Unlike the live and standing skyline systems, both lines are constantly moving during carriage inhaul. A running skyline is still analyzed as a four segment system in this program. However, the calculations are simpler in that the tensions in the fourth segment are assumed to be equal to those of the second segment.

3.7.2 Running Skyline Limited by Yarder Tension

A major difference between the running skyline analysis and the analysis for other systems in this program is that it considers the ability of the yarder to provide tension to the haulback. In some cases, particularly when yarding downhill, it is the tensioning ability of the yarder that limits payload rather than the allowable tension of the wire rope (Hartsough, et. al., 1985). Hartsough, et. al. provide a running skyline analysis that models the drive trains of many of the running skyline yarders currently in use. An elaborate model such as that one is beyond the scope of this project, however, an estimate of the maximum tension dependent payload can be obtained based on a user specified maximum haulback drum torque.
3.7.3 Calculation of Torque-Limited Haulback Tension

In this model, only the maximum torque that can be applied at the center of the haulback drum, the haulback drum dimensions and the haulback line diameter are required. The maximum tension that can be applied to the haulback line at the drum can then be calculated. The maximum haulback tension derived from the haulback drum torque is then compared to the allowable haulback line tension and the lesser of the two is used as the maximum haulback tension in calculating the payload. The maximum tension in the haulback line occurs at the point of highest elevation of the haulback line, which will be at either the head spar or the tail spar. The tension at the haulback drum is converted to the tension at the highest elevation using the catenary relationship that the upper end tension is equal to the lower end tension plus the change in elevation multiplied by the unit weight of the haulback line.

\[ T_U = T_L + w_y \]  

(2)

In making this adjustment, it is assumed that the center of the haulback drum is 10 feet above the ground.

Once the maximum haulback tension has been calculated, the running skyline is completed in essentially the same way as the live skyline analysis with constant clearance.
4. Optional Features

4.1 Skyline Adjustment

An option is included that will adjust the skyline height between the carriage and the head spar so that the skyline and mainline will not be below the ground profile. The program first calculates the skyline geometry using the user specified minimum payload clearance. Using a rigid link assumption, it then checks whether a straight line from the top of the head spar to the carriage will pass below any terrain point between the head spar and the carriage. If so, the skyline, or haulback line for a running skyline, is adjusted so that it barely touches the ground at the critical terrain point.

4.2 Unloaded Skyline Tension and Line Length

For both the standing skyline and multispan systems, the unloaded skyline tension and line length can be calculated. The "unloaded" skyline is without a carriage or payload attached. The skyline is assumed to be loaded by its own weight. The tension and the difference between the stretched and unstretched line length are the result of the skyline supporting its own weight.
REFERENCES


Hartsough, Bruce R., John A. Miles and James E. Burk. Improved Skyline Analysis. Final Report to the USDA-Forest Service Pacific Northwest Forest and Range Experiment Station on Cooperative Agreement PNW-84-401.


Miller, M. 1990. Single Segment Analysis. FE 564 Class
Notes, Unpublished course material, Oregon State University, Dept. of Forest Engineering.


APPENDIX A
LIST OF EQUATIONS
APPENDIX A
LIST OF EQUATIONS

Source of Equations

The equations used in this program were either taken directly or adapted from the work of other authors. The sources of the catenary relationships were Carson (1977) and Miller (1990). Sessions (1989), Miller (1990), Tobey (1981), and Carson (1975) were the primary sources of the equations used in the rigid link model. Equations used to calculate partial suspension and carriage height were taken from Sessions (1989) and Falk (1982).

Terminology and Sign Conventions

Except as otherwise noted, the terminology and sign conventions used below are those used in Figure 1 of the text of this document.

Carriage Height (HC)

When the payload is fully suspended,

\[ HC = y_c + loght + L + LC + CD \]  

(3)

Where:

- \( y_c \) = y coordinate of ground beneath the carriage
- \( CD \) = depth of the carriage from point of tagline attachment to skyline
- \( loght \) = height of the bottom of the log above the ground

When the payload is partially (front end) suspended,
\[ \beta = \sin^{-1}\left(\frac{\text{Length} \cos \theta}{L}\right) \]  

\[ \alpha = \arctan\left[\frac{L}{\text{Length} \tan(\theta + \beta)} \right] + \left[ \frac{L - D \tan(\theta + \beta)}{L} \right] \frac{\cos \theta - \mu \sin \theta}{\sin \theta + \mu \cos \theta} \]  

\[ HC = \frac{LC \sin(\alpha - \theta) + L \sin \beta + D \cos \beta}{\cos \theta} \]  

**Skyline Geometry**

\[ h_1 = y_c - y_a \]  

\[ h_2 = y_b - y_c \]  

\[ h_3 = h_1 \]  

\[ h_4 = h_2 \]  

\[ d_1 = x_c - x_a \]  

\[ d_2 = x_b - x_c \]  

\[ d_3 = d_1 \]  

\[ d_4 = d_2 \]  

Where:

- \((x_a, y_a)\) = coordinates at the top of the headspar
- \((x_b, y_b)\) = coordinates at the top of the tailspar
(x_i, y_i) = coordinates at the top of the carriage

Segment Lengths (Rigid Link Model)

\[ L_1 = \sqrt{\dot{x}_1^2 + \dot{y}_1^2} \]  \hspace{1cm} (15)

\[ L_2 = \sqrt{\dot{x}_2^2 + \dot{y}_2^2} \]  \hspace{1cm} (16)

\[ L_3 = L_2 \]  \hspace{1cm} (17)

\[ L_4 = L_2 \]  \hspace{1cm} (18)

The resultant force (R) of the weight of each line segment can be expressed as:

\[ R_1 = \omega_1 L_1 \]  \hspace{1cm} (19)

\[ R_2 = \omega_2 L_2 \]  \hspace{1cm} (20)

\[ R_3 = \omega_3 L_3 \]  \hspace{1cm} (21)

\[ R_4 = \omega_4 L_4 \]  \hspace{1cm} (22)
Maximum Payload Calculation

A. Rigid Link

1. Skyline and Haulback Tensions Are Known

The first calculation of payload always assumes the skyline tension and the haulback tension are known.

For a three segment live or standing skyline, the known haulback tension is zero. Line tensions are calculated segment by segment as follows:

**Segment 1**

\[
H_1 = \frac{T_1 d_1}{L_1} \sqrt{1 - \left( \frac{2w_1}{T_1 d_1} \right)^2 + \frac{w_1 h_1 d_1}{2L_1}} \tag{23}
\]

\[
V_1^c = \frac{H_1 h_1}{d_1} - \frac{P_1}{2} \tag{24}
\]

**Segment 2**

\[
T_2^c = T_1^c \tag{26}
\]

\[
H_2 = \frac{T_2^c d_2}{L_2} \sqrt{1 - \left( \frac{2w_2}{T_2^c d_2} \right)^2 + \frac{w_2 h_2 d_2}{2L_2}} \tag{27}
\]
\[ V_2^C = \sqrt{(T_2^C)^2 - (H_2)^2} \quad (28) \]

**Segment 4**

\[ T_4^C = T_4^s + h_4 \omega_4 \quad (29) \]

\[ h_4 = \frac{T_4^C d_4}{L_4} \sqrt{1 - \left(\frac{2w_4}{T_4^C d_4}\right)^2 + \frac{w_4 h_4 d_4}{2L_4}} \quad (30) \]

\[ V_4^C = \sqrt{(T_4^C)^2 - (H_4)^2} \quad (31) \]

**Segment 3: Full Suspension**

\[ H_4 = H_3 - H_2 - H_4 \quad (32) \]

\[ V_3^C = \frac{H_3 h_3}{d_3} - \frac{R_3}{2} \quad (33) \]

\[ V_3^s = V_3^C \pm \omega_3 L_3 \quad (34) \]
The remaining Segment 3 tensions are calculated using Equations 33, 34 and 35.

**Payload: Full Suspension**

\[ W = V_1^c + V_2^c + V_3^c + V_4^c - W_c \]  

\[ T_s = \tan(\alpha) \left( H_1 - H_2 - H_4 \right) - \left( V_1^c - V_2^c + V_3^c - W_c \right) \]  

\[ T = \frac{V_3^c - H_2 - H_4}{\tan(\alpha)} \]  

**Payload: Partial Suspension**

\[ T_s = \frac{H_1 - H_2 - H_4}{\tan(\alpha)} \]  

Where the sign is (+) if the headspar is higher than the carriage and (-) if the headspar is lower than the carriage.
\[ N = \frac{T \cos\theta}{\sin\theta + \mu \cos\theta} \]  

\[ W = T \sin\theta + N \cos\theta - \mu N \sin\theta \]  

2. **Skyline and Mainline Tensions Are Known**  
   If the tension in the mainline (Eq. 35) is greater than the allowable mainline tension, the mainline tension is set equal to its allowable and the program solves for the haulback tensions and the payload. The skyline tensions calculated in Equations 23 - 28 are retained.

   **Segment 3**

\[ H_3 = \frac{T_3 d_3}{L_3} + \sqrt{1 - \left(\frac{2w_3}{T_3 d_3}\right)^2 + \frac{w_3 h_3 d_3}{2L_3}} \]  

\[ V_F = \frac{H_3 h_3}{d_3} - \frac{R_3}{2} \]  

**Segment 4: Full Suspension**

\[ H_4 = H_1 + H_5 - H_3 \]
Segment 4: Partial Suspension

\[ H_4 = \frac{\tan \left( \frac{H_1 + H_2 - H_3}{d_4} \right) \left( V_1^c + V_2^c + V_3^c - W_L \right) + \frac{R_4}{2}}{d_4 + \tan} \] (44)

\[ V_4^c = \tan \left( \frac{H_1 + H_2 - H_4}{d_4} \right) \left( V_1^c + V_2^c + V_3^c - W_L \right) \] (45)

Payload

The full suspension payload is calculated using Equation 37 and the partial suspension payload is calculated using Equations 38 - 41.

3. Mainline Limited

Where the assumptions used by Tobey (1981) are applicable, Tobey's direct solution for payload is used. The equations used in this program are taken directly from that paper. If Tobey's assumptions are not applicable, this program solves the maximum payload problem with a secant iterative search routine that adjusts the value of the maximum skyline tension and recycles through the equations listed above until the calculated mainline tension is within an acceptable tolerance of the allowable mainline tension.
4. Running Skyline

The maximum payload for the running skyline analysis uses Equations 23 - 28 and 32 - 41. The following equations are substituted for Equations 29 - 31.

\[ H_4 = H_2 \] (46)

\[ V_4^c = V_2^c \] (47)

B. Catenary

The solution order of the individual line segments is the same as described for the rigid link model. In the catenary model, all segment tensions can be calculated using one of two methods. If the horizontal component of the tension \( H \) is known, a direct solution is possible. If \( H \) is not known, an iterative solution method must be used. In both cases, the solution begins by calculating the catenary parameter \( n \).

Solution When \( H \) is Known

\[ m = \frac{H}{W} \] (48)

\[ s = \sqrt{h^2 + \left[ 2m \sinh \left( \frac{d}{2m} \right) \right]^2} \] (49)
Solution When H is not Known

The program first selects the upper end tension of the segment. If only the lower end tension is known, the upper end tension is calculated by

\[ T_U = \frac{\omega}{2} \left[ \frac{d}{2m} \right] \cot h \left( \frac{d}{2m} \right) + h \]  

(50)

\[ V_u = \sqrt{(T_u)^2 - (H)^2} \]  

(51)

\[ V_L = V_u - \omega L \]  

(52)

\[ T_e = \sqrt{(V_L)^2 + (H)^2} \]  

(53)

The catenary parameter \( m \) is calculated using an iterative approximation. The resultant force of the weight of the segment \( R \) and the distance \( e \) from the point at which \( R \) acts to the lower end of the segment are calculated using the following equations
Equations 57 - 63 are then repeated until consecutive calculations of \( \pi \) are within an acceptable tolerance.

\[
a = 1 + \frac{h^2}{d^2} \tag{57}
\]

\[
b = \frac{2 \, R \, h \, e}{d^2} \tag{58}
\]

\[
c = \frac{r^2 \, e^1 - r^2}{d^2} \tag{59}
\]

\[
m = \frac{-b + \sqrt{b^2 - 4ac}}{2 \, a} \tag{60}
\]

\[
s = \sqrt{h^2 + \left[2m \sinh\left(\frac{d}{2m}\right)\right]^2} \tag{61}
\]
The horizontal component of the segment's tension \( H \) is calculated by

\[
H = \omega m
\]  \hspace{1cm} (64)

The vertical components of the tension at the upper and lower ends of the segment are calculated using Equations 51 and 52.
APPENDIX B

PARTIAL SUSPENSION
APPENDIX B

PARTIAL SUSPENSION ANALYSIS FOR STANDING SKYLINE

Most standing skyline partial suspension algorithms have calculated the carriage height and then used iterative solution methods to solve for the log-to-ground geometry (Carson 1975, Tobey 1980, Falk 1981). This technique includes the calculation of the partial suspension load factors, v and h, which then permit the direct calculation of payload when the skyline tension is the factor limiting the maximum payload. Tobey and Falk provide direct solutions of the mainline limited payload using the partial suspension load factors.

Tobey’s algorithm is applicable only to uphill yarding and to three segment skyline systems or running skylines. Since the derivation of the mainline limiting case involves some fairly complex substitutions, it is not readily adapted to other systems such as a live skyline with a haulback.

Falk’s algorithm is a relatively simple calculation of the mainline limited case, but it makes a simplifying assumption that introduces a small error in the mainline limited payload calculation (Falk 1981). It can be used for either uphill or downhill yarding. It still models only three segment systems or running skylines, but can be easily
adapted to model four segment systems. Neither of these models consider elastic stretch in the skyline.

Sessions (1990) provides a direct solution of the skyline limited payload that does not require the calculation of the partial suspension load factors. The mainline limited case is then calculated by adjusting the skyline tension until the mainline tension equals its allowable.

This program combines elements of both methods in an effort to reduce the number of iterative searches required in solving payload analysis problems. Reducing the number of iterative searches reduces the computational time required to solve the problems.

Rigid Link Algorithm

In all cases, the program first assumes the skyline is the limiting line. That being the case, the program solves the skyline limiting case with Sessions' direct solution that does not require the partial suspension load factors. If all lines are at or below the allowable tension, the solution has been found and no other calculations are necessary.

If the mainline tension exceeds its allowable, the mainline limited solution is calculated. If the assumptions are appropriate, Tobey's direct solution is used. Once the payload from the skyline limited case is known, the partial
suspension load factors can be calculated by the following equations.

\[
v = \frac{V_1c + V_2c + V_3c + V_4c - W_c}{\text{payload}}
\]  \hspace{1cm} (65)

\[
h = \frac{H_1 + H_3 - H_2 - H_4}{\text{payload}}
\]  \hspace{1cm} (66)

If \(v\) and \(h\) are known, all variables needed for Tobey's algorithm are known. If Tobey's algorithm cannot be used, for example, in the case of downhill yarding or where elastic stretch is taken into account, an iterative solution procedure is used. The iterative algorithm used here involves a secant search routine that adjusts the skyline tension until the calculated mainline tension equals the allowable mainline tension.

**Catenary Algorithm**

Like the rigid link algorithm, the catenary algorithm begins by assuming that the skyline tension is limiting. The skyline tensions are calculated using the catenary relationships described by Carson (1977). Although this requires an iterative search for the catenary parameter \(m\), the search normally converges very quickly.

The initial estimate of the mainline tension is made using the rigid link equations described above. The
The horizontal tension component is then adjusted, if necessary, through the partial suspension load factors. Tobey (1980) has shown that v and h are dependent only on the log-to-ground geometry and are independent of the method used to calculate the tensions in the working lines. That being the case, the partial suspension load factors calculated using the rigid link estimate can be used to adjust the mainline tension for the catenary solution.

The steps in adjusting the mainline tension and calculating the payload are as follows:
1. Calculate the payload using rigid link assumptions for the mainline and catenary assumptions for all other lines.
2. Calculate the partial suspension load factors, v and h
3. Calculate the vertical component of the mainline tension (V1) using catenary relationships. Use the rigid link calculation of the horizontal component of the mainline tension (H3) as the initial estimate of H3 in the catenary equations.
4. Calculate the payload using the value of $V_{1c}$ from Step 3 and the vertical components of the tension in the other line segments from Step 1.

$$\text{payload} = \frac{V_{1c} + V_{2c} + V_{3c} + V_{4c} - V_c}{v}$$ (68)

5. Using the payload calculated in Step 4, calculate the partial suspension load factor $h$

$$h = \frac{H_1 + H_3 - H_2 - H_4}{\text{payload}}$$ (69)

Compare $h$ to the value of $h$ calculated in Step 1

6. Often, especially when the mainline is near the safe working load, the rigid link estimate of $H_3$ will yield a catenary solution that is within the acceptable tolerance. In that case the solution has been found.

7. If the solution from Step 5 is not within an acceptable tolerance, adjust $H_3$ and repeat Steps 3 to 5 until the calculated value of $h$ is within the acceptable tolerance.
In a standing skyline load path analysis, the payload is known and the analyst solves for the line tensions and payload clearances at each terrain point. A commonly used technique is to estimate the skyline tension and then solve for the payload in the same manner as is done for the maximum payload problem. A search technique such as the secant method or a gradient search is then used to make successive approximations of the skyline tension until the resulting calculated payload is within an allowable tolerance of the known payload. Earlier standing skyline load path analysis algorithms have used the elliptical load path (Carson, 1976) to determine the height of the carriage. For the partial suspension case, a second iterative solution step is required to solve for the log-to-ground angle $\alpha$, the angle of the tagline $\beta$, and the height of the leading edge of the payload.

Kendrick and Sessions (1991) proposed a solution algorithm for the standing skyline load path analysis that required only one iterative solution step. That method differed from earlier algorithms primarily by basing the first search routine on approximations of the height of the payload rather than on the skyline tension. Once the height
of the payload is known (or assumed), $a$, $\beta$ and the height of the carriage $H$ can be calculated directly using equations 2 to 4 in Appendix A.

The skyline tension can be calculated using the relationship between the stretched skyline length and the unstretched skyline length. Using the rigid link model, once the carriage height is known, the stretched skyline length can be calculated directly using Equation 1 of this paper. By rearranging Equation 1, the average tension in the skyline can be calculated directly

$$T_{\text{ave}} = \left( \frac{S_{\text{end}}}{S_0} - 1 \right)AR$$

The above solution becomes more complicated in the catenary model. In the catenary model, the stretched skyline length is not the sum of the straight line distances between the carriage and the two supports as it is in the rigid link model. Rather, the line length for any given carriage height can vary depending on the tension in the line. A second iterative search routine is required to adapt the load path algorithm to the catenary model.

The solution steps in the catenary model are as follows:

1. Estimate the payload height as in the rigid link model
2. Estimate the tension in the skyline
3. Calculate the stretched skyline length using catenary
relationships and the known geometry of the carriage and the supports

4. Calculate the stretched skyline length using Equation 1

5. Repeat Steps 2 through 4 until the two calculated values of the stretched skyline length are within an acceptable tolerance

6. Calculate the payload using the skyline tension and line length calculated in Steps 2 through 5

7. Repeat Steps 1 through 6 until the calculated payload is within an acceptable tolerance of the known payload.

**Initial Estimate of Payload Height**

Experience has shown that iterative solution methods will sometimes fail to converge on to an acceptable tolerance when the payload is near the transition between full suspension and partial suspension. Since it is not initially known whether the payload is fully or partially suspended, this program forces the solution into either full or partial suspension after the initial estimate of the log height.

1. The first estimate of the log height assumes partial suspension with the log almost vertical.

2. If the resulting payload is less than the known payload, the skyline must be lowered to improve deflection. Therefore, the payload must be partially suspended. The log height is then lowered in equal intervals until the calculated payload is greater than
the known payload. At this point, the correct value of the log height has been bracketed and is quickly found using a secant search.

3. If the payload calculated using the initial estimate with the log almost vertical is greater than the known payload, the payload must be fully suspended, since raising the skyline to lower the payload would necessarily produce a fully suspended payload. The program then raises the skyline until the correct payload height has been bracketed and then switches to a secant search routine. In raising the skyline, the program first checks to make sure the estimate of the log height does not put the carriage above the chord slope, and adjusts the log height estimate if necessary.

4. In some cases, the initial log height estimate that assumes partial suspension will yield a payload estimate indicating that the payload is actually fully suspended, while a full suspension payload estimate with a log clearance equal to zero indicates that the payload is partially suspended. This is the situation discussed earlier in which the iterative search does not converge on an answer. In that case, the program forces a full suspension analysis. The program reports a payload clearance of zero even though the calculated
carriage clearance might indicate a log clearance that is one to two feet below the ground surface.
APPENDIX D

LIST OF VARIABLES
## APPENDIX D

### LIST OF VARIABLES

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Definition</th>
</tr>
</thead>
</table>
| adjustskyline | \(-1 = \) adjust carriage height to keep skyline above ground profile between terrain point and headspar
|               | \(0 = \) do not adjust carriage height |
| ae            | Product of metallic cross-sectional area of skyline multiplied by the modulus of elasticity of skyline wire rope |
| aew           | Product of variable "ae" multiplied by unit weight of skyline |
| alpha         | The angle of the tagline |
| beta          | Log-to-ground angle |
| carriagetype  | 1 = nonclamping carriage
|               | 2 = clamping carriage with active mainline
|               | 3 = clamping carriage with slack mainline |
| carriagedepth | Depth of the carriage from point of tagline attachment to the skyline |
| carriageht    | Height of the bottom of the carriage above the ground |
| carriagewt    | Carriage weight |
| carriagex     | \(x\) coordinate of the carriage |
| carriagey     | \(y\) coordinate of the skyline at the carriage |
| chokepoint    | Distance from the leading end of the log to the point of choker attachment |
| chordslope    | Slope of the line from the top of the headspar to the carriage |
| critical()    | \(-1 = \) shortest unstretched line length occurs at terrain point "tp"
|               | \(0 = \)
shortest unstretched line length does not occur at terrain point "tp"

criticalpayload: Minimum payload that can be taken to the landing from a given terrain point.
criticaltp: Standing skyline terrain point with the shortest maximum skyline length.
cy: Vertical distance from top of headspar to carriage.
d0: Horizontal distance between supports.
dir: 1 = tailspar terrain point is greater than headspar terrain point.
     -1 = headspar terrain point is greater than tailspar terrain point.
drumht: Height from ground to the center of the haulback drum.
dSpan: Horizontal distance between supports for multispan system.
dSpan(): Variable "dSpan" for a specific terrain point.
dtail: Horizontal distance from the carriage to the tailspar.
dtower: Horizontal distance from the headspar to the carriage.

EffectiveRadius: Effective radius of the haulback drum.
elev(): y-coordinate of top of support for multispan system.
emptyspans(): Unstretched line length of unloaded spans for multispan.
epsilon: Distance from log center of gravity to point of choker attachment.
fm: Function calculated for Newton-Raphson iterative search routine.
<table>
<thead>
<tr>
<th>Identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fprimem</td>
<td>First derivative of &quot;fm&quot;</td>
</tr>
<tr>
<td>fullclearance()</td>
<td>Height above ground of bottom of log for full suspension analysis</td>
</tr>
<tr>
<td>h</td>
<td>Ratio of the horizontal component of the tagline tension to the payload</td>
</tr>
<tr>
<td>h0</td>
<td>Vertical distance between two supports</td>
</tr>
<tr>
<td>H1</td>
<td>Horizontal component of the tension in line segment 1</td>
</tr>
<tr>
<td>H1XXX</td>
<td>Estimate of the horizontal component of the tension in line segment 1. Used in secant search routines. XXX = alpha-numeric designation specific to subroutine.</td>
</tr>
<tr>
<td>H2</td>
<td>Horizontal component of the tension in line segment 2</td>
</tr>
<tr>
<td>H2XXX</td>
<td>Estimate of the horizontal component of the tension in line segment 2. Used in secant search routines. XXX = alpha-numeric designation specific to subroutine.</td>
</tr>
<tr>
<td>H3</td>
<td>Horizontal component of the tension in the mainline</td>
</tr>
<tr>
<td>H3skylinelimited</td>
<td>Horizontal component of the tension in the mainline calculated with skyline at the maximum tension. Used in approximating tensions in mainline limited case.</td>
</tr>
<tr>
<td>H3XXX</td>
<td>Estimate of the horizontal component of the mainline tension. Used in secant routines. XXX = alpha-numeric designation specific to subroutine</td>
</tr>
<tr>
<td>H4</td>
<td>Horizontal component of the tension in line segment 4</td>
</tr>
<tr>
<td>H4rigidlink</td>
<td>Rigid link calculation of H4 used as first approximation in catenary calculations</td>
</tr>
<tr>
<td>H4XXX</td>
<td>Estimate of horizontal component of tension in line segment 4. Used in secant routines. XXX = alpha-numeric designation specific to subroutine.</td>
</tr>
<tr>
<td>Haulback</td>
<td>Maximum tension in the haulback line</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Haulbackwt</td>
<td>Unit weight of the haulback line</td>
</tr>
<tr>
<td>HBdiameter</td>
<td>Diameter of the haulback line</td>
</tr>
<tr>
<td>HBdrumradius</td>
<td>Haulback drum barrel radius</td>
</tr>
<tr>
<td>HBdrumwidth</td>
<td>Haulback drum flange width</td>
</tr>
<tr>
<td>HBlength</td>
<td>Total length of the haulback line</td>
</tr>
<tr>
<td>HBlineondrum</td>
<td>Length of line on the haulback line</td>
</tr>
<tr>
<td>HBrequired</td>
<td>Total length of haulback line required</td>
</tr>
<tr>
<td>hg</td>
<td>Estimate of the parameter (h) used in iterative calculations of payload</td>
</tr>
<tr>
<td>Horiz</td>
<td>Horizontal component of tension in unloaded skyline span</td>
</tr>
<tr>
<td>Horiz()</td>
<td>&quot;Horiz&quot; for a specific span</td>
</tr>
<tr>
<td>Hspan()</td>
<td>Vertical distance between tops of two spans</td>
</tr>
<tr>
<td>htail</td>
<td>Vertical distance from the tailspar to the skyline at the carriage location</td>
</tr>
<tr>
<td>htower</td>
<td>Vertical distance from the headspar to the skyline at the carriage location</td>
</tr>
<tr>
<td>inneryardinglimit</td>
<td>Inner yarding limit</td>
</tr>
<tr>
<td>interval</td>
<td>Increase or decrease in variable used in iterative search subroutines</td>
</tr>
<tr>
<td>KeepLifts</td>
<td>Variable used to temporarily store number of user specified lifts for live skyline analysis</td>
</tr>
<tr>
<td>keeppt</td>
<td>Variable used to temporarily store lift terrain point for live skyline analysis</td>
</tr>
<tr>
<td>lastsupport</td>
<td>Total number of supports in a multispansystem (including headspar and tailspar)</td>
</tr>
<tr>
<td>Layers</td>
<td>Number of layers of wire rope on haulback drum (not rounded)</td>
</tr>
<tr>
<td>LayersOnDrum</td>
<td>Number of layers of wire rope on haulback drum rounded down to the nearest integer</td>
</tr>
</tbody>
</table>
liftpoint()  Terrain point at which skyline is raised
liftpt()   -1 = terrain point is a lift point
          0 = terrain point is not a lift point
limt    Terrain point at which payload is limiting
linelength  Unstretched line length
linelength()  Unstretched line length at a specific terrain point
linewt       Unit weight of line segments 1 and 2
liveoption$ String variable indicating whether live skyline analysis will include lifts with variable carriage clearance or constant carriage clearance
logdiameter Log diameter in inches
loght     Height above ground of the front end of the log for partial suspension analysis or height above ground of the bottom of the log for full suspension analysis
loghtX   Estimate of "loght" used in iterative search routines. X = 1 or 2
loglength  Length of log from point of choker attachment to the bottom or trailing end of log
ltail    Rigid link calculation of the length of line segment 2
ltower  Rigid link calculation of the length of line segment 1
m        Intermediate variable used in iterative search routines
m1       Intermediate variable used in Newton-Raphson search routine
Mainline     Maximum tension in the mainline
Mainlinel  Intermediate storage of maximum tension in mainline
<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mainlinewt</td>
<td>Unit weight of the mainline</td>
</tr>
<tr>
<td>maxelev</td>
<td>Maximum y coordinate of the skyline in a multispan system</td>
</tr>
<tr>
<td>Maxhaulback</td>
<td>Maximum allowable haulback tension</td>
</tr>
<tr>
<td>MaxHBTorque</td>
<td>Maximum torque at the center of the haulback drum</td>
</tr>
<tr>
<td>Maxmainline</td>
<td>Maximum allowable mainline tension</td>
</tr>
<tr>
<td>Maxskyline</td>
<td>Maximum allowable skyline tension</td>
</tr>
<tr>
<td>minchordslope</td>
<td>Minimum (including negative values) chord slope from top of headspar to skyline at the carriage location</td>
</tr>
<tr>
<td>mindrag</td>
<td>Minimum height above ground of the leading end of the log for partial suspension analysis</td>
</tr>
<tr>
<td>minfly</td>
<td>Minimum height above ground of the bottom of the log for full suspension analysis</td>
</tr>
<tr>
<td>minline</td>
<td>Minimum line length required for a standing or multispan system</td>
</tr>
<tr>
<td>minline()</td>
<td>Minimum unstretched line length at a specific terrain point</td>
</tr>
<tr>
<td>n</td>
<td>Number of support being evaluated</td>
</tr>
<tr>
<td>normal</td>
<td>Component of the payload acting normal to the ground slope</td>
</tr>
<tr>
<td>outeryardinglimit</td>
<td>Outer yarding limit</td>
</tr>
<tr>
<td>oyl</td>
<td>Outer yarding limit for live skyline with lifts</td>
</tr>
<tr>
<td>part1</td>
<td>Intermediate variable</td>
</tr>
<tr>
<td>part2</td>
<td>Intermediate variable</td>
</tr>
<tr>
<td>part3</td>
<td>Intermediate variable</td>
</tr>
<tr>
<td>payload</td>
<td>Net payload</td>
</tr>
<tr>
<td>phase1</td>
<td>Maximum payload toggle</td>
</tr>
<tr>
<td></td>
<td>= -1 for maximum payload analysis</td>
</tr>
<tr>
<td></td>
<td>= 0 for load path analysis</td>
</tr>
</tbody>
</table>
phase2  Load path toggle
       = 0 for maximum payload analysis
       = -1 for load path analysis
recalc  True/false toggle to set recalculation of
tensions and payload
s       Catenary line length calculation
skip    True/false toggle to direct order of
calculations
Skyline Maximum tension at any point on the skyline
skylinewt Unit weight of the skyline
span    Number of span being analyzed for multispansystem
spanlength() Unstretched line length between two
       supports
stail   Catenary calculation of the length of line
       segments 2 and 4
stower  Catenary calculation of the length of line
       segments 1 and 3
support Number of skyline support with headspar = 1
support() Terrain point number of a specified
       support
support1() Terrain point number of support closest to
       headspar
support2() Terrain point number of support closest to
       tailspar
supportht() Height of a specified support
supportpoint(); -1 = terrain point has an intermediate
       support
            0 = terrain point does not have an
               intermediate support
suspensiontype  1 = full suspension
                2 = one end suspension
suspensiontype() Specifies "suspensiontype" for each terrain
sX Estimate of catenary line length used in iterative search routines
syst$ Description of type of yarding system to be analyzed
systemtype
1 = live skyline
2 = live skyline w/haulback
3 = running skyline
4 = standing skyline
5 = standing skyline w/haulback
6 = multispan skyline
7 = load path analysis (standing skyline)
8 = load path analysis (standing skyline w/haulback)

T() Skyline tension at top of a specified support
T1 Maximum tension in line segment 1
T1carriage Tension in line segment 1 at the carriage
T1torque Maximum tension in line segment 1 based on haulback torque for a running skyline
T1tower Tension in line segment 1 at the top of the headspar
T2 Maximum tension in line segment 2
T2carriage Tension in line segment 2 at the carriage
T2tail Tension in line segment 2 at the top of the tailspar
T3carriage Tension in line segment 3 at the carriage
T3tower Tension in line segment 3 at the top of the headspar
T4 Maximum tension in line segment 4
T4carriage Tension in line segment 4 at the carriage
T4tail Tension in line segment 4 at the top of the tailspar
taglength Length of tagline plus choker from
bottom of carriage to point of choker attachment to log

tailht | Height above ground of line segment 2 at tailspar
tailtp | Terrain point number of tailspar
tailx | x coordinate of tailspar
taily | y coordinate of tailspar
Tdelta | Change in skyline tension in binary search routine
Thaulback | Tension applied by yarder operator to haulback line for live or standing skyline system with haulback
theta | Ground slope at terrain point
ThisSpan | Number of the span being analyzed
Tmax | Maximum tension in line segments 1 and 2 for other than multispan system
TMaxMS | Maximum tension in line segments 1 and 2 for multispan system
TmaxX | Estimate of Tmax used in iterative search routines. X = 1 or 2
Tnextsupport | Tension at top of the next support in direction of tailspar
TotalLifts | Total number of lifts in live skyline analysis. Minimum = 1 unless constant carriage clearance option is used
totloglength | Total length of log
towerh | Height above ground of line segment 1 at the headspar
towertp | Terrain point number of headspar location
towerx | x coordinate of headspar
towery | y coordinate of headspar
tp | Terrain point number. Numbered
<table>
<thead>
<tr>
<th><strong>Ts</strong></th>
<th><strong>Upper</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ts</strong></td>
<td><strong>Upper</strong></td>
</tr>
<tr>
<td><strong>Tsupport</strong></td>
<td>Tension in skyline at top of support being analyzed</td>
</tr>
<tr>
<td><strong>Tsupport2</strong></td>
<td>Tension at top of support closest to tailspar for span being analyzed</td>
</tr>
<tr>
<td><strong>Tsupport2()</strong></td>
<td>Variable &quot;Tsupport2&quot; for a specified span</td>
</tr>
<tr>
<td><strong>TUpper</strong></td>
<td>Upper end skyline tension for an unloaded span</td>
</tr>
<tr>
<td><strong>u</strong></td>
<td>Log-to-ground coefficient of friction</td>
</tr>
<tr>
<td><strong>unloadedlength</strong></td>
<td>Length of the unloaded skyline for standing or multispan system</td>
</tr>
<tr>
<td><strong>unloadedskyline</strong></td>
<td>Maximum tension in an unloaded skyline</td>
</tr>
<tr>
<td><strong>UseHaulback</strong></td>
<td>Haulback toggle for live or standing skyline system</td>
</tr>
<tr>
<td><strong>UseMainline</strong></td>
<td>Mainline toggle for live or standing skyline system</td>
</tr>
<tr>
<td><strong>v</strong></td>
<td>Vertical component of tension in line</td>
</tr>
<tr>
<td><strong>Vlcarriage</strong></td>
<td>Vertical component of tension in line</td>
</tr>
<tr>
<td><strong>Vltower</strong></td>
<td>Vertical component of tension in line</td>
</tr>
<tr>
<td><strong>V2carriage</strong></td>
<td>Vertical component of tension in line</td>
</tr>
<tr>
<td><strong>V2tail</strong></td>
<td>Vertical component of tension in line</td>
</tr>
</tbody>
</table>

consecutively beginning with 1.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>V3carriage</td>
<td>Vertical component of tension in line segment 3 at the carriage</td>
</tr>
<tr>
<td>V3tower</td>
<td>Vertical component of tension in line segment 3 at the top of the headspar</td>
</tr>
<tr>
<td>V4carriage</td>
<td>Vertical component of tension in line segment 4 at the carriage</td>
</tr>
<tr>
<td>vertchange</td>
<td>Change in elevation from top of tailspar to top of headspar</td>
</tr>
<tr>
<td>Wh</td>
<td>Horizontal component of tagline tension</td>
</tr>
<tr>
<td>Wv</td>
<td>Vertical component of tagline tension</td>
</tr>
<tr>
<td>xsupport1()</td>
<td>x coordinate of the support closest to headspar for span being analyzed</td>
</tr>
<tr>
<td>xsupport2()</td>
<td>x coordinate of the support closest to tailspar for span being analyzed</td>
</tr>
<tr>
<td>ysupport1()</td>
<td>y coordinate of the support closest to headspar for span being analyzed</td>
</tr>
<tr>
<td>ysupport2()</td>
<td>y coordinate of the support closest to tailspar for span being analyzed</td>
</tr>
<tr>
<td>zXXXY</td>
<td>Intermediate variable in secant routines. XXX = alpha designation specific to subroutine. Y = 1 or 2</td>
</tr>
</tbody>
</table>
APPENDIX E

PROGRAM LISTING
REM CAT.BAS
REM Skyline analysis program using Catenary assumptions.
REM Revised 1/30/92, David Kendrick

REM This program will analyze live, standing, and running
REM skyline systems. Live and standing skylines can be
REM analyzed with or without a haulback line. Standing
REM skylines can be analyzed as either single or multiple span
REM systems. All systems can be analyzed for the maximum
REM payload. A load path analysis is also available for
REM single span standing skylines.
REM Live or standing skyline systems without a haulback
REM line can be analyzed for either a clamping or a
REM nonclamping carriage. The clamping carriage analysis can
REM include either an active mainline in which the mainline
REM passes through the carriage or a slack mainline in which
REM the weight and tension in the mainline are assumed to
REM be zero.

REM This program is written in Microsoft QuickBASIC

REM Declare subprograms to solve Catenary line segment
REM tensions

DECLARE SUB CatenaryTdh (T!, d!, h!, wI, Horiz!)
DECLARE SUB CatenaryHdh (Horizl, dl, hI, w!, TI)

CatenaryTdh calculates segment tensions and line length
using catenary assumptions when upper end tension (T),
horizontal distance between support and carriage (d), and
vertical distance between support and carriage (h) are
known.

CatenaryHdh calculates segment tensions and line length
using catenary assumptions when horizontal component of
tension (H), d and h are known.

DIM x(50), y(50), p(50), s(50), slp(50), beta(50), phi(50),
cy(50)

DEF FNquadratic (sign) = (-b + sign * SQR(b * b - 4 * a * c)) / 2 / a

DEF FNhorizontal (w, d, h, 1, T) = w * d * h / 2 / 1 + T * d
/ 1 * SQR(1 - (w * d / T / 2)^2)

DEF FNsinh (x) = (EXP(x) - EXP(-x)) / 2

DEF FNcosh (x) = (EXP(x) + EXP(-x)) / 2

DEF FNcoth (x) = FNcosh(x) / FNsinh(x)

DEF FNstretch (T, w, ht) = (T - w * ht / 2) / ae + 1
DEF FNarcsin (x) = ATN(x / SQR(1 - x * x))
DEF FNtension (T, sign, h, w) = T + sign * (h < 0) * ABS(h) * w
DEF FNw = (V1carriage + V2carriage + V3carriage + V4carriage - carriagewt) / v
DEF FNz = -phasel * (Mainline - Maxmainline) - phase2 * (payload - designpayload)
DEF FNtag = payload * v / SIN(alpha)

REM FNquadratic calculates the roots of the quadratic equation ax^2 + bx + c
REM FNhorizontal calculates the horizontal component of tension using rigid link assumptions
REM FNMainh calculates the hyperbolic sine of the function x
REM FNcosh calculates the hyperbolic cosine of the function x
REM x
REM FNcoth calculates the hyperbolic cotangent of the function x
REM FNstretch calculates the factor used to stretch or unstretch a line segment
REM FNw calculates the payload when the vertical components of tension at the carriage and the decimal percent of the payload in the vertical component of the tagline tension (v) are known. Used to recalculate the payload in iterative searches after v has been calculated
REM FNz calculates the error in calculated mainline tension or calculated payload for maximum payload analysis or load path analysis respectively.
REM Phasel = -1 and phase2 = 0 for max. payload analysis.
REM Phase1 = 0 and phase2 = -1 for load path analysis. Both were included in one function because this is essentially the only difference between the two types of analysis and including them in one function allows the same search routine to be used for both cases.
REM FNtag calculates the tagline tension.

REM This program is intended to be part of a larger program.
REM All program lines between here and the label "Calculations:" should be deleted.
REM All variables defined between this statement and the label "Calculations:" are necessary input variables that must be defined, either by the user or by program defaults, earlier in the program.

REM The "DATA" statement below is the set of profile coordinates used to test the program.
DATA 0,5000,100,4950,200,4925,300,4850,400,4800
DATA 500,4720,600,4780,700,4800

systemtype = 1 '1 = live skyline
designpayload = 7000
carriagetype = 1
'liftpoint(l) = 3
TotalLifts = 1
towertp = 1
tailtp = 8

'2 = live skyline w/haulback
'3 = running skyline
'4 = standing skyline
'5 = standing skyline w/haulback
'6 = multispan
'7 = standing skyline load path
'8 = standing skyline w/haulback load path
'adjustskyline = -1
'-1 = adjust skyline clearance so
'that it does not touch the ground
' 0 = skyline not adjusted
liveoption$ = " Constant Clearance"
'Live skyline with
'constant clearance
liveoption$ = " Lifts"
'suspensiontype = 2
'1 = full suspension
'2 = front end suspension
designpayload = 7000
'Design payload used in load path
calculations
carriagetype = 1
'1 = nonclamping carriage
'2 = clamping carriage w/active
mainline
'3 = clamping carriage w/slack
mainline
'liftpoint(l) = 3
'Terrain point number at which the
nth lift is specified for a live
skyline with liveoption$ = Lifts
'Total number of lifts for a live
skyline
'Must be >= 1 if liveoption$ = Lifts
towertp = 1
'tailsparr terrain point
tailtp = 8
'Tailsparr terrain point

IF towertp > tailtp THEN dir = -1 ELSE dir = 1
outeryardinglimit = tailtp - dir
'Outer yarding limit
inneryardinglimit = towertp + dir
'Inner yarding limit
totalpoints = towertp + tailtp - 1
'x(tp) and y(tp)
'represent the
FOR tp = 1 TO totalpoints
'READ x(tp), y(tp)
NEXT
towerht = 50
'Height from ground to top of headspar
tailht = 30
'Height from ground to top of tailspar
Maxskyline = 37000
'Maximum allowable skyline tension
Maxmainline = 30000
'Maximum allowable mainline tension
Maxhaulback = 17100
'Maximum allowable haulback tension
Thaulback = 15000
'line for a live or standing skyline
'w/haulback
'Program will always apply this
tension unless doing so will cause
'other lines to exceed the maximum
HBdiameter = 3 / 4
HBlength = 5000
HBdrumradius = 9
HBdrumwidth = 30
MaxHBtorque = 40000

'HBdiameter = 3 / 4
'Length of the haulback line
'HBdrumradius = 9
'Unit weight of the mainline plus slackpulling line
'mainlinewt = 2.34

'skylinewt = 2.34
'inaintlinewt = 1.85
'mainlinewt = 1.85

'Unit weight of the mainline plus slackpulling line
'HBdrumwidth = 30
'maxlinewt = 2.57

'Unit weight of the mainline plus slackpulling line

'Height from the ground to the center
'drumht = 10
'of the haulback drum (ft.)
'
haulbackwt = 1.04
'Point of tagline attachment on the
'carriagedepth = 0
'carriage
'
totloglength = 32
'total length of the payload

'Point of tagline attachment on the
'chokepoint = 0
'choker attachment
'
loglength = totloglength - chokepoint
logdiameter = 2

diameter of log (ft.)
taglength = 10

'Length of tagline from point of
'choker attachment to carriage

aew = 3500000

'AE/w; ratio of metallic
'cross-sectional area of skyline
'time modulus of elasticity of
'skyline divided by the unit weight
'of the skyline

mindrag = 2

'Minimum front end clearance of the
'log for one-end suspension

minfly = 2

'Minimum clearance of the bottom of
'the log for full suspension

u = .60001

'Log-to-ground friction coefficient
'total number of supports (including
'headspar and tailspar) for multispan
'system

ex, lastsupport = 3
'with a headspar, a tailspar, and one
'intermediate support

support(2) = 1

Terrain point of the nth support for
'a multispan system
'headspar = support(1)
tailspar = support(lastsupport)
intermediate support #1 = support(2)

'Only intermediate support points need
'to be specified since the following
'program assumes the headspar is
'support #1 and the tailspar is the
'lastsupport.
supportht(2) = 100  'Height of the nth support. Only
           ' needed for intermediate supports.

REM The following loop defines whether full or partial
REM suspension is required at each terrain point.
REM Since the user should be able to specify a different
REM clearance type for each terrain point, this loop
REM should be deleted and the variable suspensiontype(tp)
REM included in the user interface.

FOR tp = inneryardinglimit TO outeryardinglimit STEP dir
    suspensiontype(tp) = suspensiontype
NEXT

Calculations:

towerx = x(towertp)  'Establish geometry for
towery = y(towertp) + towerht  'headspar and tailspar
tailx = x(tailtp)
taily = y(tailtp) + tailht
ae = aew * skylinewt

loght = mindrag
minchordslope = -10 ^ 10
vertchange = taily - towery

IF tailx > towerx THEN dir = 1 ELSE dir = -1

ON systemtype  GOSUB live, LiveHB, Running, standing, StandingHB, multispan, standing, StandingHB

END

live:  'Live skyline, no haulback (shotgun or flyer)

syst$ = "Live Skyline"
GOSUB PrintHeader
linewt = skylinewt
V4carriage = 0
H4 = 0
Haulback = 0
SELECT CASE liveoption$

'Sets conditions for live
' skyline analysis.
"Constant Clearance" analyses
' each terrain point at a user
' specified minimum clearance.
"Lifts" analyses the system
' with the skyline raised at
' user specified terrain points.
'When the "Lifts" option is
' used, the system is analyzed
' as a standing skyline with the
 CASE "Constant Clearance"
 FOR tp = inneryardinglimit TO outeryardinglimit STEP dir
 Tmax = Maxskyline
 carriagex = x(tp)
 IF suspensiontype(tp) = 1 THEN
 loght = minfly
 GOSUB FullSuspension
 ELSE
 GOSUB PartialSuspension
 END IF
 GOSUB SkylineGeometry
 IF adjustskyline THEN GOSUB LineAdjustment
 SELECT CASE carriagetype
 CASE IS = 3
 GOSUB T1Limited
 IF T2 > Tmax THEN GOSUB T2Limited
 Skyline = Maxskyline
 Mainline = 0
 CASE ELSE
 GOSUB SkylineTensions
 GOSUB MainlineTensions
 IF Mainline > Maxmainline THEN
 Mainline = Maxmainline
 CALL CatenaryTdh(Mainline, dtower, htower, mainlinewt, H3)
 T3carriage = FNtension(Mainline, 1, htower, mainlinewt) ^ 2)
 V3carriage = SGN(-htower) * SQR(T3carriage ^ 2 - H3 ^ 2)
 GOSUB SkylineAdjustment
 END IF
 IF carriagetype = 2 THEN
 IF FNtag > Maxmainline + htower
 * mainlinewt THEN
 GOSUB SkylineAdjustmentCC
 END IF
 END IF
 END SELECT
 GOSUB PrintAnswer
 NEXT
 CASE ELSE
 phase1 = -1
 phase2 = 0
 GOSUB SetLiftPoints
oyl = outeryardinglimit
tp = inneryardinglimit
criticalpayload = 10 + 10
FOR outeryardinglimit = inneryardinglimit TO oyl
  STEP dir
  GOSUB Shortline
  IF minline(outeryardinglimit) = minline(outeryardinglimit - dir) THEN
    tp = outeryardinglimit
    GOSUB LiftRoutine
  ELSE
    FOR tp = inneryardinglimit TO outeryardinglimit STEP dir
      GOSUB LiftRoutine
    NEXT tp
  END IF
  GOSUB PrintAnswer
  tp = outeryardinglimit + dir
  NEXT outeryardinglimit
PrintLifts = 0
END SELECT
RETURN
LiftRoutine:
  UseHaulback = 0
carriagex = x(tp)
Tmax = Maxskyline
SELECT CASE carriagetype
  CASE IS = 3
    GOSUB GeometrySearch
    GOSUB TLimited
    IF T2 > Tmax THEN GOSUB T2Limited
    Skyline = Maxskyline
    Mainline = 0
  CASE ELSE
    GOSUB StandingSkylineTensions
    Skyline = Tmax
    IF Mainline > Maxmainline THEN GOSUB SkylineAdjustmentSS
    IF carriagetype = 2 THEN
      IF Fntag > Maxmainline + htower * mainlinewt THEN
        GOSUB SkylineAdjustmentCC
      END IF
    END IF
  END SELECT
END IF
END SELECT
IF tp = outeryardinglimit AND suspensiontype(tp) = 1 THEN
  susp$ = "Full"
ELSEIF tp = outeryardinglimit AND suspensiontype(tp) = 2 THEN
susp$ = "Partial"
ELSE
END IF
IF payload < criticalpayload THEN
criticalpayload = payload
limtp = tp
END IF
RETURN

LiveHb: 'Live skyline with haulback (Slackline)'
syst$ = "Live Skyline with Haulback"
GOSUB PrintHeader
linewt = skylinewt
Tmax = Maxskyline
SELECT CASE liveoption$
CASE "Constant Clearance"
FOR tp = inneryardinglimit TO outeryardinglimit STEP dir
carriagex = x(tp)
IF suspensiontype(tp) = 1 THEN
loght = minfly
GOSUB FullSuspension
ELSE
GOSUB PartialSuspension
END IF
GOSUB SkylineGeometry
IF adjustskyline THEN GOSUB LineAdjustment
GOSUB HaulbackTensions
GOSUB SkylineTensions
GOSUB MainlineTensions
IF Mainline > Maxmainline THEN GOSUB HaulbackAdjustment
GOSUB PrintAnswer
NEXT
CASE ELSE
PrintLifts = -1
phasel = -1
phase2 = 0
GOSUB SetLiftPoints
cyl = outeryardinglimit
tp = inneryardinglimit
criticalpayload = 10 ^ 10
FOR outeryardinglimit = inneryardinglimit TO cyl
STEP dir
GOSUB Shortline
IF minline(outeryardinglimit) ~ minline(outeryardinglimit - dir) THEN
tp = outeryardinglimit
GOSUB LiftRoutineHB
ELSE

FOR tp = inneryardinglimit TO outeryardinglimit STEP dir
  GOSUB LiftRoutineHB
NEXT tp
END IF
GOSUB PrintAnswer
  tp = outeryardinglimit + dir
NEXT outeryardinglimit
PrintLifts = 0
END SELECT
RETURN

LiftRoutineHB:
UseHaulback = -1
carriagex = x(tp)
Tmax = Maxskyline
GOSUB StandingSkylineTensions
Skyline = Tmax
IF Mainline > Maxmainline THEN GOSUB HaulbackAdjustmentSS
  IF tp = outeryardinglimit AND suspensiontype(tp) = 1 THEN
    susp$ = "Full"
  ELSEIF tp = outeryardinglimit AND suspensiontype(tp) = 2 THEN
    susp$ = "Partial"
  ELSE
  END IF
IF payload < criticalpayload THEN
  criticalpayload = payload
  limtp = tp
END IF
RETURN

Running:
  'Running Skyline
syst$ = "Running Skyline"
GOSUB PrintHeader
FOR tp = inneryardinglimit TO outeryardinglimit STEP dir
carriagex = x(tp)
linewt = haulbackwt
Tmax = Maxhaulback
IF suspensiontype(tp) = 1 THEN
  light = minfly
  GOSUB FullSuspension
ELSE
  GOSUB PartialSuspension
END IF
GOSUB SkylineGeometry
IF adjustskyline THEN GOSUB LineAdjustment
GOSUB TorqueLimited
GOSUB SkylineTensions
STANDING:

'Standing Skyline, no haulback'

'Standing Skyline load path with no haulback

SELECT CASE syste

CASE IS = 4
phas1 = -1
phas2 = 0
syst$ = "Standing Skyline"
GOSUB PrintHeader

CASE IS = 7
phas1 = 0
phas2 = -1
syst$ = "Load Path; Standing Skyline"
GOSUB PrintHeader

END SELECT

linewt = skylinewt
Tmax = Maxskyline
V4carriage = 0
H4 = 0
UseHaulback = 0
GOSUB Shortline
GOSUB UnloadedTension
FOR tp = inneryardinglimit TO outeryardinglimit STEP dir
carriagex = x(tp)
Tmax = Maxskyline
SELECT CASE carriagetype
CASE IS = 3
GOSUB GeometrySearch
GOSUB T2Limited
IF T2 > Tmax THEN GOSUB T2Limited
Skyline = Maxskyline

END IF
Haulback = Skyline
Skyline = 0
GOSUB PrintAnswer

NEXT
RETURN

STANDING:

'Standing Skyline, no haulback'

'Standing Skyline load path with no haulback

SELECT CASE syste

CASE IS = 4
phas1 = -1
phas2 = 0
syst$ = "Standing Skyline"
GOSUB PrintHeader

CASE IS = 7
phas1 = 0
phas2 = -1
syst$ = "Load Path; Standing Skyline"
GOSUB PrintHeader

END SELECT

linewt = skylinewt
Tmax = Maxskyline
V4carriage = 0
H4 = 0
UseHaulback = 0
GOSUB Shortline
GOSUB UnloadedTension
FOR tp = inneryardinglimit TO outeryardinglimit STEP dir
carriagex = x(tp)
Tmax = Maxskyline
SELECT CASE carriagetype
CASE IS = 3
GOSUB GeometrySearch
GOSUB T2Limited
IF T2 > Tmax THEN GOSUB T2Limited
Skyline = Maxskyline

END IF
Haulback = Skyline
Skyline = 0
GOSUB PrintAnswer

NEXT
RETURN
Mainline = 0
CASE ELSE
GOSUB StandingSkylineTensions
Skyline = Tmax
IF Mainline - Maxmainline > .5 THEN GOSUB SkylineAdjustmentSS
Skyline = Tinax
IF phase2 AND ABS(FNz) > .5 THEN GOSUB SkylineAdjustmentSS
IF carriageType = 2 THEN
   IF FNtag > Maxmainline + hTower * mainlineW THEN
      GOSUB SkylineAdjustmentCC
END IF
END IF
CASE IS = 5
   phasell = -1
   phase2 = 0
   syst$ = "Standing Skyline with Haulback"
   GOSUB PrintHeader
CASE IS = 8
   phasell = 0
   phase2 = -1
   syst$ = "Load Path; Standing Skyline with Haulback"
   GOSUB PrintHeader
END SELECT
linewt = skylineW
Tmax = Maxskyline
GOSUB UnloadedTension
FOR tp = innerYardingLimit TO outerYardingLimit STEP dir
   UseHaulback = -1
   carriageX = x(tp)
   Tmax = Maxskyline
   GOSUB StandingSkylineTensions
   Skyline = Tmax
   IF phase1 THEN
      IF Mainline - Maxmainline > .5 THEN GOSUB
   END IF
   IF phase2 THEN
      IF Mainline - Maxmainline > .5 THEN GOSUB
   END IF
   IF carriageType = 2 THEN
      IF FNtag > Maxmainline + hTower * mainlineW THEN
         GOSUB SkylineAdjustmentCC
      END IF
   END IF
   END IF
NEXT
PRINT USING "Unloaded Skyline Tension = ###,### lbs."
   UnloadedTension
PRINT USING "Unloaded Skyline Length = ###,### ft."
   UnloadedLineLength
RETURN
StandingHB: 'Standing Skyline with haulback
   or
   'Standing Skyline load path with haulback line

SELECT CASE systype
   CASE IS = 5
      phasell = -1
      phase2 = 0
      syst$ = "Standing Skyline with Haulback"
      GOSUB PrintHeader
   CASE IS = 8
      phasell = 0
      phase2 = -1
      syst$ = "Load Path; Standing Skyline with Haulback"
      GOSUB PrintHeader
END SELECT
linewt = skylineW
Tmax = Maxskyline
GOSUB Shortline
GOSUB UnloadedTension
FOR tp = innerYardingLimit TO outerYardingLimit STEP dir
   UseHaulback = -1
   carriageX = x(tp)
   Tmax = Maxskyline
   GOSUB StandingSkylineTensions
   Skyline = Tmax
   IF phase1 THEN
      IF Mainline - Maxmainline > .5 THEN GOSUB
   END IF
   IF phase2 THEN
      IF Mainline - Maxmainline > .5 THEN GOSUB
   END IF
   IF carriageType = 2 THEN
      IF FNtag > Maxmainline + hTower * mainlineW THEN
         GOSUB SkylineAdjustmentCC
      END IF
   END IF
   END IF
NEXT
PRINT USING "Unloaded Skyline Tension = ###,### lbs."
   UnloadedTension
PRINT USING "Unloaded Skyline Length = ###,### ft."
   UnloadedLineLength
RETURN
HaulbackAdjustmentSS
IF Mainline - Maxmainline > .5 THEN GOSUB SkylineAdjustmentSS
ELSE
   IF phase2 AND ABS(FNz) > .5 THEN GOSUB SkylineAdjustmentSS
END IF
GOSUB PrintAnswer
NEXT
PRINT USING "Unloaded Skyline Tension = ##,### lbs."
UnloadedTension
PRINT USING "Unloaded Skyline Length = ##,### ft."
unloadedlinelength
RETURN

multispan: 'Multispan skyline, with or without haulback 'Will not analyze load path
support(1) = tower1
supportht(1) = towerht
support(lastsupport) = tailtp
supportht(lastsupport) = tailht
phase1 = -1
phase2 = 0
syst$ = "Multispan Skyline"
GOSUB PrintHeader
linewt = skylinewt
Tmax = Maxskyline
ThbDefault = Thaulback
FOR tp = inneryardinglimit TO outeryardinglimit STEP dir
   FOR support = 1 TO lastsupport
      IF tp = support THEN
         supportpoint(tp) = -1
      ELSE
         supportpoint(tp) = 0
      END IF
   NEXT support
NEXT tp
GOSUB SpanLengths
GOSUB AssignSpans
GOSUB ShortlineMS
GOSUB UnloadedTensionMS
FOR tp = inneryardinglimit TO outeryardinglimit STEP dir
   IF NOT supportpoint(tp) THEN
      FOR span = 1 TO lastsupport - 1
         IF x(tp) * dir > xsupport1(span) * dir THEN
            IF x(tp) * dir < xsupport2(span) * dir THEN
               ThisSpan = span
            END IF
         END IF
      NEXT span
   END IF
END IF
END IF
NEXT span

recalc:

IF Thaulback > 0 THEN UseHaulback = 1 ELSE UseHaulback = 0

carriagex = x(tp)

tmax = Maxskyline

GOSUB StandingSkylineTensions

Skyline = Tmx

IF Thaulback > 0 THEN

IF FNz > .5 THEN GOSUB HaulbackAdjustmentSS

END IF

IF FNz > .5 THEN GOSUB SkylineAdjustmentSS

IF Haulback = Thaulback > .5 THEN GOSUB SkylineAdjustmentHB

END IF

GOSUB PrintAnswer

NEXT tp

PRINT USING "Unloaded tension at top of headspar = ##,### lbs."; UnloadedTension

PRINT USING "Unloaded skyline length = ##,### ft."; unloade linelength

RETURN

FullSuspension: 'Log geometry for full suspension analysis

carriagey = y(tp) + loght + loglength + taglength +
carriagedepth

fullclearance(tp) = loght

alpha = 1.570796

'Set alpha = vertical

'Alpha = tagline angle in

'radians

RETURN

PartialSuspension: 'Log geometry for partial suspension analysis

epsilon = totloglength / 2 - chokepoint

theta = ATN((y(tp) - y(tp + dir)) / ABS(x(tp + dir) - x(tp)))

beta = FNarcsin(loght * COS(theta) / loglength)

IF u = -TAN(theta) THEN u = u + .00001 'u must not =

'slope percent

'or fatal error

'will occur in

'alpha calc.

IF -u > TAN(theta) THEN beta = 180 / 57.296 - beta

part1 = COS(theta) - u * SIN(theta)

part2 = SIN(theta) + u * COS(theta)
part3 = TAN(theta + beta)
alpha = ATN((loglength / epsilon) * (part3 + logdiameter/loglength)) + ((loglength / epsilon - (logdiameter / epsilon) * part3 - 1) * part1 / part2)
carriageht = (taglength * SIN(alpha - theta) + logdiameter * COS(beta) + loglength * SIN(beta)) / COS(theta)
carriagey = y(tp) + carriageht + carriagedepth
partialclearance(tp) = loght
RETURN

SkylineGeometry: 'Establish geometry of skyline segments.
dtower = ABS(carriagex - towerx)
htower = carriagey - towery
dtail = ABS(tailx - carriagex)
htail = tally - carriagey
ltower = SQR(dtower^2 + htower^2) 'Rigid link estimate
ltail = SQR(dtail^2 + htail^2) 'Rigid link estimate
RETURN

SkylineTensions: 'Calculate segment 1 and 2 tensions
Ttower = FNtension(Tmax, 1, -vertchange, linewt)
SELECT CASE syst$
CASE "Running Skyline"
IF Titorque < Ttower THEN Ttower = Titorque
CASE ELSE
END SELECT
T1 = FNtension(Ttower, -1, -htower, linewt) 'Max. tension in 'segment 1
CALL CatenaryTdh(T1, dtower, htower, linewt, H1)
'THorizontal tension component in segment 1
Ticarriage = FNtension(T1, 1, htower, linewt)
Vicarriage = SGN(-htower) * SQR(T1carriage^2 - H1^2)
T2 = FNtension(T1carriage, -1, -htail, linewt) 'Max. tension in 'segment 2
CALL CatenaryTdh(T2, dtail, htail, linewt, H2)
'THorizontal tension component in segment 2
V2carriage = SGN(htail) * SQR(T1carriage^2 - H2^2)
Skyline = FNtension(Ttower, -1, -vertchange, linewt)
RETURN

MainlineTensions: 'Calculates mainline tensions when skyline tensions are known
HaulbackTensions: 'Calculates tensions in the haulback line of a four segment live or standing skyline when the operator applied haulback tension is specified by user.

\[ T_{4tail} = FN_{tension}(\text{Thaulback}, 1, \text{vertchange}, \text{haulbackwt}) \]

\[ T_4 = FN_{tension}(T_{4tail}, -1, \text{htail}, \text{haulbackwt}) \]

CALL CatenaryThdh(T4, \text{dtail}, \text{htail}, \text{haulbackwt}, H4)

\[ T_{4carriage} = T_{4tail} - \text{htail} \times \text{haulbackwt} \]

IF \( T_{4carriage}^2 < H_4^2 \) THEN \( T_{4carriage} = H_4 \)

\[ V_{4carriage} = SGN(\text{htail}) \times \sqrt{T_{4carriage}^2 - H_4^2} \]

\[ Haulback = FN_{tension}(T_{4tail}, -1, \text{vertchange}, \text{haulbackwt}) \]

RETURN

FullPayload: 'Calculates mainline tension with full suspension using catenary assumptions

\[ H_3 = H_2 + H_4 - H_1 \]

IF \( H_3 < 0 \) THEN 'Eliminate mainline if static equilibrium is not achievable with tension in mainline. Cannot be achieved with tension in skyline when the operator applied haulback tension is specified by user.

\[ V_{carriage} = 0 \]

\[ \text{Mainline} = 0 \]

GOSUB FullPayloadHB

ELSE

CALL CatenaryBdh(H3, \text{dtower}, \text{htower}, \text{mainlinewt}, \text{Mainline})

\[ T_{carriage} = FN_{tension}(\text{Mainline}, 1, \text{htower}, \text{mainlinewt}) \]

\[ V_{3carriage} = SGN(-\text{htower}) \times \sqrt{T_{3carriage}^2 - H_3^2} \]

\[ \text{payload} = V_{3carriage} + V_{2carriage} + V_{1carriage} + \]

\[ \text{GOSUB PayloadRecalc} \]

IF \( \text{ABS}(h_g - h) > .00001 \) THEN GOSUB MainlineSearch

END IF

RETURN
V4carriage - carriagewt

v = 1
h = 0
H3skylinelimited = H3 'Save H3 for initial estimate of
skyline tension in mainline 'limited case. (See
'SkylineAdjustment subroutine)

END IF

RETURN

PartialPayload: 'Calculates mainline tension withpartial
'suspension using rigid link assumptions

H3 = (TAN(alpha) * (H1 - H2 - H4) - (Vicarriage + V2carriage + V4carriage - carriagewt) + mainlinewt * ltower / 2) /
(-htower / dtower - TAN(alpha))

IF H3 < 0 THEN 'Eliminate mainline if static
H3 = 0 'equilibrium cannot be achieved
V3carriage = 0 'with tension in the mainline
Mainline = 0 GOSUB PartialPayloadHB

ELSE
V3carriage = TAN(alpha) * (H1 + H3 - H2 - H4) - Vicarriage - V2carriage - V4carriage - carriagewt
T3carriage = SQRT(htower) * SQRT(V3carriage * 2 + H3 * H3)
V3tower = V3carriage - SQRT(htower) * ltower * mainlinewt
T3tower = SQRT(H3 * H3 + V3tower * 2)

IF T3tower > T3carriage THEN Mainline = T3tower ELSE
Mainline = T3carriage

Wh = H1 + H3 - H2 - H4
Wv = Vicarriage + V2carriage + V3carriage + V4carriage - carriagewt
normal = Wh / (SIN(theta) + u * COS(theta))
part1 = (1 - logdiameter / loglength * TAN(theta + beta)) * (COS(theta) - u * SIN(theta))
part2 = (TAN(theta + beta) + logdiameter / loglength) * (SIN(theta) + u * COS(theta))
payload = normal * loglength / epsilon * (part1 + part2)
v = Wv / payload
h = Wh / payload

H3skylinelimited = H3 'Save H3 for initial estimate of
'skyline tension in mainline 'limited case. (See
'SkylineAdjustment subroutine)

END IF

RETURN

HaulbackAdjustment: 'Adjusts haulback tension for four
'segment systems when user specified
'tension applied to haulback would cause mainline tension to exceed allowable

Mainline = Maxmainline
CALL CatenaryTdhd(Mainline, dtower, htower, mainlinewt, H3)
'Calculate mainline tension components maximum allowable tension
T3carriage = FNTension(Mainline, 1, htower, mainlinewt)
V3carriage = SGN(-htower) * SQR(T3carriage ^ 2 - H3 ^ 2)
IF suspensiontype(tp) = 1 THEN
    GOSUB FullPayloadHB
    IF H4 = 0 THEN
        GOSUB SkylineAdjustment
        'Branch to subroutine to reduce skyline tension if haulback must be <= 0 for static equilibrium
        'If H4 is not = 0 then payload calculation is correct
    ELSE
        GOSUB PartialPayloadHB
        IF H4 = 0 THEN
            GOSUB SkylineAdjustment
            'Branch to subroutine to reduce skyline tension if haulback must be <= 0 for static equilibrium
        ELSE
            CALL CatenaryHdh(H4, dtail, htail, haulbackwt, T4)
            'Calculate haulback tensions using rigid link estimate of H4
            GOSUB PayloadRecalcHB
            IF ABS(hg - h) > .00001 THEN GOSUB HaulbackSearch
            'Recalculates mainline tensions if rigid link estimate of H3 was not within tolerance.
            'Tolerance is the value of "h" calculated using the calculated value of H3 minus the value of "h" calculated using the estimated value of H3..
        END IF
        GOSUB SkylineAdjustment
    ELSE
        CALL CatenaryTdhd(Mainline, dtower, htower, mainlinewt, H3)
        T3carriage = FNTension(Mainline, 1, htower, mainlinewt)
        V3carriage = SGN(-htower) * SQR(T3carriage ^ 2 - H3 ^ 2)
        END IF
        RETURN
HaulbackAdjustmentSS:
Mainline1 = Mainline
    'Save Mainline in case it must be reset below (IF H4 = 0)
Mainline = Maxmainline
CALL CatenaryTdhd(Mainline, dtower, htower, mainlinewt, H3)
T3carriage = FNTension(Mainline, 1, htower, mainlinewt)
V3carriage = SGN(-htower) * SQR(T3carriage ^ 2 - H3 ^ 2)
IF suspensiontype(tp) = 1 THEN
    GOSUB FullPayloadHB
    IF H4 = 0 THEN
Mainline = Mainline1
GOSUB SkylineAdjustmentSS
'Reset Mainline and skyline tension if H4 = 0
END IF
'If H4 is not = 0 then payload calculation is correct
ELSE
GOSUB PartialPayloadHB
IF H4 = 0 THEN
Mainline = Mainline1
GOSUB SkylineAdjustmentSS
'Reset Mainline and skyline tension if H4 = 0
ELSE
CALL CatenaryHdh(H4, dtail, htail, haulbackwt, T4)
'Calculate haulback tensions using rigid link estimate of H4
GOSUB PayloadRecalcHB
IF ABS(hg - h) > .00001 THEN GOSUB HaulbackSearch
'Recalculates mainline tensions if rigid link estimate of H3 was not within tolerance.
'Tolerance is the remainder of the value of "h" calculated using the estimate of H3 minus the 'value of "h" calculated using calculated value of H3
END IF
END IF
RETURN

FullPayloadHB:
'Calculates haulback tensions and payload when mainline and skyline are both at max. allowable tension and tension is applied to the haulback
H4 = H1 + H3 - H2
IF H4 = 0 THEN
'H4 = 0
Eliminate haulback if static equilibrium
H4 = 0
V4carriage = 0 'cannot be attained with tension in the haulback
UseHaulback = 0 'Toggle to turn off haulback options
ELSE
CALL CatenaryHdh(H4, dtail, htail, haulbackwt, T4)
'Uses catenary calc. of H4 so adjustment is not needed
T4carriage = FNtension(T4, 1, -htail, haulbackwt)
V4carriage = SGN(htail) * SQR(T4carriage ^ 2 - H4 ^ 2)
T4tail = FNtension(T4carriage, -1, -htail, haulbackwt)
Haulback = FNtension(T4tail, 1, -vertchange, haulbackwt)
payload = FRw
END IF
PartialPayloadHB: 'Calculates mainline and haulback tensions when mainline and skyline are both at max. allowable tension and a user specified haulback tension is applied. Partial suspension; rigid link assumptions for haulback. Used as first estimate of haulback tensions for catenary haulback calculations

\[
H4 = \left( \tan(\alpha) \cdot (H1 + H3 - H2) - (V1\text{carriage} + V2\text{carriage} + V3\text{carriage} - \text{carriagewt}) \cdot t\text{tail} \cdot \text{haulbackwt} / 2 \right) / \left( h\text{tail} / \tan(\alpha) \right)
\]

IF \( H4 < 0 \) THEN

Eliminate haulback if static equilibrium attained with tension in the haulback line
UseHaulback = 0 'Toggle to turn off haulback options
ELSE

\[
V4\text{carriage} = \tan(\alpha) \cdot (H1 + H3 - H2 - H4) - (V1\text{carriage} + V2\text{carriage} + V3\text{carriage} - \text{carriagewt})
\]

\[
T4\text{carriage} = \sqrt{H4^2 + V4\text{carriage}^2}
\]

\[
T4\text{tail} = \text{FNtension}(T4\text{carriage}, -1, -h\text{tail}, \text{haulbackwt})
\]

\[
\text{Haulback} = \text{FNtension}(T4\text{tail}, 1, -\text{vert-change}, \text{haulbackwt})
\]

\[
\text{payload} = \text{FNw}
\]

END IF

\( H4\text{rigidlink} = H4 \)

RETURN

TorqueLimited: 'Calculates maximum tension based on available torque for a running skyline.

\[
\text{HBrequired} = l\text{tower} + 2 \cdot t\text{tail} + \text{towerht} - \text{druhtm}
\]

\[
\text{HBlineOnDrum} = \text{HBlength} - \text{HBrequired} - \text{towerht}
\]

\[
\text{layers} = (-\text{HBdrumradius} + \sqrt{\text{HBdrumradius}^2 + \text{HBdiameter}^2} \cdot \text{HBlineOnDrum} / .268 / \text{HBdrumwidth}) / \text{HBdiameter}
\]

'layers' is unrounded number of layers of rope on haulback drum

\[
'\text{LayersOnDrum}' \text{is rounded number of layers}
\]

\[
\text{EffectiveRadius} = (\text{HBdrumradius} + \text{LayersOnDrum} \cdot \text{HBdiameter} - \text{HBdiameter} / 2) / 12
\]

\[
\text{Titorque} = \text{MaxHBtorque} / \text{EffectiveRadius} + \text{towerht} - \text{druhtm}
\]

RETURN

Shortline: 'Calculates unstretched line length for a single span standing skyline

IF \( \text{suspensiontype(tp)} = 1 \) THEN \( \text{loght} = \text{minfly} \) ELSE \( \text{loght} = \)
Tltower = FNtension(Maxskyline, 1, -vertchange, skylinewt)
liftpoint(0) = towertp
Keeplifts = TotalLifts 'Save "TotalLifts" to reset later for 'live skyline analysis with more than 'one lift specified.
keeptp = tp
REM **** Establish lift points for terrain point being ****
REM **** analyzed
IF LEFT$(syst$, 4) = "Live" AND TotalLifts = 1 THEN
liptpoint(TotalLifts) = tp
ELSEIF LEFT$(syst$, 4) = "Live" AND TotalLifts > 1 THEN
FOR lift = 2 TO Keeplifts
IF liftpoint(lift) * dir > tp * dir THEN TotalLifts
= TotalLifts - 1
NEXT
liftpoint(TotalLifts) = tp
ELSE
liftpoint(TotalLifts) = outeryardinglimit
END IF
REM **** Calculate unstretched line lengths ****
FOR lift = TotalLifts TO 1 STEP -1
FOR tp = liftpoint(lift - 1) + dir TO liftpoint(lift)
STEP dir
carriagex = x(tp)
ON suspensiontype(tp) GOSUB FullSuspension,
PartialSuspension
GOSUB LineLengthCalc
linelength(tp) = linelength
IF linelength(tp) < minline THEN
minline = linelength(tp)
criticaltp = tp
END IF
NEXT tp
FOR tp = liftpoint(lift - 1) + dir TO liftpoint(lift)
STEP dir
IF criticaltp = tp THEN critical(tp) = -1 ELSE
critical(tp) = 0
minline(tp) = minline
NEXT tp
NEXT lift
TotalLifts = Keeplifts 'Reset "TotalLifts" for subsequent 'terrain points
REM **** Select whether each terrain point is a lift ****
REM **** point
IF TotalLifts > 1 THEN
  tp = inneryardinglimit - dir
  FOR lift = 1 TO TotalLifts
    WHILE skip = 0
      tp = tp + dir
      IF liftpt(tp) = -1 THEN
        liftpoint(lift) = tp
        skip = -1
      END IF
    WEND
    skip = 0
  NEXT
  tp = keeptp 'Reset tp to current terrain point number'
RETURN

StandingSkylineTensions: 'Calculate tensions for standing skyline analysis, multispans analysis, and load path analysis

IF LEFT$(syst$, 5) = "Multi" THEN 'Calculate skyline tensions at headspar and tailspar for multispansystem
  GOSUB SpanTensions
  Tmax = FNtension(Tmax, 1, -(htower + htail), linewt)
  T2tail = FNtension(Tmax, 1, htower + htail, linewt)
ELSE 'Calculate skyline tensions at headspar and tailspar for single span system
  Tmax = FNtension(Tmax, 1, -vertchange, linewt)
  T2tail = FNtension(Tmax, 1, vertchange, linewt)
END IF

GOSUB GeometrySearch 'Establish log and skyline geometry
T1carriage = FNtension(T1, 1, htower, linewt)
V1carriage = SGN(-htower) * SQR(T1carriage^2 - H1^2)
V2carriage = SGN(htail) * SQR(T1carriage^2 - H2^2)
IF UseMainline THEN
  IF UseHaulback THEN GOSUB HaulbackTensions 'Calculate haulback tensions for four segment live or standing skyline
  GOSUB MainlineTensjons
ELSE ON suspensiontype(tp) GOSUB FullPayloadHB, PartialPayloadHB
END IF
IF LEFT$(syst$, 5) = "Multi" THEN Tmax = TmaxMS
RETURN

SkylineAdjustments: 'Adjusts skyline tension for standing
Tmax1 = Tmax
zsAS1 = FNz
recalc = -1
Tmax2 = 0
WHILE recalc
  'Recalculates maximum skyline tension based on 'criteria in subroutine "GeometrySearch"
  IF recalc THEN Tmax2 = Tmax2 + (Tivax1 - Tmax2) * .5
  recalc = 0
  Tmax = Tmax2
  GOSUB StandingSkylineTensions
  zSAS2 = FNz
WEND
iterations = 0
WHILE ABS(zSAS2) > .5 AND NOT stuck 'Secant search routine
  in = (zSAS2 - zSAS1) / (Tivax2 - Tivax1)
  Tmax1 = Tmax2
  zSAS1 = zSAS2
  Tmax2 = Tivax2 - zSAS2 / in
  Tmax = Tmax2
  GOSUB StandingSkylineTensions
  zSAS2 = FNz
  iterations = iterations + 1
  IF iterations > 10 THEN stuck = -1
WEND
Skyline = Tmax
RETURN

SkylineAdjustmentHB: 'Adjusts skyline tension for a four 'segment system when the mainline is 'not needed and the allowable 'haulback tension is limiting. 'This occasionally occurs in steep 'downhill yarding with full 'suspension

UseMainline = 0 'Toggle off mainline calculations
Tmax1 = Tmax
zsAH1 = Haulback - Thaulback
Tmax = Tmax * .5
Tmax2 = Tmax
GOSUB StandingSkylineTensions
zSAH2 = Haulback - Thaulback
WHILE ABS(zSAH2) > .5
m = (zSAH2 - zSAH1) / (Tmax2 - Tmax1)
Tmax1 = Tmax2
zSAH1 = zSAH2
Tmax2 = Tmax2 - zSAH2 / m
Tmax = Tmax
GOSUB StandingSkylineTensions
zSAH2 = Haulback - Thaulback
WEND
Skyline = Tmax
UseMainline = -1
'Calculations for next tp
RETURN

GeometrySearch: 'Calculates payload and skyline geometry
IF suspensiontype(tp) = 1 THEN
'Assume payload is flying
'or dragging based upon
'user supplied
'specifications
GOSUB Flying
ELSE
suspensiontype(tp) = 2
zGS1 = 0
IF critical(tp) THEN
'Enter with binary search
loght = mindrag
interval = 2
'If tp is critical based on
'line length, use a small
'interval in search
ELSE
'If tp is not critical
loght = .999 * loglength 'start at nearly vertical
'and divide into 5 equal
'intervals
interval = -(loght - mindrag) / 5
END IF
GOSUB PartialSuspension
GOSUB LineLengthCalc
loght2 = loght
zGS2 = linelength - minline(tp)
IF linelength < minline(tp) THEN
'Only happens
'when original
'loght was nearly
'vertical
GOSUB Dragging
ELSEIF linelength > minline(tp) AND interval = 2 THEN
'Only happens
'at critical tp
GOSUB Dragging
ELSEIF linelength > minline(tp) + .01 THEN

GOSUB Flying

'This only happens when original loght was nearly vertical 'and linelength was 'too long

ELSE
END IF
END IF

IF suspension type(tp) = 2 THEN partial clearance(tp) = loght
RETURN

Dragging:

'Geometry search subroutine for partial 'suspension

WHILE zGS1 * zGS2 >= 0

'Enter with binary search using 'interval selected in subroutine

loght1 = loght2
zGS1 = zGS2
loght2 = loght2 + interval
loght = loght2
GOSUB PartialSuspension
GOSUB LineLengthCalc
zGS2 = linelength - minline(tp)
WEND

WHILE ABS(zGS2) > .05

'Switch to secant search when 'loght is bracketed

m = (zGS2 - zGS1) / (loght2 - loght1)
loght1 = loght2
zGS1 = zGS2
loght2 = loght2 - zGS2 / m
loght = loght2
GOSUB PartialSuspension
GOSUB LineLengthCalc
zGS2 = linelength - minline(tp)
WEND

RETURN

Flying:

'Calculates geometry for full suspension

zGS1 = 10 ^ 10
suspension type(tp) = 1
loght = 0
GOSUB FullSuspension
GOSUB LineLengthCalc
loght2 = loght
zGS2 = linelength - minline(tp)
WHILE zGS1 * zGS2 >= 0 AND ABS(zGS2) < ABS(zGS1)

'Enter with 'binary 'search

IF zGS1 > 0 THEN interval = 10 ELSE interval = -2
loght1 = loght2 'Use large positive interval
zGS1 = zGS2 'if linelength is too long
loght2 = loght + interval 'small negative interval if
loght = loght2 'linelength is too short
GOSUB FullSuspension
GOSUB LineLengthCalc
zGS2 = linelength - minline(tp)
WEND
WHILE ABS(zGS2) > .05 AND NOT recalc 'Switch to secant
   m = (zGS2 - zGS1) / (loght2 - loghti) 'search routine
   loghti = loght2 'when loght is bracketed.
   zGS1 = zGS2 'If skyline tension is too
   loght2 = loght2 - zGS2 / m 'small, the line length
   loght = loght2 'based on tension will be
GOSUB FullSuspension 'shorter than the straight
GOSUB LineLengthCalc 'line distance between
zGS2 = linelength - minline(tp) 'support and carriage and
   'the "recalc" conditions
   'will exist.
IF ABS(ABS(zGS2) - ABS(zGS1)) < .05 AND ABS(zGS2) > .05 THEN recalc = -1 'Recalculate skyline
                   'tension in
                   '"SkylineAdjustmentSS if
                   '"recalc" conditions exist
WEND
REM In some cases, a full suspension analysis at a given log
REM height will indicate that the load must be partially
REM suspended and a partial suspension analysis at the same
REM height will indicate that the load must be fully
REM suspended. This most commonly occurs in a load path
REM analysis. The payload tolerance never converges and the
REM program goes into an endless loop. This algorithm forces
REM that case to show a full suspension payload and the
REM following line forces the log clearance in that situation
REM to be displayed as = 0. The carriage ht. calculated in
REM this subroutine would give a log ht. < 0. However, the
REM program would not have branched to this subroutine if it
REM had been possible to get convergence on a partial
REM suspension analysis
IF loght < 0 THEN fullclearance(tp) = 0 ELSE fullclearance(tp) = loght
RETURN

LineLengthCalc: 'Subroutine to calculate unstretched line
   'length for standing skyline or multispan
   'analysis
IF LEFT$(syst$, 5) = "Multi" THEN
  "Multi" indicates multispan analysis
  GOSUB GeometryMS 'Calc. skyline geometry for loaded segment
ELSE
  GOSUB SkylineGeometry
END IF

Ti = FNtension(Titower, -1, -htower, linewt) 'Upper end tension of segment 1 of loaded span

CALL CatenaryTdh(T1, dtower, htower, linewt, H1) 'Horiz. tension component for segment 1 stower = s / FNstretch(T1, linewt, ABS(htower)) 'Unstretched line length between headspar and carriage

T2 = FNtension(T2tail, -1, htail, linewt) 'Upper end tension of segment 2 of loaded span

CALL CatenaryTdh(T2, dtail, htail, linewt, H2) 'Horiz. tension component for segment 1 stall = s / FNstretch(T2, linewt, ABS(htail)) 'Unstretched line length between tailspar and carriage

IF LEFT$(syst$, 5) = "Multi" THEN
  linelength = stower + stall + emptyspans(ThisSpan) 'Add loaded span line length to line lengths for unloaded spans
ELSE
  linelength = stower + stall 'Unstretched line length from headspar to tailspar
ENDIF

RETURN

MainlineSearch: 'Adjusts mainline tension if rigid link estimate of mainline tension did not meet specified tolerance

H3Old = H3
zMS1 = 10 ^ 6 * (hg - h)
H3New = H3 - .1 * SGN(zMS1) * H3 'Make small adjustment to H3 since rigid link error is normally small

H3 = H3New
CALL CatenaryHdh(H3, dtower, htower, mainlinewt, Mainline) GOSUB PayloadRecalc

zMS2 = 10 ^ 6 * (hg - h) 'Secant search routine
WHILE ABS(zMS2) > 1
  m = (zMS2 - zMS1) / (H3New - H3Old)
  H3Old = H3New
  zMS1 = zMS2
  zMS2 = zMS2 + m
  H3New = H3New - zMS2 / m
END WHILE

RETURN
CALL CatenaryHdh(H3, dtower, htower, mainlinewt, Mainline)
GOSUB PayloadRecalc
ZMS2 = 10 ^ 6 * (hg - h)
WEND
RETURN

SkylineAdjustinent: 'Adjust skyline tension when mainline tension is limiting. Estimate horizontal component of tension since it allows a direct calculation of other tension components. An estimate of T1 would require an iterative calculation of "m" to get tension components

H1 = H3 / H3skylinelimited * H1 'Initial estimate of horizontal tension component of T1 is proportional to change in H3 for mainline limited case compared to skyline limited case
GOSUB SkylineRecalc

IF ABS(hg - h) > .00001 THEN 'Adjust H1 if initial estimate did not meet specified tolerance
H1old = H1
ZSA1 = (hg - h) * 10 ^ 6
Hinew = H1 + SGN(ZSA1) * .1 * H1
H1 = Hinew
GOSUB SkylineRecalc
ZSAl = (hg - h) * 10 ^ 6
WHILE ABS(ZSA2) > 10 'Secant search routine
m = (ZSA2 - ZSA1) / (Hinew - H1old)
ZSA1 = ZSA2
H1old = Hinew
Hinew = Hinew - ZSA2 / m
H1 = Hinew
GOSUB SkylineRecalc
ZSA2 = (hg - h) * 10 ^ 6
WEND
ELSE
END IF
Mainline = Maxmainline
IF vertchange > 0 THEN Skyline = T1carriage + ABS(htail) * linewt ELSE Skyline = T1carriage + ABS(htower) * linewt
RETURN

SkylineRecalc: 'Recalculate skyline tensions for mainline limited case.
CALL CatenaryHdh(H1, dtower, htower, linewt, T1)
IF SGN(htower) < 0 THEN T1carriage = T1 + htower * linewt ELSE
T1carriage = T1
V1carriage = SGN(-htower) * SQR(T1carriage^2 - H1^2)
IF SGN(htail) < 0 THEN T2 = T1carriage ELSE T2 = T1carriage + htail * linewt
CALL CatenaryTdh(T2, dtail, htail, linewt, H2)
V2carriage = SGN(htail) * SQR(T1carriage^2 - H2^2)
IF systemtype = 3 THEN
H4 = H2
'Running skyline
V4carriage = V2carriage
END IF
payload = (V1carriage + V2carriage + V3carriage + V4carriage - carriagewt) / v
hg = (H1 + H3 - H2 - H4) / payload
'Decimal percent of the payload in the horizontal 'component of the tagline tension
IF towery > tally THEN Skyline = T1 ELSE Skyline = T2
RETURN

PayloadRecalc:
'Recalculate mainline tension components 'and payload for skyline limited case
T3carriage = FNtension(Mainline, i, htower, mainlinewt)
V3carriage = SQR(T3carriage^2 - H3 * H3) * SGN(-htower)
payload = FNw
hg = (H1 + H3 - H2 - H4) / payload
'Decimal percent of the payload in the horizontal 'component of the tagline tension
RETURN

HaulbackSearch:
'Adjusts haulback tension for live 'or 'standing skyline when both skyline 'tension and mainline tension are known.
H4Old = H4
zHS1 = 10 * 6 * (hg - h)
H4New = H4 - 0.1 * SGN(zHS1) * H4
CALL CatenaryHdh(H4New, dtail, htail, haulbackwt, T4)
H4 = H4New
GOSUB PayloadRecalcHB
zHS2 = 10 * 6 * (hg - h)
WHILE ABS(zHS2) > 10
'Secant search routine
m = (zHS2 - zHS1) / (H4New - H4Old)
H4Old = H4New
zHS1 = zHS2
H4New = H4New - zHS2 / m
H4 = H4New
CALL CatenaryHdh(H4New, dtail, htail, haulbackwt, T4)
GOSUB PayloadRecalcHB
zHS2 = 10 ^ 6 * (hg - h)

RETURN

PayloadRecalcHB:  'Recalculates haulback tension components and payload when skyline tension and mainline tension are known

T4carriage = FNtension(T4, 1, -htail, haulbackwt)
V4carriage = SGN(htail) * SQR(T4carriage ^ 2 - H4 ^ 2)

payload = FNw

hg = (H1 + H3 - H2 - H4) / payload

'Decimal percent of the payload in the horizontal component of the tagline tension

RETURN

SetLiftpoints:  'Determines whether each terrain point is (-1) or is not (0) a lift point based on user input

tp = inneryardinglimit - dir

liftpoint(TotalLifts) = outeryardinglimit

FOR lift = 1 TO TotalLifts
    WHILE skip = 0
        tp = tp + dir
        IF liftpoint(lift) = tp THEN
            liftpt(tp) = -1  'Terrain point is a lift point
            skip = -1
        ELSE
            liftpt(tp) = 0  'Terrain point is not a lift point
        END IF
    WEND
    skip = 0
NEXT

RETURN

SpanTensions:  'Calculates tension at top of each support for each span for multispan analysis.

"Tsupport2(n)" is the tension at the support farthest from the headspar for the nth span.

Tsupport2(0) = FNtension(Tmax, 1, towery - maxelev, linewt)

'Tension at top of headspar

T(1) = FNtension(Tsupport2(0), -1, -hSpan(1), linewt)

'maximum skyline tension for first span

Tsupport2(1) = FNtension(T(1), 1, hSpan(1), linewt)

'tension at first intermediate support
FOR span = 2 TO lastsupport - 1
  T(span) = FNtension(T(support2(span) - 1), -1, -hSpan(span), linewt)
  'Maximum skyline tension at either support for each span
  Tsupport2(span) = FNtension(T(span), 1, hSpan(span), linewt)
NEXT

TmaxMS = Tmax
RETURN

SpanLengths: 'Calculates length of skyline for unloaded 'spans (spans other than the span in which the 'carriage is located) using catenary 'assumptions

maxelev = 0
FOR n = 1 TO lastsupport
  'Find maximum elevation in
  elev(n) = x(support(n)) + supportht(n) 'the multispans
  IF elev(n) > maxelev THEN maxelev = elev(n) 'system
NEXT

Tsupport2(0) = FNtension(Tmax, 1, towery - maxelev, skylinewt)
  'Tension at top of headspar
FOR span = 1 TO lastsupport - 1
  xsupport1(span) = x(support(span)) 'x and y
  xsupport2(span) = x(support(span + 1)) 'coordinates
  ysupport1(span) = y(support(span)) + supportht(span) 'span
  ysupport2(span) = y(support(span + 1)) + supportht(span + 1)
  dSpan = ABS(xsupport2(span) - xsupport1(span))
    'Horizontal distance between supports
  hSpan(span) = ysupport2(span) - ysupport1(span)
    'Vertical distance between supports
  T(span) = FNtension(T(support2(span) - 1), -1, -hSpan(span), linewt)
    'Maximum skyline tension at any point in the span
  Tsupport2(span) = FNtension(T(span), 1, hSpan(span), linewt)
    'Skyline tension at top of support farthest from 'headspar for the span
  CALL CatenaryTdh(T(span), dSpan, hSpan(span), skylinewt, Horiz(span))
  spanlength(span) = s / FNstretch(T(span), skylinewt, hSpan(span))
    'Unstretched line length of span
  dSpan(span) = dSpan
NEXT span
RETURN
AssignSpans: 'Calculate unstretched line length of all unloaded spans and assigns that length to the span in which the carriage is located.

FOR span = 1 TO lastsupport - 1
    emptyspans(span) = 0
    FOR ThisSpan = 1 TO lastsupport - 1
        IF ThisSpan <> span THEN emptyspans(span) = emptyspans(span) + spanlength(ThisSpan)
    NEXT ThisSpan
NEXT span
RETURN

ShortlineMS: 'Calculates the unstretched line length required to attain the user specified minimum payload clearance for each terrain point and assigns that line length to the entire analysis

minline = 10 ^ 10 'Start with large minimum unstretched line length
FOR span = 1 TO lastsupport - 1
    IF support(span + 1) - support(span) <> 1 THEN 'Check to see if supports are at consecutive terrain points
        FOR tp = support(span) + dir TO support(span + 1) - dir STEP dir
            IF NOT supportpoint(tp) THEN 'Assign intermediate supports
                support1(tp) = support(span) 'to each terrain point
                support2(tp) = support(span + 1)
                carriagex = x(tp)
            END IF
        END FOR
    END IF
NEXT span

GO SUB FullSuspension, PartialSuspension

Thisspan = span
GO SUB GeometryMS
Tsupport = FNtension(T(span), -1, -hSpan(span), skylinewt) 'Tension at support closest to headspar
T1 = FNtension(Tsupport, -1, -htower, skylinewt) 'Maximum tension in segment closest to headspar
CALL CatenaryTdh(T1, dtower, htower, skylinewt, H1)
stower = s / FNstretch(T1, skylinewt, ABS(htower))

'Unstretched line length of segment closest to headspar
Tnextsupport = FNtension(T(span), 1, hSpan(span), skylinewt)

T2 = FNtension(Tnextsupport, -1, htail, skylinewt)

CALL CatenaryTdh(T2, dtail, htail, skylinewt, H2)

stail = s / FNstretch(T2, skylinewt, ABS(htail))

linelength = stower + stail + emptyspans(span)

FOR tp = inneryardinglimit TO outeryardinglimit STEP dir

IF criticaltp = tp THEN critical(tp) = -1 ELSE critical(tp) = 0

END IF

END FOR

IF linelength < minline THEN 'Define terrain point associated with the shortest line length for each span as a critical terrain point

minline = linelength

criticaltp = tp

END IF

END IF

FOR tp = inneryardinglimit TO outeryardinglimit STEP dir

IF NOT supportpoint(tp) THEN minline(tp) = minline

NEXT tp

FOR tp = inneryardinglimit TO outeryardinglimit STEP dir

stall = s / FNstretch(T2, skylinewt, ABS(htail))

linelength = spanlength(span) + emptyspans(span)

IF linelength < minline THEN 'Unstretched line length of two consecutive terrain points

minline = linelength

END IF

END FOR

NEXT span

Tnextsupport = FNtension(T(span), 1, hSpan(span), skylinewt)

T2 = FNtension(Tnextsupport, -1, htail, skylinewt)

CALL CatenaryTdh(T2, dtail, htail, skylinewt, H2)

stail = s / FNstretch(T2, skylinewt, ABS(htail))

linelength = stower + stail + emptyspans(span)

IF linelength < minline THEN 'Define terrain point associated with the shortest line length for each span as a critical terrain point

minline = linelength

criticaltp = tp

END IF

END IF

END IF

FOR tp = inneryardinglimit TO outeryardinglimit STEP dir

IF NOT supportpoint(tp) THEN minline(tp) = minline

NEXT tp

NEXT tp
GeometryMS: 'Calculate geometry of span in which carriage 'is located for a multispan system

dtower = ABS(carriagex - x(support1(tp)))
htower = carriagey - (y(support1(tp)) + supportht(ThisSpan))
dtail = ABS(x(support2(tp)) - carriagex)
htail = (y(support2(tp)) + supportht(ThisSpan + 1)) - carriagey

ltower = SQR(dtower^2 + htower^2) 'Rigid link line
ltail = SQR(dtail^2 + htail^2) 'lengths (stretched 'length)

RETURN

T1Limited: 'Segment 1 tensions for clamping carriage 'analysis with slack mainline when Segment 1 'tension is limiting

T1 = Maxskyline
CALL CatenaryTdh(T1, dtower, htower, linewt, H1)
stower = s / FNstretch(T1, linewt, ABS(htower)) 'Unstretched line length of seg. 1
Ticarriage = FNtension(T1, 1, htower, linewt)
Vicarriage = SGN(-htower) * SQR(Ticarriage^2 - H1^2)

ON suspensiontype(tp) GOSUB FullPayloadCCTi, PartialPayloadCCTi

RETURN

FullPayloadCCTi: 'Segment 2 tensions and payload when 'Segment 1 tension is limiting and payload 'is fully suspended

H2 = H1
CALL CatenaryHdh(H2, dtail, htail, linewt, T2)
stail = s / FNstretch(T2, linewt, ABS(htail)) 'Unstretched line length of seg. 2
T2carriage = FNtension(T2, -1, -htail, linewt)
V2carriage = SGN(htail) * SQR(T2carriage^2 - H2^2)
payload = Vicarriage + V2carriage - carriagewt
v = 1
h = 0

RETURN

PartialPayloadCCTi: 'Segment 2 tensions and payload when 'Segment 1 tension is limiting and payload 'is partially suspended. Make initial 'estimate of H2 and payload using Rigid 'Link assumptions and adjust using
'caterany assumptions

\[ H_2 = \left( \tan(\alpha) \times H_1 - (V_{\text{carriage}} - \text{carriage wt}) - l_{\text{tail}} \times \text{linewt} / 2 \right) / \left( h_{\text{tail}} / dt + \tan(\alpha) \right) \]

\[ V_{\text{carriage}} = \tan(\alpha) \times \left( H_2 - H_1 \right) - (V_{\text{carriage}} - \text{carriage wt}) \]

\[ T_{\text{carriage}} = \sqrt{H_2^2 + V_{\text{carriage}}^2} \]

\[ W_h = V_{\text{carriage}} + V_{\text{carriage}} - \text{carriage wt} \]

\[ \text{normal} = W_h / (\sin(\theta) + u \times \cos(\theta)) \]

\[ \text{part1} = (1 - \log\text{diameter} / \log\text{length} \times \tan(\theta + \beta)) \times \cos(\theta) - u \times \sin(\theta) \]

\[ \text{part2} = \tan(\theta + \beta) + \log\text{diameter} / \log\text{length} \times \sin(\theta) + u \times \cos(\theta) \]

\[ \text{payload} = \text{normal} \times \log\text{length} / \epsilon \times \text{part1 + part2} \]

'rigid link estimate of payload

\[ v = W_v / \text{payload} \]

\[ h = W_h / \text{payload} \]

CALL CatenaryHdh(H2, dtail, htail, linewt, T2)

'caterany estimate of payload using rigid link estimate of H2

GOSUB PayloadRecalcCC

IF ABS(hg - h) > .00001 THEN 'Secant search routin to adjust

\[ H_2\text{old} = H_2 \]

'payload if rigid link estimate \[ H_2\text{new} = H_2 + .1 \times \text{sgn}(z_{\text{CCT}}) \times H_2 \]

CALL CatenaryHdh(H2\text{new}, dtail, htail, linewt, T2)

\[ H_2 = H_2\text{new} \]

GOSUB PayloadRecalcCC

\[ z_{\text{CCT}} = 10 \times 6 \times (h_g - h) \]

WHILE ABS(z_{\text{CCT}}) > 10

\[ m = (z_{\text{CCT}} - z_{\text{CCT}}) / (H_2\text{new} - H_2\text{old}) \]

\[ H_2\text{old} = H_2\text{new} \]

\[ z_{\text{CCT}} = z_{\text{CCT}} \]

\[ H_2\text{new} = H_2\text{new} - z_{\text{CCT}} / m \]

CALL CatenaryHdh(H2, dtail, htail, linewt, T2)

GOSUB PayloadRecalcCC

\[ z_{\text{CCT}} = 10 \times 6 \times (h_g - h) \]

WEND

\[ s_{\text{tail}} = s / \text{FNstretch}(T_2, \text{linewt}, \text{ABS}(htail)) \]

'Unstretched line length of segment 2

END IF

RETURN
T2Limited: 'Segment 2 tensions for clamping carriage
'analysis with slack mainline when Segment 2
'tension is limiting

T2 = Maxskyline
CALL CatenaryTdh(T2, dtail, htail, linewt, H2)
stail = s / FNstretch(T2, linewt, ABS(htail))
'Unstretched line length for Segment 2
T2carriage = FNtension(T2, -i, -htail, linewt)
V2carriage = SGN(htail) * SQR(T2carriage^2 - H2^2)
ON suspensiontype(tp) GOSUB FullPayloadCCT2,
PartialPayloadCCT2
RETURN

FullPayloadCCT2: 'Segment 1 tensions and payload for
'clamping carriage analysis with slack
'mainline when Segment 2 tension is
'limiting and payload is fully suspended

H1 = H2
CALL CatenaryHdh(H1, dtower, htower, linewt, Ti)
stower = s / FNstretch(Ti, linewt, ABS(htower))
'Unstretched line length for Segment 2
T1carriage = FNtension(Ti, i, htower, linewt)
V1carriage = SGN(-htower) * SQR(T1carriage^2 - H1^2)
payload = V1carriage + V2carriage - carriagewt
v = 1
h = 0
RETURN

PartialPayloadCCT2: 'Segment 1 tensions and payload for
'clamping carriage analysis with slack
'mainline when Segment 2 tension is
'limiting and payload is partially
'suspended. Initial estimate is made
'using Rigid Link assumptions and
'adjusted using catenary assumptions.

H1 = (TAN(alpha) * (-H2) - (V2carriage - carriagewt) + ltower
* linewt / 2) / (-htower / dtower * TAN(alpha))
V1carriage = TAN(alpha) * (H1 - H2) - (V2carriage -
carriagewt)
T1carriage = SQR(H1^2 + V1carriage^2)
V1tower = V1carriage - SGN(htail) * linewt * ltower
Titower = SQR(V1tower^2 + H1^2)
W0 = H1 - H2
Wv = V1carriage + V2carriage - carriagewt
normal = Wh / (SIN(theta) + u * COS(theta))
part1 = (1 - logdiameter / loglength * TAN(theta + beta)) *
(COS(theta) - u * SIN(theta))
part2 = (\tan(\theta + \beta) + \log\text{diameter} / \log\text{length}) * 
\(\sin(\theta) + u \cos(\theta)\)

\[\text{payload} = \text{normal} \cdot \log\text{length} / \epsilon \cdot (\text{part1 + part2})\]

'Rigid link estimate

\[v = \frac{Wv}{\text{payload}}\]

\[h = \frac{Wh}{\text{payload}}\]

CALL CatenaryHdh(H1, dtower, htower, linewt, T1)

'Catenary payload calculation using rigid link estimate of H1

GOSUB PayloadRecalcCC1

\[\text{payload} = \text{normal} \cdot \log\text{length} / \epsilon \cdot (\text{part1 + part2})\]

'Rigid link estimate

\[v = \frac{Wv}{\text{payload}}\]

\[h = \frac{Wh}{\text{payload}}\]

CALL CatenaryHdh(H1, dtower, htower, linewt, T1)

'H1 = H1new

GOSUB PayloadRecalcCC1

\[zCCT1 = 10^6 \times (h - h')\]

WHILE ABS(zCCT2) > 10

\[m = \frac{zCCT2 - zCCT1}{H1new - H1old}\]

\[H1old = H1new\]

\[zCCT1 = zCCT2\]

\[H1new = H1new - zCCT2 / m\]

\[H1 = H1new\]

CALL CatenaryHdh(H1, dtower, htower, linewt, T1)

GOSUB PayloadRecalcCC1

\[zCCT2 = 10^6 \times (h - h')\]

END IF

RETURN

PayloadRecalcCC: 'Recalculate payload estimate for clamping carriage/partial suspension analysis if previous estimates were not within tolerance and Segment 1 tension is limiting

\[T2\text{carriage} = \text{FNtension}(T2, 1, -htail, \text{linewt})\]

\[V2\text{carriage} = \text{SGN}(htail) \cdot \sqrt{(T2\text{carriage}^2 - H2^2)}\]

\[\text{payload} = \frac{(V1\text{carriage} + V2\text{carriage} - \text{carriagewt})}{v}\]

\[h = \frac{(H1 - H2)}{\text{payload}}\]

RETURN
PayloadRecalcCC:  'Recalculate payload estimate for clamping carriage/partial suspension analysis if previous estimates were not within tolerance and Segment 1 tension is limiting

T1_carriage = FN_tension(T1, 1, htower, linewt)
V1_carriage = SGN(-htower) * SQR(T1_carriage^2 - H1^2)
payload = (V1_carriage + V2_carriage - carriagewt) / v
bg = (H1 - H2) / payload
RETURN

SkylineAdjustmentCC:  'Adjust skyline tensions when tagline tension is limiting for clamping carriage analysis with active mainline

Tmax1 = Tmax
zSAC1 = FN_tag - (Maxmainline + htower * mainlinewt)
GOSUB SkylineTensions 'tagline tension and mainline tension
GOSUB MainlineTensions 'at the carriage when the mainline is at the allowable
zSAC2 = FN_tag - (Maxmainline + htower * mainlinewt)
WHILE ABS(zSAC2) > .5 'Secant search to adjust skyline
t = (zSAC2 - zSAC1) / (Tmax2 - Tmax1) 'tension until tension is allowable
Tmax1 = Tmax2
zSAC1 = zSAC2
Tmax2 = Tmax2 - zSAC2 / t
Tmax = Tmax2
GOSUB SkylineTensions
GOSUB MainlineTensions
zSAC2 = FN_tag - (Maxmainline + htower * mainlinewt)
WEND
Skyline = Tmax
RETURN

NewGeometry:  'Recalculates skyline geometry when skyline is adjusted to keep the skyline above the ground profile using the "AdjustSkyline" option

cy = towery + minchordslope * dtower
'a = y coordinate at top of headspar (towery)
'b = slope of line from top of headspar to ground profile where skyline touches the ground
(x, minchordslope)'
x = horizontal distance from headspar to carriage location (dtower)
y = y coordinate of carriage (cy)
IF cy - y(tp) - loglength - taglength - carriage-depth > 0 THEN
   'Calculate whether log is fully or partially suspended
   loght = cy - y(tp) - loglength - taglength
   carriage-depth
   suspension-type(tp) = 1
ELSE
   suspension-type(tp) = 0
   'Calculate height of front end
   loght1 = loght
   zNG1 = carriage - cy
   loght = .999 * loglength
   'Calculate of carriage
   'Calculated above
   GOSUB PartialSuspension
   loght2 = loght
   zNG2 = carriage - cy
   WHILE ABS(zNG2) > .05
      'Secant search to calculate log
      'height to within specified
      'tolerance
      m = (zNG2 - zNG1) / (loght2 - loght1)
      loght1 = loght2
      zNG1 = zNG2
      loght2 = loght2 - zNG2 / m
      loght = loght2
      GOSUB PartialSuspension
      zNG2 = carriage - cy
   WEND
END IF

carriage-y = cy
GOSUB SkylineGeometry
RETURN

LineAdjustment:
   'Calculate the minimum slope of the line
   'from the top of the headspar to the
   'critical terrain point. The critical
   'terrain point is the one at which the
   'skyline will just touch the ground
   'profile
   chordslope = htower / dtower
   IF chordslope < minchordslope THEN
      GOSUB NewGeometry
   ELSE
      minchordslope = chordslope
   END IF
RETURN

UnloadedTension:
   'Calculate tension and line length in the
   'unloaded skyline. The assumption used is
   'that the skyline is loaded only by its
   'own weight.
dO = ABS(tailx - towerx)  'Horizontal distance between supports
hO = ABS(taily - towery)  'Vertical distance between supports
zULS1 = 0
zULS2 = 0
s = minline * FNstretch(Tmax / 2, linewt, hO)
'Initial estimate of stretched line length is line length
'at maximum skyline tension calculated by elastic strain
'formula
GOSUB NewtonRaphson
'Calculate catenary parameter "m" using line length
'estimate
Tupper = linewt / 2 * (s * FCcoth(dO / 2 / m) + hO)
'Calculate upper end tension using catenary formula and
'original value of stretched line length
s0 = s / FNstretch(Tupper, linewt, hO)
'Calculate unstretched line length using catenary
'calculation of tension and elastic strain formula
s2 = s
zULS1 = s0 - minline
interval = .5 * SGN(zULS2)
WHILE zULS1 * zULS2 >= 0
'Enter with binary search and an
'interval of + or - .5 ft. until
'known unstretched line length
'is bracketed
s1 = s2
zULS1 = zULS2
s = s - interval
IF s < minline THEN s = s + .1
GOSUB NewtonRaphson
Tupper = linewt / 2 * (s * FCcoth(dO / 2 / m) + hO)
s0 = s / FNstretch(Tupper, linewt, hO)
s2 = s
zULS2 = s0 - minline
WEND
WHILE ABS(zULS2) > .1
'Switch to secant search routine when known
'unstretched line length is bracketed
m = (zULS2 - zULS1) / (s2 - s1)
s1 = s2
zULS1 = zULS2
s2 = s2 - zULS2 / m
s = s2
GOSUB NewtonRaphson
Tupper = linewt / 2 * (s * FCcoth(dO / 2 / m) + hO)
s0 = s / FNstretch(Tupper, linewt, hO)
zULS2 = s0 - minline
WEND
unloadedskyline = Tupper
unloadedlinelength = s
RETURN
NewtonRaphson: 'Search routine to determine catenary parameter "m"

\[ m_1 = 0 \]
\[ m = d_0 * d_0 / \sqrt{12 * (s * s - h_0 * h_0 - d_0 * d_0)} \]

WHILE \( |m - m_1| > 0.5 \)

\[ f_m = s * s - h_0 * h_0 - (2 * m * \text{FNsinh}(d_0 / 2 / m)^2 \]
\[ fprimes = -8 * m * \text{FNsinh}(d_0 / 2 / m) * (\text{FNsinh}(d_0 / 2 / m) - d_0 / 2 / m * \text{FNcosh}(d_0 / 2 / m)) \]
\[ m_1 = m - f_m / fprimes \]

WEND

RETURN

UnloadedTensionMS: 'Subroutine to calculate unloaded tension and line length for a multispansystem.
'Solution technique is essentially the same as for the above case of the single span (subroutine UnloadedTension:)

d_0 = dSpan(l)
h_0 = hSpan(l)
s = spanlength(l) * \text{FNstretch}(T(l) * 1.1, \text{linewt}, \text{ABS}(h_0))
s_2 = s

GOSUB LineLengthRecalcMS

zULS2 = s_0 - minline

Tdelta = \text{SGN}(-zULS2) * T(l) * .05

WHILE zULS1 <= 0

s_1 = s_2
zULS1 = zULS2
T(l) = T(l) + Tdelta
s_2 = s_2 * \text{FNstretch}(T(l), \text{linewt}, \text{ABS}(h_0))
s = s_2

GOSUB LineLengthRecalcMS

zULS2 = s_0 - minline

WEND

WHILE ABS(zULS2) > .5

m = (zULS2 - zULS1) / (s_2 - s_1)
s_1 = s_2
zULS1 = zULS2
s_2 = s_2 - zULS2 / m
s = s_2

GOSUB LineLengthRecalcMS

zULS2 = s_0 - minline

WEND

unloadedskyline = \text{FNtension}(T(i), -1, -hSpan(i), \text{linewt})
unloadedlinelength = s_0

RETURN


```
LineLengthRecalcMS: 'Calculate unstretched line length of all 
'spans for a multispan system

PARAMETERS:
d0 = dSpan(1)
h0 = hSpan(1)

D0 = dSpan(1)
h0 = hSpan(1)

GOSUB NewtonRaphson

T(1) = linewt / 2 * (s * FNcoth(d0 / 2 / m) + h0)
Tsupport2 = FNtension(T(1), 1, h0, linewt)
unloadedlinelength = s
s0 = s / FNstretch(T(1), linewt, ABS(h0))

FOR span = 2 TO lastsupport - 1
  d0 = dSpan(span)
h0 = hSpan(span)
  T(span) = FNtension(Tsupport2, -1, -h0, linewt)
  CALL CatenaryTdh(T(span), d0, h0, linewt, Horiz)
  unloadedlinelength = unloadedlinelength + s
  s0 = s0 + s / FNstretch(T(span), linewt, ABS(h0))
  Tsupport2 = FNtension(T(span), 1, h0, linewt)
NEXT

RETURN

REM The following print routines are included only to allow
REM testing of the program. It is assumed that a programmer
REM will include a more sophisticated output routine as part
REM of the user interface.

PrintHeader:
PRINT
PRINT
PRINT
PRINT
PRINT
PRINT
PRINT
PRINT syst$;
PRINT " Catenary " ;
PRINT " " ;
PRINT DATES
PRINT LEFT$(syst$, 4) = "Live" AND TotalLifts = 2 THEN
  PRINT "Lifts at TP 3 and TP 7"
ELSEIF LEFT$(syst$, 4) = "Live" AND TotalLifts = 1 AND
  liveoption$ <> "Constant Clearance" THEN
  PRINT "Lift at TP 7"
ELSEIF LEFT$(syst$, 4) = "Live" AND liveoption$ = "Constant
  Clearance" THEN
  PRINT liveoption$
ELSEIF END IF
PRINT
PRINT USING "Headspar ht. = ### ft."; towerht
PRINT USING "Tailspar ht. = ### ft."; tailht
```
PRINT USING "Carriage wt. = #### lbs."; carriagewt
PRINT USING "Carriage depth = # ft."; carriagedepth
PRINT USING "Tagline length = ## ft."; taglength
PRINT USING "Log length = ## ft."; totloglength
PRINT USING "Choker attached ## ft. from end of log";
    chokepoint
PRINT USING "Log diameter = ## in."; logdiameter * 12
PRINT USING "Allowable Skyline tension = ###,### lbs.";
    Maxskyline
PRINT USING "Allowable Mainline tension = ###,### lbs.";
    Maxmainline
PRINT USING "Allowable Haulback tension = ###,### lbs.";
    Maxhaulback
IF RIGHT$(syst$, 8) = "Haulback" THEN
    PRINT USING "Applied haulback tension = ###,### lbs.";
    Thaulback
END IF
PRINT USING "Skyline wt./ft. = #.## lbs."; skylinewt
PRINT USING "Mainline wt./ft. = #.## lbs."; mainlinewt
PRINT USING "Haulback wt./ft. = #.## lbs."; haulbackwt
PRINT
IF liveoption$ = "Lifts" AND LEFT$(syst$, 4) = "Live" THEN

PRINT "TP Payload Payload Critical Suspension
      Log " at TF to Yarder TP Type
PRINT "Clearance"
PRINT
ELSE
PRINT 
PRINT "TP Skyline Mainline Haulback Payload
      Alpha Beta Clearance"
PRINT
END IF
RETURN

PrintAnswer:
IF liveoption$ = "Lifts" AND LEFT$(syst$, 4) = "Live" THEN
    a$ = "\  "

PRINT USING "## ###,###", tp - dir; payload; criticalpayload; limtp;
PRINT USING a$; susp$;
PRINT USING "##", loght
ELSE
    IF LEFT$(syst$, 5) = "Multi" AND supportpoint(tp) THEN
        PRINT USING "## ft. intermediate support"; tp; supportht(ThisSpan + 1)
    ELSEIF suspensiontype(tp) = 1 THEN
        PRINT USING "## ###,###,###,###..#"; tp; Skyline; Mainline; Haulback; payload; fullclearance(tp)
    ELSE
        PRINT USING "## ###,### ###,### ###,###,###.#.#"; tp; Skyline; Mainline; Haulback; payload; alpha * 57.296; beta * 57.296; partialclearance(tp)
    END IF
END IF
RETURN
SUB CatenaryHdh (Horiz, d, h, w, T) STATIC
SHARED s

m = Horiz / w
x = d / 2 / m
s = SQR(h * h + (2 * m * (EXP(x) - EXP(-x)) / (2) ^ 2)
T = w / 2 * (s * ((EXP(x) + EXP(-x)) / (EXP(x) - EXP(-x))) + ABS(h))
END SUB

SUB CatenaryTdh (T, d, h, w, Horiz) STATIC
SHARED s

r = w * SQR(h * h + d * d)
e = d / 2
m = 1
WHILE ABS(m - ml) > .01
ml = m
a = 1 + h * h / d / d
b = 2 * r * ABS(h) * e / d / d
c = r * r * e * e / d / d - T * T
m = (-b + SQR(b * b - 4 * a * c)) / 2 / a / w
x = d / 2 / m
s = SQR(h * h + (2 * m * ((EXP(x) - EXP(-x)) / 2)) ^ 2)
r = w * s
E = d / 2 - ABS(h) / s * (m - d / 2 * ((EXP(x) + EXP(-x)) / (EXP(x) - EXP(-x))))
WEND
Horiz = w * m
END SUB